Group 8 - HI, SM, MA, FH – Tool For All

Hagverdi Ibrahimli
hii22001@student.mdu.se

Samuel Melinder
smr19003@student.mdu.se

Mohammad Alsamhoury
may22001@student.mdu.se

Frederic Himbert
fht22002@student.mdu.se

Keywords

Flutter mobile application, Firebase database, Dart programming language, Json-structured database, Firebase Authentication, Firebase Realtime Database, Firebase Storage.

1. INTRODUCTION

The main idea behind this application, as the name 'Tool for All' suggests, is to facilitate the process of renting tools such as drills, hammers, screwdrivers, and items in similar categories that a typical person might not find around when needed. After registering an account in the application, the user can easily solve this problem by scrolling through the listings of items that other users have published available for rent. To be more concise, the app enables the user to:

- Scroll through many numbers of listings of items other users are lending out,
- Register an account,
- Delete their account and/or change their account credentials,
- Lend out an item by providing the necessary information in the form and capturing/uploading photo(s) of the item,
- Search through large numbers of items,
- Contact the owner of the listing by the ease of redirection to the phone app or via SMS,
- Give feedback on the rented item by writing a short description and rate the item out of a scale of five,
- Tap on the listings to view them in detailed information and scroll horizontally through their images,
- View the located area of the item in google maps, embedded into the application in a form of a small rectangular box,
- View the history of items they have rented, lent out, and feedback of other users.

2. TOOLS USED FOR DEVELOPMENT

The application has been developed using Flutter, which is a cross-platform-oriented programming language developed by Google. This means that the same application can be built for Android, iOS, web browsers, and desktop apps while maintaining the same fundamental code [5]. Since all the group members chose to use this programming language for the labs deployed in the course, there was an agreement to use this language for the project as well. As the app itself is reliant on storing data in a database, a type of Database technology is also important to implement. After a considerable amount of research, the discussion was narrowed down to either employing a SQL-based server or a Firebase server. Since each group member had prior experience using SQL-based servers, the initial idea was to utilize it for the implementation of the project as well. However, after some research, it became evident that connecting a Firebase server to a Flutter application would be more optimal. This is because Flutter has built-in packages for utilizing Firebase and also is a newer technology with a strong demand in the industry [5]. Firebase, which is a collection of services and tools for developing both mobile applications as well as web applications, is also developed by Google. Firebase includes several technologies for creating an application, however, this app utilizes only 3 of them: Firebase Real-time database, Firebase Authentication, and Firebase Storage.

- Firebase Authentication is a tool that is used for managing accounts for different users. In our mobile application, users can register accounts using email and passwords as identifiers for their accounts. It also provides methods for facilitating the user login/logout and keeping track of the current user logged in. This was a fundamental part of the app since it is user-oriented [2].
- Firebase Realtime database is a tool for storing data in a JSON format which highly simplifies the process of reading and writing from/to the database. It has been crucial for our mobile application as we need to store different types of data about user accounts as well as information about reviews, listings, and requests for items [1].
- Firebase storage is a tool for storing different types of files. As a comparison, in the real-time database, everything that is stored is different types of strings, numbers, booleans, etc. Meanwhile in the storage, files such as images and videos can be stored efficiently. In the case of our application, users adding new items to the database should be able to add pictures of the items to the database as well [4].
- Visual studio code is a free, powerful, and widely used code editor for building and debugging programs. It is used to build modern web and cloud applications. It provides the developer with a huge number of extensions to improve the work [7].
- GitHub is a website for programmers/developers to manage and publish their code/projects. It is very suitable for team-work projects such that each member of the team can easily collaborate with others as they can track each other's work and inspect what they have done line by line [6].

3. DESIGN PROCESS

The design procedure of our mobile application is as follows:

1. Paper sketches: we sketched the very first design of the mobile application with just pen and paper. At this step, we have a “Login page”, the first page to be displayed once the application is launched. The “Login page” contains two search fields: username and password along with two elevated buttons: login and register. Once entered the username and password, a single click on a login button would take the user to the “Main page” of the application. If the user has no account registered in the application, the register button would direct the user to the corresponding “Register page”. On the “Register page”, it is possible to view four text fields: username, email, password, and confirm password. Once the user enters the necessary information for all the fields, a click on the elevated confirm button would register the account for the user and redirect the user back to the “Login page”. Once the user is logged in to the application, the “Main page” pops up with all listed items as cards, displaying two cards per each row on the screen. For each card, we have a two-rowed design where the first row contains the image of the item and the second row holds the necessary information about it. Also on the “Main page”, it is possible to view the simple sketch of the bottom navigation bar with three icons placed on top: “Main page” icon, “New item page” icon, and the “User page” icon. In terms of the appearance of the icons, “New item page” icon appears to be floating since it resides in the middle of the navigation bar. When the user clicks on an item card, the “Item details page” will pop up with a back button on the top left. On this page, the image and all the information of the item such as title, published date, description, etc. is shown in a rectangular box. Right below this rectangular box, it is possible to spot another rectangular box showing the located area of the item in google maps. Below the maps, we have located two elevated buttons horizontally: renter and request. Once clicked on the request button, the user is sending a request for the item to the publisher. If clicked on the “renter” button, the “Seller page” appears with the username of the lender and the overall rating represented as stars. At this point, the user can switch between:
    - Listings: The items that belong to the lender.
    - Reviews: The comments previous renters gave to that lender.

   Lastly, the “User page” is identical to the “Lender page” with minor differences such that users can view and edit their account credentials such as username, email, and password (See Figure 1).
2. Uizard website: Following the simple sketch with pen and paper, it was necessary to utilize a modern tool such as Uizard to bring our designs to a professional scope [8]. At this phase of the project, we have the “Welcome page” as the first page to pop up once the application is launched. In the “Welcome page”, the logo of our application along with three buttons are displayed: login, register, and skip. The functionality of the login and register buttons remain the same, however, with the new skip button (a text button) users are now able to navigate to and browse through the “Main page” without logging in (See Figure 2.1). Once on the ”Main page”, it is possible to spot significant changes from the previous version such as the new search field. The main reason behind adding the search field to our application is to facilitate the look-up process of items by their names, especially when the user wants to search among a huge number of listings. Also, the “New item page” icon in the middle of the bottom navigation bar is now at the same level with the two other icons instead of floating to keep the design consistent (See Figure 2.2). In the “New item page”, there are displayed two text fields: title and description. Below the text fields the appropriate button is placed to enable the user to upload the images of the item. Once all the fields are completed, a single click on the elevated add item button publishes the item (See Figure 2.3). In the “User page”, we made certain changes to the design as all the fields such as the username, email, phone number, reviews, my items, and rent history are represented as list tiles with their corresponding icons and descriptions (See Figure 2.4). Since this is a user-specific page, we placed the change text buttons in front of the fields: username, email, and phone number to enable the user to change these credentials whenever needed. Once clicked on the reviews tile, the “Reviews page” pops up where all the reviews that the user has gotten from other users are displayed. The reviews are displayed in the form of a list where each review contains the username, the name of the item rented, and the description for the review. Furthermore, each review displays the number of stars the renter gave to the item, out of a scale of five stars. Similar to the “Main page” of the application, the “Reviews page” also contains a search field on the app bar, along with a back button to navigate back to the “User page”. The reason for the search field to be put into the “Reviews page” as well is, again, to facilitate the look-up process of reviews for specific items (See Figure 2.5). Once clicked on the my items tile, the owner's items are displayed as cards, identical to the “Main page” of the application. Similarly, the rent history tile displays a page where all the items that the user has rented are displayed. Below these list tiles, we have two elevated buttons for the users to change their password and delete their account along with a text button to perform a logout. The design of the renter page is identical to the “User page” with minor differences such that the viewer can not edit their information.

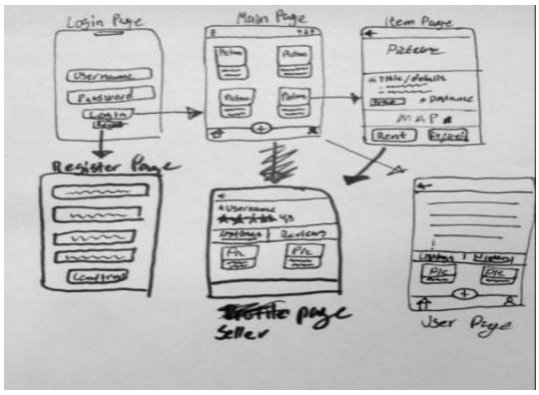

Figure 1. The first design using pen and paper

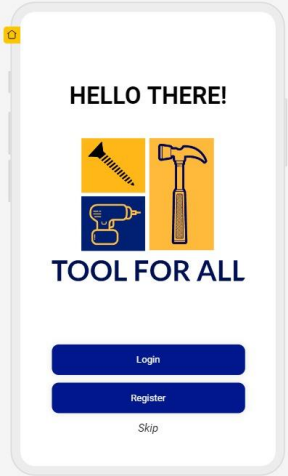

Figure 2.1. Welcome page

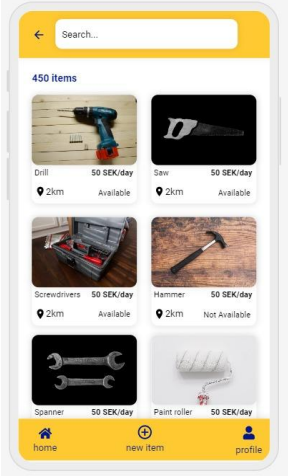

Figure 2.2. Main page

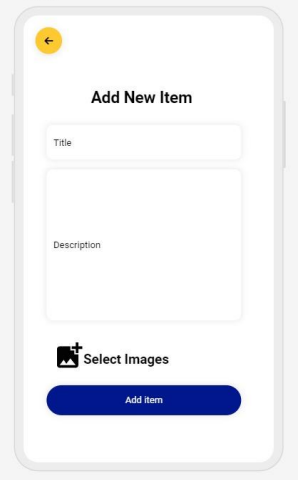

Figure 2.3. New item page

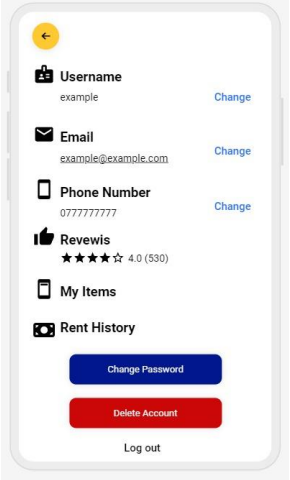

Figure 2.4. User page

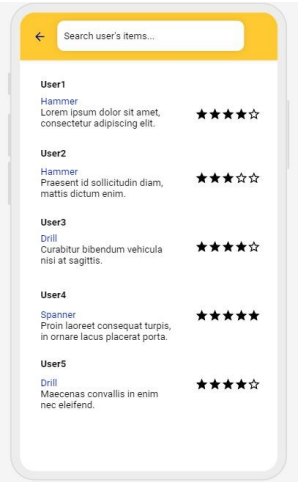

Figure 2.5. Reviews page

4. DEVELOPMENT PROCESS

Throughout the development of this project, our team followed the agile approach. We had loose definitions of what the app would appear as and what functionalities it should encompass from the beginning. Incrementally, we made necessary changes to the design, layout, and inclusion of the functionalities. However, due to heavy time constraints, the overall planning and the design remained rather similar throughout the whole project.

During the beginning of the development phase, we investigated the proper ways to distribute the workload among the group members. For the first week of the project, our main task was to focus on the user interface and complete the overall look of all the pages that our application would contain. Consequently, during the second week, we began implementing the app's core functionalities. At this stage, as a group, we took the decision that each group member should continue working on the same pages which they had completed the user interface of in the previous week. The motive behind this decision was the result of the weekly conducted group meetings as each group member agreed upon the idea that more progress would be possible to achieve in a short period of time if each member continues working on their own pages. After the feedback received during the presentation of the same week, we began having a further discussion about which functionalities to include and/or exclude in order to make our application more usable. One main topic of discussion, which is also included in section 5, was the inclusion or exclusion of the pricing of the tools. The choice was related to if the time would be sufficient to implement prices as we would also need to implement some type of payment methods. Also, the discussion of what type of contact methods users should have for renting tools was another topic. Throughout the whole project, the main constraint has been time. As the project scope of this application was rather great, we had to choose very wisely about which functionalities and pages to develop for the product.

To manage the distribution and state of the tasks, we used the GitHub project backlog for creating and assigning tasks. Although each task was not assigned individually to each GitHub account, one could clearly locate which tasks were finished, in progress. or waiting to be picked by some group member. Whenever a new task was agreed upon, it was added to the “New” section in the backlog. Then whenever this task was picked, the person picking the task moved it from this section to the “In progress” section. This section is for all tasks that are currently under development. Once the development of a specific task is finished, it is moved to the next section “In review” where the other group members are to review and test the feature and try to implement it into their and/or the main branch. If no errors or bugs are found by the group members, the task is finally placed in the “Completed” section.

For version control, GitHub was also utilized. The GitHub repository of the project was structured having one main branch and a separate branch for each group member. As some members lacked experience in working with GitHub, it was a steep learning curve. However, in the end, everyone was rather accustomed to working in the GitHub environment. Commits to the repository were primarily done once a task was completed. However, once the commit is done, the task is again not marked as completed until another team member has been able to test the feature and approve it.

Finally, as for merges to the main branch, merges were primarily done when the focal task was marked as completed. However, if one feature was close to being completed but too far back on the commits available in the main branch, that branch would rather

merge the code on the main branch to their branch. This was primarily done if changes in the main branch would affect the code being developed in the focal branch.

5. USER TESTS AND FEEDBACK

During the progress of our project, we showed our work to the other groups of the course. The most contributive feedback to our mobile application can be categorized as follows:

1. App ergonomics

   As a group, we aimed for an application that would scale very high in terms of the user experience. We received a lot of feedback about the aesthetic of our application. Most of the feedback was concentrated on the issues such as the size of the buttons, margin between the fields, and color choices. The main purpose was to make the user interface both practical and pleasing for the users to a greater extent. As a group, the most important issue for us was to consider the fact that users were more imprecise with their fingers on a mobile application than with a mouse on a computer website. For that specific reason, we spent a considerable amount of time working on the size, margin, and positioning of the widgets.

2. Fixing features

   The second category of the received feedback included the topic of fixing and/or upgrading the existing features. A notable example would be the automatic redirection to the "Main page" of the application after the user has registered an account, instead of displaying the "Login page" to the user again, which would cause frustration for some users apparently. Additionally, there were a lot of comments received about showing the precise location of the item on google maps for a greater user experience. Although a beneficial future, showing the exact location of the item, thus of the lender, was a less secure option as for our group discussions. For that reason, we decided to show the located area of the item rather than the precise location of it to further protect the security of the users.

3. Adding features

   Lastly, a significant number of the comments received were about adding the price tags to the items and handling the transactions through the mobile application itself. However, with the given time constraints, implementing the transactions by protecting the security of the user was a complicated issue from a legal point of view. Moreover, it went against our policy: Create a community to help those in need. Nevertheless, after thorough group discussions, we took the decision to enable the lenders to add price tags to their items. Consequently, it was up to the users to agree upon a payment method by contacting one another.

6. ACHIEVEMENTS

All the functionalities listed in the project goals have been implemented with close attention to code reusability. Additionally, throughout the design and development processes, we managed to add extra features to the application to increase both the quality of the application and user interactivity. Such features include the "Welcome page", rental history, and a list view of the user reviews. Alongside these extra features, enabling the user to have a smooth experience with the application was of great concern. As a result, we tried to reduce the number of clicks as much as possible and added necessary redirections throughout the pages such as:

- Redirect to the "Main page" after the registration,
- Redirect to the "Main page" after publishing a new item,
- Check the login state of the user and redirect to the "Main page" at the app launch, if logged in,
- Prompt the user to login/register whenever trying to publish a new item and/or viewing the "User page", if not logged in.

Although the extra features are beneficial for the user side, achieving these functionalities with a manageable code structure was of high importance. We consider this crucial to the further maintainability of the project, thus a solid groundwork facilitates the process of upgrading the application by adding new features.

7. POSSIBLE IMPROVEMENTS

Given enough time, it is possible to add certain improvements to the application both on the user-friendliness and performance sides. For example, as for the current state of the application, the user is only allowed to register an account by providing their full name and email credentials. However, it is possible to enable the users to register through their already existing google and/or Facebook accounts. This should be a great addition to the user-friendliness of the application as almost everyone has one such account nowadays. Secondly, implementing an in-app chatting system would serve a great value for our application and make it even easier to contact the lender of the item than being redirected to the phone/SMS app. Thirdly, filtering the items by location, categories, prices, availability, and the most reviewed items, this functionality will improve the user experience. Fourthly, adding payment methods like swish, PayPal, and cards such as Visa, Master card, and maestro. Fifthly, it will be great to make the application multilingual. Also, the current state of the application uses the cloud-hosted Firebase Realtime Database which certainly provides enough capabilities for us to manage the data. However, as the application advances to a certain level where the amount of data is getting larger and larger as the number of users grows, it is of highly doubt whether the Realtime Database would make our job easier. Instead, moving the application to the Cloud Firestore, another product of the Firebase company, would prove to bring high efficiency and out-of-the-box tools to manage large amounts of data with ease [3].

8. REFERENCES

[1] Firebase (n.d.). *Firebase Realtime Database*. Retrieved January 10, 2023, from https://firebase.google.com/products/realtime-dat.abase

[2] Firebase (n.d.). *Firebase Authentication*. Retrieved January 10, 2023, from https://firebase.google.com/products/auth

[3] Firebase (n.d.). *Firebase Cloud Firestore*. Retrieved January 10, 2023, from https://firebase.google.com/docs/firestore

[4] Firebase (n.d.). *Firebase Storage*. Retrieved January 10, 2023, from https://firebase.google.com/products/storage

[5] Flutter (n.d.). *Flutter Multi-Platform*. Firebase. Retrieved January 10, 2023, from https://flutter.dev/multi-platform

[6] Kinsta (2022, December 13). *What Is GitHub? A Beginner's Introduction to GitHub*. Retrieved January 10, 2023, from https://kinsta.com/knowledgebase/what-is-github/

[7] Microsoft (n.d.). *Docs*. Visual Studio Code. Retrieved January 10, 2023, from https://code.visualstudio.com/docs

[8] Uizard | App, Web, & UI Design Made Easy | Powered By AI. (n.d.). Retrieved February 1, 2023, from https://uizard.io/