Elisabet Pujol Ventosa

AUTOMATIC PROJECT EXECUTION CONTROL BOARD

FINAL DEGREE PROJECT

Supervised by
Inés Rodrigo Gómez and David Gàmez Alari

Bachelor's degree in computer engineering

UNIVERSITAT ROVIRA I VIRGILI

Tarragona
2025

Acknowledgements

I would like to express my sincere gratitude to all the people who have supported me throughout the development of this project.

First and foremost, I want to thank my supervisors, Inés Rodrigo and David Gàmez, for their guidance, patience, and valuable feedback during each stage of the project. Their support and expertise have been fundamental to completing this work.

I would also like to thank Lear Corporation, and in particular the people from the different departments who supported me throughout the development of this project.

I am particularly grateful to the different teams within the Engineering Department for their guidance and technical support. In particular, I would like to thank Jaume Segarra's Team, Felipe's Team, and Xavier Areste's Team for their collaboration and valuable contributions during this process.

To my classmates, friends, and the trainees at Lear, thank you for your help, motivation, and for sharing this academic journey with me.

Finally, A special thank you goes to my family and my partner for their unwavering support, encouragement, and understanding throughout this process. Without them, none of this would have been possible.

Abstract

In the realm of data management, team leaders often face challenges in gathering all the necessary data to create a comprehensive overview of their projects. This difficulty arises primarily because the information is scattered across various platforms and comes in different file formats, such as Excel or JSON. Additionally, organizing and correlating this diverse data is a complex task. To address these issues, we developed a dashboard using the Power BI tool.

This innovative dashboard centralizes all information, effectively eliminating the aforementioned problems. It structures the data and provides clear visualizations. Power BI's user-friendly interface enhances interaction and allows the integration of multiple data sources into a single dashboard.

To efficiently manage and filter the data, we utilized the Python programming language. Python enables us to handle various types of data and filter it according to our needs, such as dividing the data by project and storing it in a shared folder. This data is then directly connected to the dashboard, facilitating easy project-based integration.

Furthermore, we can automate the refresh of all data within the dashboard using the Power BI Service platform. This automation ensures that each panel within the dashboard is updated regularly. The panels are color-coded to indicate their status, highlighting areas that require urgent attention.

Resum

En l'àmbit de la gestió de dades, els líders d'equip sovint s'enfronten a dificultats a l'hora de recopilar tota la informació necessària per obtenir una visió global dels seus projectes. Aquesta dificultat sorgeix principalment perquè la informació es troba dispersa en diverses plataformes i en diferents formats de fitxer, com ara Excel o JSON. A més, organitzar i correlacionar aquestes dades tan diverses és una tasca complexa. Per fer front a aquests problemes, es va desenvolupar un panell de control utilitzant l'eina Power BI.

Aquest innovador panell centralitza tota la informació, eliminant de manera efectiva els problemes esmentats. Estructura les dades i proporciona visualitzacions clares. La interfície intuïtiva de Power BI en facilita la interacció i permet la integració de múltiples fonts de dades en un únic panell.

Per gestionar i filtrar les dades de manera eficient, s'ha utilitzat el llenguatge de programació Python. Python permet gestionar diferents tipus de dades i filtrar-les segons les necessitats, com ara dividir-les per projecte i emmagatzemar-les en una carpeta compartida. Aquestes dades es connecten directament al panell, facilitant així una integració senzilla per projecte.

A més, és possible automatitzar l'actualització de totes les dades del panell mitjançant la plataforma Power BI Service. Aquesta automatització assegura que cada panell es mantingui actualitzat de manera regular. Els panells utilitzen un sistema de codificació per colors per indicar-ne l'estat, destacant aquelles àrees que requereixen una atenció immediata.

Resumen

En el ámbito de la gestión de datos, los líderes de equipo suelen enfrentarse a desafíos a la hora de recopilar toda la información necesaria para obtener una visión global de sus proyectos. Esta dificultad surge principalmente porque la información se encuentra dispersa en diversas plataformas y en distintos formatos de archivo, como Excel o JSON. Además, organizar y correlacionar estos datos tan variados representa una tarea compleja. Para abordar estos problemas, se desarrolló un panel de control utilizando la herramienta Power BI.

Este innovador panel centraliza toda la información, eliminando eficazmente los problemas mencionados. Estructura los datos y proporciona visualizaciones claras. La interfaz intuitiva de Power BI mejora la interacción y permite la integración de múltiples fuentes de datos en un único panel.

Para gestionar y filtrar los datos de manera eficiente, se empleó el lenguaje de programación Python. Python permite manejar distintos tipos de datos y filtrarlos según las necesidades del proyecto, como dividir la información por proyecto y almacenarla en una carpeta compartida. Estos datos se conectan directamente al panel, lo que facilita una integración sencilla basada en cada proyecto.

Además, es posible automatizar la actualización de todos los datos del panel mediante la plataforma Power BI Service. Esta automatización garantiza que cada panel se mantenga actualizado de forma regular. Los paneles utilizan un sistema de codificación por colores para indicar su estado, destacando aquellas áreas que requieren atención urgente.

Table of Contents

1 INTRODUCTION AND OBJECTIVES 7

2 REQUIREMENTS 8

2.1 FUNCTIONAL REQUIREMENTS 8

2.1.1 Guide of the user 8

2.1.2 Glossary of Requirements 8

2.1.3 Business Rules 8

2.1.4 Use Cases 9

2.2 NON-FUNCTIONAL REQUIREMENTS 15

3 REQUIREMENT ANALYSIS 16

3.1 CLASS DIAGRAM OF ENTITIES 16

3.2 TRANSLATION OF ACTIVITIES 17

3.2.1 Use Case 01. Open Dashboard 17

3.2.2 Use Case 02. Milestones Panel 18

3.2.3 Use Case 03. Staffing Panel 20

3.2.4 Use Case 04. Requirements Panel 21

3.2.5 Use Case 05. Change Management Panel 23

3.2.6 Use Case 06. Defects Panel 24

3.2.7 Use Case 07. Quality Panel 26

3.2.8 Use Case 08. Budget Panel 27

3.2.9 Use Case 09. CAPEX Panel 29

3.2.10 Use Case 10. Expenses Panel 30

3.2.11 Use Case 11. Compliance Panel 32

3.2.12 Use Case 12. MCM Panel 33

4 DESIGN 35

4.1 DESIGN OF ARCHITECTURE 35

4.2 DESIGN OF THE INTERFACE 36

4.2.1 Global View 36

4.2.2 LearPro Milestones and Deliverables 37

4.2.3 Staffing 38

4.2.4 Requirements 40

4.2.5 Change Management 41

4.2.6 Defects 42

4.2.7 Quality Assurance 44

4.2.8 Budget 45

4.2.9 Lifetime CAPEX 46

4.2.10 Lifetime Expenses 47

4.2.11 Compliance Indexes 49

4.2.12 Material Cost Management 50

5 IMPLEMENTATION 52

5.1 TOOL SELECTION 52

5.1.1 Basic Architecture 52

5.1.2 Data Storage Selection 54

6 EVALUATION 56

7 PROJECT PLANNING 61

8 CONCLUSION 0

8.1 Achievement of Objectives 0
8.2 Additional Objectives Identified During Development 1
8.3 Future Steps 1

Figures Index

Figure 2.1. Use Case Diagram 9
Figure 3.1. Entity Diagram 16
Figure 3.2. Activity Diagram for Use Case 01 17
Figure 3.3. Class Diagram for Use Case 01 17
Figure 3.4. Sequence Diagram for Use Case 01 18
Figure 3.5. Activity Diagram for Use Case 02 18
Figure 3.6. Class Diagram for Use Case 02 19
Figure 3.7. Sequence Diagram for Use Case 02 19
Figure 3.8. Activity Diagram for Use Case 03 20
Figure 3.9. Class Diagram for Use Case 03 20
Figure 3.10. Sequence Diagram for Use Case 03 21
Figure 3.11. Activity Diagram for Use Case 04 21
Figure 3.12. Class Diagram for Use Case 04 22
Figure 3.13. Sequence Diagram for Use Case 04 22
Figure 3.14. Activity Diagram for Use Case 05 23
Figure 3.15. Class Diagram for Use Case 05 23
Figure 3.16. Sequence Diagram for Use Case 05 24
Figure 3.17. Activity Diagram for Use Case 06 24
Figure 3.18. Class Diagram for Use Case 06 25
Figure 3.19. Sequence Diagram for Use Case 06 25
Figure 3.20. Activity Diagram for Use Case 07 26
Figure 3.21. Class Diagram for Use Case 07 26
Figure 3.22. Sequence Diagram for Use Case 07 27
Figure 3.23. Activity Diagram for Use Case 08 27
Figure 3.24. Class Diagram for Use Case 08 28
Figure 3.25. Sequence Diagram for Use Case 08 28
Figure 3.26. Activity Diagram for Use Case 09 29
Figure 3.27. Class Diagram for Use Case 09 29
Figure 3.28. Sequence Diagram for Use Case 09 30
Figure 3.29. Activity Diagram for Use Case 10 30
Figure 3.30. Class Diagram for Use Case 10 31
Figure 3.31. Sequence Diagram for Use Case 10 31
Figure 3.32. Activity Diagram for Use Case 11 32
Figure 3.33. Class Diagram for Use Case 11 32
Figure 3.34. Sequence Diagram for Use Case 11 33
Figure 3.35. Activity Diagram for Use Case 12 33
Figure 3.36. Class Diagram for Use Case 12 34
Figure 3.37. Sequence Diagram for Use Case 12 34
Figure 4.1. System Architecture 35
Figure 4.2. Global view of the dashboard 37
Figure 4.3. Milestone's Global View 37
Figure 4.4. Milestone's Expanded View 38
Figure 4.5. Staffing's Global View 39
Figure 4.6. Staffing's Expanded View 39
Figure 4.7. Requirements Both View s 40
Figure 4.8. Change Management's Global View 41
Figure 4.9. Change Management's Expanded View 42
Figure 4.10. Defects Global View 43
Figure 4.11. Defects Expanded View 43
Figure 4.12. Quality Assurance's Global View 44
Figure 4.13. Quality Assurance's Expanded View 44
Figure 4.14. Budget's Global View 45

FIGURE 4.15. BUDGET'S EXPANDED VIEW 46
FIGURE 4.16. LIFETIME CAPEX'S GLOBAL VIEW 47
FIGURE 4.17. LIFETIME CAPEX'S EXPANDED VIEW 47
FIGURE 4.18. LIFETIME EXPENSES GLOBAL VIEW 48
FIGURE 4.19. LIFETIME EXPENSES EXPANDED VIEW 48
FIGURE 4.20. COMPLIANCE INDEXES GLOBAL VIEW 49
FIGURE 4.21. COMPLIANCE INDEXES EXPANDED VIEW 49
FIGURE 4.22. MCM GLOBAL VIEW 50
FIGURE 4.23. MCM EXPANDED VIEW 51
FIGURE 7.1. GANTT DIAGRAM 0

Table Index

TABLE 5.1. TOOL SELECTION EXPLANATION 53
TABLE 6.1. TEST USE CASE 01. OPEN DASHBOARD 56
TABLE 6.2. TEST USE CASE 02. MILESTONES PANEL 57
TABLE 6.3. TEST USE CASE 03. STAFFING PANEL 57
TABLE 6.4. TEST USE CASE 04. REQUIREMENTS PANEL 57
TABLE 6.5. TEST USE CASE 05. CHANGE MANAGEMENT PANEL 58
TABLE 6.6. TEST USE CASE 06. DEFECTS PANEL 58
TABLE 6.7. TEST USE CASE 07. QUALITY PANEL 59
TABLE 6.8. TEST USE CASE 08. BUDGET PANEL 59
TABLE 6.9. TEST USE CASE 09. CAPEX PANEL 59
TABLE 6.10. TEST USE CASE 10. EXPENSES PANEL 60
TABLE 6.11. TEST USE CASE 11. COMPLIANCE PANEL 60
TABLE 6.12. TEST USE CASE 12. MCM PANE 60

1 Introduction and Objectives

In large-scale companies such as Lear, managing and visualizing project data poses a considerable challenge. This complexity arises from the vast volume of data generated throughout a project's lifecycle, as well as the variety of platforms used across departments. Each platform presents data differently, making it difficult to correlate and consolidate information into a cohesive overview. As a result, understanding the global status of a project becomes inefficient and time-consuming.

The main objective of this project is to develop a unified, interactive dashboard that brings together all relevant data into a single, centralized location. This will facilitate clearer insights into project status, streamline data access, and enhance decision-making processes. By using Power BI as the primary visualization tool, the project seeks to transform scattered, unstructured data into standardized, visually intuitive graphics that allow users to monitor progress and detect potential issues quickly.

To ensure data security and confidentiality, access to the dashboard is restricted to authorized users only, specifically, the individuals directly involved in the project. Another key goal is to improve data correlation, enabling connections between datasets that were previously isolated across different platforms.

Given that raw data is often stored in large, unfiltered Excel files, a significant part of the project involved implementing an effective data filtering process. This ensures that only relevant data for each project is displayed, keeping the dashboard concise and efficient. To achieve this, custom scripts were developed using Python, with each visualization element handled by a dedicated function for better readability and maintainability of the code.

The project was developed in several phases:

1. **Understanding Power BI**: Learning how to connect and visualize data effectively.
2. **Data Filtering**: Determining how to extract and format the necessary information for each project.
3. **Data Storage**: Choosing a reliable and accessible location for storing filtered data.
4. **Dashboard Automation**: Enabling automatic data refresh to keep the dashboard updated daily.

Additionally, visual indicators were implemented to provide a quick assessment of each panel’s status. A color-coded system—green, yellow, and red—was used to represent normal, warning, and critical conditions respectively, allowing users to quickly identify areas that require attention.

In summary, the project aims to:

- Gather all project-related data in a single, accessible location.
- Ensure restricted access to authorized team members only.
- Enable correlation of data previously dispersed across multiple platforms.
- Deliver a user-friendly interface for improved usability.
- Automate data refresh to ensure real-time accuracy.
- Provide quick and secure access to project dashboards.

2 Requirements

In this chapter, the requirements of the project are described, including both functional and non-functional requirements.

2.1 Functional Requirements

Functional requirements define the specific behavior or functions of a system. They describe what the system should do, including tasks, services, and functions the system must perform.

2.1.1 Guide of the user

Log in with your company credentials to access the Power BI dashboard. Once inside, select the relevant project you are working on. After making your selection, you will be able to view different sections, analyze your data, and make necessary adjustments.

You can access an initial section that provides an overview of the project's status, including all the panels of each section. Here, you can review key metrics, identify any critical issues, and see overall performance indicators. You can access the expanded view of each panel, which allows you to see the data in more detail and even display new information. This helps you focus on the most relevant information.

Each panel has a color rule that identifies if the panel is in a critical state or not: green if all is good, yellow if something is somewhat wrong, and red if immediate attention is needed. In some panels, you can click on a link that will take you to another dashboard within the company.

2.1.2 Glossary of Requirements

Dashboard: A web platform used to visualize the data with a visual interface.

2.1.3 Business Rules

1. The user must be authenticated.

2. A user can only access the dashboard if they have access to it.

2.1.4 Use Cases

Diagrams of Use Cases

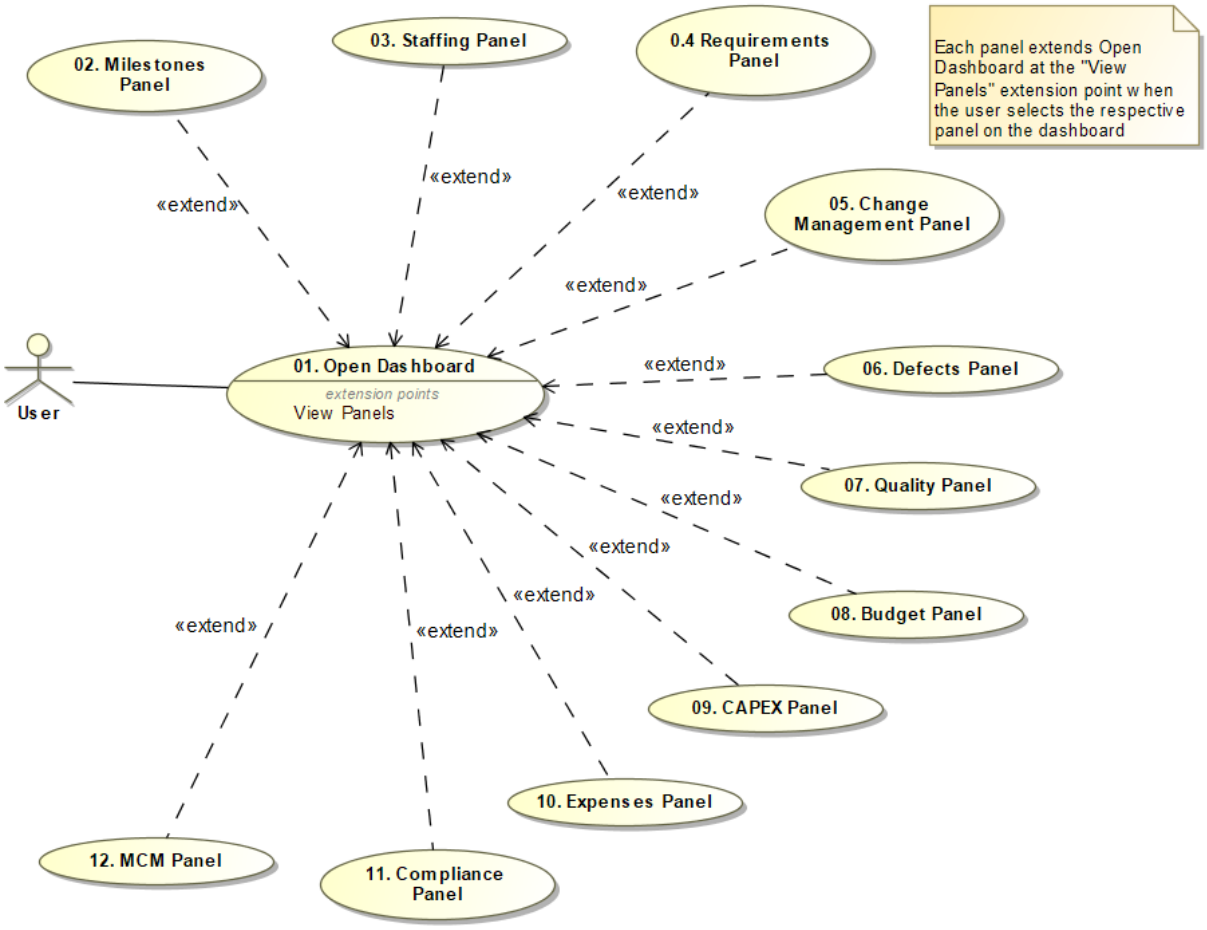


Figure 2.1. Use Case Diagram

Textual Specification of Use Cases

Use Case 01. Open Dashboard

Summary of Functionality: Allows the user to select and open one of the project dashboards available on the Power BI Service platform.

Input parameters: None.

Output parameters: The selected dashboard is displayed.

Actors: **User**.

Precondition: The user must be registered and have authorization to view the dashboard.

Postcondition: The selected dashboard is open and ready for interaction.

Main normal process:

1. The **User** accesses the Power BI Service platform.
2. The **User** clicks on the dashboard they want to open.
3. The system presents the selected dashboard.
4. The **User** interacts with the dashboard.

Process Alternatives and Exceptions:

3a. The system rejects opening the dashboard because the User does not have authorization.

4a. The **User** may select specific panels within the dashboard to view detailed information.

4a1. The system opens the selected panel.

4b. The **User** may click a number shown in a dashboard panel table to view linked item details.

4b1. The system opens the linked item in a new window.

Use Case 02. Milestones Panel

Summary of Functionality: Displays detailed project Milestones information when selected from the Open Dashboard in Power BI Service.

Input parameters: None.

Output parameters: The LearPro Milestones & Deliverables panel is displayed.

Actors: **User**.

Precondition: The Open Dashboard is already open.

Postcondition: The selected dashboard is open and ready for interaction. Also, the User can return to the Open Dashboard at any time.

Main normal process:

1. The **User** selects the Milestones Panel from the Open Dashboard.
2. The system opens the Milestones Panel.
3. The **User** views and interacts with the milestones data.

Process Alternatives and Exceptions:

3a. At any point, the **User** may return to the Open Dashboard.

Use Case 03. Staffing Panel

Summary of Functionality: Displays detailed project Staffing information when selected from the Open Dashboard in Power BI Service.

Input parameters: None.

Output parameters: The Staffing panel is displayed.

Actors: **User**.

Precondition: The Open Dashboard is already open.

Postcondition: The selected dashboard is open and ready for interaction. Also, the **User** can return to the Open Dashboard at any time.

Main normal process:

1. The **User** selects the Staffing Panel from the Open Dashboard.
2. The system opens the Milestones Panel.
3. The **User** views and interacts with the staffing data.

Process Alternatives and Exceptions:

3a. At any point, the **User** may return to the Open Dashboard.

Use Case 04. Requirements Panel

Summary of Functionality: Displays detailed project Requirements information when selected from the Open Dashboard in Power BI Service.

Input parameters: None.

Output parameters: The Requirements panel is displayed.

Actors: **User**.

Precondition: The Open Dashboard is already open.

Postcondition: The selected dashboard is open and ready for interaction. Also, the **User** can return to the Open Dashboard at any time.

Main normal process:

1. The **User** selects the Requirements Panel from the Open Dashboard.
2. The system opens the Requirements Panel.
3. The **User** views and interacts with the requirements data.

Process Alternatives and Exceptions:

3a. At any point, the **User** may return to the Open Dashboard.

Use Case 05. Change Management Panel

Summary of Functionality: Displays detailed project Change Management information when selected from the Open Dashboard in Power BI Service.

Input parameters: None.

Output parameters: The Change Management panel is displayed.

Actors: **User**.

Precondition: The Open Dashboard is already open.

Postcondition: The selected dashboard is open and ready for interaction. Also, the **User** can return to the Open Dashboard at any time.

Main normal process:

1. The **User** selects the Change Management Panel from the Open Dashboard.
2. The system opens the Change Management Panel.
3. The **User** views and interacts with the data.

Process Alternatives and Exceptions:

3a. At any point, the **User** may return to the Open Dashboard.

Use Case 06. Defects Panel

Summary of Functionality: Displays detailed project Defects information when selected from the Open Dashboard in Power BI Service.

Input parameters: None.

Output parameters: The Defects panel is displayed.

Actors: **User**.

Precondition: The Open Dashboard is already open.

Postcondition: The selected dashboard is open and ready for interaction. Also, the **User** can return to the Open Dashboard at any time.

Main normal process:

1. The **User** selects the Defects Panel from the Open Dashboard.
2. The system opens the Defects Panel.
3. The **User** views and interacts with the data.

Process Alternatives and Exceptions:

3a. At any point, the **User** may return to the Open Dashboard.

Use Case 07. Quality Panel

Summary of Functionality: Displays detailed project Quality information when selected from the Open Dashboard in Power BI Service.

Input parameters: None.

Output parameters: The Quality panel is displayed.

Actors: **User**.

Precondition: The Open Dashboard is already open.

Postcondition: The selected dashboard is open and ready for interaction. Also, the **User** can return to the Open Dashboard at any time.

Main normal process:

1. The **User** selects the Quality Panel from the Open Dashboard.
2. The system opens the Quality Panel.
3. The **User** views and interacts with the data.

Process Alternatives and Exceptions:

3a. At any point, the **User** may return to the Open Dashboard.

Use Case 08. Budget Panel

Summary of Functionality: Displays detailed project Budget information when selected from the Open Dashboard in Power BI Service.

Input parameters: None.

Output parameters: The Budget panel is displayed.

Actors: **User**.

Precondition: The Open Dashboard is already open.

Postcondition: The selected dashboard is open and ready for interaction. Also, the **User** can return to the Open Dashboard at any time.

Main normal process:

1. The **User** selects the Budget Panel from the Open Dashboard.
2. The system opens the Budget Panel.
3. The **User** views and interacts with the data.

Process Alternatives and Exceptions:

3a. At any point, the **User** may return to the Open Dashboard.

Use Case 09. CAPEX Panel

Summary of Functionality: Displays detailed project CAPEX information when selected from the Open Dashboard in Power BI Service.

Input parameters: None.

Output parameters: The CAPEX panel is displayed.

Actors: **User**.

Precondition: The Open Dashboard is already open.

Postcondition: The selected dashboard is open and ready for interaction. Also, the **User** can return to the Open Dashboard at any time.

Main normal process:

1. The **User** selects the CAPEX Panel from the Open Dashboard.
2. The system opens the CAPEX Panel.
3. The **User** views and interacts with the data.

Process Alternatives and Exceptions:

3a. At any point, the **User** may return to the Open Dashboard.

Use Case 10. Expenses Panel

Summary of Functionality: Displays detailed project Expenses information when selected from the Open Dashboard in Power BI Service.

Input parameters: None.

Output parameters: The Expenses panel is displayed.

Actors: **User**.

Precondition: The Open Dashboard is already open.

Postcondition: The selected dashboard is open and ready for interaction. Also, the **User** can return to the Open Dashboard at any time.

Main normal process:

1. The **User** selects the Expenses Panel from the Open Dashboard.
2. The system opens the Expenses Panel.
3. The **User** views and interacts with the data.

Process Alternatives and Exceptions:

3a. At any point, the **User** may return to the Open Dashboard.

Use Case 11. Compliance Panel

Summary of Functionality: Displays detailed project Compliance information when selected from the Open Dashboard in Power BI Service.

Input parameters: None.

Output parameters: The Compliance panel is displayed.

Actors: **User**.

Precondition: The Open Dashboard is already open.

Postcondition: The selected dashboard is open and ready for interaction. Also, the **User** can return to the Open Dashboard at any time.

Main normal process:

1. The **User** selects the Compliance Panel from the Open Dashboard.
2. The system opens the Compliance Panel.
3. The **User** views and interacts with the data.

Process Alternatives and Exceptions:

3a. The **User** may click a number shown in the panel table to view linked item details.

3a1. The system opens the linked item in a new window.

3b. At any point, the **User** may return to the Open Dashboard.

Use Case 12 MCM Panel

Summary of Functionality: Displays detailed project MCM information when selected from the Open Dashboard in Power BI Service.

Input parameters: None.

Output parameters: The MCM panel is displayed.

Actors: **User**.

Precondition: The Open Dashboard is already open.

Postcondition: The selected dashboard is open and ready for interaction. Also, the **User** can return to the Open Dashboard at any time.

Main normal process:

1. The **User** selects the MCM Panel from the Open Dashboard.
2. The system opens the MCM Panel.
3. The **User** views and interacts with the data.

Process Alternatives and Exceptions:

3a. At any point, the **User** may return to the Open Dashboard.

2.2 Non-Functional Requirements

Security Requirements

- Data Protection: Ensure that all the confidential data is stored safely and make it only accessible to the people with the corresponding permissions. Only the people working on that project will be able to check the dashboard.

Performance Requirements

- Response time: the response of the system must be fast. To accomplish this, every project has a different dashboard. Also, to make it faster, the data is filtered beforehand and every project in Power BI will only have the data of that project.
- Scalability: Each project has a dashboard that has the same template. For now, we have tested this in three projects. Also, the data is stored in excel files and connected directly to the Power BI.

Reliability Requirements

- Uptime: The data is available anytime of the day.
- Error Rate: While the python script is functioning and the columns or types in the dashboard are not changed, the system will work properly.

Availability Requirements

- Operational Continuity: The dashboard is published online. If the internet functions, the dashboard will be available. Moreover, the dashboard can be downloaded at any moment.

Usability Requirements

- Ease of Use: The interface must be intuitive and coherent to the user. The navigation of the interface is also important and needs to be easy to go through and have a good design.
- Accessibility: Every authenticated user with the right permissions can access the dashboard.

Maintainability Requirements

- Modularity: Each panel of the dashboard connects to its own Excel file and has a dedicated Python function to separate one panel from another. If it is necessary to change the script because we want to update the data displayed on the dashboard, the code is modified locally. Once tested, the updated code is uploaded to the server, ensuring the system continues to operate normally. While the code is being changed, the old one is still executing until we update the version with the new one.
- Documentation: The code is commented on and has a structured design, making it easy to make changes if necessary. Each panel of the dashboard includes an explanation of what is displayed and the intended purpose in the documentation.

3 Requirement Analysis

Now we'll analyze all the different Use Cases specified in the description of the requirements.

3.1 Class diagram of entities

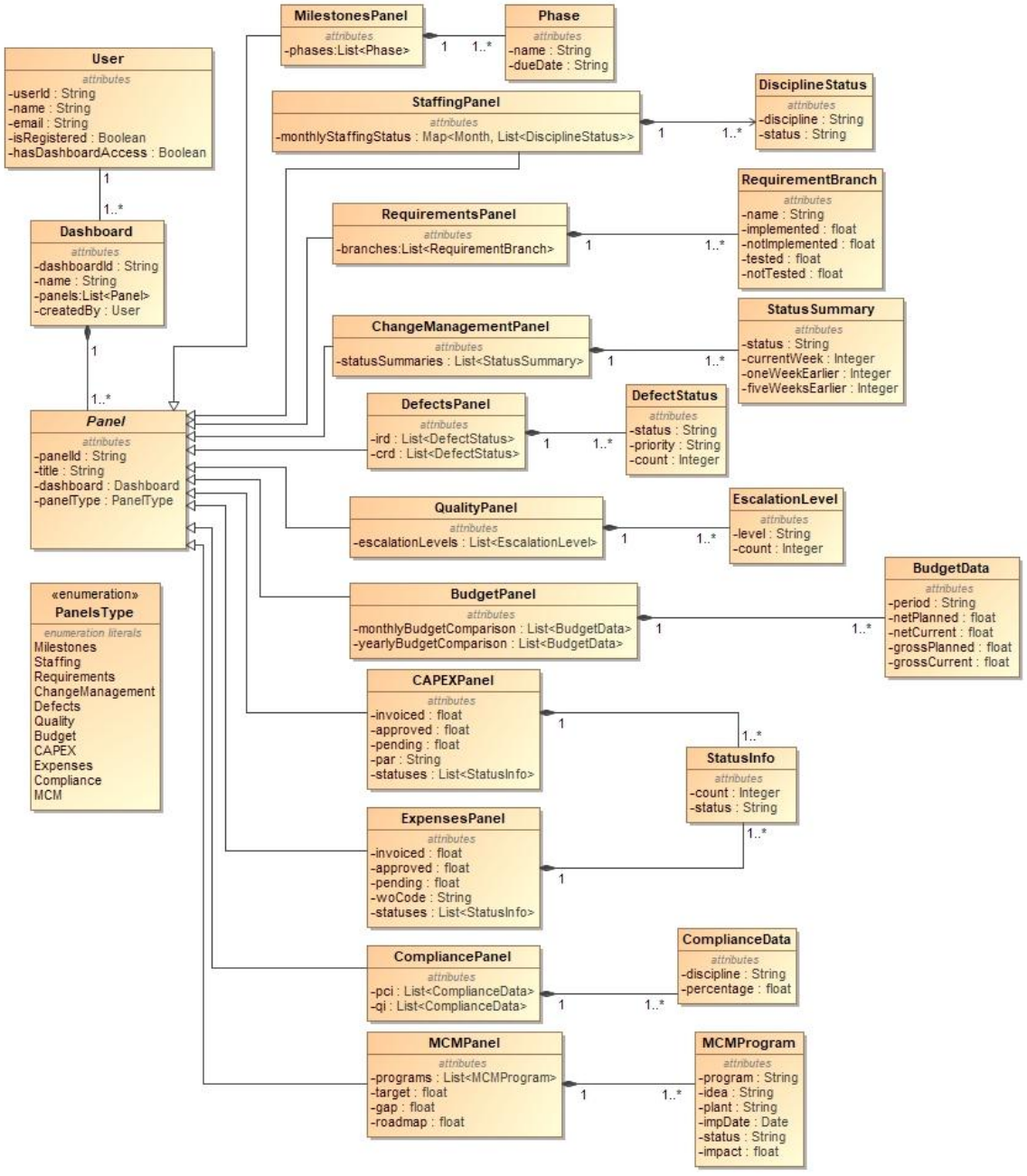


Figure 3.1. Entity Diagram

3.2 Translation of activities

3.2.1 Use Case 01. Open Dashboard

Activity Diagram

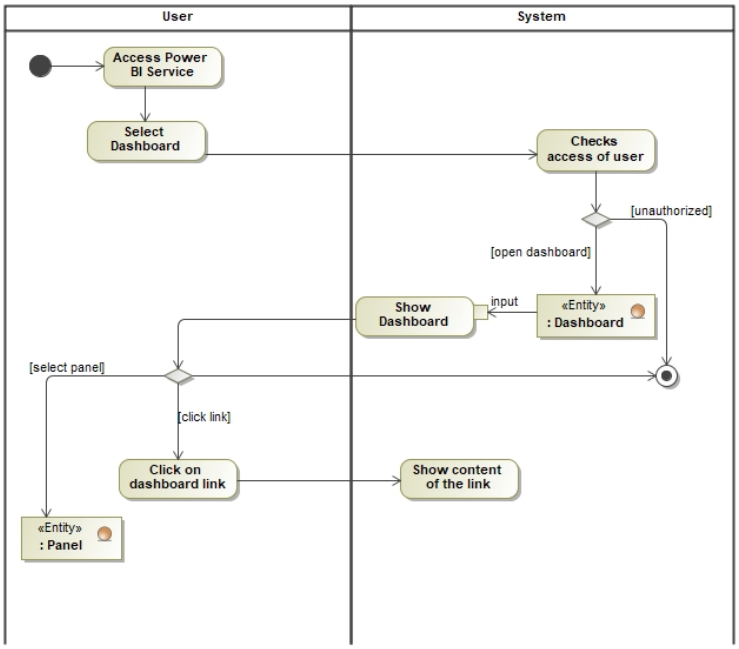


Figure 3.2. Activity Diagram for Use Case 01

Class Diagram

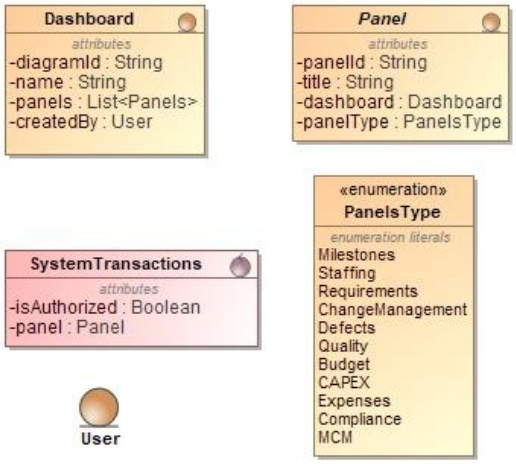


Figure 3.3. Class Diagram for Use Case 01

Sequence Diagram

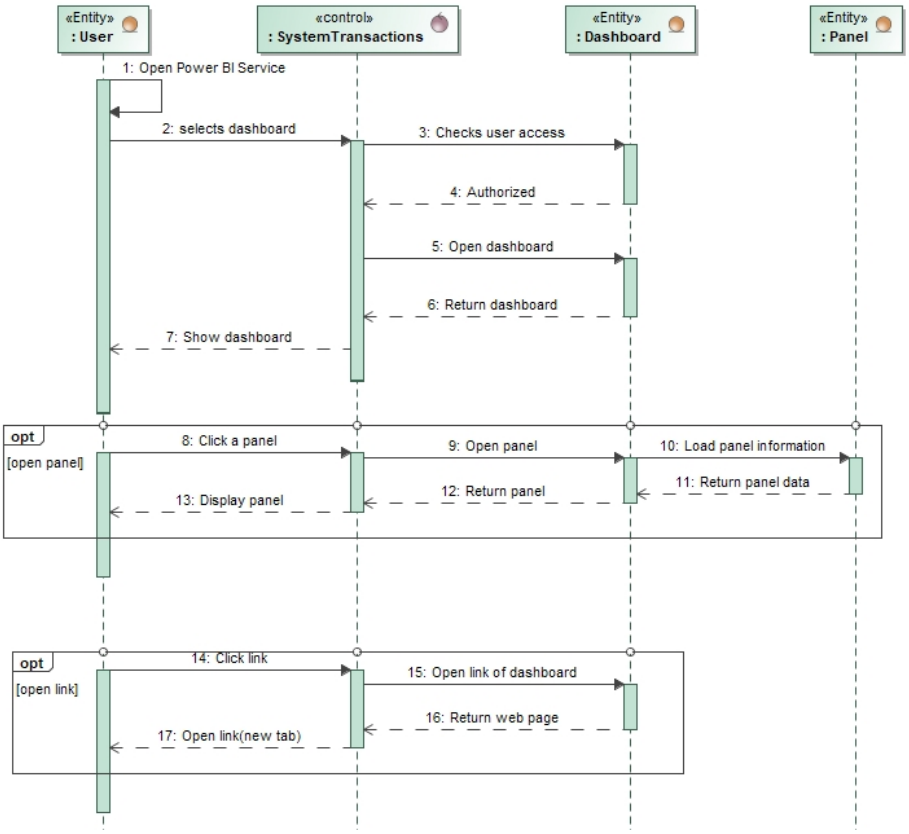


Figure 3.4. Sequence Diagram for Use Case 01

3.2.2 Use Case 02. Milestones Panel

Activity Diagram

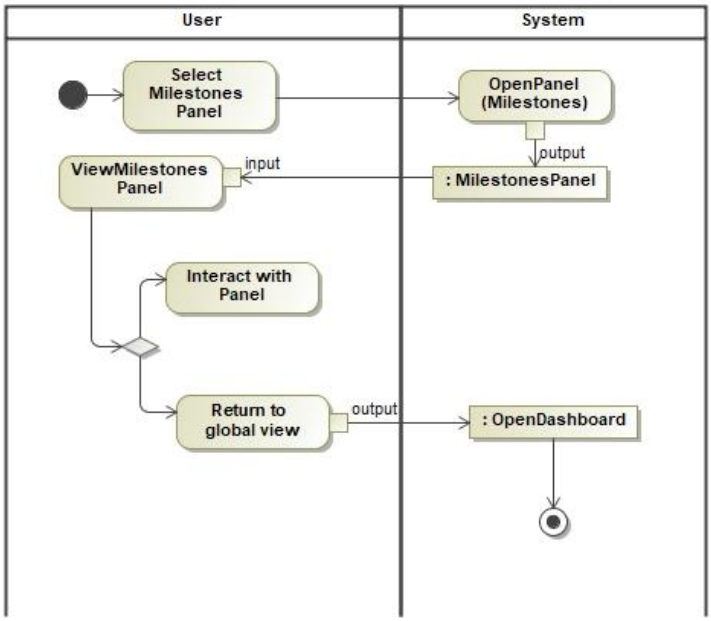


Figure 3.5. Activity Diagram for Use Case 02

Class Diagram

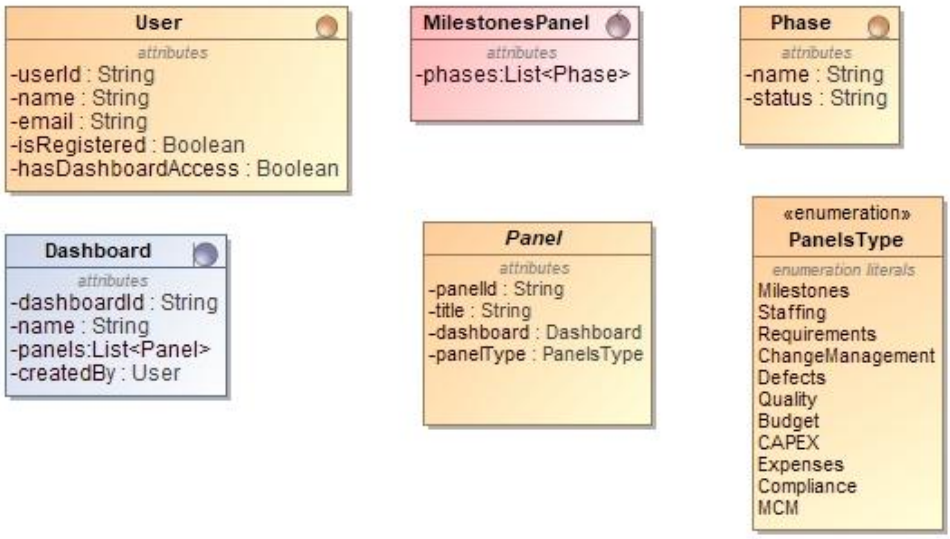


Figure 3.6. Class Diagram for Use Case 02

Sequence Diagram

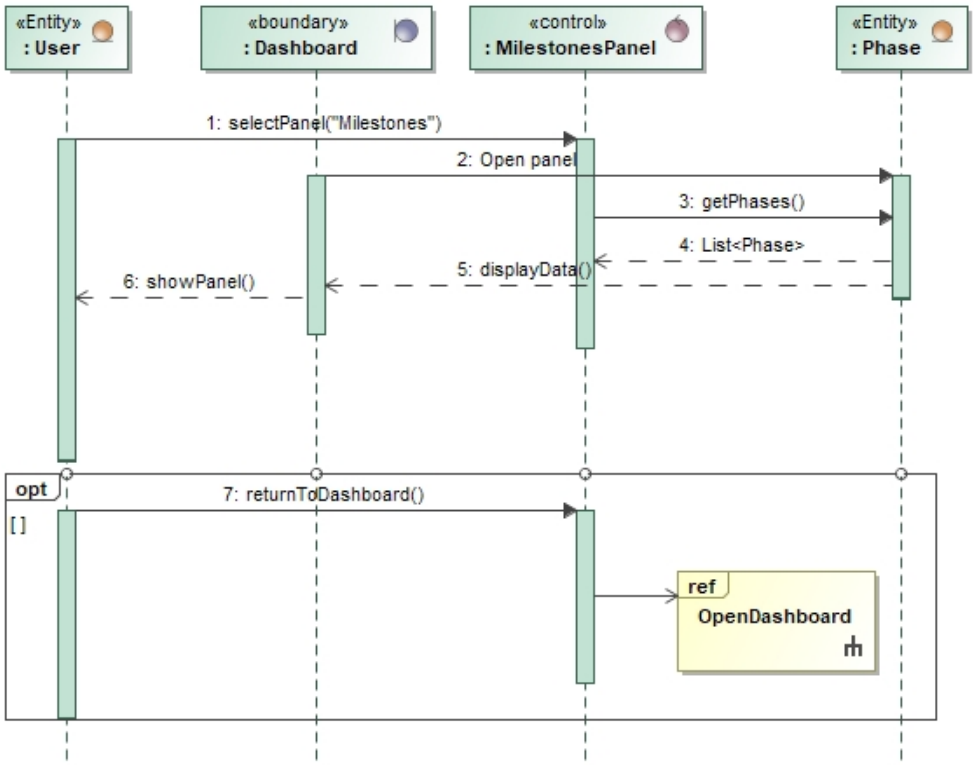


Figure 3.7. Sequence Diagram for Use Case 02

3.2.3 Use Case 03. Staffing Panel

Activity Diagram

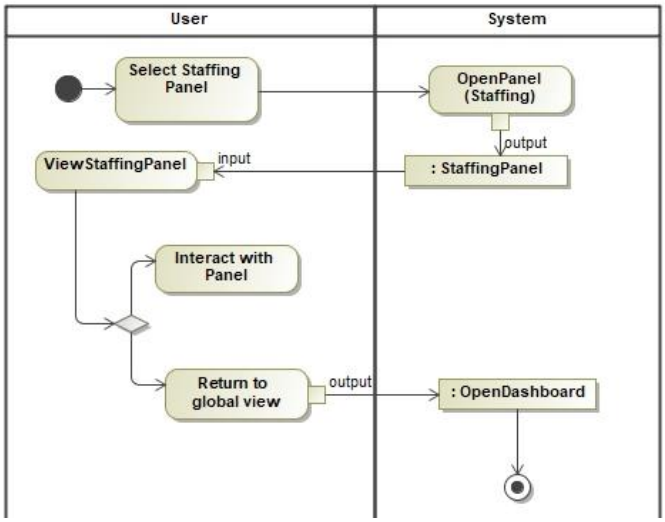


Figure 3.8. Activity Diagram for Use Case 03

Class Diagram

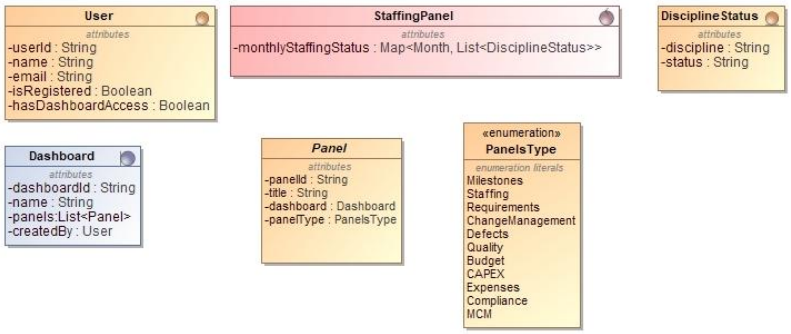


Figure 3.9. Class Diagram for Use Case 03

Sequence Diagram

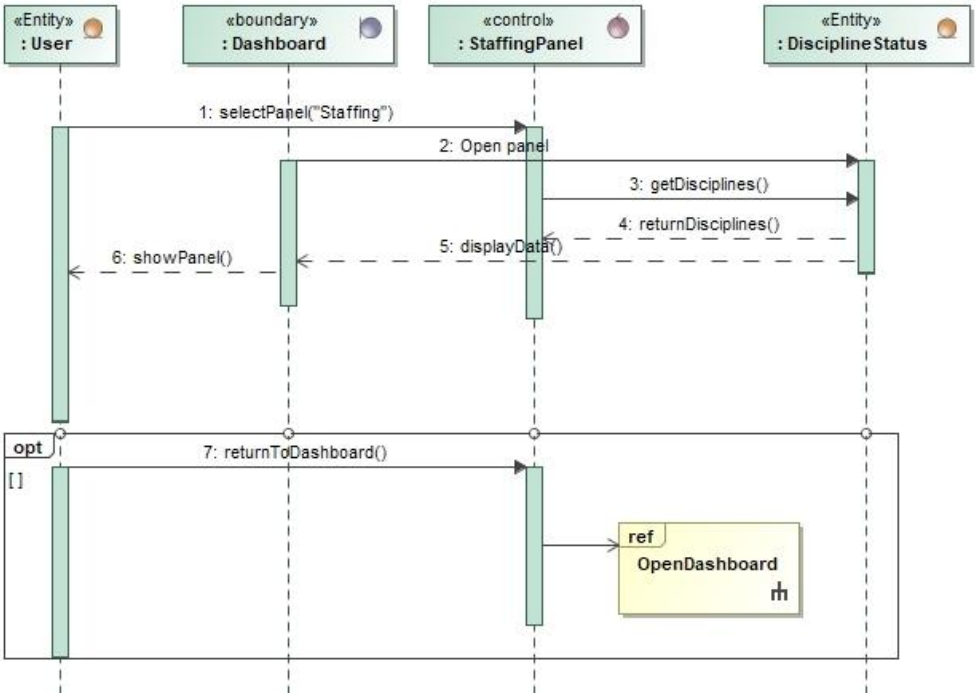


Figure 3.10. Sequence Diagram for Use Case 03

3.2.4 Use Case 04. Requirements Panel

Activity Diagram

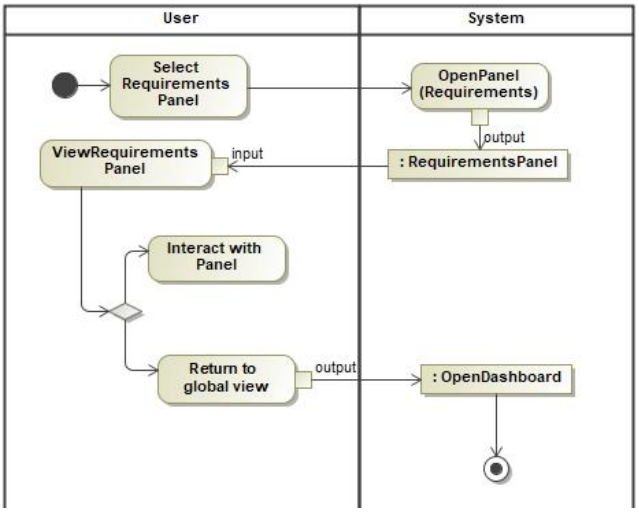


Figure 3.11. Activity Diagram for Use Case 04

Class Diagram

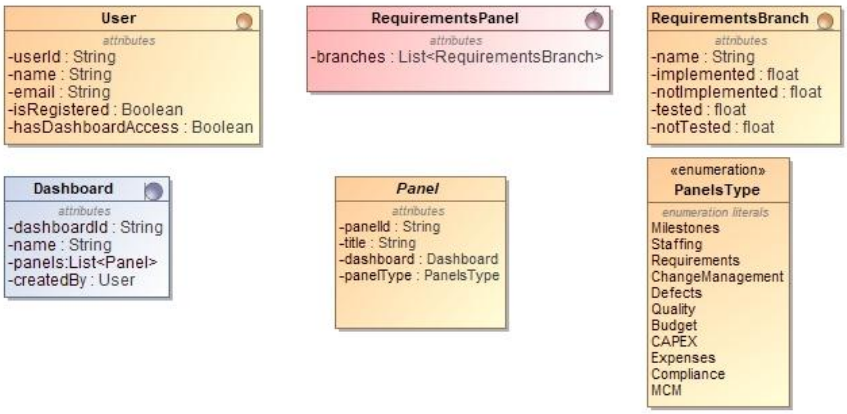


Figure 3.12. Class Diagram for Use Case 04

Sequence Diagram

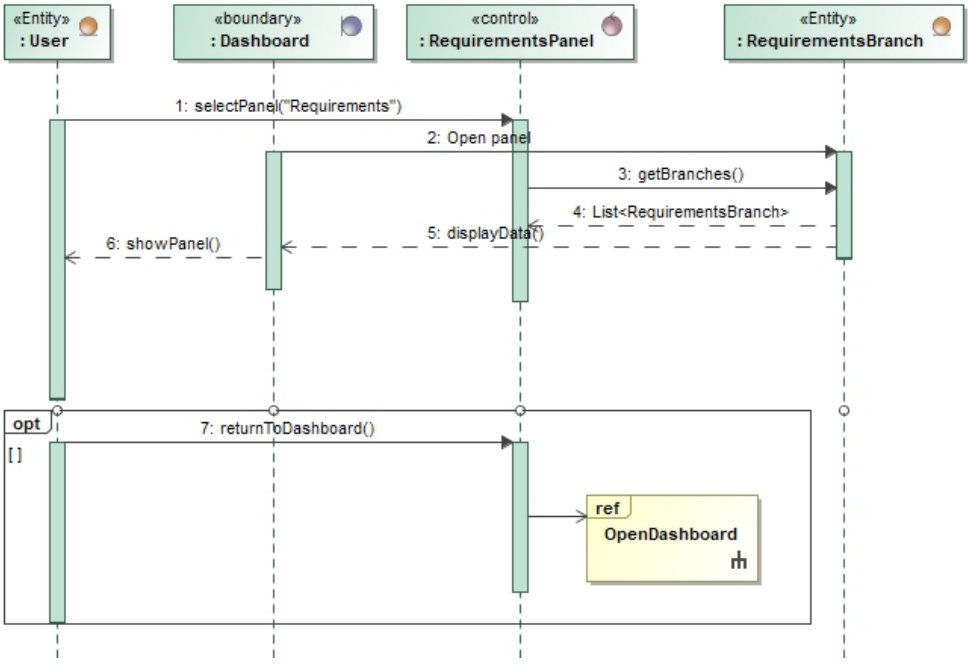


Figure 3.13. Sequence Diagram for Use Case 04

3.2.5 Use Case 05. Change Management Panel

Activity Diagram

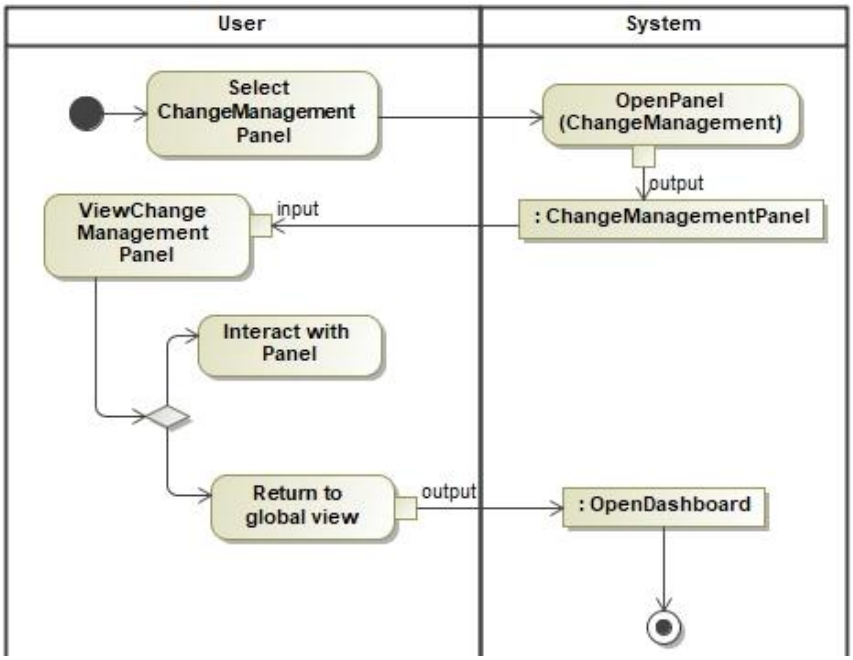


Figure 3.14. Activity Diagram for Use Case 05

Class Diagram

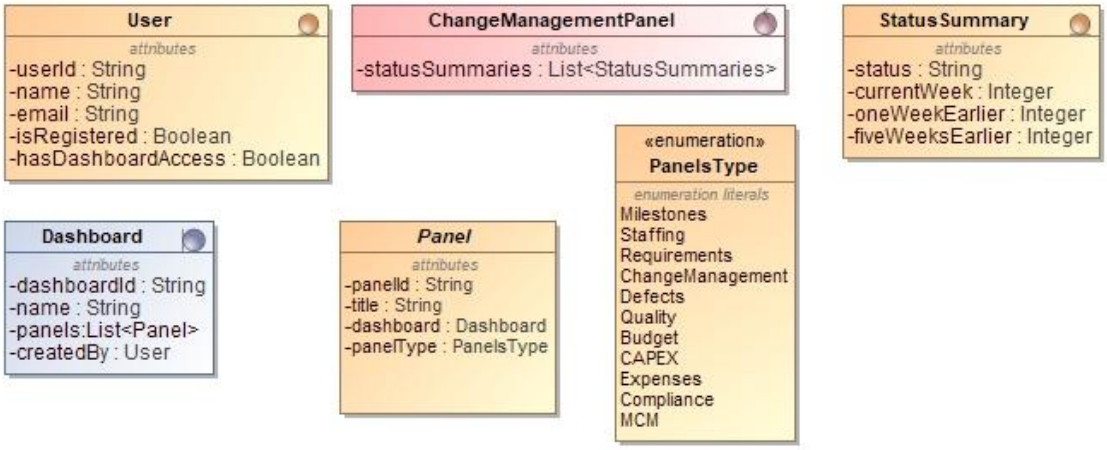


Figure 3.15. Class Diagram for Use Case 05

Sequence Diagram

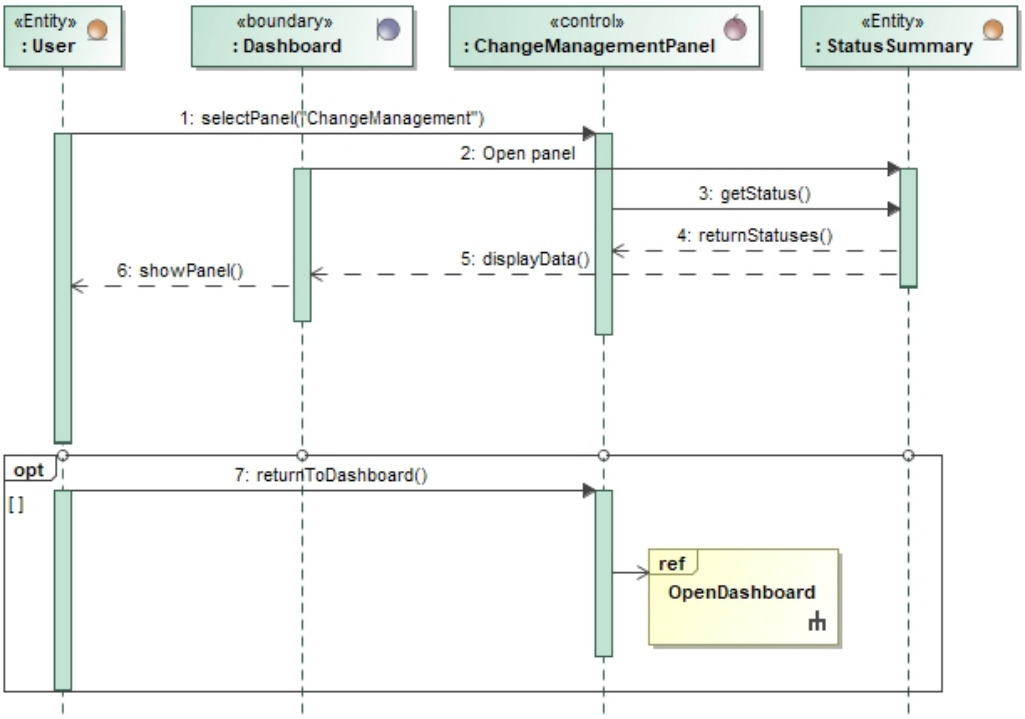


Figure 3.16. Sequence Diagram for Use Case 05

3.2.6 Use Case 06. Defects Panel

Activity Diagram

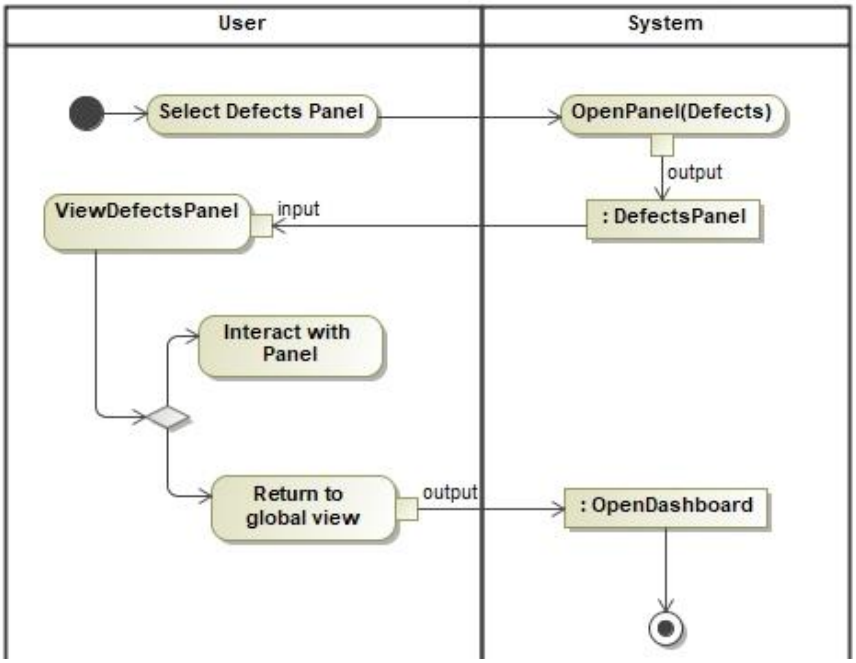


Figure 3.17. Activity Diagram for Use Case 06

Class Diagram

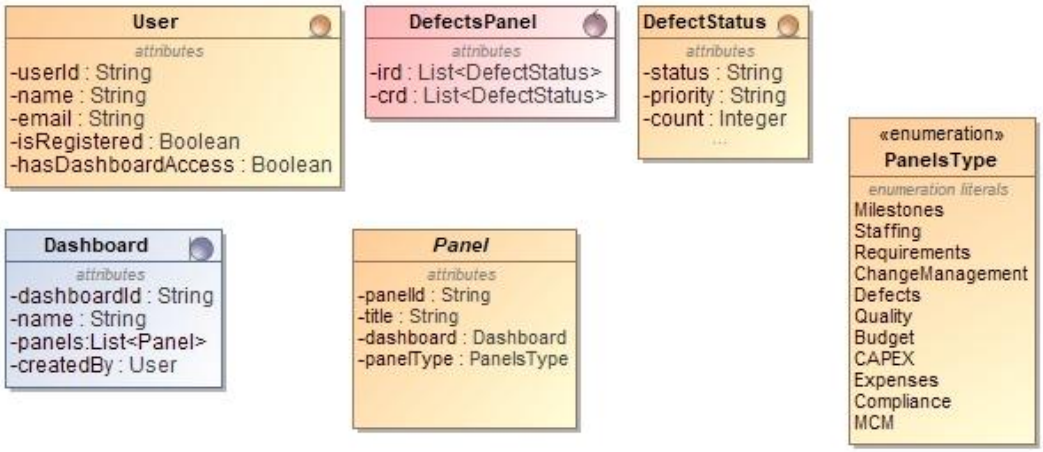


Figure 3.18. Class Diagram for Use Case 06

Sequence Diagram

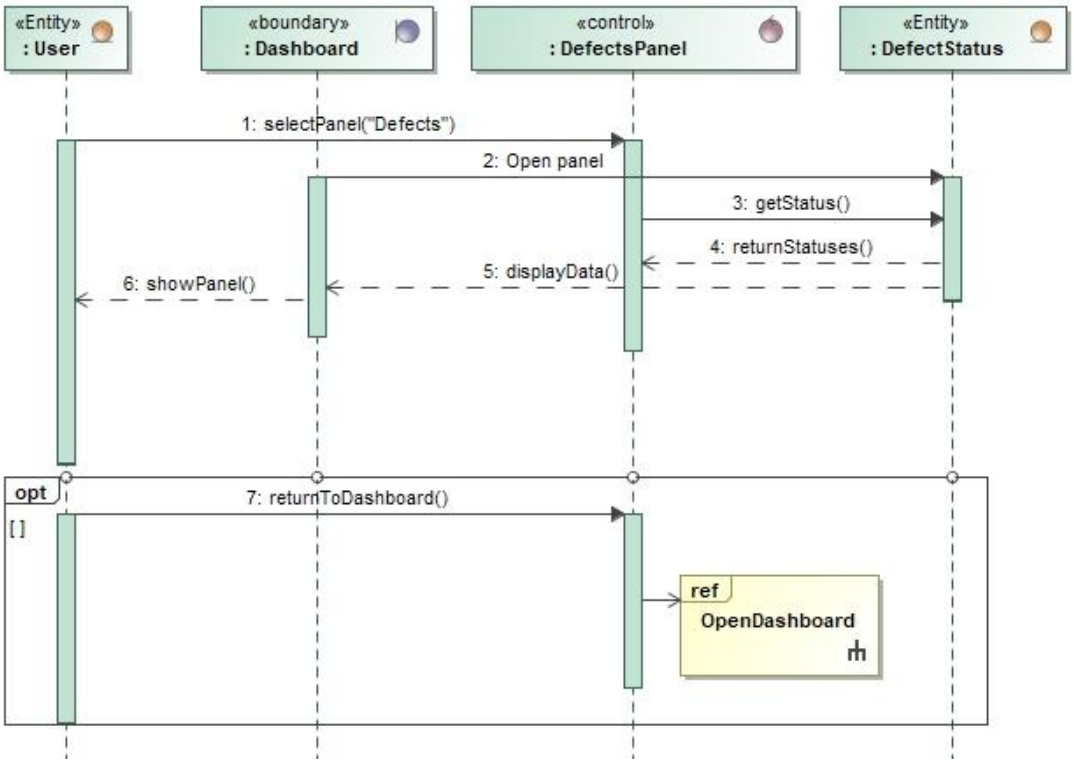


Figure 3.19. Sequence Diagram for Use Case 06

3.2.7 Use Case 07. Quality Panel

Activity Diagram

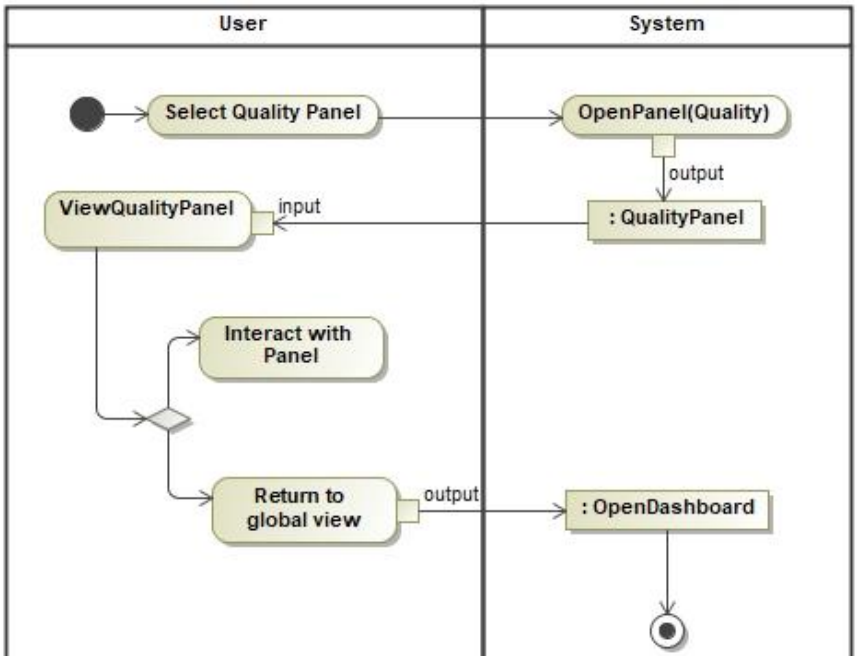


Figure 3.20. Activity Diagram for Use Case 07

Class Diagram

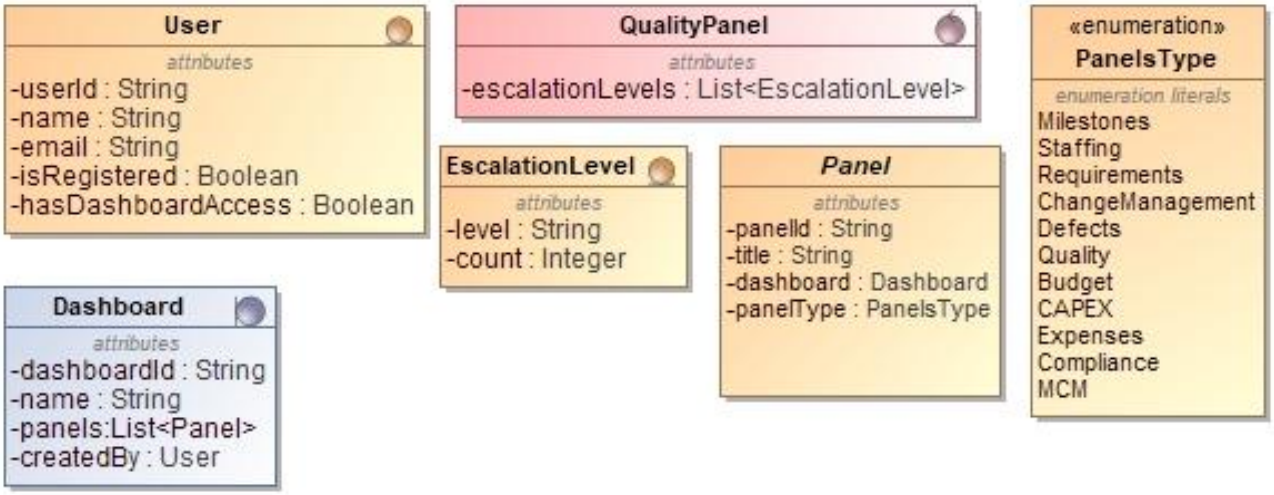


Figure 3.21. Class Diagram for Use Case 07

Sequence Diagram

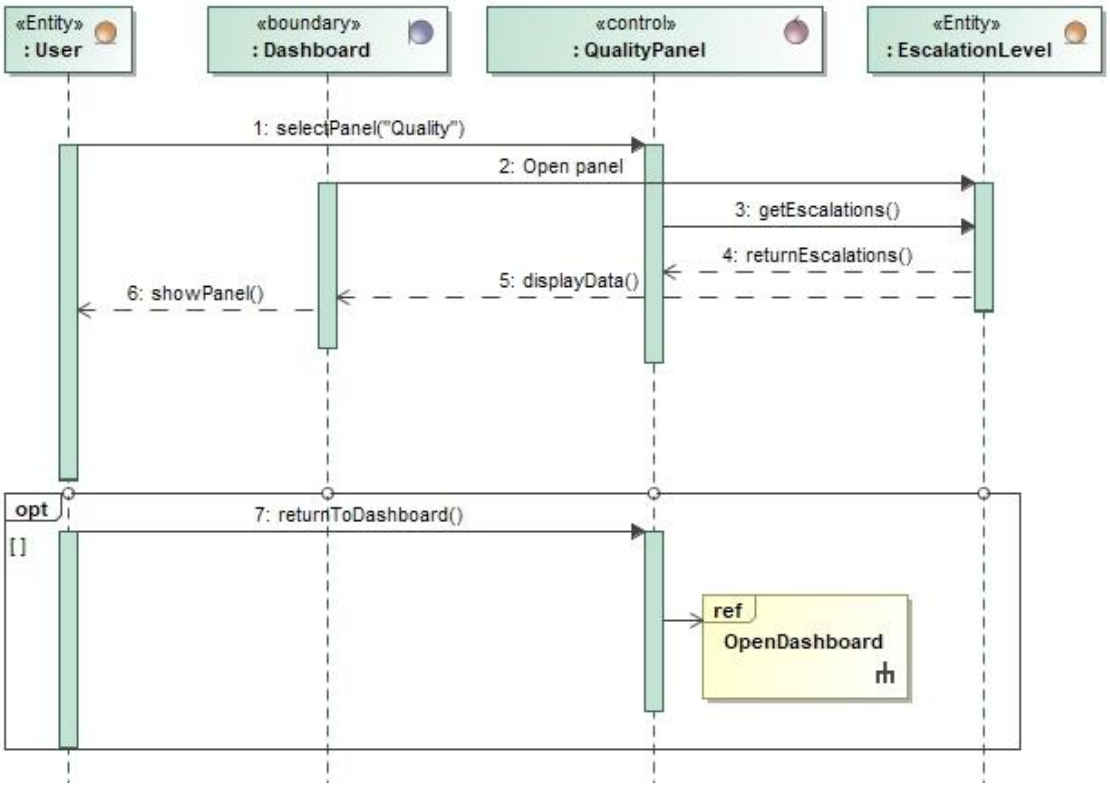


Figure 3.22. Sequence Diagram for Use Case 07

3.2.8 Use Case 08. Budget Panel

Activity Diagram

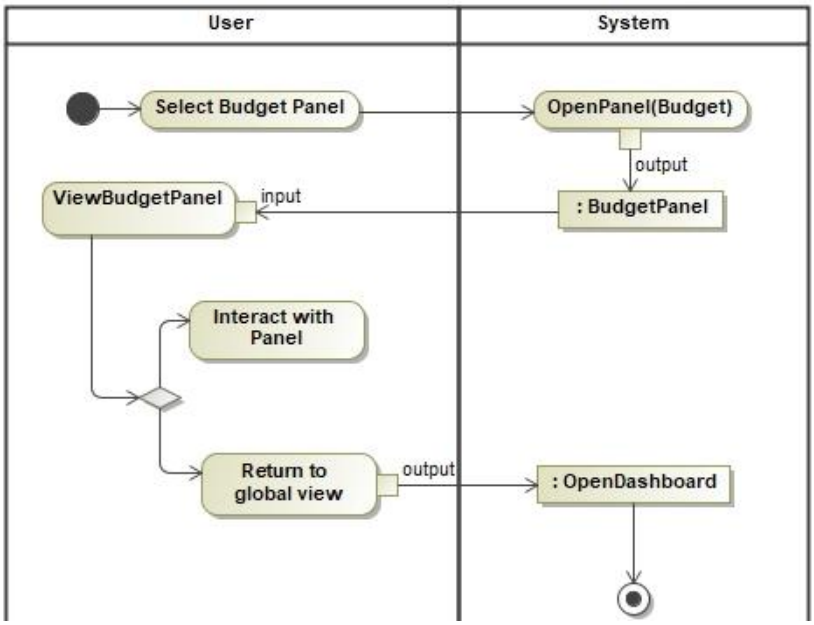


Figure 3.23. Activity Diagram for Use Case 08

Class Diagram

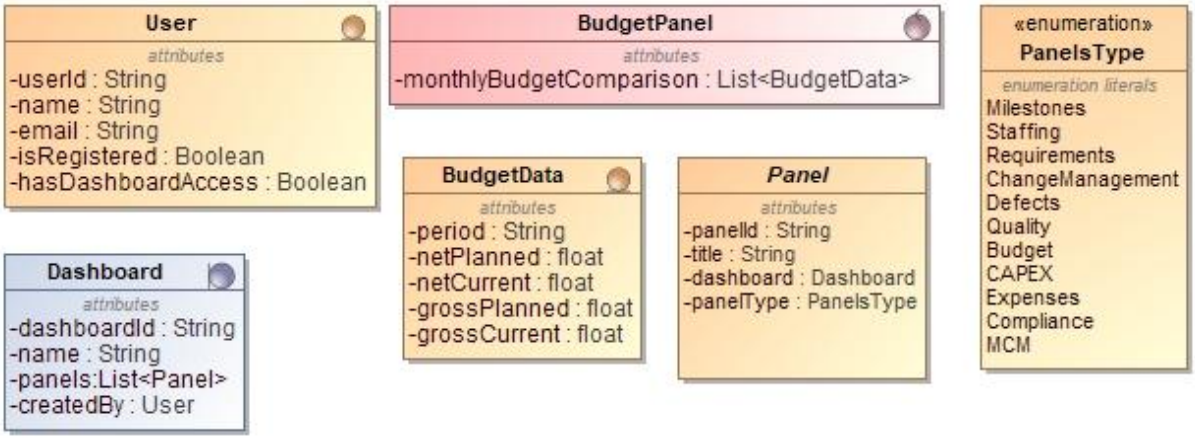


Figure 3.24. Class Diagram for Use Case 08

Sequence Diagram

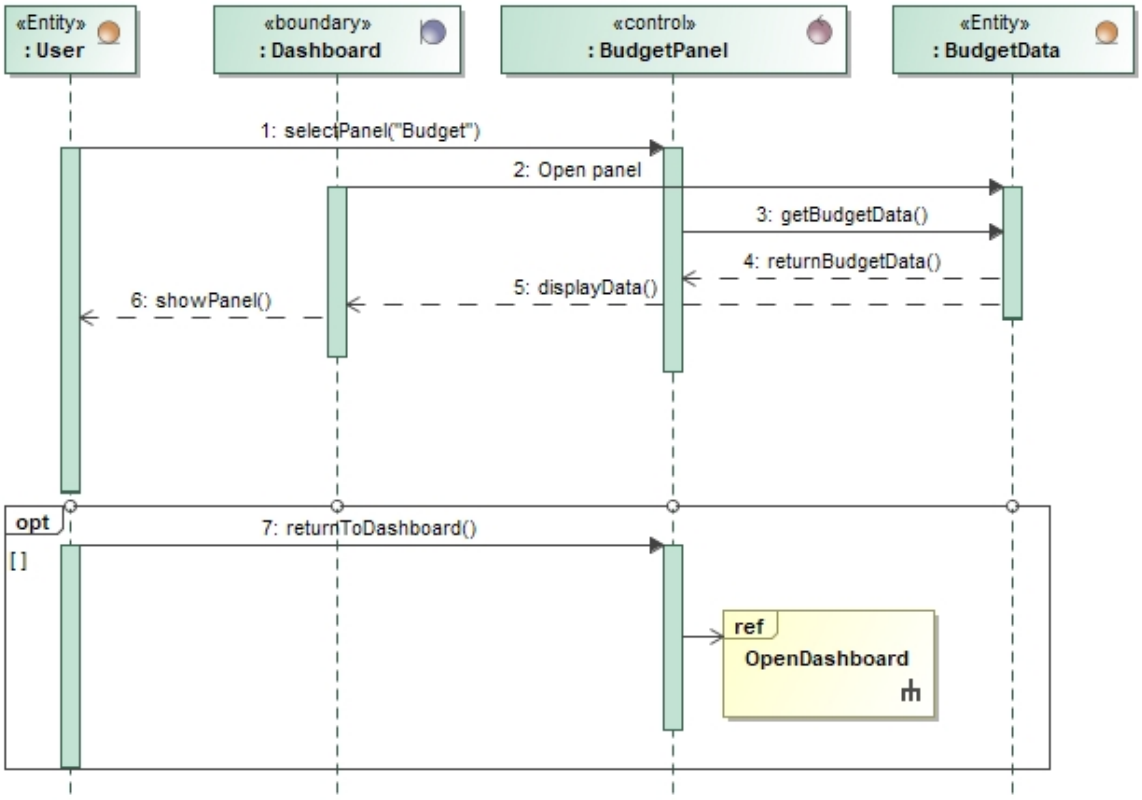


Figure 3.25. Sequence Diagram for Use Case 08

3.2.9 Use Case 09. CAPEX Panel

Activity Diagram

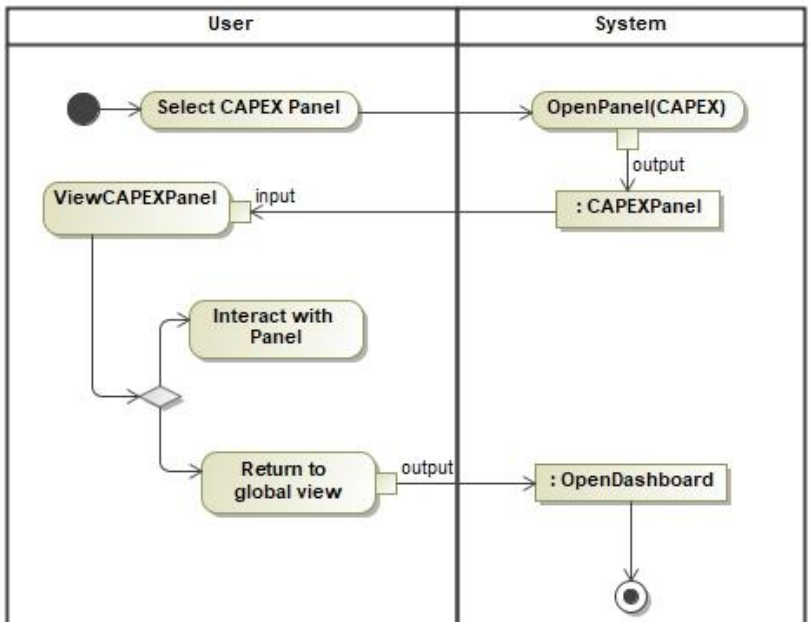


Figure 3.26. Activity Diagram for Use Case 09

Class Diagram

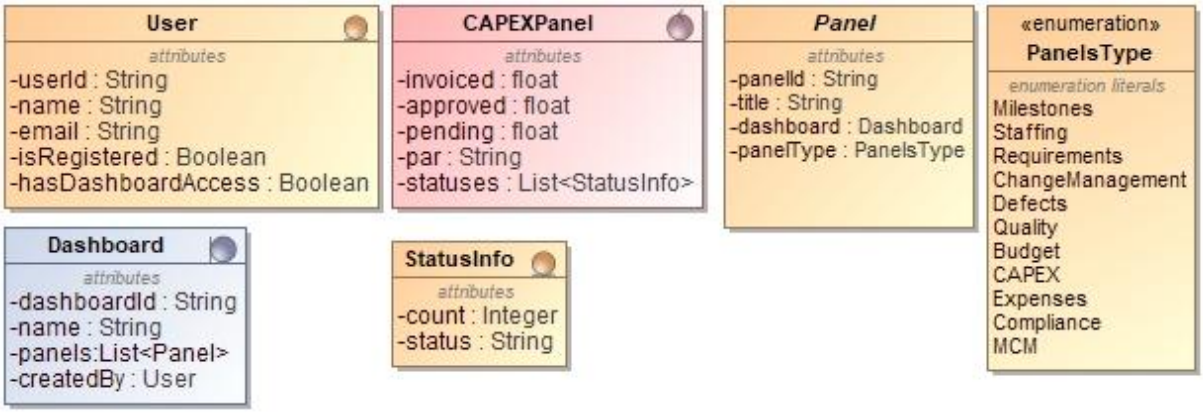


Figure 3.27. Class Diagram for Use Case 09

Sequence Diagram

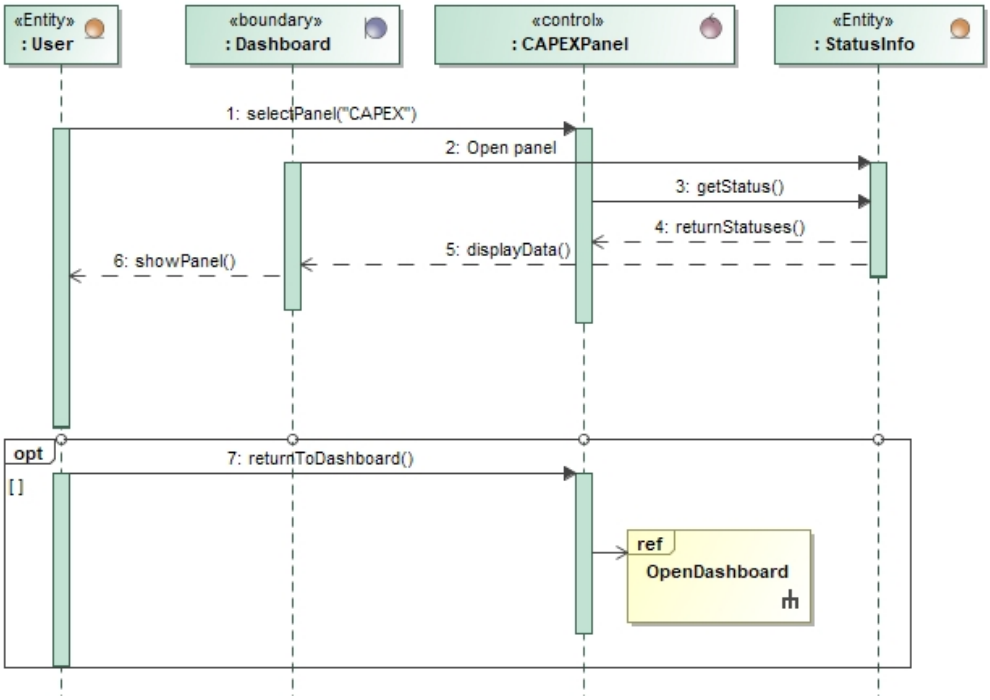


Figure 3.28. Sequence Diagram for Use Case 09

3.2.10 Use Case 10. Expenses Panel

Activity Diagram

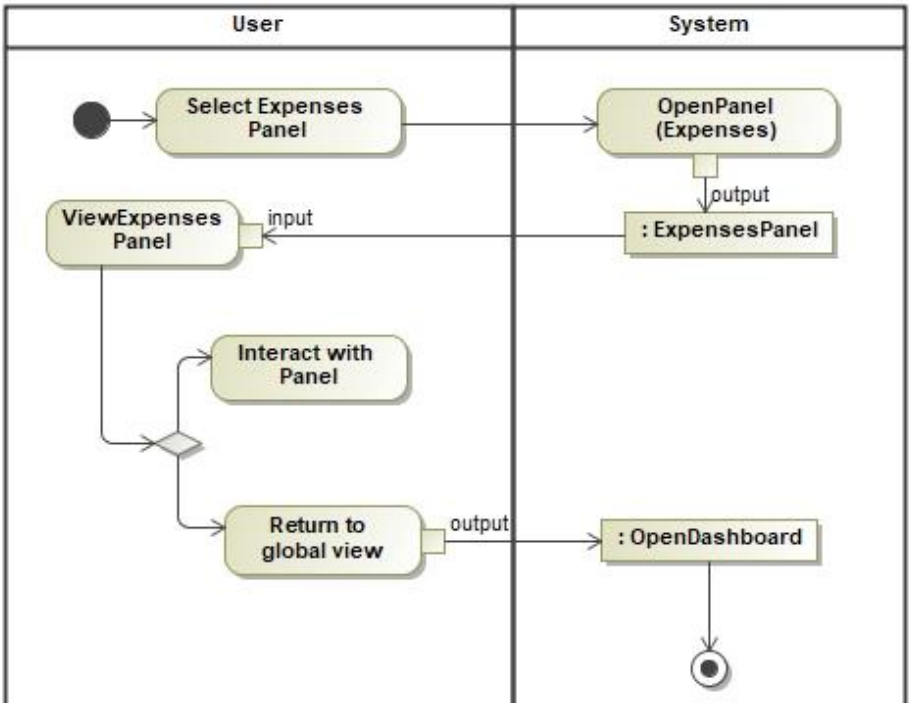


Figure 3.29. Activity Diagram for Use Case 10

Class Diagram

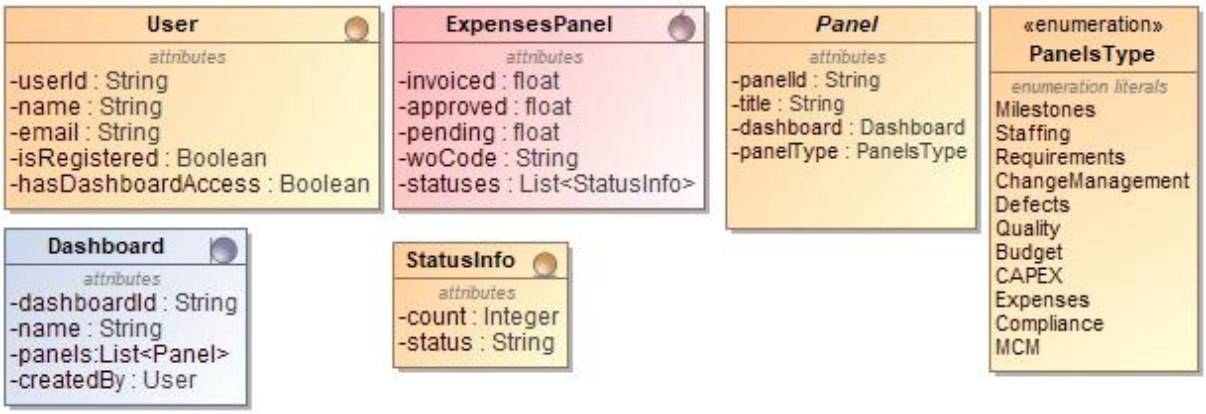


Figure 3.30. Class Diagram for Use Case 10

Sequence Diagram

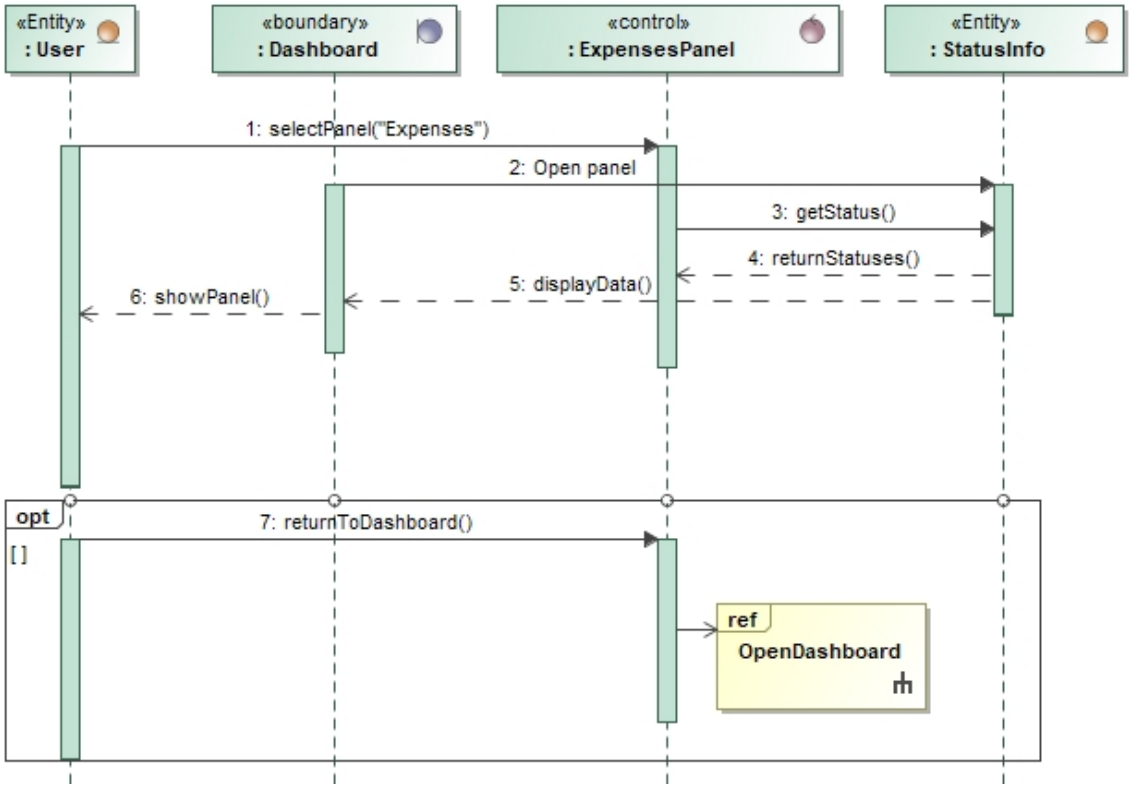


Figure 3.31. Sequence Diagram for Use Case 10

3.2.11 Use Case 11. Compliance Panel

Activity Diagram

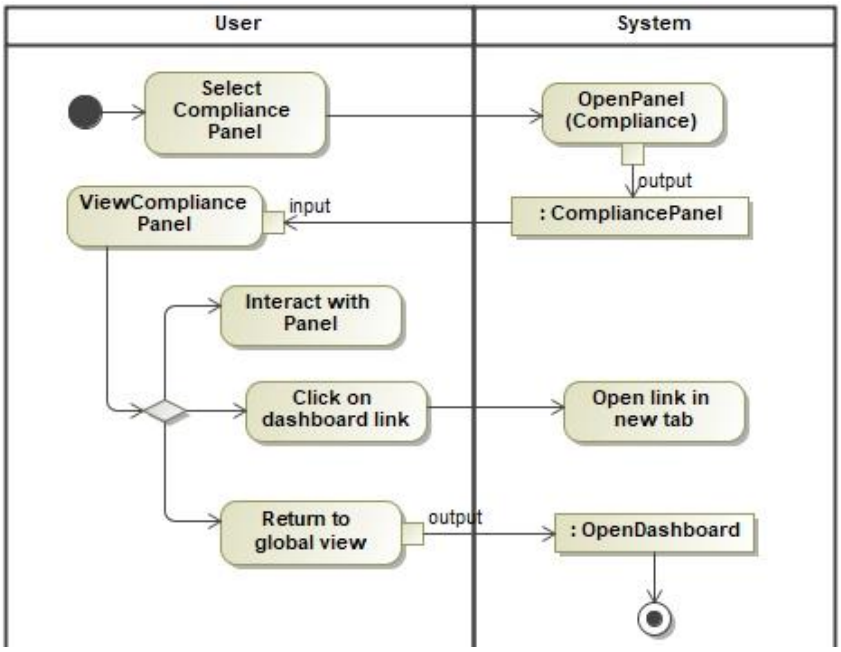


Figure 3.32. Activity Diagram for Use Case 11

Class Diagram

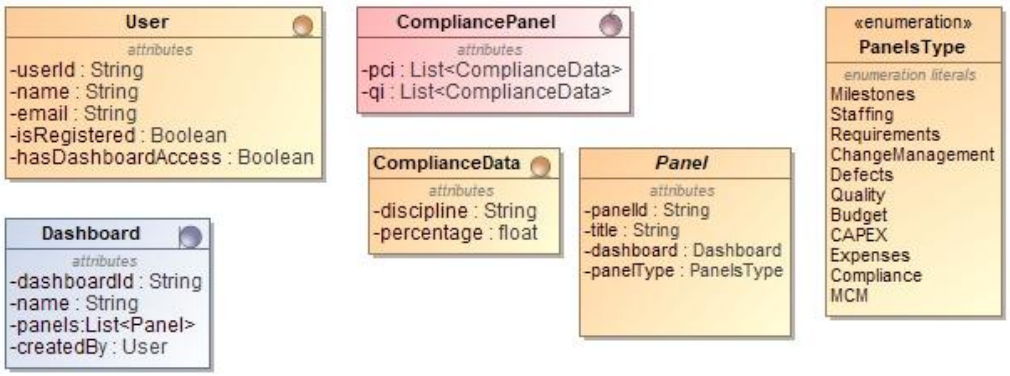


Figure 3.33. Class Diagram for Use Case 11

Sequence Diagram

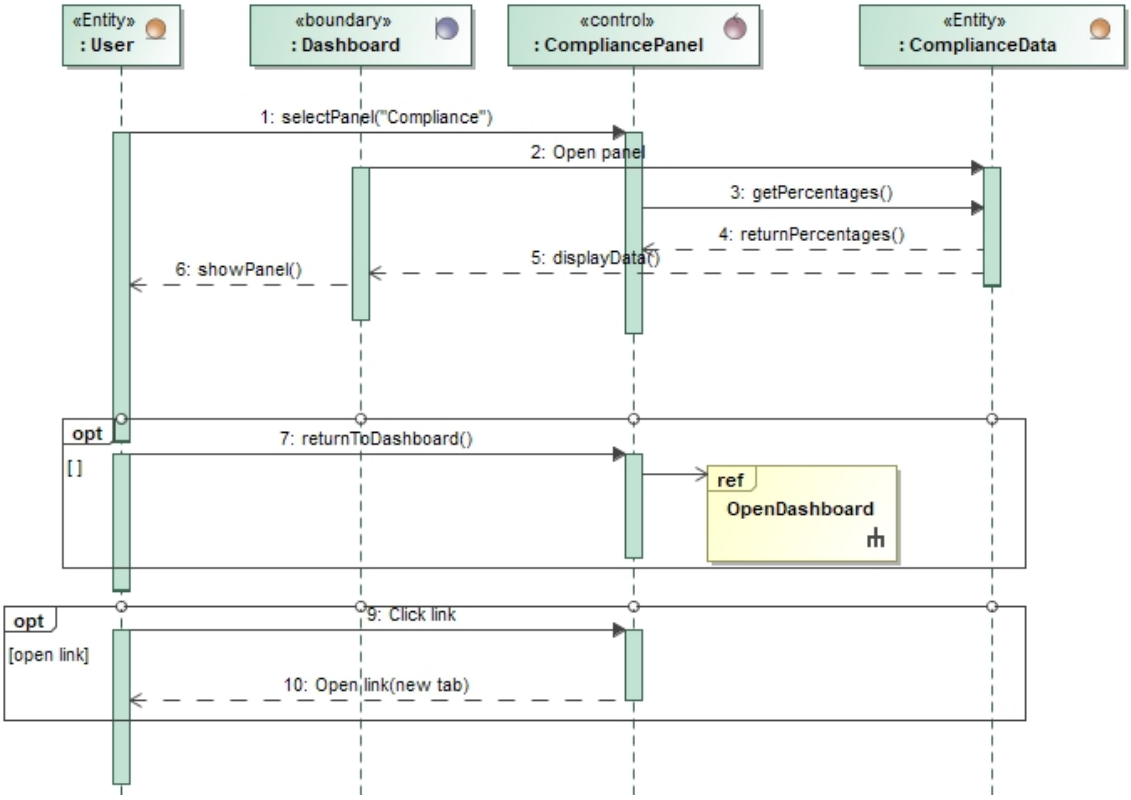


Figure 3.34. Sequence Diagram for Use Case 11

3.2.12 Use Case 12. MCM Panel

Activity Diagram

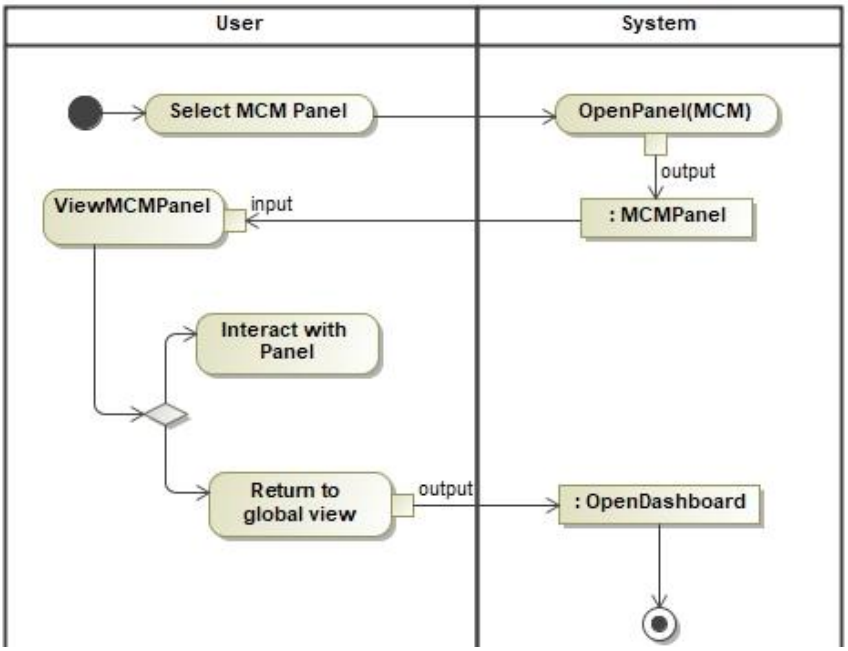


Figure 3.35. Activity Diagram for Use Case 12

Class Diagram

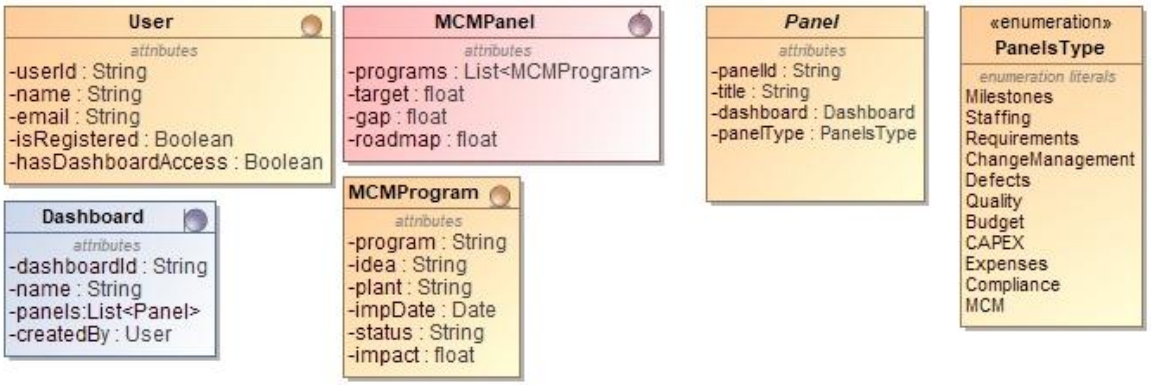


Figure 3.36. Class Diagram for Use Case 12

Sequence Diagram

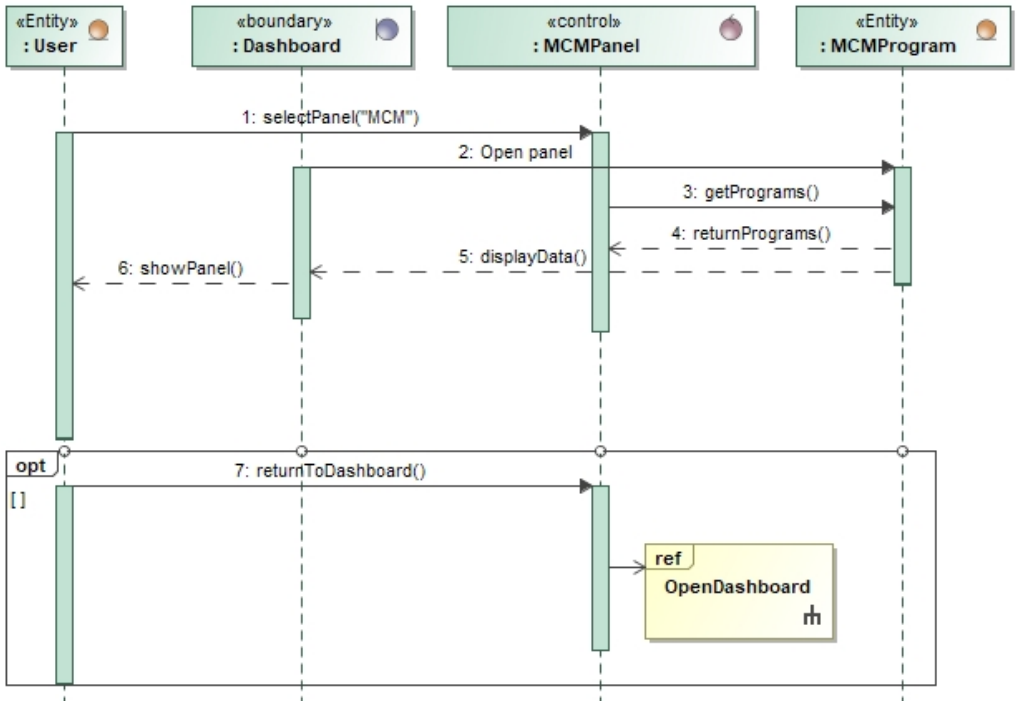


Figure 3.37. Sequence Diagram for Use Case 12

4 Design

In this chapter I'll explain the design of the architecture of the application, the user interface and how the data is connected to the dashboard. With this we want to clarify the structure of the project and how to implement the use cases described in an earlier chapter as well as the different functional requirements and non-functional requirements.

4.1 Design of architecture

In *Figure 4.1*, we can observe how the components relate to each other.

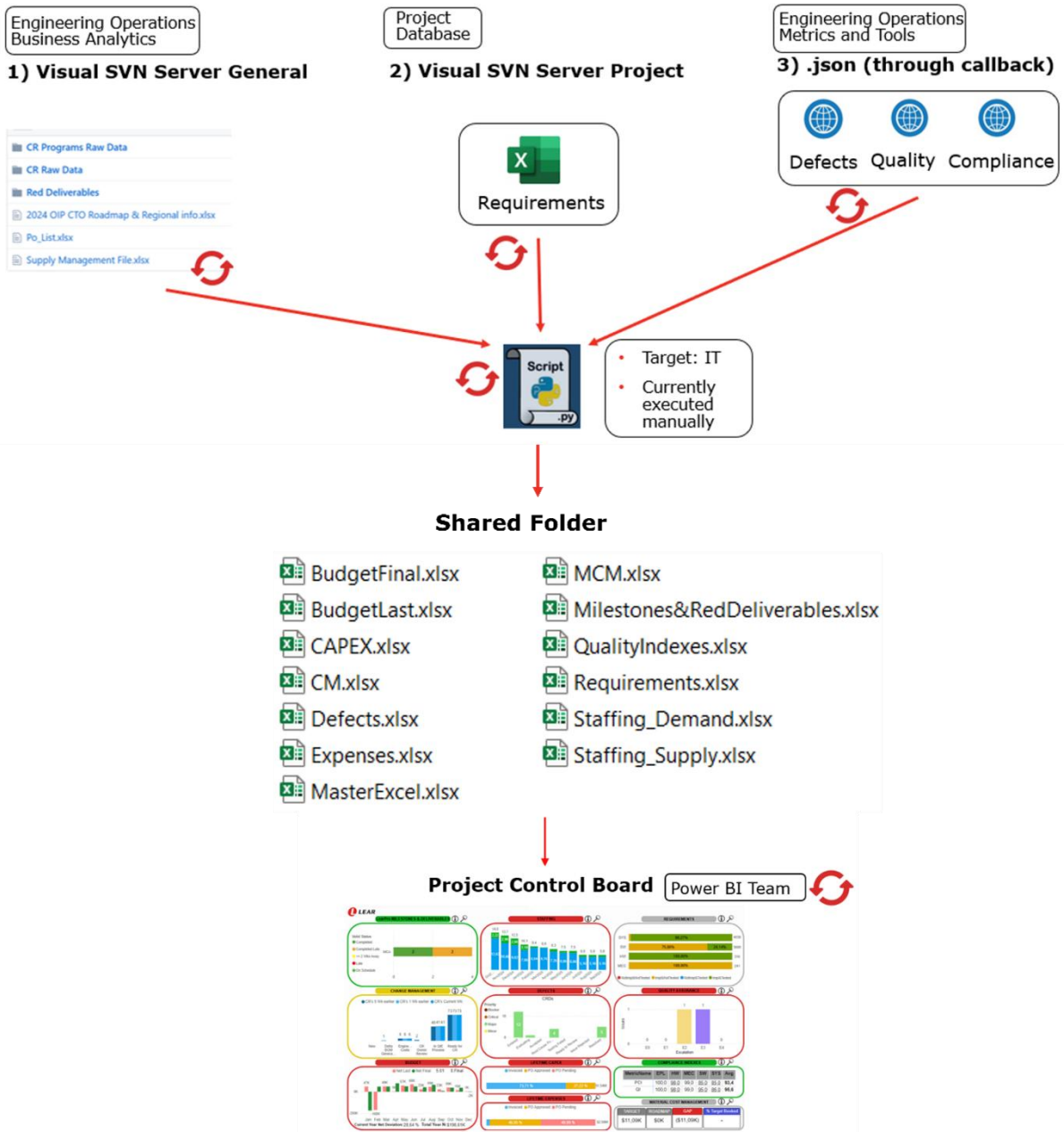


Figure 4.1. System Architecture

To populate the dashboard with the necessary information, the project relies on three primary data sources:

- **Visual SVN Server - General**: This source provides data for the following panels: Milestones & Deliverables, Change Management, Budget, CAPEX, Expenses, and Material Cost Management.
- **Visual SVN Server – Project Specific**: This source is used exclusively for the Requirements panel.
- **JSON Files (via callback)**: These files supply data for the Defects, Quality Assurance, and Quality Indexes panels.

A Python script is responsible for retrieving and filtering data from these sources. The script is executed automatically every day on a dedicated server. Once processed, the data is saved into Excel spreadsheets, with one file dedicated to each panel. These Excel files are then stored in a shared network folder.

The folder structure is organized by project: each project has its own directory within the shared folder, containing the Excel files corresponding to its specific dashboard panels. Power BI is configured to connect directly to these Excel files, ensuring that each dashboard is linked to its respective project data. Additionally, the script relies on a MasterExcel Excel file, which contains the various parameters needed for its execution.

Finally, the dashboards are set to refresh automatically on a daily basis, ensuring that users always have access to the most recent and accurate project information.

4.2 Design of the interface

The design of the dashboard enables interaction with data from various departments within Lear, making it easier for project team members to gain a comprehensive overview of the project.

4.2.1 Global View

The image below shows the current state of the dashboard. Over time, the layout and distribution of the panels have changed, resulting in the current order, names, and color rules applied. At the top center, the title of the project is displayed. The images in this file use mock data, but in the actual dashboard, data is pulled from real sources such as Excel files or JSONs.

There are four different panel colors, each indicating a specific status:

- Green: The panel is performing well.
- Yellow: There is a potential issue, but it is not yet urgent.
- Red: There is a critical issue that needs immediate attention.
- Grey: There is no metric available yet to determine the panel’s status.

The “i” icon displays the upload date of the dashboard when the user hovers the mouse over it. The magnifying glass, when clicked, opens an expanded view with more detailed information about the panel. Each panel has its own expanded view.

To visualize the dashboard, the user must have the appropriate access permissions. If access is not granted, the user should contact the individual responsible for managing dashboard permissions to request access. This request will only be fulfilled if the user is a member of the project team.

Figure 4.2. Global view of the dashboard

4.2.2 LearPro Milestones and Deliverables

This panel outlines the various milestones of the project along with their status, indicating whether a project phase is complete or still in progress.

The metrics used to determine the color rules are as follows:

- RED: #Late <> 0.
- YELLOW: #<= 2 Wks Away <> 0.
- GREEN: Other cases.

The data originates from the platform **LearPro**, and it is extracted from the Excel file named ESD_deliverables.csv.

This is how the panel appears in the global overview of the project:

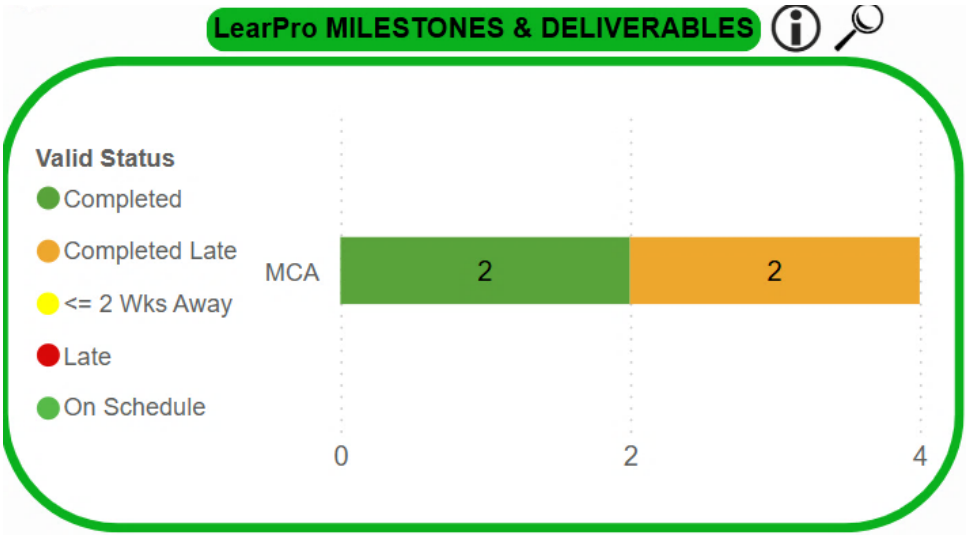


Figure 4.3. Milestone's Global View

This is the display for the expanded panel:

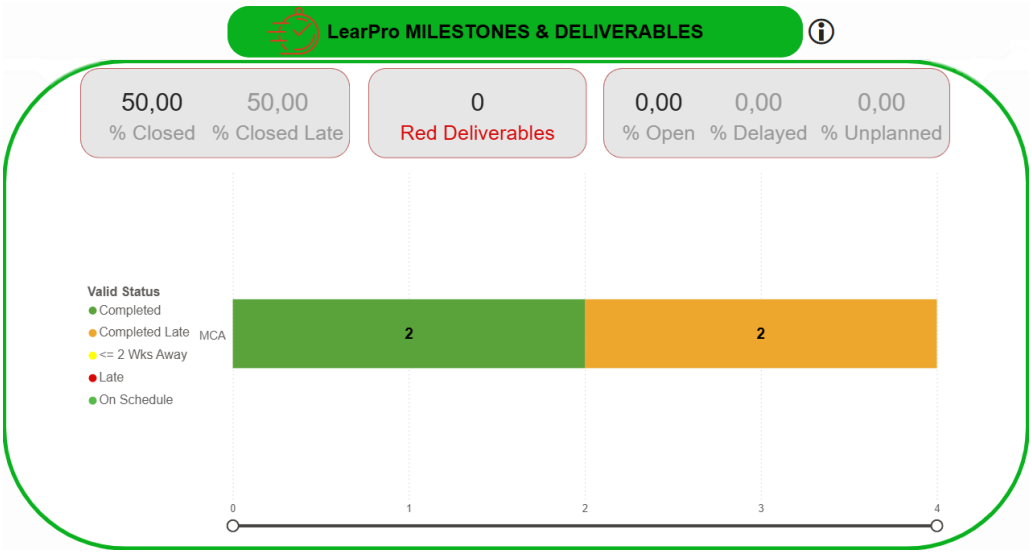


Figure 4.4. Milestone's Expanded View

In this new view, additional percentages are displayed:

- % Closed: counts deliverables with the status *Completed*.
- % Closed Late: counts deliverables with the status *Completed Late*.
- % Open: counts deliverables with the statuses *<=2 Wks Away*, *Late* and *On Schedule*.
- % Delayed: counts deliverables with the status *Late*.
- % Unplanned: counts deliverables with any status not listed in the legend.

Another new feature is that the panel counts milestones that are *Late* and categorizes them as *Red Deliverables*.

4.2.3 Staffing

This panel displays the fluctuations in project staffing, broken down by month. Each bar color represents the difference between supply and demand:

- Blue: current staff assigned to the project.
- Green: excess staff supply that the project currently does not need.
- Red: a shortage of staff, meaning the project currently has fewer staff than required.

The metrics used to determine the color rules are as follows:

- RED: sum(demand-supply)/12 greater or equal than 0.1.
- YELLOW: sum(demand-supply)/12 between 0.05 and 0.1.
- GREEN: sum(demand-supply)/12 less or equal 0.05.

The data originates from **PCM** and **Supply Data**, and it is extracted from the following Excel files:

- Supply Management File.xlsx
- Supply Management File (PIT, Elec, Electrif, EO).xlsx
- WOLifetime.xlsx
- WOForecast.xlsx

This is how the panel appears in the global overview of the project:

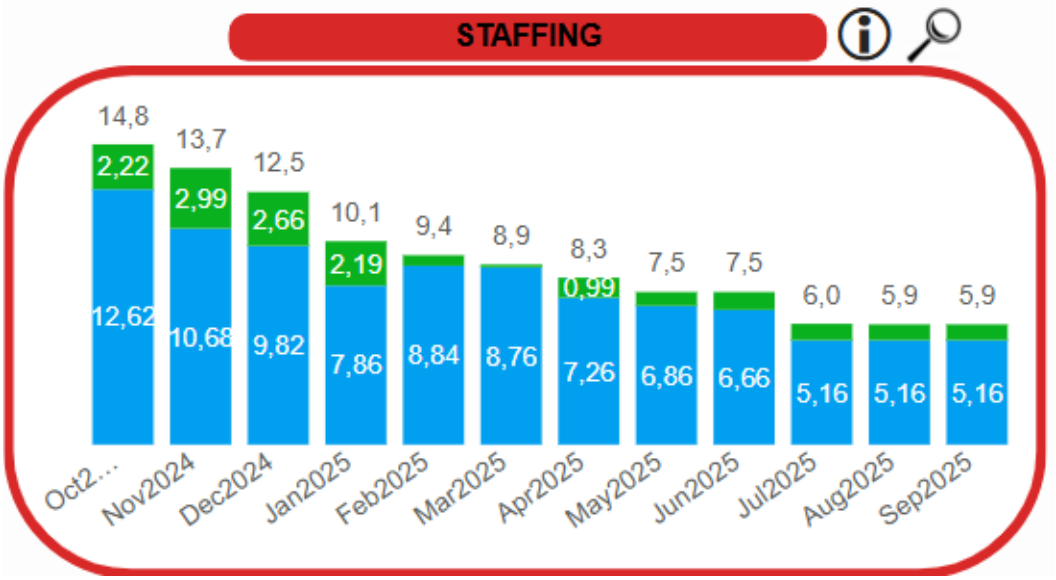
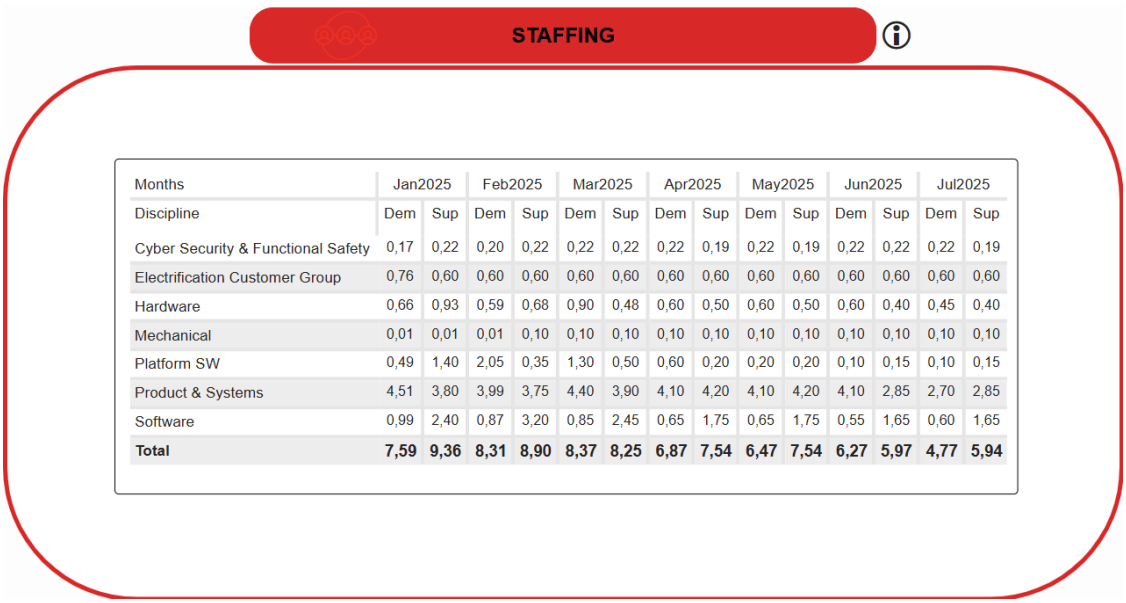


Figure 4.5. Staffing's Global View

This is the display for the expanded panel:



Months	Jan2025		Feb2025		Mar2025		Apr2025		May2025		Jun2025		Jul2025	
Discipline	Dem	Sup	Dem	Sup	Dem	Sup	Dem	Sup	Dem	Sup	Dem	Sup	Dem	Sup
Cyber Security & Functional Safety	0,17	0,22	0,20	0,22	0,22	0,22	0,22	0,19	0,22	0,19	0,22	0,22	0,22	0,19
Electrification Customer Group	0,76	0,60	0,60	0,60	0,60	0,60	0,60	0,60	0,60	0,60	0,60	0,60	0,60	0,60
Hardware	0,66	0,93	0,59	0,68	0,90	0,48	0,60	0,50	0,60	0,50	0,60	0,40	0,45	0,40
Mechanical	0,01	0,01	0,01	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,10
Platform SW	0,49	1,40	2,05	0,35	1,30	0,50	0,60	0,20	0,20	0,20	0,10	0,15	0,10	0,15
Product & Systems	4,51	3,80	3,99	3,75	4,40	3,90	4,10	4,20	4,10	4,20	4,10	2,85	2,70	2,85
Software	0,99	2,40	0,87	3,20	0,85	2,45	0,65	1,75	0,65	1,75	0,55	1,65	0,60	1,65
Total	**7,59**	**9,36**	**8,31**	**8,90**	**8,37**	**8,25**	**6,87**	**7,54**	**6,47**	**7,54**	**6,27**	**5,97**	**4,77**	**5,94**

Figure 4.6. Staffing's Expanded View

In this view, the table displays the different disciplines into which the staff is divided across various months. It also shows the demand and supply numbers for each month and discipline.

4.2.4 Requirements

This panel displays the various project requirements. Each bar indicates the different disciplines: system, software, hardware and mechanics. For each discipline, percentages indicate the status of requirements - how many have been tested or not, and how many have been implemented or not.

There are no metrics available yet to determine the color rules applied to this panel.

The data originates from **Project Database** (each project has its own database), and it is extracted from the following Excel files:

- SWRT-SW-FULL.xlsx
- SWRT-SYS-FULL.xlsx
- HWRT-HW-FULL.xlsx
- ELMRT-ELM-FULL.xlsx

This is the panel's appearance in the global overview and expanded view of the project:

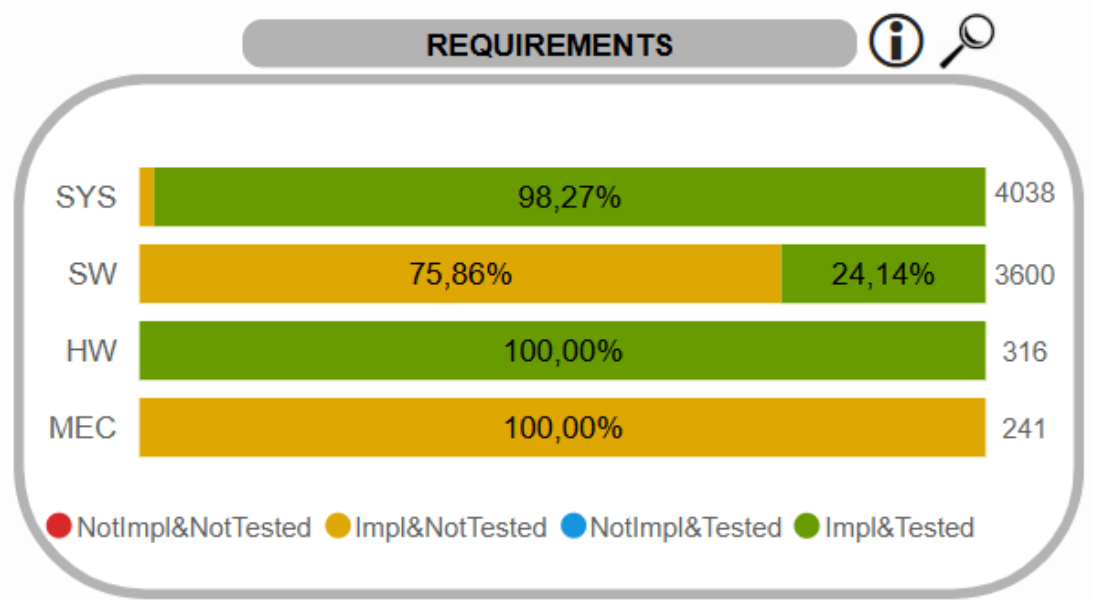


Figure 4.7. Requirements Both View s

4.2.5 Change Management

This panel displays the various project change management statuses. For each status, it shows values from three different time points: the current week, one week earlier, and five weeks earlier.

The metrics used to determine the color rules are as follows:

- GREEN: Engineering Costs & CR Owner Review & Delta BOM generation<= 5 AND ((Status 5 weeks ago - current status) for SUM(Engineering Costs & CR Owner Review & Delta BOM generation) >= 0).
- YELLOW: Engineering Costs & CR Owner Review & Delta BOM generation<= 5 AND ((Status 5 weeks ago - current status) for SUM(Engineering Costs & CR Owner Review & Delta BOM generation) >= -5).
- RED: Other cases.

The data originates from the **SEM** platform and is extracted from three Excel files named using the format %Y-%m-%d.xlsx, where the placeholders are replaced with the corresponding dates. The files represent data from the current week, the previous week, and five weeks prior.

This is how the panel appears in the global overview of the project:

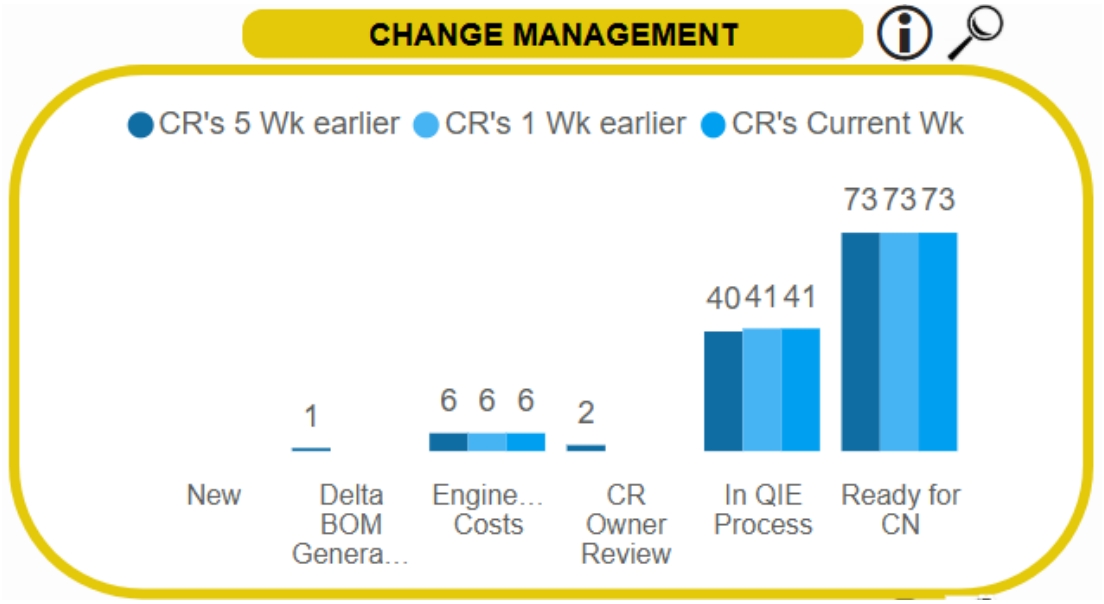


Figure 4.8. Change Management's Global View

This is the display for the expanded panel:

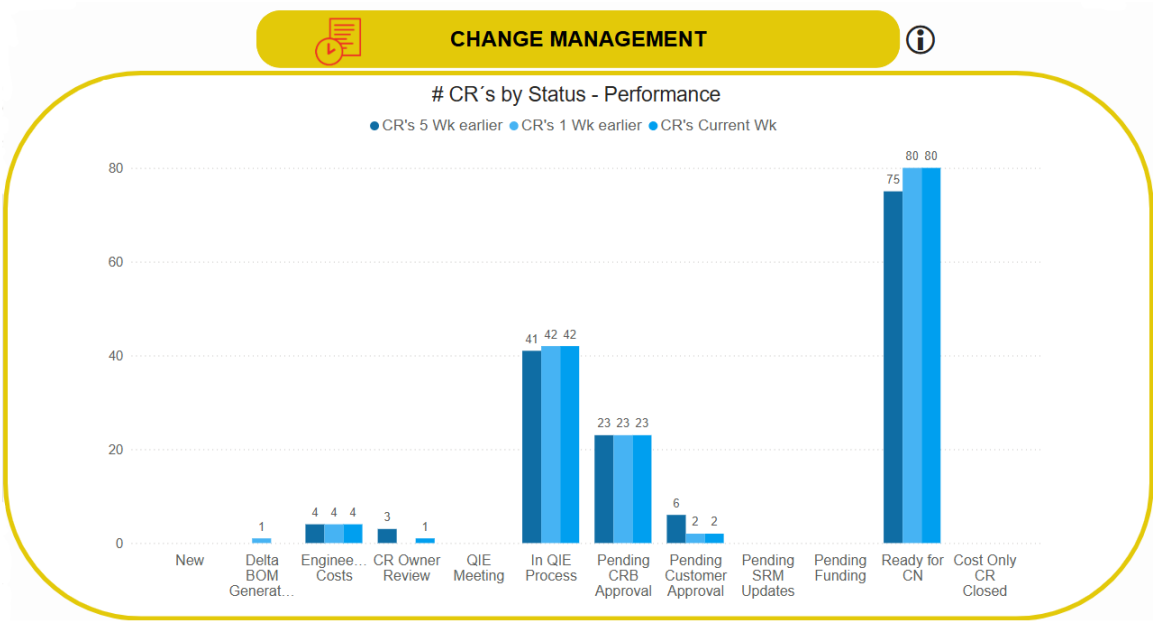


Figure 4.9. Change Management's Expanded View

This view includes more statuses than the global view, making it easier to review all project statuses from the change management.

4.2.6 Defects

This panel presents the defects of the CRDs within the project, detailing their respective statuses and the associated priority levels for each status.

The metrics used to determine the color rules are as follows:

- GREEN: #defects <= 10.
- YELLOW: #defects > 10 AND #defects <= 15.
- RED: #defects > 15 OR #defectsBlocker <> 0 OR #defectsCritical <> 0.

The data originates from the **Engineering Operations Metrics and Tools** team and is extracted from various webpage JSON files generated from JIRA database. Each project has its own data structure and naming conventions, so these will vary from one project to another.

This is how the panel appears in the global overview of the project:

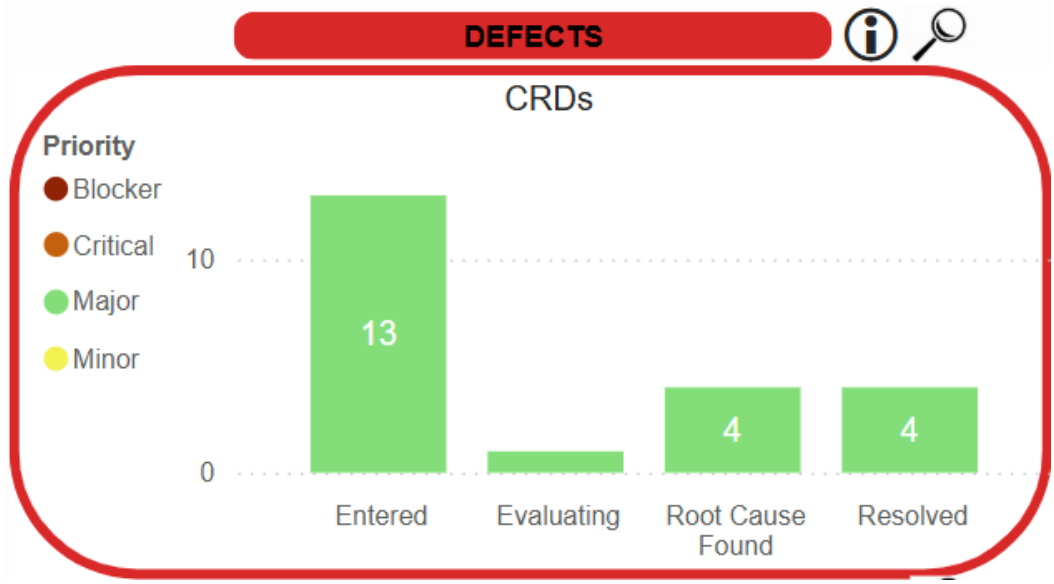


Figure 4.10. Defects Global View

This is the display for the expanded panel:

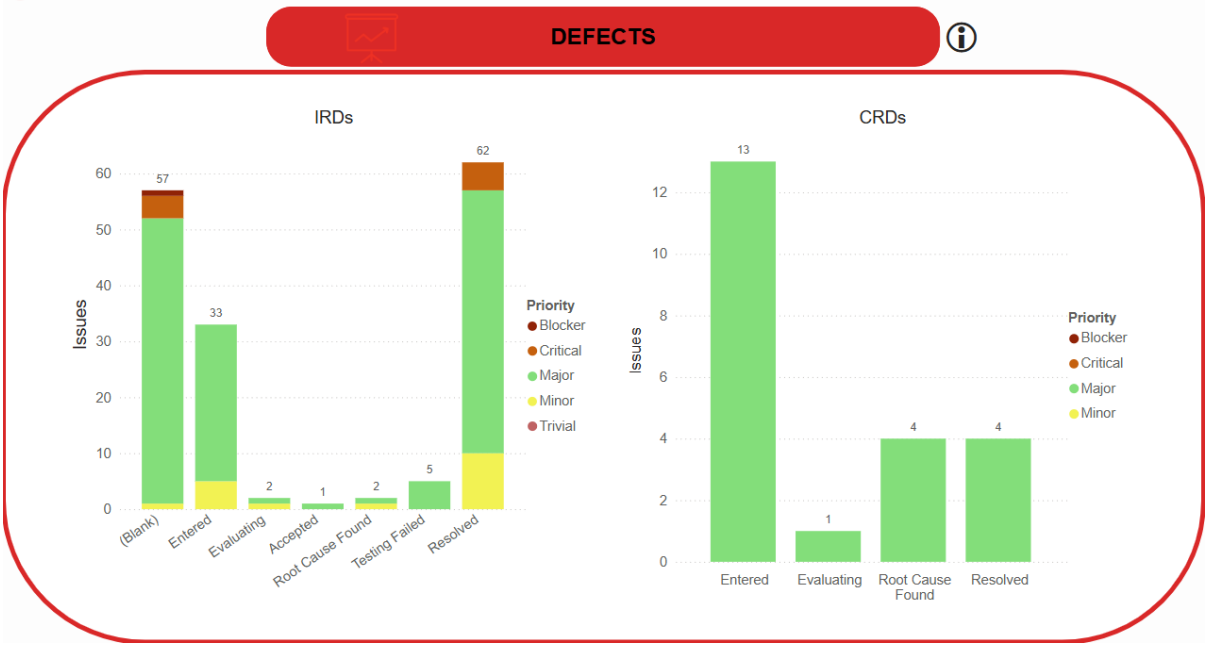


Figure 4.11. Defects Expanded View

In this view, the panel displays the same statuses and priorities as in the global overview and additionally includes IRDs alongside the CRDs shown in the global overview.

4.2.7 Quality Assurance

This panel displays the various escalation levels within the project and the corresponding issues associated with each level.

The metrics used to determine the color rules are as follows:

- GREEN: E1 <= 3.
- RED: E3 or above <> 0.
- YELLOW: Other cases.

The data originates from the **Engineering Operations Metrics and Tools** team and is extracted from the webpage JSON file all.json that is originated from JIRA database.

This is how the panel appears in the global overview of the project:

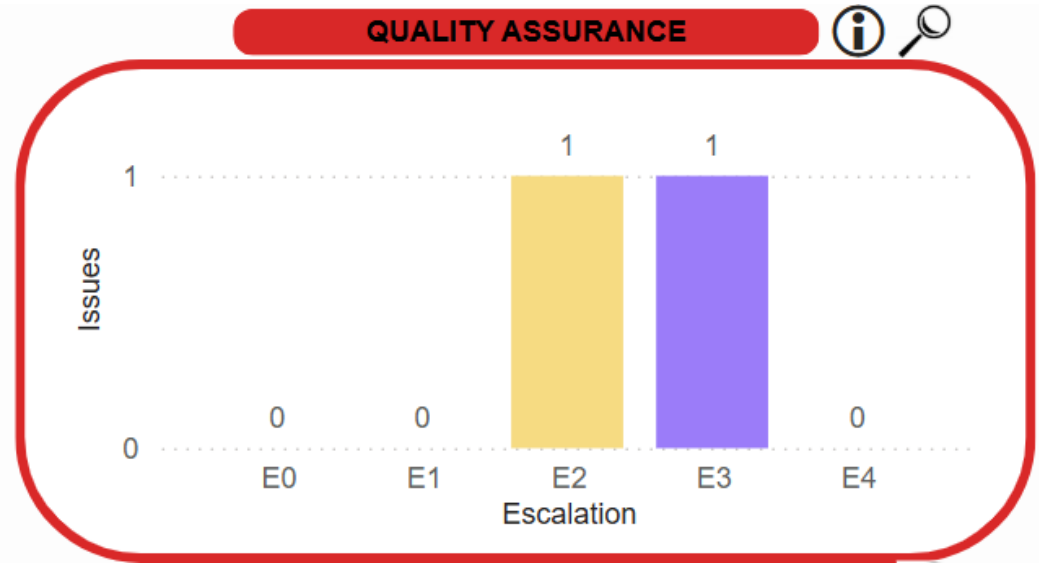
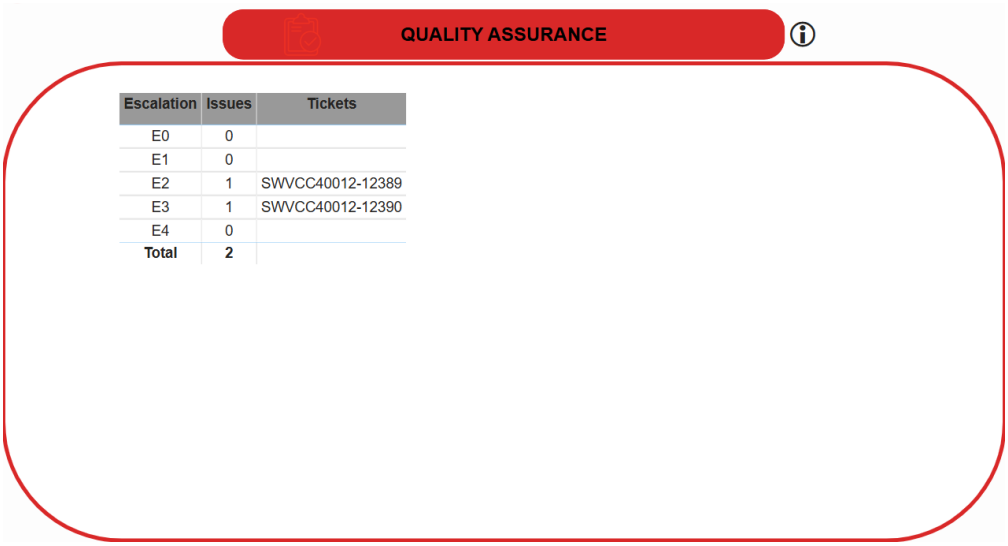


Figure 4.12. Quality Assurance's Global View

This is the display for the expanded panel:



Escalation	Issues	Tickets
E0	0	
E1	0	
E2	1	SWVCC40012-12389
E3	1	SWVCC40012-12390
E4	0	
Total	**2**	

Figure 4.13. Quality Assurance's Expanded View

In this view, in addition to the escalation levels and their associated issues, the panel also indicates the different tickets to which they belong.

4.2.8 Budget

This panel displays the project's net budget broken down by month, as well as the total net budget for the entire year. For each month, the following metrics are shown:

- Net Last: The most recent net result available from the data.
- Net Final: The initial final budget of the project.

This is how to calculate the current Year Net Deviation percentage:

```
Percentage of Increment=
var net_final = SUM(BudgetFinal[N])
var net_last = SUM(BudgetLast[N])
RETURN
If(net_final <> 0,
  ABS((net_last - net_final) / net_final)), BLANK())
```

The metrics used to determine the color rules are as follows:

- GREEN: Net Forecast Accuracy <= 1%.
- YELLOW: Net Forecast Accuracy > 1% && Net Forecast Accuracy <= 5%.
- RED: Net Forecast Accuracy > 5%.

The data originates from **PCM**, and it is extracted from the following Excel files:

- WOLifetime.xlsx
- WOForecast.xlsx

This is how the panel appears in the global overview of the project:

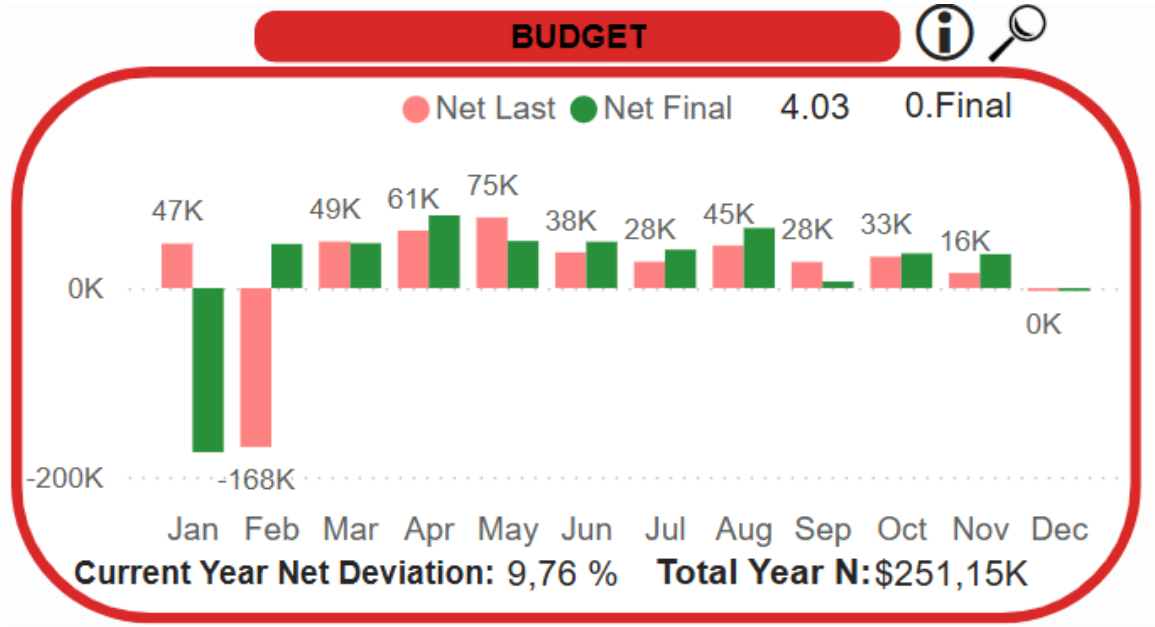


Figure 4.14. Budget's Global View

This is the display for the expanded panel:

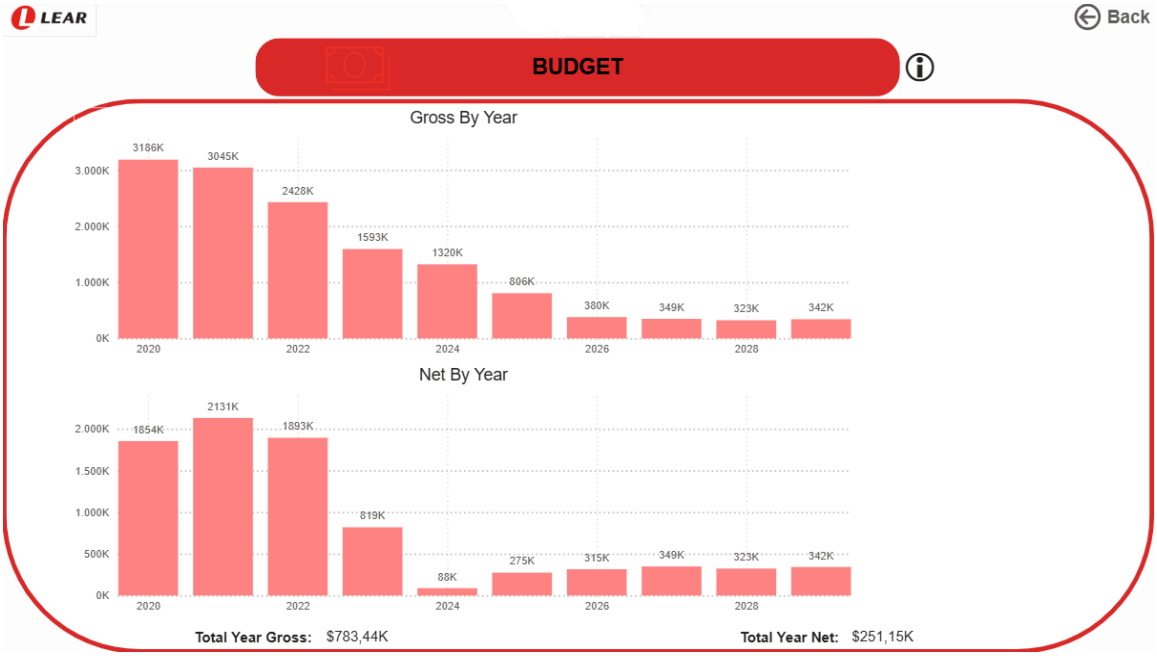


Figure 4.15. Budget's Expanded View

In this view, there are two separate graphs:

- The first graph shows the gross amounts by year.
- The second graph shows the net amounts by year.

At the bottom of the panel, the totals for the current year are displayed for both net and gross amounts.

4.2.9 Lifetime CAPEX

This panel displays the PARs assigned to the project. Each PAR can hold multiple statuses simultaneously.

- Invoiced: Represents, as a percentage, the PARs with a Closed status.
- PO Approved: Represents, as a percentage, the PARs with Soft Closed and Issued statuses.
- PO Pending: Represents, as a percentage, the PARs with any status not included in the categories above.

The metrics used to determine the color rules are as follows:

- RED: SUM(Invoiced & PO Approved) > 100%.
- GREEN: Other.
- YELLOW: None.

The data originates from **PCM** and **COUPA**, and it is extracted from the following Excel files:

- budget_line_list.xlsx
- Po_List.xlsx

This is how the panel appears in the global overview of the project:

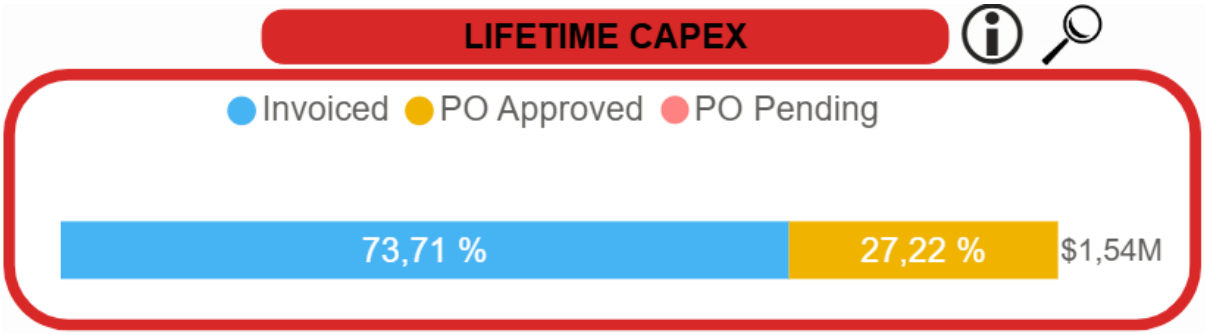


Figure 4.16. Lifetime CAPEX's Global View

This is the display for the expanded panel:

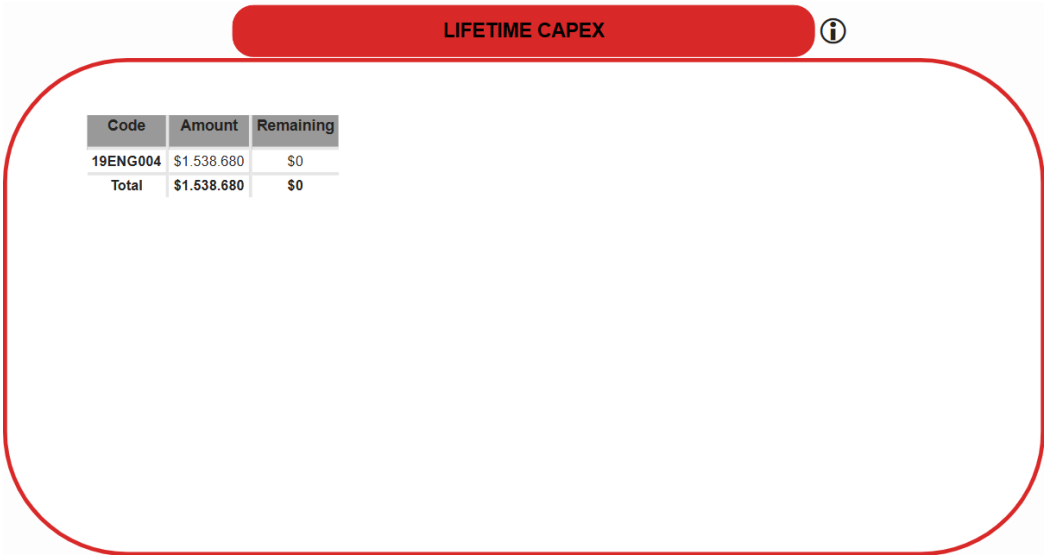


Figure 4.17. Lifetime CAPEX's Expanded View

In this view, the table displays the different PARs, the initial amount allocated at the beginning of the project, and the remaining balance of those funds.

4.2.10 Lifetime Expenses

This panel displays the WO Codes assigned to the project. Each WO Code can hold multiple statuses simultaneously.

- Invoiced: Represents, as a percentage, the WO Codes with a Closed status.
- PO Approved: Represents, as a percentage, the WO Codes with Soft Closed and Issued statuses.
- PO Pending: Represents, as a percentage, the WO Codes with any status not included in the categories above.

The metrics used to determine the color rules are as follows:

- RED: SUM(Invoiced & PO Approved) > 100%.
- YELLOW: SUM(Invoiced & PO Approved) >= 90%.
- GREEN: Other.

The data originates from **PCM** and **COUPA**, and it is extracted from the following Excel files:

- WOLifetime.xlsx
- Po_List.xlsx

This is how the panel appears in the global overview of the project:

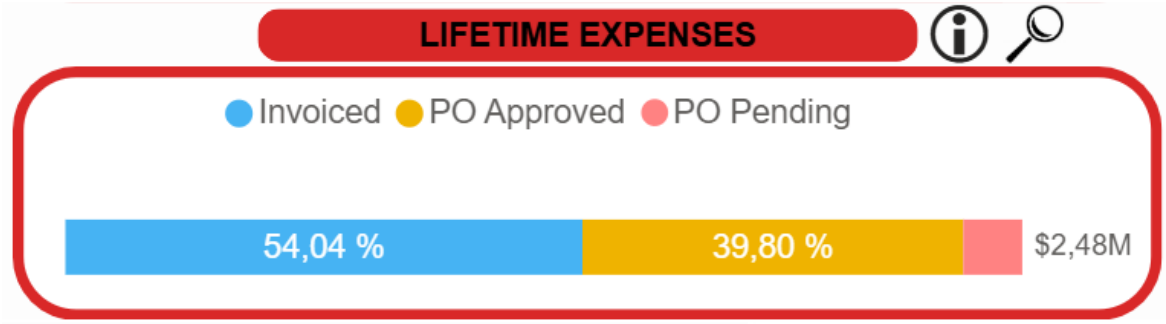


Figure 4.18. Lifetime Expenses Global View

This is the display for the expanded panel:

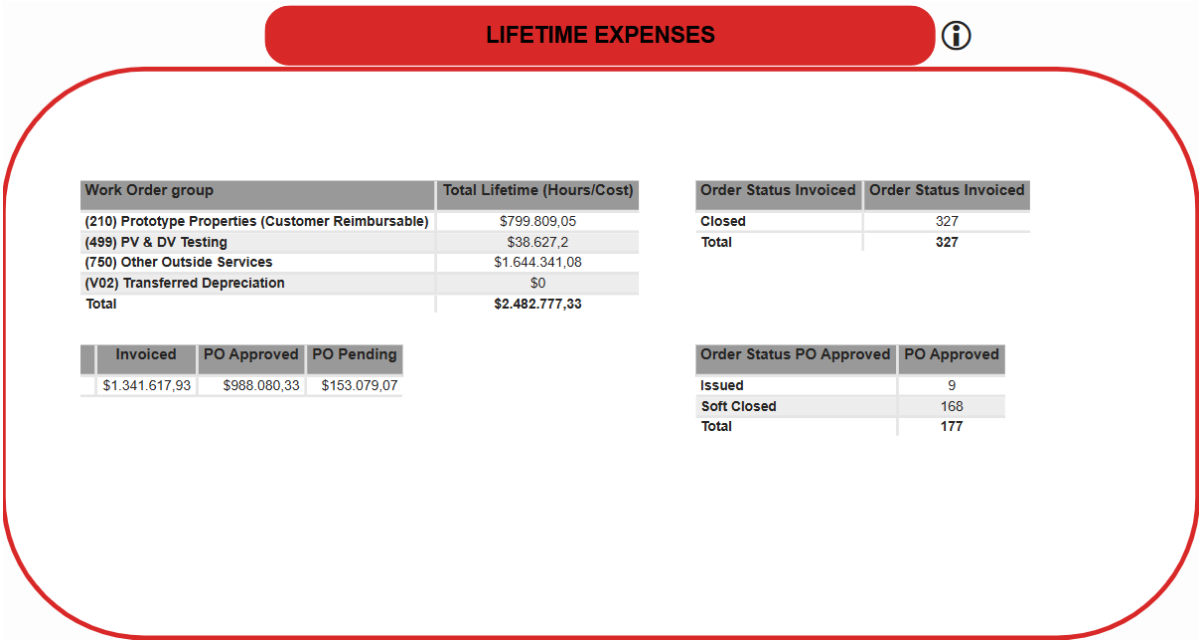


Figure 4.19. Lifetime Expenses Expanded View

In this view, there are four distinct tables:

- The first table presents the various work order groups associated with the project, along with their respective totals.
- The second table shows the different percentages in the global view in dollars.
- The third table shows the statuses of invoiced items and their corresponding totals.
- The fourth table presents the statuses of PO approved items and their respective totals.

4.2.11 Compliance Indexes

This panel displays the different compliance indexes by their metric names (PCI and QI). For each metric, multiple associated disciplines are listed, each with its corresponding percentage.

Another feature is that clicking on a percentage opens a new window with a dedicated dashboard for the discipline associated with that percentage.

The metrics used to determine the color rules are as follows:

- YELLOW: Any index below 85%.
- RED: Any index below 50%.
- GREEN: Other cases.

The data originates from the **Engineering Operations Metrics and Tools** team and is extracted from various webpage JSON files originated from project databases:

- all.json
- %Y-%m-%d.json, where the placeholders are replaced with the corresponding current date.

This is how the panel appears in the global overview of the project:

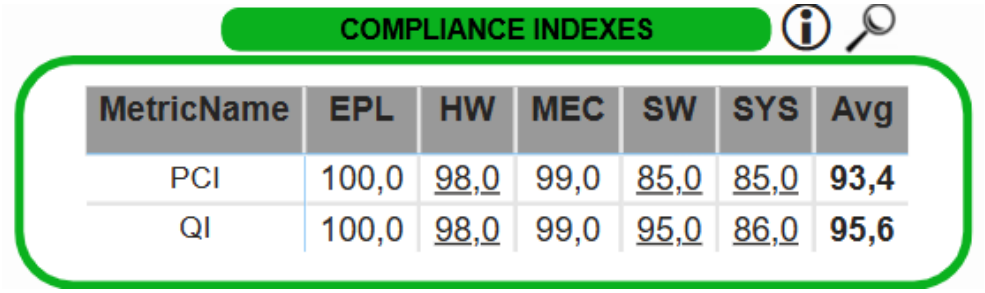


Figure 4.20. Compliance Indexes Global View

This is the display for the expanded panel:

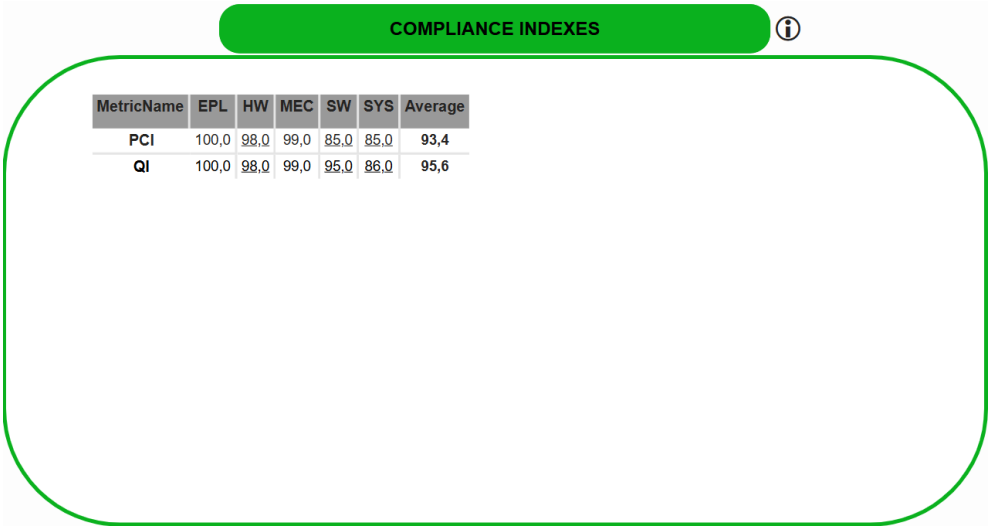


Figure 4.21. Compliance Indexes Expanded View

In this view, the display currently remains the same as in the global view of the dashboard.

4.2.12 Material Cost Management

This panel's objective is to improve the piece/price ratio by introducing changes in design and materials. It includes the following elements:

- Target: Represents the project's cost-saving or performance improvement goal.
- Roadmap: Represents the cumulative progress or savings achieved so far through implemented changes.
- GAP: Represents the difference between the Target and the current progress (Roadmap), indicating what is still needed to reach the goal.
- % Target Booked: Represents the percentage of the Target that has been booked.

The percentage of the Target Booked is calculated like this:

```
CALCULATE(
    IF(
        ISBLANK(
            DIVIDE(
                SUM(MCM OIP data[Impact]),
                SUM(MCM ALL PERFORMANCES[TOTAL Target])
            ) * 100
        ),
        "-",
        DIVIDE(
            SUM(MCM OIP data[Impact]),
            SUM(MCM ALL PERFORMANCES[TOTAL Target])
        ) * 100
    ),
    MCM OIP data[Status] = "B"
)
```

There are currently no available metrics to determine the appropriate color rules for this panel.

The data originates from **CTO Team Tracking Sheet**, and it is extracted from the following Excel files:

- %Y OIP CTO Roadmap & Regional info.xlsx, where the placeholder %Y is replaced with the corresponding year.
- Po_List.xlsx

This is how the panel appears in the global overview of the project:

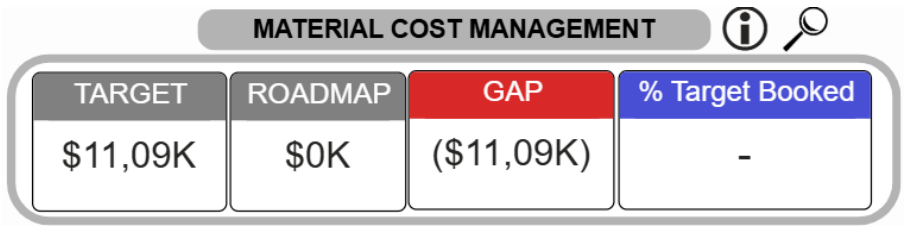


Figure 4.22. MCM Global View

This is the display for the expanded panel:

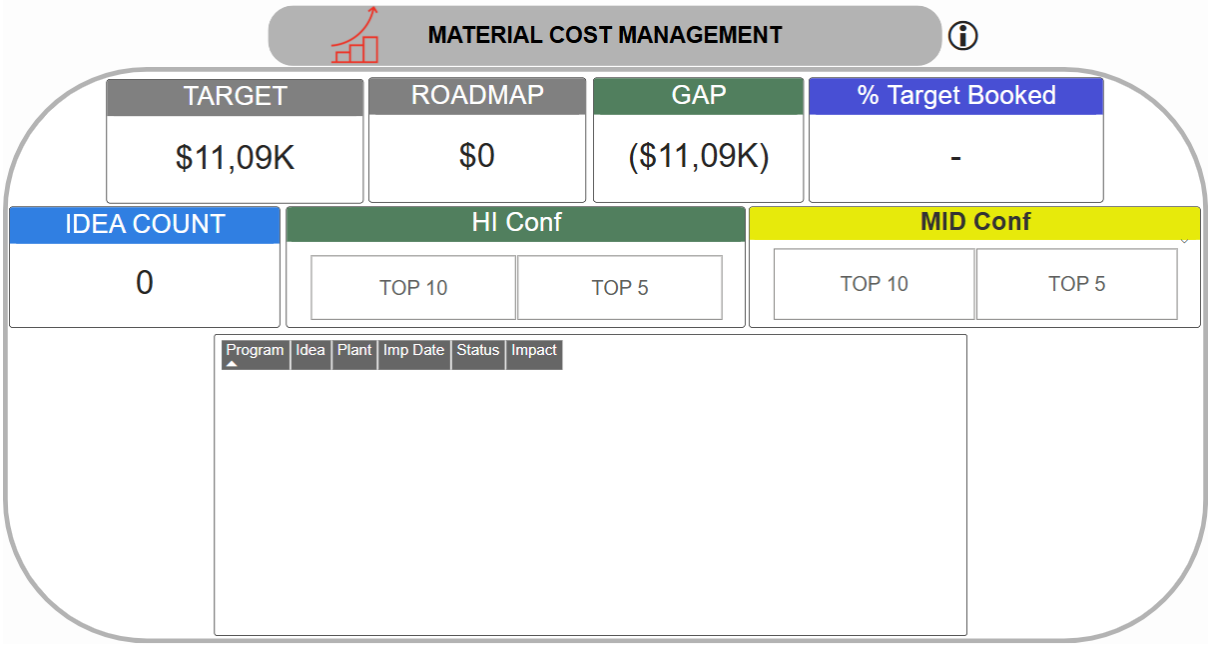


Figure 4.23. MCM Expanded View

In this view, in addition to all previously mentioned metrics, several new ones are introduced. The first is the count of ideas associated with the project. There is also a table containing the following columns:

- Program: The name of the project.
- Idea: The specific idea under consideration.
- Plant: The location of the production site.
- Imp Date: The implementation date.
- Status: The various statuses that an idea can have.
- Impact: The financial impact of the idea on the project (in dollars).

5 Implementation

5.1 Tool Selection

The election of tools used in the project is very important because it allows us to represent the data in a visually appealing format and ensures quick access for all users who need to view the dashboard. Additionally, it is crucial that this implementation is scalable and refreshed daily.

5.1.1 Basic Architecture

Power BI

Power BI is a powerful business analytics tool developed by Microsoft. It helps users connect to various data sources, visualize data, and share insights across an organization.

Power BI Desktop:

This is the primary tool for creating reports and visualizations. It allows users to connect to various data sources, transform data, and design interactive reports.

Power BI Service:

An online platform where users can publish, share, and collaborate on reports and dashboards. It supports real-time data updates (with automatic refreshes) and interactive visualizations.

Power BI Mobile Apps:

Available for iOS, Android, and Windows devices, these apps enable users to access and interact with reports and dashboards on the go. In this project, we didn't use this part. We centered in the two before this one.

Key Features

- **Data Connectivity**: Power BI can connect to a wide range of data sources, including Excel, SQL Server, Azure, and various cloud-based services.
- **Data Transformation**: With Power Query, users can clean, transform, and prepare data for analysis. For example, remove duplicates, handle missing values or filtrate data that it is not used.
- **Measure**: Once the data is transformed, you can also calculate data from different sources with the DAX language, like calculating the average of various columns.
- **Visualization**: Create a variety of visualizations, such as charts, graphs, maps, and tables, to represent data in an easily understandable format.

AI Integration: Power BI includes AI capabilities to help users uncover insights, detect patterns, and make predictions:

- **Custom Visuals**: Users can create custom visuals or use those developed by the community to enhance their reports.
- **Embedded Analytics**: Power BI reports and dashboards can be embedded into other applications, such as Microsoft Teams, SharePoint, or custom web apps.

Advanced Capabilities

- **Real-Time Analytics**: Power BI supports real-time data streaming and analytics, allowing users to monitor live data and make timely decisions.

- **Security and Governance**: Power BI provides robust security features, including row-level security, data encryption, and compliance with industry standards.
- **Collaboration**: Users can collaborate on reports and dashboards, share insights, and provide feedback within the Power BI service.
- **Export**: Export your reports and visualizations to various formats, such as PDF or PowerPoint.

Python

It is a high-level, general-purpose programming language known for its readability and versatility. The libraries most used in this project are:

- To filtrate the data: pd, re, datetime, io.
- To read and download excels, Json's, .csv: requests, HTTPBasicAuth, dataframes.
- To save the data: pd.ExcelWriter, dataframes.

Tool Selection Justification

Tool	Used in other LearEng Groups	Availability	Support available	Initial knowledge	Tool flexibility	Auto refresh	Points
Java	No	Yes	No	Yes	No	Yes	3
Power BI + Python	Yes	Yes	Yes	No	Yes	Yes	5
Excel + Python	Yes	Yes	Yes	Yes	No	No	4
Foundry	No	No	No	No	Yes	Yes	2

Table 5.1. Tool Selection Explanation

The selection of tools for this project was based on a combination of strategic, technical, and practical considerations. One of the main objectives was to leverage applications already in use within the company, particularly those designed for creating dashboards. Licensing constraints also played a role; for example, some tools, such as Foundry, were not initially available, which limited their usability at the start of the project.

Another important factor in the selection process was my own familiarity with the various tools. While some options were already known to me, others represented an opportunity to expand my technical skillset. Ultimately, **Power BI combined with Python** was selected as the final toolset for the project.

This decision was supported by several compelling reasons, as reflected in the evaluation table:

- **Used in Other LearEng Groups**: Power BI + Python is already in use by other teams within the company. This not only supports consistency across groups but also provides the opportunity to learn from existing implementations.
- **Availability and Support**: Both tools are fully available and supported within the organization, ensuring seamless access and assistance when required.

- **Tool Flexibility**: Power BI offers a high degree of flexibility in terms of dashboard design and data interaction, which enables the creation of dynamic and tailored visualizations.
- **Auto-Refresh Capability**: A key advantage of Power BI is its ability to refresh dashboards automatically, ensuring that users always see the most up-to-date data without manual intervention.

In addition to Power BI's visualization strengths, **Python** was chosen as the preferred tool for data processing and filtration. Python's rich library ecosystem allows for advanced data manipulation and integration with various data formats, such as JSON and Excel (which are commonly used in this project). Furthermore, Python provides fast processing times and versatile data handling capabilities, including saving and exporting data in multiple formats.

Compared to other tools in the evaluation, **Power BI + Python** achieved the highest score (5 points), demonstrating its superiority in key areas such as organizational alignment, support, functionality, and flexibility.

In conclusion, the integration of Power BI and Python not only met the technical and operational needs of the project but also provided me with the opportunity to develop new skills in data visualization and programming. This made it the most strategic and effective choice for delivering a high-quality, maintainable solution.

5.1.2 Data Storage Selection

During the development of the project, selecting an appropriate method for data storage became a key decision, as it directly impacted how data could be accessed and connected to the dashboard.

The initial proposal was to store the data, extracted and processed using Python, in Excel files on OneDrive. This seemed like a suitable approach, as Power BI integrates easily with OneDrive, allowing dashboards to automatically update when the source files change. However, a significant limitation emerged: although the dashboard could connect to OneDrive, the Python script could not upload or download data to/from OneDrive. To enable such interactions, it was necessary to configure authentication through Azure, which involved obtaining a client ID and setting up additional permissions. This was an approach that proved too complex given the access restrictions and lack of administrative control available during the project.

The second alternative was to use the SVN (Subversion) server as the data storage solution. In this case, it was possible to download data and process it using Python, as long as the script was run with a user who had access to the SVN repository. However, uploading data to the SVN server using Python was much more difficult. While there are Python libraries that theoretically support uploading to SVN, their implementation was complicated and required specific configurations, such as managing access tokens and setting up permissions.

To address these limitations, the final solution was to create a shared network folder accessible to all relevant team members. In this structure, the data was organized by project, with each project having its own subfolder containing multiple Excel files. These files stored the processed data extracted from the various platforms used throughout the project. This

approach allowed Python scripts to read from and write to the shared location without requiring complex permission setups, while also ensuring a clear and organized structure for data management. Although it did not offer the same level of automation as a cloud-based solution, it provided a practical and accessible alternative that met the project's operational needs.

6 Evaluation

To validate the correct functioning of the developed dashboard system and ensure that it meets the functional requirements outlined in the specification phase, a comprehensive suite of test cases has been designed and executed. These tests are directly derived from the use cases previously analyzed, including core functionalities such as opening the dashboard, interacting with various project panels, and navigating through detailed project information.

The primary objective of this testing phase is to verify that the system responds appropriately to both valid and invalid user interactions, maintains data integrity, and handles edge cases effectively. Each test case is structured to clearly define the scenario being evaluated, the expected system behavior, and the actual outcome observed during execution.

Use Case 01. Open Dashboard		
User with valid session opens dashboard.	Dashboard view loads with available panels.	Pass
Unauthenticated user tries to access dashboard.	Redirected to login or sees access denied.	Pass
User selects a dashboard from list.	Selected dashboard loads.	Pass
Dashboard displays correct data and layout.	All expected widgets/panels are rendered.	Pass
User clicks on panel within dashboard.	Panel opens (delegates to respective use case).	Pass
User clicks on a link in a panel table.	Link action is triggered / information is shown.	Pass
Panel color updates in real time based on new status.	Each panel refreshes and updates its color immediately when the underlying status changes.	Pass

Table 6.1. Test Use Case 01. Open Dashboard

Use Case 02. Milestones Panel		
Milestones Panel opens when selected in dashboard.	Panel appears with correct title/data.	Pass
Milestones data is loaded from backend.	Phases and milestones show up in structured format.	Pass
User navigates back to Dashboard.	Returns to previous dashboard view.	Pass
Color updates correctly when status changes dynamically.	Panel color updates in real time based on new status.	Pass

Table 6.2. Test Use Case 02. Milestones Panel

Use Case 03. Staffing Panel		
Staffing Panel opens when selected in dashboard.	Panel appears with correct title/data.	Pass
Staffing data is loaded from backend.	Phases and milestones show up in structured format.	Pass
User navigates back to Dashboard.	Returns to previous dashboard view.	Pass
Color updates correctly when status changes dynamically.	Panel color updates in real time based on new status.	Pass

Table 6.3. Test Use Case 03. Staffing Panel

Use Case 04. Requirements Panel		
Requirements Panel opens when selected in dashboard.	Panel appears with correct title/data.	Pass
Requirements data is loaded from backend.	Phases and milestones show up in structured format.	Pass
User navigates back to Dashboard.	Returns to previous dashboard view.	Pass
Color updates correctly when status changes dynamically.	Panel color updates in real time based on new status.	Pass

Table 6.4. Test Use Case 04. Requirements Panel

Use Case 05. Change Management Panel		
Change Management Panel opens when selected in dashboard.	Panel appears with correct title/data.	Pass
Change Management data is loaded from backend.	Phases and milestones show up in structured format.	Pass
User navigates back to Dashboard.	Returns to previous dashboard view.	Pass
Color updates correctly when status changes dynamically.	Panel color updates in real time based on new status.	Pass

Table 6.5. Test Use Case 05. Change Management Panel

Use Case 06. Defects Panel		
Defects Panel opens when selected in dashboard.	Panel appears with correct title/data.	Pass
Defects data is loaded from backend.	Phases and milestones show up in structured format.	Pass
User navigates back to Dashboard.	Returns to previous dashboard view.	Pass
Color updates correctly when status changes dynamically.	Panel color updates in real time based on new status.	Pass

Table 6.6. Test Use Case 06. Defects Panel

Use Case 07. Quality Panel		
Quality Panel opens when selected in dashboard.	Panel appears with correct title/data.	Pass
Quality data is loaded from backend.	Phases and milestones show up in structured format.	Pass
User navigates back to Dashboard.	Returns to previous dashboard view.	Pass
Color updates correctly when status changes dynamically.	Panel color updates in real time based on new status.	Pass

Table 6.7. Test Use Case 07. Quality Panel

Use Case 08. Budget Panel		
Budget Panel opens when selected in dashboard.	Panel appears with correct title/data.	Pass
Budget data is loaded from backend.	Phases and milestones show up in structured format.	Pass
User navigates back to Dashboard.	Returns to previous dashboard view.	Pass
Color updates correctly when status changes dynamically.	Panel color updates in real time based on new status.	Pass

Table 6.8. Test Use Case 08. Budget Panel

Use Case 09. CAPEX Panel		
CAPEX Panel opens when selected in dashboard.	Panel appears with correct title/data.	Pass
CAPEX data is loaded from backend.	Phases and milestones show up in structured format.	Pass
User navigates back to Dashboard.	Returns to previous dashboard view.	Pass
Color updates correctly when status changes dynamically.	Panel color updates in real time based on new status.	Pass

Table 6.9. Test Use Case 09. CAPEX Panel

Use Case 10. Expenses Panel		
Expenses Panel opens when selected in dashboard.	Panel appears with correct title/data.	Pass
Expenses data is loaded from backend.	Phases and milestones show up in structured format.	Pass
User navigates back to Dashboard.	Returns to previous dashboard view.	Pass
Color updates correctly when status changes dynamically.	Panel color updates in real time based on new status.	Pass

Table 6.10. Test Use Case 10. Expenses Panel

Use Case 11. Compliance Panel		
Compliance Panel opens when selected in dashboard.	Panel appears with correct title/data.	Pass
Compliance data is loaded from backend.	Phases and milestones show up in structured format.	Pass
User navigates back to Dashboard.	Returns to previous dashboard view.	Pass
Color updates correctly when status changes dynamically.	Panel color updates in real time based on new status.	Pass

Table 6.11. Test Use Case 11. Compliance Panel

Use Case 12. MCM Panel		
MCM Panel opens when selected in dashboard.	Panel appears with correct title/data.	Pass
MCM data is loaded from backend.	Phases and milestones show up in structured format.	Pass
User navigates back to Dashboard.	Returns to previous dashboard view.	Pass
Color updates correctly when status changes dynamically.	Panel color updates in real time based on new status.	Pass

Table 6.12. Test Use Case 12. MCM Pane

7 Project Planning

The project was structured across multiple phases from July 2024 to June 2025, as visualized in the *Figure 7.1*. Tasks were planned both sequentially and in parallel to optimize resources and accelerate development where possible.

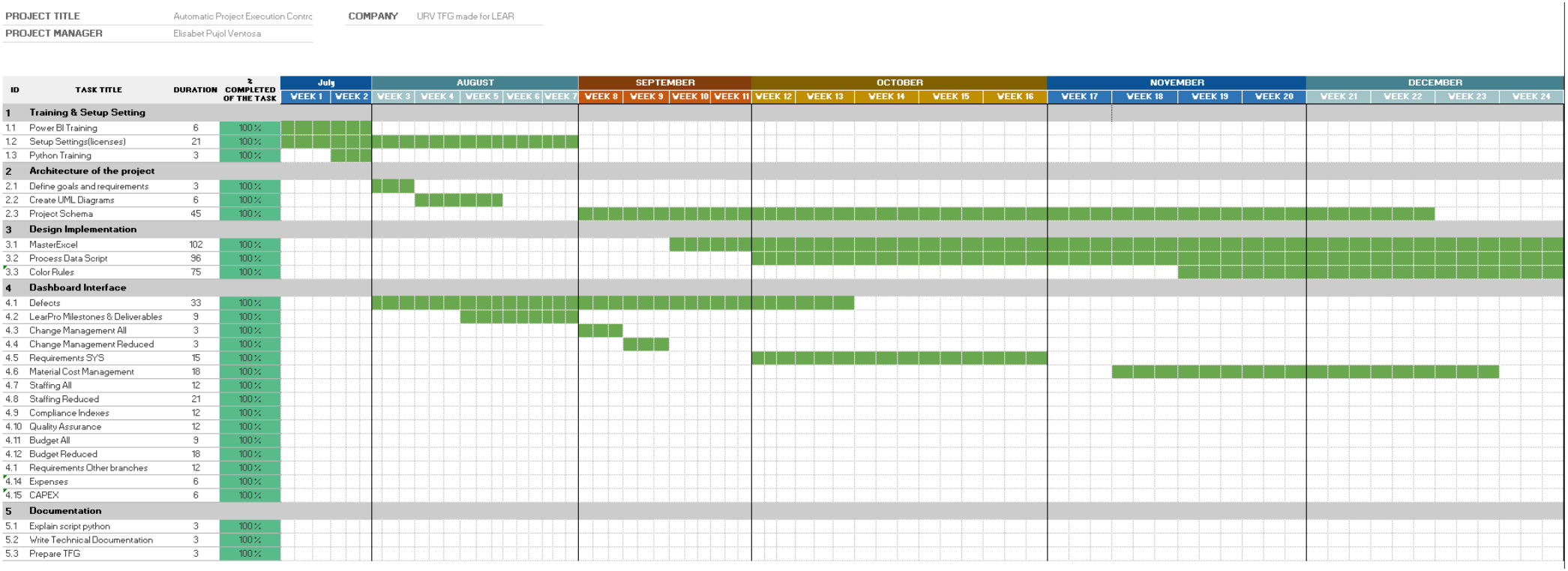
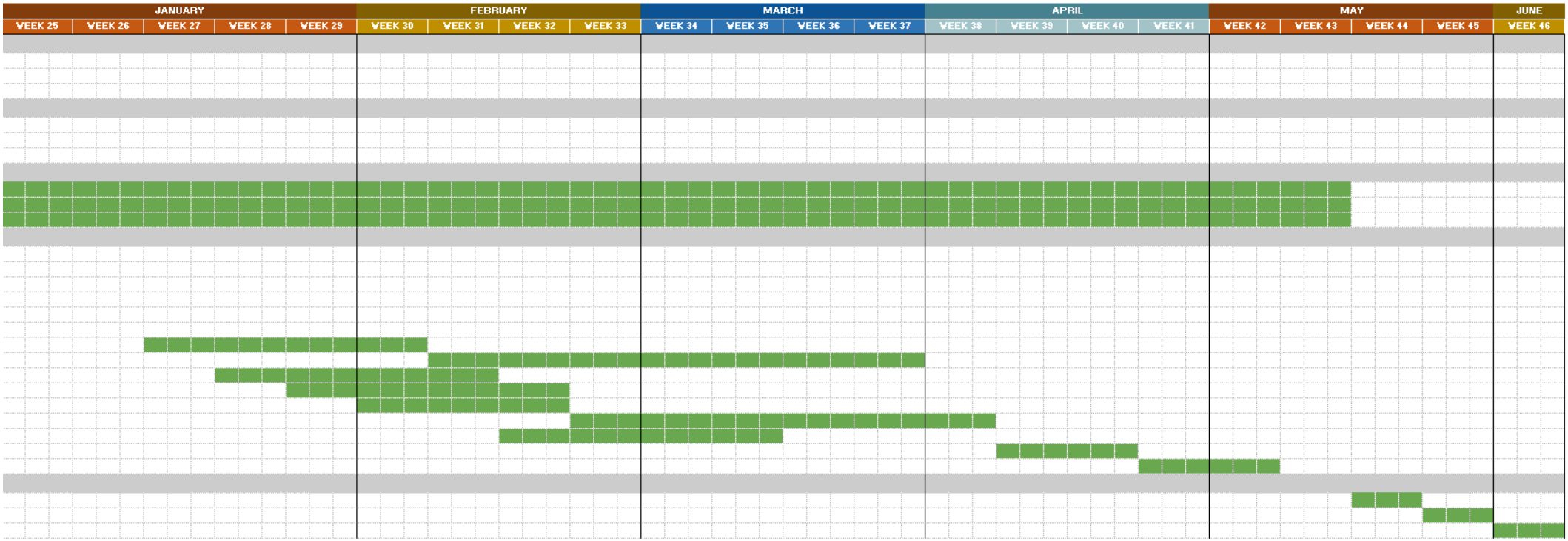


Figure 7.1. Gantt Diagram


JANUARY
FEBRUARY
MARCH
APRIL
MAY
JUNE
WEEK 25
WEEK 26
WEEK 27
WEEK 28
WEEK 29
WEEK 30
WEEK 31
WEEK 32
WEEK 33
WEEK 34
WEEK 35
WEEK 36
WEEK 37
WEEK 38
WEEK 39
WEEK 40
WEEK 41
WEEK 42
WEEK 43
WEEK 44
WEEK 45
WEEK 46

8 Conclusion

The implementation of this project successfully addressed the primary objectives outlined at the beginning, offering a scalable and efficient solution for project data visualization and management at Lear. Each objective was tackled through the thoughtful integration of Power BI, Python, and strategic data handling methods.

8.1 Achievement of Objectives

Centralized Data Collection

The dashboard enabled the **c**onsolidation of information from various platforms into a single, accessible location, eliminating the previous fragmentation of data. This centralization made it significantly easier to monitor project status holistically and reduced the effort required to manually gather and combine datasets.

Authorized Access Only

To maintain confidentiality and data integrity, access to the dashboard was restricted to authorized personnel only. This ensures that sensitive project information is only visible to those directly involved, in line with company security policies and best practices.

Correlation of Previously Dispersed Data

One of the project's major achievements was the improvement of data correlation. By consolidating previously scattered data into a unified format, the dashboard allowed for meaningful cross-platform comparisons and analysis, something that was not feasible before.

User-Friendly Interface

Power BI was selected for its capability to create clear, visually intuitive dashboards. The standardized visual format, along with the use of color-coded indicators (green, yellow, red), provided users with an easy and immediate understanding of the project's health and critical areas requiring attention.

Automatic Data Refresh

The inclusion of **automatic data refresh mechanisms** ensured that the dashboard remained current without requiring manual intervention. This improved the reliability and timeliness of the information presented, supporting more agile and informed decision-making.

Quick and Practical Access

While cloud storage solutions posed technical limitations, the implementation of a shared network folder emerged as a reliable alternative. This approach ensured quick access to the most up-to-date data while avoiding permission-related complications, striking a balance between practicality and security.

Delivery of a Fully Functional Dashboard

One of the most significant accomplishments of the project was the successful delivery of a fully functional dashboard. Not only did the final product meet all technical and visual expectations, but it also integrated seamlessly with Python scripts and met real operational requirements. It provided a solid, working solution, not just a concept or prototype, that could be used immediately by the intended team.

8.2 Additional Objectives Identified During Development

As the project evolved, several new needs and improvement opportunities were identified, leading to additional enhancements:

Color Rules for Status Indicators

A color-coded status system was implemented to help users quickly identify the state of each project area, green for normal, yellow for warnings, and red for critical issues.

Links to Other Dashboards

To enhance navigation and data exploration, interactive links to related dashboards were added, allowing users to move across different views and projects easily.

Expanded Panel Views

Each panel in the dashboard now supports an expanded view, enabling deeper analysis without overwhelming the default interface.

Scalability to Other Projects

Although the dashboard currently supports three projects, it has been structured to be easily scalable across all future projects within the company, laying the groundwork for broader adoption.

8.3 Future Steps

While the project has successfully delivered a fully functional and valuable dashboard solution, several future improvements have already been identified to enhance its impact and scalability. These steps are categorized into short-, mid-, and long-term goals:

Short-Term Plan: Company-Wide Adoption

The immediate goal is to expand the use of this dashboard across the entire organization. By making it available as a resource for a broader group of stakeholders, the dashboard can serve as a standard tool for quickly understanding project status. Encouraging adoption among teams will help unify reporting practices and improve cross-departmental communication.

Mid-Term Plan: Enhanced Automation and Integration

In the mid-term, the focus will be on increasing automation and connectivity:

- Automate the remaining manual data downloads, eliminating the need for human intervention and reducing the risk of delays or errors.
- Integrate the dashboard with additional internal dashboards and web platforms, creating a more interconnected and data-rich environment for users to navigate and analyze project performance in a more holistic way.

Long-Term Plan: Role-Based Views and Deeper Insights

For the long-term evolution of the dashboard, the objective is twofold:

- Expand the level of detail and usefulness of each panel, offering more actionable insights and tailored visualizations that adapt to user needs.
- Implement role-based views, allowing users to access customized interfaces based on their project roles (e.g., team lead, manager, quality assurance). This would ensure that each user sees the most relevant information, improving efficiency and decision-making across different project layers.

These future developments aim to build upon the strong foundation established in this project, continuously increasing the value and adaptability of the dashboard as both a strategic and operational tool.

Bibliography

3 Ways to Add Last Refresh Date in Power BI - Power Tech Tips. (2023, October 19). https://www.powertechtips.com/add-last-refresh-date-power-bi/

B, Chris. (2016, September 6). How to get JSON from webpage into Python script [Forum post]. Stack Overflow. https://stackoverflow.com/q/12965203

Bobbitt, Zach. (2022, July 11). How to Count Duplicates in Pandas (With Examples). Statology. https://www.statology.org/pandas-count-duplicates/

Casado, Enrique Benito. (2021, July 15). SecureAuth cause - ShareplumRequestError: Shareplum HTTP Post Failed : 403 Client Error: Forbidden for url [Forum post]. Stack Overflow. https://stackoverflow.com/q/68390250

deoos. (2020, September 27). Power BI. Matrix two columns under one column [Forum post]. Stack Overflow. https://stackoverflow.com/q/56626197

Extract specific data from excel python. (s.d.). Bing. Recuperat 6 June 2025, de https://www.bing.com/search?q=Extract+specific+data+from+excel+python&cvid=eaa10a704d4e437e9193284dd2e7ebc4&gs_lcrp=EgZjaHJvbWUyBggAEEUYOTIGCAEQABhAMgYIAhAAGEAyBggDEAAYQDIGCAQQABhAMgYIBRAAGEAyBggGEAAYQDIGCAcQABhAMgYICBAAGEAyCAgJEOkHGPxV0gEJMjQ4MDJqMGoxqAIAsAIA&FORM=ANAB01&PC=U531

G, Jeremy. (2022, December 24). Replacing Header with Top Row [Forum post]. Stack Overflow. https://stackoverflow.com/q/31328861

How to Create and Add Buttons in Power BI. (2023, October 6). https://www.popautomation.com/post/power-bi-create-and-add-buttons

itsPav. (2018, March 15). Hover Over Text in PowerBI [Forum post]. Stack Overflow. https://stackoverflow.com/q/49303148

jeffci. (2013, October 30). How can I get the previous week in Python? [Forum post]. Stack Overflow. https://stackoverflow.com/q/19686856

JulCsc. (s.d.). Sort one column by another column in Power BI - Power BI. Recuperat 6 June 2025, de https://learn.microsoft.com/en-us/power-bi/create-reports/desktop-sort-by-column

Lookup multiple values in DAX - SQLBI. (s.d.). Recuperat 6 June 2025, de https://www.sqlbi.com/articles/lookup-multiple-values-in-dax/

Meet101. (2017, May 1). How to send excel file in multipart/form-data with POST Requests in python? [Forum post]. Stack Overflow. https://stackoverflow.com/q/43721269

Newest Questions. (s.d.). Stack Overflow. Recuperat 6 June 2025, de https://stackoverflow.com/questions

Pandas Cheat Sheet – Dataquest. (s.d.). Recuperat 6 June 2025, de https://www.dataquest.io/cheat-sheet/pandas-cheat-sheet/

Python requests module: How to upload files (form-data)—Sling Academy. (s.d.). Recuperat 6 June 2025, de https://www.slingacademy.com/article/python-requests-module-file-upload/

s666. (2016, May 15). Using Pandas to read in excel file from URL - XLRDError [Forum post]. Stack Overflow. https://stackoverflow.com/q/37243121

sparrow. (2016, August 29). Convert text data from requests object to dataframe with pandas [Forum post]. Stack Overflow. https://stackoverflow.com/q/39213597

svn: Intuitive Subversion wrapper. (Version 1.0.1). (s.d.). [Software]. Recuperat 6 June 2025, de https://github.com/dsoprea/PySvn

Will. (2023, April 2). How to Create a Power BI Information Button (Hover and Bookmark)—Smantin Data. https://smantindata.com/power-bi-information-button/

Annex

Python Script

This section content is removed from the public version to protect Lear confidential information.