

## Reflection Report: Jira Project Management

### Section 1: Sprints, Epics, and Issues

#### Epics Overview

The project was divided into five Epics to align with the customer's requirements and ensure clear focus areas:

1. **Staff Management:** Core functionalities for managing staff attendance and statuses.
2. **Delivery Tracking:** Features to manage and monitor deliveries.
3. **User Experience:** Enhancements to improve usability and visual appeal.
4. **Navigation:** Creation of a navigation system for future expansion.
5. **Data Management:** Backend tasks to validate data and integrate APIs.

#### Why These Issues Were Chosen

Each issue was directly derived from the requirements provided by WeDeliverTECH:

- **Clock-in/Clock-out System:** Essential for tracking staff attendance.
- **Delivery Board:** Enables delivery tracking, a key functionality.
- **API Integration:** Provides realistic staff data.
- **Toast Notifications:** Ensures that overdue statuses are handled promptly.
- **Validation and Styling:** Improves user experience and aligns with company branding.

#### Screenshots

- **Epics Panel:** Displaying all created Epics.
- **Issues View:** Showing the breakdown of tasks within each Epic.
- **Backlog:** Organized issues for sprint planning.

*(Please see attached screenshots of the Jira-Scrum project)*

### Section 2: Project Timeline, Board, and Backlog

#### Timeline

The project was structured into four sprints, each lasting one week, from **18 November to 15 December**:

1. **Sprint 1:** Focused on Staff Management core features.
2. **Sprint 2:** Implemented Delivery Tracking functionalities.
3. **Sprint 3:** Enhanced User Experience with animations and styling.
4. **Sprint 4:** Finalized with testing, bug fixes, and documentation.

## Progress Summary

- **Sprint 1:** Completed foundational tasks (clock-in/out, status updates).
- **Sprint 2:** Delivered a functional Delivery Board and notification system.
- **Sprint 3:** Improved usability through animations and a digital clock.
- **Sprint 4:** Conducted testing and resolved all bugs.

## Screenshots

- **Timeline View:** Highlighting task distribution across sprints.
- **Active Sprint Board:** Tasks moved through "To Do," "In Progress," and "Done."
- **Backlog:** Organized by priority and dependencies.

*(See attached screenshots at the end.)*

## Section 3: Project Management Reflection

### Planning and Organization

Using Jira Scrum methodology ensured:

1. **Clear Prioritization:** Tasks were broken into Epics and Issues to address dependencies.
2. **Efficient Progress Tracking:** The Board provided visibility into task status and sprint progress.
3. **Iterative Delivery:** Features were delivered incrementally, reducing risks.

### Detailed Feature Highlights

1. **Staff Table:**
  - **Design:** Displays a list of staff with their picture, name, status, and time details.
  - **Functionality:** Allows the receptionist to toggle staff status (In/Out) and updates the table dynamically.
  - **Utility:** Ensures clear visibility of staff availability and absence durations.
2. **Toast Notification:**
  - **Feature:** Provides alerts when staff members exceed their expected return time.
  - **Behavior:** Remains on screen until dismissed by the receptionist.
  - **Importance:** Highlights critical updates for immediate action.
3. **Digital Clock:**
  - **Design:** A live-updating clock at the bottom of the page.
  - **Format:** Displays the current date and time in "Day, Month, Year, Hour:Minute:Second" format.
  - **Purpose:** Helps users stay aware of time-sensitive tasks.
4. **Delivery Board:**
  - **Design:** Tracks delivery driver details, including vehicle type, name, and return time.

- **Icons:** Uses vehicle icons for quick visual reference.
- **Functionality:** Allows the receptionist to add, update, and clear driver entries with validation.

## Challenges and Solutions

1. **API Integration:**
  - Challenge: Converting API responses into usable objects.
  - Solution: Developed robust classes and tested API calls extensively.
2. **Digital Clock:**
  - Challenge: Ensuring that the clock dynamically updated every second without lag.
  - Solution: Leveraged JavaScript's `setInterval()` function and optimised DOM manipulation to minimise performance issues.
3. **Validation of Staff Return Time:**
  - Challenge: Ensuring the return time was accurately calculated based on the entered out-time and duration.
  - Solution: Developed a time calculation function to convert minutes to hours and minutes, then validate the input format (hh:mm) for consistency.
4. **Expected Return Time Calculation:**
  - Challenge: Displaying the expected return time in a user-friendly format while ensuring consistency with the current digital clock.
  - Solution: Created utility functions to add durations to out-times and handle edge cases (e.g., crossing over midnight).
5. **Toast Notifications:**
  - Challenge: Ensuring notifications appeared only once and were user-controlled.
  - Solution: Used event listeners and state management to track active notifications and prevent duplicates.
6. **User Validation:**
  - Challenge: Implementing validation for multiple input types (e.g., text, dropdowns, and time inputs).
  - Solution: Applied regular expressions, error messages, and real-time input checks to guide users.

## Technologies Used

1. **Frontend:**
  - HTML5, CSS3, Bootstrap 5.
  - FontAwesome for icons.
2. **Scripting:**
  - JavaScript (ES6+).
  - OOP concepts (Classes, Methods, Data Encapsulation).
  - jQuery for simplified DOM manipulation.
3. **APIs:**
  - Random User API for generating mock staff data.

## Achievements

- Delivered a fully functional Reception Management Dashboard.
- Implemented real-time clock updates and hover animations for enhanced UX.

- Aligned with WeDeliverTECH's branding guidelines.
- Validated user inputs to ensure accurate data capture.


### **Learning Outcomes**

- **Improved Proficiency in Scrum:** Enhanced ability to prioritize and manage tasks using Jira Scrum boards.
- **Technical Growth:** Strengthened skills in OOP design, dynamic UI updates, and API integration.
- **Team Collaboration:** Fostered better communication and teamwork through structured sprint goals.

## References

- LinkedIn Learning. (2024). *Courses for professional development*. Available at: <https://www.linkedin.com/learning/> (Accessed: 14 December 2024).
- Noroff University College. (2024). *Studies in technology and design*. Available at: <https://www.noroff.no/en/studies/university-college> (Accessed: 14 December 2024).
- Beck, K. et al. (2001). *Manifesto for Agile Software Development*. Available at: <https://agilemanifesto.org/> (Accessed: 14 December 2024).
- Bootstrap. (2024). *Introduction to Bootstrap Framework*. Available at: <https://getbootstrap.com/docs/5.0/getting-started/introduction/> (Accessed: 14 December 2024).
- MDN Web Docs. (2024). *JavaScript Basics*. Available at: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide> (Accessed: 14 December 2024).

## Appendix: Screenshots

**Abdul Fatah**  
Software project

PLANNING

Summary NEW

Timeline

Backlog

**Board**

Forms

+ Add view

DEVELOPMENT

Code


Project pages


Project settings

Archived issues NEW

Projects / Abdul Fatah



**BED Sprint 1**  
Staff management table, validation and API population

**AH**





Epic ▾

TO DO 1



Display Staff Status  
**STAFF MANAGEMENT**  
 BED-7 



+ Create issue



IN PROGRESS 1


Develop Clock in/out functionality  
**STAFF MANAGEMENT**  
 BED-6 

DONE 3 ✓

Overdue notifications  
**STAFF MANAGEMENT**  
 BED-8 ✓ 

Use <https://randomuser.me> API to populate staff data.  
**DATA MANAGEMENT**  
 BED-12 ✓ 

Validate user inputs for clock-in/out operations.  
**DATA MANAGEMENT**  
 BED-13 ✓ 

**Abdul Fatah**  
Software project

PLANNING

Summary NEW

Timeline

Backlog

**Board**

Forms

+ Add view

DEVELOPMENT

Code


Project pages


Project settings

Archived issues NEW

Projects / Abdul Fatah



**BED Sprint 1**  
Staff management table, validation and API population



**AH**







Epic ▾



TO DO 5

Develop Clock in/out functionality  
**STAFF MANAGEMENT**  
 BED-6 

Display Staff Status  
**STAFF MANAGEMENT**  
 BED-7 

Overdue notifications  
**STAFF MANAGEMENT**  
 BED-8 

Use <https://randomuser.me> API to populate staff data. ...  
**DATA MANAGEMENT**  
 BED-12 

Validate user inputs for clock-in/out operations.  
**DATA MANAGEMENT**  
 BED-13 

+ Create issue

IN PROGRESS

DONE ✓



Jira

Your work

Projects

Filters

Dashboards

Teams

Plans

Apps

Create

Abdul Fatah

Software project

PLANNING

Summary NEW

Timeline

Backlog

Board

Forms

+ Add view

DEVELOPMENT

Code

Project pages

Project settings

Archived issues NEW

Projects / Abdul Fatah

BED Sprint 1

Staff management table, validation and API population

Q Search

AH

Epic

TO DO

IN PROGRESS

DONE 5 ✓

STAFF MANAGEMENT

BED-8 ✓

Use <https://randomuser.me> API to populate staff data.

DATA MANAGEMENT

BED-12 ✓

Display Staff Status

STAFF MANAGEMENT

BED-7 ✓

Develop Clock in/out functionality

STAFF MANAGEMENT

BED-6 ✓

Validate user inputs for clock-in/out operations.

DATA MANAGEMENT

BED-13 ✓

Backlog

Q Search

AH

Epic

Insights

View settings

Epic

Issues without epic

Staff Management

Delivery Tracking

User Experience

Data Management

Navigation

Project finalization and testing

+ Create epic

BED Sprint 5

Add dates

0 issues

Start sprint

Plan your sprint

Drag issues from the **Backlog** section, or create new issues, to plan the work for this sprint. Select **Start sprint** when you're ready.

+ Create issue

Backlog

0 issues

Create sprint

Your backlog is empty.

+ Create issue

Jira

Your work

Projects

Filters

Dashboards

Teams

Plans

Apps

Create

Abdul Fatah

Software project

PLANNING

Summary NEW

Timeline

Backlog

Board

Forms

+ Add view

DEVELOPMENT

Code

Project pages

Project settings

Archived issues NEW

Projects / Abdul Fatah

BED Sprint 1

Staff management table, validation and API population

Q Search

AH

Epic

TO DO

IN PROGRESS

DONE 5 ✓

STAFF MANAGEMENT

BED-8 ✓

Use <https://randomuser.me> API to populate staff data.

DATA MANAGEMENT

BED-12 ✓

Display Staff Status

STAFF MANAGEMENT

BED-7 ✓

Develop Clock in/out functionality

STAFF MANAGEMENT

BED-6 ✓

Validate user inputs for clock-in/out operations.

DATA MANAGEMENT

BED-13 ✓

Backlog

Q Search

AH

Epic

Insights

View settings

Epic

Issues without epic

Staff Management

Delivery Tracking

User Experience

Data Management

Navigation

Project finalization and testing

+ Create epic

BED Sprint 5

Add dates

0 issues

Start sprint

Plan your sprint

Drag issues from the **Backlog** section, or create new issues, to plan the work for this sprint. Select **Start sprint** when you're ready.

+ Create issue

Backlog

0 issues

Create sprint

Your backlog is empty.

+ Create issue