

Florida International University
Knight Foundation School of Computing and Information Sciences

Software Engineering Focus

Final Deliverable

Community Connect
Spring 2023

Team Members: Castor Velasquez, Fabiana Fernandez Licon, Jordan Rainey, Rafael Rubio Rodriguez, Allan Prieb

Product Owner(s): Shahin Vassigh

Mentor(s): Biayna Bogosian

Instructor: Masoud Sadjadi

The MIT License (MIT)
Copyright (c) 2016 *Florida International University*

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Abstract

This document presents the information necessary to gain a good understanding of the current iteration of the Community Connect website.

In this iteration of the Community Connect website project, our main focus was to fully implement the news page and forums page. We aimed to add new functionalities to both of these pages. The news page was designed to allow users to view articles and updates related to their community. We added features such as filtering news articles by category.

Similarly, on the forums page, we added new features that allowed users to create posts and comment on other users' posts. Overall, we aimed to create a more interactive and engaging platform for users to connect with their community and share their thoughts and ideas about the problems they face on a day-to-day basis.

Table of Contents

Introduction	7
Current System	7
Purpose of New System	11
User Stories	12
Implemented User Stories	12
Pending User Stories	13
Project Plan	14
Hardware and Software Resources	14
Sprints Plan	15
Sprint 1	15
Sprint 2	16
Sprint 3	17
Sprint 4	18
Sprint 5	19
Sprint 6	20
System Design	21
Architectural Patterns	21
System and Subsystem Decomposition	21
<i>Front-end architecture tools</i>	22
<i>Back-end architecture tools</i>	22
<i>Database</i>	23
<i>Deployment Diagram</i>	23
Design Patterns	24
System Validation	24
Glossary	25
Appendix	26
Appendix A - UML Diagrams	26
1. Class Diagram	26

Appendix B - User Interface Design	38
Appendix C - Sprint Review Reports	46
Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents	51
Community Connect Prototype (skip all docker commands for now)	51
Required Services	51
Structure of the project	51
Running Project Locally	51
1. Clone the repository	52
2. Navigate to the project's root directory (i.e. where package.json, docker-compose.yml, and .dockerignore files are located)	52
3. Install node dependencies for the Frontend	52
Windows Backend Installation Instructions	52
4. Create a virtual environment for the backend.	52
5. Activate the virtual environment.	52
6. Install the dependencies in the environment.	52
7. Migrate the database models.	52
Mac/Linux Backend Installation Instructions	53
4. Create a virtual environment for the backend.	53
5. Activate the virtual environment.	53
6. Install the dependencies in the environment.	53
7. Migrate the database models.	53
Running the project locally (from the root directory, no environment)	53
Running the frontend	53
Running the backend	53
Windows	53
Mac	53
Deployment through Ocelot using docker	54
4. Build the docker image (skip for now)	54

5. <i>Make and migrate database models</i>	54
6. <i>Create an admin user for the Backend (skip for now)</i>	54
7. <i>Change your Windows settings/MacOs settings in your host file (skip for now)</i>	54
8. <i>Run the project (skip docker commands)</i>	55
9. <i>Shutting down the project</i>	56
Setting up the Backend Development Environment (Outdated)	56
1. <i>Install pipenv</i>	56
2. <i>Enter the shell environment</i>	56
3. <i>Installing all required packages</i>	56
References	57

INTRODUCTION

The purpose of this capstone project is to continue the development of the Community Connect website, with the goal of creating an interactive platform that allows users to connect with their community and stay informed about urgent issues. This semester we aim to make the website even more collaborative by leveraging the different functionalities of the forums page. By enabling users to create posts and comment on others' posts, we intended to establish a more engaging platform that encourages cooperation and community creation.

Current System

First iteration

The first iteration of the Community Connect Website was completed by the Capstone I team of Gabriel Perez, Samara Ruiz-Sandoval, Mark Guadalupe, Nathanael Garcia, Eugene Theodorovych. The focus of the work completed prior was the custom user interface specifically with the addition of front-end components to the base forum, user experience, as well as deployment of the system to both development and production servers. This involved detailed research of the project packages that would fit the desired stack. Research on Docker, Django, and Node.js was done during the development of the previous system as well as research of the complex Misago package structure, and proper troubleshooting techniques. The search bar was implemented with elastic search to yield results for basic profile information of users that are registered in the forum, keywords mentioned in threads, and a link to provide a full page of results found across both threads and users. A theme was previously developed using Adobe XD by Sophia Cabral to assist in the specification of colors, fonts, and styles for the pages within the discussion forum. Graphics were also implemented from the mock-up designs to enforce the purpose of the website that incorporates news relevant to the user, trending content from their location, and connecting to their community. Finally, custom card components were added for future development of the new features.

Second iteration

The second iteration of the Community Connect Website was completed by the Capstone II team of Gabriel Perez, Samara Ruiz-Sandoval, Mark Guadalupe, Nathanael Garcia, Eugene Theodorovych. The purpose of the new system was focused on designing new features that would make the system more attractive and intuitive for the user experience. The new system would have to serve as the platform for discussions on health-related issues and possible

solutions for those, as well as the point of connection between different vendors and the customers. As the second iteration of the existing project, the development involved the creation of the new features that included cards in the dashboard that redirect to news pages for Guides, Trends, and News, a colored theme for the main and splash pages and improved December 2, 2021 Page 7 of 82 Final Deliverable Mitigating the impacts of COVID19 searching across the forum. Research and connection of different APIs that tightened all processes in the back end resulted in proper visualization, which allowed the content to be rendered in the needed way, eventually simplifying the user experience and fulfilling the needs of the project. Bing News and Google Trends API call, after certain data cleaning, allowed data population to correspond to pages. Cron jobs were scheduled on the servers to update the JSON files needed for the Trends page. IPAPI call was connected to get the user location and display relevant data on both News and Trends pages. An Embedded Google map was rendered to the Trends page to allow users to see the nearest vaccination sites.

Third iteration

The third iteration of the Community Connect Website was completed by the Capstone II and Senior Project team composed of Gretel Gomez Rodriguez, Juan Grajales, Dayana Garcia, Carolina Rodriguez, and Norlan A. Castillo. The team began a new system which recreated the previous semester's community connect project without the use of Misago. The purpose of the new system was to redesign the web application's architecture in order to decouple the frontend and backend to ensure a better and more organized structure for the project files. The new system has the same intended purpose to serve as the platform for discussions on health-related issues and possible solutions for those, as well as the point of connection between different vendors and the customers. The team was able to recreate the frontend using React and created the foundation for the backend using Django and Docker. A basic news API was added towards the end of the last sprint.

Fourth iteration

The fourth iteration of the Community Connect Website was completed by the Capstone II and Senior Project team composed of Adrian Gonzalez, Niurbelis Gonzalez, Sebastian Gomez, Shaoting Wu, Chábeli Castano Arango, Narmeen Kibria, Manuel Kamboykos, and David Garcia. The purpose of this iteration is to continue the work of the team that completed the previous iteration. The team began with a complete overhaul of the design of the site, redesigning every page to the new UI/UX spec. The entire project was also restructured for a more streamlined developer experience when it comes to project installation, building, and running. Docker images were also refactored for better system efficiency and a reduction of the image size, improving

build times. New systems and technologies were installed, such as Next.js, to utilize the built-in features that come with the new framework, such as server-side rendering, dynamic routing, and static site generation. The backend was also redone, removing unnecessary and unused endpoints. Towards the end of the last sprint, the foundation for a Carbon Footprint Calculator was implemented, but still needs to be connected to an API, such as Carbon Interface, and requires interactive graphs to be implemented with Chart.js.

Fifth iteration

The fifth iteration of the Community Connect Website was completed by the Capstone II and Senior Project team composed of Francisco Mazzei, Leinier Perez, Yelise Sanchez, Brock Williams and Kevin Escoto. The original purpose for the website was to be a platform for discussions on health-related issues and possible solutions for those, as well as the point of connection between different vendors and the customers. While this is still an intended purpose of the website, the latest iteration has expanded its purpose to provide an additional focus on the environment and sustainability. With this renewed purpose in mind, the goal of this iteration was to continue the work of the past team in building on the website design and functionality. This work consisted of adding new pages to the site, completing several design modifications from the original UX/UI specifications given to the team at the start, as well as completing the carbon calculator and news feed components. The carbon calculator on the Carbon Footprint page was successfully connected to the Carbon Interface API, effectively using real world data to calculate an estimate of a user's carbon footprint based on their inputs in various fields. The news feed was also successfully connected to the Google Scholar API as well as the News API on the News page, so users may filter between reliable sources of scholarly articles or news articles about their selected topics of interest. While the News page is functional in development it is not functional after deployment because the News API key is only free for development purposes. Using the News API key in deployment requires a subscription payment which the Product Owner has decided to postpone. The Forums page is not yet functional as more work is needed for posting discussions and replying to ongoing threads created by other community members. No changes have been made to the technologies utilized in the prior iteration of the project, and the updated website has been deployed for public use through FIU's server.

Sixth iteration

The sixth iteration of the Community Connect Website was completed by the Capstone II and Senior Project team composed of Hunter Badillo, Zackary Santana, Natasha Rodriguez, Eduardo Lopez, Michael Olivero, Miguel Ferreiro, and Raul Gutierrez Niubo. The purpose of this

platform is to connect people in a particular community who may be facing some natural or man-made disaster of any kind. While the previous rendition focused on allowing users to gauge their environmental impact, this iteration focuses on refining the places where most users would be interacting with each other; the News and Forum page. The News page is a place for users to find out what is happening in their community. This page is supposed to display articles based on the user's chosen topics of interest. These are predetermined topics that have to do with environmental / man-made events or disasters. The forum page is supposed to show posts from users who are in the same general area as the user who is viewing them. For example, a user who is in a place prone to hurricanes should be able to see other users posting content about their hurricane tips and experiences. This gives users who are in the same community the opportunity to collaborate and possibly receive help from one another.

Purpose of New System

Seventh iteration

The seventh iteration of the Community Connect Website was Completed by the Capstone II team composed of Castor Velasquez, Fabiana Fernandez Licon, Jordan Rainey, Rafael Rubio Rodriguez, and Allan Prieb. In this iteration of the Community Connect website project, the main focus was to fully implement the news page and forums page. We added new functionalities to both of these pages. The news page was designed to allow users to view articles and updates related to their community. We've added features like filtering news by category, and while the current system only shows news from the Miami community, we've laid the groundwork so we can show news from other communities in the future. Similarly, on the forums page, we added new features that allowed users to create posts and comment on other users' posts. Overall, we aimed to create a more interactive and engaging platform for users to connect with their community and share their thoughts and ideas about the problems they face on a day-to-day basis.

USER STORIES

The following section provides the detailed user stories that were implemented in this iteration of the Community Connect project. These user stories served as the basis for the implementation of the project's features in this iteration. This section also shows the user stories that are to be considered for future development.

Implemented User Stories

- S01D – As a developer, I should have all the dependencies installed and the development environment set up (2 pts)
- S01 - As a user, I can navigate to the news page and view news relevant to the topic I select (2 pts)
- S02 - As a user, I can view the latest news in the news page based on my current location (4 pts)
- S03 - As a user, I can select multiple topics in the news page and see the latest news about the different topics (5 pts)
- S04 – As a user, I should be able to read a disclaimer for the default community (5 points)
- S05 – As a user, I should be able to see updated news (5 points)
- S06 – As a user, I should be able to view news from a specific community (5 points)
- S07 – As a user, I should be able to only see news based on the topics I select (5 points)
- S08 – As a user, I should be able to clearly know where to post a discussion (5 points)
- S09 - As a user, I should be able to enter a title and select a topic from a drop-down menu (5 points)
- S10 - As a user, I should view Posts filtered by selected topics (5 points)
- S11 – As a user, I should be able to see tags related to topics in posts (5 points)
- S12 – As a user, I will be visually pleasant when seeing the start a Discussion Card (5 points)
- S13 – As a user, I should be able to add name to Posts and Replies (5 points)
- S14 – As a user, I should be able to use the website in my mobile (5 points)
- S15 - As a user, I should read “Topics of Interest” on the News instead of “Select Topics of Interest” (5 points)

Pending User Stories

- S02 - As a user, I can view the latest news in the news page based on my current location (4 pts)
- S07 – As a user, I should be able to view news from a specific community (5 points)

Disclaimer:

The user stories listed above were not implemented due to the limit of free calls of the API we used, as decided by the product owner.

PROJECT PLAN

This section describes the planning that went into the realization of this project. This project incorporated agile development techniques and as such required the sprints to be planned. These sprint plannings are detailed in the section. This section also describes the components, both software and hardware, chosen for this project.

Hardware and Software Resources

The following is a list of all hardware and software resources that were used in this project:

- **Docker:** A free and open-source platform of OS-level virtualization to host the needed images(parts) of the project.
- **Docker desktop application** - optional visualization tool for Docker images
- **GitHub:** A web-based version-control and collaboration platform for software developers.
- **GitHub desktop:** Optional support application that enables Git collaboration.
- **Visual Studio Code:** a source-code editor developed by Microsoft for Windows
- **Bootstrap:** An open-source CSS framework that is also used for quickly and efficiently creating the designs and layouts for web pages.
- **Django:** A Python free-open source framework that manages the back-end of the entire application and custom apps added to Misago core.
- **React:** An open-source, front-end, JavaScript library for building user interfaces or UI components that serve open-source as the engine for the front-end of the application.
- **Next.js:** An open-source development framework that is built on top of Node.js a React that provides additional developer functionality and support for features such as server-side rendering, static site generation, built-in routing and more.
- **PostgreSQL:** A free, open-source database management system that has a separate Docker container and servers as the database for the application.

Sprints Plan

Sprint 1

Sprint Planning Meeting Minutes:

Attendees: Castor Velasquez, Fabiana Fernandez, Jordan Rainey, Allan Prieb, Rafael Rubio

Start time: 1:00 PM

End time: 1:30pm

After discussion, the velocity of the team was estimated to be 10 hours.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

- S01D – As a developer, I should have all the dependencies installed and the development environment set up (2 pts)
- S01 - As a user, I can navigate to the news page and view news relevant to the topic I select (2 pts)
- S02 - As a user, I can view the latest news in the news page based on my current location (4 pts)
- S03 - As a user, I can select multiple topics in the news page and see the latest news about the different topics (5 pts)

The team members indicated their willingness to work on the following user stories.

- All team members:
 - S01D
 - S01
- Castor Velasquez, Fabiana Fernandez, Jordan Rainey:
 - S03
- Allan Prieb, Rafael Rubio:
 - S02

Sprint 2**Sprint Planning Meeting Minutes:**

Attendees: Castor Velasquez, Fabiana Fernandez, Jordan Rainey, Allan Prieb, Rafael Rubio

Start time: 5:40 PM

End time: 6:00 PM

After discussion, the velocity of the team was estimated to be 10 points.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

- S02 - As a user, I can view the latest news in the news page based on my current location (10 pts)

The team members indicated their willingness to work on the following user stories.

- All team members:
 - S02

Sprint 3

Sprint Planning Meeting Minutes:

Attendees: Castor Velasquez, Fabiana Fernandez, Allan Prieb, Rafael Rubio, Jordan Rainey

Start time: 5:45 PM

End time: 6:15 PM

After discussion, the velocity of the team was estimated to be 32.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

- S05 - Add disclaimer for default community (5 points)
- S06 – Create a scheduled job to fetch news from the API and store them in Postgres (5 points)
- S07 – Update view to get all news from a specific community (5 points)
- S08 – Add call to our backend when user visits the News page (5 points)
- S09 – Show only topics selected by the user (5 points)
- S10 – Add unit tests to the news endpoint (5 points)

The team members indicated their willingness to work on the following user stories.

- Castor Velasquez
 - S06, S10
- Allan Prieb
 - S07
- Rafael Rubio
 - S05
- Fabiana Fernandez
 - S09
- Jordan Rainey
 - S08

Sprint 4

Sprint Planning Meeting Minutes:

Attendees: Castor Velasquez, Fabiana Fernandez, Allan Prieb, Rafael Rubio, Jordan Rainey

Start time: 5:45 PM

End time: 6:15 PM

After discussion, the velocity of the team was estimated to be 30.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

- S11 – UI/UX Improvements (5 points)
- S12 - Style the Forums page (5 points)
- S13 - Style the Post page (5 points)
- S14 - Create and Modify Post Views (5 points)
- S15 – Add Unit Tests to the Post App (3 points)
- S16 – Connect Frontend and Backend (5 points)
-

The team members indicated their willingness to work on the following user stories.

- Castor Velasquez
 - S15
 - S14
- Allan Prieb
 - S16
- Rafael Rubio
 - S11
- Fabiana Fernandez
 - S13
- Jordan Rainey
 - S12

Sprint 5**Sprint Planning Meeting Minutes:**

Attendees: Castor Velasquez, Fabiana Fernandez, Allan Prieb, Rafael Rubio, Jordan Rainey

Start time: 5:45 PM

End time: 6:15 PM

After discussion, the velocity of the team was estimated to be 30 points.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

- S17 – Display tags in posts (5 points)
- S18 - Change order of Cards (5 points)
- S19 - Improve the Start a Discussion Card (5 points)
- S20 - Add name to Posts and Replies (5 points)
- S21 - Make Forums page mobile friendly (5 points)

The team members indicated their willingness to work on the following user stories.

- Castor Velasquez
 - S17
- Fabiana Fernandez
 - S18
- Allan Prieb
 - S21
- Rafael Rubio
 - S20
- Jordan Rainey
 - S19

Sprint 6

Sprint Planning Meeting Minutes:

Attendees: Castor Velasquez, Fabiana Fernandez, Allan Prieb, Rafael Rubio, Jordan Rainey

Start time: 5:45 PM

End time: 6:15 PM

After discussion, the velocity of the team was estimated to be 30 points.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

- S22 – Enhance UX on Forums page by changing “Topics of Interest” to “Select Discussion of Interest” (5 points)
- S23 – Enhance UX on forums page by making the relationship between topics card and discussions card more visually apparent. (5 points)
- S24 – Fix the width spacing on the forums page between the 3 cards. (5 points)
- S25 – Extend the bottom of the post card to be level with the other two cards.(5 points)
- S26 - Change “Topics of Interest” on the News page to “Select Topics of Interest” (5 points)

The team members indicated their willingness to work on the following user stories.

- Castor Velasquez
 - S23
- Fabiana Fernandez
 - S22
- Allan Prieb
 - S24
- Rafael Rubio
 - S25
- Jordan Rainey
 - S26

SYSTEM DESIGN

This section contains information on the design decisions that went into this project. The architecture patterns are outlined and explained. The entire system is shown in a package diagram and the subsystems are explained. Finally, the design patterns used in the project are discussed.

Architectural Patterns

The current architecture for Community Connect utilizes React for the client and Django for the server while still using Docker. This is a full-stack application that utilizes React Javascript framework as the main frontend component, whereas Django Python framework manages backend processes. Additionally, the main database is managed by PostgreSQL in order to manage the cache of the application. The project has 2 versions for development and deployment accordingly. When the local-development version consists of parts described above, the deployed version additionally consists of an Nginx proxy and SSL certificate handler as well utilizing Docker. Each part of the application has a separate Docker image and the application as the whole is composed in one big Docker container before being deployed to the server. The backend runs completely separate from the frontend, so if for some reason the backend is broken, it does not affect the way the user sees the application and its components.

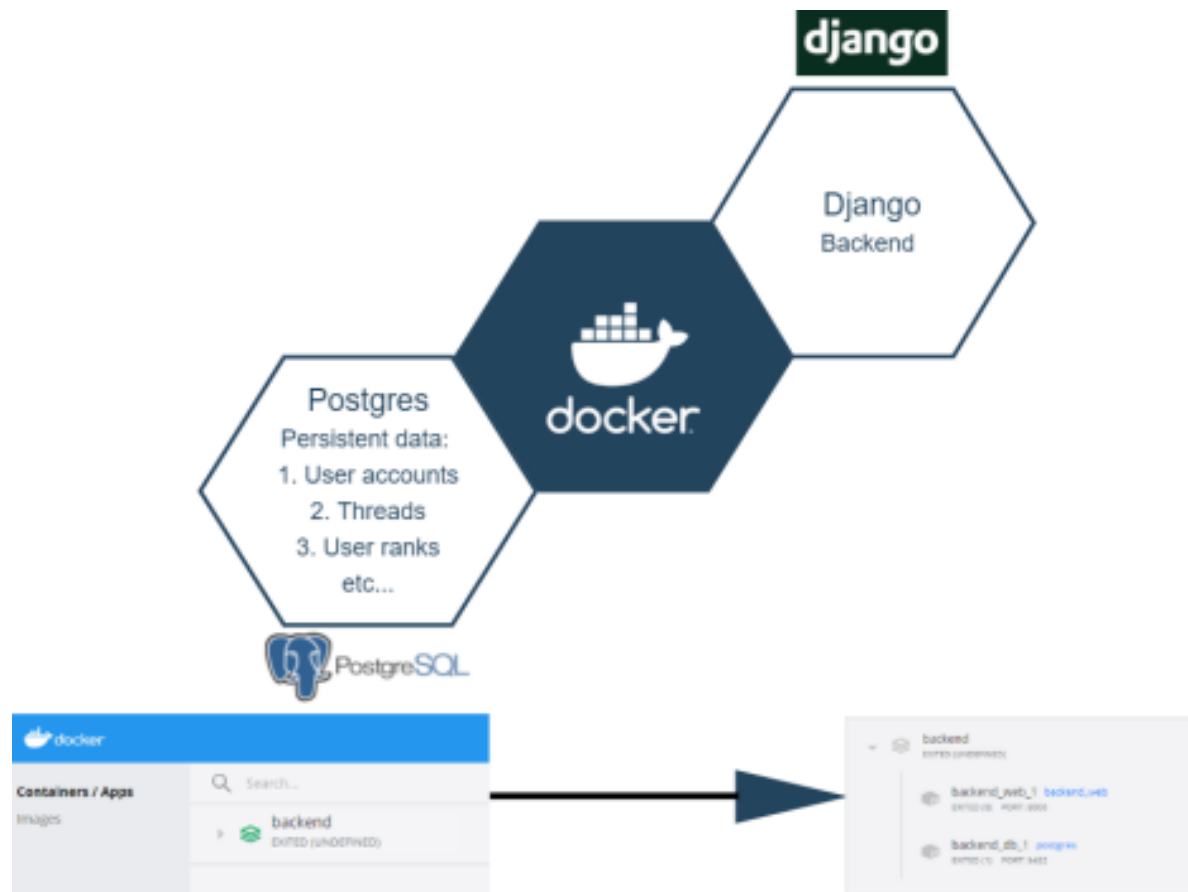
System and Subsystem Decomposition

The current architecture was decided by the previous team and was created with the following benefits in mind: decouple frontend, backend, and services, facilitate new developer onboarding, and scope CSS to each component. The front-end components are handled by React using hooks. We are using Sass for styling and most components have their own Sass file for styling. There are also several global style files included to facilitate changes in the future such as color theme, button, and fonts. The backend uses Django and Postgres.

Front-end architecture tools

This project is built off of the model, view, controller architecture to separate each of the project's components into easily understandable parts. The model is our database and data structure, the view is the user interface, and the controller accepts and converts the input.

Back-end architecture tools

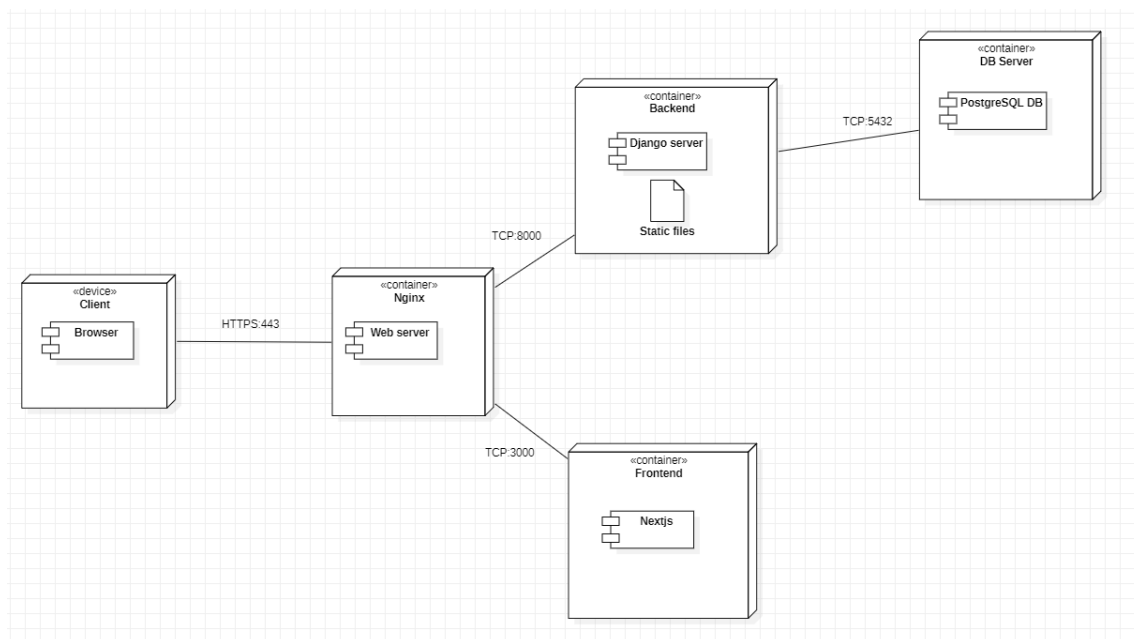


Database

The current database structure was decided upon by the third iteration team but will have to be redesigned without the use of Misago as the architecture of the project no longer uses Misago. Since our priorities were focused on improving the website design as well as implementing the News and Forums page functionalities, the database will be redesigned in upcoming semesters.

Deployment Diagram

Different developers commit changes, and fetch updates to/from the shared GitHub repository. All component changes are merged and then pulled to the deployment server hosted on an Ubuntu OS where the Docker container resides and is restarted after updates. The project is then served to the Community Connect domain using a Nginx web server where it can be seen by other users.



Design Patterns

This project uses the Model-View-Controller design pattern. By adopting the MVC pattern, we were able to achieve a more organized, scalable, and maintainable codebase. The separation of concerns allowed us to make changes to one component of the application without affecting the other components, making it easier to modify, update, and extend the application over time. Additionally, the MVC pattern provided a clear separation of responsibilities, enabling us to assign specific tasks to each team member more efficiently.

SYSTEM VALIDATION

During this iteration, our team conducted fundamental testing to ensure that the UI/UX of the website is consistent and functional across various browsers, operating systems, and screen sizes. Our primary focus was on the appearance and functionality of the website, ensuring that it is visually appealing, easy to navigate, and user-friendly, regardless of the device or platform being used. Our goal was to provide a seamless experience for users, allowing them to access the information and functionalities of the website with ease, regardless of their device or browser preferences. By using the command `./dev test`, the project will run the pytest suite which will provide expressive and readable unit tests for each of the site's applications. These unit tests have not yet been implemented; however, this will be worked on as the project nears completion in future iterations.

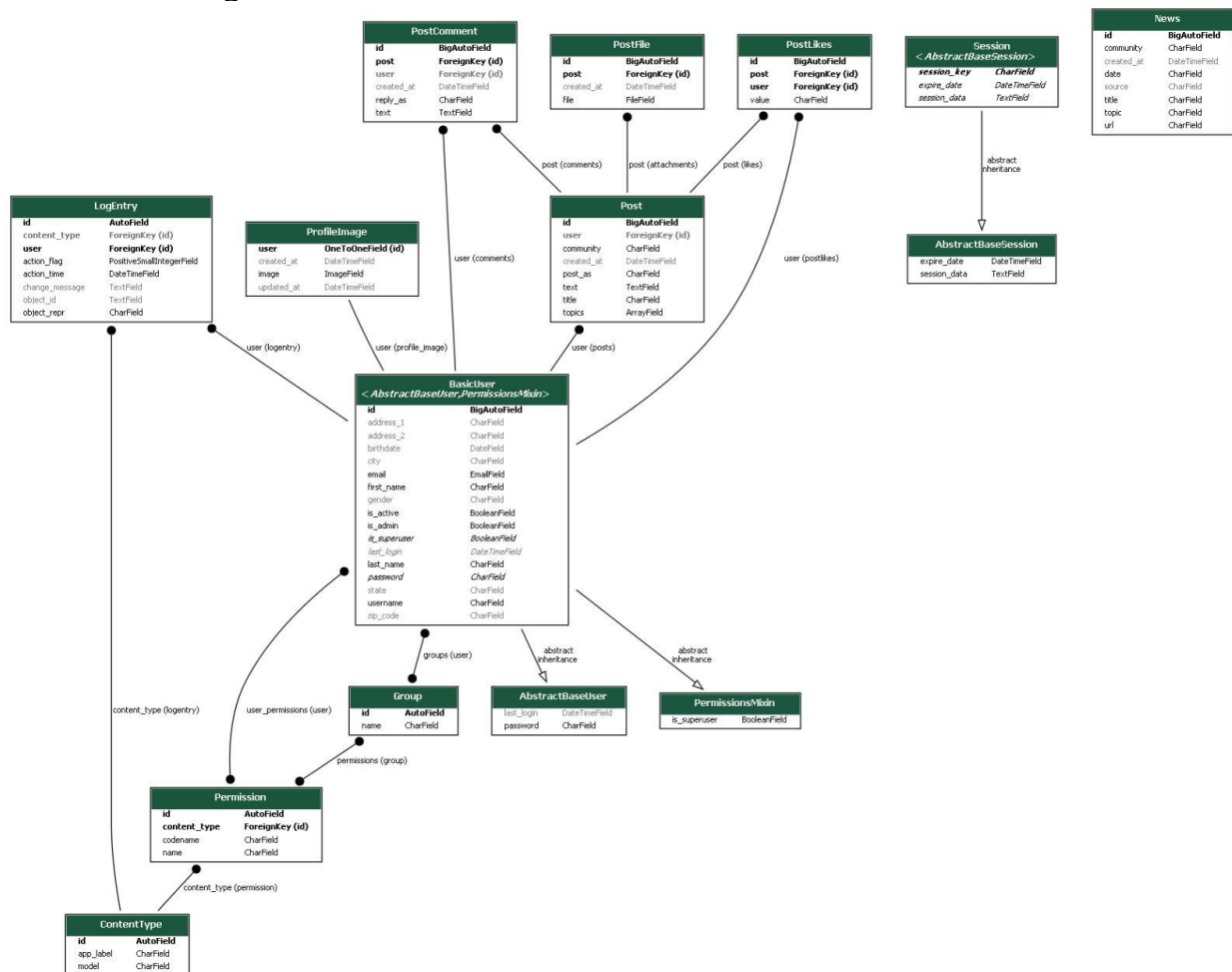
GLOSSARY

- **Framework:** A free and open-source platform of OS-level virtualization to host the needed images(parts) of the project.
- **API:** Application programming interface allowing two pieces of software to communicate with each other.
- **News API:** REST API that returns search results for current and historic news articles published by over 80,000 worldwide sources.
- **Carbon Interface:** API that generates accurate emissions estimates from flights, vehicles, shipping, electricity consumption and fuel combustion based on the latest data and methodologies.
- **Open-Source:** optional visualization
- **Docker:** A free and open-source platform of OS-level virtualization to host the needed images(parts) of the project.
- **Docker desktop application** - optional visualization tool for Docker images.
- **GitHub:** A web-based version-control and collaboration platform for software developers.
- **GitHub desktop:** Optional support application that enables Git collaboration.
- **FIU Server:** Linux server on FIU premises that hosts actual, production versions of the application.
- **Visual Studio Code:** a source-code editor developed by Microsoft for Windows.
- **Bootstrap:** An open-source CSS framework that is also used for quickly and efficiently creating the designs and layouts for web pages.
- **Django:** A Python free-open source framework that manages the back-end of the entire application and custom apps added to Misago core.
- **React:** An open-source, front-end, JavaScript library for building user interfaces or UI components that serve open-source as the engine for the front-end of the application.
- **Next.js:** An open-source development framework that is built on top of Node.js a React that provides additional developer functionality and support for features such as server-side rendering, static site generation, built-in routing and more.
- **PostgreSQL:** A free, open-source database management system that has a separate Docker container and serves as the database for the application.
- **VM:** Virtual machine is the virtualization/emulation of a computer system.
- **OS:** An operating system is system software that manages computer hardware, software resources, and provides common services for computer programs.

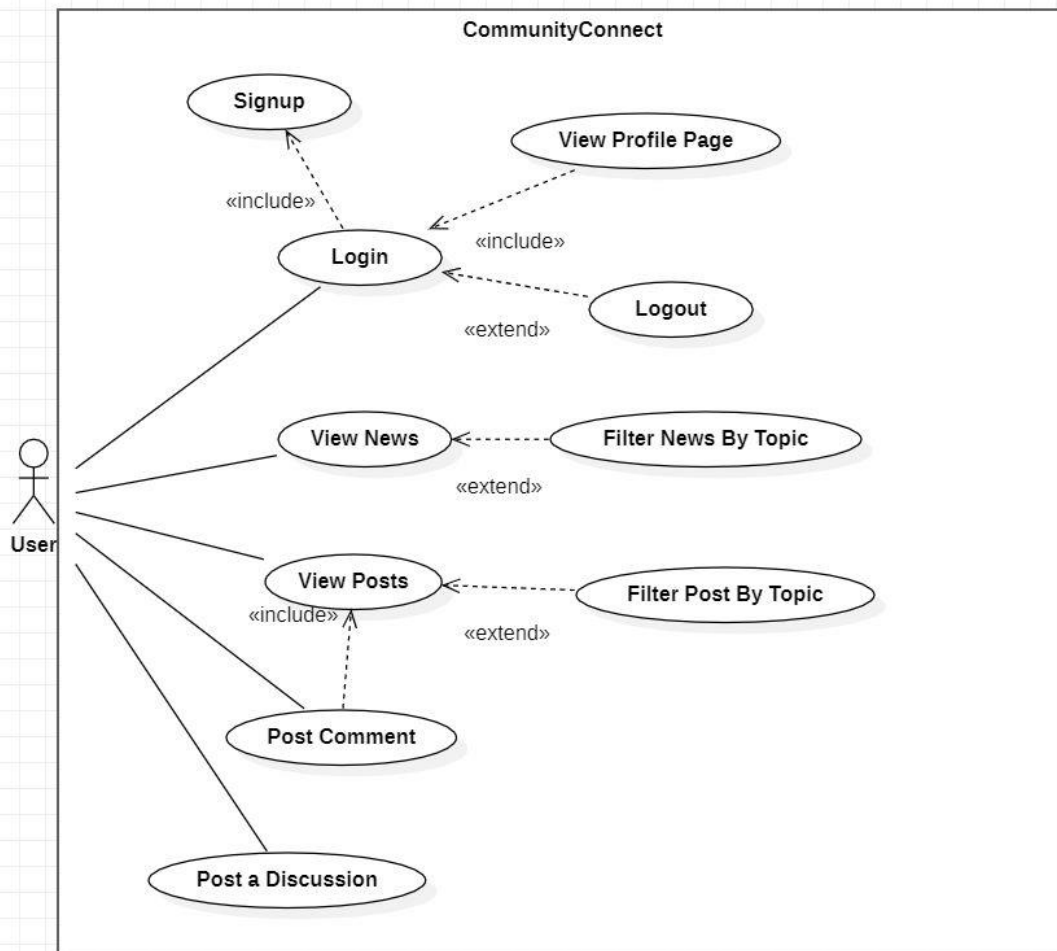
APPENDIX

Appendix A - UML Diagrams

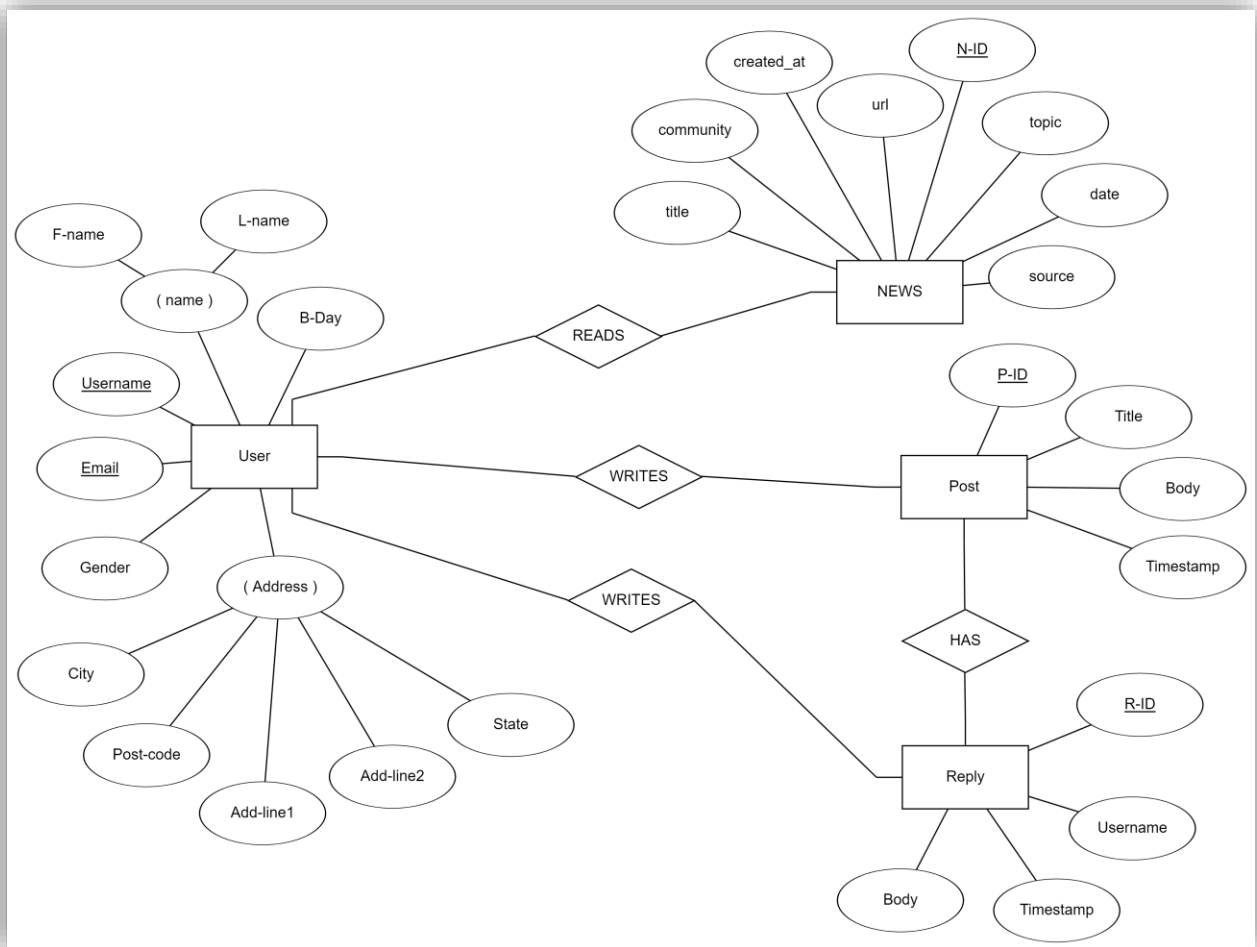
1. Class Diagram



2. Use Case Diagram

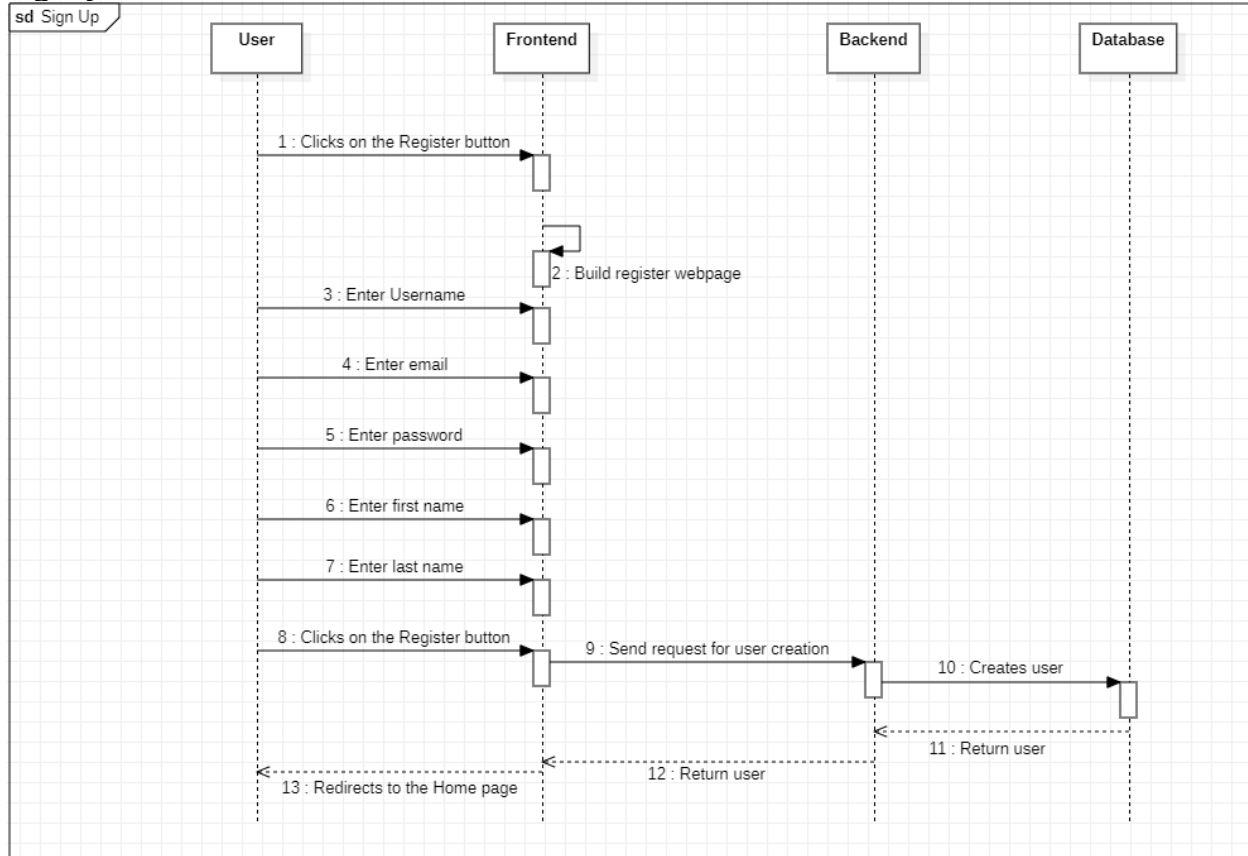


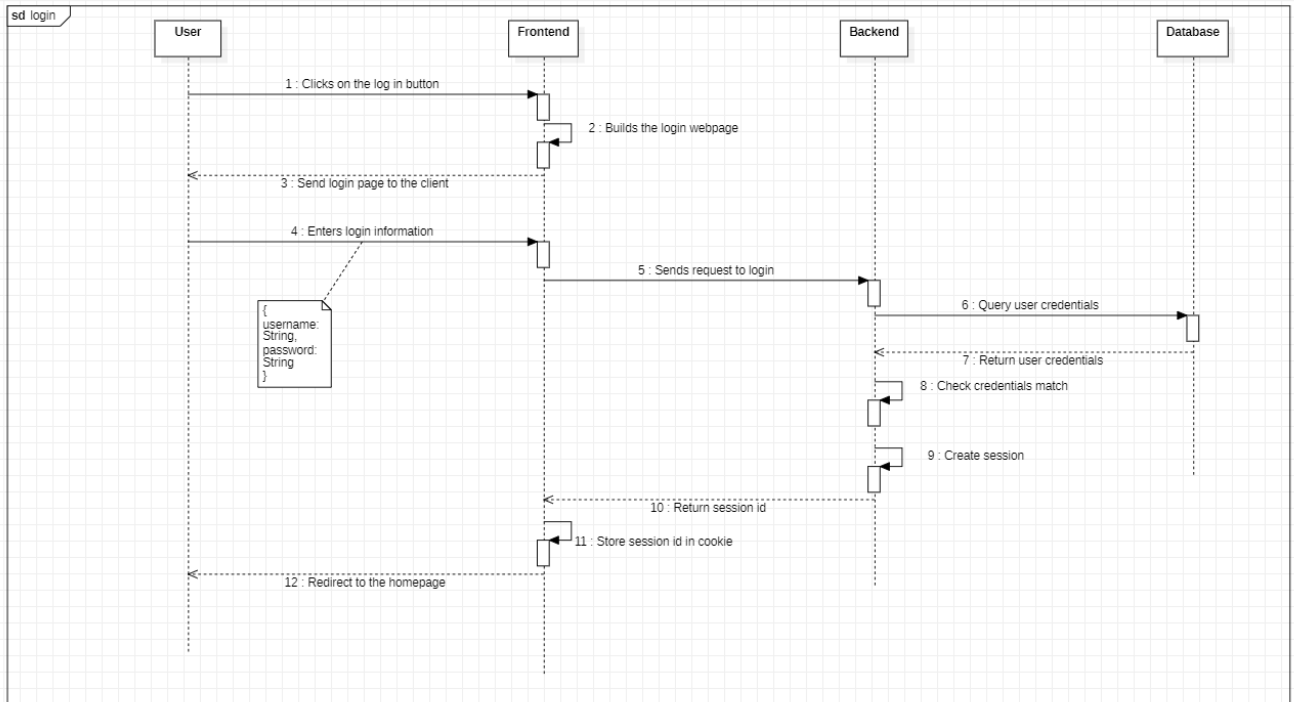
3. ER diagram.



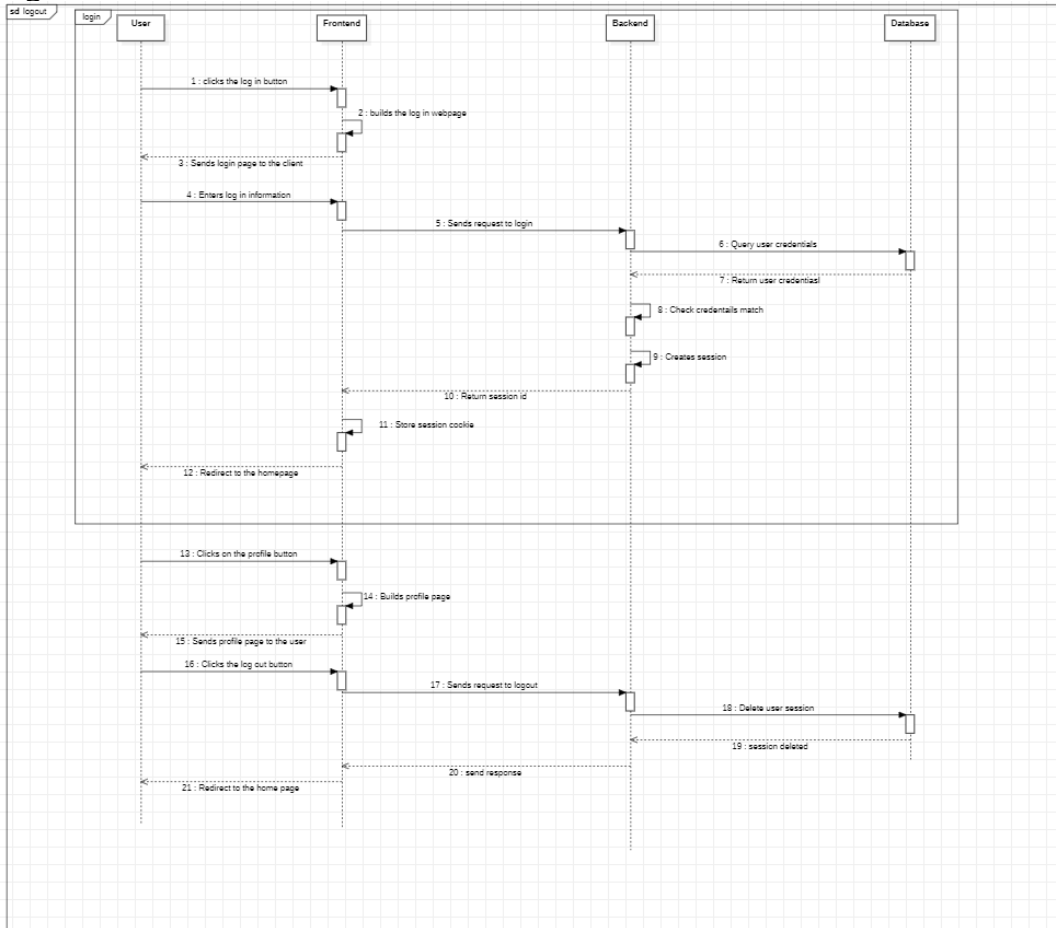
4. Sequence Diagrams

Signup

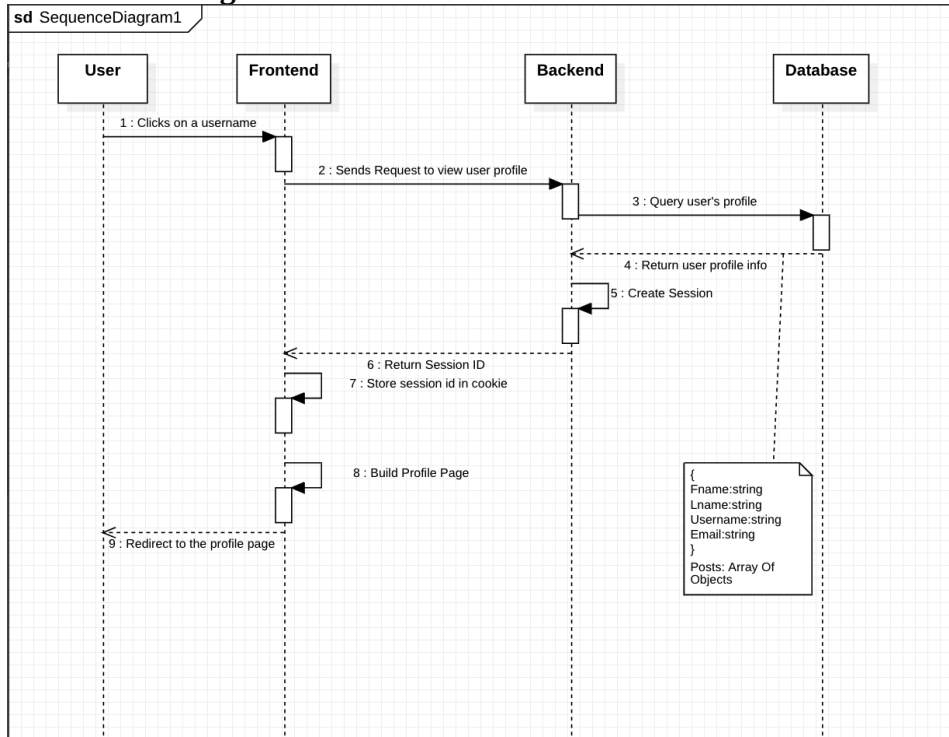


Login

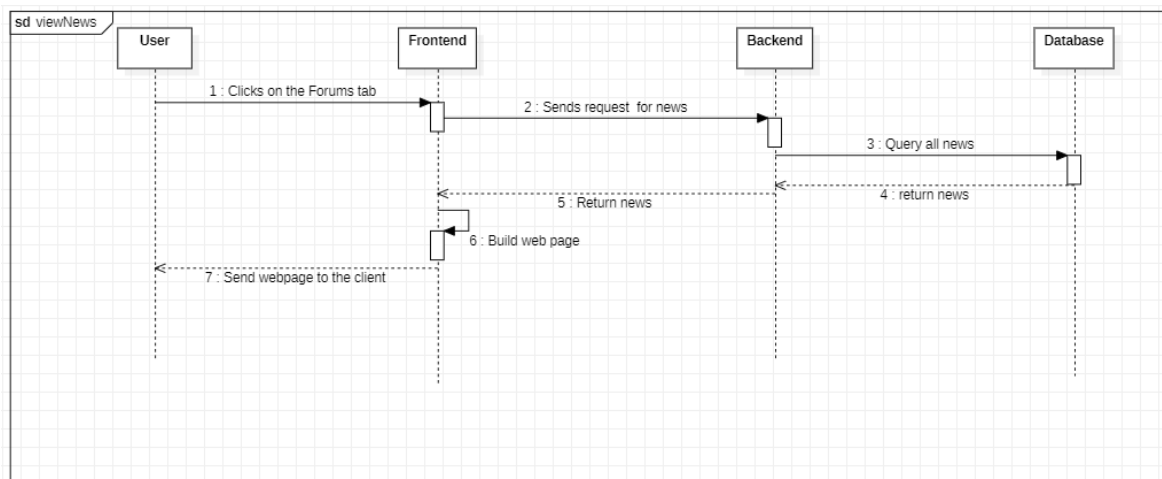
Logout



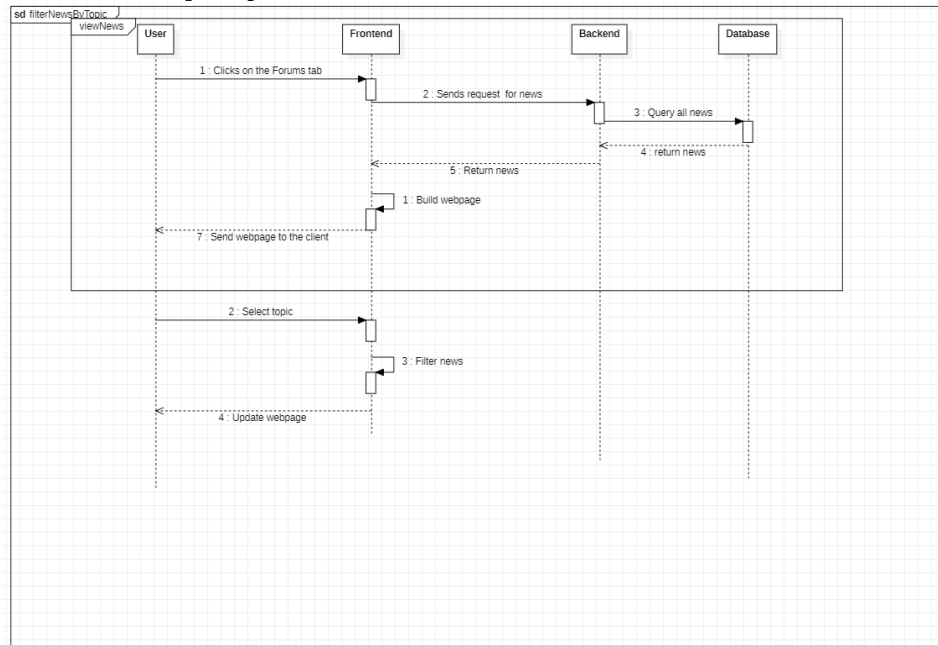
View Profile Page



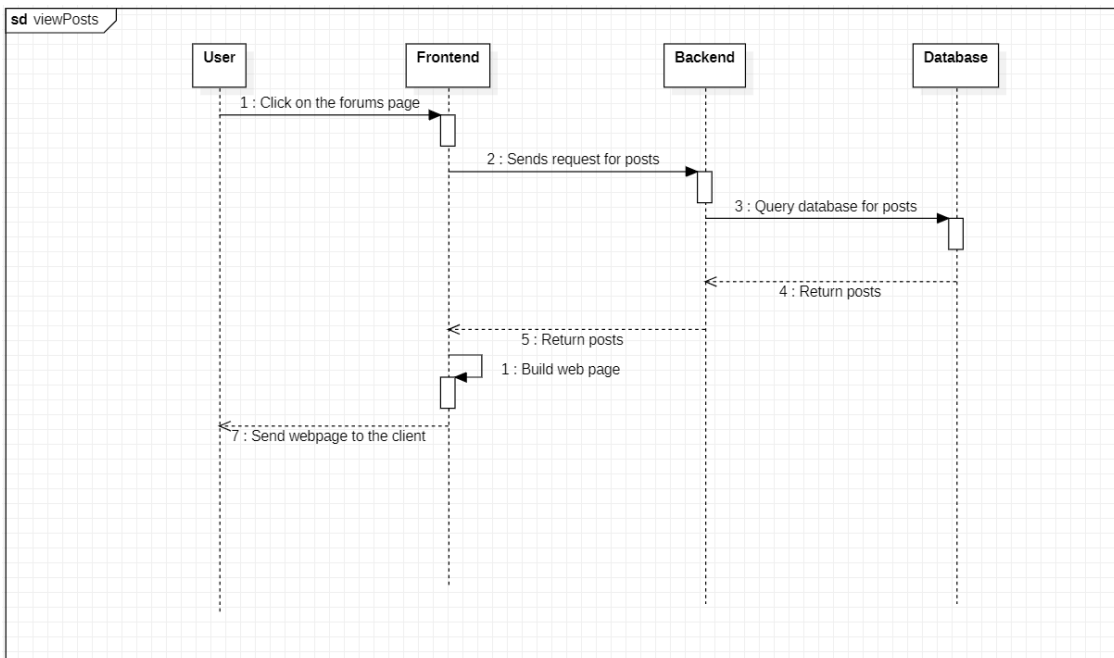
View News



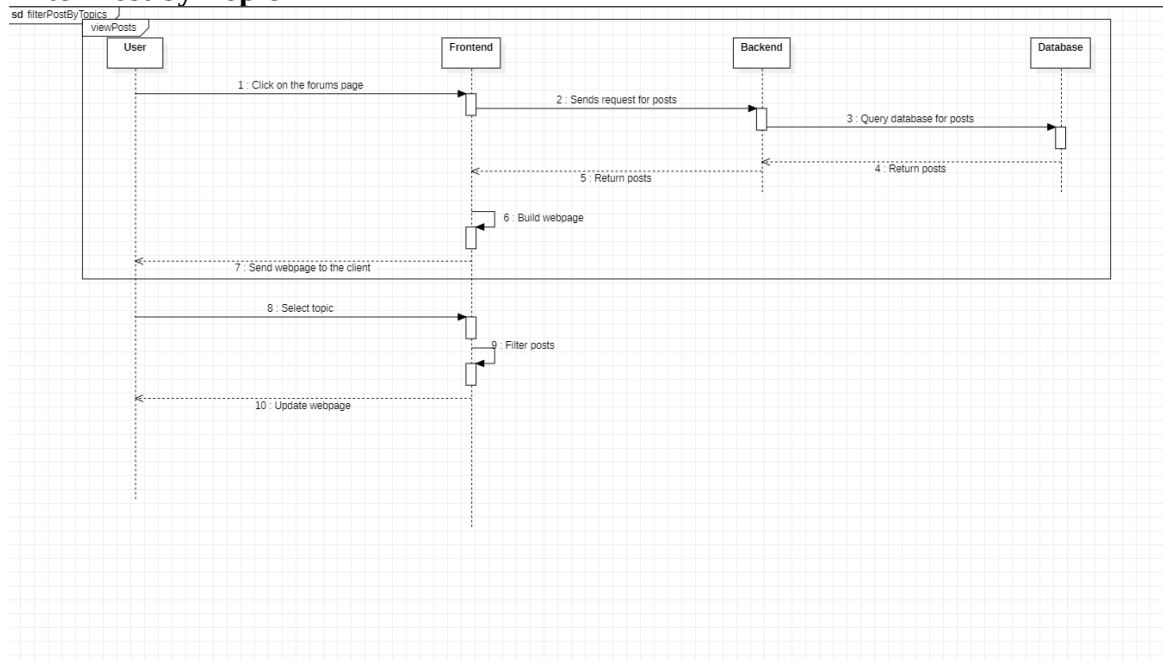
Filter News by Topic



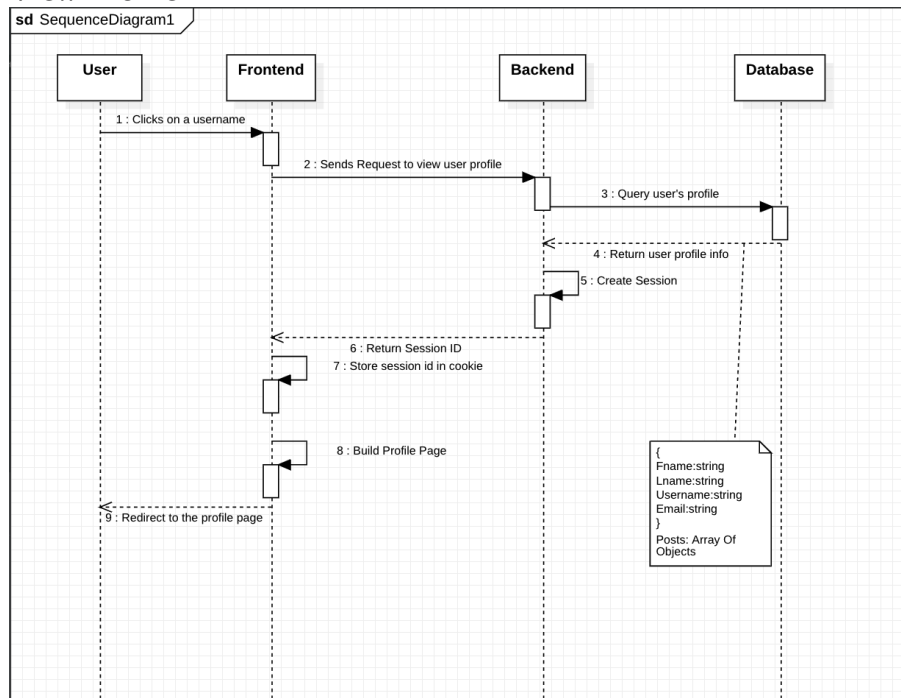
View Posts

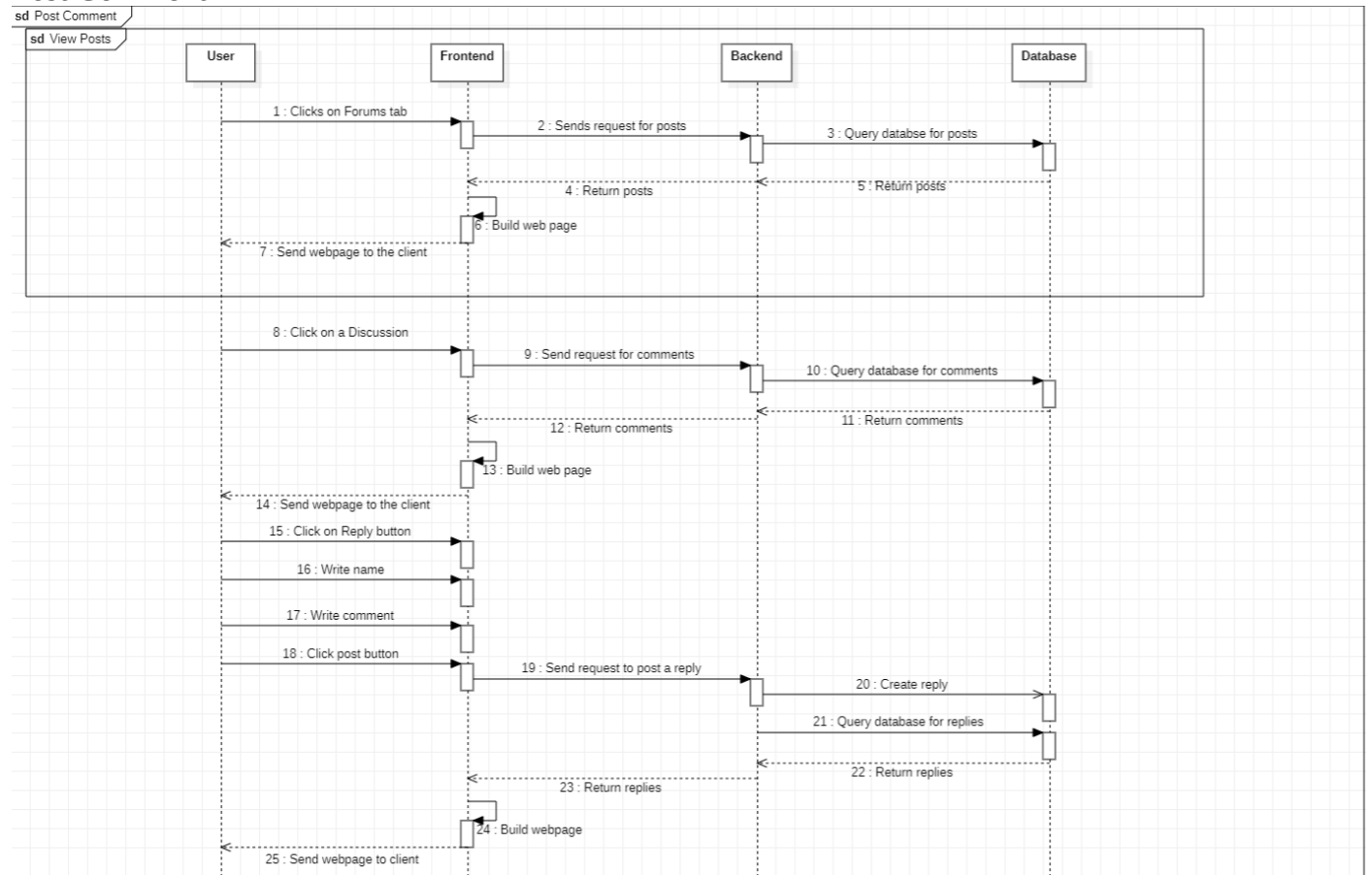


Filter Post by Topic



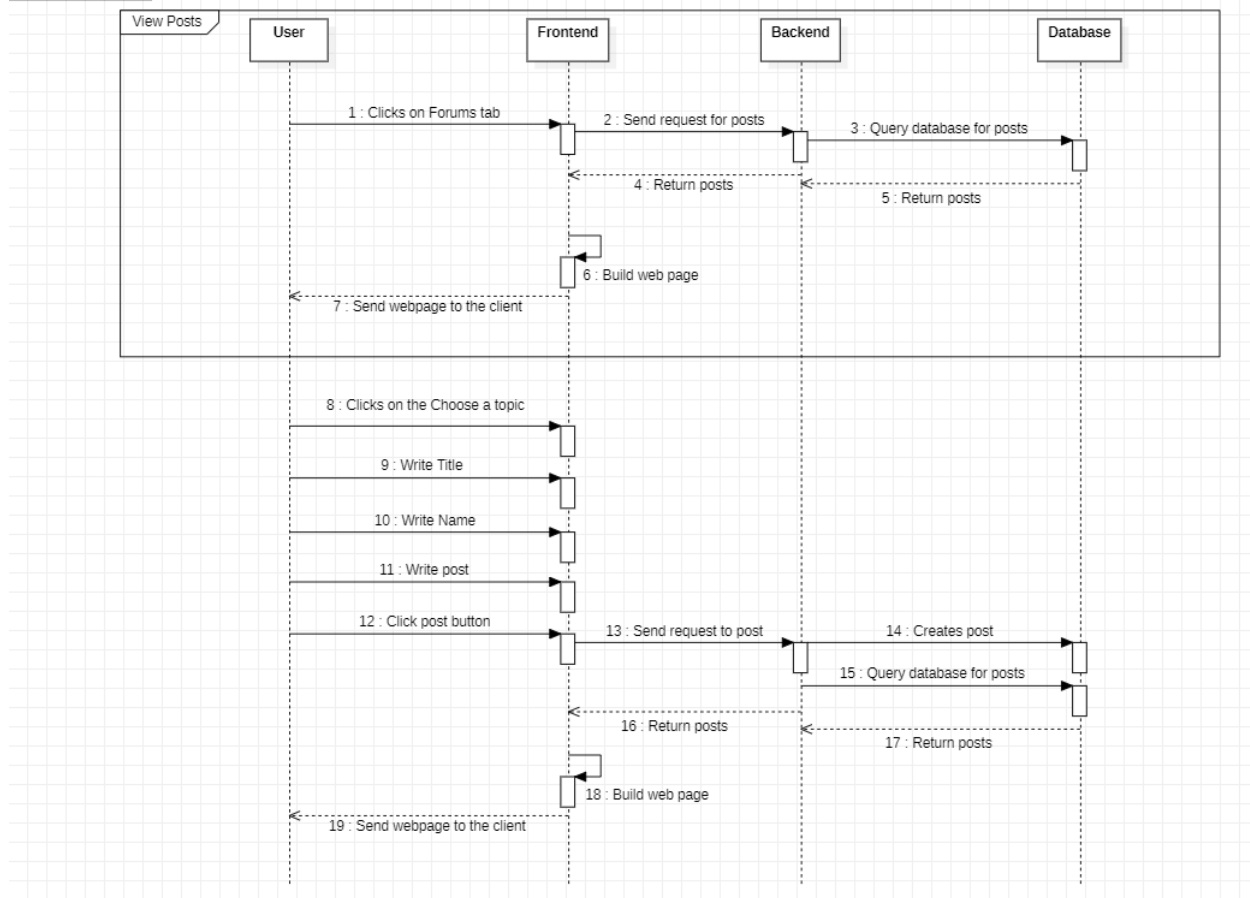
View Profile



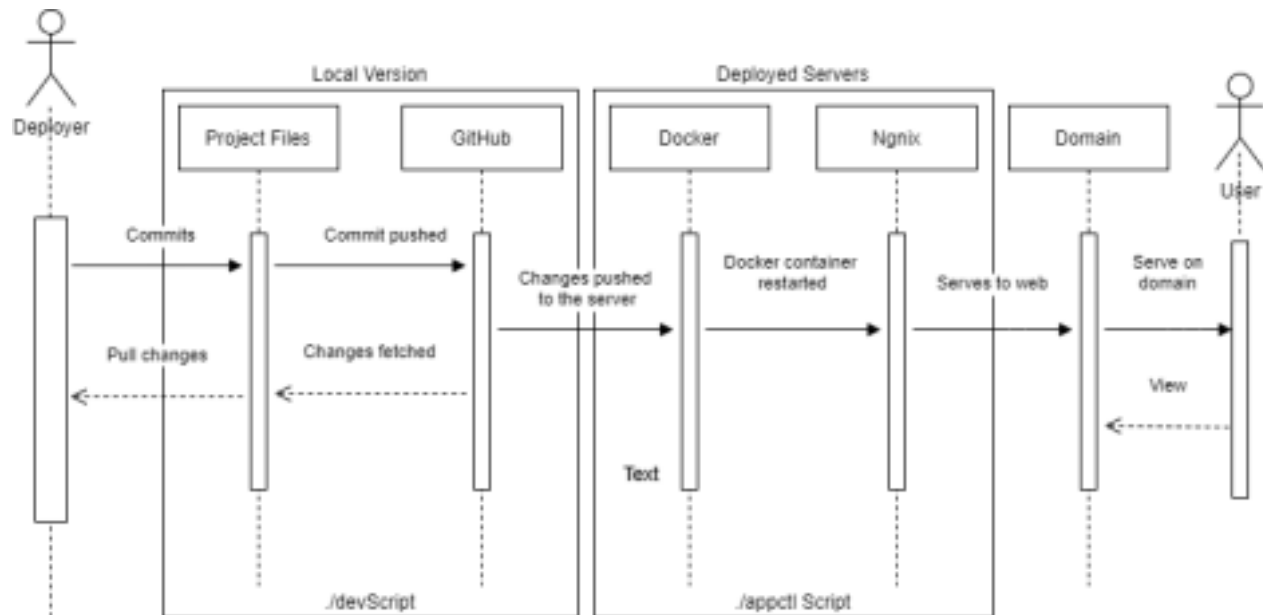
Post Comment

Post a Discussion

sd Post a Discussion

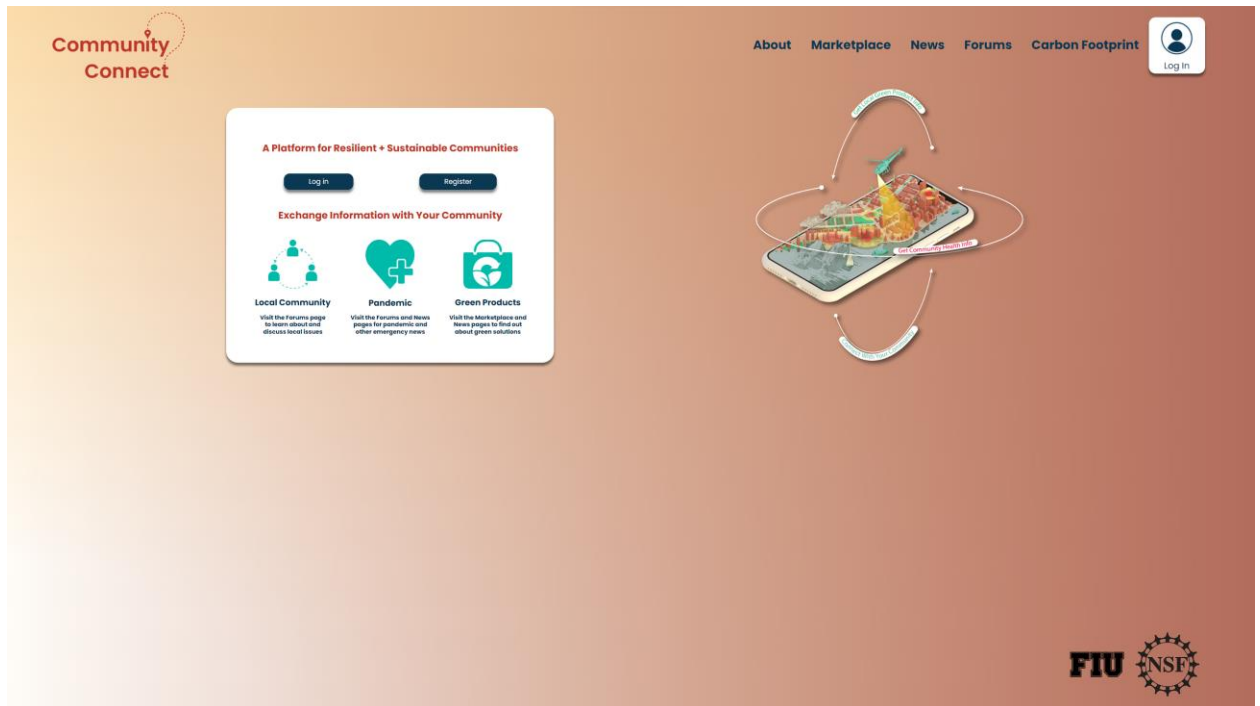


5. Development Cycle Sequence Diagram



Appendix B - User Interface Design

1. Home Page

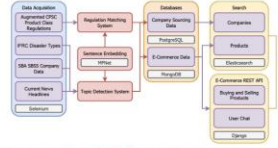


2. Marketplace



[About](#) [Marketplace](#) [News](#) [Forums](#) [Carbon Footprint](#) 

Community-Connect Marketplace App: Block Diagram of the Current Software Architecture



Software Architecture of the AI-Driven Marketplace App

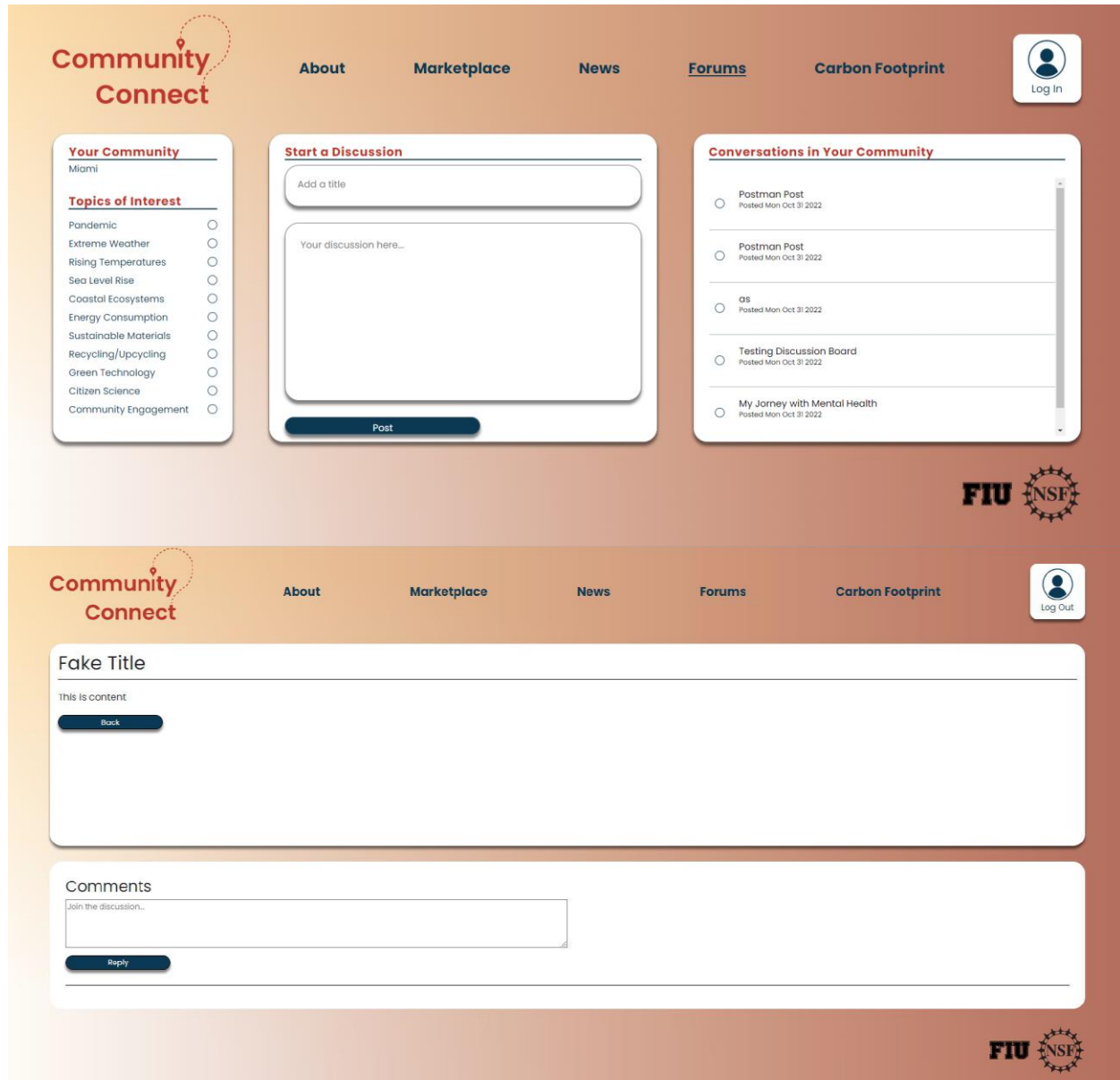


Community-Connect Marketplace App Demo

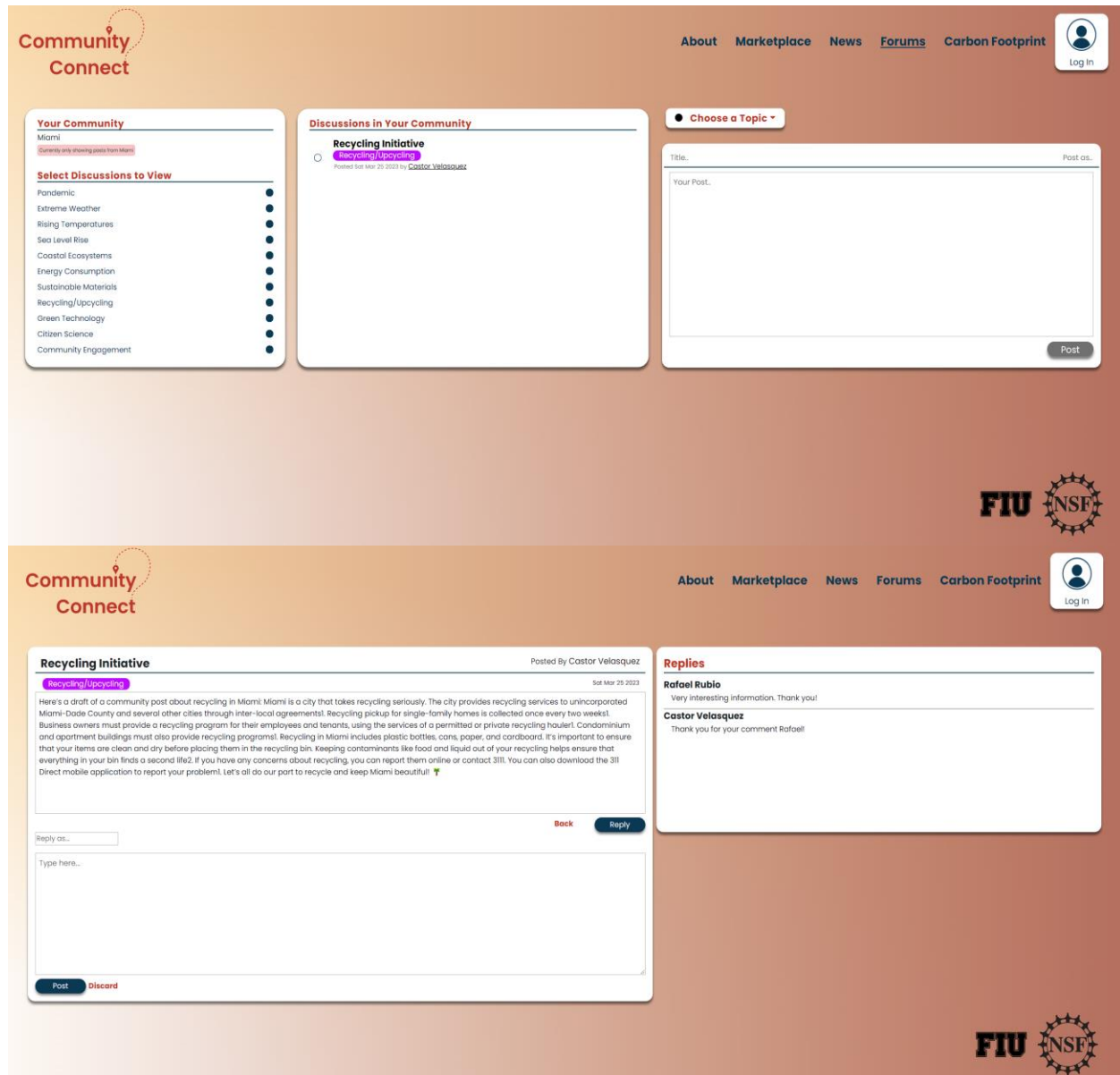


3. Forums Page

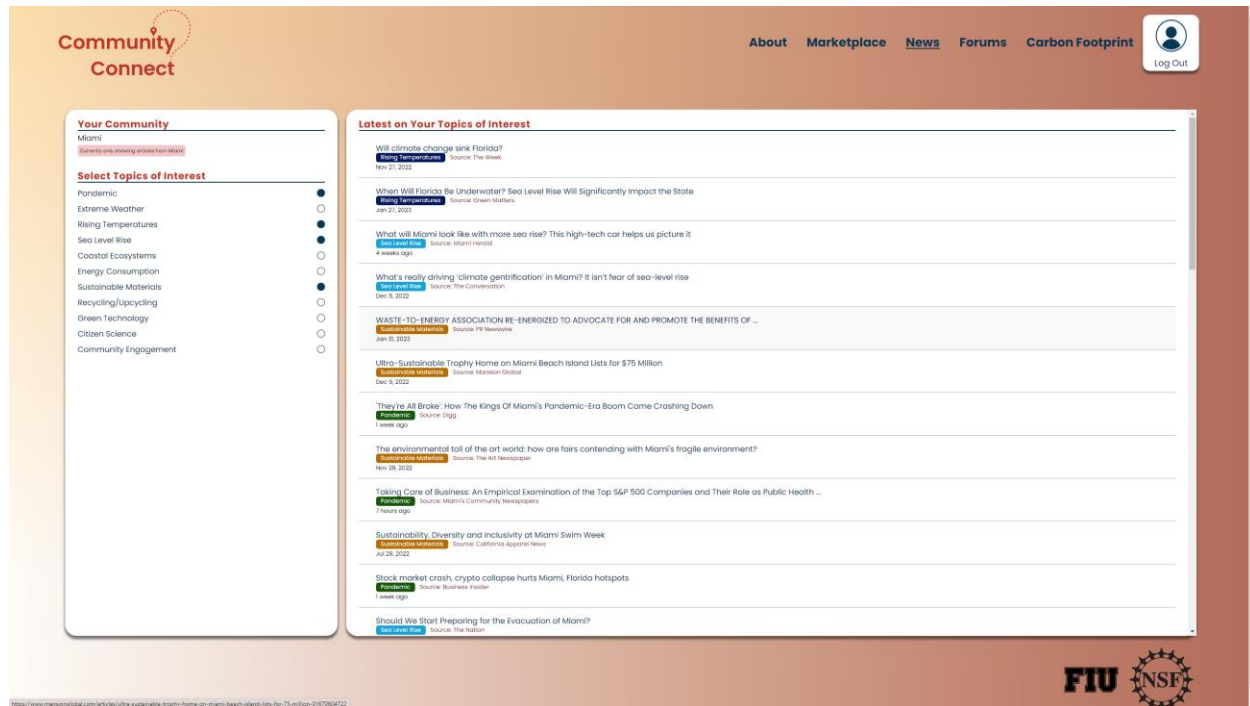
Old design:



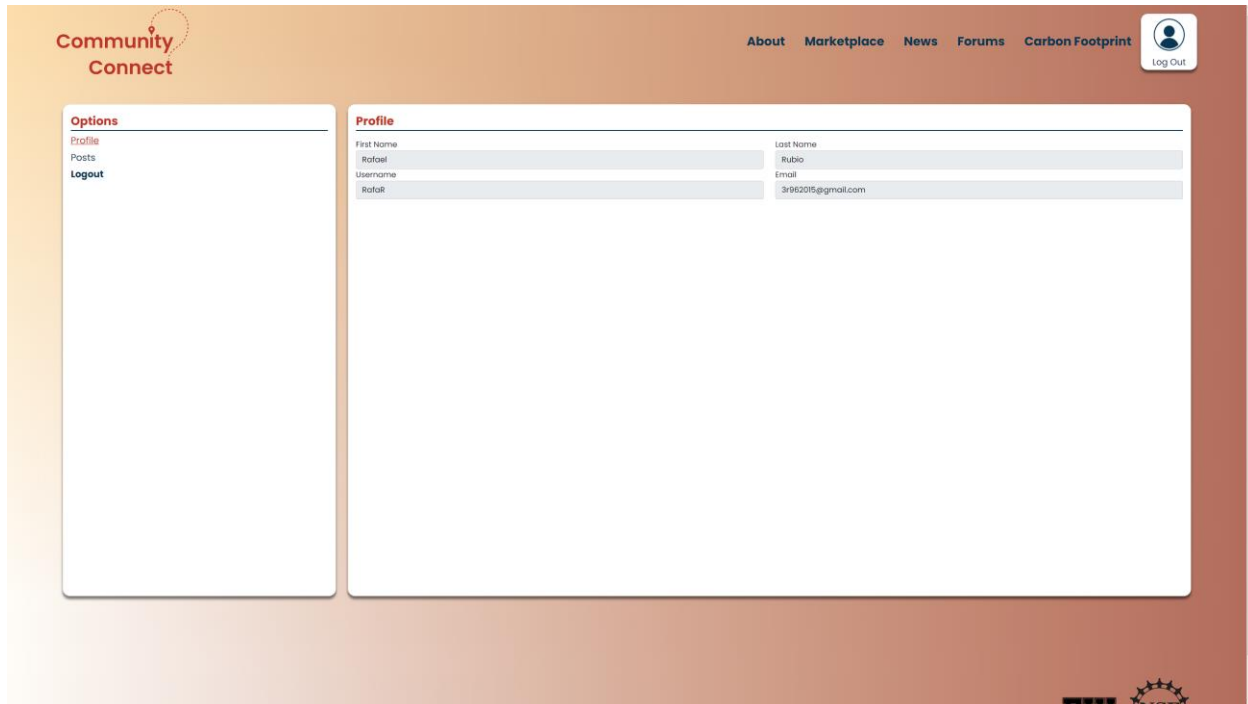
New design:



4. News Page



5. Profile Page



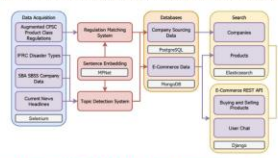
The screenshot displays the 'Community Connect' profile page. The header features the 'Community Connect' logo on the left and navigation links for 'About', 'Marketplace', 'News', 'Forums', and 'Carbon Footprint' on the right. A 'Log Out' button with a user icon is also present. The main content area is divided into two columns. The left column, titled 'Options', contains links for 'Profile', 'Posts', and 'Logout'. The right column, titled 'Profile', contains a form with fields for 'First Name' (Rafael), 'Last Name' (Rubio), 'Username' (Rafael), and 'Email' (3r962015@gmail.com). The page has a light orange background with a darker orange footer area.

Community Connect	
Options	Profile
Profile	First Name
Posts	Rafael
Logout	Last Name
	Rubio
	Username
	Rafael
	Email
	3r962015@gmail.com

6. Marketplace Page

[About](#) [Marketplace](#) [News](#) [Forums](#) [Carbon Footprint](#) 

Community-Connect Marketplace App: Block Diagram of the Current Software Architecture



Software Architecture of the AI-Driven Marketplace App



Community-Connect Marketplace App Demo

Watch on  YouTube



7. Carbon Footprint Page

Community Connect

Press F11 to exit full screen

About Marketplace News Forums Carbon Footprint Log In

What is your Carbon Footprint?



Use our calculator to learn about your share of carbon emission, evaluate your biggest source of contribution, and think of ways to reduce your impact on earth!

1. ELECTRICITY ESTIMATE

Where is your home located?

What is the main source of your electricity?

☐ Natural Gas ☐ Fossil Fuel ☐ Renewable Energy

How much is your homes average electricity consumption per month?

☐ Very High ☐ High ☐ Low ☐ Very Low



FIU NSF

Appendix C - Sprint Review Reports

Sprint Review - Sprint 1

Sprint Review Meeting Minutes

Attendees: Castor Velasquez, Fabiana Fernandez, Allan Prieb, Rafael Rubio, Jordan Rainey

Start time: 5:30 PM

End time: 5:45 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

- User Story S01 (relevant news)
- User Story S03 (multiple topics)

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- User Story S02
- We need to further research ways to get news based on location.

Sprint Review - Sprint 2

Sprint Review Meeting Minutes

Attendees: Castor Velasquez, Fabiana Fernandez, Allan Prieb, Rafael Rubio, Jordan Rainey

Start time: 5:30 PM

End time: 5:45 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

- Research of an appropriate news API that allows to search for news based on location.

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- User Story S02: We need to complete the implementation.

Sprint Review - Sprint 3

Sprint Review Meeting Minutes

Attendees: Castor Velasquez, Fabiana Fernandez, Jordan Rainey, Rafael Rubio, Allan Prieb

Start time: 6:00

End time: 6:30

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

- S05 - Add disclaimer for default community (5 points)
- S06 – Create a scheduled job to fetch news from the API and store them in Postgres (5 points)
- S07 – Update view to get all news from a specific community (5 points)
- S08 – Add call to our backend when user visits the News page (5 points)
- S09 – Show only topics selected by the user (5 points)
- S10 – Add unit tests to the news endpoint (5 points)

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- None

Sprint Review - Sprint 4**Sprint Review Meeting Minutes**

Attendees: Castor Velasquez, Fabiana Fernandez, Jordan Rainey, Rafael Rubio, Allan Prieb

Start time: 6:00

End time: 6:30

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

- S11 – UI/UX Improvements (5 points)
- S12 - Style the Forums page (5 points)
- S13 - Style the Post page (5 points)
- S14 - Create and Modify Post Views (5 points)
- S15 – Add Unit Tests to the Post App (3 points)
- S16 – Connect Frontend and Backend (5 points)

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- None

Sprint Review - Sprint 5

Sprint Review Meeting Minutes

Attendees: Castor Velasquez, Fabiana Fernandez, Jordan Rainey, Rafael Rubio, Allan Prieb

Start time: 6:00

End time: 6:30

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

S17 – Display tags in posts (5 points)

S18 – Change order of cards (5 points)

S19 – Improve the start a Discussion card (5 points)

S20 – Add name to Posts and Replies (5 points)

S21 – Make Forums page mobile friendly (5 points)

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- None

Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents

Installation

Community Connect Prototype (skip all docker commands for now)

Required Services

- [Git](#)
- [Node v16.13.0](#)
- [Yarn](#)
- [Python](#)
- [Docker](#)
- [Docker-Compose](#) only if you're using Linux
- [Firefox Browser](#)
- [PostgreSQL 13+](#)

Structure of the project

- Postgres database
- [Django Rest Framework](#) backend which is a rest api Note: it is different from default Django tools
- [Nextjs](#) / [React](#) Front-end
- Frontend and Backend are connected using a javascript library called [axios](#)
- Docker is used to containerize and deploy

highly recommend to look through the documentation and understand tools as a beginner ###
Service installation guides for: | MacOS | Windows | Linux |

Running Project Locally

Once you've installed all of the required services, open a **Terminal Window** and follow these instructions:

1. Clone the repository

HTTPS

```
$ git clone https://github.com/fiu-rdflab-communityconnect/Community-Connect.git
```

SSH

```
$ git clone git@github.com:fiu-rdflab-communityconnect/Community-Connect.git
```

2. Navigate to the project's root directory (i.e. where package.json, docker-compose.yml, and .dockerignore files are located)

```
$ cd Community-Connect
```

3. Install node dependencies for the Frontend

```
$ yarn front:i
```

Windows Backend Installation Instructions

4. Create a virtual environment for the backend.

```
$ cd backend
```

```
$ py -m venv VE
```

5. Activate the virtual environment.

```
$ VE\Scripts\activate
```

6. Install the dependencies in the environment.

```
$ py -m pip install -r requirements.txt
```

```
$ py -m pip install TK
```

If it tells you that you are missing a module, install that module using python.

7. Migrate the database models.

```
$ py manage.py makemigrations
```

```
$ py manage.py migrate
```

*Mac/Linux Backend Installation Instructions**4. Create a virtual environment for the backend.*

```
$ cd backend/
```

```
$ python3 -m venv VE
```

5. Activate the virtual environment.

```
$ source VE/bin/activate
```

6. Install the dependencies in the environment.

```
$ python3 -m pip install -r requirements.txt
```

```
$ python3 -m pip install TK
```

If it tells you that you are missing a module, install that module using python.

7. Migrate the database models.

```
$ python manage.py makemigrations
```

```
$ python manage.py migrate
```

Running the project locally (from the root directory, no environment)**Running the frontend**

```
$ cd frontend
```

```
$ npm run dev
```

Running the backend*Windows*

```
$ cd backend
```

```
$ VE\Scripts\activate
```

```
$ py manage.py runserver
```

Mac

```
$ cd backend/
```

```
$ source VE/bin/activate
```

```
$ python manage.py runserver
```

Deployment through Ocelot using docker

This section was the old method of testing the backend, but it is now only in use for deployment.

4. Build the docker image (skip for now)

```
$ docker-compose build
```

5. Make and migrate database models

This builds the DB models:

```
$ docker-compose run web python manage.py makemigrations
```

After the top command finishes executing, run the following:

```
$ docker-compose run web python manage.py migrate
```

6. Create an admin user for the Backend (skip for now)

```
$ docker-compose run web python manage.py createsuperuser
```

- You will be prompted to create an account with a username, email (not necessary), and password.

7. Change your Windows settings/MacOs settings in your host file (skip for now)

This change will change the web url community-connect.fiu.edu to your local machine

Run Notepad as an administrator

Windows directory: MainDrive (exmp C:)\Windows\System32\drivers\etc\hosts

Open it as a notepad and at the bottom of the file add:

```
127.0.0.1 community-connect.fiu.edu
```

MacOs directory: open terminal and type

```
$ sudo nano /etc/hosts
```

At the end of the file add

```
127.0.0.1 community-connect.fiu.edu
```

then click control + X to save and press y and enter and then enter again to accept changes and save the file

8. Run the project (skip docker commands)

To run with live logs: (skip)

```
$ docker-compose up
```

To run in detached mode (no logs): (skip)

```
$ docker-compose up -d
```

To run backend server:

```
cd Community-Connect\backend python manage.py runserver
```

To run frontend:

```
cd Community-Connect\frontend npm run dev
```

- Once completed, you should be able to go to <http://localhost:3000> and view the website

- You can also go to <http://localhost:8000/admin> to access the admin page of the backend service

***Note:** It may take several seconds to load the first page.

9. Shutting down the project

If you're running the project with the live logs, you can press CTRL+C to shutdown the project, followed by entering the following command:

This will remove all declared and anonymous volumes that were attached to the containers

```
$ docker-compose down -v
```

- If you're running in detached mode, just run command above in the root directory of the project

Setting up the Backend Development Environment (Outdated)

This section is only necessary if you are working in the Backend so you can get IntelliSense support from the used libraries.

1. Install pipenv

On Linux and Windows, run:

```
$ pip3 install pipenv
```

On MacOS, run:

```
$ brew install pipenv
```

2. Enter the shell environment

```
$ pipenv shell
```

3. Installing all required packages

```
$ pip3 install -r backend/requirements.txt
```


Shortcomings

In this iteration of the project, we could not implement the ability for users to get news from different communities, other than Miami, due to budget constraints. Fetching news from different communities requires a high number of API calls to SerpApi, which would require a premium license from SerpApi, it was the Product Owner's decision to only allow users to get news from Miami and if interest in the project evolves, we would revisit the idea of allowing more communities.

Moreover, we rebuilt the UI for the forums and news page using responsive design, so they resize for different screen sizes, but we were not able to improve the current design of the home, about, marketplace, and carbon footprint page due to time constraints.

Wishlist

Below is a list of items that were out of the scope for this iteration, but would be nice to have and should be considered for future iterations:

- Ability to edit and delete posts.
- Ability to edit and delete comments.
- Ability to like/dislike comments and posts.
- Ability to upload files to posts/comments.
- Ability to see liked posts in the profile page.

REFERENCES

<https://docs.docker.com>

<https://www.django-rest-framework.org>

<https://reactjs.org>

<https://nextjs.org/>

<https://serpapi.com/>