**Sprints**

**Sprint 1 (Day 1):**

1. Create a specification document and document application flow

2. Create a welcome screen: - Design a welcome screen with the application name and developer details. - Implement options for the user to interact with and access various features of the application.

2. Implement data structures: - Define appropriate data structures to store camera information (brand, model, per-day rental amount) and wallet balance. - Set up data structures for efficient sorting and searching.

4. Implement camera listing: - Create a function to list the cameras available for rent. - Display camera details such as brand, model, and per-day rental amount. - Handle cases where the camera list is empty.

**Sprint 2 (Day 2):**

1. Implement camera rental functionality: - Allow the user to select a camera from the listing to rent. - Check the user's wallet balance and deduct the per-day rental amount if sufficient balance is available. - Display appropriate messages based on the outcome (sufficient balance, insufficient balance).

2. Implement wallet functionality: - Create a sub-menu to add or view the balance in the user's wallet. - Implement the functionality to deposit an amount and update the wallet balance accordingly. - Display messages to indicate the status of the wallet operations.

3. Application flow and exception handling: - Document the application flow in the specification document. - Implement exception handling for scenarios such as invalid user inputs or errors during rental or wallet operations.

4. Test and debug: - Test the implemented features to ensure they function correctly. - Debug any issues or errors that arise during testing