|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Software Design Specifications**  ***[BUILD EASE]***  **Version: [1.5]**   |  |  | | --- | --- | | Project Code | F23-139B | | Supervisor | Ms. Sobia Iftikhar | | Co Supervisor | - | | Project Team | Ayaz Hassan (20k-1044)  HussainTahir (20k-0185)  Syed Arsalan (20k-1718) | | Submission Date | 10th Dec 2023 | |   **[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 | Hussain Tahir | 28 Nov | Started Working on the Document | | 1.1 | Ayaz Hassan | 1 Dec | Added Further Details | | 1.2 | Ayaz Hassan | 3 Dec | Added Data Dictionaries | | 1.3 | Syed Arsalan | 5 Dec | Added Software Level Architecture | | 1.4 | Hussain Tahir | 6 Dec | Added ERD and Sequence Diagram | | 1.5 | Syed Arsalan | 8 Dec | 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** | | Ms. Sobia Iftikhar | Supervisor | | - |  | | Mr. Faisal Ali, Ms. Nida Munawwar | 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 | Build Ease |
| Document | Software Design Specification |
| Document Version | 1.0 |
| Status | Draft |
| Author(s) | Hussain Tahir, Syed Arsalan, Ayaz Hasan |
| Approver(s) |  |
| Issue Date | 3 Dec 2023 |
| Document Location | FAST NUCES, Main Campus, Karachi |
| Distribution | Advisor  Project Coordinator’s Office (through Advisor) |

**Definition of Terms, Acronyms and Abbreviations**

|  |  |
| --- | --- |
| **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](#_Toc154659657)

[1.1 Purpose of Document 8](#_Toc154659658)

[1.2 Intended Audience 8](#_Toc154659659)

[1.3 Document Convention 8](#_Toc154659660)

[1.4 Project Overview 8](#_Toc154659661)

[1.5 Scope 9](#_Toc154659662)

[2 Design Considerations 10](#_Toc154659663)

[2.1 Assumptions and Dependencies 11](#_Toc154659664)

[2.2 Risks and Volatile Areas 11](#_Toc154659665)

[3 System Architecture 12](#_Toc154659666)

[3.1 System Level Architecture 12](#_Toc154659667)

[3.2 Software Architecture 13](#_Toc154659668)

[4 Design Strategy 14](#_Toc154659669)

[5 Detailed System Design 15](#_Toc154659670)

[5.1 GUI Design 15](#_Toc154659671)

[5.2 Database Design 17](#_Toc154659672)

[5.2.1 ER Diagram 17](#_Toc154659673)

[5.2.2 Data Dictionary 18](#_Toc154659674)

[5.3 Application Design 23](#_Toc154659675)

[5.3.1 Sequence Diagram 23](#_Toc154659676)

[5.3.2 State Diagram 29](#_Toc154659677)

[1 References 31](#_Toc154659678)

[2 Appendices 32](#_Toc154659679)

# 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, “Build Ease ” provides a platform construction service providers and the clients in need to interact with each other. The system provides an interactive interface for effective communication and collaboration between the client and the service providers, it also provides a comparative data to help the users of the platform to make good and efficient decisions that will help them financially. The contractor and the client both will have a menu of options which will help them in their pursuit of selecting the suitable clients and the contractor. The user has to login, then there are many options to choose from such as view the running project , message the contractors, view the progress of the projects ,hire contractor and calculate cost of the project and also view daily prices. The contractor can add the members working on the projects, accept the hiring request, message clients, monitor progress and has some common functionalities with regard to client. The Admin will see the projects posted and monitor the listings and the service provider for potential

scams and monitor the environment.

## Scope

The scope of web application extends to all the clients who want to hire a contractor for the construction of their own houses, they can find contractors and the contractors can also reach to the clients with an interactive interface that provides data for the easy section for the suitable client or the contractor for the both type of the users, this also provides an interactive platform for the users with comparative and real time data, it can used by all the users, it also provides an interface to monitor the projects and see the daily prices and updates about the progress. This will also have an AI recommender system that will show users in their desired locations to connect with and chat with.

# 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 client and the service provider?
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?
7. What frameworks will be used for the recommender system in the

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 and the required documentation that is necessary and help with features

## Assumptions and Dependencies

We assume that the users off our website:

* Have a stable internet connection
* Understand English and Urdu
* Users for this website are local contractor
* Our system requires a hosting service that is able to connect to the servers.
* System depends on a stable internet connection.
* Requires an active database connection to save the changes made.

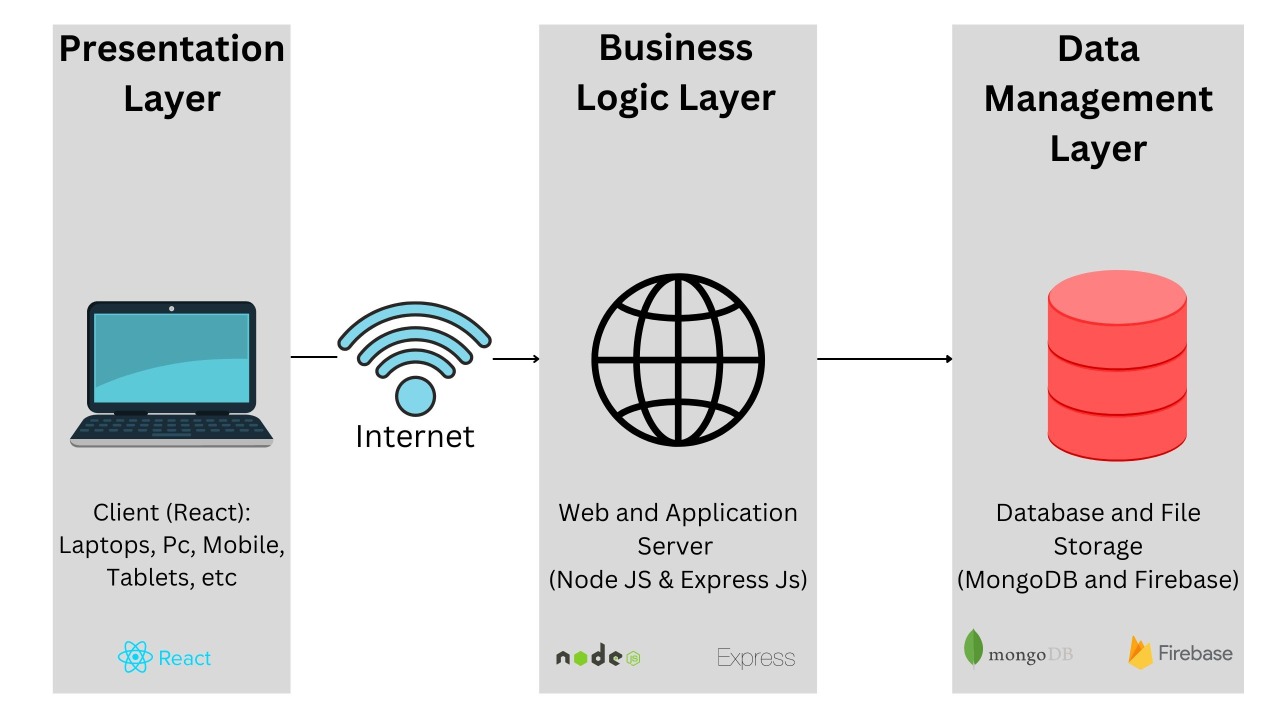
## Risks and Volatile Areas

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 if the new requirements emerge we will determine if they are feasible or if they are a priority then they will be the part of the system

# 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

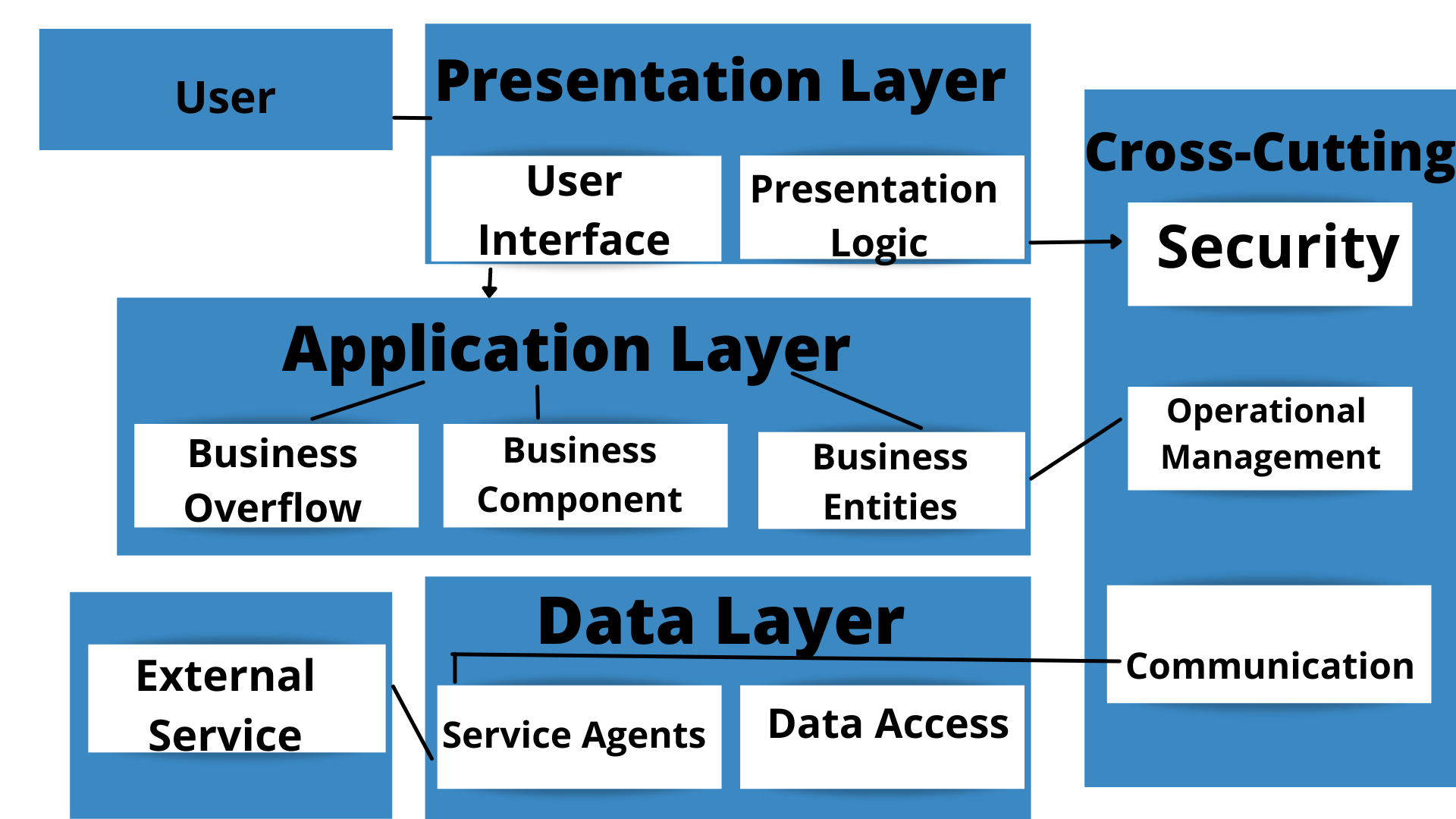


## Software Architecture

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. MySql (relatoional database)

Diagram:

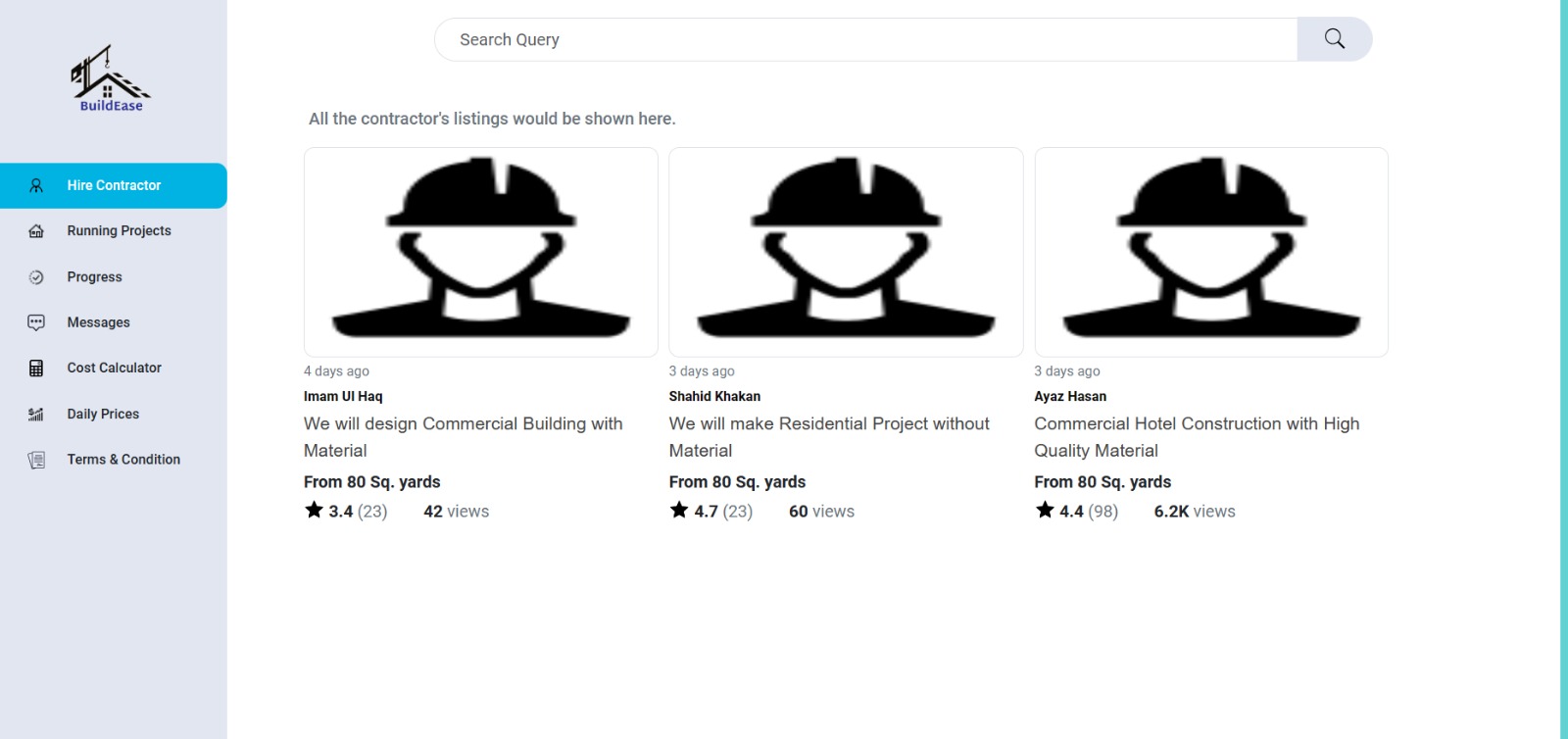
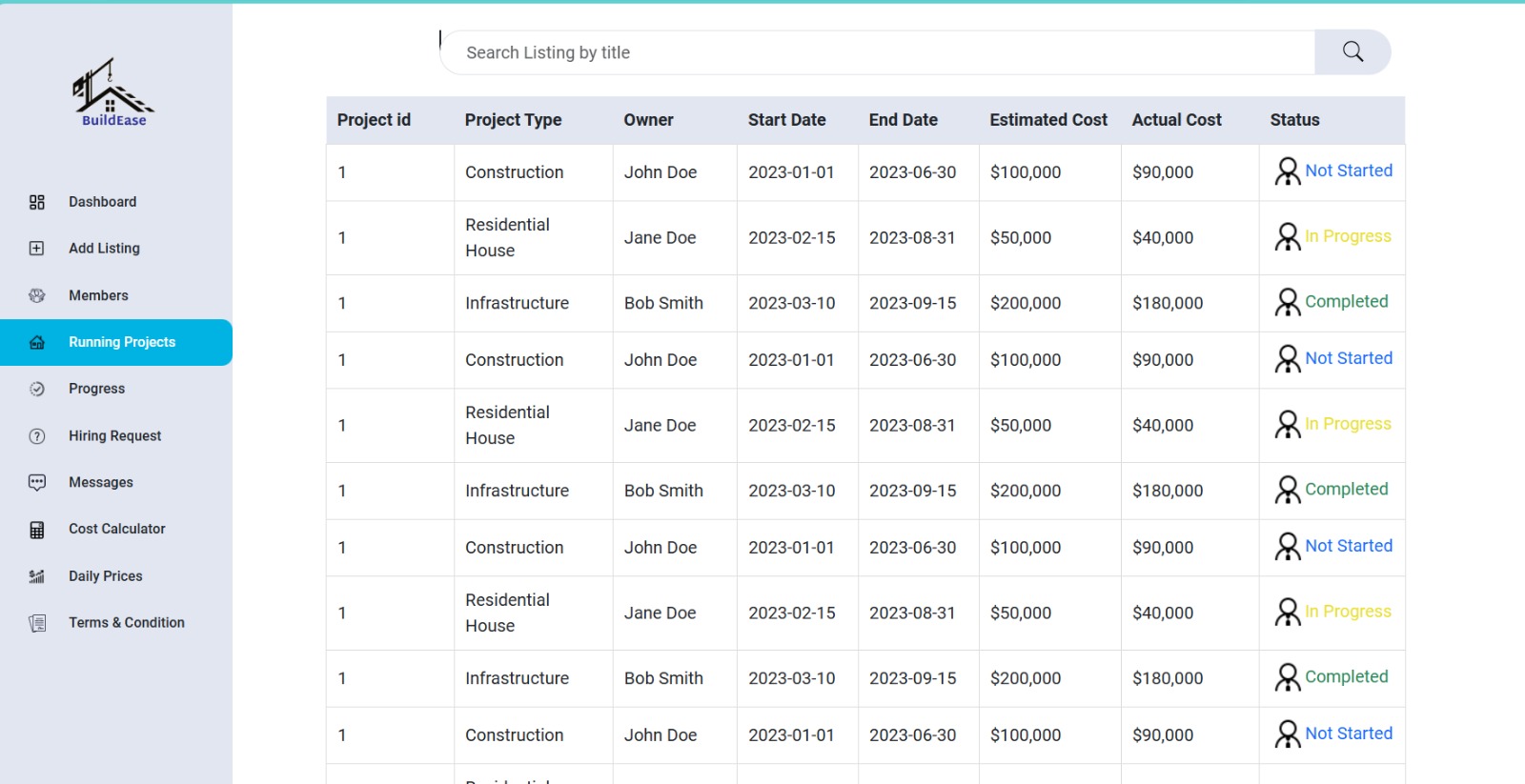
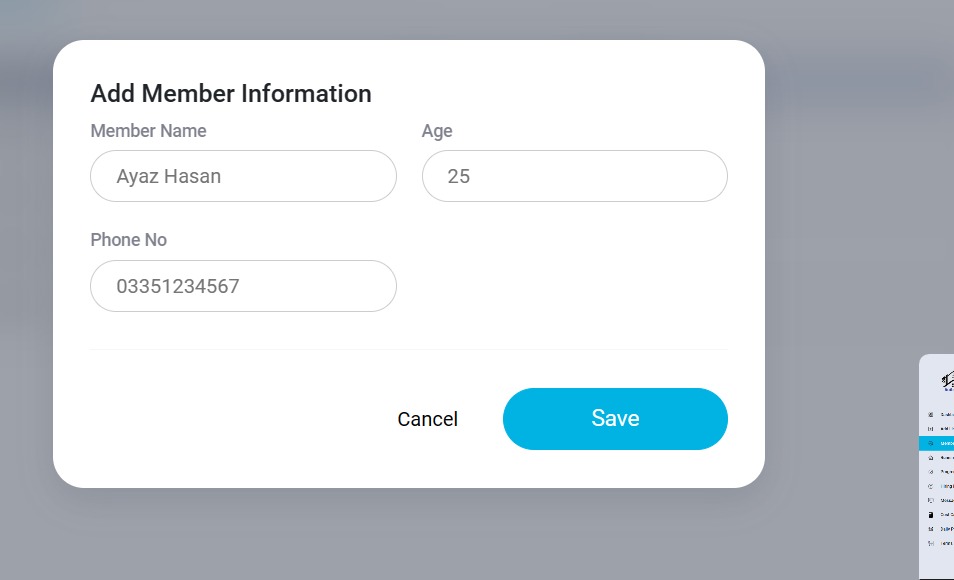
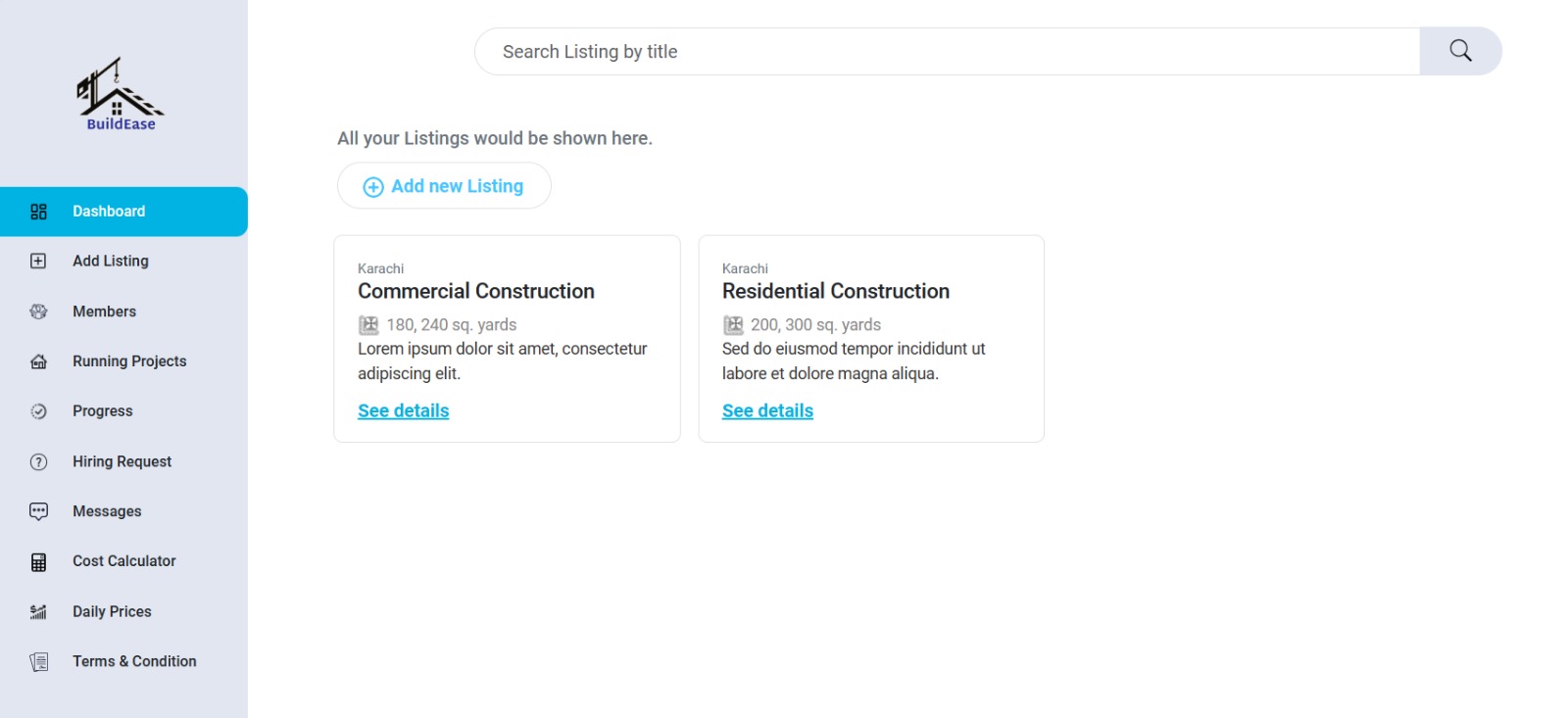
**

# Design Strategy

Our system is currently designed to solve the basic problems of unavailability of data and to bring the construction service providers and the clients together on a single platform we wanted to make sure the design is interactive simple and easy to use for the clients and the service providers. We wanted to make sure that there is a database connectivity. Basic UI and UX design have been applied to make an appealing and consistent user interface that will eye catching and easy to use, feedbacks and checks for input validations. A relational database will be used for showing complex relationships between the data.

# Detailed System Design

## GUI Design



## Database Design

### ER Diagram



### Data Dictionary

#### Client Dictionary

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Client Dictionary** | | | | | | | |
| **Name** | | Client | | | | | |
| **Alias** | | None | | | | | |
| **Where-used/how-used** | | This is used to store the details of the registered client and use it at the time of logging | | | | | |
| **Content description** | | The Credentials of all the client 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 client* | | *String* | *-* | *No* | *“”* | *Unique* |
| *Email* | *Email of the client account* | | *String* | *-* | *No* | *“”* | *Unique* |
| *Password* | *Password of the client account* | | *String* | *-* | *No* | *“”* |  |
| *isAdmin* | *This field defines whether the client 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* |  |

#### Contractor Dictionary

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Contractors Dictionary** | | | | | | | |
| **Name** | | Contractors | | | | | |
| **Alias** | |  | | | | | |
| **Where-used/how-used** | | This is used to store the details of the contractors | | | | | |
| **Content description** | | All the information about contractors 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* |
| *username* | *Username of the contractors* | | *String* | *-* | *No* | *“”* | *Unique* |
| *Email* | *Email of the contractors account* | | *String* | *-* | *No* | *“”* | *Unique* |
| *Password* | *Password of the contractors account* | | *String* | *-* | *No* | *“”* |  |
| *dateCreated* | *This field is autogenerated at the time of creation of object.* | | *Date* | *-* | *No* | *Creation Date and time* |  |

#### Project Dictionary

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Dictionary** | | | | | | | |
| **Name** | | Project | | | | | |
| **Alias** | |  | | | | | |
| **Where-used/how-used** | | This is used to store the information about the construction projects of clients | | | | | |
| **Content description** | | All the information of each project 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 project* | | *String* | *-* | *No* | *“”* | *Unique* |
| Picture | *Picture of the assigned contractor* | | *File* | *-* | *Yes* | *-* |  |
| *Description* | *Description About the Project* | | *String* | *-* | *No* | *“”* |  |

#### Contact Dictionary

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Contact Dictionary** | | | | | | | |
| **Name** | | Contact | | | | | |
| **Alias** | |  | | | | | |
| **Where-used/how-used** | | This is used to store the details of the contacts between contractors and client | | | | | |
| **Content description** | | All the information about each contacts 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* |
| *message* | *messages of the client and contractors* | | *String* | *-* | *No* | *“”* |  |
| *picture* | *Picture attached in messages* | | *File* | *-* | *Yes* | *“”* |  |
| *events* | *Events of deletion and edition* | | *Array* | *-* | *No* | *“”* |  |
| *description* | *Description of the given client or contractors messages* | | *String* | *-* | *No* | *“”* |  |

#### Progress Dictionary

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Progress Dictionary** | | | | | | | |
| **Name** | | Projects | | | | | |
| **Alias** | |  | | | | | |
| **Where-used/how-used** | | This is used to store the progress of ongoing projects | | | | | |
| **Content description** | | All the information about Progress 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* |
| *checkpoints* | *Checkpoint of progress* | | *String* | *-* | *No* | *“”* |  |
| *picture* | *picture of the given Project progress* | | *File* | *-* | *Yes* | *“”* |  |
| *videos* | *videos of the given Project progress* | | *File* | *-* | *No* | *“”* |  |
| *progress* | *Percentage of completion* | | *string* | *-* | *No* | *“”* |  |

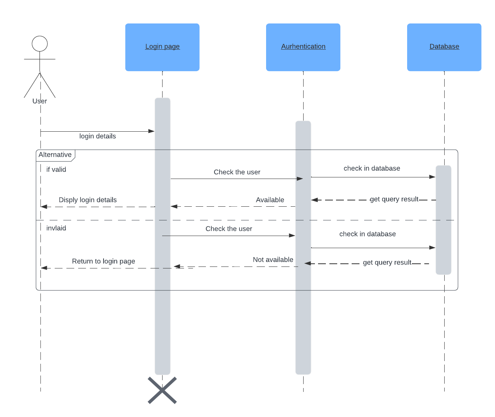
#### Running Project Dictionary

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Running Project Dictionary** | | | | | | | |
| **Name** | | Project data | | | | | |
| **Alias** | |  | | | | | |
| **Where-used/how-used** | | This is used to store the details of the project which being completed ,in progress or not started | | | | | |
| **Content description** | | All the information about each project 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 project* | | *String* | *-* | *No* | *“”* |  |
| *type* | *type of the given project* | | *String* | *-* | *No* | *“”* |  |
| *Owner name* | *owner of the given project* | | *String* | *-* | *No* | *“”* |  |
| *Start Date* | *Start date of the given project* | | *Date* | *-* | *No* | *“”* |  |
| *End Date* | *End Date of the given project* | | *Date* | *-* | *No* | *“”* |  |
| *Estimation* | *Estimate cost of the given project* | | *Number Array* | *-* | *No* | *0* |  |
| *Actual Cost* | *Estimate cost of the given project* | | *Number Array* | *-* | *No* | *0* |  |
| *Status* | *Status of Project(Not start,in progress, completed* | | *String* | *-* | *No* | *“”* |  |

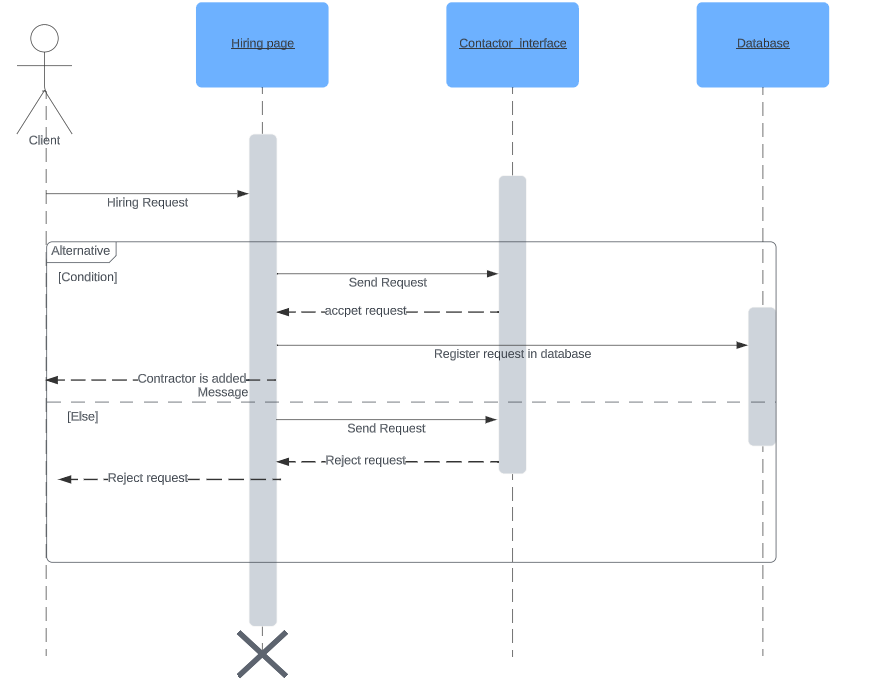
## Application Design

### Sequence Diagram

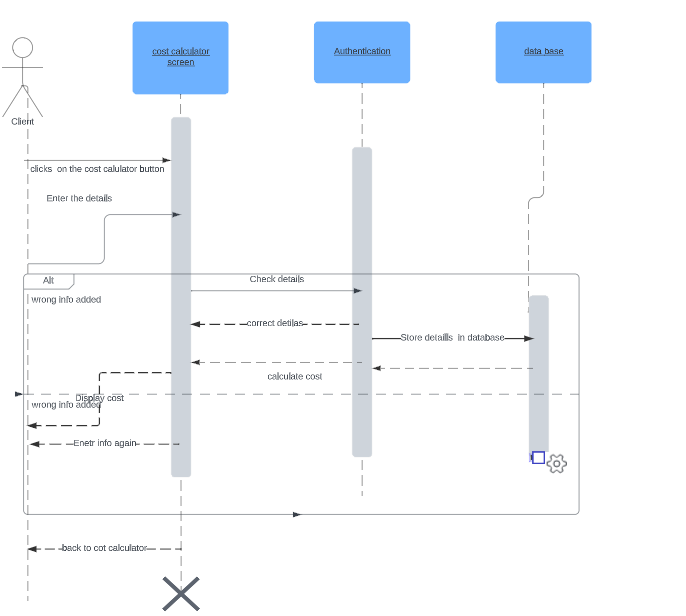
#### Login



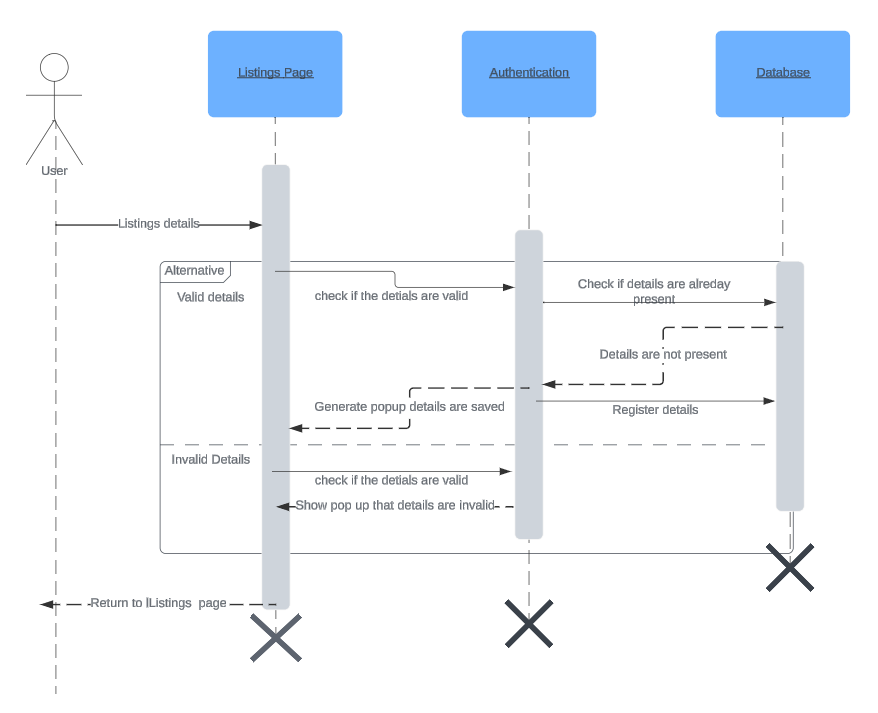
#### Hire contractor



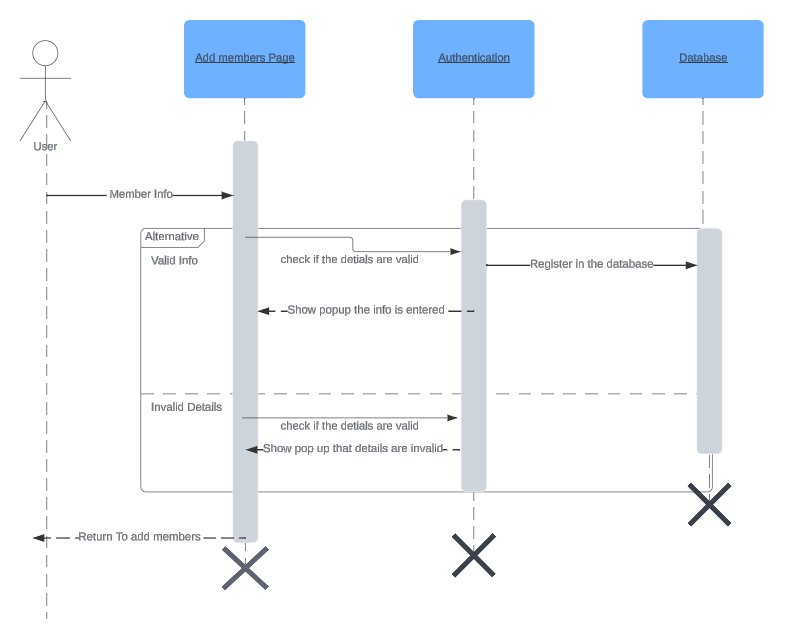
#### Calculate cost

**

#### Add Listings



#### Add Members

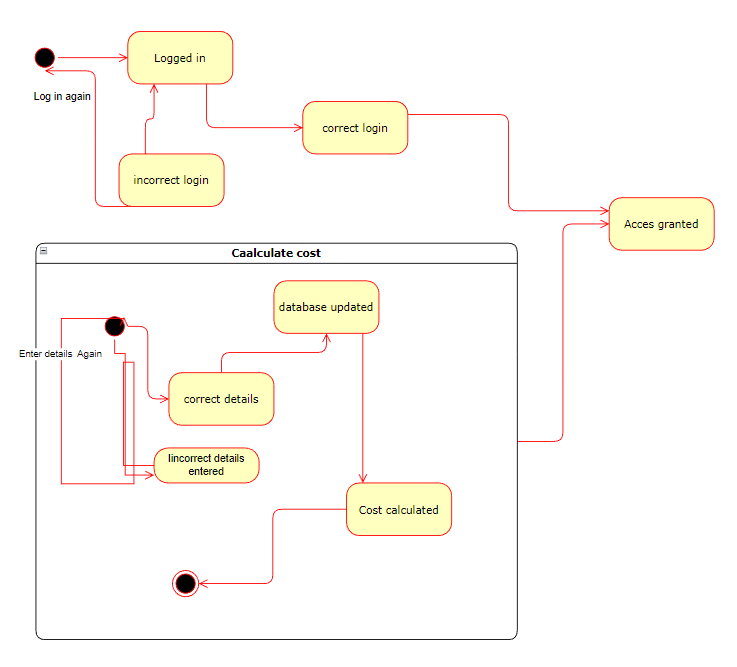


#### Request Page



### State Diagram

* + - 1. **Add Listing**

****

# 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*