A logo for a company

Description automatically generatedA logo with a book and a shield

Description automatically generated

**School of Computing and Information Technology**

Bachelor of Computer Science

CSIT314 - Software Development Methodologies

WipeOut

Final Report

19th May 2025

|  |  |
| --- | --- |
| **Members** | **UOW ID** |
| Chea Darayuth | 8550864 |
| Hein Htet Zaw | 8575939 |
| Ho Ka Yan Jeslyn | 8535383 |
| Kaung Minn Khant | 8551674 |
| Koung Khant Saung | 7877705 |
| Law Jun Wei | 8378411 |

**Taiga Link:** https://tree.taiga.io/project/jw\_law-sim2025q2\_wipeout/us/1?milestone=444461

**Table of Content**

[UML Diagram for Sprint 1 7](#_Toc149775178)

[Data Persistence Diagram for Sprint 1 7](#_Toc2098508310)

[Test Driven Development (TDD) 7](#_Toc2059611312)

[Sprint 2 7](#_Toc86087138)

[Meeting Minutes for Sprint 2 7](#_Toc300824007)

[Taiga Overview for Sprint 2 7](#_Toc2049796243)

[Gantt Chart Overview for Sprint 2 7](#_Toc1909658284)

[Test Plan for Sprint 2 7](#_Toc182503273)

[User Stories for Sprint 2 7](#_Toc438474479)

[[Sprint 2] Stakeholder #1 – User Administrator 7](#_Toc1499180271)

[[Sprint 2] Stakeholder #2 – Home Owner 7](#_Toc489433563)

[[Sprint 2] Stakeholder #3 – Cleaner 7](#_Toc352584847)

[[Sprint 2] Stakeholder #4 – Platform Manager 7](#_Toc1174086227)

[UML Diagram for Sprint 2 7](#_Toc1737478222)

[Sprint 3 7](#_Toc1898903808)

[Meeting Minutes for Sprint 3 7](#_Toc708317795)

[Taiga Overview for Sprint 3 7](#_Toc1990449658)

[Gantt Chart Overview for Sprint 3 7](#_Toc1114136555)

[Test Plan for Sprint 3 7](#_Toc265969072)

[User Stories for Sprint 3 7](#_Toc235614643)

[[Sprint 3] Stakeholder #1 – User Administrator 7](#_Toc1627914573)

[[Sprint 3] Stakeholder #2 – Home Owner 7](#_Toc1380823908)

[[Sprint 3] Stakeholder #3 – Cleaner 7](#_Toc381291293)

[[Sprint 3] Stakeholder #4 – Platform Manager 7](#_Toc1340906091)

[UML Diagram for Sprint 3 7](#_Toc155573953)

[Data Persistence Diagram for Sprint 3 7](#_Toc834260101)

[Sprint 4 7](#_Toc1250418457)

[Meeting Minutes for Sprint 4 7](#_Toc858240352)

[Taiga Overview for Sprint 4 7](#_Toc891349844)

[Gannt Chart Overview for Sprint 4 7](#_Toc889886332)

[Test Plan for Sprint 4 7](#_Toc573166256)

[User Stories for Sprint 4 7](#_Toc1237370256)

[[Sprint 4] Stakeholder #3 – Cleaner 7](#_Toc1997779032)

[[Sprint 4] Stakeholder #4 – Platform Manager 7](#_Toc1985474775)

[UML Diagram for Sprint 4 7](#_Toc1189103241)

[Data Persistence Diagram for Sprint 4 7](#_Toc1536274882)

[SQL Schema Table 7](#_Toc134558970)

[Test Data 7](#_Toc897360943)

[Agile Methodology 7](#_Toc1080585213)

[CI/CD 7](#_Toc723922510)

[Ethical Issues 7](#_Toc27424764)

[Data-Driven Software Development (DDD) 7](#_Toc454939360)

[Gantt Chart Overview 7](#_Toc1494197927)

**Member Contribution**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Full Name** | **UOW ID** | **Contribution** | **Contribution %** | **Signature** |
| Chea Darayuth | 8550864 | **For User Stories #UA-05, #UA-06, #UA-10, #UA-11, #CL-05, #CL-06:**   1. User Story 2. Use Case Description 3. BCE Diagram 4. Sequence Diagram 5. Wireframe 6. Test Case 7. Test Data 8. Programming   **For All:**   1. Data Persistence Diagram | 100 |  |
| Hein Htet Zaw | 8575939 | **For User Stories #UA-03, #UA-08, #HO-03, #HO-05, #HO-06, #HO-08,**  **#CL-03, #PM-03:**   1. User Story 2. Use Case Description 3. BCE Diagram 4. Sequence Diagram 5. Wireframe 6. Test Case 7. Test Data 8. Programming   **For All:**   1. Test Plan 2. Test Driven Development (TDD) 3. Continuous Integration/Continuous Deployment (CI/CD) | 100 |  |
| Ho Ka Yan Jeslyn | 8535383 | **For User Stories #UA-04, #UA-09, #CL-04, #CL-08, #CL-09, #CL-10,**  **#PM-04:**   1. User Story 2. Use Case Description 3. BCE Diagram 4. Sequence Diagram 5. Wireframe 6. Test Case 7. Test Data 8. Programming   **For All:**   1. Agile Methodology 2. Product Demo | 100 |  |
| Kaung Minn Khant | 8551674 | **For User Stories #PM-05, #PM-06, #PM-07, #PM-08, #PM-09, #PM-10:**   1. User Story 2. Use Case Description 3. Diagram Sequence Diagram 4. Wireframe 5. Test Case 6. Test Data 7. Programming   **For All:**   1. UML Class Diagram | 100 | A signature on a white background  AI-generated content may be incorrect. |
| Koung Khant Saung | 7877705 | **For User Stories #UA-07, #UA-12, #HO-04, #HO-07, #HO-09, #CL-07, #CL-11:**   1. User Story 2. Use Case Description 3. BCE Diagram 4. Sequence Diagram 5. Wireframe 6. Test Case 7. Test Data 8. Programming   **For All:**   1. Data-Driven Software Development (DDD) 2. Ethical Issues | 100 |  |
| Law Jun Wei | 8378411 | **For User Stories #UA-01, #UA-02, #HO-01, #HO-02, #CL-01, #CL-02**  **#PM-01, #PM-02:**   1. User Story 2. Use Case Description 3. BCE Diagram 4. Sequence Diagram 5. Wireframe 6. Test Case 7. Test Data 8. Programming   **For All:**   1. Meeting Minutes 2. Gannt Chart 3. Use Case Diagram | 100 | **A black line on a white background  AI-generated content may be incorrect.** |

**User Stories Overview**

|  |  |
| --- | --- |
| **User Administrator** | |
| UA-01 | As a User Administrator, I want to log in to the system so that I can access admin functionalities. |
| UA-02 | As a User Administrator, I want to log out of the system so that I can exit securely. |
| UA-03 | As a User Administrator, I want to create user accounts so that new users can access the platform. |
| UA-04 | As a User Administrator, I want to view all user accounts so that I can monitor the users. |
| UA-05 | As a User Administrator, I want to update user account information so that user data is accurate and current. |
| UA-06 | As a User Administrator, I want to suspend user accounts so that I can restrict access when necessary to maintain system security. |
| UA-07 | As a User Administrator, I want to search for user accounts so that I can quickly find specific users. |
| UA-08 | As a User Administrator, I want to create user profiles so that users have clearly defined roles. |
| UA-09 | As a User Administrator, I want to view all user profiles so that I can understand roles and permissions. |
| UA-10 | As a User Administrator, I want to update user profiles so that I can reflect changes in user roles. |
| UA-11 | As a User Administrator, I want to suspend user profiles so that I can temporarily restrict access. |
| UA-12 | As a User Administrator, I want to search for user profiles so that I can manage users easily. |

|  |  |
| --- | --- |
| **Home Owner** | |
| HO-01 | As a Home Owner, I want to log in to the system so that I can access services. |
| HO-02 | As a Home Owner, I want to log out of the system so that I can exit securely. |
| HO-03 | As a Home Owner, I want to view available cleaning services so that I can browse and evaluate service options. |
| HO-04 | As a Home Owner, I want to search for cleaning services so that I can find services that meet my needs. |
| HO-05 | As a Home Owner, I want to save cleaning services to a shortlist so that I can compare and select later. |
| HO-06 | As a Home Owner, I want to view my favourite list so that I can review the cleaning services I have shortlisted. |
| HO-07 | As a Home Owner, I want to search my favourite list so that I can find the shortlisted cleaning services easily. |
| HO-08 | As a Home Owner, I want to view my past cleaning services filtered by service and date, so that I can review all past bookings. |
| HO-09 | As a Home Owner, I want to search my cleaning history by service and date range, so that I can easily find a past booking. |

|  |  |
| --- | --- |
| **Cleaner** | |
| CL-01 | As a Cleaner, I want to log in to the system so that I can manage my services. |
| CL-02 | As a Cleaner, I want to log out of the system so that I can exit securely. |
| CL-03 | As a Cleaner, I want to create cleaning services so that Home Owners can book my services. |
| CL-04 | As a Cleaner, I want to view my cleaning services so that I can monitor what I offer. |
| CL-05 | As a Cleaner, I want to update my cleaning services so that I can ensure details are up to date. |
| CL-06 | As a Cleaner, I want to delete my cleaning services so that I can remove outdated ones. |
| CL-07 | As a Cleaner, I want to search my services so that I can find specific services efficiently. |
| CL-08 | As a Cleaner, I want to view how many times my profile was viewed so that I can track service visibility. |
| CL-09 | As a Cleaner, I want to view how many times I was shortlisted so that I can gauge my service popularity. |
| CL-10 | As a Cleaner, I want to view my confirmed match history filtered by services and date period so that I can review my completed jobs. |
| CL-11 | As a Cleaner, I want to search my confirmed match history filtered by services and date period so that I can efficiently find past jobs. |

|  |  |
| --- | --- |
| **Platform Manager** | |
| PM-01 | As a Platform Manager, I want to log in to the system so that I can access the platform management features. |
| PM-02 | As a Platform Manager, I want to log out of the system so that I can exit securely. |
| PM-03 | As a Platform Manager, I want to create cleaning services categories so that I can add new types of services to the platform. |
| PM-04 | As a Platform Manager, I want to view cleaning services categories so that I can monitor existing categories. |
| PM-05 | As a Platform Manager, I want to update cleaning services categories so that I can keep category information accurate and up to date. |
| PM-06 | As a Platform Manager, I want to delete cleaning services categories so that I can remove outdated or unused categories. |
| PM-07 | As a Platform Manager, I want to search cleaning services categories so that I can find specific categories quickly. |
| PM-08 | As a Platform Manager, I want to generate daily reports so that I can monitor short-term activity. |
| PM-09 | As a Platform Manager, I want to generate weekly reports so that I can track performance over the week. |
| PM-10 | As a Platform Manager, I want to generate monthly reports so that I can analyze long-term performance. |

**Sprint 1**

**Meeting Minutes for Sprint 1**

|  |  |
| --- | --- |
| **Meeting Minutes for Project Meeting 1** | **Taiga ID: #517** |
| WipeOut  Project Meeting 1  **Date / Time** **:** 08 April 2025 / 06.30pm  **Location :** Bench outside SIM Block B Classroom    **Present :** Chea Darayuth Team Member  Hein Htet Zaw Team Member  Ho Ka Yan Jeslyn Team Member  Kaung Minn Khant Team Member  Koung Khant Saung Team Member  Law Jun Wei Team Member  **Absent**  **:** -  **In Attendance :** -     |  |  |  | | --- | --- | --- | | **S/N** | **ITEM** | **ACTION** | | **A** | **Procedural (Opening)** |  | | **A1** | **Apologies for Absence** |  | | 1 | There were no absentees. |  | | **B1** | **Matters Arising** |  | | **1**  1.1  1.2  1.3  1.4  1.5  1.6  1.7  1.8  1.9  1.10  1.11 | **Overview of User Stories Tasks**  The team went through the tasks under each user story together, with JunWei sharing explanations and clarifications for the following.  Use Case Diagram  JunWei mentioned he would clarify with Mr. Terence, whether the team can consolidate use case diagrams by stakeholder (total of four diagrams) instead of creating one per user story. This change aims to save time and streamline efforts.  Use Case Description  JunWei explained that while Mr. Terence emphasized the Pre-condition, Normal Flow and Alternate Flow during tutorials, a complete use case description also requires fields such as Use Case Name, ID, Stakeholders, Use Case Goal, Description, Actors, Trigger point and sub flows. He went through all aspects using one user story as an example to ensure clarity.  Sequence Diagram  Since this was covered in class, JunWei only highlighted the need for consistent naming conventions across the Sequence Diagram and BCE Class Diagram.  Class Diagram (UML)  This task will be completed in a later project phase.  Class Diagram (BCE)  As this was also taught in class, JunWei reiterated the importance of consistent naming conventions between BCE and Sequence Diagram.  Data Persistence Diagram  To be done at a later stage of the project.  User Interface – Wireframe  Mr. Terence mentioned that instead of drawing wireframes, the team may submit screenshots of the final product as substitute.  Test Plan  Scheduled for a later stage.  Test Cases (in User Perspective)  To be done at a later stage.  Test Data (Legitimate Data)  The team will generate realistic test data using actual names later in the project.  Sample Work  JunWei shared that he will be producing a sample for the items mentioned above for one user story to serve as a reference for the rest of the team. |  | | **2** | **Git Repository**  Hein volunteered to set up the Git repository for the team |  | | **3** | **Task Division Plan**  The team discussed task distributions:   * JunWei proposed dividing the 42 use cases equally among 6 members (7 each). * Hein suggested splitting by task type (e.g. CRUDS) and mentioned he would assign and share the user stories in the group for everyone to choose their tasks before starting. |  | | **4** | **Technology Stack Decision**  The team decided to proceed with Java as the main programming language and MySQL as the database solution for the project. |  | | **5** | **Any Other Business**  There was no AOB. |  | | **C**  **1**  **2**  **3** | **Actions and Responsibilities**  JunWei to send a sample user story task as reference.  Hein to set up the Git repository for team collaboration.  All team members to complete their assigned user story tasks before the next meeting. | JunWei to note  Hein to note  All to note | | **D**  **1** | **Date of Next Meeting**  The next project meeting will be on 17 April 2025, 12.00pm, Online (Discord).  Meeting was adjourned at 7.40pm |  | |  | Recorded By: Law Jun Wei  Vetted By: Jeslyn Ho Ka Yan |  | | |

**Taiga Overview for Sprint 1**

|  |
| --- |
| **Sprint 1** |
| **A screenshot of a phone  AI-generated content may be incorrect.** |

**Gantt Chart Overview for Sprint 1**

|  |  |
| --- | --- |
| **Sprint 1** | **Taiga ID: #518** |
| **A screenshot of a project  AI-generated content may be incorrect.** | |

**Test Plan for Sprint 1**

|  |  |
| --- | --- |
| **Sprint 1** | **Taiga ID: #50, #91, #60, #70, #80, #101, #111** |
| **Introduction:**  The purpose of this test plan is to ensure the correct functionality of all the logins for user administrator, home owner, cleaner and platform manager. And we ensure each different user profile can ‘Create’ their respective objects. Testing is both manual and with the TDD script we provided.  **Objectives:**   1. Validates the functionality of the login feature for each user role. 2. Ensure each user role can ‘Create’ instances of classes. 3. Identify and document any bugs or issues encountered during.   **Resources:**  Test Environment: Local Environment Setup  **Testing Machine:**   1. Operating System:  Windows 11 Home, Version 24H2, OS Build 26100.4061 2. Processor: 11th Gen Intel(R) Core(™) i7-11370H @ 3.30GHz 3.30 GHz 3. Installed RAM: 16.0 GB 4. System Type: 64-bit operating system, x64-based processor     **Preconditions to run test:**  These need to be installed before running the test:   1. IntelliJ IDEA Community Edition 2025.1 2. MySQL Server (version 8.0.42) 3. Java (Eclipse temurin 22.0.2) 4. Java Fx (javafx-sdk-24.0.1) 5. JDBC-Connector (version 8.3.0)   **Test Completion Criteria:**  All test cases are accepted with no bugs or errors and actual test results match the expected results. The test cases will be rejected and re-done if any of the test cases were to fail.  This test plan aims to ensure the reliability and functionality of the login function for all users within the system. Through the execution of this test plan, we aim to identify and resolve any potential issues, ensuring a smooth and secure login experience for all users. | |

**User Stories for Sprint 1**

|  |  |
| --- | --- |
| **Sprint 1** | |
| UA-01 | As a User Administrator, I want to log in to the system so that I can access admin functionalities |
| UA-03 | As a User Administrator, I want to create user accounts so that new users can access the platform. |
| HO-01 | As a Home Owner, I want to log in to the system so that I can access services. |
| CL-01 | As a Cleaner, I want to log in to the system so that I can manage my services. |
| CL-03 | As a Cleaner, I want to create cleaning services so that Home Owners can book my services. |
| PM-01 | As a Platform Manager, I want to log in to the system so that I can access the platform management features. |
| PM-03 | As a Platform Manager, I want to create cleaning services categories so that I can add new types of services to the platform. |

**[Sprint 1] Stakeholder #1 – User Administrator**

|  |  |
| --- | --- |
| **User Story #UA-01** | **Taiga ID: #1** |
| As a User Administrator, I want to log in to the system so that I can access admin functionalities. | |

|  |  |
| --- | --- |
| **Use Case Diagram #UA-01** | **Taiga ID: #43** |
|  | |

|  |  |
| --- | --- |
| **Use Case Description #UA-01** | **Taiga ID: #44** |
| **Name:** Log In to User Administrator Account | |
| **Stakeholders and Goals:** User Administrator **–** wants to securely access admin functionalities by logging into the system. | |
| **Description:** The User Administrator wants to log in to the system to access admin functionalities. | |
| **Actors:** User Administrator | |
| **Trigger:** User Administrator selects their user profile and clicks on the “Login” button. | |
| **Pre-Condition:** The user must have an active user administrator’s account with valid login credentials. | |
| **Normal Flow:**   1. User selects their user profile as “User Administrator”, enters their login credentials and clicks the “Login” button. 2. System verifies the entered credentials against the database and if valid, logs the user into the system and grants access to the admin dashboard. 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**    **2a. Invalid Credentials:** If the user enters invalid credentials (e.g., wrong username or password), the system displays an error message stating “wrong username/password” and prompts the user to retry or reset their password. | |

|  |  |
| --- | --- |
| **BCE Diagram #UA-01** | **Taiga ID: #45** |
| A close-up of a login controller  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Sequence Diagram #UA-01** | **Taiga ID: #46** |
|  | |

|  |  |
| --- | --- |
| **Wireframe #UA-01** | **Taiga ID: #49** |
| **Login Page**    **User Administrator Dashboard** | |

|  |  |
| --- | --- |
| **Test Case #UA-01** | **Taiga ID: #51** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC  #UA-01  -1 | Start Login Page | NIL | Login Page appears | Login Page appears | pass | - | - | | TC  #UA-01  -2 | Enter valid username and password | Username = useradmin1, Password = UA1\_password | Login success, admin dashboard appears | Login success, admin dashboard appears | pass | - | - | | TC  #UA-01  -3 | Enter invalid credentials | Username = useradmin1, Password = wrongpassword | Error message: “Invalid Login Credentials! Please try again” displayed | Error message: “Invalid Login Credentials! Please try again” displayed | pass | - | - | | |

|  |  |
| --- | --- |
| **User Story #UA-03** | **Taiga ID: #3** |
| As a User Administrator, I want to create user accounts so that new users can access the platform. | |

|  |  |
| --- | --- |
| **Use Case Diagram #UA-03** | **Taiga ID: #84** |
| A diagram of a diagram  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #UA-03** | **Taiga ID: #85** |
| **Name:** Create User Account | |
| **Stakeholders and Goals:** User Administrator – wants to create new user accounts so that users can access and use the platform based on their assigned roles. | |
| **Description:** The User Administrator logs into the system and navigates to the User Administrator Dashboard to create a new user account. The administrator enters the necessary details. Upon successful submission, the system saves the new user account into the database. | |
| **Actors:** User Administrator | |
| **Trigger:** User Administrator clicks on the “Create New User Account” button. | |
| **Pre-Condition:**   1. Must be logged in as User Administrator. 2. Must provide valid user details. | |
| **Normal Flow:**   1. The User Administrator clicks on the "Create New User Account" button. 2. The system displays the Create User Account form. 3. The User Administrator fills in all required fields and clicks the "Create Account" button. 4. The system validates the input, saves the new user account to the database, and displays a confirmation message: "User account created successfully!" 5. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**    **2a. Input Validation or Account Creation Fails:** If the input is invalid or account creation fails due to business or system rules, the system displays an appropriate error message stating "Text Fields cannot be empty", “Username already existed", etc. | |

|  |  |
| --- | --- |
| **BCE Diagram #UA-03** | **Taiga ID: #86** |
|  | |

|  |  |
| --- | --- |
| **Sequence Diagram #UA-03** | **Taiga ID: #87** |
|  | |

|  |  |
| --- | --- |
| **Wireframe #UA-03** | **Taiga ID: #90** |
| **Create User Account Page** | |

|  |  |
| --- | --- |
| **Test Case #UA-03** | **Taiga ID: #92** |
| |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | | **Screenshot if fail** | | | TC  #UA-3  -1 | Start on User Management Page | Logged in as User Admin | "Create New User Account" button is visible | "Create New User Account" button is visible | Pass | - | | - | | | TC  #UA-3  -2 | Enter valid user details | Username = "user100", Password = "pass123", Full Name = "Test User", ProfileID = 2 | Confirmation: "User account created successfully!" | Confirmation: "User account created successfully!" | Pass | - | | - | | | TC  #UA-3  -3 | Submit with empty fields | Username = "", Password = "", Full Name = "", ProfileID = "" | Error: "Text fields cannot be empty" | Error: "Text fields cannot be empty" | Pass | - | | - | | | TC  #UA-3  -4 | Use duplicate username | Username = "cleaner1" (already exists) | Error: "Username already exists" | Error: "Username already exists" | Pass | - | | - | | | TC  #UA-3  -5 | Invalid profile ID | ProfileID = "abc" | Error: "Profile ID must be a valid number" | Error: "Profile ID must be a valid number" | Pass | | - | | - | | |

**[Sprint 1] Stakeholder #2 – Home Owner**

|  |  |
| --- | --- |
| **User Story #HO-01** | **Taiga ID: #13** |
| As a Home Owner, I want to log in to the system so that I can access services. | |

|  |  |
| --- | --- |
| **Use Case Diagram #HO-01** | **Taiga ID: #53** |
| A diagram of a logo  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #HO-01** | **Taiga ID: #54** |
| **Name:** Log In to Home Owner Account | |
| **Stakeholders and Goals:** Home Owner **–** wants to securely log in to access home cleaning and related services. | |
| **Description:** The Home Owner logs in to the system to gain access to available services in the system. | |
| **Actors:** Home Owner | |
| **Trigger:** Home Owner selects their user profile and clicks on the “Login” button. | |
| **Pre-Condition:** The user must have an active home owner’s account with valid login credentials. | |
| **Normal Flow:**   1. User selects their user profile as “Home Owner”, enters their login credentials and clicks the “Login” button. 2. System verifies the entered credentials against the database and if valid, logs the user into the system and grants access to the admin dashboard. 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**    **2a. Invalid Credentials:** If the user enters invalid credentials (e.g., wrong username or password), the system displays an error message stating "wrong username/password" and prompts the user to retry or reset their password.  **2b. Account Suspended:** If the user’s account is suspended, the system displays a message: “Your account has been suspended. Please contact support.” | |

|  |  |
| --- | --- |
| **BCE Diagram #HO-01** | **Taiga ID: #55** |
| A close-up of a login controller  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Sequence Diagram #HO-01** | **Taiga ID: #56** |
| A diagram of a flowchart  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Wireframe #HO-01** | **Taiga ID: #59** |
| **Login Page**    **Home Owner Dashboard** | |

|  |  |
| --- | --- |
| **Test Case #HO-01** | **Taiga ID: #61** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC  #HO-01  -1 | Start Login Page | NIL | Login Page appears | Login Page appears | Pass | - | - | | TC  #HO-01  -2 | Enter valid username and password | Username = homeowner1, Password = HO1\_password | Login success, home owner dashboard appears | Login success, home owner dashboard appears | Pass | - | - | | TC  #HO-01  -3 | Enter invalid credentials | Username = homeowner1, Password = wrongpassword | Error message: “Invalid Login Credentials! Please try again” displayed | Error Message: “Invalid Login Credentials! Please try again” displayed | Pass | - | - | | TC  #HO-01  -4 | Enter credentials for suspended account | Username = homeowner2, Password = HO2\_password | Error message: “Your account has been suspended. Please contact support.” | Error Message: “Your account has been suspended! Please contact support” | Pass | - | - | | |

**[Sprint 1] Stakeholder #3 – Cleaner**

|  |  |
| --- | --- |
| **User Story #CL-01** | **Taiga ID: #22** |
| As a Cleaner, I want to log in to the system so that I can manage my services. | |

|  |  |
| --- | --- |
| **Use Case Diagram #CL-01** | **Taiga ID: #63** |
| A diagram of a login  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #CL-01** | **Taiga ID: #64** |
| **Name:** Log In to Cleaner Account | |
| **Stakeholders and Goals:** Cleaner **–** wants to securely log in to manage home cleaning and related services. | |
| **Description:** The Cleaner logs in to the system to manage their services through the system. | |
| **Actors:** Cleaner | |
| **Trigger:** Cleaner selects their user profile and clicks on the “Login” button. | |
| **Pre-Condition:** The user must have an active cleaner’s account with valid login credentials. | |
| **Normal Flow:**   1. User selects their user profile as “Cleaner”, enters their login credentials and clicks the “Login” button. 2. System verifies the entered credentials against the database and if valid, logs the user into the system and grants access to the admin dashboard. 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**    **2a. Invalid Credentials:** If the user enters invalid credentials (e.g., wrong username or password), the system displays an error message stating "wrong username/password" and prompts the user to retry or reset their password.  **2b. Account Suspended:** If the user’s account is suspended, the system displays a message: “Your account has been suspended. Please contact support.” | |

|  |  |
| --- | --- |
| **BCE Diagram #CL-01** | **Taiga ID: #65** |
| A close-up of a login controller  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Sequence Diagram #CL-01** | **Taiga ID: #66** |
| A diagram of a software project  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Wireframe #CL-01** | **Taiga ID: #69** |
| **Login Page**    **Cleaner Dashboard** | |

|  |  |
| --- | --- |
| **Test Case #CL-01** | **Taiga ID: #71** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC  #CL-01  -1 | Start Login Page | NIL | Login Page appears | Login Page appears | Pass | - | - | | TC  #CL-01  -2 | Enter valid username and password | Username = cleaner1, Password = CL1\_password | Login success, cleaner dashboard appears | Login success, cleaner dashboard appears | Pass | - | - | | TC  #CL-01  -3 | Enter invalid credentials | Username = cleaner1, Password = wrongpassword | Error Message: “Invalid Login Credentials! Please try again” displayed | Error Message: “Invalid Login Credentials! Please try again” displayed | Pass | - | - | | TC  #CL-01  -4 | Enter credentials for suspended account | Username = cleaner2, Password = CL2\_password | Error Message: “Your account has been suspended! Please contact support” | Error Message: “Your account has been suspended! Please contact support” | Pass | - | - | | |

|  |  |
| --- | --- |
| **User Story #CL-03** | **Taiga ID: #24** |
| As a Cleaner, I want to create cleaning services so that Home Owners can book my services. | |

|  |  |
| --- | --- |
| **Use Case Diagram #CL-03** | **Taiga ID: #94** |
| A black and white diagram with text  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #CL-03** | **Taiga ID: #95** |
| **Name:** Create Cleaning Service | |
| **Stakeholders and Goals:** Cleaner – wants to list new cleaning services on the platform so that Home Owners can view and book them. | |
| **Description:** The Cleaner logs into the system and navigates to the Cleaner Dashboard to create a new service. The Cleaner provides details such as the service title, description, and price. Once submitted, the system validates and stores the new cleaning service for public viewing. | |
| **Actors:** Cleaner | |
| **Trigger:** Cleaner clicks on the “Create Cleaning Service” button. | |
| **Pre-Condition:**   1. Must be logged in as Cleaner. 2. Must provide valid cleaning service details. | |
| **Normal Flow:**   1. The Cleaner clicks on the “Create Cleaning Service” button. 2. The system displays the Create Cleaning Service form. 3. The Cleaner fills in all required fields and clicks on the “Create Cleaning Service” button. 4. The system validates the input, saves the new service to the database, and displays a success message: “Service created successfully!” 5. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  **2a. Invalid Input or Business Rule Violations:** If the Cleaner enters invalid inputs or violates business rules when creating the service, the system will display appropriate messages stating “Invalid price”, “Title too long”, etc. | |

|  |  |
| --- | --- |
| **BCE Diagram #CL-03** | **Taiga ID: #96** |
|  | |

|  |  |
| --- | --- |
| **Sequence Diagram #CL-03** | **Taiga ID: #97** |
|  | |

|  |  |
| --- | --- |
| **Wireframe #CL-03** | **Taiga ID: #100** |
| **Create Cleaning Service Page** | |

|  |  |
| --- | --- |
| **Test Case #CL-03** | **Taiga ID: #102** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC  #CL-03  -1 | Start Create Cleaning Service Page | NIL | Form page is displayed | Form page is displayed | Pass | - | - | | TC  #CL-03  -2 | Submit valid input | Title = newService Desc = Eco-friendly cleaning Price = 88 | Service created successfully! | Service created successfully! | Pass | - | - | | TC  #CL-03  -3 | Invalid price (non-numeric) | Title = testService Desc = Spring cleaning Price = abc | Error: “Invalid price.” | Error: “Invalid price.” | Pass | - | - | | TC  #CL-03  -4 | Title too long | Title = aaaaaaaaaaaaaaaaaaaaa (21 chars) Desc = Deep clean Price = 55 | Error: “Title too long (max 20 chars).” | Error: “Title too long (max 20 chars).” | Pass | - | - | | TC  #CL-03  -5 | Price zero | Title = QuickClean Desc = Basic cleaning Price = 0 | Error: “Price must be > 0.” | Error: “Price must be > 0.” | Pass | - | - | | TC  #CL-03  -6 | Price negative | Title = BudgetClean Desc = Affordable cleaning Price = -10 | Error: “Price must be > 0.” | Error: “Price must be > 0.” | Pass | - | - | | TC  #CL-03  -7 | Simulate DB error | Title = SimDBFail Desc = Error trigger Price = 55 | Error: Database error occurred. | Error: Database error occurred. | Pass | - | - | | |

**[Sprint 1] Stakeholder #4 – Platform Manager**

|  |  |
| --- | --- |
| **User Story #PM-01** | **Taiga ID: #33** |
| As a Platform Manager, I want to log in to the system so that I can access the platform management features. | |

|  |  |
| --- | --- |
| **Use Case Diagram #PM-01** | **Taiga ID: #73** |
| A diagram of a person's relationship  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #PM-01** | **Taiga ID: #74** |
| **Name:** Log In to Platform Manager Account | |
| **Stakeholders and Goals:** Platform Manager **–** wants to securely log in to access platform management features. | |
| **Description:** The platform manager logs in to the system to access tools for overseeing and managing the platform. | |
| **Actors:** Platform Manager | |
| **Trigger:** Platform Manager selects their user profile and clicks on the “Login” button. | |
| **Pre-Condition:** The user must have an active platform manager’s account with valid login credentials. | |
| **Normal Flow:**   1. User selects their user profile as “Platform Manager”, enters their login credentials and clicks the “Login” button. 2. System verifies the entered credentials against the database and if valid, logs the user into the system and grants access to the admin dashboard. 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**    **2a. Invalid Credentials:** If the user enters invalid credentials (e.g., wrong username or password), the system displays an error message stating "wrong username/password" and prompts the user to retry or reset their password.  **2b. Account Suspended:** If the user’s account is suspended, the system displays a message: “Your account has been suspended. Please contact support.” | |

|  |  |
| --- | --- |
| **BCE Diagram #PM-01** | **Taiga ID: #75** |
| A close-up of a login controller  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Sequence Diagram #PM-01** | **Taiga ID: #76** |
| A diagram of a software project  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Wireframe #PM-01** | **Taiga ID: #79** |
| **Login Page**    **Platform Manager Dashboard** | |

|  |  |
| --- | --- |
| **Test Case #PM-01** | **Taiga ID: #81** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC  #PM-01  -1 | Start Login Page | NIL | Login Page appears | Login Page appears | Pass | - | - | | TC  #PM-01  -2 | Enter valid username and password | Username = platformmanager1, Password = PM1\_password | Login success, platform manager dashboard appears | Login success, platform manager dashboard appears | Pass | - | - | | TC  #PM-01  -3 | Enter invalid credentials | Username = platformmanager1, Password = wrongpassword | Error Message: “Invalid Login Credentials! Please try again” displayed | Error Message: “Invalid Login Credentials! Please try again” displayed | Pass | - | - | | TC  #PM-01  -4 | Enter credentials for suspended account | Username = platformmanager2, Password = PM2\_password | Error Message: “Your account has been suspended! Please contact support” | Error Message: “Your account has been suspended! Please contact support” | Pass | - | - | | |

|  |  |
| --- | --- |
| **User Story #PM-03** | **Taiga ID: #1** |
| As a Platform Manager, I want to create cleaning services categories so that I can add new types of services to the platform. | |

|  |  |
| --- | --- |
| **Use Case Diagram #PM-03** | **Taiga ID: #104** |
| A diagram of a service  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #PM-03** | **Taiga ID: #105** |
| **Name:** Create Service Category | |
| **Stakeholders and Goals:** Platform Manager – wants to add new cleaning service categories to the system so that services can be properly grouped and organized. | |
| **Description:** The Platform Manager logs into the system and navigates to the Platform Manager Dashboard to create a new service category. The manager provides the necessary information and the system validates and stores it for future use. | |
| **Actors:** Platform Manager | |
| **Trigger:** Platform Manager clicks on the “Create Service Category” button. | |
| **Pre-Condition:**   1. Must be logged in as Platform Manager 2. Category details are prepared and ready to be entered | |
| **Normal Flow:**   1. The Platform Manager clicks on the “Create Service Category” button. 2. The system displays the Create Service Category form. 3. The Platform Manager fills in the required fields and clicks on “Create”. 4. The system validates the input, saves the new category to the database, and displays a confirmation message: “Category created successfully.” 5. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**    **2a. Invalid Input or Creation Fails:** If the Platform Manager enters invalid inputs or creation fails due to system rules, the system will display the following messages stating “Fields cannot be empty”, “Category name already existed”, etc. | |

|  |  |
| --- | --- |
| **BCE Diagram #PM-03** | **Taiga ID: #106** |
|  | |

|  |  |
| --- | --- |
| **Sequence Diagram #PM-03** | **Taiga ID: #107** |
|  | |

|  |  |
| --- | --- |
| **Wireframe #PM-03** | **Taiga ID: #110** |
| **Create Service Category Page**    **Create Service Category Page – After Create** | |

|  |  |
| --- | --- |
| **Test Case #PM-03** | **Taiga ID: #112** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC  #PM-03  -1 | Start Create Service Page | NIL | Create Service Page appears | Create Service Page appears | Pass | - | - | | TC  #PM-03  -2 | Submit valid input | Name= category101, Description = This is a new test category. | Category created successfully! | Category created successfully! | Pass | - | - | | TC  #PM-03  -3 | Leave name blank | Name = Description = Dusting services | Error: Category name cannot be empty. | Error: Category name cannot be empty. | Pass | - | - | | TC  #PM-03  -4 | Leave description blank | Name = TestCategory , Description =- | Error: Category description cannot be empty. | Error: Category description cannot be empty. | Pass | - | - | | TC  #PM-03  -5 | Use existing name | Name = category1 Description = Duplicate test | Error: Category name already exists. | Error: Category name already exists. | Pass | - | - | | TC  #PM-03  -6 | Name too long | Name = "A" \* 51 Description = Too long | Error: Category name too long (max 50 characters) | Error: Category name too long (max 50 characters) | Pass | - | - | | TC  #PM-03  -7 | Simulate DB error | Name = SimulatedError Description = Valid desc | Error: Database error occurred. | Error: Database error occurred. | Pass | - | - | | |

**UML Diagram for Sprint 1**

|  |  |
| --- | --- |
| **Sprint 1** | **Taiga ID: #47, #57, #67, #88, #77, #98, #108** |
| A screenshot of a computer program  AI-generated content may be incorrect. | |

**Data Persistence Diagram for Sprint 1**

|  |  |
| --- | --- |
| **Sprint 1** | **Taiga ID: #48, #58, #68, #78, #89, #99, #109** |
| **A screenshot of a computer  AI-generated content may be incorrect.**  **A screenshot of a computer  AI-generated content may be incorrect.**  **A screenshot of a computer  AI-generated content may be incorrect.**  **A black and white screen with white text  AI-generated content may be incorrect.**  **A black screen with white text  AI-generated content may be incorrect.**  **A black rectangular object with a black border  AI-generated content may be incorrect.** | |

**Test Driven Development (TDD)**

|  |  |  |
| --- | --- | --- |
| **Test Driven Development** | **Taiga ID: #519** | |
| The LoginTest class demonstrates the use of Test-Driven Development (TDD) to verify the correctness of the verifyLogin(...) function within the UserAccount entity. It ensures that user authentication works as expected across different roles and scenarios. We wrote unit tests before completing the full login logic, following the TDD cycle:   * Write test cases for expected behaviours and for different profiles. * Implement minimal code to pass the tests. * Refactor and validate again.   We tested 6 different cases:   1. Login as User Admin. 2. Login as Home Owner. 3. Login as Cleaner 4. Login as Platform Manager. 5. Login as a ‘Suspended’ User. 6. Login with invalid information.   The final test code is included inside the source code. Since we’re using Junit library to test, we need to add in dependency inside our pom.xml file. | | |
| **Test Driven Development** | **Taiga ID: #519** |
| The LoginTest class demonstrates the use of Test-Driven Development (TDD) to verify the correctness of the verifyLogin(...) function within the UserAccount entity. It ensures that user authentication works as expected across different roles and scenarios. We wrote unit tests before completing the full login logic, following the TDD cycle:   * Write test cases for expected behaviors and for different profiles. * Implement minimal code to pass the tests. * Refactor and validate again.   We tested 6 different cases,   1. Login as User Admin. 2. Login as Home Owner. 3. Login as Cleaner 4. Login as Platform Manager. 5. Login as a ‘Suspended’ User. 6. Login with invalid information.   The final test code is included inside the source code. Since we’re using Junit library to test, we need to add in dependency inside our pom.xml file. | |

**Sprint 2**

**Meeting Minutes for Sprint 2**

|  |  |
| --- | --- |
| **Meeting Minutes for Project Meeting 2** | **Taiga ID: #520** |
| WipeOut  Project Meeting 2  **Date / Time** **:** 17 April 2025 / 12.15pm  **Location :** e-Meeting on Discord    **Present :** Chea Darayuth Team Member  Hein Htet Zaw Team Member  Ho Ka Yan Jeslyn Team Member  Kaung Minn Khant Team Member  Koung Khant Saung Team Member  Law Jun Wei Team Member  **Absent**  **:** -  **In Attendance :** -     |  |  |  | | --- | --- | --- | | **S/N** | **ITEM** | **ACTION** | | **A** | **Procedural (Opening)** |  | | **A1** | **Apologies for Absence** |  | | 1 | There were no absentees. |  | | **B1** | **Matters Arising** |  | | **1** | **Sprint Planning on Taiga**  The team discussed sprint planning using Taiga. Sprint weeks were set and JunWei explained how user stories would be split across the sprints to manage the workload effectively. |  | | **2**  2.1  2.2  2.3 | **Task Volunteer Updates**  Data Persistence Diagram  Darayuth volunteered to take on this task.  UML Diagram  Kaung volunteered to handle this task.  Test Plan  Hein volunteered to prepare this deliverable. | Darayuth to note  Kaung to note  Hein to note | | **3** | **Coding Progress**  JunWei suggested that the group should complete a full set of code (including Boundary, Controller and Entity) for at least one user story. This would be presented during the next tutorial session for Mr. Terence to vet and ensure alignment with the use case. | All to note | | **4**  4.1  4.2  4.3 | **Question for Mr. Terence**  Should the Test Plan, UML Diagram and Data Persistence Diagram be completed before or after coding? before  For Sequence Diagrams, can alternate flows be combined if they share the same logic but differ only in scenario?  Can Use Case Diagrams be combined across multiple user stories if they share the same stakeholder? can |  | | **5** | **Any Other Business**  **JunWei** - reminded the team that the submitted user stories were already approved by Mr. Terence and their actors and actions are fixed. However, members may adjust the reasoning of a user story if it better supports their use case description. Any such changes should be communicated to JunWei for updates in the report. | All to note | | **C**  **1**  **2**  **3**  **4** | **Actions and Responsibilities**  Darayuth to work on Data Persistence Diagram  Kaung to work on UML Diagram  Hein to work on Test Plan  Team to continue on their user stories’ task and start working on the full set of code for at least one user story. | Darayuth to note  Kaung to note  Hein to note  All to note | | **D**  **1** | **Date of Next Meeting**  The next project meeting will be on 23 April 2025, 3.00pm, after tutorial.  Meeting was adjourned at 1.55pm |  | |  | Recorded By: Law Jun Wei  Vetted By: Jeslyn Ho Ka Yan |  | | |

|  |  |
| --- | --- |
| **Meeting Minutes for Project Meeting 3** | **Taiga ID: #521** |
| **WipeOut**  **Project Meeting 3**  **Date / Time** **:** 23 April 2025 / 2.30pm  **Location :** SIM HQ, Block D, Level 3    **Present :** Chea Darayuth Team Member  Ho Ka Yan Jeslyn Team Member  Kaung Minn Khant Team Member  Koung Khant Saung Team Member  Law Jun Wei Team Member  **Absent :** Hein Htet Zaw Team Member  **In Attendance :** -     |  |  |  | | --- | --- | --- | | **S/N** | **ITEM** | **ACTION** | | **A** | **Procedural (Opening)** |  | | **A1** | **Apologies for Absence** |  | | 1 | Hein was absent due to illness. |  | | **B1** | **Matters Arising** |  | | **1**  1.1  1.2  1.3 | **Response from Mr. Terence (from Previous Questions)**  Timeline for Key Diagrams  JunWei shared Mr. Terence’s feedback that the Test Plan, UML Class Diagram and Data Persistence Diagram should all be planned and completed before coding begins. This is to ensure clear direction during development.  Sequence Diagram – Alternate Flows  Mr. Terence advised that each alternate flow scenario should be drawn out separately, even if they share the same logic.  Combining Use Case Diagram  Use Case Diagrams can be combined by sprint, based on the user stories within that sprint. Stakeholders should still be clearly separated within each diagram.  Mr. Terence also noted that:   * The Test Plan should also be created sprint-wise. * UML Class and Data Persistence Diagrams should be updated cumulatively across sprints. For example, the diagram in sprint 2 should include both sprint 1 and 2 content. | All to note  All to note  All to note | | **2**  2.1  2.2 | **User Story Allocation and Sprint Deployment**  Splitting User Story Tasks Across Sprints  JunWei confirmed with Mr. Terence that user stories must be completed fully within the assigned sprint. Tasks from one user story cannot be split across different sprints.  After this clarification, the team agreed to redeploy user stories into respective sprints according to their feasibility and completeness.  New User Story Addition  Kaung highlighted that the Platform Manager report functionality (daily/weekly/monthly reports) could not be completed due to the absence of a service booking process.  After discussion, the team added the following user story:  ***HO-05: As a Home Owner, I want to book a cleaning service with a selected cleaner so that I can confirm and schedule the service.***  This new user story was assigned to Kaung | All to note  Kaung to note | | **3** | **Coding and Lab Preparation**  JunWei encouraged everyone to focus on preparing for the upcoming lab test.  After lab test preparation:   * Begin coding one assigned user story in full (Boundary, Controller and Entity). * Ensure the logic and flow aligns with the Use Case Description, BCE Diagram and Sequence Diagram for consistency. * Submit the coded sample to Mr. Terence for vetting during the next tutorial. Once given the greenlight, continue with the rest of the user stories. | All to note | | **4**  4.1 | **Any Other Business**  Admin Report Clarification  **Kaung** – raised concern about the lack of detail for the admin report. JunWei suggested Kaung to draft a sample report and present it to Mr. Terence for feedback during the next tutorial. Once approved, it can be used to complete the related tasks and coding. |  | | **C**  **1**  **2**  **3** | **Actions and Responsibilities**  Kaung to complete new user story (HO-05).  Kaung to draft a sample admin report.  Team to prepare for lab test and begin full implementation of one user story. | Kaung to note  Kaung to note  All to note | | **D**  **1** | **Date of Next Meeting**  The next project meeting is tentatively scheduled for 28 April 2025, 3.00pm, after tutorial.  Any changes to the meeting schedule will be informed via the Group Telegram Chat.  Meeting was adjourned at 4.30pm |  | |  | Recorded By: Law Jun Wei  Vetted By: Jeslyn Ho Ka Yan |  | | |

**Taiga Overview for Sprint 2**

|  |
| --- |
| **Sprint 2** |
| **A screenshot of a phone  AI-generated content may be incorrect.** |

**Gantt Chart Overview for Sprint 2**

|  |  |
| --- | --- |
| **Sprint 2** | **Taiga ID: #522** |
| **A screenshot of a calendar  AI-generated content may be incorrect.** | |

**Test Plan for Sprint 2**

|  |  |
| --- | --- |
| **Sprint 2** | **Taiga ID: #121, #131, #141, #151, #161, #171, #181, #191, #201, #211, #221, #231, #241, #251, #261, #271, #281, #291** |
| **Introduction:**  The purpose of this test plan is to ensure the correct functionality of all the log outs and the CRUDS for user administrator, home owner, cleaner and platform manager. We make sure all foreign key mappings don’t affect the update/delete aspects. Testing will be manual and based on database entries.  **Objectives:**  1. Validates the functionality of the log out feature for each user role.  2. Ensure CRUDS queries work across Entity and database.  3. Identify and document any bugs or issues encountered during.  **Resources:**  Test Environment: Local Environment Setup  Testing Machine:   1. Operating System:  Windows 11 Home, Version 24H2, OS Build 26100.4061 2. Processor: 11th Gen Intel(R) Core(™) i7-11370H @ 3.30GHz 3.30 GHz 3. Installed RAM: 16.0 GB 4. System Type: 64-bit operating system, x64-based processor   **Preconditions to run test:**  These need to be installed before running the test:   1. IntelliJ IDEA Community Edition 2025.1 2. MySQL Server (version 8.0.42) 3. Java (Eclipse temurin 22.0.2) 4. Java Fx (javafx-sdk-24.0.1) 5. JDBC-Connector (version 8.3.0)   **Test Completion Criteria:**  All test cases are accepted with no bugs or errors and actual test results match the expected results. The test cases will be rejected and re-done if any of the test cases were to fail.  This test plan aims to ensure the reliability and functionality of the log out function for all users within the system. Through the execution of this test plan, we aim to identify and resolve any potential issues, ensuring functionality of CRUDS operations for each user role. | |

**User Stories for Sprint 2**

|  |  |
| --- | --- |
| **Sprint 2** | |
| UA-02 | As a User Administrator, I want to log out of the system so that I can exit securely. |
| UA-04 | As a User Administrator, I want to view all user accounts so that I can monitor the users. |
| UA-05 | As a User Administrator, I want to update user account information so that user data is accurate and current. |
| UA-07 | As a User Administrator, I want to search for user accounts so that I can quickly find specific users. |
| UA-08 | As a User Administrator, I want to create user profiles so that users have clearly defined roles |
| UA-09 | As a User Administrator, I want to view all user profiles so that I can understand roles and permissions. |
| UA-10 | As a User Administrator, I want to update user profiles so that I can reflect changes in user roles. |
| UA-12 | As a User Administrator, I want to search for user profiles so that I can manage users easily. |
| HO-02 | As a Home Owner, I want to log out of the system so that I can exit securely. |
| HO-03 | As a Home Owner, I want to view available cleaning services so that I can browse and evaluate service options. |
| HO-05 | As a Home Owner, I want to save cleaning services to a shortlist so that I can compare and select later. |
| HO-06 | As a Home Owner, I want to view my favourite list so that I can review the cleaning services I have shortlisted. |
| HO-08 | As a Home Owner, I want to view my past cleaning services filtered by service and date, so that I can review all past bookings. |
| CL-02 | As a Cleaner, I want to log out of the system so that I can exit securely. |
| CL-04 | As a Cleaner, I want to view my cleaning services so that I can monitor what I offer. |
| CL-05 | As a Cleaner, I want to update my cleaning services so that I can ensure details are up to date. |
| PM-02 | As a Platform Manager, I want to log out of the system so that I can exit securely. |
| PM-05 | As a Platform Manager, I want to update cleaning services categories so that I can keep category information accurate and up to date. |

**[Sprint 2] Stakeholder #1 – User Administrator**

|  |  |
| --- | --- |
| **User Story #UA-02** | **Taiga ID: #2** |
| As a User Administrator, I want to log out of the system so that I can exit securely. | |

|  |  |
| --- | --- |
| **Use Case Diagram #UA-02** | **Taiga ID: #114** |
| A diagram of a logistic company  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #UA-02** | **Taiga ID: #115** |
| **Name:** Log Out of the User Administrator Account | |
| **Stakeholders and Goals:** User Administrator **–** wants to securely exit the system after completing administrative tasks. | |
| **Description:** The User Administrator intends to log out of the system to ensure that no unauthorised access can occur after their session ends. | |
| **Actors:** User Administrator | |
| **Trigger:** User Administrator clicks on the “Logout” button. | |
| **Pre-Condition:** The user must be logged into the system. | |
| **Normal Flow:**   1. User Administrator clicks the “Logout” button. 2. The system terminates the current session and securely logs the user out. 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

|  |  |
| --- | --- |
| **BCE Diagram #UA-02** | **Taiga ID: #116** |
| A screenshot of a browser window  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Sequence Diagram #UA-02** | **Taiga ID: #117** |
| A diagram of a user administrator  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Wireframe #UA-02** | **Taiga ID: #120** |
| **Logout Button on User Administrator Dashboard**    **After Logout – Redirect to Cleaning Services Platform Login Page** | |

|  |  |
| --- | --- |
| **Test Case #UA-02** | **Taiga ID: #122** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC  #UA-02  -1 | Click on the “Logout” button or icon | NIL | System logs out the system administrator | System logs out the system administrator | Pass | - | - | | |

|  |  |
| --- | --- |
| **User Story #UA-04** | **Taiga ID: #4** |
| As a User Administrator, I want to view all user accounts so that I can monitor the users. | |

|  |  |
| --- | --- |
| **Use Case Diagram #UA-04** | **Taiga ID: #124** |
| A diagram of a view user  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #UA-04** | **Taiga ID: #125** |
| **Name:** View All User Accounts | |
| **Stakeholders and Goals:** User Administrator – wants to view all user accounts to monitor user activity, ensure system compliance, and manage user access effectively. | |
| **Description:** The User Administrator logs into the system and navigates to the User Administrator Dashboard to view a list of all user accounts. This enables the user administrator to oversee user operations and maintain platform integrity. | |
| **Actors:** User Administrator | |
| **Trigger:** User Administrator clicks on the “View All User Accounts” button. | |
| **Pre-Condition:**   1. The User Administrator must be logged into the system 2. The account must have administrative privileges. | |
| **Normal Flow:**   1. The User Administrator clicks on the "View All User Accounts" button. 2. The system retrieves and displays a list of all user accounts. 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  **2a. No User Accounts Available:** If no user accounts are present in the database (e.g., if the database is empty or no users have been registered yet), the system will display a message. Message: "No users available to display."  **2b. Insufficient Privileges:** If the User Administrator lacks the appropriate privileges to view user accounts, the system will display a message. Message: "You do not have the required privileges." | |

|  |  |
| --- | --- |
| **BCE Diagram #UA-04** | **Taiga ID: #126** |
| Picture 1, Picture | |

|  |  |
| --- | --- |
| **Sequence Diagram #UA-04** | **Taiga ID: #127** |
| Picture 1, Picture | |

|  |  |
| --- | --- |
| **Wireframe #UA-04** | **Taiga ID: #130** |
| **View All User Accounts Page**  Picture 1, Picture | |

|  |  |
| --- | --- |
| **Test Case #UA-04** | **Taiga ID: #132** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC  #UA-04  -1 | Click “View All User Accounts” on dashboard | userID= 1,  username = useradmin1, profileStatus = User Admininstrator,  fullname = User 1,  accountStatus = Active | User account table is displayed with all user records. | User account table is displayed with all user records. | Pass | - | - | | TC  #UA-04  -2 | Table loads empty when no accounts exist | NIL (no accounts in DB) | Display: “No users available to display.” error message. | Display: “No users available to display.” error message. | Pass | Optional test if table is empty | - | | TC  #UA-04  -3 | Attempt to access with unauthorized role | Username = cleaner1  Password = CL1\_password | Error message: “You do not have the required privileges.” shown. | Error message: “You do not have the required privileges.” shown. | Pass | Cleaner should not access UA pages | - | | |

|  |  |
| --- | --- |
| **User Story #UA-05** | **Taiga ID: #5** |
| As a User Administrator, I want to update user account information so that user data is accurate and current. | |

|  |  |
| --- | --- |
| **Use Case Diagram #UA-05** | **Taiga ID: #134** |
| A diagram of a software update  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #UA-05** | **Taiga ID: #135** |
| **Name:** Update User Account Information | |
| **Stakeholders and Goals:** User Administrator – wants to ensure that user account information is accurate and up to date to maintain system integrity and effective user management. | |
| **Description:** The User Administrator logs into the system and navigates to the User Manager Dashboard to update user account information. The changes are saved and confirmed by the system to ensure all user data remains accurate and current. | |
| **Actors:** User Administrator | |
| **Trigger:** User Administrator clicks on the “Update User Account” button. | |
| **Pre-Condition:**   1. The User Administrator must be logged into the system with administrative privileges. 2. The target user account must exist in the system. | |
| **Normal Flow:**   1. The User Administrator selects a user account, makes changes to the information, and clicks the "Update" button. 2. The system stores the updated information and returns a confirmation message along with the updated user details. 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  **2a. Blank Fields:** If any required fields are left blank, the system will displaymessage stating “Fields cannot be empty.” | |

|  |  |
| --- | --- |
| **BCE Diagram #UA-05** | **Taiga ID: #136** |
|  | |

|  |  |
| --- | --- |
| **Sequence Diagram #UA-05** | **Taiga ID: #137** |
|  | |

|  |  |
| --- | --- |
| **Wireframe #UA-05** | **Taiga ID: #140** |
| **Update User Account Page**    **Update User Account Function** | |

|  |  |
| --- | --- |
| **Test Case #UA-05** | **Taiga ID: #142** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC  #UA-05  -1 | Start Update User Account | NIL | Update User Account Page Appear | Update User Account Page Appear | Pass | - | - | | TC  #UA-05  -2 | Enter valid Update Full name and password | Full Name =  User admin 1  and  Password = UA1\_password | User Update successfully,  Show updated User Account Data | User Update successfully,  Show updated User Account Data | pass | Fullname can be mixed with number | - | | TC  #UA-05  -3 | Blank Fields | Full Name = Blank  And  Password = Blank | Error message : “Fields cannot be empty” | Error message : “Fields cannot be empty” | pass | - | - | | |

|  |  |
| --- | --- |
| **User Story #UA-07** | **Taiga ID: #7** |
| As a User Administrator, I want to search for user accounts so that I can quickly find specific users. | |

|  |  |
| --- | --- |
| **Use Case Diagram #UA-07** | **Taiga ID: #144** |
| A diagram of a search user  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #UA-07** | **Taiga ID: #145** |
| **Name:** Search User Account | |
| **Stakeholders and Goals:** User Administrator – wants to search for specific user accounts using keywords in order to locate users for account review or management. | |
| **Description:** The User Administrator logs into the system and navigates to the User Administrator Dashboard. From there, they can perform a search using keywords to retrieve matching accounts for further action. | |
| **Actors:** User Administrator | |
| **Trigger:** User Administrator clicks on the “Search User Accounts” button. | |
| **Pre-Condition:**   1. The User Administrator is logged into the system with proper access rights. 2. The user account being searched for must exist in the database. | |
| **Normal Flow:**   1. The User Administrator clicks on the “Search User Accounts” button. 2. The User Administrator enters search criteria (e.g., username, full name, or user ID) and clicks “Search.” 3. The system retrieves and displays a list of matching user accounts. 4. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  **3a. No Matching Accounts Found:** If no user accounts match the entered criteria, the system displays a message stating “No results found.” | |

|  |  |
| --- | --- |
| **BCE Diagram #UA-07** | **Taiga ID: #146** |
|  | |

|  |  |
| --- | --- |
| **Sequence Diagram #UA-07** | **Taiga ID: #147** |
|  | |

|  |  |
| --- | --- |
| **Wireframe #UA-07** | **Taiga ID: #150** |
| **Search User Account Page** | |

|  |  |
| --- | --- |
| **Test Case #UA-07** | **Taiga ID: #152** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC  #UA-7  -1 | Start“Search User Account” | Nil | Search User Account page appears | Search User Account page appears | Pass | - | - | | TC  #UA-7  -2 | Enter a keyword and press the “Search” button. | 1 | User account with keyword 1 in data field is displayed | User account with keyword 1 in data field is displayed | Pass | - | - | | |

|  |  |
| --- | --- |
| **User Story #UA-08** | **Taiga ID: #8** |
| As a User Administrator, I want to create user profiles so that users have clearly defined roles. | |

|  |  |
| --- | --- |
| **Use Case Diagram #UA-08** | **Taiga ID: #154** |
| A diagram of a create process  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #UA-08** | **Taiga ID: #155** |
| **Name:** Create User Profile | |
| **Stakeholders and Goals:** User Administrator – wants to define and manage different user roles by creating new user profiles with appropriate names and descriptions. | |
| **Description:** The User Administrator logs into the system and navigates to the User Administrator Dashboard to create a new user profile. This allows the administrator to assign clear role definitions that can later be linked to user accounts. | |
| **Actors:** User Administrator | |
| **Trigger:** User Administrator clicks on the “Create New User Profile” button. | |
| **Pre-Condition:**   1. Must be logged in as User Administrator. 2. Must provide valid profile details. | |
| **Normal Flow:**   1. The User Administrator navigates to the User Management Dashboard and clicks on the “Create New User Profile” button. 2. The system displays the Create User Profile form. 3. The User Administrator fills in all required fields and clicks on the “Create Profile” button. 4. The system validates the input, saves the new profile to the database, and displays a confirmation message: “Profile created successfully!” 5. End. | |
|  | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  **2a. Input Validation or Account Creation Fails:** If the input is invalid or account creation fails due to business or system rules, the system displays error message stating: “Fields cannot be empty”, “Profile ID must be a valid number”, etc. | |

|  |  |
| --- | --- |
| **BCE Diagram #UA-08** | **Taiga ID: #156** |
|  | |

|  |  |
| --- | --- |
| **Sequence Diagram #UA-08** | **Taiga ID: #157** |
|  | |

|  |  |
| --- | --- |
| **Wireframe #UA-08** | **Taiga ID: #160** |
| **Create User Account** | |

|  |  |
| --- | --- |
| **Test Case #UA-08** | **Taiga ID: #162** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC  #UA-8  -1 | Start on User Management Page | UserID = 1 (User Admin) | "Create New User Profile" button is visible and clickable | "Create New User Profile" button is visible and clickable | Pass | - | - | | TC  #UA-8  -2 | Enter valid profile data | ProfileName = "Tester", ProfileID = 5 | Confirmation message: "Profile created successfully!" | Confirmation message: "Profile created successfully!" | Pass | - | - | | TC  #UA-8  -3 | Submit with empty fields | ProfileName = "", ProfileID = "" | Error message: "Fields cannot be empty" | Error message: "Fields cannot be empty" | Pass | - | - | | TC  #UA-8  -4 | Submit with invalid ID | ProfileID = "abc" | Error message: "Profile ID must be a valid number" | Error message: "Profile ID must be a valid number" | Pass | - | - | | |

|  |  |
| --- | --- |
| **User Story #UA-09** | **Taiga ID: #9** |
| As a User Administrator, I want to view all user profiles so that I can understand roles and permissions. | |

|  |  |
| --- | --- |
| **Use Case Diagram #UA-09** | **Taiga ID: #164** |
| A diagram of a view user  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #UA-09** | **Taiga ID: #165** |
| **Name:** View All User Profiles | |
| **Stakeholders and Goals:** User Administrator – wants to view all user profiles to understand the roles and permissions assigned to different users and manage access levels appropriately. | |
| **Description:** The User Administrator logs into the system and navigates to the User Administrator Dashboard to view a list of all user profiles. This allows the User Administrator to monitor access rights and maintain security governance across the platform. | |
| **Actors:** User Administrator | |
| **Trigger:** User Administrator clicks on the “View All User Profiles” button. | |
| **Pre-Condition:**   1. The User Administrator must be logged into the system. 2. The account must have administrative privileges. | |
| **Normal Flow:**   1. The User Administrator clicks on the "View All User Profiles" button. 2. The system retrieves and displays a list of all user profiles. 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  **2a. No User Profiles Available:** If no user profiles are present in the database (e.g., if no profiles have been created yet), the system will display a message stating “No user profiles available to display.”  **2b. Insufficient Privileges:** If the User Administrator lacks the appropriate privileges to view user profiles, the system will display a message stating “You do not have the required privileges.” | |

|  |  |
| --- | --- |
| **BCE Diagram #UA-09** | **Taiga ID: #166** |
| Picture 1, Picture | |

|  |  |
| --- | --- |
| **Sequence Diagram #UA-09** | **Taiga ID: #167** |
| Picture 1, Picture | |

|  |  |
| --- | --- |
| **Wireframe #UA-09** | **Taiga ID: #170** |
| **View All User Profiles**  Picture 1, Picture | |

|  |  |
| --- | --- |
| **Test Case #UA-09** | **Taiga ID: #172** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC #UA-09-1 | Click “View All User Profiles” on dashboard | profileID = 1,  profileName= User Administrator | User profile  table is displayed with all user profile records. | User profile  table is displayed with all user profile records. | Pass | - | - | | TC #UA-09-2 | Table loads empty when no profiles exist | NIL (no accounts in DB) | Display: “No user profiles available to display ” error message is shown. | Display: “No user profiles available to display ” error message is shown. | Pass | Optional test if table is empty | - | | TC #UA-09-3 | Attempt to access with unauthorized role | Username = cleaner1  Password = CL1\_password | Error message: “You do not have the required privileges.” is shown. | Error message: “You do not have the required privileges.” is shown. | Pass | Cleaners should not access this feature | - | | |

|  |  |
| --- | --- |
| **User Story #UA-10** | **Taiga ID: #10** |
| As a User Administrator, I want to update user profiles so that I can reflect changes in user roles. | |

|  |  |
| --- | --- |
| **Use Case Diagram #UA-10** | **Taiga ID: #174** |
| A diagram of a software update  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #UA-10** | **Taiga ID: #175** |
| **Name:** Update User Profile | |
| **Stakeholders and Goals:** User Administrator – wants to ensure that user profiles are accurate and reflect the correct user roles to maintain effective system access control. | |
| **Description:** The User Administrator logs into the system and navigates to the User Profile Management section to update user profiles. This ensures that all user roles remain relevant and properly defined in the system. | |
| **Actors:** User Administrator | |
| **Trigger:** User Administrator clicks on the “Update User Profile” button. | |
| **Pre-Condition:**   1. The User Administrator must be logged into the system with administrative privileges. 2. The target user profile must exist in the system. | |
| **Normal Flow:**   1. The User Administrator selects a user profile, edits the profile information and clicks the "Update Profile" button. 2. The system stores the updated profile and displays the updated information for confirmation. 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  **2a. Blank Fields:** If any required fields (e.g., profile name or description) are left blank, the system will displaymessage stating “Fields cannot be empty.”  **2b. No Profile Selected:** If the User Administrator does not select a profile before attempting to update, the system will display message stating“Please select a profile.” | |

|  |  |
| --- | --- |
| **BCE Diagram #UA-10** | **Taiga ID: #176** |
|  | |

|  |  |
| --- | --- |
| **Sequence Diagram #UA-10** | **Taiga ID: #177** |
|  | |

|  |  |
| --- | --- |
| **Wireframe #UA-10** | **Taiga ID: #180** |
| **Update User Profile Page** | |

|  |  |
| --- | --- |
| **Test Case #UA-10** | **Taiga ID: #182** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC  #UA-10  -1 | Start Update User Profile | NIL | Update User Profile Page Appear | Update User Profile Page Appear | pass | - | - | | TC  #UA-10  -2 | Enter valid User profile, Profile Name and Description | User profile  = User administrator and  Profile Name  = User Administrator and  Description = User admin’s description | Profile Update successfully,  Show updated User Profile Data | Profile Update successfully,  Show updated User Profile Data | pass | - | - | | TC  #UA-10  -3 | Blank Fields | User profile  = Blank  or  Profile Name  = Blank  or  Description = Blank | Error message : “Fields cannot be empty” | Error message : “Fields cannot be empty” | pass | - | - | | |

|  |  |
| --- | --- |
| **User Story #UA-12** | **Taiga ID: #12** |
| As a User Administrator, I want to search for user profiles so that I can manage users easily. | |

|  |  |
| --- | --- |
| **Use Case Diagram #UA-12** | **Taiga ID: #184** |
| A diagram of a search engine  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #UA-12** | **Taiga ID: #185** |
| **Name:** Search for User Profiles | |
| **Stakeholders and Goals:** User Administrator – wants to locate and manage user profiles efficiently for the purpose of updating, reviewing, or validating user role information. | |
| **Description:** The User Administrator logs into the system and navigates to the User Management Panel to search for user profiles. The administrator can enter keywords such as profile name or description to locate relevant profiles for further action. | |
| **Actors:** User Administrator | |
| **Trigger:** The User Administrator clicks on the “Search” button. | |
| **Pre-Condition:**   1. The User Administrator has access to the User Management Panel. 2. User profile records are stored in the system. | |
| **Normal Flow:**   1. The user administrator enters a search keyword and presses “Search”. 2. The system processes the input and displays matching user profile details. 3. Ends | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  **3a. No Matching Profiles Found:** If no user profiles match the entered keyword, the system displays an error message stating “No matching user profiles found.” | |

|  |  |
| --- | --- |
| **BCE Diagram #UA-12** | **Taiga ID: #186** |
|  | |

|  |  |
| --- | --- |
| **Sequence Diagram #UA-12** | **Taiga ID: #187** |
|  | |

|  |  |
| --- | --- |
| **Wireframe #UA-12** | **Taiga ID: #190** |
| **Search User Account Page** | |

|  |  |
| --- | --- |
| **Test Case #UA-12** | **Taiga ID: #442** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC  #UA-12  -1 | Press “Search User Profile” button | Nil | Display the list of user profiles | Display the list of user profiles | Pass | - | - | | TC  #UA-12  -2 | Enter “keyword” and press Search button | User Administrator | User profile details should appear | User profile details should appear | Pass | - | - | | |

**[Sprint 2] Stakeholder #2 – Home Owner**

|  |  |
| --- | --- |
| **User Story #HO-02** | **Taiga ID: #14** |
| As a Home Owner, I want to log out of the system so that I can exit securely. | |

|  |  |
| --- | --- |
| **Use Case Diagram #HO-02** | **Taiga ID: #194** |
| A diagram of a logo  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #HO-02** | **Taiga ID: #195** |
| **Name:** Log Out of the Home Owner Account | |
| **Stakeholders and Goals:** Home Owner **–** wants to securely exit the system after accessing services. | |
| **Description:** The Home Owner intends to log out of the system to ensure that no unauthorised access can occur after their session ends. | |
| **Actors:** Home Owner | |
| **Trigger:** Home Owner clicks on the “Logout” button. | |
| **Pre-Condition:** The user must be logged into the system. | |
| **Normal Flow:**   1. Home Owner clicks the “Logout” button. 2. The system terminates the current session and securely logs the user out. 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

|  |  |
| --- | --- |
| **BCE Diagram #HO-02** | **Taiga ID: #196** |
| A white box with black text  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Sequence Diagram #HO-02** | **Taiga ID: #197** |
| A diagram of a login page  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Wireframe #HO-02** | **Taiga ID: #200** |
| **Logout Button on Home Owner Dashboard**    **After Logout – Redirect to Cleaning Services Platform Login Page** | |

|  |  |
| --- | --- |
| **Test Case #HO-02** | **Taiga ID: #202** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC  #HO-02  -1 | Click on the “Logout” button or icon | NIL | System logs out the home owner | System logs out the home owner | Pass | - | - | | |

|  |  |
| --- | --- |
| **User Story #HO-03** | **Taiga ID: #15** |
| As a Home Owner, I want to view available cleaning services so that I can browse and evaluate service options. | |

|  |  |
| --- | --- |
| **Use Case Diagram #HO-03** | **Taiga ID: #204** |
| A close-up of a diagram  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #HO-03** | **Taiga ID: #205** |
| **Name:** View Cleaning Services | |
| **Stakeholders and Goals:** Home Owner – wants to browse the list of cleaning services available on the platform to compare and evaluate suitable options for booking. | |
| **Description:** The Home Owner logs into the system and navigates to the Browse Services page to view all cleaning services offered by registered Cleaners. This allows the Home Owner to assess the services, including titles, descriptions, prices, and categories. | |
| **Actors:** Home Owner | |
| **Trigger:** Home Owner clicks on the “View Available Cleaning Services” button | |
| **Pre-Condition:** Must be logged in as Home Owner | |
| **Normal Flow:**   1. The Home Owner clicks on the “View Available Cleaning Services” button. 2. The system retrieves and displays a list of available cleaning services. 3. The Home Owner views the service details. 4. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows: none** | |

|  |  |
| --- | --- |
| **BCE Diagram #HO-03** | **Taiga ID: #206** |
|  | |

|  |  |
| --- | --- |
| **Sequence Diagram #HO-03** | **Taiga ID: #207** |
|  | |

|  |  |
| --- | --- |
| **Wireframe #HO-03** | **Taiga ID: #210** |
| **View Available Cleaning Service Page** | |

|  |  |
| --- | --- |
| **Test Case #HO-03** | **Taiga ID: #212** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC  #HO-3  -1 | Start on Home Owner Dashboard | HomeOwnerID = 41 | "Browse Services" button is visible and clickable | "Browse Services" button is visible and clickable | Pass | - | - | | TC  #HO-3  -2 | Click "Browse Services" | HomeOwnerID = 41 | List of all available cleaning services displayed | List of all available cleaning services displayed | Pass | - | - | | TC  #HO-3  -3 | View service details | Click on any service title (e.g., "service1") | Service details are shown (title, description, price, category) | Service details are shown (title, description, price, category) | Pass | - | - | | |

|  |  |
| --- | --- |
| **User Story #HO-05** | **Taiga ID: #17** |
| As a Home Owner, I want to save cleaning services to a shortlist so that I can compare and select later. | |

|  |  |
| --- | --- |
| **Use Case Diagram #HO-05** | **Taiga ID: #214** |
|  | |

|  |  |
| --- | --- |
| **Use Case Description #HO-05** | **Taiga ID: #215** |
| **Name:** Add cleaning service to shortlist | |
| **Stakeholders and Goals:** Home Owner – To browse the list of available cleaning services provided by various cleaners so that they can evaluate and select a suitable service. | |
| **Description:** This use case allows the Home Owner to view a list of all available cleaning services, along with key information such as cleaner name, service type, service date, time slot, and price. | |
| **Actors:** Home Owner | |
| **Trigger:** The Home Owner selects the “View Available Cleaning Services” option from Home Owner their dashboard. | |
| **Pre-Condition:**   1. The Home Owner must be logged in. 2. The Home Owner has successfully navigated to the Home Owner Dashboard. | |
| **Normal Flow:**   1. The Home Owner clicks on “View Available Cleaning Services.” 2. The system retrieves all available services from the database and displays them in a table format. 3. Each row contains: Service Title, Cleaner Name, Service Date, Time Slot, and Price. 4. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  **2a. Service Already Shortlisted:** If the service is already in the shortlist, the system displays an error message like “Already shortlisted or failed.” | |

|  |  |
| --- | --- |
| **BCE Diagram #HO-05** | **Taiga ID: #216** |
| A screenshot of a computer  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Sequence Diagram #HO-05** | **Taiga ID: #217** |
|  | |

|  |  |
| --- | --- |
| **Wireframe #HO-05** | **Taiga ID: #220** |
| **Available Cleaning Services Page – “ Add to Shortlist” button**    **Success message - “Service added to shortlist.”** | |

|  |  |
| --- | --- |
| **Test Case #HO-05** | **Taiga ID: #222** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC  #HO-5  -1 | Start from Dashboard | HomeOwnerID = 41 | "View Available Cleaning Services" option is visible and clickable | "View Available Cleaning Services" option is visible and clickable | Pass | - | - | | TC  #HO-5  -2 | Click view available services | HomeOwnerID = 41 | List of services shown in table format with service title, cleaner name, date, time, and price | List of services shown in table format with service title, cleaner name, date, time, and price | Pass | - | - | | TC  #HO-5  -3 | Add already shortlisted service | HomeOwnerID = 41, ServiceID = 63 | Error message: “Already shortlisted or failed.” | Error message: “Already shortlisted or failed.” | Pass | - | - | | TC  #HO-5  -4 | Simulate empty service list | Temporary DB state = empty | System shows: “No available cleaning services.” | System shows: “No available cleaning services.” | Pass | - | - | | |

|  |  |
| --- | --- |
| **User Story #HO-06** | **Taiga ID: #18** |
| As a Home Owner, I want to view my favourite list so that I can review the cleaning services I have shortlisted. | |

|  |  |
| --- | --- |
| **Use Case Diagram #HO-06** | **Taiga ID: #224** |
|  | |

|  |  |
| --- | --- |
| **Use Case Description #HO-06** | **Taiga ID: #225** |
| **Name:** View Services Favourite List | |
| **Stakeholders and Goals:** Home Owner – wants to view the list of cleaning services they have shortlisted in order to compare and make informed booking decisions. | |
| **Description:** The Home Owner logs into the system and navigates to Home Owner Dashboard to view cleaning services they have previously shortlisted. This feature allows the Home Owner to conveniently review and compare services before confirming a booking. | |
| **Actors:** Home Owner | |
| **Trigger:** Home Owner clicks on the “View Favourite List” button. | |
| **Pre-Condition:** The user must be logged into the system   1. Must be logged in as a Home Owner. 2. The Home Owner has previously shortlisted one or more cleaning services. | |
| **Normal Flow:**   1. The Home Owner clicks on the "View Favourite List" button. 2. The system retrieves and displays a list of all cleaning services the Home Owner has added to their shortlist. 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows: none** | |

|  |  |
| --- | --- |
| **BCE Diagram #HO-06** | **Taiga ID: #226** |
| A close-up of a screen  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Sequence Diagram #HO-06** | **Taiga ID: #227** |
|  | |

|  |  |
| --- | --- |
| **Wireframe #HO-06** | **Taiga ID: #230** |
| **View favourite list** | |

|  |  |
| --- | --- |
| **Test Case #HO-06** | **Taiga ID: #232** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC  #HO-6  -1 | Start Favourite List Page | NIL | Favourite list page loads | Favourite list page loads | Pass | - | - | | TC  #HO-6  -2 | Valid shortlist exists | HomeOwnerID = 41 | List of shortlisted services shown (e.g., service63, service79) | List of shortlisted services shown (e.g., service63, service79) | Pass | - | - | | TC  #HO-6  -3 | No shortlist exists | HomeOwnerID = 42 | Message: “You have not shortlisted any services.” | Message: “You have not shortlisted any services.” | Pass | - | - | | TC  #HO-6  -4 | Simulate DB fail | Any user ID | Message: “Failed to retrieve shortlist.” | Message: “Failed to retrieve shortlist.” | Pass | - | - | | |

|  |  |
| --- | --- |
| **User Story #HO-08** | **Taiga ID: #20** |
| As a Home Owner, I want to view my past cleaning services filtered by service and date, so that I can review all past bookings. | |

|  |  |
| --- | --- |
| **Use Case Diagram #HO-08** | **Taiga ID: #234** |
|  | |

|  |  |
| --- | --- |
| **Use Case Description #HO-08** | **Taiga ID: #235** |
| **Name:** View Past Cleaning bookings | |
| **Stakeholders and Goals:** Home Owner – wants to review their previous cleaning bookings, filtered by service and date period, to track and reflect on past activity. | |
| **Description:** The Home Owner logs into the system and navigates to the Booking History page to view all completed cleaning services. Filters can be applied by service and date range to narrow the results and focus on specific bookings. | |
| **Actors:** Home Owner | |
| **Trigger:** Home Owner clicks on the “View Full Service History” button | |
| **Pre-Condition:**   1. Must be logged in as Home Owner 2. Past cleaning service records must exist in the system | |
| **Normal Flow:**   1. The Home Owner navigates to the Home Owner Dashboard and clicks on the “View Full Service History” button. 2. The Home Owner applies filters by selecting a service and date range. 3. The system retrieves and displays a list of matching past cleaning service bookings. 4. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  **3a. No History :** If the system fails to retrieve any history record the system will display the message stating “You have no service history yet”. | |

|  |  |
| --- | --- |
| **BCE Diagram #HO-08** | **Taiga ID: #236** |
|  | |

|  |  |
| --- | --- |
| **Sequence Diagram #HO-08** | **Taiga ID: #237** |
|  | |

|  |  |
| --- | --- |
| **Wireframe #HO-08** | **Taiga ID: #240** |
|  | |

|  |  |
| --- | --- |
| **Test Case #HO-08** | **Taiga ID: #242** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC  #HO-8  -1 | Start View Booking History Page | NIL | Booking history page appears | Booking history page appears | Pass | - | - | | TC  #HO-8  -2 | Valid filter - service only | Service = category1  Date range = blank | List of completed bookings for category1 shown | List of completed bookings for category1 shown | Pass | - | - | | TC  #HO-8  -3 | Valid filter - date range only | Service = All Services  Date: 2025-01-01 to 2025-03-01 | List of bookings during range is shown | List of bookings during range is shown | Pass | - | - | | TC  #HO-8  -4 | Valid filter - both fields | Service = category1  Date: 2025-01-01 to 2025-03-01 | Relevant filtered bookings displaye | Relevant filtered bookings displaye | Pass | - | - | | TC  #HO-8  -5 | No history exists | Service = category1  Date: 2030-01-01 to 2030-02-01 | Message: “You have no service history yet.” | Message: “You have no service history yet.” | Pass | - | - | | TC  #HO-8  -6 | Database fail (simulate) | Any valid input | Message: “Failed to retrieve data” | Message: “Failed to retrieve data” | Pass | - | - | | |

**[Sprint 2] Stakeholder #3 – Cleaner**

|  |  |
| --- | --- |
| **User Story #** | **Taiga ID: #23** |
| As a Cleaner, I want to log out of the system so that I can exit securely. | |

|  |  |
| --- | --- |
| **Use Case Diagram #CL-02** | **Taiga ID: #244** |
| A diagram of a logo  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #CL-02** | **Taiga ID: #245** |
| **Name:** Log Out of the Cleaner Account | |
| **Stakeholders and Goals:** Cleaner **–** wants to securely exit the system after managing services. | |
| **Description:** The Cleaner intends to log out of the system to ensure that no unauthorised access can occur after their session ends. | |
| **Actors:** Cleaner | |
| **Trigger:** Cleaner clicks on the “Logout” button. | |
| **Pre-Condition:** The user must be logged into the system. | |
| **Normal Flow:**   1. Cleaner clicks the “Logout” button. 2. The system terminates the current session and securely logs the user out. 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

|  |  |
| --- | --- |
| **BCE Diagram #CL-02** | **Taiga ID: #246** |
| A screen shot of a website  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Sequence Diagram #CL-02** | **Taiga ID: #247** |
| A diagram of a computer program  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Wireframe #CL-02** | **Taiga ID: #250** |
| **Logout Button on Cleaner Dashboard**    **After Logout – Redirect to Cleaning Services Platform Login Page** | |

|  |  |
| --- | --- |
| **Test Case #CL-02** | **Taiga ID: #252** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC  #CL-02  -1 | Click on the “Logout” button or icon | NIL | System logs out the cleaner | System logs out the cleaner | Pass | - | - | | |

|  |  |
| --- | --- |
| **User Story #CL-04** | **Taiga ID: #25** |
| As a Cleaner, I want to view my cleaning services so that I can monitor what I offer. | |

|  |  |
| --- | --- |
| **Use Case Diagram #CL-04** | **Taiga ID: #254** |
| A close-up of a diagram  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #CL-04** | **Taiga ID: #255** |
| **Name:** View My Cleaning Services | |
| **Stakeholders and Goals:** Cleaner – wants to view the list of their cleaning services to monitor and manage offerings, ensure accuracy of details, and track availability for bookings. | |
| **Description:** The Cleaner logs into the system and navigates to the Cleaner Dashboard to view a list of their cleaning services. This allows the Cleaner to keep service details updated and ensure that their offerings are visible to potential customers. | |
| **Actors:** Cleaner | |
| **Trigger:** Cleaner clicks on the “View My Cleaning Services” button. | |
| **Pre-Condition:**   1. The Cleaner must be logged into the system. 2. The account must have appropriate role-based access to view personal service listings. | |
| **Normal Flow:**   1. The Cleaner clicks on the "View My Cleaning Services" button. 2. The system retrieves and displays a list of all cleaning services offered by the Cleaner. 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  **2a. Retrieval Error:** If the system is unable to retrieve the data (e.g., database error), the system will display message stating “Unable to retrieve services.”  **2b. No Services Available:** If the Cleaner does not have any services listed in the system, the system will display message stating “You have no cleaning services listed.” | |

|  |  |
| --- | --- |
| **BCE Diagram #CL-04** | **Taiga ID: #256** |
| Picture 1, Picture | |

|  |  |
| --- | --- |
| **Sequence Diagram #CL-04** | **Taiga ID: #257** |
| A diagram of a cleaning service  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Wireframe #CL-04** | **Taiga ID: #260** |
| **View My Cleaning Services**  Picture 1, Picture | |

|  |  |
| --- | --- |
| **Test Case #CL-04** | **Taiga ID: #262** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC #CL-04-1 | Click “View My Services” on dashboard | Cleaner ID = 7 | Services table displays cleaner1’s listed services. | Services table displays cleaner1’s listed services. | Pass | - | - | | TC #CL-04-2 | No services created by the cleaner | cleaner10 (assumed has no services) | Message: “You have no cleaning services listed.” appears. | Message: “You have no cleaning services listed.” appears. | Pass | - | - | | TC #CL-04-3 | System failure (simulate DB disconnect or crash) | Trigger DB error manually or simulate backend issue | Message: “Unable to retrieve services.” appears. | Message: “Unable to retrieve services.” appears. | Pass | - | - | | |

|  |  |
| --- | --- |
| **User Story #CL-05** | **Taiga ID: #26** |
| As a Cleaner, I want to update my cleaning services so that I can ensure details are up to date. | |

|  |  |
| --- | --- |
| **Use Case Diagram #CL-05** | **Taiga ID: #264** |
| A close-up of a sign  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #CL-05** | **Taiga ID: #265** |
| **Name:** Update Cleaning Services | |
| **Stakeholders and Goals:** Cleaner – wants to keep their cleaning service details accurate and up to date so that Home Owners and User Administrators can view correct service information. | |
| **Description:** The Cleaner logs into the system and navigates to the cleaning services section to update the details of their services. This ensures that their listings reflect the most current and accurate service offerings on the platform. | |
| **Actors:** Cleaner | |
| **Trigger:** Cleaner clicks on the “Update Cleaning Services” button | |
| **Pre-Condition:**   1. The Cleaner must be logged into the system. 2. The cleaning service being updated must belong to the Cleaner. | |
| **Normal Flow:**   1. The Cleaner navigates to their list of cleaning services, selects a service to edit, updates the details, and clicks the "Update" button. 2. The system validates the inputs and, if all fields are filled correctly, stores the updates and displays a success message: "Services are updated!" 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  **2a. Blank Fields:** If any required fields are left empty, the system will display error message stating “All fields must be filled.”  **2b. Error Data Type:** If the price field is not a valid float data type, the system will display the error message stating “Update Failed.” | |

|  |  |
| --- | --- |
| **BCE Diagram #CL-05** | **Taiga ID: #266** |
|  | |

|  |  |
| --- | --- |
| **Sequence Diagram #CL-05** | **Taiga ID: #267** |
|  | |

|  |  |
| --- | --- |
| **Wireframe #CL-05** | **Taiga ID: #270** |
| **Update cleaning service page**    **Update cleaning service function**      **\*No changing service category in update cleaning service page** | |

|  |  |
| --- | --- |
| **Test Case #CL-05** | **Taiga ID: #272** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC  #CL-05  -1 | Start Update Cleaning Service | NIL | Update Cleaning Service Page Appear | Update Cleaning Service Page Appear | pass | - | - | | TC  #CL-05  -2 | Enter valid  Title,  Description,  Price | Title = service1  And  Description = service1’s description  And  price = integer | Cleaning Service successfully,  Show updated User Profile Data | Cleaning Service successfully,  Show updated User Profile Data | pass | - | - | | TC  #CL-05  -3 | Any Blank Fields | Title = Blank  or  Description = Blank  or  Price = Blank / not integer | Error message: “Fields cannot be empty” | Error message: “Fields cannot be empty” | pass | - | - | | TC  #CL-05  -4 | Price <= 0 | Price  = -1 or 0 | Error message:  “Your price must be higher than 0” | Error message:  “Your price must be higher than 0” | pass | - | - | | TC  #CL-05  -5 | Cleaner update cleaning service but cannot change service category | No option to change service category | “Service Category updated” | No option to change service category | fail | - | \*No changing service category in update cleaning service page | | |

## [Sprint 2] Stakeholder #4 – Platform Manager

|  |  |
| --- | --- |
| **User Story #PM-02** | **Taiga ID: #34** |
| As a Platform Manager, I want to log out of the system so that I can exit securely. | |

|  |  |
| --- | --- |
| **Use Case Diagram #PM-02** | **Taiga ID: #274** |
| A diagram of a logistic  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #PM-02** | **Taiga ID: #275** |
| **Name:** Log Out of the Platform Manager Account | |
| **Stakeholders and Goals:** Platform Manager **–** wants to securely exit the system after managing platform features. | |
| **Description:** The Platform Manager intends to log out of the system to ensure that no unauthorised access can occur after their session ends. | |
| **Actors:** Platform Manager | |
| **Trigger:** Platform Manager clicks on the “Logout” button | |
| **Pre-Condition:** The user must be logged into the system | |
| **Normal Flow:**   1. Platform Manager clicks the “Logout” button. 2. The system terminates the current session and securely logs the user out. 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

|  |  |
| --- | --- |
| **BCE Diagram #PM-02** | **Taiga ID: #276** |
| A screenshot of a browser  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Sequence Diagram #PM-02** | **Taiga ID: #277** |
| A diagram of a diagram  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Wireframe #PM-02** | **Taiga ID: #280** |
| **Logout Button on Platform Manager Dashboard**    **After Logout – Redirect to Cleaning Services Platform Login Page** | |

|  |  |
| --- | --- |
| **Test Case #PM-02** | **Taiga ID: #282** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC  #PM-02  -1 | Click on the “Logout” button or icon | NIL | System logs out the platform manager | System logs out the platform manager | Pass | - | - | | |

|  |  |
| --- | --- |
| **User Story #PM-05** | **Taiga ID: #37** |
| As a Platform Manager, I want to update cleaning services categories so that I can keep category information accurate and up to date. | |

|  |  |
| --- | --- |
| **Use Case Diagram #PM-05** | **Taiga ID: #284** |
|  | |

|  |  |
| --- | --- |
| **Use Case Description #PM-05** | **Taiga ID: #285** |
| **Name:** Update Cleaning Service Categories | |
| **Stakeholders and Goals:** Platform Manager – wants to update cleaning service categories to ensure that category names and descriptions are accurate, well-structured, and reflect the current range of services offered on the platform. | |
| **Description:** The Platform Manager logs into the system and navigates to the Platform Manager Dashboard to update cleaning service category information. This helps ensure the platform’s services are correctly categorized and easily understood by users. | |
| **Actors:** Platform Manager | |
| **Trigger:** Platform Manager clicks on the “Update Service Category” button. | |
| **Pre-Condition:**   1. The Platform Manager is logged into the system. 2. The account has the appropriate privileges to access and manage service categories. | |
| **Normal Flow:**   1. Platform Manager clicks on the "Update Cleaning Service Category" button. 2. The Platform Manager selects the desired category and updates the necessary information (e.g., category name, description). 3. The system saves the updated information and displays the updated category for confirmation. 4. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  **2a. Blank Fields:** If the Platform Manager leaves any required fields empty, the system will display an error message stating “All fields must be filled.” | |

|  |  |
| --- | --- |
| **BCE Diagram #PM-05** | **Taiga ID: #286** |
|  | |

|  |  |
| --- | --- |
| **Sequence Diagram #PM-05** | **Taiga ID: #287** |
|  | |

|  |  |
| --- | --- |
| **Wireframe #PM-05** | **Taiga ID: #290** |
| **Update Cleaning Service Category Page**    **Update Cleaning Service Category Function** | |

|  |  |
| --- | --- |
| **Test Case #PM-05** | **Taiga ID: #292** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC  #PM-05  -1 | Start Update Service category | NIL | Update Service category Page Appear | Update Service category Page Appear | pass | - | - | | TC  #PM-05  -2 | Enter valid  Name,  Description, | Name = category1  And  Description = service1’s description | Service category Update successfully,  Show updated Service category Data | Service category Update successfully,  Show updated Service category Data | pass | - | - | | TC  #PM-05  -3 | Any Blank Fields | Name = Blank  or  Description = Blank | Error message: “Fields cannot be empty” | Error message: “Fields cannot be empty” | pass | - | - | | |

**UML Diagram for Sprint 2**

|  |  |
| --- | --- |
| **UML Diagram Sprint 2** | **Taiga ID: #118, #128, #138, #148, #158, #168, #178, #188, #198, #208, #218, #228, #238, #248, #258, #268, #278, #288** |
| **A diagram of a uml class  AI-generated content may be incorrect.** | |

**Data Persistence Diagram for Sprint 2**

|  |  |
| --- | --- |
| **Sprint 2** | **Taiga ID: #119, #129, #139, #149, #159, #169, #179, #189, #199, #209, #219, #229, #239, #249, #259, #269, #279, #289** |
| **A diagram of a computer language  AI-generated content may be incorrect.**  **A black and white screen with white text  AI-generated content may be incorrect.**  **A screenshot of a computer  AI-generated content may be incorrect.**  **A black square with white text  AI-generated content may be incorrect.** | |

**Sprint 3**

**Meeting Minutes for Sprint 3**

|  |  |
| --- | --- |
| **Meeting Minutes for Project Meeting 4** | **Taiga ID: #523** |
| **WipeOut**  **Project Meeting 4**  **Date / Time** **:** 30 April 2025 / 12.10pm  **Location :** e-Meeting on Discord    **Present :** Chea Darayuth Team Member  Ho Ka Yan Jeslyn Team Member  Hein Htet Zaw Team Member  Koung Khant Saung Team Member  Law Jun Wei Team Member  **Absent :** Kaung Minn Khant Team Member  **In Attendance :** -     |  |  |  | | --- | --- | --- | | **S/N** | **ITEM** | **ACTION** | | **A** | **Procedural (Opening)** |  | | **A1** | **Apologies for Absence** |  | | 1 | Kaung was absent due to an urgent personal matter. |  | | **B1** | **Matters Arising** |  | | **1** | **Stakeholder Dashboard Code**  Htet pointed out that the dashboard interfaces for stakeholders (e.g., Home Owner, Cleaner, Platform Manager, User Admin) were not included within individual user stories.  Jun Wei responded that he would take responsibility for coding all stakeholder dashboards to maintain consistency and integration. | JunWei to note | | **2** | **User Story Sprint Reallocation**  Since Htet was not present at the previous sprint planning session, Jun Wei asked him to confirm how he would like to reallocate his user stories.  Htet confirmed the following allocation:   * **Sprint 1:** UA-03, CL-03, PM-03 * **Sprint 2:** UA-08, HO-03, HO-06, HO-07, HO-09 | All to note | | **3** | **ID Generation in Create Functions**  Htet inquired whether IDs used in create functions should be manually assigned or system-generated.  Jun Wei suggested using auto-generated IDs, since users like Home Owners and Cleaners would not know the appropriate IDs to input.  Htet shared two implementation options:   * Auto-generate using MySQL Workbench * Auto-generate using Java code   Jun Wei recommended the Java-based implementation, so the logic will be visible during code demonstration to Mr. Terence. | Htet to note  Htet to note | | **4**  4.1  4.2 | **Any Other Business**  JavaFX Approval Check  **JunWei** – requested Htet to confirm with Mr. Terence if his code is approved before the team proceeds with switching to JavaFX for the front-end implementation.  Database Functionality Check  **JunWei** – asked if the team had tested the database structure.  Htet confirmed that he had tested several components and everything appears to be working correctly. | Htet to note | | **C**  **1**  **2** | **Actions and Responsibilities**  JunWei to work on the stakeholder dashboards.  Htet to get approval from Mr. Terence on his JavaFX code. | JunWei to note  Htet to note | | **D**  **1** | **Date of Next Meeting**  The next project meeting is tentatively scheduled for 7 May 2025, 12.00pm, after tutorial.  Any changes to the meeting schedule will be informed via the Group Telegram Chat.  Meeting was adjourned at 12.55pm |  | |  | Recorded By: Law Jun Wei  Vetted By: Jeslyn Ho Ka Yan |  | | |

|  |  |
| --- | --- |
| **Meeting Minutes for Project Meeting 5** | **Taiga ID: #524** |
| **WipeOut**  **Project Meeting 5**  **Date / Time** **:** 07 May 2025 / 11.00am  **Location :** SIM HQ, Block A, Level 5, LAB A.5.17A/A.5.17B    **Present :** Chea Darayuth Team Member  Ho Ka Yan Jeslyn Team Member  Hein Htet Zaw Team Member  Kaung Minn Khant Team Member  Koung Khant Saung Team Member  Law Jun Wei Team Member  **Absent**  **:** -  **In Attendance :** -     |  |  |  | | --- | --- | --- | | **S/N** | **ITEM** | **ACTION** | | **A** | **Procedural (Opening)** |  | | **A1** | **Apologies for Absence** |  | | 1 | There were no absentees. |  | | **B1** | **Matters Arising** |  | | **1** | **Clarification from Mr. Terence**  JunWei confirmed with MR. Terence that no diagrams need to be uploaded to Taiga. Taiga is used strictly for tracking purposes only. | All to note | | **2**  2.1  2.2  2.3  2.4  2.5  2.6  2.7  2.8  2.9 | **Project Task Planning and Allocation**  Test Case (User Perspective)  Testing to be conducted from the user’s point of view, focusing on functional flow and validation.  Test Report (Assigned to: JunWei)  To summarize the results and expectations of all test cases, highlighting successful and failed scenarios.  Test Data (Assigned to: JunWei)  Dummy data to be prepared for use in testing all key functionalities and user scenarios.  Test-Driven Development (Assigned to: Hein)  Use a test calculator to write and validate test cases. Include result statements such as:   * This test is a positive testing and the result is correct. * This test is a positive testing and the result is incorrect. * This test is a negative testing and the result is correct. * This test is a negative testing and the result is incorrect.   CI/CD Pipeline (Assigned to: Hein)  Screenshot of GitHub CI/CD (e.g., GitHub Actions) to be included in the final report.  Data-Driven Software Development (Assigned to: Koung)  Draft a 1.5-page proposal outlining how machine learning could improve the system using existing or potential data.  Proposal to be presented during Tutorial 6/7.  Ethical Issues (Assigned to: Koung)  Prepare 0.5-page write-up on ethical considerations:   * 2 product-related issues * 2 process-related issues   Proposal to be presented during Tutorial 6/7.  Unit Testing (Assigned to: All)  Each team member to carry out unit testing on their own modules before integration.  Demo Video (Assigned to: Jes)  Each team member to carry out unit testing on their own modules before integration.  Video should:   * Be in .mp4 format * Showcase all main actor functions * Include audio narration * Use separate accounts for different roles | All to note  JunWei to note  JunWei to note  Hein to note  Hein to note  Koung to note  Koung to note  All to note  Jes to note | | **3** | **Final Submission Checklist**   1. **Final Report**  * Sprint-based UML Class Diagram and Data Persistence Diagrams  1. **Source Code** in .zip format (no.rar) 2. **Taiga ID** 3. **Demo Video** in .mp4 format  * Must include walkthrough of all main actor functions. * Must feature voiceover. * Must be done with different accounts per member.  1. **Gannt Chart** (Assigned to: JunWei) 2. **Other supporting annexes** in PDF format | All to note  JunWei to note | | **4** | **Any Other Business**  There was no AOB. |  | | **C**  **1**  **2**  **3**  **4**  **5**  **6**  **7**  **8** | **Actions and Responsibilities**  JunWei to compile and summarise test report.  JunWei to generate and prepare test data.  Hein to set up CI/CD and prepare report screenshots.  Hein to design and document TDD scenarios.  Koung to draft DDD proposal (1.5 pages) by tutorial 6/7.  Koung to write ethical issues section (0.5 pages) by tutorial 6/7.  Jes to prepare and manage demo video recording.  All members to perform unit testing on individual modules. | JunWei to note  JunWei to note  Hein to note  Hein to note  Koung to note  Koung to note  Jes to note  All to note | | **D**  **1** | **Date of Next Meeting**  The next project meeting is tentatively scheduled for 14 May 2025, 3.00pm, after tutorial.  Any changes to the meeting schedule will be informed via the Group Telegram Chat.  Meeting was adjourned at 11.38am. |  | |  | Recorded By: Law Jun Wei  Vetted By: Jeslyn Ho Ka Yan |  | | |

**Taiga Overview for Sprint 3**

|  |
| --- |
| **Sprint 3** |
| **A screenshot of a phone  AI-generated content may be incorrect.** |

**Gantt Chart Overview for Sprint 3**

|  |  |
| --- | --- |
| **Sprint 3** | **Taiga ID: #525** |
|  | |

**Test Plan for Sprint 3**

|  |  |
| --- | --- |
| **Sprint 3** | **Taiga ID: #301, #311, #321, #331, #341, #351, #361, #371, #381, #391, #401** |
| **Introduction:**  The purpose of this test plan is to ensure User Administrator can suspend or unsuspend User Accounts, and User Profiles. A suspended user account , or all user accounts under a suspend profile should not be able to log in. We make sure cleaners can check out their stats (view counts, shortlisted counts) etc so that they can know about their service popularity and visibility. Also some CRUDS, across all user roles.  **Objectives:**   1. Ensure suspend user account and suspend profile , stops people from logging in. 2. To let cleaners track their service and profile stats.(counts) 3. Some SEARCH and DELETE queries.   **Resources:**  Test Environment: Local Environment Setup  Testing Machine:   1. Operating System:  Windows 11 Home, Version 24H2, OS Build 26100.4061 2. Processor: 11th Gen Intel(R) Core(™) i7-11370H @ 3.30GHz 3.30 GHz 3. Installed RAM: 16.0 GB 4. System Type: 64-bit operating system, x64-based processor   **Preconditions to run test:**  These need to be installed before running the test:   1. IntelliJ IDEA Community Edition 2025.1 2. MySQL Server (version 8.0.42) 3. Java (Eclipse temurin 22.0.2) 4. Java Fx (javafx-sdk-24.0.1) 5. JDBC-Connector (version 8.3.0)   **Test Completion Criteria:**  All test cases are accepted with no bugs or errors and actual test results match the expected results. The test cases will be rejected and re-done if any of the test cases were to fail.  Through the execution of this test plan, we aim to make suspending works and cleaners can track their stats. | |

**User Stories for Sprint 3**

|  |  |
| --- | --- |
| **Sprint 3** | |
| UA-06 | As a User Administrator, I want to suspend user accounts so that I can restrict access when necessary to maintain system security. |
| UA-11 | As a User Administrator, I want to suspend user profiles so that I can temporarily restrict access. |
| HO-04 | As a Home Owner, I want to search for cleaning services so that I can find services that meet my needs. |
| HO-07 | As a Home Owner, I want to search my favourite list so that I can find the shortlisted cleaning services easily. |
| HO-09 | As a Home Owner, I want to search my cleaning history by service and date range, so that I can easily find a past booking. |
| CL-06 | As a Cleaner, I want to delete my cleaning services so that I can remove outdated ones. |
| CL-08 | As a Cleaner, I want to view how many times my profile was viewed so that I can track service visibility. |
| CL-09 | As a Cleaner, I want to view how many times I was shortlisted so that I can gauge my service popularity. |
| CL-10 | As a Cleaner, I want to view my confirmed match history filtered by services and date period so that I can review my completed jobs. |
| PM-06 | As a Platform Manager, I want to delete cleaning services categories so that I can remove outdated or unused categories. |
| PM-07 | As a Platform Manager, I want to search cleaning services categories so that I can find specific categories quickly. |

**[Sprint 3] Stakeholder #1 – User Administrator**

|  |  |
| --- | --- |
| **User Story #UA-06** | **Taiga ID: #6** |
| As a User Administrator, I want to suspend user accounts so that I can restrict access when necessary to maintain system security. | |

|  |  |
| --- | --- |
| **Use Case Diagram #UA-06** | **Taiga ID: #294** |
| A diagram of a system  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #UA-06** | **Taiga ID: #295** |
| **Name:** Suspend User Account | |
| **Stakeholders and Goals:** User Administrator – wants to suspend user accounts in order to restrict access to the platform and enforce platform policies when necessary. | |
| **Description:** The User Administrator logs into the system and navigates to the User Management Dashboard to suspend a specific user account. Once suspended, the affected user will be unable to log in or access platform functionalities, thereby maintaining system security. | |
| **Actors:** User Administrator | |
| **Trigger:** User Administrator clicks on the “Suspend User Account” button. | |
| **Pre-Condition:**   1. The User Administrator is logged into the system with administrative privileges. 2. The user account to be suspended exists in the system. | |
| **Normal Flow:**   1. The User Administrator selects the target user account, changes the account status to "Suspended," and clicks the "Suspend User Account" button. 2. The system updates the account status, restricts login access for the suspended user, and displays a confirmation message. 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  **2a. Account Status Not Selected:** If the account status field is left empty, the system will display the error message stating “Please select a valid status.” | |

|  |  |
| --- | --- |
| **BCE Diagram #UA-06** | **Taiga ID: #296** |
|  | |

|  |  |
| --- | --- |
| **Sequence Diagram #UA-06** | **Taiga ID: #297** |
|  | |

|  |  |
| --- | --- |
| **Wireframe #UA-06** | **Taiga ID: #300** |
| **Suspend User Account Page**    **Suspend user account function** | |

|  |  |
| --- | --- |
| **Test Case #UA-06** | **Taiga ID: #302** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC  #UA-06  -1 | Start Suspend User Account | NIL | Suspend User Account Page Appear | Suspend User Account Page Appear | pass | - | - | | TC  #UA-06  -2 | Change account status to “Suspended” | Account status = “Suspended” | User Suspended successfully,  Show updated User Account Data | User Suspended successfully,  Show updated User Account Data | pass | - | - | | TC  #UA-06  -3 | Blank Fields | Account status = blank | Error message : “Please select a valid account status” | Error message : “Please select a valid account status” | pass | - | - | | |

|  |  |
| --- | --- |
| **User Story #UA-11** | **Taiga ID: #11** |
| As a User Administrator, I want to suspend user profiles so that I can temporarily restrict access. | |

|  |  |
| --- | --- |
| **Use Case Diagram #UA-11** | **Taiga ID: #304** |
| A diagram of a system  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #UA-11** | **Taiga ID: #305** |
| **Name:** Suspend User Profile | |
| **Stakeholders and Goals:** User Administrator – wants to suspend a user profile to temporarily restrict the associated user account from accessing specific platform functionalities. | |
| **Description:** The User Administrator logs into the system and navigates to the User Profile Management section to suspend a specific user profile. Once suspended, any user associated with that profile will have their access limited based on platform rules. | |
| **Actors:** User Administrator | |
| **Trigger:** User Administrator clicks on the “Suspend User Profile” button. | |
| **Pre-Condition:**   1. The User Administrator is logged into the system with admin privileges. 2. The user profile to be suspended exists in the system. | |
| **Normal Flow:**   1. The User Administrator selects the target user profile, chooses the "Suspended" option from the status dropdown, and clicks the "Update Status" button. 2. The system updates the status of the selected user profile in the database and displays a confirmation message. 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  **2a. No User Profile Selected:** If the User Administrator does not select any user profile, the system will display the error message stating “Please select a valid user profile.” | |

|  |  |
| --- | --- |
| **BCE Diagram #UA-11** | **Taiga ID: #306** |
|  | |

|  |  |
| --- | --- |
| **Sequence Diagram #UA-11** | **Taiga ID: #307** |
|  | |

|  |  |
| --- | --- |
| **Wireframe #UA-11** | **Taiga ID: #310** |
| **Suspend User Profile Page**    **Suspend User Profile function** | |

|  |  |
| --- | --- |
| **Test Case #UA-11** | **Taiga ID: #312** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC  #UA-11  -1 | Start Suspend User Profile | NIL | Suspend User Profile Page Appear | Suspend User Profile Page Appear | pass | - | - | | TC  #UA-11  -2 | Select valid user profile and Status | User profile = User administrator  And  Status = Suspended | Profile Suspended successfully,  Show updated User Profile Data | Profile Suspended successfully,  Show updated User Profile Data | pass | - | - | | TC  #UA-11  -3 | No User Profile selected | User profile  = Blank | Error message: “Please select a valid user profile” | Error message: “Please select a valid user profile” | pass | - | - | | TC  #UA-11  -4 | Suspended User Admin Profile | User Profile = “1 – User Administrator” | Error message:  “User Administrator Cannot be deleted” | NIL | fail | User Admin profile should not be suspended, it will ruin the admin function as we cannot log in as user annymore | Suspended User Admin Profile Error |   **Suspended User Admin Profile Error**    **Log In page after suspended “User Administrator”** | |

**[Sprint 3] Stakeholder #2 – Home Owner**

|  |  |
| --- | --- |
| **User Story #HO-04** | **Taiga ID: #16** |
| As a Home Owner, I want to search for cleaning services so that I can find services that meet my needs. | |

|  |  |
| --- | --- |
| **Use Case Diagram #HO-04** | **Taiga ID: #314** |
| A black and white image of a search engine service  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #HO-04** | **Taiga ID: #315** |
| **Name:** Search Cleaning Services | |
| **Stakeholders and Goals:** Home Owner – wants to search for available cleaning services using keywords or filters to quickly find suitable options for booking. | |
| **Description:** The Home Owner logs into the system and navigates to the cleaning services management page to search through available services. They can enter a keyword into the search bar to filter the list of services and view relevant results based on their preferences. | |
| **Actors:** Home Owner | |
| **Trigger:** Home Owner clicks on the “Search” button | |
| **Pre-Condition:**   1. The Home Owner has access to the cleaning services management page. 2. Cleaning service records are stored in the system. | |
| **Normal Flow:**   1. The Home Owner enters a search keyword. 2. Home Owner clicks the “Search” button. 3. The system processes the input and displays a list of matching cleaning services. 4. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  **2a. No Matching Services Found:** If no cleaning services match the entered keyword, the system displays an error message stating “No matching services found.” | |

|  |  |
| --- | --- |
| **BCE Diagram #HO-04** | **Taiga ID: #316** |
|  | |

|  |  |
| --- | --- |
| **Sequence Diagram #HO-04** | **Taiga ID: #317** |
|  | |

|  |  |
| --- | --- |
| **Wireframe #HO-04** | **Taiga ID: #320** |
| **Search Cleaning Service Page** | |

|  |  |
| --- | --- |
| **Test Case #HO-04** | **Taiga ID: #322** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC  #HO-4  -1 | Start“Search Cleaning Services” | NIL | Search Cleaning services page appears | Search Cleaning services page appears | pass | - | - | | TC  #HO-4  -2 | Enter a valid keyword and press the “Search” button. | service1 | Related cleaning service detail is displayed | Related cleaning service detail is displayed | pass | - | - | | |

|  |  |
| --- | --- |
| **User Story #HO-07** | **Taiga ID: #19** |
| As a Home Owner, I want to search my favourite list so that I can find the shortlisted cleaning services easily. | |

|  |  |
| --- | --- |
| **Use Case Diagram #HO-07** | **Taiga ID: #324** |
|  | |

|  |  |
| --- | --- |
| **Use Case Description #HO-07** | **Taiga ID: #325** |
| **Name:** Search Services Favourite List | |
| **Stakeholders and Goals:** Home Owner – wants to easily locate specific shortlisted cleaning services by searching within their favourite list. | |
| **Description:** The Home Owner logs into the system and accesses the Favourite List page to search through previously shortlisted cleaning services. This allows the Home Owner to quickly find a specific service based on keywords. | |
| **Actors:** Home Owner | |
| **Trigger:** Home Owner clicks on the “Search Favourite List” button. | |
| **Pre-Condition:**   1. Must be logged in as Home Owner. 2. Must have one or more cleaning services in the favourite list. | |
| **Normal Flow:**   1. The Home Owner enters a keyword in the search bar 2. Home Owner clicks the “Search” button. 3. The system filters and displays matching shortlisted cleaning services. 4. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

|  |  |
| --- | --- |
| **BCE Diagram #HO-07** | **Taiga ID: #326** |
|  | |

|  |  |
| --- | --- |
| **Sequence Diagram #HO-07** | **Taiga ID: #327** |
|  | |

|  |  |
| --- | --- |
| **Wireframe #HO-07** | **Taiga ID: #330** |
| **Search Service Favourite List** | |

|  |  |
| --- | --- |
| **Test Case #HO-07** | **Taiga ID: #332** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC  #HO-7  -1 | Press “Search Favourite List” button | NIL | Search Favourite List page appears | Search Favourite List page appears | pass | - | - | | TC  #HO-7  -2 | Select a serviceid and press “Search” | Service63 | Data from user’s favourite appears | Data from user’s favourite appears | pass | - | - | | |

|  |  |
| --- | --- |
| **User Story #HO-09** | **Taiga ID: #21** |
| As a Home Owner, I want to search my cleaning history by service and date range, so that I can easily find a past booking. | |

|  |  |
| --- | --- |
| **Use Case Diagram #HO-09** | **Taiga ID: #334** |
|  | |

|  |  |
| --- | --- |
| **Use Case Description #HO-09** | **Taiga ID: #335** |
| **Name:** Search Service History | |
| **Stakeholders and Goals:** Home Owner – wants to find service history filtered by service type and date period. | |
| **Description:** Home Owners can search their service history by selecting specific services and filtering by date range in order to quickly locate past jobs. | |
| **Actors:** Home Owner | |
| **Trigger:** Home Owner clicks on the “Search Service History” button | |
| **Pre-Condition:**   1. Home Owner has access to the Home Owner Services page. 2. Past service records are available in the system. | |
| **Normal Flow:**   1. Home Owner selects a date range and service type, then presses the “Search” button. 2. The system processes the input and displays the corresponding data. 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

|  |  |
| --- | --- |
| **BCE Diagram #HO-09** | **Taiga ID: #336** |
| A diagram of a company  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Sequence Diagram #HO-09** | **Taiga ID: #337** |
| A diagram of a company  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Wireframe #HO-09** | **Taiga ID: #340** |
| **Search Service History Page**    **Search Service History Page – “After applied date”**    **Search Service History Page – “After applied date and service”** | |

|  |  |
| --- | --- |
| **Test Case #HO-09** | **Taiga ID: #342** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC #HO-09 - 1 | Press “Search Service History” button | NIL | Search Service History page appears | Search Service History page appears | pass | - | - | | TC #HO-09 - 2 | Select all services and press “Search” | NIL | Data related to user’s service history appears | Data related to user’s service history appears | pass | - | - | | |

**[Sprint 3] Stakeholder #3 – Cleaner**

|  |  |
| --- | --- |
| **User Story #CL-06** | **Taiga ID: #27** |
| As a Cleaner, I want to delete my cleaning services so that I can remove outdated ones. | |

|  |  |
| --- | --- |
| **Use Case Diagram #CL-06** | **Taiga ID: #344** |
| A close-up of a sign  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #CL-06** | **Taiga ID: #345** |
| **Name:** Delete Cleaning Service | |
| **Stakeholders and Goals:** Cleaner – wants to delete outdated or inactive cleaning services to ensure clients only see currently available offerings. | |
| **Description:** The Cleaner logs into the system and navigates to the services section to delete any outdated or no longer active cleaning services. This helps reduce confusion for clients and ensures that only valid services are listed on the platform. | |
| **Actors:** Cleaner | |
| **Trigger:** Cleaner clicks on the “Delete Cleaning Service” button. | |
| **Pre-Condition:**   1. The Cleaner is logged into the system with access to cleaner functionalities. 2. The cleaning service exists in the system and belongs to the logged-in Cleaner. | |
| **Normal Flow:**   1. The Cleaner selects the cleaning service they wish to delete and clicks the "Delete Service" button. 2. The system deletes the selected cleaning service from the database and displays a confirmation message to the Cleaner. 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  **2a. No Cleaning Service Selected:** If the Cleaner does not select a cleaning service before attempting to delete, the system will display the error message stating “Please select a cleaning service to delete.” | |

|  |  |
| --- | --- |
| **BCE Diagram #CL-06** | **Taiga ID: #346** |
|  | |

|  |  |
| --- | --- |
| **Sequence Diagram #CL-06** | **Taiga ID: #347** |
|  | |

|  |  |
| --- | --- |
| **Wireframe #CL-06** | **Taiga ID: #350** |
| **Delete cleaning service page** | |

|  |  |
| --- | --- |
| **Test Case #CL-06** | **Taiga ID: #352** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC  #CL-06  -1 | Start Delete Cleaning Service | NIL | Delete Cleaning Service Page Appear | Delete Cleaning Service Page Appear | pass | - | - | | TC  #CL-06  -2 | Select a cleaning service  To delete | A valid Cleaning Service is selected | Cleaning Service deleted,  Show remain Cleaning Service | Cleaning Service deleted,  Show remain Cleaning Service | pass | - | - | | TC  #CL-06  -3 | No Cleaning Service selected | Selected Cleaning Service = NULL | Error message : “Please select a service to delete” | Error message : “Please select a service to delete” | pass | - | - | | |

|  |  |
| --- | --- |
| **User Story #CL-08** | **Taiga ID: #29** |
| As a Cleaner, I want to view how many times my profile was viewed so that I can track service visibility. | |

|  |  |
| --- | --- |
| **Use Case Diagram #CL-08** | **Taiga ID: #354** |
| A diagram of a view profile  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #CL-08** | **Taiga ID: #355** |
| **Name:** View Profile Views | |
| **Stakeholders and Goals:** Cleaner – wants to track the visibility of their profile by viewing how many times it has been viewed by potential customers. | |
| **Description:** The Cleaner logs into the system and navigates to the Cleaner Dashboard to view the number of times their profile has been viewed. This helps the Cleaner assess the visibility and popularity of their profile to improve engagement or service offerings if needed. | |
| **Actors:** Cleaner | |
| **Trigger:** Cleaner clicks on the “View Profile Views” button. | |
| **Pre-Condition:** The Cleaner must be logged into the system | |
| **Normal Flow:**   1. The Cleaner clicks on the "View Profile Views" button. 2. The system retrieves and displays the total number of views for the Cleaner's profile. 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  **2a. Insufficient Data:** If the system is unable to retrieve the profile view count due to a data issue or backend error, the system will display message stating “Unable to retrieve profile views.” | |

|  |  |
| --- | --- |
| **BCE Diagram #CL-08** | **Taiga ID: #356** |
| Picture 1, Picture | |

|  |  |
| --- | --- |
| **Sequence Diagram #CL-08** | **Taiga ID: #357** |
| Picture 1, Picture | |

|  |  |
| --- | --- |
| **Wireframe #CL-08** | **Taiga ID: #360** |
| **View Profile Views**  Picture 1, Picture | |

|  |  |
| --- | --- |
| **Test Case #CL-08** | **Taiga ID: #362** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC #CL-08-1 | Click “View profile views” on dashboard | Cleaner ID = 7 | Services table displays cleaner1’s listed services. | Services table displays cleaner1’s listed services. | Pass | - | - | | TC #CL-08-2 | System fails to retrieve view count | Cleaner ID = 20 (test DB failure) | Message displayed: "Unable to retrieve profile views." | Message displayed: "Unable to retrieve profile views." | Pass | - | - | | |

|  |  |
| --- | --- |
| **User Story #CL-09** | **Taiga ID: #30** |
| As a Cleaner, I want to view how many times I was shortlisted so that I can gauge my service popularity. | |

|  |  |
| --- | --- |
| **Use Case Diagram #CL-09** | **Taiga ID: #364** |
| A diagram of a shortlist  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #CL-09** | **Taiga ID: #365** |
| **Name:** View Shortlist Count | |
| **Stakeholders and Goals:** Cleaner – wants to view how often their services have been shortlisted to understand service popularity and customer interest. | |
| **Description:** The Cleaner logs into the system and navigates to the Cleaner Dashboard to view the total number of times their services have been shortlisted. This allows the Cleaner to assess demand and optimize their service offerings accordingly. | |
| **Actors:** Cleaner | |
| **Trigger:** Cleaner clicks on the “View Shortlist Count” button. | |
| **Pre-Condition:** The Cleaner is logged into the system | |
| **Normal Flow:**   1. The Cleaner clicks on the "View Shortlist Count" button. 2. The system retrieves and displays the total number of times the Cleaner's services have been shortlisted. 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  **2a. Insufficient Data:** If the system fails to retrieve shortlist data due to an error or missing records, the system will display message stating “Unable to retrieve shortlist data.” | |

|  |  |
| --- | --- |
| **BCE Diagram #CL-09** | **Taiga ID: #366** |
| Picture 1, Picture | |

|  |  |
| --- | --- |
| **Sequence Diagram #CL-09** | **Taiga ID: #367** |
| Picture 1, Picture | |

|  |  |
| --- | --- |
| **Wireframe #CL-09** | **Taiga ID: #370** |
| **View Shortlist Count**  Picture 1, Picture | |

|  |  |
| --- | --- |
| **Test Case #CL-09** | **Taiga ID: #372** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC #CL-09-1 | Click “View Shortlist Count” on dashboard | Cleaner ID = 7 | Label displays: “You have been shortlisted 386 times.” | Label displays: “You have been shortlisted 386 times.” | Pass | - | - | | TC #CL-09-2 | Simulate DB error or no data for cleaner | Cleaner ID = 999 | Error message shown: “Unable to retrieve shortlist data.” | Error message shown: “Unable to retrieve shortlist data.” | Pass | - | - | | |

|  |  |
| --- | --- |
| **User Story #CL-10** | **Taiga ID: #31** |
| As a Cleaner, I want to view my confirmed match history filtered by services and date period so that I can review my completed jobs. | |

|  |  |
| --- | --- |
| **Use Case Diagram #CL-10** | **Taiga ID: #374** |
| A diagram of a view match  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #CL-10** | **Taiga ID: #375** |
| **Name:** View Confirmed Match History | |
| **Stakeholders and Goals:** Cleaner – wants to view and filter their confirmed match history by service type and date to assess past jobs | |
| **Description:** The Cleaner logs into the system and navigates to the Cleaner Dashboard to view a list of confirmed jobs that have been matched with them. The Cleaner can filter the results by service title and a specific date period to help track work history and performance over time. | |
| **Actors:** Cleaner | |
| **Trigger:** The Cleaner clicks on the “View Match History” button. | |
| **Pre-Condition:**   1. The Cleaner is logged into the system. 2. The Cleaner has confirmed match records stored in the system. | |
| **Normal Flow:**   1. The Cleaner navigates to the Cleaner Dashboard and clicks on the "View Match History" button. 2. The Cleaner selects a service and date range to filter the history. 3. The system retrieves and displays a list of confirmed match records based on the selected filters. 4. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  **3a. No Confirmed Matches Found:** If no confirmed match records match the selected service and date filters, the system will display message stating “You have no confirmed matches yet.” | |

|  |  |
| --- | --- |
| **BCE Diagram #CL-10** | **Taiga ID: #376** |
| A screenshot of a computer  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Sequence Diagram #CL-10** | **Taiga ID: #377** |
| Picture 1, Picture | |

|  |  |
| --- | --- |
| **Wireframe #CL-10** | **Taiga ID: #380** |
| **View Match History**  Picture 1, Picture  **View Match History – After Date Range Filter Applied**  Picture 1, Picture  **View Match History – After Service Filter Applied**  Picture 1, Picture | |

|  |  |
| --- | --- |
| **Test Case #CL-10** | **Taiga ID: #382** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC #CL-10-1 | Select service and date range, then click “View Match History” | service = "service97", dateFrom = "2021-01-01", dateTo = "2026-05-30" | Table displays confirmed match records for the selected filters | Table displays confirmed match records for the selected filters | Pass | - | - | | TC #CL-10-2 | Select filter with no confirmed jobs in range | service = "serviceX", dateFrom = "2020-01-01", dateTo = "2020-12-31" | Message shown: “You have no confirmed matches yet.” | Message shown: “You have no confirmed matches yet.” | Pass | - | - | | |

**[Sprint 3] Stakeholder #4 – Platform Manager**

|  |  |
| --- | --- |
| **User Story #PM-06** | **Taiga ID: #38** |
| As a Platform Manager, I want to delete cleaning services categories so that I can remove outdated or unused categories. | |

|  |  |
| --- | --- |
| **Use Case Diagram #PM-06** | **Taiga ID: #384** |
|  | |

|  |  |
| --- | --- |
| **Use Case Description #PM-06** | **Taiga ID: #385** |
| **Name:** Delete Cleaning Services Categories | |
| **Stakeholders and Goals:** Platform Manager – wants to remove outdated or unused cleaning service categories to ensure that only relevant and current offerings are displayed on the platform. | |
| **Description:** The Platform Manager logs into the system and navigates to the Platform Manager Dashboard to delete an existing cleaning service category. This allows them to effectively manage and maintain the relevance of category listings, ensuring that outdated or unused services are removed from the platform. | |
| **Actors:** Platform Manager | |
| **Trigger:** Platform Manager clicks on the “Delete Service Categories” button | |
| **Pre-Condition:**   1. The Platform Manager is logged into the system with appropriate privileges. 2. The Service Category is existed to be deleted in the system. | |
| **Normal Flow:**   1. The Platform Manager chooses the service category to be deleted from the dropdown box, and then clicks the "Delete Category" button. 2. The system deletes the selected category from the database and displays a success message. 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  **2a. No Category Selected:** If the Platform Manager clicks the “Delete Cleaning Service Category” button without selecting a category, the system will display an error message stating “Please choose a valid category.”  **2b. Deletion Failed:** If the system encounters an error while attempting to delete the selected category, it will display an error message stating “Delete unsuccessfully”. | |

|  |  |
| --- | --- |
| **BCE Diagram #PM-06** | **Taiga ID: #386** |
|  | |

|  |  |
| --- | --- |
| **Sequence Diagram #PM-06** | **Taiga ID: #387** |
|  | |

|  |  |
| --- | --- |
| **Wireframe #PM-06** | **Taiga ID: #390** |
| **Delete Service Category page**    **Delete Service Category – After delete “Category1”** | |

|  |  |
| --- | --- |
| **Test Case #PM-06** | **Taiga ID: #392** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC  #PM-06  -1 | Start Delete Service Category | NIL | Delete Service Category Page Appear | Delete Service Category Page Appear | pass | - | - | | TC  #PM -06  -2 | Select a Service Category | A valid Service Category is selected | Service Category deleted,  Show remain Cleaning Service | Service Category deleted,  Show remain Cleaning Service | pass | - | - | | TC  #PM -06  -3 | No Service Category selected | Selected Service Category = NULL | Error message: “Please select a valid service category” | Error message: “Please select a valid service category” | pass | - | - | | |

|  |  |
| --- | --- |
| **User Story #PM-07** | **Taiga ID: #39** |
| As a Platform Manager, I want to search cleaning services categories so that I can find specific categories quickly. | |

|  |  |
| --- | --- |
| **Use Case Diagram #PM-07** | **Taiga ID: #394** |
| A diagram of a search engine  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #PM-07** | **Taiga ID: #395** |
| **Name:** Search Cleaning Services Categories | |
| **Stakeholders and Goals:** Platform Manager – wants to quickly locate specific cleaning service categories for review or management using keyword-based search. | |
| **Description:** The Platform Manager logs into the system and navigates to the Platform Manager Dashboard to search through existing cleaning service categories. This allows them to efficiently locate a specific category based on keyword input or filters. | |
| **Actors:** Platform Manager | |
| **Trigger:** Platform Manager clicks on the “Search Categories” button. | |
| **Pre-Condition:**   1. The Platform Manager is logged into the system with appropriate privileges. 2. There are existing cleaning service categories in the system. | |
| **Normal Flow:**   1. The Platform Manager clicks on the "Search Service Categories" button, enters a keyword into the search bar, and clicks the "Search" button. 2. The system retrieves and displays the matching cleaning service categories in a table format. 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  **2a. No Matching Results:** If no categories match the entered search keyword, the system will display an error message stating“No matching categories found.” | |

|  |  |
| --- | --- |
| **BCE Diagram #PM-07** | **Taiga ID: #396** |
| A screenshot of a computer  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Sequence Diagram #PM-07** | **Taiga ID: #397** |
| A diagram of a service diagram  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Wireframe #PM-07** | **Taiga ID: #400** |
| **Search Service Category** | |

|  |  |
| --- | --- |
| **Test Case #PM-07** | **Taiga ID: #402** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC #PM-07 - 1 | Start“  Search Category” | NIL | Search Service category page appears | Search Service category page appears | pass | - | - | | TC #PM-07 - 2 | Enter a valid keyword and press the “Search” button. | category1 | A row from service categories table with selected data(category1) is displayed | A row from service categories table with selected data(category1) is displayed | pass | - | - | | TC #PM-07 - 3 | Enter a invalid keyword and press the “Search” button. | asdf | No data appear | No data appear | pass | - | - | | |

**UML Diagram for Sprint 3**

|  |  |
| --- | --- |
| **Sprint 3** | **Taiga ID: #298, #308, #318, #328, #338, #348, #358, #368, #378, #388, #398** |
| **A diagram with text on it  AI-generated content may be incorrect.** | |

**Data Persistence Diagram for Sprint 3**

|  |  |
| --- | --- |
| **Sprint 3** | **Taiga ID: #299, #309, #319, #329, #339, #349, #359, #369, #379, #389, #399** |
| **A diagram of a computer  AI-generated content may be incorrect.** | |

**Sprint 4**

**Meeting Minutes for Sprint 4**

|  |  |
| --- | --- |
| **Meeting Minutes for Project Meeting 6** | **Taiga ID: #526** |
| WipeOut  Project Meeting 6  **Date / Time** **:** 14 May 2025 / 1.00pm  **Location :** SIM HQ, Block A, Level 5, LAB A.5.14/A.5.15    **Present :** Chea Darayuth Team Member  Ho Ka Yan Jeslyn Team Member  Hein Htet Zaw Team Member  Kaung Minn Khant Team Member  Koung Khant Saung Team Member  Law Jun Wei Team Member  **Absent**  **:** -  **In Attendance :** Mr. Terence Chew Tutor     |  |  |  | | --- | --- | --- | | **S/N** | **ITEM** | **ACTION** | | **A** | **Procedural (Opening)** |  | | **A1** | **Apologies for Absence** |  | | 1 | There were no absentees. |  | | **B1** | **Matters Arising** |  | | **1**  1.1  1.2  1.3  1.4  1.5  1.6  1.7 | **Document Clarification**  Sequence Diagram Clarification  Jes asked whether combo box display logic should be shown in the sequence diagram.  Mr. Terence clarified that only the return list is needed, not the UI display logic.  Naming Convention Consistency  Yuth was advised to rename all boundary classes from “XXBoundary” to “XXPage” to maintain consistency across the system.  Error Message Handling  Mr. Terence reminded Hein to ensure that all error messages are handled within the boundary class, not in the controller.  Additionally, depending on the number of error cases, Yuth and Hein may need to switch from returning String to returning int for better error handling design.  Admin Report Layout  Kaung requested changes to the layout of his platform manager admin report.  FXML Consistency Check  Hein asked if using FXML versus non-FXML approaches between members would affect code consistency.  Mr. Terence responded that FXML files come before boundaries, so it will not affect consistency. Only Boundary, Controller, and Entity files will be assessed.  Use Case Diagram Format  JunWei showed Mr. Terence three variations of how he represented use case diagrams.  Mr. Terence confirmed that any format is acceptable as long as it matches the user story.  JunWei decided to adopt a “one use case diagram per user story” format for better clarity and navigation during the presentation.  Screenshot UI Consistency  JunWei asked if screenshots from different computers (e.g., Mac vs. Windows) need to be consistent.  Mr. Terence clarified that visual consistency is not required across systems. | Jes to note  Yuth to note  Hein to note  Hein and Yuth to note  Kaung to note  All to note  JunWei to note  All to note | | **2**  2.1  2.2  2.3  2.4 | **Project Timeline and Task Coordination**  Development & Compilation Deadline  All team members are to update their code by the end of today: **14 May 2025 (Wednesday)**.  Hein will begin compiling all submitted code today and aims to complete the compilation by **Friday (16 May 2025)**.  Demo Video Preparation  Jes will begin preparing and recording the demo video on **Saturday (17 May 2025)**.  The video will include walkthroughs for all stakeholder roles with voiceover and role separation.  Documentation Workflow  JunWei will complete the first draft of the final report **by Friday (16 May 2025)** and share it with the team in view-only mode.  Group members are to review and provide feedback by end of **Saturday (17 May 2025)**.  JunWei will then update and finalize the documentation on **Sunday (18 May 2025)**.  Only JunWei will make edits to the final file to ensure consistency.  Code & Report Consistency  The team agreed to ensure codebase formatting, file naming conventions, and written documentation follow unified standards before submission. | All to note  Hein to note  Jes to note  All to note  All to note | | **3** | **Final Submission Checklist**   1. **Final Report**  * Sprint-based UML Class Diagram * Sprint-based Data Persistence Diagrams  1. **Source Code**  * Must be zipped in .zip format (no .rar allowed)  1. **Taiga ID** 2. **Demo Video** in .mp4 format  * Must include walkthrough of all main actor functions. * Must feature voiceover. * Must use separate accounts for each role  1. **Gannt Chart** (Assigned to: JunWei) 2. **Other supporting annexes** (PDF format) | All to note  JunWei to note  Hein to note  JunWei to note  Jes to note  JunWei to note | | **4** | **Any Other Business**  There were no other matters raised. |  | | **C**  1  2  3  4  5  6  7  8  9  10  11  12 | **Actions and Responsibilities**  Rename XXBoundary to XXPage for consistency.  Move error message handling to boundary class.  Change error return type to int if needed.  Update platform manager admin report layout.  Finalize use case diagram format.  Ensure code is updated by 14 May.  Compile and integrate code by 16 May.  Complete first draft of report by 16 May.  Record and edit demo video on 17 May.  Review and send feedback on report by 17 May.  Finalize and edit final report on 18 May.  Ensure all submission items are complete by 18 May, 11:59pm. | Yuth to note  Yuth and Hein to note  Yuth and Hein to note  Kaung to note  JunWei to note  All to note  Hein to note  JunWei to note  Jes to note  All to note  JunWei to note  All to note | | **D**  **1** | **Date of Next Meeting**  The next project meeting is tentatively scheduled for 18 May 2025, 12.00pm.  Any changes to the meeting schedule will be informed via the Group Telegram Chat.  Meeting was adjourned at 2.50pm. |  | |  | Recorded By: Law Jun Wei  Vetted By: Jeslyn Ho Ka Yan |  | | |

|  |  |
| --- | --- |
| **Meeting Minutes for Project Meeting 7** | **Taiga ID: #527** |
| WipeOut  Project Meeting 7  **Date / Time** **:** 18 May 2025 / 1.06pm  **Location :** e-Meeting on Discord    **Present :** Chea Darayuth Team Member  Ho Ka Yan Jeslyn Team Member  Hein Htet Zaw Team Member  Kaung Minn Khant Team Member  Koung Khant Saung Team Member  Law Jun Wei Team Member  **Absent**  **:** -  **In Attendance :** -     |  |  |  | | --- | --- | --- | | **S/N** | **ITEM** | **ACTION** | | **A** | **Procedural (Opening)** |  | | **A1** | **Apologies for Absence** |  | | 1 | There were no absentees. |  | | **B1** | **Matters Arising** |  | | **1** | **Coding Integration Status**  Integration is in progress, with the only remaining code to be integrated from Koung.  Debugging tasks include:   * PM-04 user story code issues * Suspend function for User Admin (error when suspending a user admin profile) * Cleaner update issue (unable to update service category when editing a cleaning service) | All to note | | **2** | **Code Standards – UML Compliance**  Hein reminded the team that UML class and variable naming conventions must follow camel case.   * E.g., userId, cleanerId, serviceId   Ensure that all userID fields in UML and code follow this format. | All to note | | **3** | **User Stories’ Tasks Progress**  JunWei shared with the team on the user stories’ tasks progress report:   * 40% ported over to final report * 20% ported over to final report but need updates * 40% not ported over to the final report | All to note | | **4** | **Progress on Report Tasks (Others)**   |  |  |  | | --- | --- | --- | | **Task** | **Assigned To** | **Progress Summary** | | Table of Content | JunWei | Not yet added | | Member Contribution | JunWei | Added but needs update | | User Stories Overview | JunWei | Completed | | Meeting Minutes | JunWei | Up to Sprint 3 completed, Sprint 4 short of one more meeting minutes | | Taiga Overview | JunWei | Not yet added | | Gantt Chart | JunWei | Not yet added | | Test Plan | Hein | Not yet added | | User Stories for Sprints | JunWei | Not yet added | | UML Class Diagram | Kaung | Not yet added | | Data Persistence Diagram | Yuth | Completed | | Test Driven Development | Hein | Not yet added | | SQL Schema Table | JunWei | Completed | | Test Data | JunWei | Completed | | Agile Methodology | Jes | Completed | | CI/CD | Hein (AUG) | Not yet added | | Ethical Issues | Koung (FELIZ) | Added but needs revision | | Data-Driven Software Development | Koung (FELIZ) | Added but needs revision | | All to note | | **5** | **Any Other Business**  There were no other matters raised. |  | | **C**  **1**  **2**  **3**  **4**  **5**  **6**  **7**  **8** | **Actions and Responsibilities**   |  |  | | --- | --- | | Final code integration and debug for PM-04, suspend,  and cleaner update  Begin demo video preparation after code integration is completed  UML Class Diagram  Gantt Chart  Data Persistence Diagram  CI/CD  Ethical Issues  Data-Driven Software Development | All Members | | Koung and Hein to note  Jes to note  Kaung to note  JunWei to note  Yuth to note  Hein to note  Koung to note  Koung to note | | **D**  **1** | **Date of Next Meeting**  The will be the final meeting. No more next meeting.  Any changes will be informed via the Group Telegram Chat.  Meeting was adjourned at 2.17pm |  | |  | Recorded By: Law Jun Wei  Vetted By: Ho Ka Yan Jeslyn |  | | |

**Taiga Overview for Sprint 4**

|  |
| --- |
| **Sprint 4** |
| **A screenshot of a computer  AI-generated content may be incorrect.** |

**Gantt Chart Overview for Sprint 4**

|  |  |
| --- | --- |
| **Sprint 4** | **Taiga ID: #528** |
|  | |

**Test Plan for Sprint 4**

|  |  |
| --- | --- |
| **Sprint 4** | **Taiga ID: #411, #421, #431, #441, #451, #461** |
| **Introduction:**  The purpose of this test plan is mainly as a platform manager, we want to generate daily, weekly and monthly reports of stats, calculating revenue and total bookings done on the app during a selected timeframe. Make sure all functions are working and all different codes are integrated together properly.  **Objectives:**  1. Generate daily, weekly and monthly reports so that as a platform manager, we can run stats on our system data.  2. Integrate all codes together and make sure they work combined together.  3. Check thru all user stories we have listed, one by one and check the code satisfies all user stories.  **Resources:**  Test Environment: Local Environment Setup  Testing Machine:   1. Operating System:  Windows 11 Home, Version 24H2, OS Build 26100.4061 2. Processor: 11th Gen Intel(R) Core(™) i7-11370H @ 3.30GHz 3.30 GHz 3. Installed RAM: 16.0 GB 4. System Type: 64-bit operating system, x64-based processor   **Preconditions to run test:**  These need to be installed before running the test:   1. IntelliJ IDEA Community Edition 2025.1 2. MySQL Server (version 8.0.42) 3. Java (Eclipse temurin 22.0.2) 4. Java Fx (javafx-sdk-24.0.1) 5. JDBC-Connector (version 8.3.0)   **Test Completion Criteria:**  All test cases are accepted with no bugs or errors and actual test results match the expected results. The test cases will be rejected and re-done if any of the test cases were to fail.  This test plan aims to ensure functionality of generating daily reports, weekly reports, monthly reports and combining all the finished parts together. Through the execution of this test plan, we aim to identify and resolve any potential issues, ensuring a smooth and working application for all users. | |

**User Stories for Sprint 4**

|  |  |
| --- | --- |
| **Sprint 4** | |
| CL-07 | As a Cleaner, I want to search my services so that I can find specific services efficiently. |
| CL-11 | As a Cleaner, I want to search my confirmed match history filtered by services and date period so that I can efficiently find past jobs. |
| PM-04 | As a Platform Manager, I want to view cleaning services categories so that I can monitor existing categories. |
| PM-08 | As a Platform Manager, I want to generate daily reports so that I can monitor short-term activity. |
| PM-09 | As a Platform Manager, I want to generate weekly reports so that I can track performance over the week. |
| PM-10 | As a Platform Manager, I want to generate monthly reports so that I can analyze long-term performance. |

**[Sprint 4] Stakeholder #3 – Cleaner**

|  |  |
| --- | --- |
| **User Story #CL-07** | **Taiga ID: #28** |
| As a Cleaner, I want to search my services so that I can find specific services efficiently. | |

|  |  |
| --- | --- |
| **Use Case Diagram #CL-07** | **Taiga ID: #404** |
| A black and white image of a search engine service  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #CL-07** | **Taiga ID: #405** |
| **Name:** Search Cleaning Services | |
| **Stakeholders and Goals:** Cleaner – wants to access specific cleaning services efficiently. | |
| **Description:** Cleaners wants to access cleaning services efficiently by filter out the cleaning service that they want to see the details. | |
| **Actors:** Cleaner | |
| **Trigger:** Cleaner selects the “Search Cleaning Services” button. | |
| **Pre-Condition:**   1. Cleaner has access to the “Cleaner Dashboard”. 2. Cleaner has services inside the database. | |
| **Normal Flow:**   1. Cleaner enters the service name and presses the “Search” button. 2. System processes the input and displays matching data. 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

|  |  |
| --- | --- |
| **BCE Diagram #CL-07** | **Taiga ID: #406** |
|  | |

|  |  |
| --- | --- |
| **Sequence Diagram #CL-07** | **Taiga ID: #407** |
|  | |

|  |  |  |
| --- | --- | --- |
| **Wireframe #CL-07** | **Taiga ID: #410** | |
| **Search Cleaning Service Page**    **Search Cleaning Service Page – “After applied service & date”** | | |
| **Test Case #CL-07** | **Taiga ID: #412** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC #CL-07 - 1 | Start“  Search service” | NIL | Search Services page appears | Search Services page appears | Pass | - | - | | TC #CL-07 - 2 | Enter a valid keyword and press the “Search” button. | service40 | Relevant data appears | Relevant data appears | Pass | - | - | | TC #CL-07 - 3 | press the “Search” button. | NIL | Data related to the cleaner appears | Data related to the cleaner appears | Pass | - | - | | |

|  |  |
| --- | --- |
| **User Story #CL-11** | **Taiga ID: #32** |
| As a Cleaner, I want to search my confirmed match history filtered by services and date period so that I can efficiently find past jobs. | |

|  |  |
| --- | --- |
| **Use Case Diagram #CL-11** | **Taiga ID: #414** |
| A black and white image of a search box  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #CL-11** | **Taiga ID: #415** |
| **Name:** Search Match History | |
| **Stakeholders and Goals:** Cleaner – wants to find completed or matched jobs. | |
| **Description:** Cleaners can view their matched service history filtered by service type and date range. | |
| **Actors:** Cleaner | |
| **Trigger:** Cleaner selects the “Search Match History” button. | |
| **Pre-Condition:**  1. Cleaner has access to the “Cleaner Dashboard”.  2. Cleaner has a history of confirmed matches. | |
| **Normal Flow:**   1. Cleaner selects the “Service Title”, chooses “Start Date” and “End Date”, then presses the “Search” button. 2. System processes the request and displays matching data upon submission. 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  2a. **No Result Found** – If there is no data during the selected date range, the system returns no value. | |

|  |  |
| --- | --- |
| **BCE Diagram #CL-11** | **Taiga ID: #416** |
|  | |

|  |  |
| --- | --- |
| **Sequence Diagram #CL-11** | **Taiga ID: #417** |
|  | |

|  |  |
| --- | --- |
| **Wireframe #CL-11** | **Taiga ID: #420** |
| **View Match History Page**    **View Match History – “After Date Filter Applied”**    **View Match History – After Service Filter Applied** | |

|  |  |
| --- | --- |
| **Test Case #CL-11** | **Taiga ID: #422** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC #CL-11 - 1 | Start“  Search Match History” | NIL | Search Match History page appears | Search Match History page appears | Pass | - | - | | TC #CL-11 - 2 | Enter valid inputs and press the “Search” button. | service29,  5/4/2023, 5/4/2025 | All the rows related to the selected data appears | All the rows related to the selected data appears | Pass | - | - | | |

## [Sprint 4] Stakeholder #4 – Platform Manager

|  |  |
| --- | --- |
| **User Story #PM-04** | **Taiga ID: #36** |
| As a Platform Manager, I want to view cleaning services categories so that I can monitor existing categories. | |

|  |  |
| --- | --- |
| **Use Case Diagram #PM-04** | **Taiga ID: #424** |
| A diagram of a service  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #PM-04** | **Taiga ID: #425** |
| **Name:** View Cleaning Service Categories | |
| **Stakeholders and Goals:** Platform Manager – wants to view and monitor the list of cleaning service categories to ensure services are properly organized and reflect the current offerings. | |
| **Description:** The Platform Manager logs into the system and navigates to the Platform Manager Dashboard to view a list of cleaning service categories. This allows the Platform Manager to oversee service organization and ensure that all category information remains accurate and aligned with customer needs. | |
| **Actors:** Platform Manager | |
| **Trigger:** The Platform Manager clicks on the “View Cleaning Service Categories” button. | |
| **Pre-Condition:**   1. The Platform Manager is logged into the system. 2. The account has the appropriate privileges to access cleaning service category data. | |
| **Normal Flow:**   1. The Platform Manager clicks on the "View Cleaning Service Categories" button. 2. The system retrieves and displays a list of all cleaning service categories (e.g., deep cleaning, regular cleaning, specialized services). 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:**  **2a. Insufficient Privileges:** If the Platform Manager does not have the required permissions to access the categories page, the system will displaymessage stating “You do not have the required privileges to view cleaning service categories.” | |

|  |  |
| --- | --- |
| **BCE Diagram #PM-04** | **Taiga ID: #426** |
| Picture 1, Picture | |

|  |  |
| --- | --- |
| **Sequence Diagram #PM-04** | **Taiga ID: #427** |
| A diagram of a service  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Wireframe #PM-04** | **Taiga ID: #430** |
| **View Service Categories**  Picture 1, Picture | |

|  |  |
| --- | --- |
| **Test Case #PM-04** | **Taiga ID: #432** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC #PM-04-1 | Click “View Categories” on dashboard | userID= 3  Username = plateformmanager1 | Service categories table is displayed showing ID, name, description, services | Service categories table is displayed showing ID, name, description, services | Pass | - | - | | TC #PM-04-2 | Access category page without proper role | Logged in as: cleaner1, role = Cleaner | Message shown: “You do not have the required privileges to view cleaning service categories.” | Message shown: “You do not have the required privileges to view cleaning service categories.” | Pass | Cleaners should not access this feature | - | | |

|  |  |
| --- | --- |
| **User Story #PM-08** | **Taiga ID: #40** |
| As a Platform Manager, I want to generate daily reports so that I can monitor short-term activity. | |

|  |  |
| --- | --- |
| **Use Case Diagram #PM-08** | **Taiga ID: #434** |
| A diagram of a general report  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #PM-08** | **Taiga ID: #435** |
| **Name:** Generate Daily Reports | |
| **Stakeholders and Goals:** Platform Manager – wants to monitor short-term activity on the platform by generating daily reports that summarize performance and operations. | |
| **Description:** The Platform Manager logs into the system and navigates to the reporting section to generate a daily report. This report helps the Platform Manager review the number of confirmed bookings and the estimated revenue generated on a specific day, supporting short-term performance tracking and operational decisions. | |
| **Actors:** Platform Manager | |
| **Trigger:** Platform Manager clicks on the “Generate Daily Report” button. | |
| **Pre-Condition:**   1. The Platform Manager is logged into the system with access to reporting functionalities. 2. Data is available for the selected day. | |
| **Normal Flow:**   1. The Platform Manager clicks on the “Generate Daily Report” button, specifies the date and click on “Generate” button. 2. The system retrieves and compiles relevant platform data for the selected date and generates the daily report. 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

|  |  |
| --- | --- |
| **BCE Diagram #PM-08** | **Taiga ID: #436** |
| A screenshot of a computer  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Sequence Diagram #PM-08** | **Taiga ID: #437** |
| A diagram of a diagram  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Wireframe #PM-08** | **Taiga ID: #440** |
| **Generate Daily Report**    **Generate Daily Report – After Applied date “24/2/2025”** | |

|  |  |
| --- | --- |
| **Test Case #PM-08** | **Taiga ID: #442** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC #PM-08-1 | Press “Generate Daily Report” button | NIL | Generate Daily Report page appears | Generate Daily Report page appears | Pass | - | - | | TC #PM-08-2 | Select a date and press “Generate” | 2/26/2025 | Relevant data  appear | Relevant data appears | Pass | - | - | | TC #PM-08-3 | Do not select the date but press “Generate”button | NIL | No data appear | No data appear | Pass | - | - | | |

|  |  |
| --- | --- |
| **User Story #PM-09** | **Taiga ID: #41** |
| As a Platform Manager, I want to generate weekly reports so that I can track performance over the week. | |

|  |  |
| --- | --- |
| **Use Case Diagram #PM-09** | **Taiga ID: #444** |
| A diagram of a generation  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #PM-09** | **Taiga ID: #445** |
| **Name:** Generate Weekly Reports | |
| **Stakeholders and Goals:** Platform Manager – wants to track platform performance over the week to identify trends and support decision-making. | |
| **Description:** The Platform Manager logs into the system and navigates to the reporting section to generate a weekly report. This report helps the Platform Manager review the number of confirmed bookings and estimated revenue across a 7-day period, supporting performance tracking and strategic planning. | |
| **Actors:** Platform Manager | |
| **Trigger:** Platform Manager clicks on the “Generate Weekly Report” button. | |
| **Pre-Condition:**   1. The Platform Manager is logged into the system with access to reporting functionalities. 2. Data is available for the selected 7-day period. | |
| **Normal Flow:**   1. The Platform Manager clicks on the "Generate Weekly Report" button, specifies the start date of the 7-day period and click on “Generate” button. 2. The system retrieves and compiles relevant platform data from the selected week and generates the weekly report. 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

|  |  |
| --- | --- |
| **BCE Diagram #PM-09** | **Taiga ID: #446** |
| A screenshot of a computer  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Sequence Diagram #PM-09** | **Taiga ID: #447** |
| A diagram of a diagram  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Wireframe #PM-09** | **Taiga ID: #450** |
| **Generate Weekly Report**    **Generate Weekly Report – After Date Applied “23/2/2025”** | |

|  |  |
| --- | --- |
| **Test Case #PM-09** | **Taiga ID: #452** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC #PM-09-1 | Press “Generate Weekly Report” button | NIL | Generate weekly page appears | Generate weekly page appears | Pass | - | - | | TC #PM-09-2 | Select a date and press “Generate” | 2/26/2025 | Relevant data from the selected period appears | Relevant data from the selected period appears | Pass | - | - | | TC #PM-09-3 | Do not select the date but press “Generate” button | NIL | No data appear | No data appear | Pass | - | - | | |

|  |  |
| --- | --- |
| **User Story #PM-10** | **Taiga ID: #42** |
| As a Platform Manager, I want to generate monthly reports so that I can analyze long-term performance. | |

|  |  |
| --- | --- |
| **Use Case Diagram #PM-10** | **Taiga ID: #454** |
| A diagram of a person with text  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Use Case Description #PM-10** | **Taiga ID: #455** |
| **Name:** Generate Monthly Reports | |
| **Stakeholders and Goals:** Platform Manager – wants to analyze long-term platform performance to support strategic planning, track user behavior, and assess revenue and service trends. | |
| **Description:** The Platform Manager logs into the system and navigates to the reporting section to generate a monthly report. This report provides insights into platform usage, confirmed bookings, estimated revenue, and other key metrics for the selected month to support long-term decision-making and planning. | |
| **Actors:** Platform Manager | |
| **Trigger:** Platform Manager clicks on the “Generate Monthly Report” button. | |
| **Pre-Condition:**   1. The Platform Manager is logged into the system with access to reporting functionalities. 2. Data is available for the selected month. | |
| **Normal Flow:**   1. The Platform Manager clicks on the "Generate Monthly Report" button, specifies the start date of the 1-month period and click on “Generate” button. 2. The system retrieves and compiles relevant platform data from the selected month and generates the monthly report. 3. End. | |
| **Sub-flows:** None | |
| **Alternative/Exceptional flows:** None | |

|  |  |
| --- | --- |
| **BCE Diagram #PM-10** | **Taiga ID: #456** |
| A screenshot of a computer  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Sequence Diagram #PM-10** | **Taiga ID: #457** |
| A diagram of a diagram  AI-generated content may be incorrect. | |

|  |  |
| --- | --- |
| **Wireframe #PM-10** | **Taiga ID: #460** |
| **Generate Monthly Report**    **Generate Monthly Report – After applied date “1/2/2025”** | |

|  |  |
| --- | --- |
| **Test Case #PM-10** | **Taiga ID: #462** |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Test Step** | **Test Step Direction** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** | **Screenshot if fail** | | TC #PM-10-1 | Press “Generate Monthly Report” button | NIL | Generate Monthly Report page appears | Generate Monthly Report  page appears | Pass | - | - | | TC #PM-10-2 | Select a date and press “Generate” | 2/26/2025 | Relevant data from the selected period appears | Relevant data from the selected period appears | Pass | - | - | | TC #PM-10-3 | Do not select the date but press “Generate” button | NIL | No data appear | No data appear | Pass | - | - | | |

**UML Diagram for Sprint 4**

|  |  |
| --- | --- |
| **Sprint 4** | **Taiga ID: #408, #418, #428, #438, #448, #458** |
| **A screenshot of a computer program  AI-generated content may be incorrect.** | |

## **Data Persistence Diagram for Sprint 4**

|  |  |
| --- | --- |
| **Sprint 4** | **Taiga ID: #409, #419, #429, #439, #449, #459** |
| **A diagram of a computer  AI-generated content may be incorrect.**  **A black and white screen with numbers and text  AI-generated content may be incorrect.** | |

## **SQL Schema Table**

|  |  |  |
| --- | --- | --- |
| **useraccount** | | |
| **Field Name** | **Data Type** | **Remark** |
| userID | int | PK |
| profileID | int | FK to userprofile(profileID) |
| username | varchar(45) | - |
| password | varchar(45) | - |
| fullname | varchar(60) | - |
| accountStatus | varchar(45) | Eg. Active, Suspended |
| **userprofile** | | |
| **Field Name** | **Data Type** | **Remark** |
| profileID | int | PK |
| profileName | varchar(45) | - |
| profileDescription | longtext | - |
| profileStatus | varchar(45) | Eg. Active, Suspended |
| **cleaningservices** | | |
| **Field Name** | **Data Type** | **Remark** |
| serviceID | int | PK |
| cleanerID | int | FK to useraccount(userID) |
| serviceTitle | varchar(255) | - |
| serviceDescription | longtext | - |
| price | float | - |
| viewCount | int | - |
| shortlistCount | int | - |
| dateCreated | date | - |
| **servicecategory** | | |
| **Field Name** | **Data Type** | **Remark** |
| categoryID | int | PK |
| categoryName | varchar(60) | - |
| categoryDescription | longtext | - |
| **servicecategorymap** | | |
| **Field Name** | **Data Type** | **Remark** |
| serviceID\* | int | PK, FK to cleaningservices(serviceID) |
| categoryID\* | Int | PK, FK to servicecategory(categoryID) |
| \*Composite PK = (serviceID, categoryID) | | |
| **serviceavailability** | | |
| **Field Name** | **Data Type** | **Remark** |
| availabilityID | int | PK |
| serviceID | int | FK to cleaningservices(serviceID) |
| availableDate | date | - |
| availableTimeSlot | varchar(45) | - |
| **shortlist** | | |
| **Field Name** | **Data Type** | **Remark** |
| homeownerID\* | int | PK, FK to useraccount(userID) |
| serviceID\* | int | PK, FK to cleaningservices(serviceID) |
| \*Composite PK = (homeownerID, cleaningservicesID) | | |
| **bookinghistory** | | |
| **Field Name** | **Data Type** | **Remark** |
| bookingID | int | PK |
| serviceID | int | FK to cleaningservices(serviceID) |
| cleanerID | int | FK to useraccount(userID) |
| homeownerID | int | FK to useraccount(userID) |
| price | float | - |
| bookingDate | date | - |
| serviceDate | date | - |
| serviceTimeSlot | varchar(45) | - |
| bookingStatus | varchar(45) | - |
| **adminreport** | | |
| **Field Name** | **Data Type** | **Remark** |
| reportID | int | PK |
| reportType | varchar(45) | Eg. Daily/Weekly/Monthly |
| reportPeriodStartDate | date | - |
| reportPeriodEndDate | date | - |
| totalBookings | int | - |
| estimatedRevenue | float | - |
| generatedDate | date | - |
| generatedBy | int | FK to useraccount(userID) |

**Test Data**

|  |  |
| --- | --- |
| **Test Data** | **Taiga ID: #52, #62, #72, #82, #93, #103, #113, #123, #133, #143, #153, #163, #173, #183, #193, #203, #213, #223, #233, #243, #253, #263, #273, #283, #293, #303, #313, #323, #333, #343, #353, #363, #373, #383, #393, #403, #413, #423, #433, #443, #453, #463** |
| **userprofile – 4 rows of data**    **useraccount – 100 rows of data**  **A screen shot of a computer  AI-generated content may be incorrect.**  **shortlist – 100 rows of data**  **A screenshot of a black and white table  AI-generated content may be incorrect.**  **servicecategorymap – 268 rows of data**  **A screenshot of a computer  AI-generated content may be incorrect.**  **servicecategory – 100 rows of data**  **A screenshot of a computer  AI-generated content may be incorrect.**  **servicecategory – 119 rows of data**  **A black and white table with numbers and numbers  AI-generated content may be incorrect.**  **cleaningservices – 100 rows of data**  **A black and white table with white text  AI-generated content may be incorrect.**  **bookinghistory – 120 rows of data**  **A black and white table with numbers and numbers  AI-generated content may be incorrect.**  **adminreport – 100 rows of data**  **A screenshot of a black and white table  AI-generated content may be incorrect.** | |

**,**

**Agile Methodology**

|  |  |
| --- | --- |
| **Agile Methodology** | **Taiga ID: #529** |
| For this project, our team adopted the Scrum framework to manage and execute the development of our system. We used Taiga.io to plan, track, and organize tasks across each development phase, from requirements to testing.  The project was divided into 4 sprints, each with defined goals and user stories. Although Scrum typically encourages feature-based development per sprint, we followed a practical model where each team member focused on their assigned use cases throughout the project lifecycle. This approach ensured consistent progress across modules, such as service management, user administration, and report generation.  Taiga.io helped us:   * Break down user stories into tasks (e.g., “View Cleaning Services”, “Generate Monthly Report”) * Track completion status and individual contributions * Maintain clear visibility of what was in progress, completed, or pending.   We conducted:   1. Sprint planning meetings to prioritize essential platform features like service searching and profile management, 2. Mid-sprint reviews to resolve blockers or adjust workloads, 3. Sprint retrospectives to improve collaboration and task flow.   The agile methodology allowed us to iteratively build and test core features of the platform. For example, during the sprint focused on the Home Owner module, we refined the logic and UI for browsing cleaning services—an important part of the platform’s value. The iterative process gave us room to respond to changes in requirements and feedback from our tutor, leading to a more polished and user-centered application.  By applying Scrum and using Taiga.io, we ensured structured progress, clear team coordination, and an adaptive workflow that supported the successful delivery of our cleaning service matching system. | |

## **CI/CD**

|  |  |
| --- | --- |
| **CI/CD** | **Taiga ID: #530** |
| For our wipeout project, we utilized GitHub to integrate Continuous Integration/Continuous Development (CI/CD) methodologies to our coding process. We set a main branch and each of us had a separate branch, we then combined the codes one by one until all codes exist together in main. | |

## **Ethical Issues**

|  |  |
| --- | --- |
| **Ethical Issues** | **Taiga ID: #531** |
| **Ethical Consideration and Discussion**  **Process Ethics**   * **Fairness**   Within our project team, we acknowledged the varying levels of technical proficiency among members. While some had prior experience in backend programming or frontend integration, others were more comfortable with documentation, UI testing, or task coordination. To ensure fairness and inclusiveness, we held an open discussion during sprint planning to match responsibilities to each member's strengths and growth goals. For example, tasks involving database integration and API logic were supported by members with programming expertise, while others handled UI refinement, user testing, and report writing.  This distribution ensured that no member was overwhelmed and that everyone could contribute to the best of their ability. Over time, this approach also enabled skill development, as members who started with non-coding tasks gradually took on more technical responsibilities under guidance. By fostering a supportive and collaborative environment, we promoted equal opportunity and fair contribution, which are critical to ethical teamwork.   * **Accountability**   To ensure we stayed aligned with our project objectives and deadlines, we implemented weekly progress check-ins, sprint retrospectives, and transparent communication using tools like Trello, GitHub, and WhatsApp. Each task assigned was documented and tracked, with completion statuses reviewed as a team.  In terms of code accountability, since the project deals with sensitive operational logic (e.g., booking history, login credentials, notification services), we maintained tight access control on our GitHub repository. Only authenticated members could push changes, and pull requests were reviewed to avoid the risk of accidental deletions or insecure code. This approach helped maintain system integrity and ensured that contributions were traceable and verifiable.  **Product Ethics**   * **Security & Privacy**   Our platform stores and manages user data such as email, phone number, and booking history. To ethically manage this data, we implemented a custom login and role-based access control system in our Java backend. Users must log in to access their account-specific features, and access is restricted by role. Additionally, role-based access control was strictly enforced—for instance:   * + Home Owners can only access their own booking records.   + Cleaners can only view statistics related to their services.   + User Administrators have visibility over user accounts but not personal data like passwords.   Furthermore, we avoided exposing internal database fields such as user IDs or sensitive backend logs to the frontend. This minimizes the surface area for potential exploitation or misuse. Encryption is handled via Firebase’s built-in data protection protocols, and user sessions are securely maintained to prevent unauthorized access.  By implementing these privacy measures, we ensured that personal data was accessed only on a need-to-know basis and was not shared beyond the intended scope—aligning with ethical standards for data protection.   * **Transparency**   Transparency was one of our key design principles. Each feature was implemented with clear feedback mechanisms so users always understand the state of the system and the results of their actions. For example:   * + When Home Owners perform a search, applied filters are visibly displayed and the result count is updated accordingly.   + Cleaners can see how many times their profile has been viewed or shortlisted, giving them insights into their engagement levels.   + Platform Managers have access to daily, weekly, and monthly reports that summarize activity on the platform, ensuring visibility into usage trends and system performance.   This transparency fosters trust between users and the system and helps them make informed decisions while navigating the platform. | |

## 

## **Data-Driven Software Development (DDD)**

|  |  |
| --- | --- |
| **Data-Driven Software Development** | **Taiga ID: #532** |
| **Data-Driven Development**  Data-Driven Development (DDD) enables our platform to make intelligent, personalized decisions based on actual user behaviour and service performance. In the context of our cleaning service platform, DDD is particularly valuable in improving the search and discovery process. Users interact with the platform by entering search queries, filtering results, viewing cleaner profiles, and shortlisting preferred services. These interactions generate meaningful data that can be analysed to enhance search relevance, user satisfaction, and platform engagement.  Instead of displaying search results based only on keyword matches or chronological listings, a data-driven system can learn from past behaviours to predict which services are more likely to meet a user's needs. Over time, this leads to a smarter search function that understands context, preferences, and historical patterns.  **Model Requirements**  To integrate intelligent ranking into our platform, a machine learning model is required to sort and prioritize search results based on relevance. The goal is to display cleaning services that are most aligned with a user's intent and preferences.  The model uses various inputs, including:   * Service metadata: title, description, price, category, and availability * User behaviour: search queries, filters used, services viewed, and shortlists * Interaction metrics: such as how frequently a service is viewed or shortlisted   To successfully implement this machine learning-based feature, the following requirements were identified:  This model is embedded into the backend logic of the search system and operates seamlessly behind the scenes, returning ranked results in real time as users search for services.  **Data Collection**  The success of any machine learning model depends on high-quality, relevant data. For our platform, we collect the following types of data from our database and system logs:   * **Cleaning Service Information**: Each service includes fields such as serviceTitle, serviceDescription, price, viewCount, and shortlistCount from the cleaningservices table. * **Service Categories and Types**: Data from servicecategorymap and servicecategory tables that indicate the nature of each service. * **Availability Records**: Date and time availability from the serviceavailability table that aligns with user filters. * **User Interactions** *(captured through logging)*: Includes search\_query, filters used, user\_id, clicked services, and shortlisted services.   These datasets serve as the foundation for model training, feature engineering, and evaluation.  **Data Cleaning**  Before training the model, we ensure that all collected data is clean, consistent, and usable. Data cleaning steps include:   1. **Removing incomplete entries**: Records missing critical fields (e.g., service title, category, or availability) are excluded. 2. **Text normalization**: Convert service titles and descriptions to lowercase and remove punctuation or special characters to ensure consistency in matching. 3. **Deduplication**: Remove duplicate services or repeated user interaction logs caused by refreshes or page reloads. 4. **Filtering anomalies**: Exclude extreme outliers in viewCount or shortlistCount that might be caused by bots or testing activity. 5. **Handling missing values**: Replace missing values (e.g., null viewCount) with category-level averages to avoid introducing bias.   These steps result in a reliable dataset suitable for training an effective and generalizable model.  **Data Labelling**  To enable supervised learning, we label data based on actual user behaviour:   * **Positive labels**: Services that were clicked or shortlisted by users after appearing in search results. * **Negative labels**: Services that were displayed in search results but ignored by the user.   Each (search session, service) pair becomes a labelled data point. These labels teach the model to differentiate between services that attract attention and those that do not. They also help the model generalize relevance based on service attributes and search context.  **Feature Engineering**  We extract and construct features from raw data to improve the model's performance and decision-making ability. These features include:   * **Text relevance**: Measures similarity between the user’s search query and the service’s title or description. * **Popularity indicators**: Quantifies service popularity using viewCount, shortlistCount, and their ratios. * **Category match score**: Checks alignment between selected category filters and the service’s mapped category. * **Availability alignment**: Verifies whether the service is available during the requested time frame. * **Price relevance**: Assesses whether the service’s price falls within the user’s filter range (if applicable). * **Recency score**: Gives preference to recently updated or newly added services.   These features serve as the input for the model and enable it to make accurate relevance predictions.  **Model Training**  The prepared dataset is used to train a supervised machine learning model, such as:   * **Logistic Regression**: Effective for binary relevance classification. * **Gradient Boosted Trees (e.g., XGBoost)**: Powerful for ranking and capturing non-linear relationships.   Training involves feeding feature vectors and their corresponding labels into the model. The model learns to predict a relevance score for each service based on patterns in the data. We split the dataset into training and testing sets (e.g., 80/20) to ensure that the model generalizes well to unseen data.  **Model Evaluation**  We evaluate the model’s accuracy and effectiveness using several standard metrics:   * **Precision@k**: Measures the proportion of relevant services in the top k search results. * **Recall@k**: Measures how many of the relevant services are included within the top k. * **F1 Score**: Harmonic mean of precision and recall, representing balanced performance. * **Mean Reciprocal Rank (MRR)**: Assesses how early a relevant result appears in the ranking list.   We also compare the model’s performance to a baseline keyword-matching approach to determine improvement levels.  **Model Deployment**  Once validated, the model is deployed as part of the backend search logic. The process works as follows:   1. The user submits a search query with filters (e.g., category or availability). 2. The backend queries the cleaningservices and related tables for candidate services. 3. These services are passed through the trained model, which assigns a relevance score to each. 4. The list is sorted by relevance and returned to the frontend for display.   This deployment makes the system intelligent while keeping the UI and user flow unchanged.  **Model Monitoring**  To ensure ongoing effectiveness, the deployed model is continuously monitored for:   * **Click-through rate (CTR)**: Indicates how often top-ranked services are interacted with. * **Shortlist/book conversion**: Tracks whether users engage further with ranked results. * **Data drift**: Monitors changes in user behaviour or service listings that might reduce model accuracy. * **Model retraining**: The model is updated regularly (e.g., monthly) with new data to stay current and relevant.   A robust monitoring plan ensures that the intelligent ranking system remains aligned with actual platform usage over time. | |

**Gantt Chart Overview**

|  |  |
| --- | --- |
| **Overview** | **Taiga ID: #518, #522, #525, #528** |
|  | |