



FACULTY OF COMPUTER SCIENCE, ELECTRONICS
AND TELECOMMUNICATIONS OF AGH UNIVERSITY
OF SCIENCE AND TECHNOLOGY

SOFTWARE ENGINEERING

2023-2024 SM1

TEAM 4

ERASMUS EVENTS MANAGER APPLICATION

Julio del Junco Prieto	jprieto@student.agh.edu.pl	Universidad Rey Juan Carlos
Martin Díaz-Benito Álvarez	mdbalvarez@student.agh.edu.pl	Universidad Rey Juan Carlos
Marta Cuevas Rodríguez	martacuevas@student.agh.edu.pl	Universidad de Málaga

Versions of the document

Version	Date	Details
v1	16/10/2023	Creation of the document Writing until 1.2
v2	19/10/2023	Writing until 1.6.2
v3	26/10/2023	Writing until 2.1
v4	08/11/2023	Writing until 2.3
v5	16/11/2023	Writing until 2.4
v6	23/11/2023	Writing until 2.5 (Use case diagrams)
v7	30/11/2023	Writing until 2.5 (Activity diagrams)
v8	06/12/2023	Writing until 2.6
v9	13/12/2023	Writing until 2.7
v10	21/12/2023	Fixing Activity Diagram
v11	1/11/2024	Writing until 4 and minor fixes
v12	17/1/2024	Finishing the project

Table of Content

1. Project Description.....	4
1.1 Area and Subject of Modeling.....	4
1.1.1 Problem domain on the basis of an organizational schema.....	5
1.2 Modeling area.....	6
1.2.1 Description which part of organization structure.....	6
1.2.2 Posts description.....	6
1.3. System's responsibilities' identification of domain activity areas.....	8
1.3.1. Identification of the Domain Activity Areas.....	8
1.3.2 Description of all interesting business procedures' in the named Activity Areas.....	9
1.4 Verified system's responsibilities.....	9
1.5 Short project name.....	10
1.6 Goals of the project.....	10
1.6.1 Product's aims.....	10
1.6.2 Design aims.....	10
2. Requirements description.....	11
2.1 Functional requirements.....	11
2.2 Requirements for data.....	12
2.3 Requirements for interface.....	14
2.4. Quality requirements, organizational needs, collaboration and the external factors.....	15
2.5 Functional requirements' detail analysis and Use Case Diagrams.....	16
2.6 Functional requirements for additional system functions.....	43
2.7 Non-functional requirements - a systematic approach.....	44
3. Analysis of data structure.....	45
4. Model of system's behavior in time.....	46
5. Comparison of models.....	47
6. System's Architecture.....	48
7. Project of the interface.....	48
8. Testing.....	48
9. Implementation description.....	52
10.Summary and conclusions.....	1
10.1 Verification of the project.....	1
10.2 Verified assumptions for implementation.....	1
10.3 Final remarks and conclusions.....	1

1. Project Description.

1.1 Area and Subject of Modeling- Julio del Junco Prieto

Nowadays, there is really no way to enroll online to the ESN events other than by links in a WhatsApp group, for so, we have thought of creating an app which administers the events and lets the members register to them.

The app will let the user see the current events in which they can join as well as the previous events in which they participated. Once you enter an event it will have a tab with the participants and another one with documents, which can be a presentation if needed for the event or the photos and videos gathered through the event.

In order to join an event, the user will have to go through a process of registration to the event in which they will have to put their name, their cellphone number, their ID or Passport and they will have to pay if needed. For the payment, it will appear a pop-up after the other information is filled and it will process the payment with their bank account.

There is also somebody who is responsible for organizing events, providing a venue, additional staff if needed and the plan itself. These event organizers can propose events on the app, see statistics about them to decide whether they are profitable or not, and add footage taken after events.

On the management side, the manager decides what events are taking place, approving or rejecting them and posting those in the app after approval. They could also see all participants.

1.1.1 Problem domain on the basis of an organizational schema-Julio del Junco Prieto

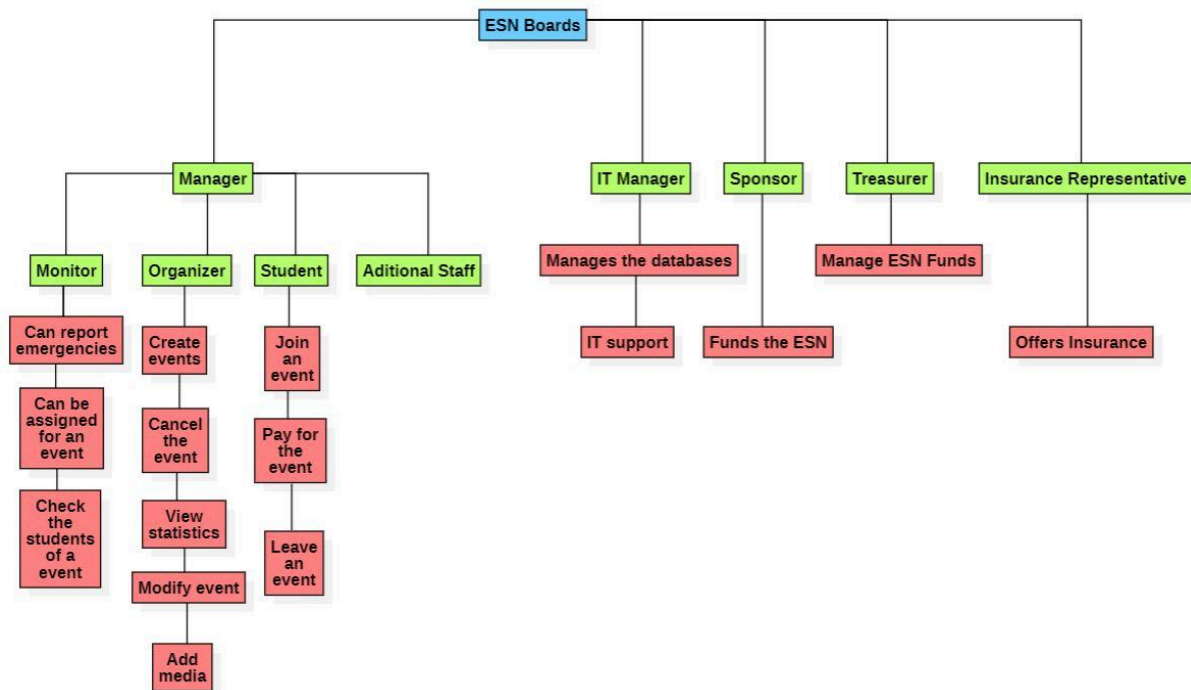


Figure 1 - Organizational schema

This is what each role makes in the system:

Manager: The manager will see the upcoming events with their participants which they could throw if needed, as well as the past events. Also, as administrators, they will be responsible for approving the different events proposed by organizers.

Organizer: The organizer will be responsible for creating the events with a name, a date, a location, a description and the price if it is not free. After that, they are posted on a specific tab for them, marking them as pending approval. When approved they will be able to see the statistics of the current number of students signed and money collected. They could also cancel the events if there aren't enough participants, and upload videos and pictures taken after the event.

Monitors: Monitors are there to make sure everything is okay, and can check how many students show up to the events (to gather analytics) and report emergencies or faults, in the monitor area of the app.

Students: The students are going to be able to enroll in the events themselves with a registration and they are going to be able to pay for the event in case the event is not free.

Treasurer: The treasurer manages the money of the organization, the payments made by partnerships and those made by the students that join the events. They also manage the payments made by the association to the volunteers to organize the events.

Insurance Company: The insurance company will offer insurance to the students of the events, by paying an extra fee when they register to the event.

Sponsor: The ESN will look for sponsors, which could partially or completely fund the events in exchange for some kind of advertising, or offer some gifts in the events as well as exclusive discounts and offers.

Admin of IT systems: The admin of IT systems will be in charge of all the computer infrastructure, including databases, system maintenance and security and taking care of IT support.

1.2 Modeling area

1.2.1 Description which part of organization structure - Marta Cuevas Rodríguez

The app we are planning on creating is going to be responsible for the administration of the events and for the actors appearing on them.

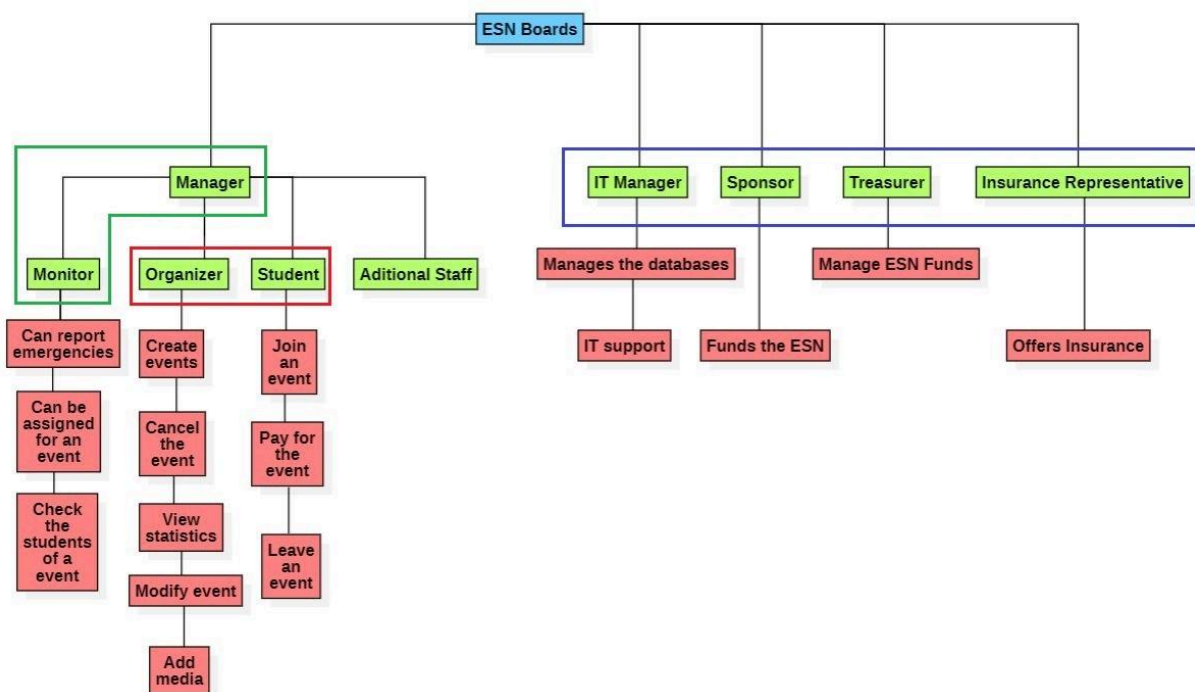


Figure 2 : Organizational schema

It will be divided in three parts, the first one with the economic staff and the IT manager it will be called Economic & IT Staff(DA-01) and it will be done by Marta Cuevas, the second one, with Manager and Monitor it will be called Management Staff(DA-02) and it will be made by Juilo del Junco, and the last one with Organizer and Student, it will be called Student Organization(DA-03) and it will be made by Martín Diaz-Benito.

1.2.2 Posts description- Martín Diaz-Benito Álvarez

I, as a student, want to be able to browse and view a list of available events, select the ones that interest me, and register for them so that I can participate in the events.

I, as a student, want to be able to make payments within the app if an event requires payment, so that I can complete the registration process and secure my spot at the event.

I, as a student, want the option to cancel my registration for an event if I am unable to attend, and if applicable, receive a refund, so that I can manage my participation effectively.

I, as a monitor, need access to a list of students who have registered for an event, so that I can check attendees and ensure the event's smooth operation and safety.

I, as a monitor, need the capability to report emergencies through the app during events to alert managers and take necessary actions, ensuring the safety and well-being of attendees and improving safety for future events.

I, as an organizer, need the ability to cancel an event if unforeseen circumstances or issues require its cancellation, so that attendees can be informed and necessary actions can be taken.

I, as an organizer, want to create and submit event proposals within the app, providing details and descriptions, so that students can browse and join the events I organize.

I, as an organizer, want access to event statistics and metrics, including the number of attendees, revenue, and other relevant data, to assess the success and areas for improvement in my events.

I, as an organizer, want the ability to update event details, such as the event date, location, or description, in case there are changes or improvements to be made.

I, as the treasurer, need access to a financial management feature within the app to manage and record ESN funds related to event payments and expenses accurately.

I, as an insurance company representative, want to provide insurance offers within the app to students, allowing them to purchase event insurance to cover potential risks and issues.

I, as the IT Manager, am responsible for overseeing the IT infrastructure, ensuring that the application operates smoothly, and managing server configurations, updates, and overall system reliability.

I, as the IT Manager, am responsible for the safe and efficient management of user databases, ensuring that student data remains secure and accessible to authorized users, including students, monitors, organizers, and other stakeholders.

I, as a student, want to be able to register an account in the app by providing my name, email, and a password.

I, as any type of user, want to be able to reset my password if I forget it by receiving a password reset link via email.

I, as a student, want to use search filters to find events by location, date, and event type, to discover events that suit my interests and schedule.

I, as a student, want to click on an event to access detailed information, including the event's location, date and time, a description, and information about the organizers.

I, as a student, want to receive a confirmation email after successfully registering for an event, which includes event details and payment information.

I, as a student, want to receive event reminders and notifications, such as email or push notifications, to ensure I don't miss events I've registered for.

I, as a manager, want to be able to approve or reject event proposals submitted by organizers, ensuring they meet the application's standards for safety and quality.

I, as any type of user, want to see a list of upcoming events along with event details and participant information to oversee and coordinate event activities.

I, as any type of user, want a secure login process, requiring my username and password, to access my account securely and conveniently.

I, as any type of user, want to have a log out process, to make sure nobody else has access to my account.

1.3. System's responsibilities' identification of domain activity areas

1.3.1. Identification of the Domain Activity Areas -Marta Cuevas Rodríguez

Domain Area 01: Economic & IT Staff (DA-01) - Managed by Marta Cuevas

In DA-01, the focus is on financial management, insurance, sponsorships, and IT support. The Treasurer handles financial transactions, budgets, and payments, while the Insurance Representative offers insurance options for events and addresses related queries. Sponsors are sought to fund or support events, and the Admin of IT Systems is responsible for maintaining the app's IT infrastructure, data security, and providing technical support.

Domain Area 02: Management Staff (DA-02) - Managed by Julio del Junco

DA-02 oversees event management. The Manager is responsible for approving and overseeing events, while Monitors ensure smooth event operations and collect attendance data. They collaborate to ensure that events run effectively and efficiently.

Domain Area 03: Student Organization (DA-03) - Managed by Martín Diaz-Benito

In DA-03, the focus is on organizing events and student participation. Organizers create and manage events, including setting event details. Students enroll in events and make payments if required. This domain area handles the logistical aspects of event creation, organization, and student engagement.

1.3.2 Description of all interesting business procedures' in the named Activity Areas - Martín Diaz-Benito Álvarez

Firstly for **Economic & IT Staff**, the app must support financial management, event insurance, sponsorships, and IT infrastructure. The Treasurer requires a financial management dashboard to record payments, categorize expenses, and generate financial reports. The app should also facilitate secure online payment processing. The Insurance Representative needs a feature to offer event insurance during registration, including a claims management system. Sponsors should register or apply for sponsorships within the app, with dedicated spaces for sponsor advertising. The Admin of IT Systems manages infrastructure, with access to the app's backend for tasks like database administration and system maintenance. A helpdesk or support ticket system should also be provided for users to report technical issues.

In the **Management Staff**, the focus is on event management and monitoring. The Manager needs a dashboard for event approval, scheduling, and management, with event creation functionality. Monitors require real-time access to event data and the ability to report emergencies.

The **Student Organization** domain area centers around event creation, student enrollment, and event organization. Organizers need access to an event creation portal, where they can input event details, set pricing, and submit events for approval. The app should provide event templates and offer event statistics, such as registration numbers and financial data. Students should be able to easily browse and enroll in events through the app, with a straightforward registration process. Additionally, organizers should have the tools to manage event details, monitor registrations, and track student check-outs in real-time for accurate attendance tracking. Lastly, they can upload the footage of previous events to the database if any was taken. These technical features ensure the app supports the business procedures effectively, creating a user-friendly and streamlined experience for all users involved in event management and participation.

1.4 Verified system's responsibilities- Marta Cuevas Rodríguez

We have some different areas of responsibility in our app that correspond with the subdomains of our app. The management area of responsibility is crucial for managing events. It encompasses event approval and event details, as well as it oversees the technical aspects of the application, including system infrastructure, security, and maintenance. The student's responsibilities are related to payment, attendance and

registration for events. The economy staff is responsible for payments and funding for events. The other area of responsibility is the Events staff, who make events happen, oversee, create and cancel events. The system's responsibilities are mainly: handling users (registration of students, monitors, authentication, administration of accounts,...), handling events (publishing events, signing up for events,...), calculation of different metrics and statistics for event organizers, reporting emergencies and storing and showing all footage taken during the events and managing payment for different events.

1.5 Short project name- Julio del Junco Prieto

Erasmus Event Management Application E.E.M.A

1.6 Goals of the project

1.6.1 Product's aims - Martín Diaz-Benito Álvarez

The software we are developing has some aims we want to cover.

First of all, the necessity of organization for those students that are interested in joining new activities. We will provide an user-friendly platform that simplifies the process of organizing, managing, and participating in ESN events, with sort of options with everything already arranged where each person could select as much as they want. They do not have to book any place or trip, the activities are designed just to join on it.

Furthermore, we simplify the search for people to sign up for insurance companies and also for owners of the places we book.

Monitors will have the control of every single person signed on each activity in the application. They will see how many people there are, their information and when they join.

For organizers, the application simplifies their work providing the possibility to open a new activity, change one that is already open or close those that are finished. It also offers tools for event organizers and managers to gather and analyze data about events, enabling better decision-making and improved event planning.

1.6.2 Design aims - Martín Diaz-Benito Álvarez

The application should have an intuitive and user-friendly interface that is easy for all users to navigate and understand. It also has to implement robust security measures to protect user data and financial information. It must enable integration with other systems such as payment gateways, insurance providers, or event scheduling platforms.

2. Requirements description

2.1 Functional requirements - Martín Diaz-Benito Álvarez

To develop the application, we need to have in mind the required functions for each user of our software.

- **Sign in.** All users must sign in to use the app by filling in their name, surname, birth date, email address and phone number, as well as a password.
- **Log in.** All users must put their email address as well as their password in order to log in.
- **Log out.** The users will be able to log out of the app by clicking a button on the screen.
- **Reset password.** If a user needs to, they can change their password in order to log in.
- **See event.** The users will be able to see the information about an event when they click on it. This includes: name of the event, when, where, list of students, and a photo about it.
- **Search bar.** All users will have a search bar in the main screen in which they will be able to search events and filter them by time, place, or payment needed.
- **Join an activity.** To complete this task, the student must introduce their personal data in the application system for the event they want to sign up for, such as their name, surname, phone number and email address.
- **Resign an event.** To leave an event, the student should describe the reason, and delete their sign in the activity.
- **Pay for an event.** This function only would be necessary in those events that there is a cost to join in. In those cases, students have to introduce their bank account number or their credit card, the date of expediting of the card and CVV; after joining the event.
- **Consult activities.** This function will allow all the users to consult news in the system such as new activities, or changes in events.
- **Event Proposal:** Organizers can submit event proposals through the app, providing details about the event, such as name of the event, a brief description, its purpose, logistics, and costs.
- **Event Management:** If their event proposals are approved, organizers are responsible for planning, organizing, and executing the events, including promoting them and managing attendees, as well as canceling events and notifying students in that case, and refunding their money; or even modify any event that needs to be uploaded.
- **Content Submission:** Organizers can upload event-related content, such as photos and videos, to share with attendees after the event.
- **Metrics and Feedback:** Organizers can access event metrics (e.g., attendance and revenue) and receive feedback from attendees, helping them improve future events.
- **Emergency Reporting:** If they encounter emergencies or problems during events, monitors should promptly report them through the app to managers, explaining in detail what the problem is.

- **Check-in:** Monitors should make sure all whether students signed up for the events have shown up or not.
- **Event Approval:** Managers are responsible for reviewing event proposals submitted by organizers. They can approve or reject these proposals based on certain criteria, such as feasibility, safety, and alignment with the goals of the application
- **Emergency Response:** In the event of an emergency reported by monitors, managers coordinate and respond appropriately, taking actions to ensure the safety and well-being of attendees.
- **Control the application.** Checking by the IT manager if the functioning of the app is the correct one, if everything runs the way it should.
- **Database support.** The IT manager is the person in charge of the databases of the system.
- **Funds events.** The sponsors must pay for the development of the current events, giving the necessary money.
- **Manage Funds.** Treasurer is the person in charge of controlling and checking the correct distribution of the money funds. He must give each section the percentage that corresponds depending on expenses.
- **Offers insurance.** The insurance representative must provide users mostly with medical insurance .
- **Mail confirmation.** The app will send an email confirming the users that they have registered to an event as well as a reminder email the day before the event

2.2 Requirements for data - Marta Cuevas Rodríguez

According to our system, there is some data that need to be stored:

Students:

- Name
- Surname
- Student ID
- Email Address
- Phone number
- Bank account Number

Monitors:

- Name
- Surname
- Phone Number
- Email Address
- Role associated
- Events related
- Emergency reports history

Organizers:

- Name
- Surname
- Email Address
- Phone Number
- Events related

Manager:

- Name
- Surname
- Phone number

IT Manager:

- Name
- Surname
- Contact phone number

Sponsor:

- Name of the company
- Contact number

Treasurer:

- Name
- Surname
- Contact number

Event:

- Name
- Description
- Price
- Date
- Media
- Organizer
- List of participants
- Monitors list

2.3 Requirements for interface - Marta Cuevas Rodríguez

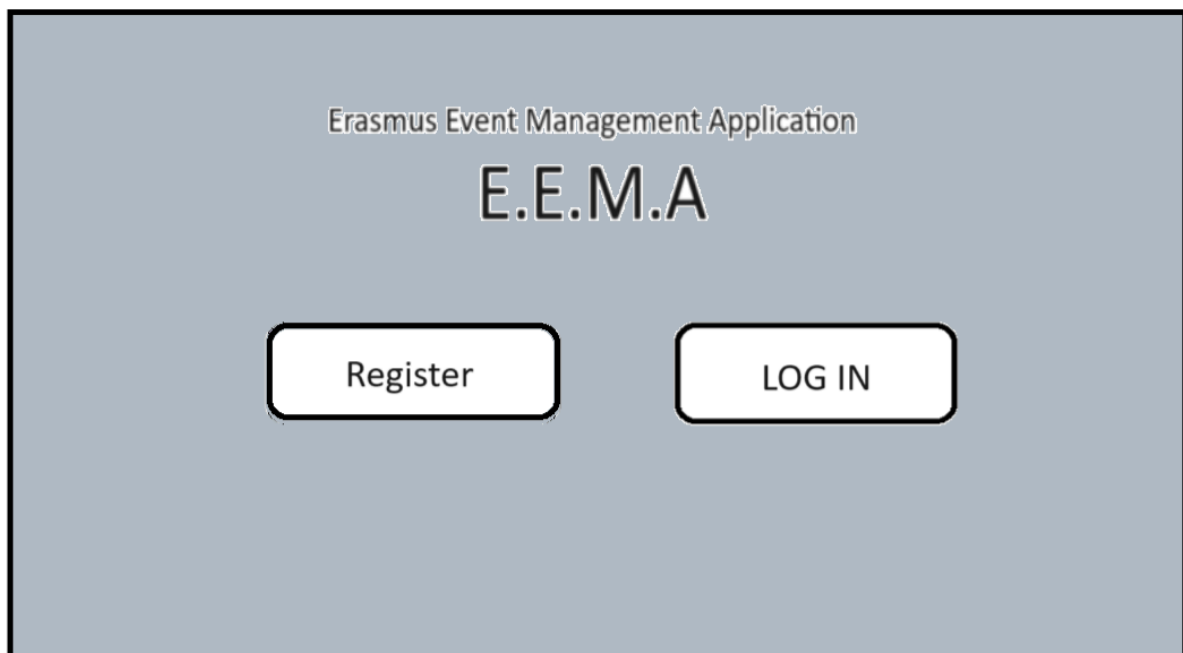


Image 1: Login screen example

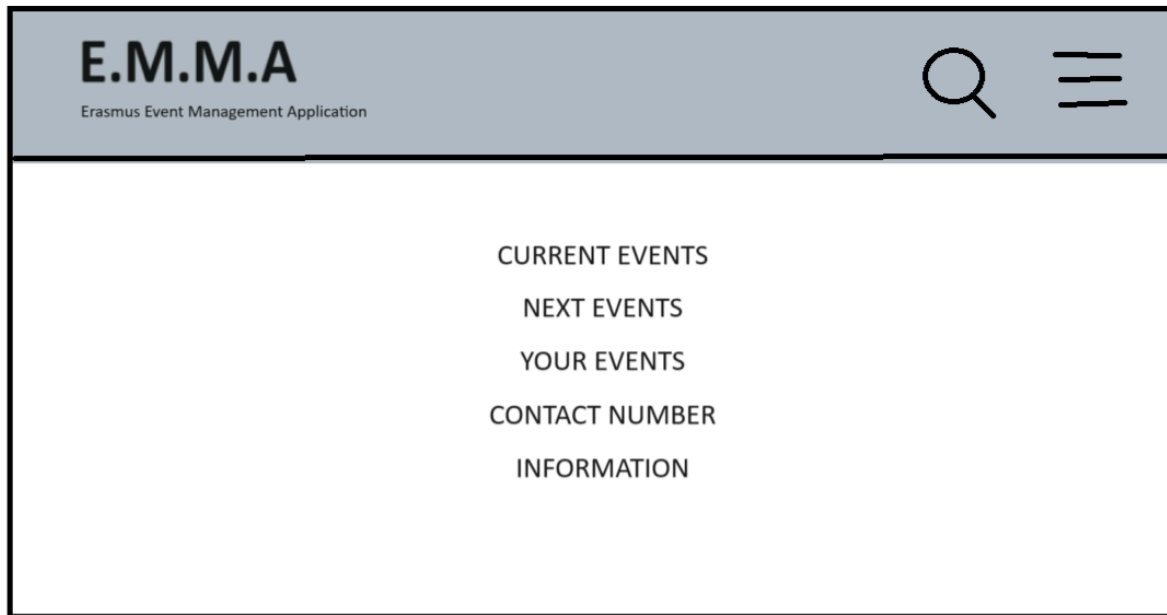


Image 2: Main screen example

	Input	Output
Student	Registration, secure payment options	Detailed event info, successful registrations, payment confirmations
Organizer	Input fields for event details (name, date, location, description, pricing) with submission, edit, approval tools.	Submitted events for approval, edited details, monitoring pending approvals.
Treasurer	Financial data, transactions, budget info, secure payment processing, transaction history tracking	Financial insights, processed transactions, budget updates
Insurance	Fee calculation based on event specifics, secure payment.	Insurance enrollment, fee confirmation.
Monitor	Mobile-friendly real-time event monitoring, attendance data, emergency reporting.	Real-time attendance, reported emergencies.
IT Admin	Backend access for database admin, system maintenance, helpdesk/support ticket system.	Managed databases, maintained systems, resolved issues.
Manager	Approval, scheduling, event analytics,	Approved events, scheduled

	statistics, financial data for planning and optimization.	timings, detailed analytics, financial insights.
--	---	--

Table 2.3.1 Input and output

2.4. Quality requirements, organizational needs, collaboration and the external factors - Julio del Junco Prieto

Quality Requirements:

- **Performance:** Optimize the app for fast loading times, particularly when accessing event listings and multimedia content.
- **Reliability:** Ensure the app is hosted on a reliable server with minimal downtime.
- **Usability:** Design an intuitive and user-friendly interface, considering the diverse user base of students, organizers, and monitors.
- **Security:** Implement secure authentication with a strong password, use HTTPS, and encrypt sensitive data, which are payment information and student's data. Make sure availability, disponibility and integrity are not compromised.
- **Scalability:** Design the app to handle increased user activity during peak times, such as when many students are registering for events.
- **Data Integrity:** A strong data integrity mechanism, using electronic certificates, is necessary to ensure accurate financial records, event details, and user information.

Organizational Needs:

- **Branding:** Ensure the app's design aligns with the branding guidelines of the ESN organization.
- **Content Management:** Implement an easy-to-use content management system for event organizers to update event details.
- **Analytics:** Integrate analytics tools to track user engagement, popular events, and other relevant metrics.
- **Integration with Other Systems:** Consider integration with other ESN tools or databases to streamline organizational processes.

Collaboration:

- **Collaboration Tools:** Use project management tools to coordinate tasks, track progress, and manage collaboration effectively.
- **Real-Time Updates:** Features like real-time updates and notifications ensure that all involved parties are informed promptly, such as event cancellations or important notifications from monitors.

External Factors:

- **Regulatory Compliance:** Ensure the app complies with Regulation 2016/679 and Directive 2016/680 from the European Union about data security and management.

- **User Demographics:** Consider the demographics of ESN members when designing and updating the app.

2.5 Functional requirements' detail analysis and Use Case

Diagrams - Marta Cuevas Rodríguez, Julio del Junco Prieto, Martín Díaz-Benito Álvarez

UC01 - IT Maintenance

Author: Marta Cuevas

Objective: To perform routine maintenance and ensure the proper functioning of the IT system.

Context: Ensure the IT system's reliability, security, and performance through regular maintenance.

Level: blue

Main Actor: IT Manager

Participants: None

Triggers: A problem with the system has been detected or maintenance is scheduled

Initial Condition: The IT system is operational

Minimal Guarantee: At the end of maintenance, the IT system is in a stable and improved state compared to before the maintenance, even if the maintenance encounters issues.

Main Scenario of Success:

1. IT Manager schedules routine maintenance or responds to an alert indicating a potential issue.
2. The system enters a maintenance mode, restricting user access temporarily.
3. IT Manager performs diagnostics to identify any existing issues or potential areas of improvement.
4. Necessary updates, patches, or fixes are applied to the system.
5. System configurations are reviewed and optimized for performance.
6. Security protocols are validated and strengthened.
7. Maintenance mode is deactivated, and the IT system is returned to normal operation.

Extensions: no extension

UC02 - Manage Student Database

Author: Marta Cuevas

Objective: To manage and maintain the student database, ensuring accurate and up-to-date student information.

Context: Maintain a reliable and comprehensive student database to support various functions within the ESN events management application.

Level: blue

Main Actor: IT Manager

Participants: None

Trigger: Regular database maintenance schedule or receipt of updated student information.

Initial Condition: The student database is accessible and contains existing student records.

Minimal Guarantee: At the end of the process, the student database is updated, and student information is accurate and consistent.

Main Scenario of Success:

1. Database Administrator initiates the process by scheduling routine maintenance or receives updated student information.
2. The system restricts write access to the database temporarily to prevent concurrent modifications.
3. Database Administrator reviews incoming student data for accuracy and completeness.
4. Outdated or incorrect student records are identified and flagged for correction.
5. New student records are added to the database.
6. Modified or corrected records are updated in the database.
7. The system's write access is restored, allowing normal operations to resume.

Extension: No extension

UC03 - Manage ESN Funds

Author: Marta Cuevas

Objective: To manage and oversee the funds allocated to the ESN events, ensuring proper distribution and financial transparency.

Context: Effectively control and distribute funds to support various ESN events and activities while maintaining financial accountability.

Level: blue

Main Actor: Treasurer

Participants: Sponsors, Managers

Trigger: New funds need to be budgeted.

Initial Condition: The ESN funds are available, and there are pending funding requests or scheduled distributions.

Minimal Guarantee: At the end of the process, the funds are appropriately distributed according to the financial plan.

Main Scenario of Success:

1. Treasurer initiates the process by reviewing funding requests received from event organizers or based on a scheduled distribution plan.
2. The system restricts write access to the funds temporarily to prevent concurrent modifications.
3. Treasurer reviews the funding requests, ensuring they align with the financial plan and event budgets.
4. Funds are allocated to approved events based on the treasurer's review.
5. The treasurer updates financial records to reflect the fund allocations.
6. A notification is sent to event organizers regarding the approved fund allocation.
7. The system's write access to funds is restored, allowing normal financial operations to resume.

Extension:

4.1 UC06-Record ESN event expenses. The treasurer makes sure that all the expenses are correctly updated in the system.

4.2. UC07-Record ESN event payments. The treasurer checks the payments and updates the system.

UC04 - Fund the ESN

Author: Marta Cuevas

Objective: To secure funding for the ESN organization through various means, including sponsorships, donations, and grants.

Context: Ensure the financial sustainability of ESN by actively seeking and securing funds to support its events and activities.

Level: blue

Main Actor: Sponsor

Participants:None

Trigger: Identification of a need for additional funds, initiation of fundraising campaigns, or pursuit of grant opportunities.

Initial Condition: ESN has identified a need for additional funds, or the fundraising manager proactively seeks opportunities for financial support.

Minimal Guarantee: At the end of the fundraising activities, the targeted funds are secured, and appropriate records are maintained.

Main Scenario of Success:

1. Fundraising Manager initiates the process by identifying the need for additional funds or actively seeking fundraising opportunities.
2. The fundraising strategy is formulated, including target amounts, sources (sponsors, donors, grants), and a timeline.
3. Sponsorship proposals are sent to potential sponsors, emphasizing the benefits of supporting ESN events and activities.
4. Donor engagement campaigns are launched, utilizing various channels such as social media, email, and events.
5. Grant applications are submitted to relevant granting organizations, highlighting the alignment of ESN goals with the grant criteria.
6. Funds are secured through successful sponsorships, donations, and grants.
7. Acknowledgment and appreciation messages are sent to sponsors, donors, and granting organizations.
8. The secured funds are appropriately recorded in the financial records of the ESN events management application.

UC05 - Offer Insurance

Author: Marta Cuevas

Objective: To provide insurance options to users, primarily focusing on offering medical insurance coverage for participants in ESN events.

Context: Enhance the safety and well-being of participants in ESN events by offering insurance coverage and support in case of medical emergencies.

Level: blue

Main Actor: Insurance Representative

Participants: Users (participants in ESN events)

Trigger: Identification of the need for insurance coverage for participants in ESN events or the initiation of insurance-related campaigns.

Initial Condition: ESN has identified the need for insurance coverage for event participants or has planned insurance-related initiatives.

Minimal Guarantee: At the end of the insurance offering process, participants have access to insurance coverage, and relevant records are maintained.

Main Scenario of Success:

1. Insurance Representative initiates the process by identifying the need for insurance coverage for participants in ESN events.
2. The insurance offering strategy is formulated, including coverage details, cost, and communication channels.
3. Information about the insurance coverage is communicated to participants through various channels such as the ESN events management application, website, and promotional materials.
4. Participants are provided with options to enroll in the insurance coverage during event registration or through a dedicated insurance enrollment process.
5. Participants who opt for insurance coverage provide necessary details, such as personal information and any required medical history.
6. Insurance premiums are collected from participants who choose to enroll in the coverage.
7. Acknowledgment and policy details are sent to participants who have successfully enrolled in the insurance coverage.
8. The system maintains records of participants with insurance coverage and relevant payment transactions.

Extension: No extension

UC06 - Record ESN Events Payments

Author: Marta Cuevas

Objective: To record and manage payments made by participants for ESN events, ensuring accurate financial tracking and transparency.

Context: Maintain a reliable and transparent financial record of payments for ESN events to support financial planning and reporting..

Level: blue

Main Actor: Treasurer

Participants: Users attending ESN events

Trigger: Completion of event registration or the initiation of payment processes for ESN events.

Initial Condition: Participants have registered for ESN events and are required to make payments as part of the event registration process.

Minimal Guarantee: At the end of the payment recording process, accurate financial records are maintained, and participants receive acknowledgment of their payments.

Main Scenario of Success:

1. Treasurer initiates the payment recording process upon completion of event registration or the initiation of payment processes.
2. The system generates invoices or payment details for participants based on the event registration information.
3. Participants receive notifications or access to their invoices, detailing the amount due, payment methods, and deadlines.
4. Participants make payments through specified payment channels (e.g., credit card, bank transfer) within the designated time frame.
5. The system records payment transactions, updating the financial records of the ESN events management application.
6. Acknowledgment messages are sent to participants upon successful payment, providing details of the transaction.
7. The system marks participants as fully registered or paid for the respective events.
8. Financial reports are generated, reflecting the updated payment status for each event.

Extension: No extension

UC-07 Record ESN Events Expenses

Author: Marta Cuevas

Objective of the Use Case: To record and manage expenses incurred during ESN events, ensuring accurate financial tracking, transparency, and budget management.

Context: Maintain a comprehensive and transparent financial record of expenses for ESN events to support effective financial planning, reporting, and analysis.

Level: blue

Main Actor: Treasurer

Participants: Event Organizers, Suppliers

Trigger: Completion of ESN events or the initiation of the expense recording process.

Initial Condition: ESN events have taken place, and organizers or suppliers have submitted expense details for reimbursement.

Minimal Guarantee: At the end of the expense recording process, accurate financial records are maintained, and relevant stakeholders are informed of the financial status.

Main Scenario of Success:

1. Treasurer initiates the expense recording process upon completion of ESN events or the initiation of the expense reporting period.
2. Event Organizers and Suppliers submit detailed expense reports, including receipts, invoices, and a breakdown of incurred expenses.
3. The system validates expense reports for completeness and adherence to ESN financial policies.
4. Validated expense reports are approved by the Treasurer, allowing for reimbursement or processing of payments to suppliers.
5. The system updates the financial records, reflecting the recorded expenses for each event.
6. Acknowledgment messages are sent to Event Organizers and Suppliers upon the successful processing of their expense reports.
7. Financial reports are generated, reflecting the updated expense status for each event.

Extension: No extension

UC08 - Viewing List of Upcoming Events

Author: Julio del Junco

Objective: To allow managers and monitors to access a list of upcoming events, view event details, and participant information for overseeing and coordinating event activities.

Context : The aim is to provide users with a comprehensive overview of scheduled events, enabling efficient coordination and oversight.

Level: blue

Main Actor: Manager and Monitor

Participants: Database system for event information retrieval.

Trigger: User requests access to the list of upcoming events and event details.

Initial Condition: The user is logged into the system and has access permissions to view event-related information.

Minimal Guarantee: Even if the retrieval of event information fails, the system ensures that the user's access and privacy are protected, providing appropriate error messages or alternative actions.

Main Scenario of Success:

1. User navigates to the "Upcoming Events" section within the application.
2. System retrieves and displays a list of scheduled events.
3. User selects a specific event from the list.
4. System presents detailed information about the selected event, including date, time, location, participants, and event description.
5. User navigates back to the event list or continues to review other events.

Extensions: No extensions

UC09 - Accessing List of Students for an Event

Author: Julio del Junco

Objective: To facilitate monitors' access to a list of students registered for an event, enabling them to check attendees and ensure the event's smooth operation and safety.

Context: This Use Case addresses the need for event monitors to have efficient access to participant information for event oversight and safety management.

Level: blue

Main Actor: Monitor

Participants: Database system for student registration data retrieval.

Trigger: Monitor selects the event they wish to oversee and requests access to the list of registered students for that event.

Initial Condition: The monitor is logged into the system with appropriate access permissions to view event-related information.

Minimal Guarantee: Even if there's a failure in retrieving student registration details, the system ensures that the monitor's access and privacy are protected, providing appropriate error messages or alternative actions.

Main Scenario of Success:

1. Monitor navigates to the "Event Monitoring" section within the application.
2. Monitor selects the specific event they are overseeing.

3. System retrieves and displays a list of students who have registered for the selected event.
4. Monitor reviews the list, checking attendees against the expected participants and ensuring event safety and smooth operation.
5. Monitor may further inquire about specific participant details or contact information if needed.

Extensions:

3.1: If the selected event has no registered students or no data is available, the system provides an appropriate message indicating the lack of registrations.

UC10 - Reporting Emergencies during Events

Author Julio del Junco

Objective: To provide monitors with the capability to report emergencies through the app during events, alert managers, and take necessary actions to ensure the safety and well-being of attendees, thereby improving safety for future events.

Context: This Use Case addresses the need for a structured process to report emergencies during events, facilitating prompt actions for attendee safety and future event improvements.

Level: blue

Main Actor: Monitor

Participants: Manager

Trigger: Monitor identifies an emergency situation during an event and requires immediate action.

Initial Condition: The monitor is logged into the system with appropriate access permissions to report emergencies.

Minimal Guarantee: Even if the emergency reporting fails, the system ensures that the monitor's attempt to alert the relevant authorities is recorded, and alternative communication channels or emergency protocols are suggested.

Main Scenario of Success:

1. Monitor identifies an emergency situation during the event that requires immediate attention.
2. Monitor accesses the app's emergency reporting feature.
3. Monitor provides details of the emergency, such as type, location, and severity, through a dedicated form or interface.
4. System triggers immediate notifications to managers and relevant personnel designated for emergency response.

5. Managers receive the alert, acknowledge it, and initiate necessary actions to address the reported emergency.

Extensions: No extension

UC11 - Approving or Rejecting Event Proposals

Author: Julio del Junco

Objective: To empower managers to review and make decisions on event proposals submitted by organizers, ensuring compliance with safety and quality standards set by the application.

Context: This Use Case addresses the need for a structured process for managers to evaluate event proposals, maintaining safety and quality across organized events.

Level: blue

Main Actor: Manager

Participants: Organizers, Event Staff, Notification System

Trigger: A new event proposal is submitted by an organizer and awaits manager approval (UC-22).

Initial Condition: The manager is logged into the system with appropriate access permissions to review and approve event proposals.

Minimal Guarantee: Even if the approval or rejection process fails, the system ensures that the proposal's status is recorded, and appropriate follow-up actions or notifications are initiated.

Main Scenario of Success:

1. Manager receives a notification or accesses the dashboard indicating a new event proposal.
2. Manager navigates to the section for reviewing event proposals within the application.
3. Manager accesses the details of the submitted event proposal, including event description, logistics, safety measures, and any other relevant information.
4. Manager evaluates the proposal against predefined safety and quality standards or criteria.
5. The Manager rejects, the system informs the organizer with reasons for rejection, providing an opportunity for revisions or additional information.

Extensions:

5.1: The Manager approves the proposal, the system updates the event status, notifying the organizer and showing the event on the app.

UC-12- Browsing Events

Author: Martín Díaz-Benito

Objective: Enable users to explore and view details of available events within the application.

Context: Enhancing user engagement by providing a convenient platform for discovering and learning about events.

Level: Blue

Main Actor: Student

Trigger: The student selects the "Browse Events" feature.

Initial Condition: The student is logged into the application, and the application displays the default or last selected event category.

Minimal guarantee: The events should be able to be browsed by all users even if an error occurs.

Main Scenario of Success:

1. The system displays a list of available events, categorized by type or date.
2. The student views event details, including name, date, location, and description.
3. The student can apply filters to narrow down the list of events based on preferences.
4. The student can click on an event to sign up.
5. The student can navigate back to the main menu.

Extensions: No extensions

UC-13 - View Statistics

Author: Martin Diaz-Benito

Objective: To provide event organizers with access to statistics and analytics related to their events within the application.

Context: This Use Case addresses the need for organizers to have insights and data analysis capabilities to understand event performance, attendee engagement, and other relevant metrics.

Level: Blue

Main Actor: Event Organizer

Participants: Database System, Analytics Module

Trigger: Event Organizer accesses the statistics section within the application.

Initial Condition: The event organizer is logged into the system and navigates to the section for viewing event statistics.

Minimal Guarantee: Even if the process of accessing statistics fails, the system ensures that the event data is preserved, and appropriate follow-up actions or notifications are initiated.

Main Scenario of Success:

1. Event Organizer accesses the statistics section within the application.
2. System presents a dashboard or interface displaying various statistics related to the organizer's events.

Extensions:

None

UC-14 - Event Management

Author: Martín Díaz-Benito

Objective of the Use Case: To provide event organizers with comprehensive tools and functionalities to manage events within the application.

Context: This Use Case addresses the need for organizers to have a centralized system for managing events, including creating, canceling, and modifying event details.

Level: Blue

Main Actor: Event Organizer

Participants: Database System, Notification System

Trigger: Event Organizer accesses the event management section within the application.

Initial Condition: The event organizer is logged into the system and navigates to the section for event management.

Minimal Guarantee: Even if the event management process fails, the system ensures that the event data is preserved, and appropriate follow-up actions or notifications are initiated.

Main Scenario of Success:

1. Event Organizer accesses the event management section within the application.
2. System presents a dashboard or interface displaying existing events organized by the user.
3. Organizer selects a specific event to view, manage, or make changes.
4. Organizer manages the event details, such as:
 - Viewing event information (date, time, location, description)
 - Managing participant lists
 - Handling event logistics and resources

Extensions:

4.1 If the organizer intends to create a new event, event triggers Create Event (UC-22).

4.2 If the organizer decides to cancel an event, event triggers Cancel Event (UC-17).

4.3 If the organizer wishes to modify event details, event triggers Modify Event (UC-18).

UC-15 - Uploading Media

Author: Martín Díaz-Benito

Objective: To allow event organizers to upload media content related to events (e.g., photos, videos) into the system for students to access and view.

Context: This Use Case addresses the need for organizers to share event-related media content with students, enhancing their event experience and engagement within the application.

Level: Blue

Main Actor: Event Organizer

Participants: Media Upload Module, Database System

Trigger: Event Organizer accesses the media upload section within the application after an event.

Initial Condition: The event organizer is logged into the system and navigates to the section for uploading media.

Minimal Guarantee: Even if the media upload process fails, the system ensures that the uploaded media content is preserved, and appropriate follow-up actions or notifications are initiated.

Main Scenario of Success:

1. Event Organizer accesses the media upload section within the application.
2. Organizer selects the event for which they want to upload media content.
3. Organizer selects the media files (e.g., photos, videos) from their local storage or device.
4. System initiates the upload process, transferring the selected media files to the system's database.
5. System confirms the successful upload of media content related to the specified event.

Extensions: No extensions

UC-16 - Join an Event

Author: Martín Díaz-Benito

Objective: To enable users to successfully join free events, manage their participation status, leave events if needed, and handle payment processes for paid events.

Context: This Use Case addresses the need for users to seamlessly join free events and manage their participation status, allowing them to leave events or handle payment procedures for paid events as extensions.

Level: Blue

Main Actor: User (Students, Attendees)

Participants: Event Staff, Payment Gateway System (if applicable)

Trigger: User accesses the event management section within the application.

Initial Condition: The user is logged into the system and wants to join or manage their participation in an event.

Minimal Guarantee: Even if the event management process fails, the system ensures that the user's participation status or payment information is preserved, and appropriate follow-up actions or notifications are initiated.

Main Scenario of Success:

1. User browses available events and selects a free event they wish to join.
2. User clicks the "Join" button.
3. User fills up a form and clicks a button, confirming their participation in the event.
4. System updates the user's participation status, adding them to the event attendee list.

Extensions:

- 4.1: User Handles Payment for a Paid Event: System triggers payment for the event (UC-20)
4.2: User Decides To Leave the Event: System triggers leave event (UC-19)

UC-17 - Cancel Event

Author: Martín Díaz-Benito

Objective of the Use Case: To provide event organizers with a systematic process to cancel an event within the application.

Context: This Use Case addresses the need for organizers to have the capability to cancel events, updating the event status and notifying relevant stakeholders.

Level: Blue

Main Actor: Event Organizer

Participants: Database System, Notification System

Trigger: Event Organizer decides to cancel a specific event.

Initial Condition: The event organizer is logged into the system and navigates to the section for canceling events.

Minimal Guarantee: Even if the event cancellation process fails, the system ensures that relevant event data is preserved, and appropriate follow-up actions or notifications are initiated.

Main Scenario of Success:

1. Event Organizer accesses the event cancellation section within the application.
2. System presents a list of existing events organized by the user.
3. Organizer selects the specific event to be canceled.
4. System prompts the organizer for confirmation.
5. System updates the event status to "canceled" and records the cancellation details.
6. Upon successful cancellation, the system triggers notifications to relevant stakeholders, such as notifying attendees about the event cancellation and updating the event listing to reflect the new status.

Extensions:

3.1: If there's a delay in retrieving event details or processing the cancellation, the system provides notifications or a progress indicator to inform the organizer about the process status.

UC-18 - Modify Event

Author: Martín Díaz-Benito

Objective of the Use Case: To enable event organizers to make changes or modifications to event details within the application.

Context: This Use Case addresses the need for organizers to have the capability to update or modify event information, ensuring accuracy and relevance.

Level: Blue

Main Actor: Event Organizer

Participants: Database System, Notification System

Trigger: Event Organizer decides to modify specific details of an existing event.

Initial Condition: The event organizer is logged into the system and navigates to the section for modifying events.

Minimal Guarantee: Even if the event modification process fails, the system ensures that relevant event data is preserved, and appropriate follow-up actions or notifications are initiated.

Main Scenario of Success:

1. Event Organizer accesses the event modification section within the application.
2. System presents a list of existing events organized by the user.
3. Organizer selects the specific event they want to modify.
4. Organizer updates or modifies the event details
5. Organizer finalizes the modifications and submits the updated event details.
6. System processes the submitted modifications and updates the event details in the database.

Extensions:

6.1: Upon successful modification, the system triggers notifications to relevant stakeholders, such as notifying attendees about changes and updating the event listing to reflect the updated details.

UC-19 - Leaving an Event

Author: Martín Díaz-Benito

Objective: To allow users to leave an event they've previously registered for, removing themselves from the attendee list and updating their participation status.

Context: This Use Case addresses the need for users to manage their participation status in events by providing a structured process to leave events they no longer wish to attend.

Level: Blue

Main Actor: User (Students, Attendees)

Participants: Event Staff, Notification System

Trigger: User decides to leave an event they've previously registered for.

Initial Condition: The user is logged into the system and has registered for one or more events.

Minimal Guarantee: Even if the process of leaving an event fails, the system ensures that the user's participation status is preserved, and appropriate follow-up actions or notifications are initiated.

Main Scenario of Success:

1. User accesses the event management section within the application.
2. User selects the event they want to leave.
3. User clicks the "Leave" button or initiates the process to disassociate themselves from the event.
4. System updates the user's participation status, removing them from the event attendee list.

Extensions:

3.1: If the leave process encounters errors or confirmation steps, the system guides the user through the necessary actions to successfully leave the event.

UC-20 - Paying for an Event

Author: Martin Diaz-Benito

Objective: To enable users to make payments for events that require a fee, facilitating their participation in paid events.

Context: This Use Case addresses the need for users to complete payment processes seamlessly within the application for events that have associated fees.

Level: Blue

Main Actor: User (Students, Attendees)

Participants: Payment Gateway System, Event Organizers, Notification System

Trigger: User selects a paid event they wish to attend and proceeds to complete the payment process.

Initial Condition: The user is logged into the system and has selected a paid event for which they intend to make a payment.

Minimal Guarantee: Even if the payment process fails, the system ensures that the user's payment information is protected, and appropriate follow-up actions or notifications are initiated.

Main Scenario of Success:

1. User accesses the event details section within the application.
2. User selects a paid event they want to attend and proceeds to the payment section.
3. User enters the necessary payment details (e.g., credit card information, payment method).
4. System processes the payment through the integrated Payment Gateway System.
5. Payment Gateway verifies the transaction and confirms the successful payment.
6. System updates the user's participation status, adding them to the event attendee list.

Extensions:

3.1: If the payment process encounters errors or validation issues, the system prompts the user to re-enter the payment details or choose an alternative payment method.

UC-21 - Sending Confirmation Email for Event Registration

Author: Martín Díaz-Benito

Objective: To automatically send a confirmation email to users after they successfully register for an event, acknowledging their participation and providing essential event details.

Context: This Use Case addresses the need to provide users with immediate confirmation and event details after they've successfully registered for an event within the application.

Level: Blue

Main Actor: System (Automated Process)

Participants: User (Attendee), Event Organizers

Trigger: User successfully registers for an event through the application.

Initial Condition: The user completes the registration process for an event within the application.

Minimal Guarantee: Even if the confirmation email sending process fails, the system ensures that the user's registration status is preserved, and appropriate follow-up actions or notifications are initiated.

Main Scenario of Success:

1. User successfully registers for an event through the application.
2. System triggers an automated process to generate a confirmation email for the registered user.
3. System sends the confirmation email to the registered user's provided email address.

Extensions: No extensions

UC-22 - Create Event

Author: Martín Díaz-Benito

Objective of the Use Case: To provide event organizers with a streamlined process to create and add new events proposal to the system within the application.

Context: This Use Case addresses the need for organizers to create and manage new events efficiently within the application.

Level: Blue

Main Actor: Event Organizer

Participants: Database System, Notification System

Trigger: Event Organizer decides to create a new event.

Initial Condition: The event organizer is logged into the system and navigates to the section for creating events.

Minimal Guarantee: Even if the event creation process fails, the system ensures that the event data is preserved, and appropriate follow-up actions or notifications are initiated.

Main Scenario of Success:

1. Event Organizer accesses the event creation section within the application.
2. System presents a form or interface prompting the organizer to input event details.

3. Organizer fills in the required event details.
4. Organizer submits the event creation form.
5. System processes the submitted information and adds the new event to the system's database.
6. The system ensures that the newly created event proposal is visible for the manager.

Extensions: This event triggers UC-11 (approving or rejecting event proposals).

UC-23: End Event

Author: Martín Díaz-Benito

Objective of the Use Case: The objective is to provide a mechanism for ending an event when a monitor presses an end event button.

Context : The business goal is to streamline the process of concluding events.

Level: Blue

Main Actor: Subsystem 2

Participants (other Actors): Event System

Trigger: The trigger is the Monitor pressing the "End Event" button in the event management system.

Initial condition: An active event is in progress, and the event management system is operational.

Minimal guarantee: The system ensures that the event is properly concluded.

Main scenario of the success:

1. Monitor logs into the event management system.
2. Monitor selects the ongoing event.
3. Monitor navigates to the event details page.
4. Monitor locates and clicks the "End Event" button.
5. The system confirms the decision with a prompt.
6. Monitor confirms the end of the event.

Extensions: No extensions

UC-24: Generate Statistics

Author: Martin Diaz-Benito

Objective of the Use Case: The objective is to automatically generate detailed statistics for a concluded event.

Context: The business goal is to provide event organizers and stakeholders with comprehensive statistics to analyze the success and impact of events.

Level: Blue.

Main Actor: Monitor

Participants (other Actors): Event Data Repository, Analytics Database

Trigger (Initiating event – what starts the Use Case): The event is concluded successfully.

Initial condition (what state of the world we expect at the beginning): An event has been successfully concluded, and relevant data is available in the Event Data Repository.

Minimal guarantee: The system ensures that basic statistics are generated, even if some unexpected error occurs during the process.

Main scenario of the success:

1. The Statistics Generator retrieves event data from the Event Data Repository for the concluded event.
2. The Statistics Generator processes the data to calculate various statistics such as attendance, engagement metrics, and other relevant performance indicators.
3. The system stores the generated statistics in the event analytics database.
4. The Reporting Module is notified of the newly generated statistics.

Extensions: No extensions

UC-25 - Add Proposal to Events List

Author: Julio del Junco

Objective of the Use Case: To automatically add proposed events to the events list after they've been successfully admitted.

Context: This Use Case addresses the need for a streamlined process to incorporate approved event proposals into the official events list, ensuring visibility and access for users.

Main Actor: System

Participants: Database System

Trigger: Successful approval of an event proposal through UC-004.

Initial Condition: An event proposal has been submitted, reviewed, and approved through the proposal submission process.

Minimal Guarantee: Even if the process of adding the proposal to the events list fails, the system ensures that the approved event data is preserved, and appropriate follow-up actions or notifications are initiated.

Main Scenario of Success:

1. Event proposal undergoes the approval process (UC-004) and is successfully approved.
2. System automatically updates the events list by adding the approved event to the official events list.
3. Updated events list now includes the newly approved event, making it visible and accessible to users within the application.

Extensions:

No extensions

UC-26 - Login

Author: Martín Díaz-Benito Álvarez

Level: Blue

Objective: To log in to the system

Context: Authorization of the user for storing and processing their data

Main Actor: User (not logged in)

Initial Condition: The user is not logged in

Trigger: The user initiates the login process

Main Success Scenario:

1. The system presents the user with a login page containing fields for entering the username and password.
2. The user provides their login credentials.
3. The system successfully validates the user-provided data.
4. The user is granted access, and the system logs them in.

Extension:

3.1 – The system, unable to validate the user's credentials, suggests UC-28 for the password reminder procedure. The user provides login and password for the reminder process. (point 2).

UC-27 - User Registration

Author: Martín Díaz-Benito

Level: Blue

Objective: To register a new user in the system

Context: User wishes to create a new account to access the system's features and services

Main Actor: User (not registered)

Initial Condition: The user is not registered in the system

Trigger: User initiates the registration process

Main Success Scenario:

1. The system presents the user with a registration page containing fields for essential information, such as username, password, email, etc.
2. The user provides the required registration details.
3. The system validates the entered information.
4. Upon successful validation, the system creates a new user account.
5. The user receives a confirmation of successful registration.

Extensions:

No extension

UC-28 - Password Reminder

Author: Martin Diaz-Benito

Level: Blue

Objective: To assist users in recovering or resetting their forgotten password

Context: User has forgotten their password and needs assistance in recovering or resetting it

Main Actor: User (logged out due to forgotten password)

Initial Condition: The user is logged out and unable to access the system due to a forgotten password

Trigger: User initiates the password reminder process

Main Success Scenario:

1. The system presents the user with a password reminder page containing fields to enter the registered username or email associated with the account.
2. The user provides the system with the required information (username or email).
3. The system validates the entered information against the registered user data.
4. Upon successful validation, the system generates and sends a password reset link or code to the user's registered email address.
5. The user receives the password reset information and follows the provided instructions to set a new password.
6. The system confirms the successful password reset, allowing the user to log in with the new credentials.

Extensions:

No extension

UC-29 - Log out

Author: Julio del Junco Prieto

Level: Blue

Objective: To log out the system

Context: Authorization of the user for storing and processing their data

Main Actor: User (logged in)

Initial Condition: The user is logged in

Trigger: The user clicks the logout button

Main Success Scenario:

1. The user clicks the button
2. The data changed during the log is saved
3. The system resets the user to not logged user
4. The system goes to the login screen (UC-26)

Extensions:

No extension

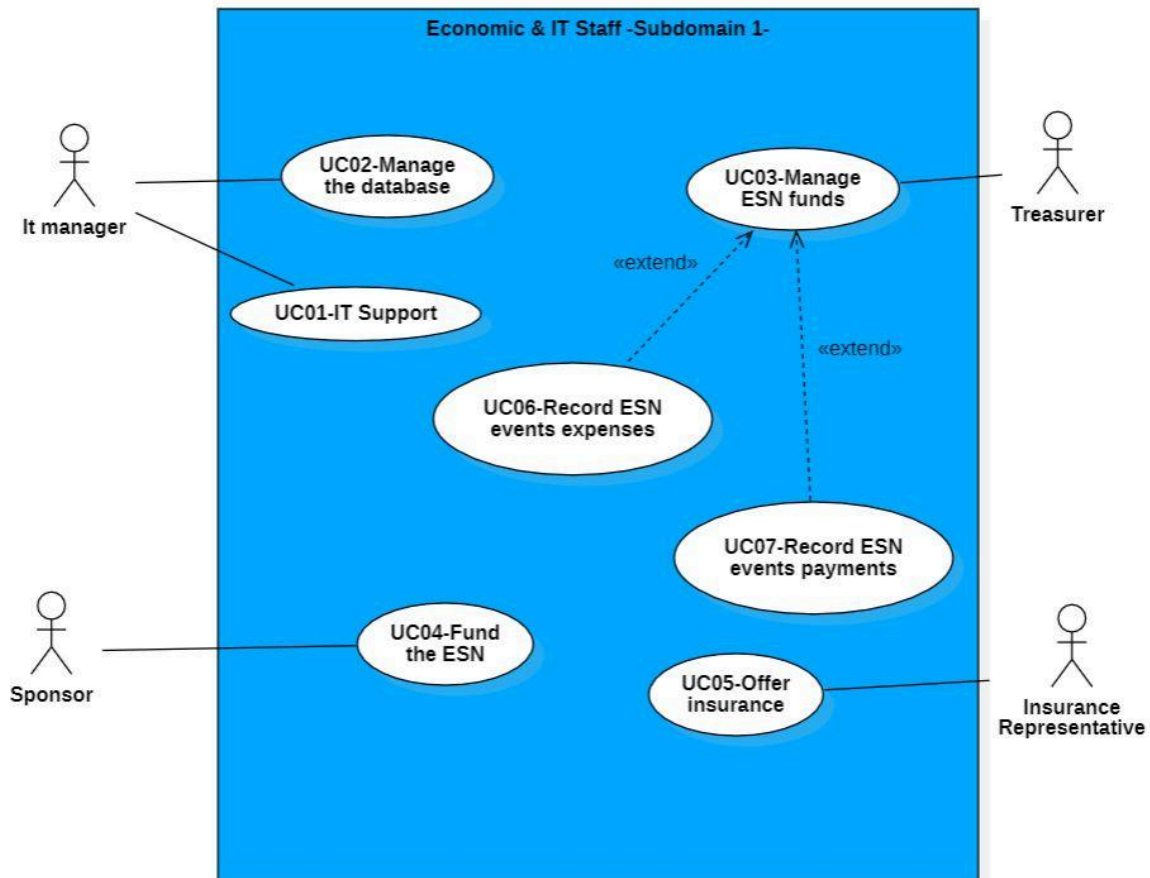


Figure 3: Use Case diagram for Economic & IT Staff. Marta Cuevas

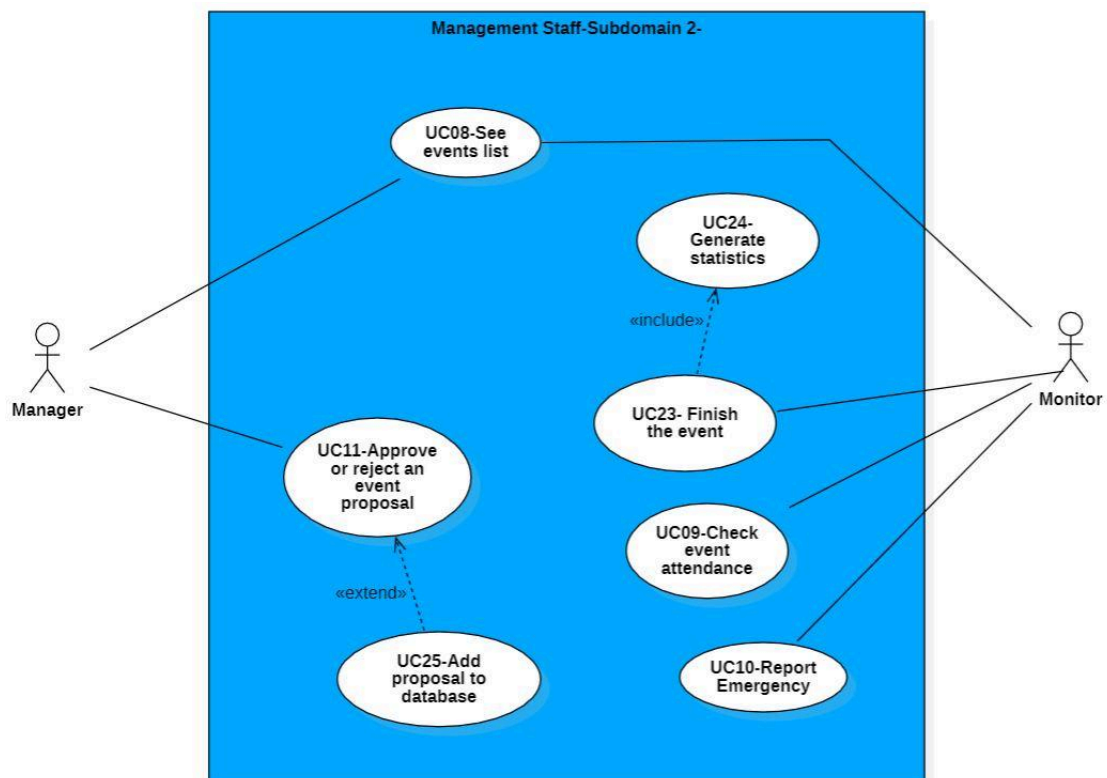


Figure 4: Use Case diagram for Management Staff. Julio del Junco

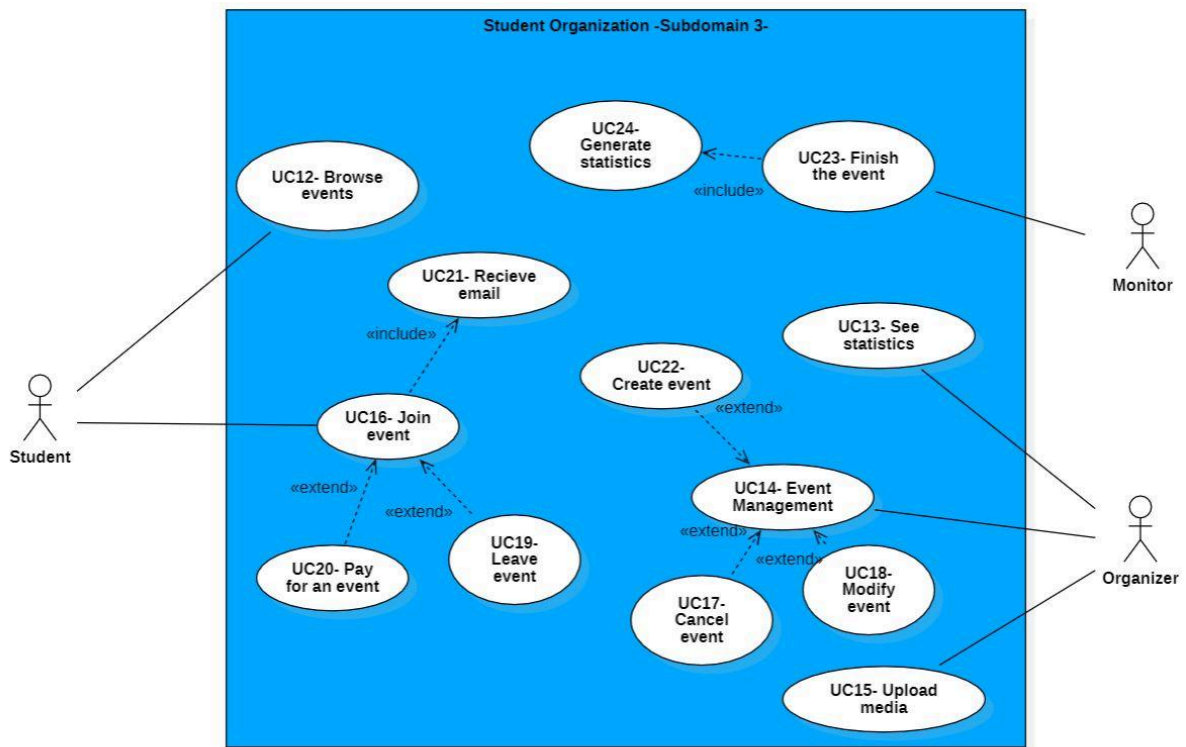


Figure 5: Use Case diagram for Student Organization. Martín Diaz-Benito

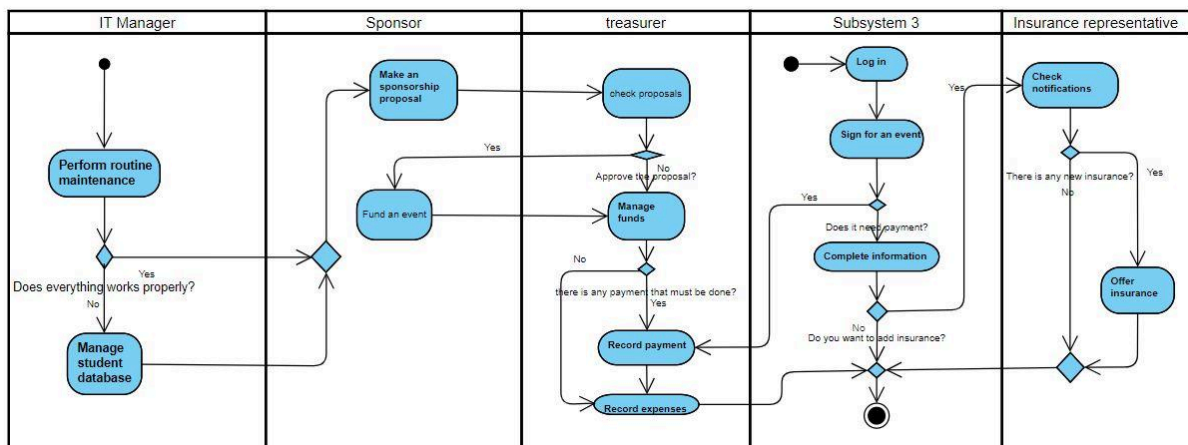


Figure 6: Activity diagram for Economic & IT Staff. Marta Cuevas

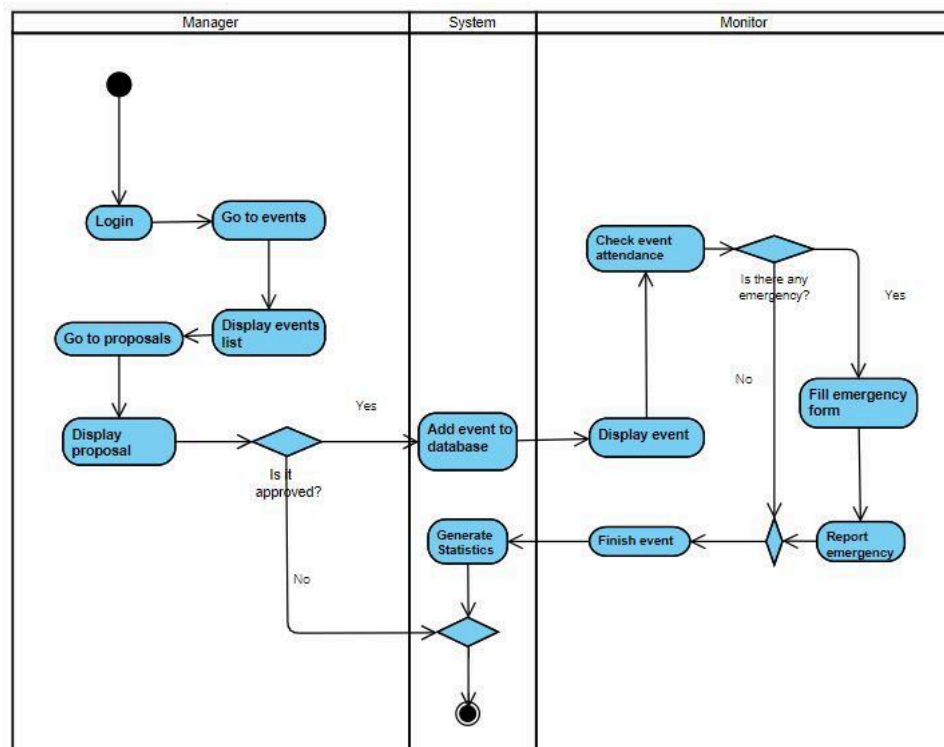


Figure 7: Activity diagram for Management Staff. Julio del Junco

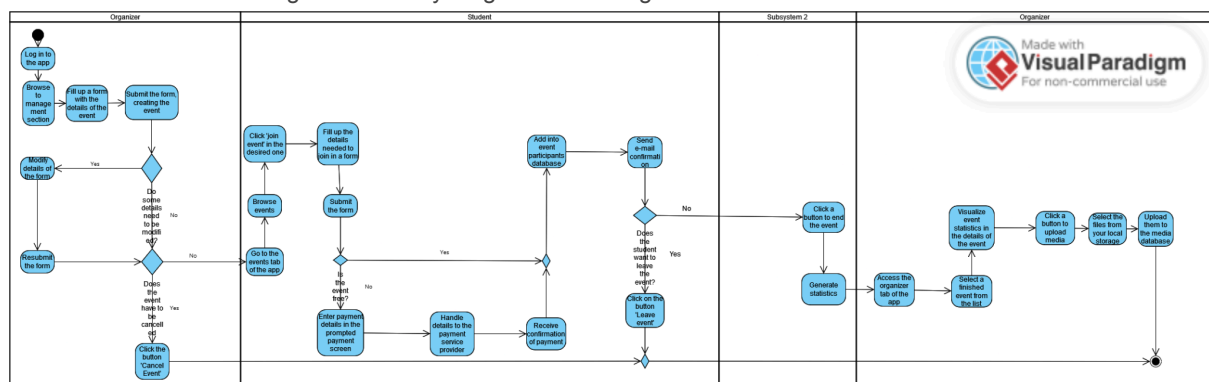


Figure 8: Activity diagram for Student Organization. Martín Diaz-Benito

2.6 Functional requirements for additional system functions - Julio del Junco Prieto

Administrative Functions:

- **Access Control:** Implement role-based access control to restrict system functionalities based on user roles (e.g., admin, organizer, participant).
- **Event Approval:** Allow managers to review and approve/disapprove event proposals submitted by organizers. Approved events are then made visible to participants.
- **Financial Reporting:** Provide administrators and treasurers with tools to generate financial reports, including income, expenses, and profit/loss statements.

- **Database Backup and Recovery:** Implement regular automated backups of the system database to ensure data integrity. Provide a mechanism for administrators to restore data if needed.

Common Functions:

- **Notification System:** Implement a notification system to inform users about important events, such as system updates.
- **Audit Trail:** Record and maintain an audit trail of user activities within the system for security and accountability purposes.
- **Search and Filter:** Provide users with a robust search and filtering mechanism to easily find events, users, or financial transactions based on various criteria.

Internal Functions:

- **System Monitoring:** Implement tools for the IT manager to monitor system performance, identify bottlenecks, and ensure optimal operation.
- **Error Handling and Logging:** Include a comprehensive error handling system that logs errors for debugging purposes and presents user-friendly error messages when needed.
- **Security Measures:** Implement security measures such as encryption, secure communication protocols, and regular security audits to safeguard user data and system integrity.
- **Emergency Response System:** Description: Enable the system to trigger emergency responses in case of critical issues, such as notifying managers and taking predefined actions.

2.7 Non-functional requirements - a systematic approach - Julio del Junco Prieto

Product:

- **RNF01-Do a clear design:** Design an user-friendly interface, considering users are mostly young people.
- **RNF02-Use the brand's motif:** The app must use the patrons and motifs of the ESN.
- **RNF03-Create statistics:** Integrate analytics to track user engagement and popular events

Organisational:

- **RNF04-Updates and maintenance:** The app must be updated and has to have maintenance
- **RNF05-Scalability:** The app must support an increased amount of people and be designed for it.
- **RNF06-Reliability:** The app will be hosted in a server that supports all the events with all the students without failing

- **RNF07-Performance:** The app has to be able to do all the necessary processes easily and quickly.

External:

- **RNF08- Law reliability:** The app must comply with Regulation 2016/679 and Directive 2016/680 from the European Union about data security and management.
- **RNF09-Security measures:** Make sure the app is secure from any type of attack and the data is well stored.
- **RNF10-Integration with payment systems:** Consider integration with payment systems in order to pay for the events.

3. Analysis of data structure - Martín Diaz-Benito Álvarez

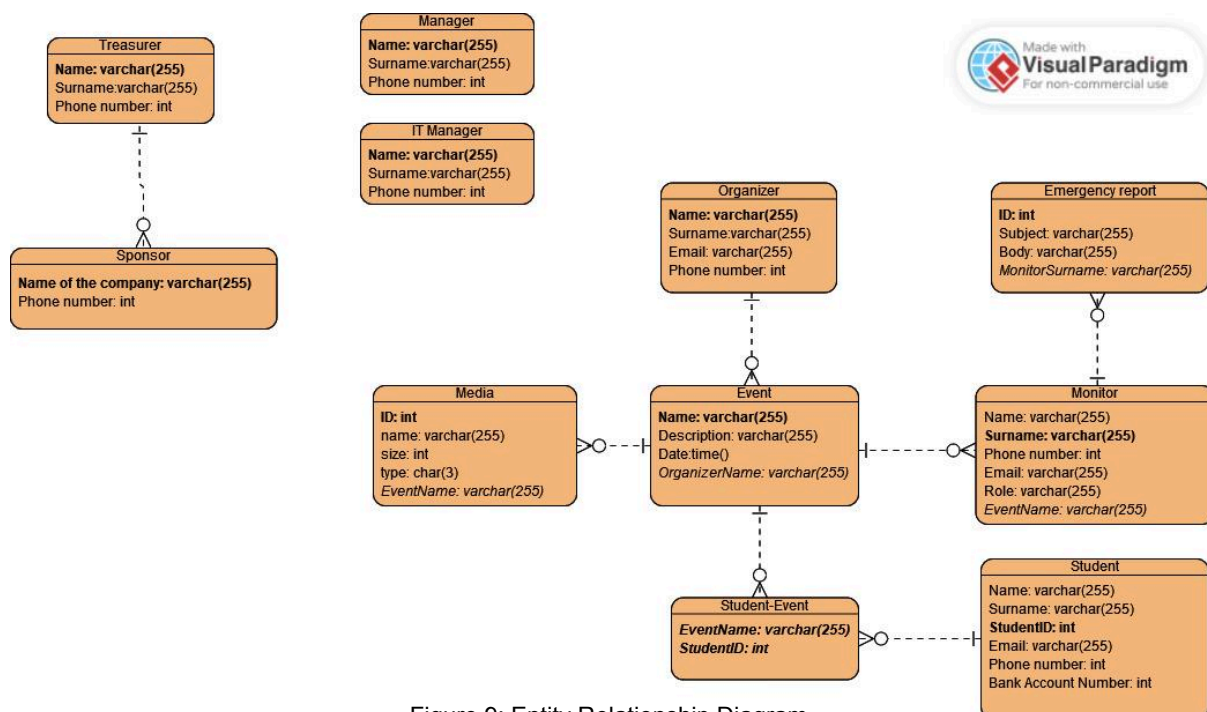


Figure 9: Entity Relationship Diagram.

There is a one to many relationship between sponsor and treasurer, as several sponsors can fund the ESN.

There is a one to many relationship between Emergency reports and Monitor (instructor) as a monitor can report many times.

There is a one to many relationship between Event and Monitor, as one event can have several monitors but a monitor can only attend one event at a time.

There is a many to many relationship between Student and Event, because a student can attend different events and an event has a list of attendants (students).

There is a one to many relationship between Organizer and Event, as an organizer can organize several events.

There is a one to many relationship between Media and Event, as one event can have more than one picture taken from it.

Manager and IT Manager are unattached as they oversee all events in general (manager) and IT infrastructure.

4. Model of system's behavior in time- Julio del Junco Prieto

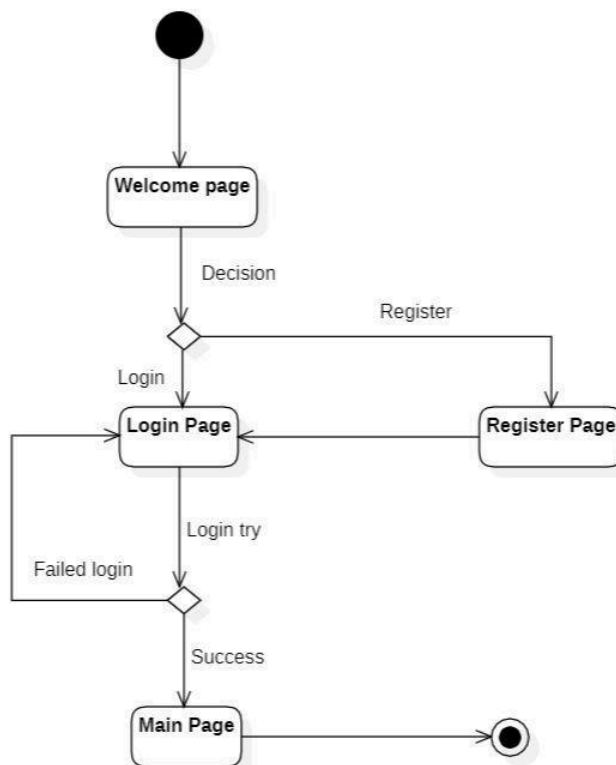


Figure 10: State Machine Diagram for Login and Register

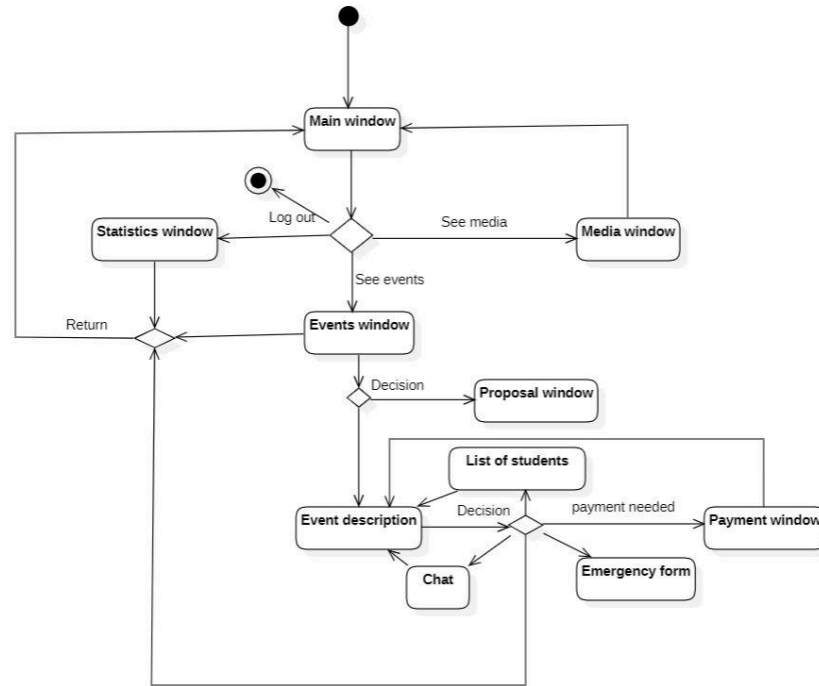


Figure 11: State Machine Diagram for App functioning

5. Comparison of models - Martín Diaz-Benito Álvarez

We checked all the content of this project, including all the diagrams, and we realized we didn't put the login, logout and the reset password as use cases, so we added them.

Subdomain 1 - Economic and IT Staff

Firstly, to check this subdomain, we will look at the organizational schema, and check that all the people represented there have some kind of duty within it, represented in their user stories. Then, we will make sure that all these user stories are being reflected in the use cases and that the diagrams are coherent with the use cases. For this subdomain, all user stories, activity and use case diagrams made sense within them, making sure it would be able to work as a single subsystem.

Subdomain 2 - Management staff

In this case we will follow a similar procedure than for subdomain 1, taking a look at everything and making sure everything is okay. For the user stories, they seem correct, and there are a bunch of them, and all functionalities are mentioned in the use cases. After several corrections in previous versions of this document, the use case diagrams and activity diagrams are also correct. After reading the goals for this subsystem, we can definitely conclude that it complies with what we need for this project.

Subdomain 3 - Student Organization

We will start the check by the user stories as usual. In this user stories, we have found one that talks about promotion for the events within the app that we didn't really take into account, so we decided to remove it because otherwise we would have to change everything. This is the only one mistake, as the rest of the use case, diagrams and extra requirements are well described.

6. System's Architecture- Julio del Junco Prieto

The system consists of a mobile application based on ESN events that can be accessed by students (who are the main users), organizer, manager and monitors. It also can be used as a web page.

The application is built on an ESN ust server and takes use of a database to store both users information and complement things like permissions.

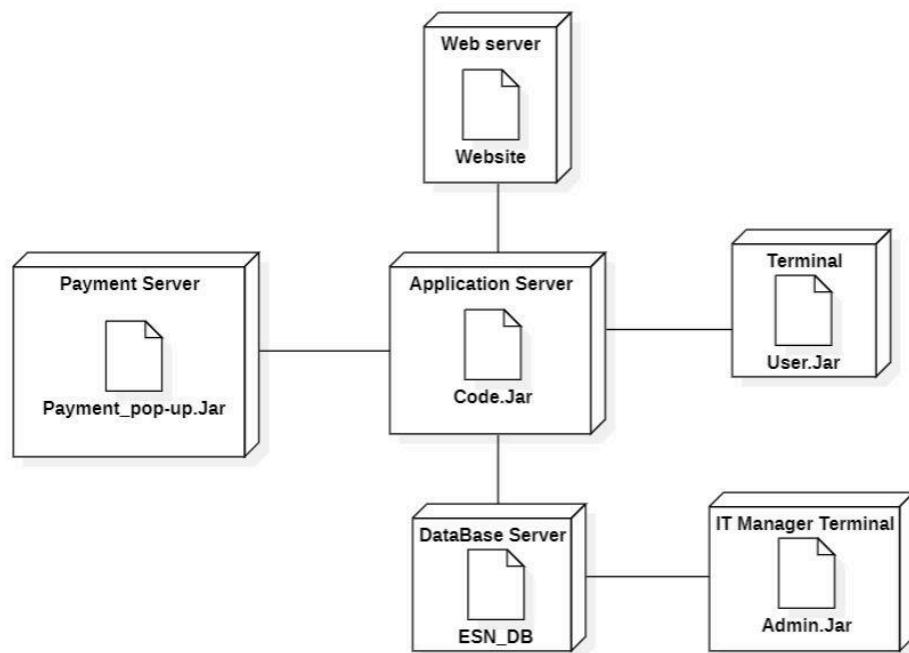


Figure 12: Deployment Diagram of System.

7. Project of the interface -Julio del Junco Prieto, Martín Díaz-Benito Álvarez

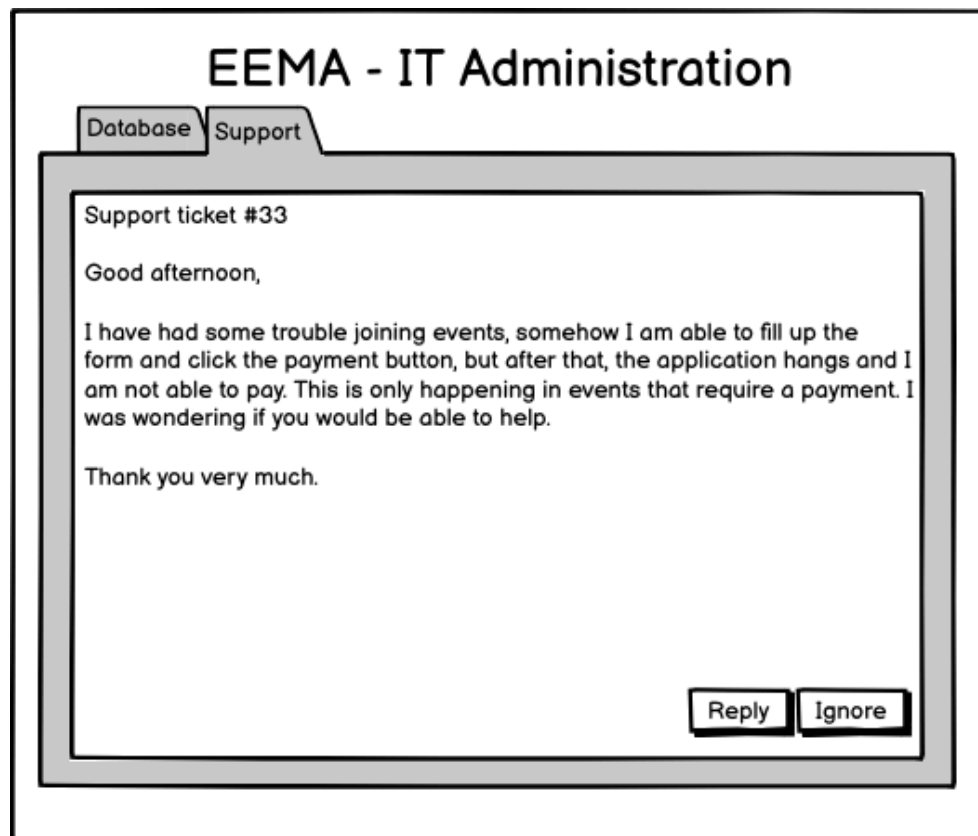


Figure 13: IT support screen mockup image. Marta Cuevas Rodríguez

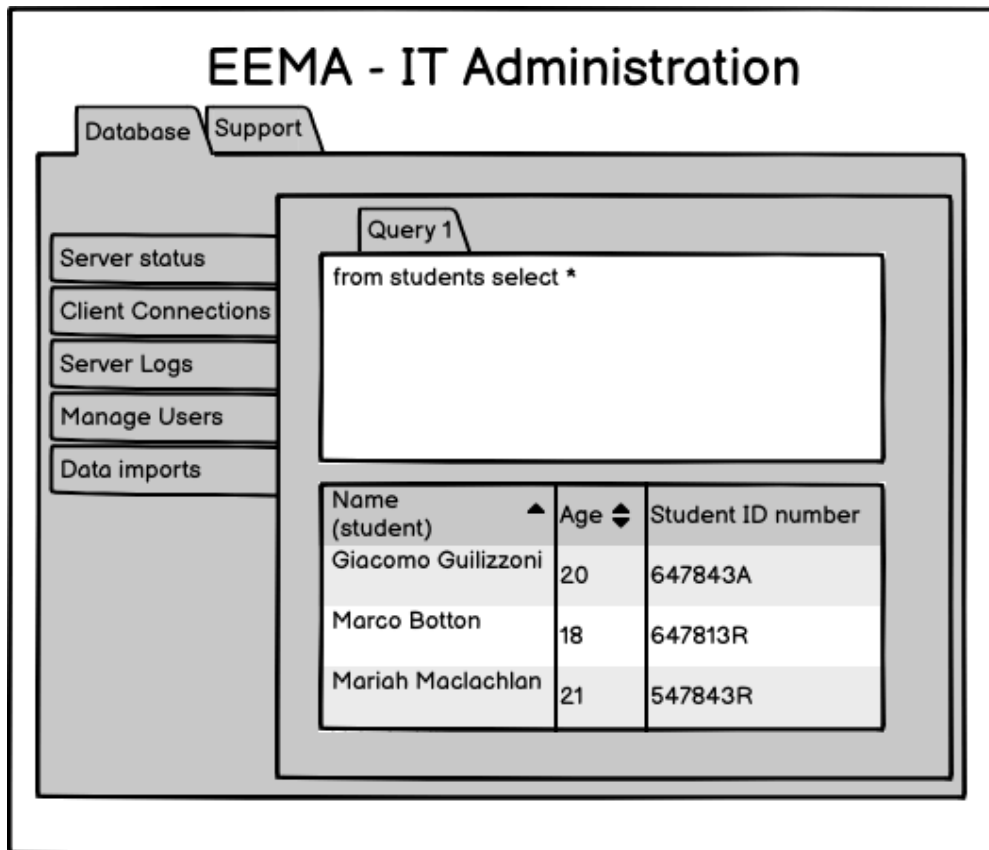


Figure 14: Database screen mockup image. Marta Cuevas Rodríguez

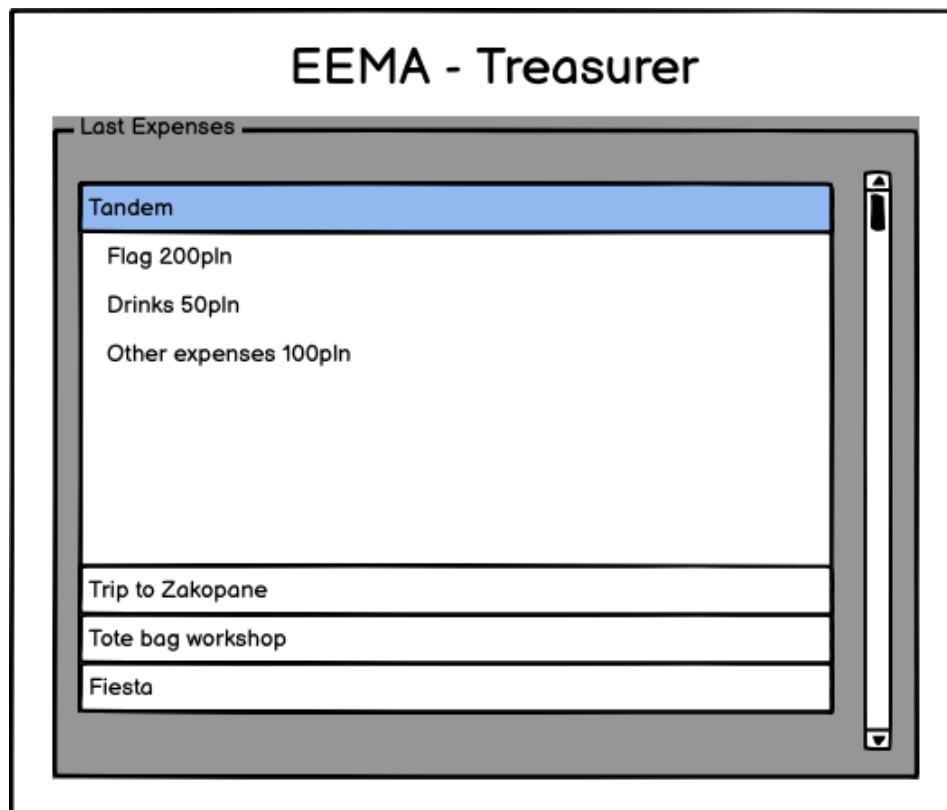


Figure 15: Treasurer screen mockup image. Marta Cuevas Rodríguez

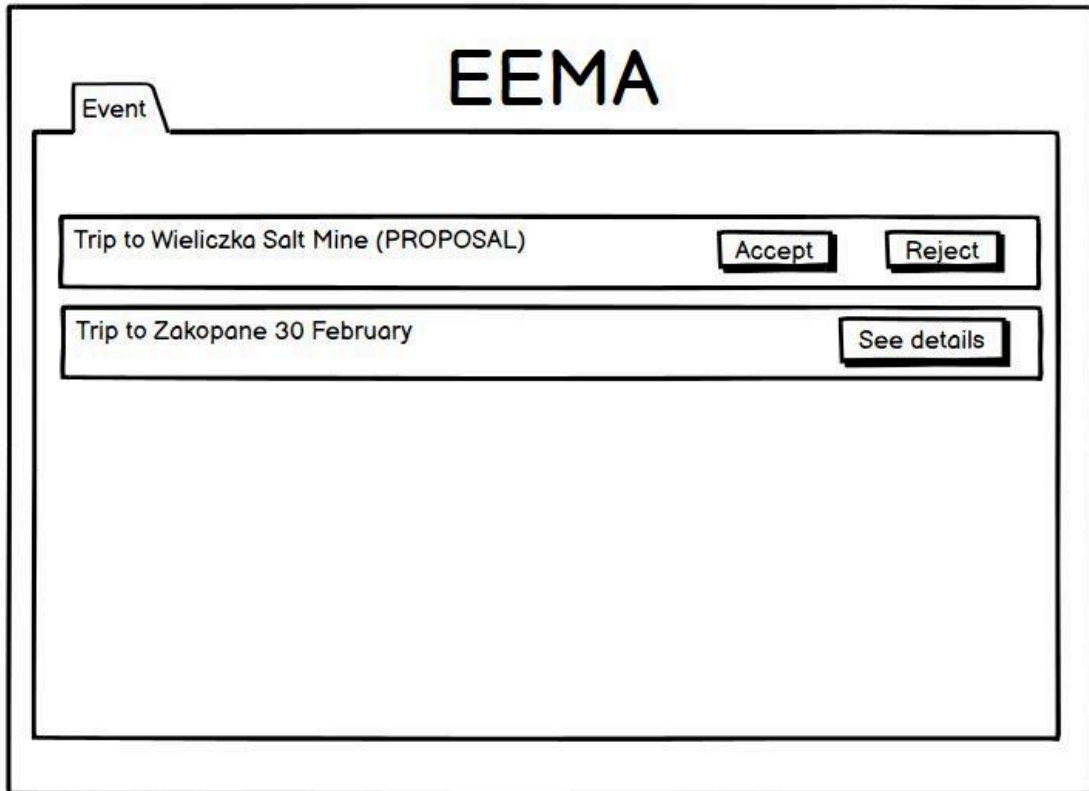


Figure 16: Proposal and events screen mockup image. Julio del Junco Prieto

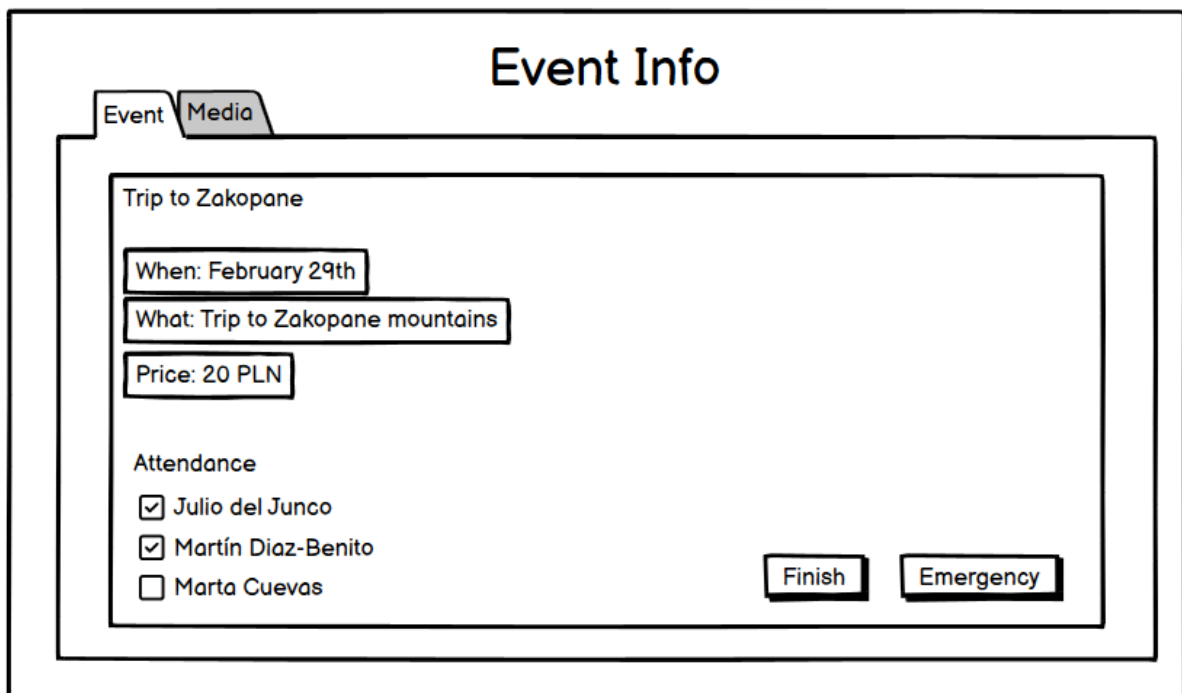
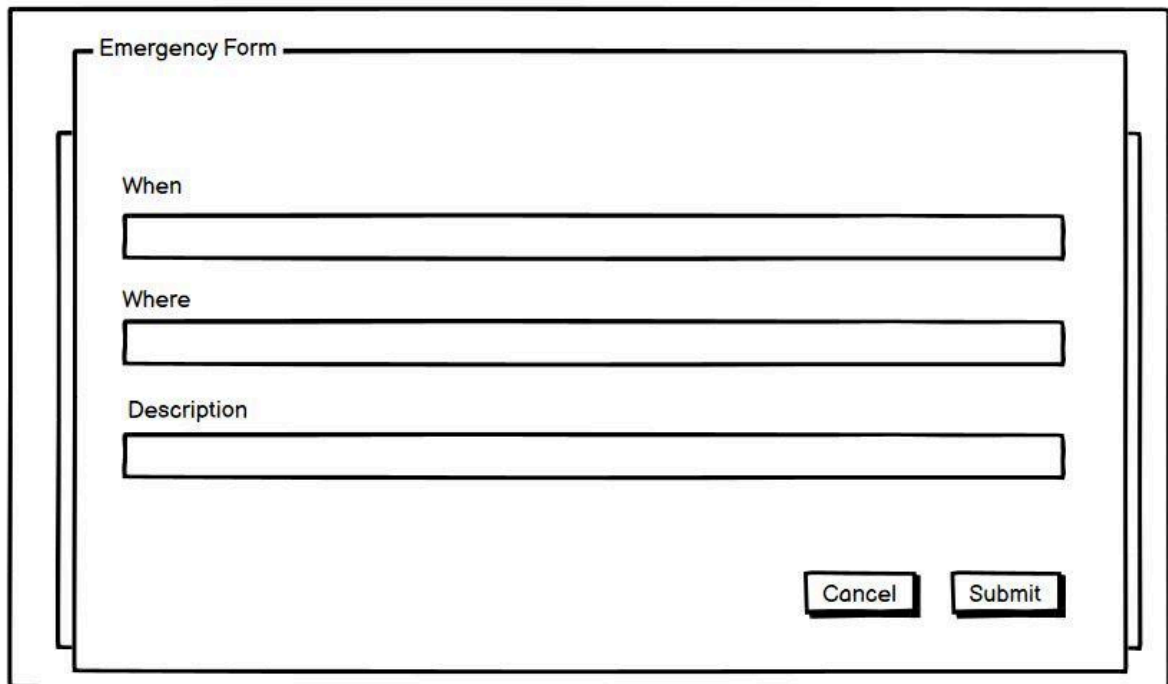


Figure 17: Events details screen mockup image for Monitor. Julio del Junco Prieto



Emergency Form

When

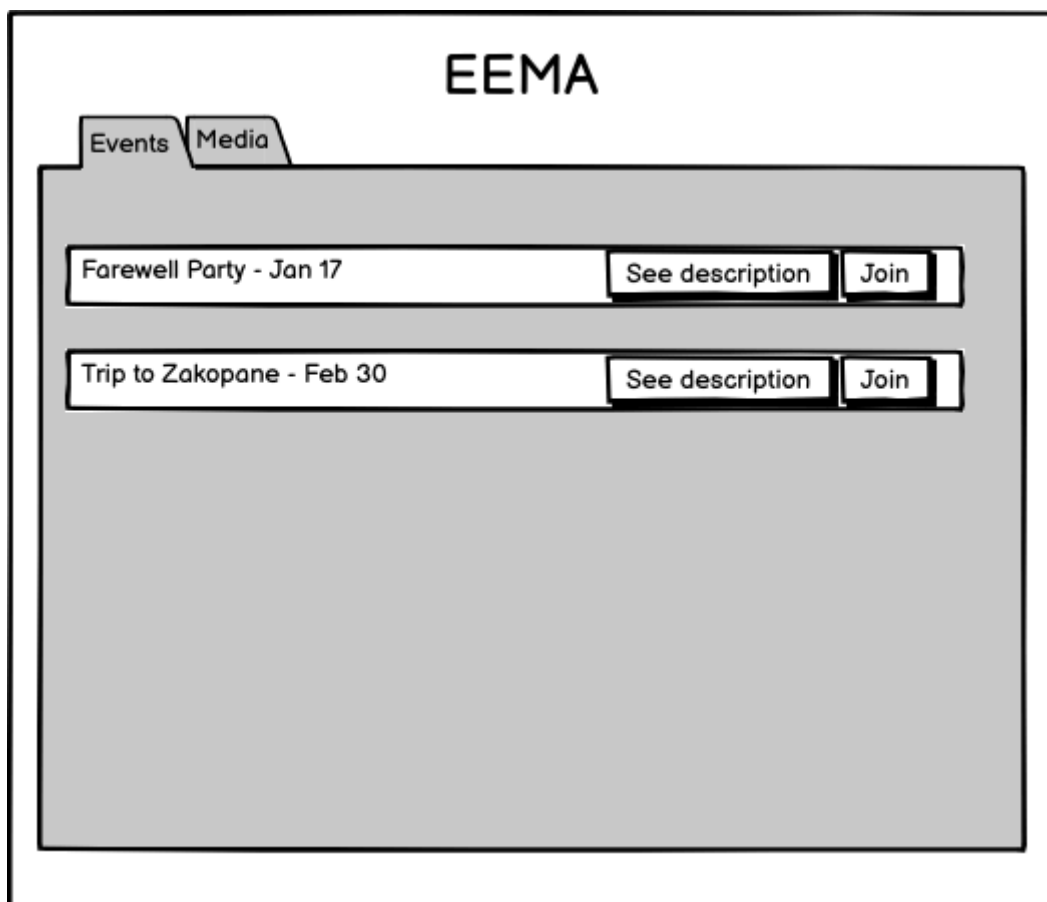
Where

Description

Cancel Submit

This is a screen mockup for an 'Emergency Form'. It features a title bar at the top with the text 'Emergency Form'. Below the title bar, there are three input fields: 'When', 'Where', and 'Description'. At the bottom right of the form, there are two buttons: 'Cancel' and 'Submit'.

Figure 18: Emergency form screen mockup image. Julio del Junco Prieto



EEMA

Events Media

Farewell Party - Jan 17 See description Join

Trip to Zakopane - Feb 30 See description Join

This is a screen mockup for the 'EEMA' application. The title 'EEMA' is centered at the top. Below the title, there are two tabs: 'Events' and 'Media'. The 'Events' tab is selected. Below the tabs, there is a list of events. Each event entry consists of a text box containing the event name and date, followed by two buttons: 'See description' and 'Join'. The events listed are 'Farewell Party - Jan 17' and 'Trip to Zakopane - Feb 30'.

Figure 19:Events list screen mockup image. Martín Diaz-Benito Alvarez

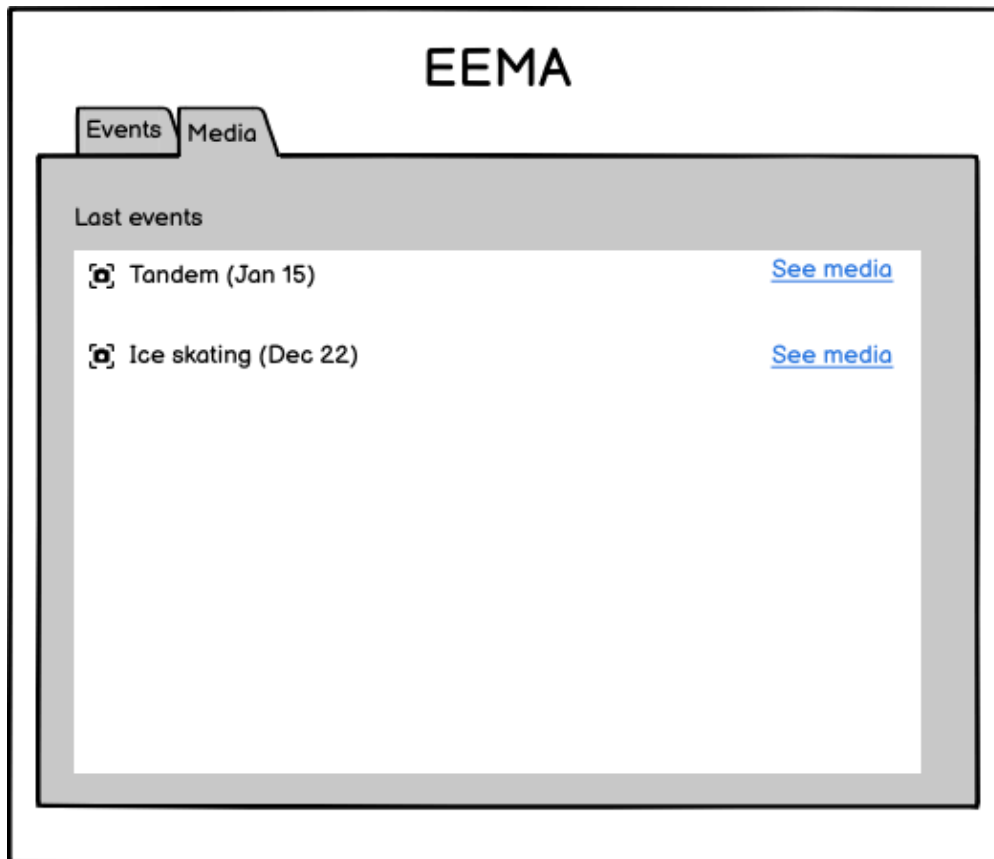


Figure 20: Media screen mockup image. Martín Diaz-Benito Alvarez

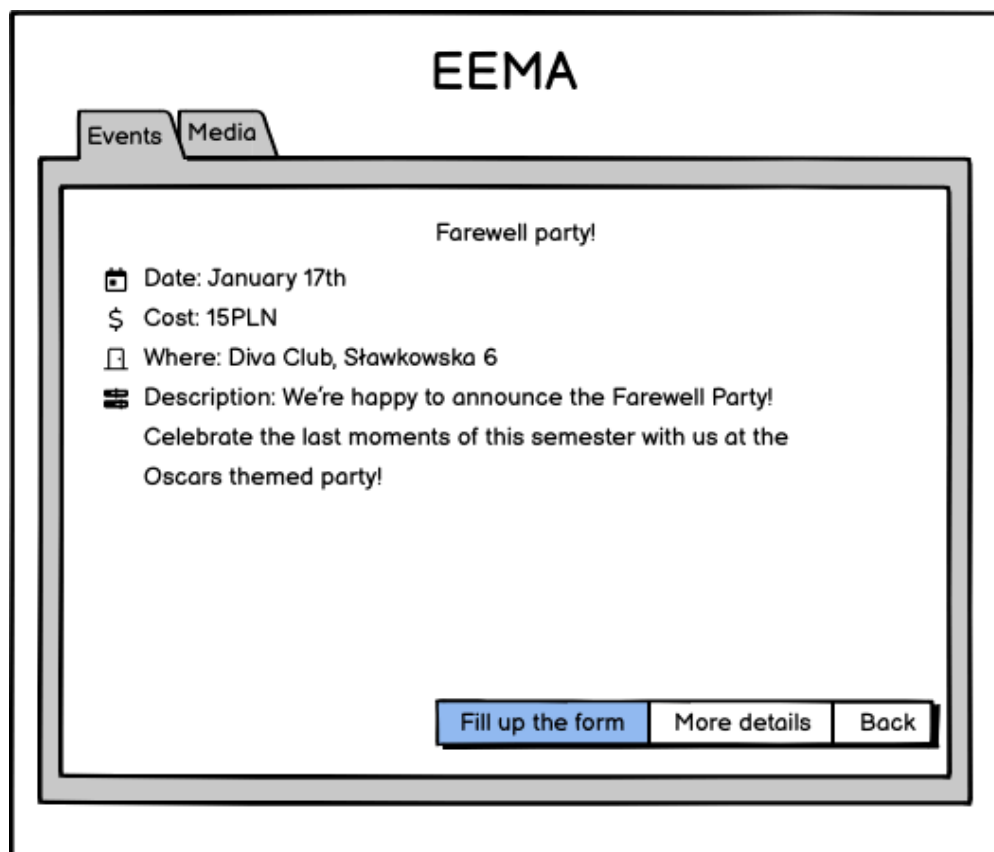


Figure 21: Event description screen mockup for Student image. Martín Diaz-Benito Alvarez

8. Testing - Marta Cuevas Rodríguez

IT Maintenance Test Cases		
Test case	Pass	Fail
Schedule routine maintenance	X	
Respond to alert for potential issue	X	
Enter maintenance mode	X	
Perform diagnostics	X	
Apply Updates, patches or fixes	X	
Review and optimize system configurations	X	
Validate and strengthen security protocols		X
Deactivate maintenance Mode	X	
Validate system stability and performance post-maintenance		X

Table 8.1. IT maintenance Test Cases

Record ESN Events Payments Test Cases		
Test case	Pass	Fail
Initiate Payment Recording Process	X	
Generate Invoices or Payment Details	X	
Notify Participants of payments	X	
Record payment transactions	X	
Acknowledge successful payments	X	
Mark participants as fully registered or paid	X	
Generate financial reports		X

Table 8.2. Record ESN Events Payments Test Cases

Manage ESN Funds Test Cases		
Test case	Pass	Fail
Initiate Funds Management Process	X	
Restrict write access to funds	X	
Review funding requests	X	
Allocate funds to approved events	X	
Update financial records	X	
Notify Event organizers	X	
Restore write access to funds		X
Check record ESN event expenses	X	
Check record ESN event payments	X	

Table 8.3. Manage ESN Events Payments Test Cases

Viewing List of upcoming Events Test Cases		
Test case	Pass	Fail
Display Upcoming events section	X	
Retrieve and display List of scheduled events	X	
Select a specific event	X	
Display detailed information about selected event	X	
Navigate back to event list	X	

Table 8.4. Viewing List of upcoming Events Test Cases

Reporting emergencies during Events Test Cases		
Test case	Pass	Fail
Identify emergency situation	X	
Access emergency reporting feature	X	

Provide details of the emergency		X
Trigger immediate notifications	X	
Managers receive and acknowledge alert	X	

Table 8.5. Reporting emergencies during Events Test Cases

Approving or Rejecting Event proposals Test Cases		
Test case	Pass	Fail
Receive notification of new events proposal	X	
Display event proposals section	X	
Access details of submitted event proposals	X	
Evaluate proposal against standards		X
Reject proposal and provide reasons		X
Update status	X	
Notify organizer of rejection or approval	X	

Table 8.6. Approving or rejecting Event proposals

Browsing Events Test Cases		
Test case	Pass	Fail
Select browse events feature	X	
Display List of events categorized by type or date	X	
Show event details	X	
Apply filters to narrow down event list		X
click on event sign up	X	
Navigate back to main menu	X	

Table 8.7. Browsing Events Test Cases

Event Management Test Cases		
Test case	Pass	Fail
Access event management section	X	
Display dashboard of existing events	X	
Select specific event for viewing/managing	X	
Show even information	X	
Manage participant list		X
Notify Event organizers	X	
Handle event logistics and resources		X
Create a new event	X	
Cancel an event	X	
Modify event details	X	

Table 8.8. Event Management Test Cases

Join an Event Test Cases		
Test case	Pass	Fail
Browse available events	X	
Select an Event to join	X	
Select join button	X	
Fill up confirmation form	X	
Confirm participation		X
Update user's participation status		X
Validate user added to attendee list		X
Show payment for paid event	X	
Leave the event	X	

Table 8.9. Join an Event Test Cases

9. Implementation description - Marta Cuevas Rodríguez

The implementation of the ESN Events Management System is envisioned as a holistic approach, treating it as a complete system rather than a partial prototype. This decision is rooted in the diverse range of functionalities required to cater to the needs of event organizers, participants, and administrators. Opting for a comprehensive system ensures a seamless integration of features, providing end-to-end functionality.

Our development strategy embraces an iterative and incremental model, facilitating continuous refinement and enhancement of features throughout different phases. Beginning with a thorough analysis of functional and non-functional requirements, we collaborate closely with stakeholders to polish and validate system specifications.

The subsequent system design phase focuses on creating a robust architecture and database structure to support all planned functionalities. Simultaneously, user interface design aims at delivering an intuitive and user-friendly experience.

Implementation involves leveraging modern web development technologies for both frontend and backend components. Secure payment gateways will be integrated to handle event payments, and stringent user authentication and authorization mechanisms will ensure data security. Workflows for event proposal submission and approval will also be developed.

The testing phase will be rigorous, encompassing unit tests, integration tests, and system tests to identify and address any issues that may arise. Deployment will involve placing the system on a reliable hosting infrastructure, with configurations for databases, servers, and other necessary components.

Training sessions will be conducted for administrators, organizers, and participants, accompanied by comprehensive documentation for both users and developers. The maintenance and updates phase will involve continuous monitoring of system performance and periodic updates to meet emerging requirements and enhance overall efficiency.

While the initial release will cover core functionalities, subsequent iterations will focus on enhancements and refinements based on user feedback and evolving needs within the ESN community. Additionally, the implementation strategy may involve developing prototypes for specific features or modules to gather early feedback and align with user expectations.

10. Summary and conclusions

10.1 Verification of the project - Marta Cuevas Rodríguez

The verification of the ESN Events Management System will be a thorough and systematic process. We will begin by tracing each requirement outlined in the project's specifications

and design documents to specific features or components in the final implementation. Functional testing will be conducted to ensure that every aspect of the system performs as intended, with test cases created for each functionality. User Acceptance Testing (UAT) will involve end-users to gather feedback on usability and alignment with their needs.

Performance and security testing will be integral, assessing the system's ability to handle various loads and ensuring robust protection of user data. A meticulous code review will address coding standards and security vulnerabilities, while documentation will be reviewed to ensure accuracy and alignment with implemented features. Compliance checks will verify adherence to relevant regulations and standards. Continuous feedback from stakeholders will drive iterative improvements, and a final review will ensure that the implemented system meets the project's objectives and user requirements.

10.2 Verified assumptions for implementation - Marta Cuevas Rodríguez

Verified assumptions for the ESN Events Management System implementation include the comprehensive nature of the envisioned system, treating it as a complete entity rather than a partial prototype. Assumptions also involve the successful utilization of an iterative and incremental development model, ensuring continuous refinement of features throughout various phases. Collaborative stakeholder engagement for requirement validation is another verified assumption, along with the effective integration of modern web development technologies for frontend and backend components. The successful deployment of secure payment gateways, robust user authentication, and authorization mechanisms, as well as the implementation of event proposal workflows, are key assumptions verified for the system's implementation. Ongoing training sessions, documentation, and iterative improvements based on user feedback are integral components of the envisioned implementation process.

In our case, we do not have any implementations for the project so verification can not be done, only assumptions for implementation.

10.3 Final remarks and conclusions - Martín Diaz-Benito Álvarez

This project was a big challenge for us, since we had never had any courses dedicated to software engineering and most things were very new to us, but we think that it has allowed us to have a good vision on making something big, so we would be able to design our own app in the future. We had been taught a lot of programming but this project has made us able to think how to apply all of this knowledge for software development. We also learnt how to use multiple tools like visual paradigm or starUML, which will be helpful in our career, and to draw diagrams that can even be useful in other areas of computer engineering, like entity relationship diagrams. We will for sure use all of this knowledge in the future and we think it's important for every single computer scientist.

SOURCES

Balsamiq tutorial <https://balsamiq.com/tutorials/>

Star UML Tutorial <https://docs.staruml.io/>

Activity diagram tutorial, visual paradigm

<https://www.visual-paradigm.com/tutorials/how-to-draw-activity-diagram-in-uml/>

Deployment diagram tutorial, visual paradigm

<https://www.visual-paradigm.com/tutorials/how-to-draw-deployment-diagram-in-uml/>

State machine Diagram tutorial, visual paradigm

<https://www.visual-paradigm.com/tutorials/how-to-draw-state-machine-diagram-in-uml/>

Software testing, IBM <https://www.ibm.com/topics/software-testing>

Use Cases

<https://www.usability.gov/how-to-and-tools/methods/use-cases.html#:~:text=A%20use%20case%20is%20a,when%20that%20goal%20is%20fulfilled.>

LIST OF FIGURES, TABLES

Figure 1 - Organizational schema

Figure 2 : Organizational schema

Figure 3: Use Case diagram for Economic & IT Staff

Figure 4: Use Case diagram for Management Staff

Figure 5: Use Case diagram for Student Organization

Figure 6: Activity diagram for Economic & IT Staff.

Figure 7: Activity diagram for Management Staff.

Figure 8: Activity diagram for Student Organization.

Figure 9: Entity Relationship Diagram.

Figure 10: State Machine Diagram for Login and Register

Figure 11: State Machine Diagram for App functioning

Figure 12: Deployment Diagram of System.

Figure 13: IT support screen mockup image

Figure 14: Database screen mockup image.

Figure 15: Treasurer screen mockup image

Figure 16: Proposal and events screen mockup image.

Figure 17: Events details screen mockup image for Monitor.

Figure 18: Emergency form screen mockup image.

Figure 19: Events list screen mockup image.

Figure 20: Media screen mockup image

Figure 21: Event description screen mockup for Student image

Image 1: Login screen example

Image 2: Main screen example

Table 2.3.1 Input and output

Table 8.1. IT maintenance Test Cases

Table 8.2. Record ESN Events Payments Test Cases
Table 8.3. Manage ESN Events Payments Test Cases
Table 8.4. Viewing List of upcoming Events Test Cases
Table 8.5. Reporting emergencies during Events Test Cases
Table 8.6. Approving or rejecting Event proposals
Table 8.7. Browsing Events Test Cases
Table 8.8. Event Management Test Cases
Table 8.9. Join an Event Test Cases