
# ESN Ticketing App 2.0

## Project Documentation

Version 2.1.0

Pelivanov[1], Milivojčević[1], and Zrimec[1]

[1]University of Primorska

August 30, 2024

# Authors

Students: Dimitar Pelivanov, Milan Milivojć
Mentor: dr. Tatjana Zrimec
*University of Primorska, Faculty of Mathematics, Natural Sciences and Information Technologies*

# Links

GitHub: *Link to repository (Branch: new-UI/backend)*[1]
Notion: *Link to Notion team workspace*[2]
Demo Video: *Link to Drive*[3]

# Contents



[1] https://github.com/milivojcevic6/ticketing-app.git
[2] https://shorturl.at/m4T7W
[3] https://drive.google.com/file/d/1RDunMQZZ7B8_xjljO0o0OjSxPSTJxd-i/view?usp=sharing

# 1 Introduction

The Erasmus Student Network (ESN) is a renowned student organization, dedicated to supporting international students during their study abroad. ESN sections, established within universities worldwide, play a pivotal role in organizing events, trips, and social activities, fostering a welcoming and inclusive environment for international students.

ESN volunteers tirelessly dedicate their time and effort to ensure the smooth operation of these events, allowing international students to connect with the local community, make friends, and broaden their horizons. However, as the scale and complexity of the events increase, it becomes crucial to leverage the technology and streamline the administrative tasks to enhance the effectiveness and efficiency of event management.

In this context, the need for applications tailored to the needs of ESN volunteers becomes evident. These applications can provide centralized platforms for event management, simplifying processes such as event posting, attendee registration, ticket generation, and ticket scanning. By automating these tasks, ESN volunteers can focus more on creating meaningful experiences for international students, rather than getting caught up in administrative burdens.

In the following sections, this document presents a problem statement, highlighting the challenges faced by ESN volunteers and proposes a solution that aligns with ESN's vision. By creating an application that addresses these challenges, ESN sections can efficiently manage events, simplify registrations, generate tickets, and streamline the check-in process. This comprehensive solution not only facilitates the work of ESN volunteers, but also enhances the overall experience of international students participating in ESN events.

Our work is second edition of this project, building upon the foundation laid by previous students, Karolina T. and Milan M., who developed the initial solution. In their work, they successfully created a system with two distinct user roles: "section" and "student." The section user had the capability to create, update, and delete events, while the student user could register for these events. Upon registration, a ticket featuring a QR code was generated for each participant, which the section user could then scan to approve the participant.

In this current iteration, the objectives are to enhance the user interface, enable students to provide feedback on the events they attended, and introduce an online payment system for event registration. Additionally, section users will now have the ability to view user feedback and access comprehensive statistics on all past and present events through their profiles.

# 2 Problem Definition

In most ESN sections, the current event registration process involves several manual steps, leading to inefficiencies and time-consuming tasks. The process typically includes the following:

1. Student registration: Students interested in attending an event fill out an online form, providing their details and preferences.
2. Excel-based registrations: ESN sections receive an Excel file containing the registration data, which needs to be manually downloaded, sorted, and organized by section volunteers.
3. In-person payments: Students are required to make payments in person, either at the ESN section office hours or during designated additional payment sessions.
4. Manual attendance marking: ESN sections mark the attendance of paid students manually in the Excel sheet, which can be error-prone and time-consuming.
5. Manual attendance checking: During the event, ESN volunteers cross-reference the Excel sheet to verify the attendance of registered students, resulting in a cumbersome process.

The current manual event registration process is outdated and inefficient, relying on Excel sheets and manual data management. It presents challenges in terms of accuracy, data organization, and the time spent by ESN volunteers. Additionally, maintaining real-time updates and seamless communication between ESN sections and event attendees becomes challenging, as well as data gets lost.

To address these issues, version one of this project managed to simplify event registration, enhance data accuracy, and improve overall efficiency in event management.

Now, the goal is to enhance the data analysis process by integrating dashboards, enabling online payments, and providing users with the ability to follow the activities of their selected sections.

# 3 System Architecture

The proposed ESN Ticketing App follows a classical client-server architecture. The frontend is developed using React JS [4], a JavaScript library for building user interfaces, while the backend is now built with Django [5], a high-level Python web framework known for its simplicity and scalability. This is a bit different in comparisons to the previous version, since it was using Java Spring Boot, but this time we decide to go with the Python framework since it allows us easier data manipulation for our data visualizations.

The frontend communicates with the backend through HTTP requests using the Axios library [6], a widely adopted JavaScript library for making asynchronous HTTP requests. The backend interacts with the frontend through RESTful APIs, which provide endpoints for various functionalities such as event retrieval, registration submission, payment processing, and QR code generation. These APIs are implemented using Django's views and Django REST Framework (DRF), which map incoming requests to appropriate view methods.

For database interactions, the backend utilizes Django's Object-Relational Mapping (ORM) system, which provides an abstraction layer for database operations. This allows developers to work with Python objects rather than writing SQL queries directly, simplifying data access. Django ORM handles the mapping of objects to database tables, query generation, and transaction management.

Overall, the ESN Ticketing App follows a client-server architecture, where the React frontend communicates with the Django backend through HTTP requests. The backend processes these requests, interacts with the MySQL database using Django ORM, and returns the appropriate responses to the frontend. This architecture ensures separation of concerns, scalability, and maintainability of the application.

[4] v17.0.2, https://react.dev/

[5] v4.2, https://www.djangoproject.com/

[6] https://axios-http.com/

# 4 Functional Requirements

1. **Authentication and Authorization:** The app should provide secure authentication and authorization mechanisms to ensure that only authorized users can access and perform specific actions within the system. Users will need to authenticate using their registered email address and password to access the app. Role-based access control will be implemented to assign different levels of access and permissions based on user roles.

2. **Event Creation:** ESN sections should be able to create and publish events on the platform. They should be able to provide event details such as title, description, date, time, location, capacity, and any associated fees.

3. **Event Registration:** Students should be able to browse and search for events on the app. They should be able to register for events they are interested in attending by providing their personal details.

4. **Event Management:** ESN sections should have access to a dashboard or administrative interface where they can manage events. This includes editing event details, tracking event registrations, and viewing attendee lists.

5. **Ticket Generation:** After successful event registration, the app should generate a unique QR code ticket for each registered student, which can be stored and accessed from their profile.

6. **Ticket Scanning:** ESN volunteers should be able to use the app to scan QR codes on tickets to verify attendee presence at the event. The app should provide instant confirmation of the ticket's validity.

7. **User Profiles:** The app should allow both, students and sections, to view and manage their profiles and update information. Students should be able to add ESN sections to their list of followed sections, view registered events, and access their QR code tickets.

# 5 System Design

## 5.1 Database Schema

In the database schema of the ESN Ticketing App, the following entities and their relationships are defined (Figure 1):

1. **Section:** Represents an ESN section. A section can organize multiple events, but each event is tied to one specific section. This establishes a one-to-many relationship between the Section and Event entities.

2. **Event:** Represents an event organized by a section. Each event can have multiple unique tickets associated with it, creating a one-to-many relationship between the Event and Ticket entities. Django model: Event model will reference the Section model with a ForeignKey, and will have its own OneToMany relationship with the Ticket model.

3. **Ticket:** Represents a unique ticket for an event. A ticket is associated with a single user who has registered for the event, establishing a one-to-one relationship between a ticket and a user, and a many-to-one relationship between the Ticket and Event entities. Django model: Ticket model will have ForeignKey fields for both the Event and User models, ensuring a connection between each ticket, its event, and the user who owns it.

4. **User:** Represents a student using the ESN Ticketing App. A user can register for multiple events, resulting in multiple tickets being associated with them. Therefore, there's a one-to-many relationship between the User and Ticket entities. Django model: User model (extended from Django's default User model or custom) will have a ForeignKey relationship with the Ticket model.

5. **UserSection:** Represents the relationship between a user and a section. A user can follow multiple sections, and each section can have multiple users as members. This establishes a many-to-many relationship between the User and Section entities. Django model: The UserSection relationship will be implemented using Django's ManyToManyField in either the User or Section models.

6. **UserSection:** Represents the relationship between a user and a section. A user can follow multiple sections, and each section can have multiple users as members. This establishes a many-to-many relationship between the User and Section entities. Django model: The UserSection

relationship will be implemented using Django's ManyToManyField in either the User or Section models.

7. **Feedback:** Represents feedback left by a user for a specific event. The feedback includes a grade (from 1 to 5) that represents the rating given by the user to the event. There is a many-to-one relationship between the Feedback and Event entities, as each event can receive multiple feedback entries. Similarly, there is a many-to-one relationship between Feedback and User, where each user can leave multiple feedbacks but only one per event. Django model: Feedback model will have ForeignKey fields for both the Event and User models, ensuring each feedback is uniquely associated with a user-event pair.

8. **Card:** Represents a unique card associated with each user. Each user has exactly one card, and this card contains details such as the code, expiration date, and activation date. There is a one-to-one relationship between the Card and User entities, ensuring each user has a single card, and each card is associated with only one user.

### 5.2 Application design

The ESN Ticketing App was designed from scratch, starting with a paper prototype to conceptualize its features and user interface. The frontend design of the app incorporates the official ESN colors and follows the ESN Visual Identity guidelines to maintain brand consistency. The app's logo was created using Canva, ensuring a visually appealing and recognizable brand identity. The Bootstrap framework was utilized to enhance the frontend presentation, providing a responsive and visually appealing layout. On the backend, the app leverages the Django REST Framework, enabling efficient handling of requests and responses. The models, views, serializers, and controllers were developed from scratch, tailored to the specific requirements of the ESN Ticketing App. This custom backend design ensures a robust and scalable architecture to support the app's functionalities.

## 6 Implementation

The implementation of the ESN Ticketing App involved the use of several technologies and tools. The development environment consisted of Webstorm and PyCharm as the primary integrated development environments

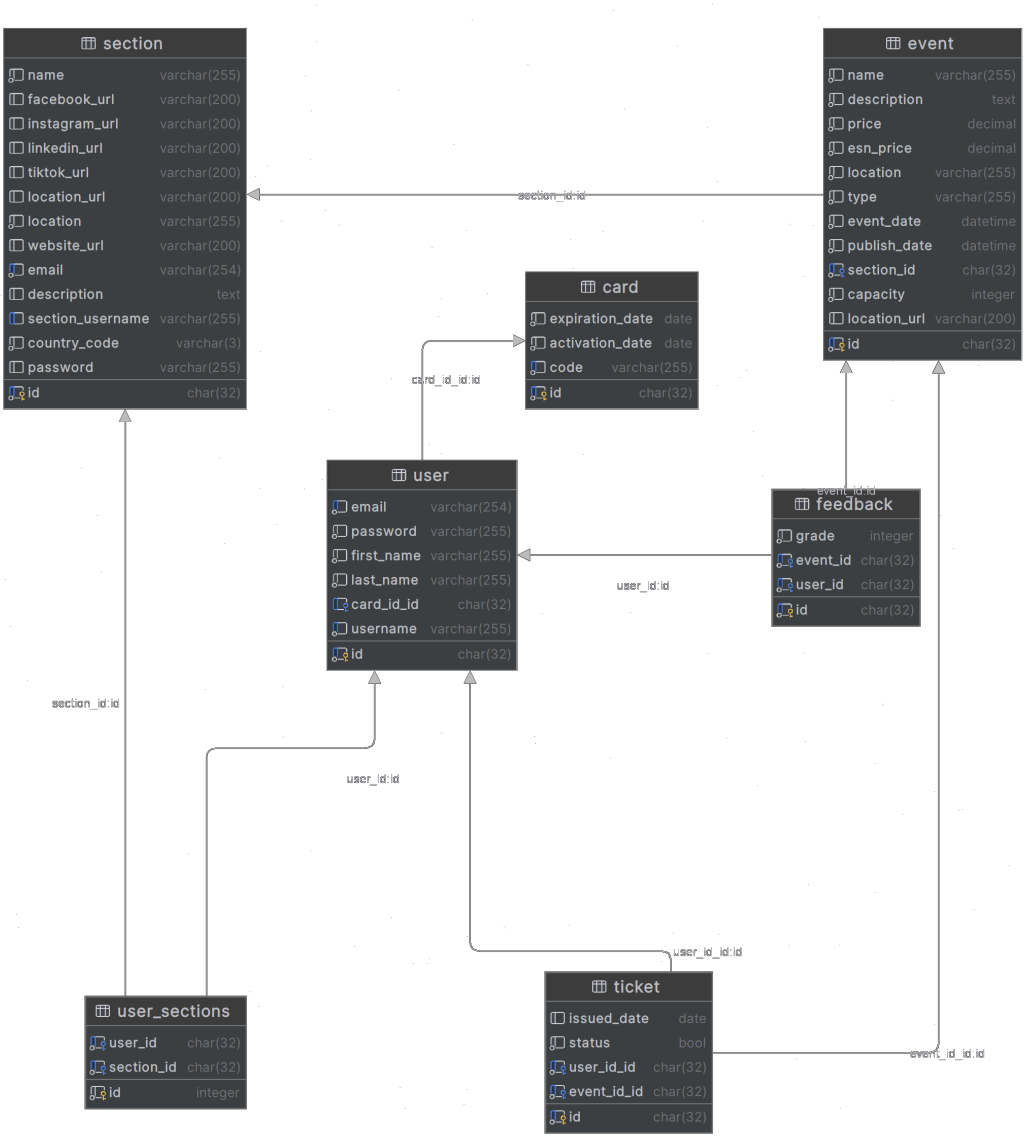


Figure 1: ESN Ticketing App database schema

(IDEs) for frontend and backend development, respectively. Postman was used to test the API endpoints throughout development.

The codebase was structured into separate frontend and backend directories. Git, a distributed version control system, was employed for source code management and collaboration among team members.

In the backend, we implemented CRUD (Create, Read, Update, Delete) operations for the user, section, event, and ticket entities. This allowed for efficient management and manipulation of data related to these entities. Authentication and authorization were implemented using Spring Security, which provided robust security features and role-based access control. The backend consisted of controllers, models, repositories, and services for handling various functionalities related to authentication, events, sections, tickets, and users. Additionally, the WebSecurityConfig class was created to configure Spring Security and manage path access restrictions.

The frontend of the app was developed using React, a popular JavaScript library for building user interfaces. Axios, a promise-based HTTP client, was utilized to make API requests from the frontend to the backend. The frontend codebase included six components: Home, Login, Profile, TicketWallet, CheckTickets, and Footer. The LoginContext was used to store the current user information, while the TicketWallet component is accessible only to students, and the CheckTickets component is accessible only to the sections. A friendly phone view was also developed with a significant emphasise on the responsiveness of the app.

One of the significant challenges encountered during the implementation phase was integrating the latest version of Spring Security, which introduced significant changes compared to the previous version. Overcoming this challenge required thorough understanding and adaptation of the new features and configurations provided by Spring Security.

## 6.1 Graphical User Interface

The Graphical User Interface (GUI) of the ESN Ticketing App was meticulously designed to provide an intuitive and visually appealing experience for both, ESN sections and students. By incorporating the official ESN colors and following the ESN Visual Identity guidelines, the GUI maintains brand consistency and fosters a sense of familiarity for users. The frontend development was implemented using React, a popular JavaScript library for building user interfaces, and the Bootstrap framework was utilized to ensure a responsive and aesthetically pleasing layout.

### 6.1.1 Login and Registration

The login and registration pages, represented in Figures 2 and 3, respectively, demonstrate a minimalist and user-friendly design. The login page provides a straightforward interface for existing users to access their accounts. If users are not logged in, they are automatically redirected to the login page. To register, users can easily switch to the registration form by selecting "Register" at the top of the form. The registration page (Figure 3) adapts the login form to include the necessary fields for user registration.


Figure 2: Login Page

### 6.1.2 Home Page

Figure 8 showcases the home page of the app, which serves as the primary entry point for users. The design offers an engaging layout, with options for students to explore and search for events they are interested in attending. The responsive design of the home page ensures a seamless user experience for both desktop and mobile users, allowing them to conveniently access event information and registration options.

### 6.1.3 Footer

The footer section of the app displays essential information, including the names of the developers, the ESN organization, and the app logo. It also offers easy access to ESN's social media links, promoting smooth communication


Figure 3: Registration Page

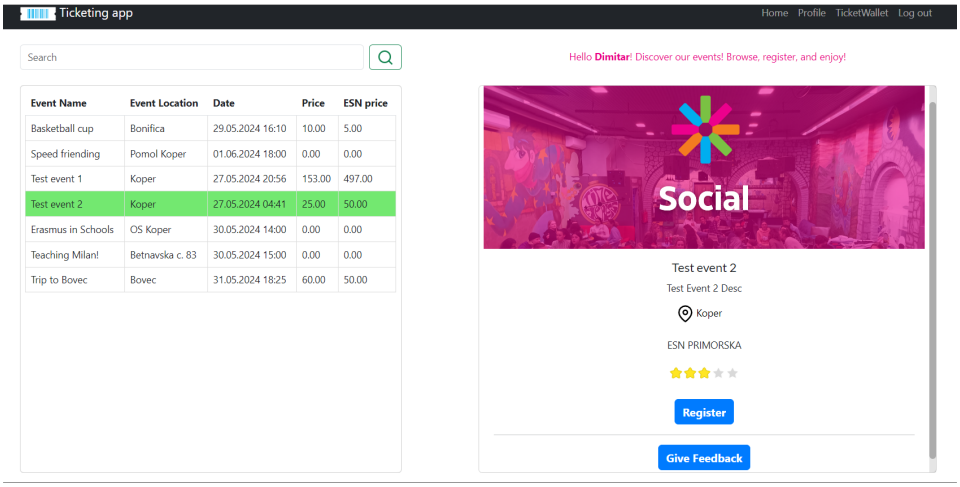


Figure 4: Home Page - User view.

and engagement with the ESN community. The inclusion of the app logo enhances brand recognition, insuring that users can easily identify and relate to the app within the context of the ESN ecosystem.

### 6.1.4 Editing user profile and following sections

The user profile page, as shown in Figure 6, offers users the option to edit certain attributes, including their first name, last name, and password. Additionally, users can select ESN sections to follow, facilitating personalized event recommendations and a tailored experience within the app.

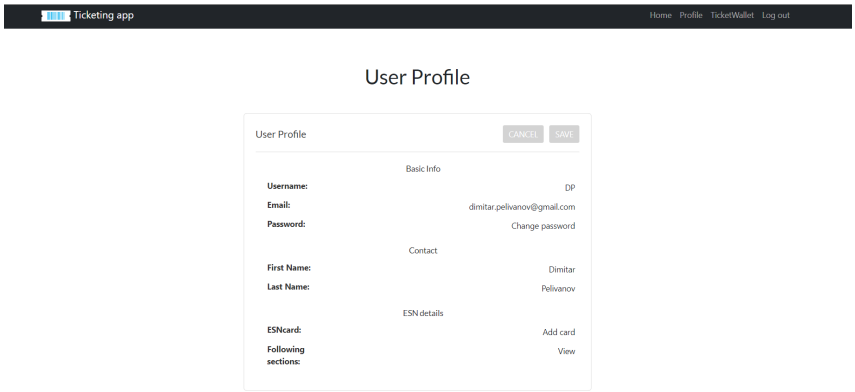


Figure 5: User profile.

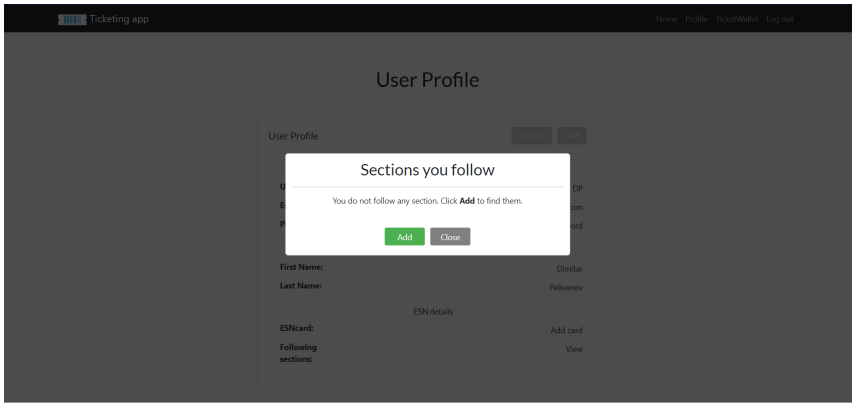


Figure 6: Editing user profile and choosing sections to follow.

### 6.1.5 Home Page for ESN Section

Figure 7 illustrates the home page view, tailored specifically for ESN sections. This view provides a comprehensive overview of all events organized by the section, allowing section administrators to easily track and manage their events. The home page design enables efficient event editing and ensures a smooth user experience for section administrators.

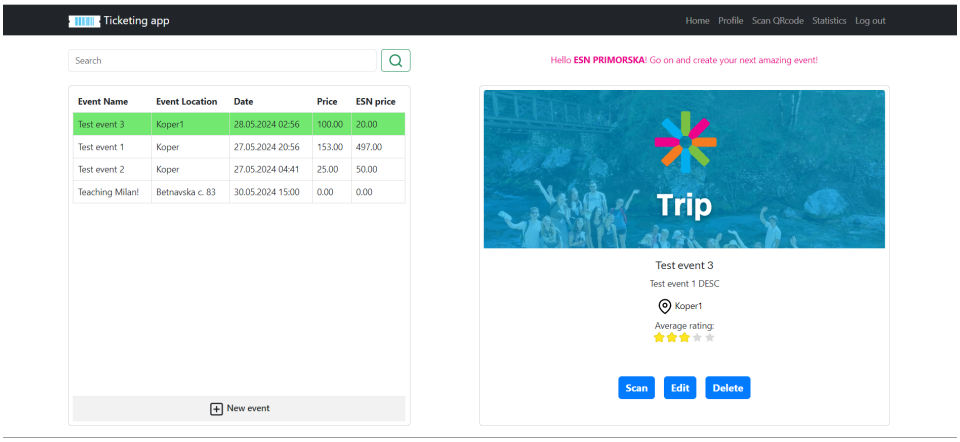


Figure 7: Home Page - Section view.

### 6.1.6 Event Registration for Student User

Once a student user has specified the sections they wish to follow, the home page view (Figure 8) displays events from the followed sections. This enables students to conveniently explore and register for events that align with their interests, streamlining the event registration process.

### 6.1.7 User Wallet

Figure 9 shows the page design for the user's wallet, where students can access their QR code tickets for registered events. The user wallet provides a comprehensive view of their registered events, and offers easy navigation for mobile users.

### 6.1.8 QR Code Reader

Figure 10 illustrates the QR code reader functionality, displaying the information contained in the QR code. The QR code provides essential details, including the event name, the user's full name, and the date when the ticket

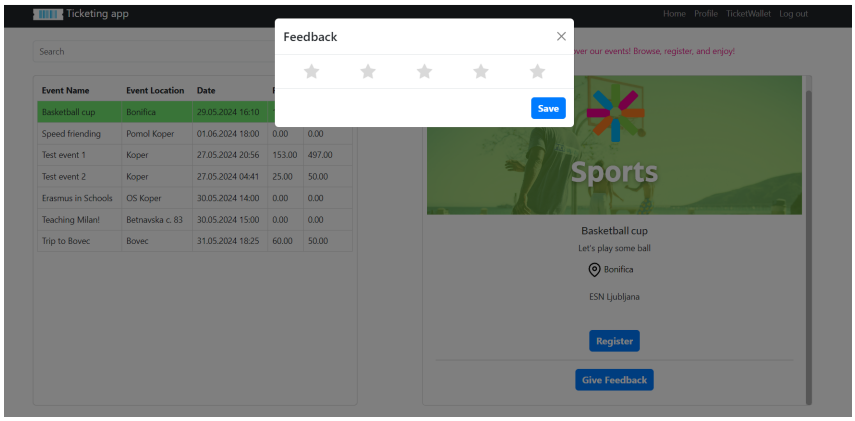


Figure 8: Event Registration and Event Feedback for student users.

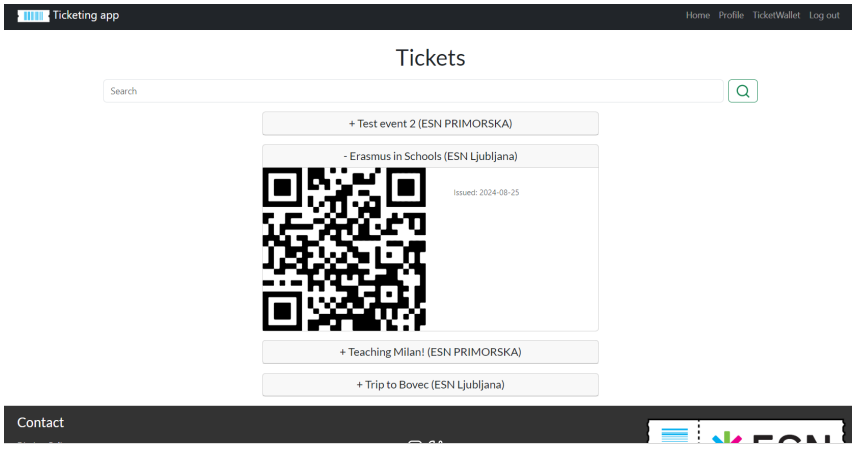


Figure 9: Student user ticket wallet.

was created. This feature enables seamless and efficient ticket scanning during event check-ins.

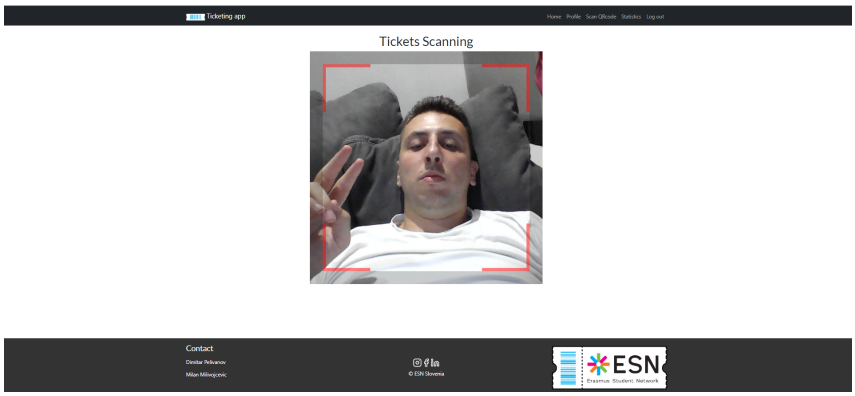


Figure 10: Section ticket scanner.

The thoughtfully crafted User Interface of the ESN Ticketing App ensures an enjoyable and efficient user experience. It reflects the dedication of the development team to creating an application that aligns with the needs and visual identity of ESN. The UI not only streamlines event management processes for ESN sections but also offers a seamless event registration and ticketing journey for students, ultimately enhancing the overall effectiveness and engagement of ESN events.

### 6.1.9 Payment

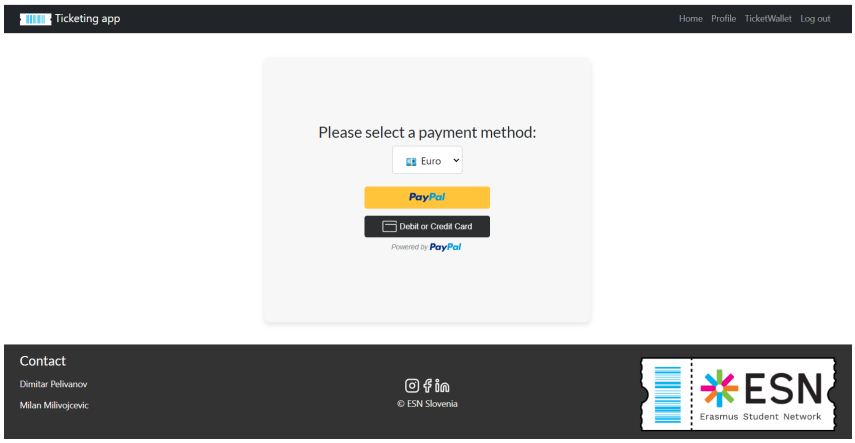


Figure 11: Payment initial interface.

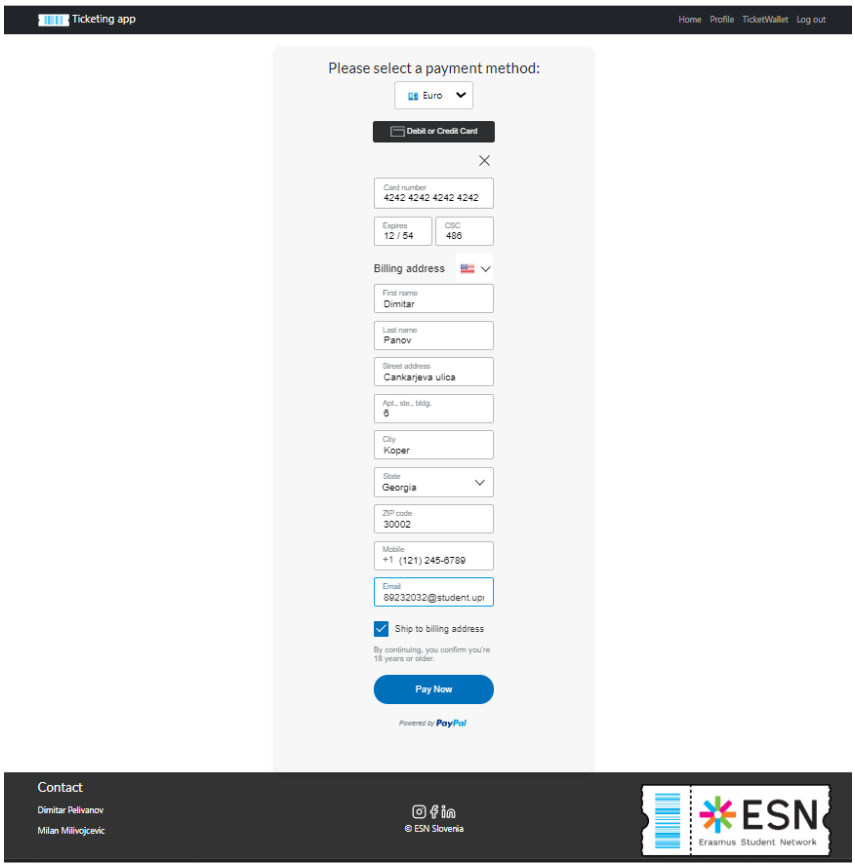


Figure 12: Payment example.

### 6.1.10 Events Dashboard

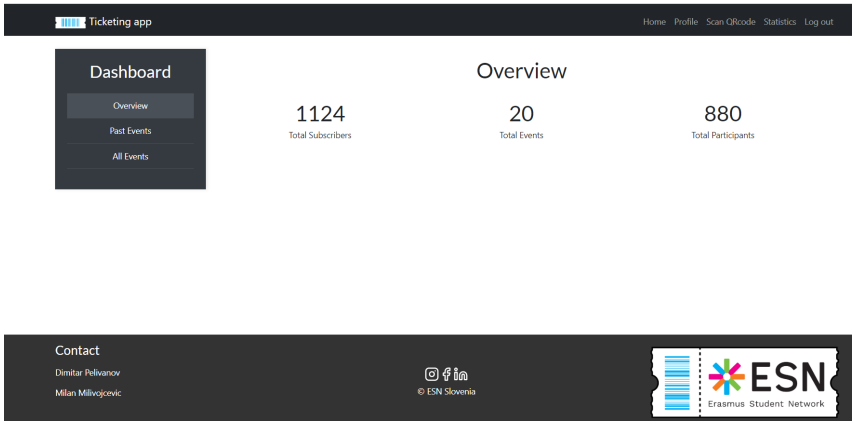


Figure 13: Events Board.


Figure 14: Events Dashboard.

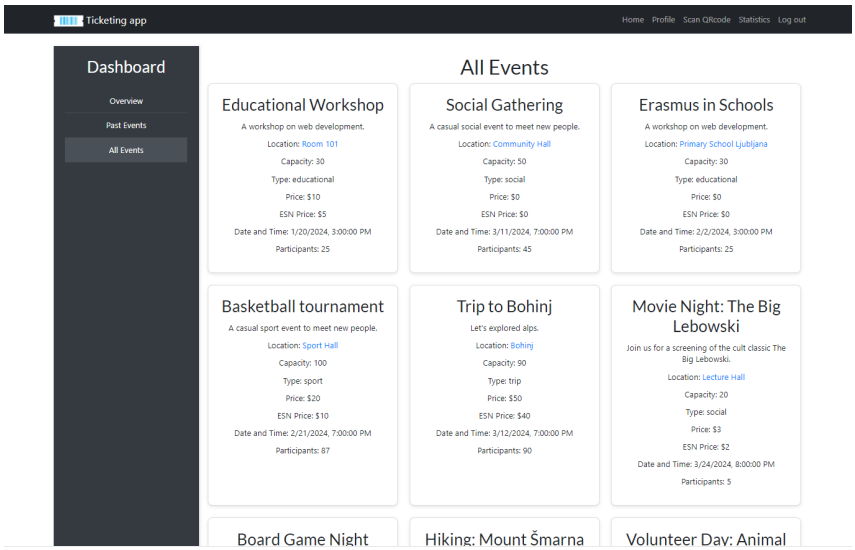


Figure 15: Events Board.

# 7 Future Work

In the future, we have identified several areas for improvement and additional features to enhance the ESN Ticketing App. Our next steps will involve improving the user experience by implementing a more intuitive and visually appealing interface. We aim to focus on refining the user interface design and optimizing the app's usability to provide a seamless event registration and ticketing experience for both ESN sections and students.

Moreover, we plan to integrate login and registration options via social media platforms, as well as the option of directly sharing and promoting events on them.

Additionally, we aim to improve event statistics more to help ESN sections make informed decisions. On the user side, implementing a recommended system will enhance the app by providing personalized event suggestions. These future steps will ensure a more user-friendly, efficient, and engaging experience for both ESN sections and students.

# 8 Contributions

Throughout the development of the ESN Ticketing App v2.0, the project was carried out by a developer team consisting of two members: Dimitar Pelivanov and Milan Milivojčević. Initially, both team members were involved in full-stack development, working collaboratively to tackle various aspects of the project.

To ensure efficient task management, a Notion task tracking system was implemented, allowing the team to organize and assign specific tasks from specific sprints (agile development). During this phase, Dimitar and Milan worked closely together, dividing the tasks based on their expertise and workload (Appendix A).

As the development progressed, a natural division of tasks occurred, with Milan primarily focusing on frontend development and Dimitar taking charge of backend tasks. This division of responsibilities allowed for increased productivity and specialization in their respective areas of expertise.

Despite the specialization, the team maintained a collaborative approach throughout the project. They continuously supported and helped each other in debugging and problem-solving, fostering a cohesive and dynamic development environment. Both team members have a comprehensive understanding of the overall project structure and functionalities.

# Appendices

## A Task Management

All tasks | Board | New

| Aa Task name | Assignee | Sprint | Status |
|---|---|---|---|
| Write project proposal | Milan Milivojcevic, Dimitar P. | Sprint 1 | Done |
| Schedule kick-off meeting | Milan Milivojcevic, Dimitar P. | Sprint 1 | Done |
| Requirements | Dimitar P., Milan Milivojcevic | Sprint 1 | Done |
| Evaluate ESN Ticketing App 1.0 | Dimitar P., Milan Milivojcevic | Sprint 1 | Done |
| Creating Intro Presentation | Milan Milivojcevic, Dimitar P. | Sprint 2 | Done |
| Update CRUD For Users in Django | Dimitar P. | Sprint 3 | Done |
| Update CRUD For Tickets in Django | Milan Milivojcevic | Sprint 3 | Done |
| Update CRUD For Events in Django | Dimitar P., Milan Milivojcevic | Sprint 3 | Done |
| Update GUI of user profile page | Milan Milivojcevic | Sprint 4 | Done |
| Add admin permissions | Dimitar P., Milan Milivojcevic | Sprint 4 | Done |
| Update DB with the respect to requirements | Milan Milivojcevic, Dimitar P. | Sprint 4 | Done |
| Create user section only as admin | Milan Milivojcevic | Sprint 4 | Done |
| Add option to share event on social media | Dimitar P., Milan Milivojcevic | Sprint 5 | Done |
| Update Scan code functionality | Dimitar P., Milan Milivojcevic | Sprint 5 | Done |
| Registration, OAuth, via Social Media | Dimitar P. | Sprint 6 | Done |
| Invoice generator pdf | Dimitar P., Milan Milivojcevic | Sprint 6 | Done |
| Event analytics (Event Attendance) | Dimitar P., Milan Milivojcevic | Sprint 7 | Done |
| Online payment | Dimitar P. | Sprint 7 | Done |