



University of El Oued
Faculty of Exact Sciences

End of Year project report

Attendance Justification Processing System

Authors:

Ahmed Yacine Lacheheb
Abdelmalek Bourouba
Akram Said Khtara

Class:

3rd year INFO

Supervised by:

Mr. Othmani

Academic Year: 2023/2024

Acknowledgment

I would like to express my deepest gratitude to all those who have contributed to the completion of this project. Firstly, I extend my sincere appreciation to my supervisor, **Supervisor Othmani**, for their invaluable guidance, encouragement, and support throughout this journey. Their expertise and insightful feedback have been instrumental in shaping this project.

I am also thankful to the faculty members of the **Exact Sciences** for their constructive criticism and suggestions that have helped enhance the quality of this work.

I am indebted to my classmates and friends for their moral support and encouragement during challenging times. Their encouragement kept me motivated and focused on achieving my goals.

Furthermore, I would like to acknowledge the assistance provided by the staff at the **Faculty** for their resources and facilities that were crucial for the successful completion of this project.

Last but not least, I am grateful to my family for their unwavering support, understanding, and patience throughout this endeavor.

Abstract

This project aims to address the challenges faced by educational institutions in efficiently managing and processing student attendance justifications. Utilizing Laravel and MySQL for the backend, and Vue.js with Electron for the frontend, we developed a comprehensive system comprising two phases.

In the first phase, administrators can input class lists and manage student attendance records, facilitating the process of identifying and recording absences. The second phase involves the creation of a user-friendly website interface for students to submit attendance justifications.

Our system not only streamlines the process of managing attendance justifications but also enhances accountability and transparency within educational institutions. While potential challenges such as integration issues and user adoption exist, the implementation of this system represents a significant advancement in attendance management practices.

Future research could explore additional features to further improve the system's functionality, including automated notifications, attendance analytics, and integration with existing student information systems.

Keywords : Attendance management, Justification processing system, Laravel, Vue.js, Electron

Contents

List of Figures	5
General introduction	6
1 Project Setting	7
1.1 Introduction	7
1.2 Introduction to the University	7
1.3 Administrative Structure of the University	7
1.4 Project Introduction	8
1.5 Current Absentee Justification Method	8
1.5.1 Drawbacks and Challenges	9
1.6 Proposed Solution	9
1.6.1 Desktop Management Software	9
1.6.2 Student Website	10
1.6.3 Expected Benefits	10
2 System Design	11
2.1 Introduction	11
2.2 Use Case Diagrams	11
2.3 Sequence Diagrams	15
2.4 Activity diagram	18
2.5 Class Diagrams	22
3 Environment and Tools	24
3.1 Introduction	24
3.2 Development Environment	24
3.2.1 Laravel Framework	24
3.2.2 Vue.js	24
3.2.3 Vuetify.js	24
3.2.4 Electron.js	24
3.3 Programming Languages Used	25
3.3.1 HTML	25
3.3.2 CSS	25
3.3.3 JavaScript	25
3.3.4 PHP	25
3.4 Tools Used	25
3.4.1 Visual Studio Code	25
3.5 Website Presentation	26

Conclusion	29
Bibliography	30

List of Figures

2.1	General Use Case Diagram.	11
2.2	Teacher Use Case Diagram.	12
2.3	Student Use Case Diagram.	12
2.4	Staff Use Case Diagram.	13
2.5	Faculty Admins Use Case Diagram.	13
2.6	Moderators Use Case Diagram.	14
2.7	Login Sequence Diagram.	15
2.8	insert Student Sequence Diagram.	15
2.9	Manage Users Sequence Diagram.	16
2.10	Justification Check Sequence Diagram.	16
2.11	Justification Manage Sequence Diagram.	17
2.12	Justification Submit Sequence Diagram.	18
2.13	Teacher Activity Diagram.	18
2.14	Student Activity Diagram.	19
2.15	Staff Activity Diagram.	19
2.16	Faculty Admins Activity Diagram.	20
2.17	Moderators Activity Diagram.	21
2.18	Class Diagrams	22

General introduction

In schools and colleges, keeping track of students' attendance is a big job. It's important because it helps teachers know who's coming to class and who's not. But doing this job manually can be slow and sometimes mistakes happen. That's why we're working on a new system to make it easier and faster to manage attendance.

The Problem, Solutions and Objectives :

The problem we're solving is that current ways of managing attendance are old-fashioned and take up a lot of time. Writing down who's in class and who's not, and dealing with students' reasons for missing class can be a headache. Our goal is to create a system that makes this process smoother and less stressful for everyone involved.

Our main aim is to make a system that helps teachers and school staff keep track of attendance more easily. Specifically, we want to make it simple for both teachers and students to enter and check attendance records. By doing this, we hope to save time and reduce mistakes in the attendance process.

To make this happen, we're using modern technology like computers and special software. We'll use a mix of different tools and methods to create the system step by step. This includes things like building the system's 'brains' using a programming language called Laravel and making it look good and easy to use with a tool called Vue.js. We'll also use a method called Agile to make sure we're building the system in the best way possible.

Our project includes making the whole system from start to finish. This means creating the software, making guides on how to use it, and training materials for teachers and students. We'll also write a report explaining everything we did and what we found out along the way.

1. Project Setting

1.1 Introduction

The study of what exists is the first and main stage we rely on in our project. It aims to gather as much information as possible about the system under study. Through preliminary study of the project, we reach a detailed examination of the reality itself, and we can, through it, carry out an organizational solution and create a future vision for the system.

1.2 Introduction to the University

A university is a high-level educational institution that offers academic teaching and research in a variety of disciplines and fields of study. The university is a center for learning and cognitive development, where diverse study programs are provided at different levels, including bachelor's, master's, and doctoral degrees.

1.3 Administrative Structure of the University

The administrative structure of the university usually consists of several different components and administrative units. These components and units may vary depending on the structure and size of the university and its internal organization. Below are some of the basic components that may be included in university management plans:

1. University President:

- Responsible for the overall leadership of the university and making strategic decisions.
- Deals with policies, external relations, finance, and general guidance of the university.
- Supervises administrative and academic work at the university.

2. Vice Dean:

- Assists the dean in supervising the colleges or subsidiary colleges of the university.
- Assumes administrative responsibility for the colleges and assists in formulating academic policies and procedures.
- Collaborates with the dean in developing and evaluating academic programs and appointing faculty members.

3. Department Chair:

- Manages and supervises their academic department.
- Develops and implements academic policies within the department.
- Oversees academic planning and evaluates the performance of departmental staff.

4. Professor:

- Works as a faculty member in the academic department.
- Prepares and implements academic programs and guides students.
- Engages in academic research and publishing, reviews student papers, and manages academic projects.

1.4 Project Introduction

The project aims to revolutionize how University student absenteeism is managed in by introducing a streamlined system for submitting and processing absentee justifications. By harnessing the power of technology, the project seeks to simplify and enhance the entire process, from submission to verification, while ensuring utmost security and efficiency.

1.5 Current Absentee Justification Method

Currently, the process involves a tedious manual procedure:

1. Students must physically visit the college administration to submit their absentee justifications.
2. Verification of these justifications and determining their validity.

-
3. Upon approval, the administration manually logs the student's registration number into a pre-prepared Excel form, alongside maintaining an another Excel spreadsheet database for student records.
 4. The remaining details on the form are then auto-populated, and the approved justification is printed for the student, who then distributes it to respective professors for attendance reconciliation.

1.5.1 Drawbacks and Challenges

This manual approach poses several challenges:

1. **Time-Consuming and Complex:** The manual verification process is time-intensive, leading to delays in submission and processing.
2. **Document Vulnerability:** Paper-based justifications are prone to loss or damage during distribution, complicating tracking and management.
3. **Accuracy Concerns:** Human errors in data entry may compromise the accuracy of attendance records.
4. **Financial Implications:** Printing and distributing paper justifications incur additional costs in terms of resources.
5. **Sustainability Issues:** In the era of technological advancement, reliance on manual processes seems outdated and unsustainable.

1.6 Proposed Solution

To address these challenges, we propose a two-fold solution:

1.6.1 Desktop Management Software

A user-friendly software for administrators to:

1. **Effortlessly Manage Student Lists:** Administrators can input student data at the beginning of the academic year, streamlining subsequent processes.
2. **Swiftly Record and Manage Absences:** Absences can be recorded efficiently, eliminating the need for manual paperwork.
3. **Generate Electronic Absentee Justifications:** The system facilitates the electronic distribution of absentee justifications to professors, enhancing efficiency and reducing paperwork.

1.6.2 Student Website

An intuitive website empowering students to:

1. **Submit Absentee Justifications Digitally:** Students can submit justifications online, specifying details and attaching supporting documents seamlessly.
2. **Track Justification Status:** Real-time updates allow students to monitor the progress of their justifications, ensuring transparency.
3. **Receive Automated Notifications:** System-generated notifications keep students informed about their justification status, enhancing communication.

1.6.3 Expected Benefits

- **Optimized Efficiency:** Automation streamlines processes, boosting overall efficiency.
- **Time and Resource Savings:** Reduced reliance on manual tasks saves time and resources.
- **Enhanced Accuracy:** Digital systems minimize human errors, ensuring data precision.
- **Improved User Experience:** User-friendly interfaces enhance satisfaction for both students and administrators.
- **Transition to Electronic Documentation:** Embracing digital solutions aligns with modern trends and promotes sustainability.

Conclusion

With the expected benefits of optimized efficiency, time and resource savings, enhanced accuracy, and a transition towards The Future, the proposed system aligns with modern technological trends and promises long-term sustainability.

2. System Design

2.1 Introduction

After thoroughly studying and gaining a deeper understanding of the system, we are now ready to delve into the analysis phase. Here, our goal is to ensure the successful execution of the project. Hence, it is imperative for us to develop a precise model to provide effective guidance during programming and implementation. Throughout this stage, we will explore four diagrams that will offer us a comprehensive and insightful view of the design dynamics: **the State diagram, Sequence diagram, Class diagram, and Activity diagram.**

2.2 Use Case Diagrams

Use Case Diagrams are utilized by system administrators to comprehend the flow of services and operational units, aiming to deliver optimal construction levels.

1. General Use Case Diagram:

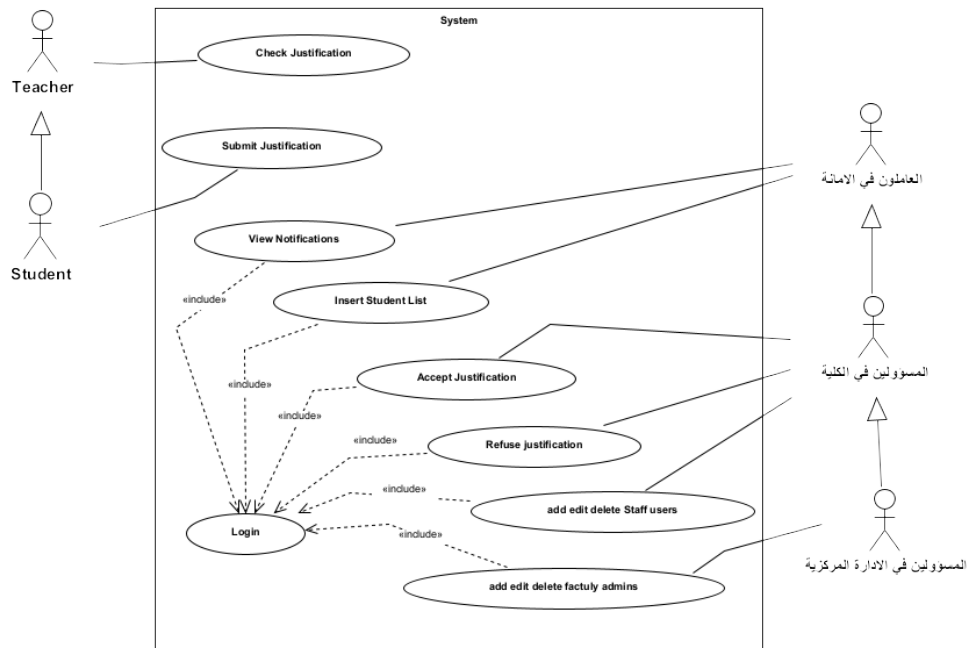


Figure 2.1: General Use Case Diagram.

2. Teacher:

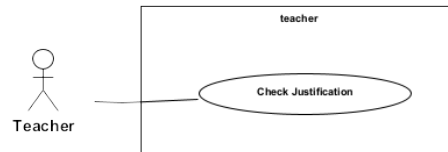


Figure 2.2: Teacher Use Case Diagram.

- Teachers can check the justification submitted by students.

3. Student :

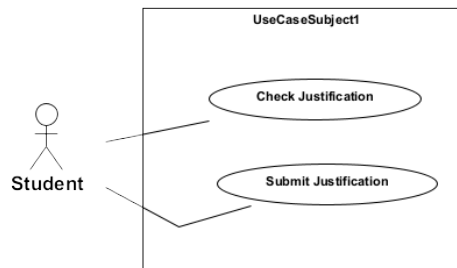


Figure 2.3: Student Use Case Diagram.

- Students can submit justifications for absences or other issues.
- Students can view their submitted justifications status.

4. Staff:

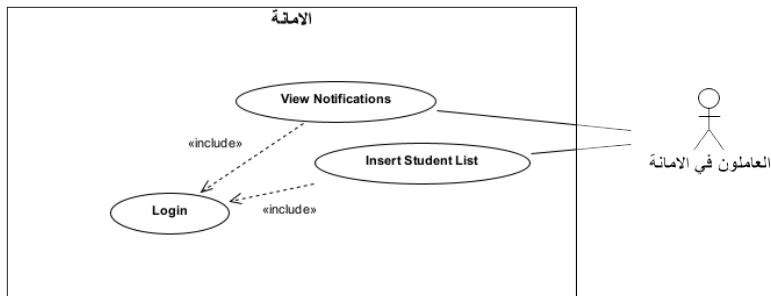


Figure 2.4: Staff Use Case Diagram.

- Staff members must log in to access their functionalities.
- Staff members can insert a list of students into the system.
- Staff members only can view the state of all the Current or preprocessed requests that submitted by the students that share the same faculty with the staff.

5. Faculty Admins:

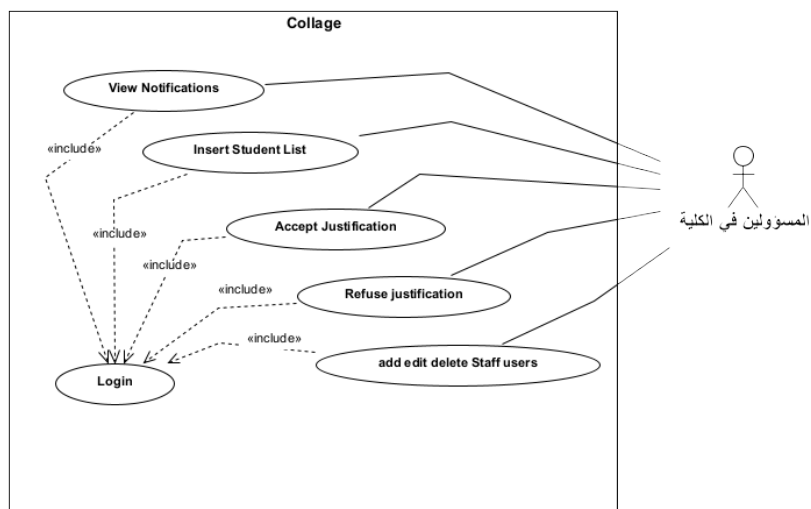


Figure 2.5: Faculty Admins Use Case Diagram.

- Faculty admins must log in to access their functionalities.
- Faculty admins have the authority to accept and refuse submitted justifications submitted by students that is related to them (same Faculty).
- Faculty admins can manage (add, edit, delete) the staff accounts that is related to them (same Faculty).

6. Central Administration (Moderators):

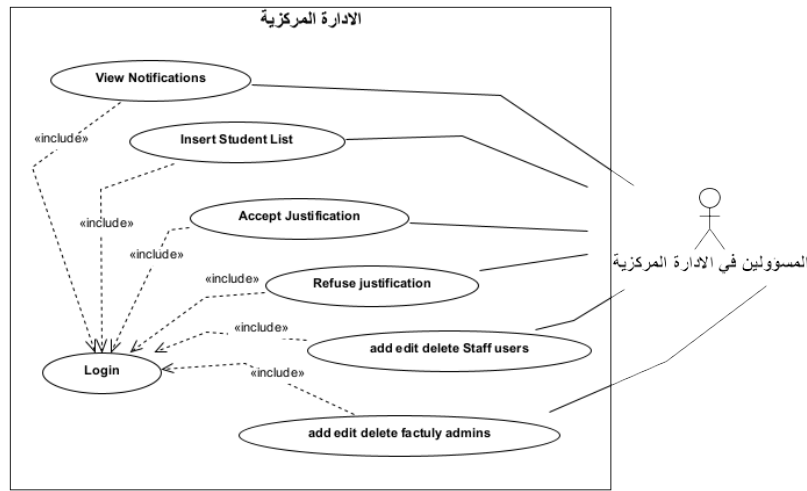


Figure 2.6: Moderators Use Case Diagram.

- Moderators must log in to access their functionalities.
- Moderators have the ability to manage (add, edit, delete) faculty admins and Staff from all the colleges accounts.
- Moderators have the authority to accept and refuse submitted justifications submitted by students from all the colleges.

2.3 Sequence Diagrams

Sequence Diagrams rely on various system components operating sequentially to execute functions, establishing connections between activities and elements.

1. Login:

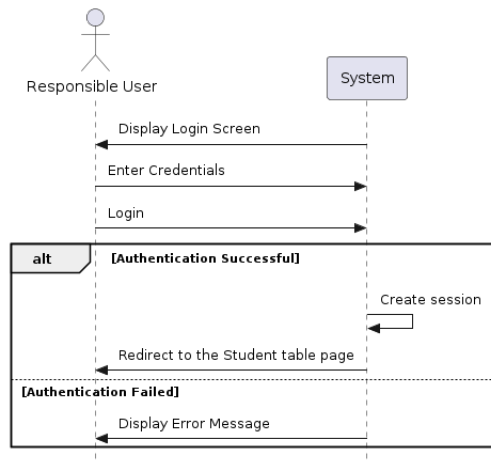


Figure 2.7: Login Sequence Diagram.

2. insert Student:

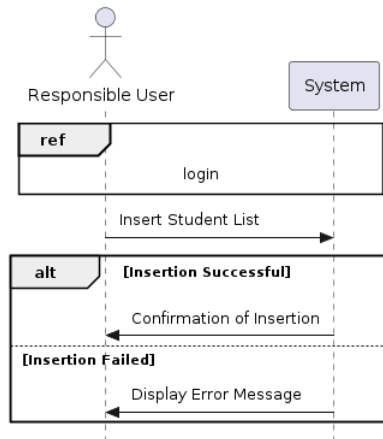


Figure 2.8: insert Student Sequence Diagram.

3. Manage Users:

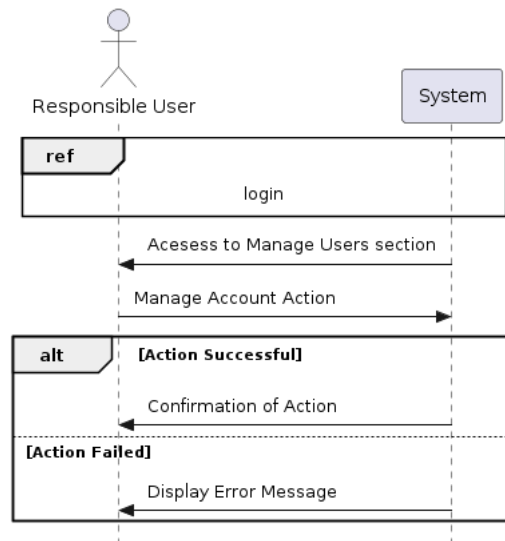


Figure 2.9: Manage Users Sequence Diagram.

4. Justification Check:

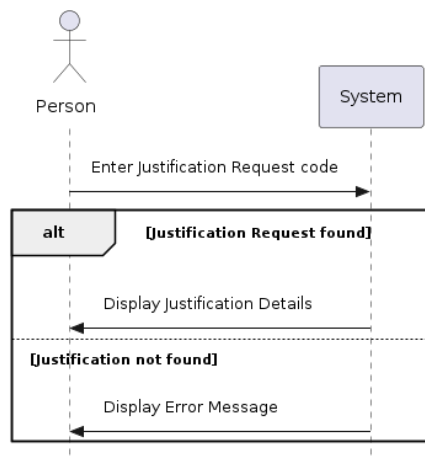


Figure 2.10: Justification Check Sequence Diagram.

5. Justification Manage:

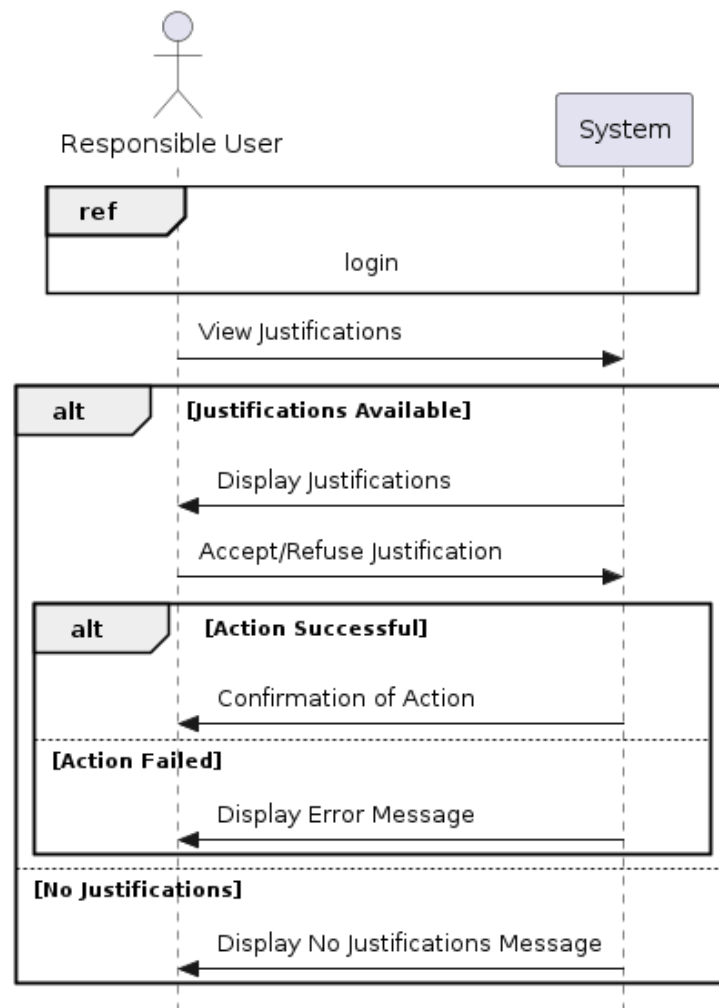


Figure 2.11: Justification Manage Sequence Diagram.

6. Justification Submit:

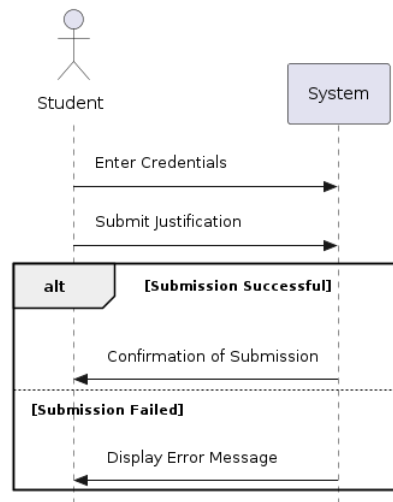


Figure 2.12: Justification Submit Sequence Diagram.

2.4 Activity diagram

It is a graphical representation that shows the steps and actions in a workflow. It's often used in software development to display how different parts of a system interact and the sequence of tasks.

1. Teacher:

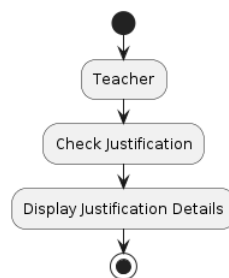


Figure 2.13: Teacher Activity Diagram.

2. Student :

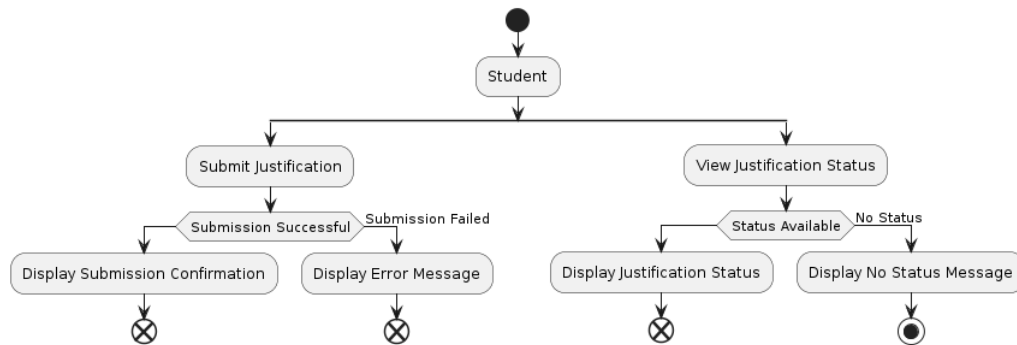


Figure 2.14: Student Activity Diagram.

3. Staff :

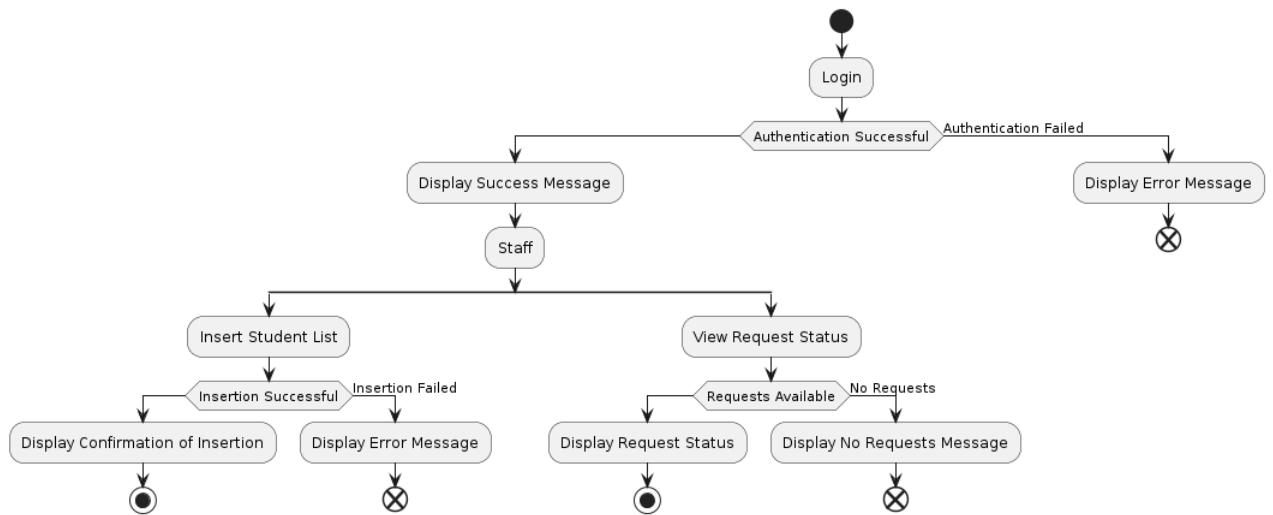


Figure 2.15: Staff Activity Diagram.

4. Faculty Admins :

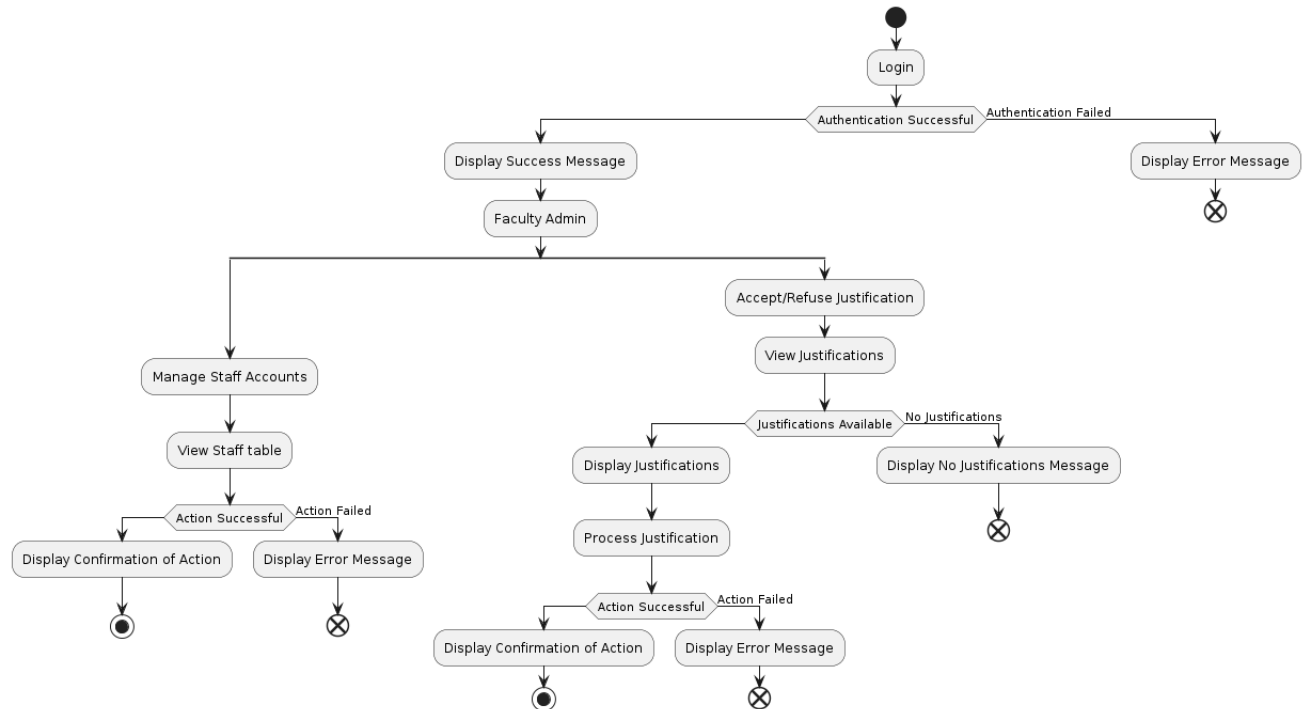


Figure 2.16: Faculty Admins Activity Diagram.

5. Central Administration (Moderators):

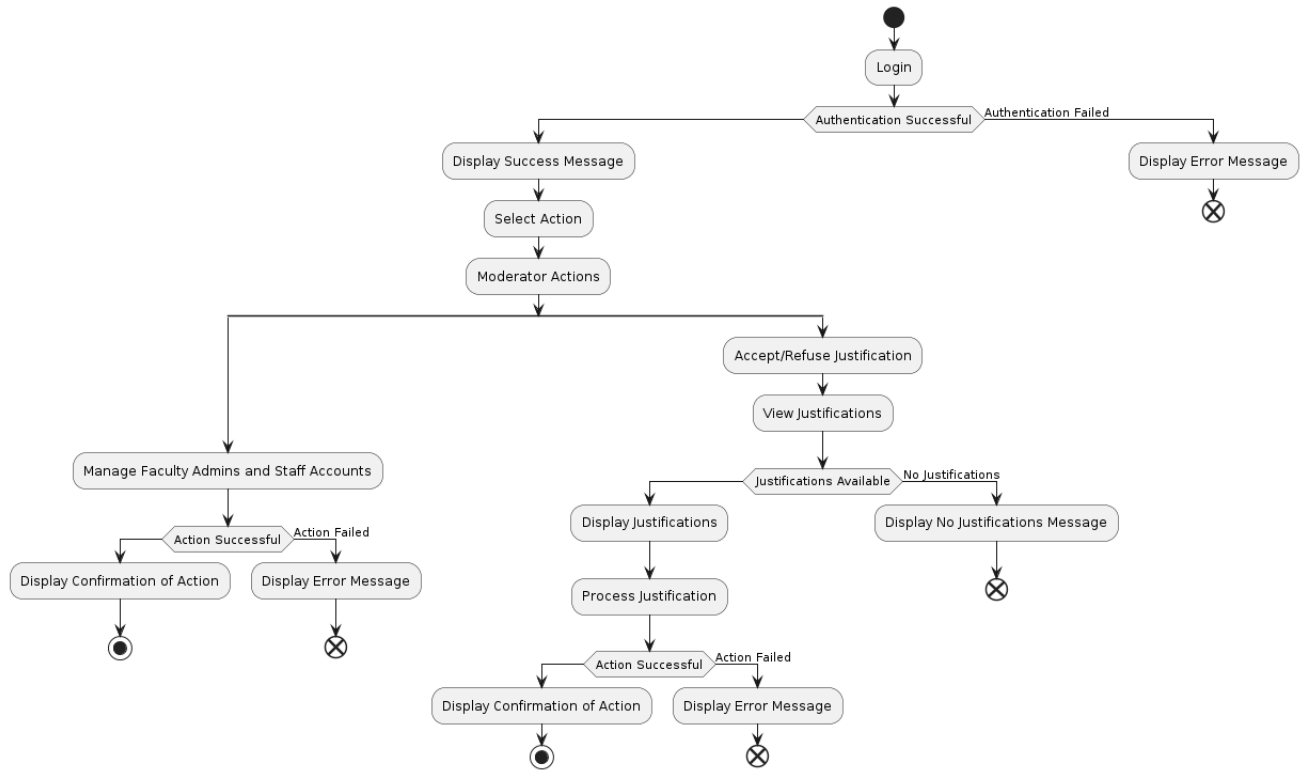


Figure 2.17: Moderators Activity Diagram.

2.5 Class Diagrams

Sequence Diagrams rely on various system components operating sequentially to execute functions, establishing connections between activities and elements.

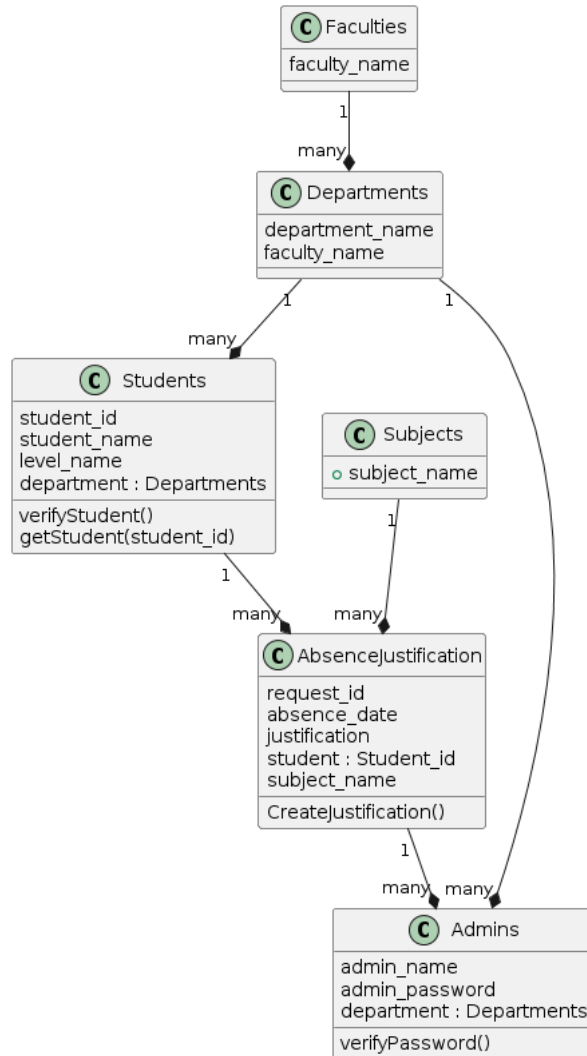


Figure 2.18: Class Diagrams

Table name	Variable name	Type
Faculties	faculty_name	String
Departments	department_name faculty_name	String String
Students	student_id student_name level_name department	Integer String String Departments
Subjects	subject_name	String
AbsenceJustification	request_id absence_date justification student subject_name	Integer Date String Students String
Admins	admin_name admin_password department	String String Departments

Table 2.1: Class Diagram Representation

Conclusion

In this chapter, we provided a detailed explanation of the system's functionality. By completing the necessary diagrams, we have developed a clear vision of the program's implementation.

3. Environment and Tools

3.1 Introduction

After completing the design and presentation of our system in the previous chapter, we now move on to the practical implementation phase of the project. This chapter presents the concepts, tools, and software used in the creation, as well as an overview of the chosen development environment for building and developing this website.

3.2 Development Environment

3.2.1 Laravel Framework

Laravel is an open-source web application framework written in PHP. It is one of the most popular PHP frameworks among web developers due to its flexibility and ease of use. It follows the MVC (Model-View-Controller) pattern.

3.2.2 Vue.js

Vue.js is an open-source JavaScript framework used for building user interfaces and single-page applications. It provides data-reactive components with a simple and flexible API.

3.2.3 Vuetify.js

Vuetify.js is a Material Design component framework for Vue.js. It helps developers create aesthetically pleasing web applications with pre-designed components following Google's Material Design guidelines.

3.2.4 Electron.js

Electron.js is an open-source framework developed by GitHub for building cross-platform desktop applications with web technologies (HTML, CSS, and JavaScript). It allows developers to build desktop applications using the same technologies used for web development.

3.3 Programming Languages Used

3.3.1 HTML

HTML is a markup language used for structuring and presenting content on the web. It is one of the oldest and most widely used languages in web design.

3.3.2 CSS

CSS (Cascading Style Sheets) is a style sheet language used to describe the presentation of a document written in HTML. It is always used alongside HTML to design and create visually appealing web pages. CSS code can be written either within the HTML file or in an external file with a `.css` extension.

3.3.3 JavaScript

JavaScript is a programming language used by developers to create interactive web pages. It is used to provide dynamic web applications, including games. As a client-side scripting language, it is a core technology of the web, with frameworks like Node.js and React.js being popular choices.

3.3.4 PHP

PHP is a server-side scripting language designed primarily for web development. It can also be used to create standalone programs. Some of its features include:

- Generating dynamic page content.
- Creating complex interactive websites like e-commerce and social networking sites.
- Opening, reading, writing, and creating files on the server.
- Adding, deleting, and modifying data in a database.

3.4 Tools Used

3.4.1 Visual Studio Code

A software that allows for the development of graphical user interfaces, scripts, web applications, android and Desktop applications. It includes a code editor with syntax highlighting, code refactoring capabilities, and a built-in debugger.

3.5 Website Presentation

- justification search (status and sender info) page :

جامعة الشهيد حقّه لخضر - الوادي
UNIVERSITY OF EL OUED

عن الجامعة الهياكل الكليات فضاء التبادل فضاء الطالب الملصقات الرقمية حاضنة الأعمال الرقمية

تقديم مدير التحقق من التقرير

البحث عن التقرير

- justi form page:

جامعة الشهيد حقّه لخضر - الوادي
UNIVERSITY OF EL OUED

عن الجامعة الهياكل الكليات فضاء التبادل فضاء الطالب الملصقات الرقمية حاضنة الأعمال الرقمية

تقديم مدير التحقق من التقرير

نموذج التقرير

اسم الطالب رقم الطالب المستوى

حفي لخضر 2123903... ▼

تخصص الطالب موج الطالب

▼ 4

اسم المحاضرة نوع التقرير

السيستم ▼

المدة تاريخ البدء تاريخ الانتهاء

21... mm / dd / yyyy mm / dd / yyyy

- Student Management page:

The screenshot shows the 'Students' management page. The left sidebar contains links for Dashboard, Notifications, Students (active), User Management, and Print Document. The main content area features a search bar, an 'INSERT STUDENTS LIST' button, and a table of students.

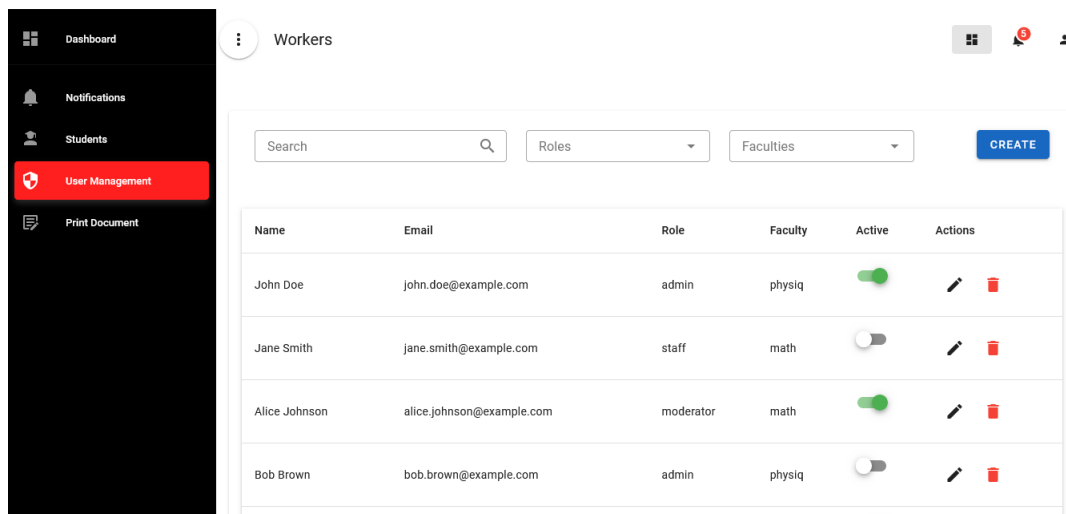
Id	Name	Department	Actions
1	John Doe	12345	
2	Jane Smith	67890	
3	Alice Johnson	54321	
4	Bob Brown	09876	

- Notification page for new justifications request page:

The screenshot shows the 'Notifications' page. The left sidebar contains links for Dashboard, Notifications (active), Students, User Management, and Print Document. The main content area features a search bar, 'FILTER' and 'UPDATE' buttons, and a table of notifications.

Name	Registration Number	Date Sent	Actions
John Doe	12345	2023-05-01 10:00	
Jane Smith	67890	2023-05-02 11:00	
Alice Johnson	54321	2023-05-03 12:00	
Bob Brown	09876	2023-05-04 13:00	
Emily Davis	13579	2023-05-05 14:00	
Michael Wilson	24680	2023-05-06 15:00	

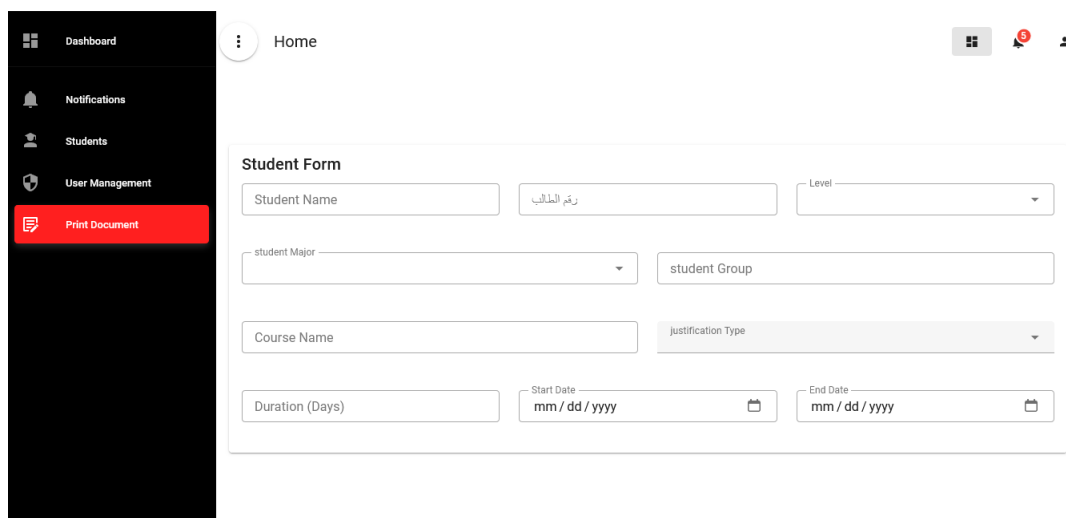
- User Management page:



The screenshot shows the 'User Management' page. The sidebar on the left contains links for Dashboard, Notifications, Students, User Management (highlighted in red), and Print Document. The main content area is titled 'Workers' and features a search bar, dropdowns for Roles and Faculties, and a 'CREATE' button. Below these is a table with columns: Name, Email, Role, Faculty, Active, and Actions.

Name	Email	Role	Faculty	Active	Actions
John Doe	john.doe@example.com	admin	physiq	☒	Edit Delete
Jane Smith	jane.smith@example.com	staff	math	☐	Edit Delete
Alice Johnson	alice.johnson@example.com	moderator	math	☒	Edit Delete
Bob Brown	bob.brown@example.com	admin	physiq	☐	Edit Delete

- Local Treatment for justification page:



The screenshot shows the 'Local Treatment for justification' page. The sidebar on the left contains links for Dashboard, Notifications, Students, User Management, and Print Document (highlighted in red). The main content area is titled 'Home' and features a 'Student Form' with various input fields.

Student Form

Student Name رقم الطالب Level

student Major student Group

Course Name Justification Type

Duration (Days) Start Date End Date

Conclusion

Concluding the Project on University Student Attendance Justification Management

- We endeavored to create a website tailored to our university's needs by designing and implementing effective Attendance Justification Management. This involved compiling Justification forms in a database and presenting them in an accessible format, along with features like easy Justification code search.
- Starting with a thorough analysis of project requirements and information, we proceeded to design the website using UML diagrams (state, sequence, class, and activity diagrams). This structured approach ensured clarity and alignment with project goals.
- In the final phase, we successfully implemented the website, showcasing various system aspects through completed designs and functional pages. This marked the culmination of our efforts in realizing a comprehensive timetable management solution for our university.
- Looking forward, areas for future enhancement include further refining user interfaces based on feedback, integrating additional functionalities for enhanced user experience, and exploring potential optimizations in database management and system scalability.

Bibliography

- [1] Laravel docs. <https://laravel.com/docs/11.x>.
- [2] Vue docs. <https://vuejs.org/guide>.
- [3] Vuetify docs. <https://vuetifyjs.com/getting-started/installation>.
- [4] Electron vite docs. <https://electron-vite.org/guide/>.
- [5] Html. <https://wikipedia.org/wiki/html>.
- [6] Css. <https://naqrah.net/what-is-css>.
- [7] Javascript. <https://aws.amazon.com/what-is/javascript>.
- [8] Php. <https://wikipedia.org/wiki/php>.
- [9] Visual studio code docs. <https://code.visualstudio.com/docs>.