**Group Members:**

Daniyal Khawar (22i-1066)

Muhammad Iqrash Qureshi (22i-1174)

Hammad Shabbir (22i-1140)

**Sprint 1 Module**

User Registration & Authentication Features:

* + User can create an account
  + User can log in/log out
  + Admin can manage users

**A- Software Requirements Specification (SRS)**

**User Stories**

A **User Story** follows this format:

*"****As a*** *[role]****,***

***I want to*** *[action],*

***so that*** *[benefit]."*

### Example User Stories for Sprint 1

1. **User Registration**

*As a vehicle owner, I want to register in the system, so that I can access parking services.*

### User Login

*As a registered user, I want to log in securely, so that I can manage my parking sessions.*

### Admin Management

*As an admin, I want to view and manage registered users, so that I can maintain system security.*

### Password Recovery

*As a user, I want to reset my password, so that I can regain access to my account if I forget it.*

**Structured Specifications (Four per User Story)**

Structured specifications provide a detailed breakdown of how the system should behave for each user story.

1. **User Registration (Structured Specifications)**

|  |  |
| --- | --- |
| **ID** | **Requirement Description** |
| UR-1 | The system must allow users to enter their name, email, password. |
| UR-2 | The system must validate the entered email format and ensure uniqueness |
| UR-3 | The password must meet security requirements (e.g., at least 8 characters, one  special character, etc.). |
| UR-4 | A confirmation message pop up after successful registration. |

1. **User Login (Structured Specifications)**

|  |  |
| --- | --- |
| **ID** | **Requirement Description** |
| UL-1 | The system must allow registered users to log in using their email and password. |
| UL-2 | If incorrect login credentials are entered, the system must display an error message. |
| UL-3 | Upon successful login, the user must be redirected to the dashboard. |

1. **Admin Management (Structured Specifications)**

|  |  |
| --- | --- |
| **ID** | **Requirement Description** |
| AM-1 | The system must allow admins to log in with admin credentials. |
| AM-2 | The admin must be able to view a list of all registered users. |
| AM-3 | The admin must be able to delete users from the system. |

1. **Password Recover (Structured Specifications)**

|  |  |
| --- | --- |
| **ID** | **Requirement Description** |
| PR-1 | The system must provide a "Forgot Password" option on the login page. |
| PR-2 | The new password must meet security requirements before updating. |

**ACCORDING TO NEW PDF**

1. **System Functional Requirements**

|  |  |
| --- | --- |
| ID | Requirement Description |
| FR-1 | The system must allow users to register by entering their name, email, password, phone number, and vehicle details. |
| FR-2 | The system must validate the entered email format and ensure uniqueness. |
| FR-3 | Users must be able to log in securely using their email and password. |
| FR-4 | If incorrect login credentials are entered, an error message must be displayed. |
| FR-6 | Admins must be able to view a list of all registered users. |
| FR-7 | Admins must be able to delete users from the system. |
| FR-9 | Users must be able to reset their password if they forget it. |

1. **System Non Functional Requirements**

### ****Product Requirements****

* The system should be responsive and accessible on Web services
* The system should support secure authentication mechanisms.
* The password reset functionality should ensure secure (token) id-based authentication.

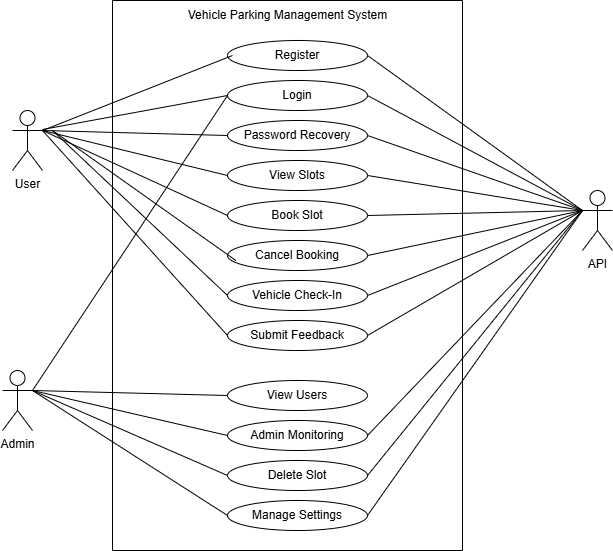
### ****Organizational Requirements****

* User data should be stored securely with encryption.

### ****External Requirements****

* The login authentication process should adhere to industry security standards.

1. **USE CASE**



1. **User Stories with Pre/Post Conditions**

### ****User Registration****

**User Story:** As a vehicle owner, I want to register in the system, so that I can access parking services.

**Pre-Conditions:**

* The user has not previously registered.
* The user must provide valid credentials.

**Post-Conditions:**

* The system sends a confirmation email/SMS.
* The user can now log in.

### ****User Login****

**User Story:** As a registered user, I want to log in securely, so that I can manage my parking sessions.

**Pre-Conditions:**

* The user must be registered in the system.
* The correct email and password must be entered.

**Post-Conditions:**

* The user is redirected to the dashboard.
* The system logs the login activity.

### ****Admin Management****

**User Story:** As an admin, I want to view and manage registered users, so that I can maintain system security.

**Pre-Conditions:**

* The admin must be authenticated.
* The admin must have necessary permissions.

**Post-Conditions:**

* Admin can view or delete user accounts.
* Admin can reset user passwords upon request.

### ****Password Recovery****

**User Story:** As a user, I want to reset my password, so that I can regain access to my account if I forget it.

**Pre-Conditions:**

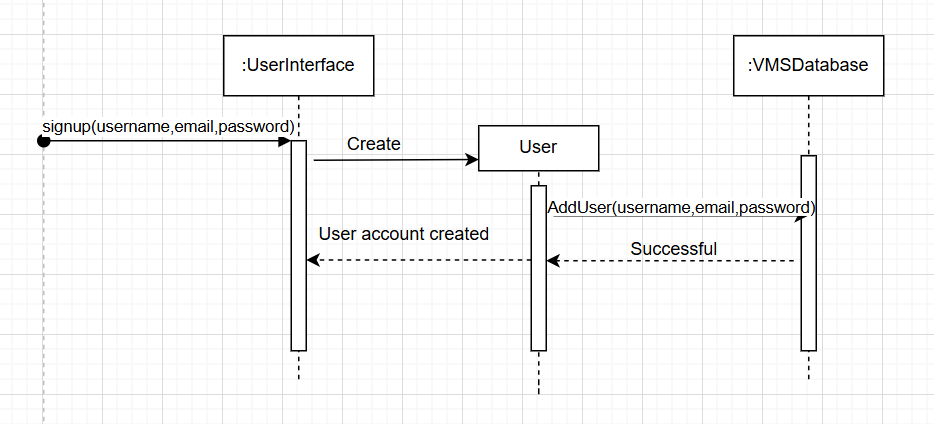
* The user must have a registered email.
* The "Forgot Password" option must be available.

**Post-Conditions:**

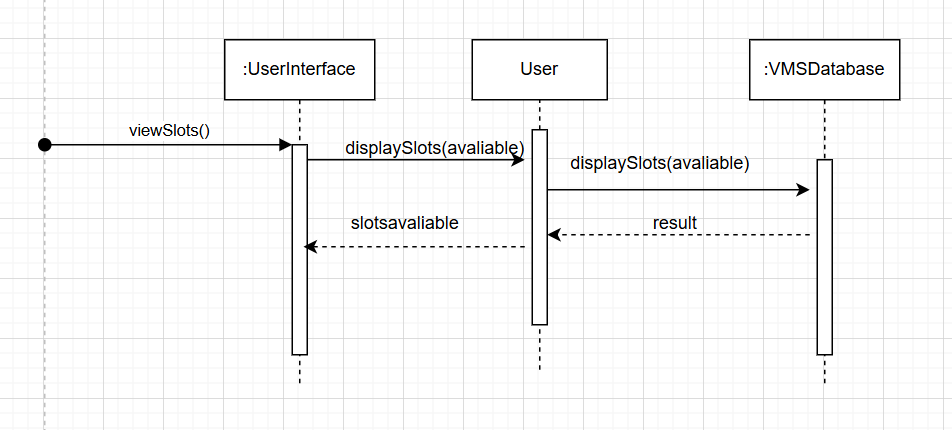
* The user receives a password reset link.
* The new password must meet security requirements before updating.

1. **Sequence Diagram**

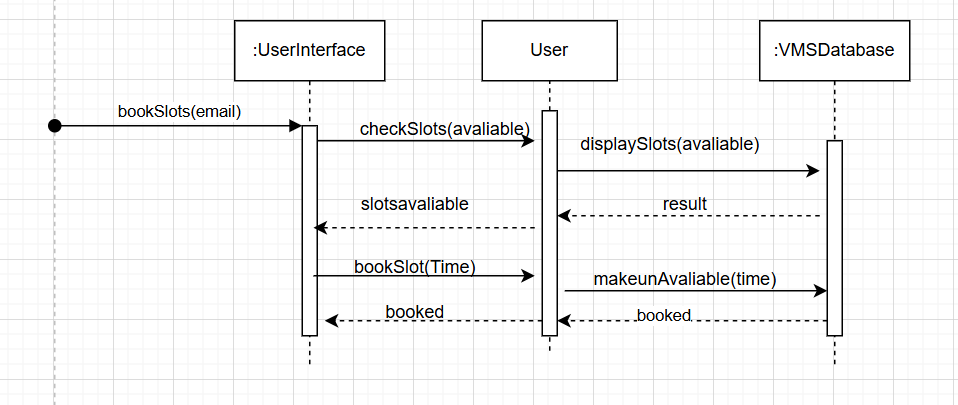
**1)**



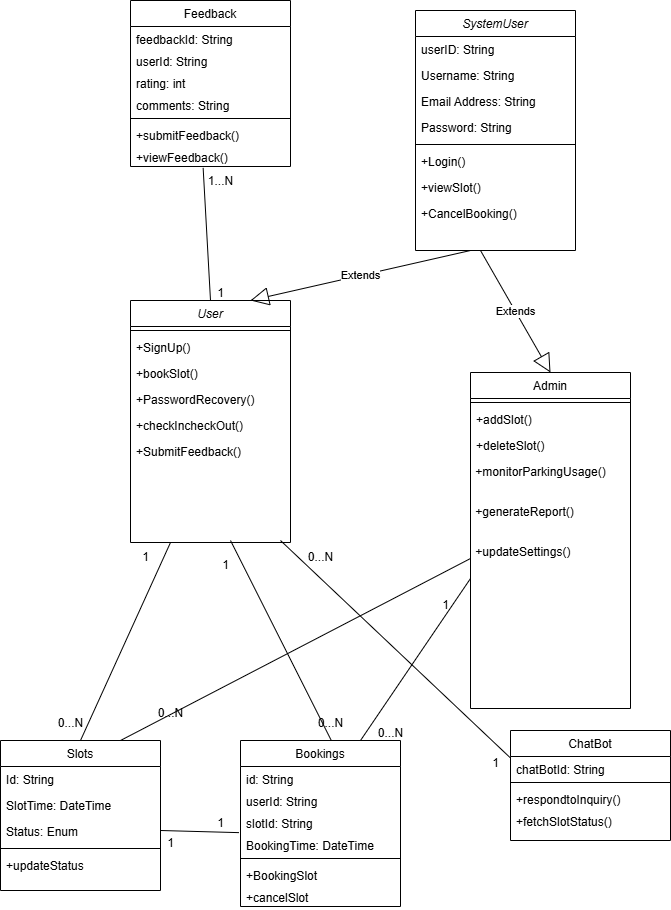
**2)**



**3)**



1. **Class Diagram**



B- Product Backlog

### ****User Stories with Priority Levels****

### ****1. User Registration (High Priority)****

**User Story:**  
As a vehicle owner, I want to register in the system, so that I can access parking services.

**Acceptance Criteria:**

* Users must be able to enter name, email, password, phone number, and vehicle details.
* Email format must be validated and checked for uniqueness.
* Password must meet security requirements (at least 8 characters, one special character, etc.).

### ****2. User Login (High Priority)****

**User Story:**  
As a registered user, I want to log in securely, so that I can manage my parking sessions.

**Acceptance Criteria:**

* Users must be able to log in using their email and password.
* Incorrect credentials must display an error message.
* Upon successful login, users must be redirected to the dashboard.

### ****3. Admin Management (Medium Priority)****

**User Story:**  
As an admin, I want to view and manage registered users, so that I can maintain system security.

**Acceptance Criteria:**

* Admins must be able to log in with admin credentials.
* Admins must be able to view a list of all registered users.
* Admins must be able to delete users from the system.

### ****4. Password Recovery (Medium Priority)****

**User Story:**  
As a user, I want to reset my password, so that I can regain access to my account if I forget it.

**Acceptance Criteria:**

* "Forgot Password" option must be available on the login page.
* The new password must meet security requirements before updating.

C- Sprint Backlog

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | |  |  |  | | --- | | **User Story** | | |  | | --- | |  |  |  | | --- | | **Priority** | | |  | | --- | |  |  |  | | --- | | **Task Breakdown** | | |  | | --- | |  |  |  | | --- | | **Status** | |
| User Registration | |  | | --- | | High |  |  | | --- | |  | | - Create registration form  - Implement email validation  - Implement password security rules  - Send confirmation email/SMS | Done |
| User Login | High | - Implement login page  - Validate credentials  - Error handling for wrong passwords  - Redirect to dashboard on success | Done |
| Admin Management | Medium | |  | | --- | |  |  |  | | --- | | - Create admin login page  - Display registered users  - Implement delete/reset password functionality | | Done |

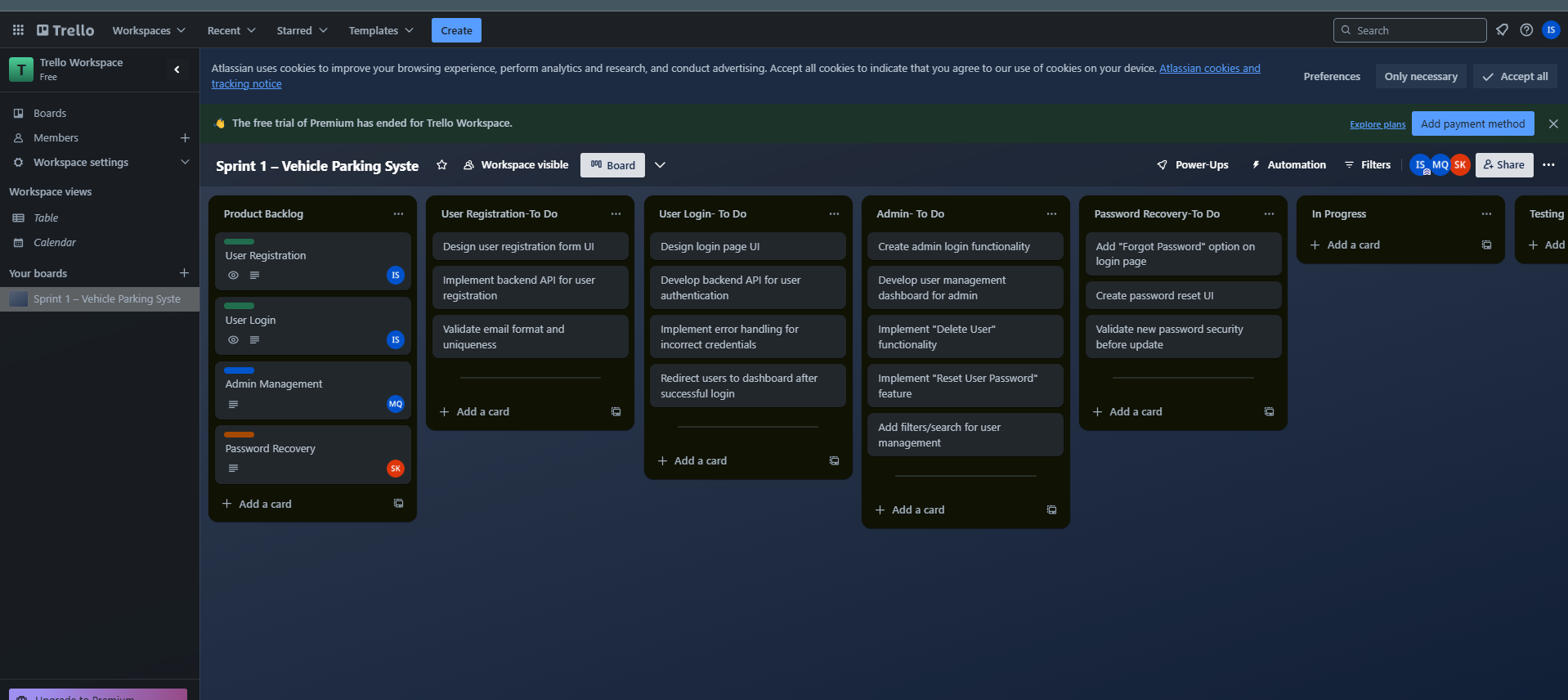
### 

### MY Subset Sprint2:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **User Story** | |  | | --- | |  |  |  | | --- | | **Priority** | | Task Breakdown | |  | | --- | |  |  |  | | --- | | **Status** | |
| Password Recovery | Medium | - Add "Forgot Password" option  - Validate new password security | To Do |

E- Trello Board

My S1



**Sprint 2 - Vehicle Parking Management System**

**1. Sprint 2 Overview**

**Sprint Goal:** Implement Parking Slot Management and Vehicle Check-in/Check-out System.

**Scope of Sprint 2**

1. **Parking Slot Management**
   * View available parking slots
   * Book a parking slot
   * Cancel a booking
2. **Vehicle Check-in & Check-out System**
   * Log vehicle entries and exits
   * Calculate parking duration and fees
3. **Admin Controls**
   * Monitor parking slot usage
   * Override bookings when necessary

**2. User Stories**

Each story follows the format: *"As a [role], I want to [action], so that [benefit]."*

1. **View Parking Slots**  
   *As a user, I want to view available parking slots, so that I can choose a convenient one.*
2. **Book a Parking Slot**  
   *As a user, I want to reserve a parking slot, so that I have a guaranteed parking space.*
3. **Cancel Booking**  
   *As a user, I want to cancel my parking reservation, so that I can free up the slot for others.*
4. **Vehicle Check-in**  
   *As a user, I want to check in my vehicle, so that the system logs my entry time.*
5. **Vehicle Check-out**  
   *As a user, I want to check out my vehicle, so that the system logs my departure and calculates my fee.*
6. **Admin Monitoring**  
   *As an admin, I want to track parking slot usage, so that I can manage availability efficiently.*
7. **Structured Specifications**

**View Parking Slots**

|  |  |
| --- | --- |
| **ID** | **Requirement Description** |
| VPS-1 | The system must display a list of available parking slots with relevant details (e.g., location, price, availability status). |
| VPS-2 | Users must be able to filter parking slots based on location, price, and availability. |
| VPS-3 | The system should update slot availability when a user books or frees a slot. |

**Book Parking Slot**

|  |  |
| --- | --- |
| **ID** | **Requirement Description** |
| BPS-1 | Users must be able to select an available parking slot and confirm the booking. |
| BPS-2 | The system should update the slot status to 'reserved' after booking. |
| BPS-3 | A confirmation message should be displayed after successful booking. |

**Cancel Booking**

|  |  |
| --- | --- |
| **ID** | **Requirement Description** |
| CB-1 | Users must be able to cancel their active parking slot reservation. |
| CB-2 | The system should update the slot status to 'available' upon cancellation. |
| CB-3 | A confirmation message should be displayed after successful cancellation. |

**Vehicle Check-In**

|  |  |
| --- | --- |
| **ID** | **Requirement Description** |
| VCI-1 | Users must be able to check in their vehicle upon arrival. |
| VCI-2 | The system should log the check-in time. |
| VCI-3 | A confirmation message should be displayed after a successful check-in. |

**Vehicle Check-Out**

|  |  |
| --- | --- |
| **ID** | **Requirement Description** |
| VCO-1 | Users must be able to check out their vehicle upon departure. |
| VCO-2 | The system should log the check-out time and calculate parking fees. |
| VCO-3 | A confirmation message should be displayed after a successful check-out. |

**Admin Monitoring**

|  |  |
| --- | --- |
| **ID** | **Requirement Description** |
| AM-1 | Admins must be able to view parking slot usage data. |
| AM-2 | The system should provide real-time updates on slot availability. |
| AM-3 | Admins must be able to see reports on parking trends. |

1. **System Functional Requirements**

|  |  |
| --- | --- |
| ID | Requirement Description |
| FR-10 | Users must be able to view available parking slots with availability status. |
| FR-11 | Users must be able to filter parking slots based on location. |
| FR-12 | Users must be able to reserve a parking slot. |
| FR-13 | The system must update the slot status upon booking. |
| FR-14 | Users must be able to cancel their reservation, updating the slot status accordingly. |
| FR-15 | Users must be able to check in their vehicle, and the system should log the check-in time |
| FR-16 | Users must be able to check out their vehicle, and the system should log the check-out time and calculate the parking fee. |
| FR-17 | Admins must be able to track parking slot usage and view real-time slot availability. |
| FR-18 | The system must provide admins with reports on parking trends. |

1. **System Non Functional Requirements**

### **Product Requirements**

• The system should update parking slot availability dynamically when booked, canceled, or freed.

• The system should provide a simple and intuitive interface for users to book, cancel, check in, and check out.

• The system should support real-time updates for parking slot monitoring by admins.

### **Organizational Requirements**

• The system should ensure data integrity by properly managing slot status changes.

• User check-in and check-out logs should be stored securely for future reference.

### **External Requirements**

• The parking fee calculation should align with predefined pricing rules.

• The system should ensure secure access for admin monitoring functions.

1. **User Stories with Pre/Post Conditions**

**1) View Parking Slots**  
*As a user, I want to view available parking slots, so that I can choose a convenient one.*

**Pre-Conditions:**

• The user is logged into the system.

• Parking slots exist in the system database.

**Post-Conditions:**

• The user can see available parking slots with details.

• The system updates the slot availability accordingly.

**2) Book Parking Slot**  
*As a user, I want to reserve a parking slot, so that I have a guaranteed parking space.*

**Pre-Conditions:**

• The user is logged in.

• The selected parking slot is available.

**Post-Conditions:**

• The system marks the slot as reserved.

• The user receives a booking confirmation.

**3) Cancel Booking**  
*As a user, I want to cancel my parking reservation, so that I can free up the slot for others*

**Pre-Conditions:**

• The user is logged in.

• The user has an active parking reservation.

**Post-Conditions:**

• The system updates the slot status to 'available'.

• The user receives a cancellation confirmation.

**4) Vehicle Check-in**  
*As a user, I want to check in my vehicle, so that the system logs my entry time.*

**Pre-Conditions:**

• The user has an active parking reservation.

• The user arrives at the parking location.

**Post-Conditions:**

• The system logs the check-in time.

• The user receives a check-in confirmation.

**5) Vehicle Check-Out**  
*As a user, I want to check out my vehicle, so that the system logs my departure and calculates my fee.*

**Pre-Conditions:**

• The user has checked in their vehicle.

• The user is leaving the parking location.

**Post-Conditions:**

• The system logs the check-out time.

• The system calculates the parking fee.

• The user receives a check-out confirmation.

**6) Admin Monitoring**  
*As an admin, I want to track parking slot usage, so that I can manage availability efficiently.*

**Pre-Conditions:**

• The user is logged in as an admin.

• Parking data is available in the system.

**Post-Conditions:**

• The admin can monitor parking slot usage.

• The system provides updated slot availability information.

B- Product Backlog

**1) View Parking Slots(**High)  
*As a user, I want to view available parking slots, so that I can choose a convenient one.*

• Users must be able to view available parking slots with location and availability status

• Users must be able to filter parking slots based on location.

• The system must update availability when a slot is booked or freed.

**2) Book Parking Slot(**High)  
*As a user, I want to reserve a parking slot, so that I have a guaranteed parking space.*

• Users must be able to reserve a parking slot.

• The system should update the slot status upon booking.

• A confirmation message must be shown.

**3) Cancel Booking(Medium)**  
*As a user, I want to cancel my parking reservation, so that I can free up the slot for others*

• Users must be able to cancel their reservation.

• The system should update the slot status upon cancellation.

• A confirmation message must be shown.

**4) Vehicle Check-In (**High)  
*As a user, I want to check in my vehicle, so that the system logs my entry time.*

• Users must be able to check in their vehicle.

• The system should log the check-in time.

• A confirmation message must be shown.

**5) Vehicle Check-Out(**High)  
*As a user, I want to check out my vehicle, so that the system logs my departure and calculates my fee.*

• Users must be able to check out their vehicle.

• The system should log the check-out time and calculate the fee.

• A confirmation message must be shown.

**6) Admin Monitoring (**Medium)  
*As an admin, I want to track parking slot usage, so that I can manage availability efficiently.*

• Admins must be able to track parking slot usage.

• The system should provide real-time updates on availability.

• Admins must be able to access parking trend reports.

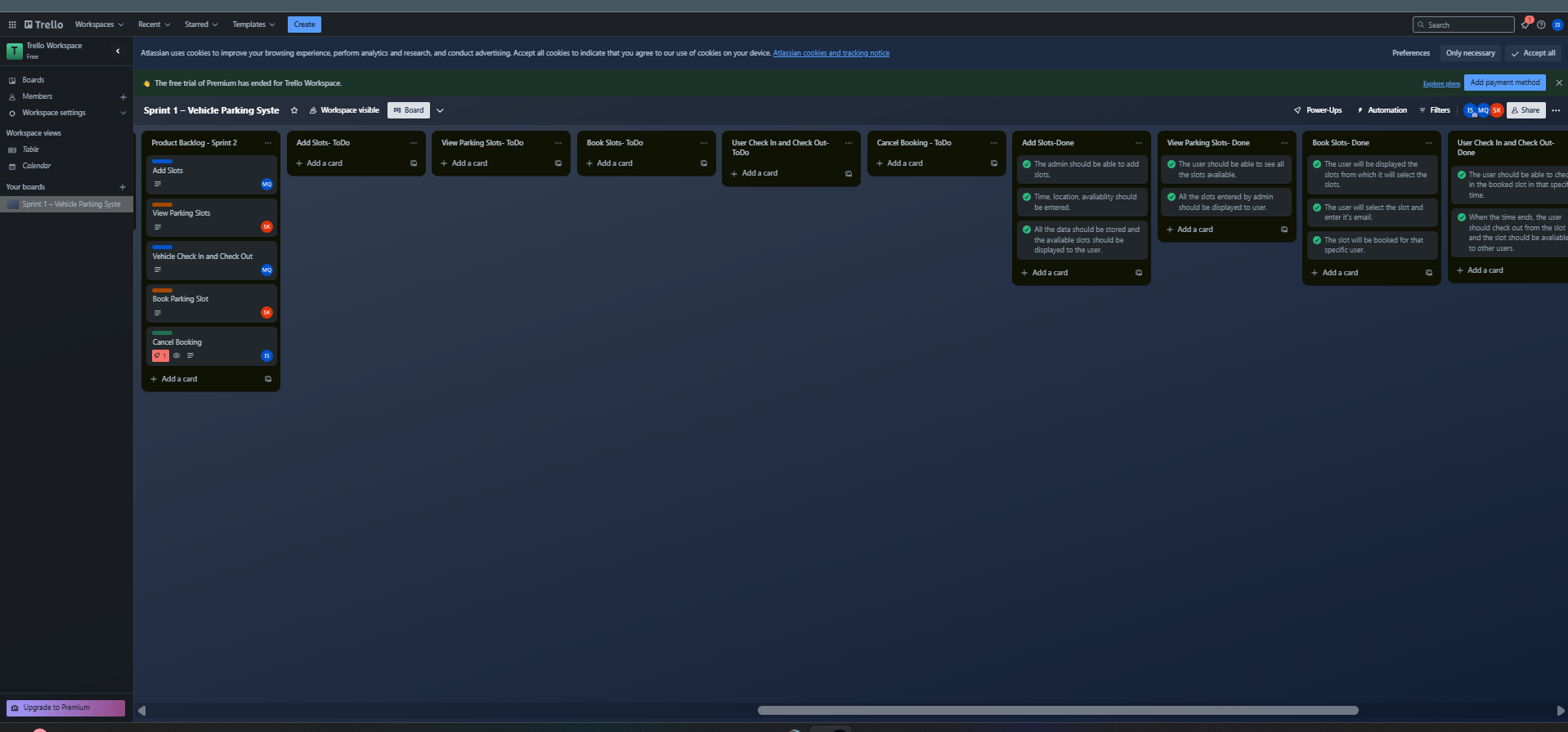
**C. Sprint Backlog**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **User Story** |  |  |  |  |  |  |  |  |  |  | **Priority** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **Task Breakdown** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **Status** |
| View Parking Slots |  |  |  |  |  |  |  |  |  |  | High |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | - Implement parking slot listing  - Add filtering options (location)  - Update slot availability dynamically |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Done |
| Book a Parking Slot |  |  |  |  |  |  |  |  |  |  | High |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | - Implement slot selection feature  - Update slot status upon booking  - Display booking confirmation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Done |
| Cancel Booking |  |  |  |  |  |  |  |  |  |  | Medium |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | - Implement cancel reservation feature  - Update slot status to available  - Display cancellation confirmation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Done |
| Vehicle Check-in |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | - Implement check-in functionality  - Log check-in time  - Display check-in confirmation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Done |
| Vehicle Check-out |  |  |  |  |  |  |  |  |  |  | High |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | - Implement check-out functionality  - Log check-out time  - Calculate parking fee  - Display check-out confirmation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Done |
| Admin Monitoring |  |  |  |  |  |  |  |  |  |  | Medium |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | - Implement parking usage tracking  - Display real-time slot availability  - Generate parking trend reports |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Done |

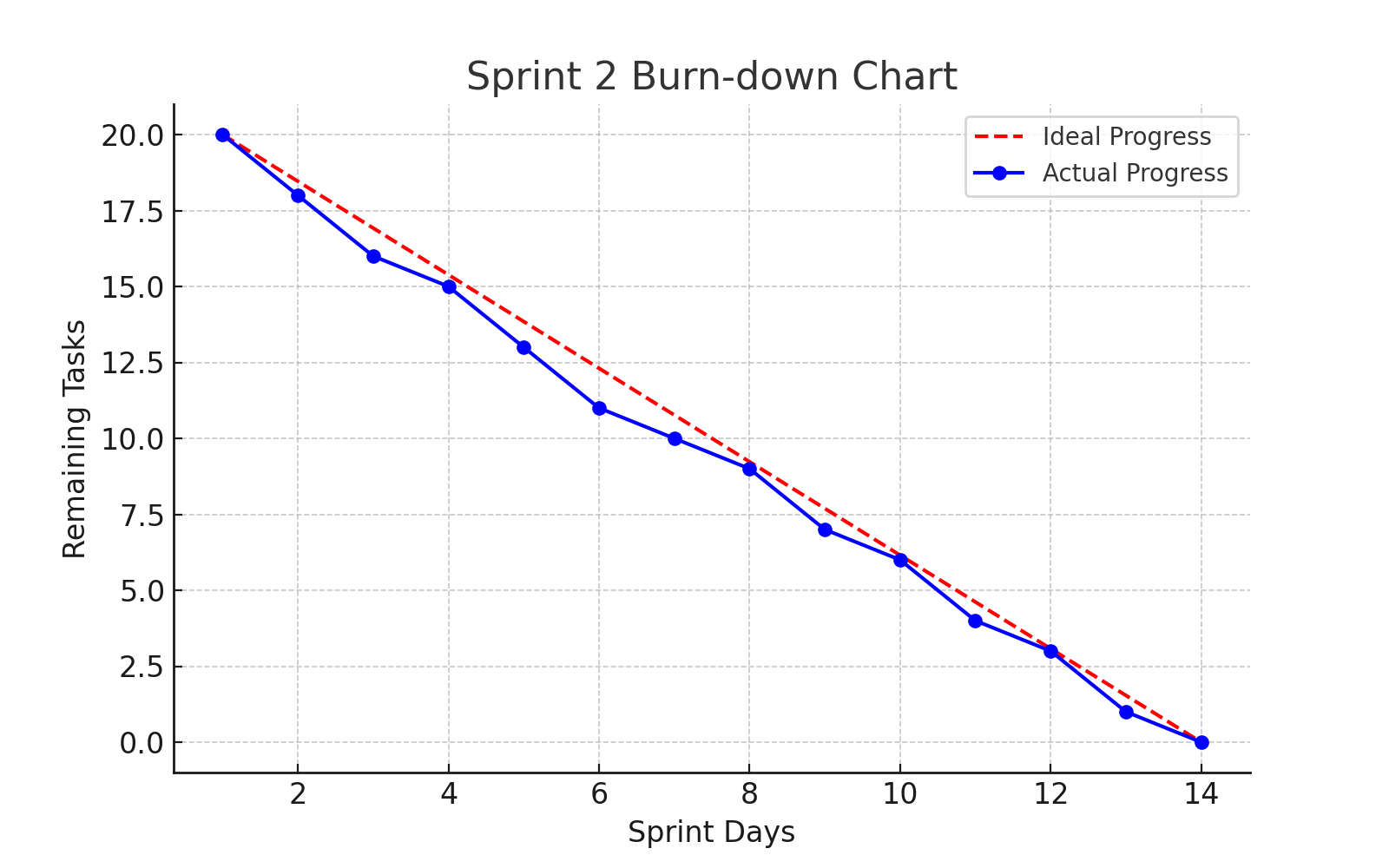
D. Version Control

<https://github.com/hammadshabbir10/Vehicle-Management-System>

E- Trello Board

****

**11. Burn-down Chart**



**Sprint 3**

### 1. Overview

**Goal:** Implement a Chat-bot, Delete Slot, Analytics, Settings Management, and Feedback System.

**Scope:**

1. **Chat-bot**
   * Implement a chat-bot for user inquiries (e.g., slot availability, parking status).
2. **Delete Slot**
   * Enable deletion of parking slots.
   * Ensure validation and error handling during slot deletion.
3. **Analytics**
   * Display parking slot usage statistics (e.g., popular slots, peak times).
   * Provide visualizations (charts/graphs) for users and admins.
4. **Settings**
   * Admin portal to manage settings: change portal name, admin email, and password.
   * Implement secure password change functionality with validation.
5. **Feedback**
   * Add a feedback system to collect user reviews after parking sessions.
   * Allow users to rate their experience and provide comments.

**2. User Stories**

Each story follows the format: *"As a [role], I want to [action], so that [benefit]."*

 **Chat-bot Interaction**

* As a user, I want to interact with a chat-bot, so that I can easily get information about parking availability and status.

 **Delete Slot**

* As an admin, I want to delete parking slots, so that I can manage the parking system effectively and remove unused slots.

 **View Analytics**

* As an admin, I want to view parking slot usage statistics, so that I can make informed decisions about parking space allocation.

 **Manage Settings**

* As an admin, I want to update the portal name, admin email, and change my password, so that I can maintain system configuration and security.

 **Provide Feedback**

* As a user, I want to provide feedback about my parking experience, so that I can help improve the system for future users.

**3. Structured Specifications**

**Delete Slots**

|  |  |
| --- | --- |
| **ID** | **Requirement Description** |
| DS-1 | The system must allow admins to view a list of existing parking slots. |
| DS-2 | The system must allow admins to select a parking slot for deletion. |
| DS-3 | The system must confirm the deletion action with a confirmation prompt. |
| DS-4 | Once confirmed, the selected parking slot must be permanently deleted from the system. |
| DS-5 | The system must notify the admin with a success or failure message upon deletion. |

**Chat-bot Interaction**

|  |  |
| --- | --- |
| **ID** | **Requirement Description** |
| CB-1 | The system must implement a chat-bot capable of handling user inquiries related to parking services. |
| CB-2 | The chat-bot must respond to queries such as slot availability, parking status, and other general inquiries. |
| CB-3 | The chat-bot must be accessible through the user interface (UI) and be responsive in real-time. |
| CB-4 | The chat-bot should be able to provide relevant responses based on user inputs. |
| CB-5 | The system should support natural language understanding for better interaction. |

**View Analytics**

|  |  |
| --- | --- |
| **ID** | **Requirement Description** |
| AN-1 | The system must provide parking slot usage statistics, such as popular slots and peak times. |
| AN-2 | The system must generate and display visualizations (charts/graphs) for users and admins. |
| AN-3 | The system must update statistics and visualizations in real-time or periodically. |
| AN-4 | The visualizations must be user-friendly and easy to understand. |
| AN-5 | Admins should be able to view detailed analytics on parking slot usage and patterns. |

**Manage Settings**

|  |  |
| --- | --- |
| **ID** | **Requirement Description** |
| ST-1 | The system must provide an admin portal for managing settings, such as changing portal name, admin email, and password. |
| ST-2 | The system must allow the admin to securely change the password with proper validation (e.g., minimum length, special character). |
| ST-3 | The system must verify the admin’s identity before allowing sensitive settings modifications (e.g., password change). |
| ST-4 | The system must provide a confirmation prompt for changes made in settings. |
| ST-5 | The system must notify the admin upon successful or failed changes to settings. |

**1.System Functional Requirements**

|  |  |
| --- | --- |
| **ID** | **Requirement Description** |
| FR-19 | The system must provide a chat-bot for users to inquire about parking slot availability and parking status. |
| FR-20 | The chat-bot must provide real-time responses based on user input and available data. |
| FR-21 | The system must allow users and admins to access parking slot usage statistics. |
| FR-22 | The system must generate and display visualizations (charts/graphs) of parking trends for users and admins. |
| FR-23 | The system must allow admins to change portal settings (e.g., name, email, and password) securely. |
| FR-24 | Admins must be able to securely update their password with validation. |
| FR-25 | Users must be able to submit feedback after their parking session, including a rating and comment. |
| FR-26 | The system must store feedback data securely and provide an analytics dashboard for feedback analysis. |
| FR-27 | The system must ensure that feedback submission is anonymous, ensuring user privacy. |
| FR-28 | Admins must be able to view feedback data to assess user satisfaction. |

**2.System Non-Functional Requirements**

**Product Requirements**

|  |  |
| --- | --- |
| **ID** | **Requirement Description** |
| NFR-1 | The system should ensure that chat-bot responses are fast and efficient, with minimal latency. |
| NFR-2 | The system should support real-time updates of parking trends and feedback data. |
| NFR-3 | Visualizations (charts/graphs) should load and render promptly, even with large datasets. |
| NFR-4 | The system should be intuitive for both users and admins when interacting with the chat-bot, viewing analytics, and managing settings. |
| NFR-5 | Feedback submission and viewing should be seamless and fast for users and admins. |
| NFR-6 | The system must provide secure password management and data protection for user and admin accounts. |

**Organizational Requirements**

|  |  |
| --- | --- |
| **ID** | **Requirement Description** |
| NFR-7 | The system should ensure data integrity and consistency when updating feedback, settings, and analytics data. |
| NFR-8 | Feedback data should be stored securely and be accessible only by authorized personnel (admins). |
| NFR-9 | The system must store user and admin interactions (e.g., feedback submissions, settings changes) for future reference and analytics. |

**External Requirements**

|  |  |
| --- | --- |
| **ID** | **Requirement Description** |
| NFR-10 | The system must follow secure authentication protocols to protect admin functionalities and user data. |
| NFR-11 | The system should comply with relevant data privacy regulations for storing user feedback and other sensitive information (e.g., GDPR). |
| NFR-12 | Feedback data must be processed and stored in accordance with predefined rules for secure handling of user-generated content. |

**3.User Stories with Pre and Post Conditions**

### 1. Chat-bot

**User Story:** As a user, I want to interact with a chat-bot to inquire about parking slot availability and parking status, so that I can make informed decisions quickly.

**Pre-Conditions:**

* The user is logged into the system.
* The chat-bot is active and integrated into the system.
* Parking slot data is available and up-to-date.

**Post-Conditions:**

* The user can ask questions to the chat-bot and receive relevant responses regarding parking availability and status.
* The system provides real-time responses based on current data.
* The chat-bot records interactions for future reference.

### 2. Analytics

**User Story:** As an admin, I want to view parking slot usage statistics and trends, so that I can manage parking resources effectively.

**Pre-Conditions:**

* The admin is logged into the system.
* The system has collected enough parking slot usage data.
* Analytics and visualization features are enabled in the system.

**Post-Conditions:**

* The admin can view real-time parking slot usage statistics.
* The system displays relevant visualizations (e.g., charts/graphs) of parking trends.
* Admins can use the data to analyze and make informed decisions.

### 3. Settings Management (Admin Portal)

**User Story:** As an admin, I want to manage portal settings, such as changing the portal name, admin email, and password, so that I can ensure proper configuration of the system.

**Pre-Conditions:**

* The admin is logged into the system with necessary permissions.
* The admin portal is accessible with settings management options enabled.

**Post-Conditions:**

* The admin can successfully update the portal name, admin email, and password.
* The system validates password changes and confirms successful updates.
* Any changes made are reflected immediately across the system.

### 4. Feedback System

**User Story:** As a user, I want to submit feedback after my parking session, including a rating and comments, so that I can share my experience with the system.

**Pre-Conditions:**

* The user has completed a parking session.
* The user is logged into the system and can access the feedback form.

**Post-Conditions:**

* The user successfully submits feedback, including a rating and optional comments.
* The feedback is stored securely in the system for future analysis.
* The system acknowledges the feedback submission and thanks the user.

### 5. Delete Slot

**User Story:** As an admin, I want to delete parking slots, so that I can manage the parking system effectively and remove unused slots.

**Pre-Conditions:**

* The admin is logged into the system with the necessary permissions to delete slots.
* The parking slot(s) to be deleted exist in the system.
* The admin has access to the list of available parking slots.

**Post-Conditions:**

* The selected parking slot is successfully deleted from the system.
* The system updates the availability status of parking slots accordingly.
* The admin receives a confirmation message indicating the success or failure of the deletion.

B- Product Backlog

### 1) ****Chat-bot (High)****

As a user, I want to interact with a chat-bot to inquire about parking slot availability and parking status, so that I can make informed decisions quickly.

* Users must be able to ask questions about parking slot availability and get real-time responses.
* The chat-bot should use natural language processing to handle user inquiries.
* The system should integrate with the backend to retrieve parking slot availability information.
* The chat-bot must be available on both the web interface and mobile view.

### 2) ****Analytics (High)****

As an admin, I want to view parking slot usage statistics and trends, so that I can manage parking resources effectively.

* Admins must be able to view detailed parking slot usage statistics (e.g., popular slots, peak times).
* The system must provide visualizations (charts/graphs) of parking trends for admins and users.
* The analytics should update in real-time or periodically based on new data.
* Admins should be able to filter and download detailed reports of parking slot usage.

### 3) ****Settings Management (Admin Portal) (High)****

As an admin, I want to manage portal settings, such as changing the portal name, admin email, and password, so that I can ensure proper configuration of the system.

* Admins must be able to update the portal name, admin email, and password through a secure interface.
* The system must validate the admin’s password changes according to predefined rules.
* Admins must confirm sensitive changes (e.g., password update) before they are applied.
* The system should securely store any updated admin credentials or portal settings.

### 4) ****Feedback System (Medium)****

As a user, I want to submit feedback after my parking session, including a rating and comments, so that I can share my experience with the system.

* Users must be able to rate their parking experience with a numeric rating system (e.g., 1 to 5 stars).
* Users must be able to provide additional comments with their feedback.
* The feedback data must be stored securely for future analysis.
* Admins should be able to access and analyze feedback data (e.g., user satisfaction).
* The system must send a confirmation message after feedback submission.

### 5) ****Delete Slot (High)****

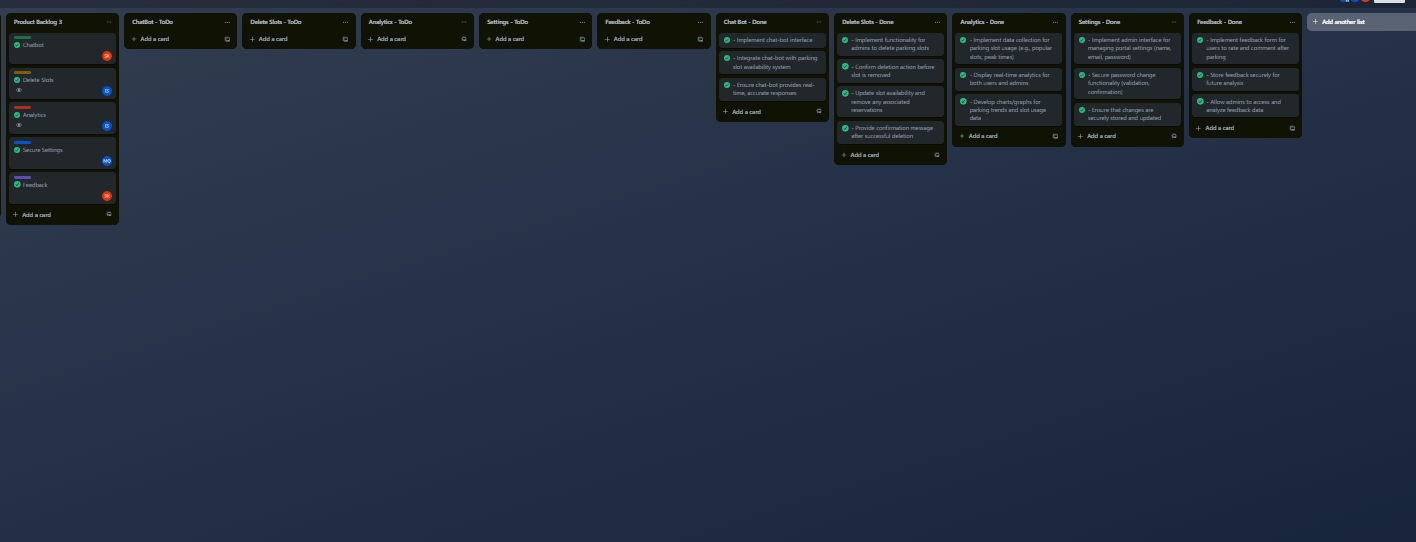
As an admin, I want to delete parking slots, so that I can manage the parking system effectively and remove unused slots.

* Admins must be able to view the list of parking slots and select which ones to delete.
* The system should confirm the deletion action before it is performed.
* Once a slot is deleted, the system must update the availability of slots and remove any reservations associated with the deleted slot.
* Admins must receive feedback on whether the deletion was successful or failed.

B- Sprint Backlog

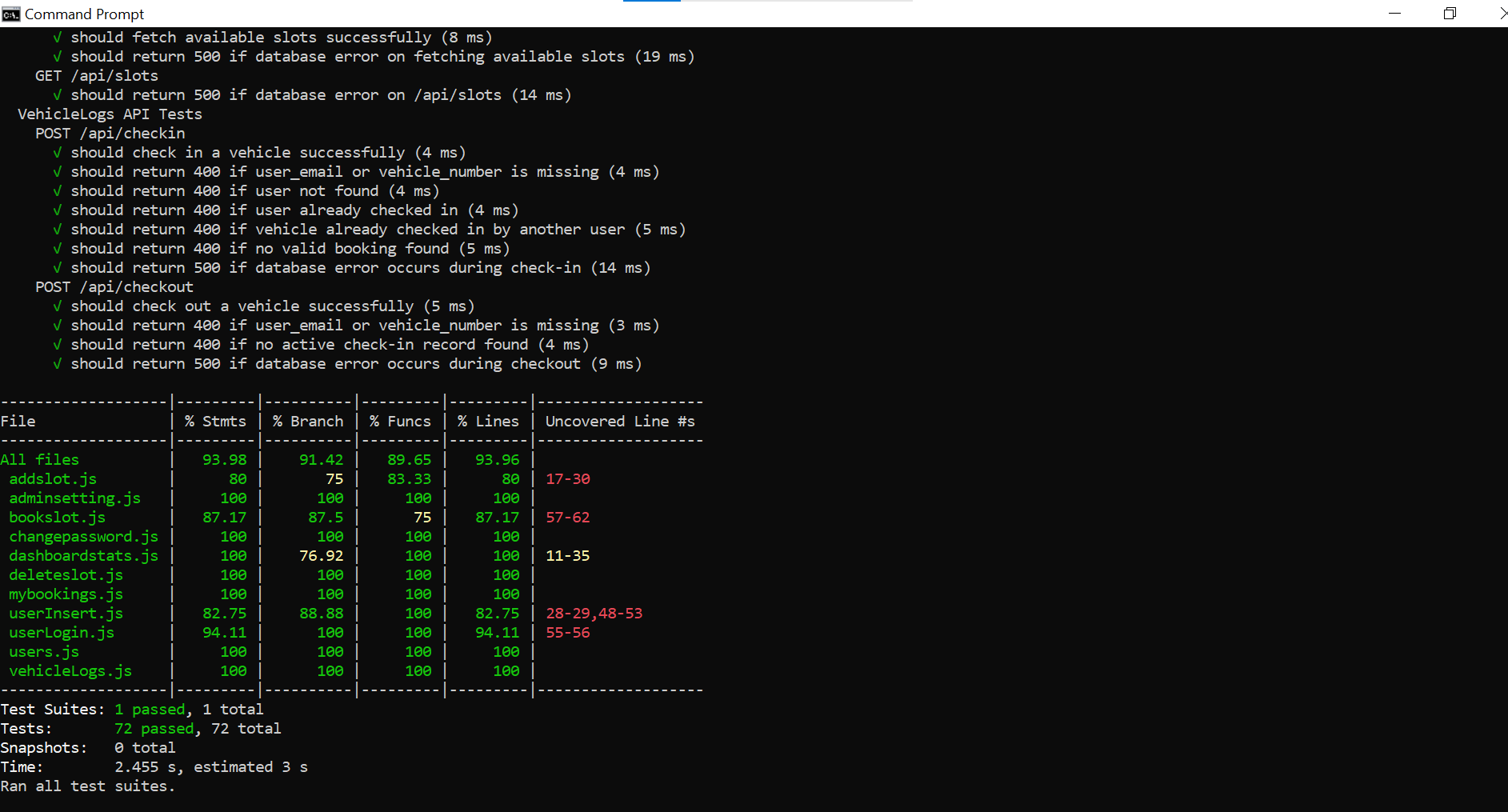
| **User Story** | **Priority** | **Task Breakdown** | **Status** |
| --- | --- | --- | --- |
| **Chat-bot** | High | - Implement chat-bot interface |  |
|  |  | - Integrate chat-bot with parking slot availability system |  |
|  |  | - Ensure chat-bot provides real-time, accurate responses | Done |
|  |  | - Implement natural language processing for user inquiries |  |
| **Analytics** | High | - Implement data collection for parking slot usage (e.g., popular slots, peak times) |  |
|  |  | - Develop charts/graphs for parking trends and slot usage data |  |
|  |  | - Display real-time analytics for both users and admins |  |
| **Settings Management** | High | - Implement admin interface for managing portal settings (name, email, password) | Done |
|  |  | - Secure password change functionality (validation, confirmation) |  |
|  |  | - Ensure that changes are securely stored and updated |  |
| **Feedback System** | Medium | - Implement feedback form for users to rate and comment after parking | Done |
|  |  | - Store feedback securely for future analysis |  |
|  |  | - Allow admins to access and analyze feedback data |  |
| **Delete Slot** | High | - Implement functionality for admins to delete parking slots |  |
|  |  | - Confirm deletion action before slot is removed | Done |
|  |  | - Update slot availability and remove any associated reservations |  |
|  |  | - Provide confirmation message after successful deletion |  |

E- Trello Board

****

**Testing:**

**White Box**



**Summary:**

For white box testing of our Node.js Express-based Vehicle Parking Management System, we used Jest as the testing framework. We wrote unit and integration tests for all major backend modules, focusing on verifying core logic, handling edge cases, and simulating database errors.

**Covered Areas:**

All core routes for vehicle check-in/check-out, slot management, user login/registration, and dashboard statistics are covered. Tests include both positive and negative scenarios, such as:

Missing parameters (400 errors)

Duplicate or conflicting user actions

Simulated database errors (500 errors)

Edge cases and validation logic are tested thoroughly. Each major module (e.g., bookslot.js, vehicleLogs.js, userInsert.js) has coverage above 80%.

**Un-Covered Areas:**

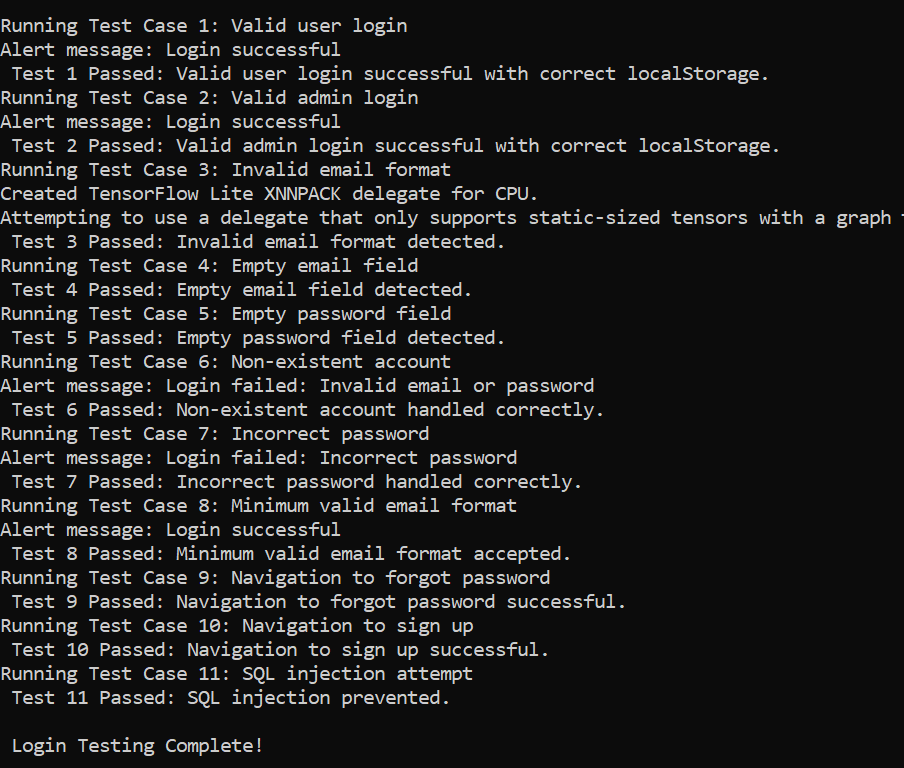
addslot.js (lines 17–30): Due to unreachable paths or admin-specific logic not triggered in tests.

bookslot.js (lines 57–62): Possibly nested error handling or fallback behavior.

UserInsert.js and userLogin.js: Includes error or database fallback code that may not execute unless specific runtime conditions occur.

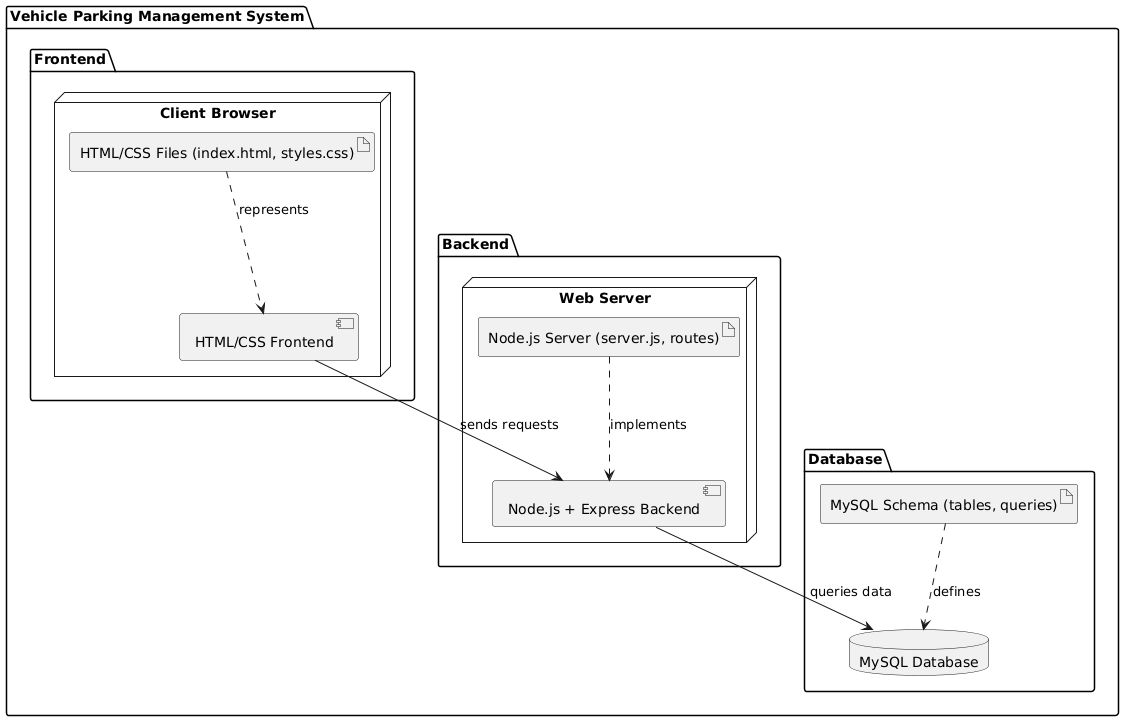
Some minor parts may involve external modules or third-party libraries which are not unit-tested directly.

**Black Box**

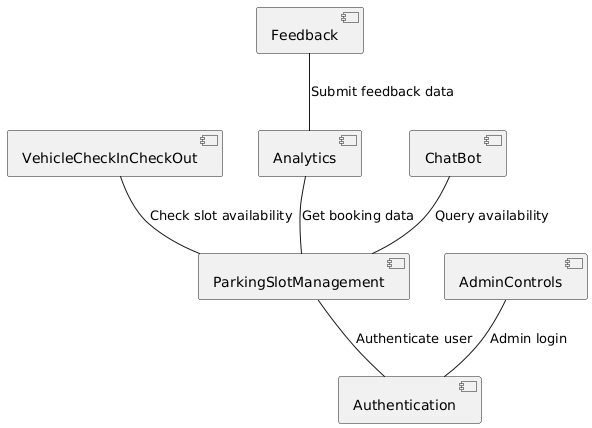
* 

**System Architecture:**

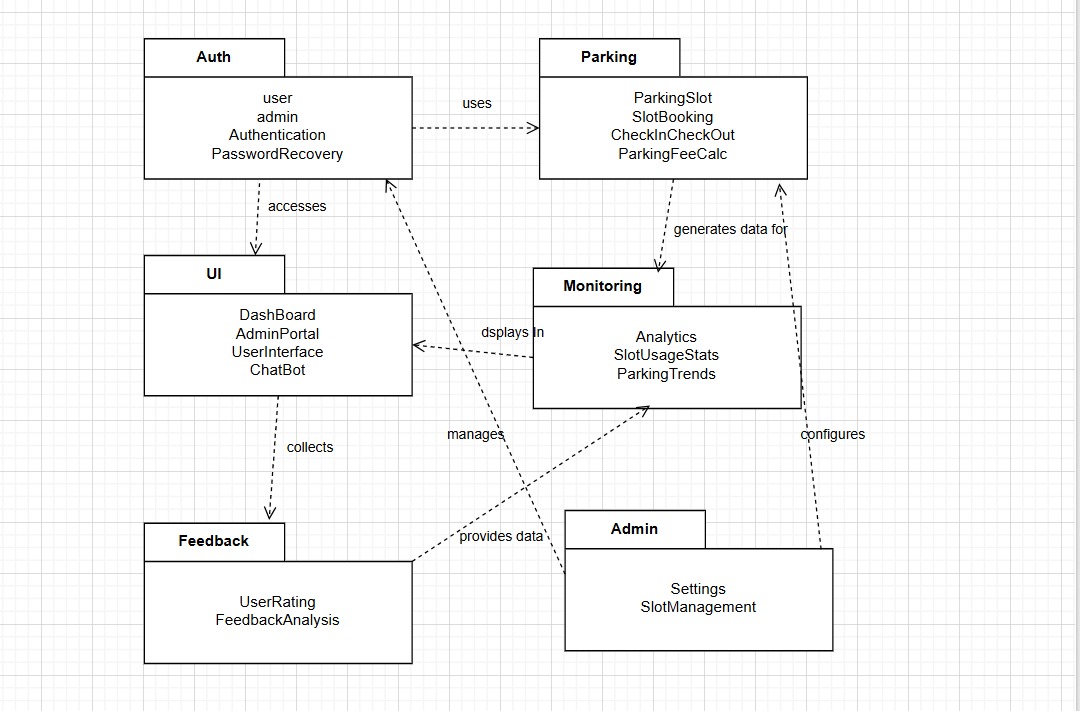
**Deployment:**



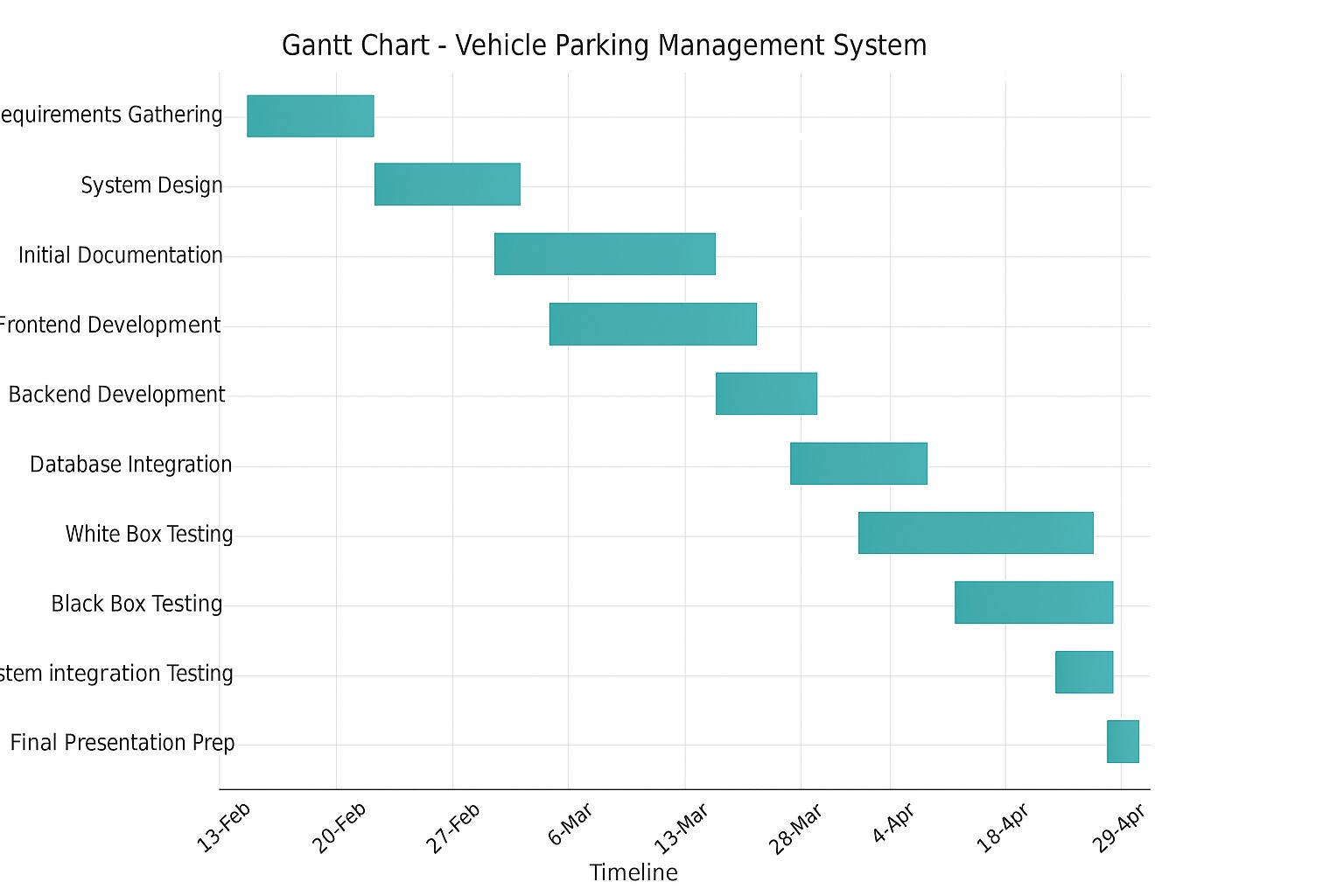
**Component:**



**Package:**

****

**Gantt Chart**



**WBS**

**1. Planning & Setup**

Requirement Gathering

System Design

Tech Stack Selection (HTML, CSS, Node)

Database Selection and Design (MySQL)

**2. User Features**

Sign Up

Login

Book slot

Vehicle Check In and Check Out

Submit Feedback

Change Password

Chat Bot

**3. Admin Dashboard**

View Users

Add slot

View Analytics

Review Feedback

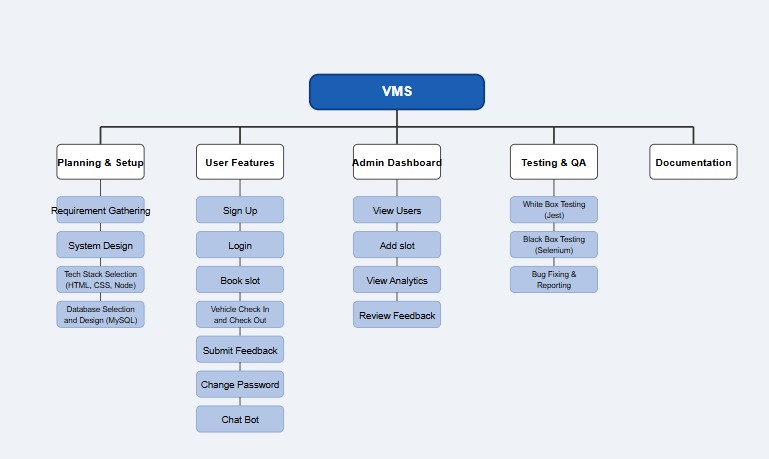
**4. Testing & QA**

White Box Testing (Jest)

Black Box Testing (Selenium)

Bug Fixing & Reporting

**5. Documentation**

****

**Lessons Learned**

Through this project, we learned the value of structured development and teamwork. Designing the Vehicle Parking Management System enhanced our skills in system modeling, backend integration, and user interface design. Dividing work into different sprints helped us manage our work more effectively, while testing phases ensured the system was reliable and user-friendly. Overall, the project gave us hands-on experience in building a real-world application from start to finish.