C-SW312 – Deliverable #3 Report #1

Automated University Garage Management System (AUGMS)

Course: C-SW312: Introduction to Software Engineering

Semester: Fall 2025

Submission Date: November 23, 2025

Group Members: Aly Hassan, Mohamed Ehab, Alaa Shaaban, Kenzy Zedan, Youssef Osama

---

1. Introduction

- Domain class modeling using the Noun Technique
- Validation and refinement of use cases using the CRUD matrix technique
- Behavioral modeling: Activity Diagrams, System Sequence Diagrams, and State Machine Diagrams

All work is built upon Deliverable #2 (Event Decomposition & User Stories) and Deliverable #1 outputs.

I. Domain Classes: Noun Technique

1.1 Step 1 – List of Nouns (from previous deliverables)

Noun / Noun Phrase	Source
Student	Stakeholder document
Faculty / Junior TA	Stakeholder document
User	General to all stakeholders
Vehicle	User Story document
License Plate	Use case document
Personal details	Use case document
Vehicle details	User Story document
Registered records / Vehicle Registration	Use Case doc
Entry gate	Use Case doc
Exit gate	Use Case doc
Parking space / Parking spot	User Story doc
Dashboard	Use Case doc
Service Request (EV charging / car cleaning)	User Story doc
Administrator / Garage Admin	Use Case doc

Occupancy	Events doc
Service statistics / Usage patterns / Reports	Events doc
Log / Log entry / Timestamped log	Use Case doc
Sensor / Sensor malfunction	Use Case doc
Garage status (FULL/AVAILABLE)	Use Case doc

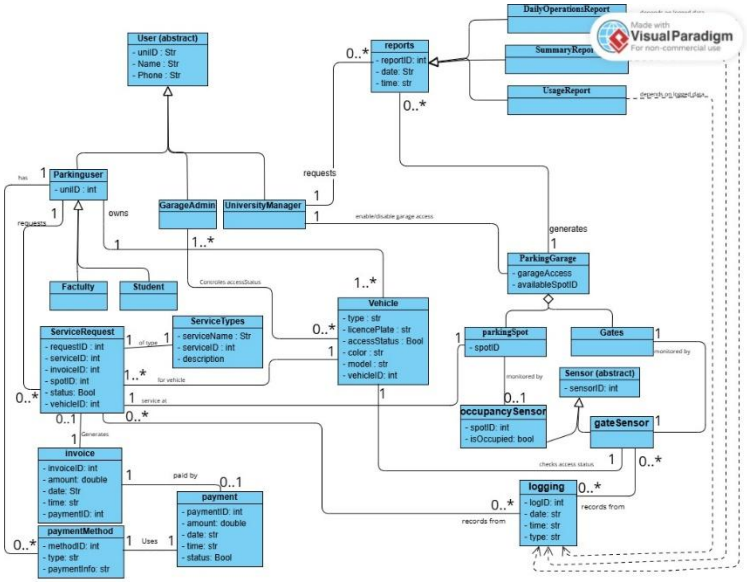
1.2 Step 2 – Classification Table

Noun / Phrase	Type	Reason
User	Abstract Class	generalizes ParkingUser, GarageAdmin, UniversityManager
Student / Faculty	Class (subtype of ParkingUser)	both are Parking Users – share identical parking behavior → generalize
Vehicle	Class	Core entity
Parking spot / Parking space	Class (ParkingSpot)	needed for sensor to determine occupancy
Service (EV charging, car cleaning)	Class (ServiceType)	list of services
Service request	Class (ServiceRequest)	Contains all details regarding the service, the vehicle, invoice and vehicle spot
Invoice	Class	
Payment	Class	
Payment Method	Class	
Administrator / Garage Admin	Class (subtype of User)	Functional class (accepts or rejects pending registration and more functions)
University Management / President	Class (subtype of User)	Functional class (requests reports, have access to close and open the garage)
Log / Log entry	Class (LogEntry)	Keeps history or records. (for reporting and security purposes)
Report / Usage report / Daily/Monthly report	Class (Report with subtypes)	Processes collected data and generates reports
Sensor malfunction / Sensor	Class (Sensor) + (SensorErrorAlert)	Class with function returning car credentials when called – and shows sensor errors
Gate Sensor	Subclass of sensor	Validates entry
License Plate	Attribute	Unique identifier of Vehicle
Personal details	Attribute	Of User
Vehicle details	Attribute	Of Vehicle
Registered records / Vehicle Registration	Class (registrationStatus)	(Pending/Approved/Rejected/Expiration date)

1.3 Step 3 – Final List of Domain Classes:

1. User (abstract)
2. ParkingUser (inherits from User)
3. Student (inherits from ParkingUser)
4. Faculty (inherits from ParkingUser)
5. GarageAdmin (inherits from User)
6. UniversityManager (inherits from User)
7. Vehicle
8. ParkingGarage
9. ParkingSpot
10. Sensor
11. occupancySensor
12. GateSensor
13. ServiceType
14. ServiceRequest
15. Report (abstract)
16. DailyOperationsReport (inherits from Report)
17. SummaryReport (weekly/monthly) (inherits from Report)
18. UsageReport (inherits from Report)
19. Logging

1.4 Step 4 – Graphical domain classes using UML class diagram:



II. Crud Technique: modifying use cases and the use case diagram:

2.1 Step 1 – Identify main entities in the AUGMS Smart Garage System:

1. Vehicle Registration / Vehicle
2. User (Student/Faculty/Staff)
3. Parking Spot / Garage Occupancy
4. Service Request (EV charging, cleaning, etc.)
5. Usage / Entry-Exit Logs
6. Daily/Weekly/Monthly Reports
7. Sensor Status / Alerts
8. User Access/Permissions

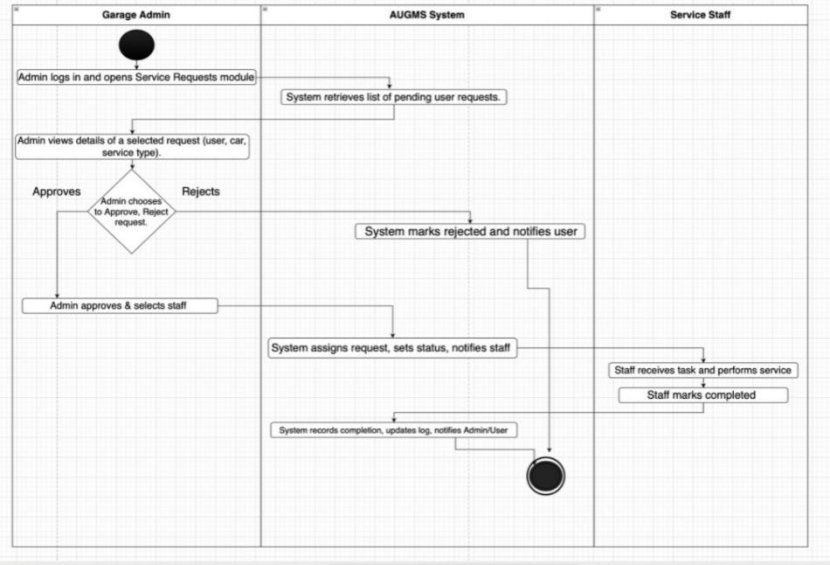
2.2 Step 2 – Mapping existing use cases against CRUD operations:

Data entity / domain class	CRUD	Use case
User (Student/Faculty)	Create	Register User Account (UC-106)
	Read/report	View Parking Status (UC-104) – shows user's vehicle status Generate Usage Report (UC-301)
	Update	Update Vehicle Details (UC-107) Manage User Access – enable/disable (UC-302)
	Delete (archive)	Manage User Access – disable account (UC-302)
Vehicle	Create	Register Vehicle (UC-101)
	Read/report	View Parking Status (UC-104) Monitor Garage Occupancy (UC-202)
	Update	Update Vehicle Details (UC-107) Manage Vehicle Registration – approve/reject (UC-201)
	Delete (archive)	Remove Vehicle Registration (UC-204)
Parking Spot / Occupancy	Create	-
	Read/report	View Parking Status (UC-104) Monitor Garage Occupancy (UC-202)
	Update	Update Garage Status to "full" (UC-501) Update Garage Status to "available" (UC-502)
	Delete (archive)	-
Service Request	Create	Request Service (UC-105)

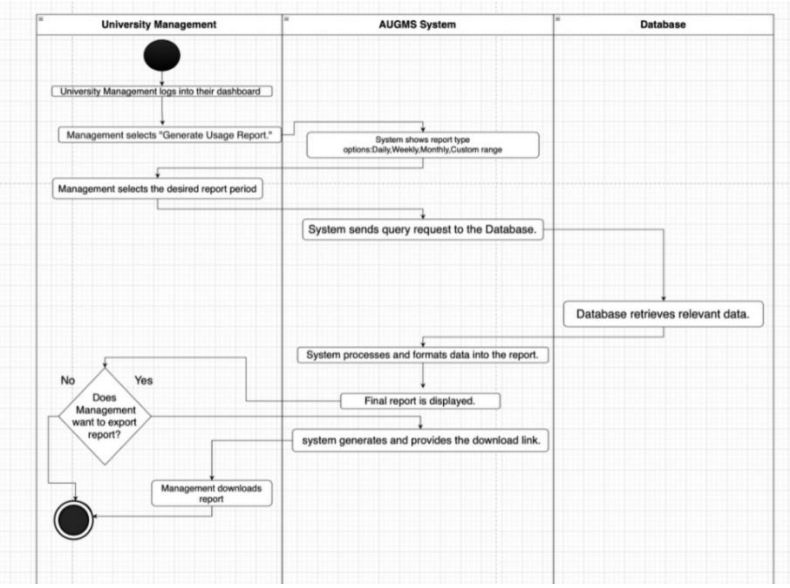
	Read/report	Manage Service Request (UC-203)
	Update	Manage Service Request – track/assign (UC-203) Log Completed Service (UC-403)
	Delete (archive)	Manage Service Request – close/complete (UC-203)
Entry/Exit Log	Create	Validate Vehicle Entry (UC-102) Process Vehicle Exit (UC-103)
	Read/report	Generate Daily Operations Report (UC-401) Generate Usage Report (UC-301)
	Update	-
	Delete (archive)	-
Sensor Alert	Create	Generate Sensor Error Alert (UC-503)
	Read/report	viewed errors by admin/maintenance
	Update	-
	Delete (archive)	-
Reports (Usage, Daily, Summary)	Create	Generate Usage Report (UC-301) Generate Daily Operations Report (UC-401) Generate Weekly/Monthly Summary Report (UC-402)
	Read/report	All reports mentioned above
	Update	on demand
	Delete (archive)	-

III. Activity, System Sequence and state machine diagrams

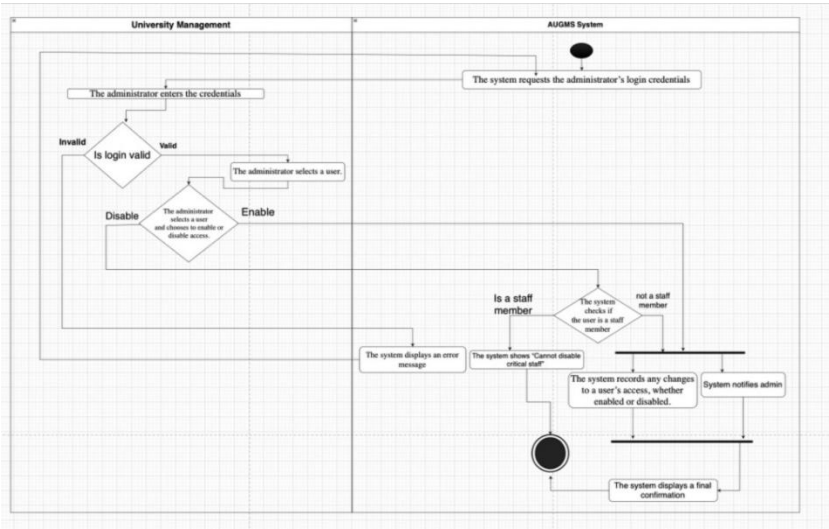
3.1.1 – Activity diagram 1:



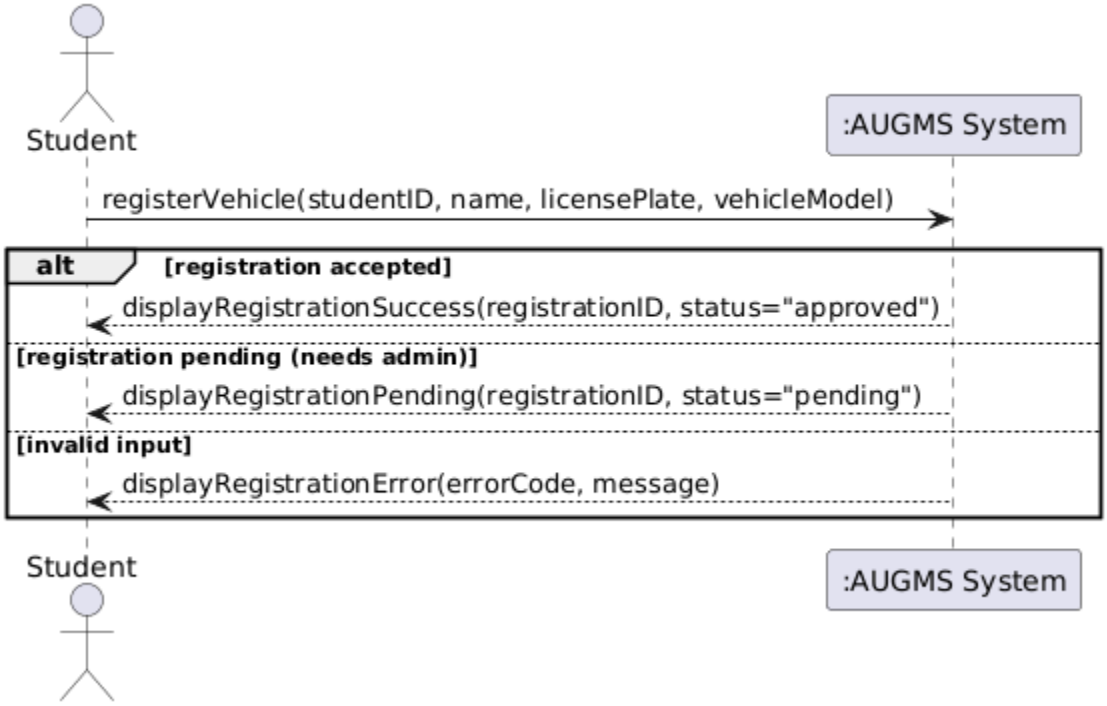
3.1.2 – Activity diagram 2:



3.1.3 – Activity diagram 3:

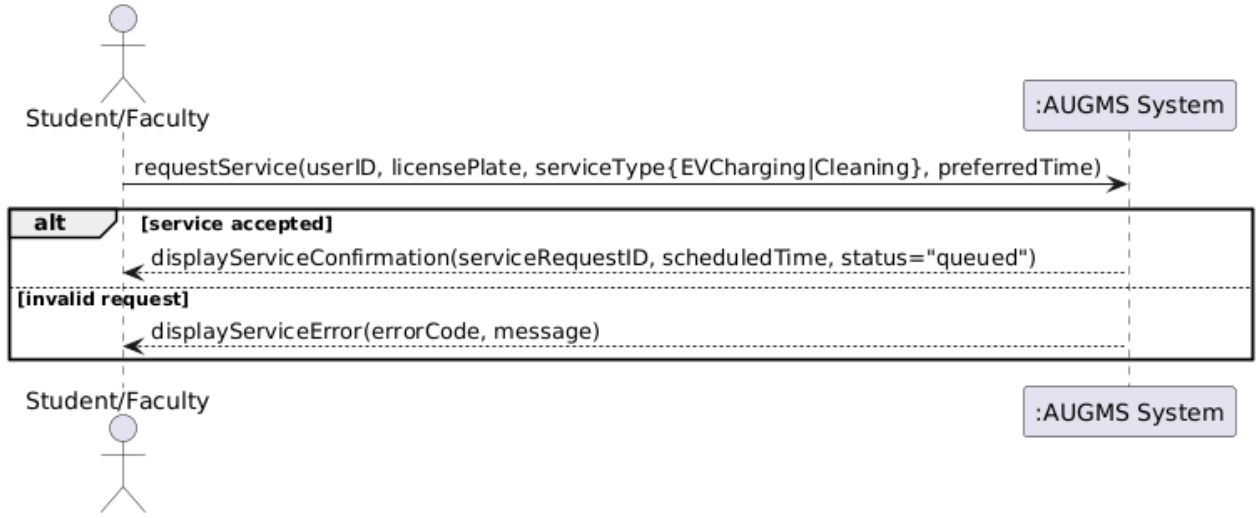


3.2.1 – System Sequence Diagram 1:



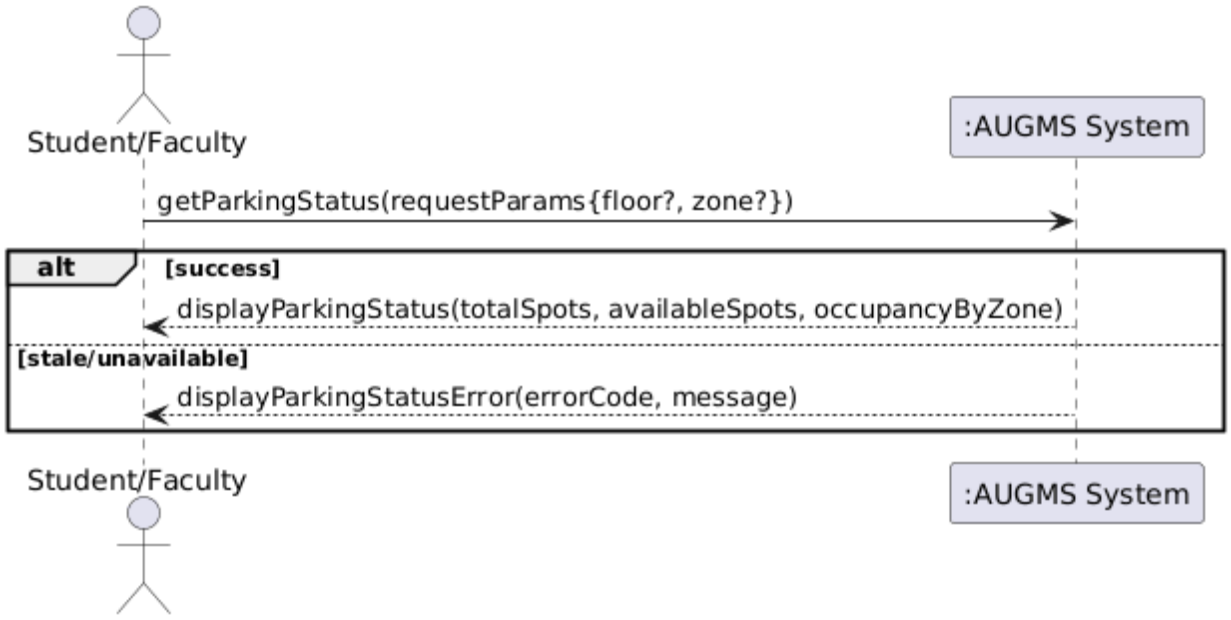
3.2.2 – System Sequence Diagram 2:

3.2.3 – System Sequence Diagram 3:

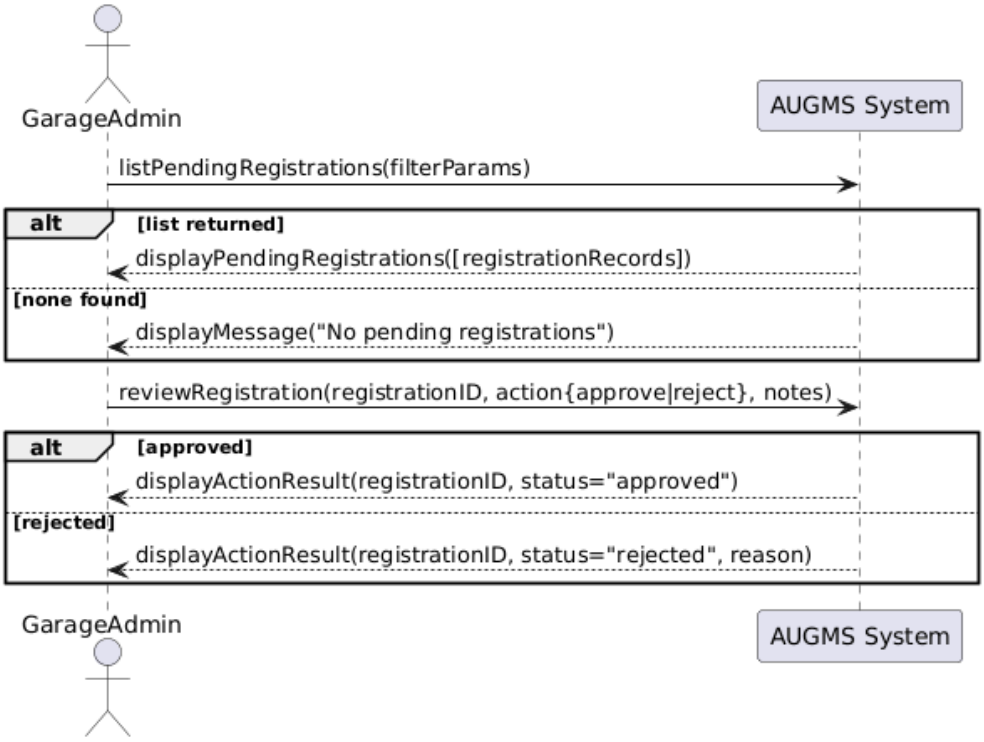


3.2.4 – System Sequence Diagram 4:

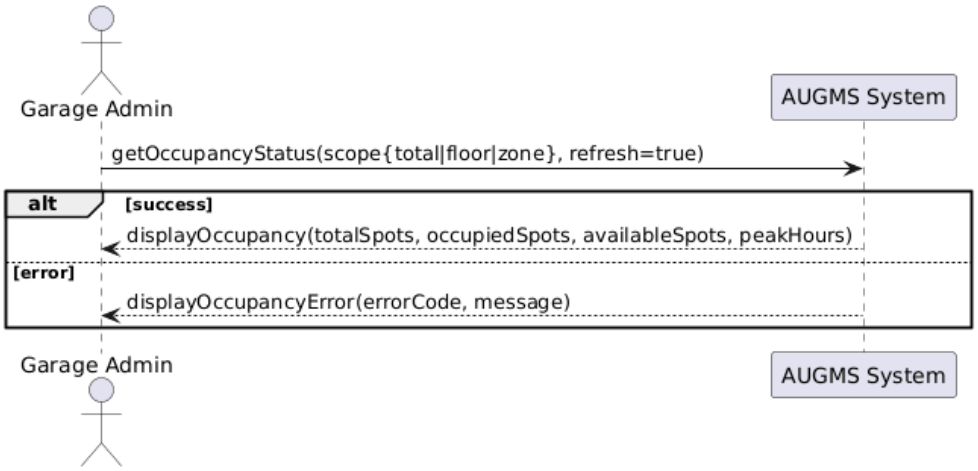
3.2.5 – System Sequence Diagram 5:



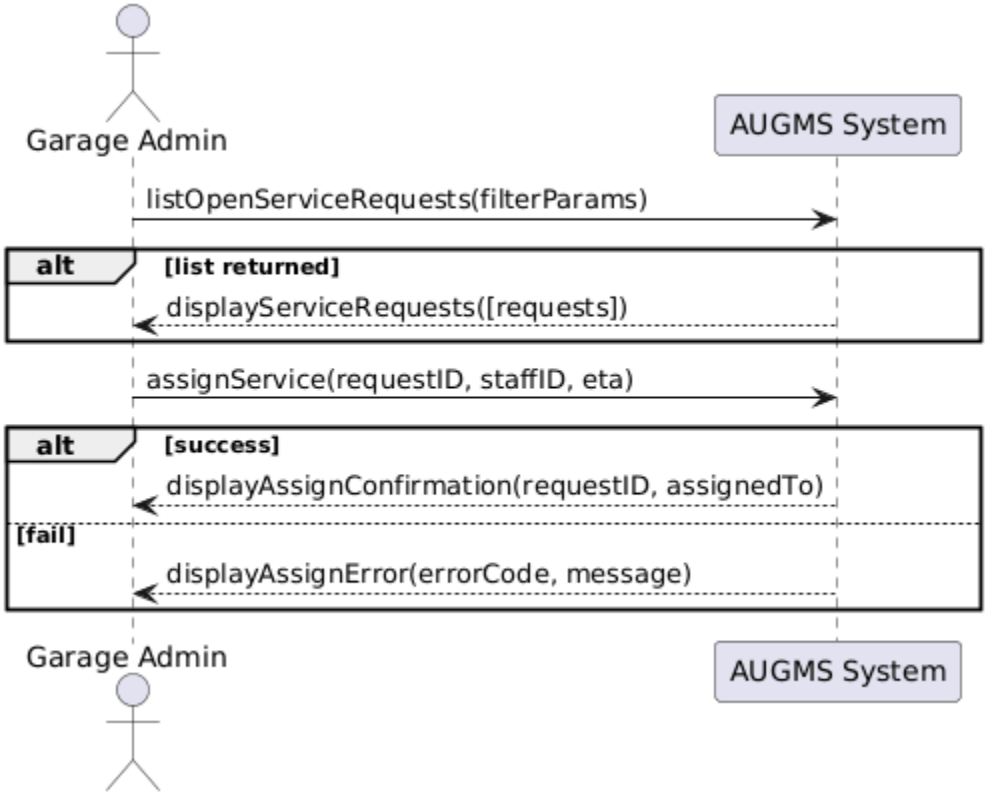
3.2.6 – System Sequence Diagram 6:



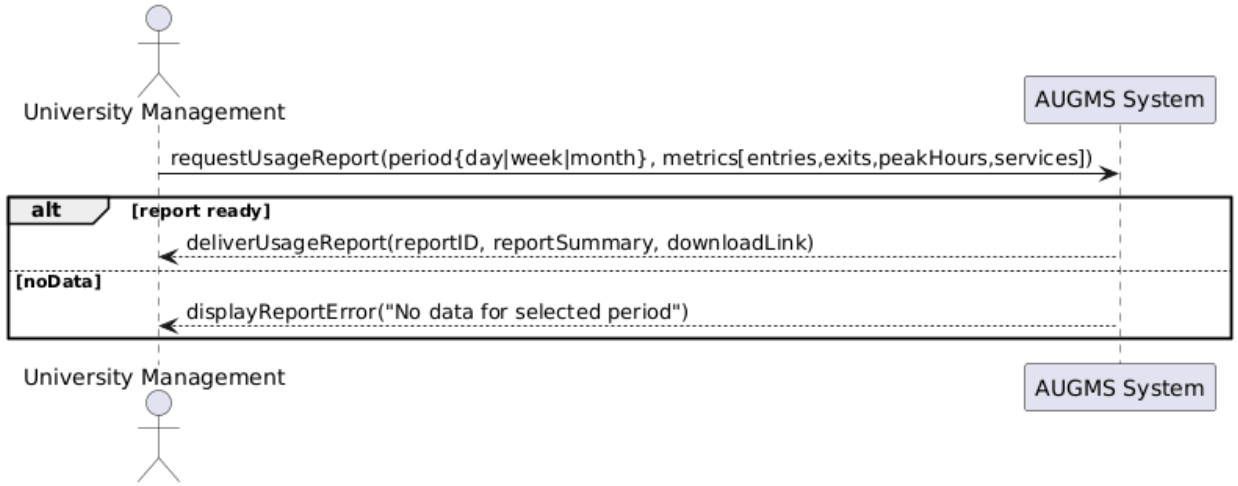
3.2.7 – System Sequence Diagram 7:



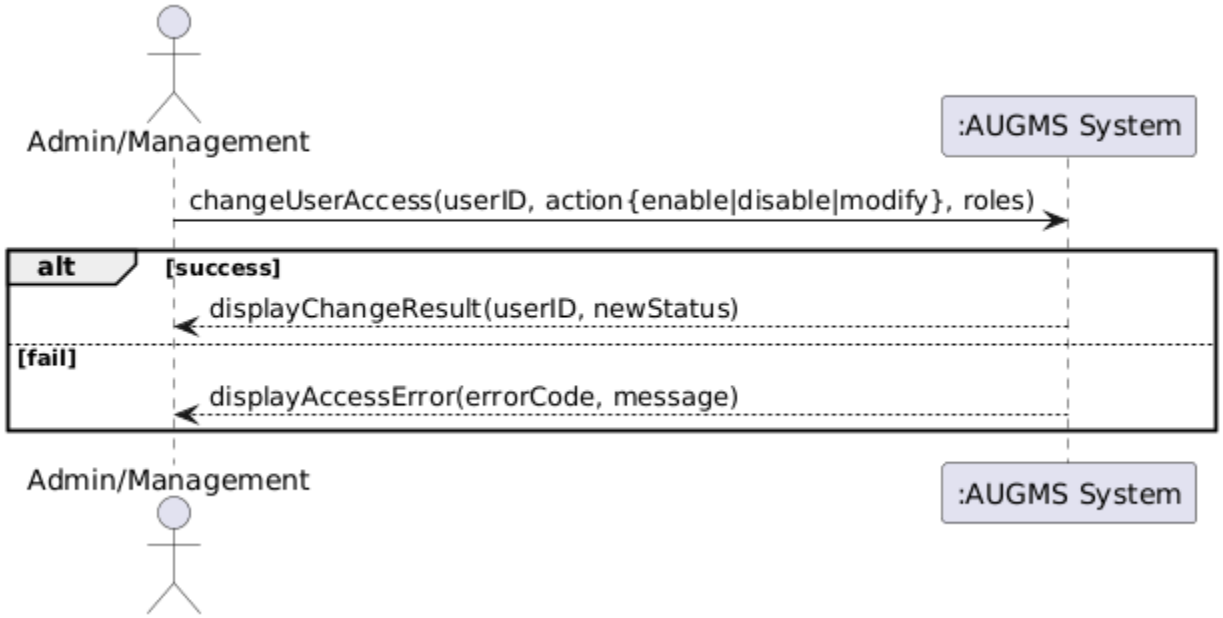
3.2.8 – System Sequence Diagram 8:



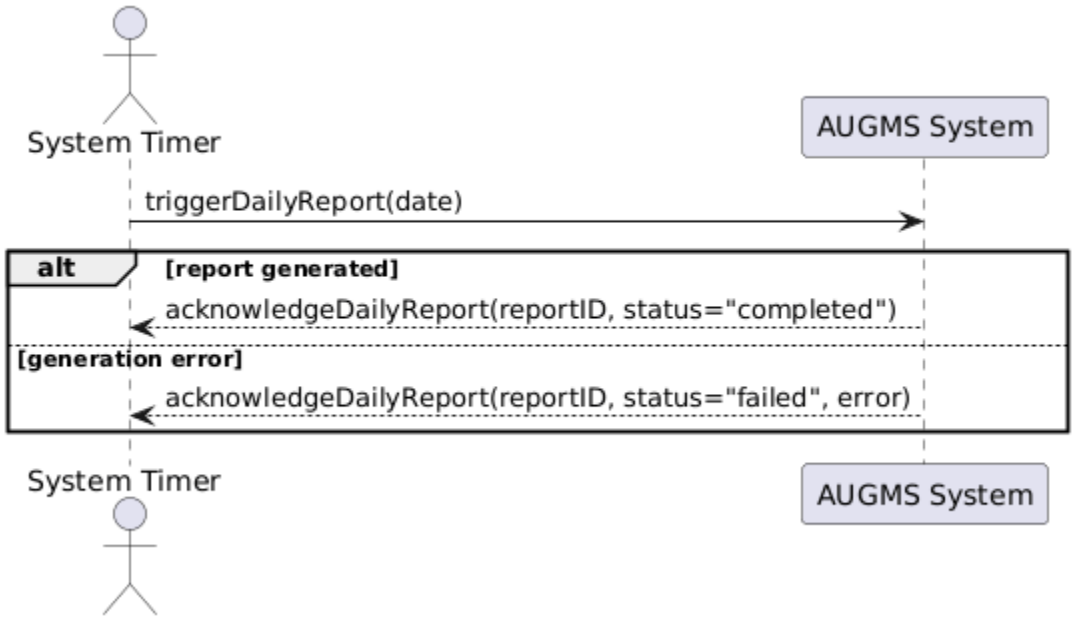
3.2.9 – System Sequence Diagram 9:



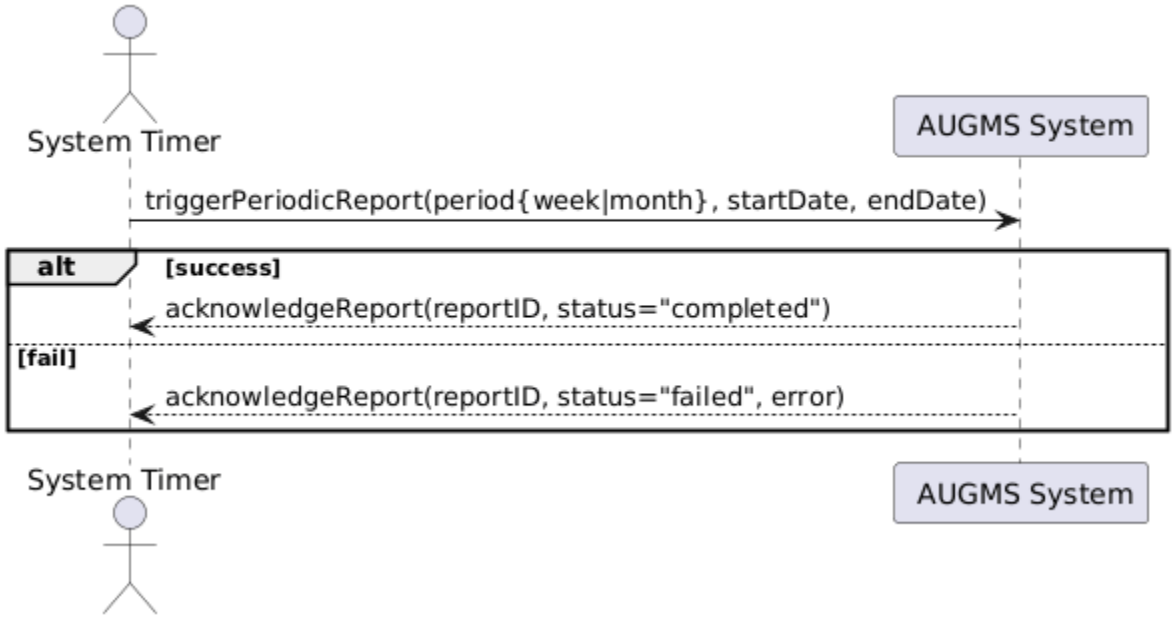
3.2.10 – System Sequence Diagram 10:



3.2.11 – System Sequence Diagram 11:

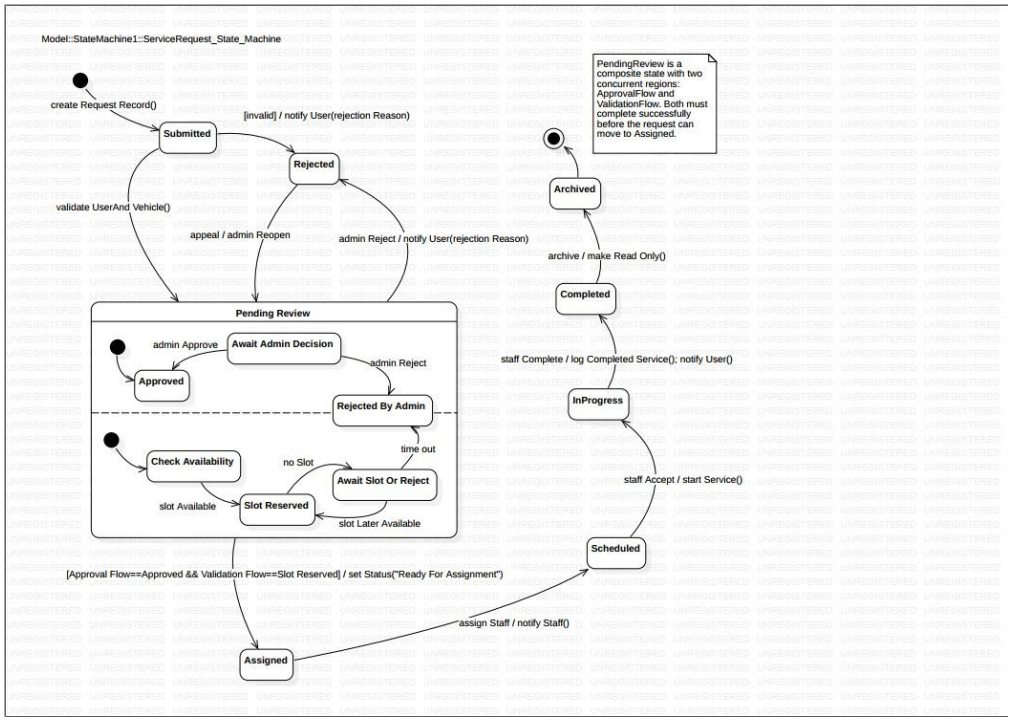


3.2.12 – System Sequence Diagram 12:

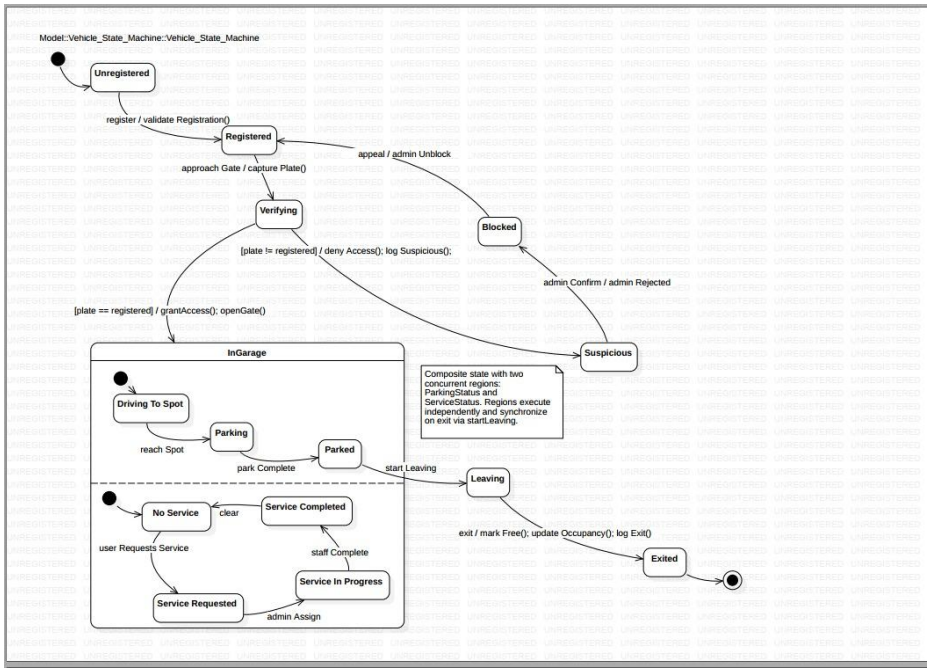


3.2.13 – System Sequence Diagram 13:

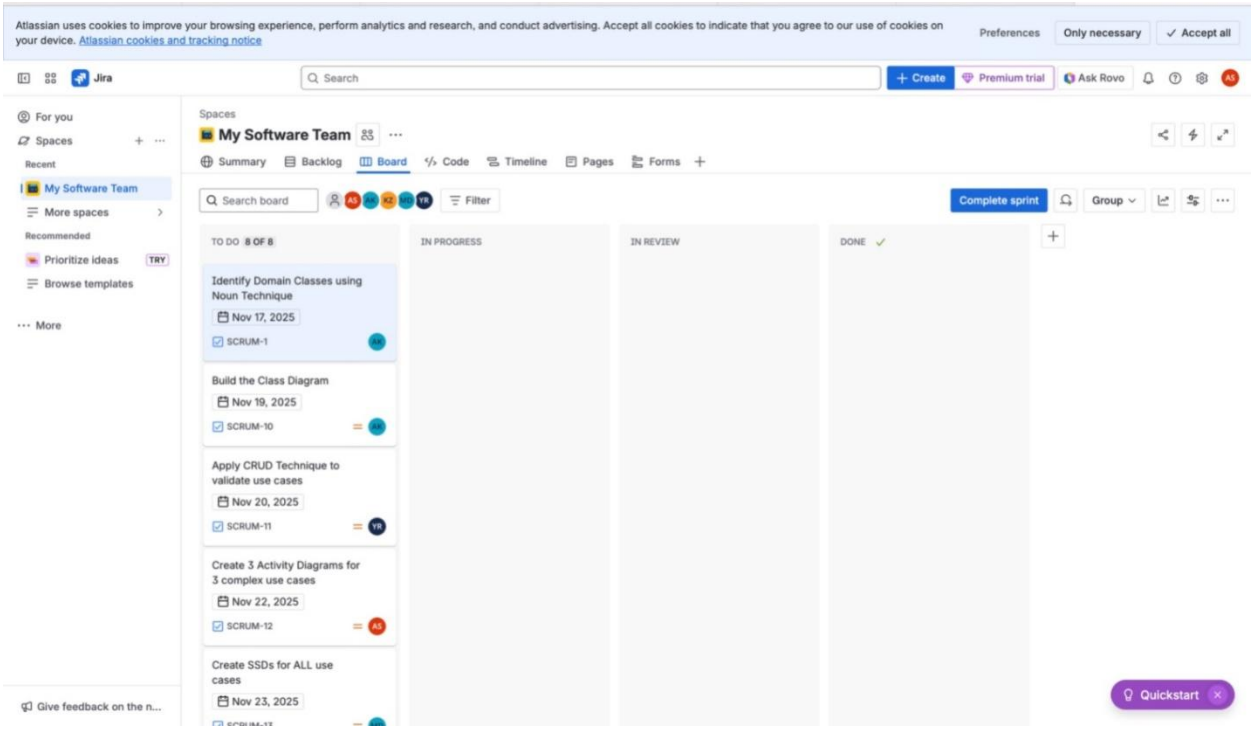
3.3.1 – State Machine Diagram 1: (Service Request)

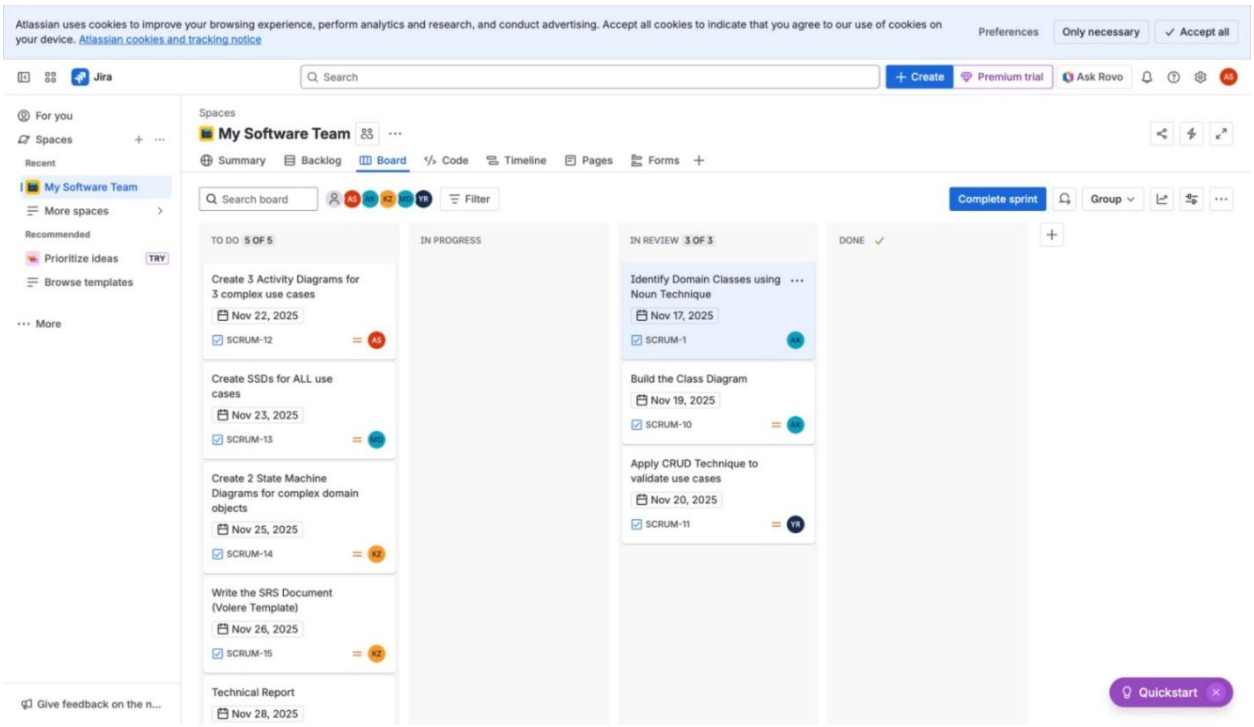


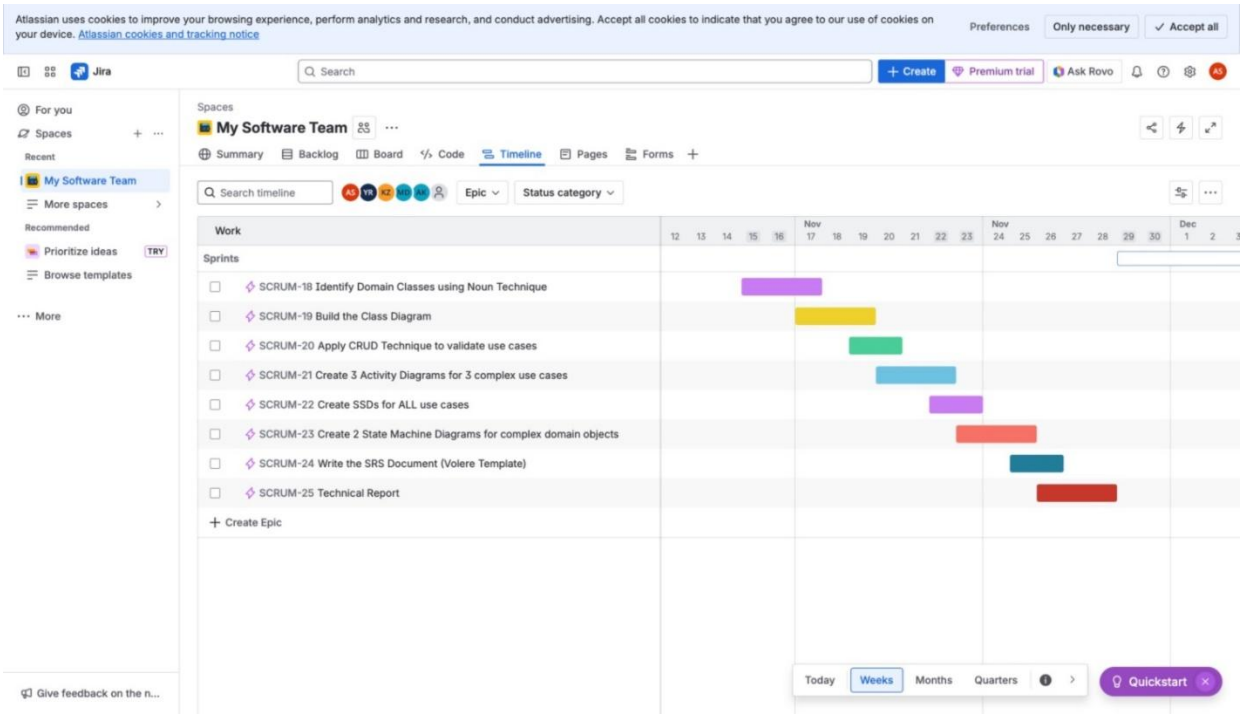
3.3.2 – State Machine Diagram 2: (Vehicle)

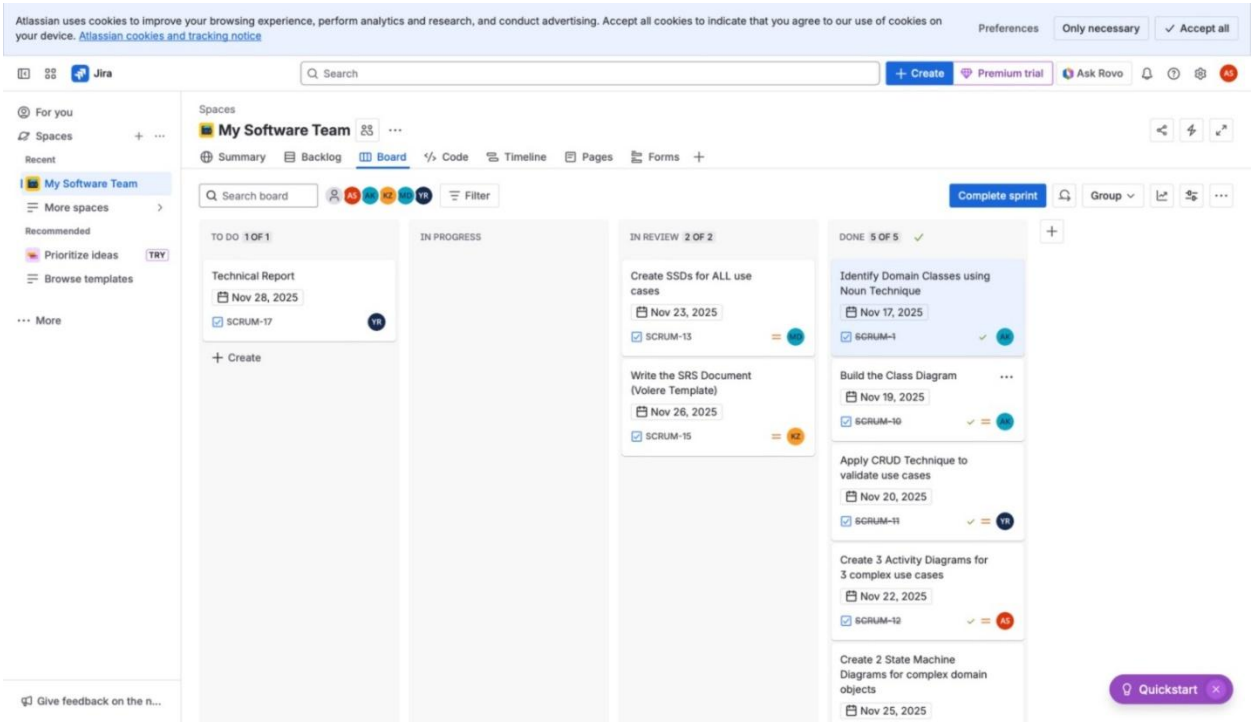


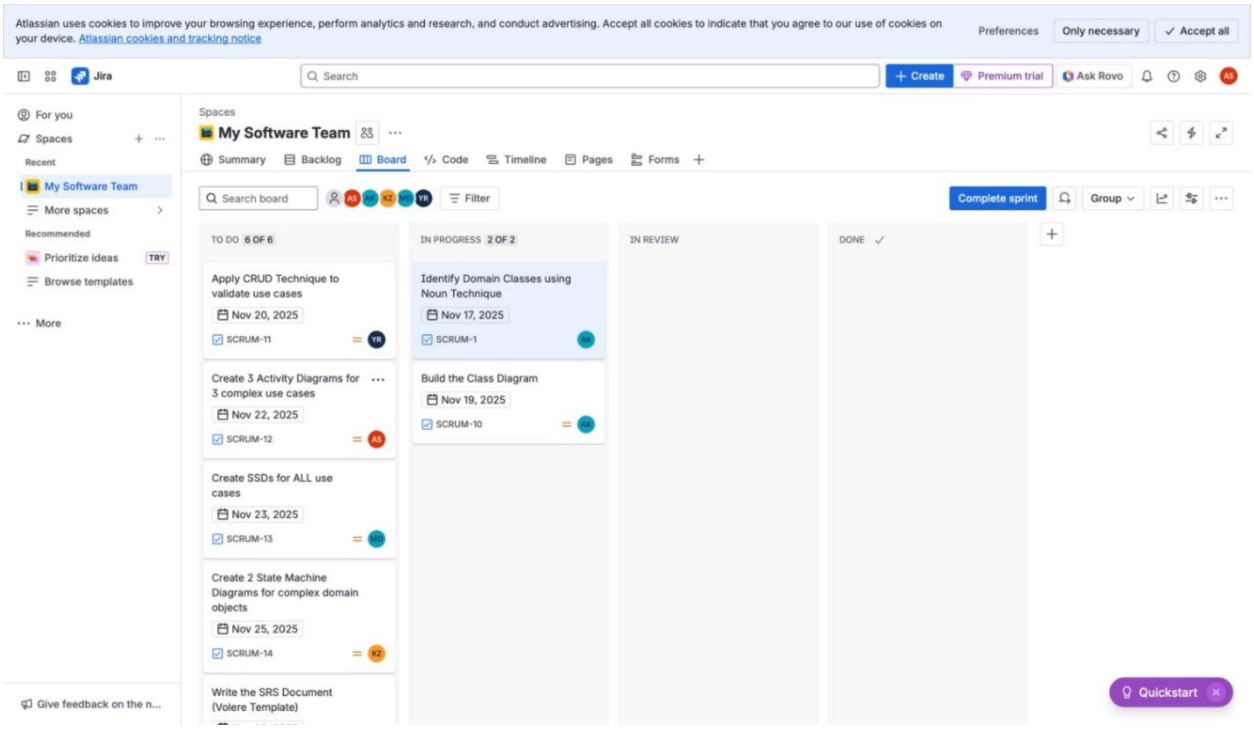
IV. Jira Timeline:

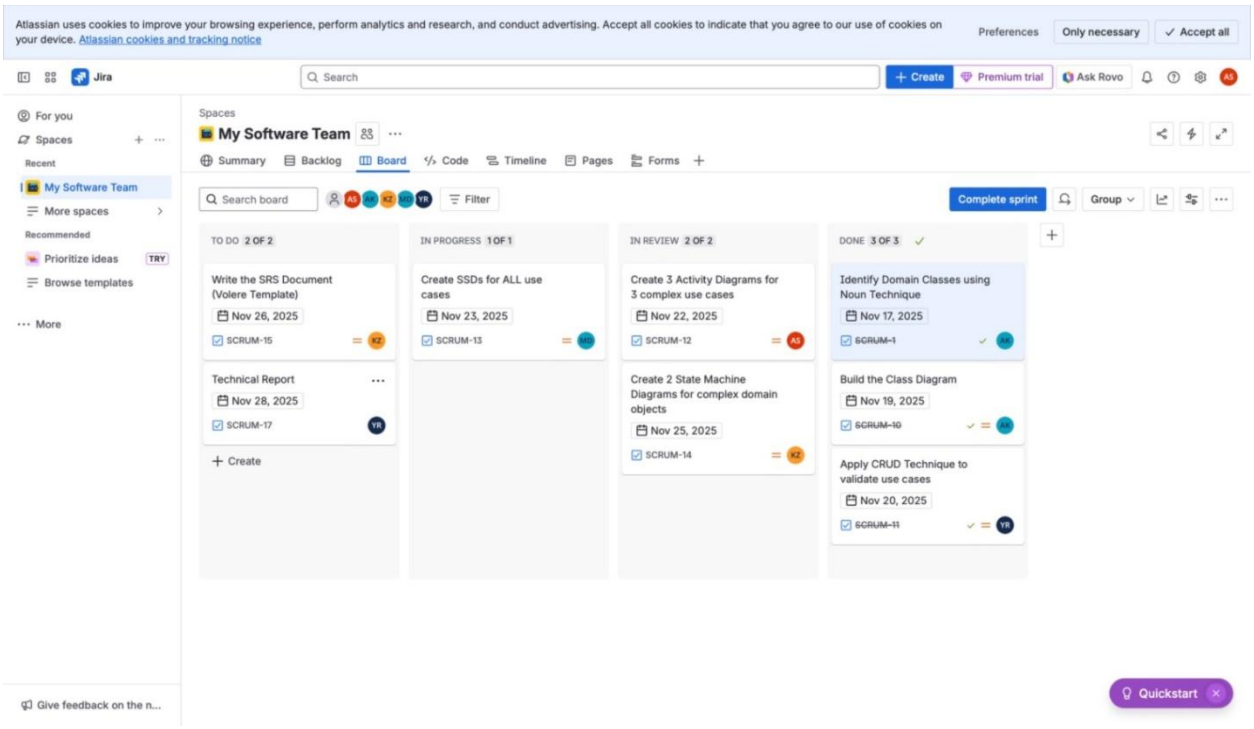


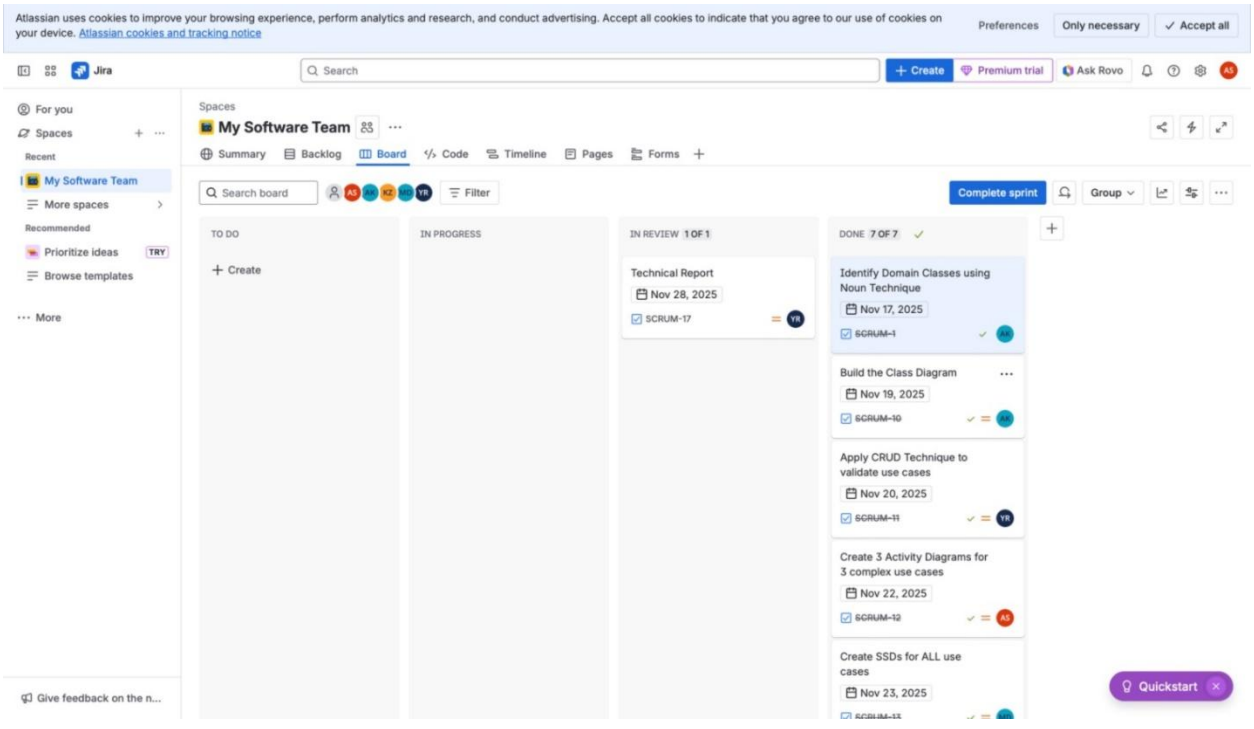

Atlassian uses cookies to improve your browsing experience, perform analytics and research, and conduct advertising. Accept all cookies to indicate that you agree to our use of cookies on your device. Atlassian cookies and tracking notice
Preferences
Only necessary
Accept all
Jira
Search
Create
Premium trial
Ask Rovo
For you
Spaces
Recent
My Software Team
More spaces
Recommended
Prioritize ideas
TRY
Browse templates
More
Give feedback on the n...
Spaces
My Software Team
Summary
Backlog
Board
Code
Timeline
Pages
Forms
Search board
Filter
Complete sprint
Group
TO DO 5 OF 5
IN PROGRESS
IN REVIEW 3 OF 3
DONE
Create 3 Activity Diagrams for 3 complex use cases
Nov 22, 2025
SCRUM-12
Create SSDs for ALL use cases
Nov 23, 2025
SCRUM-13
Create 2 State Machine Diagrams for complex domain objects
Nov 25, 2025
SCRUM-14
Write the SRS Document (Volere Template)
Nov 26, 2025
SCRUM-15
Technical Report
Nov 28, 2025
Identify Domain Classes using Noun Technique
Nov 17, 2025
SCRUM-1
Build the Class Diagram
Nov 19, 2025
SCRUM-10
Apply CRUD Technique to validate use cases
Nov 20, 2025
SCRUM-11
Quickstart

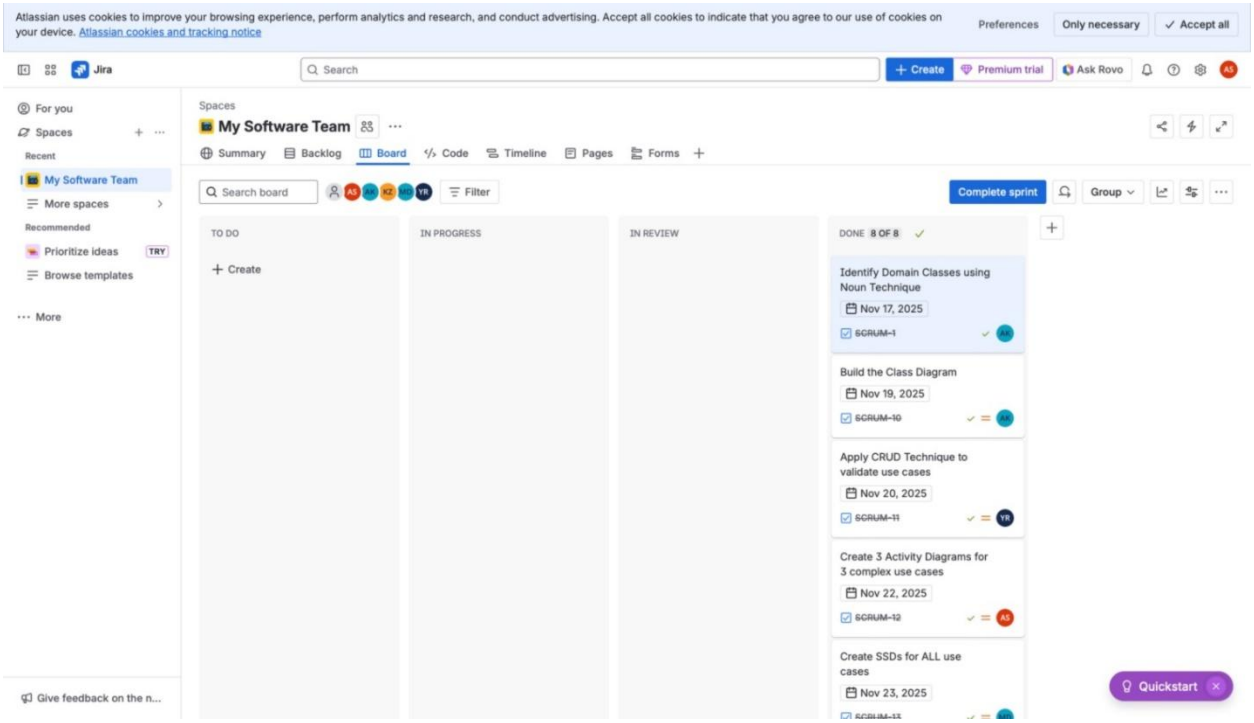

Atlassian uses cookies to improve your browsing experience, perform analytics and research, and conduct advertising. Accept all cookies to indicate that you agree to our use of cookies on your device. Atlassian cookies and tracking notice
Preferences
Only necessary
Accept all
Jira
Search
Create
Premium trial
Ask Rovo
For you
Spaces
Recent
My Software Team
More spaces
Recommended
Prioritize ideas
TRY
Browse templates
More
Give feedback on the n...
Spaces
My Software Team
Summary
Backlog
Board
Code
Timeline
Pages
Forms
Search timeline
Epic
Status category
Work
Nov
Dec
Sprints
SCRUM-18 Identify Domain Classes using Noun Technique
SCRUM-19 Build the Class Diagram
SCRUM-20 Apply CRUD Technique to validate use cases
SCRUM-21 Create 3 Activity Diagrams for 3 complex use cases
SCRUM-22 Create SSDs for ALL use cases
SCRUM-23 Create 2 State Machine Diagrams for complex domain objects
SCRUM-24 Write the SRS Document (Volere Template)
SCRUM-25 Technical Report
Create Epic
Today
Weeks
Months
Quarters
Quickstart


Atlassian uses cookies to improve your browsing experience, perform analytics and research, and conduct advertising. Accept all cookies to indicate that you agree to our use of cookies on your device. Atlassian cookies and tracking notice
Preferences
Only necessary
Accept all
Jira
Search
Create
Premium trial
Ask Rovo
For you
Spaces
Recent
My Software Team
More spaces
Recommended
Prioritize ideas
TRY
Browse templates
More
Give feedback on the n...
Spaces
My Software Team
Summary
Backlog
Board
Code
Timeline
Pages
Forms
Search board
Filter
Complete sprint
Group
TO DO 1 OF 1
Technical Report
Nov 28, 2025
SCRUM-17
Create
IN PROGRESS
IN REVIEW 2 OF 2
Create SSDs for ALL use cases
Nov 23, 2025
SCRUM-13
Write the SRS Document (Volere Template)
Nov 26, 2025
SCRUM-15
DONE 5 OF 5
Identify Domain Classes using Noun Technique
Nov 17, 2025
SCRUM-1
Build the Class Diagram
Nov 19, 2025
SCRUM-10
Apply CRUD Technique to validate use cases
Nov 20, 2025
SCRUM-11
Create 3 Activity Diagrams for 3 complex use cases
Nov 22, 2025
SCRUM-12
Create 2 State Machine Diagrams for complex domain objects
Nov 25, 2025
Quickstart


Atlassian uses cookies to improve your browsing experience, perform analytics and research, and conduct advertising. Accept all cookies to indicate that you agree to our use of cookies on your device. Atlassian cookies and tracking notice
Preferences
Only necessary
Accept all
Jira
Search
Create
Premium trial
Ask Rovo
For you
Spaces
Recent
My Software Team
More spaces
Recommended
Prioritize ideas
TRY
Browse templates
More
Give feedback on the n...
Spaces
My Software Team
Summary
Backlog
Board
Code
Timeline
Pages
Forms
Search board
Filter
Complete sprint
Group
TO DO 6 OF 6
Apply CRUD Technique to validate use cases
Nov 20, 2025
SCRUM-11
Create 3 Activity Diagrams for 3 complex use cases
Nov 22, 2025
SCRUM-12
Create SSDs for ALL use cases
Nov 23, 2025
SCRUM-13
Create 2 State Machine Diagrams for complex domain objects
Nov 25, 2025
SCRUM-14
Write the SRS Document (Volere Template)
IN PROGRESS 2 OF 2
Identify Domain Classes using Noun Technique
Nov 17, 2025
SCRUM-1
Build the Class Diagram
Nov 19, 2025
SCRUM-10
IN REVIEW
DONE
Quickstart


Atlassian uses cookies to improve your browsing experience, perform analytics and research, and conduct advertising. Accept all cookies to indicate that you agree to our use of cookies on your device. Atlassian cookies and tracking notice
Preferences
Only necessary
Accept all
Jira
Search
Create
Premium trial
Ask Rovo
For you
Spaces
Recent
My Software Team
More spaces
Recommended
Prioritize ideas
TRY
Browse templates
More
Give feedback on the n...
Spaces
My Software Team
Summary
Backlog
Board
Code
Timeline
Pages
Forms
Search board
Filter
Complete sprint
Group
TO DO 2 OF 2
Write the SRS Document (Volere Template)
Nov 26, 2025
SCRUM-15
Technical Report
Nov 28, 2025
SCRUM-17
Create
IN PROGRESS 1 OF 1
Create SSDs for ALL use cases
Nov 23, 2025
SCRUM-13
IN REVIEW 2 OF 2
Create 3 Activity Diagrams for 3 complex use cases
Nov 22, 2025
SCRUM-12
Create 2 State Machine Diagrams for complex domain objects
Nov 25, 2025
SCRUM-14
DONE 3 OF 3
Identify Domain Classes using Noun Technique
Nov 17, 2025
SCRUM-1
Build the Class Diagram
Nov 19, 2025
SCRUM-10
Apply CRUD Technique to validate use cases
Nov 20, 2025
SCRUM-11
Quickstart


Atlassian uses cookies to improve your browsing experience, perform analytics and research, and conduct advertising. Accept all cookies to indicate that you agree to our use of cookies on your device. Atlassian cookies and tracking notice
Preferences
Only necessary
Accept all
Jira
Search
Create
Premium trial
Ask Rovo
For you
Spaces
Recent
My Software Team
More spaces
Recommended
Prioritize ideas
TRY
Browse templates
More
Give feedback on the n...
Spaces
My Software Team
Summary
Backlog
Board
Code
Timeline
Pages
Forms
Search board
Filter
Complete sprint
Group
TO DO
Create
IN PROGRESS
IN REVIEW 1 OF 1
Technical Report
Nov 28, 2025
SCRUM-17
DONE 7 OF 7
Identify Domain Classes using Noun Technique
Nov 17, 2025
SCRUM-1
Build the Class Diagram
Nov 19, 2025
SCRUM-10
Apply CRUD Technique to validate use cases
Nov 20, 2025
SCRUM-11
Create 3 Activity Diagrams for 3 complex use cases
Nov 22, 2025
SCRUM-12
Create SSDs for ALL use cases
Nov 23, 2025
Quickstart



V. GitHub Repository: https://github.com/alyhassankamel/Automated-University-ParkingSystem.git

---

VI. Conclusion:

- This report apply core software engineering techniques, including domain modeling, CRUD analysis, and behavioral modeling, to refine and validate system requirements.

- The Noun Technique helped identify domain classes, forming a foundation for the system's architecture.

- The CRUD matrix ensured that all system use cases align with data operations, revealing missing interactions and improving requirement completeness.

- Activity diagrams, system sequence diagrams, and state machine diagrams provided detailed behavioral insights, clarifying system workflows and interactions between actors and components.

- Tools like Jira and GitHub supported clear task management and collaborative version control, enabling efficient teamwork throughout the deliverable.