UserManual.md - Grip

Functional Software Requirements Application

Project Submission for SENG 350 - Fall 2018

Matthew Fortier, Tristan Giles, Amritijot Madahar, Geena Smith

The web application described in this manual was designed and built as a project for SENG 350 in the fall of 2018. The objective of this application was to create a project management tool to enable architects to define and manage functional requirements in the form of use cases.

The requirements of the application included the following:

- The app must be web-based and built using TypeScript, Node, Express, MongoDB, and Angular.
- The app must support different user IDs and each user should be able to create, update and delete multiple projects.
- There must be an "admin" user, which cannot be deleted, that has the ability to create or delete other regular user IDs.
- Users can invite other users to view or edit projects, however only the project owner can delete a project.
- Users can surrender project ownership to another user.
- Projects consist of a title, a description, and a set of use cases.
- Each use case is defined in a structure according to the "Cockburn Template".
- The app must provide an overview page that lists all use cases accessible by a user and allows the user to search for use cases.

Set Up

The application source files for this project can be found here

To be able to run the application, the following software dependencies must be installed

- TypeScript (v2.9.2)
- Node (v8.10.0)
- NPM (v3.5.2)
- Express (v4.16.4)
- MongoDB (v3.1.7)
- Python (v2.7.15rc1)

After the installation of these dependencies, the application itself can be installed. First, we must start the database. Open a terminal in the Mongo installation directory and type

```
sudo ./mongod
```

This will start the database and enable us to send and retrieve data.

Next, we build the application. Open a terminal in the application installation directory and type:

```
npm install --save
```

This will install all node dependencies and will also run a full build of the Angular modules.

Next, we fill the database with startup data. In the same terminal as before, enter the command

```
python scripts/dbinit.py
```

followed by

```
python scripts/dbpopulate.py
```

which will initialize our database to work with our application. (NOTE: We are using Python 2 for these scripts, if any errors arise from uninstalled python dependencies, running a pip install should resolve any issues)

Once the database is set up, and we have built the application, we can start the application:

- Start the database by typing

```
sudo ./mongod
```

in the Mongo installation directory

- Launch the application by typing

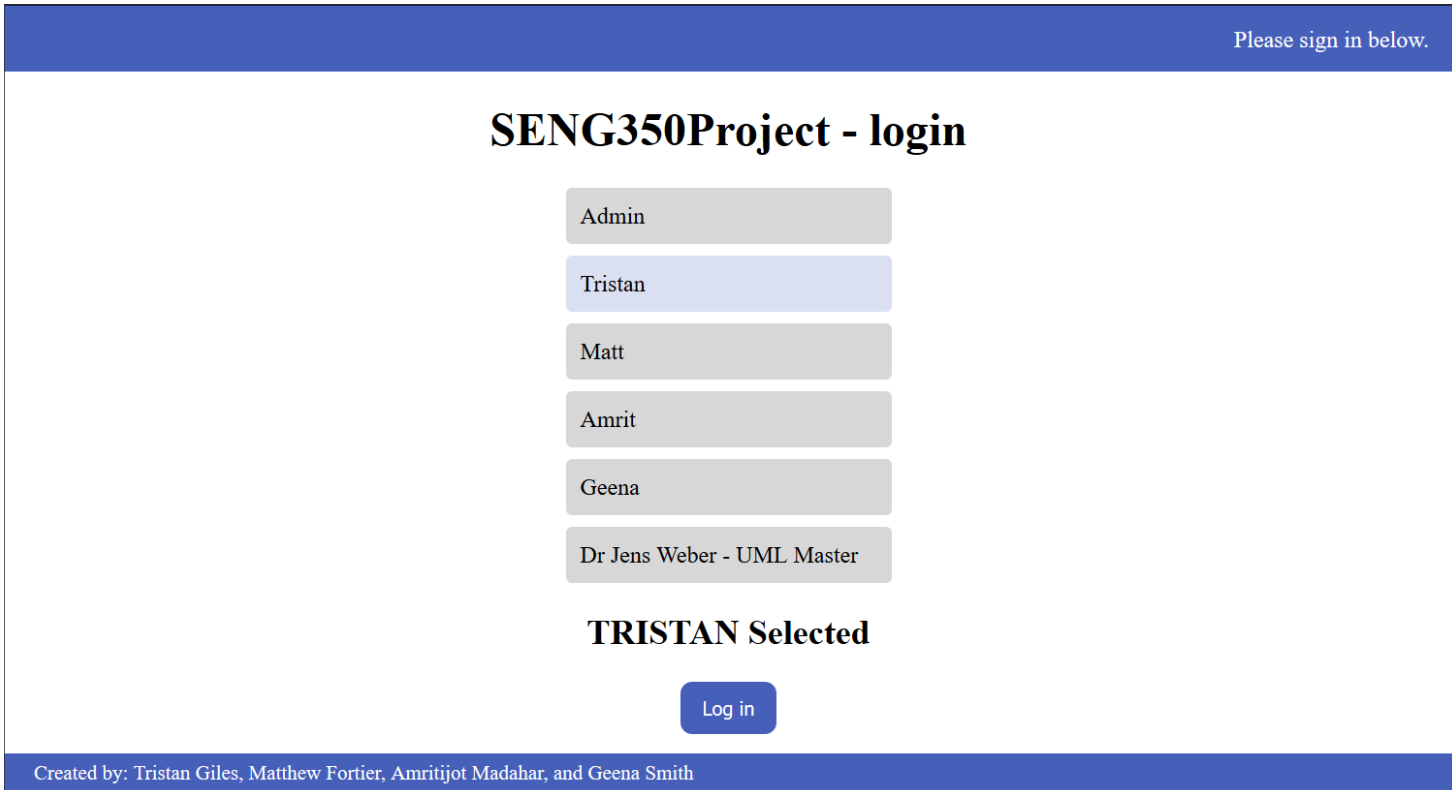
```
node server
```

in the application installation directory.

The application will now be running on localhost:3000

Logging in

After starting the application and navigating to the above link, the user will be prompted to login as a user. To log in as a user, select the user from the list and click the "log in" button, as shown in the below image.



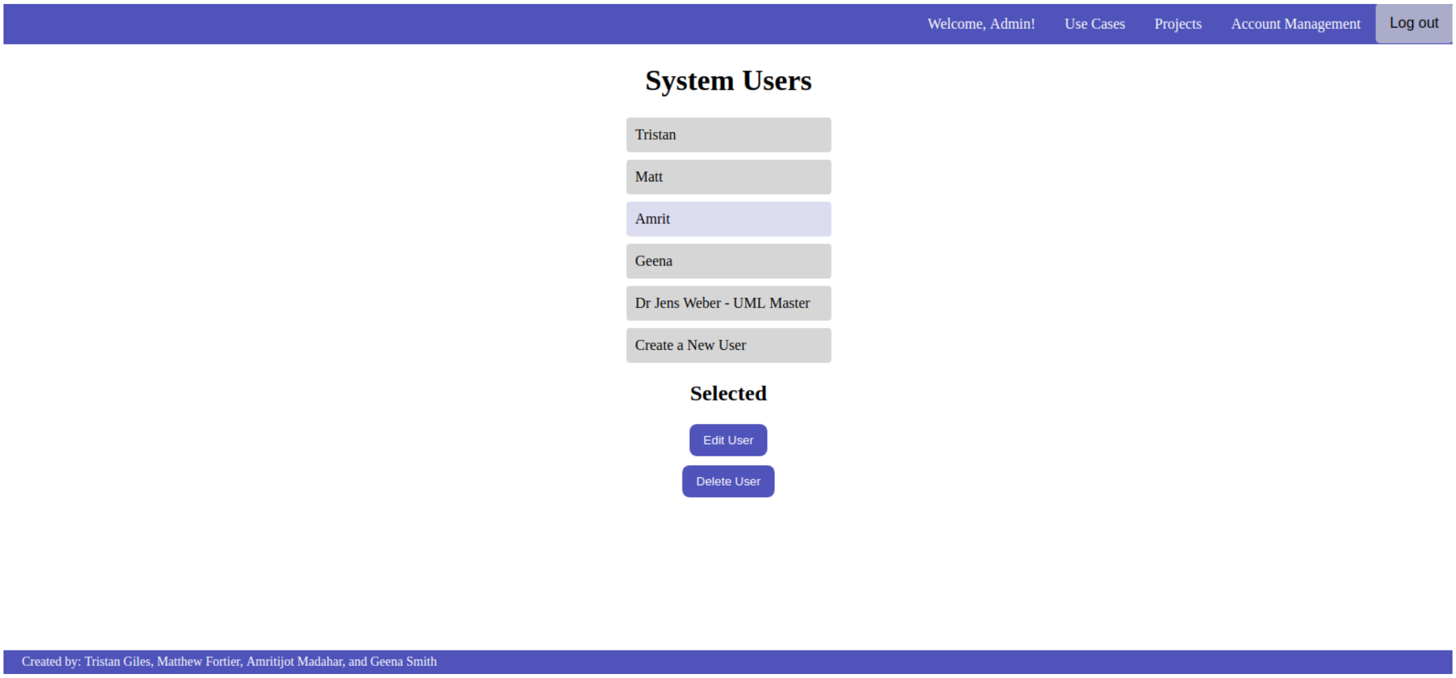
After logging in, the user will be redirected to an overview page. On this page will be all use cases that the current user has any access rights to (read/write/owner).

Logging Out

Once logged in, a user can log out at any time by simply selecting the "log out" button in the upper right hand corner, pictured below.

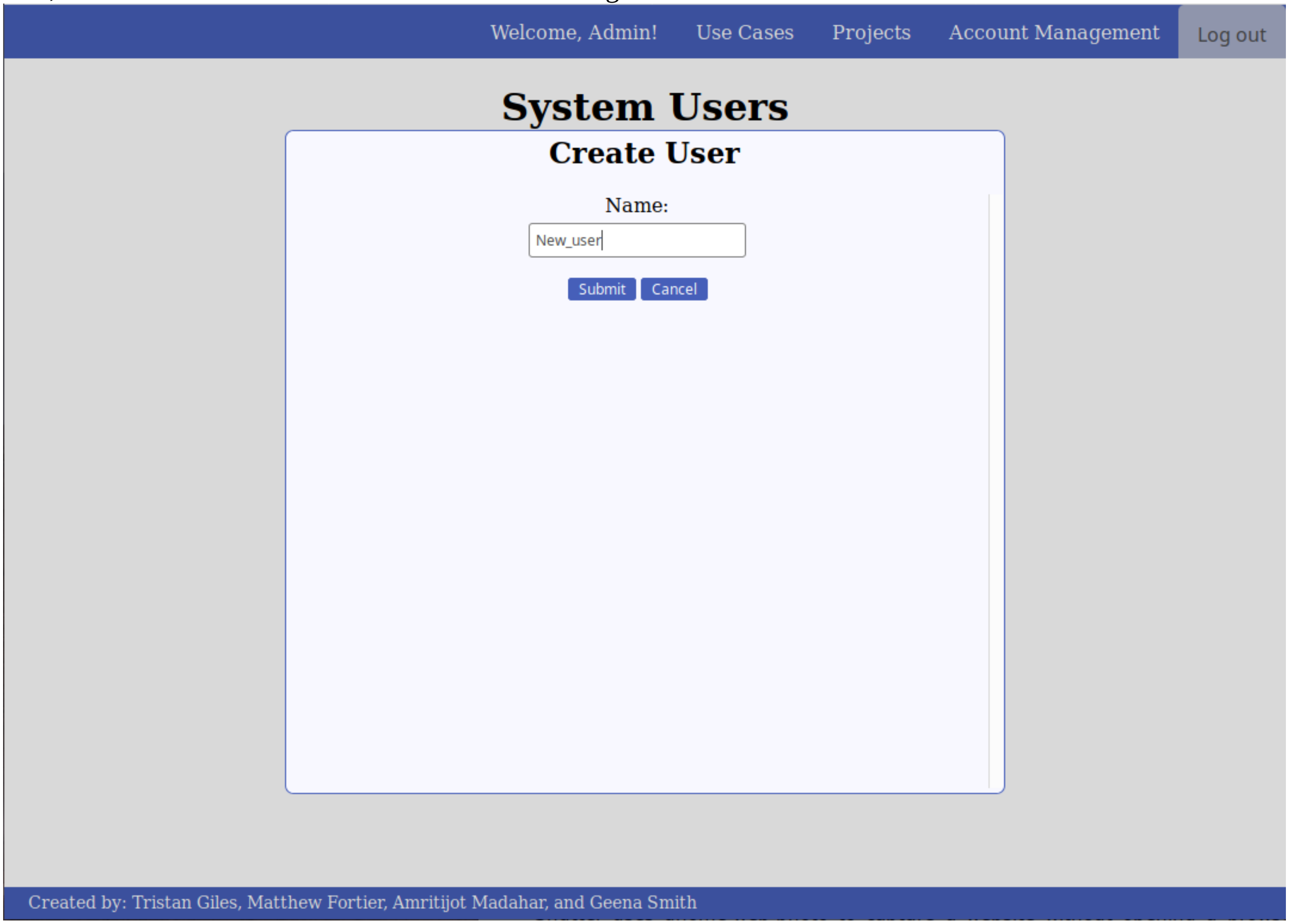
Admin Panel

When logged in as the "Admin" account, the user has access to administrator functionality. This can be accessed by clicking the "Account Management" button in the navigation bar. The admin panel is shown below.



Creating a User

To create a new user, the user must first be on the admin panel. Selecting "Create a New User" will cause a modal to appear. The admin will be prompted to give the new user a name. Submitting this will create the new user, and enable new users to select this account at login.



Deleting a User

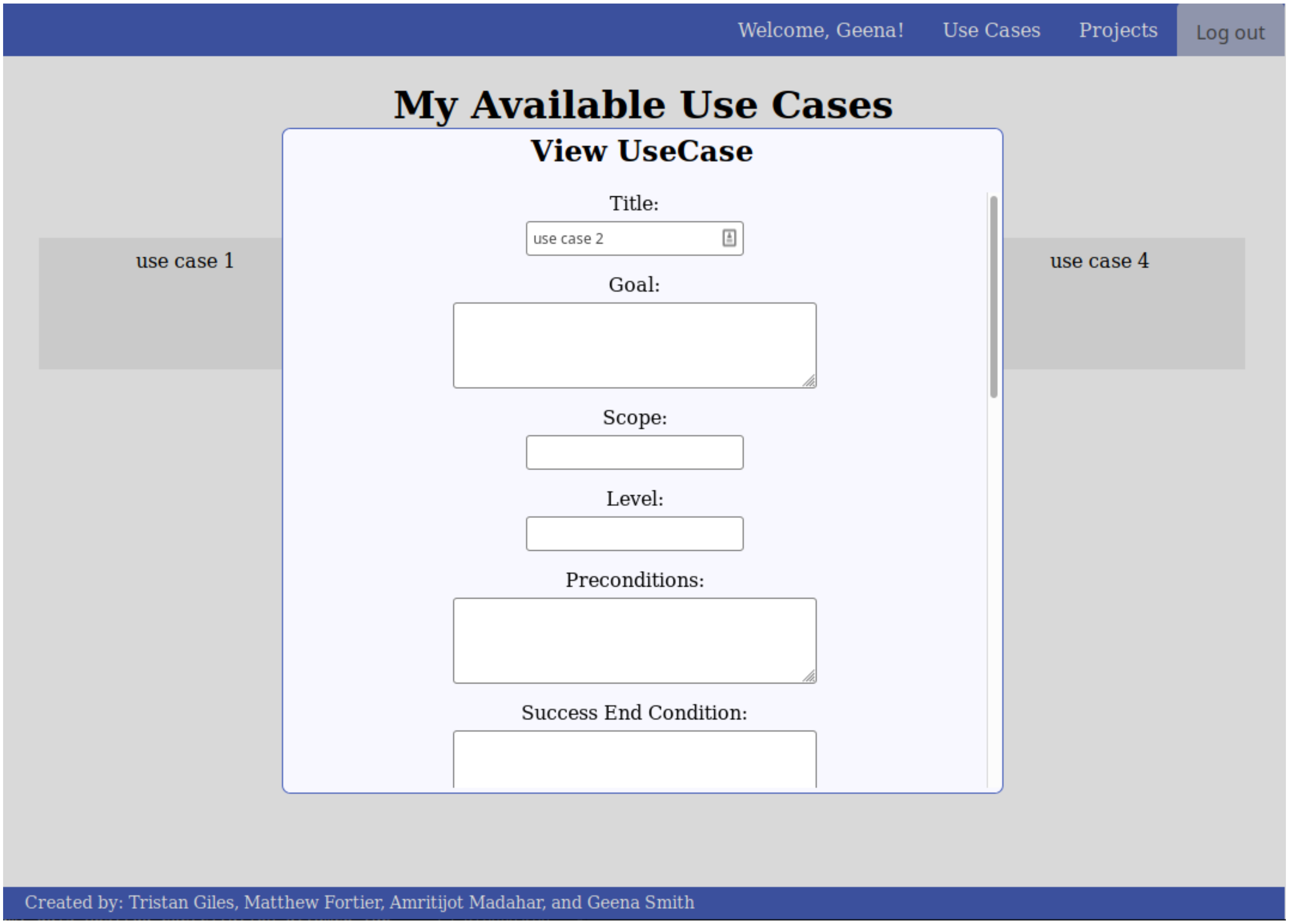
The process for deleting a user is similar to creating a user. From the admin panel, select a user to delete. After clicking the delete button, a confirmation modal will appear, and will allow the user to either confirm or cancel the deletion. The former will remove the user and all owned projects, and the latter will cancel the action and close the modal. This is shown below.

Use Case Home View

To access the use case home view, the user must first be logged in. The user then selects the "Use Cases" button from the top navigation bar. This will bring the user to the page where they can view all use cases for all projects they have access to. This view is pictured below.

Viewing a Use Case

Viewing a use case can be done in two ways. The first way is to navigate to the use case home. All available use cases will be displayed to the user. The user selects the use case they wish to view, then selects "View Usecase". A modal will appear displaying the use case information to the user. This modal is pictured below.



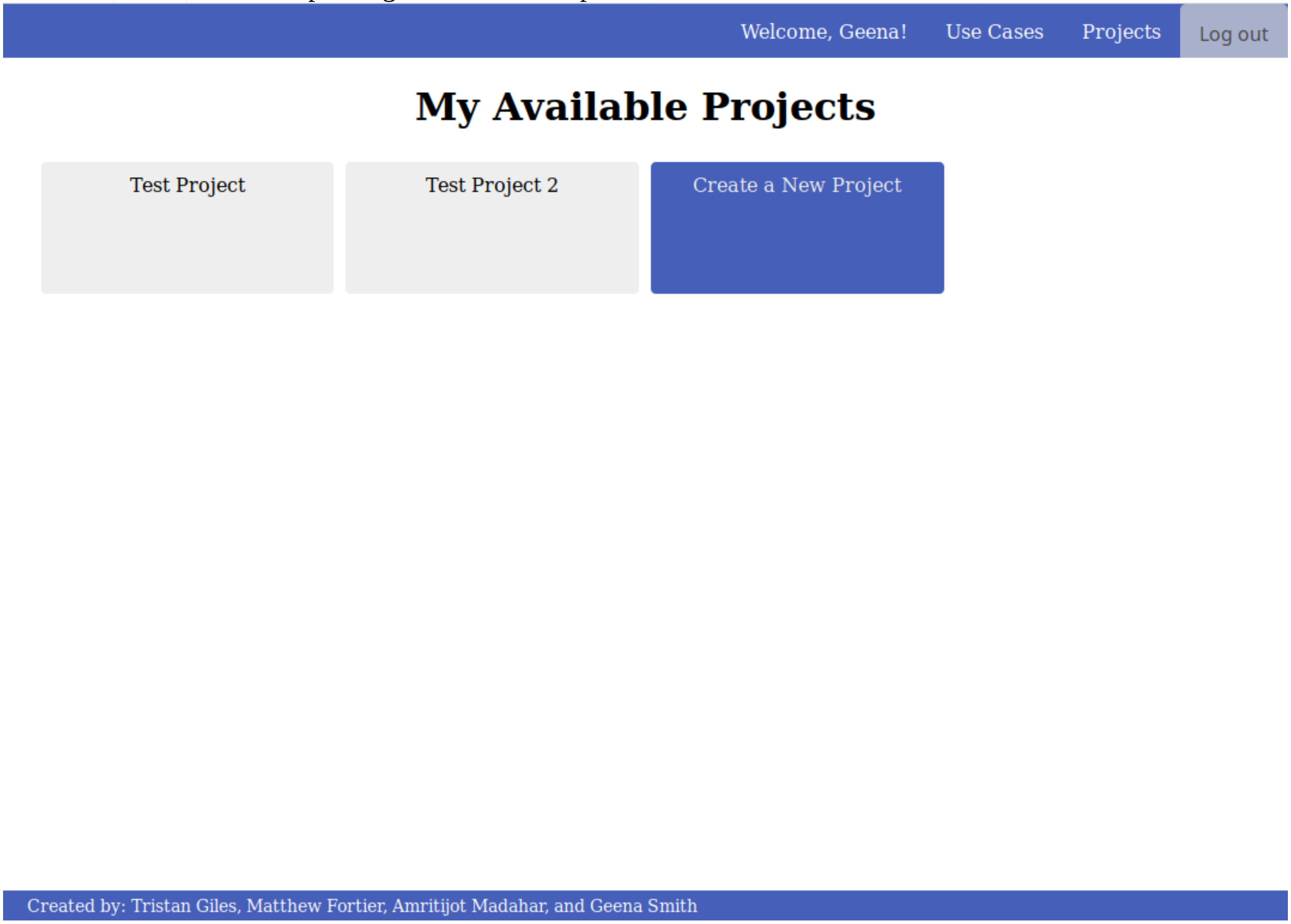
The second way to view a use case is cocvered under "Editing a Use Case" below.

Searching for a Use Case

To search for a use case, the user must be on the use case home view. Located at the top of the page is a search bar. As the user types a title into the search bar, all use cases not matching the text are filtered out. This functionality is pictured below. The search bar is currently case-sensitive, though this is subject to change in future versions.

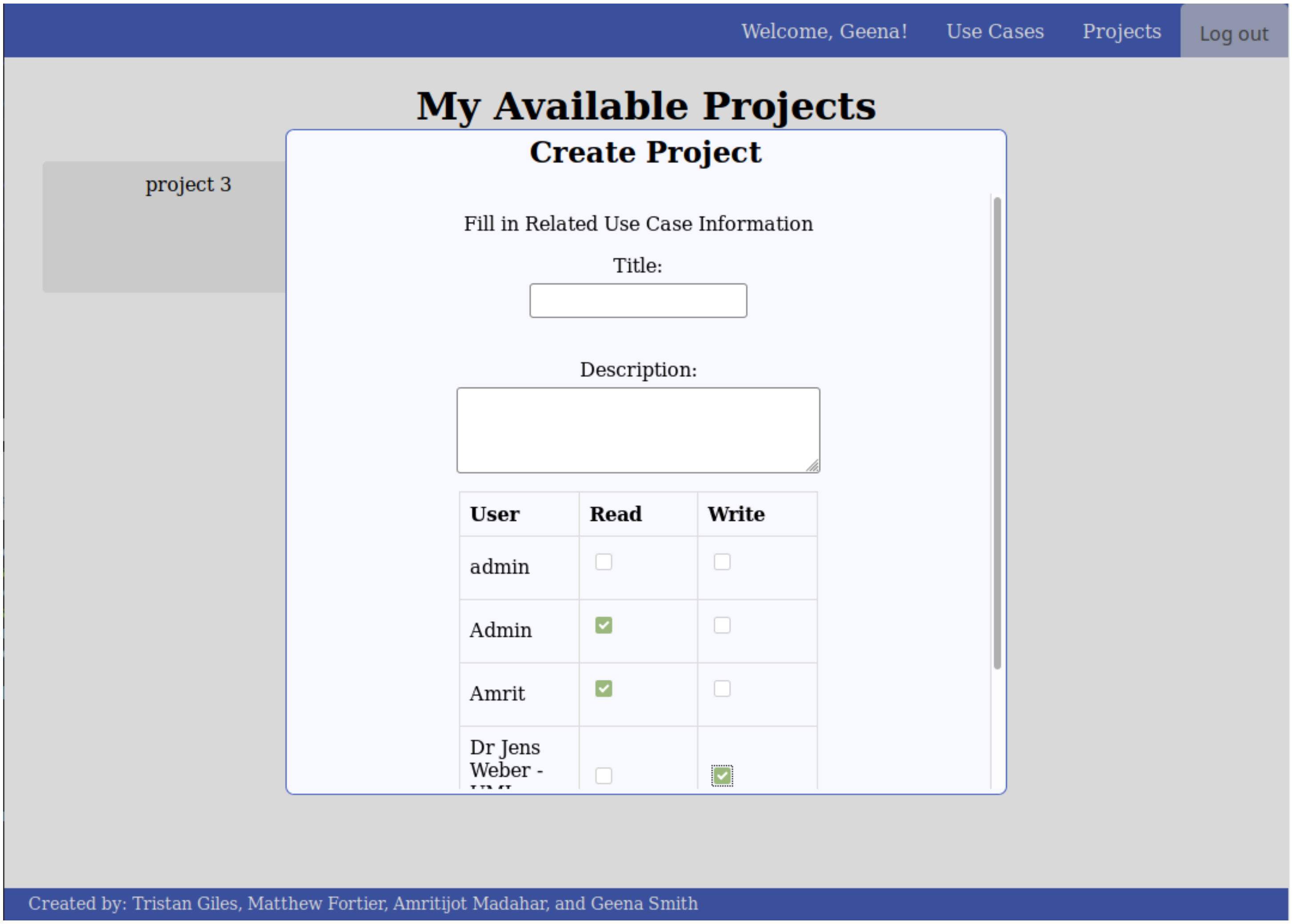
Project Home View

To access the project home view, the user must first be logged in. The user then selects the "Projects" button from the top navigation bar. This will bring the user to the page where they can view all projects for which they have read, write, or owner privileges. This view is picture below.



Creating a Project

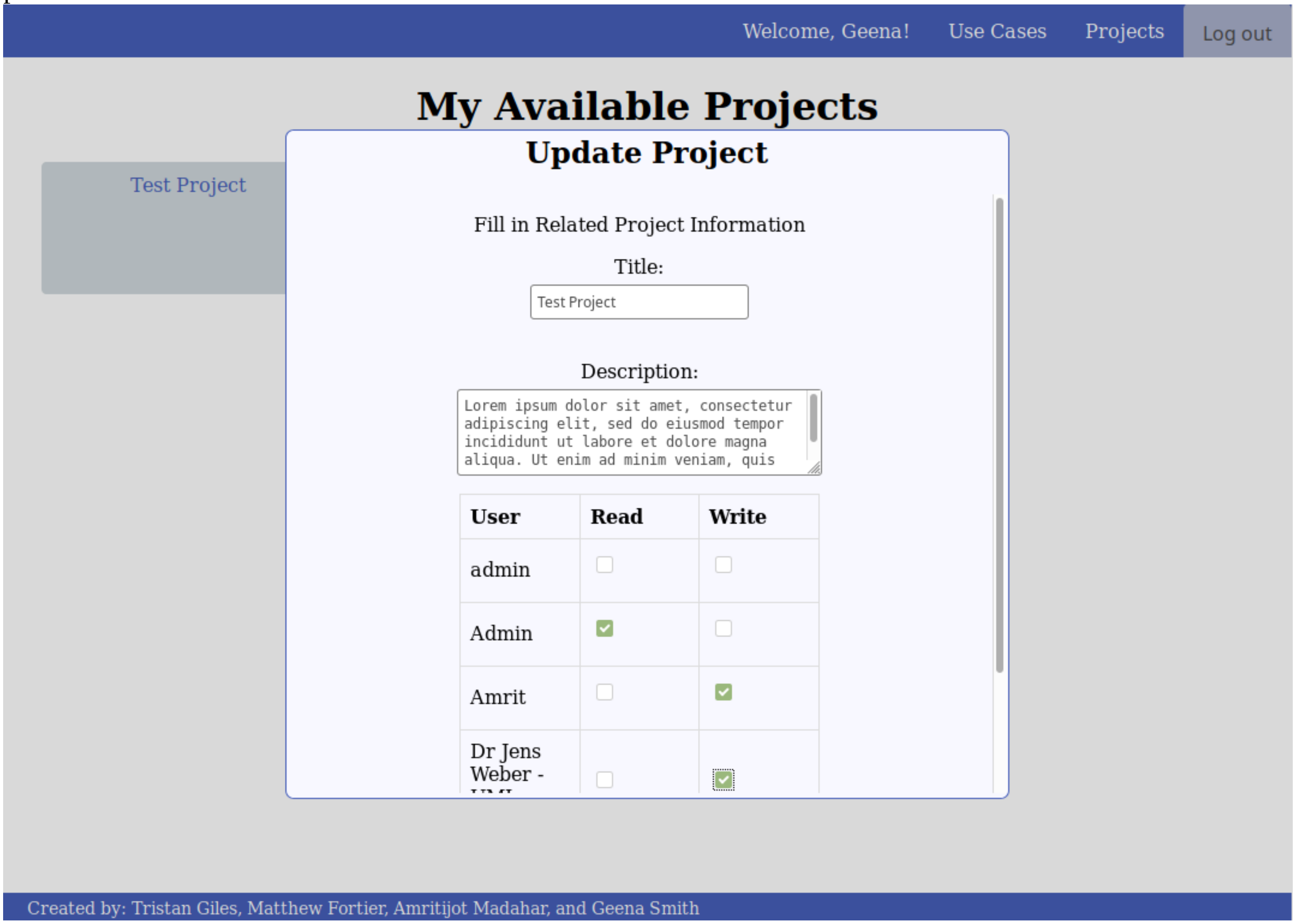
To create a project, the user must be on the project home page. The user then selects "Create a new project". This will display a modal with a form to enter the information for a new project. Fill in the title of the project, a project description, and select the project permissions for other users (read/write). Once this information is entered, select the "submit" button at the bottom of the modal.



This will create the new project, and will now be visible in the projects overview page for all those with permissions to either view or edit.

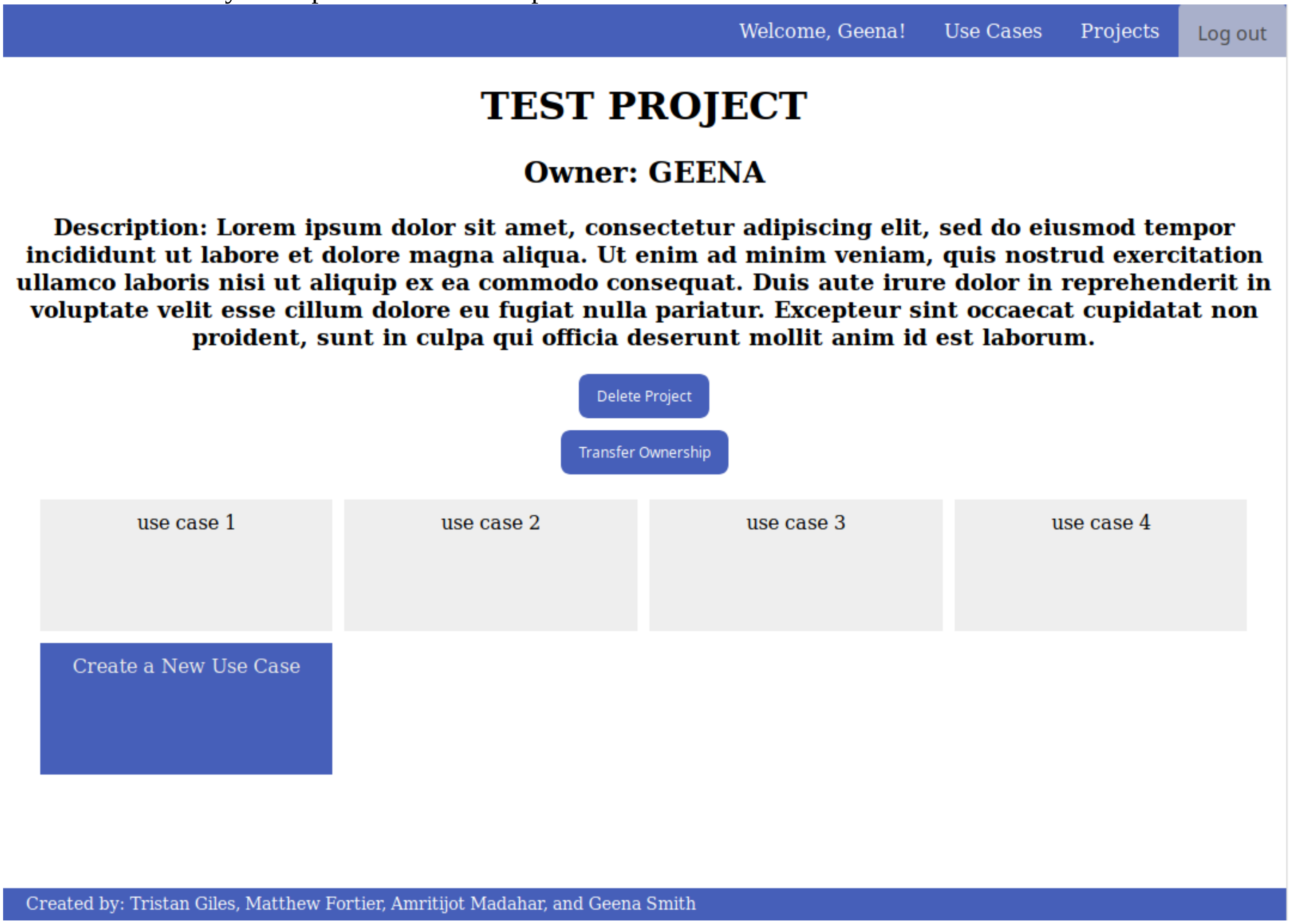
Editing a Project / Changing Project Permissions

To edit a project, the user must be on the project home page. Select the desired project, then select the "edit project" button. A modal will open on the current page with the project information. From this modal, the user can update the project name, description, and, if the user is the project owner, change permissions. This modal is pictured below.



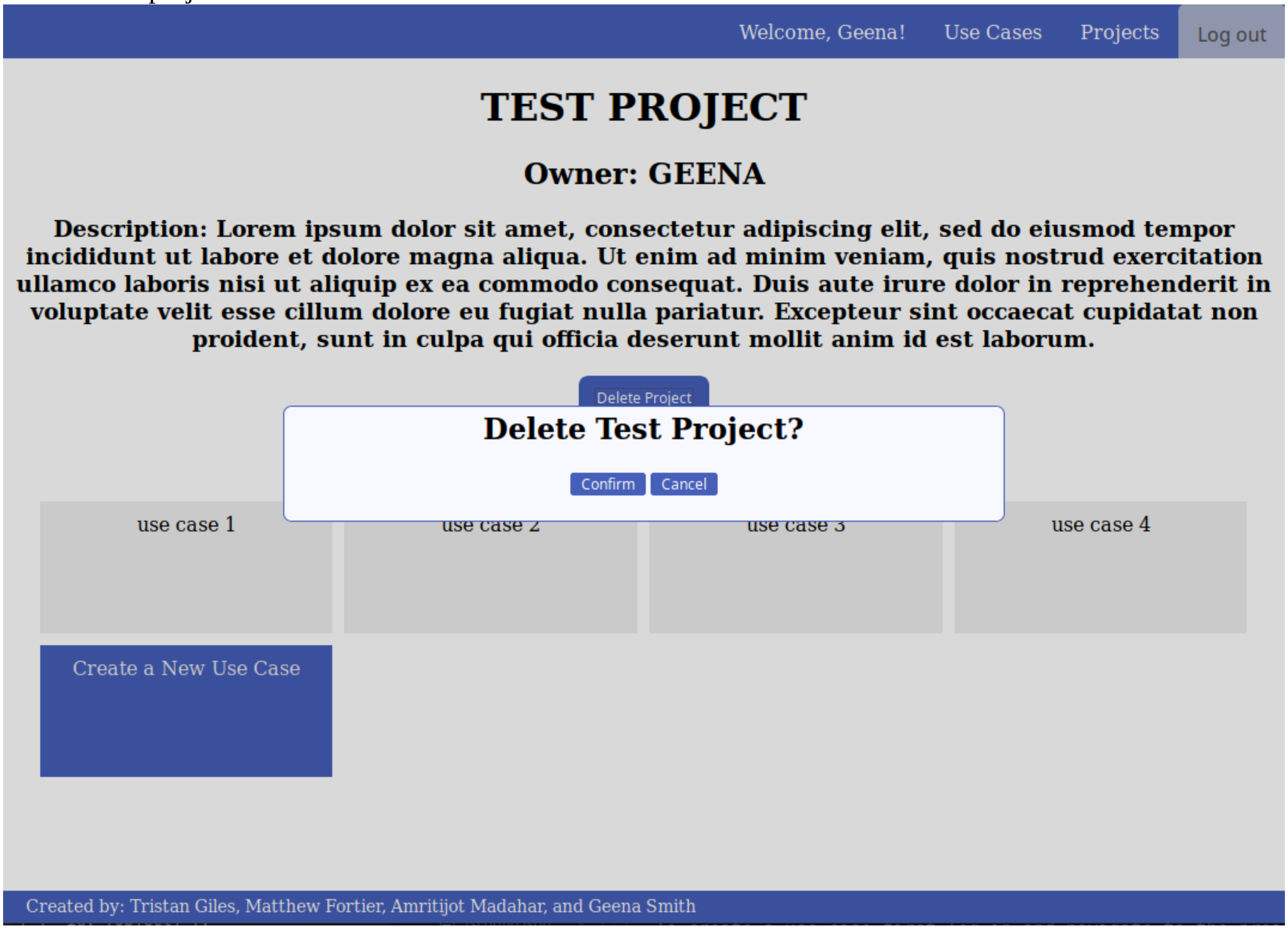
Project View

To view the details of a particular project, start from the Project Home View as discussed above. Select a project in the home view, then select "View Project". This will bring the user to the specific page for that project, where various functionality takes place. This view is pictured below.



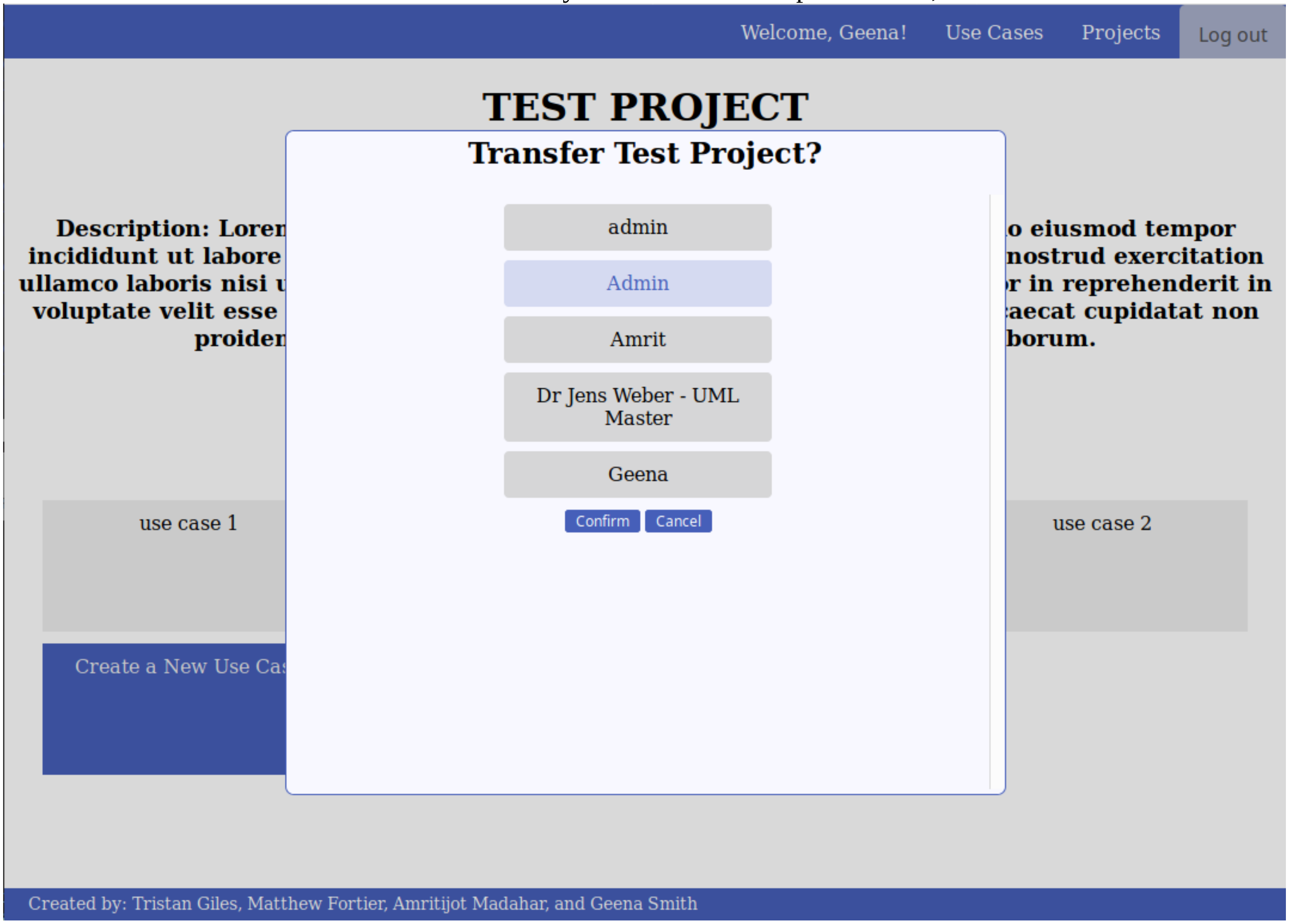
Deleting a Project

From the project view, if the user is the owner, there will be a button labeled "Delete Project". To delete a project, select this button. A modal will appear asking the user to confirm the project deletion. Select "confirm" to delete the project.



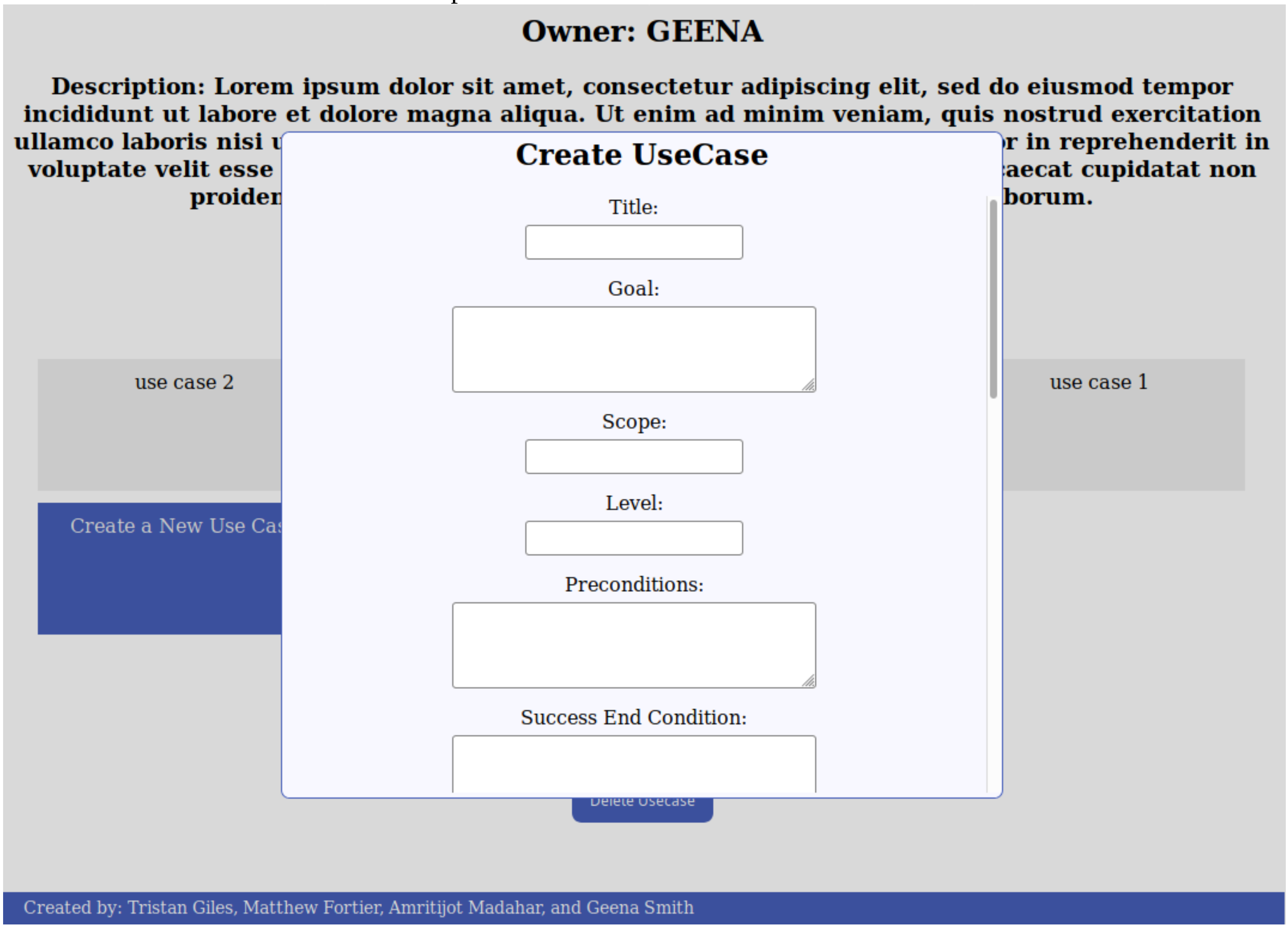
Changing Project Ownership

From the project view, if the user is the owner, there will be a button labelled "Transfer Ownership". To change a project's owner, select this button. A modal will be displayed on the current page to select the new owner. Select the new owner from the list of users who currently have read or write permissions, and click confirm.



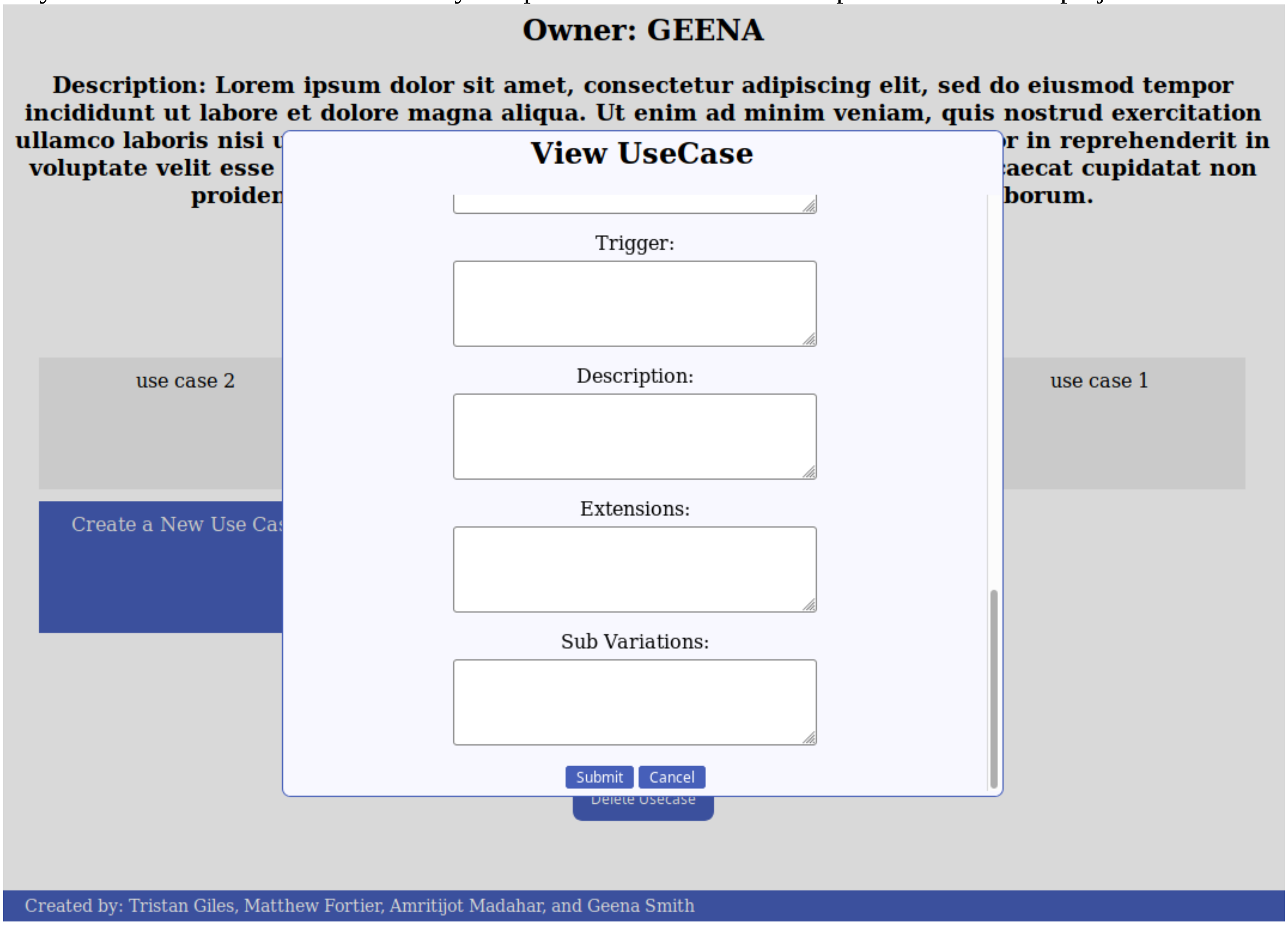
Creating a Use Case

To create a use case, first log in and navigate to the project view. If the user has "write" permission on the project, there will be a large blue button labelled "Create a New Use Case". Select this button. This will open a modal with a form to enter information for the new use case. Once the information is filled in, select "submit" at the bottom of the modal. This modal is pictured below.



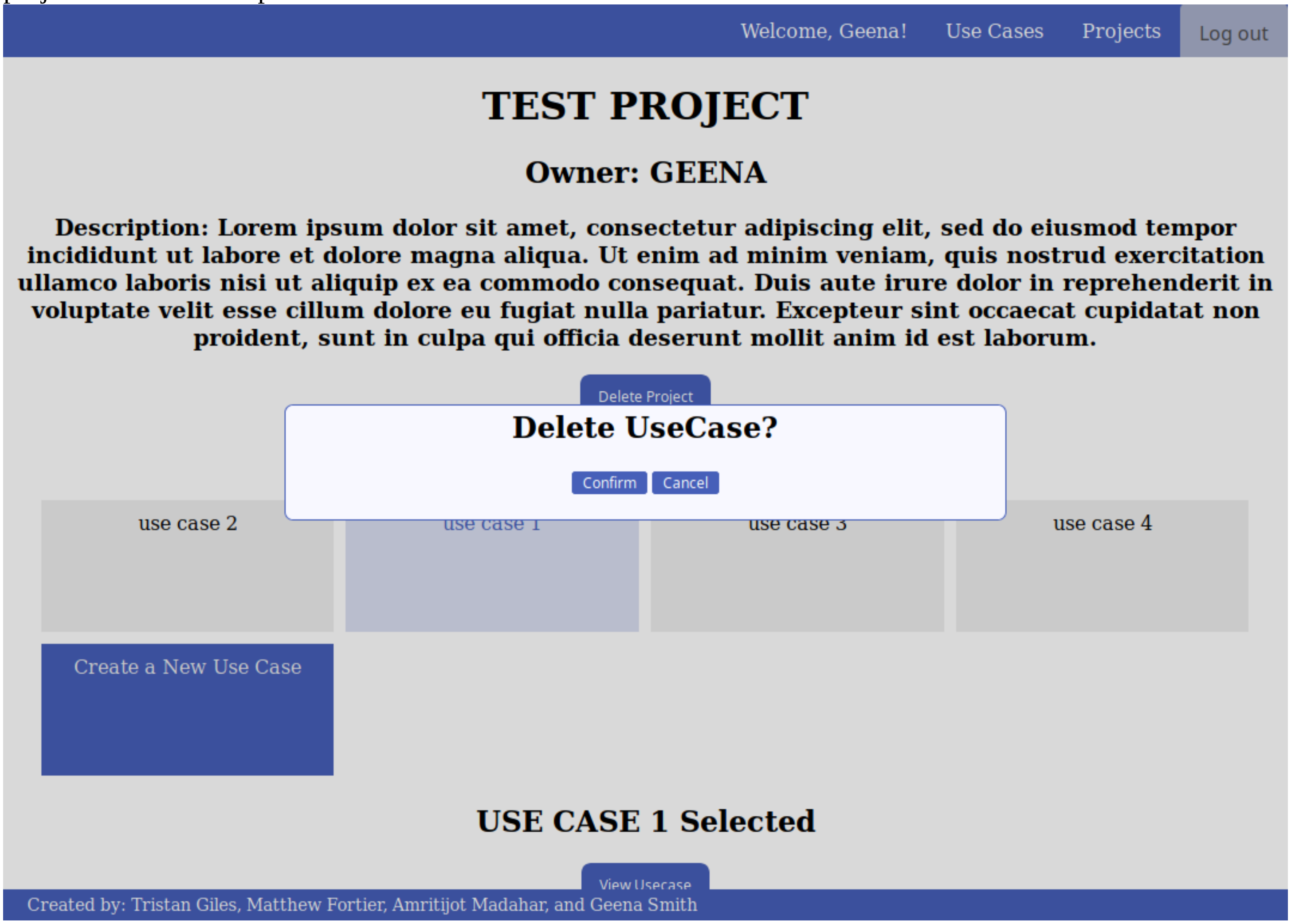
Editing a Use Case

To edit a use case, first log in and navigate to the project view. Select the use case you wish to edit, and select the "View Use Case" button at the bottom of the screen. This will present a modal with all the use case information. Edit the information as desired and select the "submit" button. Note that the "submit" button will only be visible and the use case will only be updated if the user has write permissions for the project.



Deleting a Use Case

To delete a use case, first log in and navigate to the project view. Select the use case you wish to delete. If the user has "write" permission on the project, there will be a "Delete Usecase" button at the bottom of the screen. Selecting this button will present a delete confirmation modal. Select "confirm" to delete the usecase from the project. This modal is pictured below.



Known Problems and Limitations

After delevoping the application, we have noted the following problems and limitations:

- Use case structure used in project may not be comprehensive enough for all users. The form for creating a new use case does not allow the users to add their own custom additional information (aside from the fields in the form) and makes the use of the application somewhat limited to what we've determined to be enough.
- Duplicate naming. Currently, the application allows users with duplicate names, and projects with duplicate names. Calls to the API use the object ID to distinguish between objects rather than the name. To remedy this, we would either add a unique identifier to duplicate names, or prevent users from even creating an account whose name already exists in the system.
- The application is not scalable with its current structure. The application uses client side rendering and processing which means the data displayed on the page is only visible once it has completely been received. This makes the application appear slow at times, since the data can arrive slower than the browser takes to reload the page.
- When creating users/projects/use cases the new object does not always appear on the page instantly. This is because of how asynchronous calls are handled in the API. The response is often sent before the database call is fully resolved; therefore, new information is not always immediately available.