
Why These Epics, Sprints & Issues Were Created?

The **main goal** of the project was to develop a **Staff Reception Management Dashboard** with features like:
✅ **Clock-in/out tracking** for staff members.
✅ **Absence tracking** for better workforce management.
✅ **Real-time toast notifications** for late staff.
✅ **Delivery scheduling** to manage outgoing deliveries.

To manage these different functionalities efficiently, I divided the project into **4 major Epics** and organized the workflow into **Sprints** and **Issues**.

Epic 1 : Staff Management (VRM-11)

Reason for this Epic:
The core functionality of the system is to **track employee attendance**, which is essential for workforce planning. This Epic focuses on **ensuring accurate check-in/out tracking** and **logging absences** efficiently.

✅ **Key Issues Created:**

- **Make API call and process attendance data (VRM-11-1)** → Needed for real-time data storage.
- **Store absence duration (VRM-11-2)** → Helps generate attendance reports.
- **Prompt staff on late clock-ins (VRM-11-3)** → Sends notifications if staff members arrive late.
- **"Out" button function (VRM-11-4)** → Allows proper tracking of when an employee leaves work.

Why These Issues?
Each issue breaks down the **attendance tracking logic** into manageable parts. By developing API communication first, the system ensures **data reliability** before implementing UI components.

Epic 2 : Delivery Management (VRM-12)

Reason for this Epic:

Since the reception dashboard also needs to handle **outgoing deliveries**, it was important to create a structured system to **schedule, track, and validate deliveries**.

✅ **Key Issues Created:**

- **Design table to track deliveries (VRM-12-1)** → Ensures deliveries are logged properly.
- **Validate input for new deliveries (VRM-12-2)** → Prevents incorrect data entry.
- **Add button for scheduling deliveries (VRM-12-3)** → Allows easy manual entry.
- **Create delivery dashboard (VRM-12-4)** → Provides a structured UI for the reception team.

Why These Issues?

The issues follow a **logical development sequence**:

1.First, structure the **data model** for deliveries.

2.Then, ensure **validation and error handling**.

3.Finally, build the **UI components** to interact with this data.

Epic 3. : Reception Dashboard (VRM-13)

Reason for this Epic:

A **user-friendly dashboard** is required for the reception team to **monitor staff attendance and deliveries** in one place. This Epic ensures that the UI/UX of the system is well-designed and intuitive.

✅ **Key Issues Created:**

- **Create navigation bar (VRM-13-1)** → Needed for structured page access.
- **Design user overview panel (VRM-13-2)** → Displays staff status in real time.
- **Add hover animations (VRM-13-3, VRM-13-4)** → Enhances user experience.
- **Implement real-time updates (VRM-13-5)** → Ensures accurate data display.

Why These Issues?

Since the system involves **real-time tracking**, a well-designed dashboard ensures **clarity**. The UI was developed in phases:

1.**Navigation and structure** first.

2.**Interactive elements and animations** next.

3.**Real-time updates and API integration** last.

Epic 4 : System Integration (VRM-14)

Reason for this Epic:

To ensure **seamless API communication**, this Epic focuses on integrating **attendance tracking, deliveries, and notifications** into one **functional system**.

✅ **Key Issues Created:**

- **Connect front-end with attendance API (VRM-14-1)** → Ensures real-time staff data.
- **Ensure all fields are filled before submission (VRM-14-2)** → Prevents user errors.
- **Validate user inputs (VRM-14-3)** → Enhances data integrity.
- **Add hover animations for buttons (VRM-14-4)** → Improves UI interaction.

Why These Issues?

This Epic was crucial because **without API integration, the dashboard would be non-functional**. The development approach:

1.**API first** → Ensuring backend functionality.

2.**Validation & security** → Making sure data flows correctly.

3.**UI refinements** → Enhancing usability.

Why This Sprint Structure?

Each Sprint followed this structure:

Sprint 1 → Core API functionalities (attendance, deliveries).

Sprint 2 → UI development (dashboard, navigation).

Sprint 3 → Refinements, testing, and integrations.

Sprint 4 → Final Testing & Optimizations.

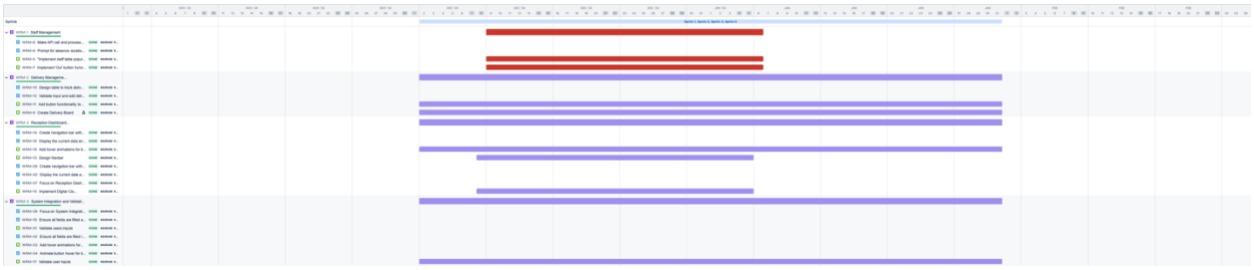
By organizing work in Sprints, I ensured that the **most critical functionalities were completed first** before adding **enhancements and optimizations**.

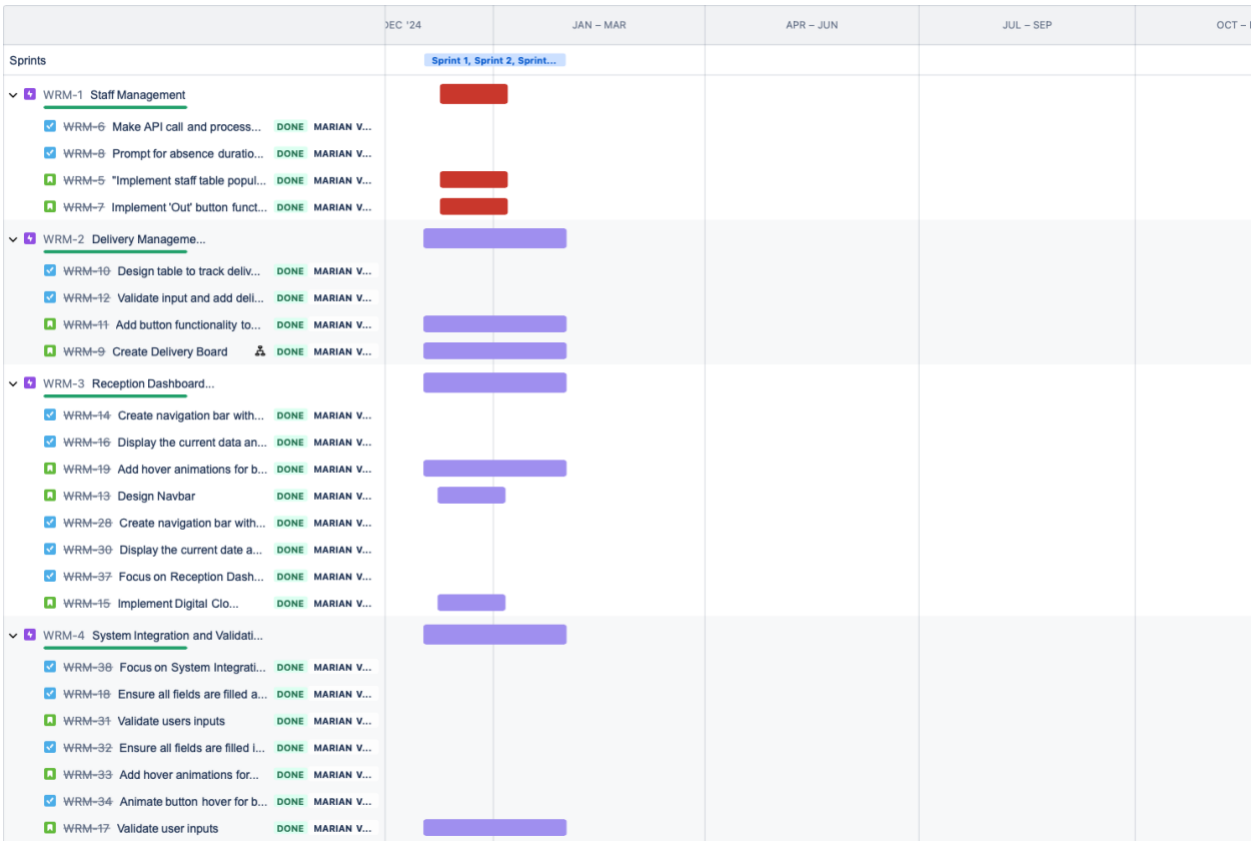
Final Thoughts

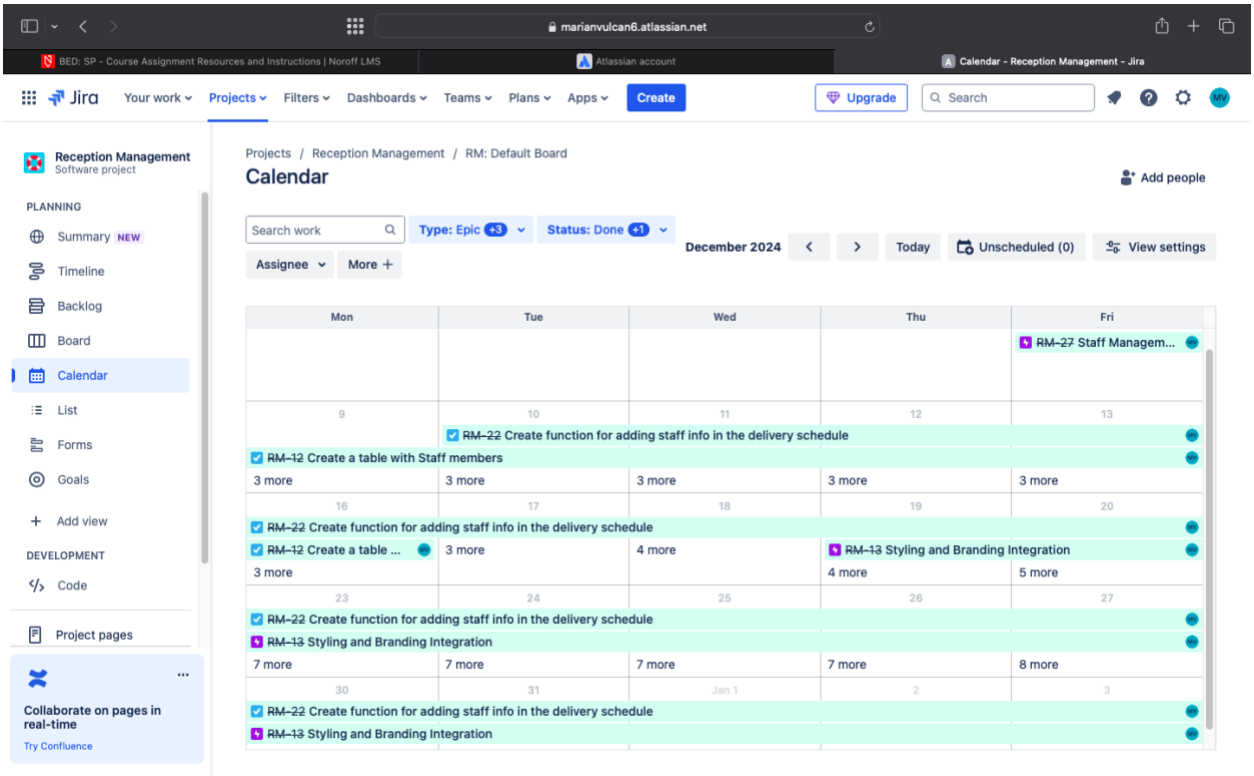
By **structuring the project with Epics, Sprints, and Issues**, the development process was **efficient and well-organized**.

- **Core features** (attendance & deliveries) were developed first.
- **User experience** was improved step by step.
- **API and system integration** ensured smooth communication.

This approach allowed me to **track progress effectively** and **deliver a functional, structured, and user-friendly system**.



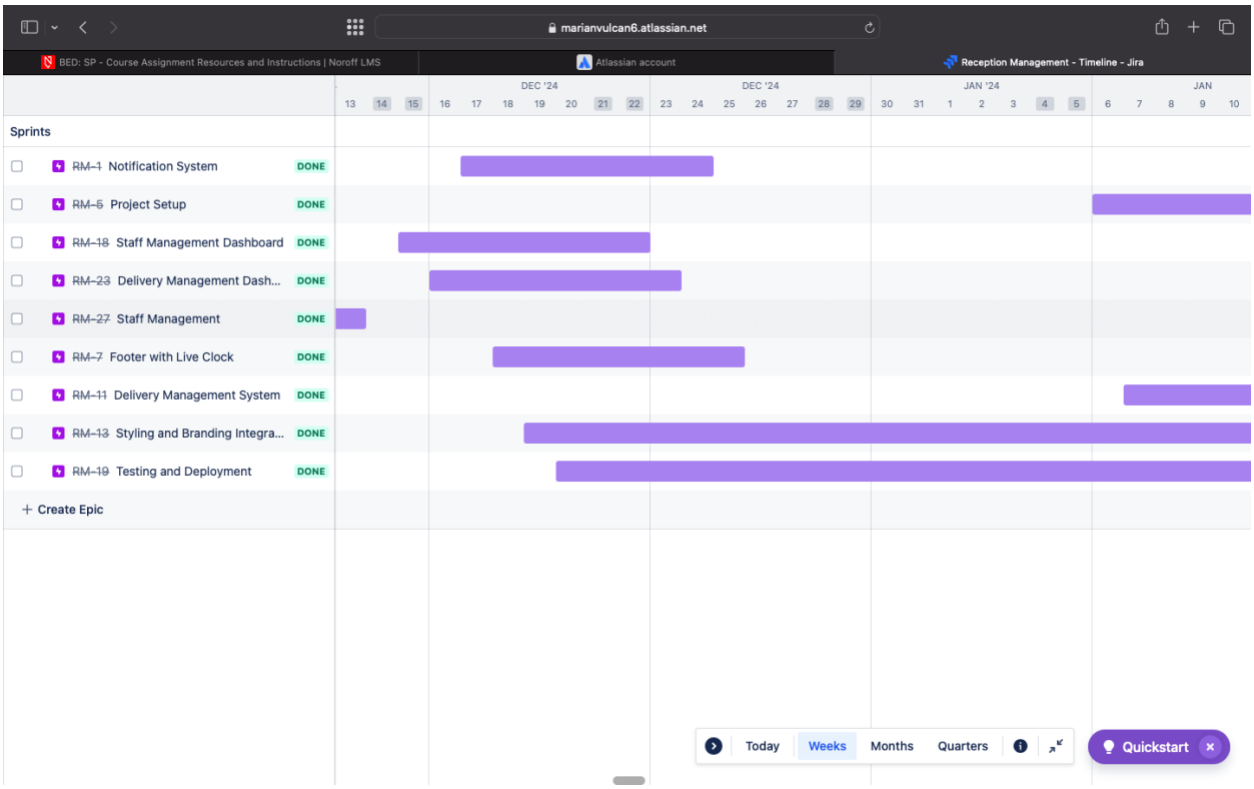

DEC '24
JAN – MAR
APR – JUN
JUL – SEP
OCT – I
Sprints
Sprint 1, Sprint 2, Sprint...
WRM-1 Staff Management
WRM-6 Make API call and process... DONE MARIAN V...
WRM-8 Prompt for absence duratio... DONE MARIAN V...
WRM-5 "Implement staff table popul... DONE MARIAN V...
WRM-7 Implement 'Out' button funct... DONE MARIAN V...
WRM-2 Delivery Manageme...
WRM-10 Design table to track deliv... DONE MARIAN V...
WRM-12 Validate input and add deli... DONE MARIAN V...
WRM-11 Add button functionality to... DONE MARIAN V...
WRM-9 Create Delivery Board DONE MARIAN V...
WRM-3 Reception Dashboard...
WRM-14 Create navigation bar with... DONE MARIAN V...
WRM-16 Display the current data an... DONE MARIAN V...
WRM-19 Add hover animations for b... DONE MARIAN V...
WRM-13 Design Navbar DONE MARIAN V...
WRM-28 Create navigation bar with... DONE MARIAN V...
WRM-30 Display the current date a... DONE MARIAN V...
WRM-37 Focus on Reception Dash... DONE MARIAN V...
WRM-15 Implement Digital Clo... DONE MARIAN V...
WRM-4 System Integration and Validati...
WRM-38 Focus on System Integrati... DONE MARIAN V...
WRM-18 Ensure all fields are filled a... DONE MARIAN V...
WRM-31 Validate users inputs DONE MARIAN V...
WRM-32 Ensure all fields are filled i... DONE MARIAN V...
WRM-33 Add hover animations for... DONE MARIAN V...
WRM-34 Animate button hover for b... DONE MARIAN V...
WRM-17 Validate user inputs DONE MARIAN V...

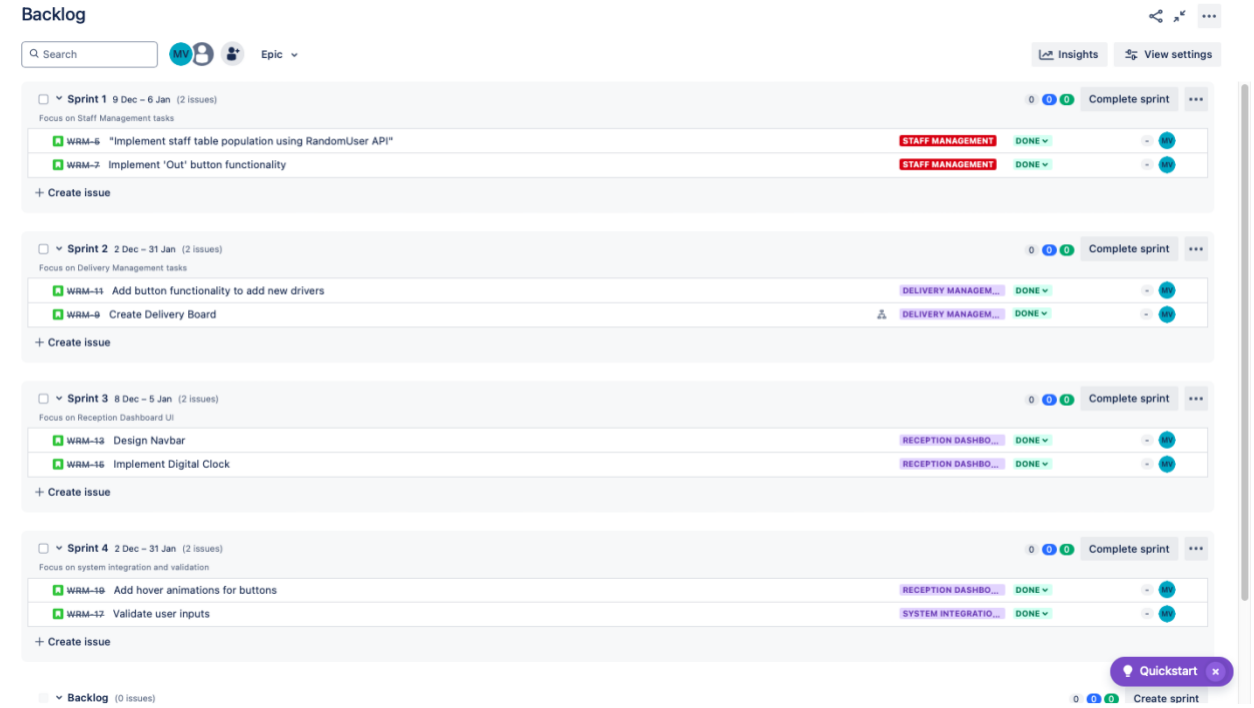

marianvulcan6.atlassian.net
BED: SP - Course Assignment Resources and Instructions | Noroff LMS
Atlassian account
Calendar - Reception Management - Jira
Jira
Your work
Projects
Filters
Dashboards
Teams
Plans
Apps
Create
Upgrade
Search
Reception Management
Software project
PLANNING
Summary NEW
Timeline
Backlog
Board
Calendar
List
Forms
Goals
Add view
DEVELOPMENT
Code
Project pages
Collaborate on pages in real-time
Try Confluence
Projects / Reception Management / RM: Default Board
Calendar
Add people
Search work
Type: Epic +3
Status: Done +1
Assignee
More
December 2024
Today
Unscheduled (0)
View settings
Mon
Tue
Wed
Thu
Fri
RM-27 Staff Managem...
RM-22 Create function for adding staff info in the delivery schedule
RM-12 Create a table with Staff members
RM-12 Create a table ...
RM-13 Styling and Branding Integration
3 more
4 more
5 more
7 more
8 more

All sprints

Search | Epic 4 | Sprint 4 | Clear filters

TO DO	IN PROGRESS	18 OF 8 MAX: 20	ISSUES
+ Create issue		Add hover animations for buttons RECEPTION DASHBO... ~~WRM-10~~	
		"Implement staff table population using RandomUser API" STAFF MANAGEMENT ~~WRM-5~~	
		Add button functionality to add new drivers DELIVERY MANAGEM... ~~WRM-11~~	
		Design Navbar RECEPTION DASHBO... ~~WRM-13~~	
		Create Delivery Board DELIVERY MANAGEM... ~~WRM-9~~	
		Validate user inputs SYSTEM INTEGRATIO... ~~WRM-17~~	
		Implement 'Out' button functionality STAFF MANAGEMENT ~~WRM-7~~	
		Implement Digital Clock RECEPTION DASHBO... ~~WRM-15~~	


marianvulcan6.atlassian.net
BED: SP - Course Assignment Resources and Instructions | Noroff LMS
Atlassian account
Reception Management - Timeline - Jira
DEC '24
DEC '24
JAN '24
JAN
13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 1 2 3 4 5 6 7 8 9 10
Sprints
RM-1 Notification System DONE
RM-5 Project Setup DONE
RM-18 Staff Management Dashboard DONE
RM-23 Delivery Management Dash... DONE
RM-27 Staff Management DONE
RM-7 Footer with Live Clock DONE
RM-11 Delivery Management System DONE
RM-13 Styling and Branding Integra... DONE
RM-19 Testing and Deployment DONE
Create Epic
Today
Weeks
Months
Quarters
Quickstart


Backlog
Search
Epic
Insights
View settings
Sprint 1 9 Dec – 6 Jan (2 issues)
Focus on Staff Management tasks
Complete sprint
WRM-6 "Implement staff table population using RandomUser API"
STAFF MANAGEMENT
DONE
WRM-7 Implement 'Out' button functionality
STAFF MANAGEMENT
DONE
Create issue
Sprint 2 2 Dec – 31 Jan (2 issues)
Focus on Delivery Management tasks
Complete sprint
WRM-11 Add button functionality to add new drivers
DELIVERY MANAGEM...
DONE
WRM-9 Create Delivery Board
DELIVERY MANAGEM...
DONE
Create issue
Sprint 3 8 Dec – 5 Jan (2 issues)
Focus on Reception Dashboard UI
Complete sprint
WRM-13 Design Navbar
RECEPTION DASHBO...
DONE
WRM-16 Implement Digital Clock
RECEPTION DASHBO...
DONE
Create issue
Sprint 4 2 Dec – 31 Jan (2 issues)
Focus on system integration and validation
Complete sprint
WRM-19 Add hover animations for buttons
RECEPTION DASHBO...
DONE
WRM-17 Validate user inputs
SYSTEM INTEGRATIO...
DONE
Create issue
Quickstart
Backlog (0 issues)
Create sprint