

INTRODUCTION

ProUML is a UML diagramming web application where users can create and edit UML diagrams, import Java code to translate into UML diagrams, and collaborate with other users on UML diagrams in real-time. This application has been developed using a variety of languages and frameworks. The languages used to develop *ProUML* include Go for hosting the website and Next.js with React in TypeScript for the front end of the application. The frameworks and libraries used in *ProUML* include AntV X6 for frontend diagram design support, Tailwind for our UI design, Redis to implement our live sharing feature for user collaboration, and Postgres for user data storage. *ProUML* is also open source, so other developers can also contribute to the application's development.

ARCHITECTURE

Figure 1 portrays the overall architecture of *ProUML*, which includes both a frontend and backend. The frontend of *ProUML* controls the page routing and page access, while the backend is in charge of server responsibilities, which includes the responsibility of translating code to UML, user data access, and hosting the website. The frontend follows a Model-View-Controller (MVC) architecture, which is a common design utilized among the software engineering industry. *ProUML* uses this design mainly to support user authentication, as the controller determines which pages can be routed to in the view if the users are logged in or not, and the model contains the data that may be visualized in the view. This figure mainly portrays the architecture of *ProUML* when the server translates imported code, as seen in the backend design.

Figure 1: Architectural Design

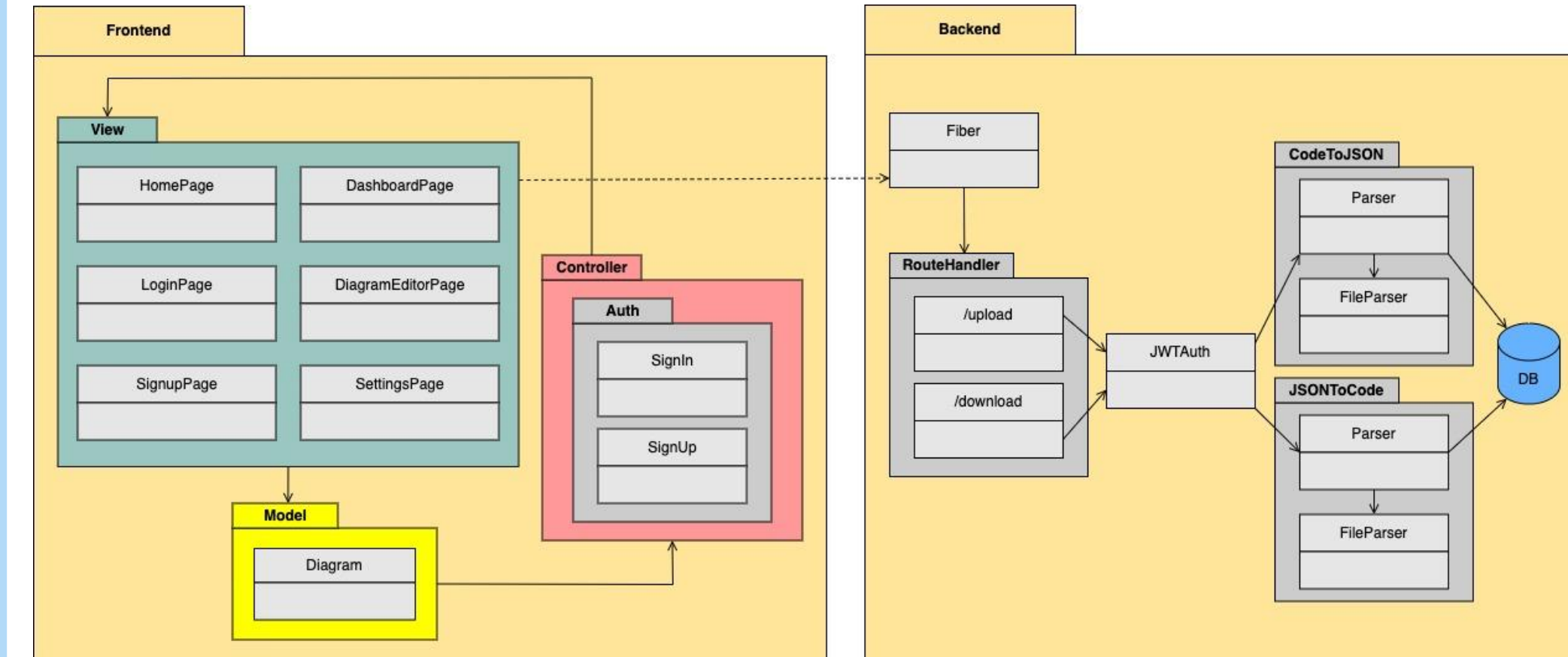
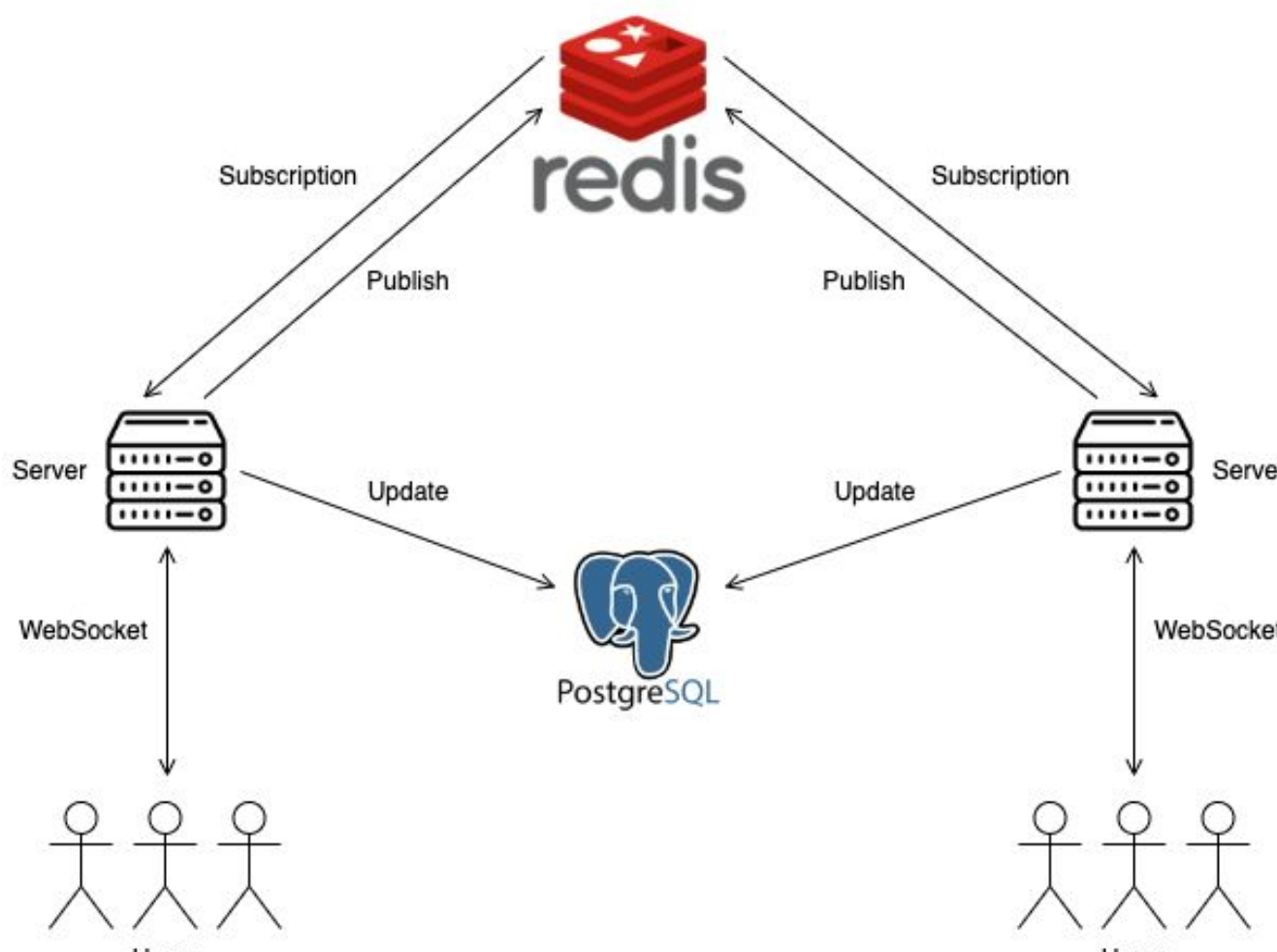


Figure 2 visualizes the design of *ProUML*'s live sharing feature, which allows users to collaborate with each other in real-time on UML diagrams. In this design, multiple users from different distributed *ProUML* servers can communicate (subscribe) to Redis, which is a millisecond response time database with the capabilities of supporting user collaboration across multiple servers. Once users make changes on collaborated UML diagrams, the changes are directly sent from Redis back to the server (publish). These changes are then sent to Postgres to update the current saved diagram data.

Figure 2: Real-time Collaboration Design



AUTHORS & ABSTRACT

STUDENT TEAM: MARIN R. MIRASOL, AMER YONO, COREY C. TAYLOR

FACULTY ADVISER: DR. XIAOCONG (SIMON) FAN

INDUSTRY MENTOR: DR. XIAOCONG (SIMON) FAN

SPONSORED BY: CSUSM CSTEM PROGRAM

ABSTRACT

With the complexity of software increasing, as technology becomes more advanced, many software engineers today need to use tools to make complex software systems more understandable. One tool that many software engineers use is Unified Modeling Language (UML), which is a modeling language that visualizes software designs from various perspectives. Although UML helps to make software design more understandable, it is very time-consuming because engineers need to manually construct UML diagrams, and then translate diagrams into code, and vice versa. With *ProUML*, engineers can easily import source code from software projects to translate automatically into UML. Conversely, they can do the opposite, which is to translate constructed UML diagrams into source code. Not only is *ProUML* convenient for design activities, but it can also be easily expanded to support newer programming languages if other developers are willing to implement additional language support for our system. Overall, with *ProUML*, our goals are to make software development more efficient, less costly, and less time-consuming.

RESULTS

Figure 3 shows *ProUML*'s home page, which is the initial landing page when users navigate to our application's URL: <https://prouml.com/>. This page will show information about our application to inform new users of the features of *ProUML*. Additionally this page will include a button for existing users to navigate to the login page to log into their accounts, along with a button for new users to navigate to the signup page to create a new account.

Figure 3: Home Page

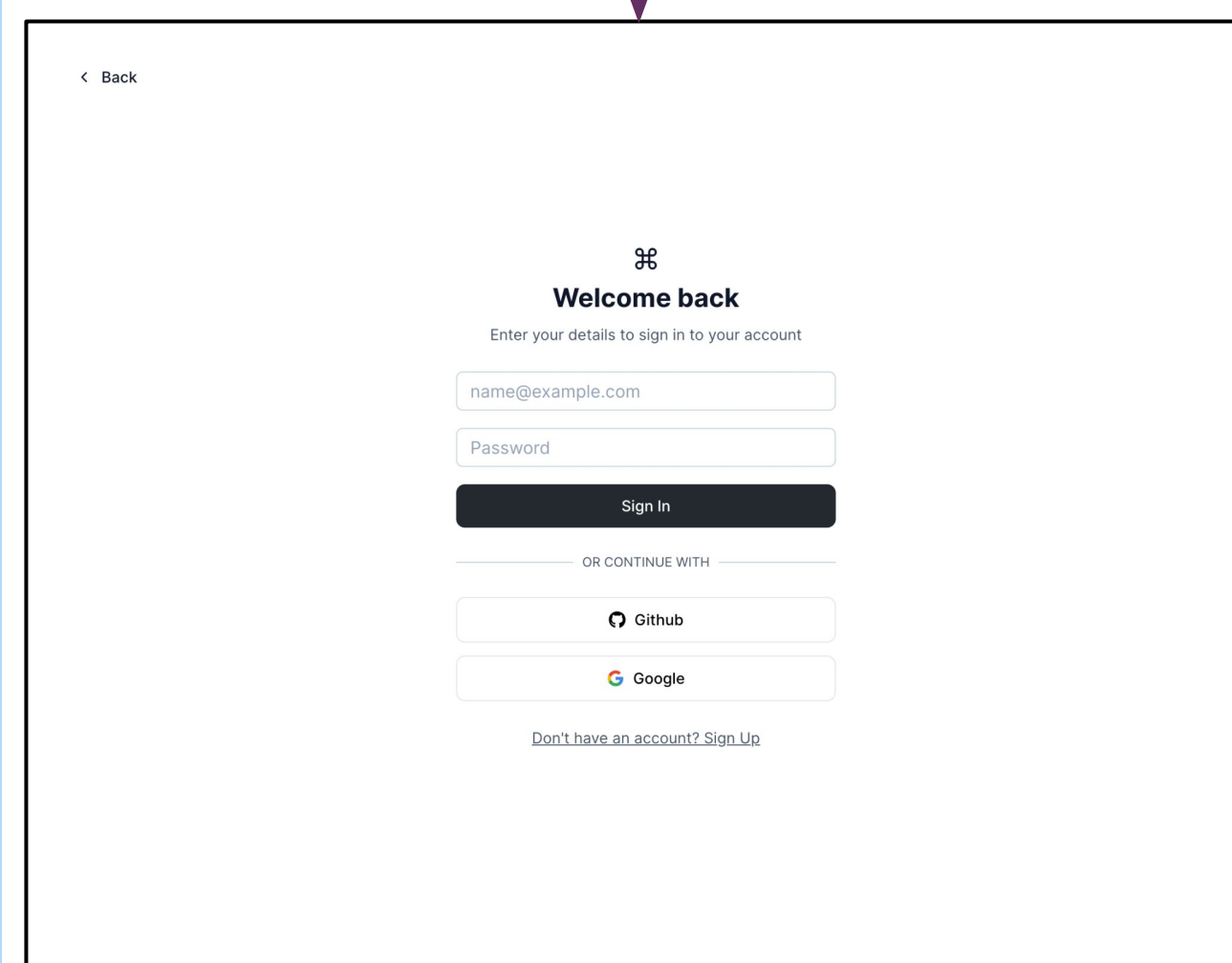
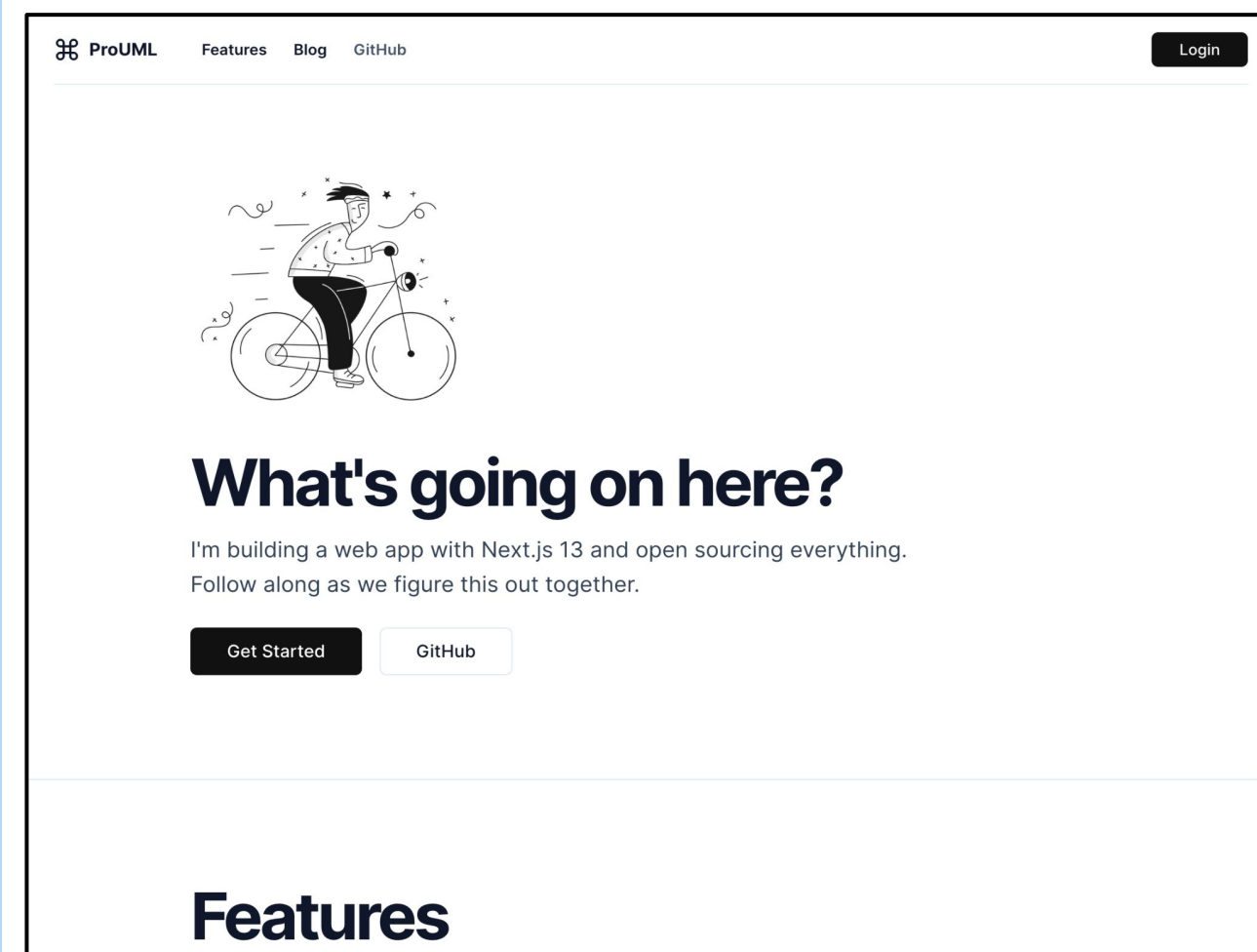


Figure 4: Login Page

Figure 4 shows *ProUML*'s login page, where existing users can log into their accounts. Once logged in, users are navigated to the dashboard page, where they can access all of their saved diagrams.

Figure 5: Signup Page

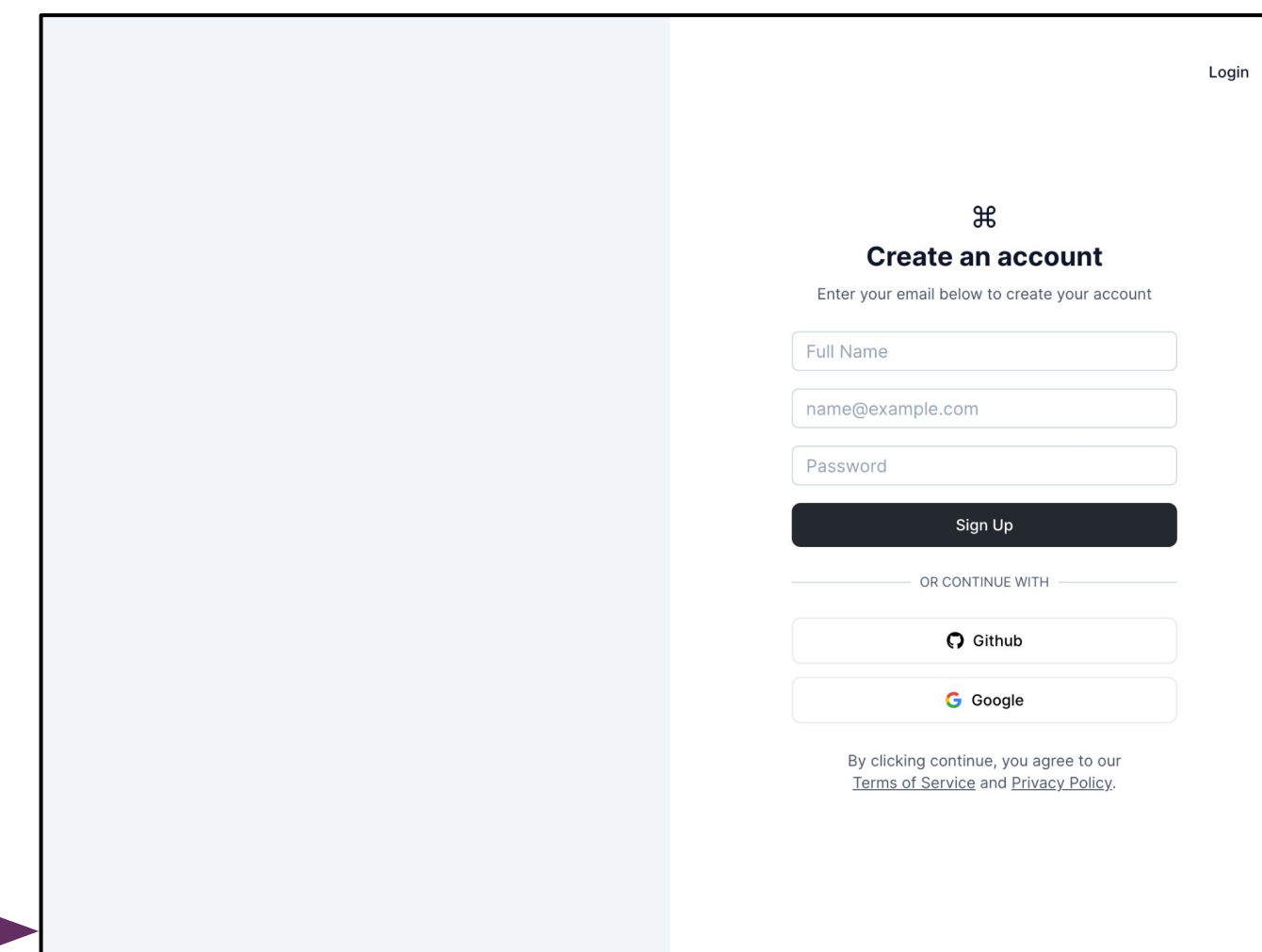


Figure 5 shows *ProUML*'s signup page, which is where new users can create an account to utilize the main features of the application. To create an account, users must input their name, email, and password. Users can also make an account by using their GitHub or Google accounts. Once new users create their accounts, they will be navigated to their dashboard page, which is where they will be prompted to start creating new UML diagrams.

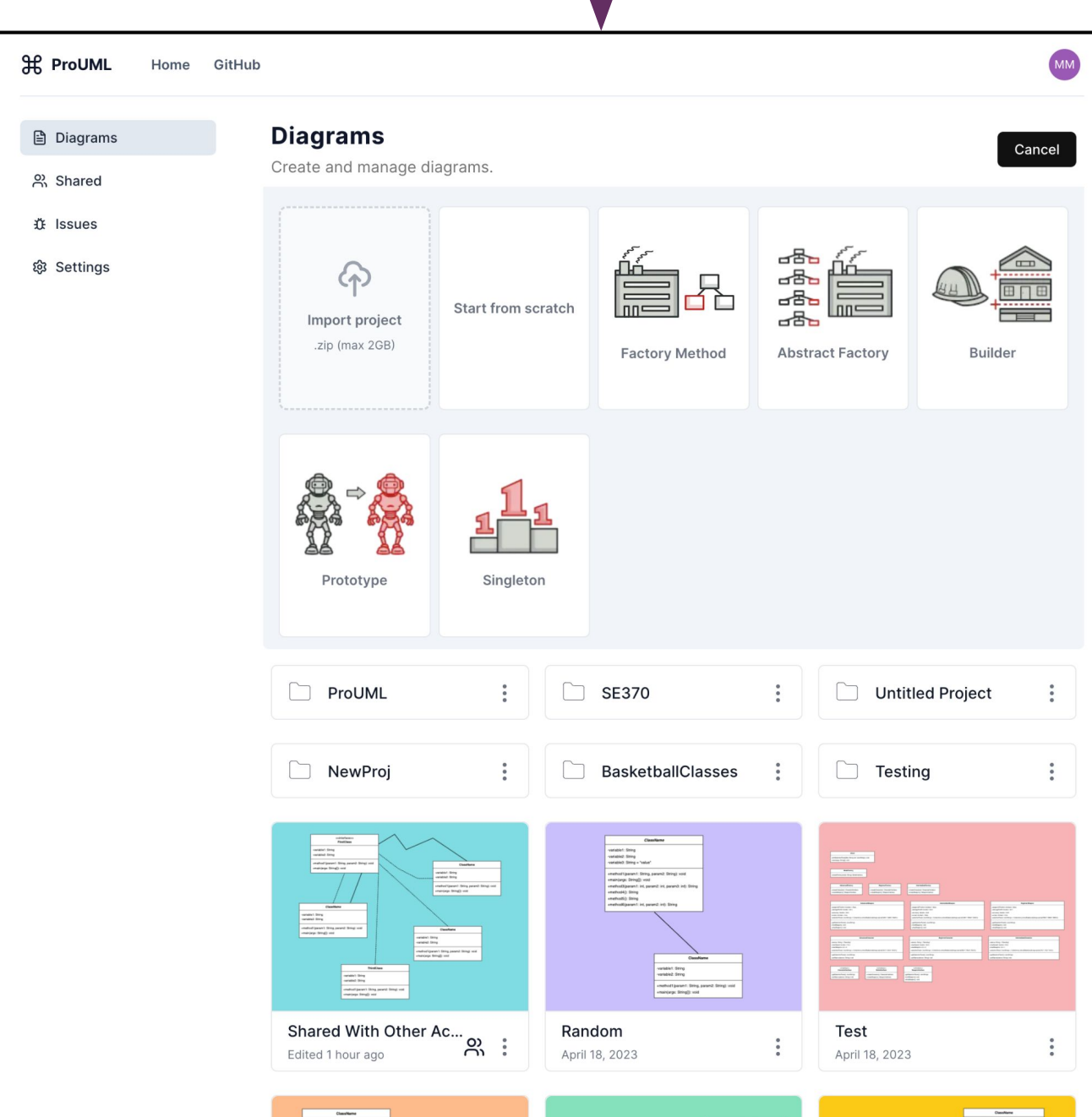


Figure 6: Dashboard Page

Figure 6 shows *ProUML*'s dashboard page, where users will be shown all of their saved diagrams. From this page, users can also create new UML diagrams and can either create diagrams from scratch, from a template, or by importing code to be translated to UML. If users click on any of these diagrams in this page, they will be navigated to the diagram editor page, where they can edit and update the clicked diagram.

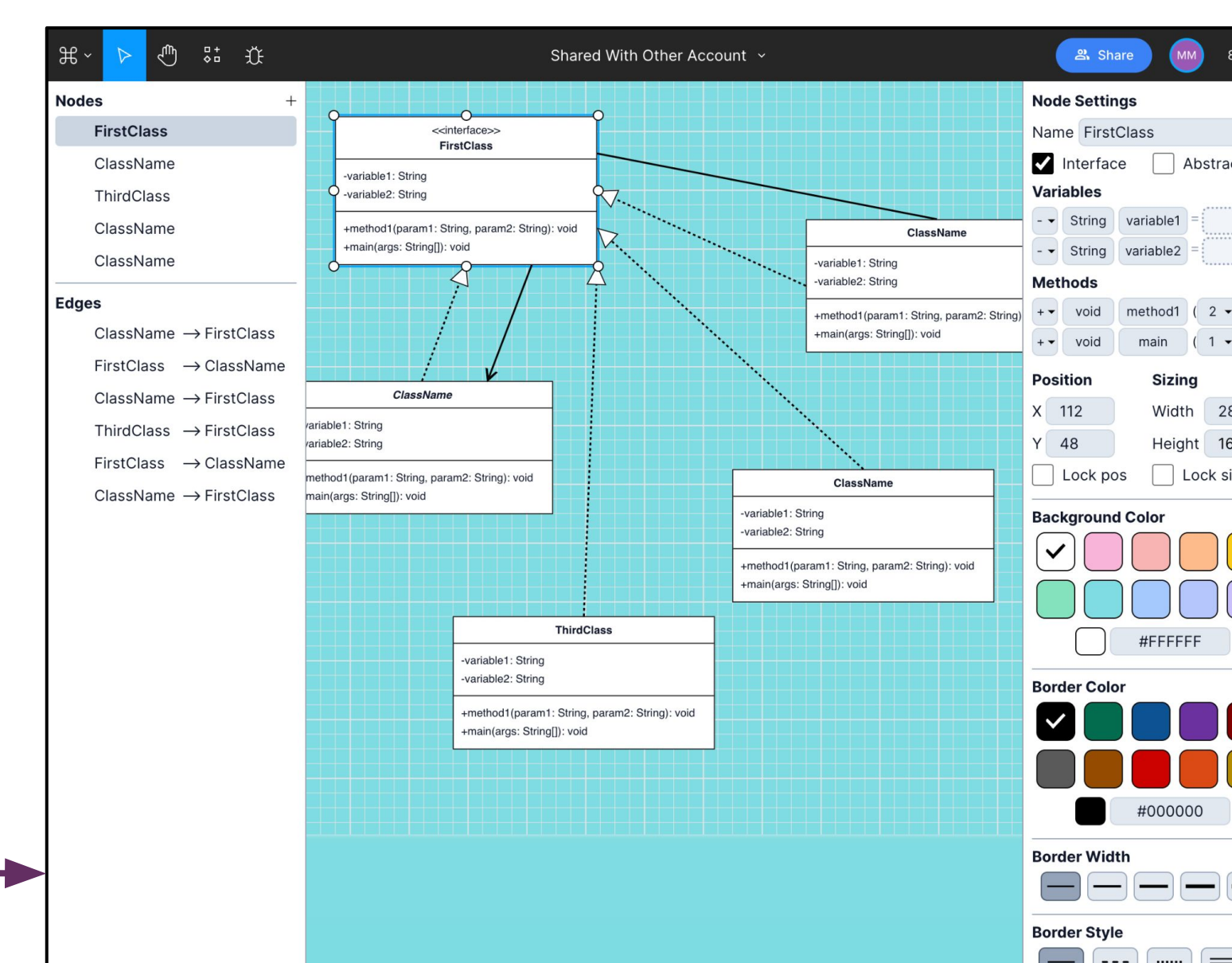


Figure 7: Diagram Editor Page

Figure 7 shows *ProUML*'s diagram editor page, where users can edit a specific diagram. There are many tools on this page, which consist of a left and right sidebar. In the left sidebar, users will be able to see all of the UML diagram objects that are contained within the diagram. If users click on any of these objects, the clicked object will be focused on in the diagram editor so that the user can see the object visually. In the right sidebar, users will be able to see all of the settings to edit the objects in the UML diagram. If a user has not selected any UML object or clicks on the diagram background, the customizable general diagram settings will be shown in this right sidebar, which consist of the grid, zoom, and color settings. If a UML object is clicked, the settings for that specific object will be shown in the right object, which consist of customizable styling and content settings.

BROADER IMPACTS

- ❖ Anyone with an internet-connected mobile or computer device, with a web browser, can access our application and may utilize all of the features of *ProUML*. These users may not have much knowledge in software or UML diagramming at all. However, they can use our application to learn how to construct and manage UML diagrams to learn how the language works and how it relates to software design.
- ❖ Individuals with an knowledge in UML diagramming, including software engineers and students, can take advantage of the full feature-set provided by *ProUML*. These users may have some technical knowledge and require an intuitive interface to use the application to its full potential.
- ❖ Educators can also utilize *ProUML* to educate students about UML, which especially may be beneficial to students who specialize in Computer Science, Software Engineering, or any other studies that deal software design. These users may have high knowledge of UML and software design.

THIRD-PARTY SOFTWARE



REFERENCES

- ❖ AntV X6 Documentation: <https://x6.antv.vision/en>
- ❖ Go Documentation: <https://go.dev/doc/>
- ❖ Next.js Documentation: <https://nextjs.org/docs/getting-started>
- ❖ PostgreSQL Documentation: <https://www.postgresql.org/docs/>
- ❖ React Documentation: <https://reactjs.org/docs/getting-started.html>
- ❖ Redis Documentation: <https://redis.io/docs/>
- ❖ Tailwind CSS Documentation: <https://tailwindcss.com/docs/installation>
- ❖ TypeScript Documentation: <https://www.typescriptlang.org/docs/>