**UpApp Report**

*Gift Langa and Kwazi Thusi*

1. Introduction

This report briefly explains the system architecture and the work done so far. Future plans are detailed in order to guide future developments of the application. This report briefly explains the system architecture and the work done so far. Future plans are detailed in order to guide future developments of the application. The app is aimed at helping all the university community at large mostly first year students or anyone who is not familiar with the campus find their desired destination anytime more conveniently on their phone. This creates a better experience for first years more especially. The app is to be downloaded from Google Store or iStore.

The most important features that have to be in the first release of the app are (mapping of buildings, mapping of different classrooms, locate the user, calculate the shortest distance, emergency contacts). The application will be successful if it can help the user navigate between two places.

1. Backlog

Table 1 shows the backlog of the application; the user stories mentioned are to be included in the MVP (Minimum Viable Product). The Trello board can be join using the link <https://trello.com/invite/b/Fkqdf89g/d0597b4fa3c12da6b8c00dd18281e278/up-app> .

Table 1: The backlog of the front end features and their priority

|  |  |
| --- | --- |
| **User story** | **Priority level** |
| 1. **As a user I must be able to see all types of buildings on campus (Main page)** |  |
| * 1. As a user I want to be able to view lecture halls, laboratories, dining halls, kudu terminals, exam halls, shops, service buildings and residences in their own page so that I can find a building based on its type. | 1 |
| 1. **As a user I must should see the whole map of the main campus (*Main Campus* button)** |  |
| * 1. As a user I want to pan over the map of the whole campus to see the overall area. | 1 |
| 1. **As a user I want to be able to be directed from where I am to the preferred destination** |  |
| * 1. As a user I want to navigate to a particular building just by clicking on it. | 1 |
| * 1. As user I want to be able to search for a building in case I forget the buildings type. | 2 |
| 1. **I want to have favourite buildings to access them faster** |  |
| * 1. As a user I want to add my favourite building on to a list | 3 |
| * 1. As a user I want to delete one of my favourite buildings | 3 |
| 1. **As a user I want to be able to access emergency contacts in case of emergency** | 1 |

1. System Architecture

Figure 1 shows the architecture of the system. The system is divided into presentation, Logic, and database layer. Each layer is explained below. The database is a SQL database. The server side code is written in javascript (Node.js). The presentation layer is developed for mobile devices; currently only the android platform is being developed.

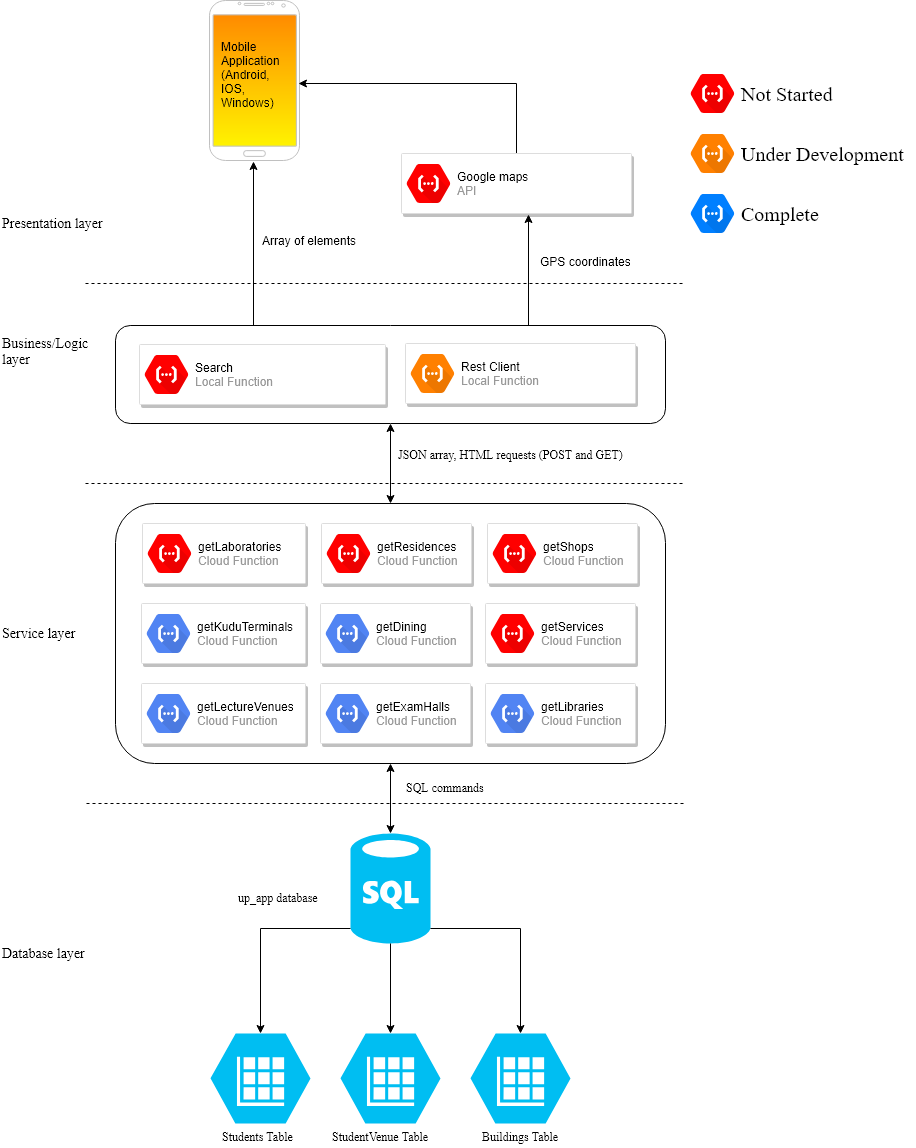


Figure 1: The architecture of the system

* 1. Presentation Layer

The presentation layer can be developed to suit any platform i.e. web, mobile, smart devices. Currently the application is being developed for Android to test the concept. If successful the application will be developed for other platforms. A mock-up of the mobile presentation is presented in <https://www.fluidui.com/editor/live/preview/cF9RUHVwOG40QjhXNDJ5d0Z5RWVadURXRkppSHN4V2lOdg>== .

* 1. Logic/Business Layer

The logic layer is the bridge between the service layer and the presentation layer. Two main functionality are needed, the search bar and a REST client to the server. The spring rest framework is used to implement the rest client. Currently there are problems with connecting to any URL.

* 1. Service Layer

The service layer is a set of functions that execute specific SQL commands. They are executed based on a URL accessed. The detailed documentation of the API is found in <https://documenter.getpostman.com/view/5019248/RzfasBuh> . The service layer is written in JavaScript; “mssql” is used to handle http requests.

* 1. Database Layer

There is one database that holds all the information needed by the web application. Three tables are created; the “Students”, “Building” and “StudentVenue”. The “Students” table holds the first name, last name, student Id (primary key) and password. The “Buildings” table holds the building Id (primary key), building name, abbreviation, longitude and latitude. The abbreviation of the building is the usual name of the building (e.g. Chamber of mines is usually called Chamber), this should make it is easy to query the database. The “StudentVenue” table holds the student ID and building ID, they are both foreign keys; the table was created to simulate the many to many relationship between students and buildings. The database is developed in SQL.

1. Future Developments

The map has to able to help the user navigate between different floors of a building. A stairs icon should appear to signify the need to climb stairs. The map should transition from one floor area to the next depending on the user’s position. The challenge with the feature is that the altitude of the user is not known. A 3D coordinate system should be developed to realise this.

The app should be able to alert the user about the account balance of the Kudu app. This feature will require a login system for students in order to access their information, furthermore, it will create a dependency with the institution’s database.

1. Conclusion

The application needs to be developed further; focus should be put on developing the rest client, the Google maps API and the front end.