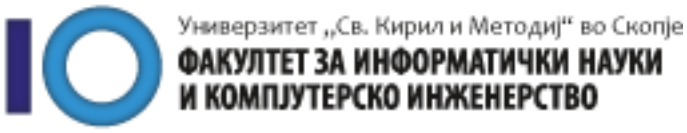


ИНФОРМИРАЈСЕ

БИДЕТЕ ВО ТЕК СО НОВОСТИТЕ ОД ИТ СВЕТОТ

InformirajSe – Documentation

Mentor

Boban Joksimoski

Authors

Antonio Stefanovski

Enes Sejfovski

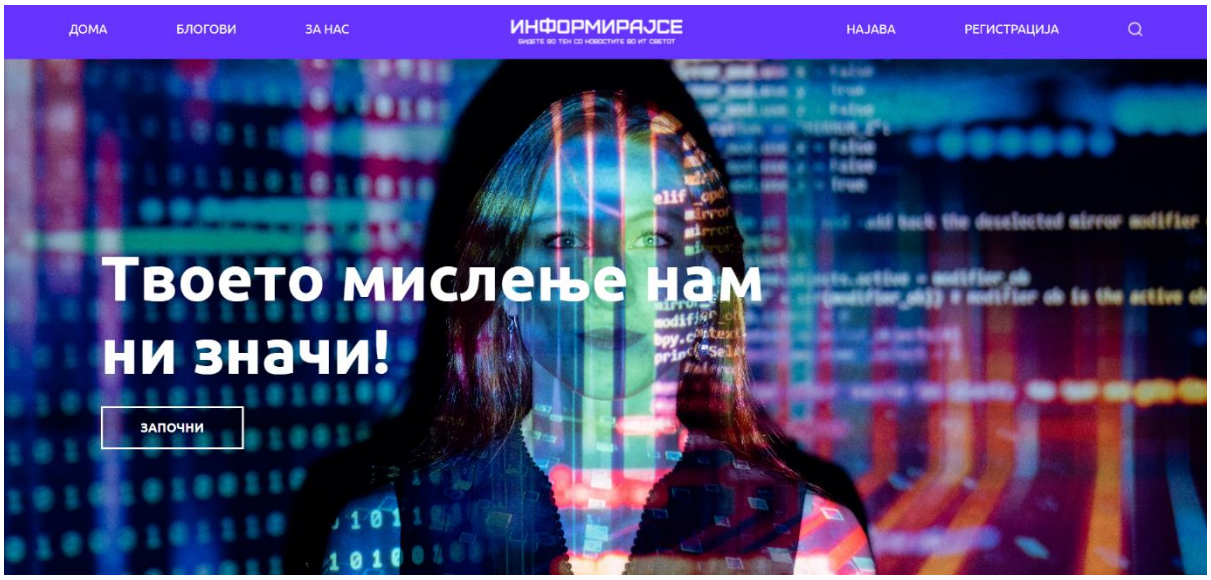
Skopje, June 2024

Contents

Overview

InformirajSe is a dynamic social platform designed to foster engagement and knowledge sharing within the IT community. Targeting enthusiasts, professionals, and scholars, InformirajSe offers a dedicated space for users to post insightful blogs, share the latest news, and discuss developments in the ever-evolving IT world, including cutting-edge advancements in artificial intelligence (AI). Users can interact by commenting on each other's posts, facilitating vibrant discussions and networking opportunities. InformirajSe aims to be the go-to hub for IT and AI enthusiasts to stay informed, exchange ideas, and contribute to a collaborative learning environment.

Goals - Our first goal is to offer space on internet where generally people from our country can gather and talk topics from the IT area that is going to be completely free. We think that our country was in need of such blog space that is approaching modern standards. Secondly we wanted to make a platform where people can interact with people with same interests as them and make one of the most visited also most active Macedonia based social platform.



Getting Started

- **Prerequisites** – No special equipment needed to surf on our platform. It is enough to have internet on your device and you are ready to go.
- **Installation** - Since this is a web based application no additional installation is need. You can search informirajse.mk from your browser and we are a click away from you.
- **Local** - If you want to try the application on your own machine, you can use this guide:
  npm start
  Runs the React App in the development mode.

Open http://localhost:3000 to view it in your browser.
Our backend runs on http://localhost:8080 (that is where we send our api calls).

Detailed Installation Guide

Prerequisites for Development

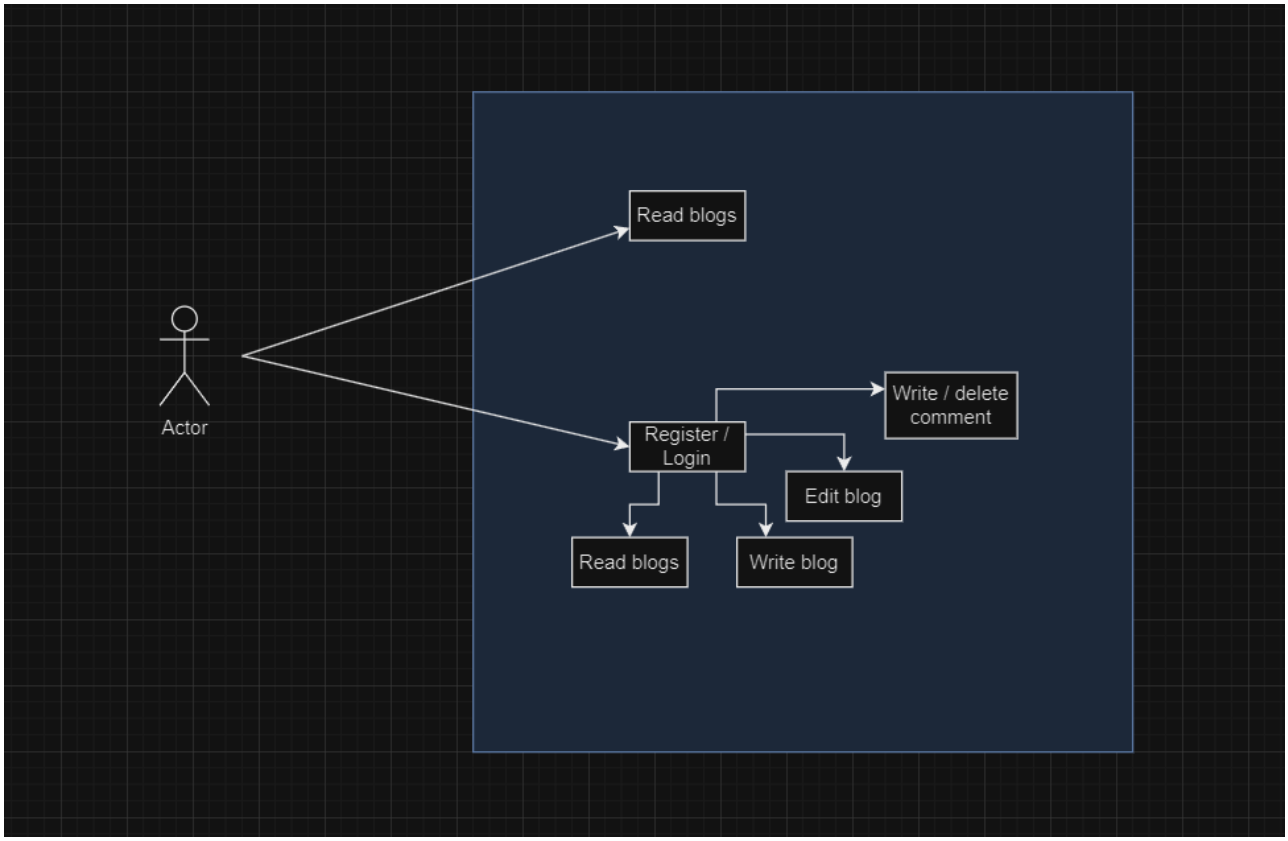
- **Node.js**: Ensure you have the latest version of Node.js installed.
- **.NET Core SDK**: Download and install the .NET Core SDK.
- **PostgreSQL**: Install PostgreSQL and set up your database.

Setting Up the Development Environment

- **Clone the Repository**: Provide the Git command to clone the repository.
- **Environment Variables**: Explain how to set up environment variables for development, including database connection strings.
- **Installing Dependencies**: Detail how to install dependencies for both frontend and backend using npm install and dotnet restore.

Architecture

High-level Design – Once you reach our application the home page will appear first and it is up to you if you want to register and surf in our content. Below there is a diagram where you can see the architecture of the components and flow in our application. We tried to make it as simple as possible in order to make a user friendly system that can be used by different types of users from non-experienced to more experienced ones.



Key Components – In the process of development of our app we use several main components here are they:

- **User Interface** – Our user interface is designed to provide a user-friendly and intuitive experience, accommodating a wide range of devices from 375px wide screens to 1440px large screens. We crafted our design in Figma to ensure a cohesive and aesthetically pleasing look throughout the application. The interface is built with React.js, with all components custom-coded by us, avoiding the use of pre-made components. For styling, we utilized Sass CSS to leverage its flexibility.

- **Backend API** – Initially, we began developing our backend API with Java Spring Boot. However, due to unforeseen issues with CORS configuration, we transitioned to the ASP .NET Core environment in subsequent stages. We

chose ASP .NET Core primarily for its enhanced security features and its compatibility with modern UI frameworks.

- **Database -** PostgreSQL, managed with Entity Framework Core (EF Core), forms the backbone of the main app, ensuring reliable data storage and retrieval.

Codebase

- **Directory structure** – Since we used the ASP .NET Core framework our application is built up to that standards. The Informirajse.UI directory contains the React.js project which is our user interface. In this directory we placed all necessary components and configurations. The components placed in this directory are: assets, enumerations, models, services and views. On the other side in the main project directory we have InformirajSe.Domain, InformirajSe.Repository, InformirajSe.Service and InformirajSe.Web which are common directories in ASP .NET Core framework project.

- **Key files and directories** – one of the key files in our application for the user interface is axios-config directory under the Informirajse.UI directory. This directory contains one file axios.ts which file is used for axios configuration. This configuration file is used to make instance from axios which makes easier to send requests to external api's.

Detailed Features

- **User Profiles -** Users can create and customize their profiles, adding information about their interests and expertise. Profiles also display users' activity, including posts and comments.

- **Blogging Capabilities -** Users can write, edit, and publish blog posts. The editor supports rich text formatting, enabling users to create engaging and informative content.

- **Comment Moderation -** Comments can be flagged for review, and users can report inappropriate content. Moderators can review and take action on reported comments to maintain a respectful community environment.

API Reference

- **Authentication** – Our application is designed in order to the user of the application register themselves in order to share blogs and comments. Since we use ASP .NET Core we use the authentication from the framework. As we mentioned earlier one of the reasons why we choosed ASP .NET Core to make migrate the system was the security provided by this framework. Since it is robust and flexible for implementing authentication allowing us to integrate various authentication schemes.

- **Endpoints** – Since our application has built in more functionalities it responds to many different url's. Here are they and with their description:

    - **/** - This is our application's home page. this is the page that is first visible for our visitors.
    - **/profile** – This endpoint is visible for our authenticated users only. Since this is the endpoint where every user can check his profile with the blogs he posted already the not registered users can't reach this endpoint.
    - **/register** – This is the endpoint where our new users can register in our system in order to make profile themselves and join our community.
    - **/login** – When a user is registered he can use this endpoint to log in next time to get to his profile.
    - **/all-blogs** – This is the endpoint where our users can see all the blogs shared in our community. Also the users can use the search form to filter thru the blogs and search more precisely by keywords, author name, start and end date.
    - **/most-interestng-blogs** – if the users want to see which blogs had most of the traffic in our community and take the attention of our community this is the endpoint they should visit. Here they can find the blogs that have most comments in the last times.
    - **/blog** - By clicking on some of the blog carts that appear on the /all-blogs our users can open the blog that thay read only the title and the description. Here they can find full version of the blog, read it and leave comment if they want.
    - **/edit-blog** – If the user visited /blog endpoint and they are the author of that blog, the edit blog button will be visible and clicking on it will take them on this endpoint where they can edit and submit changes on their blogs.
    - **/news** – If the users want to read something more about the news from the IT world they can visit this endpoint. This endpoint uses external API call that takes news data. But since we are students and couldn't find free news API this endpoint is out of function.

Error Handling

Our API provides clear and consistent error messages to help users understand and resolve issues. Common error codes include 400 (Bad Request), 401 (Unauthorized), and 404 (Not Found).

Additional Resources

As we mentioned earlier we wanted to find free news API and make API call to fetch data in /news endpoint. As normal we should mention that it is not our data its fetched data from another external API source. But because of we couldn't find free news API the endpoint stays empty and we hope we will fix this issue in coming versions of this application.

Security and Privacy

At InformirajSe, we prioritize the security and privacy of our users. Our platform employs robust security measures to safeguard user data and ensure a safe browsing experience. Here are some key aspects of our approach:

Data Protection

We adhere to best practices in data protection, including encryption of sensitive information and secure storage practices. User passwords are hashed using industry-standard algorithms to prevent unauthorized access.

Privacy Policy

Our privacy policy outlines how we collect, use, and protect personal information. We are committed to transparency regarding data handling practices and provide users with control over their personal data.

Compliance

InformirajSe complies with relevant data protection regulations, including GDPR where applicable. We continuously review and update our practices to align with evolving legal requirements and industry standards.

Reporting Security Issues

We encourage responsible disclosure of security vulnerabilities. If you identify a security issue within our platform, please report it to our security team promptly so that we can address it effectively.

User Education

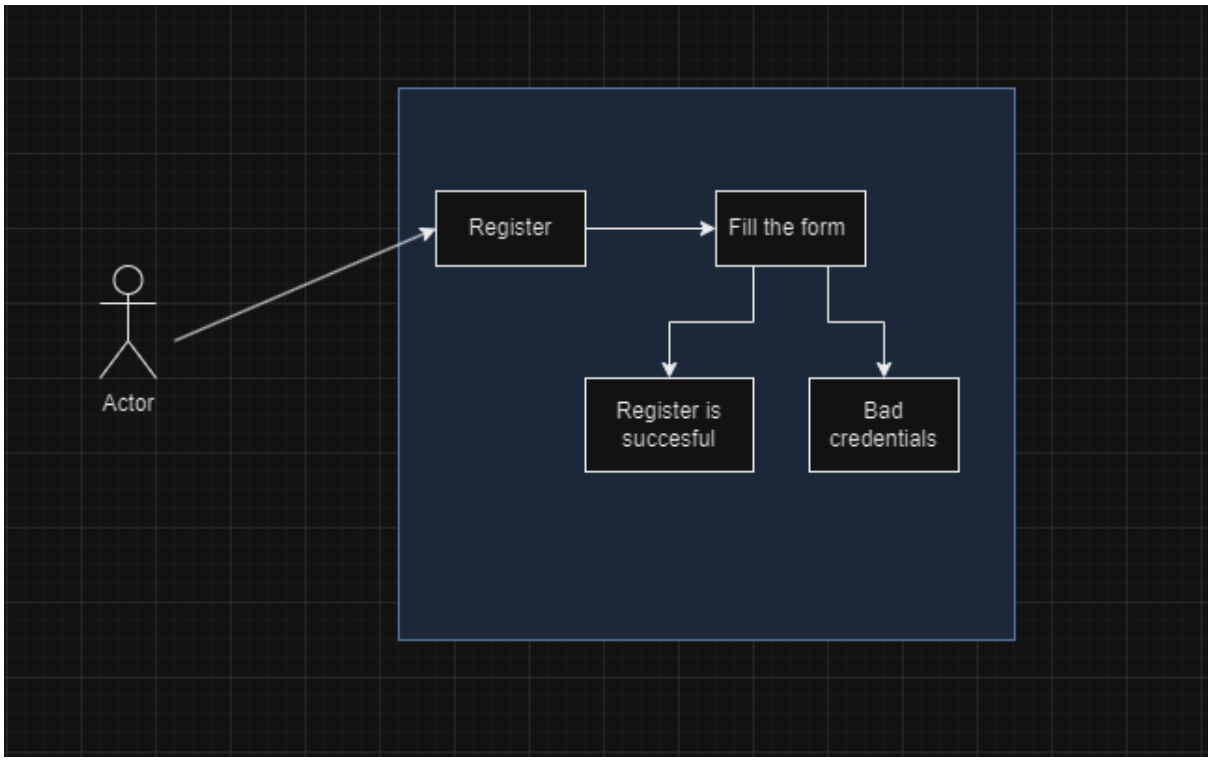
We believe in empowering users with knowledge about online security. Our platform provides resources and tips to help users protect their accounts and personal information.

Continuous Improvement

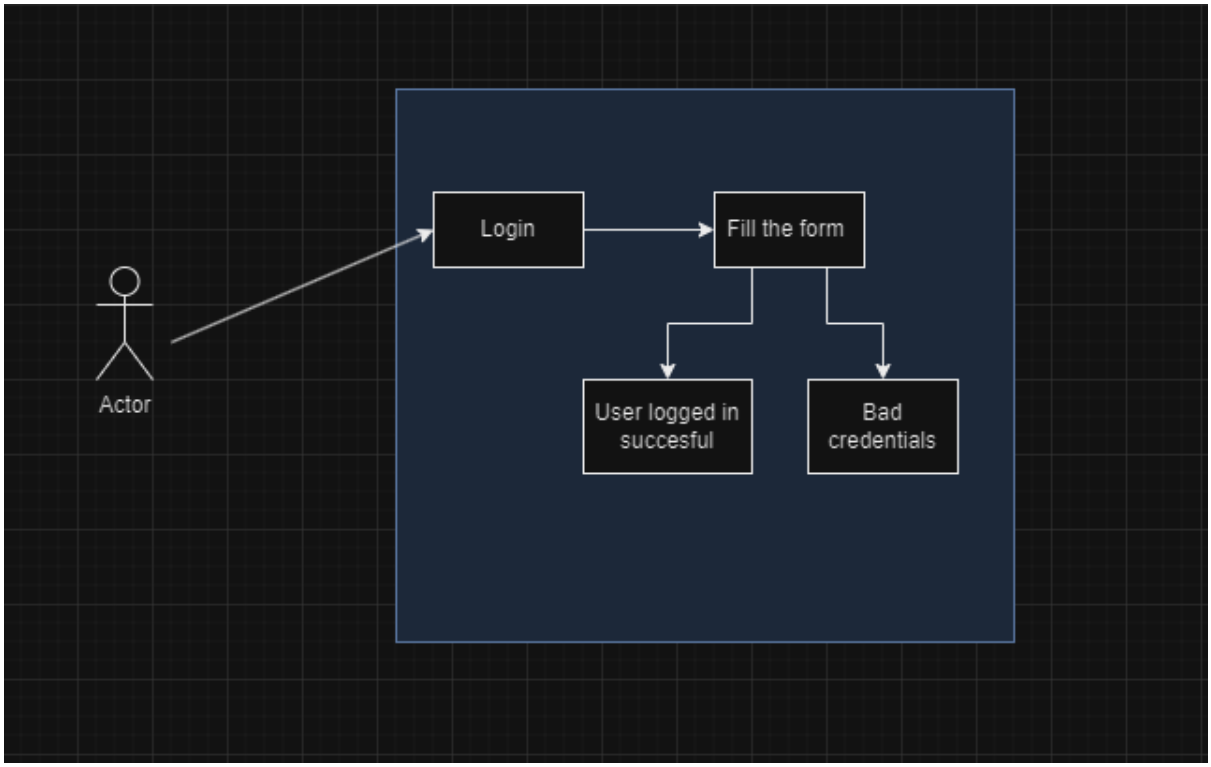
Security is an ongoing priority at InformirajSe. We conduct regular security assessments, implement updates promptly, and engage in proactive measures to mitigate risks.

Use Cases

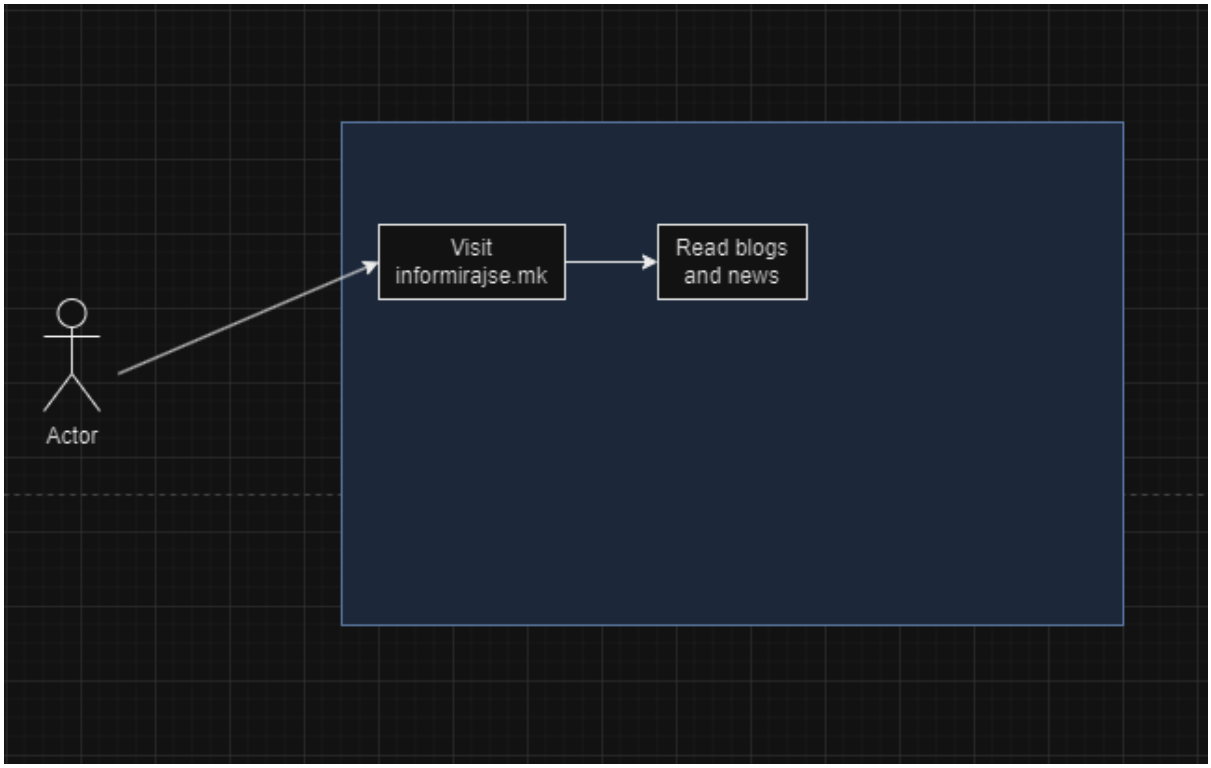
Register

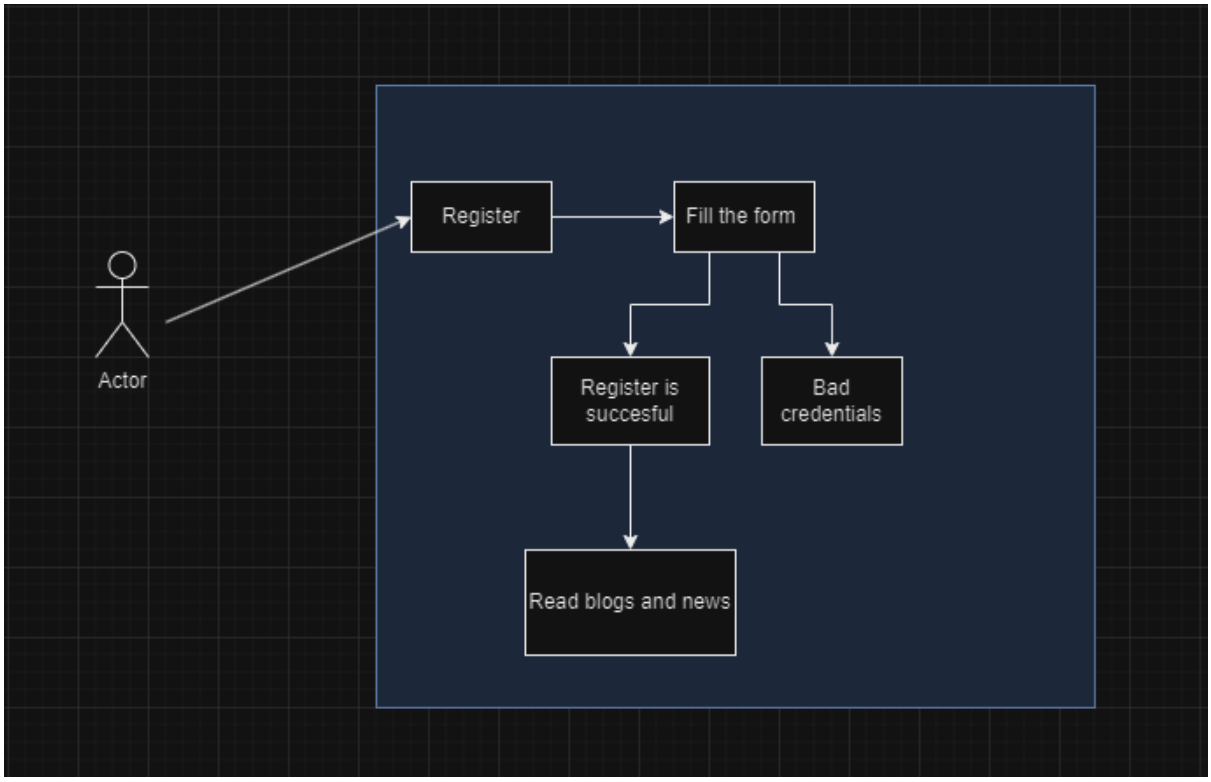


Login

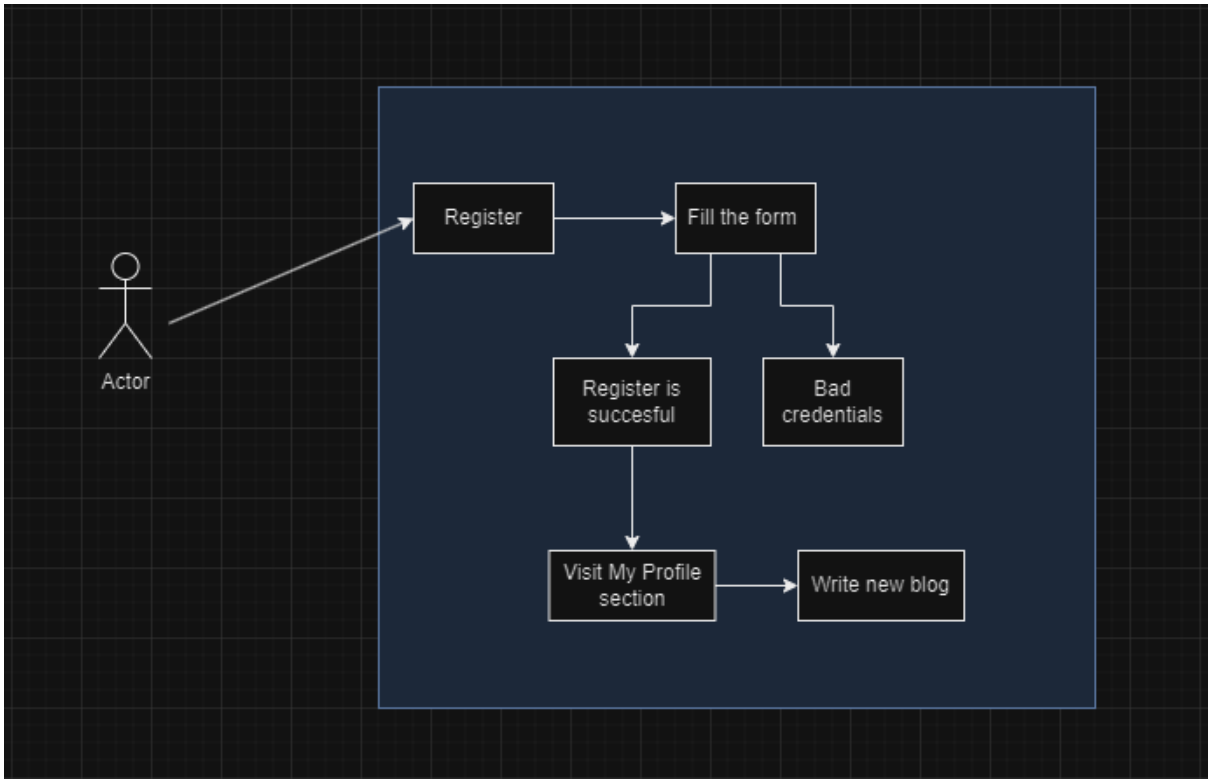


Read Blogs - this scenario can be performed both by authenticated and not authenticated users so it can be visualized in two ways.

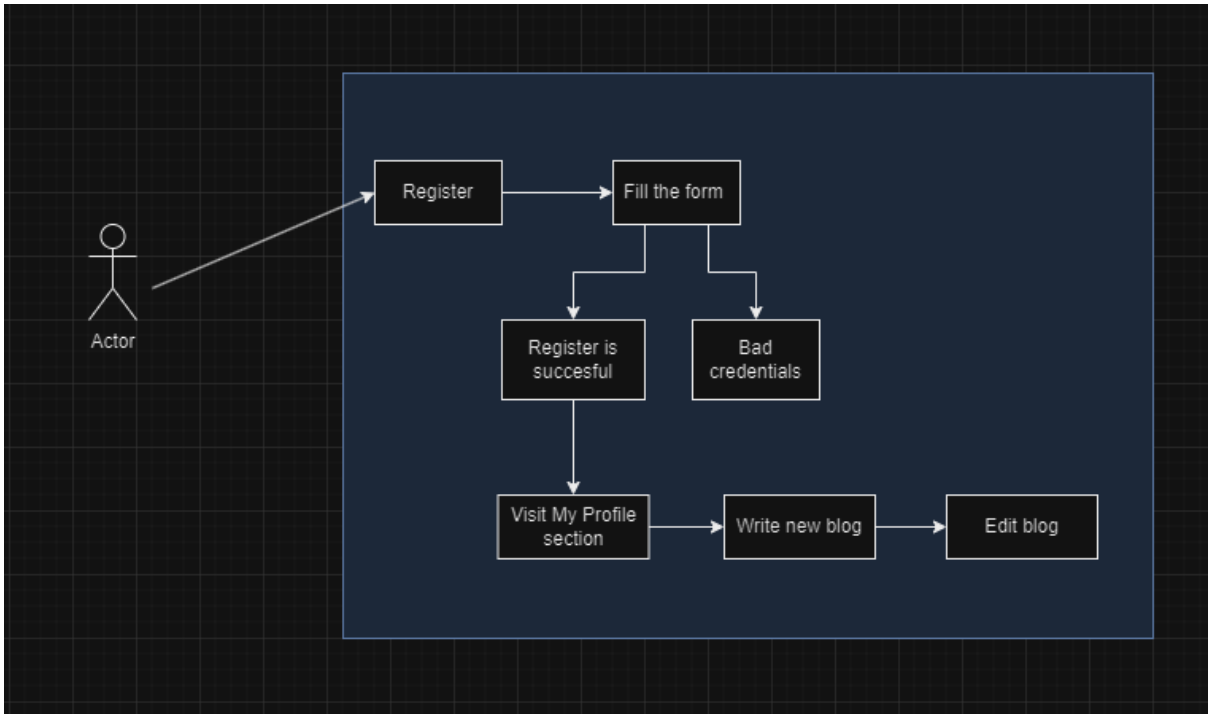



Register
Fill the form
Actor
Register is succesful
Bad credentials
Read blogs and news

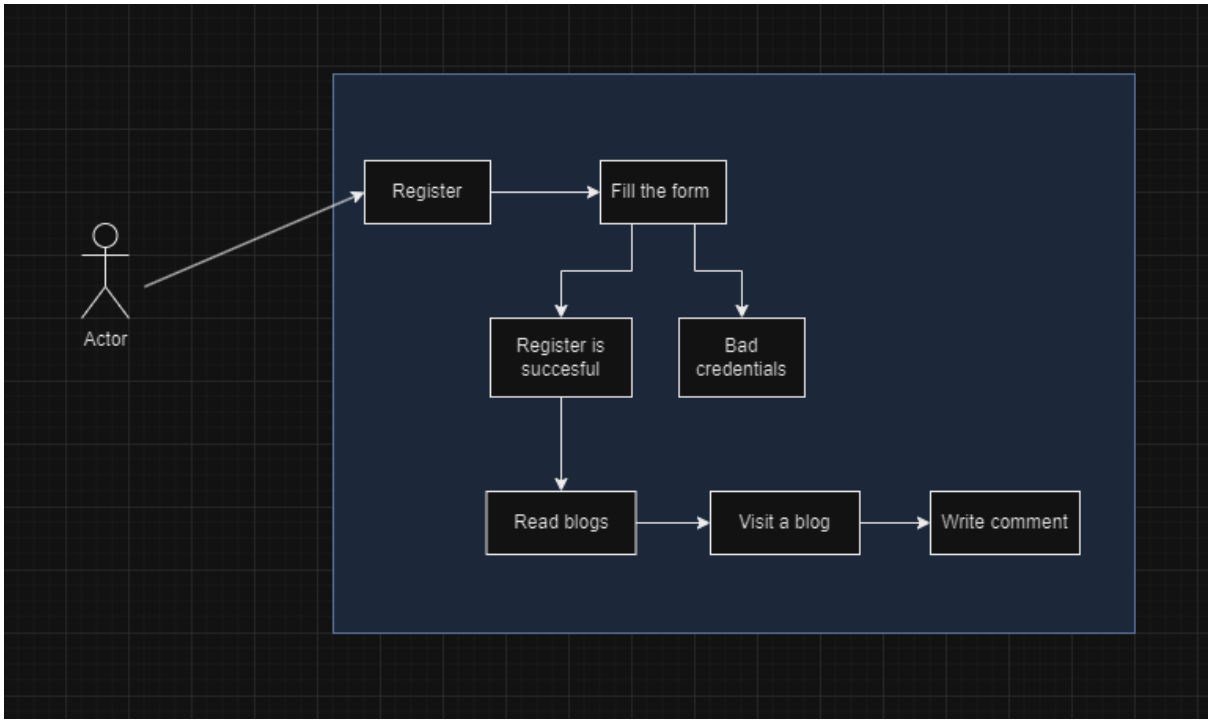
Add new blog


Register
Fill the form
Actor
Register is succesful
Bad credentials
Visit My Profile section
Write new blog

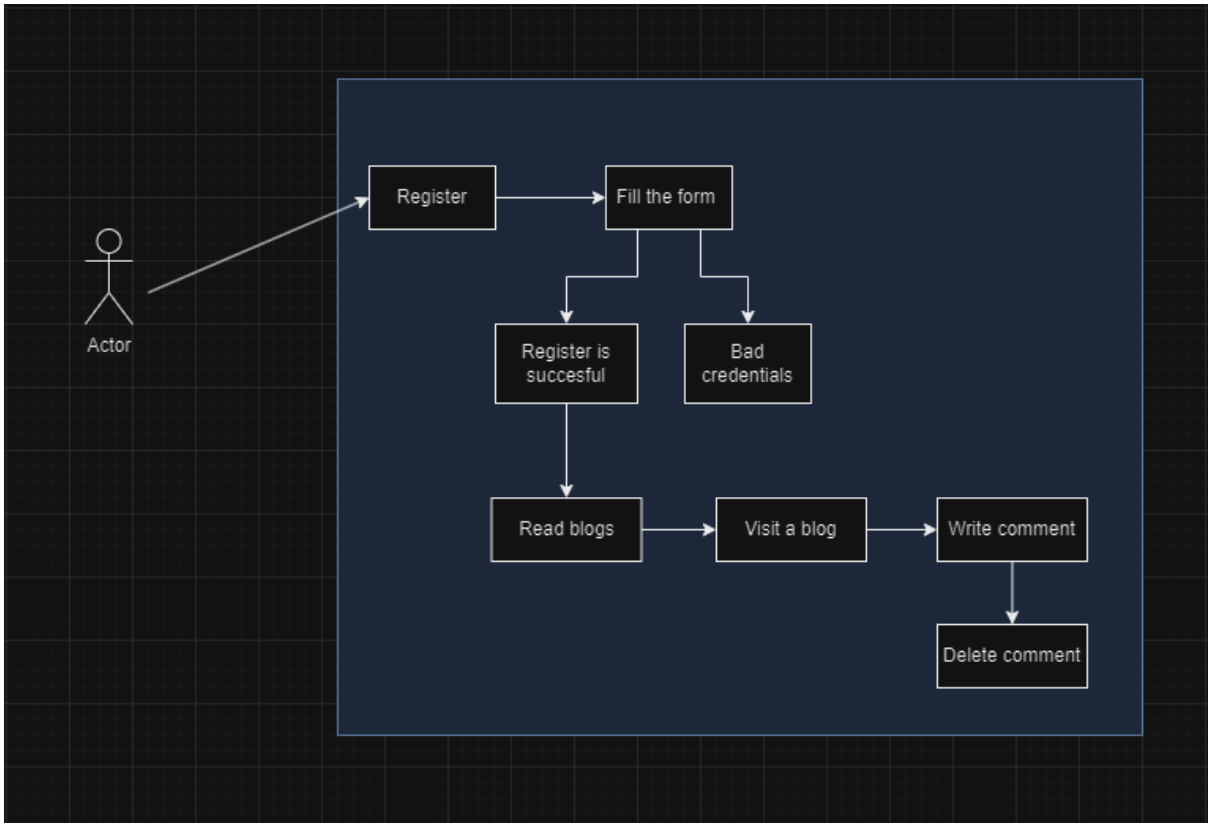
Edit blog



Write comment



Delete comment



Future Roadmap

Planned Features

- **Mobile App**: Develop a native mobile app to provide a seamless experience on smartphones and tablets.

- **Advanced Search**: Implement advanced search features, including filters and sorting options, to help users find content more easily.

- **Community Events**: Introduce virtual events, such as webinars and Q&A sessions, to foster community engagement.

Community Contributions

We welcome contributions from the community. Users can submit feature requests, report bugs, and contribute code to help improve the platform.

Community and Support

Getting Support

Users can get support through our community forum, help desk, or by accessing our comprehensive documentation. We also offer live chat support during business hours.

Contributing to the Project

We encourage users to contribute to the project. Whether it's through code contributions, bug reports, or feature requests, every contribution helps make InformirajSe better.

Conclusion

In conclusion, InformirajSe emerges as a pivotal platform within the Macedonian IT community, designed to foster engagement, collaboration, and knowledge sharing. By combining robust backend technologies like ASP .NET Core and PostgreSQL with a sleek, user-centric React.js frontend, InformirajSe offers a seamless user experience. The platform caters not only to IT enthusiasts but also to professionals and scholars seeking a vibrant space to discuss cutting-edge topics such as artificial intelligence.

InformirajSe's commitment to accessibility is evident through its web-based architecture, ensuring easy access for all users with minimal prerequisites beyond internet connectivity. The platform's intuitive design, crafted in Figma and implemented with custom-coded components, enhances usability across a wide range of devices, from smartphones to large screens.

Key features such as user profiles, blogging capabilities, and comment moderation empower users to interact meaningfully while maintaining a respectful community environment. The platform's API-driven structure supports various endpoints, facilitating actions from blog creation to user authentication seamlessly.

Looking forward, InformirajSe's roadmap includes ambitious plans for a native mobile app, advanced search functionalities, and community events, underscoring its commitment to continuous improvement and community-driven development. As the platform evolves, contributions from its users—whether through feedback, bug reports, or code submissions—remain integral to shaping its future.

In summary, InformirajSe not only meets but exceeds the modern standards expected of a dynamic IT community platform. It stands poised as the go-to hub for Macedonian IT enthusiasts, offering a comprehensive resource for staying informed, sharing insights, and participating in a thriving digital community.