**COURSEWORK SUBMISSION FORM**

|  |  |  |  |
| --- | --- | --- | --- |
| **STUDENT USE** | | **STAFF USE** | |
| Module Name | Mobile Applications Development | First Marker’s  (acts as signature) |  |
| Module Code | 5BUIS012C | Second Marker’s  (acts as signature) |  |
| Lecturer Name | Said Abduvaliev | Agreed Mark |  |
| UoW Student IDs |  |  | |
| WIUT Student IDs | **00009407** |
| Deadline date | 7 December, 2021 |
| Assignment Type | Individual CW  (Report) |

**Words Count:765**

Contents

[Catalogue of Requirements 3](#_Toc89606976)

[Activity Diagram in UML 5](#_Toc89606977)

[Prototyping 6](#_Toc89606978)

[Cycle of Mobile Application Development 12](#_Toc89606979)

[Portability of Applications 13](#_Toc89606980)

[Requirements for the Operating System and Hardware 14](#_Toc89606981)

[Testing in a black box 14](#_Toc89606982)

[Bibliography 16](#_Toc89606983)

# Catalogue of Requirements

If software engineers want to create a long-term successful project, they need invest effort setting particular requirements. Developers and clients will be on the same page if requirements are precise and well-defined. Poorly shown requirements may cause misunderstanding among the team and the client.

***Dictionary mobile application***

Functional Requirements - The functional requirements of an application define what the application should be able to do. The intended behavior of a system is captured by such software requirements. A functional detail should be tied to business requirements to avoid project failure. The following functional parameters should be considered while creating a Dictionary use app:

**Functions of the application that should be expected**

Users will be directed to the **Login Page** after launching the application, where they will enter their username and password and then proceed to the **Home Page**, where they will be able to add new words to the **Words Page** by clicking the Plus Button, which will provide detailed information, and users will also be able to search the word that was previously added. They may access the Navigation Drawer by pressing the Menu Button or swiping to the right from the Home Page.

There will be a Home Button and a Currency Button in the Navigation Drawer, both of which will have a great design with image and text.

Users may check up-to-date exchange rates on the **Currency Page**, which can assist them assess whether they have foreign currency. They can return to the **Home Page** by clicking the Back Button from this page.

Users may add their terms to the **Words Page** with information such as Word, Synonym, Antonym, pick a Word (*from which the word originated*), Note (*comments from other users*), and Date. Otherwise, a system report (*Please fill all forms*) will appear to notify users if they do not provide all of the necessary information.

Users can add the device to their **Home Page** by clicking the Create Button, or they can terminate the process by clicking the Cancel or Back Buttons, which will take them back to the **Home Page**.

Users may add data to the system and CREATE, READ, UPDATED, and DELETE it.

**Requirements that aren't functional** – A non-functional requirement is a requirement that describes how the system performs a certain function. It also outlines how a system should behave as well as its restrictions in terms of functionality. When the development process was completed, the following requirements were met.

**Performance** – When users interact with the app, it runs smoothly; loading panels and features takes less than 2 seconds.

**Scalability** – More data may be added to the program (users can contribute as many words as they wish) without affecting its speed.

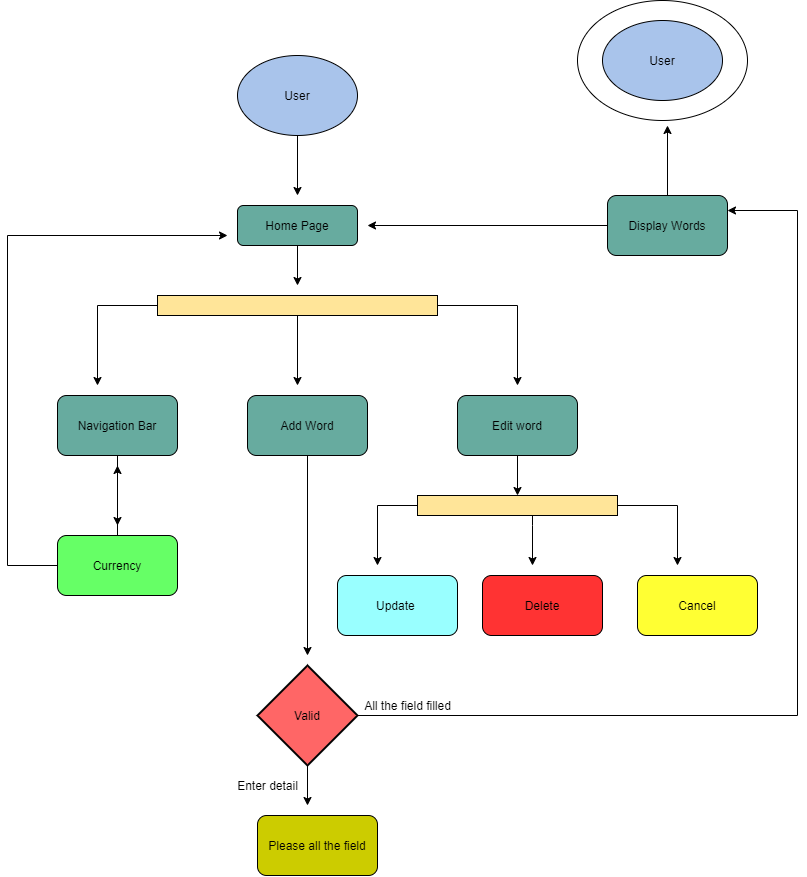
**Responsiveness** - The app will preserve its state if it is interrupted externally by a call or notice.

**Use-ability** – The UX (User Experience) design of the app will be simple and intuitive to use. All buttons will be self-explanatory and will not require any further explanation.

**Availability** **–** The software will be available on the Google Play Store, which is home to over 3 billion mobile devices. This location can also assist developers in receiving more user input.

**Screen Adaption –** When building an app, keep in mind that it will need to adapt to multiple screen sizes. The DP design measuring unit will be utilized in Android Studio to do this.

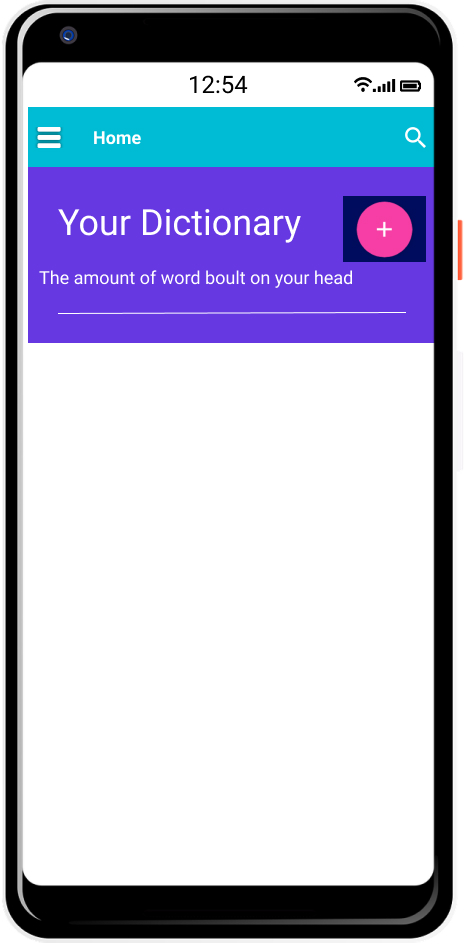
# Activity Diagram in UML



# Prototyping

**Home Page**

<https://www.figma.com/file/ojC2AO9wb85gpr3GPqTwme/Home?node-id=303%3A24>



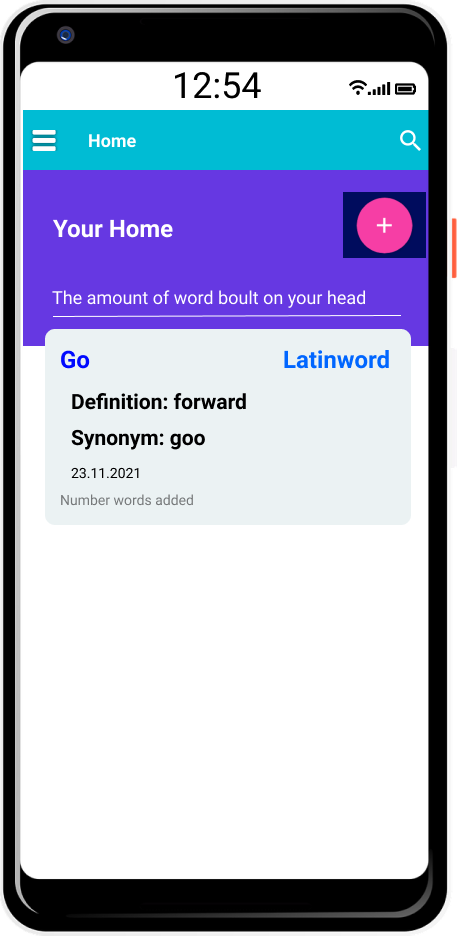
**Add Word Page**

<https://www.figma.com/file/qD37FLaLqcoMVXJPUsaqCZ/Add-new-word?node-id=110%3A36>

****

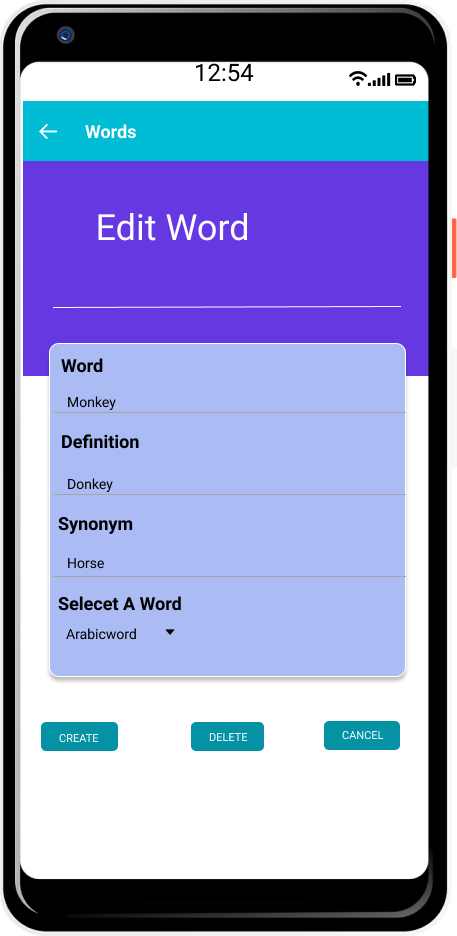
Detailed Page

<https://www.figma.com/file/zTStfm4iitV0IMJMrfxjFC/Home-Page-with--word-added?node-id=3%3A0>



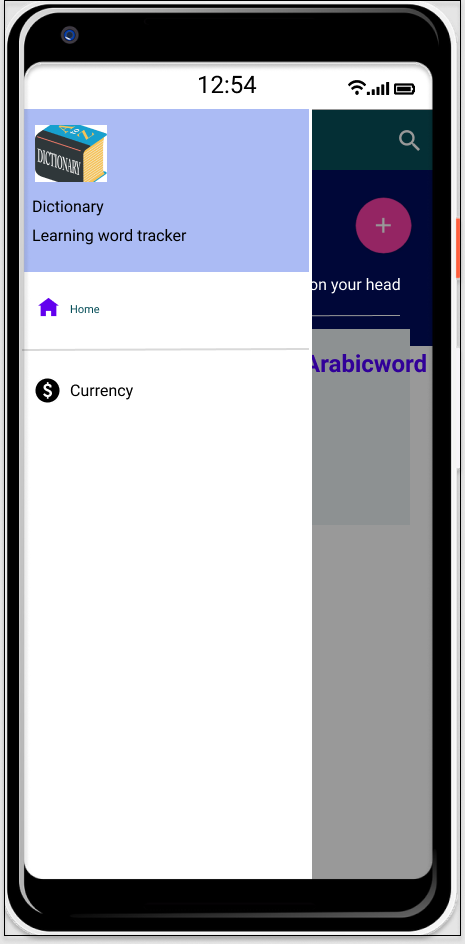
**Edit Word**

<https://www.figma.com/file/aLSb4HA2EPFs0wohAzh30C/Edit-Word?node-id=0%3A1>

****

**Navigation bar**

<https://www.figma.com/file/zgUSt4xicCxvJY7ijMeE0w/Navigation-bar-dictionary?node-id=0%3A1>

****

Currency Page

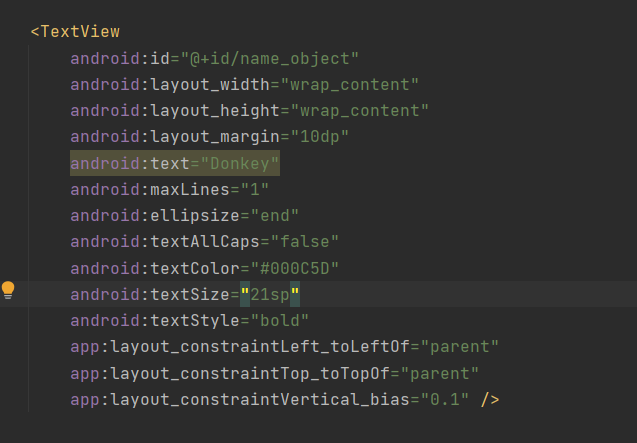
<https://www.figma.com/file/LIB282W7WRP8NkDJID8K5e/Currency-dictionary?node-id=1%3A66>

****

# Cycle of Mobile Application Development

# Portability of Applications

The app was developed for Android devices with API levels ranging from 24 (Android 7.0) to 30 (Android 11.0), covering about 73.7 percent of handsets. Android devices include phones and tablets (Android Things: Phones & Tablets). Furthermore, the program is roughly 4MB in size, which allows users to install and retain the software quickly while without adding to the operating system's workload. Because the program is only partially reliant on the internet and can function without it, it uses less energy. SP – stands for scale-independent pixels used to scale "text sizes," which resizes text based on the user's preferred text size. DP – stands for density independent pixels that allow app design to fit any "resolution of screens," and DP – stands for density independent pixels that allow app design to fit any "resolution of screens." DP and SP employed the following techniques to ensure application portability during the development process:



# Requirements for the Operating System and Hardware

The API levels on the customers' handsets should be 24 (Android 7.0) to 30 (Android 8.0) in order to correctly run the application (Android 11.0). To view the current currency conversion rates, users' cellphones must be connected to the Internet after starting the app. However, the absence of the Internet has no impact on the application's other features.

# Testing in a black box

|  |  |  |  |
| --- | --- | --- | --- |
| Description of the test | Expected outcomes | Actual outcomes | Comments |
| Data from the JSON API is shown. | The software may display data from the Internet about current currency conversion rates. | Web API has been successfully built and is displaying without issues. | Attempt to gather data from the web in the form of terms that will assist consumers in using it... |
| CRUD | The program has the ability to create, read, update, and delete data. | Users may do CRUD activities, as intended. They can use precise information, such as green information, to develop their words. Update the **Home Page** if necessary, and erase the device data from the database. |  |
| Virtual devices were used to test the software: Pixel 4 XL API 24 to 30. | Expected to function correctly... | All functionalities perform perfectly without lagging when tested on virtual devices in Android Studio. | More emulators were used in the testing. |
| Real-world devices were used to test the app:  Samsung Galaxy S8 API 29 (Android 10.0) & Xiaomi Redmi Note 7 API 28 (Android 9.0). | Expected to work properly… | I put it to the test on my smartphone and got positive results. | More real-world gadgets were used in the testing. |
| The time it takes for the system to respond. | It is estimated that the response time will be 0.5 seconds. | The end outcome is more than satisfactory and sooner than planned. | N/A |

# 

# Bibliography

Youtube.com. 2018. *Getting exchange rates with API - part 1*. [online] Available at: <https://www.youtube.com/watch?v=lDEfoSwyYFg> [Accessed 20 November 2021].

Youtube.com. 2020. *Android Navigation Drawer Menu Material Design | Android studio tutorial | Part 1,2,3*. [online] Available at: <https://www.youtube.com/watch?v=HwYENW0RyY4> [Accessed 20 November 2021].

Bhattacharya, S., 2020. *ListView in Android Studio*. [online] Youtube.com. Available at: <https://www.youtube.com/watch?v=NhiUTjm2BrE> [Accessed 20 November 2021].

Youtube.com. 2019. *Android SQLite Tutorial | Android CRUD Tutorial with SQLite (Create, Read, Update, Delete)*. [online] Available at: <https://www.youtube.com/results?search\_query=crud+operations+in+android+studio+using+sqlite+> [Accessed 20 November 2021].