Chapter 2

2. Literature Survey

For creating the JU Query Web application, a diligent and dedicated analysis of Requirements have been performed along with elaborate research for choosing technology stack.
This exploration has been backed up by 25 years of development and upgrades that has been observed in the field of Web technologies and JavaScript Ecosystem which includes React JS front-end library, ReduxJS library and Firebase cloud domain for APIs, Databases and Authentication. Right from its creation, to brief about each year's development the topics are covered with its respective years over here:

Year	Web technologies which were discovered and published in the respective year.
1990–1995	● HTML, CSS and JavaScript are invented
1996–1999	● Standardization efforts begin ● Browser compliance is terrible. ● Browser wars ignite.
2000–2004	● CSS frameworks begin to emerge. ● jQuery is born. ● Front-end package management.
2005–2009	● W3C specification compliance is met. ● Chrome browser takes the lead. ● Responsive designs and frameworks are introduced.
2010–2015	● JavaScript Frameworks are born: Backbone, Ember, AngularJS, React, Angular, Vue. ● HTML5 is announced. ● Awareness about firebase and Software as a Service came into the picture. ● The reason for ReactJS 'highly efficient performance is that the framework is essentially the virtual DOM feature of the framework. ReactJS is deployed to develop reusable user interface (UI) components. ● Redux (javascript library) came into development ● React Redux is the official React binding for Redux. ● It lets your React components read data from a Redux store, and dispatch actions to the central data store to update data.

2016 - 2021	● How Agile based methodology can solve the Problem is the clients should also make an effort to understand a company's work processes thoroughly. ● Individuals at the top level of management can be hesitant to embrace Agile models even in the fire base's backend-as-a-service feature. ● Community-based Question and Answering (CQA) administrations have conveyed users to another period of information dispersal by enabling users to make inquiries and to answer other users' questions. ● And we analysed the need of question answering site in our university

2.1 Related Work

A revised 2017 Research highlights the different problems defined to implement React in market growth and popularity. These problems are mainly about ("big Development Community"), ("React core Architecture"), ("React is fast and Agile"). They highlighted that using React makes it easier to create interactive user interfaces. It efficiently updates through rendering the exact components to the view of each state and makes the data changes in the application.

The next research on Firebase, It is a backend platform for building Web applications. It offers a real time database, different APIs, multiple authentication types, hosting platform and much more. Firebase frees developers to focus on crafting fantastic user experiences. They don't need to manage servers. You don't need to write APIs. Firebase is your server, your API and your data store, all written so generically that you can modify it to suit most needs. Overall, it has a great deal of information providing surface knowledge on business benefits of SaaS. It has a ton of useful features that can all be tuned and tweaked to power the back-end of a **SaaS** application.

2.2 Existing Work

Some existing work on one of the features which the model directly or indirectly provides to or rather highlights it to the viewers. Redux proprietary research topic in 2016 by Alex , a front-end web developer emphasized its concentration on Redux; it is one of the hottest libraries in front-end development these days. However, many people are confused about what it is and what its benefits are.

Majority of our works and discussions are raised on 2021.Community-based Question Answering destinations enable users to find solutions to complex, point by point and individual questions from different users. Be that as it may, since answering a question relies upon the capacity and readiness of users to address the asker's needs, a huge portion of the questions stay unanswered. A fresher just took admission in Jain University, does not know anything about college and does not have any connection to others. He/ She can directly take our application and can find out his/ her queries. If his / her query is not present then can post his/ her query here.

System will ensure the students do not provide any fake answers or events through our site. Students will be able to see the live events happening on campus.

This research is also provided with Comprehensive details about SDLC. The Software Development Life Cycle (SDLC) refers to a methodology with clearly defined processes for creating high-quality software. In detail, the SDLC methodology focuses on the following phases of software development.In this we can assign every team member to contribute their work separately through respective phases.

2.3 Problems in Existing system

Here are the list of problems, some of them are already explained in related and existing work

- Risk of mismanagement and of data when the project is under development.
- It will be more complicated when new features associates with environment settings and will become cost-bulky
- ReactJS standard framework reusing methods and services saves the programming effort and improves reliability of the application

- Communication between customer and owner is not direct.
- The complete hierarchy is not implemented manually.
- A system can be developed to be precise and that will be used if install must still be a good speculation for the organization.
- Still Will there be no hostility from the user that will undermine the promising application benefits

2.4 Proposed System

This addresses the demand of our project's idea and those services help the participating students or teachers to find better answers from so many queries and also learn about our university in non-invasive ways.
"JUQuery" is a platform for the purpose of knowing about the academic and nonacademic environment, campus life of a student, faculty and student relationship of Jain University in virtual way.This application provides us with complete information about campus life in student point of view, academic information in college point of view.
As stated, it provides guidance in following aspects,

- It gives an entrancing view of the whole process and services of JUQuery web application.
- The structure of the measurement system is on the basis of the findings of an earlier stage of the research that had resulted in the development of a number of process-based KPIs and the key issues they encompass.
- It follows Agile Methodology so get the benefits like Regular adaptation to changing circumstances , Even late changes in requirements are welcomed,
- And also testing practices have worked wonders for numerous organizations with positive aspects.
- On initial stages fire base provides an extensive help of A wide range of services and features and it is well known for backend-as-a-service. It has a ton of **useful** features that can all be tuned and tweaked to power the back-end of a **SaaS** application.

Chapter 3

3. System Design

3.1 Architecture

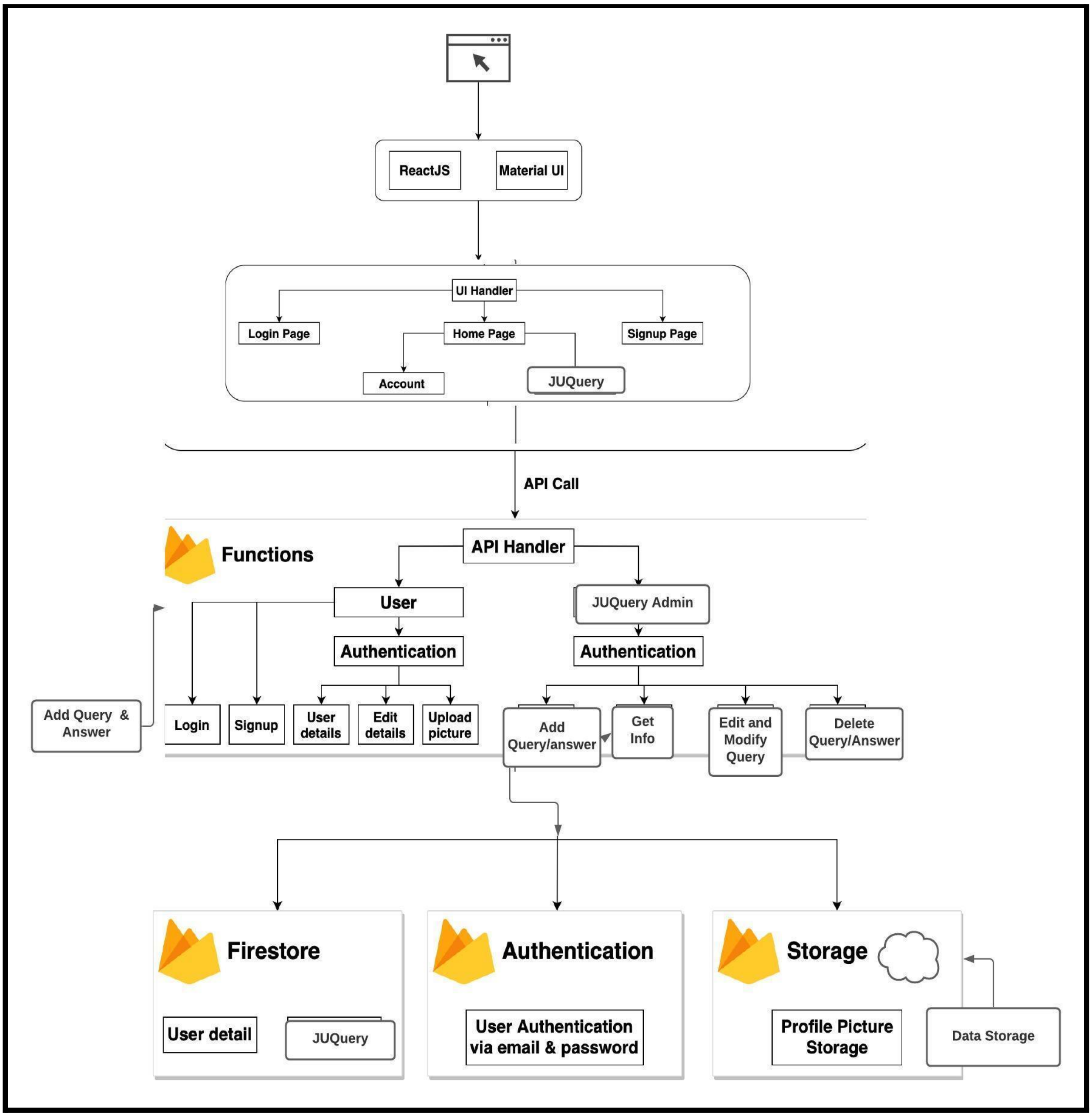


Figure 3.1.1: Architecture and Flow Proposed By Guide

Our guide recommends the following actions to be performed which leads to a hassle free and easy environment for **“JUQuery”** web application development.

1. It all starts with the decision on “what type of application needs to be developed for our university and also based on the current business needs”. This is essential and acts as a base for deciding what do agile users expect as an outcome and what kind of technology needs to be included to achieve it. It also helps in further evaluation of cloud services that can be used for achieving business logic.

2. Requirements for software are laid, which encompasses all types of requirements such as mission, goal, scalability, audience, feature set, etc. This helps in gathering information about users and market and how they react with each other creating demand for the application. This is a part of MVP development scheme that is used in combination with agile in this guide.

3. Then a thorough analysis is done on what kind of cloud provider will be best suited and the decision for a provider is mainly based on cost, legal, and compliance.

4. Once the provider is decided, its platform is evaluated, and its services can help us to achieve the solution. The main agenda is to look at what kind of web application is being developed and which kind of target audience would be its market. This can bring clarity to the services which needs to be put into action to achieve the development.

5. Agile development methodology is much more beneficial as compared to other traditional software development methodologies

6. As a part of Agile development, the team gets to choose the ReactJs and ReduxJS libraries. The guide doesn’t specify the user to pick a specific one as depending on the type of application and adaptation of the developers, any of the two can be picked and followed to manage modules.

7. MVP or Minimum Viable Product is the correct balance of Minimum and Viability factor which ensures proper interests of end users and also takes in the minimum amount of effort that needs to be put in the development. Efforts here can be a replacement of cost, scalability, mobility and sometimes even features. A product of MVP development method is a subset of Minimum and Viable.

8. The mixture of Agile values and MVP development gives a much better, clear and clean solution as it
    - Ensures delivery of working software in weeks rather than months.
    - Instead of customer satisfaction, customer interest is prioritized.
    - Sustainable development with a constantly evolving pace depending on evaluating customer wants and delivering a MVP in process.
    - Simplicity in workflow ensures maximizing the amount of work undone.
    - Welcoming changing requirements
    - Regularly the team reflects how to cash in customers’ interest and become more effective.

9. In the implementation phase, implementation is laid where business logic, backend, front end is implemented using suitable technologies.

10. Every version produced through the development proceedings is tested both locally and on the cloud in a pseudo production environment keeping all the edge cases into consideration.

11. In the final stage, it deployed in the fire base since it is a cloud-ready architecture there are no major lifts and shifts during deployment which leads to the final stage of maintenance.

12. In the final stage of maintenance, continuous improvement takes place based on the changing in customer or user needs. This guide truly promotes team growth over process and tools and always is adaptable to change, the model can endure change and keep up with it.

Firebase is a Backend-as-a-Service (Baas). It provides developers with a variety of tools and services to help them develop quality apps.

- The Firebase API is secured via tokens—in order to generate such a token, we need to call Firebase's Client SDK and log in with a valid user/password credential.
- When successful, Firebase will send a token back in the response which we can then add to the header of any following request we want to perform.
- Firebase gives us complete control over authentication by allowing you to authenticate users or devices using secure JSON Web Tokens (JWTs).
- we generate these tokens on your server, pass them back to a client device, and then use them to authenticate via the signInWithCustomToken() method To reduce problems promoting code changes from development to staging to production, instead of including API keys in the code itself, either set them as environment variables or include them in a configuration file.
- Firebase data is retrieved by either a one-time call to GetValueAsync() or attaching to an event on a Firebase Database reference.

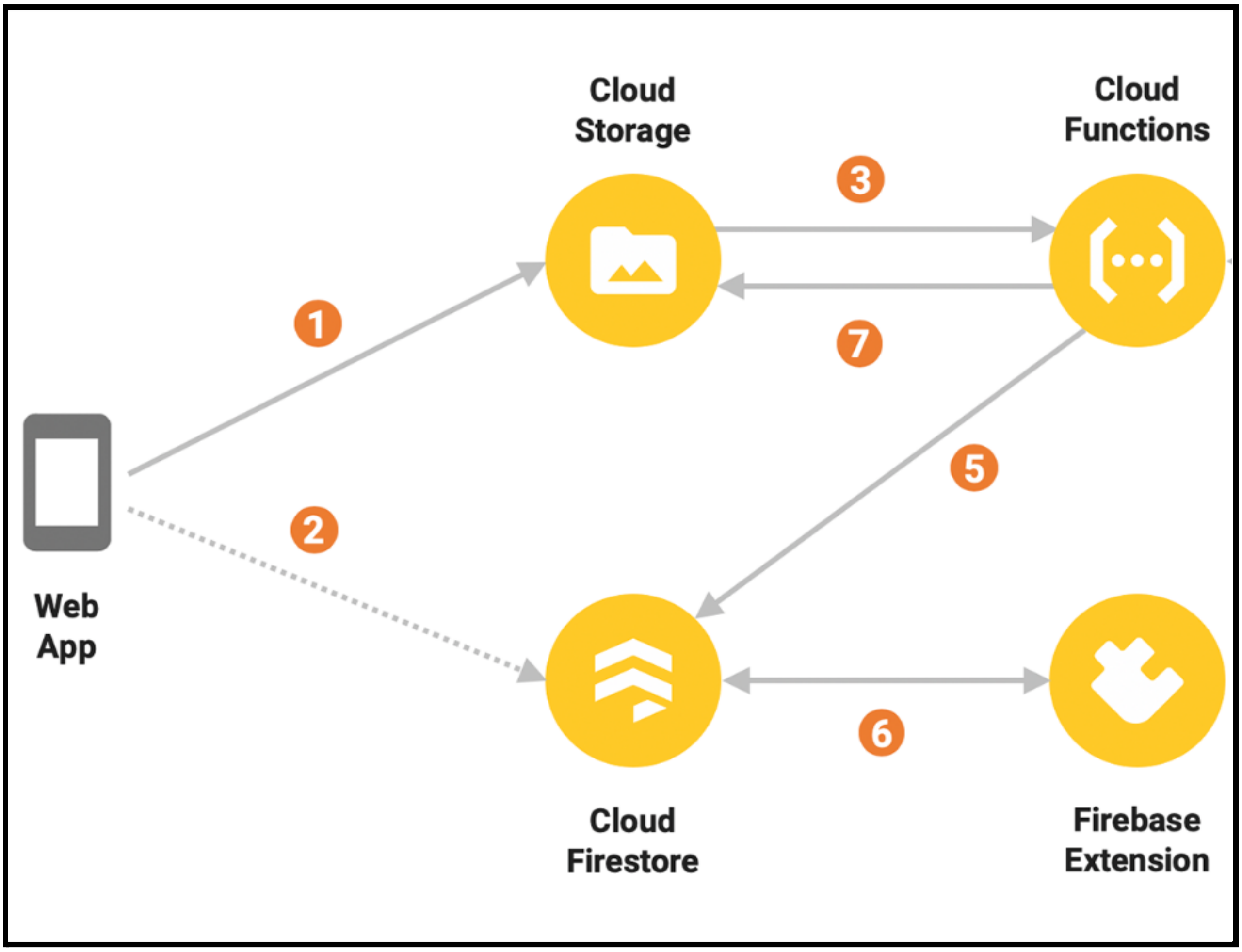


Figure 3.1.2: Application’s connection with Cloud Firestore

3.2 Use case Diagram

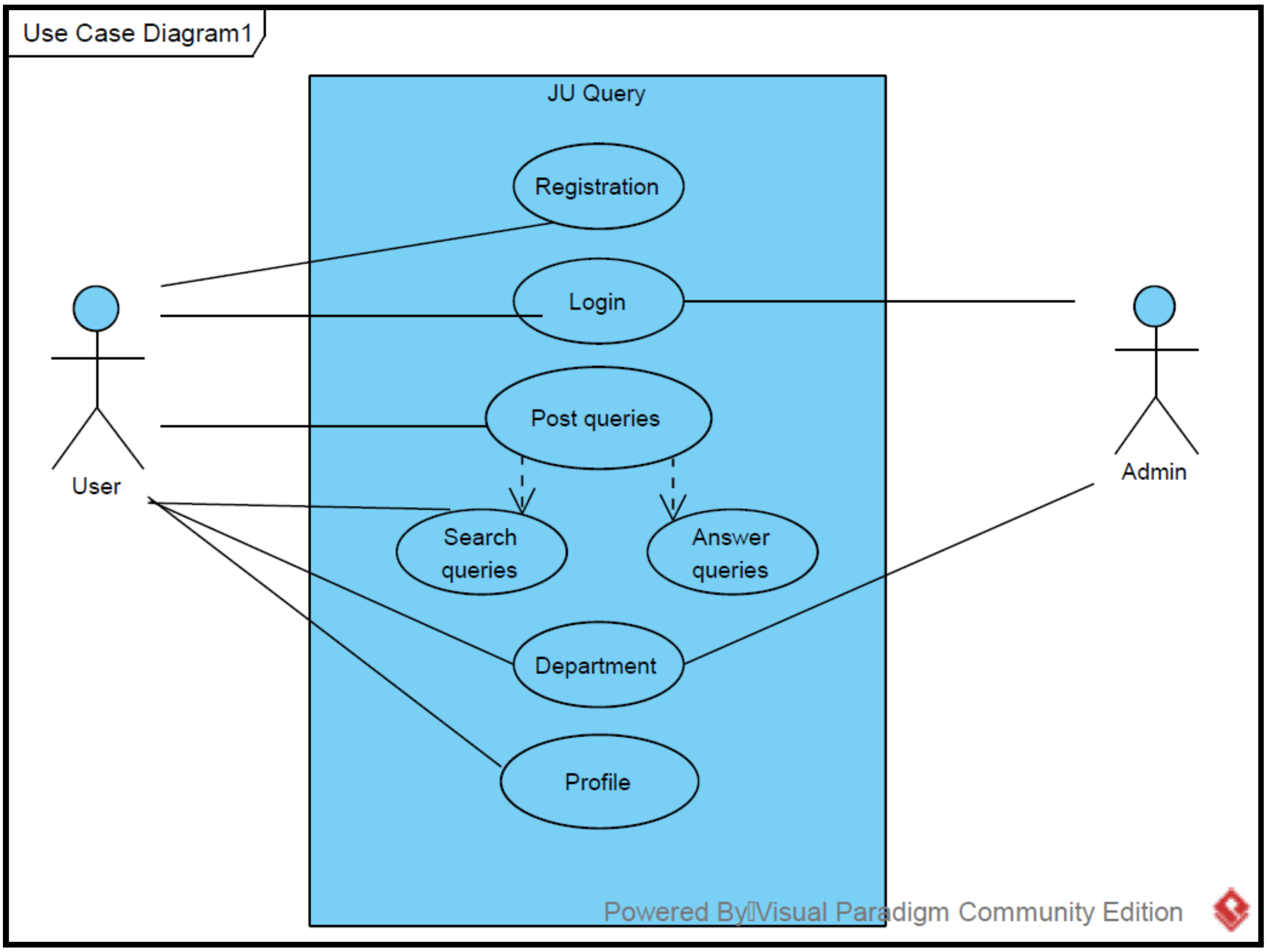


Figure 3.2: A general use case used in the architecture

3.3 Activity Diagram

3.3.1 Activity Diagram for User Side

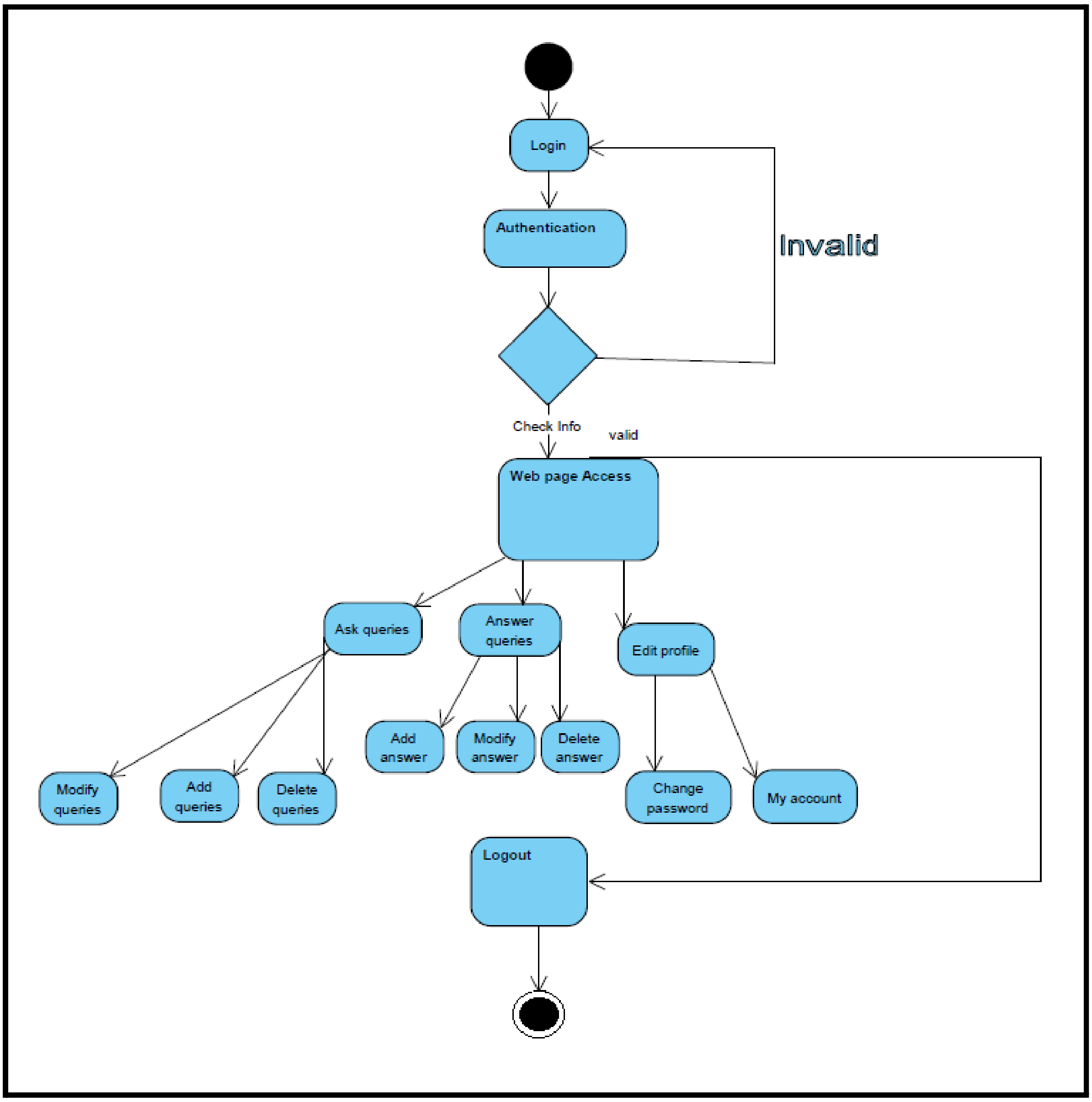


Figure 3.3.1: An activity diagram depicting the user side

3.3.2 Activity Diagram for Admin Side

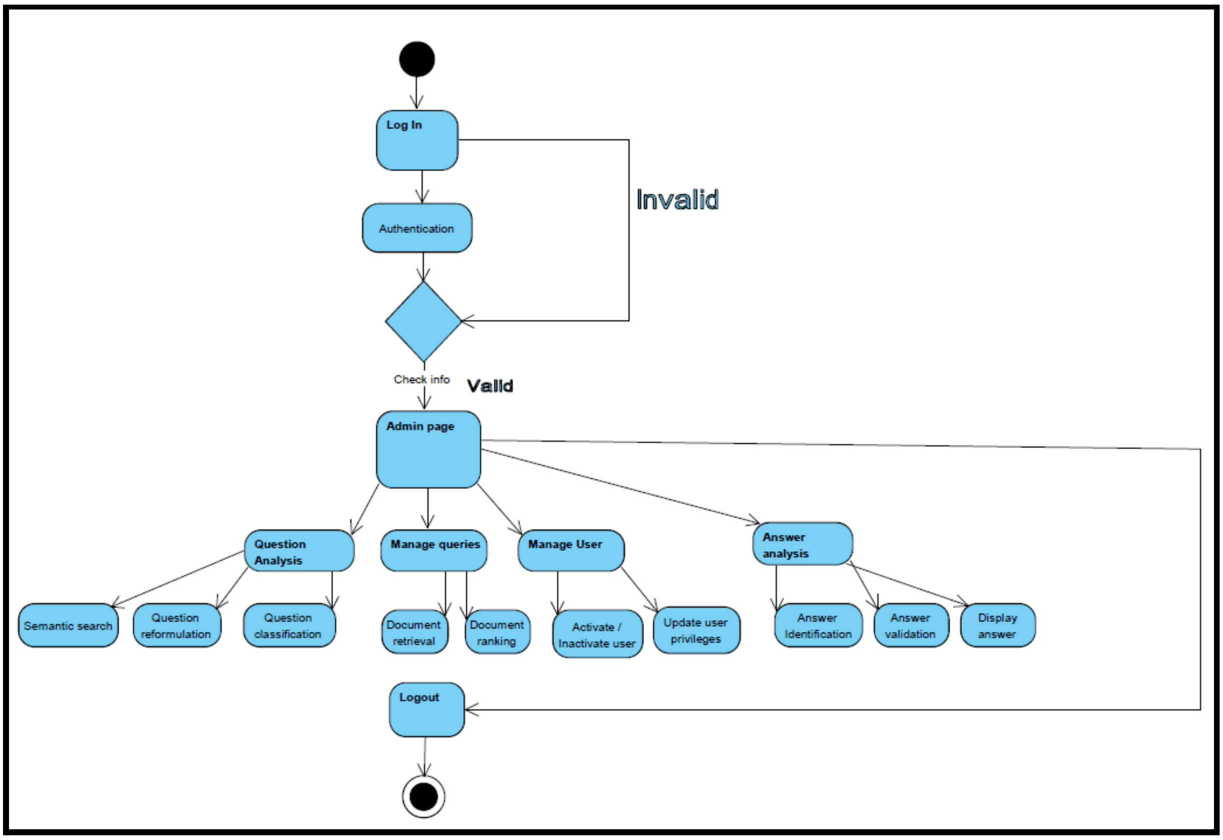


Figure 3.3.2: An activity diagram depicting the admin side

3.4 Class Diagram

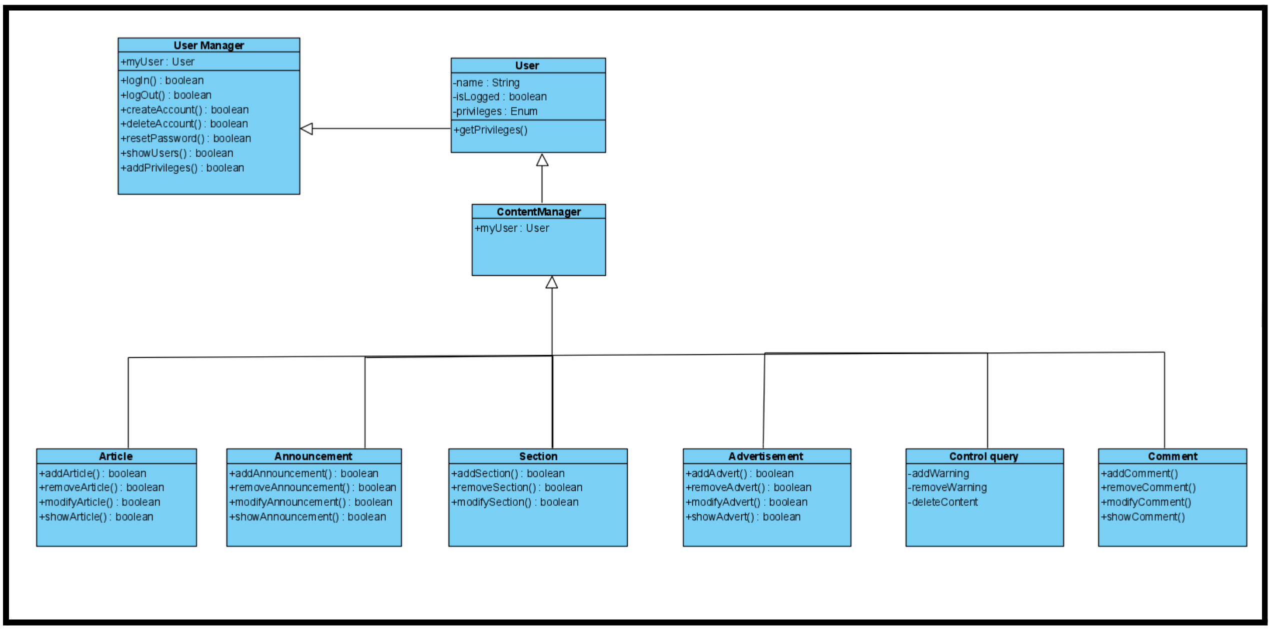


Figure 3.4: A class diagram depicting the coding architecture

3.5 Sequence Diagram

3.5.1 User side Sequence diagram:

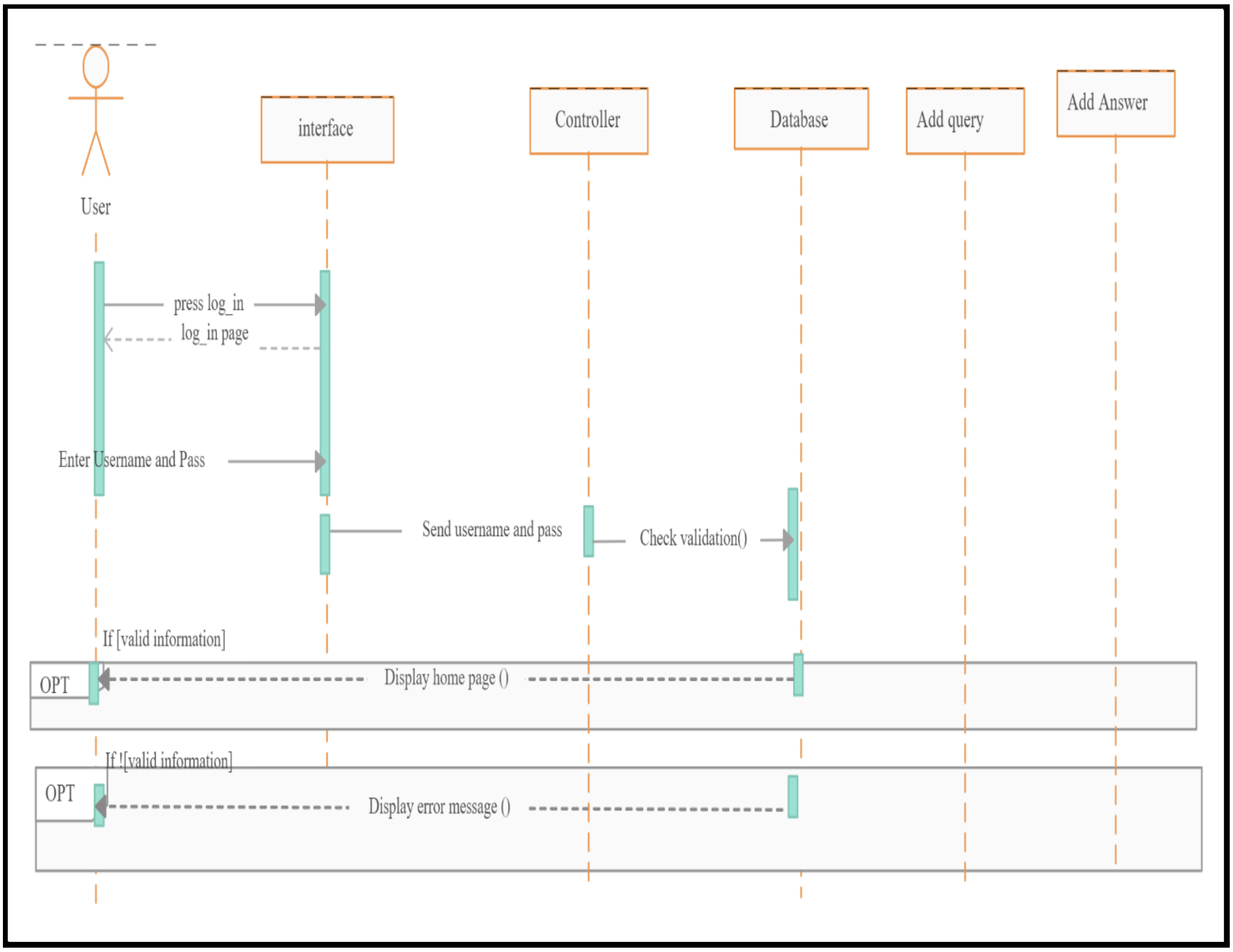


Figure 3.5.1: A general sequence flow used in the architecture

3.5.2 Admin side Sequence diagram:

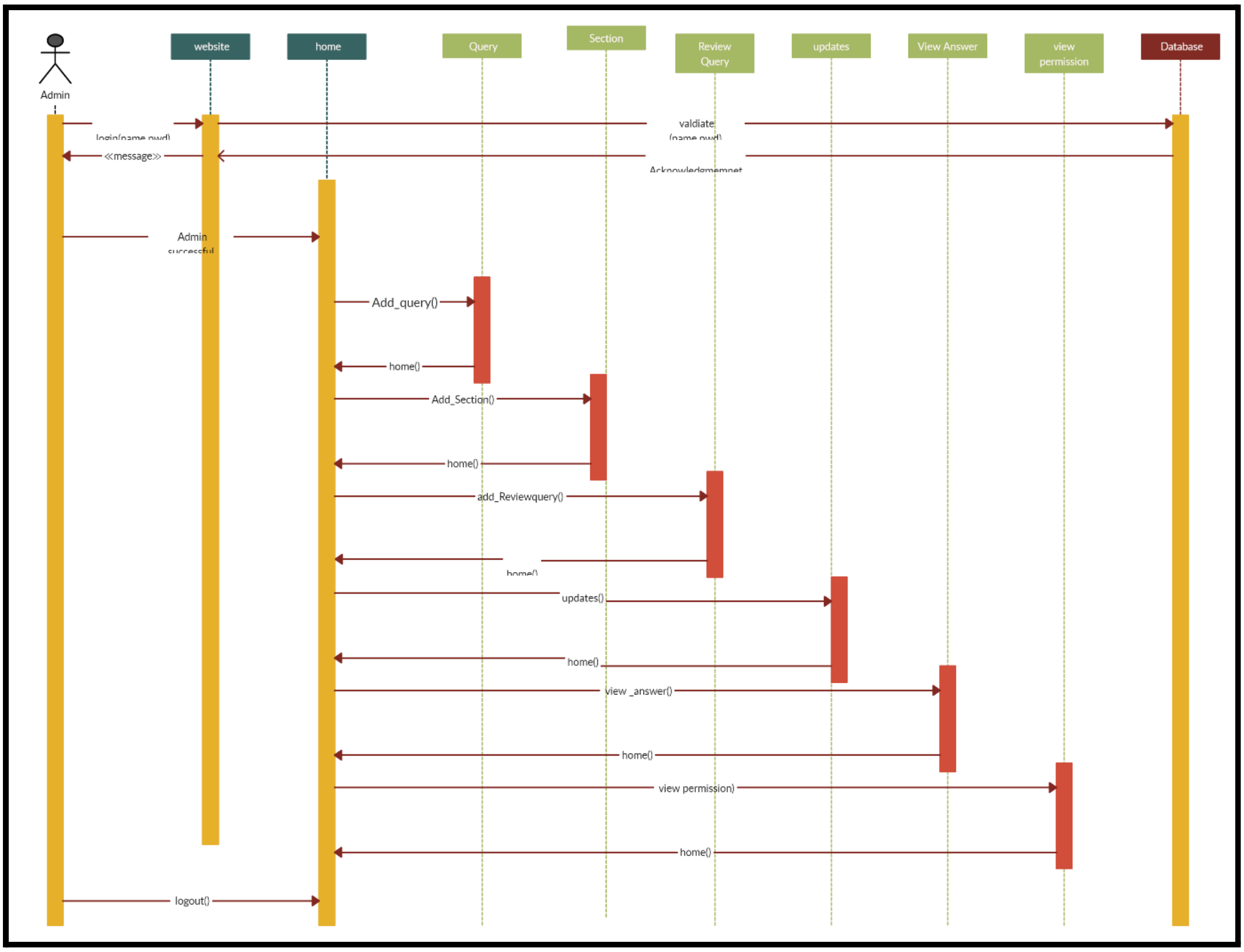


Figure 3.5.2: A general sequence flow used in the architecture

3.6 ER Diagram

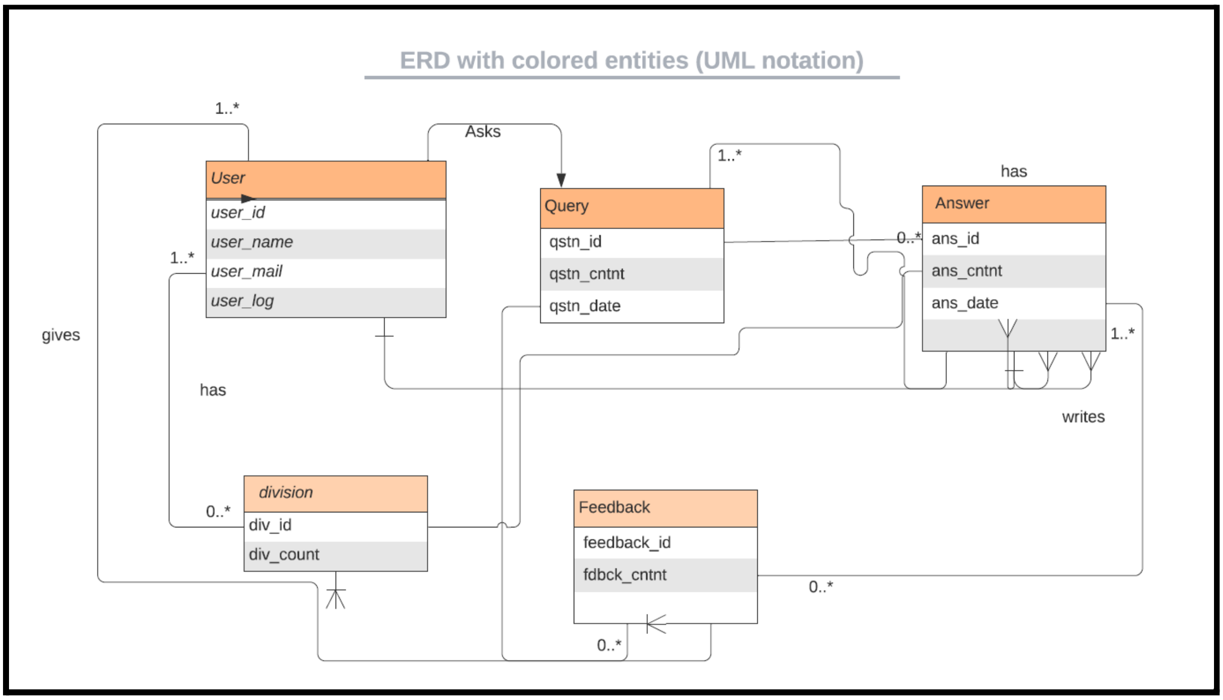


Figure 3.6: An activity diagram depicting the user side

Chapter 4

4. Tool Description

4.1 Hardware Requirements: Description

The hardware requirements for Developers, Users and Admins at JU Query web application are as follows:

- An Intel Pentium 4 processor or later that's SSE3 capable
- 1 GB of RAM
- 1.6 GHz or faster processor
- Hard disk space: 10 GB or above
- Keyboard and mouse or other pointing device
- Network adapters for connection to the Internet
- Graphics display resolution

4.2 Software Requirements: Description

The Software requirements for Developers, Users and Admins at JU Query web application are divided into subsections as follows:

4.2.1 Operating System Requirements

- OS X El Capitán (10.11+)
- Windows 7 (with .NET Framework 4.5.2), 8.0, 8.1 and 10 (32-bit and 64-bit)
- Linux (Debian): Ubuntu Desktop 16.04, Debian 9
- Linux (Red Hat): Red Hat Enterprise Linux 7, CentOS 8, Fedora 24

4.2.2 Client-side Software Requirements

- Google Chrome
- Visual Studio Code
- Git Bash
- Command Prompt
- Windows Powershell

4.2.3 Developer libraries

- React.js
- React-modal
- Redux.js
- Material-ui
- Firebase
- React-testing-library

These requirements satisfy the Developers, Users and Admins to access the JU Query Web Application for seamless user experience, easy access to the database of Queries and answers through API-requests and secure authentication using Firebase Auth functions.

Chapter 5

5. Implementation

“JUQuery " is a platform that provides us with complete information about campus life from a student point of view, and academic information from a college point of view.

It features a familiar and well-thought-out and attractive user interface, much like posting a query, giving a response to a query, helping a student through a personal chat if required.

A user has complete freedom to post any query, and can also make a funny post in a legal way. Also students can register/ host any campus related events and also can take the feedback from other students through this platform.

So, the following techniques have been taken into the implementation of the web application:

5.1. Functional programming using functional components

- Functional programming is a concept of creating pure functions for software logic.
- It avoids concepts of mutable data and a shared state as used in Object-oriented programming. The functional programming is based more on expressions and declarations rather than the statements.
- The functional programming depends only on the arguments passed to the function.
- In React, a functional component is a plain JavaScript function that returns JSX.
- In functional programming, data cannot be stored in objects and it can only be transformed by creating functions.
- Functional components lack a significant amount of features as compared to class-based components. The gap is made up with the help of a special ReactJS concept called “hooks”. Hooks are special functions that allow ReactJS features to be used in functional components.
- So, in our project we have incorporated functional programming using functional components where we pass data as shared state through functional arguments.

5.2. State management using Redux.js library

- Redux is a predictable state container designed to help us write JavaScript apps that behave consistently across client, server, and native environments and are easy to test.
- With Redux, the state of your application is kept in a store, and each component can access any state that it needs from this store.
- State management is essentially a way to facilitate communication and sharing of data across components.
- It creates a tangible data structure to represent the state of your app that you can read from and write to.
- As it's mostly used as a state management tool with React, we have taken advantage of it to manage states of our platform event triggers such as user login/logout, show/hide popup, user inputs to provide a user friendly experience.

5.3. Specific re-rendering using ReactDOM in React.js

- In React every UI piece is a component, and each component has a state.
- React follows the observable pattern and listens for state changes.
- When the state of a component changes, React updates the virtual DOM tree.
- Once the virtual DOM has been updated, React then compares the current version of the virtual DOM with the previous version of the virtual DOM. This process is called "diffing".
- Once React knows which virtual DOM objects have changed, then React updates only those objects, in the real DOM.
- This makes the performance far better when compared to manipulating the real DOM directly.
- This makes React standout as a high performance JavaScript library.
- ReactDOM.render() controls the contents of the container node you pass in. Any existing DOM elements inside are replaced when first called. Later calls use React's DOM diffing algorithm for efficient updates.
- We have taken advantage of this React.js feature to re-render only those parts of the website which require a refresh, saving a lot of data usage from the user' experience with our platform.

5.4. Secure API authentication using Firebase

- Firebase Authentication is an extensible token-based auth system and provides out-of-the-box integrations with the most common providers such as Google, Facebook, and Twitter, among others.
- It enables us to use custom claims which we'll leverage to build a flexible role-based API.
- The Firebase API is secured via tokens—in order to generate such a token, we need to call Firebase's Client SDK and log in with a valid user/password credential.
- When successful, Firebase will send a token back in the response which we can then add to the header of any following request we want to perform.
- We take leverage over this secure features of Firebase Authentication API to keep our users secure and gain their trust over our platform's user experience
- Firebase gives us complete control over authentication by allowing you to authenticate users or devices using secure JSON Web Tokens (JWTs).
- We generate these tokens on your server, pass them back to a client device, and then use them to authenticate via the signInWithCustomToken() method To reduce problems promoting code changes from development to staging to production, instead of including API keys in the code itself, either set them as environment variables or include them in a configuration file.

5.5. Real time fetching data using Firebase Firestore database

- Firebase data is retrieved by either a one time call to GetValueAsync() or attaching to an event on a Firebase Database reference.
- The event listener is called once for the initial state of the data and again anytime the data changes.
- Firebase is a Backend-as-a-Service (Baas). It provides developers with a variety of tools and services to help them develop quality apps, grow their user base, and earn profit.
- It is built on Google's infrastructure.
- It is categorized as a NoSQL database program, which stores data in JSON-like documents.
- Stores data as collections of documents.
- Simple data is easy to store in documents, which are very similar to JSON. Complex, hierarchical data is easier to organize at scale, using sub collections within documents.
- Requires less denormalization and data flattening.
- Firestore also features richer, faster queries and scales further which is very helpful with Real time fetching of data from backend and rendering data on front-end with a very low latency which makes the user experience very effective.

This is how we have created the “JU Query” Web application, by using these technologies and techniques along with testing each component of the codebase locally and in production using in-built react-testing-library.

For each feature, a React component is implemented to follow the separation of concern principle which is connected together using Dependency injection and component re-rendering to make coding implementation effective along with ease to extend and maintain in the future making it scalable and flexible.

This component structure provides the user with on-time delivery of update releases and upgrades.

Chapter 6

6. Results and Discussion

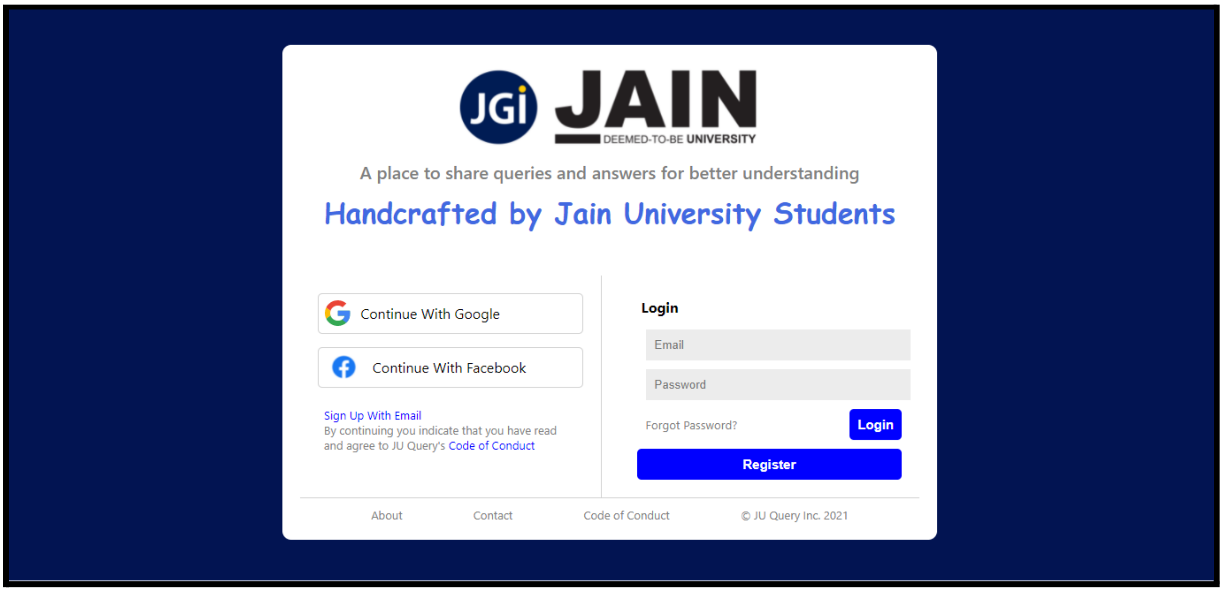
"JU Query" web application is a deliverable for students and faculties of the college incorporated with production-grade coding standards. This application is quite composite in nature and requires certain technical prerequisites to understand the implementation used in the background.

The web application is intended for college students and faculties making them the target audience for this platform and mainly focuses on promoting virtual interactions and helps each and every student to know what is exactly going on the college campus.

The technology stack used in the process of preparation of this:

- HTML5
- CSS3
- JavaScript with ES6 features
- NPM
- React JS
- Redux JS
- React-testing-library
- Material-UI
- Git & GitHub
- Firebase Firestore
- Firebase Authentication
- Firebase Deploy

6.1 Registration page



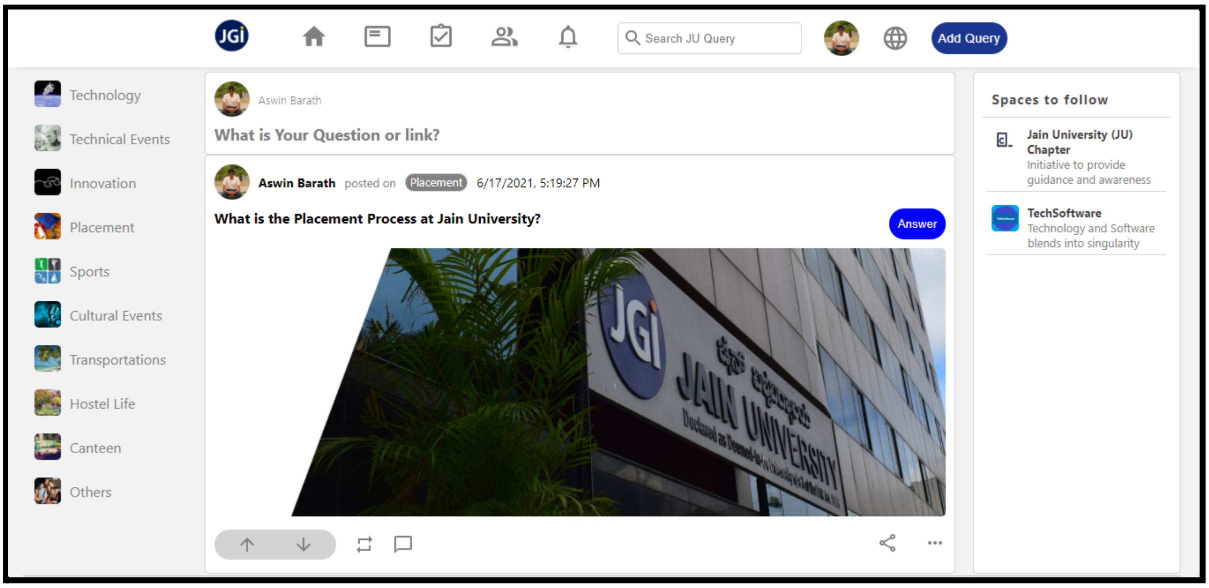
A place to share queries and answers for better understanding

Handcrafted by Jain University Students

Sign Up With Email
By continuing you indicate that you have read and agree to JU Query's Code of Conduct

Figure 6.1: A screenshot of registration page of JU Query

6.2 Home page



Aswin Barath

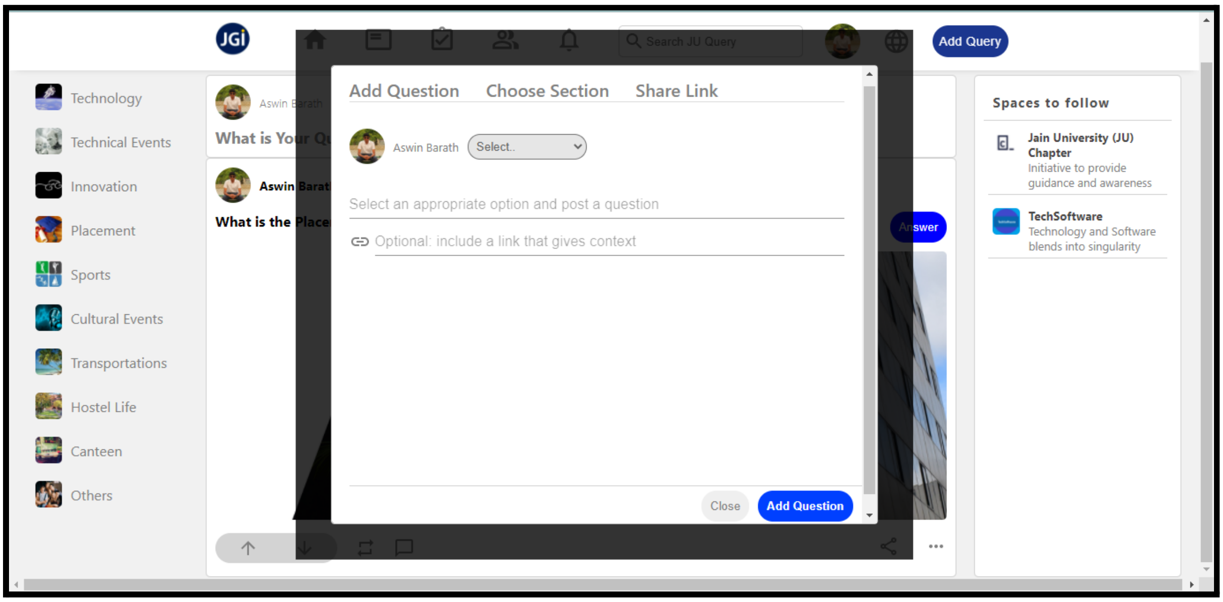
What is Your Question or link?

Aswin Barath posted on Placement 6/17/2021, 5:19:27 PM

What is the Placement Process at Jain University?

Figure 6.2: A screenshot of Live Question & Answer feature on JU Query web app

6.3 Add Query feature



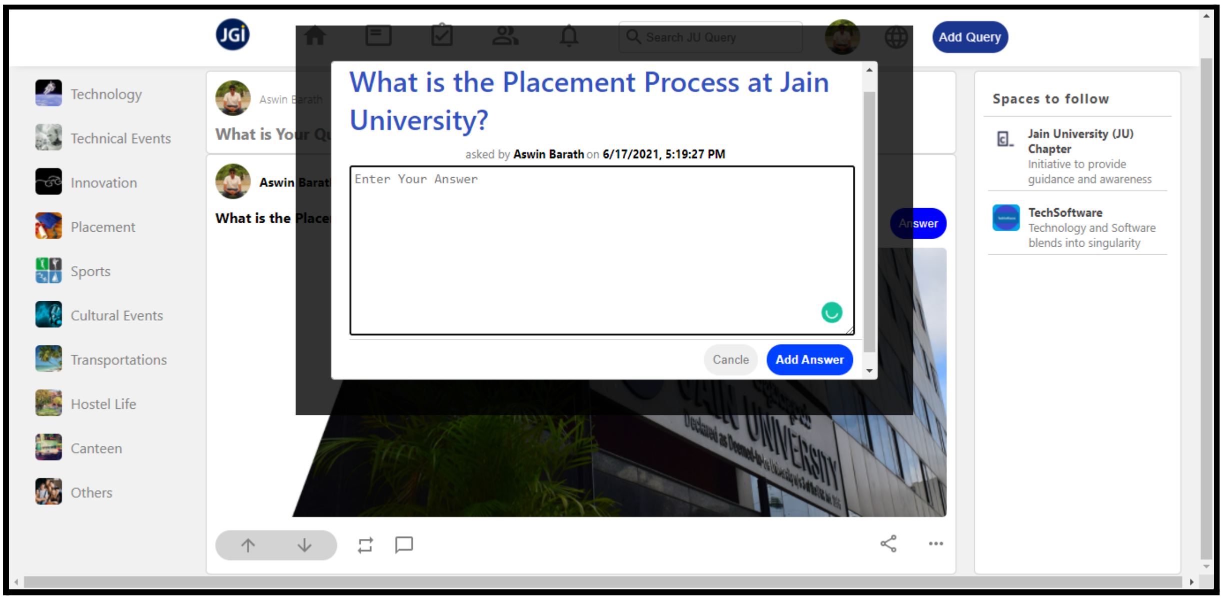
Aswin Barath

Select an appropriate option and post a question

Optional: include a link that gives context

Figure 6.3: A screenshot of Live Add query on JU Query web app

6.4 Add Answer feature



What is the Placement Process at Jain University?

asked by Aswin Barath on 6/17/2021, 5:19:27 PM

Enter Your Answer

Spaces to follow

Jain University (JU) Chapter
Initiative to provide guidance and awareness

TechSoftware
Technology and Software blends into singularity

Figure 6.4: A screenshot of Live Add Answer feature on JU Query web app

Chapter 7

7. CONCLUSIONS AND FUTURE SCOPE

In this project, we have done an application that helps the students of a college to get to know about their academic and non-academic information from the students perspective.

And every student of the college can post any query they want to post and anyone who wants to answer the post can do that. This process enables students across the campus even from different departments to interact with each other and get collaborative knowledge.

And we have 9 different kinds of categories where we can have questions about and if the question the student wants to ask is not listed in the type of the category he can select others and can still post.

And in the future, there will be features like a chat option where students in the college can connect to a fellow student and can have a conversation with the student. And also students can organize an event through this application and students can register for the event they wanted to through this application.