Tunisian Republic

Ministry of Higher Education and Scientific Research

Higher Institute of Computer Science of Mahdia

Graduation Project Order Number: …….

FINAL PROJECT FOR OBTAINING A LICENSE IN Computer Science

SPECIALTY: «Software and Information Systems Engineering »

Presented by:

Wissem Ncir

DESIGN AND DEVELOPMENT OF

UNIVERSITY MANAGEMENT SYSTEM

Defended on ……………., in the front of the PFE exam jury:

|  |  |  |
| --- | --- | --- |
|  | ................... | President |
|  | ................... | Supervisor |
|  | ................... | Examiner |

Academic year 2022/2023

Dedication & Special Thanks

I’m ecstatic to dedicate this work only to “my mother”,

for her continuous fighting, struggle, support, patient, and encouragement during my entire formation.

First of all, we would like to thank the entire teaching team of the Higher Institute of Computer Science of Mahdia specially “the administration” and “the pfe exam jury”.

Before starting this report, I would like to take this opportunity to thank my professor Mem “Chebil Wiem” who encourage me during the duration of the project. I would also like to thank her for the help and advice concerning the tasks mentioned in this report.

I would like to thank all my professors for encouraging me to work by making their experiences and skills available to us.

**Wissem Ncir**

Table of contents

[Chapter1: Project Framework 12](#_Toc145389839)

[I. Presentation of the project: 12](#_Toc145389840)

[II. Study of the existing: 12](#_Toc145389841)

[A. ISIMA: 12](#_Toc145389842)

[B. Weak point: 13](#_Toc145389843)

[C. Proposed solution: 13](#_Toc145389844)

[III. The working methodology: 13](#_Toc145389845)

[A. Agile methods: 13](#_Toc145389846)

[B. The Scrum Method: 14](#_Toc145389847)

[1) Why Scrum Framework? 14](#_Toc145389848)

[2) Scrum Roles: 15](#_Toc145389849)

[3) Scrum Artifacts: 15](#_Toc145389850)

[4) Scrum Events: 16](#_Toc145389851)

[C. The modeling language: 17](#_Toc145389852)

[D. Work Breakdown Chart: 18](#_Toc145389854)

[Chapter2: Planning, analysis and](#_Toc145389855) [specification of needs 20](#_Toc145389856)

[I. Architectural design: 20](#_Toc145389857)

[A. Architecture physique: 20](#_Toc145389858)

[1) Presentation tier 21](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145389860)

[2) Logic tier 21](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145389861)

[3) Data tier 21](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145389862)

[B. Logic architecture: 22](#_Toc145389863)

[C. Work environment: 23](#_Toc145389864)

[1) Hardware environment: 23](#_Toc145389865)

[2) Software environment: 24](#_Toc145389866)

[II. Needs planification: 27](#_Toc145389867)

[A. Sprints Planification: 27](#_Toc145389868)

[B. System parts fragmentation using WBS (Work Breakdown Chart): 29](#_Toc145389869)

[C. Gantt Diagram 30](#_Toc145389870)

[III. Needs Analysis and Specification: 31](#_Toc145389871)

[A. Product Backlog: 31](#_Toc145389872)

[B. Identification of the SCRUM team: 33](#_Toc145389873)

[C. Identification of actors: 33](#_Toc145389874)

[D. Functional needs: 33](#_Toc145389875)

[1) Functional needs of the “Login Manager” actor: 33](#_Toc145389876)

[2) Functional needs of the “Administrator” actor: 34](#_Toc145389877)

[3) Functional needs of the “Professor” actor: 34](#_Toc145389878)

[4) Functional needs of the “Library Member” actor: 35](#_Toc145389879)

[E. Non-functional needs: 35](#_Toc145389880)

[IV. Global Diagrams: 36](#_Toc145389881)

[A. Global use cases diagram: 37](#_Toc145389882)

[B. Global class Diagrams: 38](#_Toc145389883)

[1) Management system global class Diagram: 38](#_Toc145389884)

[2) Library management system global class Diagram: 40](#_Toc145389885)

[Chapter3: Sprint 1](#_Toc145389887) [and Sprint 2 42](#_Toc145389888)

[I. The first sprint: 42](#_Toc145389889)

[A. First Sprint Backlog: 42](#_Toc145389890)

[B. Sprint 1 use case diagram: 44](#_Toc145389891)

[C. Uses cases Analysis: 45](#_Toc145389892)

[1) “Manage access to the main system” use Analysis case: 45](#_Toc145389893)

[a) Textual description: 45](#_Toc145389894)

[b) Sequence diagram of “Manage Access to the system” use case: 46](#_Toc145389895)

[c) Activity diagram of “Manage Access to the system” use case: 47](#_Toc145389896)

[2) “Authenticates” use case Analysis: 47](#_Toc145389897)

[a) Textual description: 47](#_Toc145389898)

[b) Sequence diagram of “Authenticates“ use case: 49](#_Toc145389899)

[c) Activity diagram of “Authenticates” use case: 50](#_Toc145389900)

[d) Realization: 50](#_Toc145389901)

[e) Test: 52](#_Toc145389902)

[3) “University Details consultation” Uses cases Analysis: 53](#_Toc145389903)

[a) Textual description: 53](#_Toc145389904)

[b) sequence diagram of “University Details consultation” use case: 54](#_Toc145389905)

[c) Activity diagram of “University Details consultation” use case: 55](#_Toc145389906)

[d) Realisation: 56](#_Toc145389907)

[e) Test: 57](#_Toc145389908)

[II. The second sprint: 58](#_Toc145389909)

[A. Second Sprint Backlog: 58](#_Toc145389910)

[B. Sprint 2 use case diagram: 59](#_Toc145389911)

[C. Uses cases Analysis: 60](#_Toc145389912)

[1) “Student management” uses cases Analysis: 60](#_Toc145389913)

[a) Textual description 60](#_Toc145389914)

[b) sequence diagram of “Student Details management” use case: 61](#_Toc145389915)

[c) Activity diagram of “Student Details management” use case: 62](#_Toc145389916)

[d) Realisation: 63](#_Toc145389917)

[e) Test: 63](#_Toc145389918)

[2) “Staff management” uses cases analysis: 64](#_Toc145389919)

[a) Textual description: 64](#_Toc145389920)

[b) sequence diagram of “Staff Details management” use case: 65](#_Toc145389921)

[c) Activity diagram of “Staff Details management” use case: 66](#_Toc145389922)

[d) Realisation: 67](#_Toc145389923)

[e) Test: 67](#_Toc145389924)

[3) University resources uses cases Analysis: 68](#_Toc145389925)

[a) Textual description: 68](#_Toc145389926)

[b) sequence diagram of “University Resources Details management” use case: 69](#_Toc145389927)

[c) Activity diagram of “University Resources Details management” use case: 70](#_Toc145389928)

[d) Realisation: 71](#_Toc145389929)

[e) Test: 71](#_Toc145389930)

[Chapter 4: Sprint 3 74](#_Toc145389933)

[I. The third sprint: 74](#_Toc145389934)

[A. Third Sprint Backlog: 74](#_Toc145389935)

[B. Sprint 3 use case Diagram: 75](#_Toc145389936)

[C. Uses cases Analysis: 78](#_Toc145389937)

[1) “Students grades management” uses cases Analysis: 78](#_Toc145389938)

[a) Textual description: 78](#_Toc145389939)

[b) Sequence diagram of “students grades management” use case: 79](#_Toc145389940)

[c) Activity diagram of “Students grades management” use case: 81](#_Toc145389941)

[d) Realisation : 82](#_Toc145389942)

[e) Test : 83](#_Toc145389943)

[2) “Timetables management” uses cases Analysis: 83](#_Toc145389944)

[a) Textual description: 83](#_Toc145389945)

[b) Sequence diagram of “Timetables management” use case: 84](#_Toc145389946)

[c) Activity diagram of “Timetables management” use case: 85](#_Toc145389947)

[d) Realisation: 86](#_Toc145389948)

[e) Test: 86](#_Toc145389949)

[3) “Library management” uses cases Analysis: 87](#_Toc145389950)

[a) Textual description: 87](#_Toc145389951)

[b) Sequence diagram of “Library management” use case: 88](#_Toc145389952)

[c) Activity diagram of “Library management” use case: 89](#_Toc145389953)

[d) Realisation: 91](#_Toc145389954)

Tables List

[Table 1: Hardware environment 23](#_Toc145390175)

[Table 2: Software environment 24](#_Toc145390176)

[Table 3: Sprint planification 28](#_Toc145390177)

[Table 4: Product Backlog 31](#_Toc145390178)

[Table 5: Our Scrum team members 33](#_Toc145390179)

[Table 6: First Sprint Backlog 42](#_Toc145390180)

[Table 7: Textual description of use case “Authenticates” 45](#_Toc145390181)

[Table 8: Textual description of authenticates use case 47](#_Toc145390182)

[Table 9:Textual description of use case “University Details consultation” 53](#_Toc145390183)

[Table 10: Sprint2 Backlog 58](#_Toc145390184)

[Table 11: Textual description of use case “Student management” 60](#_Toc145390185)

[Table 12: Textual description of use case “Staff management” 64](#_Toc145390186)

[Table 13: Textual description of use case “University Resources Details management” 68](#_Toc145390187)

[Table 14: Third Sprint Backlog 74](#_Toc145390188)

[Table 15: Textual description of “Students grades management” use case 78](#_Toc145390189)

[Table 16: Textual description of “Timetables management” use case 83](#_Toc145390190)

[Table 17: Textual description of “Library management” use case 87](#_Toc145390191)

Figures List

[Figure 1: isima Logo [1] 12](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390268)

[Figure 2: Agile methodology [2] 14](#_Toc145390269)

[Figure 3: Scrum Team [5] 15](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390270)

[Figure 4: Scrum Artifacts [7] 16](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390271)

[Figure 5: Scrum events [9] 17](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390272)

[Figure 6: UML Logo [11] 18](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390273)

[Figure 7: Three-tier architecture [14] 21](#_Toc145390274)

[Figure 8: MVC (Model View Controller) [15] 22](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390275)

[Figure 9: Anaconda Logo 24](#_Toc145390276)

[Figure 10: Visual Studio Logo 24](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390277)

[Figure 11: Python Logo 25](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390278)

[Figure 12: PHP Logo 25](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390279)

[Figure 13: XAMPP Logo 25](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390280)

[Figure 14: Diagrams.net Logo 26](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390281)

[Figure 15: MySQL 26](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390282)

[Figure 16: GitHub 27](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390283)

[Figure 17: WBS (Work Breakdown Chart) 29](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390284)

[Figure 18: Gantt Diagram 30](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390285)

[Figure 19: Global use cases Diagram 37](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390286)

[Figure 20: Management system global class Diagram 39](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390287)

[Figure 21: Library management system global class Diagram 40](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390288)

[Figure 22: Sprint1 use cases Diagram 44](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390289)

[Figure 23: Manage Access to the system sequence Diagram 46](#_Toc145390290)

[Figure 24: Manage Access to the system activity Diagram 47](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390291)

[Figure 25: Sequence diagram “Authenticate“ 49](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390292)

[Figure 26: Activity Diagram “Authenticate” 50](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390293)

[Figure 27: Admin Login interface 50](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390294)

[Figure 28: user Login interface 51](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390295)

[Figure 29: Password reset using Ai Artificial Recognition interface 51](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390296)

[Figure 30: user is logged test 52](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390297)

[Figure 31: Admin is logged test 52](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390298)

[Figure 32: face\_photo matching test 53](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390299)

[Figure 33: Sequence diagram “University Details consultation” 54](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390300)

[Figure 34: Activity Diagram “University Details consultation” 55](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390301)

[Figure 35: Main System Dashboard 56](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390302)

[Figure 36: University Details section 56](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390303)

[Figure 37: Consulting About University section test 57](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390304)

[Figure 38: Consulting University internal system test 57](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390305)

[Figure 39: Sprint 2 use case Diagram 59](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390306)

[Figure 40: Sequence Diagram “Student Details management” 61](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390307)

[Figure 41: Activity diagram “Student Details management” 62](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390308)

[Figure 42: Student Management interface 63](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390309)

[Figure 43: Student Management section test 63](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390310)

[Figure 44: Sequence diagram “Staff Details management” 65](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390311)

[Figure 45: Activity diagram “Staff Details management” 66](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390312)

[Figure 46: Staff Management interface 67](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390313)

[Figure 47: Staff Management section test 67](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390314)

[Figure 48: Sequence Diagram “University Resources Details management” 69](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390315)

[Figure 49: Activity diagram “University Resources Details management” 70](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390316)

[Figure 50: University Resources Management interface 71](#_Toc145390317)

[Figure 51: University Resources Management section test 71](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390318)

[Figure 52: Sprint 3 use case Diagram 77](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390319)

[Figure 53: Sequence diagram of “students grades management” 80](#_Toc145390320)

[Figure 54: Activity Diagram 81](#_Toc145390321)

[Figure 55: Grades section Login interface 82](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390322)

[Figure 56: Student Grades node system interface 82](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390323)

[Figure 57: Grades management interface 82](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390324)

[Figure 58: Grades Admin Section test 83](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390325)

[Figure 59: Activity Diagram “Timetables management” 85](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390326)

[Figure 60: Timetables management interface 86](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390327)

[Figure 61: Timetables interface test 86](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390328)

[Figure 62: sequence diagram “Library management” 89](#_Toc145390329)

[Figure 63: Activity diagram “Library management” 90](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390330)

[Figure 64: Member and Librarian Login interface 91](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390331)

[Figure 65: Library member home page 91](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390332)

[Figure 66: Librarian home page 92](file:///C:\Users\Administrateur\OneDrive\Desktop\Reaport%20(final%20version)%20copy1.docx#_Toc145390333)

General introduction

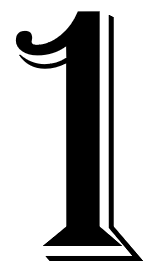
A university management system is an essential tool for effective and efficient management of administrative activities in a university. It provides a centralized solution for managing and tracking various processes such as students and staff details management, university resources management, library management, and much more.

The importance of a University Management System lies in its ability to optimize processes, improve efficiency, enhance student experience, reduce costs, and facilitate informed decision-making. By providing organized, classified and centralized all-in-one data, automating repetitive tasks, eliminating human errors, and providing real-time data, this system can help universities better manage their administrative tasks and improve the quality of their academic programs. and to provide students with a better academic experience.

That's why we chose our project to be “University Management System”.



# Project Framework



# Chapter1: Project Framework

Introduction:

In this chapter we will do a presentation of our project, preceded by an existence study which is the basis of the proposed solution.

## Presentation of the project:

My end-of-study project was carried out within the institute ISIMA, which helped me in the design and development of a university management system. For this system we used Advanced Python TKinter Framework for development. The system aims to simplify and automate the administrative processes of a university environment. The goal is to improve the experience for all administrators, professors and students.

## Study of the existing:

The study of the existing is the main part of the analysis of a project. This is a very important step in the project to identify the system weak points. It contains 2 parts, first identify the global weak point, then set

the proposed solution.

### ISIMA:

Figure 1: isima Logo [1]

ISIMA is a public university for both citizens students and from outside Tunisia. That offers multi diplomas, like license and master. Also, too many Erasmus+ certifications in various domains like programming machine learning, networking, cybersecurity ...

### Weak point:

The administration university system of ISIMA is lacking a feature that allows professors to add students grades remotely, which could save the administration staff too much time, resources, and reduce human mistakes.

### Proposed solution:

In our university management system, we will introduce a new feature that will allow professors to add student grades remotely and directly into the university system.

## The working methodology:

### Agile methods:

Agile is an iterative and incremental approach for project management and software development that emphasizes flexibility, collaboration, and customer satisfaction. It involves breaking down large projects into smaller, more manageable pieces, and working on them in short, iterative cycles called sprints. The agile approach emphasizes frequent communication between team members, including the customer or end-user, to ensure that the project is meeting their needs and requirements. It also emphasizes adaptability and the ability to quickly respond to changes in the project or marketplace, rather than following a rigid plan. Agile methodologies include Scrum, Kanban, and Lean, and many others.

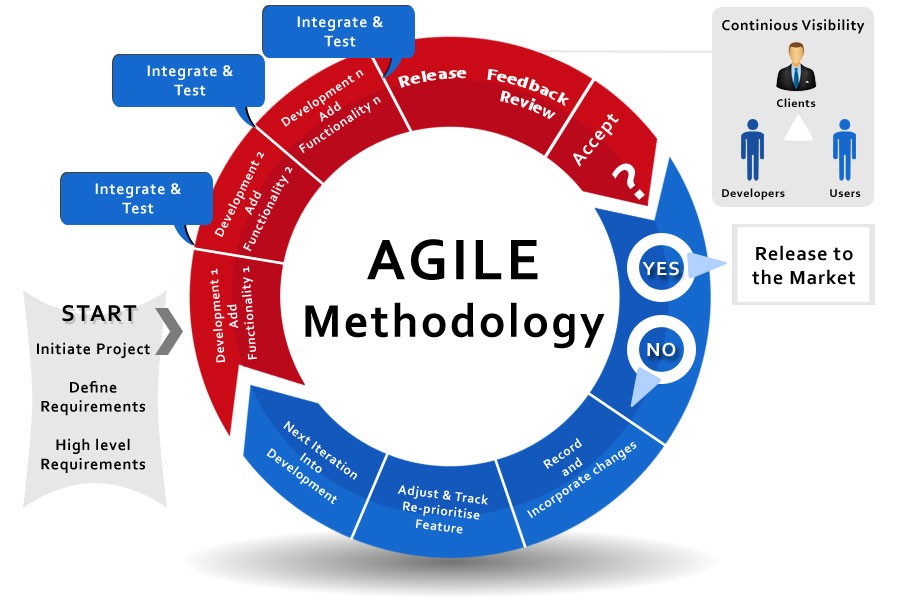


Figure 2: Agile methodology [2]

In this project we will use the most popular Agile methodology which is Scrum.

### The Scrum Method:

In this presentation, we will explore Scrum, an Agile Framework. This exploration will include Scrum employs specific roles, artifacts, events to enhance teamwork, and Burndown Chart.

#### Why Scrum Framework?

-Ensure Employs iterative and incremental development.

-Enhances teamwork, communication, and adaptability.

-Facilitates quick delivery of high-quality products. [3]

#### Scrum Roles:

-Scrum Master: Ensures adherence to Scrum practices and removes obstacles.

-Product Owner: Manages product backlog and prioritizes features.

-Development Team: Self-organizes to deliver working increments. [4]

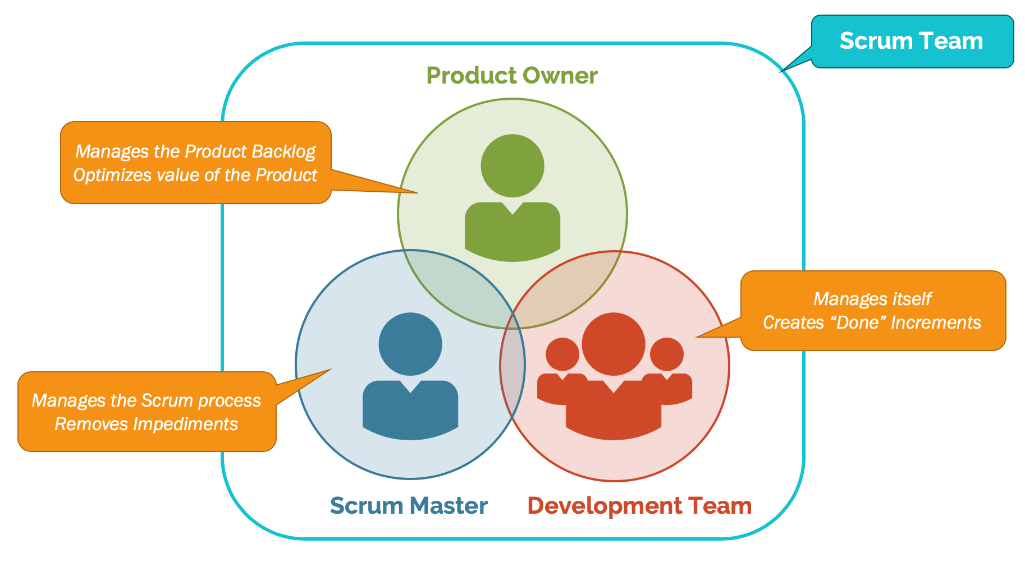


Figure 3: Scrum Team [5]

#### Scrum Artifacts:

-Product Backlog: Ordered list of features, enhancements, and fixes.

-Sprint Backlog: Selected items for the current sprint.

-Increment: Usable product increment resulting from a sprint. [6]

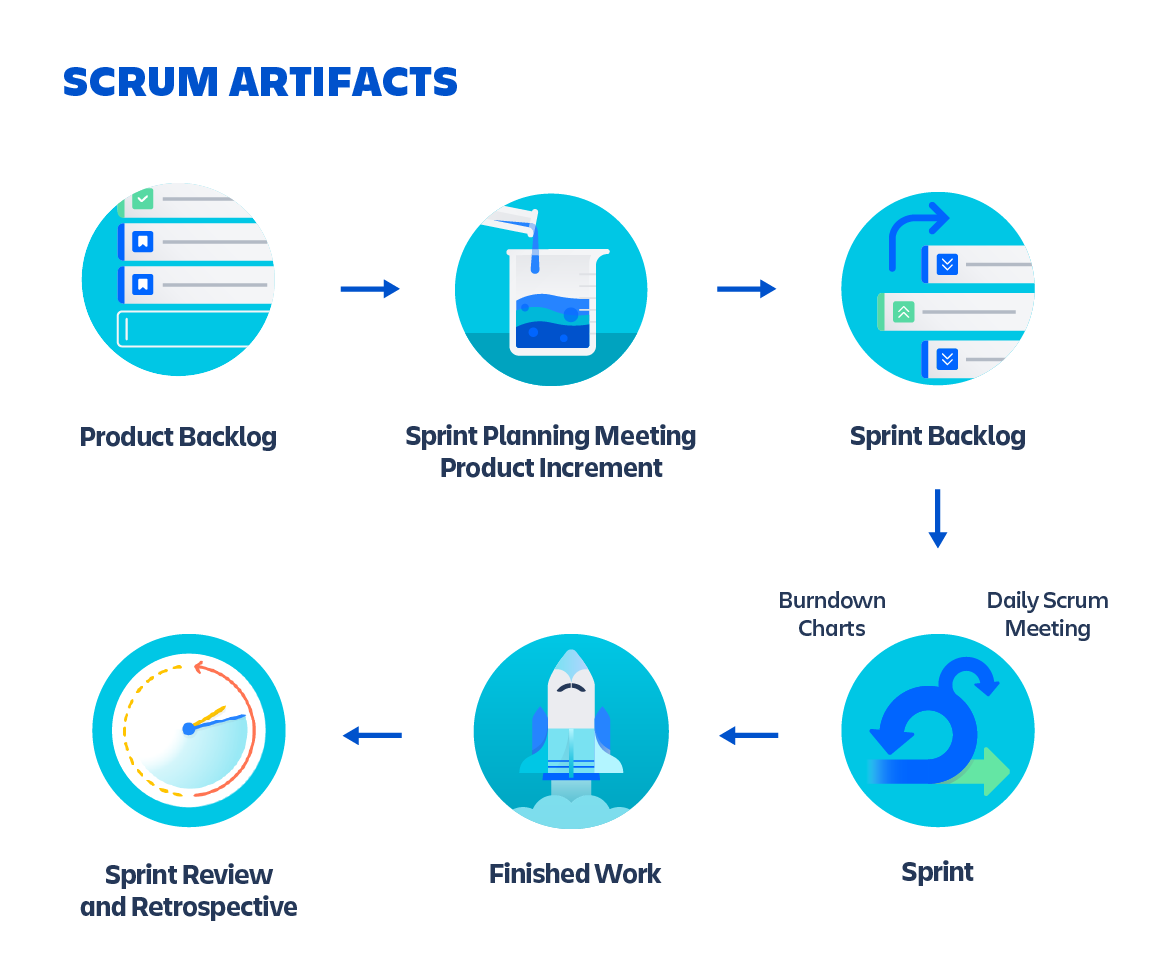


Figure 4: Scrum Artifacts [7]

#### Scrum Events:

+Sprint: (Time-boxed iteration, typically 2-4 weeks long)

● Daily Stand-up: Brief daily meeting to discuss progress and obstacles.

● Sprint Review: Demonstrates the increment to stakeholders.

● Sprint Retrospective: Reflects on the sprint to improve future performance.

+Burndown Chart: is a visual representation of the progress of a project over time. It is typically used in Agile project management methodologies, such as Scrum, to help the team monitor their progress and adjust their approach if needed. The chart shows the amount of work remaining to be done over the course of the project, with the amount of work decreasing as time goes on. This allows the team to identify any potential roadblocks or delays and take action to address them. The Burndown Chart is a useful tool for tracking progress and keeping the team on track towards meeting their goals. [8]

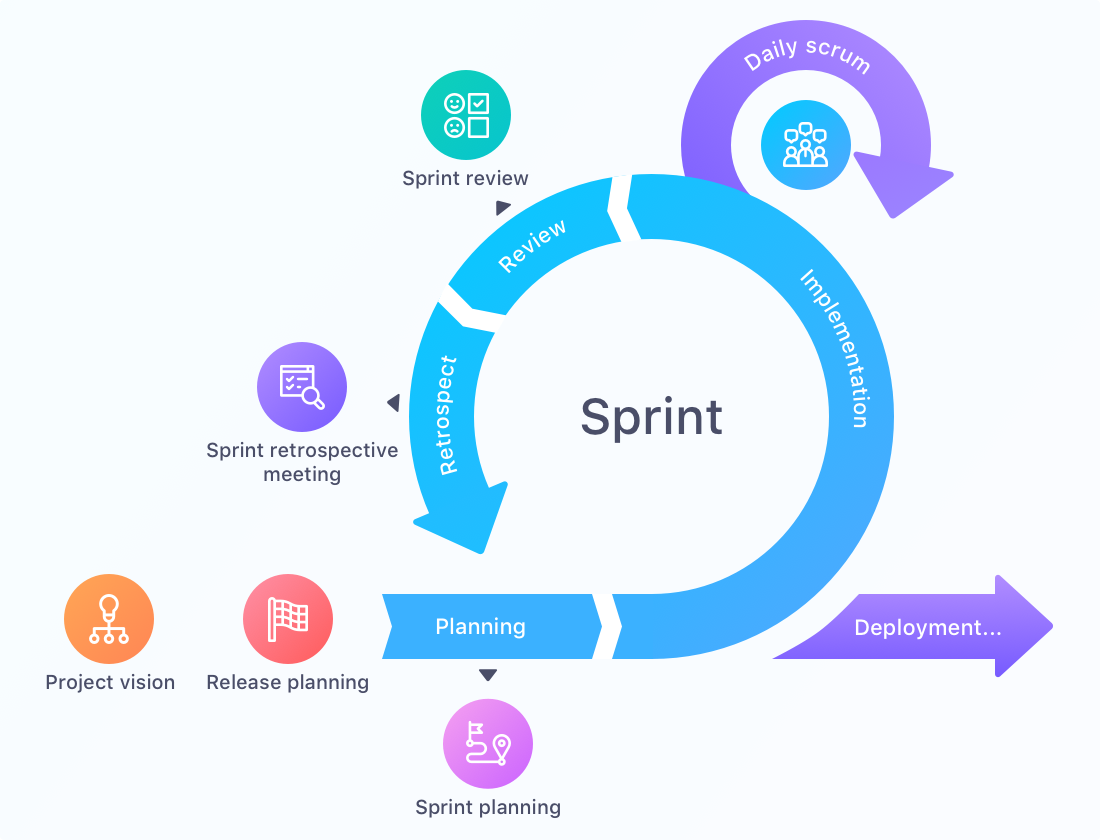


Figure 5: Scrum events [9]

### The modeling language:

The modeling language is a language used to represent and communicate different aspects of a system or process. It is a visual language that helps people to understand and analyze complex systems by representing them in a structured and organized way.

One popular modeling language is the Unified Modeling Language (UML), which is commonly used in software engineering to represent software designs and architectures.

It's a graphical language used for modeling software systems. It provides a standard set of notations and diagrams for representing the structure, behavior, and relationships of a software system. [10]

## 

## 

Figure 6: UML Logo [11]

### Work Breakdown Chart:

A Work Breakdown Chart (WBC) is a project management tool that breaks down a large project into smaller, more manageable components or tasks. It provides a visual representation of the work that needs to be done and helps to identify the scope, time, and cost of the project. The WBC is usually presented in a hierarchical format, with the main project broken down into smaller, more specific tasks. [12]

Conclusion:

In this chapter, we gave a general overview of this project by describing the host institution "Isima", and the context of the project. We also presented our work methodology to execute this project, such as the Scrum method. The rest of the report is organized according to this approach. In the next chapter we will discuss, planning, analysis and requirements specification.



Planning, analysis and specification of needs

# 

# Chapter2: Planning, analysis and

# specification of needs

Introduction

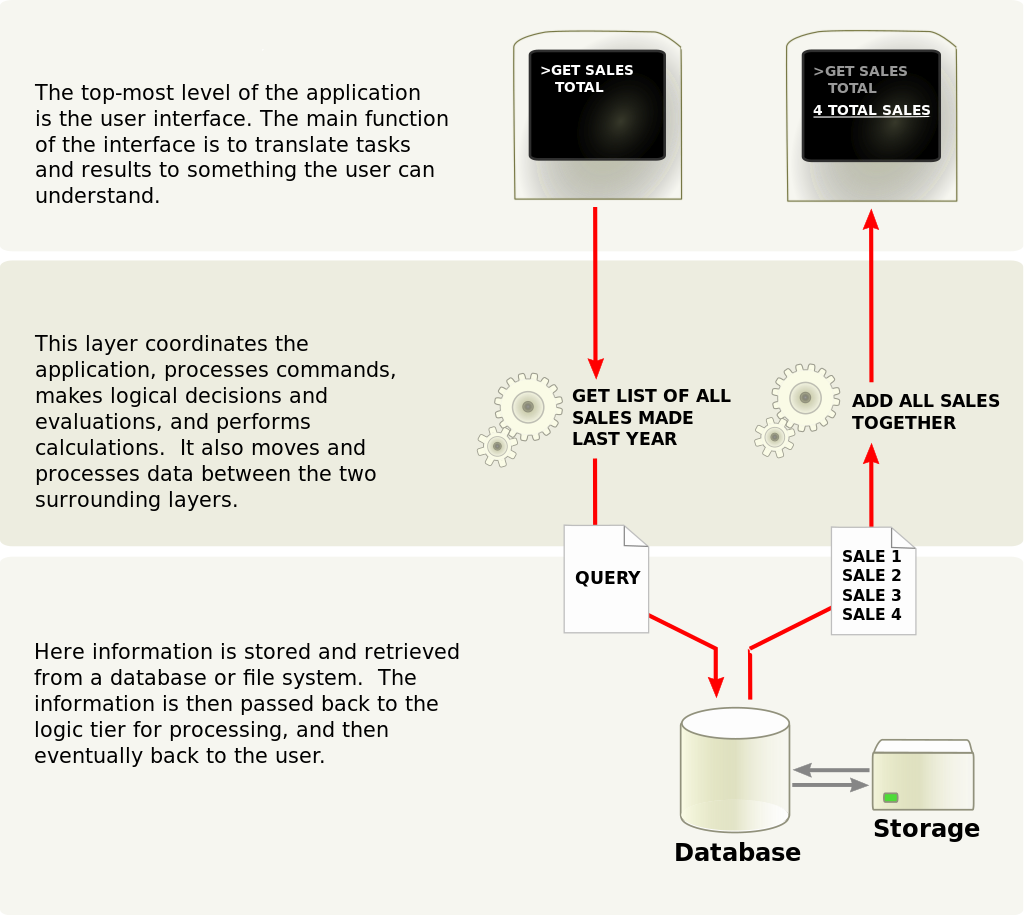
In this chapter we will introduce the used architecture and our work environment, then do sprints planification, product backlogs generation, needs specification and detailed conceptual study.

## Architectural design:

The architecture is the models and techniques used to generate and create softwares and applications. [13]

### Architecture physique:

Three-tier architecture is a client-server software architecture pattern in which the user interface (presentation), functional process logic ("business rules"), computer data storage and data access are developed and maintained as independent modules.



#### Presentation tier

#### Logic tier

#### Data tier

Figure 7: Three-tier architecture [14]

### Logic architecture:

MVC stands for model-view-controller. Here's what each of those components mean:

**Model**: The backend that contains all the data logic (OLAP Models, DataBases…)

**View**: The frontend or graphical user interface (GUI)

**Controller**: The brains of the application that controls how data is displayed

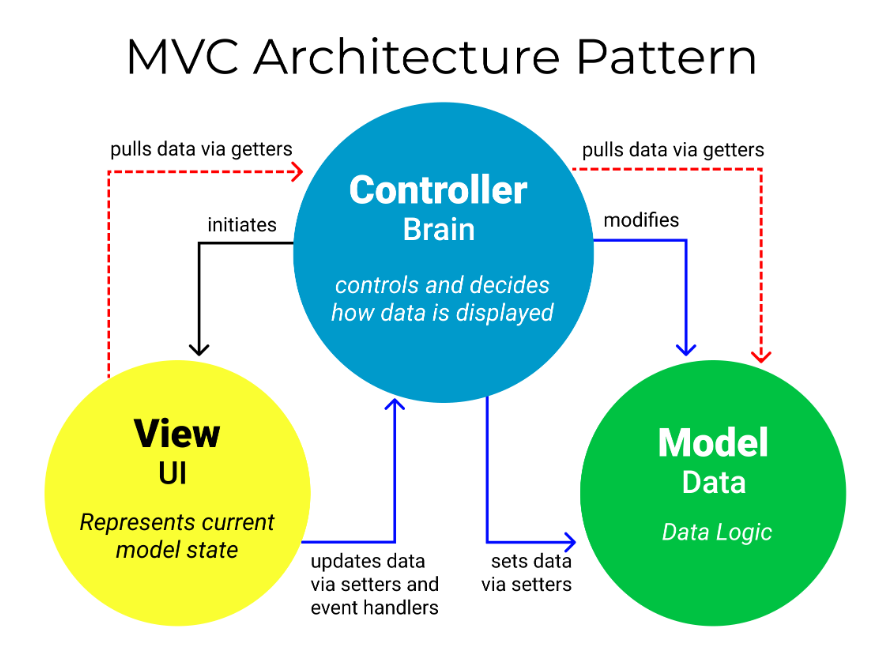


Figure 8: MVC (Model View Controller) [15]

### Work environment:

In this section, we will expose the work environment that we used.

#### Hardware environment:

Table : Hardware environment

|  |  |
| --- | --- |
| **Brand** | MSI |
| **Processor** | Intel(R) Core (TM) i5-10500H CPU  5 GHZ |
| **RAM** | 32 GB |
| **Hard Disk** | 512 GB SSD |
| **O.S** | Microsoft Windows 11 Pro |

#### Software environment:

Table : Software environment

|  |  |
| --- | --- |
| Logo | Description |
| Figure : Anaconda Logo | **Anaconda**: is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012. As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, neither of which are free. [16] |
| **Figure 10: Visual Studio Logo** | **Visual Studio**: is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs including websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code. [17] |
| **Figure 11: Python Logo** | **Python**: is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.  Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library. [18] |
| **Figure 12: PHP Logo** | **PHP**: is a general-purpose scripting language geared towards web development. It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1993 and released in 1995. The PHP reference implementation is now produced by the PHP Group. PHP was originally an abbreviation of Personal Home Page, but it now stands for the recursive initialism PHP: Hypertext Preprocessor. [19] |
| **Figure 13: XAMPP Logo** | **XAMPP**: is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible. [20] |
| **Figure 14: Diagrams.net Logo** | **Diagrams.net**: diagrams.net (previously draw.io) is a cross-platform graph drawing software developed in HTML5 and JavaScript. Its interface can be used to create diagrams such as flowcharts, wireframes, UML diagrams, organizational charts, and network diagrams. Parts of its source code are provided under the Apache 2 open-source license.  diagrams.net is available free of charge as an online web app, and as an offline desktop application for Linux, macOS, and Windows. Its offline application is built using the Electron framework. The web app does not require online login or registration and can open from and save to the local hard drive. Supported storage and export formats to download include PNG, JPEG, SVG, and PDF. [21] |
| **Figure 15: MySQL** | **MySQL**: is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter My, and "SQL", the acronym for Structured Query Language. A relational database organizes data into one or more data tables in which data may be related to each other; these relations help structure the data. SQL is a language that programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups. [22] |
| github.png  **Figure 16: GitHub** | **GitHub**: is a platform and cloud-based service for software development and version control using Git, allowing developers to store and manage their code. It provides the distributed version control of Git plus access control, bug tracking, software feature requests, task management, continuous integration, and wikis for every project. Headquartered in California, it has been a subsidiary of Microsoft since 2018. [23] |

## Needs planification:

This section is about, system parts fragmentation using Work Breakdown Chart, sprints planification and Gantt diagram generation.

### **Sprints Planification:**

At this level we are about to prepare the work plan, we have already chosen to develop 3 sprints.

Table : Sprint planification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Sprint | Description | Start date | End date |
| 1 | Sprint 1 | -Access management to the system.  -Authentication.  -University Details  Consultation. | 05 February | 01March |
| 2 | Sprint 2 | -Students management.  -Staff management.  -University Resources management. | 02 March | 01 April |
| 3 | Sprint 3 | -Timetables management.  -Students grades management.  -Library management | 02 April | 01 May |

### System parts fragmentation using WBS (Work Breakdown Chart):

Figure 17: WBS (Work Breakdown Chart)

### **Gantt Diagram**

Figure 18: Gantt Diagram

## Needs Analysis and Specification:

### Product Backlog:

Table : Product Backlog

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Theme | Id User Story | User story | Importance | Estimation |
| 1 | Manage access to the main system | 1.1 | As a Logs Manager, I can consult and manage the access of an administrator or a professor (after have the confirmation from the university administration), which is include add, remove and edit an access. | High | 15 |
| University Details consultation | 1.2 | As an administrator I can consult the internal system of the institution. | Low |
| 1.3 | As an administrator I can consult about university section. | Low |
| 1.4 | As a Professor I can consult the internal system of the institution. |
| 1.5 | As a Professor I can consult about university section. |
| 2 | Student management | 2.1 | As an administrator I can consult, add, remove, edit and search students informations. | High | 20 |
| Staff  management | 2.2 | As an Administrator I can consult, add, remove, edit and search staff details. | High |
| University resources details | 2.3 | As an administrator, I can consult, add, remove, edit and search the university resources informations (or details). | High |
| 3 | Timetables  management | 3.1 | As an administrator I can consult, add, remove, edit and search the classes timetables. | High | 15 |
| Students grades management | 3.2 | As administrator I can consult, add, remove, edit and search students grades. | High |
| 3.3 | As Professor I can consult, add, remove, edit and search students grades Remotely. |
| Library management | 3.4 | As an administrator (Librarian) I can manage books records, pending books requests, pending memberships registrations and update a library member balance. | High |
| 3.5 | As a Library member, I can request a membership, consult the rest of my balance, consult the available books in the university library, and request a book. | High |

### Identification of the SCRUM team:

Table 5: Our Scrum team members

|  |  |  |  |
| --- | --- | --- | --- |
| Role | Product Owner | SCRUM Master | SCRUM Team |
| Member | isima | Mme Chebil  Wiem | Mme Chebil Wiem    Ncir Wissem |

### Identification of actors:

• The Logs Manager:

The Logs Manager should have the access to the system to be able to manage Logs (or the access) to the main system.

• The Administrator:

The Administrator should have the access to the system to benefit from many services such as: student management, staff management, Library Management …

• The Professor:

The Professor should have the access to the system to be able to: view university details and manage students grades.

### Functional needs:

#### Functional needs of the “Login Manager” actor:

• Authenticate:

The Login Manager can access the Login Manager section using a password given by the developer.

• Manage access to the main system:

The Login Manager is able to give, remove and edit an administrator or a professor login. (The administrator or the professor should send a request to the university administration manually, then the administration send confirmation to the Login Manager before he can give, remove or edit a login).

#### Functional needs of the “Administrator” actor:

• Authenticate (Login):

An administrator can access his allowed section inside the main system using an email or username and a password given by the Login Manager.

• consult University Details section:

The administrator is able to consult the About University section that contains “University Details” and “The internal system of the institution”.

• Manage Student:

The administrator is able to add, remove, edit, search and consult students data.

• Manage Staff:

The administrator is able to add, remove, edit, search and consult staff infos.

• Manage university resources Details:

The administrator able to add, remove, edit, search, and consult a university resource details (computers, video projector, printers, desks and chairs, cams …).

• Manage the Timetables section:

The administrator can generate, remove, edit, search, filter and consult classes Timetables.

• Manage Students Grades section:

The administrator is able to add, remove, edit, search and consult students grades.

• Manage Library:

The administrator (Librarian) is able to manage books records, pending books requests, pending memberships registrations and update a library member balance.

#### Functional needs of the “Professor” actor:

• Authenticate:

A Professor can access the main system using an email or username and a password given by the Login Manager.

• consult University Details section:

The Professor is able to consult the About University section that contains “University Details” and “The internal system of the institution”.

• Manage Students Grades section:

A Professor is able to add, remove, edit, search and consult students grades.

#### Functional needs of the “Library Member” actor:

• Membership request:

A student can become a library member by requesting a library membership.

• Authenticate (Login):

A Library Member can access his library account using an email or username and a password given by the Librarian (Administrator).

• Available books consultation:

A Library Member, can consult available books in the library using his account.

• Balance consultation:

A Library Member, can consult his rest of balance in his account.

### Non-functional needs:

• Performance:

This refers to the speed and efficiency of the system in handling user requests and processing data. It includes factors such as response time, throughput (productivity), and scalability.

• simplicity:

The user interface must be simple and easy to understand and handle.

• Security:

This refers to the protection of the system and its data from unauthorized access, theft, or damage. It includes factors such as authentication, authorization and encryption.

•Availability:

Availability, in this context, refers to the system's ability to be operational and accessible when needed, without excessive downtime or interruptions. It is concerned with ensuring that the system is reliable and accessible, often expressed as a percentage of uptime over a specified period.

## Global Diagrams:

### Global use cases diagram:

Figure : Global use cases Diagram

### Global class Diagrams:

#### Management system global class Diagram:

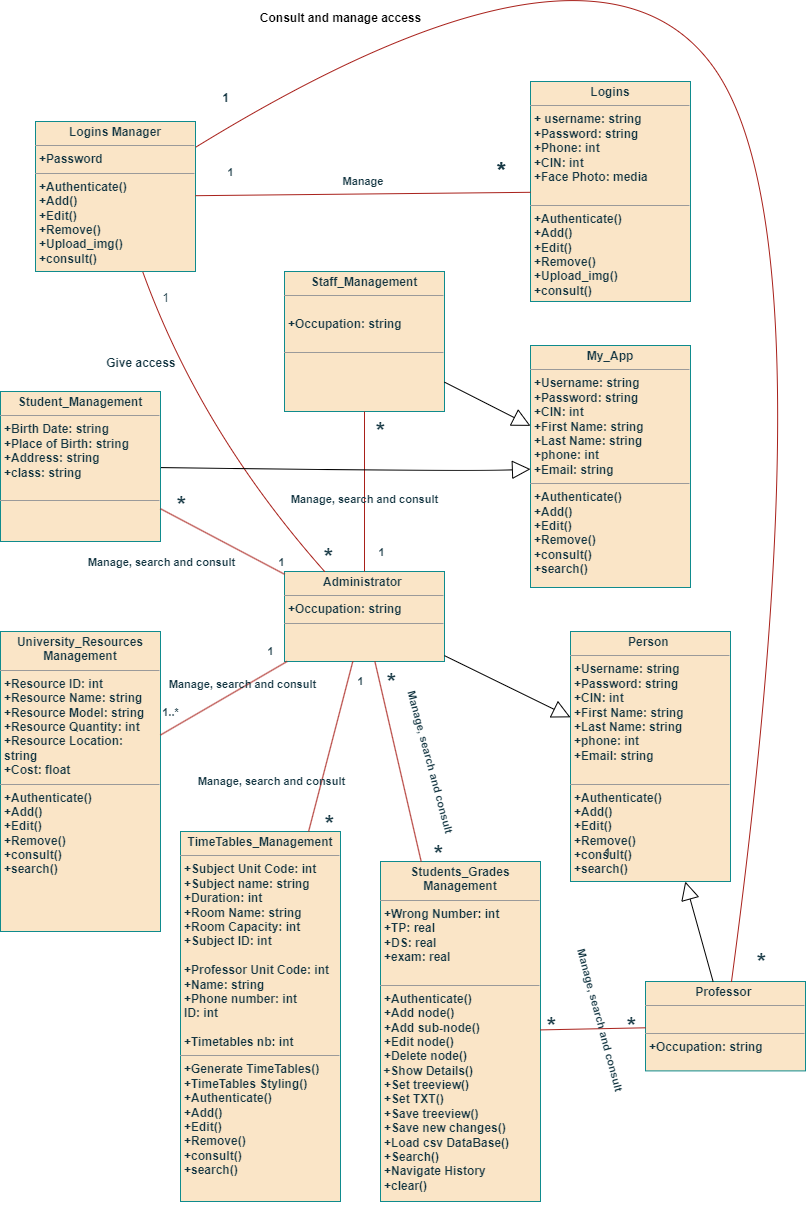


Figure : Management system global class Diagram

#### Library management system global class Diagram:

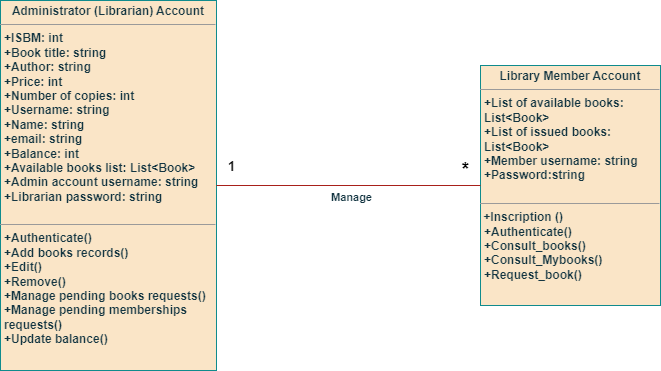


Figure 21: Library management system global class Diagram

conclusion

In this chapter we used Scrum prototyping to track our project progress, we introduced the sprints plan that was made up according to the product backlog, we showed the global use cases diagram, and we described the functional and non-functional requirements.

The next step will be devoted to the development of the first and the second sprint.



# Sprint 1 and Sprint 2

3

# Chapter3: Sprint 1

# and Sprint 2

Introduction

In this chapter we will develop sprint1 and sprint2, by providing a detailed conceptual study for all functionalities, proceeded by demonstration then a test ensuring our system quality.

## The first sprint:

### First Sprint Backlog:

Table 6: First Sprint Backlog

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Theme | Id User Story | User story | Importance | Estimation |
| 1 | Manage access to the main system | 1.1 | As a Login Manager, I can give access (username and password) to an administrator or a professor (after have the confirmation from the university administration). | High | 15 |
| 1.2 | As a Login Manager, I can remove an access of an administrator or a professor. | High |
| 1.3 | As a Login Manager, I can edit a log, such as edit: username/email address, password, user type(Administrator or Professor), user face photo. | High |
| Authenticate | 2.1 | As a Login Manager, I should authenticate using my password to be able to access the Login Manager section. | High |
| 2.2 | As an administrator, I should authenticate using my username and password to access the main system. | High |
| 2.3 | As a Professor I should authenticate using my username and password to access students grades section and consult university details. | High |
| 2.4 | As an administrator, if I forgot my password, I can use Ai Facial Recognition instead to login. |
| 2.5 | As a professor, if I forgot my password or username or username and password I can use  Ai Facial Recognition instead to login. |
| University Details consultation | 3.1 | As an administrator I can view the internal system of the institution | Low |
| 3.2 | As an administrator I can view about university section | Low |
| 3.3 | As a Professor I can view the internal system of the institution |
| 3.4 | As a Professor I can view about university section |

### Sprint 1 use case diagram:



Figure 22: Sprint1 use cases Diagram

### Uses cases Analysis:

#### “Manage access to the main system” use Analysis case:

##### Textual description:

Table 7: Textual description of use case “Authenticates”

|  |  |
| --- | --- |
| Use case | Manage access to the main system |
| Description | The Login manager, wants to manage to access to the system. |
| Actor | Login Manager. |
| Precondition | The Login Manager must be authenticated. |
| Postcondition | Administrator or Professor logins can be consulted,  added, removed, edited, or searched. |
| scenario | 1-The Login Manager access the GUI Login manager section.  2-The system display logins data.  3-The Login Manager consult and select the required Login.  4-The Login Manager can choose to create, remove, or edit Login. |

##### Sequence diagram of “Manage Access to the system” use case:

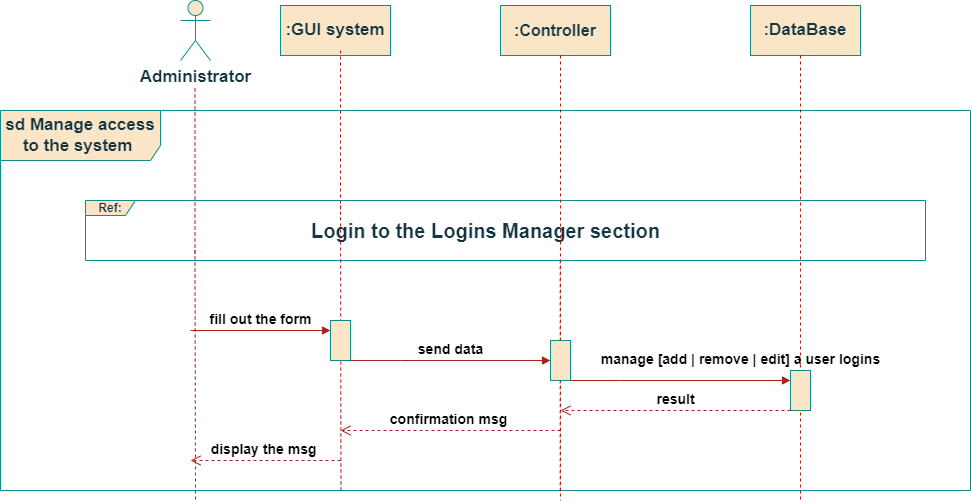


Figure : Manage Access to the system sequence Diagram

##### Activity diagram of “Manage Access to the system” use case:

Figure 24: Manage Access to the system activity Diagram

#### “Authenticates” use case **Analysis:**

##### Textual description:

Table 8: Textual description of authenticates use case

|  |  |
| --- | --- |
| Use case | Authentication |
| Description | The Login manager, the administrator or the professor must authenticate  to benefit from all the services of the system. |
| Actor | Login Manager, Administrator, Professor |
| Postcondition | The administrator or the professor is authenticated and redirected tothe  system interface |
| scenario | 1-The system displays the login interface.  2-The user enters his identifiers (email, password)  2.0-if the user forgot his username/email or and password, he can use the  Ai facial recognition option to login.  3-The user validates the form.  4-The system verifies the form of the data entered.  5-The system checks the validity of the identifiers.  6-the system directs the user to the GUI. |
| Alternative  scenario | 1.1-the user enters incorrect or missing identifiers.  1.2-the system displays an error.  1.3-return to stage 1 of the scenario. |

##### Sequence diagram of “Authenticates“ use case:

Figure 25: Sequence diagram “Authenticate“

##### Activity diagram of “Authenticates” use case:

Figure 26: Activity Diagram “Authenticate”

##### Realization:

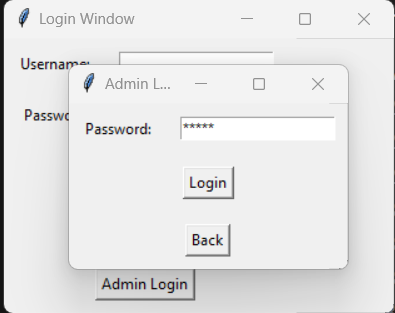


Figure 27: Admin Login interface

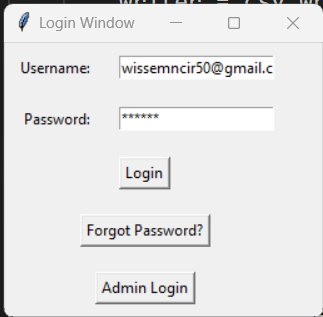


Figure 28: user Login interface

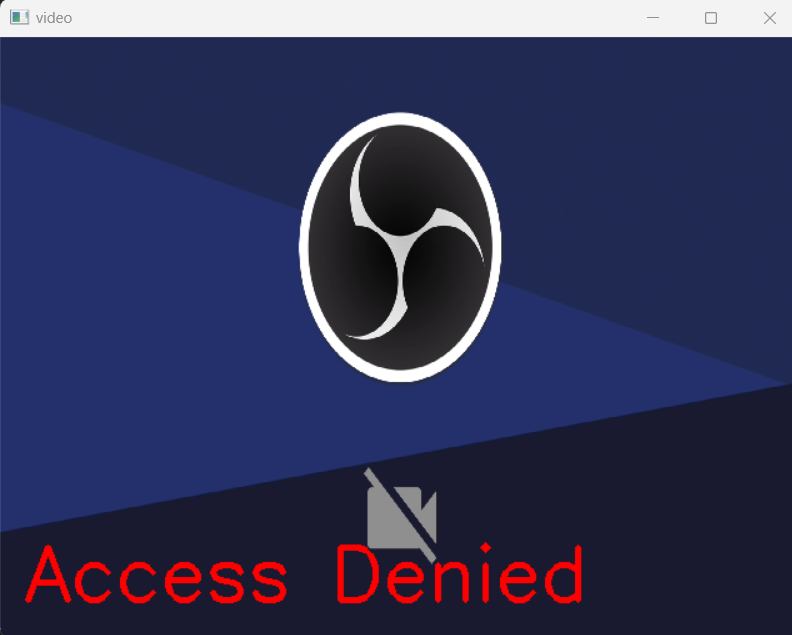


Figure : Password reset using Ai Artificial Recognition interface

##### Test:

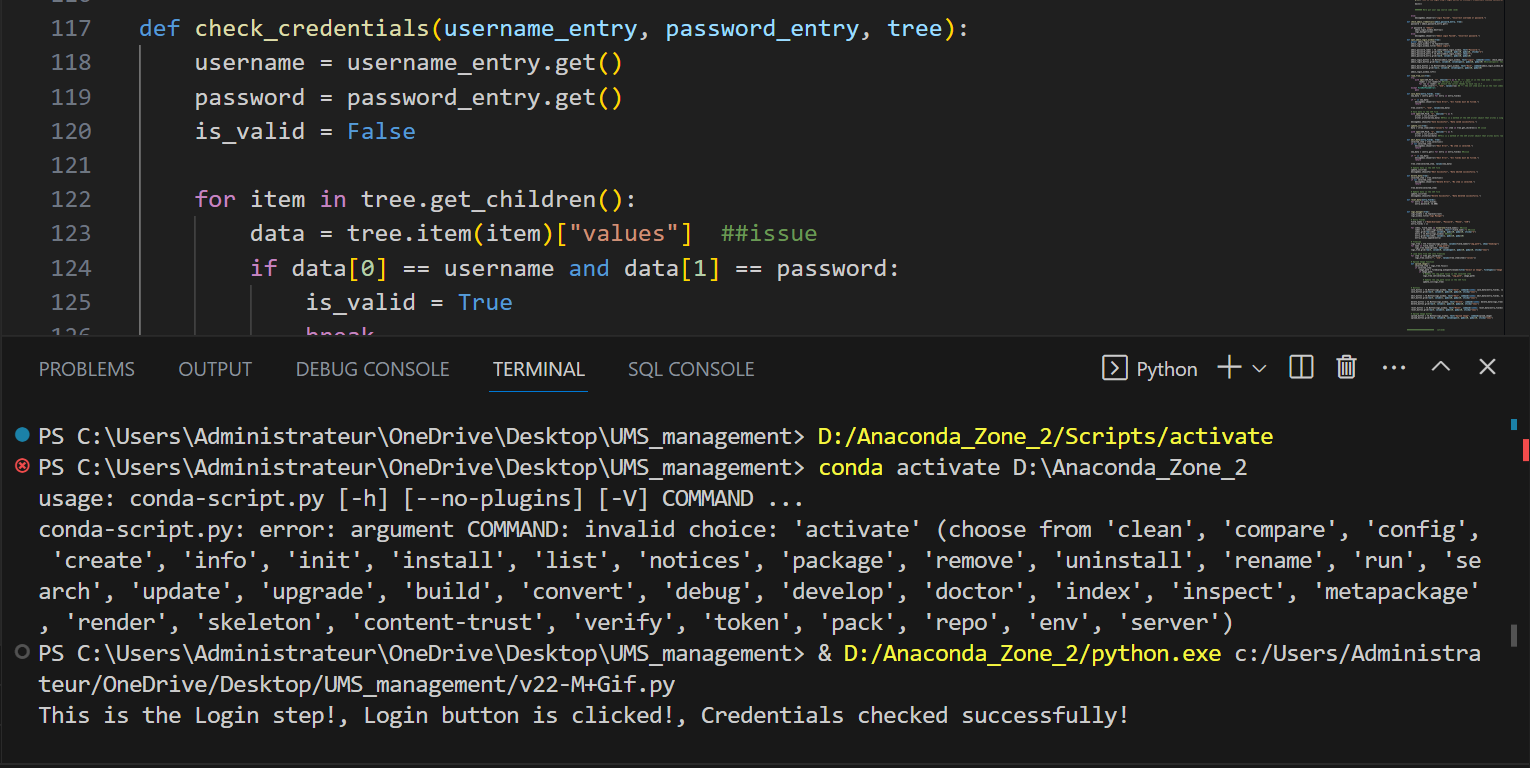


Figure 30: user is logged test

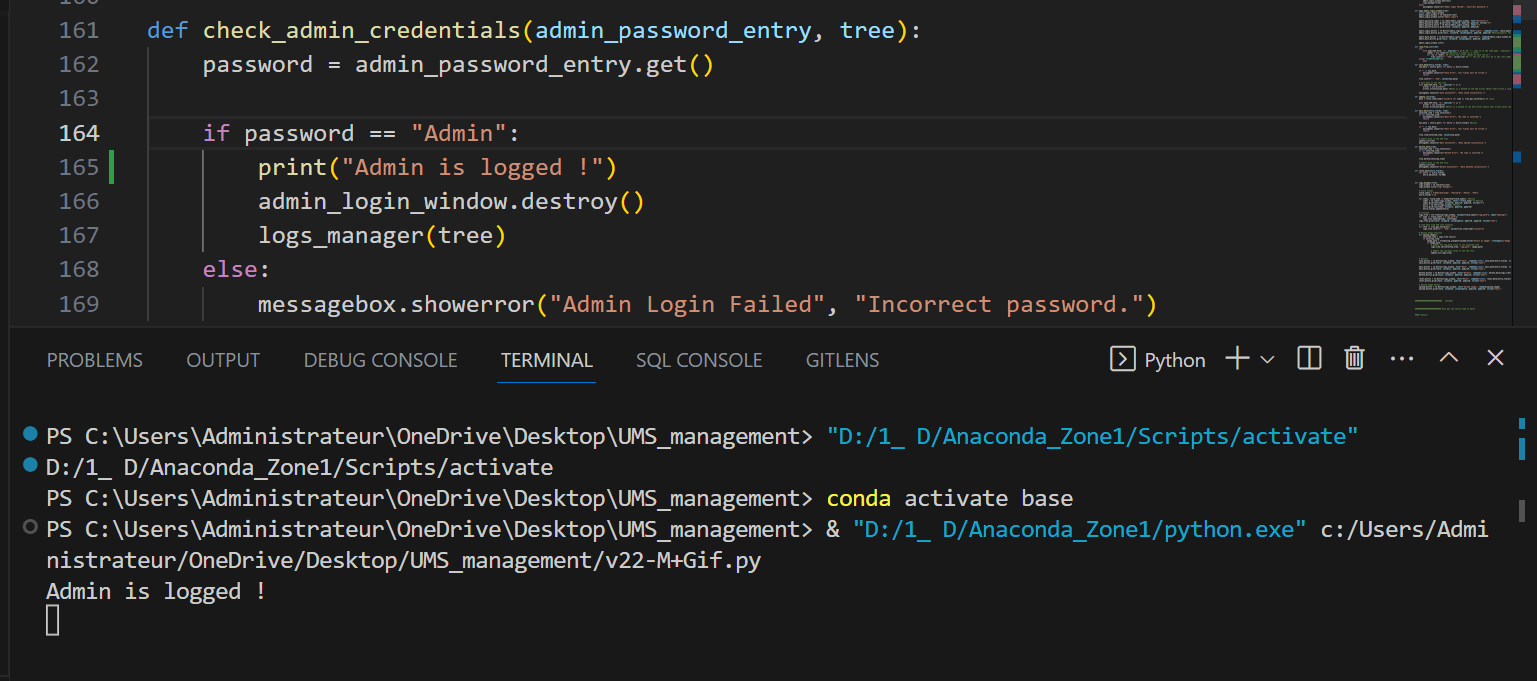


Figure 31: Admin is logged test

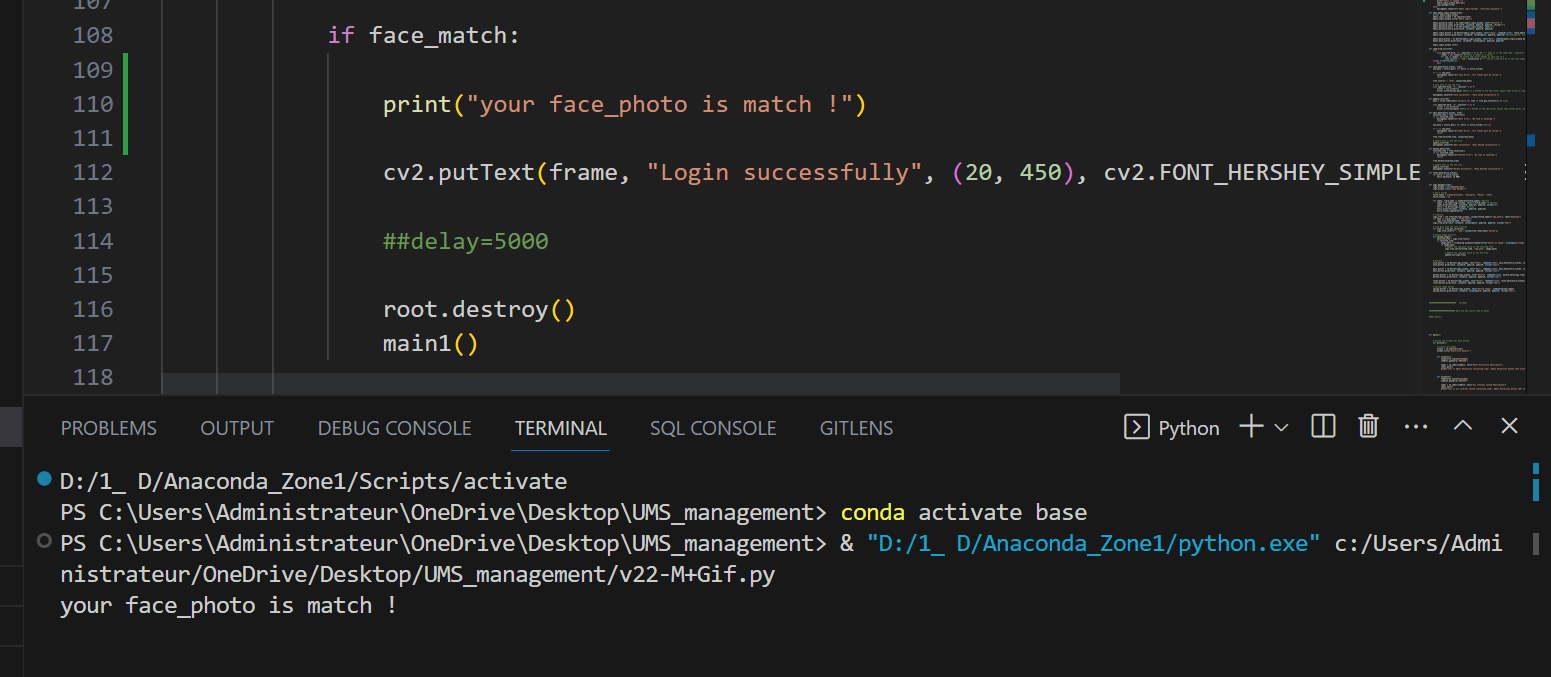


Figure 32: face\_photo matching test

#### “University Details consultation” Uses cases Analysis:

##### Textual description:

Table 9:Textual description of use case “University Details consultation”

|  |  |
| --- | --- |
| Use case | University Details consultation |
| Description | The administrator or a professor, can view the university details section. |
| Actor | Administrator, Professor. |
| Precondition | The administrator or the professor must access the system. |
| Postcondition | The administrator or the professor is redirected the university details section. |
| scenario | 1-The system displays the GUI .  2-The administrator or the professor get into the university details section.  3-The administrator or the professor get into “About University” section.  4-The administrator or the professor can consult informations within “About University”. |
| Alternative  scenario | 1-The system displays the GUI  2- The administrator or the professor get into the university details section.  3- The administrator or the professor get into “Institution Internal System” section.  4- The administrator or the professor can consult items into the Institution Internal System. |

##### sequence diagram of “University Details consultation” use case:

Figure 33: Sequence diagram “University Details consultation”

##### Activity diagram of “University Details consultation” use case:

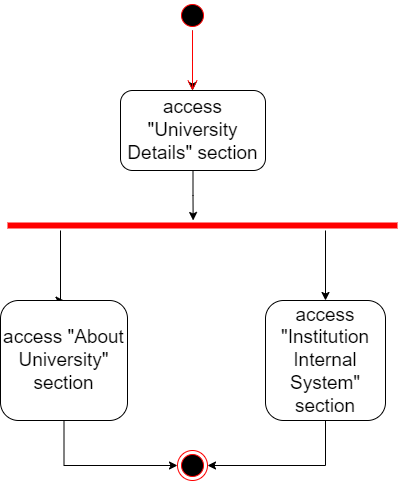


Figure 34: Activity Diagram “University Details consultation”

##### Realisation:

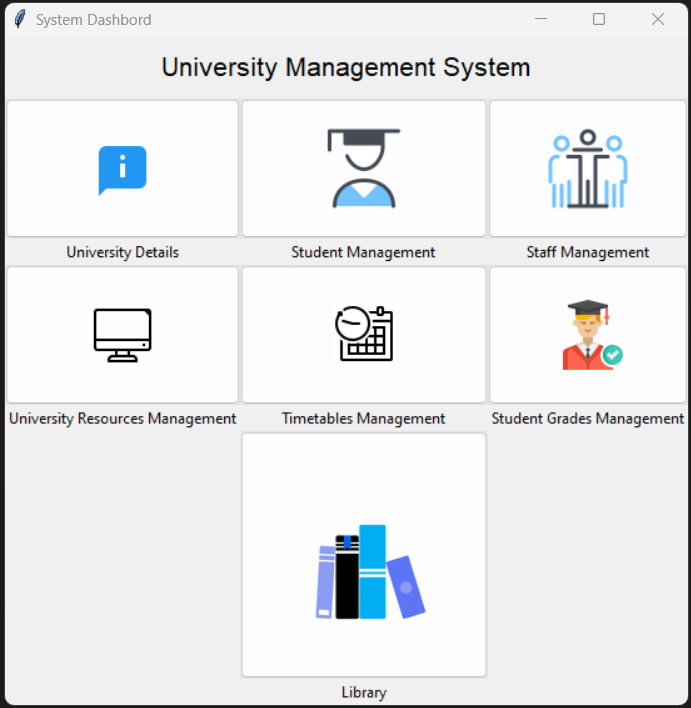


Figure 35: Main System Dashboard



Figure 36: University Details section

##### Test:

Figure 37: Consulting About University section test

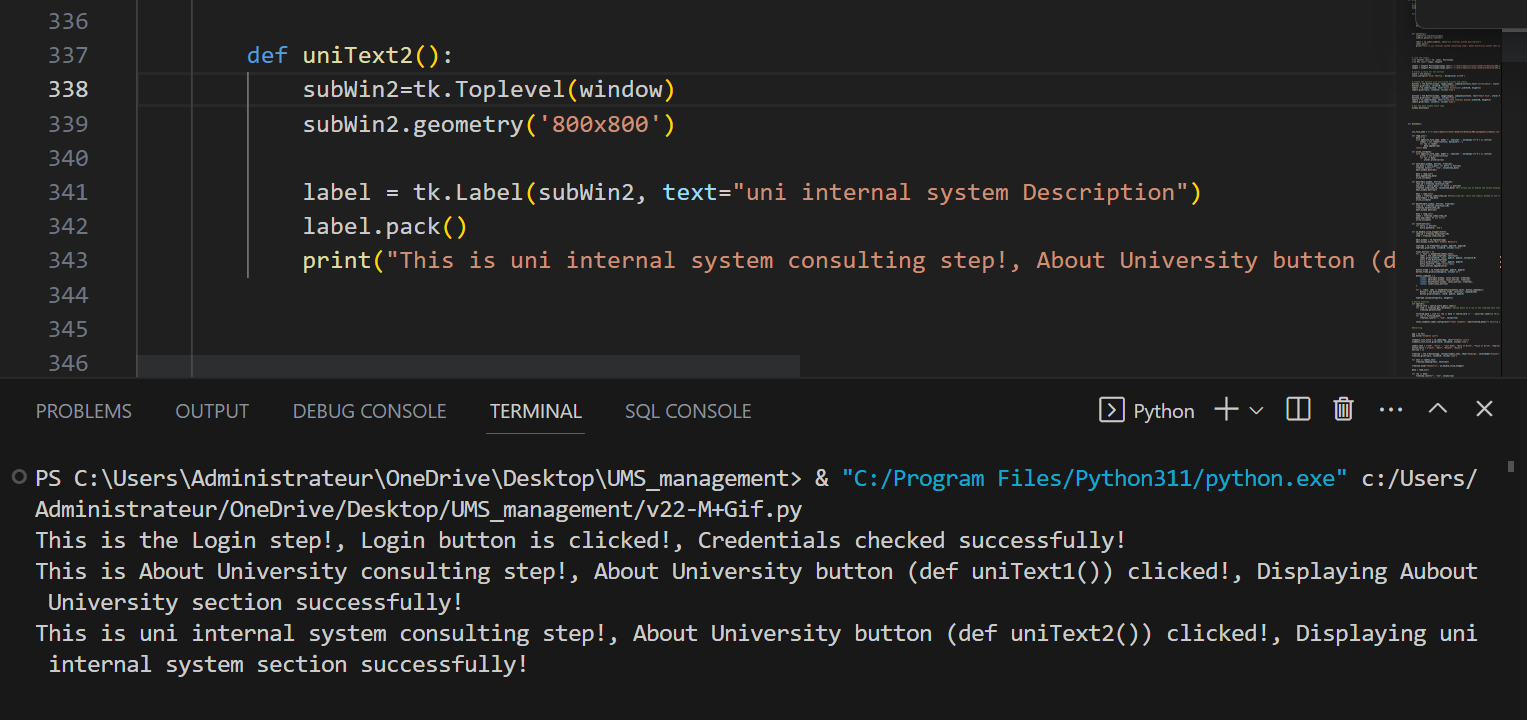


Figure 38: Consulting University internal system test

## The second sprint:

### Second Sprint Backlog:

Table 10: Sprint2 Backlog

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Theme | Id User Story | | User story | | Importance | Estimation |
| 2 | Student management | | 2.1 | | As an administrator I can consult, add, remove, edit and search students details. | High | 20 |
| Staff  management | | 2.2 | | As an Administrator I can consult, add, remove, edit and search staff details. | High |
| University resources details | | 2.3 | | As an administrator, I can consult, add, remove, edit and search the university resources details. | High |

### Sprint 2 use case diagram:

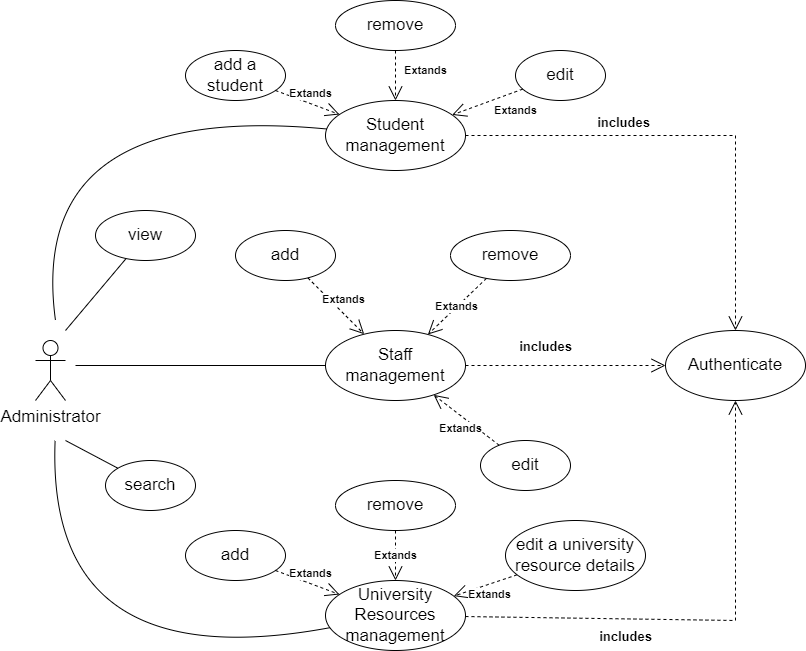


Figure 39: Sprint 2 use case Diagram

### Uses cases Analysis:

#### “Student management” **uses cases Analysis:**

##### Textual description

Table 11: Textual description of use case “Student management”

|  |  |
| --- | --- |
| Use case | Student details management |
| Description | The administrator wants to manage student details (informations). |
| Actor | Administrator. |
| Precondition | As an administrator should be authenticated. |
| Postcondition | Student details can be consulted, added, edited removed or searched. |
| scenario | 1-The administrator accesses the system GUI.  2-The system displays all students details.  3-The administrator search the required student.  4-the system displays all students details (infos).  5-The administrator have the chose to add or remove or edit student(s) details. |
| Alternative  scenario | 3.1- The administrator enters incorrect filtering data.  3.2- The system displays nothing.  3.3- Return to the step 2 of the scenario. |

##### sequence diagram of “Student Details management” use case:

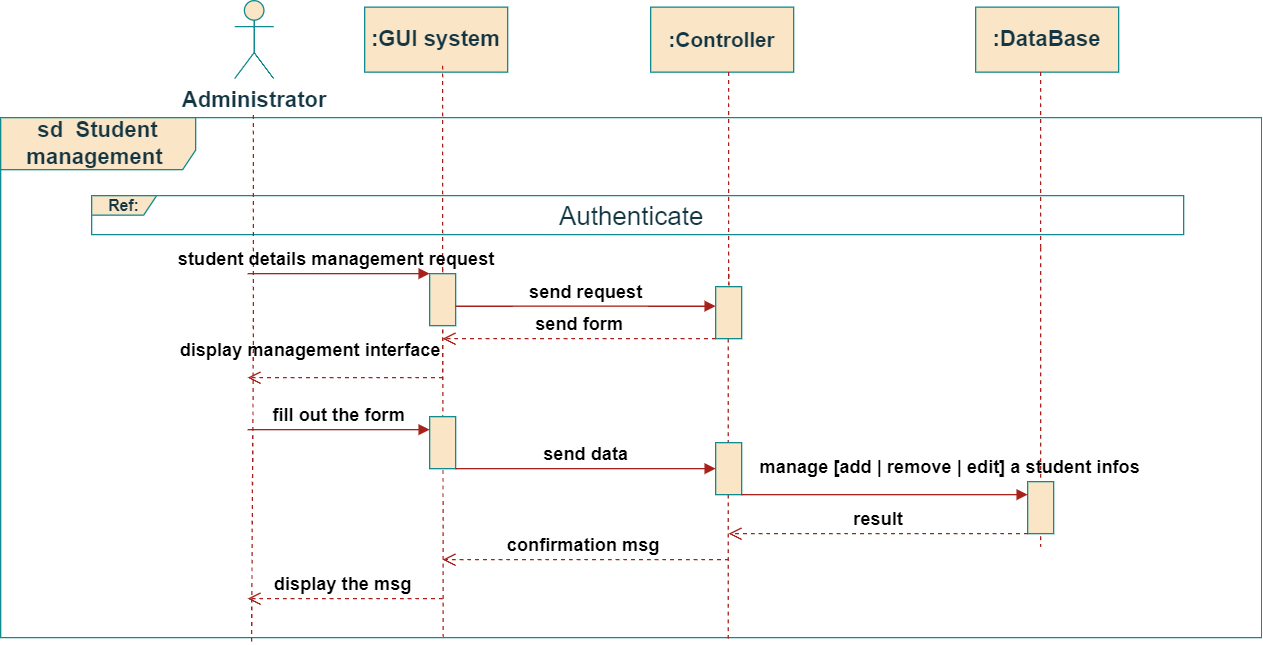


Figure 40: Sequence Diagram “Student Details management”

##### Activity diagram of “Student Details management” use case:

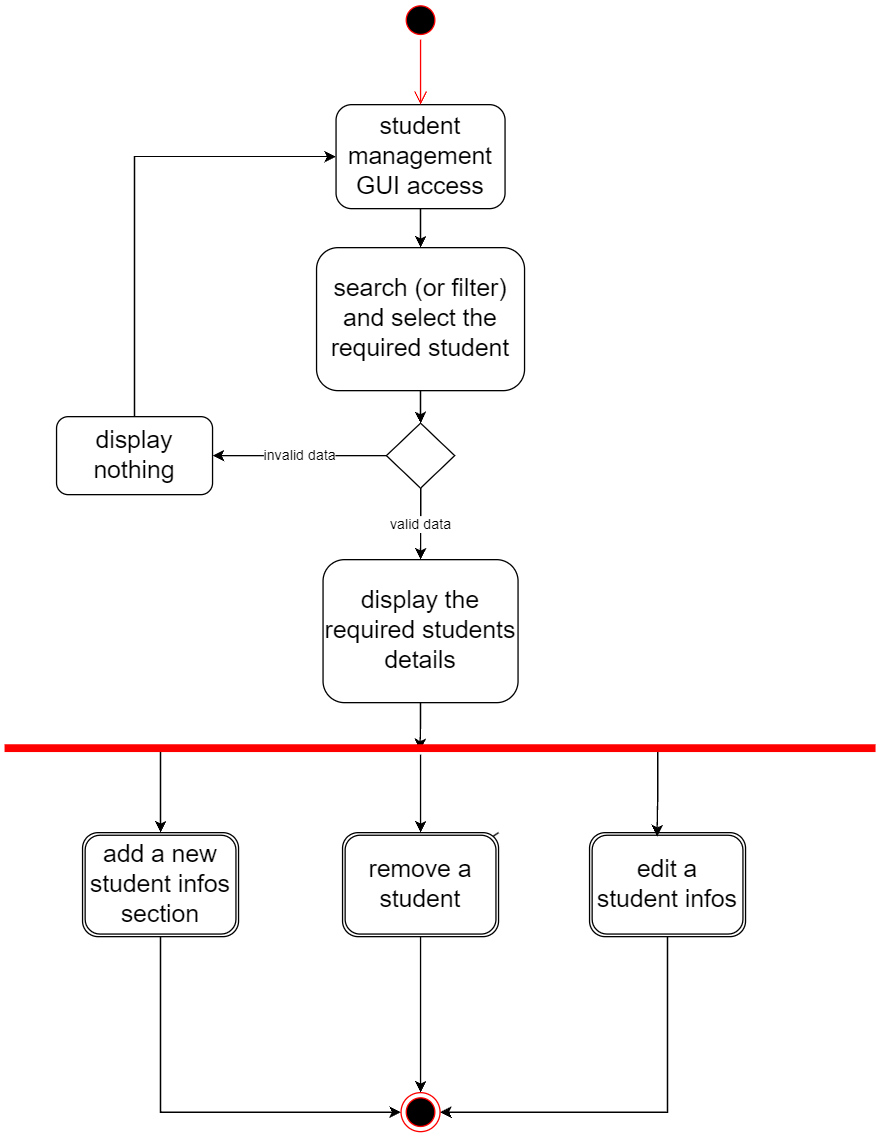


Figure 41: Activity diagram “Student Details management”

##### Realisation:

Figure 42: Student Management interface

##### Test:

Figure 43: Student Management section test

#### “Staff management” uses cases analysis:

##### Textual description:

Table 12: Textual description of use case “Staff management”

|  |  |
| --- | --- |
| Use case | Student details management |
| Description | The administrator wants to manage staff details. |
| Actor | Administrator. |
| Precondition | The administrator should be authenticated |
| Postcondition | Staff details can be filtered or searched or viewed or added or removed or edited. |
| scenario | 1-The administrator accesses the system GUI.  2-The system displays all staff details.  3-The administrator search the required Staff member.  5-The administrator have the chose to view, search, add, remove or edit staff member(s) details. |
| Alternative  scenario | 3.1- The administrator enters incorrect filtering data.  3.2- The system displays nothing.  3.3- Return to the step 2 of the scenario. |

##### sequence diagram of “Staff Details management” use case:

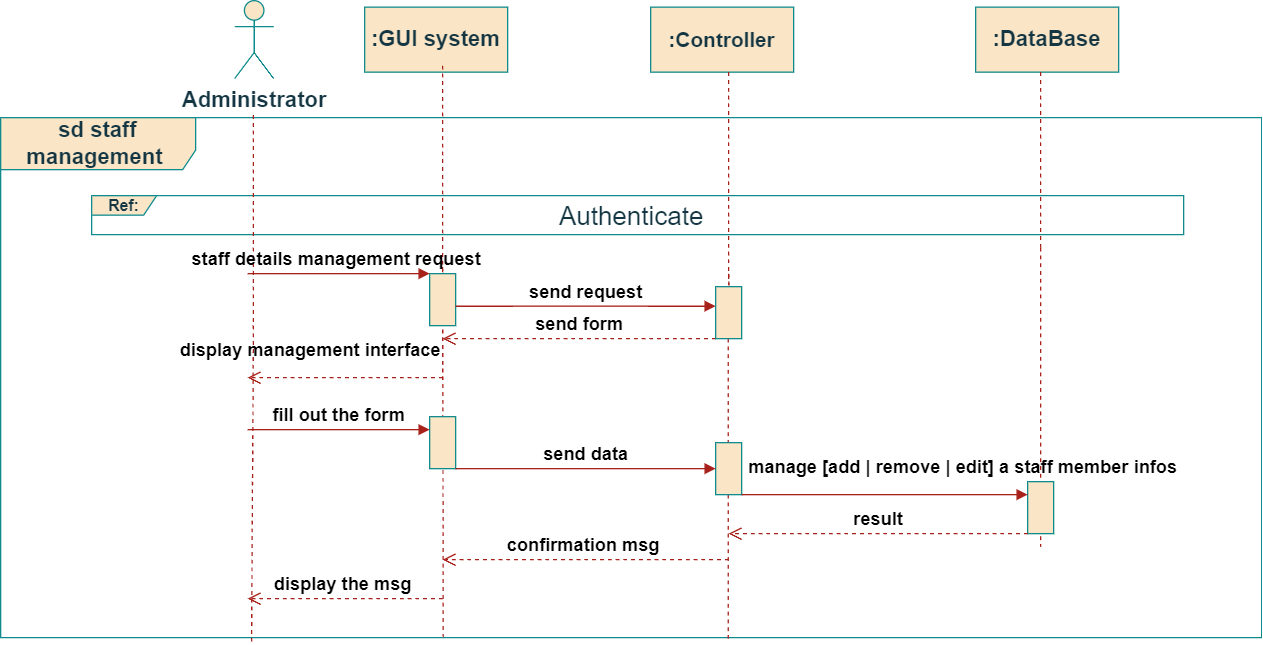


Figure 44: Sequence diagram “Staff Details management”

##### Activity diagram of “Staff Details management” use case:

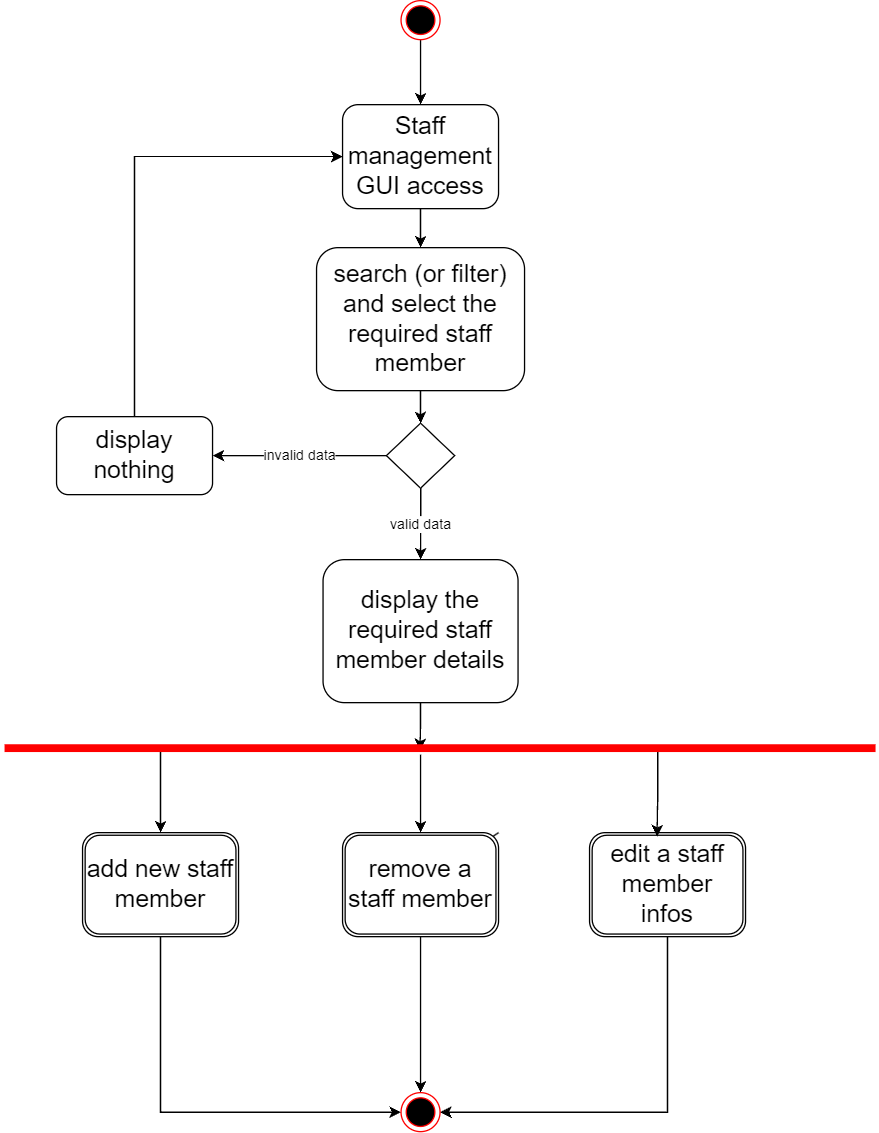


Figure 45: Activity diagram “Staff Details management”

##### Realisation:

Figure 46: Staff Management interface

##### Test:

Figure 47: Staff Management section test

#### University resources uses cases Analysis:

##### Textual description:

Table 13: Textual description of use case “University Resources Details management”

|  |  |
| --- | --- |
| Use case | University Resources Details management |
| Description | The administrator wants to manage University Resources Details. |
| Actor | Administrator. |
| Precondition | The administrator should be authenticated. |
| Postcondition | University Resources Details can be consulted, added, edited removed or searched. |
| scenario | 1-The administrator accesses the system GUI  2-The system displays all University Resources Details  3-The administrator search the required university resource.  4-the system displays all university resources details.  5-The administrator have the choice to consult, add, remove, edit or search a university resource (s) details. |
| Alternative  scenario | 3.1- The administrator enters incorrect filtering data.  3.2- The system displays nothing.  3.3- Return to the step 2 of the scenario. |

##### sequence diagram of “University Resources Details management” use case:

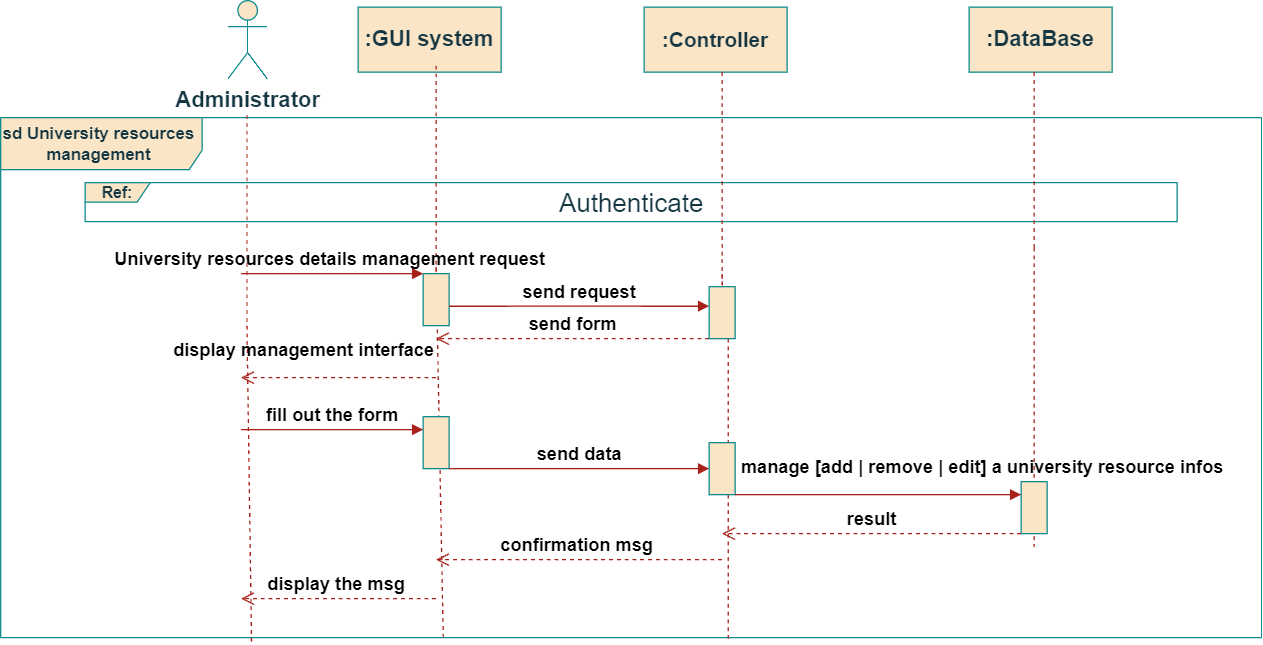


Figure 48: Sequence Diagram “University Resources Details management”

##### Activity diagram of “University Resources Details management” use case:

Figure 49: Activity diagram “University Resources Details management”

##### Realisation:

Figure 50: University Resources Management interface

##### Test:

Figure 51: University Resources Management section test

conclusion

This part showed us the functionalities of the first and the second sprint, from the study of use cases to their implementation. The next chapter will be devoted to setting up the third sprint.



# Sprint 3

# 

# 

# Chapter 4: Sprint 3

Introduction

In this chapter we will develop sprint 3, which the most difficult and delicate one, by providing a detailed conceptual study for all functionalities, proceeded by demonstration then a test ensuring our system quality.

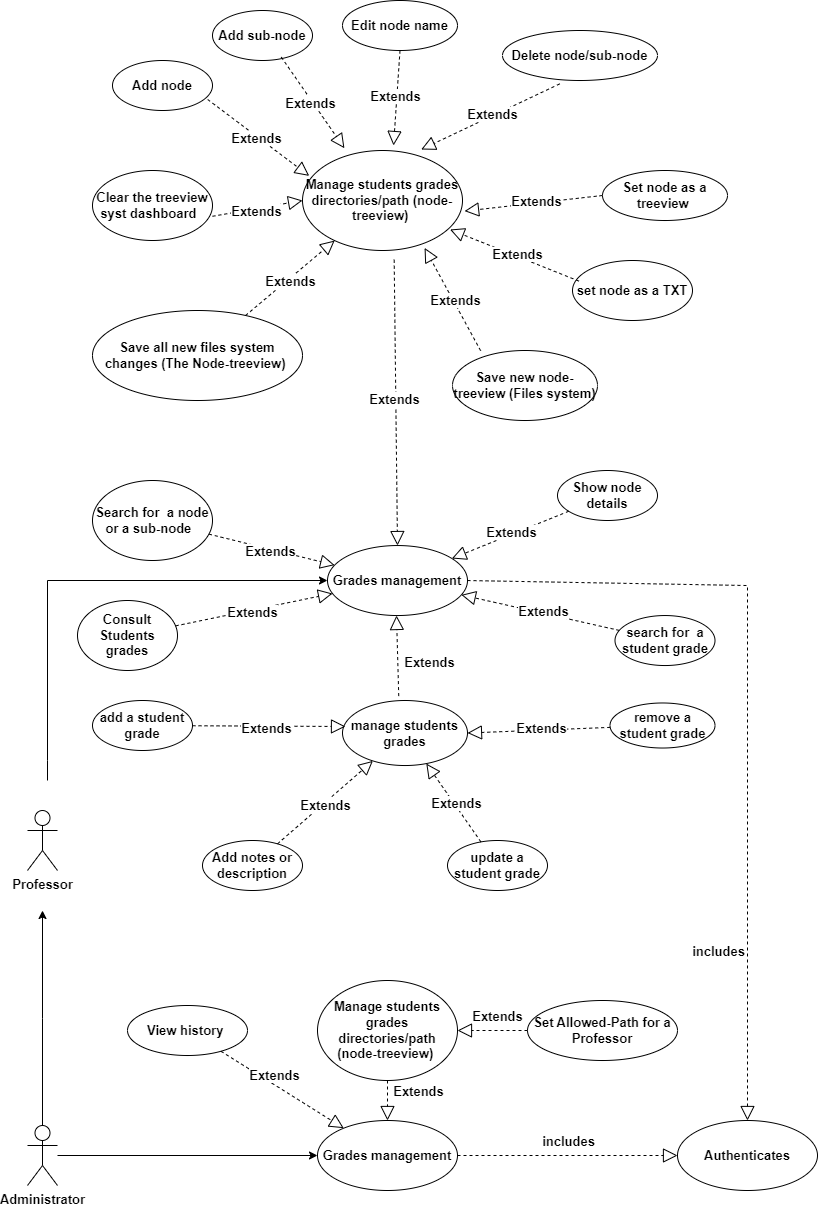
## The third sprint:

### Third Sprint Backlog:

Table 14: Third Sprint Backlog

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Theme | Id User Story | User story | Importance | Estimation |
| 3 | Timetables  management | 3.1 | As an administrator, I can consult, add, remove, edit and search the classes timetables. | High | 15 |
| 3.2 | As administrator, I can consult, add, remove, edit and search students grades. |
| Students grades management | 3.3 | As a professor, I can consult, add, remove, edit and search students grades Remotely. | High |
| Library management | 3.4 | As an administrator (Librarian) I can manage books records, pending books requests, pending memberships registrations and update a library member balance. | High |
| 3.5 | As a Library member, I can request a membership, consult the rest of my balance, consult the available books in the university library, and request a book. | High |

### Sprint 3 use case Diagram:



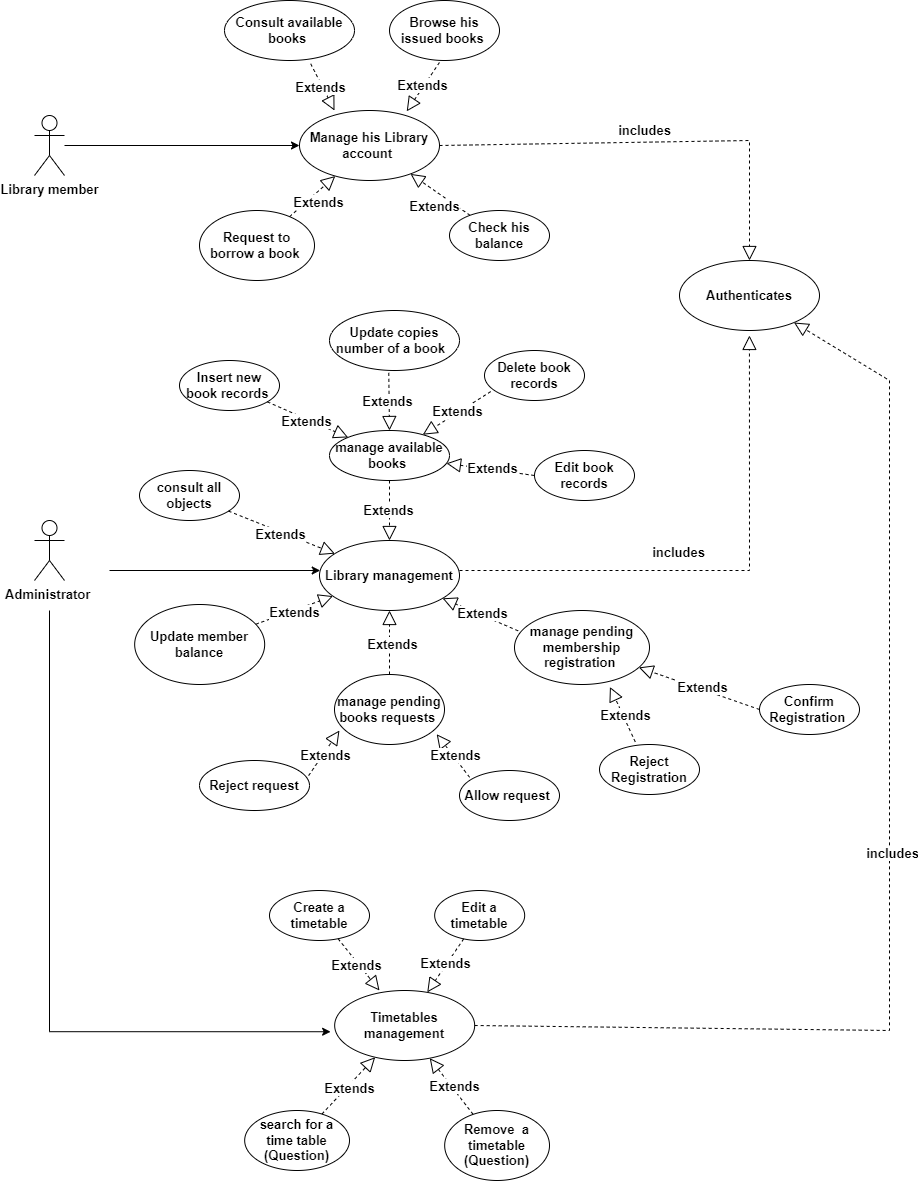


Figure 52: Sprint 3 use case Diagram

### Uses cases Analysis:

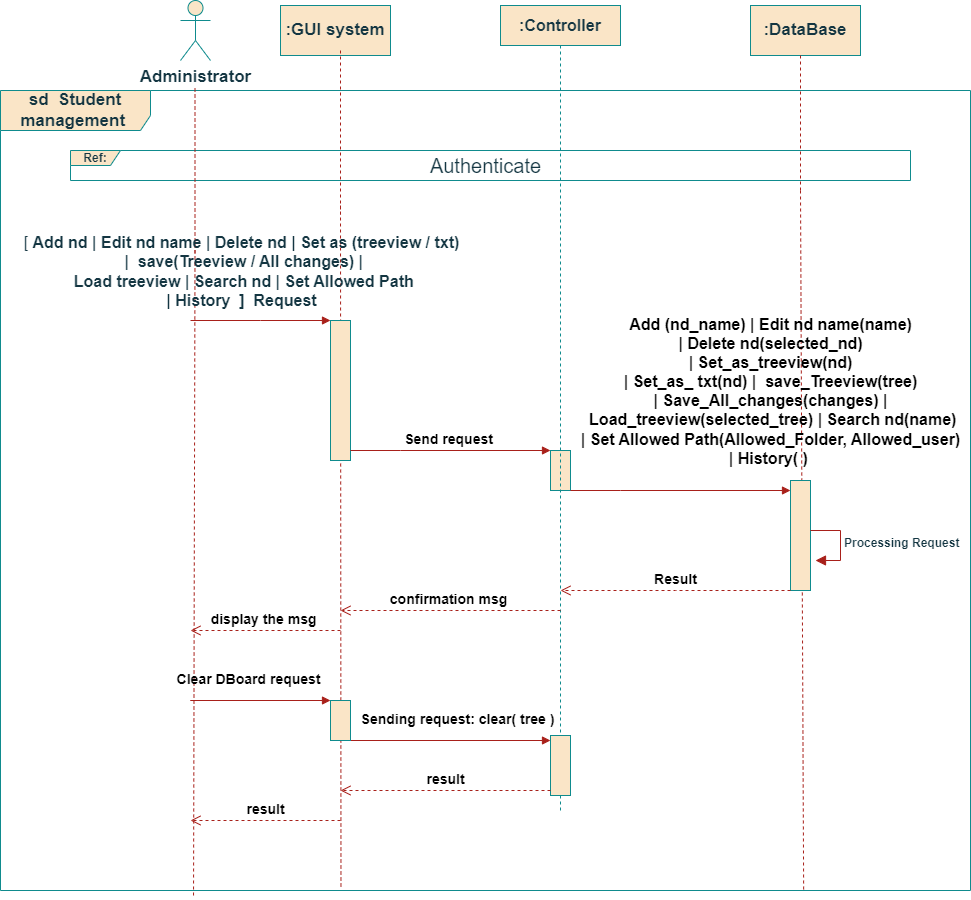
#### “Students grades management” u**ses cases Analysis:**

##### Textual description:

Table 15: Textual description of “Students grades management” use case

|  |  |
| --- | --- |
| Use case | Student grades management |
| Description | The administrator or the Professor wants to manage the corresponding files system first, then manage students grades. |
| Actor | Administrator, Professor. |
| Precondition | Administrator or Professor should be authenticated. |
| Postcondition | Students grades files system (which is a collection of nodes) inside the main system can be added, named or renamed, edited, removed, node content displaying, node seted as a treeview or a txt file, saved, changes saved, loaded, searched, it recent activities or operations browsed using the history tab, allowed or denied access to a user and treeview system dashboard cleared.  Now after the building of the node files treeview system, students grades can be viewed, added, edited, deleted and searched. |
| scenario | 1-The administrator or the professor access the system GUI.  2-The system displays all students grades.  3-The administrator or the professor search the required students grades.  5-The administrator have the choice to view, add node or sub-node, edit node name, remove a node, node content displaying, set a node as a treeview or txt file, save new files system(treeview), save new treeview changes, load a treeview, search a node or clear treeview system dashboard.  6-Also he has the choice to view, search, add, edit, remove student grades itself (After the building of the node files treeview system). |
| Alternative  scenario | 3.1- The administrator or the professor enters incorrect filtering data.  3.2- The system displays nothing or an error message.  3.3- Return to the step 2 of the scenario. |

##### Sequence diagram of “students grades management” use case:



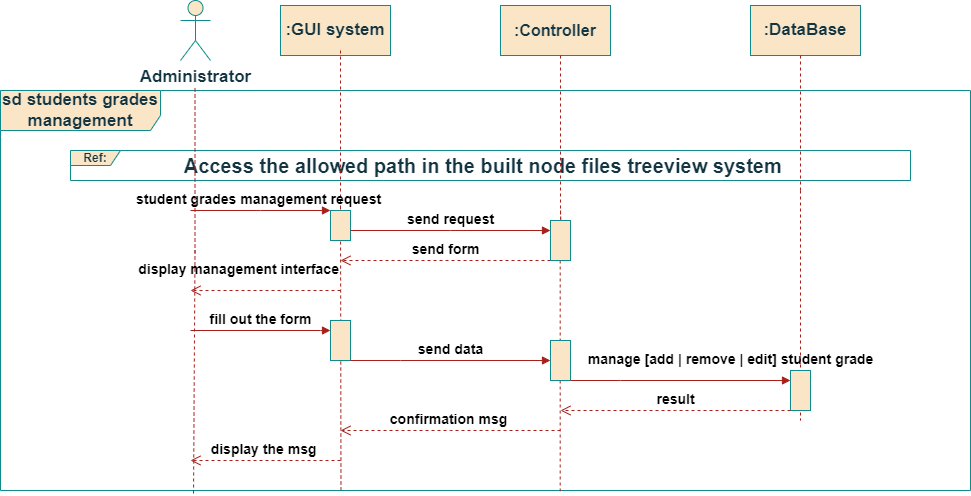


Figure 53: Sequence diagram of “students grades management”

##### Activity diagram of “Students grades management” use case:

Figure 54: Activity Diagram

“Students grades management”

##### Realisation :

Figure : Grades section Login interface

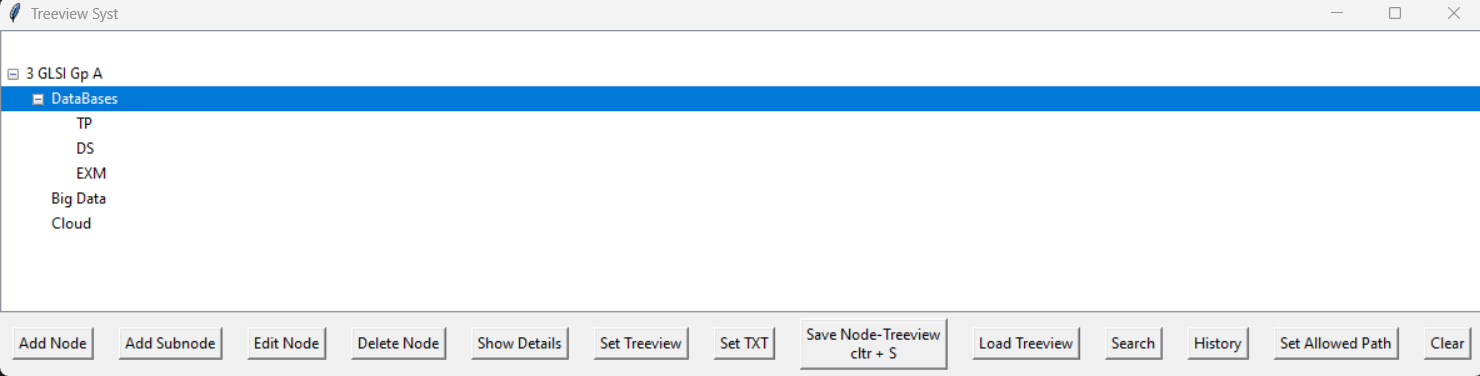


Figure 56: Student Grades node system interface

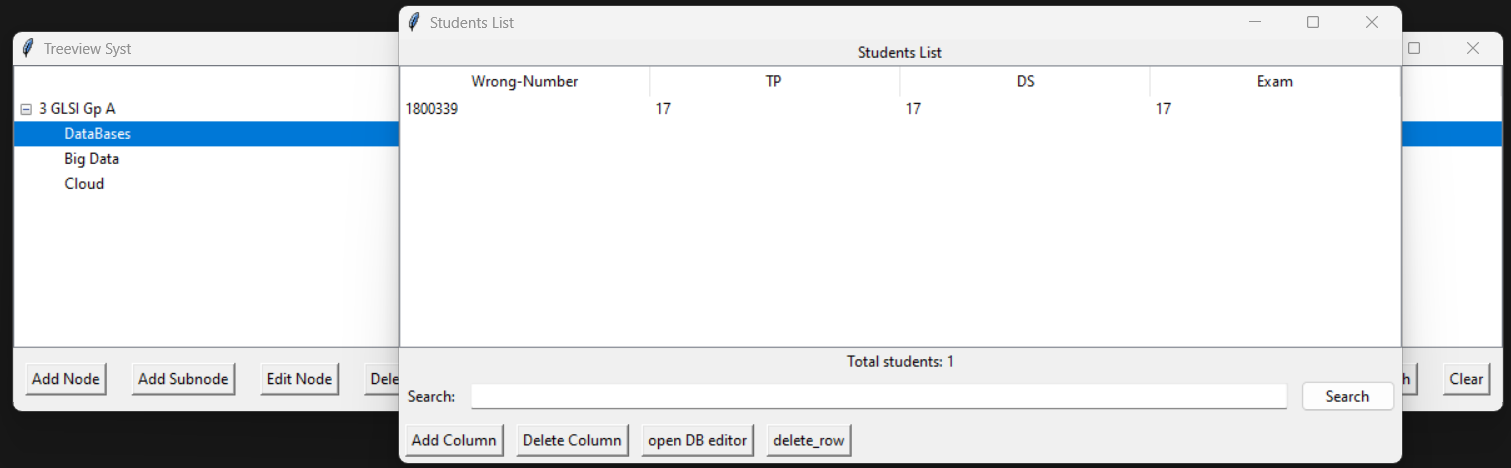


Figure 57: Grades management interface

##### Test :

Figure 58: Grades Admin Section test

#### “Timetables management” uses cases Analysis:

##### Textual description:

Table 16: Textual description of “Timetables management” use case

|  |  |
| --- | --- |
| Use case | Timetables management |
| Description | The administrator wants to manage timetables. |
| Actor | Administrator. |
| Precondition | Administrator should be authenticated. |
| Postcondition | Timetables can be, consulted, created, removed, searched or edited. |
| scenario | 1-The administrator accesses the system GUI of Timetables management section.  2-The system displays all timetables.  3-The administrator search and select the required timetable.  5-The administrator have the choice to consult, remove, edit, create or search a timetable. |
| Alternative  scenario | 3.1- The administrator enters incorrect filtering data.  3.2- The system displays nothing.  3.3- Return to the step 2 of the scenario. |

##### Sequence diagram of “Timetables management” use case:

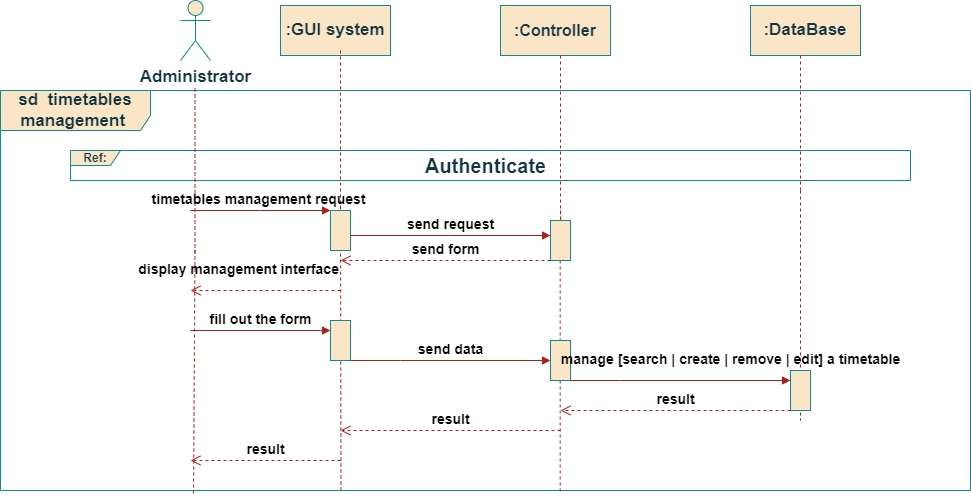


Figure 33: Sequence Diagram “Timetables management”

##### Activity diagram of “Timetables management” use case:

Figure 59: Activity Diagram “Timetables management”

##### Realisation:

Figure 60: Timetables management interface

##### Test:

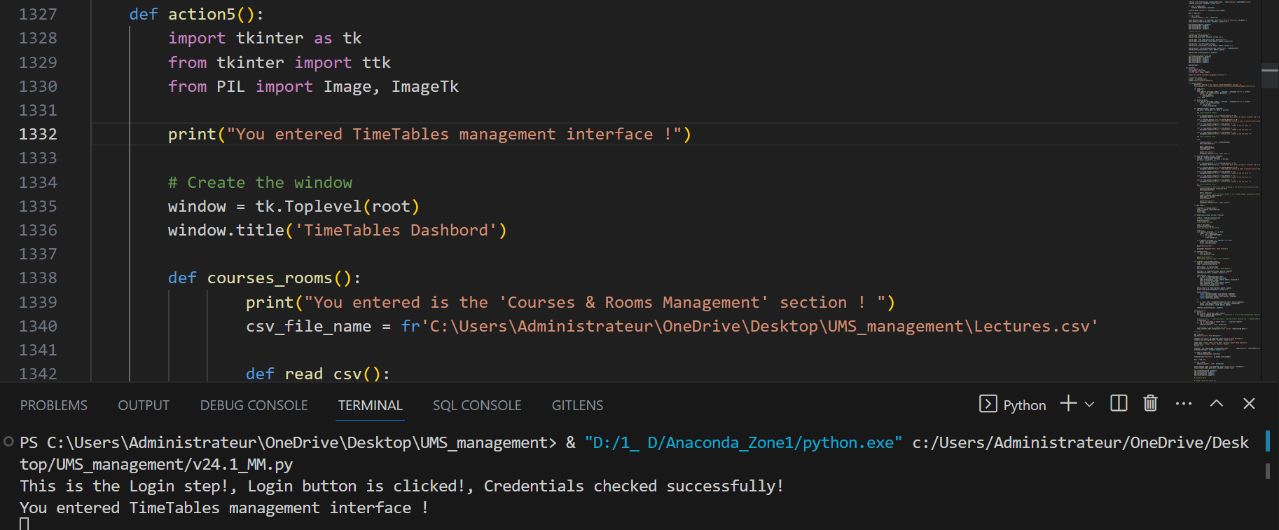


Figure 61: Timetables interface test

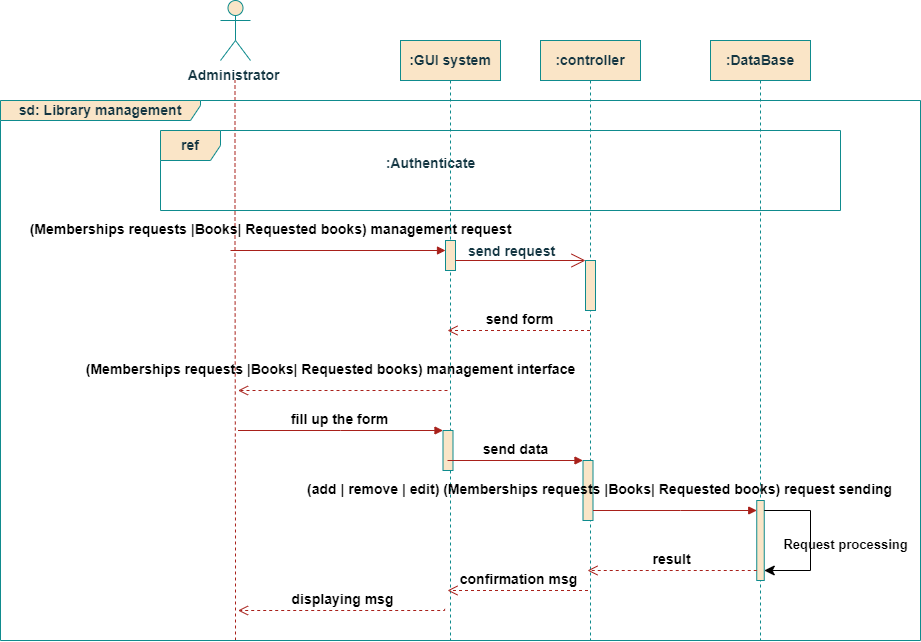
#### “Library management” uses cases Analysis:

##### Textual description:

Table 17: Textual description of “Library management” use case

|  |  |
| --- | --- |
| Use case | Library management |
| Description | The administrator wants to manage the university library. |
| Actor | Administrator (Librarian), Library member. |
| Precondition | Administrator should be authenticated. |
| Postcondition | The available books of the library can be managed by insert new book records, update books copies number manually (cause the library can receive new shipment or quantity of new books) and delete or edit book records.  Also pending memberships registration can be managed (accept or reject registration).  Also, manage pending books requests (accept or reject request).  In addition, a member balance can be updated. |
| scenario | 1-The administrator access the GUI system of the library section.  2-The system displays all library members and available books.  3-The administrator select the required member or book.  5-The administrator have the choice to remove, edit or add a book informations.  6- A member request a membership registration.  7- The Administrator can accept or reject this request.  8- A registered member consult “My Books” section, his balance and request a book.  9-The Administrator can accept or reject this request.  10- The Administrator can update a member balance. |
| Alternative  scenario | 3.1- The administrator enters incorrect filtering data.  3.2- The system displays nothing or an error message.  3.3- Return to the step 2 of the scenario. |

##### Sequence diagram of “Library management” use case:



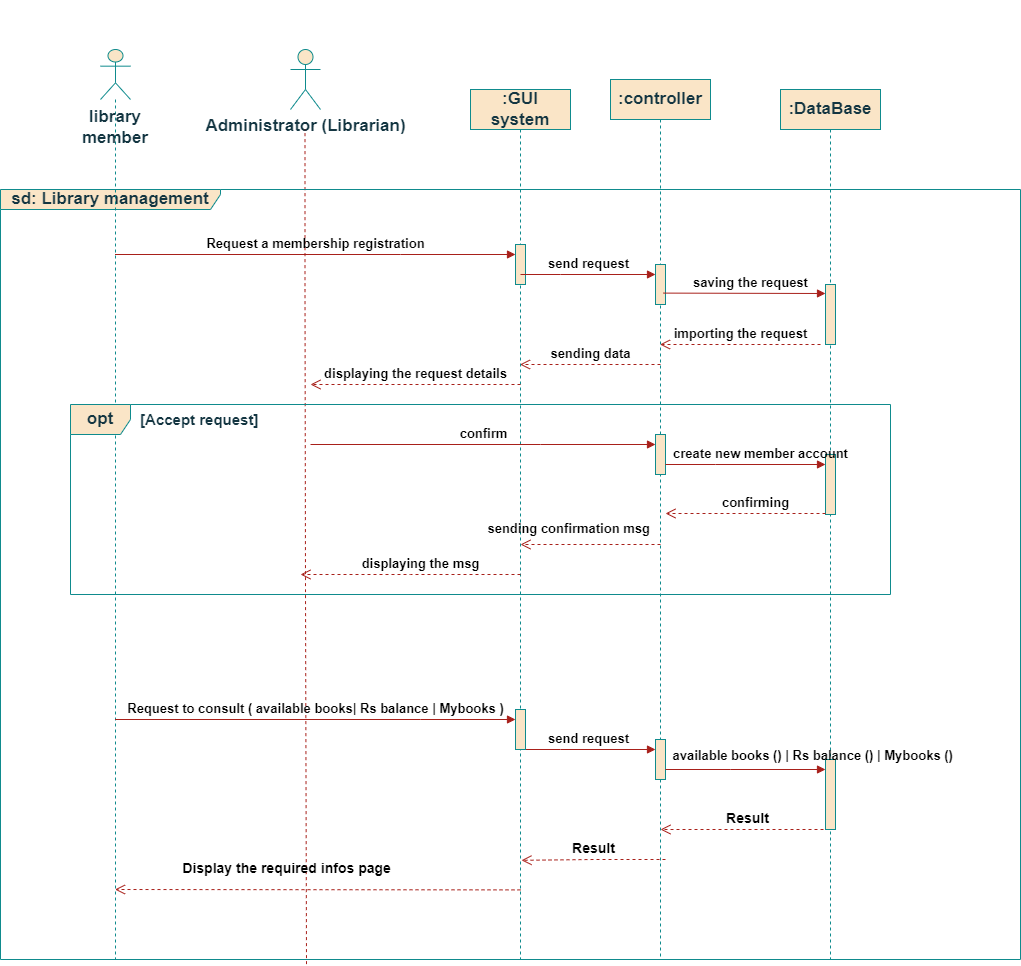


Figure 62: sequence diagram “Library management”

##### Activity diagram of “Library management” use case:

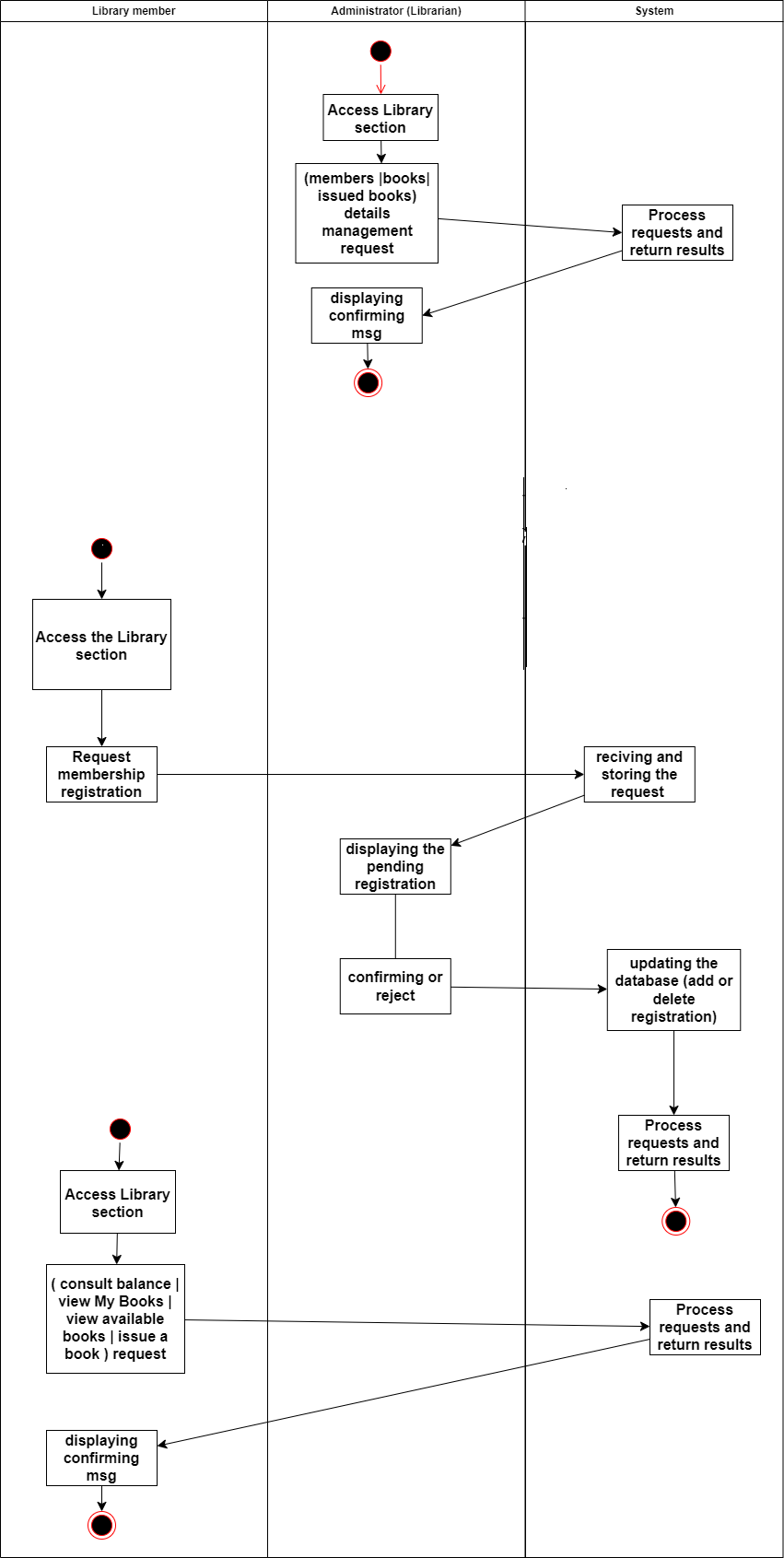


Figure 63: Activity diagram “Library management”

##### Realisation:

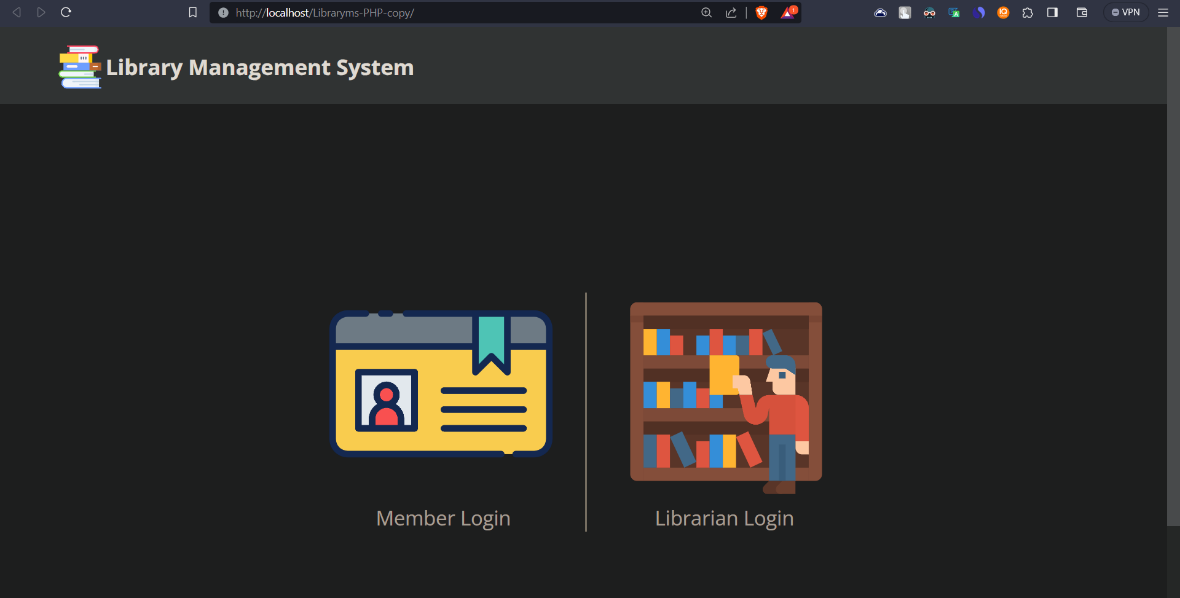


Figure 64: Member and Librarian Login interface

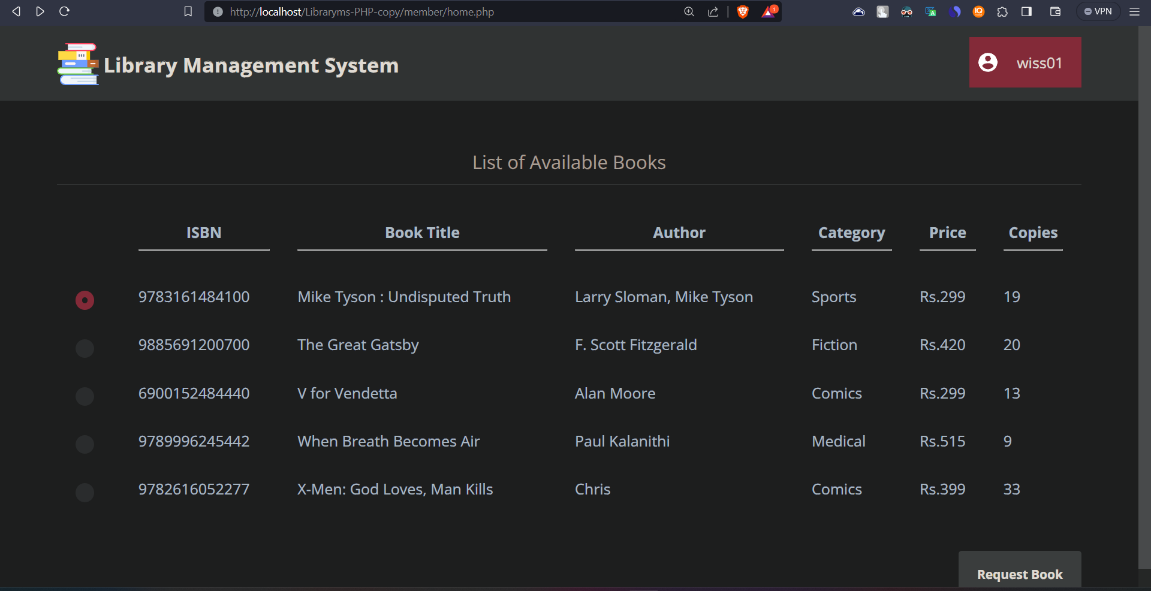


Figure 65: Library member home page

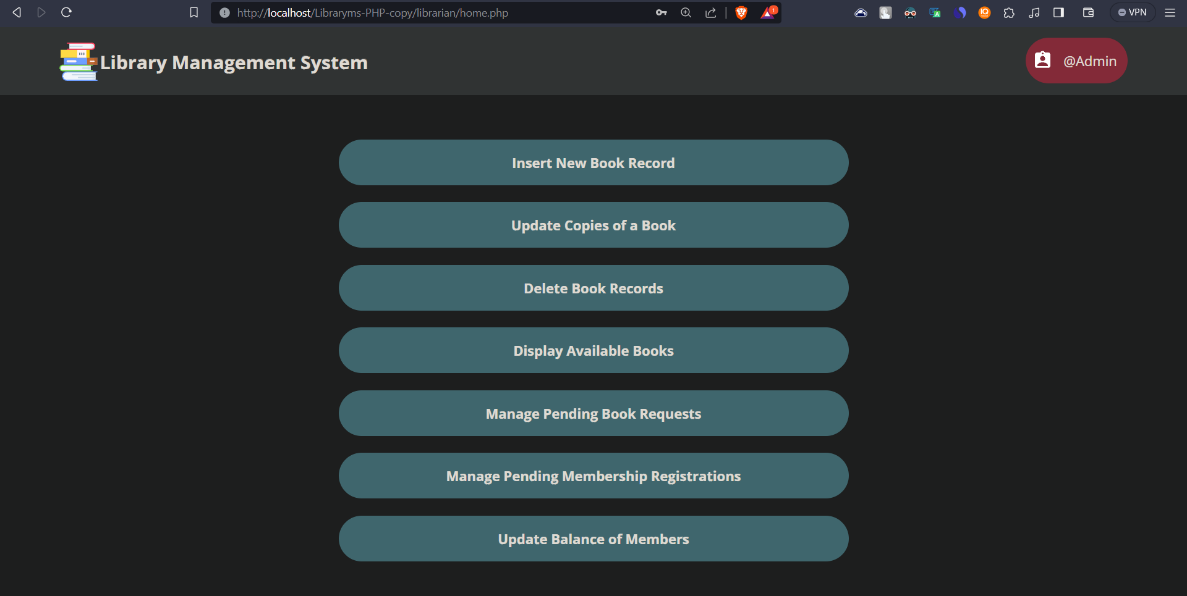


Figure 66: Librarian home page

conclusion

In this chapter we have processed the different functionalities (features) of sprint 3 in our project.

General conclusion and perspectives

At the end of this report, we can distinguish that this training offers us an excellent opportunity to compare my theoretical knowledge in the practical environment, and sincerely there is gap between the two.

Following this training we discovered new conceptual techniques, development of innovation, etc. We have a chance to exercise our skills acquired during our university courses.

Our idea is not limited here, it still to provide a powerful security system from the university sending grades software, cause a powerful security system is crucial for maintaining the confidentiality, integrity, and availability of sensitive grade-related data, protecting the university's reputation and legal compliance, and fostering trust among students, faculty, and administrators, by Preventing Data Breaches, improve user authentication, **Secure Communication** (encrypt Communication between user and system), **Audit Trails** (A security system can track who accessed what data and when, creating an audit trail. This is valuable for accountability and investigating any suspicious activities), **Protection Against Insider Threats** (Even within an organization, there can be malicious or unintentional actions that compromise data. A strong security system helps mitigate the risk of insider threats), Long-Term Data Storage (Universities often need to retain student data for a long time. A security system ensures the data's integrity and accessibility over time) and add **Honeypot** Security system [24], that is a computer security mechanism set to detect, deflect, or, in some manner, counteract attempts at unauthorized use of information systems, this system can even use bots to let attackers fell like they are hacking the system, but the system at the same time already trace the attackers locations and informations. Also, we can make the grades system suitable to any DB or it DB installable on any other main system, improve the GUI style and design and real web deployment using PythonAnyWhere.

Webographies

[1] : <https://duckduckgo.com>

<https://www.ecoles.com.tn/sites/default/files/styles/large/public/universite/logo/isima-logo.jpg?itok=_bFFya0r>

[2] : <https://www.scrum.as/academy/2/gfx/c2.1.jpg>

[3]: <https://scrummastercertification.ie/why-choose-scrum>

[4] : <https://www.visual-paradigm.com/scrum/what-are-scrum-events/#toc1>

[5] : <https://www.scrum.org/resources/blog/how-do-3-scrum-roles-promote-self-organization>

<https://scrumorg-website-prod.s3.amazonaws.com/drupal/inline-images/2019-01/scrum%20team.png>

[6]: <https://www.atlassian.com/agile/scrum/artifacts>

[7]: <https://www.atlassian.com/agile/scrum/artifacts>

[8]: <https://www.visual-paradigm.com/scrum/what-are-scrum-events>

[9] : <https://www.eclee.com/training/scrum>

[10] : <https://www.eclee.com/wp-content/uploads/2020/01/Scrum_Pic.png>

[11] : <https://en.wikipedia.org/wiki/Unified_Modeling_Language>

[12] : <https://en.wikipedia.org/wiki/Work_breakdown_structure>

[13] : <https://en.wikipedia.org/wiki/Software_architecture>

[14] : <https://en.wikipedia.org/wiki/Multitier_architecture#Three-tier_architecture>

[15] : <https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvc-architecture-and-frameworks-explained>

[16] : <https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)>

[17] : <https://en.wikipedia.org/wiki/Visual_Studio>

[18]: <https://en.wikipedia.org/wiki/Python_(programming_language)>

[19] : <https://en.wikipedia.org/wiki/PHP>

[20] : <https://en.wikipedia.org/wiki/XAMPP#References>

[21] : <https://en.wikipedia.org/wiki/Diagrams.net>

[22] : <https://en.wikipedia.org/wiki/MySQL>

[23] : <https://en.wikipedia.org/wiki/GitHub>

[24] : <https://en.wikipedia.org/wiki/Honeypot_(computing)>