**Project Deliverable 2 – Documentation and Coding**

**Faculty Name:** Information Technology

**Module Code:** ITPJA3-34

**Module Name:** Project

**Module Leader:** Ms Kariboba Mpwampu

**Copy Editor:** Mr. Kyle Keens

**Submission Date:** First Block Week 7

|  |  |
| --- | --- |
| **Group Name** | ILT Lads |
| **Group Number** |  |
| **Student Names** | Robert Njawaya, Russel Mlanga, AL Mujati, Tonderai Chadambuka |
| **Student Numbers** | qd54y35l2, t2vrf3cl9, 4fng5cdn3, h2cp1j9l3 |
| **Project Title** | MaryAnne Hair Salon System |
| **Submission Date** | 09/08/2023 |

Table of Contents

[2.1 Introduction 3](#_Toc145101212)

[2.2 Designing Diagrams 3](#_Toc145101213)

[Class Responsibility Collaborator (CRC) cards 4](#_Toc145101214)

[Database Design 6](#_Toc145101215)

[Entity Relationship Diagram (ERD) 7](#_Toc145101216)

[Activity Diagram: 8](#_Toc145101217)

[Context Diagram 14](#_Toc145101218)

[Sequence Diagram 15](#_Toc145101219)

[Data Flow Diagram (DFD) 23](#_Toc145101220)

[Use Case Diagram 24](#_Toc145101221)

[Security Risks 26](#_Toc145101222)

[2.3 Conclusion 29](#_Toc145101223)

[2.4 Sign-off 31](#_Toc145101224)

## Introduction

In this subsection, you should discuss an overview of how you plan to approach the design of your system. Here we should be able to see the flow of the system as it solves the client’s problem stated in Deliverable 1.

## Designing Diagrams

This section will have to do with showing the system in a diagrammatical form. State all the business rules and the assumptions you have based on your client’s system. These diagrams will help you know the beginning and the end of the system. Some of the diagrams to include are

### Class Responsibility Collaborator (CRC) cards

|  |  |
| --- | --- |
| Customer | Collaborators |
| Register an account.  Log into the system.  Make a booking appointment.  Access booking features.  View personal profile.  View discounts and rewards.  View personal booking history. | Booking System  Customer Dashboard  Rewards System |

|  |  |
| --- | --- |
| Admin | Collaborators |
| Add, update, delete content from the CMS database.  Manage pictures displayed on the website.  Add or update service offerings.  Create new Employees(Stylists/Nail Techs) | CMS (Content Management System)  Service Offerings Database  Employee Management System |

|  |  |
| --- | --- |
| Employee (Stylist/Nail Tech) | Collaborators |
| View daily appointments.  Set availability status.  Update your own profile (e.g., specialties, services offered).  Update own security login(password) | Booking System  Employee Dashboard  Security System, |

|  |  |
| --- | --- |
| Booking System | Collaborators |
| Handle customer bookings.  Display available slots for customers.  Notify employees about their appointments. | Customer  Employee |

|  |  |
| --- | --- |
| Customer Dashboard | Collaborators |
| Display customer's profile.  Show available discounts and rewards.  Display customer's booking history | Customer |

|  |  |
| --- | --- |
| Employee Dashboard | Collaborators |
| Display employee's daily appointments.  Show and update employee's availability.  Provide interface for profile update. | Employee |

|  |  |
| --- | --- |
| CMS (Content Management System) | Collaborators |
| Store and manage content for the website.  Provide interface for admins to manage content. | Admin |

|  |  |
| --- | --- |
| Rewards System | Collaborators |
| Track customer's earned rewards and discounts.  Update rewards and discounts based on customer actions. | Customer Dashboard  Customer |

|  |  |
| --- | --- |
| Service Offerings Database | Collaborators |
| Store information about available services.  Provide interface for Admins to add or modify services. | Admin |

|  |  |
| --- | --- |
| Security System | Collaborators |
| Handle security logins  Allow employees to update passwords | Employee  Employee Dashboard |

### Database Design

****

### Entity Relationship Diagram (ERD)

**A screenshot of a computer program

Description automatically generated**

**Dd**

Given the use of the SingleTable Inheritance (STI) model, which is inherently a simpler model, the need for an Enhanced Entity-Relationship Diagram (EERD) is reduced. Here are  few reasons as to why:

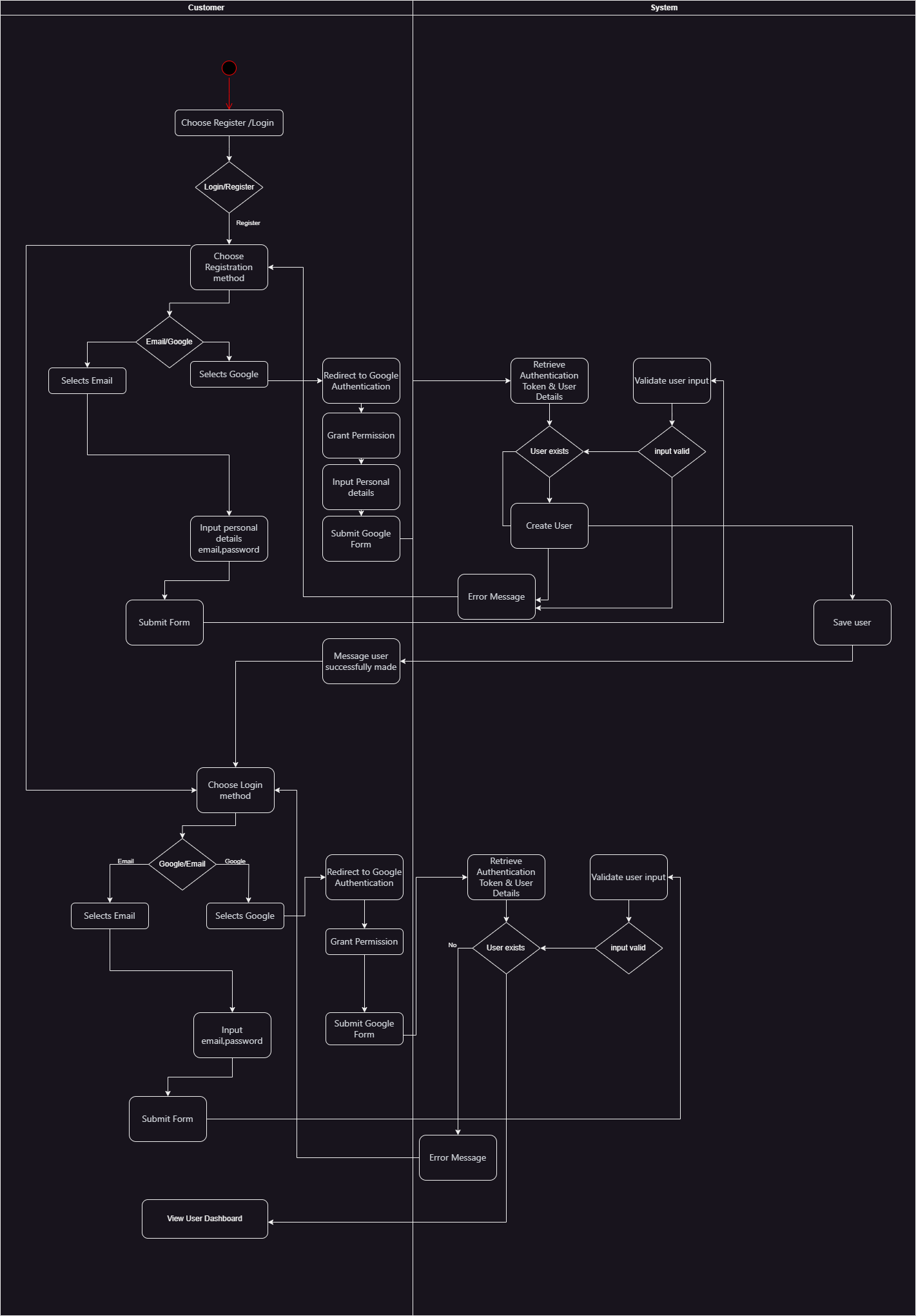
**First, No Specialization/Generalization**: One of the primary reasons to use an EERD is to represent specialization (subclasses) or generalization (superclasses) hierarchies, where subclasses inherit attributes and relationships from the superclass. In STI, this inheritance is implicit because all user roles are in the same table. Therefore there is no need to separate tables for each role, which might necessitate using EERD features.

**Second, No Categories/Union Types**: Another feature of EERDs is the ability to represent categories or union types, where an entity can be a part of one out of several entity sets. With the STI model, the different types of users are simply different rows in the same table, represented by different values in the **UserType** field.

**Third, No Complex Relationships**: The relationships outlined so far, between users, bookings, services, etc., are straightforward.There are no relationships between relationships or other complex structures that would require an EERD construct.

In conclusion, for the STI model and the relatively straightforward relationships that have previously been described, a standard ERD would suffice. As such we can reserve the use of the EERD for more complex scenarios where its additional constructs are necessary.

### Activity Diagram:

****

**A screenshot of a computer

Description automatically generated**

**A black screen with white lines and dots

Description automatically generated**

**A screenshot of a computer

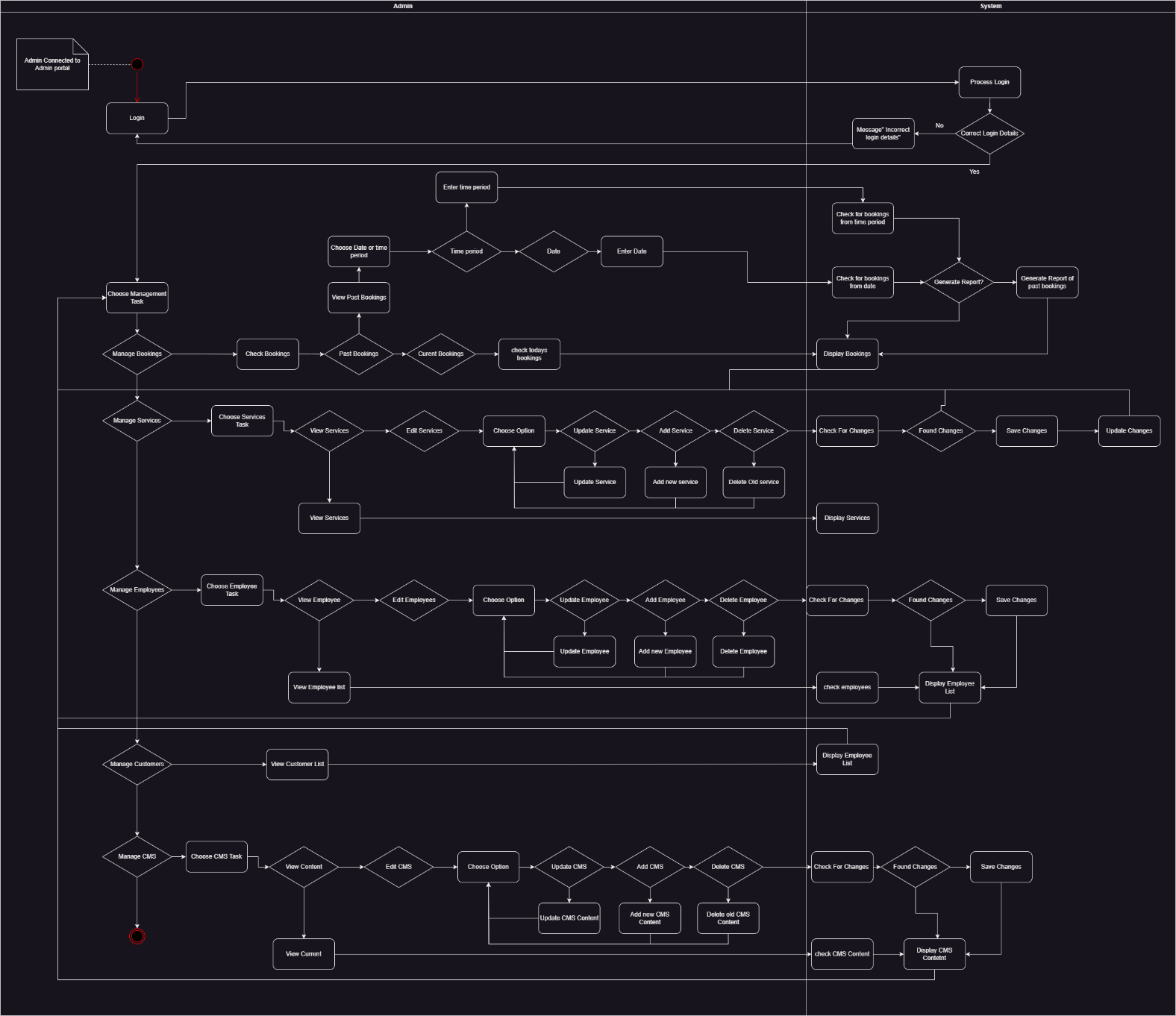
Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

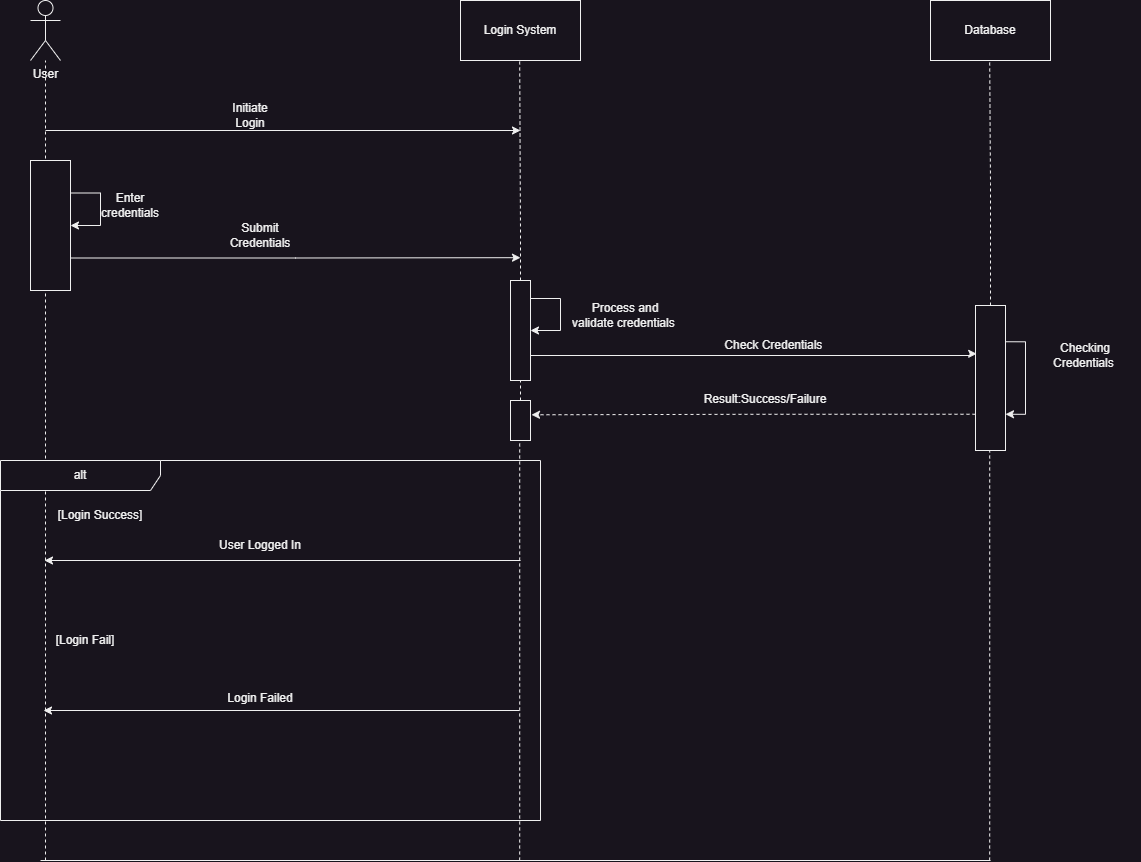
****

### Context Diagram

**A diagram of a diagram

Description automatically generated**

### Sequence Diagram

****

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer screen

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A diagram of a company

Description automatically generated with medium confidence**

**A screenshot of a computer screen

Description automatically generated**

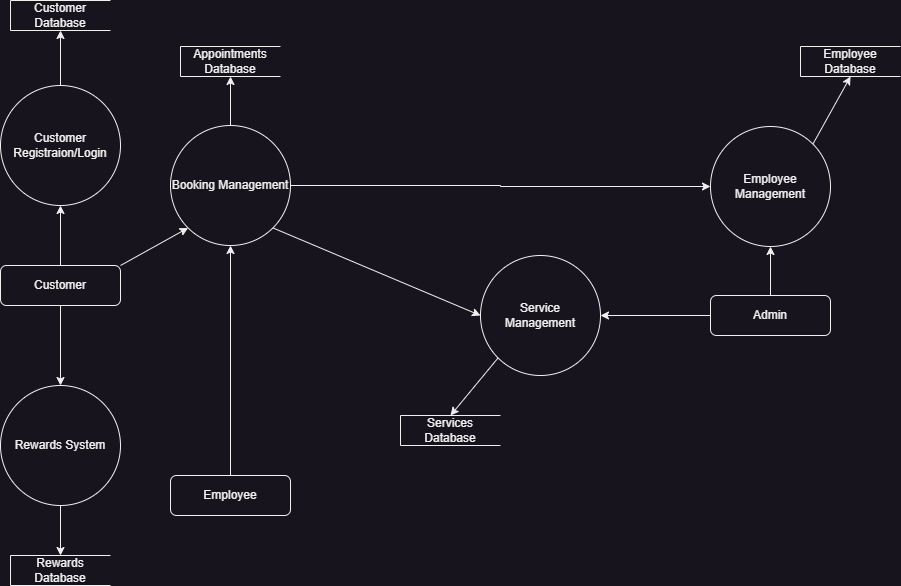
**A diagram of a system

Description automatically generated**

**A screen shot of a computer

Description automatically generated**

### Data Flow Diagram (DFD)



### Use Case Diagram

**A diagram of a system

Description automatically generated**

**Misuse Case Diagram**

**A diagram of a hair loss system

Description automatically generated with medium confidence**

### Security Risks

**1. Unauthorized Access / Account Hijacking**

Risk: Attackers or malicious users could gain unauthorized access to an account (customer, employee, or admin) and misuse the privileges.

Mitigation:

* Implement strong password policies.
* Use multi-factor authentication (MFA) for added security, especially for admin accounts.
* Set up account lockouts after several failed login attempts.
* Monitor for unusual login patterns and set up alerts.

**2. Data Breach**

Risk: Confidential data (like customer personal details, appointment details, or payment info) could be exposed or stolen.

Mitigation:

* Use data encryption both in transit (with protocols like TLS) and at rest.
* Conduct regular backups and ensure backup security.
* Grant data access strictly on a need-to-know basis.
* Undertake periodic vulnerability assessments and penetration tests.

**3. Injection Attacks (e.g., SQL Injection)**

Risk: Attackers could exploit input fields or queries to inject malicious commands or scripts, potentially gaining unauthorized data access or causing other harmful effects.

Mitigation:

* Utilize parameterized queries and prepared statements.
* Validate and sanitize all inputs to the system.
* Use web application firewalls (WAFs) to detect and prevent suspicious activities.

**4. Denial of Service (DoS) or Distributed Denial of Service (DDoS)**

Risk: Attackers could flood the system with bogus requests, making the system unavailable for legitimate users.

Mitigation:

* Leverage DDoS protection and mitigation tools or services.
* Regularly patch and update systems to address known vulnerabilities.
* Monitor traffic for unusual patterns that could signal an attack.

**5. Cross-Site Scripting (XSS)**

Risk: Malicious scripts could be injected into web pages, which are then executed by another end user's browser.

Mitigation:

* Sanitize and validate all user inputs.
* Implement content security policies to block unauthorized script execution.
* Keep web frameworks and libraries updated.

**6. Fake Appointments**

Risk: Malicious users or competitors could flood the system with fake appointments, blocking genuine users and disrupting operations.

Mitigation:

* Introduce CAPTCHAs or similar verification tools during the booking process.
* Limit the number of bookings a non-verified user can make.
* Implement anomaly detection to recognize suspicious booking patterns.

**7. Insider Threats**

Risk: Employees or insiders with malicious intent or negligence could misuse their access.

Mitigation:

* Grant minimum required privileges (principle of least privilege).
* Monitor system usage for irregularities.
* Provide regular security training to employees.

**8. Man-in-the-Middle (MitM) Attacks**

Risk: Attackers could intercept and potentially alter communication between the user and the system.

Mitigation:

* Enforce HTTPS for all web traffic.
* Ensure secure and updated SSL/TLS configurations.
* Educate users about risks, especially when connecting via public or unsecured Wi-Fi.

**9. Physical Security Breaches**

Risk: The physical infrastructure (servers, storage devices) could be accessed, leading to data theft or system tampering.

Mitigation:

* Secure physical locations with measures like biometric access, surveillance cameras, and security personnel.
* Restrict access only to essential personnel.

**10. Outdated Software and Systems**

Risk: Outdated software could have vulnerabilities that attackers can exploit.

Mitigation:

* Implement a regular update and patch management process.
* Monitor for new vulnerabilities in third-party software or libraries and apply patches as soon as they're available.

## Conclusion

**1. Entity-Relationship Diagram (ERD):**

* **System Structure**: It provided an understanding of how data is structured within the system, showcasing the relationships between different data entities.
* **Database Design**: ERD was crucial in designing the base database, ensuring that data is stored efficiently and can be queried effectively.
* **Relationships**: Demonstrated how different entities relate to one another, like how a customer relates to bookings or how services relate to employees.

**2. Use Case Diagram:**

* **User Interaction**: Gave a clear picture of how different users (actors) interact with the system and what functionalities are available to them.
* **Functional Requirements**: Offered a snapshot of the system's primary functions, which is crucial for both development and testing.
* **System Scope**: Helped in identifying the boundaries of the system and its interactions with external entities.

**3. Misuse Case Diagram:**

* **Threat Awareness**: Highlighted potential malicious activities or unintended actions that could be carried out against the system.
* **Security Planning**: By identifying areas of vulnerability, it provided direction on where to implement security measures.

**4. Activity Diagram:**

* **Process Flow**: Offered a step-by-step visualization of how specific functionalities or features would work from start to finish.
* **Decision Points**: Highlighted areas where decisions or choices could affect the flow of an operation, such as choosing between logging in via email or Google.

**5. Sequence Diagram:**

* **Interactions Over Time**: Showcased how different system components interacted with one another over time for specific operations.
* **System Responses**: Helped in understanding the system's responses or actions for user requests, providing a clear flow of events.

**6. Data Flow Diagram:**

* **Data Movement**: Demonstrated how data moves between different parts of the system, including external entities, data stores, and processes.
* **Data Processing**: Gave insights into where and how data gets processed or transformed within the system.

**7. Context Diagram:**

* **System Overview**: Presented a high-level view of the system and its interactions with the outside world, helping all the stakeholders involved get an understanding of the system's core essence without getting into deep details.

Each of these diagrams serves a distinct purpose and, when combined, offers a comprehensive understanding of the MaryAnne Hair Salon system. They did facilitate a better understanding for stakeholders, developers, designers, in our team.

## Sign-off

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Mary Anne Hair Salon, Project Client  Date: \_\_\_09/08/2023\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |  | Robert Njawaya, Project Manager  Date: 09/08/2023\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |