**Building an employee information search application for Android devices**

Icon

Description automatically generated

Introduction

This document shows the development of building Employee Information Search app. I will report the initial approach from the beginning stage to the implement stage to state the entire developing process of the given tasks.

In this app I had done implementation part using some of the concepts such as user interface, scrollable list, working on portrait and landscape modes. Then I chose navigation concept to navigate from one fragment to another fragment hosted by an activity. Moreover, to store the data I have used SQLite database concept so it can be possible for performing the CRUD operations. The design pattern in the app follows single-ton design pattern as we share data between multiple objects as part of object-oriented programming paradigm. To achieve this, Mutable live data is used to ensure the data is consistent with multiple fragments within an activity.

Structure

The following four steps provide the process of how our mobile application is made in practice.

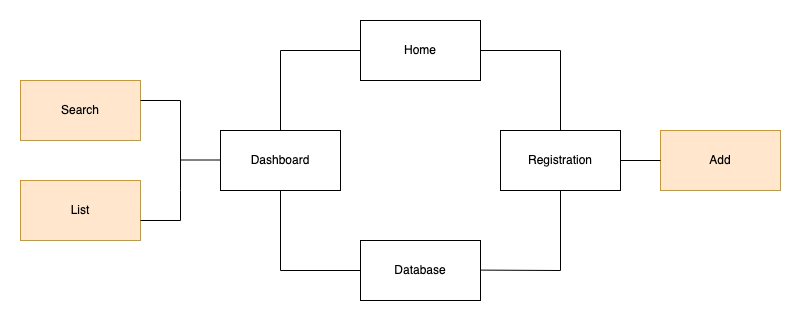
1. Definition and Planning
2. Application Architecture
3. Functional Requirements
4. Reflection
5. **Definition and Planning**

In the brainstorming session, I agreed in consensus about the main concept for mobile application and decided on the critical features and UI design. Then, I started to create draft sketches and descriptions of each screen from the initial idea, after that I could create the structure of the system. Therefore, it will be helpful to imagine how users can interact with each screen while they are using the app. Eventually, I have used the diagram to develop more specific features to improve usability for potential users. Finally, I have focused on basic functional requirements such as registering and searching the stored information using the employee’s name as key.

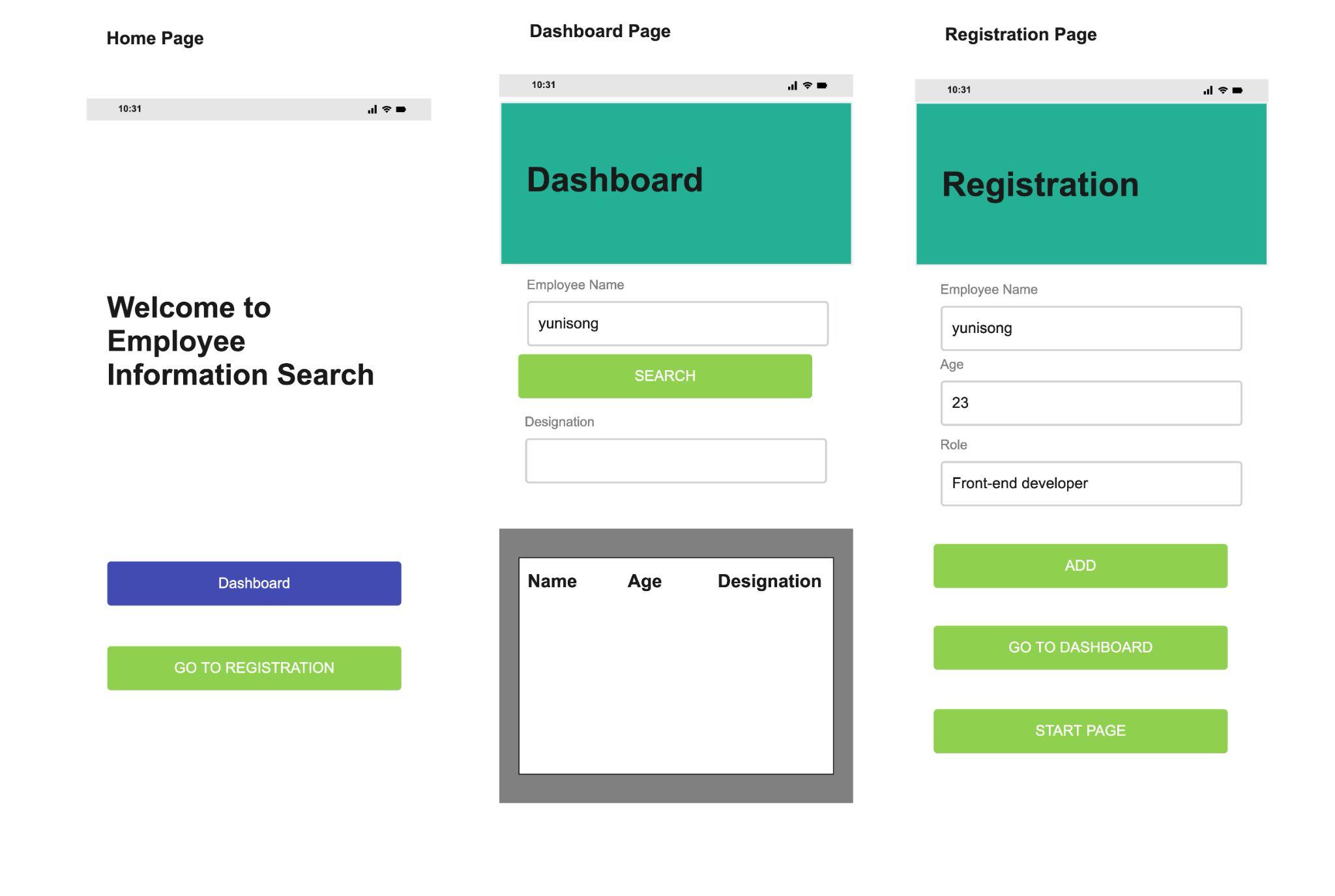
1. **Application Architecture**

The structure of the app architecture was identified to be (Figure 1): Home page, Dashboard page, Registration page, and Database. A mock-up UI design (Figure 2) of the app is made based on the initial architecture diagram (Figure 1).

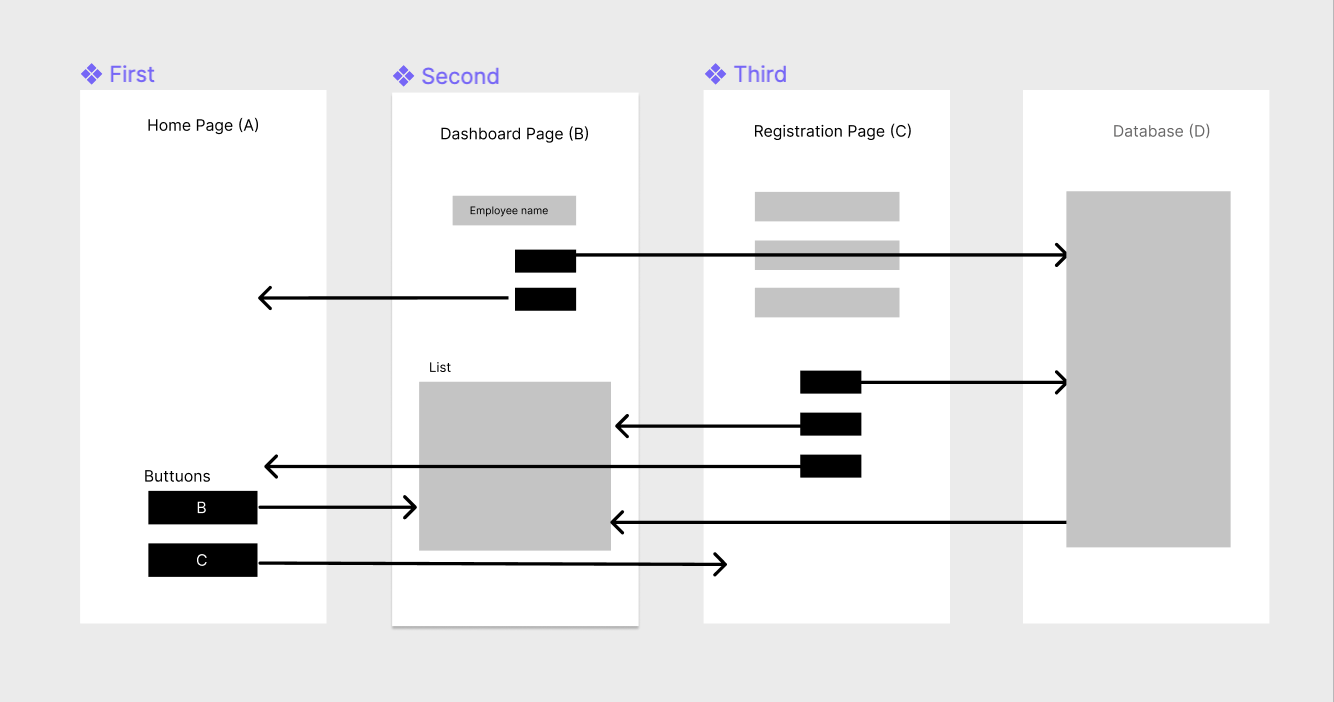
(Figure 3) explains about a clear overflow of our navigation components. More detailed description about the functionality of each page is explained in section 3.



(Figure 1. Diagram for the System Structure)



(Figure 2. A mockup UI design)



(Figure 3. Navigation overflow)

1. **Functional Description**

**Homepage:**

Homepage in our app acts as a starting fragment and it includes the static Textview having welcome messages to the user and two buttons, moving to ‘Dashboard’ and ‘Registration.’ Here I have applied the concept of navigation during the event of onclick button. On click of the first button, the Homepage fragment(‘A’ fragment) will be navigated to the dashboard(‘B’ fragment). In a similar way, on clicking the ‘Registration’ button the homepage fragment(‘A’ fragment) will be navigated to the registration fragment(‘C’ fragment).

**Dashboard:**

Dashboard is the second navigation component. It contains two fragments:

UI fragment and a list fragment. In this component, I applied the concept of mutable live data and the sqlite database for storing the employee records.

The UI fragment consists of two text fields and a search button. Here I are searching the designation of the employee based on the employee name.

The list fragment makes use of the Recycler view as a scrollable list and displays all the data present in the database.

For the deletion of the employee records in the recycler view, it can be done on clicking the specific row.

**Registration:**

Registration fragment is mainly used to add the details of the employee. It has

Three text view fields and three buttons namely add, go to dashboard, and start

page. On clicking dashboard and start page buttons the registration

fragment will be navigated to their respective fragments.

**SqliteDatabase:**

This concept is being used so that I can perform the insert, update, delete

and select operations on the employee records. Since all the records are

stored in the form of tables with rows and columns, cursor will be act as a

pointer to iterate through each record present in the database during the time

of data retrieval. Both the dashboard and registration components are linked

to the SQLite database.

1. **Reflection:**

Once I had completed our development changes, I did a round of unit testing and had a peer review with my fellow mates in my university with regards to code and perform user acceptance testing.