**asif**

**American International University-Bangladesh (AIUB)**  
 Department of Computer Science

Faculty of Science & Technology (FST)

**PROJECT TITLE**

**Semester: Fall 25-26**

|  |  |  |
| --- | --- | --- |
| ***Group:*** | | ***Section:*** |
| **SL** | **Student Name** | **Student ID** |
| 1 | MD.ASHIF HOSSAIN | 23-55082-3 |
| 2 | MAHFUJUR RAHAMAN | 23-50498-1 |
| 3 | RITTIK CHANDRA DEY | 23-53485-3 |
| 4 | MD. ABDUL ALIM | 23-51105-1 |
| 5 | MST. TAZMUN NAHER AKHE | 23-50845-1 |

Table of Contents

[1. Project Proposal xx](#_Toc214214201)

[1.1 Background to the Problem](#_Toc214214203) xx

[1.2 Selection of Process Model xx](#_Toc214214204)

[2. Software Requirements Specification (SRS) / PRD xx](#_Toc214214202)

[2.1 Scopes and Features xx](#_Toc214214203)

[2.2 User Story Table xx](#_Toc214214204)

[2.3 Requirements Traceability Matrix xx](#_Toc214214205)

[2.3.1 Functional Requirements xx](#_Toc214214212)

[2.3.2 Non-Functional Requirements xx](#_Toc214214216)

[3. Software Design xx](#_Toc214214202)

[3.1 System Deisgn xx](#_Toc214214203)

[3.2 UI Deisgn using Figma xx](#_Toc214214203)

[4. Git Workflow xx](#_Toc214214202)

[5. Software Testing xx](#_Toc214214202)

[6. Conclusion xx](#_Toc214214202)

**1. PROJECT PROPOSAL**

* **Background to the Problem**

In modern households, getting essential home services such as electrical repair, cleaning support, or gardening help often becomes slow and inconvenient. People usually depend on manual searching or making multiple phone calls, which wastes time and causes delays in urgent situations. There is no proper tracking system, no organized communication, and no instant request system.

To solve these issues, our group is developing a **Smart Doorbell Service System**, where a user can simply press a doorbell or open the system interface to request **Electrical**, **Cleaning** or **Gardening** services instantly. The system notifies the service provider immediately and stores the request in the database. This makes service access faster, organized, and more efficient.

* 1. **Selection of Process Model**

**Process Model Selection (Scrum)**

The Scrum model is chosen because the project involves multiple features and user roles, and the requirements may change during development. Scrum supports an incremental and flexible development process through short sprints, allowing regular feedback and continuous improvement. This approach helps the team manage complexity, adapt to changes easily, and deliver working features on time within a team-based academic environment.

**Reason for Using the Scrum Model**

The Scrum model is chosen for the DoorBell project because it is well suited for developing a complex system in a structured and flexible way. The DoorBell system includes many features such as user registration and login, service booking, payment processing, profile management, reviews, and an admin control panel. Developing all these features at once would be difficult, so Scrum helps by breaking the project into smaller parts called sprints. Each sprint focuses on completing a set of features, making the development process more organized and manageable.

The project also involves multiple user roles, including general users, service providers (employees), and an admin. Each role has different requirements and functionalities. Scrum allows the team to prioritize these requirements in a product backlog and implement them step by step. This makes it easier to handle role-based features and ensure that all users get the correct functionality.

Another important reason for using Scrum is that project requirements can change during development. In academic projects, feedback from teachers or supervisors may require changes or new features. Scrum is flexible and supports change, allowing the team to update the backlog and include new requirements in the next sprint without affecting completed work.

Scrum also improves teamwork and communication. Since the project is developed by a team within a limited semester timeframe, Scrum helps divide tasks clearly and track progress regularly. Each sprint produces a working version of the system, which helps identify errors early and improve overall quality. For these reasons, the Scrum model is an effective and practical choice for the DoorBell project.

## 2. SOFTWARE REQUIREMENTS SPECIFICATIONS (SRS) / PRODUCT REQUIREMENTS DOCUMENT (PRD)

**2.1 Scopes and Features**

**2.2 User Story Table**

**2.3 Requirements Traceability Matrix**

2.3.1 Functional Requirements

1. **User Related Functional Requirements**
2. The system shall allow a new user to create an account using phone number or email address.
3. The system shall validate user signup information (unique email/phone, minimum password length, user type selection).
4. The system shall allow registered users to log in securely using valid credentials.
5. The system shall provide a password recovery option for users who forget their password.
6. The system shall allow users to view and update their profile information.
7. The system shall allow users to change their password and profile picture.
8. The system shall allow users to log out securely from the system.
9. **Service Request Functional Requirements**
10. The system shall display available service categories (Electrical, Cleaning, Gardening).
11. The system shall allow users to view detailed information of each service.
12. The system shall allow users to request a service through the system interface or doorbell trigger.
13. The system shall store all service requests in the database.
14. The system shall notify the assigned service provider instantly after a service request is made.
15. The system shall show booking status (Pending, Accepted, Completed).
16. **Admin & Employee Functional Requirements**
17. The system shall allow admin to manage service categories and services.
18. The system shall allow admin to view and manage users, employees, and bookings.
19. The system shall allow employees to view assigned work and booking details.
20. The system shall allow employees to track completed and pending services.
21. The system shall allow admin to view transaction and payment records.
22. **Payment Related Functional Requirements**
23. The system shall allow users to choose a payment method for service booking.
24. The system shall confirm booking after successful payment.
25. The system shall display error messages if payment fails.

**2.3.2 Non-Functional Requirements**

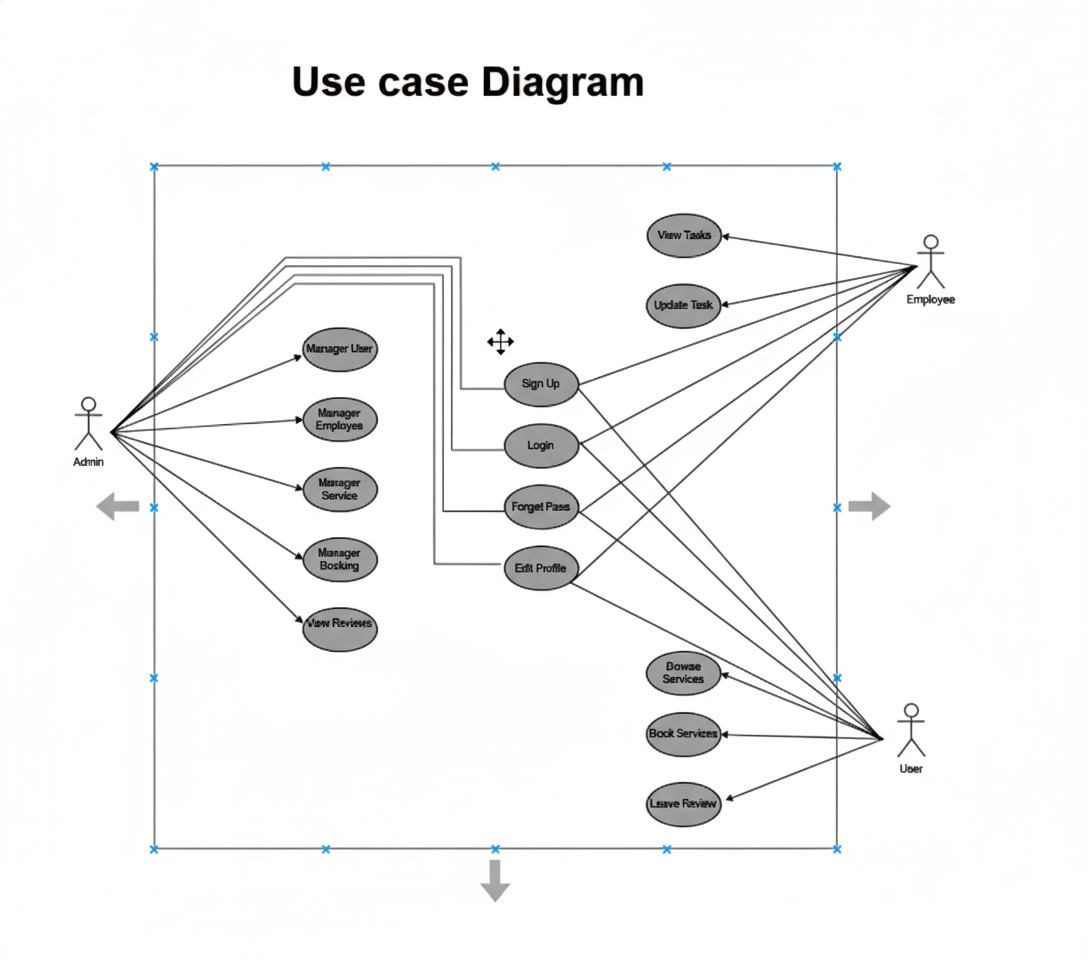
1. **Performance Requirements**
2. The system shall respond to user requests within an acceptable time.
3. The system shall handle multiple service requests simultaneously without performance degradation.
4. **Usability Requirements**
   1. The system interface shall be simple, clean, and user-friendly for all users.
   2. The system shall provide easy navigation between services, bookings, and profile pages.
5. **Security Requirements**
6. The system shall protect user data using secure authentication and authorization mechanisms.
7. The system shall not allow unauthorized access to admin or employee features.
8. **Availability Requirements**
9. The system shall be available 24/7 except during scheduled maintenance.
10. **Reliability Requirements**
11. The system shall store all service requests, bookings, and transactions accurately.
12. The system shall ensure no data loss during unexpected system failures.
13. **Scalability Requirements**
14. The system shall support future expansion such as adding new service categories.

|  |  |
| --- | --- |
| **1.0** | **Usability** |
| **1.0.1** | Role-based intuitive UI (Admin, Employee, User) |
| **1.0.2** | Clear navigation and consistent design |
| **1.0.3** | Meaningful error messages with corrective steps |
| **2.0** | **Reliability** |
| **2.0.1** | Data consistency in bookings and transactions |
| **2.0.2** | Daily automated backup of database |
| **2.0.3** | System recovery within 30 minutes after failure |
| **3.0** | **Security** |
| **3.0.1** | Secure authentication for all user types |
| 3.0.2 | Encrypted password storage |
| 3.0.3 | Role-based access control (RBAC). |
| **3.0.4** | Account lock after 5 failed login attempts |
| **4.0** | **Maintainability** |
| 4.0.1 | Modular design for easy updates |
| 4.0.2 | Editable configuration without code change |
| **5.0** | **Scalability** |
| **5.0.1** | Support 10,000 registered users |
| 5.0.2 | Handle 1,000 daily transactions |
| 6.0 | **Performance** |
| 6.0.1 | Dashboard load within 3 seconds |
| 6.0.1 | Booking confirmation within 5 seconds |
| 6.0.2 | Search results within 2 seconds for 50k records |
| 7.0 | **Availability** |
| 7.0.1 | 99.5% uptime during business hours |
| 7.0.2 | Maintenance outside service hours |
| **8.0** | **Scalability** |
| **8.0.1** | Support 10,000 registered users |
| **8.0.2** | Handle 1,000 daily transactions |
| **9.0** | **Data Storage** |
| **9.0.1** | Store 5 years of transaction history |
| **9.0.2** | Retrieve archived data within 10 seconds |

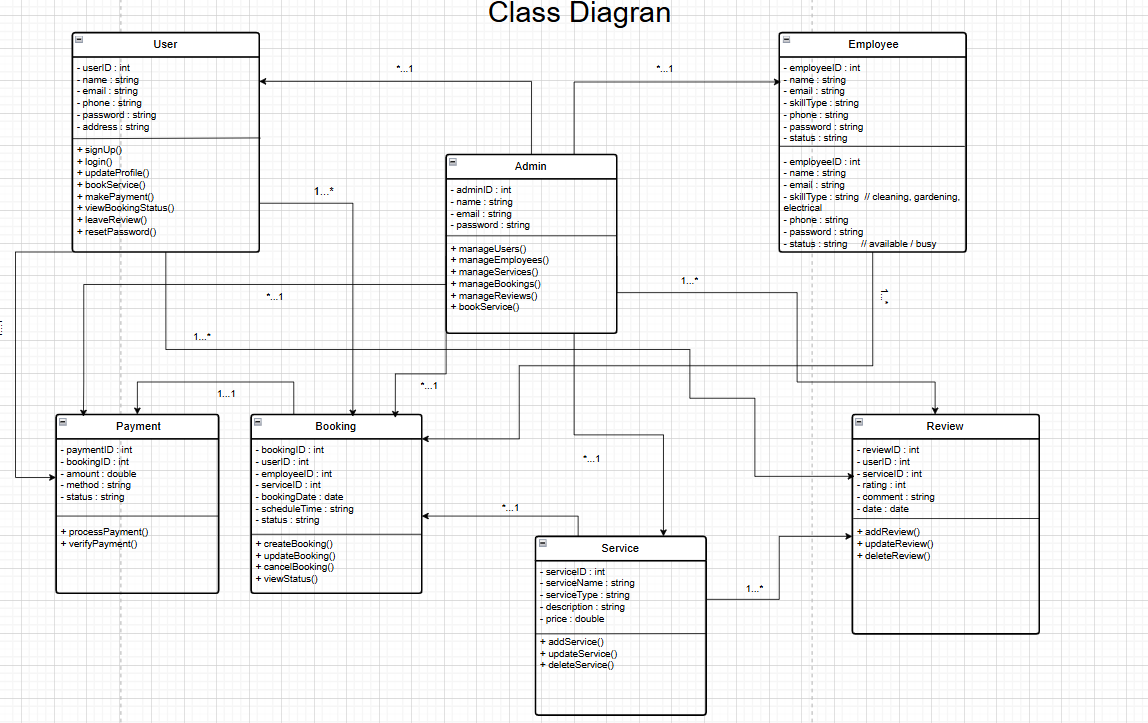
# 3. SOFTWARE DESIGN

**3.1 System Design:** Draw the system design for your project using **Draw.io** or **LucidChart**.

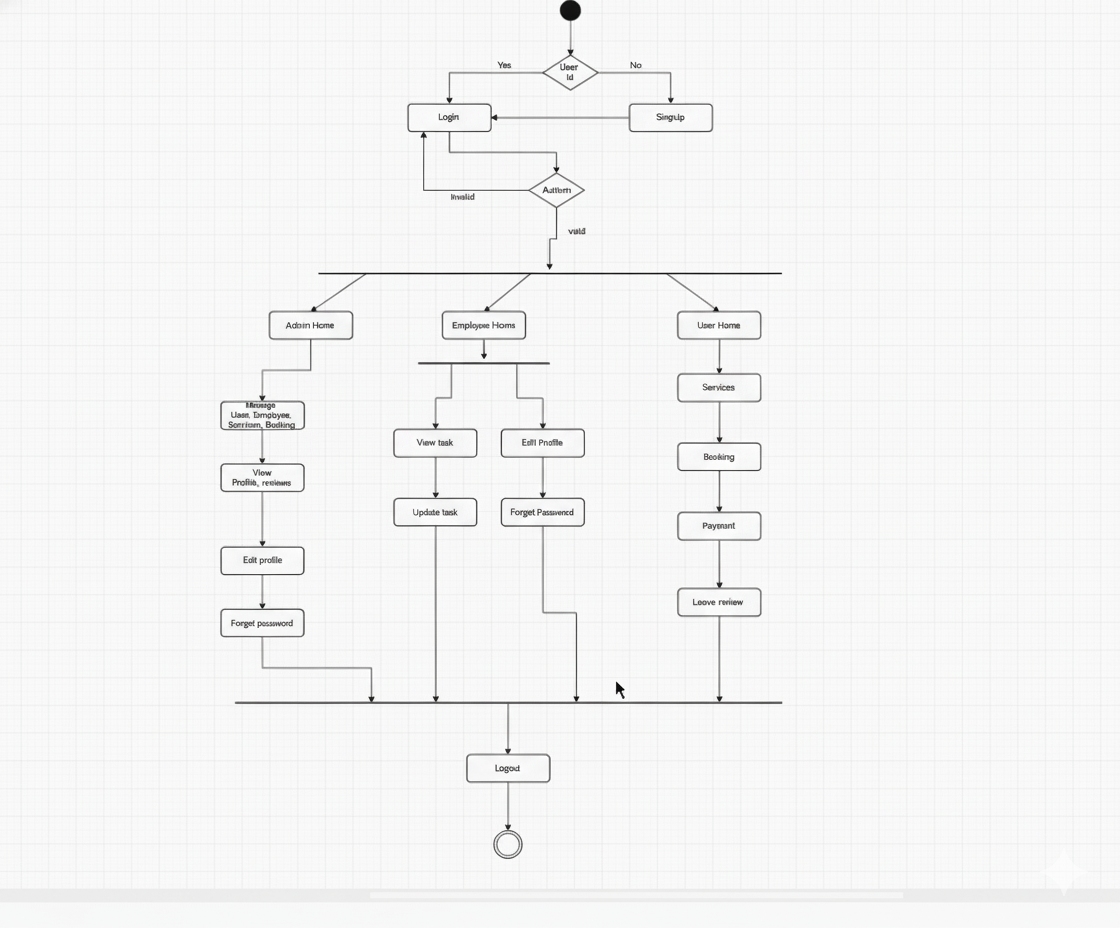
* Prepare a **Use Case Diagram** by first defining all users (actors) and their roles. Show each actor’s interactions with the system through use cases inside a system boundary. Include relationships like include or extend where needed.



* Prepare a **Class Diagram** by identifying the main classes from your project. Add attributes and operations for each class, and show associations, generalizations, aggregations, or compositions between them.

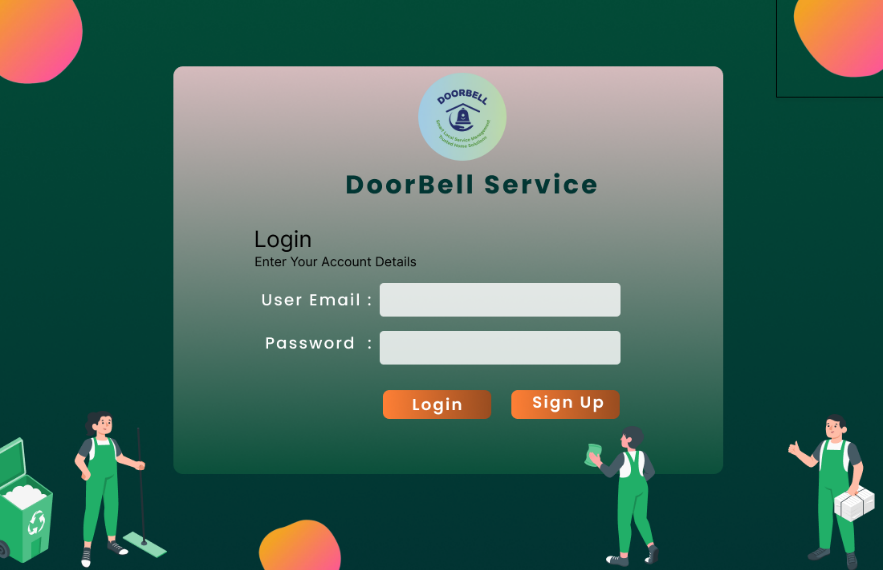
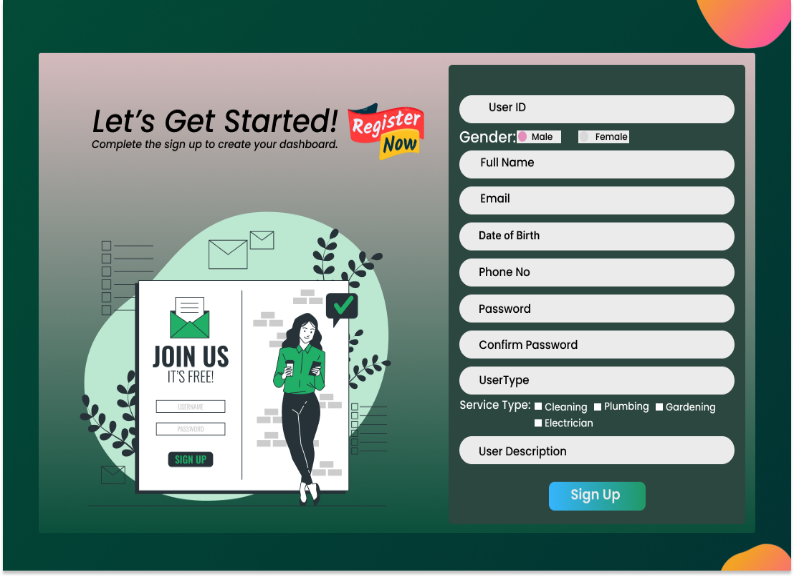


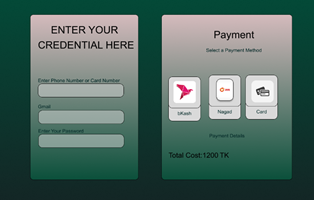
* Prepare an **Activity Diagram** that visually represents the workflow of a system or process.

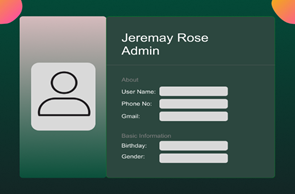
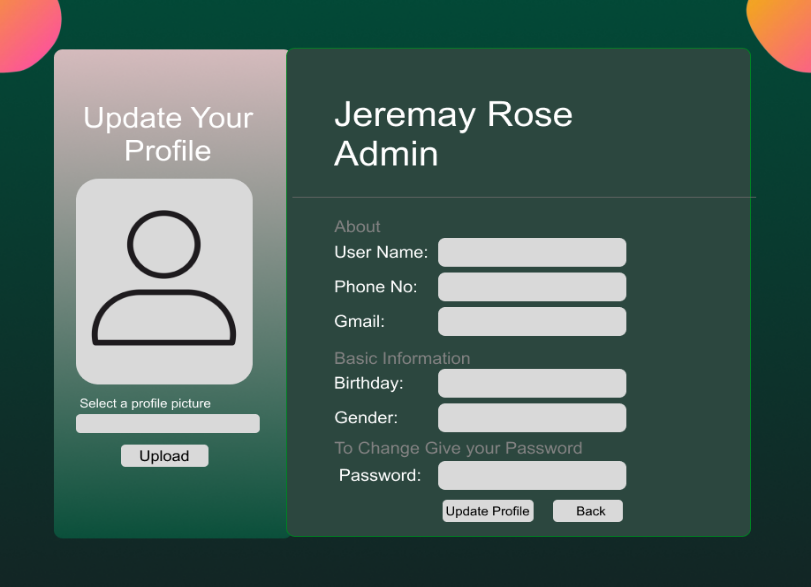


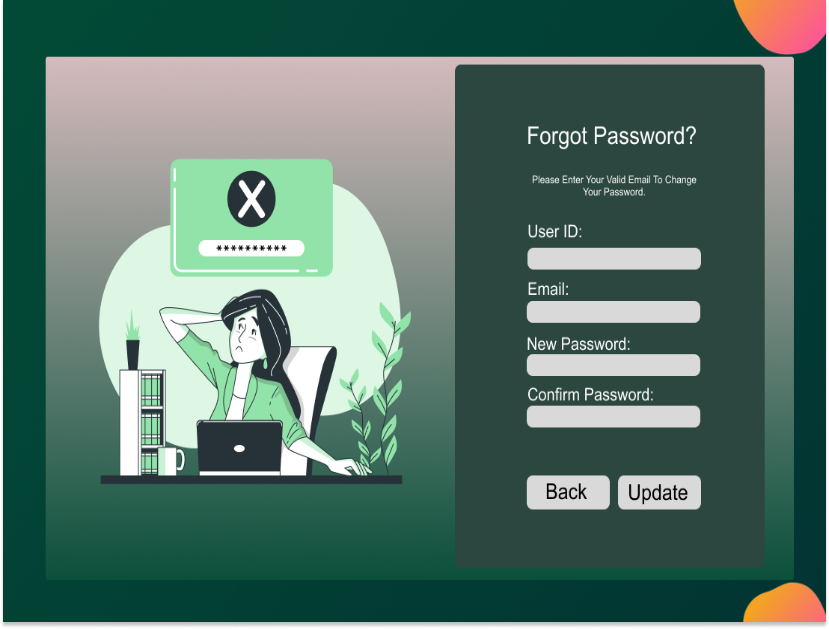
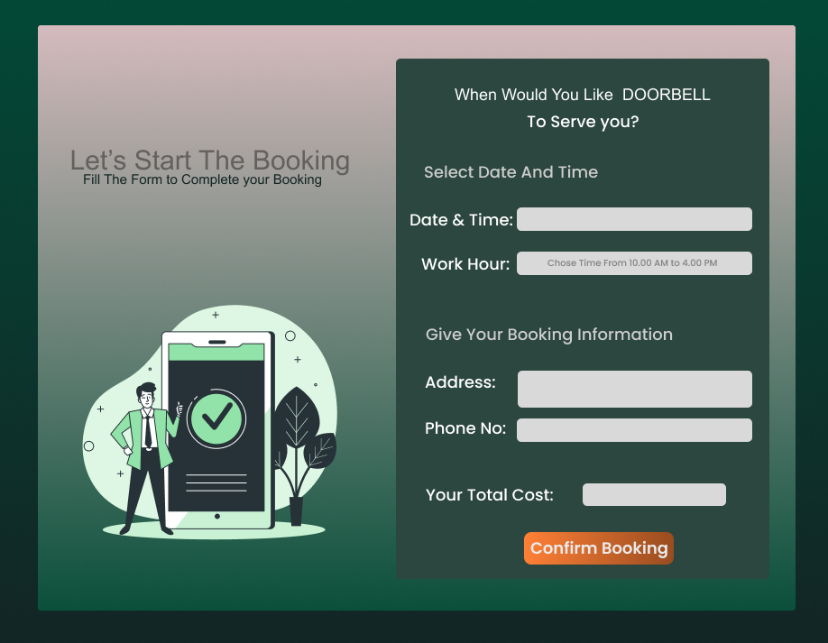
## UI / Wireframe Design using Figma

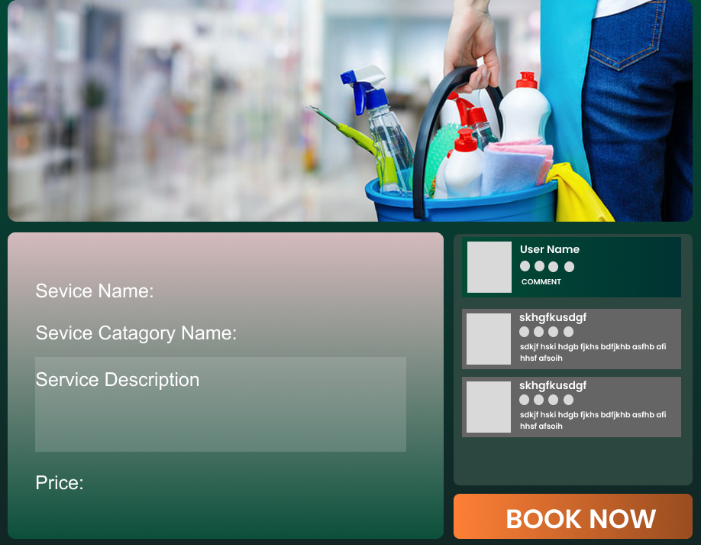
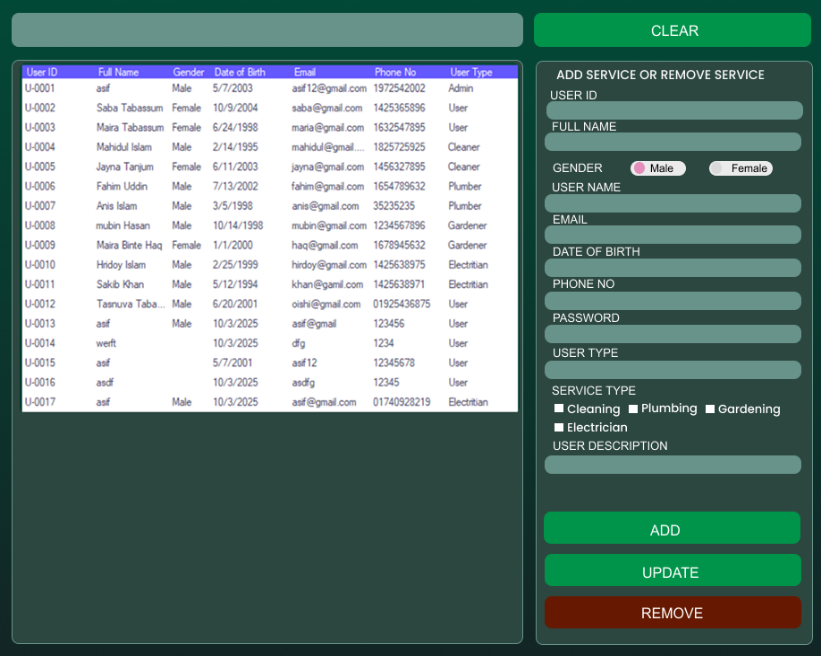
* Build the wireframes directly from your **SRS**. Each functional requirement defined in the SRS should have a corresponding screen or component in the UI.
* Use the **Trello user board** (or equivalent task board) as a reference to decide which features and user flows need to be represented in the prototype.
* Create clickable wireframes that connect the main screens together, showing how a user will navigate through the system.
* Ensure the wireframe matches the actors, roles and functionalities described in the SRS and system design diagrams.
* Export the prototype and include screenshots or a link to your design as part of the report submission.

# 4. GIT WORKFLOW

* Create a central repository for the project on GitHub and set the **master (or main) branch** as the primary branch for integration.
* Each member should clone the repository and create their own **feature branches** for assigned tasks. Work on new features or fixes within these branches.
* Add files, stage them and commit changes with clear messages that describe the purpose of each update.
* Push commits from the feature branches to the remote repository so other members can see progress.
* Use **pull** to fetch and integrate changes from the remote repository into local copies, ensuring everyone stays updated.
* Merge feature branches into the **master/main branch** only after the work is tested and reviewed, resolving any conflicts that occur.
* Show evidence of collaboration by maintaining a clear commit history (using logs) with multiple commits, merges and contributions from all group members.
* Keep the repository organized with a clean history that tracks the project workflow from initialization to completion.

# 5. SOFTWARE TESTING

* Identify some testing methods that you want to use in the testing phase later for your project.
* Prepare 3 **test cases** using a manual test case template which template taught you in the class. Suppose if you have total 4 members, then the total test cases will be 12.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Doorbell | | | Test designed by : mahfuj | | |
| Test case : | | | This designed date : 23-12-2025 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by : Ashif | | |
| Module Name : User Registration | | | Test Execution date : 05-01-2026 | | |
| Test title : Verify service booking with valid date, time, and user details | | | | | |
| Description: Test the user sign-up functionality on the DOORBELL application | | | | | |
| Precondition: 1. User is on the registration page. 2. Internet connection is available. | | | | | |
| Dependencies: None | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Go to Sign Up page | \_\_ | Registration page should load successfully | | As expected | Pass |
| 2. Enter User ID | Asif23 | User ID should be accepted | | As expected | Pass |
| 3. Select Gender | Male | Gender option should be selectable | | As expected | Pass |
| 4. Enter Full Name | Asif Hossain | Full name should be accepted | | As expected | Pass |
| 5. Enter Email | asif@gmail.com | Valid email should be accepted | | As expected | Pass |
| Select Date of Birth | 01/01/2002 | Date of birth should be selected | | As expected | Pass |
| 7. Enter Phone No | 0171234567 | Phone number should be accepted | | As expected | Pass |
| 8. Enter Password | 12345 | Password should be masked | | As expected | Pass |
| 9. Enter Confirm Password | 12345 | Passwords should match | | As expected | Pass |
| 10. Select User Type | Customer | User type should be selected | | As expected | Pass |
| 11. Select Service Type | |  | | --- | |  |  |  | | --- | | Cleaning | | Service type should be selectable | | As expected | Pass |
| 12. Enter User Description | Home service user | Description should be accepted | | As expected | Pass |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Project Name:**Doorbell | | | Test Designed by: Tazmun Naher akhe. | | |
| **Test case ID:**TC\_FM\_01 | | | Test Designed date:06/01/2026 | | |
| **Test Priority(Low,Medium,High):**Medium | | | Test Executed by: | | |
| **Module Name:** Payment Method | | | Test Execution date: 22/08/2025 | | |
| **Test Title:**Verify adding payment method with valid details | | |  | | |
| **Description:**Test the payment method add/update functionality | | |  | | |
| **Precondition:**User in logged in | | | | | |
| **Dependencied**:If any | | | | | |
| **Test Steps** | **Test Data** | **Expected Results** | | **Actual Results** | **Status(Pass/Fail)** |
| Open Payment Method Page | Logged-in user | Payment method page open successfully | | Successfull | Pass |
| Click Add Fayment Method | \_ | Payment form opens | | Successfull | Pass |
| Enter Payment type | Card/Mobile Banking | Payment type accepted | | Successfull | Pass |
| Enter card/mobile details | Valid details | Details validated | | Error | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project Name: Doorbell | | Text designed by : | | |
| Test case : | | This designed date : | | |
| Test Priority (Low, Medium, High): High | | Test Executed by : | | |
| Module Name : User Management / Service Management | | Test Execution date : | | |
| Test title : Verify User Management panel add, update, remove and service assignment | | | | |
| Description: Test the User Management panel to ensure user data listing, add, update, remove, and service assignment functionalities work correctly. | | | | |
| Precondition: 1. Admin is logged into the DOORBELL system  2. User Management panel is accessible | | | | |
| Dependencies: None | | | | |
| Test Steps | Test Data | Expected Results | Actual Results | Status (Pass/Fail) |
| 1. Open User Management panel | \_\_ | User table should load successfully | As expected | Pass |
| 2. Verify user table columns | User ID, Name, Gender, DOB, Email, Phone, User Type | All columns should be visible | As expected | Pass |
| 3. Select a user from table | U-0004 | Selected user data should populate form fields | As expected | Pass |
| 4. Enter new user details | Valid data | All input fields should accept data | As expected | Pass |
| 5. Select Gender | Male / Female | Only one gender should be selectable | As expected | Pass |
| 6. Select User Type | Admin / User / Cleaner / Plumber etc. | User type should be selected correctly | As expected | Pass |
| 7. Select Service Type | Cleaning, Plumbing | Multiple services should be selectable | As expected | Pass |
| 8. Click ADD button | --- | New user should be added to the table | As expected | Pass |
| 9. Update existing user data | Change phone no | User data should be updated | As expected | Pass |
| 10. Click UPDATE button | \_\_\_ | Updated data should reflect in table | As expected | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project Name: Doorbell | | Text designed by : | | |
| Test case : | | This designed date : | | |
| Test Priority (Low, Medium, High): Medium | | Test Executed by : | | |
| Module Name : Service Booking | | Test Execution date : | | |
| Test title : Verify service booking with valid date, time, and user details | | | | |
| Description: Test the service booking form on the DOORBELL application | | | | |
| Precondition: 1. User is on the booking page.   2. User has selected a service. | | | | |
| Dependencies: None | | | | |
| Test Steps | Test Data | Expected Results | Actual Results | Status (Pass/Fail) |
| 1. Go to booking page | \_\_ | Booking page should load successfully | As expected | Pass |
| 2. Select Date & Time | 08/12/2025  Time: 10:00 AM | Date and time should be accepted | As expected | Pass |
| 3. Select Work Hour | 10:00 AM – 4:00 PM | Work hour should be selected | As expected | Pass |
| 4. Enter Address | Dhaka, Bangladesh | Address should be entered successfully | As expected | Pass |
| 5. Enter Phone No | 01723456778 | Phone number should be accepted | As expected | Pass |
| 6. Verify Total Cost | 1300 BDT | Correct total cost should be displayed | As expected | Pass |
| 7. Click Confirm Booking | \_\_ | Booking should be confirmed and success message shown | As expected | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project Name: Doorbell | | Text designed by : | | |
| Test case : | | This designed date : | | |
| Test Priority (Low, Medium, High): Medium | | Test Executed by : | | |
| Module Name : Home Panel | | Test Execution date : | | |
| Test title : Verify Home Panel UI and service navigation | | | | |
| Description: Test the Home Panel of the DOORBELL application to verify service categories visibility and navigation. | | | | |
| Precondition: 1.User has successfully opened the DOORBELL application 2. User is logged in (if required) | | | | |
| Dependencies: None | | | | |
| Test Steps | Test Data | Expected Results | Actual Results | Status (Pass/Fail) |
| 1.Open the application | \_\_ | Home panel should load successfully | As expected | Pass |
| 2. Verify Home Panel title | Home Panel | “Home Panel” title should be visible | As expected | Pass |
| 3. Verify Electrical service card |  | Electrical service card should be displayed | As expected | Pass |
| 4. Verify Cleaning service card |  | Cleaning service card should be displayed | As expected | Pass |
| 5. Verify Gardening service card |  | Gardening service card should be displayed | As expected | Pass |
| 6. Click on Electrical service | Electrical | User should be redirected to Electrical service booking page | As expected | Pass |
| 7. Navigate back to Home Panel | \_\_ | Home panel should load again | As expected | Pass |
| 8. Click on Cleaning service | Cleaning | User should be redirected to Cleaning service booking page | As expected | Pass |
| 9. Navigate back to Home Panel |  | Home panel should load again | As expected | Pass |
| 10. Click on Gardening service | Gardening | User should be redirected to Gardening service booking page | As expected | Pass |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Doorbell | | | Test Designed by: Mahfujur Rahaman | | |
| Test Case ID:TC\_01 | | | Test Designed date:06/01/2026 | | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by: Mahfujur Rahaman | | |
| Module Name: Login | | | Test Execution date: 01/02/2026 | | |
| Test Title: Verify sidebar navigation links are working properly | | |  | | |
| Description: Verify login with valid username and password | | |  | | |
| Precondition: User must be registered in the system. | | | | | |
| Dependencies: Test the login functionality using valid user credentials. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Open the application login page. 2. Enter valid email address. 3. Enter valid password      1. Click on the login button. | Email: user@gmail.com   Password: user1234 | User should be logged in successfully.    User should be redirected to the dashboard page. | | As expected. | Pass. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Doorbell | | | Test Designed by: Mahfujur Rahaman | | |
| Test Case ID:TC\_03 | | | Test Designed date:06/01/2026 | | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by: Mahfujur Rahaman | | |
| Module Name: Sidebar | | | Test Execution date: 01/02/2026 | | |
| Test Title: Verify sidebar navigation links are working properly | | |  | | |
| Description: Test the sidebar navigation functionality to ensure all sidebar menu options redirect to the correct pages | | |  | | |
| Precondition: User must be logged into the system successfully. | | | | | |
| Dependencies: if any | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Open the application dashboard. 2. Locate the sidebar menu. 3. Click on each sidebar option (Home, Profile, Settings, Logout). 4. Observe the page navigation. | Home   Profile   Settings   Logout | Sidebar should be visible after login.     Each menu item should redirect to its respective page correctly.    Logout should redirect to the login page. | | As expected. | Pass. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Doorbell | | | Test Designed by: Mahfujur Rahaman | | |
| Test Case ID:TC\_02 | | | Test Designed date:06/01/2026 | | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by: Mahfujur Rahaman | | |
| Module Name: Reset Password | | | Test Execution date: 01/02/2026 | | |
| Test Title: Verify login with valid username and password | | |  | | |
| Description: Test the password reset functionality | | |  | | |
| Precondition: 1. User/Admin/Employee is logged in 2. User is on Dashboard/Home screen | | | | | |
| Dependencies: Services must exist in the system (for positive flow) | | | | | |
| **Test Steps** | **Test Data** | **Expected Results** | | **Actual Results** | **Status (Pass/Fail)** |
| 1. Open Dashboard/Home screen | — | “All Services” button should be clearly visible | | As expected | Pass |
| 2. Click on “All Services” button | — | Full list of service categories should open | | As expected | Pass |
| 3. Verify service categories list | Cleaning, Gardening, Electrical, Plumbing | All available categories should be displayed | | As expected | Pass |
| 4. Verify service name & icon | — | Each service should show name and icon/image | | As expected | Pass |
| 5. Check page load behavior | — | Service list should load quickly without errors or blank items | | As expected | Pass |
| 6. Click on a service category | Cleaning | Selected service details page should open | | As expected | Pass |
| 7. Navigate back from service details | — | User should return to the service list page | | As expected | Pass |
| 8. Verify behavior when no services exist | No services in system | “No services available” message should be shown | | As expected | Pass |
| 9. Verify access for different roles | User, Admin, Employee | Button should work according to role permissions | | As expected | Pass |
| 10. Verify responsive layout | Mobile & Desktop view | Layout should display correctly on all devices | | As expected | Pass |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Doorbell | | | Test Designed by: Mahfujur Rahaman | | |
| Test Case ID:TC\_02 | | | Test Designed date:06/01/2026 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: Mahfujur Rahaman | | |
| Module Name: Dashboard / Services | | | Test Execution date: 01/02/2026 | | |
| Test Title: Verify functionality of “All Services” top button | | |  | | |
| Description: Test the visibility, functionality, and navigation behavior of the “All Services” button | | |  | | |
| Precondition: 1. User/Admin/Employee is logged in 2. User is on Dashboard/Home screen | | | | | |
| Dependencies: Services must exist in the system (for positive flow) | | | | | |
| **Test Steps** | **Test Data** | **Expected Results** | | **Actual Results** | **Status (Pass/Fail)** |
| 1. Open Dashboard/Home screen | — | “All Services” button should be clearly visible | | As expected | Pass |
| 2. Click on “All Services” button | — | Full list of service categories should open | | As expected | Pass |
| 3. Verify service categories list | Cleaning, Gardening, Electrical, Plumbing | All available categories should be displayed | | As expected | Pass |
| 4. Verify service name & icon | — | Each service should show name and icon/image | | As expected | Pass |
| 5. Check page load behavior | — | Service list should load quickly without errors or blank items | | As expected | Pass |
| 6. Click on a service category | Cleaning | Selected service details page should open | | As expected | Pass |
| 7. Navigate back from service details | — | User should return to the service list page | | As expected | Pass |
| 8. Verify behavior when no services exist | No services in system | “No services available” message should be shown | | As expected | Pass |
| 9. Verify access for different roles | User, Admin, Employee | Button should work according to role permissions | | As expected | Pass |
| 10. Verify responsive layout | Mobile & Desktop view | Layout should display correctly on all devices | | As expected | Pass |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Project Name:** DOORBELL | | | Test Designed by: Mahfujur Rahaman | | |
| **Test Case ID:** TC\_04 | | | Test Designed date:06/01/2026 | | |
| **Test Priority (Low, Medium, High):** High | | | Test Executed by: Mahfujur Rahaman | | |
| **Module Name:** Service Details & Booking | | | Test Execution date: 01/02/2026 | | |
| **Test Title:** Verify service details display and Book Now functionality | | |  | | |
| **Description:** Test the service details page including service information, reviews, and booking action | | | | | |
| **Precondition:** 1.User is logged in 2.User has selected a service from the service list | | | | | |
| **Dependencies:** Service must exist in the system | | | | | |
| **Test Steps** | **Test Data** | **Expected Results** | | **Actual Results** | **Status (Pass/Fail)** |
| 1. Open a service details page | Cleaning Service | Service details page should load successfully | | As expected | Pass |
| 2. Verify Service Name | House Cleaning | Service name should be displayed correctly | | As expected | Pass |
| 3. Verify Service Category Name | Cleaning | Service category name should be visible | | As expected | Pass |
| 4. Verify Service Description | Full home cleaning service | Service description should be displayed properly | | As expected | Pass |
| 5. Verify Service Price | 1500 BDT | Correct price should be shown | | As expected | Pass |
| 6. Verify service image/banner | — | Relevant service image should be displayed | | As expected | Pass |
| 7. Verify user reviews/comments section | User comments | User reviews should be visible with user name and rating | | As expected | Pass |
| 8. Scroll through comments | — | Comments section should be scrollable | | As expected | Pass |
| 9. Click “Book Now” button | — | User should be redirected to booking form/page | | As expected | Pass |
| 10. Verify booking navigation | — | Correct service details should be carried to booking page | | As expected | Pass |
| 11. Verify access control | User role | Only authorized users should be able to book service | | As expected | Pass |
| 12. Check responsive layout | Mobile & Desktop | Page layout should display correctly on all devices | | As expected | Pass |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Project Name:** DOORBELL | | | Test Designed by: Mahfujur Rahaman | | |
| **Test Case ID:** TC\_05 | | | Test Designed date:06/01/2026 | | |
| **Test Priority (Low, Medium, High):** High | | | Test Executed by: Ashif | | |
| **Module Name:** Payment | | | Test Execution date: 01/02/2026 | | |
| **Test Title:** Verify payment details and transparent pricing before booking | | |  | | |
| **Description:** Test the payment details screen to ensure correct pricing, validation, and mandatory field checks | | |
| **Precondition:**1. User is logged in. 2. User has selected a service and reached the payment page | | | | | |
| **Dependencies:** Selected service must have defined pricing | | | | | |
| **Test Steps** | **Test Data** | **Expected Results** | | **Actual Results** | **Status (Pass/Fail)** |
| 1. Open Payment Details page | — | Payment page should load successfully | | As expected | Pass |
| 2. Verify service cost breakdown | Service price, taxes, total | Full cost breakdown should be clearly displayed | | As expected | Pass |
| 3. Verify service details in payment | Cleaning – 1500 BDT | Payment info should match selected service | | As expected | Pass |
| 4. Enter valid email address | [user@email.com](mailto:user@email.com) | Email should be accepted | | As expected | Pass |
| 5. Enter invalid email address | useremail.com | System should show validation error | | As expected | Pass |
| 6. Enter valid phone number | 01798765432 | Phone number should be accepted | | As expected | Pass |
| 7. Leave phone number empty | — | System should prevent proceeding and show error | | As expected | Pass |
| 8. Change selected service | Gardening Service | Payment amount should update correctly | | As expected | Pass |
| 9. Leave required fields empty | Email / Phone | User should not be able to proceed | | As expected | Pass |
| 10. Click Continue/Proceed Payment | — | User should proceed only if all fields are valid | | As expected | Pass |

# 6. CONCLUSION

In this project, we successfully developed the concept and design of a Smart Doorbell Service System that provides three essential home services: Electrical, Cleaning, and Gardening. Through detailed requirement analysis, system design, and careful planning, we created a structured solution that ensures quick service access, organized communication, and improved user convenience. The project demonstrates how everyday household tasks can be simplified through digital automation and proper service management.

We used the Scrum Process Model, which helped our team work in short, manageable sprints. Each sprint allowed us to focus on specific modules, receive feedback, and implement improvements. This iterative approach significantly enhanced team collaboration, productivity, and delivery quality. The modular nature of Scrum also ensured that changes could be made without disrupting the entire development process.

Overall, this project helped us gain practical knowledge of software engineering concepts such as SRS, diagrams, UI design, testing, and Git workflow. It strengthened our understanding of real-life system development and improved our teamwork and communication skills. We believe this Smart Doorbell Service System can be further developed into a fully functional application that will provide convenient and reliable home services to users.

**Instructions:**

* Minimum of 3 members and Maximum of 5 members per group.
* Font: Times New Roman ; Size: 12; Justify the para [Ctrl + J].
* Delete the highlighted part after completing this project report.
* The completed report should be within the range of 40 to 50 pages.
* **Submission:** Bring a hard copy of this report [per group] on the project evaluation day. Also, you will need to upload the soft copy later.