



Assignment 3: Smart City Management Platform

CS346: Software Engineering Laboratory

Assignment-3 Report

Group 2A Task:

Develop an integrated platform designed to improve urban living through technological innovation and interconnectivity.

Authors:

Group 2A

Instructor:

Prof. Pradip K. Das, Dept. of CSE, IITG

Contents

| | Page |
|--|----------|
| 1 <u>Problem</u> | 4 |
| 1.1 Problem Statement | 4 |
| 1.2 Goal and Motivation | 4 |
| 2 <u>Software Requirements</u> | 4 |
| 3 <u>RAD Model</u> | 5 |
| 4 <u>Team Division</u> | 6 |
| 5 <u>Detailed Pathway of the Solution</u> | 7 |
| 5.1 User registration & Management | 7 |
| 5.1.1 Introduction | 7 |
| 5.1.2 Features | 7 |
| 5.1.3 Entity Relationship Diagram | 8 |
| 5.1.4 Data Flow Diagram | 8 |
| 5.1.5 Interface | 9 |
| 5.2 Professional Services Directory | 12 |
| 5.2.1 Introduction | 12 |
| 5.2.2 Features | 12 |
| 5.2.3 Entity Relationship Diagram | 13 |
| 5.2.4 Data Flow Diagram | 14 |
| 5.2.5 Interface | 15 |
| 5.3 Employment Portal | 16 |
| 5.3.1 Introduction | 16 |
| 5.3.2 Features | 17 |
| 5.3.3 Entity Relationship Diagram | 17 |
| 5.3.4 Data Flow Diagram | 18 |
| 5.3.5 Interface | 19 |

| | | |
|-------|--|----|
| 5.4 | Education Management | 21 |
| 5.4.1 | Introduction | 21 |
| 5.4.2 | Features | 21 |
| 5.4.3 | Entity Relationship Diagram | 23 |
| 5.4.4 | Data Flow Diagram | 24 |
| 5.4.5 | Interface | 28 |
| 5.5 | Healthcare Management | 32 |
| 5.5.1 | Introduction | 32 |
| 5.5.2 | Features | 32 |
| 5.5.3 | Entity Relationship Diagram | 33 |
| 5.5.4 | Data Flow Diagram | 34 |
| 5.5.5 | User Interface of Module | 38 |
| 5.6 | Transportation Management | 40 |
| 5.6.1 | Introduction | 40 |
| 5.6.2 | Features | 41 |
| 5.6.3 | Entity Relationship Diagram | 42 |
| 5.6.4 | Data Flow Diagram | 45 |
| 5.6.5 | Interface | 49 |
| 5.7 | Festival and Event Planning Module | 54 |
| 5.7.1 | Introduction | 54 |
| 5.7.2 | Features | 54 |
| 5.7.3 | Entity Relationship Diagram | 55 |
| 5.7.4 | Data Flow Diagram | 56 |
| 5.7.5 | Interface | 59 |
| 5.8 | Complaint and Redressal Portal | 62 |
| 5.8.1 | Introduction | 62 |
| 5.8.2 | Features | 62 |

| | | |
|----------|--|------------|
| 5.8.3 | Entity Relationship Diagram | 68 |
| 5.8.4 | Data Flow Diagram | 69 |
| 5.9 | Administrative Hierarchy Interface | 71 |
| 5.9.1 | Introduction | 71 |
| 5.9.2 | Features | 71 |
| 5.9.3 | Entity Relationship Diagram | 73 |
| 5.9.4 | Data Flow Diagram | 75 |
| 5.9.5 | Interface | 81 |
| 5.10 | Library | 90 |
| 5.10.1 | Introduction | 90 |
| 5.10.2 | Features | 91 |
| 5.10.3 | Entity Relationship Diagram | 97 |
| 5.10.4 | Data Flow Diagram | 98 |
| 5.11 | Banking | 99 |
| 5.11.1 | Introduction | 99 |
| 5.11.2 | Features | 99 |
| 5.11.3 | Entity Relationship Diagram | 100 |
| 5.11.4 | Data Flow Diagram | 101 |
| 5.11.5 | Interface | 102 |
| 6 | Further Improvements | 105 |
| 6.1 | User Registration and Management | 105 |
| 6.2 | Transportation Management | 105 |
| 6.3 | Administrative Hierarchy Interface | 105 |
| 6.4 | Banking | 106 |
| 6.5 | Extra Features: | 106 |
| 7 | Conclusion | 106 |

1 Problem

1.1 Problem Statement

The Smart City Management System is an integrated platform designed to improve urban living through technological innovation and interconnectivity. It centralizes essential services, providing residents with easy access to a wide range of professionals like doctors and electricians. Educational institutions, employment opportunities, and healthcare services are seamlessly integrated, enhancing the city's efficiency and the welfare of its inhabitants.

1.2 Goal and Motivation

The motivation here is to create a software where the specified features are integrated to provide a great experience to the user during the stay in the Smart City. We have broken down the various features that can be provided into multiple ministries and grouped similar features into modules. Students were divided into groups to take up the responsibility of one or several modules.

2 Software Requirements

The software requirements from each module are listed below:

1. **User Registration & Management:** This module should create a way for users to create a profile on the platform, manage their profile and gain access to the several services that are available on the platform. For this purpose, we implement a dashboard showing all the services available.
2. **Professional Services Directory:** This module deals with the professional services provided by the users or citizens of the Smart City. These services can be further broken down into two kinds, Employment and Regular services. The employment portal should be the place for the administrators to post their requirements for a job and for the citizens to seamlessly look for jobs that match their skills. The regular services booking portal is for the service providers like electricians, plumbers, cooks etc. They should be able to give their working days and the citizens should be able to book their services whenever the service providers are available.
3. **Educational Institutions Interface:** As the name suggests, this module deals with the educational and academic institutions of the Smart City. There are two kinds of academic institutions, schools and e-learning services. The portal should provide a citizen with a list of schools in the locality and the feature of taking admission in them. As a part of e-learning, the portal should also contain a feature of joining the available online courses and take online examinations to receive certificates.
4. **Healthcare Management:** This module deals with the healthcare of the Smart city providing list of hospitals. It should provide a feature of booking appointments with the doctor, buying medicines, ambulance services and tracking the health of the city.

5. **Transportation Management:** This module deals with the transport facilities of the Smart city. It must contain features having the routes and timing of public bus facilities and ride sharing services like car pooling etc. On further note, we can also implement features that take care of toll booth and driving license.
6. **Festival & Event Planning:** This module deals with organizing festivals and events in the Smart city. It is responsible for having features that help in booking venue, vendors and people that will be held accountable for conducting the event.
7. **Complaint & Redressal Portal:** This module is made for taking in the complaints of the citizens of the Smart city. Since it is a huge city with many ministries and modules, some issues are bound to arise. This module comes into picture when citizens decide to make a complaint regarding the failure of any feature in the Smart city platform. This module immediately rectifies and solve the error for the ease of it's users.
8. **Administrative Hierarchy Interface:** This module is mainly responsible for conducting free and fair elections for the Smart city. Apart from this, it should also have features to help the users understand the organizational structure of the city.

Apart from the above mentioned modules, we have also found some other modules that can be added to improve the Smart City Management Platform. These modules are listed below:

1. **Library:** This module is responsible for creating an online library and catering to the needs of the readers and writers of the Smart city.
2. **Banking:** We know Smart City has a lot of flow of money with multiple services being offered. The transactions can be taken care of by having a bank which is implemented by the banking module.

3 RAD Model

The RAD (Rapid Application Development) model is a type of incremental software development process that prioritizes rapid prototyping and iteration over long planning and design phases. It emerged in the 1980s as a response to the shortcomings of traditional waterfall methodologies, which often led to lengthy development cycles and limited stakeholder involvement until the later stages of development.

We have adopted this model for the development of **Smart City Management System** because of the following reasons:

- The complete project's development timeframe was **under a month**, necessitating *Rapid Development*.
- Extensive **modularity** was observed in the software, as elaborated in subsequent sections, fulfilling a crucial requirement for RAD model adoption.
- With a team size of 31 members, ample human resources were available, enabling the effectiveness of RAD paradigm.

4 Team Division

The teams were created based on the interest and compatibility of students. We have made a team for each module. Some teams are handling more than one module. Below is the project organization chart that explains the division of students into teams.

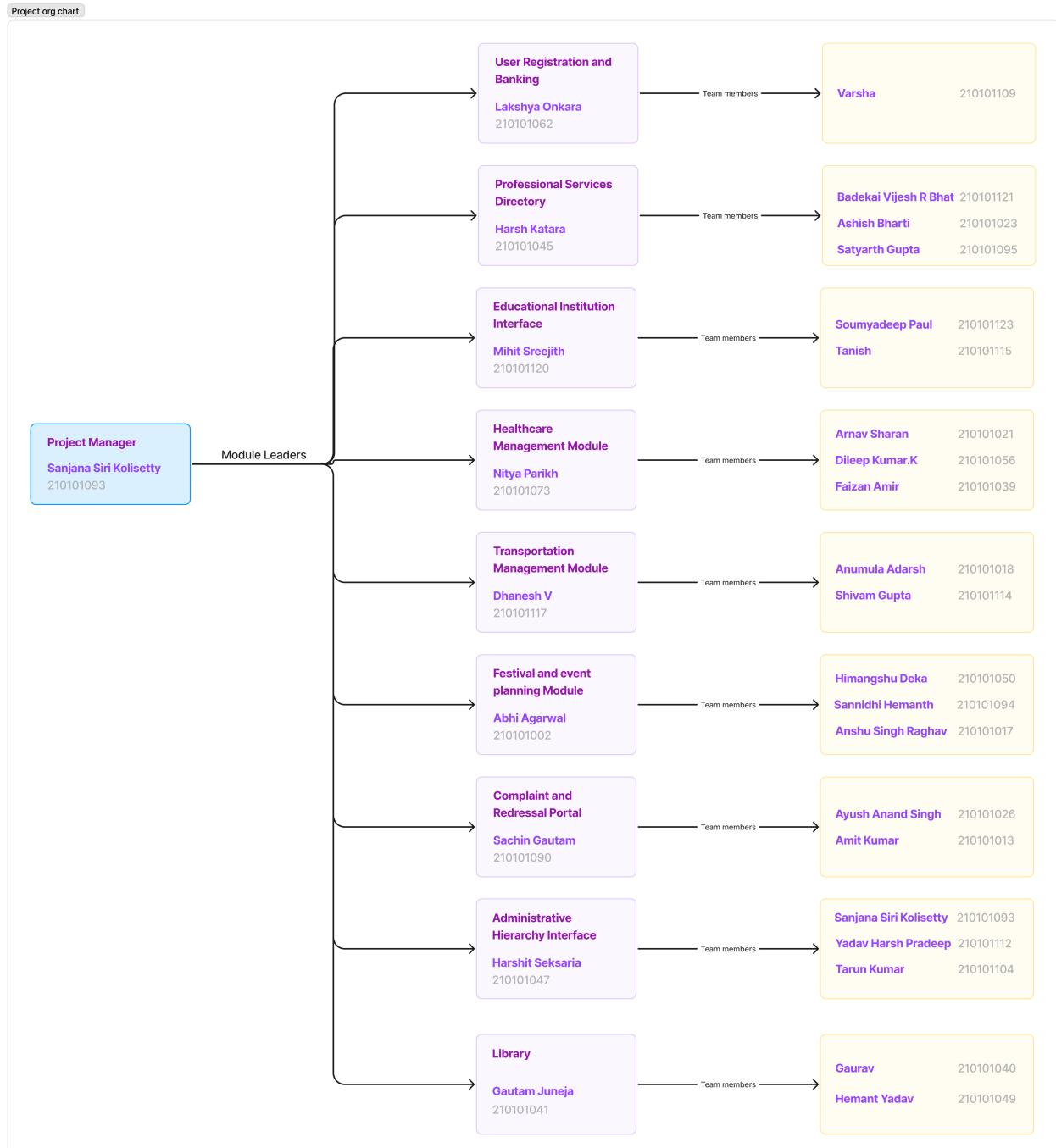


Figure 1: Project Organization Chart

5 Detailed Pathway of the Solution

5.1 User registration & Management

5.1.1 Introduction

The User Registration and Management module is an essential component of the Smart City Management System, providing users with a seamless experience to access and manage their accounts. This module ensures secure authentication, facilitates account recovery, and enables users to maintain their profile information. It acts as a gateway for residents to create profiles, manage personal details, and access various city services. Additionally, it provides a secure login mechanism for module admins, with passwords being periodically updated for heightened security.

5.1.2 Features

1. **Login Page:** Users can access the Smart City Portal by logging in with their credentials. The system verifies the entered credentials against the Users database. Upon successful verification, users are directed to the home page.
2. **Forgot Password:** Users who have forgotten their password can initiate a password recovery process. They receive a One-Time Password (OTP) on their registered email address. After OTP verification, users can reset their password securely.
3. **User Registration (Sign Up):** New users can sign up for an account. Upon registration, they receive an OTP on their email for verification. After OTP verification, users are directed to a details page where they enter basic information such as date of birth, phone number, profile picture etc. The entered details are saved in the Users database.
4. **Profile Update:** Users have the option to update their profile information. This feature allows users to edit their basic information and upload a new profile picture if desired. Any changes made are updated in the Users database.
5. **Admin Login:** Module admins can log in using the unique credentials reserved for them in the Users table. Passwords for admins are changed after elections for enhanced security, ensuring robust protection of sensitive module data.
6. **Database Integration:** User registration and profile information are stored and managed in the Users database, ensuring data integrity and accessibility across the application.

5.1.3 Entity Relationship Diagram

| users | |
|---------------|------------------------|
| user_id | integer |
| name | character_varying(255) |
| email | character_varying(255) |
| dob | date |
| age | integer |
| age_group | character_varying(10) |
| profile_photo | varbinary |
| gender | character_varying(10) |
| password | character_varying(20) |
| phone_number | character_varying(10) |
| house_number | integer |
| ward_number | integer |
| occupation | character_varying(50) |
| guardian_uid | integer |

Figure 2: Users table

5.1.4 Data Flow Diagram

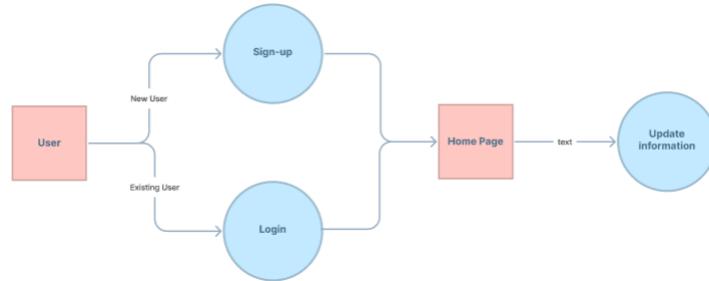


Figure 3: Level-1 DFD of Users Module

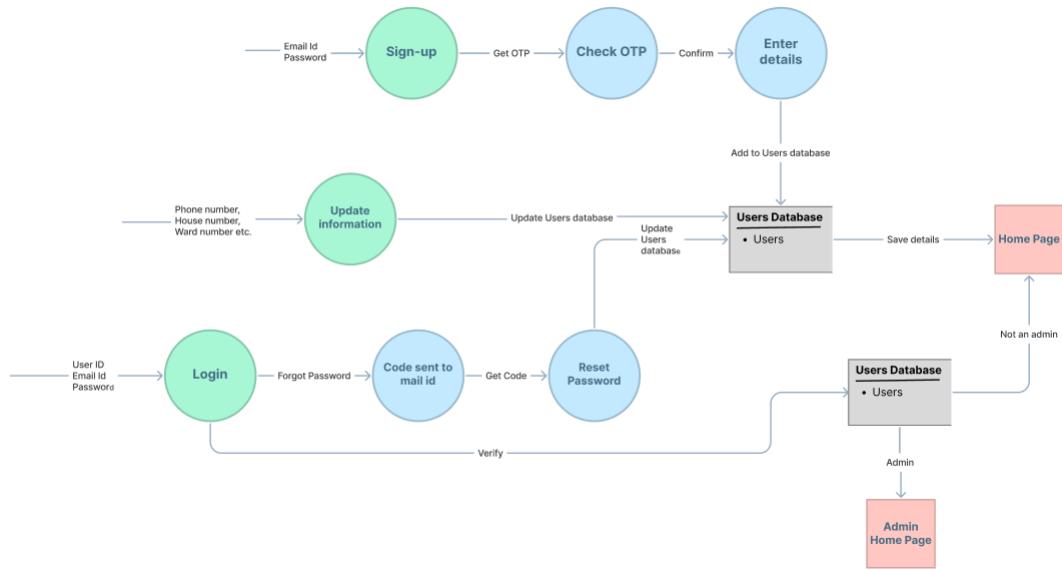


Figure 4: Level-2 DFD of Users Module

5.1.5 Interface



Figure 5: Landing page of the Smart City Management System

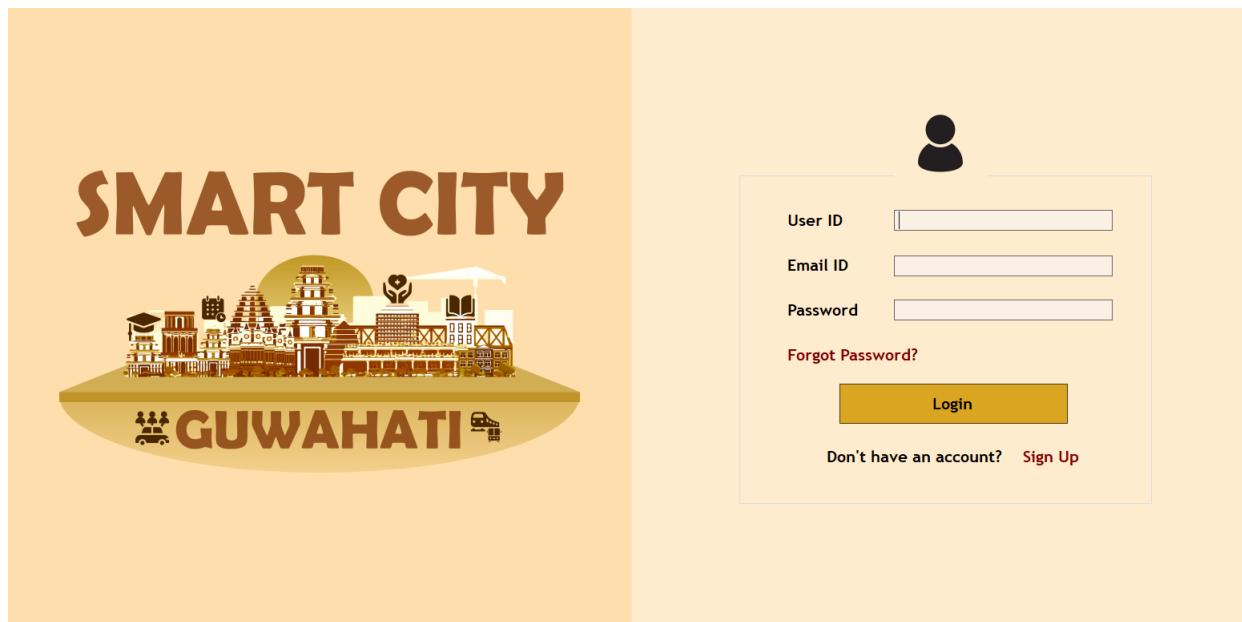


Figure 6: Login page for the Smart City Management System

Note Similar UI has been adopted for the user sign-up page and the OTP verification page.

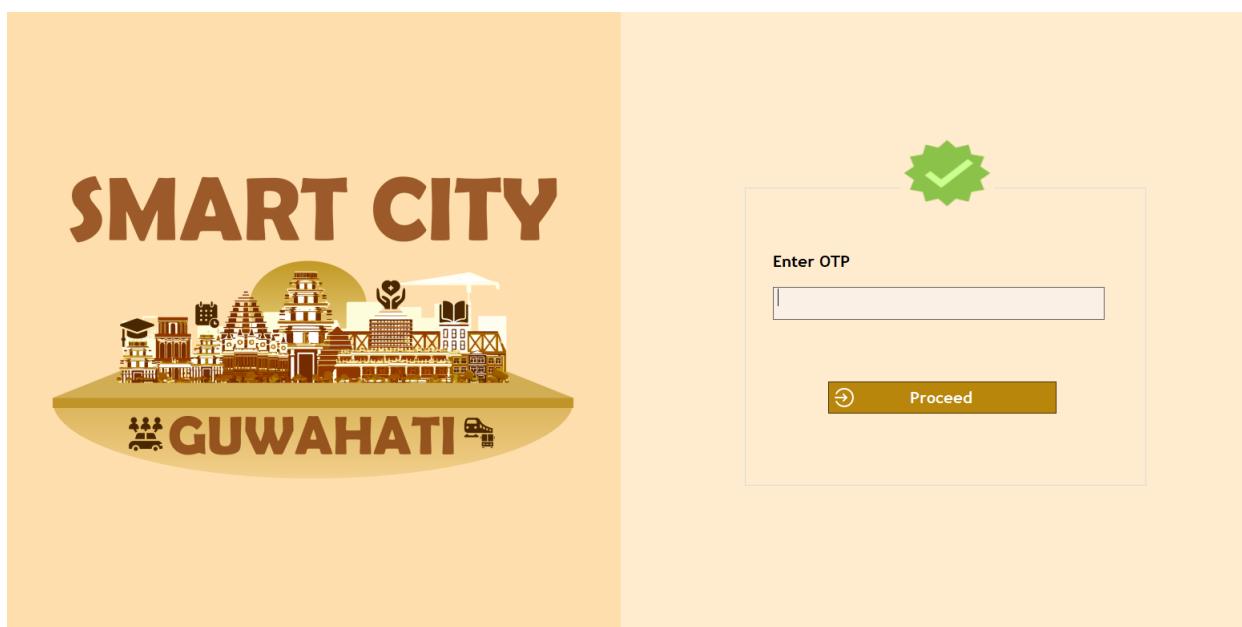


Figure 7: OTP verification page for the forgot password option



Enter Details

| | | | |
|---------------|----------------------|----------------|---|
| Full Name | <input type="text"/> | Profile Photo |  |
| Date of Birth | 15 March 2024 | Upload Picture | <input type="button" value="Upload Picture"/> |
| Age | <input type="text"/> | Age Group | <input type="text"/> |
| Gender | <input type="text"/> | Guardian UID | <input type="text"/> |
| Phone Number | <input type="text"/> | | |
| House Number | <input type="text"/> | | |
| Ward Number | <input type="text"/> | | |
| Occupation | <input type="text"/> | | |

Figure 8: Details page where users can enter basic information



{Name} {UID} [Logout](#)

| | | | | |
|---|---|---|--|--|
|  Home |  Professional Services |  Employment Portal |  Education Department |  Healthcare Department |
|  Important Notices |  Transportation Department |  Festival and Event Planning |  Complaint and Redressal |  Administrative Hierarchy |
|  Update Profile |  Library |  Banking | | |

Figure 9: Home Page

The screenshot shows the 'Update Profile' page of the SmartGhy system. The top navigation bar includes the 'SMART CITY' logo and 'GUWAHATI' text, along with a 'Logout' button. On the left, there's a sidebar with icons for 'Home', 'Important Notices', and 'Update Profile'. The main content area has several input fields: 'Full Name' (placeholder: {Name}), 'Date of Birth' (placeholder: 15 March 2024), 'Age' (placeholder: {Age}), 'Gender' (placeholder: {Gender}), 'Phone Number' (placeholder: {Phone number}), 'House Number' (placeholder: {House Number}), 'Ward Number' (placeholder: {Ward Number}), 'Occupation' (placeholder: {Occupation}), 'Profile Photo' (with a large black placeholder icon and a 'Re-Upload Picture' link), 'Age Group' (dropdown menu), and 'Guardian UID' (placeholder: {Guardian UID}). A 'Save' button is at the bottom.

Figure 10: Update profile page allows users to edit their basic information

Note In some places, a placeholder values have been used, which would be duly changed during development according to the entries from database.

5.2 Professional Services Directory

5.2.1 Introduction

The basis of any smart city revolves around facilitating seamless service provision and access to everyone. To tackle this, we propose the creation of a service directory platform that serves the needs of both service users and providers. The main objective is to optimize the service request and provision processes, thereby improving convenience and efficiency for all stakeholders involved.

5.2.2 Features

- Search Service:** For Users, the platform provides comprehensive search functionalities, allowing them to efficiently locate pertinent services. These search functions incorporate advanced filtering mechanisms based on various criteria, remuneration, industry domains, and job categories, in addition to keyword-based search capabilities.
- Service Request Functionality:** The process of requesting assistance from our organization is streamlined to enhance user satisfaction. This entails the creation of user-friendly application forms, allowing individuals to effortlessly complete and submit their requests, and providing timely feedback on service scheduling. Additionally, users are granted the flexibility to cancel scheduled services as needed. To preserve the integrity of our platform, charges are applied to both service requests and cancellations, acting as a safeguard against potential flooding attacks.
- Comprehensive Service Provider Tools:** In the context of service procurement, we offer a comprehensive set of instruments aimed at optimizing the servicing process. This

encompasses functions such as service posting, service withdrawal, schedule viewing, in-voucher examination, and real-time status monitoring of job postings.

4. **History:** Our platform has incorporated a "View History" feature, accessible to both users and service providers. This feature allows users to view a comprehensive record of the services they have received in the past. On the other hand, service providers can utilize this feature to view and verify all the services they have rendered, cross-referencing them with their respective invoices.
5. **Modular Development:** The portal is constructed utilizing a modular methodology, facilitating extensibility and adaptability in the incorporation of novel functionalities or interfacing with external systems in the future. This methodology ensures the portal's capacity to undergo modifications in response to evolving requirements and advancements in technology.

5.2.3 Entity Relationship Diagram

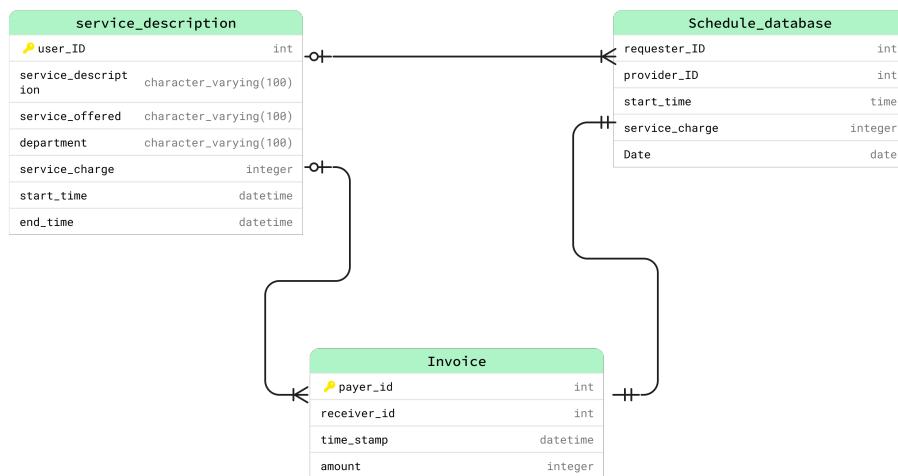


Figure 11: ER-Diagram for Service Directory module

Note In this ER-Diagram all the columns of **Schedule Database** form a composite primary key.

5.2.4 Data Flow Diagram

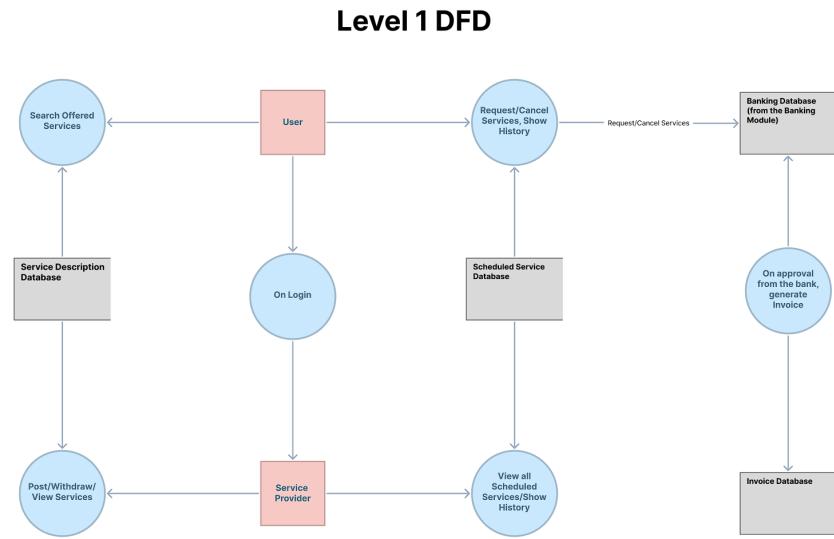


Figure 12: Level-1 DFD of Service Directory Module

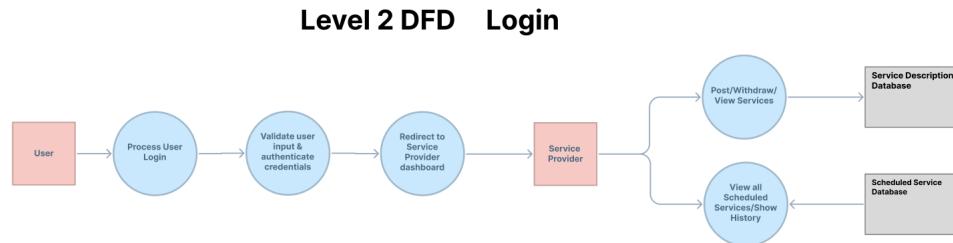


Figure 13: Level-2 DFD of Service Directory Module

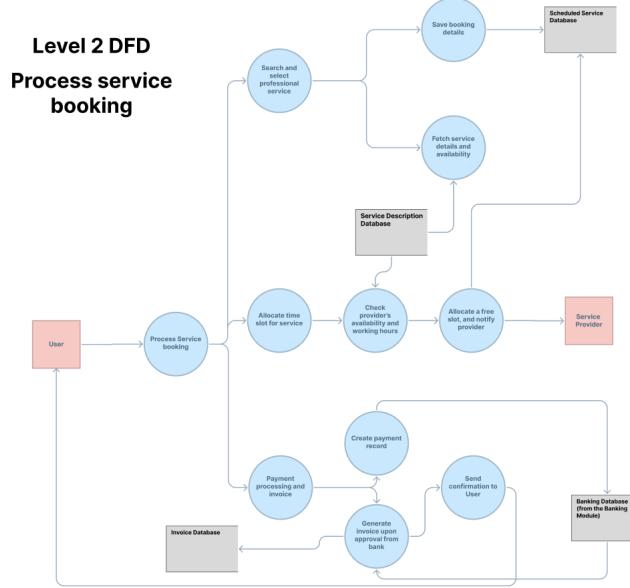


Figure 14: Level-2 DFD of Service Directory Module

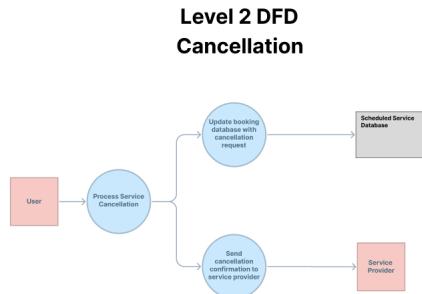


Figure 15: Level-2 DFD of Service Directory Module

5.2.5 Interface

The following is the preliminary designs for the interface for *Service Directory Module*. The team has followed template based approach in which NavajoWhite colour palette is used.

Note All the further designs will be based on this template and all the features to be added have been shown.



Figure 16: Landing page where listing of all available services for user will be shown.

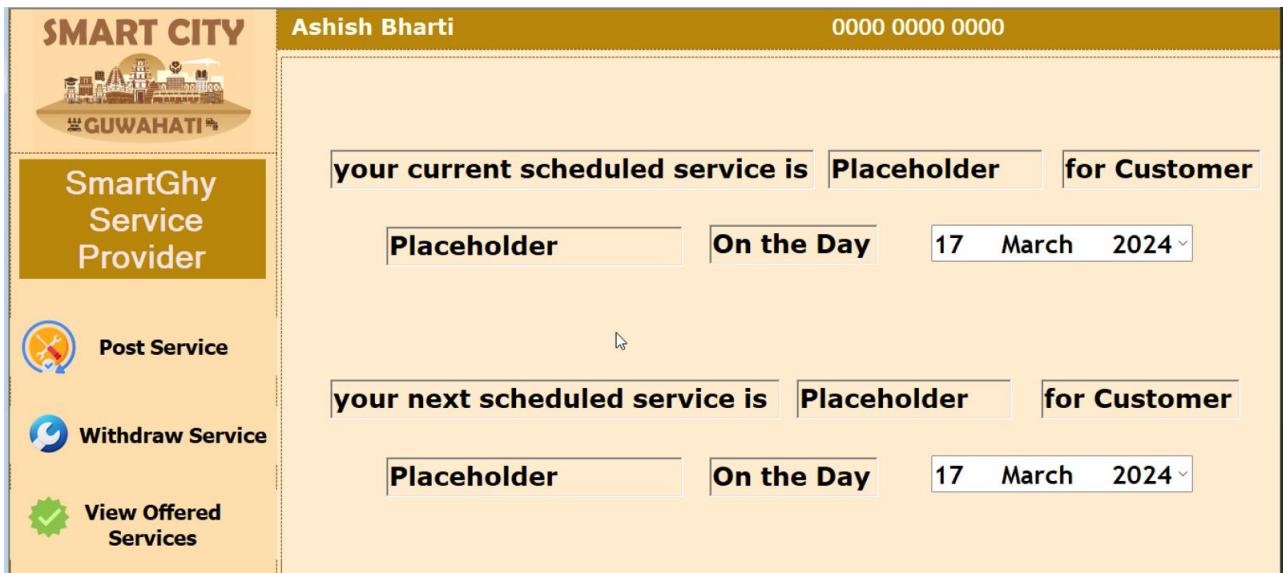


Figure 17: Landing page for the Service Provider.

5.3 Employment Portal

5.3.1 Introduction

In the context of developing a smart city, one crucial aspect is ensuring efficient employment opportunities and recruitment processes. To address this, we propose the development of an employment portal that caters to both job seekers and recruiters. The primary aim is to

streamline the job search and recruitment processes, enhancing convenience and effectiveness for both parties involved.

5.3.2 Features

1. **Efficient Job Search Functionality:** For job seekers, we develop robust search functionality that enables them to find relevant job listings quickly and easily. This includes advanced search filters based on criteria such as location, salary, industry, and job type, as well as keyword search capabilities.
2. **Streamlined Application Process:** We aim to streamline the job application process to minimize friction for job seekers. We design and develop user-centric application forms with autofill capabilities to facilitate seamless completion and provide real-time updates on their status. Furthermore, job seekers can withdraw their applications and conveniently access a comprehensive record of all submitted applications.
3. **Comprehensive Recruitment Tools:** To meet the needs of recruiters, we offer a comprehensive suite of instruments crafted to streamline the recruitment journey. This offering encompasses capabilities such as customizable job posting templates, applicant tracking systems, and interview scheduling functionalities. Additionally, recruiters are endowed with tools that facilitate withdrawing job postings, editing existing postings, and employing criteria-driven filters to assess job applications. Furthermore, our platform provides seamless channels for updating and communicating application statuses with candidates.

5.3.3 Entity Relationship Diagram

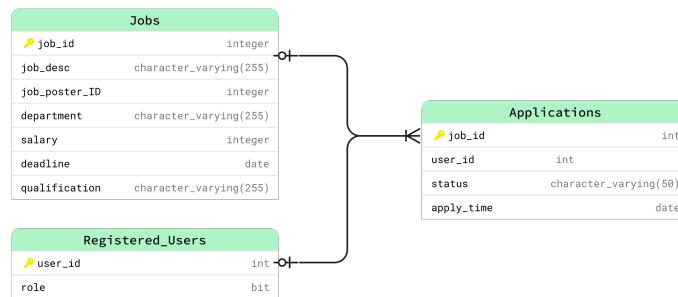


Figure 18: ER-Diagram for Employment module

Note In this ER-Diagram column job_id and user_id in **Applications** Database form a composite primary key.

5.3.4 Data Flow Diagram

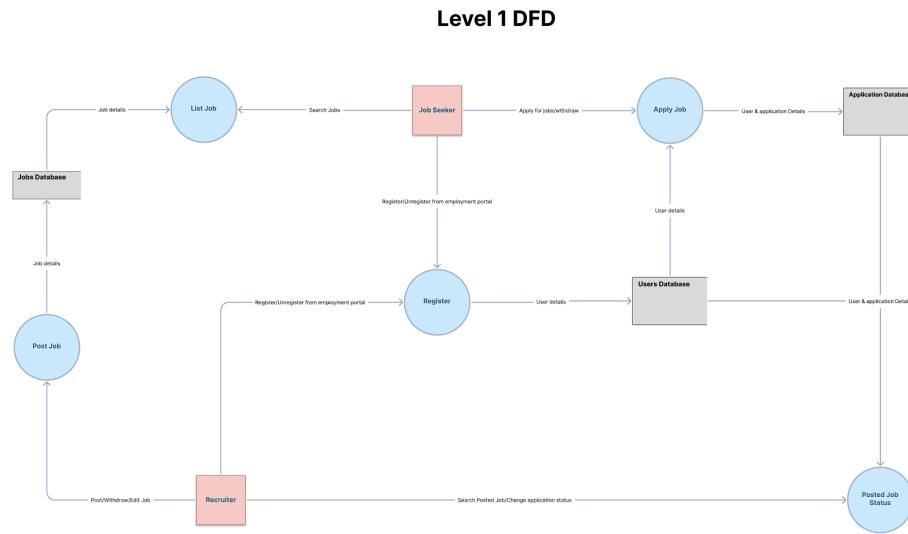


Figure 19: Level-1 DFD of Employment Module

Level 2 DFD Login

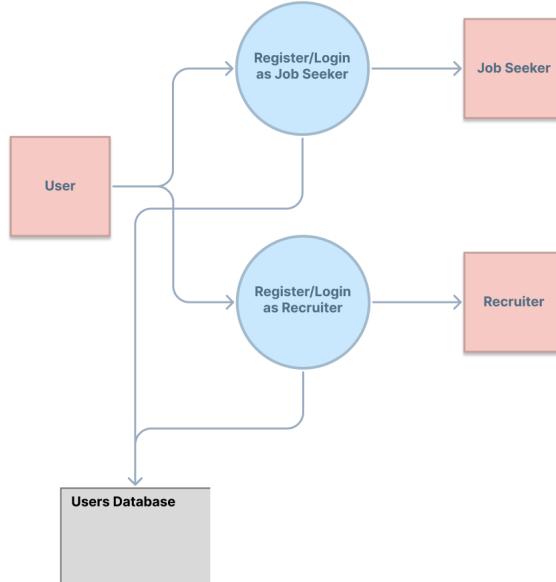


Figure 20: Level-2 DFD of Employment Module

Level 2 DFD Recruiter functions

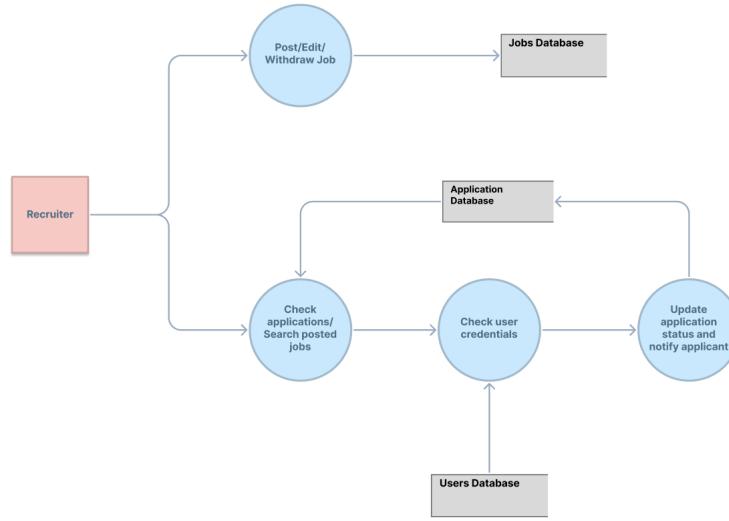


Figure 21: Level-2 DFD of Employment Module

Level 2 DFD Employee functions

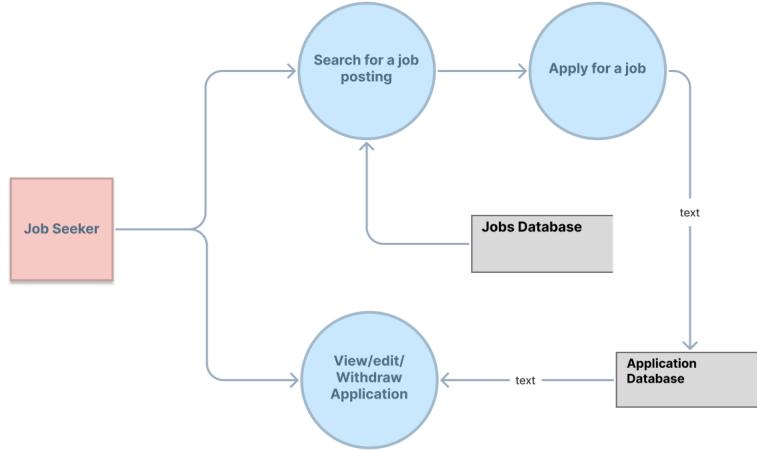


Figure 22: Level-2 DFD of Employment Module

5.3.5 Interface

The following is the preliminary designs for the interface for *Employment Module*. The team has followed template based approach in which NavajoWhite colour palette is used.

Note All the further designs will be based on this template and all the features to be added have been shown.

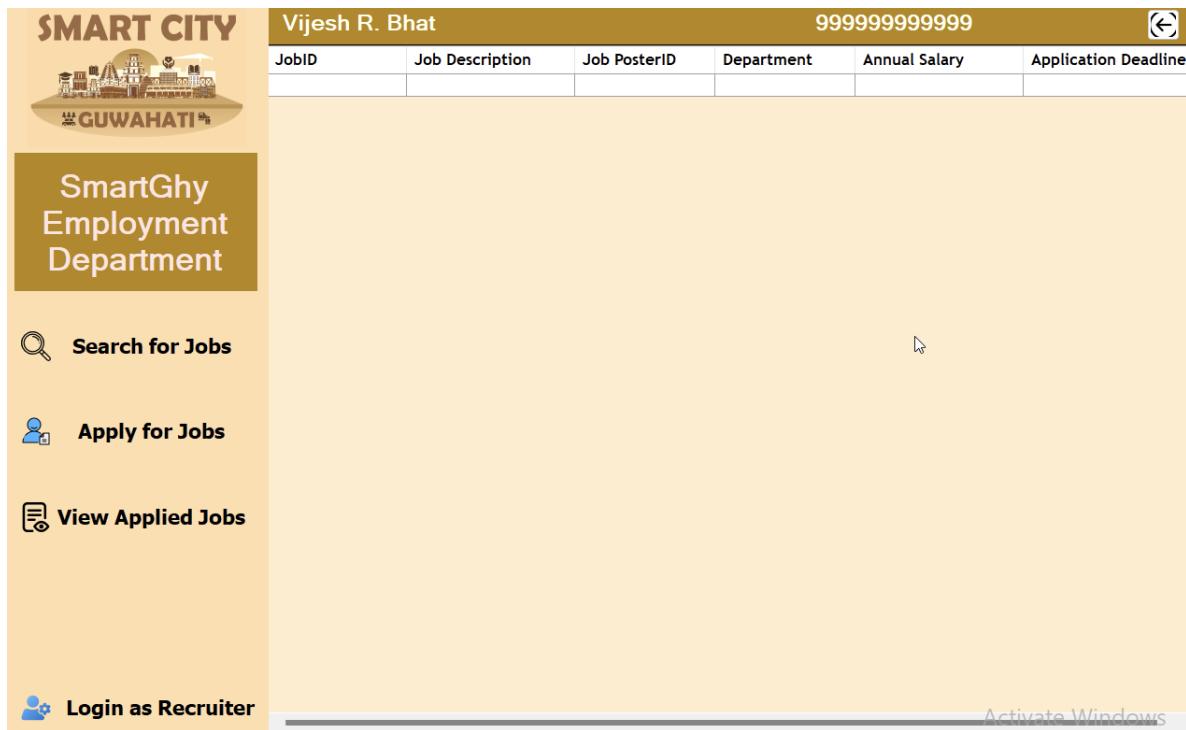


Figure 23: Landing page where listing of all available jobs will be shown.

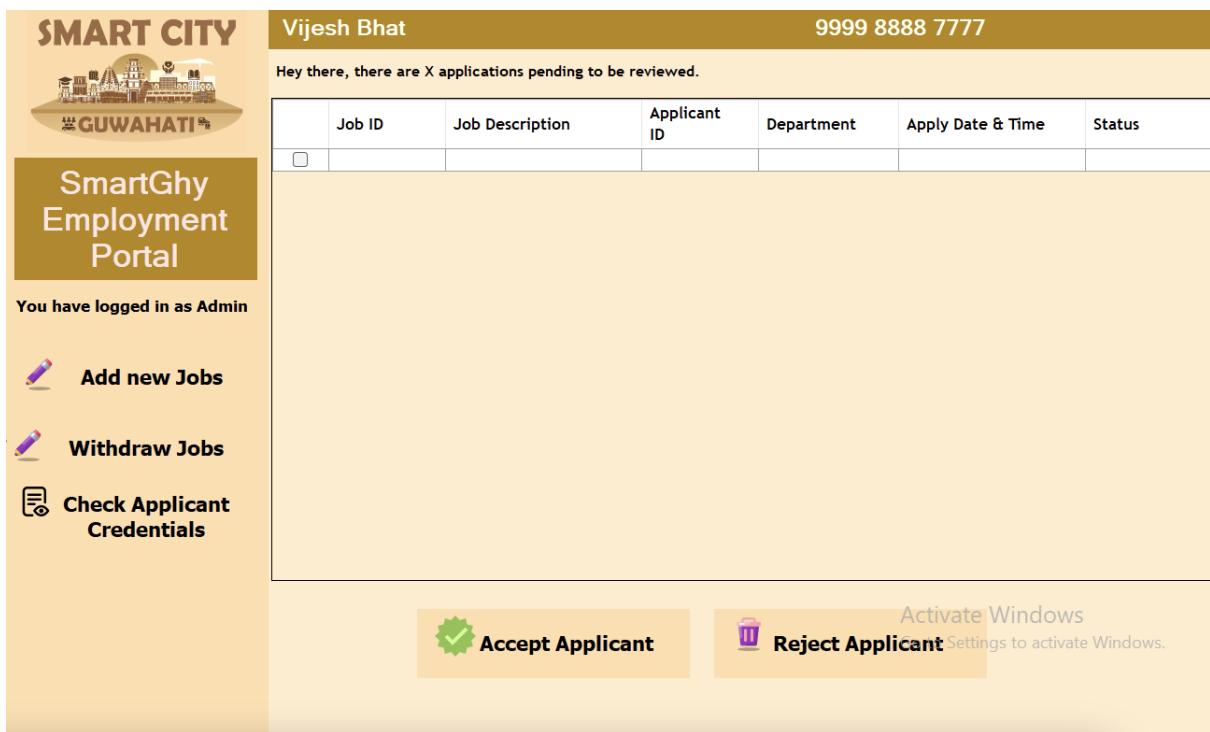


Figure 24: Landing page of an admin who can accept/reject job applications.

5.4 Education Management

5.4.1 Introduction

The Education management comes under the Ministry of Education in the smart city, which is elected through the *Administrative Hierarchy module*. This section of the Smart City Management provides various services aimed at enhancing the educational experience for all stakeholders. From student enrollment and course management to teacher facilitation and administrative oversight, our platform offers a comprehensive suite of tools and resources to ensure seamless operations and promote academic excellence.

5.4.2 Features

Main Users:

1. **Students:** Any resident, irrespective of profession, enrolled in educational programs.
2. **Teacher (Ministry):** Professionals employed by the Ministry of Education.
 - a. Institute Teachers: Educators affiliated with specific institutes.
 - b. E-Course Teachers: Instructors offering courses through the e-learning platform.
3. **Admins:**
 - a. Institute Admins: Administrators responsible for managing operations within their respective institutes.
 - b. E-Course Platform Admins: Administrators overseeing the e-learning platform's functionality and content.
4. **Education Minister:** Elected official holding a fixed account for administrative purposes within the education system.

User wise features:

- **Students:**

- **E-Course:**

1. Join courses by paying a designated course fee.
2. Engage in quizzes and utilize video lessons to fulfill course requirements.
3. Obtain certificates upon successful completion.
4. Provide feedback by rating the courses they've undertaken.

- **Institute:**

1. Apply for admission to educational institutions.
2. Pay tuition fees.
3. Check and avail bus facilities provided by the institution.
4. Receive pass certificates and extracurricular certificates upon completion of programs.

- **Moodle:**

1. Participate in online institute classes, if available.
2. Submit assignments and access learning resources.
3. View grades and assessments.

– **Entrance Exams:**

1. Enroll for city-wide entrance exams.
2. Receive admit cards containing exam date and venue information.
3. Obtain result certificates when results declared by the Education Minister.

• **Teachers:**

– **E-Course:**

1. Apply to add e-courses either on their personal account or on behalf of their institute.
2. Develop quizzes, lessons, and manage course content.

– **Moodle:**

1. Create institute classes within the Moodle platform.
2. Review student submissions and provide grades accordingly.

• **Admins:**

– **E-Course Admin:**

1. Review and approve/reject applied e-courses.
2. Manage existing courses, including dropping them if necessary.

– **Institute Admin:**

1. Approve/reject student admission applications.
2. Recruit employees for the institute, including teachers, bus drivers, security guards, etc.
3. Manage current employees, including salary payments.
4. Oversee student management and fee administration.
5. Generate pass certificates by inputting marks and other necessary information.

• **Minister:**

– **E-Course:**

1. Administer the same features as e-course admin, including approving/rejecting e-courses and managing existing courses.
2. Add e-course admins to assist in managing e-learning courses.

– **Institute:**

1. Add new institutions to the system.
2. Edit institution details, such as contact information and addresses.
3. Hire administrators, principals, and other staff members.

– **Entrance Exams:**

1. Set exam dates and venues.
2. Define exam syllabus and topics.
3. Release results and ranks to participants.

5.4.3 Entity Relationship Diagram



Figure 25: ER-Diagram for Education module

Note In this ER-Diagram the foreign key relationship of UID with the main **UserDB** isn't shown to avoid clumsiness.

5.4.4 Data Flow Diagram

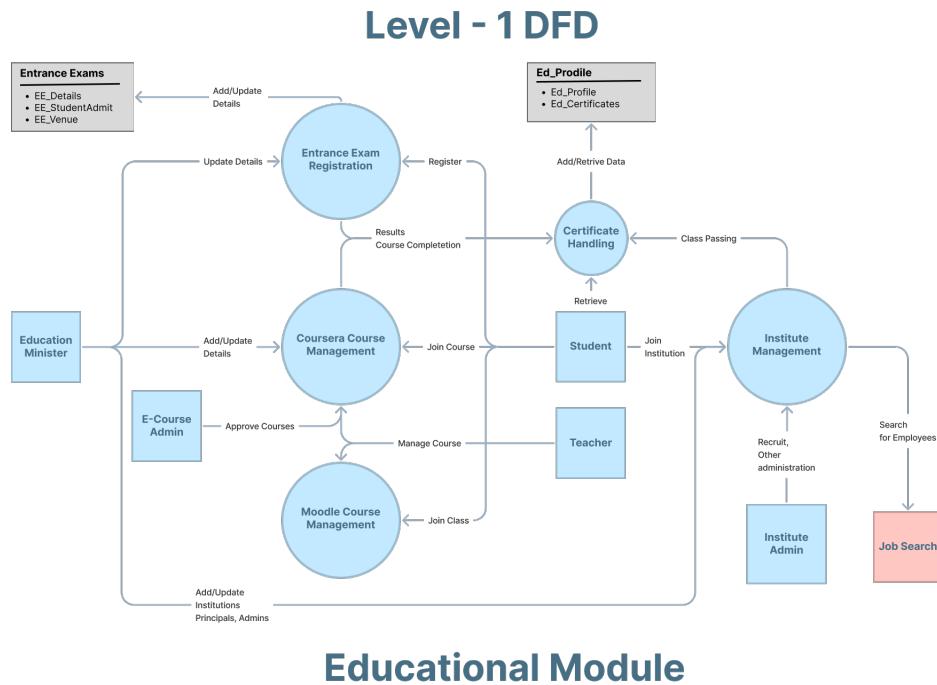


Figure 26: Level-1 DFD of Education Module

This depicts the four main processes within the Educational Module - Entrance Exam Registration, E-Course Management , Institute Management ,Moodle Course Management and how the main entities - Minister,Student,Teacher,Admin interact with these processes and the overall data flow across the system.

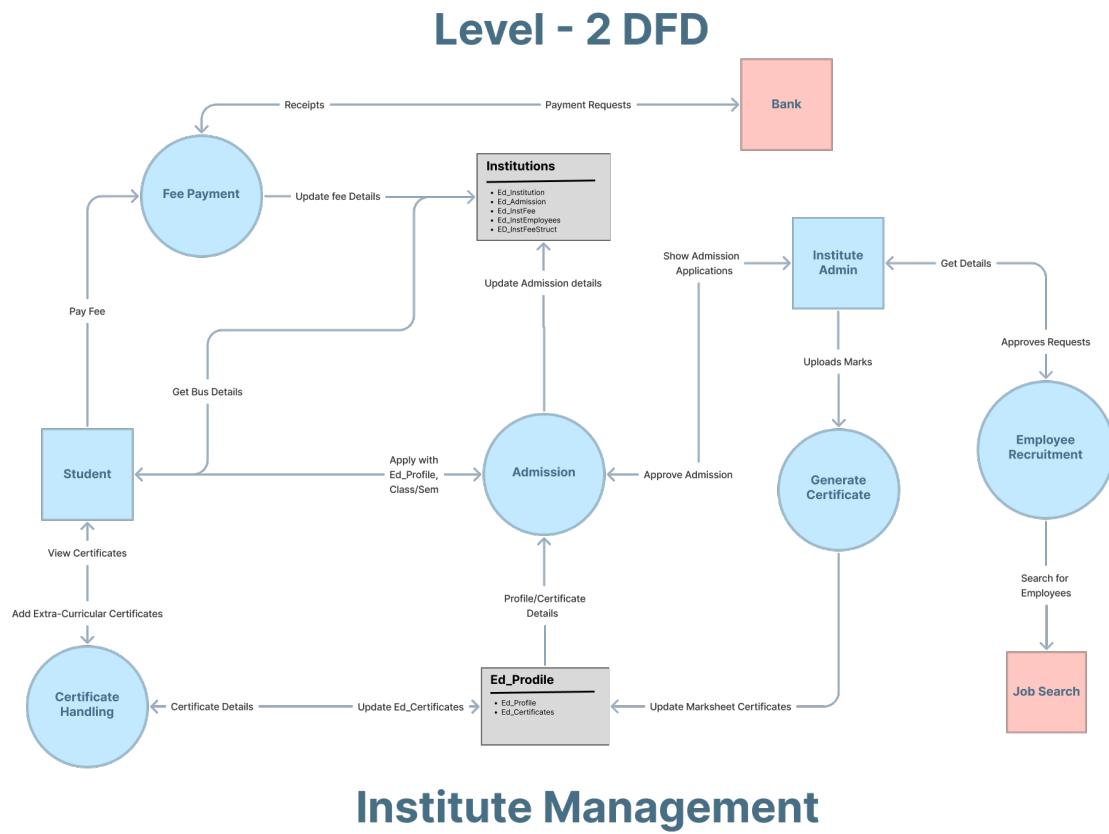
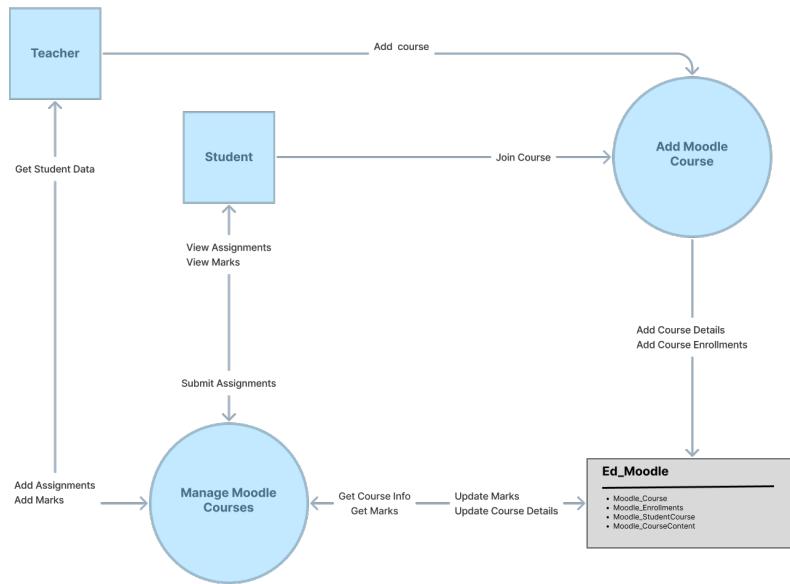


Figure 27: Level-2 DFD - Institute Management

It describes how -

1. Students can apply for admission to an academic institution and Admins can approve/reject their application.
2. Certificates are generated, stored, and viewed for each student.
3. Student's academic fee payment is handled.
4. Students can check and avail bus facilities provided by the institution.
5. Institute admins recruit employees.

Level - 2 DFD



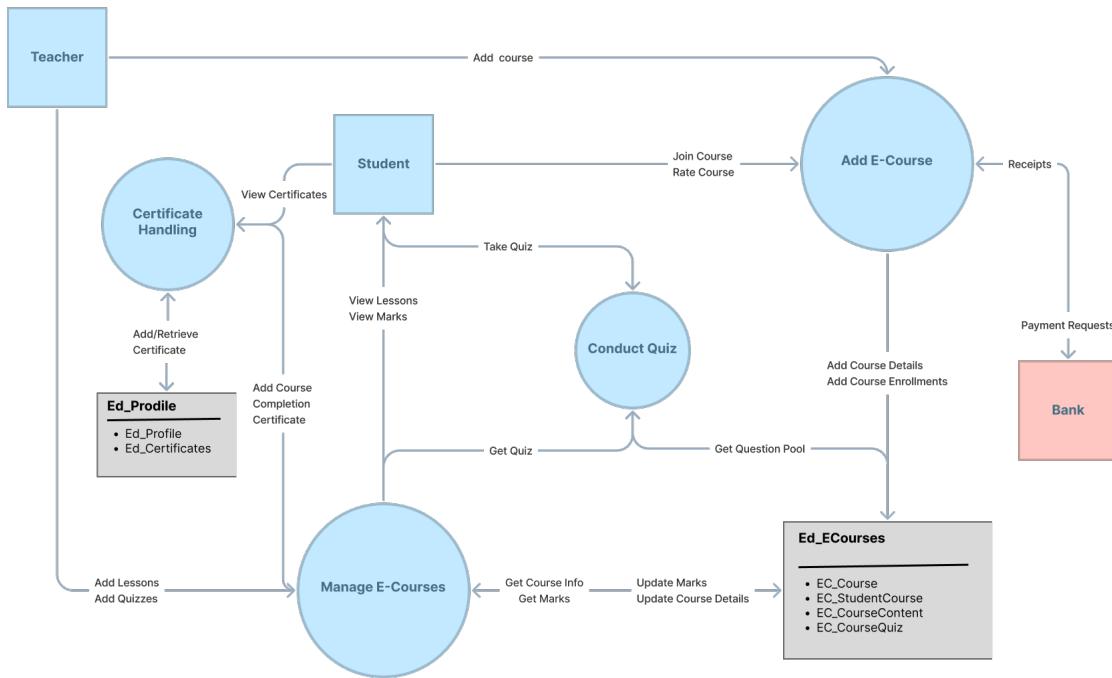
Moodle Course Management

Figure 28: Level-2 DFD - Moodle Course Management

It describes how -

1. Students can join institute classes within the Moodle platform.
2. Students can view their assessments and marks.
3. Teachers can create institute classes within the Moodle platform.
4. Teachers can review student submissions and provide grades accordingly.

Level - 2 DFD



E-Course Management

Figure 29: Level-2 DFD - E-Course Management

It describes how -

1. Students join courses by paying a designated course fee.
2. Students can engage in quizzes and utilize video lessons to fulfill course requirements.
3. Students obtain certificates upon successful completion.
4. Students can provide feedback by rating the courses they've undertaken.
5. Teachers can add quizzes, lessons, and manage course content.

5.4.5 Interface

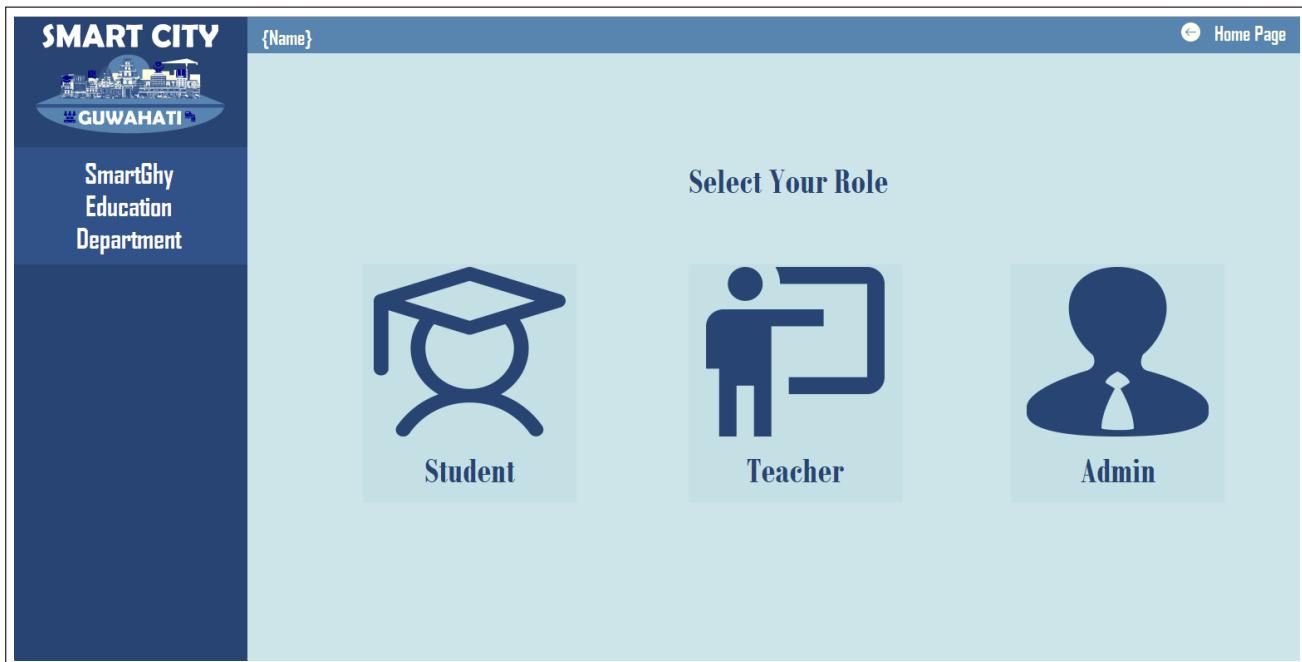


Figure 30: Landing Page - Select Role



Figure 31: Student Dashboard

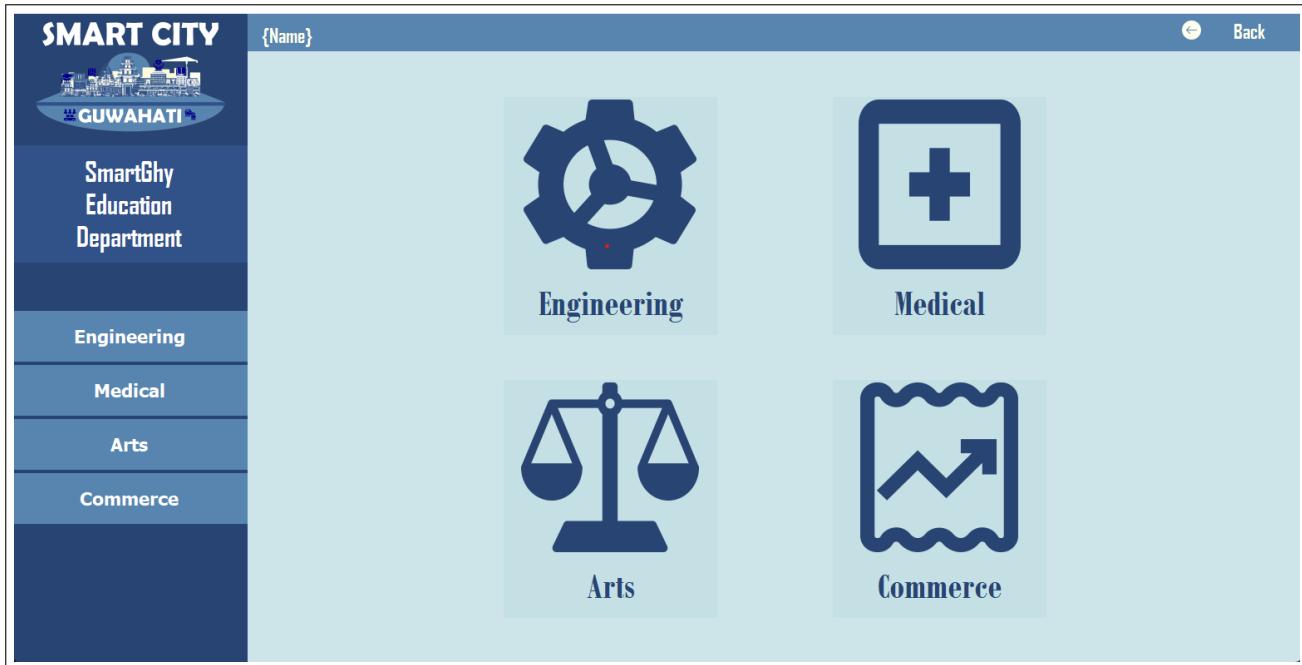


Figure 32: Student Entrance Exam

| Category | Course Details | Action |
|-------------|--|----------------------|
| Physics | { Course Name } { Course Instructor } { Course Institution } | { Fees } View |
| Chemistry | { Course Name } { Course Instructor } { Course Institution } | { Fees } View |
| Mathematics | { Course Name } { Course Instructor } { Course Institution } | { Fees } View |
| Biology | { Course Name } { Course Instructor } { Course Institution } | { Fees } View |
| Computer | { Course Name } { Course Instructor } { Course Institution } | { Fees } View |
| Management | { Course Name } { Course Instructor } { Course Institution } | { Fees } View |
| Humanities | { Course Name } { Course Instructor } { Course Institution } | { Fees } View |
| Sports | { Course Name } { Course Instructor } { Course Institution } | { Fees } View |
| Arts | { Course Name } { Course Instructor } { Course Institution } | { Fees } View |
| Finance | { Course Name } { Course Instructor } { Course Institution } | { Fees } View |
| Others | { Course Name } { Course Instructor } { Course Institution } | { Fees } View |

Figure 33: Student E-Course - Home

The screenshot shows a web interface for managing student institutions. On the left is a vertical sidebar with a logo for 'SMART CITY GUWAHATI' and a 'SmartGhy Education Department'. The main content area has a header with a search bar and a 'Home Page' link. Below the header, there is a table with four rows, each representing an institution. Each row contains a small school building icon and a set of placeholder text fields: '{ Institute Name }', '{ Institute Address }', '{ Contact no. }', and '{ Email Address }'.

Figure 34: Student Institute List

The screenshot shows a web interface for joining a classroom. On the left is a vertical sidebar with a logo for 'SMART CITY GUWAHATI' and a 'SmartGhy Education Department'. The main content area has a header with a search bar and a 'Back' link. A central modal window is titled 'JOIN CLASSROOM' and contains two input fields: 'Room ID' and 'Pass Key'. Below the fields is a 'Join' button.

Figure 35: Student Moodle - Join Course

The screenshot shows the 'Manage E-Course' section of the SmartGhy system. On the left sidebar, under the 'SmartGhy Education Department' heading, there are links for 'Add E-Course', 'Manage E-Course', 'Add Moodle Course', and 'Manage Moodle Course'. The main content area has a header with a search bar and a back button. Below the header, there are four course entries, each represented by a card with a graduation cap icon and a computer monitor icon. Each card contains placeholder text for 'Course Name' and 'Course Institution', and a rating of '4.8 ★'. To the right of each card is a blue 'Edit' button.

Figure 36: Teacher - Manage E-Course

The screenshot shows the 'Approve Student Admissions' section of the SmartGhy system. On the left sidebar, under the 'SmartGhy Education Department' heading, there are links for 'Admission', 'Fees', 'Recruit Employees', 'Manage Employees', and 'Manage Students'. The main content area has a header with a search bar and a back button. Below the header, there are four student profiles, each represented by a card with a person icon. Each card contains placeholder text for 'Student Name', 'Recent Activity', 'DOB', 'Contact no.', and 'Email Address'. To the right of each card are two buttons: a green 'Approve' button and a red 'Reject' button.

Figure 37: Institute Admin - Approve Student Admissions

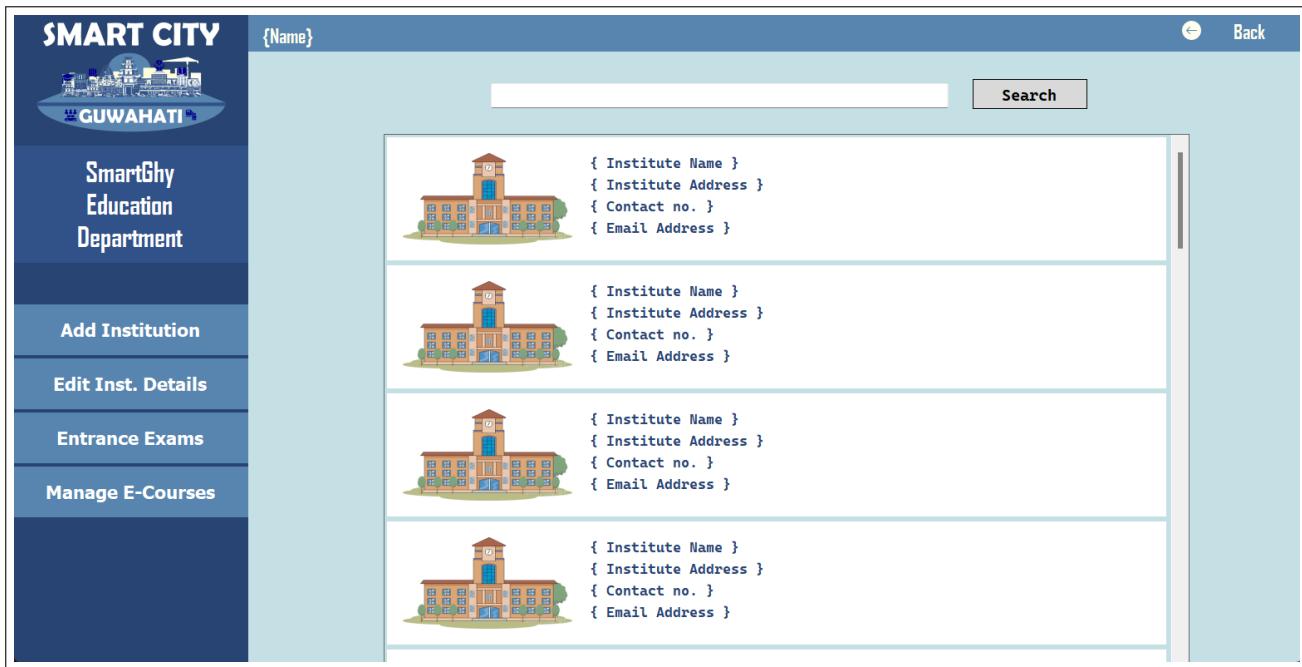


Figure 38: Education Minister - Institution List

5.5 Healthcare Management

5.5.1 Introduction

The Healthcare module of our project aims to streamline and enhance various aspects of Healthcare Management. It comprises several key features

- managing patient admissions and scheduling appointments
- tracking the availability of doctors, nurses, and other staff members.
- requesting ambulance services quickly
- managing the inventory of medicines ,medical supplies and dispensing medications in pharmacies .
- maintain comprehensive records of patient medical histories such as past illnesses and medications.
- managing blood donation activities
- maintaining records of births and deaths within a healthcare facility

5.5.2 Features

1. Hospital Management:

- This feature focuses on managing hospital operations efficiently. It includes functionalities such as managing bed allocation and scheduling appointments. It also helps in tracking the availability of doctors, nurses, and other staff members.

2. Pharmacy Management:

- The pharmacy management feature helps in managing the inventory of medicines and medical supplies. It includes functionalities of generating purchase orders, and dispensing medications.

3. Health Record Tracking:

- This feature allows the maintenance of comprehensive records of patient medical histories. It includes details such as past illnesses, medications and blood donation. This information is crucial for providing effective treatment and ensuring patient safety.

4. Ambulance Booking:

- The ambulance booking feature enables users to request ambulance services quickly and efficiently. It includes functionalities for booking ambulances and ensuring prompt response to emergency situations.

5. Blood Donation Management:

- This feature helps in managing blood donation activities, including donor registration and blood collection. It also includes functionalities for organising blood donation drives and maintaining donor records.

6. Birth and Death Record Management:

- This feature focuses on maintaining records of births and deaths within a healthcare facility. It includes functionalities for registering births and deaths, generating birth and death certificates, and maintaining accurate records for legal and administrative purposes.

5.5.3 Entity Relationship Diagram

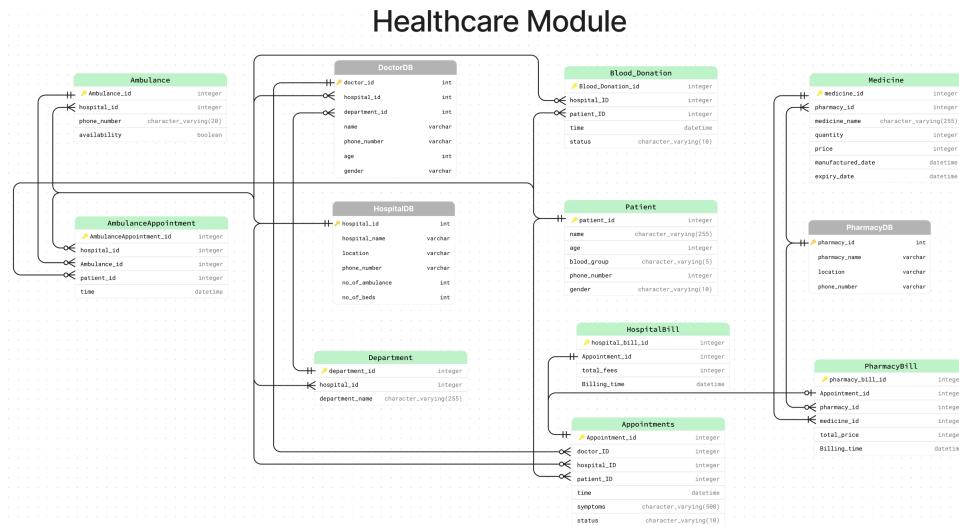


Figure 39: ER-Diagram for Healthcare module

Entities being used in the database are:

- **Ambulance:** Represents a vehicle used for transporting patients.
- **AmbulanceAppointment:** Represents an appointment for an ambulance service.
- **Doctor:** Represents a medical professional who provides healthcare services.
- **Hospital:** Represents a healthcare facility where medical services are provided.
- **Department:** Represents a specific area or division within a hospital.
- **BloodDonation:** Represents the act of donating blood.
- **Patient:** Represents an individual receiving medical treatment or care.
- **HospitalBill:** Represents a bill for medical services provided by a hospital.
- **Appointments:** Represents scheduled appointments for medical services.
- **Medicines:** Represents medications used for treatment.
- **Pharmacy:** Represents a facility where medications are dispensed.
- **PharmacyBill:** Represents a bill for medications provided by a pharmacy.

5.5.4 Data Flow Diagram

This shows the overall dataflow between the module and different submodules involved

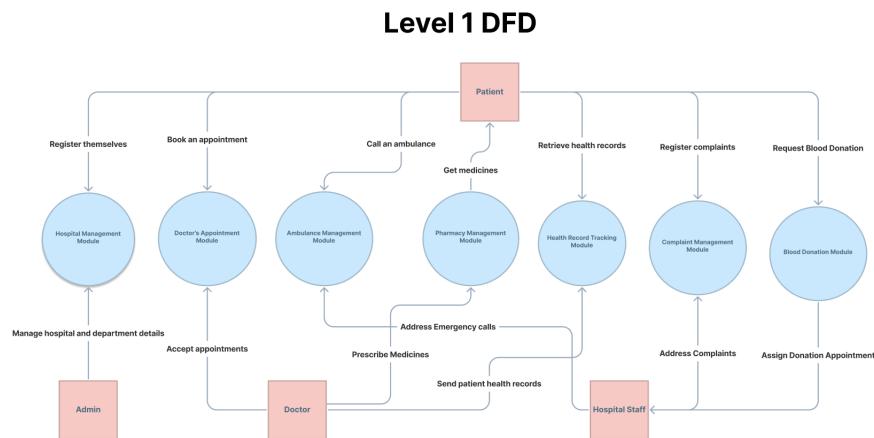


Figure 40: Level-1 DFD of Healthcare Module

Level 2 DFD

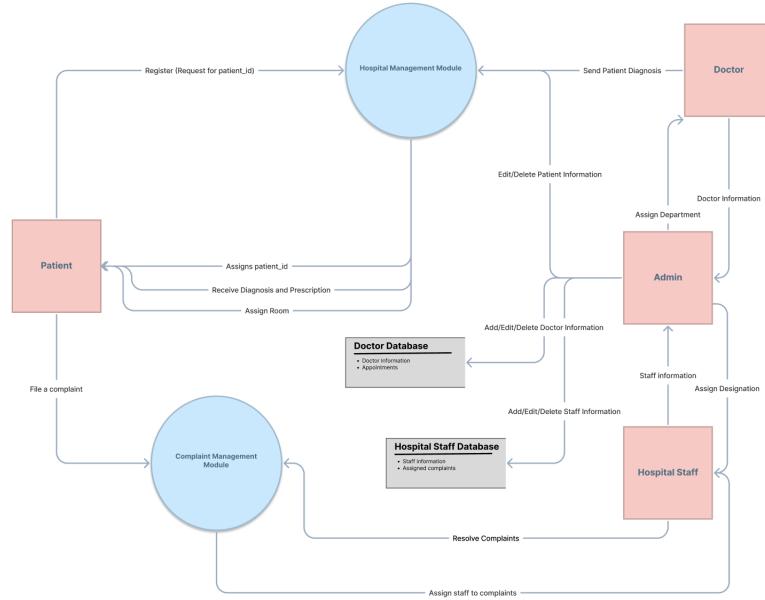


Figure 41: Level-2 DFD part1

Level 2 DFD

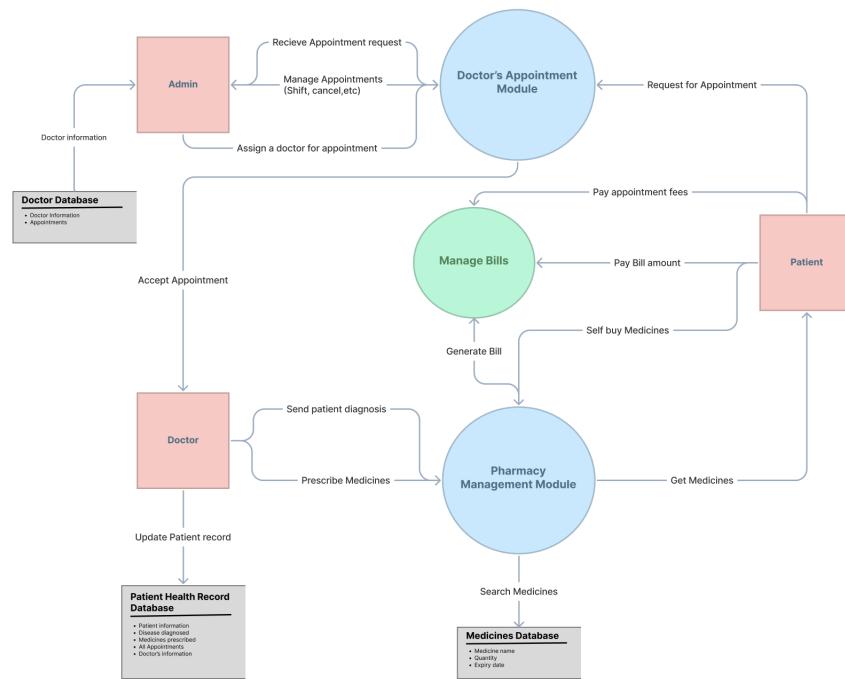


Figure 42: Level-2 DFD part2

Level 2 DFD

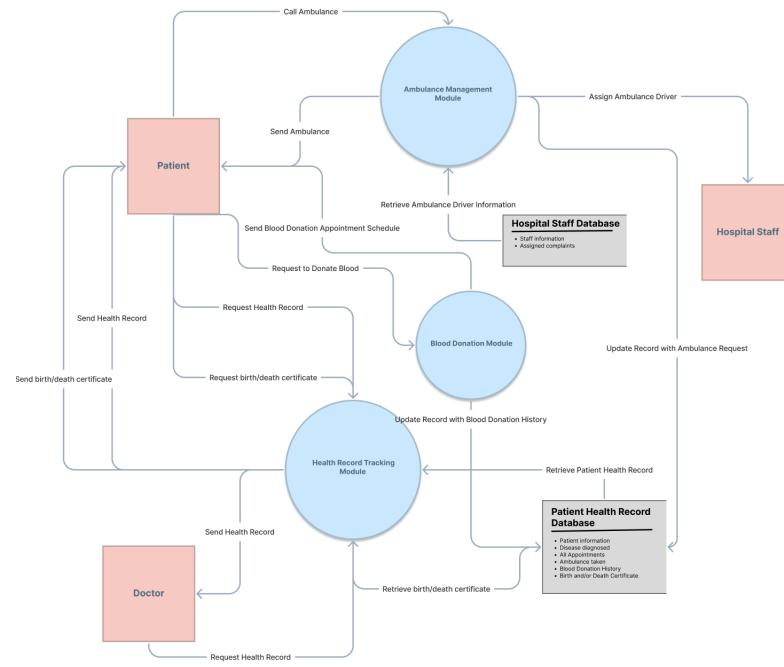


Figure 43: Level-2 DFD part3

5.5.5 User Interface of Module

The following are the preliminary designs for the interface

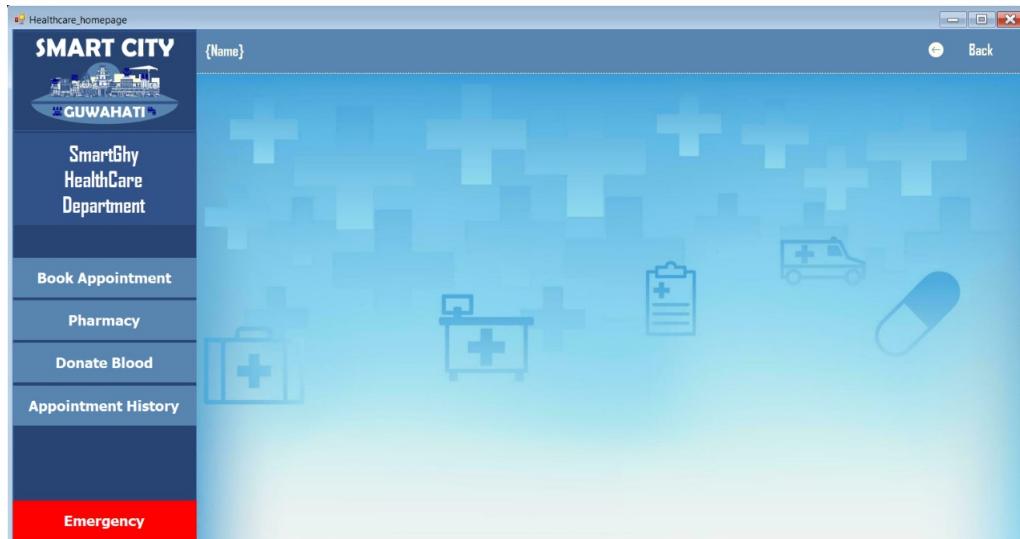


Figure 44: Home screen for user

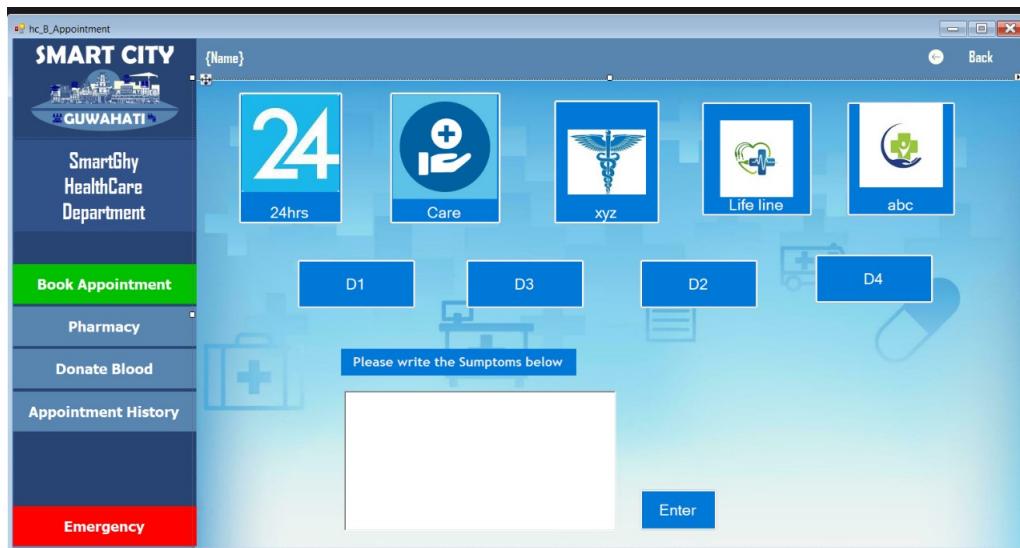


Figure 45: Appointment scheduling screen for users



Figure 46: Page for buying and searching medicine in pharmacies

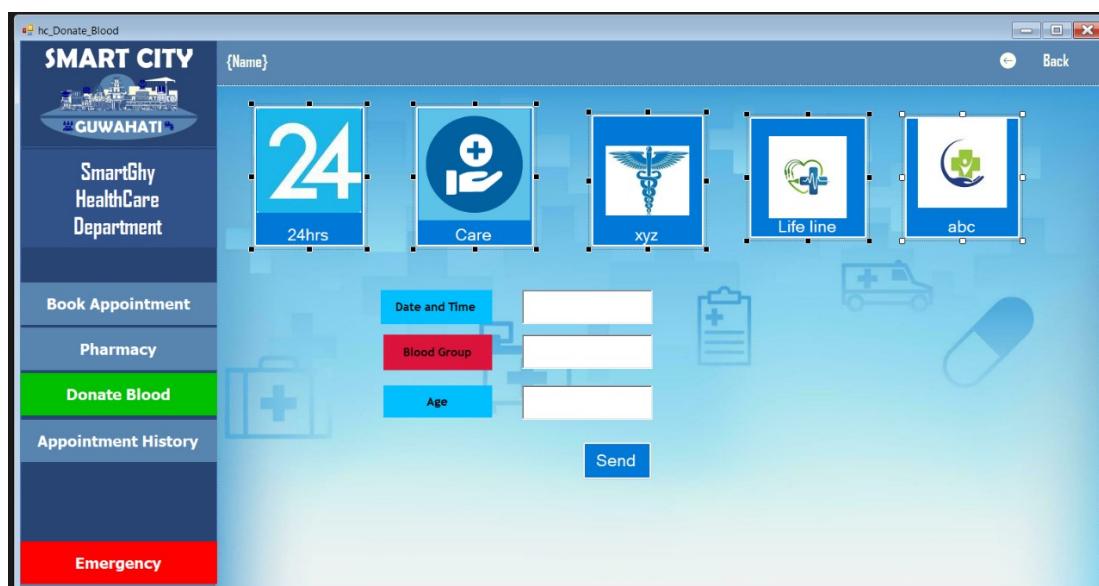


Figure 47: Page for users to donate blood

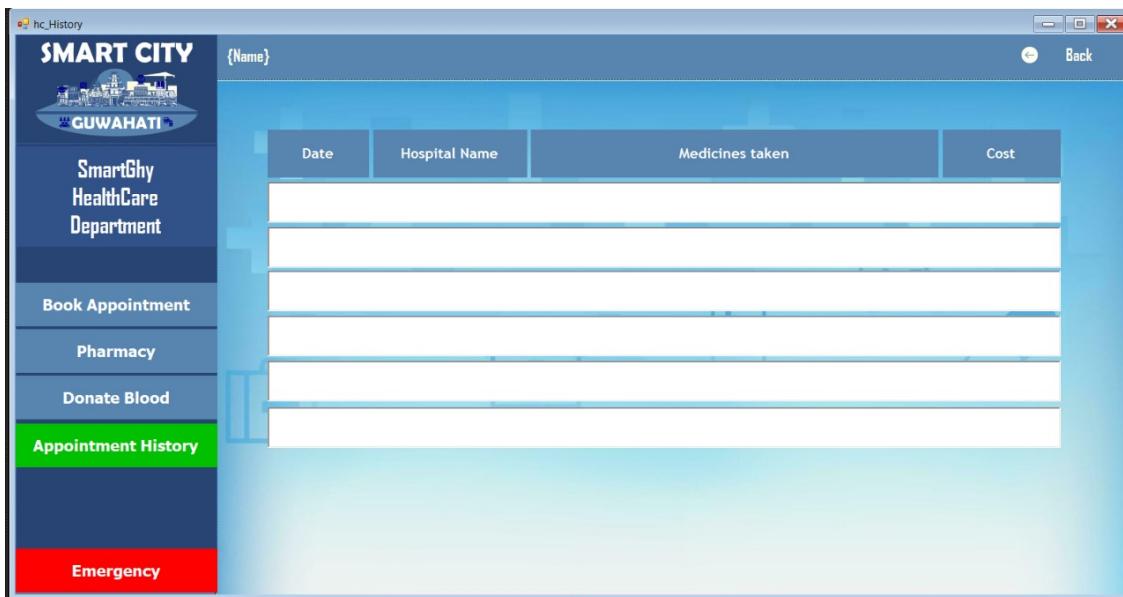


Figure 48: The page where the user can see his/her past medical records like appointments

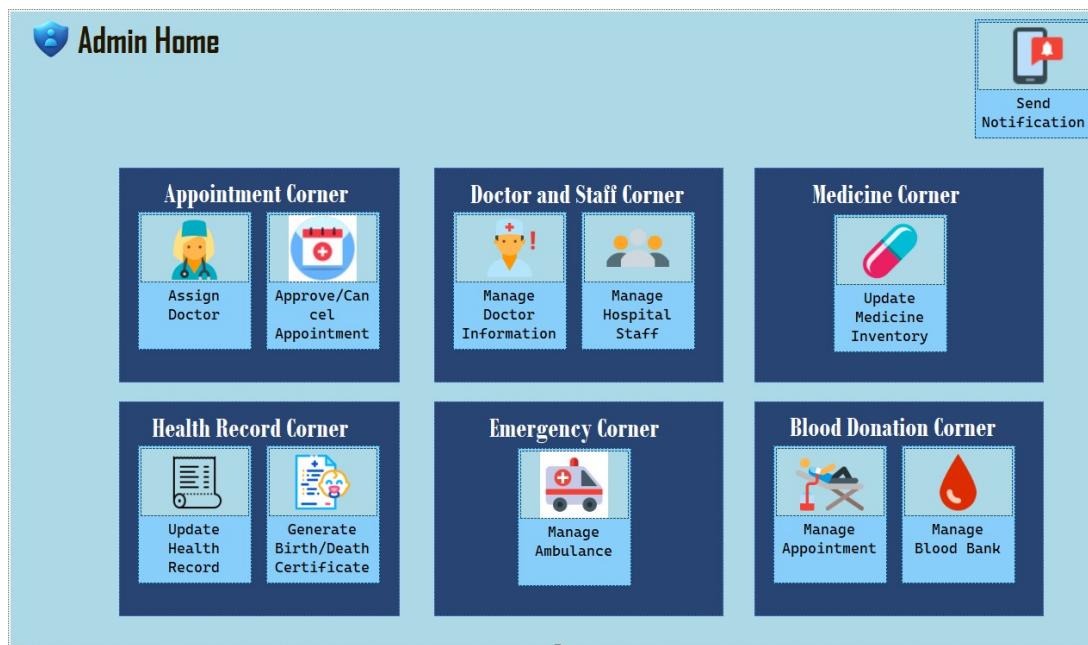


Figure 49: Home page for admin of healthcare department

5.6 Transportation Management

5.6.1 Introduction

The Transportation management comes under Ministry of Transportation in the smart city, who will be elected in the election conducted by *Administrative Hierarchy module*. This Section of the Smart City Management provides various services like Driving License Acquisition, Toll Booth Management, Ride Sharing, Public Transport Management, Vehicle Registration and also administrative features for the minister of transportation.

5.6.2 Features

1. Administrative Features:

- The Transportation minister is the admin and he/she after duly elected, gets access to the admin features.
- The minister can add the officers for different sub departments like **R.T.O** (for driving license and vehicle registration), **Public Transportation** (for bus schedules), **Roadways** (For tolls), **Car-pooling** (For ride sharing), who can be chosen from the pool of officers employed through *Employment Portal*.

2. Public Bus Schedules:

- This platform enables citizens to view bus schedules and buy tickets for the buses based on seats available.
- The citizens can filter bus schedules based on the starting and ending bus-stops and all the buses plying between these will be displayed along with the map showing the route between the two places.
- The Admin for this submodule can add, update, delete bus schedules and can update the details regarding bus stops. They can also appoint drivers to the buses who can be employed through *Employment Portal*.

3. Ride Sharing:

- This platform enables individuals with valid **Driving License** (acquired through the *Driving License Request* part of this module) to offer and find carpooling opportunities, aiming to mitigate pollution and enhance resource utilization.
- Users, both riders and vehicle owners, can communicate through the chat feature provided in this module.
- Drivers can accept riders after passengers have paid a fee specified by the driver until their seats specified left are filled.
- The administrative functionalities linked to this aspect entail an administrator's approval of citizen posts before they are displayed to ensure public decency and prevent spamming. Only upon approval by the admin will the posts be visible to everyone.

4. Driving License:

- This allows citizens above 18 years of age to apply and renew for a driver's license.
- The request will be reviewed by the R.T.O Officer who conducts the test and issues the driver's license.

5. Vehicle Registration:

- This allows the user to apply for registration of number plate for vehicle on producing the invoice.
- The admin reviews and issues a number plate for the vehicle.

6. Toll Booth Management:

- This allows the Road transport division to manage the toll booth and FASTAG Plans, enabling citizens to pass the tollgate without waiting for an operator to allow a vehicle.
- The citizens can view and buy FASTAG Plans and can renew or top-up them from the *Banking Module*.

5.6.3 Entity Relationship Diagram

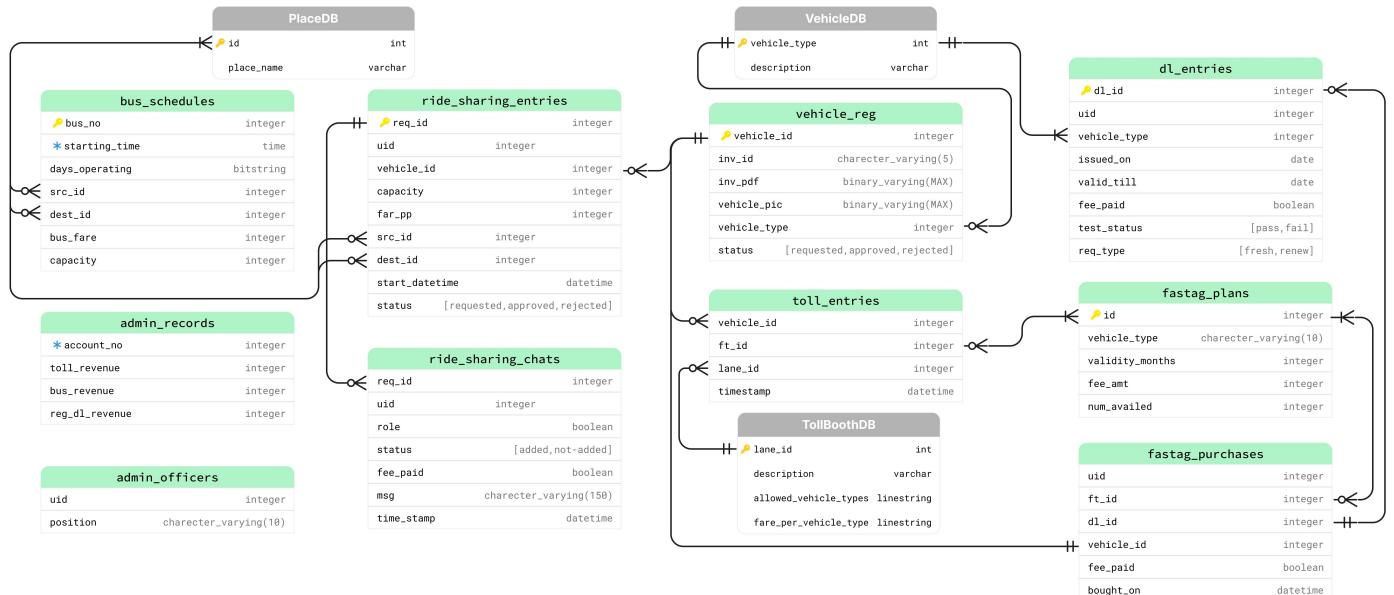


Figure 50: ER-Diagram for Transportation module

Note In this ER-Diagram the foreign key relationship of UID with the main **UserDB** isn't shown to avoid clumsiness.

1. **PlaceDB**: This database is used to store all information regarding checkpoints(busstops) present in the city
 - **id**: Unique ID to the place, serves as Primary Key
 - **place_name**: Name of the place
2. **TollBoothDB**: This database is used to store all information about lanes in Toll Plaza.
 - **lane_id**: Unique id assigned to each lane, serves as Primary Key
 - **description**: Some information about that lane
 - **allowed_vehicle_types**: Vehicle types that are allowed in each lane
 - **fare_per_vehicle_type**: Amount to be paid at the lane for that vehicle type
3. **VehicleDB**: This database is used to store all information regarding checkpoints(busstops) present in the city
 - **id**: Unique ID to the place, serves as Primary Key

- **place_name:** Name of the place

4. **bus_schedules:** This database stores all information regarding bus schedules provided to citizens.

- **bus_no:** Unique id assigned to each bus, serves as primary key
- **starting_time:** Time at which bus starts at origin/source
- **days_operating:** On what days will the bus operates
- **src_id:** Unique id assigned to starting location, foreign key to id(PlaceDB)
- **dest_id:** Unique id assigned to destination, foreign key to id(PlaceDB)
- **bus_fare:** Amount to paid to travel on the bus
- **capacity:** Number of passengers that can ride on the bus at a time

5. **admin_records:** This database is used to store all financial information for each ministry

- **account_no:** Bank account number of that minister
- **toll_revenue:** Revenue from toll gates
- **bus_revenue:** Revenue from bus fare
- **reg_dl_revenue:** Revenue from registration fees

6. **admin_officers:** This database stores the admins uid and their role/position.

- **uid:** Unique id given to all citizens, foreign key to uid(citizen database)
- **position:** The citizen's role in Transportation department

7. **ride_sharing_entries:** This database stores all ride sharing requests given by citizens.

- **req_id:** Unique id given to each request, serves as primary key
- **uid:** Unique id of the driver/carpool driver, foreign key to uid(user database)
- **vehicle_id:** Unique id given to each vehicle at the time of registration, foreign key to vehicle.id(vehicle_reg)
- **capacity:** Number of passengers the driver want to take on the ride
- **fare_pp:** Amount to be paid per person
- **src_id:** Unique id assigned to starting location, foreign key to id(PlaceDB)
- **dest_id:** Unique id assigned to destination, foreign key to id(PlaceDB)
- **start_datetime:** Date and time at which driver starts his journey
- **status:** Status of that ride-sharing request (requested, accepted, rejected)

8. **ride_sharing_chats:** This database stores chats

- **req_id:** Unique id given to each request, foreign key to req_id(ride_sharing_entries)
- **uid:** Unique id of the citizen that sent the msg
- **role:** Whether the sender is driver or rider
- **status:** Whether approved or rejected by admin

- **fee_paid:** Stores whether fee is paid for the ride or not
- **msg:** Stores the msg sent
- **time_stamp:** Store date and time

9. **vehicle_reg:** This database stores about the vehicles that are requested to be registered.

- **vehicle_id:** Unique id given to each vehicle, serves as primary key
- **inv_id:** Unique id given to Invoice at the time of purchase of vehicle
- **inv_pdf:** Invoice in pdf format
- **vehicle_pic:** Vehicle photo
- **vehicle_type:** Type of vehicle, foreign key to vehicle_type(VehicleDB)
- **status:** Whether the vehicle registration request is approved/rejected/requested

10. **toll_entries:** This database stores toll gate entries of all vehicles

- **vehicle_id:** Unique id assigned to each vehicle, foreign key to vehicle_id(vehicle_reg)
- **ft_id:** Unique id issued after registering for fastag plan, foreign key to id(fastag_plans)
- **lane_id:** Unique id given to each lane, foreign key to lane_id(TollBoothDB)
- **timestamp:** Stores the date and time at the time when vehicle passes through the toll gate

11. **dl_entries:** This database stores all information about driving licenses

- **dl_id:** Unique id given to each citizen that applies for a driving license, serves as primary key.
- **uid:** Unique id of the citizen, foreign key to uid (user database).
- **vehicle_type:** Type of vehicle, foreign key to vehicle_type (VehicleDB).
- **issued_on:** Date at which the driving license is issued.
- **valid_till:** Validity of the driving license.
- **fee_paid:** Whether the registration fee is paid or not.
- **test_status:** Whether the citizen passed or failed the driving test.
- **req_type:** Whether it is a new request or renewal of an old license.

12. **fastag_plans:** This database Stors all information about fastag plans available to purchase

- **id:** Unique id given to each fastag plan.
- **vehicle_type:** Type of vehicle, foreign key to vehicle_type (VehicleDB).
- **validity_months:** Validity of the fastag plan.
- **fee_amt:** Fee for the fastag plan.
- **num_availed:** Number of people who bought that plan.

13. **fastag_purchases:** This database stores which fastag plan is purchased by the driver

- **uid:** Unique id given to each citizen, foreign key to uid (user database).

- **ft_id:** Unique id given to each fastag plan, foreign key to id (fastag_plans).
- **dl_id:** Unique id given to each citizen that applies for a driving license, foreign key to dl_id (dl_entries).
- **vehicle_id:** Unique id assigned to each vehicle, foreign key to vehicle_id (vehicle_reg).
- **fee_paid:** Whether the amount is paid or not.
- **bought_on:** Time at which the plan started to be active.

5.6.4 Data Flow Diagram

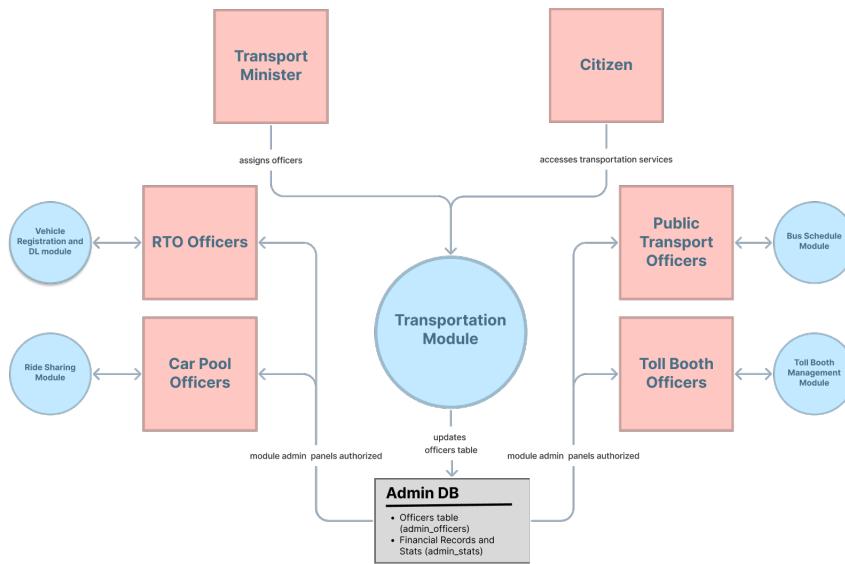


Figure 51: Level-1 DFD of Transportation Module

This shows the overall dataflow between the module and different submodules involved. Each sub-module has a subadmin.

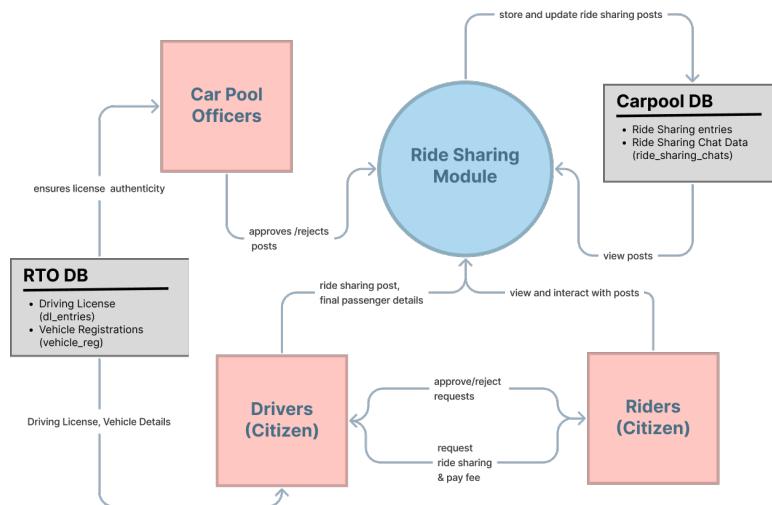


Figure 52: Level-2 DFD of RideSharing

Ride Sharing Module : This feature facilitates citizens who wish to share rides, connecting drivers with passengers.

1. Drivers can create posts for their available rides, and passengers can browse and engage with these posts by accepting or rejecting them, as well as managing fee payments.
2. All ride-sharing posts and associated data are securely stored within the Carpool DB, including entries in the `ride_sharing_entries` table and related chat interactions in the `ride_sharing_chats` table.
3. Carpool Officers are responsible for verifying the authenticity of driving licenses and vehicle details submitted by users, subsequently approving or rejecting ride requests based on this verification process.

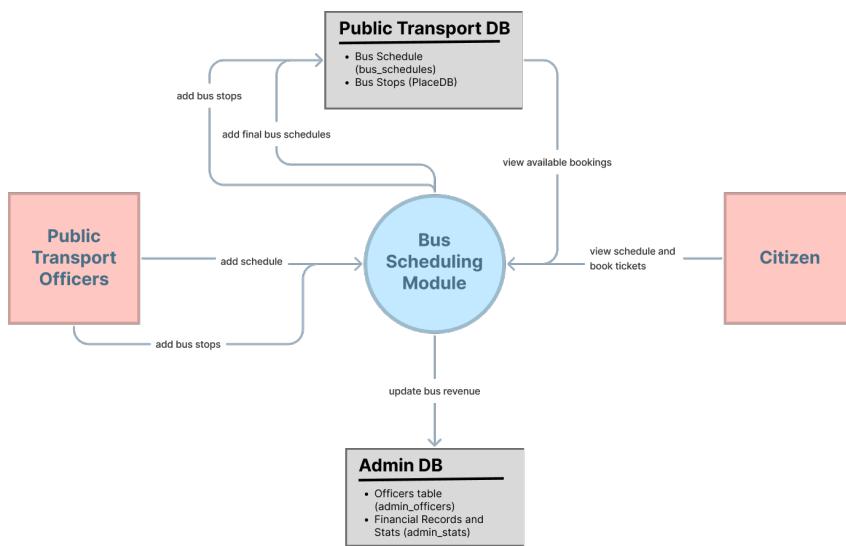


Figure 53: Level-2 DFD of BusSchedule Management

Bus Scheduling Module : This module handles the addition of bus schedules and booking of bus tickets by citizens.

1. The Public Transport Officers start the process by adding bus stops and adding bus schedules to the `bus_schedules` table of the Public Transport Database.
2. Now the Citizens can view the available schedules and book tickets.
3. Once the tickets are booked, `bus_revenue` of `admin_records` table is updated in the Admin Database.

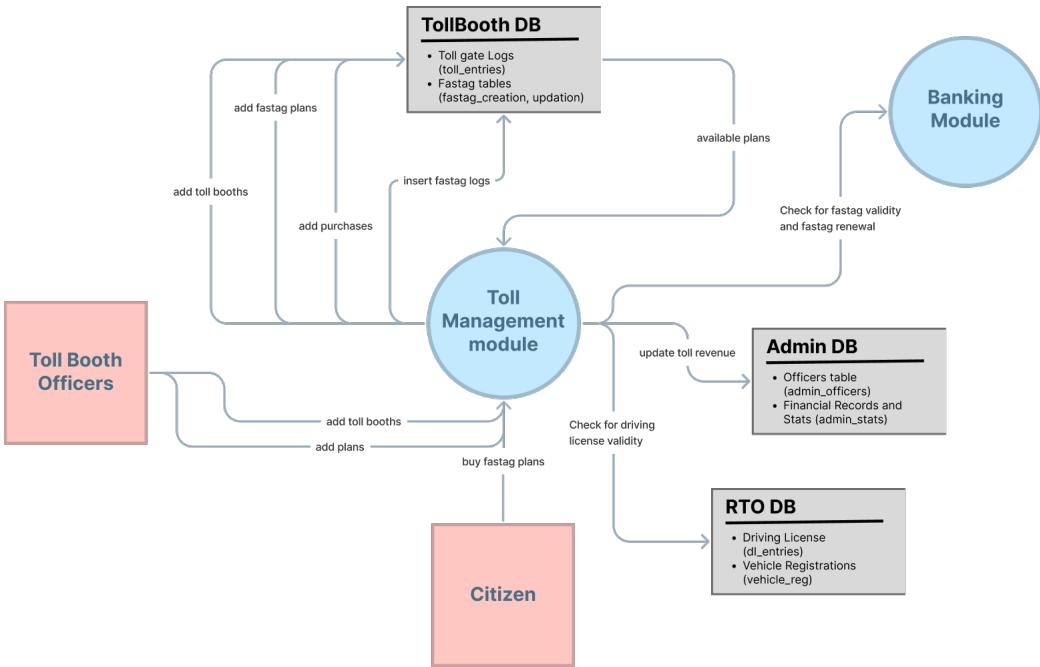


Figure 54: Level-2 DFD of Toll Management

Toll Management Module : This module handles fastag plans creation and purchases including various checks required during the process.

1. Toll Booth Officers add toll booths and fastag plans to `fastag_plans` table of the Toll-Booth Database.
2. Then Citizens buy fastag plans whose entries are added to `fastag_purchases` table of the TollBooth Database.
3. When a citizen comes to buy fastag plans, its driving license validity is checked from `dl_entries` table of RTO Database.
4. This module adds fastag logs in `toll_entries` table of the TollBooth Database.
5. This module also interacts with the Banking module which checks for fastag validity and fastag renewal.
6. At last whenever the purchase is made, the `toll_revenue` is updated in `admin_records` table of Admin Database.

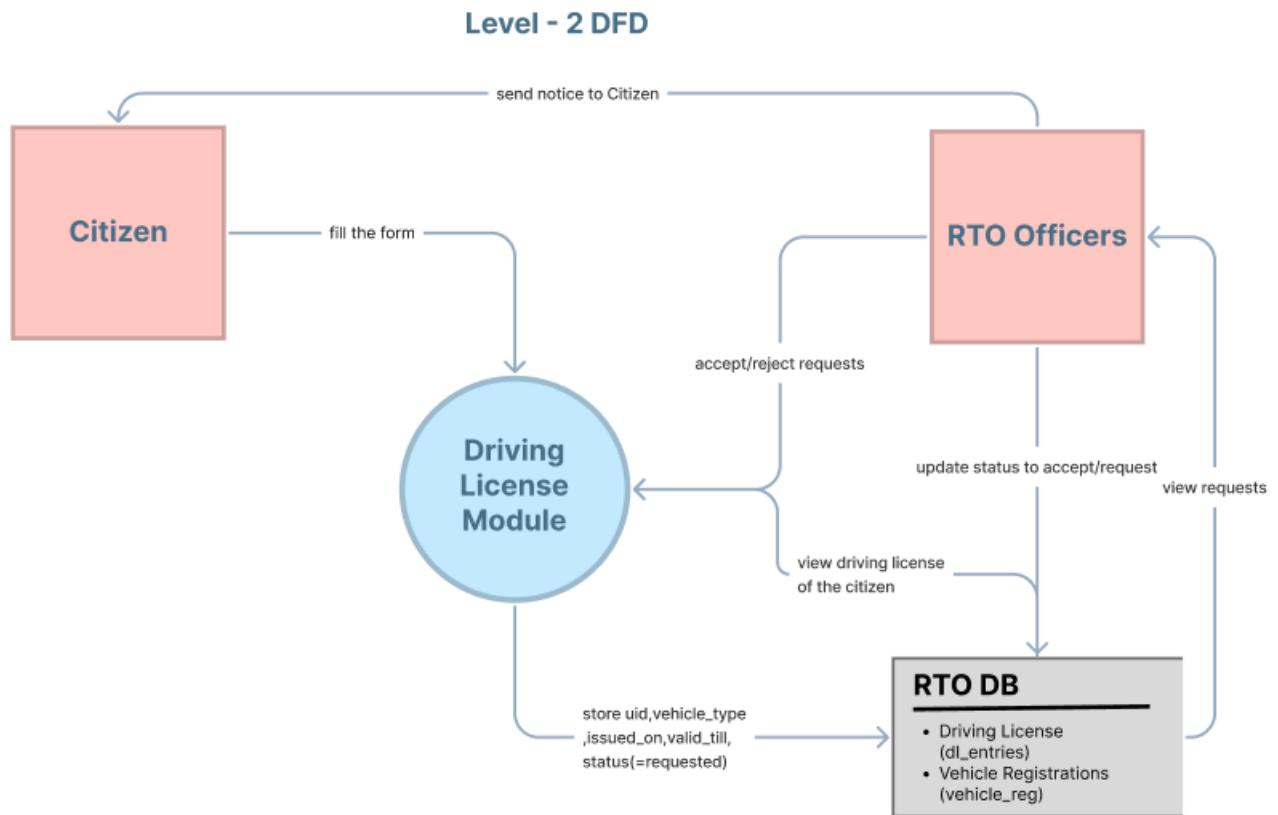


Figure 55: Level-2 DFD of Driving License Registration

Driving License Registration Module: This module handles citizens' requests for driving licenses, either new or renewal.

1. The Citizen can view their basic information already displayed on the page and should fill in the vehicle type and pay the registration fee.
2. The request is stored in the database, and the admin can accept/reject the request.
3. A notice is sent to the citizen about their request.
4. They can view their Driving License.

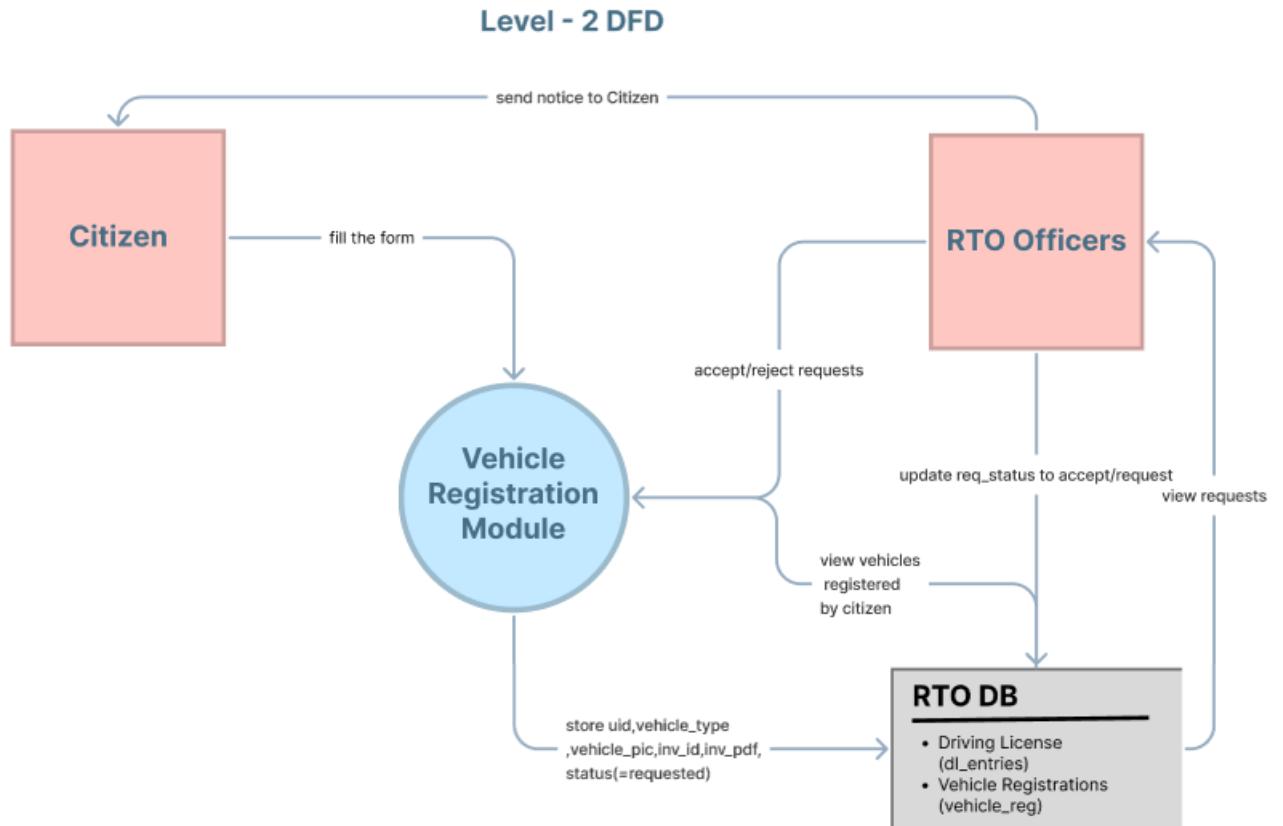


Figure 56: Level-2 DFD of Vehicle Registration

Vehicle Registration Module: This module handles citizens' requests for vehicle registration.

1. The Citizen can view their basic information already displayed on the page and should fill in other details and pay the registration fee.
2. The request is stored in the database, and the admin can accept/reject the request.
3. A notice is sent to the citizen about their request.
4. They can view their registered vehicles.

5.6.5 Interface

The following are the preliminary designs for the interface for *Transportation Module*. The team has followed template based approach in which NavajoWhite colour palette is used, The designs of unique screens are only documented in this report, the remaining screens would be similar to those given below.

Note In some places, a placeholder name like dummyVal is used, which would be duly changed during development according to the entries from database.



Figure 57: Ride Sharing Screen, where any citizen can post and view the status of their posts as well as see posts of other users.

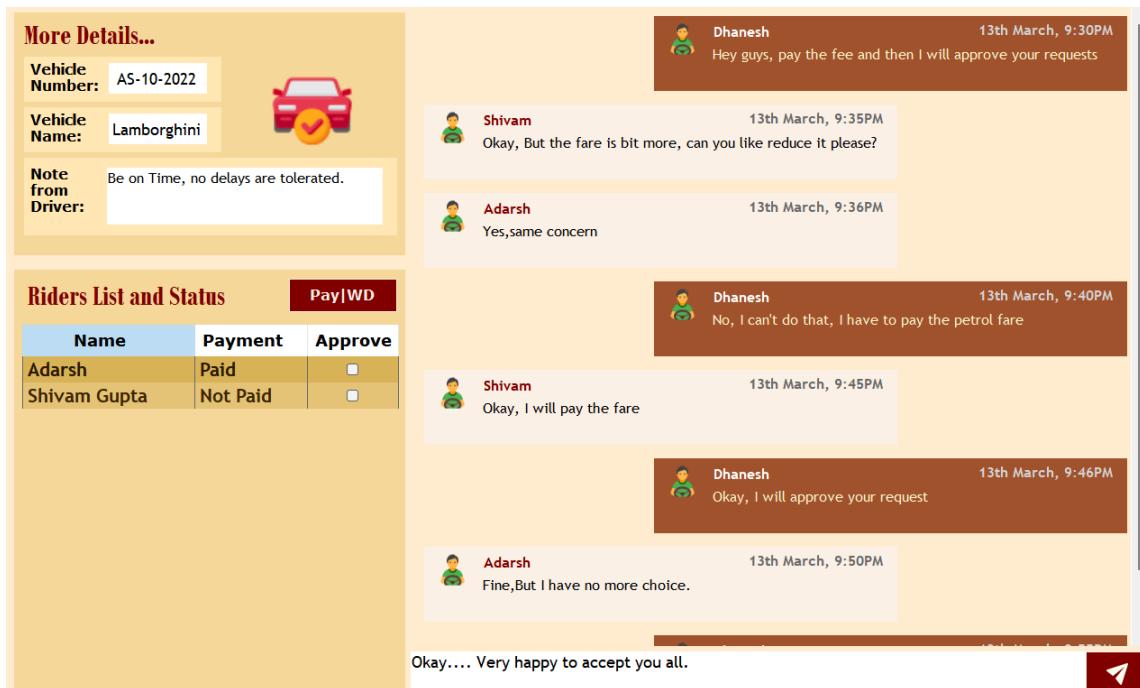


Figure 58: Ride Sharing chats, where the riders and citizens can chat and pay the fee. The drivers can accept the riders using the checkbox.



Figure 59: Homepage, that is visible only to the admins like Minister of Transportation

| Aadhar No. | Name | Designation | |
|------------|----------|-------------|--|
| DummyVal | DummyVal | DummyVal | |

Figure 60: The page where Admin can add and update officers for the sub modules.

The screenshot shows a dashboard titled "Financial Records". On the left sidebar, under "SmartGhy Transport Department", there are links for Driving License, Bus Schedule, Car Pooling, Toll Plaza, Vehicle Registration, and Admin Only. The main area displays four summary boxes: "Employee Count: {Value}", "Toll Revenue: {Value}", "DL Revenue: {Value}", and "Bus Revenue: {Value}".

Figure 61: The page where the minister can view the financial records pertaining to transport ministry

The screenshot shows a "Bus Schedule" page. The sidebar includes links for Driving License, Bus Schedule, Car Pooling, Toll Plaza, Vehicle Registration, and Admin Only. The main content area shows a table of bus routes from Adabari to Maligaon, with 42 seats available for each. To the right, there are dropdowns for "Current Day" (Monday), "Source" (Adabari), and "Destination" (Maligaon). Below these is a map showing the route from Adabari to Maligaon, with travel times of 13 min (5.4 km) and 14 min (3.7 km) indicated.

| Bus Number | Pickup Time | Fare | Seats Left |
|------------|-------------|----------|------------|
| 57A | Adabari | Maligaon | 42 |
| 57A | Adabari | Maligaon | 42 |
| 57A | Adabari | Maligaon | 42 |
| 57A | Adabari | Maligaon | 42 |
| 57A | Adabari | Maligaon | 42 |
| 57A | Adabari | Maligaon | 42 |
| 57A | Adabari | Maligaon | 42 |
| 57A | Adabari | Maligaon | 42 |

Figure 62: Bus Schedules page for citizens to view and get ticket for bus schedules by starting and final destination.



Figure 63: Fastag plans creation and buying screen, where user can view the plans sorted according to the vehicle type and can buy it.

Figure 64: Driving License Request Screen, where the citizen can request and view their driving license.

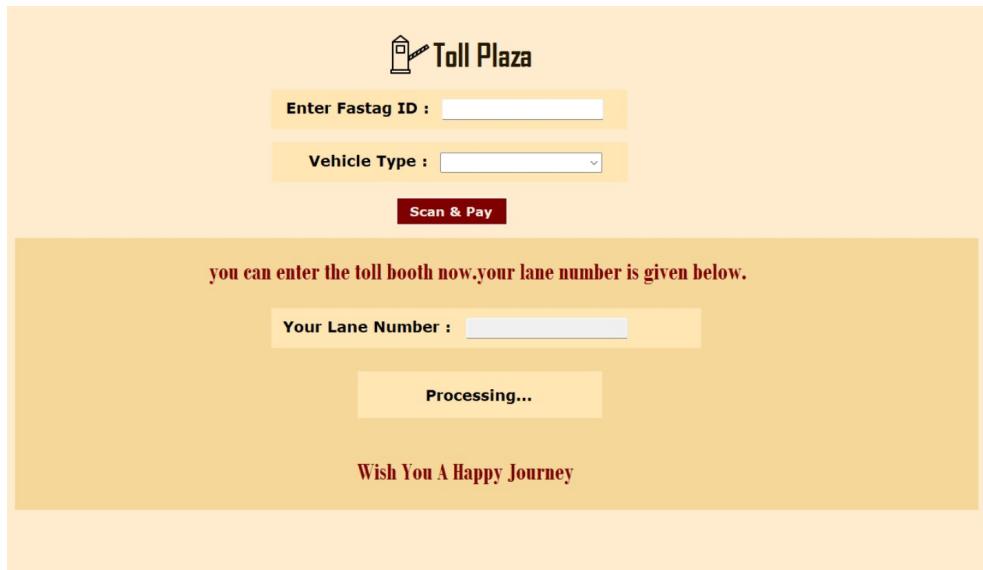


Figure 65: Toll booth user side, the user can request to enter a toll booth and then the processing happens using their FASTAG automatically, after which they can move out of the lane.

5.7 Festival and Event Planning Module

5.7.1 Introduction

The event management module offers a centralized platform for vendor registration, enabling service providers to showcase their offerings. Customers can conveniently browse and book events with their chosen vendors, tailored to their desired dates and preferences, ensuring hassle-free event planning and execution.

5.7.2 Features

1. **Vendor Registration Page:** Vendors can easily register on the platform, providing essential details such as contact information, services offered, portfolio, and pricing. This streamlined process enhances their visibility to potential customers.
2. **New Event Registration by Customer:** Customers can effortlessly register new events tailored to their preferences. From weddings to corporate functions, customers can explore vendors, select services, and book dates seamlessly through the intuitive interface.
3. **Vendor Login:**
Registered vendors gain access to a personalized dashboard where they can view their scheduled events, manage bookings, and track the history of events hosted by them. This feature enables vendors to stay organized and effectively manage their commitments.
4. **Customer Login :** Customers have the convenience of logging in to their accounts to view their scheduled events and access invoices for both past and upcoming events. This functionality empowers customers to track their event history and seamlessly manage payments, enhancing overall transparency and convenience.

5.7.3 Entity Relationship Diagram

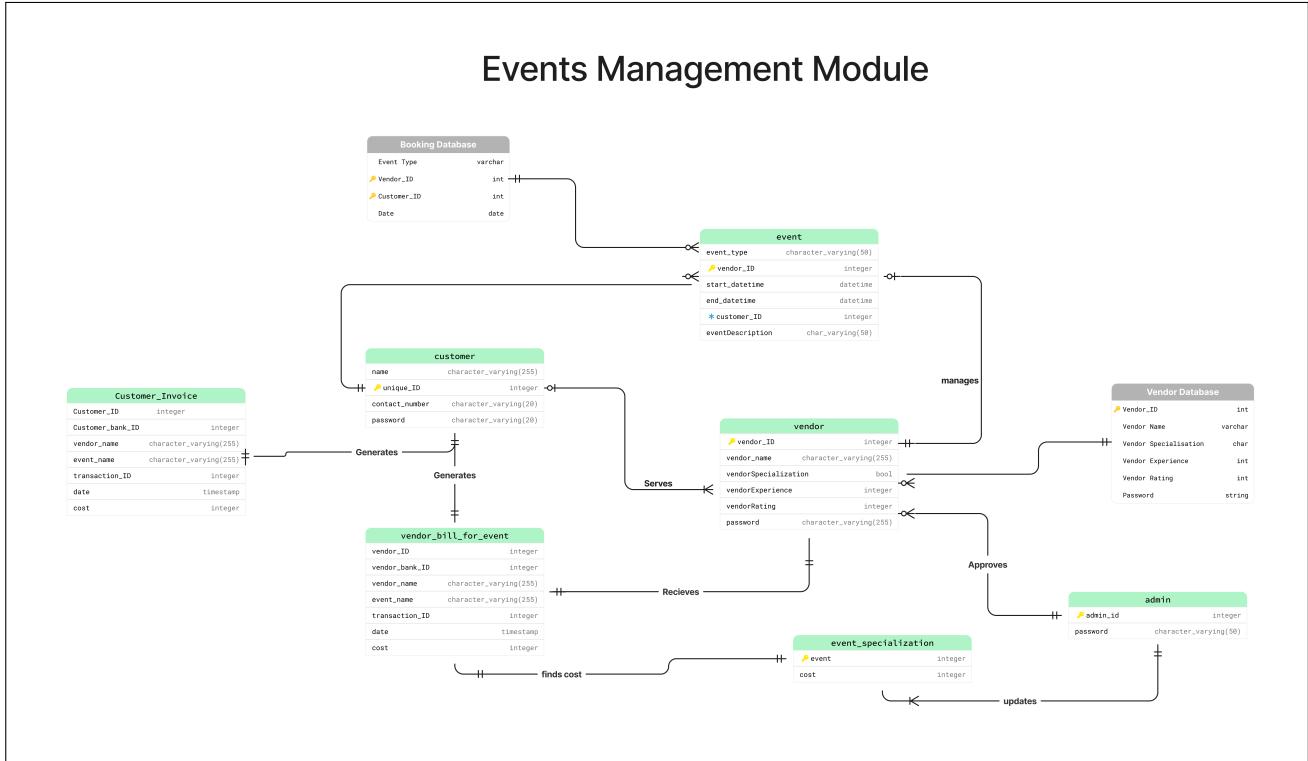


Figure 66: ER Diagram of Event Management Module

- Event**: The "Event" entity stores details such as event type, start and end date-time, vendor and customer IDs, and event description, facilitating efficient event organization.
- Customer**: The "Customer" entity records customer information including name, unique ID, contact number, and password, ensuring personalized and secure access to event management services.
- Vendor**: The "Vendor" entity represents service providers, capturing vendor details such as ID, name, specialization, experience, rating, and password for effective vendor management.
- Admin**: The "Admin" entity manages system operations, with attributes including admin ID and password, enabling administrators to oversee user accounts and system activities.
- Customer Invoice**: The "Customer Invoice" entity tracks billing details for customers, including customer and vendor IDs, event name, transaction ID, date, and cost, ensuring transparent invoicing processes.
- Vendor Bill for Event**: The "Vendor Bill for Event" entity records billing information from the vendor's perspective, with attributes such as vendor and event details, transaction ID, date, and cost, facilitating efficient payment management for services rendered.

5.7.4 Data Flow Diagram

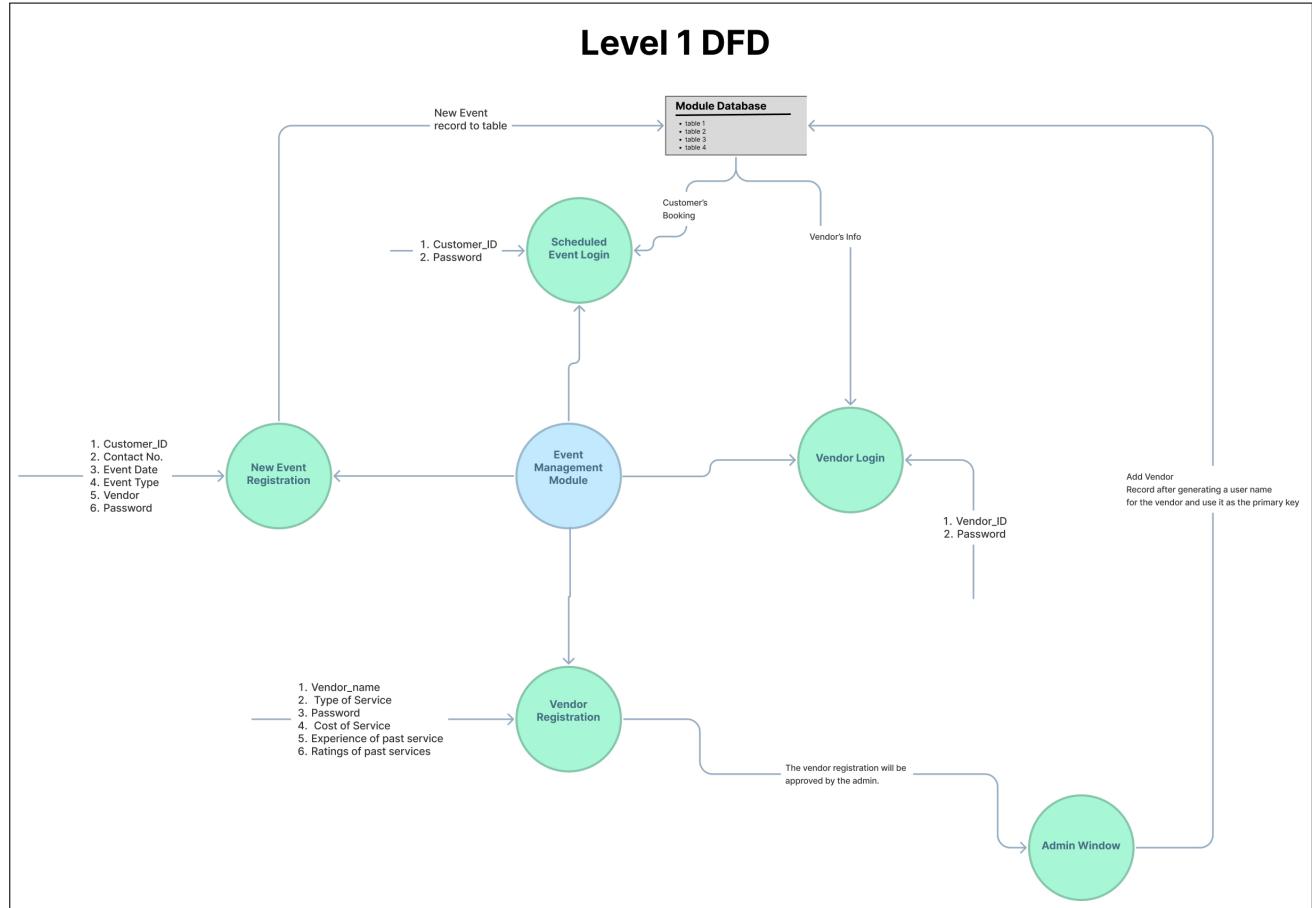


Figure 67: DFD Level 1 of Event Management Module

The event management module is structured around four integral submodules: vendor registration, vendor login, new event registration, and existing bookings login. These components serve distinct functions within the system, enabling seamless interaction between vendors and event organizers. Centralizing the data flow, a robust database houses essential tables storing vendor details and event bookings, ensuring efficient management and retrieval of critical information.

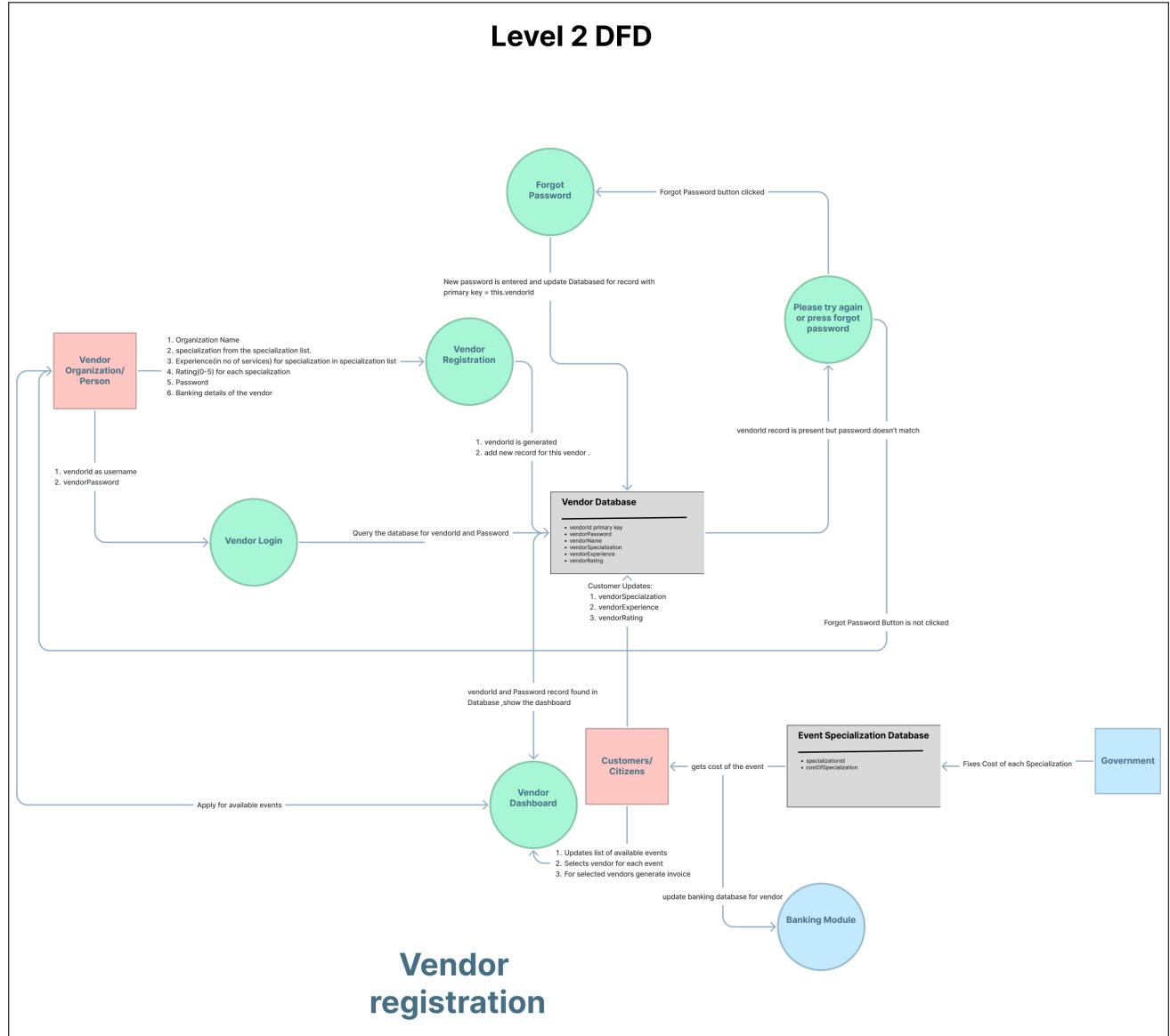


Figure 68: DFD Level 2 for Vendor Registration Submodule

When a vendor registers for the first time, he brings along with him the data as envisioned in the above DFD, thereby yielding to the module, which delivers a "vendorID" that shall be his identifier in this module and fixes a password for access later. This data is then inserted into the Vendor Database. During future attempts at login the vendor shall be prompted to provide this vendorID, which shall function as his username and the password. We endeavour to rescue the vendor in case he loses track of his password and thereby when a mismatch in password occurs upon a successful query of the aforementioned database, does the option to trigger the "Forgot Password" module gets triggered, a click upon which resets the password and updates the said database; else the user is redirected back to the login screen, denoted above as the dataflow; back to the vendor. Upon a successful login/register, the vendor is directed henceforth to a "Vendor Dashboard" that facilitates the viewing of past history of service and the Invoices appended. Him, specifying his desired specialization in fact locks him to only be eligible for selection to that particular specialization. The parameters of Rating and his Experience are interdependent on the Customers (herein the "Event Registration Page" of the Customer in

question) which upon positively selecting a vendor updates his record in the vendor Database for the aforementioned parameters. The Cost of each event of a specialization type shall remain fixed and decided by the Government and is tabulated in the "Event Specialization Database", which our sub module shall query to bill the event for which service is rendered. Finally our sub module imperatively also queries/updates the Banking database in the said situation.

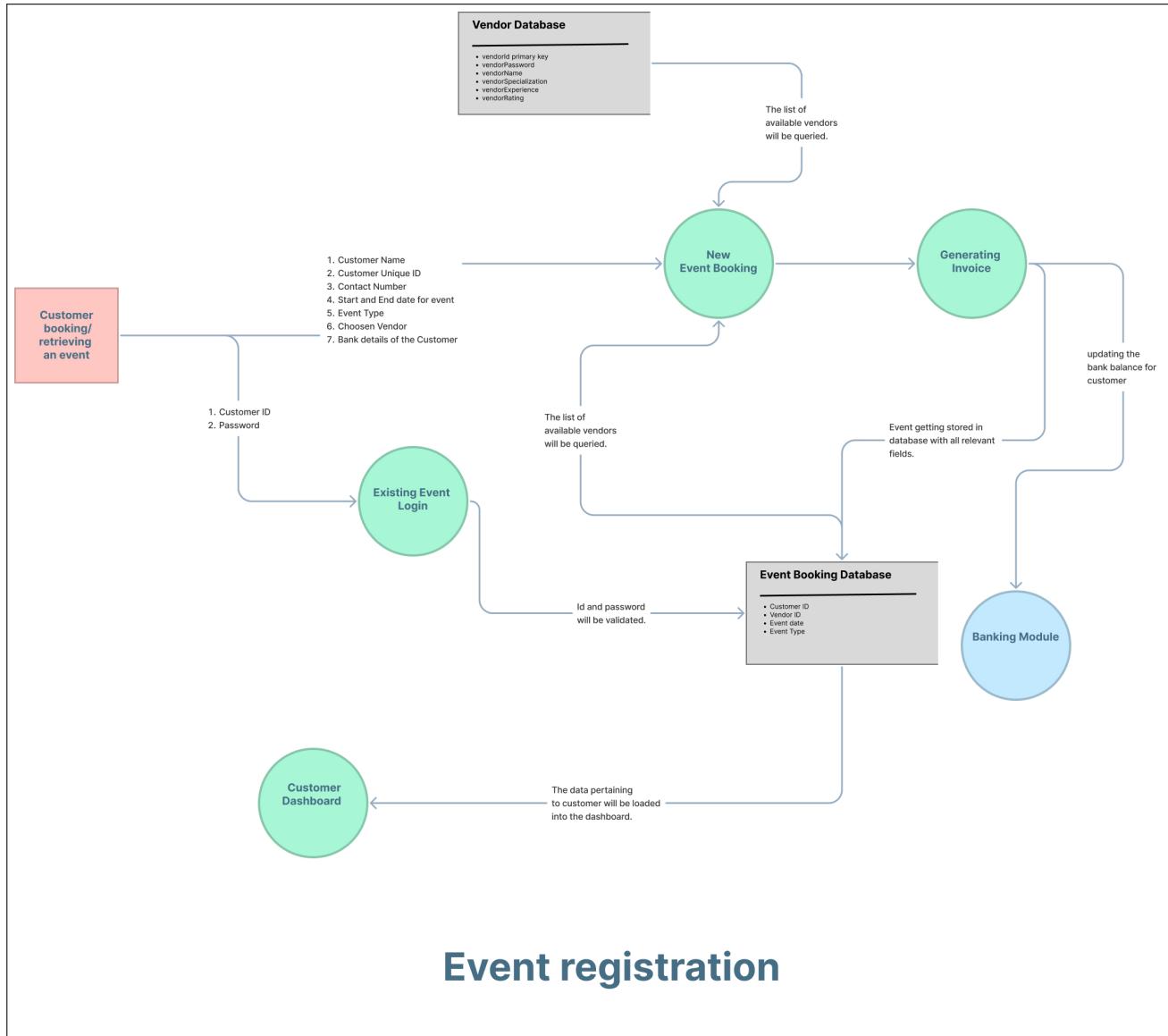


Figure 69: DFD Level 2 for Event Registration Submodule

The event registration submodule streamlines the process of booking new events by guiding customers through inputting essential event details. After providing this information, customers are presented with a list of available vendors to choose from. Upon selecting a vendor, they proceed to the payment interface to complete the booking. Customers can conveniently log in using their unique credentials to access their dashboard, where they can view their booked events and make necessary modifications, ensuring a seamless and personalized experience tailored to their preferences.

5.7.5 Interface

The screenshot shows the 'Event Registration' page of the SmartGhy Event Management system. The left sidebar has links for Event Registration, Existing Event Login, Vendor Registration, Vendor Login, and Admin Only. The main area has fields for Name, Aadhaar Number, Contact No., Event Start Date (set to 15 March), Event End Date (set to 15 March), Event Type, Vendor ID, and Password. A 'Register' button is at the bottom. To the right is a 'Vendor List' table with four rows:

| Vendor Name | Rating | Experience | |
|---------------------|--------|------------|--|
| ABC Traders | 4.7 | 52 | |
| Ramesh and Sons | 3.9 | 142 | |
| Modern Trade Center | 4.88 | 12 | |

Below the table are payment details: Amount: ₹0.00 and Total: ₹0.00.

Figure 70: Event Registration Dashboard: This is the registration form that is universally accessible to all customers who wish to register an event and then select an available vendor

The screenshot shows the 'Customer Dashboard' page. The left sidebar has links for Event Registration, Existing Event Login, Vendor Registration, Vendor Login, and Admin Only. The main area displays 'Booked Events' with three entries:

| Event Date | Vendor Name | Transaction ID | |
|------------------|---------------------|----------------|--|
| 18 April 2024 | ABC Traders | 121551215451 | |
| 22 May 2024 | Ramesh and Sons | 898551215451 | |
| 26 November 2024 | Modern Trade Center | 999962454225 | |

Figure 71: Customer Screen: This screen gets displayed when an customer logs in for his pre-booked events.

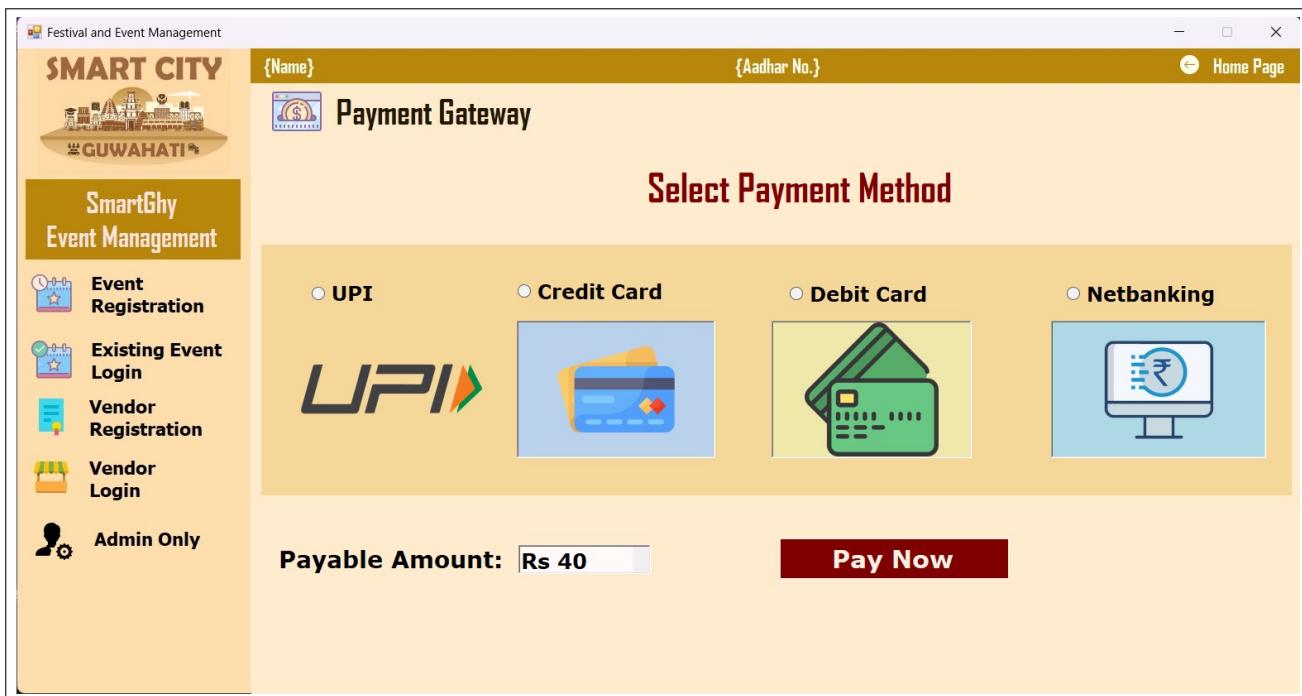


Figure 72: Payment Interface Window: The user will be redirected to this payment page upon clicking register button.



Figure 73: Vendor Registration Dashboard: This is the inner screen through which the vendor registers one of his specialisation

| INVOICE | | | |
|-----------------------|--|---------|--------------------|
| Billed To | 432790,Customer_1 | | |
| Pay To | 1235890, Vendor_1 Bharalumukh, Guwahati-781009 | | |
| Bank | ABCD Bank | | |
| Account Number | xxx-xxx-xxxx | | |
| TRANSACTION ID | RATE | DAYS | AMOUNT |
| 902749224 | 1024 | 4 | |
| | | | |
| Sub Total | ₹4096 | Tax 138 | TOTAL ₹4234 |

Figure 74: Upon clicking the "SHOW" button in the vendor and customer dashboards the INVOICE for that transaction shall open

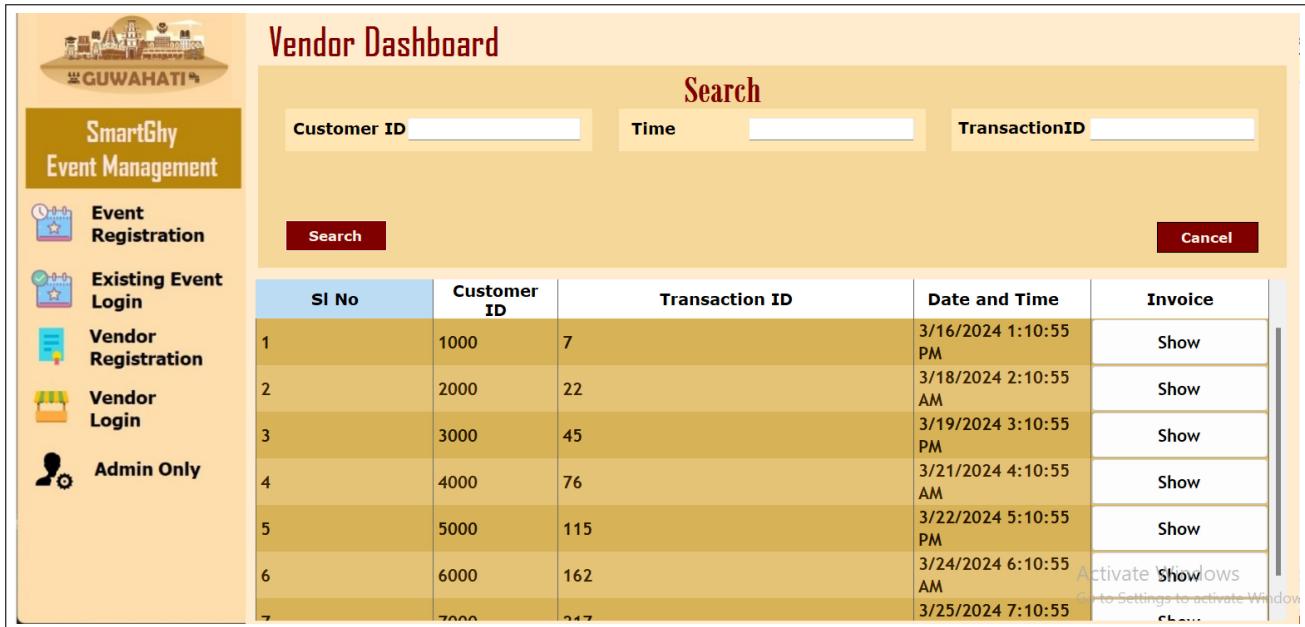


Figure 75: Vendor Dashboard:Upon Successful login of the vendor,this inner screen enables the vendor to view his service history

5.8 Complaint and Redressal Portal

5.8.1 Introduction

Welcome to the Complaint and Redressal Portal, a vital component of our commitment to building a smarter, more responsive city. Designed to address concerns swiftly and efficiently, our platform serves as a conduit for citizens to voice their grievances and seek resolution effectively. With a focus on precision and conciseness, we provide a streamlined interface for lodging complaints and tracking their progress, ensuring transparency and accountability in our governance.

Here, users can easily submit complaints regarding different departments like electricity and police. Department administrators promptly address these issues, ensuring efficiency and transparency.

5.8.2 Features

The Complaint and Redressal Portal Portal has a variety of features which makes it a modern complaint management system. The features can be explained user-wise namely, Admin and User.

1. User:

- (a) **Dashboard:** In the user dashboard, user can conveniently track the status of your complaints. In the dashboard user can find the number of complaints in progress, those resolved, and those not yet processed.

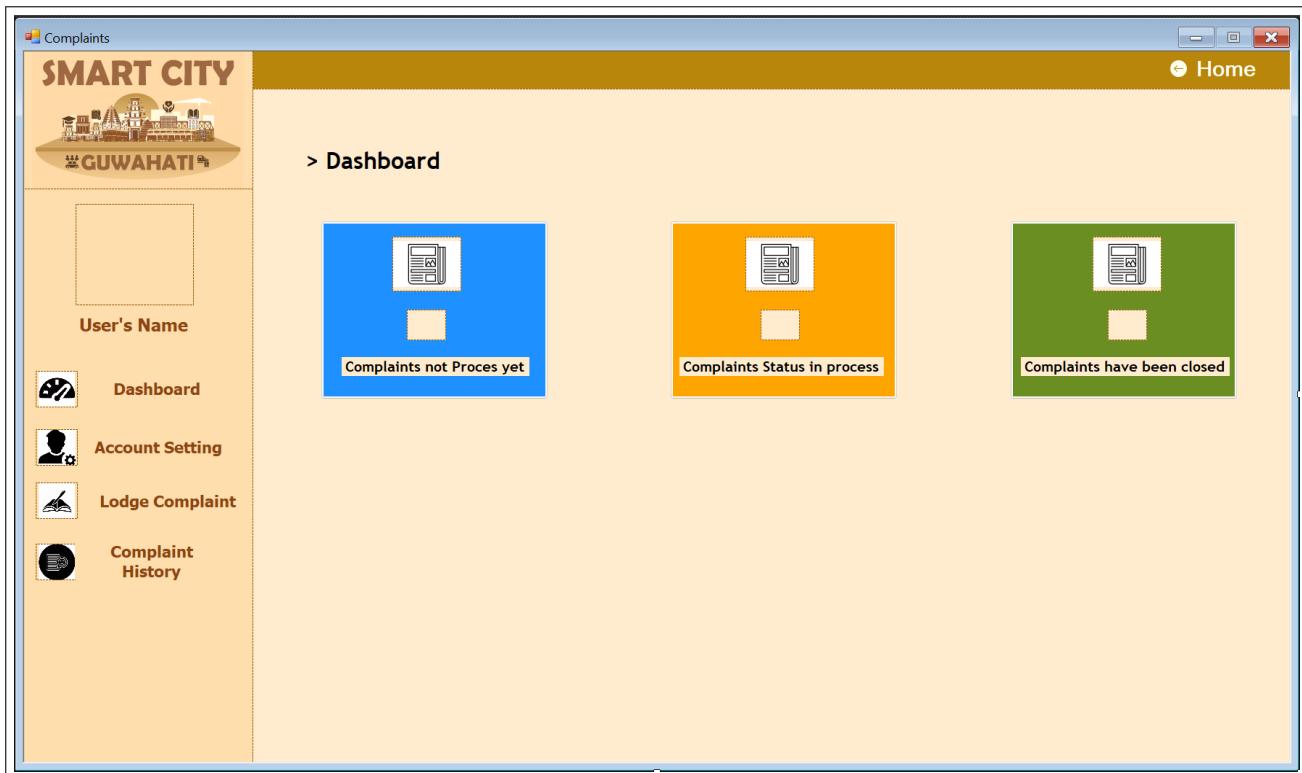


Figure 76: User Dashboard: Upon Successful login of the User, this inner screen enables the Admin to view his complaint numbers

- (b) **Lodge Complaint:** Users have the ability to submit new complaints through this page, providing essential details including the department (e.g., Police, Hospital), priority level (Low, Medium, High), nature of the complaint (General Query or Formal Complaint), as well as a concise title and detailed description.

The screenshot shows the 'Register Complaint' form. It has a similar sidebar to the dashboard, with the 'Lodge Complaint' option selected. The main form area is titled '> Register Complaint' and contains the following fields:

| | |
|---------------------------------------|----------------------|
| Department | Priority |
| <input type="text"/> | <input type="text"/> |
| Nature Of Complaint | Complaint Title |
| <input type="text"/> | <input type="text"/> |
| Description | |
| <input type="button" value="Submit"/> | |

Figure 77: This screen Shows the complaint form

- (c) **Complaint History:** Users have the capability to access their historical complaints, along with the current status and remarks provided by administrators. Additionally, users can efficiently search for specific complaints and access detailed information regarding each complaint's status and progression.

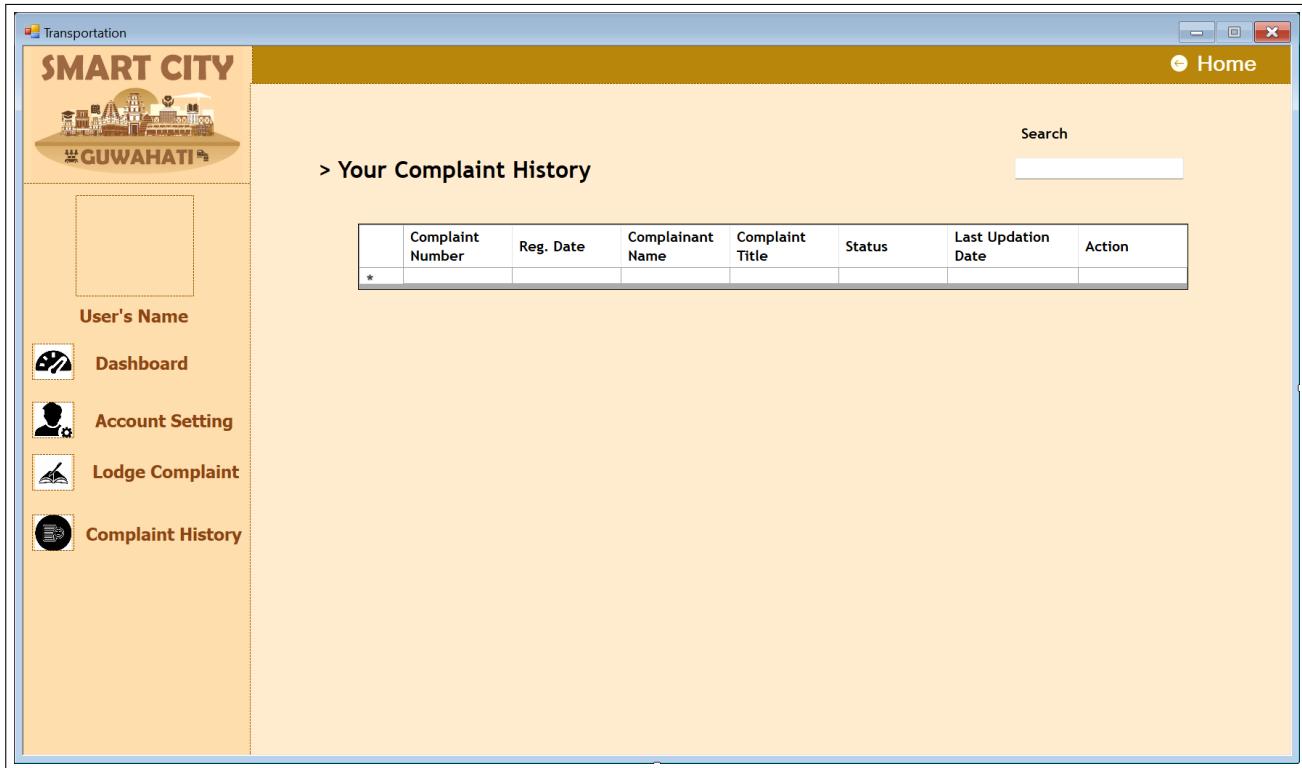


Figure 78: This screen Shows the complaint History

- (d) **Account Info:** The Account Settings page provides users with the functionality to both review and edit their personal details. These details include their name, email address, user ID, physical address, pin-code, and contact number.

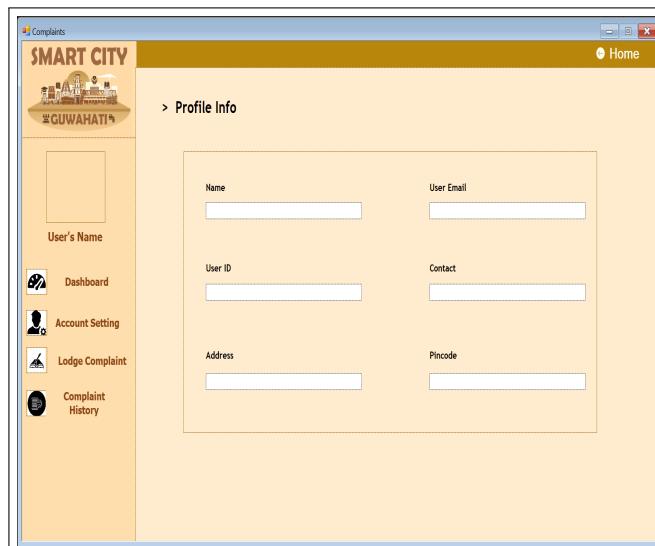


Figure 79: This screen Shows the User Profile Details

2. Admin:

- (a) **Dashboard:** In the Admin dashboard, admin can conveniently track the status of your complaints filed by the users in that department. In dashboard admin find the number of complaints in progress, those resolved, and those not yet processed.

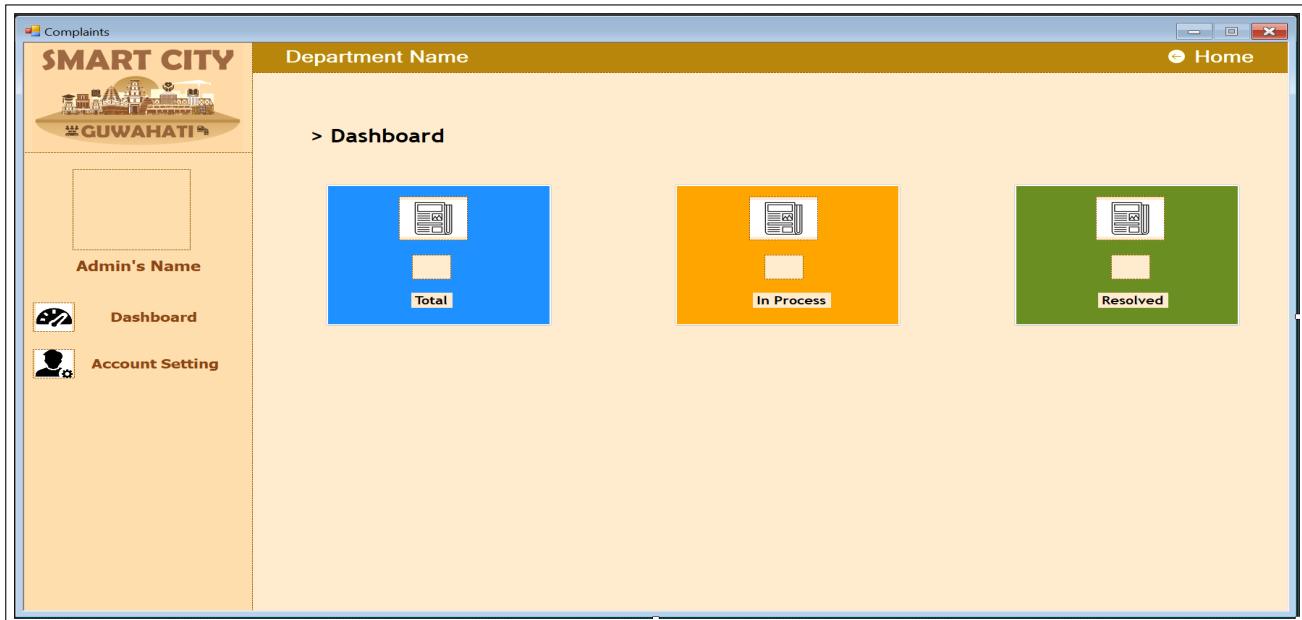


Figure 80: Admin Dashboard: Upon Successful login of the admin, this inner screen enables the Admin to view complaint numbers

- (b) **Complaint History:** Admins now have access to a comprehensive overview of complaints filed by users, organized in a convenient table format. This table displays the total number of complaints, categorizing them based on their current status:

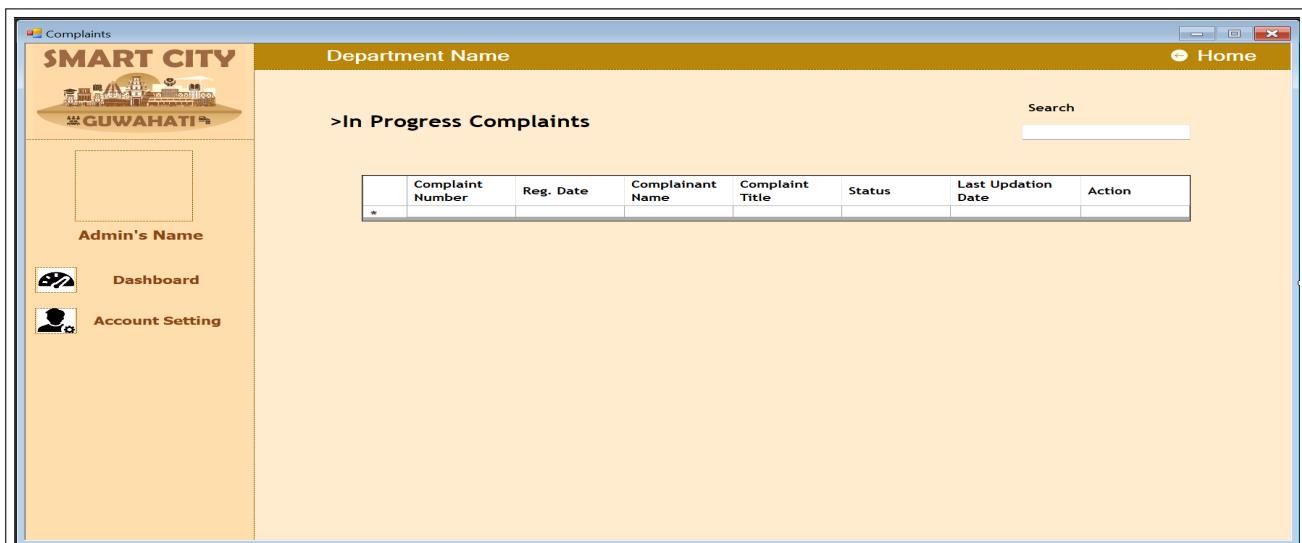


Figure 81: This screen Shows the all complaint filed in that department

- i. **View Complaint:** In the Complaint History page, admins can now select a complaint to view all associated details, including the title, type of complaint, registration date, and its description. Additionally, on this page, admins have the option to take action on the complaint and view the details of the user who filed the complaint.

The screenshot shows a window titled "Complaints" with a "SMART CITY GUWAHATI" logo. On the left sidebar, there are links for "User's Name", "Dashboard", and "Manage Complaint". The main content area is titled "Department Name" and contains a form for viewing complaint details. The form includes fields for Complaint Number, Complainant Name, Reg Date, Complaint Title, Last Updation Date, Department, Final Status, and Remark. At the bottom, there are buttons for "Take Action" and "View User Details".

Figure 82: This screen shows details of selected complaint from filed complaints

- ii. **Take Action:** Now, admins have the capability to take action on the complaint by modifying its status, such as marking it as "in progress" or "resolved." Additionally, admins can provide remarks or comments regarding the complaint, facilitating effective communication and documentation of the resolution process.

The screenshot shows a window titled "Complaint Number" with a dropdown menu for "Status". Below the status dropdown is a large text area for "Remark". At the bottom, there are buttons for "Submit" and "Close this window".

Figure 83: This screen shows take action page

iii. **View User Details:** Admins see the details of user who filed the complaint.

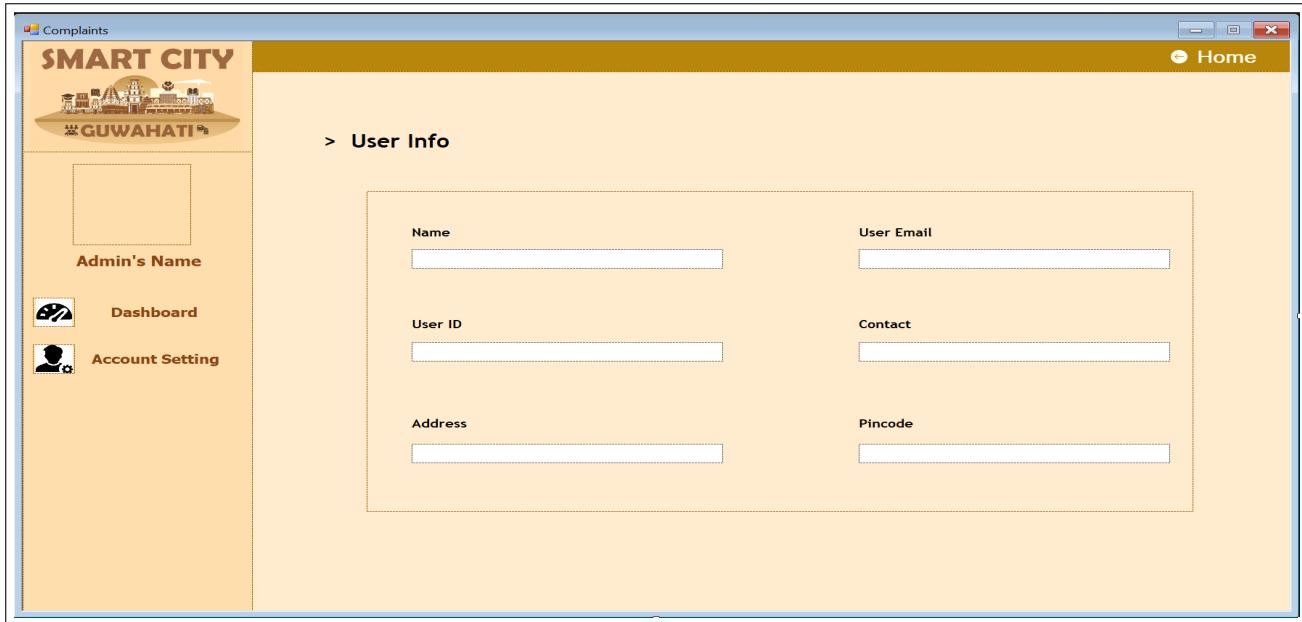


Figure 84: This screen shows details of complainant

- (c) **Account Info:** The Account Settings page provides admins with the functionality to both review and edit their personal details. These details include their name, email address, user ID, physical address, pin-code, and contact number.

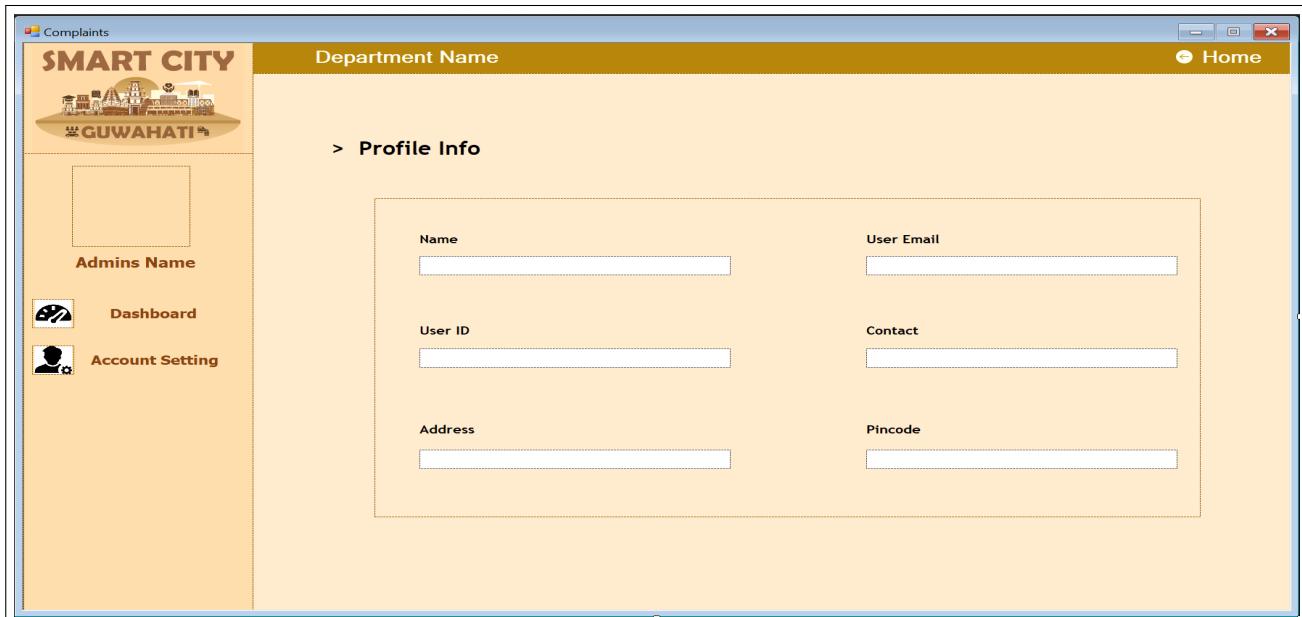


Figure 85: This screen shows account info of admin

5.8.3 Entity Relationship Diagram

Departments: This entity represents the departments against which complaints can be filed. It has two attributes:

- **DepartmentID:** The primary key, which uniquely identifies each department.
- **DepartmentName:** The name of the department.

Complaints: This entity represents the complaints filed by users. It has several attributes:

- **ComplaintID:** The primary key, which uniquely identifies each complaint.
- **UserID:** A foreign key that links to the Users entity, indicating which user filed the complaint.
- **DepartmentID:** A foreign key that links to the Departments entity, indicating against which department the complaint is filed.
- **Description:** The details of the complaint.
- **Status:** The current status of the complaint.
- **Timestamp:** The time when the complaint was filed.
- **PrincipalRemark:** Remarks made by the principal.
- **AdminRemark:** Remarks made by the administration.

AdminActions: This entity represents the actions taken by the administration on the complaints. It has five attributes:

- **action_Id:** The primary key, which uniquely identifies each action taken by the administration.
- **admintid:** A foreign key that links to the Administrations entity. It indicates which administration member took the action.
- **complaintId:** A foreign key that links to the Complaints entity. It indicates on which complaint the action was taken.
- **action:** This attribute describes the action taken by the administration on the complaint. It could be an action like 'Resolved', 'Pending', 'Escalated', etc.
- **timestamp:** This attribute records the time when the action was taken.

User and Admins: Users and admins can be directly utilized from the user and registration module.

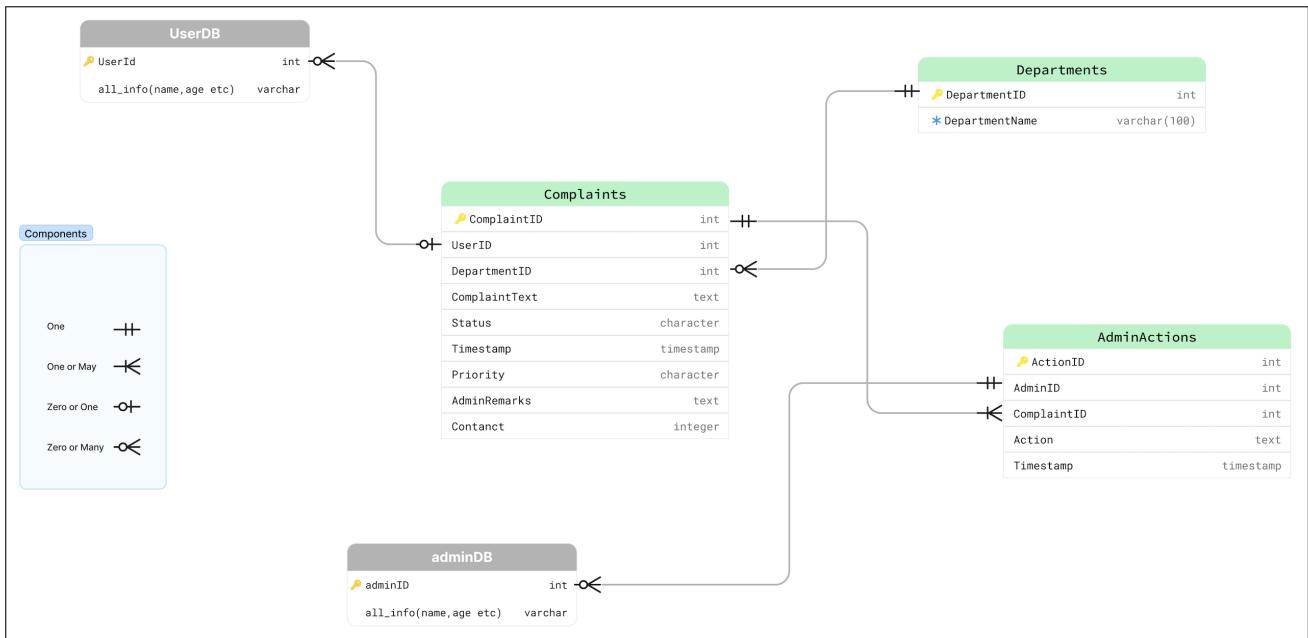


Figure 86: ER Diagram for Complaint and Redressal Module

5.8.4 Data Flow Diagram

External Entities:

- **User:** The person who logs in, registers a new complaint, or tracks filed complaints.
- **Admin:** The person who logs in, views the complaints filed by users in their department, and updates the status of the complaints.

Processes:

- **Register Complaint:** When a user logs in, they can register a new complaint. The data for the new complaint is stored in the “Complaint Form” data store.
- **Track Complaints:** When a user logs in, they can track the progress of their filed complaints through an “Account Dashboard.”
- **Update Complaint Status:** When an admin logs in, they can view the complaints filed by users in their department on the dashboard. They can select any new complaint and update its status. The updated status is stored in the “Status of Filed Complaints” data store.

Data Stores:

- **Complaint Form:** This stores the data of the new complaints registered by the user.
- **Status of Filed Complaints:** This stores the updated status of the complaints.

Data Flows:

- **From User to Register Complaint:** Represents the user registering a new complaint.

- **From Register Complaint to Complaint Form:** Represents the new complaint data being stored.
- **From User to Track Complaints:** Represents the user tracking their filed complaints.
- **From Track Complaints to Account Dashboard:** Represents the progress of the filed complaints being displayed on the dashboard.
- **From Admin to Update Complaint Status:** Represents the admin selecting a new complaint to update.
- **From Update Complaint Status to Status of Filed Complaints:** Represents the updated status of the complaint being stored.

Level-1 DFD

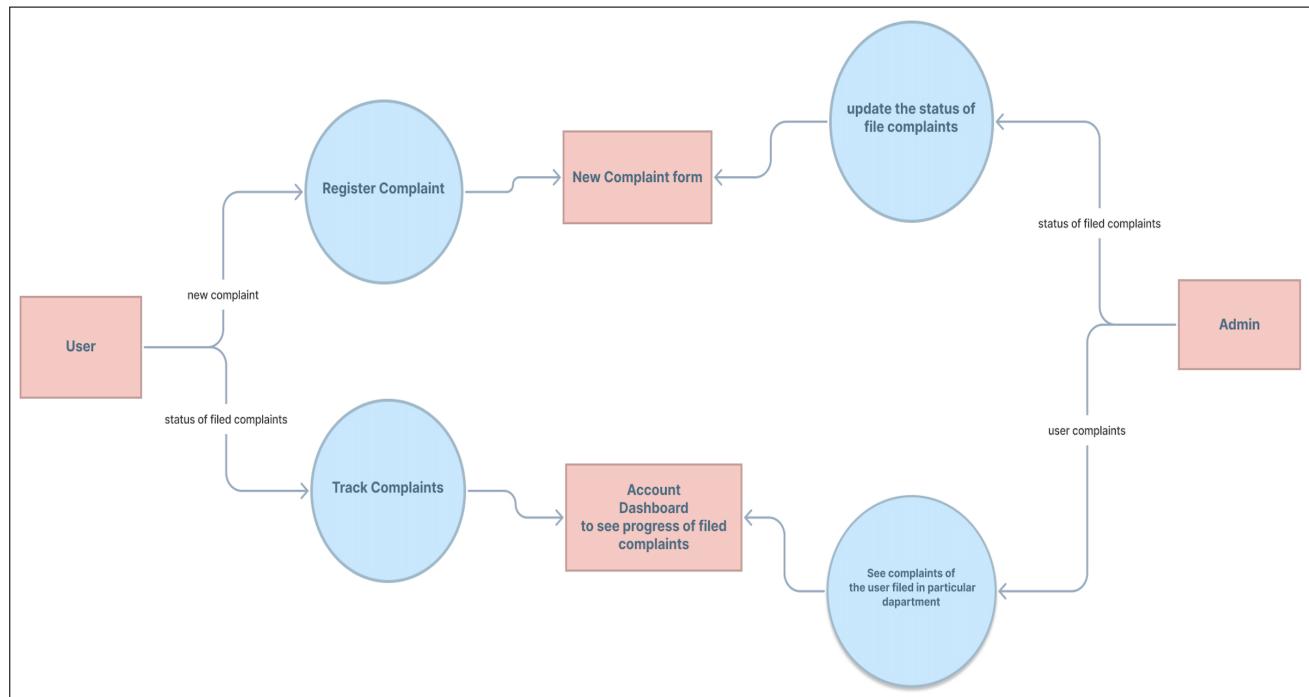


Figure 87: Level-1 DFD of Complaint and Redressal Module

Level-2 DFD

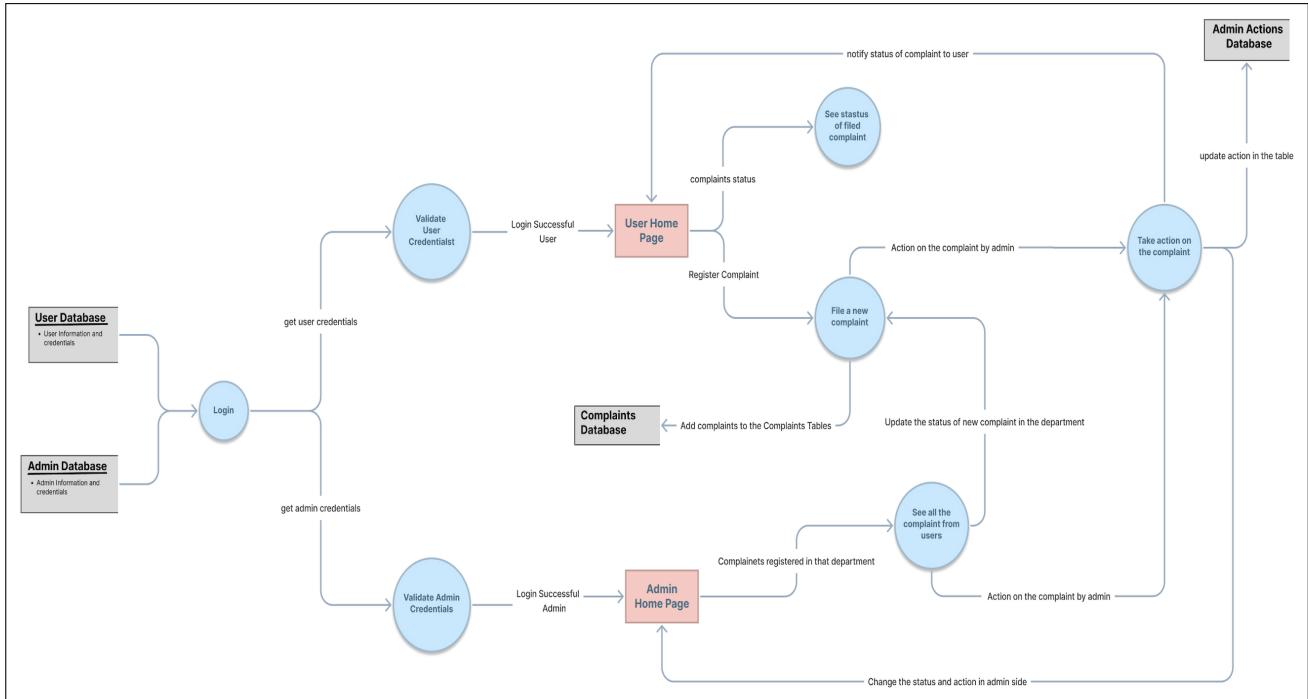


Figure 88: Level-2 DFD of Complaint and Redressal Module

5.9 Administrative Hierarchy Interface

5.9.1 Introduction

Administrative Hierarchy is responsible for the administration part of the Smart City which is the responsibility of the admins of each module/portal. In our solution, we assume that there are several ministries and each ministry's admin is the minister of it. Each ministry might have one or several modules/portals under it. The admin of this module is the Election Commissioner, who is assumed to be appointed by the Government.

Thus, this module is responsible for creating awareness among the citizens about the ministries and the ministers and the rights they have as citizens to access information. It also conducts free and fair elections and shows the organizational structure of the Smart City.

5.9.2 Features

This module boasts several features which are listed below:

1. Administrative Features:

- The admin can conduct elections and all the important activities related to it can be easily set up with the help of features offered. These features include **announcing the timeline of elections, updating code of conduct, accepting/rejecting the nominations filed, reviewing the code of conduct violations, conducting the election and releasing the list of elected candidates**.

- After the announcement of results, the organizational structure is automatically updated with the details of the newly elected ministers.
- Apart from elections, the admin can also send the appeals by citizen to gain access to certain information about the city to the concerned departments.

2. Citizen Features:

The module offers many features to ease the process of election, gaining access to information and understanding the organizational structure of the Smart City.

- (a) **Register as Voter:** Every citizen above the age of 18 can register himself/herself as a voter. Basic details are to be filled up and he/she will be given the chance to vote in the upcoming election
 - (b) **Nominate Yourself:** Any citizen can nominate themselves for any of the positions. Their nomination will be reviewed by the admin and upon acceptance, they will be able to participate in the election.
 - (c) **Know Your Candidate:** Citizens should be aware of the candidates contesting for each position. They can find the details of all the candidates in this and can decide whom to vote for.
 - (d) **Timeline:** Election is a very busy process. There are lots of deadlines and important dates involved. Citizen can find out about all the dates here and can keep themselves informed about the proceedings.
 - (e) **Code of Conduct:** Citizens can view the Code of Conduct for the elections.
 - (f) **Cast Your Vote:** Citizens can vote for each position and submit their ballot.
 - (g) **View Results:** This feature shows the results of the recent elections.
 - (h) **View Statistics:** In this feature, the citizen can view statistics of each election like the total turnout, turnout per ministry, gender ratio of the ministers elected and nominated, age-wise voters etc.
 - (i) **Past Elections:** Citizens can view the details of the past elections.
 - (j) **Report Violations:** Candidates who have violated the Code of Conduct can be reported by the citizens.
3. **Organization Structure:** Citizens can view all the ministries and ministers at one place.
 4. **Right To Information:** Citizens can submit their appeals to any ministry to gain access to certain information.

5.9.3 Entity Relationship Diagram

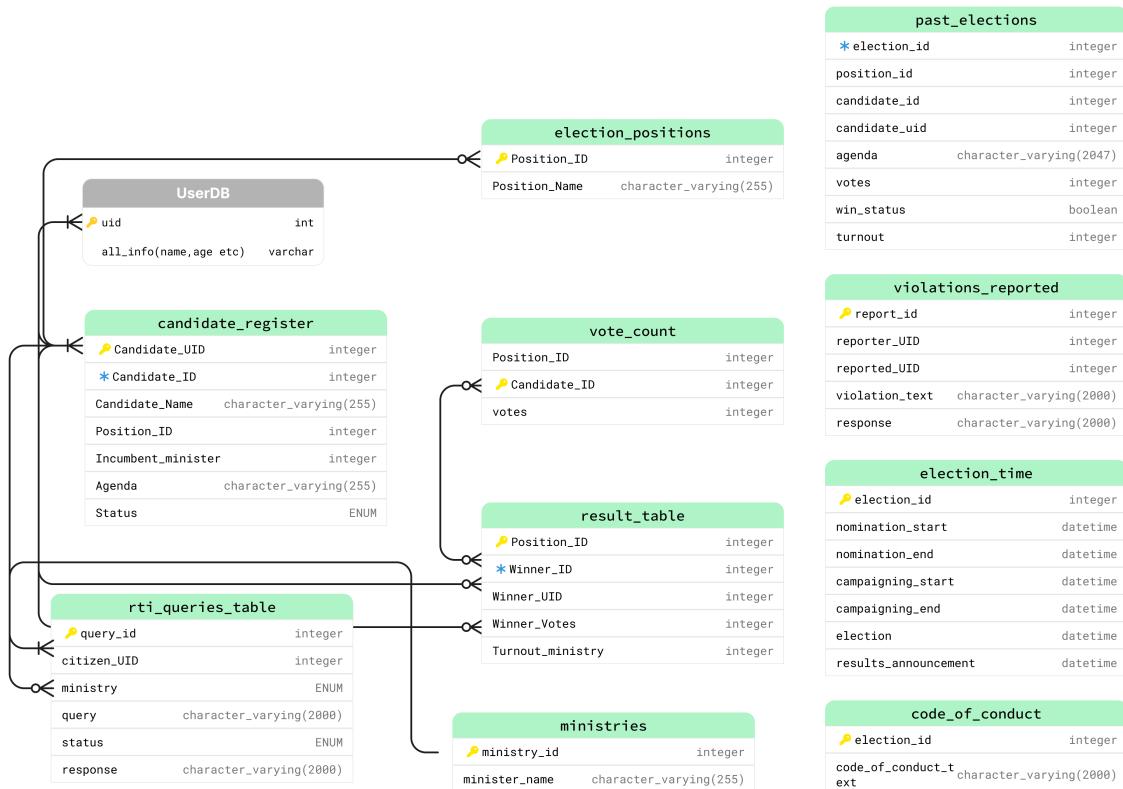


Figure 89: ER Diagram of Administrative Hierarchy Module

Note: In this ER-Diagram the foreign key relationship of UID with the main UserDB isn't shown to avoid clumsiness.

Description of each table in the diagram is given below:

1. **election_position**: Stores the positions for which the elections are held and their id.
 - **Position_ID**: Position Id of the corresponding position
 - **Position_Name**: Name of the position, for instance, education minister
2. **candidate_register**: Stores the details of the candidates that have been nominated.
 - **Candidate_UID**: UID taken from the UserDB
 - **Candidate_ID**: Each candidate that has been nominated is given an ID. Please note that this is not the same as the UID
 - **Candidate_Name**: It is the name that the candidate has filled in the nomination site

- **Position_ID:** Position ID of the position for which the citizen has been nominated
- **Incumbent_minister:** Whether he/she is currently a minister or not
- **Agenda:** Agenda of the candidate
- **Status:** Status of nomination - pending, accepted, rejected

3. **vote_count:** Stores the number of votes received by each candidate who has contested in the election

- **Position_ID:** Position ID of the position the candidate is contesting for.
- **Candidate_ID:** Candidate ID of the candidate
- **votes:** Votes received by the candidate

4. **election_time:** Maintains the dates of each election.

- **election_id:** Election ID of the election
- **nomination_start:** Date when nomination started
- **nomination_end:** Date when nomination ended
- **campaigning_start:** Date when campaigning started
- **campaigning_end:** Date when campaigning ended
- **election:** Date of the election
- **results_announcement:** Date when results are announced

5. **result_table:** Stores the results of the election.

- **Position_ID:** Position ID of the position for which the candidate has contested for
- **Winner_ID:** Candidate ID of the candidate who has won the election for the given position
- **Winner_UID:** UID of the candidate who won the election
- **Winner_Votes:** Votes received by the candidate
- **Turnout_ministry:** Total turnout for the position

6. **ministries:** Stores the names and ids of the various ministries in the Smart City.

- **ministry_id:** ID given to a ministry
- **ministry_name:** Name of the ministry
- **minister_name:** Name of the current minister

7. **rti_queries_table:** Stores the RTI queries of the citizens.

- **query_id:** ID of the query
- **citizen_UID:** UID of the citizen who has submitted the query
- **ministry:** Ministry to which the RTI is submitted
- **query:** Query
- **status:** status of the query

- **response:** response to the query by the concerned admin

8. **past_elections:** stores the details of past elections

- **election_id:** ID of the election
- **position_id:** ID of the position
- **candidate_id:** ID of the candidate
- **candidate_uid:** UID of the candidate
- **agenda:** Agenda submitted
- **votes:** Votes received by the candidate
- **win_status:** Whether the candidate has won the election or not

9. **code_of_conduct:** Stores the code of conduct of every election

- **election_id:** Election ID of the election
- **code_of_conduct_text:** Code of conduct text

10. **violations_reported:** Stores the violations reported during an election

- **report_id:** ID of the report
- **reporter_UID:** UID of the citizen who has reported the violation
- **reported_UID:** UID of the citizen who has been reported
- **violation_text:** Textual report
- **response:** Response by the admin for the report of the violation

5.9.4 Data Flow Diagram

Given below is the Level 1 Data Flow Diagram of the Administrative Hierarchy module.

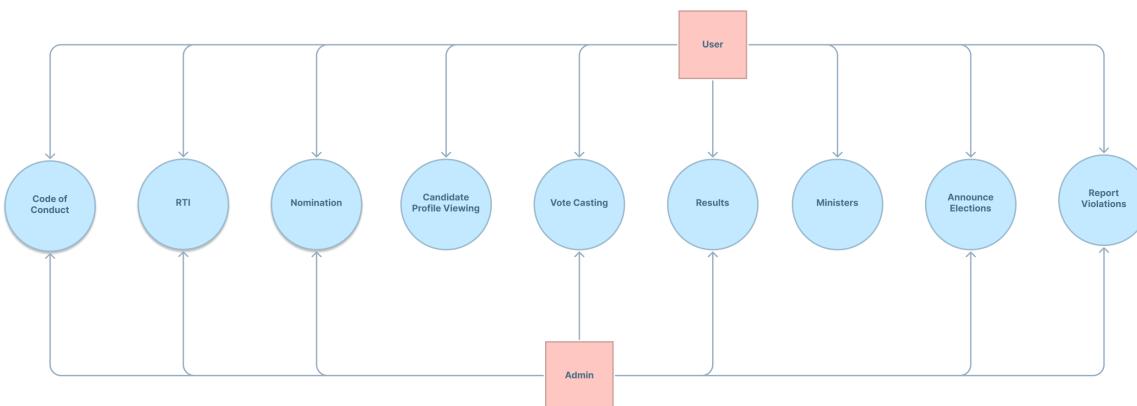


Figure 90: Level 1 Data Flow Diagram

There are 9 sub-modules.

- Code of Conduct
- RTI
- Nomination
- Candidate Profile Viewing
- Vote Casting
- Results
- Ministers
- Announce Elections
- Report Violations

Level 2 Data Flow Diagrams of each sub module are given below.

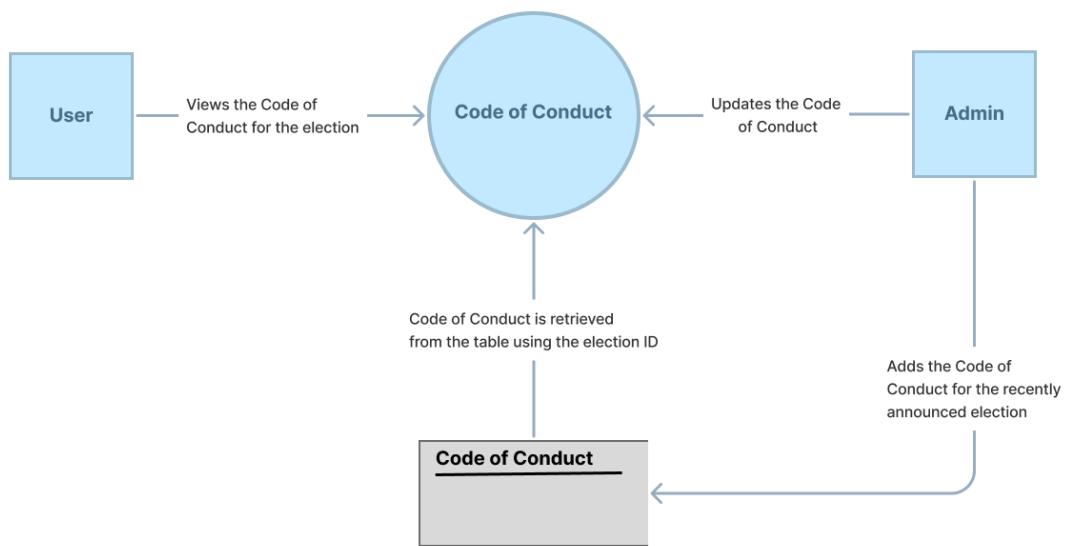


Figure 91: Level 2 Data Flow Diagram of Code of Conduct

Code of Conduct: In this module, the admin sets and updates the Code of Conduct for every election. The citizens can view these and abide by it.

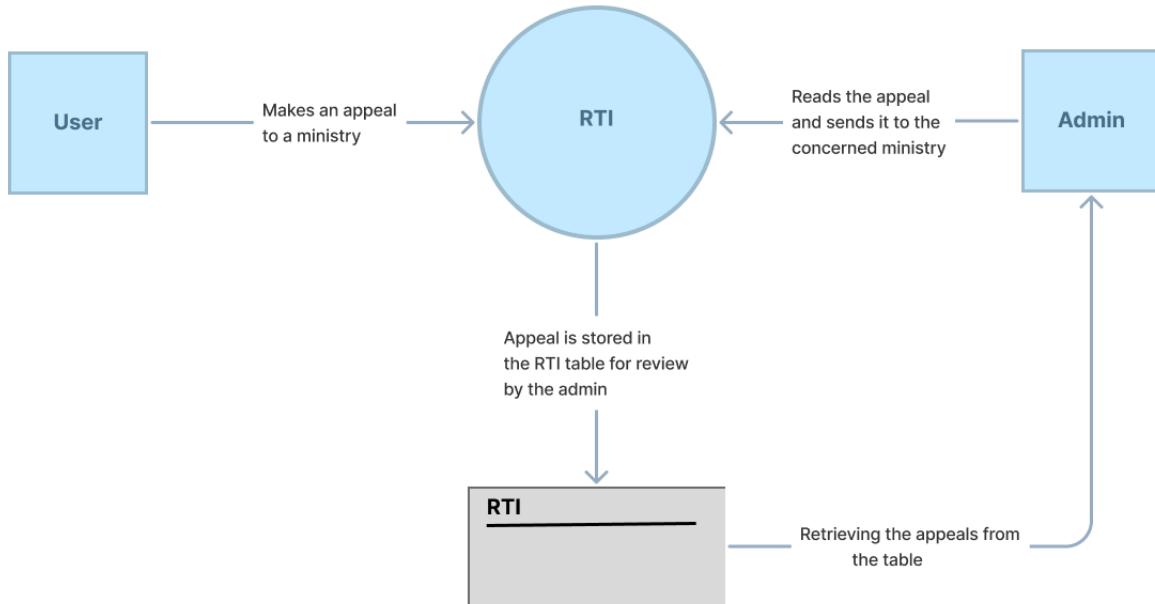


Figure 92: Level 2 Data Flow Diagram of RTI

RTI: In this module, citizens can make appeals to various ministries, which are reviewed by the admin of the administrative hierarchy module to provide the citizen with the requested information.

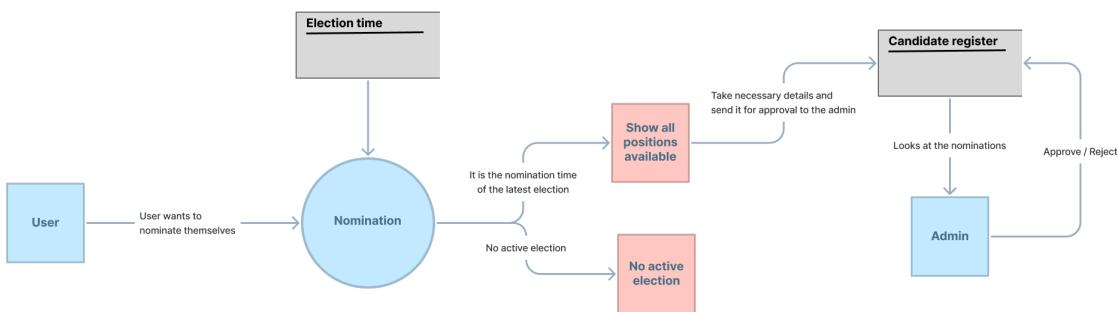


Figure 93: Level 2 Data Flow Diagram of Nomination

Nomination: In this module, citizens can nominate themselves by providing the required

information. But before being able to contest in elections, these nominations are reviewed by the admin and either are rejected or accepted. Once the nomination is approved, the candidate's details are provided to the public.

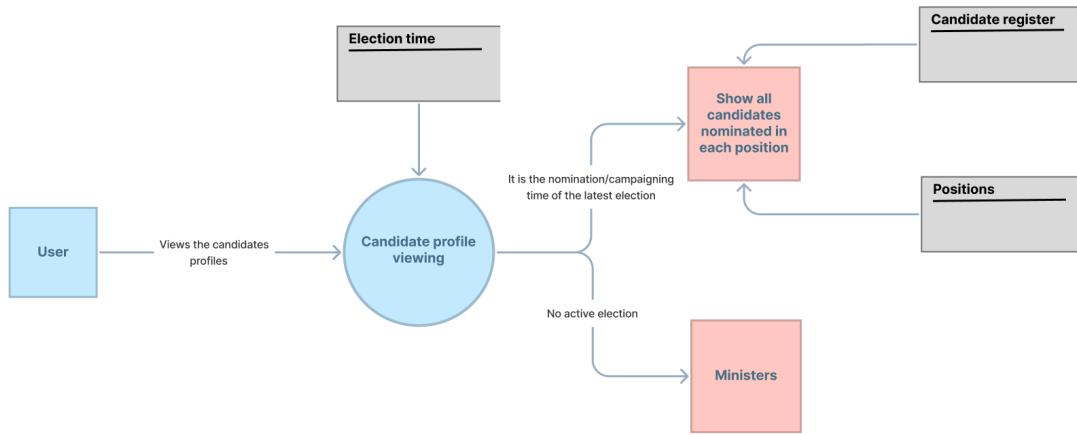


Figure 94: Level 2 Data Flow Diagram of Candidate Profile Viewing

Candidate Profile Viewing: In this module, citizens can view the details of the nominated candidates to take decisions.

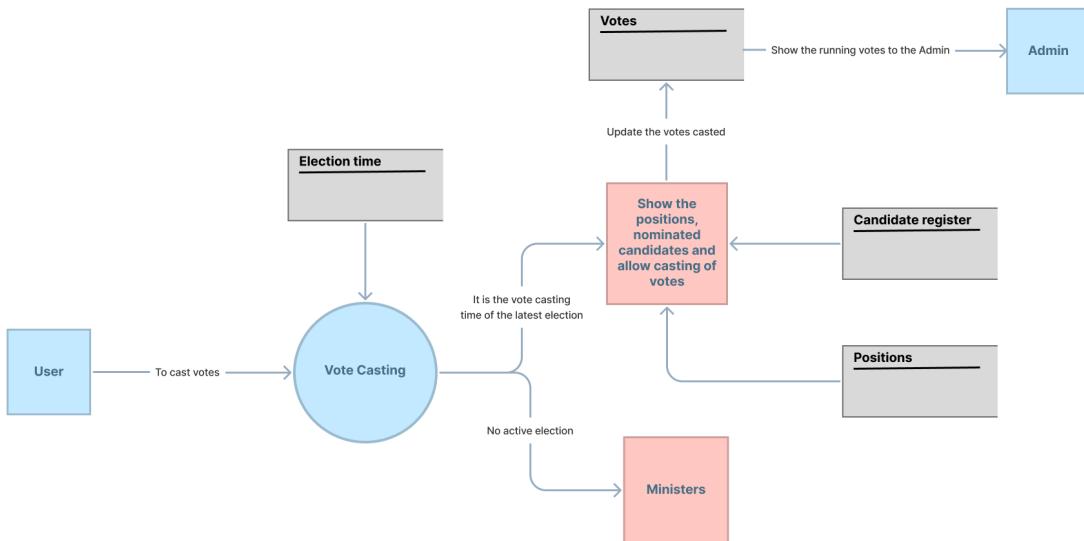


Figure 95: Level 2 Data Flow Diagram of Vote Casting

Vote Casting: Citizens can submit their ballot for all the positions and these votes are updated in the Votes table. Thus, the admin can view the live voting process.

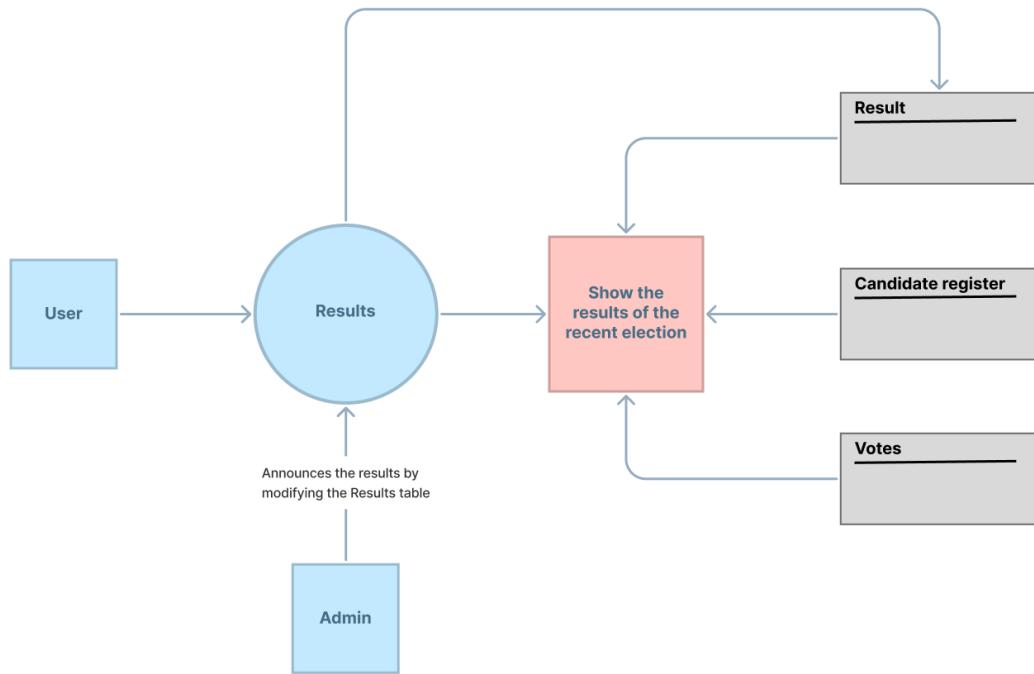


Figure 96: Level 2 Data Flow Diagram of Results

Results: The admin announces the results and the citizens can view them.

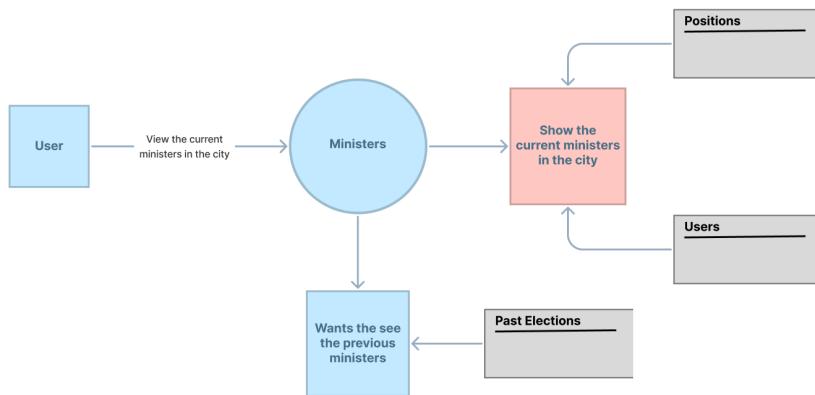


Figure 97: Level 2 Data Flow Diagram of Ministers

Ministers: Citizens can view the list of ministers of the Smart City.

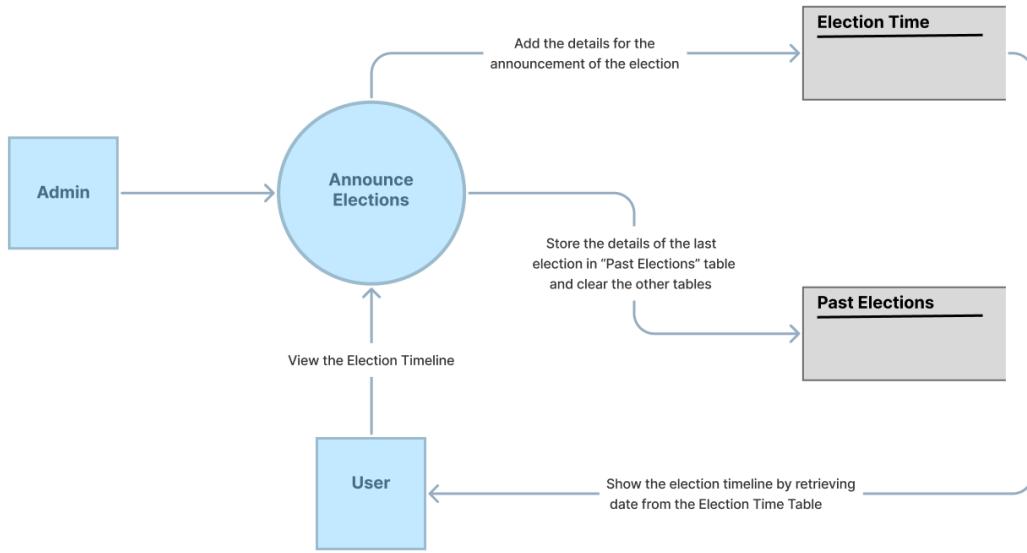


Figure 98: Level 2 Data Flow Diagram of Announce Elections

Announce Elections: The admin can schedule elections and make an announcement. Citizens can view this schedule.



Figure 99: Level 2 Data Flow Diagram of Violations

Violations: The citizens can report a candidate for the violation of code of conduct. The admin will have a look at these reports and can take the necessary action.

5.9.5 Interface

Given below are the images of the basic interface created for the administrative hierarchy module.



Figure 100: Home Page of Administrative Hierarchy Module



Figure 101: Shows the Organizational structure of the Smart City to the citizens

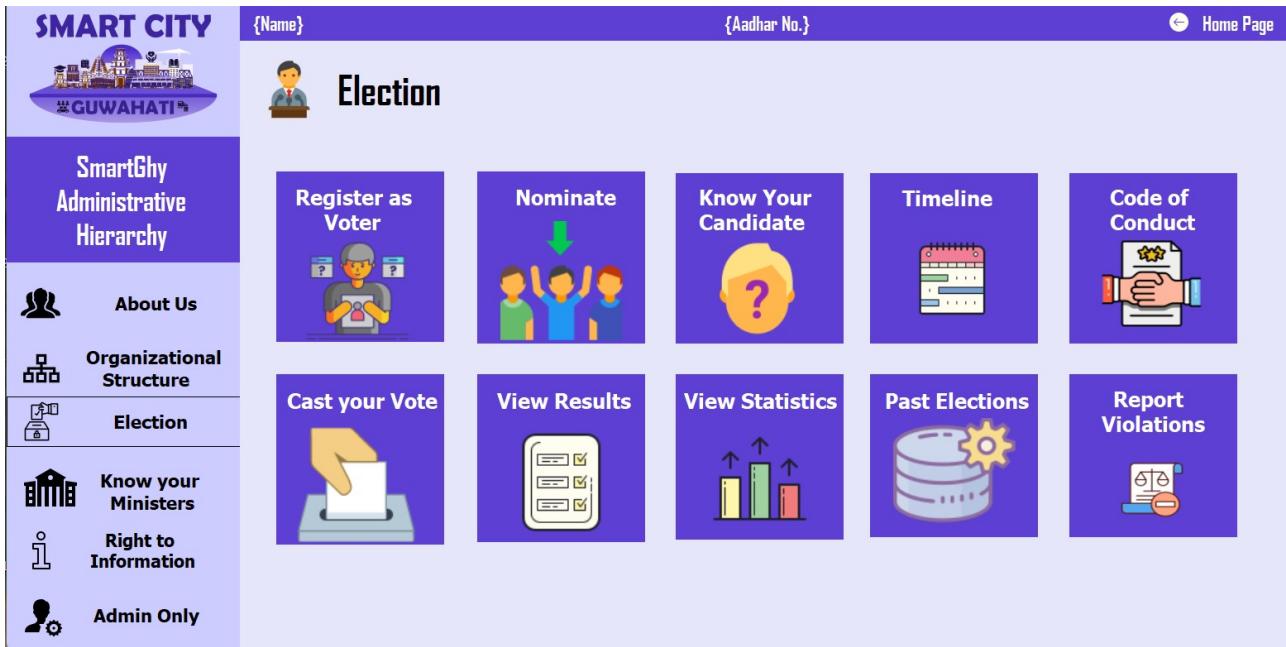


Figure 102: Citizen Menu

The 'Register as Voter' form interface. The top navigation bar includes fields for {Name} and {Aadhar No.}, and a Home Page button. The main content area is titled 'Register as Voter' and features a placeholder for a 'Profile Picture'. To the right, there are five input fields for personal information:

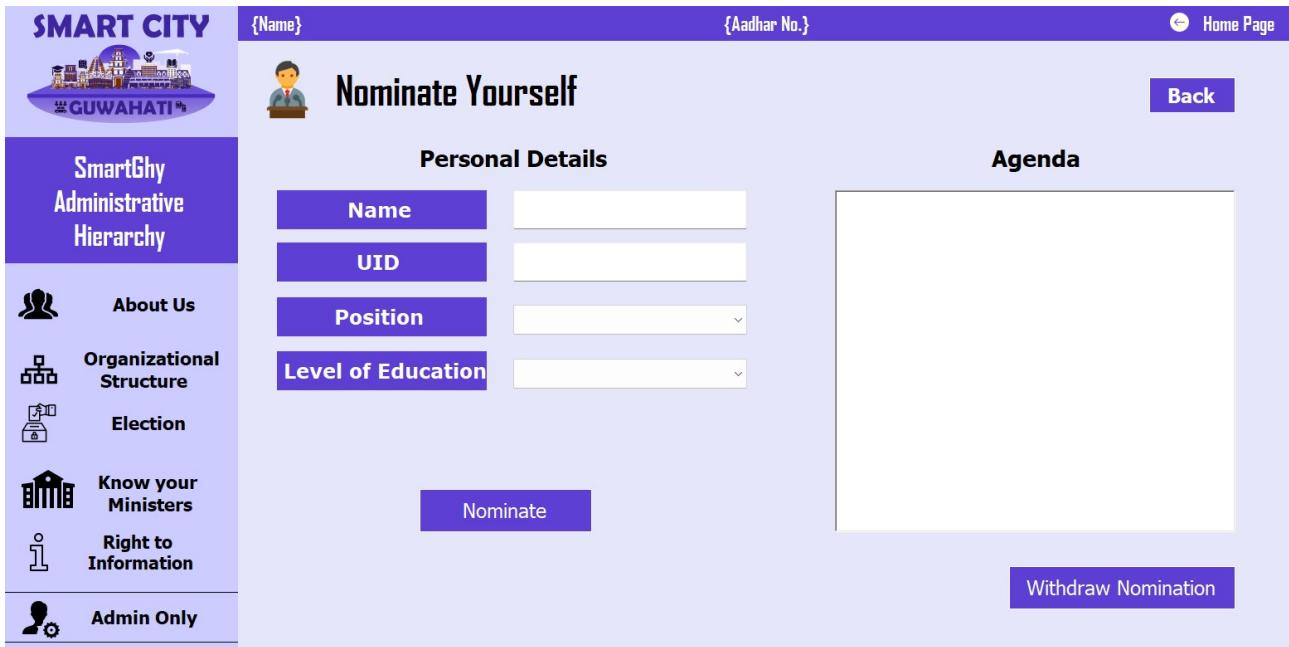
| | |
|--------------|----------------------|
| Name | <input type="text"/> |
| UID | <input type="text"/> |
| Age | <input type="text"/> |
| House Number | <input type="text"/> |
| Ward Number | <input type="text"/> |

A checkbox labeled 'All the above information is true to my best knowledge.' is located below the input fields. A 'Register' button is at the bottom right.

The left sidebar lists various menu items with icons:

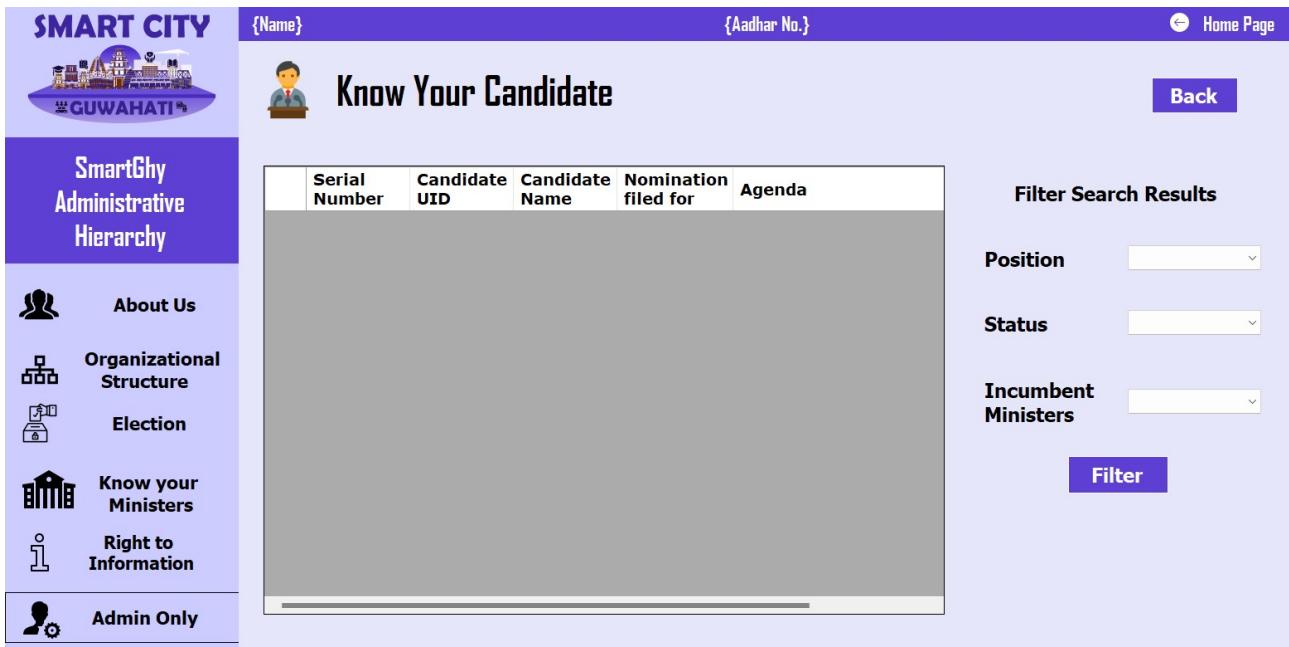
- SmartGhy Administrative Hierarchy**
- About Us**
- Organizational Structure**
- Election**
- Know your Ministers**
- Right to Information**
- Admin Only**

Figure 103: Register as Voter



The screenshot shows the 'Nominate Yourself' section of the SmartGhy administrative interface. At the top, there are input fields for '{Name}' and '{Aadhar No.}' with a 'Home Page' link. Below this, a title 'Nominate Yourself' is displayed next to a small profile icon. A 'Back' button is located in the top right corner. The main area is titled 'Personal Details' and contains four input fields: 'Name', 'UID', 'Position', and 'Level of Education'. To the right of these fields is a large, empty rectangular box labeled 'Agenda'. At the bottom center is a blue 'Nominate' button, and at the bottom right is a blue 'Withdraw Nomination' button.

Figure 104: Citizen can nominate to contest elections



The screenshot shows the 'Know Your Candidate' section of the SmartGhy administrative interface. At the top, there are input fields for '{Name}' and '{Aadhar No.}' with a 'Home Page' link. Below this, a title 'Know Your Candidate' is displayed next to a small profile icon. A 'Back' button is located in the top right corner. The main area features a table with columns: Serial Number, Candidate UID, Candidate Name, Nomination filed for, and Agenda. The table body is currently empty. To the right of the table is a 'Filter Search Results' section containing three dropdown menus: 'Position', 'Status', and 'Incumbent Ministers', followed by a 'Filter' button.

Figure 105: Know Your Candidate: Citizens can view the details of the candidates contesting in elections.

| Events | Dates |
|---------------------------|------------|
| Nomination Start Date | 18.03.2024 |
| Nomination End Date | 20.03.2024 |
| Campaigning Start Date | 22.03.2024 |
| Campaigning End Date | 25.03.2024 |
| Election Date | 27.03.2024 |
| Results Announcement Date | 30.03.2024 |

Figure 106: Timeline: Citizens can view the election timeline

Election Code of Conduct

Preamble:
The following code of conduct is established to ensure fairness, transparency, and integrity in all aspects of the electoral process. It aims to uphold democratic principles, protect the rights of voters, candidates, and electoral officials, and foster an environment conducive to free and fair elections.

- 1. Respect for Democratic Principles:**
 - a. All parties, candidates, and stakeholders shall uphold the principles of democracy, including the rule of law, respect for human rights, and adherence to electoral regulations.
 - b. Every effort shall be made to ensure that elections are conducted in a free, fair, and transparent manner, and that the will of the electorate is respected.
- 2. Non-Discrimination:**
 - a. No individual or group shall be discriminated against on the basis of race, ethnicity, religion, gender, disability, sexual orientation, or any other characteristic.
 - b. All candidates and parties shall campaign and conduct themselves in a manner that promotes inclusivity and diversity.

Figure 107: Code of Conduct: Citizens can view the Code of Conduct of the election



SMART CITY
GUWAHATI

SmartGhy Administrative Hierarchy

- About Us**
- Organizational Structure**
- Election**
- Know your Ministers**
- Right to Information**
- Admin Only**

{Name} {Aadhar No.} [Home Page](#)

Cast your Vote

[Back](#)

| | | | |
|-------------------|----------------------|---------------------|----------------------|
| Employment | <input type="text"/> | Power | <input type="text"/> |
| Education | <input type="text"/> | Health | <input type="text"/> |
| Culture | <input type="text"/> | Broadcasting | <input type="text"/> |
| Finance | <input type="text"/> | IT | <input type="text"/> |
| Transport | <input type="text"/> | | |

[Submit Ballot](#)

Figure 108: Cast your Vote: Citizens can submit their ballot here for all positions



SMART CITY
GUWAHATI

SmartGhy Administrative Hierarchy

- About Us**
- Organizational Structure**
- Election**
- Know your Ministers**
- Right to Information**
- Admin Only**

{Name} {Aadhar No.} [Home Page](#)

View Results

[Back](#)

| Position | Elected Candidate Name |
|--------------|------------------------|
| Employment | Neha Sharma |
| Education | Rajesh Patel |
| Health | Priya Singh |
| Transport | Arjun Kumar |
| Culture | Pooja Mehta |
| Power | Sanjay Gupta |
| Finance | Anjali Dubey |
| Broadcasting | Vikram Chauhan |
| IT | Ritu Verma |

Figure 109: View Results: Citizens can view the results of the recent election

The screenshot shows the 'Past Elections' section of the SmartGhy website. The top navigation bar includes fields for Name and Aadhar No., and a Home Page link. On the left, a sidebar menu lists 'About Us', 'Organizational Structure', 'Election', 'Know your Ministers', 'Right to Information', and 'Admin Only'. The main content area displays a table titled 'Past Elections' with columns for Position, Candidate Name, Votes Received, and Turnout for Position. All entries in the table are placeholder text ('DummyVal'). A 'Year' dropdown menu is also present.

| Position | Candidate Name | Votes Received | Turnout for Position |
|----------|----------------|----------------|----------------------|
| DummyVal | DummyVal | DummyVal | |

Figure 110: Past Elections: Citizens can view the details of the past elections

The screenshot shows the 'Report Violation' section of the SmartGhy website. The top navigation bar includes fields for Name and Aadhar No., and a Home Page link. On the left, a sidebar menu lists 'About Us', 'Organizational Structure', 'Election', 'Know your Ministers', 'Right to Information', and 'Admin Only'. The main content area displays a form titled 'Report Violation' with a 'Candidate' dropdown menu. A large empty text area is available for reporting details, and a 'Report Violation' button is at the bottom right.

Figure 111: Violation: Citizens can report candidates for violation of Code of Conduct

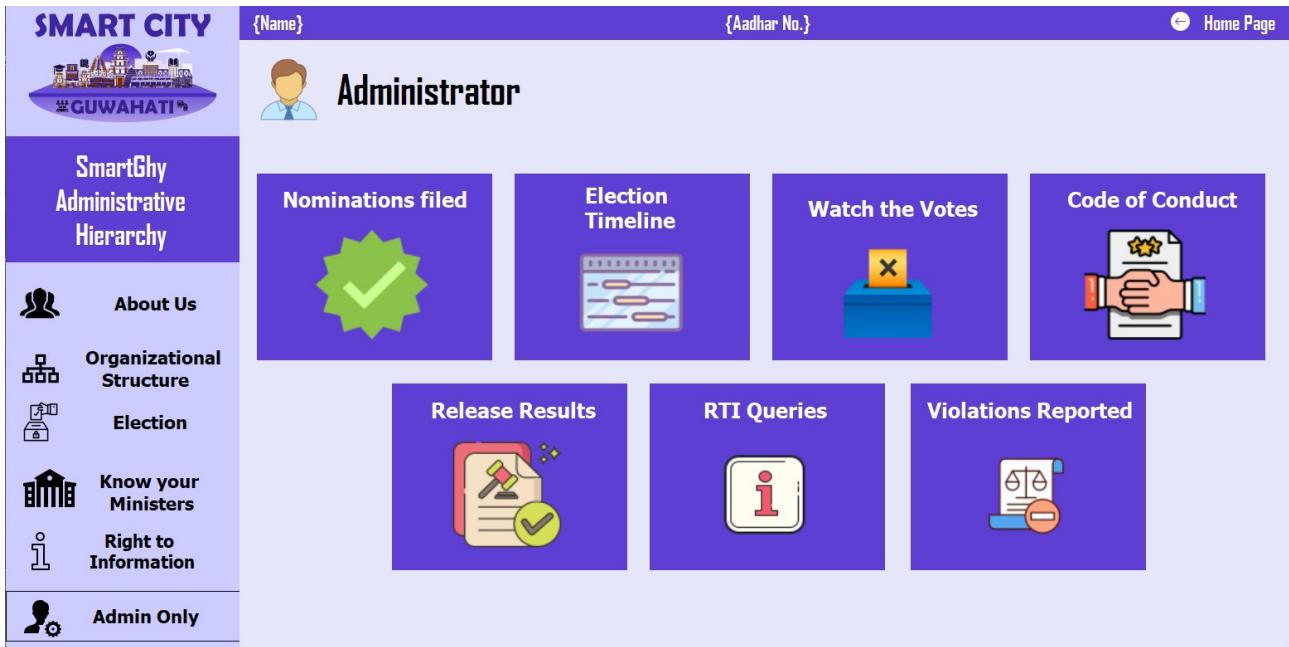


Figure 112: Admin Menu

The screenshot shows the "Nominations Filed" page. The top navigation bar includes fields for {Name} and {Aadhar No.}, and a Home Page link. The main content area is titled "Nominations Filed". It features a table with nomination details and a filter search results section:

| Serial Number | Candidate UID | Candidate Name | Nomination filed for | Agenda |
|---------------|---------------|----------------|----------------------|--|
| 1 | 121 | Neha Sharma | Power Minister | Implementing renewable energy initiatives to reduce reliance on fossil fuels and promote sustainability. Modernizing and upgrading existing power infrastructure to ensure reliable and efficient electricity supply for citizens. |
| 2 | 256 | Rajesh Patel | Finance Minister | Implementing fiscal policies to stimulate economic growth and creation while maintaining fiscal discipline. Enhancing financial inclusion and |

Filter Search Results

Position:

Status:

Incumbent Ministers:

Filter

Figure 113: Nominations Filed: Admin can view the nominations filed and can approve or reject the nomination

Figure 114: Election Timeline: Admin can schedule the election and announce it

Figure 115: Watch the votes: Admin can watch the votes during the election

The screenshot shows the 'Release Results' section of the SmartGhy administrative interface. At the top, there are input fields for '{Name}' and '{Aadhar No.}' with a 'Home Page' link. Below this is a title 'Release Results' with a 'Back' button. A message states: 'Results of the latest election are given in the table below.' A table displays the results of the latest election with columns for Position, Candidate Name, Votes Received, and Turnout for Position. All entries in the table are 'DummyVal'. On the left sidebar, there are links for 'About Us', 'Organizational Structure', 'Election', 'Know your Ministers', 'Right to Information', and 'Admin Only'. A 'Release Results' button is located at the bottom right of the main content area.

| Position | Candidate Name | Votes Received | Turnout for Position |
|----------|----------------|----------------|----------------------|
| DummyVal | DummyVal | DummyVal | |

Figure 116: Release Results: Once the voting time is over, the admin will receive the list of elected candidates and can release the results

The screenshot shows the 'Right To Information' section of the SmartGhy administrative interface. At the top, there are input fields for '{Name}' and '{Aadhar No.}' with a 'Home Page' link. Below this is a title 'Right To Information' with a 'Past Appeals' button. A 'Appeal to Ministry' button is visible above a large empty rectangular area. On the left sidebar, there are links for 'About Us', 'Organizational Structure', 'Election', 'Know your Ministers', 'Right to Information', and 'Admin Only'.

Figure 117: RTI: Citizens can make appeals to various ministries

| | Ministry | Query | Status | Response |
|---|----------|-------|--------|----------|
| * | | | | |

Figure 118: Past Appeals: Citizens can also track the status of their previous appeals

| | Serial Number | Candidate UID | Ministry | Query | Status | Response |
|---|---------------|---------------|----------|-------|--------|----------|
| * | | | | | | |

Figure 119: RTI Admin: Admin can review the appeals made by the citizens

5.10 Library

5.10.1 Introduction

Welcome to the central online portal for Guwahati's library system, where knowledge meets convenience at your fingertips. Administered with precision and care, our platform offers a

diverse range of digital resources, from e-books and academic journals to multimedia collections, catering to the intellectual pursuits of students, researchers, and enthusiasts alike. Join us in unlocking the wealth of information and fostering a culture of learning in the vibrant city of Guwahati.

5.10.2 Features

The Central Library Portal has a variety of features which makes it a modern library management system. The features can be explained user-wise namely, Admin and User.

1. Admin:

- (a) **Dashboard:** In the dashboard, We get an overview of key statistics related to the library system such as Total Books in the library, Borrowed Books, Overdue Books, fines due and fines collected.

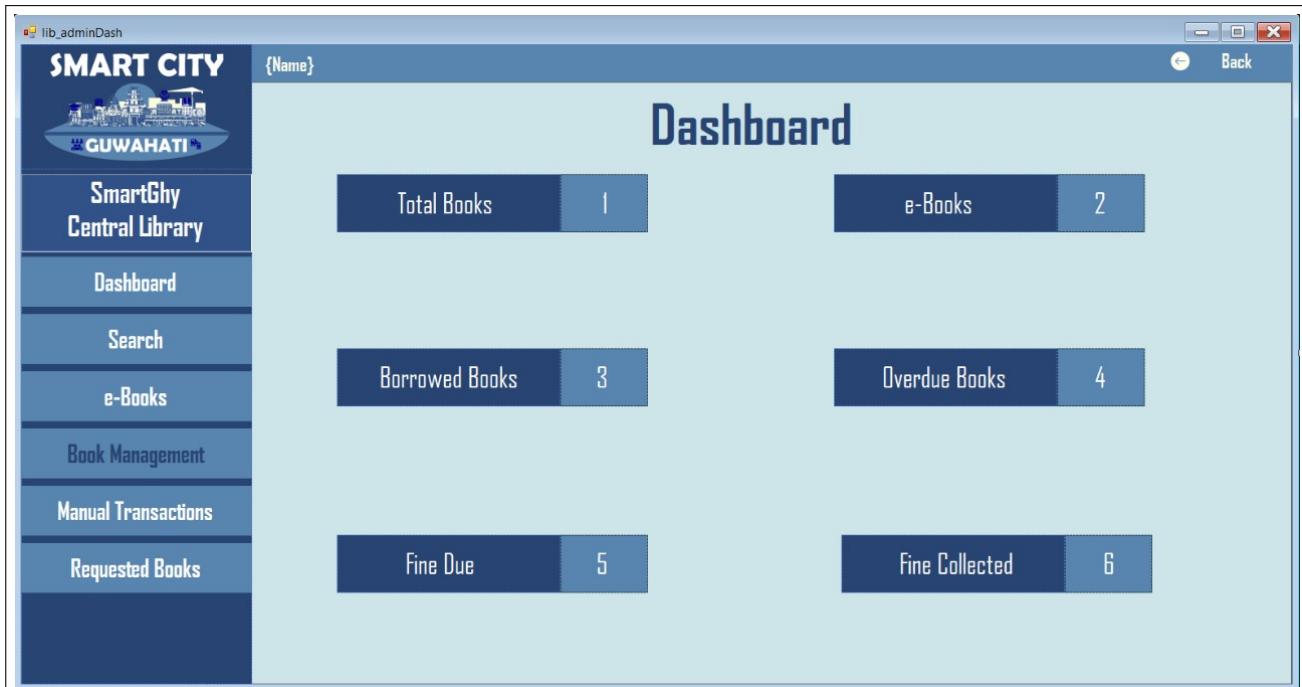


Figure 120: Admin Dashboard: Upon Successful login of the Admin, this inner screen enables the Admin to view his service history

- (b) **Requested Books:** This screen Shows all the requested Books that are requested by users to get issued by the admin and out of these admin can check the status of whether it was issued or not before.

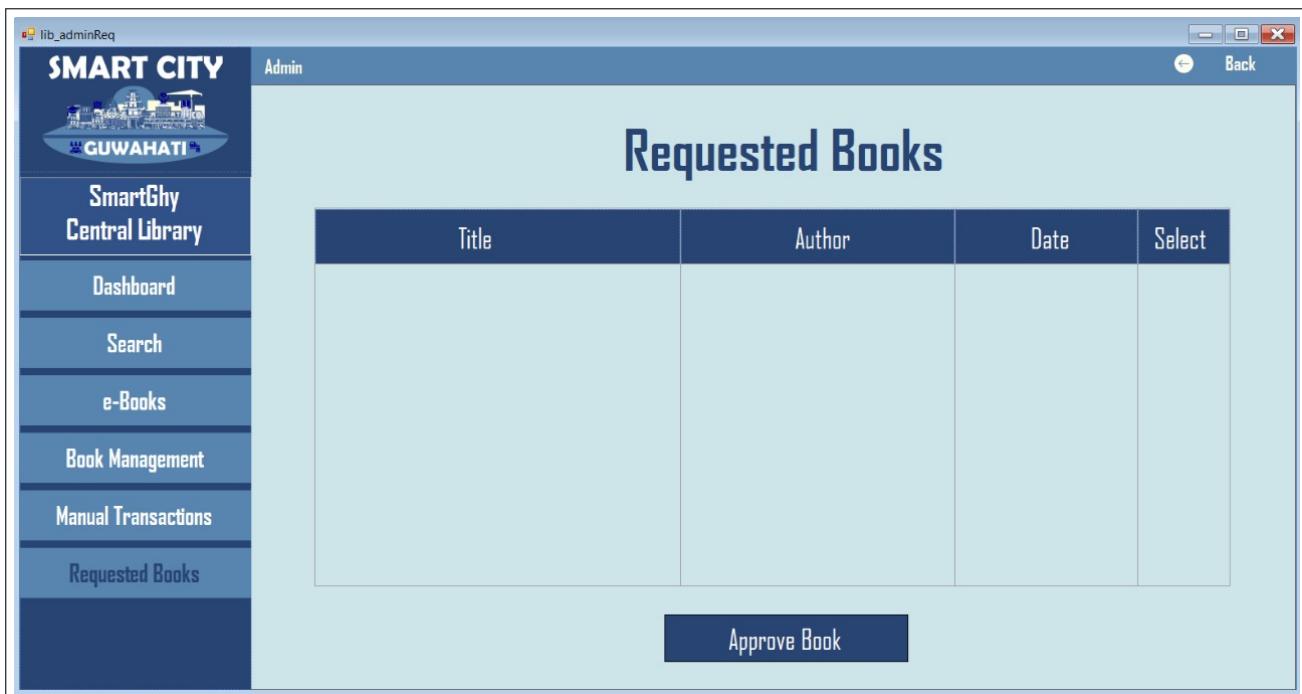


Figure 121: This screen Shows all the requested Books

- (c) **Book Management:** Admin can add a book Adds a new book to the library database using the book details provided. Admin can update a book that modifies the details of an existing book with a specified book ID in the database. Admin can remove a book with a specified book ID from the library database.

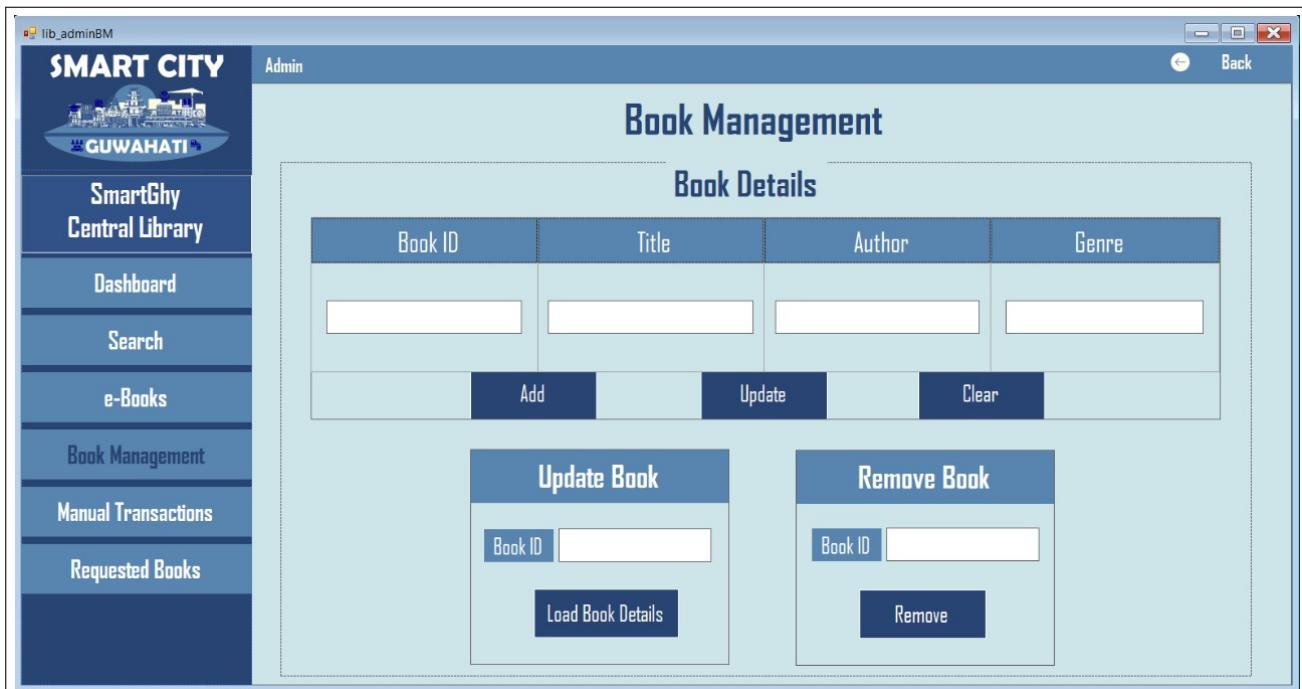


Figure 122: This Screen is used for managing Books via updating and Removing the Books from the catalogue by the admin.

- (d) **Manual Transaction:** In this window: Admin can issue a book with a specified book ID to a user with a specified user ID. Admin can return a book with a specified book ID from a user with a specified user ID to the library. Admin can renew a book with a specified book ID to a user with the specified ID. Admin can pay a fine of a user. It will be cut from the user's balance. Admin can add money to the balance of a user.

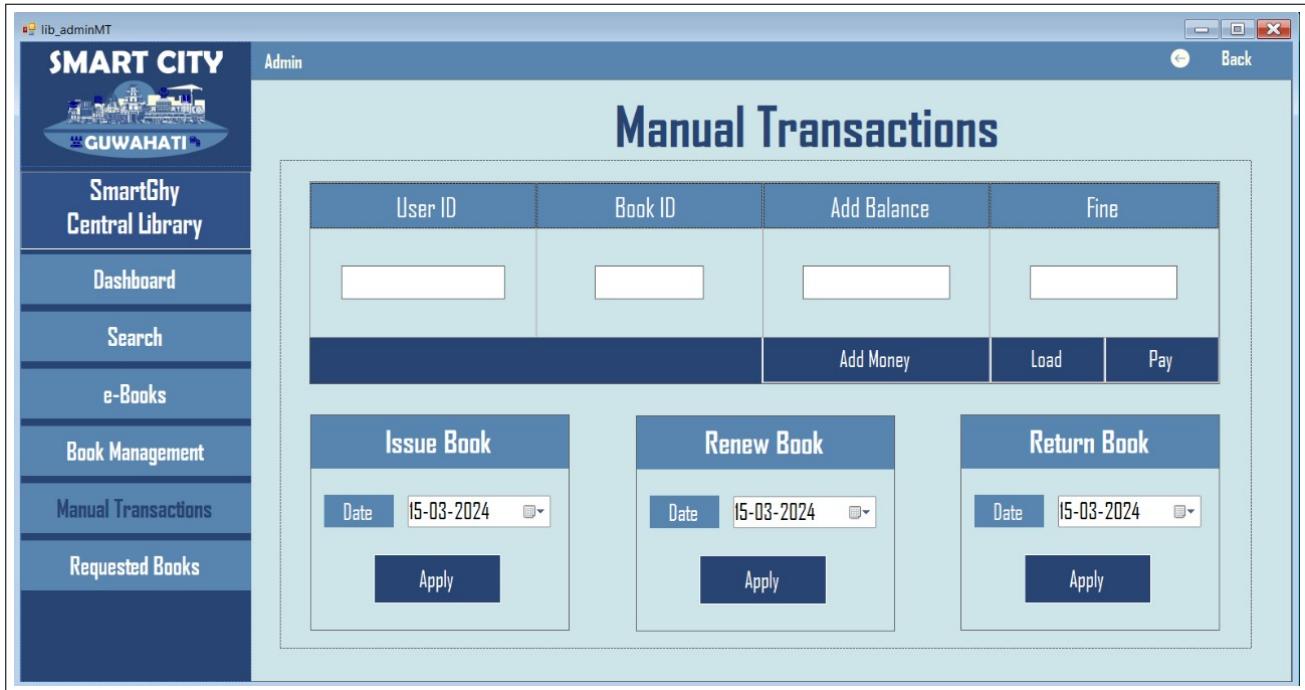


Figure 123: This Figure shows the Manual transaction page of admin along with its features.

2. User:

- (a) **Overdue Books:** This feature allows users to view a list of books that they have borrowed but have not returned within the specified timeframe. It serves as a reminder for users to return borrowed items promptly to avoid late fees or penalties.

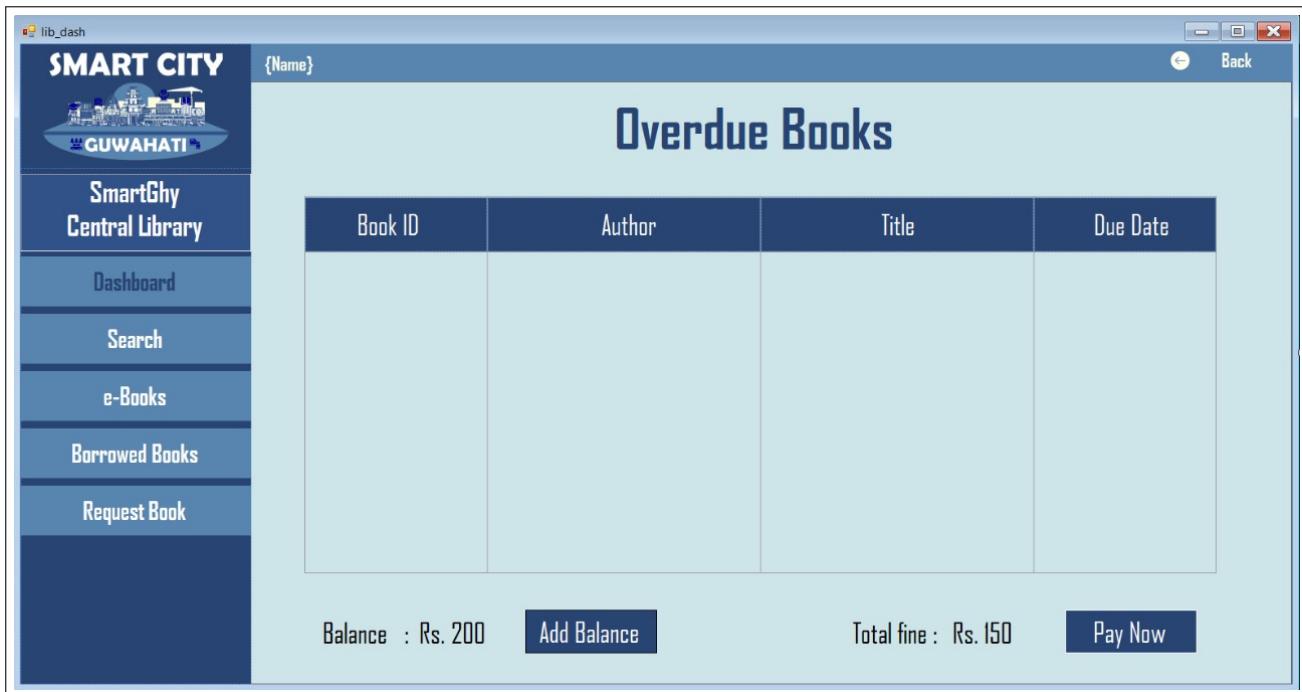


Figure 124: Overdue Books of the Given User.

- (b) **Search Books:** Users can utilize this feature to search for specific books within the library's collection. With a robust search functionality, users can find books by title, author, genre, or keywords, streamlining the process of locating desired reading material.

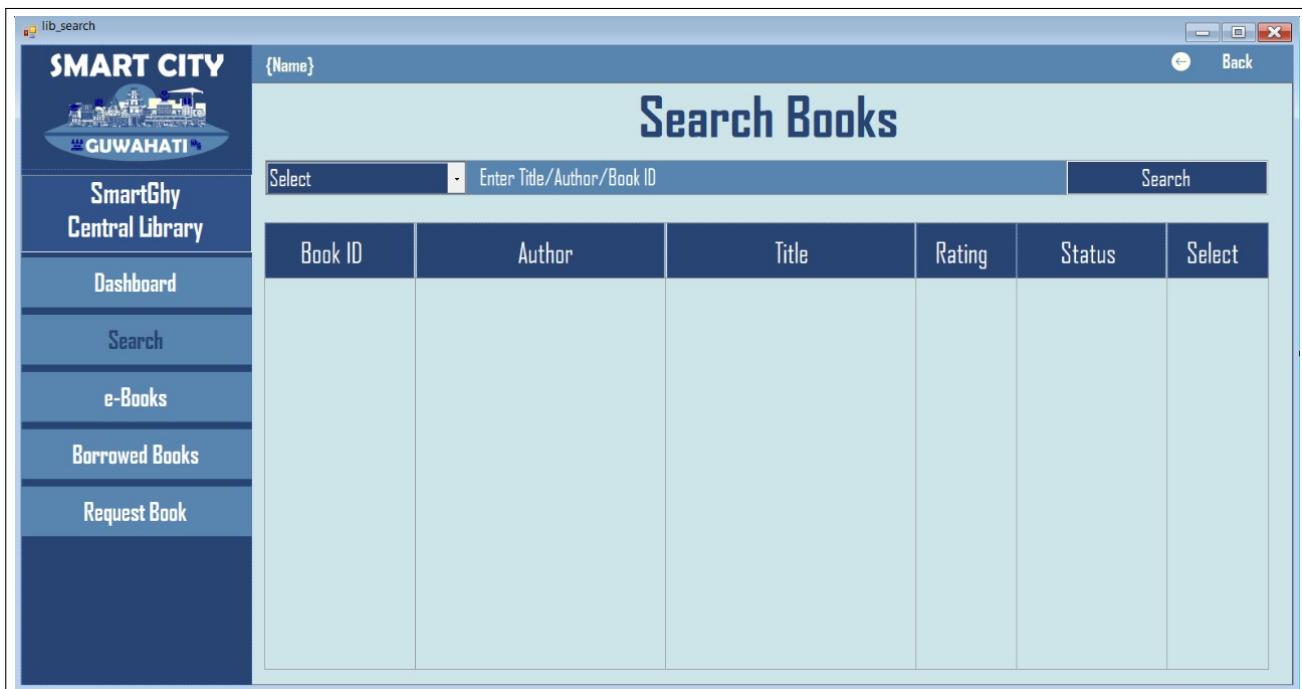


Figure 125: Search Feature for any book used by User

- (c) **E-Books:** Access to a digital library of e-books provides users with the convenience of reading electronic versions of books on various devices. This feature expands access to literature and educational resources beyond the physical constraints of the library, offering flexibility and convenience to users.

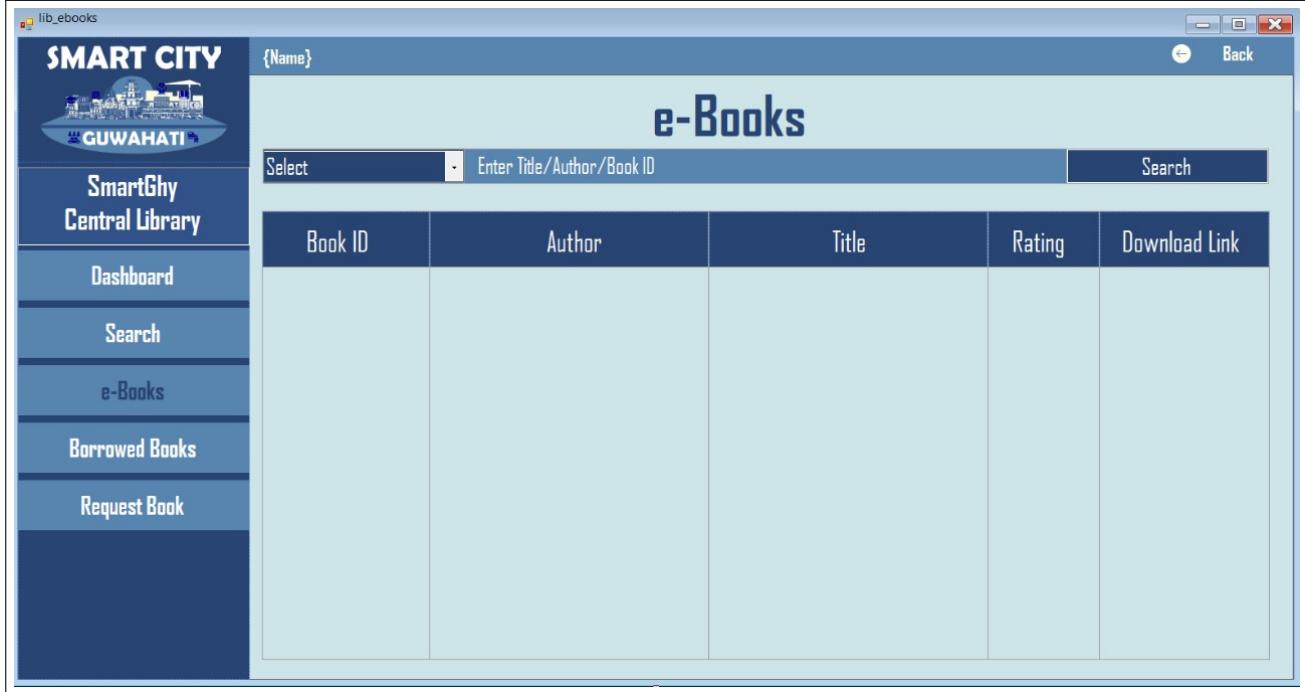


Figure 126: Figure Showing E-Books Feature for the User

- (d) **Borrowed Books:** Users can easily track the books they have borrowed through this feature. It provides information on the titles, due dates, and status of borrowed items, empowering users to manage their borrowing activities efficiently and ensuring timely returns.



Figure 127: List of Books borrowed by the user

- (e) **Requested Books:** Users can request books that are currently unavailable or in high demand through this feature. By submitting a request, users express their interest in specific titles, allowing the library to prioritize acquisitions and fulfil user demands, enhancing the overall user experience.



Figure 128: List of Books Requested by the user

5.10.3 Entity Relationship Diagram

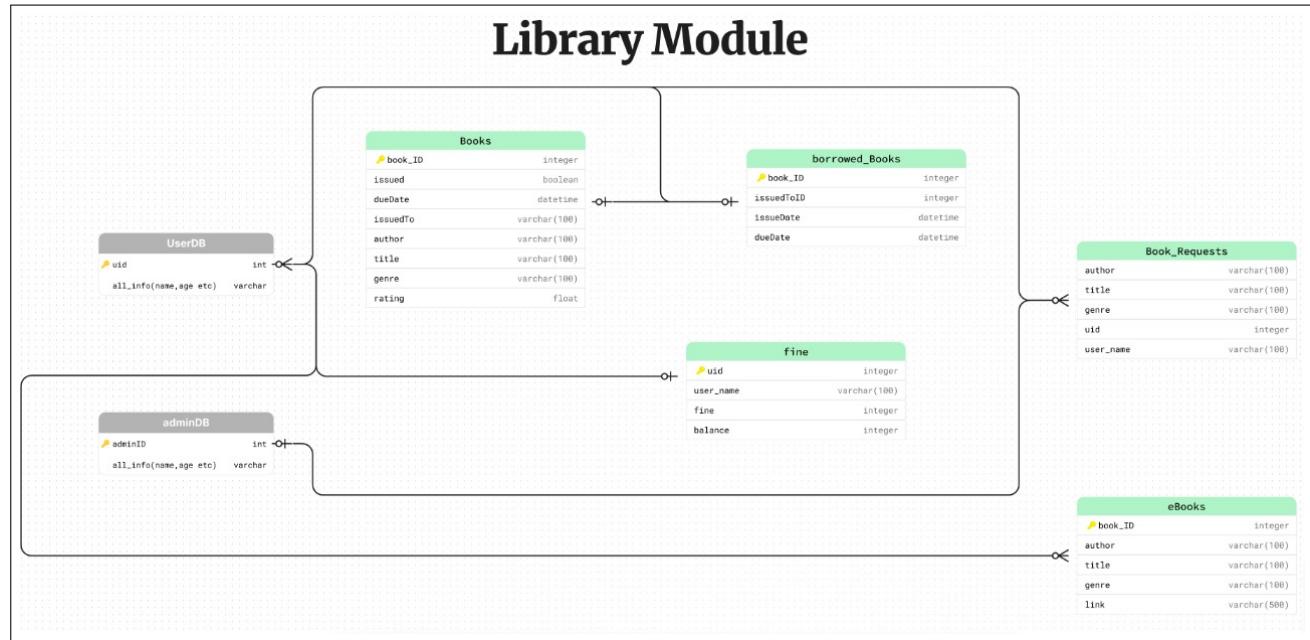


Figure 129: ER Diagram for the Library module

5.10.4 Data Flow Diagram

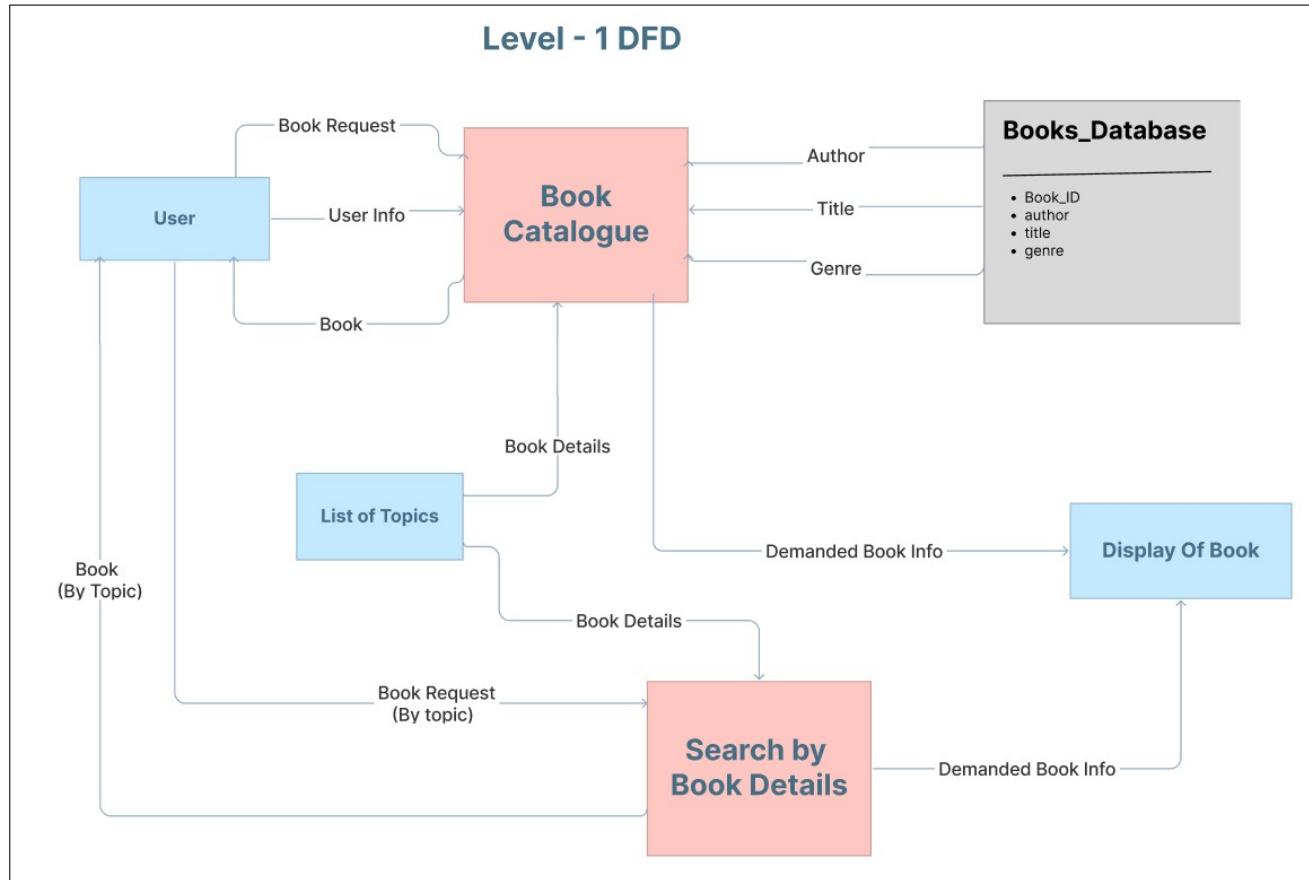


Figure 130: Level-1 DFD of the Library module

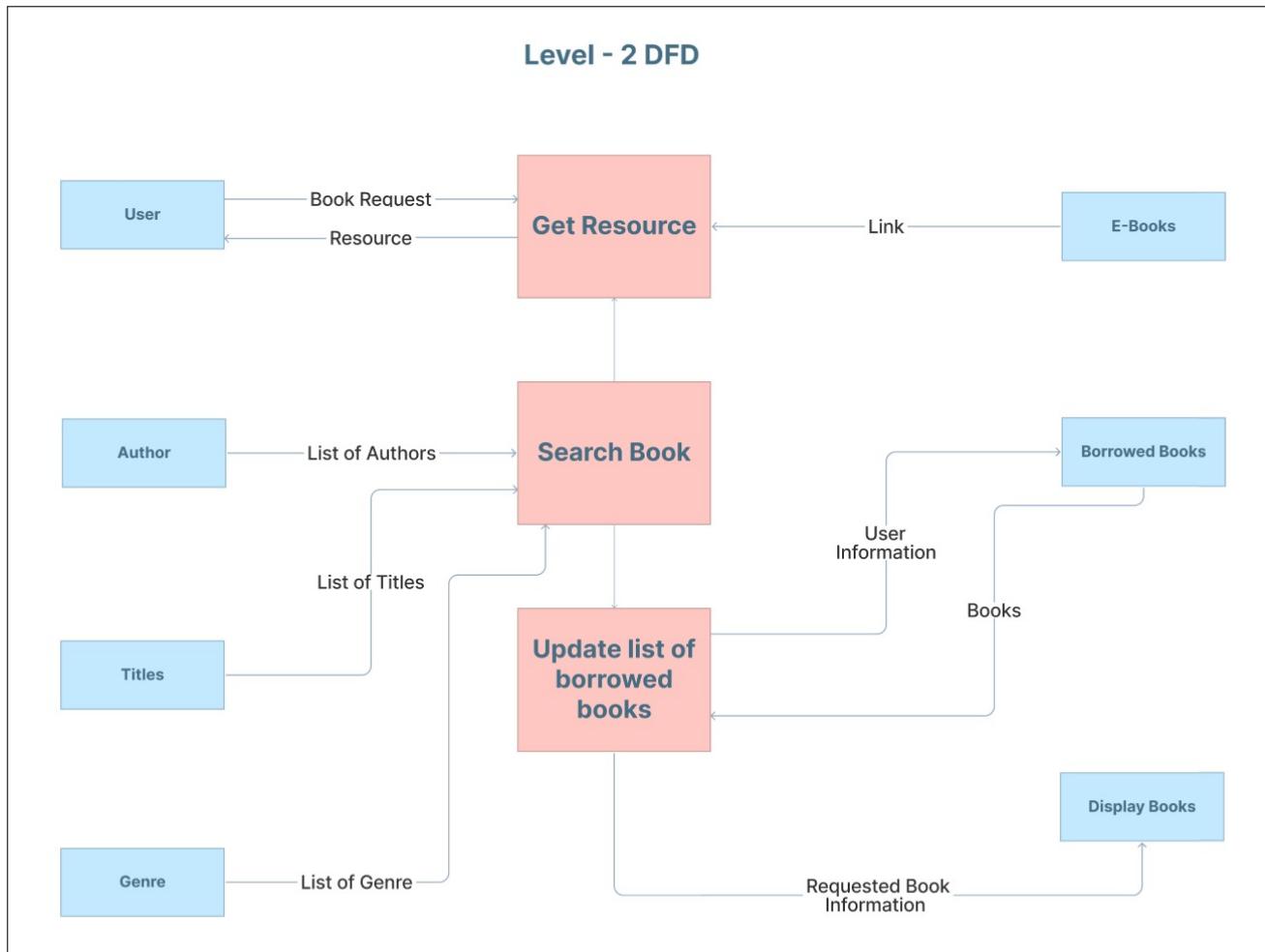


Figure 131: Level-2 DFD of the Library module

5.11 Banking

5.11.1 Introduction

The Banking module within the Smart City Management System offers users a convenient and secure platform to manage their financial activities. From creating a new bank account to accessing account information and performing transactions, this module ensures seamless banking services for users within the smart city ecosystem.

5.11.2 Features

- Account Creation:** New users can create a bank account directly within the Smart City Management app. During the account creation process, users set a separate password to ensure the security of their account. Upon successful creation, users receive a unique account number, and a new record is added to the accounts table in the banking database.
- Account Dashboard:** Users can access their bank account dashboard by logging in through the banking login page. The system verifies the entered credentials against the accounts table in the banking database. Once authenticated, users gain access to various banking functionalities.

3. **Check Balance:** Users can check their account balance directly from the app. The balance information is fetched from the accounts table in the banking database, providing users with real-time visibility into their financial status.
4. **View Transactions:** Users have the option to view all their past transactions from the transaction history. Transaction records are retrieved from the transactions table in the banking database, allowing users to track their financial activities conveniently.
5. **Bill Payments:** Users can make bill payments through the Smart City Management app by entering the recipient's UID. Before processing the payment, users are prompted to re-enter their banking password for authentication. All bill payments are securely recorded in the transactions table for future reference.
6. **Fastag Recharge:** The app facilitates fastag recharge for users' convenience. Fastag recharge functionality allows users to top up their fastag accounts directly from the app, enabling smooth toll payments and travel experiences.

5.11.3 Entity Relationship Diagram

Banking Module

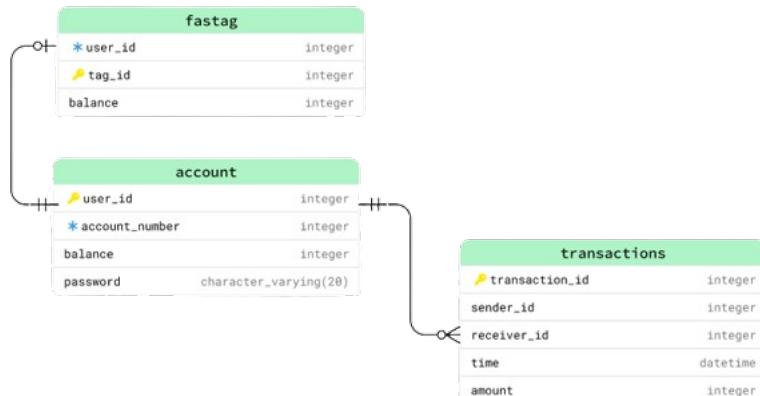


Figure 132: ER Diagram for the banking module

5.11.4 Data Flow Diagram

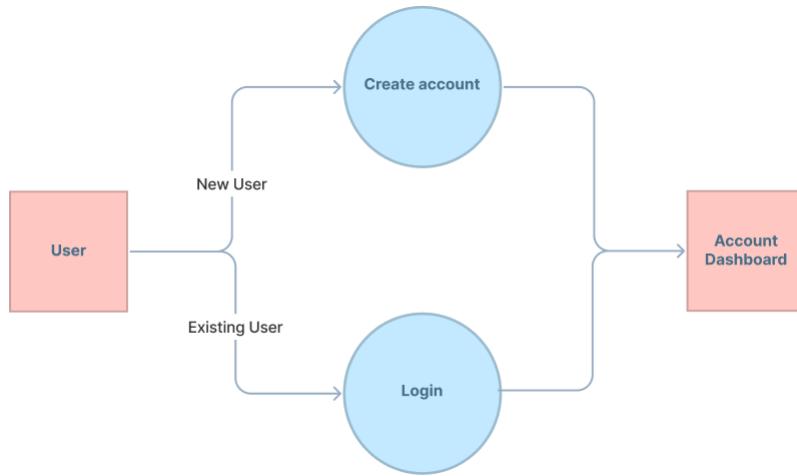


Figure 133: Level1 DFD of the Banking module

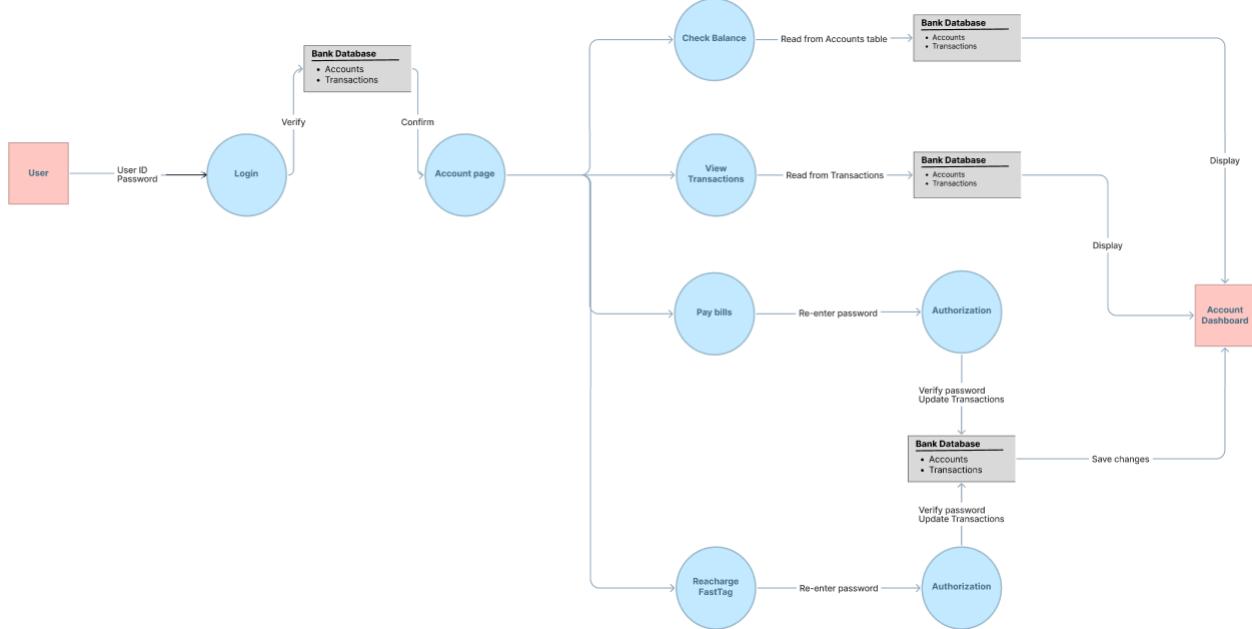


Figure 134: Level 2 DFD of the Banking module

5.11.5 Interface

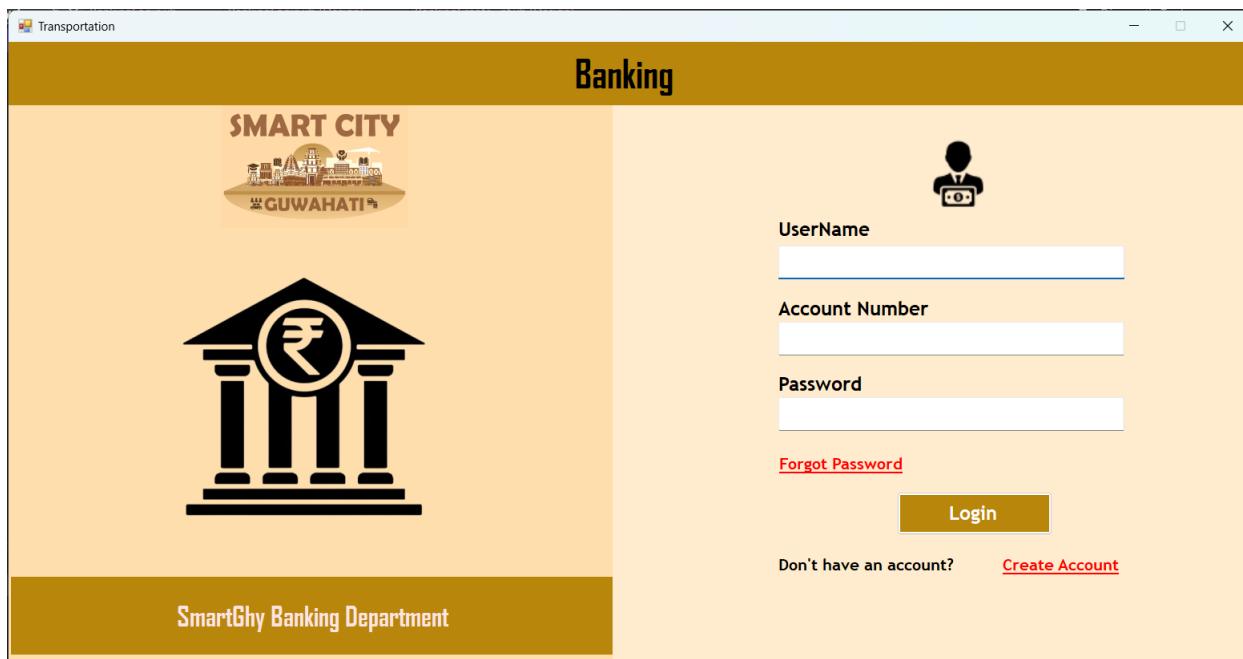


Figure 135: Login page for the banking module

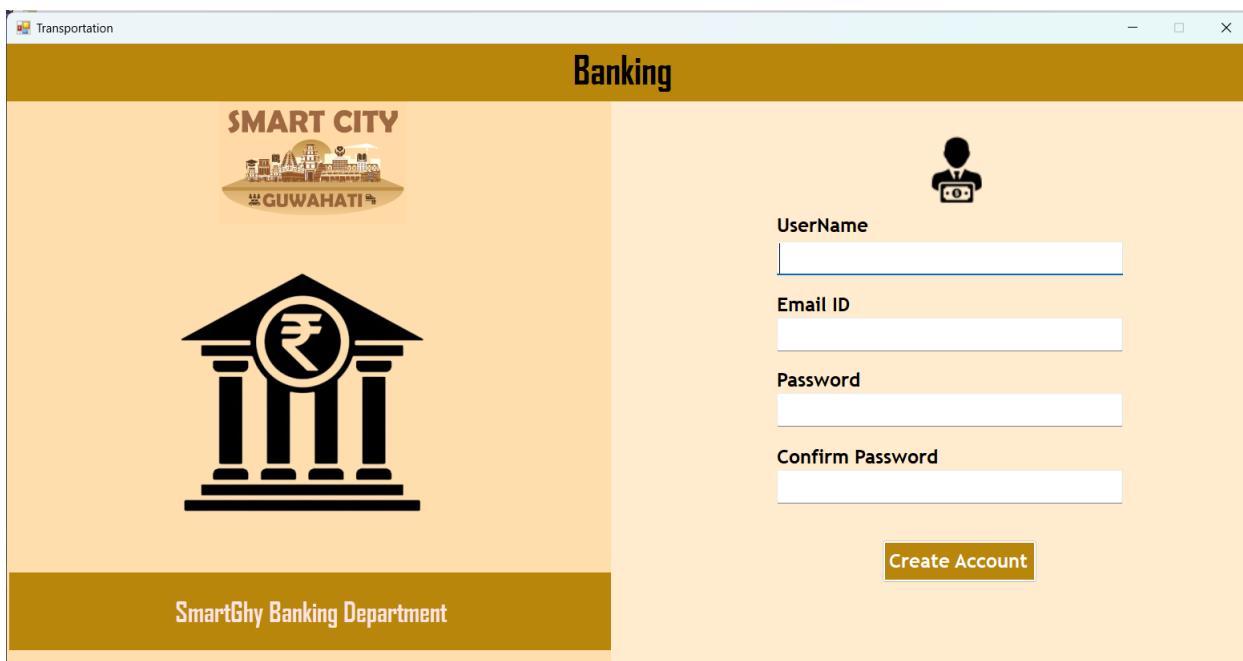


Figure 136: Create account page enables new users to create a bank account

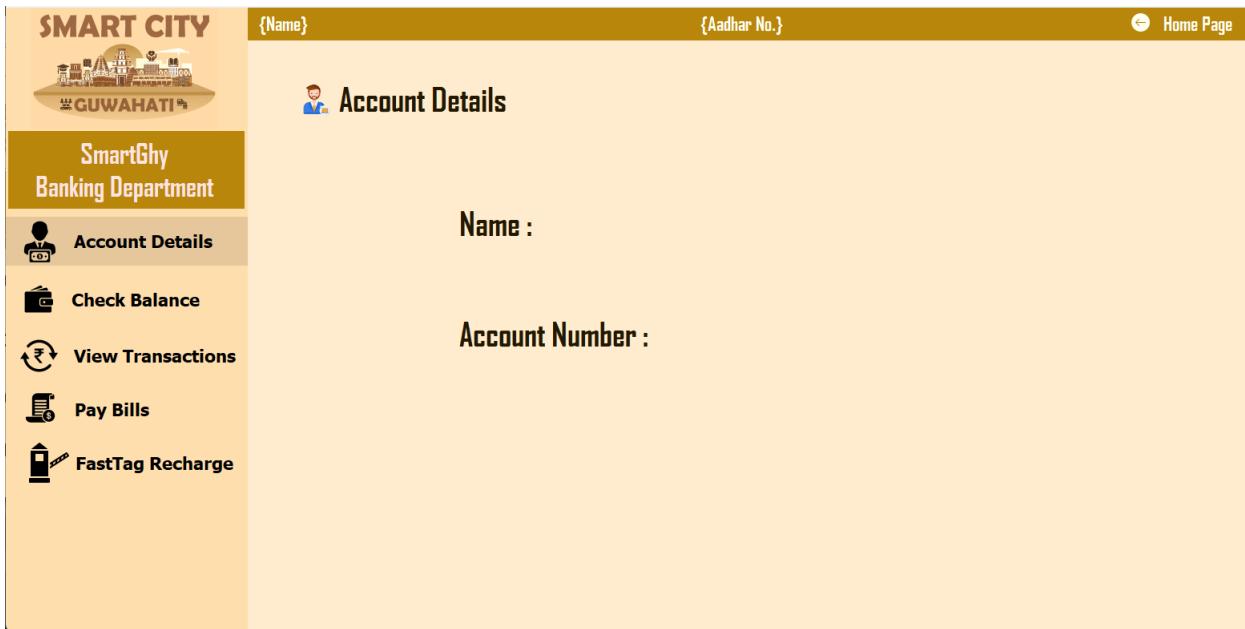


Figure 137: Users can view their account details from the account details page

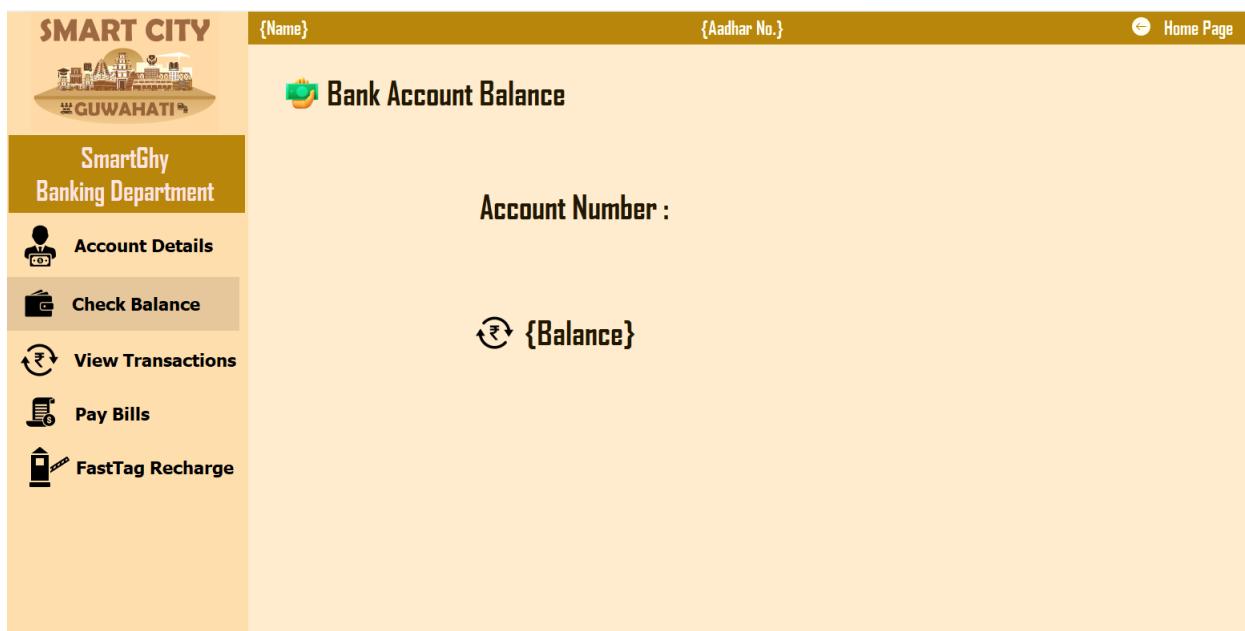


Figure 138: Users can check their account balance using the check balance option

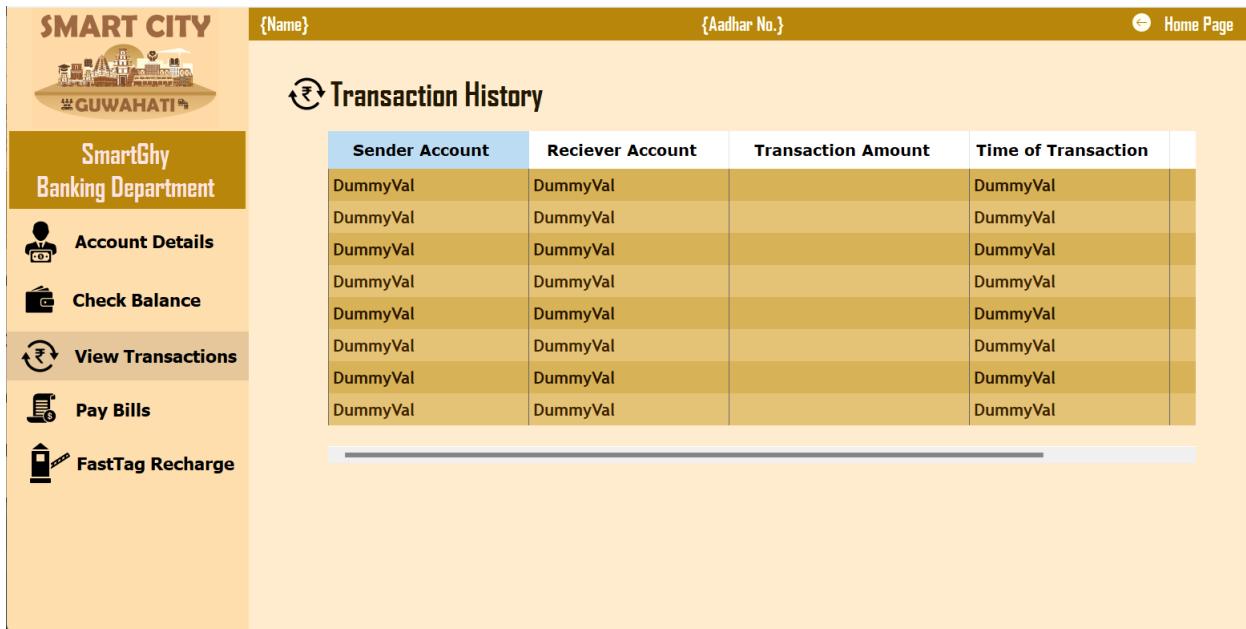


Figure 139: View transactions page allows users to view all their past transactions



Figure 140: Users can make bill payments via the pay bills page

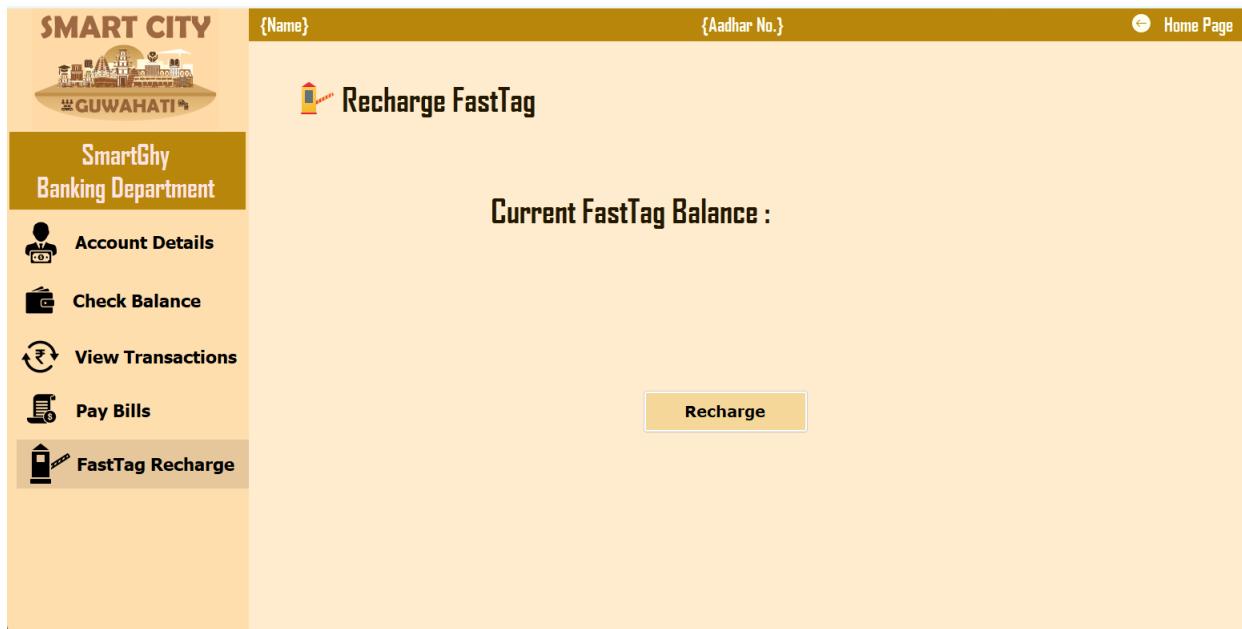


Figure 141: Recharge FastTag page lets users recharge their FastTag

6 Further Improvements

6.1 User Registration and Management

- Allow users to authenticate using external identity providers, such as social media accounts or third-party authentication services like Google or Facebook. This can streamline the registration process and improve user convenience.

6.2 Transportation Management

- Currently, only point to point buses between source and destination are considered for providing the bus schedules due to scope of Visual Basic, but buses for which the source and destination are intermediate bus stops can be considered, which would require application of graph algorithms.
- A portal for Parking Space management, in which cars would be automatically allotted parking space can be made.
- Railway ticket booking system, in which citizens can book both inter-city and local train tickets can be developed.
- A smart-bike management portal can be developed in which the cycles can be tracked in the portal.

6.3 Administrative Hierarchy Interface

- It can interact with the **Festivals & Event Management** module to schedule the election speeches at certain venues.

6.4 Banking

- We can further improve this module by adding the option of loans.
- We can implement features that pay the bills for the services booked, electricity, water, phone and can also implement the tax feature.

6.5 Extra Features:

- **Important Announcement/News:** In this, we will have a module that would be responsible for announcing important activities in the Smart city. It would be the newspaper for the Smart city. Journalists would be publishing their news articles. Admins of each ministry can also publish important announcements. There can be a section for citizen published articles. We can also create a children's magazine as a part of this module.
- **Mail/Chat:** A mail/chat feature can be implemented where citizens can message each other. We can also elevate it by introducing the feature of group chats where citizens can form groups.

7 Conclusion

The Smart City Management Platform consists of several modules that put together creates a software which can be of great help to the citizens in their daily life. Implementing these features and integrating all the modules is going to be a hard task but definitely possible. We aim to create a software that is efficient and easy to use.
