SAED Web Application

Anthony Crespo, Patricio Mendoza

July 11, 2023

1 Introduction

1.1 Problem definition

According to the "Reglamento de Carrera y Escalafón" of Yachay Tech University, academic staff must comply with the following activities:

a) Teaching.

b) Research.

c) Relationship with society.

d) Educational management.

Therefore, at the end of the semester, a report of the activities carried out by the professor must be generated, and it must be supported by the corresponding evidence.

1.2 Justification

Currently, the collection of evidence is carried out manually in Google Drive. Additionally, the final report of each teacher is made based on a Google Forms. However, this makes this process difficult and inefficient. For this reason, the development of a web application that allows automating these processes is of great interest.

1.3 Goals and scopes

- Allow the automation of processes related to teaching activities, such as evidence collection and report generation, through developing a web application for Yachay Tech University.
- Develop a responsive web application that allows the streamlining of processes related to the registration of teaching activities for both for professors and people in charge of supervising these processes.

2 Reference framework

2.1 Web Technology

For developers and enterprises, the decision between web applications and conventional desktop programs has become critical in the current digital era. Unlike traditional desktop applications, which are downloaded and run locally on a user's computer, web applications are run on web browsers and accessed online. To meet user expectations and optimize software development processes, it is crucial to comprehend the key differences between these two application kinds.

2.1.1 Architecture

Traditional desktop applications are often created using programming languages like C++, Java, or C#, whereas web applications are created using web technologies like HTML, CSS, and JavaScript. Web applications use client-server architecture, where the client (web browser) interacts with a distant server to retrieve data and carry out operations. Desktop programs, in contrast, adhere to a standalone design in which all processing takes place directly on the user's PC.

2.1.2 Accessibility

Since they can be viewed from any device having a web browser and an internet connection, web applications provide excellent accessibility. On a variety of hardware (computers, tablets, and smartphones) and operating systems (Windows, macOS, and Linux), users can access web applications. Traditional desktop programs, on the other hand, are platform-specific and must be written independently for each target operating system, which restricts accessibility.

2.1.3 Deployment

Web applications are installed on servers, where changes can be made rapidly without the end user's involvement. Updates can be disseminated effectively and widely thanks to this central deployment strategy. Traditional desktop applications, on the other hand, need to be manually installed and updated on each user's computer, which makes the distribution process more difficult and time-consuming.

2.1.4 User Interface

Web applications make advantage of web technology to provide dynamic and interactive user interfaces. They frequently adhere to the principles of responsive design, adjusting to various screen sizes and orientations. On the other hand, traditional desktop applications have more control over the user interface, offering a richer and more personalized experience. They can use native UI components and have direct access to system resources.

2.1.5 Maintenance

The advantage of centralized maintenance for web applications is that updates and bug fixes can be made on the server, guaranteeing that all users receive the improvements right away. The maintenance of typical desktop applications, on the other hand, is difficult, especially for large-scale deployments, as each user's PC must be updated individually. This decentralized maintenance strategy makes it more difficult to resolve problems quickly and raises the chance of version discrepancies.

2.1.6 Performance

Traditional desktop programs frequently run better because they execute locally on the user's computer and may utilize all of its resources. Contrarily, web applications rely on the user's internet connection speed and the client device's variable processing capability. However, improvements in browser functionality and web technology have considerably reduced this performance disparity.

2.1.7 Security

Web applications' online nature presents particular security challenges. Cross-site scripting (XSS) and cross-site request forgery (CSRF) are two potential vulnerabilities that need to be addressed by developers, along with appropriate authentication and data encryption procedures. Although traditional desktop apps run in a more regulated environment and are less vulnerable to web-specific threats, they nevertheless have some security risks.

2.2 Background

There are a number of significant contenders in the field of web-based applications for academic use that meet various demands and goals. Three different web technologies—Sistema Académico de Evidencia Docente (SAED), Blackboard, and Moodle—will serve as a reference background for the developing technology. While Blackboard and Moodle serve as comprehensive learning management systems (LMS) with broader capability, SAED focuses on offering a customized record system for instructors at Yachay Tech University to record their activities and provide PDF reports. It is better to understand how these technologies meet the unique demands of educational institutions and their stakeholders by looking at the breadth, customization, evidence documentation, reporting capabilities, and target users of each program.

2.2.1 Description of the Background Technologies

Blackboard is a widely used web-based learning management system (LMS) that serves educational institutions worldwide. It provides a comprehensive platform for managing courses, delivering content, and facilitating communication between instructors and students. Blackboard offers features such as course materials management, assignment submission, grading, discussion boards, and online collaboration tools. While Blackboard offers broad functionality for academic purposes, its focus is primarily on supporting teaching and learning activities rather than specific evidence documentation for professors.

Moodle is an open-source learning management system that provides a flexible and customizable platform for creating online courses and managing educational content. Moodle offers features for course management, content delivery, assessments, and communication. It allows instructors to create assignments, quizzes, and discussion forums. Moodle's open-source nature makes it highly customizable and adaptable to various educational institutions' needs. However, similar to Blackboard, Moodle's primary focus is on managing courses and facilitating student learning rather than specifically addressing evidence documentation for professors.

2.2.2 Scope and Focus

- SAED: SAED is specifically designed for Yachay Tech University to record and store professors' activities and generate PDF reports based on evidence. Its primary focus is on providing a system for professors to document their activities and generate summarized reports.
- Blackboard: Blackboard is a comprehensive LMS that focuses on managing courses, delivering content, and facilitating communication between instructors and students. It offers a wide range of tools and features for teaching and learning activities.
- Moodle: Moodle is an open-source LMS that provides a flexible platform for creating online courses and managing educational content. It offers features for course management, content delivery, and assessments.

2.2.3 Evidence Documentation and Reporting

- SAED: SAED focuses on evidence documentation and reporting, allowing professors to upload evidence, categorize it, and generate PDF reports summarizing their activities.
- Blackboard: Blackboard offers tools for managing assignments, grading, and providing feedback to students. While it supports evidence documentation in terms of graded assignments, it doesn't have a dedicated feature for generating summarized reports for professors' activities.
- Moodle: Moodle supports various forms of assessments and assignment submissions, but it does not have a specific feature for evidence documentation and generating summarized reports for professors' activities.

By concentrating exclusively on evidence documentation and reporting for instructors at Yachay Tech University, SAED sets itself apart from the competition. Although Blackboard and Moodle provide thorough learning management systems with a wider range of functionality, they lack specific features for proof documentation and producing reports that are distilled of professors' activities. Yachay Tech University's academics have unique needs, and SAED's specialized approach caters to those needs by providing a unified system for monitoring and reporting academic activity.

3 Methodology

3.1 General approach for implementation

The methodology to follow to carry out this project consists of a total of 5 phases: planning, design, web development, testing and debugging, and deployment, as detailed in Figure 1.

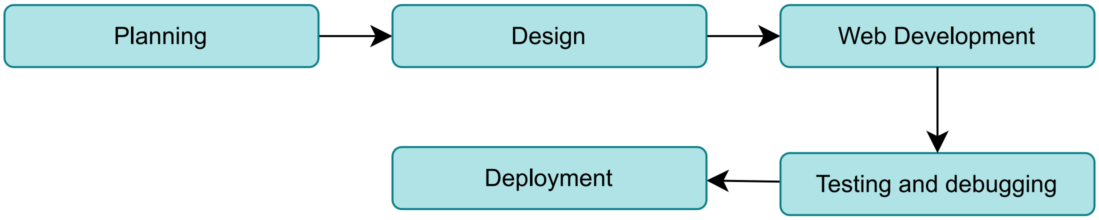


Figure 1: Implementation phases

3.1.1 Planning

In this phase, the objectives, requirements and scope of the project are established after a series of dialogues to find out the problems that want to be solved through the implementation of the web application. In this way, each of the functionalities that must be present on the web page can be defined.

3.1.2 Design

In this phase, we will design each web application view and its components. Likewise, the database design will be carried out to save and recover the data efficiently. The design of the graphical interface will be carried out through the use of Figma, which allows for showing the flow between each view and the interaction with the components of a specific view. Additionally, the technologies, programming languages, frameworks, and types of databases that best adapt to the requirements established in the previous phase will also be defined.

3.1.3 Web Development

In this stage, the development of both the Frontend and the Backend will be carried out using the technologies defined in the previous phase in order to implement each of the requirements defined in the planning phase, together with the design defined in the previous phase.

3.1.4 Testing and Debugging

In this stage, different tests will be carried out in order to verify that the web application works without errors or to identify failures. Likewise, the respective corrections will be made to allow the web application's correct function.

3.1.5 Deployment

This is the final phase of web application development. In this case, it is optional since deploying the web application in a real environment depends on several external factors.

3.2 Tools for Front-end

The tools to be used to develop the Front-end include React, Bootstrap, HTML, and CSS. React is a popular JavaScript library for building user interfaces, allowing for the creation of dynamic and interactive components. Bootstrap, on the other hand, is a widely-used CSS framework that provides pre-built components and responsive design capabilities, enabling developers to quickly create visually appealing and mobile-friendly websites. HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets) are fundamental languages for structuring and styling web pages, respectively.

3.3 Tools for Back-end

The Back-end will be developed using the Django framework, which is built on top of the Python programming language. Django is a powerful and popular web framework that allows efficient handling of tasks such as URL routing, database integration, user authentication, and form handling. Furthermore, we will use PostgreSQL as the database management system for the Back-end development. PostgreSQL is a powerful open-source relational database that offers advanced features such as ACID compliance, concurrency control, and support for complex queries. It provides a reliable and scalable solution for storing and managing data in our web application. By combining the Django framework with PostgreSQL, we can create a robust and efficient Back-end that handles data storage and retrieval seamlessly.

4 Proposal

4.1 Description

SAED (Sistema Académico de Evidencia Docente) was created to meet the needs of teachers at Yachay Tech University. SAED's main goal is to give academics an effective record-keeping system. It does this by enabling them to record their activities during each semester and produce detailed PDF reports.

SAED provides a user-friendly interface that makes for simple navigation and efficient communication. Professors can access their personal dashboards, which give them a comprehensive glimpse of their academic activity, by securely logging onto their accounts. Professors can appropriately describe their actions with the application's support for a variety of evidence categories, including Docencia (Teaching), Vinculación (Vinculation), Administración (Administration), and Investigación (Research).

SAED offers subtypes for each category of evidence that further categorize the uploaded evidence. Professors can record the wide range of activities they take part in throughout the semester using this classification system. Professors can easily keep track of their actions by entering pertinent information such titles, descriptions, dates, durations, and relationships between different types and subtypes of evidence. In order to enhance the recorded evidence, SAED also gives users the option to attach files or documents that support their claims.

The SAED's capacity for automatic report generating is one of its noteworthy features. SAED can produce customized PDF reports detailing each professor's actions after they have uploaded their activity and the semester has ended. Each activity's brief textual description and the amount of hours it took to complete are included in these reports. These papers offer a concrete depiction of professor contributions and make it simple to assess their accomplishments by giving a thorough overview.

Data privacy and security are highly valued by SAED. Professors can rest easy knowing that their online activities and private data are secure because to strong encryption and access controls. To guarantee the data's availability and integrity, regular backups are made.

In addition to making the process of tracking and documenting academics' actions simpler, SAED helps to manage and evaluate academic programs effectively. It facilitates performance evaluation, resource allocation, and strategic decision-making by offering a consolidated store of evidence. Additionally, the application encourages responsibility and openness, establishing a culture of professionalism and constant growth among the faculty.

4.2 Functionalities Description

4.2.1 Functional Requirements

1. User Registration and Login:
   - Professors can create accounts and securely log in to the SAED system.
   - User authentication ensures that only authorized individuals can access and manage their records.
2. Dashboard:
   - Professors are provided with a personalized dashboard upon logging in.
   - The dashboard provides an overview of their activities and easy access to various features and functionalities.
3. Activity Logging:
   - Professors can log their activities for each semester.
   - Activities can be associated with specific evidence types, including Docencia, Vinculación, Administración, and Investigación.
   - SAED supports the classification of activities into subtypes within each evidence type, allowing for accurate categorization.
4. Evidence Upload:
   - Professors can upload supporting files or documents related to their activities, providing additional context or evidence.
   - SAED allows seamless attachment of files to enhance the evidence presented.
5. PDF Report Generation:
   - SAED automatically generates PDF reports at the end of each semester.
   - Reports summarize the activities conducted by professors, including a textual description of each activity and the number of hours spent.
   - The reports offer a comprehensive overview of professors' contributions and achievements during the semester.
6. Activity Tracking and Evaluation:
   - SAED serves as a centralized repository of professors' activities, allowing for effective tracking and evaluation.
   - The system assists in performance assessment, resource allocation, and strategic decision-making.
7. Institutional Management Support:

- SAED contributes to academic management and evaluation within Yachay Tech University.
- The application promotes transparency, accountability, and a culture of continuous improvement among the faculty.

4.2.2 Non-functional Requirements

1. Security:
   - SAED should ensure the confidentiality and integrity of professors' data, preventing unauthorized access or data breaches.
   - It should implement robust authentication mechanisms and secure data transmission protocols.
2. Performance:
   - SAED should have a responsive and performant user interface, providing quick loading times and smooth navigation.
   - It should handle a large number of users and activities efficiently, without significant delays or system slowdowns.
3. Scalability:
   - SAED should be designed to handle increasing data and user loads as the number of professors and activities grow over time.
   - It should support horizontal scalability, allowing for the addition of more servers or resources to accommodate increased demand.
4. Availability:
   - SAED should have a high level of availability, ensuring that professors can access the system whenever needed.
   - It should have minimal downtime for maintenance and upgrades, with provisions for scheduled maintenance windows.
5. Usability:
   - SAED should have an intuitive and user-friendly interface, requiring minimal training for professors to navigate and use the system effectively.
   - It should provide clear instructions and guidance to users, minimizing confusion and errors.
6. Compatibility:
   - SAED should be compatible with various web browsers and operating systems commonly used by professors.

- It should adhere to web standards and accessibility guidelines to ensure compatibility across different devices and assistive technologies.

7. Data Backup and Recovery:
   - SAED should regularly back up professors' data to prevent data loss in case of hardware failures or system issues.
   - It should have mechanisms in place to recover data quickly in the event of any unforeseen data loss incidents.
8. Compliance:
   - SAED should adhere to relevant privacy regulations and guidelines to protect professors' personal information and data.
   - It should comply with any institutional or legal requirements regarding data storage, retention, and protection.
9. Support and Maintenance:
   - SAED should have a support mechanism in place to assist professors in case of any issues or questions they may have.
   - It should have a well-defined maintenance process to apply updates, bug fixes, and enhancements regularly.
10. Performance Monitoring and Analytics:
    - SAED should have monitoring capabilities to track system performance, identify bottlenecks, and proactively address any performance issues.
    - It should provide analytics and reporting features to gain insights into system usage, user activity, and performance metrics.

4.3 Data Model (E-R Model)

The database model for SAED is designed to efficiently store and manage the data related to professors' activities, evidence types, subtypes, and generated reports. The model consists of several interconnected tables that represent different entities and their relationships. Here's a description of the main entities and their attributes:

1. Semester
   - semester_id , semester_name , date_start , date_end
2. School
   - school_id , school_name , school_abbreviation
3. Career

- career_id , career_name , school_id

4. Professor_Denomination
    - professor_denomination_id , denomination
5. Professor
    - professor_id , professor_degree , professor_names , professor_lastnames , career_id , professor_denomination
6. Semester_School
    - semester_id , school_id , dean_id
7. Semester_Career
    - career_id , semester_id , coordinator_id
8. Activity_Type
    - activity_type_id , activity_type
9. Evidence_Type
    - evidence_type_id , evidence_type , activity_type
10. Document
    - document_id , document_comment , document_uploadDate , document_pathToFile , professor_id , activity_type , evidence_type , semester_id
11. Report
    - report_id , teaching_report_summary , teaching_report_hoursPerWeek , teaching_report_hoursPerWeekIntersemester , management_report_summary , management_report_hoursPerWeek , management_report_hoursPerWeekIntersemester , vinculation_report_summary , vinculation_report_hoursPerWeek , vinculation_report_hoursPerWeekIntersemester , investigation_report_summary , investigation_report_hoursPerWeek , investigation_report_hoursPerWeekIntersemester , report_name , report_uploadDate , report_professorComment , report_revisorComment , report_conclusion , report_isApproved , report_pathToFile , professor_id , semester_id , report_reviewedBy , report_approvedBy

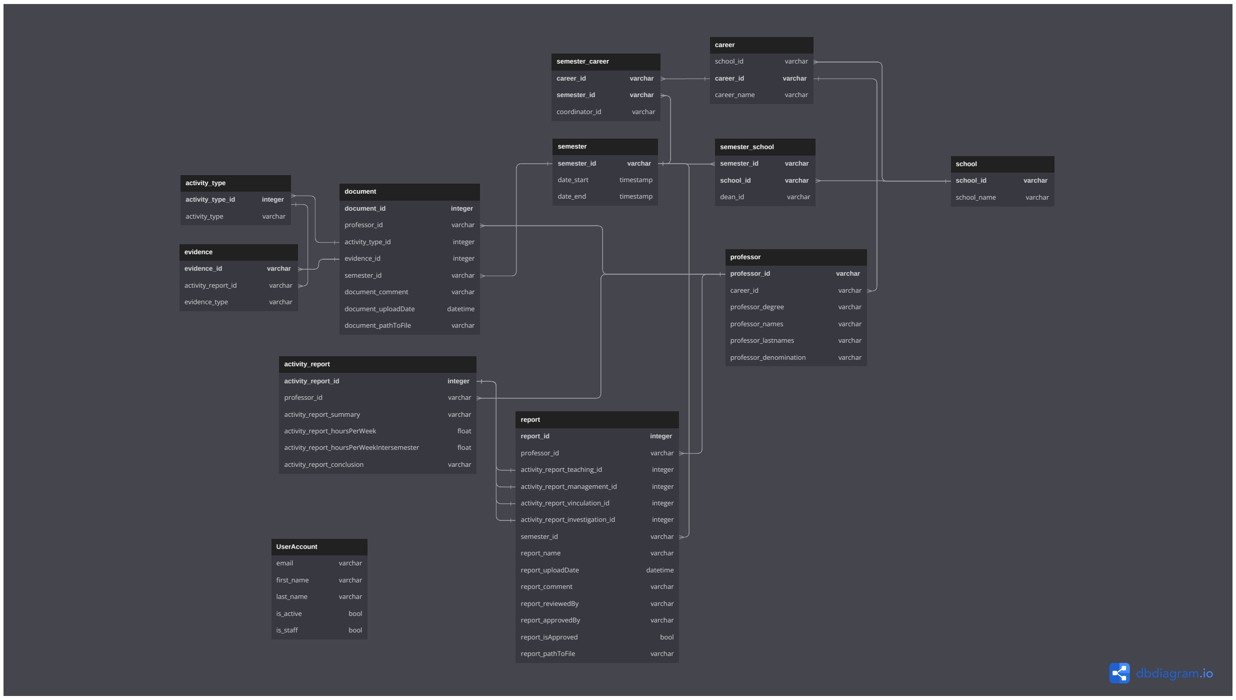
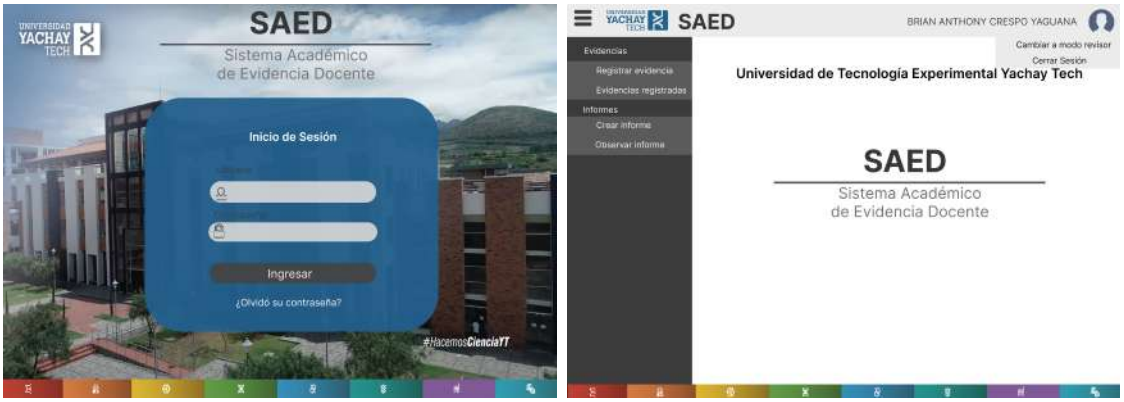


Figure 2: Database General Model

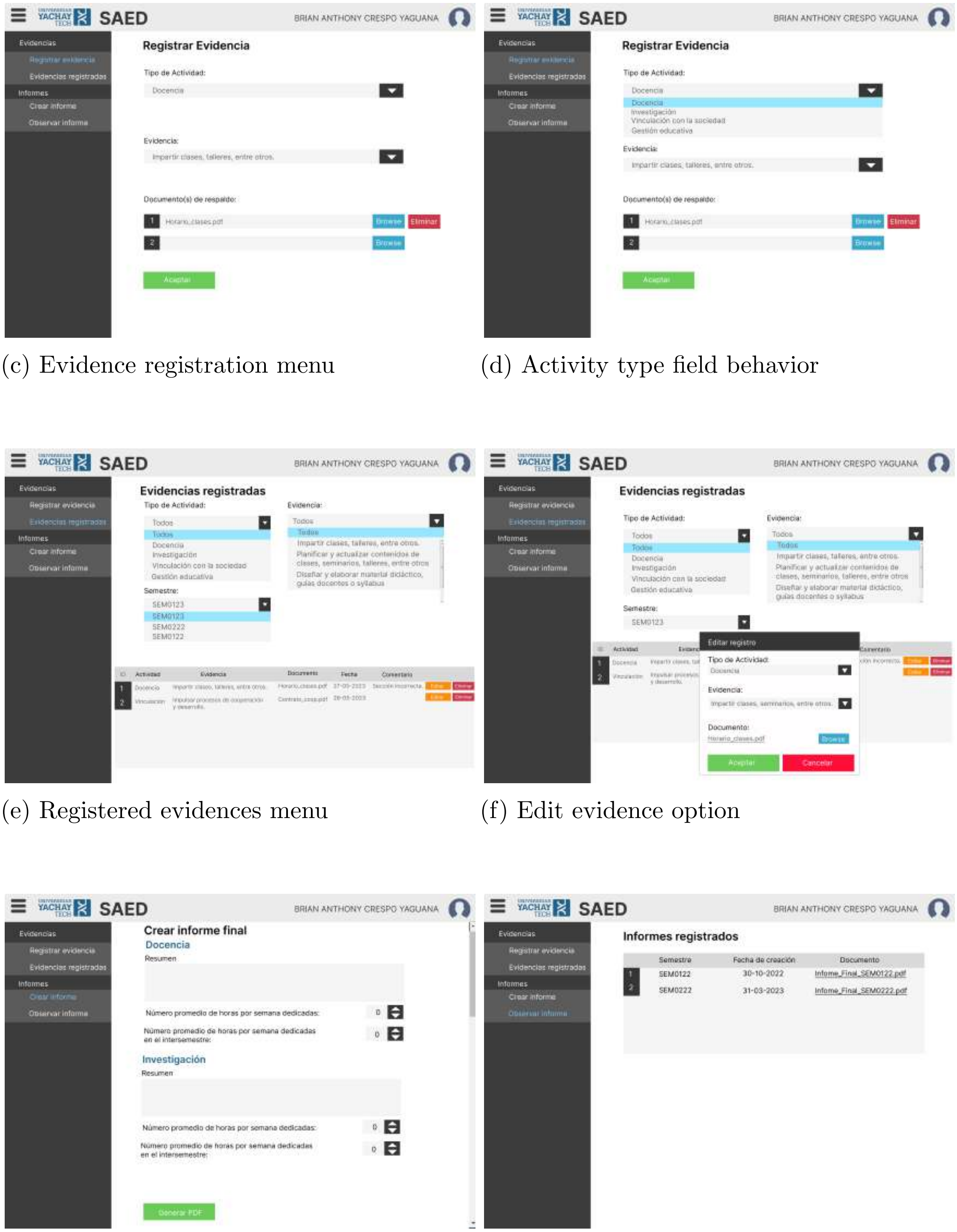
4.4 Prototype

4.4.1 Basic interactions



(a) Login

(b) Home



ID	Actividad	Evidencia	Documento	Fecha	Comentario		
1	Docencia	Impartir clases, talleres, entre otros.	Horario_clases.pdf	27-03-2023	Sección incorrecta.	Editar	Eliminar
2	Vinculación	Impulsar procesos de cooperación y desarrollo.	Contrato_coop.pdf	28-03-2023		Editar	Eliminar

	Semestre	Fecha de creación	Documento
1	SEM0122	30-10-2022	Infome_Final_SEM0122.pdf
2	SEM0222	31-03-2023	Infome_Final_SEM0222.pdf

(c) Evidence registration menu

(d) Activity type field behavior

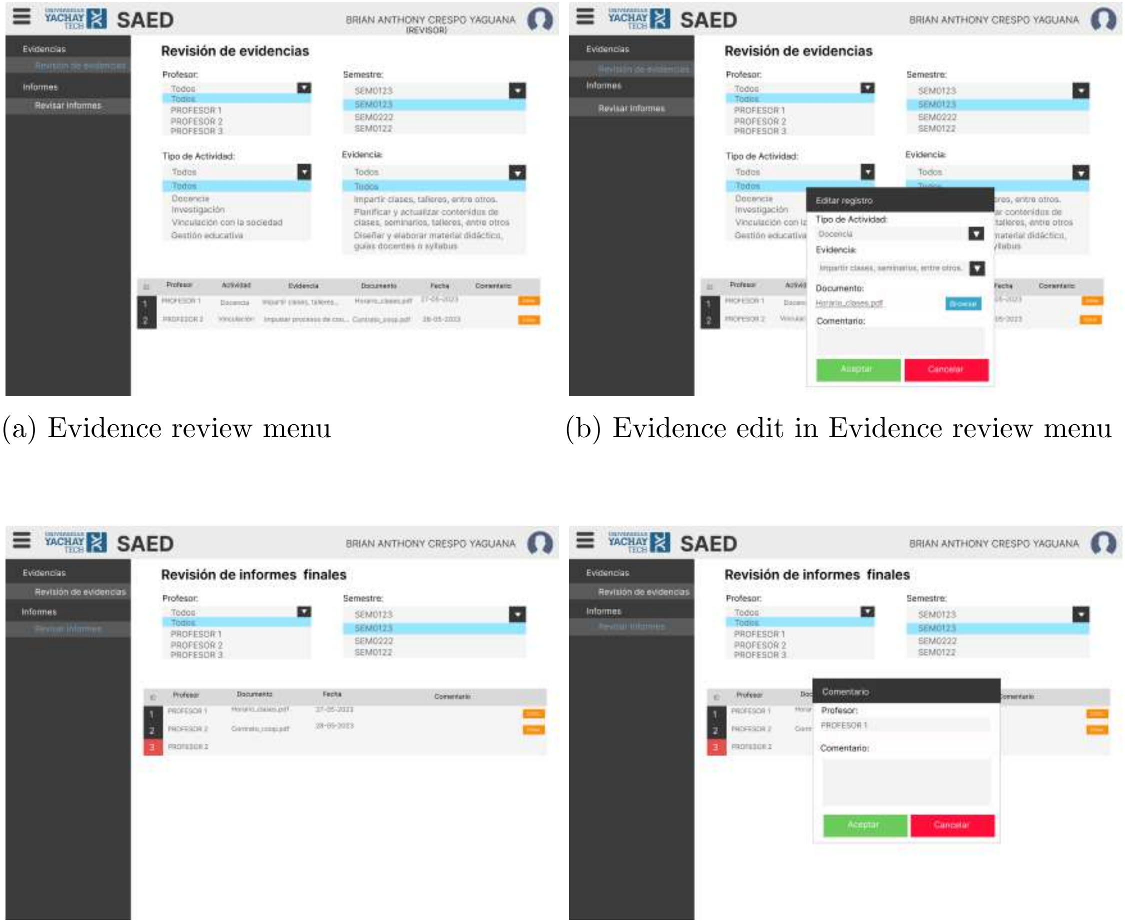
(e) Registered evidences menu

(f) Edit evidence option

(g) Create report menu

(h) Created reports menu

Figure 3: Professor views



ID	Profesor	Actividad	Evidencia	Documento	Fecha	Comentario
1	PROFESOR 1	Docencia	Impartir clases, talleres...	Horario_clases.pdf	27-05-2023	
2	PROFESOR 2	Vinculación	Impulsar procesos de coo...	Contrato_coop.pdf	28-05-2023	

ID	Profesor	Documento	Fecha	Comentario
1	PROFESOR 1	Horario_clases.pdf	27-05-2023	
2	PROFESOR 2	Contrato_coop.pdf	28-05-2023	
3	PROFESOR 2			

(a) Evidence review menu

(b) Evidence edit in Evidence review menu

(c) Report review menu

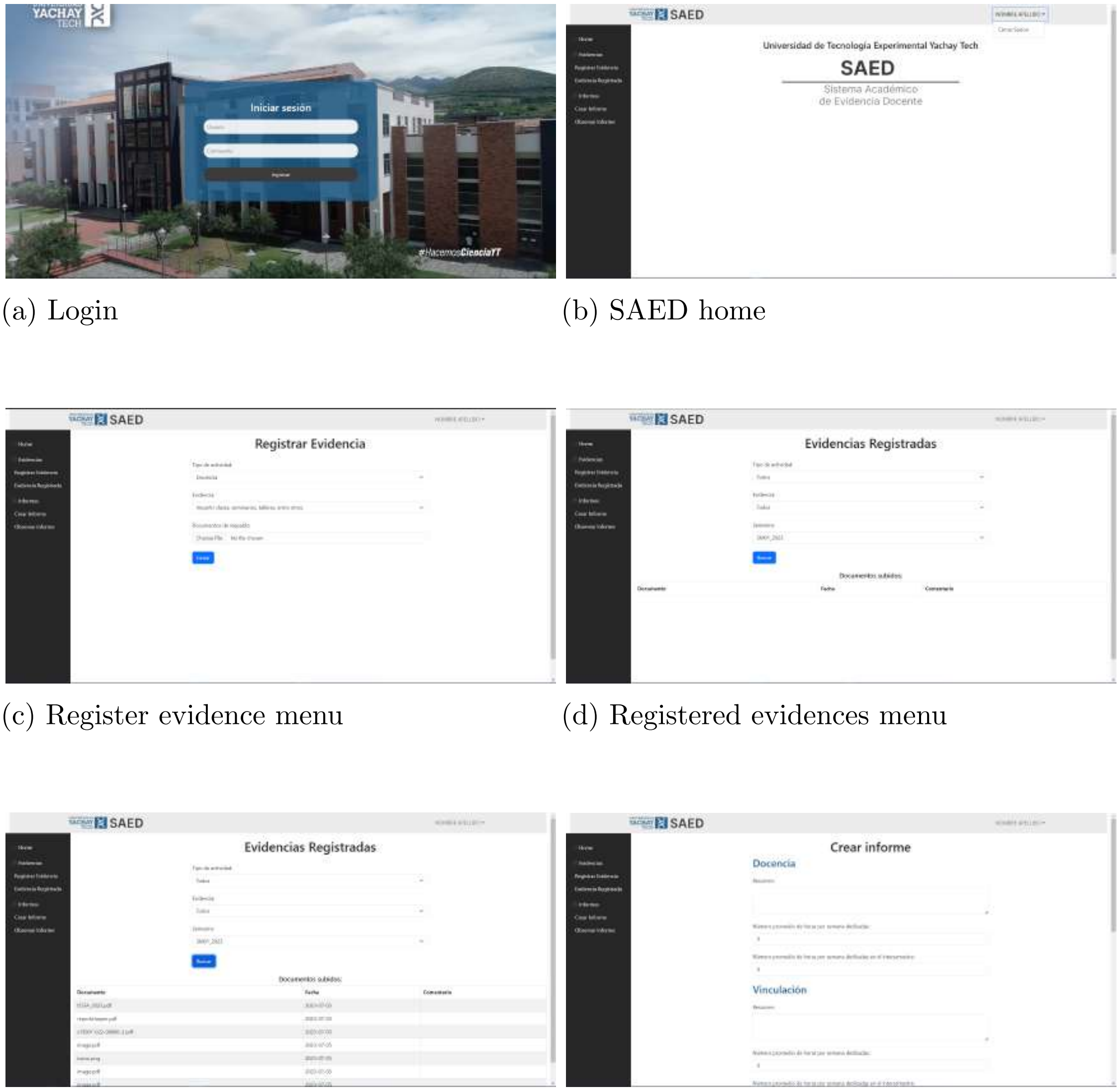
(d) Report edit in Report review menu

Figure 4: Revisor views

Figure 3 shows the designs of the views of the web application for a teacher user, who will only have permission to upload evidence, observe and edit their evidence, create reports, and finally, edit their final report. Figure 4 shows the views for a user with the role of reviewer, who will have permissions to access and edit the evidence and reports uploaded by a specific group of professors. It is important to mention that at Yachay Tech, a reviewer is also a teacher, so the role change will be done by clicking on the username and then choosing the desired role, as shown in Figure 3b. Specific options are displayed in the left side menu depending on one role or another.

4.4.2 Current version

Front-end



(a) Login

(b) SAED home

(c) Register evidence menu

(d) Registered evidences menu

(e) Registered evidenced search

(f) Create final report menu

Figure 5: SAED current Front-end

Figure 5 shows the current state of SAED Front-end implementation. Until now, the login view, the home page, and the menus corresponding to the creation of evidence, observation of registered evidence, and creation of a final report have been implemented. Except for the login form, these forms work together with the API to retrieve and insert new information in the database.

Back-end

SAED's backend is implemented using Django, a high-level Python web framework known for its robustness, security, and scalability. Django follows the Model-View-Controller (MVC) architectural pattern, where models represent the data structures, views handle the logic and interactions, and templates define the presentation layer.

Models:

- SAED's backend utilizes Django's Object-Relational Mapping (ORM) to define the database models that represent the entities in the system.
- Models such as Professor, Semester, Evidence_Type, Activity_Type, Document, and Report are implemented as Django models with their respective attributes.

Views:

- Django's views handle the logic and business operations for processing incoming requests and generating responses.
- Views are responsible for tasks such as handling user authentication and authorization, validating and processing data, and generating appropriate HTTP responses.
- SAED's backend utilizes Django's class-based views or function-based views to define the logic for different actions, such as creating activities, generating reports, and managing user sessions.

URLs and Routing:

- Django's URL routing mechanism maps incoming requests to the corresponding views.
- SAED's backend defines URL patterns using Django's URLconf, which maps specific URLs to the appropriate views and actions.
- The routing system ensures that requests are directed to the correct view functions for processing.

Forms and Validation:

- SAED utilizes Django's forms to handle user input and perform data validation on the backend.
- Django's form system allows for defining form fields, applying validation rules, and rendering forms in the frontend.
- The backend validates the submitted form data, ensuring data integrity and security.

Authentication and Authorization:

- Django provides robust authentication and authorization mechanisms out-of-the-box.
- SAED's backend utilizes Django's built-in authentication system to handle user registration, login, and session management.

- User roles and permissions are used to control access to specific views or actions within the application.

API Development:

- SAED's backend provides an API (Application Programming Interface) using Django's REST framework.
- The RESTful API allows for integration with other systems, providing data access and functionality to external applications.

4.5 Deployment

4.5.1 How to install and use

Basically, in order to run the SAED project it is necessary to have node.js and Python 3 installed. Once these requirements are met, the project can be cloned from the `https://github.com/AnthonyCrespo/Professor_Evidence/` repository. To run it, we need to have two terminals open. In the first one, we will access the main directory, in which the `requirements.txt` file is present. Preferably, using a virtualenv, we will do a `pip install -r requirements.txt`. That will install the Python 3 libraries the project requires, including Django. Once the libraries are installed, we will execute the `python manage.py runserver` command. Thus, the backend of the application that contains the API will be working, which will allow interaction of the Front-end with the database. In the second terminal we access the client directory and proceed to execute the `npm install` command, which will install all the dependencies related to node. Once this step is completed, we execute the `npm run dev` command, and thus the front server will be raised.

5 Test Plan

5.1 General Description

The purpose of this test plan is to outline the approach and strategies for testing a web application. The goal is to ensure the application's functionality, usability, performance, and security meet the desired standards.

5.2 Test Types (What and how?)

Functional Testing:

- Validate the functionality of SAED's core features, such as evidence upload, activity logging, report generation, and user management.
- Test individual functions and their interactions to ensure they perform as expected.

- Verify that the application meets the specified functional requirements.

Usability Testing:

- Evaluate the user-friendliness and ease of use of SAED's interface.
- Test the application's navigation, layout, and overall user experience.
- Identify any usability issues or areas for improvement to enhance the user interface.

Security Testing:

- Assess the security measures implemented in SAED to protect professors' data and ensure data privacy.
- Test for vulnerabilities such as SQL injection, cross-site scripting (XSS), and authentication and authorization issues.
- Verify that proper access controls are in place and sensitive data is securely stored and transmitted.

Performance Testing:

- Evaluate the performance and responsiveness of SAED under expected loads and stress conditions.
- Test the application's response time, scalability, and resource usage.
- Identify potential bottlenecks, performance issues, or areas for optimization.

Compatibility Testing:

- Verify the compatibility of SAED with different web browsers and operating systems commonly used by professors.
- Test the application's functionality and appearance across different browser versions and platforms.
- Ensure a consistent experience for users regardless of their chosen browser or operating system.

Integration Testing:

- Test the integration between different modules or components within SAED.
- Verify that data flows correctly between various parts of the application.
- Identify and resolve any integration issues or inconsistencies.

6 Scope of the current phase of development

The current version of the SAED (Sistema Académico de Evidencia Docente) project will not encompass the complete set of planned functionalities due to time constraints. However, the development efforts have prioritized and focused on implementing key features essential for its core functionality.

It is expected that, at the end of this stage, the final version of SAED will include the following features:

Professor and Revisors Sessions: SAED includes a user authentication and session management system to ensure secure access to the application. Professors can create accounts, log in securely, and access their personalized dashboards to manage their activities effectively. Furthermore, SAED includes a toggle mechanism between professor and reviewer roles. Revisors can approve and comment on other professors' evidence for the reports to be finally approved by the dean.

Evidences and Report Registration and Actualization: Professors can register and edit their activities across the four evidence types: Docencia, Vinculación, Administración, and Investigación. Each of the evidences that are uploaded must be in PDF format. The objective is to allow uploading one or more PDFs simultaneously for the same activity and type of evidence. Professors can also generate their semester final reports by filling in the fields corresponding to each activity type. These reports can later be deleted or updated by the person who created them.

Review of evidence and final reports: The reviewers have access to the evidence uploaded by each of the teachers. In addition, they can comment on and edit certain information in those records. Additionally, they can approve or comment on the final reports generated by each professor. Each necessary form to register information will include field validations to ensure that the stored information in the system is accurate and appropriate.

PDF Generation: The necessary efforts will be made to allow the generation of a PDF of the final report based on the evidence and information provided by each teacher. However, the feasibility of implementing this feature is still under discussion given the short time available.