



Higher Institute
of Information
and Communication Technologies
الجامعة العليا لتكنولوجيات المعلومات والاتصال

TUNISIAN REPUBLIC
MINISTRY OF HIGHER EDUCATION
AND SCIENTIFIC RESEARCH

UNIVERSITY OF CARTHAGE

HIGHER INSTITUTE OF INFORMATION AND
COMMUNICATION TECHNOLOGIES



Université de Carthage
University of Carthage
جامعة قرطاج

END OF STUDY PROJECT REPORT



By
Mariem Rjab
Raed Said
Tarek Msolli

Academic year: 2022 - 2023

Contents

General introduction	1
1 Needs Specification	2
1.1 Introduction	2
1.2 Identification of functional needs	2
1.3 Identification of use cases	2
1.4 Identification of non-functional needs	2
1.5 Identification of actors	3
1.6 Use case diagram	4
1.7 Product backlog	4
1.8 Work environment	5
1.8.1 The SCRUM method	5
1.8.2 Softwares and tools	5
1.9 Conclusion	5
2 sprint 0	6
2.1 Introduction	6
2.2 Identification of the backlog for release 1	6
2.3 Refinement of sprint 0	6
2.3.1 Refinement of the use case 'Authenticate':	6
2.3.2 Refinement of the use case 'Register':	7
2.3.3 Refinement of the use case consult props :	8
2.4 Sprint 0 Design	9
2.4.1 "Authenticate" Use Case Design:	9
2.4.2 "Register" Use Case Design:	10
2.4.3 "consult props" Use Case Design:	11
2.5 Implementation of Sprint 0:	12
2.5.1 Authenticate	12
2.5.2 Register	12
2.5.3 Consult props	13
2.6 Conclusion:	13
3 Design of Sprint 1	14
3.1 Introduction	14
3.2 Identification of the backlog for release 1	14
3.3 Refinement of sprint 1	15
3.3.1 Refinement of the use case 'manage account':	15
3.3.2 Refinement of the use case 'manage props':	16
3.3.3 Refinement of the use case 'approve props':	17
3.4 Design Release 1	17
3.4.1 Reslease 1 concept	17
3.4.2 manage props Use Case Design:	17
3.4.3 manage profile Use Case Design:	18
3.4.4 approve props Use Case Design:	20
3.5 Implementation of Sprint 1	21
3.5.1 Manage props	21
3.5.2 Manage profile	22

3.6 Conclusion:	22
General conclusion	23

List of Figures

1.1	Prophub use case diagram	4
2.2	Refinement of the use case authenticate	7
2.3	Refinement of the use case authenticate	7
2.4	Refinement of the use case consult props	8
2.5	Authenticate class diagram	9
2.6	Authenticate sequence diagram	10
2.7	Register class diagram	10
2.8	Register sequence diagram	11
2.9	Consult props class diagram	11
2.10	Authenticate sequence diagram	12
2.11	Authenticate screenshot	12
2.12	Register screenshot	13
2.13	Consult props screenshot	13
3.14	Refinement of the use case Manage account	15
3.15	Refinement of the use case Manage props	16
3.16	Refinement of the use case Approve props	17
3.17	manage props class diagram	18
3.18	manage props sequence diagram	18
3.19	manage profile class diagram	18
3.20	manage profile sequence diagram	19
3.21	approve props class diagram	20
3.22	approve props sequence diagram	21
3.23	manage props screenshot	21
3.24	manage profile screenshot	22

List of Tables

1.1	Roles of actors in the system	3
1.2	Product backlog	4
1.3	Softwares and tools	5
2.4	Product backlog for realease 1	6
2.5	refinement of the use case 'Authenticate'	7
2.6	refinement of the use case 'Register'	8
2.7	refinement of the use case 'Consult props'	9
3.8	Product backlog for realease 1	14
3.9	refinement of the use case 'Manage account'	15
3.10	refinement of the use case 'Manage props'	16
3.11	refinement of the use case 'Approve props'	17

General introduction

Prophub, the innovative and dynamic online platform dedicated to providing a cutting-edge and user-friendly space for both game designers and gamers. Our mission is to foster a vibrant community where creative minds can collaborate, showcase their talents, and access valuable resources to take the art and science of gaming to new heights. We are committed to being the go-to destination for passionate gamers who are pushing the boundaries of what is possible in this exciting industry. We believe that gaming is not just a hobby but an art form that requires skill, creativity, and dedication. Our platform is designed to support both the gaming community and game designers by providing them with the necessary tools, resources, and inspiration to create and enjoy the best gaming experiences possible. Our platform is constantly evolving, and we are dedicated to staying ahead of the curve in terms of the latest technology, trends, and best practices. To achieve this goal, we have followed an iterative and incremental design methodology, using the SCRUM method to model our project. This approach allows us to be agile and responsive to the needs of our users and the ever-changing landscape of the gaming industry.

Chapter 1: Titled "Needs Specification," this document presents the identification of functional and non-functional requirements, actors, and use cases.

Chapter 2: Titled "Sprint 0," this is a detailed description of the first release to be studied.

Chapter 3: Titled "Sprint 1," this is a detailed description of the first release to be studied.

Chapter 1

NEEDS SPECIFICATION

1.1 Introduction

A functional requirement is a requirement that specifies an action that a system must be able to perform, without considering any physical constraints. It is a user's need from their point of view. Our project aims to address the following needs:

1.2 Identification of functional needs

A functional need is a requirement that specifies an action that a system must be able to perform, without considering any physical constraints. It is a requirement from the user's point of view. Our project aims to meet the following needs:

- Register
- Authenticate
- Manage account
- Consult props
- Manage props

1.3 Identification of use cases

- **Register:** At this stage, the user is anonymous, but to move to a more advanced stage, they must register as a 'user'.
- **Authenticate:** after the registration phase, the user is allowed to authenticate and, depending on their privileges, will be guided as either a user or a game developer otherwise they will have access as an administrator.
- **Manage account:** After registration and login , each user has access to modify their personal information in the settings section of their profile.
- **Consult props:** Now after logging to our website the user can consult the game props as a user or a game developer by interacting with others.
- **Manage props:** at this stage the game developers can upload and share their works with others to interact with it

1.4 Identification of non-functional needs

Non-functional needs are quality indicators for the execution of functional needs, which are important because they indirectly affect the outcome and performance of the user, and therefore should not be neglected. To achieve this, the following requirements must be met:

- **Reliability:** The website must function consistently without errors and must be satisfactory.
- **Errors:** Ambiguities must be signaled by well-organized error messages to guide the user and familiarize them with our website.
- **Usability and good interface:** The website must be user-friendly without requiring any effort (clear and easy to use) in terms of navigation between different pages, colors, and text formatting.
- **Security:** Our solution must prioritize the confidentiality of customers' personal data, which remains one of the most important constraints .
- **Maintainability and reusability:** The system must conform to a standard and clear architecture that allows for maintenance and reusability.

1.5 Identification of actors

An actor is a physical or legal person taking part in or affected by the action or project in question. It is therefore necessary to start by clearly specifying in relation to which action or sequence of actions we are seeking to determine who the actors are and what they are. Our platform contains three actors who directly interact with the system:

- **The user:** Any person who accesses the platform has the possibility to create and manage their account, as well as consult props and communicate with other users ..
- **The admin:** It is the supervisor who has all the control permissions to manage the users and props.

Actor	Role
Admin	Approve props
User	Authenticate Register Consult props Manage props

Table 1.1: Roles of actors in the system

1.6 Use case diagram

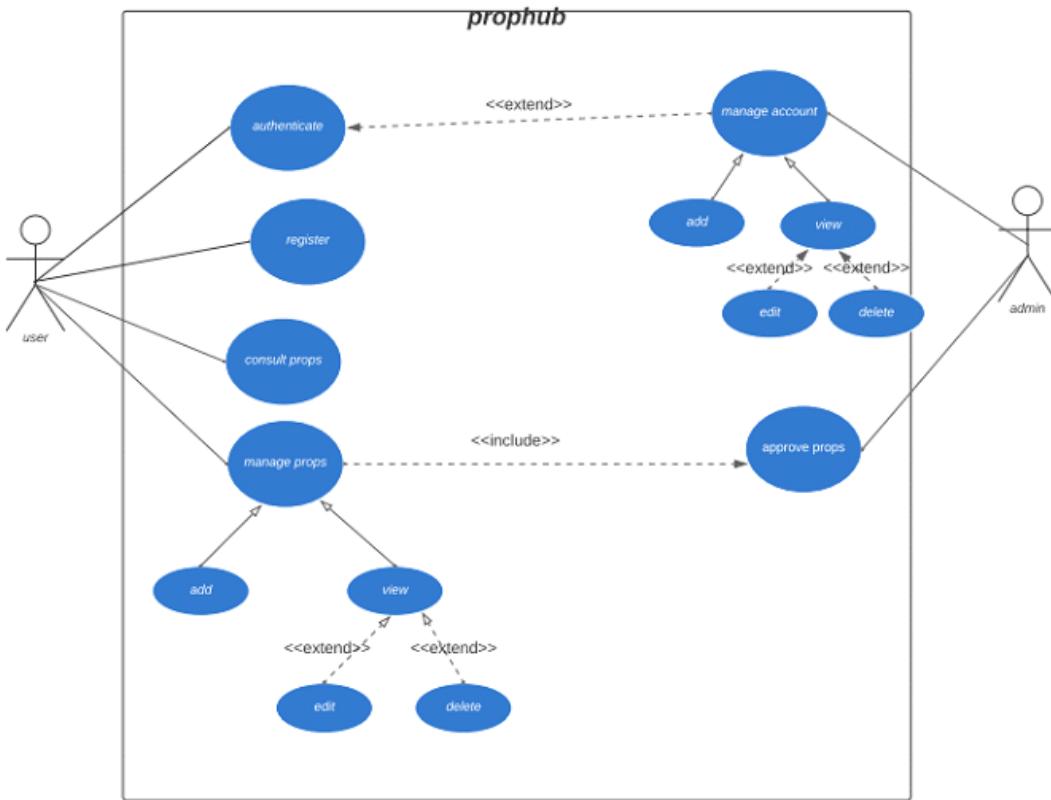


Figure 1.1: Prophub use case diagram

1.7 Product backlog

Product backlog	Priority	Estimation	Planning
As a user, I can authenticate	1	average	Sprint 0
As a user, I can register an account	1	average	Sprint 0
As a user, I can consult props	1	average	Sprint 0
As a user, I can manage props	2	average	Sprint 0
As a user, I can manage an account	2	average	Sprint 1
As an admin, I can approve props	2	average	Sprint 1

Table 1.2: Product backlog

1.8 Work environement

1.8.1 The SCRUM method

The SCRUM method is an iterative and agile project management approach that focuses on team collaboration, regular delivery of functional products, and quick response to changes. It is based on principles such as transparency, inspection, and adaptation. The Scrum framework consists of events, roles, and artifacts, which work together to provide a clear and effective structure for project management.

1.8.2 Softwares and tools

During our mini project, we used the following softwares and tools:

Software	Definition
 Visual Studio Code	Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux
	Node.js (Node) is an open source, cross-platform runtime environment for executing JavaScript code
	MySQL Workbench is a unified visual tool for database architects, developers, and DBAs. MySQL Workbench provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration, backup, and much more.
	Git is a distributed version control system that tracks changes in any set of computer files, usually used for coordinating work among programmers collaboratively developing source code during software development.
	XAMPP is a free and open-source software package that provides an easy way to install and run a web server, database server, and PHP interpreter on a local computer.
	Postman is a popular API testing and development tool that allows developers to easily design, test, and document APIs.

Table 1.3: Softwares and tools

1.9 Conclusion

In this chapter, we have performed a detailed functional breakdown of our future solution through the use of use case diagrams and the product backlog.

Chapter 2

SPRINT 0

2.1 Introduction

In this chapter, we will present the first release of the project which is 'sprint 0' that details the priority 1 use cases. The study of this sprint includes refinement, design, and implementation.

2.2 Identification of the backlog for release 1

The backlog for the first release that we present below contains a list of backlog items that need to be completed in sprint 0:

Product backlog	Priority	Estimation	Planning
As a user i can register an account	1	average	Sprint 0
As a user , I can authenticate	1	average	Sprint 0
As a user, I can consult props	1	average	Sprint 0

Table 2.4: Product backlog for realease 1

2.3 Refinement of sprint 0

In this section, we focus on the following use cases:

- authenticate
- register
- Consult props

2.3.1 Refinement of the use case 'Authenticate':

Authentication is the primary requirement for the processing and security of other use cases. In order for actors to execute their own needs, they must go through authentication. Figure 2.3.1 illustrates the use case diagram 'Authenticate':

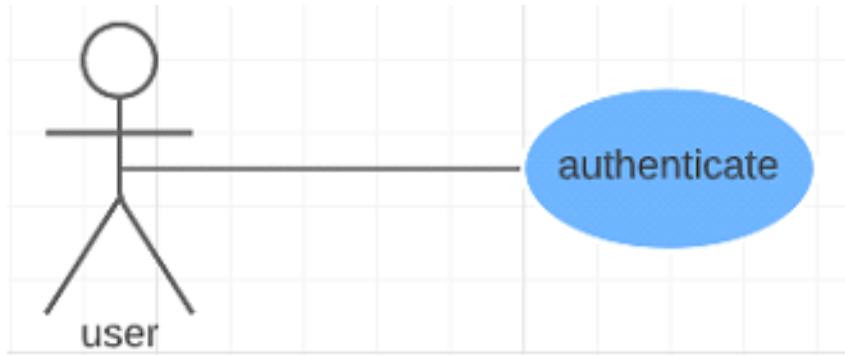


Figure 2.2: Refinement of the use case authenticate

Use case	Authenticate
Actor	User
Pre-Condition	The system is up and running
Post-Condition	User authenticated
Description of the main scenario	<p>The system displays the authentication interface.</p> <p>The user enters their email address and password.</p> <p>The user clicks the "Login" button.</p> <p>The system verifies the email address (or username) and password combination.</p> <p>The system redirects the user to the home page of the website</p>
Exception scenario	The system displays an error message if the data is incorrect

Table 2.5: refinement of the use case 'Authenticate'

2.3.2 Refinement of the use case 'Register':

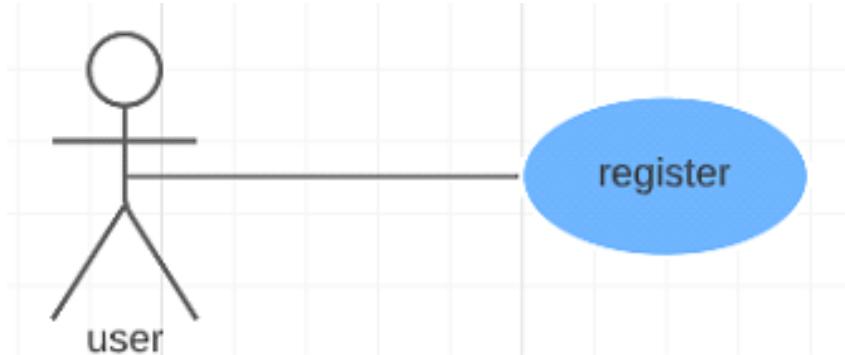


Figure 2.3: Refinement of the use case authenticate

Use case	Register
Actor	User
Pre-Condition	Email address and password are validated
Post-Condition	Registered user
Description of the main scenario	<p>The system displays the registration interface.</p> <p>The user enters their username, email address, and password.</p> <p>The user clicks the "Register" button.</p> <p>The system displays the registration interface.</p> <p>The user enters their personal information (name, surname, email address, username, display name, and telephone number).</p> <p>The user clicks the "Signup" button.</p> <p>The system verifies the information.</p> <p>The system confirms the validity of the registration with a message displayed on the screen.</p> <p>The system redirects the user to the home page of the application.</p>
Exception scenario	The system displays an error message if the data is incorrect

Table 2.6: refinement of the use case 'Register'

2.3.3 Refinement of the use case consult props :

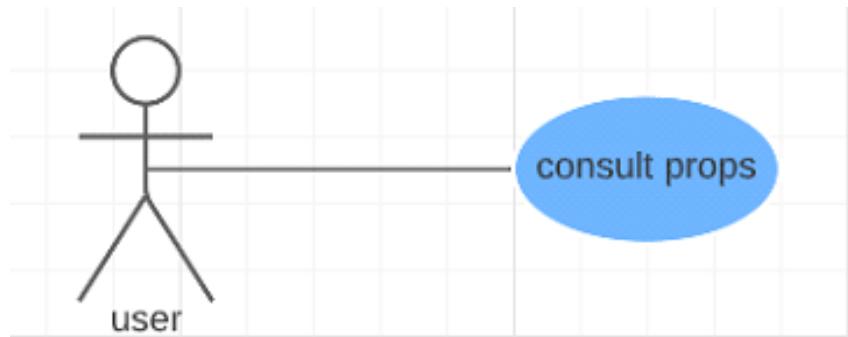


Figure 2.4: Refinement of the use case consult props

Use case	Consilt Props
Actor	User
Pre-Condition	The inteface is ready
Post-Condition	User authenticated
Description of the main scenario	<p>The system displays the authentication interface.</p> <p>The user scrolls the interface and chose a card.</p> <p>The user clicks the "card".</p> <p>The user can get the information of the card.</p> <p>The user can download any card</p>
Exception scenario	

Table 2.7: refinement of the use case 'Consult props'

2.4 Sprint 0 Design

Design is a significant activity for understanding the development of a system in order to make it more reliable and faithful to the client's needs. Therefore, we will specify our system before implementing it.

Design of Sprint 0

Design is a very necessary phase to better understand the development of a system in order to make this development more reliable and effective for the user's needs. In this stage, we will model our system before implementing it. We will examine the following use cases:

- Authenticate
- Register
- Consult props

2.4.1 "Authenticate" Use Case Design:

Class diagram: We will use the class diagram which allows us to present the static structure of the application by schematizing the interfaces, controllers, and entities in a simplified way in order to make our class diagram more readable

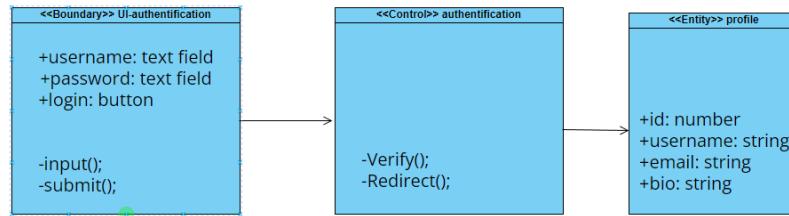


Figure 2.5: Authenticate class diagram

Sequence Diagram: Sequence diagrams are the most commonly used UML diagrams and are used to illustrate the use cases described in the previous chapter. They allow the chronological succession of operations performed by an actor to be represented, which move from one object to another to represent scenarios.

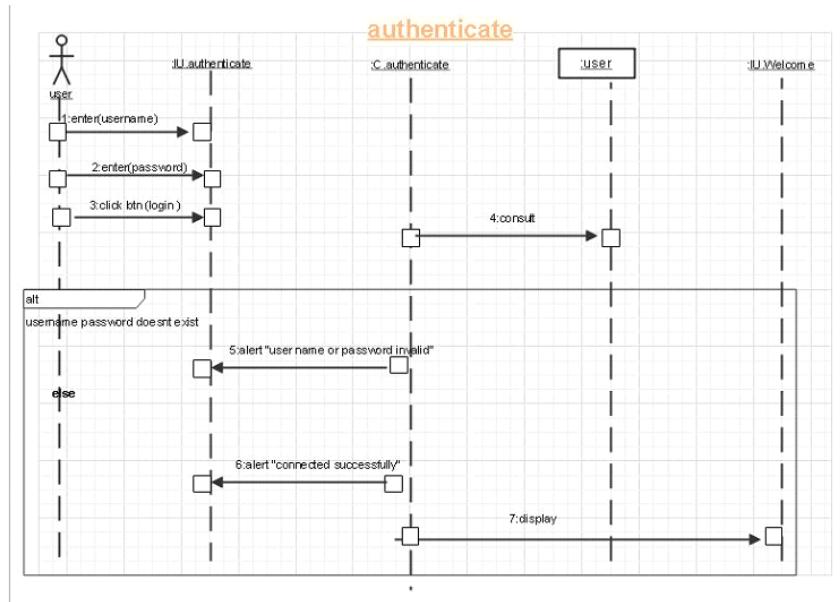


Figure 2.6: Authenticate sequence diagram

2.4.2 "Register" Use Case Design:

Class diagram:

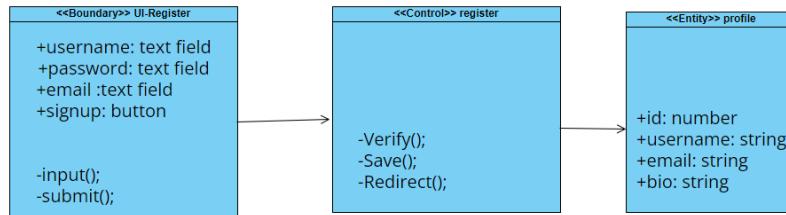


Figure 2.7: Register class diagram

Sequence Diagram:

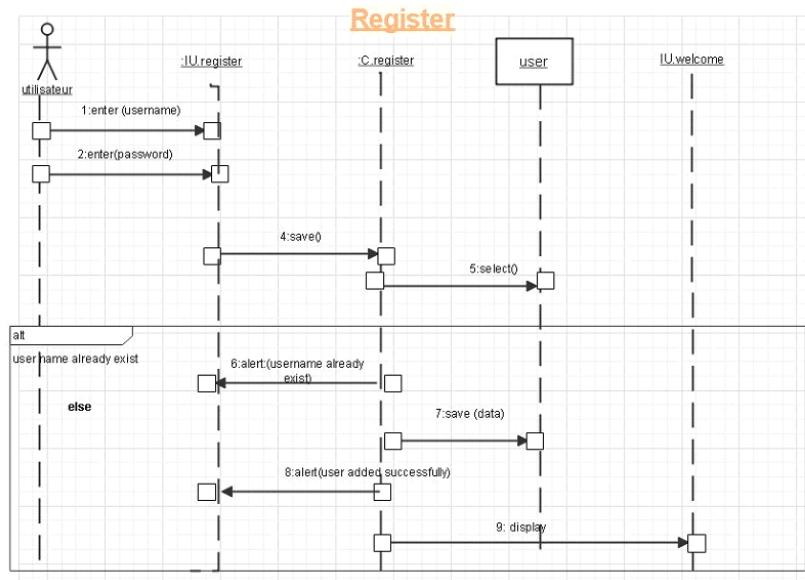


Figure 2.8: Register sequence diagram

2.4.3 "consult props" Use Case Design:

Class diagram:

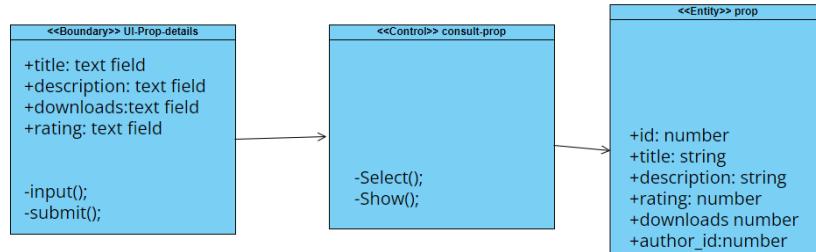


Figure 2.9: Consult props class diagram

Sequence Diagram:

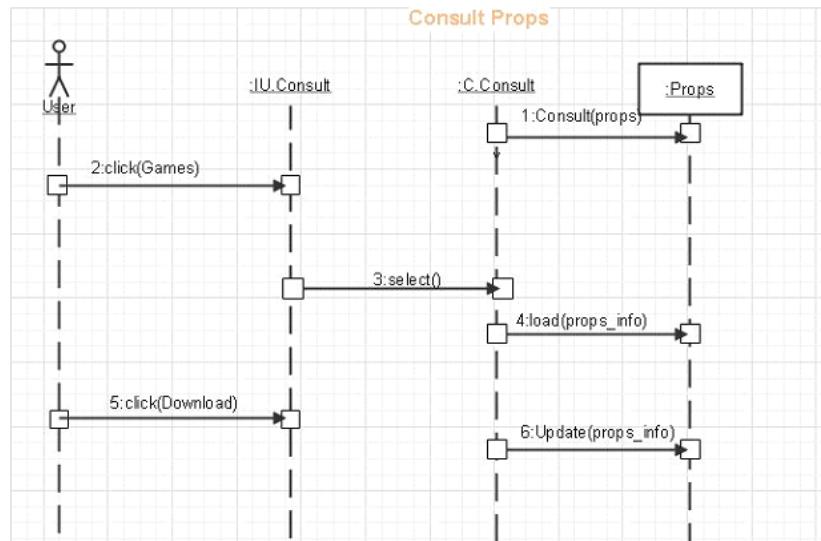


Figure 2.10: Authenticate sequence diagram

2.5 Implementation of Sprint 0:

2.5.1 Authenticate

This interface allows the user to enter their email and password in order to access their own account.

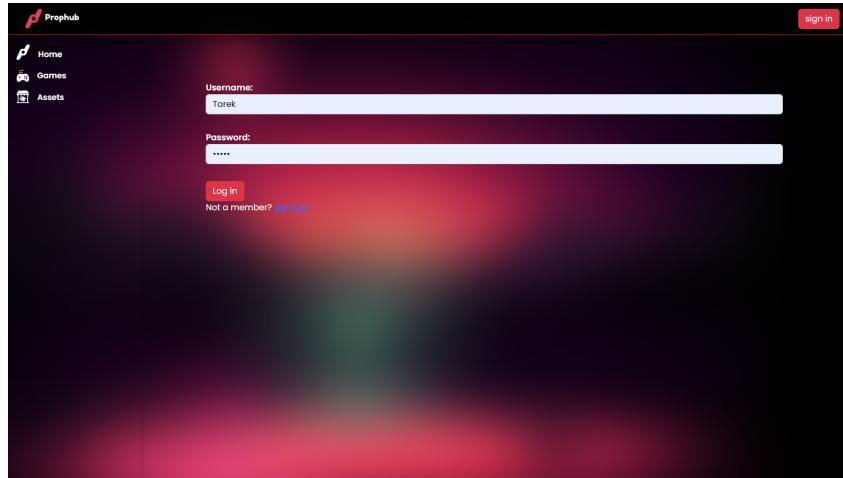


Figure 2.11: Authenticate screenshot

2.5.2 Register

The user enters their username, email, and password.

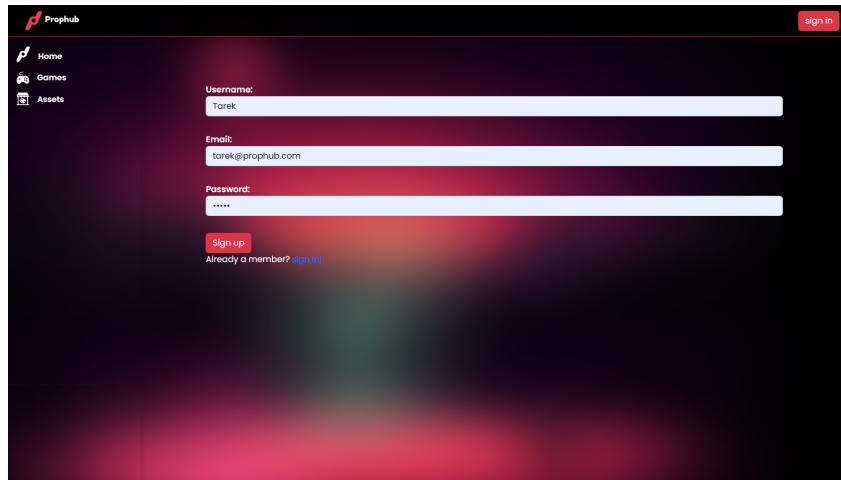


Figure 2.12: Register screenshot

2.5.3 Consult props

The user chose any card on consult props page.

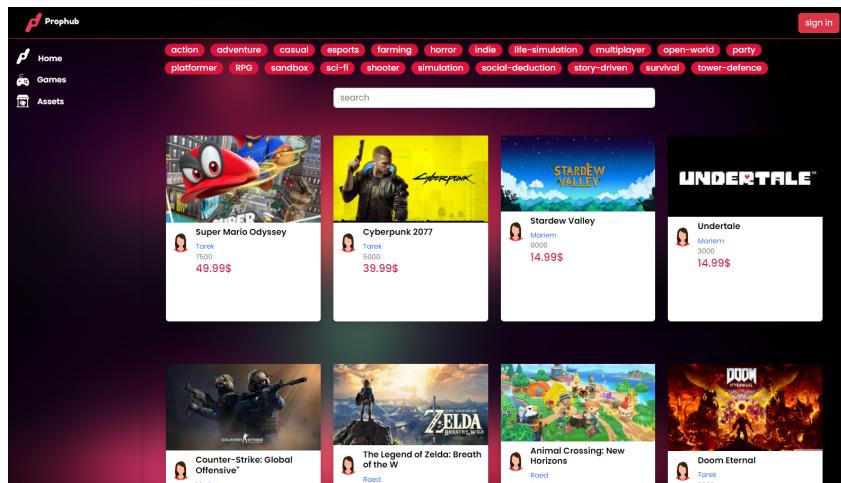


Figure 2.13: Consult props screenshot

2.6 Conclusion:

Through this chapter, we have made a detailed design of the authenticate ,register and consult props use cases.

Chapter 3

DESIGN OF SPRINT 1

In this stage, we will model our system before implementing it. We will examine the following use cases:

- Manage account
- Manage props
- Approve props

3.1 Introduction

In this chapter, we will present the first release of the project which is 'sprint 1' that details the priority 2 use cases. The study of this sprint includes refinement, design, and implementation.

3.2 Identification of the backlog for release 1

The backlog for the first release that we present below contains a list of backlog items that need to be completed in sprint 1:

Product backlog	Priority	Estimation	Planning
As a user , i can manage props	2	average	Sprint 1
As a user i can manage an account	2	average	Sprint 1
As an admin i can approve props	2	average	Sprint 1

Table 3.8: Product backlog for realease 1

3.3 Refinement of sprint 1

In this section, we focus on the following use cases:

- Manage account
- Manage props
- Approve props

3.3.1 Refinement of the use case 'manage account':

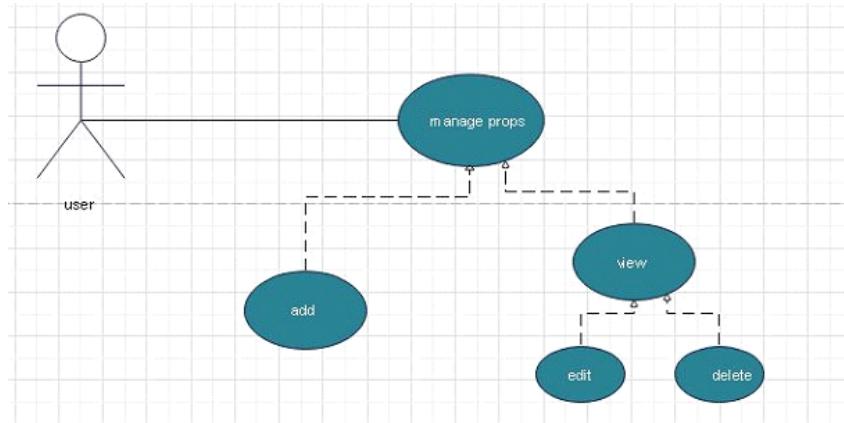


Figure 3.14: Refinement of the use case Manage account

Use case	Manage account
Actor	User
Pre-Condition	Existed account
Post-Condition	The profile has been managed by the user
Description of the main scenario	The system displays the setting profile page. The user can update their profile picture, first name, last name, username, phone number, email address, and password..
Exception scenario	The system displays an error message if there are issues with the data.
Scenarios	Add View Edit Delete

Table 3.9: refinement of the use case 'Manage account'

3.3.2 Refinement of the use case 'manage props':

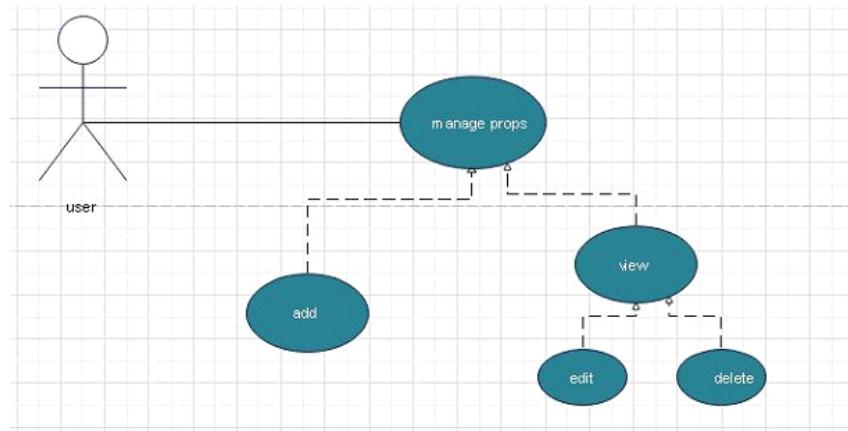


Figure 3.15: Refinement of the use case Manage props

Use case	Manage props
Actor	User
Pre-Condition	User loged in
Post-Condition	The profile has been managed by the user
Description of the main scenario	The system displays the props interface . The user can add a fan art, discription, tags, and categories The user can user clicks on upload props.
Exception scenario	The system displays an error message if there are issues with the data.
Scenarios	Add View Edit Delete

Table 3.10: refinement of the use case 'Manage props'

3.3.3 Refinement of the use case 'approve props':

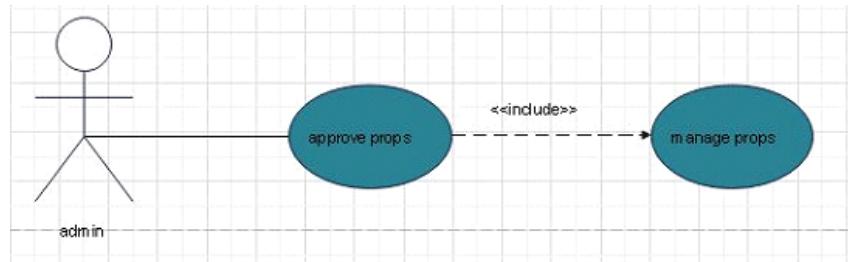


Figure 3.16: Refinement of the use case Approve props

Use case	Approve props
Actor	admin
Pre-Condition	
Post-Condition	The profile has been managed by the user
Description of the main scenario	The system displays the props interface . The user can add a fan art, description, tags, and categories The user can user clicks on upload props.
Exception scenario	The system displays an error message if there are issues with the data.
Scenarios	Add View Edit Delete

Table 3.11: refinement of the use case 'Approve props'

3.4 Design Release 1

3.4.1 Reslease 1 concept

In this stage, we will model our system before implementing it. We will examine the following use cases:

- Manage props
- Manage profile
- Approve props

3.4.2 manage props Use Case Design:

Class diagram Sequence diagram

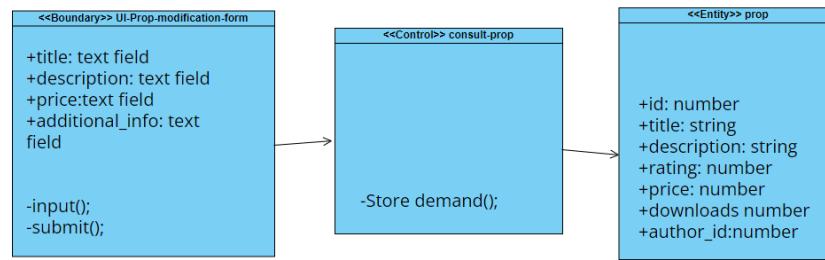


Figure 3.17: manage props class diagram

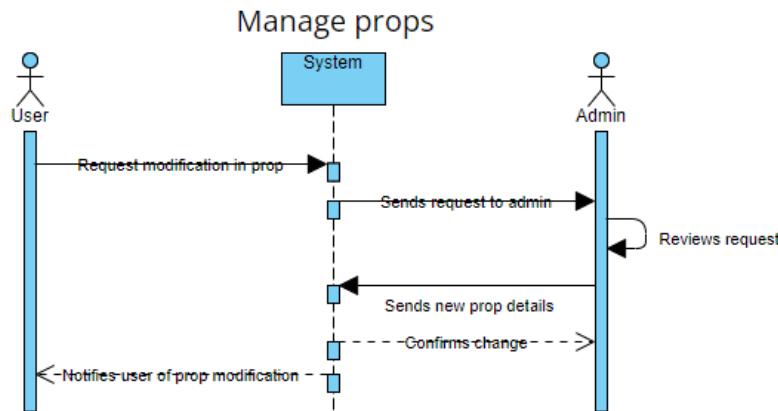


Figure 3.18: manage props sequence diagram

3.4.3 manage profile Use Case Design:

Class diagram

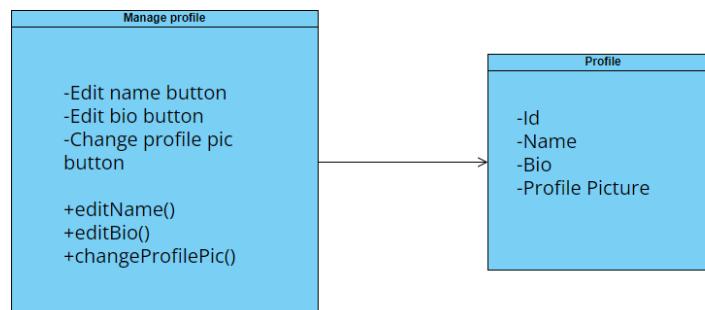


Figure 3.19: manage profile class diagram

Sequence diagram

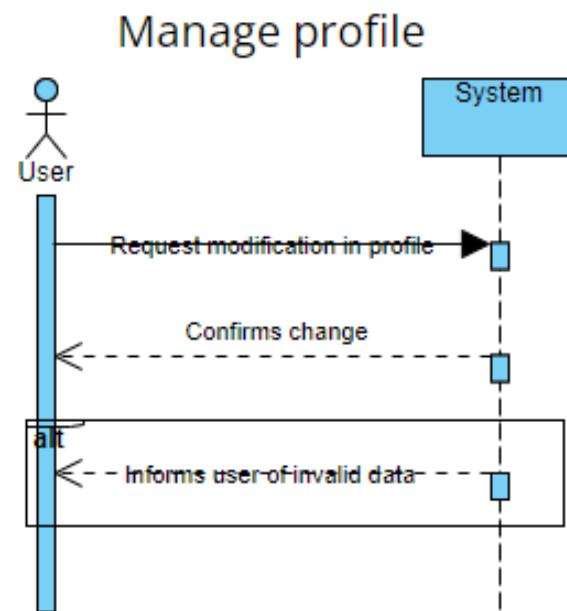


Figure 3.20: manage profile sequence diagram

3.4.4 approve props Use Case Design:

Class diagram

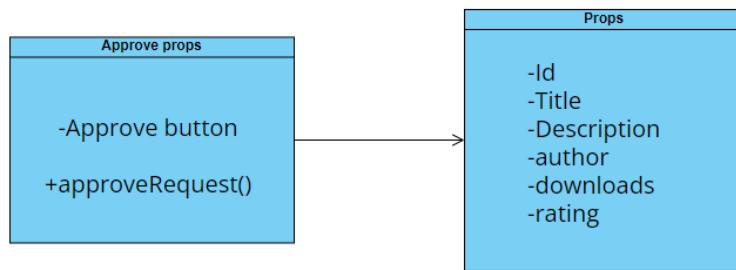


Figure 3.21: approve props class diagram

Sequence diagram

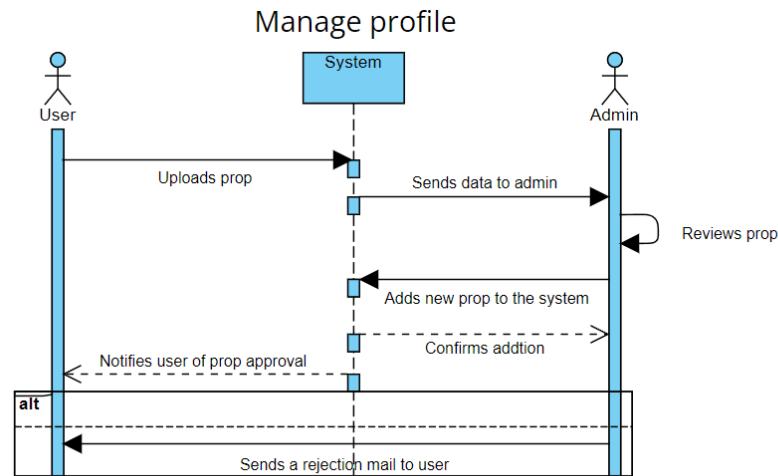


Figure 3.22: approve props sequence diagram

3.5 Implementation of Sprint 1

3.5.1 Manage props

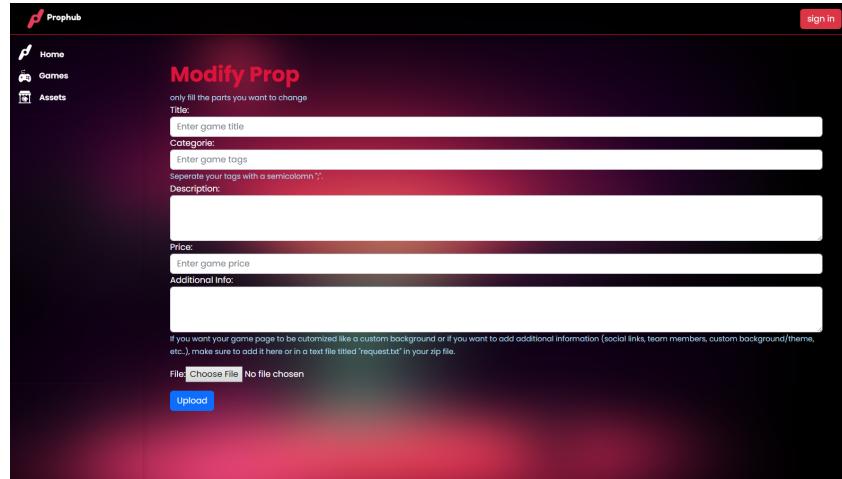


Figure 3.23: manage props screenshot

3.5.2 Manage profile

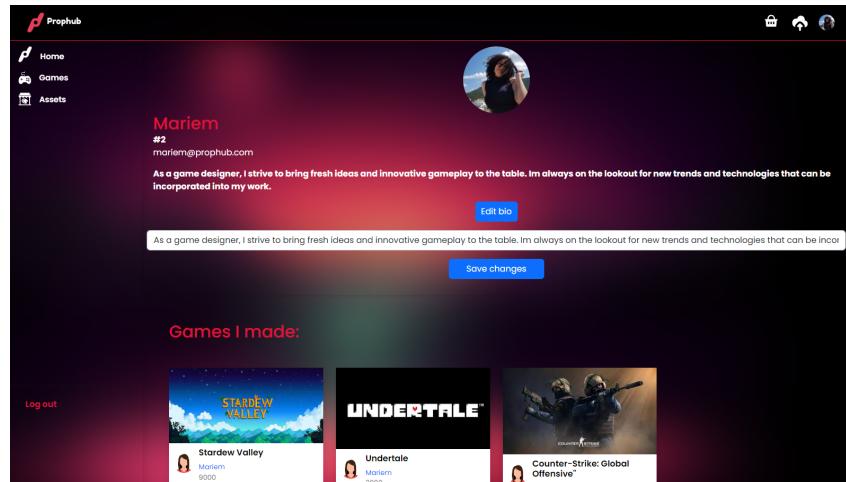


Figure 3.24: manage profile screenshot

3.6 Conclusion:

In this chapter, we have completed the use cases for sprint 1, and then we have presented some interfaces along with their analysis.

General conclusion

Upon concluding this report, it can be inferred that this project has provided us with a valuable opportunity to apply our theoretical knowledge in a practical environment. Through taking on various responsibilities, we were able to further solidify our understanding of both practical and theoretical aspects. Additionally, the frequent troubleshooting of errors during the project also significantly contributed to our overall learning experience.