Shehroz Aziz

Akhri Aramgah

General

general12345

National University of Computer and Emerging Sciences, Islamabad

Document Details

Submission ID

trn:oid:::1:3217285324

Submission Date

Apr 15, 2025, 10:00 AM GMT+5

Download Date

Apr 15, 2025, 11:04 AM GMT+5

File Name

Akhri_Aramgah.pdf

File Size

3.1 MB

95 Pages

21,971 Words

121,413 Characters

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

National University of Computer and Emerging Sciences, Lahore

Akhri Aramgah

Shehroz Aziz............................21L-7521 BS(CS)

Muhammad Hammad...............21L-5388 BS(CS)

Muhammad Abdullah Asim.....21L-5365 BS(CS)

Supervisor: Dr. Rana Asif Rehman

Final Year Project

April 14, 2025

Author’s Declaration

This states Authors’ declaration that the work presented in the report is their own, and has not been submitted/presented previously to any other institution or organization.

Abstract

The process of finding a grave for the deceased has become a burdensome task due to the lack of burial space in big cities. Additionally, a grieving family has to make a lot of arrangements, so the process of death and burial becomes increasingly difficult. Akhri Aramgah aims to provide a convenient solution by developing an ecosystem of applications that facilitate the booking of graves and post-funeral services. This system connects the professionals with the users and enables the authorities to oversee the operations, providing better transparency. This system also provides smart features like AI-based facial recognition to search morgues for missing people and IoT-based possible damage detection to generate warnings.

Executive Summary

This report concerns the development of Akhri Aramgah, Our final-year project. The motivation behind this idea comes from the decentralized nature of funeral and burial services in our society. The hassle of booking a grave and relevant funeral services after the passing of a loved one is a cumbersome task. We aim to provide a centralized system of connected applications that will bring the users, graveyards and service providers to one platform which will make booking these services a walk in the park. Our smart algorithms will greatly facilitate the professionals associated with this field by providing them with a platform that will find convenient orders for them. The entire platform will be monitored by the admin side to ensure a safe and respectful experience for everyone. The main goal of this project is to build a highly scalable and standardized solution that will tend to these problems.

This sector faces a multitude of problems. The growing population will eventually cause burial space scarcity and finding a graveyard with empty burial space will become increasingly difficult. Our algorithms will not only see the nearest burial spaces but also map new graveyards better to solve the disoriented nature of graveyards. The professionals linked to this field also feel the absence of a platform that connects them with the nearest clients, our System will use location-based allocation of orders to these providers. There is also a complete lack of transparency in the matters of graveyard management, Our solution aims to significantly increase this transparency by enabling the admins to oversee the entire operation.

The project scope consists of users, admins, Gorkans (graveyard managers) service providers (transporters, caterers) and morgues, each having their application in one connected ecosystem. The user will have the option to book graves and funeral services. The Gorkan will be responsible for preparing and maintaining the booked graves. The third-party service providers will be allocated orders and they will complete these orders. The entire operation will be monitored by admins and they will also have the option to add new graveyards or view the detailed records of existing ones. The system will incorporate smart features like automated complaint escalation using Natural Language Processing (NLP), an Internet of Things (IoT) device that will monitor soil moisture levels to predict possible damage to graves, and an artificial intelligence model that will be used for facial recognition used in the identification of unidentified bodies in a morgue. Furthermore, the first Sustainable Development Goal (SDG) for this project is Industry, Innovation, and Infrastructure as this project promotes the use of technological innovations such as Natural Language Processing (NLP), Artificial Intelligence (AI), and the Internet of Things (IoT) to solve a prevalent issue. The second Sustainable Development Goal (SGD) is Sustainable Cities and Communities as our project seeks to improve graveyard management and enhance the transparency and efficiency of services provided to communities.

The constraints of this project are majorly the scammers that will try to abuse the system by registering as either users giving fake orders or as service providers that will deteriorate the user experience. However, we can minimize these by making sure all payment is done upfront and by not releasing the payments to service providers until after the job is completed. This will eliminate any motivation for scammers to register on our platform making it mostly safe. We will also implement a complaint system that could lead to the admin banning troublesome users and service providers from the platform.

OpusXenta has developed a solution in the form of a website that connects the service providers to the users that require these services. There are also FindAGrave and BillionGraves, these apps store information about various gravesites around the world. FindFace has developed an application that detects and matches face images across large databases. There are a plethora of applications that allow for digitized graveyard mapping and management such as PlotBox, Cemify, and Chronicle. Everplans offers a solution catered to the needs of people to set preferences about their burial and funeral services. IoT Agriculture Monitoring System developed an IoT system that measures factors like temperature and soil moisture to generate alerts if the conditions are suitable for plant growth in the area. Complaint Assistant developed a system that uses AI and NLP to understand and evaluate complaints before escalating them accordingly.

This report incorporates software requirement specifications and also lists all the features of the system, functionalities of each actor have also been discussed as well as the functional and non-functional requirements. All the assumptions that were taken are also highlighted in detail and all the use cases are elaborated. The database design is also shown to indicate the relationship between various entities. Risk Analysis is done to predict the potential risks of this project. This document provides a complete overview of the project including all constraints, goals, and objectives. The system architecture diagram and the component diagrams are also added to the document to show the detailed structure of the project. Other diagrams like class diagrams, sequence diagrams, and use case descriptions provide a better view of the high and low-level design of the system.

Table of Contents

List of Figures **xiii**

List of Tables **xiv**

1 Introduction **1**

1.1 Purpose of this Document 1

1.2 Intended Audience 1

1.3 Definitions, Acronyms, and Abbreviations 1

1.4 Conclusion 2

2 Project Vision **3**

2.1 Problem Domain Overview 3

2.2 Problem Statement 3

2.3 Problem Elaboration 3

2.4 Goals and Objectives 4

2.5 Project Scope 5

2.6 Sustainable Development Goal (SDG) 6

2.7 Constraints 6

2.8 Business Opportunity 6

2.9 Stakeholders Description 7

2.9.1 Stakeholders Summary 7

2.9.2 Key High-Level Goals and Problems of Stakeholders 7

2.10 Conclusion 7

3 Related Applications **9**

3.1 Definitions, Acronyms, and Abbreviations 9

3.2 Detailed Applications Review 9

3.2.1 OpusXenta - Arrangements and Bookings 9

3.2.2 PlotBox - Cemetery Management Software 10

3.2.3 Everplans - End-of-Life Planning Made Easy 10
3.2.4 Cemify - Simplifying Cemetery Management 11
3.2.5 Chronicle - Memorialization and Cemetery Management 11
3.2.6 Find A Grave - Millions of Cemetery Records 12
3.2.7 CemetryFind - Cloud Records and Mapping for Cemeteries 13
3.2.8 BillionGraves - The World's Largest Resource for Grave Records 13
3.2.9 FindFace Multi - A New Approach to Facial Recognition 14
3.2.10 Recognito - Face Biometrics & ID Verification Solutions 15
3.2.11 Complaint Assistant App 15
3.2.12 IoT Agriculture Monitoring System 16
3.3 Related Applications Summary Table 16
3.4 Conclusion 18

4 Software Requirement Specifications 19
4.1 List of Features 19
4.2 Functional Requirements 19
4.2.1 Functional requirements for users 19
4.2.2 Functional requirements for service providers 20
4.2.3 Functional requirements Gorkan 20
4.2.4 Functional requirements for Admins 20
4.2.5 Functional requirements for Morgue Staff 20
4.3 Quality Attributes 20
4.3.1 Reliability 21
4.3.2 Maintainability 21
4.3.3 Usability 21
4.3.4 Security 21
4.3.5 Responsiveness 21
4.4 Non-Functional Requirements 21
4.4.1 Performance 21
4.4.2 Independence of services 21
4.4.3 Separation of concerns 22
4.4.4 Security 22
4.4.5 Error handling 22
4.4.6 User friendly interface 22
4.5 Assumptions 22

4.5.1 Government backing . . . 22
4.5.2 Mobile literacy . . . 22
4.5.3 Internet accessibility . . . 22
4.5.4 Training availability . . . 22
4.5.5 Government Control . . . 23
4.6 Use Cases . . . 23
4.6.1 Login . . . 23
4.6.2 Search Reports . . . 23
4.6.3 Register . . . 24
4.6.4 Book Grave . . . 24
4.6.5 Book Cabin in Morgue . . . 25
4.6.6 Upload Unidentified Body Image . . . 25
4.6.7 Apply as Service Provider . . . 26
4.6.8 Manage Complaints . . . 26
4.6.9 Add New Service Provider . . . 27
4.6.10 Manage Graveyards . . . 27
4.6.11 Request Service from Caterer . . . 28
4.6.12 Request Service from Transporter . . . 28
4.6.13 Respond to Grave Order . . . 29
4.6.14 Generate Reports . . . 29
4.6.15 Add New Graveyard . . . 29
4.6.16 Request Grave Maintenance . . . 30
4.6.17 Search Missing Person . . . 30
4.6.18 Manage Service Providers . . . 31
4.6.19 Update Grave Status . . . 31
4.6.20 Respond to Maintenance Request . . . 31
4.7 Hardware and Software Requirements . . . 32
4.7.1 Hardware Requirements . . . 32
4.7.2 Software Requirements . . . 32
4.8 Graphical User Interface . . . 32
4.9 Database Design . . . 37
4.9.1 ER Diagram . . . 37
4.9.2 Data Dictionary . . . 38
4.10 Risk Analysis . . . 39

4.11 Conclusion 40

5 High-Level and Low-Level Design 41

5.1 System Overview 41

5.2 Design Considerations 41

5.2.1 Assumptions and Dependencies 41

5.2.2 General Constraints 42

5.2.3 Goals and Guidelines 42

5.2.4 Development Methods 43

5.3 System Architecture 43

5.3.1 Architecture Diagram 44

5.3.2 Component Diagram 45

5.3.3 Subsystem Architecture 46

5.4 Architectural Strategies 46

5.4.1 Programming Language 47

5.4.2 Database Design 47

5.4.3 Product Enhancement and Extensibility 47

5.4.4 User-Centric Interface Design 47

5.5 Domain Model/Class Diagram 47

5.6 Policies and Tactics 49

5.6.1 Coding Guidelines and Conventions 49

5.6.2 Testing Plans 49

5.6.3 Algorithm Selection 49

5.6.4 Maintenance Plans 49

5.7 Conclusion 49

6 Implemenation and Test Cases 50

6.1 Implementation 50

6.1.1 Implementation of User Authentication 50

6.1.2 Implementation of User Web and Mobile Application 50

6.1.3 Implementation of Service Providers' Mobile Apps 51

6.1.4 Implementation of Map Integration 51

6.1.5 Implementation of Admin Side Functionalities 51

6.1.6 Implementation of Graveyard Mapping 51

6.1.7 Implementation of 360-Degree Graveyard View 52

6.1.8 Implementation of Facial Recognition (Shanakht Gar) 52
6.1.9 Integration with Morgue Service Provider Mobile Application 52
6.1.10 Implementation of NLP Complaint Escalation (Shikayat Nazim) 53
6.1.11 Implementaion of Soil Moisture Sensor Module (Khaak Been) 53
6.1.12 Implementation Overview 53
6.2 Test Cases 54
6.3 Test Metrics 69

7 User Manual 70

7.1 User 70
7.1.1 Sign Up 70
7.1.2 Sign In 70
7.1.3 Book Grave 70
7.1.4 Book Morgue 71
7.1.5 Book Catering 71
7.1.6 Book Transportation 71
7.1.7 Search Morgue 71
7.1.8 View Order History 72
7.1.9 View Graves 72
7.1.10 Schedule Grave Maintenance 72
7.1.11 Register Complaint 72
7.1.12 Track Complaints 72
7.1.13 Update Settings 73
7.2 Gorkan 73
7.2.1 Sign In 73
7.2.2 View Graveyard 73
7.2.3 Prepare Grave Confirmation 73
7.2.4 Grave Maintenance Confirmation 73
7.3 Morgue Manager 74
7.3.1 Sign In 74
7.3.2 Prepare Cabin Confirmation 74
7.3.3 Upload Image 74
7.3.4 Delete Image 74
7.4 Catering Service Provider 74
7.4.1 Sign In 74

7.4.2 Book Order 75
7.4.3 Complete Order 75
7.4.4 Change Details 75
7.5 Transportation Service Provider 75
7.5.1 Sign In 75
7.5.2 Book Order 76
7.5.3 Complete Order 76
7.5.4 Change Details 76
7.6 Admin 76
7.6.1 Add Graveyard 76
7.6.2 Add Morgue 77
7.6.3 Review Complaints 77
7.6.4 Edit Graveyard 77
7.6.5 Search Graves 77
7.6.6 Add Service Providers 78
7.6.7 Ban/Warn Service Providers 78
7.6.8 View Analytics 78

8 Conclusion 79

List of Figures

2.1 Figure Illustrating the Sustainable Development Goals Addressed by the Project . . 6

4.1 Graphical Representation of the Website's Landing Page Interface 33
4.2 Graphical Representation of the Admin Dashboard Webpage Interface 34
4.3 Graphical Representation of the User Sign-In Webpage Interface 34
4.4 Graphical Representation of the Book Grave Module in the Mobile Application . . . 35
4.5 Graphical Representation of the Gorkan Mobile Application Interface 35
4.6 Graphical Representation of the Service Providers Application Interface 36
4.7 Entity Relationship Diagram Representing the Project's Data Model 37

5.1 Architecture Diagram Illustrating the Project's Structural Design 44
5.2 Component Diagram Showing System Architecture and Interactions 45
5.3 Class Diagram Depicting the Structure and Relationships of the Project's Classes . 48

List of Tables

3.1 Table Summarizing the Key Features, Relevance to Our Project, and Limitations of Related Applications 17

4.1 Table Displaying the Project's Data Dictionary Details 38

6.1 Test Case Metrics 69

Chapter 1 Introduction

The decentralized nature of burial and funeral services in the country makes the process of acquiring these services a huge problem. Moreover, this absence of a platform affects the service providers in this field as there is a lack of connectivity between these service providers and the users. Our system aims to bridge this gap by offering an ecosystem of connected applications that will allow the users to book these services with ease as well as provide better opportunities for the linked professionals.

There is also a lack of transparency that negatively affects the management of burial grounds in the country. Our solution would also assist the authorities in keeping a better check and balance on things and significantly boost transparency. The payment of dues through the application will eliminate any risk of low-level corruption.

1.1 Purpose of this Document

This document provided a complete overview of Akhri Aramgah. It will cover each aspect of the project in great detail highlighting the strengths and the weaknesses of the project and giving the readers an accurate idea about the implementation of the system through proper documentation featuring UML standard diagrams.

1.2 Intended Audience

The intended audience of this project is the users who struggle to arrange funeral and burial services. Additionally, the service providers related to this field feel that offering their services on a dedicated and centralized platform can help boost their income. Lastly, the authorities would prefer a digital platform to oversee all operations regarding burial grounds and services with increased transparency.

1.3 Definitions, Acronyms, and Abbreviations

UML: Unified Modeling Language

SDG: Sustainable Development Goal

FYP: Final Year Project

NLP: Natural Language Processing

IoT: Internet of Things

AI: Artificial Intelligence

GUI: Graphical User Interface

OTP: One Time Password

RAM: Random Access Memory

JDK: Java Development Kit

IDE: Integrated Development Environment

ERD: Entity Relationship Diagram

API: Application Programming Interface

SQL: Structured Query Language

DOM: Document Object Model

MVC: Model View Controller

KYC: Know Your Customer

JSON: JavaScript Object Notation

JWT: JSON Web Token

CNN: Convolutional Neural Network

GitHub: Platform that allows developers to share their code and projects with the world.

Gorkan: Graveyard manager responsible for the administration and operational oversight of a cemetery.

Shikayat Nazim: NLP-based Complaint Escalation Module used in Akhri Aramgah.

Shanakht Gar: AI-based Facial Recognition Module used in Akhri Aramgah for identity verification.

Khaak Been: IoT-based Soil Moisture Monitoring Module for assessing grave soil conditions.

IoT-based: Technology that connects devices to internet to communicate and share data for automation.

Arduino-based: A device built in Arduino platform and involves an Arduino microcontroller board.

AI-based: Technology that utilizes artificial intelligence to perform tasks.

Location-based: System that utilizes real-time location to deliver efficient services.

NoSQL: Non-relational (non-tabular) database that allows flexible data models and enables scalability.

1.4 Conclusion

In conclusion, Akhri Aramgah focuses on providing a solution that caters to the needs of people who require and provide funeral services while providing the authorities with adequate control over the system to oversee things efficiently. This chapter provides a brief overview of the motivation behind the project and the goals that are being targeted, as well as the audience that such an application would appeal to.

Chapter 2 Project Vision

To understand this system and its functionalities clearly, it is imperative to consider the motivation behind this idea. Starting from the problem domain and moving on to other key factors like scope, goals, and constraints, this chapter will be oriented in such a way as to help the reader develop a clear image of the project vision.

2.1 Problem Domain Overview

Our project targets the development of a system that will digitalize the burial and funeral services sector. There is not a single application in the market that provides the option of booking these services and as a result, the family of the deceased has to go through a long and tedious process of finding a burial space and booking the funeral services manually. Furthermore, The available spaces in burial grounds are rapidly decreasing and due to lack of connectivity people are not aware of new burial grounds, our system would use location-based algorithms to provide the user with a list of nearby graveyards with available spaces as well as map new graveyards efficiently to prevent wastage of space.
The service providers also suffer the absence of a system that would connect them with the people who require their services. Our system is intended to not only connect them with clients but also assign them orders that would be more convenient for them thanks to location-based algorithms. There is a huge lack of accountability in the management of graveyards due to lack of transparency. Our system is meant to take care of this problem by enabling the authorities to oversee everything that is going on and giving them admin controls.

2.2 Problem Statement

The absence of a centralized platform for funeral and burial services affects the people who require these services and the relevant service providers, as well as the authorities that struggle to oversee the day-to-day operations of this sector.

2.3 Problem Elaboration

In the era of digitalization where all kinds of services and domains have been given an online presence, the burial and funeral services sector has been constantly overlooked. Due to the lack of a digital solution, the process of finding burial space and booking funeral services has become a challenge, especially in this era where all the old graveyards are filled.

The family of the deceased has to make a lot of manual effort to arrange the necessary services. Moreover, The problems not only extend to the users but also to the professionals that provide these services as they do not have any dedicated platform that directly connects them to the clients. This lack of connectivity produces a disparity where some service providers get all the orders while the rest hardly get any orders. The authorities have often complained about the transparency of transactions and activities in many fields and this field isn't any different. The staff employed to take care of these burial grounds have enjoyed no checks and balances and as a result, you can often see mismanagement on their behalf and the authorities failing to do anything about it.

2.4 Goals and Objectives

Akhri Aramgah has the following goals and objectives:

- Digitalize the burial and funeral services sector by bringing all relevant parties to one platform and leveraging the latest technologies.
- To make booking services related to the funeral and burial sector easier.
- Facilitating the service providers associated with this field by providing them with a platform to get more work.
- Support authorities for better management and control of burial grounds and services.
- The development of an easy and user-friendly interface to facilitate all the various kinds of users of the platform.
- To leverage the use of location-based algorithms to help the users find the nearest burial spaces and to help the service providers find convenient orders.
- Deployment of IoT-based devices that will monitor soil moisture levels to generate alerts regarding possible damage to the graves.
- Utilization of NLP to sort user complaints into critical or non-critical to help the admins respond to more critical complaints first.
- Identification of unidentified dead bodies in the morgue through the use of AI-based facial recognition.
- Data collection and Excel generation to provide datasets for possible research work regarding this sector.

2.5 Project Scope

This project can be implemented and deployed at the city or state level as this solves the prevalent issue of mismanagement in the sector of burial and funeral services. The system will be composed of a series of connected web and mobile apps that will each provide their respective user connectivity to the entire system. Akhri Aramgah will have the following applications:

- A Mobile-based and Web-based application that will allow the users to find and book burial spaces, and management for the graves and funeral services for their deceased.
- The Android-based applications for the service providers where will be assigned orders using location-based algorithms to make it more convenient.
- The Android-based application for the Gorkan (graveyard manager) where the grave orders and maintenance orders will be assigned to him.
- The Android-based applications for the morgue where the worker of the morgue will respond to cabin booking requests and upload images of unidentified bodies for facial recognition.
- A Web-based application for the admin where features to manage the entire system will be provided.

Mobile-based and Web-based applications are to be built in React-Native and React respectively with Node.js as the backend of applications. Android-based applications are to be built using Java and XML. Some technical modules will also be part of the system.

- The use of IoT devices (Arduino-based) to monitor soil moisture levels to detect possible damage to graves.
- AI will be used for facial recognition to help identify unidentified dead bodies.
- NLP will be used to sort the user complaints into critical and non-critical so that attention can be paid to the critical complaints first.

We believe that such a project has the potential to digitalize an entire sector of society and positively impact all the people who are associated with this field. Implementation of such a project on the government level can lead to significant improvements in efficiency, transparency, and accessibility of services. The project will be developed using the principles of the waterfall model and some practices of agile methodology will also be incorporated.

2.6 Sustainable Development Goal (SDG)

The main aim of our project is to facilitate the various people associated with the field of funeral and burial services. Whether it is the user who requires burial/funeral services or the service provider who makes a livelihood providing these services, Our application provides ease to them both. So the main sustainable development goal is **”Sustainable Cities and Communities”**.
We also plan to use highly innovative features such as AI, NLP and IoT to provide special services like detecting damages to graves or automated sorting of complaints. So the second sustainable development goal is **”Industry, Innovation, and Infrastructure”**.

9 INDUSTRY, INNOVATION AND INFRASTRUCTURE
11 SUSTAINABLE CITIES AND COMMUNITIES

Figure 2.1: Figure Illustrating the Sustainable Development Goals Addressed by the Project

2.7 Constraints

Akhri Aramgah has the following constraints:
Time: All the modules of this project and their documentation must be completed within 1 year.
Dataset: No dataset for facial recognition when eyes are closed is present, so it must be made manually.

2.8 Business Opportunity

While the main purpose of Akhri Aramgah is to provide ease to the public, there are several financial benefits this project will provide. Providing one platform to connect service providers with clients will help boost the sales of these service providers and the location-based smart algorithms will calculate the most feasible orders for each provider, further increasing his profits.
The government as well will reap the financial benefits as a centralized system will help eliminate corruption, thus eventually saving money.

2.9 Stakeholders Description

The stakeholders of Akhri Aramgah would be users, service providers, morgue staff, admin, and Gorkan (graveyard manager).

2.9.1 Stakeholders Summary

2.9.1.1 Users

The public that requires funeral and burial services will be able to book these services using our app.

2.9.1.2 Service Provides

The transporters and caterers will offer their services on our platform.

2.9.1.3 Morgue Staff

The staff of mortuaries will be responsible for responding to user requests.

2.9.1.4 Admin

The officials representing the authorities that will use the app to manage all affairs.

2.9.1.5 Gorkan

The graveyard manager will use the app to respond to user requests and prepare graves.

2.9.2 Key High-Level Goals and Problems of Stakeholders

The major high-level goal of Akhri Aramgah is to make the process of acquiring burial and funeral services simpler while also promoting transparency, which was traditionally lacking in this field. Aiming to reduce the manual effort it took to arrange these services is central to our system. The biggest challenge that the stakeholders will face is the fact that they are not accustomed to such a system and in a sector that is traditionally managed offline this is a very big leap. The introduction of highly technical features like AI, NLP or IoT will require special training. Putting their trust in a centralized digital platform is a big challenge for the stakeholders.

2.10 Conclusion

Akhri Aramgah aims to impact the funeral and burial services sector by bringing digitalization to an otherwise totally manual field. It will ease the process of booking burial and relevant services. The project will be the first of its kind and will serve as a blueprint for future research and development work in this field. The use of AI, NLP and IoT will set the roots of new technological advancements in this sector. This chapter was an attempt to explain the clear vision of the project to the readers by shining a light on the key factors like goals, problems, and constraints.

Chapter 3 Related Applications

Our area of work is extremely unique. It is so unique that there is not a single working application with a backend out there that has the features that Akhri Aramgah will provide. That being said, there are some loosely related projects and applications that offer some of the features that our system intends to provide.

3.1 Definitions, Acronyms, and Abbreviations

AI: Artificial Intelligence
NLP: Natural Language Processing
IoT: Internet of Things
GPS: Global Positioning System.
KYC: Know Your Customer
GitHub: Platform that allows developers to share their code and projects with the world.
IoT-based: Technology that connects devices to internet to communicate and share data for automation.
AI-based: Technology that utilizes artificial intelligence to perform tasks.

3.2 Detailed Applications Review

3.2.1 OpusXenta - Arrangements and Bookings

3.2.1.1 Summary

OpusXenta provides a solution to the burial services sector by connecting relevant businesses to customers. Instead of an application [1] merely provides a website that connects customers with service providers acting like a mediator and not as the solution provider.

3.2.1.2 Critical Analysis

The system in [1] lacks any kind of detailed and tailored solution. It fails to truly bring together the various stakeholders of this industry to create a useful and centralized platform. It also lacks any kind of security and fraud prevention mechanism, nor does it provide the functionality to make online payments to the user. It is essentially just a piece of paper in this modern digital world.

3.2.1.3 Relationship to the project

The only similarity between our project and the system in [1] is the fact that they both target the funeral service industry and aim to bridge the gap between the clients and the service providers. Aside from that, our project is a completely tailored solution that provides not only an application but an ecosystem of connected applications that make the platform and its scope much bigger than the system [1] is providing.

3.2.2 PlotBox - Cemetery Management Software

3.2.2.1 Summary

PlotBox [2] is a graveyard management tool that allows graveyard managers to manage burial sites and their records. It allows the user to schedule burials, uses cloud storage for record-keeping and provides options for the overall maintenance of graveyards.

3.2.2.2 Critical Analysis

The solution provided by [2] is targeted only to the graveyard managers and does not allow the general users to book services or graves. Furthermore, it does not use modern software technology like AI or IoT. The overall scope of this application is very limited.

3.2.2.3 Relationship to the project

While there are similarities between one module of Akhri Armagah and the system developed in [2], the overall scope of both these applications differs greatly. The use of highly advanced features such as AI, NLP, and IoT as well as the centralized nature of our system makes it a much better alternative.

3.2.3 Everplans - End-of-Life Planning Made Easy

3.2.3.1 Summary

Everplans [3] is a complete-centralized solution that allows users to plan for the end of their lives by providing a platform for them to store their wills and other important documents, as well as make burial plans and preferences. It ensures that all relevant information remains accessible to the family members.

3.2.3.2 Critical Analysis

The system discussed in [3] is a comprehensive and centralized solution for end-of-life planning and ensuring that all the important details and documents remain accessible to the family members after a person passes away, and it also allows the user to set burial preferences. However, this application is majorly focused on preserving the documents and lacks any features that allow the user to book any of the services, as it allows only the expression of preferences and not any actual services.

3.2.3.3 Relationship to the project

Everplans [3] is only lightly linked to our system as they both share some elements of burial preferences but the scope of both these projects are largely disconnected with this system focusing on things like document preservations and wills while our system focuses on providing a centralized digital solution to the entire sector of funeral and burial services.

3.2.4 Cemify - Simplifying Cemetery Management

3.2.4.1 Summary

Cemify [4] is a graveyard management software that focuses on graveyard mapping and operations. It provides the graveyard managers with all the tools they need to digitize the entire management of their graveyards by adding features like cemetery mapping, grave status tracking, and maintenance scheduling. It provided a great boost to the productivity of graveyard operations.

3.2.4.2 Critical Analysis

The system discussed in [4] is adept at streamlining the everyday operations of a graveyard manager or staff but it lacks any features to extend these services to the general users.
The system only caters to a very small scope and lacks any innovation or technical features. It lacks any room for growth and is designed only for administrative purposes.

3.2.4.3 Relationship to the project

The cemetery mapping and graveyard management features of Cemify [4] resonate with the features of our application, and it is a great administrative tool, but the scope of our system is far greater than just administration as it is a centralized platform that caters to the needs of an entire sector of society and the use of innovative features like AI, IoT and NLP set the project apart.

3.2.5 Chronicle - Memorialization and Cemetery Management

3.2.5.1 Summary

Chronicle [5] is an online graveyard management and mapping system that uses drones and GPS to map graveyards in an optimized way. The use of highly advanced tools for mapping and an optimized way of managing graveyard tasks are the primary selling points of this application, making it truly special.

3.2.5.2 Critical Analysis

The system discussed in [5] possesses a great method of graveyard mapping that is yet to be seen elsewhere. The use of GPS and drones sets it apart due to the use of highly technical equipment and makes this application truly innovative. However, as with most of the available applications, it is limited in its scope to only cater to the mapping and other tasks of the graveyard and has no features that help the general user.

3.2.5.3 Relationship to the project

The system [5] focuses primarily on the graveyard mapping task and other managerial tasks similar to parts of our system. The use of innovative technology in this sector is also a good similarity between the two applications, even if the technologies in question are not related. The difference however is the fact that this application is just merely a mapping and management tool, while our application is a completed catered solution that just has a scope much larger than just management.

3.2.6 Find A Grave - Millions of Cemetery Records

3.2.6.1 Summary

The system in [6] is a record-keeping web application. The purpose of this application is to keep track of the information about the people buried in graves around the world. Its primary focus is preserving the memory of notable individuals by keeping detailed information about their burial sites.

3.2.6.2 Critical Analysis

The system in [6] keeps track of only notable personalities and their grave sites and lacks accuracy. It provides no solution to the funeral service industry or the public. It is just a memorial site for famous people and where they are buried around the world.

3.2.6.3 Relationship to the project

The only similarity between our project and the system in [6] is the record keeping of graves. Our system intends to do this for all the people that use our platform, while this application only maintains records of famous people. Our project is a complete solution that brings together all the stakeholders of this sector as well as keeps records of all the graves, making our scope much bigger than this application.

3.2.7 CemetryFind - Cloud Records and Mapping for Cemeteries

3.2.7.1 Summary

CemetryFind [7] is a cloud-based system, designed for record management and mapping. It can be used on different operating systems and devices. Its purpose is to do cemetery scanning for records and maps.

3.2.7.2 Critical Analysis

This system in [7] is only designed for record keeping and data management, and lacks any kind of burial or post-funeral service. The data displayed is often inaccurate, and therefore the credibility of the system is controversial.

3.2.7.3 Relationship to the project

There is similarity between CemeteryFind [7] and our project in the sense that both keep track of data of graves but there is a very key difference as the purpose of the data collection in [7] is to allow users to search for graves but the data collected in our application is for the record keeping of the authorities and is not available to public.

3.2.8 BillionGraves - The World's Largest Resource for Grave Records

3.2.8.1 Summary

The system in [8] is also a record-keeping system but it also has a feature of extracting data from a headstone of the grave before storing it, this makes this application's scope that much more big. It depends upon volunteers to capture images of headstones of graves and their location and feed to their application for record keeping.

3.2.8.2 Critical Analysis

The system in [8] keeps track of graves through headstones, but most of the old graves do not have headstones or the text on them has faded away. It provides no solution to the funeral service industry or the public. It is heavily reliant on volunteers, making it a bit too optimistic especially since there is no reward for the volunteers. Furthermore, it asks for payment first without really providing any real service.

3.2.8.3 Relationship to the project

There are a few similarities between our project and the application in [8] as record keeping is also part of our system, but it is only a small part of our system, and it will be automated from the user input while booking a grave, but this application uses optical character recognition and digital image processing to gather data. While our project also makes use of such technologies, they serve a different and more meaningful purpose. Despite some technological and domain similarities, this project is quite different from our project on the whole.

3.2.9 FindFace Multi - A New Approach to Facial Recognition

3.2.9.1 Summary

FindFace [9] is an advanced tool for facial recognition developed by NtechLab that was originally intended to be a social media identification tool, but it has been since adopted into professional applications in the field of security. The system is capable of analyzing faces and searching them in extremely large databases, making it critical for solving criminal cases.
FindFace [9] is equipped with the latest features that allow it to operate in real-time and match faces within seconds. It can handle millions of comparisons simultaneously, making it suitable for large database searches. The system has been deployed in a wide variety of settings, including public transportation hubs, law enforcement agencies, and private security firms.

3.2.9.2 Critical Analysis

FindFace [9] uses techniques of deep learning and neural networks. The picture of the face is converted into a feature map and features such as eyes, nose, and jawline are used to calculate similarity with the other faces in the database. The first step is upscaling and enhancing the image using the techniques of digital image processing. It uses cloud-based architectures to ensure maximum speed. This application is extremely efficient and therefore used all over the world.

3.2.9.3 Relationship to the project

While both FindFace [9] and our project utilize AI-based facial recognition for identification purposes, the major difference is the fact that we aim to match a person's face with eyes open to the image of a dead body with eyes closed. The most distinct feature of the face is the eyes and with that out of the question we have to utilize different strategies than the one FindFace [9] used another difference is the size of the database that this application [9] has to search in millions so it must have algorithms catering to this while our application does not expect that many images as data.

3.2.10 Recognito - Face Biometrics & ID Verification Solutions

3.2.10.1 Summary

Recognito [10] is an online platform for facial recognition and ID verification. The purpose of this system is to do electronic-KYC, ID verification, online banking, etc. Recognito [10] is an NIST-certified system, that runs offline and provides complete data control and privacy.

3.2.10.2 Critical Analysis

The system in [10] is perfect for its purpose that is working on a video-based input to recognize a person's face and yields perfect results and that is reflected by the fact that this system is widely adopted by organizations worldwide.

3.2.10.3 Relationship to Project

There is similarity between one of our modules that features facial recognition of dead bodies, but there is a key difference as [10] does not use AI, rather it works solely on image processing.

3.2.11 Complaint Assistant App

3.2.11.1 Summary

The Complaint Assistant App [11] is an application on GitHub that uses AI and NLP to streamline the process of customer support work by understanding and identifying the critical score of a complaint and then assigning a priority to it. This system allows the customer support team to respond to critical requests faster. This system also suggests the possible response to user complaints, making it easier for agents to respond.

3.2.11.2 Critical Analysis

The system discussed in [11] is a great step in the field of NLP by making the whole process of complaint resolution streamlined and reducing the response times by a huge factor. However, the system is still somewhat inaccurate in its judgment and is capable of making plenty of mistakes that can potentially harm the user experience. The success of such a system depends on the availability of the training data and the hardware.

3.2.11.3 Relationship to the project

There is a direct similarity between this system [11] and the complaint module in our system, as both of these systems use AI and NLP to understand and evaluate the criticality of the complaint before escalating it further. The main difference is that the sole focus of this system is on NLP and complaint escalation, while our project only has this as a single module and has a scope that extends far beyond the scope of this application.

3.2.12 IoT Agriculture Monitoring System

3.2.12.1 Summary

The IoT Agriculture Monitoring System [12] monitors various factors that play a key role in the development of a plant such as soil moisture and temperature. It generates notifications and communicates them to the server when it feels that the conditions are suboptimal for plant growth. The system can also control water pumps to water the plants when the moisture level is low.

3.2.12.2 Critical Analysis

The system discussed in [12] is a great tool to monitor and improve the agriculture sector through the use of advanced technology. However, the system is limited by many factors, such as lack of quality internet connections and the accuracy of the sensors used. The major challenge that the system faces is that different plant species require different conditions to thrive, and this limits the practicality of this system.

3.2.12.3 Relationship to the project

The system [12] is similar to our project in the sense that there is some overlap between our IoT module and their soil moisture sensor module, as both these modules generate alerts when the moisture level reaches a certain level. The difference is the context in which both these systems are used. The IoT system in this application is used to monitor the ideal conditions for plant growth, while our system uses it to predict structural damages to the graves.

3.3 Related Applications Summary Table

Table 3.1: Table Summarizing the Key Features, Relevance to Our Project, and Limitations of Related Applications

Application	**Features**	**Relevance**	**Limitations**
OpusXenta [1]	Digital Presence and connectivity	Provides connectivity between service providers and clients of the funeral industry, similar to our project.	Fails to provide a proper centralized solution and merely connects the users with service providers without providing a platform.
PlotBox [2]	Graveyard management and record keeping	Provides graveyard management services like our application.	Lacks the use of modern technology and user-oriented services.
Everplans [3]	Funeral and Burial preferences	Allows the user to set funeral and burial preferences somewhat resonating with our system.	Provides only the option to set these preferences and not any actual services.
Cemify [4]	Graveyard management and mapping	Provided dynamic graveyard mapping and management services to the graveyard managers.	Provides only administrative features and lacks any user-oriented services.
Chronicle [5]	Smart Graveyard mapping and management	Provides accurate graveyard mapping through the use of technology and adds simple management features.	Provides only administrative features and mapping while lacking any user-oriented services.
FindAGrave [6]	Record keeping	Maintains the records of gravesites and allows users to search graveyards for any grave.	Only keeps track of the graves of famous personalities and does not have any data about commoners.
CemeteryFind [7]	Graveyard Record and mapping management	keeps the record of graves in graveyards and their location.	Lacks any burial or post-funeral services and data inaccuracy.

Application	**Features**	**Relevance**	**Limitations**
BillionGraves [8]	Record keeping and Data collection	Maintains the records of graveyards and allows users to upload data about graveyards and each grave in it.	It has Limited accuracy and relies on volunteers to collect data.
FindFace [9]	Facial Recognition	Provides an accurate and efficient solution for implementing AI-based facial recognition across large databases.	It works fine for images with open eyes but struggles to identify images when the eyes in the image are closed.
Recognito [10]	Facial Recognition	Provides facial recognition like our app does for unidentified bodies.	Does not use AI and is not tested for closed eye detection.
Complaint Assistant App [11]	AI-based complaint escalation	Understands and evaluates the criticality of a complaint before escalating it accordingly.	The system has limited accuracy and struggles with complex complaints.
IoT Agriculture Monitoring System [12]	Generate alerts on environment conditions	Monitors and generates alerts on soil moisture levels similar to our project.	The system is limited by the accuracy of the devices and the varying nature of plants.

3.4 Conclusion

In this chapter, we reviewed existing solutions like FindFace and OpusXenta, comparing their features with our project. While these systems offer valuable insights, our platform stands out by integrating AI, IoT, and real-time service management, offering a more comprehensive and specialized solution for cemetery and funeral service management.

Chapter 4 Software Requirement Specifications

This chapter takes a look at all the conditions that the system must adhere to and what makes up the quality attributes of the system. The functional and non-functional requirements are also noted. It also includes elements of GUI and various other forms of documentation.

4.1 List of Features

The major features of Akhri Aramgah are the following:

- Centralized platform connecting the users and various stakeholders of the funeral services sectors.
- Location-based algorithms to find nearest burial spaces and allocation of convenient orders to service providers.
- IoT-based monitoring of soil moisture levels to predict possible grave damage
- AI-based facial recognition for identifying unidentified bodies in morgues
- NLP-based complaint sorting to prioritize critical issues
- Data collection and report generation for research purposes
- Admin features for adding and managing graveyards and viewing detailed records

4.2 Functional Requirements

Following are the functional requirements for each of the stakeholders of this project.

4.2.1 Functional requirements for users

- The system shall allow the users to register and sign in.
- The system shall allow the users to book burial and funeral services.
- The system shall allow the users to securely make payments.
- Users shall pay upfront and the payment will be held until the completion of service.
- The system shall allow the users to register complaints.
- The system shall generate warnings on possible damage to a grave and show it to users.
- The system shall allow users to search the morgues for missing relatives.

4.2.2 Functional requirements for service providers

- The system shall allow the service providers to sign in.
- The system shall allocate orders to service providers based on their location.
- The system shall allow service providers to view, accept, and reject orders.
- The system shall release payments to service providers on completion of orders.

4.2.3 Functional requirements Gorkan

- The system shall allow the Gorkan to log in.
- The system will allow the Gorkan to view orders and maintenance requests.
- The system will allow the Gorkan to update the order status.

4.2.4 Functional requirements for Admins

- The system shall allow the admin to sign in.
- The system shall show the details of each and everything to the admin.
- The system shall allow the admin to create and update graveyards.
- The system shall allow the admin to view and respond to user complaints.
- The system shall allow the admin to ban any user or service provider.
- The system shall show relevant analytics to the admin.

4.2.5 Functional requirements for Morgue Staff

- The system shall allow the morgue staff to sign in.
- The system shall allow the morgue staff to upload images and details of every unidentified body.
- The system shall allow the morgue staff to view and respond to user requests.

4.3 Quality Attributes

Following are the quality attributes of Akhri Armagah

4.3.1 Reliability

The system shall be designed with a clear focus on reliability. It shall ensure consistent performance and great accuracy under various situations. It should have the users' trust that it will function correctly at all times.

4.3.2 Maintainability

The system shall be coded in such a way that it is easier to update and add new code to it. This is the key to making sure the system keeps improving with time and does not feel outdated at any time.

4.3.3 Usability

The system shall feature a familiar and easy-to-use interface which will allow the users to navigate the system with ease. Keeping the interface style familiar by keeping in mind the general standard will make sure that the system feels highly accessible to the users.

4.3.4 Security

The system shall ensure that all the data being stored and passed through it must be secured. The system will utilize the industry-standard security protocols to ensure maximum security.

4.3.5 Responsiveness

The system shall respond to user actions quickly and provide immediate feedback to the user to ensure a seamless experience for the users.

4.4 Non-Functional Requirements

The system has the following non-functional requirements.

4.4.1 Performance

The system shall be able to provide quick responses to users preferably less than 3 seconds on average.

4.4.2 Independence of services

The system architecture shall ensure that when one service failure occurs, the other services do not get affected.

4.4.3 Separation of concerns

The system shall ensure proper separation of concerns between the various kinds of users using the platform.

4.4.4 Security

The system shall ensure that the important data of users is properly secured using encryption and various security measures.

4.4.5 Error handling

The system shall ensure that the user will receive clear and helpful error messages to guide them.

4.4.6 User friendly interface

The system shall ensure a user-friendly interface based on the general standards so the user does not feel uncomfortable.

4.5 Assumptions

The system will be designed taking the following assumptions in mind.

4.5.1 Government backing

It is assumed that this project will have government backing and will be implemented by the government itself and used by government officials.

4.5.2 Mobile literacy

It is assumed that the service providers and the Gorkans have basic mobile literacy and can use it as part of their jobs.

4.5.3 Internet accessibility

It is assumed that all users of this application will have stable access to the internet at all times.

4.5.4 Training availability

It is assumed that the government will organize basic training on how to use this system for all relevant officials.

4.5.5 Government Control

It is assumed that all the graveyards of the city are control and managed by government officials.

4.6 Use Cases

The use cases for the various functionalities of the Akhri Aramgah are described as follows.

4.6.1 Login

Name	Login		
Actors	Admin, User, Gorkan, Transporter, Caterer, Morgue		
Summary	The user shall provide their email and password on the login form, and after successful verification, redirect the user to the home page		
Pre-Conditions	The user must be registered and must not already be logged in		
Post-Conditions	The credentials are successfully verified and shall be redirected to the home page		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The user opens the login page	2	The login page is displayed asking for email and password
2	The user enters valid email and password	4	The system verifies the email and password, establishes session and redirects the user to the home page
Alternative Flow			
3	The user enters invalid email or password	4-A	The system responds with an error message: Incorrect email or password entered

4.6.2 Search Reports

Name	Search Reports		
Actors	Admin		
Summary	The admin shall be able to search reports from saved reports.		
Pre-Conditions	The admin must be logged in to the app.		
Post-Conditions	The admin has found/not found the report.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The admin opens the dashboard	2	The system displays the dashboard
2	The admin selects search report and enters report name	3	The system finds the report and opens it
Alternative Flow			
		3-A	The system does not find report and responds with an error message: No Report Found

4.6.3 Register

Name	Register		
Actors	User		
Summary	The user shall provide their email, phone number, CNIC, and password on the registration form, and verification, redirect the user to the login page.		
Pre-Conditions	The user must not be registered already.		
Post-Conditions	The user's details are successfully verified and shall be redirected to login page.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The user opens the registration page	2	The registration page is displayed asking for email, password and other details
2	The user enters valid details	4	The system verifies the details, establishes a session for user and sends an OTP to user
4	The user enters OTP	5	The system verifies the OTP, and redirects the user to the home page
Alternative Flow			
3	The user enters invalid email or password	4-A	The system responds with an error message: Incorrect details entered

4.6.4 Book Grave

Name	Book Grave		
Actors	User		
Summary	The user shall be able to book a grave in their desired graveyard		
Pre-Conditions	The user must be logged in to the app.		
Post-Conditions	The user has successfully booked a grave.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The user opens the booking page	2	The booking page is displayed asking for details of deceased person
2	The user enters details and selects a grave	4	The system verifies the details, and asks for payment
5	The user enters payment details	6	The system verifies the payment details, deduct amount and display message: Grave booked successfully
Alternative Flow			
3	The user enters invalid details	4-A	The system responds with an error message: Incorrect details entered
5	The user enters invalid payment details or insufficient balance	6-A	The system responds with an error message: Payment failed

4.6.5 Book Cabin in Morgue

Name	Book Cabin in Morgue		
Actors	User, Morgue		
Summary	The user shall be able to book a cabin in a morgue		
Pre-Conditions	The user must be logged in to the app.		
Post-Conditions	The user has successfully booked a cabin in morgue.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The user opens the book cabin page	2	The bookings page is displayed asking for details of deceased person and location of user
2	The user enters details and enters his location	4	The system successfully verifies the details, and asks for payment if cabins are available
4	The user enters payment details	6	The system verifies the payment details, deduct amount and display message: Cabin booked successfully
Alternative Flow			
3	The user enters invalid details	4-A	The system responds with an error message: Incorrect details entered
		4-B	The system responds with an error message: Cabin not available
5	The user enters invalid payment details or insufficient balance	6-A	The system responds with an error message: Payment failed

4.6.6 Upload Unidentified Body Image

Name	Upload Unidentified Body Image		
Actors	Morgue		
Summary	The morgue shall be able to upload an image of an unidentified body.		
Pre-Conditions	The morgue user must be logged in to the app.		
Post-Conditions	The morgue user has successfully uploaded the image.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The morgue user opens the upload unidentified body page	2	The system asks to upload image
2	The morgue user selects an image and uploads it	4	The system adds image to database and display message: Image Added Successfully
Alternative Flow			
3	The user uploads invalid image	4-A	The system responds with an error message: Invalid Image

4.6.7 Apply as Service Provider

Name	Apply as Service Provider		
Actors	Transporter, Caterer		
Summary	The service provider shall provide their email, service details, password on the registration form, and after successful verification, redirect the service provider to the login page.		
Pre-Conditions	The service provider must not be registered in the app, and must have their business registered with the government.		
Post-Conditions	The service provider is successfully verified and shall be redirected to the login page.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The service provider opens the registration page	2	The registration page is displayed asking for details of service provider
2	The service provider enters valid details	4	The system verifies the details, establishes a session for the user and sends an OTP to service provider
4	The service provider enters OTP	5	The system verifies the OTP, and redirects the service provider to the home page
Alternative Flow			
3	The service provider enters invalid email or password	4-A	The system responds with an error message: Incorrect details entered

4.6.8 Manage Complaints

Name	Manage Complaints		
Actors	Admin		
Summary	The admin shall be able to manage complaints.		
Pre-Conditions	The admin must be logged in to the app.		
Post-Conditions	The admin has checked and managed complaints.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The admin opens manage complaints page	2	The system shows manage complaints page
2	The admin selects search a complaint and update anything if needed	3	The system updates changes made.
Alternative Flow			
		3-A	The system couldn't update changes and responds with an error message: Changes Not Made

4.6.9 Add New Service Provider

Name	Add New Service Provider		
Actors	Admin		
Summary	The admin shall be able to add new service providers.		
Pre-Conditions	The admin must be logged in to the app.		
Post-Conditions	The admin has successfully added new service providers.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The admin opens manage service providers page	2	The system displays manage service providers page
2	The admin selects add new service provider	3	The system displays service providers that applied
3	The admin selects a service provider and adds him	4	The system adds service provider and displays message: Service Provider Added Successfully
Alternative Flow			
		2-A	The system couldn't find any service provider and responds with an error message: No Service Provider Found

4.6.10 Manage Graveyards

Name	Manage Graveyards		
Actors	Admin		
Summary	The admin shall be able to manage graveyards.		
Pre-Conditions	The admin must be logged in to the app.		
Post-Conditions	The admin has made a change to graveyards.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The admin opens manage graveyards page	2	The system displays the manage graveyards page
2	The admin selects search a graveyard and update anything if needed	3	The system updates changes made
Alternative Flow			
		3-A	The system could not to update changes and responds with an error message: Changes Not Made

4.6.11 Request Service from Caterer

Name	Request Service from Caterer		
Actors	User, Caterer		
Summary	The user shall be able to book catering services.		
Pre-Conditions	The user must be logged in to the app.		
Post-Conditions	The user has successfully booked catering service.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The user opens the book services page and selects book catering	2	The booking page is displayed asking for catering type and location of user
2	The user selects boxes/pot from dropdown and enters the location	4	The caterer accepts it, and the system asks for payment
4	The user enters payment details	6	The system verifies the payment details, deduct amount and display message: Catering services booked successfully
Alternative Flow			
3	The user enters invalid location	4-A	The system responds with an error message: Incorrect location entered
5	The user enters invalid payment details or insufficient balance	6-A	The system responds with an error message: Payment failed

4.6.12 Request Service from Transporter

Name	Request Service from Transporter		
Actors	User, Transporter		
Summary	The user shall be able to book transport for dead bodies/people.		
Pre-Conditions	The user must be logged in to the app.		
Post-Conditions	The user has successfully booked a transport.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The user opens the book services page and selects book transport	2	The booking page is displayed asking for a transport and location of user
2	The user selects people/dead body from dropdown and enters the location	4	The transporter accepts transport, and the system asks for payment
4	The user enters payment details	6	The system verifies the payment details, deduct amount and display message: Transport booked successfully
Alternative Flow			
3	The user enters invalid location	4-A	The system responds with error message: Incorrect location entered
5	The user enters invalid payment details or insufficient balance	6-A	The system responds with an error message: Payment failed

4.6.13 Respond to Grave Order

Name	Respond to Grave Order		
Actors	Gorkan		
Summary	The Gorkan shall be able to respond to grave order.		
Pre-Conditions	The Gorkan must be logged in to the app.		
Post-Conditions	The Gorkan has successfully responded to grave order.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The Gorkan opens the grave orders page and selects order	2	The order details are displayed
2	The Gorkan completes the request and responds by done	3	The system notifies the user

4.6.14 Generate Reports

Name	Generate Reports		
Actors	Admin		
Summary	The admin shall be able to generate reports from the dataset.		
Pre-Conditions	The admin must be logged in to the app.		
Post-Conditions	The admin has successfully generated the report.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The admin opens the dashboard	2	The system displays the dashboard
2	The admin selects generate report	3	The system generate report and downloads and saves it

4.6.15 Add New Graveyard

Name	Add New Graveyard		
Actors	Admin		
Summary	The admin shall be able to add new graveyard.		
Pre-Conditions	The admin must be logged in to the app.		
Post-Conditions	The admin has added a new graveyard.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The admin opens manage graveyards page	2	The system displays graveyards page
2	The admin inputs details of new graveyard	4	The system adds in the database
Alternative Flow			
3	The admin enters invalid details of graveyard	4-A	The system responds admin with an error message: Invalid Details

4.6.16 Request Grave Maintenance

Name	Request Grave Maintenance		
Actors	User		
Summary	The user shall be able to request maintenance for a grave		
Pre-Conditions	The user must be logged in the app and must have a grave booked in the past.		
Post-Conditions	The user has successfully requested maintenance.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The user opens the booking history	2	The booking history is displayed
2	The user selects request maintenance and selects appropriate option	3	The system asks for payment
3	The user enters details	5	The system verifies payment details and deducts amount
Alternative Flow			
4	The user enters invalid payment details or insufficient balance	5-A	The system responds with an error message: Payment failed

4.6.17 Search Missing Person

Name	Search Missing Person		
Actors	User		
Summary	The user shall be able to search for missing persons.		
Pre-Conditions	The user must be logged in to the app.		
Post-Conditions	The user has looked for missing person.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The user opens the search for missing person page	2	The page is displayed asking for image of missing person
2	The user selects an image and submit it	3	The system searches in database from morgue, and the person is found
		4	The system displays morgue details, and display message:Person found successfully
Alternative Flow			
	.	3-A	The system searches in database from morgue, and the person is not found
		4-A	The system displays a message: Not Found

4.6.18 Manage Service Providers

Name	Manage Service Providers		
Actors	Admin		
Summary	The admin shall be able to manage service providers.		
Pre-Conditions	The admin must be logged in to the app.		
Post-Conditions	The admin has made changes to service providers.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The admin opens manage service providers page	2	The system displays manage service providers page
2	The admin searches selects a service provider and selects remove and temporarily disable	3	The system removes or temporarily disable them

4.6.19 Update Grave Status

Name	Update Grave Status		
Actors	Gorkan		
Summary	The Gorkan shall be able to update grave status.		
Pre-Conditions	The Gorkan must be logged in to the app.		
Post-Conditions	The Gorkan has successfully updated grave status.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The Gorkan opens the grave orders page and selects an order	2	The order details are displayed
2	The Gorkan selects change grave status and updates grave status	3	The system notifies the user

4.6.20 Respond to Maintenance Request

Name	Respond to Maintenance Request		
Actors	Gorkan		
Summary	The Gorkan shall be able to respond to maintenance requests for graves by user		
Pre-Conditions	The Gorkan must be logged in to the app.		
Post-Conditions	The gorkan has successfully responded to maintenance request.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The Gorkan opens the maintenance requests page and selects a request	2	The request details are displayed
2	The Gorkan completes the request	3	The system notifies the user

4.7 Hardware and Software Requirements

Following are the hardware and software requirements for Akhri Aramgah.

4.7.1 Hardware Requirements

The following are Akhri Aramgah's hardware requirements.

- A computer system with 16 gigabytes of RAM
- A stable internet connection
- A capacitive moisture sensor
- Arduino ESP-8266 Wi-Fi module
- Signal wire for Sensor
- Jumper wires for connectivity

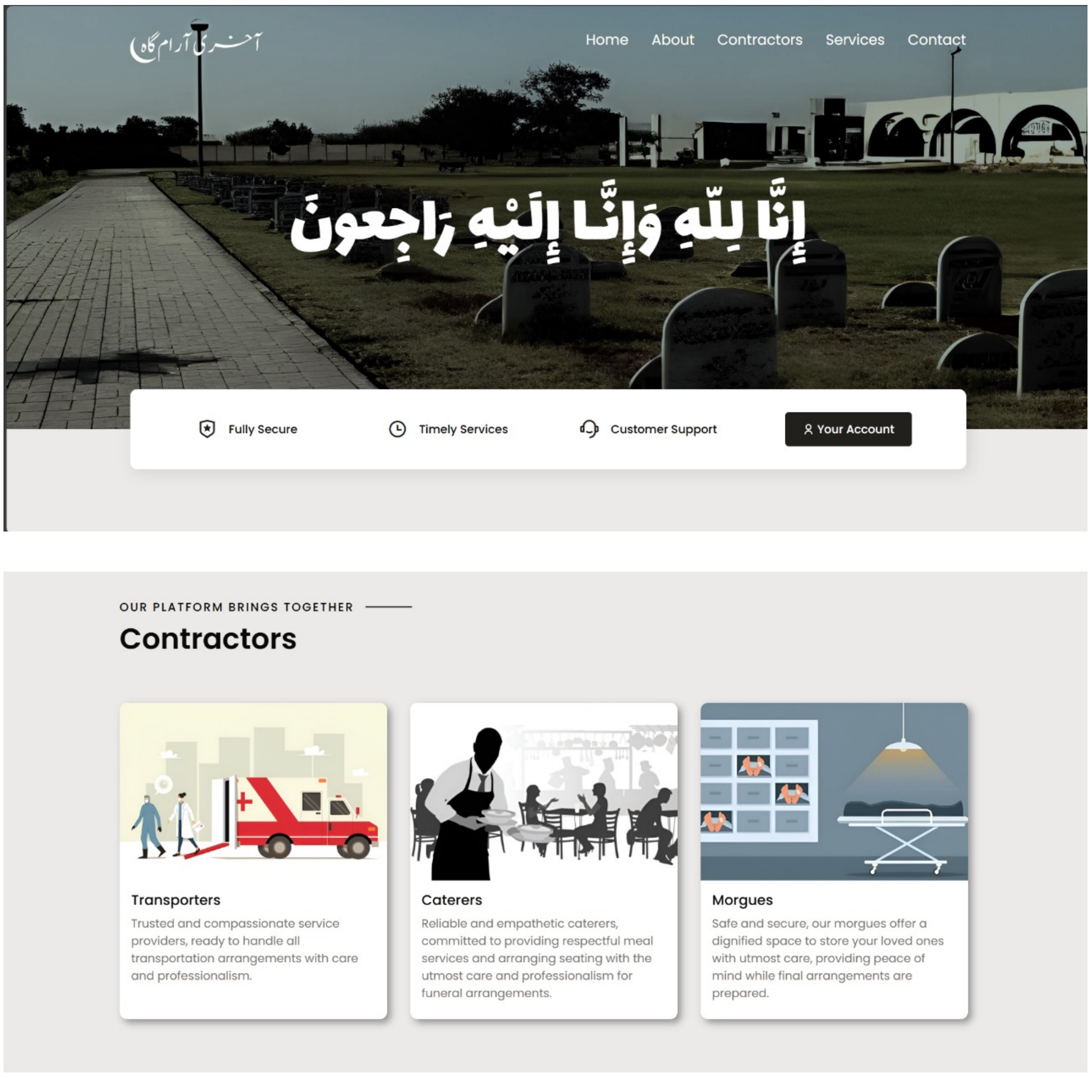
4.7.2 Software Requirements

The following are Akhri Aramgah's software requirements.

- ReactJS, ExpressJS, NodeJS, and Flask for development web applications and backend.
- React Native and Native (JDK) for development of mobile applications.
- Python and JavaScript 3rd party libraries for multiple features.
- Firebase will serve as the database
- A JavaScript-enabled browser
- GitHub for collaboration
- Windows Terminal
- Visual Studio Code and Android Studio as the IDEs

4.8 Graphical User Interface

The following are Graphical User Interfaces of Akhri Aramgah Applications.



Transporters
Trusted and compassionate service providers, ready to handle all transportation arrangements with care and professionalism.

Caterers
Reliable and empathetic caterers, committed to providing respectful meal services and arranging seating with the utmost care and professionalism for funeral arrangements.

Morgues
Safe and secure, our morgues offer a dignified space to store your loved ones with utmost care, providing peace of mind while final arrangements are prepared.

Figure 4.1: Graphical Representation of the Website's Landing Page Interface

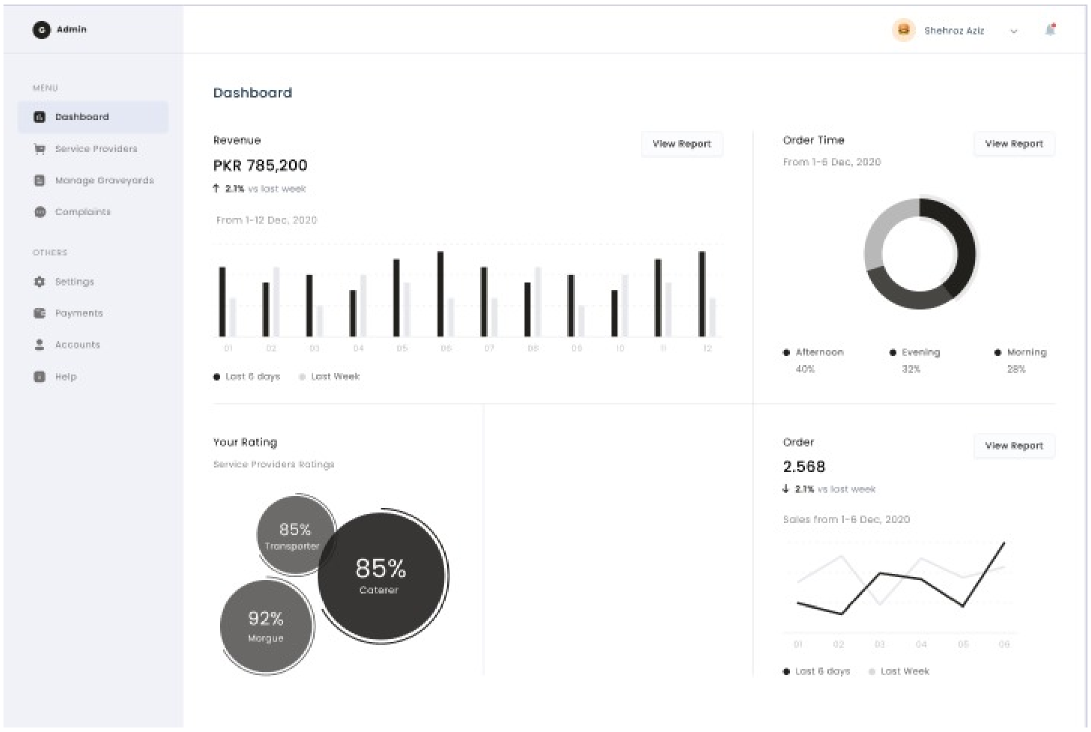


Figure 4.2: Graphical Representation of the Admin Dashboard Webpage Interface

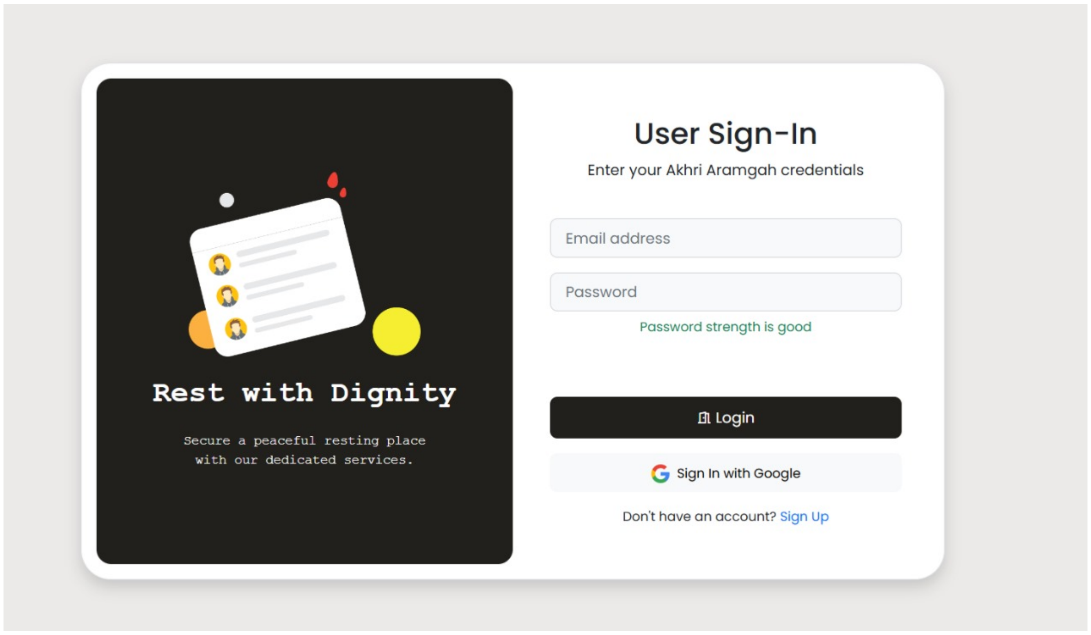
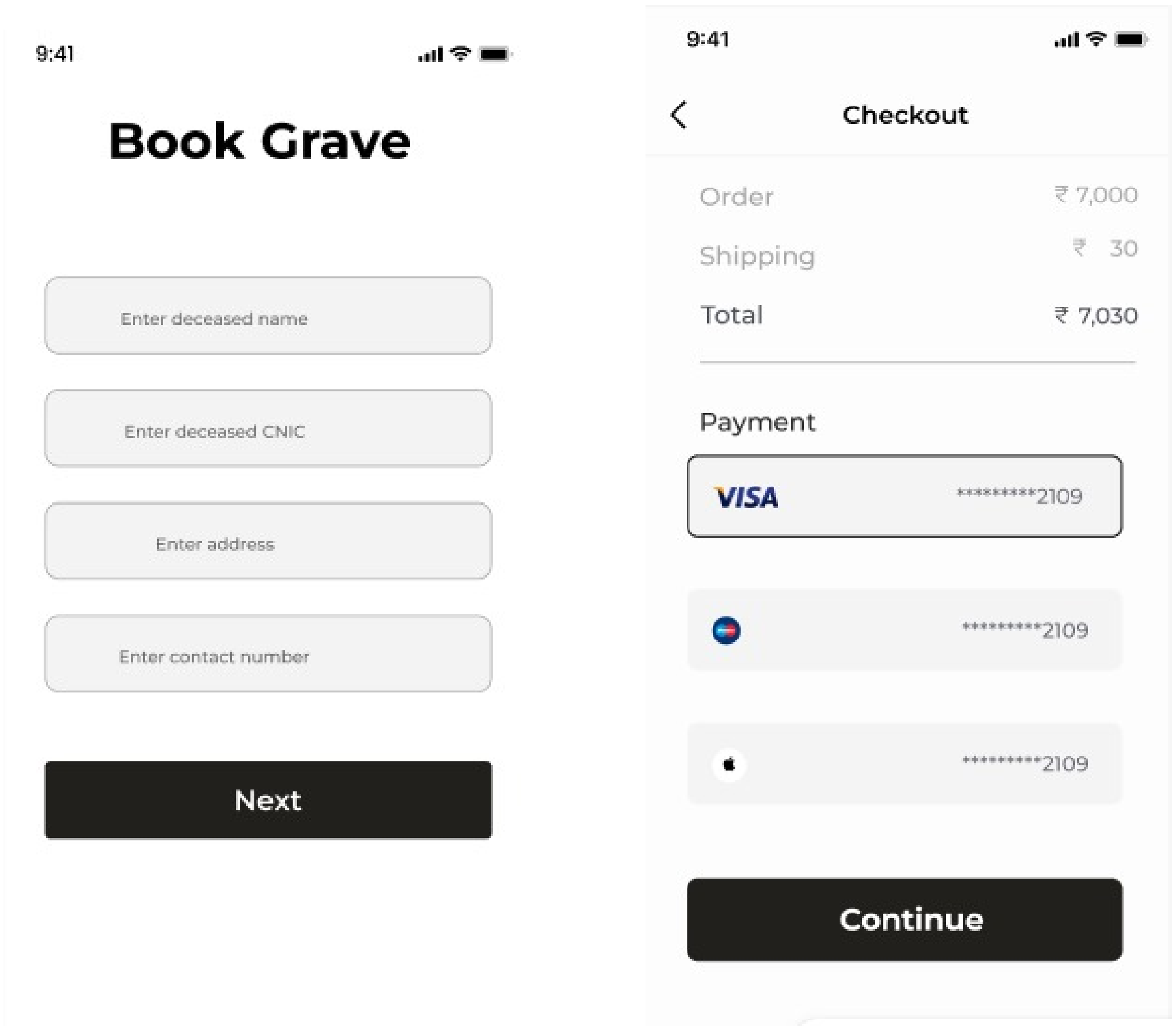

Rest with Dignity
Secure a peaceful resting place with our dedicated services.
User Sign-In
Enter your Akhri Aramgah credentials
Email address
Password
Password strength is good
Login
Sign In with Google
Don't have an account? Sign Up

Figure 4.3: Graphical Representation of the User Sign-In Webpage Interface



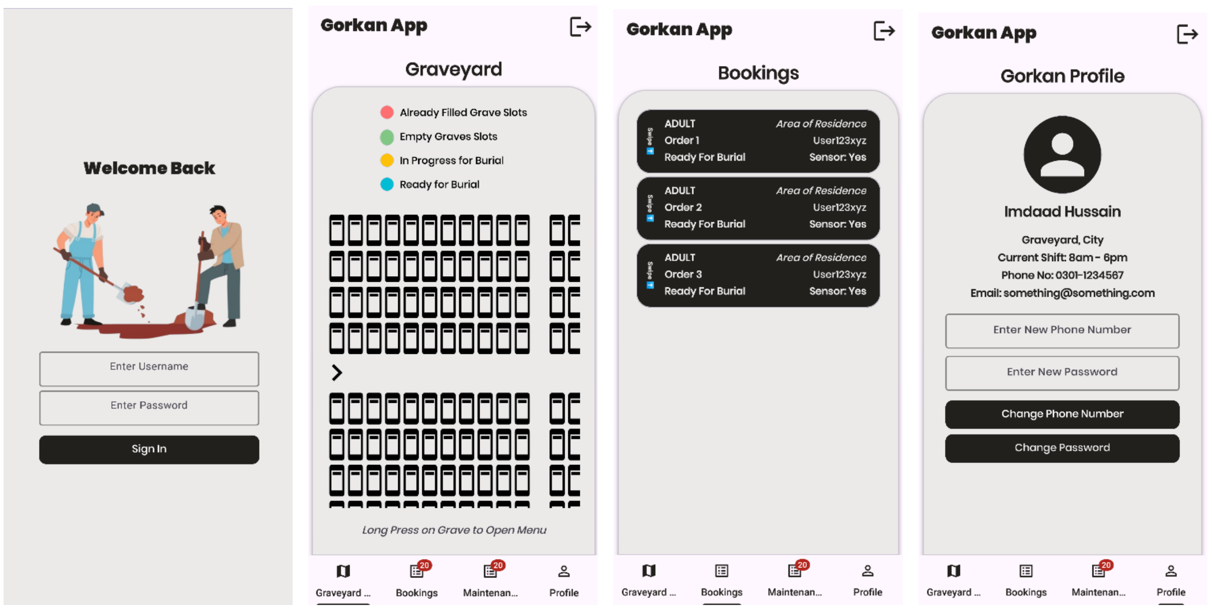
Order	₹ 7,000
Shipping	₹ 30
Total	₹ 7,030

**********2109

**********2109

**********2109

Figure 4.4: Graphical Representation of the Book Grave Module in the Mobile Application

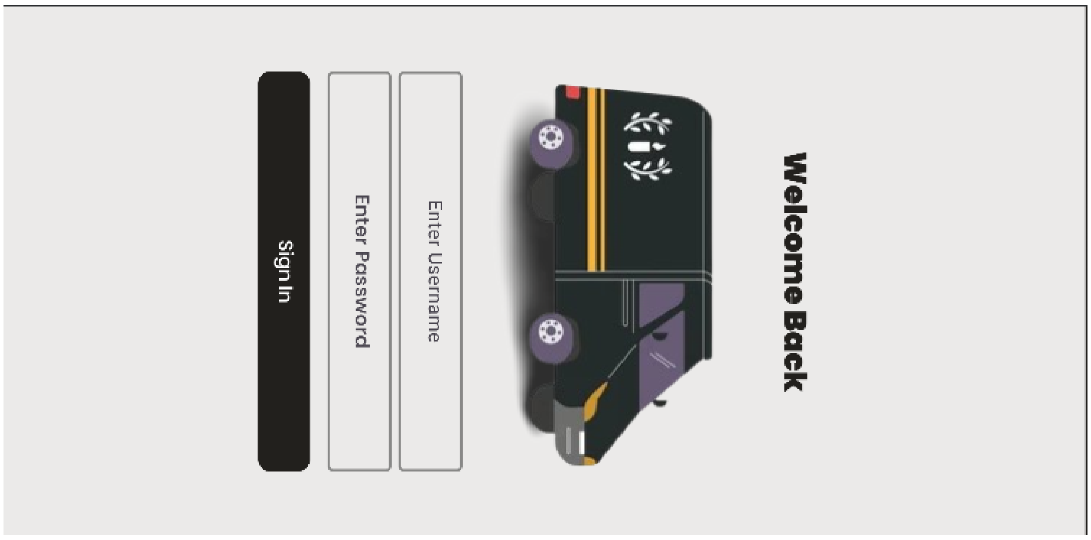


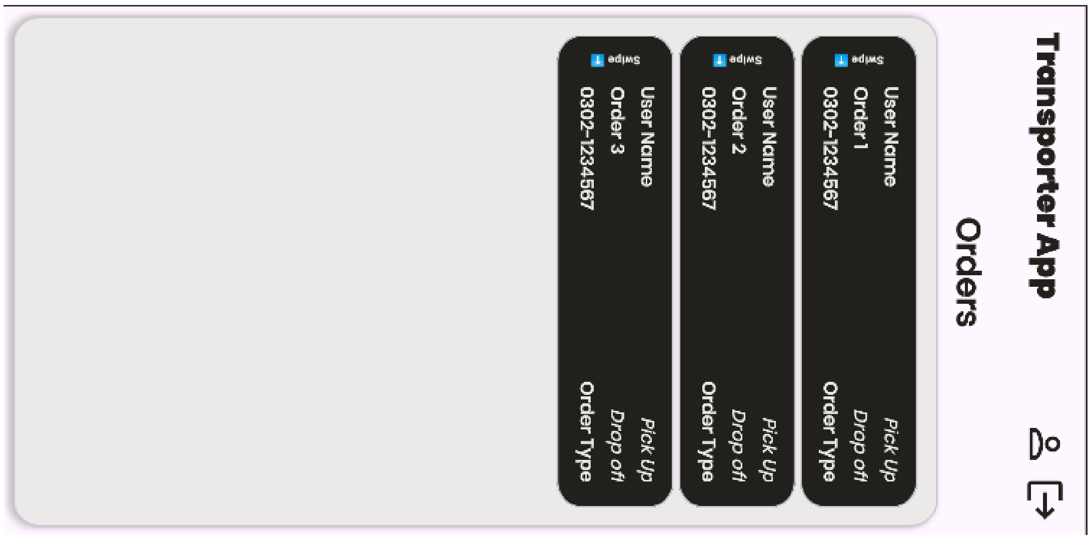
🟥 Already Filled Grave Slots
🟩 Empty Graves Slots
🟨 In Progress for Burial
🟦 Ready for Burial

ADULT	Area of Residence
Order 1	User123xyz
Ready For Burial	Sensor: Yes
ADULT	Area of Residence
Order 2	User123xyz
Ready For Burial	Sensor: Yes
ADULT	Area of Residence
Order 3	User123xyz
Ready For Burial	Sensor: Yes

Imdaad Hussain
Graveyard, City
Current Shift: 8am - 6pm
Phone No: 0301-1234567
Email: something@something.com

Figure 4.5: Graphical Representation of the Gorkan Mobile Application Interface





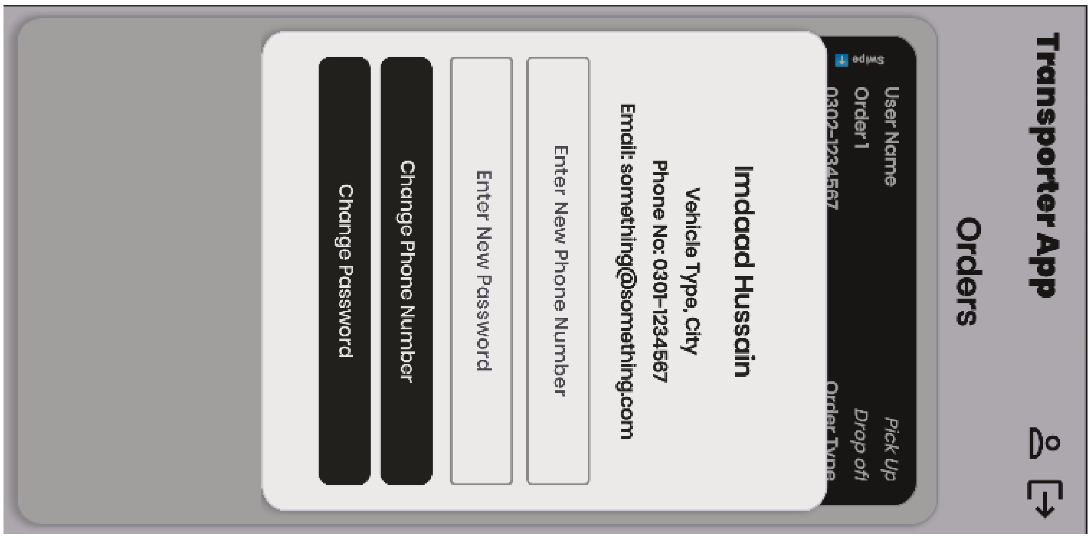


Figure 4.6: Graphical Representation of the Service Providers Application Interface

4.9 Database Design

4.9.1 ER Diagram

The following is the Entity Relationship Diagram of Akhri Aramgah.

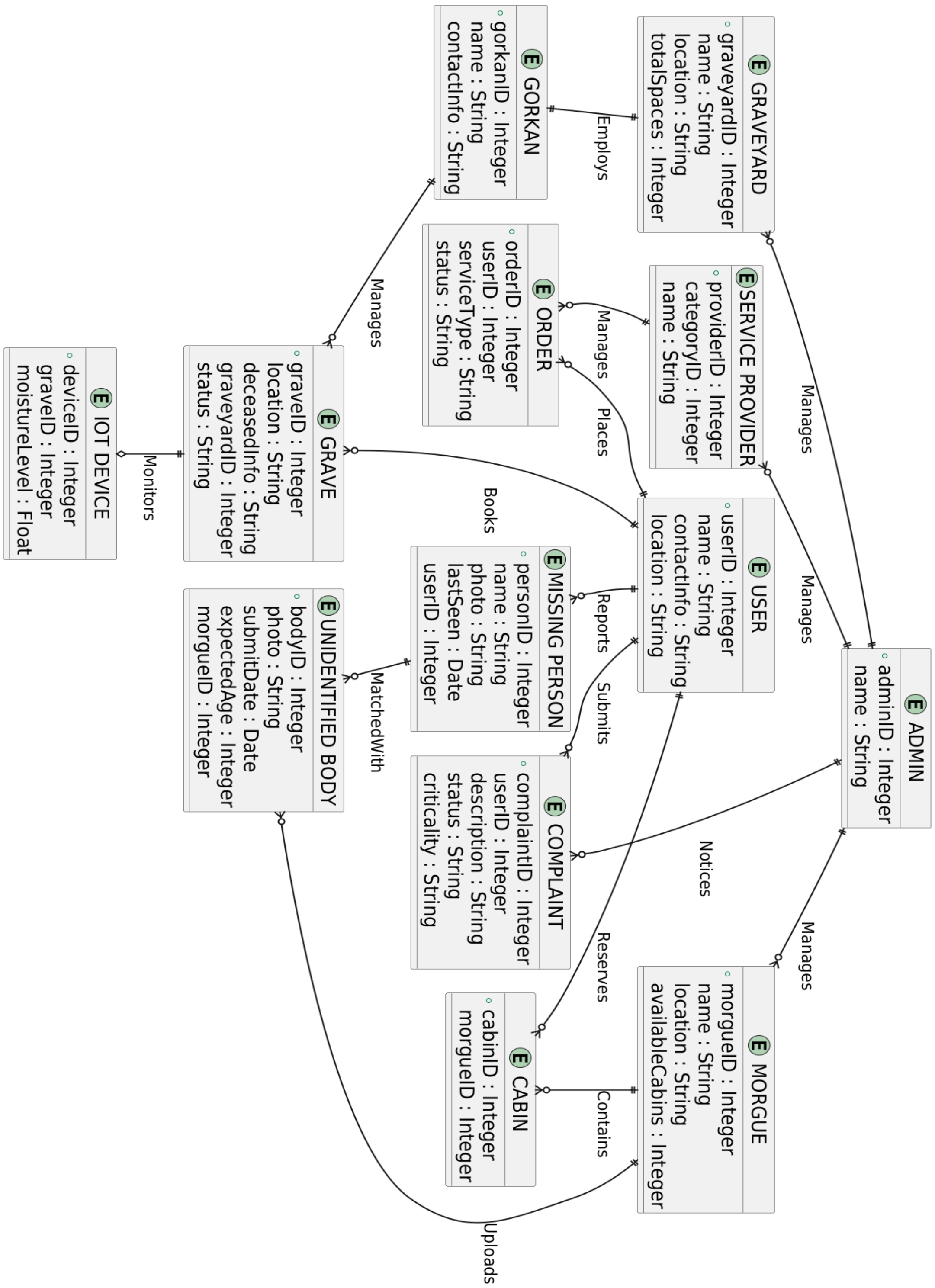


Figure 4.7: Entity Relationship Diagram Representing the Project's Data Model

4.9.2 Data Dictionary

The following is the Data Dictionary of Akhri Aramgah.

Table 4.1: Table Displaying the Project's Data Dictionary Details

Entity	Attribute	DataType	Nullable	Description
USER	userID	Integer	No	Primary Key
	name	String	No	User's Name
	contactInfo	String	No	Contact Information
	location	String	Yes	User's Location
PAYMENT	paymentID	Integer	No	Primary Key
	userID	Integer	No	Foreign Key referencing User
	amount	Integer	No	Amount of the Payment
GRAVEYARD	graveyardID	Integer	No	Primary Key
	name	String	No	Graveyard's Name
	location	String	No	Graveyard's Location
	totalSpaces	Integer	No	Number of Burial Spaces
GRAVE	graveID	Integer	No	Primary Key
	location	String	No	Grave's Location
	deceasedInfo	String	No	Deceased Person Information
	graveyardID	Integer	No	Foreign key referencing GRAVEYARD
	status	String	Yes	Status of the Grave
IOT DEVICE	deviceID	Integer	No	Primary Key
	graveID	Integer	No	Foreign Key referencing GRAVE
	moistureLevel	Float	Yes	Moisture Level of the Grave
GORKAN	gorkanID	Integer	No	Primary Key
	name	String	No	Gorkan's Name
	contactInfo	String	No	Contact Information of Gorkan
ADMIN	adminID	Integer	No	Primary Key
	name	String	No	Admin's Name
SERVICE	providerID	Integer	No	Primary Key
PROVIDER	categoryID	Integer	Yes	Service Category ID
	name	String	No	Service Provider's Name
COMPLAINT	complaintID	Integer	No	Primary Key

Entity	Attribute	DataType	Nullable	Description
	userID	Integer	No	Foreign Key referencing USER
	description	String	Yes	Complaint Description
	status	String	Yes	Current Status of the Complaint
	criticality	String	Yes	Criticality Level of the Complaint
ORDER	orderID	Integer	No	Primary Key
	userID	Integer	No	Foreign Key referencing USER
	serviceType	String	No	Type of Service Ordered
	status	String	Yes	Current Status of the Order
MORGUE	morgueID	Integer	No	Primary Key
	name	String	No	Morgue's Name
	location	String	No	Morgue's Location
	availableCabins	Integer	No	Number of Available Cabins
CABIN	cabinID	Integer	No	Primary Key
	morgueID	Integer	No	Foreign Key referencing MORGUE
MISSING	personID	Integer	No	Primary Key
PERSON	name	String	No	Name of the Missing Person
	photo	String	No	Photo of the Missing Person
	lastSeen	Date	No	Last Seen Date of the Missing Person
	userID	Integer	No	Foreign Key referencing USER
UNIDENTIFIED	bodyID	Integer	No	Primary Key
BODY	photo	String	No	Photo of the Unidentified Body
	submitDate	Date	No	Submission Date of the Body
	expectedAge	Integer	Yes	Estimated Age of the Body
	morgueID	Integer	No	Foreign Key referencing MORGUE

4.10 Risk Analysis

Risk analysis for Akhri Aramgah involves potential challenges the project might face.

- The major problem that the project will face is the resistance to technology that the stakeholders might display as this project is targeting a previously technology-averse sector and this might affect user engagement and system usability.
- The reliance on government support might become an issue as multiple governments have differ-

ent priorities and the system will always face uncertainty.

- The use of highly technical elements like location algorithms, artificial intelligence models for facial recognition and complaints escalation, and IoT devices might be affected by low accuracy due to insufficient data availability.

These risks will need to be mitigated to ensure the success of the project.

4.11 Conclusion

This chapter provides a comprehensive overview of the Akhri Aramgah system and all the key elements needed for the development and implementation of the project. All the functional and non-functional requirements of the project are discussed, along with the quality attributes the project means to follow. The use cases and assumptions help understand how the users will interact with the system. Risk analysis highlights the potential problems the project expects to face. The hardware and software requirements for the development of the project are mentioned as well.

Chapter 5 High-Level and Low-Level Design

Akhri Aramgah is a centralized digital solution to the sector of funeral and burial services. The system connects various stakeholders of this sector including general users, service providers, graveyard managers, morgue staff, and admins. The primary aim of the system is to bring digitalization to a traditionally manual and disconnected sector in the hopes of facilitating all the stakeholders.

5.1 System Overview

The platform is designed with efficiency and accessibility considerations, with a particular focus on modularity such that the modules each work independently even when a part of the system is down. The use of access for all users is a key factor in the thought process behind the design of the system. The full focus is on providing an intuitive user interface, especially since the nature of the service being provided is quite sensitive

The system will be divided into several modules, each addressing the needs of a particular stakeholder. Following are the modules of the project.

- Users will be able to book graves and funeral services as well as register complaints and search for missing people in morgues.
- Service providers will be able to view and accept orders that will be assigned to them based on their location.
- Graveyard managers will be notified of grave booking and maintenance requests.
- Morgue staff will be notified of cabin booking requests and will be tasked to upload images of unidentified dead bodies in the morgue.
- Admins will be able to oversee the system and respond to queries and complaints.

5.2 Design Considerations

5.2.1 Assumptions and Dependencies

5.2.1.1 Assumptions

Following are the assumptions for design considerations of Akhri Aramgah.

- Basic technological literacy of the service providers will be a key assumption when designing the system.

- The project assumes that all graveyards are government property and that the government has centralized access to them.
- While there are security measures in place, the system assumes that there would not be automation attacks.

5.2.1.2 Dependencies

Following are the dependencies for design considerations of Akhri Aramgah.

- The system will be largely compatible with all kinds of devices that have a browser while the mobile apps will target Android devices but the mobile app for users will be cross-platform.
- Any changes to the government policies regarding burial services will require updates to the system.
- The system will depend on the hardware devices like servers and IOT devices.

5.2.2 General Constraints

Akhri Aramgah faces the following constraints.

- The system relies heavily on the availability of the internet, which renders the system ineffective in areas where there is limited internet connectivity.
- The integration of IoT devices and the use of cameras to capture images of unidentified bodies makes the system dependent upon hardware which might not always be that accurate.
- The transition from manual record keeping to a digital platform will be a major constraint.
- The changes in government policies impact the system's functionality.
- The development and maintenance of such a system on a professional scale require funds.
- The system and all of its components must be developed within a year.

5.2.3 Goals and Guidelines

5.2.3.1 Goals

Akhri Aramgah aims to achieve the following goals.

- The major goal of the system is to provide a centralized solution to connect the various stakeholders of the funeral services sector.
- To provide an easy way of booking graves and funeral services.

- Improve transparency at the government level by providing admins with centralized controls.
- To help boost the finances of the service providers by connecting them with more clients.

5.2.3.2 Guidelines

Akhri Aramgah has the following guidelines.

- The users of the system should familiarize themselves with the usage of digital tools provided by the system.
- It is customary for the service providers to have the same basic knowledge of smart mobile phones and how to use them.
- The system will occasionally ask the users to update their information to maintain accurate records
- It is recommended that all the stakeholders should adhere to the government policies regarding this sector.

5.2.4 Development Methods

This project will be developed using the waterfall model because the requirements of this project are set by us and are locked. There are no changes expected in the requirements and the waterfall method is well-suited to such projects. Another driving factor behind choosing this model of development is the fact that this project is very long and extensive due to the development of more than 5 connected applications. Waterfall seems the most feasible model for development. There is room for adopting the techniques of other models as well such as the scrum because this project will feature regular meetings but the major model used will still be the former. The Waterfall methodology provides clarity and structure, making it suitable for this project where requirements are well-defined and unlikely to change that much during development.

5.3 System Architecture

The architecture of Akhri Aramgah system is designed to be modular and efficient. The primary focus is on the independence of modules such that when one service goes down, the others do not get impacted by it. To design such an architecture both the front-ends and back-ends will be segregated in such a way that the general user can still connect with the other actors like service providers on the same backend. The database will remain singular. The front ends consist of distinct applications: a web and mobile app for general users, a web app for admins, and mobile apps for Gorkans, morgue staff, transporters, and caterers. This essentially means that all the other actors will get their backend and the user will be able to connect with all these backends through API calls.

This will ensure that when one backend server is down the user will still be able to connect and use the other features. The project will also feature a dedicated backend server for just the admin side as their work requires regular updates.

5.3.1 Architecture Diagram

The architecture of the system can be clearly seen in this diagram.

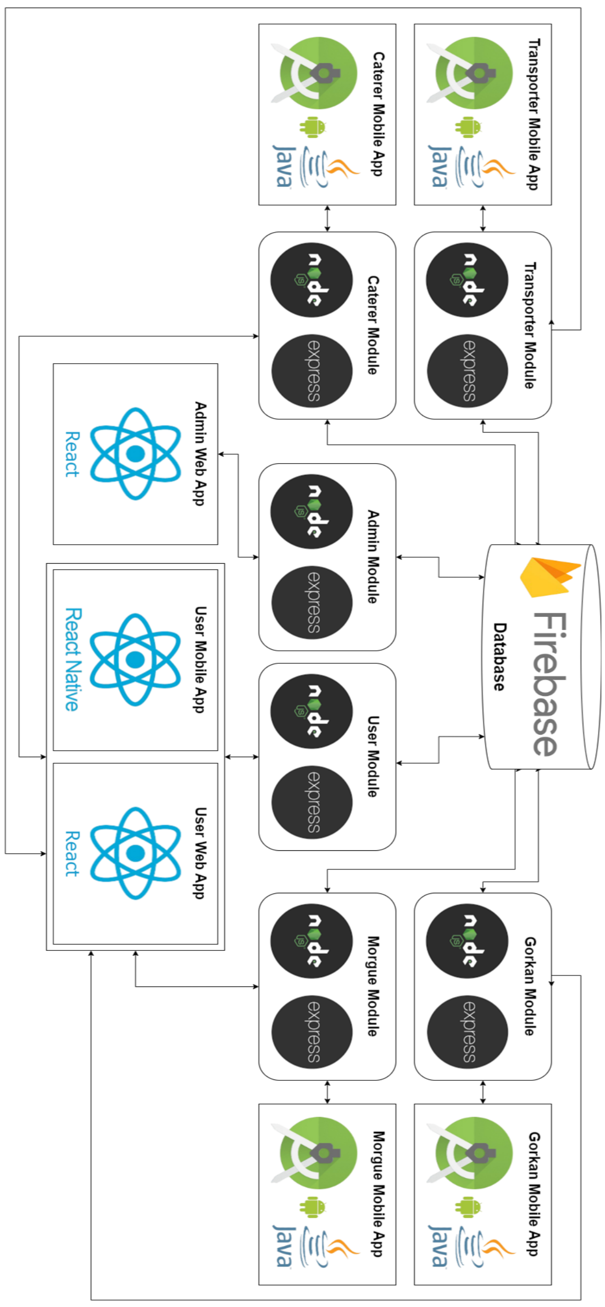


Figure 5.1: Architecture Diagram Illustrating the Project's Structural Design

5.3.2 Component Diagram

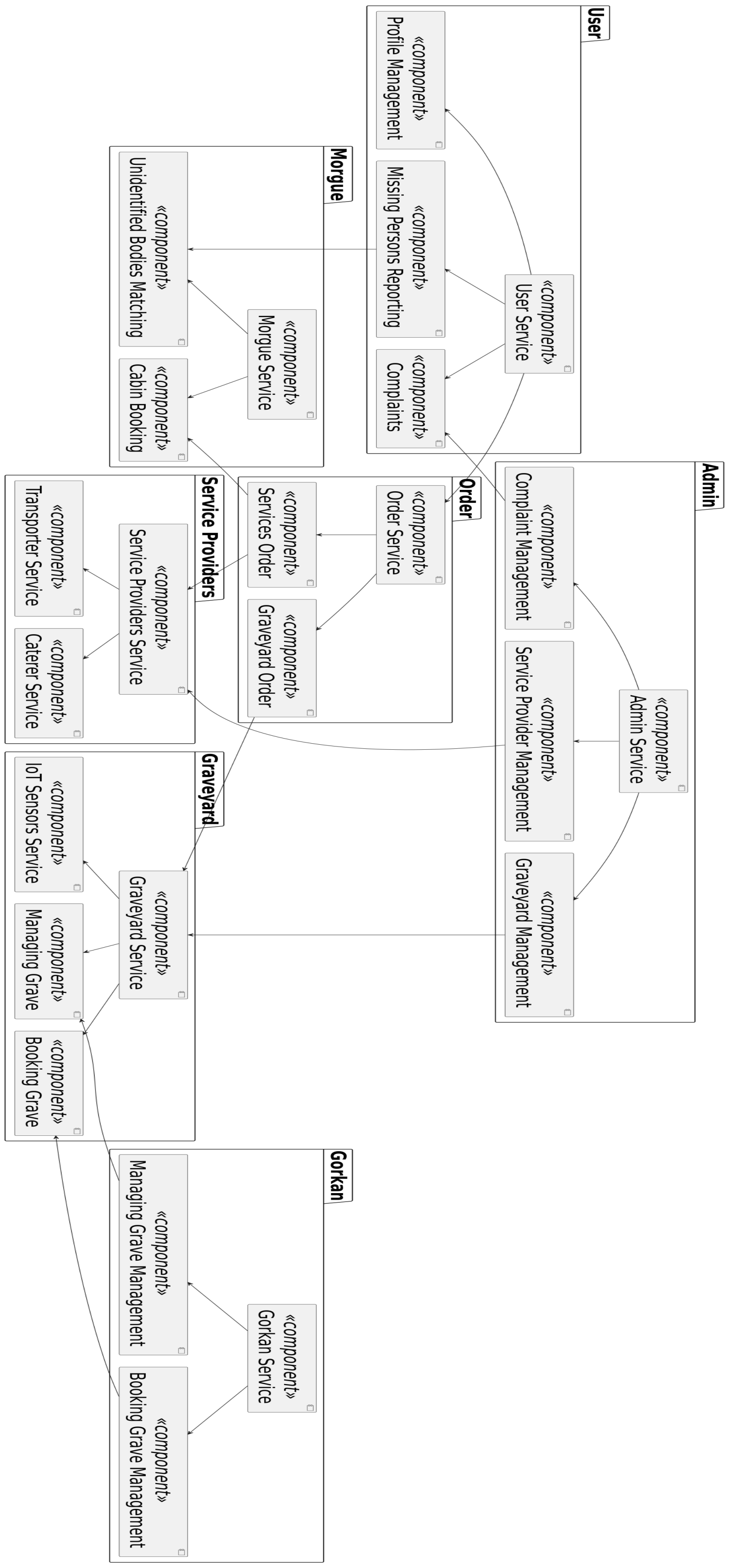

User
«component» Profile Management
«component» Missing Persons Reporting
«component» Complaints
«component» User Service
Morgue
«component» Unidentified Bodies Matching
«component» Cabin Booking
«component» Morgue Service
Admin
«component» Complaint Management
«component» Service Provider Management
«component» Graveyard Management
«component» Admin Service
Order
«component» Order Service
«component» Services Order
«component» Graveyard Order
Service Providers
«component» Service Providers Service
«component» Transporter Service
«component» Caterer Service
Graveyard
«component» IoT Sensors Service
«component» Managing Grave
«component» Booking Grave
«component» Graveyard Service
Gorkan
«component» Managing Grave Management
«component» Booking Grave Management
«component» Gorkan Service

Figure 5.2: Component Diagram Showing System Architecture and Interactions

5.3.3 Subsystem Architecture

The system is divided into the following modules.

5.3.3.1 Gorkan-User module

This module will connect the user with the Gorkan and allow the user to book graves and the Gorkan to receive grave booking and maintenance orders. By sharing the backend, the module isolates itself from the other functionality of the system and will not be affected by the downtime of other services.

5.3.3.2 User module

This module will be available only to the user and allow the user to register complaints and access other technical features. This system will also utilize NLP to escalate complaints based on their seriousness. The system will also have the functionality to search for missing people in morgues.

5.3.3.3 Admin module

This module will be available only to the admin side and all the features of the admin side will be present here. All the analytics and all the complaints handling will be done here.

5.3.3.4 Morgue-user module

This module will connect the morgue with the user and allow the user to book cabins in the morgue. The morgue staff will be notified of user requests and will upload images of unidentified dead bodies to the database.

5.3.3.5 Transporter-user

This module will connect the transporters to the user and allow users to book transport to transfer their deceased. The transporters will also get their orders through this module.

5.3.3.6 Caterer-user

This module will connect the caterers to the user and allow users to book catering services for funerals. The caterers will also get their orders through this module.

5.4 Architectural Strategies

Akhri Aramgah uses the following architecture strategies

5.4.1 Programming Language

JavaScript, Python, and Java are all used in this project. JavaScript is the major language because nearly all front-end and back-end will be implemented in JavaScript frameworks like ReactJS, NodeJS, ExpressJS, and React Native. Python will be used for some backend services where the use of AI and NLP is needed as the support for these kinds of features is extremely high in Python. Java will be used in the development of mobile apps for the morgue staff, gorkans, and service providers to connect them to users to make API calls.

5.4.2 Database Design

Akhri Aramgah will use Firebase as its singular database, leveraging its real-time nature. This system's main objective is client servicing, and clients demand real-time services, so this database aligns perfectly with our goals. Security is also a key consideration while choosing a database, and Firebase's authentication feature caters to this need as well. Firebase also provides NoSQL flexibility to store data in tree-like structures or documents in the collection, which helps handle data and scale applications easily.

5.4.3 Product Enhancement and Extensibility

The use of independent modules keeps the door for future development open as if one of the modules is being extended then it can easily be taken down improved and sent back live again all while the other modules keep on running all the same. This architecture will help in future development due to the fact that there is no inter-dependence between the modules and modules like the admin side can be modified without any impact to the other users.

5.4.4 User-Centric Interface Design

The interface design is an important part of the system as this is a user-service system and focusing on accessibility must be an absolute priority. There are dedicated front-end applications for each stakeholder to provide a tailored solution to each of them. By prioritizing the user experience the system enhances the usability and market.

5.5 Domain Model/Class Diagram

The following is the Class Diagram of Akhri Aramgah.

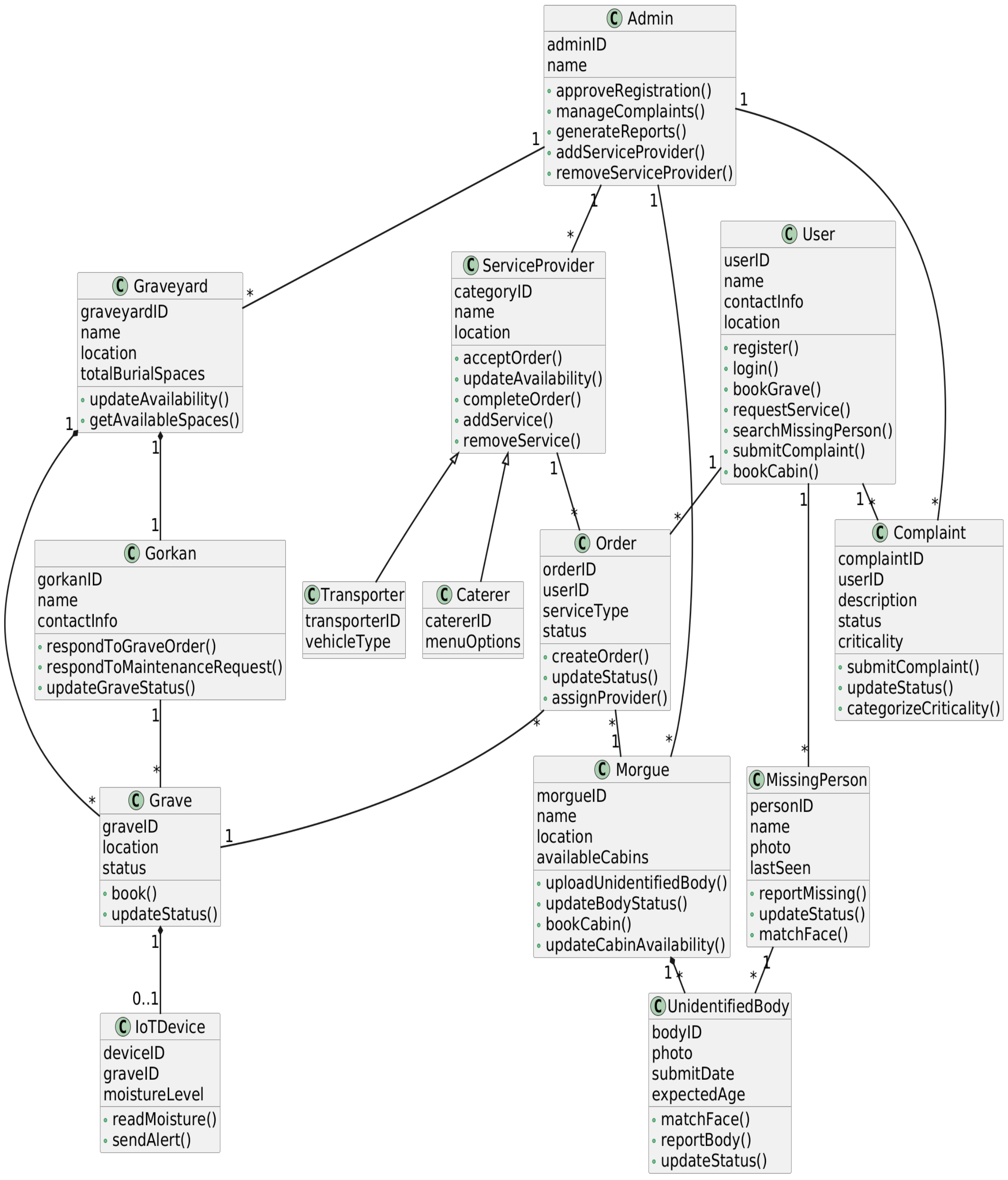


Figure 5.3: Class Diagram Depicting the Structure and Relationships of the Project's Classes

5.6 Policies and Tactics

The following are the considerations for the policies that would be followed in the Akhri Aramgah system:

5.6.1 Coding Guidelines and Conventions

The code will ensured to be consistent so that the readability and maintainability of the code are not negatively affected. Semantics will be taken into account when naming variables and functions along with the popular naming conventions.

5.6.2 Testing Plans

Thorough testing of the system will be conducted and the boundary conditions will especially be checked on each type of user input. The functional and non-functional requirements of the system will be thoroughly tested.

5.6.3 Algorithm Selection

Algorithms will be chosen based on their performance and the system's needs. The non-functional requirements and the system's quality attributes will be considered when choosing algorithms. Using locations and AI-based functionalities, in particular, rely upon this choice of algorithms.

5.6.4 Maintenance Plans

Thanks to our system architecture, a structured maintenance plan will be established, and modules will be updated one by one to ensure the least possible downtime. Regular maintenance is an absolute need in such a system, but policies must be developed to do this without causing the users any problems.

5.7 Conclusion

This chapter examined the design considerations for the Akhri Aramgah system. Details like the system architecture, policies, and strategies were discussed to convey maximum understanding of the project. The components of the entire system were shown through a class diagram. This careful planning sets the tone for a reliable and efficient solution to the funeral and burial services sector.

Chapter 6 Implemenation and Test Cases

6.1 Implementation

This chapter outlines the technical implementation of the core features developed for Akhri Aramgah so far. The project combines modern web and mobile technologies to deliver a seamless and secure experience for users, service providers, and administrators within the Akhri Aramgah environment. Its key areas of focus will include secure user authentication, web and mobile app development, real-time communication, and efficient data management. React.js, Node.js, Java, and Firebase are ensured for smooth functioning while technologies such as Web Sockets make it possible for real-time updates. All the features are developed to be scalable and user-friendly, therefore, having a well-structured system which meets all the requirements of the all the various actors of the system.

6.1.1 Implementation of User Authentication

User authentication in Akhri Aramgah ensures maximum security through JWT tokens and crypt. When a user logs in, the server generates a JWT stored in the browser's local storage to maintain the session for up to three days, after which it expires. Passwords are hashed with crypt before storage, ensuring they are never stored in a readable format, which protects sensitive data. On the frontend, Higher Order Components are used to guard protected routes by checking for a valid JWT token before granting access to pages. If the token is missing, users are sent to the login page. This architecture provides robust protection against unauthorized access, ensuring a secure and seamless experience for authenticated users that gives them adequate security.

6.1.2 Implementation of User Web and Mobile Application

The user web application uses React.js for dynamic interfaces, Bootstrap 5 for responsive design, and vanilla CSS for custom styling and mobile application uses React Native for platform independant application for both IOS and Android. The backend is made using Node.js and Express.js, with a strict MVC (Model-View-Controller) structure for proper management of code. Firebase Real-time Database is used to store data, and the configuration.js file handles the integration. The features that users can access include tracking order statuses, service requests, updating profile information, monitoring graves, requesting maintenance, and viewing order history. Each feature is implemented on separate pages using React Router to leverage React's multipage capabilities. RESTful APIs manage communication between the frontend and back end, ensuring smooth data flow and synchronization.

6.1.3 Implementation of Service Providers' Mobile Apps

Four mobile apps, developed in Java using Android Studio, serve caterers, transporters, morgue staff, and graveyard managers. These apps connect to the user web application through the same Wi-Fi network, ensuring seamless data synchronization with the Node.js backend. Through Web Sockets, real-time communication is established between the users and service providers to ensure that updates are done in real-time. For instance, if a user places an order, it will be sent to the appropriate provider's app immediately through Web Sockets to ensure quick responses. Every app is tailored to the service it supports, meaning that providers can see order details, accept or decline requests, and update order statuses. All data is synchronized with Firebase in real-time, ensuring accuracy and efficiency.

6.1.4 Implementation of Map Integration

The service booking application features an interactive map designed using OpenStreetMap and React Leaflet, enhancing the quality of user experience. However, the map is limited to Lahore as its geographical focus, and it's bounded along the longitude and latitude coordinates of Lahore that ensures that users do not accidentally drag it elsewhere. It also has search bar where any well-known locations in Lahore can easily be searched. These result as reference points for various service requests. Users can select their destination visually, while the OpenStreetMap API calculates distance to the nearest providers. The service providers to be assigned to a client are those that would cut down on travel time.

6.1.5 Implementation of Admin Side Functionalities

The admin web application is also created using React.js with Bootstrap 5 and vanilla css for the front end and the Node.js/Express.js backend that follows the MVC structure to ensure that it remains easily manageable and convenient to update. Its functionality is partially implemented for now, limiting it to the features that the 4 service provider applications might need. The features include creating new graveyards, adding new service providers and viewing the details of the orders and analytics of whole system. Each feature has a dedicated page in react that is connected using react router DOM. The front end and the backend are connected using the RESTful APIs. The system exists also uses Firebase as its database.

6.1.6 Implementation of Graveyard Mapping

Grave mapping was implemented in such a way that it uses a custom algorithm for optimization in allocating graves. The algorithm assumes the rectangular dimensions of the graveyard plot and uses the length and width of each grave along with the spacing between them to make the best possible utilization of space. It creates a dynamic grid layout, ensuring proper spacing and fitting in as many graves as possible into the defined area. It is integrated with Firebase, which updates graveyard layout in real-time. This provides an actual and accurate record of available plots for future bookings, making grave allocation efficient and well-organized. This system also has a separate space in each graveyard for smaller graves to ensure better optimization of space.

6.1.7 Implementation of 360-Degree Graveyard View

To provide a better user experience, a 360-degree view of the graveyard feature was added to the user web application. With a 360-degree camera, we took panoramic photos of various graveyards and gave users an immersive view prior to grave booking. The photos were embedded within the web application using a dedicated React library that offers interactive 360-degree image rendering. Customers are now able to explore the cemeteries virtually. The feature not only enhances transparency but also gives a touch of modernity to the service booking.

6.1.8 Implementation of Facial Recognition (Shanakht Gar)

Shanakht Gar module is the one that deals with facial matching to help with identification of the deceased. We trained the ArcFace model from a dataset containing more than 25,000 labeled images that were categorized under identity. This AI model operates on a Flask server and acts on images passed from the morgue mobile app. As soon as there is a new image uploaded, Shanakht Gar compares it with the available database and reports back to the web-based gallery. This system enhances the speed and accuracy of the identification process, assisting the morgue personnel and administrators in verifying identities efficiently.

6.1.9 Integration with Morgue Service Provider Mobile Application

The morgue app, written in Java and created with Android Studio, takes photographs and uploads them directly to the Flask server that hosts Shanakht Gar. The app itself does not perform facial recognition but allows personnel to take and transmit images with efficiency. All uploaded photos are retained in a centralized gallery on the server for subsequent matching and reference. In addition, the application provides morgue personnel with the ability to control real-time cabin reservations, with updates being synchronized via Web Sockets. This configuration provides efficient communication with the backend and optimizes operations at the morgue.

6.1.10 Implementation of NLP Complaint Escalation (Shikayat Nazim)

The Shikayat Nazim module is used to automate complaint classification and escalation in the user web application. A custom dataset was developed and grown using k-fold augmentation to achieve diversity. We trained a DistilBERT model with transformers library to categorize complaints into meaningful categories like ”critical” and ”non-critical.” After a user registers a complaint, the NLP model analyzes it and sends it automatically to the concerned service provider or admin. The integration allows faster resolution of critical problems. The module is integrated with the backend using RESTful APIs, providing real-time complaint handling and minimizing manual intervention.

6.1.11 Implementaion of Soil Moisture Sensor Module (Khaak Been)

The Khaak Been Module is designed to monitor soil moisture levels of graves efficiently. The module integrates an ESP32 microcontroller, which acts as the core processing unit, interfaced with a Capacitive Soil Moisture Sensor v1.2 for accurate and reliable moisture detection. The system is powered by a combination of a power bank/batteries, ensuring portability and continuous operation in graveyard. The ESP32 is programmed using the Arduino IDE, enabling seamless data collection from the sensor. The acquired soil moisture readings are transmitted wirelessly to a specialized backend Node.js server, which processes and stores the data in a database. Users can access real-time grave soil moisture data on dedicated applications, providing an intuitive interface for monitoring and informed decision-making. This integrated solution offers users to plan timely maintenanace of the graves.

6.1.12 Implementation Overview

The implementation of Akhri Aramgah has evolved into a comprehensive end-to-end funeral service management platform. The user web application is fully developed, offering dynamic interfaces and core features like service booking, profile management, complaint escalation via the Shikayat Nazim NLP module, and a 360-degree graveyard view for informed decision-making.
Mobile applications for service providers ensure real-time communication and efficient order handling. The morgue application is integrated with the Shanakht Gar AI module, a facial recognition system used for identity verification. The admin web application is also complete, supporting critical tasks such as graveyard creation, service provider management, and overall platform oversight.
Enhanced features like interactive graveyard mapping, optimized space planning, and automated complaint classification improve operational efficiency. The Khaak Been IoT module contributes to the system by continuously monitoring grave soil moisture levels.
Together, these components provide a secure, efficient, and user-friendly experience for all stakeholders.

6.2 Test Cases

Following are the test cases designed, developed and conducted for Akhri Aramgah.

User Registration			
TC-001			
Test Case ID:	*1*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 10, 2025*	**Use Case Reference(s):**	*User Registration*
Revision History:	*None*		
Objective:	*To test whether a new user can register successfully on the system.*		
Product/Module:	*User Registration Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*User has a valid email and other required credentials.*		
Pre-Requisite:	*The user is on the registration page.*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to registration page.*	*Registration page is displayed.*	
2	*Fill all required fields with valid data.*	*All fields are filled with valid data.*	
3	*Click "Register" button.*	*Account is created and confirmation message is displayed.*	
Comments: The test case passed successfully. The system is working as per requirements.			
Passed			

Successful Login			
TC-002			
Test Case ID:	*2*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 10, 2025*	**Use Case Reference(s):**	*User Authentication*
Revision History:	*None*		
Objective:	*To verify that a registered user can login successfully with valid credentials.*		
Product/Module:	*Authentication Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*User is registered in the system.*		
Pre-Requisite:	*The user is on the login page.*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to login page.*	*Login page is displayed.*	
2	*Enter valid credentials*	*Credentials are entered.*	
3	*Click "Login" button.*	*User is logged in and redirected to dashboard.*	
Comments: The test case passed successfully. Login functionality works as expected.			
Passed			

Failed Login (Invalid Credentials)			
TC-003			
Test Case ID:	*3*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 10, 2025*	**Use Case Reference(s):**	*User Authentication*
Revision History:	*None*		
Objective:	*To verify that login fails with invalid credentials.*		
Product/Ver/ Module:	*Authentication Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*User is registered in the system.*		
Pre-Requisite:	*The user is on the login page.*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to login page.*	*Login page is displayed.*	
2	*Enter invalid email or password.*	*Invalid credentials are entered.*	
3	*Click "Login" button.*	*Error message displayed: "Invalid username or password".*	
Comments: The test case passed successfully. System handles invalid login attempts.			
Passed			

Failed Registration (Existing Email)			
TC-004			
Test Case ID:	*4*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 10, 2025*	**Use Case Reference(s):**	*User Registration*
Revision History:	*None*		
Objective:	*To verify that registration fails with existing email.*		
Product/Ver/ Module:	*User Registration Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*User with test email already exists.*		
Pre-Requisite:	*The user is on the registration page.*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to registration page.*	*Registration page is displayed.*	
2	*Fill form with existing email.*	*Form is filled with existing email.*	
3	*Click "Register" button.*	*Error message: "Email already registered".*	
Comments: The test case passed successfully. System prevents duplicate email registrations.			
Passed			

Book Grave (Success)			
TC-005			
Test Case ID:	*5*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 11, 2025*	**Use Case Reference(s):**	*Grave Booking*
Revision History:	*None*		
Objective:	*To verify that user can book a grave successfully.*		
Product/Ver/ Module:	*Grave Booking Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*User is logged in.*		
Pre-Requisite:	*User has valid payment details.*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to grave booking section.*	*Grave booking section is displayed.*	
2	*Select available grave plot.*	*Available grave plot is selected.*	
3	*Fill deceased details.*	*Deceased details are filled.*	
4	*Make payment.*	*Payment is processed successfully.*	
5	*Submit booking.*	*Grave is booked and confirmation displayed.*	
Comments: The test case passed successfully. Grave booking process works as expected.			
Passed			

Book Grave (Already Booked)			
TC-006			
Test Case ID:	*6*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 11, 2025*	**Use Case Reference(s):**	*Grave Booking*
Revision History:	*None*		
Objective:	*To verify that booking fails for already booked grave.*		
Product/Ver/ Module:	*Grave Booking Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*Grave plot is already booked.*		
Pre-Requisite:	*User is logged in.*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to grave booking section.*	*Grave booking section is displayed.*	
2	*Select already booked grave plot.*	*Already booked grave plot is selected.*	
3	*Attempt to book.*	*Error message: "Selected grave is already booked".*	
Comments: The test case passed successfully. System prevents booking of reserved graves.			
Passed			

Book Transport (Success)			
TC-007			
Test Case ID:	*7*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 10, 2025*	**Use Case Reference(s):**	*Book Transport*
Revision History:	*None*		
Objective:	*Verify that user can book transport successfully*		
Product/Ver/ Module:	*Transport Booking Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*Transport services are available for booking.*		
Pre-Requisite:	*User is logged in.*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to transport booking section.*	*Transport booking section is displayed.*	
2	*Select transport type and date.*	*Transport type and date are selected.*	
3	*Provide pickup/drop-off details.*	*Pickup/drop-off details are provided.*	
4	*Make Payment.*	*Payment is successfully processed.*	
5	*Submit booking.*	*Transport is booked and confirmation displayed.*	
Comments: The test case passed successfully. The system is working as per requirements.			
Passed			

View Past Orders			
TC-016			
Test Case ID:	*16*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 10, 2025*	**Use Case Reference(s):**	*Order Management*
Revision History:	*None*		
Objective:	*Verify that user can view past completed orders*		
Product/Ver/ Module:	*Orders Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*User is logged in with past orders*		
Pre-Requisite:	*User is logged in with past orders*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to orders section.*	*Orders section is displayed.*	
2	*Click ”Past Orders” tab.*	*List of past orders displayed with details.*	
Comments: The test case passed. User can successfully view past completed orders.			
Passed			

Book Transport (Invalid Date)			
TC-008			
Test Case ID:	*8*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 10, 2025*	**Use Case Reference(s):**	*Book Transport*
Revision History:	*None*		
Objective:	*Verify that transport booking fails for past date*		
Product/Ver/ Module:	*Transport Booking Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*System validates booking dates.*		
Pre-Requisite:	*User is logged in.*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to transport booking section.*	*Transport booking section is displayed.*	
2	*Select transport type.*	*Transport type is selected.*	
3	*Enter past date.*	*Past date is entered.*	
4	*Attempt to book.*	*Error message: ”Date cannot be in the past”*	
Comments: The test case passed successfully. The system properly validates date inputs.			
Passed			

Book Morgue (Success)			
TC-009			
Test Case ID:	*9*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 10, 2025*	**Use Case Reference(s):**	*Book Morgue*
Revision History:	*None*		
Objective:	*Verify that user can book morgue successfully*		
Product/Ver/ Module:	*Morgue Booking Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*Morgue facilities are available for booking.*		
Pre-Requisite:	*User is logged in.*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to morgue booking section.*	*Morgue booking section is displayed.*	
2	*Select morgue facility.*	*Morgue facility is selected.*	
3	*Enter deceased details.*	*Deceased details are entered.*	
4	*Make payment.*	*Payment is successfully processed.*	
5	*Submit booking.*	*Morgue is booked and confirmation displayed.*	
Comments: The test case passed successfully. The system is working as per requirements.			
Passed			

Book Morgue (Capacity Full)			
TC-010			
Test Case ID:	*10*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 10, 2025*	**Use Case Reference(s):**	*Book Morgue*
Revision History:	*None*		
Objective:	*Verify that morgue booking fails when capacity is full*		
Product/Ver/ Module:	*Morgue Booking Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*System tracks morgue capacity accurately.*		
Pre-Requisite:	*Morgue is at full capacity.*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to morgue booking section.*	*Morgue booking section is displayed.*	
2	*Select full morgue facility.*	*Full morgue facility is selected.*	
3	*Attempt to book.*	*Error message: "Selected morgue is at full capacity"*	
Comments: The test case passed successfully. The system validates morgue capacity.			
Passed			

Book Caterer (Success)			
TC-011			
Test Case ID:	*11*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 10, 2025*	**Use Case Reference(s):**	*Book Caterer*
Revision History:	*None*		
Objective:	*Verify that user can book caterer successfully*		
Product/Ver/ Module:	*Caterer Booking Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*Catering services are available for booking.*		
Pre-Requisite:	*User is logged in.*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to caterer booking section.*	*Caterer booking section is displayed.*	
2	*Select location.*	*Location is selected.*	
3	*Select menu.*	*Menu is selected.*	
4	*Make Payment.*	*Payment is successfully processed.*	
5	*Submit booking.*	*Caterer is booked and confirmation displayed.*	
Comments: The test case passed successfully. The system is working as per requirements.			
Passed			

Search Morgue (Success)			
TC-012			
Test Case ID:	*12*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 10, 2025*	**Use Case Reference(s):**	*Search Morgue*
Revision History:	*None*		
Objective:	*Verify that user can search for unidentified bodies*		
Product/Ver/ Module:	*Morgue Search Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*Image recognition system is working properly.*		
Pre-Requisite:	*Unidentified bodies exist in system.*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to morgue search section.*	*Morgue search section is displayed.*	
2	*Enter user image.*	*User image is uploaded.*	
3	*Click search button.*	*List of matching unidentified body displayed.*	
Comments: The test case passed successfully. The search functionality works as expected.			
Passed			

Search Morgue (No Results)			
TC-013			
Test Case ID:	*13*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 10, 2025*	**Use Case Reference(s):**	*Search Morgue*
Revision History:	*None*		
Objective:	*Verify proper response when no matches found*		
Product/Ver/ Module:	*Morgue Search Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*No bodies match search criteria*		
Pre-Requisite:	*User is logged in*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to morgue search section.*	*Morgue search section is displayed.*	
2	*Enter user image.*	*User image is uploaded.*	
3	*Click search button.*	*Message displayed: ”No matching records found”*	
Comments: The test case passed successfully. The system handles no-result scenarios.			
Passed			

Register Complaint (Success)			
TC-014			
Test Case ID:	*14*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 10, 2025*	**Use Case Reference(s):**	*Complaint Register*
Revision History:	*None*		
Objective:	*Verify that user can register a complaint*		
Product/Ver/ Module:	*Complaint Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*User has valid complaint details*		
Pre-Requisite:	*User is logged in*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to complaints section.*	*Complaints section is displayed.*	
2	*Fill complaint form.*	*Complaint form is filled.*	
3	*Submit complaint.*	*Complaint is registered and confirmation displayed.*	
Comments: The test case passed successfully. The complaint registers as expected.			
Passed			

Register Complaint (Empty Fields)			
TC-015			
Test Case ID:	*15*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 10, 2025*	**Use Case Reference(s):**	*Complaint Register*
Revision History:	*None*		
Objective:	*Verify that complaint fails with empty required fields*		
Product/Ver/ Module:	*Complaint Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*User is logged in*		
Pre-Requisite:	*User has access to complaint form*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to complaints section.*	*Complaints section is displayed.*	
2	*Leave required fields empty.*	*Required fields are not filled.*	
3	*Attempt to submit.*	*Error message: ”Please fill all required fields”*	
Comments: The test case passed. Validation for empty fields is working as expected.			
Passed			

View Pending Orders			
TC-017			
Test Case ID:	*17*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 10, 2025*	**Use Case Reference(s):**	*Order Management*
Revision History:	*None*		
Objective:	*Verify that user can view pending orders*		
Product/Ver/ Module:	*Orders Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*User is logged in with pending orders*		
Pre-Requisite:	*User is logged in with pending orders*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to home page.*	*Home page is displayed.*	
2	*Click ”Pending Orders” tab.*	*List of pending orders displayed with status.*	
Comments: The test case passed. User can successfully view pending orders.			
Passed			

View Active Orders			
TC-018			
Test Case ID:	*18*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 10, 2025*	**Use Case Reference(s):**	*Order Management*
Revision History:	*None*		
Objective:	*Verify that user can view active orders*		
Product/Ver/ Module:	*Orders Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*User is logged in with active orders*		
Pre-Requisite:	*User is logged in with active orders*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to home page.*	*Home page is displayed.*	
2	*Click ”Active Orders” tab.*	*List of active orders displayed with details.*	
Comments: The test case passed. User can successfully view active orders.			
Passed			

Change Username (Success)			
TC-019			
Test Case ID:	*19*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 10, 2025*	**Use Case Reference(s):**	*Profile Management*
Revision History:	*None*		
Objective:	*Verify that user can change username successfully*		
Product/Ver/ Module:	*User Profile Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*User is logged in*		
Pre-Requisite:	*User is logged in*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to settings section.*	*Settings section is displayed.*	
2	*Select "Change Username".*	*Username change form is displayed.*	
3	*Enter new username.*	*Form is filled with required information.*	
4	*Submit changes.*	*Username is updated and confirmation displayed.*	
Comments: The test case passed. Username change functionality works as expected.			
Passed			

Change Username (Existing Username)			
TC-020			
Test Case ID:	*20*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 10, 2025*	**Use Case Reference(s):**	*Profile Management*
Revision History:	*None*		
Objective:	*Verify that username change fails if new username exists*		
Product/Ver/ Module:	*User Profile Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*New username already exists in system*		
Pre-Requisite:	*User is logged in and new username already exists in system*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to settings section.*	*Settings section is displayed.*	
2	*Select "Change Username".*	*Username change form is displayed.*	
3	*Enter existing username.*	*Form is filled with existing username.*	
4	*Attempt to submit.*	*Error message: "Username already taken"*	
Comments: The test case passed. Validation for existing username is working as expected.			
Passed			

Change Password (Success)			
TC-021			
Test Case ID:	*21*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 10, 2025*	**Use Case Reference(s):**	*Profile Management*
Revision History:	*None*		
Objective:	*Verify that user can change password successfully*		
Product/Ver/ Module:	*User Profile Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*User is logged in*		
Pre-Requisite:	*User is logged in*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to settings section.*	*Settings section is displayed.*	
2	*Select "Change Password".*	*Password change form is displayed.*	
3	*Enter current and new password.*	*Form is filled with required information.*	
4	*Submit changes.*	*Password is updated and confirmation displayed.*	
Comments: The test case passed. Password change functionality works as expected.			
Passed			

Logout (Success)			
TC-027			
Test Case ID:	*27*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 10, 2025*	**Use Case Reference(s):**	*Profile Management*
Revision History:	*None*		
Objective:	*Verify that user can logout successfully*		
Product/Ver/ Module:	*User Profile Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*User is logged in*		
Pre-Requisite:	*User is logged in*		
Step No.	**Execution description**	**Procedure result**	
1	*Click logout button.*	*User is logged out and redirected to login page.*	
Comments: The test case passed. Logout functionality works as expected.			
Passed			

Change Password (Incorrect Current Password)			
TC-022			
Test Case ID:	*22*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 10, 2025*	**Use Case Reference(s):**	*Profile Management*
Revision History:	*None*		
Objective:	*Verify that password change fails with wrong current password*		
Product/Ver/ Module:	*User Profile Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*User is logged in*		
Pre-Requisite:	*User is logged in*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to settings section.*	*Settings section is displayed.*	
2	*Select ”Change Password”.*	*Password change form is displayed.*	
3	*Enter incorrect current password.*	*Form is filled with incorrect password.*	
4	*Attempt to submit.*	*Error message: ”Current password is incorrect”*	
Comments: The test case passed. Validation for incorrect current password is worked.			
Passed			

Change Email (Success)			
TC-023			
Test Case ID:	*23*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 10, 2025*	**Use Case Reference(s):**	*Profile Management*
Revision History:	*None*		
Objective:	*Verify that user can change email successfully*		
Product/Ver/ Module:	*User Profile Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*User is logged in*		
Pre-Requisite:	*User is logged in*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to settings section.*	*Settings section is displayed.*	
2	*Select ”Change Email”.*	*Email change form is displayed.*	
3	*Enter new email.*	*Form is filled with required information.*	
4	*Submit changes.*	*Email is updated and confirmation displayed.*	
Comments: The test case passed. Email change functionality works as expected.			
Passed			

Change Email (Existing Email)			
TC-024			
Test Case ID:	*24*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 10, 2025*	**Use Case Reference(s):**	*Profile Management*
Revision History:	*None*		
Objective:	*Verify that email change fails if new email exists*		
Product/Ver/ Module:	*User Profile Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*New email already exists in system*		
Pre-Requisite:	*User is logged in and new email already exists in system*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to settings section.*	*Settings section is displayed.*	
2	*Select "Change Email".*	*Email change form is displayed.*	
3	*Enter existing email.*	*Form is filled with existing email.*	
4	*Attempt to submit.*	*Error message: "Email already registered"*	
Comments: The test case passed. Validation for existing email is working as expected.			
Passed			

Change Phone (Success)			
TC-025			
Test Case ID:	*25*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 10, 2025*	**Use Case Reference(s):**	*Profile Management*
Revision History:	*None*		
Objective:	*Verify that user can change phone number successfully*		
Product/Ver/ Module:	*User Profile Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*User is logged in*		
Pre-Requisite:	*User is logged in*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to settings section.*	*Settings section is displayed.*	
2	*Select "Change Phone".*	*Phone change form is displayed.*	
3	*Enter new phone number.*	*Form is filled with required information.*	
4	*Submit changes.*	*Phone number is updated and confirmation displayed.*	
Comments: The test case passed. Phone number change functionality works as expected.			
Passed			

Change Phone (Invalid Format)			
TC-026			
Test Case ID:	*26*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 10, 2025*	**Use Case Reference(s):**	*Profile Management*
Revision History:	*None*		
Objective:	*Verify that phone change fails with invalid format*		
Product/Ver/ Module:	*User Profile Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*User is logged in*		
Pre-Requisite:	*User is logged in*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to settings section.*	*Settings section is displayed.*	
2	*Select "Change Phone".*	*Phone change form is displayed.*	
3	*Enter invalid phone number format.*	*Form is filled with invalid information.*	
4	*Attempt to submit.*	*Error message: "Invalid phone number format"*	
Comments: The test case passed. Invalid phone number format is properly detected.			
Passed			

View Grave Booking Details			
TC-028			
Test Case ID:	*28*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 10, 2025*	**Use Case Reference(s):**	*Profile Management*
Revision History:	*None*		
Objective:	*Verify that user can view grave booking details*		
Product/Ver/ Module:	*User Profile Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*User has active grave booking*		
Pre-Requisite:	*User has active grave booking*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to grave orders.*	*Grave orders section is displayed.*	
2	*Select grave booking.*	*Detailed information about grave booking displayed.*	
Comments: The test case passed. Grave booking details are properly displayed.			
Passed			

Cancel Grave Booking (Success)			
TC-029			
Test Case ID:	*29*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 10, 2025*	**Use Case Reference(s):**	*Profile Management*
Revision History:	*None*		
Objective:	*Verify that user can cancel grave booking*		
Product/Ver/ Module:	*User Profile Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*User has active grave booking*		
Pre-Requisite:	*User has active grave booking*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to active orders.*	*Active orders section is displayed.*	
2	*Select grave booking.*	*Grave booking details are displayed.*	
3	*Click cancel button.*	*Cancellation confirmation dialog is displayed.*	
4	*Confirm cancellation.*	*Booking is cancelled and status updated.*	
Comments: The test case passed. Grave booking cancellation functionality worked.			
Passed			

View Complaint Details			
TC-030			
Test Case ID:	*30*	**QA Test Engineer:**	*Abdullah Asim*
Test case Version:	*1.0*	**Reviewed By:**	*Shehroz Aziz*
Test Date:	*April 10, 2025*	**Use Case Reference(s):**	*Profile Management*
Revision History:	*None*		
Objective:	*Verify that user can view complaint details*		
Product/Ver/ Module:	*User Profile Module*		
Environment:	*Internet is connected and Website is running on a compatible browser. The backend of the system is fully operational.*		
Assumptions:	*User has submitted complaints*		
Pre-Requisite:	*User has submitted complaints*		
Step No.	**Execution description**	**Procedure result**	
1	*Navigate to complaints section.*	*Complaints section is displayed.*	
2	*Select complaint.*	*Detailed complaint information displayed.*	
Comments: The test case passed. Complaint details are properly displayed.			
Passed			

6.3 Test Metrics

Table 6.1: Test Case Metrics

Metric	**Value**
Number of Test Cases	30
Number of Test Cases Passed	30
Number of Test Cases Failed	0
Test Case Defect Density	0
Test Case Effectiveness	0
Traceability Matrix	Traceability is maintained between requirements and test cases through unique IDs.

Chapter 7 User Manual

Following section is the user manual of our platform encompassing the user apps, admin app , service provider apps and the staff apps.

7.1 User

7.1.1 Sign Up

Follow these steps to sign up:

- Click on the Sign Up button on the landing page.
- Enter your personal details and password.
- Click on the Sign Up button.

7.1.2 Sign In

Follow these steps to sign in:

- Click on the Sign In button on the landing page.
- Enter your email and password.
- Click on the Sign In button.

7.1.3 Book Grave

Follow these steps to book a grave:

- Click on the Book Grave button on the user dashboard.
- Choose a graveyard from the list or search for one and click on the Next button.
- Check the panorama shot view of the graveyard and click on the Next button.
- Select a grave spot from the available graves marked as green by clicking on them and click on the Next button.
- Enter the details about the deceased and click the Next button.
- Enter your payment details and click on the Book button.

7.1.4 Book Morgue

Follow these steps to book a morgue cabin:

- Click on the Book Morgue button on the user dashboard.
- Choose a morgue from the list or search for one and click on the Next button.
- Enter details about the deceased and click on the Next button.
- Enter your payment details and click on the Book button.

7.1.5 Book Catering

Follow these steps to book catering for the funeral:

- Click on the Book Catering button on the user dashboard.
- Select the destination pin from the interactive map that pops up and click on the Next button.
- Select the menu items by using the counter on the right of them and click on the Next button.
- Enter your payment details and click on the Book button.

7.1.6 Book Transportation

Follow these steps to book transportation for the corpse:

- Click on the Book Transport button on the user dashboard.
- Select the source pin from the interactive map that pops up and click on the Next button.
- Select the destination pin from the interactive map that pops up and click on the Next button.
- Check the calculated distance and fare.
- Enter your payment details and click on the Book button.

7.1.7 Search Morgue

Follow these steps to search the morgue for any missing individuals:

- Click on the Search Morgue button on the user dashboard.
- Upload an image of the missing person by clicking on the Upload button and choosing an image from your device.
- If a match is found, the image will be shown to you along with the name of the morgue.

7.1.8 View Order History

Follow these steps to view order history:

- Click on the Orders button in the sidebar.
- The orders will be visible to you along with their details.

7.1.9 View Graves

Follow these steps to view graves associated with your account:

- Click on the Graves button in the sidebar.
- The graves will be visible to you with their details.

7.1.10 Schedule Grave Maintenance

Follow these steps to schedule grave maintenance:

- Click on the Graves button in the sidebar.
- The graves will be visible to you with their details and condition.
- Click on the Schedule Maintenance button.
- Enter your payment details and click on the Schedule button to book a maintenance.

7.1.11 Register Complaint

Follow these steps to register complaints:

- Click on the Complaints button in the sidebar.
- Click on the Register Complaint button.
- Choose the type of complaint.
- Fill in the information about the complaint.
- Click on the Submit button.

7.1.12 Track Complaints

Follow these steps to track complaints:

- Click on the Complaints button in the sidebar.
- You can see your complaints and their statuses on the page.

7.1.13 Update Settings

Follow these steps to update settings:

- Click on the Settings button in the sidebar.
- Choose which information to update.
- Enter new information.
- Click on the Update button.

7.2 Gorkan

7.2.1 Sign In

Follow these steps to sign in:

- Enter graveyard ID.
- Enter password.
- Click on the Sign In button.

7.2.2 View Graveyard

Follow these steps to view graveyard:

- Navigate to the graveyard tab.
- The graveyard will be displayed in a 2D grid.

7.2.3 Prepare Grave Confirmation

Follow these steps to do a grave confirmation:

- Navigate to the grave orders tab.
- Slide right on the order card.
- From the options, choose confirm.

7.2.4 Grave Maintenance Confirmation

Follow these steps to do a grave maintenance confirmation:

- Navigate to the grave maintenance tab.

- Slide right on the card.
- From the options, choose confirm.

7.3 Morgue Manager

7.3.1 Sign In

Follow these steps to sign in:

- Enter morgue ID.
- Enter password.
- Click on the Sign In button.

7.3.2 Prepare Cabin Confirmation

Follow these steps to confirm a cabin booking request:

- When an order is placed, a pop-up will appear.
- On the popup, choose confirm.

7.3.3 Upload Image

Follow these steps to upload an image of the unidentified body:

- Click on the upload button.
- Choose from the gallery or take a live shot of the face.

7.3.4 Delete Image

Follow these steps to delete an image of the body that has been identified:

- Click on the identified image from the grid.
- Choose the delete option.

7.4 Catering Service Provider

7.4.1 Sign In

Follow these steps to sign in:

- Enter the registered phone number.
- Enter the password.
- Click on the login button.

7.4.2 Book Order

Follow these steps to book an order:

- From the list, choose an order and swipe right.
- You can see details about the order so check if it is suitable to you.
- Confirm booking.

7.4.3 Complete Order

Follow these steps to mark an order as completed:

- On the order screen, the order details will be visible to you.
- Click on the complete order button and do the confirmation.

7.4.4 Change Details

Follow these steps to change account details:

- Navigate to the profile page.
- Choose the information to update.
- Confirm old password.
- Enter new details in the input field.
- Confirm change.

7.5 Transportation Service Provider

7.5.1 Sign In

Follow these steps to sign in:

- Enter the registered phone number.
- Enter the password.

- Click on the login button.

7.5.2 Book Order

Follow these steps to book an order:

- From the list, choose an order and swipe right.
- You can see details about the order so check if it is suitable to you.
- Confirm booking.

7.5.3 Complete Order

Follow these steps to mark an order as completed:

- On the order screen, the order details will be visible to you.
- Click on the complete order button and do the confirmation.

7.5.4 Change Details

Follow these steps to change account details:

- Navigate to the profile page.
- Choose the information to update.
- Confirm old password.
- Enter new details in the input field.
- Confirm change.

7.6 Admin

7.6.1 Add Graveyard

Follow these steps to add a graveyard:

- Click on Add Graveyard button on the admin dashboard.
- Choose the location for the graveyard on the map.
- Enter the dimensions of the graveyard and generate a layout by clicking on the generate button.
- Enter graveyard details.

- Click on the submit button.

7.6.2 Add Morgue

Follow these steps to add a morgue:

- Click on Add Morgue button on the admin dashboard.
- Choose the location for the morgue on the map.
- Enter morgue details.
- Click on the submit button.

7.6.3 Review Complaints

Follow these steps to review complaints:

- Click on Review Complaints button on the admin dashboard.
- Click on a complaint to review.
- In the popup window, announce a verdict.

7.6.4 Edit Graveyard

Follow these steps to edit graveyard information:

- Click on the graveyard button in the navigation bar.
- Search and choose a graveyard.
- Click on the edit button.
- Enter new information.
- Click on the save button.

7.6.5 Search Graves

Follow these steps to search information about the grave:

- Click on the graves button in the navigation bar.
- Search for a grave by name or CNIC number of the deceased.
- The information will be visible to you.

7.6.6 Add Service Providers

Follow these steps to manage service providers:

- Click on the service providers button in navigation bar.
- Choose the add service provider button.
- Select the type and enter details.
- Click on the confirm button.

7.6.7 Ban/Warn Service Providers

Follow these steps to ban service providers:

- Click on the service provider button in navigation bar.
- Search for a service provider by ID or name.
- Choose the ban service provider option or warn service provider option.

7.6.8 View Analytics

Follow these steps to view analytics:

- Click on the analytics button in the navigation bar.
- Different graphs can be viewed and monitored.

Chapter 8 Conclusion

This chapter is the concluding chapter of Akhri Aramgah. The project is aimed at introducing digital revolution in the funeral and burial service industry. Akhri Aramgah integrates real-time communication between the user and the provider apps streamlining the process of order placement and uses smart features like AI-based facial recognition, IoT grave monitoring and NLP based complaint escalation. The following part provides a summary on the scope and future advancements for this project.

The funeral and burial services industry is a huge sector and just like every other sector, it also needs innovation and digitization. Our platform Akhri Aramgah, is an entire ecosystem of connected web and mobile applications that cater to the various use cases and stakeholders of this sector. With the use of modern tools and technologies, our platform reduces the amount of manual effort needed to avail the services within this sector and provided an easy to use bridge between the service providers and their target clientele.

Before we started the development, we conducted thorough research to check if a similar solution was already in the market but our search came up dry as this sector has been deprived of any technological integration. We refined our idea and shortlisted the features that we wanted to add to this platform and we documented them in great detail. We documented every little detail including the use cases , software requirements and low level design of the system as well as identified different modules of the project and documented their implementation details.

Akhri Aramgah consists of many different inter-connected web and mobile based applications targeting the users, admins, service providers and authorities. We ensured that each stakeholder has its own application in order to simplify things and to ensure separation of concern and modularity. The platform makes use of AI to search morgues for missing individuals. One module uses IoT to monitor grave condition and generate alerts. The project also uses NLP to classify user complaints as critical or non-critical. This all once connected gives the experience of a large ecosystem that seems intuitive and easy to use for everyone involved.

For future work, the project has space for many new features that may include data analytics and data gathering as this is a sector that has very little digital records and the uniqueness of this platform makes this a great opportunity to gather data. Admin side functionalities can be extended to include things like human resource management and payroll management. Optimizing the platform for high user load is also a problem that can be tackled. In conclusion, Akhri Aramgah leaves room for further innovation as well.

Bibliography

[1] OpusXenta, "Opusxenta - arrangements and bookings." Available: https://opusxenta.com/features/arrangements-and-bookings/. [Accessed: 26-Sep-2024].

[2] PlotBox, "Plotbox - cemetery management software." Available: https://plotbox.com/. [Accessed: 26-Sep-2024].

[3] Everplans, "Everplans - end-of-life planning made easy." Available: https://www.everplans.com/. [Accessed: 26-Sep-2024].

[4] Cemify, "Cemify - simplifying cemetery management." Available: https://www.cemify.com/. [Accessed: 26-Sep-2024].

[5] Chronicle, "Chronicle - memorialization and cemetery management." Available: https://chronicle.rip/. [Accessed: 26-Sep-2024].

[6] F. A. Grave, "Find a grave - millions of cemetery records." Available: https://www.findagrave.com/. [Accessed: 26-Sep-2024].

[7] Docufree, "Cloud records and mapping for cemeteries." Available: https://cemeteryfind.com/. [Accessed: 4-Dec-2024].

[8] BillionGraves, "Billiongraves - the world's largest resource for grave records." Available: https://billiongraves.com/. [Accessed: 26-Sep-2024].

[9] NtechLab, "Findface multi - a new approach to facial recognition." Available: https://ntechlab.com/findface-multi/. [Accessed: 26-Sep-2024].

[10] Recognito, "Face biometrics id verification solutions." Available: https://recognito.vision/. [Accessed: 4-Dec-2024].

[11] H. DS, "Complaint assistant app." Available: https://github.com/haolin-ds/ComplaintAssistantApp. [Accessed: 9-Oct-2024].

[12] J. Leong, "Iot agriculture monitoring system." Available: https://github.com/jxwleong/iot-agriculturue-monitoring-system. [Accessed: 9-Oct-2024].