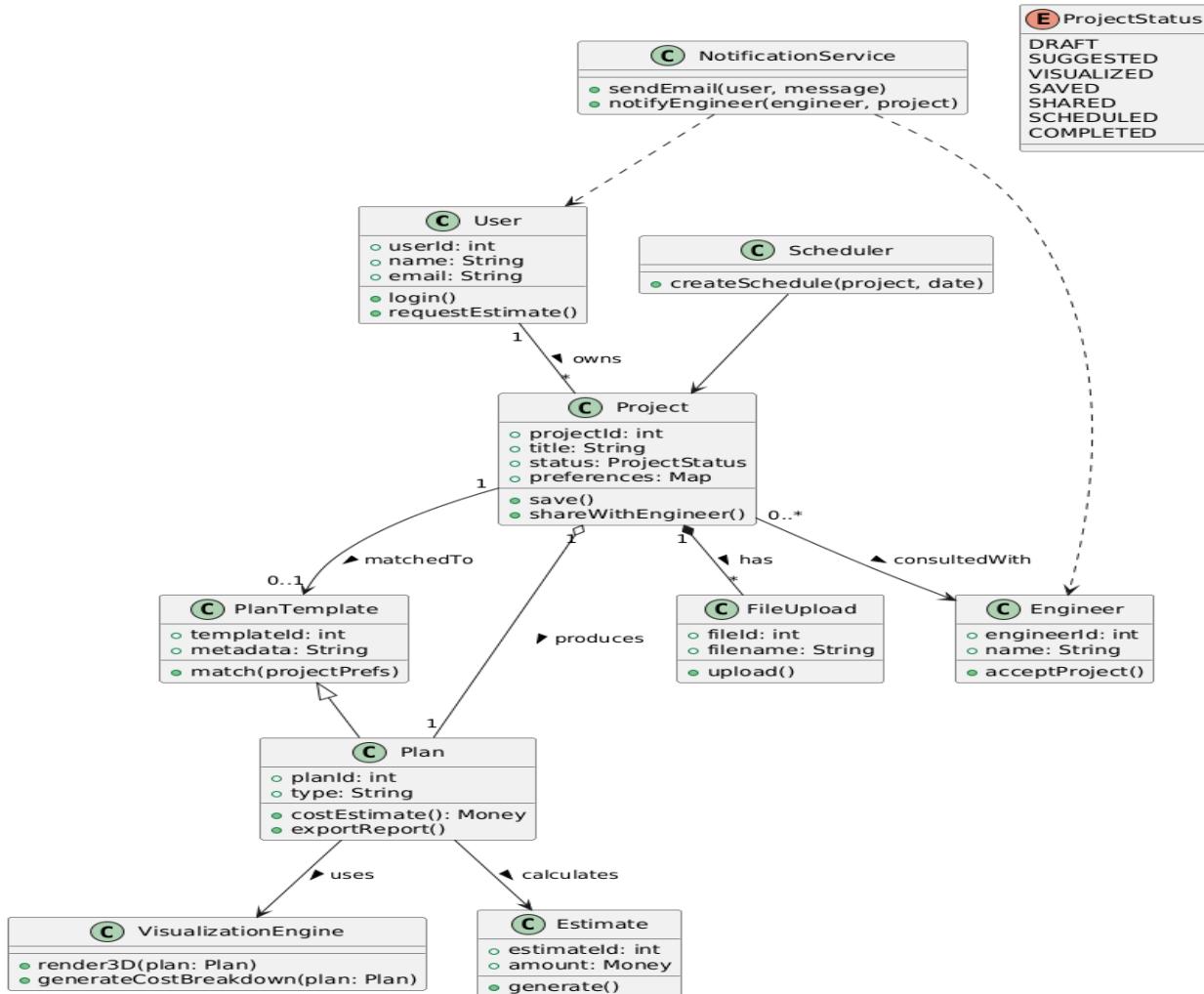


UML DIAGRAMS

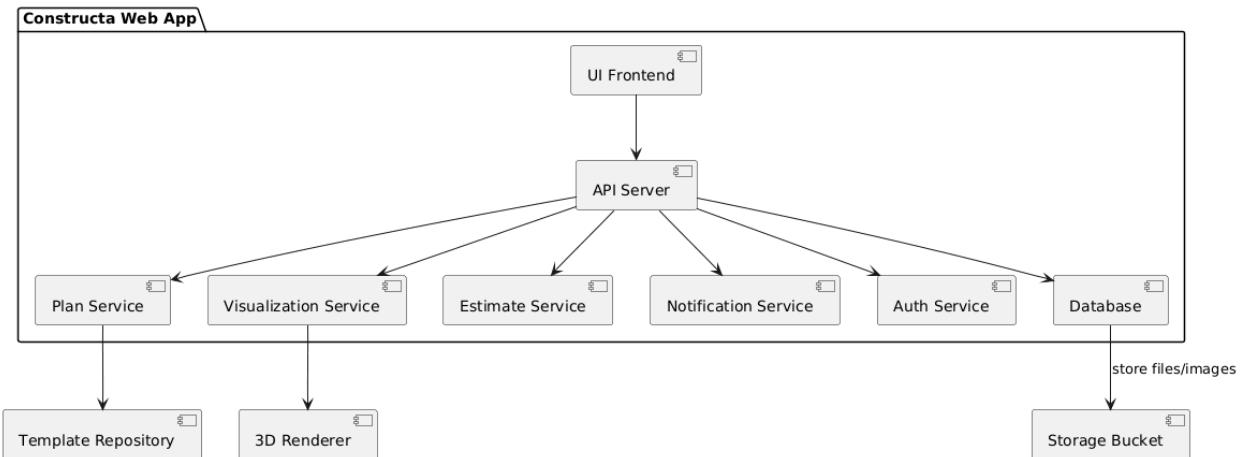
CLASS DIAGRAM

The Class Diagram represents the static structure of the Constructa system. It defines the main classes such as User, HouseRequirement, FloorPlan, Engineer, CostEstimation, Material, Payment, and Feedback along with their attributes and relationships. This diagram helps in understanding the system architecture and serves as the foundation for database and table design.



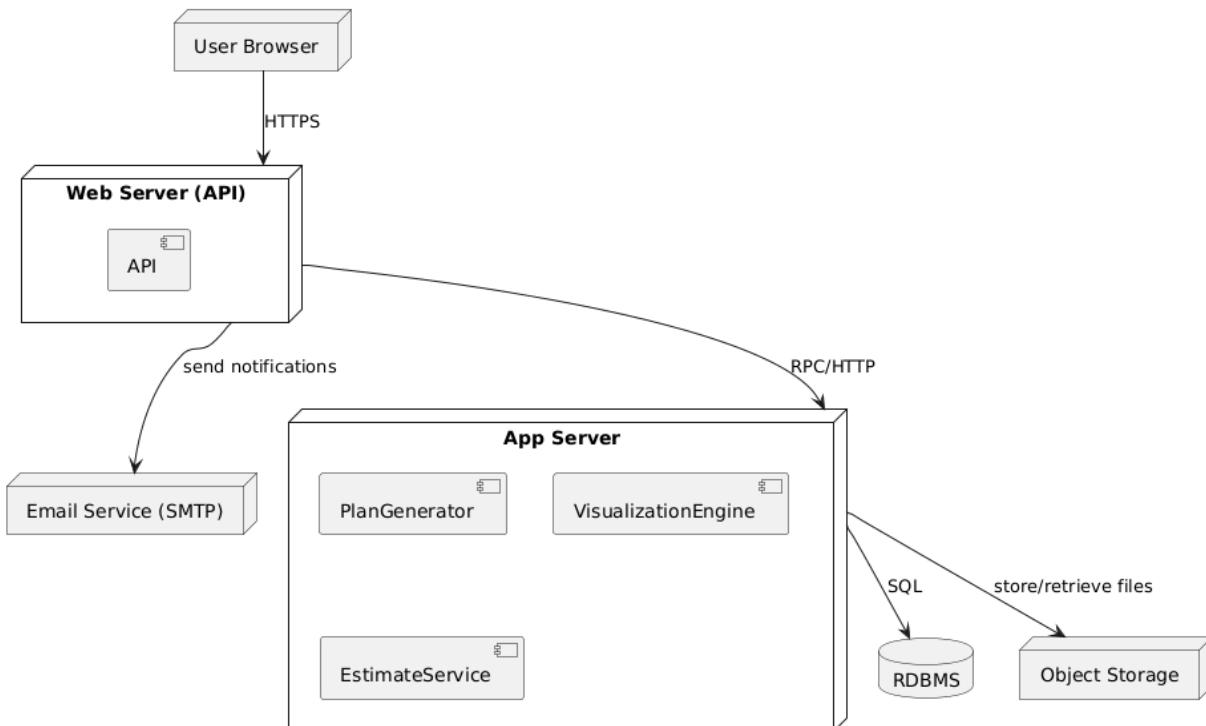
COMPONENT DIAGRAM

The Component Diagram provides a high-level view of system components such as Web Interface, Application Server, Authentication Module, Plan Recommendation Engine, Cost Estimation Engine, and Database.



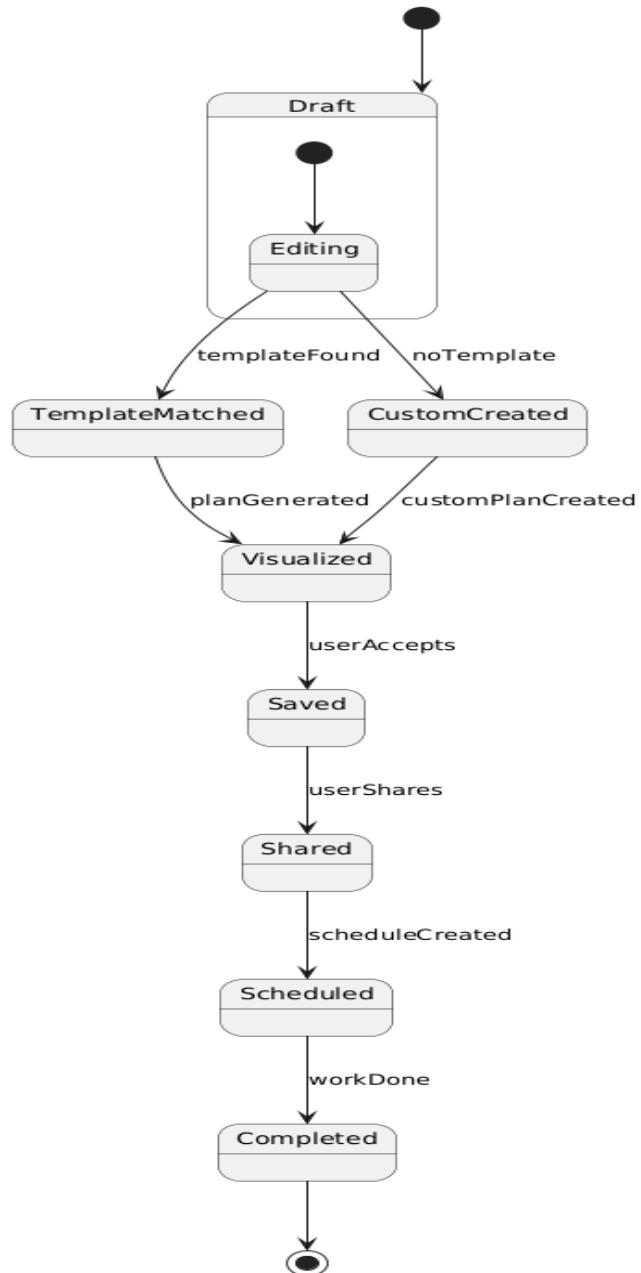
DEPLOYMENT DIAGRAM

The Deployment Diagram illustrates the physical deployment of the Constructa system. It shows the user accessing the system via a web browser, requests handled by the web server and application server, database storage on a separate server, and optional cloud-based services for scalability.



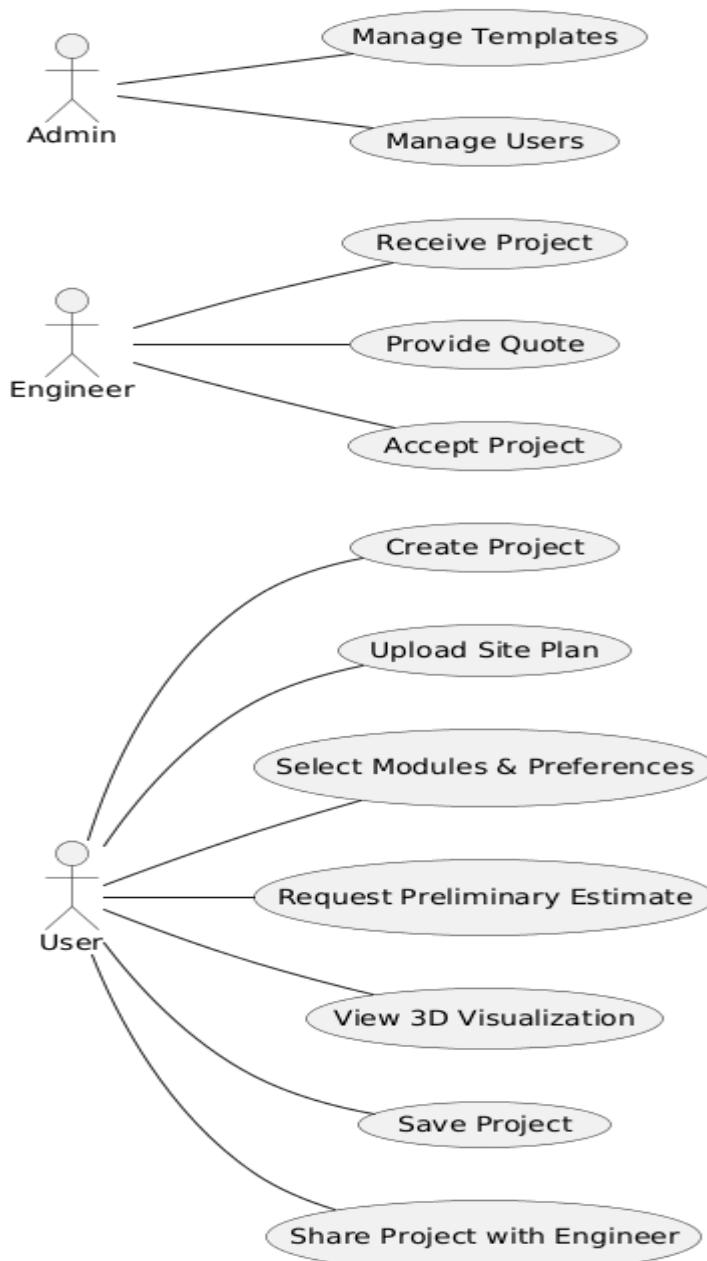
STATE CHART DIAGRAM

The State Chart Diagram describes the lifecycle of a house construction request. States include Submitted, Under Review, Plan Generated, Cost Estimated, Approved, and Completed. Transitions occur based on user actions and system validations.



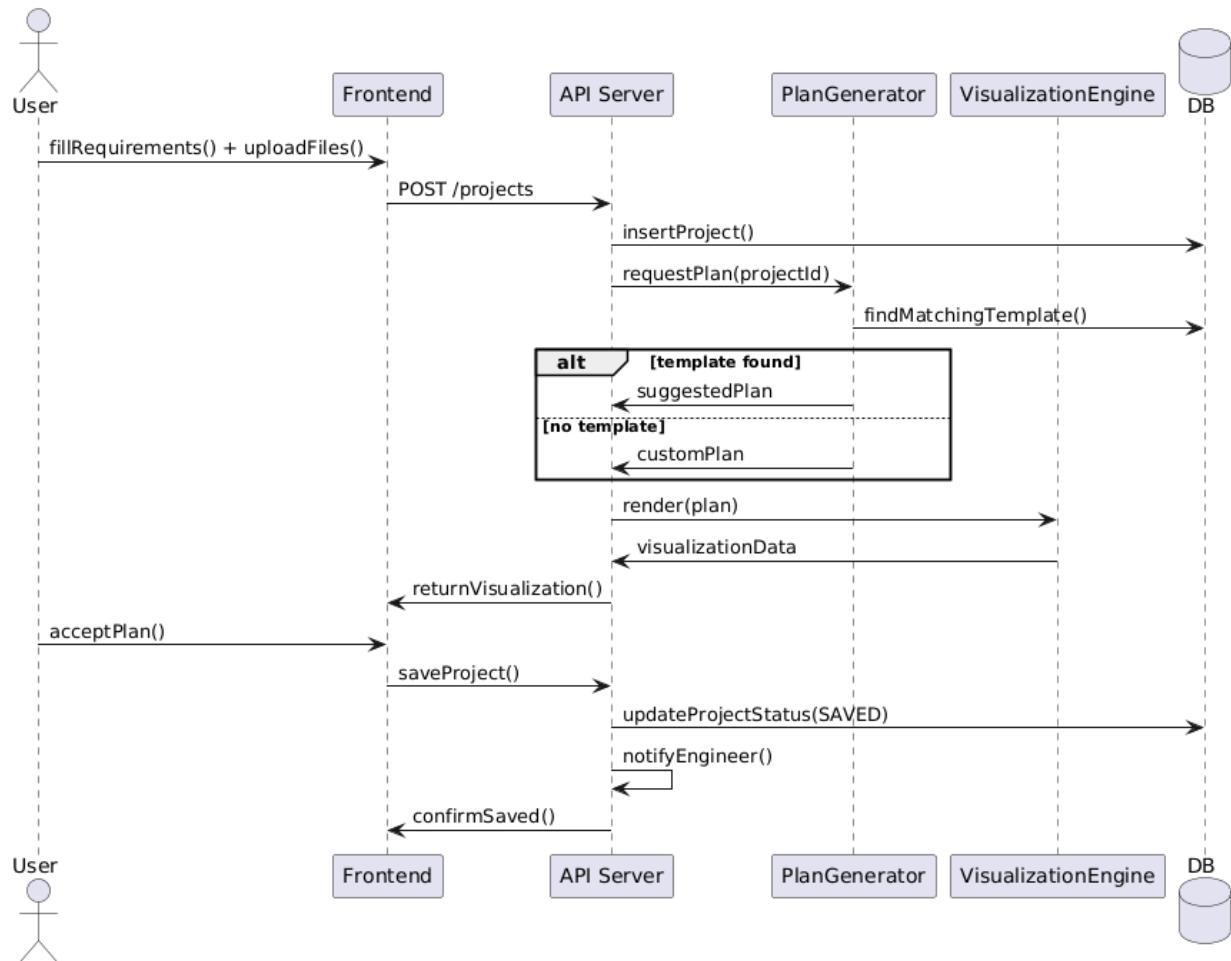
USE CASE DIAGRAM

The Use Case Diagram depicts the interaction between actors and the Constructa system. The primary actors include Admin, User (Homeowner), and Engineer. The diagram highlights key functionalities such as user registration, entering house requirements, viewing recommended plans, generating cost estimates, communicating with engineers, and managing system data.



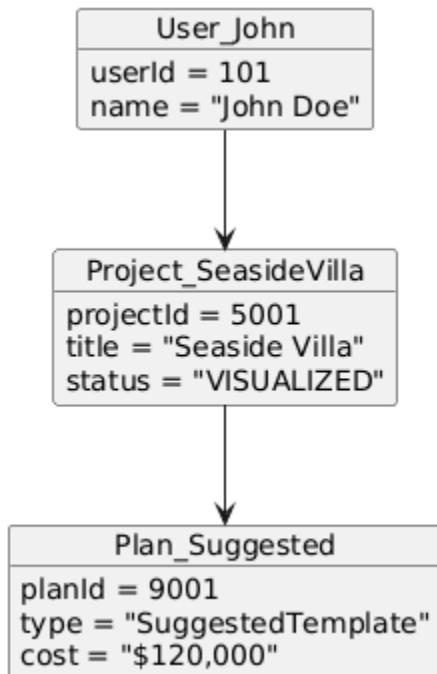
SEQUENCE DIAGRAM

The Sequence Diagram illustrates the flow of interactions when a user generates a house plan and cost estimation. The user submits requirements, the system validates the input, processes plan recommendations, calculates estimated costs, and finally displays the results to the user, ensuring a clear time-based interaction flow.



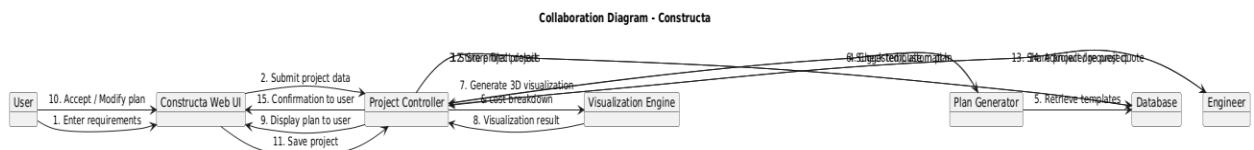
OBJECT DIAGRAM

The Object Diagram provides a snapshot of the system at a specific point in time. It shows instances such as a registered user, a selected house plan, an assigned engineer, and a generated cost estimation, illustrating how objects interact during real-time usage of the Constructa application.



COLLABORATION DIAGRAM

The Collaboration Diagram focuses on how system components work together to fulfill a request. It shows interactions among User Interface, Requirement Module, Plan Recommendation Engine, Cost Estimation Module, and Notification Service while generating and delivering a personalized house construction plan.



ACTIVITY DIAGRAM

The Activity Diagram represents the workflow of the Constructa system. It begins with user login, followed by entering house requirements, validating inputs, generating plan suggestions, estimating costs, and displaying results. Decision points handle invalid data and allow users to re-enter details if needed.

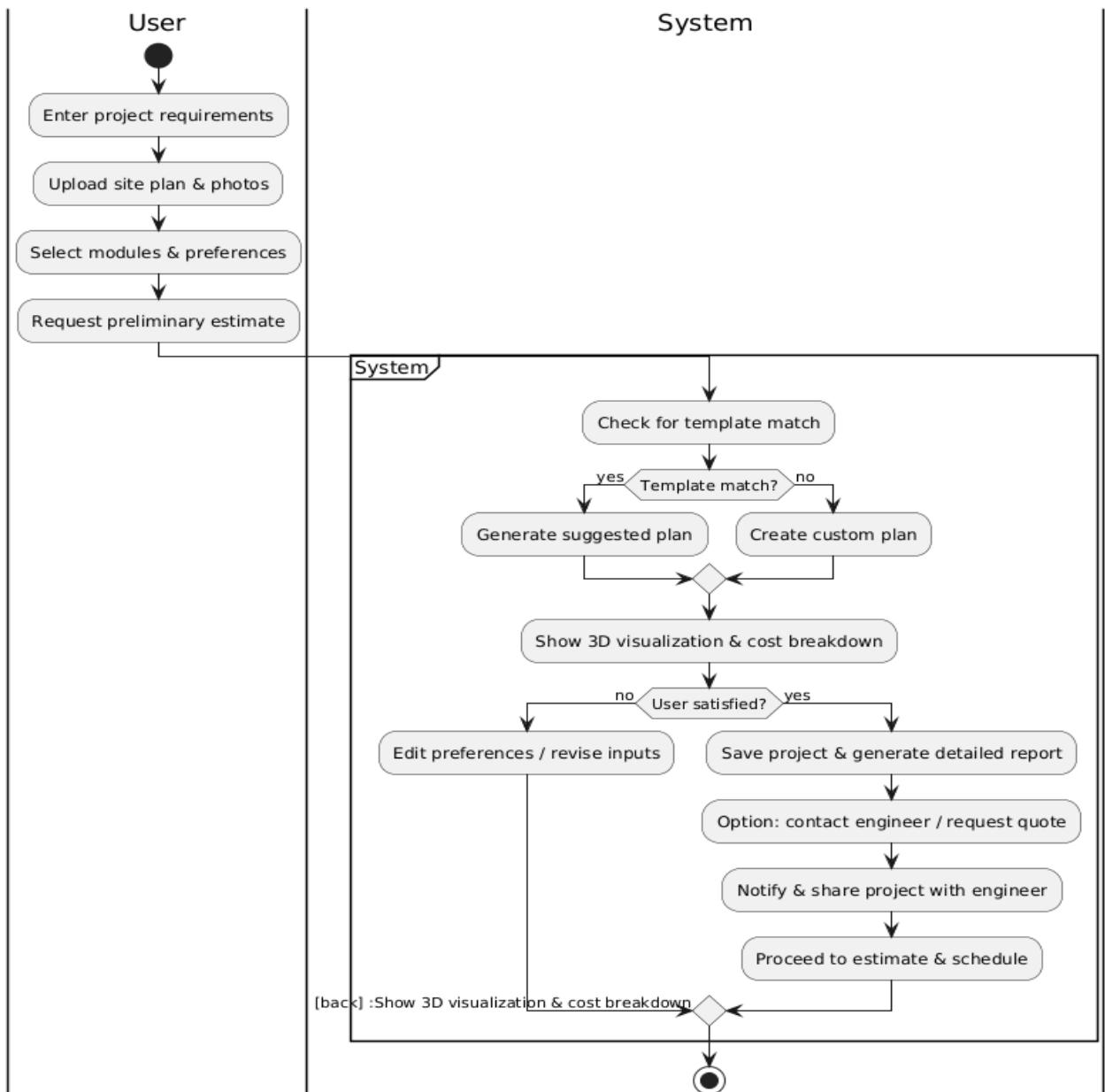


Table Design – Constructa

Users Table

COLUMN	Type	Description
user	Int pk	Unique user identifier
Name	Varchar(100)	User full name
email	Varchar(100)	User email address
password	VARCHAR(255)	Encrypted password
phone	VARCHAR(255)	Contact number
role	VARCHAR(255)	User role (Admin/User/Engineer)

House Requirements Table

Column	Type	Description
requirement_id	INT PK	Unique requirement ID
user_id	INT FK	References users(user_id)
plot_size	VARCHAR(50)	Plot dimensions
floors	INT	Number of floors
rooms	INT	Number of rooms
budget	DECIMAL	User budget

Floor_Plans Table

Column	Type	Description
plan_id	INT PK	Unique plan ID
requirement_id	INT FK	References house_requirements
plan_type	VARCHAR(50)	Type of house plan
area	DECIMAL	Total built-up area
description	TEXT	Plan details

Engineers Table

Column	Type	Description
engineer_id	INT PK	Unique engineer ID
name	VARCHAR(100)	Engineer name
specialization	VARCHAR(100)	Area of expertise
experience	INT	Years of experience
contact	VARCHAR(15)	Contact number

Cost_Estimation Table

Column	Type	Description
estimation_id	INT PK	Unique estimation ID
requirement_id	INT FK	References house_requirements
material_cost	DECIMAL	Material cost
labor_cost	DECIMAL	Labor cost
total_cost	DECIMAL	Total estimated cost

Payments Table

Column	Type	Description
payment_id	INT PK	Unique payment ID
user_id	INT FK	References users(user_id)
amount	DECIMAL	Payment amount
status	VARCHAR(20)	Payment status
payment_date	DATE	Date of payment

TABLE NORMALISATION

First Normal Form (1NF)

A table is said to be in **First Normal Form (1NF)** if:

- Each field contains **atomic (single) values**
- There are **no repeating groups or multivalued attributes**

- Each record can be uniquely identified by a **primary key**

Application in Constructa

- The **Users** table stores single values for name, email, phone, and role.
- The **House_Requirements** table stores one plot size, number of floors, rooms, and budget per record.
- The **Payments** table stores one payment amount and status per transaction.

All Constructa tables satisfy **1NF** because there are no multivalued or repeating attributes.

Second Normal Form (2NF)

A table is in **Second Normal Form (2NF)** if:

- It is already in **1NF**
- All non-key attributes are **fully functionally dependent** on the primary key
- There is **no partial dependency**

Application in Constructa

- Each table uses a **single-column primary key** such as `user_id`, `requirement_id`, `plan_id`, etc.
- In the **House_Requirements** table, attributes like plot size, floors, rooms, and budget depend only on `requirement_id`.
- In the **Cost_Estimation** table, material cost, labor cost, and total cost depend entirely on `estimation_id`.

Since there are no composite primary keys, **partial dependency does not exist**, and all tables are in **2NF**.

Third Normal Form (3NF)

A table is in **Third Normal Form (3NF)** if:

- It is already in **2NF**
- There is **no transitive dependency**
- Non-key attributes do not depend on other non-key attributes

Application in Constructa

- User contact details are stored only in the **Users** table and not repeated in other tables.
- Engineer details are stored separately in the **Engineers** table.
- Cost-related data is stored only in the **Cost_Estimation** table and linked using foreign keys.
- Payment details are stored independently in the **Payments** table.

