Test Document

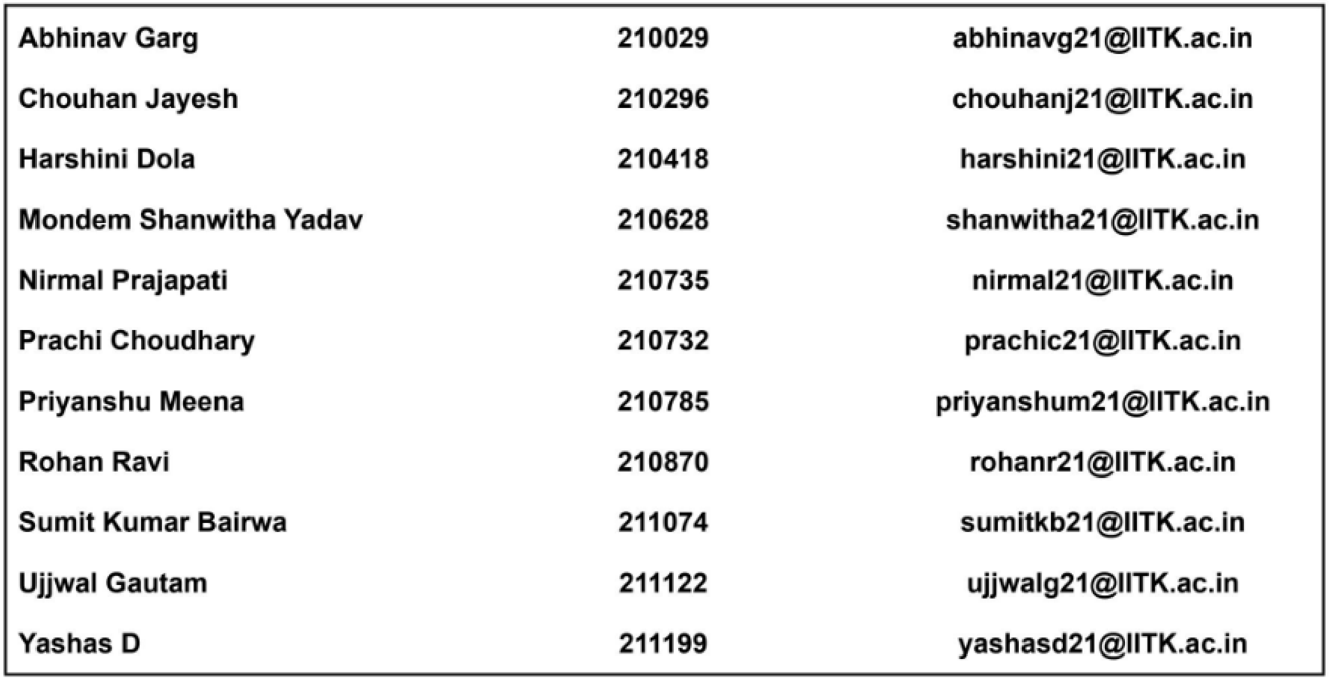
for

mapGo

Version 1.0

Prepared by

Group 6: CS253 Group Name: Wall-E(s)



Abhinav Garg	210029	abhinavg21@IITK.ac.in
Chouhan Jayesh	210296	chouhanj21@IITK.ac.in
Harshini Dola	210418	harshini21@IITK.ac.in
Mondem Shanwitha Yadav	210628	shanwitha21@IITK.ac.in
Nirmal Prajapati	210735	nirmal21@IITK.ac.in
Prachi Choudhary	210732	prachic21@IITK.ac.in
Priyanshu Meena	210785	priyanshum21@IITK.ac.in
Rohan Ravi	210870	rohanr21@IITK.ac.in
Sumit Kumar Bairwa	211074	sumitkb21@IITK.ac.in
Ujjwal Gautam	211122	ujjwalg21@IITK.ac.in
Yashas D	211199	yashasd21@IITK.ac.in

Course: CS253

Project Mentor : Mr. Anirudh Nanduri

Date:04/02/2023

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft Type and Number	Full Name	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.	00/00/00

1 Introduction

Mention the test strategy

The main test strategy used on our system was manual Testing.The backend was tested using the Postman software to see all the routes covered by the system.This covered most of the bugs and errors of the frontend, backend and the integration part of our code.

When was the testing conducted?

The testing part of the code was done in parallel with the development.As soon as a bug was discovered,the testing team contacted the developers, which then rectified the code.This helped to use the size of the team beneficially to make the code running better in shorter time.

Who were the testers?

We had an Independent Testing team composed of 4 of the group members.The reason for this was that both the processes were running simultaneously, as soon we found a bug the soon it got fixed.

What coverage criteria were used?

The coverage criteria that were primarily used were that all the different test cases were taken manually and the testing was implemented until all the different inputs are exhausted. In this fashion, most of the bugs were eliminated.

Have you used any tool for testing?

For backend testing , the postman software was used to cover all the routes.For the rest we did not use any tools as manual testing was enough to cover all the bugs of the system.

2 Unit Testing

Our Unit testing is divided into major parts :-

1.) Unit Testing for User
2.) Unit Testing for Host

1.) Unit Testing for User

1.1) Sign_up Api

Unit Details: The Signup API is used to render the register page. It takes the input as the username, IITK email and password to register the user.

Test Owner: Sumit Kumar Bairwa

Test Date: 03/30/2023 - 03/30/2023

Test Results: The user can register themself on the website by entering username, email ID, and setting a password successfully after verification of IITK mail Id. An OTP will be sent for mail ID verification. But following bugs occurred in the respective process ,which were removed later.

1. The user can make more than one profile using a single mail ID, which shall be restricted to a unique mail ID. [different profiles using single mail id]

2. On the sign_up page, the user can register successfully even after entering different entries in password and confirming password. [pw and confirm pw must be same, which is violated and still registration is successful]

3. Users can create more than one profile with the same username and different mail IDs. For registration, the same OTP works for both IDs. [same otp works for different mail id with same username]

4. If a user enters an invalid mail ID, then there should be an alert pop-up asking the user to verify and enter a valid mail ID. [no notification for invalid email id]

5. All fields should be made compulsory to be filled.

Structural Coverage: Users can not sign_up by any other form of mail Id except “abc@iitk.ac.in”. And can’t sign_up using registered mail id or registered username. In the above conditions the system will throw an error message.

Additional Comments: None

1.2) Sign_in Api

Unit Details: The SignIn API is used to render the sign in page. It takes the input as the username and password of the already registered user. And there is a forgot password button by which password can be changed.

Test Owner: Chouhan Jayesh

Test Date: 03/30/2023 - 03/30/2023

Test Results:

Registered user successfully sign_in by his credentials(username and password) successfully. If wrong credentials are entered ,the system will throw an error. User can change or reset his password by “ forgot password ” option. But following bugs occurred in the respective process ,which were removed later.

- Forgot password does not throw errors on entering wrong credential, including wrong otp.
- The last otp works for changing password multiple times for the same username.

Structural Coverage: Given valid username and wrong password, invalid username and wrong password and also left any of the fields empty. In all these cases the system throws an error message.

Additional Comments: None

1.3) Map

Unit Details: This api is used for map navigation.

Test Owner: Sumit Kumar Bairwa

Test Date: 03/30/2023 - 03/30/2023

Test Results: Map is working well. It perfectly shows the direction to reach from point A to point B. We can put two pins on the map and it will show direction for it. Pop-ups are working well on the map for created events. The map is working specially for IITK-campus , its two pin functionality is restricted for the areas other than campus. But it will show the direction if you put location by search option , not by putting pins on two locations if a location is from an area other than campus.

Structural Coverage: check the navigation system , we put our start pin and destination pin on the map and check it again and again to ensure it is working well. And try to put pins for areas other than campus and try to find a route.

Additional Comments:None

1.4) Subscribe Host Api

Unit Details: This api is used for subscribing to the host.

Test Owner: Chouhan Jayesh
Test Date: 03/30/2023 - 03/30/2023

Test Results: Host is subscribed successfully and the subscriber has access to view all the events which are created by the host.

Structural Coverage : subscribe to the host and check their created event.
Additional Comments : None

1.5) View Event of subscribe Host

Unit Details: This api is used to view all events which have been created by the user's subscribed host.

Test Owner: Chouhan Jayesh
Test Date:03/30/2023 - 03/30/2023

Test Results: user successfully view all events of subscribed host.

Structural Coverage: subscribe different - different host and check their event every time after subscribing the host.

Additional Comments: None

1.6) Set reminder for events

Unit Details: This Api is used for setting a reminder of an event for the user.

Test Owner: Sumit Kumar Bairwa
Test Date:03/30/2023 - 03/30/2023

Test Results: user successfully set a reminder for an event.

Structural Coverage: set a reminder for an event which has gone away , in this condition the system shows an error.

Additional Comments: None

1.7) Logout

Unit Details: Logout api is used to logout the user.

Test Owner: Sumit Kumar Bairwa
Test Date: 03/30/2023 - 03/30/2023

Test Results: user can successfully logout from his account by logout navbar.

Structural Coverage: After clicking the logout button check the user log out and return on the sign_in page. Tried it multiple times.

Additional Comments: None

2.) Unit Testing for Host

2.1) Sign_in

Unit Details: The Sign_In API is used to render the host sign in page. It takes the input as the hostname and password of the already registered hosts which were registered by us. And there is a forgot password option by which the host can be able to change the password.

Test Owner: Chouhan Jayesh
Test Date:03/30/2023 - 03/30/2023

Test Results: Registered host successfully sign_in by his credentials (hostname and password) successfully. If wrong credentials are entered, the system will throw an error. Host can change or reset his password by the " forgot password " option.
a Bug occurred in the respective process, which was removed later.

- forgot password doesn't work not well.

Structural Coverage: Given valid hostname and wrong password, invalid hostname and wrong password and also left any of the fields empty. In all these cases the system throws an error message.

Additional Comments: None

2.2) Create event

Unit Details: Create event api is used for hosts to create events by filling the details of the event.

Test Owner:Sumit Kumar Bairwa
Test Date:03/30/2023 - 03/30/2023

Test Results: Host is able to create an event successfully by filling all the required fields.
But the following bug occurred in the respective process, which was removed later.

- *Same events can be created multiple times.*

Structural Coverage: : Different possible events were created and were tried to render. Such events include those types which have one or many incomplete or invalid fields. After a few attempts all the input cases were exhausted.

Additional Comments: None

2.3) Show events

Unit Details: This api is used for the host to view all events which are created by him.

Test Owner: Sumit Kumar Bairwa
Test Date:03/30/2023 - 03/30/2023

Test Results: This functionality is working successfully. Host able to see his created event.

Structural Coverage: Check every time that all the events show there after creating a new event multiple times by the host.

Additional Comments: None

2.4) Edit event

Unit Details: This API is used for hosts to edit events.

Test Owner:Chouhan Jayesh
Test Date: 03/30/2023 - 03/30/2023

Test Results: This functionality is working well. Host is able to edit his created event and can also delete it.

Structural Coverage: By clicking the edit event, the host can edit all fields and it is updated on-time after saving it.

Additional Comments: None

2.5) Logout

Unit Details: This api is used for hosts to logout from the website.

Test Owner: Chouhan Jayesh
Test Date: 03/30/2023 - 03/30/2023

Test Results: Host can successfully logout by clicking logout.
Structural Coverage: After clicking the logout button check the host log out successfully and return on the host sign_in page. Tried it multiple times.

Additional Comments: None

3 Integration Testing

1.) User Signup and Email Verification

Module Details: In this integration test we integrated two majority different units present in our web app which are user registration and their email verification. Just after the user enters their email id and requests for OTP, an OTP sends to the user's mail id and after it they have to verify the email id by OTP. Only after the mail id verification, the user is able to register successfully. The user can be allowed to sign in using their credentials.

Test Owner: Prachi choudhary
Test Date: 03/31/2023 - 03/31/2023

Test Results: This integrated system works perfectly fine with all different user inputs. Some bugs were found but they were also removed quickly. Some simple Structural coverage used to find different errors were inputting invalid email id and waiting it to verify, closing the window and trying to access the sign in page directly and so on.

Additional Comments: we need many mail ids to test our web app so we created a bug in it and started testing with our limited ids.

2.) User Sign in and Rendering that user's data from Database

Module Details: In this module, users are allowed to sign in by entering his personal credentials only , the user was fetched from the database to the backend and finally it was rendered in the frontend. The basic data specific to a particular user includes all the hosts he/she has subscribed to ,and their (host) events details. he/she can see the map on the home screen.

Test Owner: Priyanshu Meena
Test Date:03/31/2023 - 03/31/2023

Test Results: All the user data were fetched correctly and thus rendered correctly as well in the frontend. Since all unit testings were already done, the integration testing of this section did not create new bugs and thus the page worked flawlessly.

Additional Comments: None

3.) Host sign in and operations on Event

Module Details: In this module ,host were allowed to sign in ,and thus host will come at the the profile page and the host data specific to that host only .The basic operation on event to a particular user includes create an event, can be able to edit , delete and fetch the data of any event created by that host only.

Test Owner: Prachi Choudhary

Test Date:03/31/2023 - 03/31/2023

Test Results: All the host data were fetched correctly and thus rendered correctly as well in the frontend. There were some issues like events created multiple time but then our team began to solve that immediately.

Additional Comments: None

4.) Subscribe to host by user and rendering host's data

Module Details*:* The user can subscribe to any host by searching the name of host in search bar after subscribing the host, user can get information of subscribed host that includes all events created by host, when and where event will be performed ,and also can unsubscribe the host.

Test Owner:Prachi Choudhary

Test Date: 03/31/2023 - 03/31/2023

Test Results: After the test we found that we couldn't unsubscribe the host. After some short time we removed the bug from the web app and all other things were working well.

Additional Comments: None

5.) User interaction with map

Module Details: In this module, the user can access the map and can search a path for his destination by setting one pin on starting position and second pin on destination position.

Test Owner: Priyanshu Meena

Test Date: 03/31/2023 - 03/31/2023

Test Results: Map is working properly. user can saerch a path by putting pins or search location for start and end points.

6.) Integrating the fronted with the backend as well as the database

Module Details: This module required a significant amount of work as we had to integrate the three major components of our software. The different groups that worked on Database, Frontend, and Backend were involved and had to collaborate for this testing.

Test Owner: Priyanshu Meena
Test Date:03/31/2023 - 03/31/2023

Test Results: There were a few issues with the design of the frontend and backend which rendered some features incompatibly, and our team immediately began work to resolve them.

4 System Testing

1. **Requirement: PERFORMANCE REQUIREMENT**

Test Owner: Sumit Kumar Bairwa
Test Date: 04/01/23 - 04/01/23

Test Results: Performance issue is one of the most important problems we faced since it was our first time developing any complete web app. MongoDB database is used. The maps API used is not effective. The interface lags behind in responsiveness, the pop-ups for when OTP is sent are delayed.

Additional Comments: None

2. **Requirement: EXTERNAL INTERFACE REQUIREMENT**

Test Owner: Chouhan Jayesh

Test Date: 04/01/23 - 04/01/23

Test Results: The UI Interface, being the first interaction of the user with the web app, is the main aspect of any web application. The web-app is tested to be responsive in various operating systems viz. MacOS, Linux, Windows, Android, iOS. It was also tested using the responsive feature of the chrome developer tools.

Additional Comments: None

3. **Requirement: FUNCTIONAL REQUIREMENTS**

Test Owner: Prachi Choudhary

Test Date: 04/01/23 - 04/01/23

Test Results: Various functionalities of our platform were tested as the user can successfully register, log in and then subscribe to various hosts and search for various events. Clubs/societies or any other verified host can maintain their channel and add, edit or delete any event. Some issues faced were that alert pop-up would not generate on entering any invalid entry.

Additional Comments: None

4. Requirement: SAFETY AND SECURITY REQUIREMENTS

Test Owner: Priyanshu Meena
Test Date: 04/01/23 - 04/01/23

Test Results: For security of user data, OTP is generated for registering new users to the platform. The Verify-email system is tested thoroughly and no possible bugs are prevalent. The Host cannot sign-up on their own but can sign-in after their profiles are made by the developers of mapGo, hence not cluttering the events section of the user's profile. Following bugs were found during security testing-

1. Different users with the same username can sign-in using the same OTP.
2. Mail including the OTP is sent even to invalid mail IDs.

Additional Comments:None

5.Conclusion

How Effective and exhaustive was the testing?

The unit testing and most parts of integration testing are manual. We have tried to maximize the exhaustiveness of our testing process.The unit testing has been done quite extensively, covering almost all the functions. Integration testing was done for most of the connected modules. System testing was done for all the requirements of the SRS which have been implemented by us.

Which components have not been tested adequately?

All the parts of the code that were complete have been tested appropriately and thoroughly , as far as we were capable of.

What difficulties have you faced during testing?

The most difficult part of the testing process was understanding the code,all the routes in the backend and the functionalities implemented by the developers.

How could the testing process be improved?

The testing could have been better if we could have imposed automated testing on some of the functionalities.

Appendix A - Group

<*Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist in determining the effort put forth to produce this document*>