FACHHOCHSCHULE KIEL
HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN | UNIVERSITY OF APPLIED SCIENCES

Knowledge Information System (KIS) Dashboard

Submitted by:

Mallika Mudda (927904)
Sravan Kumar Katta (927937)
Vinitha Thanasekaran (930026)
Amrita Das (930024)
Ojeswini Balasani (928382)

Table of Contents

ABSTRACT

This documentation details the work undertaken as a part of building the Application KIS Dashboard i.e. Kiel International Students Dashboard. The Dashboard aims at providing information to International students in and about to study in Germany. Apart from that, it acts a communication medium for students to find friends, have a discussion in various categories and chat with other registered users. It has a section where students can create an event for setting up a fun skilled training class and earn extra income. Also, included within is a mini-blog kind of atmosphere where participants can post their articles.

The entire project was built in the Meteor environment using JavaScript for the web interaction and HTML5 and Bootstrap to build an easy on the eye User Interface. The KIS dashboard is the result of the project for the developed for the master's course "PM102-Advanced Application Programming" at the Fachhochschule Kiel, Germany which aims in obtaining full-stack developer capabilities.

1. INTRODUCTION

The project KIS-Dashboard was undertaken with the aim of providing a useful, friendly experience for the International Students in Kiel, Germany. Any registered user can gain access to a set of functionalities that give detailed information on any topic or enable him to interact with other students. The Events section is dedicated to setting up an event or participating in one. It is a fun and informative platform for students to advertise their skills, earn some pocket money and interact new people along the way. It also has detailed information for to guide students through each stage of applying for a university in Germany to arriving here and setting up their new home.

Since the team had no idea how to start a full-fledged software project with multiple sources like NodeJS and integrating it with a database MongoDB and then feeding the objects to the server, we decided to use a framework. After a few weeks of exploring, we found a comfortable platform called Meteor, which would get us started right into programming, avoiding most of the setup issues. Meteor had its range of issues which were easily tackled thanks to the extensive tutorials on their website. By the end of 8weeks, we were able to deliver a fully functional application using Meteor. In the future, we will try to make it more interactive by implementing an email mechanism, and automatic alerts on the mobile phone, for instance.

2. OBJECTIVES

Main Objectives of the Project is to develop a full-scale web application with the following:

- JS on the Front End
- JS on the Back End
- GIT Version Control
- Database Access
- Authentication & User-sign Up
- Web Socket Implementation

3. METHODS

3.1 Project Management

The team consists of five members. The distribution of the tasks is depicted in the Task Matrix below. To involve all the team in every aspect of the development that tasks are divided as functionalities and the team member has to develop the functionality assigned to them and that includes UI design, database design, testing, routing, etc.,

Team meetings were regularly conducted to track the progress of the project and to discuss the future objectives that need to be achieved.

Table 1 - Task Matrix

Task	Amritha Das	Mallika Mudda	Ojeswini Balasani	Sravan Kumar Katta	Vinitha Thanasekaran
Brain Storming & Research	✓	✓	✓	✓	✓
Project Management		✓		✓	
Documentation & Poster	✓	✓		✓	✓
Initial Iron-Router Setup					✓
Initial Database Setup				✓	
Authentication				✓	
Authorization	✓	✓		✓	✓
Articles Functionality	✓				
Events Functionality				✓	
User Profile				✓	
Search/Find Friends					✓
Chat Application			✓		
Contact Functionality		✓			
Discussion Forum Functionality		✓			
Static UI & Home UI Design		✓			

3.2 GITHUB

The project repository was hosted on GITHUB and made available to all the team members. GITHUB allowed us work in collaboration and maintain the versions of the system. Members maintained separate branches while working on each functionality and at the end, everything is merged to the master branch. The master branch served as the main release branch, which was regularly updated from the branches and tested. Below is the link to the GITHUB repository of the project:
https://github.com/muddamallika/PM102-JS-Project

3.3 Framework

Meteor Framework is used in developing the project. The framework helps to develop a web application in less time compared to other frameworks. Meteor provides many inbuilt functionalities that help in developing the project. Below diagram depicts the framework implementation in the present project at a high level.

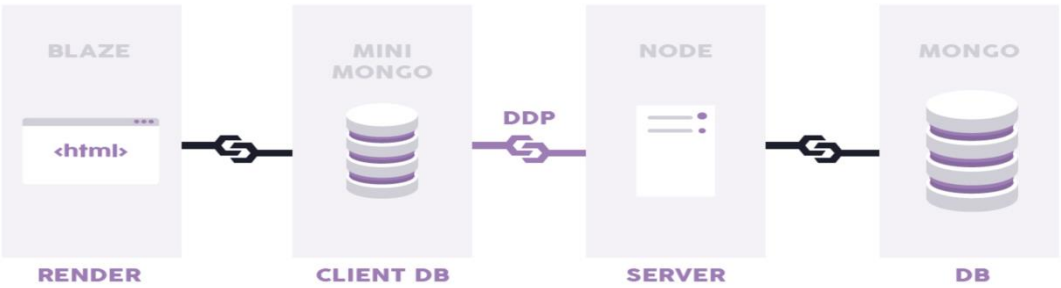


Figure 1: Meteor Framework Representation

3.4 Technologies

3.4.1 Blaze JS - Blaze JS is used on the client side to render the pages. It is a powerful library for creating user interfaces using reactive HTML templates. It eliminates the need for updating logic in the web application because it listens for the data changes and manipulates the DOM.

3.4.2 Mini Mongo - Mini Mongo which also resides on the client side pretends to be a database and always keeps in sync with the Mongo. Mini Mongo achieves this by communicating data changes using DDP. Below figure depicts how Mini Mongo works

3.4.3 Node JS - Node JS is an open source and cross-platform JavaScript runtime environment for developing different applications. As it uses a non-blocking, I/O mode thus makes it lightweight and efficient.

3.4.4 Mongo - Every meteor project comes with an automatically created database called “Mongo” without any setup and configuration required.

3.5 Architecture - System Landscape

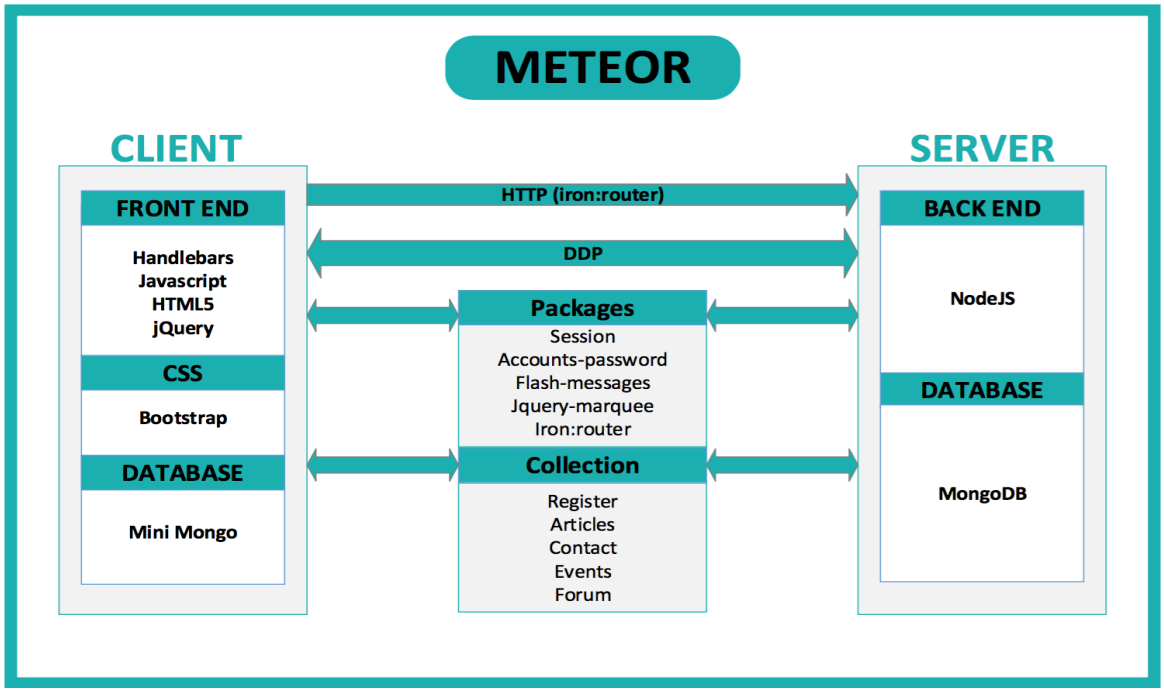


Figure 2: System Architecture

3.6 Web sockets

In Meteor data synchronization between client and server are done well using the DDP(Distributed Data Protocol) instead of HTTP Protocol. Meteor uses web sockets with a protocol called Distributed Data Protocol, and this helps the client and server to share everything. Below diagrams depict how the communication between client and server using AJAX and Meteor differ.

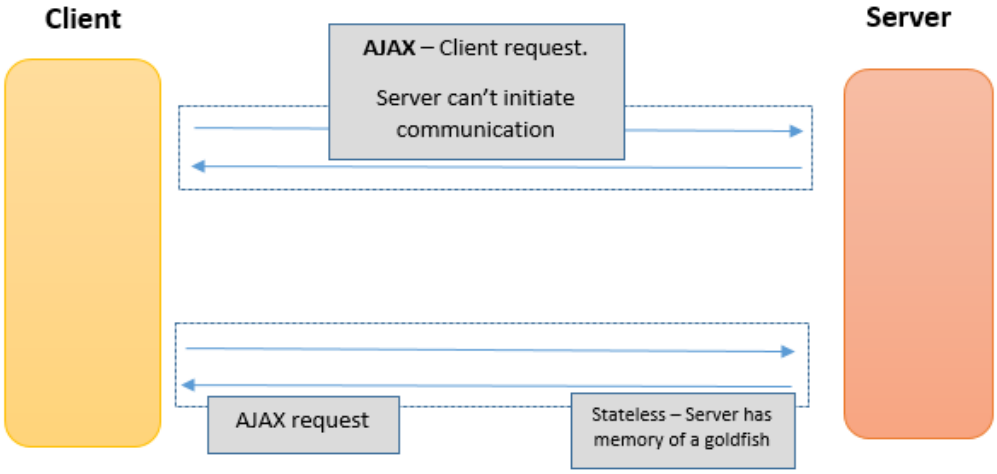


Figure 3: Client and Server Communication by AJAX

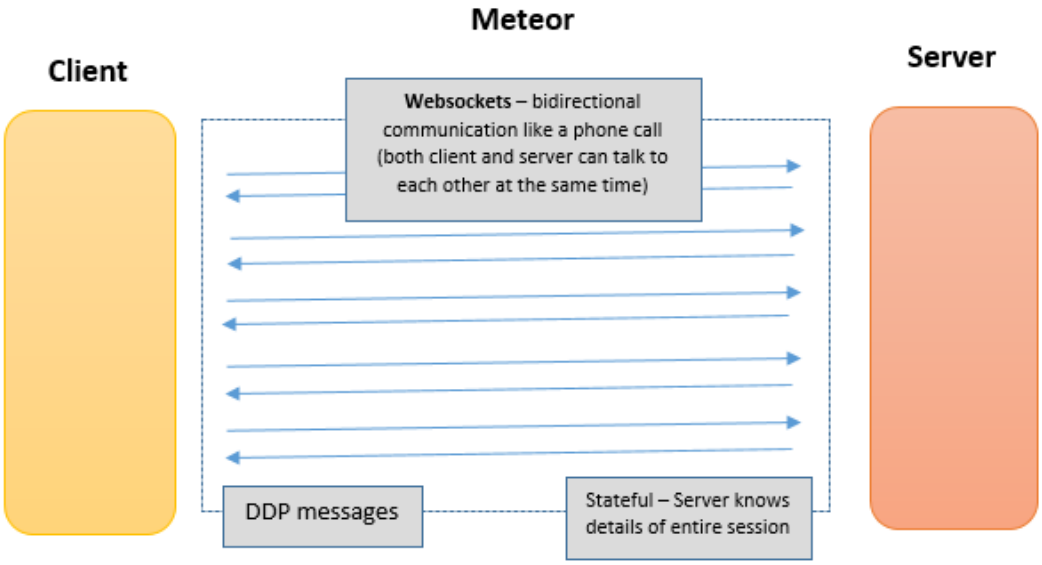


Figure 4: Client and Server Communication by Meteor

3.7 Development Methodology

To achieve better results, the team followed Agile development methodology and worked in sprints to achieve the desired objectives.

3.8 Functionalities

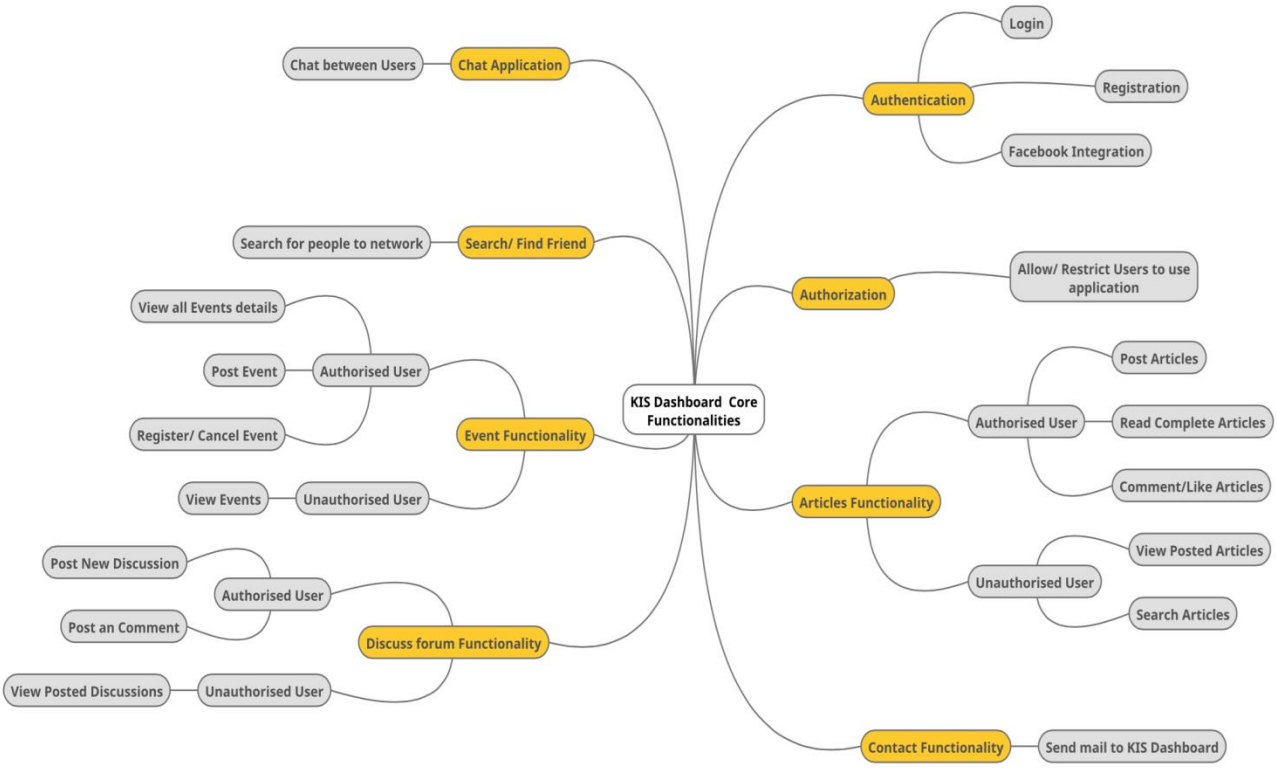


Figure 5: Mind map diagram of Knowledge Information System (KIS) Dashboard

3.9 Test Methodology

The manual Testing methodology has been used to test the KIS-Dashboard application. The test cases include all the functionalities considering different input scenarios and type of user. The expected and actual results are compared with different input parameters. The bugs resulted from failed test cases are fixed. The application was thoroughly retested until all the issues are resolved. KIS-Dashboard is now a stable product with no defects.

4. RESULTS

The KIS-Dashboard can now serve as the starting point for the International students who wish to pursue their education in Germany. As planned the project was able to achieve the core functionalities within the stipulated period. The tests performed on the final results achieved desired results.

5. DISCUSSIONS & LIMITATIONS

The outcome of the project is a full-scale web application with authorization and authentication capabilities, along with the main features designed for the project. The present website allows students to create events and participates in the events created by others. It also enables them to write and share articles which span in different areas, participate in the discussion forums and get valuable answers. The students can search for their acquaintances and increase the chances of communication and also provides chat application that enables the communication among

the registered members. The project strived to achieve most of the features. But features like Forgot Password, Email verification couldn't be achieved due to the limited knowledge of the team in programming using Meteor.

6. CONCLUSIONS & OUTLOOK

The KIS-Dashboard has been developed on Meteor Framework with blaze.JS on the front end and MongoDB as backend. It will be a knowledge hub for all the international student who aspires to study in Kiel, Germany. It will facilitate users to initiate a discussion forum, post their articles and create an event. A creative podium which will not only help the students to earn while learning but also will be a social platform to interact with other users. The project being developed on Meteor framework provides user-friendly applications, excellent UI with a quick response time.

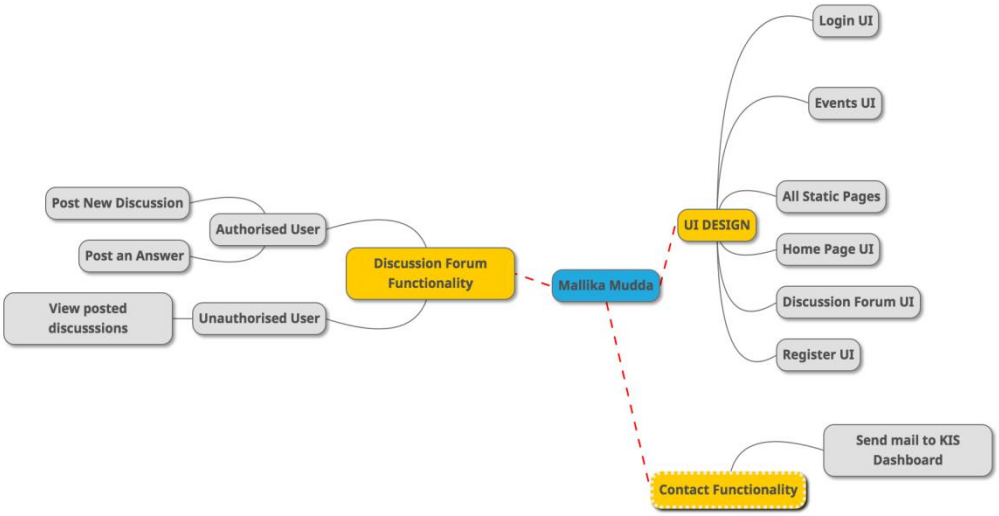
Though most of the key functionalities have been developed keeping in mind the need of international students, new features could be added in future to make the Dashboard much more effective. Few functionalities such as File/Photo Upload option, Pagination of the Articles, Email Verification could have been implemented if some more time was available.

REFERENCES

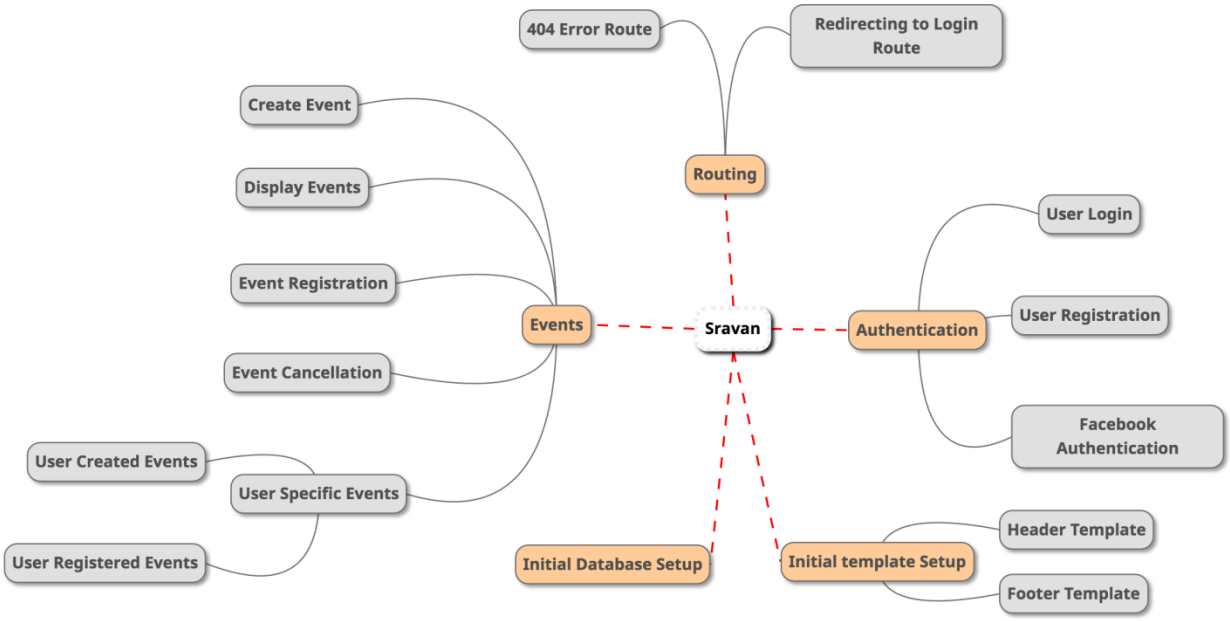
[1] Strack, *Getting Started with Meteor JavaScript Framework*. Packt Publishing Ltd, 2012.
[2] K. Chodorow and M. Dirolf, *MongoDB: The Definitive Guide*. O'Reilly Media, Inc., 2010.
[3] A. Weil, *Learn Meteor - Node.js and MongoDB JavaScript platform*. Lulu.com, 2016.
[4] C. Niska, *Extending Bootstrap*. Packt Publishing Ltd, 2014.
[5] M. D. Group, "Build Apps with JavaScript | Meteor." [Online]. Available: https://www.meteor.com/. [Accessed: 16-Dec-2016].
[6] "Bootstrap · The world's most popular mobile-first and responsive front-end framework." [Online]. Available: http://getbootstrap.com/. [Accessed: 16-Dec-2016].
[7] B. Syed, *Beginning Node.js*. Apress, 2014.

APPENDIX

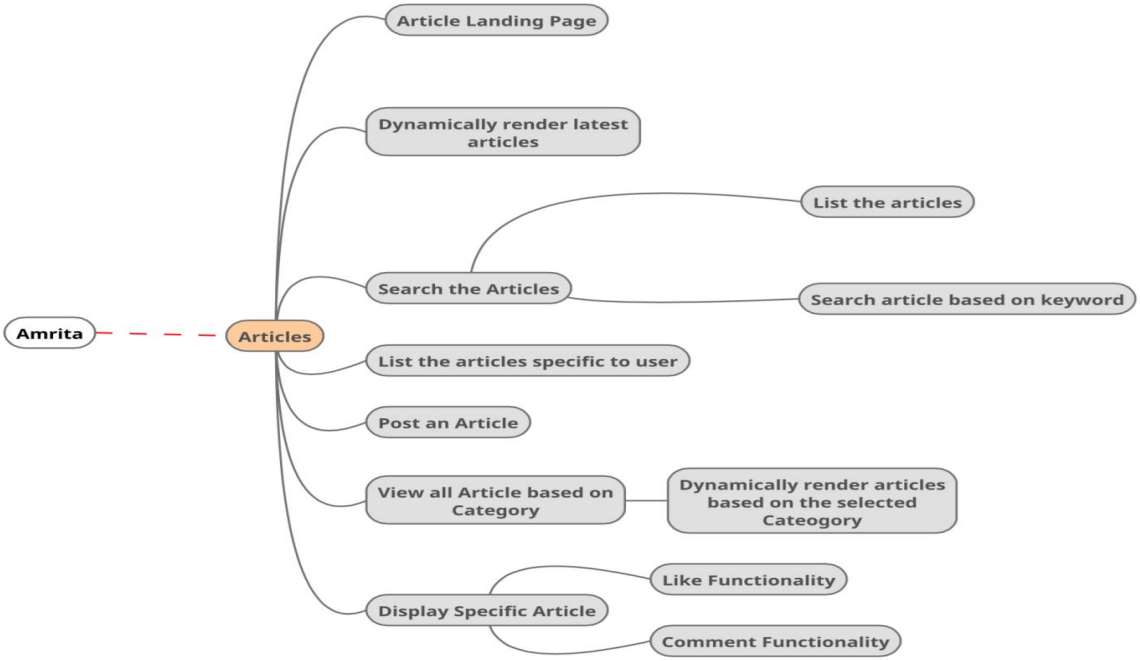
Mallika Mudda



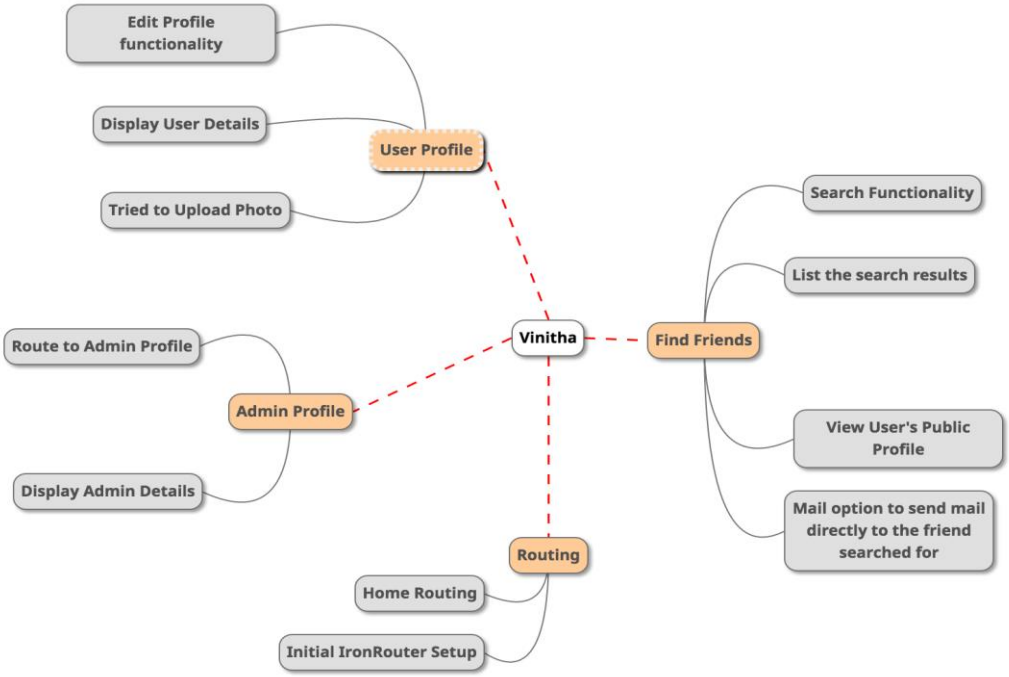
Sravan Kumar Katta



Amrita Das



Vinitha Thanasekaran



Ojeswini Balasani