VIT®
UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)
VELLORE ▪ CHENNAI
www.vit.ac.in

Project Report

for

Request Based Grouping System

Version 1.0

Prepared by

Nikita Negi	**16BCE2038**	**nikita.negi2016@vitstudent.ac.in**
Palash Golecha	**16BCE0716**	**palash.golecha2016@vitstudent.ac.in**
Mehak Malik	**16BCE0802**	**mehak.malik2016@vitstudent.ac.in**

Instructor: **Prof. Karthikeyan T.**

Course: **Software Engineering (CSE3001)**

Lab Section: ***L35+L36***

Content

1 Introduction

1.1 Document Purpose

The purpose of the document is to gather and formalize all requirements of the system. In this document all ideas related to the system will be collected and analyzed to define the system, with the aim to finalize the full specification of the system that will be delivered later. This document will primarily server as a reference document for developing the initial version of system, this will be the go to document in cases where complications or disagreements arise.

The document will also explain system limitations, interface and interactions with user or other external applications. In short, the purpose of this SRS document is to provide a detailed overview of our software system. This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client, team and audience see and interact with the product and its functionality.

1.2 Product Scope

Most activities today in VIT or other colleges require groups of people. We work in teams whether it is for CAL projects or from starting research in our respective fields. Even sharing of cabs or autos requires finding people with the same requirements to reduce costs and increase safety and security.

Our software system aims to provide a forum to form these groups and make division of money and resources very easy. The uniqueness of this system is that it is not limited to one group activity. The same software can be used to form groups to share cabs as well as form project groups. It will group people with matching requirements and display the results to the individual. Then if both parties consent to exchange their contact details, then the details are shared between the involved parties. It also divides any financial cost required between the parties.

1.3 Intended Audience and Document Overview

The scope of our project is limited to VIT as we are trying to provide solutions to problems faced by students in VIT with respect to finding project partners and members to share cabs, etc. The project will mainly cater to the needs of students of VIT, specifically developers in need of partners for projects, research scholars looking to find experts in the similar field, professionals looking for opportunities or even people looking for partners for any organised outdoor activity such as dinner, cab sharing etc.

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter. General description of product, including user interface screen shots, dependency analysis, and enterprise requirements.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product. Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

The fourth chapter defines any non-functional requirement that might be needed while framing the structure of the project and finally the fifth chapter defines any other requirements that might have been missed during the course of framing the document.

Appendixes in the end of the document include the all results of the requirement prioritization and a release plan based on them.

1.4 Definitions, Acronyms and Abbreviations

USER - A non-admin user of the application
GM – Group Moderator - A non-admin user who has posted a group request, therefore becoming the effective moderator of that particular group.
ADMIN- The main admininstrator.
SPOTLIGHT- The first information on the landing page.

1.5 Document Conventions

1.5.1 Formatting Conventions

In general this document follows the IEEE formatting requirements. It uses Arial font size 11, or 12 throughout the document for text. It uses italics for comments. Document text is single spaced and maintains 1" margins. Number notation is used to refer to text.

1.5.2 Naming Conventions

All diagrams/figures are provided with a caption including a number and title. Every table in the document has a number and title placed above the table.

1.6 References and Acknowledgments

[1] IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements

Specifications. IEEE Computer Society, 1998.

[2] S. Al-Fedaghi, ―Flow-based description of conceptual and design levels‖, IEEE International Conference on Computer Engineering and Technology 2009, January 22–24, 2009. Singapore.

[3] S. Al-Fedaghi, ―Scrutinizing UML activity diagrams‖, 17th International Conference on Information Systems Development, Paphos, Cyprus, August 25–27, 2008.

[4] S. Al-Fedaghi, ―Software engineering interpretation of information processing regulations‖, IEEE 32nd Annual International Computer Software and Applications Conference, Turku, Finland, July 28–August 1, 2008.

[5] V. De Castro, J. V. Vara, and E. Marcos, ―Model transformation for service-oriented Web applications development‖, in Workshop Proceedings of 7th International Conference on Web Engineering, July 20, 2007, 184–198.
[6] EclipseCon, ―Modeling Web applications: Detailed description‖, (2009, access date). http://www.eclipse.org/m2m/atl/usecases/webapp.modeling/description.php#using.model.annotation

[7] H. Foster, H. S. Uchitel, J. Magee, and J. Kramer, (2003) ―Model-based verification of Web service compositions‖, 18th IEEE International Conference on Automated Software Engineering, Montreal, Canada. 2003, http://www.doc.ic.ac.uk/~hf1/phd/papers/ase03_long_foster_h.pdf

[8] F. Leymann, ―Web services: Distributed applications without limits‖, in Proc. BTW'03 (Leipzig, Germany, February 26–28, 2003), Lecture Notes in Informatics, vol. P-26, Gesellschaft fuer Informatik (GI), Bonn, Germany.

[9] M. P. Papazoglou, and D. Georgakopoulos, ―Serviced-oriented computing‖, Communications of ACM, 46 (2003), 10, 25–28

[10] J. M. Vara Mesa, ―ATL/AMW use case modeling Web applications: Detailed description and user guide‖, (2009, access date),
http://www.eclipse.org/m2m/atl/usecases/webapp.modeling/resources/User.Guide.pdf

2 Overall Description

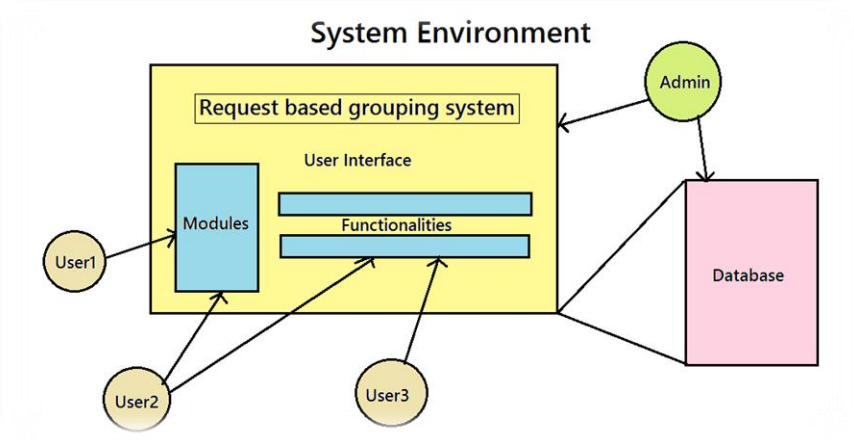
This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe what type of stakeholders that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

2.1 Product Perspective

This product is a stand-alone project and is the first version of its type. The Request based Grouping System is a virtually self-contained module; however, it will require users to have access to a web browser on their

workstation computer. This means that the users of the system do not need to invest in any other software to get the most out of the software system as any Windows based PC comes installed with a web browser, and any non-Windows machine can use FireFox or other freeware browsers. For the user interface the user only needs to open the web application on the browser, using a system with steady internet connection, where they can put up their requests. Other than that, there are no hardware interfaces required in it.

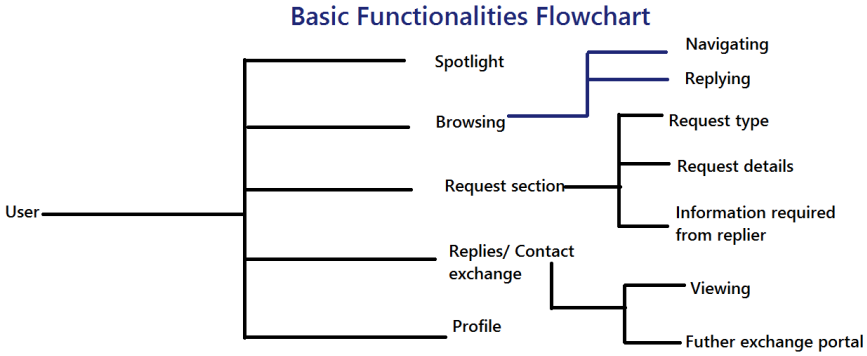
Since this is a data-centric product it will need somewhere to store the data. For that, a database will be used. The web portal will communicate with the database. All of the database communication will go over the Internet.



2.2 Product Functionality

The final product will have much functionality to streamline the process of group formation. The functionalities will enable the application to collect as much data as needed to produce a more targeted and user specific result. The functionalities are as follows:

- The application has a unique profile for each user and there they can fill up basic information that might be helpful for them in future endeavours.
- The landing page will have a spotlight section where any current or popular requests will be displayed for the user on a random basis.
- The browsing section will enable the user to see all the upcoming requests in all the categories but here the requests can be filtered as per the wish of the user. The user can also attempt to reply to any lucrative offers that he/she might come across.
- The request section enables a user to put out request of their choice. The request section will require information about the request type, various details about the request like capacity, time, date, place, special requests from interested replier etc.
- The replies section is where the user can view all the people who are interested in the request that was put out by the user. Seeing the automated generated and manually entered replies, the user can wish to extend the communication. The users can now exchange contact details once both the counterparts send their consent signal.



2.3 Users and Characteristics

There are three types of people that will be manipulating and dealing with the application. They are the unauthorized users, authorized users and the administrator.

The unauthorized user will be having minimum access to the portal. They can only open the landing page of the application but will have to create an account if they wish to access more of the application.

The major traffic will constitute of the authorized users. They will have full access to all the functionalities of the application and they will populate and interact with the system. They only need to be comfortable to work with a rather user friendly graphical user interface which requires basic computer knowledge and internet ethics. They will have access to all major sections such as spotlight, browsing section, request, reply, profile etc. However these users will not have any access to the backend and the database that forms the base of the grouping system.

Finally, the system administrators are users who are able to setup the system from the initial installation and maintain the systems user accounts. They automatically have the functionality of authorized users within the normal operation of the system, however have additional menu options which allow them to maintain the system. They can manipulate the framework of the application and can also access the database.

2.4 Operating Environment

There is no dedicated software that needs to be installed inorder to use the system defined and also there is no external hardware component needed. The Request based Grouping System is a self-contained module; however, it will require users to have access to a web browser on their workstation computer. This means that the users of the system do not need to invest in any other software to get the most out of the software system as any Windows based PC comes installed with a web browser, and any non-Windows machine can use FireFox or other freeware browsers. Any GUI, IOS, Windows or android based workstation can be used, be it PC, smartphones or laptop etc. For the user interface the user only needs to open the web application on any standard browser (there is no constraints here), using a system with steady internet connection, where they can put up their requests. Other than that, there are no hardware interfaces required in it.

2.5 Design and Implementation Constraints

- The usage of the system depends on an active internet connection and cannot be use offline or in the manner of an in-device application.
- The counterparts in a group cannot interact with one another unless both the parties consent on giving personal contact information to one another, no matter how eager one of the counterpart is.
- There is no provisional feature of direct communication between users unless they partake in an activity together.
- System is to be developed for distributed use as a web application. This will limit the ability for real time updates to the system.
- Data must be stored in a relational database for quick queries and storage.
- System must maintain correct information all the time
- The application is only available in English language.
- The application system can only generate groups or applicants if there are enough people as stated by the request. Partial requests might not be processed and the result might not always be instant.

2.6 User Documentation

Inorder to operate the project module, the user just needs a basic idea as to how to operate the graphical user interface of a web application. The interface is user friendly and does not require dedicated guide books and tutorials. If any complexity arises, the viewer can always contact the developers via the contact mentioned.

2.7 Assumptions and Dependencies

One assumption about the product is that it will always be used on mobile phones and workstations that have enough performance. If they do not have enough hardware resources available for the application, for example the users might have allocated them with other applications, there may be scenarios where the application does not work as intended or even at all. Another assumption is that the modular components in all devices work in the same way. If the phones have different interfaces to front end, the application need to be specifically adjusted to each interface and that would mean the integration with the front end would have different requirements than what is stated in this specification.

3 Specific Requirements

3.1 External Interface Requirements

This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

3.1.1 User Interfaces

The first time a user opens the website he should see the description of the system with a sidebar containing the login modal, all of the core functionality requires authentication.
If the user is not registered, he/she should be able to do that after clicking the register button on the login modal. When the user is logged in and is authenticated, he is redirected to the spotlight which is a central hub to all the functionalities of the application. In all screens after logging in there will be a common navigation bar providing links to all the functionalities of the app, the leftmost section of the dashboard will be the user section where a user can find links to view profile, view his/her settings and log out.

Spotlight -
The spotlight serves as a hub for all of the functionalities, it will show a quick status update on the all your activities. It shows the status of your projects and the requests you have received. The page also highlights trending projects and allows you to have a overview of what is happening in the application

Profile

Every user has a profile page where the basic information along with the projects they are working on are displayed. In addition to the basic info, specific information such as links to the github account or the linkedin account may as well be provided by the user.

Browse Projects

The browse projects view will provide a searchable interface to all the active projects.
The user interface will have a search bar on top followed by filter options in the sidebar. The filter options will allow user to narrow down the project that they might find interesting.

Create Project

This will provide an interface to to publish a new project in which group members are needed.
The interface will ask for the details about the project in a form like view. User is supposed to enter all paramaters about the project. Examples of the form fields could be name of the project, category, group members needed etc.

Request to join

After finding a project a user can apply to be a part of it. For this interface a popup modal will appear when the user clicks to join a projects. The popup modal will have a fields asking about the user and why do they want to join the project. Clicking submit will send the request to the GM.

Request Management

The GM will have a request management interface containg all the requests. The GM will be first asked to select a project if he is managing multiple of them. After selecting a project the GM will get a tabular paginated view of all the requests that the project has received. The GM can indiviually see all the applications that the user has submitted. After reviewing the application, the GM can decide to approve or deny and two buttons for the same will be provided to him/her.

Messaging

Users can chat and communicate with other users, to do so, one may click the Message icon on one's profile page which will open the chat interface.The chat interface will have all the active chats in the sidebar, clicking on them will lead to opening of chat in the main view. The user can than type his message in the textbox and sent it.

3.1.2 Hardware Interfaces

The web application which is accessible by a simple web browser does not have any designated hardware, it does not have any direct hardware interfaces, a normal mouse and keyboard is enough to navigate and use the interfaces. Other hardware interfaces that are required are a network connection which the software doe'snt need to bother about, the underlying operating system takes care of it.

3.1.3 Software Interfaces

The client side is interfaced via the web browser, the type of web browser doesn't matter as long as it follows the HTML5 RFC specification. Due to its nature, the application is OS-neutral and works cross platform.Other software libraries such as animate.js and jquery.js will be used in order to facilitate the quick development process.*On the server side, the application use django-rest-framework , a software framework used to create REST API. The django-rest-framework works on top of another software framework called django which is python framework to create backend applications.*

3.1.4 Communications Interfaces

The web application communicates to the server via the HTTP protocol interface which works on another protocol suite of TCP/IP . Ocassionaly WebRTC protocol might be used to facitilate bi-directional instantaneous communication with low overhead.

3.2 Functional Requirements

This section includes the requirements that specify all the fundamental actions of the software system.

3.2.1 User Class 1 - The User

3.2.1.1 Functional requirement 1.1

I**D: FR1**
TITLE: Globally accessible application
DESC: Any user wanting to use the application should be able to use it without any obstruction.
Thus the application should be accessible through a memorisable domain name registered on the internet.
DEP: None

3.2.1.2 Functional requirement 1.2

I**D: FR2**
TITLE: User Registeration
DESC: Given that user can access the website, he/she should be able to register themselves on the website filling basic contact information such as name, phone number, email, college , city and password.
DEP: FR1

3.2.1.3 Functional requirement 1.3

I**D: FR3**
TITLE: User Login
DESC: Given that user registered on the application, he/she should be able to login and access the features of the application. The login information should be persisted across session via cookies or any other session mechanism.
DEP: FR1,FR2

3.2.1.4 Functional requirement 1.4

I**D: FR4**
TITLE: User Reset Password
DESC: Given that user has registered on the application , in circumstances such as when the user has forgot his credentials he/she should be able to retrieve a reset link on their registered email.
DEP: FR3

3.2.1.5 Functional requirement 1.5

I**D: FR5**
TITLE: User profile page
DESC: Each user of the application will have their own profile page detailing all the information about them. Such a page will be helpful when a user decides to review requests to join their project.
DEP: FR3

3.2.1.6 Functional requirement 1.6

I**D: FR6**
TITLE: Browsing projects
DESC: A user should be able to browse different projects on the application,he should also be able to filter them by his/her preferences.

DEP: FR3

3.2.1.7 Functional requirement 1.7
I**D: FR7**
TITLE: Searching projects
DESC: A user should be able to do a full text search on all projects including their name , descriptions.
DEP: FR3

3.2.1.8 Functional requirement 1.8
I**D: FR8**
TITLE: Request collaboration
DESC: After finding a suitable project a user should be able to request to join the group. This request should go to the group admin
DEP: FR3

3.2.1.9 Functional requirement 1.9
I**D: FR9**
TITLE: Submitting a project
DESC: If the user needs to form a group for a project, he should be able to post the details and allow interested people to apply. This is the central function of the application.
DEP: FR3

3.2.1.10 Functional requirement 1.10
I**D: FR10**
TITLE: Communication between users.
DESC: Communication is the basis of all group projects, therefore the app must facitilate communation between the users. When interested people apply to join a project, it is important to have a robust communication interface to allow GM to know the other person better.
DEP: FR3,FR8/FR9

3.2.1.11 Functional requirement 1.11
I**D: FR11**
TITLE: Request management
DESC: The GM should have the functionality to approve/deny requests to join the group. He should also be able to specify the reason for the same
DEP: FR3,FR9

3.2.1.12 Functional requirement 1.12
I**D: FR12**
TITLE: Logging out
DESC: Authenticated user should be able to log out of the system to destroy the active session when the user wishes.
DEP: FR3

3.2.2 User Class 2 - The Admin

3.2.2.1 Functional requirement 2.1
I**D: FR13**
TITLE: Creating / Updating / Deletion User

DESC: The admin use should be able to manually create a user, update a user's detail, such as in case of a hard-reset is required or even delete nefarious users.
DEP: NONE

3.2.2.2 Functional requirement 2.2

ID: FR14
TITLE: Updating / Deleting Group Requests
DESC: The admin use should be able to manually a group request's detail, or delete a group request
DEP: NONE

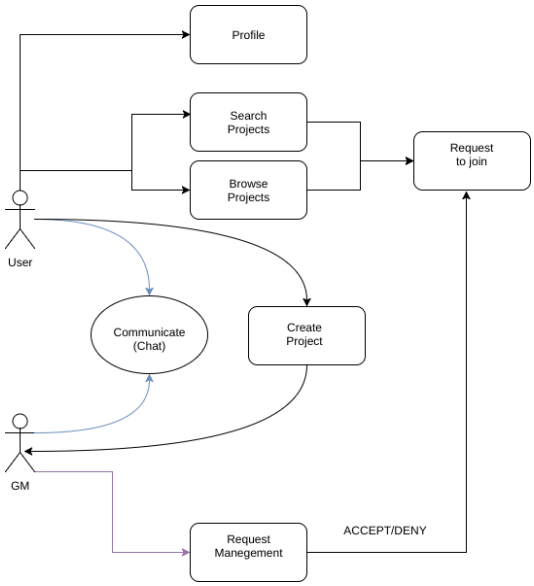
3.2.2.3 Functional requirement 2.3

ID: FR15
TITLE: Data statistics and raw data access.
DESC: The admin use should be able to see the statistics of each and every parameter of the site in order to adminster it. He should be able to see all kind of data that exists on the system in its raw form.
DEP: NONE

3.3 Behaviour Requirements

3.3.1 Use Case View

This is a basic diagram that encapsulate all the functionalities and actors for the system. Each user has a profle and can search for projects etc, They can interact with the group moderator of the group and work effectively only when both parties give confirmation.



4 Other Non-functional Requirements

4.1 Performance Requirements

1. The system should allow different users to send requests concurrently.
2. It should not take more than 5 seconds for the request to be stored on the database.
3. If the software loses connection to the Internet on the system, the user must be informed of the same.
4. User must be able to choose among a range of activities.
5. User must be able to a request even if other requests are pending.
6. User should be able to edit their account information after logging in.
7. If a user tries to log in to the system with a non-existing account then the admin should not be logged in. The user should be notified about log-in failure

4.2 Safety and Security Requirements

1. Contact information of user will not be given to any other user without consent from concerned user.
2. Once consent is given by a particular user to share contact information
3. Every user account will be identified with a username and protected with a password consisting of minimum 6 characters with at least one number and special character.
4. No images shall be shared on the software.
5. The server on which the software data resides will have its own security to prevent unauthorized read/write access.

4.3 Software Quality Attributes

4.3.1 Adaptability

This software will be adaptable, as it will allow user to not just find partners for CAL projects, but also other group activities like cab sharing.

4.3.2 Portability

The software will be portable on iOS and Android smartphones, as well as laptops of windows or Linux based operating systems.

4.3.3 Maintainability

The software will be ready to accept future changes and improvements in design, or backend to implement additional functionalities.

4.3.4 Testability

Test environments will be built for the application to allow testing of the applications different functions.

4.3.5 Correctness

The software will ensure correctness of information that is displayed to each user. The contact information of the user must match exactly with the information the user has entered.

4.3.6 Flexibility

The software will be flexible, as it will be used for all miscellaneous group activities.

5. Data Dictionary

	Name	Description
1.	User	Anyone who uses the application.
2.	Admin	The creator of the application having total control over the frontend and backend of the application.
3.	Database	Collection of information of users
4.	Spotlight	First window seen on landing page
5.	Browsing/ Navigating	Shuffling through the group requests
6.	Request Section	Posting users' needed requests
7.	Contact exchange	Consensual exchange of contact between two interested people
8.	Profile	Unique profile for each user containing their basic records
9.	GM	Group moderator
10.	Further exchange portal	Space that will enable two parties to exchange contact information.
11.	Replies	Replying to group requests.

6. Design Introduction

6.1 Purpose

This software design specification document will detail the implementation of the requirements as defined in the Software Requirements Specification – Request Based Grouping System – Version1. It lists all the design perspective of the project, analyzed to optimize the process of software development. In short, the purpose of this SRS document is to provide a detailed overview of our software system based on design. This document describes the project's target audience and its user interface. It defines how our client, team and audience see and interact with the product and its functionality.

6.2 System Overview

Most activities today in VIT or other colleges require groups of people. We work in teams whether it is for CAL projects or from starting research in our respective fields. Even sharing of cabs or autos requires finding people with the same requirements to reduce costs and increase safety and security.

Our software system aims to provide a forum to form these groups and make division of money and resources very easy. The uniqueness of this system is that it is not limited to one group activity. The same software can be used to form groups to share cabs as well as form project groups. It will group people with matching requirements and display the results to the individual. Then if both parties consent to exchange their contact details, then the details are shared between the involved parties. It also divides any financial cost required between the parties.

6.3 Design Map

The entire document is divided into 7 major sections. The section 1 gives an overview of the project and states the objective of the design. Section 2 defines under what constraints, in which environment and how the project is being developed. Section 3 states the architecture that provides the top level design view of a system and provides a basis for more detailed design work. Section 4 states the database design and components of the main database of this system. Section 5 defines the flow of different modules of the project. Section 6 provides low-level design descriptions that directly support construction of modules. Normally this section would be split into separate documents for different areas of the design. Section 7 provides user interface design descriptions that directly support construction of user interface screens.

6.4 Definitions and Acronyms

USER- A non-admin user of the application
GM– (Group Moderator) A non-admin user who has posted a group request, therefore becoming the effective moderator of that particular group.
ADMIN- The main administrator.
SPOTLIGHT- The first information on the landing page.

7. Design Considerations

7.1 Assumptions

One assumption about the product is that it will always be used on mobile phones and workstations that have enough performance. If they do not have enough hardware resources available for the application, for example the users might have allocated them with other applications, there may be scenarios where the application does not work as intended or even at all. Another assumption is that the modular components in all devices work in the same way. If the phones have different interfaces to front end, the application need to be specifically adjusted to each interface and that would mean the integration with the front end would have different requirements than what is stated in this specification.

7.2 Constraints

- The usage of the system depends on an active internet connection and cannot be use offline or in the manner of an in-device application.
- The counterparts in a group cannot interact with one another unless both the parties consent on giving personal contact information to one another, no matter how eager one of the counterpart is.
- There is no provisional feature of direct communication between users unless they partake in an activity together.
- System is to be developed for distributed use as a web application. This will limit the ability for real time updates to the system.
- Data must be stored in a relational database for quick queries and storage.
- System must maintain correct information all the time
- The application is only available in English language.
- The application system can only generate groups or applicants if there are enough people as stated by the request. Partial requests might not be processed and the result might not always be instant.

7.3 System Environment

There is no dedicated software that needs to be installed inorder to use the system defined and also there is no external hardware component needed. The Request based Grouping System is a self-contained module; however, it will require users to have access to a web browser on their workstation computer. This means that the users of the system do not need to invest in any other software to get the most out of the software system as any Windows based PC comes installed with a web browser, and any non-Windows machine can use FireFox or other freeware browsers. Any GUI, IOS, Windows or android based workstation can be used, be it PC, smartphones or laptop etc. For the user interface the user only needs to open the web application on any standard browser (there is no constraints here), using a system with steady internet connection, where they can put up their requests. Other than that, there are no hardware interfaces required in it.

7.4 Design Methodology

The process model that we have chosen is the Entry-Task-Validation-Exit (ETVX). This model is the most suitable for our system, the Request Grouping System because unlike other models like the Waterfall model we can revisit the tasks involved in a process if they do not produce the required result. Especially with our software system, which is based on the interaction of people and how to make that interaction secure and as per requirement, we constantly need to verify our or result and see if it provides the required functionality. Also, this model specifies entry and exit level criteria, which ensure that certain requirements are achieved before moving to the next phase.

7.5 Risks and Volatile Areas

One risk factor might be about the product is that it will always be used on mobile phones and workstations that have enough performance. If they do not have enough hardware resources available for the application, for example the users might have allocated them with other applications, there may be scenarios where the application does not work as intended or even at all. Another volatility is that the modular components in all devices work in the same way. If the phones have different interfaces to front end, the application need to be specifically adjusted to each interface and that would mean the integration with the front end would have different requirements than what is stated in this specification.

8. Architecture

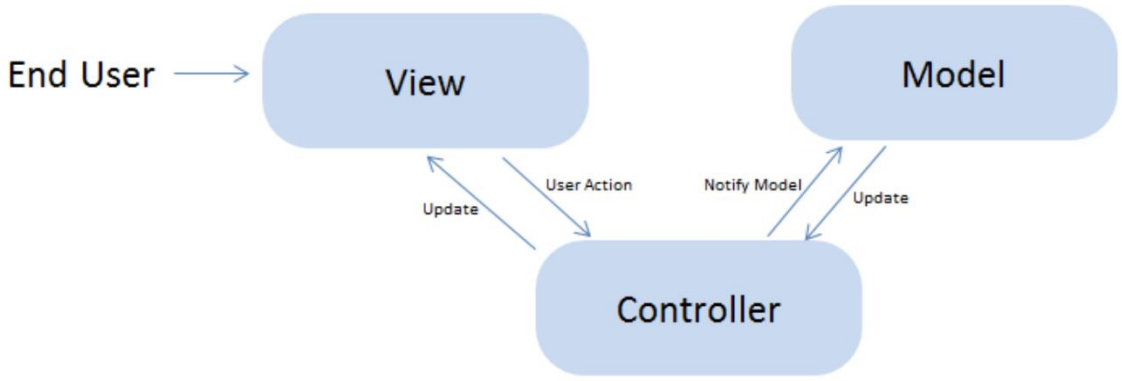
8.1 Overview

Our system is designed with extensibility and scalability in mind. We are taking great care in designing a framework which can be updated easily. Many of the anticipated changes to our system in future phases will only require adding new types of data and changing the user presentation code to make use of these new data. The framework we have designed will only require "plugging in" these new types of data without refactoring the logic that passes the data over the network, retrieves and updates the database, etc.

MVC Style separates presentation and interaction from the system data. The system is structured into three logical components that interact with each other.

- The Model component -Manages the system data and associated operations on that data.
- The View component- Defines and manages how the data is presented to the user.

- The Controller component- Manages user interaction and passes these interactions to the View and the Model.



8.2 Subsystem, Component, or Module 1 …N

8.2.1 User Module

The user module is responsible for handling the profiles of users, their details such as contact information. It is responsible for authentication of users and hence also responsible for security.

8.2.1.1 Spotlight -

The spotlight serves as a hub for all of the functionalities; it will show a quick status update on the all your activities. It shows the status of your projects and the requests you have received. The page also highlights trending projects and allows you to have a overview of what is happening in the application

8.2.1.2 Profile

Every user has a profile page where the basic information along with the projects they are working on is displayed. In addition to the basic info, the user may as well provide specific information such as links to the Github account or the LinkedIn account.

8.2.2 Project Module

The project module is responsible for creating projects, viewing all active projects, which involves the details of

projects such as number of members, topic, subject etc.

8.2.2.1 Browse Projects

The browse projects view will provide a searchable interface to all the active projects. The user interface will have a search bar on top followed by filter options in the sidebar. The filter options will allow user to narrow down the project that they might find interesting.

8.2.2.2 Create Project

This will provide an interface to publish a new project in which group members are needed. The interface will ask for the details about the project in a form like view. User is supposed to enter all parameters about the project. Examples of the form fields could be name of the project, category, group members needed etc.

8.2.3 Request Management Module

The GM will have a request management interface containing all the requests. The GM will be first asked to select a project if he is managing multiple of them. After selecting a project the GM will get a tabular paginated view of all the requests that the project has received. The GM can individually see all the applications that the user has submitted. After reviewing the application, the GM can decide to approve or deny and two buttons for the same will be provided to him/her.

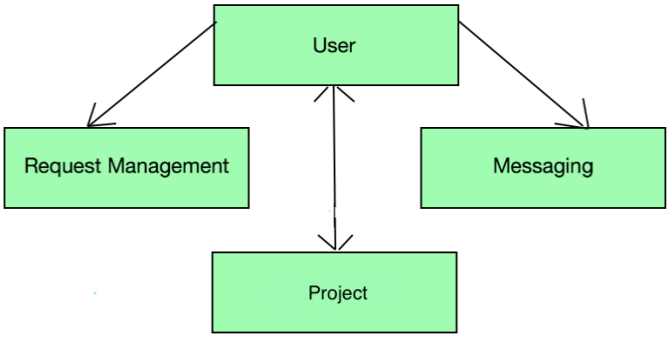
8.2.3.1 Request to join

After finding a project a user can apply to be a part of it. For this interface a popup modal will appear when the user clicks to join a projects. The popup modal will have a fields asking about the user and why do they want to join the project. Clicking submit, will send the request to the GM.

8.2.4 Messaging

Users can chat and communicate with other users, to do so, one may click the Message icon on one's profile page, which will open the chat interface.

The chat interface will have all the active chats in the sidebar, clicking on them will lead to opening of chat in the main view. The user can than type his message in the textbox and sent it.



8.3 Strategy 1…N

We will use this MVC Style for the System because, there are multiple ways to view and interact with data. Also used when the future requirements for interaction and presentation of data are unknown. In some software systems the code between the process logic and interface are mixed. This will reduce the modularity of application and make the system more difficult to maintain. To avoid this problem we have decided to use MVC architectural style to separate the application logic with the interface. The main advantage of this is style allows the data to change independently of its representation and vice versa. Support presentation of the same data in different ways with changes made in one representation shown all of them.

9.Database schema

9.1 Tables, Fields and Relationships

9.1.1 Databases

- RSystem
- Administrative

9.1.2 New Tables

9.1.2.1 RSystem
- Users
- Projects
- Requests
- Chats

9.1.2.2 Administrative
- Admin_users
- Logs
- Site_settings

9.1.3 New Fields(s)
- RSystem

Project Table

Field Name	**Field Type**	**Constraints**
project_id	Integer	Primary Key
project_name	varchar(30)	Not null
project_description	varchar(400)	Not null
project_tags	varchar(200)	Not null
project_admin	Integer	foreign key User(user_id)
project_requirements	varchar(200)	Not null
project_timeframe	varchar(50)	Not null
no_people_required	Integer	Not null

User Table

Field Name	**Field Type**	**Constraints**
user_id	Integer	Not null
Username	varchar(30)	Not null
full_name	varchar(50)	Not null
Mobile	varchar(12)	Not null
Password	varchar(50)	Not null
Email	varchar(50)	Not null
Verified	boolean	Default False
reset_token	varchar(30)	
verification_token	varchar(30)	
github_link	varchar(50)	
linkedin_link	varchar(50)	
other_link	varchar(50)	
about_bio	varchar(400)	

Request Table

Field Name	Field Type	Constraints
request_id	Integer	primary key
request_user	Integer	foreign key User(user_id)
request_project	Integer	foreign key Project(project_id)
request_status	Boolean	default False
request_user_comments	varchar(200)	not null
request_admin_comments	varchar(200)	
chat_id	Integer	not null

Additional Information:

Indexes -

1. user_id primary key for User table
2. Project_id primary key for Project table
3. Request_id primary key for request table
4. (Chat_id,message_id) primary key for Chat table
5. User_id primary key for admin_user table
6. Log_id primary key for log table
7. Settings_id primary key for settings table

Stored procedure

1. Hash_password - To generate a sha1 hash to store the password with a random salt
   Returns string
2. Verify_password - Verify a plaintext password with its hashed equivalent
   Returns boolean

Relationship

1. Project_admin in *Project* table references user_id from *User* table
2. Request_user from *Request* table references user_id from *User* table
3. Request_project from *Request* references project_id from *Project* table
4. Sender from *Chat* table references user_id from *User* table
5. User from *Log* Table references user_id from *User* table

9.1.5 All Other Changes

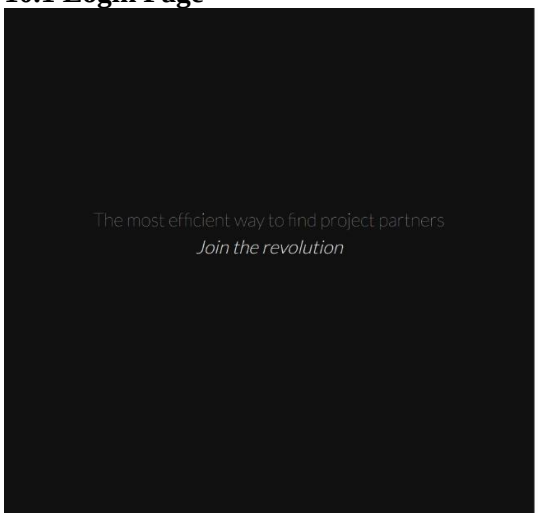
No other changes are requested.

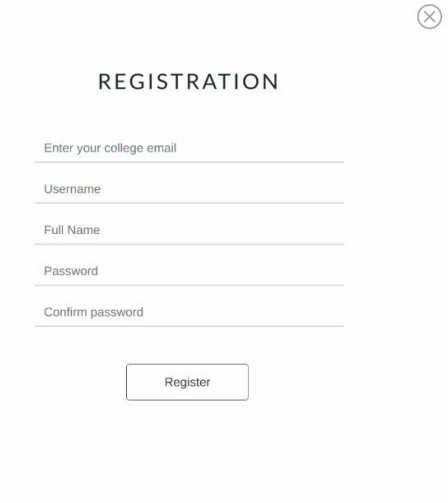
9.2 Data Migration

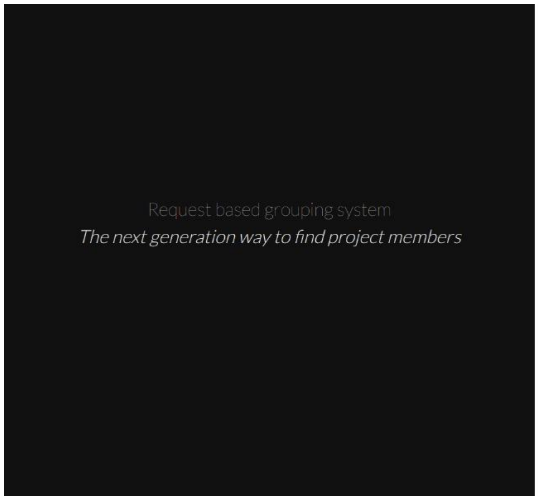
The transitions are depicted in the tables above.

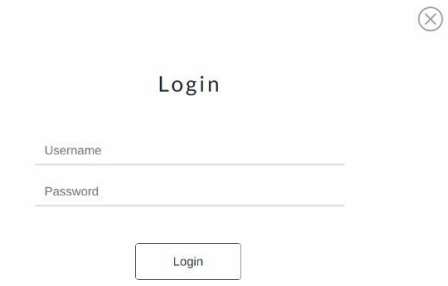
10. High level design

10.1 Login Page









Users will enter his login credentials and press submit to enter his personalized login page. The user will now have an option to create a new project or browse all active projects. The user can also request to join a particular project after login. The user can also click on the chat option to message other users regarding discussions about the project once both parties agree on it.

10.2 User Interface Modifications

As the user gives consent to the interactions, their contact of the user will be transferred to the concerned party. If no category matches the request sent by the user an alert will be displayed on the screen informing the user of the same. A list of active projects is available to the user in his/her login. The user who created the project can decide which other members will be accepted for the same.

10.3 Workflow sub-processes

Workflow sub processes will be created for each request initiated by the user. Each request must result in the storage of details regarding the particular project in the database. Also the particular project can be requested to join in by other users. Every time a new member joins the project the details of that particular project will be updated.

11. Low level design

11.1 User Module

11.1.1 Login

In the login view, username and password would be asked from the user. On clicking the submit button an ajax request (background request) will be made to the server. Upon successful authentication, the page will redirect itself to the dashboard.

10.1.2 Registration

In the the login view, below the submit button will be a registration button, on click that the login view will slide up revealing the registration form. The registration form will ask for common details like name,mobile, password etc. Username will be actively queried to the server to verify if the specific username is available. Upon passing all validation parameters, the signup button will become active. After successful registration, a verification email will be sent to the user's email to confirm the email address.

10.1.3 Forgot password

A forgot password page will cater to users incase they have forgotten their passwords. A reset link with be sent to the user's email. Upon clicking the link the user can set a new password.

10.1.4 Profile

A profile page will be assigned to every user, where all their public details could be viewed. The profile page could be accessed from clicking on the user's name anywhere it appears.

11.2 Project module

10.2.1 Browsing project

An ecommerce like process would be followed where user first select a category and view the paginated list of all process of the category. In the sidebar the filter system allows more control over the project list. Multiple filters could be combined to exactly pin down the projects required.
The user can also search for project from the top navigation bar.

10.2.2 Submitting project

Submitting a project involves filling out a form and giving the necessary details like project title, description, category, tags, no of people required. Upon filling the correct details, the next screen will be that of the project administration.

10.2.3 Project administration

The project administration dashboard will have one-stop access to all the projects. First the user is presented with the list of all projects that the user is admin of. After selecting a project the admin can edit the project details. He/she can also delete the selected project after which it would be unavailable in the project browsing.

11.3 Request Module

10.3.1 Request view
A user after browsing or searching the projects can ask to join the selected project. To do this , he can click *Request to join* button next to any project and he will be presented with a modal where he can mention additional comments as to why he/she will be a perfect fit to be in the project. After clicking submit, a request will be created and a email notification will be sent to the admin of the project notifying him of the request application. Also requesting to be a part of project will also give the admin access to contact email and phone numbers of the requesters.

10.3.2 Request Management
The admin user will have a request management view where all the requests are cumulatively shown. The admin can click on any request and view both the profile of the applicant as well the his/her comments specifying her additional comments related to project provided by the user.
The admin can then click the accept/reject icons. The accept/reject action would be accompanied by a confirmation dialog. Upon any action by the admin the same will be conveyed to the requester by both email as well as in-app notification.

11.4 Messaging

10.4.1 Start Chat
Any user can click on any user's profile and click the '*Chat*' button to start chatting to the other user. The user will be provided the start chat notification both in-app and via email. Upon click the start chat button, a chat modal will popup in the bottom.

10.4.2 Chat View
The main chat view will consist of list of chats on the sidebar as well as the selected chat opened in the main view. The selected chat will have text in bubbles with the orientation and color of bubble separating the different users. The main chat view will be accompanied by a textbox at the bottom , through which the communication text is entered.

12. User interface design

12.1 Application Controls

The user first encounters the landing page that asks the user to login or asks new users to create an account using a combination of form inputs and drop down list.
Next the user lands on the main page that has spotlight options that displays all the important messages and new updates made to the site and also a menu bar at the top which will help the user to use various modules of the application by clicking it. The user always has access to their profile, located at the top right corner of the screen where they can reconfigure their account. Once a module is chosen its screen will replace the spotlight area that was previously occupying majority of the space.

The browsing section will enable the users to navigate the new offers using the scroll bar and they can reply to it using the reply button that redirects them to a page where they can basically fill out a form requested by the application using form inputs, drop down lists etc.
The request section enables the user to put out a request. They can add a new field by clicking and can chose whether they want descriptive or multiple choice answers.

The contact exchange/ communication section will basically enable the user to see the status of their request and applications. If they wish they can chose to contact each other personally using messaging once both parties agree.

12.2 Screen 1… N

State transition diagrams of each module has been provided. However whenever the process is complex, the diagrams are accompanied by a service sequence diagram.

12.2.1 User Module

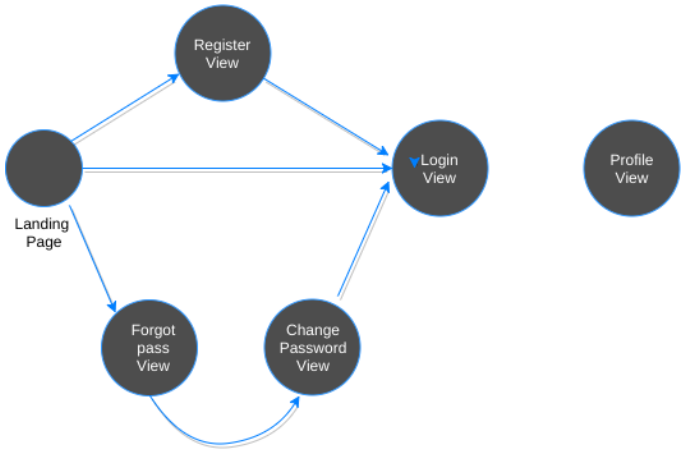


Fig 11.2.1 User module State transition diagram

The main process in user module is the **registration and authentication**

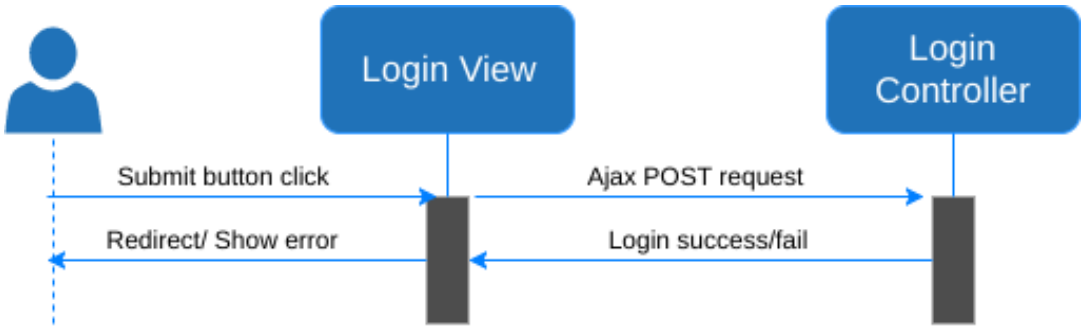


Fig 12.2.2 Loggin in Service Sequence Diagram

12.2.2 Project Module

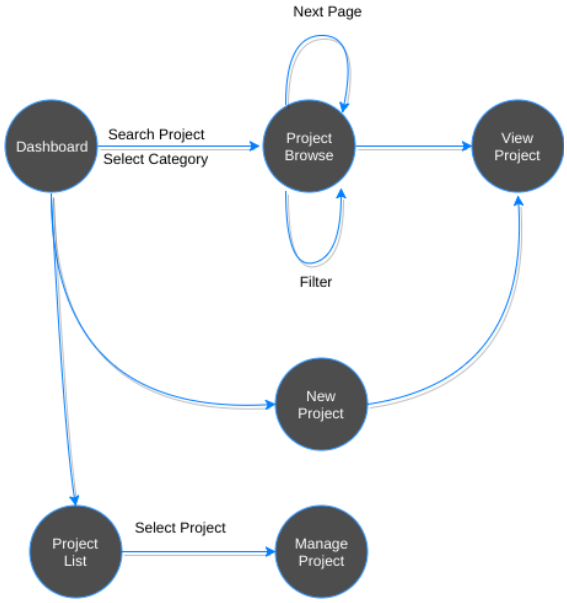


Fig 12.2.4 Project module State transition diagram

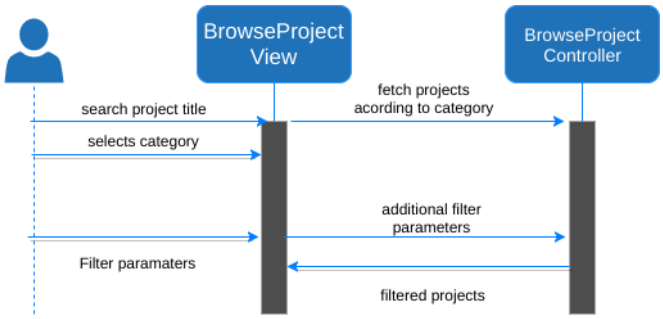


Fig 12.2.5 User module Service Sequence Diagram

12.2.3 Request Module

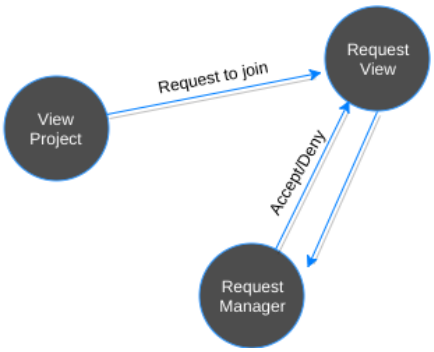
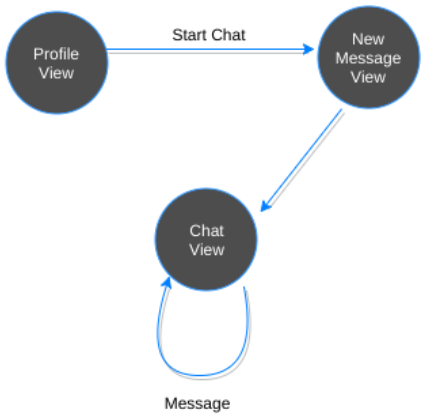


Fig 12.2.6 Request Module state transition diagram

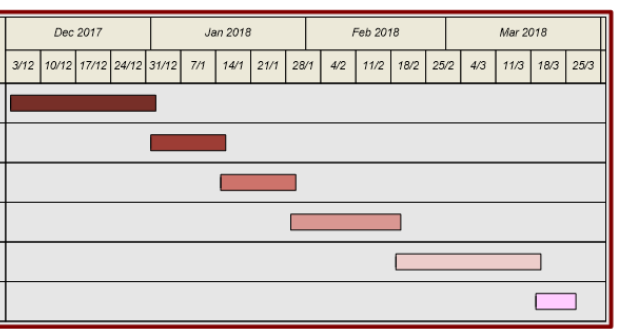


12.2.4 Chat module

Fig 12.2.7 Chat Module state transition diagram

12. Project timeline

ID	Task Name	Start	Finish	Duration
1	SRS Document	12/4/2017	1/1/2018	4w 1d
2	SDS Document	1/1/2018	1/15/2018	2w 1d
3	User Interface Development	1/15/2018	1/29/2018	2w 1d
4	Modules Development	1/29/2018	2/19/2018	3w 1d
5	Test Report	2/19/2018	3/19/2018	4w 1d
6	Deployment	3/19/2018	3/26/2018	1w 1d



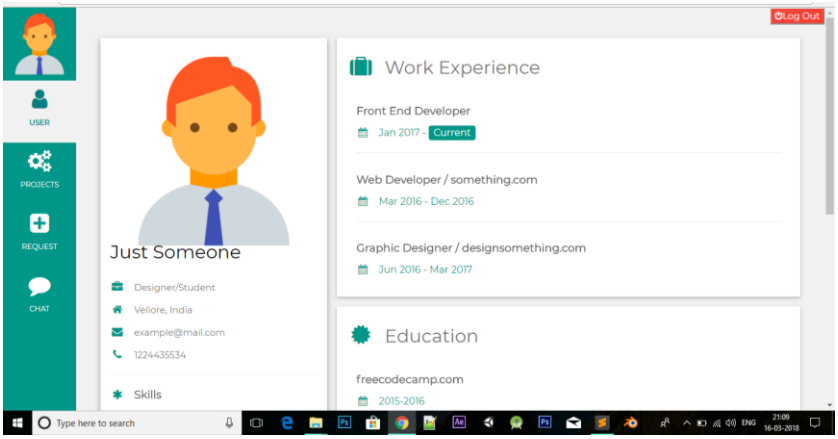
14. Testing

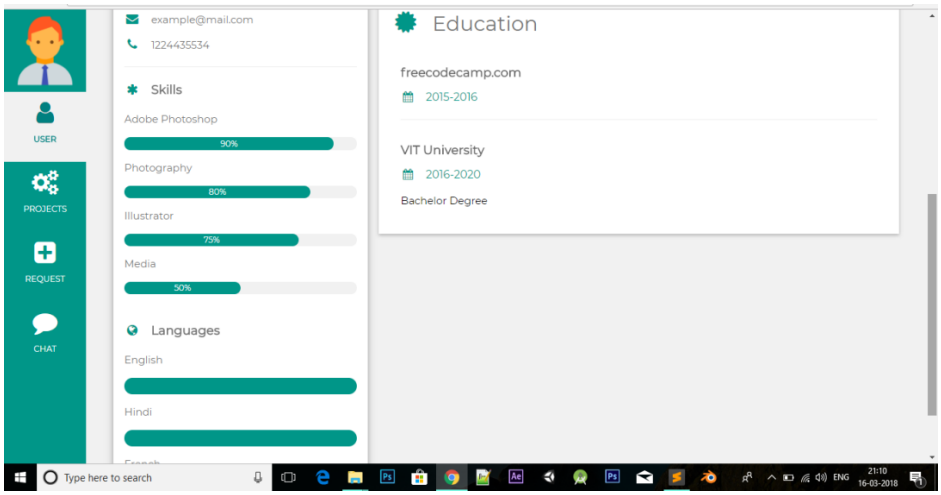
14.1 Test Cases: Module 1

1. Name: **User Module**

2. Introduction: This module is basically responsible for the login and sign up process of the users. It also deals with the profile section of the user, their credentials etc.

3. Screenshot:

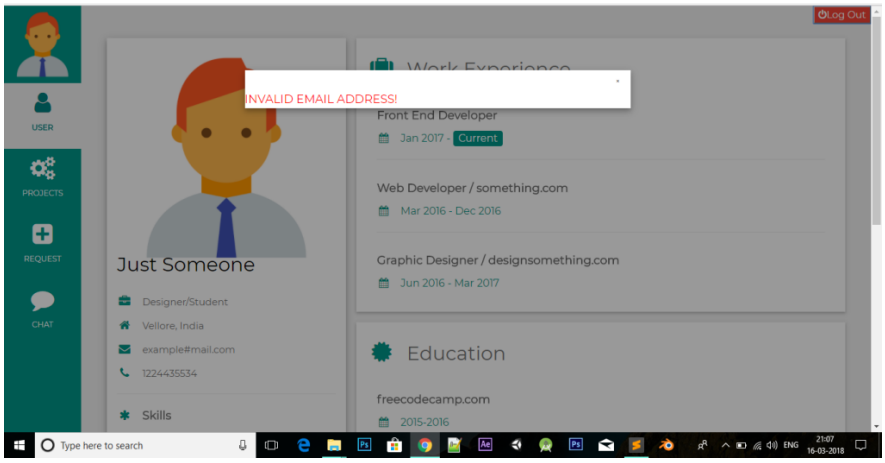




4. Test case 1:

Test Input: Adding invalid email id format that gives a pop-up to enter the mail id again redirecting them to the input form.

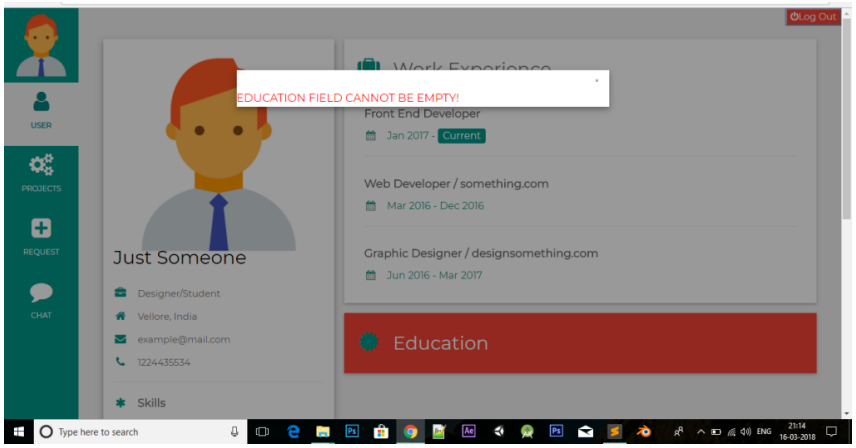
Test Output:



5. Test case 2:

Test Input: The education field is left empty which prompts the user to enter data in that field.

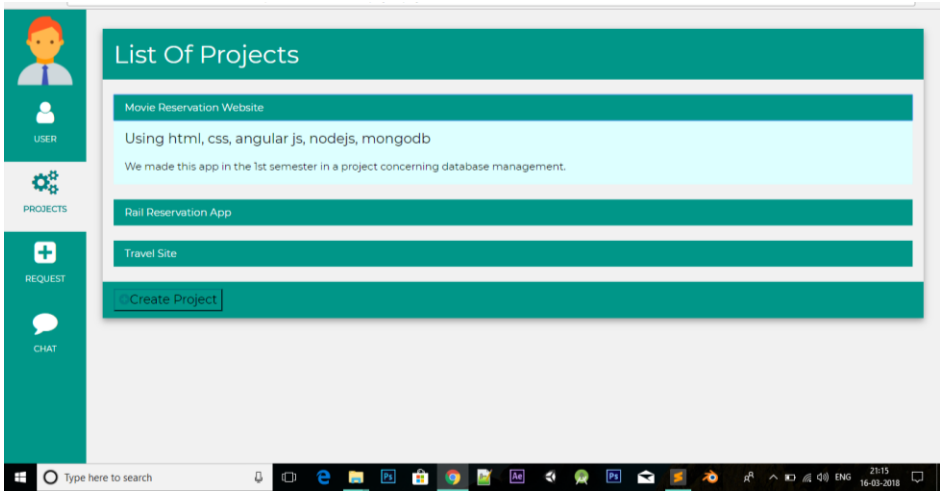
Test Output:

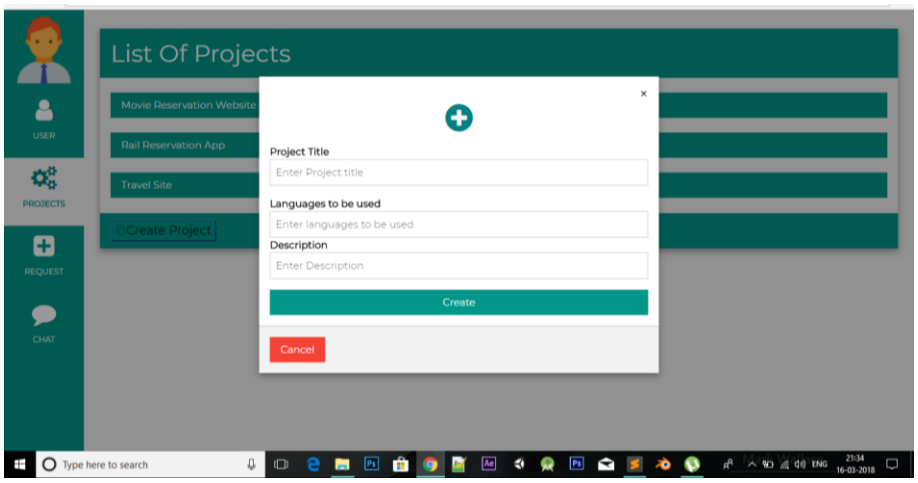


Profile Strength	
Attribute	Grade
Skills	10
Languages	8
Projects	4
Experience	6

14.2 Test Cases: Module 2

1. Name: Project module
2. Introduction: The project module is designed to show the various projects created by different users and also allows the current user to put up a project of their own.
3. Screenshot:

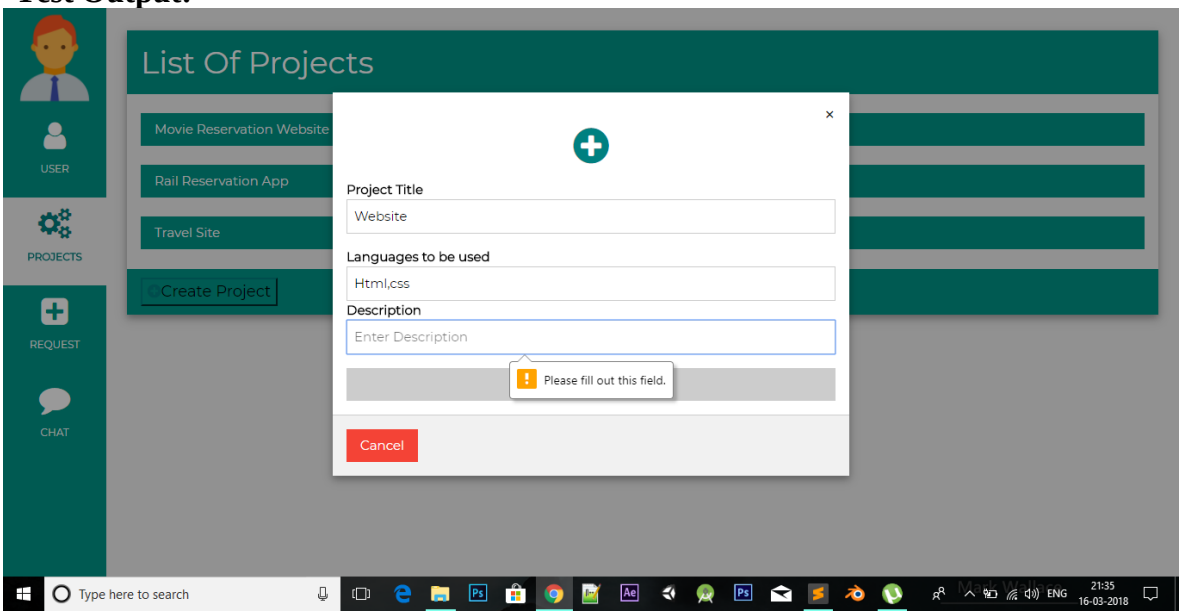




4. Test case 1:

Test Input: No fields can be left empty. This prompts the user to reenter the data.

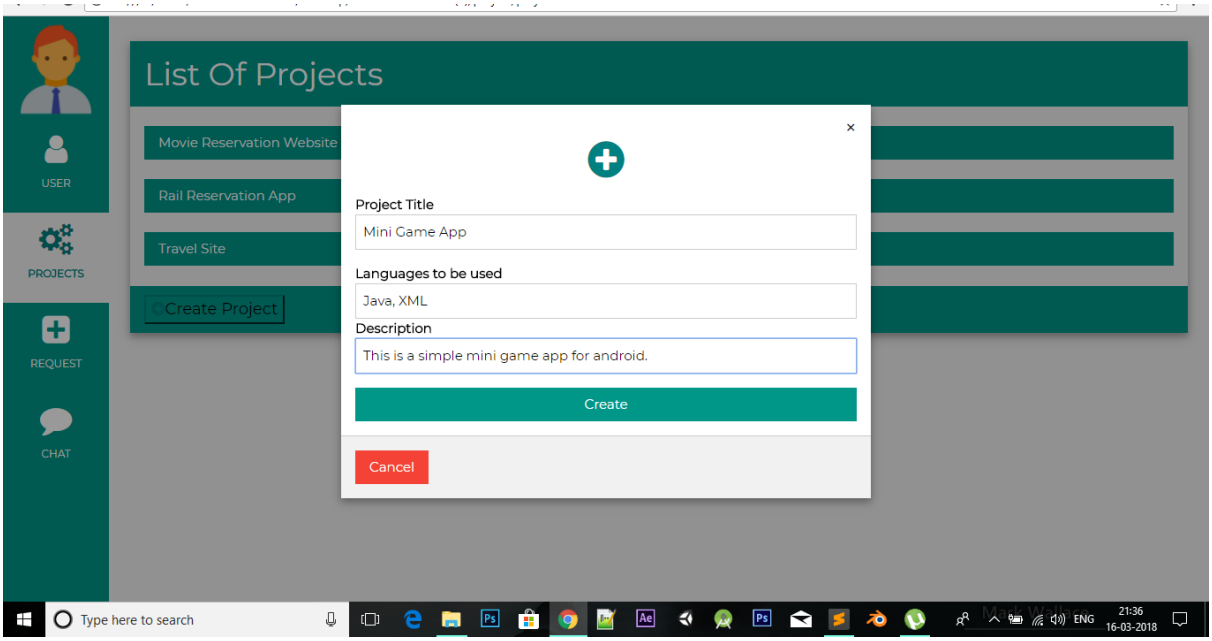
Test Output:

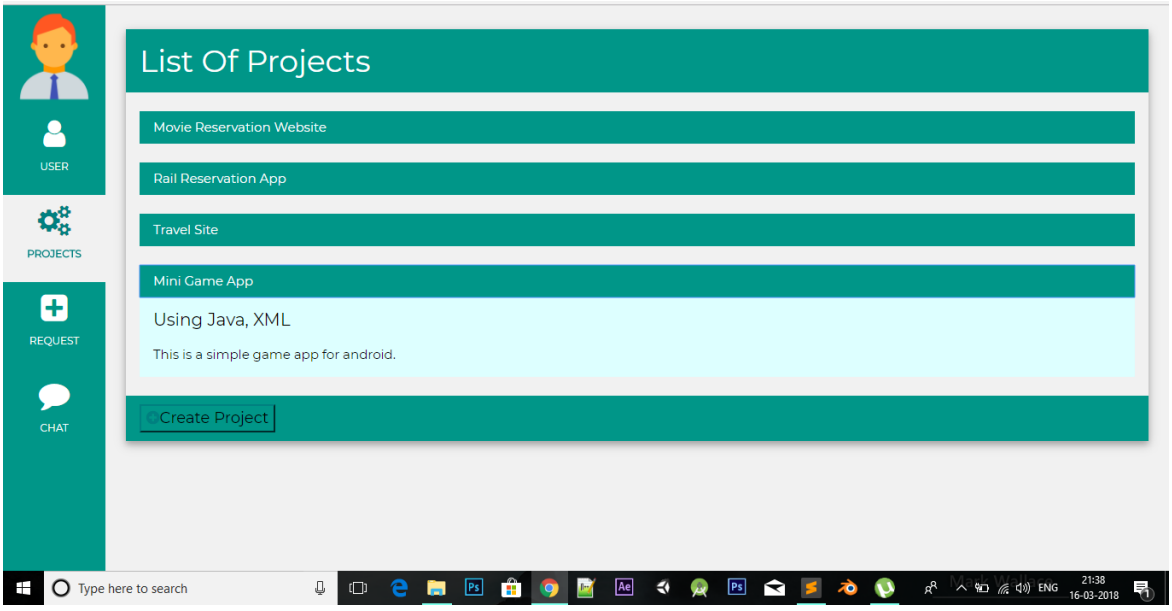


5. Test case 2:

Test Input: Testing whether on pushing the submit button, whether the project is getting added to the project list.

Test Output:

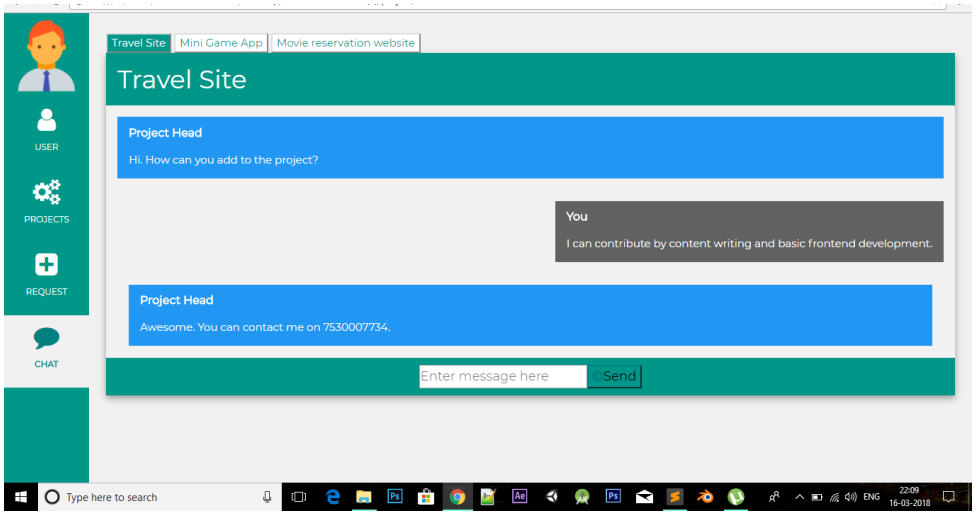




Project statistics		
Name	Views	Replies
Clean water	54	0
Cloud Server	21	13
AES algorithm	66	2

14.3 Test Cases: Module 3

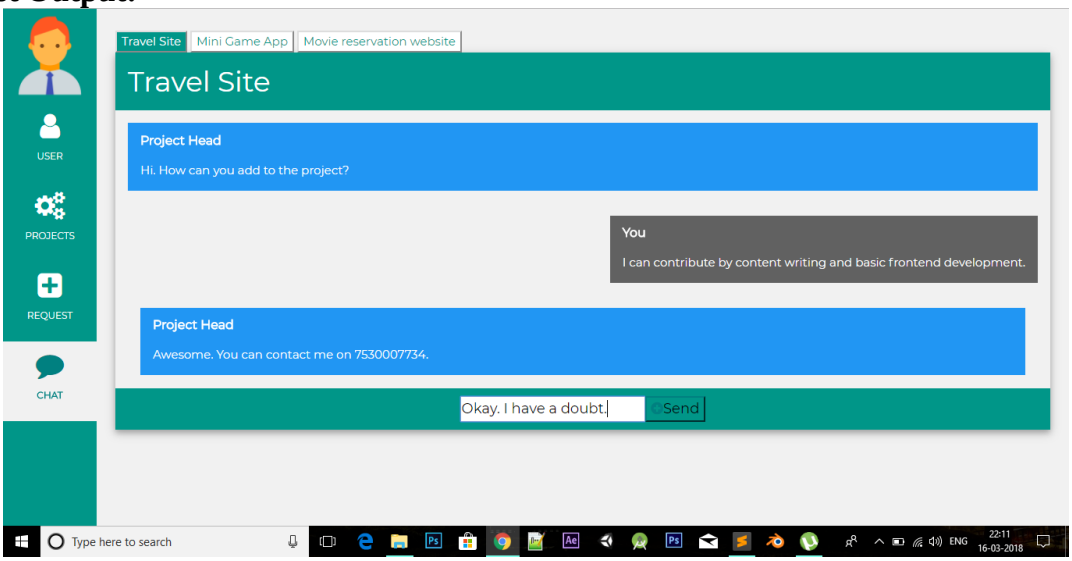
1. Name: Chat Module
2. Introduction: This module is responsible for handling the conversations between the requester and the provider of the application. This window will only be available between the parties who have both agreed on this communication.
3. Screenshot:

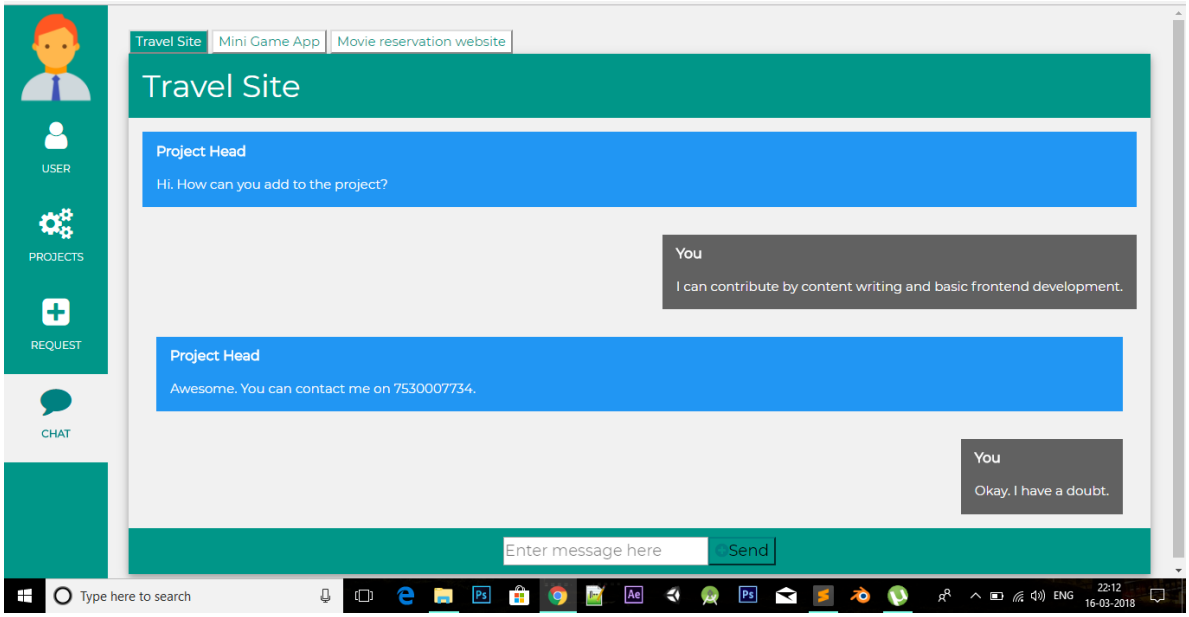


4. Test case 1:

Test Input: The messages that are "sent" are displayed on the chat screen to the other end user.

Test Output:

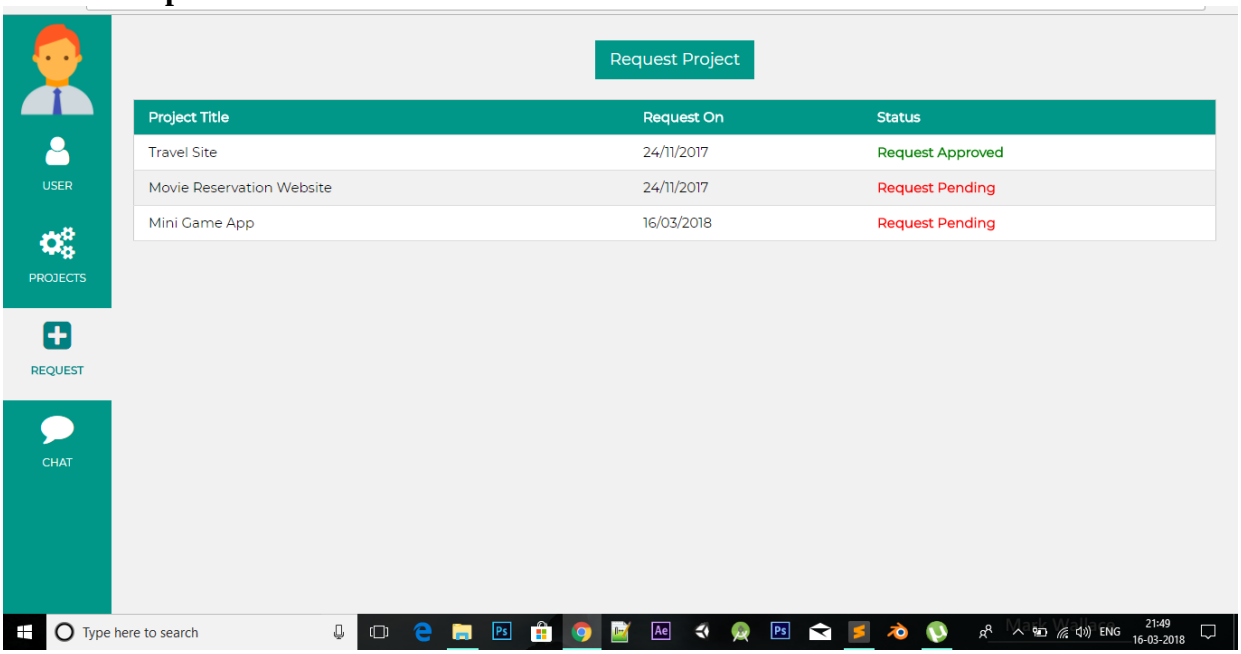


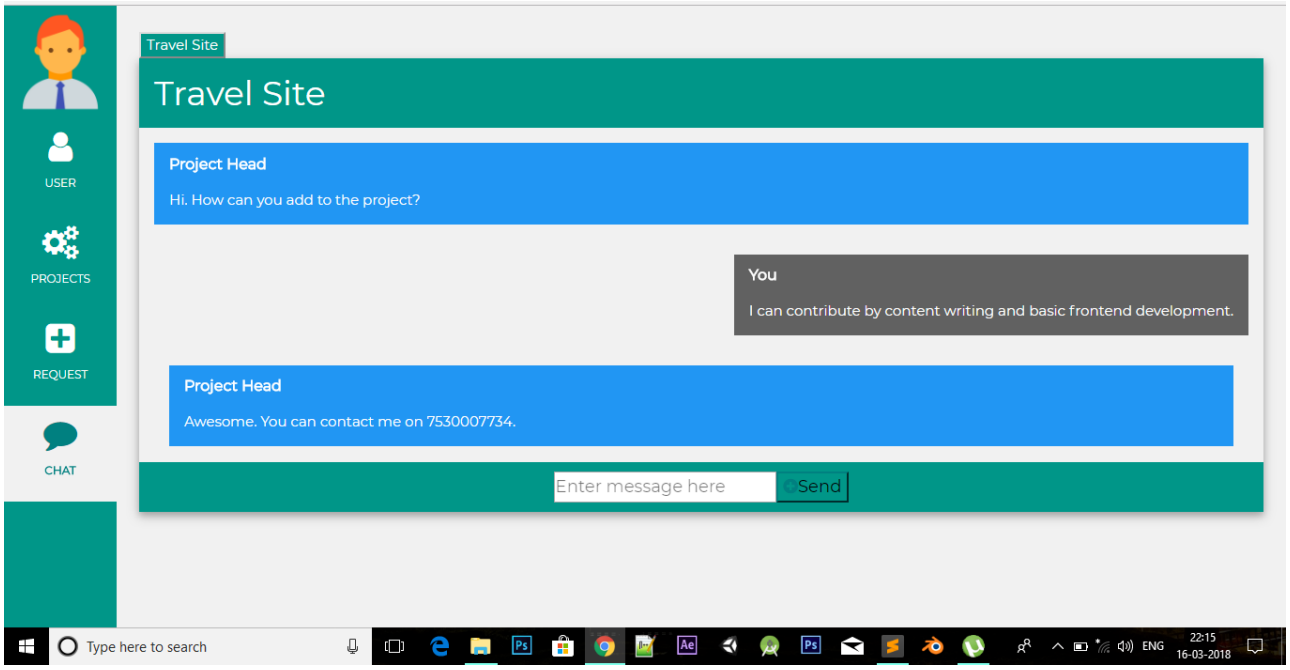


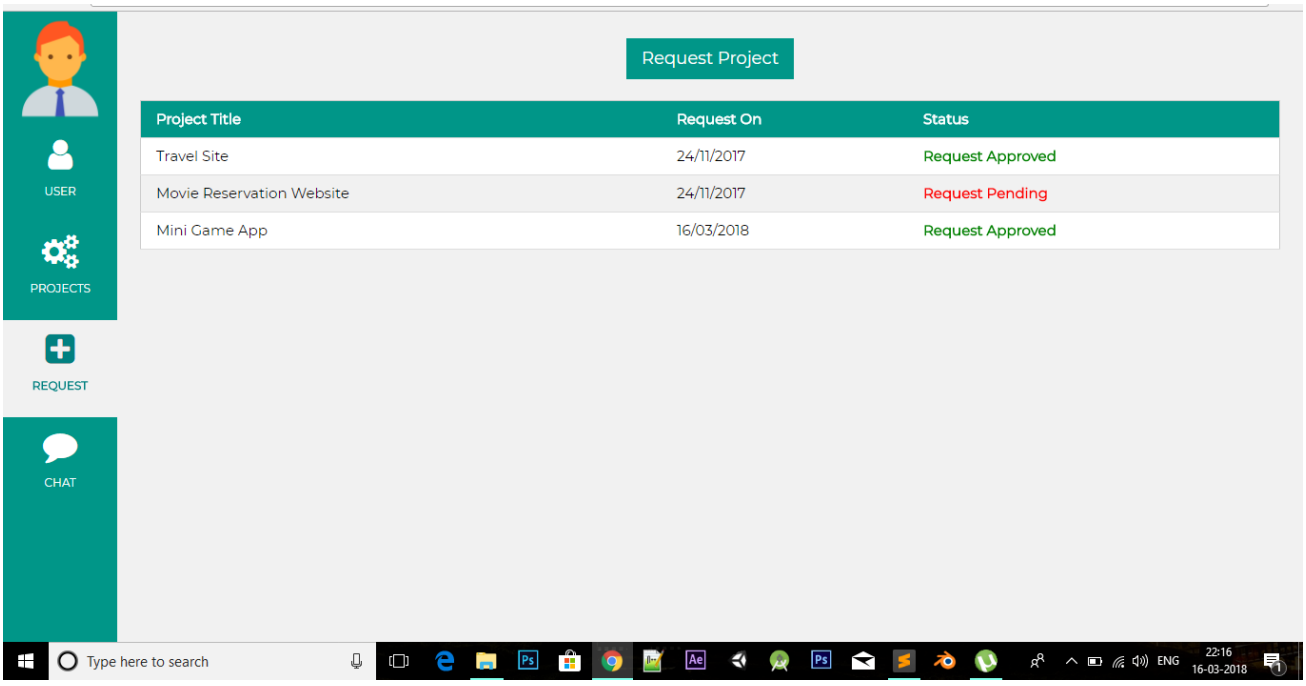
5. Test case 2:

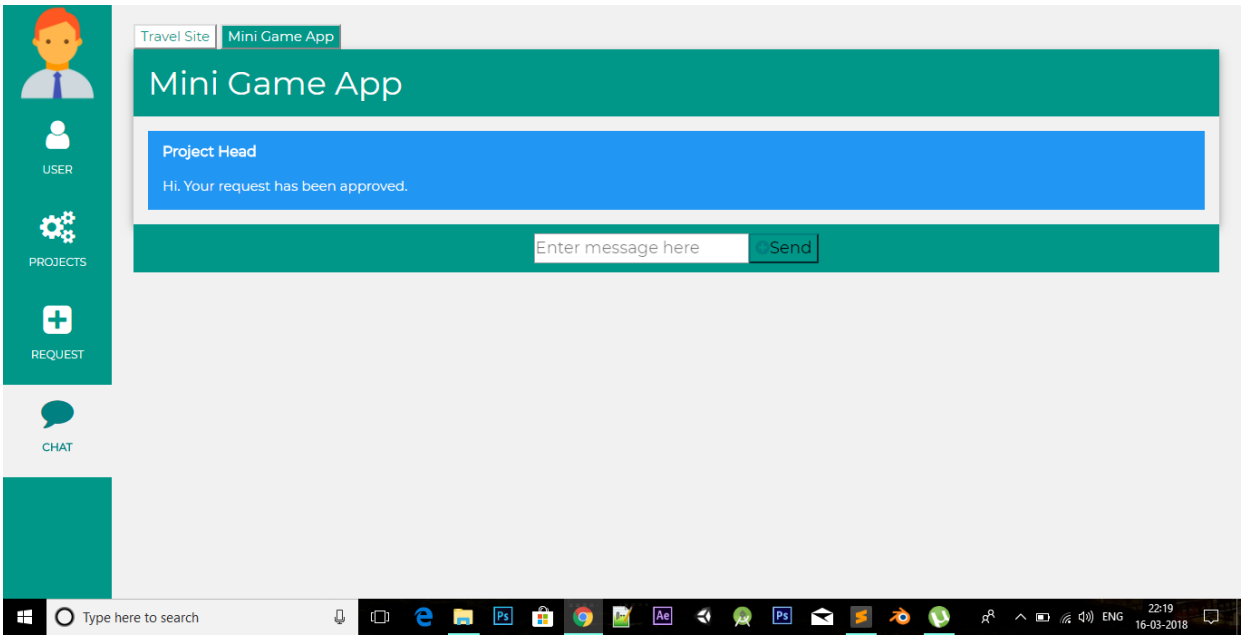
Test Input: Only the approved requests are shown in the chat module.

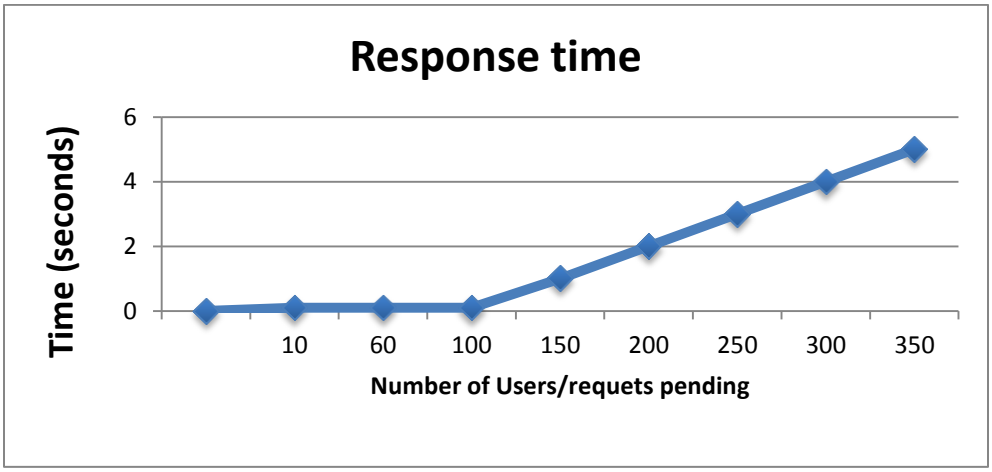
Test Output:




Travel Site
Travel Site
USER
PROJECTS
REQUEST
CHAT
Project Head
Hi. How can you add to the project?
You
I can contribute by content writing and basic frontend development.
Project Head
Awesome. You can contact me on 7530007734.
Enter message here
Send
Type here to search
ENG
22:15
16-03-2018


Request Project
USER
PROJECTS
REQUEST
CHAT
Project Title
Request On
Status
Travel Site
24/11/2017
Request Approved
Movie Reservation Website
24/11/2017
Request Pending
Mini Game App
16/03/2018
Request Approved
Type here to search
ENG
22:16
16-03-2018



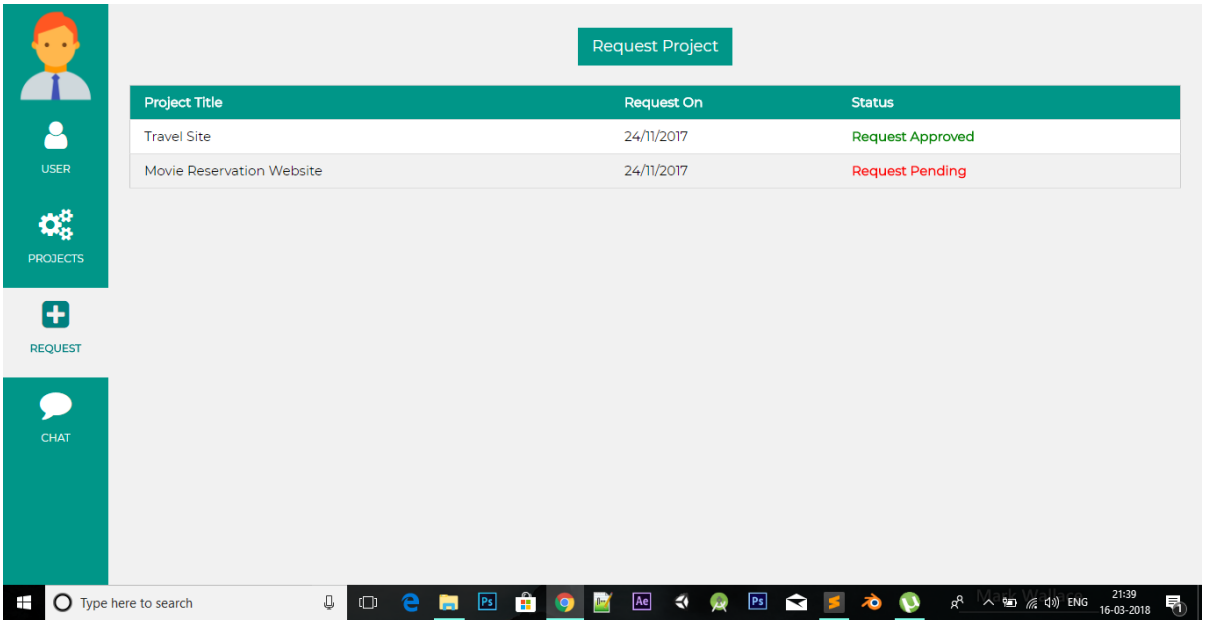


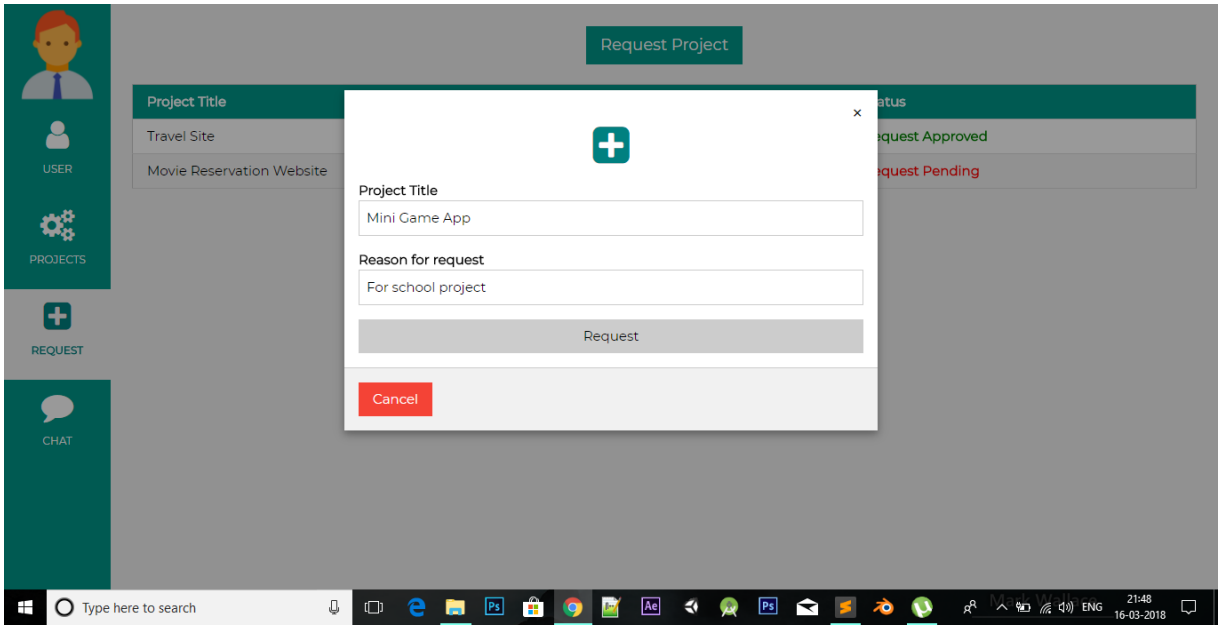
14.4 Test Cases: Module 4

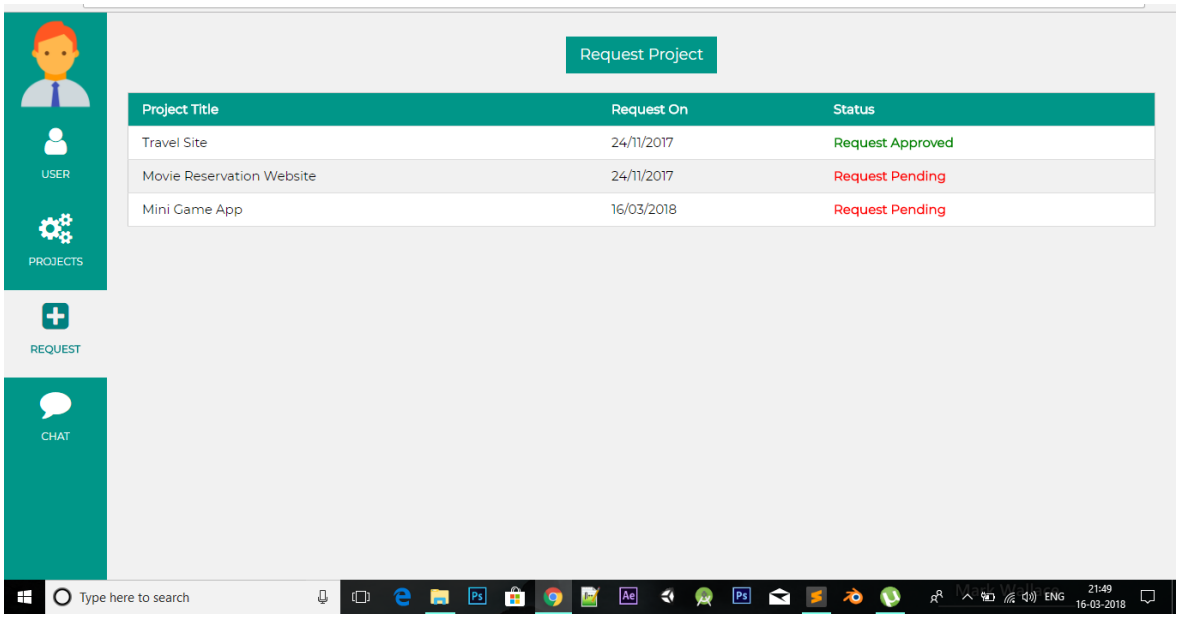
1. Name: Request module

2. Introduction: This module consists of a user's ability to request participation in projects and also see the status of pre-requested projects.

3. Screenshot:


Request Project
Project Title
Request On
Status
Travel Site
24/11/2017
Request Approved
Movie Reservation Website
24/11/2017
Request Pending
USER
PROJECTS
REQUEST
CHAT
Type here to search
ENG
21:39
16-03-2018

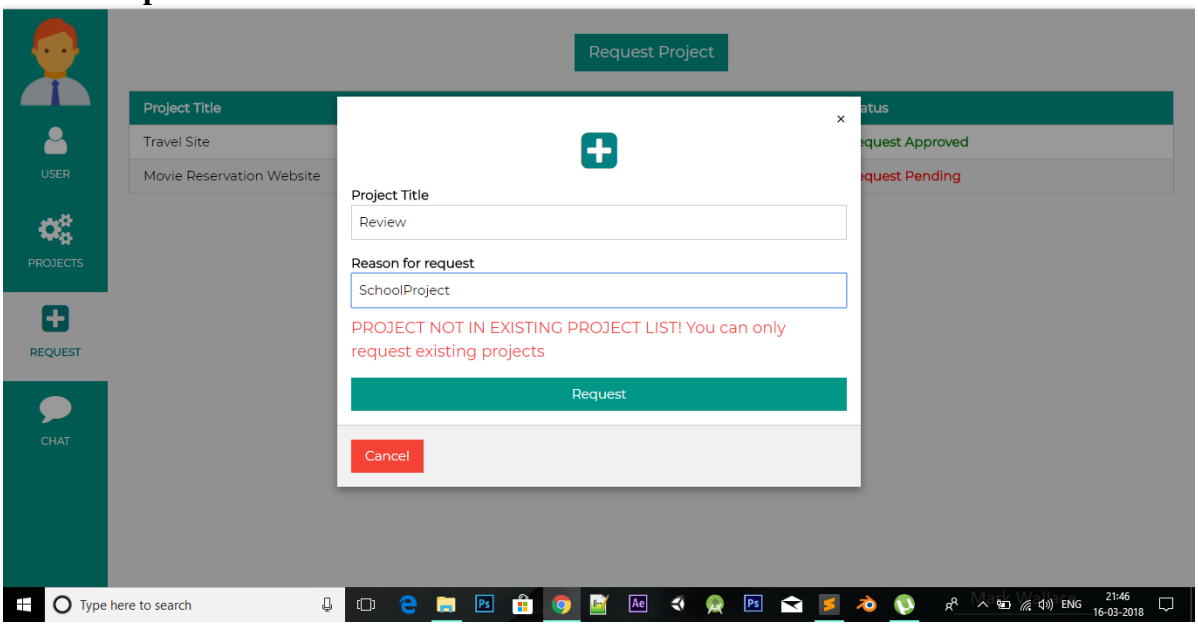

Request Project
Project Title
Travel Site
Movie Reservation Website
Project Title
Mini Game App
Reason for request
For school project
Request
Cancel
USER
PROJECTS
REQUEST
CHAT
Type here to search
ENG
21:48
16-03-2018



4. Test case 1:

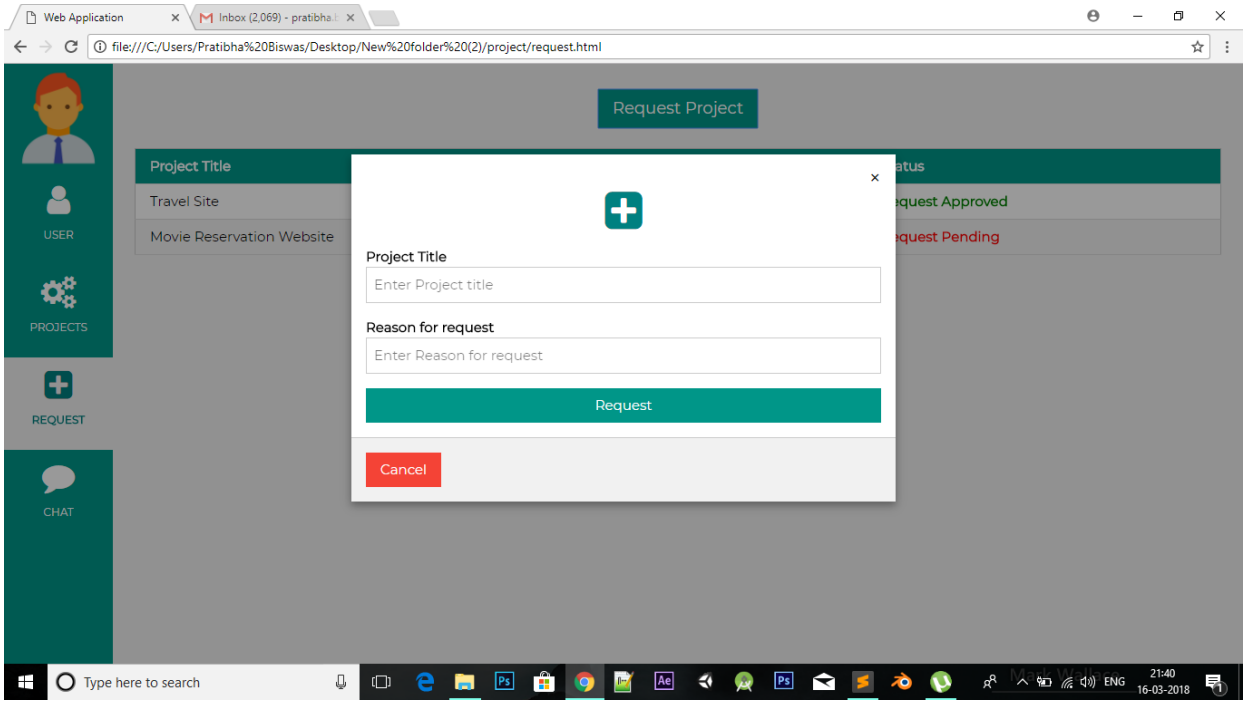
Test Input: Requesting for a project that does not exist will not process the request and display an error message.

Test Output:



5. Test case 2:

Test Input: Testing whether a request that has been processed successfully is added to the project list.



Test Output: