Résumé

Ce rapport a été rédigé dans le cadre de notre projet de fin d'études pour obtenir le diplôme national en licence informatique à l'Institut Supérieur d'Informatique et de Mathématiques de Monastir. Le projet a été réalisé chez Mobelite dans le but de concevoir et développer une application solide de visualisation et de génération de rapports en utilisant la plateforme Forge d'Atlassian.

Mots-clés: Rapports, Visualisation de données, Journal des travaux, Temps passé, Atlassian Jira, Forge, D3.js, Node.js, React.js.

Abstract

This report has been produced as our end of studies project to obtain the national bachelor in Computer Science at the Higher Institute of Computer Science and Mathematics of Monastir. The project was carried out at Mobelite with the aim of designing and developing a robust application for visualization and report generation using the Atlassian Forge platform.

Keywords: Reporting, Work logs, Time Sheets, Data Visualization, Atlassian, Forge, D3.js, Node.js, React.js.

Dedication

With love and gratitude, I dedicate this work to my parents, **Fakheur** and **Saadia**. Your unwavering support and encouragement have made my journey possible. I am truly grateful for your love and guidance.

I also dedicate this report to my dear brother, **Mohamed Ali**, and my dear sister, **Oula**. You have always been there for me, through thick and thin. I am so lucky to have you in my life.

To my beloved uncle **Foued**, I dedicate this work with all my love and admiration. Your presence in my life means the world to me, and I am grateful for the immense love and care you have always shown me.

I could not have completed this report without the love and support of my family and friends. Thank you for everything.

[illegible]

Dedication

I would like to dedicate this work to all the individuals who have supported and motivated me throughout this journey.

To my beloved parents, and my entire family; your unwavering love, encouragement, and sacrifices have been the foundation of my success. Your belief in me has fueled my determination to overcome challenges and strive for excellence.

To my closest friends, and all those who have unwaveringly supported and believed in me throughout this journey.

I am grateful for the opportunities, lessons, and growth that this project has provided. This dedication is a tribute to all those who have played a part in shaping me into the person I am today.

Farah Nelali

Acknowledgment

We extend our sincerest gratitude to the teaching team at ISIMM, our professional counselors, and Professor **Mr. Malek BEN SALEM** for his invaluable guidance and support throughout the development of this project. His expertise and dedication have been instrumental in shaping our knowledge and skills in the field of computer science. We are honored to have had the opportunity to work under his supervision, and we are deeply grateful for his mentorship and constructive feedback.

Furthermore, we would like to express our profound appreciation to **Mr. Mamoun BOUSSIDA** for his unwavering commitment and mentorship during the internship at "Mobelite." His expertise, professionalism, and trust in our abilities have greatly contributed to our personal and professional growth. We are truly grateful for the knowledge and experiences he shared, and we consider ourselves fortunate to have had him as our supervisor.
Lastly, we would like to extend our sincere thanks to the esteemed members of the jury for their time and expertise in evaluating our project. Their evaluation and feedback are invaluable to our academic journey. We are deeply grateful for their commitment to academic excellence and their contributions to the success of our project.

TABLE OF CONTENTS

List of Figures

List of Tables

ACRONYMS

JFM	: **J**ira **f**or **M**obelite
LLC	: **L**imited **L**iability **C**ompany
KPI	: **K**ey **P**erformance **I**ndicator
CLI	: **C**ommand **L**ine **I**nterface
FaaS	: **F**unction-**a**s-**a**-**S**ervice
REST	: **Re**presentational **S**tate **T**ransfer
API	: **A**pplication **P**rogramming **I**nterface
HTTP	: **H**yper**t**ext **T**ransfer **P**rotocol
UI/UX	: **U**ser **I**nterface/**U**ser **E**xperience
AWS	: **A**mazon **W**eb **S**ervices

General Introduction

In today's fast-paced business environment, efficient project management is crucial for organizational success. Like many enterprises, Mobelite recognizes the importance of effective project and task management to remain competitive. To address this need, we created the “Jira for Mobelite” (JFM) application, a cloud-based project management tool specifically tailored for Mobelite.

JFM application empowers the Mobelite team by providing them with powerful features to visualize project data, generate customized reports, and leverage real-time data visualization capabilities. These functionalities enable teams and their managers to proactively stay on track with the done, current and future tasks, identify potential bottlenecks, and achieve timely project delivery.

Our end of studies project covers our development cycle for the JFM application, including the knowledge and insights we gained throughout the process.

This report provides a comprehensive overview of our contributions and achievements, structured into four distinct chapters. The first chapter, **“Preliminary study”**, is dedicated to presenting the general context and the emerging idea of the project. Then, we study the different existing solutions and their functionalities. We also made a detailed study of the work methodology at the end of this chapter. The second chapter, **“Project Management with Agile Scrum”**, will be dedicated to the practical implementation of the Agile Scrum project management methodology, with a focus on the used technologies. The third and fourth chapters will focus on the analysis, specification of requirements, and the implementation of the five sprints.

Lastly, we conclude this report with a general conclusion which summarizes these main points.

CHAPTER 1 PRELIMINARY STUDY

1.1. Introduction

This section is dedicated to being an overview of the company, introducing the problems and exploring existing solutions, presenting insights into the demanded needs, and describing the methodologies we will use to achieve and implement the product.

1.2. General context

This work is the graduation project to obtain the national bachelor's degree in computer science from the Higher Institute of Computer Science and Mathematics of Monastir belonging to Monastir University. The internship was carried out at the ease of Mobelite, the hosting company presented in the next section.

1.3. Hosting Company

"Mobelite" is an LLC company that specializes in the development of mobile and web applications and platforms. It offers a range of services, including UI/UX design, development of e-commerce solutions, and digital marketing. Figure 1.1 represents Mobelite's logo.

mobelite

Figure 1.1 Mobelite Logo

1.4. Emerging idea

Our project's main idea is to develop a plugin/application that can be integrated into Mobelite's Jira cloud platform. This plugin will provide a powerful tool for collecting, aggregating, and displaying valuable information related to a specific set of KPIs that are tailored to Mobelite's specific business objectives. By leveraging this plugin, Mobelite's employees will be able to track project performance in real-time, identify issues and bottlenecks early on, and make wiser and informed decisions to ensure that projects stay on track and on budget. Ultimately, this will help improve project's efficiency and productivity, leading to better outcomes for the company and its clients.

1.5. Market Study

The market research aims to deepen the analysis of innovative aspects of a project, while the project is being developed, to prepare for its implementation. This may involve studying the relevance, feasibility, or sustainability of the envisioned project. By understanding the market landscape, we can identify opportunities for our proposed solution to address specific needs and challenges.

1.5.1. Available solutions and analysis

The study of the existing is intended to enrich the idea of our project. It helps us to learn from related apps and identify the main features of our application and respect certain restrictions. In this part, we will outline some samples of Jira apps like our project and discuss their pros and cons. Then we will end up by mentioning our solution.

1.5.1.1. "Tempo Timesheet" plugin

"Tempo Timesheet" is primarily focused on time tracking and reporting, allowing users to log time against Jira issues and track how much time has been spent on each task. It provides a range of reporting features that help users generate time-based reports to aid with project management, billing, and resource allocation. Figure 1.2 represents the logo of "Tempo Timesheet" plugin.

Figure 1.2 "Tempo Timesheet" Solution

Strengths:

- It is accessible and easy to use and provides a simple and intuitive interface for logging time against Jira issues.
- Offers a range of reporting features to help teams analyze work logs data and generate time-based reports.
- Integrates seamlessly with Jira, making it easy to track time alongside other project management tasks.

Weaknesses:

- Can be expensive for larger teams or organizations.
- May not be as customizable as other time tracking tools, limiting flexibility for certain use cases.

1.5.1.2. "Worklogs - Time Tracking and Reports" plugin

"Worklogs - Time Tracking and Reports" is a Jira Atlassian plugin that enables users to track time spent on different Jira issues and generate detailed timesheets and reports. With Worklogs, users can easily log their time against individual Jira issues or across multiple issues, and then generate reports to analyze their productivity and billable hours. Figure 1.3 represents the logo of “Worklogs - Time Tracking and Reports”.

Figure 1.3 “Worklogs - Time Tracking and Reports” Solution

Strengths:

- Increased productivity: “Worklogs” can help increase productivity by providing a clear view of the time devoted to various tasks. This information can assist individuals and teams in identifying areas where they can improve their effectiveness and make better use of their time.
- Accurate Billing and Invoicing: For businesses that charge for their services based on time spent, “Worklogs” can help ensure accurate billing and invoicing. This can help prevent disputes and ensure that companies are compensated fairly for their work.
- Performance evaluation: “Worklogs” can help managers evaluate employee performance based on their productivity, efficiency, and ability to meet deadlines. This information can be used to identify areas where employees need improvement and to recognize top performers.

Weaknesses:

- Time-consuming: “Worklogs” can be time-consuming, especially if they require manual entry. This can be a burden for employees who already have a lot on their plate.
- Inaccurate data: “Worklogs” rely on accurate data entry, and if data is entered incorrectly or incomplete, it can lead to inaccurate reporting. This can lead to misunderstandings, disputes, and mistrust.
- Privacy concerns: “Worklogs” may raise privacy concerns, especially if they are used to monitor employee activity. Employers need to be transparent about how they use worklogs and ensure that they respect employee privacy.

1.5.2. Critics of existing solutions

To provide a comprehensive understanding of the existing plugins, we conducted a comparative study. This comparison focused on the needs and functionalities we aim to develop in our Jira application. The results are summarized in Table 1.1, which provides a general analysis of the existing plugins.

Table 1.1 General Analysis of the Existing plugins

Aspect	Tempo Timesheets	Worklogs - Time Tracking and Reports
Automation/Scripting	Yes	Yes
Time tracking	Yes	Yes
Reporting	Yes	Yes
Integration	Yes	Yes
Learning curve	Moderate	Easy
Cost	Paid	Free

The explanation provides insights into each aspect, highlighting the similarities and differences between the two platforms:

- Automation/Scripting: Both "Tempo Timesheet" and "Worklogs - Time Tracking and Reports" offer automation and scripting capabilities to automate repetitive tasks and integrate with other tools.
- Time tracking: Both "Tempo Timesheet" and "Worklogs - Time Tracking and Reports" offer time tracking functionality to track time spent on tasks and projects.
- Reporting: Both "Tempo Timesheet" and "Worklogs - Time Tracking and Reports" offer reporting functionality to generate reports and visualize data.
- Integration: Both "Tempo Timesheet" and "Worklogs - Time Tracking and Reports" offer integration with other tools, such as JIRA and Trello.
- Learning curve: "Tempo Timesheet" has a moderate learning curve due to its comprehensive feature set, whereas "Worklogs - Time Tracking and Reports" has an easy learning curve due to its simple interface and limited features.
- Cost: "Tempo Timesheet" is a paid tool, whereas "Worklogs - Time Tracking and Reports" is a free tool.

1.6. Proposed Solution

After conducting a thorough market study of "Tempo" and "Worklogs", we have developed a tailored solution that combines the best features of both tools. Our proposed solution is a comprehensive and user-friendly application hosted within Jira. It leverages the power of Function-as-a-Service (FaaS) serverless architecture to deliver an efficient and scalable work logs reporting and visualization platform. Our solution integrates seamlessly with Jira, providing a cohesive project management experience. Key components of our solution include customizable gadgets displaying work logs, a project page for comprehensive insights, and a settings and management admin page for application-specific settings and customizations. With our solution, the company can streamline work logs tracking, enhance reporting capabilities, and eventually optimize resource utilization. By tailoring the application to meet specific needs, executives can make informed decisions, improve productivity, and achieve better projects outcomes.

The benefits of the proposed solution are:

• No resource waste: Using FaaS (Function-as-a-Service) ensures optimal use of computing resources through a serverless architecture, allowing for efficient, cost-effective, and scalable application development.

• Customizable data reports: With JFM, you can create custom reports based on your specific needs. This allows you to generate reports that are tailored to the organization's unique workflows, making it easier to track time effectively.

• Data Security: The app is built on a FaaS serverless architecture, which increase security; because the app is not tied to a specific server or infrastructure, there is no single point of failure that can be exploited by attackers. Additionally, the app can be automatically scaled up or down based on demand, which helps to prevent overloading and potential security breaches.

• Automated Data Entry: accurate data entry, with features like automatic time tracking and real-time synchronization. This helps to eliminate errors and ensure that the data is always up-to-date and accurate.

•Better user experience: Custom apps can improve the user experience of Jira by adding new features, simplifying workflows, or providing more intuitive interfaces. This can help teams get more value from Jira and reduce frustration and confusion.

• Data visualization: Offers interactive, visual representations of data to help users better understand and analyze their time data.

• Integration: Integrates seamlessly with Jira, allowing for a cohesive and efficient project management experience.

1.7. Development methodology: SCRUM

To effectively manage our project, we have chosen the Scrum methodology as it ensures good development performance in terms of quality and productivity. By adopting Scrum, we can organize our project process efficiently and adapt to changes as needed. This methodology allows us to break down the project into smaller, manageable chunks and prioritize features based on customer feedback. It enables us to identify and address risks and issues early on, ensuring testing and validation at each development stage. Collaboration and communication are emphasized, ensuring everyone works towards the same goals and resolving issues promptly. By adopting Scrum, we achieve high-quality and efficient development while remaining flexible and adaptable to changes. Scrum is a widely used agile approach known for its transparency, scalability, and evaluation, dividing projects into tasks to be completed within 1 to 4-week periods. Figure 1.4 represents the Scrum process.

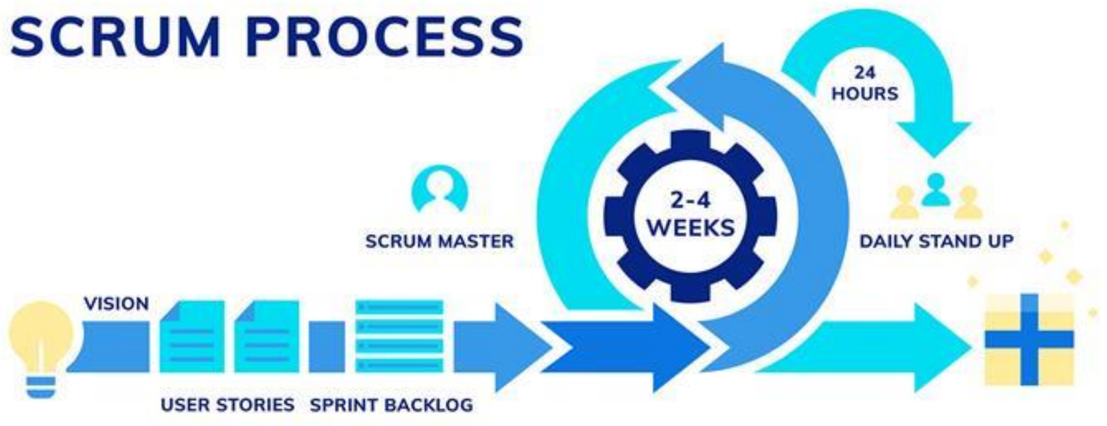


Figure 1.4 SCRUM PROCESS

A Scrum team is composed of a Scrum Master, a product owner, and up to 8 members in the development team to optimize communication and productivity. For each member of a Scrum team, there is a highly precise role to play. Let's take a look:

• The product owner: He is an expert who can set clear direction by defining functional specifications and prioritizing the work. He should not only understand the customer, but he should also have a vision and then ensure the value scrum team is delivering to the customer. His duties are: Managing the scrum backlog, managing stakeholders, and finally, releasing management which is the management of the process of preparing new software and ensuring a smooth and successful product launch.

• The Scrum master: The scrum master is a servant leader which represents a supportive manner of leadership. He is to help the product owner define value by serving him in sprint planning and sprint reviews. He is also essential to the development team as he helps them deliver the value by creating a Scrum supporting environment which is carried out through focusing on: Transparency, self-organization, courage, focus, commitment, respect, and openness.

• The development team: All members are expected to give their all to accomplish the assignments in a Scrum sprint. There are no defined positions in the team other than doing the handiwork. For that, it comes in no surprise that teammates often cooperate to determine goals and construct a plan to achieve them. Their duties include using data to forge the best development practices, assist in sprint planning and test products.

Now that we have seen the roles within a Scrum team, it is high time we moved on to inspecting the way a Scrum project is handled. In consequence, we will demonstrate the different stages of the project's progress.

• Sprint Planning: Before the next sprint, a team meeting identified as sprint planning takes place. During which, the development team evaluates its backlog and determines which things to prioritize for the following sprint. Once it's done, the team is left with two concepts: the first one is a target which is an overview of next sprint' plan, while the second is sprint's backlog which is the next sprint's list of tasks.

• Daily Scrum: The daily scrum, also known as the standup, is a 15-minute daily meeting in which the team plans out its work for the day and identifies any potential roadblocks.

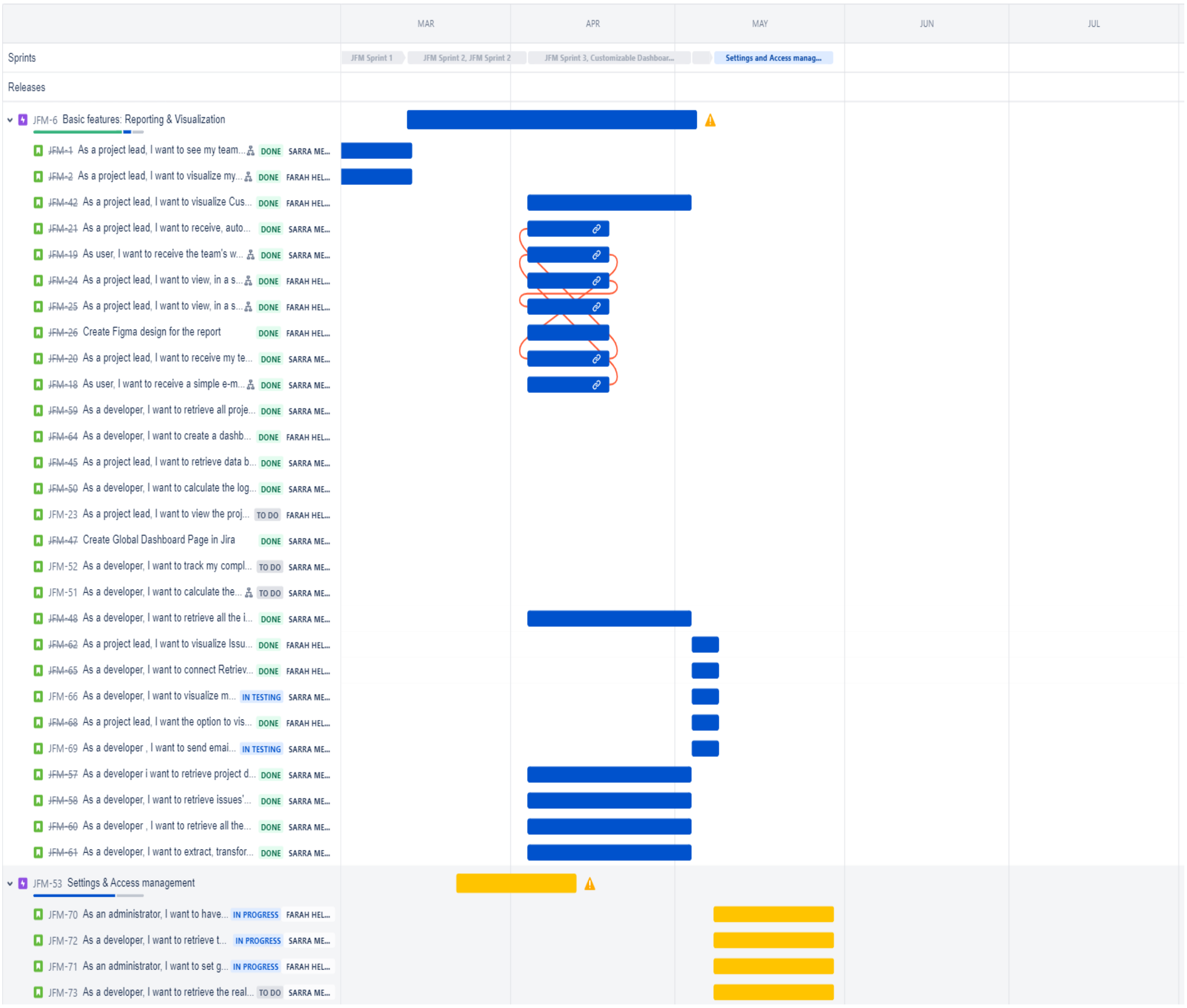
• Sprint Review: The sprint review is a critical perspective of the Scrum strategy. It is a normal assembly held at the end of a sprint to assess what went well during the previous sprint and how the following one can be improved. For my project, these meetings were held every fortnight at 4 pm during which, I got to update my product backlog and my project progress by sharing the recently finished tasks.

• Sprint Retrospective: The retrospective assembly takes place half an hour after the sprint review. A Sprint Retrospective seeks to improve the entire system so that the team's work flows swiftly and more harmoniously, whereas a Sprint Review focuses on improving, hence, delivering a better product.

• Product Backlog: A product backlog is an evolving, prioritized list of what needs to be done to improve a product. It is the only source of work embraced by the Scrum Group. [7]

Jira Roadmap

By referring to this roadmap, we were able to effectively track progress, align goals, and communicate project details with stakeholders. Figure 1.5 illustrates the Jira roadmap we used to plan our project's schedule.


MAR
APR
MAY
JUN
JUL
Sprints
JFM Sprint 1
JFM Sprint 2, JFM Sprint 2
JFM Sprint 3, Customizable Dashboar...
Settings and Access manag...
Releases

JFM-6	Basic features: Reporting & Visualization		
~~JFM-1~~	As a project lead, I want to see my team...	DONE	SARRA ME...
~~JFM-2~~	As a project lead, I want to visualize my...	DONE	FARAH HEL...
~~JFM-42~~	As a project lead, I want to visualize Cus...	DONE	FARAH HEL...
~~JFM-21~~	As a project lead, I want to receive, auto...	DONE	SARRA ME...
~~JFM-19~~	As user, I want to receive the team's w...	DONE	SARRA ME...
~~JFM-24~~	As a project lead, I want to view, in a s...	DONE	FARAH HEL...
~~JFM-25~~	As a project lead, I want to view, in a s...	DONE	FARAH HEL...
~~JFM-26~~	Create Figma design for the report	DONE	FARAH HEL...
~~JFM-20~~	As a project lead, I want to receive my te...	DONE	SARRA ME...
~~JFM-18~~	As user, I want to receive a simple e-m...	DONE	SARRA ME...
~~JFM-59~~	As a developer, I want to retrieve all proje...	DONE	SARRA ME...
~~JFM-64~~	As a developer, I want to create a dashb...	DONE	FARAH HEL...
~~JFM-45~~	As a project lead, I want to retrieve data b...	DONE	SARRA ME...
~~JFM-50~~	As a developer, I want to calculate the log...	DONE	SARRA ME...
JFM-23	As a project lead, I want to view the proj...	TO DO	FARAH HEL...
~~JFM-47~~	Create Global Dashboard Page in Jira	DONE	SARRA ME...
JFM-52	As a developer, I want to track my compl...	TO DO	SARRA ME...
JFM-51	As a developer, I want to calculate the...	TO DO	SARRA ME...
~~JFM-48~~	As a developer, I want to retrieve all the i...	DONE	SARRA ME...
~~JFM-62~~	As a project lead, I want to visualize Issu...	DONE	FARAH HEL...
~~JFM-65~~	As a developer, I want to connect Retriev...	DONE	FARAH HEL...
JFM-66	As a developer, I want to visualize m...	IN TESTING	SARRA ME...
~~JFM-68~~	As a project lead, I want the option to vis...	DONE	FARAH HEL...
JFM-69	As a developer , I want to send emai...	IN TESTING	SARRA ME...
~~JFM-57~~	As a developer i want to retrieve project d...	DONE	SARRA ME...
~~JFM-58~~	As a developer, I want to retrieve issues'...	DONE	SARRA ME...
~~JFM-60~~	As a developer , I want to retrieve all the...	DONE	SARRA ME...
~~JFM-61~~	As a developer, I want to extract, transfor...	DONE	SARRA ME...
JFM-53	Settings & Access management		
JFM-70	As an administrator, I want to have...	IN PROGRESS	FARAH HEL...
JFM-72	As a developer, I want to retrieve t...	IN PROGRESS	SARRA ME...
JFM-71	As an administrator, I want to set g...	IN PROGRESS	FARAH HEL...
JFM-73	As a developer, I want to retrieve the real...	TO DO	SARRA ME...

Figure 1.5 Jira Roadmap

1.8 Conclusion

To conclude what we saw in this part, we started by presenting the host organization, moving on to grasping the main idea of our project through an analysis of its general context and finishing up with specifying the methodology that would later be adapted in this project. For the upcoming chapter, we will be going through project management with Agile Scrum.

CHAPTER 2

PROJECT MANAGEMENT WITH « AGILE SCRUM »

2.1. Introduction

In the previous chapter, we decided to adopt the "Scrum" methodology for the design of our future system. The first phase of this methodology is the planning and architecture phase, which is the most important in the "Scrum" development cycle as it directly influences the success of the sprints. The work done during this period leads to building a good product vision, identifying user roles, and identifying the main functionalities in order to produce the initial backlog and a first sprint planning. We will also define the tools chosen for the development of the application. At this stage, we begin the state-of-the-art phase of our application. We identify the working environments, the technologies used, and the different components of the developed system.

2.2. Teams and Roles

In a SCRUM project, the team plays a fundamental role, it allows the optimization of productivity and flexibility. Indeed, it must be self-organized and multi-functional. There are three different SCRUM roles for members of a SCRUM team: the Product Owner, Scrum Master, and Development Team. Table 2.1 represents the actors of our project according to the "Scrum" methodology.

Table 2.1: Scrum Methodology Roles and Responsibilities

Actor	Role
Marouen BEN MOUSSA	Product Owner
Mamoun BOUSSIDA	Scrum Master
Sarra MEHREZ	Development Team
Farah HELALI	Development Team

2.3. Product backlog

The "Product backlog" is a prioritized list of tasks that define the characteristics of a product. It is one of the fundamental elements of the Scrum methodology. It is the main working tool of the Product Owner, who is responsible for gathering the needs of stakeholders and transforming them into a list of features ready to be developed by the development team. In the following table, we present the list of "user stories" (functional characteristics) indicating their priorities and the participating actors. Table 2.2 represents the product backlog.

Table 2.2: Product Backlog Table

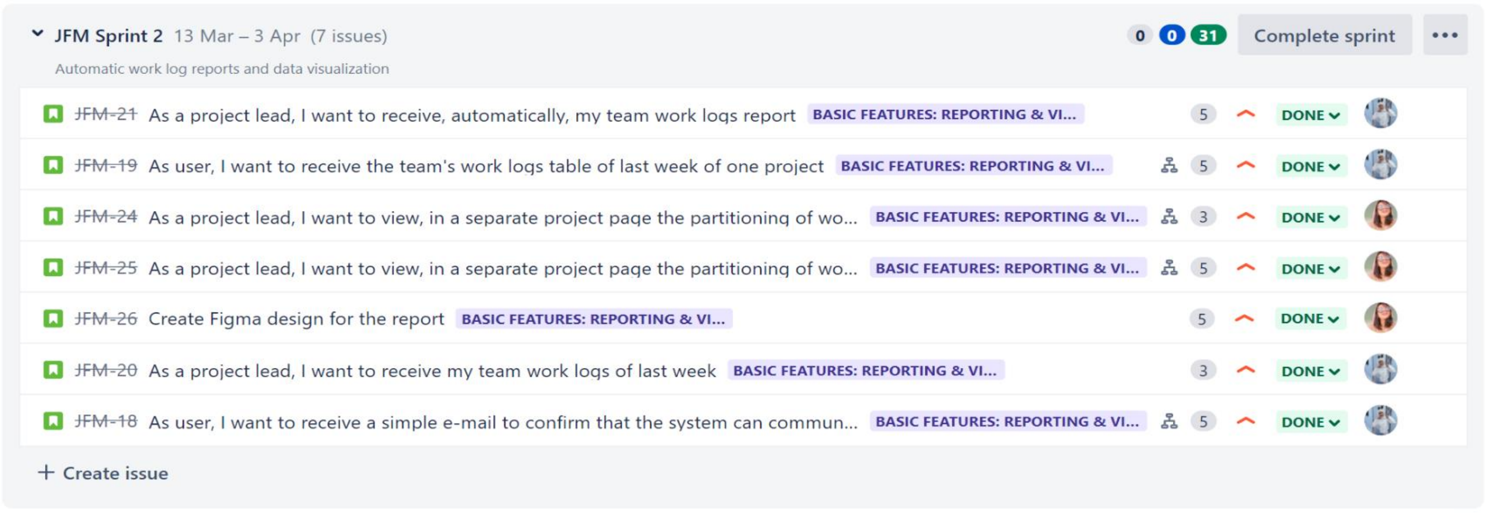
Key	User Story	Priority	Story Point estimate
JFM-18	As user, I want to receive a simple e-mail to confirm that the system can communicate with me	High	5
JFM-19	As user, I want to receive the team's work logs table of last week of one project	High	5
JFM-20	As a project lead, I want to receive my team work logs of last week	High	3
JFM-21	As a project lead, I want to receive, automatically, my team work logs report	High	5
JFM-24	As a project lead, I want to view, in a separate project page the partitioning of work logs by assignee in a pie chart with fake data	High	3
JFM-25	As a project lead, I want to view, in a separate project page the partitioning of work logs by assignee in a pie chart with real data	High	5
JFM-26	Create Figma design for the report	High	5
JFM-42	As a project lead, I want to visualize one gadget in the Jira Dashboard	Highest	8
JFM-45	As a project lead, I want to retrieve data based on specific group	High	5
JFM-50	As a developer, I want to calculate the log time for a Jira issue	High	3
JFM-57	As a developer, I want to retrieve project data and issue data to my dashboard gadget	High	3
JFM-58	As a developer, I want to retrieve issues' priorities	Medium	3
JFM-59	As a developer, I want to retrieve all project categories.	Medium	3
JFM-60	As a developer, I want to retrieve all the sprints by board id.	High	5
JFM-61	As a developer, I want to extract, transform, and regroup data from various APIs efficiently.	Highest	5
JFM-62	As a project lead, I want to visualize Issue Type gadget in the Jira Dashboard	Highest	5
JFM-66	As a developer, I want to visualize my gadgets with real data.	Highest	5
JFM-73	As a developer, I want to retrieve the real-time data of the global access restrictions.	Highest	8
JFM-70	As an administrator, I want to have a section dedicated to reporting configuration	Medium	5
JFM-73	As a developer, I want to retrieve the real-time data of the reporting configuration.	Highest	8
JFM-71	As an administrator, I want to set global access restrictions.	Medium	5
JFM-75	As a developer, I want to visualize data in the Project Page	High	8
JFM-76	As a developer, I want to visualize my data in a project Page	High	5
JFM-77	As a user, I want to visualize my project overview	High	5
JFM-78	As a user, I want to see a responsive project page	Medium	3

2.4. Identifying system sprints

To ensure effective application decomposition and SCRUM methodology implementation, we split the work into sprints. Each sprint has a defined backlog of product items chosen by the Product Owner to provide maximum value.

- **Sprint 1: Work Log reporting via email**

This sprint covers all aspects of comprehensive work log reporting and efficient data management. Figure 2.1 represents the first sprint backlog.



JFM Sprint 2 13 Mar – 3 Apr (7 issues) 0 0 31 Complete sprint •••

Automatic work log reports and data visualization

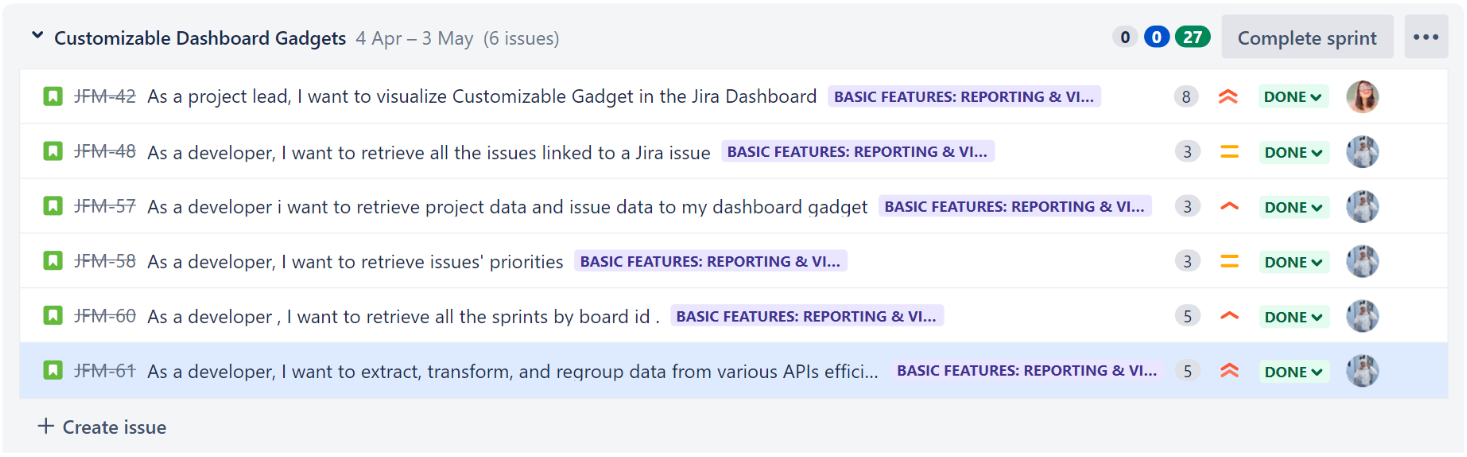
~~JFM-21~~	As a project lead, I want to receive, automatically, my team work logs report	BASIC FEATURES: REPORTING & VI...	5		DONE
~~JFM-19~~	As user, I want to receive the team's work logs table of last week of one project	BASIC FEATURES: REPORTING & VI...	5		DONE
~~JFM-24~~	As a project lead, I want to view, in a separate project page the partitioning of wo...	BASIC FEATURES: REPORTING & VI...	3		DONE
~~JFM-25~~	As a project lead, I want to view, in a separate project page the partitioning of wo...	BASIC FEATURES: REPORTING & VI...	5		DONE
~~JFM-26~~	Create Figma design for the report	BASIC FEATURES: REPORTING & VI...	5		DONE
~~JFM-20~~	As a project lead, I want to receive my team work logs of last week	BASIC FEATURES: REPORTING & VI...	3		DONE
~~JFM-18~~	As user, I want to receive a simple e-mail to confirm that the system can commun...	BASIC FEATURES: REPORTING & VI...	5		DONE

+ Create issue

Figure 2.1 Sprint 1 backlog

- **Sprint 2: Customizable Dashboard Gadgets**

Our second sprint is dedicated to customizing data in a dashboard gadget, specifically focusing on data visualization. Figure 2.2 represents the second sprint backlog.



Customizable Dashboard Gadgets 4 Apr – 3 May (6 issues) 0 0 27 Complete sprint •••

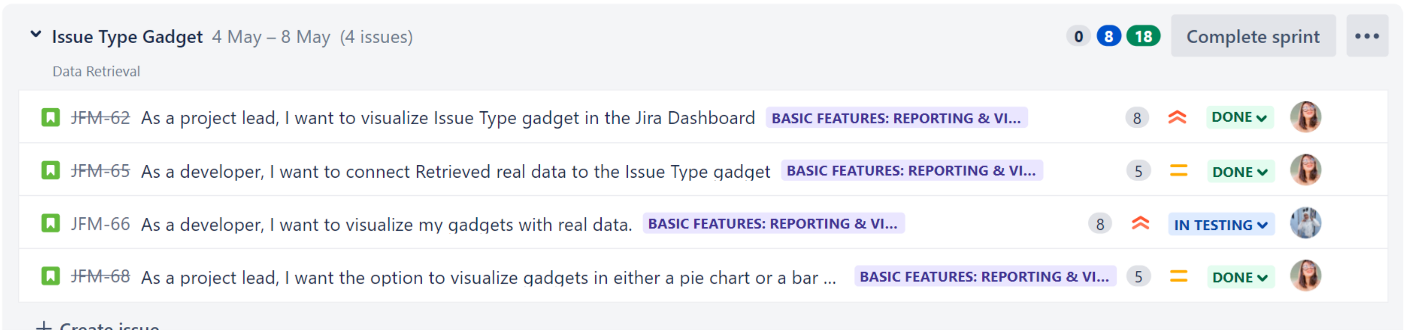
~~JFM-42~~	As a project lead, I want to visualize Customizable Gadget in the Jira Dashboard	BASIC FEATURES: REPORTING & VI...	8		DONE
~~JFM-48~~	As a developer, I want to retrieve all the issues linked to a Jira issue	BASIC FEATURES: REPORTING & VI...	3		DONE
~~JFM-57~~	As a developer i want to retrieve project data and issue data to my dashboard gadget	BASIC FEATURES: REPORTING & VI...	3		DONE
~~JFM-58~~	As a developer, I want to retrieve issues' priorities	BASIC FEATURES: REPORTING & VI...	3		DONE
~~JFM-60~~	As a developer , I want to retrieve all the sprints by board id .	BASIC FEATURES: REPORTING & VI...	5		DONE
~~JFM-61~~	As a developer, I want to extract, transform, and regroup data from various APIs effici...	BASIC FEATURES: REPORTING & VI...	5		DONE

+ Create issue

Figure 2.2 Sprint 2 backlog

- **Sprint 3: Issue Type Gadget**

Our sprint focuses on creating a dynamic dashboard for the client. We aim to integrate the Issue Type gadget into the Jira Dashboard, displaying actual data for enhanced project management. Figure 2.3 presents the third sprint backlog.



Issue Type Gadget 4 May – 8 May (4 issues) 0 8 18 Complete sprint •••

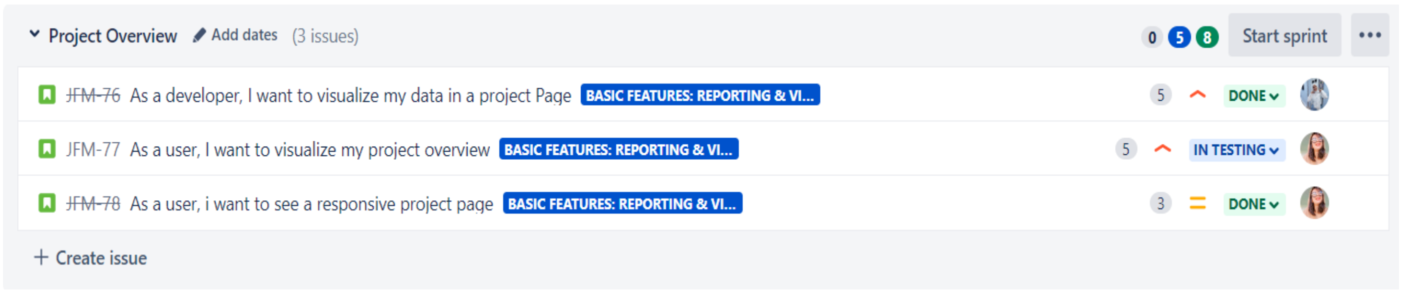
Data Retrieval

~~JFM-62~~	As a project lead, I want to visualize Issue Type gadget in the Jira Dashboard	BASIC FEATURES: REPORTING & VI...	8		DONE
~~JFM-65~~	As a developer, I want to connect Retrieved real data to the Issue Type gadget	BASIC FEATURES: REPORTING & VI...	5		DONE
JFM-66	As a developer, I want to visualize my gadgets with real data.	BASIC FEATURES: REPORTING & VI...	8		IN TESTING
~~JFM-68~~	As a project lead, I want the option to visualize gadgets in either a pie chart or a bar ...	BASIC FEATURES: REPORTING & VI...	5		DONE

Figure 2.3 Sprint 3 Backlog

- **Sprint 4: Project Page Overview**

In the fourth sprint, we implemented the Project Page, a feature that offers a comprehensive summary of projects and users present in the system. Figure 2.4 presents the fourth sprint backlog.


Project Overview Add dates (3 issues) 0 5 8 Start sprint

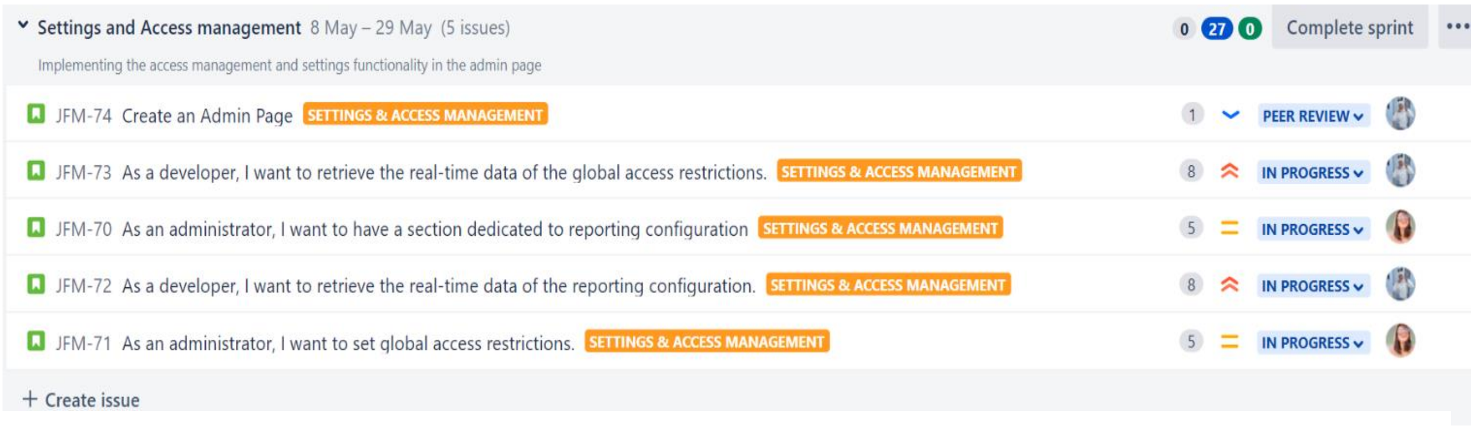
~~JFM-76~~ As a developer, I want to visualize my data in a project Page	BASIC FEATURES: REPORTING & VI...	5	DONE	
JFM-77 As a user, I want to visualize my project overview	BASIC FEATURES: REPORTING & VI...	5	IN TESTING	
~~JFM-78~~ As a user, i want to see a responsive project page	BASIC FEATURES: REPORTING & VI...	3	DONE	

+ Create issue

Figure 2.4 Sprint 4 backlog

- **Sprint 5: Settings and Access management**

This sprint focuses on implementing access management and settings functionality in the admin page. Figure 2.5 presents the last sprint backlog.


Settings and Access management 8 May – 29 May (5 issues) 0 27 0 Complete sprint

Implementing the access management and settings functionality in the admin page

JFM-74 Create an Admin Page	SETTINGS & ACCESS MANAGEMENT	1	PEER REVIEW	
JFM-73 As a developer, I want to retrieve the real-time data of the global access restrictions.	SETTINGS & ACCESS MANAGEMENT	8	IN PROGRESS	
JFM-70 As an administrator, I want to have a section dedicated to reporting configuration	SETTINGS & ACCESS MANAGEMENT	5	IN PROGRESS	
JFM-72 As a developer, I want to retrieve the real-time data of the reporting configuration.	SETTINGS & ACCESS MANAGEMENT	8	IN PROGRESS	
JFM-71 As an administrator, I want to set global access restrictions.	SETTINGS & ACCESS MANAGEMENT	5	IN PROGRESS	

+ Create issue

Figure 2.5 sprint 5 backlog

2.5. Identifying Releases

The release plan consists of a series of future sprints and outlines the anticipated contents of these sprints. Our approach to implementing the proposed solution involves dividing it into two primary releases, which are as follows:

1. First RELEASE
   - Sprint 1: **Work Log Reporting via Email**
   - Sprint 2: **Customizable Dashboard Gadget**
   - Sprint 3: **Custom Issue Dashboard**
2. Second RELEASE
   - Sprint 4: **Project Page Overview**
   - Sprint 5: **Settings and Access management**

2.6. Identifying actors

- **Collaborator**: Users who can customize reports, save settings, visualize projects, export data, switch between projects, fill forms, and view and analyze charts.
- **Administrator:** has the same permissions as a Collaborator, but with additional privileges, including managing permissions, configuring specific permissions, and controlling visibility of other users' work logs.

2.7. Non-Functional requirements

This project's primary goals were to answer new user requirements and complaints about their experiences. Therefore, it is imperative to identify all non-functional requirements that will play a key part in this project in addition to the functional requirements.

- **Security:** The security of our plugin is effectively managed through the implementation of Forge Sandboxing. In the following section 2.8, we will delve into the comprehensive details of forge platform security, discussing how it contributes to safeguarding user data and preventing unauthorized access.
- **Scalability:** Our plugin should be able to support a high number of users and projects while maintaining performance and stability. This scalability is achieved through the effective utilization of FaaS (Function as a Service) or Forge, enabling our plugin to dynamically scale resources to meet increasing demands and ensure a seamless experience for all users.
- **Maintainability and Extensibility**: Our plugin should have comprehensive documentation and well-structured code, making it simple to maintain and modify over time. It should also have a modular architecture and flexible design to support upcoming extensions or improvements.
- **Data Egress:** Our plugin has data egress capabilities that allow users to export their data from Jira without affecting the app's performance. This feature ensures a seamless user experience, enabling users to transfer and analyze their data outside of Jira easily.

2.8. Forge Platform Security (Sandboxing)

Atlassian's Forge Sandboxing is a comprehensive security feature that provides isolated and secure environments for developers to test and deploy their programs. By leveraging runtime isolation and restricting access to system resources, Forge Sandboxing prevents malicious code from executing, ensuring the security and reliability of apps on Atlassian's platform. Figure 2.4 represents the implementation of sandboxing; it illustrates the different layers involved. [4]

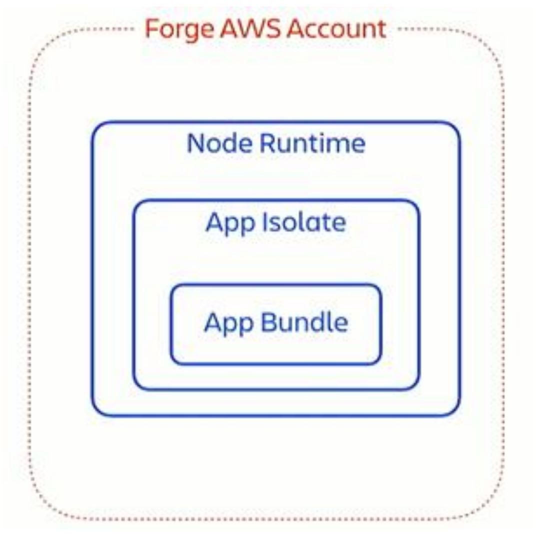


Figure 2.6 App Sandbox [4]

The sandbox consists of multiple layers, each contributing to the secure execution of apps:

- App bundle: The app bundle is the packaged app code.
- App isolate: App isolates are instantiated per request. Isolates provide per-request isolation within a single app. Isolates are customizable which enables Forge to control exactly what the app can and can't do.
- Node runtime: Forge apps run in AWS as lambdas. AWS Lambda provides per-app isolation.
- Forge AWS account: The Forge platform runs lambdas using multiple AWS accounts to distribute the load. All these accounts are separate from the other Atlassian services. Using dedicated accounts to run apps enables Forge to minimize privileges. This means that if an app escapes its sandbox due to a vulnerability, then running it in a less privileged account limits what it can do. [4]

With this layered approach, Forge Sandboxing ensures strong isolation, control, and reduced privileges, creating a robust security framework for app development on Atlassian's platform.

2.9. The architecture of the application

In this section, we will discuss the logical and physical architecture of our application, highlighting its well-organized structure and efficient design.

2.9.1 Logical architecture (Serverless)

Forge, Atlassian's serverless app development platform, forms the basis of our architecture. With Forge, we embrace a serverless approach, where the complexities of infrastructure management are abstracted away. This enables us to focus exclusively on developing the core logic of our application without having to deal with the intricacies of server provisioning, scaling, or maintenance. By leveraging the serverless capabilities of Forge, we benefit from a streamlined development experience and a highly scalable architecture. This aligns with Atlassian's vision for a serverless environment, where developers can fully concentrate on creating powerful applications within the Atlassian ecosystem, confident that the underlying infrastructure is automatically managed and maintained by the Forge platform. Figure 2.7 represents the Forge architecture. [6]

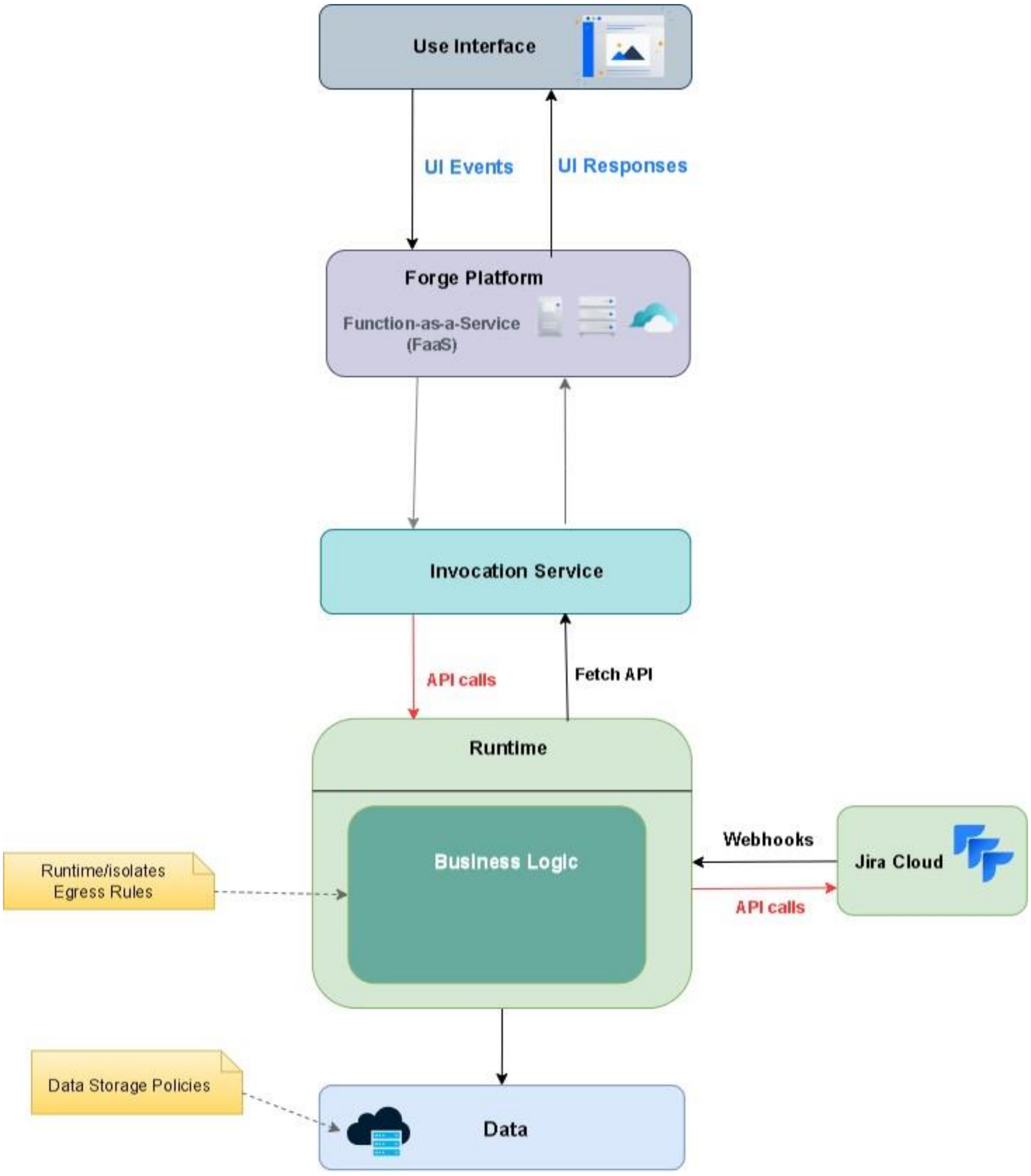


Figure 2 .7: Forge Architecture

- **UI Components:** The UI component refers to the user interface elements of the app, it is built using Forge UI, which provides a declarative way to define UI elements such as buttons, forms, and panels.

The UI component allows users to interact with the app and perform actions like editing content within the editor.

- **Event Handlers:** Forge apps can respond to various events within Jira, such as issue updates, comments, or workflow transitions. The serverless functions can be triggered by these events to perform specific actions or execute custom logic.
- **Forge Platform:** The Forge Platform serves as the underlying infrastructure that hosts and manages the serverless functions. It handles the scaling, execution, and lifecycle management of the functions, relieving developers from infrastructure management concerns. The Forge platform also provides services and APIs for integrating with Jira and other external systems.
- **Serverless Functions**: Forge uses a serverless FaaS model, where you write server-side code in the form of individual functions. These functions are event-driven and executed in response to specific events or triggers, such as HTTP requests or Jira-specific events like issue updates or comments.
- **Invocation Service:** The invocation service is part of the Forge platform, it handles authentication and routing tasks, and is triggered when the app is invoked, for example, when a user interacts with the UI component. It invokes the necessary function associated with the app to perform the required actions.
- **Business Logic:** The business logic defines the specific functionality, data processing, integration, error handling, security measures, and customization capabilities of the application. It drives how the application behaves, processes data, interacts with external systems, handles errors, and ensures security. Business logic is essential for implementing the unique business rules and requirements of the application, ultimately delivering its intended functionality and value.
- **API Gateway:** Forge includes an API Gateway component that serves as the entry point for incoming requests to the serverless functions. This component takes care of routing the requests to the relevant function and manages authentication and authorization processes. By making API calls, the app can interact with Jira's APIs, allowing it to retrieve data or update issue information. These API calls facilitate seamless integration between the app and Jira, granting the application access to relevant data and the ability to modify it as needed.
- **Webhooks:** This component represents the use of webhooks in the app. Webhooks are HTTP callbacks that Jira Cloud sends to the app when specific events occur, such as issue creation or updates. In the context of Forge, the app can use the Forge Trigger API to receive webhook events, enabling real-time updates and event-driven workflows.
- **Integration with Jira:** Forge apps can interact with Atlassian's APIs (like Jira) to perform operations such as creating or updating issues, retrieving issue data, and performing other Jira-related tasks. The serverless functions can make API calls to Jira to integrate the app's functionality with the Jira platform.
- **Data Storage:** Forge apps can leverage data storage services provided by the Forge platform. The data component provides storage, organization, retrieval, and manipulation capabilities for the application's data. It interacts with business logic, allowing it to retrieve, manipulate, validate, and store data based

on specific requirements. The data component ensures data integrity, security, and adherence to defined data storage policies, enabling the business logic to effectively operate on the data within the application.

2.9.2 Physical architecture: Cloud Computing

The physical architecture of Forge, provided by Atlassian, is built on cloud computing principles. Atlassian's cloud infrastructure serves as the foundation for executing and hosting Forge apps, relieving developers from the burden of managing their own infrastructure. The architecture encompasses various components, including cloud computing resources, a user interface component, app logic, the Forge platform by Atlassian, and data storage capabilities. This configuration enables developers to concentrate on app development while leveraging Atlassian's cloud infrastructure for enhanced scalability and reliability. Figure 2.8 illustrates the physical architecture of our plugin.

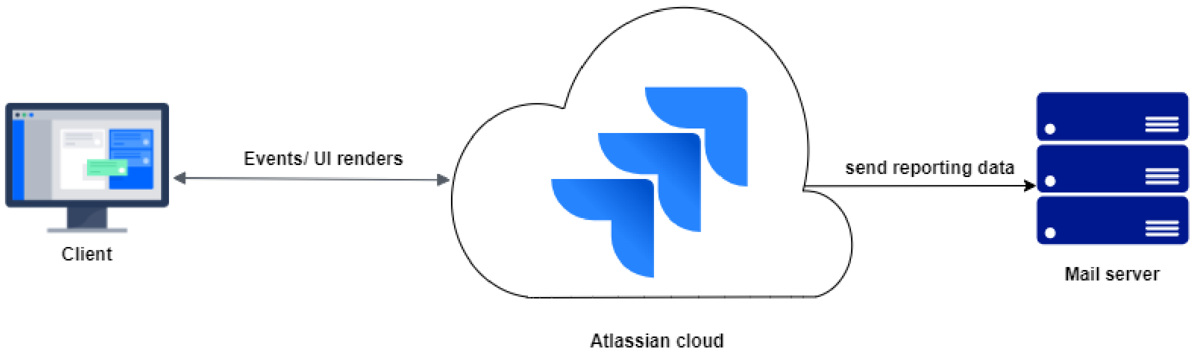


Figure 2.8 Physical Architecture

The architecture consists of these important components:

- **Client:** This component represents the device used by the user to interact with the system. The client interacts with Atlassian Cloud (Jira) through events triggers to perform various actions or retrieve information.
- **Atlassian Cloud (Jira):** This component represents the cloud-based instance of Jira, It provides a centralized platform for managing projects, tasks, and workflows. Users interact with Jira through their web browsers or Jira client applications installed on their devices.
- **Mail Server:** This component represents the mail server responsible for handling email communication. Jira in Atlassian Cloud can send reporting data to the mail server. The mail server processes and delivers these emails to the intended recipients.

2.10. Working environment

In this chapter, we will discuss the technologies utilized in our application's development, taking into consideration the design specifications outlined in the previous sections. Additionally, we will conduct a thorough comparison between two prominent development options: Forge and Atlassian Connect. Our primary objective is to delve into the reasons that led us to select Forge as the preferred development platform.

2.10.1 Used Technologies

The following section will present each programming language that was implemented to create our application:

- React.js

React is a JS library for creating UIs with free access and open-source support from Meta and a community of developers and companies.

Figure 2.9 React.js logo

- Node.js

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser.

Figure 2.10 Node.js logo

- Jira

Jira is a project management tool developed by Atlassian, which is used to manage and track project issues, bugs, and tasks. The Forge app is built on top of Jira and integrates with Jira's APIs to access and manipulate data.

Figure 2.11 Jira logo

- Forge

A cloud development platform for building and deploying apps in Jira and Confluence. It provides a set of APIs and tools that make it easier for developers to build and deploy apps.

Figure 2.12 Forge logo

- Forge CLI

A command-line tool that enables developers to create, build, and deploy Jira and Confluence apps using Forge. It provides a set of commands for creating new apps, testing them locally, and deploying them to the cloud.

Figure 2.13 Forge CLI

- GitHub

A web-based hosting service for Git repositories. It provides features for version control, issue tracking, and pull requests, among others.

Figure 2.14 Git Hub logo

- Docker

It is a Linux container-based virtualization technology that allows you to create containers containing applications with their dependencies.

Figure 2.15 Docker logo

- D3Js

D3 is a JavaScript library used for data visualization on the web. It provides tools for creating dynamic, interactive, and customizable visualizations using SVG, HTML, and CSS. D3 is known for its powerful data manipulation and transformation capabilities.

Figure 2.16 D3Js logo

- Atlaskit

Atlaskit is the Atlassian's official UI library, built according to the Atlassian Design Guidelines.

Figure 2.17 Atlaskit logo

2.10.2. Forge vs. Connect

In this section, we will delve into a comparison between Forge and Connect. We will provide an overview of Connect. Next, we will shift our focus to Forge, highlighting its unique characteristics and advantages. Finally, we will present a comparison table that highlights the key differences between Forge and Connect [2].

- **Connect**

Connect is a development framework provided by Atlassian that enables developers to extend Atlassian products and integrate them with external systems. It offers a wide range of capabilities for building add-ons and customizing Atlassian applications.

- **Forge**

Forge is a modern development platform introduced by Atlassian announced in 2020 and released for general availability in 2021 that aims to simplify the process of building and deploying apps for the Atlassian Cloud. It provides a streamlined and efficient environment for developers, offering numerous advantages for building applications within the Atlassian ecosystem. Key features and benefits of Forge include:

- **Comparison between Connect and Forge**

Table 2.3 represents a comparison between Forge and Connect.

Table 2.3: Comparison between Connect Vs Forge [2]

Feature	Forge	Connect
Development Environment	Cloud-only	Cloud and Server
Supported Products	Jira, Confluence, and Bitbucket	Jira, Confluence, Bitbucket, and other Atlassian products
Frontend Development	Custom UI Framework	Custom UI or Iframe
Backend Development	JavaScript (Node.js)	Java, .NET, PHP, Python, and other languages
Infrastructure Management	Managed by Atlassian	Self-managed
UI Extensibility	Limited options	More flexibility with custom UI
Data Storage	Persistent storage in the Forge ecosystem	Use of REST APIs for data storage
Hosting and Deployment	Deployed and managed by Atlassian	Self-hosted or hosted by Atlassian
Marketplace and Distribution	Atlassian Marketplace integration	Marketplace integration and custom distribution channels
Security	Secured by Atlassian	Dependent on self-hosted or Atlassian-managed environment
API Compatibility	Version-independent, auto-upgrading	Dependent on specific API versions

We have chosen Forge because it offers several advantages that make it a favorable choice. Forge provides a cloud-only development environment, a custom UI framework for frontend development, and Atlassian manages the infrastructure, eliminating the need for self-management. Forge also offers greater UI extensibility, persistent data storage, and simplifies hosting, deployment, marketplace integration, security, and API compatibility. By leveraging these features, Forge simplifies application development in the Atlassian Cloud, providing a streamlined, secure, and efficient platform for building powerful applications.

2.11 Conclusion

Now that we've laid a solid basis for our project, including its architecture, deployment strategy, and technological infrastructure, we can focus on designing the platform's end-user features. This allows us to design a truly compelling and user-friendly solution that meets the needs of our target audience. Our goal is to provide a comprehensive and effective platform that improves the user experience and meets their needs. The next chapter will be dedicated to the detailed analysis of the first release.

CHAPTER 3

DESIGN AND IMPLEMENTATION OF RELEASE 1

3.1 Introduction

In our first release, we focused on improving the user experience and functionality of our plugin through three sprints. Sprint 1 introduced the Work Logs Report, which allows project leads to track their team's work activities and progress. Sprint 2 added the Customizable Dashboard Gadget, which allows users to tailor their dashboard to their needs and preferences. Finally, Sprint 3 introduced the Custom Issue Dashboard, providing a dynamic and personalized view of project issues for better management.

3.2. Realization of Sprint 1: Work Log reporting via e-mail

3.2.1 Specification of needs

A good specification is important for a successful design. It helps developers understand what the client needs and improves communication. This section will cover the requirements for the sprint 1, the specification, and technical constraints.

- **Extract Data:** In this use case, our plugin system extracts data related to work logs from the Jira platform for the previous week. This data includes information such as assignees, time spent, and dates. The extracted data is then passed to the Node.js Server for further processing.
- **Process Data:** Once the Node.js Server receives the extracted data, it performs several operations to process and treat the data. This includes calculating total work hours for each assignee, organizing the data into a tabular format, and applying any necessary formatting or transformations. The result is a table that represents the work logs of the previous week.
- **Send Email:** After the data has been processed and converted into a table format, the Node.js Server uses the Email Sender component to send an email containing the generated table. The email is addressed to the intended recipient, which could be the Scrum Master or another designated user. The email is typically scheduled to be sent every Monday at 9 am, ensuring that the recipient receives the weekly work log report in a timely manner.

3.2.2 Conception of sprint

This section is intended to present the design of the "Work Log reporting via e-mail ".

3.2.2.1 Detailed sequence diagram

This sequence diagram showcases the interactions within the system for extracting work log data, processing it, and sending the generated report via email. The plugin system extracts data from Jira for the previous week, which is then passed to the Node.js Server. The server processes the data by calculating total work hours, organizing it into a table format, and applying necessary formatting. Finally, the Email Sender component handles sending the email with the generated table to the designated recipient, scheduled for weekly delivery on Mondays at 9 am. The sequence diagram "Send Email Weekly" presented in Figure 3.1.

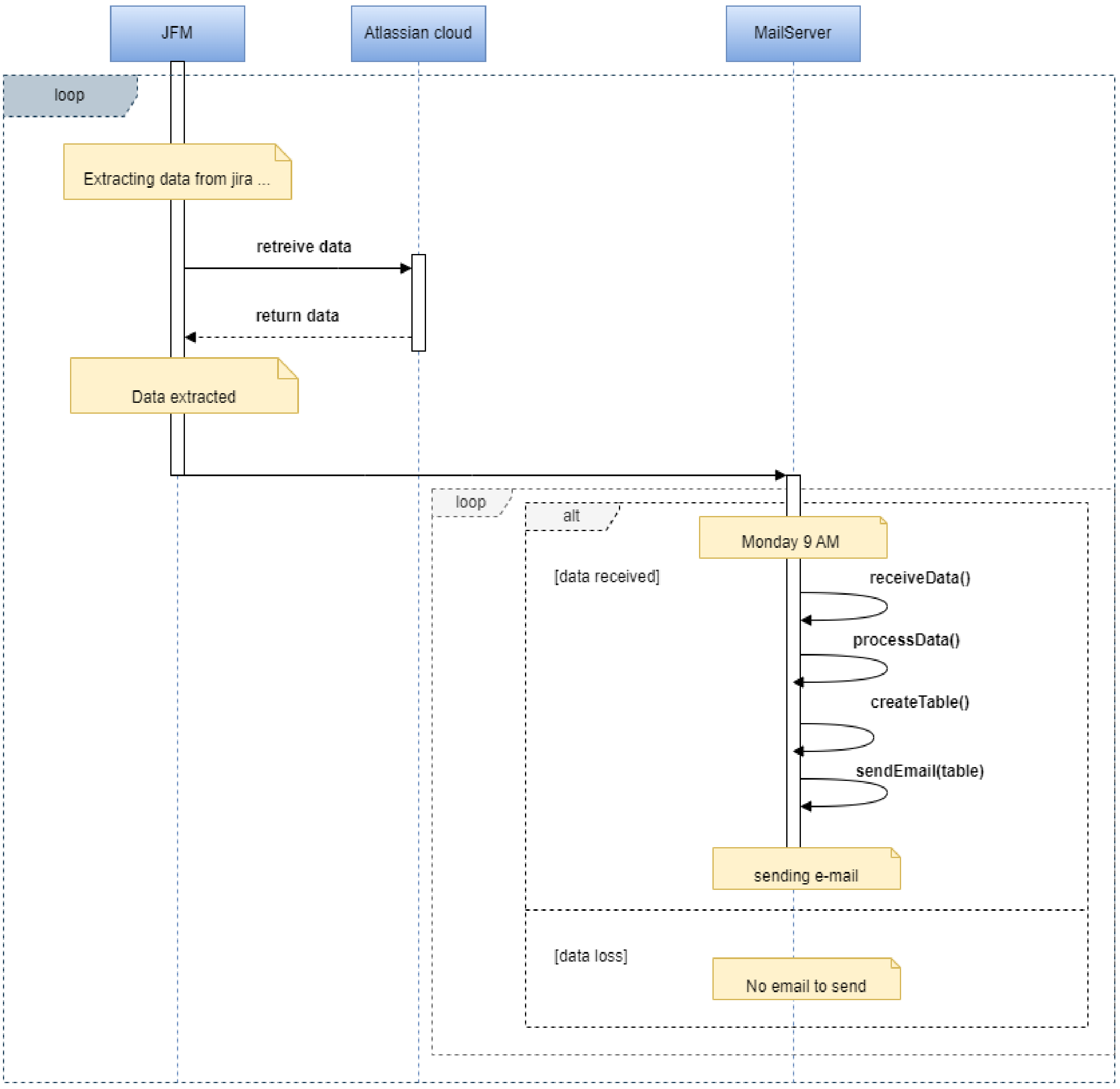


Figure 3.1 Sequence diagram "Send Email Weekly"

3.2.2.1 Component diagram

The component diagram provides an overview of the key components within the Global Sprint system and their interconnections. It depicts the composition relationships among the Jira Atlassian system, the Node.js Server, and the Email Sender, which collaborate to generate and deliver the weekly work log report. This diagram offers insights into the organization and interactions of these components, highlighting their roles in extracting, processing, and sending work log data.

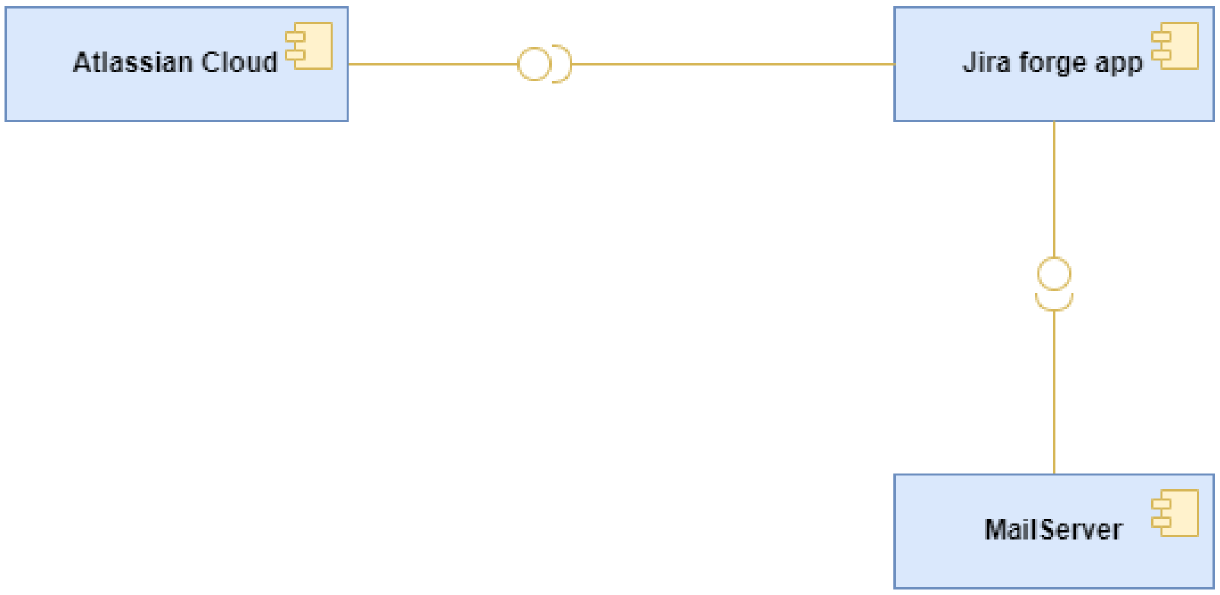
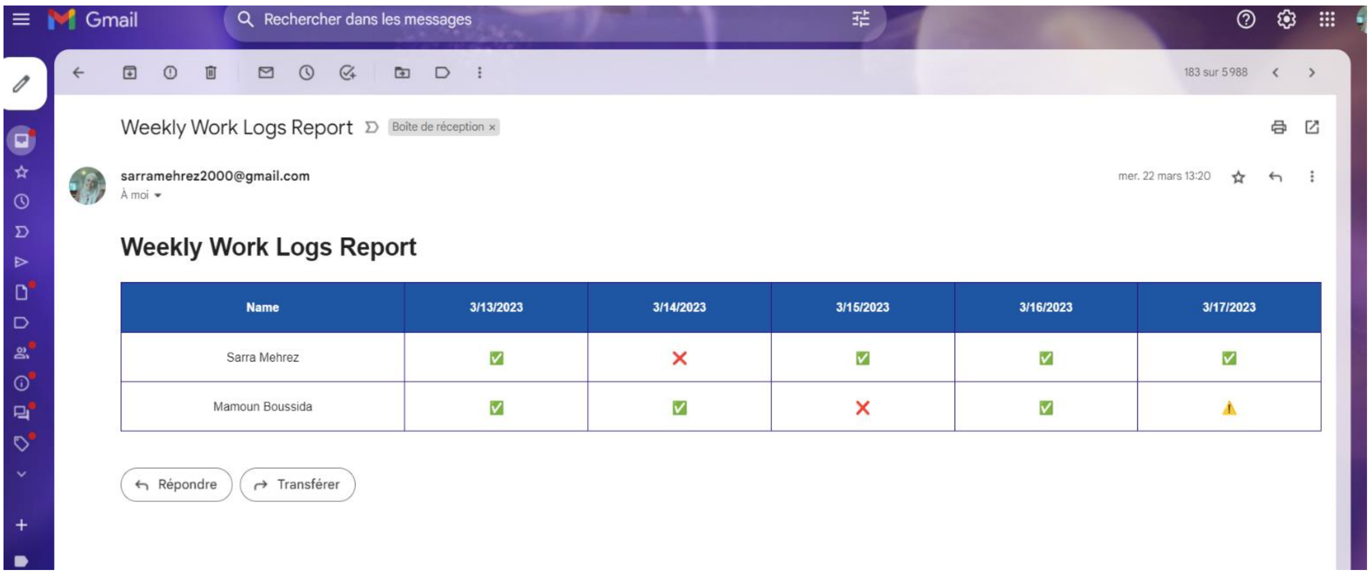


Figure 3.2 Component diagram of sprint 1

3.2.3. Implementation of Sprint 1:

The implementation consists of an automated system that sends weekly work log reports to users. These reports offer a comprehensive summary of the work logs for each assignee, categorized by project, during the previous week. The system diligently tracks the work hours of each assignee for five days, with the target set at 7.30 hours per day. The reports provide clear indications of whether assignees have successfully met, surpassed, or fallen short of the expected work hours. This functionality facilitates an effortless assessment of assignee engagement and productivity for users. Figure 3.3 represents an example of a weekly work logs report sent via e-mail.



Weekly Work Logs Report

sarramehrez2000@gmail.com
À moi
mer. 22 mars 13:20

Weekly Work Logs Report

Name	3/13/2023	3/14/2023	3/15/2023	3/16/2023	3/17/2023
Sarra Mehrez	✅	❌	✅	✅	✅
Mamoun Boussida	✅	✅	❌	✅	⚠️

Figure 3.3 Implementation of Sprint 1

3.3. Realization of Sprint 2: Customizable Dashboard Gadget

This section covers the requirements, specifications, and technical constraints for Sprint 2, which focuses on the development of a customizable dashboard gadget.

3.3.1 Specification of needs

- **Visualize gadget:** The actor selects the relevant filters using the form provided in the gadget. The system generates a chart based on the selected filters and displays it on the gadget's interface. The actor can view the chart and analyze the data.
- **Customizable Appearance:** The system allows the actor to customize the appearance of the chart, such as changing the colors, font sizes, and labels. This allows the actor to create a chart that is visually appealing and easy to understand.
- **Fill Form:** The actor selects the relevant filters such as project type, time period, and other criteria using the form, and the system generates a chart based on the selected filters. The purpose of the Select Filters use case is to ensure that the actor is able to select the appropriate filters to generate the desired chart.

3.3.1.1 Detailed use case diagram

The Use case diagram of Sprint 2 presented in Figure 3.4.

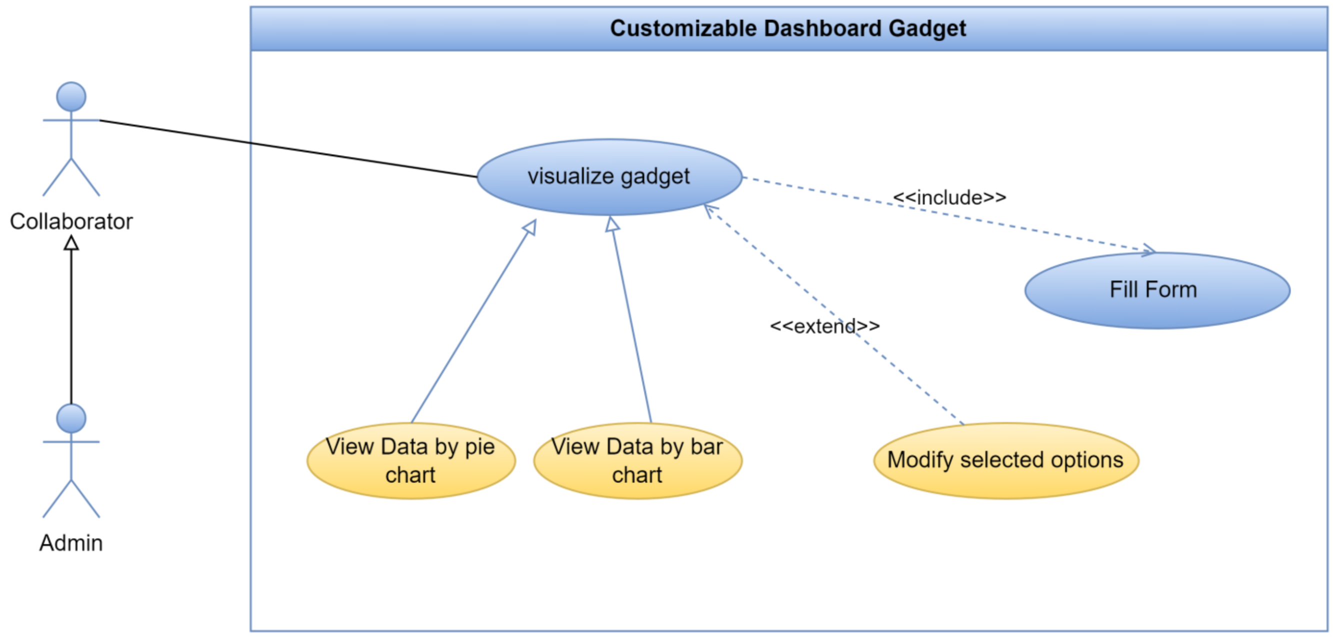


Figure 3.4 Use case diagram "Customizable Dashboard Gadget"

3.3.2 Conception of sprint 2

During the conceptual phase of Sprint 2, we have created several detailed diagrams to provide a visual representation of the sprint management system.

3.3.2.1 Detailed sequence diagram

This sequence diagram illustrates the flow of interactions between the Admin, Forge application (FaaS) and Atlassian System (external system), while retrieving real data and filling the form with the data in the Forge application. The sequence diagram of "Customizable Dashboard Gadget" presented in Figure 3.5.

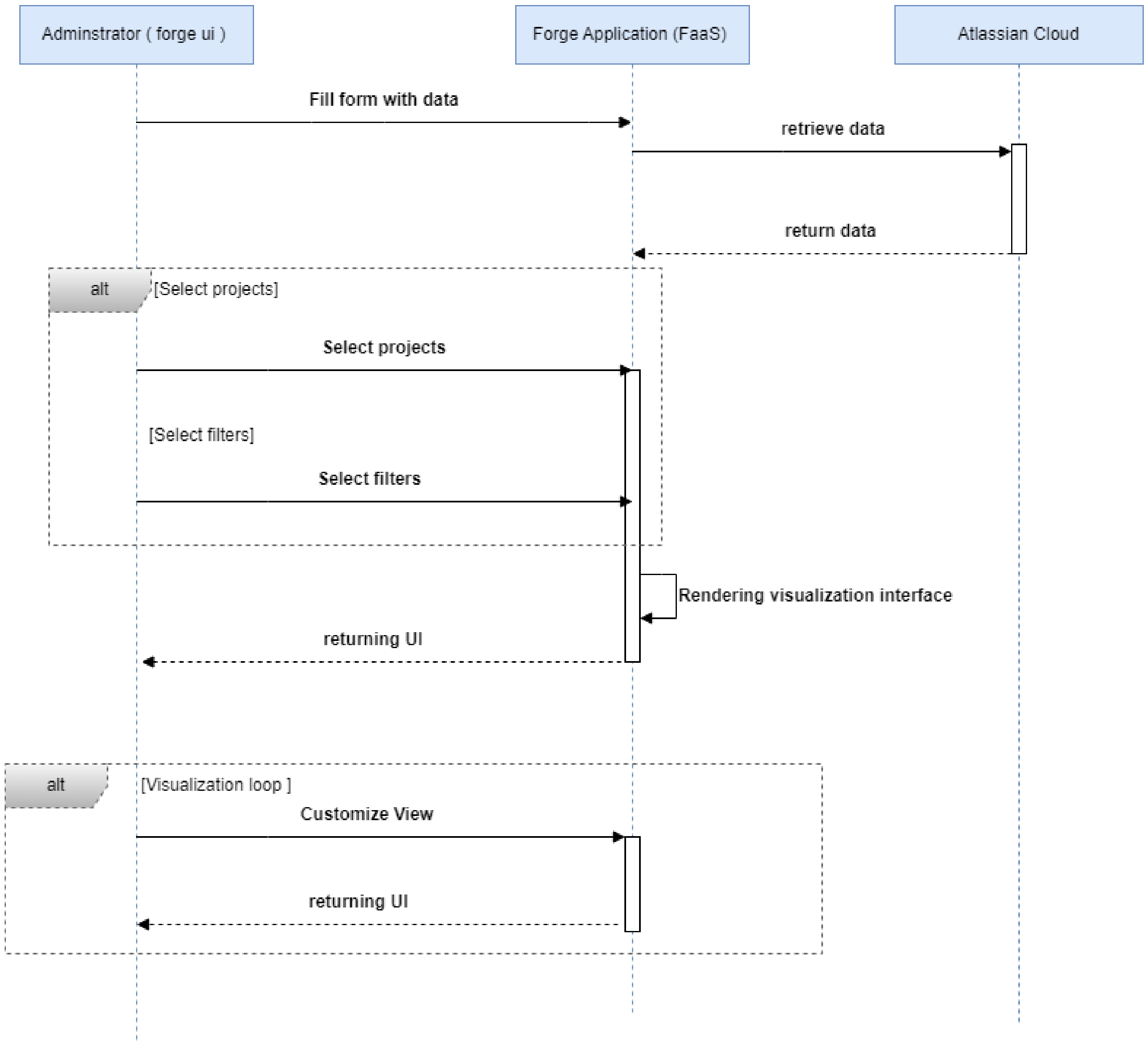


Figure 3.5 Sequence diagram "Customizable Dashboard Gadget"

3.3.2.2 Class diagram

The "Gadget" class represents the visual representation of the application, which includes charts that can be customized with changing filters. The "Form" class represents the form where the collaborator can select filters. The "Chart" class represents the generated chart based on the selected filters. The "Filters" class represents the selected filters for customizing the data displayed in the chart. The "Collaborator" class represents the individual users interacting with the application. This section introduces the class diagram for the customizable dashboard gadget system.

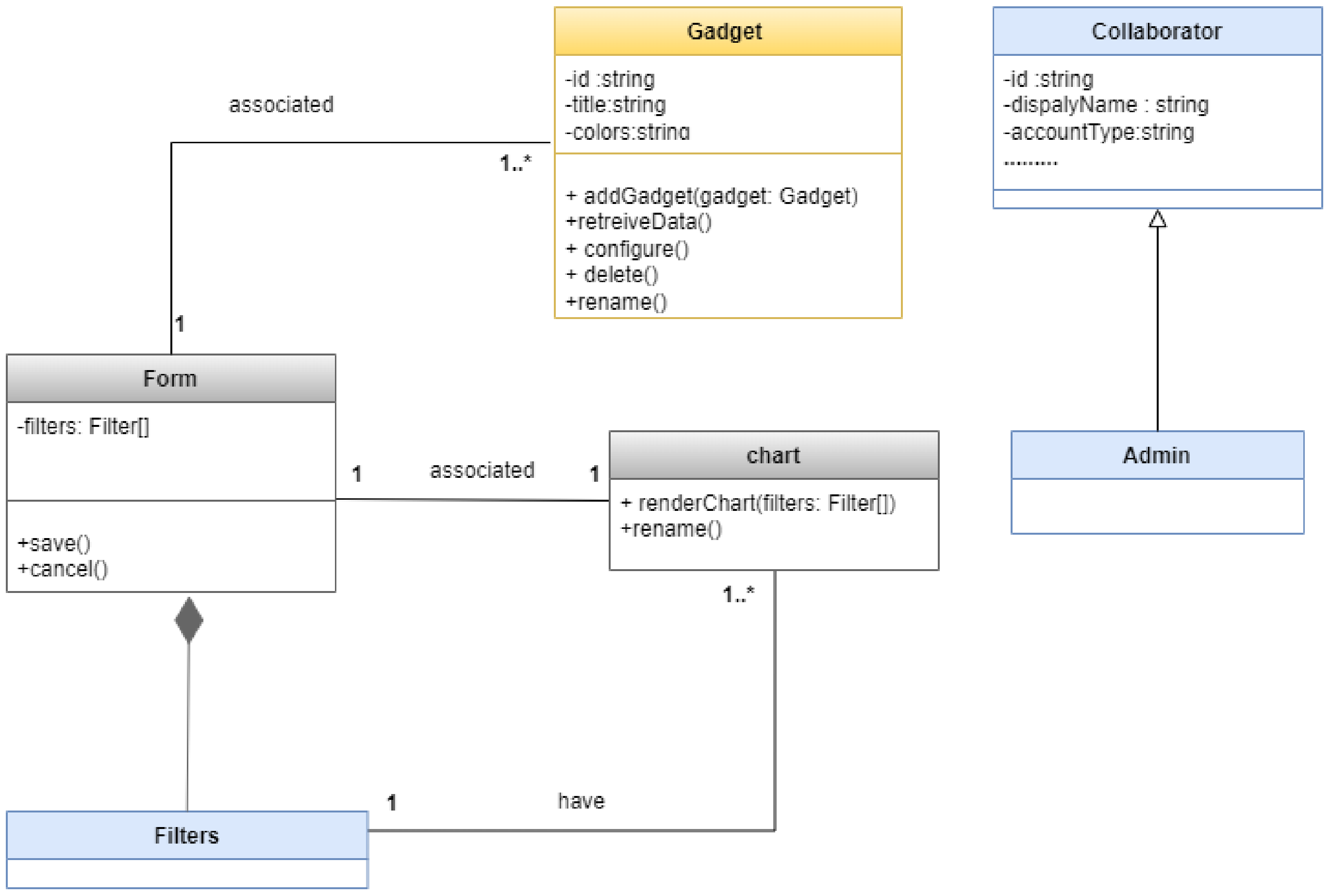


Figure 3.6 Class diagram "Customizable Dashboard Gadget"

3.3.2.3 Activity diagram

This diagram focuses on a specific aspect of the system's functionality, which involves visualizing and customizing charts based on selected filters.

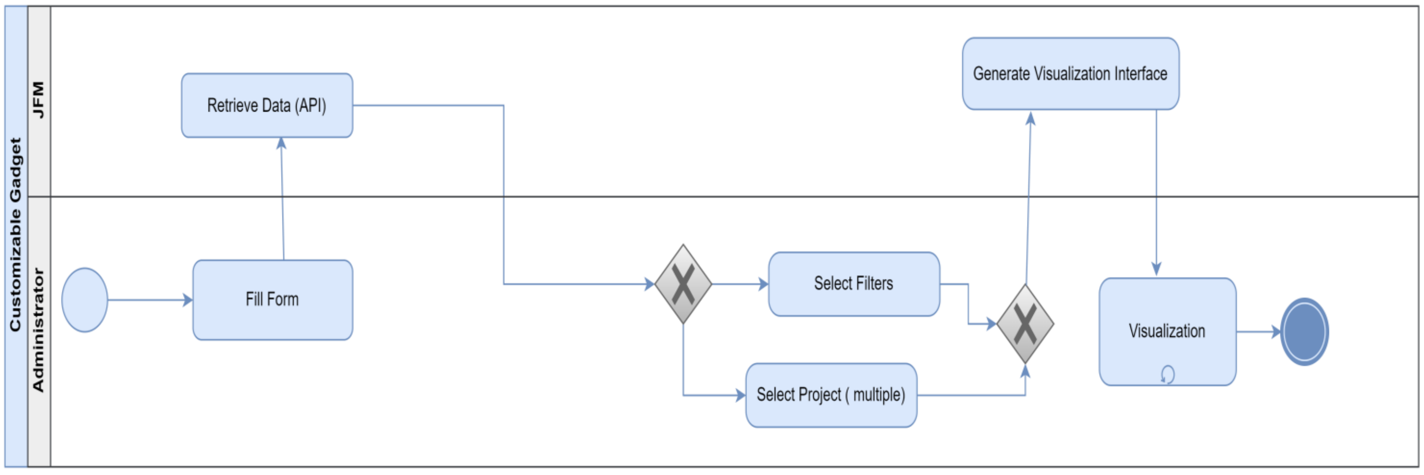
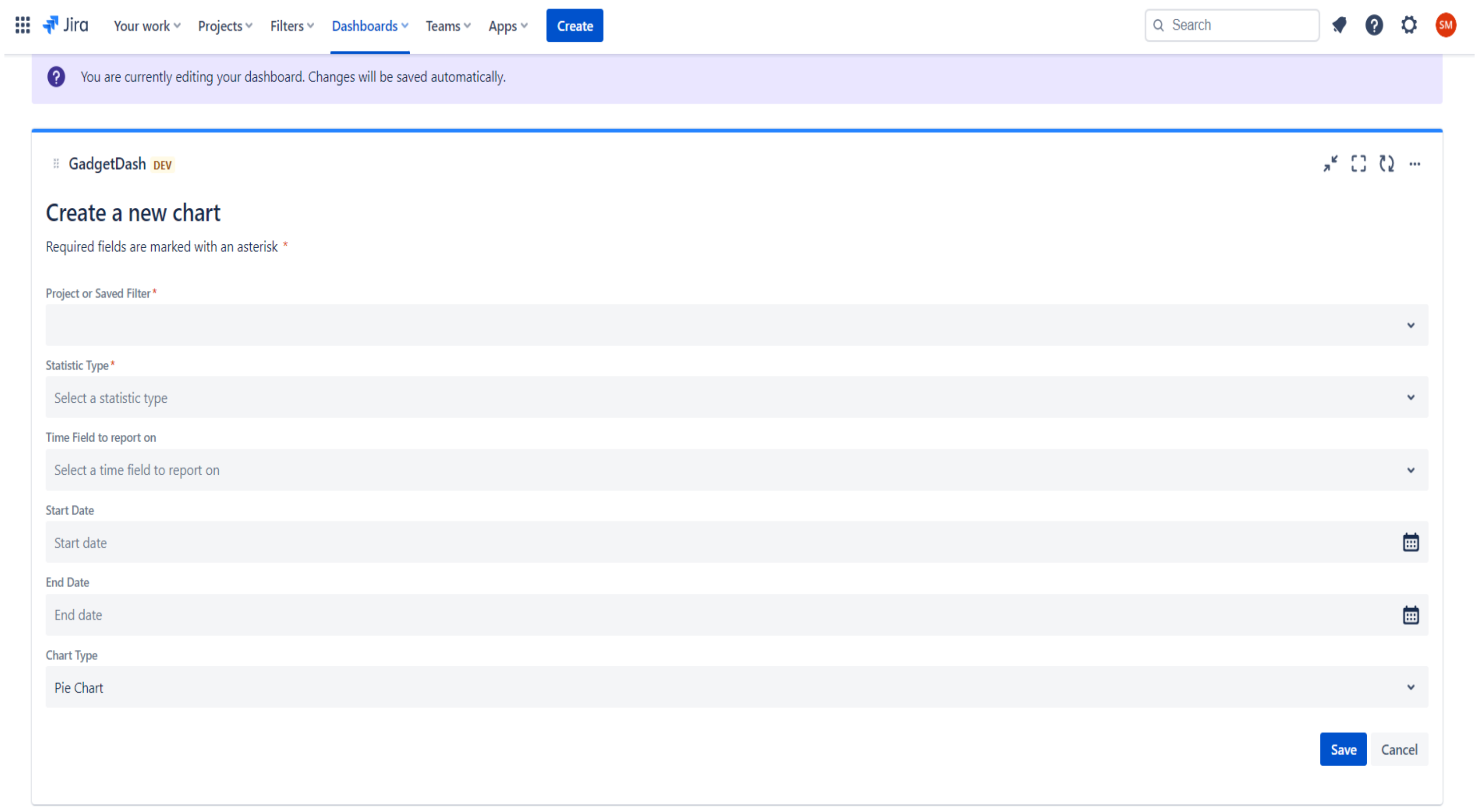


Figure 3.7 Activity diagram "Customizable Dashboard Gadget"

3.3.3 Implementation of Sprint 2

In this section, we will delve into the implementation of customizable gadgets. Figure 3.8 showcases the implemented form that provides users with a comprehensive set of filters to customize their view.



You are currently editing your dashboard. Changes will be saved automatically.

GadgetDash DEV

Create a new chart

Required fields are marked with an asterisk *

Project or Saved Filter*	
Statistic Type*	Select a statistic type
Time Field to report on	Select a time field to report on
Start Date	Start date
End Date	End date
Chart Type	Pie Chart

Save Cancel

Figure 3.8 "Fill Filters"

Figure 3.9 visually summarizes the distribution of work logs among different sprints, giving an overview of the project's progress.

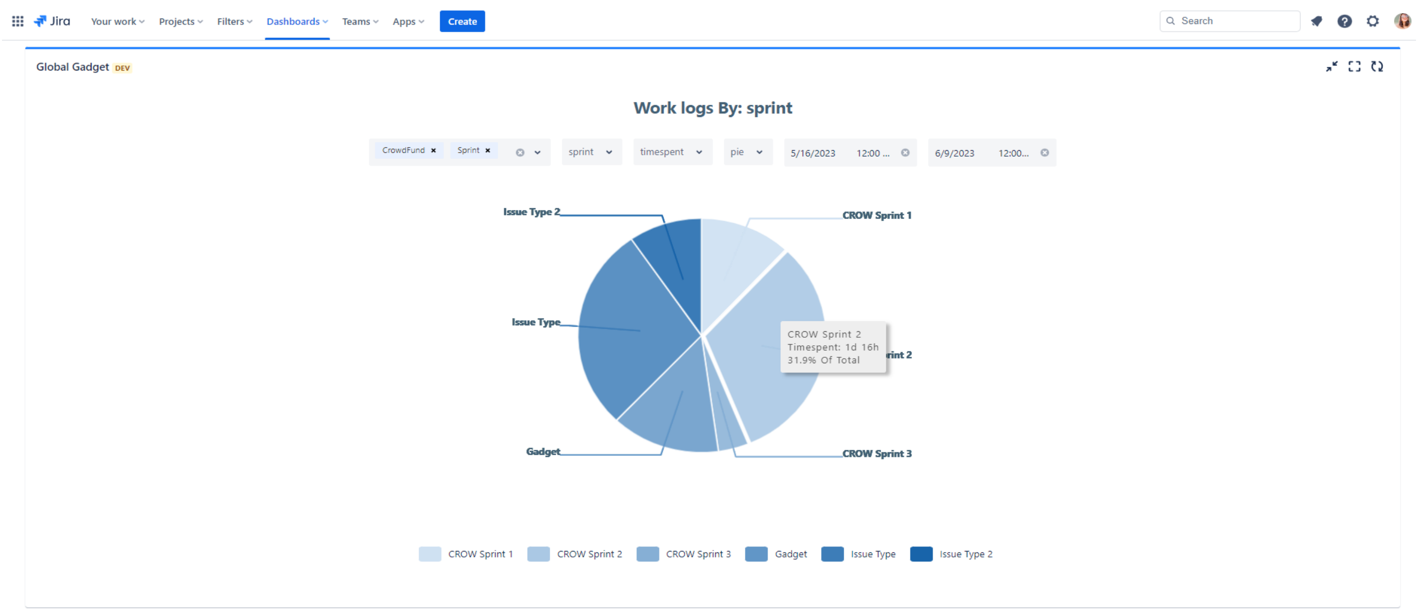


Figure 3.9 “Pie Chart Gadget: Work Logs Distribution by Sprints”

Figure 3.10 represents two gadgets side by side, each presenting a different chart type.

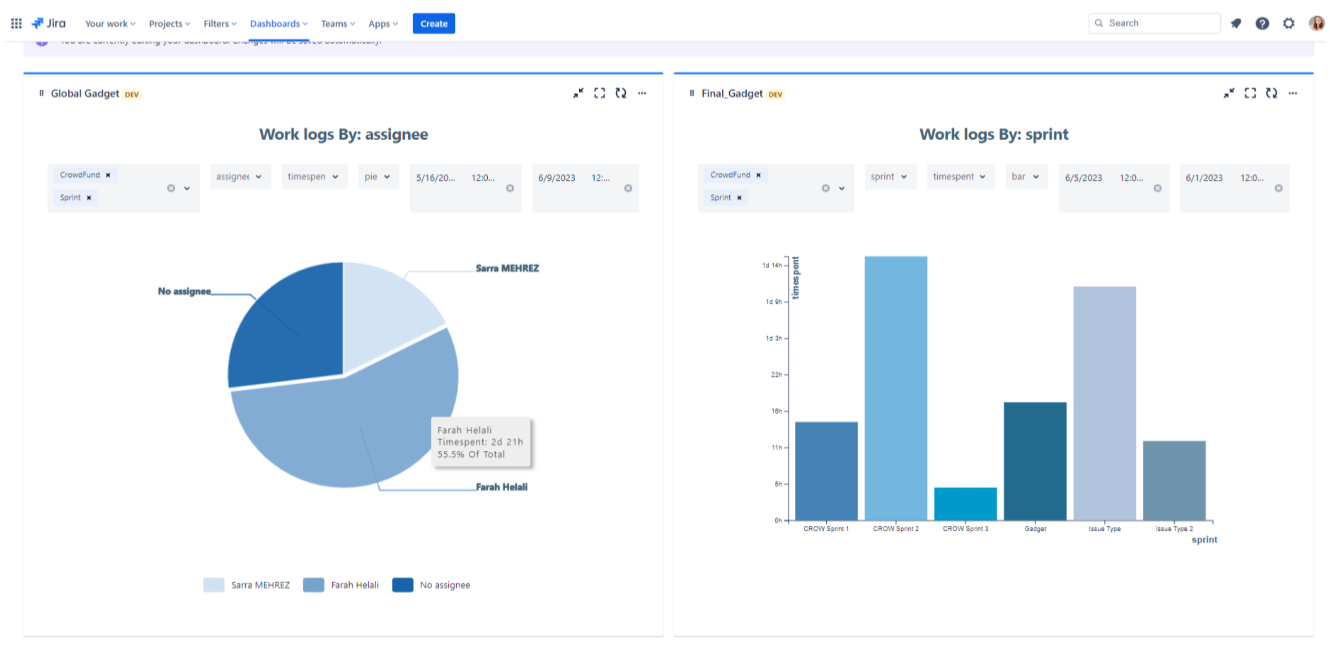


Figure 3.10 "Gadget Showcase: Pie Chart and Bar Chart for Data Visualization"

3.4. Realization of Sprint 3: Custom Issue Gadget

This section covers the requirements, specifications, and technical constraints for Sprint 2, which focuses on the development of a customizable issue gadget.

3.4.1 Specification of needs

In this section, we will discuss the analysis and specification of the requirements for Sprint 3 of Release 1. The functional requirements are outlined in the following list:

- **Fill Form:** The actor selects the relevant filters such as projects, issue type, time period, and other criteria using the form, and the system generates a chart based on the selected filters. The purpose of the Select Filters use case is to ensure that the actor is able to select the appropriate filters to generate the desired chart.
- **Visualize gadget:** The system generates a chart based on the selected filters and displays it on the gadget's interface. The actor can view the chart and analyze the data.
- **Customizable Appearance:** The system allows the actor to customize the appearance of the chart, such as changing the colors, font sizes, and labels. This allows the actor to create a chart that is visually appealing and easy to understand.

3.4.1.1 Use Case Diagram

The use case diagram of "Custom Issue Gadget" represented by figure 3.11.

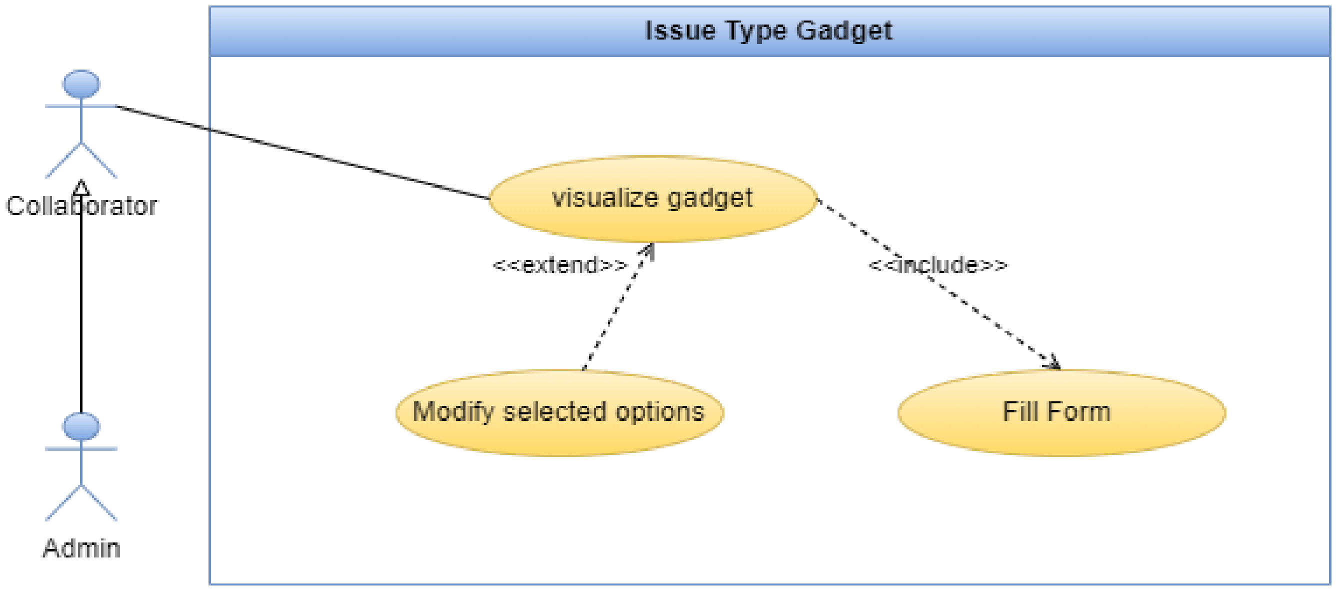


Figure 3.11 Use case diagram "Custom issue Gadget"

3.4.2 Conception of sprint 3

During the conceptual phase of Sprint 3, we have created several detailed diagrams to provide a visual representation of the sprint management system.

3.4.2.2 Detailed Sequence Diagram

This sequence diagram demonstrates the flow of interactions between the Admin, Forge application (FaaS), and Atlassian System. The admin requests the addition of a gadget, and the Forge application retrieves data from the Atlassian System. It generates a visualization interface based on the data and allows the admin to customize the view using filters. Figure 3.11 represents the sequence diagram of "Custom Issue Gadget".

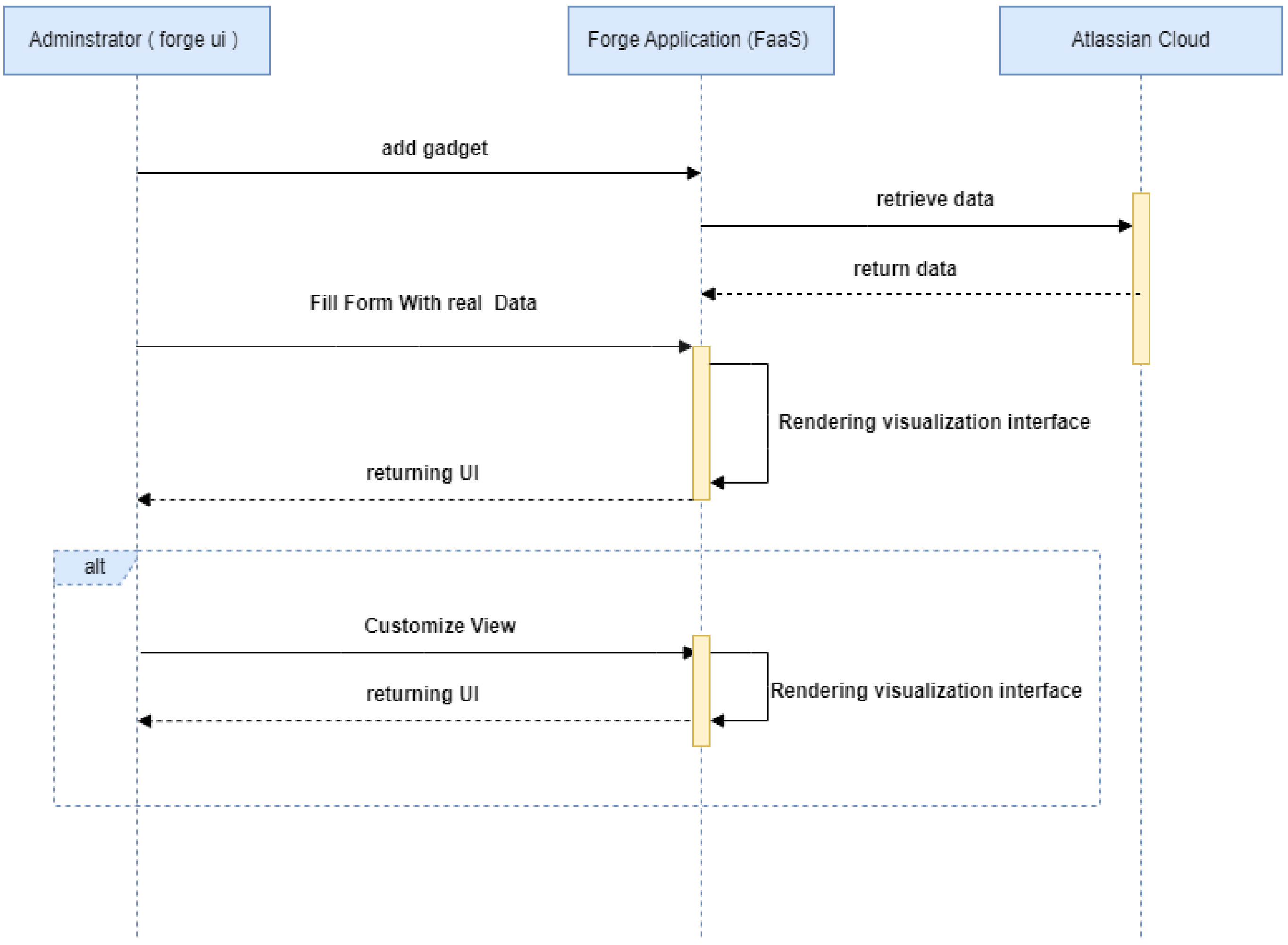


Figure 3.12 sequence diagram of "Custom Issue Gadget".

3.4.2.1 Activity diagram

This diagram focuses on a specific aspect of the system's functionality, which involves visualizing and customizing charts based on selected filters.

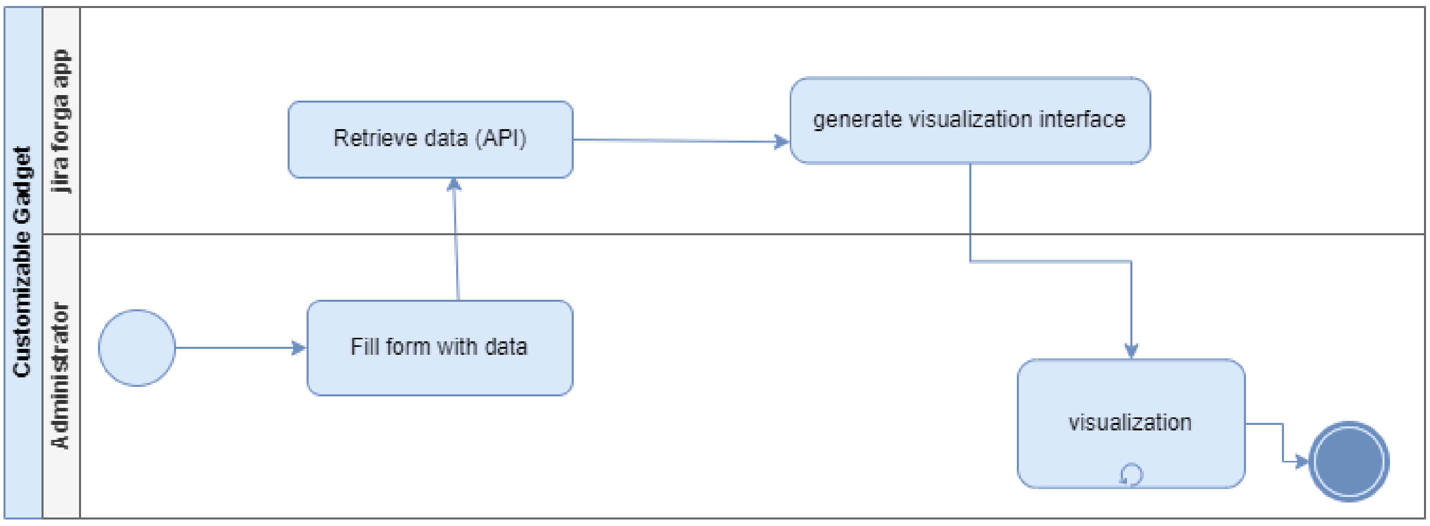
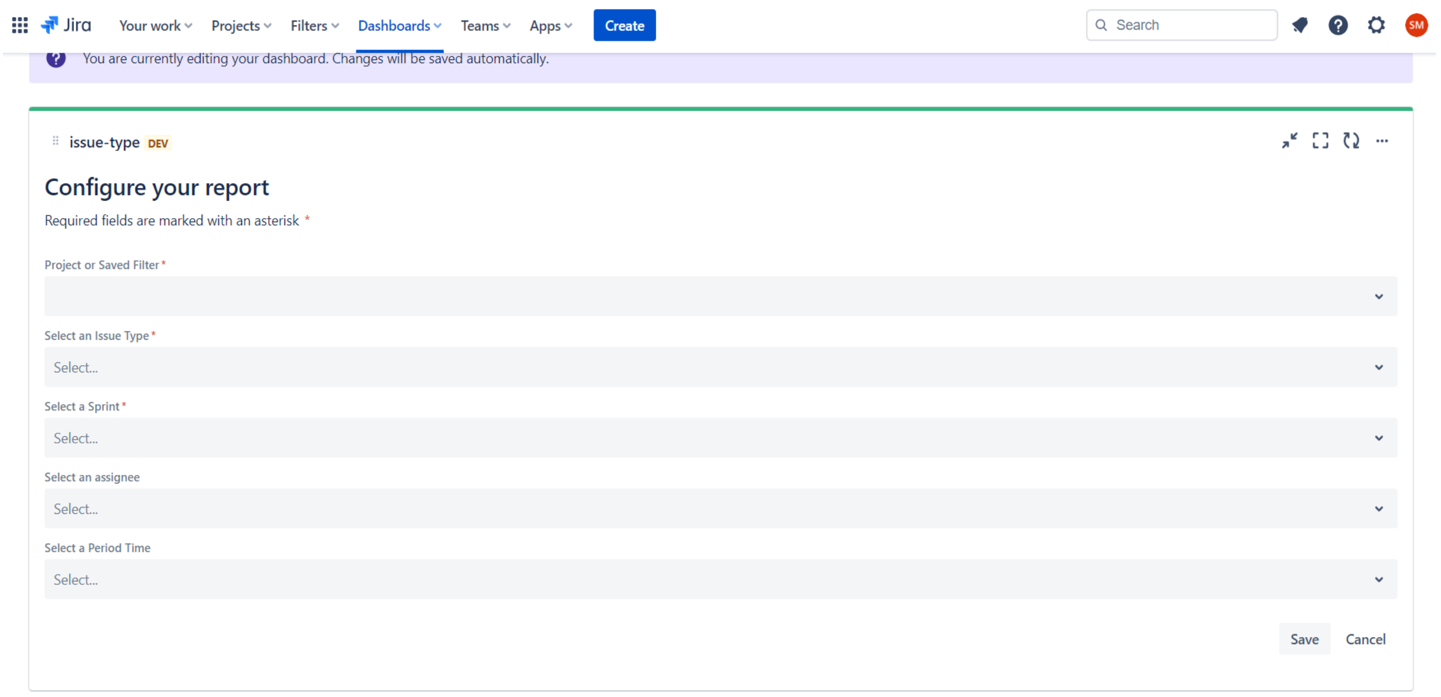


Figure 3.13 Activity diagram OF "Custom Issue Gadget"

3.4.2 Implementation of Sprint 3:

In this section, we will delve into the implementation of custom issue gadget. Figure 3.14 showcases a visually appealing and user-friendly form that offers a comprehensive set of filters for users to customize their issue type view.



You are currently editing your dashboard. Changes will be saved automatically.

issue-type DEV

Configure your report

Required fields are marked with an asterisk *

Project or Saved Filter *

Select an Issue Type *
Select...

Select a Sprint *
Select...

Select an assignee
Select...

Select a Period Time
Select...

Save Cancel

Figure 3.14 Customizable Issue Type Gadget Form

Figure 3.15 showcases a dynamic donut feature illustrating the distribution of assigned tickets across assignees, sprints, projects, and issue types.

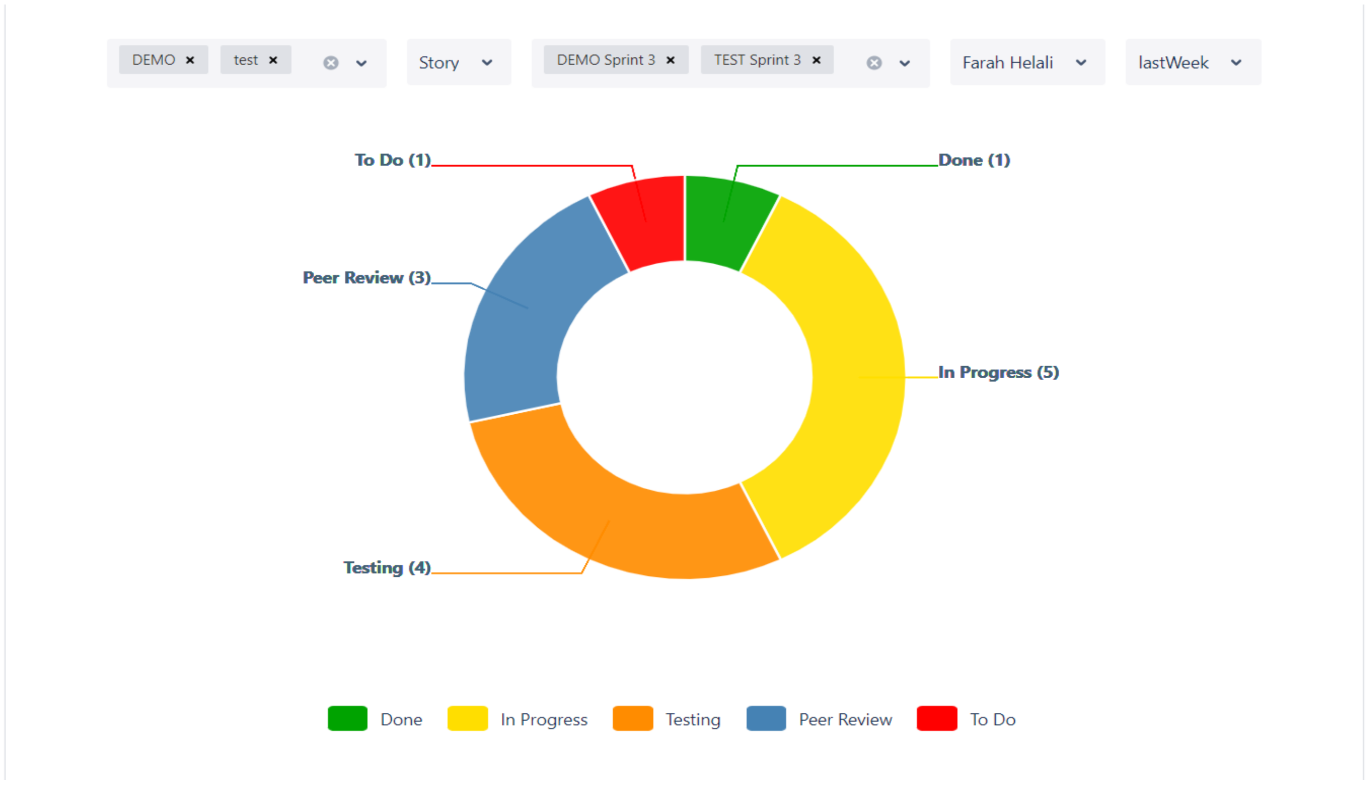

DEMO × test × | Story | DEMO Sprint 3 × TEST Sprint 3 × | Farah Helali | lastWeek

Figure 3.15 Sprint Progress Donut Chart

3.5. Conclusion

Thanks to this chapter, we have presented the analysis and design of the tasks we have performed, as well as the presentation of the interfaces of our application during Release 1. In the following chapter, we will present Release 2 of the proposed solution.

CHAPTER 4

DESIGN AND IMPLEMENTATION OF RELEASE 2

4.1. Introduction

In our final release, we dedicated two sprints to enhancing the settings (admin page) of the app and introducing a global project page view. In Sprint 4, we introduced the Global Dashboard, which provides a comprehensive overview of projects and users within the system. In Sprint 5, we implemented access management and settings functionality within the admin page. This feature allows users to easily view and manage project assignments, track progress, and collaborate with team members on a global scale.

4.2. Realization of Sprint 4: Project Page Overview

4.2.1 Specification of needs

This section covers the requirements, specifications, and technical constraints for Sprint 5, which focuses on developing the project page. The goal of this sprint is to create a clear overview of the project's time spent.

- **Visualize Project Page:** Users will be able to access a visually informative dashboard that presents an overview of the project's time spent. It will display relevant metrics, such as total time spent by issue type and assignee, as well as other project-related data.
- **Export Table in Excel Format:** Users will have the option to export the project's time spent data in a Excel format. This feature allows for easy data extraction and analysis using external tools.
- **Export Table in CSV Format:** Users will have the option to export the project's time spent data in a CSV (Comma-Separated Values) format. This feature allows for easy data extraction and analysis using external tools.
- **Dynamic Project Data Switching:** ensure that the data on the project page is dynamically updated based on the project that the user is currently viewing. Whenever a user switches from one project to another, the data on the page will automatically change to reflect the relevant information for the selected project.

4.2.2.1 Detailed Use Case Diagram

Figure 4.1 represents the use case diagram of the project’s time spent.

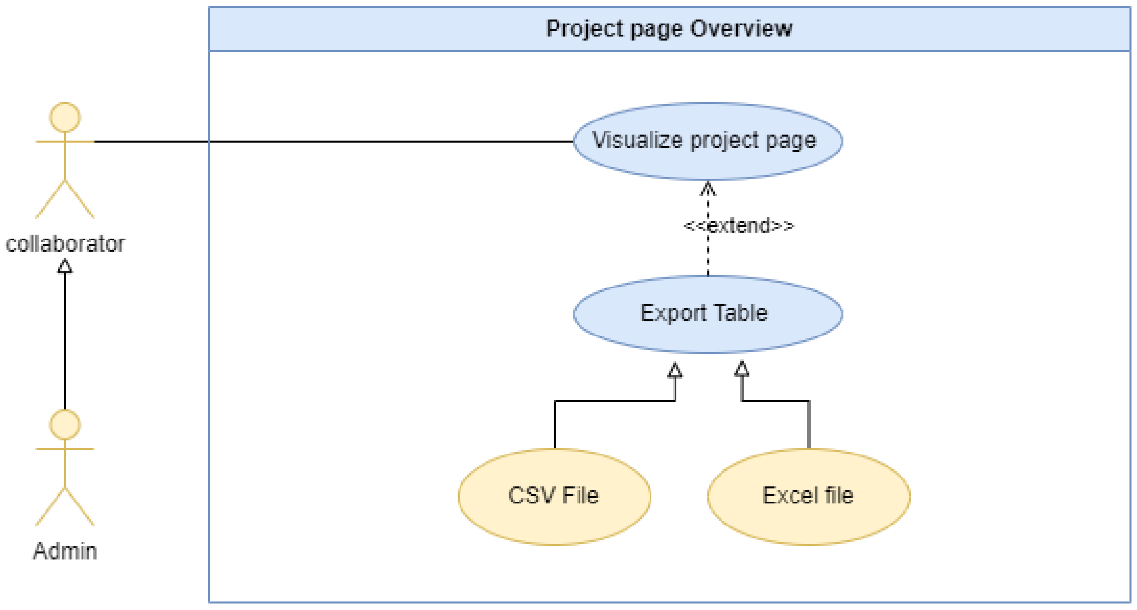


Figure 4.1 use case diagram of the project's time spent.

4.2.2 Conception of sprint 4

4.2.2.2 Detailed Sequence Diagram

This sequence diagram illustrates interactions for the collaborator project overview. Users access a visually informative dashboard displaying project time spent metrics. They can export the data in Excel or CSV formats for external analysis. The system dynamically updates the project data on the page as users switch between projects. Figure 4.2 represents the Sequence diagram of the project's time spent.

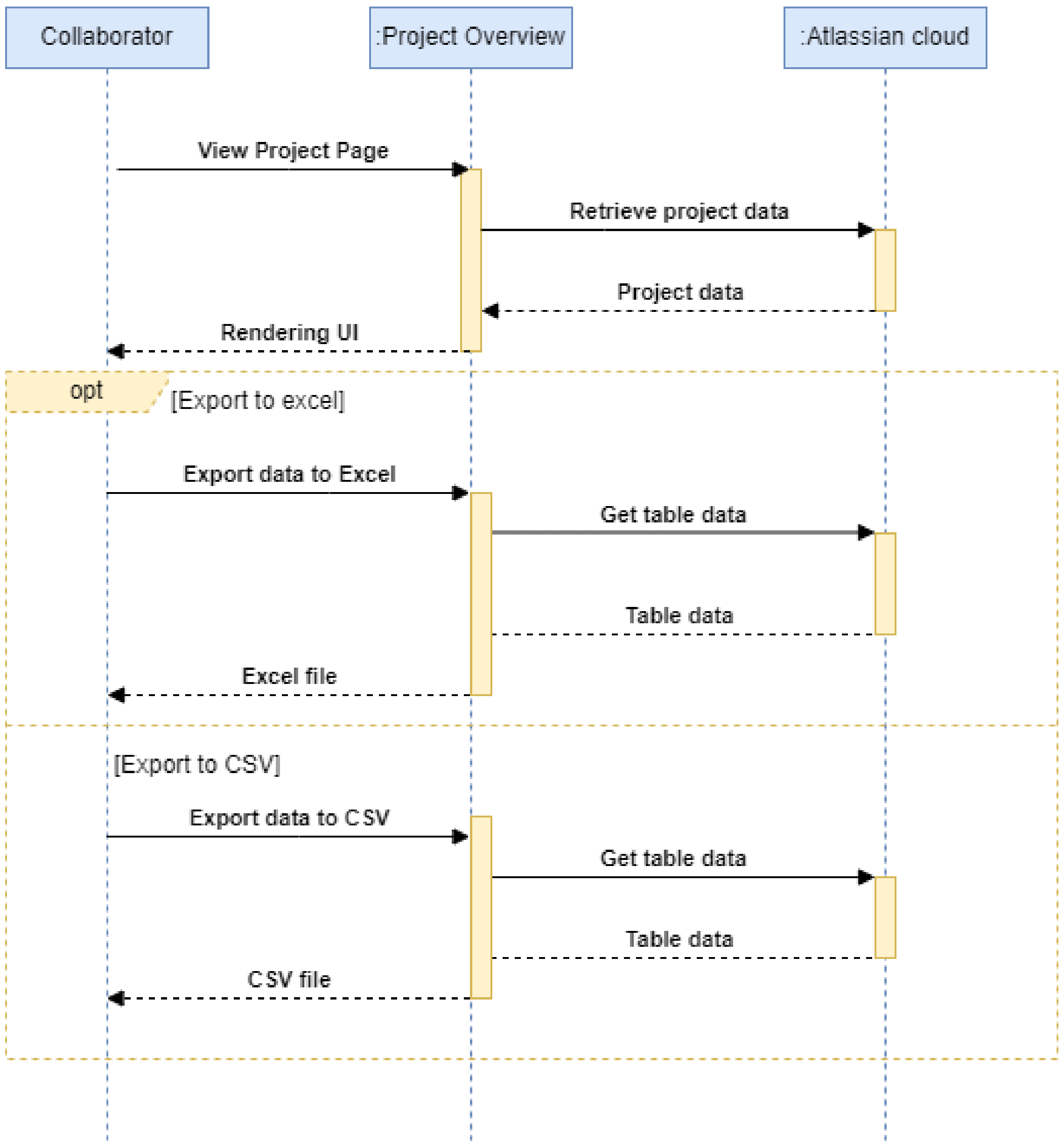


Figure 4.2 Sequence diagram of the project's time spent

4.2.2.3 Class Diagram

The class diagram represents the collaborator project overview, consisting of the "Collaborator," "Project," and "Pie Chart" classes. Each project page is dedicated to displaying project-specific details and generating dynamic pie charts for analyzing time spent by issue type and assignee. The diagram highlights the entities involved in accessing project data and creating interactive pie charts to facilitate effective analysis. Figure 4.3 represents the Class diagram of the project's time spent.

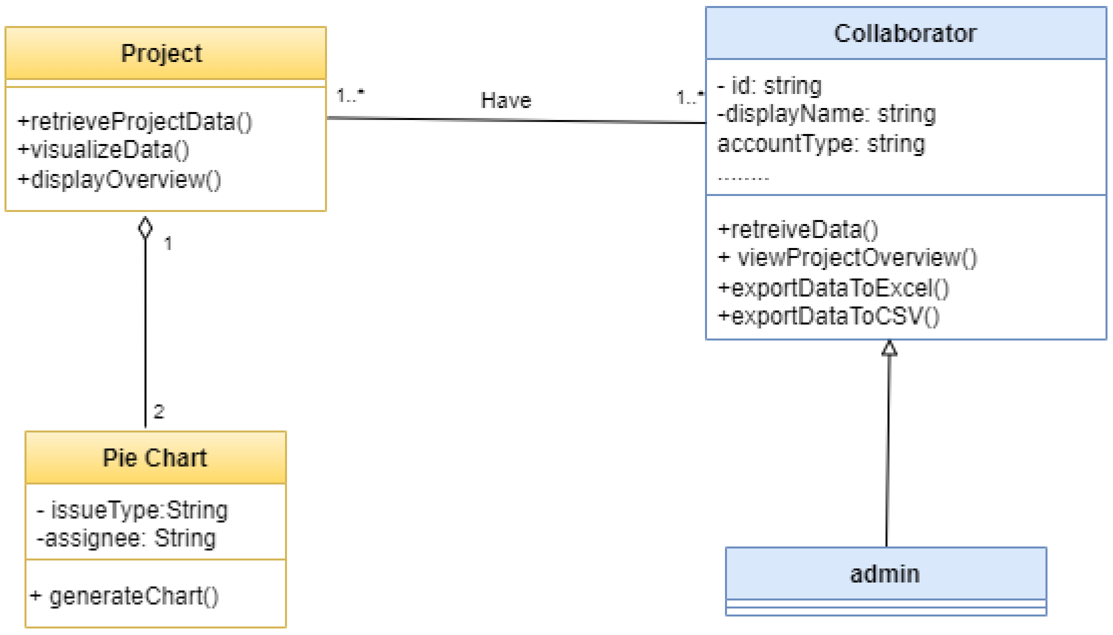
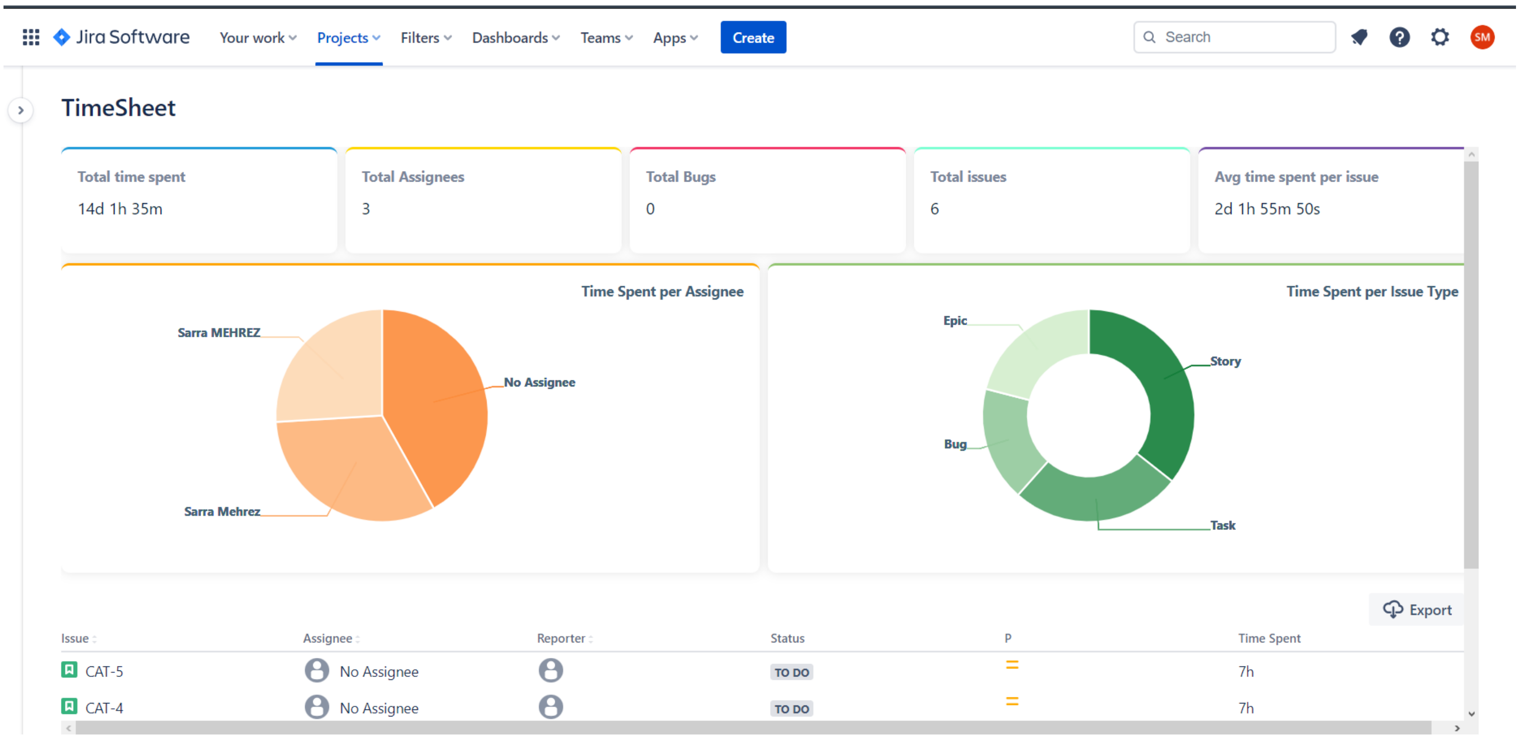


Figure 4.3 Class diagram of the project's time spent

4.2.3 Implementation of Sprint 4:

In this section, we will explore the execution of the project page implementation. Figure 4.4 presents an aesthetically pleasing and intuitive project page design.



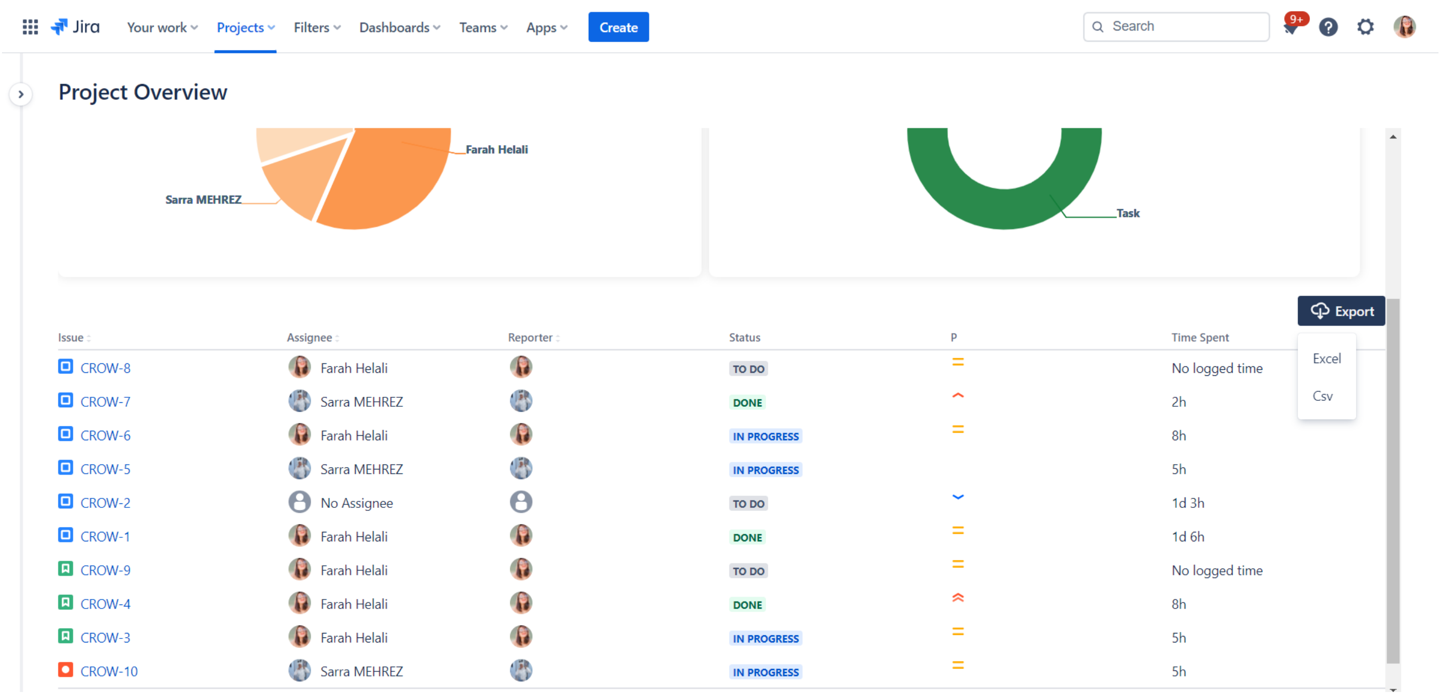
TimeSheet

Total time spent	Total Assignees	Total Bugs	Total issues	Avg time spent per issue
14d 1h 35m	3	0	6	2d 1h 55m 50s

Issue	Assignee	Reporter	Status	P	Time Spent
CAT-5	No Assignee		TO DO	=	7h
CAT-4	No Assignee		TO DO	=	7h

Figure 4.4 Project overview

Figure 4.5 presents an aesthetically pleasing and intuitive project page design with different project.



Issue	Assignee	Reporter	Status	P	Time Spent
CROW-8	Farah Helali		TO DO	=	No logged time
CROW-7	Sarra MEHREZ		DONE	^	2h
CROW-6	Farah Helali		IN PROGRESS	=	8h
CROW-5	Sarra MEHREZ		IN PROGRESS		5h
CROW-2	No Assignee		TO DO	˅	1d 3h
CROW-1	Farah Helali		DONE	=	1d 6h
CROW-9	Farah Helali		TO DO	=	No logged time
CROW-4	Farah Helali		DONE	⌃⌃	8h
CROW-3	Farah Helali		IN PROGRESS	=	5h
CROW-10	Sarra MEHREZ		IN PROGRESS	=	5h

Figure 4.5 Project Overview

4.3 Realization of Sprint 5: Settings and Access management

4.3.1. Specification of needs

This section covers the requirements, specifications, and technical constraints for Sprint 4, which focuses on developing covers the requirements**,** developing the settings and access management functionality for the admin page. This includes implementing access management features and settings in the admin page.

- **Customize Reports:** Users, including admins and collaborators, can customize their reports by selecting specific projects and applying filters. They can focus on relevant work logs and choose the date ranges for report delivery. Additionally, users can select the groups to which they want to send their work logs, facilitating effective communication and collaboration. This feature empowers users to personalize their reports, ensuring that the right information is shared with the intended recipients.
- **Save Settings:** Users can save their customized report settings by clicking the "Save Settings" button. This securely sends the data to a server with a dedicated database, where it is stored and processed to ensure future reports are delivered according to the user's preferences. This feature provides convenience and eliminates the need for repetitive adjustments, allowing users to maintain their desired report configurations effortlessly.
- **Access Permissions Interface:** The admin has exclusive access to the permissions interface, enabling them to manage work log permissions. Within this interface, the admin can define who has access to view work logs within the system. They can assign different roles, such as admin or collaborator, to individual users and configure specific permissions accordingly. This feature ensures that the admin

maintains complete control over work log access and actions, safeguarding the system's security and integrity. Additionally, the admin can specify who can see other users' work logs, allowing for granular control over visibility.

4.3.2.1 Detailed Use Case Diagram

The Use Case Diagram of the Administrator Settings page presented in Figure 4.6.

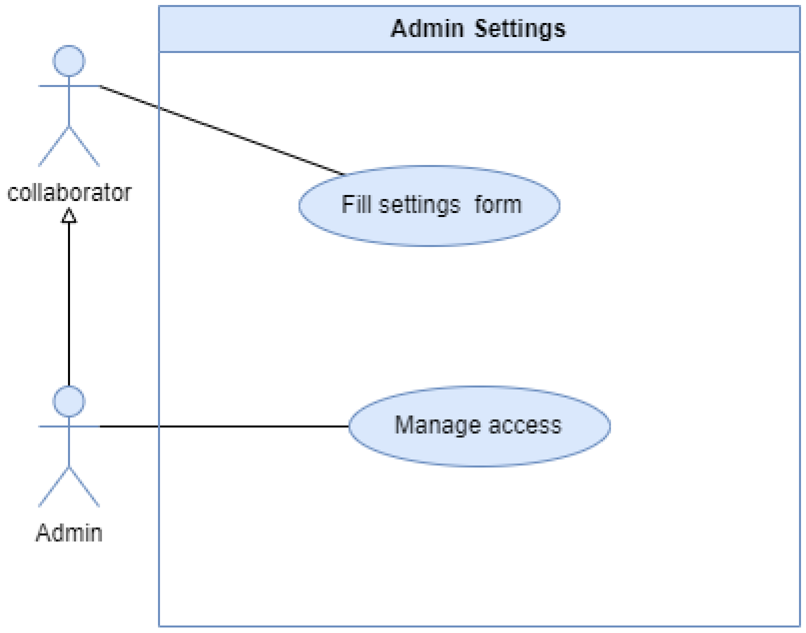


Figure 4.6 Use case Diagram of admin settings

4.3.2. Conception of needs

4.3.2.2 Detailed Sequence Diagram of use case “Fill Settings form”

This sequence diagram showcases the interactions for the "Fill Settings form" use case. Users, including admins and collaborators, can customize reports by selecting projects and applying filters. They can focus on relevant work logs, set date ranges, and choose recipient groups for the reports. Saving settings securely stores the data in a dedicated database, ensuring future reports align with user preferences and reducing repetitive adjustments. The Sequence diagram of the use case “fill settings form “is presented in Figure 4.7.

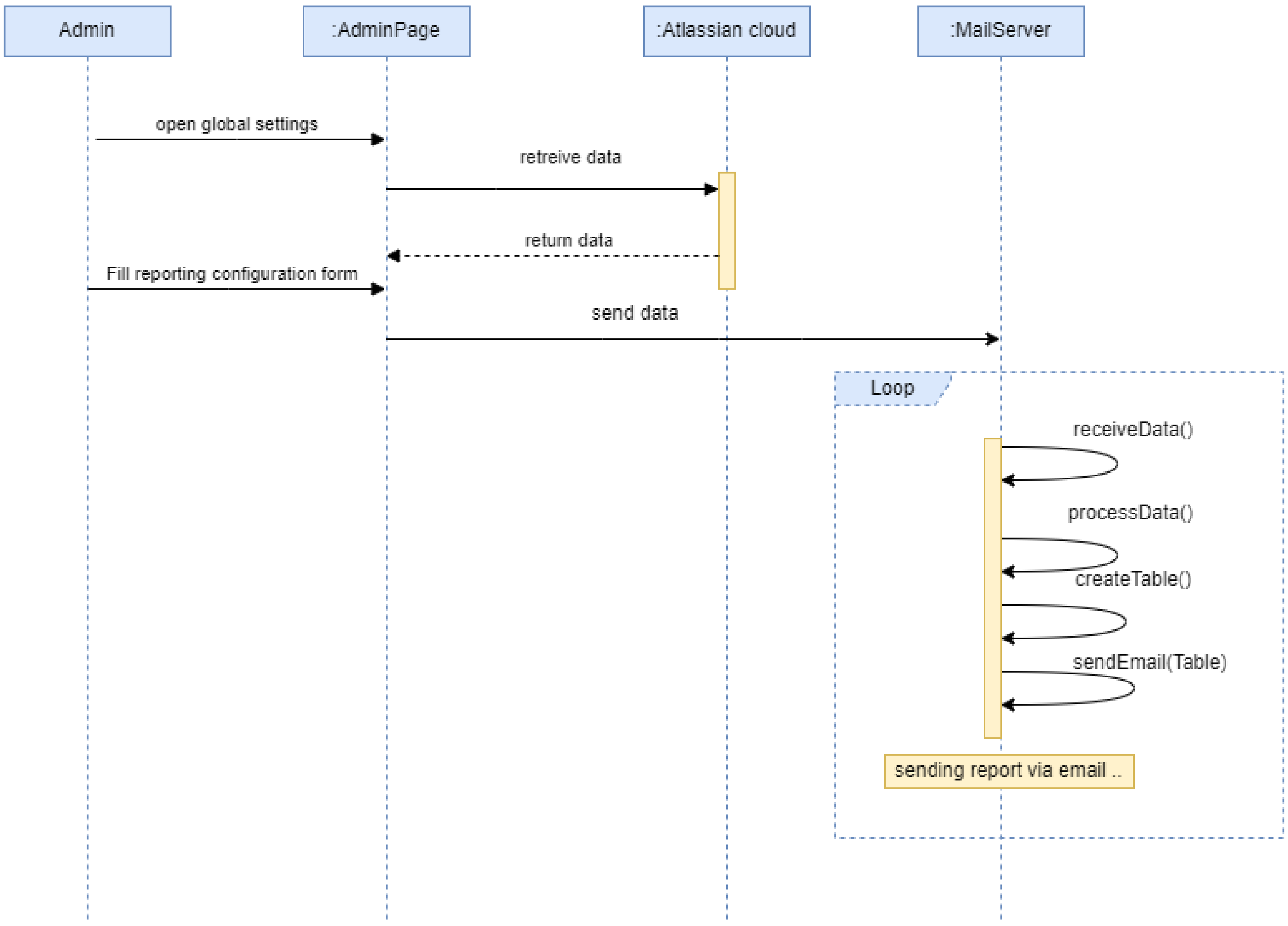


Figure 4.7 Sequence Diagram of "Fill Settings form"

4.4.2.3 Detailed Sequence Diagram of use case "manage permissions"

This sequence diagram illustrates the interactions for the "Manage Permissions" use case. The admin accesses the permissions interface to control work log access. They assign roles (admin, collaborator) and configure specific permissions for users. This feature grants complete control over work log access, enhances system security, and allows fine-grained visibility control, including the ability to specify who can view other users' work logs. The Sequence diagram of the use case "manage permissions "is presented in Figure 4.8.

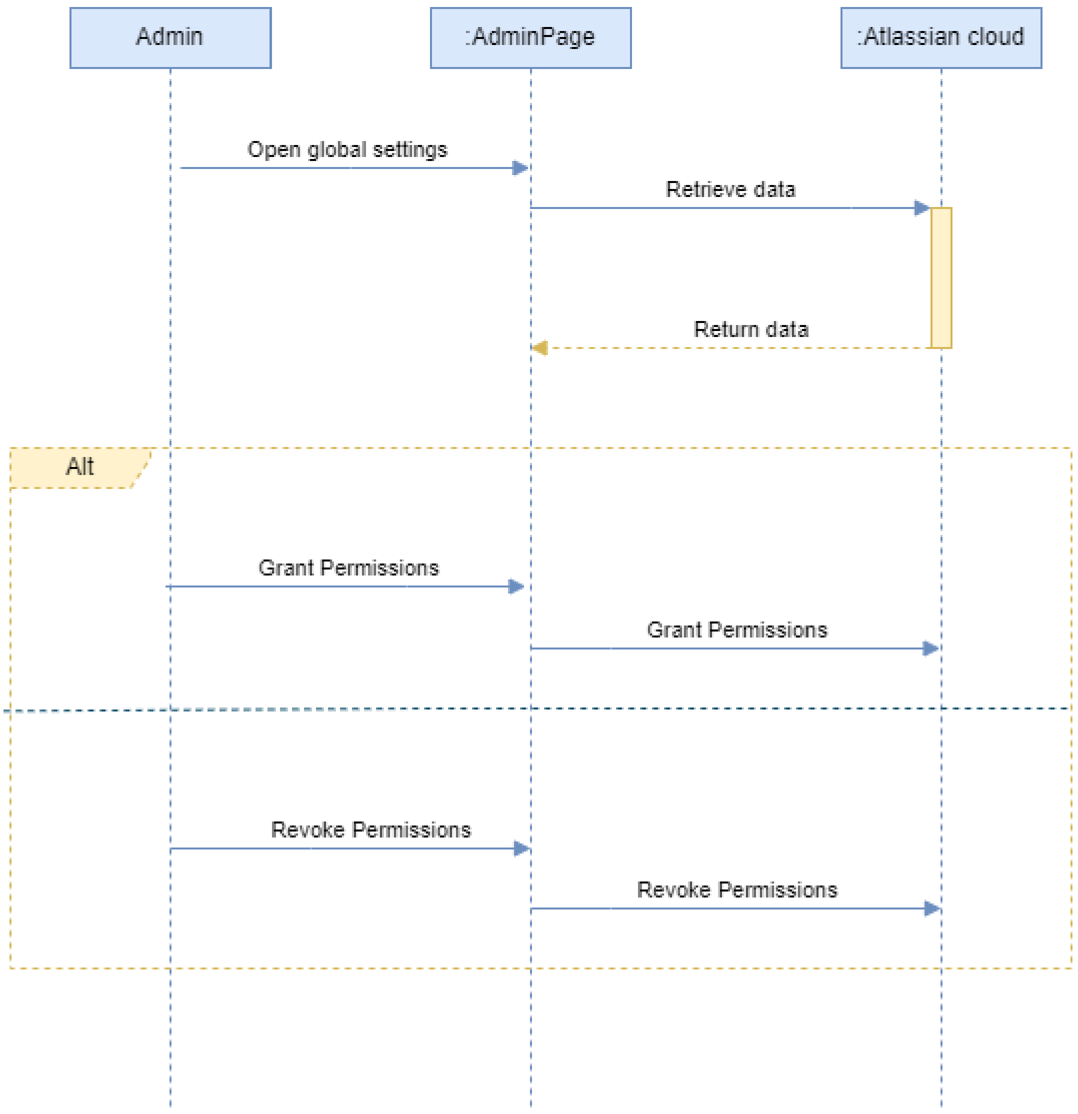


Figure 4.8 Sequence diagram of the use case "manage permissions "

4.3.2.4 Class diagram

In this diagram, the "Collaborator" class represents the individual users who can access and customize the settings. The "Role" class represents the different roles that can be assigned to collaborators, such as admin or regular user. The "Group" class represents the recipient groups to which the customized reports can be sent. The "Settings" class represents the settings data that is customized by the collaborators. Finally, the "Permission" class represents the specific permissions that can be assigned to users for accessing and modifying the app. The Class diagram of the admin page is presented in Figure 4.8.

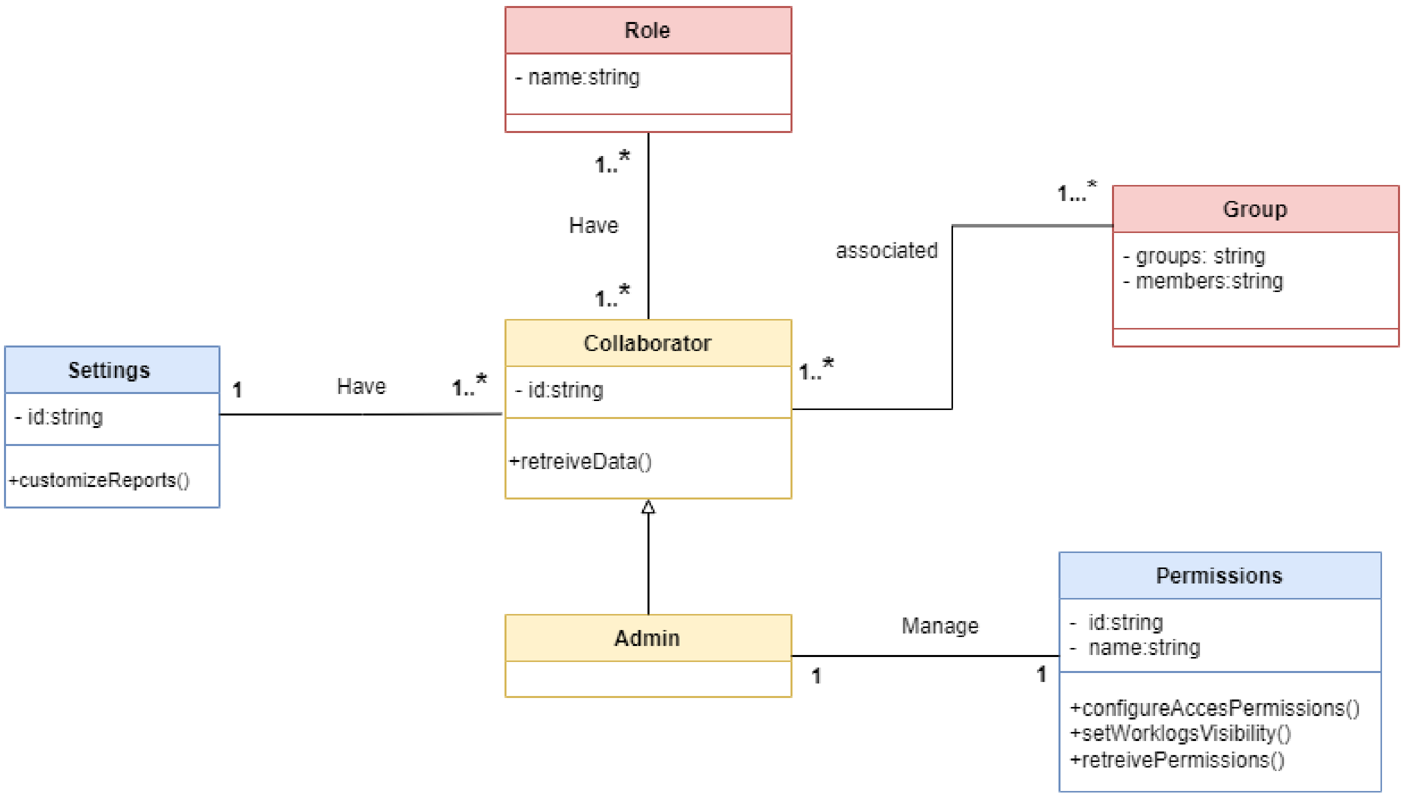


Figure 4.9 Class diagram of the admin page.

4.3.3 Implementation of Sprint 5:

Figure 4.10 showcases a settings admin page interface that is visually appealing and user-friendly in its design.

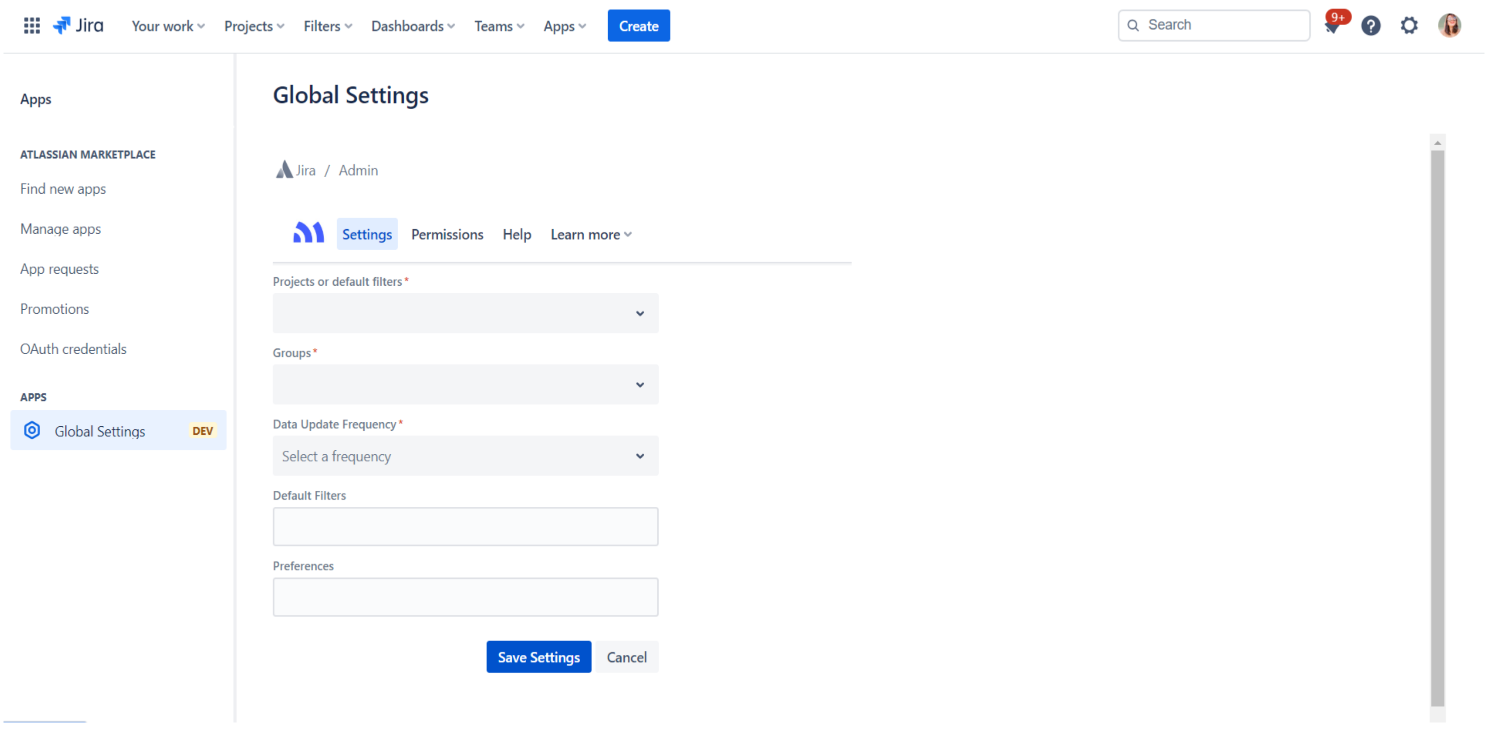
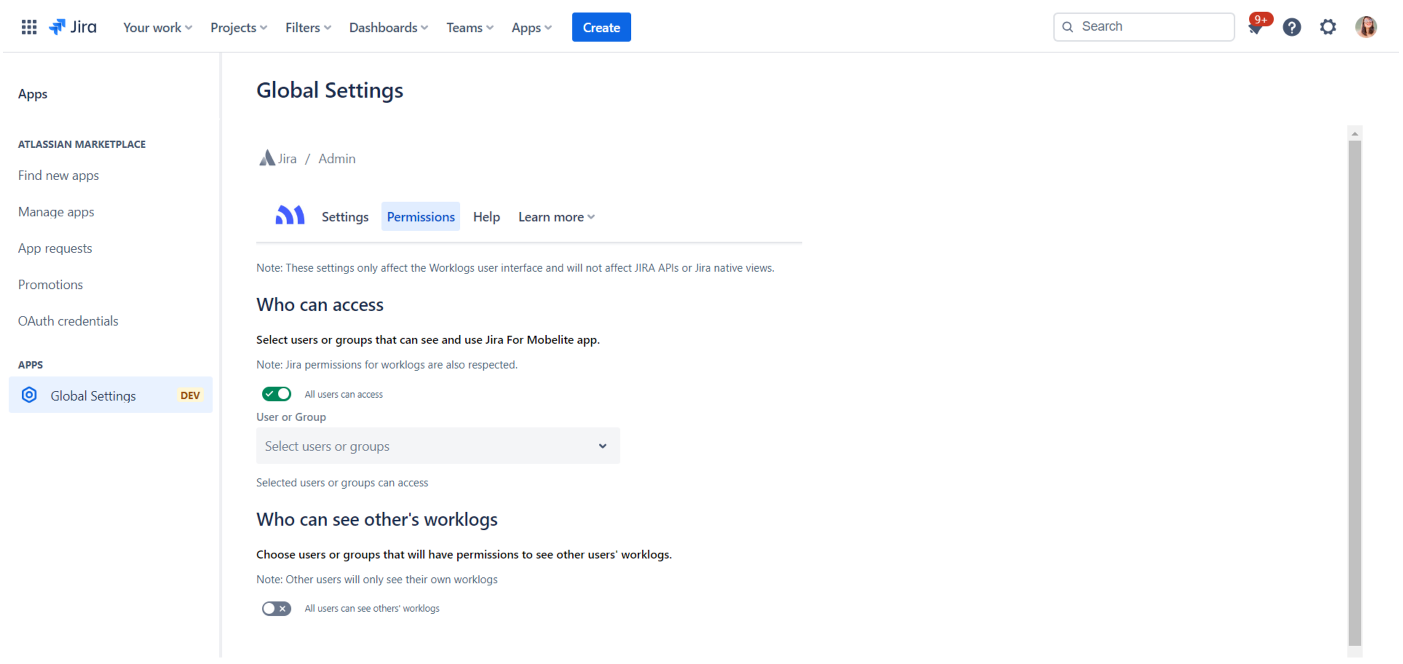


Figure 4.10 Settings interface

Figure 4.11 showcases a permissions admin page interface.



Global Settings

Jira / Admin

Note: These settings only affect the Worklogs user interface and will not affect JIRA APIs or Jira native views.

Who can access

Select users or groups that can see and use Jira For Mobelite app.

Note: Jira permissions for worklogs are also respected.

✓ All users can access

User or Group

Select users or groups

Selected users or groups can access

Who can see other's worklogs

Choose users or groups that will have permissions to see other users' worklogs.

Note: Other users will only see their own worklogs

✕ All users can see others' worklogs

Figure 4.11 Permissions interface

4.3.4. Conclusion

Thanks to this chapter, we have presented the analysis and design of the tasks that we have carried out, as well as the presentation of the interfaces of our application during release 2.

CONCLUSION AND PROSPECTS

This work was carried out within "Mobelite" as part of our end of studies project to obtain the bachelor's degree in computer science from the Higher Institute of Computer Science and Mathematics of Monastir. During our internship, we have managed to design and develop an Atlassian plugin "Jira For Mobelite" that has empowered the Mobelite team with powerful features, enabling them to visualize project data, generate customized reports, and leverage real-time data visualization capabilities.

This project offered a valuable opportunity for our growth, both technically and personally. From a technical perspective, we were able to successfully develop a Jira application using Forge platform, applying the knowledge we gained from ISIMM. This endeavor required us to become proficient in various technologies such as Atlaskit, React.js, D3.js, REST API, Node.js, and more. Becoming an Atlassian developer was challenging yet rewarding. This experience has also provided us with a comprehensive understanding of the professional world and emphasized the continuous learning needed in this dynamic field.

As we look ahead, we are excited about the prospects of our project, as Atlassian Forge is set to release new features in the upcoming months. We are eagerly anticipating the chance to incorporate these enhancements, which include permissions, security, and AWS integration, into our application. By doing so, we aim to bolster the security measures, enhance the user experience, and improve the management of data within our application.

Bibliography

[1] Anon., s.d. *Atlaskit.* [En ligne]
Available at: https://atlaskit.atlassian.com/ visited on 05/18/2023

[2] Anon., s.d. *Atlassian Connect vs. Forge — which way faster to effective Cloud app development?.* [En ligne]
Available at: https://medium.com/deviniti-technology-driven-blog visited on 03/20/2023

[3] Asma KEREKNI, "*Conception des Systèmes d'Information",* 2021/2022, ISIM Monastir

[4] Atlassian developer, s.d. *forge security.* [En ligne]
Available at: https://developer.atlassian.com/platform/forge/security/ visited on 04/03/2023

[5] Atlassian, n.d. *Atlaskit.* [Online], Available at: https://atlaskit.atlassian.com/, visited on 05/20/2023

[6] Atlassian, s.d. *Forge architecture.* [En ligne]
Available at: https://www.atlassian.com/atlascamp/watch-sessions/2019/forge-preview/forge-under-the-hood visited on 06/3/2023

[7] Medium, s.d. *Guide to Scrum Methodology by atlassian.* [En ligne] Available at:
https://www.atlassian.com/agile/scrum visited on 03/15/2023