SOFTWARE PROJECT FINAL REPORT

"STUDY BUDDY"

Team members

Joel Bellapurla - 2820745

Ashwin Korra - 2827384

Nivedha Kuzhandai Velu -2826551

Sriniketh Sowmithri - 2826981

Date : 12/06/2022

Table of Contents

3.1. Stakeholders for the system

3.2. Use cases

3.2.1. Graphic use case model

3.2.2. Textual Description for each use case

3.3. Rationale for your use case model

3.4. Non-functional requirements

4. Architecture

4.1. Architectural style(s) used

4.2. Architectural model (includes components and their interactions)

4.3. Technology, software, and hardware used

5. Design

5.1. User Interface design

5.2. Components design (static and dynamic models of each component)

5.3. Database design

6. Test Management

6.1. A complete list of system test cases

6.2. Traceability of test cases to use cases

6.3. Techniques used for test case generation

6.4. Test results and assessments (how good are your test cases? How good is your software?)

6.5. Defects reports

7. Conclusions

7.1. Outcomes of the project (are all goals achieved?)

7.2. Lessons learned

7.3. Future development

References

1. Introduction

1.1. Purpose and Scope

This web application “Study Buddy” is a useful platform where any student can find a study partner with their available days on-campus. The main objective of this project is to bridge students and the students who can provide help in their own available days. The big picture of the project is to make a user-friendly web application where any student can find a study partner inside the campus. This will technically improve an individual student's academic performance.

1.2. Product Overview

The project serves the purpose of a full stack application which will be used by the students to perform well in their academics. This application is implemented in such a way that every student can utilize it also, students can help each other in their available timings.

1.3. Structure of the Document

The project deals with such a policy, that only one student can search for one subject only. Also, the student can select one buddy only.

1.4. Terms, Acronyms, and Abbreviations

The term being used in the project, “the buddy” which denotes the student who can provide academic assistants.

2. Project Management Plan

2.1. Project Organization

The project organization involves students and admin if there is any technical issue with the application. Here the students are going to be the instructors. I.e Study Buddies and Students who are going to get help from other students.

2.2. Life Cycle Model Used

This project requires a waterfall project management methodology. The process of developing this application involves the following steps.

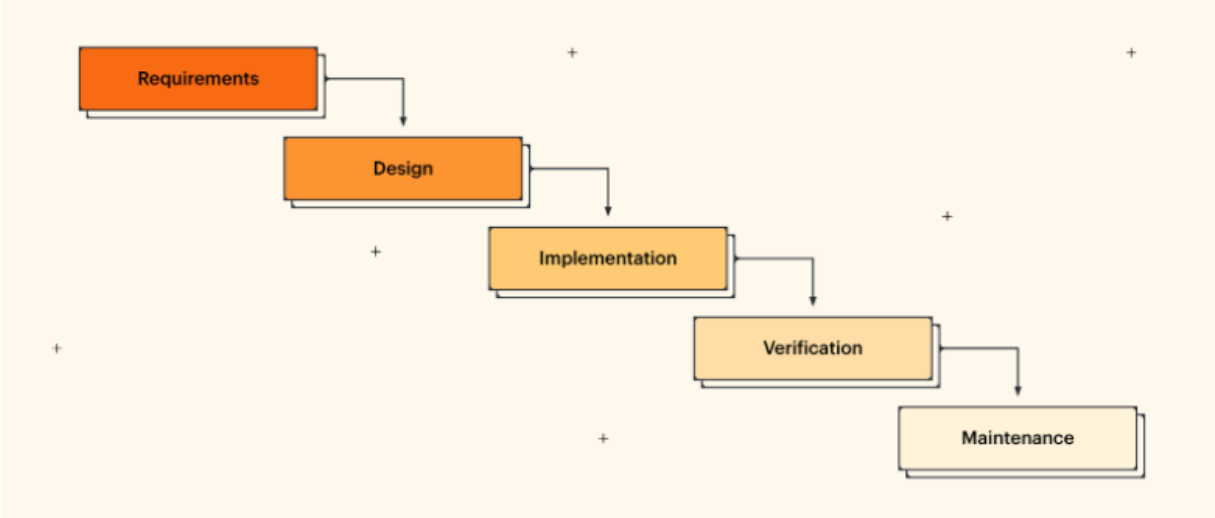


Image Source : Internet

Requirement : We have built a rough draft of the requirement document initially. Then as per the design discussed among the team, we built the requirement document. At the stage of Requirement, we discussed the GUI, which should be user friendly. The database to be used. Features to be created and developed.

Design : The web application was designed as per our requirements and made it so simple and user-friendly without any complications.The color, background, layout and the icons, everything comes under a unique design which makes the user attractive view.

Implementation : Implementation was a bit tricky as the features had to be placed in the right place and work. The functionalities of this application were designed in such a way that students do not face any trouble using it.

Verification : In the verification stage, we have checked whether the functionalities (Login, sign up, Need a buddy, i am a buddy, Submit buttons) work well.

Testing : In the testing phase, we have tested the user interface and also the functionalities of the application. Right from fonts, color, layout till the database connectivity.

2.3. Risk Analysis

During the risk analysis, we did not meet with any major risks.

2.4. Hardware and Software Resource Requirements

The project can be developed using Github Desktop and Visual Studio IDE. Hardware requirements would be Windows 9 and more. Also Mac OS can be used . Nothing more than that.

2.5. Deliverables and schedule

The deliverables show that each step has an outcome of another activity. Finally the project's outcome would be the results fetched from the database which is Firebase.

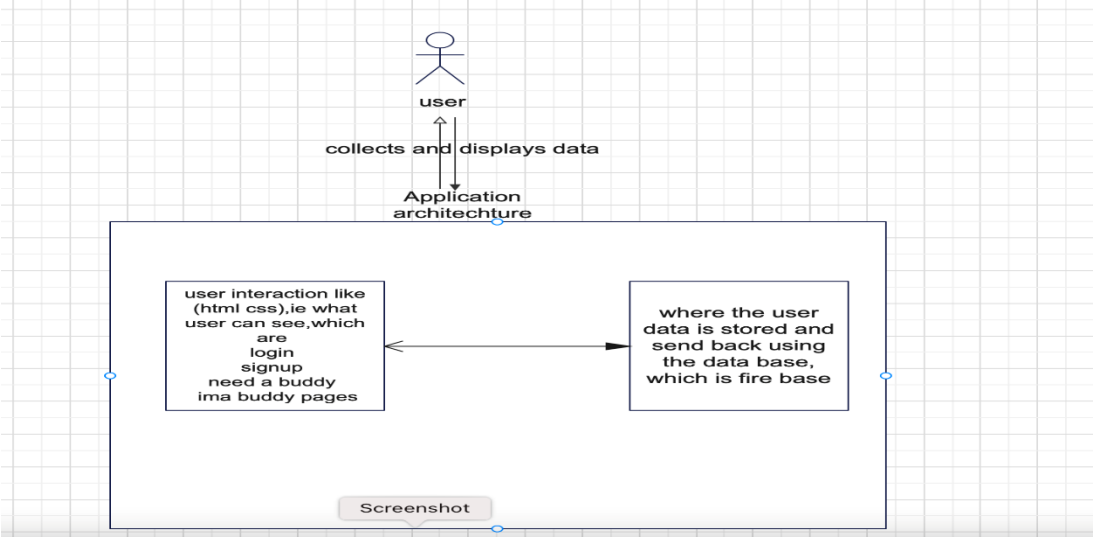
3. Requirement Specifications

3.1. Stakeholders for the system

Here,the stakeholders are the students. The users are students here who are going to use the application and who get benefited by the application.

3.2. Use cases

3.2.1. Graphic use case model



3.2.2. Textual Description for each use case

The user (student) registers as a Buddy , if he can provide help as a buddy. Another student can enroll/register in the app as a user account and search for buddies in his available timings. Also the student(user) can see the contact mail id for the buddies for further communication.

3.3. Rationale for your use case model

The purpose of this use case diagram describes the entire structure of the project in an effective way to understand the application easily and better.

3.4. Non-functional requirements

The performance and the usability of the use case diagram is being described here. The usability of this application is high and the performance is good when being demonstrated in the demo session.

4. Architecture

4.1. Architectural style(s) used

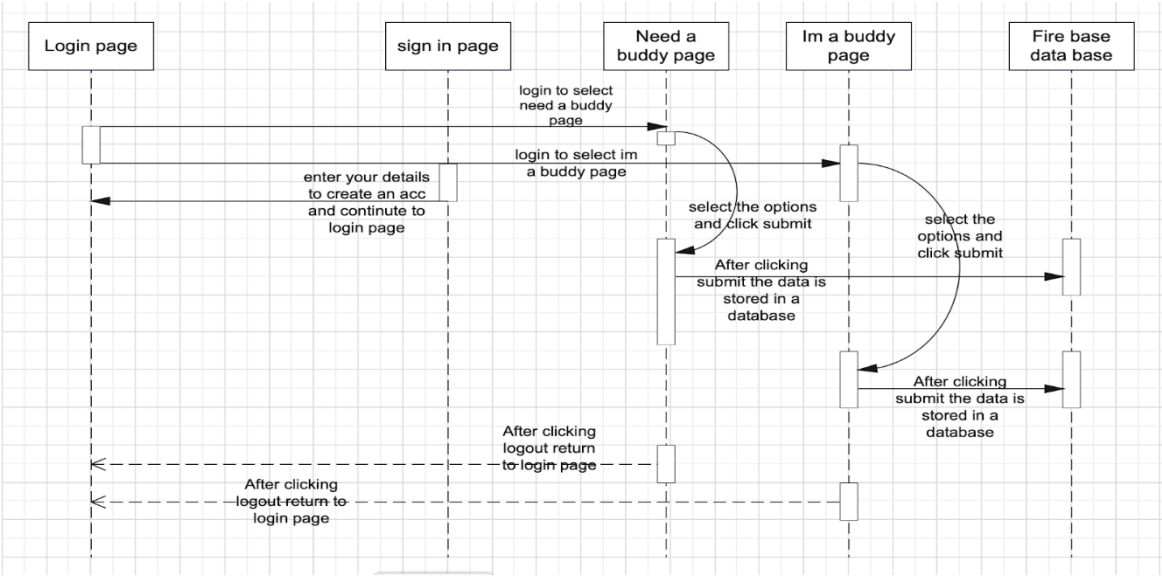
The architecture is planned on the basis of the State dig , Activity dig and class diagram , and the website design is done by using the basic concept of the Html And CSS stylesheet for proper orientation for the proper web page.

4.2. Architectural model (includes components and their interactions)

The interactions are shown by:

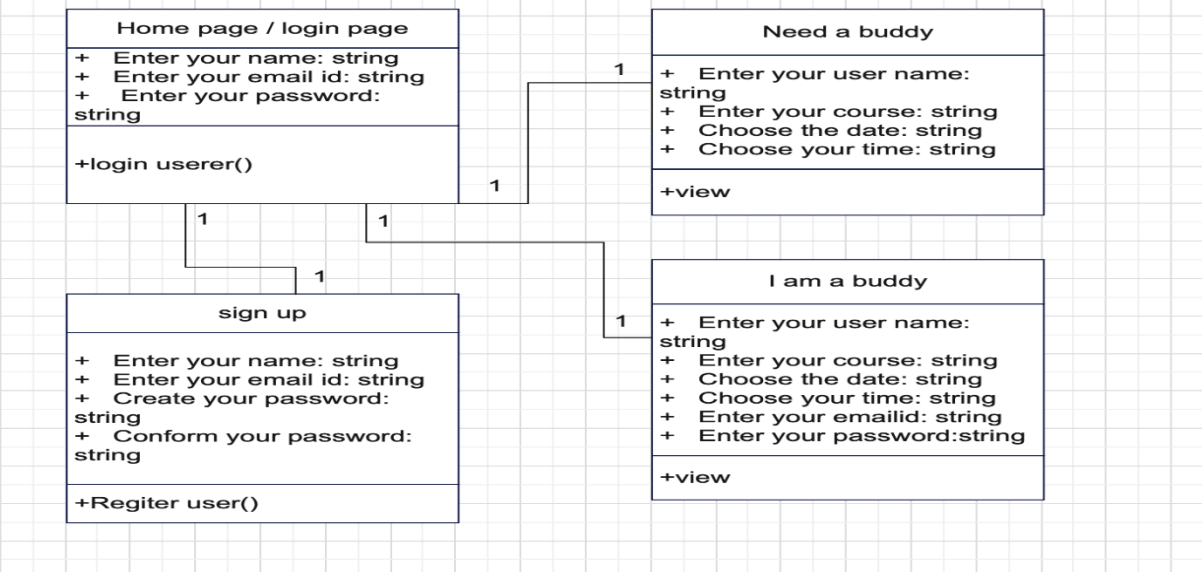
Sequential diagram:

Any user can create an Account after creating an account the user will login, and if the user requires assistance he is going to select a buddy page , if he can provide help as a buddy. Another student can enroll/register in the app as a user account and search for buddies in his available timings. Also the student(user) can see the contact mail id for the buddies for further communication.



Class diagram:

After creating an account the user will login, and if the user requires assistance he is going to select a buddy page , if he can provide help as a buddy. Another student can enroll/register in the app as a user account and search for buddies in his available timings. Also the student(user) can see the contact mail id for the buddies for further communication.



4.3. Technology, software, and hardware used

Technologies used on the software level are:

(i) VS code

(ii) FireBase(realtime database.)

Hardware configurations are:

A personal computer of basic configuration will work

5. Design

5.1. User Interface design

1. **Sign in page:**

When the new user accesses the application, the user must be able to create an account with a valid email id and password. If the mail id does not have '@', the user must be able to see the error message., the user must be able to see the error message. If the password is less than 6 characters, the user should be able to see the error message that the password is too short.

2. **Login page:**

After creating an account, the user will use the credentials and login to the home page.

3. Home page:

The user should be able to see and click the “I am a Buddy” Button and “Need a Buddy button”.

4. Need a Buddy:

The user should be selecting a valid input from the dropdown to proceed with further actions and able to select a valid input from the dropdown to proceed with further actions. After that user should be able to view the results fetched from the database when the user clicks “find” Button.

5. I am a Buddy:

The user should be able to use the dropdown to select the subjects and should be able to select the available days from the dropdown for further actions. The user should be able to see the error message if the user skips a step of selecting a valid input.

5.2. Components design (static and dynamic models of each component)

Input:

1)For login and Sign up:

· Users access the application as new user, and able to click the create account button

· The user should be able to enter the mail id with’@’ and a valid password of length of 6-10 characters

· The user should be able to enter the valid email and password, which was used to create an account and login to the application.

· The user should be able to enter the password and confirm the password to complete the sign-up process

2)For Need a Buddy feature:

· The user should be able to select the available days with a valid input from the dropdown

· The user should be able to select the valid input (subject) from the dropdown

· The user should be able to view the results when the user able to click the “find” button

3)For I am a Buddy feature:

· The user should be able to provide the subject name in the textbox / select it from the dropdown with a valid subject name.

· The user should be able to select the available days from the dropdown

· The user should be able to select the valid inputs from the dropdown, if not the user should get the error message.

Output exceptions:

1)For login and sign up:

· The user should be able to navigate to next page

2)For Need a Buddy page:

· The user should be able to select the valid input from the dropdown.

· The user should view the results.

3)For I am a Buddy page:

· The user should enter the valid subject’s name in the textbox/ select it from the dropdown

· The user should be able to provide the valid input else, the user should be able to see the error message. “Select a valid day” and “Select a valid subject”.

5.3. Database design

Firebase is a set of hosting services for any type of application. It offers NoSQL and real-time hosting of databases, content, social authentication, and notifications, or services, such as a real-time communication server. We used firebase to store the student/ instructor details while signing up. A user can login into the system when Firebase authenticates the user login information this is done using.

6. Testing

6.1. A complete list of system test cases

Sign up Page and Login up :

Test_ID	Test Input	Expected Output	Description
01	Users access the application as a new user, and are able to click the create account button.	Navigates to Sign up page	When the new user accesses the application, he must be able to create an account with a valid email id and password.
02	The user should be able to enter the mail id with'@' and a valid password of length of 6-10 characters	User is able to enter the email id and the password and create an account	If the mail id does not have '@', the user must be able to see the error message., the user must be able to see the error message. If the password is less than 6 characters, the user should be able to see the error message that the password is too short.
03	The user should be able to enter the valid email and password , which was used to create an account and login to the application.	User is able to login with valid email id and password	If the user does not use the valid email id and password, it shows the error message that "Account not found"

04	The user should be able to enter the password and confirm the password to complete the signup process	The user should be able to create an account	If the password doesn't match, the user should be able to see the error message that "password doesn't match".

Homepage:

Test_ID	Test Input	Expected Output	Description
11	The user should be able to click the "I am a Buddy" Button.	The user should be able to navigate to next page	The user should use the page and proceed with further actions
12	The user should be able to click the "Need a Buddy" Button	The user should be able to navigate to the next page.	The user should use the page and proceed with further actions

"I am a Buddy" page

Test_ID	Test Input	Expected Output	Description

21	The user should be able to provide the subject name in the textbox / select it from the dropdown with a valid subject name.	The user should enter the valid subject name in the textbox/ select it from the dropdown	The user should be able to use the dropdown to select the subjects.
22	The user should be able to select the available days from the dropdown	The user should select available days from the dropdown	The user should be able to select the available days from the dropdown for further actions.
23	The user should be able to select the valid inputs from the dropdown, if not user should get the error message.	The user should be able to provide the valid input else, the user should be able to see the error message. “Select a valid day” and “Select a valid subject”.	The user should be able to see the error message if the user skips a step of selecting a valid input.

“Need a Buddy”

Test_ID	Test_input	Expected Output	Description

31	The user should be able to select the available days with a valid input from the dropdown	The user should be able to select the valid input from the dropdown.	The user should be selecting a valid input from the dropdown to procced with further actions.
32	The user should be able to select the valid input (subject) from the dropdown	The user should be able to select the valid input from the dropdown	The user should be able to select a valid input from the dropdown to proceed with further actions.
33	The user should be able to view the results when the user able to click the “find” button	The user should view the results.	The user should be able to view the results fetched from the database when the user clicks “find” Button.

6.2. Traceability of test cases to use cases

Test Scenario ID	Test Case ID	Status	Defects
1	User access the application as new user, and able to click the create account button.	Passed	None
2	The user should be able to enter the mail id with’@’ and a valid password of length of 6-10 characters	Passed	None
3	The user should be able to enter the valid email and password , which was used to create an account and login to the application.	Passed	None
4	The user should be able to enter the password and confirm the password to complete the sign up process	Passed	None

11	The user should be able to click the “I am a Buddy” Button.	Passed	None
12	The user should be able to click the “Need a Buddy” Button	Passed	None
21	The user should be able to provide the subject name in the textbox / select it from the dropdown with a valid subject name.	Passed	None
22	The user should be able to select the available days from the dropdown	Passed	None
31	The user should be able to select the available days with a valid input from the dropdown	Passed	None
32	The user should be able to select the valid input (subject) from the dropdown	Passed	None
33	Session expiration	Failed	DT01

6.3. Techniques used for test case generation

Decision Table based testing - It is also called as the cause-effect table, This software testing technique is used for functions which respond to a combination of inputs or events. For example, a submit button should be enabled if the user has entered all required fields.

The first task is to identify functionalities where the output depends on a combination of inputs. If there are large input sets of combinations, then divide it into smaller subsets which are helpful for managing a decision table.
For every function, you need to create a table and list down all types of combinations of inputs and its respective outputs. This helps to identify a condition that is overlooked by the tester.

State Transition - In State Transition technique changes in input conditions change the state of the Application Under Test (AUT). This testing technique allows the tester to test the behavior of an AUT. The tester can perform this action by entering various input conditions in a sequence. In State transition technique, the testing team provides positive as well as negative input test values for evaluating the system behavior.

6.4. Test results and assessments (how good are your test cases? How good is your software?)

We have passed all the test conditions but are unable to maintain the session when the application is opened in a different browser.

6.5. Defects reports

Test_ID	Test Input	Expected Output	Description
DT01	Opening application in a different browser.	Session should not expire and user should be able to login and get the results	When the user logs into the application and makes changes. When opened in a different broweser simultaneously, the results should be displayed, but the session is expiring and the user is unable to make changes

7. Conclusions

7.1. Outcomes of the project (are all goals achieved?)

The outcomes of the project are effective search of Study buddies and check for the available timings. The goals are achieved in accordance with the effective code. It was developed and tested to avoid the errors while working on the application.

7.2. Lessons learned

The lessons learnt are useful to upgrade our skills and enhance our profile. The software development cycle- waterfall model is learnt and implemented throughout the project. We followed all the steps based on the waterfall model.The frontend was designed using

HTML and CSS. The backend part was developed using Javascript. And the NoSQL database, Firebase is the new skill we learnt , where we got to learn it to use effectively.

7.3. Future development

The future work is to develop the feature for contact details to contact the buddy, directly from the application which redirects to the gmail account/outlook. Also, we are planning to add the feature of the enquiry page.

References

1. Full Stack Web Development: Vision, Challenges and Future Scope Gurjeet Singh1, Madiha Javed2, Dr. Balwinder Kaur Dhaliwal3

2. Full Stack Web Development Teaching: Current Status and a New Proposal Anna Petrikoglou and Theodore H. Kaskalis Department of Applied Informatics, University of Macedonia, 156 Egnatia Str., 54636, Thessaloniki, Greece

3. PATDOC - An Online Appointment Management System, June 2021DOI:10.13140/RG.2.2.20191.18086, Report number: 01061216, Affiliation: Jain University

4. https://www.guru99.com/software-testing-techniques.html