



REPUBLIC OF BENIN
MINISTRY OF HIGHER EDUCATION AND
SCIENTIFIC RESEARCH

UNIVERSITY OF ABOMEY-CALAVI
INSTITUTE OF TRAINING AND
RESEARCH IN COMPUTER SCIENCE

BP 526 Cotonou Tel : +229 21 14 19 88
<http://www.ifri-uac.net> Courriel : contact@ifri.uac.bj



THESIS

for obtaining a

Bachelor degree in Computer Science

Option : Software Engineering

Presented by :

Jordy Florien Venceslas GNANIH

Implementation of a Web Application for University Timetable Management: A Case Study at IFRI

Under the supervision of :

Dr. Eng. Vinasetan Ratheil HOUNDJI
Lecturer at IFRI,UAC

Examination Committee

Amadou Tidjani SANDA MAHAMA	Doctor (MA)	IMSP	President
Honorine ADJIBI DEDJINOU	Engineer	IFRI	Examiner
Ratheil Vinasetan HOUNDJI	Doctor (MA)	IFRI	Secretary

Academic Year : 2022-2023

Overview

Dedication	ii
Acknowledgements	iii
Abstract	iv
List of figures	vi
List of tables	vii
Acronyms and Abbreviations	viii
Glossary	ix
1 Literature review	3
2 Conception and technical choices	16
3 Results and discussion	26
Bibliography	41
Webography	42
French Summary	44
Contents	49

Dedication

To

My father, Florent **GNANIH**

My mother, Marguerite **TOGBE**

Acknowledgements

As a sign of my deep gratitude, I would like to express my sincere thanks to all the people who contributed, directly or indirectly, to the success of my final year internship and to the preparation of this modest work.

First of all, I would like to thank Professor **Eugène C. EZIN**, Director of the Institute of Training and Research in Computer Science (IFRI), for his sound advice and the invaluable training he gave us.

I would like to acknowledge Dr. Ing. **Vinasetan Ratheil HOUNDJI** for the quality of his teaching, his invaluable advice and his undeniable interest in all students.

I would like to express my deep gratitude to the entire IFRI faculty for the considerable work they do to provide us with the best possible learning conditions.

Although, it's impossible to mention all the names, my thanks go to all the friends and colleagues who, from near and far, have contributed to the production of this dissertation through their advice and expertise.

Abstract

Managing timetables in academic entities is a complex task, requiring meticulous organization to avoid conflicts and ensure consistent allocation of resources. At the Institute of Training and Research in Computer Science (IFRI), this responsibility falls to the coordinator who, despite his efforts, faces recurring difficulties such as the multiple allocation of classrooms or courses to professors at the same time, causing conflicts that disrupt the organization of courses. A solution to improve this management and minimize errors is therefore crucial for the institute. With this in mind, we offer a web-based application, [SchedU](#), for efficient timetable management, aimed at resolving problems of allocation conflicts while optimizing repetitive planning tasks. Our platform enables resources such as rooms, teachers, and courses to be managed seamlessly. Thanks to an automatic conflict detection system, the application warns users in the event of double allocation and offers the possibility of correcting errors before validation. In addition, it enables class managers to consult and confirm lessons while automatically calculating remaining time masses, simplifying the monitoring of completed lesson hours. The user-friendly interface allows for easy navigation, ensuring that even those with minimal technical expertise can manage schedules effectively. By implementing SchedU, IFRI aims to streamline its scheduling processes, enhance communication among faculty members, and ultimately foster a more organized educational environment that benefits both students and instructors. This innovative approach not only saves time but also enhances the overall efficiency of the academic operations at IFRI, paving the way for improved educational outcomes.

Mots clés: SchedU - Laravel - Vue JS - API - MYSQL

Résumé

La gestion des emplois du temps dans les entités académiques est une tâche complexe, qui nécessite une organisation minutieuse pour éviter les conflits et assurer une allocation cohérente des ressources. A l'Institut de Formation et de Recherche en Informatique (IFRI), cette responsabilité incombe au coordinateur qui, malgré ses efforts, est confronté à des difficultés récurrentes telles que l'attribution multiple de salles ou de cours à des professeurs en même temps, provoquant des conflits qui perturbent l'organisation des cours. Une solution pour améliorer cette gestion et minimiser les erreurs est donc cruciale pour l'institut. Dans cette optique, nous proposons une application web, SchedU, pour une gestion efficace des emplois du temps, visant à résoudre les problèmes de conflits d'attribution tout en optimisant les tâches répétitives de planification. Notre plateforme permet de gérer de manière transparente les ressources telles que les salles, les enseignants et les cours. Grâce à un système de détection automatique des conflits, l'application avertit les utilisateurs en cas de double affectation et offre la possibilité de corriger les erreurs avant validation. De plus, elle permet aux responsables de classe de consulter et de confirmer les cours tout en calculant automatiquement les masses horaires restantes, ce qui simplifie le suivi des heures de cours effectuées. L'interface conviviale permet une navigation aisée, garantissant que même les personnes disposant d'une expertise technique minimale peuvent gérer les emplois du temps de manière efficace. En mettant en œuvre SchedU, l'IFRI vise à rationaliser ses processus de planification, à améliorer la communication entre les membres du corps professoral et, en fin de compte, à améliorer la qualité de l'enseignement.

Key words : SchedU - Laravel - Vue JS - API - MYSQL

List of figures

1.1	Trello Interface	7
1.2	Asana Interface	8
1.3	Google Calendar Interface	8
1.4	ClickUp interface	9
1.5	TimeTabler interface	10
2.1	Use Case Diagram	18
2.2	Class Diagram	20
2.3	Sequence Diagram for creating a Timetable	22
2.4	System architecture	24
3.1	Coordinator login	27
3.2	Class manager login	27
3.3	Year Choosing	28
3.4	Filiere Choosing	28
3.5	Weekly Course Calendar	29
3.6	Course Modal Creation	29
3.7	A course in a Calendar	30
3.8	Full week Calendar	30
3.9	Server response	31
3.10	Class manager dashboard	31
3.11	Class manager academic year confirmation	32
3.12	Choose class branch	32
3.13	Class manager EC list	33
3.14	Enter number of hour for a course	33
3.15	Consult timetable	34
3.16	List of scheduled course weeks	34
3.17	Course displaying in a camendar	35
3.18	Global view of a timetable	35
3.19	Configuration view	36
3.20	Unit Tests	36
3.21	laravel middleware set up	38
	newpage	

List of tables

1.1

Resume of comparison of existing solution

14

2.1

Technical Choices and Alternatives

23

Acronyms and Abbreviations

API :

Application Programming Interfaces [24](#),

EC :

Component element

HTTP :

Hypertext Transfer Protocol [24](#),

IFRI :

Institut de Formation et de Recherche en Informatique [3](#), [6](#), [16](#), [25](#), [39](#), [49](#),

L1 :

Licence 1 [6](#),

L2 :

Licence 2 [6](#),

L3 :

Licence 3 [6](#),

UE :

Teaching Unit [21](#),

UML :

Unified Modeling Language

Glossary

- Backend :** The backend is the server-side part of an application that handles the logic, database interactions, and server configurations. It processes requests, manages data, and communicates with the frontend. [ix](#),
- CSS :** CSS is a styling language used to define the appearance and layout of a web page. It allows developers to control the color, font, spacing, alignment, and other visual aspects of HTML elements. CSS can be used with HTML to enhance the design and user experience of web applications.
- Front - End :** The frontend is the client-side part of an application that users interact with directly. It involves designing and implementing the visual aspects and user experience of a website or application, including layouts, buttons, forms, and other elements. [ix](#),
- SchedU :** A web application used to settle time table in school [iv](#), [1](#), [16](#), [25](#)
- Timetable :** A timetable is a schedule that organizes activities, events, or tasks by specifying when they should occur. It is often used in educational institutions, workplaces, or public transportation systems to plan classes, meetings, or trips at designated time. [ix](#),

General Introduction

Efficient management of timetables is essential to ensure the smooth running of academic activities in institutions. At IFRI, conflicts over the allocation of rooms or courses are frequently encountered, affecting the organization of courses.

1. Context and justification

The timetable, as a fundamental tool in the organization of courses, plays an important role in maximizing productivity, and ensuring a healthy balance between different responsibilities. Effective management of timetables is essential to ensure the smooth running of academic activities, especially in university settings. The timetable generally includes information such as date, time, teachers and subjects. The manual elaboration of this task requires a great deal of time and attention to avoid conflicts with classrooms or teachers. Faced with this reality, it is crucial to implement a solution to facilitate this process. It is in this context that we are introducing our solution, ShedU, which aims to facilitate the process of drawing up timetables for lesson planning. Focusing on IFRI, we'll concentrate our study on the tasks involved in planning a timetable, including allocating courses, teachers, classrooms and calculating time masses. [SchedU](#) is a web-based platform designed to improve the efficiency of various aspects of academic program planning, notably by reducing time wastage and minimizing conflicts.

2. Problematic

As part of the organization of academic activities within universities, the development of timetables is a crucial but complex issue. Faced with the diversity of courses and limited resources, the traditional manual method of planning is showing its limitations. This situation does not spare IFRI, which is often confronted with difficulties such as human error, wasted time and stress for administrators. To deal with this situation, we have developed a web application designed to simplify time management.

3. Objectives

General objective

Facilitate the management of IFRI's timetables, by optimizing the organization of courses, avoiding Scheduling conflicts, and improving communication between teachers, supervisors and staff.

Specific objectives

- Facilitate the creation of timetables by the competent authorities
- Minimize scheduling conflicts
- Publish schedules for institute members after approval by the relevant authorities
- Ensure consistency and planning of subjects in the creation of timetables

4. Structure of document

The structure of the document is as follows :

- The first chapter begins with conceptual clarification for a clear understanding of our work.
- The second chapter focuses on conceptual analysis and technical choices. We have used UML to model the system architecture, including use case, class, sequence and activity diagrams to illustrate the interactions and functionalities of our application. This chapter also explains the technologies chosen for the development of our application.
- We conclude by presenting the results obtained and the challenges encountered during development.

Literature review

Introduction

Managing timetables - whether for courses, exams or, in our case, the planning of academic activities at [IFRI](#) - is a major challenge for higher education establishments. Over the years, numerous technological solutions have emerged to facilitate this complex task, offering essential support to various players such as teachers, administrators and students. It is therefore necessary to understand these different approaches and the previous work carried out in this field. The aim of this literature review is to critically evaluate existing solutions, highlighting their advantages and limitations, as well as the lessons that can be learned to improve timetable management.

1.1 Timetable

A timetable is an essential planning document or tool that organizes and distributes the activities of an individual or group over a given period (day, week, month, semester, etc.). It is used to indicate not only when these activities will take place, but also where and with whom they will take place. The main aim is to optimize the management of time and resources, to ensure that the various tasks and events run smoothly and without conflict.

An efficient timetable needs to take several factors into account:

- Planned activities: courses, meetings, events, exams, etc.
- Who benefits: Students, teachers, employees, or specific groups.
- Locations: classrooms, laboratories, lecture halls, offices, or any other space required for the activity.
- Resources: Tools or equipment required (computers, projectors, special equipment).
- Constraints : Availability of teachers, students, rooms, etc.

1.2 Overview of TimeTable

A timetable, beyond its simple function of organizing activities according to time, has many applications in various contexts, both individual and collective. It can be used to manage time more efficiently, allocate resources and coordinate complex tasks. Here's a detailed presentation of the different uses of a timetable.

1.2.1 Personal organization and management

In everyday life, a personal timetable is an essential tool for organizing activities and managing time. It enables you to :

- Plan daily tasks: This includes professional tasks (work, meetings) and personal tasks (exercise, hobbies, shopping).
- Increase productivity: Having a structured schedule helps you stay focused and get things done on time.
- Reduce stress and procrastination: Having an overview of the tasks at hand makes it easier to manage time and reduce the stress associated with the unexpected.
- Keeping track of priorities: A timetable allows you to prioritize activities according to their importance and deadlines.

1.2.2 School and university planning

In the educational sector, timetabling is a crucial tool on several levels: for pupils, students, teachers and administrators. Here are a few specific examples of uses in this context:

- Distribution of courses and examinations: In educational establishments, timetables are used to distribute courses, laboratories, seminars and examinations over the whole semester or year. They must take into account the availability of teachers, students and classrooms.
- Distribution of courses and examinations: In educational establishments, timetables are used to distribute courses, laboratories, seminars and examinations over the whole semester or year. They must take into account the availability of teachers, students and classrooms.
- Teacher coordination: The timetable allows you to synchronize the activities of a large number of teachers, allocating their teaching hours, their availability for consultations and administrative tasks.
- Material resource management: The allocation of resources such as classrooms, laboratories, multimedia equipment and other infrastructure is a key component of a university timetable.
- Flexibility and adjustment: Modern university timetables often include real-time adjustment possibilities.

1.2.3 Company planning

In the professional world, scheduling is just as essential to task management, employee productivity and resource optimization. Here's how it's used in the workplace:

- **Project management:** Companies use schedules to plan the stages of a project, assign tasks to different employees or teams, and set deadlines.
- **Team coordination:** In large companies, it's crucial to synchronize team schedules to avoid scheduling conflicts, especially in environments where projects require the collaboration of several departments.
- **Human resources planning:** In certain sectors, such as catering or industrial production, the timetable is used to plan employees' work schedules.
- **Appointment scheduling and customer management:** Schedules are widely used in the health-care, consulting and service sectors to manage customer and patient appointments.

1.3 Types of timetable

The timetable is an essential planning tool in various areas of daily and professional life. There are several types of timetable, each meeting specific needs and used in different contexts. Here's a detailed presentation of the main types of timetable and their characteristics.

1.3.1 Personal Timetable

The Personal Timetable is designed to help individuals organize their daily, weekly or monthly activities according to their priorities and objectives. It allows you to structure your days to balance work, leisure and responsibilities. Its aim is to maximize personal time management to improve productivity, maintain work-life balance, and reduce the stress associated with poor organization.

1.3.2 School timetable

The school timetable is commonly used in schools, colleges and universities. Its purpose is to organize and distribute classes, teaching activities and examinations for students and teachers throughout the school year. Its aim is to facilitate the management of courses, timetables and classrooms, to avoid overlaps and ensure the smooth running of teaching.

1.3.3 University timetable

University timetables are similar to school timetables, but they present additional complexities due to the greater number of teachers, students, courses and resources to be managed. They are used to plan lectures, seminars, tutorials, practical work, examinations and consultations between students and teachers. The aim is to organize all of a university's teaching activities, taking into account the different departments, varied programs and limited resources (classrooms, lecture theaters, laboratories).

1.4 Importance and benefits of a timetable

The timetable is a fundamental tool in time management, whether for personal, school, university or professional use. It helps to structure activities, prioritize and better coordinate collective efforts. Here's an in-depth look at the importance and many benefits of using a timetable:

- Time management improvement
- Increased productivity
- Better organization of resources

1.5 Timetable planning at IFRI

At [IFRI](#), timetable management is based on a weekly process. The coordinator is responsible for gathering the information needed for planning, such as the availability of professors, the classrooms available, the courses to be taught and the time masses associated with each subject. Based on this data, he or she draws up timetables for each academic level, in particular for [L1](#), [L2](#) and [L3](#) students. This process ensures that courses are correctly distributed, while taking into account the logistical and pedagogical constraints of the institution. It does, however, present a number of problems.

Indeed, the process of creating schedules at [IFRI](#) is entirely manual, making it particularly tedious and error-prone. This approach can lead to overlapping schedules or conflicts over classroom availability. What's more, the same process is repeated every week for all three academic years (Licence 1, Licence 2 and Licence 3), making the task even more cumbersome. Another limitation relates to the calculation of in-progress course timetables, which is not automated, and to the absence of a summary table or overall statistical monitoring of subject evolution by year ([L1](#), [L2](#), [L3](#)). Finally, the publication of timetables to class leaders and teachers is done manually, often via channels such as WhatsApp or e-mail. This method becomes particularly time-consuming when the timetables have to be sent to several teachers simultaneously.

To overcome the drawbacks of the traditional method of scheduling, several solutions have been put in place, which we will detail in the next section.

1.6 Overview of existing solutions

At this level, we carry out a comparative study of a number of time management solutions. Our study is based on the functionalities, advantages and limitations of each of these solutions. The applications selected for our study are as follows:

1.6.1 Trello

Trello is a visual project management tool based on the Kanban board system, offering a simple and flexible way of organizing individual or collaborative tasks. Each project is represented in tabular form, divided into lists representing phases or categories, and cards representing specific tasks. Cards can be enriched with details, due dates, labels and files to help manage priorities. Trello is highly appreciated for its ease of use and its ability to visualize the progress of a project at a glance. The application also lets you collaborate with teams, assign tasks and track project progress in real time. Thanks to its integrations with other popular tools, such as Google Drive and Slack, and its automation features with Butler, Trello is an ideal choice for individuals and teams who want a simple, visual solution for organizing their tasks.

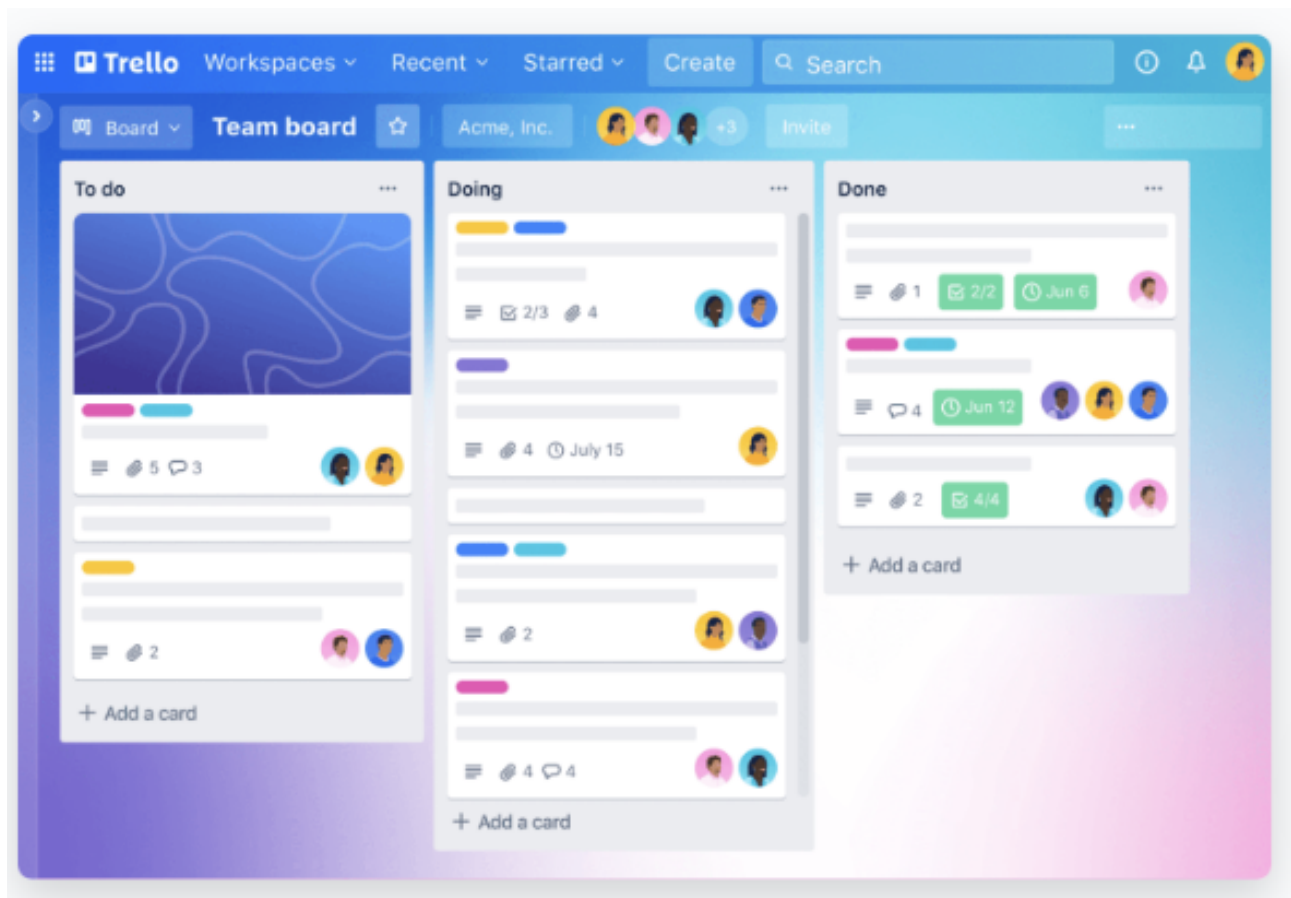


Figure 1.1: Trello Interface

1.6.2 Asana

Asana is a robust project management application that helps teams coordinate their efforts, stay organized and achieve their goals more efficiently. It enables users to create tasks, assign them to team members, set deadlines and track work progress in a structured way. Asana offers several visualization modes, including lists, Kanban-style boards, calendars, and Gantt charts, allowing teams to choose the view best suited to their working style. In addition to task management, Asana lets you define long-term goals and track their progress through linked tasks. Thanks to its integration with various external tools and its ease of use, Asana is particularly useful for medium-sized to large teams managing complex projects involving multiple stakeholders.



Figure 1.2: Asana Interface

1.6.3 Google Calendar

Google Calendar is a free online calendar management application developed by Google. It lets you create, organize and share events, appointments and schedules. Accessible on the web and via mobile applications, Google Calendar offers features such as the creation of recurring events, the sending of notifications and reminders, and integration with other Google tools (such as Gmail and Google Meet). It is also possible to share calendars with other users to facilitate collaboration and group planning.

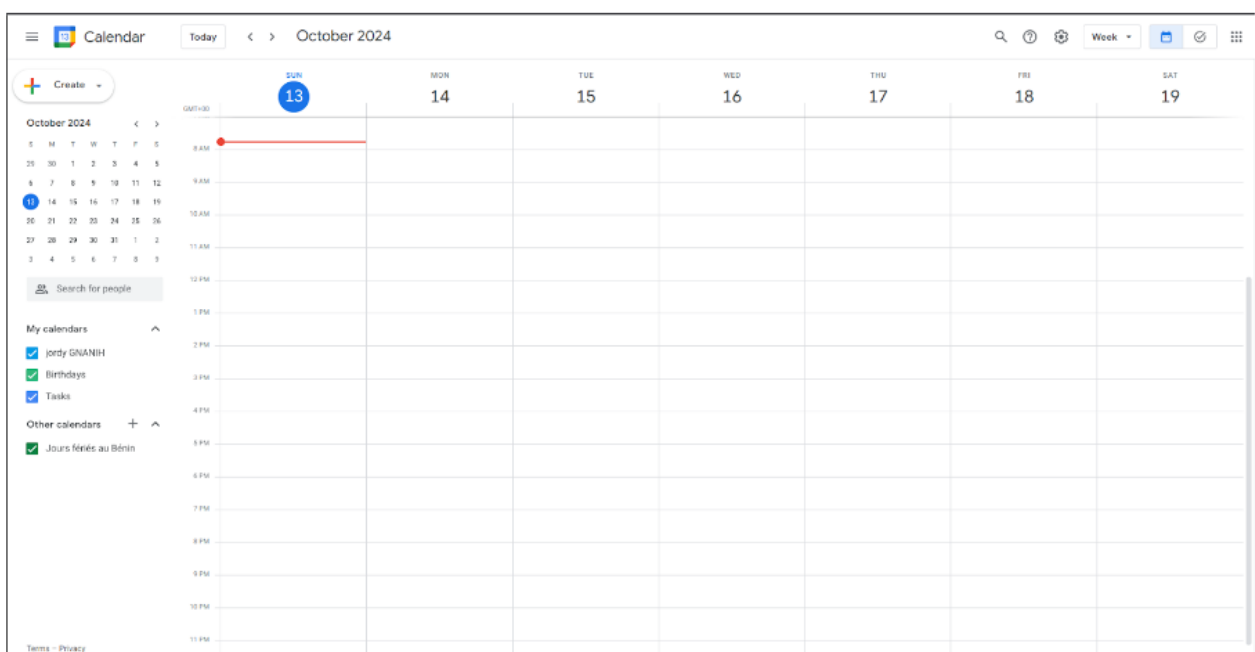


Figure 1.3: Google Calendar Interface

1.6.4 ClickUp

ClickUp is a comprehensive project management platform offering a wide range of features to meet the needs of teams and individuals looking to optimize their organization and productivity. It offers several views for task management, such as Kanban boards, lists, calendars and Gantt charts, providing flexibility in the way work is tracked and organized. ClickUp not only lets you manage tasks, but also create automations to simplify repetitive workflows. What's more, the application tracks the time spent on each task, as well as providing tools for measuring performance and managing objectives. Thanks to its numerous integrations with popular tools such as Slack, Google Drive and Zapier, and its real-time collaboration features, ClickUp is a preferred choice for companies that need a comprehensive, customizable and efficient tool to manage large-scale, complex projects.

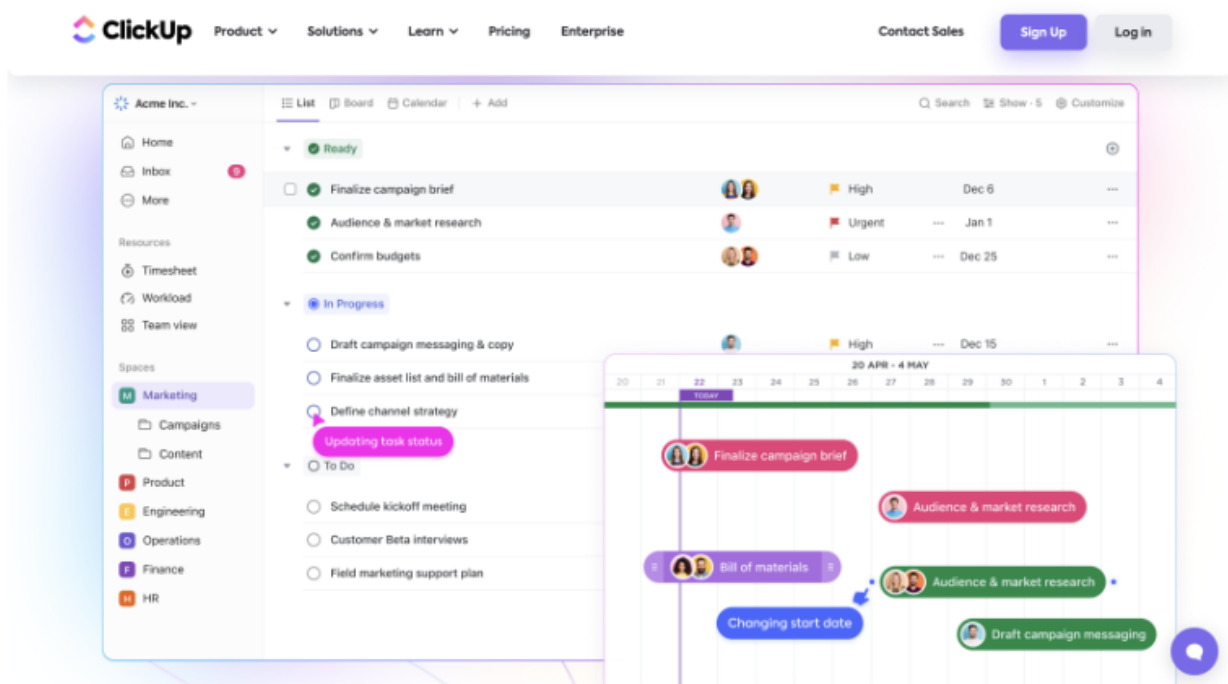


Figure 1.4: ClickUp interface

1.6.5 TimeTabler

Timetabler is a software package designed specifically for creating and managing school timetables. It offers several features to facilitate planning and the resolution of scheduling conflicts.

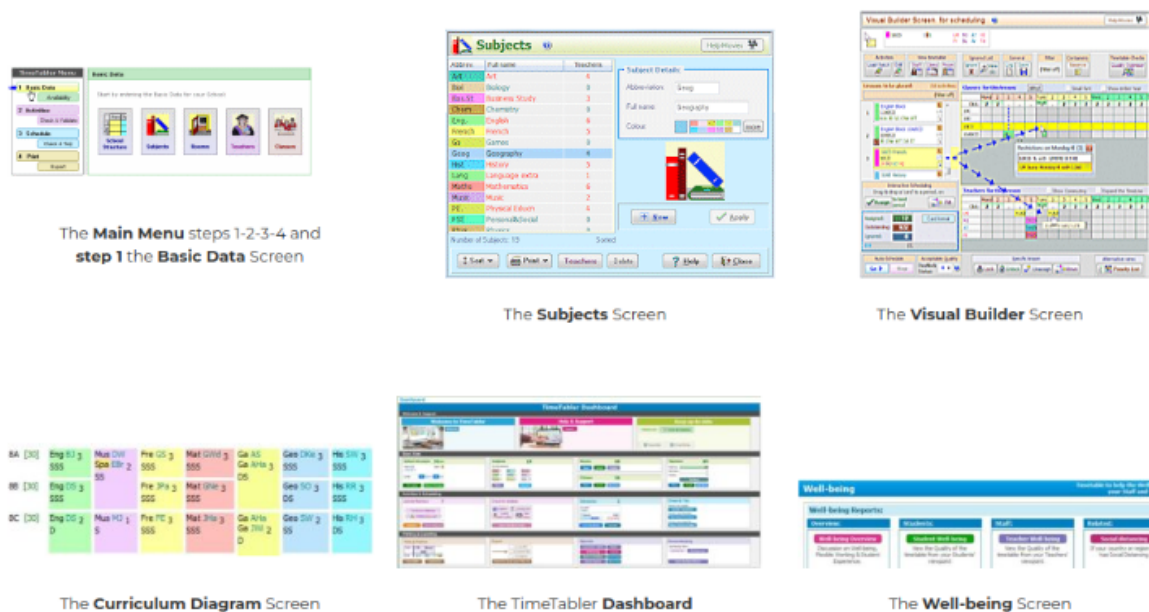


Figure 1.5: TimeTabler interface

1.7 Comparison of existing solutions

In our exploration of project management and scheduling solutions, we compared several options, including Trello, Asana, Google Calendar, ClickUp, and Timetabler, evaluating their key features, benefits, and limitations.

1.7.1 Trello

Functionalities:

- Project management based on Kanban boards.
- Cards to manage tools such as Google Drive, Slack, etc.
- Automation functions via Butler (for repetitive actions). tasks, assign members, add deadlines and labels.
- Integration with

Benefits:

- Simple, intuitive interface, easy to use for beginners.
- Perfect for small teams and simple projects.
- Clear visuals thanks to its Kanban approach.
- Fully functional free version.

Limits:

- Lack of monitoring of resource availability
- Lack of conflict management
- Inadequat for schools and universities

1.7.2 Asana

Functionalities:

- Project management with multiple views (Kanban board, list, calendar)
- Manage tasks, subtasks, deadlines, priorities, and assign responsibilities.
- Track project progress with reporting views.
- Automations for recurring tasks.

Benefits:

- Good management of task dependencies
- Efficient collaboration for medium-sized to large teams.
- Monitor project progress and objectives using dashboards and visual reports.

Limits:

- Complexity of the interface for new users compared to Trello.
- Some important features (such as advanced automation and complex dependencies) are reserved for paid versions.
- Can become cumbersome for simple projects
- Inadequat for schools and universities

1.7.3 Google Calendar

Functionalities:

- Support for the creation of recurring events
- Calendar sharing
- Colors and categories
- Notifications and reminders
- Integrations
- Cross-platform access

Benefits:

- Free and accessible
- Ease of use
- Real-time collaboration
- Personalization

Limits:

- Lack of specialized features
- Limited conflict management
- Limited visualization for large schedules
- Limited automation
- Data monitoring and analysis
- Inadequat for schools and universities

1.7.4 ClickUp

Functionalities:

- All-in-one project management with multiple views (Kanban, list, Gantt, calendar).
- Tasks, subtasks, task dependencies and resource management.
- Advanced automation and workflow customization
- Collaboration tools: comments, discussions, notifications.
- Detailed reports, dashboards, time tracking.

Benefits:

- Highly complete and flexible for both small and large teams, with extensive functionality.
- Combines several types of management (Kanban, Gantt, calendar, etc.) in a single tool.
- Robust dependency management, goal tracking and automation.
- Good value for money, with a free version offering multiple options.

Limits:

- The interface can seem complex and overloaded for users who need a simple solution.
- Higher learning curve compared to Trello or Asana.
- May be overkill for small projects or teams.
- Inadequat for schools and universities

1.7.5 Timetabler

Functionalities:

- Automated schedule creation.
- Conflict management.
- Customized constraints.
- Track hours taught
- Reports and visualization.

Benefits:

- Efficient automation
- Advanced conflict management
- Suitable for schools and universities
- Detailed reporting

Limits:

- Complexity of user
- Cost
- Dependence on local environment

Comparative table of existing solutions.

Table 1.1: Resume of comparison of existing solution

Critères	Trello	Asana	Google Calendar	ClickUp	Timetabler
Avantages	- Interface visuelle simple et intuitive - Flexibilité dans l'organisation des tâches - Bon pour la collaboration en temps réel	- Gestion claire des tâches et projets - Notifications et rappels personnalisables - Suivi des charges de travail	- Création d'événements récurrents - Partage facile de calendriers - Intégration avec d'autres outils Google	- Vues multiples (liste, calendrier, Gantt) - Gestion des dépendances et priorités - Personnalisation avancée des flux de travail	- Algorithme de gestion automatique des conflits - Personnalisation des matières et niveaux - Rapports détaillés sur les charges horaires
Limites	- Pas de gestion automatique des conflits - Limité pour les projets complexes - Manque de fonctionnalités spécialisées pour l'éducation	- Courbe d'apprentissage plus élevée - Moins adapté pour des horaires complexes - Gestion des ressources limitée	- Pas de gestion automatique des conflits - Visualisation encombrée pour de grands emplois du temps - Fonctionnalités limitées pour les établissements scolaires	- Peut être trop complexe pour les petites équipes - Configuration initiale peut être chronophage - Coût potentiel pour des fonctionnalités avancées	- Principalement conçu pour l'éducation, donc moins flexible pour d'autres usages - Peut être coûteux pour certaines institutions - Limité en matière d'intégration avec d'autres outils

1.8 Our solution

Faced with a variety of available solutions that didn't fully meet our needs, we recognized the necessity for a tailored approach. After thorough analysis of existing tools, we identified their strengths and limitations, which inspired us to design our own solution. This led to the development of the SchedU web application, a comprehensive platform that integrates the most effective functionalities of existing solutions while addressing their shortcomings.

SchedU aims to streamline project management and scheduling processes, enhancing user experience and improving efficiency. By focusing on user feedback and iterative design, we ensured that SchedU not only meets our specific requirements but also adapts to the evolving needs of its users. This application represents a significant advancement in our quest for an optimal solution, and it is the central focus of this thesis.

Conclusion

As part of the development of our web application, we undertook a process that began with an analysis of the requirements and constraints associated with the project. This was followed by project modeling, and then the actual design phase. Along the way, a number of technical choices were made, which are described in detail in the next chapter. We will now present our solution, highlighting its architecture and the tools we chose to develop it.

Conception and technical choices

Introduction

IFRI has to take various factors into account when drawing up its timetables, such as the availability of classrooms and time masses. However, this task is becoming increasingly complex due to the limitations of academic resources. In addition, human error can lead to scheduling conflicts, making the process even more delicate. To simplify this process and minimize conflicts, we have developed a solution called [SchedU](#). In this chapter, we will detail the steps involved in modeling [SchedU](#), as well as the technologies used to implement it.

2.1 Needs Analysis

To design the **SchedU** application, we defined expectations concisely. We identified two types of user in our system : **class prefects** and the **coordinator** (responsible for setting up time tables).

2.1.1 Administrator

Administrator must log on to the platform and have the ability to:

- view statistics
- consult schedules

2.1.2 The coordinator

Before accessing the platform, the **coordinator** must also log in. Once connected, he has the possibility to:

- Manage teachers
- Manage the sectors
- Manage materials
- View statistics

- Create schedules
- Share schedules

Once the coordinator shares the schedule, the people involved in that schedule receive an email.

2.1.3 Class supervisors

Class supervisors must first authenticate themselves on the platform. This will enable them to :

- provide days and time of classes attended

Indeed, this task allows the platform to automatically update the hourly masses of the Ecs in the platform.

2.2 Solution modeling

For the modeling of our system, we opted for the Unified Modeling Language ().

2.2.1 UML, what is it about?

The **UML** is a graphical modeling language used primarily in software development and systems design. It provides a modeling standard to represent software architecture. It includes, among other things, the following elements:

- use case diagram
- class diagram
- sequence diagram

2.2.2 Use case diagram

In the use case diagram presented in Figure 2.1, we show the different actions that our actors carry out.



Figure 2.1: Use Case Diagram

Administrator

- Login: The administrator logs into the web application to access its functionalities.
- View timetables: The administrator can view timetables for different years or departments to monitor organization.
- View timetables: The administrator can view timetables for different years or departments to monitor organization.
- View statistics: The administrator accesses statistics related to timetables or application use, which may include the number of scheduled classes, hours taught, etc.

Coordinator

- Login: Logs in to access advanced management functions.
- Manage teachers: The coordinator can add new teachers to the database, modify existing Teachers' information (name, subject, availability, etc.), or delete teachers.
- Teachers' information (name, subject, availability, etc.), or delete teachers.
- Manage streams: The coordinator can create new academic streams, modify existing ones (name of stream, study programs), or delete them if they are no longer relevant.
- Manage timetables: He/she is responsible for creating new timetables for each course and academic year. He can also modify existing timetables (changing times, rooms or teachers assigned), and delete them if necessary.
- Managing TUs/Ecs: The coordinator manages Teaching Units (TUs) and Elements (ECs), adding, modifying or deleting them according to changes in the study program.

Manager :

- Login: The manager logs in to access his/her specific functions.
- Enter the number of teaching hours completed: The manager registers the hours actually taught by the teachers, which makes it possible to monitor hourly workloads and check that lessons have been delivered according to schedule.

2.2.3 Class diagram

Figure 2.2 is a representation of the class diagram of our system. It shows the classes involved in designing the system and the relationships between them

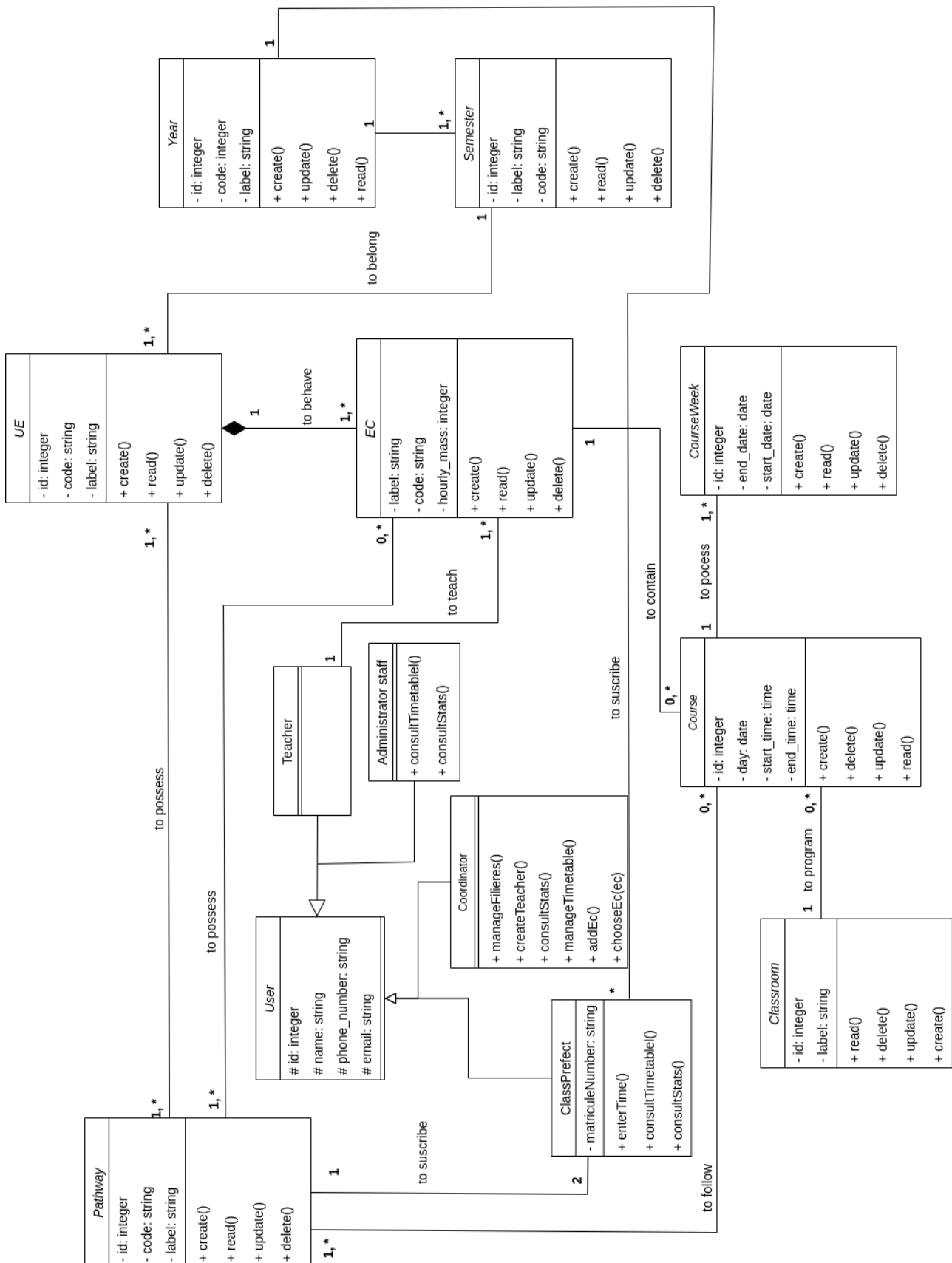


Figure 2.2: Class Diagram

This class diagram represents the main entities and relationships of a university timetable management application.

Streams and UEs (Unités d'Enseignement) are entities containing ECs (Éléments Constitutifs), each being associated with a UE. A year has several semesters, to which the UEs are attached. Users are divided into subtypes (teacher, administrator, coordinator, manager) with specific roles. For example, the coordinator manages courses and timetables, and can choose subjects, while the the class prefect keeps track of the hours worked. Courses are scheduled in classes over defined weeks. Each course is associated with a subject and a teacher.

The diagram also shows important relationships such as user enrolment in courses, course scheduling, and ownership of courses by courses and subjects. Interactions between actors enable timetables to be managed, subjects to be created and academic progress to be supervised.

2.2.4 Sequence diagram

This diagram shows the sequence of messages between objects during an interaction.

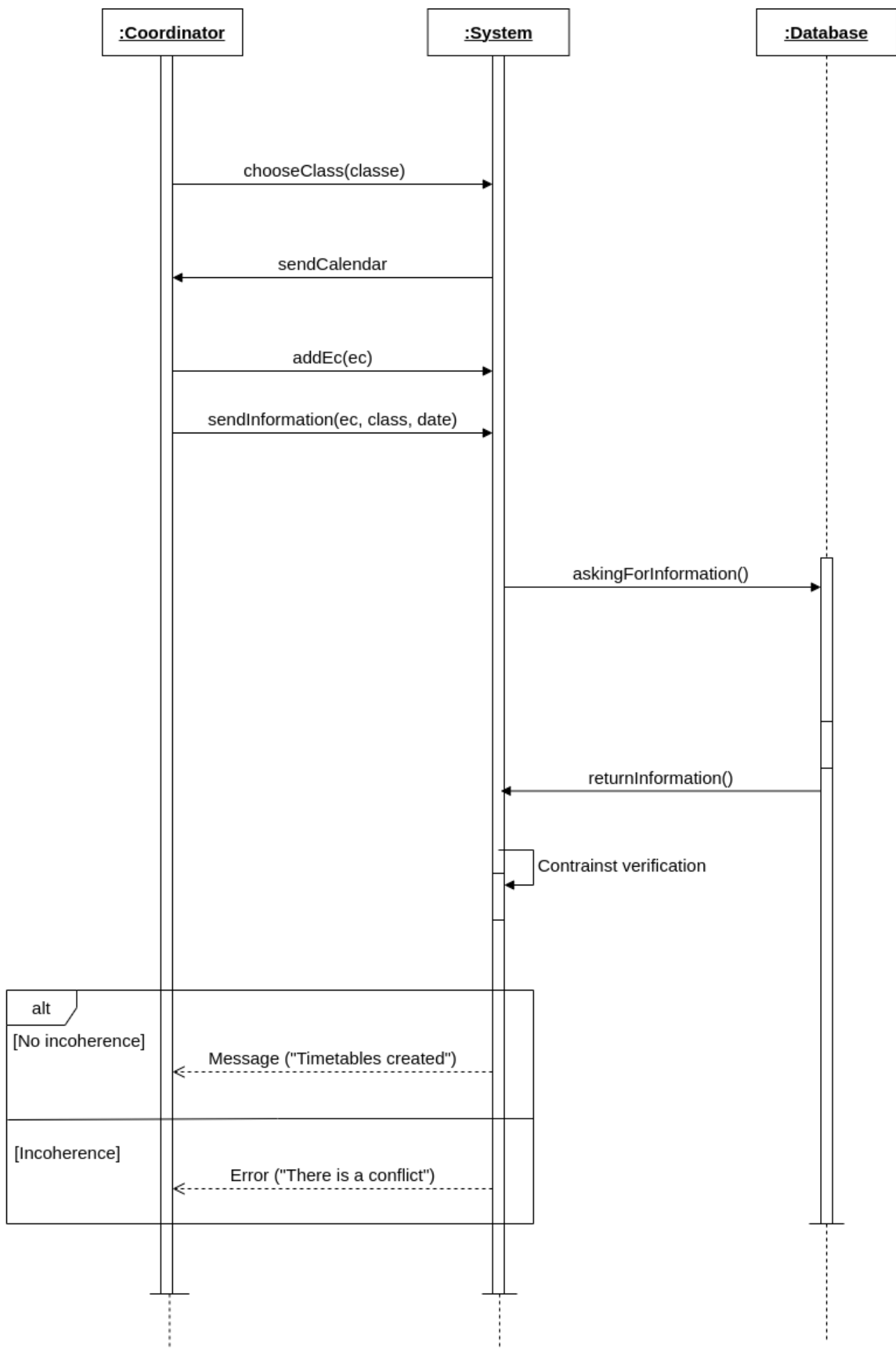


Figure 2.3: Sequence Diagram for creating a Timetable

This sequence diagram represented in figure 2.3, illustrates the interactions between three entities: the Coordinator, the System, and the Database in the creation of schedules. The Coordinator first selects a class and sends the schedule information to the System. Next, he adds a CE, and the system returns the CE information. The system requests additional information from the database to check for constraints (e.g. time conflicts). If the response is positive (no conflicts), a message confirms that the schedule has been created. If conflicts exist, an error is returned, indicating a schedule conflict.

2.3 Technical choices

To create our application, we used various technologies for every aspect of our system, from the user-friendly interface to data processing and database management. Here's how each technical component contributes to the creation of our solution.

Table 2.1: Technical Choices and Alternatives

Technology	Justification	Alternatives
HTML	HTML is universally supported by all browsers, ensuring maximum compatibility. It is the foundational language for web development.	Markdown (quick content generation but limited for complex pages), XML (less popular for front-end development).
CSS	CSS allows precise control over the appearance of elements on the web page, offering styling flexibility.	Sass (CSS preprocessor with variables and mixins), Less (similar to Sass but with different syntax).
Tailwind CSS	Tailwind CSS provides a utility-first approach, allowing custom designs quickly within the HTML structure.	Bootstrap (predefined components but less flexible), Bulma (lightweight and responsive, but less flexible than Tailwind).
Vue.js	Vue.js is a progressive JavaScript framework, ideal for dynamic user interfaces and Single Page Applications (SPAs).	React (popular library but requires more configuration), Angular (complete framework but heavier and more complex).
PHP	PHP is a reliable and widely used server-side scripting language for managing server logic, user requests, and databases.	Node.js (server-side JavaScript, non-blocking architecture), Python (Django, Flask: powerful web frameworks in Python).
Laravel	Laravel is a modern PHP framework that facilitates rapid, structured development with tools like Eloquent, Artisan, and database migrations.	Symfony (robust but more complex), CodeIgniter (lighter but with fewer built-in features).

2.3.1 Development methodology

The development process of our application follows the Scrum methodology, consisting of several distinct stages:

- **Product Backlog Creation**

We made a prioritized list of features, enhancements, bug fixes, and other requirements for

the project. The requirements were then broken in smaller features called user stories. They are manageable features that bring tangible value to the end-user.

- **Sprint Planning**

At the beginning of each sprint, we select items from the Product Backlog to work on. The goal is to create a Sprint Backlog, which defines the tasks to be completed during the sprint. This involves evaluating the project requirements, performance needs, and scalability with existing systems.

- **Sprint execution**

We work on the tasks defined in the Sprint Backlog, aiming to deliver a functional increment of the product. This includes writing code, creating user interfaces, and implementing backend services.

- **Weekly Sprint Review**

At the end of each week, we hold a Sprint Review to present the completed work. This gives us the opportunity to assess the progress, gather feedback, and make necessary adjustments to the Product Backlog for the following week.

- **Weekly Sprint Retrospective**

After the Sprint Review, we conduct a Sprint Retrospective to discuss what went well, what didn't, and how we can improve in the next sprint. This step is essential for continuous improvement.

2.4 System architecture

The system architecture is based on a three-layer approach. The backend is developed with Laravel, a PHP framework that manages the [API](#) and business logic. This [API](#) is responsible for processing requests, interacting with the MySQL database, and managing the application's functionalities.

The front-end is built with Vue.js, a progressive JavaScript framework for creating dynamic, responsive user interfaces. It communicates with the Laravel [API](#) via [HTTP](#) requests, delivering a fluid user experience.

MySQL, as a relational database, stores all the information required by the application, such as user data, course parameters and much more. The overall architecture guarantees a clear separation between presentation, business logic and data, ensuring flexibility and scalability. The architecture of ShedU is illustrated in [2.4](#).

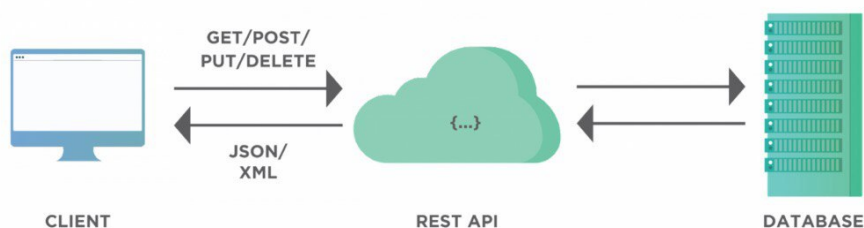


Figure 2.4: System architecture

Conclusion

Our application, [SchedU](#), automates the scheduling of academic activities at [IFRI](#) by efficiently organizing examination panels, managing date and time allocations, and ensuring that scheduling conflicts are avoided. [SchedU](#) handles key tasks such as creating optimized timetables for courses based on the availability of teachers, faculty, and classrooms; tracking important dates to maintain a comprehensive calendar of key academic events; and resolving scheduling conflicts by detecting and addressing any overlaps in exam schedules or panel assignments. This functionality can be extended to other educational entities, providing a flexible and adaptable approach to academic scheduling across various contexts.

Results and discussion

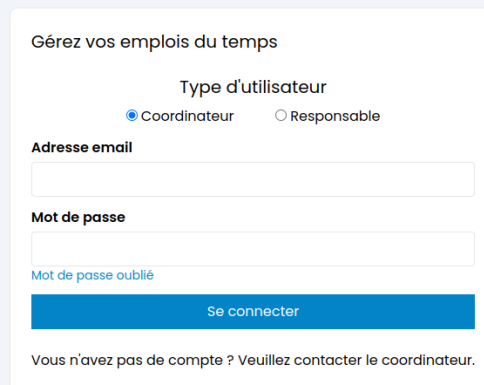
Introduction

This section outlines the methodical approach utilized, including the UML analysis, technical decisions, and a detailed presentation of the interfaces. Additionally, we showcase the key functionalities developed, while critically evaluating potential areas for improvement in the application.

3.1 Description of function

3.1.1 Authentication

Whether you are a administrator, a class manager or a coordinator, access to the various services requires prior identification. This is done by entering an e-mail address (matricule number for class manager) and a password,as illustrated in Figures [3.1](#) and [3.2](#)



Gérez vos emplois du temps

Type d'utilisateur

☒ Coordinateur ☐ Responsable

Adresse email

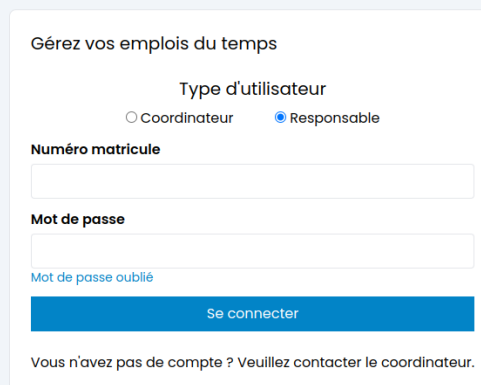
Mot de passe

[Mot de passe oublié](#)

Se connecter

Vous n'avez pas de compte ? Veuillez contacter le coordinateur.

Figure 3.1: Coordinator login



Gérez vos emplois du temps

Type d'utilisateur

☐ Coordinateur ☒ Responsable

Numéro matricule

Mot de passe

[Mot de passe oublié](#)

Se connecter

Vous n'avez pas de compte ? Veuillez contacter le coordinateur.

Figure 3.2: Class manager login

3.1.2 The Coordinator

The coordinator can create a schedule when logged into the application. To do so, he/she follows the steps below:(see figure 3.3).

The screenshot shows a web application interface for creating a new timetable. On the left is a 'Menus' sidebar with three items: 'Tableau de bord' (Dashboard), 'Configurations', and 'Créer un emploi du temps' (Create a timetable), which is currently selected. The top header shows the user 'A. Ratheil (en ligne)' and a close button. The main content area is titled 'Créer un nouvel emploi du temps' with the subtitle 'Programmer des cours pour une semaine de votre choix'. Below this is a section titled 'Choix de la classe' (Class choice) containing three radio button options: 'L1 (S1 & S2)' (selected), 'L2 (S3 & S4)', and 'L3 (S5 & S6)'. A blue 'Suivant →' (Next) button is positioned to the right of the options.

Figure 3.3: Year Choosing

After choosing year, he/she needs to choose the concerned branch(see figure 3.4).

The screenshot shows the second step of the timetable creation process. The sidebar and header are identical to Figure 3.3. The main content area is titled 'Créer un nouvel emploi du temps' with the subtitle 'Programmer des cours pour une semaine de votre choix'. Below this is a section titled 'Choix des filières' (Branch choice) containing five checkbox options: 'Génie Logiciel (GL)' (checked), 'Intelligence Artificiel (IA)' (checked), 'Internet et Multimédia (IM)' (checked), 'Système Embarqué et Internet Of Thing (SEIOT)' (unchecked), and 'Sécurité Informatique (SI)' (unchecked). At the bottom of the section are two blue buttons: '← Précédent' (Previous) and 'Suivant →' (Next).

Figure 3.4: Filiere Choosing

After that he/she get a calendar who contain the week witch contain the current day, as illustrated in figure 3.5.

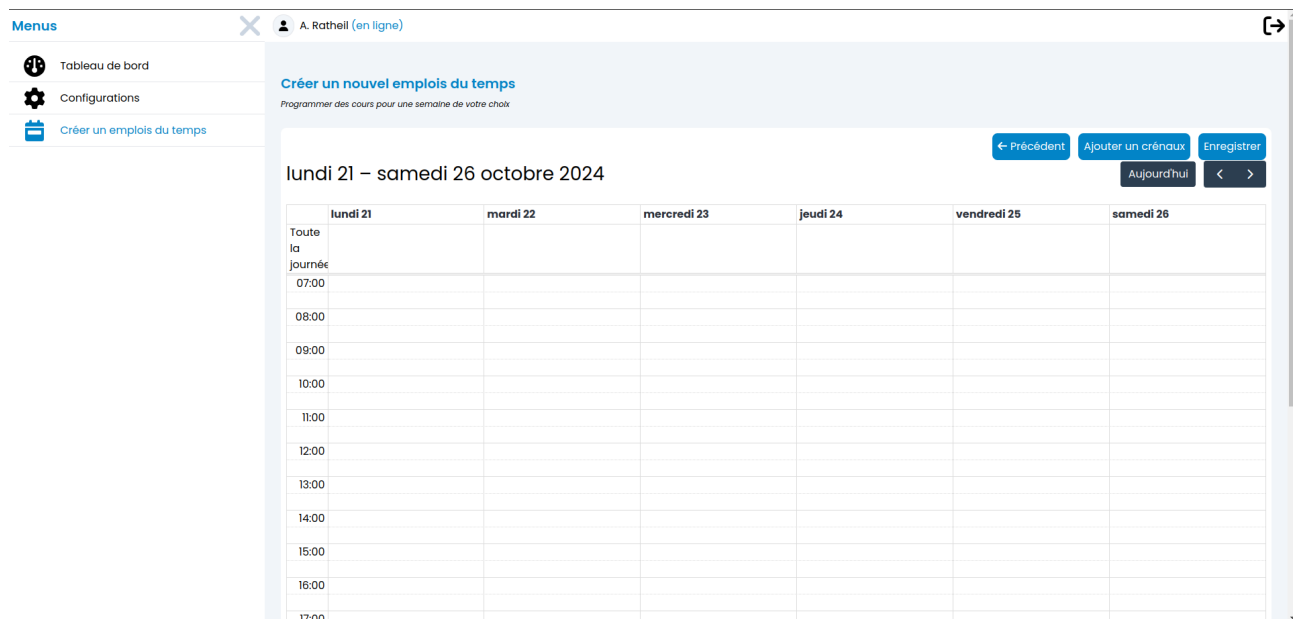


Figure 3.5: Weekly Course Calendar

Then figure 3.6 shows the process of creating the first schedule. He/she can also modify the creation week as desired. To create a course, simply click on a box on the calendar (the start time). The system then displays a modal window allowing the user to choose the information relating to the course.

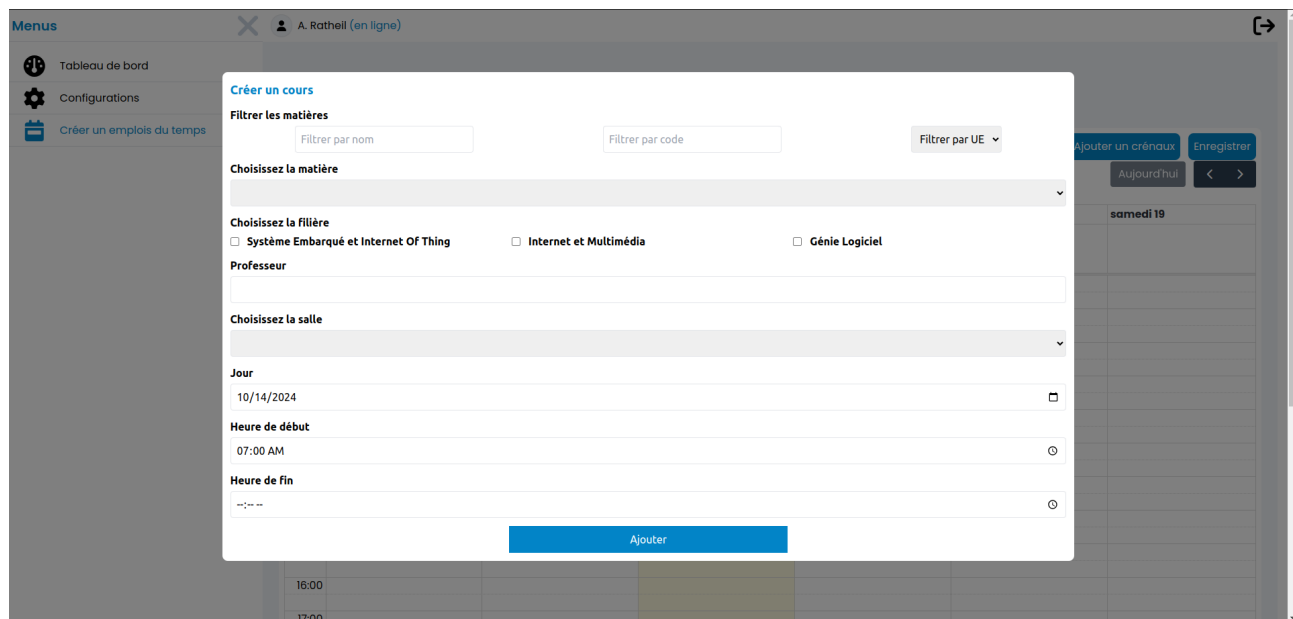


Figure 3.6: Course Modal Creation

He choose information he wanted and then, after validation, the course is added to a calendar view like that(see figure 3.7).

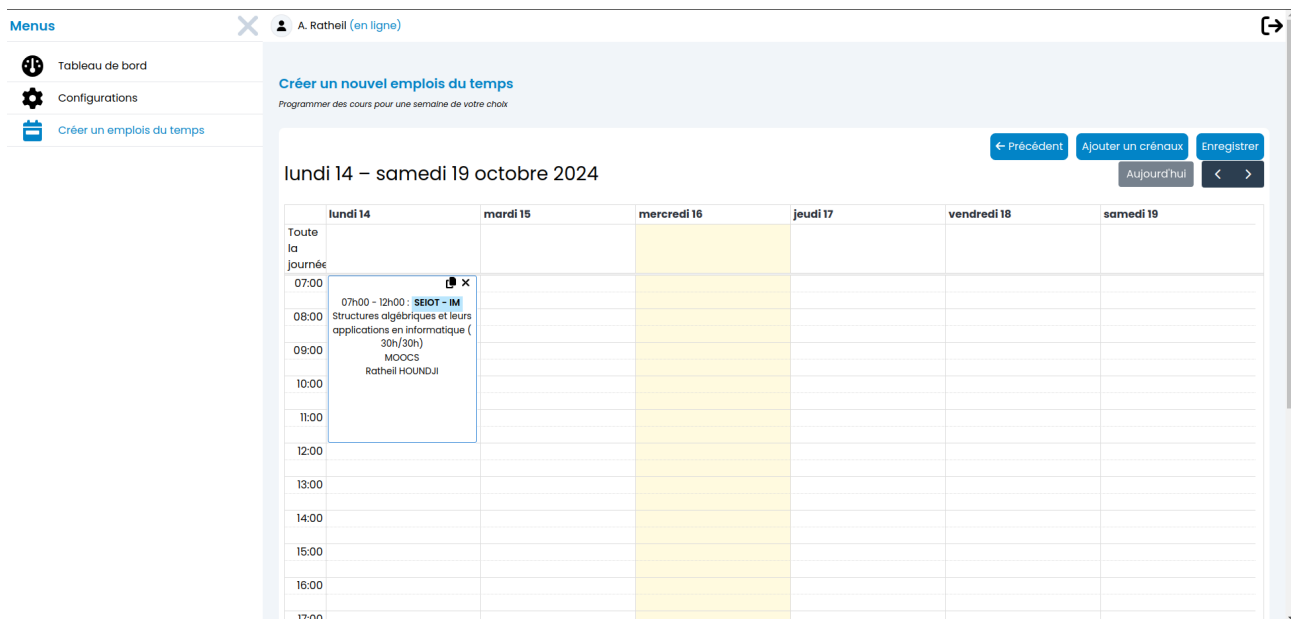


Figure 3.7: A course in a Calendar

As we see, he can copy and past this course in other case (for more simplicity). It can also update this course by clicking on it. And finally, he also have the possibilities to resize the courses directly on the view for modify the hours.

After all that, he obtains a calendar for a week ,as illustrated in figure 3.8.

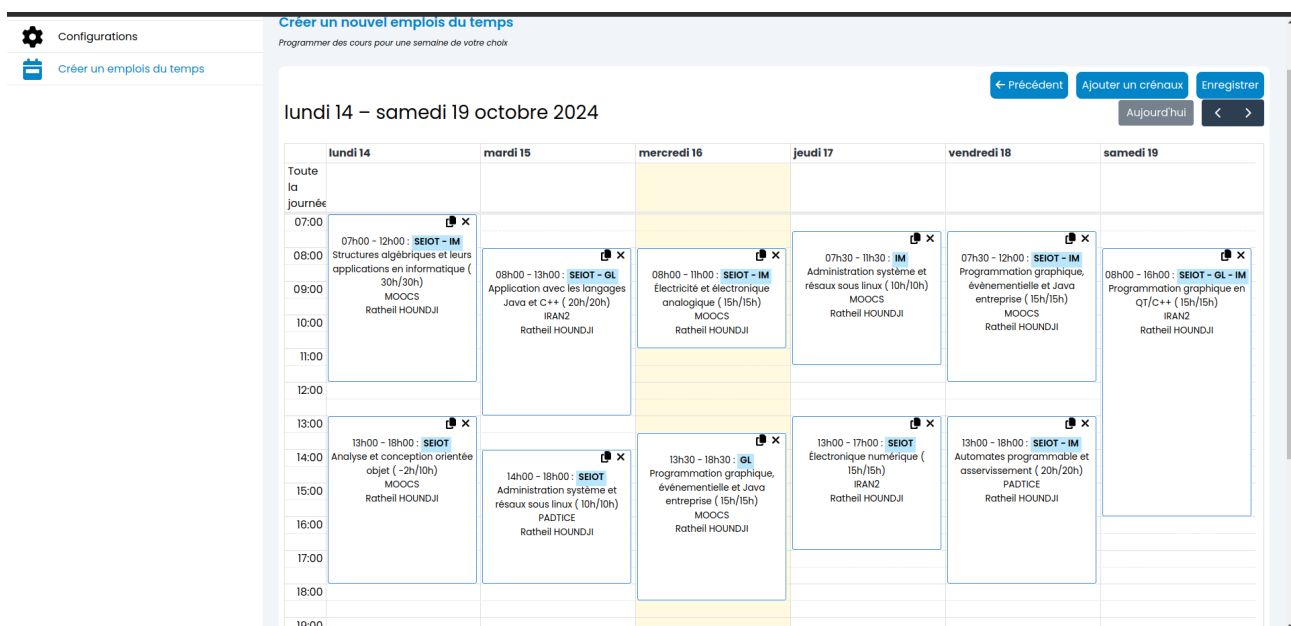


Figure 3.8: Full week Calendar

After that, he can save the calendar. And then, the system will make a request to our verification service of conflict. If the response is good, the calendar is saved, else the errors it resent to the coordinator. The server response is illustrated in figure 3.9 Then, the timetable is created. And the coordinator can create another or consult the created timetable.

Name	×	Headers	Payload	Preview	Response	Initiator	Timing
 timetables	▼	General					
	Request URL:		http://127.0.0.1:8000/api/timetables				
	Request Method:		POST				
	Status Code:		 201 Created				
	Remote Address:		127.0.0.1:8000				
	Referrer Policy:		strict-origin-when-cross-origin				
	▶ Response Headers (7)						
	▶ Request Headers (16)						

Figure 3.9: Server response

3.1.3 Class supervisor

When class supervisor is connected to the app, he has a dashboard who allow his to add information about done courses in the week (see figure 3.10).

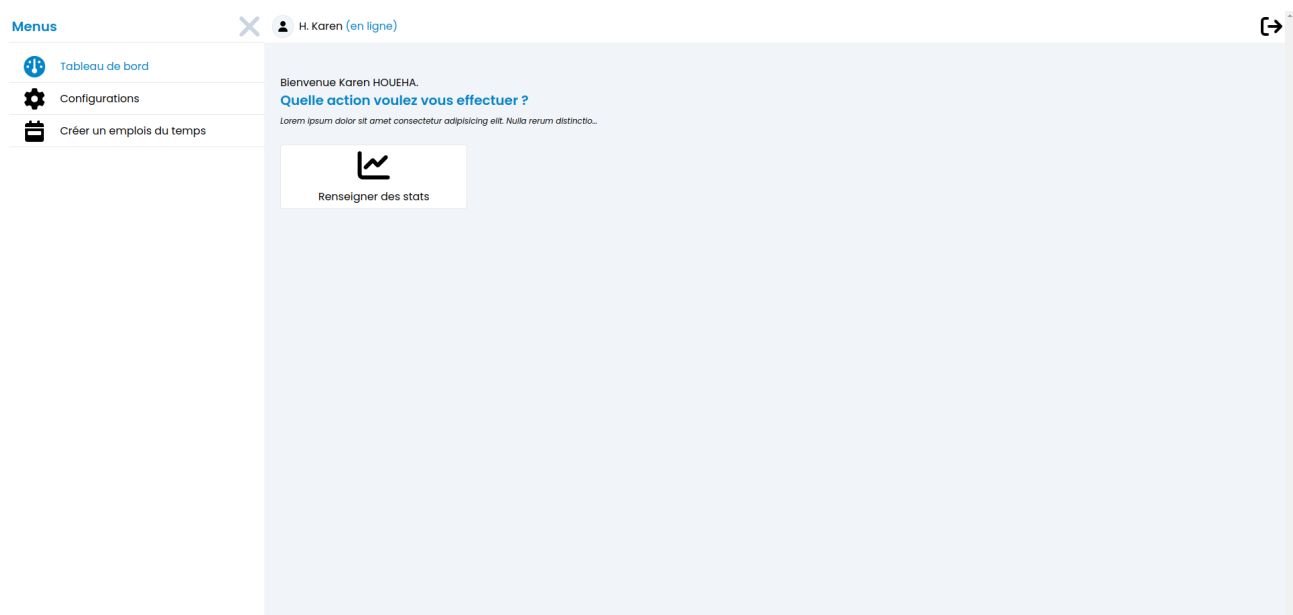


Figure 3.10: Class manager dashboard

They must confirm their academic year and course of study before they can view the ecs that apply to them. The system will automatically detect those informations but it just a confirmation.

After confirmation, he get a list of his EC and have the possibilities to choose on of them and enter number of courses (see figures 3.11 and 3.12).

The screenshot shows a web application interface. On the left is a sidebar menu with the title "Menus" and three items: "Tableau de bord", "Configurations", and "Créer un emploi du temps". The main area has a header with a user profile "H. Karen (en ligne)" and a close button. Below the header, it says "Bienvenue Karen HOUÉHA." and "Quelle action voulez vous effectuer ?". There is a placeholder text "Lorem ipsum dolor sit amet consectetur adipiscing elit. Nulla rerum distinctio..." and a button labeled "Renseigner des stats". A modal dialog titled "Renseignez les stats" is open, asking to "Confirmez l'année" with radio buttons for "L1", "L2" (selected), and "L3". A "Continuer" button is at the bottom of the modal.

Figure 3.11: Class manager academic year confirmation

The screenshot shows the same web application interface as Figure 3.11. The modal dialog titled "Renseignez les stats" is open, asking to "Confirmez la filière". It has radio buttons for "Génie Logiciel" (selected), "Intelligence Artificiel", "Internet et Multimédia", "Système Embarqué et Internet Of Thing", and "Sécurité Informatique". A "Continuer" button is at the bottom of the modal.

Figure 3.12: Choose class branch

Les matières (ECs) de la GL
Choisissez la matière dont vous voulez mettre à jour la masse horaire.

Afficher 10 ECs Rechercher

Code	Label	Masse Horaire	Masse Horaire restante	Action
IINF1322	Analyse et conception orientée objet	10 h	~2 h	Modifier
IINF1323	Structure de données et applications avec C/Python	20 h	20 h	Modifier
IINF1324	Programmation graphique, événementielle et Java entreprise	15 h	15 h	Modifier
IINF1325	Programmation graphique en QT/C++	15 h	15 h	Modifier
IINF1326	Aspects avancés des technologies web	10 h	10 h	Modifier
IINF1327	Assurance qualité et test de logiciel	10 h	10 h	Modifier
IINF1328	Maintenance des appareils électroniques	5 h	5 h	Modifier
IMTH1321	Structures algébriques et leurs applications en informatique	30 h	30 h	Modifier
2INF1322	Application avec les langages Java et C++	20 h	20 h	Modifier
2INF1324	Projet de validation des acquis en Java	10 h	10 h	Modifier

Affichage de 1 à 10 sur 11 ECs

Figure 3.13: Class manager EC list

Mettre à jour la masse horaire de la matière [Modifier](#)

Informations générales

Code : IINF1324
 Label : Programmation graphique, événementielle et Java entreprise
 Masse horaire totale : 15h
 Masse horaire restante : 15h

Modification

Date du cours : 10/16/2024 [🗑️](#)
 Nombre d'heure : 4

[Sauvegarder](#)

Figure 3.14: Enter number of hour for a course

The figures ?? and 3.14 illustrate respectively the list of all the courses and how the class manager chooses the numbers of hours allocated to the courses completed. After these steps, the hourly mass is automatically updated in our database.

3.1.4 The Administrator

The administrator can just view statistics and consult created timetable. The coordinator and class manager also can (see figure 3.15). Then in the figure 3.16, the administrator has the possibility to view, the list of weekly schedules.

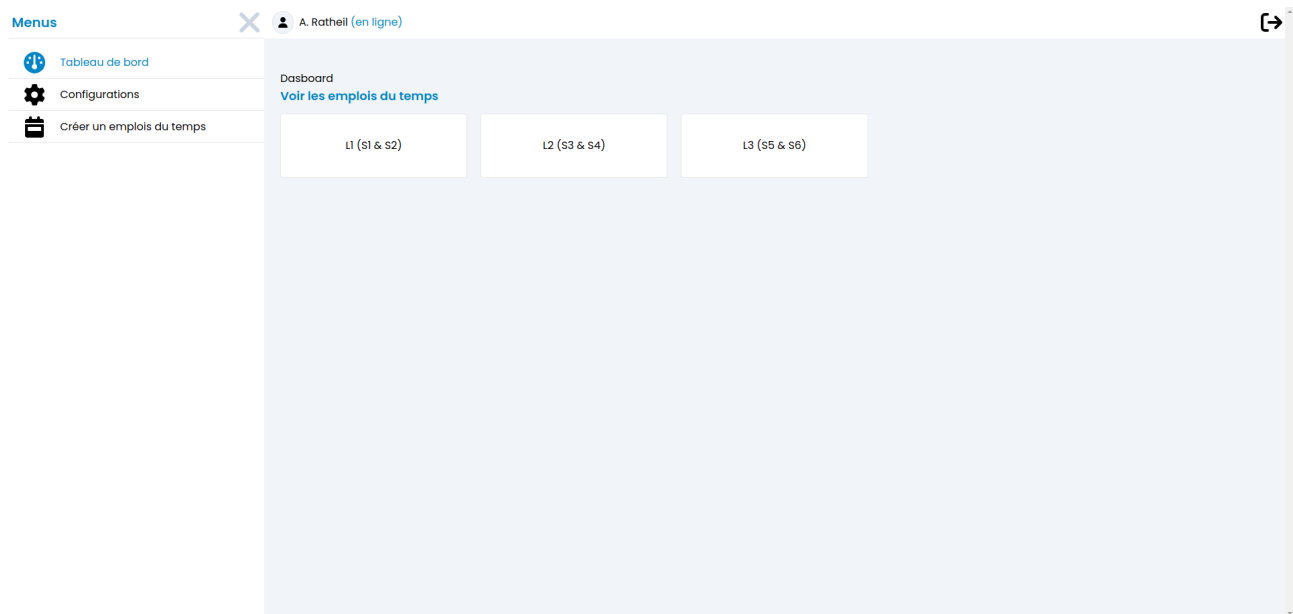


Figure 3.15: Consult timetable

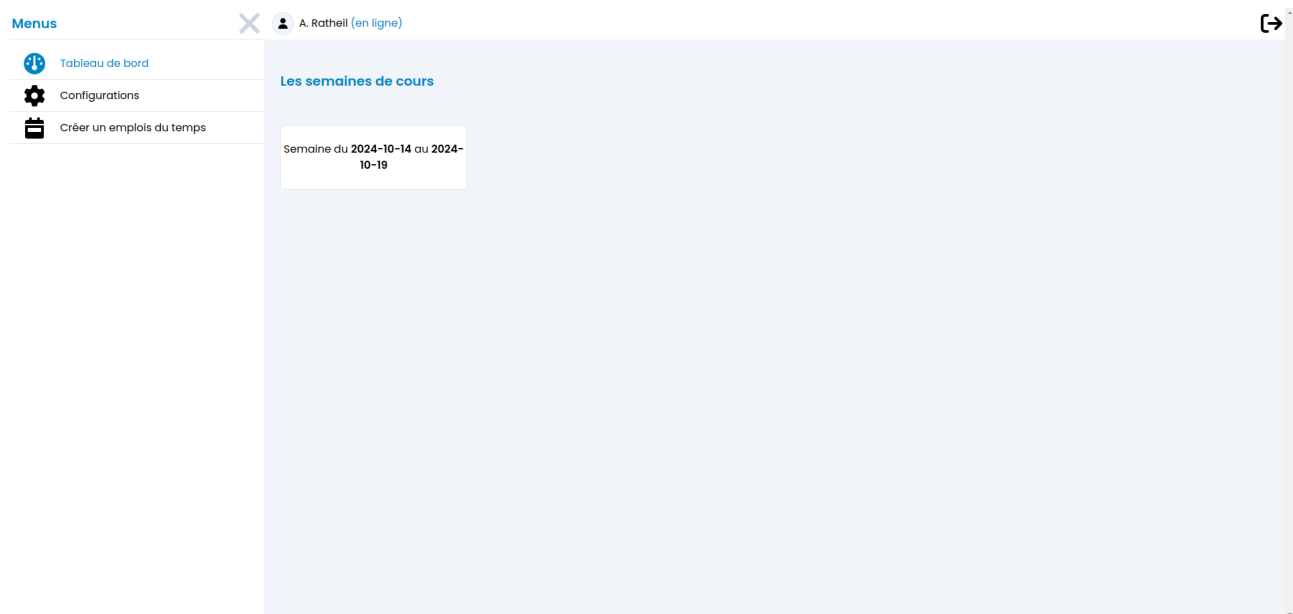


Figure 3.16: List of scheduled course weeks

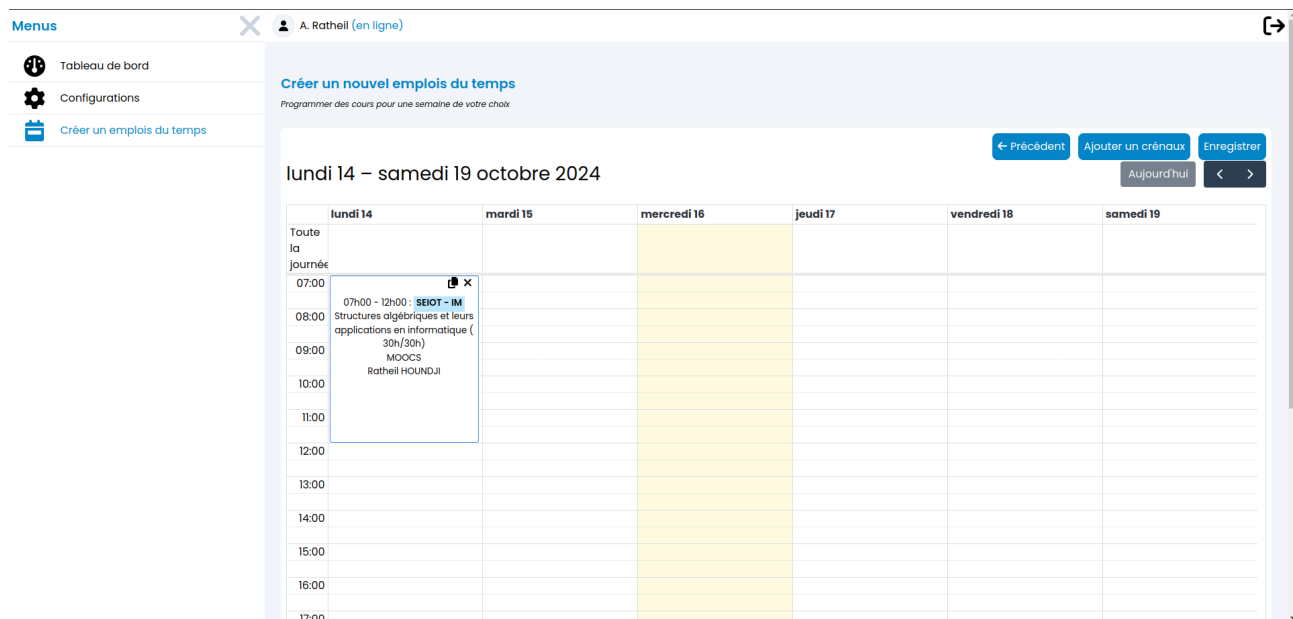


Figure 3.17: Course displaying in a camendar

Figure 3.18 Show a large view of what a weekly courses schedule,does look like.It contains days of the week,courses names,teachers and time allocated to each courses.

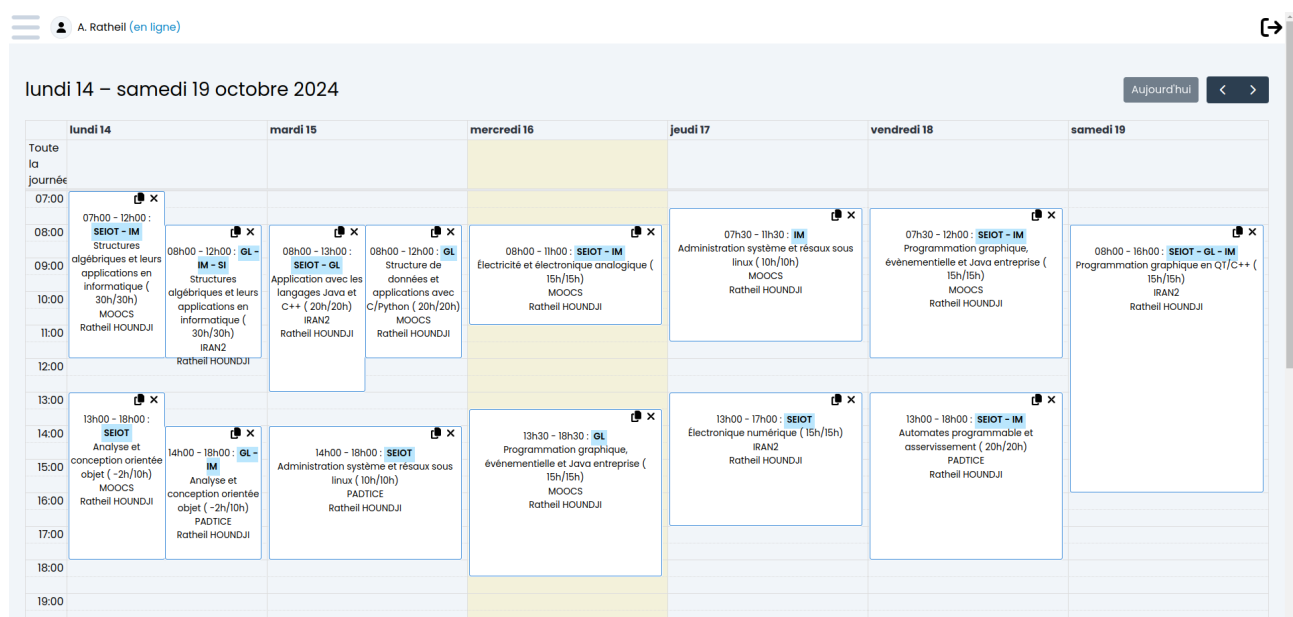


Figure 3.18: Global view of a timetable

3.1.5 Others

After all of that, the coordinator can manage or consult something like resources in the application. He has the possibilities to make some configuration(see figure 3.19).

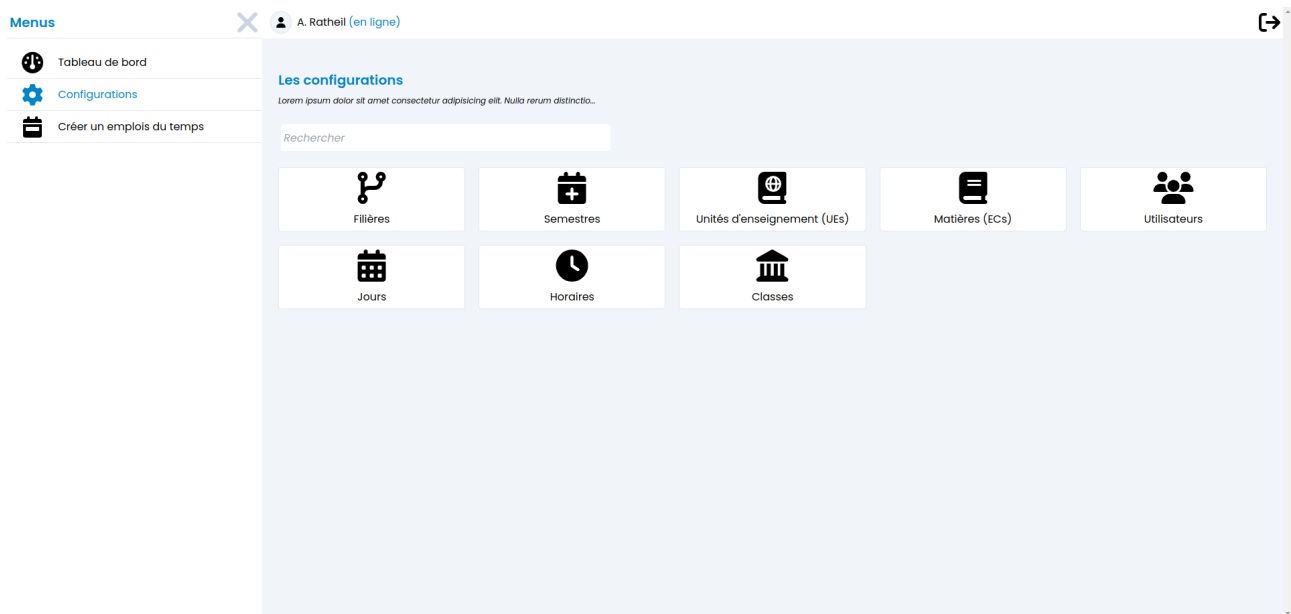


Figure 3.19: Configuration view

3.2 Alert System (Via E-mail)

Once the timetable has been created by the coordinator, it can be published. Upon publication, the teachers and class leaders concerned receive an e-mail notification containing the shared timetable.

3.3 Tests

In software development, unit testing is essential to guarantee code quality and reliability. They consist in checking the correct operation of individual code units, such as functions, methods or classes, by isolating each unit and testing them independently. A number of unit tests have been written for this purpose. The figure 3.20 shows the results obtained after testing the store and get timetable functionalities.

```
jordy@jordy-HP-ZBook-15u-G5:/media/jordy/jordy/projet de licence/nouveau/projet/backend-gestion-emplois-du-temps$ php artisan test

PASS Tests\Unit\ExampleTest
✓ that true is true

PASS Tests\Feature\CourseWeekServiceTest
✓ example
✓ store timetable
✓ get timetable

PASS Tests\Feature\ExampleTest
✓ the application returns a successful response

PASS Tests\Feature\UserServiceTest
✓ example
✓ user password hasher

Tests: 7 passed (7 assertions)
Duration: 0.23s

jordy@jordy-HP-ZBook-15u-G5:/media/jordy/jordy/projet de licence/nouveau/projet/backend-gestion-emplois-du-temps$
```

Figure 3.20: Unit Tests

3.4 Application deployment aspects

For the deployment of my system designed with a Laravel API, a Vue.js front-end and a MySQL database, several aspects need to be taken into account. Firstly, I need to host the Laravel API on a server capable of handling PHP applications, such as an Apache or Nginx server, with support for PHP 8+ and MySQL. The Vue.js front-end can be served either by the same server or by a different one, depending on the architecture you choose (monolithic or separate). If I choose a separate deployment, the frontend could be hosted on a service like Netlify or Vercel, while the Laravel API would be hosted on a separate server. Sanctum integration for authentication requires special attention to ensure that front-end and back-end are on the same domain or sub-domains to facilitate management of session cookies and CSRF tokens. Finally, database management services, such as Amazon RDS or a MySQL instance on the same server, can be used to host the database. A version management system such as Git and a CI/CD pipeline (e.g. GitHub Actions or GitLab CI) can also be set up to automate system updates and manage ongoing deployments.

3.5 Security aspects

We used Laravel for the backend, leveraging its robust built-in security features that help safeguard our application against common vulnerabilities.

SQL Injection Protection: To prevent SQL injection attacks, we implemented measures such as prepared statements and parameterized queries. This ensures that user input is properly sanitized and treated as data rather than executable code, thus protecting our database from unauthorized access.

Cross-Site Request Forgery (CSRF): We utilized Laravel's CSRF protection to prevent unauthorized commands from being executed on behalf of authenticated users. This involves generating CSRF tokens that must be included in forms and AJAX requests, ensuring that only legitimate requests are processed.

Cross-Site Scripting (XSS): To mitigate the risk of XSS attacks, we employed data sanitization techniques. This includes escaping output data to prevent malicious scripts from being executed in users' browsers, protecting user interactions and sensitive information.

Authentication and Authorization: We established a robust authentication system using Laravel's built-in features, allowing users to securely log in and access their accounts. Additionally, we implemented role-based access control to restrict functionalities based on user roles, ensuring that only authorized users can access sensitive features and data.

Data Encryption: We utilized encryption methods for storing sensitive data, such as user passwords and personal information, ensuring that this data remains secure even if accessed by unauthorized individuals. We also enforced secure communication protocols (HTTPS) to protect user information during transmission between the client and server.

Middleware: We have set up middleware to secure and verify our requests before they reach our server.

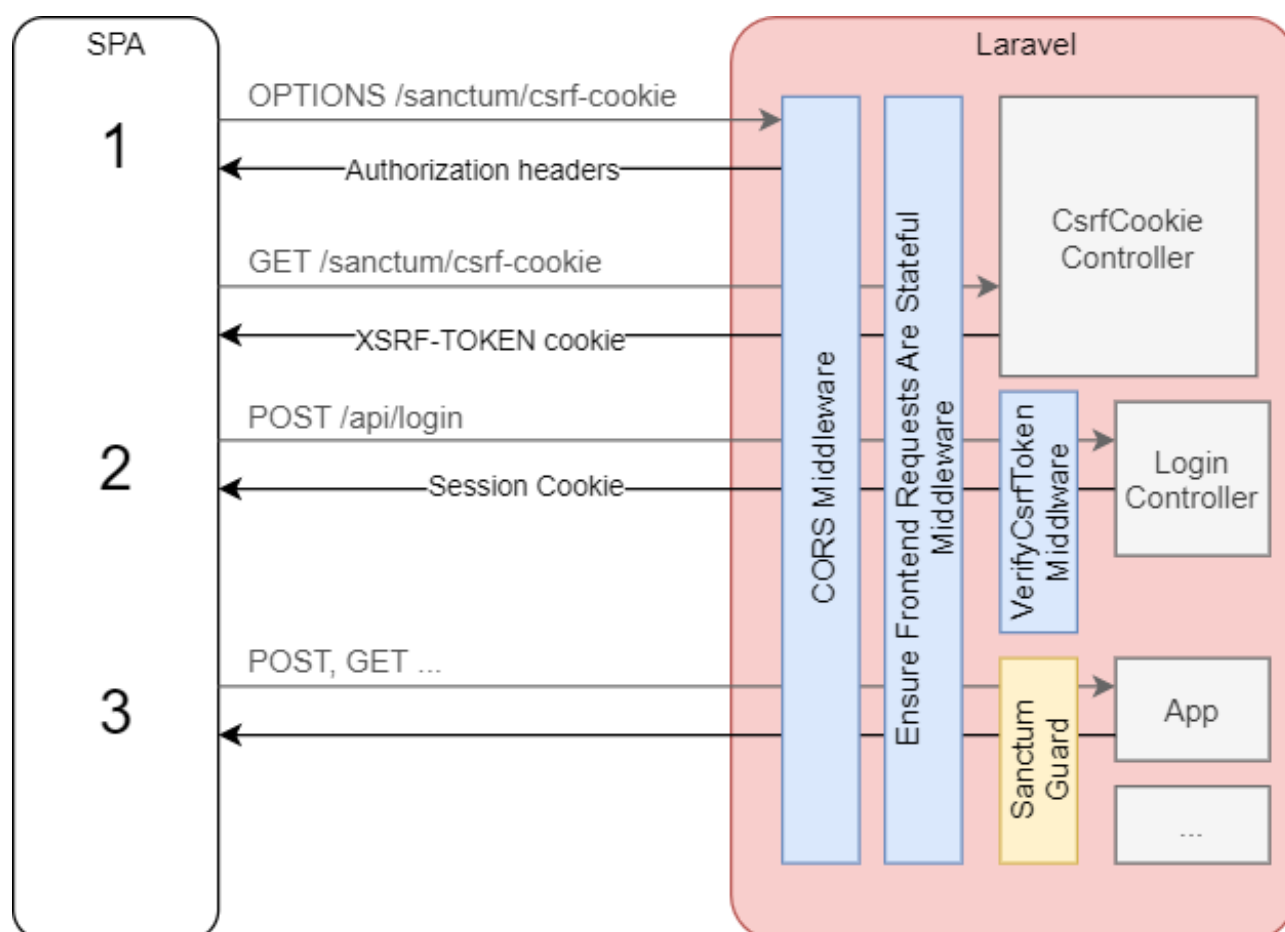


Figure 3.21: laravel middleware set up

This diagram illustrates the authentication and request management process in an application using Laravel and a single-page application (SPA), with the integration of Sanctum for CSRF (Cross-Site Request Forgery) cookie management. Here are the main steps:

- Initial query
The SPA sends an 'OPTIONS' request to the '/sanctum/csrf-cookie' endpoint to obtain a CSRF cookie. This request passes through the CORS middleware.
The Laravel server generates an XSRF-TOKEN cookie which is sent back to the SPA.
- Authentication
The SPA sends a 'POST' request to '/api/login' with the connection information.
If authentication is successful, Laravel returns a secure session cookie.
- Authenticated requests
Subsequent SPA requests (GET, POST, etc.) automatically include authentication cookies (CSRF and session) to enable secure interactions with the application.
Laravel middleware, such as 'CORS Middleware', 'VerifyCsrfToken Middleware', and 'Sanctum Guard', verify and secure requests and responses.

3.6 Discussions

ShedU provides a solution for organizing academic timetables tailored to the needs of the administration. The platform assists in creating and managing schedules for classes, exams, and other academic activities, ensuring that all time slots are effectively utilized. The application simplifies the scheduling process by enabling the coordinator to choose dates, times, and locations according to the availability of teachers and classrooms. By integrating real-time data, ShedU effectively prevents double bookings and scheduling conflicts, thereby enhancing the efficiency and reliability of the overall timetable management. The impact of ShedU is evident through its practical implementation. The platform has been successfully utilized to manage the timetables for numerous courses and academic events at [IFRI](#). This extensive usage highlights ShedU's capability to handle large volumes of scheduling tasks while ensuring accuracy and compliance with academic standards. Additionally, the platform includes features for generating essential administrative documents, such as class schedules, and notification emails for teachers. These documents facilitate clear communication and further enhance administrative workflows.

Conclusion

Through this project, we developed **ShedU**, a web application aimed at streamlining the scheduling of **timetables** for students at [IFRI](#). This journey has been fulfilling both professionally and personally, allowing us to grow and learn as a team. We acknowledge that there are numerous additional features and enhancements we have yet to explore, which could significantly enhance the functionality and user experience of our solution. Looking ahead, we are excited about the possibilities for future development, including the integration of advanced analytics for better resource allocation to continuously refine the platform. With ongoing collaboration and innovation, we aim to expand ShedU's capabilities and ensure its relevance in various educational contexts.

General Conclusion and Perspectives

The implementation of ShedU at the Institute of Training and Research in Computer Science, IFRI, marks a significant advancement in the way academic timetables are managed. Traditionally, creating timetables for courses and exams has been a manual and labor-intensive process, often resulting in numerous challenges. Administrators faced difficulties in balancing the availability of teachers and classroom resources, leading to scheduling conflicts, double bookings, and inefficiencies that could disrupt the academic calendar. This cumbersome approach not only consumed valuable time and effort but also hindered the overall effectiveness of the educational process. ShedU addresses these challenges head-on by offering a comprehensive solution tailored specifically for IFRI. Our application simplifies the scheduling process by enabling administrators to easily select dates, times, and locations based on real-time availability of faculty and facilities. The integration of features such as conflict detection and automatic adjustments ensures that potential issues are mitigated before they arise. Furthermore, ShedU includes functionalities such as generating class schedules, managing exam timetables, and creating essential administrative documents, including notification letters and timetables for students. This level of automation enhances the efficiency and reliability of the scheduling process, allowing staff to focus on more critical tasks. Additionally, ShedU provides a user-friendly interface that enables both administrators and students to access and understand their schedules effortlessly. The ability to update timetables in real-time ensures that all stakeholders remain informed about any changes, fostering clear communication within the academic community. By addressing both logistical and administrative challenges, ShedU not only streamlines the management of timetables but also contributes to a smoother academic experience for students and faculty alike. Looking forward, the potential for further development of ShedU is vast. We recognize that additional functionalities could enhance the platform's capabilities, such as the integration of analytics for data-driven decision-making and tools for student feedback to continuously improve the user experience. Expanding ShedU's features to include mobile access and personalized notifications could further elevate its utility, ensuring that it remains relevant and effective in meeting the evolving needs of IFRI. ShedU has the potential to revolutionize the way academic timetables are managed at IFRI and beyond. By transitioning from a manual scheduling process to a streamlined digital solution, we can significantly enhance the efficiency of academic operations. The success of ShedU at IFRI underscores its applicability to other educational institutions, paving the way for broader adoption and demonstrating our commitment to improving the academic landscape through innovative technology. As we continue to refine and expand ShedU's functionalities, we remain dedicated to fostering an environment conducive to learning and academic excellence.

Bibliography

- [1] High performance MySQL, Baron Schwartz, Vadim Tkachenko, Peter Zaitsev, Derek J. Balling, juin 2008
- [2] UML2, par la pratique , Pascal Roques, Eyrolles, 2011
- [3] Mastering Laravel, Christopher John Pecoraro, 2015

Webography

- [4] How to choose the best methodology for you, <https://www.atlassian.com/agile/scrum/agile-vs-scrum#:~:text=Scrum%20is%20an%20agile%20framework,helps%20them%20achieve%20this%20cadence>, Last accessed: August 10, 2024
- [5] Hypertext Markup Language (HTML), <https://developer.mozilla.org/fr/docs/Web/HTML>, Last accessed: August 15, 2024
- [6] Cascading Styling Sheet, https://developer.mozilla.org/fr/docs/Learn/Getting_started_with_the_web/CSS_basics, Last Accessed: August 16, 2024
- [7] JavaScript, <https://developer.mozilla.org/fr/docs/Web/JavaScript>, Last Accessed: August 17, 2024
- [8] Laravel, <https://laravel.sillo.org/>, Last Accessed: August 16, 2024
- [9] <https://www.php.net/>, Last Accessed: August 17, 2024
- [10] Vue.js, <https://vuejs.org/>, Last accessed: August 18, 2024
- [11] , Tailwind-CSS, <https://tailwindcss.com/>, Last accessed: August 19, 2024
- [12] , <https://www.pass-education.fr/qu-est-ce-qu-un-emploi-du-temps/>, Last accessed: August 20, 2024
- [13] , Calendar and scheduling components, <https://www.componentsource.com/function/calendar-scheduling-component>, Last accessed: September 15, 2024
- [14] , Smart.Scheduler-TimeTable, <https://www.htmlelements.com/docs/scheduler-time-table>, Last accessed: September 17, 2024
- [15] , Scheduling-components, <https://lp.datablend.com/knowledge/scheduling-components>, Last accessed: September 20, 2024
- [16] , Schedule apps, <https://dribbble.com/tags/schedule-app>, Last accessed: September 24, 2024
- [17] , applications de gestion d'emploi du temps, <https://clickup.com/fr-FR/blog/118124/applications-de-gestion-du-temps-meilleures>, Last accessed: September 27, 2024

-
- [18] , educap, <https://educap.io/>, Last accessed:October 10,2024
- [19] , notion, <https://www.notion.so/fr>, Last accessed:October 10, 2024
- [20] , Asana, <https://asana.com/fr>, Last Accessed:October 11, 2024
- [21] , trello, <https://trello.com/fr>, Last accessed:October 11, 2024
- [22] , Understanding Agile scrum in 10 minutes, <https://www.tuleap.org/agile/agile-scrum-in-10-minutes>, Last accessed: October 15, 2024
- [23] , Agile scrum method explained, <https://www.inflectra.com/Solutions/Methodologies/Scrum.aspx>, Last Accessed: October 15 , 2024

French Summary

Mise en Contexte

Le présent mémoire porte sur la conception et le développement d'une application web nommée SchedU, destinée à optimiser la gestion des emplois du temps au sein de l'Institut de Formation et de Recherche en Informatique (IFRI). Depuis plusieurs années, l'établissement rencontre des difficultés dans l'organisation manuelle des emplois du temps, une tâche essentielle mais complexe, qui est pourtant effectuée sans l'aide d'outils numériques spécifiques. Chaque semaine, le coordinateur académique doit collecter et intégrer de nombreuses informations : la disponibilité des salles, les horaires des professeurs, les cours à dispenser et les contraintes liées aux filières et aux années académiques. Ce processus est non seulement chronophage, mais également sujet à de nombreuses erreurs, comme les chevauchements d'horaires ou les attributions incorrectes de salles. Ces erreurs entraînent des réajustements fréquents, créant des retards et des perturbations dans le déroulement des cours.

Face à ces défis croissants, il est devenu impératif de développer une solution numérique capable de centraliser et de simplifier la gestion des emplois du temps. SchedU a ainsi été conçue dans le but de répondre à ces besoins spécifiques de l'IFRI. En automatisant la collecte des informations et en intégrant des mécanismes de vérification des conflits, l'application vise à réduire considérablement les erreurs humaines et à alléger la charge de travail du coordinateur. Elle permettra également d'assurer une meilleure coordination entre les différents acteurs impliqués dans l'élaboration des emplois du temps, notamment les professeurs, les responsables de filière et les étudiants.

Introduction de SchedU

La création de SchedU s'inscrit dans une dynamique plus large de transformation numérique des établissements d'enseignement supérieur, qui cherchent à tirer parti des technologies modernes pour améliorer leur efficacité organisationnelle. En digitalisant la gestion des emplois du temps, l'IFRI espère améliorer la précision, la rapidité et la transparence de ce processus. L'application vise également à offrir une expérience utilisateur intuitive pour les différents types d'utilisateurs (coordinateurs, professeurs, responsables de classe), tout en étant capable d'évoluer pour intégrer de nouvelles fonctionnalités au fil du temps.

Ce mémoire présente donc le développement de SchedU comme une solution répondant aux défis actuels de l'IFRI, en permettant une meilleure gestion des ressources académiques, une réduction des erreurs de planification, et une communication plus fluide entre les parties prenantes. Le projet ambitionne ainsi de contribuer à la modernisation de la gestion des emplois du temps dans le cadre académique, et de s'intégrer dans une stratégie plus large de numérisation des processus éducatifs.

Généralité sur SchedU

SchedU est une application web spécialement conçue pour répondre aux besoins de gestion des emplois du temps dans les établissements académiques, et plus précisément à l'Institut de Formation et de Recherche en Informatique (IFRI). Cette plateforme a pour objectif de centraliser et d'automatiser l'ensemble du processus de planification des cours, prenant en compte les contraintes liées à la disponibilité des professeurs, des salles et des créneaux horaires.

Avec SchedU, le coordinateur académique peut gérer facilement les informations relatives aux filières, aux matières et aux années académiques, tout en s'assurant d'éviter les conflits d'horaires, comme les chevauchements de salles ou de cours. L'application offre également une interface intuitive permettant aux professeurs et aux responsables de classe d'accéder rapidement à leurs emplois du temps et d'être notifiés en cas de modifications.

L'un des atouts majeurs de SchedU réside dans sa capacité à vérifier automatiquement les conflits potentiels lors de la création des emplois du temps, puis d'alerter les utilisateurs concernés. De plus, elle permet de suivre les heures de cours déjà dispensées par les professeurs, afin d'assurer une meilleure répartition des masses horaires.

En intégrant les dernières technologies comme Laravel pour le backend et Vue.js pour le frontend, SchedU se veut une solution évolutive et adaptée aux besoins de l'IFRI. Elle vise non seulement à simplifier la gestion des emplois du temps, mais aussi à améliorer la coordination entre tous les acteurs de la vie académique, tout en réduisant les erreurs humaines et en rendant le processus de planification plus efficace et transparent.

Technologies Utilisées et Fonctionnalités

Le framework Laravel 11 a été choisi pour le backend en raison de sa fiabilité et de sa capacité à gérer des applications complexes. Laravel offre des fonctionnalités prêtes à l'emploi, telles que l'authentification, la gestion des bases de données et la validation des données, ce qui accélère le développement tout en garantissant une sécurité et une performance élevées. Sa structure modulaire facilite également la maintenance et l'ajout de nouvelles fonctionnalités à l'avenir.

Pour le frontend, Vue.js a été privilégié pour sa simplicité d'intégration et sa réactivité. Vue.js permet de créer des interfaces utilisateur dynamiques et interactives, améliorant ainsi l'expérience utilisateur. Sa nature progressive permet d'ajouter facilement des composants réactifs à une application existante, offrant une flexibilité idéale pour ce projet.

Sanctum a été intégré pour la gestion de l'authentification, car il permet une authentification sécurisée via des tokens API tout en restant simple à configurer et à utiliser. Il est adapté aux

applications SPA (Single Page Application) comme SchedU, qui nécessitent une gestion sécurisée des sessions utilisateur.

Enfin, FullCalendar a été sélectionné pour la visualisation des emplois du temps en raison de sa capacité à gérer des calendriers interactifs et flexibles. Il permet de visualiser facilement les cours, de gérer les créneaux horaires, et d'interagir avec les événements du calendrier de manière intuitive. La base de données MySQL a été choisie pour sa stabilité et sa performance dans la gestion de grandes quantités de données, ce qui est essentiel pour un projet de gestion académique. Ces technologies, combinées, offrent une solution complète et évolutive pour répondre aux besoins spécifiques de l'IFRI.

État de l'art de l'Existant et Avancées de KidneyDoc

L'état de l'art concernant la gestion des emplois du temps académiques révèle plusieurs solutions existantes, mais celles-ci présentent souvent des limites dans leur capacité à s'adapter aux spécificités des établissements d'enseignement supérieur. La plupart des institutions utilisent encore des méthodes manuelles ou semi-manualisées pour organiser leurs emplois du temps, généralement avec des outils comme Excel ou des calendriers partagés, ce qui les expose à des erreurs humaines fréquentes, telles que des conflits d'horaires, des chevauchements de salles, ou des oublis de disponibilités des professeurs.

Certaines solutions logicielles dédiées existent sur le marché, comme Google Calendar ou Timetabler, mais elles sont souvent coûteuses, rigides et difficiles à personnaliser pour répondre aux besoins spécifiques d'établissements comme l'IFRI. De plus, ces outils sont généralement destinés à des structures de grande envergure, ce qui les rend parfois inadaptés aux contextes locaux ou à des établissements de taille plus modeste. Leur complexité et le manque de flexibilité dans la personnalisation des fonctionnalités réduisent leur adoption dans des environnements académiques spécifiques, où la variabilité des contraintes est plus importante.

Face à ces limites, la création d'une solution sur-mesure comme SchedU prend tout son sens. En effet, contrairement aux solutions génériques du marché, SchedU est entièrement personnalisée pour s'adapter aux spécificités et aux exigences de l'IFRI, notamment en matière de gestion des salles, des professeurs et des conflits d'horaires. Cela permet d'offrir une réponse plus précise et efficace à des problématiques qui restent insuffisamment adressées par les solutions actuelles.

Notre solution

SchedU est une application web développée pour répondre aux besoins spécifiques de gestion des emplois du temps académiques à l'Institut de Formation et de Recherche en Informatique (IFRI). Conçue comme une solution intuitive, performante et évolutive, SchedU a pour objectif principal de centraliser l'ensemble du processus de planification des cours tout en éliminant les conflits d'horaires, améliorant ainsi l'efficacité de l'organisation académique. Les fonctionnalités principales sont:

- Gestion centralisé des emplois du temps
- Détection automatique des conflits

-
- Création et modification des plannings
 - Notifications et communication
 - Suivi des heures de cours
 - Personnalisation et flexibilité
 - Interface utilisateur intuitive
 - Évolutivité et intégration

Résultats obtenus ant: près la conception de SchedU

La conception et la mise en œuvre de l'application SchedU ont permis d'apporter des améliorations significatives dans la gestion des emplois du temps à l'Institut de Formation et de Recherche en Informatique (IFRI). Les résultats obtenus démontrent une optimisation du processus de planification, une réduction des erreurs et une meilleure coordination entre les différents acteurs académiques.

Tout d'abord, l'une des principales réussites de SchedU est la centralisation des données relatives aux emplois du temps. Grâce à cette plateforme, toutes les informations nécessaires à la création des plannings (disponibilité des professeurs, salles, horaires des cours, etc.) sont réunies au sein d'un même système, accessible en temps réel par les coordinateurs académiques. Cette centralisation permet une vue d'ensemble qui facilite grandement la prise de décisions et réduit les risques d'oubli ou de confusion dans la gestion des ressources.

Ensuite, l'optimisation du processus de planification constitue un autre résultat majeur. Avant la mise en place de SchedU, la gestion manuelle des emplois du temps entraînait régulièrement des conflits d'horaires: des chevauchements de cours, des attributions erronées de salles ou encore des incohérences dans les disponibilités des enseignants. Grâce à l'algorithme intégré dans SchedU, ces conflits sont désormais détectés automatiquement lors de la création des emplois du temps. Cela permet d'éviter des erreurs courantes et de garantir que les plannings respectent les contraintes de chacun. Ce système de validation en temps réel a considérablement réduit le nombre de réajustements nécessaires, ce qui a permis de gagner en temps et en efficacité.

Un autre résultat clé est l'amélioration de la communication et de la transparence entre les différents acteurs. SchedU permet de générer et de partager facilement les emplois du temps avec les professeurs et les responsables de filière. Ces derniers peuvent consulter les plannings en ligne via une interface simple et intuitive, ce qui favorise une meilleure coordination. En outre, les notifications automatiques en cas de modification de planning permettent de tenir tous les utilisateurs informés en temps réel, évitant ainsi les malentendus ou les retards dans l'exécution des cours.

Par ailleurs, l'application permet également de suivre les heures de cours effectuées par les professeurs, offrant une vision claire et précise de la répartition des charges horaires. Ce suivi automatisé des heures permet de calculer les masses horaires restantes pour chaque enseignant et chaque matière, facilitant ainsi la gestion des ressources académiques. Cela contribue à une meilleure répartition des heures de travail, évitant les surcharges pour certains professeurs et assurant une couverture complète des cours planifiés.

Enfin, la facilité d'utilisation de SchedU constitue un atout majeur qui a été salué par les utilisateurs. L'interface utilisateur, conçue avec Vue.js, est intuitive et simple à prendre en main, permettant aux coordinateurs et aux professeurs de gérer leurs tâches sans avoir besoin de

formation technique approfondie. De plus, la flexibilité de la plateforme permet de l'adapter à de nouveaux besoins ou contraintes académiques, garantissant ainsi une évolutivité du système à long terme.

Conclusion et Perspectives

En conclusion, la mise en place de SchedU a permis de répondre efficacement aux problématiques de gestion des emplois du temps à l'IFRI. Grâce à une approche sur mesure et à l'utilisation de technologies modernes comme Laravel et Vue.js, l'application a centralisé et optimisé la planification des cours, réduit les erreurs et amélioré la communication entre les différents acteurs académiques. Les résultats obtenus démontrent une nette amélioration en termes de gain de temps, de réduction des conflits d'horaires, et de suivi des charges horaires des enseignants. Cette solution a non seulement simplifié le processus, mais a également permis une meilleure organisation et coordination au sein de l'institution.

En termes de perspectives, plusieurs axes d'amélioration peuvent être explorés. Il serait pertinent d'envisager l'intégration de nouveaux modules, tels que la gestion des absences ou des imprévus, l'ajout de fonctionnalités collaboratives pour permettre aux enseignants de proposer des modifications de leur emploi du temps, ainsi que la possibilité d'utiliser des algorithmes d'optimisation avancés pour une gestion encore plus performante des ressources. De plus, une extension de l'application à d'autres départements ou établissements offrirait à SchedU un potentiel d'adoption plus large, tout en garantissant sa flexibilité et son évolutivité. Ces évolutions permettront à SchedU de rester une solution pertinente, capable de s'adapter aux besoins croissants de la gestion académique moderne.

Contents

Dedication	ii
Acknowledgements	iii
Abstract	iv
.....	iv
.....	v
List of figures	vi
List of tables	vii
Acronyms and Abbreviations	viii
Glossary	ix
1 Literature review	3
Introduction	3
1.1 Timetable	3
1.2 Overview of TimeTable	4
1.2.1 Personal organization and management	4
1.2.2 School and university planning	4
1.2.3 Company planning	5
1.3 Types of timetable	5
1.3.1 Personal Timetable	5
1.3.2 School timetable	5
1.3.3 University timetable	5
1.4 Importance and benefits of a timetable	6
1.5 Timetable planning at IFRI	6
.....	6
.....	6
1.6 Overview of existing solutions	6
1.6.1 Trello	7
1.6.2 Asana	7
1.6.3 Google Calendar	8
1.6.4 ClickUp	9
1.6.5 TimeTabler	9
1.7 Comparison of existing solutions	10
1.7.1 Trello	10

1.7.2	Asana	11
1.7.3	Google Calendar	11
1.7.4	ClickUp	12
1.7.5	Timetabler	13
1.8	Our solution	14
	Conclusion	14
2	Conception and technical choices	16
	Introduction	16
2.1	Needs Analysis	16
2.1.1	Administrator	16
2.1.2	The coordinator	16
2.1.3	Class supervisors	17
2.2	Solution modeling	17
2.2.1	UML, what is it about?	17
2.2.2	Use case diagram	17
2.2.3	Class diagram	19
2.2.4	Sequence diagram	21
2.3	Technical choices	23
2.3.1	Development methodology	23
2.4	System architecture	24
	Conclusion	25
3	Results and discussion	26
	Introduction	26
3.1	Description of function	26
3.1.1	Authentication	26
3.1.2	The Coordinator	27
3.1.3	Class supervisor	31
3.1.4	The Administrator	33
3.1.5	Others	35
3.2	Alert System (Via E-mail)	36
3.3	Tests	36
3.4	Application deployment aspects	37
3.5	Security aspects	37
3.6	Discussions	39
	Conclusion	39
	Bibliography	41
	Webography	42
	French Summary	44
	Contents	49
