



BSCS-S24-015

03-134211-009 Hafiz Abdul Haseeb

03-134211-020 Muhammad Abdullah Shahzad

**CITYCRAFT: YOUR ONE-STOP PLATFORM FOR
CONNECTING CONSTRUCTION COMPANIES WITH
CLIENTS!**

In partial fulfilment of the requirements for the degree of
Bachelor of Science in Computer Science

Supervisor: Shahid Mehmood

Department of Computer Sciences
Bahria University, Lahore Campus

January 2025

Certificate



We accept the work contained in the report titled

“CityCraft”

written by

Hafiz Abdul Haseeb

Muhammad Abdullah Shahzad

as a confirmation to the required standard for the partial fulfilment of the degree of
Bachelor of Science in Computer Science.

Approved by:

Supervisor: **Shahid Mehmood**

(Signature)

January 05, 2025

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Enrolment	Name	Signature
03-134211-009	Hafiz Abdul Haseeb	
03-134211-020	Muhammad Abdullah Shahzad	

Date : January 05, 2025

Specially dedicated to
my beloved grandmother, mother and father
(Hafiz Abdul Haseeb)
my beloved grandmother, mother and father
(Muhammad Abdullah Shahzad)

ACKNOWLEDGEMENTS

We would like to thank everyone who had contributed to the successful completion of this project. I/We would like to express my/our gratitude to my research supervisor, Mr Shahid Mehmood for his invaluable advice, guidance and his/her enormous patience throughout the development of the research.

In addition, We would also like to express my gratitude to my/our loving parent and friends who had helped and given me encouragement.

Hafiz Abdul Haseeb
Muhammad Abdullah SHahzad

CityCraft: Your one-stop platform for connecting construction companies with clients!

ABSTRACT

The construction industry faces persistent challenges in connecting construction companies with genuine clients and vice versa. Traditional methods of project procurement often lack transparency and efficiency, resulting in frustration for builders seeking projects and clients searching for trustworthy construction services. The absence of a centralized platform further complicates the process, leaving both parties with limited options. To address these issues, our project introduces a web-based platform that facilitates seamless collaboration between clients and construction companies. This platform empowers clients to upload project details, enabling builders to bid with comprehensive proposals, including cost estimates and service breakdowns. Builders can also create detailed profiles showcasing their past projects, establishing credibility and trust.

By incorporating user-friendly interfaces, direct communication channels, project tracking, and secure payment integration, the platform fosters a transparent and competitive environment. Clients benefit from streamlined access to the best construction deals, while builders gain a direct avenue to demonstrate their expertise and secure projects. This innovative solution revolutionizes the construction industry by providing a centralized hub for project procurement, promoting transparency, efficiency, and effective communication throughout the process.

TABLE OF CONTENTS

DECLARATION	ii
ACKNOWLEDGEMENTS	iv
ABSTRACT	v
TABLE OF CONTENTS	vi
LIST OF TABLES	ix
LIST OF FIGURES	xi
LIST OF APPENDICES	xiv

CHAPTERS

1	INTRODUCTION	1
	1.1 Background	1
	1.2 Problem Statements	1
	1.3 Aims and Objectives	2
	1.4 Scope of Project	2
2	LITERATURE REVIEW (and/ SRS)	4
	2.1 Existing Platforms Analysis	4
	2.2 Construction Industry Challenges	5
	2.3 Use case and identifier	6
	2.3.1 Signup(CC-01-001)	6
	2.3.2 Login (CC-01-002)	7
	2.3.3 User()	9
	2.3.4 Display Interpreted CityCraft(CC-01-004)	9
	2.4 Technologies in Construction Management	11
	2.4.1 Role of Web-based Platforms	11
	2.4.2 Bidding Systems Benefits	11
	2.5 Software Requirements Specification (SRS)	13

	2.5.1	Major Functional Requirements:	13
	2.5.2	Dashboard()	15
	2.5.3	Create Project()	16
3		DESIGN AND METHODOLOGY	19
	3.1	Platform Architecture	19
	3.1.1	Client Dashboard:	19
	3.1.2	Builder Dashboard:	20
	3.1.3	Client Dashboard()	21
	3.1.4	Builder Dashboard()	22
	3.1.5	Builder profile()	23
	3.2	Development Methodology	25
	3.2.1	Technology Stack	25
	3.2.2	Development Phases	26
	3.2.3	Workflow of the Platform	27
	3.2.4	Bidding system()	28
	3.2.5	Accept/Reject bid()	29
	3.3	Core Functional Modules	30
	3.3.1	View Bid Status()	31
	3.3.2	Manage Projects()	32
4		DATA AND EXPERIMENTS (and/or IMPLMENTATION)	34
	4.1	Database Design	34
	4.1.1	Materialratelist()	35
		For Houses:	36
		For High-Rise Buildings:	37
	4.1.2	ConstructionRoadmap	38
	4.2	Technology Stack	40
	4.3	Functional Testing	41
	4.3.1	User Experience Testing	41
5		RESULTS AND DISCUSSIONS	42
	5.1	Platform Performance	42

5.2	General User Guide	42
5.2.1	Accessing the Platform	42
5.2.2	Table for Accessing the Platform	43
6	CONCLUSION AND RECOMMENDATIONS	45
6.1	Summary of Achievements	45
6.2	Future Work	45
6.3	Lessons Learned	46
6.3.1	Challenges and Solutions	46
	REFERENCES	73
	APPENDICES	74

LIST OF TABLES

TABLE	TITLE	PAGE
Table 2.1:	Signup	6
Table 2.2:	Login	7
Table 2.3:	User	9
Table 2.3:	Display Interpreted CityCraft	9
Table 2.5:	Dashboard Page	15
Table 2.6:	Create Project Page	16
Table 3.1:	Client Dashboard	21
Table 3.2:	Builder Dashboard	22
Table 3.3:	Builder Profile	23
Table 3.4:	Bidding System	28
Table 3.5:	Accept/Reject Bids	29
Table 3.:	Bid status	31
Table 3.7:	Manage Projects	32
Table 4.1:	Material Ratelist	35

Table 4.2: Construction Roadmap	38
Table 5.1: Clients/Builders Functionality	43
Table 5.2: Detailed Representation	44

LIST OF FIGURES

Figure No.	Title	Page No.
1	Use Case Diagram	47
2	Class Diagram	48
3	ERD Diagram	49
4	User Login Sequence Diagram	51
5	Signup Sequence Diagram	51
6	Create a Project (Client) Sequence Diagram	52
7	Bid on a Project (Builder) Sequence Diagram	52
8	Accept/Reject Bid (Client) Sequence Diagram	53
9	View Material Rates Sequence Diagram	53
10	View Roadmap Sequence Diagram	54
11	Showcase Previous Project (Builder) Sequence Diagram	55
12	Delete Previous Projects (Builder) Sequence Diagram	55
13	User Login Activity Diagram	56
14	Signup Activity Diagram	57
15	Create a Project Activity Diagram	58
16	Bid on a Project Activity Diagram	59
17	Accept/Reject Bid Activity Diagram	60
18	View Material Activity Diagram	61
19	View Roadmap Activity Diagram	61
20	Showcase Previous Projects Activity Diagram	62
21	Delete Previous Projects Activity Diagram	63
22	Component Diagram of CityCraft	64
23	Deployment Diagram	66
24	User Account State Diagram	68
25	Project State Diagram	69

Figure No.	Title	Page No.
26	Bid State Diagram	70
27	Builder Project State Diagram	71

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
	APPENDIX A: Project Diagrams	74
	APPENDIX B: Test Cases	76
	APPENDIX C: User Manual	76
	APPENDIX D: Glossary of Terms	7676

CHAPTER 1

INTRODUCTION

1.1 Background

The construction sector is one of the most significant sectors in the economy, but it still faces many challenges. Builders and clients often have a hard time finding each other, and there is usually a lack of good communication between the two parties. Clients usually do not know the correct cost or materials needed for their projects, while builders usually struggle to find the right opportunities. To solve these problems, we designed CityCraft, an online platform that will make connecting clients with builders efficient.

1.2 Problem Statements

Our project addresses such main problems:

No unified platform is available for clients and builders to engage and work together.

Lack of transparency: Bidding and project pricing is obscure.

Poor communication: The clients and builders share poor communication, which delays the whole process.

Absence of material price tracking presents challenges in locating current material prices necessary for precise project cost estimation.

1.3 Aims and Objectives

CityCraft aims to:

Offer a user-friendly platform that enables clients to articulate their construction project specifications.

Allow builders to bid on projects and showcase their past work.

Make the process of bidding clear, so clients understand what they're paying for.

Help clients and builders track project progress.

Provide tools such as material pricing catalogs and construction plans to make the process easier.

Improve the construction industry by offering equity, enhancing communication skills, and promoting transparency.

1.4 Scope of Project

The primary issues of CityCraft involve Pakistan's construction industry issues. Amongst its key features it offers:

User Profiles: Clients can post projects while bidders can also post their work.

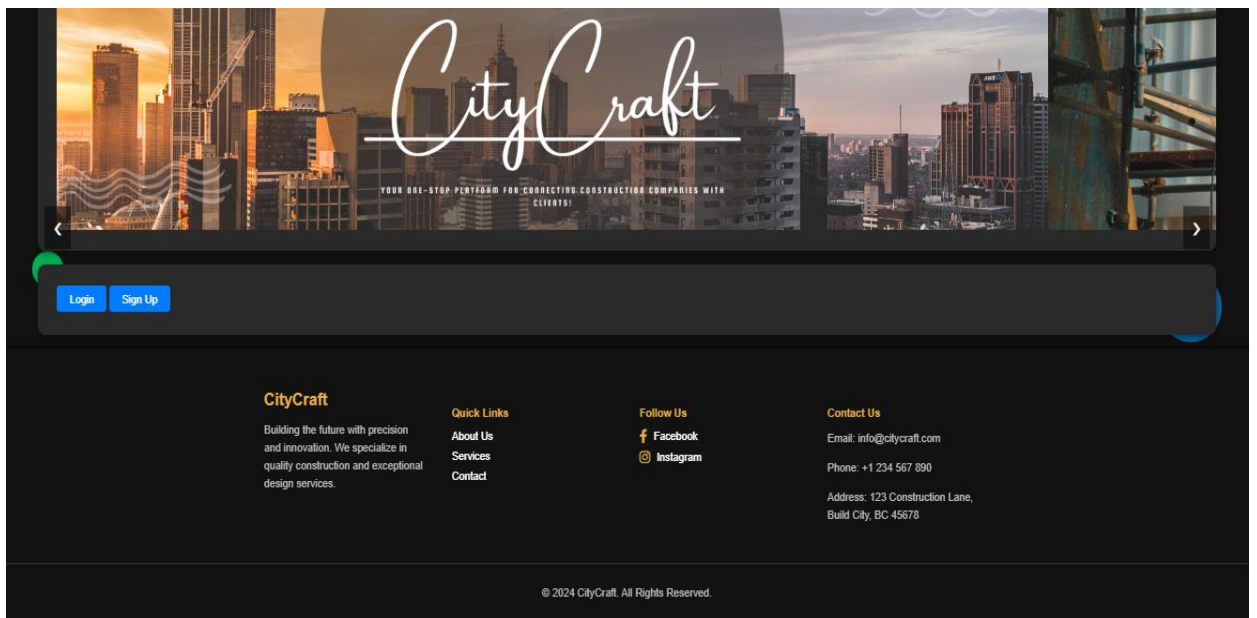
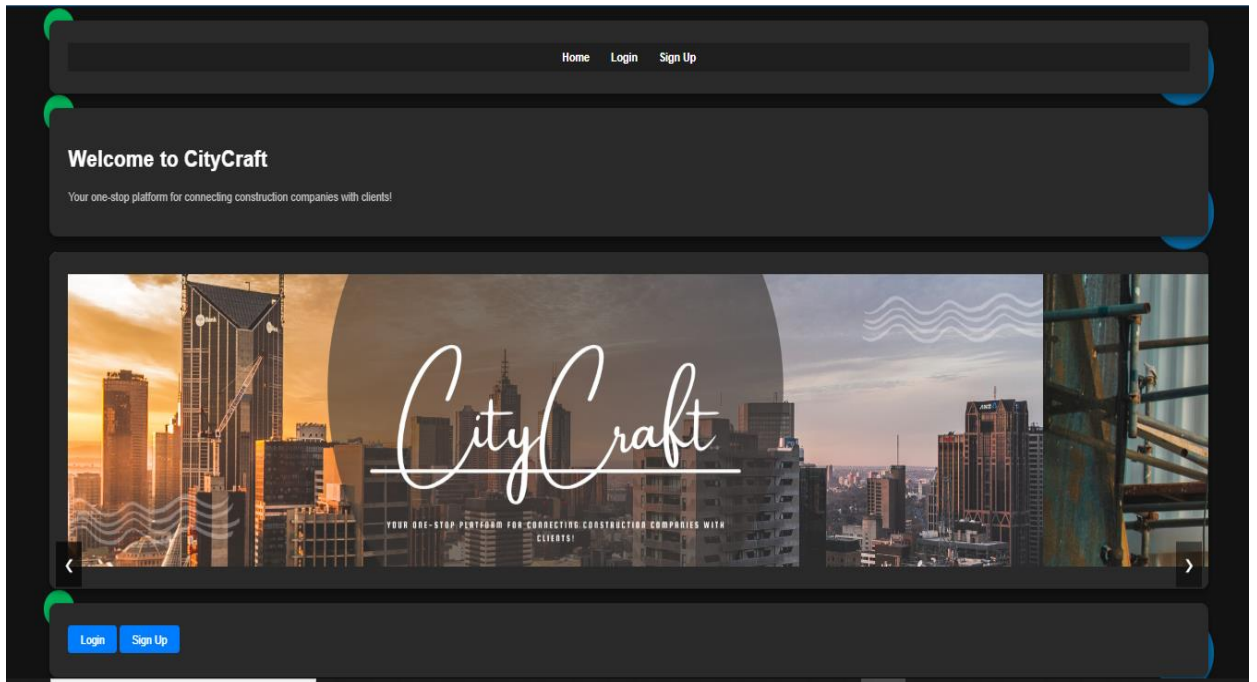
Bidding System: Builders can bid on projects with detailed cost breakdowns.

Material Prices: General information on updated prices for construction materials to clients and builders.

Construction Roadmaps: Detailed sequential instructions for undertaking residential and commercial construction initiatives.

Communication Tools: Real-time messaging and notifications to improve collaborative work.

1.4.1 CityCraft Home Page



CHAPTER 2

LITERATURE REVIEW (and/ SRS)

2.1 Existing Platforms Analysis

Globally and within Pakistan, there are a few platforms trying to connect clients with service providers in the construction industry. Nonetheless, quite a few of these platforms fail to cater to the unique concerns and hurdles of the Pakistani construction space, Here's a look at several platforms:

Houzz:

Houzz is one of the most used global platform which connect homeowners with the construction and interior design industry professionals. Its focus is on sharing/project portfolios and connecting clients with service providers. But Houzz also lacks localized features like current construction material prices or detailed bidding system.

Archinect Pakistan:

Archinect focuses on connecting architects and designers with clients. It is useful for the initial design and planning phases but does not cover execution stages such as material procurement or builder bidding.

Limitations of Existing Platforms in Pakistan:

No material price updates: No platform offers current construction material prices, which are a fundamental factor in cost estimation.

No project management tools: Some of the platforms just connect the clients with builders but offer no features such as progress tracking or a bidding system.

Low transparency: Many platforms don't allow the clients to compare detailed bids or know the cost break-up of a project.

Limited to specific phases: Most platforms focus only on connecting parties or initial project design, leaving the rest of the process unaddressed.

This gap highlights the need for a dedicated platform like CityCraft, which is tailored to address these specific challenges in the Pakistani construction industry.

2.2 Construction Industry Challenges

The construction industry in Pakistan faces numerous challenges that hinder efficiency and productivity. Some of the key issues are:

Delays in Project Completion:

Lack of coordination between clients and builders often results in extended timelines.

Lack of clear expectations and goals often leads to repeated changes during execution.

Cost Overruns:

Inaccurate budgeting and unanticipated price fluctuations in materials cause projects to exceed their budgets.

Clients often fail to get fair and competitive prices.

Poor Communication:

Clients and builders do not have streamlined communication channels, leading to misunderstandings.

Updates about progress are often delayed, causing frustration on both sides.

Lack of Transparency:

Clients find it difficult to verify builders' past work and reliability.

Builders struggle to find clients who trust them, especially if they are new in the industry.

Material Price Instability:

Construction material prices in Pakistan change frequently due to market conditions, making it hard to budget accurately.

These challenges emphasize the importance of developing a comprehensive solution like CityCraft, which combines communication, transparency, and up-to-date information to resolve these issues.

2.3 Use case and identifier

2.3.1 Signup(CC-01-001)

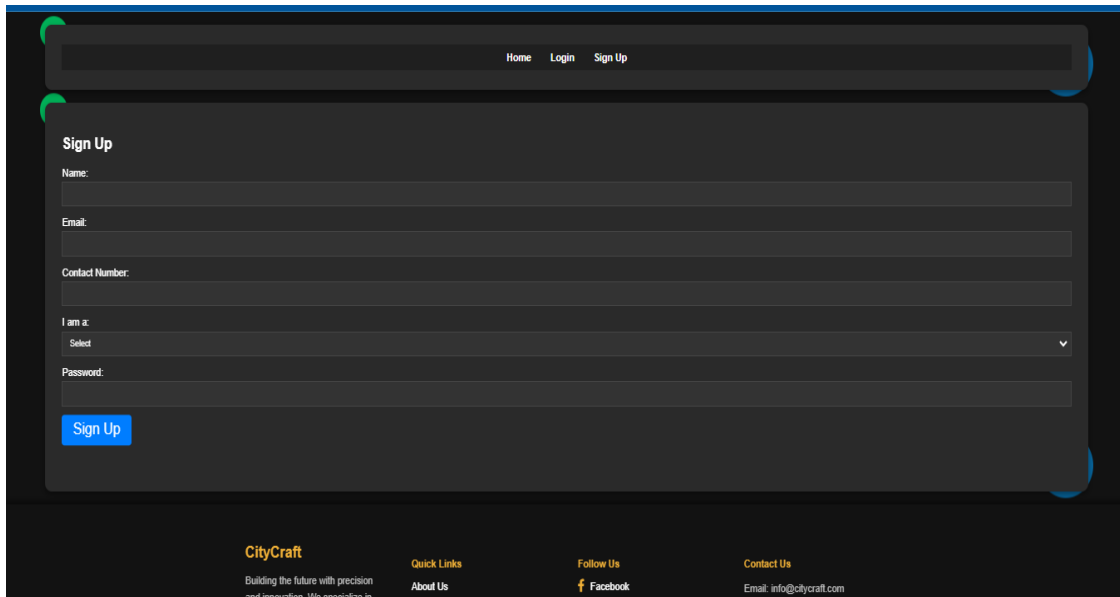
This use case is about the user. He wants to create an account to the web application. The Users are individuals with difficulty speaking.

Table 2.1: Signup

Name	Signup Page
Unique Identifier	CC-01-009
Objective	Register a new user (client or builder) by collecting their name, email, contact number, user type, and password.
Priority	High
Actors	Users (Client or Builder)
Basic Flow	<ol style="list-style-type: none"> 1. User navigates to the signup page. 2. User enters their name, email, contact number, user type, and password. 3. User submits the signup form. 4. The system processes the data and calls the <code>signup</code> method from <code>AuthController</code>. 5. If the signup is successful, the user is redirected to the login page.
Alternative Flow	<p>Alt-1. Missing required fields (e.g., contact number or user type): Display an error message and prompt user to fill all required fields.</p> <p>Alt-2. Signup failure (e.g., email already exists): Display an error message.</p>
Preconditions	The user must provide valid information (name, email, contact number, user type, password).

Name	Signup Page
Postconditions	A new user account is created. The user is redirected to the login page for authentication.
Notes/Issues	Ensure proper validation for the contact number and email format. Sanitize user input to prevent security vulnerabilities. Handle duplicate email error gracefully.

2.3.1.1 Signup Page



2.3.2 Login (CC-01-002)

User login on the Web application with the username and password

Table 2.2: Login

Name	Login Page
Unique Identifier	CC-01-002
Objective	Authenticate users (clients or builders) using email and password.
Priority	High
Actors	Users (Client or Builder)

Name	Login Page
Basic Flow	<ol style="list-style-type: none"> 1. User navigates to the login page. 2. User enters email and password. 3. User submits the login form. 4. The system processes the login using <code>AuthController</code>. 5. If authentication is successful, the user is redirected to their dashboard (client or builder).
Alternative Flow	<p>Alt-1. User is not registered: Redirect to the signup page.</p> <p>Alt-2. Incorrect email or password: Show an error message prompting the user to retry.</p>
Preconditions	<p>The user must have a registered email and password.</p> <p>The form must be correctly filled out.</p>
Postconditions	<p>User is authenticated and granted access to their dashboard.</p> <p>User session is initiated upon successful login.</p>
Notes/Issues	<p>Ensure form validation on both client and server side.</p> <p>Handle errors such as database connectivity issues.</p> <p>Implement security features like input sanitization and password hashing.</p>

2.3.2.1 Login Page

Home Login Sign Up

Login

Email:

Password:

[Login](#)

[Forgot Password?](#)

CityCraft
Building the future with precision and innovation. We specialize in quality construction and exceptional design services.

Quick Links
[About Us](#)
[Services](#)
[Contact](#)

Follow Us
[Facebook](#)
[Instagram](#)

Contact Us
Email: info@citycraft.com
Phone: +1 234 567 890
Address: 123 Construction Lane,
Build City, BC 45678

2.3.3 User()

Table 2.3: User

Name	User Table
Unique Identifier	CC-01-003
Objective	Manage user information for authentication and authorization.
Priority	High
Actors	Users (Clients and Builders).
Basic Flow	1. Create a new user account (Signup). 2. Enter username/email and password for login. 3. Authenticate and redirect users to their respective dashboards.
Alternative Flow	Alt-1. Invalid input during signup: Show error message. Alt-2. Authentication fails: Show login error.
Preconditions	Valid username/email and password. User must be assigned a user type (Client or Builder).
Postconditions	User information is securely stored in the database. Authentication succeeds; access granted.
Notes/Issues	Ensure strong password hashing (e.g., bcrypt). Prevent duplicate email addresses. Internet connection required.

2.3.4 Display Interpreted CityCraft(CC-01-004)

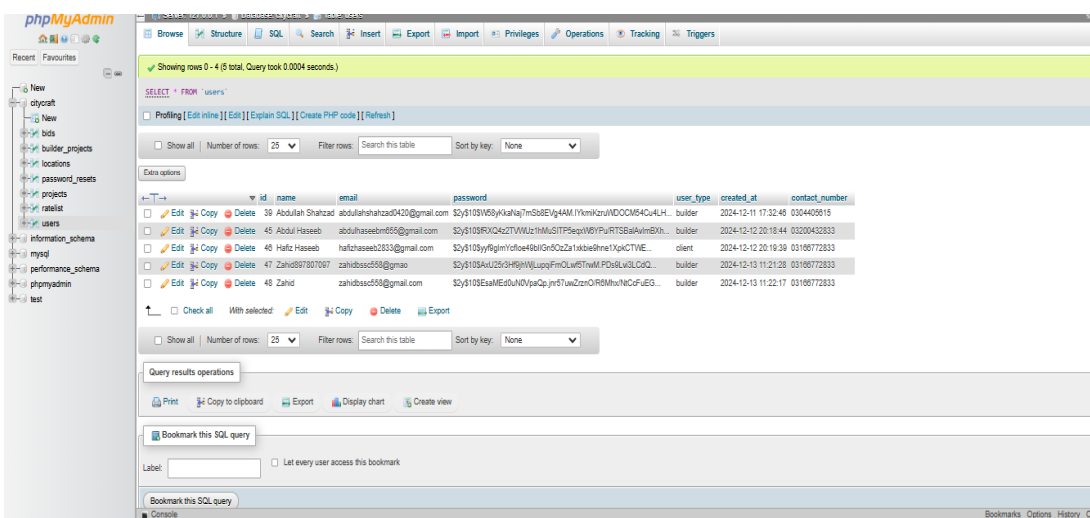
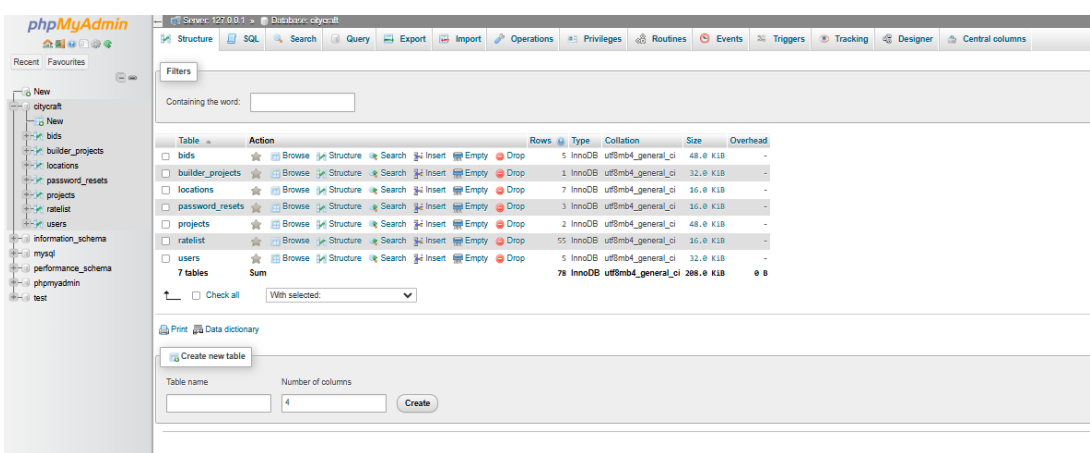
The user can see the interpreted CityCraft on the web application

Table 2.4: Display Interpreted

Name	Display Interpreted CityCraft Table
Unique Identifier	CC-01-004
Objective	Display user data and projects tailored to the user's type.
Priority	High
Actors	Users (Clients and Builders).
Basic Flow	1. Retrieve user details from the database. 2. Fetch associated data (e.g., projects, bids). 3. Display projects on the dashboard. 4. Ensure data is formatted and styled based on user type.
Alternative Flow	Alt-1. User data retrieval fails: Show error message. Alt-2. Dashboard fails to load: Redirect to an error page.

Name	Display Interpreted CityCraft Table
Preconditions	Valid user session exists. Data is stored and accessible in the database.
Postconditions	User data is displayed correctly on the dashboard. Dashboard content dynamically changes based on the user type.
Notes/Issues	Ensure database connectivity. Use efficient queries to fetch data to avoid delays. Optimize front-end layout for clarity and responsiveness.

2.3.4.1 Database



2.4 Technologies in Construction Management

2.4.1 Role of Web-based Platforms

Web-based platforms play an imperative role in making the building process more efficient through bringing centralized systems where all partners can collaborate. They

Connect clients and builders: Online platforms act as a market where clients can find credible contractors and where contractors can obtain project opportunities.

Streamlining processes: Project posting, bidding, and tracking among other features help automate most menial tasks, which automatically reduce errors and delays.

Make accessible: With the platform, it becomes convenient for builders and clients to share updates or even share documents anytime, anywhere.

Improves decision-making: It is possible to get good decisions with the following details on project specifications, and some bidding data, including all the communication tools.

2.4.2 Bidding Systems Benefits

Web-based platforms have a system called bidding systems, offering many benefits, including;

Fair Competition: Builders bid competitively to win contracts as they offer detailed proposals.

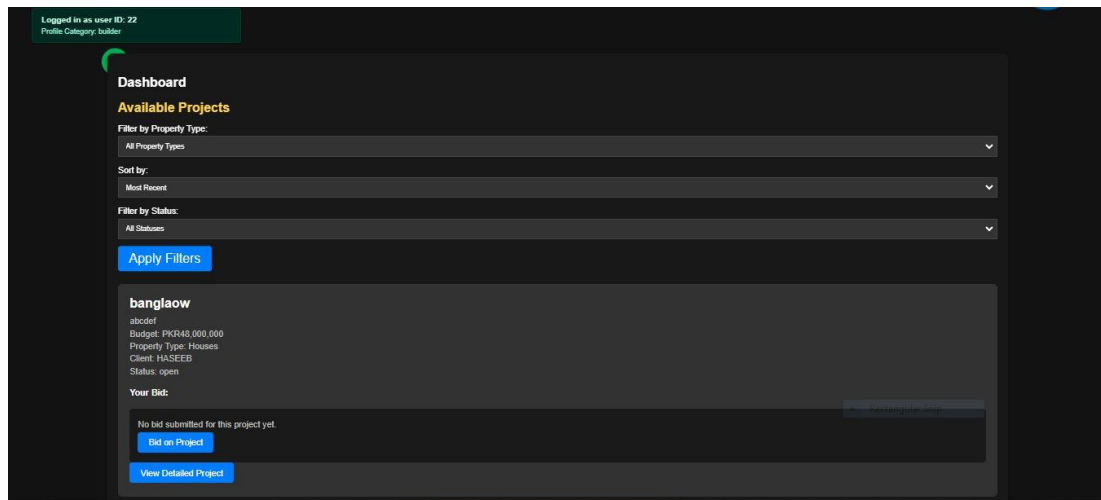
Transparency: Clients can compare bids based on cost, services offered, and timelines before making a decision.

Budget Management: Bids with clear breakdowns help clients understand where their money will be spent.

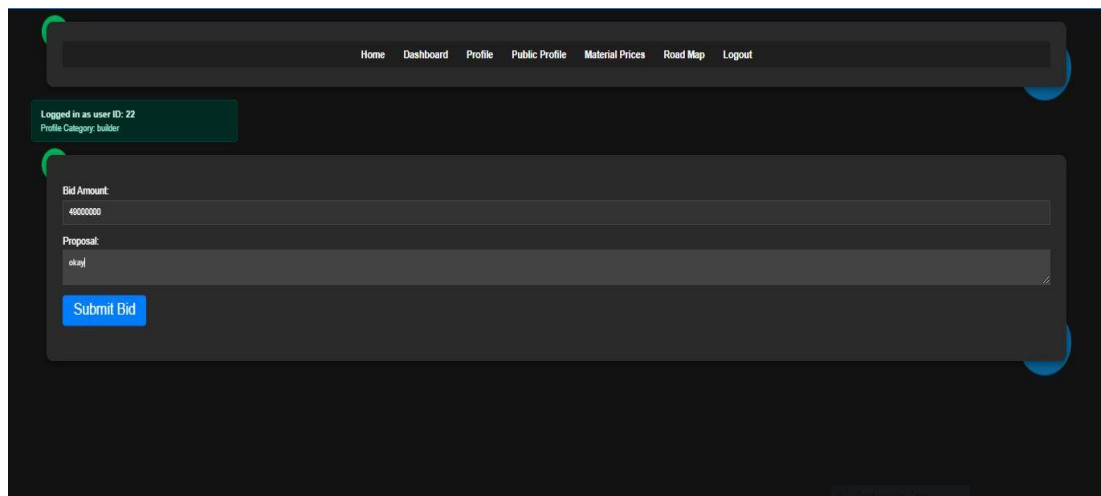
Builder Credibility: Builders can showcase their past projects within their bids, improving trust and increasing their chances of winning a project.

2.4.3 Bidding System

2.4.3.1 Projects



2.4.3.2 Bid On Project



2.4.3.3 Edit your Bid

The screenshot shows a web application interface for editing a bid. At the top, there is a navigation menu with links: Home, Dashboard, Profile, Public Profile, Material Prices, Road Map, and Logout. Below the navigation, a green box indicates the user is logged in as 'user ID: 45' with a profile category of 'builder'. The main content area is titled 'Edit Your Bid' and contains two input fields: 'Bid Amount' with the value '45000000' and 'Bid Description' with the value 'okay'. A blue 'Submit Bid' button is located below the description field. The footer contains the CityCraft logo and tagline, quick links (About Us, Services, Contact), social media links for Facebook and Instagram, and contact information (Email: info@citycraft.com, Phone: +1 234 567 890, Address: 123 Construction Lane, Build City, BC 45678).

2.5 Software Requirements Specification (SRS)

The Software Requirements Specification (SRS) describes the technical and functional requirements of the CityCraft platform.

2.5.1 Major Functional Requirements:

User Management:

Builders and clients must register with the platform, having detailed profiles.

Clients must input their project requirements, and builders must display their portfolios.

Bidding System:

Builders must submit proposals in detail, including cost estimates, timelines, and service breakdowns.

Clients must be able to accept, reject, or negotiate bids.

Material Prices:

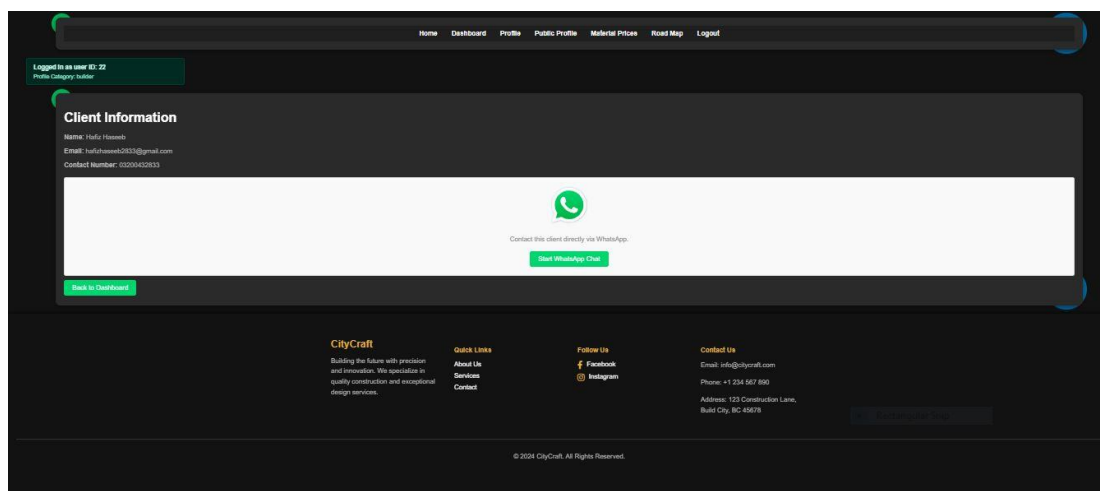
A dynamic system to display updated material prices, which would be sourced from local markets in Pakistan, must be available for reference.

Communication Tools:

Clients and builders must have direct chat options to discuss projects.

Notifications must be sent for bid updates, project milestones, and changes.

2.5.1.1 Chat Option



Construction Roadmaps:

Visual guides for the construction process must be available for both clients and builders, and they should be tailored for residential and commercial projects

2.5.2 Dashboard()

Table 2.5 : Dashboard

Here's the table for the **Dashboard** page (combining both client and builder functionality):

Name	Dashboard Page
Unique Identifier	CC-01-007
Objective	Display personalized content based on user type (Client or Builder).
Priority	High
Actors	Clients and Builders
Basic Flow	<ol style="list-style-type: none"> 1. User logs in and accesses the dashboard. 2. The system determines the user type (Client or Builder). 3. For clients, display their projects with associated bids. 4. For builders, display open projects with the builder's bid details. 5. Both users can interact with the projects and bids (e.g., accept/reject bids for clients, submit bids for builders).
Alternative Flow	<p>Alt-1. No projects found: Display a message indicating no projects are available.</p> <p>Alt-2. No bids for a project: Display a message indicating no bids yet.</p> <p>Alt-3. Error loading data: Display an error message and prompt retry.</p>
Preconditions	<p>User must be logged in.</p> <p>Projects and bids must be available in the database.</p>
Postconditions	<p>Dashboard is populated with relevant data based on the user type.</p> <p>Users can interact with projects and bids.</p>
Notes/Issues	<p>Ensure smooth handling of dynamic content for both client and builder views.</p> <p>Optimize queries for performance to avoid delays in data loading.</p> <p>Handle edge cases, such as no available bids or no active projects.</p>

2.5.3 Create Project()

Table 2.6 : Create Project

Name	Create Project Page
Unique Identifier	CC-01-006
Objective	Allow clients to create new projects, specifying project details such as title, description, budget, and location.
Priority	High
Actors	Clients
Basic Flow	<ol style="list-style-type: none"> 1. Client logs in and accesses the "Create Project" form. 2. Client fills in the project details (e.g., title, description, budget, location, property type). 3. Client submits the form to create the project. 4. Project is saved in the database and displayed on the client's dashboard.
Alternative Flow	<p>Alt-1. Missing or invalid data: Show an error message and prompt the user to correct the input.</p> <p>Alt-2. User navigates away from the form without submitting: Data is not saved.</p>
Preconditions	<p>User must be logged in as a client.</p> <p>Location and property type data must be available.</p>
Postconditions	<p>A new project is successfully created and saved in the database.</p> <p>The project appears on the client's dashboard.</p>
Notes/Issues	<p>Ensure proper form validation (e.g., for required fields, numeric budget).</p> <p>Handle file uploads for images (if applicable).</p> <p>Improve form layout and user experience with modern design and animations.</p>

2.5.3.1 Create Project

Logged in as user ID: 23
Profile Category: client

Create New Project

Project Title:
SPANISH

Project Description:
House with 6 bedrooms

Budget (PKR):
3740000

Project Location:
Lahore

Property Type:
Houses

Category:
House

Area Size:
10 Mails

Front Elevation:
Spanish

No. of Stories:
More

[Create Project](#)

Home Dashboard Profile Create Project Material Prices Road Map Logout

Logged in as user ID: 23
Profile Category: client

Bid status updated successfully!

Dashboard

Your Projects

[Create New Project](#)

villa
Budget: PKR78,000,000

Bids
Muhammad Haseeb: PKR98,000,000.00
Description: ok
Status: accepted
[View Bidder Profile](#)
[View Detailed Project](#)

SPANISH
House with 6 bedrooms
Budget: PKR37,400,000

Bids
No bid submitted for this project yet.
[View Detailed Project](#)

2.5.3.2 Project Details

Logged in as user ID: 46
Profile Category: client

Project Details

Home

Description: I want a 10 Marla House with following requirements: Ground Floor Bedrooms: 2 Bedrooms with attached bathrooms for comfort and privacy. Spacious wardrobes or closets for storage. Living Room: A cozy living room for family gatherings, located near the dining area. Drawing Room: A formal drawing room for guests, designed with elegant interiors. Dining Room: A dedicated dining area near the kitchen for family meals. Kitchen: A modern kitchen with ample storage cabinets, countertop space, and provisions for appliances. Powder Room: A small powder room for guests. Lawn/Garden: A small front or back lawn for greenery and outdoor relaxation. First Floor Bedrooms: 2-3 additional bedrooms with attached bathrooms for other family members or guests. Living Family Lounge: A central lounge for casual family time. Kitchennette: A small kitchennette for convenience on the first floor. Terrace: An open terrace for outdoor seating and enjoying fresh air. Storage: A dedicated storage room or area. Other Features Garage/Car Porch: Covered parking space for 1-2 vehicles. Staircase: A centrally or side-located staircase for seamless movement between floors. Roof Access: Stairs leading to the roof, which can be used for solar panels or additional utility purposes.

Budget: PKR42,000,000

Location: Lahore

Property Type: Houses

Additional Information

Front Elevation: Modern

Stories: Double

Builder Bids

Abdul Haseeb: PKR44,000,000.00
Bid Status: accepted
Description: I will give you all these requirements as you mentioned in the project.

[Back to Dashboard](#)

CHAPTER 3

DESIGN AND METHODOLOGY

3.1 Platform Architecture

CityCraft is designed with a two-dashboard architecture tailored to the needs of its two major users: clients and builders.

3.1.1 Client Dashboard:

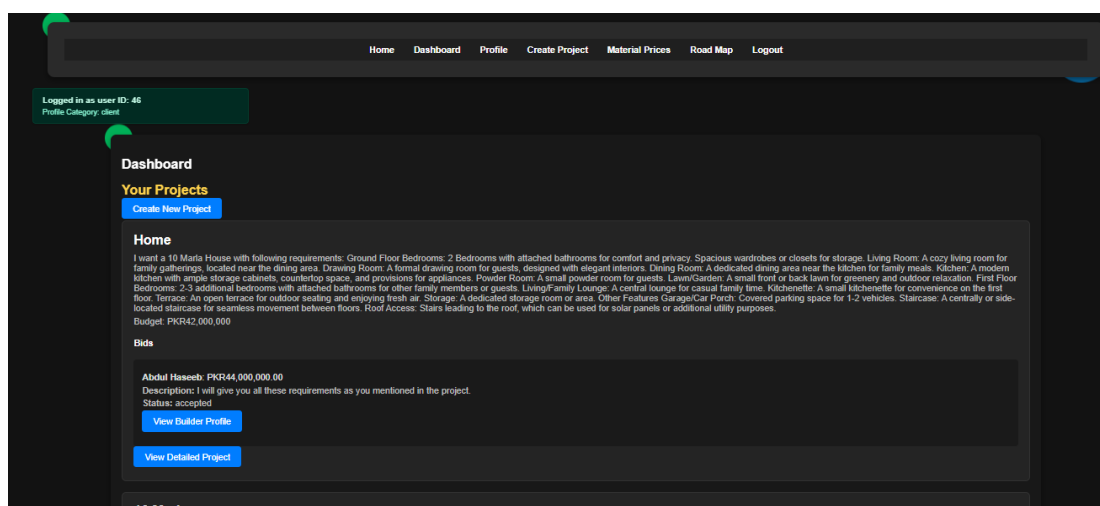
This dashboard gives the client an easy means to manage their projects. The client can:

Publish a detailed project specification including budget and deadlines.

View submitted builder bids and analyze cost breakdowns.

Track their on-going projects to be able to know whether they are within schedule.

3.1.1.1 Client Dashboard



3.1.2 Builder Dashboard:

The builder's dashboard is a tool that empowers construction companies to express themselves. Builders can:

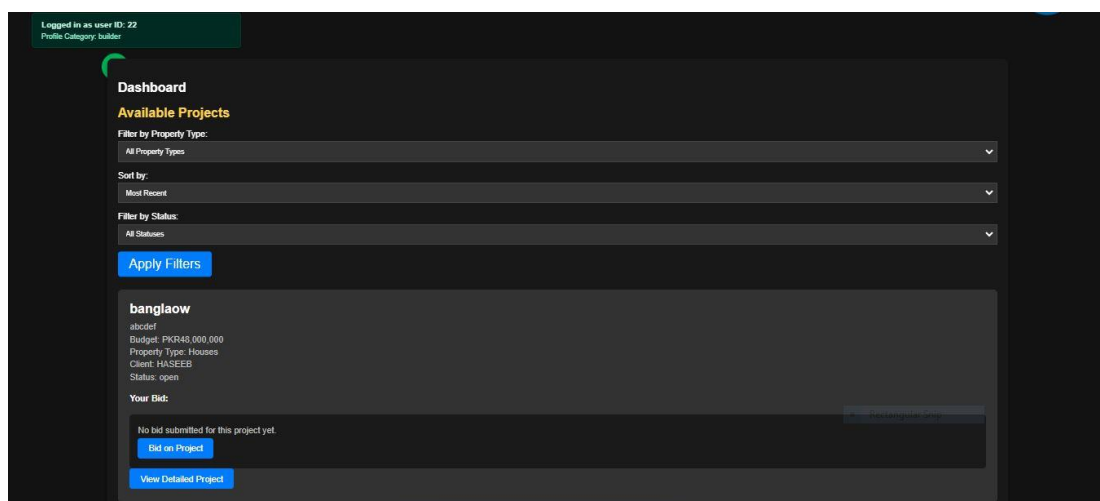
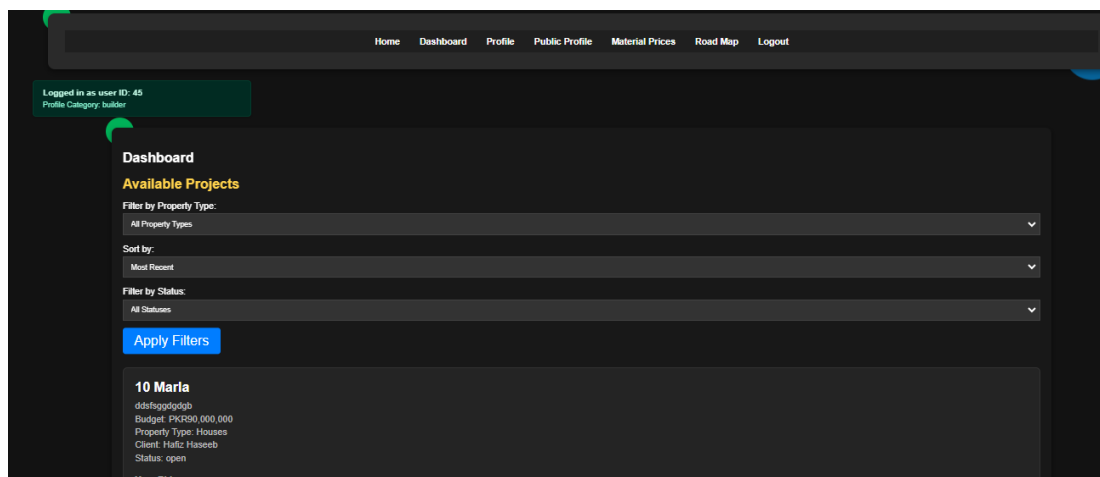
Update their profiles and portfolios with completed projects and achievements.

Access a list of available projects posted by clients.

Submit bids on projects, including service details and cost estimates.

The architecture is such that it ensures smooth information flow between the clients and builders. This way, the whole process of project management will be transparent and efficient.

3.1.2.1 Builder Dashboard



3.1.3 Client Dashboard()

Table 3.1 : Client Dashboard

Name	Client Dashboard Page
Unique Identifier	CC-01-007
Objective	Display a personalized dashboard for clients, showing their posted projects along with bids from builders.
Priority	High
Actors	Clients
Basic Flow	<ol style="list-style-type: none"> 1. Client logs in and accesses their dashboard. 2. The system fetches the client's projects using <code>getProjectsByUserId</code>. 3. For each project, the bids are fetched using <code>getBidsByProjectIdWithBuilderName</code>, and builder names are included. 4. Client views their project details and bid information. 5. Client can choose to accept or reject bids using <code>updateBidStatus</code>.
Alternative Flow	<p>Alt-1. No projects found: Display a message indicating no projects are available.</p> <p>Alt-2. No bids for a project: Display a message indicating no bids yet.</p> <p>Alt-3. Error loading projects/bids: Display an error message prompting the user to refresh.</p>
Preconditions	<p>Client must be logged in.</p> <p>Client must have at least one project.</p> <p>Bids must exist for projects.</p>
Postconditions	<p>Client's dashboard is populated with their projects and associated bids.</p> <p>The client can interact with bids (accept/reject).</p>
Notes/Issues	<p>Ensure data consistency between projects and bids.</p> <p>Optimize data retrieval for quick dashboard loading.</p> <p>Handle the case where no bids have been placed yet.</p>

3.1.4 Builder Dashboard()

Table 3.2 : Builder Dashboard

Name	Builder Dashboard Page
Unique Identifier	CC-01-008
Objective	Display a personalized dashboard for builders, showing open projects from clients along with their bids for each project.
Priority	High
Actors	Builders
Basic Flow	<ol style="list-style-type: none"> 1. Builder logs in and accesses their dashboard. 2. The system fetches all open projects using <code>getAllOpenProjectsWithClientNames</code>. 3. For each project, the builder's bid is fetched using <code>getBuilderBidForProject</code>. 4. Builder views the project details, client name, and their bid status. 5. Builder can update their bid for any project or submit new bids.
Alternative Flow	<p>Alt-1. No open projects available: Display a message indicating no open projects are available.</p> <p>Alt-2. No bid placed for a project: Display a message indicating no bid has been placed.</p> <p>Alt-3. Error loading projects/bids: Display an error message prompting the user to refresh.</p>
Preconditions	<p>Builder must be logged in.</p> <p>Open projects must be available.</p> <p>Builder must have made bids on the open projects.</p>
Postconditions	<p>Builder's dashboard is populated with open projects and bid statuses.</p> <p>Builder can interact with their bids and projects.</p>
Notes/Issues	<p>Ensure proper handling of bid submission and updates.</p> <p>Optimize data retrieval for faster dashboard loading.</p> <p>Handle the case where no bid has been placed for a project yet.</p>

3.1.5 Builder profile()

Table 3.3 : Builder Profile

Name	Builder Profile Page
Unique Identifier	CC-01-009
Objective	Display the builder's profile information, including their previous projects.
Priority	High
Actors	Builder (logged in), Client (viewing builder profile via bid).
Basic Flow	<ol style="list-style-type: none"> 1. User navigates to the builder profile page. 2. The system checks if the user is logged in and has the correct user type. 3. If the user is a client, fetch the builder's ID based on the bid information. 4. The system fetches and displays the builder's profile information (name, email) and previous projects (title, description, location, property type, images). 5. The builder can manage their projects through a provided button if they are logged in as a builder. 6. The page is displayed with relevant information.
Alternative Flow	<p>Alt-1. If the user is not logged in or does not have a valid session, display an error message.</p> <p>Alt-2. If no builder profile or project information is found, show an appropriate message.</p> <p>Alt-3. If a project does not have an image, display a "No image available" message.</p>
Preconditions	User must be logged in, and user type should be either "client" or "builder". For clients, a valid bid ID is required.
Postconditions	Builder profile information is displayed, including their previous projects. The builder can manage their projects if logged in.
Notes/Issues	<p>Ensure that the database query for fetching builder details and projects handles errors properly.</p> <p>Consider sanitizing the output to avoid security issues such as XSS.</p>

3.1.5.1.1 Create Builder Projects

Home Dashboard Profile Public Profile Material Prices Road Map Logout

Logged in as user ID: 45
Profile Category: builder

Manage Projects

Add New Project

Title

Description

Location

Property Type

House

Upload Image

Choose file No file chosen

Add Project

3.1.5.1.2 Builder Profile

Home Dashboard Profile Public Profile Material Prices Road Map Logout

Logged in as user ID: 22
Profile Category: builder


Builder Profile

Name: Hafiz Abdul Hasseb
Email: abdulhasseb1995@gmail.com

Previous Projects

2 kanal

Description: Spanish House
Location: Valencia, Lahore
Property Type: House



3.2 Development Methodology

The development of *CityCraft* follows a structured approach, leveraging an efficient combination of frontend and backend technologies. The primary goal was to create a seamless, user-friendly web platform to connect clients and builders, facilitating a transparent and efficient project procurement process in the construction industry.

3.2.1 Technology Stack

To develop *CityCraft*, we selected the following technologies:

- **XAMPP Server:**

XAMPP is an open-source, cross-platform web server solution stack package. It includes the Apache HTTP server, MySQL database, and interpreters for PHP and Perl. It is widely used for local development and testing purposes due to its simplicity and ease of setup.

- **PHP (Backend Language):**

PHP was chosen for the server-side scripting due to its reliability, speed, and flexibility. It is ideal for handling dynamic content, interacting with the database, and performing server-side operations, such as user authentication, bidding, and project management functionalities. PHP is well-suited for building secure, scalable web applications.

- **Frontend Technologies (HTML, CSS, JavaScript):**

- **HTML** provides the structure and content of the platform. We used HTML5 for its modern features, such as multimedia support, forms, and structural elements.
- **CSS** was used to style the platform, making it visually appealing and user-friendly. CSS3 features like Flexbox and Grid were employed for responsive design, ensuring compatibility with different screen sizes.
- **JavaScript** was used to add interactivity to the platform. With JavaScript, users can interact with elements on the page dynamically, such as submitting forms, displaying modal dialogs, and updating content without reloading the page.

3.2.2 Development Phases

The development of *CityCraft* followed a step-by-step methodology:

1. **Planning and Requirement Analysis:**

During the initial phase, we defined the core features of the platform, such as builder profiles, client project listings, a bidding system, project tracking, and secure payment integration. The aim was to ensure that the system would be user-centric, secure, and scalable.

2. **System Architecture Design:**

We designed a modular architecture that separated the frontend and backend, allowing for easy management and scalability. The architecture consisted of:

- **A Frontend Layer:** HTML, CSS, and JavaScript were used for client-side interaction, ensuring a smooth user experience.
- **A Backend Layer:** PHP scripts handled user management, project submission, bidding, and database interactions.
- **A Database Layer:** MySQL (included with XAMPP) was used to store user data, project details, bids, and transaction information.

3. **Frontend Development:**

The user interface was designed with simplicity and functionality in mind. We focused on making navigation intuitive, using a combination of HTML for structure, CSS for styling, and JavaScript for dynamic content rendering. Features like form validation, project previews, and interactive bidding mechanisms were implemented using JavaScript.

4. **Backend Development:**

The backend was developed using PHP. PHP scripts handle:

- User authentication (login, registration, password recovery).
- Bidding system (builder proposals, client project specifications).
- Secure project tracking and milestone updates.

The backend interacts with a MySQL database to store and retrieve data as needed, ensuring the platform's functionality.

5. Database Design:

The database was designed to handle key elements, such as:

- **Users:** Information on clients and builders.
- **Projects:** Project details, timelines, budgets, and bids.
- **Bids:** Builder submissions, including cost estimates and project timelines.

6. Testing and Debugging:

Once the core features were implemented, thorough testing was conducted.

The testing phases included:

- **Unit Testing:** Ensuring individual functions and components performed as expected.
- **Integration Testing:** Verifying that different parts of the system (frontend, backend, and database) worked together seamlessly.
- **Security Testing:** Ensuring secure login, data encryption, and safe payment processing.
- **User Testing:** Gathering feedback from potential users to identify any usability issues.

7. Deployment and Maintenance:

After successful testing, *CityCraft* was deployed on a live server. XAMPP's local development environment was initially used for testing, and once fully functional, the application was migrated to a production environment. Continuous maintenance was planned to handle bug fixes, performance optimization, and feature enhancements.

3.2.3 Workflow of the Platform

1. **Client Interaction:** Clients can register on the platform, upload their project specifications, and receive bids from construction companies. The client has the ability to track progress, compare proposals, and select the best bid.
2. **Builder Interaction:** Builders register, create detailed profiles showcasing their previous work, and submit proposals for listed projects. They can also communicate directly with clients, ask questions, and finalize project details.

3. Bidding System:

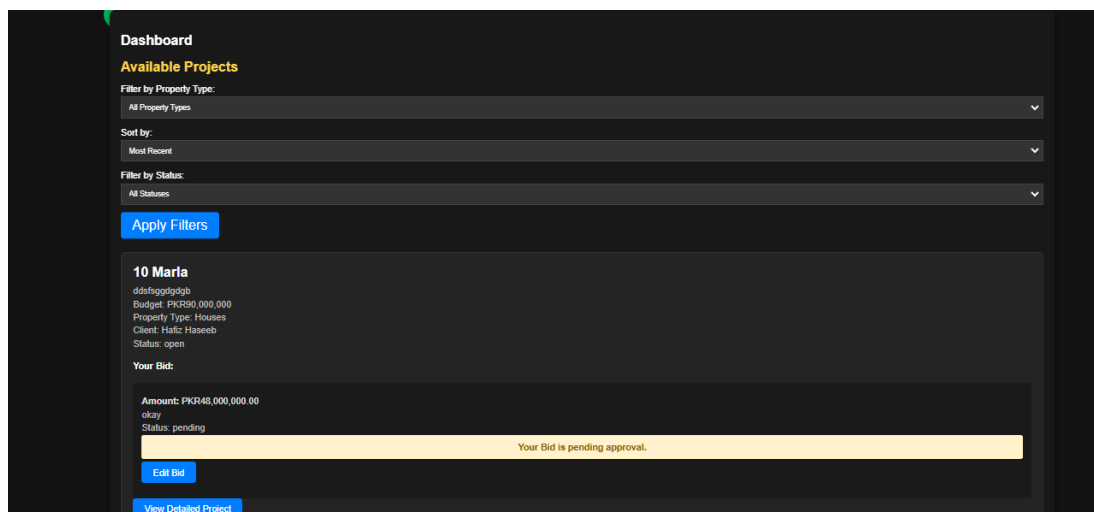
A transparent and competitive bidding system allows builders to propose cost estimates and timelines for client projects. Clients evaluate bids based on builder profiles, proposals, and past project success.

3.2.4 Bidding system()

Table 3.4 : Bidding system

Name	Bidding System Page
Unique Identifier	CC-02-010
Objective	Allow builders to place bids on a project by specifying a bid amount and proposal.
Priority	High
Actors	Builder (logged in to submit a bid)
Basic Flow	<ol style="list-style-type: none"> 1. Builder navigates to the bidding page for a specific project. 2. The system fetches and displays the details of the project. 3. Builder enters their bid amount and proposal in the form. 4. The system inserts the bid into the <code>bids</code> table in the database. 5. Success or error message is displayed after submission. 6. Builder is redirected or stays on the same page with feedback.
Alternative Flow	<p>Alt-1. If no <code>project_id</code> is found in the URL, the page displays an error ("No project selected.").</p> <p>Alt-2. If the project does not exist, an error message is shown ("Project not found.").</p> <p>Alt-3. If the bid submission fails, an error message with the details is displayed.</p>
Preconditions	Builder must be logged in to place a bid.
Postconditions	The bid is successfully recorded in the <code>bids</code> table, and a success message is shown.
Notes/Issues	<p>Ensure builder authentication is properly handled.</p> <p>Validate and sanitize user input for bid amount and proposal.</p> <p>Handle cases where no project is selected or if the project does not exist.</p> <p>Consider redirecting the builder after a successful bid submission to avoid resubmission.</p>

3.2.4.1 After Submit the Bid it will shown On the Project



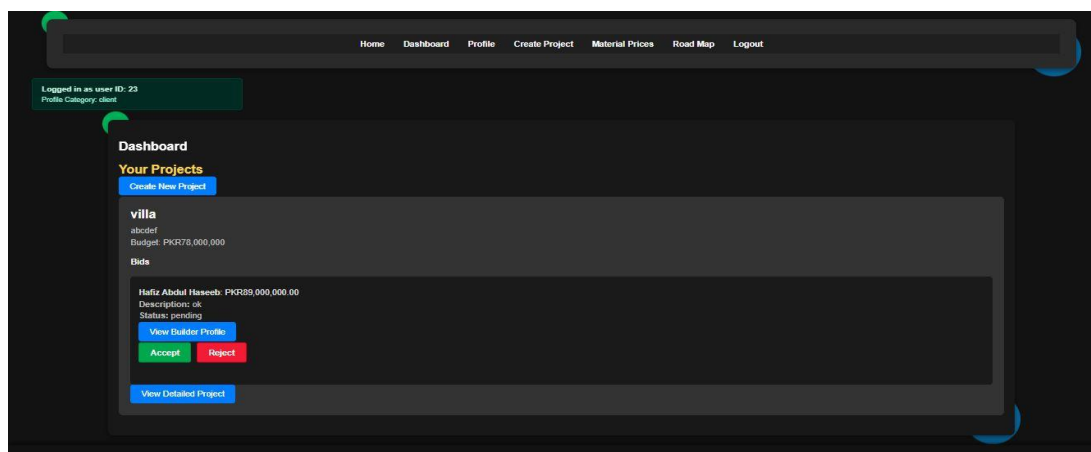
3.2.5 Accept/Reject bid()

Table 3.5 : Accept/Reject Bids

Name	Accept and Reject Bid System
Unique Identifier	CC-02-011
Objective	Allow clients to accept or reject bids placed by builders on their projects.
Priority	High
Actors	Client (logged in to accept or reject bids)
Basic Flow	<ol style="list-style-type: none"> 1. Client is logged in and views bids on their project. 2. Client selects a bid to accept or reject. 3. Upon accepting a bid, the bid status is updated to "accepted", the project is closed, and other bids are rejected. 4. Upon rejecting a bid, the bid status is updated to "rejected". 5. After accepting or rejecting a bid, the client is redirected to the project page with a success message.
Alternative Flow	<p>Alt-1. If the client is not logged in or is not of type "client", they are redirected to the login page.</p> <p>Alt-2. If bid acceptance or rejection fails, an error message is displayed.</p>

Name	Accept and Reject Bid System
Preconditions	Client must be logged in to accept or reject bids. The project must have bids placed on it.
Postconditions	The bid's status is updated to "accepted" or "rejected", and the system ensures only one bid is accepted per project.
Notes/Issues	<ul style="list-style-type: none"> - Ensure client authentication is correctly handled. - Bid status updates must reflect across all views (project, bids, etc.). - The <code>updateBidStatus</code> method should handle status changes properly. - The project closure logic should handle transitions properly (e.g., preventing further changes). - Ensure no accidental acceptance of multiple bids for the same project.

3.2.5.1 Accept/Reject Bids



3.3 Core Functional Modules

The User Management Module is the heart of CityCraft, which enables clients and builders to:

Create Accounts: Users can sign up and log in to access the platform.

Manage Profiles:

Clients can upload personal details, project requirements, and view bid statuses.

Builders can update their portfolios, including past projects and areas of expertise.

Update Information:

Clients can edit project details, such as deadlines or budgets.

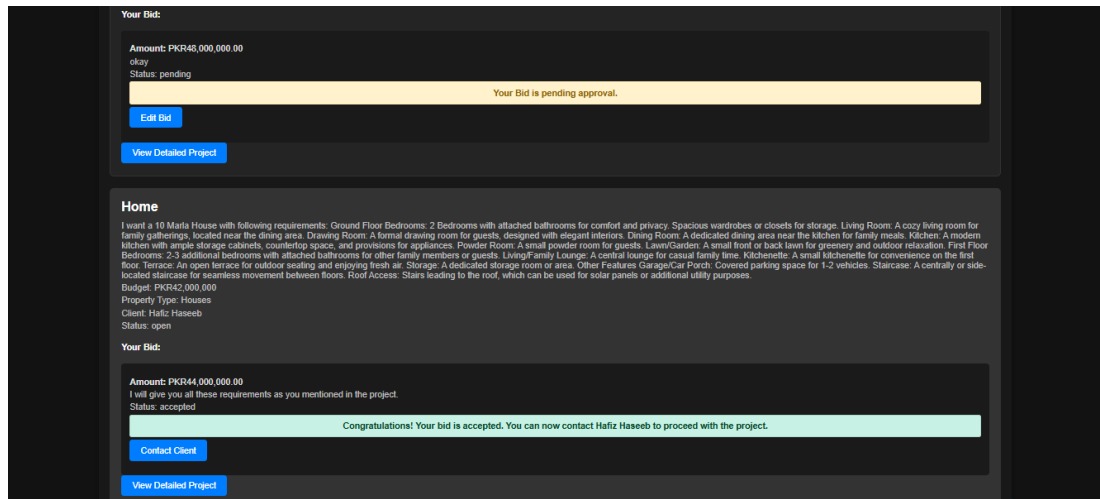
Builders can revise portfolio elements or modify their bids before final submission.

3.3.1 View Bid Status()

Table 3.6 : Bid Status

Name	View Bid Statuses Functionality
Unique Identifier	CC-02-012
Objective	Display bid details for a project, including builder's name, bid amount, and description, and provide a link to the builder's profile.
Priority	Medium
Actors	Client (viewing bids on their project), Builder (profile view)
Basic Flow	<ol style="list-style-type: none"> 1. The system retrieves and displays a list of bids associated with a project. 2. For each bid, the builder's name, bid amount (formatted as PKR), and description are shown. 3. A "View Builder Profile" button is displayed alongside each bid. Clicking it redirects the user to the builder's profile page.
Alternative Flow	<ul style="list-style-type: none"> - If no bids are available for the project, a message can be displayed indicating no bids are available. - If the builder profile link is unavailable, the button may be hidden or display a message indicating an issue with accessing the profile.
Preconditions	<p>The project should have bids associated with it in the database. The bid information (builder name, bid amount, description) must be properly linked to the project.</p>
Postconditions	<p>The bid details and the associated builder's profile button are displayed on the page.</p>
Notes/Issues	<ul style="list-style-type: none"> - Ensure proper sanitization of user input for security (e.g., <code>htmlspecialchars</code> for preventing XSS attacks). - Add validation or error handling for the case when bid details or builder profiles are missing. - Consider pagination if the number of bids is large, for better user experience.

3.3.1.1 Bid Status



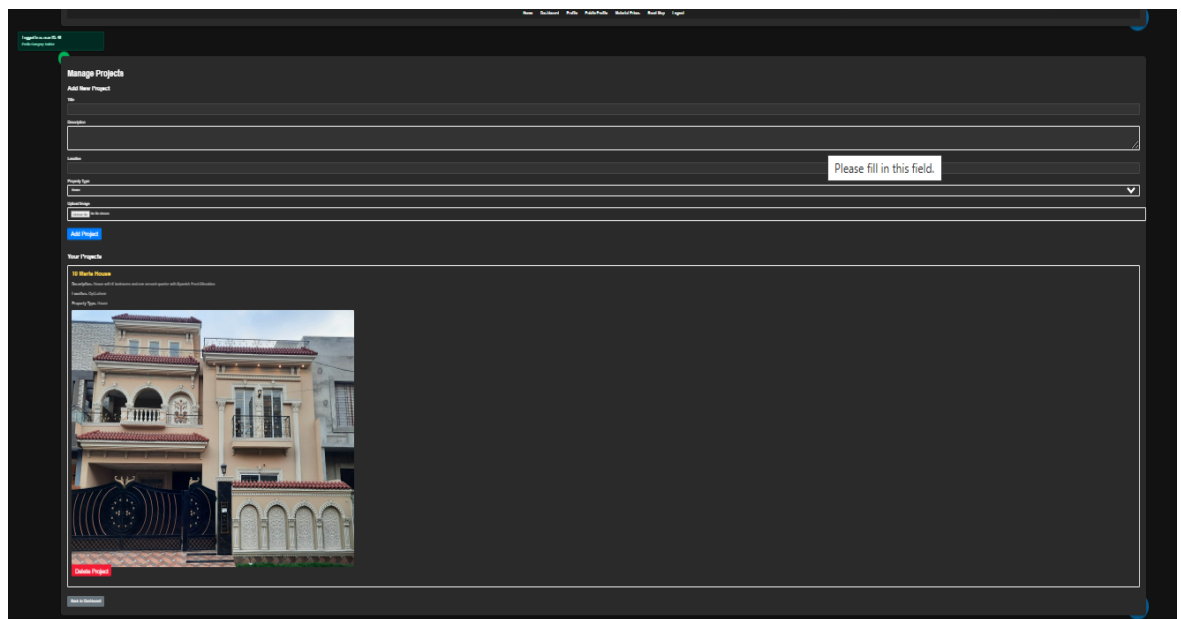
3.3.2 Manage Projects()

Table 3.7 : Manage Projects

Name	Manage Projects Page
Unique Identifier	CC-01-013
Objective	Allow builders to add, view, and delete their projects.
Priority	High
Actors	Builder (logged in as builder)
Basic Flow	<ol style="list-style-type: none"> 1. Builder navigates to the manage projects page. 2. The system checks if the user is logged in as a builder. 3. Builder can add a new project by filling out the form with project details and uploading an image. 4. Builder can view their existing projects with the option to delete any project. 5. Upon submitting the form, the new project is added to the database, or the selected project is deleted. 6. Success or error messages are displayed accordingly.
Alternative Flow	<p>Alt-1. If the builder is not logged in as a builder, an error message is displayed.</p> <p>Alt-2. If no projects are available, the page will show a message stating "No projects uploaded yet."</p>

Name	Manage Projects Page
	Alt-3. If the project image upload fails, an error message is displayed. Alt-4. If deletion of a project fails, an error message is displayed.
Preconditions	User must be logged in as a builder.
Postconditions	Builder can add or delete projects from their portfolio.
Notes/Issues	Ensure that the database query for adding, deleting, and fetching projects handles errors properly. Validate and sanitize form inputs to prevent malicious data. Ensure uploaded images are stored securely, and only images with allowed file types are uploaded.

3.3.2.1 Manage Projects



CHAPTER 4

DATA AND EXPERIMENTS (and/or IMPLEMENTATION)

4.1 Database Design

The database is a core component of the platform, designed to store and manage all the necessary information securely and efficiently. The following key entities are part of the database:

User Profiles:

Contains information about clients and builders, including their names, contact information, and profiles.

Builders' profiles also include their portfolios and completed projects.

Projects:

Stores client-uploaded project details, such as project title, specifications, deadlines, and budget.

Tracks the status and progress of every project .

Tenders:

Keeps a record of all proposals received by builders, such as estimated costs, timelines, and service descriptions.

Associates each tender with its particular project and builder.

Construction Material Prices:

Provides a real-time database of updated construction material prices sourced from local markets.

Helps clients and builders estimate costs accurately.

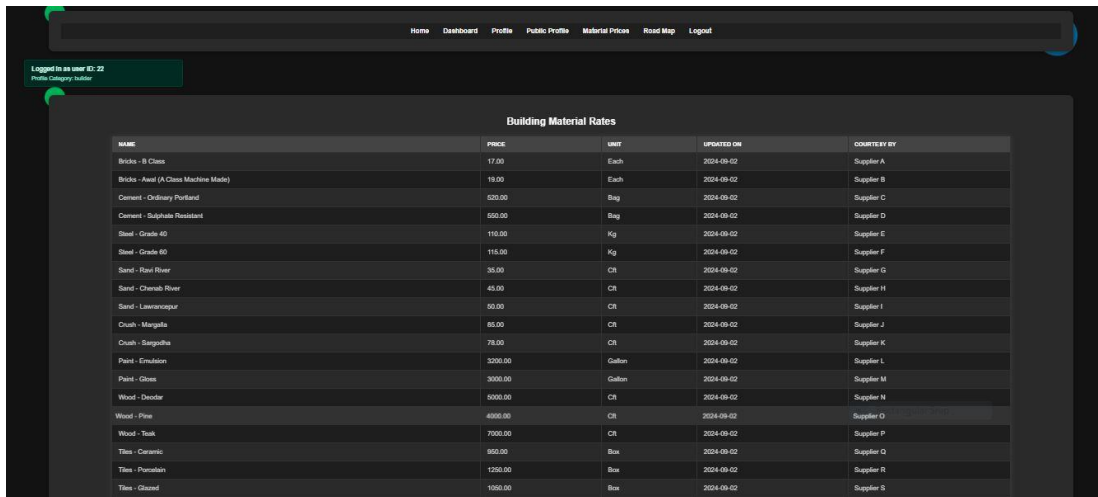
MySQL and is used with phpMyAdmin with a XAMPP server to provide for data processing of a strong, robust and highly secure system.

4.1.1 Materialratelist()

Table 4.1 : Material Ratelist

Name	Material List Functionality
Unique Identifier	CC-02-014
Objective	Display a list of building materials with their prices, units, update dates, and the entity providing them.
Priority	Medium
Actors	Client, Builder (viewing material rates)
Basic Flow	<ol style="list-style-type: none"> 1. The system fetches data from the <code>ratelist</code> table in the database. 2. Material name, price, unit, updated date, and courtesy provider are displayed in a table. 3. If no data is found, a message "No data found" is shown.
Alternative Flow	<ul style="list-style-type: none"> - If the database is unreachable or query fails, show an error message. - If no materials are available, show a message like "No data available."
Preconditions	<ul style="list-style-type: none"> - The <code>ratelist</code> table should be populated with material data. - The system must have a working connection to the database. - Ensure the <code>public/css/style.css</code> is loaded for styling the table and page.
Postconditions	The material list is displayed to the user in a structured table format.
Notes/Issues	<ul style="list-style-type: none"> - Ensure that the data is properly sanitized to avoid XSS or SQL injection vulnerabilities. - The CSS class <code>styled-table</code> should be defined in the <code>style.css</code> for proper table styling. - Optimize for responsiveness across different devices.

4.1.1.1 Material Prices



NAME	PRICE	UNIT	UPDATED ON	COURTESY BY
Bricks - B Class	17.00	Each	2024-09-02	Supplier A
Bricks - Aerial (A Class Machine Made)	19.00	Each	2024-09-02	Supplier B
Cement - Ordinary Portland	520.00	Bag	2024-09-02	Supplier C
Cement - Sulphate Resistant	550.00	Bag	2024-09-02	Supplier D
Steel - Grade 40	110.00	Kg	2024-09-02	Supplier E
Steel - Grade 60	115.00	Kg	2024-09-02	Supplier F
Sand - Rawl River	35.00	CB	2024-09-02	Supplier G
Sand - Chenab River	45.00	CB	2024-09-02	Supplier H
Sand - Lawanaspur	60.00	CB	2024-09-02	Supplier I
Crush - Mangala	85.00	CB	2024-09-02	Supplier J
Crush - Sargolha	78.00	CB	2024-09-02	Supplier K
Paint - Emulsion	3200.00	Crabon	2024-09-02	Supplier L
Paint - Glass	3000.00	Crabon	2024-09-02	Supplier M
Wood - Deodar	8000.00	CB	2024-09-02	Supplier N
Wood - Pine	6900.00	CB	2024-09-02	Supplier O
Wood - Teak	7000.00	CB	2024-09-02	Supplier P
Tiles - Ceramic	850.00	Box	2024-09-02	Supplier Q
Tiles - Porcelain	1280.00	Box	2024-09-02	Supplier R
Tiles - Glazed	1050.00	Box	2024-09-02	Supplier S

Construction Roadmap:

This **Construction Roadmap** provides a step-by-step visual guide for two construction types: **Houses** and **High-Rise Buildings**. The roadmap highlights various construction phases, each with an associated icon and description.

For Houses:

Step 1: Digging (Excavation)

Description: Excavation and leveling of the plot for foundation work.

Step 2: Foundation

Description: Laying base footings, damp-proof course, and reinforced steel framework.

Step 3: Grey Structure

Description: Building walls, columns, roofs, and lintels with concrete and bricks.

Step 4: Utilities

Description: Installing electrical wiring, plumbing, and HVAC systems.

Step 5: Finishing

Description: Applying plaster, flooring, painting, fixtures, and doors/windows.

For High-Rise Buildings:**Step 1: Site Clearance**

Description: Clearing debris and marking boundaries for construction.

Step 2: Piling

Description: Drilling deep into the ground to create strong piles for the foundation.

Step 3: Foundation

Description: Pouring reinforced concrete for slabs and pile caps.

Step 4: Superstructure

Description: Constructing columns, beams, and core walls for multiple floors.

Step 5: Facade and Exterior

Description: Installing glass panels, cladding, and external elevators.

Step 6: MEP Services

Description: Fitting mechanical, electrical, and plumbing systems for all floors.

Step 7: Interior Works

Description: Painting, flooring, ceiling works, and final finishes for units.

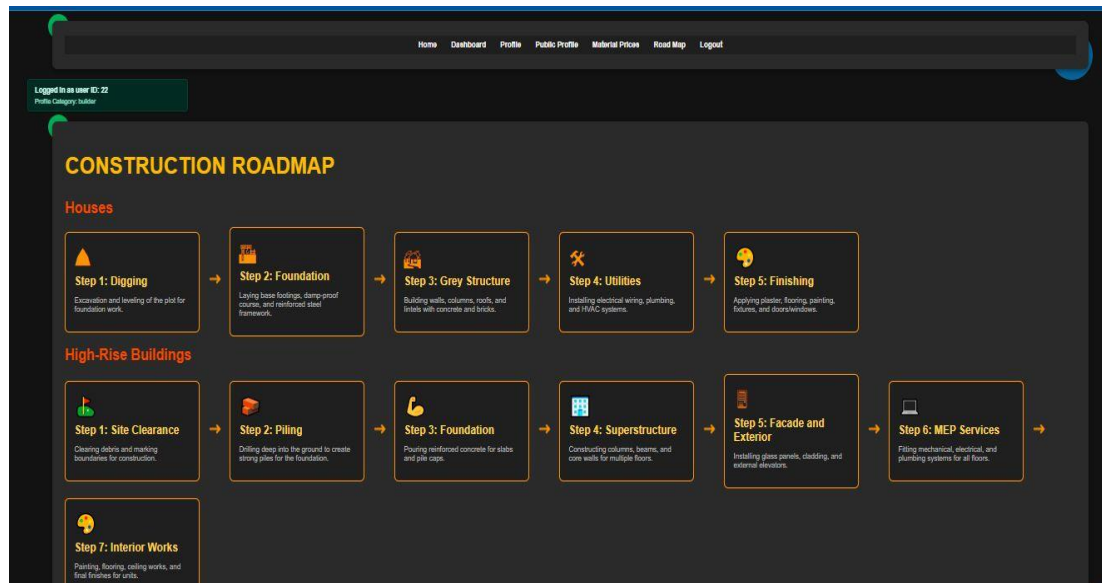
4.1.2 Construction Roadmap

Table 4.2 : Construction Roadmap

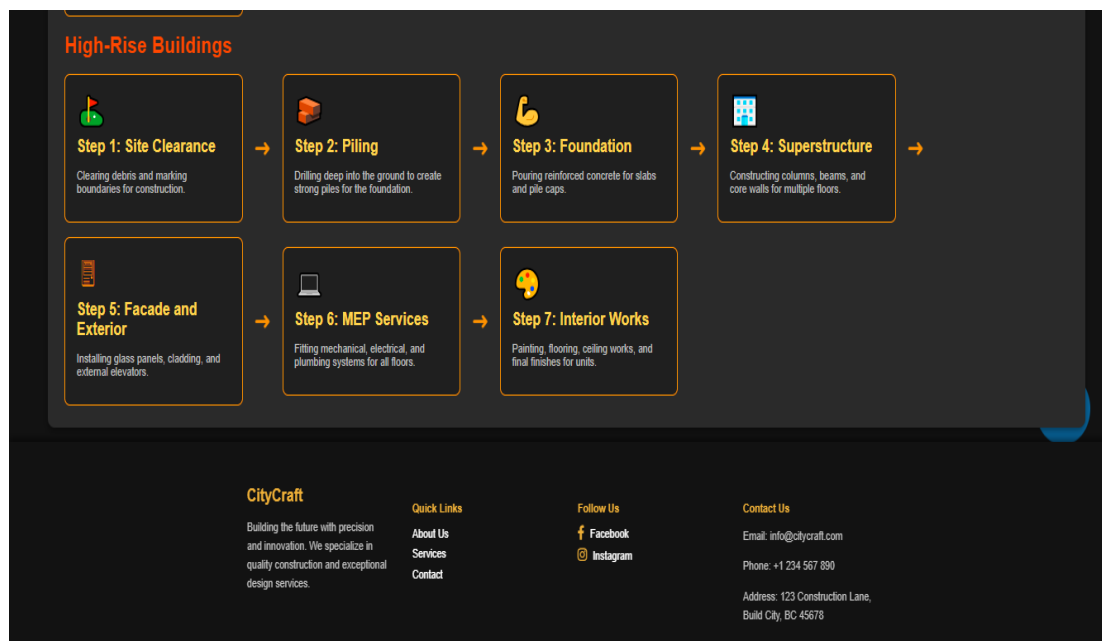
Name	Construction Roadmap Functionality
Unique Identifier	CC-02-015
Objective	Display the step-by-step construction roadmap for houses and high-rise buildings.
Priority	Medium
Actors	Client, Builder (viewing the construction process)
Basic Flow	<ol style="list-style-type: none"> 1. The user views the construction roadmap for two categories: Houses and High-Rise Buildings. 2. For each category, the roadmap steps are displayed in sequential order with icons and descriptions. 3. The roadmap includes key steps for each construction type, such as "Digging," "Foundation," "Utilities," etc. for houses, and "Site Clearance," "Piling," "Superstructure," etc. for high-rise buildings.
Alternative Flow	- If no roadmap steps are available, display a message like "No construction roadmap available."
Preconditions	<ul style="list-style-type: none"> - The system must have predefined roadmap steps for houses and high-rise buildings. - Ensure that proper HTML and CSS for icons, styling, and layout are available to render the roadmap sections visually.
Postconditions	The construction roadmap is displayed clearly in sections for each category (Houses, High-Rise Buildings).
Notes/Issues	<ul style="list-style-type: none"> - Ensure that the roadmap is responsive for different devices (e.g., mobile, tablet, desktop). - Check if all the icons are properly displaying and relevant to the respective construction step. - Add a feature to possibly filter or customize the roadmap depending on user selection or preferences.

4.1.2.1 Construction Roadmap

4.1.2.1.1 For Houses



4.1.2.1.2 For Buildings



4.2 Technology Stack

It employs cutting edge technologies to achieve ease and smooth flow. The essential building blocks of the technology stack include are:

Front-end:

HTML, CSS and JavaScript - Used for website construction that interacts and visually rich with customers.

Enables both clients and builders to easily access and use the platform for posting a project, bidding, and viewing material prices.

Backend

PHP: This aspect deals with the server logic, including user authentication, creating and managing projects, submittal of bids and updating material prices.

Allows for secure communication between frontend and database.

Database

MySQL via phpMyAdmin - Manages and stores in a structured format all user details, project information, bids placed, and material prices.

Ensures data integrity and allows easy retrieval of data for different operations.

Server Environment:

XAMPP: It is a local server which runs the application during development and testing. This is a combination of Apache, MySQL, PHP, and Perl for a smooth development environment.

4.3 Functional Testing

4.3.1 User Experience Testing

To make sure that the platform is intuitive and user-friendly, we performed several rounds of testing. Key tests include:

Navigation:

Verified that users can navigate the platform with ease from creating profiles to posting projects and viewing bids.

Ensure all buttons and links were found working without broken paths.

Project Posting

Tested the project posting process to ensure that the clients could add project details and requirements without errors.

Bidding

Verified that the builders could view the available projects, submit their proposals, and edit the proposal if necessary.

Ensured that clients could view, accept, or reject tenders with ease.

Material Price Updates:

Confirmed that the material price list displayed accurate and up-to-date prices.

Communication Tools:

Verified the chat system to ensure smooth communication between clients and builders without delays.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 Platform Performance

CityCraft bridges the gap between clients and builders, ensuring smooth communication and cooperation. The bidding system is running fine, allowing builders to input proposals and clients to analyze them without delay. Navigating is pretty straightforward for both user types because of the intuitive interface.

5.2 General User Guide

5.2.1 Accessing the Platform

For Customers: Sign in to your account, post a project with descriptions, and go through bids made by the contractors. You can accept or reject any offer depending on what pleases you.

For Builders: Access the site, update their profile with a portfolio of completed projects, browse client-posted projects, and post bids. They can further track projects they have been awarded.

5.2.2 Table for Accessing the Platform

Table 5.1 : Customer and Builder Functionality

User Type	Functionality
Customers	<ol style="list-style-type: none"> 1. Sign in to Account: Login to an existing account to access project posting features. 2. Post a Project: Customers can post projects, including descriptions and details. 3. Browse Bids: Customers can view bids made by builders on their posted projects. 4. Accept or Reject Bids: Customers can accept or reject builder bids based on their preferences and project requirements.
Builders	<ol style="list-style-type: none"> 1. Sign in to Account: Builders can log into the system using their existing credentials. 2. Update Profile: Builders can update their profile with a portfolio showcasing their previous projects. 3. Browse Client-Posted Projects: Builders can view various projects posted by clients. 4. Post Bids: Builders can post bids for client projects, including pricing and project proposals. 5. Track Awarded Projects: Builders can track the projects they have been awarded by clients.

Table 5.2 : Detailed Representation

Feature	Customers	Builders
Sign In to Account	<input checked="" type="checkbox"/> Log in to access project posting features.	<input checked="" type="checkbox"/> Log in to update profile, bid on projects, and track projects.
Post a Project	<input checked="" type="checkbox"/> Can post projects with descriptions and requirements.	<input checked="" type="checkbox"/> Not applicable.
Browse Bids	<input checked="" type="checkbox"/> View bids made by contractors for their projects.	<input checked="" type="checkbox"/> Not applicable.
Accept or Reject Bids	<input checked="" type="checkbox"/> Accept or reject bids from builders based on preferences.	<input checked="" type="checkbox"/> Not applicable.
Update Profile	<input checked="" type="checkbox"/> Not applicable.	<input checked="" type="checkbox"/> Can update profile with portfolio and project details.
Browse Client-Posted Projects	<input checked="" type="checkbox"/> Not applicable.	<input checked="" type="checkbox"/> Can browse client projects and bid on them.
Post Bids	<input checked="" type="checkbox"/> Not applicable.	<input checked="" type="checkbox"/> Can post bids for client projects, including pricing.
Track Awarded Projects	<input checked="" type="checkbox"/> Not applicable.	<input checked="" type="checkbox"/> Track the status of projects awarded by clients.

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

6.1 Summary of Achievements

CityCraft addresses the major inefficiencies in the construction industry by giving a centralized platform for both clients and builders to find each other. It streamlines the bidding process, gives transparency, and simplifies the tracking of projects, therefore providing an innovative solution to problems that have been long known.

6.2 Future Work

To enhance functionality and extend reach, we suggest:

AI Recommendations: Implementing a system to suggest the most appropriate builders for a project based on past performance and client reviews.

International Expansion: Enabling builders and clients from different regions to use the platform, therefore encouraging global collaboration.

6.3 Lessons Learned

6.3.1 Challenges and Solutions

Challenge: Up-to-date construction material prices proved to be a major problem.

Solution: We implemented an automated system that fetches the real-time price updates from the local market databases, ensuring the accuracy and relevance.

Through overcoming these challenges and through user feedback, CityCraft successfully delivers a robust and user-centric platform.

LIST OF FIGURES

UML DIAGRAMS

Figure:1.1 USE CASE DIAGRAM

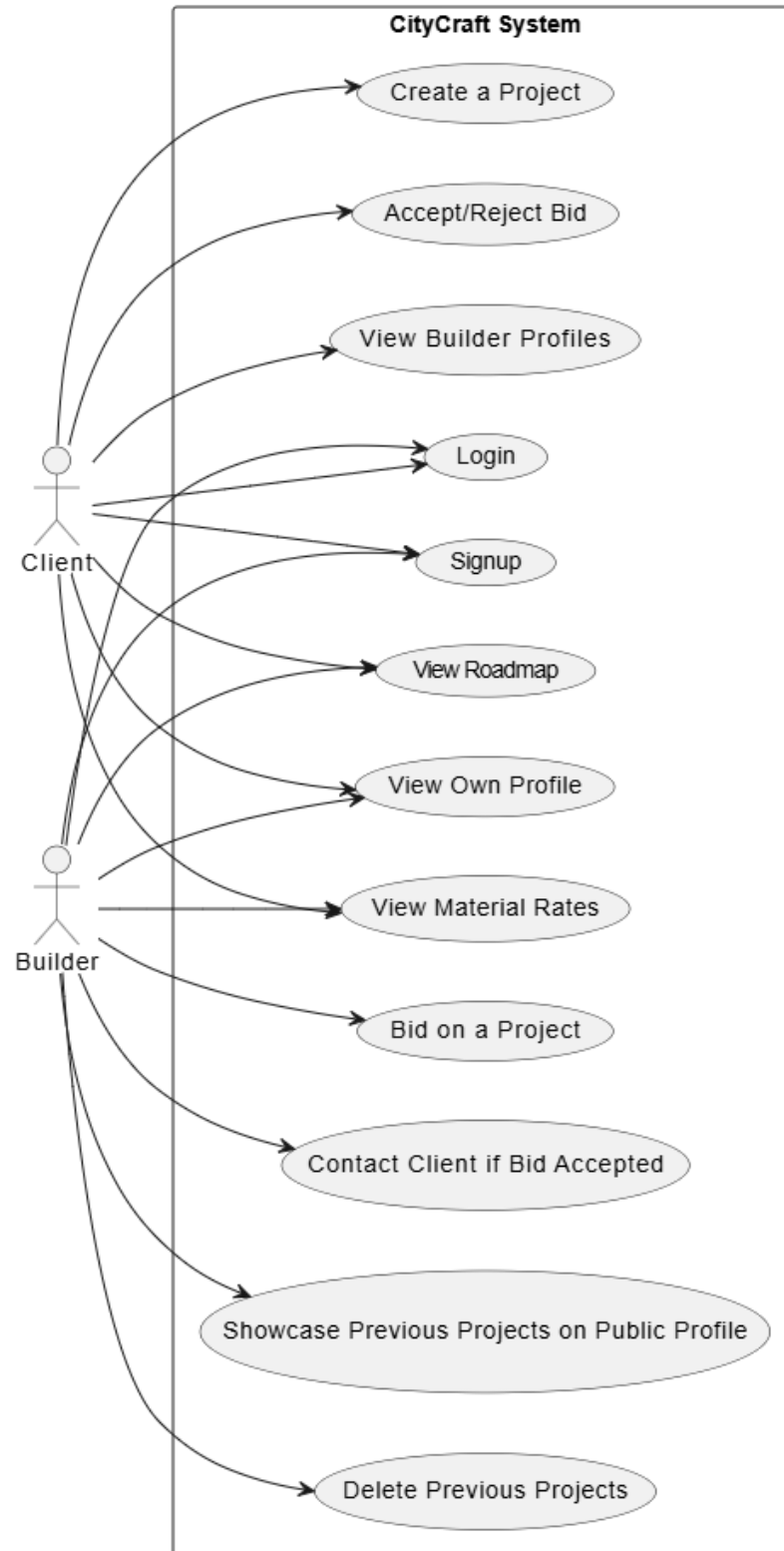


Figure:1.2 CLASS DIAGRAM

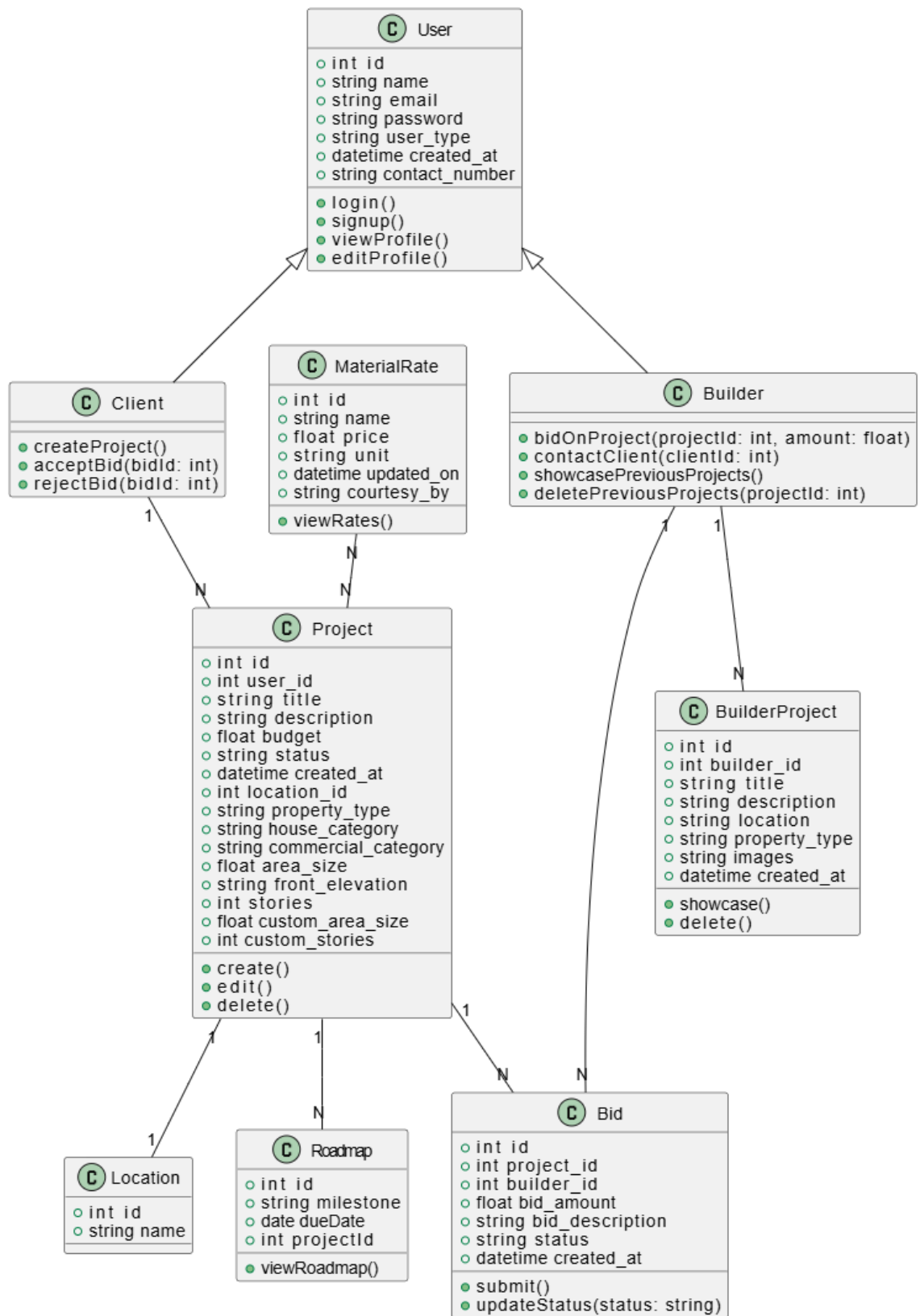
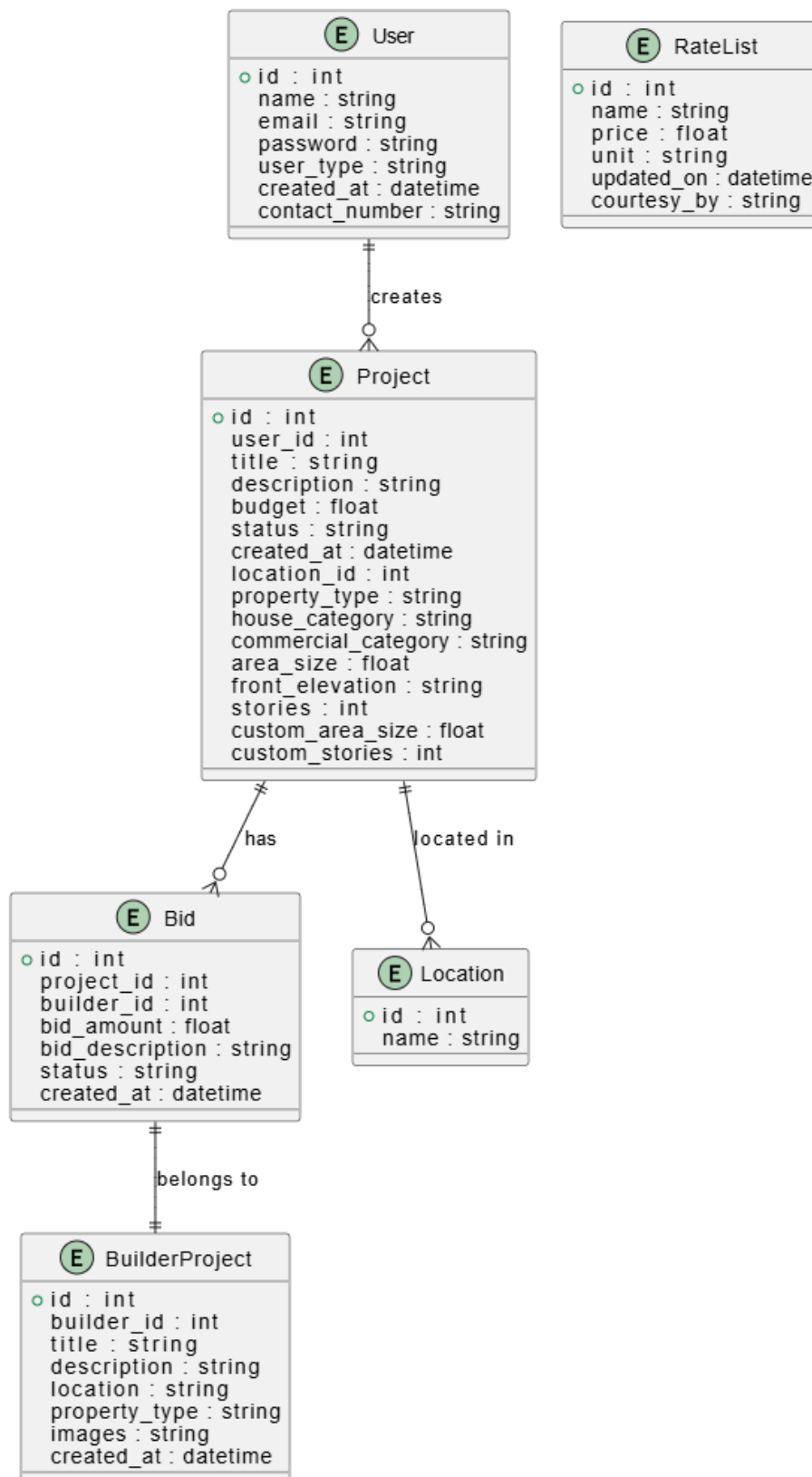


Figure:1.3 ERD DIAGRAM



Explanation of the ERD:

Entities:

- **User:** Represents the users of the system (clients or builders). Attributes include id, name, email, password, user_type, created_at, and contact_number.
- **Project:** Represents projects created by clients. Attributes include id, user_id (foreign key to User), title, description, budget, status, created_at, and various property attributes like location_id, property_type, etc.
- **Bid:** Represents bids placed by builders on projects. Attributes include id, project_id (foreign key to Project), builder_id (foreign key to Builder), bid_amount, bid_description, status, and created_at.
- **BuilderProject:** Represents a builder's portfolio or completed projects. Attributes include id, builder_id, title, description, location, property_type, images, and created_at.
- **RateList:** Stores material rates used for projects. Attributes include id, name, price, unit, updated_on, and courtesy_by.
- **Location:** Represents geographical locations of projects. Attributes include id and name.

Relationships:

- **User to Project:** A user (client) can create many projects. This is a one-to-many relationship (represented as User ||--o{ Project).
- **Project to Bid:** A project can have many bids placed on it. This is also a one-to-many relationship (Project ||--o{ Bid).
- **Bid to BuilderProject:** A bid belongs to a builder, and once accepted, the builder may showcase the project in their portfolio. This is a many-to-one relationship (Bid ||--|| BuilderProject).
- **Project to Location:** A project is located in a specific location, and each project has a location (Project ||--o{ Location).

Figure:1.4 USER LOGIN SEQUENCE DIAGRAM

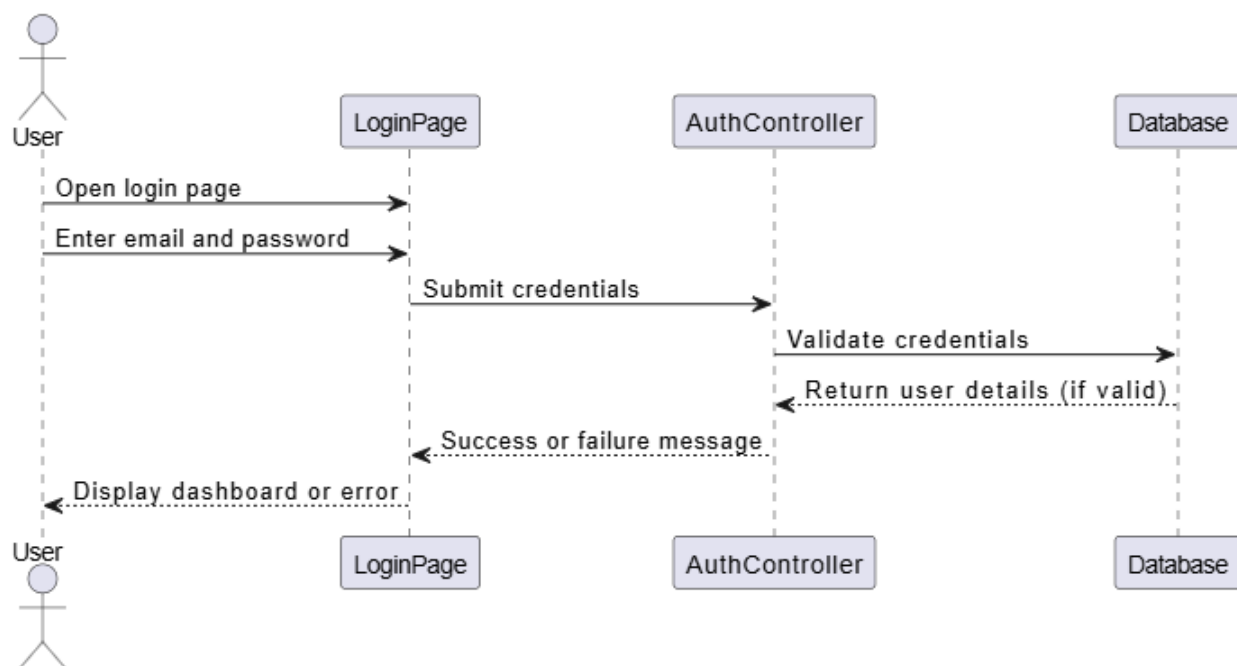


Figure:1.5 SIGNUP SEQUENCE DIAGRAM

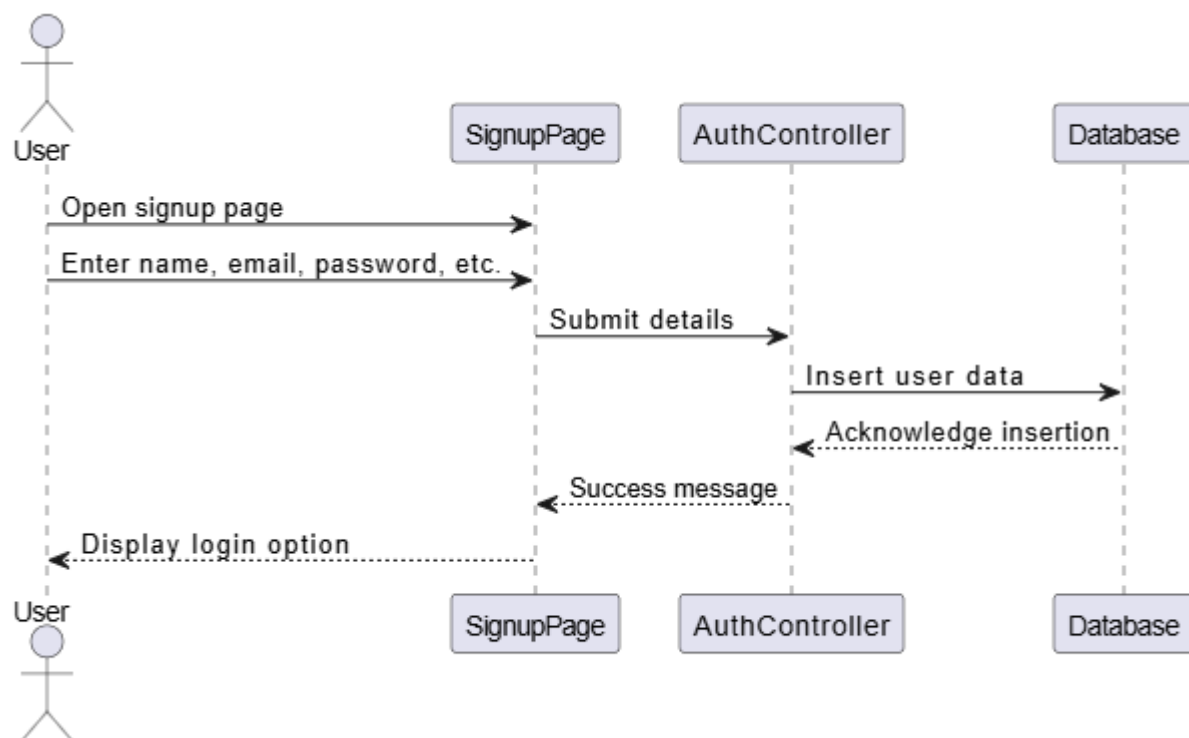


Figure:1.6 CREATE A PROJECT (CLIENT) SEQUENCE DIAGRAM

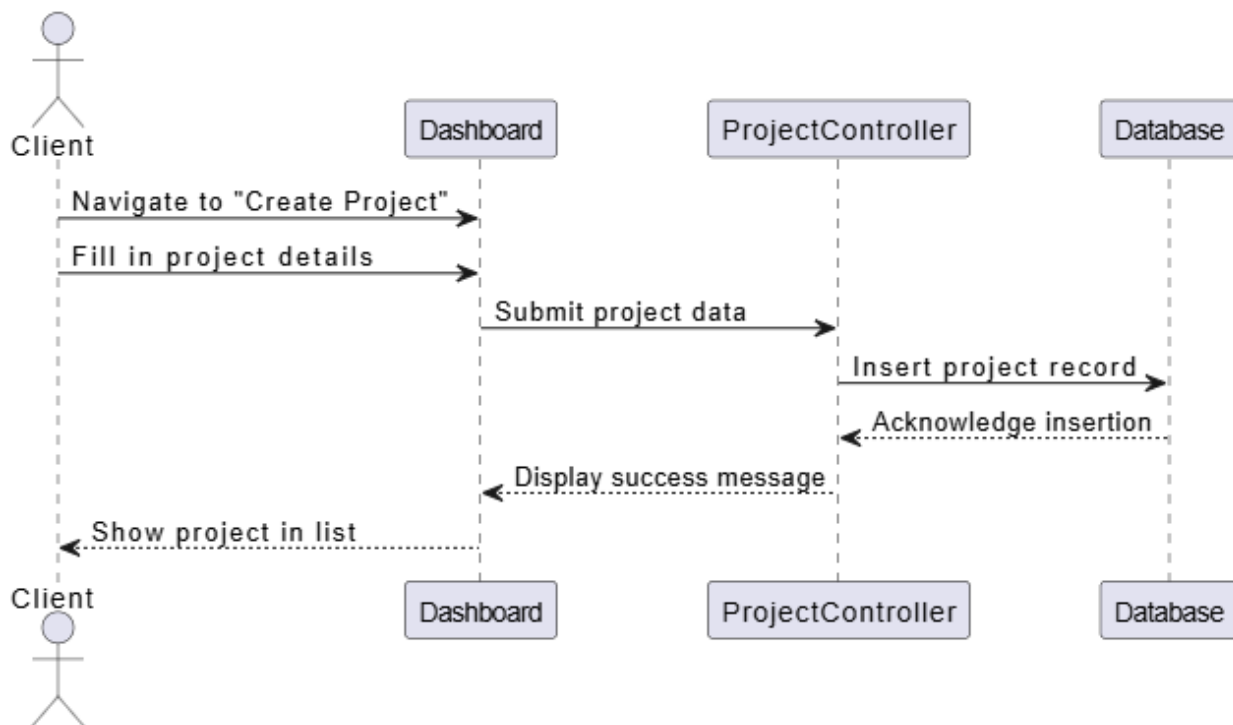


Figure:1.7 BID ON A PROJECT (BUILDER) SEQUENCE DIAGRAM

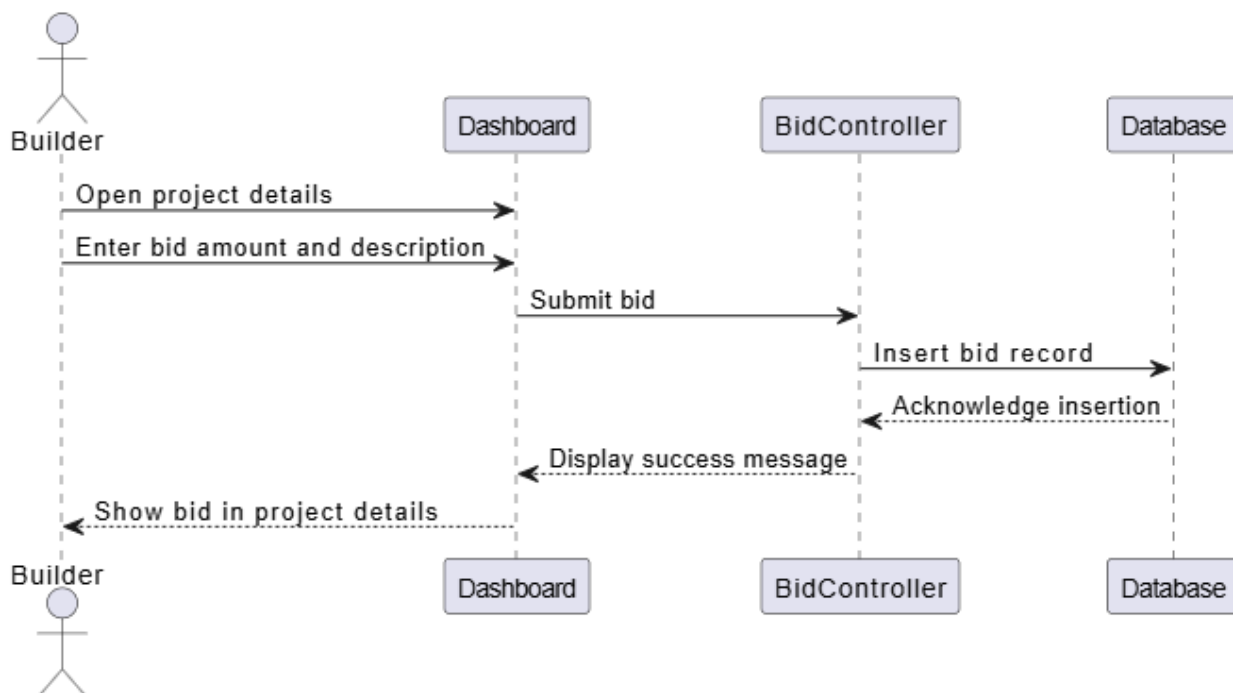


Figure:1.8 ACCEPT/REJECT BID (CLIENT) SEQUENCE DIAGRAM

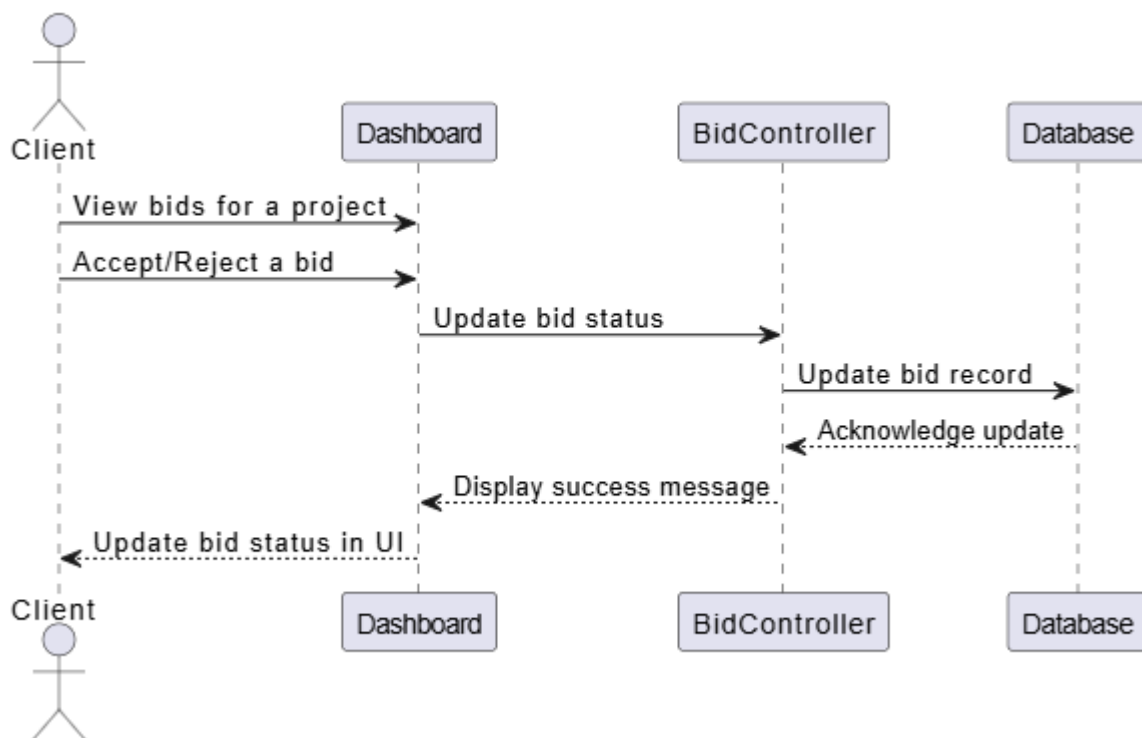


Figure:1.9 VIEW MATERIAL RATES SEQUENCE DIAGRAM

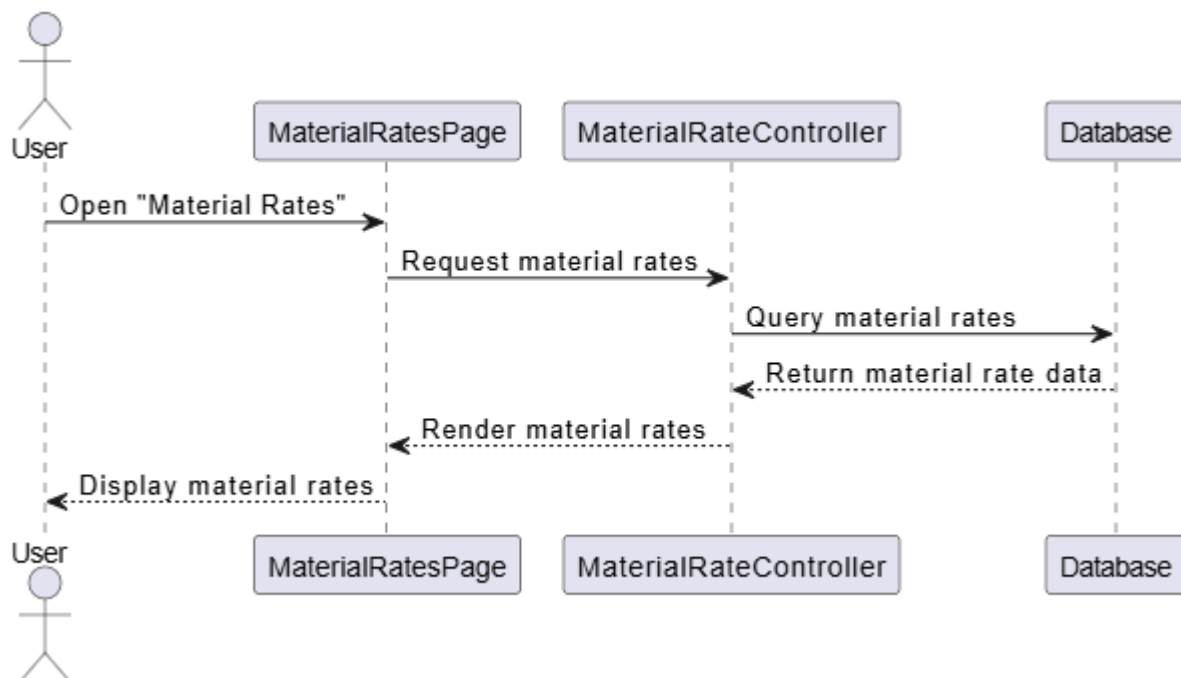
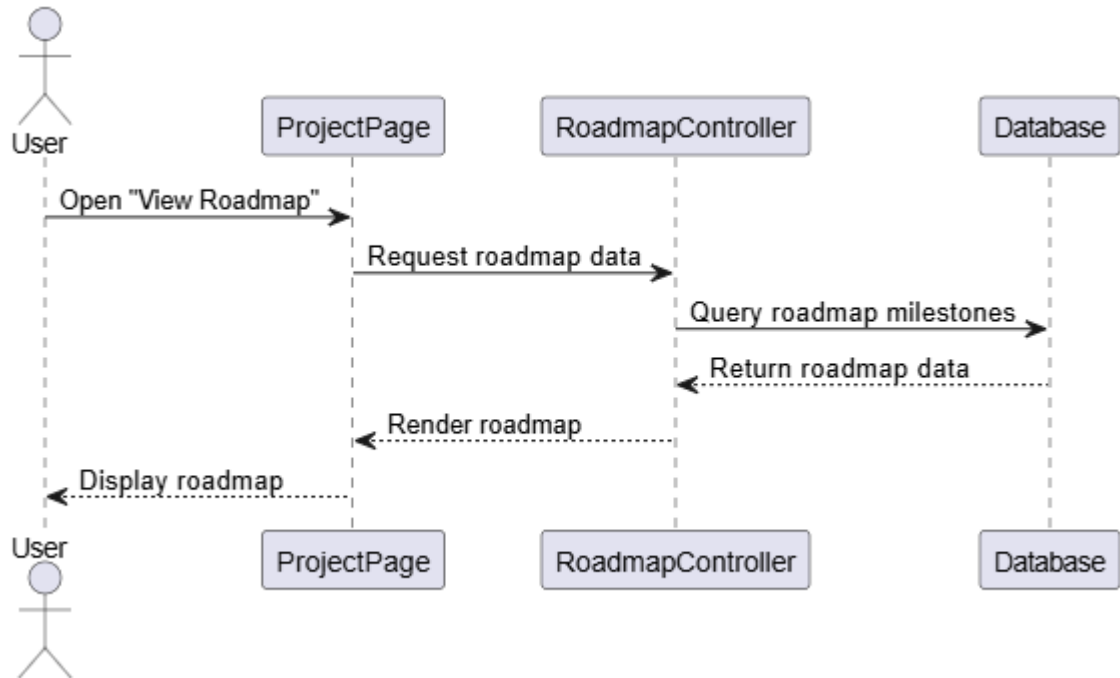
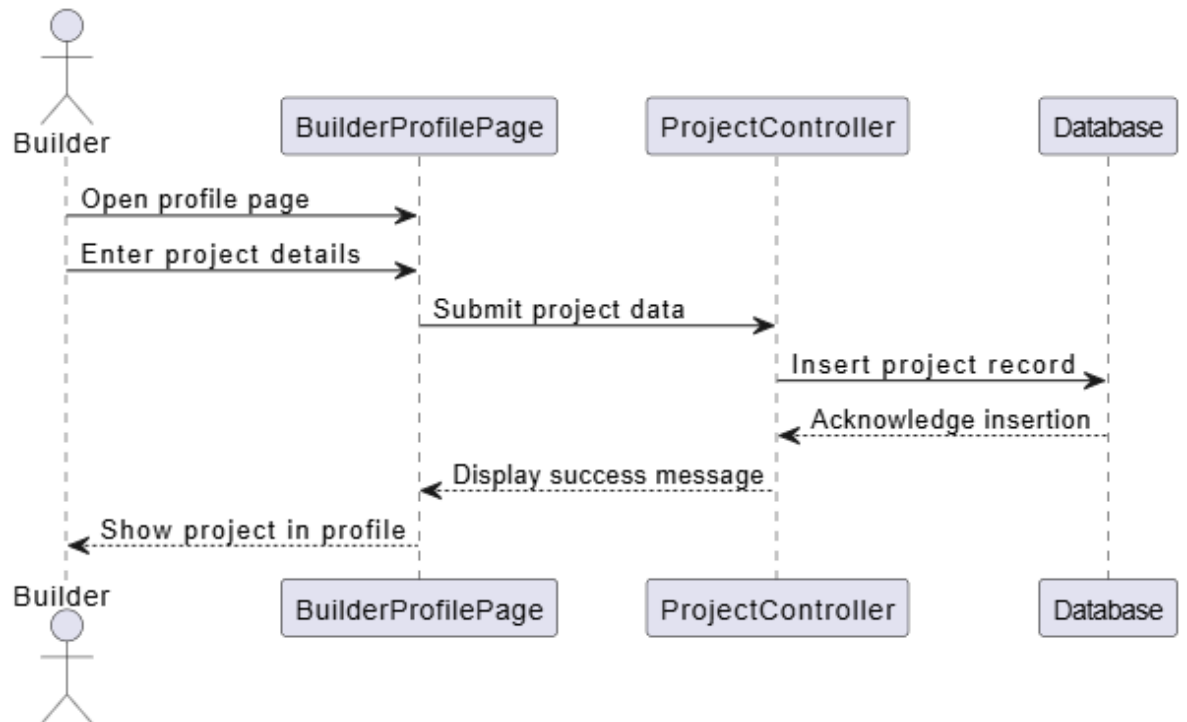
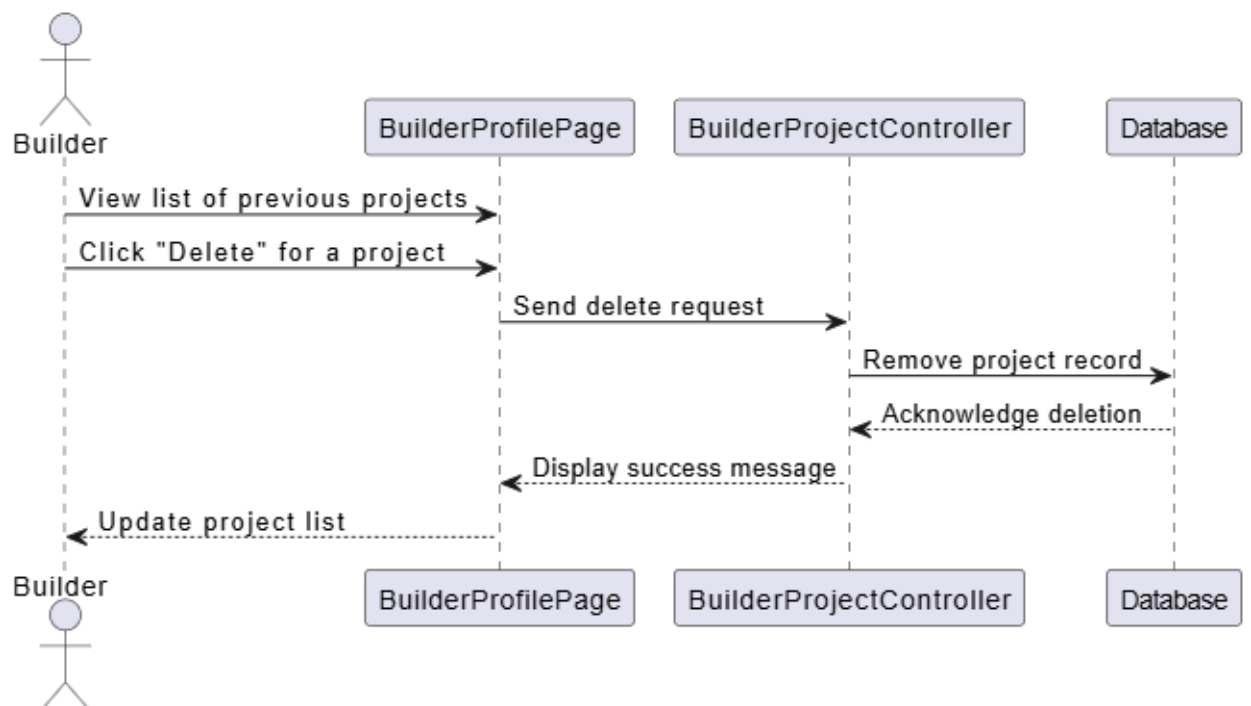


Figure:1.10 VIEW ROADMAP SEQUENCE DIAGRAM

**Figure:1.11 SHOWCASE PREVIOUS PROJECT (BUILDER)
SEQUENCE DIAGRAM**



**Figure:1.12 DELETE PREVIOUS PROJECTS (BUILDER)
SEQUENCE DIAGRAM**



ACTIVITY DIAGRAMS

Figure:1.13 USER LOGIN

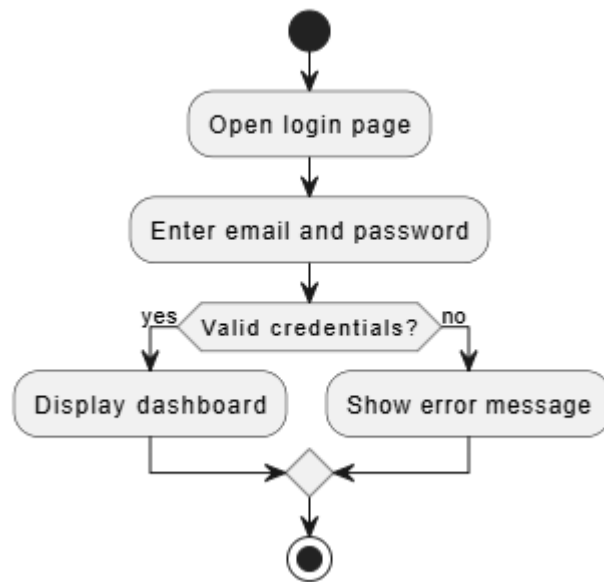


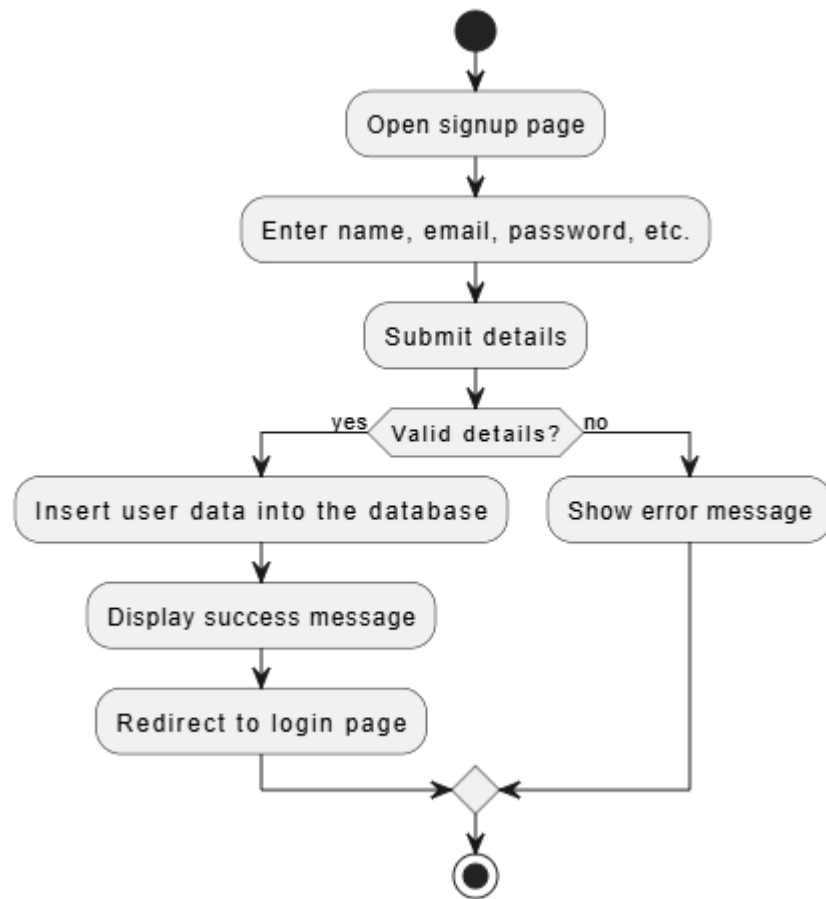
Figure:1.14 SIGNUP

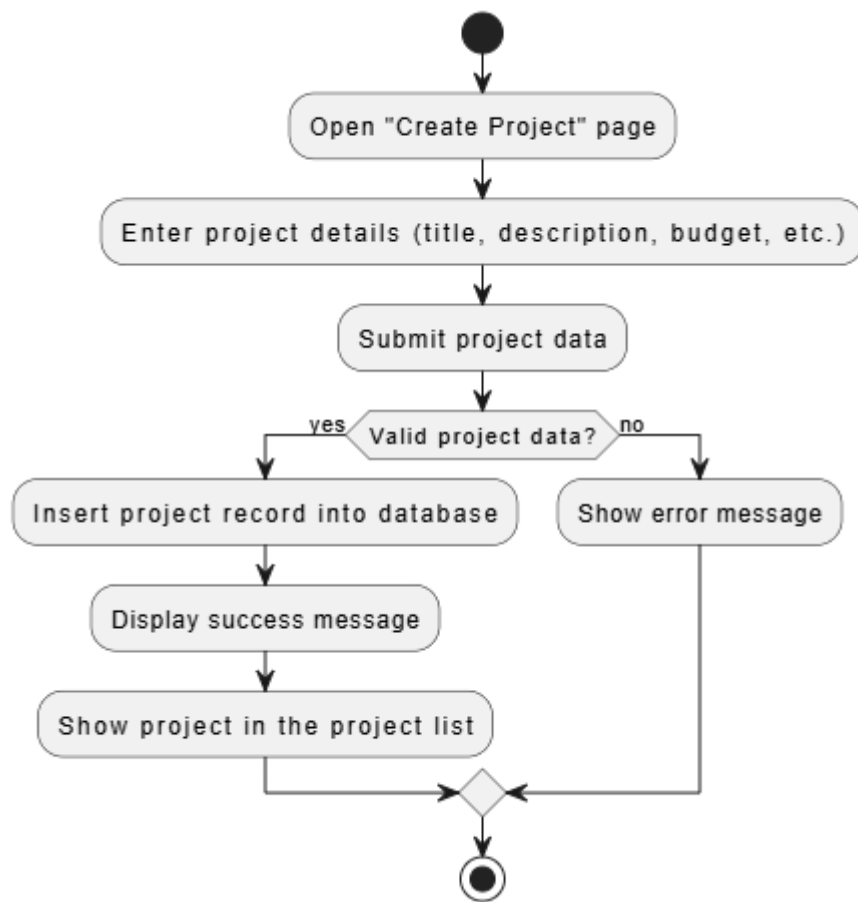
Figure:1.15 CREATE A PROJECT

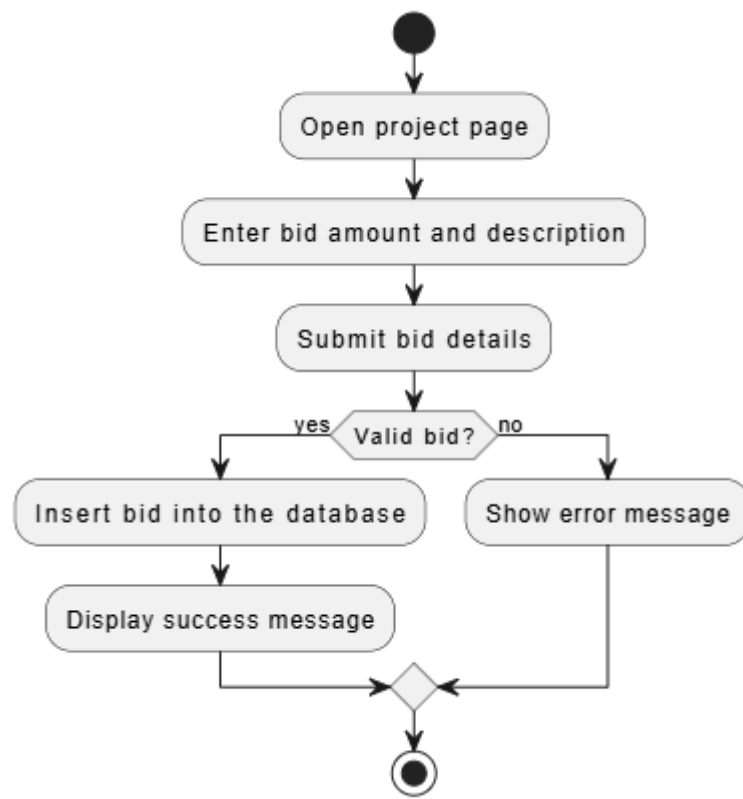
Figure:1.16 BID ON A PROJECT

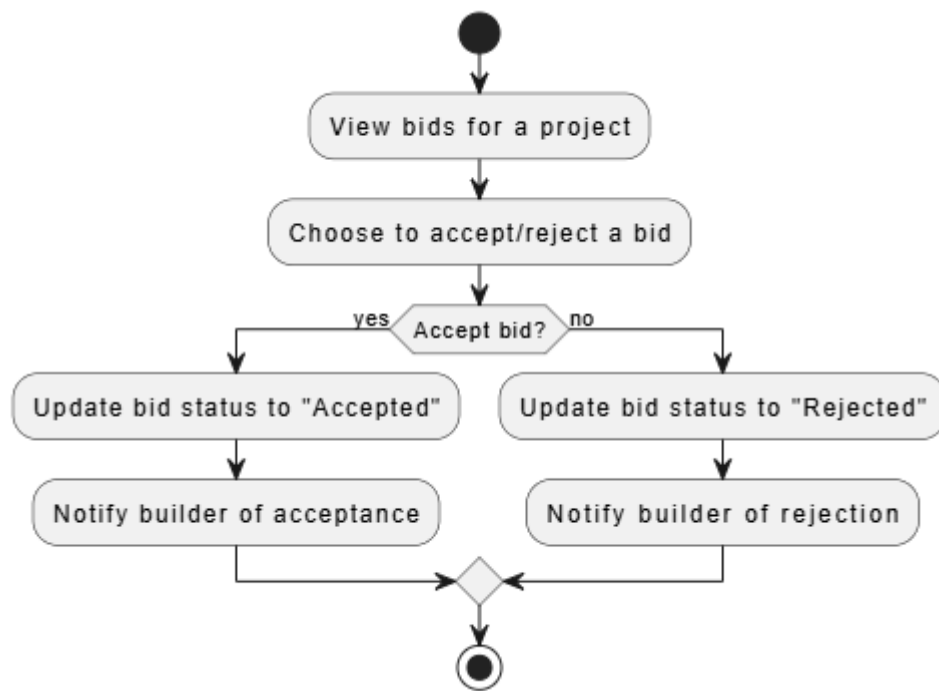
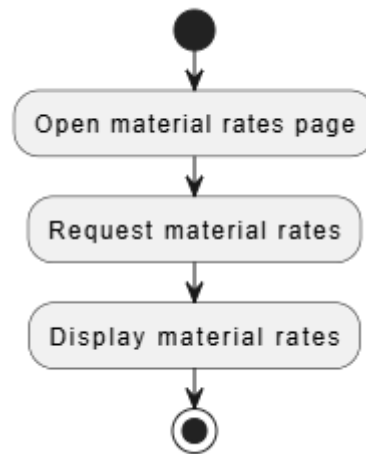
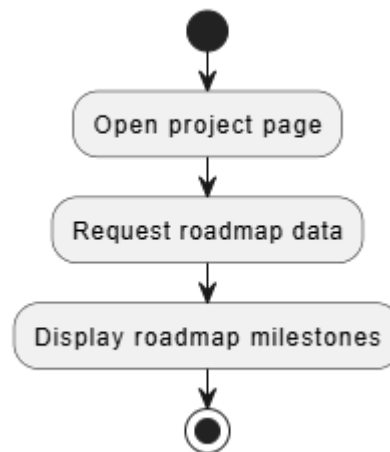
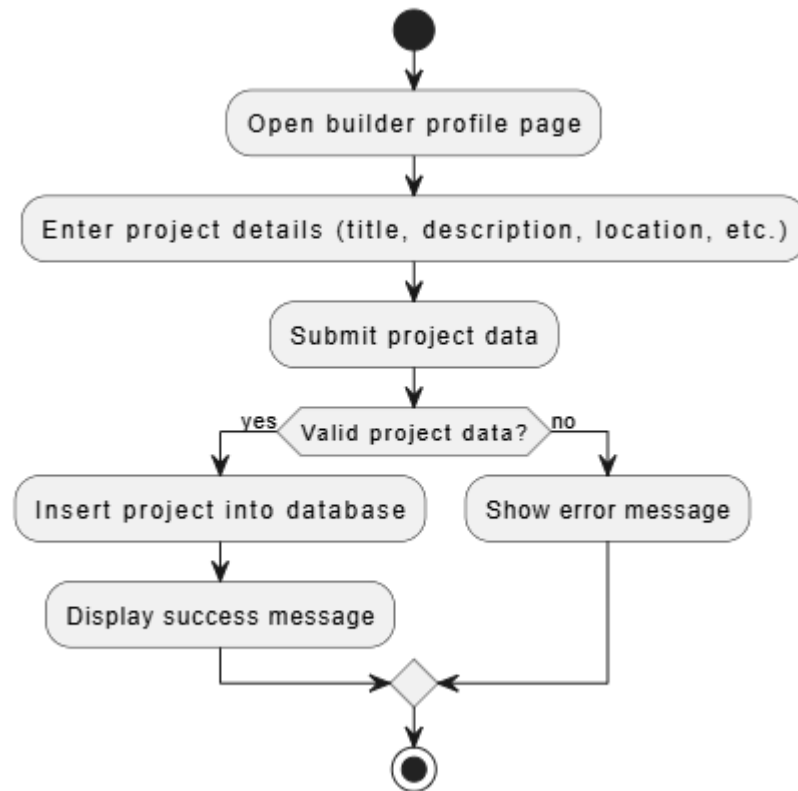
Figure:1.17 ACCEPT/REJECT BID

Figure:1.18 VIEW MATERIAL**Figure:1.19 VIEW ROADMAP**

**Figure:1.20 SHOWCASE
PREVIOUS PROJECTS (BUILDER)**



**Figure:1.21 DELETE PREVIOUS PROJECTS
(BUILDERS)**

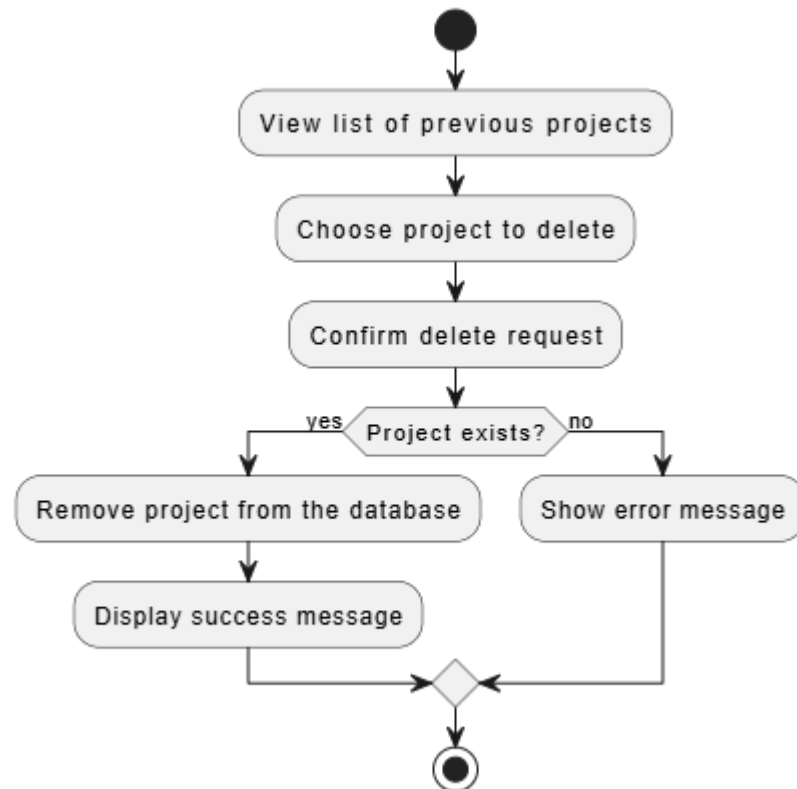
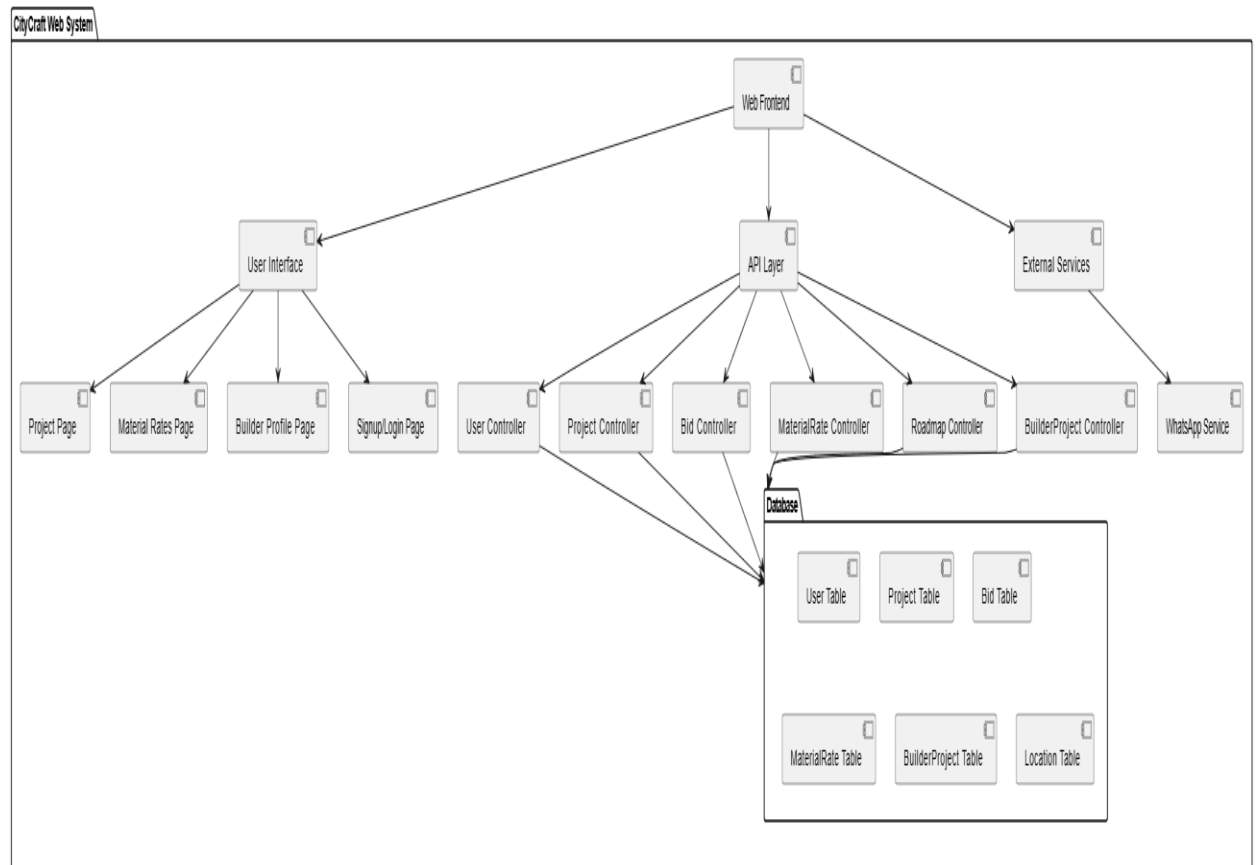


Figure:1.22 COMPONENT DIAGRAM OF CITYCRAFT

Explanation of the Components:

Web Frontend:

User Interface: Represents the user-facing components like the login page, project page, material rates page, and builder profile page.

API Layer: Handles communication between the frontend and backend. It communicates with controllers for user, project, bid, material rates, roadmap, and builder project.

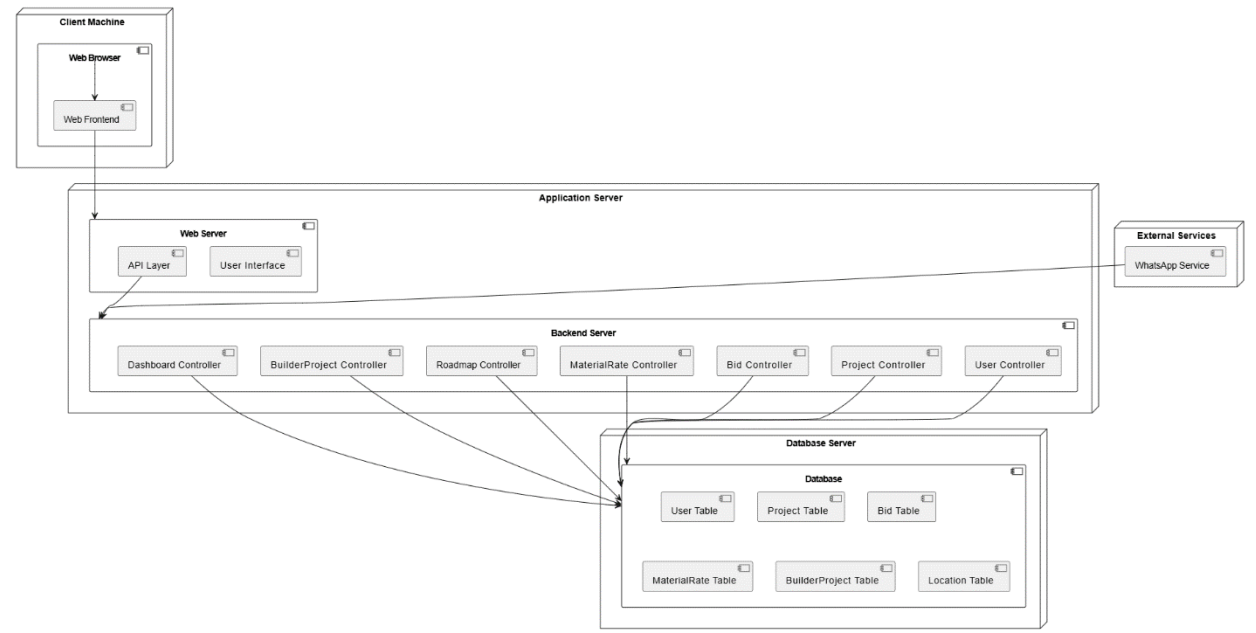
Controllers:

- **User Controller:** Handles user-related actions (login, signup, profile).
- **Project Controller:** Manages project creation, updates, and viewing.
- **Bid Controller:** Manages the bidding process.
- **MaterialRate Controller:** Handles material rates.
- **Roadmap Controller:** Manages construction roadmap milestones.
- **BuilderProject Controller:** Manages builder's projects.

Database:

- **User Table:** Stores user information.
- **Project Table:** Stores project details.
- **Bid Table:** Stores bid-related data.
- **MaterialRate Table:** Stores material rates.
- **BuilderProject Table:** Stores builder-specific projects.
- **Location Table:** Stores location details.

Figure:1.23 DEPLOYMENT DIAGRAM



Deployment Diagram Structure:

Client Machine:

Web Browser: The client (user or builder) interacts with the system through a web browser. The web browser runs the Web Frontend component, which is responsible for rendering the user interface, handling user interactions, and sending requests to the Web Server.

Application Server:

Web Server:

User Interface: This component manages the display of pages like the login page, project page, material rates, etc. It interacts with the API Layer for data retrieval and actions.

API Layer: Acts as an intermediary between the User Interface and the backend. It handles API requests from the frontend and routes them to the corresponding controllers.

Backend Server:

User Controller: Manages user-related actions like login, signup, and profile management.

Project Controller: Handles actions related to project creation, updates, and viewing.

Bid Controller: Manages the bidding process where builders place bids on client projects.

MaterialRate Controller: Handles material rate data, including viewing and updating material prices.

Roadmap Controller: Manages the construction roadmap and milestones of projects.

BuilderProject Controller: Manages builder-specific projects, such as showcasing previous works and managing active projects.

Dashboard Controller: Manages the client's dashboard, displaying active projects, bids, and project status.

Database Server:

Database: Stores all the persistent data for the system. It includes:

User Table: Stores user information such as name, email, password, user type (client or builder), and contact details.

Project Table: Stores details about the projects like title, description, budget, status, and location.

Bid Table: Stores bid data, including bid amount, description, and status.

MaterialRate Table: Stores material prices and units.

BuilderProject Table: Stores information about builder projects and related media.

Location Table: Stores information about project locations.

Connections:

Web Frontend -> Web Server: The frontend (UI) communicates with the Web Server to request and display data (like project details, bids, material rates).

API Layer -> Backend Server: The API layer routes incoming frontend requests to the backend server, where the logic is processed (for example, fetching data or performing actions like login, creating projects, etc.).

Controllers -> Database: Each controller (e.g., User Controller, Project Controller) interacts with the Database to fetch or store data related to users, projects, bids, etc.

Web Browser -> Web Frontend: The web browser is used by users to interact with the system and display the content delivered by the Web Frontend.

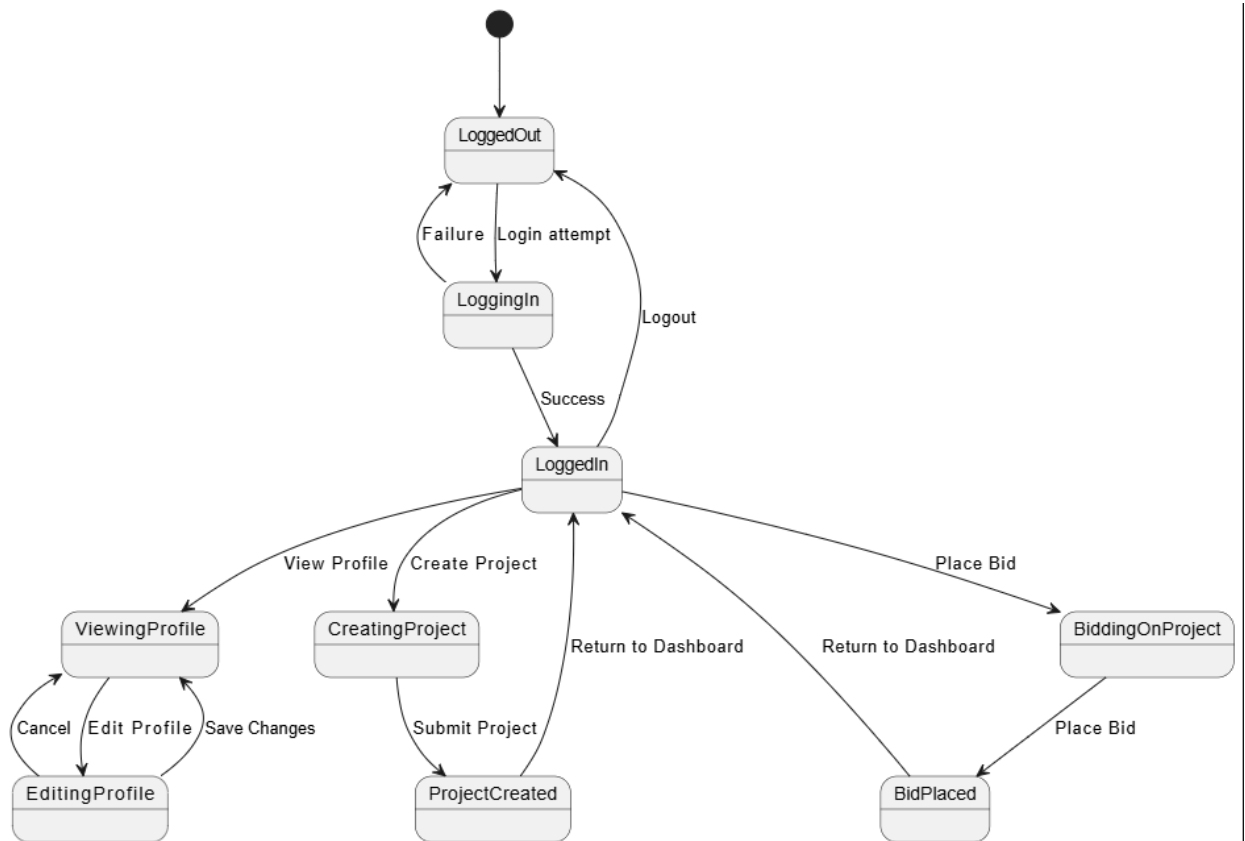
Figure:1.24 USER ACCOUNT STATE DIAGRAM

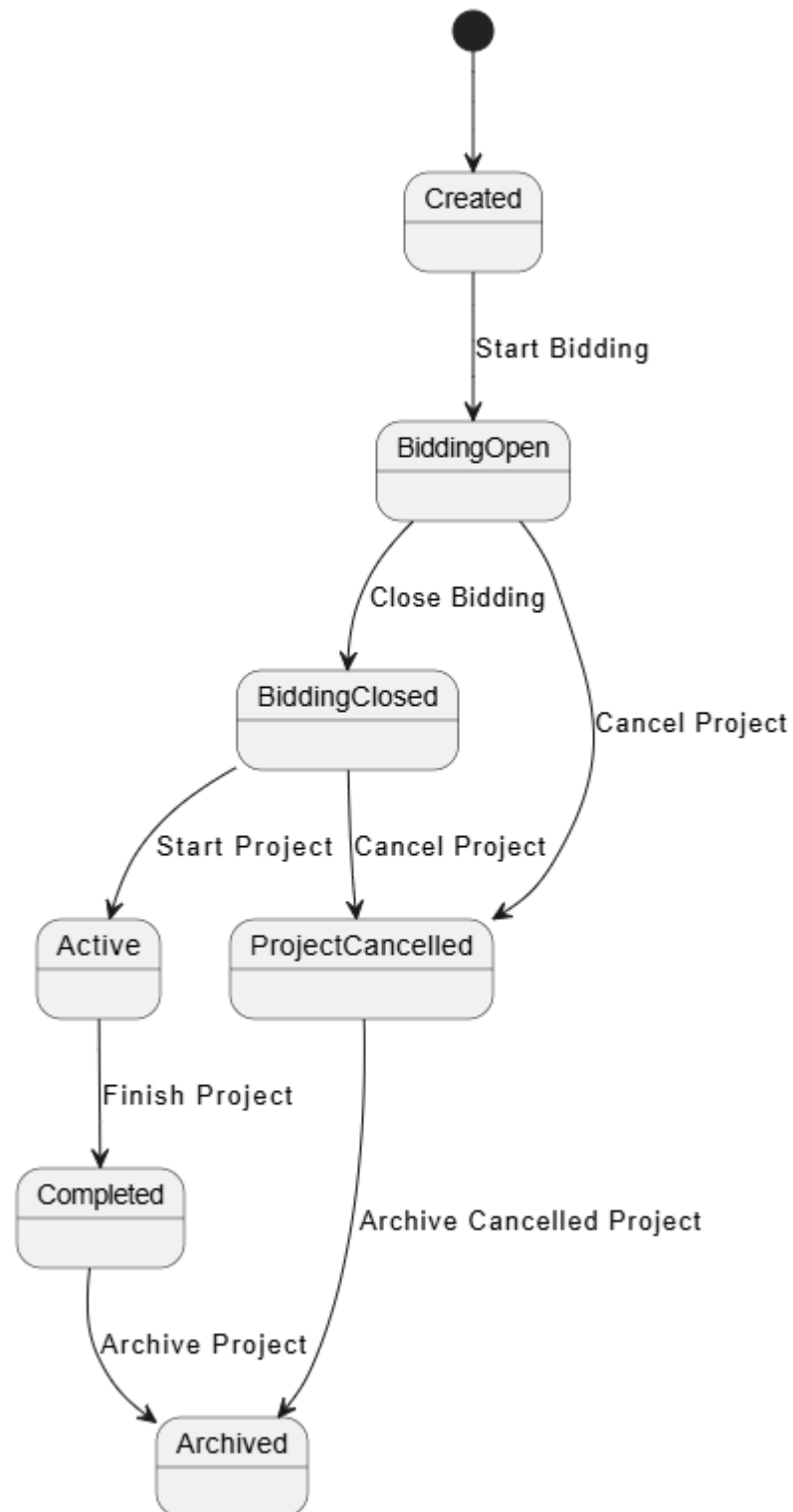
Figure:1.25 PROJECT STATE DIAGRAM

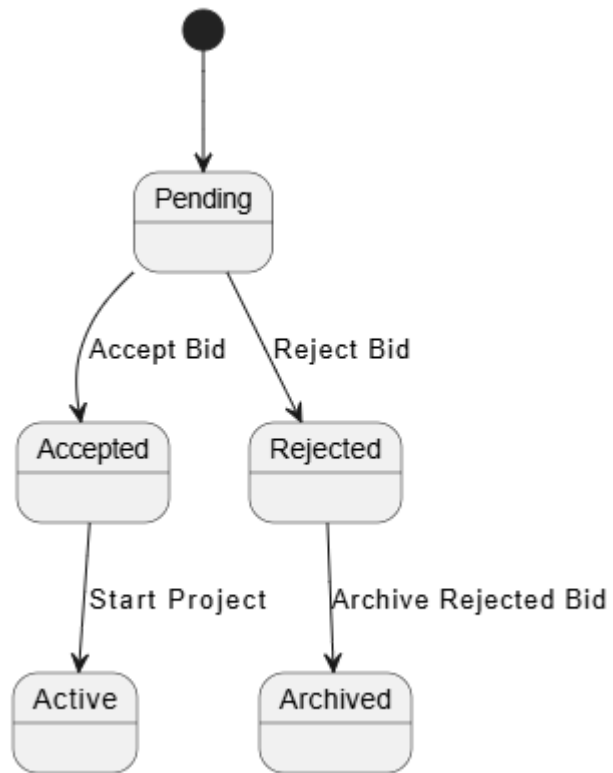
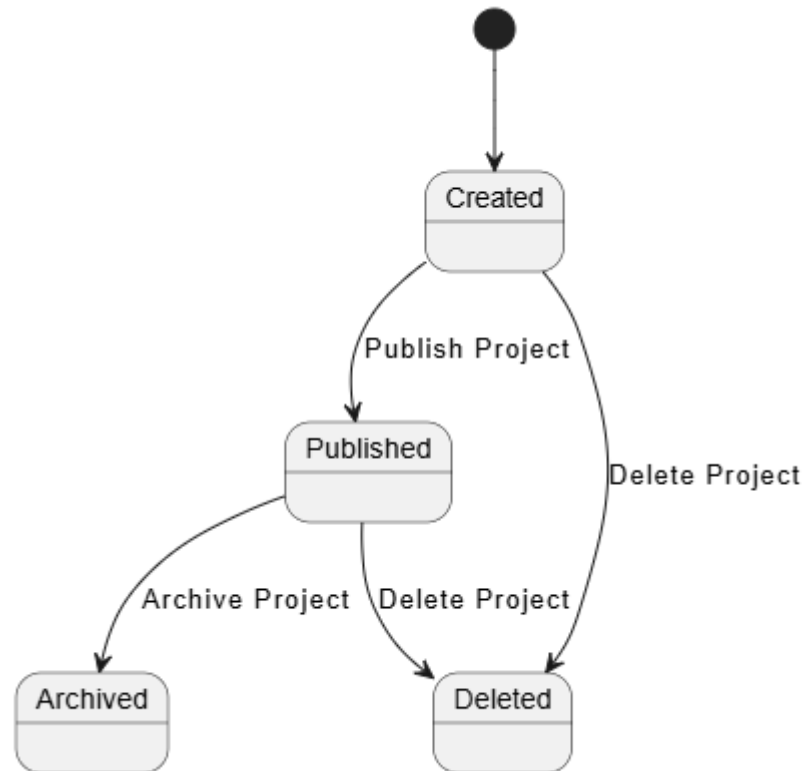
Figure:1.26 BID STATE DIAGRAM

Figure:1.27 BUILDER PROJECT STATE DIAGRAM

User Account State Diagram

- The user starts in the LoggedOut state.
- They transition to LoggingIn and then to LoggedIn on a successful login or stay in LoggedOut if login fails.
- In LoggedIn, they can interact with their profile, create projects, or bid on projects.
- Users can view their profile, create projects, and place bids.
- State transitions like editing profiles or submitting projects occur within these states.

Project State Diagram

- The project starts in the Created state when it is first created.
- After creation, the project can enter the BiddingOpen state, where builders can place bids.
- Once bidding is closed, the project can transition to Active (start of construction) and eventually to Completed after construction finishes.
- If the project is cancelled at any stage, it goes to the ProjectCancelled state and later to Archived.

Bid State Diagram

- The bid starts in the Pending state after the builder places it.
- The client either accepts or rejects the bid, causing the bid to transition to Accepted or Rejected.
- If the bid is accepted, it transitions to Active (the builder starts the project).
- Rejected bids are archived.

REFERENCES

1. **W3Schools PHP Tutorials**
2. **MySQL Database Documentation**
3. **XAMPP Official Website**

APPENDICES

Appendix A: Project Diagrams

System Architecture

- A detailed diagram showing the architecture of CityCraft, including:
 - **Frontend Layer:** HTML, CSS, JavaScript handling the user interface.
 - **Backend Layer:** PHP scripts managing logic, database interaction, and API calls.
 - **Database Layer:** MySQL database storing user data, project details, bids, and payment records.

Use-Case Diagrams

- Diagrams illustrating:
 - User registration and login flows.
 - Builder profile creation and project bidding process.
 - Client project posting and bid selection.

Flowchart for Bidding System

- Visual representation of how clients and builders interact:
 1. Client posts a project.
 2. Builders submit bids.
 3. Client evaluates and selects a bid.

Appendix B: Test Cases

Functional Testing

Test Case ID	Description	Steps	Expected Result	Status
TC-01	User Registration	Fill registration form and submit.	User account is created.	Passed
TC-02	User Login	Enter valid credentials and log in.	User is redirected to dashboard.	Passed
TC-03	Invalid Login Attempt	Enter incorrect credentials and log in.	Error message is displayed.	Passed

Security Testing

Test Case ID	Description	Steps	Expected Result	Status
ST-01	SQL Injection Prevention	Submit SQL query as input in login form.	Input is sanitized; no injection.	Passed
ST-02	Session Management	Test session timeout after inactivity.	User is logged out automatically.	Passed

Appendix C: User Manual

Steps to Use CityCraft

1. **Register as a User:**
 - Click "Register" on the homepage.
 - Fill in your details and submit the form.
2. **Post a Project (Clients):**
 - Log in and navigate to "Post a Project."
 - Enter project details (e.g., budget, timeline).

3. **Submit Bids (Builders):**
 - Log in and view available projects.
 - Submit a detailed bid with cost estimates.
4. **Select a Builder (Clients):**
 - Compare bids and choose the most suitable builder.
5. **Track Progress:**
 - Monitor milestones and updates via the dashboard.
6. **Make Payments:**
 - Use the integrated payment gateway to release funds securely.

Appendix D: Glossary of Terms

Term	Definition
Bidding System	A mechanism where builders propose their cost and timeline for a project posted by a client.
Dashboard	The user interface where clients and builders manage their projects, bids, and profiles.
Escrow	A secure payment system that holds funds until specific project milestones are completed.
Milestones	Specific phases or goals within a construction project, used to track progress.

