BSCS FINAL PROJECT

Requirements Specification

ProLabour: Smart Job Matching Platform for Skilled Laborers

Project Advisor

Asim Raza

Presented by:

Group ID: S25BS030

L1S22BSCS0108	Muhammad Sami Khan
L1F21BSCS1269	Danish Nawaz
L1S22BSCS0106	Muhammad Abubakar

Faculty of Information Technology & Computer Science

University of Central Punjab

Software Requirements Specification

Version <Version 1.0>

ProLabour: Smart Job Matching Platform for Skilled Laborers

Advisor: Asim Raza

Group: S25BS030

Member Name	Primary Responsibility
Muhammad Sami Khan	Full Stack Development, UI/UX Design/QA/Documentation
Danish Nawaz	Frontend Development
Muhammad Abubakar	Backend Development

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

Abstract

Hiring qualified workers is often difficult, particularly in places such as Pakistan, where current platforms restrict worker sign-ups and keep salary standards inflexible. To tackle existing problems, ProLabour designs a platform that promotes openness, flexibility, and fairness in job matching. By using ProLabour, electricians, plumbers, carpenters, as well as other skilled workers, are able to register without barriers and receive pay according to each job they complete, instead of receiving a set salary. Real-time messaging on ProLabour ensures laborers connect straight with customers, resulting in better and more transparent communication. Geolocation services for discovering local openings and a rating mechanism to recognize reliability are among its features. By harnessing current technological developments, ProLabour seeks to reduce current labor-market gaps and provide a platform that helps workers maintain employment and gives customers more choices from qualified professionals.

1. Introduction and Background

1.1 Product (Problem Statement)

Pakistan's labor market is affected by ineffective laborer-customer matching. Currently used platforms work on centralized bases that limit workers' registration, impose strict payment mechanisms, and ban direct connection between laborers and customers. This results in ineffective utilization of skilled workers, unreliable job opportunity, financial unreliability of laborers, and inconvenience of customers in promptly getting reliable qualified assistance. ProLabour hopes to address these issues by establishing a decentralized platform that links workers directly to customers, facilitating flexible working schedules and equitable, job-related pay.

1.2 Background

In Pakistan, the skilled labor (electricians, plumbers, carpenters, etc.) usually operate through middlemen or try to find anyone who is in need. They remain in jobs with periods of unemployment in between, which induces economic insecurity. The customers find it difficult to engage reliable skilled manpower when the need arises, tending to settle for dodgy recommendations or take whoever comes first instead of the best candidate available.

Current digital platforms in the sector work on centralized models with very limiting characteristics. Platforms such as Maahir and Karsaz limit the number of workers that may register. They offer a set monthly wage, which may result in unfair compensation and decreased motivation for workers who finish more jobs than others. The platforms do not have the level of flexibility and openness required to develop a highly effective labor marketplace.

1.3 Scope

ProLabour shall be an end-to-end mobile platform encompassing:

- User registration and profile creation for laborers and customers.
- Job posting and search functionality with option for filtering.
- Direct messaging channels between laborers and customers.
- Network Manager to create connections among users.
- Rating and review mechanism for quality control.
- Worker availability management through online/offline status.
- Confirmation of job completion and payment handling.
- Real-time notifications for job status and messages.
- Geolocation services for job matching based on proximity.

- The platform will be offered as a mobile app targeting first the Pakistani market with scope for expansion in the future.

1.4 Objective(s)/Aim(s)/Target(s)

The most important goals of ProLabour are:

- **Unlimited registration of skilled laborers:** Permit an unlimited number of skilled laborers to be registered on the platform.
- **Real-time direct interaction:** Enable real-time direct interaction between customers and laborers through an integrated messaging/Network Manager.
- **Rating & review system:** Provide mutual ratings following jobs to develop trust and enhance service quality.
- **Network Manager Feature:** Implement a dedicated component to manage peer-to-peer connections and job coordination among users.
- **Worker availability management:** Allow workers to mark themselves offline when they are not available so that they don't get new requests.
- **Easy-to-use interface:** Give an easy, responsive mobile UI/UX to make the experience better.
- **Pay-per-job reward model:** Introduce a pay-per-job payment system that incentivizes workers per completed job instead of fixed pay.

1.5 Challenges

Building a solution presents a number of technical and design issues:

- Designing and deploying a decentralized system.
- Implementing the Network Manager to provide efficient direct communication amongst users.
- Integrating real-time updates and push notifications to notify users of job and message activity.
- Geolocation services for reliable job matching and navigation.
- Providing solid user authentication and data privacy methods.

- Developing a responsive mobile UI that performs well on a variety of devices.

1.6 Learning Outcomes

1. Mobile application development proficiency.
2. Backend development experience with Node.js and MongoDB.
3. Familiarity with decentralized systems and how to implement them.
4. Real-time data processing knowledge.
5. UI/UX design skills using Figma/Canva.
6. Third-party API integration experience with Google Maps and Firebase.

1.7 Nature of End Product

The final product will be a complete mobile application with user registration, job posting, and direct communication through the Network Manager, ratings and reviews, and workers' availability management.

1.8 Completeness Criteria

The project will be deemed finished when the following are accomplished:

- Mobile Application Development: 40%
- User Interface/UX Design: 15%
- Decentralized Registration System: 10%
- Rating & Review System: 10%
- Network Manager for Direct Communication: 15%
- Worker Availability Management: 5%
- Database Integration: 5%

1.9 Business Goals

1. To create a scalable platform that can handle an unlimited amount of skilled laborers.
2. To increase the revenue-generating potential of workers using a pay-per-job system.
3. To enhance customer satisfaction through easy access to skilled laborers.
4. To enable trust and transparency in the employment market via reviews and ratings.
5. To establish an elastic working environment for workers by enabling them to control their availability.

1.10 Related Work/ Literature Survey/ Literature Review

Current platforms such as Maahir and Karsaz are based on centralized models, with a limited number of laborers that can register and job opportunities limited to fixed monthly wages. ProLabour, on the other hand, uses a decentralized model, with unlimited registrations and direct customer-laborer interaction. Features such as the Network Manager for direct connections and offline mode for availability management make ProLabour stand out from current solutions.

1.11 Document Conventions

This SRS is formatted according to standard documents with headings in bold and use-case tables for functional requirements. External systems or components are indicated in italics.

2. Overall Description

2.1 Product Features

- Decentralized registration system for unlimited number of skilled laborers.
- Job posting and search facility for customers.
- Direct communication between customers and laborers through the Network Manager.
- Rating and review facility for both laborers and customers.
- Worker availability management with offline support.
- Real-time notifications for job updates and messages.
- Mobile-friendly interface

2.2 User Classes and Characteristics

The main user categories for ProLabour are:

Laborers/Workers:

- Trade workers like plumbers, electricians, carpenters, and painters.
- Need easy access to employment opportunities and immediate communication with customers.
- Different levels of digital literacy; app interface must be simple and user-friendly.

Customers (Employers):

- Individuals or small companies looking for labor for particular jobs.
- Need to advertise job requirements, search for laborers, and communicate directly with chosen workers.

Each user type accesses the platform in a unique way, with individual access rights customized to their role.

2.3 Operating Environment

The application will run on handheld devices. It will have internet connectivity to provide real-time functionality and possibly use GPS to deliver geolocation services.

2.4 Design and Implementation Constraints

CON-1: The application should be able to run smoothly on standard Android smartphones.
CON-2: Should be compatible with Android 10 and above.
CON-3: All core features such as messaging and job search need network connectivity.
CON-4: Adherence to data privacy regulations to secure user data (e.g., user permission for location tracking).
CON-5: Limited backend resources can impact real-time notification and data processing speed.
CON-6: Should be able to manage multiple users concurrently without causing system crashes or data loss.

2.5 Assumptions and Dependencies

Assumptions:

- Users possess smartphones with internet connectivity.
- Laborers are ready to register and utilize the platform.
- Customers are ready to employ laborers via the app.
- Direct interaction between laborers and customers will result in more effective outcomes.

Dependencies:

- Third-party services such as Firebase and Google Maps API.
- Presence of skilled laborers in the target market.

3. Functional Requirements

Use Case Diagram:

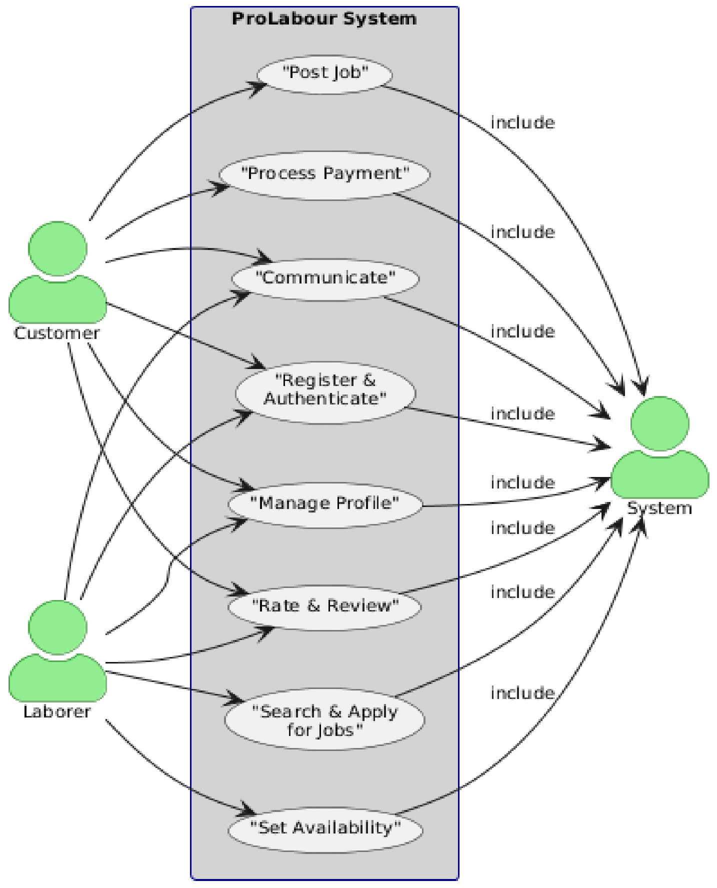


Figure 1: Use Case Diagram

3.1 User Registration and Authentication

Identifier	UC-1	
Purpose	To allow new users to create accounts and authenticate existing users	
Priority	High	
Pre-conditions	User has downloaded the application and has internet connectivity	
Post-conditions	User has a verified account and can access platform features	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	User selects registration option and chooses user type (Worker or Customer)	System displays appropriate registration form
2	User enters personal information (name, phone number, email, password)	System verifies credentials.
3	Laborer user enters skills, experience, and service area	System stores this information
4	User submits registration form	System verifies phone number through OTP and creates account
6		System creates user profile and directs to home screen
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	If credentials are incorrect	The system displays an error message.
2	Enters weak password.	Prompts for stronger password
3	User selects login instead of registration	System displays login screen
4	User enters credentials	System authenticates and logs in user

Table 1: UC-1

3.2 Profile Management

Identifier	UC-2	
Purpose	To allow users to create and manage their profiles	
Priority	High	
Pre-conditions	User is registered and logged in	
Post-conditions	User profile is created or updated	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	User navigates to profile section	System displays current profile information
2	User selects edit profile option	System presents editable profile form
3	User modifies personal information	System stores this information
4	Laborer uploads portfolio images and descriptions	System stores images and updates portfolio
6	Laborer enters or updates skills and experience	System updates skills information
7	User saves changes	System updates profile and displays confirmation
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	User attempts to leave required fields empty	The system displays an error message.

Table 2: UC-2

3.3 Job Posting

Identifier	UC-3	
Purpose	To allow customers to create and manage job postings	
Priority	High	
Pre-conditions	Customer is registered, logged in, and verified	
Post-conditions	Job is posted and visible to relevant laborers	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Customer selects "Post New Job" option	System display option
2	Customer enters job title and detailed description	System validates input
3	Customer selects job category and required skills	System stores this information
4	Customer specifies location and service timeframe	System validates location and timeframe
6	Customer sets budget range	System records budget information
7	Customer submits job posting	System creates job listing and notifies nearby qualified laborers
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	Submission fails due to validation issues	System highlights errors and requests corrections
2	Customer cancels job posting	System discards draft and returns to home screen

Table 3: UC-3

3.4 Job Search and Application

Identifier	UC-4	
Purpose	To allow laborers to find and apply for suitable jobs	
Priority	High	
Pre-conditions	Laborer is registered, logged in, and has completed profile	
Post-conditions	Laborer applies for job	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Laborer navigates to job search screen	System displays available jobs
3	Laborer selects a job to view details	System displays complete job information
4	Laborer chooses to apply for the job	System notifies customer of the application
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	Laborer finds job unsuitable after viewing details	System allows return to search results

Table 4: UC-4

3.5 Direct Communication

Identifier	UC-5	
Purpose	To facilitate direct interaction between laborers and customers	
Priority	High	
Pre-conditions	Connection established between customer and laborer for a specific job	
Post-conditions	Communication occurs and job details are clarified	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	User accesses messaging section from job details	System opens Network Manager communication channel
2	User types and sends message	System delivers message in real-time to recipient
3	User receives reply notification	System displays incoming message
4	Users discuss job details, pricing, and logistics	System maintains message history and read status
5	User sends images or location pins for clarification	System processes and delivers media content
6		
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	Users cannot reach agreement	System provides option to close communication channel

Table 5: UC-5

3.6 Rating and Review System

Identifier	UC-6	
Purpose	To build trust through feedback after job completion	
Priority	Medium	
Pre-conditions	Job marked as completed by both parties	
Post-conditions	Rating and review recorded and visible on profiles	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	User (Customer&Worker) receives notification to rate completed job	System displays rating and review form
2	User selects star rating (1-5)	System records rating
3	User submits rating and review	System processes and publishes feedback
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	User skips written review, provides only star rating	System accepts rating without review
2	User ignores rating request	System marks job as completed without rating

Table 6: UC-6

3.7 Availability Management

Identifier	UC-7	
Purpose	To allow laborers to control their work availability	
Priority	Medium	
Pre-conditions	Laborer is registered and logged in	
Post-conditions	Laborer's availability status is updated	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Laborer accesses availability controls	System displays current status and options
2	Worker toggles status between online and offline	System updates availability
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	Worker continued to work as usual	System continue to provide service

Table 7: UC-7

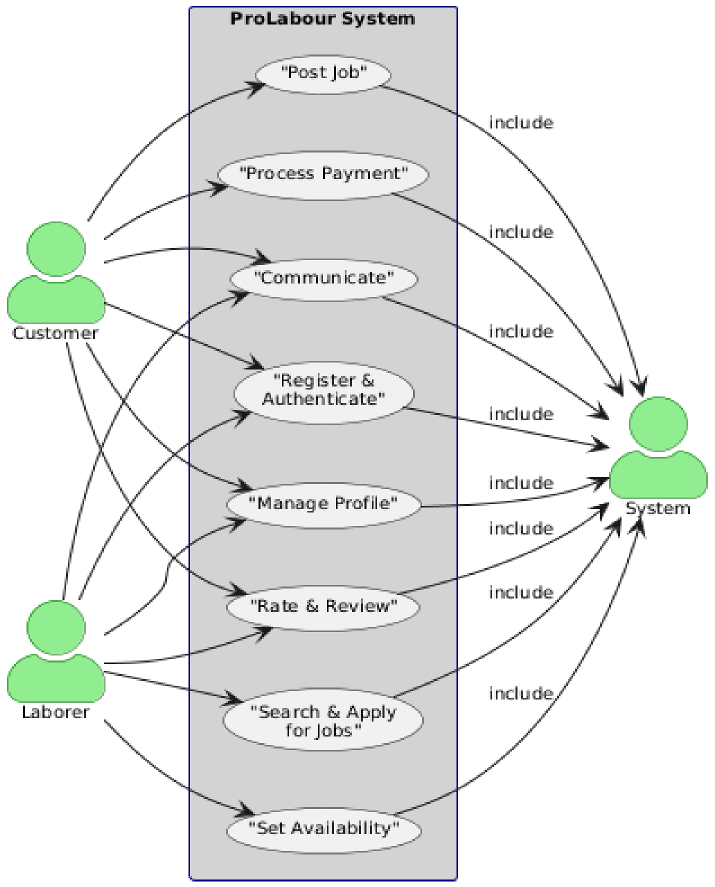
3.8 Payment Processing

Identifier	UC-8	
Purpose	To facilitate payment for completed jobs	
Priority	High	
Pre-conditions	Job marked as completed by laborer	
Post-conditions	Payment processed and recorded	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Customer receives job completion notification	System displays job details and payment options
2	Customer confirms job completion	System validates confirmation
3	Customer makes payment	System saves the information
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	No job was created	

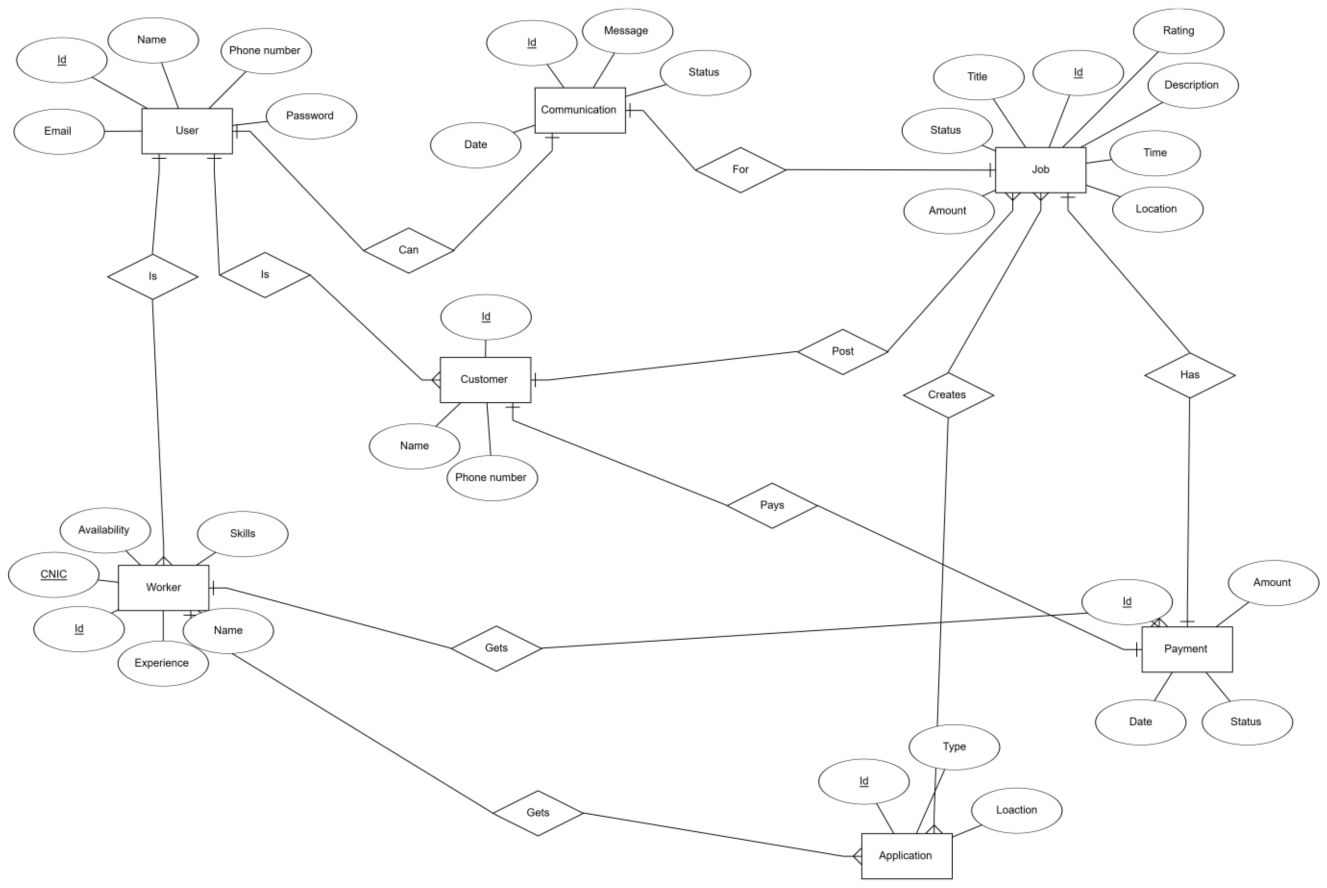
Table 7: UC-8

3.9 Requirements Analysis and Modeling

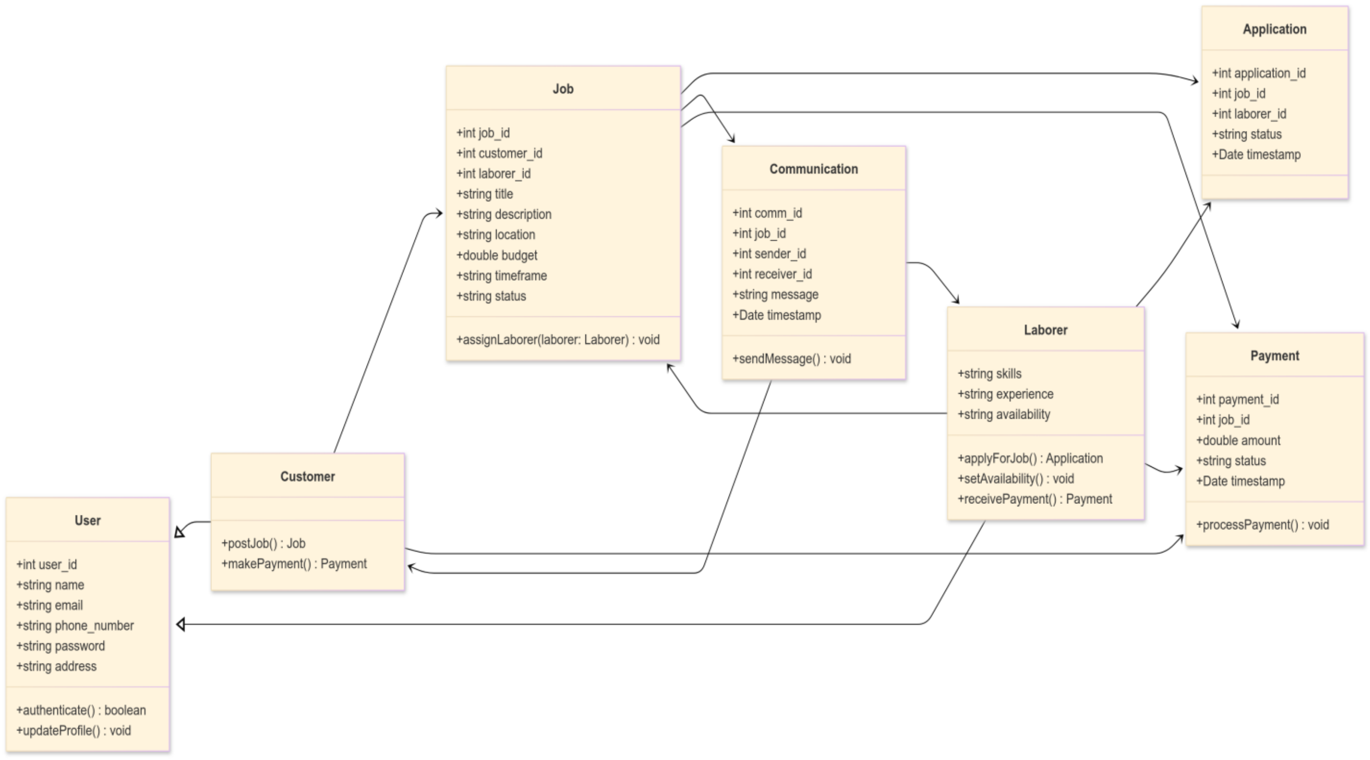
1. Use Case Diagram:



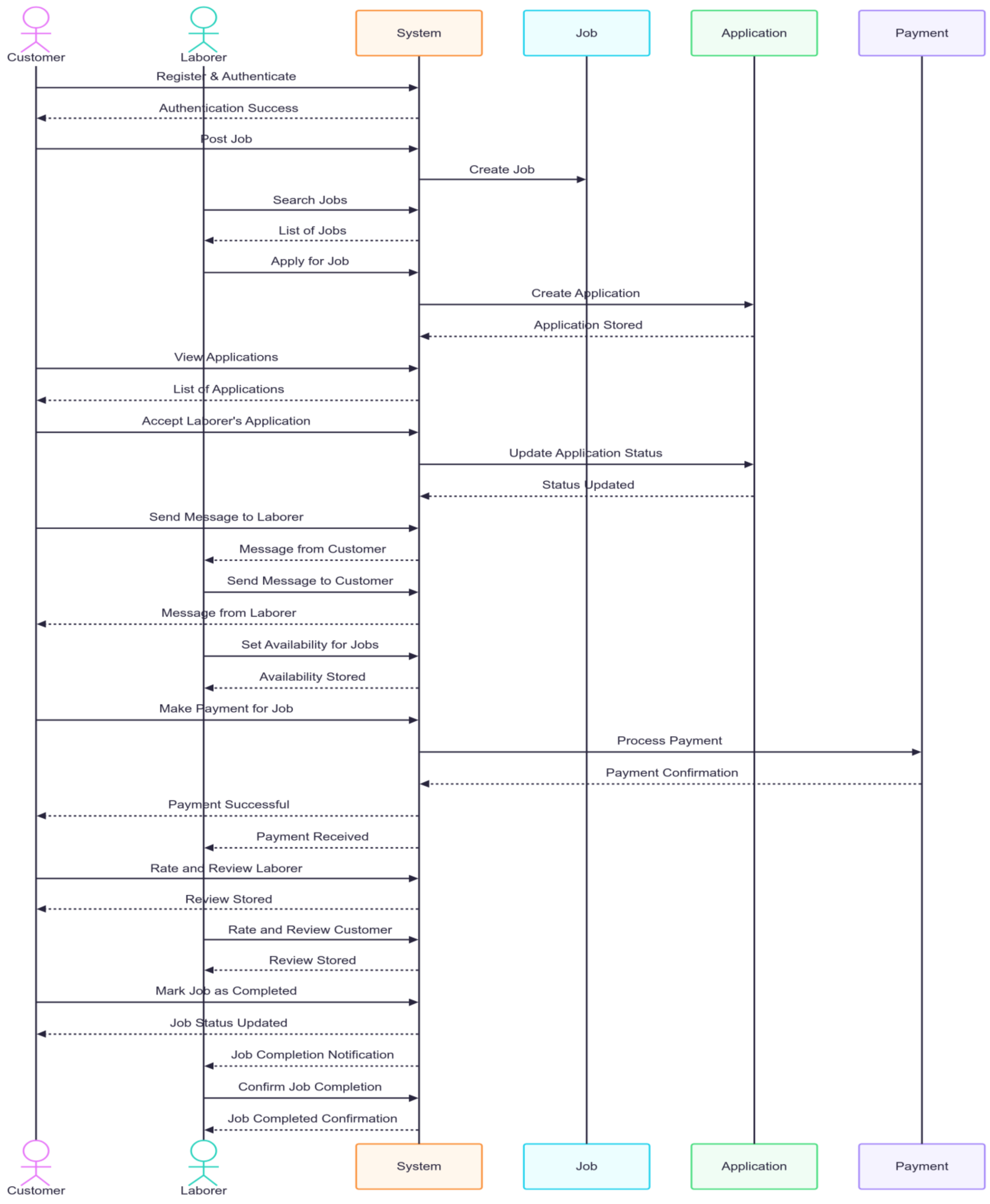
2. Entity Relation Diagram:



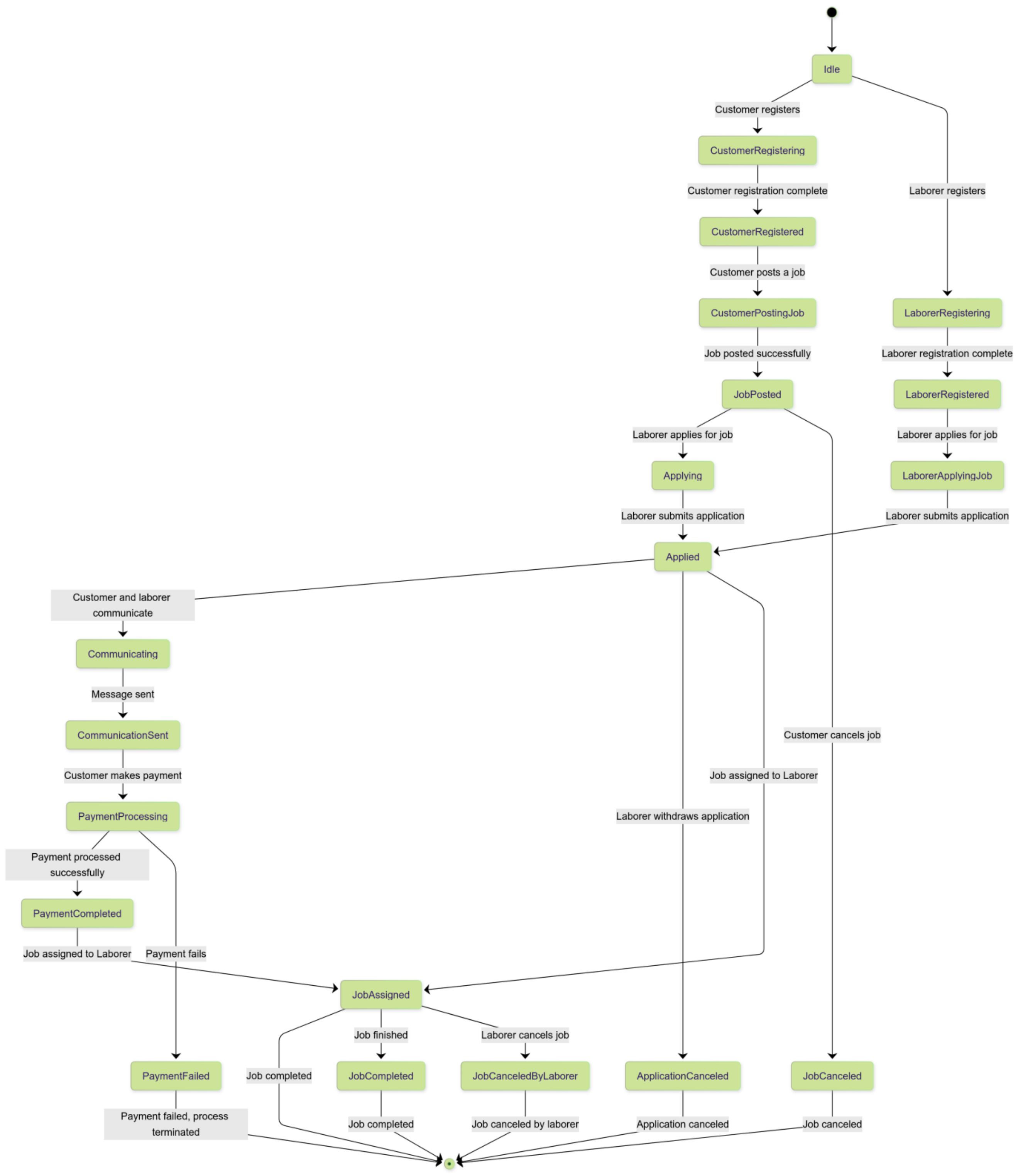
3. Class Diagram:



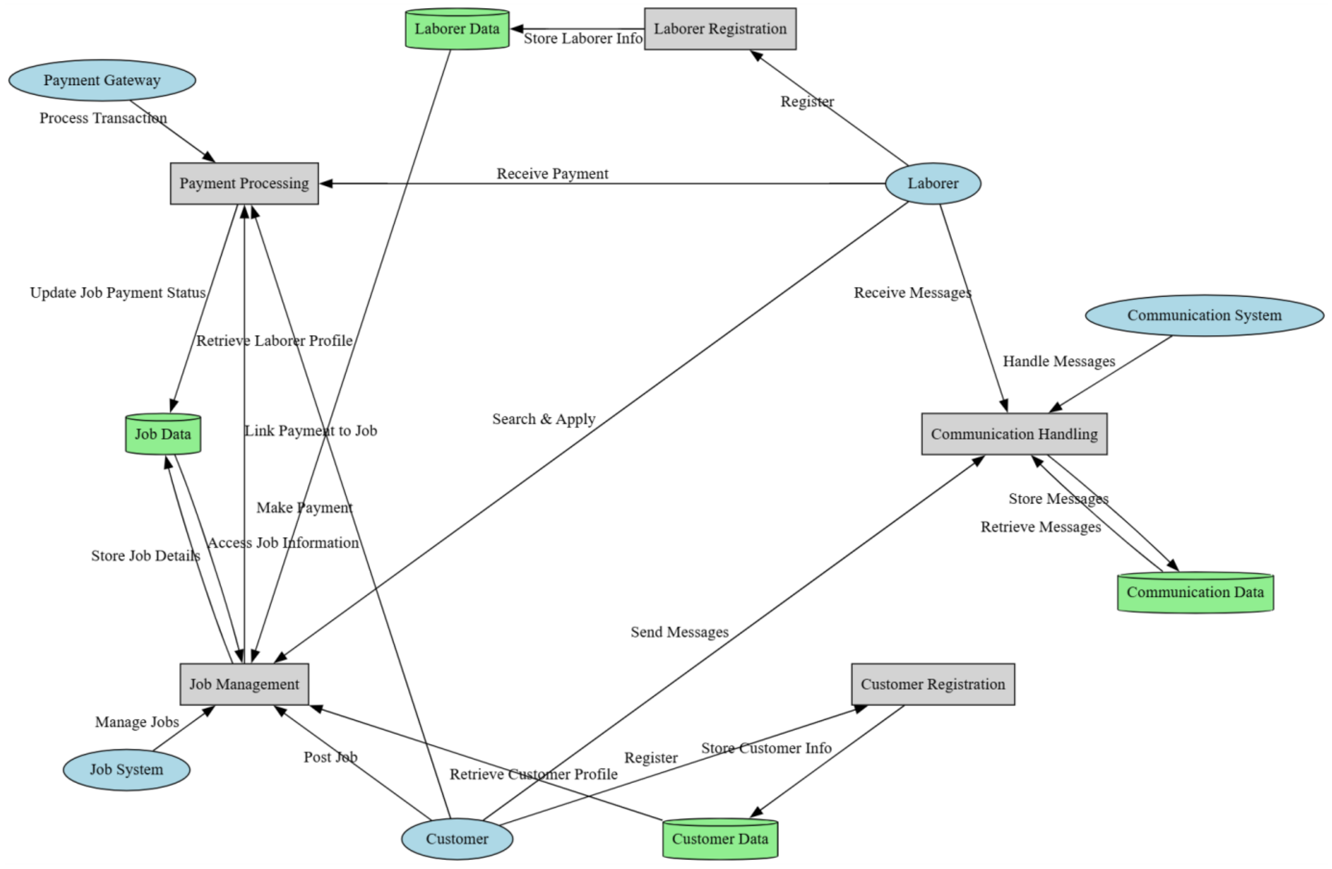
4. Sequence Diagram:



5. State Diagram:



6. Data Flow Diagram:



4. Nonfunctional Requirements

4.1 Performance Requirements

PER-1: Real-time chat messages should be received within 500 milliseconds in normal network conditions.
PER-2: Job search results will be shown within rapidly after filtering.
PER-3: New message or job request notifications shall be pushed within 2 seconds.
PER-4: Location-based job matches should be loaded quickly after opening the search screen.
PER-5: Portfolio and profile pictures should be loaded within 3 seconds when displayed.

4.2 Safety Requirements

SAF-1: Workers will be identity-verified when their accounts are created.
SAF-2: Both workers and customers will have the ability to enter emergency contact information.
SAF-3: The employment posting form will permit users to report potential job-related hazards.
SAF-4: The site shall have a "Report Incident" button on every job screen.
SAF-5: Contextual safety advice will be shown when users surf hazardous occupation categories.

4.3 Security Requirements

SEC-1: All users will authenticate through secure email/password mechanisms.
SEC-2: Passwords must be encrypted in the storage.
SEC-3: JWT tokens will be employed for session management with a 30-minute expiration.
SEC-4: All communications will be encrypted using HTTPS/SSL.
SEC-5: Unauthorized access attempts shall trigger a 403 Forbidden response and be logged.
SEC-6: Users will be permitted to ask for data erasure in line with privacy regulations.

4.4 Additional Software Quality Attributes

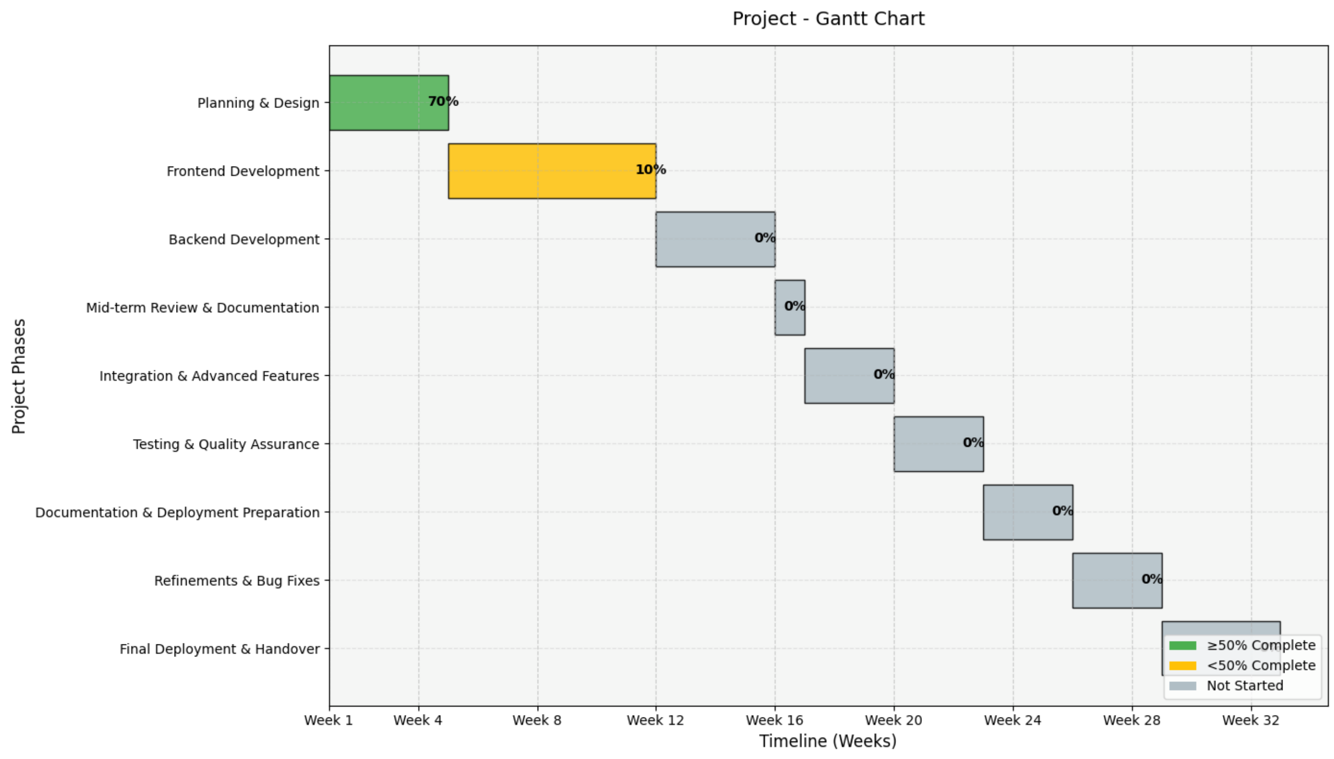
QUAL-1: The application shall have an intuitive interface usable by non-technical users.
QUAL-2: The system shall be available with 99.5% uptime excluding scheduled maintenance.
QUAL-3: All data will be backed up daily and saved safely for 30 days.
QUAL-4: Core features should be available in 3 taps from the main dashboard.
QUAL-5: English and Urdu will be supported by the platform with future localization support.
QUAL-6: UI elements should adhere to a consistent style and uphold accessibility requirements (e.g., color contrast).

5. Other Requirements

OTH-1: The system will incorporate Firebase for real-time messaging and authentication.
OTH-2: Google Maps API will be utilized for job-location functionality and proximity filtering.
OTH-3: The mobile app must be compatible with Android 10 and later.
OTH-4: The backend will be MongoDB and Node.js running on a scalable cloud platform.
OTH-5: The project will meet Pakistani data protection legislation and ethical use of IT.

6. Revised Project Plan

As per our proposal, we've done about 70% of the Planning & Design phase (Weeks 1–4) was done, this placing us into position to conclude the phase. We spent time investigating project needs, discovering pertinent insight, and sketching initial ideas for the website UI and UX. We have now developed a strong system architecture and the designs are falling into place.
The next stage of our work is the Frontend Development phase which is scheduled for (Weeks 5–-11). We're about 10% through.
Below, the Gantt chart clearly shows how the project has progressed, which tasks are completed, ongoing, and that which is upcoming in line with our plan.



7. References

· G. Haeringer and M. Wooders, *"Decentralized job matching,"* International Journal of Game Theory, vol. 40, no. 1, pp. 1–28, Jan. 2011.

· M. Nardini, S. Helmer, N. El Ioini, and C. Pahl, *"A Blockchain-Based Decentralized Electronic Marketplace for Computing Resources,"* SN Computer Science, vol. 1, art. no. 251, 2020.

· Pakistan Bureau of Statistics, *"Labour Force Statistics 2022-23,"* Government of Pakistan.

https://www.pbs.gov.pk/content/labour-force-statistics

· A. Kamal, *"Mutually beneficial: App connects job-seeking labourers with employers,"* The Express Tribune, May 14, 2014.

https://tribune.com.pk/story/786908/mutually-beneficial-app-connects-job-seeking-labourers-with-employers

· Beamexchange, *"Pakistan Labour Market Assessment – 2022,"* BEAM Exchange, 2022. https://beamexchange.org/resources/2038/

· AskSource Info, *"Labour Market Assessment – Pakistan – 2022,"* AskSource, 2022.

https://asksource.info/resources/labour-market-assessment-pakistan-2022

Appendix A: Glossary

Decentralized System	A system architecture where control and data are distributed across multiple nodes, allowing direct interaction between users without intermediaries.
Laborer/Worker	A skilled professional (e.g., electrician, plumber, carpenter) registered on the ProLabour platform to offer services.
Network Manager	A dedicated component of the ProLabour platform that facilitates peer-to-peer connections and manages direct communication between users.
OTP	One-Time Password, a temporary code sent to a user's phone for authentication during registration or login.
Pay-per-job	A payment model where laborers are compensated based on individual tasks completed rather than a fixed salary.
Portfolio	A digital collection of a laborer's work, including images, descriptions, and past job details, displayed on their profile.
Rating and Review System	A feature allowing customers and laborers to provide feedback (star ratings and written reviews) after job completion to build trust.
Real-time Notifications	Instant alerts sent to users via push notifications for job updates, messages, or application statuses.
UI/UX	User Interface/User Experience, referring to the design and interaction of the mobile application to ensure usability and satisfaction.
JWT	JSON Web Token, a standard used for secure session management and authentication in the ProLabour platform.
HTTPS/SSL	Hypertext Transfer Protocol Secure/Secure Sockets Layer, used to encrypt communications between the app and servers.
MongoDB	A NoSQL database used for storing user data, job postings, and other platform information.
Node.js	A JavaScript runtime environment used for backend development of the ProLabour platform.
Firebase	A platform used for real-time messaging, authentication, and push notifications in the ProLabour app.
Google Maps API	An application programming interface used for geolocation services, such as proximity-based job matching.

Appendix B: IV & V Report

(Independent verification & validation)
IV & V Resource

Name Signature

S#	Defect Description	Origin Stage	Status	Fix Time	
				Hours	Minutes
1					
2					
3					
…					

Table 2: List of non-trivial defects