**CITY COUNCIL**

**A PROJECT REPORT**

***Submitted by***

**PARIN ZALA-92200938201**

**SNEH FALDU-92200938211**

**NIHAR TRIVEDI-92200938213**

**JAY RAMANI-92200938214**

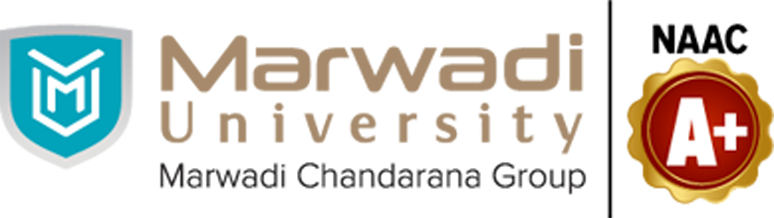
**JYOTIN TANK-92200938241**

***In partial fulfilment for the award of the degree of***

**DIPLOMA ENGINEERING**

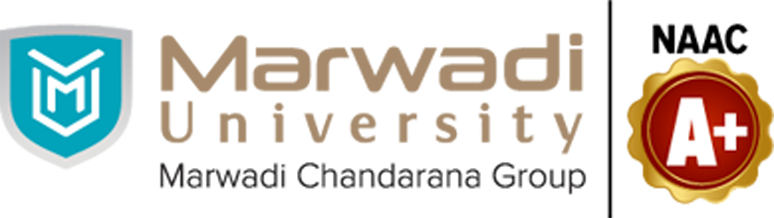
***in***

**Computer Engineering**

****

**Faculty of Diploma Studies**

**Marwadi University, Rajkot**

****

**Marwadi University, Rajkot**

**Faculty of Diploma Studies**

Computer Engineering Department

**2024-25**

**CERTIFICATE**

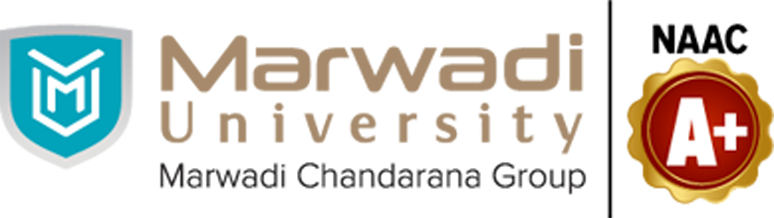
This is to certify that the project entitled **CITY COUNCIL** has been carried out by **PARIN ZALA (92200938201)** under my guidance in partial fulfilment of the degree of Diploma Engineering in Computer Engineering (5th Semester) of Marwadi University, Rajkot during the academic year 2024-25.

Date : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Internal Guide Head of the Department**

Prof. Miral Seladiya Prof. Mittal Joiser

Assistant Professor Computer Engineering

****

**Marwadi University, Rajkot**

**Faculty of Diploma Studies**

Computer Engineering Department

**2024-25**

**CERTIFICATE**

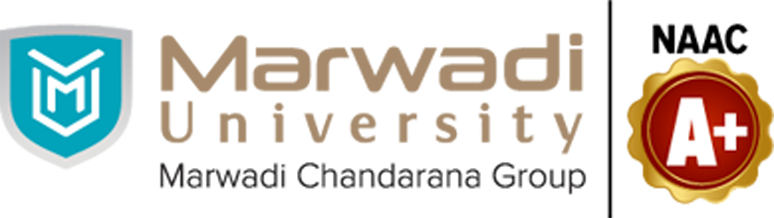
This is to certify that the project entitled **CITY COUNCIL** has been carried out by **SNEH FALDU (92200938211)** under my guidance in partial fulfilment of the degree of Diploma Engineering in Computer Engineering (5th Semester) of Marwadi University, Rajkot during the academic year 2024-25.

Date : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Internal Guide Head of the Department**

Prof. Miral Seladiya Prof. Mittal Joiser

Assistant Professor Computer Engineering

****

**Marwadi University, Rajkot**

**Faculty of Diploma Studies**

Computer Engineering Department

**2024-25**

**CERTIFICATE**

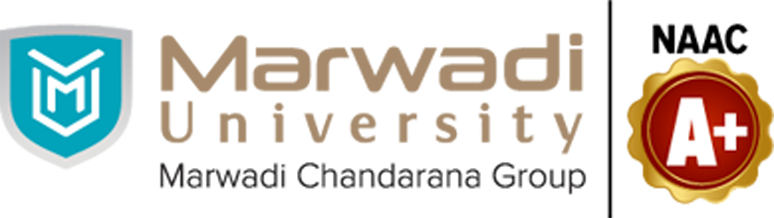
This is to certify that the project entitled **CITY COUNCIL** has been carried out by **NIHAR TRIVEDI (92200938213)** under my guidance in partial fulfilment of the degree of Diploma Engineering in Computer Engineering (5th Semester) of Marwadi University, Rajkot during the academic year 2024-25.

Date : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Internal Guide Head of the Department**

Prof. Miral Seladiya Prof. Mittal Joiser

Assistant Professor Computer Engineering

****

**Marwadi University, Rajkot**

**Faculty of Diploma Studies**

Computer Engineering Department

**2024-25**

**CERTIFICATE**

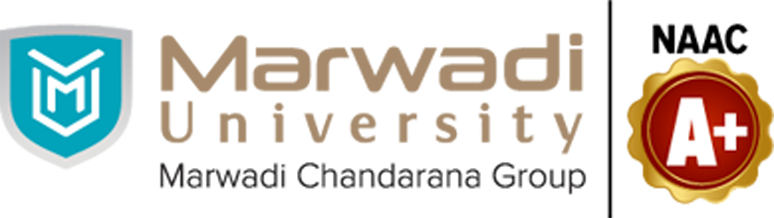
This is to certify that the project entitled **CITY COUNCIL** has been carried out by **JAY RAMANI (92200938214)** under my guidance in partial fulfilment of the degree of Diploma Engineering in Computer Engineering (5th Semester) of Marwadi University, Rajkot during the academic year 2024-25.

Date : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Internal Guide Head of the Department**

Prof. Miral Seladiya Prof. Mittal Joiser

Assistant Professor Computer Engineering

****

**Marwadi University, Rajkot**

**Faculty of Diploma Studies**

Computer Engineering Department

**2024-25**

**CERTIFICATE**

This is to certify that the project entitled **CITY COUNCIL** has been carried out by **JYOTIN TANK (92200938241)** under my guidance in partial fulfilment of the degree of Diploma Engineering in Computer Engineering (5th Semester) of Marwadi University, Rajkot during the academic year 2024-25.

Date : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Internal Guide Head of the Department**

Prof. Miral Seladiya Prof. Mittal Joiser

Assistant Professor Computer Engineering

**Contents**

**Acknowledgements I**

**Abstract II**

**List of Tables III**

**List of Figures IV**

1. **Introduction 16**
   1. Document purpose 16
   2. Product scope 16
   3. Intended audience and document overview 17
   4. Definitions and abbreviations 18
   5. Document conventions 22
   6. References and acknowledgments 24
2. **Overall description 26**
   1. Product perspective 26
   2. Product functionality 26
   3. Users and characteristics 27
   4. Operating environment 28
   5. Design and implementation constraints 29
   6. User documentation 30
   7. Assumptions and dependencies 30
3. **Specific requirements 32**
   1. External interface requirements 32
   2. Functional requirements 37
   3. Behavior requirements 39
4. **Other non-functional requirements 40**
   1. Performance requirements 40
   2. Safety and security requirements 41
   3. Software quality attributes 42
5. **Other requirements 45**

**Appendix A – Data Dictionary 46**

**Acknowledgement**

We would like to sincerely thank Prof. Miral Seladiya, our internal guide, for the continuous support and guidance throughout this project. Your advice and encouragement were essential at every step, and we learned so much from your experience and insights. This project would not have been the same without your help, and We are truly grateful for everything you did.

We also want to express our appreciation to our teammates - Parin Zala, Sneh Faldu, Nihar Trivedi, Jay Ramani, and Jyotin Tank. Working with you all has been a great experience. Each of you brought your own strengths to the team, and together we were able to achieve our goals. Your hard work and dedication played a big role in making this project successful, and We are thankful for the support and cooperation we shared.

We are also thankful to Marwadi University for providing the resources and a positive environment that made our work easier. The tools and facilities available to us were very helpful in completing our project on time and to a good standard.

With Sincere Regards,

Parin Zala.

Sneh Faldu.

Nihar Trivedi.

Jay Ramani.

Jyotin Tank.

I

**Abstract**

The main goal of this project is to create an easy-to-use online platform that helps people access various city services more efficiently. User have to sign up on the website, and after that, they will able to report non-urgent issues, find public health services, and get information about local schools and universities. It also includes features for booking public function halls for events and managing important tasks like property taxes and certificate requests. For further enquiry customer can ask question on chat box. After that customer wants to give feedback according to their experience.

The employee can view a list of all incoming requests submitted by citizens, such as reports of non-emergency issues, service inquiries, and application submissions. Employees can update the status of requests whether a task is Pending, In Progress, or Completed. Employee can view the messages sent by customer and they can reply back to customer.

Admin can create, modify, delete user account for employee. Admin can view and manage all the service request of customer. It can view payment history, hall booking records. They can also view the details of new users who have signed up. The admin can monitor how much time can consume to complete customer request. Admin can view all the feedback given by customer. In project we are going to use HTML for the basic structure, CSS for styling, Bootstrap for responsiveness, PHP for server-side processing, MySQL for database management, and JavaScript for interactivity.

II

**List of Table**

1 Table 1.1 Definations 18

2 Table 1.2 Abbreviations 20

3 Table 1 Customer 46

4 Table 2 Customer Service Request 46

5 Table 3 Employee 47

6 Table 4 Admin 47

7 Table 5 Hospital 48

8 Table 6 School 48

9 Table 7 University 49

10 Table 8 Tax Payments 49

11 Table 9 Hall Booking 50

12 Table 10 Certificates 50

13 Table 11 Wards 51

14 Table 12 Scholarships 51

15 Table 13 Messages 52

16 Table 14 Payments 52

17 Table 15 Property\_Detail 53

III

List of Figures

1 Figure. 1 Use case Diagram 1

2 Figure. 2 E.R Diagram 2

3 Figure. 3 (a) Activity Diagram Customer 3

4 Figure. 3 (b) Activity Diagram Employee 4

5 Figure. 3 (c) Activity Diagram Admin 5

6 Figure. 4 (a) Sequence Diagram Admin 6

7 Figure. 4 (b) Sequence Diagram Employee 7

8 Figure. 4 (c) Sequence Diagram Customer 8

9 Figure. 5 DFD Diagram(Level 0) 9

10 Figure. 6 DFD Diagram(Level 1) 10

11 Figure. 7 DFD Diagram(Level 2) 11

12 Figure. 8 Class Diagram 12

13 Figure. 9 (a) Flowchart Customer 13

14 Figure. 9 (b) Flowchart Employee 14

15 Figure. 9 (c) Flowchart Admin 15

IV

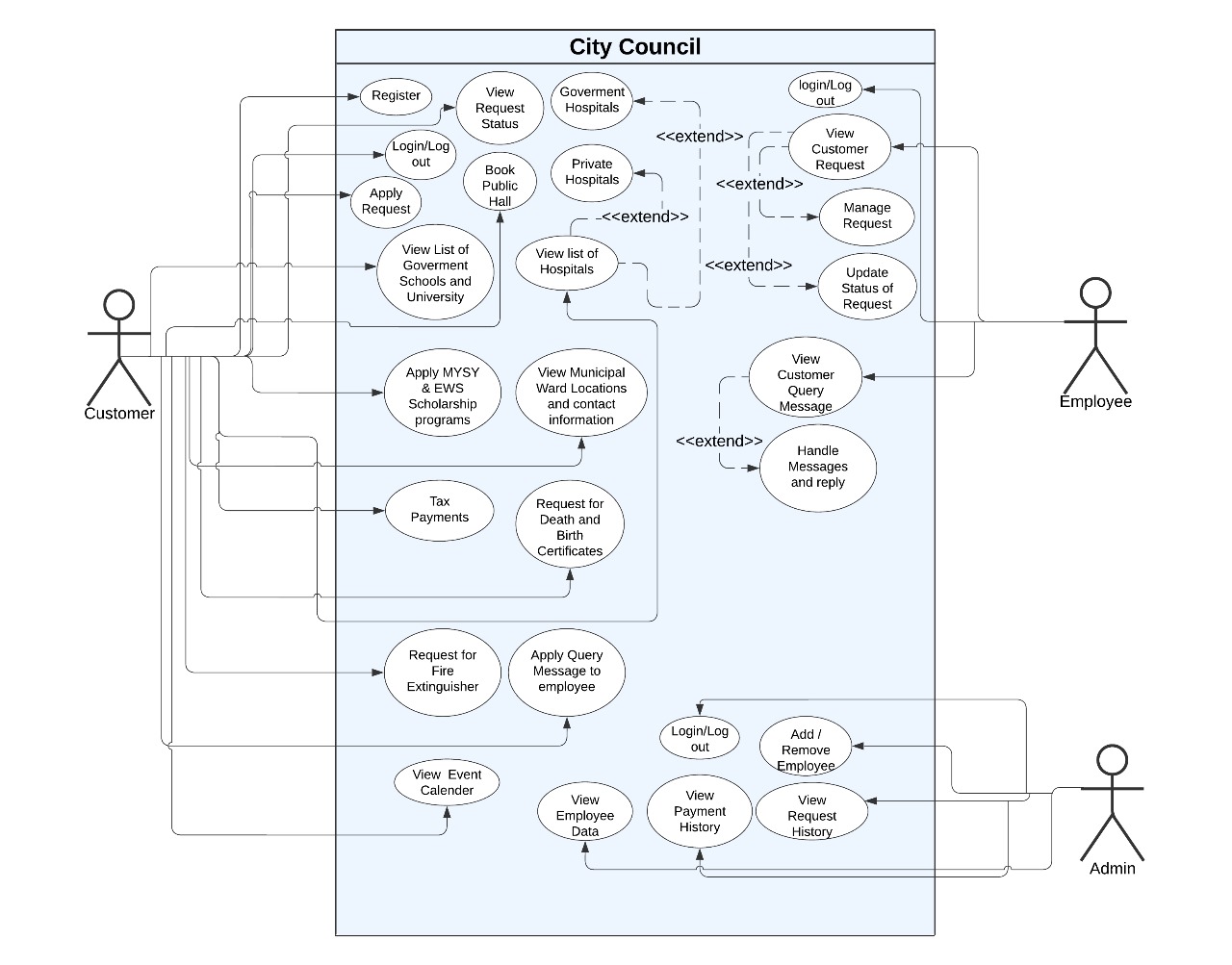


Figure. 1 Use case Diagram

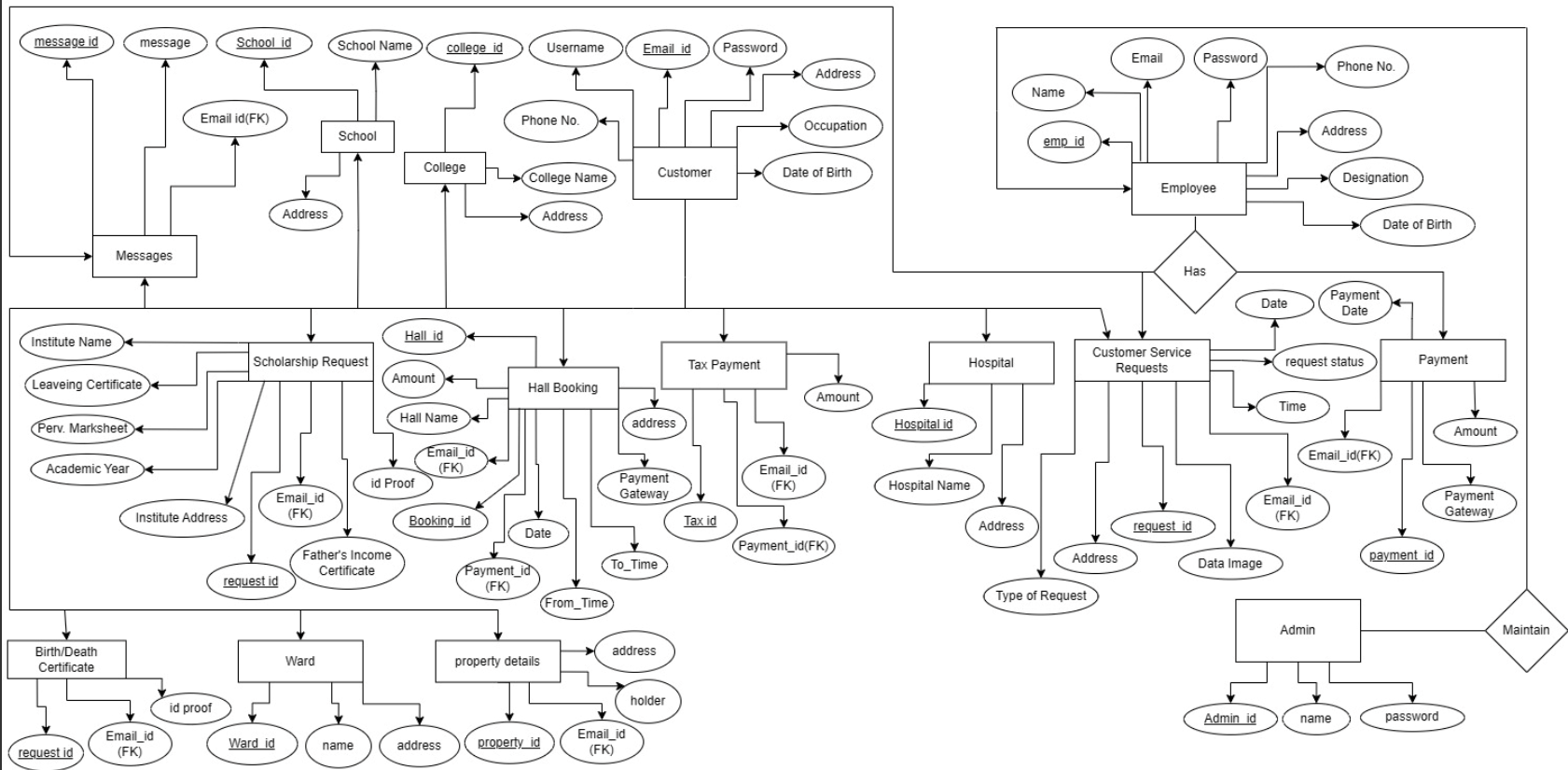


Figure. 2 E.R Diagram

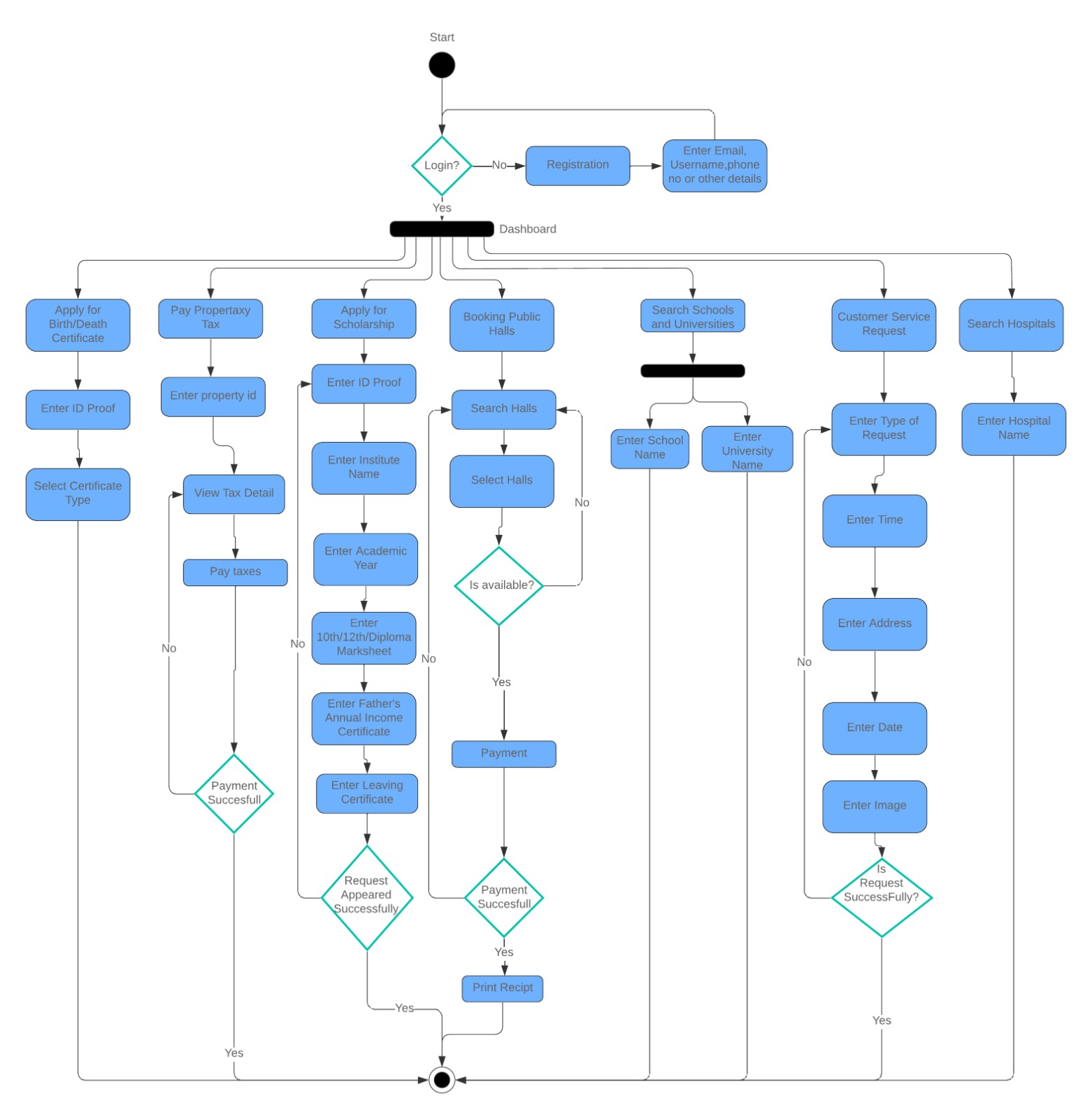


Figure. 3 (a) Activity Diagram Customer

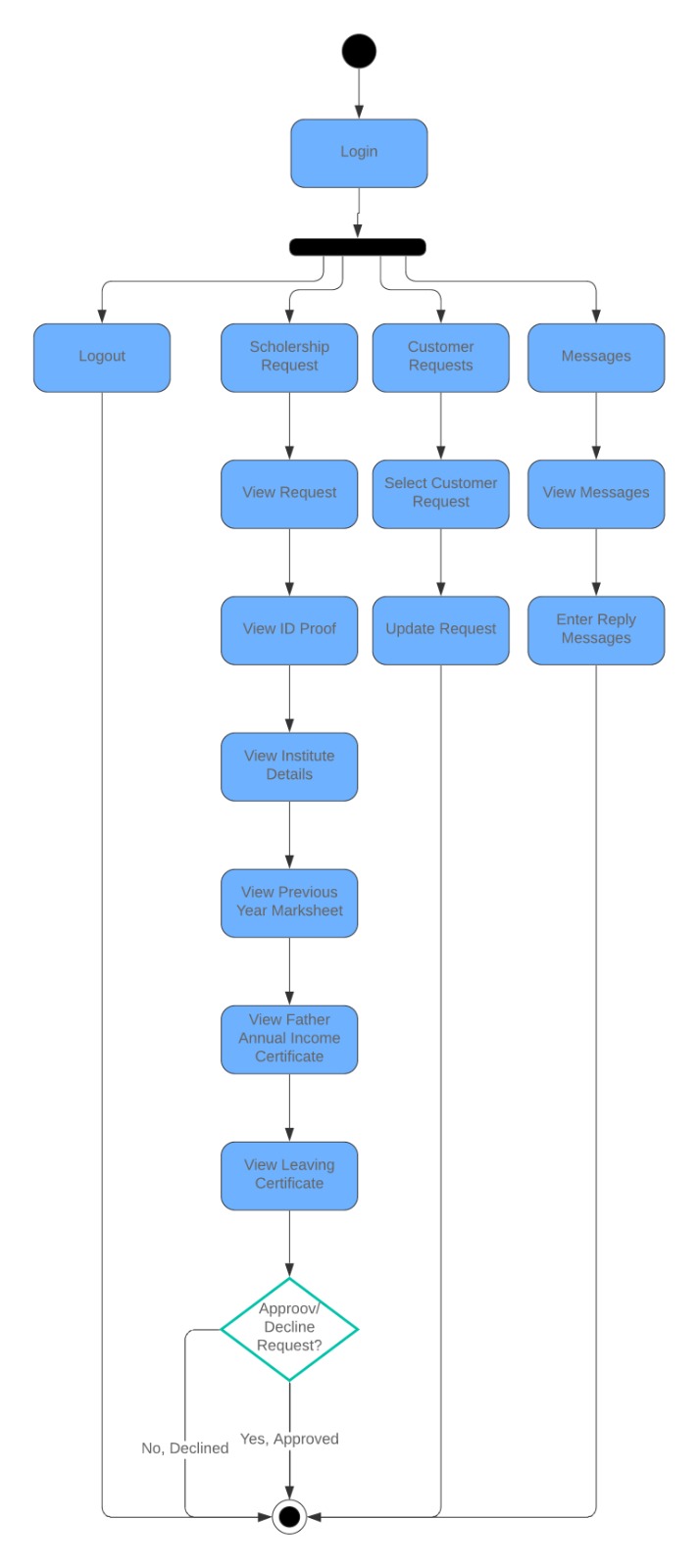


Figure. 3 (b) Activity Diagram Employee

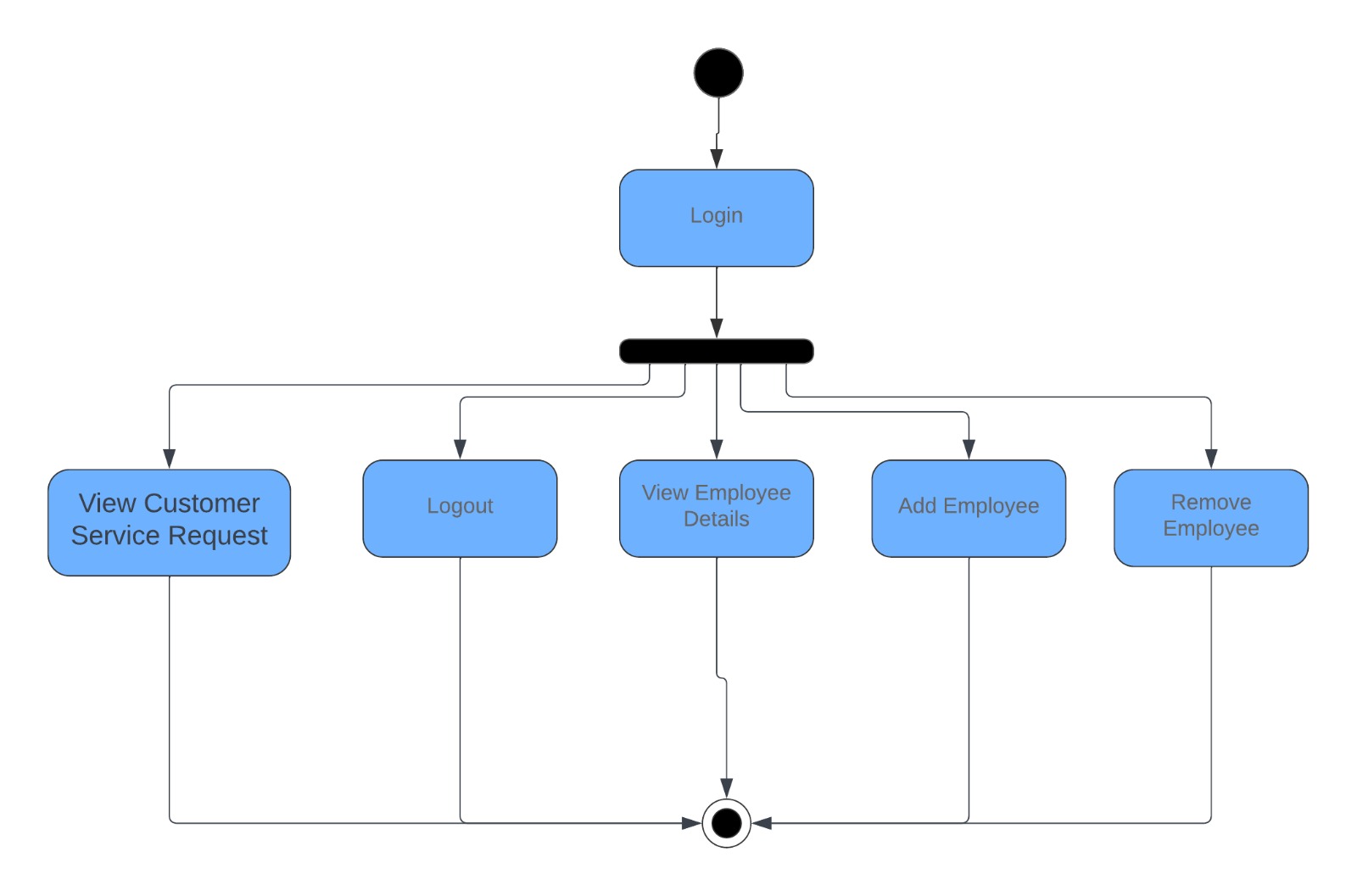


Figure. 3 (c) Activity Diagram Admin

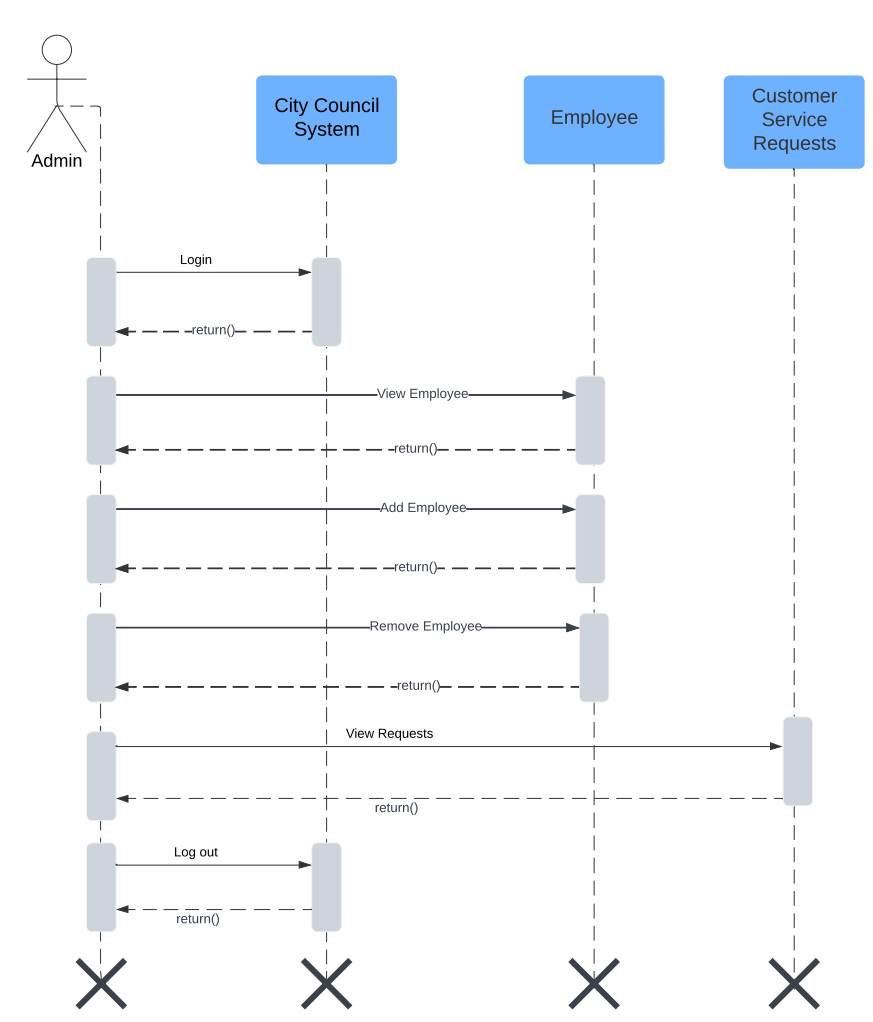


Figure. 4 (a) Sequence Diagram Admin

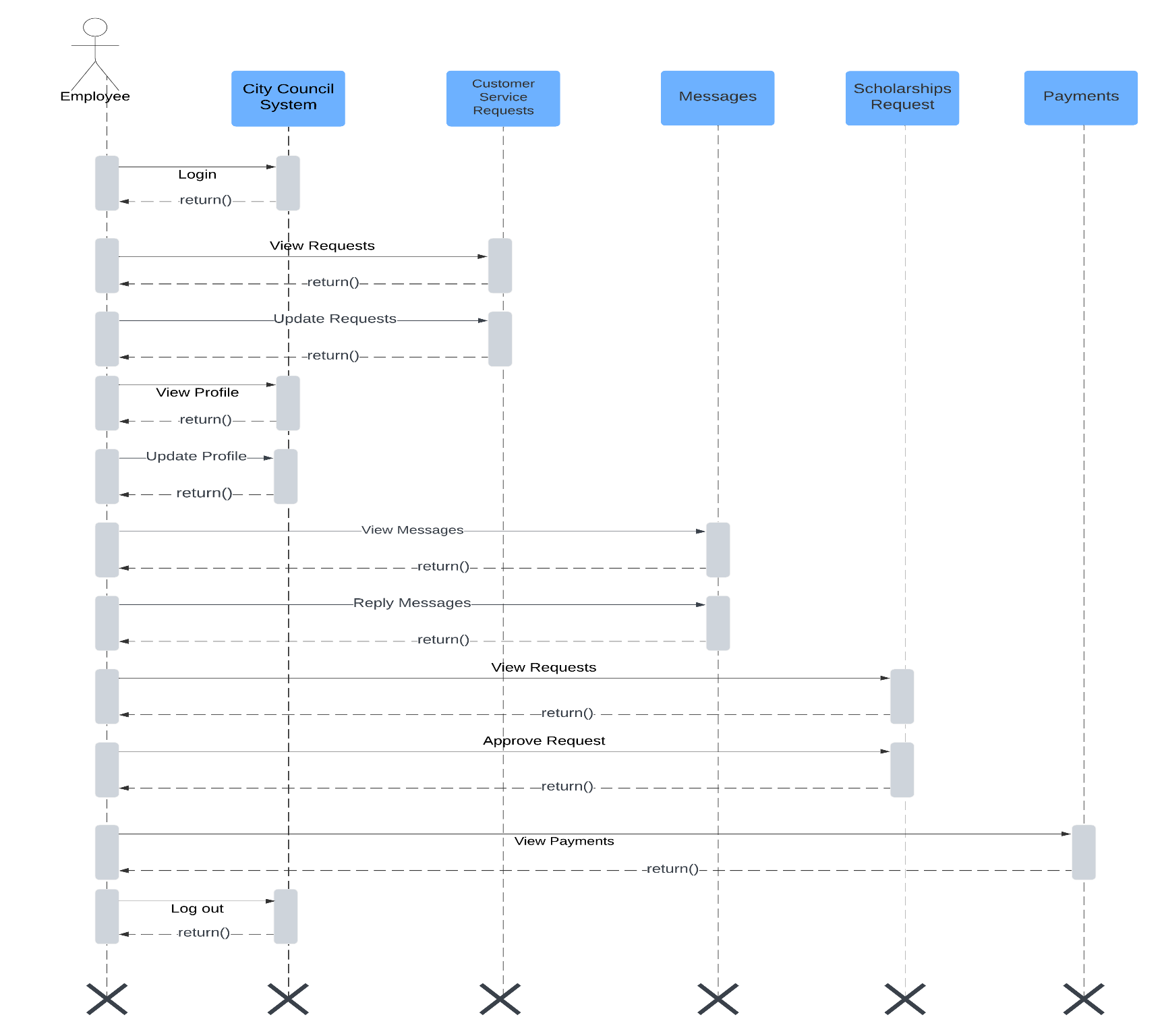


Figure. 4 (b) Sequence Diagram Employee

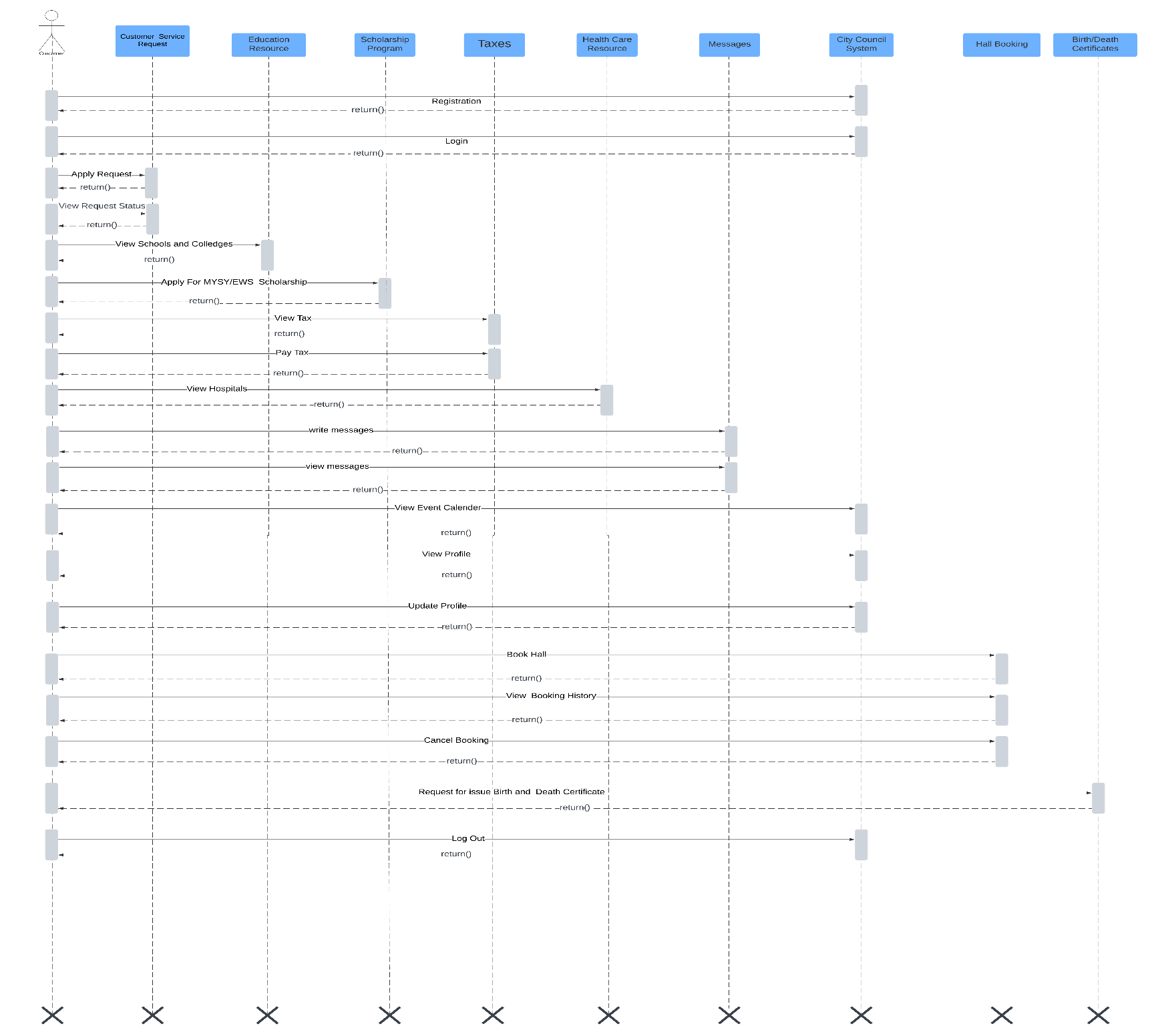


Figure. 4 (c) Sequence Diagram Customer

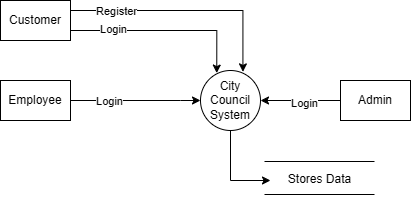


Figure. 5 DFD Diagram (Level 0)

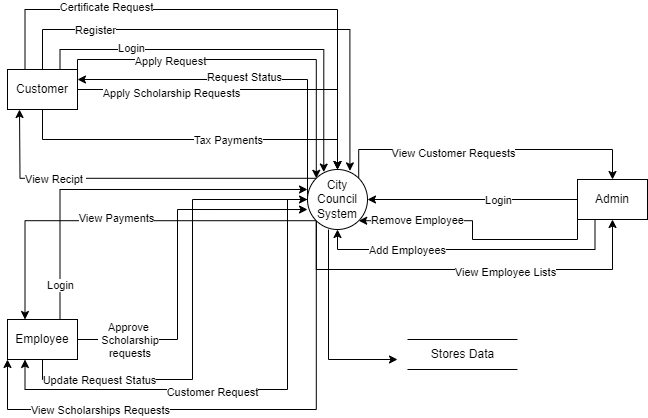


Figure. 6 DFD Diagram (Level 1)

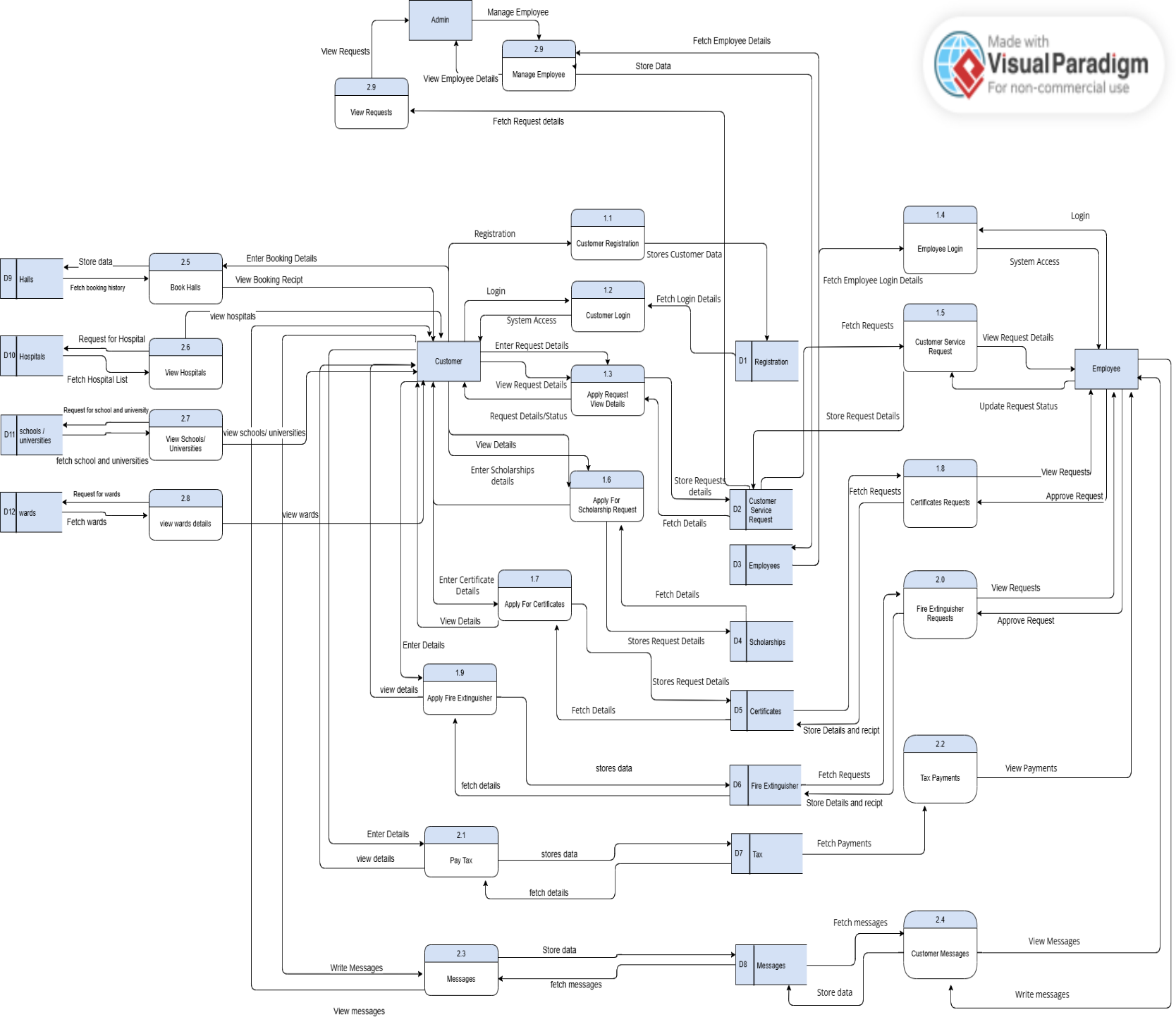


Figure. 7 DFD Diagram (Level 2)

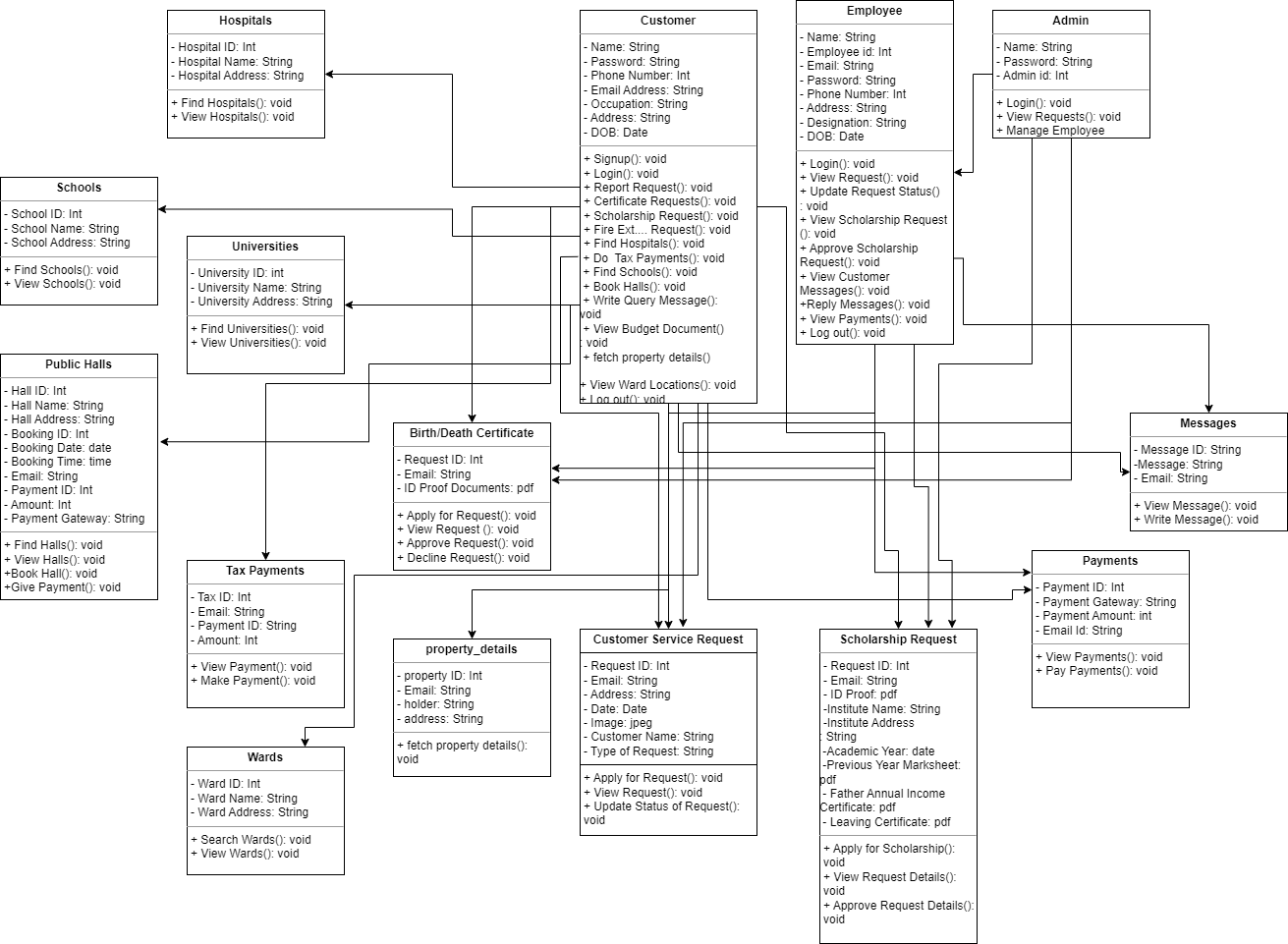


Figure. 8 Class Diagram

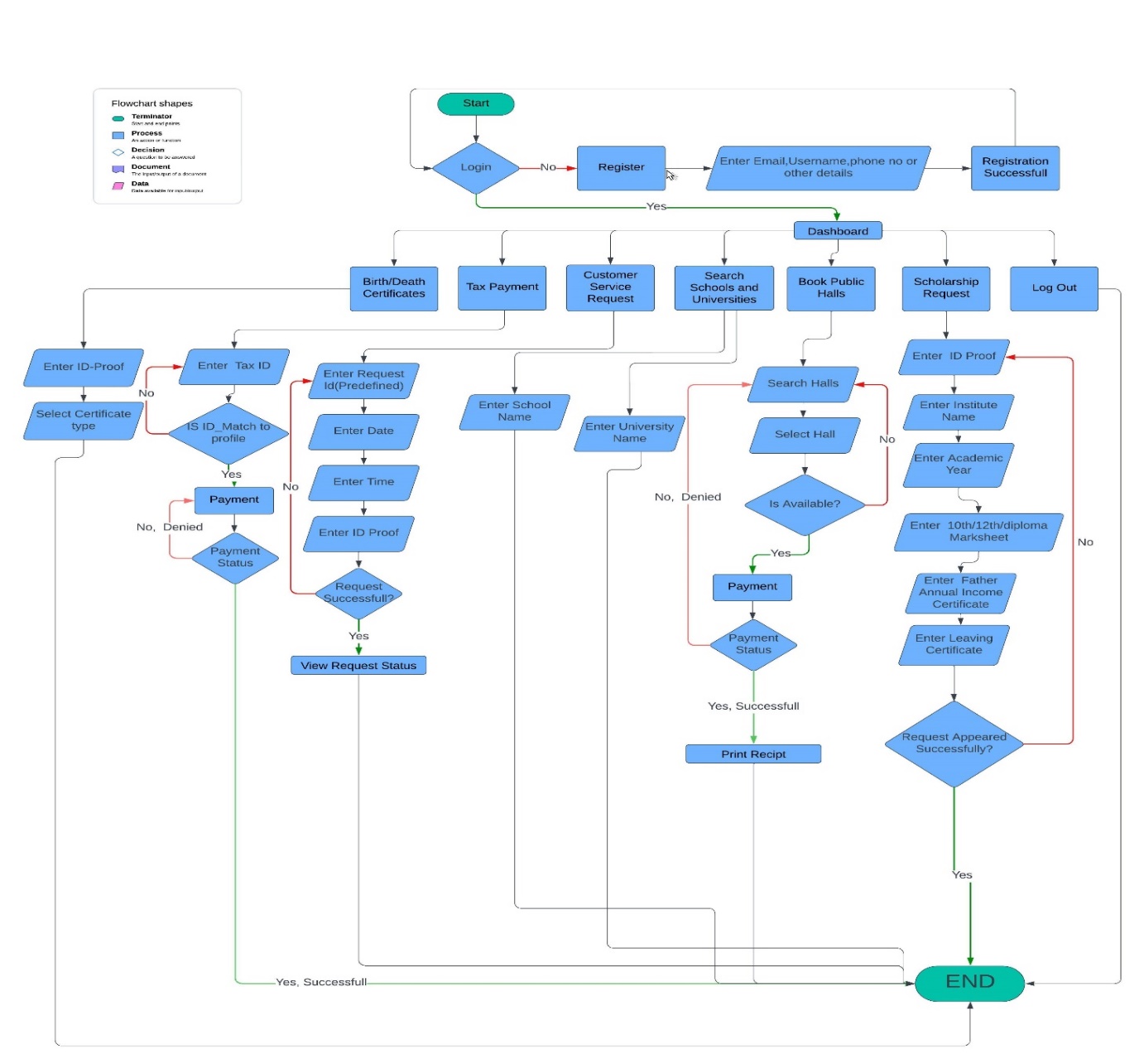


Figure. 9 (a) Flowchart Customer

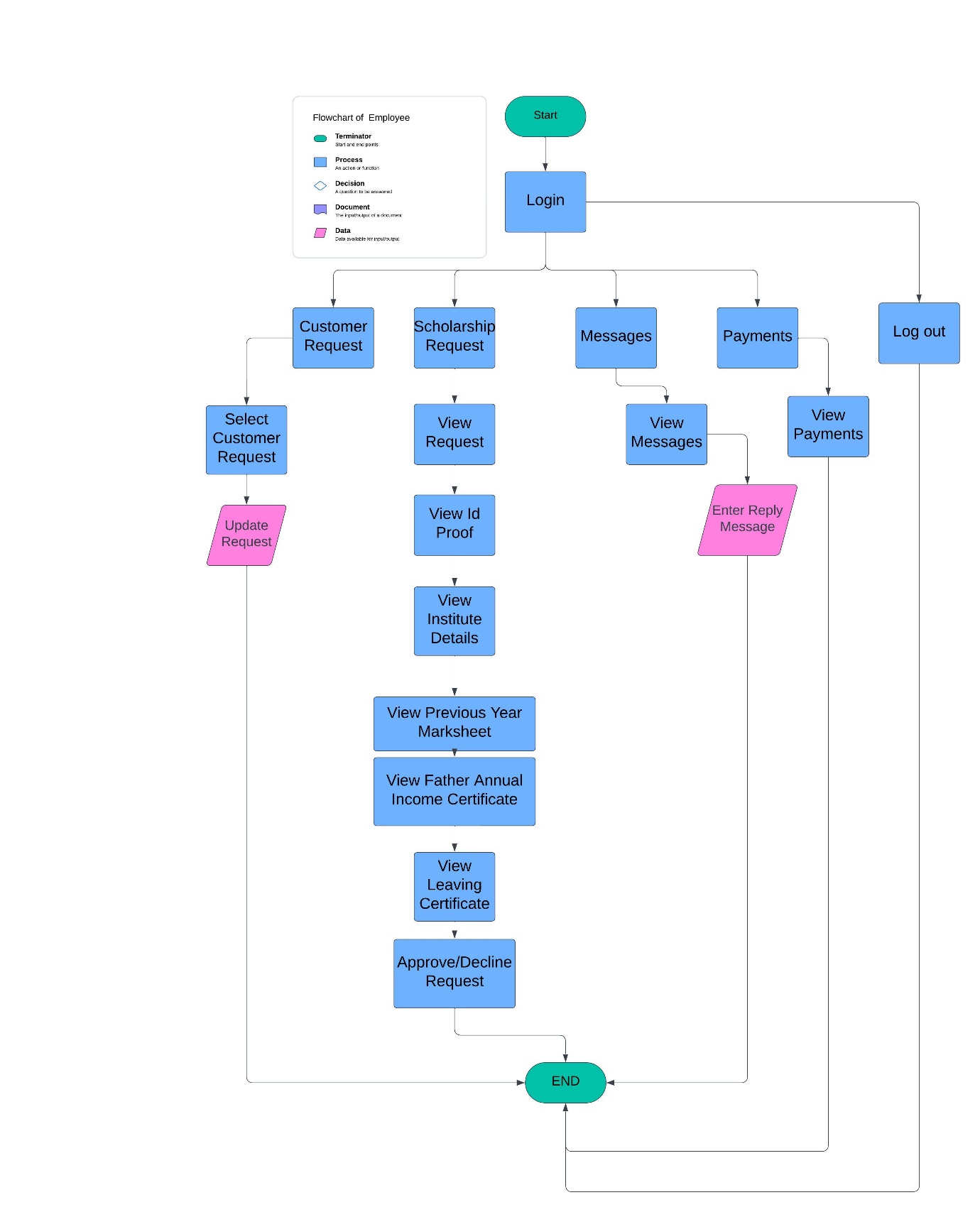


Figure. 9 (b) Flowchart Employee

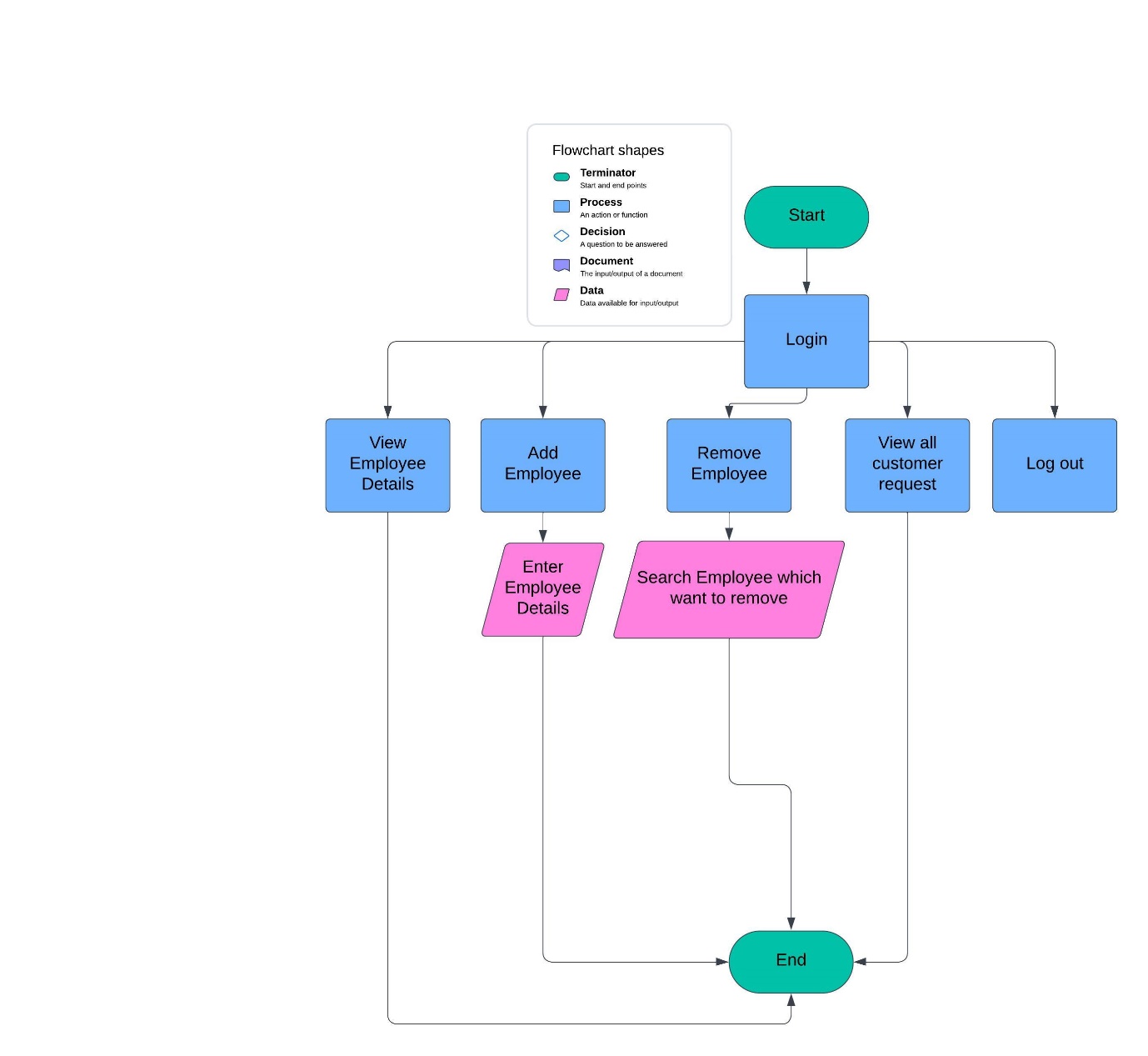


Figure. 9 (c) Flowchart Admin

# Introduction

## Document Purpose

The purpose of this document is to describe in detail the features and requirements of the City Council's online platform. This platform is designed to make accessing city services much more convenient for citizens. By signing up on the website, users can report non-urgent issues, find information about public health services, and access details about local schools and universities. The platform also offers features like booking public function halls for events, paying property taxes, and requesting certificates. These services are all organized in one place to make them easily accessible, ensuring that citizens can interact with their local government more efficiently.

This document will cover the essential functions of the system, detailing how it will be used by different types of users—citizens, employees, and administrators. For instance, citizens can submit requests or inquiries and track their progress, while employees can monitor these requests, update their statuses (whether they are pending, in progress, or completed), and communicate directly with citizens to provide updates or seek additional information. Administrators play a crucial role in managing the platform by handling user accounts, keeping track of service records, and ensuring that requests are completed in a timely manner. They can also view feedback provided by users to continually improve the platform's efficiency. The ultimate goal is to create a user-friendly system that enhances the delivery of city services and provides a better overall experience for everyone involved.

## Product Scope

The City Council's online platform is designed to be a straightforward and easy-to-use website where citizens can access a variety of city services all in one place. The platform will allow users to do things like report non-urgent issues, such as potholes or broken streetlights, and find important information about public health, schools, and universities in their area. Additionally, the platform will provide a way for citizens to book public halls for events, pay their property taxes, and request official documents like birth and death certificates. By putting all these services together in one online space, the platform aims to save citizens time and effort, making it easier for them to take care of their needs without having to navigate multiple websites or make trips to different city offices. The platform is built to be simple and accessible, so everyone can use it, regardless of their familiarity with technology.

The platform is also created with the needs of city employees and administrators in mind. Employees will be able to log in to the platform to view and manage service requests that come in from citizens. They can update the status of these requests—marking them as pending, in progress, or completed—and communicate directly with citizens if more information is needed or to provide updates on the progress. For administrators, the platform will offer tools to manage employee accounts, keep track of all the service requests and records, and monitor how quickly and efficiently these requests are being handled. They can also view feedback from citizens to see where improvements can be made to the services provided. The overall goal of the platform is to make city operations more efficient, improve communication between the city and its residents, and ensure that everyone who uses the platform has a positive experience.

## Intended Audience and Document Overview

**Intended Audience**: This document is intended for several key groups involved in the development and use of the City Council platform. The primary audience includes the project development team, which consists of the developers and designers who will be responsible for building the platform. This document will guide them through the detailed requirements and specifications necessary to ensure the platform meets the intended objectives. City officials, who will be the main users of the system, are also a key audience. They will use the document to understand how the platform will help them manage city services more efficiently and improve communication with citizens. Additionally, the quality assurance team will refer to this document to ensure that the final product meets all the required standards and functions as expected. System administrators, who will be tasked with maintaining the platform, will also rely on this document to manage user accounts, monitor system performance, and handle any technical issues that arise.

**Document Overview:** The document provides a comprehensive overview of the City Council platform, detailing its purpose, design, and functionality. It covers everything from the project’s goals and the overall system architecture to the specific features the platform must include and the performance standards it needs to meet. The document also includes sections on how users will interact with the platform, supported by use case diagrams, and explains any assumptions and dependencies that were considered during the planning phase. By outlining these details, the SRS ensures that all stakeholders have a clear understanding of the project’s objectives and the steps needed to achieve them. This shared understanding is crucial for the successful development, implementation, and maintenance of the City Council platform.

## Definitions, and Abbreviations

**Definitions:**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Admin | A person whose responsibility is to manage and maintain the  infrastructure of the system. |
| Class Diagram | A diagram consisting of a group of classes and interfaces reflecting  important entities of the domain of the system being modeled, and the  relationships between these classes and interfaces. |
| Data Flow  Diagram | A functional model of a software system that describes how outputs are  derived from inputs. A diagram contains processes, data flows, actors  and data stores. |
| Database | A collection of data or information typically stored on a computer system  and organized to facilitate retrieval and modification. |
| Database  Management  System | A software system that enables users to define, create, maintain, and  control access to a database. |
| Key | Either a Primary Key or a Foreign Key. |
| Login | The process to gain access on any system. |
| Primary Key | A set of fields in a database table that is used to uniquely identify records  in the table. |
| Record | A unique row in a table in a database consisting of a set of fields that  describe a single occurrence of some entity described by the table. |
| Table | A collection of records in a relational database. |
| Update | The process of modifying, adding or removing existing data. |
| Use Case  Diagram | A diagram that represents the use cases of the system, i.e., interaction  among the system, external entities, and the principal users of the  system. |
| Windows  Operating  System | A computer operating system by Microsoft that provides a graphical user  interface (GUI), virtual memory management, multitasking, and support  for many peripheral devices. |
| Abstract | A short piece of writing that tells you the main contents of a website, speech, etc. |
| Product Scope | The product scope is defined by the features and functions that characterize a product, service, or result. |
| Abbreviations | A shortened form of a written word or phrase used in place of the whole word or phrase. |
| Document Conventions | Document Conventions is part where you explain the formats used in the document. What are bold words for, what is italic used for and so on. |
| Intended Audience | Intended audience is defined as the group of people for which a service or product is designed. |
| User | An individual who accesses the platform to avail of city services, report issues, or request information. |
| Employee | A city worker responsible for managing and updating service requests, viewing messages, and responding to users. |
| Admin | An administrator with elevated permissions, responsible for overseeing employee accounts, service requests, payment history, and other administrative tasks. |
| Service Request | A formal submission by a user to the platform seeking assistance or service from the city. |
| Bootstrap | A front-end framework used for designing responsive and mobile-first websites. |
| PHP | A server-side scripting language used for web development. |
| MySQL | An open-source relational database management system used to store and manage data. |
| JavaScript | A programming language used to create interactive effects within web browsers. |

## Abbreviations:

|  |  |
| --- | --- |
| **ACRONYMS** | **MEANING** |
| DFD | Data Flow Diagram |
| JPEG | Joint Photographic Experts Group (Image Format) |
| OS | Operating System |
| SRS | Software Requirements Specification |
| e.g. | for example |
| i.e. | that is |
| UI | User Interface |
| UX | User Experience |
| DB | Database |
| SQL | Structure Query Language |
| HTML | Hyper Text Markup Language |
| CSS | Cascading Style Sheet |
| JS | JavaScript |
| PHP | Hypertext Preprocessor |
| API | Application Program Interface |
| HTTP | Hyper Text Transfer Protocol |
| HTTPS | Hyper Text Transfer Protocol Secure |
| QA | Quality Assurance |
| SMTP | Simple Mail Transfer Protocol |
| GUI | Graphical User Interface |
| CPU | Central Processing Unit |
| UML | Unified Modeling Language |
| ERD | Entity Relationship Diagram |
| RAM | Random Access Memory |

## Document Conventions

# ****Section Headings****: Section headings are formatted in bold and title case for clear navigation and hierarchy within the document.

# ****Subsection Headings****: Subsection headings are italicized and numbered, with appropriate indentation to distinguish them from main sections.

# ****Figures****: Figures are labeled sequentially (e.g., Figure 1) with captions placed directly below the figure. They are referenced in the text by their number.

# ****Tables****: Tables are labeled and numbered sequentially (e.g., Table 1) with titles placed above the table. They are referenced in the text by their number.

# ****Bulleted Lists****: Bulleted lists are used to present unordered items, key points, or features for quick reference and clarity.

# ****Numbered Lists****: Numbered lists are used for steps, procedures, or any ordered information that requires a clear sequence.

# ****Equations****: Equations are centered and numbered sequentially, with their reference numbers placed in parentheses to the right margin.

# ****References****: References are cited within the text by number in square brackets (e.g., [1]), and listed in numerical order at the end of the document.

# ****Citations****: Citations are placed directly in the text using numbers in square brackets, with multiple citations combined (e.g., [1, 2, 3]).

# ****Acronyms and Abbreviations****: Acronyms and abbreviations are defined at first use, with the full term followed by the acronym in parentheses. Subsequent uses only require the acronym.

# ****Font****: The primary font is a serif type, 10-point for body text, with headings in a larger or bolder version of the same font.

# ****Text Alignment****: All body text is left-aligned and justified, ensuring that the document has a clean, professional appearance without uneven spacing.

# ****Margins****: The document is formatted with 1-inch margins on all sides, ensuring that the content is well-centered and easy to read.

# ****Page Numbers****: Page numbers are placed in the bottom center of each page, providing easy reference without distracting from the content.

# ****Document Title****: The document title is centered at the top of the first page, in a larger, bold font, followed by the author’s name and affiliation in a slightly smaller font.

# ****Abstract****: The abstract is a brief summary of the document, formatted in italics and justified, located after the title and author information.

# ****Keywords****: Keywords are listed immediately following the abstract, separated by commas, and introduced with a "Keywords" heading.

# ****Acknowledgments****: The acknowledgment section is included before the references, recognizing contributions and support related to the project.

# ****Footnotes****: Footnotes are used sparingly, numbered sequentially, and placed at the bottom of the page where the reference occurs.

# ****Hyperlinks****: Hyperlinks are embedded in the text, underlined, and colored, providing direct access to external resources or additional information.

# ****Consistency****: Terms, formatting, and styles are used consistently throughout the document to ensure clarity and readability.

# ****Appendices****: Appendices are labeled alphabetically (e.g., Appendix A) and included after the references, providing supplementary information without interrupting the main content.

# ****Headers and Footers****: Headers and footers include minimal information, such as page numbers or document titles, to keep the document clean and focused.

# ****Document Layout****: The main content of the document is arranged in a single column format, with consistent spacing and alignment for easy reading.

# ****Text Color****: The document primarily uses black text on a white background for maximum readability, with other colors used sparingly in figures or diagrams.

## References and Acknowledgments

**Reference**

* <https://online.visual-paradigm.com/>
* [https://draw.io](https://draw.io/)
* [www.lucidchart.com](http://www.lucidchart.com/)

**Acknowledgment**

The presentation of this report gives us a great pleasure and satisfaction in presenting this report

on my project work as a part of the final fulfillment for the Diploma of Engineering. We would

like to take this opportunity to express our sincere gratitude to several people, without whose

help and encouragement, it would have been impossible for us to carry out the desired work.

We are thankful to Prof. Miral Borad for her help during our project duration, which enabled us to complete our project with desired results. We are also very much thankful to all other faculty members for their support throughout our study. Last but not the least the most important of all out association to Faculty members and project guide.

They not only guided us throughout the project but also gave us a constant encouragement,

which instilled in us a confidence to make the best of this project. We are heartily thankful to

them.

With Sincere Regards,

Parin Zala.

Sneh Faldu.

Nihar Trivedi.

Jay Ramani.

Jyotin Tank.

# Overall Description

## Product Perspective

The City Council platform is meant to be a key part of how the city manages its services. It will be an online portal that helps city officials and citizens communicate more easily, making it simpler for people to report issues, request services, and give feedback. This platform will work together with the city’s existing systems, like those used for managing public services and other local government websites. The goal is to make city services more accessible and responsive to citizens' needs through a user-friendly interface.

Technically, the City Council platform will connect with various backend systems to gather and process information. It will be built using common web technologies like HTML, CSS, and JavaScript, along with a backend that supports databases and user login. The platform will need to support different types of users, including citizens, city employees, and administrators, each with their own set of permissions and features. The design will focus on making the system scalable, secure, and easy to use, ensuring that it can handle more users and requests as needed while keeping all data safe. Overall, this platform aims to modernize how the city serves its residents.

## Product Functionality

The City Council platform will provide the following major functions for all user roles:

* **User Registration and Login:** Users can create accounts and log in securely based on their roles (citizens, city employees, administrators).
* **Service Request Submission:** Citizens can submit requests for city services, report issues, or request public works.
* **Request Tracking and Status Updates:** Citizens can track the status of their submitted requests. City employees can update the status of requests (Pending, In Progress, Completed).
* **Feedback and Communication:** Citizens can provide feedback on services and communicate with city officials through a chat feature. City employees can respond to citizen queries and feedback.
* **Public Information Access:** Citizens can access information about local services, schools, universities, and health facilities. All users can search a database of frequently asked questions (FAQs).
* **Booking Public Halls:** Citizens can book public function halls and other city-owned facilities for events.
* **Administrative Management:** Administrators can create, modify, and delete employee accounts. Administrators can view service requests, and monitor payments.
* **Data Security and Privacy:** The system implements security measures to protect user data and ensure compliance with data protection regulations. Administrators manage security protocols and ensure that all user data is securely handled.

## Users and Characteristics

**Citizens**: Use the platform to report issues, access public services, book halls, manage tasks, request certificates, and provide feedback.

Require easy navigation, efficient service access, tracking of requests, and simple feedback mechanisms.

**Employees**: Manage and update service requests, view and respond to customer messages.

Need efficient request management tools, communication features with citizens, and task tracking capabilities.

**Admins**: Create, modify, and delete employee accounts, manage service requests, oversee payment and booking records, monitor request completion times, and review customer feedback.

Require comprehensive administrative tools, detailed reporting and monitoring features, and control over user accounts and service management.

## Operating Environment

The City Council system operates within a specific technological and software environment to ensure optimal performance and accessibility:

1. **Operating System:** The project is designed to be compatible with all major operating systems, including Windows, macOS, and Linux for desktop users, ensuring widespread accessibility across various devices.
2. **Processor:** While the system can run efficiently on dual-core processors, a quad-core processor is recommended for optimal performance, particularly when processing multiple user requests simultaneously or handling complex administrative tasks.
3. **RAM:** A minimum of 2 GB RAM is required, with higher RAM capacities (4 GB or more) recommended for better overall performance, especially when handling large volumes of data, such as customer requests and feedback.
4. **Storage:** The application itself has a lightweight footprint, requiring less than 200 MB of storage space for installation on user devices. Server-side, it is recommended to allocate at least 10 GB of free storage space to accommodate the growing database of user accounts, requests, and feedback.
5. **Graphics:** The City Council system is primarily a text-based interface with minimal graphical requirements. However, to enhance the user experience, the system supports high-resolution displays and requires a basic GPU capable of rendering responsive design elements smoothly.

The City Council system is designed to be accessible on a range of devices and operating systems while providing an optimal user experience for both citizens and administrative staff.

## Design and Implementation Constraints

# Hardware Limitations: The system must run on a variety of devices, including older phones with limited memory and processing power. This means the developers have to ensure the system works smoothly, even on devices with as little as 2 GB of RAM. The system should also respond quickly, so developers need to optimize it to work efficiently on slower devices.

# Integration with Other Systems: The City Council system may need to connect with other existing city services and databases. This might force developers to use specific methods or tools that work with these systems, which could limit their choices.

# Required Tools and Technologies: The developers might be required to use certain technologies like HTML for the front end and specific frameworks for the backend. They might also be restricted to using certain databases like MySQL. These requirements limit the developers' options to try other tools or technologies.

# Handling Multiple Operations: The system may need to handle many tasks at once, like processing several user requests simultaneously. This requires careful planning to ensure everything runs smoothly, which can limit how the system is designed.

# Programming Language Restrictions: The programming languages used might be limited to what the team is familiar with or what the City Council’s existing systems support. This means developers might not be able to use newer or more efficient languages.

# Communication Requirements: The system needs to use certain communication methods, like secure HTTPS, to protect user data. These required methods could limit other ways the system communicates with users or other systems.

# Security Measures: Strong security features, like data encryption and user login checks, are required to protect user information. These security needs can add extra steps to the development process and limit some design choices. The system must also follow data protection laws, which could limit how data is stored and managed.

# Design and Coding Standards: The system needs to follow specific design and coding rules, especially if the City Council’s IT department will maintain it. This might include using a particular coding style or documentation method, which could limit the developers’ flexibility.

## User Documentation

For the City Council system, we will provide several types of user documentation to help users understand how to use the software effectively. This will include a user manual that explains how to perform key tasks, such as submitting requests or booking facilities. The manual will be available in both PDF and online formats, making it easy to access on any device. Additionally, we will include an online help section within the software for quick guidance, as well as video tutorials that demonstrate important features. An FAQ section will address common questions, and release notes will be provided with each update to keep users informed about changes. All documentation will be designed to be clear and easy to follow.

## Assumptions and Dependencies

**Assumptions:**

1. **User Base:** It is assumed that the user base of the platform will include a diverse range of individuals from various age groups and technical backgrounds. The design will consider this diversity to ensure the platform is user-friendly and accessible.
2. **Technology Stack:** The project assumes that the technology stack used (HTML, CSS, Bootstrap, PHP, MySQL, and JavaScript) will remain stable throughout the development cycle. Significant changes or updates to these technologies may impact the project.
3. **Internet Connectivity:** It is assumed that users will have stable internet access when interacting with the platform. Variations in internet speed or connectivity may affect user experience.
4. **Data Security:** The project assumes that data security and privacy regulations applicable to city services will be adhered to. Any changes in these regulations could necessitate modifications to the system’s security measures.
5. **Third-Party Services:** The project assumes the availability and reliability of third-party services or APIs that might be integrated for functionalities such as payments or external data fetching. Any downtime or changes in these services could impact the platform’s performance.
6. **Client Requirements:** It is assumed that the client will provide timely feedback and necessary information required for the development process. Delays in feedback or information could affect the project timeline.
7. **Browser Compatibility:** It is assumed that the platform will be used primarily on modern web browsers. Support for older or less common browsers may require additional development effort.

**Dependencies:**

1. **Database Access:** The project depends on access to a MySQL database for storing and retrieving data. Any issues with database performance or availability could impact the system’s functionality.
2. **Server Environment:** The platform’s deployment depends on a server environment that supports PHP and MySQL. Changes or issues with the server environment could affect deployment and performance.
3. **Bootstrap Framework:** The project depends on the Bootstrap framework for responsive design and layout. Updates or changes to the Bootstrap framework may require adjustments in the design.
4. **External APIs:** If the project integrates with external APIs for functionalities like payment processing or location services, it depends on the stability and availability of these APIs.
5. **Client Infrastructure:** The project is dependent on the client’s infrastructure for hosting and maintaining the platform. Any changes or issues with the client’s infrastructure could affect the platform’s operation.

# Specific Requirements

## External Interface Requirements

### User Interfaces

This project features user interfaces (UIs) carefully designed to provide users with an intuitive and efficient experience for accessing city services. These UI elements include:

1. **Home Screen:-** The central hub where users begin their interaction with the platform. It offers options for accessing services, viewing updates, and navigating to different sections of the site.
   * **Navigation Bar:** Located at the left of the screen, the navigation bar offers links to the main sections of the platform such as Home, Services, Profile, Support, Contact Us and etc.
   * **Search Bar:** A prominently displayed search bar that allows users to quickly find specific services or information by typing in keywords.
   * **Service Highlights:** A grid of icons representing the most frequently accessed services like "Report an Issue," "Pay Property Taxes," "Find Public Health Services," and "Book a Hall."
   * **City Announcements:** A section that displays the latest news, updates, and alerts from the city, helping users stay informed about important information.
2. **Service Request Screen:** A dedicated interface where users can submit and manage their service requests. This screen includes a multi-step form for detailed submissions, a history section showing past and current requests, and buttons for actions like submitting, canceling, or saving drafts.

* **Multi-step Request Form:** A detailed form that guides users through the process of submitting a request. Users can specify the type of service, provide necessary details, and upload supporting documents.
* **Request History Panel:** Displays a list of all service requests the user has submitted, with current status indicators such as "Pending," "In Progress," or "Completed."

1. **Public Function Hall Booking Screen:** This screen allows users to book public function halls for events. The interface includes a calendar view for selecting available dates, a booking form for entering event details, and standard buttons for booking, canceling, or reviewing terms and conditions.
2. **Admin Dashboard:** An interface provided for administrators to manage the platform and oversee service operations. Key features include an overview section with metrics on active service requests, tools for employee management.
3. **Employee Dashboard:** Employees can view all assigned service requests, update their status, and communicate with the user if more information is needed. Employees can read and respond to messages sent by users related to their requests.
4. **Chat box Interface:** A user-friendly chat box where users can ask questions and receive immediate assistance. The chat box is integrated across all major screens, including:

* **Chat Window:** A pop-up chat window that appears in the corner of the screen, allowing users to type their questions and receive answers from either automated responses or live support.
* **Quick Links:** Frequently asked questions or popular service links are provided within the chat box for easy access.

1. **Feedback Screen:**  
   After completing a service request, users are encouraged to provide feedback. The feedback screen includes:

* **Rating System:** A star-based rating system where users can rate their experience with the service or employee assistance.
* **Comment Section:** Users can leave detailed feedback or suggestions for improving the service.
* **Submit Button:** Once the feedback is filled out, users can submit it, and it will be stored for review by administrators.

### Hardware Interfaces

In the context of the City Council project, which primarily operates as an online platform, the interaction between the software and hardware components is minimal. However, if there are hardware interfaces involved, such as servers, client devices (like PCs, tablets, or smartphones), or any other peripheral devices, these interfaces need to be clearly defined.

**Server Hardware Interface**

1. **Logical Characteristics:**

* The server hosts the application, database, and API layers. It handles all incoming requests from clients (users, employees, admins) and processes them.
* The software interacts with the server hardware to manage data storage, processing, and network communication.
* Data interactions include reading and writing user data, handling service requests, managing feedback, and processing payments.
* Control interactions involve managing user sessions, request processing workflows, and ensuring proper execution of background tasks (e.g., updating request statuses).

1. **Physical Characteristics:**

* The server hardware may consist of physical or cloud-based infrastructure (e.g., AWS, Azure).
* Supported device types include standard server configurations, capable of running web servers (e.g., Apache, Nginx) and database management systems (e.g., MySQL, PostgreSQL).
* The server hardware should support Ethernet or Wi-Fi connections for communication over the internet.

1. **Special Libraries:**

* If cloud services are used, libraries such as **AWS SDK for PHP** or **Azure SDK for PHP** may be utilized to interact with cloud resources.
* For data storage, libraries like **PDO (PHP Data Objects)** or **MySQLi** might be used to interact with the MySQL database on localhost.**Client Devices Interface**

1. **Logical Characteristics:**

* Client devices (PCs, smartphones, tablets) access the platform via a web browser or possibly a mobile app.
* The software interacts with the client hardware primarily through the device's web browser or operating system to display the UI, process user inputs, and handle data transmission.
* Data interactions include sending requests to the server, receiving responses, and rendering web pages.
* Control interactions involve managing user inputs (e.g., form submissions) and ensuring secure communication with the server.

1. **Physical Characteristics:**

* Supported devices include any internet-enabled device with a modern web browser (e.g., Chrome, Firefox, Safari).
* The client devices should have a stable internet connection (Wi-Fi, 4G/5G, Ethernet) to interact with the server.
* No special hardware is required beyond standard computing devices.

1. **Special Libraries:**

* The front-end may utilize libraries such as React.js or Angular.js to enhance the user experience.
* If developing a mobile app, libraries like React Native or Flutter could be used to build the interface.
* For secure communication, HTTPS protocols are implemented, possibly using libraries like OpenSSL.

### Software Interfaces

### The City Council platform is designed to work across various operating systems to ensure compatibility and reliability. Below is a description of how the platform interfaces with the operating system it runs on, focusing on the key aspects of process management, file handling, and network communication.

### Operating System Interface:

Supported Operating Systems:

* + - Linux (e.g., Ubuntu, CentOS)
    - Windows Server
    - macOS (for development purposes, if necessary)

##### **Process Management:** The City Council platform relies heavily on the operating system to manage the execution of various processes. These include the web server, database server, and any background tasks that may be necessary for the operation of the platform. The operating system’s ability to schedule processes and allocate resources is crucial to maintaining optimal performance and ensuring that all tasks are carried out efficiently.

##### **File System Interaction:** Interaction with the file system is another critical aspect of the platform’s operation. The software reads and writes several types of files, including configuration files, log files, and temporary data. These files are stored on the operating system’s file system, and proper management of these files is essential for smooth operation. For example, log files generated by the application are crucial for tracking user activities and troubleshooting issues.

##### **Network Communication:** Network communication is handled through the operating system’s network stack. This includes processing all incoming and outgoing HTTP/HTTPS requests, interacting with external APIs, and managing secure connections. The operating system manages the network interfaces and handles the transmission of data between the server and the clients, ensuring that all communication is secure and efficient.

### Communications Interfaces

### The City Council platform uses standard communication protocols to manage data exchanges between users, employees, and the system. The primary protocol for web communication is HTTPS, which ensures that all interactions between the user's browser and the server are encrypted. This encryption helps protect sensitive information like user credentials and service requests from being accessed by unauthorized parties.

### In addition to web communications, the platform also handles email notifications using the SMTP protocol. To keep these email communications secure, the system uses encryption (like TLS) to ensure that emails are protected while being sent. These measures ensure that both web and email communications are secure, reliable, and in line with common industry practices. The system is also designed to handle data transfers smoothly over typical network speeds, making sure that all necessary information is synchronized effectively between users and the platform.

### 3.2 Functional Requirements

The City Council platform is designed to handle a wide range of tasks for different types of users, including customers, employees, and administrators. Below, the platform's functions are organized into key areas, with a straightforward explanation of what each area covers.

**1. Customer Functions:**

The platform provides customers with easy access to city services, enabling them to manage their interactions efficiently.

1.1 **Account Management**

* Customers can sign up and create an account by providing basic details like their name, contact info, and address.
* Once registered, they can log in, update their personal information, or reset their password if needed.
* If a customer decides to leave the platform, they can delete their account, and all their personal data will be removed from the system.

1.2 **Submitting Service Requests**

* Customers can submit requests for various services, such as paying property taxes, booking public function halls, or requesting official certificates.
* The platform allows them to fill in necessary details for each request, like preferred dates, required documents, or special instructions.

1.3 **Tracking Requests**

* After submitting a request, customers can track its progress, which is marked as Pending, In Progress, or Completed.
* The platform automatically notifies customers via email or SMS when there’s an update on their request.

1.4 **Providing Feedback and Asking Questions**

* Customers can share their feedback on the services they received, including ratings and comments.
* If they have questions or need further assistance, they can use the platform's chat feature to get in touch with city employees.

**2. Employee Functions:**

Employees use the platform to manage and respond to customer requests, ensuring that city services are delivered smoothly.

2.1 **Managing Requests**

* Employees can view all incoming service requests, organized by type, urgency, and status.
* They can update the status of these requests, assign them to other team members, or mark them as completed when done.

2.2 **Communicating with Customers**

* Employees can respond directly to messages from customers through the platform.
* They can send updates, ask for more information, or clarify details about the customer’s request.

**3. Admin Functions:**

Administrators oversee the overall operation of the platform, including user management and system performance.

3.1 **Managing Employee Accounts**

* Admins can create, modify, or delete accounts for employees, ensuring that everyone has the right access level.
* They can also assist customers with account-related issues, like access problems or data updates.

3.2 **Monitoring and Maintenance**

* Admins have tools to monitor how the platform is running, including checking server status, network activity, and error logs.
* They can also schedule maintenance, back up data, and restore system functions if needed.

3.3 **Configuring Services**

* Admins can update the details of the services offered on the platform, including descriptions, requirements, and fees.
* They also manage the booking schedule for public function halls, making sure that reservations are handled efficiently.

3.4 **Ensuring Security**

* Admins are responsible for setting up security measures, such as password policies and access controls.
* They can also review user activity logs to detect and prevent any unauthorized access or suspicious behavior.

## Behaviour Requirements

### Use Case View: A Use Case View is a fundamental component of software design and architecture. It focuses on capturing the functional requirements of a system by describing the interactions between external actors (such as users or other systems) and the system itself as per Figure 1

# Other Non-functional Requirements

## 4.1 Performance Requirements

## The City Council platform is expected to handle a variety of tasks efficiently, ensuring a smooth user experience and timely processing of requests. Below are the key performance requirements, designed to guide the development process and ensure the platform meets the necessary standards.

1. **Request Processing Time**
   * **Requirement:** Any customer service request, such as submitting a tax payment or booking a public function hall, should be processed within **5 seconds** after submission.
   * **Rationale:** Quick processing ensures that users do not experience delays, improving overall satisfaction and system efficiency.
2. **System Uptime**
   * **Requirement:** The platform must maintain an uptime of **99.9%**, allowing for a maximum downtime of 8.76 hours per year.
   * **Rationale:** High availability is crucial for a public service platform to ensure that users can access services whenever needed.
3. **Response Time for Customer Inquiries**
   * **Requirement:** The platform should generate an initial automated response to customer inquiries within **2 seconds** of submission, with a follow-up by an employee within **1 business day**.
   * **Rationale:** Prompt responses enhance user engagement and trust, while ensuring that employees have sufficient time to provide thorough answers.
4. **Concurrent User Support**
   * **Requirement:** The platform must support at least **500 concurrent users** without a noticeable drop in performance, such as increased load times or errors.
   * **Rationale:** This ensures that the system can handle peak usage periods, such as during tax season or major public events, without impacting user experience.
5. **Data Retrieval Speed**
   * **Requirement:** Any search or retrieval of data (e.g., finding a specific request or viewing past service history) should take no longer than **3 seconds**.
   * **Rationale:** Fast data retrieval is critical for both customers and employees to quickly access the information they need, ensuring efficient service and decision-making.

## Safety and Security Requirements

## For the safety and security requirement, encryption and decryption techniques must be employed to protect sensitive data. This ensures that any data transmitted between users and the system, as well as data stored within the system, is secure and cannot be accessed or tampered with by unauthorized parties.

**Data Encryption**: All sensitive data, such as personal user information, details of service requests, and payment information, must be protected through encryption using the highest industry standards. This includes using Advanced Encryption Standard (AES-256) to secure data when it's stored (data at rest) and Secure Sockets Layer (SSL) or Transport Layer Security (TLS) protocols to protect data during transmission (data in transit). These measures ensure that private information remains secure and inaccessible to unauthorized parties, both when it is being stored in the system and when it is being transferred over networks.

**Data Decryption**: Only authorized users with the appropriate decryption keys should be able to access encrypted data. Decryption must happen securely, ensuring that the data is never exposed to unauthorized users or third parties during the process.

**End-to-End Encryption**: Communication between users and the platform must be protected with end-to-end encryption to prevent interception of data while it is being transmitted, ensuring confidentiality and integrity.

## Software Quality Attributes

# Reliability: The software must be reliable, ensuring it performs consistently under specified conditions without failure. This will be achieved through comprehensive automated testing to catch and fix bugs early.

# Maintainability: The software should be easy to maintain, allowing for efficient updates and modifications. This will be achieved by using a modular design to isolate different components, making it easier to update or replace parts of the system without affecting others.

# Usability: The software must be user-friendly, ensuring that users can easily learn and use it effectively. This will be achieved by designing an intuitive and responsive user interface with clear navigation and feedback mechanisms.

# Interoperability: The software should be able to interact with other systems and software seamlessly. This will be achieved by using standard communication protocols and data formats to facilitate integration with other systems.

# Portability: The software should be portable, allowing it to run on different platforms with minimal changes. This will be achieved by writing platform-independent code using cross-platform frameworks and libraries.

# Adaptability: The software should be adaptable, allowing it to accommodate changes in requirements or environments. This will be achieved by designing the system with flexible architecture and using configuration files to manage settings.

# Availability: The software must be available and operational at all times, minimizing downtime. This will be achieved by implementing redundancy and failover mechanisms to ensure continuous operation even in case of hardware or software failures.

# Correctness: The software should perform its intended functions accurately and without errors. This will be achieved by thorough testing, including unit tests, integration tests, and user acceptance tests, to ensure all functionalities work as expected.

# Flexibility: The software should be flexible, allowing it to be easily modified to meet new requirements. This will be achieved by using a modular design and adhering to coding standards that promote easy modification and extension.

# Reusability: The software components should be reusable in different contexts or projects. This will be achieved by designing components with clear interfaces and minimal dependencies, allowing them to be easily integrated into other systems.

## Robustness: The software should be robust, capable of handling unexpected inputs and conditions without crashing. This will be achieved by implementing thorough error handling and validation checks throughout the code.

# Testability: The software should be easy to test, allowing for efficient identification and resolution of defects. This will be achieved by writing testable code with clear interfaces and using automated testing tools to streamline the testing process.

# Scalability: The software should be scalable, able to handle increased loads and growing user demands without performance degradation. This will be achieved by designing the system with scalable architecture and using load balancing and distributed computing techniques.

# Security: The software must be secure, protecting against unauthorized access and data breaches. This will be achieved by implementing strong authentication and encryption methods, conducting regular security audits, and following best practices for secure coding.

# Performance: The software should perform efficiently, providing quick response times and optimal resource usage. This will be achieved by optimizing code, using efficient algorithms, and conducting performance testing to identify and address bottlenecks.

# Other Requirements

# Database Requirements: The system must use a relational database (e.g., MySQL, PostgreSQL) to store all user data, transactions, and logs.

# Internationalization Requirements: The system must support multiple languages, including but not limited to English, Spanish, French, and Mandarin.

# Legal Requirements: Ensure compliance with data protection regulations such as GDPR, CCPA, and other relevant laws.

# Reuse Objectives: Design software components to be reusable in other projects or contexts.

# Performance Requirements: The system should respond to user actions within 2 seconds under normal load conditions.

# Environmental Requirements: The system must operate reliably in various environments, including different operating systems (Windows, macOS, Linux) and devices (desktops, tablets, smartphones).

**Appendix A – Data Dictionary**

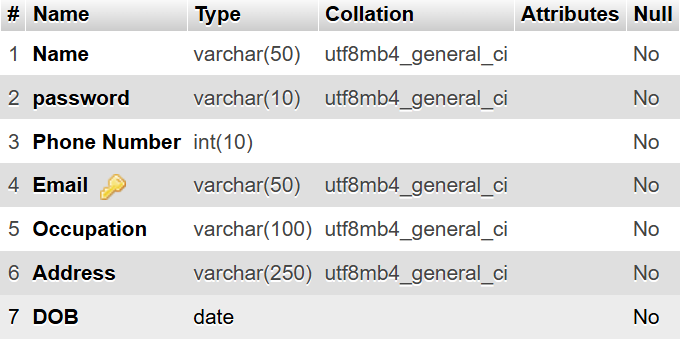
**

Table 1 Customer

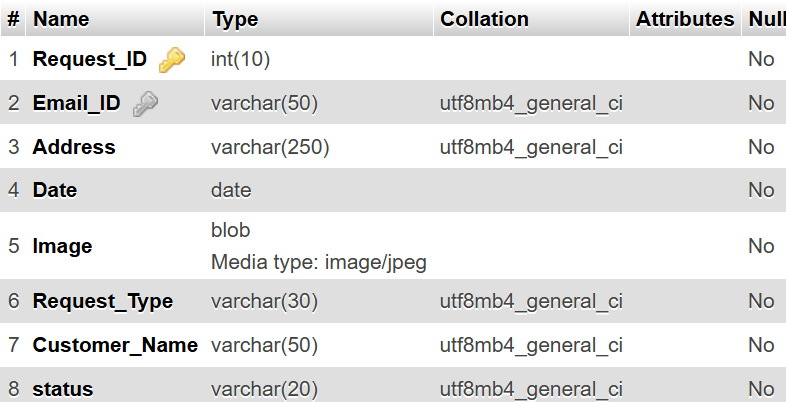


Table 2 Customer Service Request

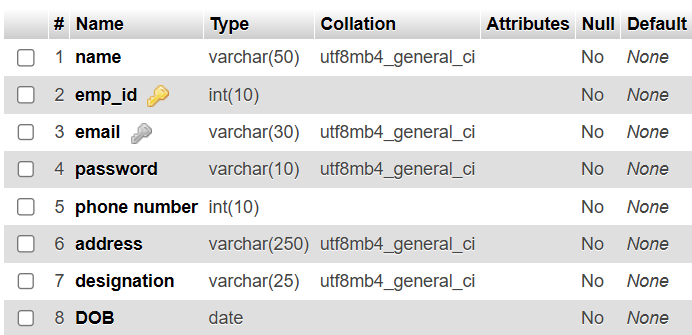


Table 3 Employee

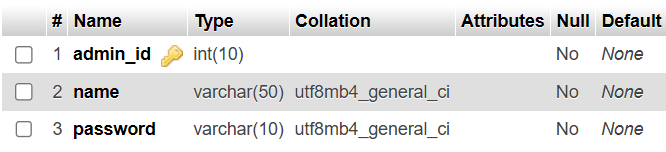


Table 4 Admin

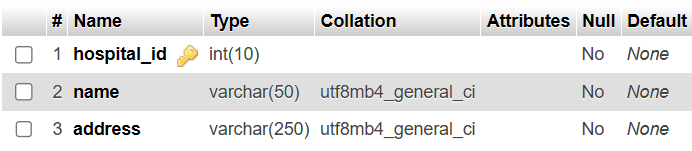


Table 5 Hospital

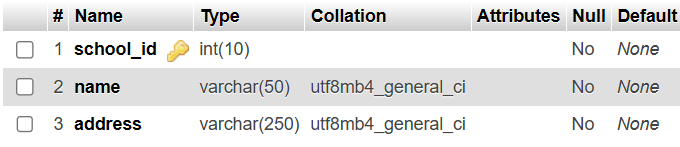


Table 6 School

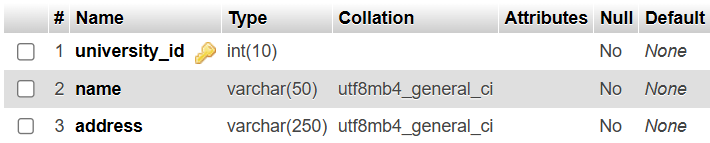


Table 7 University

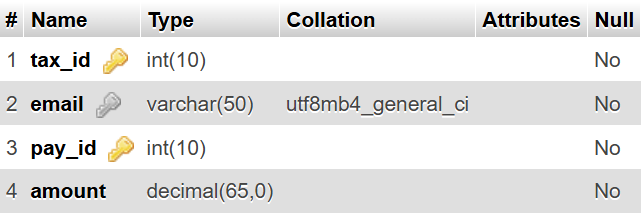


Table 8 Tax Payments

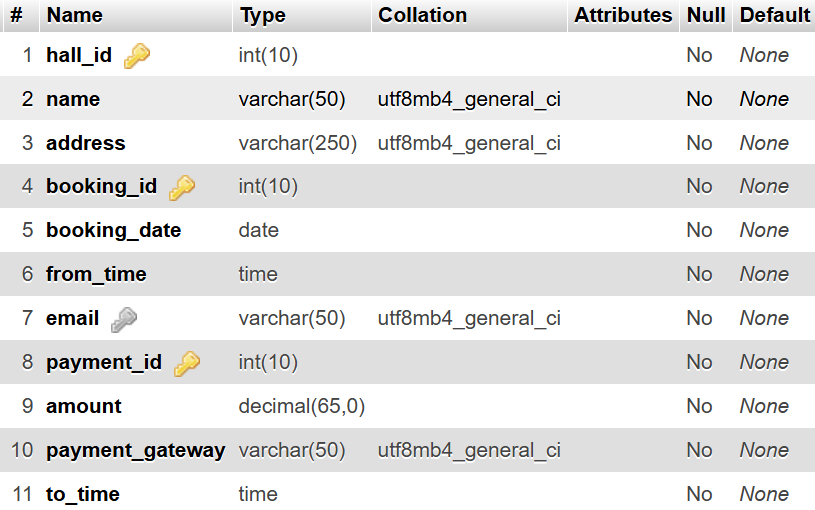


Table 9 Hall Booking

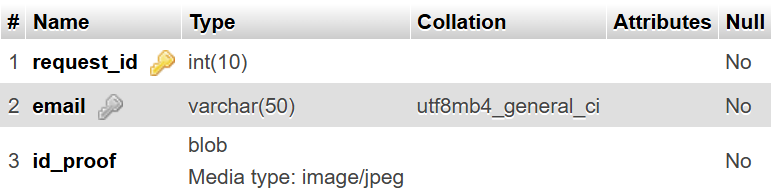


Table 10 Certificates

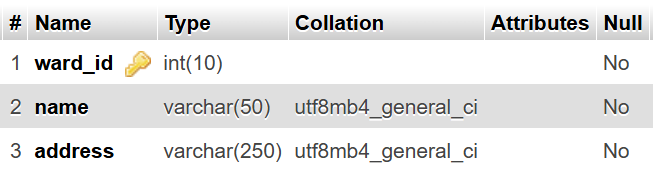


Table 11 Wards

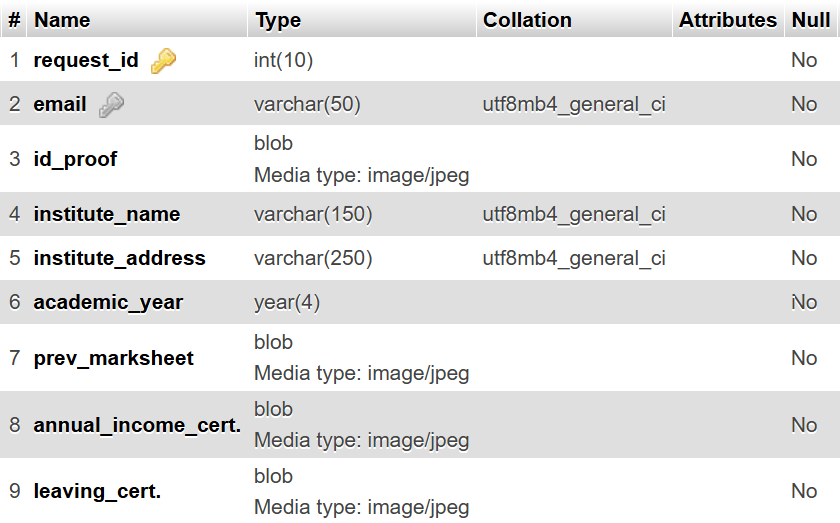


Table 12 Scholarships

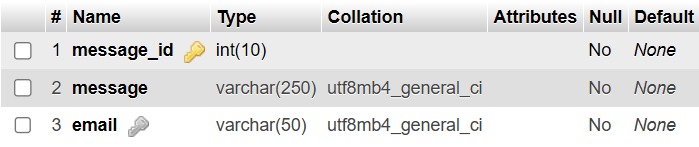


Table 13 Messages

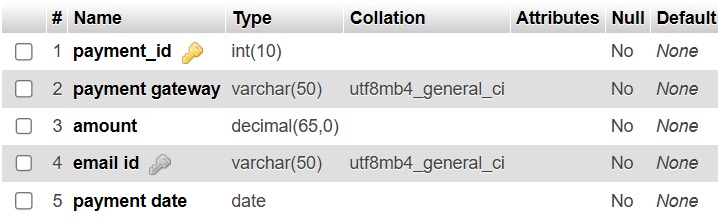


Table 14 Payments

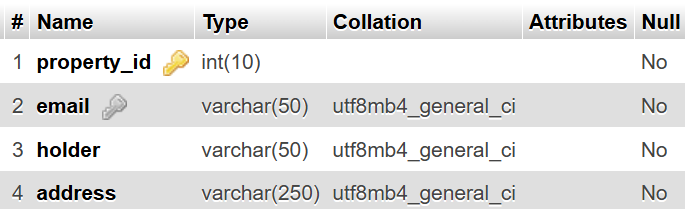


Table 15 property\_detail