|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Software Design Specifications**  ***[FAST CENTRAL HUB]***  **Version: [1.5]**   |  |  | | --- | --- | | Project Code | F22-1206 | | Supervisor | Abdul Aziz | | Co Supervisor | - | | Project Team | K191048 Amman Soomro  K191118 Naba Jafri  K191090 Mashood Nadeem | | Submission Date | 25 November 2022 | |   **[Instructions]**   * *No section of template should be deleted. You can write ‘Not applicable’ if a section is not applicable to your project. But all sections must exist in the final document.* * *All comments/examples mentioned in square brackets ([]) are in the template for explanation purposes and must be replaced / removed in final document.* * *This’ Instruction’ section should also be removed in final document.* * *MS-Word Reviewing feature must be used to get the document reviewed by PMs or supervisors.*       **Document History**  *[Revision history will be maintained to keep a track of changes done by anyone in the document.]*   |  |  |  |  | | --- | --- | --- | --- | | Version | Name of Person | Date | Description of change | | 1.0 | Naba Jafri | 10 Nov | Started Working on the Document | | 1.1 | Mashood Nadeem | 15 Nov | Added Further Details | | 1.2 | Naba Jafri | 16 Nov | Added Data Dictionaries | | 1.3 | Amman Soomro | 20 Nov | Added Software Level Architecture | | 1.4 | Naba Jafri | 21 Nov | Added ERD and Sequence Diagram | | 1.5 | Amman Soomro, Mashood Nadeem, Naba Jafri | 23 Nov | Reviewed the whole document, proofreading and made a few required changes |         **Distribution List**  *[Following table will contain list of people whom the document will be distributed after every sign-off]*   |  |  | | --- | --- | | **Name** | **Role** | | Sir Abdul Aziz | Supervisor | | - | Co Supervisor | | Dr. Usama, Ms. Bakhtawar | Jury Members |       **Document Sign-Off**  *[Following table will contain sign-off details of document. Once the document is prepared and revised, this should be signed-off by the sign-off authority.*  *Any subsequent changes in the document after the first sign-off should again get a formal sign-off by the authorities.]*   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Version** | **Sign-off Authority** | **Project Role** | **Signature** | **Sign-off Date** | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |

**Document Information**

|  |  |
| --- | --- |
| **Category** | **Information** |
| Customer | FAST-NU |
| Project | <FAST CENTRAL HUB> |
| Document | Software Design Specification |
| Document Version | 1.0 |
| Status | Draft |
| Author(s) | Naba Jafri |
| Approver(s) |  |
| Issue Date | 25-Nov-22 |
| Document Location | FAST NUCES, Main Campus, Karachi |
| Distribution | Advisor  Project Coordinator’s Office (through Advisor) |

**Definition of Terms, Acronyms and Abbreviations**

*[This section should provide the definitions of all terms, acronyms, and abbreviations required to interpret the terms used in the document properly. ]*

|  |  |
| --- | --- |
| **Term** | **Description** |
| ASP | Active Server Pages |
| DD | Design Specification |
| Teacher | Used Interchangeably with Faculty in the SRS |
| Faculty | Used Interchangeably with Teacher in the SRS |
|  |  |
|  |  |
|  |  |
|  |  |

**Table of Contents**

Table of Contents

[1 Introduction 8](#_Toc120290668)

[1.1 Purpose of Document 8](#_Toc120290669)

[1.2 Intended Audience 8](#_Toc120290670)

[1.3 Document Convention 8](#_Toc120290671)

[1.4 Project Overview 8](#_Toc120290672)

[1.5 Scope 8](#_Toc120290673)

[2 Design Considerations 10](#_Toc120290674)

[2.1 Assumptions and Dependencies 11](#_Toc120290675)

[2.2 Risks and Volatile Areas 11](#_Toc120290676)

[3 System Architecture 11](#_Toc120290677)

[3.1 System Level Architecture 12](#_Toc120290678)

[3.2 Software Architecture 12](#_Toc120290679)

[4 Design Strategy 14](#_Toc120290680)

[5 Detailed System Design 15](#_Toc120290681)

[5.1 Database Design 15](#_Toc120290682)

[5.1.1 ER Diagram 15](#_Toc120290683)

[5.1.2 Data Dictionary 16](#_Toc120290684)

[5.2 Application Design 21](#_Toc120290685)

[5.2.1 Sequence Diagram 21](#_Toc120290686)

[5.3 State Diagram 30](#_Toc120290687)

[6 References 31](#_Toc120290688)

[7 Appendices 32](#_Toc120290689)

# Introduction

## Purpose of Document

The purpose of the document is to give a detailed overview of our project along with its goals and parameters to: our supervisor, the members of the jury and to ensure that the software product meets the respective requirements.

## Intended Audience

This document will be used by the supervisor along with the jury members to evaluate the project approved.

## Document Convention

The font used for the headings and the body is Cambria. The headings have a font size of 16 while the paragraphs have a font size of 14. This format is maintained throughout the document.

## Project Overview

Our web application, “FAST Central Hub” provides a platform for users to gather all student-useful information. The dashboard of our project consists of A management page (for Admins). The users of the application can view any information related to course, faculty members, projects, departments, societies, and important announcements. The important announcements can be regarding Internships, Events, Job Opportunities, and Workshops. The students can provide feedback on courses and faculty members. As compared to current platforms which provide a niche solution to a niche problem in evaluating software needs and its development, our web application attempts to bring together multiple features together to reduce complexity levels, high intense communication needs, and remove redundancy in information sharing, alerts, events, updates, and so on. The basic design approach of our web application is Function oriented approach as we are decomposing our model into sets of interacting modules where each module has its own clearly defined function. Hence, it is developed from a functional viewpoint.

## Scope

Our web application, provides a platform to users to gather all student-useful information. The dashboard of our project consists of Logged in user information, a management page (for Admins), Navigation Menu to navigate around the Website. The users of the application can view any information related to course, faculty members, projects, departments, societies, and important announcements. The important announcements can be regarding Internships, Events, Job Opportunities, and Workshops. The students can provide feedback on courses and faculty members. As compared to current platforms which provide a niche solution to a niche problem in evaluating software needs and its development, our web application attempts to bring together multiple features together to reduce complexity levels, high intense communication needs, and remove redundancy in information sharing, alerts, events, updates, and so on.

Our system is not completely related to any learning management system as there is no concept of submissions, attendance, course registration, teachers marking anything example, course results.

# Design Considerations

For purposes of development and use, the system must always be dependable, efficient, adaptable, and flexible. In order to address the key problems that existed with the system's design on a more granular level, we spent a lot of time working toward a better design flow and structure for the project. Some of the basic design issues that we came up with are:

1. How might we interact with the database?
2. How can we control our different views between admin and user?
3. What kind of architecture do we want to base our front-end and back-end development on?
4. What would we use to build rapid User interface?
5. How can our users successfully interact with our system for example, in providing any feedback?
6. How can we increase reliability?

There are other design considerations, like:

1. Since it is a web application, we care about compatibility on different browsers enough so that it does not break for our users.
2. Extensibility - The project is created in a way that makes it simple to add new modules with additional functionality without affecting or changing the contents of existing modules.
3. Modularity - Modules are designed to be well-defined, independent components which would lead to better maintainability. Dependencies are clearly defined where needed and are well-intentioned and well-contained.
4. Fault-tolerance - this factor is not really relevant because the program is not yet intended to extend outside local development environments.
5. Reusability – System will be developed through classes and modules which can be reused later on.
6. Performance- By developing a light weighted web app the performance of software will be enhanced.
7. Security - The program can withstand and fend against harmful influences and actions.
8. Usability – By implying the UI/US standards the user will be able to understand and learn the software more easily.
9. Portability - The program will work on many mobile, desktop, and laptop platforms.

In such a scenario, our initial steps were to,

1. Complete and choose the primary project objectives that we were going to work on user requirements analysis and engineering.
2. Work breakdown structure will be used to check what will be the distribution of the tasks.
3. Review the work as necessary.

For our project, we will be using the waterfall methodology. As our milestones our defined, requirements, and scope are well understood, there will be little to no change in our requirements. Hence, this method will enable us to handle design issues in accordance with our workflow and design, which will assist in creating a code base that is extendable, manageable, adaptive, and flexible for future development.

## Assumptions and Dependencies

It is assumed that the roles of users in our system are defined and users of the system have basic knowledge of browsing and using computers. We are using a generalized and familiar user interface that can be used easily by anyone.

Our system's dependence may be assessed in such a way that we must collect the data from our university before we can test the application. Furthermore, the university's approval to allow us to test the data is essential to the success of our web application. Lastly, there are some university norms that we must follow. For instance, the processes used in our online application have been approved by our university.

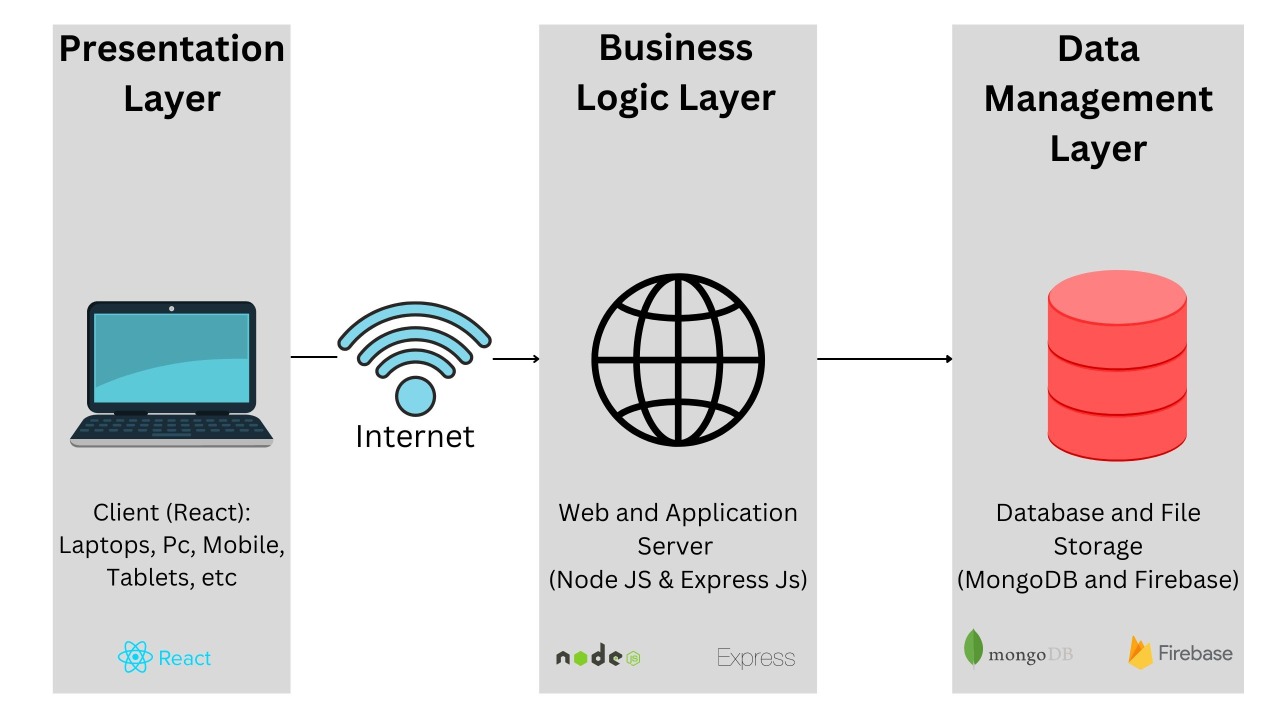
## Risks and Volatile Areas

There is little to no time to plan, develop, and rehearse a fallback option. As a result, we should not face any further changes in the requirements at this time, and our technology selection is likely to remain consistent throughout the project - we have not encountered any circumstances in which we would not use our current set of tools and implement additional functionalities with them. If any new requirements emerge, our first step would be to determine whether it is a priority, and if so, when it becomes a priority to complete first.

# System Architecture

Our application is based on three-tier architecture where layers are arranged into three logical and physical layers. The Operating system used for this app is Any Os with the internet. As discussed, the application is built using the MERN stack (MongoDB, Express JS, React Js, and Node JS) which is also a three-tier architecture where React Js represents the Presentation layer; all the user interfaces are defined at this layer. React Js makes frontend development easy and allows to build of user-friendly interfaces. For implementing business logic Express JS will be used to build the server along with node JS. Furthermore, the npm package will be used for better building of business logic. However, for the storage of Firebase and fetching of data MongoDB will be used, and to connect the database with the server, an npm package mongoose will be used. Furthermore, APIs will be used for collecting data from external sources. The presentation layer will contact the business layer which further will communicate with the data access layer to provide the information the client needs.

## System Level Architecture

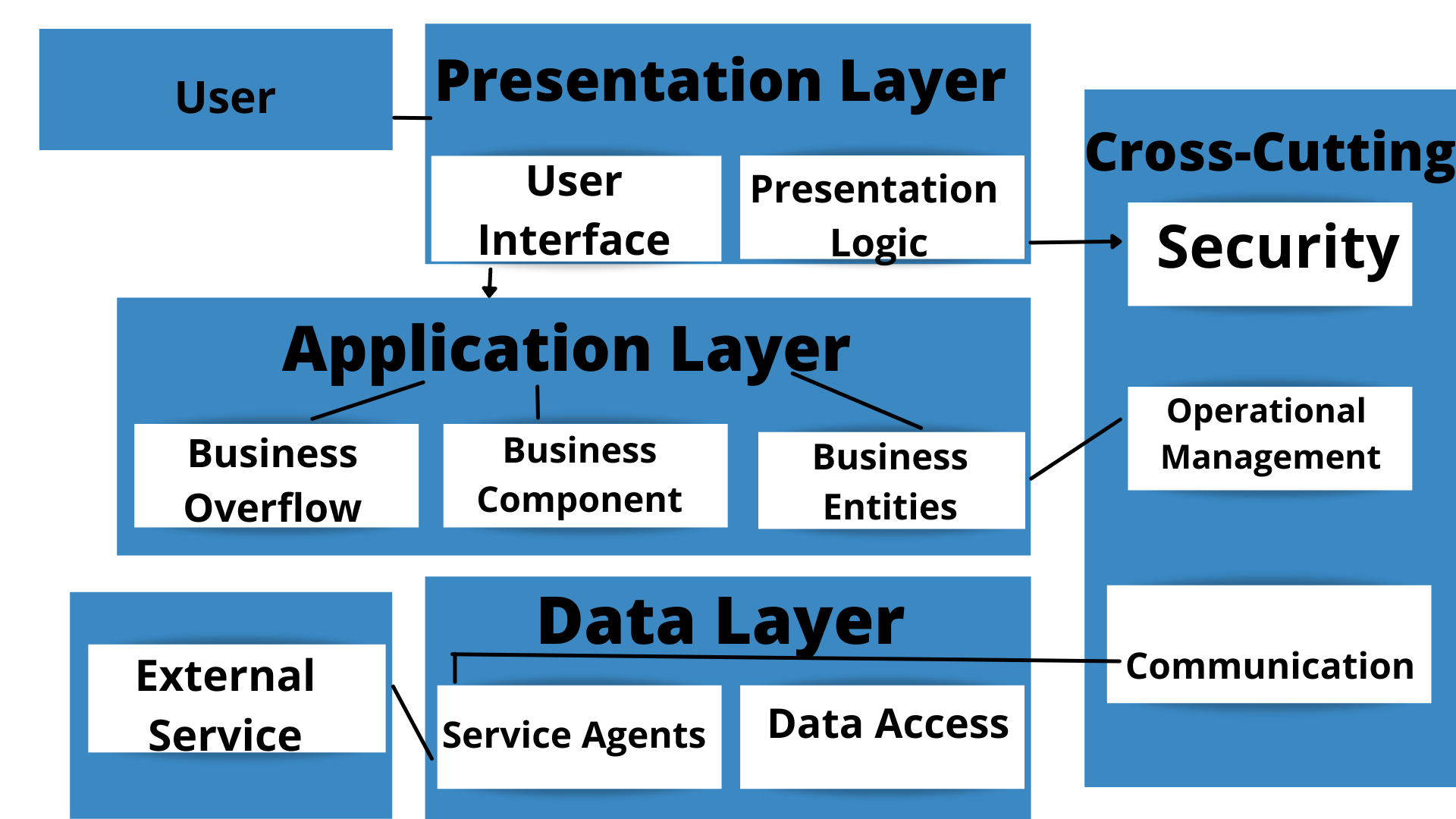


## Software Architecture

The architecture on which the project will be created is Client-Server architecture where a proper web server will be created to address the request made through the application. Also, the project will be implemented by creating an application using the following technologies:

1. React (Front-End)
2. Node JS and Express JS (Environment)
3. NoSQL (MongoDB Database)

Diagram:

**

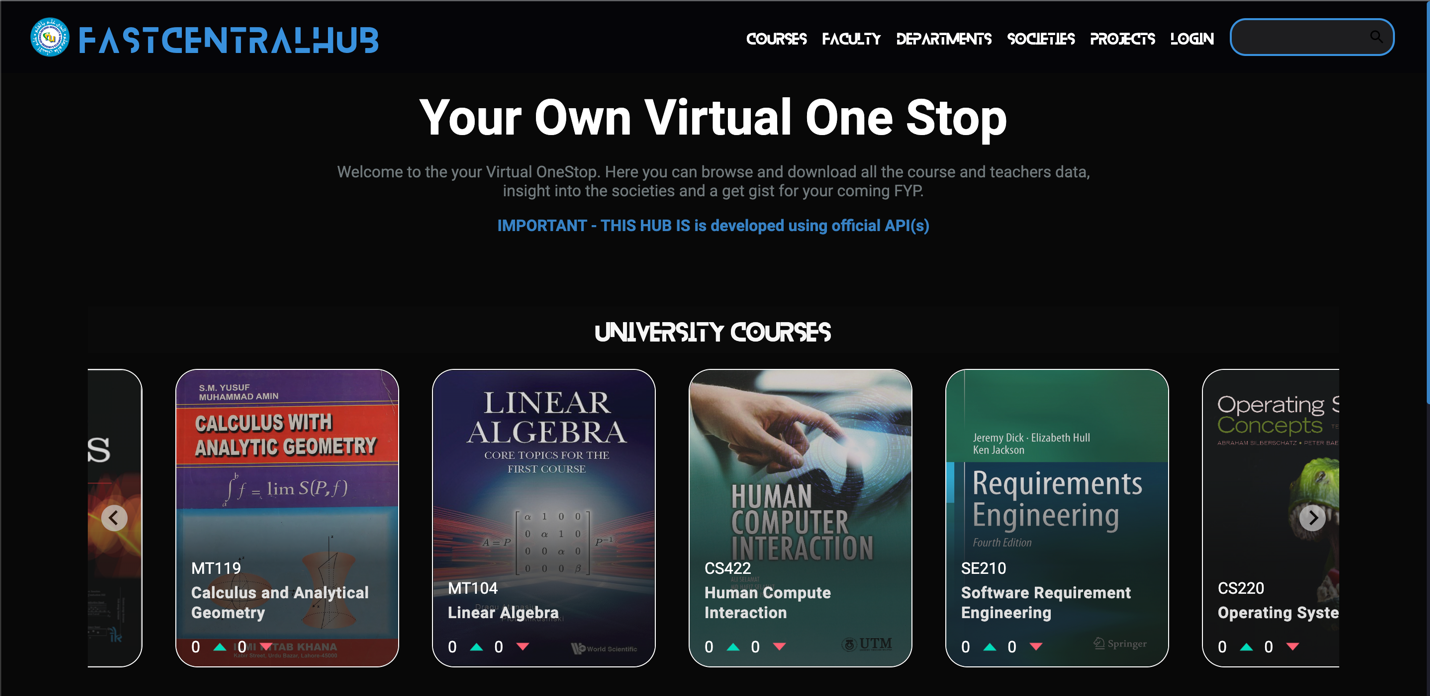
# Design Strategy

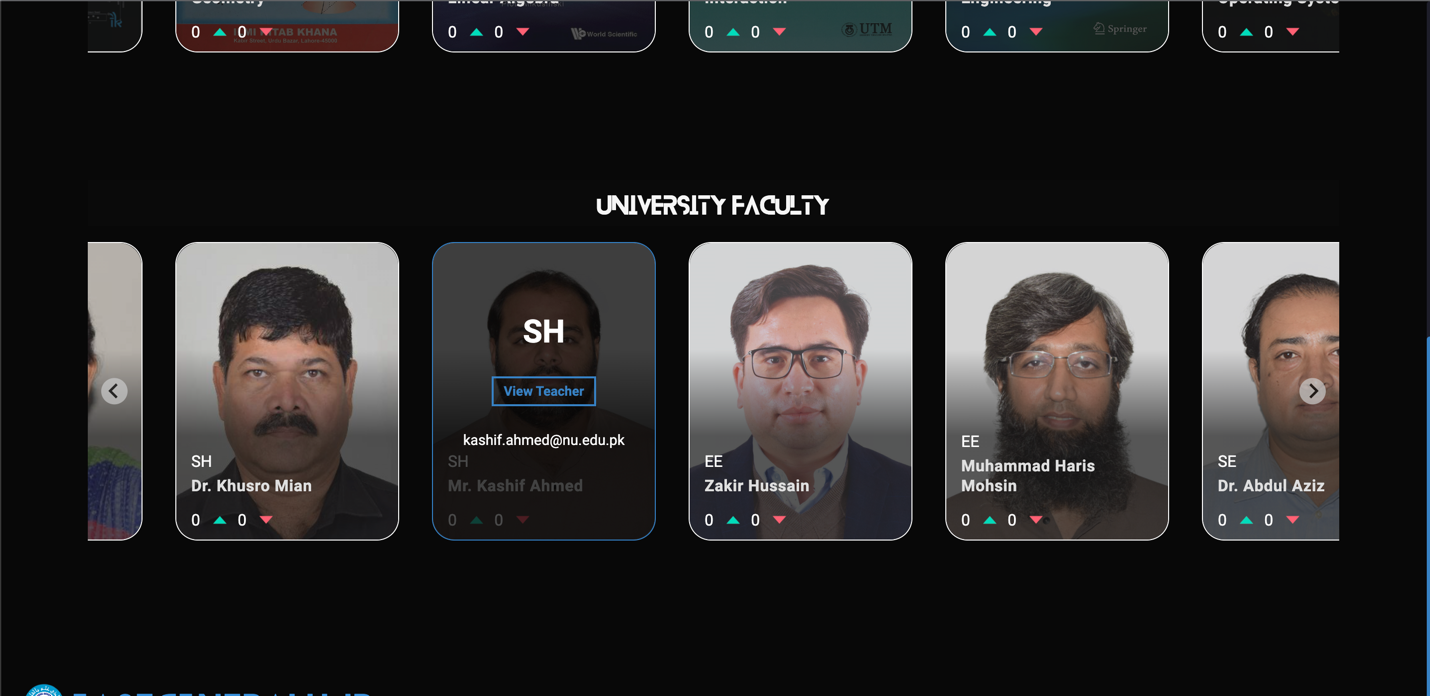
Our system is currently designed to solve the basic problems of unavailability of data. In future, we can extend it to add a university map to help students and new staff of university not waste their time. Moreover, we can also integrate a timetable notifying system in our web app. We have applied basic UI/UX standards in our system as it sets the proper hierarchy of our screens, provides feedbacks on actions of users, uses less or no technical terms, provides users the freedom to control their actions, usable by many people, flexible in such a way that every ( new or old ) user can use it easily.

In our web app we will be use MongoDB as it widely supported and is schema less. It also supports dynamic queries on documents using a document-based query language that's nearly as powerful as SQL. Moreover, there no complex joins and the structure of a single object is clear.

# Detailed System Design

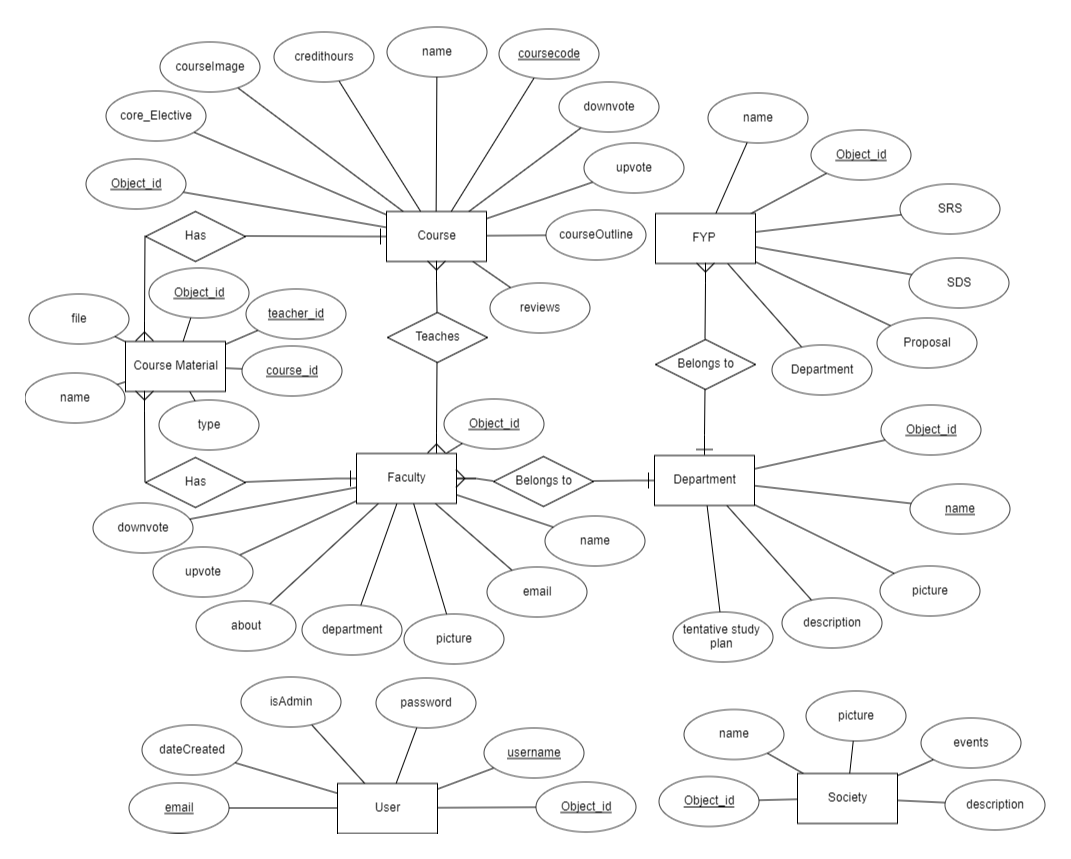
## GUI Design





## Database Design

### ER Diagram

**

### Data Dictionary

#### User Dictionary

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **User Dictionary** | | | | | | | |
| **Name** | | Users | | | | | |
| **Alias** | | None | | | | | |
| **Where-used/how-used** | | This is used to store the details of the registered users and use it at the time of logging | | | | | |
| **Content description** | | The Credentials of all the users are stored in this Data Dictionary | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null**  **able** | **Default Value** | **Key Type** |
| *Object\_id* | *This is the id created automatically by Mongo at the time of creation of object* | | *String* | *-* | *Yes* | *Auto Generated* | *Unique* |
| *username* | *Username of the user* | | *String* | *-* | *No* | *“”* | *Unique* |
| *Email* | *Email of the user account* | | *String* | *-* | *No* | *“”* | *Unique* |
| *Password* | *Password of the user account* | | *String* | *-* | *No* | *“”* |  |
| *isAdmin* | *This field defines whether the user is admin of a normal user* | | *Bool* | *-* | *No* | *False* |  |
| *dateCreated* | *This field is autogenerated at the time of creation of object.* | | *Date* | *-* | *No* | *Creation Date and time* |  |

#### Course Dictionary

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Course Dictionary** | | | | | | | |
| **Name** | | Courses | | | | | |
| **Alias** | |  | | | | | |
| **Where-used/how-used** | | This is used to store the details of the university courses | | | | | |
| **Content description** | | All the information about each course is available in this dictionary | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null**  **able** | **Default Value** | **Key Type** |
| *ÍObject\_id* | *This is the id created automatically by Mongo at the time of creation of object* | | *String* | *-* | *No* | *Auto Generated* | *Unique* |
| *coursecode* | *Course Code of the given Course* | | *String* | *-* | *No* | *“”* | *Unique* |
| *name* | *Name of the given Course* | | *String* | *-* | *No* | *“”* |  |
| *credithours* | *How many credit hours this course have* | | *Number* | *-* | *No* | *0* |  |
| *Core\_Elective* | *Is this a Core Course or an Elevtive Course* | | *Bool* | *-* | *No* | *Core* |  |
| *courseOutline* | *Course Outline for this Course* | | *Date* | *-* | *Yes* | *“”* |  |
| *courseImage* | *Image of this Course* | | *String* | *-* | *Yes* | *“”* |  |
| *Reviews* | *Reviews given to this course by the User* | | *String Array* | *-* | *Yes* | *“”* |  |
| *Upvotes* | *Upvotes given to this course by the User* | | *Number Array* | *-* | *Yes* | *0* |  |
| *Downvotes* | *Downvotes given to this course by the User* | | *Number Array* | *-* | *Yes* | *0* |  |

#### Department Dictionary

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Department Dictionary** | | | | | | | |
| **Name** | | Departments | | | | | |
| **Alias** | |  | | | | | |
| **Where-used/how-used** | | This is used to store the information about the departments of the university | | | | | |
| **Content description** | | All the information of each department is available in this dictionary | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null**  **able** | **Default Value** | **Key Type** |
| *Object\_id* | *This is the id created automatically by Mongo at the time of creation of object* | | *String* | *-* | *No* | *Auto Generated* | *Unique* |
| Name | *Name of the Department* | | *String* | *-* | *No* | *“”* | *Unique* |
| Picture | *Picture of the Department* | | *File* | *-* | *Yes* | *-* |  |
| *Description* | *Description About the Department* | | *String* | *-* | *No* | *“”* |  |
| *Tentative Study Plan* | *Tentative Study plan of each semester for this Department* | | *File* | *-* | *No* | *-* |  |

#### Society Dictionary

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Society Dictionary** | | | | | | | |
| **Name** | | Societies | | | | | |
| **Alias** | |  | | | | | |
| **Where-used/how-used** | | This is used to store the details of the university Societies | | | | | |
| **Content description** | | All the information about each Society is available in this dictionary | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null**  **able** | **Default Value** | **Key Type** |
| *Object\_id* | *This is the id created automatically by Mongo at the time of creation of object* | | *String* | *-* | *No* | *Auto Generated* | *Unique* |
| *name* | *Name of the given Society* | | *String* | *-* | *No* | *“”* |  |
| *picture* | *Picture of the given Society* | | *File* | *-* | *Yes* | *“”* |  |
| *events* | *Events that happen in this given society* | | *Array* | *-* | *No* | *“”* |  |
| *description* | *Description of the given Society* | | *String* | *-* | *No* | *“”* |  |

#### Project Dictionary

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Dictionary** | | | | | | | |
| **Name** | | Projects | | | | | |
| **Alias** | |  | | | | | |
| **Where-used/how-used** | | This is used to store the details of the university Final Year Projects | | | | | |
| **Content description** | | All the information about FYP(s) will be stored here | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null**  **able** | **Default Value** | **Key Type** |
| *Object\_id* | *This is the id created automatically by Mongo at the time of creation of object* | | *String* | *-* | *No* | *Auto Generated* | *Unique* |
| *name* | *Name of the given Society* | | *String* | *-* | *No* | *“”* |  |
| *SRS* | *SRS of the given Project* | | *File* | *-* | *Yes* | *“”* |  |
| *SDS* | *Events that happen in this given society* | | *File* | *-* | *No* | *“”* |  |
| *Proposal* | *Proposal of the given Project* | | *File* | *-* | *No* | *“”* |  |
| *Department* | *Department for this the course was developed for* | | *String* |  |  |  |  |

#### Society Dictionary

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Faculty Dictionary** | | | | | | | |
| **Name** | | Faculty | | | | | |
| **Alias** | |  | | | | | |
| **Where-used/how-used** | | This is used to store the details of the university Faculty | | | | | |
| **Content description** | | All the information about each Faculty is available in this dictionary | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null**  **able** | **Default Value** | **Key Type** |
| *Object\_id* | *This is the id created automatically by Mongo at the time of creation of object* | | *String* | *-* | *No* | *Auto Generated* | *Unique* |
| *name* | *Name of the given Faculty* | | *String* | *-* | *No* | *“”* |  |
| *email* | *Email of the given Faculty* | | *Email* | *-* | *No* | *“”* |  |
| *picture* | *Picture of the given Faculty* | | *File* | *-* | *Yes* | *“”* |  |
| *department* | *Department of the given Faculty* | | *String* | *-* | *No* | *“”* |  |
| *about* | *Description about the given Faculty* | | *String* | *-* | *Yes* | *“”* |  |
| *Upvotes* | *Upvotes given to this course by the User* | | *Number Array* | *-* | *Yes* | *0* |  |
| *Downvotes* | *Downvotes given to this course by the User* | | *Number Array* | *-* | *Yes* | *0* |  |

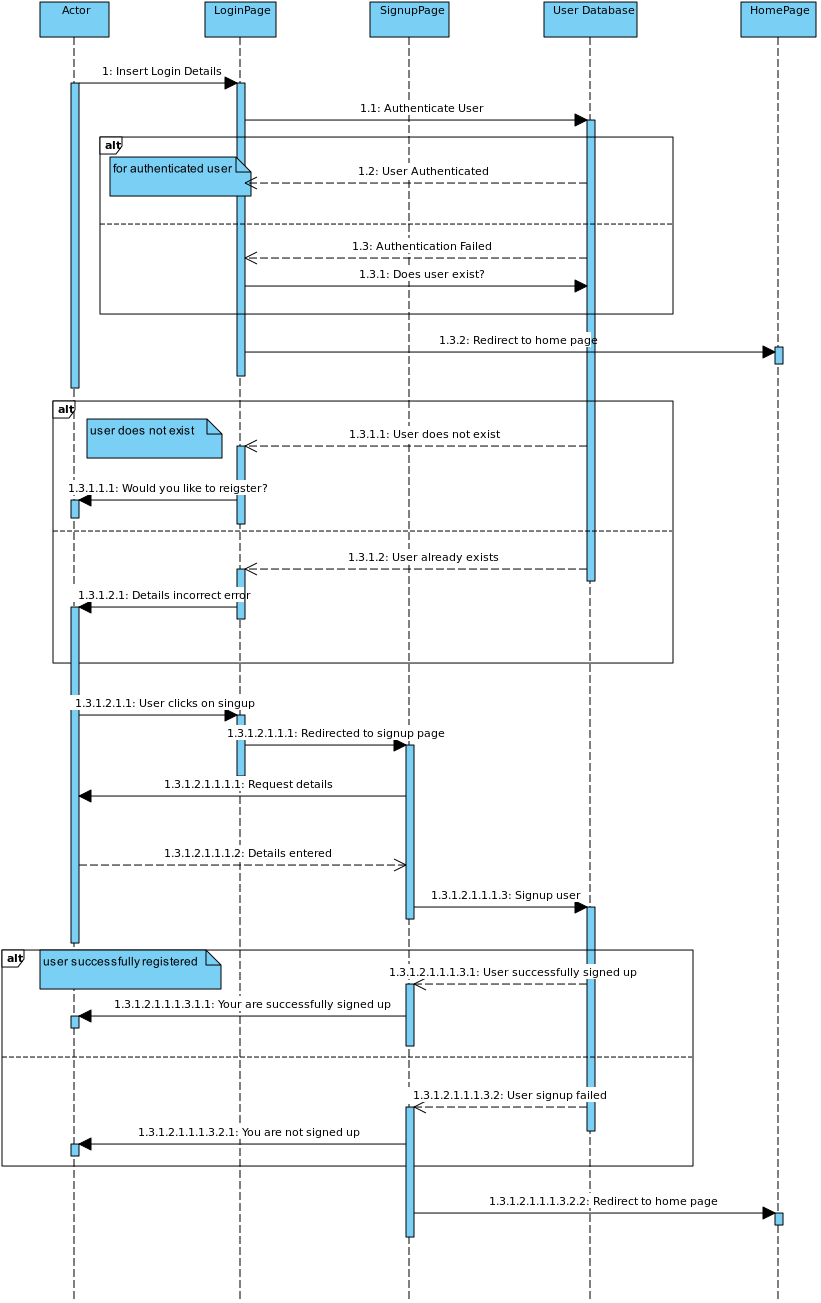
#### Material Dictionary

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Material Dictionary** | | | | | | | |
| **Name** | | Materials | | | | | |
| **Alias** | |  | | | | | |
| **Where-used/how-used** | | This is used to store the material of the university courses | | | | | |
| **Content description** | | All the material of each course is available in this dictionary | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null**  **able** | **Default Value** | **Key Type** |
| *Object\_id* | *This is the id created automatically by Mongo at the time of creation of object* | | *String* | *-* | *No* | *Auto Generated* | *Unique* |
| *File* | *The Material file uploaded by the admin* | | *File* | *-* | *No* | *“”* | *Unique* |
| *Name* | *Name of the given File* | | *String* | *-* | *No* | *“”* |  |
| *Type* | *Type of the file : Assignment/Quiz etc* | | *String* | *-* | *No* | *-* |  |
| *Course\_id* | *Id of the course this material belongs to* | | *String* | *-* | *No* | *“”* |  |
| *Teacher\_id* | *Id of the Teacher this material belongs to* | | *String* | *-* | *No* | *“”* |  |

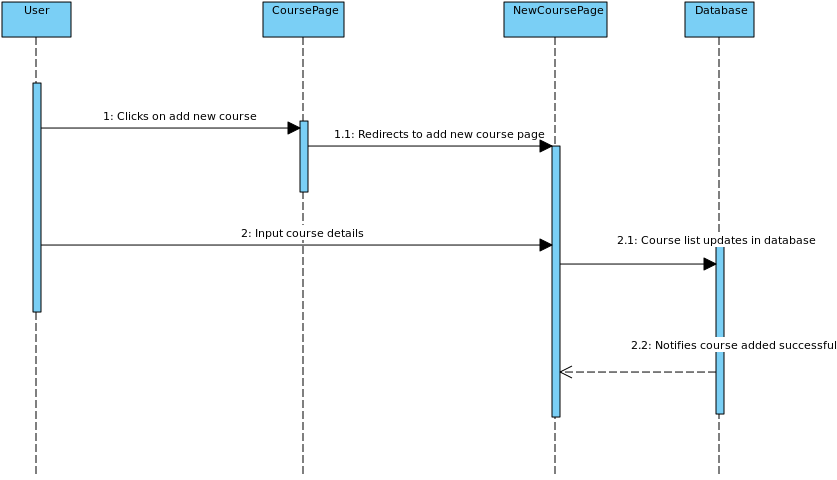
## Application Design

### Sequence Diagram

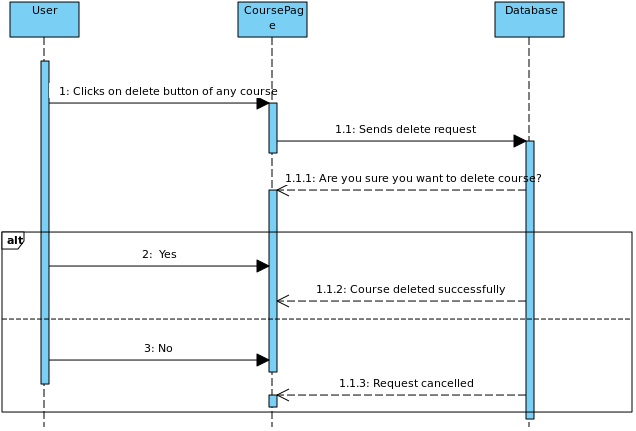
#### Login and Signup



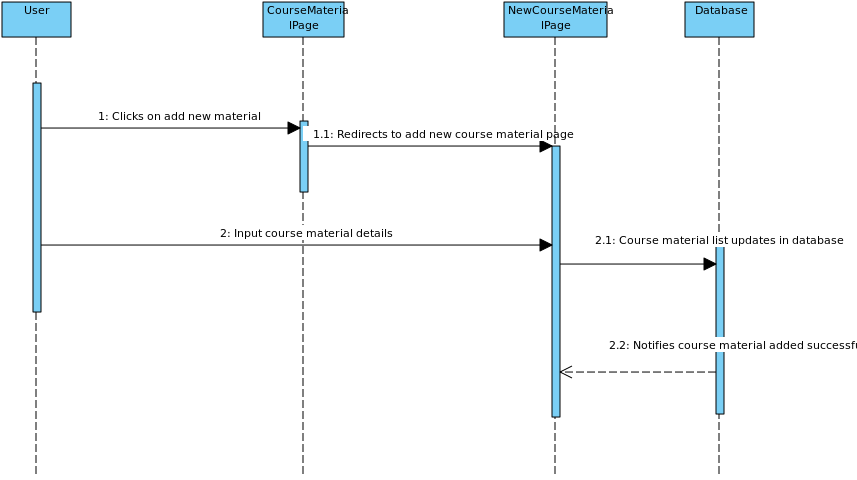
#### Course Add



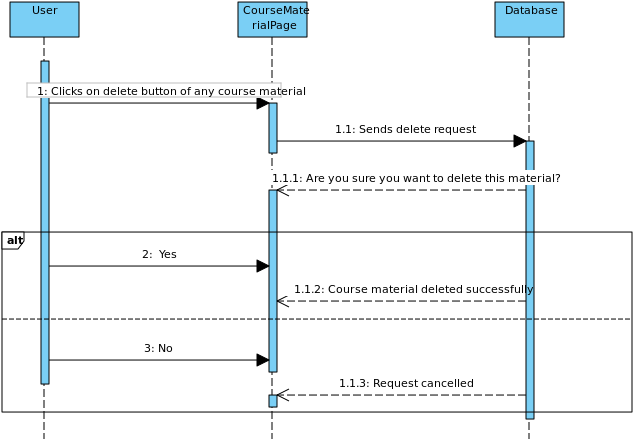
#### Course Delete

**

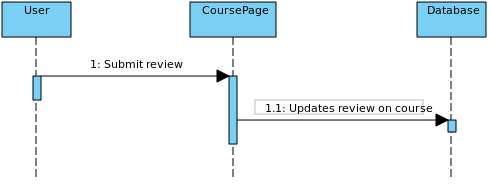
#### Course Material Add



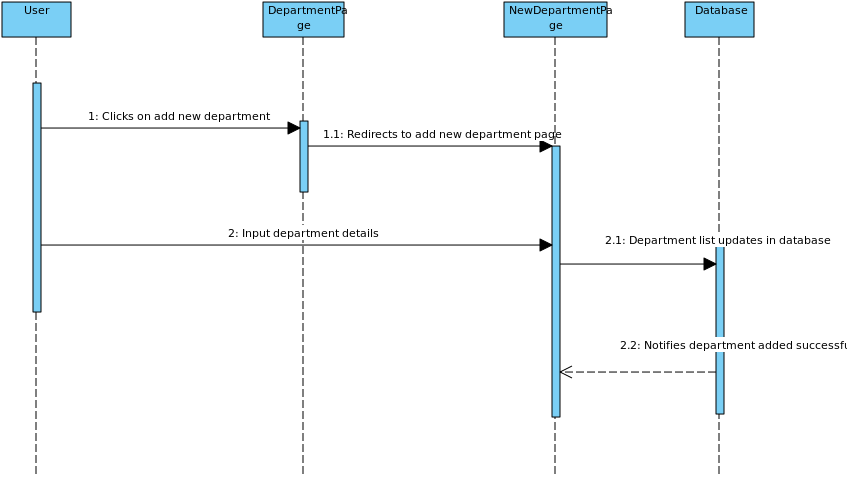
#### Course Material Delete



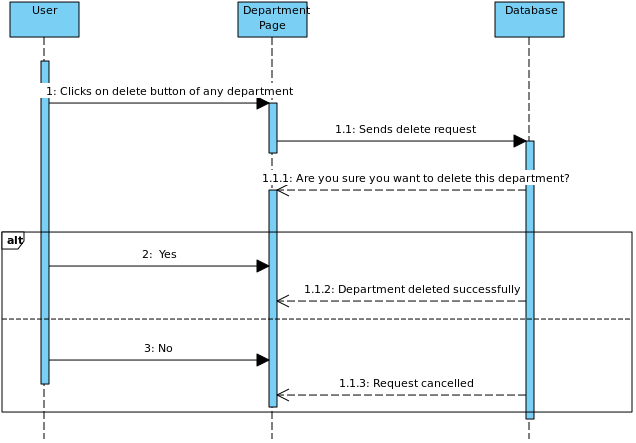
#### Course Review



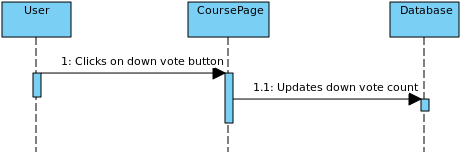
#### Department Add



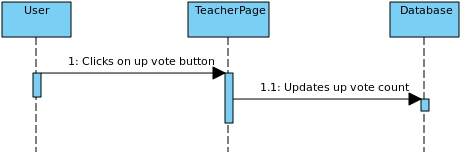
#### Department Delete



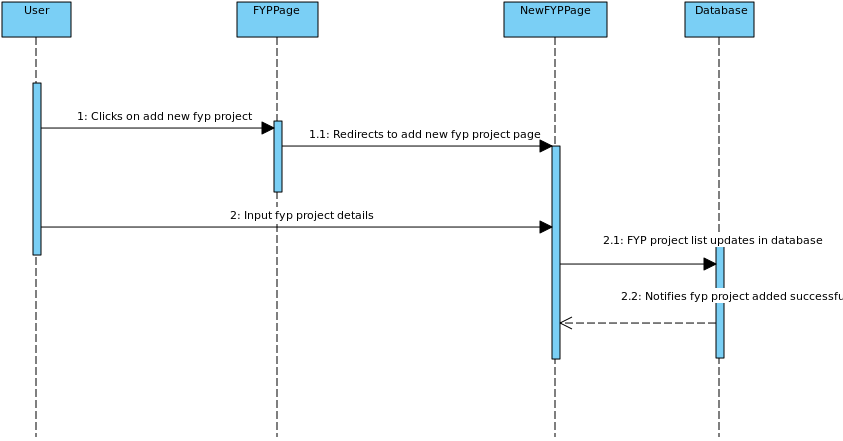
#### Down vote For Course



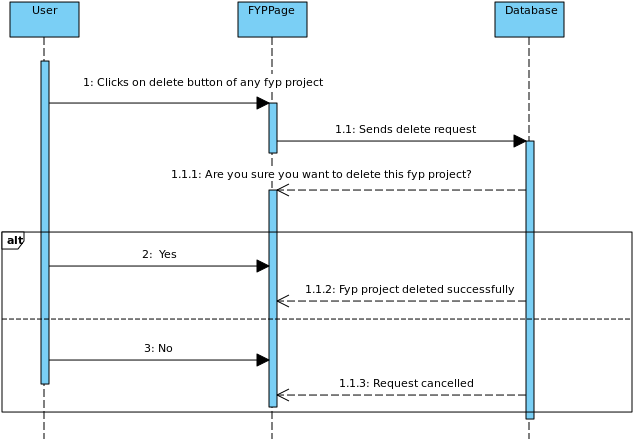
#### Down vote For Teacher



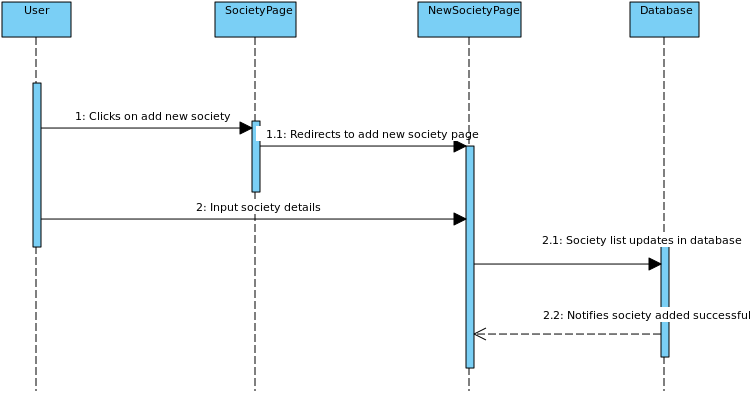
#### Final Year Project Add



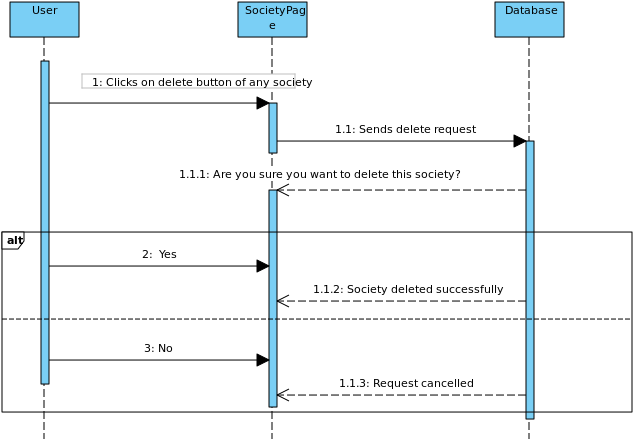
#### Final Year Project Delete



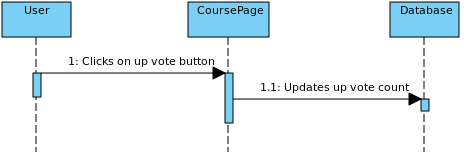
#### Society Add



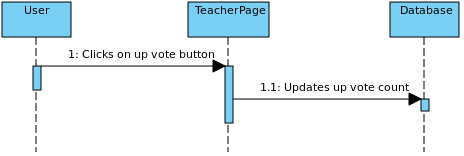
#### Society Delete



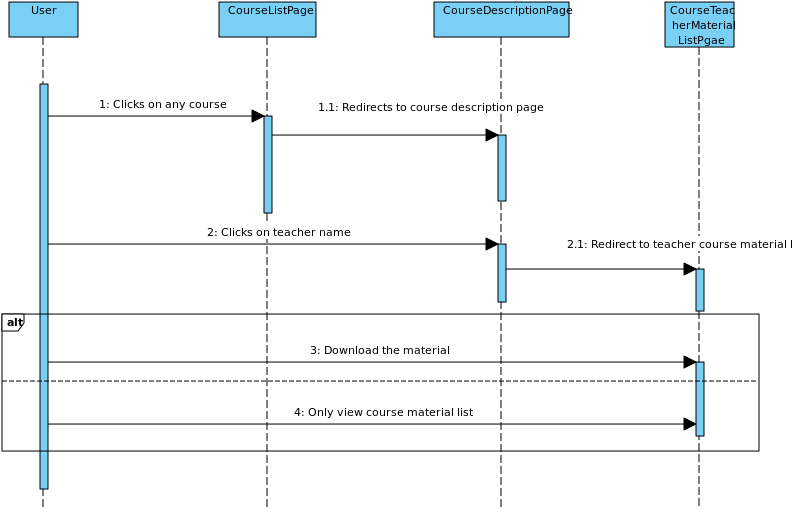
#### Up vote For Course



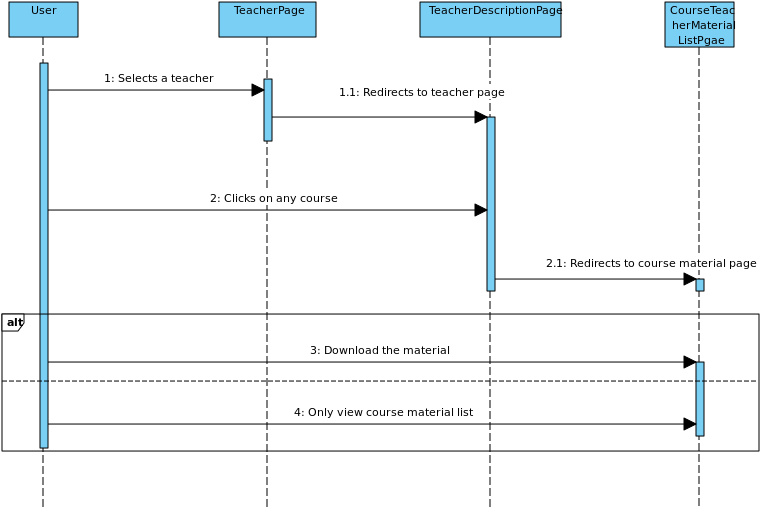
#### Up vote For Teacher



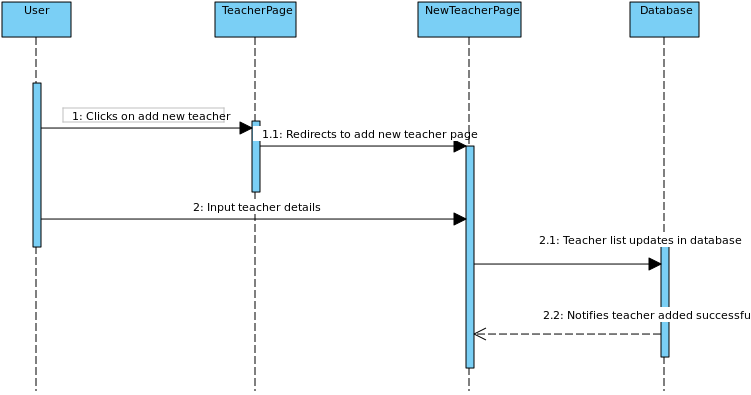
#### View Course Material by Course



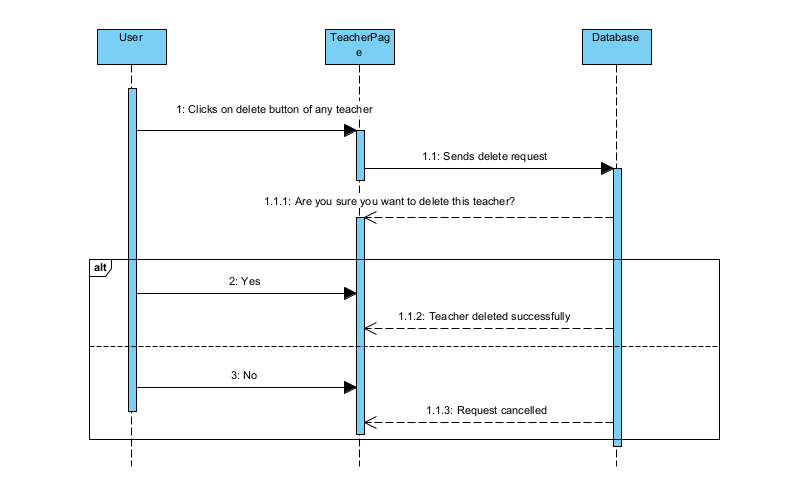
#### View Course Material by Teacher



#### Teacher Add



#### Teacher Delete



### State Diagram

# References

1. [https://www.mongodb.com](https://www.mongodb.com/cloud/atlas/lp/try4?utm_source=google&utm_campaign=search_gs_pl_evergreen_atlas_general-phrase_prosp-brand_gic-null_ww-multi_ps-all_desktop_eng_lead&utm_term=mongodb&utm_medium=cpc_paid_search&utm_ad=p&utm_ad_campaign_id=11295578158&adgroup=116363205048&gclid=CjwKCAiA7IGcBhA8EiwAFfUDsbSFRWU0eLA8QoBMUxpN03FLA_Rx-x7VabPyJFPKIs1CtTPRpmaYJhoCRN0QAvD_BwE)
2. <https://nodejs.org/en/docs/>
3. [https://reactjs.org](https://reactjs.org/docs/getting-started.html)’
4. <https://expressjs.com/>

# Appendices

*Not Applicable*