Project Plan & Reflection Report

Semester Project

Student Name: Farid Vahabzada
Date of Submission: 13 December 2023

Noroff
School of technology
and digital media

Table of Contents

Introduction

This document consists of two sections, **Project Plan** and **Reflection Report** respectively.

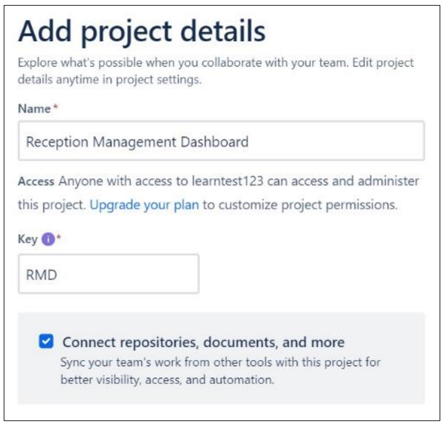
In the first section, **Project Plan**, how the whole project progressed, by using the **Jira** software, and what stages the development process included is demonstrated detailly.

In the second section, we are more focused on the problems and difficulties experienced throughout the development of the web application. And how some of the obstacles were tackled while other issues were left for the future versions of the app. This section, **Reflection Report**, has exceeded the limit, with a total of **1175** words being used excluding the figure captions, as the required range is **between 500–1000 words**.

Project Plan

First Steps

The first week of four weeks was **only** dedicated to the material coverage and refreshing the memory of the previous weeks. Starting from the second week, the Jira project named **Reception Management Dashboard** (**RMD**) was created, lasting for the whole three weeks, as shown below:



Add project details

Explore what's possible when you collaborate with your team. Edit project details anytime in project settings.

Name *

Reception Management Dashboard

Access Anyone with access to learntest123 can access and administer this project. Upgrade your plan to customize project permissions.

Key *

RMD

Connect repositories, documents, and more

Sync your team's work from other tools with this project for better visibility, access, and automation.

Figure 1. Starting a new project.

Eventually, all the tasks were handled in two weeks, leaving the last week for focusing on the ***readme.md***, ***FARID_VAHABZADA_SP1_CA_AUG23FT.txt*** and ***Jira.pdf*** files. This means however that coding was done during the weekends as well.

Timeline and Backlog

In the very beginning **Epic** creation was addressed. Here we tried to fetch the essence of what the web application must have to satisfy the customer' every need. The five major Epics created are:

1. Staff member out-of-office logging
2. Deliveries tracking
3. Current date and time
4. Company brand reflection
5. Navigation bar

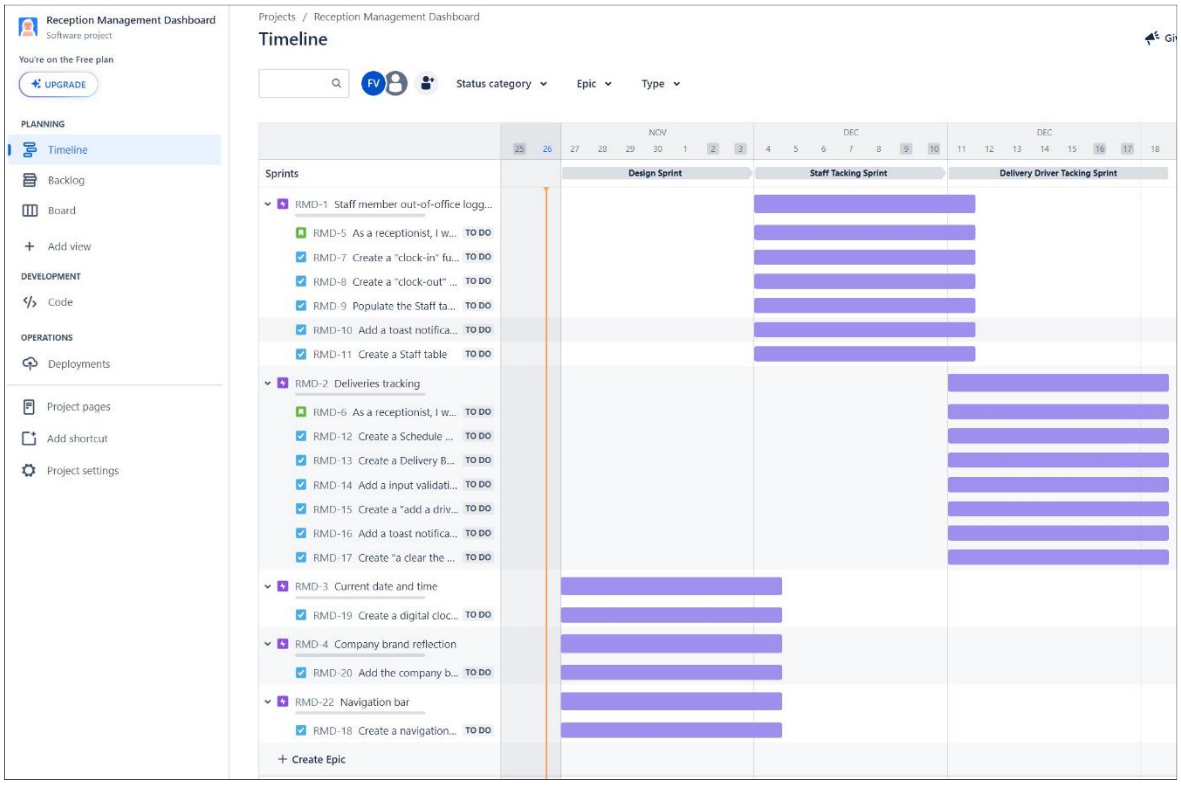
The order given above also reflects how essential every Epic is to get a final functioning web page. For example, even though navigation bar sounds more important than company brand's presence on the web page, it will not be functioning properly anyways and from the semester project instructions, the latter sounds more like a prerequisite for the customer than the former.

Later every Epic got assigned with specific **Tasks** and **Story**. In our case each Epic issue got only one Story, enough to clarify further user's demand from the relevant part of the application. Tasks were also enough in numbers, in many instances covering many subtasks in them.

Let's see from the figures below how the **Timeline** and **Backlog** looked, as a result of the changes made earlier. Note that **no story points** were used in this project.

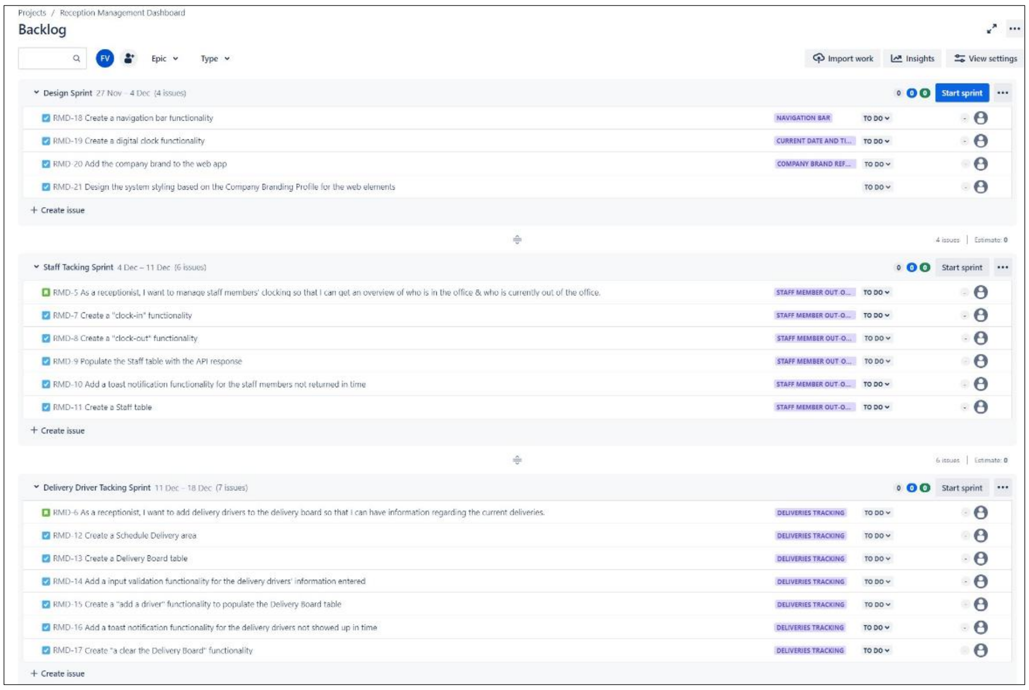
The quality of the screenshots taken is not the best, and in the Timeline figure, not the whole

Task names are displayed. In the Backlog figure and the figures followed by it, Task names can be read without any problems.



RMD-1 Staff member out-of-office logg...
RMD-5 As a receptionist, I w... TO DO
RMD-7 Create a "clock-in" fu... TO DO
RMD-8 Create a "clock-out" ... TO DO
RMD-9 Populate the Staff ta... TO DO
RMD-10 Add a toast notifica... TO DO
RMD-11 Create a Staff table TO DO
RMD-2 Deliveries tracking
RMD-6 As a receptionist, I w... TO DO
RMD-12 Create a Schedule ... TO DO
RMD-13 Create a Delivery B... TO DO
RMD-14 Add a input validati... TO DO
RMD-15 Create a "add a driv... TO DO
RMD-16 Add a toast notifica... TO DO
RMD-17 Create "a clear the ... TO DO
RMD-3 Current date and time
RMD-19 Create a digital cloc... TO DO
RMD-4 Company brand reflection
RMD-20 Add the company b... TO DO
RMD-22 Navigation bar
RMD-18 Create a navigation... TO DO
+ Create Epic

Figure 2. Timeline.



Design Sprint 27 Nov – 4 Dec (4 issues)

RMD-18 Create a navigation bar functionality NAVIGATION BAR TO DO
RMD-19 Create a digital clock functionality CURRENT DATE AND TI... TO DO
RMD-20 Add the company brand to the web app COMPANY BRAND REF... TO DO
RMD-21 Design the system styling based on the Company Branding Profile for the web elements TO DO
+ Create issue

4 issues | Estimate: 0

Staff Tacking Sprint 4 Dec – 11 Dec (6 issues)

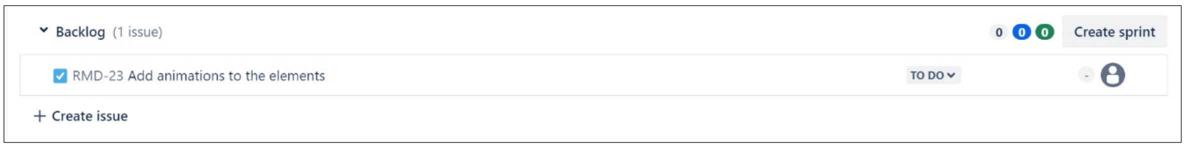
RMD-5 As a receptionist, I want to manage staff members' clocking so that I can get an overview of who is in the office & who is currently out of the office. STAFF MEMBER OUT-O... TO DO
RMD-7 Create a "clock-in" functionality STAFF MEMBER OUT-O... TO DO
RMD-8 Create a "clock-out" functionality STAFF MEMBER OUT-O... TO DO
RMD-9 Populate the Staff table with the API response STAFF MEMBER OUT-O... TO DO
RMD-10 Add a toast notification functionality for the staff members not returned in time STAFF MEMBER OUT-O... TO DO
RMD-11 Create a Staff table STAFF MEMBER OUT-O... TO DO
+ Create issue

6 issues | Estimate: 0

Delivery Driver Tacking Sprint 11 Dec – 18 Dec (7 issues)

RMD-6 As a receptionist, I want to add delivery drivers to the delivery board so that I can have information regarding the current deliveries. DELIVERIES TRACKING TO DO
RMD-12 Create a Schedule Delivery area DELIVERIES TRACKING TO DO
RMD-13 Create a Delivery Board table DELIVERIES TRACKING TO DO
RMD-14 Add a input validation functionality for the delivery drivers' information entered DELIVERIES TRACKING TO DO
RMD-15 Create a "add a driver" functionality to populate the Delivery Board table DELIVERIES TRACKING TO DO
RMD-16 Add a toast notification functionality for the delivery drivers not showed up in time DELIVERIES TRACKING TO DO
RMD-17 Create "a clear the Delivery Board" functionality DELIVERIES TRACKING TO DO
+ Create issue

Figure 3. Backlog.



Backlog (1 issue)

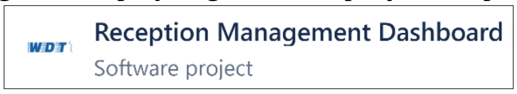
RMD-23 Add animations to the elements TO DO

+ Create issue

Figure 4. One last Task.

Later, one last Task was added to the Backlog as can be seen in the figure above. However, it was not assigned to any of the Epic issues.

And it is worth noting that our project got the company brand provided as its own icon.



Reception Management Dashboard
Software project

Figure 5. Jira project icon.

Board

As can be seen from the Timeline and Backlog figures, there exists three **Sprints** as a total, each lasting one week only. These Sprints are:

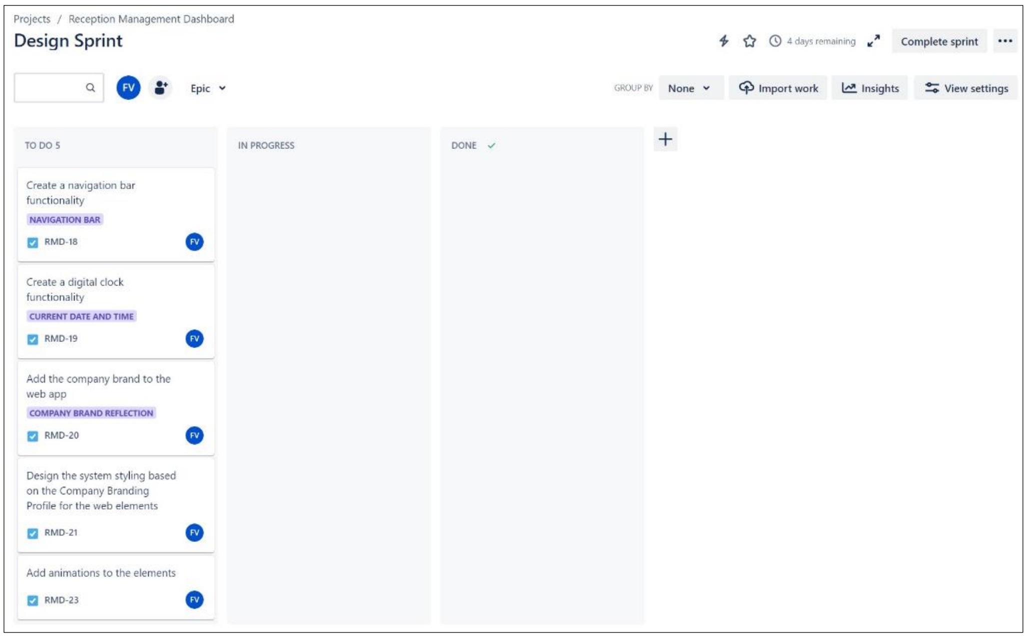
1. Design Sprint
2. Staff Tracking Sprint
3. Driver Tracking Sprint

It must be mentioned that the names of the last two were fixed after the Timeline and Backlog figures were taken.

In all these Sprints' snapshots, how the Tasks were progressed and when the Sprints were finished before the deadline can be found in **the X days remaining part**.

Design Sprint

After pressing the **Start sprint** button in the Backlog section, this Sprint was moved to the **Board** section. Where all the Tasks were assigned to Farid Vahabzada, as he was the only participant who was working on the project.



Projects / Reception Management Dashboard

Design Sprint

4 days remaining | Complete sprint

FV | Epic | GROUP BY None | Import work | Insights | View settings

TO DO 5

- Create a navigation bar functionality — NAVIGATION BAR — RMD-18 — FV
- Create a digital clock functionality — CURRENT DATE AND TIME — RMD-19 — FV
- Add the company brand to the web app — COMPANY BRAND REFLECTION — RMD-20 — FV
- Design the system styling based on the Company Branding Profile for the web elements — RMD-21 — FV
- Add animations to the elements — RMD-23 — FV

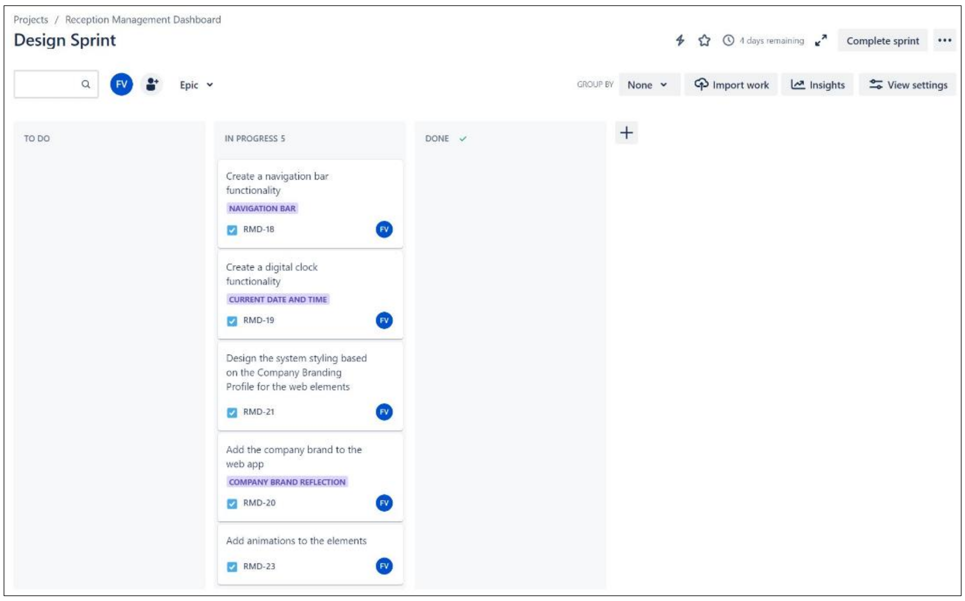
IN PROGRESS

DONE

Figure 6. Design Sprint To Do.

One remark on the last Task in Figure 4 above, which was not assigned to any of the Epic issues, was added to the Design Sprint.

All the Tasks were moved from the **To Do** column to the **In Progress** column as follows:


Projects / Reception Management Dashboard

Design Sprint

4 days remaining · Complete sprint

Epic · GROUP BY None · Import work · Insights · View settings

TO DO

IN PROGRESS 5

Create a navigation bar functionality
NAVIGATION BAR
RMD-18 FV

Create a digital clock functionality
CURRENT DATE AND TIME
RMD-19 FV

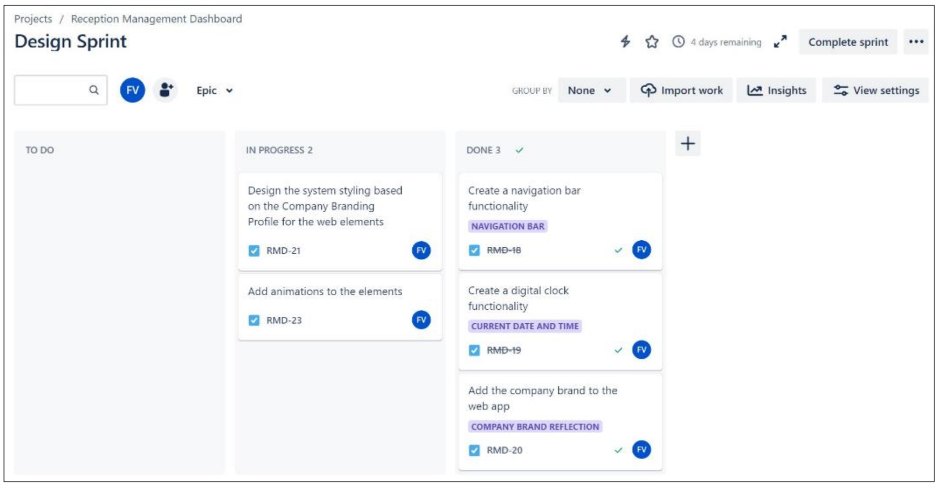
Design the system styling based on the Company Branding Profile for the web elements
RMD-21 FV

Add the company brand to the web app
COMPANY BRAND REFLECTION
RMD-20 FV

Add animations to the elements
RMD-23 FV

DONE

Figure 7. Design Sprint In Progress.


Projects / Reception Management Dashboard

Design Sprint

4 days remaining · Complete sprint

Epic · GROUP BY None · Import work · Insights · View settings

TO DO

IN PROGRESS 2

Design the system styling based on the Company Branding Profile for the web elements
RMD-21 FV

Add animations to the elements
RMD-23 FV

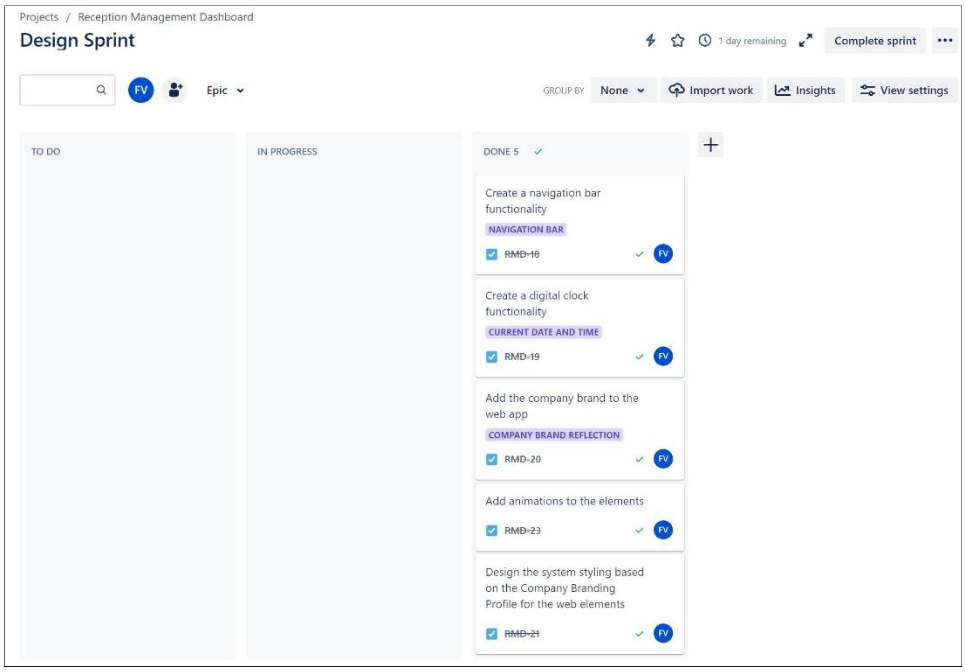
DONE 3

Create a navigation bar functionality
NAVIGATION BAR
~~RMD-18~~ FV

Create a digital clock functionality
CURRENT DATE AND TIME
~~RMD-19~~ FV

Add the company brand to the web app
COMPANY BRAND REFLECTION
RMD-20 FV

Figure 8. Design Sprint after some progress.


Projects / Reception Management Dashboard

Design Sprint

1 day remaining · Complete sprint

Epic · GROUP BY None · Import work · Insights · View settings

TO DO

IN PROGRESS

DONE 5

Create a navigation bar functionality
NAVIGATION BAR
~~RMD-18~~ FV

Create a digital clock functionality
CURRENT DATE AND TIME
~~RMD-19~~ FV

Add the company brand to the web app
COMPANY BRAND REFLECTION
RMD-20 FV

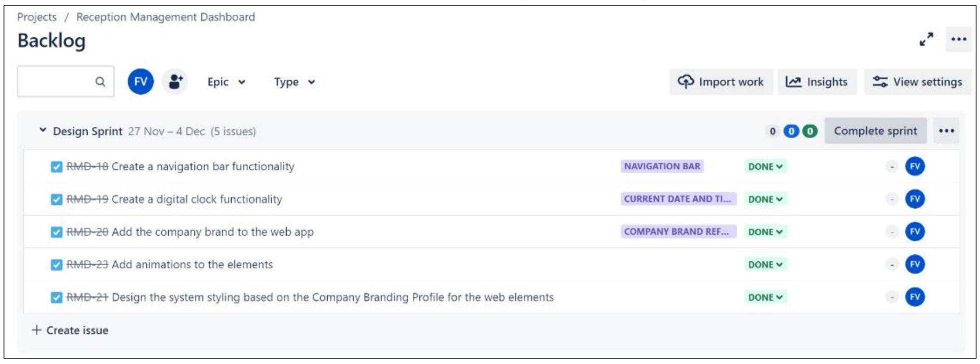
Add animations to the elements
~~RMD-23~~ FV

Design the system styling based on the Company Branding Profile for the web elements
~~RMD-21~~ FV

Figure 9. Design Sprint Done.

As the Tasks were accomplished, they were moved to the **Done** column accordingly.

This Sprint did not have any Story issues in it. However, here, most of the groundwork for the future was carried out. And it was finished before the planned Sprint deadline.



Projects / Reception Management Dashboard

Backlog

Design Sprint 27 Nov – 4 Dec (5 issues) 0 0 0 Complete sprint

- ~~RMD-18~~ Create a navigation bar functionality — NAVIGATION BAR — DONE — FV
- ~~RMD-19~~ Create a digital clock functionality — CURRENT DATE AND TI... — DONE — FV
- ~~RMD-20~~ Add the company brand to the web app — COMPANY BRAND REF... — DONE — FV
- ~~RMD-23~~ Add animations to the elements — DONE — FV
- ~~RMD-21~~ Design the system styling based on the Company Branding Profile for the web elements — DONE — FV

+ Create issue

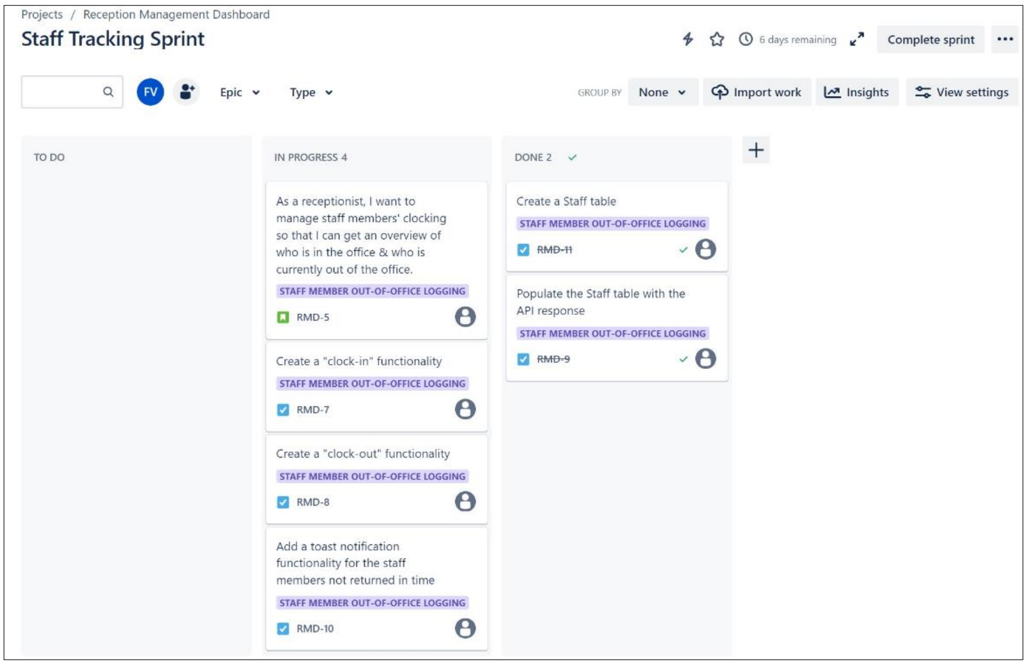
Figure 10. Complete sprint button.

After that, we can press the **Complete sprint** button from the Board or Backlog section. And then move the next Sprint to the Board section by pressing the Start sprint button.

Staff Tracking Sprint

When we started this Sprint, some of the Tasks added to it were partially or completely done in the previous Sprint. So, they were directly moved to the Done column.

We started this Sprint right after the previous one was finished, much before the planned Sprint start date.



Projects / Reception Management Dashboard

Staff Tracking Sprint — 6 days remaining — Complete sprint

GROUP BY None

TO DO

IN PROGRESS 4

- As a receptionist, I want to manage staff members' clocking so that I can get an overview of who is in the office & who is currently out of the office. — STAFF MEMBER OUT-OF-OFFICE LOGGING — RMD-5
- Create a "clock-in" functionality — STAFF MEMBER OUT-OF-OFFICE LOGGING — RMD-7
- Create a "clock-out" functionality — STAFF MEMBER OUT-OF-OFFICE LOGGING — RMD-8
- Add a toast notification functionality for the staff members not returned in time — STAFF MEMBER OUT-OF-OFFICE LOGGING — RMD-10

DONE 2 ✓

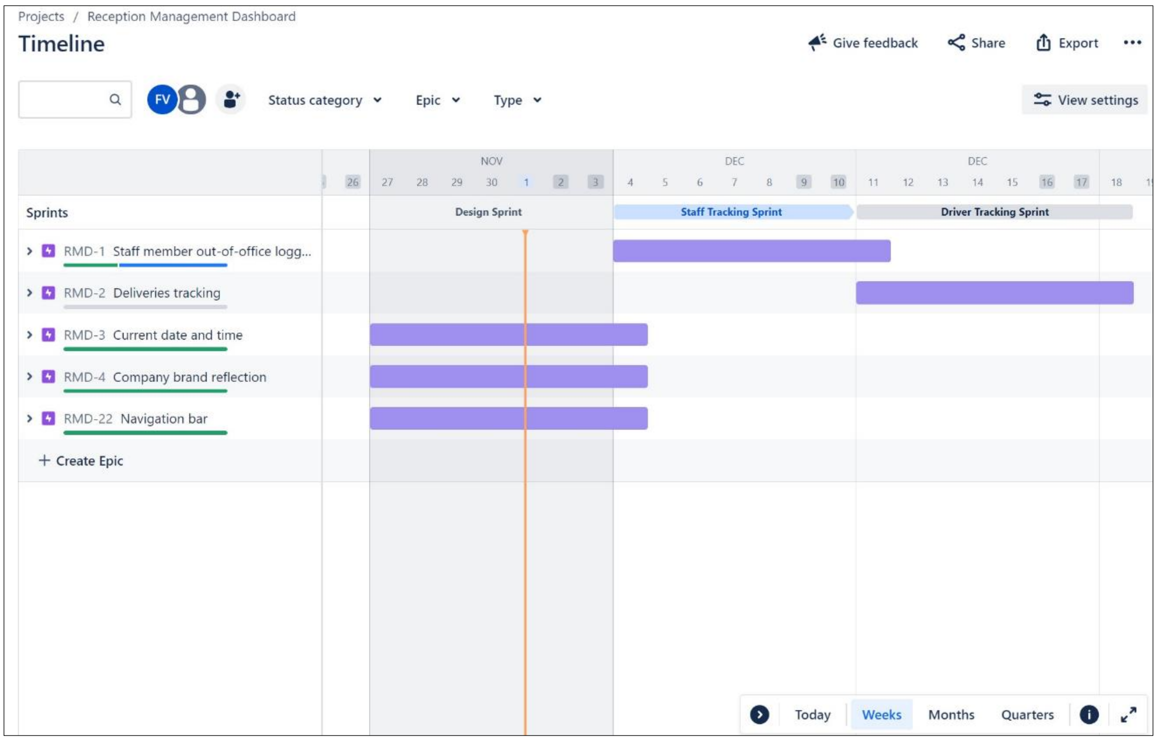
- Create a Staff table — STAFF MEMBER OUT-OF-OFFICE LOGGING — ~~RMD-11~~
- Populate the Staff table with the API response — STAFF MEMBER OUT-OF-OFFICE LOGGING — ~~RMD-9~~

Figure 11. Some tasks in Staff Tracking Sprint were finished.

One thing we should comment on is in this Sprint, we forgot to assign a participant to the issues, which was noticed only after the Sprint was completed.

All the issues we got on the Board belong to the Task type, except for the one which is the user Story type.

At this point, we can look at the Timeline and see the progress that was made in every Epic.

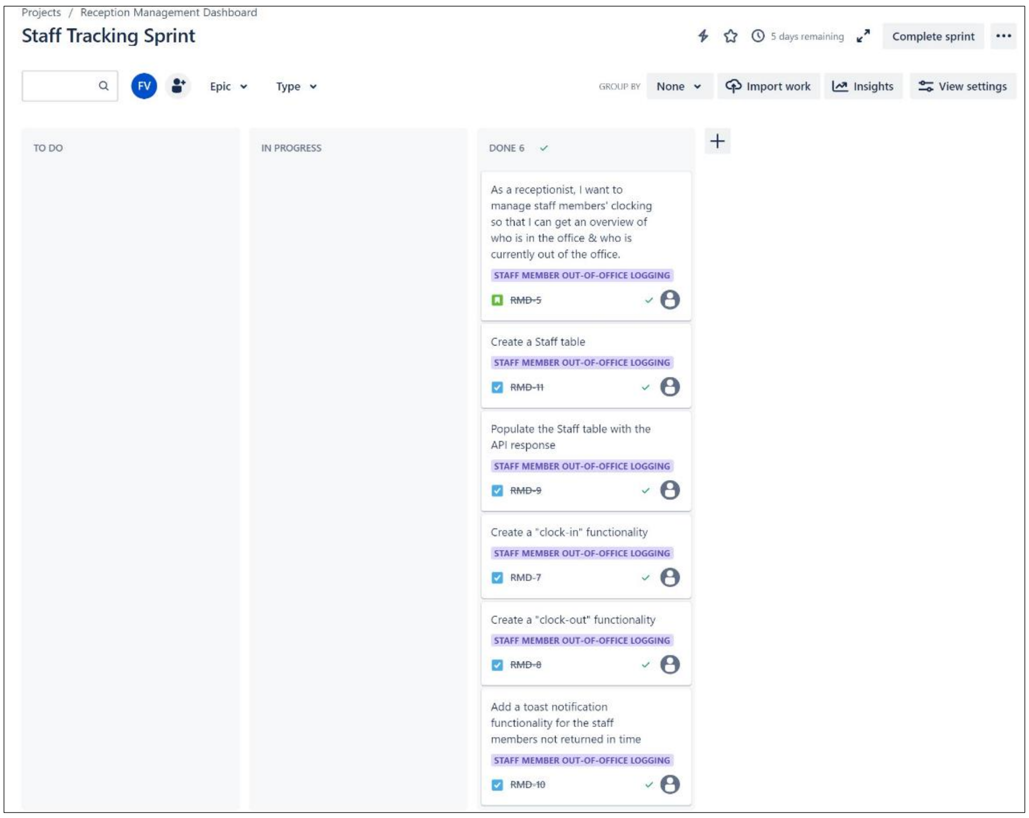

Projects / Reception Management Dashboard

Timeline

Sprints: Design Sprint, Staff Tracking Sprint, Driver Tracking Sprint

- RMD-1 Staff member out-of-office logg...
- RMD-2 Deliveries tracking
- RMD-3 Current date and time
- RMD-4 Company brand reflection
- RMD-22 Navigation bar
- + Create Epic

Figure 12. Current Staff Tracking Sprint is highlighted.


Projects / Reception Management Dashboard

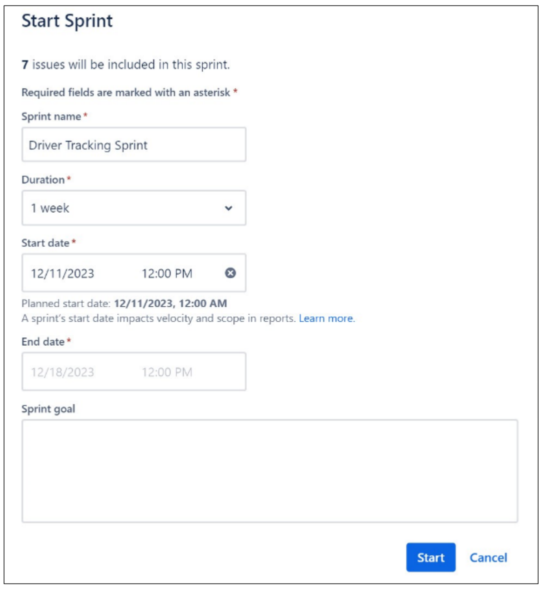
Staff Tracking Sprint

- As a receptionist, I want to manage staff members' clocking so that I can get an overview of who is in the office & who is currently out of the office. STAFF MEMBER OUT-OF-OFFICE LOGGING ~~RMD-5~~
- Create a Staff table STAFF MEMBER OUT-OF-OFFICE LOGGING ~~RMD-11~~
- Populate the Staff table with the API response STAFF MEMBER OUT-OF-OFFICE LOGGING ~~RMD-9~~
- Create a "clock-in" functionality STAFF MEMBER OUT-OF-OFFICE LOGGING RMD-7
- Create a "clock-out" functionality STAFF MEMBER OUT-OF-OFFICE LOGGING ~~RMD-8~~
- Add a toast notification functionality for the staff members not returned in time STAFF MEMBER OUT-OF-OFFICE LOGGING ~~RMD-10~~

Figure 13. Staff Tracking Sprint Done.

Driver Tracking Sprint

This Sprint got the greatest number of issues with six Tasks and one Story compared to the other two. Below, we will also be showing how a Sprint is started.



Start Sprint

7 issues will be included in this sprint.

Required fields are marked with an asterisk *

Sprint name *: Driver Tracking Sprint

Duration *: 1 week

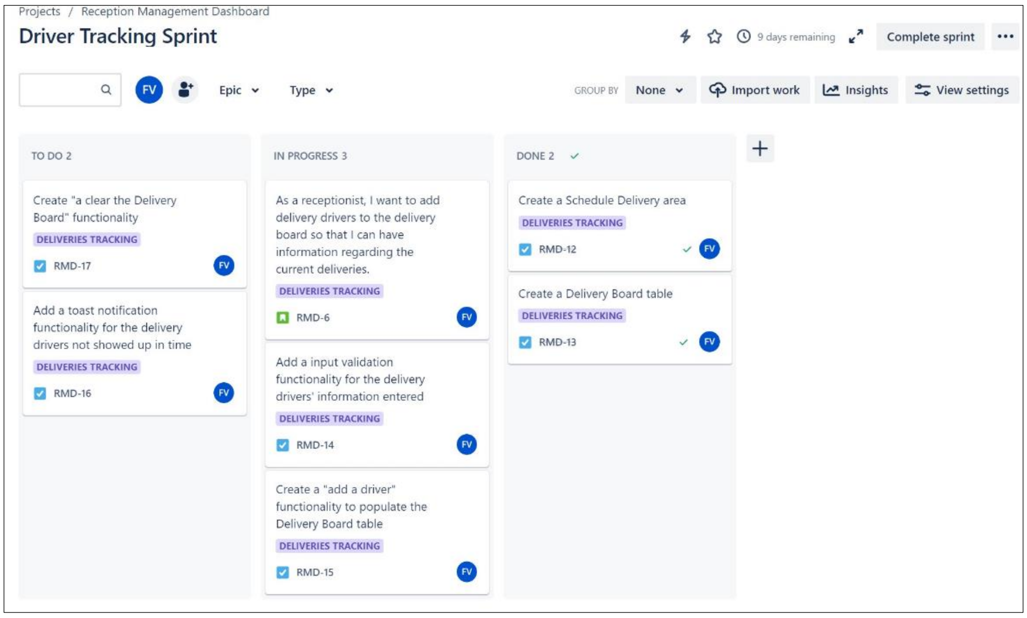
Start date *: 12/11/2023 12:00 PM

Planned start date: **12/11/2023, 12:00 AM**
A sprint's start date impacts velocity and scope in reports. Learn more.

End date *: 12/18/2023 12:00 PM

Sprint goal

Figure 14. Driver Tracking Sprint start window.

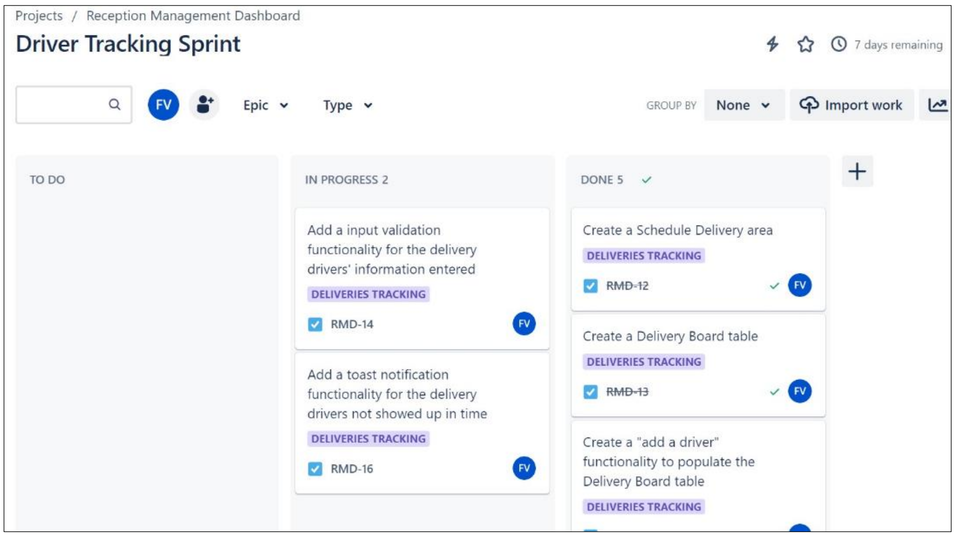


Projects / Reception Management Dashboard

Driver Tracking Sprint — 9 days remaining

TO DO 2	IN PROGRESS 3	DONE 2
Create "a clear the Delivery Board" functionality — DELIVERIES TRACKING — RMD-17	As a receptionist, I want to add delivery drivers to the delivery board so that I can have information regarding the current deliveries. — DELIVERIES TRACKING — RMD-6	Create a Schedule Delivery area — DELIVERIES TRACKING — RMD-12
Add a toast notification functionality for the delivery drivers not showed up in time — DELIVERIES TRACKING — RMD-16	Add a input validation functionality for the delivery drivers' information entered — DELIVERIES TRACKING — RMD-14	Create a Delivery Board table — DELIVERIES TRACKING — RMD-13
	Create a "add a driver" functionality to populate the Delivery Board table — DELIVERIES TRACKING — RMD-15	

Figure 15. Some tasks in Driver Tracking Sprint were finished.



Projects / Reception Management Dashboard

Driver Tracking Sprint — 7 days remaining

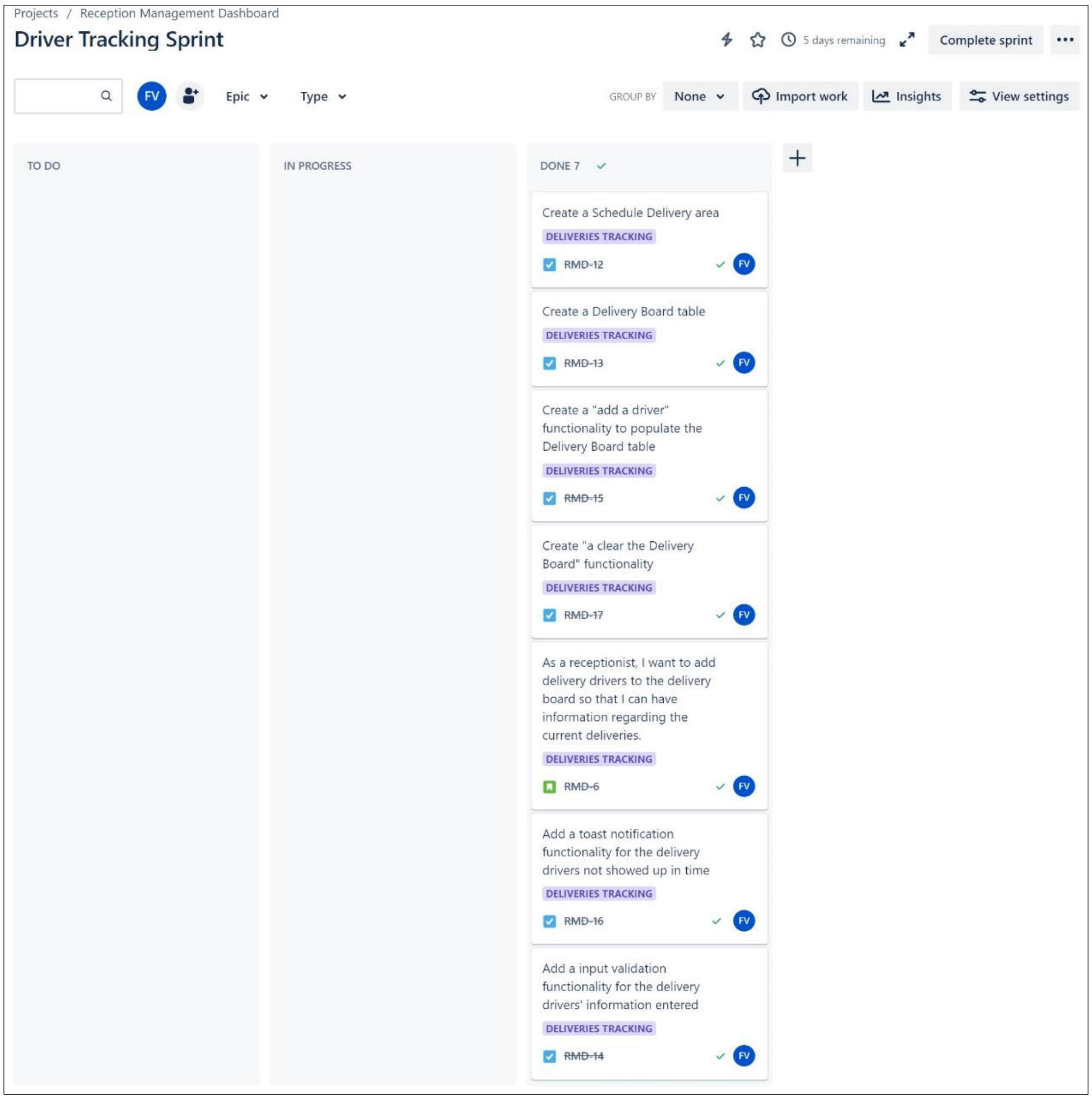
TO DO	IN PROGRESS 2	DONE 5
	Add a input validation functionality for the delivery drivers' information entered — DELIVERIES TRACKING — RMD-14	Create a Schedule Delivery area — DELIVERIES TRACKING — ~~RMD-12~~
	Add a toast notification functionality for the delivery drivers not showed up in time — DELIVERIES TRACKING — RMD-16	Create a Delivery Board table — DELIVERIES TRACKING — ~~RMD-13~~
		Create a "add a driver" functionality to populate the Delivery Board table — DELIVERIES TRACKING

Figure 16. The last two tasks in Driver Tracking Sprint.

The same thing happened here in the Driver Tracking Sprint as in the Staff Tracking Sprint, some of the Tasks were partially or completely done in the Design Sprint, and directly moved to the Done column.

Close to the end of this Sprint, only two Tasks were left in the In Progress column, one responsible for the validation and the other for the toast notification.

Five days before the deadline for the third Sprint, it was completed. However, checking the code files and adding new features took the whole weekend as well. These five days are according to the Jira software 5 days a week work arrangement.


Projects / Reception Management Dashboard

Driver Tracking Sprint

5 days remaining | Complete sprint

FV | Epic | Type | GROUP BY None | Import work | Insights | View settings

TO DO | IN PROGRESS | DONE 7

Create a Schedule Delivery area
DELIVERIES TRACKING
RMD-12 FV

Create a Delivery Board table
DELIVERIES TRACKING
RMD-13 FV

Create a "add a driver" functionality to populate the Delivery Board table
DELIVERIES TRACKING
~~RMD-15~~ FV

Create "a clear the Delivery Board" functionality
DELIVERIES TRACKING
RMD-17 FV

As a receptionist, I want to add delivery drivers to the delivery board so that I can have information regarding the current deliveries.
DELIVERIES TRACKING
RMD-6 FV

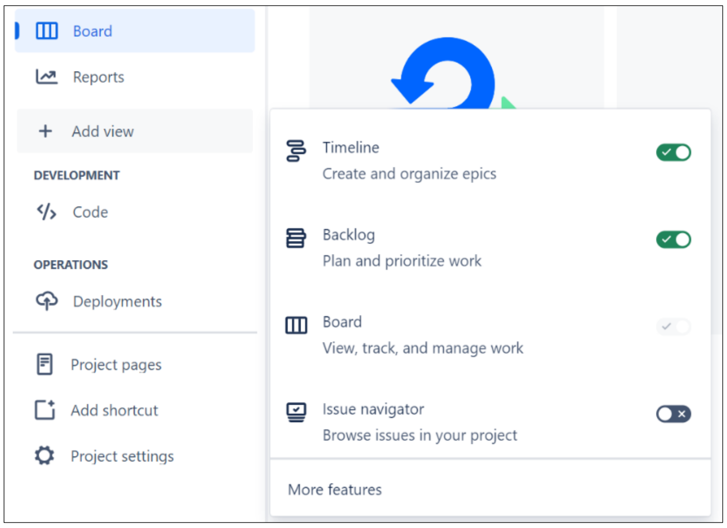
Add a toast notification functionality for the delivery drivers not showed up in time
DELIVERIES TRACKING
RMD-16 FV

Add a input validation functionality for the delivery drivers' information entered
DELIVERIES TRACKING
~~RMD-14~~ FV

Figure 17. Driver Tracking Sprint Done.

Final Accord

This project would not be complete without any graphs being supplied. For this purpose, we added **Reports** to our view on the left-hand side menu. We did it by clicking the **+ Add view** button below the Board section as shown in the figure below. And then clicked the **More features** option opened on the right-hand side menu. At this point, we got the Reports feature activated as can be seen.


Board
Reports
Add view
DEVELOPMENT
Code
OPERATIONS
Deployments
Project pages
Add shortcut
Project settings

Timeline
Create and organize epics

Backlog
Plan and prioritize work

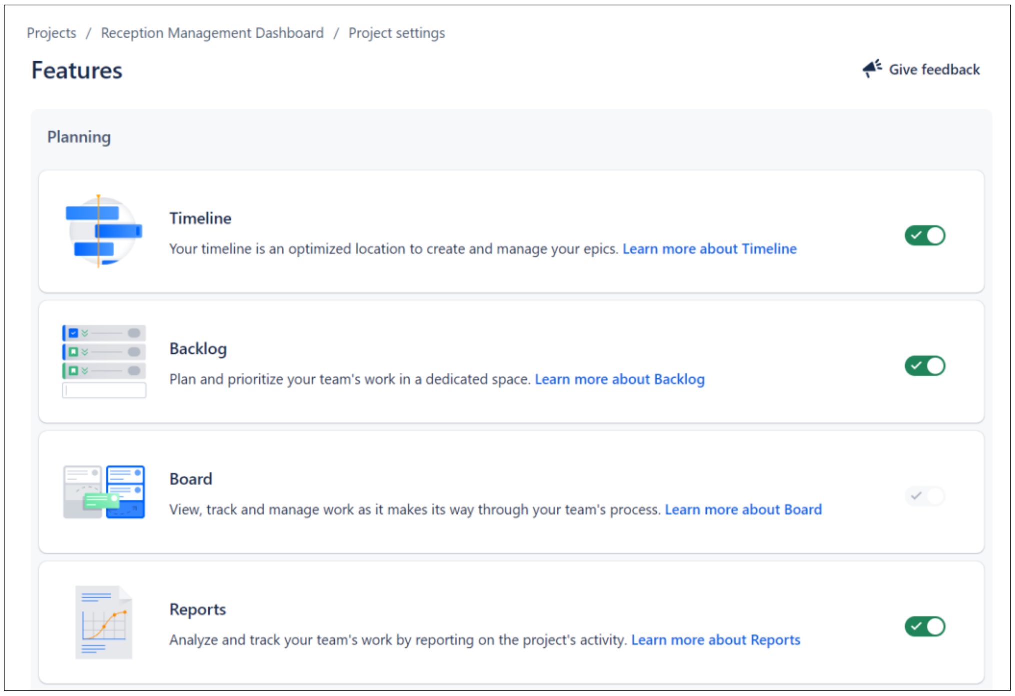
Board
View, track, and manage work

Issue navigator
Browse issues in your project

More features

Figure 18. + Add view button and More features optionality.

Right after that, a new window named **Features** will be opened, from there we can toggle the Reports feature on.


Projects / Reception Management Dashboard / Project settings

Features

Give feedback

Planning

Timeline
Your timeline is an optimized location to create and manage your epics. Learn more about Timeline

Backlog
Plan and prioritize your team's work in a dedicated space. Learn more about Backlog

Board
View, track and manage work as it makes its way through your team's process. Learn more about Board

Reports
Analyze and track your team's work by reporting on the project's activity. Learn more about Reports

Figure 19. Features window.

By pressing the Reports feature, we can get access to a variety of graphs available. As we did not use any story points. The only familiar graph that we could get was **Cumulative flow diagram**. From this diagram we can get crucial information about how the project development performance was based on the statuses of the issues and we can even detect possible bottlenecks with a glance. Here distance between To Do, In Progress and Done column lines describes how much time was needed to move from one column to another.

Additionally, below you can find the final Timeline screenshot right after the Driver Tracking Sprint was finished, showing how statuses of the Epics turned green, and the project reached its final objective on the 8th of December.

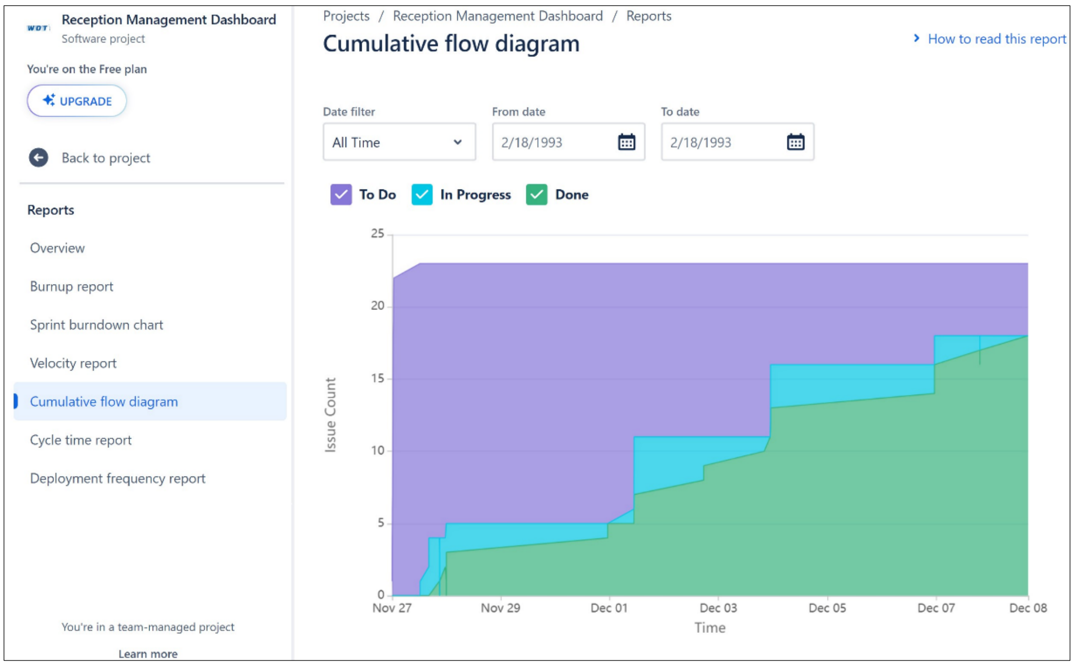


Figure 20. Cumulative flow diagram.

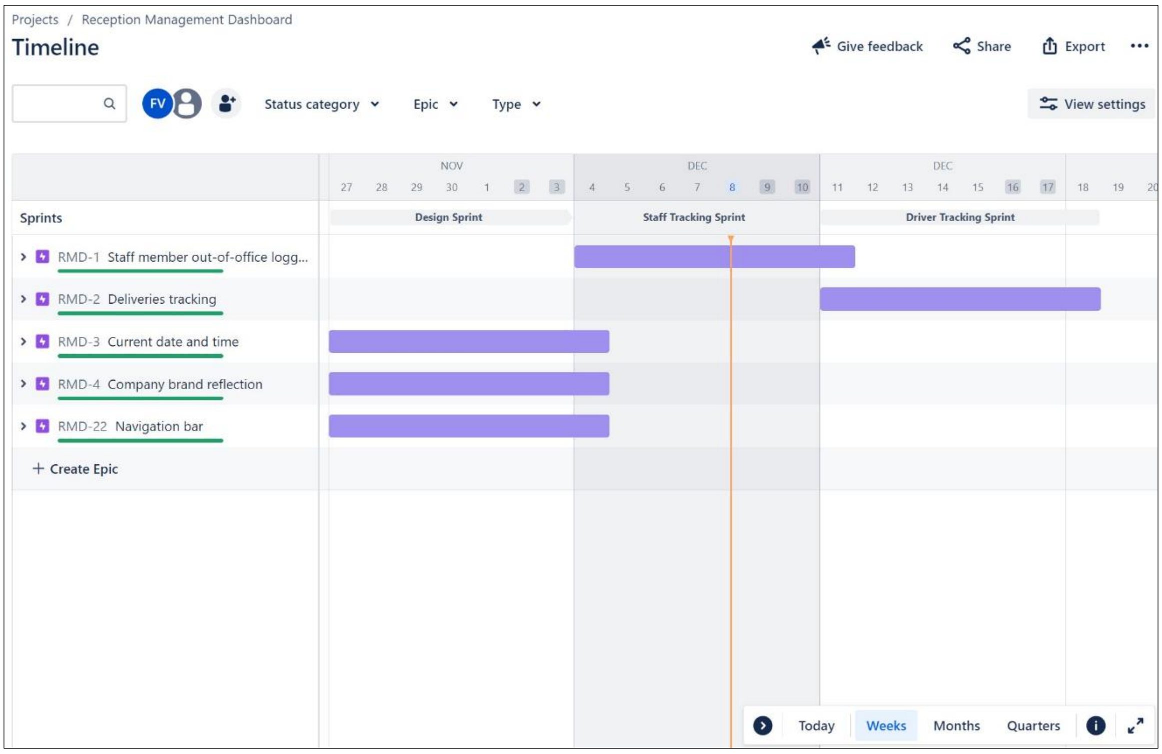


Figure 21. Timeline of the finished project.

Reflection Report

We studied meticulously the instructions, mock-up layout and demonstration video.

The code files checked for the syntax, abundant usage of spaces, placement of special characters, obedience to the tree structure alignment/indentation and more. Also, grammar check was performed. Name conventions were followed even it was late in the development phase.

The animations were reproduced. Starting with Staff table, both buttons changing sizes simultaneously and text color on hover, rounded photos, and ending with Navbar items fading or disabling with no cursor changing. Even “Vehicle Type” option was added to avoid incorrect driver instances.

The only difference would be how selection functions, one row can only be selected at the time, then not deselected anymore. It can cause a bad user experience but can also avoid undesirable outcomes.

Alert or error messages have proper font boldness, underlining or colors. Even a bell icon was added.

Dashboard menu and brand logo got links to the homepage, and they can be tabbed. The navbar size was reduced to match the ratio of it to the main page container as in the video.

Font for all the text elements were followed, even for alerts and errors. But for the input boxes it could not be verified. Note that icons needed bootstrap-icons font family.

External links, used for favicon in the form of the logo with removed background, changed background color shade in tables (HEX to gradient) and changed text opacity in navbar (HEX to RGBA) on hover, are provided in the ***readme.md*** file.

API response was automatically converted to JS object by the **ajax()** method. The URL of the call was tailored to our needs after thorough documentation study, this link can be found in the ***readme.md*** file as well.

Homepage container got rounded corners, same as the tables. Even the Delivery Board (**DB**) table header gets it when no rows are present.

Row selection color was adopted from the positive button. IN and OUT buttons got different kinds of hovering reactions than ADD and CLEAR buttons.

Some alerts don’t allow any outside clicks making sure user understands the message, while others do.

Five staff members' uniqueness was discussed with the teacher, and “**No need** to check it for the API.” received as an answer. Some ID photos were repeating from our observations, the other attributes never though.

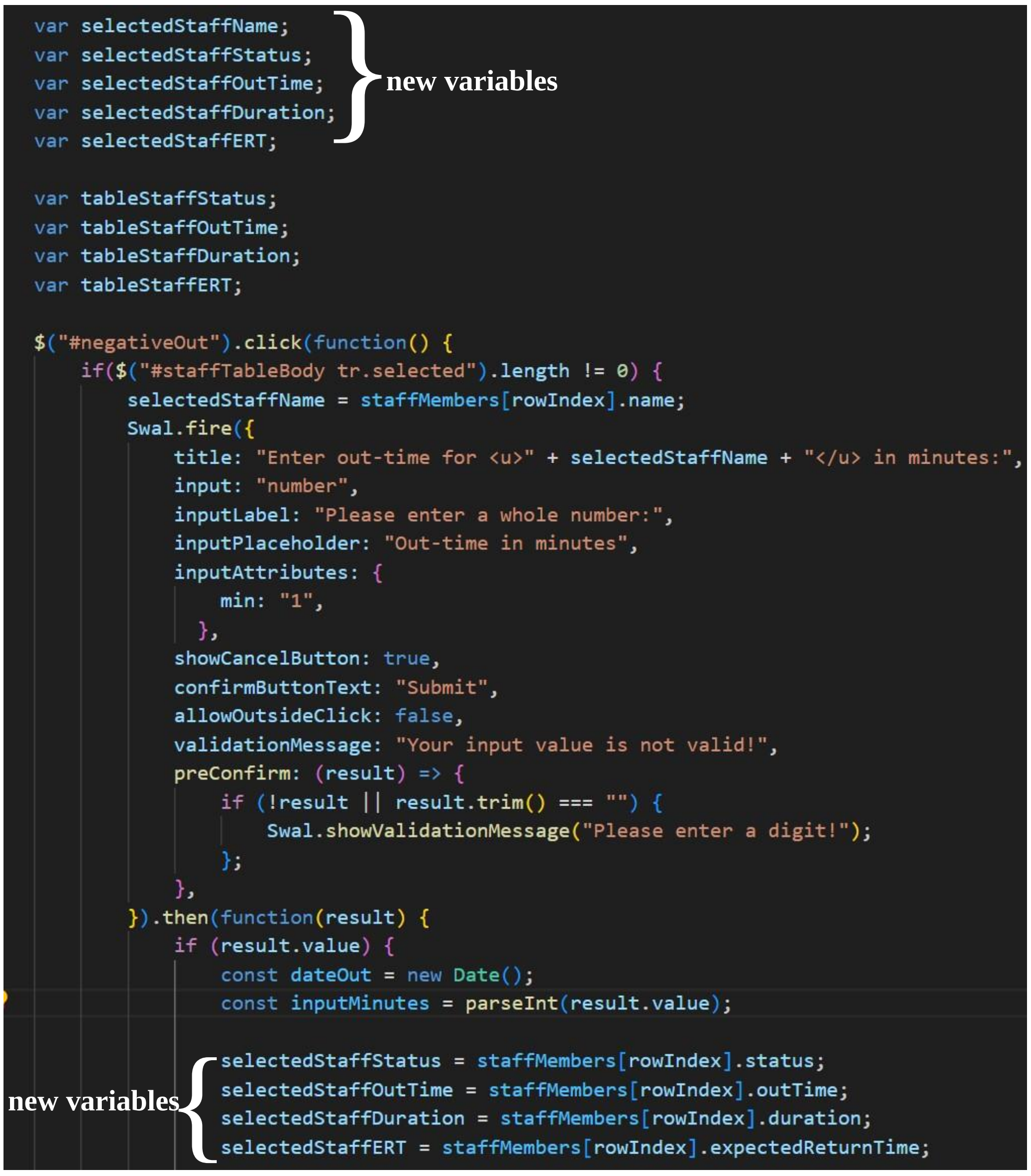
When prompted for absence minutes, we placed range validation with minimum 1 and maximum 720 minutes (12 hours). There’s a range arrow for user’s disposal, and other than digits are invalid.

Fading effect for clocking in or out was added to the Staff table (**ST**), but a problem occurred with table's inner borders and rounded corners disappearing with cell contents and appearing again. This bug was fixed by placing table elements in a container, then changing border properties from there, as the <table> tag was not responding to any changes.

Input boxes in the Schedule Delivery (**SD**) table fixated vertically to the top position, as with validation messages being added underneath them, they were moving around uncontrollably.

First letters in the Name, Surname and Delivery Address entries are capitalized automatically in both SD and DB tables, but this change just affects how the input value looks, nothing more.

Because how the Inheritance works, every time a new driver is added to the DB table, we use **randomUsers[0]** initially to create a new Driver class object. Everything is done through the objects on the tables. While developing, testing was conducted via logs. In the figure below while shortening the name conventions, attempts were made to create new variables and change the tables through them. Logging discovered that objects were not changing in the process.



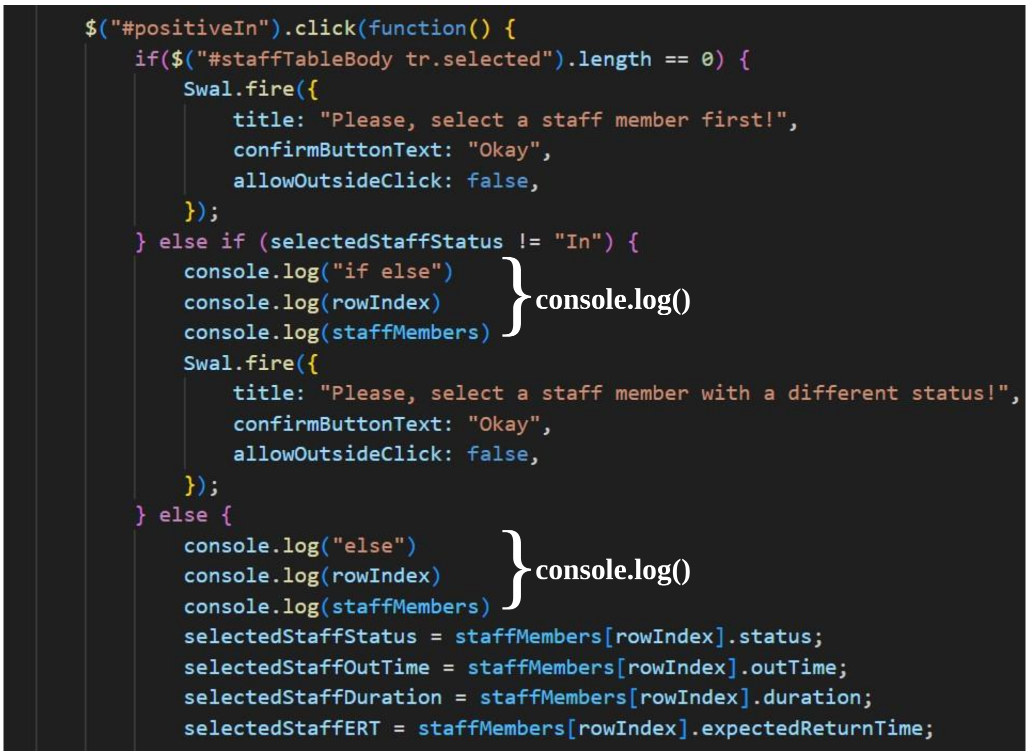
```
var selectedStaffName;
var selectedStaffStatus;
var selectedStaffOutTime;
var selectedStaffDuration;
var selectedStaffERT;

var tableStaffStatus;
var tableStaffOutTime;
var tableStaffDuration;
var tableStaffERT;

$("#negativeOut").click(function() {
    if($("#staffTableBody tr.selected").length != 0) {
        selectedStaffName = staffMembers[rowIndex].name;
        Swal.fire({
            title: "Enter out-time for <u>" + selectedStaffName + "</u> in minutes:",
            input: "number",
            inputLabel: "Please enter a whole number:",
            inputPlaceholder: "Out-time in minutes",
            inputAttributes: {
                min: "1",
            },
            showCancelButton: true,
            confirmButtonText: "Submit",
            allowOutsideClick: false,
            validationMessage: "Your input value is not valid!",
            preConfirm: (result) => {
                if (!result || result.trim() === "") {
                    Swal.showValidationMessage("Please enter a digit!");
                };
            },
        }).then(function(result) {
            if (result.value) {
                const dateOut = new Date();
                const inputMinutes = parseInt(result.value);

                selectedStaffStatus = staffMembers[rowIndex].status;
                selectedStaffOutTime = staffMembers[rowIndex].outTime;
                selectedStaffDuration = staffMembers[rowIndex].duration;
                selectedStaffERT = staffMembers[rowIndex].expectedReturnTime;
```

a) new variables declared and assigned



```
$("#positiveIn").click(function() {
    if($("#staffTableBody tr.selected").length == 0) {
        Swal.fire({
            title: "Please, select a staff member first!",
            confirmButtonText: "Okay",
            allowOutsideClick: false,
        });
    } else if (selectedStaffStatus != "In") {
        console.log("if else")
        console.log(rowIndex)
        console.log(staffMembers)
        Swal.fire({
            title: "Please, select a staff member with a different status!",
            confirmButtonText: "Okay",
            allowOutsideClick: false,
        });
    } else {
        console.log("else")
        console.log(rowIndex)
        console.log(staffMembers)
        selectedStaffStatus = staffMembers[rowIndex].status;
        selectedStaffOutTime = staffMembers[rowIndex].outTime;
        selectedStaffDuration = staffMembers[rowIndex].duration;
        selectedStaffERT = staffMembers[rowIndex].expectedReturnTime;
```

b) usage of logging to detect changes made

Figure 22. Testing if changes are registered in objects.

In the SD table, validation is only done after all the input boxes are filled and the ADD button clicked. When this happens, if any invalid entries are present then appropriate error messages are shown. There are new <span> tags added in some of the errors, but the styling used for this tag in the HTML file won't be applied here as new tags are added after the page loaded.

We could just use Bootstrap classes and restrict what and how many of the characters user types, but we selected a different approach. The Name and Surname inputs take only letters and some special characters, with at least two letters present. The Delivery Address input validation was done in the best possible way without a need for third party plugins. The Telephone field only includes digits as in the video, but a minimum of eight digits are required. The Return Time and Vehicle validation is there if nothing was added, the format is as in the video. Because how the code is written all the double spaces are trimmed automatically.

If all the fields get valid entries and the ADD button is pressed then a new driver is added to the DB table, and all the input values in the SD table are cleared.

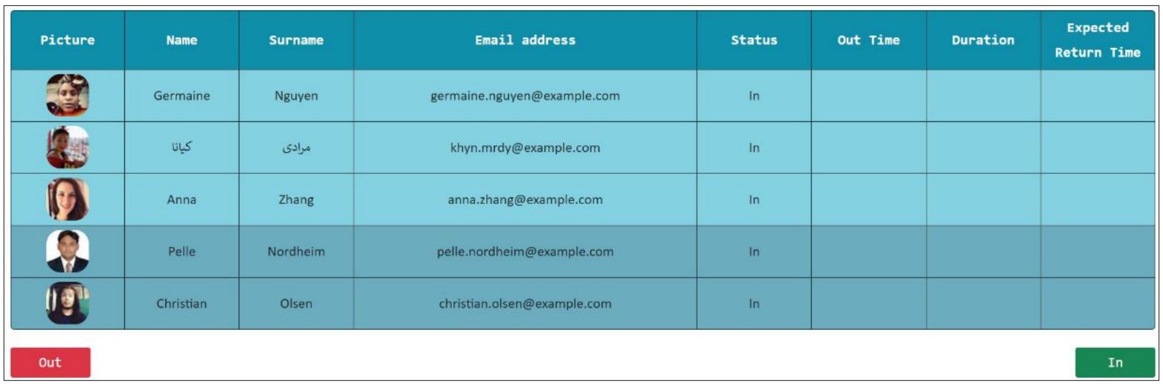
An issue when a new toast was appearing, it was bringing back the old toasts with it. This was fixed. We tried to use all available space under the page name, and we didn't limit the number of toasts that can be received.

For late driver toasts, the Return Time could be selected the next day, and midnight brought some challenges with it, though revised code fixed these problems.

Toasts will not show if the row data is changed or deleted. However, seconds are considered in the DB toasts, this was not considered in the ST toasts because the way code was written.

The DB table brought a new dynamic to the game as adding and removing rows affected selecting and highlighting process. A completely new approach was taken without a need for row indices, which proved to be shorter and more effective. Wrapping of this table's content was applied for the long input values. Vehicle icon size increased for a better UX. Uniqueness test was created so no two identical instances can exist together, which is also case insensitive. No name was given for the function removing driver from the DB table, so it's named appropriately based on the CLEAR button.

One bug found in all the tables, which was not fixed as it wasn't causing any functionality issues, was connected to the highlighted rows. The root of the issue is assumed to be cursor placement when the page is loading. As a result, one row keeps highlighted all the time. The issue is usually fixed by refreshing the page, and it was left for the next version of the app to resolve.



Picture	Name	Surname	Email address	Status	Out Time	Duration	Expected Return Time
	Germaine	Nguyen	germaine.nguyen@example.com	In			
	کیانا	مرادی	khyn.mrdy@example.com	In			
	Anna	Zhang	anna.zhang@example.com	In			
	Pelle	Nordheim	pelle.nordheim@example.com	In			
	Christian	Olsen	christian.olsen@example.com	In			

Figure 23. Highlighting bug.

Another problem was related to the class methods as they were originally designed as separate functions. This was noticed only after everything was code reviewed. They were eventually placed inside the relevant classes, but commented parts were left behind as a reminder of how it looked before.

One friendly disclaimer for users at this point would be when page reloads all the info is lost after.

Our web application's own demonstration video was created and placed in the Documentation folder.

References

Atlassian (no date) *Jira*, *Atlassian.com*.
Available at: https://www.atlassian.com/software/jira (Accessed: December 12, 2023).

Favicon Generator for perfect icons on all browsers (no date), *Realfavicongenerator.net*.
Available at: https://realfavicongenerator.net/ (Accessed: December 12, 2023).

Hex to RGBA (no date) *Rgbacolorpicker.com*.
Available at: https://rgbacolorpicker.com/hex-to-rgba (Accessed: December 12, 2023).

Kaleido AI (no date) *Remove background from image*.
Available at: https://www.remove.bg (Accessed: December 12, 2023).

MDN Web Docs (no date) *MDN Web Docs*.
Available at: https://developer.mozilla.org (Accessed: December 12, 2023).

Random user generator (no date) *Randomuser.me*.
Available at: https://randomuser.me/ (Accessed: December 12, 2023).

W3Schools online web tutorials (no date) *W3schools.com*.
Available at: https://www.w3schools.com/ (Accessed: December 12, 2023).

Gradient generator (No date) *Colordesigner.io*.
Available at: https://colordesigner.io/gradient-generator (Accessed: December 12, 2023).