DESIGNING AN EFFECTIVE USER INTERFACE FOR THE ANDROID TABLET ENVIRONMENT

by

Genevieve Chang

supervised by

Assoc. Prof Michelle Kuttel

MINOR DISSERTATION PRESENTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTERS IN INFORMATION TECHNOLOGY
IN THE DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF CAPE TOWN

February 2015

PLAGIARISM DECLARATTION

I know the meaning of Plagiarism and declare that all of the work in the document, save for that which is properly acknowledged, is my own.

ABSTRACT

With over 1.3 million applications on the Android marketplace, there is increasing competition between mobile applications for customer sales. As usability is a significant factor in an application's success, many mobile developers refer to the Android design guidelines when designing the user interface (UI). These principles help to provide consistency of navigation and aesthetics, with the rest of the Android platform. However, misinterpretation of the abstract guidelines may mean that patterns and elements selected to organise content of an application do not improve the usability. Therefore, usability tests would be beneficial to ensure that an application meets objectives efficiently and improve on user experience. Usability testing is an important and crucial step in the mobile development process Many freelance developers, however, have limited resources for usability testing, even though the advantages of usability feedback during initial development stages are clear and can save time and money in the long-run.

In this thesis, we investigate which method of usability testing is most useful for resource constrained mobile developers. To test the efficacy of Android guidelines, three alternate designs of a unique Android tablet application, Glycano, are developed. High-fidelity paper prototypes were presented to end-users for usability testing and to usability experts for heuristic evaluations.

Both usability and heuristic tests demonstrated that following the Android guidelines aids in user familiarity and learnability. Regardless of the different UI designs of the three mockups, Android guidelines provided an initial level of usability by providing familiarity to proficient users and an intuitiveness of certain patterns to new users. However, efficiency in building Glycano schematics was an issue that arose consistently. Testing with end-users and experts, revealed several navigational problems. Usability experts uncovered more general UI problems than the end-user group, who focused more on the content of the application. More refinements and suggestions of additional features to enhance usability and user experience were provided by the experts. Use of usability experts would therefore be most advantageous in initial design stages of an application. Feedback from usability testing is, however, also beneficial and is more valuable than not performing any test at all.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Assoc. Professor Michelle Kuttel, for her guidance and support during this process and having the patience to work around my full time job whilst doing this thesis. I express my gratitude to Stephen Marquard at CILT for providing valuable advice particularly on balancing my work and thesis.

Lastly, to my family and friends, your support and encouragement has been instrumental throughout my thesis. I am very grateful and love you all.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The popularisation of tablets means the development of tablet-friendly applications has become a focus for many software developers in an increasingly competitive tablet market [1]. The three largest operating systems in the mobile device space include Apple iOS, Google Android and Microsoft Windows. The Google Android framework has, however, become the leading operating system (OS) in terms of the number of devices in use, with 71% of mobile developers developing for Android as of May 2013 [2, 3]. The Android code base can be freely distributed and changed. There are currently over 800 000 applications published to the Android marketplace, surpassing the Apple application store [4]. In the case of the open Android OS, there is a proliferation of both different mobile device hardware, including tablets and televisions, and version releases.

With the large increase in the number and range of mobile applications over the last few years, producing a competitive application has become a challenge. The ubiquity of mobile devices and proliferation of available mobile-development resources provide an accessible platform for new and freelance developers to create applications. However, a successful mobile application has numerous contributing factors that need to be considered during the development phases, such as design for usability and support for hardware fragmentation through scalable user interfaces (UI) [5]. It is also important for developers to consider responsive design (the modification or adaptation of the navigational structure of the UI) and optimising for a small screen (scaling of images so that they fit on different size screens with different resolutions). Fragmentation of many Android-supported devices is a problem for developers, as applications may behave differently on different Android-compatible devices [6]. Customisation costs increase when developing for a range of screen sizes and inputs [7]. Therefore, currently, many tablet applications will install scaled-up versions of the corresponding phone application. Although this may result in a lower resolution UI and poor use of the screen estate, it may be necessary to target a range of devices in order to reach as many users as possible. These factors may require the involvement of various skilled people in the development of the application. However, when time and monetary resources are limited, especially for lone developers, determining the optimal way to assess the usability of their application may be a significant aspect in the initial stages of development.

There is an increasing body of literature on how to design for end-users and create valuable user experiences. Mobile operating systems have developed their own UI patterns and principles to foster a level of consistency amongst applications. The Android UI initially presented with a simple grid design on the introduction of the Android OS in 2009 [8]. With the next few releases, UI changes included subtle tweaks to colours and transparencies, higher resolutions, slight feedback animations and reskinning of the UI for tablets running Honeycomb, a dedicated OS release for tablets. This was due to the lack of physical buttons and a larger screen layout, which introduced the action bar and multi-pane layouts. Previously, the Android UI lacked the rich and distinct look of the current Android design, which became the defining change for Android 4.0 in 2012 [8]. The redesigned and new Android design guidelines [9] focus on UI patterns that emphasise navigation, content selection, notifications and application structure. The guidelines incorporate the OS's basic UI elements, aesthetics and navigational patterns that current users are familiar with and provide for consistent behaviour between applications. Similarly, other OS's have their own design guidelines that make their UI distinct. For example, Apple places menu and action buttons on the lower part of the screen, the shapes have rounded corners without a drop shadow, icons are unique to the OS, and typographies are different [10]. Nonetheless, users of a particular device become familiar with these features and patterns to guide navigation.

Users have an intuitive grasp of standard OS design conventions and patterns help to reduce the cognitive load experienced when switching mental models for different applications and navigational controls [11]. A proper navigational design is crucial so that an application is presented in an intuitive and meaningful manner. Design patterns also formalise the approach to solving UI and communication challenges. Specific UI patterns, however, will not work in every application, but will depend on the context of the application and the goal of the developer [12]. UI guidelines, therefore, provide a solid foundation, including a standard minimum aesthetic quality that allows for a unified development of applications. They provide users with a level of familiarity in their initial interaction with the application, enhance usability and allow the focus to remain on the content [13]. Guidelines can, however, be misinterpreted or applied incorrectly. When the design goals and benefits as well as the conditions for implementing principles are ambiguous, it reduces the impact of a unified design process amongst designers [14]. Misinterpretation of Android UI guidelines could therefore affect the usability of an application.

Usability may be the deciding factor in determining the success or failure of an application. The International Standards Organisation has defined usability as the “extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use” [15]. The five quality aspects of usability as described by Nielsen [16] include the following:

- Learnability: The ease with which a user can accomplish basic tasks upon their first interaction with a system.
- Efficiency: How quickly a user can perform tasks after becoming familiar with a system.
- Memorability: How easy it is for a user to regain proficiency of system.
- Errors: The number and severity of errors a user makes and how easily it is to recover from their mistakes.
- Satisfaction: How satisfactory a user finds the design.

In order to evaluate these attributes of a system, several usability evaluation methods, frameworks and heuristics have become established techniques in system usability analysis [17]. Empirical usability methods utilise real users or experts to assess usability and are the most commonly used techniques when testing IT systems [18]. This method has also revealed more severe, recurring and global problems than other evaluation methods [19].

When assessing mobile usability, several unique challenges become apparent, such as the smaller screen size, data entry methods and the environment in which the device is being used [20]. Various studies have been conducted that compare mobile usability evaluation techniques using interviews, observations, heuristic walkthroughs, laboratory, and field studies [21,22]. It is not always clear whether it is better to use experts or end users in an evaluation. The quality and number of usability problems discovered by these various methods differ between both groups. Where usability experts can contribute insights and identify more severe usability problems, end-users can uncover issues that may not have been considered by experts [23]. This can result in more superficial problems that end-users may uncover, which may not contribute to the aim of the study, for example focusing on the colour palette or font. Usability experts bring in industry knowledge from a usability point of view and experience in end-user behaviour, therefore providing recommendations and problems with each technical aspect of the UI, such as touch gestures and placement of buttons. However, there has been little comparison of end-user and usability expert feedback when OS guidelines have been applied to develop a new application.

1.1 Problem Statement

Given the variety and subjectivity of Android guidelines, the usability of an application may not necessarily be effective for the end-user. Misinterpretations of patterns and rigid ideas regarding the UI design may adversely affect the user experience or are unfavourable to the efficient use of an application. Furthermore, with the constrained budget experienced by freelance developers for the design and development of mobile applications, it is unclear if 1) following the Android guidelines would lead to an effective application and 2) whether experts or end-users are better for testing the usability of Android applications.

1.2 Aim

This study aims to investigate the efficacy of Android guidelines for developing an application and the effects of various interpretations on usability. The commonalities and discrepancies between end-user and usability expert feedback will be evaluated and the designs are refined from the suggestions and recommendations obtained from both groups. The elements that are considered by end-users and usability experts to make an effective UI in a new Android tablet application are assessed.

1.3 Research Questions

The following research questions will be addressed:

1. *Can an intuitive UI be designed for a new tablet application using only Android patterns?*

OS design guidelines are available to create a familiarity with navigational controls amongst Android applications. The use of these guidelines within a new application are investigated to analyse whether the efficacy of the endorsed patterns are intuitive. Will new and current Android users find any of the three UI mockups intuitive?

2. *Is usability testing or expert heuristic evaluations more beneficial for gaining feedback for refinement of an initial application design?*

Interpretation of UI patterns and navigational controls may differ between how end-users perceive it and how usability experts think end-users will understand it. Can end-users provide problems and suggestions to improve their experience with an application prototype, or does the technical knowledge and experience of usability experts provide more in-depth and critical feedback for improvements, even in an unknown application subject?

1.4 Approach

Our approach was to use an example program and design three variations of the UI that closely follow the Android guidelines. This application was used to determine usability problems through the evaluation of the interface by two groups: end-users and usability experts.

The carbohydrate-schematic program, Glycano, was used and its interface redesigned for an Android tablet-based device. This is a unique application: there are currently no similar applications developed for a mobile device and this will therefore reduce bias in the usability testing. Glycano is a Java program that is used to build 2D molecular graphs of carbohydrate molecules using symbol representations, and can be ported to the Android framework for mobile devices [24]. A molecular graph shows the connectivity of a molecule. There are various functions required to produce a schematic of the carbohydrate and these UI controls need to be adapted for the Android tablet platform. The focus is on designing an efficient tablet UI layout for Glycano. A tablet version of Glycano may be a more efficient, as finger gestures and sensitive touch controls may be easier for users to build molecular graphs. Furthermore, as a greater focus on content matter over extensive interface training is paramount for mobile applications, there is a need to make the UI intuitive and improve usability. The use of mobile devices for enriching scientific discovery and exploration is evident in the large number of such applications available on the marketplace. Williams et al. [25] demonstrate the value of using a touch screen device in industry for chemistry and drug related data analysis and sketching of chemical structures. There are also several schematic creation applications available for example drawing of engineering circuitry and mind maps. A similar technical approach has not yet been applied to chemistry applications, which could prove to be an efficient means to construct carbohydrate representations.

Usability testing was done using three varied paper prototypes specifically designed to follow the Android design guidelines as closely as possible. Navigational styles, UI layout and application structure are varied between the designs and focus on the search and selection of residues and editing of the structure. The mockups were also provided to usability experts in industry along with a heuristic survey and questionnaire . A series of adapted usability heuristics and Android design guidelines were included in the survey and involve learnability, predictability, familiarity, memorability and efficiency evaluations. Severity ratings for each heuristic were assigned by the reviewers so as to quantitatively score where the designs fall short. The expert reviews and end-user feedback will be discussed and compared. Disparities and commonalities between the groups, regarding the important elements of the UI will be covered. Two refined designs were produced based on recommendations from end-users and expert reviewers, respectively, to conclude the study.

1.5 Thesis Structure

This thesis is organised as follows:

Chapter 2 covers an overview of Android guidelines and design principles. This is followed by a discussion and explanation of its use in the three alternate designs created for the Glycano application.

Chapter 3 is a review of usability evaluations in mobile applications and touch devices, which includes user-testing and expert heuristic evaluations. The evaluation of efficacy of an application's design through various usability testing is investigated, with justification for methods used in this study.

Chapter 4 provides an overview of the Glycano application, explaining its purpose and function. A necessary component is a brief introduction to carbohydrates. A look at established and new symbol representations and nomenclature used in illustrating the characteristics of glycan structures are covered.

Chapter 5 details the design rationale and methodology in creation of the prototypes for usability testing.

Chapter 6 describes the testing methodologies and their evaluations of the end-user and expert testings.

Chapter 7 describes the testing environment and results of the usability testing. The refined design based on end-user recommendations and feedback follows.

Chapter 8 describes the testing and results of the expert heuristic evaluations. The refined design based on usability expert recommendations and feedback follows. Comparisons between end-user and expert feedback conclude the chapter.

Chapter 9 compares the results and refined designs of the end-user and expert evaluations.

Chapter 10 presents conclusions of the study and avenues for future work are covered.

CHAPTER 2

ANDROID DESIGN GUIDELINES

The Android development environment is an open source technology that encourages community development and growth of applications within the Android marketplace. This means that all parts of the platform, including the SDK and OS source codes, are accessible to developers [7]. Reusable code and resource sharing amongst developers reduces the cost and time of development for applications.

Mobile operating systems like Android encourage the use of their own UI patterns and principles to foster a level of consistency amongst applications. These principles cover aspects such as navigation, selection, notifications and content organisation. Platform-specific patterns are formalized approaches to solve common UI problems in order to meet users' needs. Design patterns are foundational building blocks that have undergone many iterations of development and have been found to work with multiple different applications. This not only provides a unique identity for the OS (for example, Google prefers developers to not emulate the design elements of other OS's) but also provides consistency in the navigational controls. It is acknowledged that regular users rapidly acquire an intuitive grasp of standard OS design conventions and recognise patterns. This reduces the cognitive load experienced when switching between different applications. A consistent application structure with regards to navigation, selections and notification elements, allows the application to conform to the rest of the platform and other applications, creating a coherent user experience. Therefore, design principles provide the user with a familiar UI in their initial interaction with the application, enhance usability and allow for focus to remain on the content. Although UI guidelines do not require strict adherence, straying too far from them may result in users experiencing a visual disconnect from the platform that they use on a daily basis. This may also be applicable to scientific applications where focus on content and functionality is more important than visual appeal.

With the low barrier to entry in Android application development, more freelance and beginner developers are placing their software on the Google application marketplace. Many developers will use the UI guidelines as a core reference when designing their UI, as patterns and building blocks can utilised together when constructing the navigational controls. However, usability practices and the appropriate selection of UI elements can be misinterpreted. The initial understanding of the UI can therefore differ between the developers, designers and end-users .

This chapter introduces touch interaction with tablets, a brief history of Android UI design and the principles defining Android design. Following this, examples of scientific applications on the Android platform are reviewed, as well as their use of the Android Guidelines. The aim is to view visual and navigational differences between real applications that have similar objectives to this study's case application (Glycano). This study will analyse if an effective UI is possible using minimal but close adherence to Android UI practices on a carbohydrate schematic application. The main focus will also allow for the possibility of improved of user experience, with different interpretations of the guidelines by conducting usability tests on multiple mockup UI interfaces.

2.1 Touch Interaction with Mobile Devices

The use of touch interactivity has shown to be an intuitive input method that maximises screen estate and decreases the use for physical buttons [26]. However, the efficiency of input and output of data from touch screen mobile devices has more challenges than with the use of conventional PC's. The smaller screen size needs to fit in the same amount of information with various touch controls that replace for example, mouse and keyboard inputs. The size and position of visual targets, such as buttons, therefore need to be large enough for the finger to accurately press yet avoid possibly hitting adjacent sensors or being perceived as a non-interactive element [26]. Additionally, the finger may also cover parts of the screen that displays the resultant interactivity meaning the user could overlook valuable device feedback. The size of hit targets, the position of these elements next to adjacent buttons to avoid interference from other buttons and visual feedback are therefore critical for proper UI controls on touch screen devices.

Important user actions for touch screen devices are gestural controls that prompt various functionalities. These gestures are a pattern of touches on the screen to trigger an effect [27]. For example, horizontal and vertical scrolling requires a swipe or drag gesture (Appendix 1). To accomplish this, the finger must press, move and then lift. This moves the on-screen content across or up and down the screen and depends on the direction and speed of the swipe. The drag-and-drop gesture allows for rearrangement of content across the screen. It is achieved by holding the finger down or long-pressing on an item, moving and then lifting. Both these gestures are used within the case study application and, although scrolling may be common to experienced users, the drag-and-drop feature may not be as intuitive. Another common gesture, using multi-touch gestures, is the zooming action, commonly used in mobile devices (Figure 2.1). This is when the user wants to increase the size of the object or text on the screen by initiating the pinching or spreading behavior with 2 fingers. The scaling and speed of the zoom also corresponds to the degree of the finger gesture.

Figure 2.1: The pinch gesture for zooming in and out of the screen [28]**.**

There are a library of common gestures that exist for touch screen devices that allows users to view and interact with the content. These mobile specific patterns are important when designing a UI as it takes into account mobile ergonomics, or the way users hold the device and ability to reach elements, and positioning interactive fields avoid these constraints. Each OS utilises its own functionality for standard gestures [28]. For example, the long-press in an Android device will bring up the contextual menu whereas in an Apple device, it places a cursor in a magnified view for text. Zooming in can also be done through using one finger to double-tap the screen in Android, but the double-tap in an iOS device requires three fingers. Ensuring that the correct gestures are used for the platform being used also maintains the consistency of user interaction and experience with the application. Touch screen gestures are therefore an important aspect when designing the UI to ensure that the interactivity between the user and the content is a seamless process.

2.2 Android UI Principles

Early UI guidelines for Android applications were neither comprehensive nor well defined. Prior to 2011, Android did not have a defined design identity in contrast to Apple, which has a distinct design aesthetic [10]. The principles and patterns for Android mobile application were introduced with Android 4.0 (code name Ice Cream Sandwich) with the goal of making Android visually unique and therefore recognisable. This version allows for consistency between applications, as well as across future OS version releases.

Distinctive and more refined UI elements were introduced, giving the OS a different look and feel both from previous versions and from other OS's (Figure 2.2). Previously, Apple's dominance with regard to the number of applications available on its marketplace, meant that many applications were ported to Android along with their existing design pattern. The Android UI guidelines cover the OS's basic UI elements, aesthetics and navigational conformities. Current Android UI components have a more sleek design than before, with attention paid to the typography, layouts and transitions, imposing a standard minimum aesthetic quality on the application. The Android platform provides maintained library support of these pre-built UI components. The official default system themes (termed Holo) cover the styling of UI elements, it includes colour, padding and font sizes. Holo can be used as it stands as a customizable template when designing the UI for an application. Android guidelines encourage use of these patterns in a way that makes the application intuitive to the user and makes it easy to switch between applications with similar navigational controls.

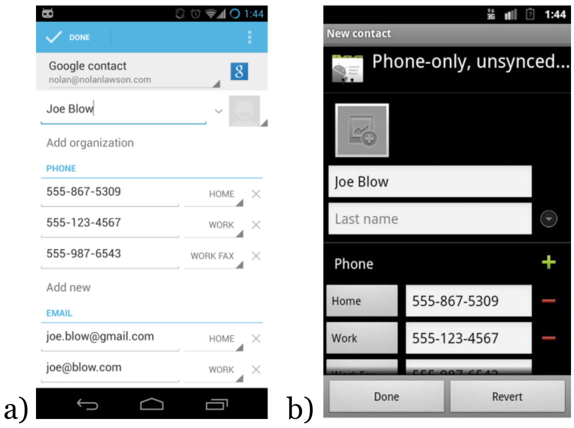


Figure 2.2: Android's new UI a) using Holo Light theme, after the release of Ice-Cream Sandwich, compared with the previous UI aesthetics of b).

The design principles expressed in Android's official documentation are subdivided into three classes (Appendix 2). The first group of principles, entitled "Enchant me" covers visual and animated feedback, as well as customisation options. A large part of the appeal of the Android OS is the ability to personalise home screens, and this feature is evidently encouraged in its supported applications. Efficiency of navigating the UI is also critical, where the use of picture or icon representation to replace text is considered ideal. This is further expressed in the group of guidelines termed "Simplify my life", where ease of use and an intuitive interface are key principles. For example, only essential features should be made visible, user-created content should be saved and the number of decisions in order to complete an action should be minimised. The last class of principles, “Make me amazing”, aim to make achieving tasks easier. These include using visual patterns to help the user remember navigations, corrective feedback for incorrect actions made as well as creating shortcuts for important and frequently used features.

Together, these design principles consider the application from a user’s perspective and aim to improve the user experience. Close adherence to the guidelines is therefore encouraged when designing an application, so as to allow the “look and feel” of the application to fit the platform. The ultimate usability of an appropriate implementation of these principles may, however, depend on the UI components selected. The following section explores some of the common UI patterns that make applications uniquely Android and that are recommended to solve common usability issues.

2.3 Android Design Patterns and UI components

The patterns and UI elements that form Google’s support library and address various application design issues are extensive. The following patterns and UI elements described are those that have been used within the context of this study (Appendix 1). They are separated into two categories, those that address user interaction and those associated with navigation and layout. As responsive design is an important aspect of mobile application development, the UI elements are given in both the context of a small phone screen and large tablet screen.

2.3.1 User Action Patterns

User action patterns dictate how users interact with the application to trigger an event from the software. In making an efficient application, the various user actions that can be performed should be conspicuous and intuitive with easily identifiable functionalities.

The action bar, located at the top of the application, is the main element that contains the menu options and important features (Appendix 1). After the discontinuation of the menu button with the release of Android 4.0, menu options have been relocated to this construct. The action bar has become an integral part of the Android UI identity. This unified container holds consistent functionality that users have become familiar with, through interaction with numerous Android applications. It is separated into four sections that can be used independently: the left side of the bar is reserved for the application icon, which also can navigate to previous screens or open a side navigation menu, while the view control part of the action bar holds the title of the application and navigational controls (such as drop-down lists). A series of quick action buttons are placed on the right, along with the overflow menu, which holds secondary functions. Action buttons that are used frequently, are important or are typical of the application are expected to be displayed at the top-level or visible on the action bar. Other functionalities can be placed in the overflow menu to prevent clutter. With larger screen sizes, the additional space created on the action bar allows for extra action buttons to be displayed.

Another feature of the action bar is its overlay (known as the contextual action bar), which holds actions or sub-tasks for individual items. The contextual menu is triggered when a single object or text item is selected through the long-press gesture. Multiple items can subsequently be selected by a single tap and then manipulated through the action bar mode. This is where related actions for selected items are displayed on the contextual menu while, in previous versions, pop-up notifications for single items were used predominantly. This custom overlay allows for quick actions on a single item. It is used for sub-tasks that do not require many features or functionality and may be an efficient method of manipulating objects. Although the pop-up option is still used (with updated aesthetics) the contextual menu holds more functionality, such as multi-select.

2.3.2 Navigation Patterns

Navigation patterns are UI elements that represent how the information is displayed and related to the content. It includes the menu displays and overall application layout of visual constructs. In the context of this study, fixed tabs, drop-down lists, popup notifications and various tablet layouts are used.

Fixed tabs are displayed within the action bar and provide a tabbed interface for navigation (Appendix 1). Tabs hold different menu options or categorised views and allow for quick navigation between these with a single tap. Each tab has the same width and height and, as they are not scrollable, the number of tabs will depend on the screen size to make optimal use of space.

List navigation (or “spinners”) is an alternative to a tabbed display when menu options become lengthy. It appears as a drop-down list button and can be used for either selecting an option in a form or to change the screen view. However, when view switching is frequent, tabs may be more efficient [29].

Multi-pane layouts are used with larger tablet screen sizes where there is more screen estate. Using the space more efficiently makes navigation easier. This view is based on a linear layout, where all panels are arranged in a linear fashion either horizontally or vertically. The left pane is usually reserved for selection of options, while the right panel displays the relevant information.

Although guidelines are not mandatory, many developers consult them as a foundational starting point when designing their applications. This aids in their application leveraging off common visual patterns that many existing users have become familiar with and allows their application to fit in with the rest of the platform. There are many user actions and navigation patterns that can be chosen when designing the UI of an application. The predicament lies in choosing the optimal patterns to not only adequately organise the content, but to create an efficient and intuitive user experience.

Android design elements also take into account usability with the hardware it runs on. For example, the back button on Android devices is a software key rendered by the OS unlike the Apple devices that lack this feature. In this regard, the up button element was developed, which appears in the top left corner of applications (Figure 2.3). This, however, may lead to confusion as the icon is represented by a left-facing arrow. This is similar to that of the OS-rendered back button that serves a different navigational direction. Where the back button will go to the previous screen that was viewed or exit the application, the up button will take the user to the screen one level higher in the navigational hierarchy.

Another example includes the discontinuation of the dedicated menu button that many Android devices had prior to the release of the Android Honeycomb version, a tablet-specific UI. The action overflow button in the action bar was introduced instead to hold context-dependent list of options. This maintains the consistency across devices that may or may not have the menu button hard coded into the device. Furthermore, features within the menu button may not be easily discoverable and developers need to consider layout options to make primary features transparent. Too many single touches to reach a feature is discouraged.

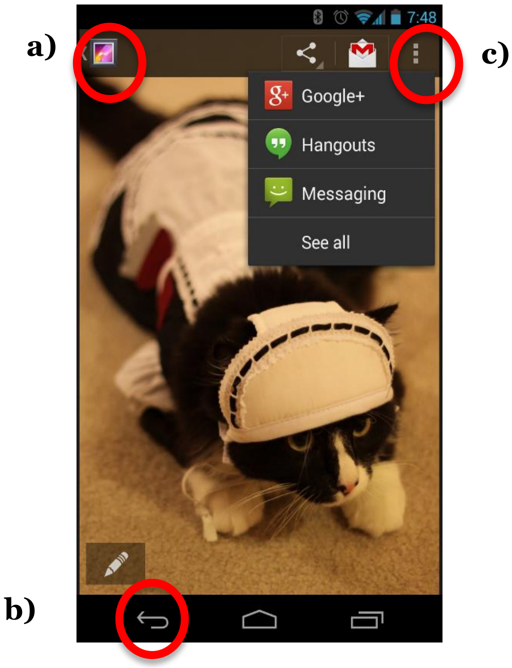


Figure 2.3: The software rendered a) back button versus the b) up button, represented by a left-pointing caret. Even though similar in appearance, the navigational destination screens differ. c) The overflow button holds secondary user options that the legacy menu button used to house.

2.4 Android Guidelines in Practice: Focus on Chemistry Applications

The application used in this case study can be classified as scientific. The objectives of scientific applications include exploration of data or construction of schematics and complex shapes for scientific purposes. There are currently many scientific applications available on the Android marketplace. Here we review a sample of applications where content and the end-user are similar to the case study, which focuses on the building of diagrams (where manipulation of shapes is necessary) or on chemistry applications.

A search amongst popular schematic-type applications presents a range of UI patterns inconsistent with the rest of the Android platform. Where several applications have utilised customised or outdated interface designs, recently updated applications have implemented the distinct Android patterns (Figure 2.4).

In Schematic, an application used for the creation of circuit diagrams, the main screen is designed using a multi-pane layout, with the editing icons placed on the left panel for quick access (Figure 2.4a) [30]. The action bar contains an additional side-swipe menu, indicated by the three aligned carats, and secondary functions that include the text description next to its icon. The popup contextual menu is used to select single objects. Most of the screen estate is dedicated to the user-generated content and this application has a clear and intuitive layout. Android guidelines are evidently integrated with the objective shapes construction, however, this is a paid application and therefore receives fewer user ratings and comments than free applications due to the number of people willing to pay for an application. The efficacy of the UI is evident in some of the comments, for example “I’ve been doing CAD for 8 years and this is very easy to use for quick designs”, “A clean and simple interface with lots of power” and “User interface is built for tablet use and works great after you get used to it and you get the settings right”. A similar and free application, SchematicMind Free Map, also incorporates some Android guidelines and has a much larger user base according to the number of downloads [31]. The application is used to create mind maps where shapes are connected to each other to form a map of organised ideas. The main pattern used is the action bar and Android icons with the option to customise the aesthetics of the content (Figure 2.4b). It installs with help documentation and the main form of adding new content is through long-pressing the background, which may not be intuitive without first reading the tutorial.

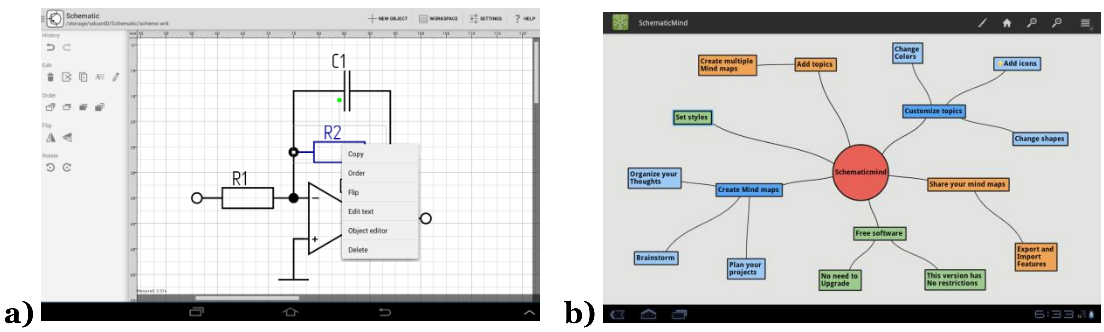


Figure 2.4: The Android schematic applications a) Schematic and b) SchematicMind Free Mind Map, which utilise endorsed Android UI patterns.

Popular applications with a chemistry-focus typically do not follow Android guidelines as closely. There are many applications for protein construction and molecular structural analysis. Many of the highly rated chemistry visualisation applications, however, do not appear to be consistent with the Android platform, as evident by their UIs (Figure 2.5). The scientific applications that follow Android guidelines are obvious through the use of the action bar, drop-down menus and multi-pane layouts. The applications that do not make use of Android patterns, appear outdated and are have an inconsistent look to the rest of the platform. Many new adopters of the Android platform may not be familiar with old or customised layouts, which may influence the uptake of the application. However, with niche applications, such as visualisation and chemistry, where aim and outcome is the primary concern, an outdated UI may be overlooked or go unnoticed. This is evident in the analysis of four chemistry-visualisation applications on the Android marketplace (Table 2.1). All four applications view either 2D or 3D structures of molecules on a database, have over ten thousand downloads and were last updated after the release of Android 4.0 guidelines. Even though these applications either use old Android aesthetics or a customised UI, the user ratings have scored highly. There is a lack of similar applications with updated Android guidelines, however, users have found these applications to be useful even if it means learning a unique UI first.

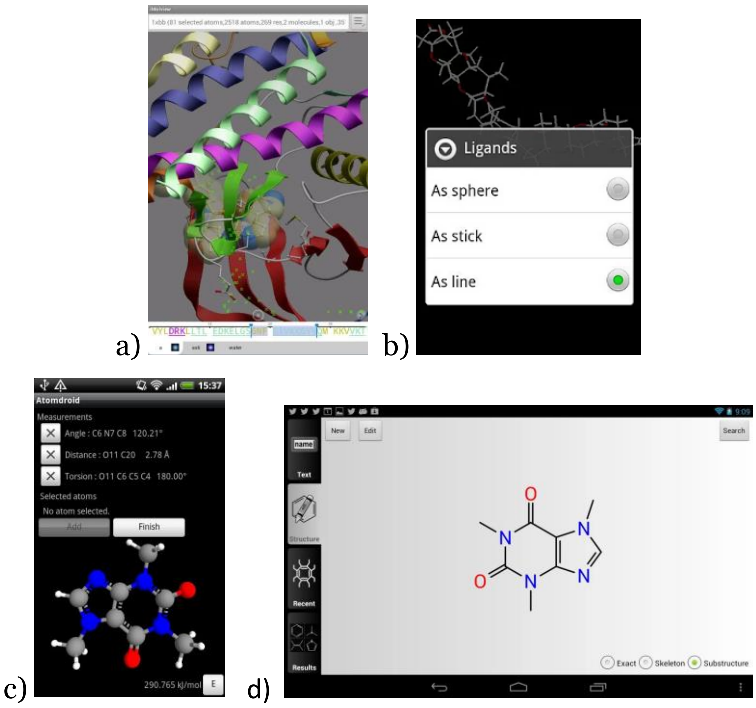


Figure 2.5: Chemistry applications, a) iMolview Lite and b) NDK-Mol that do not follow Android guidelines, c) Atomdroid Molecular versus d) ChemSpider Mobile that has Android components [32]–[35].

Table 2.1: Comparison of UI design patterns and user ratings of four popular chemistry Android applications.

Application	Design Patterns	Rating	Last Updated
iMolview Lite	Customised UI and icons, tablet and phone layouts	4.5 (210 votes)	June 2014
NDK-Mol	Previous version aesthetics with outdated popup menus	4.5 (171 votes)	July 2013
Atomdroid	Previous version aesthetics using large tabbed menu similar to iOS, outdated look of the buttons, sliders and icons	4.3 (219 votes)	June 2013
ChemSpider Mobile	Customised layout and menu, buttons and icons	4.2 (163 votes)	May 2013

An advantage of having similar applications in the marketplace is the user feedback, which not only uncovers aspects where the application falls short, but also supplies ideas of features that end-users would consider valuable. The following comments are from users of various Android chemistry visualisation applications, which have also aided in the methodology of this study:

- "For a chemoffice user, it's complicated to use. Multi-layer menu always makes me confused. The UI is terrible. Is it very hard for the developer to list all of the buttons on the top of the UI?" – Mobile Molecular DataSheet, iPad

- "More multi-touch support would also be nice: 2 finger drag to move (similar to how you change pitch in google earth), and 2 finger rotate to allow rotations in the axis perpendicular to the screen. I can see myself using this app in meetings with my supervisor." – NDKMol – Molecular Viewer, Android

- "Very smooth, except the atoms look a bit too chunky (adjustable level of detail perhaps)? Also, the dark gray background does not offer the back contrast (try to see the fullerene on it). An adjustable background color would help. Last recommendation: leave the molecule name somewhere in the window" – Molecule Viewer 3D, Android

Due to a distinctive set of functionalities and requirements that are involved in creating chemistry applications, the focus on content and features over UI and usability aspects becomes apparent. This is easily noticeable on most chemistry applications on the Android platform, where a lack of standard UI patterns and customised interface is evident. However, the constructive feedback received on these applications indicates that users expect a certain standard and consistency, even with applications that do not have many competitors. An intuitive UI and an efficient means of constructing molecular structures are therefore critical, as users prefer to focus on content rather than on learning how to use the UI.

2.5 Prototype Fidelity

Various prototyping approaches for mobile applications have been studied in order to investigate which provides optimal benefits of early usability testing. However, the decision of which usability testing method employed in practice is usually influenced by time limitations and budgetary constraints. Paper prototyping is a common technique used early in the design phase that makes potential usability problems of the UI discoverable before actual implementation begins [36]. User feedback is obtained through interactions with low or high-fidelity mockups of the application and the designer responds by switching the paper elements to match what the real system would do. Here, the fidelity of a prototype refers to how far or exact an early representation of a product design is and can include different mediums, such as on paper or software [37]. This method has been adapted and widely used in mobile usability testing, particularly using low-fidelity prototypes involving sketches or outlines of the device.

Iterative user and prototype testing are a common and preferred method for testing amongst usability practitioners, with low-fidelity prototypes proven to hold a similar level of sensitivity in uncovering usability problems as its developed software counterparts. Studies have shown low-fidelity mockups to be an easy, fast and low-cost method in early usability testing [38], however, they may not reproduce a realistic enough representation or testing experience of mobile devices and their associated controls and required input gestures. On the other hand, a very high-fidelity prototype can be time consuming and costly. The “breadth of functions” and “aesthetic refinement” therefore need to be considered in the prototype development, as these factors can influence user behaviour and evaluations.

When considering the use of high-fidelity versus low-fidelity prototypes for usability testing there are advantages and disadvantages to both. Mixed results are observed amongst various studies comparing different prototype fidelities and its influence on participants' critical and emotional feedback. However, in touch-based mobile testing, the size of controls and limited amount of detailed information are factors in usability and realistic prototypes have been shown to provide a better testing experience and improved evaluation results.

Various studies have shown that the main limitations of high-fidelity mockups are that they are time consuming to create; that they cause reluctance to change the design and that users may focus on aesthetics over usability [38, 39]. High-fidelity prototypes do, however, convey the wide range of UI possibilities effectively. As aesthetics, including colour, typography and affordances, are incorporated into the Android guidelines, this should add to the usability testing of the application rather than deviate users.

2.6 Discussion

Although current users of Android devices may be the main beneficiaries of UI guidelines, they may not necessarily recognise certain UI elements when they are placed in a different application domain or placed in a contextually dissimilar software. In these cases, the users may find that the new layouts disorienting and the properties of such constructs in a new setting can be easily confused. Furthermore, UI patterns need to cater to both expert and new users to the OS, in that the interface should be efficient when accessing features but also have a level of learnability and intuitiveness. If such guidelines and patterns are meant to be consistent across applications, then recognition and learnability of these common patterns should be easily attainable at any user level for any type of application utilising these elements.

For many casual developers, the Android marketplace is a strong platform to distribute their software. However, limited resources and expertise to produce a good UI can reduce the application's potential due to different OS aesthetics. Developers are therefore encouraged to follow the set Android UI guidelines to streamline user experience and improve the application's aesthetics. The production of UI style guidelines and patterns in combination with usability principles, however, is a lengthy and subjective process. The interpretations of these guidelines by the growing number of developers and designers, however, may not necessarily carry through to a standard level of usability in their applications. Even though users of each platform have an intuitive grasp of design conventions that they interact with daily, they may still face a learning curve when it comes to familiarising themselves with navigational controls of an application. The placement, layouts and selection of appropriate UI patterns may not be recognisable within unconventional applications. Misuse of elements and miscomprehension of conceptual principles may result in poor usability of an application, even with the best intentions of the developer. This study investigates if following Android guidelines closely will also aid in the usability in unconventional applications.

CHAPTER 3

USABILITY EVALUATION IN MOBILE APPLICATIONS

There are many factors that contribute to the successful implementation and acceptance of an IT system, with usability being a significant aspect. Jakob Nielsen describes usability as having the following five quality factors [16]:

- Learnability: The ease with which a user can accomplish basic tasks on their first interaction with a system.
- Efficiency: How quickly a user can perform tasks after becoming familiar with the system.
- Memorability: How easy it is for a user to regain proficiency with a system.
- Errors: The number and severity of errors a user makes and how easily it is to recover from mistakes.
- Satisfaction: How satisfactory users find use the design.

Major design problems can adversely affect the quality of an application and prevent users and put off potential buyers. Investment into the design of the UI architecture is therefore essential, with usability testing being a fundamental step in the program's development process. Usability techniques, or user evaluation methods (UEMs), evaluate usability through the measurement of various factors, such as users' needs, methods of interactivity and the time taken to complete tasks efficiently. The aim is to improve the UI design and overall usability of the product. Testing software usability can be conducted with standard usability techniques. Usability methodologies can also be adapted for a study. An example is Abran et al's Consolidated Usability Model [15] that combines various ISO usability models into one Nielsen [40] categorised usability evaluation methods (UEMs) into the following four basic groups:

1. Automatic (software is used to analyse specific usability measures of the UI)
2. Empirical (real users or experts are used to assess usability)
3. Formal (exact models and formulae are used to calculate usability measures)
4. Informal (based on the general skill and experience of users)

With single person or small developer teams there are limited resources for usability. It is therefore important to identify key methods for optimising usability when this is the case. There exists a large body of literature on the efficacies of various UEMs

[18, 40, 41]. Automatic evaluations aim to automate aspects of usability evaluations such as capture, analysis or critiquing of data through software [43]. This method, although reduces evaluation costs and is consistent in uncovering errors, is not used often due to its inability to capture qualitative and subjective information, such as user opinion [43]. The formal usability inspection method requires the use of a cognitive walkthrough, with use cases and task scenarios, by the interface designer and experts [42]. The aim is to uncover the chief problems in the UI. However, this method is not scalable and is not easily applied [18]. Informal methods are cost-effective techniques for summarising general opinion from target users [18]. Empirical evaluation methods are the most common techniques used in testing IT systems and software. They involve usability testing with real users. Jeffries [44] showed that empirical studies revealed more severe, recurring and global problems than other evaluations. Empirical evaluations are divided into the user testing and system inspection methods, as follows.

3.1 User Study Method: Usability testing

User study testing involves the direct testing of participants and is the most commonly used and established method of usability testing [18, 43]. The method requires target users to complete typical tasks in the application in order to expose any UI flaws. This ensures the system is adapted to end-users' requirements and goals. Various techniques can be employed when testing end-users, including laboratory testing (in a controlled environment), thinking aloud (saying out loud any issues and thoughts), guidelines (checklists) and heuristic evaluations [46]. An advantage of the user-based method is that designers and developers can observe users interact with their system and take note of any difficulties that they may have with the design. In measuring efficiency, users are timed and recorded as they navigate the system and complete set tasks. Design and usability recommendations can be proposed subsequently and changes made before production of the real system. User testing is widely known to be beneficial in early software development stages and those with minimal resources can perform such analyses with valuable outcomes [36]. However, disadvantages include the time required to build trust between the end-user and evaluator, long preparation times of the hardware or software, as well as malfunctions that may occur during testing [47]. Furthermore, certain methods employed, such as "thinking aloud", may be unnatural to end-users and difficult for usability experts [46]. Severe problems may also be overlooked and results may have a level of test-user subjectivity. User-based evaluations are, nevertheless, low cost and effective in identifying recurring problems, and have been shown to have a large impact on product and strategic improvement [48].

User testing has several requirements that contribute to its validity, due to the varying results that can occur with different tests and methods employed [49]. The validity and reliability of the test are also influenced by other factors. These include the qualification of test subjects; the tasks users are required to complete and their measurements; test materials and test environment; the instructional design or test protocol;l and final data analyses that add to test constraints [36]. For example, the fidelity of the testing situation includes the type and number of participants: too few can yield inadequate results and too many may be superfluous. Other variables to be considered are the type of participants, and their true representation of the final end-user, the use of low- or high-fidelity hardware or software prototypes, set tasks that are too complex or too easy, and an artificial testing environment [49]. Even though usability testing may not be truly reflective of the application in a real world scenario, it is considered better than not performing a test at all, as end-users can provide valuable feedback in improving the design at an early stage.

3.2 Usability Inspection Method: Cognitive Walkthroughs and Heuristic Evaluation

Usability inspection methods (UIMs) are a subset of evaluation methods that examine the software's usability and faults in a design, rather than user behaviour. This is a cost-efficient and easy testing solution and can be carried out iteratively during the application's early developmental process by a small group of usability experts. The focus is on discerning the severity of any user interface (UI) problems. There are several testing methods that utilise different procedures. For example, pluralistic walkthroughs and consistency inspections use a group of experts to analyse scenarios or compare functionality to their own software [18]. Heuristic evaluations and cognitive walkthroughs are informal methods that require a single evaluator to step through tasks through the point of view of the user [18]. Each step is assessed against several usability principles.

Heuristic evaluations are an informal method used to determine usability problem areas according to a set of guidelines developed by Neilsen and Molich [50]. Cognitive walkthroughs and heuristic evaluations are similar in that both utilise the industry-acquired knowledge of expert evaluators. Heuristic evaluations, however, provide a holistic view of the software's usability. Cognitive walkthroughs focus on user goals and learning the interface of the application. They also involve a more exploratory procedure, in which the expert steps through core tasks with the perspective of the user in mind [18]. Detailed mockups of the design are provided, without the need for a fully functional prototype. Experts view user task scenarios and move through the sequence of steps required to complete the objective, accessing possible user response and behaviour [51]. An advantage is that the method can be used early on in the development process, a disadvantage is that experts may not always assume users' actions correctly or may overestimate a users' knowledge of the system [52]. Cognitive walkthroughs are recommended as a supplementary test [52].

In a study by Jeffries [44] comparing four usability assessment methods, heuristic evaluations revealed more problems than user testing [53]. A disadvantage of heuristic evaluations is that UI specialists are required for this method of testing. Obtaining several usability experts for one testing session alone has been shown to be quite difficult and costly and results produced more minor UI problems than severe ones [53]. Minor problems included the order of menu items and confusing terminology or phrasing of labels, as opposed to major problems such as delayed error messages, unconstructive feedback messages and users likelihood of forgetting input requirements [53]. This in-depth analysis an interface can not only be time-consuming, but recommendations can either be provided too late in the development cycle to allow for any major changes to be made, or may be technically unfeasible. Outcomes of heuristic evaluations have been shown to be highly dependent on the number of experts recruited and their level of usability expertise. In comparing novice users, regular experts and double usability experts (users proficient in both usability and the interface evaluations), it was observed that double experts revealed the best insights and more usability problems than the other two groups [53]. Acquiring several users with the required skill and time to complete heuristic evaluations, however, can be difficult. Nielsen concluded that heuristic evaluations serve as a cheaper method for predicting user interface problems than empirical studies, although later studies show heuristic evaluations had identified fewer UI problems than field experiments [54][55].

3.3 Usability Testing for Mobile Devices

There are a growing number of studies that investigate the appropriate models for mobile device testing [22, 54-56]. Such usability studies need to account for the hardware constraints and the minimisation of external factors that can adversely affect user experience, such as any distractions [20]. These limitations include a smaller form factor, low processing power and bandwidth usage, as well as the limited attention span of the user while using the device [57]. Various methodologies and evaluation techniques have been developed. In general, mobile usability studies have either been on the physical handling of the device (incorporating field studies) or on the graphical UI (focusing on presentation or prototype methods) [59]. Usability testing of the physical mobile phone hardware usually involves field studies, which views users' handling of the device in a naturalistic setting, or are conducted in a laboratory-based setting [22]. Such approaches are used due to the need for mobile devices to be ergonomic and easily portable [22]. In field-testing, special equipment is worn by the user, including cameras and microphones to capture the handling of the device while the user is in motion and performing tasks. An evaluator may sometimes also supervise the participant during the testing. The efficiency and success of tasks are assessed to observe if environmental factors affect user concentration and performance while interacting with the system.

In a more controlled environment of the laboratory, experimental controls can be managed with guidance from the tester. This includes using a traditional setup, such as subjects seated at a desk with phone in hand, and fixed cameras can be used to capture data [21]. This allows any hardware problems to be observed, such as screen utilisation, interaction mechanisms, mobile processing speed and the overall design [60]. An interview and questionnaire session follows to gather the user's opinions on the device for example, feedback on the screens size, system speed and text size [57]. Evaluations are performed as discussed in Chapter 3.4.1. Laboratory-testing is less time consuming than the logistics associated with field testing, and may also be used for heuristic walkthroughs by a group of usability experts. Furthermore, laboratory testing has shown to be as effective as field testing when testing a mobile UI and provides sufficient usability data [61]. In a study by Kjeldskov et al., an overlap of critical usability problems of a mapping application were uncovered in both field and laboratory testing regarding the navigation of the system, relevance of the information and overall design [21]. These results, however, were believed to apply to mobile guidelines in general.

3.4 Summary

The Android marketplace is an easily accessible platform on which developers can publish their mobile applications. Individuals who create many of these applications more often work alone, rather than in large teams, and may lack the time and monetary resources to perform usability tests, which would be beneficial in improving the rating and download numbers of the application on first release. As usability testing is usually done within early stages of development, it would be advantageous for such freelance developers to perform these tests before development begins, to maximise the benefits of testing [62]. Costs are reduced when major developmental changes are avoided, a greater user experience can be achieved through correcting usability problems and promote innovation [58]. Of all the established UEMs, the two most important and accessible tests that can be conducted are user testing and expert evaluations. Given limited time and resources, both testing methods uncover major usability problems. End-user testing requires more participants to provide sufficient data, but allows the developer to assess if the application matches the goals and requirements of the user correctly and efficiently. Expert heuristic evaluations may uncover more severe usability issues due to industry knowledge and experience that is brought into the analysis of the application but fewer participants are needed.

CHAPTER 4

CHEMICAL SYMBOL NOTATION AND PRIMER ON GLYCANO

This study aims to design a tablet version of an existing software package called Glycano. Glycano allows building of 2D graph representations of carbohydrate structure using a symbol set. Carbohydrates occur as simple and complex biomolecules that serve as mainly energy stores and structural materials in nature [63]. Due to the many specific linear and branched structures they can form, a software application, such as Glycano, can be used as a tool to create carbohydrate structural graphics. Glycano's target users are chemistry students and researchers who require a light-weight mobile application to create carbohydrate representations for learning, exploratory or research purposes. Recognition of spatial orientations and various combinations of residues required to form a specific molecules are important aspects of carbohydrate structural understanding. Such an application would therefore be beneficial when a student wants to efficiently map out a pictorial of a carbohydrate molecule during their studies. Glycano allows for the production of uniform and editable schematics of the molecule, however, such programs require an initial training time in order for the user to navigate the interface before any user-generated content occurs. The tablet device is a popular gadget that has become commonplace for many people, and may provide a more efficient and easier means in creating carbohydrate pictorials than a PC. The touch screen capabilities holds for the effortless manipulation of shapes and building structures, and the device's portability allows users to collaborate. In this chapter, we give a brief overview of the carbohydrate primary structure, followed by an assessment of the Glycano interface.

4.1 Carbohydrate Primary Structure

Carbohydrates are potentially branched molecules built from units known as monosaccharides. The large number of monosaccharides combined with a range of linkage positions results in a huge number of possible carbohydrates that can be built from basic sugar residues. In order to convey linkage information accurately and simply, chemists draw graphs of the carbohydrate structure, with single sugar residues as nodes and chemical bonds as linkages. Nodes are represented by 2D symbols such as the Consortium of Functional Glycomics (CFG) Essentials [64] symbol nomenclature and the Oxford Glycobiology Institute (Oxford) model (Figure 4.1) [65] and many ad hoc symbols created for a single study. A molecular graph of a carbohydrate therefore comprises edges to show shorthand forms comprise a set of logical and distinguishable symbols to represent monosaccharides and linkage positions and orientation [65].

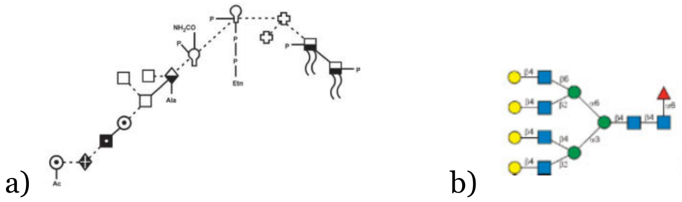


Figure 4.1: Complex carbohydrates are the largest group of biomolecules on earth and structural schematics are represented by the a) Oxford symbol nomenclature [65] and the b) CFG system [64] .

Glycano uses a symbol set created by Kuttel, UCT (unpublished) will be used (Figure 4.2). The design aims to be more easily learnt and visually distinct, which allows for better visualisation and comprehension of the expressed carbohydrate structure. The symbol set uses colours, shapes and patterns for rapid identification of residues and patterns of sugars that make up a more complex polysaccharide (Table 4.1). It also employs additional elements that denote the angle of the bond (α and β bonds), orientation of the ring-shaped monosaccharide (D and L forms), and derivatives or modifications of the sugar molecule (substituents). The ease in learning the content will also contribute in evaluating the learnability of the application interface, as targeted users will include students that are inexperienced with using carbohydrate symbol sets.

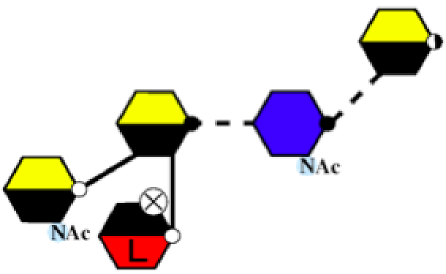


Figure 4.2: An example of a carbohydrate representations in Glycano. Solid black lines joining residues represents α bonds and the dotted lines represent β bonds.

Table 4.1: The symbols used in Glycano, from which combinations can be created to form a carbohydrate structure.

Symbol	Residue
	Altose
	Galactose
	Glucose
	Gulose
	Mannose
	Talose
	Allose
	Idose
	Xylose
	Ribose
	Lyxose
	Arabinose
	Threose
	Erythrose
	Erythulose

Symbol	Ketose
	Ribulose
	Xylulose
	Fructose
	Psicose
	Sorbose
	Tagatose
	Altra-heptulose
	Mannose

Symbol	Substituent
	Nitrogen
	Sulphur
	Phosphate
	Acetyl
	Methyl
	6COOH

4.2 Overview of the Glycano Application

Glycano is a Java based program [24] that is used to produce carbohydrate structures based on the drafted symbol set (Figure 4.3). The software works by connecting nodes with edges to form a network of shapes that corresponds to the carbohydrate structural diagram. It utilises a click and add function, whereby the mouse is used to select a shape and then a single click on the canvas places the shape, which automatically snaps to an adjacent shape creating a bond. More complex functionalities of this program include dynamic naming of the formula of the molecule as residues are added, automatic layout of the residues into a more uniform arrangement and visibility toggling of bond names. Tabs include the D and L forms of the residues and ketoses, a separate field for substituents, and buttons for choosing between alpha and beta bonds. Creating a structure is done in the following steps:

1. Select the required residue by clicking on the tab the residue is categorised.
2. An attached image of that residue on the cursor appears indicating that residue selected can be placed on the canvas.
3. Select either alpha or beta bond to determine the orientation of the new bond that will be created between the residues.
4. Moving the cursor a certain distance on corners of already placed residues will bring up a greyed out line to indicate how the new residue can be attached.
5. Click once to place the residue. A bond is automatically created between the two residues if they are in close proximity.

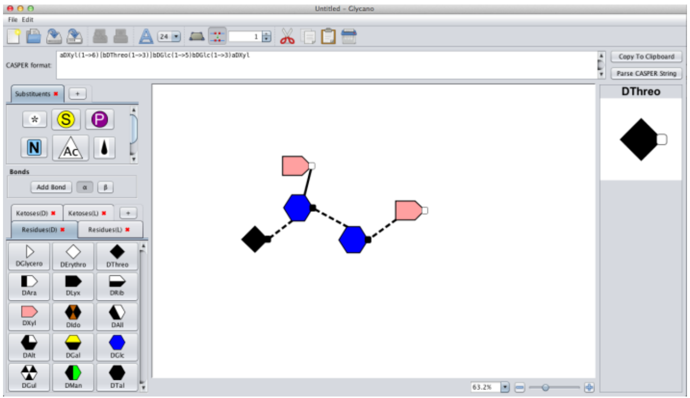


Figure 4.3: The Glycano UI of the Java based software installed on a PC.

This requires numerous steps and possibly more if a mistake or change needs to be made to either the residue or the bond. Using this application on a tablet device would be advantageous, as the addition of touch screen capability would make it easier for users to carry out these actions, making building of structures more efficient. In order to port this program to an Android device, not only would the code need to be adjusted to match Android's application framework, the UI design would need to be redesigned for an Android tablet screen. This is to allow optimal usage of the smaller, to fit the necessary functionalities, as well as a consistent aesthetic to match the platform it is running on.

The goal of the Glycano tablet application is to construct carbohydrate structural schematics efficiently with minimal time needed to learn the navigational UI. Glycano's end-users are those whose main concern will be on the content of the application and rapidly achieving the objective with minimal time in learning the UI or requiring complex functionalities. Developing an appropriate interface is therefore crucial for an intuitive application whereby new and experienced users are able to build a carbohydrate diagram seamlessly upon opening the program. The interface will consist of the main canvas on which the diagrams are created, the design options comprising the carbohydrate symbol set, and the main application functionalities, such as open, delete and save. The aim of the design of the UI is to integrate the main functionalities of Glycano with the usability standards of Android guidelines. For example, adapting the layout and navigational controls to that of the multi-pane layout and various menu patterns such as fixed tabs or spinners, which will account for menu consistency and limited screen space. The various touch gestures to trigger actions must also utillise graphic affordances and user familiarity with the platform to hint at what gestures are required. The mobile application uses a drag-and-drop gesture to move residues onto the canvas as opposed to the click-to-select method used in the desktop program due to different input methods. The lack of a cursor therefore means that the user needs to "hold" on to a shape and by removing the finger from the screen it is dropped into place on the canvas. The way users navigate through the list of residues, such as scrolling horizontally or vertically, is also important because of the ergonomics of holding the device and the frequency this functionality is used. Separation of primary and secondary functions such as editing of elements within the schematic and general application settings, respectively, need to be in an easily accessible area or hidden to not clutter the UI.

In order to achieve the objective of Glycano for the tablet, the placement and choice of appropriate Android patterns to enhance the efficiency of content creation and generate a positive user experience will be investigated. In this study, the usability will be evaluated through the use of Android guidelines and assess its expected familiarity to Android users. This will also gauge whether utilising these guidelines, even when using different interpretations of it, can be used for a novel application and allow for efficient building of carbohydrate structures.

CHAPTER 5

DESIGN OF THE PROTOTYPE INTERFACES FOR GLYCANO

Android UI guidelines help developers create a layout for an application that is consistent with other Android applications and hence familiar to users. They assist graphic-designers in creating an aesthetically pleasing and organised product. However, even though the use of such style guidelines and patterns is encouraged, it may not necessarily lead to an improved usability experience. Selecting the right elements to provide a clear and efficient navigation and discoverability of functions for both novice and expert users is an important design aspect. In this study, to evaluate which UI patterns and elements are optimal for the Glycano application, three alternate mockups were developed. Here we describe our rationale for creation of the prototypes for the Glycano software and applicable Android 4.0 UI guidelines. As this study is concerned with the initial design stages of application development, rapid and low-cost methods of prototyping are ideal. However, as prototypes are evaluated by end-users and experts, the optimal fidelity method for conveying the application concepts is required. We used low-fidelity paper prototypes for initial evaluations with mock users, to gauge whether a low-fidelity mockup was adequate or a higher level of detail was needed for the usability testing of end-users.

5.1 Approach

A range of UI patterns from the available library is used across the three prototypes. The main elements that are modified between designs are the layout of the content, navigational controls, including menu and touch gesture patterns, and item selection. These patterns contribute extensively to the consistency of an application and the identity of the Android OS, as discussed in Chapter 2. Low-fidelity prototypes of four designs were developed for an initial, pilot test on three users. The designs had a “sketched” and low graphic quality to suggest an incomplete and rough design. This was to allow users to focus on the content of the application rather than on the aesthetics of the UI. The designs were developed in adherence with Android guidelines and use various endorsed patterns and layouts that have the same objective. The results from this initial test were used to refine the designs for the high-fidelity prototypes.

5.2 Evaluation

Assessment of these designs was performed through usability testing of end-users and expert heuristic evaluations, to assess which of the patterns and interpretation of the guidelines leads to the most efficient method of accomplishing a user task. The usability testing was performed with end-users, such as chemistry students, to ensure that the design meets with objectives of the application and the content was satisfactory for the target user. Each participant was individually tested using high-fidelity paper prototypes and required to perform a series of tasks that lead them through the main functionalities of the application. They were asked to use the "think-aloud" method throughout the testing, and difficulties in completing tasks are noted to determine where UI problems lie. A questionnaire and interview to obtain their feedback and recommendations regarding the design follows.

The expert heuristics evaluations use usability experts, whose feedback relies on their industry knowledge and experience, to critically analyse the three designs from the perspective of the user. A set of usability heuristics such as learnability, predictability and efficiency, as well as the Android principles, or a list of design guidelines to assist in developing the interface, were rated, to establish the problem areas of the design (Appendix 5.4). Both groups feedback may provide different content into what is considered important for usability and which design patterns helps in providing an efficient method for constructing carbohydrate schematics. Lastly, the UIM of either using end-users or usability experts were evaluated to assess which would be best for the freelance or lone developers with limited resources. Refined designs from both groups were then developed from obtained feedback and recommendations to view the visual improvements that have been suggested.

5.3 Low-Fidelity Prototype Pilot

As few studies employ paper prototypes with Android touch screen tablet devices, an initial pilot usability test was performed to confirm fidelity for this case-study application as well as to evaluate and refine the testing approach. This pilot assisted in preparing for the usability test with paid end-users and ensures it was executed properly regarding interview and biographical questions. The first mock usability test involved testing four low-fidelity prototype designs to determine which Android elements improve learnability and efficient navigation of the application (Figure 5.1). Here, low-fidelity refers to the low quality of the visuals and the amount of detail placed on the graphics. Mockups were created in Microsoft Office PowerPoint. All interfaces were developed using distinct Android guideline components and stylistic decisions were based on current and frequently used elements and interactive Android patterns. The prototypes contained the same functionalities, but only frequently used and core application features were incorporated in the user tasks. As the discoverability of an application's options and features rely on on-screen cues or affordances, various elements can be tested when testing multiple designs. Progression through the application conforms to general navigation principles, however, efficiency when learning the application's interface was taken into account in this first round of usability testing. For example, familiarity and competence with a system's navigation controls becomes easier over time, regardless of any initial usability issues. The comparative testing is therefore focussed on determining which UI elements are most intuitive in early stages of the application's usage.

The low-fidelity designs were minimal to prevent users becoming distracted from irrelevant information, such as colour and font styles. The outline of the tablet and UI graphics were represented with a “sketched” quality, in order to denote an unfinished product and encourage users to provide constructive feedback. All designs included the action bar at the top of the application to hold the options for global user actions, such as save and delete. This is a common feature that was introduced with the release of Android 3.0 (API 11) and acts as a structural anchor for various interactive components and also provides a consistent and recognizable interface feature across applications.

The prototypes were designed with various Android patterns and layouts. Design 1 (Figure 5.1a) uses two vertical sidebars to display the main menu and their respective options. The left-most fragment is fixed, with the adjacent bar displaying the scrollable list of the selected category. This is a common layout seen in tablet applications and one experienced Android users should be familiar with. Design 2 (Figure 5.1b) incorporates the main categories into the top action bar and a horizontal scroll list beneath. This leaves most of the screen free for the building of the carbohydrate structure. It is to assess the preference to the horizontal scrolling gesture. Design 3 (Figure 5.1c) separates the two fragments to place the canvas in the centre of the screen. This allows for users to select categories with their left hand and drag residues with the right hand. Text is also used in place of the icons for users who might not understand the icon representations. Design 4 (Figure 5.1d) uses a tree structure for an expandable and collapsible navigation of the left hand menu. A long scrollable list is formed and a larger canvas is available. Users can control which category they would prefer hidden or remain visible.

5.3.1 Methodology

The testing was conducted on three users. Each screen was printed on a piece of A4 paper and placed in order of the steps required to complete a task successfully (Appendix 3.2). The participants used their finger on the mockups to simulate the gestures they would do on a tablet. As the user moved through each task and completes a step, the next paper was flipped over to reveal the effects of the correct step. Any difficulties or wrong moves would be guided or questioned such as why such a manoeuvre was made, what they thought would happen and users were asked what else could they could observe on the screen that would meet the objective. No recording equipment was used. Information for refining of the approach was gathered such as possible questions users would ask and the difficulties seen with the low-fidelity mockups.

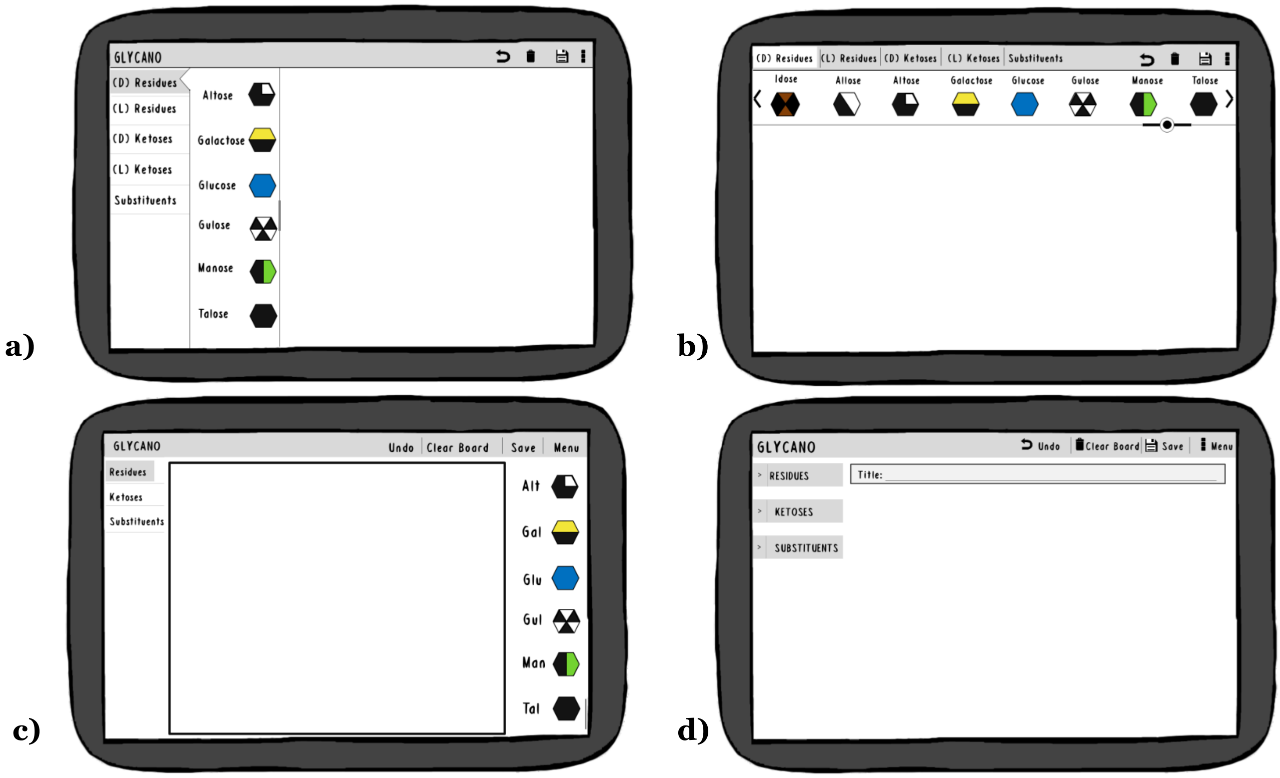


Figure 5.1: The low-fidelity mockup designs used for the pilot usability testing. The designs had various layouts including a) vertical sidebars on the left, b) a horizontal navigation with fixed tabs, c) a split layout seen in few sketching applications, d) a collapsible tree-structure navigation.

5.3.2 Results

With regards to the prototypes, a few minor image inconsistencies were identified in some of the paper prototype screens. The scripting of questions and guidance for the test was refined where unforeseen user difficulties were encountered. For example, it was discovered that users had the most difficulty in navigating the first design prototype due to a misunderstanding of tasks given. This may cause a bias, in results due to a negative or emotional response when first attempting to understand tasks and unfamiliar terminologies. As the test users were not the intended target end-users of Glycano, the wording of tasks was simplified and the exact instructions on what to do for each step for the rest of this pilot test was changed. This allowed users to concentrate on the design of each prototype and overcome the difficulty of touch screen based gestures that are not naturally be intuitive with low-fidelity paper designs.

In this pilot study, users lacked the perception of touch-based gestures when interacting with the paper prototypes. Even though wireframes were effective in constructing various initial designs and viewing conceptual layouts for navigation, a more representative model that is close to the intended solutions is required for usability testing with end-users. Communicating touch-based elements through static low-fidelity wireframes is difficult where device feedback and affordances play a significant role. For example, some interactive gestures that are natural when handling a touch screen device, such as long-pressing and reacting to haptic feedback, are not intuitive when engaging with paper mockups. Although most of the tasks involved in the mockups require only single tapping of controls, some involve gestures that may be more apparent when dealing with a higher fidelity prototype or the fully developed application on an actual device. As many users have become accustomed to mobile interactivity and its controls, downgrading the experience was shown to affect the usability and comprehension of information. For example, the abstract and unclear representations of buttons and interactive elements made completing tasks fairly difficult. Furthermore, many contextual affordances that mobile devices rely on to suggest interactivity, require a higher level of detail than is provided in a non-functioning prototype. The inflexibility of the low-detailed designs meant certain features were indistinguishable between static elements and buttons as a higher level of detail such as shading, colour and texture would suggest interactivity. The testing of four designs also became too lengthy.

Therefore we concluded that while the pilot testing was beneficial in improving the scripting and interview questions as identifying inconsistent design details that may be overlooked by the designer, it can cause confusion. Even though low-fidelity prototypes are easy to produce and are a useful method for generating many design ideas, the contextual usage of the actual device and its interaction is not conveyed and may produce misleading results [66]. This is possibly due to an ineffectual means of conveying interactive elements that did not adequately simulate a tablet application environment.

5.4 High-Fidelity Prototype Design and User Tasks

For the high-fidelity paper prototypes the number of mockups was reduced to three and the last design amended to a more intuitive and user-friendly design. Design elements were more realistic and graphically detailed, such as shadowing and colours of the Holo theme. This provided an authentic representation of the final product and better illustrated the on-screen cues needed to depict interactive fields. As Android guidelines aim to evoke a positive response from users, a more polished application is more appropriate for end-user testing. Although the customization of aesthetics can be made on top of foundational patterns, usability testing of the underlying navigational controls was the focus of this study. Therefore, the same theme (including colour and typography) was used in all designs to enhance or indicate interactive or selected fields, as recommended in the OS guidelines. Although high-fidelity mockups may be criticised for being time-consuming, there are currently numerous digital resources and templates available to reduce the time required to develop high-fidelity mockups. For example, downloading freely available Photoshop UI templates. This study focused on pre-designed and established Android patterns and styles, which avoids incorporating custom designs to reduce user feedback focus on irrelevant information.

In these designs, the functional requirements of the Glycano UI were adapted to the Android guidelines to assess the usability of the OS's endorsed patterns and principles (Appendix 5.3). Android design guidelines are incorporated into the mockups through the combination of various Android patterns (Table 5.1) [9]. The designs were clear with minimal clutter but also take advantage of the large screen space. The standard Holo Light theme was used for all three mockups, with similar styling of the font, colours and general sizing of text and controls. Icons were used more than text to reduce the cognitive load and to grasp the user's attention. The drag-and-drop feature for manipulating residues contributed to the "enchant me" principle in allowing direct manipulation of objects. Options were simplified with hidden secondary functions and obvious navigational transitions, such as with the highlighted tabs and spinner pattern, incorporated. In addressing the "make me amazing" principles, shortcuts in the contextual menu allow for multiple changes simultaneously and important functions are visibly placed on the main screen for fast access.

Different navigational design patterns were used to determine which is best for effective carbohydrate building and to allow the user to access and create content quickly (Appendix 1). The three mockups are based around a horizontal (H) (Figure 5.2a) and a vertical (V) (Figure 5.2b) navigation system and a combination (C) of both (Figure 5.2c) and were designed in the graphics editor, Adobe Photoshop CS5, due to its powerful functionalities [67].

Table 5.1: Android design patterns used in each design.

Design pattern	Design V	Design H	Design C
User action	• Action bar • Overflow menu • Custom overlay • Vertical scrolling	• Action bar • Overflow menu • Contextual menu • Horizontal scrolling	• Action bar • Contextual menu • Horizontal scrolling
Navigation	• Multi-pane layout	• Swiped tabs	• Spinner • Swiped tabs

The V prototype utilises a multi-pane layout, common in many tablet applications, that allows vertical scrolling of the residues. The space in the action bar allows for primary functionality icons to be visible and easily accessible. All secondary features are hidden with the overflow menu. To indicate the scrollable area and that there are more items within the list, a vertical scrollbar is present. This feature, however, is usually hidden and only becomes visible when the user interacts with the menu.

The H prototype consists of swiped tabs across the top action bar and an overflow menu that holds most of the functionalities. The list of residues is placed in a parallel bar to the action bar, making it a horizontally scrollable list. To indicate that the scrollable area holds more items, left and right carats are placed on either side, which can also be tapped. This leaves the majority of the screen to the blank workable canvas.

The C prototype uses both the action bar to hold the categories and the vertical left panel to hold the residues. Menu items are held in a spinner or drop down menu and all features are visible in the action bar. The residues are placed side by side to make searching for residues faster and less scrolling is required. The toggle between D and L form also reduces the number of items held in the main menu for faster searching. The canvas is also larger than in the V prototype.

The elements were chosen to support user action and navigational patterns, such as scrolling, menu and icon accessibility, organisation of the controls and arrangement of space on the screen, as well as touch gestures (Appendix 1). The comprehension of these standard UI designs in aiding the learnability of the Glycano application and their efficacy in enabling the user to achieve the given tasks was also assessed.

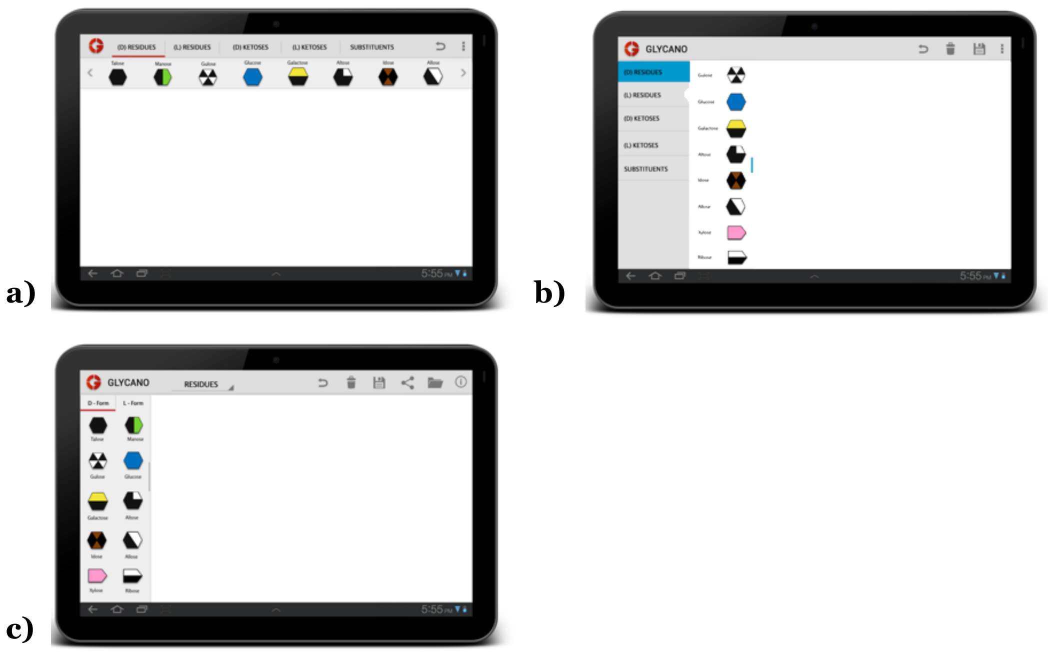


Figure 5.2: The three mockups used to represent the a) horizontal navigation, b) vertical navigation and c) a combination of vertical and horizontal navigation.

CHAPTER 6

EVALUATION

Glycano's target users are students and chemists whose main concern will be on the content of the application and rapidly achieving an objective. Developing an appropriate interface is therefore crucial for an intuitive application whereby new and experienced users are able to build a carbohydrate diagram seamlessly upon opening the program. In this study, the UIMs of user and expert testing was evaluated through use of Android guidelines integrated with a new tablet application. This also gauged whether utillising these guidelines, even when using different interpretations of it, can be used for efficient building of the carbohydrate schematics. The following chapter reviews the two usability testing methodologies and their evaluations, from which the feedback of the two subject groups are later compared.

6.1 Usability Testing Methodology

High-fidelity paper prototypes of three different designs representing various interfaces and navigations were assessed. Little emphasis is placed on replicating realistic tactile manipulation or contextual usage for the targeted touch screen device as the focus of this evaluation is on gauging user understanding of specific navigational layouts, terminology and iconography in order to provide more efficient workflow. Furthermore, this testing phase allows for horizontal prototyping of the application, where user interactions of the main functionalities across the entire system can be analysed.

The tasks were made to enable the user to observe the main functionalities and all the navigational screens of the application. The initial task involved searching for and moving a residue onto the screen. This required the user to scroll horizontally or vertically to search and then drag-and-drop the residue onto the blank canvas. The second task, again involved searching for another residue which was in a different category or form. Users were required to switch tabs or pick an option from the spinner in order to find the residue within that category. To adjoin the residue onto the previously placed shape, the user had to drop it adjacent to the residue. The third task involved editing the carbohydrate, which involved changing the bond created between the two shapes to a beta bond (indicated by a dashed line). Users had to long-press the bond to reveal a contextual menu or pop-up menu from which the option to change to a beta bond can be applied. The final task was to save the schematic to view if users understood the icons, including the overflow menu icon where the save option is located in the H prototype. These tasks represent the frequently used and main functionalities that would be required when constructing the carbohydrate schematic.

Mockups were printed on A4 paper and each screen was presented one page at a time (Appendix 4). A biographical questionnaire, a printed sheet with four tasks and post-test questionnaire were presented to the user who was required to read aloud and perform each task on the paper prototypes (Appendix 3). Thinking aloud is a method allows the evaluator to gain an insight into the thought processes as users interact with the system, along with post-task interview and questionnaires are common techniques in gathering usability problems [20, 41, 67]. These tasks represent the main functions and goals of the application and are to be repeated for each of the three alternate designs. Gestures and tapping are be performed using a pen or finger and behaviour and feedback on completing tasks is to be assessed. The prototypes are a similar size to the actual device screen and its components. Three designs are provided to evaluate user interaction with various layout possibilities and Android elements that will be considered for the final design of the UI for efficient carbohydrate structure building. A final design was developed to view what end-users expect as a more usable design in efficiently building carbohydrate schematics.

6.1.1 Test Subjects

Chemistry undergraduate students are recruited to participate in this usability test. A short biographical questionnaire was sent out prior to testing, to ensure that only chemistry students qualify for the study. Android and mobile experience is also obtained. Students are provided a monetary incentive and consent is required, as video and audio recording of the testing for note-taking purposes is used. Participants were seated at a table and presented with the paper mockups in a binder. The experimenter takes the role of the test facilitator in taking notes of user interactions, turning over pages and asking questions as a form of guidance. Users were required to navigate the application in the same context as an Android tablet application.

At the end of each mockup testing, a Likert scale questionnaire was given in which the ease of use, learnability and familiarity of controls has to be assessed on a given scale. The order of the mockups was given at random to each participant to prevent bias.

Comparisons of the layouts, presentation and preference to prototypes were questioned during the post-task interview.

6.1.2 Usability Evaluation and Metrics

Evaluations of usability testing methodologies have shown to provide varied reviews on its efficacy in extracting valuable user feedback. Such results can be attributed to the highly contextualised dependencies of each study as well as the subjective nature of UI design in itself. In this study, the scope and possible influences of usability testing was narrowed to allow for applicable methodologies to be adapted in the assessment of Android principles and patterns in user experience. Performance was measured by task completion rates for each design [49]. The same tasks are completed on each design and any difficulties observed noted. Guidance was provided should the user take too long to respond or show signs of confusion. Users are encouraged to find the answers by themselves and the time taken to complete a task signifies the level of learnability, efficiency and memorability [20].

Following the completion of the task list for each prototype, a brief questionnaire with Likert scale questions was completed. A more in-depth interview is carried out at the end of testing and the feedback from the three experts are compared. Feedback on usability issues and comparisons of the navigation and layouts between the designs are discussed. A simultaneous comparative analysis of three designs is utilised to not only gauge the effectiveness of Android controls of similar functionalities and explore alternative layouts, but to generate critical user feedback [68]. The content and functionalities are kept constant between the prototypes to assess the advantages and disadvantages of navigational controls and patterns and allow the user the freedom to be critical, as they do not have to commit to a particular design and rather forced to consider why one design is better than another [69]. This prevents users from thinking that it is a final product, seen with single mockups and reducing critical feedback, as the flexibility of navigational controls can be compared between mockups.

6.1.3 Design Refinements

The preferences and opinions that are expressed predominantly by the end-users were used to refine the three mockups to a single design. These include the overall layout, and various elements. Amendments to areas in which users had difficulty are made and additional suggestions and recommendations provided may be added.

6.2 Expert Heuristic Evaluations

For this study, three usability experts with Android development experience are recruited to provide an expert review on the prototypes. This is to ensure that mockups meet Android guideline standards, as well as to gain a holistic view of improving the usability from industry knowledge and experience.

6.2.1 Participants and Procedure

Participants who identify themselves as usability experts working within the user experience or mobile development field will be asked to take part in this study gratis for approximately an hour to complete. To overcome scheduling and location issues, the prototypes and survey were provided over email. This allows sufficient time for participants to thoroughly scrutinise all three designs and work through the survey at their own convenience and pace. A biographical questionnaire and a primer are included (Appendix 5). The primer serves to introduce the Glycano application, its objectives and target users. A flowchart of the main screens used in the usability testing was provided alongside the list of user tasks (Appendix 6). The steps required to achieve each task was explicitly described and illustrated on the prototype flowchart to gain an understanding of what is required of the user (Figure 6.1).

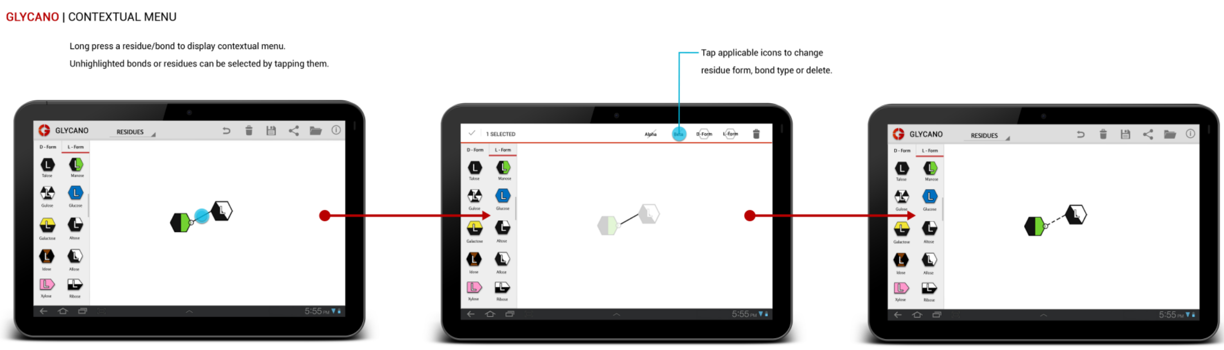


Figure 6.1: An example of a flowchart for a particular mockup design provided for expert evaluation. The flow looks at a task where the user needs to edit the bond. Each step required to achieve the task is provided with its resultant screens.

6.2.2 Expert Evaluation and Metrics

Experts were required to review all three prototype flowcharts and answer a heuristic evaluation for each. A combination of both the cognitive walkthrough method and heuristic evaluations are used. This is to evaluate the UI against a set of established heuristics to view main design problem areas, as well as to gain a general sense of users' thought processes with initial interactions of applications. Finally, in order to improve on the designs, priority can be focused on the main usability flaws, assessed in the heuristic evaluation, and refined with the recommendations provided in the cognitive walkthroughs.

The list of Android design guidelines and the most relevant usability heuristics, adapted from Nielsen's heuristics [16], were used to aid in the evaluation of the designs in determining which usability problem areas were most evident (Appendix 5.3). These heuristics were considered to be most important for the aim of the designs in the context of the study and take into account cognition and performance support. Each heuristic was provided with a brief explanation and needed to be given a severity rating based on the following scale:

- 0 = I don't agree that this is a usability problem at all
- 1 = Cosmetic problem only: need not be fixed unless extra time is available on project
- 2 = Minor usability problem: fixing this should be given low priority
- 3 = Major usability problem: important to fix, so should be given high priority
- 4 = Usability catastrophe: imperative to fix this before product can be released

Space for comments and recommendations as well as an overall questionnaire that compares the designs and preferences concluded the evaluation (Appendix 5.4).

In order to compare expert and end-user feedback, evaluation of the static and dynamic controls and layouts encountered with each mockup were grouped into five categories adapted from Park et al. [70]. These were used to aid in the evaluation of the Android components and refinement of the design (Table 6.1).

6.2.3 Design Refinements

The severity ratings for each heuristic were averaged to determine where experts found usability problems. The heuristics that scored the highest ratings were concentrated on in refining the design. Preferences and recommendations provided throughout the evaluation and final questionnaire were accounted for in developing the final design. A side by side comparison of the end-user and expert final designs were then discussed.

Table 6.1: The Android components and patterns utilised in the application and its groupings for expert review feedback analysis.

Static			
Group	*Guideline Property*	*Android Component & Pattern*	
Layout	Spacing of icons	Action bar	
	Grouping of icons	Overflow menu	
	Grouping of list items	Multi-pane layout	
	Ordering of list items		
Terminology	Abbreviations / labels on customised icons		
	Naming of secondary functions		

Dynamic			
Group	*Guideline Property*	*Android Component & Pattern*	*Gesture*
User input	Selection of menu items	Fixed tabs	Touch
		Spinners	Touch
User input	Selection of menu items	Lists	Touch
		Popup notification	Long-press
	Scrolling	Scrolling with scroll indicator	Vertical / horizontal swipe
	Moving of residues		Drag and drop

CHAPTER 7

USABILITY TESTS

In the first part of this study, usability tests are conducted to observe end-user behaviour and gain feedback on the three Glycano mockups. The aim was to evaluate the efficiency with which they learnt the UI and create a diagram. The intuitiveness of each design and ease of which targeted users can navigate the application can therefore be used to assist in the refinement of a final design. Users completed four tasks that represented the most important system functions used to build the carbohydrate structure diagrams in the Glycano application. In this chapter, we analyse data from the questionnaires to identify the Android elements that contribute to the ease of use in task completion within Glycano.

7.1 Users and Environment

Targeted users were recruited to participate in the usability testing. Criteria included: graduated from or currently studying chemistry courses. Seven users participated in the usability test. Ethical clearance was provided and participants were recruited through online advertisements on the University's student portal with a monetary incentive. Three were undergraduate chemistry students at the University of Cape Town and four were chemistry graduates. Five users were familiar with the Android operating system and two participants had no experience in using any Android devices. Testing took place in a closed office and was recorded using a web camera and laptop for later analysis (Figure 7.1).

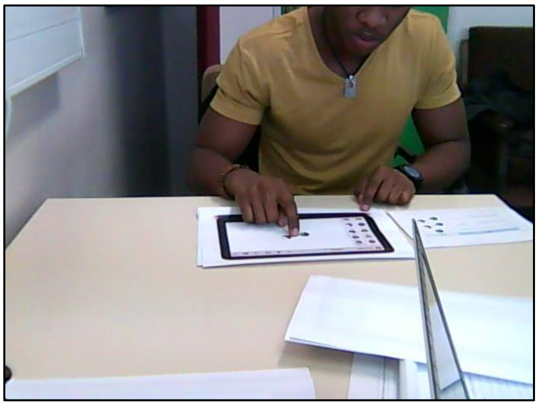


Figure 7.1: Usability testing with end-users who were required to complete tasks on paper mockups.

Qualitative and quantitative approaches were employed to measure user's familiarity, preference and usability of the mockups. These included a biographical questionnaire, a Likert scale survey, to rate user experience for each design, and a final interview (See Appendix 3). The testing took approximately half an hour to 40 minutes for each participant.

7.2 Results

Overall, users found all 3 designs to be either fairly simple to use or quick to learn (Table 7.1). Where users did not find that the UIs of design V and C to be unnecessarily complex, prototype H was only rated as neutral indicating that improvements could be made in making it more user-friendly. Users were able to recognise most of the Android control patterns in all three designs indicating user familiarity with the OS design. Prototype V was observed to be the most intuitive and fastest to learn, as users strongly disagreed on needing a considerable amount of time to get used to this system.

Table 7.1: Averaged results for matrix questionnaire.

	I found the system unnecessarily complex	I thought the system was easy to use	I recognise all of the Android controls used	I would imagine that most people would learn to use this system very quickly	I would need a lot of time to get used to working with this system
Design 1 (H)	Neutral	Agree	Agree	Agree	Disagree
Design 2 (V)	Disagree	Agree	Agree	Agree	Strongly disagree
Design 3 (C)	Disagree	Agree	Agree	Agree	Disagree

Even though users felt that the designs may be easy to use or become familiar with, the preferences with respect to navigational layout and controls differed. The prototype (H) with a horizontal navigation was deemed to be less user-friendly than the other two designs, owing to the awkward use of horizontal scrolling. In contrast, the prototype with vertical navigation was found to be a more intuitive due to user familiarity with this layout. This has also been observed in a previous study, which showed that there was a greater loss of context information when scrolling horizontally and can become cluttered, therefore was worse off than vertical scrolling [60]. Furthermore, contrary to what some studies have shown [38], users did not comment on styling or find aesthetics distracting when performing the tasks. The default theme, fonts and overall colours of the application were not critiqued and recommendations were not provided on how to improve the graphic design. We now review the users' responses to static and dynamic elements of the application.

7.2.1 Menu and Layout

Glycano's user input consists primarily of menu and item selections from the top action bar and contextual menu. Four of the users found the vertical prototype (V) with the multi-pane layout and vertical scrolling to be the easiest to understand and use. This mockup clears the top action bar of any menu options relating to the actual building of the carbohydrates. This is a common arrangement seen in tablet applications, such as in email clients, which allows for the larger tablet screen to be used more effectively. One user noted that it was more intuitive to view order from left to right and therefore identified this layout as the fastest with regards to learning and comprehension of its menu structure. Only one user found this layout difficult to use, but prototype V was the first prototype encountered for this participant's testing. However, cluttering of the menu lists on the left side of the screen does reduce the screen estate for the canvas in prototype V.

Users found the horizontal prototype's (H) to be the second easiest to use. Fixed tabs are a common feature in Android applications, although a user said that prototype H felt more like a Windows environment than a tablet application. The spinner, or dropdown menu, in prototype C was found to be the most complicated for discoverability or in discerning various elements. In this prototype, spinners are used to select an option from a list, with the currently selected option visible within its frame. It is identified via a small triangle on the corner of a text box to indicate a dropdown menu. However, users spent an average of 20 seconds more searching for the residues option located here compared to the other designs. Furthermore, the L and D form toggle button was a positive feature in this mockup.

7.2.2 Scrolling

Users found vertical scrolling to be more intuitive and natural than horizontal scrolling, which some felt to be too cumbersome for searching through long lists; common problems with touch-based devices [60]. Five out of seven users preferred the vertical direction when scrolling. The arrow buttons on either side of the horizontal list, however, did alleviate this issue, as tapping was the preferred gesture when searching through long lists. Furthermore, it was discovered that users required on screen hints, such as arrow indicators to designate scrollable areas, as the scrollable indicator or scrollbar was either overlooked, mistaken for another control or users thought the list was fixed. Android applications generally utilise a hidden scrollbar that is only made visible upon the initiation of the drag gesture to start the scroll. In the mockups, however, an extra affordance such as a semi-visible menu option at the bottom of the list area should have been used to provide further indication of a scrollable region.

7.2.3 Selection of bonds

The main difference between a contextual menu and a popup menu is the option to change multiple options simultaneously in the contextual menu. Although the popup notification was the easiest to identify with regards to the placement on the screen, and more efficient when changing single bonds or residues, four users recognised the need for the more functional contextual menu where multiple bonds and residues must be altered.

7.2.4 Icons

The placement and use of the icons within the action bar and contextual menu provide functionality at different structural levels of the application. Where commonly used functions of the application (such as save, delete, and undo) would usually be always visible, Android guidelines move secondary and less-used functions (such as sharing, gallery and about options) to the overflow menu. Some users, however, did not know the icon for this and would avoid tapping it during the tasks. Customised icons in the contextual menu, such as converting bonds to alpha or beta options, represented by a solid or dashed line respectively, were also confusing without a text label. Users had problems identifying the overflow menu or its function. Although most users either knew or guessed this option correctly, they favoured having all the icons displayed on the action bar. This is a possible solution, as there are few icons in Glycan and use of a secondary overflow menu can be avoided. This may, however, become impractical as more features are added to Glycano at a later stage. One of the major problem areas was the efficiency in searching for the required residue, which impacts on the time needed to build the carbohydrate. This was due to the time taken to identify the residue. An easier way to visually locate the object is to group the shapes according to the number of carbon atoms in each sugar residue. This was suggested by four of the participants.

7.2.4 Summary

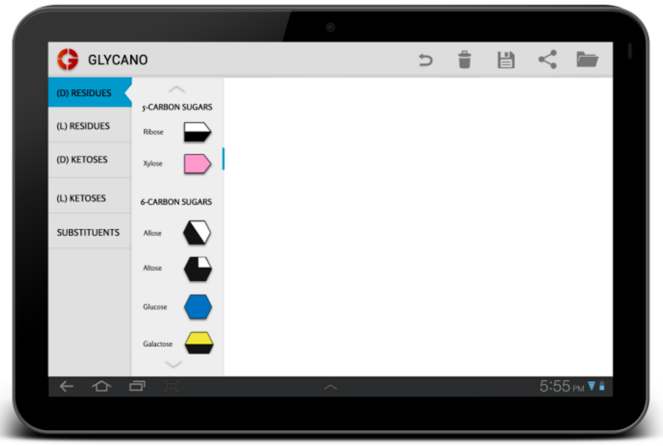
Overall, users preferred the familiar and uncluttered layout of Design V, due to its multi-pane layout that separated the residue categories and its distinctive lists. The vertical scrolling made it easier to search for a residue in a long list and the action bar was left to only hold only the application functionalities. In editing the carbohydrate, users liked the contextual menu as it allowed for multiple selection and manipulation of bonds and residues. Users also favoured the UI where functionalities were visible, such as having icons clearly displayed. They found this more efficient than having to search for options within another menu. However, this could also be due to being able to discover and learn what functionalities are available in this new application.

7.3 Refined Design: Users

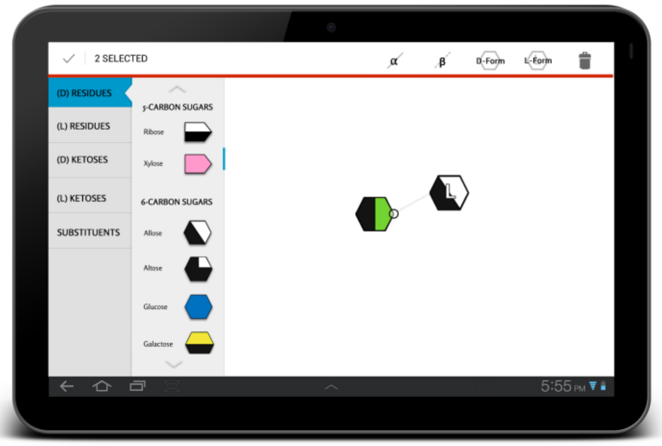
The refined design combined the end-user preferences with regards to layout and navigational controls as well as corrections for the significant usability problems encountered (Table 7.2). The participants preferred the vertical multi-layout due to its similarity to other Android applications (Figure 7.2). Even though the mockups were randomised, this layout had the most positive feedback due to the logical arrangement of the content, which made it more intuitive. All icons are displayed in the action bar, as requested by users. Residue icons were grouped according to number of atoms, for easier searching. As the detection of the scrollable area was difficult for some users, addition of the up and down arrows was included in the refined design to give a visual indication of a list of items. Part of the bottom residue is also hidden to indicate that one would need to scroll to see the rest. This is an affordance that many applications with scrollable areas may use, as the scrollbar is usually hidden. A contextual menu is used instead of a popup menu. As users were unsure of which icons changed the bond, where icons were displayed without a label, the refined design used abbreviated versions of labels on top of the icon.

Table 7.2: Example instances of usability issues encountered.

Usability Issues	Example instances
Interface elements	• It was unclear how to change between menu categories (prototype C) • Users preferred toggling between L and D form (prototype C) • Some users confused the device's status bar as part of the application (prototypes H+V+C)
List browsing	• Most users did not know to scroll to view more residue options (prototype V) • Some users had difficulty locating the correct residue due to unclear grouping of items e.g. alphabetically or by number of carbons (prototypes H+V)
Icon representation	• Some users do not know the Android overflow menu symbol (prototypes V+H) • The scrollbar indicator was overlooked by most users or mistaken for a handlebar by one user (prototype V) • Some users stated that they would not have known which icon represented the beta bond if it was not written in the task list
Appearance	• Too cluttered (prototype V)
Comparison with other Android applications	• Two users mentioned the similarities with other applications (prototype V)



a)



b)

Figure 7.2: The refined design for the Glycano application utilising end-user feedback and recommendations showing a) the main screen upon opening the application and the b) contextual screen.

7.4 Discussion

In most cases, users stated that their main method for of discovering and searching for functionality within touch-based applications is by tapping on random UI features. When users were unsure of where to navigate to complete tasks, certain interactive elements were easily dismissed or ignored, including the spinner and overflow menu icon. This confusion and guessing of various UI elements should be reduced to enhance the learnability and discoverability of an interface. Reduction of clutter, common OS patterns, organization of list options into logical groupings and generating understandable and distinct icons were therefore the main refinements required to increase efficiency of learning the applications UI and accelerate building of carbohydrates. Furthermore, objects such as the residue options, should provide a sense of being able to be directly manipulated. The refined design therefore encompasses a combination of the most favourable elements from each design.

Usability testing can contain contextualized methodologies, which depends on the aim of the study and what data is being investigated. The use of cheap paper-based prototypes as opposed to the physical experience of handling the actual touch-based tablet device should also not be overlooked. User attitudes towards interaction with non-interactive buttons may affect the way users perceive usability. Due to the lack of feedback from the application, an aspect that users rely on, concern over whether it is a system or human error can cause user apprehension when providing critical feedback on the UI. Furthermore, the use of an incomplete and small end-user representation and testing within in an artificial situation does not necessarily prove the product works or is useful [69]. Given that time and resource constraints usually prevent extensive and full-scale usability evaluations, it is difficult to fully assess the functionalities of a finished product [71]. The usability of more complex tasks and user comprehension of the overall system cannot be easily determined in traditional testing scenarios therefore early testing may not give an accurate view of the final usability. However, in this usability test a representation of the main objectives and most frequently used aspects of the application are assessed to aid in early refinement.

Limited testing may not provide absolute verification of the final product's usability but it is still a valuable approach to identify possible usability issues that can be improved upon before release of the software. Tohidi, et al. did a similar alternative design testing, and was observed that more usability problems were uncovered from the feedback than recommendations [68]. Generally, the public can be forgiving of a poor or customised UI if they find the purpose of the application valuable or if there is no alternative, in which case extra effort is placed in learning the controls and functionalities. Additionally, the development of mobile applications also includes the continual improvement and updating subsequent to its publication on the OS marketplace. Feedback and reviews from actual users are real time criticisms and suggestions that can be used for analysis for ongoing amendments to the application.

Although the most essential UI components and concepts (navigation, selections and application structure) may not necessarily dictate the best UI design or aesthetic, the proper use of these features have been observed to dominate the user experience outcome. Android guidelines and patterns contributed to the usability of the application where users familiar with the OS where able to navigate fairly quickly. However, after the first round of prototype testing, even inexperienced users became faster in recognising various elements and recalling the navigational menus and touch gestures required to complete a task. Following the guidelines and using the library of patterns is a good foundation for the UI design of an application, as it leverages off the OS platform design that users are familiar with. Even though principles may be abstract or interpreted differently, performing usability tests to confirm that the selected patterns and layouts is the best for the application's objectives is worthwhile.

This study looks at the initial large modifications that can improve upon and encourage an application's uptake in the marketplace. Furthermore, as many applications may only have been developed and maintained by one developer, extensive usability testing may not always be possible. We found that the use of high-fidelity paper prototypes during the initial mobile design stages, however, is a good cost-effective and uncomplicated option to determine major initial usability flaws in the UI.

CHAPTER 8

EXPERT HEURSITIC EVALUATIONS

Usability experts were used to evaluate the prototype designs in order to confirm adherence of the designs to the Android guidelines, as well as to gather industry expert opinion on the usability of the application. Their analysis was guided by a set of mobile usability heuristics, adapted for the context of this study, as well as the Android design principles. The experts provided recommendations based on their experiences, usability knowledge and their perception of how general users would navigate through and understand the UI controls and design. The expert reviews contrasted with the user testing to assess which UIM is more beneficial for developers in initial stages. Their feedback and recommendations contributed to the refined design.

8.1 Methods

A total of three participants, who identified themselves as either usability experts or Android developers with usability experience, were approached to analyse the same mockups presented to end-users. They volunteered to take part in the study gratis. Two of the experts have been working within the usability or mobile development field for over 5 years. The third expert is an experienced graphics designer. All three rated themselves as either "very good" or "expert" regarding their familiarity with Android 4.0 guidelines and patterns, usability heuristics and human-computer interactions.

The assessments of the mockups adopted a hybrid approach of a cognitive walkthrough and heuristic review. As participants had no experience with the application area, a primer was provided to explain the aim, target users, and functionalities of the application. High-fidelity mockups were provided along with a usability survey that served to aid experts in their analysis. A spreadsheet listing the usability and Android heuristics was provided and a score for each heuristic was required. The quantitative measurements were averaged to pinpoint the greatest usability problems that were encountered for each design and the severity rating of these were graphed (Figure 8.1). The testing was done according to the methodology discussed in Chapter 5.4. Qualitative analysis included a survey and questionnaire that was provided as part of the spreadsheet. Participants were required to complete this assessment for each of the three mockups.

8.2 Results

All reviewers agreed that the designs followed the endorsed Android guidelines and no major usability problems were identified (Appendix 2). Due to this, the learnability of all the designs and therefore the amount of time required to become familiar with the navigation was not excessive. The only Android principle that had an average severity rating of 2, meaning that fixing such issues can be given low priority, was in the amount of application feedback in Design C. This design relies on user familiarity and a level of intuitiveness, however, this could lead to problems with new and returning users who have not used the application after a period of time. Experts felt that a more explicit set of instructions or visual guidance was required, as end-users would otherwise not receive the "encouragement", as regarded in the Android principle [9]. Furthermore, they observed that various Android elements could be interpreted differently. For example, the fixed tabs on the horizontal navigation suggested that the entire screen would change instead of just the list menu. The position of the spinner in Design C was also brought up several times, as it was placed too far away from its child element. It could therefore be seen as being disconnected from the actual list it is associated with. The reviewers' outlook on the efficiency was consistent with the high efficiency ratings scored in the heuristic evaluations and was reinforced in their comments.

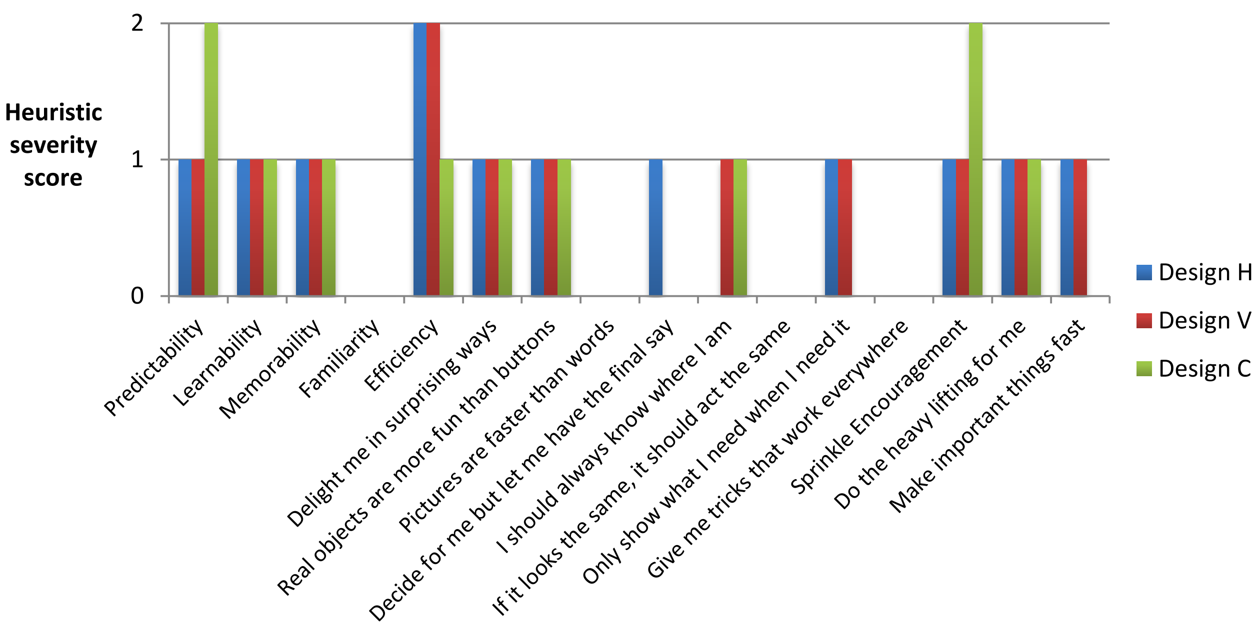


Figure 8.1: Graph showing averaged heuristic severity ratings for each design. Efficiency was a larger problem in Design H and V, whereas predictability and encouragement were issues for Design C.

Minor usability problems were mainly observed with the efficiency of Designs H and V and with the unpredictability of patterns in Design C, such as the misplacement of the spinner. The main concerns raised were the lack of feedback, navigational confusion and most significantly, the inefficiency in building the structure. Factors that contributed to the latter included overestimating the proficiency of users when initially navigating the system; cluttering of items, such as exposing high-level and secondary functions on the action bar; and an unclear focus on selected objects (Figure 8.2). As the majority of the main screen is a blank screen, users may be unsure of where to start or how to place the residues onto the canvas. This was confirmed in the usability testing, when the drag-and-drop touch gesture was not considered when asked to place a residue onto the canvas. The experts suggested that a tutorial or walkthrough be included on first start-up of the application, explain gestures and functionalities. Experts also preferred an uncluttered screen, meaning that unused icons should be kept hidden. In Design C, the visibility of all the icons was seen as too many options on the screen, which could hinder user experience and go against the Android principle of only showing what is needed. The distinction between selected and unselected objects was also seen to be vague. The lack of feedback or visual clarity from selected objects was seen as an issue as it could lead the user into making further editing mistakes.

a) **b)**

Figure 8.2: Prototype screens screen showing a) the layout of all icons representing both primary and secondary functions that are visible on the action bar and the b) unclear selection of the bond.

Reviewers stated that the layouts were more a preference than a usability concern. However, navigational elements that added to usability problems included the spinner menu and the icon representations. A lack of consistency was noticed between the icons on the action bar of the main menu compared with the text within the contextual menu. User opinions of the usability testing seemed to correlate with Android usability experts regarding the confusion between the undo and delete icons. A few end-users who questioned the functionality of the two icons when building their structures confirmed this. High-level and standard functions were seen to not require labels as they are recognisable functions. Two experts recommended moving the spinner menu to within the side-panel, instead of the action bar, which moves it closer the menu being changed.

One of the areas where the expert opinion differed the most was in their layout preference: each chose a different one for various reasons. Design V had a better spatial arrangement to aid workflow, Design H had more space allocated to the canvas, while Design C had a combination of less clutter and more working area space. This shows that even though the prototypes used Android design patterns, design is subjective and personal preference is a significant contributor to the final design.

Cosmetic problems were discovered throughout all designs that affected a range of heuristics. For example, all reviewers commented on the lack of a graphic "personality" or the dull colour scheme of the application, which also influences user experience and various usability heuristics. Recommendations were to improve the colour palette and icons. However, for this study, the default colours for the Android Holo theme was used in order to maintain consistency between designs and in keeping with the Android principles.

Experts also raised potential future usability issues, which correlates with previous studies [38]. These included learnability assistance for new users, and returning uses as well as the fragmentation issues, such as the scaling up and down of graphics for different resolution and sized devices. Toggling on and off a tutorial or help overlay was also seen as an important feature. Other functionalities beyond the scope of this study were also touched on by reviewers. Such suggestions included incorporating a dynamic naming label that states the formula or name of the structure as it is being built; selecting multiple items with one touch gesture; and importing, exporting and backing up data to a cloud service for collaboration purposes.

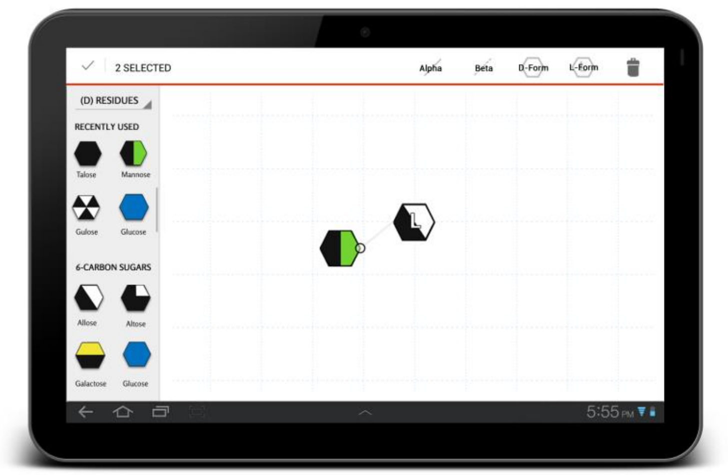
Efficiency was observed to be the main concern and highest usability problem uncovered by all of the reviewers. Various recommendations were given to overcome this problem area. The first was in the layout and organisation of the residues in the side panel. Two reviewers suggested that the side-by-side arrangements of the residues made scanning through the list faster. It was also suggested that residues be grouped either by frequently used, recently used or favourites. This would reduce the time taken to search through a long list. In combatting the confusion over where and how objects can be placed, it was suggested that a grid pattern be incorporated on top of the canvas. This would indicate a system of snap points on which users can be guided as to the placement of residues in relation to the canvas and neighbouring residues. Another useful suggestion was to use an edit button instead of the long-press feature to access the contextual menu. Even though long-pressing is closest to Android guidelines, it is confusing when used in combination with complicating the workflow and reducing efficiency.

8.3 Refined Design: Experts

The final expert design incorporated design recommendations proposed by the experts to increase usability and improve elements found to cause difficulty. Amendments were based on recommendations that reviewers had agreed upon. This included having a layout that allowed a larger canvas or working area, hints on functionalities including labels on icons, and elements that increase efficiency and aid the workflow. The main screen was arranged in a multi-pane layout, with the left pane containing the residues in a double column (Figure 8.3a) for faster searching. The residues are grouped according to "recently used" shapes (which could also be replaced with a "frequently used" option also suggested by the experts). The spinner was moved above its child element to indicate what navigation and lists that would it would effect. The D- and L-form toggle was removed and the separate lists were placed as categories within the drop-down menu to allow more room for the residues. Within the action bar, secondary functions where placed within the overflow menu to only leave the icons of primary functionality visible. Clear separation of the building and editing modes was therefore made more intuitive and evident through use of a noticeable edit feature. A tutorial overlay or walkthrough on the UI that explains main touch gestures and menus. This includes the drag-and-drop explanation when constructing the carbohydrate schematic and the spinner menu to change categories. The tutorial system only displays on first opening of the new application and can later be viewed in the "abouts" section when needed. However, tutorials, walkthroughs and on-screen overlays that display unsolicited advice are discouraged due to the implications that the application is complicated [72]. With mobile device applications, people tend to give a limited amount of their time and attention to an application on first use. Instructions are generally overlooked as people want to start using the application as soon as possible to discern if they can easily achieve their goals. Expert reviewers encouraged incorporation of a brief and visual tutorial to aid first time and returning users, however, for Glycano. A light grid system is also placed on the blank canvas to indicate where shapes can be moved and snapped into place.

Within the contextual menu screen, the action bar contained icons with the text description on top (Figure 8.3b) (Experts suggested that the customised icons have the label of the function besides the icon for new users). Selected residues and bonds were also made more obvious by greying out any unselected shapes and the number of items selected is highlighted within the top left of the action bar.

a)



b)

Figure 8.3: The refined design of a) the home screen upon opening the application and b) the contextual menu screen.

8.4 Discussion

A problem that can arise with surveys is subjectivity of the reviewers. The previous usability experience of experts may influence their effectiveness when rating heuristics, when preference over insight may dominate [62]. For example, our three experts showed preferences for difference designs. The cognitive walkthrough is also a detailed and time-consuming process, due to the need to evaluate tasks and simulate end-user thought processes. Evaluators tend to become too focused on usability issues that are task related, instead of possible issues outside of the task. In this study, the process had to be repeated three times with the alternate designs, which reviewers found tiresome. The combination of the cognitive walkthrough with the heuristic survey, however, prevented experts from providing too little or too much unfocused feedback. The heuristics provided guidance into what information was being sought and identified the important functionalities of the application, directed them to the context of this study rather than stray too often into areas outside the scope of what is required from them.

A main issue that arose during the survey was lack of device feedback provided for users in the designs. Touch screen devices use animated and haptic feedback to encourage and guide users [73]. This was an affordance not possible with paper prototypes, but is an element that should be investigated in further iterations of usability testing with functioning prototypes. An advantage of using expert users, therefore, is their ability to visualise animations and the device feedback that would occur when an action is triggered. Such factors would be accounted for in their opinions and enables them to discern which elements would be optimal for the application. Overall, the final design is an effective starting point for the developer on which to focus design efforts, as it incorporates usability issues that is seen in a broad range of Android applications. It covers the Android UI elements that would best achieve the application's objective for users as well as for optimal usage of the tablet screen. The experts also took into account what would increase the efficiency in constructing the carbohydrate and the intuitiveness of the controls to make it more user-friendly, including taking a small amount of time to learn the UI through guidance of a walkthrough.

CHAPTER 9

COMPARISON OF END-USER AND EXPERT EVALUATIONS

Expert reviewers assume end-user behaviour to evaluate whether users can successfully complete a task [51]. As this study's designs only used Android principles and guidelines, end-user behaviour with tasks could theoretically be clearly anticipated by usability experts. To investigate if experts' knowledge of user behaviour with the guidelines correlated with real end-user performance when navigating the designs, the commonalities and disparities between the feedbacks are reviewed (Table 9.1).

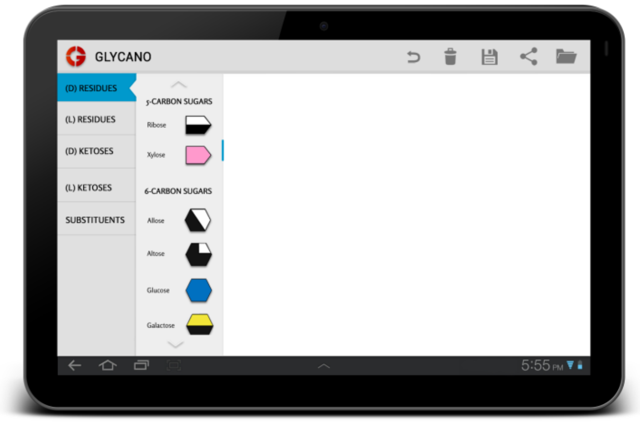
Several commonalities were found in both end-user and expert feedback of the designs. Firstly, both groups found the selection and the focus of selected objects were unclear. In all designs, selection occurs through long-pressing any single object on the canvas, be it either on a node or an edge of the structure. The long-pressing gesture is a common interaction with touch screen devices that is used to access an additional menu with actions that can occur to specific items being selected. In this application, the contextual menu indicates various functionalities, including changing the structural form of residues, the spatial arrangement of bonds, and deletion of items, that can be applied to the highlighted objects. The rest of the structure appears faded out to indicate unaffected regions unless tapped while in the contextual menu to be selected. Both the long-pressing gesture and selection emphasis proved to be confusing and found to be an unintuitive in this application's context. End-users had difficulty in deciphering how to manipulate the residues placed on the canvas, as well as which items had been selected once in the contextual menu. Experts had similar conclusions with this pattern in that selection via long-pressing was inefficient and unintuitive for first time users.

Expert users also stated that the drag-and-drop gesture may be difficult for first time users without guidance. This was evident in the usability testing when users took a longer time to process how items could be added to the canvas. Experts recommended a walk through or tutorial on opening the application, which can later be turned off.

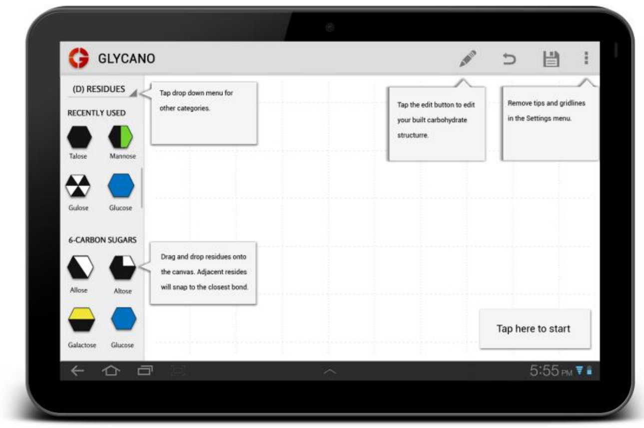
Based on the usability testing with end-users and feedback from experts, it can be seen that more critical observations and constructive recommendations were received from the latter group. Even though there were fewer expert participants than end-users, the amount of feedback was higher and justified their opinions, which allowed for further refinement and modification from the original designs. The usability responses were more general in nature. For example, both groups found searching for items difficult and suggested subsets or more logical grouping be implemented to locate residues faster. Where experts suggested the more commonly used "favourites" or "recently used" section that can be manually or dynamically changed according to user behaviour, end-users suggested the groupings be according to number of carbon atoms and alphabetically placed. The better comprehension of the subject matter aids in a more effective navigational pattern for unique lists, however, expert user recommendations could be used as an additional improvement in user experience and efficiency.

The main differences between expert and end-user feedback revolved around navigational patterns. Where end-users wanted functionalities and lists to be openly visible to get an immediate view of all options, experts preferred an uncluttered screen where only primary features can be easily accessed. For example, expert users tended to prefer the spinner or drop down menu as opposed to the tabbed menu as it allowed for additional categories to be added at a later stage and reduced the on-screen clutter. Design C was overall the best option due to the larger screen estate and the spinner was recommended in lieu of the group selectors in the other designs. End-users, however, did not perceive this element to be a clickable feature and required a longer period of time for them to consider this an option when the tasks required them to move into a different category. A few end-users even claimed to not understand this element and preferred other menu navigations. Even though designs were given at random for the usability testing, the spinner component consistently proved to be a navigational issue for users both familiar and unfamiliar to Android.

In the refined designs, the expert users suggest more changes to the original designs than in the usability test (Figure 9.1). Both groups preferred the multi-pane layout with the vertical scrolling and proper grouping of the objects for easier searching. End-users preferred all the icons visible with more visual cues, such as the up and down carats for scrolling. However, suggestions to overcome such problems were provided by the experts including the faint grid system on the canvas. Overall, end-users and expert reviewers provided varying levels of recommendations for the refined designs. Even though end-users had personal preferences to the given designs, they were comfortable in rather learning a UI than providing ideas on how to improve on them. Experts, on the other hand, gave far more suggestions and new ideas on what else could be included in the design to make the application more usable. They were able to uncover numerous problematic areas that corroborated end-user feedback and provided additional information regarding UI improvements and their limited understanding of the subject matter.



a)



b)

Figure 9.1: The refined design of the a) end-users compared to that of the b) experts.

9.1 Summary

End-users favoured and had fewer usability problems with the layout most commonly used for tablet applications: a multi-pane structure with vertical scrolling. Efficiency was most hindered when uncertainty they had arose when specific gestures are required to trigger an event. This included the drag-and-drop function for moving shapes onto the screen, and long-pressing an item to access the contextual menu. However, after this was learnt from the initial design testing, users did not face this problem again in the subsequent prototypes. Users also stated that all designs were generally usable and can be learnt fairly quickly. These results suggest that the guidelines do play a role in an improved user experience due to consistency and familiarity aspects, but confirm the experts' recommendations in including an initial tutorial overlay, a discouraged Android practice.

Expert reviewers confirmed that all the designs closely followed Android guidelines and principles. The simple and most uncluttered designs meant that in time users could learn and adapt to each design, which was confirmed by end-user feedback. However, additional affordances and feedback from the application was seen to be an essential change in aiding new and returning users. An initial walkthrough or tutorial, which can be disabled; space and logical flow of the working area regarding layout preferences; as well as faster item searches by grouping residues, were seen to be priority changes in the design refinement.

Overall, the expert users provided insightful and numerous recommendations to increase the usability of Glycano. They considered the application on a holistic level and highlighted common usability problems seen across a broad range of Android applications. These include placements of elements and content organisation, guiding users for complex actions or functionalities, and more efficient methods of searching for objects. They uncovered several problems that end-users also found difficult, such as the intuitiveness of the specific touch gestures and constructing of the schematic as well as an unclear selection of objects. More recommendations in overcoming these issues, however, were provided by the experts than the end-users. End-users were able to provide more content specific feedback, due to their understanding of the application topic. However, as new users, they preferred to have the functionalities visible in order to view what was available to them without requiring to search for it. Experts preferred an uncluttered screen, where only the necessary and frequently used options are visible. Experts also were more vocal in their personal opinions on how they would prefer the application's layout and visual appearance. Even though common problems of the UI were uncovered by the experts, their personal opinion and recommendations for the designs were indicated in the final interview question, which suggests that subjective personal preference holds a significant influence on the final design. Both end-users and experts provided valuable information on usability problems that would have gone unnoticed should testing not have been performed. The experts, however, had the advantage of industry knowledge and experience within the usability field to pinpoint major flaws and the confidence in suggesting improvements and changes to the design. Furthermore, as experts do not need to have expertise in the domain area of the application, sourcing usability experts may not be as time consuming as end-users. In this regard, approaching usability experts to critique initial prototype designs may be a faster and insightful exercise than recruiting end-users.

Table 9.1: Expert and End-user Feedback of Static and Dynamic Elements.

Static			
Group	*Guideline Property*	*Design Analysis: Experts*	*Design Analysis: End-users*
Layout	Grouping and spacing of icons	• Only most important icons to be shown on the action bar • Delete and undo icons may be confused. • Delete icon could easily be tapped by accident. • Move delete icon to the overflow menu • Good spatial arrangement but requires more canvas space.	• No hidden icons preferred. • Unfamiliar with overflow menu icon. • Delete icon preferred as a secondary function.
	Grouping and Ordering of list items	• Subsets were suggested for easier location of items e.g. “recently used” or “favourites”	• Subsets or arrangement of items were suggested for easier location of items e.g. Categorise by number of carbon atoms and/or alphabetically.
	Application layout	• “I prefer Design V since I like the spatial management and the layout seems to aid workflow and enhance the functionality of the Application.”	• Multi-pane with vertical lists the most preferred and most similar to their Android familiarity.

		• "Mockup H, it gives me the most space to draw my structure." • "Mockup C is the best. Having the high-level group name in a spinner is more efficient than Mockup 1. The long-press contextual menu is closest to Android guidelines. The canvas is a nice proportion and feels bigger. More space in the action bar means more high-level functions exposed."	• Combination design found to have more screen space but navigational options were not obvious. • Multi-pane layout required de-cluttering • Menu panel is too cluttered in combination design.
Terminology	Abbreviations / labels on customised icons	• Labels within the contextual menu lacked consistency with the unlabelled icons on the action bar.	• Preferred labels on the customised icons within the contextual menu.
	Naming of secondary functions	• Place label with the undo icon, as it is the most primary function • Preferred labels on the customised icons within the contextual menu.	• Confusion between delete and undo icons. • 'Alpha' and 'Beta' labels could be shortened to α and β, respectively.

Dynamic			
Group	*Guideline Property*	*Design Analysis - Experts*	*Design Analysis – End-users*
User input	Selection of menu items	• The spinner was the preferred group selector as it held children categories more effectively, allowed for further additions and freed up screen estate. Required movement closer to its children lists. • The horizontal grouped tabs on the action bar suggest swiping through entire work screens.	• The spinner component was overlooked as a menu button and was found to be confusing as a parent category. • The horizontal fixed tabs gave the impression of a Windows OS environment rather than an Android application and too complicated. • Few liked the D and L- form residue toggle in the combination design.
	Selection of bond	• Long-pressing was inefficient and unintuitive. • Contextual action bar required labels. • Focus on selected objects was unclear. • Contextual menu was preferred to allow for multiple changes to be made at once. • Further questions asked: What happens when you tap an item in the drawer instead of dragging it on to the working area? What happens if you tap on an item in the working area instead of long-press?	• Long-pressing was confusing and unintuitive. • Contextual action bar required labels • Focus on selected objects was unclear. • Pop up notification was easier to detect after selection but was only useful for modification of single objects.

Scrolling	• Vertical lists are easier to scroll than horizontal lists.	• Vertical lists are easier to scroll and more intuitive than horizontal lists. • Prefers to scroll vertically when there are long lists of items. • Some users liked the arrow bars. Lists required additional on-screen indications that they were scrollable and not fixed to what was thought only a few options.
Moving of residues	• Possibly confusing for first time users. Drag-and-drop would require tutorial or on-screen tip on opening of the application. Snap-to-grid system would be an option as guidance on where to place objects and distance required to form the bond.	• Drag-and-drop was confusing for the first mock-ups. Most common impression was to tap the residue and expect the object to appear on the canvas. Difficulty in determining where on the canvas they were allowed to place the object and questioning of how bonds were formed.

CHAPTER 10

CONCLUSIONS AND FUTURE WORK

In evaluating which UIM is more effective for initial design prototypes, usability testing and expert heuristic evaluations are compared. Three prototypes employing different interpretations of Android principles and guidelines were created for a unique tablet application. Two approaches were employed to evaluate the usability of the designs: end-user testing and usability expert reviews. The effective UI designs and patterns that allowed for the most efficient building of a carbohydrate structural schematic were determined for each group. End-user testing placed the application designs in front of target users and feedback was obtained through completion of task-based objectives. Usability experts were recruited to confirm Android guideline associations and assess whether designs would be appropriate forms of navigation for the objectives of the application, Glycano. They assessed the performance of the mockups against typical user behaviour, and measuring it against a set of heuristics.

In answering the research questions posed in Chapter 1.3:

- *Can an intuitive UI be designed for a new tablet application using only Android patterns?*

It can be seen that using solely Android patterns can improve usability to an extent. In general, incorporating the guidelines in the foundations of an application is beneficial for end-users and developers in producing a more usable application. However, selecting the appropriate patterns to organise the content and intuitive gestures for user action patterns are important design decisions. Even with different interpretations of the UI guidelines, minor initial usability problems arose where improvements to efficiency and learnability heuristics in creating the carbohydrate structures could further be refined.

- *Is usability testing or expert heuristic evaluations beneficial in gaining valuable feedback for refinement of an initial application design?*

Overall, expert reviewers provided more constructive and in-depth feedback than end-users, which considered longer-term application improvements and a wider target audience. Expert users also inquired and probed more into the functionality and touch gesture controls. Questioning of the environment was seen less with end-users.

However, familiarity of the scientific construction of the carbohydrates aided in their understanding of the practical usages of the application. End-user behaviour was therefore observed to conform to the provided designs, in that they agreed that they could work with any of the given designs over time, as long it was easy to learn the navigational patterns. This was seen when different mockups were given a lower usability rating than the others mainly due to them being the initial and unfamiliar testing design. Usability problems were uncovered by both groups. However, more problems and design recommendations provided by experts.

In conclusion, first launch of an application is a crucial stage in the decision-making process of the continued use of an application. An unsatisfactory initial experience means a high probability chance that the application will be uninstalled. Furthermore, there is a necessity for cheap and efficient evaluation methods for Android designs in order to produce a user-friendly application. Android guidelines and patterns will continuously be updated in future, and using these principles for the foundational design of an application is advantageous in providing a certain level of usability and consistency. Adhering to these patterns is an important foundational step when designing the application. Misinterpretations or inappropriate selection of unintuitive elements or layouts, however, is a possibility for developers when attempting to incorporate latest UI component libraries. Usability studies with end-users or experts can aid in avoiding oversights and premature optimisations. Additionally, within the context of a more specialized application with a niche target group, not only will asking end-users be advantageous in the design process, it would decrease costs in acquiring a usability expert and prevent over-designing to a specific user's preference. Putting a prototype application in front of real users can make a difference in delivering a clear and effective design for the developed product. Access to usability experts, however, has been shown here to provide more information on the overall UI design and confirms proper usage of Android principles. It is a fast method for gaining feedback on significant usability problems and are able to provide practical recommendations. Whether usability tests or expert evaluations are performed, the feedback gained is better than not running any usability tests at all. Refining of a design is an important step before start of development, but ensuring that predictability and consistency are maintained during this process may prove to improve on initial user experience and application uptake.

10.1 Future work

There are numerous complex factors that contribute to an overall successful user experience of a touch screen based application. These include responsive design and optimised coding and image management within the Android framework to improve performance on various devices. A large factor in a positive usability also includes visual, haptic and aural feedback. High-fidelity paper prototypes cannot provide the true potential that a developed prototype can afford with animations and handling of a real device. Interactivity with the device and a semi-functional application may provoke a different view or user reaction that could provide another dimension to usability and should be investigated.

Furthermore, extended use of a prototype application may also provide a more in depth response and could be used as an additional round of usability testing if enough time is afforded. An advantage of live mobile applications is that there is constant feedback from real users and regular updates to the software can be made, which adds to the success of an application. However, in this study, investigations of initial usability steps that can be taken by freelance Android developers are considered in increasing the success of their application on first release.

REFERENCES

[1] J. Y. Choi, J. Shin, and J. Lee, “Strategic demand forecasts for the tablet PC market using the Bayesian mixed logit model and market share simulations,” *Behav. Inf. Technol.*, vol. 32, no. 11, pp. 1177–1190, Oct. 2013.

[2] “Android domination to continue in 2014; iPhone loses ground » Telecoms.com.” [Online]. Available: http://www.telecoms.com/210391/android-domination-to-continue-in-2014-iphone-loses-ground/. [Accessed: 26-May-2014].

[3] “Developer Economics Q3 2013: State of the Developer Nation.” [Online]. Available: http://www.visionmobile.com/product/developer-economics-q3-2013-state-of-the-developer-nation/. [Accessed: 26-May-2014].

[4] “Research: How Many Apps Are in Each App Store? | Pure Oxygen Labs.” [Online]. Available: http://www.pureoxygenlabs.com/how-many-apps-in-each-app-store/. [Accessed: 27-Aug-2014].

[5] R. Harrison, D. Flood, and D. Duce, “Usability of mobile applications: literature review and rationale for a new usability model,” *J. Interact. Sci.*, vol. 1, no. 1, pp. 1–16, 2013.

[6] H. K. Ham and Y. B. Park, “Designing knowledge base mobile application compatibility test system for android fragmentation,” *Int. J. Softw. Eng. Its Appl.*, vol. 8, no. 1, pp. 303–314, 2014.

[7] A. Holzer and J. Ondrus, “Mobile application market: A developer’s perspective,” *Telemat. Informatics*, vol. 28, no. 1, pp. 22–31, Feb. 2011.

[8] “Android: A visual history | The Verge.” [Online]. Available: http://www.theverge.com/2011/12/7/2585779/android-history. [Accessed: 12-Nov-2014].

[9] “Android Design Principles | Android Developers.” [Online]. Available: http://developer.android.com/design/get-started/principles.html. [Accessed: 23-Feb-2013].

[10] “Differences Between iOS 7 and Android 4.” [Online]. Available: http://despreneur.com/differences-between-ios-7-and-android-4/. [Accessed: 12-Nov-2014].

[11] M. J. Albers and L. Kim, “User web browsing characteristics using palm handhelds for information retrieval,” in *Proc. IPCC/SigDoc Conf*, 2000, no. 1998, pp. 125–135.

[12] J. Lehtimaki, *Smashing Android UI*. John Wiley & Sons, 2012.

[13] M. W. Seraj and C. Yui, “A study of user interface design principles and requirements for developing a mobile learning prototype,” in *International Conference on Computer & Information Science*, 2012, vol. 2, pp. 1014–1019.

[14] N. Bevan, “The use of guidelines in menu interface design: Evaluation of a draft standard,” in *Proceedings of the IFIP TC13 Third Interational Conference on Human-Computer Interaction*, 1990, pp. 435–440.

[15] A. Abran, A. Khelifi, W. Suryn, and A. Seffah, “Consolidating the ISO Usability Models,” in *11th International Software Quality Management Conference*, 2003, pp. 1–17.

[16] “Usability 101: Introduction to Usability.” [Online]. Available: http://www.nngroup.com/articles/usability-101-introduction-to-usability/. [Accessed: 28-Aug-2014].

[17] “International ergonomic requirements for office work with visual display terminals (VDTs) - Part 11 : Guidance on usability,” *ISO/IEC 92*, 1998.

[18] J. Nielsen, H. Desurvire, R. Jeffries, and S. Jones, “Usability Inspection Methods,” in *Usability inspection methods, Conference companion on Human factors in computing systems*, 1995, pp. 377–378.

[19] R. Jeffries and H. Desurviret, “Usability testing vs heuristic evaluation: Was there a contest?,” in *ACM SIGCHI Bulletin*, 1992, no. October, pp. 39–41.

[20] D. Zhang and B. Adipat, “Challenges, methodologies, and issues in the usability testing of mobile applications,” *Int. J. Hum. Comput. Interact.*, vol. 18, no. 3, pp. 293–308, 2005.

[21] J. Kjeldskov, C. Graham, S. Pedell, F. Vetere, S. Howard, S. Balbo, and J. Davies, “Evaluating the usability of a mobile guide: The influence of location, participants and resources,” *Behav. Inf. Technol.*, vol. 24, no. 1, pp. 51–65, Jan. 2005.

[22] J. Kjeldskov and J. Stage, “New techniques for usability evaluation of mobile systems,” *Int. J. Hum. Comput. Stud.*, vol. 60, no. 5–6, pp. 599–620, May 2004.

[23] M. Tory and T. Moller, “Evaluating Visualizations: Do Expert Reviews Work?,” in *IEEE Computer Graphics and Applications*, 2005, vol. 25, no. 5, pp. 8–11.

[24] J. Watson, “Glycano.” [Online]. Available: http://people.cs.uct.ac.za/~jwatson/. [Accessed: 04-Feb-2015].

[25] A. J. Williams, S. Ekins, A. M. Clark, J. J. Jack, and R. L. Apodaca, “Mobile apps for chemistry in the world of drug discovery,” *Drug discovery today*, vol. 16, no. 21–22, Elsevier Ltd, pp. 928–39, Nov-2011.

[26] A. Butler, S. Izadi, and S. Hodges, “SideSight: Multi-‘touch’ interaction around small devices,” in *Proceedings of the 21st annual ACM symposium on User interface software and technology*, 2008, pp. 201–204.

[27] Q. Guo, H. Jin, D. Lagun, S. Yuan, and E. Agichtein, “Mining touch interaction data on mobile devices to predict web search result relevance,” in *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, 2013, pp. 153–162.

[28] E. McVicar, “Designing for Mobile, Part 2: Interaction Design | UX Booth.” [Online]. Available: http://www.uxbooth.com/articles/designing-for-mobile-part-2-interaction-design/. [Accessed: 17-Nov-2014].

[29] “Tabs | Android Developers.” [Online]. Available: http://developer.android.com/design/building-blocks/tabs.html. [Accessed: 17-Nov-2014].

[30] “Schematic - Android Apps on Google Play.” [Online]. Available: https://play.google.com/store/apps/details?id=ru.ustimov.schematic&hl=en. [Accessed: 06-Oct-2014].

[31] “SchematicMind Free mind map - Android Apps on Google Play.” [Online]. Available: https://play.google.com/store/apps/details?id=com.qdvsoftworks.schematicmind&hl=en. [Accessed: 06-Oct-2014].

[32] Molsoft, “iMolview Lite - Android Apps on Google Play,” 2014. [Online]. Available: https://play.google.com/store/apps/details?id=com.molsoft.imolview&hl=en. [Accessed: 18-Nov-2014].

[33] Biochem_fan, “NDKmol - molecular viewer - Android Apps on Google Play,” 2013. [Online]. Available: https://play.google.com/store/apps/details?id=jp.sfjp.webglmol.NDKmol&hl=en. [Accessed: 06-Oct-2014].

[34] J. Feldt, R. a Mata, and J. M. Dieterich, “Atomdroid: A Computational Chemistry Tool for Mobile Platforms.,” *J. Chem. Inf. Model.*, vol. 52, pp. 1072–1078, Mar. 2012.

[35] M. M. Informatics, “ChemSpider Mobile - Android Apps on Google Play,” 2013. [Online]. Available: https://play.google.com/store/apps/details?id=com.mmi.android.chemspider&hl=en. [Accessed: 18-Nov-2014].

[36] B. Still and J. Morris, “The Blank-Page Technique: Reinvigorating Paper Prototyping in Usability Testing,” *IEEE Trans. Prof. Commun.*, vol. 53, no. 2, pp. 144–157, Jun. 2010.

[37] R. Sefelin, M. Tscheligi, and V. Giller, “Paper prototyping - What is it good for? A comparison of paper- and computer-based low-fidelity prototyping,” in *CHI ’03 Extended Abstracts on Human Factors in Computing Systems*, 2003, pp. 778–779.

[38] J. Sauer and A. Sonderegger, “The influence of prototype fidelity and aesthetics of design in usability tests: effects on user behaviour, subjective evaluation and emotion.,” *Appl. Ergon.*, vol. 40, no. 4, pp. 670–7, Jul. 2009.

[39] M. Walker, L. Takayama, and J. a. Landay, “High-Fidelity or Low-Fidelity, Paper or Computer? Choosing Attributes when Testing Web Prototypes,” *Proc. Hum. Factors Ergon. Soc. Annu. Meet.*, vol. 46, no. 5, pp. 661–665, 2002.

[40] G. Nolan, O. Cinar, and D. Truxall, “Android Best Practices,” 2014.

[41] L. Rivero and T. Conte, “Using an empirical study to evaluate the feasibility of a new usability inspection technique for paper based prototypes of web applications,” in *Brazilian Symposium on Software Engineering*, 2012, pp. 81–90.

[42] T. Hollingsed, D. G. Novick, and E. Paso, “Usability inspection methods after 15 years of research and practice,” in *Proceedings of the 25th annual ACM International conference on Design of communication*, 2007, pp. 249–255.

[43] M. Y. Ivory and M. a. Hearst, “The state of the art in automating usability evaluation of user interfaces,” *ACM Comput. Surv.*, vol. 33, no. 4, pp. 470–516, Dec. 2001.

[44] R. Jeffries, J. R. Miller, C. Wharton, and K. M. Uyeda, “User interface evaluation in the real world : A comparison of four techniques,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1991, vol. 91, no. c, pp. 119–124.

[45] M. W. M. Jaspers, “A comparison of usability methods for testing interactive health technologies: Methodological aspects and empirical evidence,” *Int. J. Med. Inform.*, vol. 78, no. 5, pp. 340–53, May 2009.

[46] H. X. Lin, Y.-Y. Choong, and G. Salvendy, “A proposed index of usability: A method for comparing the relative usability of different software systems,” *Behav. Inf. Technol.*, vol. 16, no. 4–5, pp. 267–277, Jan. 1997.

[47] M. S. Andreasen, H. V. Nielsen, S. O. Schrøder, and J. Stage, “What happened to remote usability testing?,” in *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI ’07*, 2007, p. 1405.

[48] S. Rosenbaum, J. A. Rohn, and J. Humburg, “A toolkit for strategic usability,” in *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI ’00*, 2000, pp. 337–344.

[49] K. Seibel and B. Ru, “The influence of user expertise and prototype fidelity in usability tests,” *Appl. Ergon.*, vol. 41, pp. 130–140, 2010.

[50] J. Nielsen and R. Molich, “Heuristic evaluation of user interfaces,” in *Proceedings of the SIGCHI conference on Human factors in computing systems Empowering people*, 1990, pp. 249–256.

[51] P. G. Polson, C. Lewis, J. Rieman, and C. Wharton, “Cognitive walkthroughs: A method for theory-based evaluation of user interfaces,” *Int. J. Man. Mach. Stud.*, vol. 36, no. 5, pp. 741–773, May 1992.

[52] C. Lewis and C. Wharton, “Cognitive walkthroughs,” in *Handbook of Human-Computer Interaction*, M. G. Helander, Ed. Elsevier, 1997, pp. 717–732.

[53] J. Nielsen, “Finding usability problems through heuristic evaluation,” *Proc. SIGCHI Conf. Hum. factors Comput. Syst. - CHI ’92*, pp. 373–380, 1992.

[54] J. Nielsen and V. L. Phillips, “Estimating the relative usability of two interfaces,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1993, pp. 214–221.

[55] Z. Tang, T. R. Johnson, R. D. Tindall, and J. Zhang, "Applying heuristic evaluation to improve the usability of a telemedicine system.," *Telemed. J. E. Health.*, vol. 12, no. 1, pp. 24–34, Feb. 2006.

[56] R. Racadio, E. Rose, S. Boyd, and H. A. Nw, "Designing and evaluating the mobile experience through iterative field studies," in *Proceedings of the 30th ACM international conference on Design of communication*, 2012, pp. 191–196.

[57] A. Hussain and M. Kutar, "Apps vs devices: Can the usability of mobile apps be decoupled from the device?," *Int. J. Comput. Sci. issues*, vol. 9, no. 3, pp. 11–16, 2012.

[58] M. de Sá and L. Carriço, "Lessons from early stages design of mobile applications," in *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, 2008, p. 127.

[59] B. S. Jin and Y. G. Ji, "Usability risk level evaluation for physical user interface of mobile phone," *Comput. Ind.*, vol. 61, no. 4, pp. 350–363, May 2010.

[60] E. G. Nilsson, "Design patterns for user interface for mobile applications," *Adv. Eng. Softw.*, vol. 40, no. 12, pp. 1318–1328, Dec. 2009.

[61] A. K. Anne Kaikkonen, Aki Kekäläinen, Mihael Cankar, Titti Kallio, "Usability testing of mobile applications: A comparison between laboratory and field testing," *J. Usability Stud.*, vol. 1, no. 1, pp. 4–16, 2005.

[62] P. Tsai, "A survey of empirical usability evaluation methods," *GSLIS Indep. study*, pp. 1–18, 1996.

[63] V. S. R. Rao, *Conformation of Carbohydrates*. CRC Press, 1998.

[64] D. J. Harvey, A. H. Merry, L. Royle, M. P. Campbell, R. a Dwek, and P. M. Rudd, "Proposal for a standard system for drawing structural diagrams of N- and O-linked carbohydrates and related compounds.," *Proteomics*, vol. 9, no. 15, pp. 3796–801, Aug. 2009.

[65] D. J. Harvey, A. H. Merry, L. Royle, M. P. Campbell, and P. M. Rudd, "Symbol nomenclature for representing glycan structures: Extension to cover different carbohydrate types.," *Proteomics*, vol. 11, no. 22, pp. 4291–5, Nov. 2011.

[66] R. a. Virzi, J. L. Sokolov, and D. Karis, "Usability problem identification using both low- and high-fidelity prototypes," in *Proceedings of the SIGCHI conference on Human factors in computing systems common ground*, 1996, pp. 236–243.

[67] "Adobe Photoshop CS5." Adobe Systems, California, 2010.

[68] M. Tohidi, W. Buxton, R. Baecker, and A. Sellen, "Getting the right design and the design right: Testing many is better than one," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2006, pp. 1243–1252.

[69] J. Rubin and D. Chisnell, *Handbook of usability testing: How to plan, design, and conduct effective tests*. John Wiley & Sons, 2008.

[70] W. Park, S. H. Han, S. Kang, Y. S. Park, and J. Chun, “A factor combination approach to developing style guides for mobile phone user interface,” *Int. J. Ind. Ergon.*, vol. 41, no. 5, pp. 536–545, Sep. 2011.

[71] R. S. Dicks, “Mis-usability: On the uses and misuses of usability testing,” 2002, pp. 26–30.

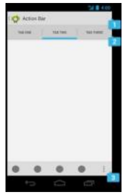
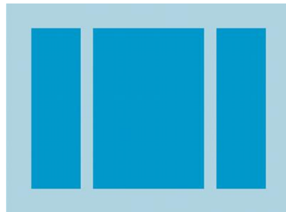
[72] “Introduction to Android Design Patterns - Tuts+ Code Article.” [Online]. Available: http://code.tutsplus.com/articles/introduction-to-android-design-patterns--cms-20808. [Accessed: 22-Aug-2014].

[73] K. Salo, L. Arhippainen, and S. Hickey, “Design guidelines for hybrid 2D/3D user interfaces on tablet devices. A user experience evaluation,” in *The Fifth International Conference on Advances in Computer-Human Interactions*, 2012, no. c, pp. 180–185.

APPENDIX 1

ANDROID PATTERNS USED IN GLYCANO

Android Pattern	Description	Example Image
Action Bar	• Located at the top of the UI view consisting of the corporate logo, screen title, contextual actions and the overflow menu for secondary menu options.	
	• Adapts to the orientation of the phone or a larger tablet screen size by stretching across the screen, which also allows for more components to be added.	
Contextual action bar	• Visible after selection of single or multiple items through long-pressing or selection of an item.	
Scrolling	• Diagrammatical representations of the a) scrolling and b) drag-and-drop finger gestures	a)  b) 

Fixed tabs	• Fixed tabs are placed within or under the action bar. They hold various menu options or screen views.	
Drop down list / spinner	• Within the view control of the action bar	
Custom overlay	• Custom overlay to show quick actions on a single item.	
Multi-pane layout	• Multi-pane layout on a tablet screen. On a phone, two separate views would be required, however, a) on a larger screen both views can be displayed at once. b) The arrangement of the panels is in a linear layout as opposed to the grid or list views.	a)  b) 

APPENDIX 2

ANDROID GUIDELINES AND DESCRIPTIONS

1. Enchant me

a. ***Delight me in surprising ways:*** Describes subtle animations and effects for feedback.

b. ***Real objects are more fun than buttons and menus:*** Directly manipulating objects are easier than menus and buttons.

c. ***Let me make it mine:*** Allow customization of the application such as changing the background.

d. ***Get to know me:*** Make user preferences accessible.

2. Simplify my life

a. ***Keep it brief:*** Keep sentences short and simple.

b. ***Pictures are faster than words:*** Utilising images gets user's attention faster and makes navigation more efficient as opposed to text-based controls.

c. ***Decide for me but let me have the final say:*** Reduce the number of choices and decisions users have to make to complete a task. Additionally, allow for an undo option.

d. ***Only show what I need when I need it:*** Hide unessential options and only show main and frequently used options.

e. ***I should always know where I am:*** Utilise transitions and feedback to show users which screen they are on and can navigate to.

f. ***Never lose my stuff:*** Save settings, creations and customisations that users have created.

g. ***If it looks the same, it should act the same:*** Different functionalities should be visually distinct.

h. ***Only interrupt me if it's important:*** Unnecessary notifications should be avoided to avoid interrupting the user.

3. Make me amazing

a. ***Give me tricks that work everywhere:*** Leverage visual patterns and muscle memory to make the application easier to use.

b. ***It's not my fault:*** Provide clear recovery instructions if an error is made. Make the user feel smart.

c. ***Sprinkle encouragement:*** Provide subtle feedback and make tasks smaller that can be easily achieved.
d. ***Do the heavy lifting for me:*** Provide shortcuts and extra functionality.
e. ***Make important things fast:*** Make the most important features the most accessible.

APPENDIX 3.1

USABILITY TESTING: BIOGRAPHICAL QUESTIONNAIRE

Glycano Usability Test

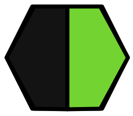
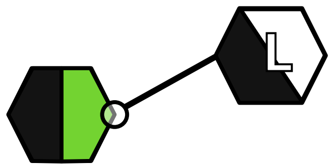
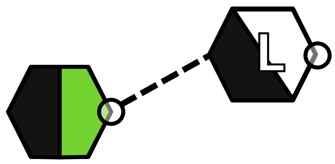
Name:		**Age:**	
Gender:	☐ Male ☐ Female		
Degree:		**Year of study:**	

1.	Do you own an Android phone?	☐ No ☐ Yes
	If so, for how long have you been using it?	
	What Android version it is running?	
3.	Do you own an Android tablet?	☐ No ☐ Yes
	If so, how long have you been using it?	
	What Android version is it running?	
3.	What are your most used apps?	

APPENDIX 3.2

USABILITY TESTING: USER TASKS

Tasks:

	Task	*Result*
1.	Add your first **residue** (Mannose) onto the blank canvas:	
2.	Join an L-Allose **residue** to the right of the Mannose residue:	
3.	You now want to change the linkage between the 2 residues to a beta bond, represented by a dashed line.	
4.	You now want to save your molecule.	

APPENDIX 3.3

USABILITY TESTING: POST-TASK SURVEY

Questionnaire

	Please tick the most applicable statement you agree with for each question:

		Strongly disagree	***Disagree***	***Neutral***	***Agree***	***Strongly agree***
1.	I found the system unnecessarily complex					
2.	I thought the system was easy to use					
3.	I recognise all of the Android controls used					
4.	I would imagine that most people would learn to use this system very quickly					
5.	I would need a lot of time to get used to working with this system					

APPENDIX 3.4

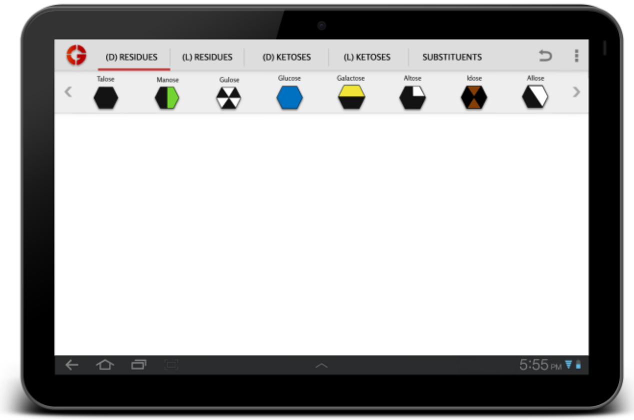
USABILITY TESTING: POST-TESTING QUESTIONNAIRE

1.	Overall, which mockup do you think is the easiest to use and why? What did you like best about it?
2.	Which mockup do you think is the hardest to use and why? What did you like least about it?
3.	In terms of design, is there anything that you would add or change to any of the designs that would make the layout easier to understand or make building the carbohydrates more efficient?
4.	Do you have any final thoughts or comments?

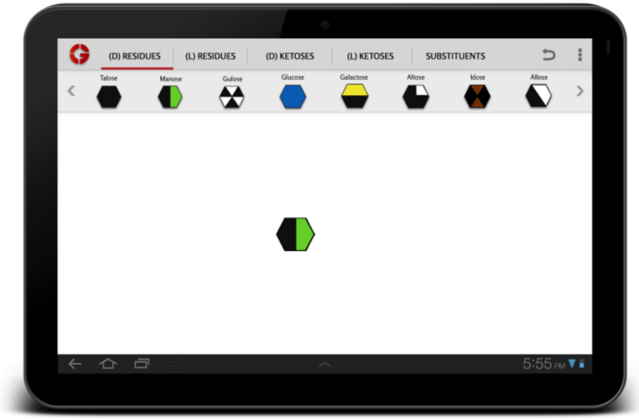
APPENDIX 4.1

USABILITY TESTING: PROTOTYPE DESIGN 1 (V) SCREENS

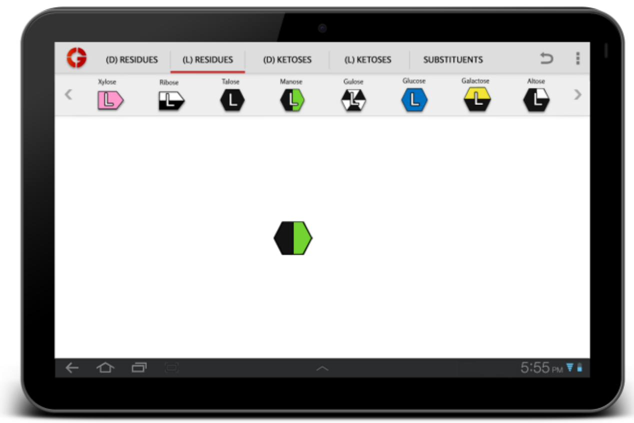
1.



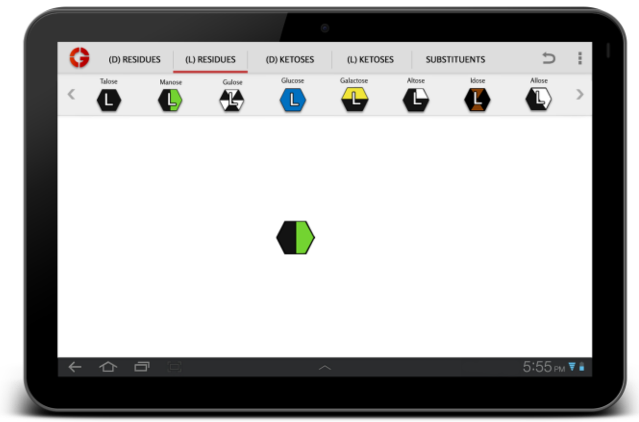
2.



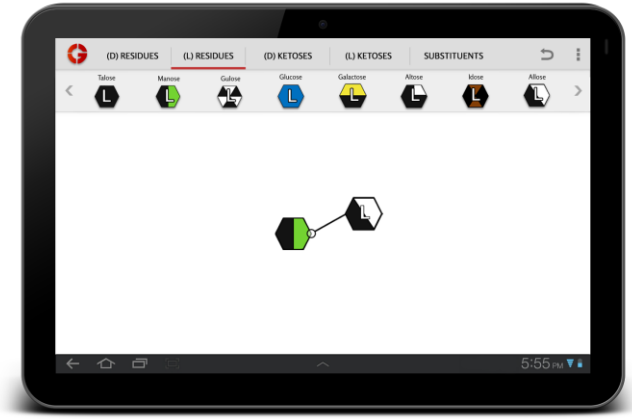
3.



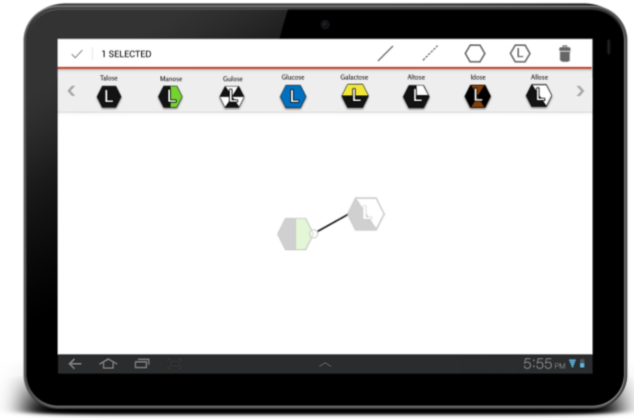
4.



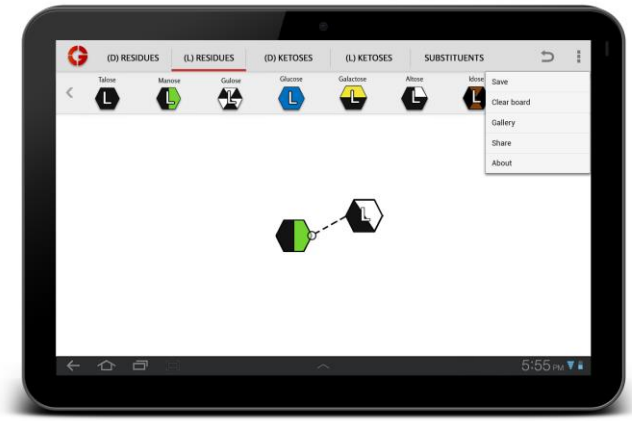
5.


(D) RESIDUES
(L) RESIDUES
(D) KETOSES
(L) KETOSES
SUBSTITUENTS
Talose
Manose
Gulose
Glucose
Galactose
Altose
Idose
Allose
5:55 PM

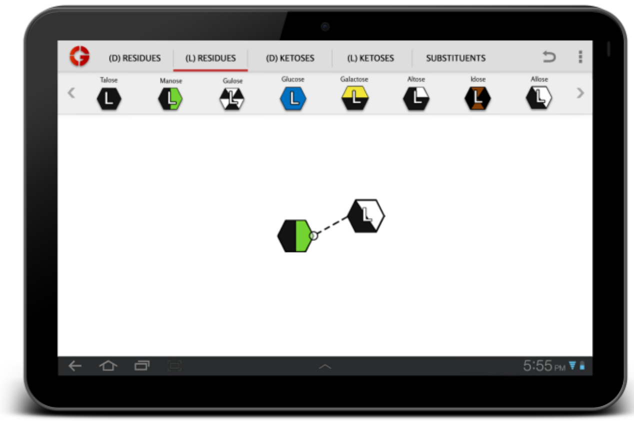
6.


1 SELECTED
Talose
Manose
Gulose
Glucose
Galactose
Altose
Idose
Allose
5:55 PM

7.


(D) RESIDUES
(L) RESIDUES
(D) KETOSES
(L) KETOSES
SUBSTITUENTS
Talose
Manose
Gulose
Glucose
Galactose
Altose
Idose
Save
Clear board
Gallery
Share
About
5:55 PM

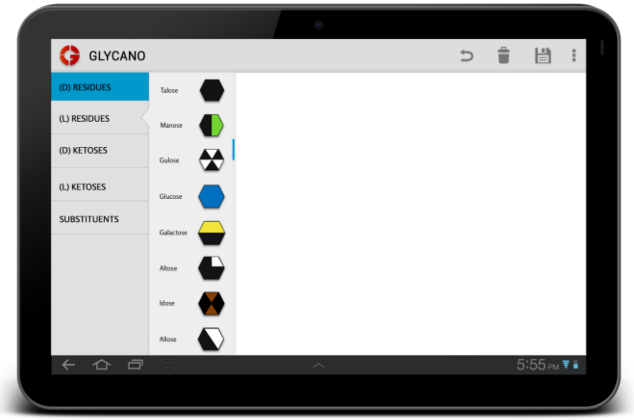
8.


(D) RESIDUES
(L) RESIDUES
(D) KETOSES
(L) KETOSES
SUBSTITUENTS
Talose
Manose
Gulose
Glucose
Galactose
Altose
Idose
Allose
5:55 PM

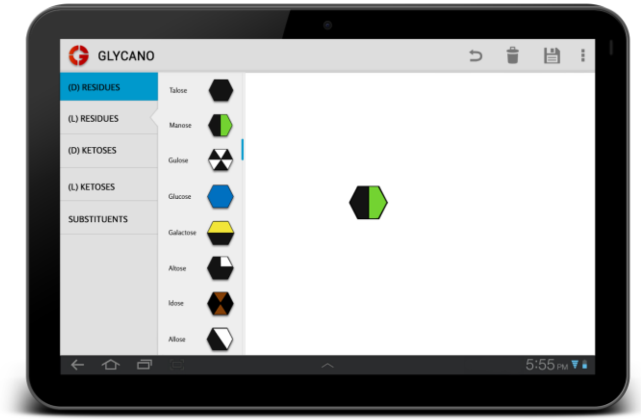
APPENDIX 4.2

USABILITY TESTING: PROTOTYPE DESIGN 2 (H) SCREENS

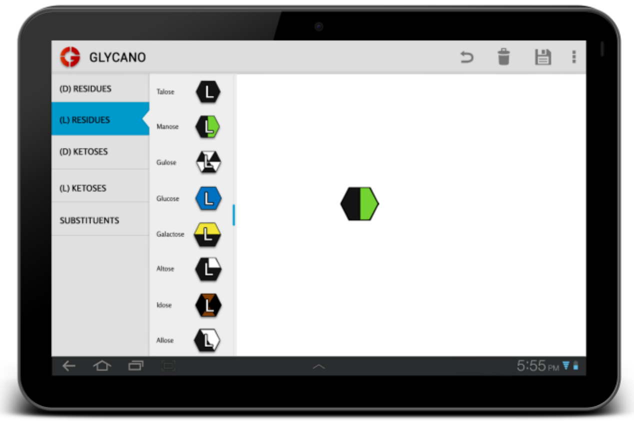
1.



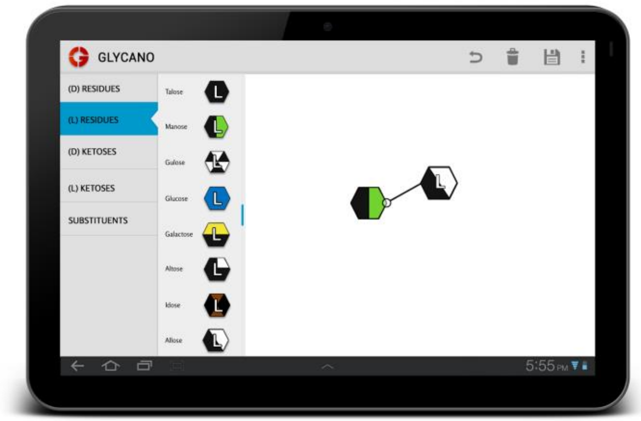
2.



3.



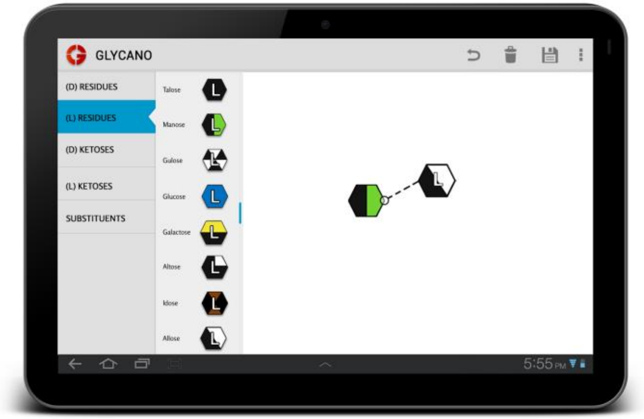
4.



5.


GLYCANO
(D) RESIDUES
(L) RESIDUES
(D) KETOSES
(L) KETOSES
SUBSTITUENTS
Talose
Manose
Gulose
Glucose
Galactose
Altose
Idose
Allose
Convert to beta bond
Delete
5:55

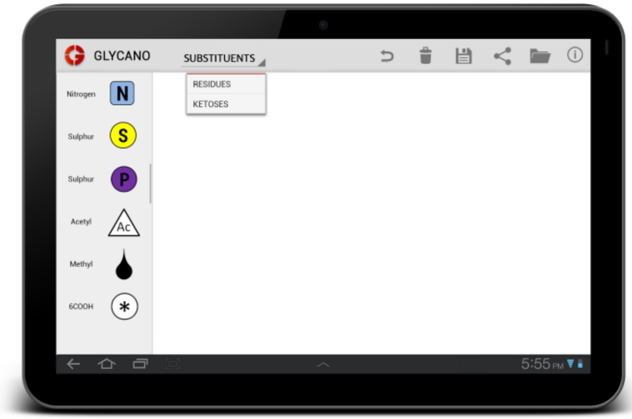
6.


GLYCANO
(D) RESIDUES
(L) RESIDUES
(D) KETOSES
(L) KETOSES
SUBSTITUENTS
Talose
Manose
Gulose
Glucose
Galactose
Altose
Idose
Allose
5:55

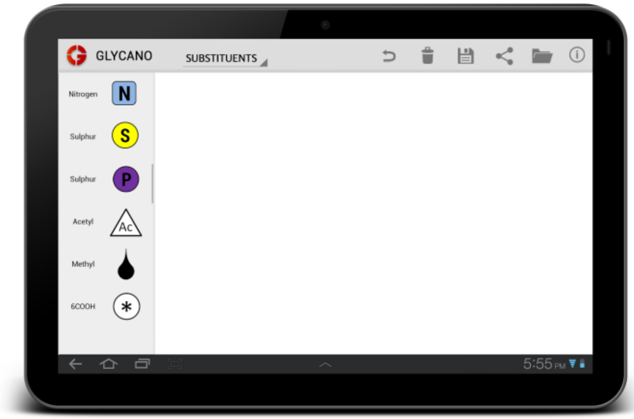
APPENDIX 4.3

USABILITY TESTING: PROTOTYPE DESIGN 3 (C) SCREENS

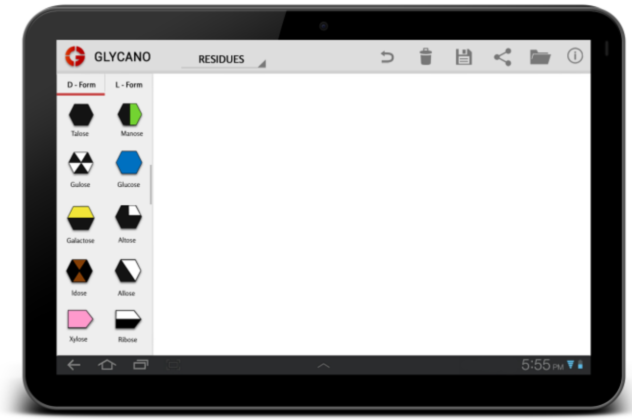
1.



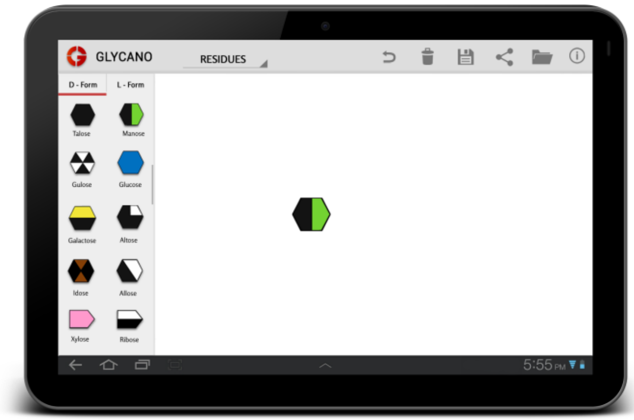
2.



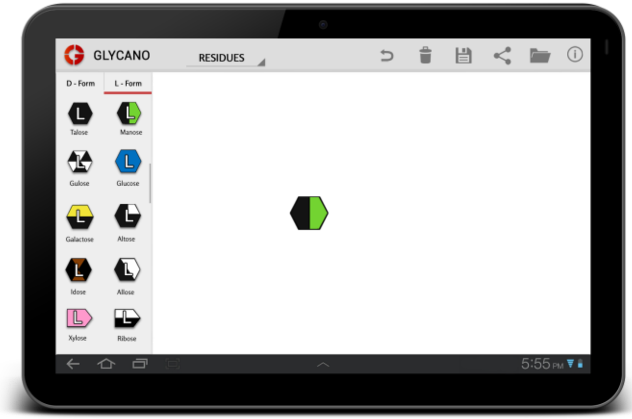
3.



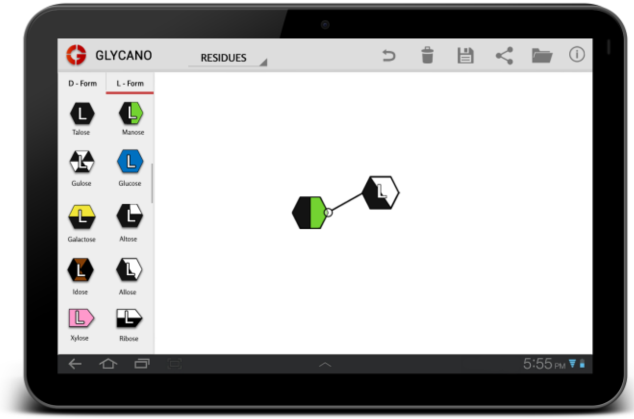
4.



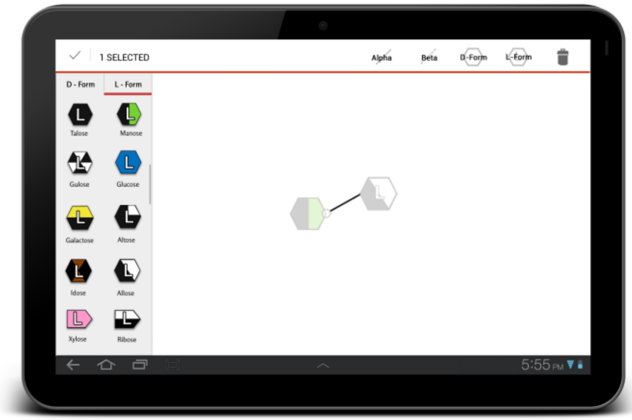
5.


GLYCANO
RESIDUES
D - Form
L - Form
Talose
Manose
Gulose
Glucose
Galactose
Altose
Idose
Allose
Xylose
Ribose
5:55 PM

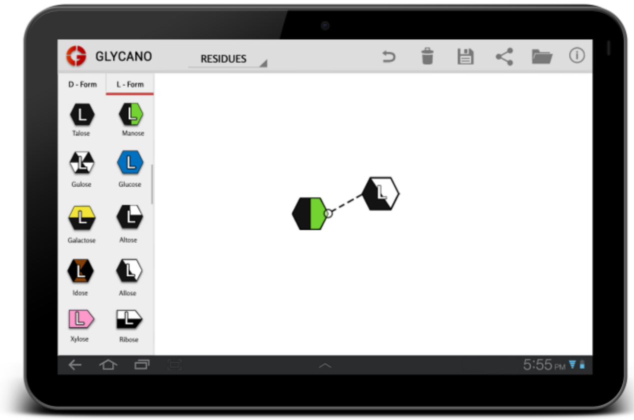
6.


GLYCANO
RESIDUES
D - Form
L - Form
Talose
Manose
Gulose
Glucose
Galactose
Altose
Idose
Allose
Xylose
Ribose
5:55 PM

7.


1 SELECTED
Alpha
Beta
D-Form
L-Form
D - Form
L - Form
Talose
Manose
Gulose
Glucose
Galactose
Altose
Idose
Allose
Xylose
Ribose
5:55 PM

8.


GLYCANO
RESIDUES
D - Form
L - Form
Talose
Manose
Gulose
Glucose
Galactose
Altose
Idose
Allose
Xylose
Ribose
5:55 PM

APPENDIX 5.1

EXPERT EVALUATIONS: BIOGRAPHICAL QUESTIONNAIRE

Android Chemistry Application Heuristic Evaluation

Section A:

Please complete the following:

Name:	
Job Title:	
Years of mobile development experience:	
Highest Level of Education:	

Please rate your familiarity of the following concepts (1 = unfamiliar; 5 = expert):

	Unfamiliar				**Expert**
	1	**2**	**3**	**4**	**5**
Usability heuristics					
Android UI principles					
Android OS navigation and UI patterns (4.0 and higher)					
Human Computer Interaction					

APPENDIX 5.2

EXPERT EVALUATIONS: GLYCANO PRIMER

ANDROID CHEMISTRY APPLICATION HEURISTIC EVALUATION

Section B:

Aims and tasks:

Thesis aim:

To investigate the efficacy of Android UI principles and its interpretations in the usability of a tablet-based scientific application. Comparisons of target end-user perception of usability for each mockup will be compared with expert reviews in refining a final design and assessing UI guidelines in the UI development of an app.

Aim and main functionalities of the application:

Glycano is an Android tablet-based application that allows for the efficient building of carbohydrate molecules. Residues are dragged and dropped onto the canvas and snap onto the closest corner of an already added residue. The form of the residue and bond can be chosen ie. Either D- or L-form residues or alpha or beta bonds. These basically provide more information about the spatial positioning of atoms within the residue and therefore the 3D structure of the carbohydrate itself.

Target population:

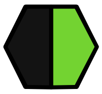
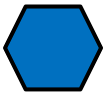
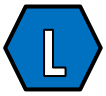
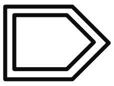
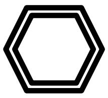
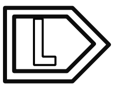
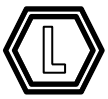
Chemistry students, teachers and scientists wanting to quickly visualise a 2D carbohydrate molecule on a tablet device.

Tasks:

Please review the following app primer and scenarios to aid in your analysis of the application and its mockups. Each mockup was created with an attempt to follow the Android usability guidelines and UI principles (http://developer.android.com/design/get-started/principles.html) as close as possible but with varying navigational structures and minimal personal design. The same scenarios were initially tested on end-users. After or during your analysis, please work through the Excel spreadsheet rating usability heuristics and adding comments and recommendation in the coloured fields for each mockup. Please be specific as possible when commenting and providing examples.

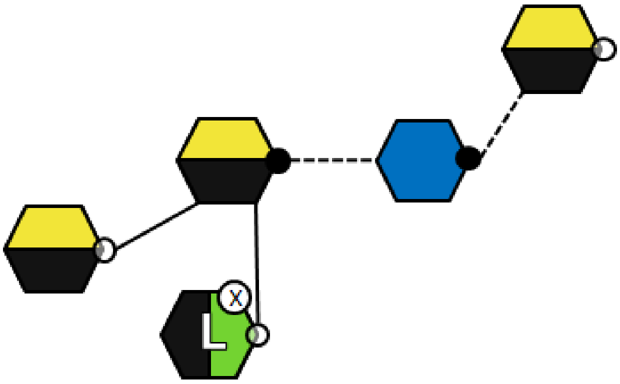
Section C:

Glycano app primer:

Group Name	***Item Examples***	***Description***
Residues (D-form)	 D-Mannose  D-Glucose	Basic sugar building blocks of carbohydrates. D-form denotes right hand structural orientation.
Residues (L-form)	 L-Mannose  L-Glucose	L-form denotes left hand structural orientation.
Ketoses (D-form)	 D-Ribulose  D-Fructose	Basic sugar building blocks that also contain a specific group of atoms. D-form denotes right hand structural orientation.
Ketoses (L-form)	 L-Ribulose  L-Fructose	L-form denotes left hand structural orientation.

Substituents		Sulphur	An atom or group of atoms that replaces a hydrogen atom on a residue.
		Phosphorous	
Bonds	———	Alpha bond	Bonds that join sugar residues together. Alpha bonds cause the joint sugars to lie flat. Beta bonds cause the joint sugars to lie out of plane.
	- - - -	Beta bond	

Example of a 2D carbohydrate structure visualisation:



Section D:

Scenarios: The 3 scenarios and its possibilities show the main functionalities to construct a carbohydrate molecule:

Scenario 1: Search and add:

A chemistry student wants to build a carbohydrate structure. He opens the application and sees the main screen with the blank canvas. He will need to find the residue he wants to add to the canvas and his structure.

In mockup 1 and 2: He searches through the options by swiping either across or up and down, and drags a glucose D-form residue and drops it adjacent to the Glucose residue already on the canvas. This will snap automatically to the closest corner of the Glucose residue.

In mockup 3: He needs to find an L-form Allose residue to start the structure. He looks for it under Residues in the spinner menu and chooses the L-form tab within the side panel.

Scenario 2: Editing residues and bonds:

The student has built a carbohydrate molecule and now wants to edit the form of the residue or bond.

In mockup 1: He wants to change the bond to a beta bond (represented by a dotted line). He long-presses the bond he wants to edit to bring up a popup menu where he can select which option he wants.

In mockup 2 and 3: He wants to change the D-Glucose residue to an L-form residue. He long-presses the residue that he wants to edit to bring up the contextual menu and taps on the L-form icon.

Scenario 3: He now wants to save his work and looks for the save option.

In mockup 1 and 3: He taps the save icon in the action bar.

In mockup 2: He taps the overflow menu followed by the save button.

APPENDIX 5.3

EXPERT EVALUATIONS: USABILITY HEURISTICS AND ANDROID GUIDELINES USED FOR EXPERT HEURISTIC SURVEY

Usability Heuristic	Description
Cognition Support	
Predictability	The user interface must produce results that are in accord with previous commands and states
Learnability	The user interface must be designed for the user to easily learn the app
Memorability	The user interface must be easy for users to remember how to use the application.
Familiarity	The user interface must be familiar to users
Performance Support	
Efficiency	The system should be efficient to use so that once the user has learned the system, a high level of productivity is possible.

Android Design Principles	
Enchant Me	
Delight me in surprising ways	Subtle effects contribute to a feeling of effortlessness and a sense that a powerful force is at hand.
Real objects are more fun than buttons	Allow people to directly touch and manipulate objects in your app. It reduces the cognitive effort needed to perform a task while making it more emotionally satisfying.
Simplify My Life	
Pictures are faster than words	Consider using pictures to explain ideas. They get people's attention and can be much more efficient than words.

Decide for me but let me have the final say	Too many choices and decisions make people unhappy. Just in case you get it wrong, allow for 'undo'.
I should always know where I am	Give people confidence that they know their way around. Make places in your app look distinct and use transitions to show relationships among screens. Provide feedback on tasks in progress.
If it looks the same, it should act the same	Help people discern functional differences by making them visually distinct rather than subtle. Avoid modes, which are places that look similar but act differently on the same input.
Only show what I need when I need it	People get overwhelmed when they see too much at once. Break tasks and information into small, digestible chunks. Hide options that aren't essential at the moment, and teach people as they go.

Make Me Amazing

Give me tricks that work everywhere	People feel great when they figure things out for themselves. Make your app easier to learn by leveraging visual patterns and muscle memory from other Android apps. For example, the swipe gesture may be a good navigational shortcut.
Sprinkle Encouragement	Break complex tasks into smaller steps that can be easily accomplished. Give feedback on actions, even if it's just a subtle glow.
Do the heavy lifting for me	Make novices feel like experts by enabling them to do things they never thought they could. For example, shortcuts that combine multiple photo effects can make amateur photographs look amazing in only a few steps.
Make important things fast	Not all actions are equal. Decide what's most important in your app and make it easy to find and fast to use, like the shutter button in a camera, or the pause button in a music player.

APPENDIX 5.4

EXPERT EVALUATIONS: HEURISTIC SURVEY AND QUESTIONNAIRE

Heuristic Principle	**Design**		
	Severity	Comments	Recommendations
Hover over a guideline for more information. Questions under each heuristic are guidelines and are there to aid in your analysis and understanding of each used heuristic.	Score each heuristic on a scale of 0 -4. Hover over the field to view the severity scale key.	Provide a short rationale for the score, such as a description of the issues found; examples of good practice and the likely impact for users.	Provide means in which problems could be corrected.
1. Cognition Support			
Predictability			
- Are actions taken understandable, clear and logical in predicting what's going to happen next in an interaction? - Are required gestures to navigate headers, add and edit residues intuitive?			
Learnability			
- Is it easy for users to find common items, such as searching			

for a carbohydrate component, without much effort? - Is the interface easy to use from the first time the user interacts with it? Are instructions necessary? - Is it easy and fast to complete specific tasks?			
Memorability			
- Would users be able to return to a level of proficiency in navigating the app and constructing molecules after a period of non-use? - Would the interface be easier to use after each time the user interacts with it? - Would location and layout of functionalities and required gestures be easily remembered?			
Familiarity			
- Are UI elements eg icons, layout and Android patterns familiar? - Does the interface provide enough evidence that it is as an Android application?			

- Would experienced Android users be able to rely on previous knowledge of the OS patterns to navigate the app proficiently?			
2. Performance Support			
Efficiency			
- Do the number of steps required to complete a task decrease efficiency? - Are key tasks made efficient as possible? Quick construction of a carbohydrate should be easy.			
Android Design Principles			
4. Enchant Me			
Delight me in surprising ways			
- Does the design empower the user with key functionalities? - Did textures and design subtlety contribute to a sense of effortlessness? - Do colour themes and styles provide a suitable environment			

conducive for productivity?			
Real objects are more fun than buttons			
- Is a lot of cognitive effort required in creating and manipulating structures?			
5. Simplify My Life			
Pictures are faster than words			
- Are customised icons effective in conveying its function?			
Decide for me but let me have the final say			
- Does the navigation and assembly of residues allows for fast creation of carbohydrates? - Are users given full control of the carbohydrate creation? - Are errors in the construction of the carbohydrate easily repairable? (recoverability)			
I should always know where I am			

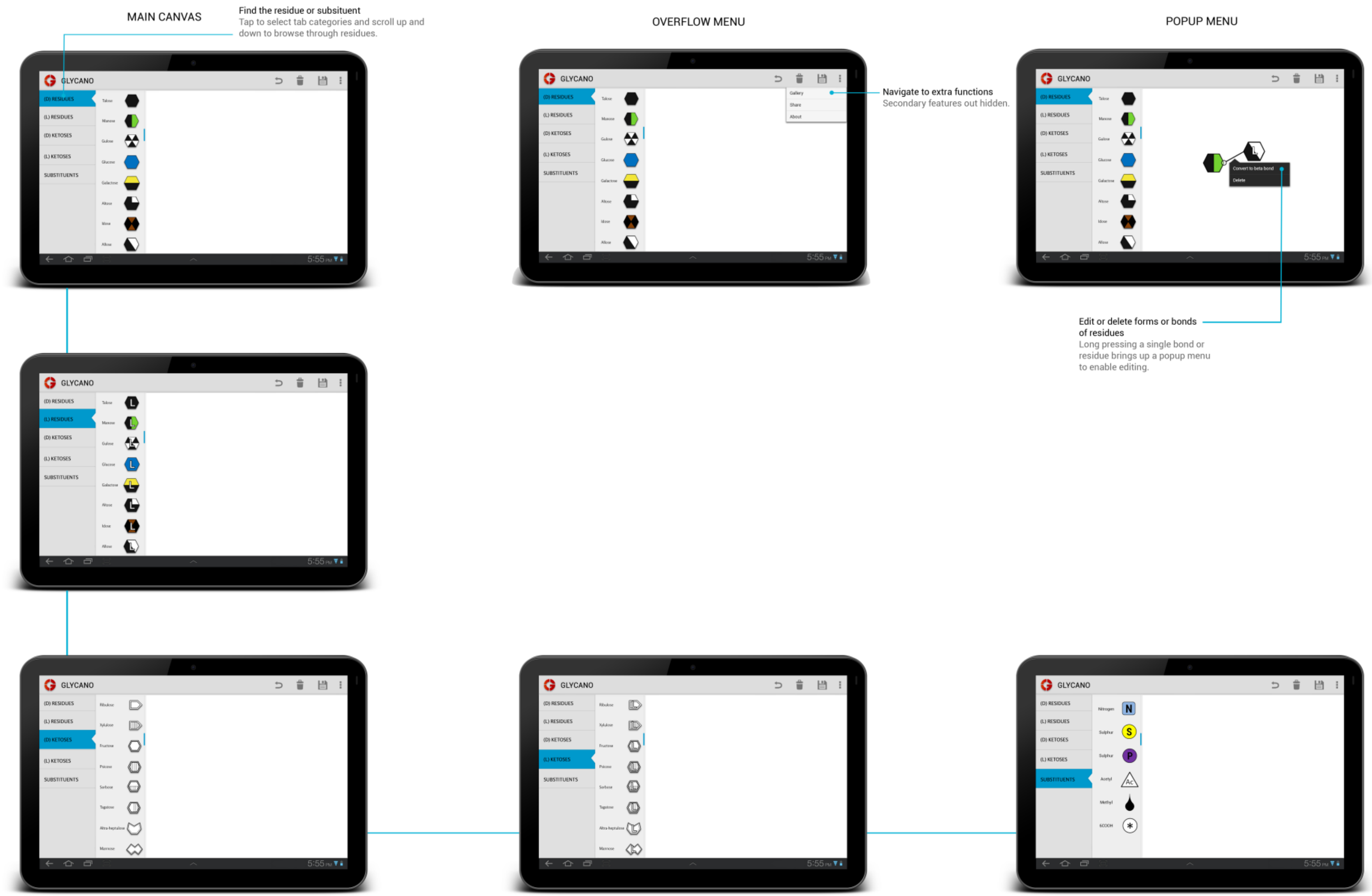
- Is there an obvious hierarchy with relationships between categories? - Are selection and location of menu headings and lists clear? - Are buttons or section labels understandable? - Is information categorised into meaningful groups? Is content placed in places where users expect to find them? - Are common tasks such as searching and adding residues simple?			
If it looks the same, it should act the same			
- Are different functions evident through visually different UI elements? - Do visually similar elements require the same touch gestures? - Are design elements consistent and generalizable? Are like-items displayed and act the same way across the application? - Do elements follow established Android conventions and standards?			
Only show what I need when I need it			
- Do navigational structure and features overwhelm users?			

- Are non-essential items hidden or allow tasks to be kept small? - Is the visual design and layout uncluttered allowing for a clear working space?			
6. Make Me Amazing			
Give me tricks that work everywhere			
- Do key Android visual and UI patterns aid in the learning of the app? - Are touch gestures intuitive when navigating the app? - Are there appropriate UI forms used to give the user the impression that there is extra information			
Sprinkle Encouragement			
- Do feedback on actions inform the user that their interaction has taken effect? - Does the system provide an appropriate level of feedback? - When users select an item, such as a header, residue or bond, is there a clear and proper feedback of selection?			

Do the heavy lifting for me			
- Do shortcuts or certain functionalities accomplish more than the user was hoping for? - Is there minimal effort required to complete a task?			
Make important things fast			
- Are main functionalities prioritized with key actions easy to find and use? - Are main functionalities accessible allowing an efficient building process?			

APPENDIX 6.1

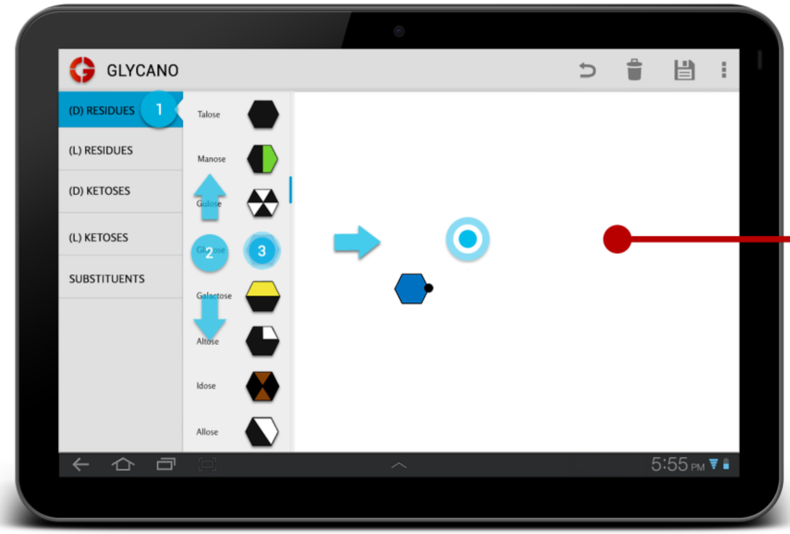
EXPERT EVALUATIONS: PROTOTYPE DESIGN 1 (V)

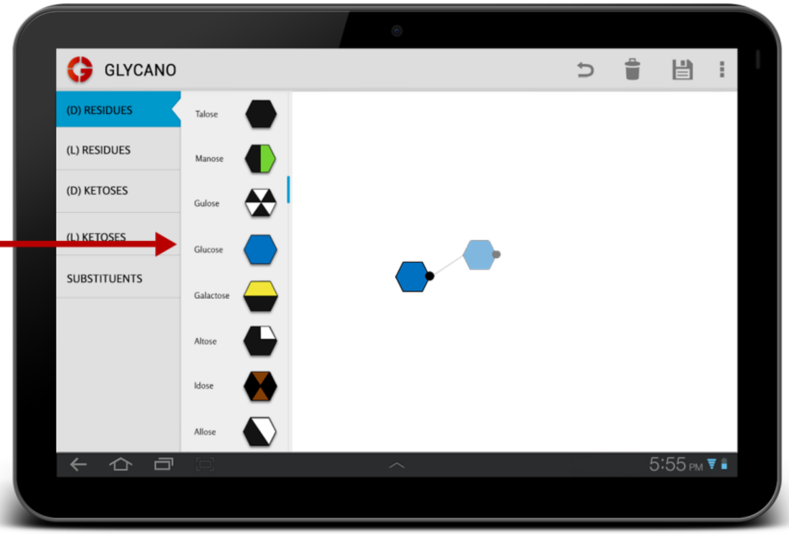

MAIN CANVAS
Find the residue or subsituent
Tap to select tab categories and scroll up and down to browse through residues.
OVERFLOW MENU
Navigate to extra functions
Secondary features out hidden.
POPUP MENU
Edit or delete forms or bonds of residues
Long pressing a single bond or residue brings up a popup menu to enable editing.
GLYCANO

GLYCANO | SEARCH AND ADD

Tap and scroll through residues to search.
Drop and drag residues onto the canvas to place shape.

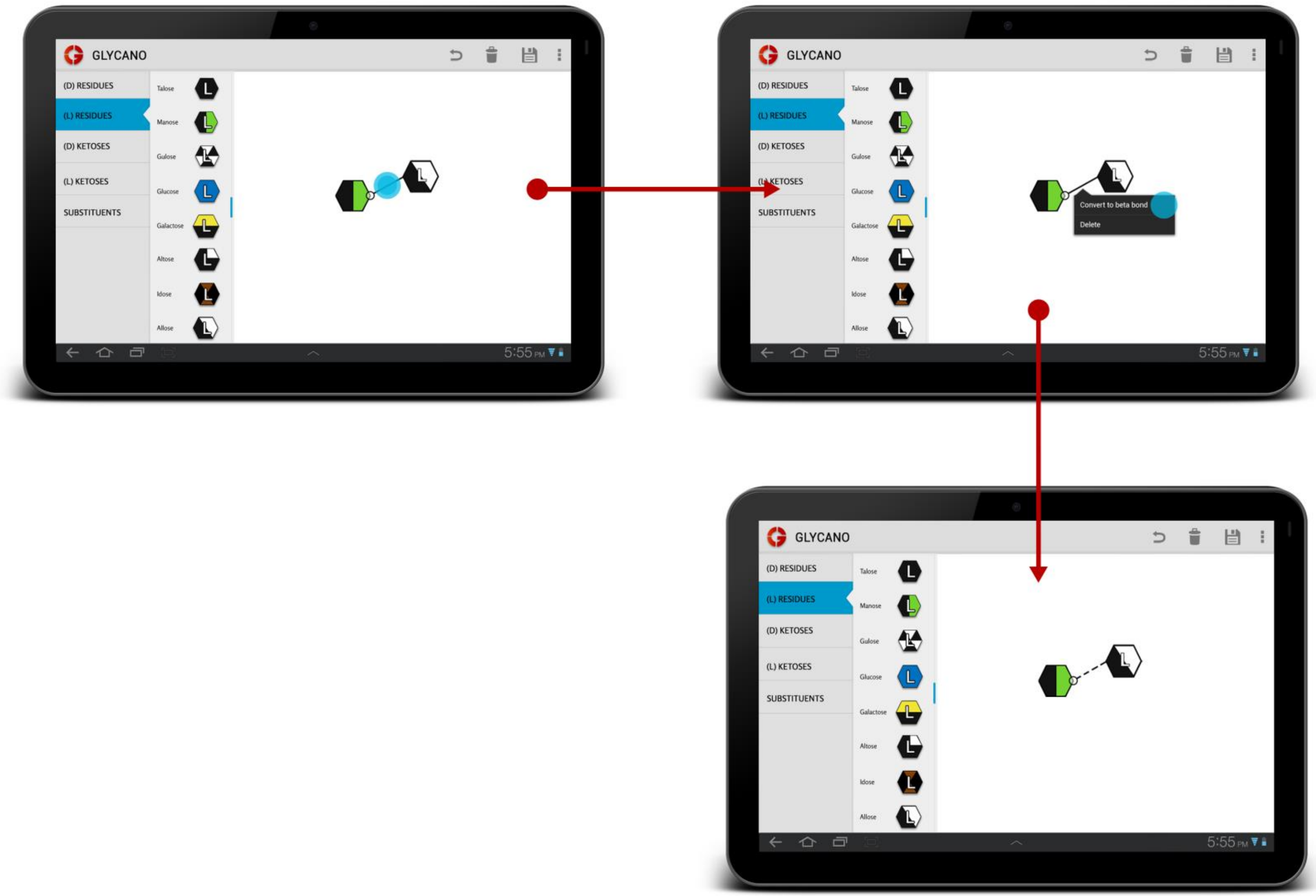
Drop and drag shapes adjacent to placed shapes to join and build the carbohydrate structure.





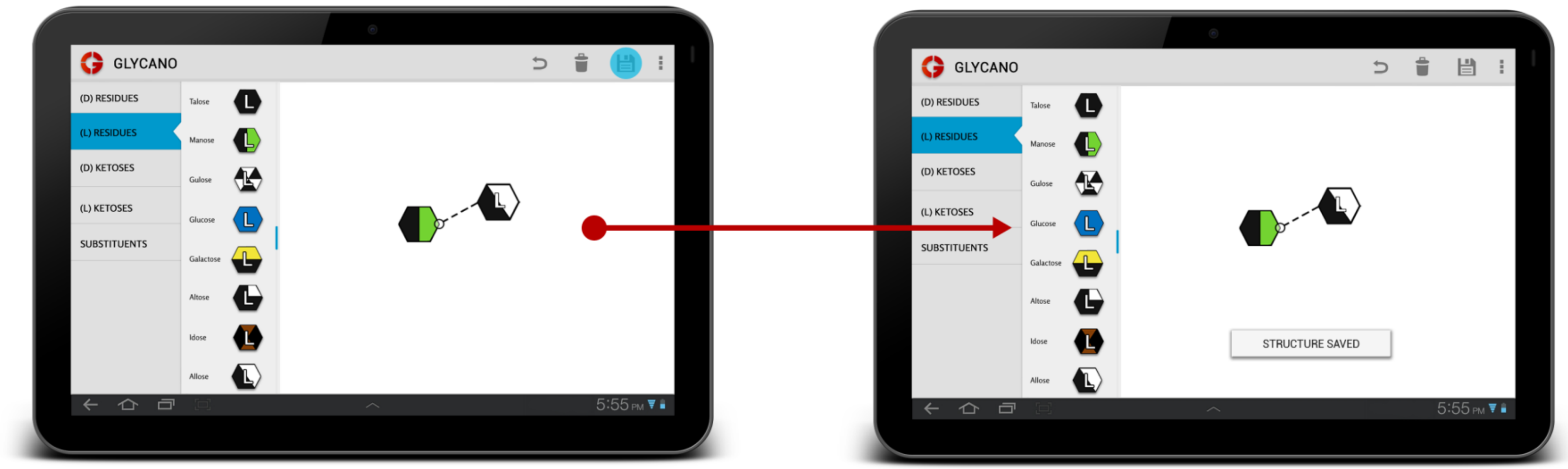
GLYCANO | POPUP MENU

Long press a residue/bond to display popup menu with the applicable options.



GLYCANO | SAVING

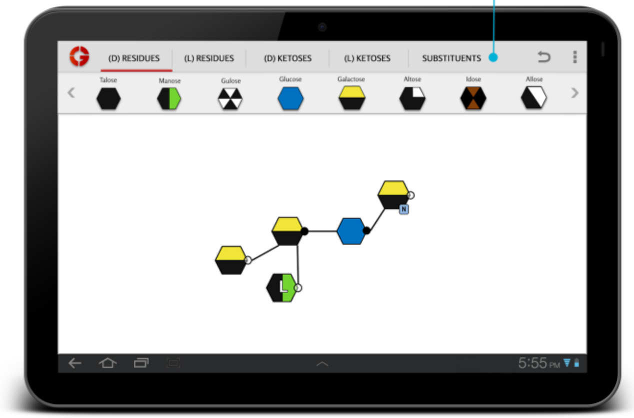
Tap on save icon in the action bar.



APPENDIX 6.2

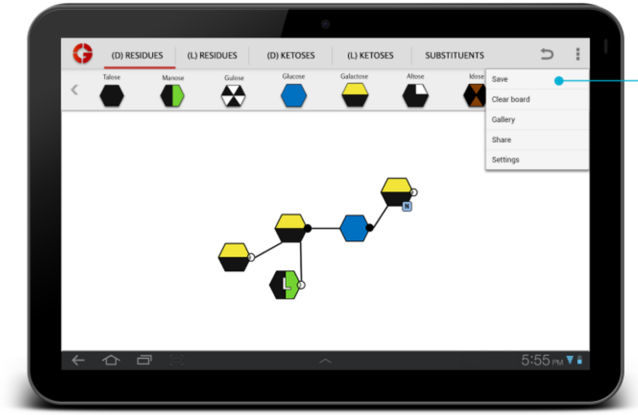
EXPERT EVALUATIONS: PROTOTYPE DESIGN 2 (H)

MAIN CANVAS



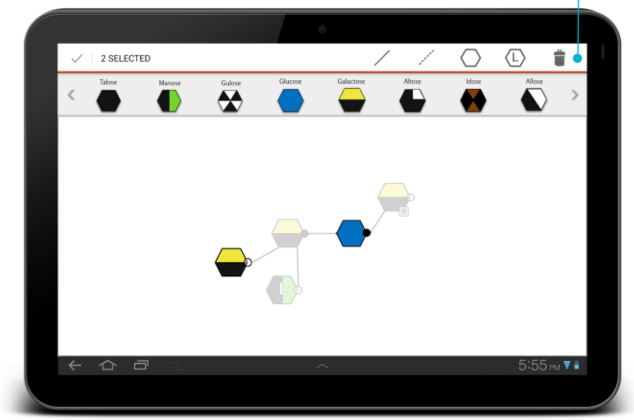
Find the residue or subsituent
Tap to select tab categories and swipe left or right for browsing residues.

OVERFLOW MENU



Navigate to extra functions
Main settings and features out of the way.

CONTEXTUAL MENU

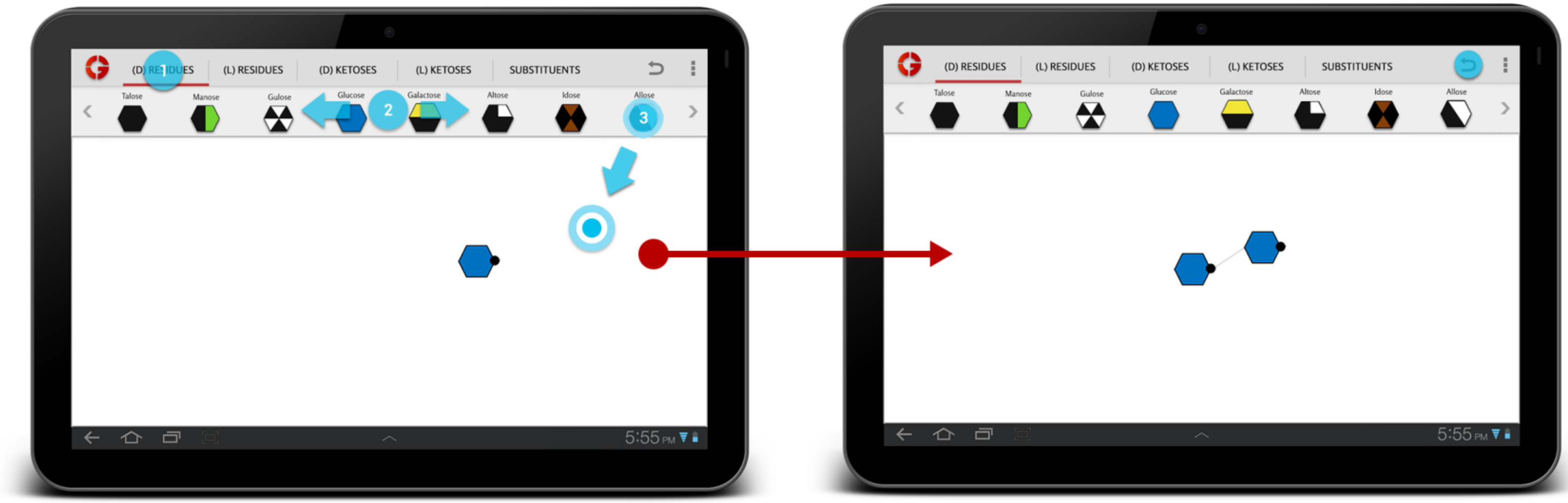


Edit or delete forms or bonds of residues
Selection of residues and bonds appear in bold and are editable.

GLYCANO | SEARCH AND ADD

Tap and scroll through residues to search.
Drop and drag residues onto the canvas to place shape.

Drop and drag shapes adjacent to placed shapes to join and build the carbohydrate structure.

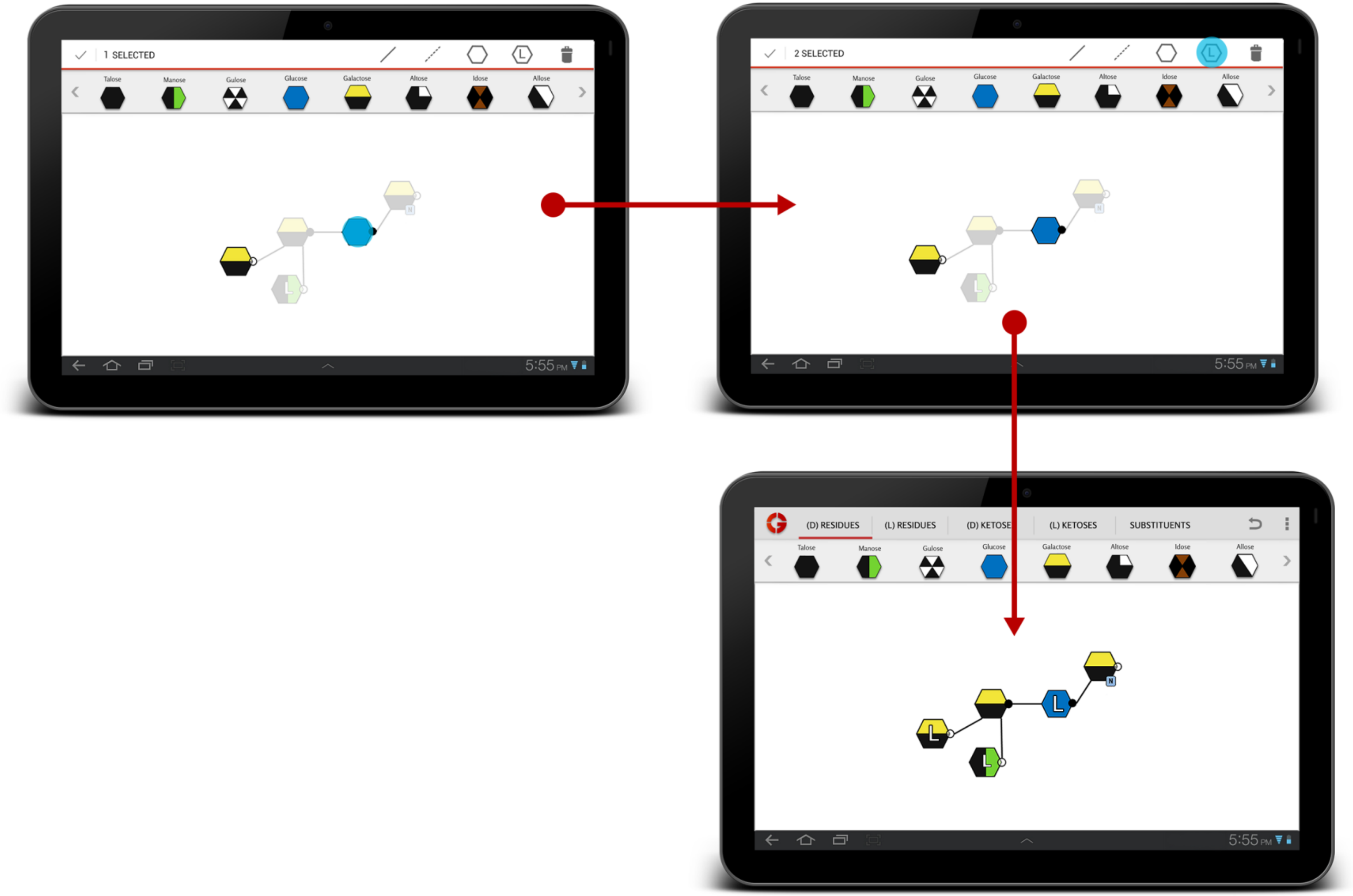


Tap the undo icon to revert back to a previous step

GLYCANO | CONTEXTUAL MENU

Long press a residue/bond to display contextual menu.
Unhighlighted bonds or residues can be selected by tapping them.

Tap applicable icons to change residue form, bond type or delete.

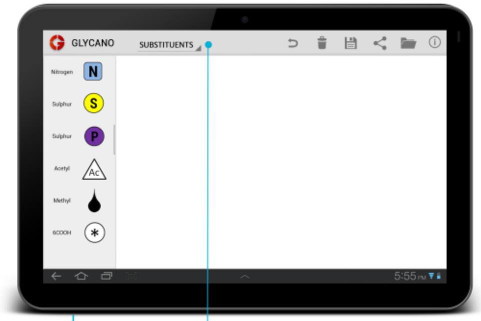


Save feature hidden within the overflow menu.

APPENDIX 6.3

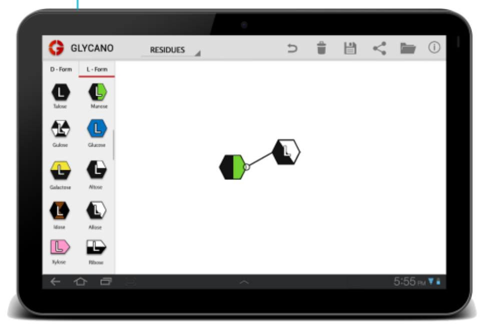
EXPERT EVALUATIONS: PROTOTYPE DESIGN 3 (C)

MAIN CANVAS

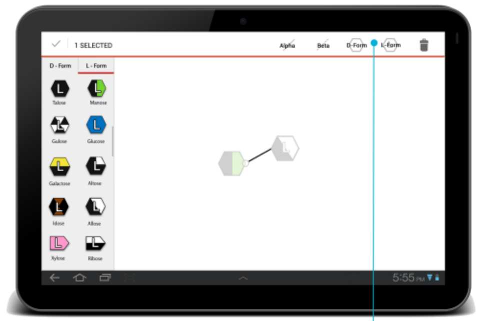


Find the residue or subsituent

Tap to select tab categories from the dropdown menu and swipe up or down for browsing residues in the left pane.



CONTEXTUAL MENU



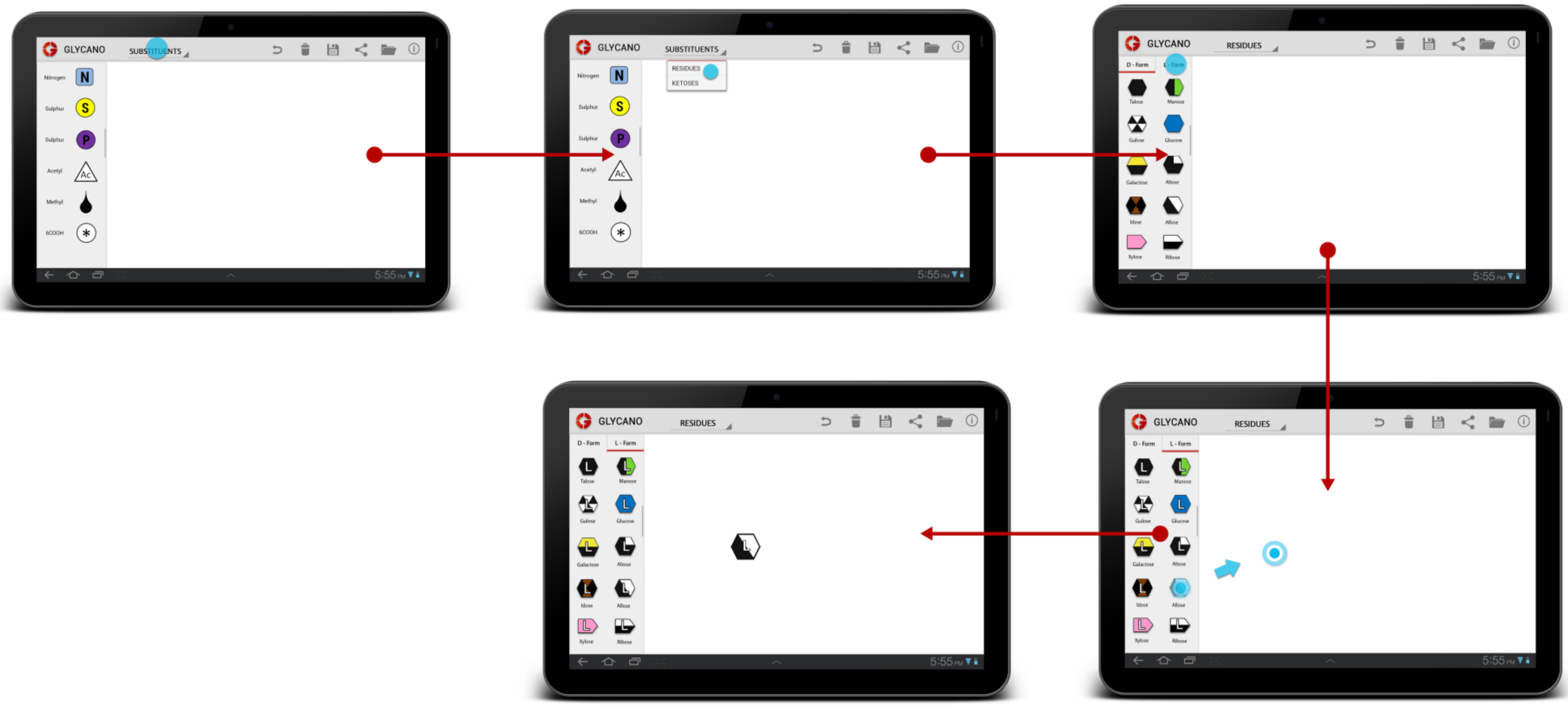
Edit or delete forms or bonds of residues

Selection of residues and bonds appear in bold and are editable.

GLYCANO | SEARCH AND ADD

Tap the dropdown menu to search for residues or substituents.

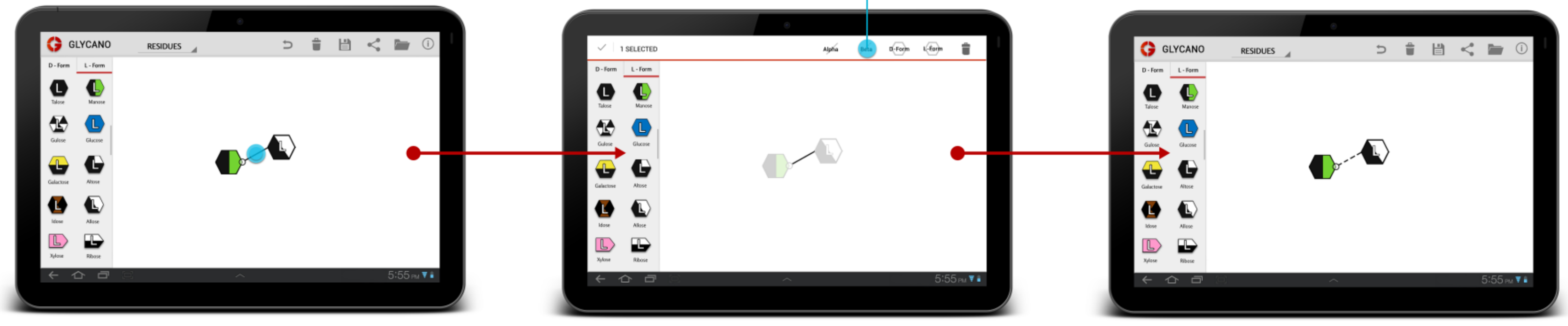
Drop and drag residues onto the canvas to place shape.



GLYCANO | CONTEXTUAL MENU

Long press a residue/bond to display contextual menu.
Unhighlighted bonds or residues can be selected by tapping them.

Tap applicable icons to change residue form, bond type or delete.



GLYCANO | SAVING

Tap on save icon in the action bar.