Portfolio Manager

A report submitted for the course named Project - II (CS300)

By

Gooty Saiteja

Semester – VI

Roll No - 19010104

Department of Computer Science and Engineering

Indian Institute of Information Technology Senapati, Manipur

May, 2022

Abstract

I have developed an android application so that, instead of hard coding content onto the live website to display, the data can be fetched from a hosted database, which contains the object model for the data that will be rendered on the site and then I have connected the app to the hosted database so that users can change content from the app.

Declaration

In this submission, I have expressed my idea in my own words, and I have adequately cited and referenced any ideas or words that were taken from another source. I also declare that I adhere to all principles of academic honesty and integrity and that I have not misrepresented or falsified any ideas, data, facts, or sources in this submission. If any violation of the above is made, I understand that the institute may take disciplinary action. Such a violation may also engender disciplinary action from the sources which were not properly cited or permission not taken when needed.

Gooty Saiteja

19010104

Date: 9th May

Acknowledgment

I would like to express my sincere gratitude to several individuals for supporting me throughout my Project. First, I wish to express my sincere gratitude to my supervisor, Dr N. Kishorjit Singh, for his enthusiasm, patience, insightful comments, helpful information, practical advice and unceasing ideas that have helped me tremendously at all times in my project and writing of this thesis. His immense knowledge, profound experience and professional expertise has enabled me to complete this project successfully. Without his support and guidance, this project would not have been possible. I could not have imagined having a better supervisor in my study.

- Gooty Saiteja

Contents

Chapter 1 - Introduction

The Portfolio Manager android application allows the users to easily edit or update the content on their personal portfolio website using their mobile phones. Each time users want to update or edit the content that is being displayed on their websites, rather than making changes to the code, they can simply tap into this Portfolio Manager application and achieve the same. This document is meant to discuss the features of Portfolio Manager, so as to serve as a guide to the developers on one hand and a software validation document for the prospective client on the other.

➢ Outline of the report

This report is organized around four main parts.

Chapter 2 gives specific requirements which the software is expected to deliver. Functional requirements are given by the use case.

Chapter 3 gives an overall description of the software. It gives what level of proficiency is expected of the user, some general constraints while making the software and some assumptions and dependencies that are assumed.

1.1. Gantt chart

Figure 1.1: Portfolio Manager Gantt Chart

Chapter 2 - Requirement Engineering

2.1 Specific Requirements

2.1.1 Functional requirements

- Login authentication: App users need to enter valid credentials in order to access the app.
- Choice: If an element is being updated or deleted, the user must correctly select the element to update or delete.
- Option: Once the element has been chosen, the user should correctly choose the operation that needs to be accomplished.
- Database: In order to prevent corruption of the database, the user needs to ensure that the database is secure.

2.1.2 Non-functional requirements

- Secure validation and profile management facilities for users.
- Authentication and validation are not fully implemented in Portfolio Manager.
- Browsing through the portfolio website to see the changes or updates that are made in each category of edit Content like (edit About, edit skills, edit projects, edit work experience, edit education, edit contact.)
- Adequate indexing strategies to improve efficiency and also to optimize Query Performance that are used to alter the data in the database.
- Maintaining and monitoring the database of user's content that is being displayed on portfolio website.
- Assuring that when application is made to start up it shouldn't take more than 3 seconds to load initial screen and also that app will not hindrance to the user input.

- When the user data increases app should be capable of handling them without delay by optimizing the way storage is done and accessed.
- App should be responsive to the user input which is of higher priority and make changes to the database accordingly.
- User should be able to understand the flow of App easily i.e., users should be able to use App without any guidelines or help from manuals.
- App should be able to render its layout to different screen sizes, along with automatic adjustment of Font size and image rendering.
- In addition to the above-mentioned points, the following are planned to be delivered if seemed necessary:

  When app gets interrupted by call, then app should be able to save state and return to same state/page which was there before it got interrupted.

2.1.3 Performance Requirements

- Assuring that when application is made to start up it shouldn't take more than 3 seconds to load initial screen and also that app will not hindrance to the user input. To display the portfolio website, the app should take no longer than 2 seconds.
- To display the portfolio website, the app should take no longer than 2 seconds.
- App should not take longer than 6 seconds to make changes to the database based on the user's input.

2.1.4 Safety Requirements

- Users have access to the option of logging in and logging out to prevent unauthorized access to the application.

2.1.5 Security Requirements

- App will use secured database.
- An ordinary user can view a website, but cannot edit any content that is being displayed.

2.2 Product Functions

- User: Administrator
- Functions:

  - The Administrator is the super user and has complete control over all the activities that can be performed.
  - Users have access to the option of logging in and logging out to prevent unauthorized access to the application.
  - Among the options that the user has at their disposal, the user is able to update, edit HTML, and edit CSS of the elements that are being rendered on the live website.
  - Users have the option to switch between light and dark themes.
  - The user is also given the option to view the website from the application so he or she can see how the changes are rendered.

2.3 User Characteristics

- The user should understand HTML and CSS so that he or she can update or delete the content that is being rendered on a live website.
- The user should be familiar with the Internet.

2.4 Constraints

- The database should be hosted so that the application can make changes to it at any time.
- Limited to HTTP/HTTPS.
- In addition to updating and deleting the content, the application should also be able to display new elements as they are added to the site. (Not implemented)

2.5 Operating Environment

- Portfolio Manager is compatible with Android versions:
  - Android 7.0-7.1.2, Nougat
  - Android 8.0-8.1, Oreo
  - Android 9.0, Pie
  - Android 10.0
  - Android 11.0
  - Android 12.0

2.6 Software interfaces

- ✓ Android operating system
- ✓ MongoDB
- ✓ Android studio
- ✓ Postman
- ✓ Wire frame cc

2.7 Communication interfaces

- ✓ A connection between the app and mongo dB atlas is established through an API.
- ✓ A Rest API connection is also established between the website and mongo dB atlas.

Chapter 3 - System Analysis, Design & Implementation

3.1 Existing System Study

Often, portfolio websites are hard-coded, so whenever you need to make a change to the live website, you have to clone the code and make the changes manually and then push them again.

An existing system has following disadvantages:

- ✓ As the content on the website is hard-coded, we are unable to change it unless we change its original code.
- ✓ The process of changing the code whenever something on a website needs to be updated is tedious.
- ✓ A user with no programming expertise cannot modify the code to update the live website.

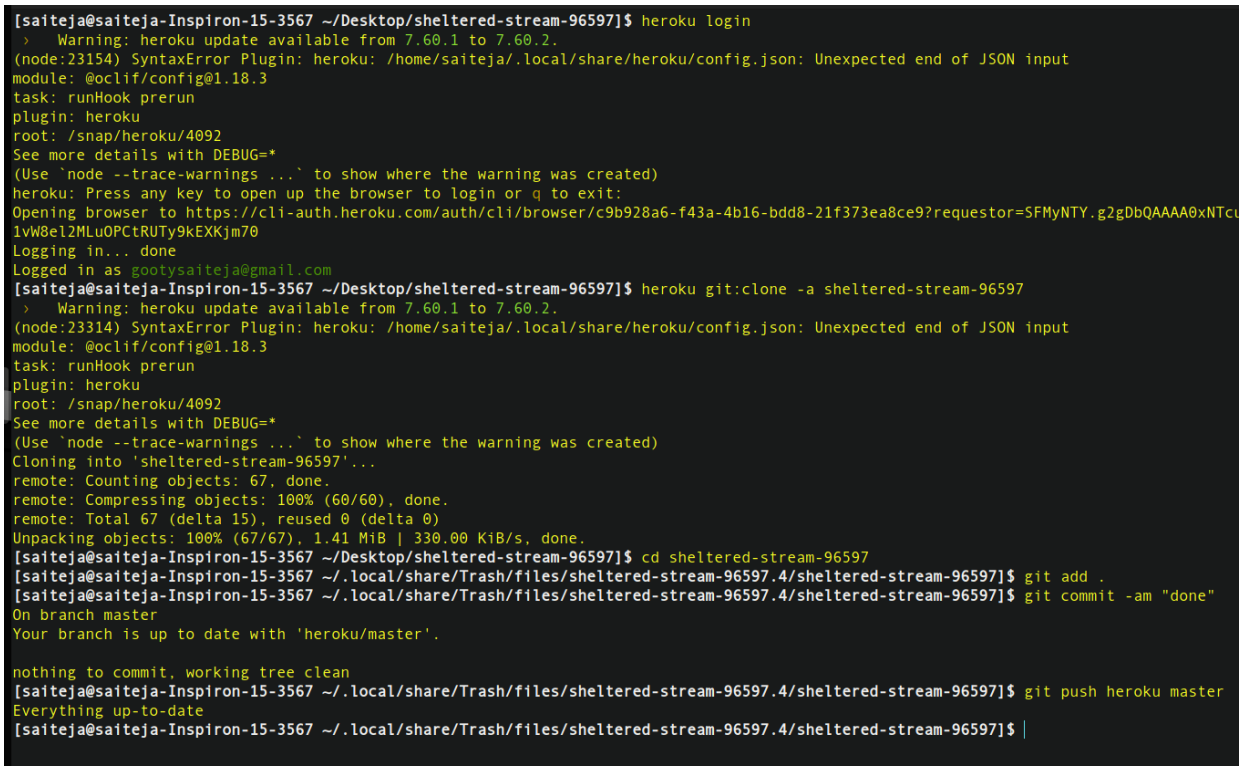


Figure 2.1: Changes to code

3.2 Proposed System

On the real-time website, information is fetched from a database, and it is not hard coded. This allows users to update information anytime by simply updating the database, as opposed to updating original code.

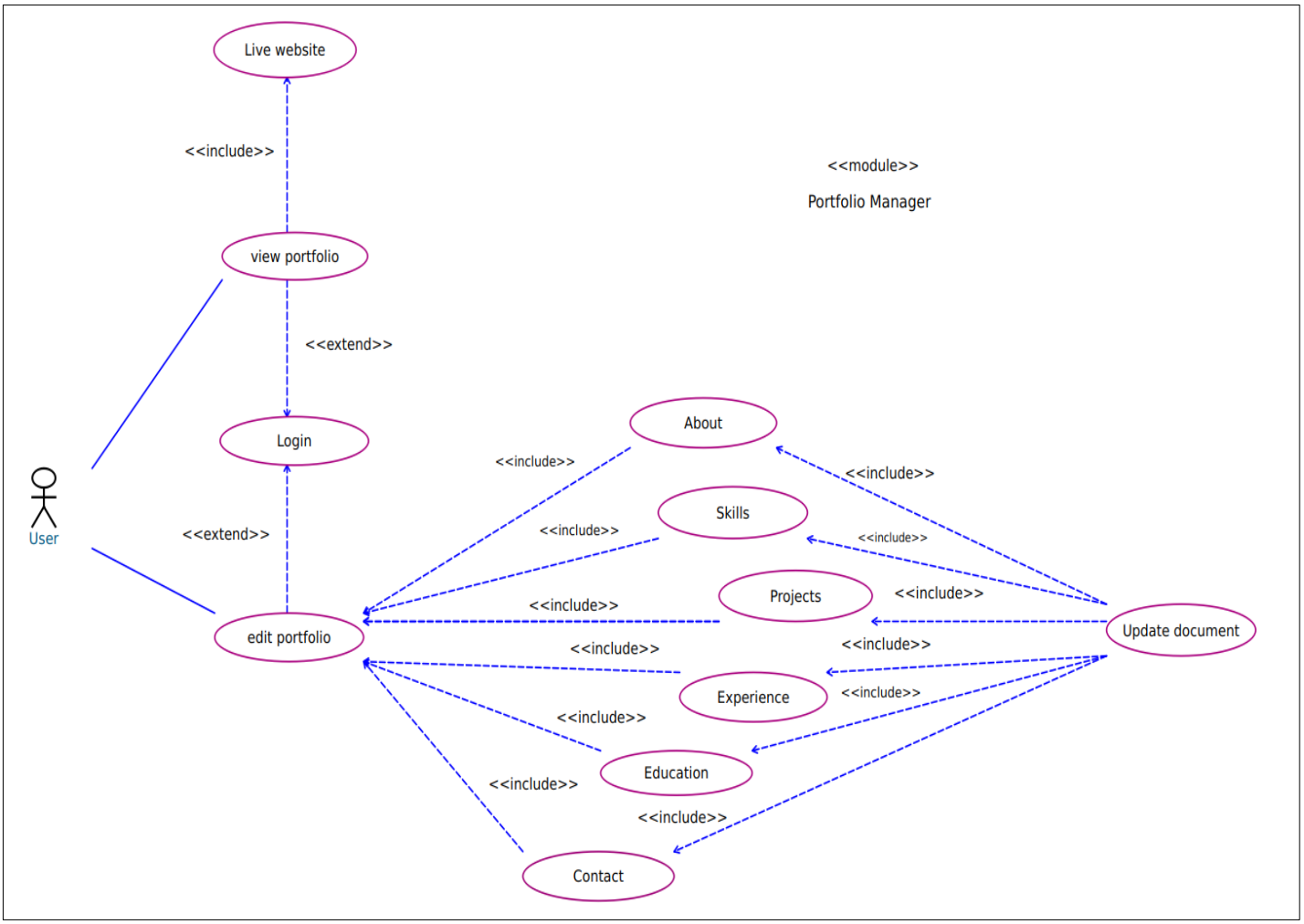
- Information that appears on the live website is pulled from a hosted database, not hard coded.
- If a user needs to update the information, he or she can just access the database rather than changing the original code.
- An individual without programming knowledge can edit the data on a live website by editing the database.

3.2.1 Advantages of the System

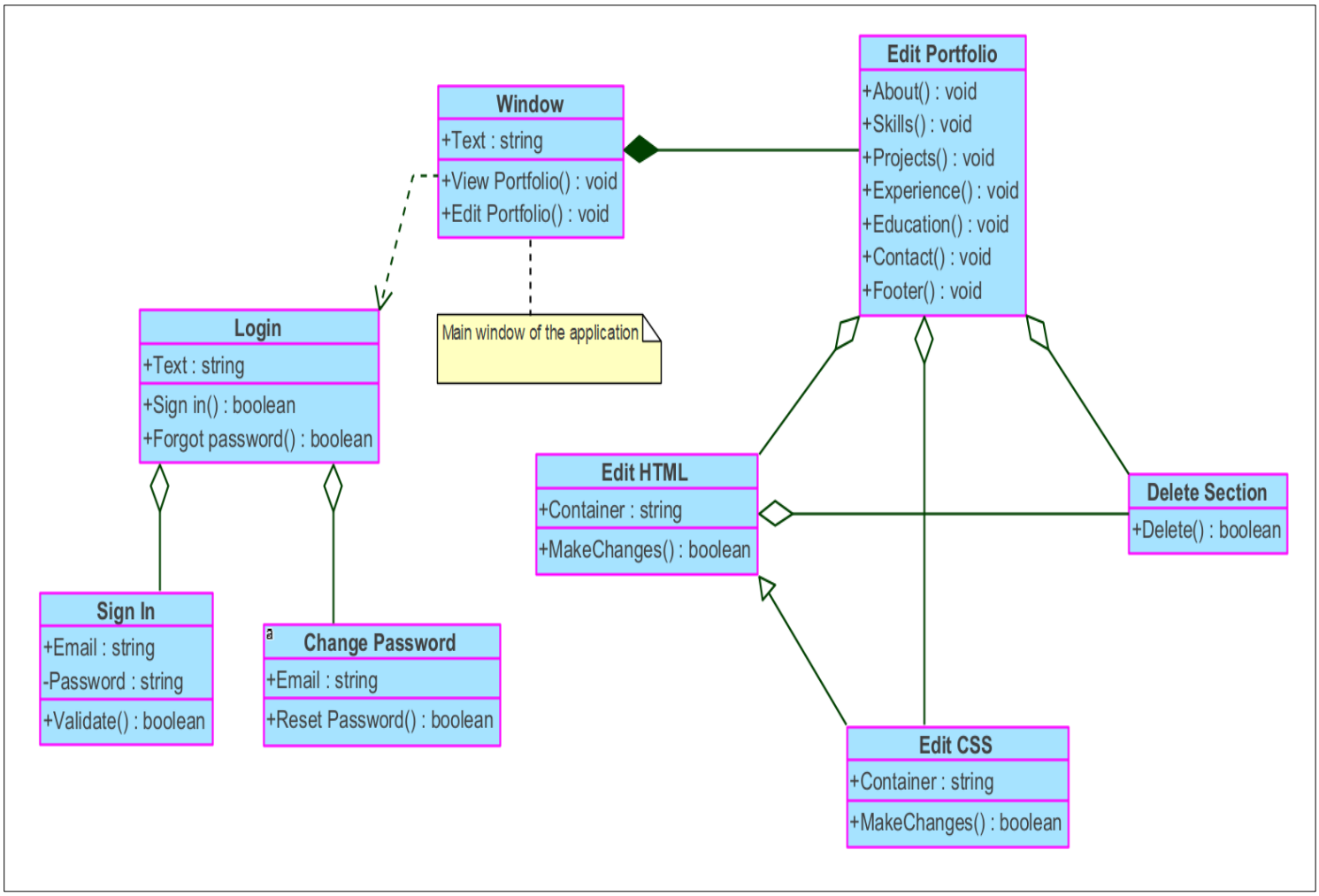
- ✓ Time Saving: By editing the database, a user can quickly modify live website content.
- ✓ Easy to Use: Even people without programming skills can edit a live website's data by editing the database.
- ✓ All in one service: Once the database has been edited, users will be able to see the reflected changes within the app by clicking on the View Portfolio button.

3.3 Overview of the System

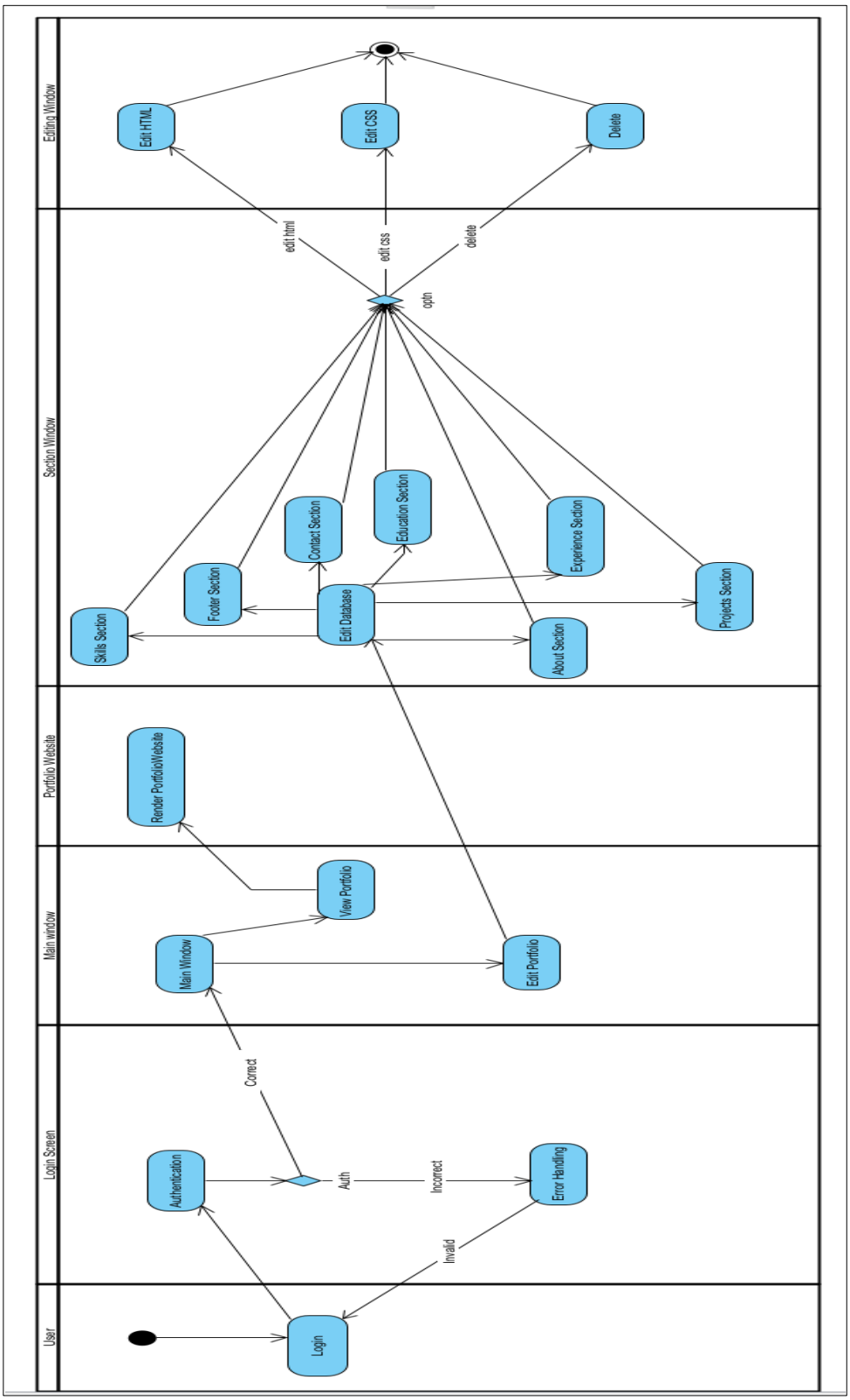
3.3.1 Use Case Diagram



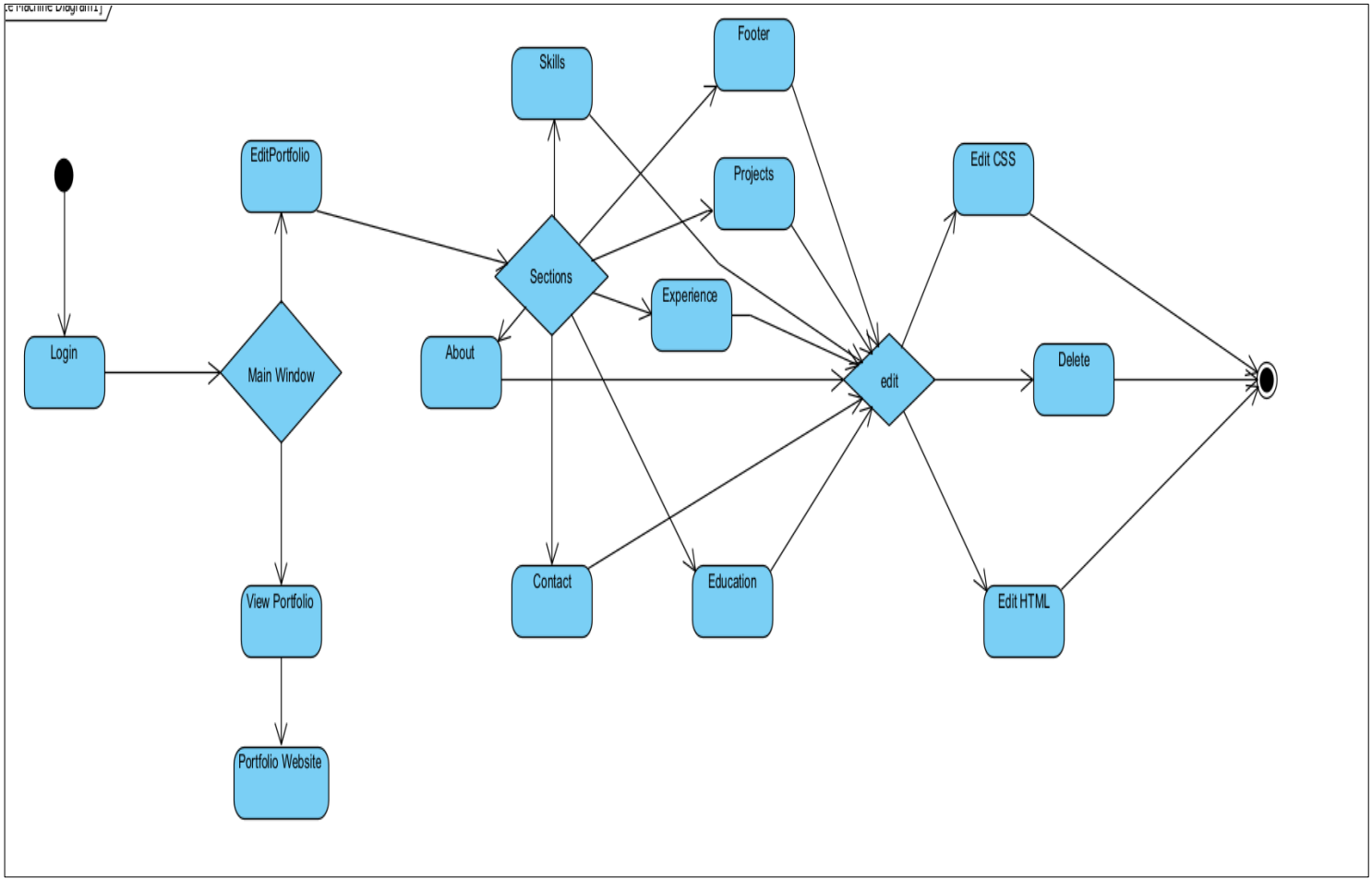
3.3.2 Class Diagram



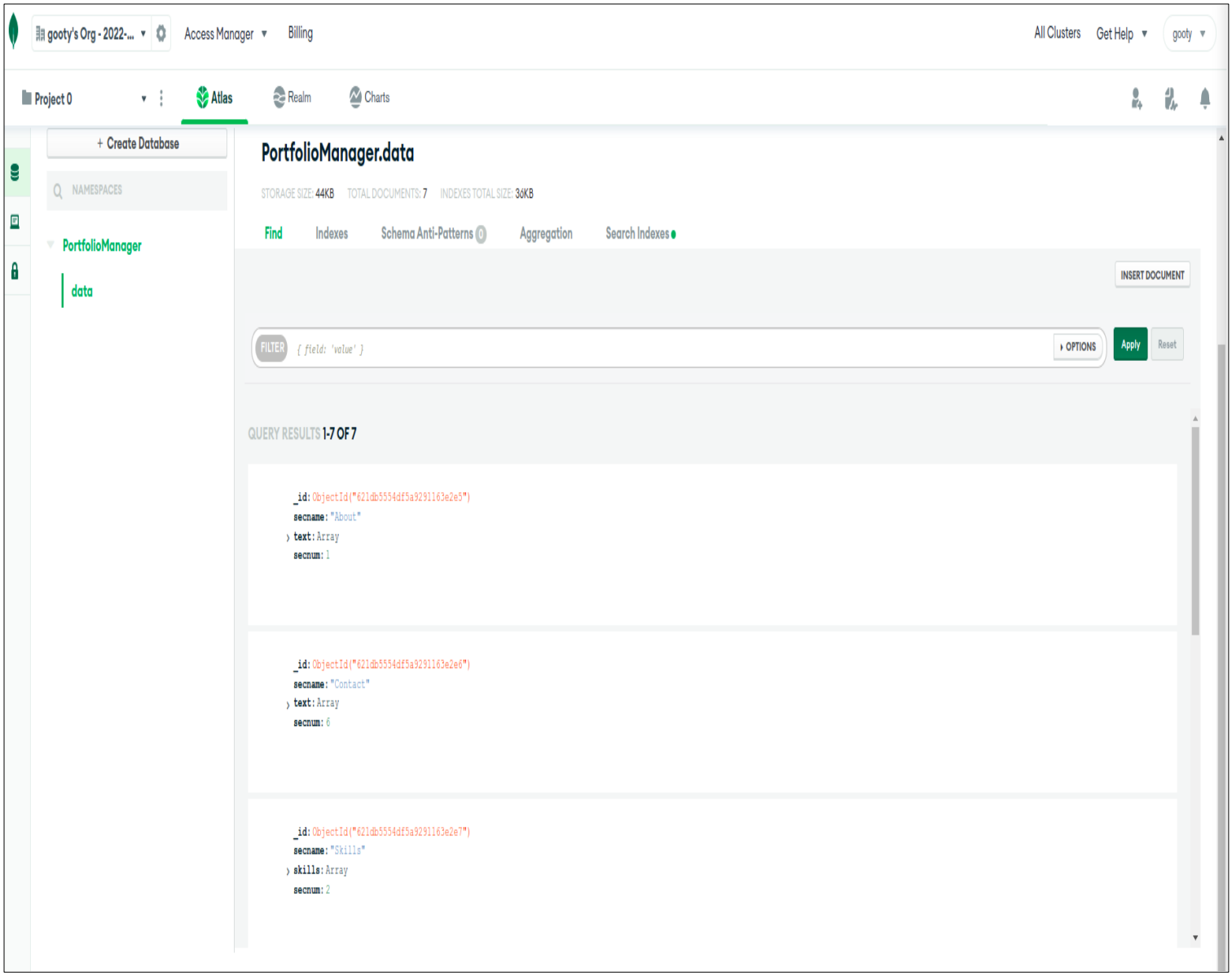
3.3.3 Activity Diagram

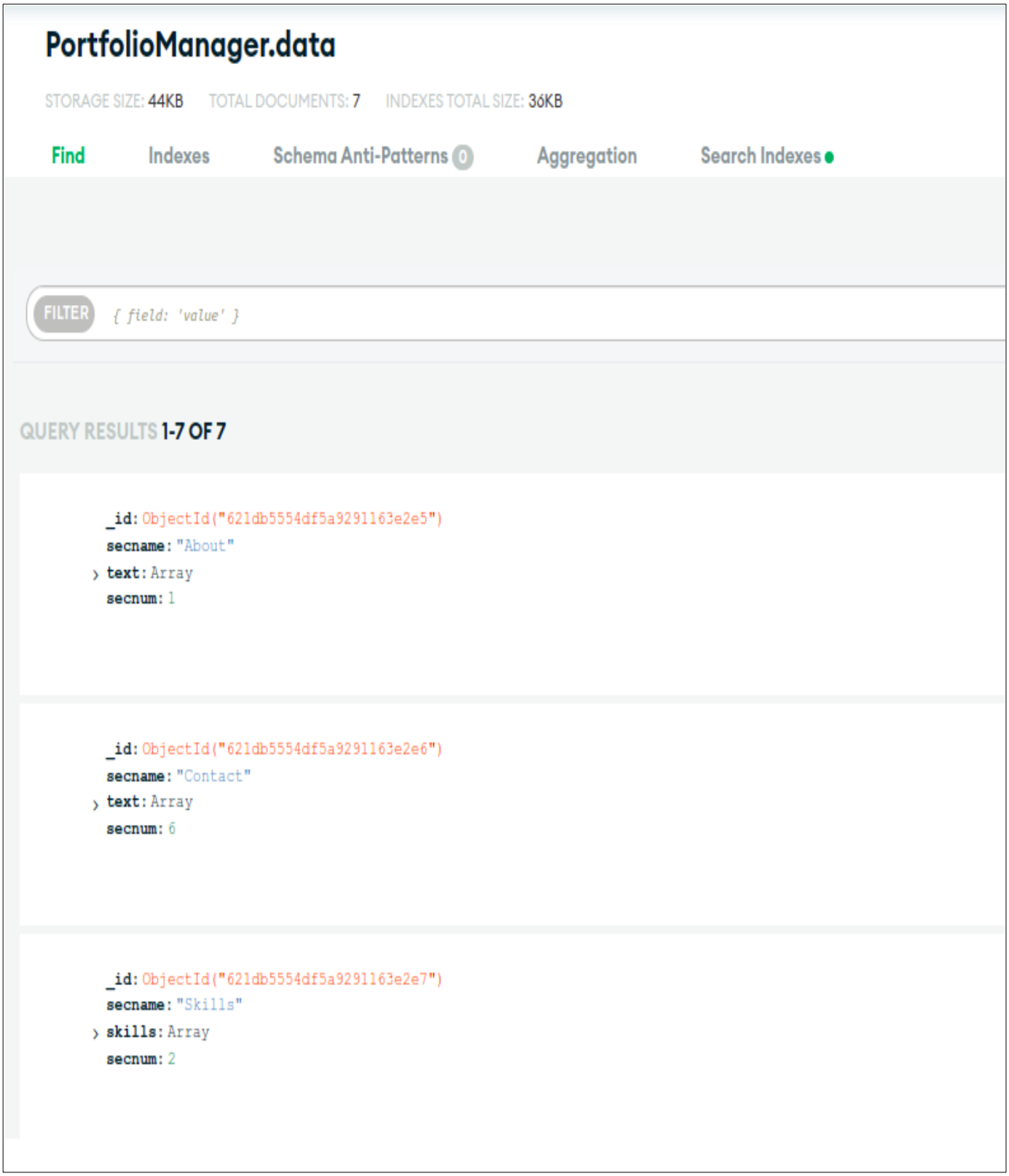


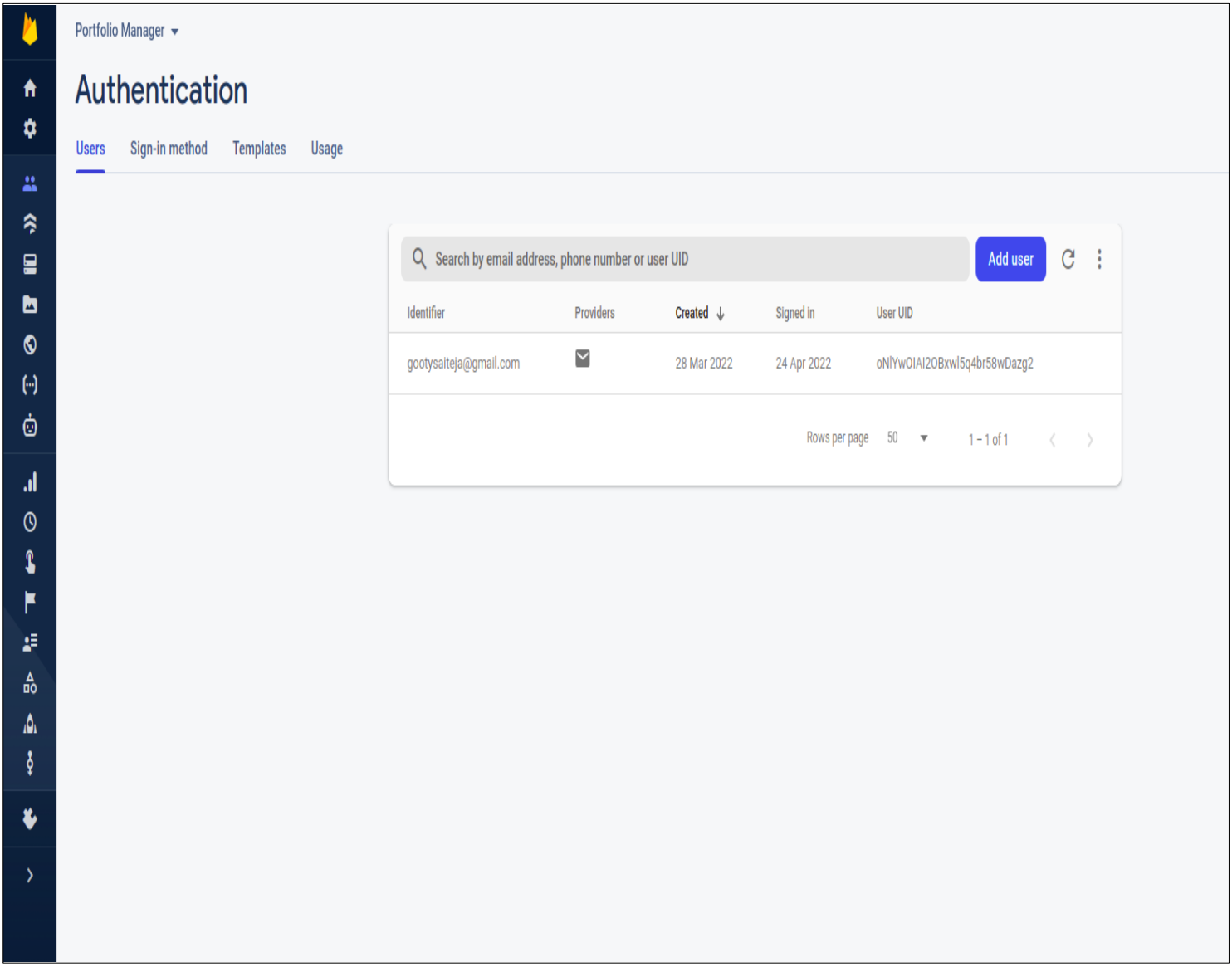
3.3.4 State Diagram



3.3.5 Databases Structure



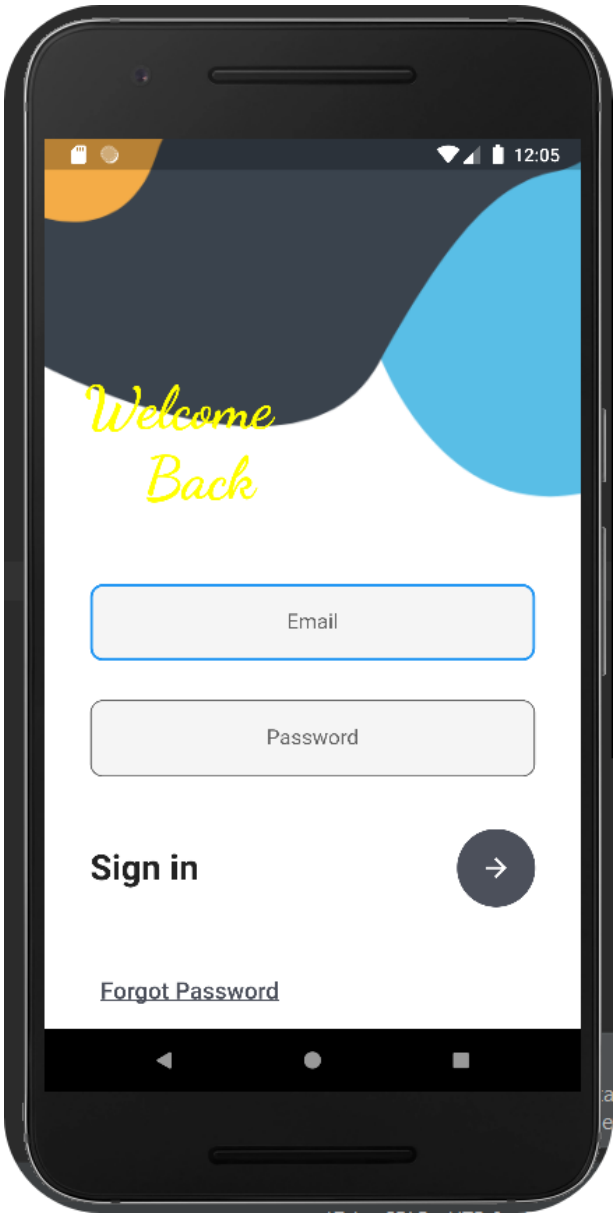

PortfolioManager.data
STORAGE SIZE: 44KB
TOTAL DOCUMENTS: 7
INDEXES TOTAL SIZE: 36KB
Find
Indexes
Schema Anti-Patterns 0
Aggregation
Search Indexes
FILTER
{ field: 'value' }
QUERY RESULTS 1-7 OF 7
_id: ObjectId("621db5554df5a9291163e2e5")
secname: "About"
text: Array
secnum: 1
_id: ObjectId("621db5554df5a9291163e2e6")
secname: "Contact"
text: Array
secnum: 6
_id: ObjectId("621db5554df5a9291163e2e7")
secname: "Skills"
skills: Array
secnum: 2


Portfolio Manager
Authentication
Users
Sign-in method
Templates
Usage
Search by email address, phone number or user UID
Add user
Identifier
Providers
Created
Signed in
User UID
gootysaiteja@gmail.com
28 Mar 2022
24 Apr 2022
oNlYwOIAI2OBxwl5q4br58wDazg2
Rows per page
50
1 – 1 of 1

3.4 Design

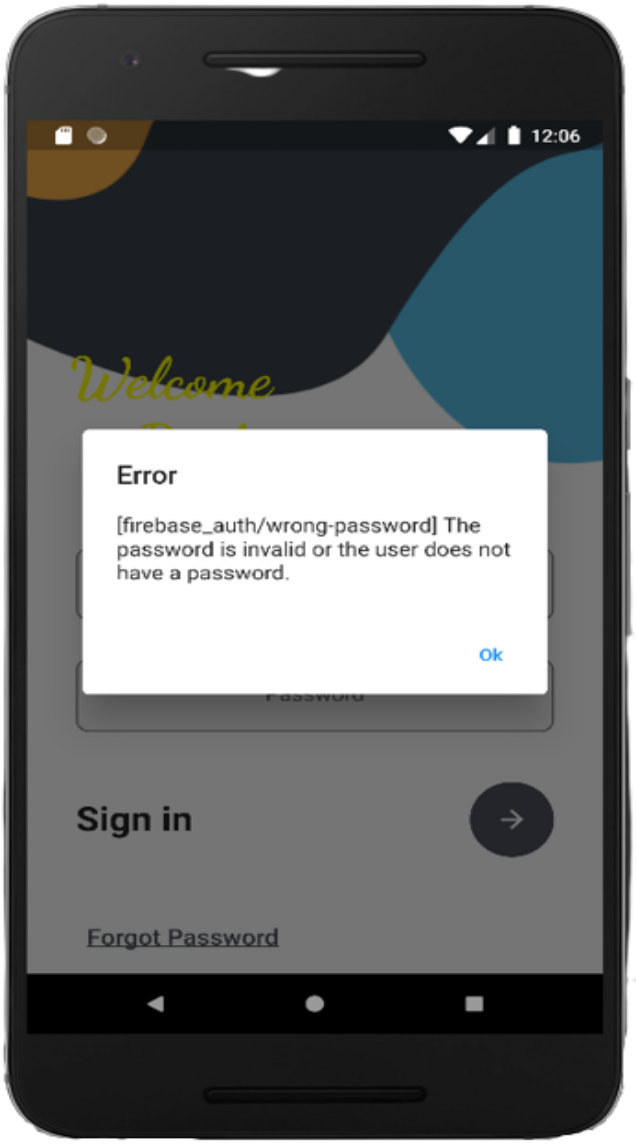
3.4.1 Login Page for user:

The initial screen is the login screen through which users would be able to access the application.



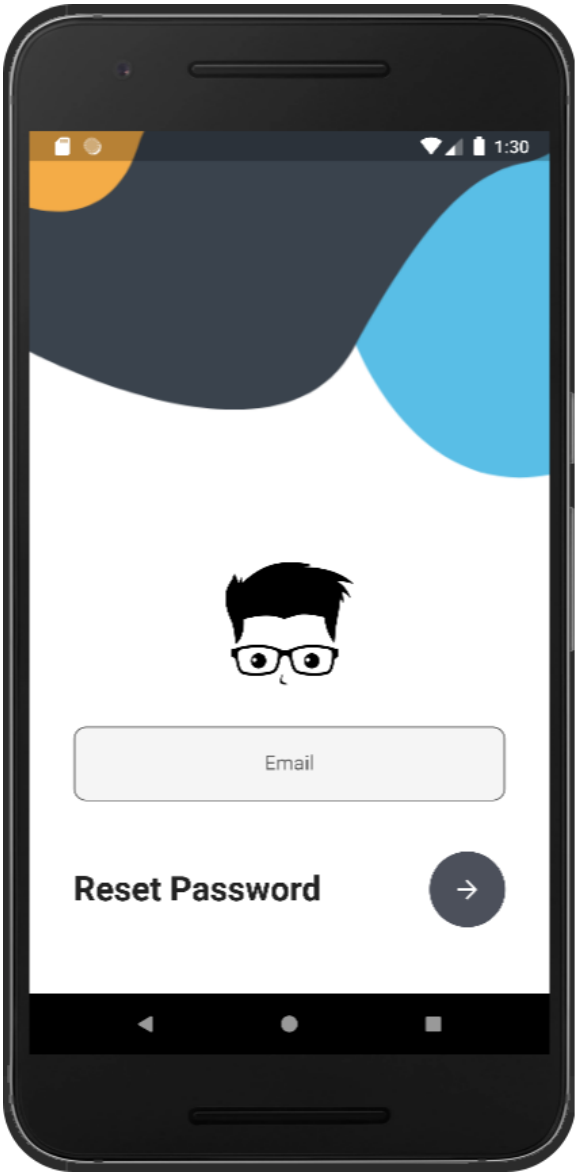
3.4.2 Authentication

An error message will appear when the user enters the wrong email address or password.



3.4.3 Additional Feature on Login page

If a user forgets his/her password then my clicking on Forgot Password user will get a reset password link on his/her registered email id.

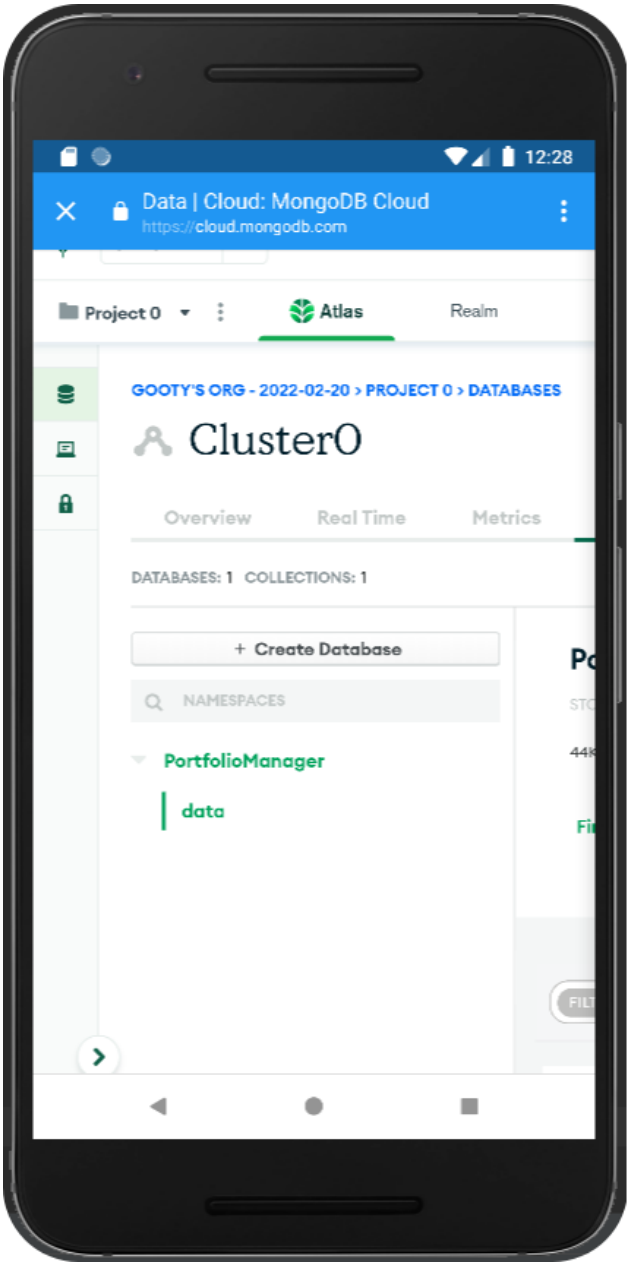


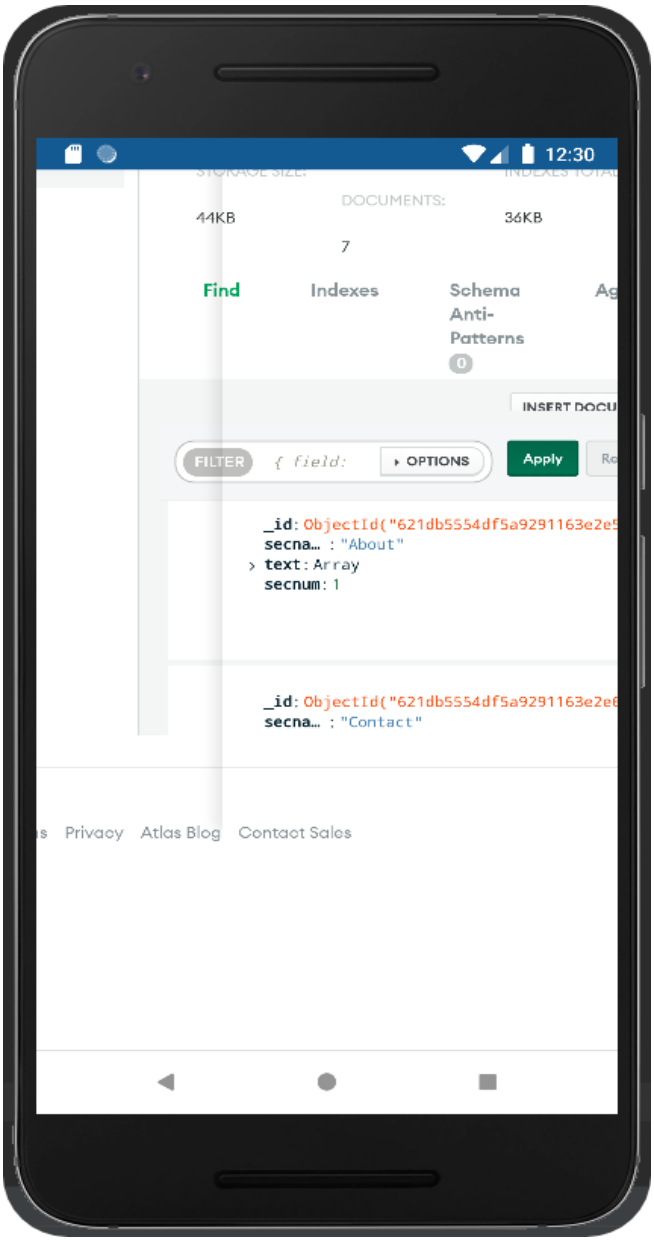
3.4.4 Launch Screen

- View Portfolio: With the view portfolio option selected, the application would render the live portfolio website on the screen.
- Edit Portfolio: Selecting the edit portfolio option will redirect the user to the database.

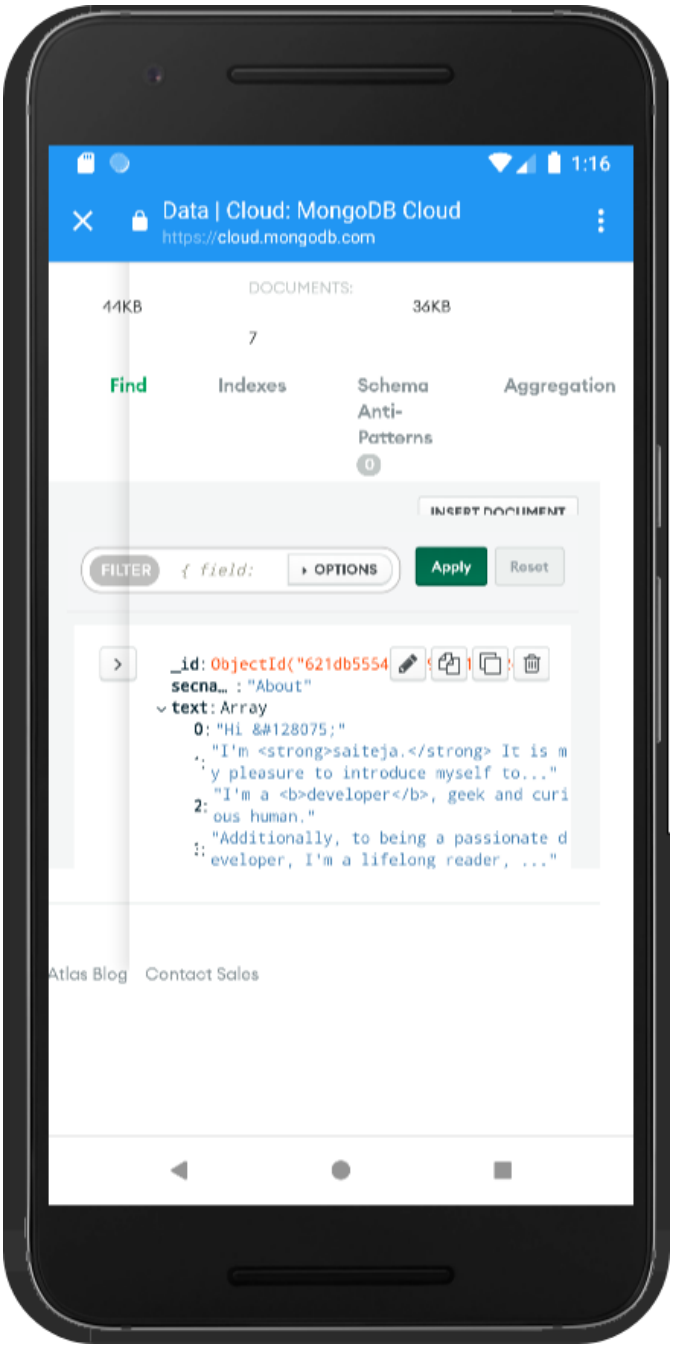
- **View Portfolio**:

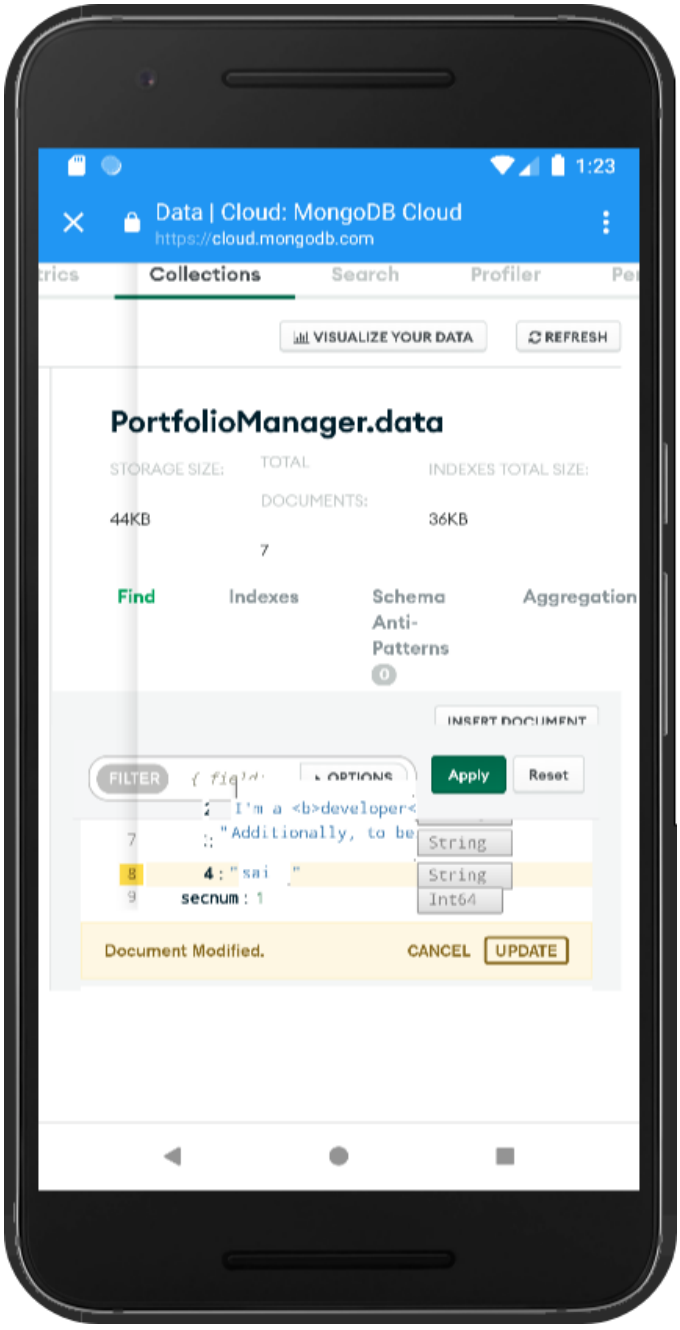
- **<u>Edit Portfolio</u>***:* Through this interface, the user can edit the database and the changes will be reflected on the live website



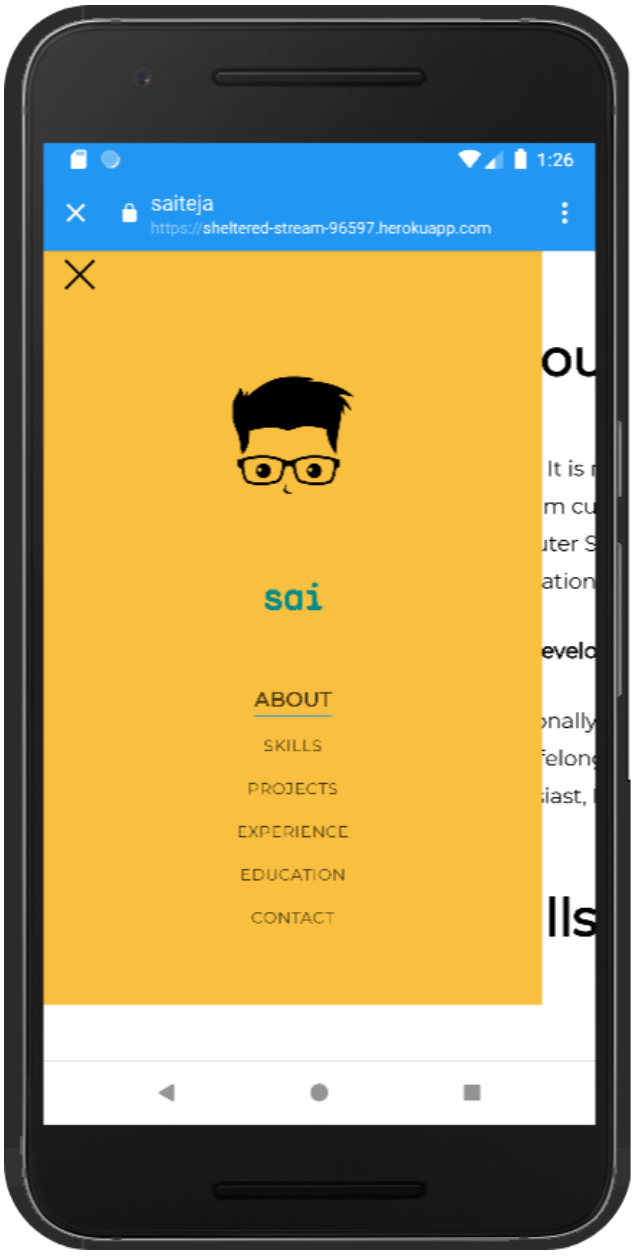


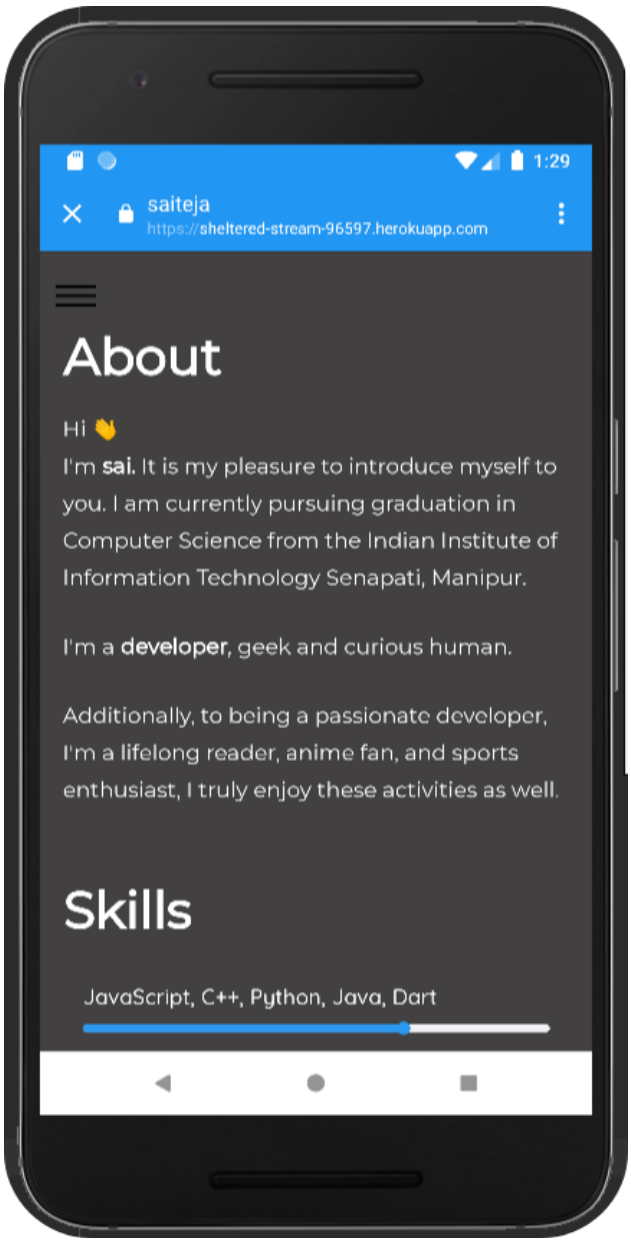
- By selecting a section to be edited, clicking on the editor icon, making necessary changes, and clicking on update document, the user can complete the changes.





➢ Check the changes made to the database by clicking on the view portfolio option on the launch screen.





3.4.5 Technology Stack

1) Dart
2) Mongo DB
3) MongoDB Atlas
4) Swift
5) Android Studio
6) Flutter
7) JSON

3.5 Testing

- **Manual Testing**

I have implemented all the mentioned functionality in my app using different types of stateful, stateless widgets and functions. I have manually tested functionality of the app and below I am providing a drive link for your referenceof testing which consists of all the required excel sheets.

Click Here: Manual Testing

Chapter 4 Conclusion

4.1 Limitations

Although I have put our best efforts to make this software flexible, easy to operate, we cannot rule limitations out even. Though the software presents a broad range of options to its users, some intricate options could not be covered on it.

- The application is only for Android and will not work for other mobile OS like blackberry, Symbian, IOS, etc.
- Only the content that appears on the website can be edited, but the style or new elements cannot be added to the website

4.2 Future Enhancements

- In future, I will give you the ability to change the style and add new elements to the website from the app itself.
- I can make a web version of the current application.

4.3 Contributions & Resources

- Gooty Saiteja (19010104):
  - Code Implementation (Portfolio Manager) & Database (Function for retrieve Data).
  - Use Case and UML Diagrams.
  - SRS implementation (Functional Requirements).
  - Bug Fixing
  - SRS Documentation
  - Database Architect and Design
  - Code Implementation (Role based Authentication)
  - Project Architect
  - Manual testing
  - Integrated Flutter with firebase.
  - Code Implementation (UI integrated with Firebase).

4.4 Resources

- Manual Testing
- GitHub

Appendix A

User manual

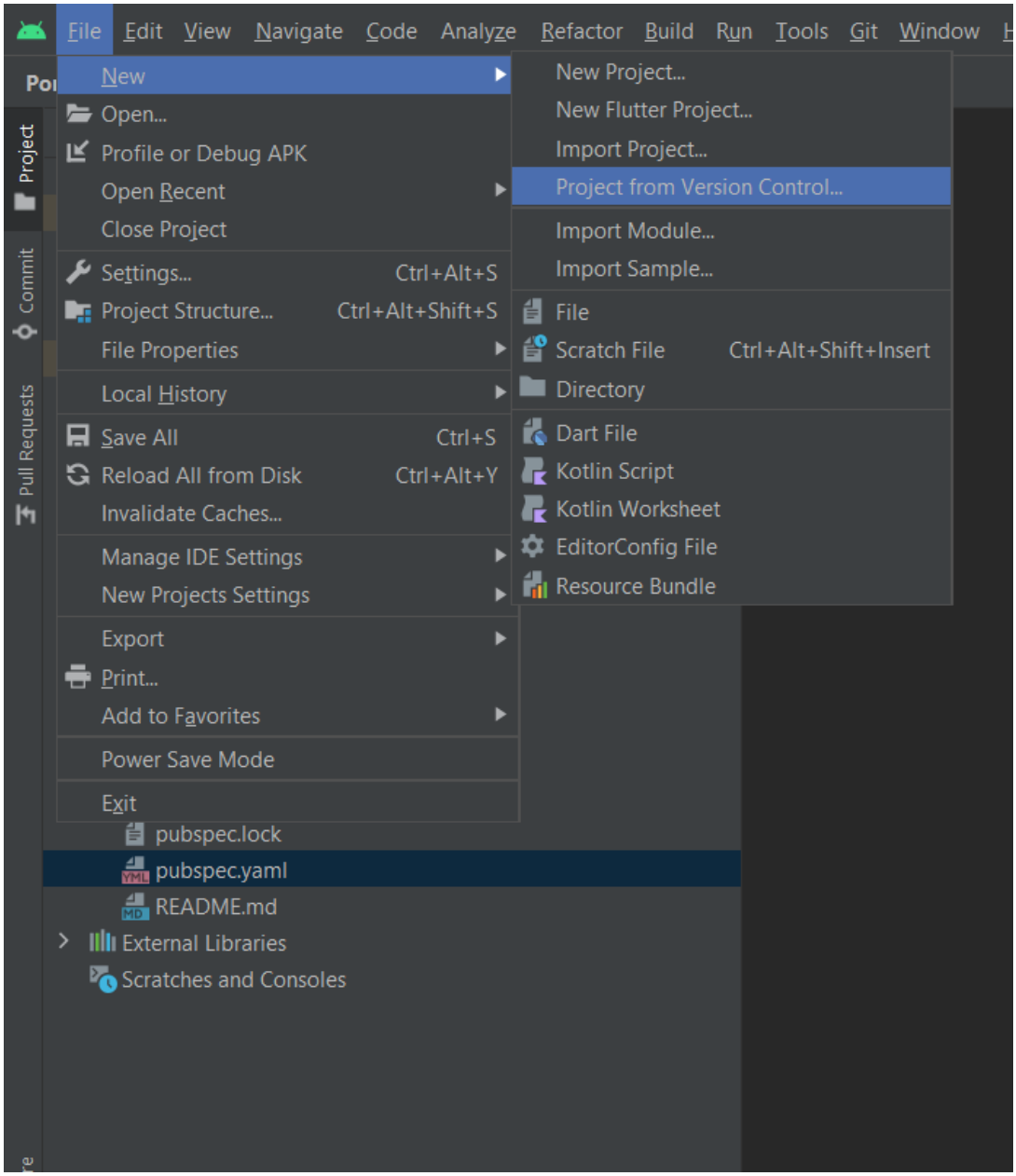
A.1 Introduction

By working on this project Portfolio Manager, I hope to enable easy editor access to the hosted website without actually having access to its source code. In order for us to update our personal portfolio website which display our work and personal information regularly, it is difficult to change the source code every time. Therefore, this app allows the user to update or edit the content that is displayed on the live website. of the system.

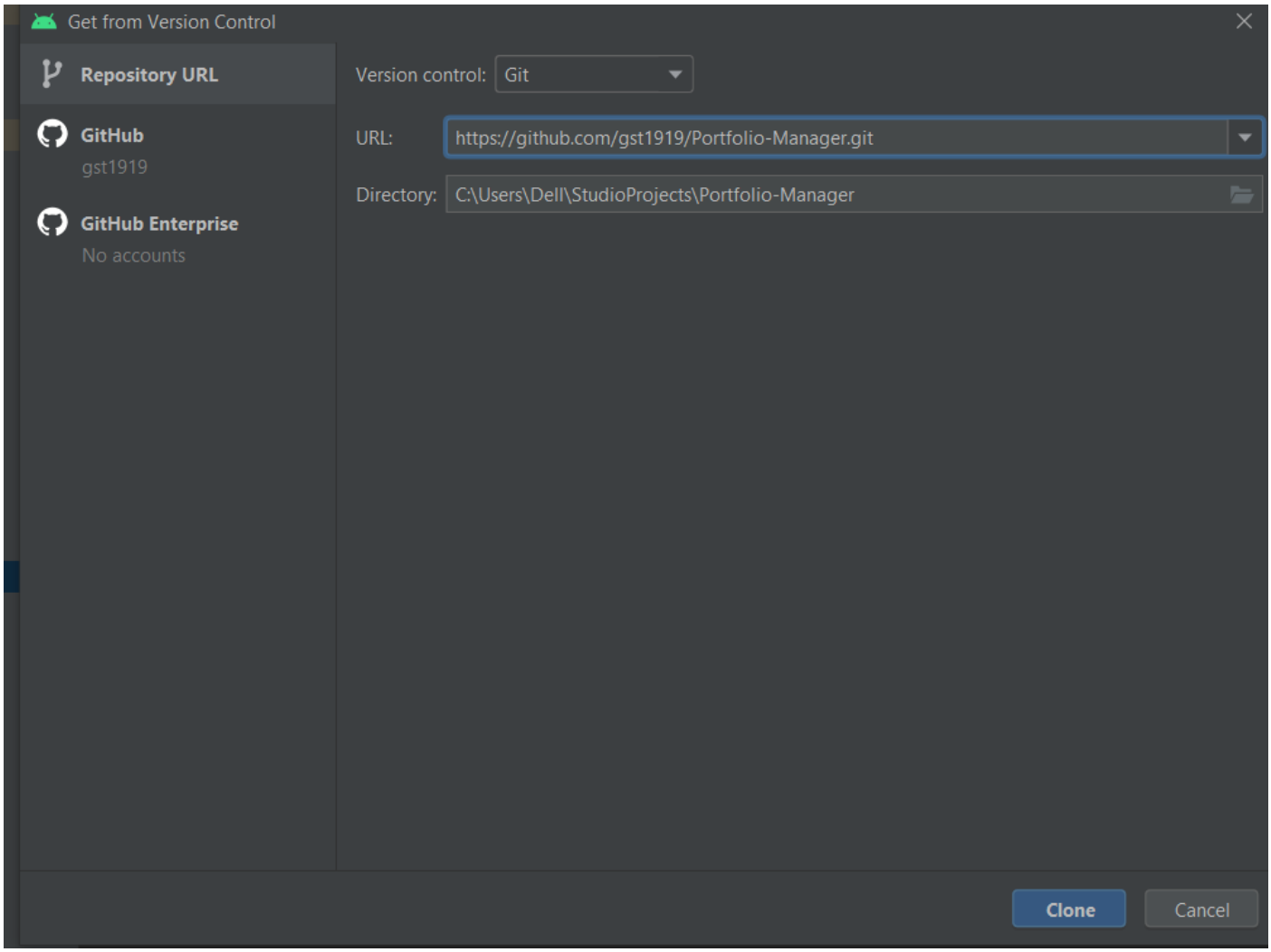
A.2 Steps to install Portfolio Manager Application

- To work on this application

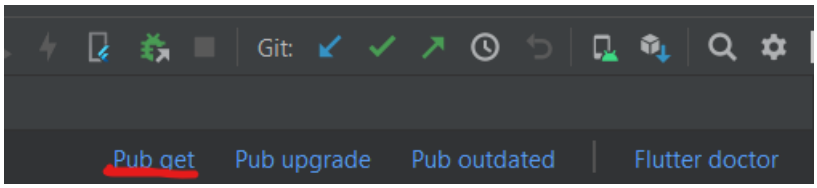
1. To develop a project from version control, open Android Studio and click on File ->New ->Project from version control as shown below.



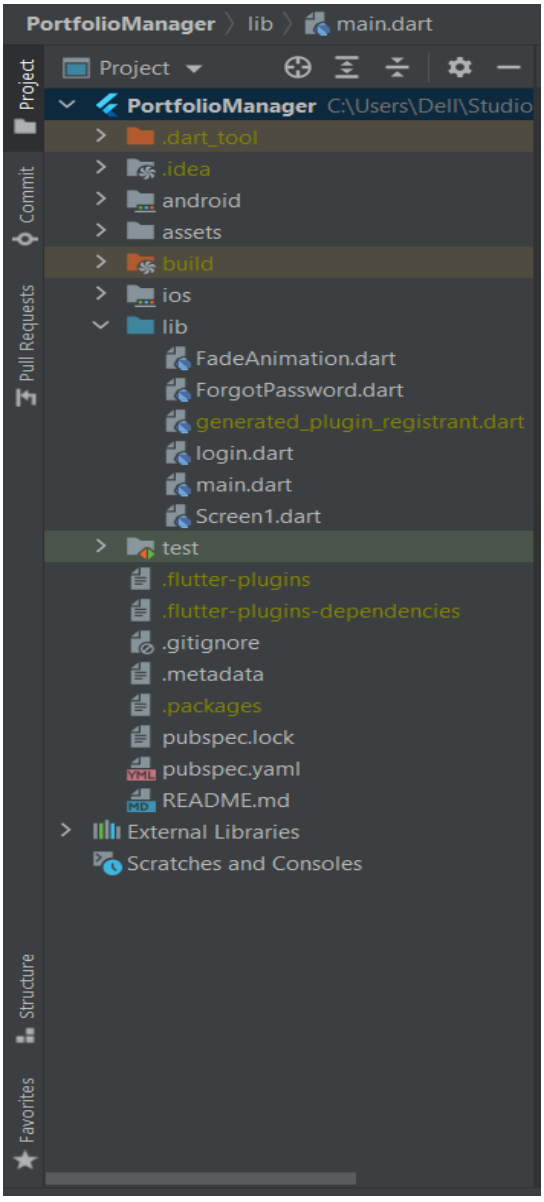
2. The version control should be set to git, and copy the URL below into the URL holder and click on clone as shown below.
   URL: https://github.com/gst1919/Portfolio-Manager.git



3. When the project files have been loaded, click **pub get** to add necessary dependencies, and you are ready to code.



4. Project File Structure



5. Application Requirements

```
dependencies:
  flutter:
    sdk: flutter
  simple_animations: ^4.0.1
  animated_text_kit: ^4.2.1

  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.4
  firebase_core: ^1.13.1
  firebase_auth: ^3.3.11
  shared_preferences: ^2.0.13
  url_launcher: ^6.0.20
  webview_flutter: ^3.0.1
  flutter_custom_tabs: ^1.0.4
  flutter_secure_storage: ^5.0.2



dev_dependencies:
  flutter_test:
    sdk: flutter
```

- **Test the application's functionality**

  Link to Apk

  Working Application

Bibliography

- IEEE 830-1998 standard for writing SRS document.
- Use case narration template referenced by OOAD by Benet (3rd Edition).
- Dart documentation ("https://dart.dev/guides").
- R. S Pressman, Software Engineering: A Practioner's Approach, 5th Ed, McGraw-Hill, 2001