

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

Faculty of Science & Technology (FST)

**PROJECT TITLE: DoorBell**

**Semester: Fall 25-26**

|  |  |  |
| --- | --- | --- |
| ***Group: 05*** | | ***Section: S*** |
| **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** |

**1.PROJECT PROPOSAL**



**1.1 Background to the Problem**

Finding reliable home service providers is a common problem. People often need services such as house cleaning, gardening, p lumbing, or electrical repair, but they usually do not know whom to contact. They depend on friends’ suggestions or online searches, which can be time-consuming and unreliable. In many cases, the service quality is poor or the workers are not trustworthy.

Service workers also face difficulties. Skilled cleaners, electricians, and gardeners often struggle to find enough customers. Managing schedules, keeping track of bookings, and handling payments manually can be confusing and inefficient. At the same time, service managers find it difficult to monitor workers, customer bookings, and payments using traditional methods.

To address these issues, this project proposes DoorBell, a centralized home service management system. DoorBell is a computer-based platform that connects customers with trusted local service workers. It allows users to book services, schedule appointments, make secure payments, and view service reviews in one place. The system also helps workers manage their assigned jobs and assists managers in organizing services, users, and transactions efficiently.

Who Will Use DoorBell?

1. Customer  
These are people who need someone to clean their house, fix a leak, or take care of their garden. They are busy and want to find a good worker quickly without stress.

* They want to book a service easily.
* They want to know the worker is trusted and skilled.
* They want to pay safely online.
* They want to read reviews from other customers.

2. Employee  
These are the cleaners, gardeners, electricians, and plumbers who do the work. They want to find more customers and keep their work organized.

* They need to see their job list for the day or week.
* They want to know the customer's address and details.
* They need to manage their schedule.
* They want a simple way to update their profile and availability.

3. Admin  
This is the admin person who makes sure DoorBell runs smoothly. They manage everyone and everything on the platform.

* They add or remove workers and services.
* They look at all bookings and payments.
* They solve problems if something goes wrong.
* They make sure the quality of service is good and customers are happy.

DoorBell helps all three groups work together smoothly, making home services simple and trustworthy for everyone.

**1.2 Selection of Process Model**

**1.2.1Process Model Selection (Scrum)**

The Scrum framework is used for this project as it has several features and roles for the users, and there might be changes while carrying out the development process. It is an agile technique that allows frequent improvements and changes as it has short sprints that ensure effective delivery within an academic team setup.

**1.2.2Reason for Using the Scrum Model**

The Scrum pattern has been selected for the DoorBell project because it allows the complex system to be developed in a structured and flexible manner. The DoorBell system comprises a number of functionalities like user registration/login, service booking, payment handling, profile management, review systems, and admin panels. It is a difficult task to implement all functionalities simultaneously. Scrum allows the whole process to be divided into smaller tasks referred to as sprints, which allows the entire process to be more organized.

The project also consists of various user roles, which include general users, service providers (employees), and an admin. Each role consists of diverse requirements and features. Scrum facilitates prioritizing the requirements in a product backlog and then executing them one by one, which is a wise way to manage features for various roles and ensure that users receive the right functionalities.

Another key reason why Scrum should be considered in projects involves change in project requirements that can occur while development takes place. This applies in academic projects where changed requirements based on evaluations from lecturers or project supervisors are needed, and Scrum allows this without any impact due to previously accomplished work.

Apart from increasing efficiency, Scrum enhances team working and collaboration. Since the project will be developed by a team within a short semester period, the application of Scrum will enable the team to break down their duties effectively. In every sprint, the team will deliver a functioning version of the system, allowing the team to identify mistakes promptly. With this understanding, the Scrum model is efficient for the application in the DoorBell project.

**2 .Product Requirements Document (PRD)**

|  |  |  |  |
| --- | --- | --- | --- |
| As A/An | I want to | so That | Acceptance Criteria |
| Admin | As a new user  I want to create an account  using phone/email | So that I can  access the  service platform. | Signup must verify a valid and unused email, ensure the  password is at least eight  characters and matches the re- entered password, and require selecting a user type. On successful signup, the user is saved in the database,  redirected to the dashboard, and  sent a confirmation email. |
|  | As a registered user  I want to log in securely with my credentials | So that I can access my dashboard. | Sign in must confirm that the email and password match the database; otherwise, an error message appears. Users must be able to navigate to signup and password recovery pages from here. |
|  | I want to reset my forgotten password | So that I can  recover access | The system must confirm the email exists, send a reset code, and require a valid new  password and matching confirmation. If valid, the  password updates successfully; otherwise, clear error messages  appear. |
|  | I want to  manage all system  information | So that I can control services, users, and business data. | The admin menu must correctly open service modules, user  tables, service tables, |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | transactions, bookings, and profile options. |
|  | I want to view my stored  information | So that I can verify and manage my  account | The profile page must show all user data without blanks and  allow navigation to edit profile. |
|  | I want to update my personal  info | So that my profile stays accurate. | Editing must validate email  format, phone number, and age, then save changes or show errors. |
|  | I want to change my profile picture | So that I can  personalize my account. | Users may upload only PNG/JPG images, and after confirming, the system must show the  updated photo. |
|  | I want to change my password | So that I can keep my account secure. | The system must verify the current password, validate the new one, and update it or show an error. |
|  | View Service Table | I can update existing services | Table with update/edit options |
|  | View Transactions | I can track all payments | Transaction list displayed with filters |
|  | Log out | I can exit the  system securely | Redirects to login screen |
|  | View payment breakdown (when needed) | I can confirm cost structure | Shows full cost breakdown clearly |
| Employee | As a new user  I want to create an account  using phone/email | So that I can  access the  service platform. | Signup must verify a valid and unused email, ensure the  password is at least eight  characters and matches the re- entered password, and require selecting a user type. On successful signup, the user is saved in the database,  redirected to the dashboard, and sent a confirmation email. |

|  |  |  |  |
| --- | --- | --- | --- |
|  | As a registered user  I want to log in securely with my credentials | So that I can access my dashboard. | Sign in must confirm that the email and password match the database; otherwise, an error message appears. Users must be able to navigate to signup and password recovery pages from here. |
|  | I want to reset my forgotten password | So that I can  recover access | The system must confirm the email exists, send a reset code, and require a valid new  password and matching confirmation. If valid, the  password updates successfully; otherwise, clear error messages appear. |
|  | I want to  manage all system  information | So that I can  complete tasks and track work. | The dashboard must show service lists, work lists,  bookings, profile, and logout, all functioning properly. |
|  | I want to view my stored  information | So that I can verify and manage my  account | The profile page must show all user data without blanks and  allow navigation to edit profile. |
|  | I want to update my personal  info | So that my profile stays accurate. | Editing must validate email  format, phone number, and age, then save changes or show errors. |
|  | I want to change my profile picture | So that I can  personalize my account. | Users may upload only PNG/JPG images, and after confirming, the system must show the  updated photo. |
|  | I want to change my password | So that I can keep my account secure. | The system must verify the current password, validate the new one, and update it or show an error. |
|  | Use “All Servicesˮ button | I can see available service categories | Displays all categories with icons |

|  |  |  |  |
| --- | --- | --- | --- |
|  | View service details | I can prepare for the work | Shows customer info & location |
|  | View work list | I can track tasks | Shows all assigned bookings |
|  | View my bookings | I can review my completed services | Booking list with status |
| User | As a new user  I want to create an account  using phone/email | So that I can  access the  service platform. | Signup must verify a valid and unused email, ensure the  password is at least eight  characters and matches the re- entered password, and require selecting a user type. On successful signup, the user is saved in the database,  redirected to the dashboard, and sent a confirmation email. |
|  | As a registered user  I want to log in securely with my credentials | So that I can access my dashboard. | Sign in must confirm that the email and password match the database; otherwise, an error message appears. Users must be able to navigate to signup and password recovery pages from here. |
|  | I want to reset my forgotten password | So that I can  recover access | The system must confirm the email exists, send a reset code, and require a valid new  password and matching confirmation. If valid, the  password updates successfully; otherwise, clear error messages appear. |
|  | I want to see a clean menu of services (Gardening,  Cleaning, Plumbing) | So that I can control services, users, and business data. | The dashboard must show service categories, bookings, profile access, and logout, with all options linking correctly. |

|  |  |  |  |
| --- | --- | --- | --- |
|  | I want to view my stored  information | So that I can verify and manage my  account | The profile page must show all user data without blanks and  allow navigation to edit profile. |
|  | I want to update my personal  info | So that my profile stays accurate. | Editing must validate email  format, phone number, and age, then save changes or show errors. |
|  | I want to change my profile picture | So that I can  personalize my account. | Users may upload only PNG/JPG images, and after confirming, the system must show the  updated photo. |
|  | I want to change my password | So that I can keep my account secure. | The system must verify the current password, validate the new one, and update it or show an error. |
|  | View clean service menu | I can quickly choose a service | Shows Gardening, Cleaning, Electrical |
|  | View specific service details | I understand what Iʼm booking | Shows details, pricing, available employees |
|  | Log out | I can exit safely | Redirect to login |
|  | Choose payment method | I can complete booking | Must select one method; success/failure message |
|  | Make secure payment | My booking becomes confirmed | payment success → “Paidˮ; failure → error shown |

**2.1 Project Scope and Features**

The DoorBell – Home Service Management System is a web-based platform designed to digitalize and streamline the connection between homeowners and trusted local service providers. The project aims to replace inefficient, manual methods of finding and booking services with a secure, user-friendly, and role-based system. It enhances accessibility, ensures service quality, and brings transparency to scheduling, payments, and reviews.

This system will serve three primary user roles: Admin, Employee (Service Provider), and Customer (Homeowner). It will support end-to-end operations including service discovery, online booking, automated scheduling, secure payment processing, and centralized management.

**2.1.1Key Features:**

The system’s functionality is divided based on user roles. The key features for each role are outlined below.

1) Admin Features:

The Admin is the system supervisor with the highest level of access, responsible for overall platform management, user oversight, and system configuration.

Key functionalities:

* Login securely using admin credentials.
* Manage all user accounts (Customers and Employees)—create, view, update, and delete.
* Manage the service catalogue—add, update, search, and delete services (e.g., Cleaning, Gardening).
* View and monitor all bookings and transactions in real-time.
* Generate and export transaction reports to Excel/PDF.
* Update personal admin profile (username and role are unchangeable).
* Configure system settings (service categories, business hours, payment methods).
* View all customer ratings and reviews.
* Handle and resolve reported issues or disputes.
* Search transactions and bookings by date, user, or service type.
* Log out users remotely if necessary.
* Ensure data consistency and system performance.

2) Employee (Service Provider) Features:

The Employee is a verified service professional (e.g., cleaner, electrician, gardener) who uses the platform to receive job assignments and manage their work.

Key functionalities:

* Login securely with employee credentials.
* View a personalized dashboard with upcoming and assigned jobs.
* See detailed job information, including customer details, service address, and special instructions.
* Update job status (e.g., Accepted, On the way, Completed).
* View personal work history and completed services.
* Update personal profile, including contact information, skills, and availability.
* Change account password securely.
* Upload or change a profile picture.
* Access a list of all available service categories.
* Receive notifications for new job assignments.

3) Customer (Homeowner) Features:

The Customer is a homeowner seeking reliable home services. They interact with the platform to find, book, and pay for services.

Key functionalities:

* Register an account and login securely.
* Browse and search for services by category (Cleaning, Gardening, Plumbing, Electrical, etc.).
* View detailed service descriptions, pricing, and available service providers.
* Book a service by selecting a date, time, and preferred provider.
* Choose from secure online payment methods.
* View a detailed breakdown of costs before confirming payment.
* Track the status of current and past bookings.
* View and download payment receipts.
* Update personal profile information (name, email, phone, address).
* Change account password and profile picture.
* Provide ratings and reviews for completed services.
* Log out of the system securely.

**2.2 User Story Table**

|  |  |  |  |
| --- | --- | --- | --- |
| As A/An | I want to | so That | Acceptance Criteria |
| admin | As a new user  I want to create an account  using phone/email | So that I can  access the  service platform. | Signup must verify a valid and unused email, ensure the  password is at least eight  characters and matches the re- entered password, and require selecting a user type. On successful signup, the user is saved in the database,  redirected to the dashboard, and  sent a confirmation email. |
|  | As a registered user  I want to log in securely with my credentials | So that I can access my dashboard. | Sign in must confirm that the email and password match the database; otherwise, an error message appears. Users must be able to navigate to signup and password recovery pages from here. |
|  | I want to reset my forgotten password | So that I can  recover access | The system must confirm the email exists, send a reset code, and require a valid new  password and matching confirmation. If valid, the  password updates successfully; otherwise, clear error messages  appear. |

|  |  |  |  |
| --- | --- | --- | --- |
|  | I want to  manage all system  information | So that I can control services, users, and business data. | The admin menu must correctly open service modules, user  tables, service tables,  transactions, bookings, and profile options. |
|  | I want to view my stored  information | So that I can verify and manage my  account | The profile page must show all user data without blanks and  allow navigation to edit profile. |
|  | I want to update my personal  info | So that my profile stays accurate. | Editing must validate email  format, phone number, and age, then save changes or show errors. |
|  | I want to change my profile picture | So that I can  personalize my account. | Users may upload only PNG/JPG images, and after confirming, the system must show the  updated photo. |
|  | I want to change my password | So that I can keep my account secure. | The system must verify the current password, validate the new one, and update it or show an error. |
|  | View Service Table | I can update existing services | Table with update/edit options |
|  | View Transactions | I can track all payments | Transaction list displayed with filters |
|  | Log out | I can exit the  system securely | Redirects to login screen |
|  | View payment breakdown (when needed) | I can confirm cost structure | Shows full cost breakdown clearly |
|  |  |  |  |
| employee | As a new user  I want to create an account  using phone/email | So that I can  access the  service platform. | Signup must verify a valid and unused email, ensure the  password is at least eight  characters and matches the re- entered password, and require selecting a user type. On successful signup, the user is |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | saved in the database,  redirected to the dashboard, and sent a confirmation email. |
|  | As a registered user  I want to log in securely with my credentials | So that I can access my dashboard. | Sign in must confirm that the email and password match the database; otherwise, an error message appears. Users must be able to navigate to signup and password recovery pages from here. |
|  | I want to reset my forgotten password | So that I can  recover access | The system must confirm the email exists, send a reset code, and require a valid new  password and matching confirmation. If valid, the  password updates successfully; otherwise, clear error messages appear. |
|  | I want to  manage all system  information | So that I can  complete tasks and track work. | The dashboard must show service lists, work lists,  bookings, profile, and logout, all functioning properly. |
|  | I want to view my stored  information | So that I can verify and manage my  account | The profile page must show all user data without blanks and  allow navigation to edit profile. |
|  | I want to update my personal  info | So that my profile stays accurate. | Editing must validate email  format, phone number, and age, then save changes or show errors. |
|  | I want to change my profile picture | So that I can  personalize my account. | Users may upload only PNG/JPG images, and after confirming, the system must show the  updated photo. |
|  | I want to change my password | So that I can keep my account secure. | The system must verify the current password, validate the new one, and update it or show an error. |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Use “All Servicesˮ button | I can see available service categories | Displays all categories with icons |
|  | View service details | I can prepare for the work | Shows customer info & location |
|  | View work list | I can track tasks | Shows all assigned bookings |
|  | View my bookings | I can review my completed services | Booking list with status |
|  |  |  |  |
| User | As a new user  I want to create an account  using phone/email | So that I can  access the  service platform. | Signup must verify a valid and unused email, ensure the  password is at least eight  characters and matches the re- entered password, and require selecting a user type. On successful signup, the user is saved in the database,  redirected to the dashboard, and sent a confirmation email. |
|  | As a registered user  I want to log in securely with my credentials | So that I can access my dashboard. | Sign in must confirm that the email and password match the database; otherwise, an error message appears. Users must be able to navigate to signup and password recovery pages from here. |
|  | I want to reset my forgotten password | So that I can  recover access | The system must confirm the email exists, send a reset code, and require a valid new  password and matching confirmation. If valid, the  password updates successfully; otherwise, clear error messages appear. |
|  | I want to see a clean menu of services | So that I can  control services, | The dashboard must show service categories, bookings, |

|  |  |  |  |
| --- | --- | --- | --- |
|  | (Gardening, Cleaning,  Plumbing) | users, and business data. | profile access, and logout, with all options linking correctly. |
|  | I want to view my stored  information | So that I can verify and manage my  account | The profile page must show all user data without blanks and  allow navigation to edit profile. |
|  | I want to update my personal  info | So that my profile stays accurate. | Editing must validate email  format, phone number, and age, then save changes or show errors. |
|  | I want to change my profile picture | So that I can  personalize my account. | Users may upload only PNG/JPG images, and after confirming, the system must show the  updated photo. |
|  | I want to change my password | So that I can keep my account secure. | The system must verify the current password, validate the new one, and update it or show an error. |
|  | View clean service menu | I can quickly choose a service | Shows Gardening, Cleaning, Electrical |
|  | View specific service details | I understand what Iʼm booking | Shows details, pricing, available employees |
|  | Log out | I can exit safely | Redirect to login |
|  | Choose payment method | I can complete booking | Must select one method; success/failure message |
|  | Make secure payment | My booking becomes confirmed | payment success → “Paidˮ; failure → error shown |

**2.3 Requirements Traceability Matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requirement ID** | **Requirement Description** | **Source (SRS/User Story)** | **Design Element / Module** | **Test Status** | **Remarks / Notes** |
| FR-01 | User can sign up with email/phone and password | US01, SRS 2.1.1 | Authentication Module – Registration Component | Pass | Valid email & password validation tested |
| FR-02 | User can log in with credentials | US02, SRS 2.1.1 | Authentication Module – Login Component | Pass | Invalid login blocked after 5 attempts |
| FR-03 | User can reset password via email | US03, SRS 2.1.1 | Authentication Module – Password Reset Component | Pass | Reset link expires after 15 minutes |
| FR-04 | User can view profile information | US04, SRS 2.1.2 | Profile Module – Profile View Component | Pass | All fields displayed correctly |
| FR-05 | User can edit profile (name, phone, address) | US05, SRS 2.1.2 | Profile Module – Profile Edit Component | Pass | Email & phone validated |
| FR-06 | User can change profile picture | US06, SRS 2.1.2 | Profile Module – Picture Upload Component | Pass | Only JPG/PNG accepted |
| FR-07 | User can change account password | US07, SRS 2.1.2 | Profile Module – Password Change Component | Pass | Old password required |
| FR-08 | Customer can browse service categories | US08, SRS 2.2.1 | Service Catalogue Module – Service Menu | Pass | Categories shown with icons |
| FR-09 | Customer can view service details | US09, SRS 2.2.1 | Service Catalogue Module – Service Details | Pass | Price & workers shown |
| FR-10 | Customer can book service with date & time | US10, SRS 2.2.2 | Booking Module – Booking Form | In Progress | Date validation pending |
| FR-11 | Customer can choose payment method | US11, SRS 2.2.2 | Payment Module – Payment Selection | Not Started | Gateway integration pending |
| FR-12 | Customer can see cost breakdown | US12, SRS 2.2.2 | Payment Module – Cost Summary | Pass | All charges visible |
| FR-13 | Customer can view booking history | US13, SRS 2.2.3 | Booking Module – History | Pass | Past & upcoming bookings shown |
| FR-14 | Employee can see service categories | US14, SRS 2.3.1 | Employee Dashboard – Service List | Pass | Load time < 2 seconds |
| FR-15 | Employee can view assigned work | US15, SRS 2.3.2 | Employee Dashboard – Work List | Pass | Customer info shown |
| FR-16 | Employee can view job details | US16, SRS 2.3.2 | Employee Dashboard – Job Details | Pass | Map integration works |
| FR-17 | Employee can view personal history | US17, SRS 2.3.3 | Employee Dashboard – History | Pass | Filter option available |
| FR-18 | Admin can manage users | US18, SRS 2.4.1 | Admin Panel – User Management | Pass | Role-based access enforced |
| FR-19 | Admin can manage service catalogue | US19, SRS 2.4.2 | Admin Panel – Service Management | Pass | CRUD tested |
| FR-20 | Admin can view all transactions | US20, SRS 2.4.3 | Admin Panel – Reporting | Pass | Real-time data shown |
| FR-21 | Admin can filter transactions | US21, SRS 2.4.3 | Admin Panel – Search & Filter | Pass | Results < 2 seconds |
| FR-22 | Admin can export data (Excel/PDF) | US22, SRS 2.4.3 | Admin Panel – Export Module | Not Started | Library pending |
| FR-23 | Admin can monitor all bookings | US23, SRS 2.4.4 | Admin Panel – Booking Monitor | Pass | Live status updates |
| FR-24 | Admin can configure system settings | US24, SRS 2.4.5 | Admin Panel – Configuration | Pass | No restart required |
| FR-25 | User can log out securely | US25, SRS 2.5.1 | Authentication Module – Logout | Pass | Session destroyed |
| NFR-01 | Dashboard loads within 3 seconds | SRS 3.1.1 | Performance Module | Fail | Optimization required |
| NFR-02 | System supports 10,000 users | SRS 3.2.1 | Scalability Module | Not Started | Load testing pending |
| NFR-03 | Passwords stored encrypted | SRS 3.3.1 | Security Module | Pass | AES-256 verified |
| NFR-04 | System uptime 99.5% | SRS 3.4.1 | Reliability Module | Not Started | Monitoring pending |
| NFR-05 | Mobile responsive UI | SRS 3.5.1 | UI/UX Module | Pass | Tested on major browsers |

**2.3.1 Functional Requirements**

## Admin Requirements

|  |  |
| --- | --- |
| ID | Requirement Description |
| 1.0 | User Authentication & Dashboard |
| 1.1 | Login |
| 1.1.1 | Input email and password |
| 1.1.2 | Forgot password option |
| 1.1.2.1 | Email verification and reset link/code |
| 1.1.3 | Submit login and redirect to dashboard |
| 1.2 | Dashboard Side Menu |
| 1.2.1 | View Profile |
| 1.2.1.1 | Show name, email, phone, role, profile picture |
| 1.2.1.2 | Edit Profile button |
| 1.2.1.3 | Change Profile Picture button |
| 1.2.1.4 | Change Password button |
| 1.3 | Manage Services |
| 1.3.1 | View service list in table format |
| 1.3.2 | Add new service button |
| 1.3.3 | Edit existing service button |
| 1.3.4 | Delete service with confirmation |
| 1.4 | Manage Users |
| 1.4.1 | View user list (admin, employee, customer) |
| 1.4.2 | Add new user (employee/admin) |
| 1.4.3 | Edit user details |
| 1.4.4 | Delete user with confirmation |
| 1.5 | View Transactions |
| 1.5.1 | View all transactions with filters |
| 1.5.2 | Export transactions to Excel/PDF |
| 1.6 | View Bookings |
| 1.6.1 | View all bookings with status |
| 1.6.2 | Filter by service, date, user |
| 1.7 | System Configuration |
| 1.7.1 | Set service categories |
| 1.7.2 | Configure payment methods |
| 1.7.3 | Manage business hours |
| 1.8 | Logout |
| 1.8.1 | Redirect to login page |

Employee Requirements

|  |  |
| --- | --- |
| **ID** | **Requirement Description** |
| 1.0 | User Authentication & Dashboard |
| 1.1 | Login [Ref: Admin 1.1] |
| 1.2 | Dashboard Side Menu |
| 1.2.1 | View Profile [Ref: Admin 1.2.1] |
| 1.2.2 | Edit Profile [Ref: Admin 1.2.1.2] |
| 1.2.3 | Change Password [Ref: Admin 1.2.1.4] |
| 1.3 | View Service List |
| 1.3.1 | See all available service categories |
| 1.3.2 | Click to view service details |
| 1.4 | View Assigned Work |
| 1.4.1 | View all assigned bookings |
| 1.4.2 | View customer info and location |
| 1.5 | View My Bookings |
| 1.5.1 | See completed services with status |
| 1.6 | Logout |
| 1.6.1 | Redirect to login page |

Customer Requirements

|  |  |
| --- | --- |
| ID | Requirement Description |
| 1.0 | User Authentication |
| 1.0.1 | Signup with email/phone and password |
| 1.0.2 | Login with credentials |
| 1.0.3 | Reset forgotten password |
| 1.1 | Browse Services |
| 1.1.1 | View service categories (Gardening, Cleaning, etc.) |
| 1.1.2 | View service details, pricing, available employees |
| 1.2 | Book a Service |
| 1.2.1 | Select service and confirm booking |
| 1.2.2 | Choose payment method |
| 1.2.3 | Complete secure payment |
| 1.3 | View My Bookings |
| 1.3.1 | See all past and upcoming bookings |
| 1.3.2 | View payment breakdown |
| 1.4 | Profile Management |
| 1.4.1 | View profile information |
| 1.4.2 | Edit personal details |
| 1.4.3 | Change profile picture |
| 1.4.4 | Change password |
| 1.5 | Logout |
| 1.5.1 | Redirect to login page |

**2.3.2 Non-Functional Requirements**

|  |  |
| --- | --- |
| 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 | Handle 1,000 Support 10,000 registered users |
| 8.0.2 | 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:**

**Use Case Diagram:**

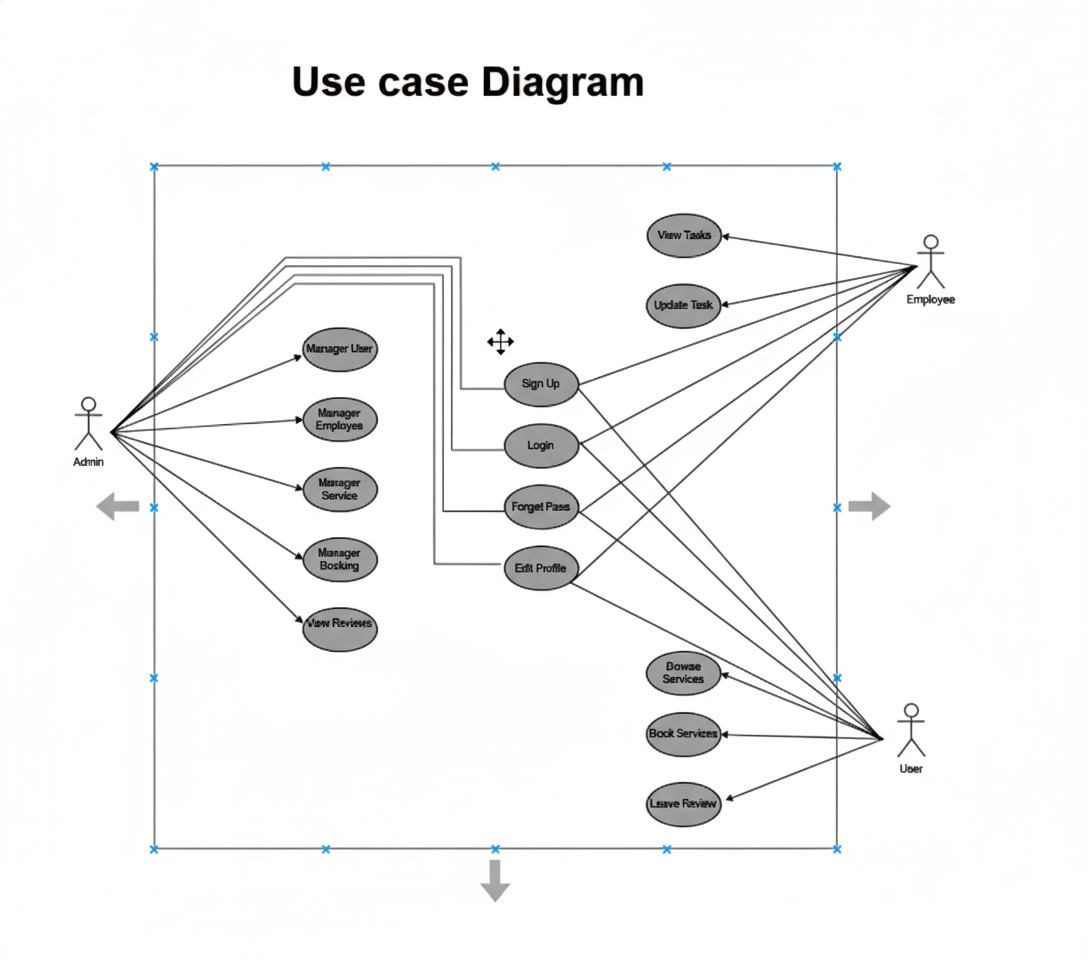


Fig: Use Case Diagram

**Class Diagram:**

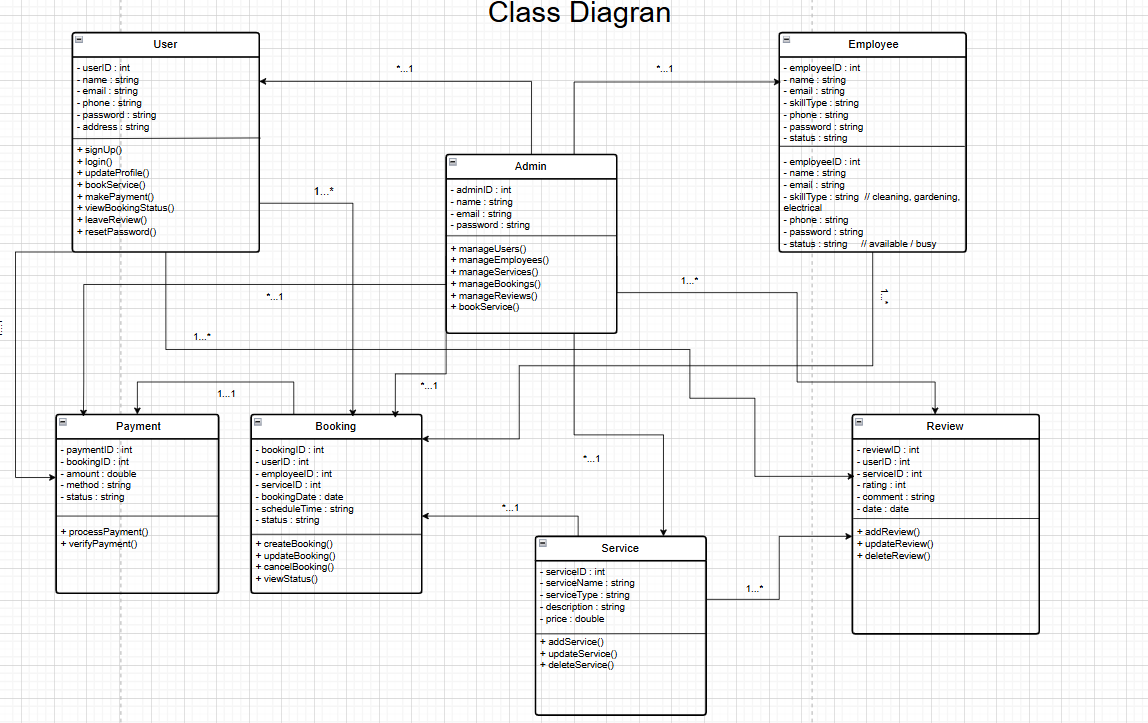


Fig: Class Diagram

Activity Diagram:

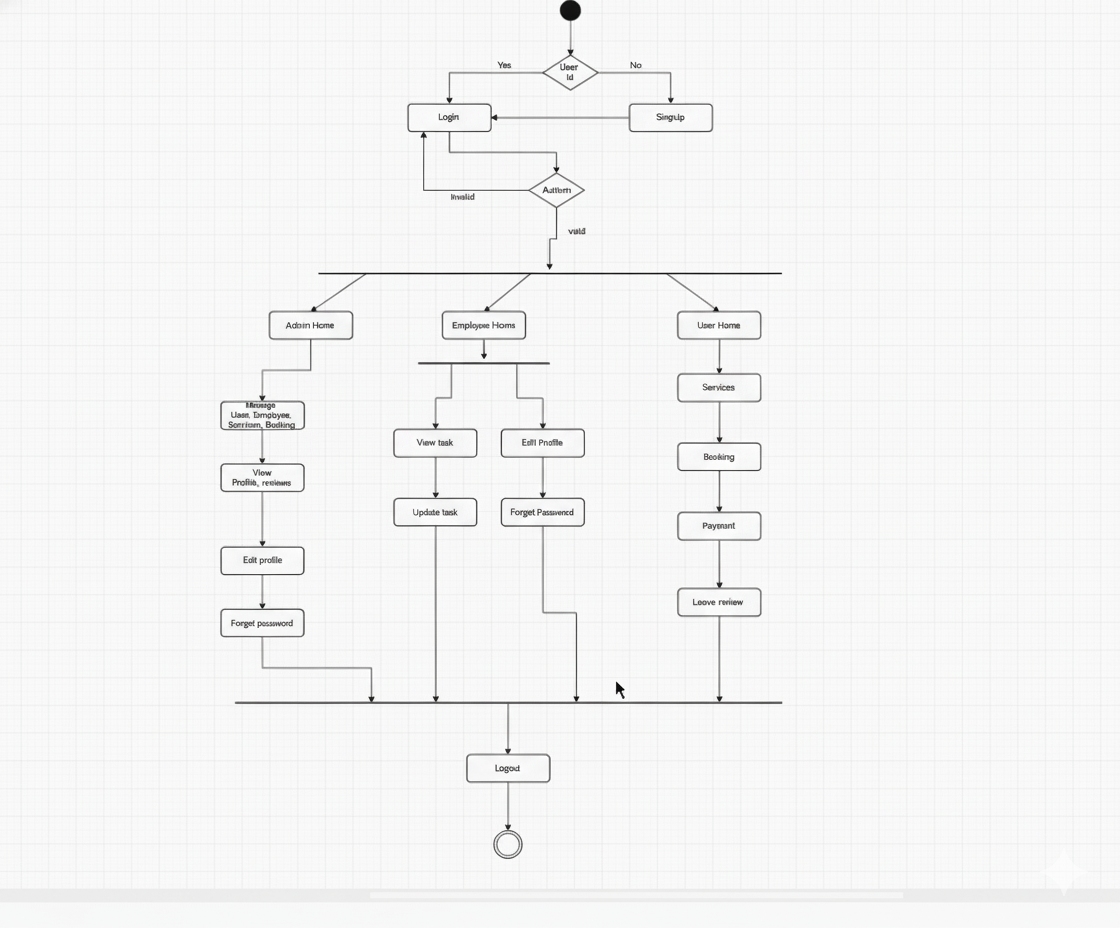


Fig: Activity Diagram

## **UI / Wireframe Design using Figma**

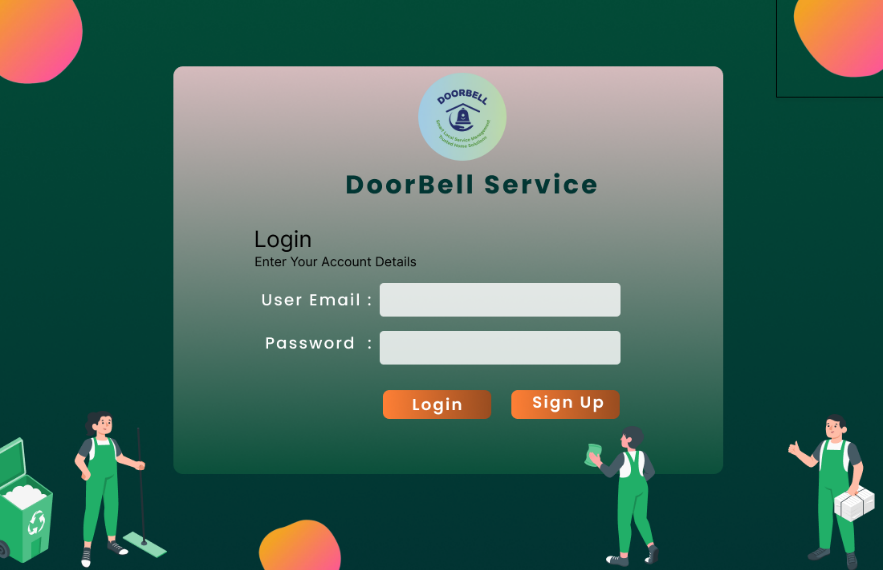
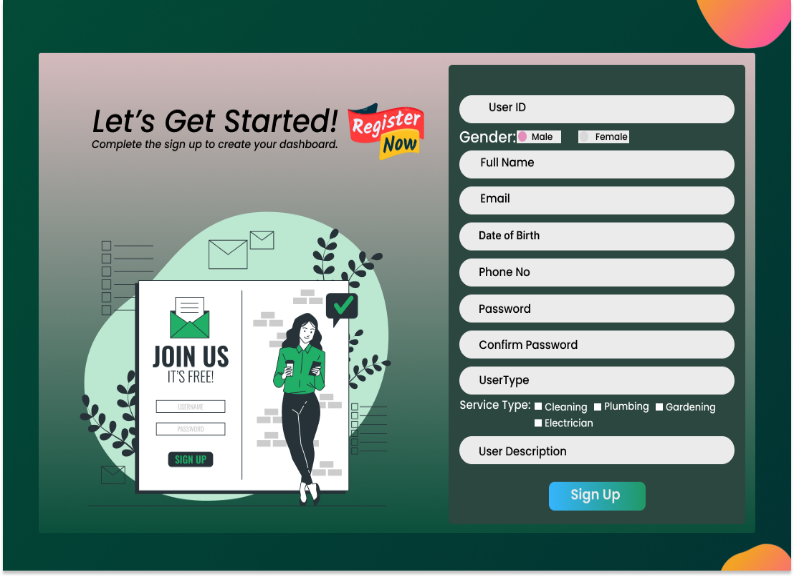
 

Fig: Login Panel Fig: Sing Up Panel

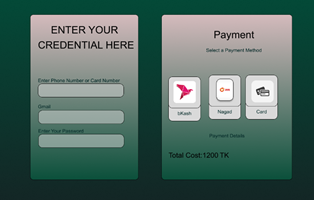
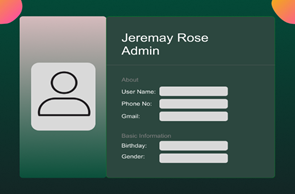
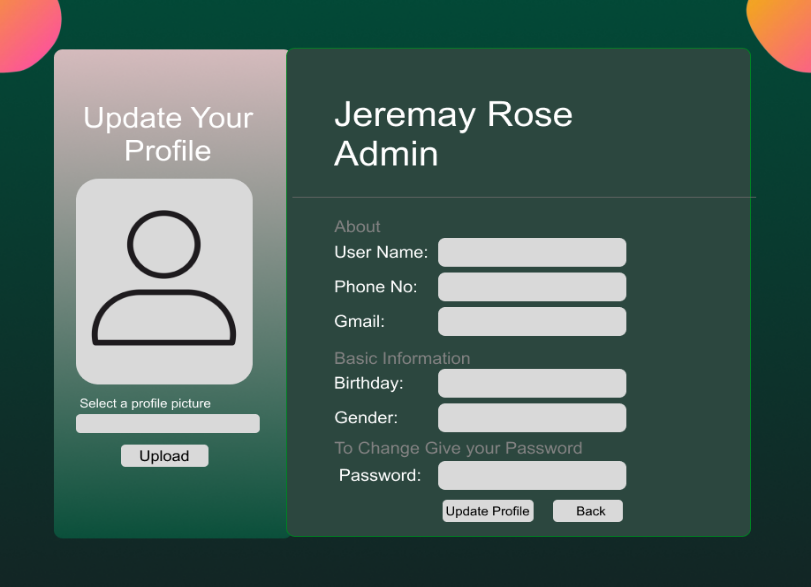
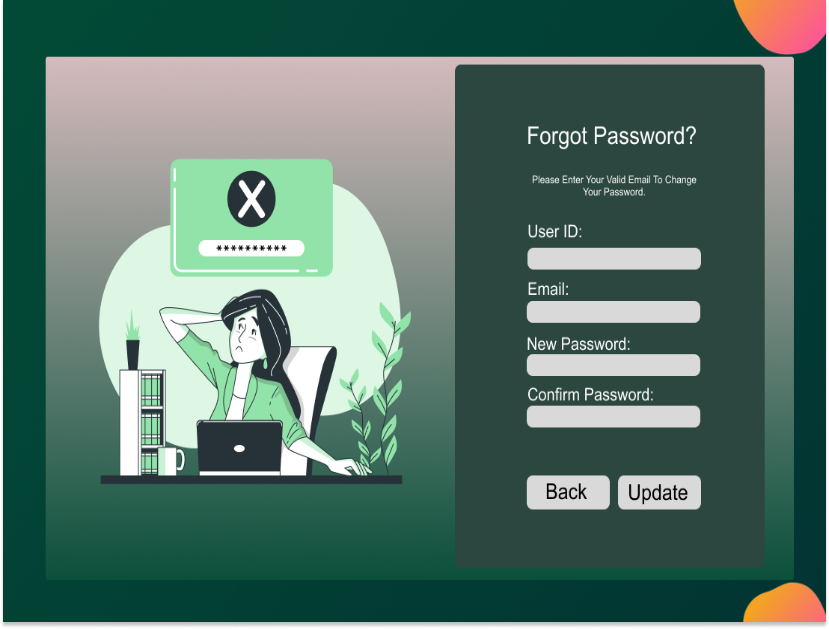
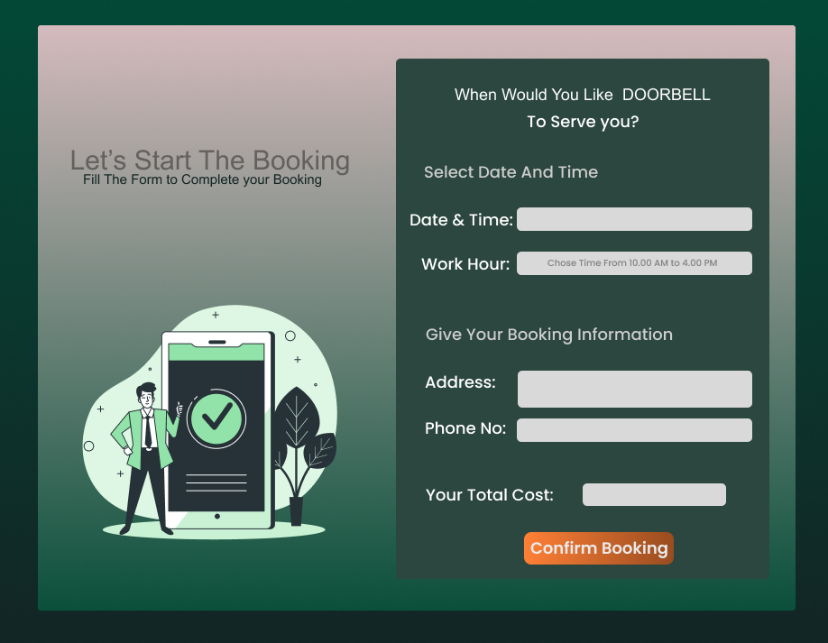
 

Fig: Home Panel Fig:Payment Panel

  Fig: Profile Panel Fig: Update Profile Panel

 Fig: Forgot Panel Fig: Booking Panel

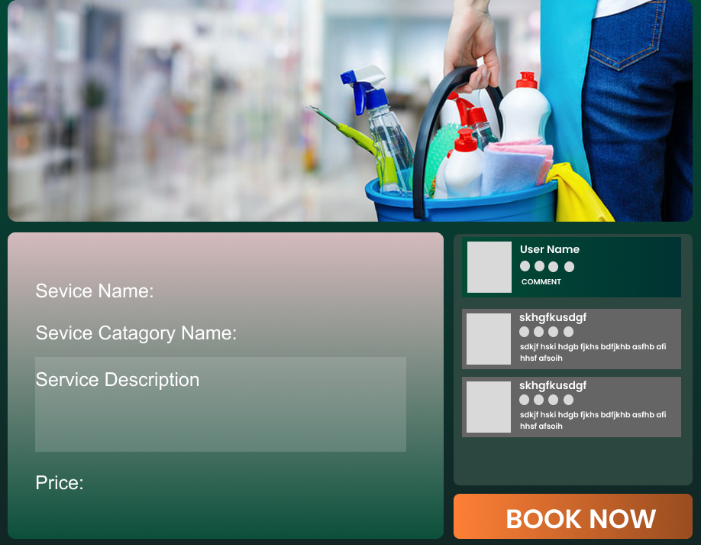
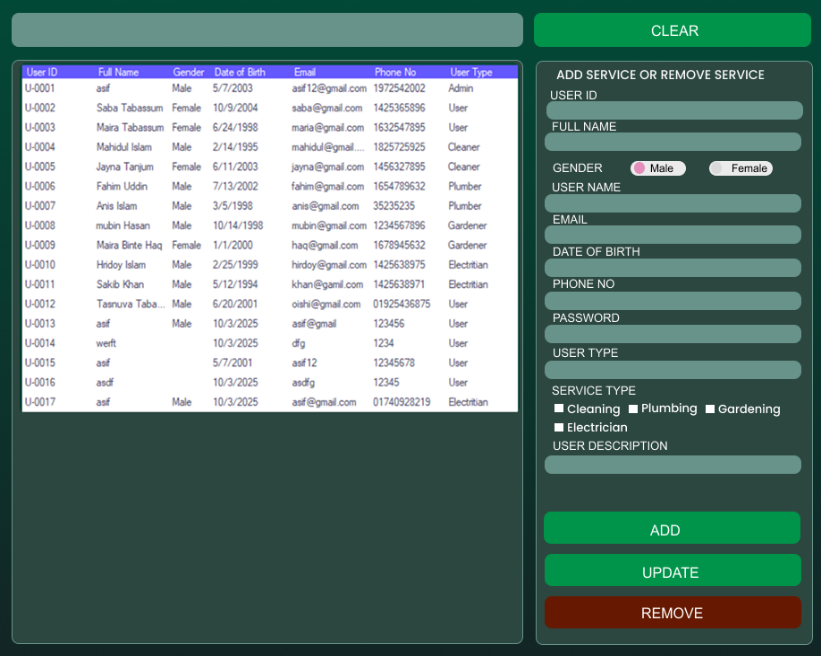
 

Fig: Home Panel Fig: Table Panel

**4. GIT WORKFLOW.**

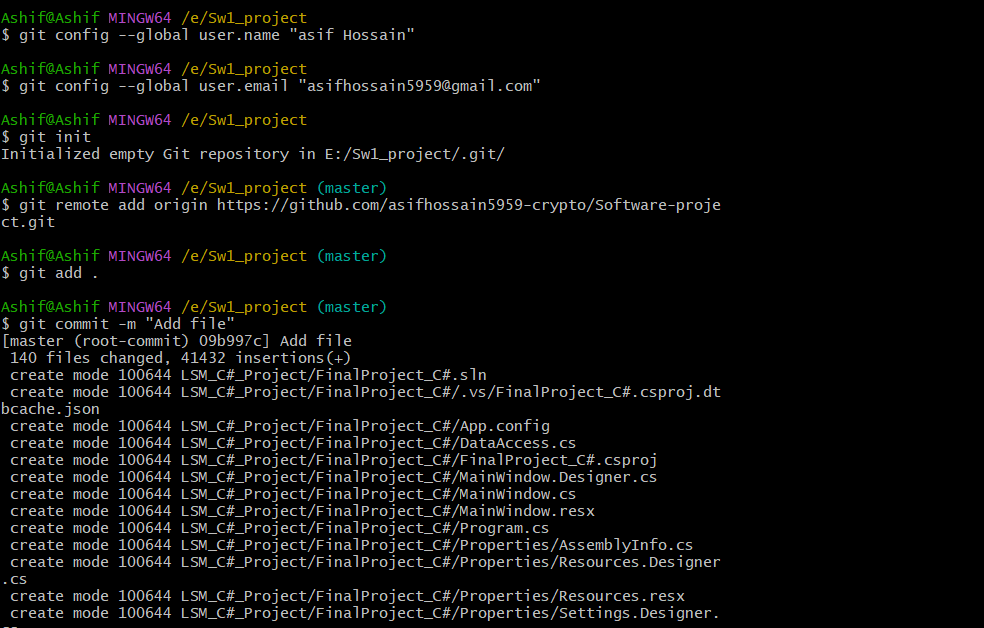


Fig: GitHub Connection

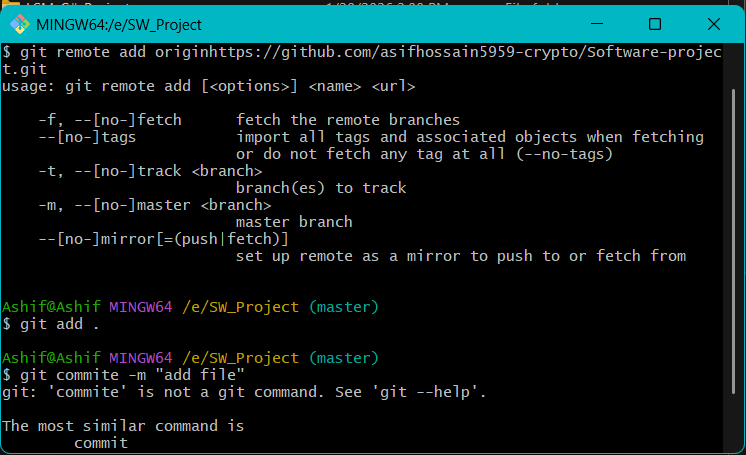


Fig: add & commite

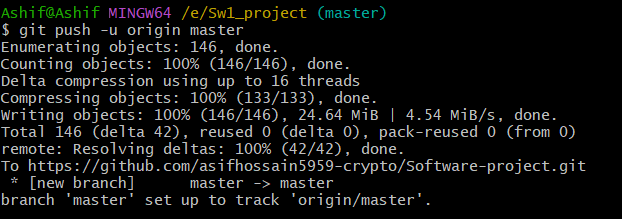


Fig: git push



Fig: git pull for update

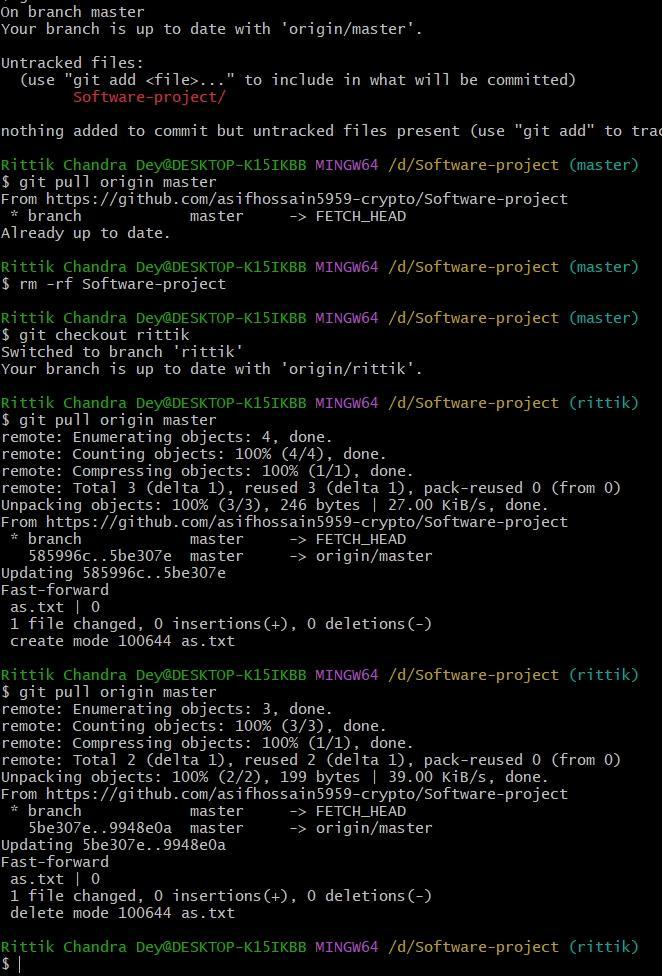


Fig: Branch Create

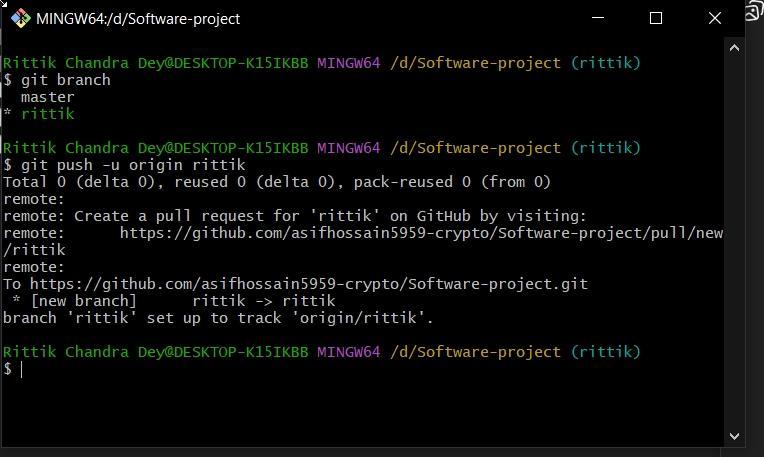


Fig: Branch push (edit or update)

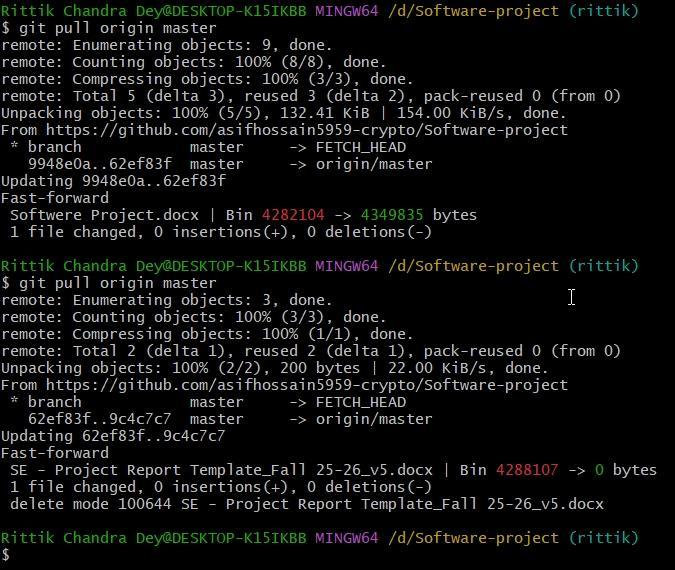


Fig: Branch (git pull)

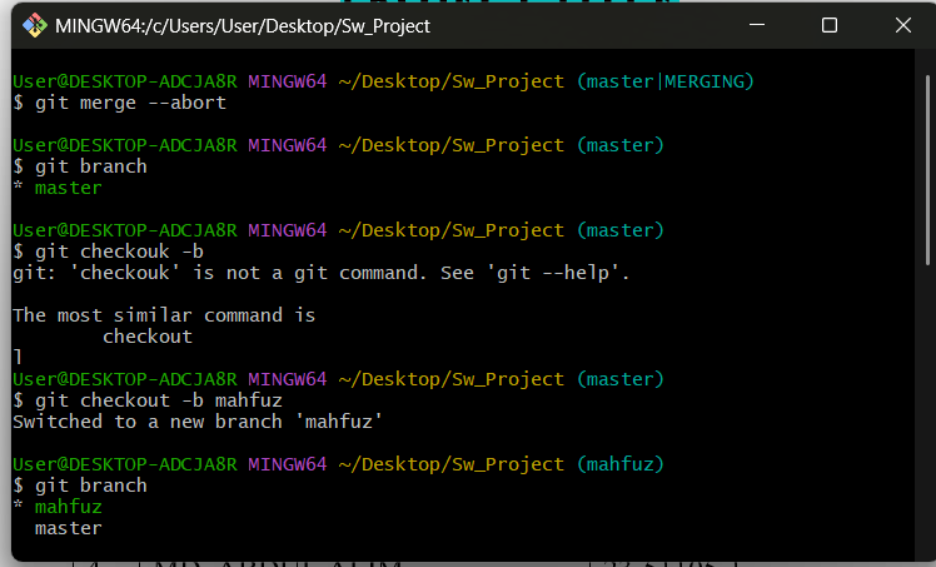


Fig: Merge conflict

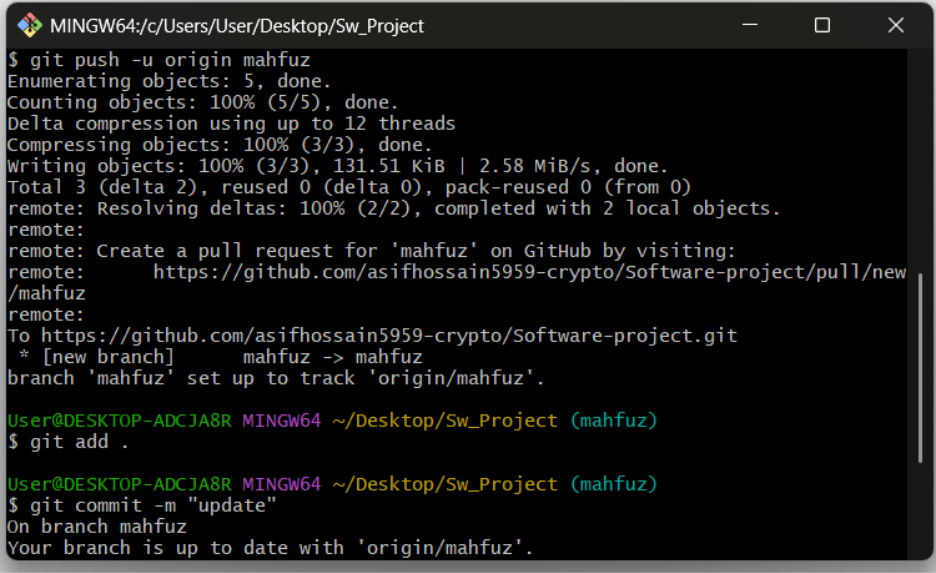
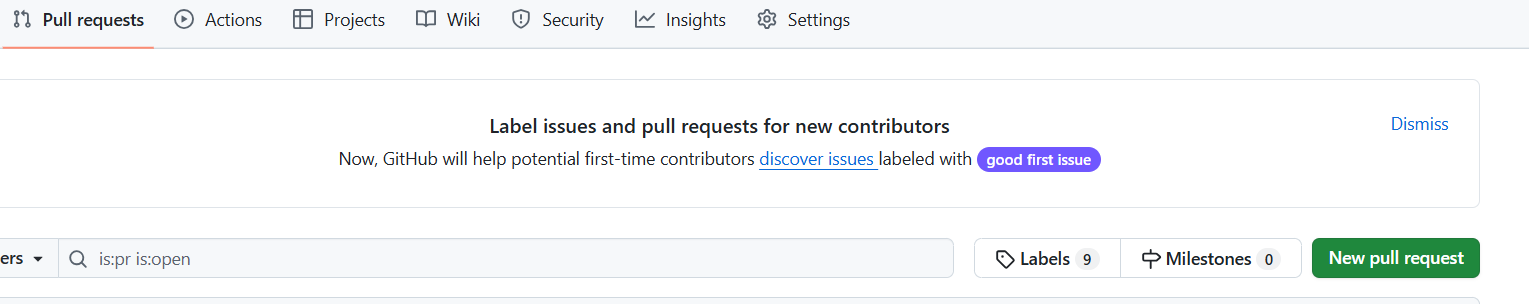
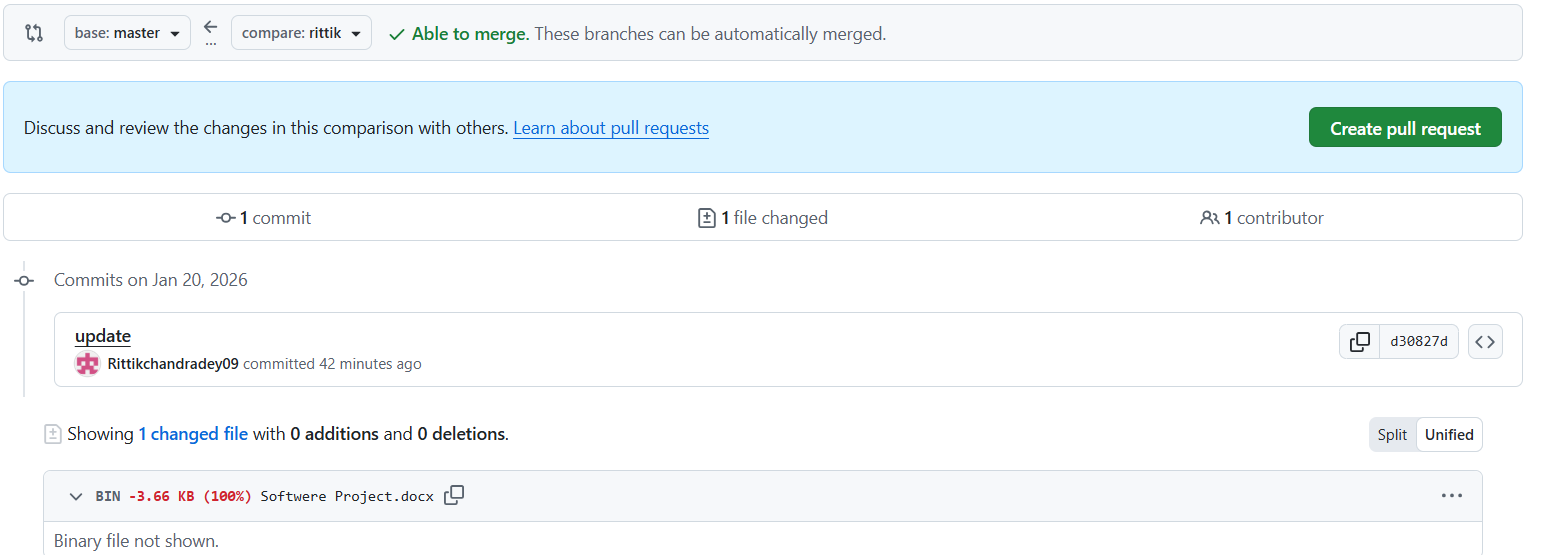
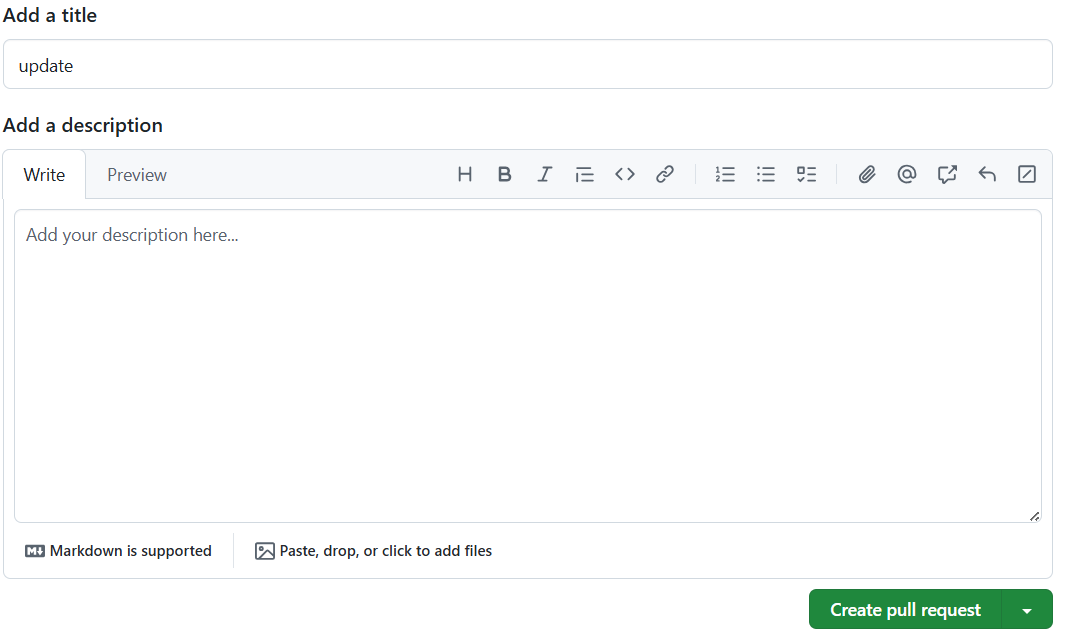
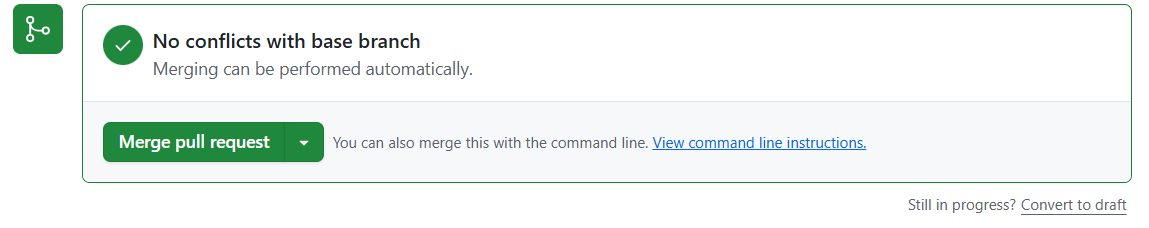


Fig: git add & commit on branch









# 5**. SOFTWARE TESTING**

5.1 **Testing Methods and Their Application**

Unit Testing :  
Unit testing means testing each small piece of our software on its own. We check the individual parts—like a login function, a booking button, or a price calculator—to make sure they work right by themselves before we connect them. This makes fixing problems easier because if something is wrong, we know it's in that one small part.

In our DoorBell project, we will unit test parts like:

* Login: Does it accept the right password? Does it block the wrong one?
* Booking: Does it save the correct date, time, and service type?
* Payment: Does it calculate the final price correctly, including any fees?

This step is very important. If a small part like the payment calculator is wrong, it could cause big problems with money later. Testing each piece first builds a strong foundation for our whole system.

Integration Testing :  
Integration testing checks what happens when we connect different parts of the system. Even if the login works alone and the booking works alone, we need to make sure they work together. We test how data moves from one part to another to prevent mistakes.

In DoorBell, we will test connections like:

* When a customer books a service, does it appear correctly on the worker’s schedule?
* When a payment is completed, is the booking marked as “Paid” for everyone—the customer, the worker, and the admin?
* If a worker updates their profile, does the change show up immediately in the admin panel?

This testing makes sure the whole system is connected properly. It stops problems where a customer thinks they have a booking, but the worker never gets notified.

System Testing :

System testing is where we test the *entire* DoorBell application as one complete product. We pretend to be real users and go through every step of a real job. We also check if the app is fast, safe, and easy to use.

We will run full scenarios like:

1. A customer signs up, finds a cleaner, books a job, pays, and leaves a review.
2. The cleaner sees the new job, goes to the location, finishes the work, and updates the status.
3. The admin sees the completed job in the system and can print a report.

We also check things like:

* Speed: Does the app slow down if many people try to use it at the same time?
* Security: Can a regular user see private admin pages?

This is our final check to make sure everything in the app works together perfectly and provides a good experience for all users.

Smoke Testing :

Smoke testing is a very quick check we do after we make any big changes to the software. We only test the most important things to see if the app is stable. If these basic things don't work, we know we have a major problem that needs to be fixed right away.

For example, after an update we quickly test:

* Can users still log in?
* Can you open the main dashboard?
* Can you start the process of looking for a service?

If the app fails these simple checks, we stop and fix it immediately. This saves us time because we don't waste effort doing deep testing on a broken version.

Alpha Testing :  
Alpha testing is the first time we use the complete app in a realistic way, but it's done only by our own team. We pretend to be customers, workers, and admins. We try to use the app like we would in real life, and we try to find problems.

Our team will test things like:

* What happens if you try to book a service for a date that has already passed?
* Is it easy to find the button to change your password?
* What if the internet connection drops during payment?

We find and fix problems during alpha testing so that when real people use the app later, they won't find obvious and annoying bugs.

Beta Testing :

Beta testing is the last testing phase. We give a nearly finished version of DoorBell to a small group of real users—actual homeowners, real service workers, and a real manager. They use it in their daily lives and tell us what works and what doesn't.

We ask for their honest feedback on things like:

* “The steps to book a service were confusing.”
* “It was hard to see all my upcoming jobs.”
* “The app was very slow when I tried to use it in the evening.”

This step is the most important for making the app user-friendly. The developers might think something is easy, but real users might find it difficult. Their feedback helps us make the final improvements so that DoorBell is truly ready and helpful for everyone when it officially launches.

**5.2 Test case :**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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. 4. 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\_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: User has forgotten their password | | | | | |
| Dependencies: if any | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Go to the password reset page. 2. Enter the registered email   Address.   1. Click on reset password. 2. Check email for reset link. 3. Enter new password. | Email: user@gmail.  com.  New Password: new1234. | User should see the reset instructions.  Password should be reset successfully. | | 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 | | Text designed by : Rittik Chandra dey | | |
| Test case : TC\_04 | | This designed date : | | |
| Test Priority (Low, Medium, High): High | | Test Executed by : Rittik Chandra dey | | |
| 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 : Rittik Chandra dey | | |
| Test case : TC\_05 | | This designed date : | | |
| Test Priority (Low, Medium, High): Medium | | Test Executed by : Rittik Chandra dey | | |
| 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 : Rittik Chandra dey | | |
| Test case : TC\_06 | | This designed date : | | |
| Test Priority (Low, Medium, High): Medium | | Test Executed by : Rittik Chandra dey | | |
| 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 : Ashif | | |
| Test case :TC -07 | | | 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: Mahfujur Rahaman | | |
| Test Case ID:TC\_08 | | | 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 : Abdul Alim | | |
| Test case :TC-09 | | | This designed date : 23-12-2025 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by : A. Alim | | |
| Module Name : Update Profile | | | Test Execution date : 05-01-2026 | | |
| Test Title: Verify Update Profile functionality with valid and invalid user information | | | | | |
| Description: This test case verifies that a logged-in user can successfully update profile information such as profile picture, username, phone number, email, birthday, gender, and password from the Update Profile page. | | | | | |
| Precondition: 1. User must be logged in. 2. User must be on the Update Profile page. 3. Internet connection must be available | | | | | |
| Dependencies: None | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Open Update Profile page | \_\_ | Update Profile page loads successfully | | As expected | Pass |
| 2. Select profile picture | Valid image (.jpg/.png) | Profile picture should be selected | | As expected | Pass |
| 3. Click Upload button | \_\_ | Profile picture should upload | | As expected | Pass |
| 4. Enter User Name | Abdul Alim | User name should be updated | | As expected | Pass |
| 5. Enter Phone Number | 01747161672 | Phone number should be accepted | | As expected | Pass |
| 6. Enter Gmail | user2912@gmail.com | Valid email should be accepted | | As expected | Pass |
| 7. Select Birthday | 0171234567 | Birthday should be selected | | As expected | Pass |
| 8. Select Gender | Male/Female | Gender should be selectable | | As expected | Pass |
| 9. Enter Password | \*\*\*\*\*\*\*\* | Passwords should match | | As expected | Pass |
| 10. Click Update Profile button | \_\_ | Profile should be updated successfully | | As expected | Pass |
| 11. Click Back button | |  | | --- | |  |  |  | | --- | | \_\_ | | User should be redirected to previous page | | As expected | Pass |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Doorbell | | | Test designed by : Abdul Alim | | |
| Test case : TC -10 | | | This designed date : 23-12-2025 | | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by : A. Alim | | |
| Module Name : Profile Panel | | | Test Execution date : 05-01-2026 | | |
| Test Title : Verify Profile Panel displays correct user information. | | | | | |
| Description: This test case verifies that the Profile Panel page correctly displays the user’s profile information such as profile picture, user name, role, phone number, email, birthday, and gender. This page is view-only and does not allow editing. | | | | | |
| Precondition: 1. User must be logged in. 2. User must be on the Profile Panel page. 3. Internet connection must be available | | | | | |
| Dependencies: None | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Open Profile Panel page | \_\_ | Profile Panel page should load successfully | | As expected | Pass |
| 2. View profile picture | Existing profile image | Profile picture should be displayed properly | | As expected | Pass |
| 3. View User Name | Abdul Alim | User name should be displayed correctly | | As expected | Pass |
| 4. View User Role | Admin | User role should be displayed | | As expected | Pass |
| 5. View Phone Number | 01747161672 | Phone number should be displayed | | As expected | Pass |
| 7. View Gmail | User2912@gmail.com | Email address should be displayed | | As expected | Pass |
| 8. View Birthday | 29/12/2000 | Birthday should be displayed | | As expected | Pass |
| 9. View Gender | Male / Female | Gender should be displayed | | As expected | Pass |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Doorbell | | | Test designed by : Abdul Alim | | |
| Test case : TC -11 | | | This designed date : 23-12-2025 | | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by : A. Alim | | |
| Module Name : Profile Details | | | Test Execution date : 05-01-2026 | | |
| Test Title: Verify Profile Details page displays complete and correct user information | | | | | |
| Description: This test case verifies that the Profile Details page correctly displays all detailed user information including profile picture, user name, role, phone number, email address, birthday, and gender. The page is read-only and does not allow modification of any data. | | | | | |
| Precondition: 1. User must be logged in. 2. User must navigate to the Profile Details page 3. Internet connection must be available | | | | | |
| Dependencies: None | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Open Profile Details page | \_\_ | Profile Details page should load successfully | | As expected | Pass |
| 2. View profile picture | Existing profile image | Profile picture should be displayed clearly | | As expected | Pass |
| 3. View User Name | Abdul Alim | User name should be displayed correctly | | As expected | Pass |
| 4. View User Role | Admin | User role should be displayed correctly | | As expected | Pass |
| 5. View Phone Number | 01747161672 | Phone number should be displayed correctly | | As expected | Pass |
| 6. View Gmail | user2912@gmail.com | Email address should be displayed correctly | | As expected | Pass |
| 7. View Birthday | 29/12/2000 | Birthday should be displayed correctly | | As expected | Pass |
| 8. View Gender | Male/Female | Gender should be displayed correctly | | As expected | Pass |
| 9. Scroll page | \_\_ | All profile details should remain visible and readable | | As expected | Pass |
| 10. Click Back button | \_\_ | User should be redirected to previous page | | As expected | Pass |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Doorbell | | | Test designed by : Tazmun Naher | | |
| Test case : TC -12 | | | This designed date : 23-12-2025 | | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by : Tazmun Naher | | |
| Module Name : Service Panel – Payment Method | | | Test Execution date : 05-01-2026 | | |
| Test title : Verify adding payment method with valid details | | | | | |
| Description : Test the payment method add/update functionality in Service Panel | | | | | |
| Precondition: User is logged in | | | | | |
| Dependencies: None | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Login to Service Panel | Username: admin Password: \*\*\*\*\*\*\*\* | Dashboard opens successfully | | successfully | Pass |
| |  |  | | --- | --- | | 2.Navigate to"Payment Method" section |  | | \_\_ | |  |  | | --- | --- | | Payment Method page opens successfully |  | | | successfully | Pass |
| |  |  | | --- | --- | | 3. Click "Add Payment Method" |  | | \_\_ | Add Payment Method form appears | | accepted | Pass |
| |  |  | | --- | --- | | 4. Enter  card number |  | | 1234 5678 9012 3456 | Card number accepted | | accepted | Pass |
| 5. Enter expiry date | 12/29 | Expiry date accepted | | accepted | Pass |
| 6. Enter CVV | 123 | CVV accepted | | accepted | Pass |
| 7. Enter  cardholder name | Tazmun Naher | Cardholder name accepted | | accepted | Pass |
| 8. Click "Save" | \_\_ | Payment method added; confirmation appears | | accepted | Pass |
| 9. Verify  added payment method | \_\_ | Newly added payment method is displayed | | accepted | Pass |
| 10. Update payment method (optional) | Update card number to 4321 8765 2109 6543 | Changes saved successfully; confirmation appears | | Card update | Pass |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Doorbell | | | Test designed by : Tazmun Naher | | |
| Test case : TC-13 | | | This designed date : 23-12-2025 | | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by : Tazmun Naher | | |
| Module Name : Payment Details | | | Test Execution date : 05-01-2026 | | |
| Test Title : Verify user can enter payment details, email, and phone number in Payment Details panel. | | | | | |
| Description: Test the functionality of entering payment details, email, and phone number for a user browsing service categories. | | | | | |
| Precondition: User is logged in and browsing service categories | | | | | |
| Dependencies: Service categories available | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Login to Payment Details panel | Username: user1 Password: \*\*\*\*\*\*\*\* | Dashboard opens successfully | | successfully | Pass |
| |  |  | | --- | --- | | 2. Browse through service categories |  | | \_\_ | List of service categories is displayed | | successfully | Pass |
| 3. Select a service category | Category: Plumbing | |  |  | | --- | --- | | Selected service details displayed |  | | | successfully | Pass |
| |  |  | | --- | --- | | 4. Click "Add Payment Details" |  | | \_\_ | |  |  | | --- | --- | | Add Payment Details form appears |  | | | appeared | Pass |
| 5. Enter card number | 1234 5678 9012 3456 | Card number accepted | | accepted | Pass |
| 6. Enter expiry date | 12/29 | Expiry date accepted | | accepted | Pass |
| 7. Enter CVV | 123 | CVV accepted | | accepted | Pass |
| 8. Enter cardholder name | |  |  | | --- | --- | | User One |  | | Cardholder name accepted | | accepted | Pass |
| 9. Enter email | User123@gmail.com | Email accepted | | accepted | Pass |
| 10. Enter phone number | 01784840535 | Phone number accepted | | accepted | Pass |
| 11. Click "Save" | \_\_ | Payment details, email, and phone number saved; confirmation displayed | | accepted | Pass |
| 12. Verify saved details | Saved payment details, email, and phone number are displayed correctly | Description should be accepted | | accepted | Pass |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name:Doorbell | | | Test Designed by: Tazmun Naher akhe. | | |
| Test case ID:TC-14 | | | 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 |

**5.2 PROJECT ESTIMATION AND SCHEDULING**

**5.2.1 Effort and Cost Estimation**

Project Scope (before Estimation)

DoorBell is an all-in-one desktop application designed to bridge the gap between homeowners and trusted local service providers. The system addresses the challenges of finding reliable services, managing schedules, ensuring accountability, and maintaining service quality. It is built for three core user roles: Admin, Employee (Service Provider), and Customer (Homeowner).

* Admin functions include secure login, comprehensive user management (customers and service providers), service catalogue management (add/edit/delete services), real-time booking and transaction monitoring, system configuration (service categories, business hours, payment gateways), profile management, and reporting/analytics with data export capabilities (Excel/PDF).
* Employee (Service Provider) functions include secure login, viewing and managing assigned work orders with customer details and locations, updating job statuses, managing personal profiles and schedules, accessing service history, and secure logout.
* Customer (Homeowner) functions include user registration and secure login, browsing and searching for services by category (Cleaning, Gardening, Electrical, Plumbing, etc.), viewing detailed service descriptions, provider profiles, and transparent pricing, booking services with automated scheduling, making secure online payments, tracking booking status, providing ratings and reviews, and managing personal account details.

This scope defines a complete digital ecosystem that streamlines discovery, booking, execution, and management of household services, forming the precise foundation for software size and effort estimation.

**Lines of Code (LOC) Estimation**

Estimated size = 70,000 LOC = 70 KLOC.  
  
Assume productivity = 600 LOC/person-month.  
  
Effort = LOC / Productivity = 70,000 / 600 ≈ 116.7 person-months  
  
Person-hours (160 hrs/month): 116.7 × 160 ≈ 18,672 person-hours  
  
Thus, the LOC-based estimate is about 117 person-months (~18,700 hours).

## **5.2.2COCOMO Estimation (Semi-detached, 70 KLOC)**

A white rectangular box with black text

AI-generated content may be incorrect.

As our project is Semi-detached thus, Coefficients for Semi-detached projects:  
• Coefficient<Effort Factor> = 3.0  
• Project Complexity, P = 1.12  
• SLOC-dependent coefficient, T = 0.35  
  
**Step 1: Effort (PM)**  
Effort, PM = Coefficient<Effort Factor> × (LOC)^P  
Effort = 3.0 × (70,000/1000)^1.12 = 349.6 person-months  
  
**Step 2: Development Time (DM)**  
DM = 2.5 × (PM)^T  
TDEV = 2.5 × (349.6)^0.35 = 19.4 months **Step 3: Required number of people (ST)**  
ST = PM / DM = 349.6 / 19.4 = 18.02 ≈ 18 people

## Estimated Results

|  |  |  |  |
| --- | --- | --- | --- |
| Estimation Method | Effort (Person-Months) | Development Time | Average Staffing |
| LOC-based | 117 PM (~18,700 hours) | ~10–12 months | 9–10 people |
| COCOMO (Semi-detached, 70 KLOC) | 349.6 PM | 19.4 months | 18 people |

# **5.2.3 Earned Value Analysis (EVA)**

## Project Status (Reporting Point)

* Total Planned Tasks: 50 sprints
* BAC (Budget at Completion): 580 person-days
* Scheduled Completion (by this time): 7 sprints
* Tasks Completed So Far: 6 sprints

|  |
| --- |
| Sprint Effort Breakdown (Table) |
| |  |  |  | | --- | --- | --- | | Sprint | Planned Effort (days) | Actual Effort (days) | | 1 | 55 | 54 | | 2 | 55 | 57 | | 3 | 60 | 59 | | 4 | 60 | 62 | | 5 | 58 | 57 | | 6 | 56 | 56 | | 7 | 60 | N/A | | 8 | 56 | N/A | | 9 | 60 | N/A | | 10 | 60 | N/A | |

## EVA Key Metrics

* Total Tasks: 50 (sprints)
* BAC (Budget at Completion): 580 person-days
* BCWS (Planned Value): 404 days
* BCWP (Earned Value): 344 days
* ACWP (Actual Cost): 345 days

## EVA Interpretation

* Schedule Performance Index (SPI): SPI = BCWP / BCWS = 344 / 404 = 0.85 → Project is slightly behind schedule (SPI < 1).
* Schedule Variance (SV): SV = BCWP − BCWS = 344 − 404 = −60 → Negative → behind schedule.
* Cost Performance Index (CPI): CPI = BCWP / ACWP = 344 / 345 = 0.997 → Project is almost on budget (≈ 1).
* Cost Variance (CV): CV = BCWP − ACWP = 344 − 345 = −1 → Slight cost overrun.
* % Schedule for Completion: BCWS / BAC = 404 / 580 = 69.6% → About 70% of planned work should be done by now.
* % Complete: BCWP / BAC = 344 / 580 = 59.3% → About 59% of work is actually completed.

# **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.