

Department of computer science

BS-CS Final Year Project

Mazdoor Online



Presented by:

Registration No:

	Names:
20-cs-085	Nouman Nasir
20-cs-087	Muhammad Huzaifa Subhani
20-cs-089	Omama Shakeel

Supervised by:

Dr. Muhammad Nouman Noor

**A Project Submitted in Partial Fulfillment of the Requirements of the Degree of Bachelor in
Computer Science (BSCS)**

DEPARTMENT OF COMPUTER SCIENCE HITEC UNIVERSITY, TAXILA

DEDICATION

This dissertation is dedicated to our beloved parents. The success of this project was made possible by their tremendous support, unending prayers, and unwavering faith in us. Additionally, we dedicate this dissertation to our siblings and the respectable professors who guided our study and made it possible for us to finish this assignment successfully. The most important thing is to nominate the friends who have contributed greatly to this fulfillment since they have supported and influenced us in every way throughout my academic career in addition to my entire existence. These are the people who have prayed and crossed their fingers for our happiness and prosperity over and over again.

Certificate

It is certified that Nouman Nasir, Muhammad Huzaifa Subhani, and Osama Shakeel (Fa20-BS-CS-085/087/089) have carried out all the work related to this thesis/project under my guidance and supervision at the Department of Computer Science, HITEC University and the work fulfills and meets the prerequisites for the award of CS degree.

Date: _____

Supervisor: Dr. Muhammad Nouman Noor

Assistant Professor

Department of Computer Science,

HITEC University, Taxila

Head of Department: Dr. Junaid Ali Khan

Department of Computer Science,

HITEC University, Taxila

PROJECT IN BRIEF

Project Title: Mazdoor Online

Organization: Department Of Computer Science

Undertaken By: Nouman Nasir (20-CS-085)

Muhammad Huzaifa Subhani (20-CS-087)

Omama Shakeel (20-CS-089)

Supervised By: Dr. Muhammad Nouman Noor (Assistant Professor)

Date Started:

Date Completed:

Technology Used: Figma, MongoDB, Visual Studio, React-Native Framework.

Operating System: Windows 10

System Used: Core i5 8th Gen, 8 GB RAM, 256 SSD, 500 GB Hard disk

DECLARATION

We, hereby as a part of this project declare that this project was neither whole nor as a part of it is copied from any source. It is further declared; we personally developed this project, and the software requirements specification report is based on our personal efforts and hard work under the guidance and supervision of our sincere supervisor. It is clearly stated that not any portion of this report has been submitted before in support of any degree or other university or institute of learning. If any violation of rules is found in SRS the report, we will be liable to be punished under the plagiarism rules of HEC.

Nouman Nasir

M. Huzaifa Subhani

Omama Shakeel



Final Approval

This project

Mazdoor Online

By:

Registration No:

Names:

20-cs-085

Nouman Nasir

20-cs-087

Muhammad Huzaifa Subhani

20-cs-089

Omama Shakeel

Under the supervision of their project advisor and approved by the project examination committee, has been accepted by the HITEC university Taxila, Pakistan, in partial fulfillment of the requirements for the four years degree of **BS computer science**.

Dr. Muhammad Nouman Noor

Project Advisor

(HOD. Dr. Junaid Ali Khan)

Chairman

Department of Computer Science & Engineering

ACKNOWLEDGMENT

Thanks to **Allah the Almighty**, gave us the strength to work on this project and complete it on time with the best possible quality. We would like to express our gratitude to our supervisor **Dr. Muhammad Nouman Noor**, who guided us throughout this project. We would also like to thanks to our family and friends who supported us in each and every step of our life, especially in the past four years of university period.

Self believe and hard work will always earn you success! It is very proud moment for us, that we completed our **Final Year Project** on time without any delay. This is all due to hard work of our project members and very supportive supervision of our supervisor **Dr. Muhammad Nouman Noor**. He helps us in every critical situation of our development phase and guided us very good. Our supervisor always advised us to stay calm and not to panic in a difficult situation. We faced different difficult and pressurized situation but our supervisor support us and with the help of his guidance and support we overcome that troubles that came our way. **Dr. Muhammad Nouman Noor** is an industrial person and it is truly shows with his efforts in work as well as guidance. We appreciate his efforts and time that he gave us during project. Big thanks to our supervisor for standing beside in all times and making this possible.

Nouman Nasir

M. Huzaifa Subhani

Omama Shakeel

ABSTRACT

Mazdoor Online, a cutting-edge React Native application, revolutionizes the landscape of labor hiring in Pakistan. Through its innovative platform, users can easily connect with verified workers for various services, facilitated by comprehensive profiles, real-time chat, secure payments, and location-based searches. Spearheaded by Dr. Muhammad Nouman Noor, this project aims to modernize Pakistan's labor market by providing ample job opportunities and efficient services. In response to the COVID-19 pandemic, Mazdoor Online emerges as a timely solution, meeting the increased demand for online services while ensuring safety and reliability. With features like NADRA CNIC verification and a robust review system, Mazdoor Online sets new benchmarks for transparency and quality assurance. Digitizing and streamlining the hiring process addresses critical market needs and catalyzes the shift toward a more efficient and trustworthy labor ecosystem in Pakistan.

Table of contents

Contents

PROJECT IN BRIEF.....	4
DECLARATION	5
ACKNOWLEGMENT	7
ABSTRACT	8
Chapter 1	16
1. Introduction	16
1.1. Project Overview	16
1.2. Project Scope	16
1.2.1. Mobile Interface	16
1.2.2. Problem Statement.....	17
1.3. Proposed Solution.....	17
1.3.1. Proposed System Component	17
1.3.1.1. Administration management.....	17
1.4. Proposed System Output	17
1.4.1. User management.....	18
1.4.2. Developer management.....	18
1.4.3. Main features of proposed solution.....	18
1.4.3.1. Efficiency	18
1.4.3.2. User friendly interface	18
1.4.3.3. Minimum redundancy	18
1.4.3.4. Facilitated data inputs	18
1.5. Data security and integrity.....	18
1.6. Technical innovation	19
1.7. Existing System.....	19
Chapter 2.....	20
2. Software requirements specifications	20
2.1. Purpose.....	20
2.2. Scope	20
2.3. Proposed Solution Overview	20

2.3.1. Modules of the proposed solution.....	20
2.3.2. Proposed Solution.....	21
2.4. Specification requirements	21
2.4.1. Functional Requirement	21
2.4.2. Non-Functional Requirement	21
2.5. Advantages.....	22
2.5.1. Advantages for the Users (Labor)	22
2.5.2. Advantages for the Users (Customer).....	22
2.6. Product functions.....	22
2.6.1. Functional requirements explanation	22
2.7. General requirements	23
2.7.1. Software requirements.....	23
2.7.2. Hardware requirements	23
2.8. Use case diagrams	23
2.8.1 User end for (Labor).....	23
2.8.2 Admin end.....	26
Chapter 3.....	29
3.1. Use case diagram	29
3.2. Class diagram.....	30
3.3. Activity diagram	31
3.3.1. Login.....	31
3.3.2. Signup	32
3.4. Sequence diagram.....	33
3.4.1. Login.....	33
3.4.2. Signup	34
3.6. Data dictionary.....	35
3.6.1. User (Labor Signup).....	35
3.6.2. User (Labors).....	35
3.6.3. Labor Work Proposal Data.....	36
3.6.4. Customer Signup	36
3.6.5. User (Customers).....	37
3.6.7. Testing API's	37

Chapter 4.....	39
4. PROJECT MANAGEMENT	39
4.1. Project deliverables.....	39
4.2. Risk Management	39
4.3. Purpose.....	40
4.4. Risk management functions	40
Chapter 5.....	41
5. IMPLEMENTATION	41
5.1. Operating environment.....	41
5.1.1. Operating system.....	41
5.1.2. React-Native.....	41
5.1.2.1. React-Native features.....	41
5.1.2.2. System requirements for React-Native	42
5.1.3. MongoDB Atlas	42
5.1.4. Java Script	42
5.1.4.1. Java-Script features.....	42
5.1.4.2. System requirements for Java Script	42
Chapter 6.....	43
6. Software Testing:.....	43
6.1. Deriving test case specifications:.....	43
6.2. Testing Environment	43
6.2.1. Hardware Requirements.....	43
6.2.2. Software Requirements	43
6.2.3. Testing Identifications	43
6.3. Testing procedure	44
6.4. Test Cases.....	44
6.4.1. Test Case for Sign up module	44
6.4.1.1. Sign up as a User	44
6.4.2. Test Case for Login Module	46
6.4.2.1. Log in as a User	46
6.4.3. Test Case for User account module	48
6.4.3.1. Update user information	48

6.4.4. Test Case for Search Module	50
6.4.5 Test Case for Submit review module	51
6.4.6 Test Case for Sign out module	54
Chapter 7.....	56
7. Project Display Screens.....	56
7.1. Splash screen.....	56
7.2. Get Started Screen	56
7.3. Main Screen.....	57
7.4. Labor Login Screen.....	57
7.5. Labor Signup Screen	58
7.6. Labor Identification.....	58
7.7. Labor Choose Skill Screen.....	59
7.8. Labor Work Proposal Screen	59
7.9. Labor Profile Screen	60
7.10. Labor Update Profile Screen.....	60
7.11. Pop up Screen.....	61
7.12. Labor Menu.....	61
7.13. Labor Offers.....	62
7.14. Labor Work Due	62
7.15. Labor History	63
7.16. Customer Home Screen.....	63
7.17. Customer Login Screen.....	64
7.18. Customer Signup Screen	64
7.19. Customer Screen after Login	65
7.20. Customer Profile Screen	65
7.21. Customer Update Profile Screen	66
7.22. Customer Dropdown List.....	66
7.23. Customer Search Location.....	67
7.24. Customer Menu.....	67
7.25. Customer Book Labor.....	68
7.26. Customer Chat	68

7.27. Customer Rating to Labor	69
7.2 References.....	69

List of Figures

Figures

Figure 2 1: User	23
Figure 2 2: Login	24
Figure 2 3: Search in desired city.....	25
Figure 2 4: Post a review	26
Figure 2 5: Maintenance	27
Figure 2 6: Verification	27
Figure 3 1: Use case diagram of whole system	29
Figure 3 2: Class diagram of whole system	30
Figure 3 3: Activity diagram of login.....	31
Figure 3 4: Activity diagram of sign up.....	32
Figure 3 5: Sequence diagram of log in.....	33
Figure 3 6: Sequence diagram of sign up.....	34
Figure 3 7: User (Labor Signup).....	35
Figure 3 8: User (Labors).....	36
Figure 3 9: Labor Work Proposal Data	36
Figure 3 10: Each society with collection name	36
Figure 3 11: User (Customers).....	37
Figure 3 12: Test createUser API.....	37
Figure 3 13: Test User Login API.....	38
Figure 3 14: Test Work Proposal API	38
Figure 7. Project Display Screens	56
Figure 7.1. Splash screen	56
Figure 7.2. Get Started Screen	56
Figure 7.3. Main Screen.....	57
Figure 7.4. Labor Login Screen.....	57
Figure 7.5. Labor Signup Screen.....	58
Figure 7.6. Labor Identification.....	58
Figure 7.7. Labor Choose Skill Screen	59

Figure 7.8. Labor Work Proposal Screen	59
Figure 7.9. Labor Profile Screen.....	60
Figure 7.10. Labor Update Profile Screen	60
Figure 7.11. Pop up Screen	61
Figure 7.12. Labor Menu	61
Figure 7.13. Labor Offers	62
Figure 7.14. Labor Work Due	62
Figure 7.15. Labor History	63
Figure 7.16. Customer Home Screen	63
Figure 7.17. Customer Login Screen.....	64
Figure 7.18. Customer Signup Screen	64
Figure 7.19. Customer Screen after Login.....	65
Figure 7.20. Customer Profile Screen	65
Figure 7.21. Customer Update Profile Screen	66
Figure 7.22. Customer Dropdown List	66
Figure 7.23. Customer Search Location	67
Figure 7.24. Customer Menu.....	67
Figure 7.25. Customer Book Labor	68
Figure 7.26. Customer Chat	68
Figure 7.27. Customer Rating to Labor	69

List of tables

Tables

Table 1	39
Table 2	46
Table 3	47
Table 4	48
Table 5	51
Table 6	54
Table 7	54

Chapter 1

1. Introduction

In a rapidly evolving digital landscape, the "Mazdoor Online App" emerges as a response to the persistent challenges in labor hiring. This innovative mobile platform reimagines the process of connecting labor seekers with skilled workers, offering convenience, transparency, and efficiency. The labor market is undergoing profound changes, marked by the gig economy's rise, increased digitalization, and a growing demand for trust, transparency, accessibility, and efficiency in labor hiring. Traditional methods have proven insufficient, emphasizing the need for modern labor marketplaces like the "Mazdoor Online App" to reshape the way we find and hire laborers.

1.1. Project Overview

Mazdoor Online is an innovative mobile app developed with React-Native, aimed at simplifying the process of hiring skilled workers in Pakistan. It serves as a digital hub, improving the efficiency, transparency, and dependability of the labor hiring mechanism. The app allows users to find workers by service type and location, engage in live chat, and view authenticated feedback and ratings. Workers gain from more employment prospects and direct communication with clients.

The app's standout features are comprehensive profiles for users and workers, searches based on location, an open feedback and rating mechanism, and a stringent verification method involving CNIC and real-time location tracking. An administrative dashboard is in place to handle user verification and review validation.

On the technical side, Mazdoor Online employs React Native for the user interface, Node.js for server-side operations, MongoDB for database management, and the Google Maps API for location-based services. This holistic strategy delivers a smooth and secure experience for hiring labor, transforming Pakistan's marketplace by offering a dependable and effective service for both clients and workers.

1.2. Project Scope

Mazdoor Online is an innovative mobile application aimed at transforming the labor hiring landscape in Pakistan. The scope of the project encompasses the development, deployment, and maintenance of a comprehensive platform designed to connect labor seekers with skilled workers efficiently and reliably enhancing user experience and trust in the labor market.

1.2.1. Mobile Interface

The user will be able to access our App through his/her android or iOS.

1.2.2. Problem Statement

In Pakistan, the process of hiring skilled laborers for various tasks is often inefficient, time-consuming, and fraught with challenges such as lack of transparency and reliability. Traditional methods of finding laborers through word-of-mouth or informal networks are outdated and fail to provide a dependable way to connect with trustworthy workers.

Key Issues:

- **Inefficiency:** Finding skilled laborers requires significant time and effort, often resulting in delays and inconvenience for those in need of services.
- **Lack of Transparency:** Current methods lack a reliable system for verifying the credentials and trustworthiness of laborers, leading to potential issues of fraud and untrustworthy services.
- **Unreliable Outcomes:** Without a platform for verified reviews and ratings, users cannot make informed decisions, often resulting in unsatisfactory service experiences.
- **Limited Access:** Skilled laborers struggle to reach potential clients, limiting their job opportunities and affecting their livelihoods.

1.3. Proposed Solution

Mazdoor Online addresses these challenges by providing a digital platform that:

- Connects labor seekers with verified and skilled laborers quickly and efficiently.
- Ensures transparency through detailed profiles and verified reviews.
- Enhances reliability by requiring laborer verification via CNIC and location checks.
- Expands job opportunities for laborers by offering a broader reach and better visibility.

1.3.1. Proposed System Component

Here are several proposed components for the system.

1.3.1.1. Administration management

Admin management in Mazdoor Online involves verifying laborers' identities, approving user reviews, handling support queries, monitoring app usage, and ensuring compliance. Admins also manage content, finances, and regulatory requirements, aiming to enhance user experience and platform efficiency.

1.4. Proposed System Output

The system output of Mazdoor Online includes user profiles detailing personal and skill information, search results listing available laborers, a favorites list for users, chat logs for communication, job booking confirmations, and verified reviews with ratings for laborers. Additionally, geolocation services aid in finding nearby laborers, while admin reports provide insights into app usage and performance.

1.4.1. User management

User management for Mazdoor Online involves enabling access via any browser, allowing users to create accounts, and providing access to society reviews. Our system securely maintains and records user information, ensuring a seamless and efficient experience for users interacting with the platform.

1.4.2. Developer management

Mazdoor Online includes deploying the system on web browsers, facilitating updates and modifications for future enhancements. Our system stores and manages developer information, ensuring a streamlined process for maintaining and improving the platform over time.

1.4.3. Main features of proposed solution

1.4.3.1. Efficiency

Mazdoor Online includes deploying the system on web browsers, facilitating updates and modifications for future enhancements. Our system stores and manages developer information, ensuring a streamlined process for maintaining and improving the platform over time.

1.4.3.2. User friendly interface

Mazdoor Online boasts a user-friendly interface with an intuitive layout and clear navigation, making it easy for users to interact with the platform. This design allows quick access to services, account creation, and review viewing, enhancing the overall user experience.

1.4.3.3. Minimum redundancy

Mazdoor Online uses a robust data management system with well-structured tables to minimize redundancy and ensure efficient data storage. This design enhances performance, maintains data integrity, and optimizes resource utilization.

1.4.3.4. Facilitated data inputs

Mazdoor Online streamlines data entry with intuitive forms, allowing users and laborers to easily input and update information. This reduces errors and effort, ensuring accurate, efficient recording of profiles, job details, and reviews with minimal redundancy.

1.5. Data security and integrity

Mazdoor Online employs robust encryption and access controls to protect user data, ensuring its confidentiality and integrity. Regular audits and compliance with data protection standards maintain the system's reliability and trustworthiness.

1.6. Technical innovation

Mazdoor Online leverages cutting-edge technology to enhance functionality and user experience. This includes real-time communication, geolocation services, and efficient data management, ensuring a modern, responsive, and robust platform.

1.7. Existing System

Other systems may lack a strong review system and may not prioritize user-generated content, resulting in limited transparency and user trust. Mazdoor Online excels over others due to its user-centric approach, which prioritizes authentic user-generated content like reviews, ensuring transparency and trust. The comprehensive platform empowers users with detailed information for informed decisions, setting it apart and providing a superior user experience.

Chapter 2

2. Software requirements specifications

2.1. Purpose

A software requirements specification (SRS) is a comprehensive document that defines the essential features, characteristics, and specifications of a project, software, or application. It serves as a foundational blueprint that should be developed prior to the commencement of any project or application development process. The primary purpose of an SRS document is to provide a concise yet detailed overview of the intended product, enabling a clear understanding of its functionalities and requirements. This document is primarily targeted towards technical professionals, such as developers, engineers, and subject matter experts, who possess the necessary expertise to comprehend and translate the specifications into a tangible product.

2.2. Scope

Mazdoor Online revolutionizes Pakistan's labor market, offering a user-friendly app that connects users with skilled laborers. With detailed profiles, real-time chat, and location-based searches, it transforms hiring into an effortless, transparent, and efficient experience, setting new standards in convenience and trust.

2.3. Proposed Solution Overview

2.3.1. Modules of the proposed solution

The modules of the Mazdoor Online system are as follows:

- **Signup:** Allows new users to create an account by providing necessary details.
- **Login:** Enables existing users to access their accounts using their credentials.
- **Post Reviews:** Users can post reviews about their experiences with laborers.
- **Admin Panel:** Admins can manage user accounts, oversee reviews, and ensure system integrity.
- **Registration:** Users can register their profiles, providing detailed information about their skills (for laborers) or requirements (for customers).
- **Search Bar:** Provides a powerful search functionality allowing users to find laborers based on skills, location, or other criteria.
- **User Interface (UI):** A user-friendly and intuitive interface that enhances user experience.
- **Real-Time Updates:** Provides real-time updates on job status, messages, and notifications.
- **Authentication:** Ensures secure login and access to the system.
- **Verify User Account:** Verification process to ensure users' identities through CNIC and location.

- **Verify Review Before Posting:** Reviews are verified to maintain authenticity and trustworthiness before being posted.

2.3.2. Proposed Solution

Our solution focuses on creating a secure and reliable platform for labor hiring. Here's how we aim to achieve this:

- **User Authentication:** We will authenticate users by verifying their live location or by requiring them to upload their CNIC (Computerized National Identity Card) or utility bill. This process ensures all users on the platform are genuine.
- **Posting Reviews:** Once authenticated, users can post reviews about different places or societies they have interacted with. This helps in maintaining the authenticity and reliability of the reviews.
- **Recommendation System:** The platform will feature a recommendation system that suggests the highest-rated societies or services based on the user's city search.

2.4. Specification requirements

2.4.1. Functional Requirement

- User Registration and Login
- Favorites Management
- Search Functionality
- Reviews and Ratings
- Admin Management
- Guest Search
- Navigation

2.4.2. Non-Functional Requirement

- Ensure secure access to confidential user data.
- Design the system for efficient performance under high user activity.
- Implement a flexible architecture for easy management of future extensions.
- The system should maintain an uptime of 99.9%, with meaningful error messages and logging for analysis and debugging.
- The code base should follow best practices and standards for readability and ease of maintenance, accompanied by comprehensive documentation.
- The mobile application should be optimized for minimal battery consumption on users' devices and compatible with both Android and iOS platforms.

2.5. Advantages

2.5.1. Advantages for the Users (Labor)

- Increased job opportunities by accessing a wider pool of potential clients.
- Verified profiles through CNIC authentication to build trust with clients.
- Real-time communication via chat feature for better coordination with clients.
- Ability to receive reviews and ratings to build reputation and attract more work.
- Efficient job management tools for handling work requests and job history.

2.5.2. Advantages for the Users (Customer)

- Access to a pool of verified and skilled laborers.
- Ability to read reviews and ratings to make informed hiring decisions.
- Real-time chat feature for direct communication with laborers.
- Convenient booking and scheduling of services.
- Location-based searches to find nearby laborers quickly.

2.6. Product functions

2.6.1. Functional requirements explanation

User Registration and Login: Users must have a valid email, phone number, and password to create an account and log in.

Favorites Management: Users can save their searched societies to their favorites list.

Search Functionality: Users can search for societies by selecting a city and society name from a dropdown menu. If only one city is selected, all societies in that city will be displayed.

Reviews and Ratings: Users can view verified reviews and ratings of societies from other residents. To submit a review and rating, users must upload their CNIC or utility bill for location verification. Users failing verification will not be allowed to submit reviews.

Admin Management: Admins manage user authentication and approve reviews submitted by verified users.

Guest Search: Users can search for societies in any city without creating an account. Account verification is required for further actions like submitting reviews or adding societies to favorites.

Navigation:

- **Bottom Navigation Menu:** Provides access to Home, Favorites, and Profile pages.
- **Home:** Takes users back to the main screen.
- **Profile:** Enables users to update their profile information.
- **Chat:** Facilitates communication between customers and laborers.

- **Work Due:** Manages tasks and deadlines.
- **History:** Provides a record of user interactions.
- **Notification:** Customers give ratings to laborers.

Side Menu: Provides access to additional pages.

- **Home:** Takes users back to the main screen.
- **Profile:** Enables users to update their profile information.
- **Chat:** Facilitates communication between customers and laborers.
- **Work Due:** Manages tasks and deadlines.
- **Search:** Customer searches for labor and then booked.
- **History:** Provides a record of user interactions.
- **Logout:** Logs users out of their accounts.

2.7. General requirements

2.7.1. Software requirements

- Database server: MongoDB Atlas.
- Development tools: VS code, React Native Framework.
- Programming language: Java Script, Type Script.

2.7.2. Hardware requirements

- Processor: Intel Core i5 8th Gen
- RAM: 8 GB
- Hard Disk: 500 GB / SSD: 256 GB.

2.8. Use case diagrams

2.8.1 User end for (Labor)

Name: "Labor",
 Email: "huzaifaa@gmail.com",
 Phone Number: "03058683549",
 CNIC: "3740684520599",
 Password: "123",

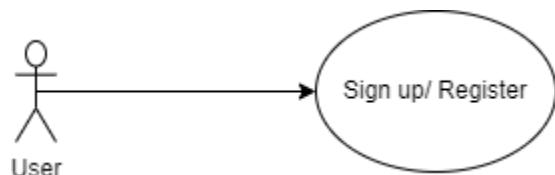


Figure 2 1: User

Pre- condition:

To access the application's services, users need to register by providing a name, valid email, phone number, CNIC, Image, City and password for creating their account.

Post-condition:

Upon validation of the user account, the registration process will be deemed successful.

Description:

To access the application's services, users are required to create an account by providing the necessary information. Following verification, the registration will be confirmed as successful.

User end for (Customer)

Name: "Customer",
Email: "huzaifaa@gmail.com",
Phone Number: "03058683549",
Password: "123",
Confirm Password: "123",
Image: "Base.url"

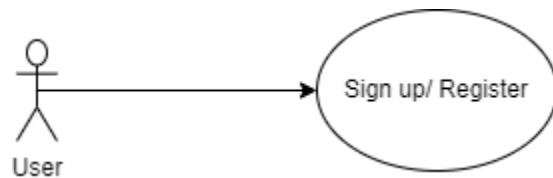


Figure 2 2: User

Pre- condition:

To access the application's services, users need to register by providing a name, valid email, phone number, image and password for creating their account.

Post-condition:

Upon validation of the user account, the registration process will be deemed successful.

Description:

To access the application's services, users are required to create an account by providing the necessary information. Following verification, the registration will be confirmed as successful.

Login

Email: huzaifaa@gmail.com
Password: 123
Actors: User (Labor/Customer)

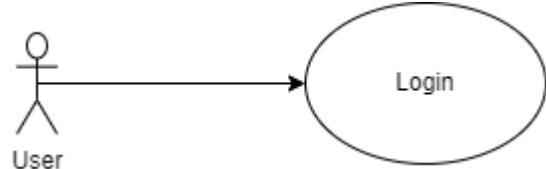


Figure 2 3: Login

Pre-condition:

Before accessing the application's features, users must log in by providing a valid email address and password. After these credentials are validated, the user gains access to all functionalities of the application.

Post-condition:

Once the login process is successfully completed and the user's credentials are verified, the user will be logged into the application.

Description:

To use the application's features, users need to log in with their email address and password. These inputs are initially checked on the client side for validation issues when the user fills out the login form and clicks the login button. The application then searches the database to verify if the account exists. If the provided credentials match a registered user in the database, the user's session begins, granting them access to the application.

Search

- Users can choose only the city from the drop-down menu to view all available laborers within that city. Efficient filtering options enable users to refine their search results based on skill type, rating and availability, ensuring they find the best match.
- The dual search option offers flexibility based on user needs.

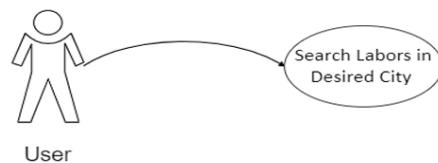
Actors: User

Figure 2 4: Search in desired city

Pre-condition:

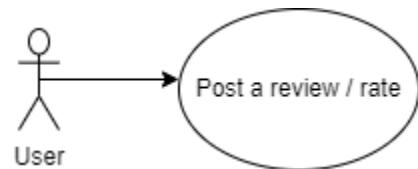
- The user must have a clear understanding of the specific labor service they require.
- Access to the Mazdoor Online app and logged-in status are necessary to perform a search operation.

Post-condition:

- The displayed results are presented once the user chooses the city or Skill from the drop-down menu or manually searches.
- A list of available labor services based on the search criteria is viewable to the user.

Description:

- The Mazdoor Online app enables users to search for labor services based on their desired city.
- Users can view all relevant labor services in their selected city by utilizing the dropdown menu options or manually searching.

Review and Rating:**Actors:** User**Figure 2 5: Post a review****Pre-condition:**

- Users are required to create an account and log in to access the review posting feature.
- After logging in, users can search for their targeted city from the drop-down menu.
- Users can then proceed to write a review and provide ratings for different aspects such as security, cleanliness, etc.

Post-condition:

Upon clicking the "Post" button, the user's review is publicly posted.

Description:

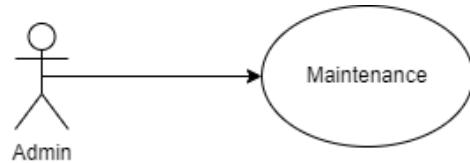
- Users must first create an account and log in to the Mazdoor Online app.
- Once logged in, users can select their city, and Labor profile for which they want to provide a review.
- Users can then proceed to rate and provide reviews for specific aspects of the society, such as security, cleanliness, etc.
- Upon clicking the "Post" button, their review is publicly posted.

2.8.2 Admin end

Maintenance:

Admin do the following:

- Checking system's working
- Database maintenance
- Environment menu



Actors: Admin

Figure 2 6: Maintenance

Pre-condition:

The administrator will assess the system's performance to identify any potential issues that could negatively impact its functioning.

Post-condition:

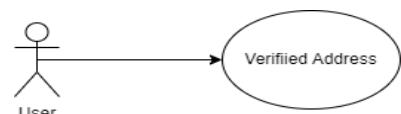
If any problems or defects are detected, the administrator will take necessary steps to resolve the identified issues and thoroughly inspect the entire system, including the database, to ensure it is operating correctly and without any remaining flaws.

Description:

If the administrator identifies a flaw or problem that could potentially degrade the system's performance, they will conduct a comprehensive evaluation of the system's performance. If any issues or errors are discovered during this evaluation, the administrator will promptly take corrective actions to address and rectify the identified problems. Subsequently, the administrator will perform a thorough examination of the entire system, encompassing the database, to verify that all components are functioning optimally and without any remaining defects or issues.

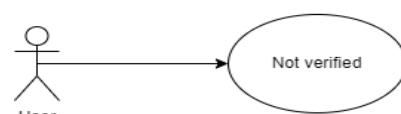
Authentication

Actors: Admin



Pre-condition:

For verification, the user (labor) will have to upload a picture of their Computerized National Identity Card (CNIC).



Post-condition:

Figure 2 7: Verification

The admin will verify the labor's address according to the society or locality address mentioned on the CNIC. If both addresses match, the admin will verify the labor's account.

Description:

To complete the verification process, the labor will be required to provide a clear photograph or scanned copy of their CNIC. The admin will cross-check the address mentioned on the CNIC against the address of the society or locality specified by the laborer during the registration process. If the addresses from both sources are an exact match, the admin will grant verification status to the labor's account on the "Mazdoor Online" app, confirming their identity and residential location.

Chapter 3

3.1. Use case diagram

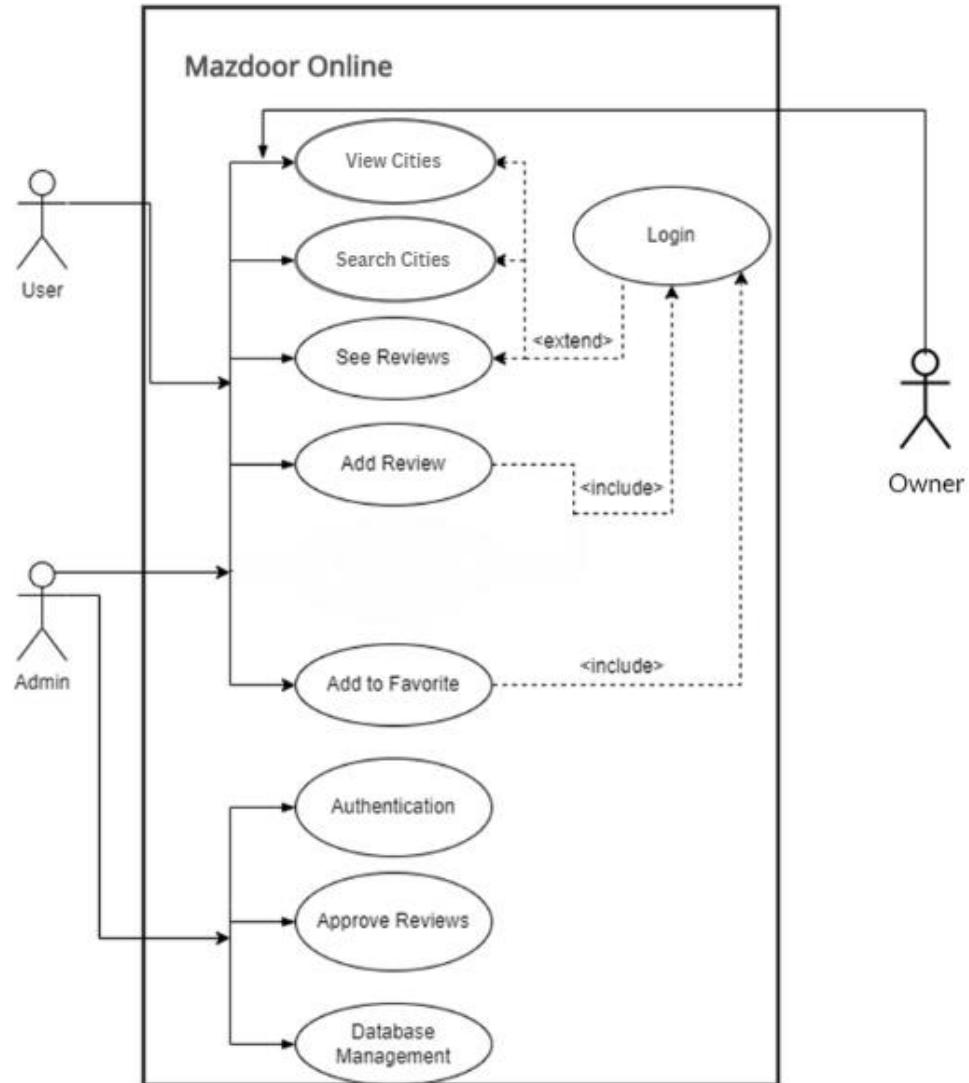


Figure 3 1: Use case diagram of whole system

3.2. Class diagram

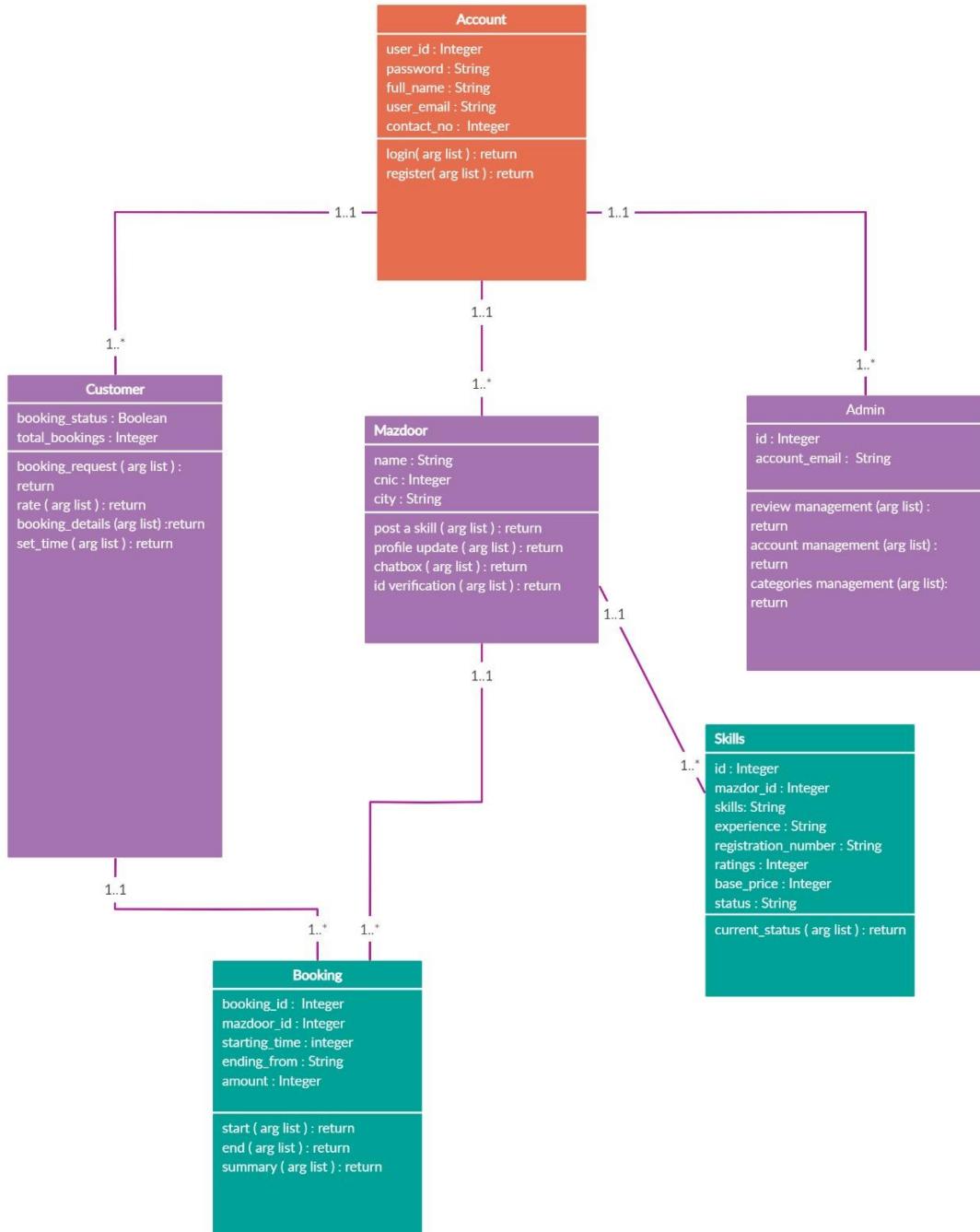


Figure 3 2: Class diagram of whole system

3.3. Activity diagram

3.3.1. Login

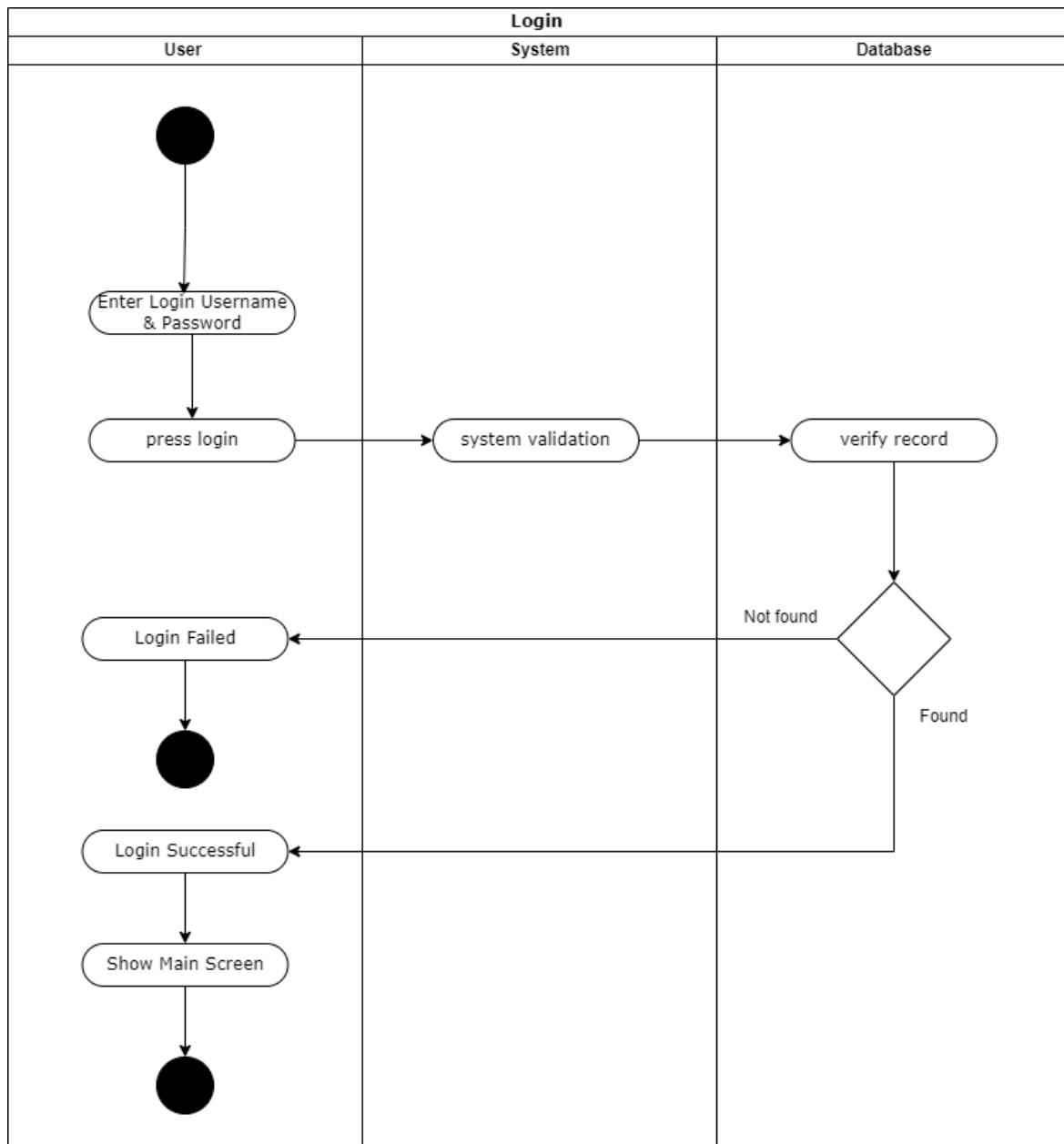


Figure 3 3: Activity diagram of login

3.3.2. Signup

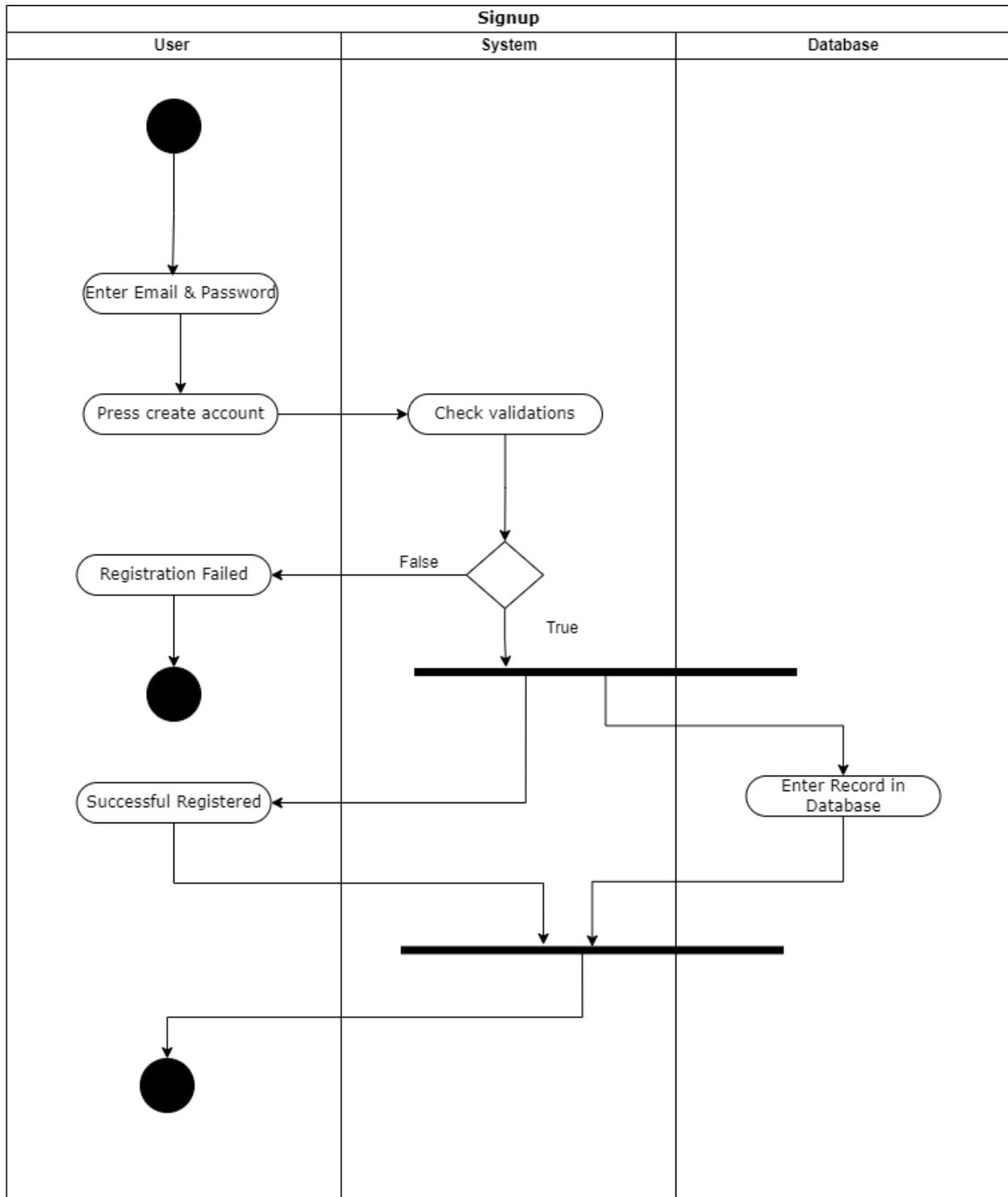


Figure 3 4: Activity diagram of sign up

3.4. Sequence diagram

3.4.1. Login

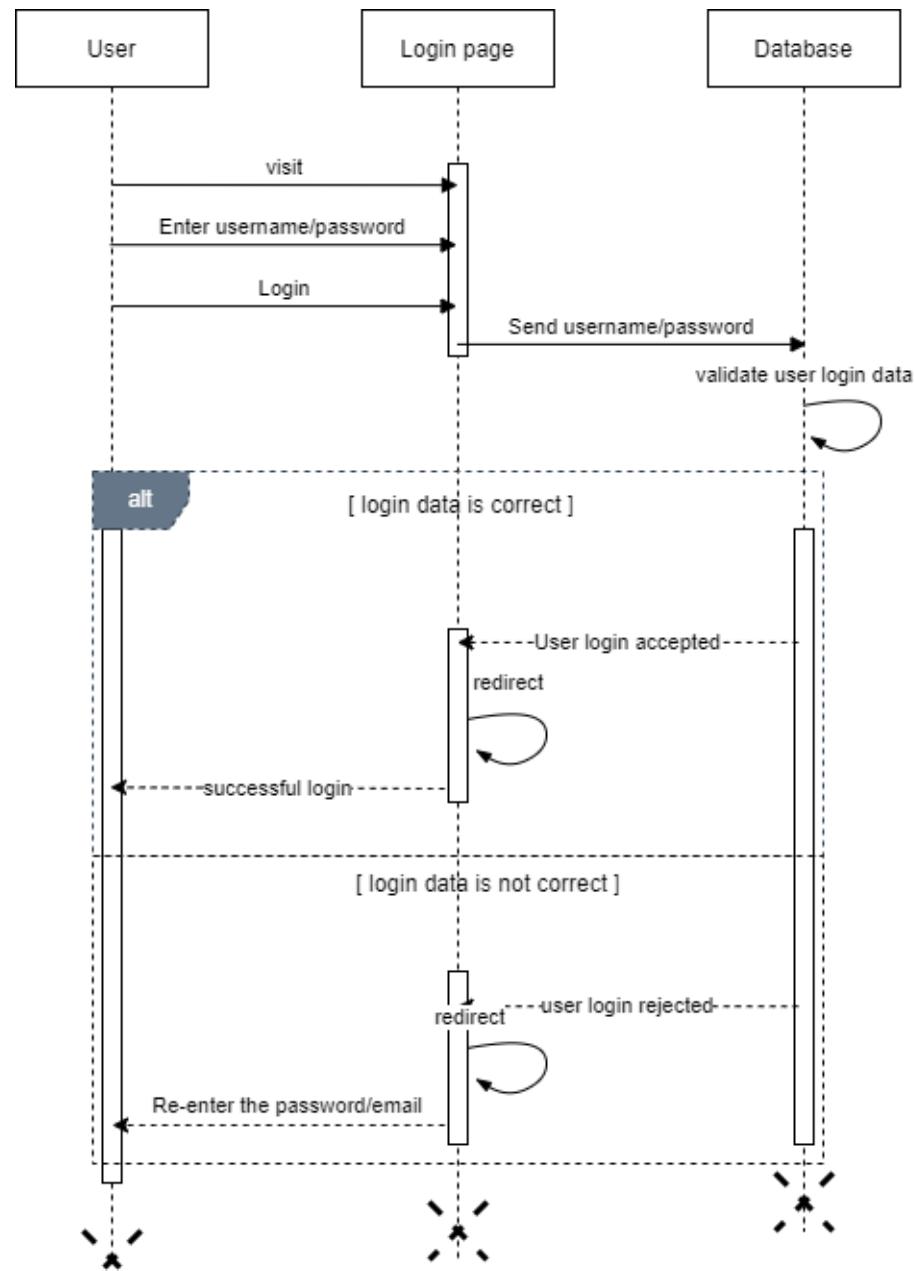


Figure 3 5: Sequence diagram of log in

3.4.2. Signup

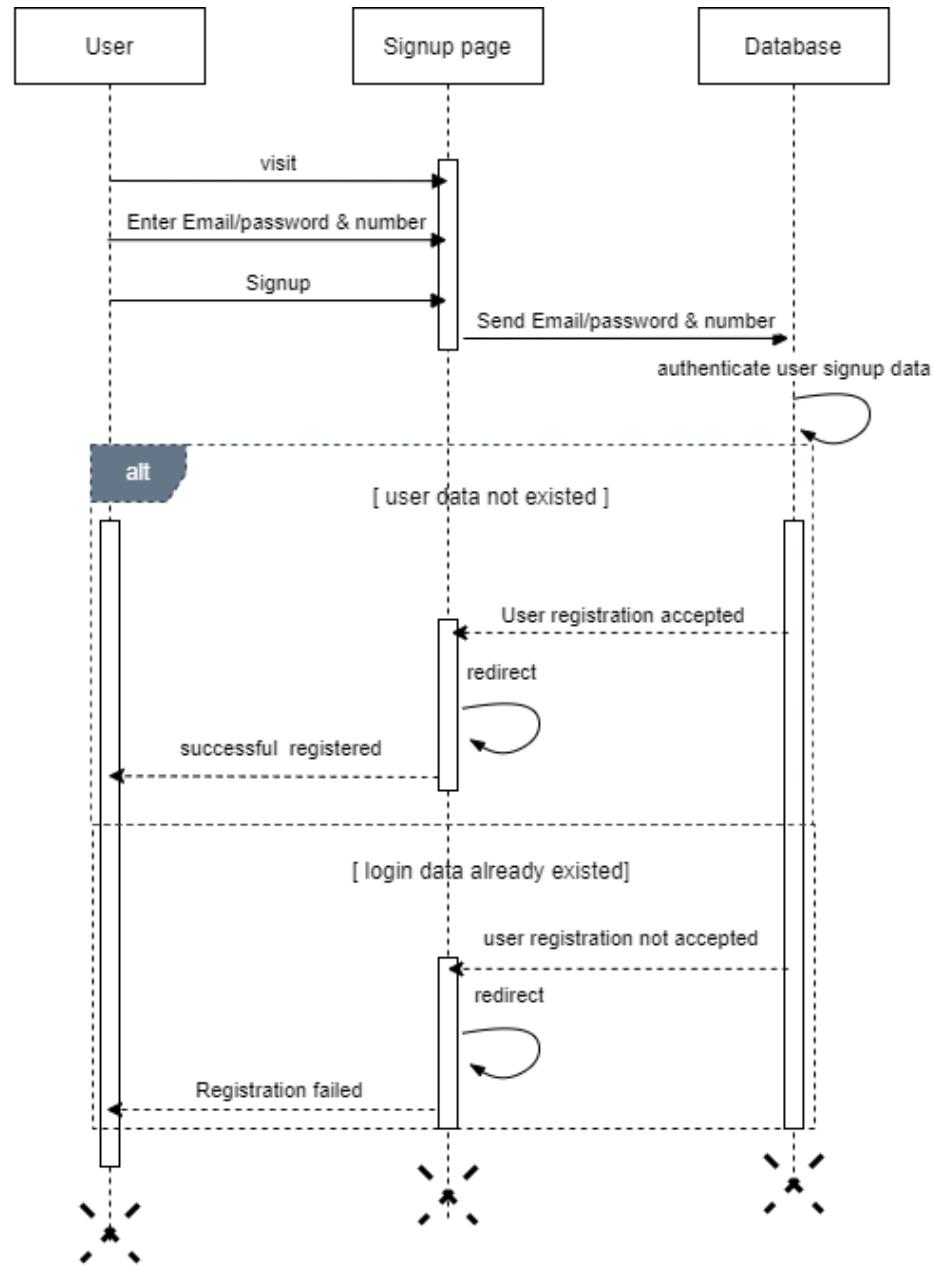


Figure 3 6: Sequence diagram of sign up

3.6. Data dictionary

3.6.1. User (Labor Signup)

The screenshot shows the MongoDB Compass interface. The left sidebar lists databases: admin, local, sample_mflix, test (which is expanded to show customers, customersignups, labordetails, labors, laborSignups, and lstats). The 'laborSignups' collection is selected and highlighted in green. The main pane displays the 'Documents' tab with 6 results. A search bar at the top says 'Type a query: { field: 'value' } or Generate query'. Below it are buttons for ADD DATA, EXPORT DATA, UPDATE, and DELETE. The results pane shows two documents:

```
_id: ObjectId('665ef846b9b1b0c20f0c0b8c')
name: "Huzaifa"
email: "huzaifaaaa@gmail.com"
phoneNumber: "+923058683549"
cnic: "3540167342571"
password: "12345678"
city: "Wah"
createdAt: 2024-06-04T11:19:34.759+00:00
__v: 0

_id: ObjectId('665ef872b9b1b0c20f0c0b90')
name: "Huzaifa"
email: "huzaifasubhani2002@gmail.com"
phoneNumber: "+923058683549"
cnic: "3740684520599"
password: "huzaifa123"
city: "Wah"
```

Figure 3 7: User (Labor Signup)

3.6.2. User (Labors)

The screenshot shows the MongoDB Compass interface. The left sidebar lists databases: admin, local, sample_mflix, test (expanded to show customers, customersignups, labordetails, labors, laborSignups, and lstats). The 'labors' collection is selected and highlighted in green. The main pane displays the 'Documents' tab with 30 results. A search bar at the top says 'Type a query: { field: 'value' } or Generate query'. Below it are buttons for ADD DATA, EXPORT DATA, UPDATE, and DELETE. The results pane shows two documents:

```
_id: ObjectId('665ccedeb01261a224373ae5')
name: "Choudhary Asif"
email: "choudhary@gmail.com"
phoneNumber: "+9230928266171"
cnic: "3422223338"
password: "Testpass"
city: "Layyah"
image: "baseUrl"
createdAt: 2024-06-02T19:58:22.099+00:00
updatedAt: 2024-06-02T19:58:22.099+00:00
__v: 0

_id: ObjectId('665ed4e4962a295c4f531fee')
name: "Huzaifa"
email: "huzaifaaaa@gmail.com"
phoneNumber: "+923058683549"
cnic: "3540167342571"
```

Figure 3 8: User (Labors)

3.6.3. Labor Work Proposal Data

The screenshot shows the MongoDB Compass interface. The left sidebar lists databases: admin, local, sample_mflix, test, customers, customersignups, labordetails, labors, laborsignups, and lstats. The 'lstats' collection is selected. The main pane displays the 'Documents' tab with 44 documents. A query builder bar at the top says 'Type a query: { field: 'value' } or Generate query'. Below it are buttons for ADD DATA, EXPORT DATA, UPDATE, and DELETE. The results show three document snippets:

```
_id: ObjectId('665efe95c910715743c9ad5c')
price : "300"
experience : "3"
laborEmail : "huzaifaaaa@gmail.com"
__v : 0

_id: ObjectId('665eff00f9416c355710dc2c')
price : "300"
experience : "6"
laborEmail : "huzaifaaaa@gmail.com"
__v : 0

_id: ObjectId('665f0a544da4aca821ba9fb6')
price : "300"
experience : "6"
```

Figure 3 9: Labor Work Proposal Data

3.6.4. Customer Signup

The screenshot shows the MongoDB Compass interface. The left sidebar lists databases: admin, local, sample_mflix, test, customers, customersignups, labordetails, labors, laborsignups, and lstats. The 'customersignups' collection is selected. The main pane displays the 'Documents' tab with 4 documents. A query builder bar at the top says 'Type a query: { field: 'value' } or Generate query'. Below it are buttons for ADD DATA, EXPORT DATA, UPDATE, and DELETE. The results show two document snippets:

```
_id: ObjectId('66595224ca8c18a8420d8a4c')
name : "Customer"
email: "huzaifaa@gmail.com"
phoneNumber : "03058683549"
password : "123"
createdAt : 2024-05-31T04:29:24.112+00:00
__v : 0

_id: ObjectId('66595a33c8a104707f4d2941')
name : "Nouman"
email: "bajwanouman7@gmail.com"
phoneNumber : "03185658477"
password : "Nouman1234"
createdAt : 2024-05-31T05:03:47.972+00:00
__v : 0
```

Figure 3 10: Each society with collection name

3.6.5. User (Customers)

The screenshot shows the MongoDB Compass interface. The left sidebar lists databases: admin, local, sample_mflix, test, and customers. The 'customers' database is selected. The main area shows the 'customers' collection with 11 documents. A query builder at the top right allows typing a query or generating one. Below it are buttons for ADD DATA, EXPORT DATA, UPDATE, and DELETE. The results table shows two documents:

```
_id: ObjectId('665f5939e954be8443a5f5ac')
name : "Customer"
email : "huzaifaa@gmail.com"
phoneNumber : "93058683549"
password : "123"
image : "Base.url"
createdAt : 2024-06-04T18:13:13.777+00:00
updatedAt : 2024-06-04T18:13:13.777+00:00
__v : 0

_id: ObjectId('6662012a11718e2ab37ba31b')
name : "Huzaifa"
email : "huzaifasubhani2002@gmail.com"
phoneNumber : "93058683549"
password : "huzaifa123"
createdAt : 2024-06-06T18:34:18.546+00:00
updatedAt : 2024-06-06T18:34:18.546+00:00
__v : 0
```

Figure 3 11: User (Customers)

3.6.7. Testing API's

The screenshot shows a Postman request to `http://192.168.226.88:4500/api/v1/auth/createUser`. The method is POST. The body contains the following JSON:

```
1 {
2   "name": "Huzaifa",
3   "email": "huzai@gmail.com",
4   "phoneNumber": "923058683549",
5   "cnic": "35401673425714",
6   "password": "12345678".
```

The response status is 201 Created, with a response time of 355 ms and a size of 591 B. The response body is:

```
1 {
2   "message": "User created successfully",
3   "user": {
4     "name": "Huzaifa",
5     "email": "huzai@gmail.com",
6     "phoneNumber": "923058683549",
7     "cnic": "35401673425714",
8     "city": "wah"
9   },
10  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
```

Figure 3 12: Test createUser API

The screenshot shows the Postman interface with the following details:

- Request URL:** `http://localhost:4500/api/v1/auth/login`
- Method:** POST
- Body Content:**

```
1 {
2   "email": "huzai@gmail.com",
3   ...
4   "password": "12345678"
5 }
```
- Response Status:** 200 OK
- Response Headers:** 142 ms, 597 B
- Response Body (Pretty JSON):**

```
1 {
2   "success": true,
3   "message": "User login successful",
4   "user": {
5     "name": "Huzaifa",
6     "email": "huzai@gmail.com",
7     "phoneNumber": "923058683549",
8     "cnic": "35401673425714",
9     "city": "wah"
10   },
11   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
```

Figure 3 13: Test User Login API

The screenshot shows the Postman interface with the following details:

- Request URL:** `http://localhost:4500/api/v1/auth/laborStats`
- Method:** POST
- Body Content:**

```
1 {
2   "price": "250",
3   "experience": "14"
4 }
```
- Response Status:** 201 Created
- Response Headers:** 235 ms, 447 B
- Response Body (Pretty JSON):**

```
1 {
2   "message": "Labor statistics created successfully",
3   "laborStats": {
4     "price": "250",
5     "experience": "14",
```

Figure 3 14: Test Work Proposal API

Chapter 4

4. PROJECT MANAGEMENT

4.1. Project deliverables

Here is the list of project deliverables:

Deliverables	Time
Proposal	1st Month
Required Specification	2nd Month
Design	3rd Month
Development & Implementation	4 th 5 th 6 th & 7 th Month
Testing & Debugging	8 th Month
Coded review	9 th Month
Documentation	10 th Month

Table 1

4.2. Risk Management

Developing secure, maintainable, and user-friendly software like the "Mazdoor Online" app requires addressing numerous potential risks. Risk management is a critical aspect in this regard. The objective of the risk management plan is to prevent data loss, system crashes, and misuse. Necessary precautions have been implemented to safeguard against insecure practices. During user interactions, the app helps in the form of email authentication and strong password requirements for secure accounts. Users also receive warning alerts when entering incorrect credentials. The system ensures complete privacy for users.

4.3. Purpose

The primary objective of risk management is to safeguard the system from various security threats, such as hacking attempts, unauthorized network traffic monitoring (sniffing), and impersonation of legitimate entities (spoofing). By exercising vigilance and implementing appropriate measures, user accounts will be shielded from potential breaches. Ensuring user privacy is another crucial aspect of risk management.

4.4. Risk management functions

We make sure our system is very safe for posting reviews on social media. We don't allow special symbols in reviews or emails to make sure only real users can use their accounts. If someone forgets their password, they can reset it by following a secure link. Our database has lots of security measures to keep it safe.

Chapter 5

5. IMPLEMENTATION

5.1. Operating environment

The operating environment for Mazdoor Online includes the following:

- Operating system: Windows
- Platform: React Native
- Database: MongoDB

Timescales: It typically takes around 6 to 10 months to finish building this system.

Testing: We will conduct unit testing, integration testing, and system testing.

5.1.1. Operating system

Windows: User-friendly with advanced functionalities.

5.1.2. React-Native

React Native is a popular framework for making mobile apps using JavaScript and React. It allows developers to create apps for both iOS and Android platforms simultaneously, saving time and effort. React Native's codebase is reusable across platforms, enabling efficient development and maintenance. Its community support and extensive library of components contribute to its widespread adoption by developers worldwide. Notable apps like Facebook, Instagram, and Airbnb have been built using React Native, showcasing its capability for building high-performance mobile applications.

5.1.2.1. React-Native features

React Native stands out as a robust framework for crafting mobile apps, providing developers with the ability to create for both iOS and Android platforms. Noteworthy features include hot reloading, facilitating real-time code updates without requiring full recompilation. It emphasizes native performance, ensuring fluid animations and responsive UIs. React Native boasts a wide array of third-party plugins, enabling seamless integration of extra functionalities. Its approach to UI design simplifies complex interfaces through reusable components. Backed by an active community and thorough documentation, React Native is favored by developers aiming for efficient and adaptable mobile app development.

5.1.2.2. System requirements for React-Native

To develop with React Native, you'll need a system running macOS, Windows, or Linux, with Node.js version 12 LTS or newer, npm version 6.x or newer, JDK 8 or newer for Android development, Android Studio version 3.6 or above, Xcode 11 or newer for iOS development, Python 2.7.x for React Native versions 0.63 and below, and hardware with at least 8 GB of RAM, 10 GB of free disk space, and a multi-core processor. Regular updates to these components ensure access to the latest features and security patches for efficient development.

5.1.3. MongoDB Atlas

Database: We'll utilize MongoDB Atlas for our database needs, a cloud-based managed MongoDB service offering scalability and reliability.

5.1.4. Java Script

In this system, we'll utilize JavaScript, a versatile programming language renowned for its compatibility across various platforms and extensive ecosystem of libraries and frameworks.

5.1.4.1. Java-Script features

JavaScript, as utilized within the React Native framework, embodies a multitude of features crucial for modern mobile application development. It supports dynamic typing and flexible object-oriented programming paradigms, allowing for versatile and scalable code structures. With its asynchronous programming support through mechanisms like promises and `async/await`, JavaScript facilitates efficient event-driven programming crucial for responsive user interfaces. Additionally, JavaScript's extensive ecosystem of libraries and packages enhances developer productivity and enables seamless integration of various functionalities into React Native applications. Furthermore, JavaScript's cross-platform compatibility ensures that React Native applications can run seamlessly on both iOS and Android platforms. These features collectively empower developers to create highly performant and engaging mobile applications using the React Native framework.

5.1.4.2. System requirements for Java Script

Developing applications with JavaScript in React Native requires flexible yet crucial system requirements. Supported OS like Windows, macOS, or Linux is essential, along with modest hardware specs: a 64-bit processor, ample disk space, and a minimum of 4 GB RAM. Installing Node.js and npm is imperative for dependency management and building React Native apps. While IDE support like Visual Studio Code or IntelliJ IDEA enhances the experience, it's optional due to JavaScript's versatility. React Native's JavaScript is efficient across diverse hardware configurations, ensuring accessibility for developers.

Chapter 6

6. Software Testing:

6.1. Deriving test case specifications:

Deriving test case specifications involves meticulously analyzing requirements to identify testable scenarios, creating comprehensive test cases covering functional and non-functional aspects, defining test data, preconditions, expected results, and pass/fail criteria, prioritizing and organizing test cases for efficient execution, and regularly reviewing and updating test cases as requirements evolve to thoroughly validate the application's behavior and ensure compliance with specifications.

6.2. Testing Environment

6.2.1. Hardware Requirements

- **Memory:** 8 GB
- **OS:** Windows 11
- **CPU:** Intel Core i7 64 bit

6.2.2. Software Requirements

- Web browser
- Internet

6.2.3. Testing Identifications

We have devised a specific testing approach for each module. Initially, we compiled a list of all the modules that required testing, and subsequently, we crafted individual test cases for each module. Here are some of the primary test case:

User

- Signup module
- Login module
- User account module (Update the user information module)
- Search module
- Submit review module
- Sign out module

6.3. Testing procedure

A way of informing us that we will test each test case is the test procedure. Additionally, we keep track of the length of time it takes to test each module and the impact each test case has on each module.

6.4. Test Cases

6.4.1. Test Case for Sign up module

6.4.1.1. Sign up as a User

Tested By	M. Huzaifa Subhani
Tested Type	Unit Testing
Test case No.	01
Test case Name	Sign up testing
Test case description	This test case examines the user sign-up procedure to confirm that users can successfully set up accounts in the system. This test is designed to confirm that the sign-up process works as expected and that creating a user account is simple and error-free.
Item to be tested	
	Fill in the necessary boxes with accurate data, including your email address, password, and phone number.
	Confirm the password by re-entering it in the appropriate field.
	Click on the "Sign Up" button to initiate the account creation process. The system will now allow the user to log in.

Specification Input	Valid user data, including a special email address that isn't already linked to any other accounts, password, and phone number.
Expected Result	<p>The user sign-up procedure is the main emphasis of the test case, which verifies that the user may successfully establish an account without running into any problems or warning messages. It checks to see if the data given in the required fields are accepted appropriately and if the confirmation password is a match for the password that was submitted for successful password confirmation. Furthermore, the user's optional information should be accepted without any problems. The account creation process should begin immediately upon clicking the "Sign Up" button, without any delays or mistakes. After signing up, the user should be effortlessly transferred to the proper page without experiencing any difficulties or unusual behaviors.</p> <p>Finally, when using the newly established account credentials to log in, the login procedure should be successful and error-free, demonstrating that the account creation process was carried out as intended.</p>
Actual Output	<p>When a user enters information in a field that is needed, that information is accepted as long as the confirmation password matches the password that was entered. Additionally, the optional data provided by the user need to be accepted without any issues. After clicking the "Sign Up" button, the process of creating an account should start instantly and without any errors. The user is seamlessly transported to the appropriate page after signing up without encountering any issues or strange behavior.</p>

	Finally, the login process is successful and error-free when using the newly created account credentials. This will show that the account creation process was completed as planned.
--	--

Table 2

6.4.2. Test Case for Login Module

6.4.2.1. Log in as a User

Tested By	M. Huzaifa Subhani
Tested Type	Unit Testing
Test case No.	02
Test case Name	Login testing
Test case description	This test case examines the user login procedure and confirms that a user can access their account. This test is used to make sure that the login process works as intended and that users can safely access their accounts.
Item to be tested	
	Navigate to the application's login page.
	To access the user's account, provide the right email address and password
	To begin the login procedure, click the "Login" button.
	Verify that the user is successfully logged in and directed to the home page.

Specification Input	Valid user data, including email and password associated with the user's account.
Expected Result	<p>There shouldn't be any problems guiding the visitor to the login page. The user's account's connected username or email address should be approved without any mistakes.</p> <p>The user's account should accept the proper password without any issues.</p> <p>The login procedure should start immediately after clicking the "Login" button, without any glitches or delays.</p> <p>The user should be led to the correct page without any problems or unusual behavior following a successful login.</p> <p>Without running into any permission problems, the user should be able to carry out various tasks and travel between various pages within the application.</p>
Actual Output	Accessing the login page successfully, accurately entering the username or email address and password, starting the login process quickly after clicking the "Login" button, and receiving a redirect to the proper page following a successful login, the User is also able to move between different application pages and carry out a variety of operations without running into authorization problems.

Table 3

6.4.3. Test Case for User account module

6.4.3.1. Update user information

Tested By	Nouman Nasir
Tested Type	Unit Testing
Test case No.	03
Test case Name	User information updation testing
Test case description	To make sure that the changes are correctly stored and reflected in the system, this test case focuses on changing the user's account information. The goal of this test is to ensure that the account update capability works as anticipated and enables users to successfully edit their information.
Item to be tested	
	Go to the user profile or account settings page.
	Find the area or fields where user data, such as name, email, and phone may be updated.
	Change one or more user information fields with the needed and appropriate adjustments.
	To make the changes effective, click on the "Update" button.
	Check to see if the modifications were properly saved without any issues or warnings.
	Verify that the user account settings or profile page accurately displays the changed user information.

Specification Input	Changes that are legitimate and wanted are made to user information fields like name, email, phone, and address.
Expected Result	<p>The profile page or account settings should be easily accessible to the user.</p> <p>The user information fields must be editable and open to changes without any limitations or problems.</p> <p>Valid modifications to the user information fields should be accepted without any warnings or problems.</p> <p>The changes should be immediately applied after clicking the "Update" button, without any glitches or delays.</p> <p>The user information should be successfully stored whenever any modifications are made without any problems.</p> <p>The changed user information should appropriately appear on the user account settings or profile page, indicating that the update was successful.</p> <p>After the update, all pertinent parts and features in the program should appropriately use the updated user data.</p>
Actual Output	Users may modify user information fields without any limits or issues and can make legitimate changes to the needed fields without running into any problems or validation messages, after successfully navigating to the user account settings or profile page. The changes need to take effect right away after pressing the "Update" button, without any hiccups or mistakes. The database was also updated. The real result would be regarded as accurate if the user information changes were successfully stored

	without any problems or mistakes. The user account settings or profile page ought appropriately to reflect the modified data, indicating that the update was successful.
--	--

Table 4

6.4.4. Test Case for Search Module

Tested By	Nouman Nasir
Tested Type	Unit Testing
Test case No.	04
Test case Name	Search module testing
Test case description	This test case focuses on the functionality of the search module, particularly the search for cities and the societies they are affiliated with. The goal of this test is to ensure that the search module effectively locates cities and societies by properly retrieving and displaying pertinent results depending on user input.
Item to be tested	
	Find the search module.
	Select the city name in the search field.
	After selecting the city name, select the society
	Check to see if the search results provide pertinent information that matches the search term you typed.
Specification Input	Suitable city names for searches. Valid society names or search terms for the city you've chosen.

Expected Result	<p>The search module needs to be easily accessible to the user.</p> <p>The search box should allow a legitimate city name selection without any issues.</p> <p>The search area should accept your choice of a legitimate society name without any problems.</p> <p>Results have to be shown on the main screen by the supplied city and its society.</p>
Actual Output	<p>The search field needs to work properly, enabling the user to type a valid city name without running into any problems or errors.</p> <p>The input is acknowledged and correctly processed.</p> <p>The user's selection of an authorized society name also is accepted by the search area without any issues. The information is accepted and correctly processed for the user to search for certain societies in the chosen city.</p> <p>The search results, which provide pertinent data depending on the provided city and its society, are shown on the home screen. The outcomes must precisely correspond to the user's search criteria and show the related cities and societies as necessary.</p>

Table 5

6.4.5 Test Case for Submit review module

Tested By	Omama Shakeel
Tested Type	Unit Testing
Test case No.	05

Test case Name	Submit review module testing
Test case description	This test case focuses on the Submit Review module, in particular the features that let users post reviews, score societies using specified criteria, and upload images of their CNICs or utility bills to verify their addresses. This test aims to confirm that users are capable of correctly submitting reviews, rating societies fairly, and supplying the required supporting evidence for address verification.
Item to be tested	
	Access the application's Submit Review section.
	Find the review textbox where people may enter their reviews.
	In the text box, write a review on the society in which you share your opinions.
	Make that the review has been accepted without any hiccups or limitations.
	Look for the part on rating criteria, where people may assess the society using the criteria supplied (such as facilities, security, and cleanliness).
	By choosing suitable values for each parameter, rate the society.
	Verify sure there are no faults or limits on the rating values.
	Use the given file upload feature to locate the file upload section where customers can upload a photo of their CNIC or bill for address verification.

	<p>Check to see whether the file upload went through without any problems or errors.</p>
Specification Input	<p>Review content that is appropriate for expressing opinions about society.</p> <p>Amounts that are appropriate for each aspect based on the characteristics supplied (for example, cleanliness, security, and amenities).</p> <p>A true photocopy of the user's CNIC or utility bill for address confirmation.</p>
Expected Result	<p>The user ought to have no trouble using the Submit Review module.</p> <p>Users should be able to input their reviews into the review textbox without running into any problems or limitations.</p> <p>The entered review must be acknowledged and stored appropriately, without any changes or data loss.</p> <p>Users should be able to rate the society using the specified parameters in the rating parameter section without any problems or limitations.</p> <p>The chosen rating values have to be accepted and saved appropriately, without any changes or data loss.</p> <p>Users should be able to upload a photo of their CNIC or bill for address verification without any problems or limitations using the file upload area.</p> <p>Click on the "Submit" button to upload the review to the database without any error.</p> <p>Without any alterations or information loss, the uploaded photo must be precisely kept and linked to the particular review.</p>

Actual Output	<p>The user can input their review without running into any problems, and the review is stored correctly. The rating feature operates as intended, accepting and accurately storing the chosen rating values. The user has no trouble uploading their CNIC or bill photo using the file upload tool, assuring successful saving. By clicking on the “Submit” button, upload the review to the database without any error.</p> <p>All contributed information, including reviews, ratings, and uploaded images, must appropriately reflect the relevant society. Any variations from these anticipated results, such as submission errors or inaccurate data association, would point to potential flaws or problems with the Submit Review module.</p>
----------------------	--

Table 6

6.4.6 Test Case for Sign out module

Tested By	Omama Shakeel
Tested Type	Unit Testing
Test case No.	06
Test case Name	Sign out module testing
Test case description	In particular, the capability that enables users to log out of their accounts is the subject of this test case, which concentrates on the Sign Out module. This test checks to make sure users can log out of the application safely and successfully.

Item to be tested	
	Go to the side menu and click on the “Logout” Button.
	Verify that the user has been successfully logged out and is being sent to the login or home screen of the program.
Specification Input	Reliable user account information.
Expected Result	<p>Go to the side menu and click on the “Logout” button.</p> <p>The logout procedure should begin when the user clicks on the “Logout” button.</p> <p>Denying them access to sites and functionality that need authentication.</p> <p>After a successful sign-out, the user should be routed to the home screen.</p> <p>Access should be denied if the user tries to access any authenticated or restricted pages or features.</p> <p>When attempting to access restricted areas, the user should be required to log in once more, signifying a successful logout.</p>
Actual Output	The user is logging out safely and blocking access to restricted areas.

Table 7

Chapter 7

7. Project Display Screens

7.1 Mobile Application

7.1. Splash screen



Figure 7 1: Splash Screen

7.2. Get Started Screen

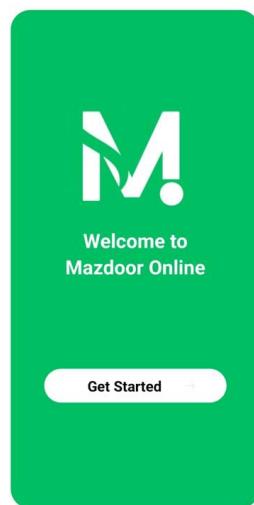


Figure 7 2: Get Started Screen

7.3. Main Screen



Figure 7 3: Main Screen

7.4. Labor Login Screen

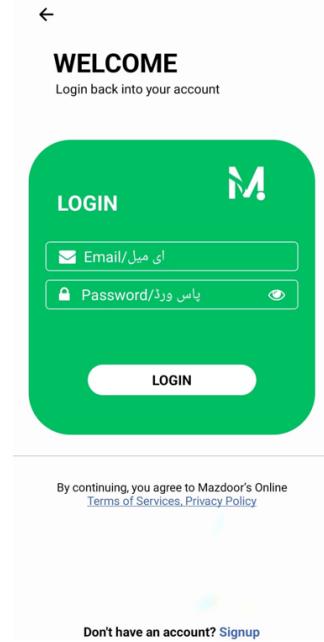


Figure 7 4: Labor Login Screen

7.5. Labor Signup Screen



Figure 7 5: Labor Signup Screen

7.6. Labor Identification

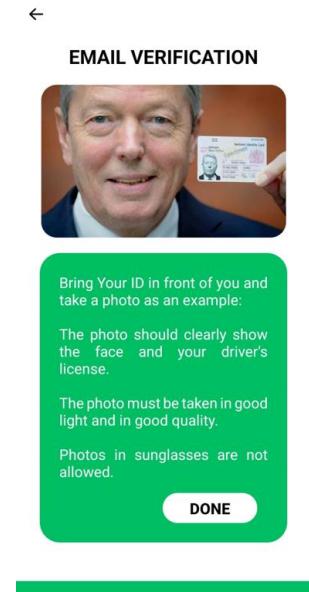


Figure 7 6: Labor Identification

7.7. Labor Choose Skill Screen



Figure 7 7: Labor Choose Skill Screen

7.8. Labor Work Proposal Screen

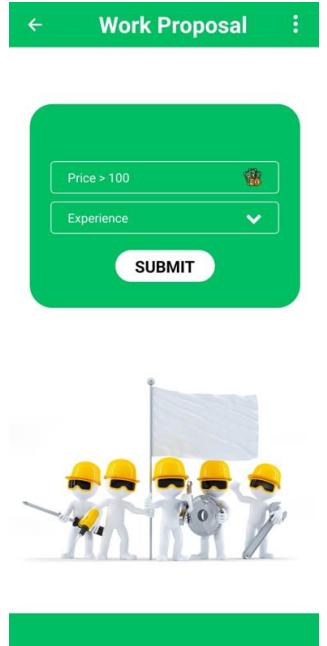


Figure 7 8: Labor Work Proposal Screen

7.9. Labor Profile Screen

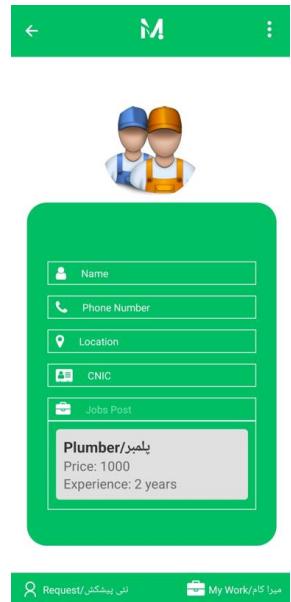


Figure 7 9: Labor Profile Screen

7.10. Labor Update Profile Screen

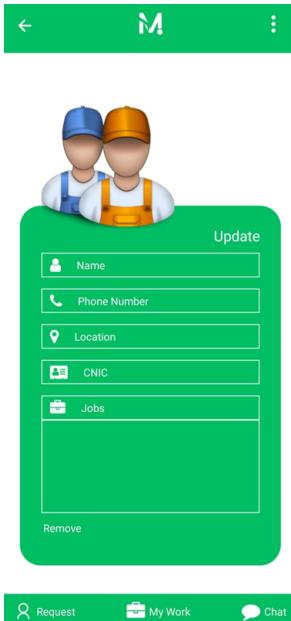


Figure 7 10: Labor Update Profile Screen

7.11. Pop up Screen

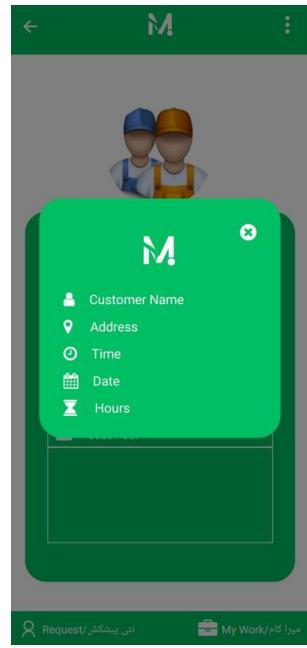


Figure 7 11: Pop up Screen

7.12. Labor Menu

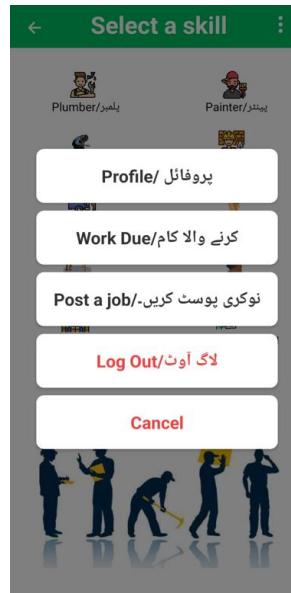


Figure 7 12: Labor Menu

7.13. Labor Offers

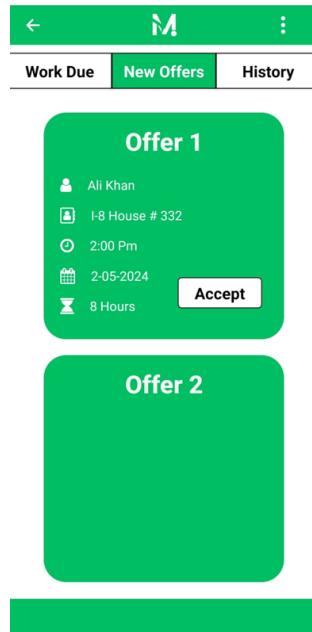


Figure 7 13: Labor Offers

7.14. Labor Work Due

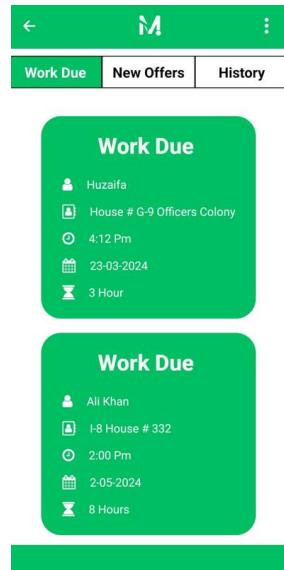


Figure 7 14: Labor Work Due

7.15. Labor History

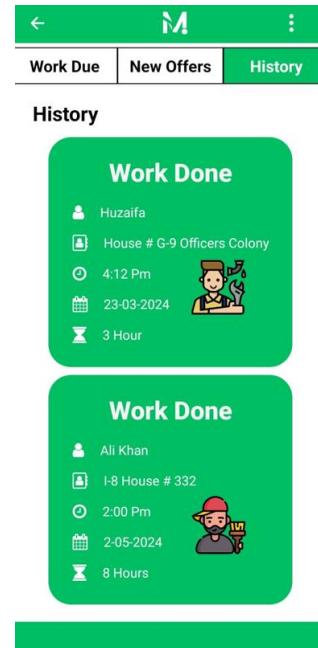


Figure 7 15: Labor History

7.16. Customer Home Screen

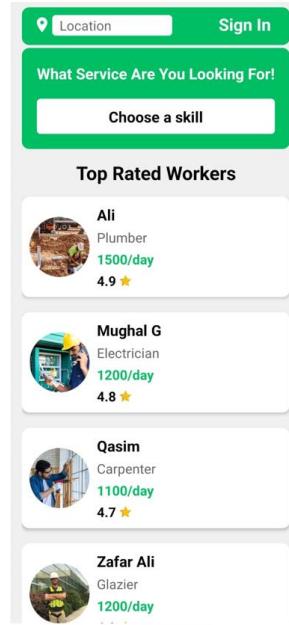


Figure 7 16: Customer Home Screen

7.17. Customer Login Screen

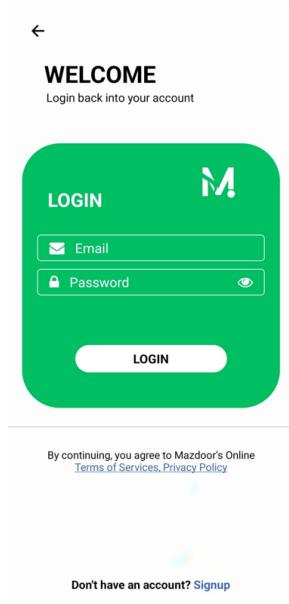


Figure 7 17: Customer Login Screen

7.18. Customer Signup Screen

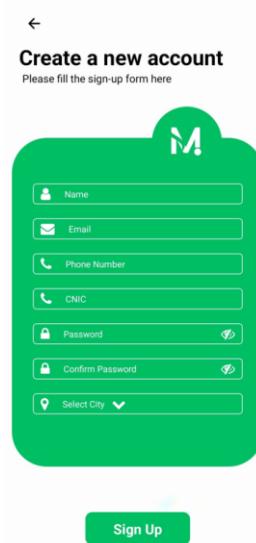


Figure 7 18: Customer Signup Screen

7.19. Customer Screen after Login



Figure 7 19: Customer Screen after Login

7.20. Customer Profile Screen

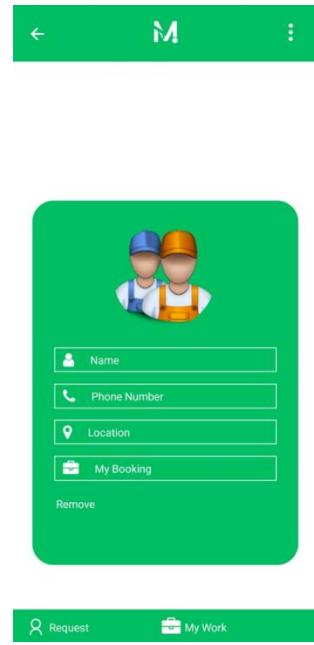


Figure 7 20: Customer Profile Screen

7.21. Customer Update Profile Screen

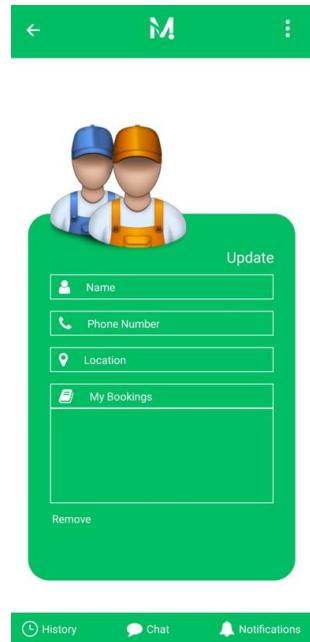


Figure 7 21: Customer Update Profile Screen

7.22. Customer Dropdown List

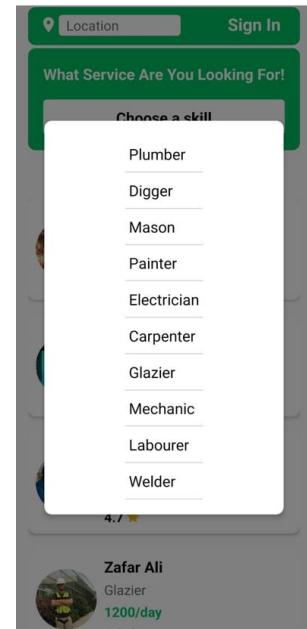


Figure 7 22: Customer Dropdown List

7.23. Customer Search Location

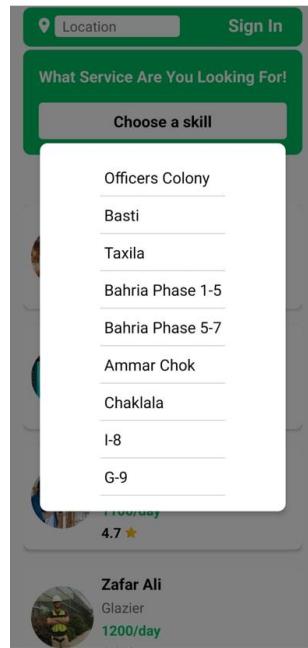


Figure 7 23: Customer Search Location

7.24. Customer Menu

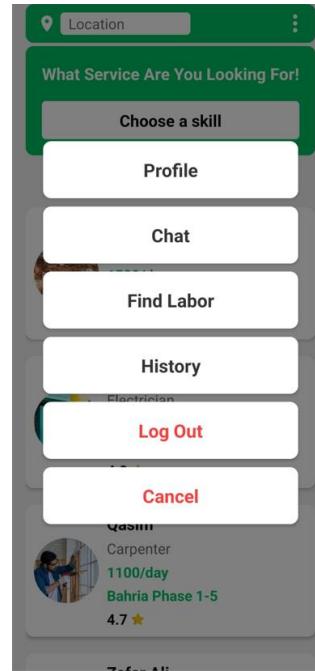


Figure 7 24: Customer Menu

7.25. Customer Book Labor

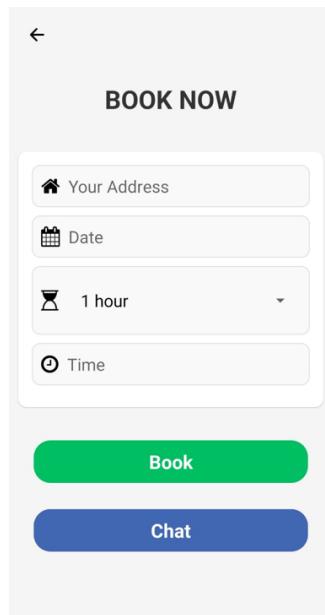


Figure 7 25: Customer Book Labor

7.26. Customer Chat



Figure 7 26: Customer Chat

7.27. Customer Rating to Labor

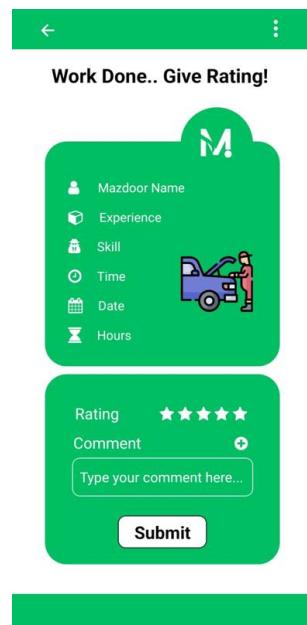


Figure 7 27: Customer Rating to Labor

7.2 References

- <https://reactnative.dev/docs/getting-started>
- <https://docs.atlas.mongodb.com/>
- <https://owasp.org/www-project-authentication-cheat-sheet/>
- <https://ieeexplore.ieee.org/document/8839394>