**Team project**

**Justin Rose, Joshua Gosnell, Seth Dean**

EKU **| October 10, 2018**

GitOut

Table of Contents

[I. Introduction 2](#_Toc528525368)

[II. System Description 2](#_Toc528525369)

[III. System Requirements 3](#_Toc528525370)

[A. Functional Requirements 3](#_Toc528525371)

[B. Non-Functional Requirements 5](#_Toc528525372)

[IV. Use Case Diagram: 6](#_Toc528525373)

[V. Domain Diagram: 7](#_Toc528525374)

[VI. Sequence Diagram 8](#_Toc528525375)

[A. Register 8](#_Toc528525376)

[B. Login 9](#_Toc528525377)

[C. Create A Project 10](#_Toc528525378)

[D. Upload 11](#_Toc528525379)

[E. Download 12](#_Toc528525380)

[F. Logout 13](#_Toc528525381)

[VII. State Diagram 14](#_Toc528525382)

[VIII. Activity Diagram 15](#_Toc528525383)

[IX. Database Design 15](#_Toc528525384)

[A. ER Schema 15](#_Toc528525385)

[B. Table Schema 16](#_Toc528525386)

[X. Conclusion 16](#_Toc528525387)

[XI. Data Dictionary: 16](#_Toc528525388)

# Introduction

In this project, we were tasked with creating a program that takes the interface of GitLab and makes it simpler for the user to upload, download, and interact with Git projects. With GitLab, you must use the command line when you first create a project or edit a project. What we hope to accomplish is simplifying this aspect to allow any user to interact with the project.

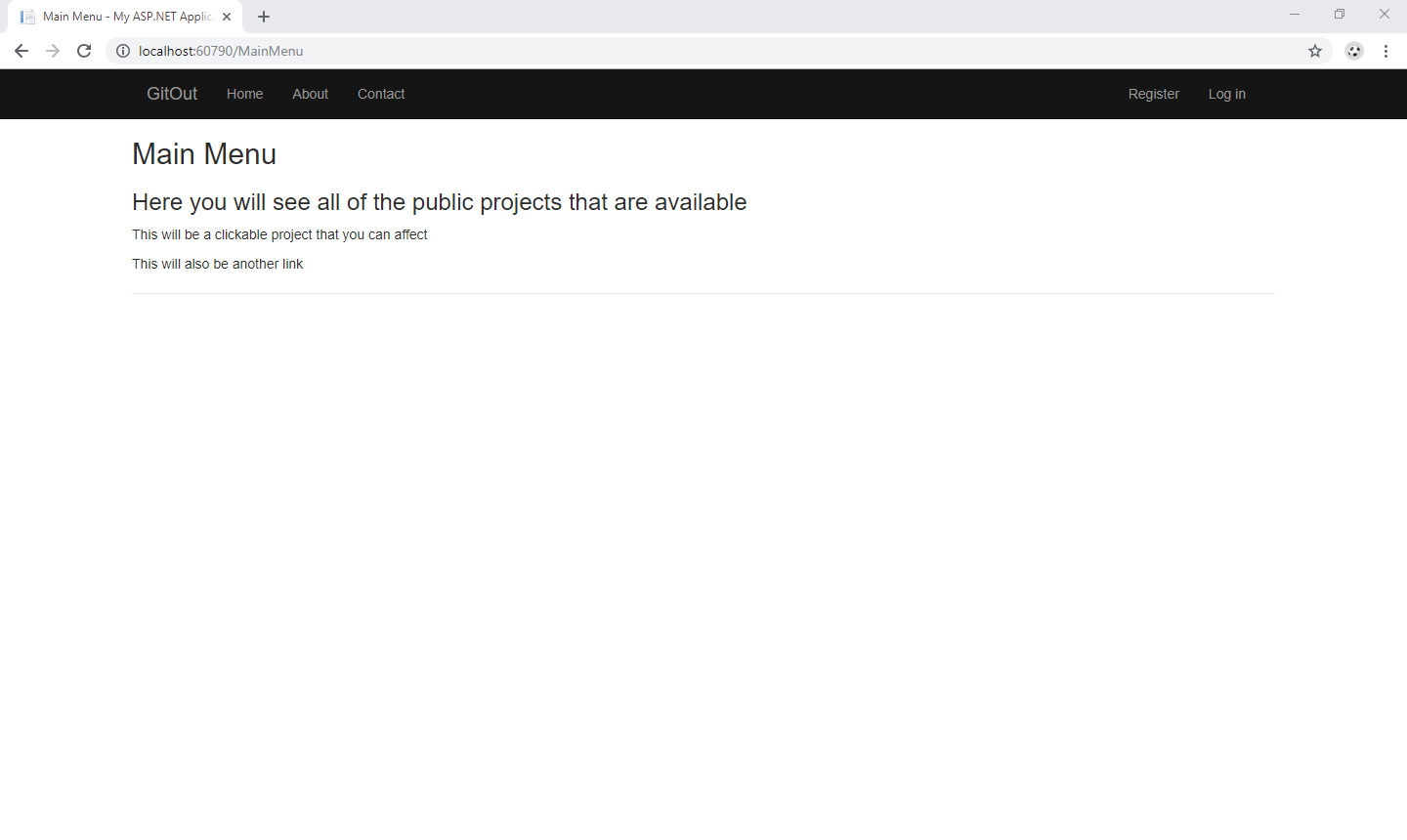
# System Description

Create a system that allows a user to interact with the Git interface with ease. The user can create an account with the system, which will allow them to log in and access projects. The user will have to the option to upload their own files to GitOut, allowing others to collaborate on the project as well as being added to other projects in a similar manner.

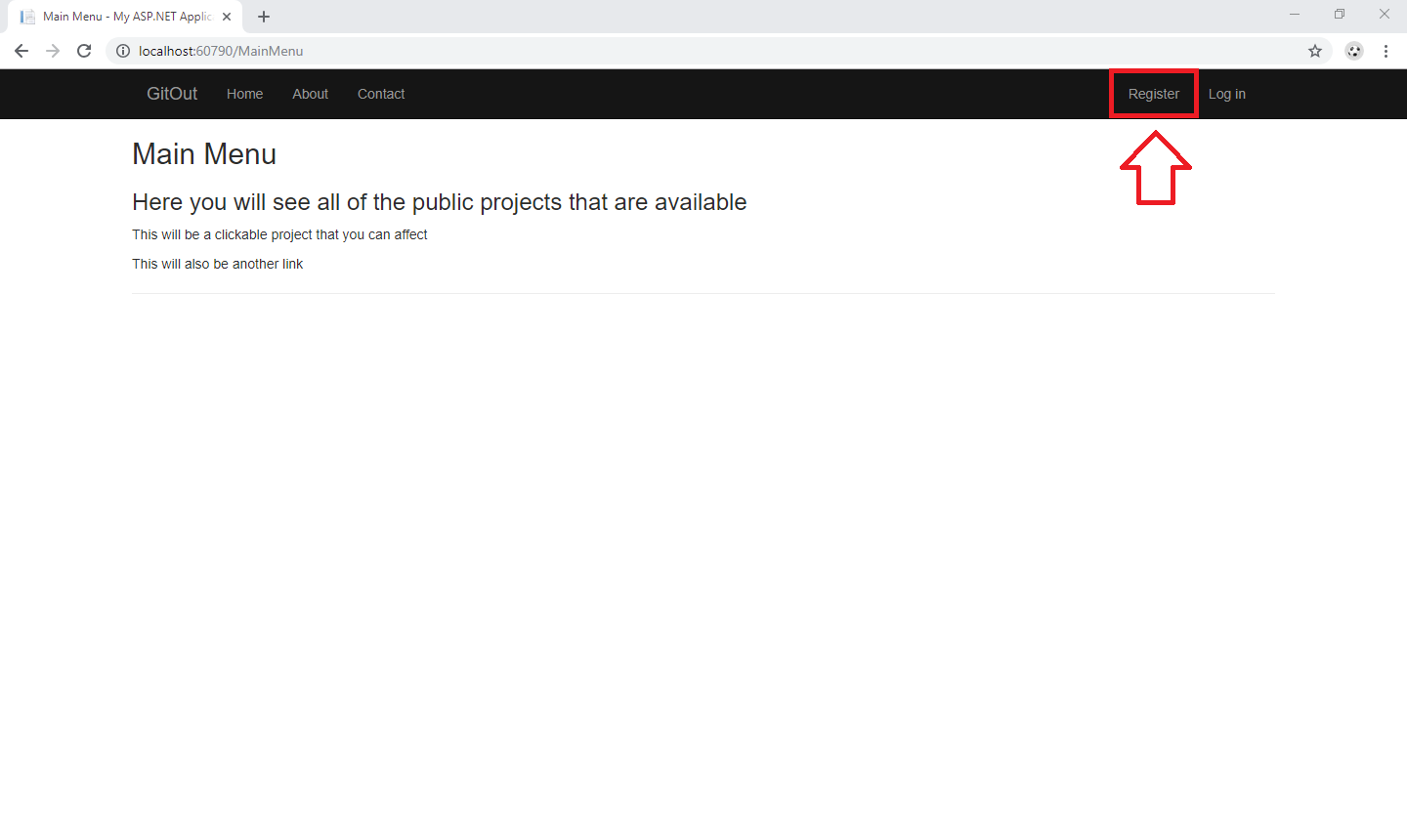
# System Requirements

## Functional Requirements

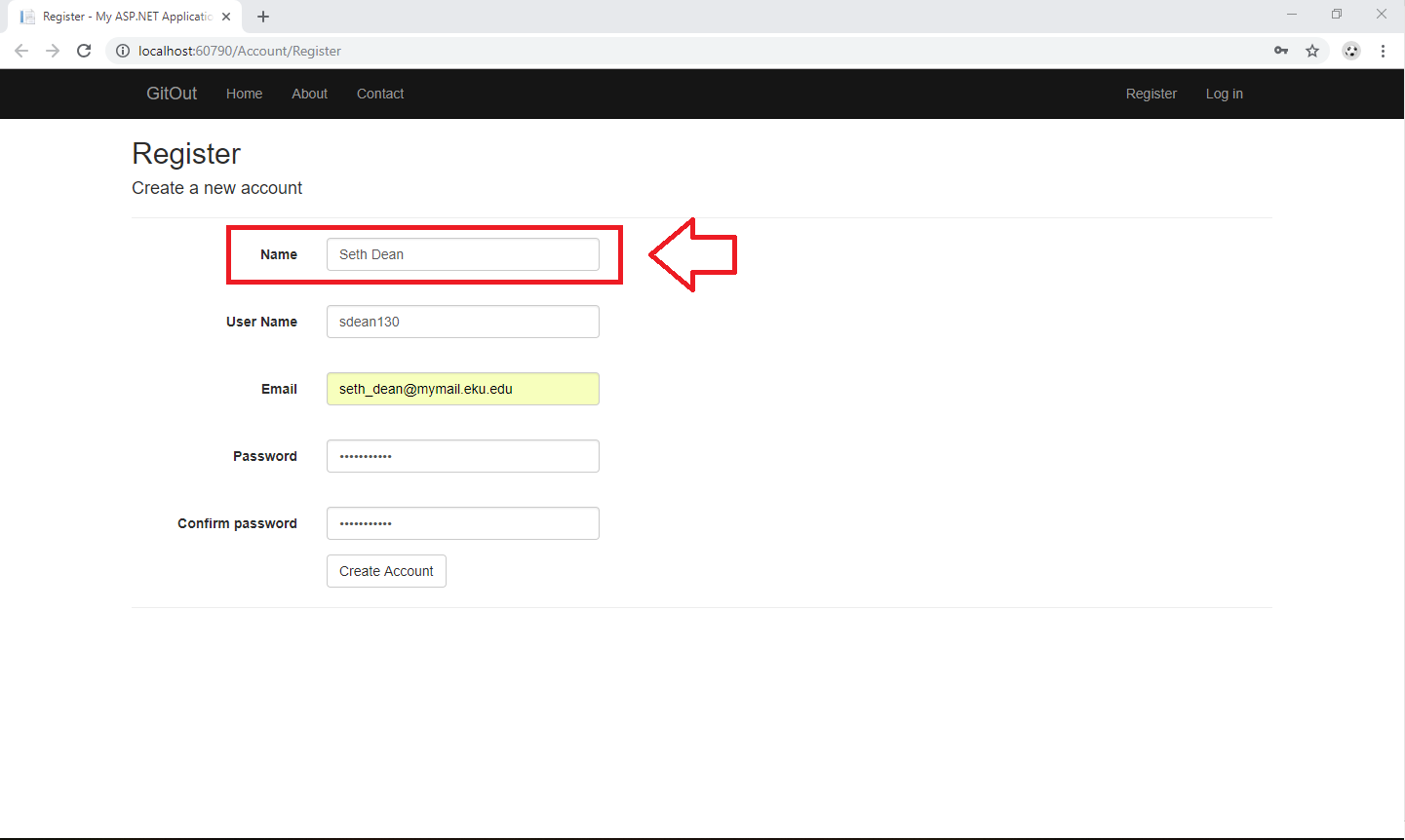
R1. The user shall create a new account



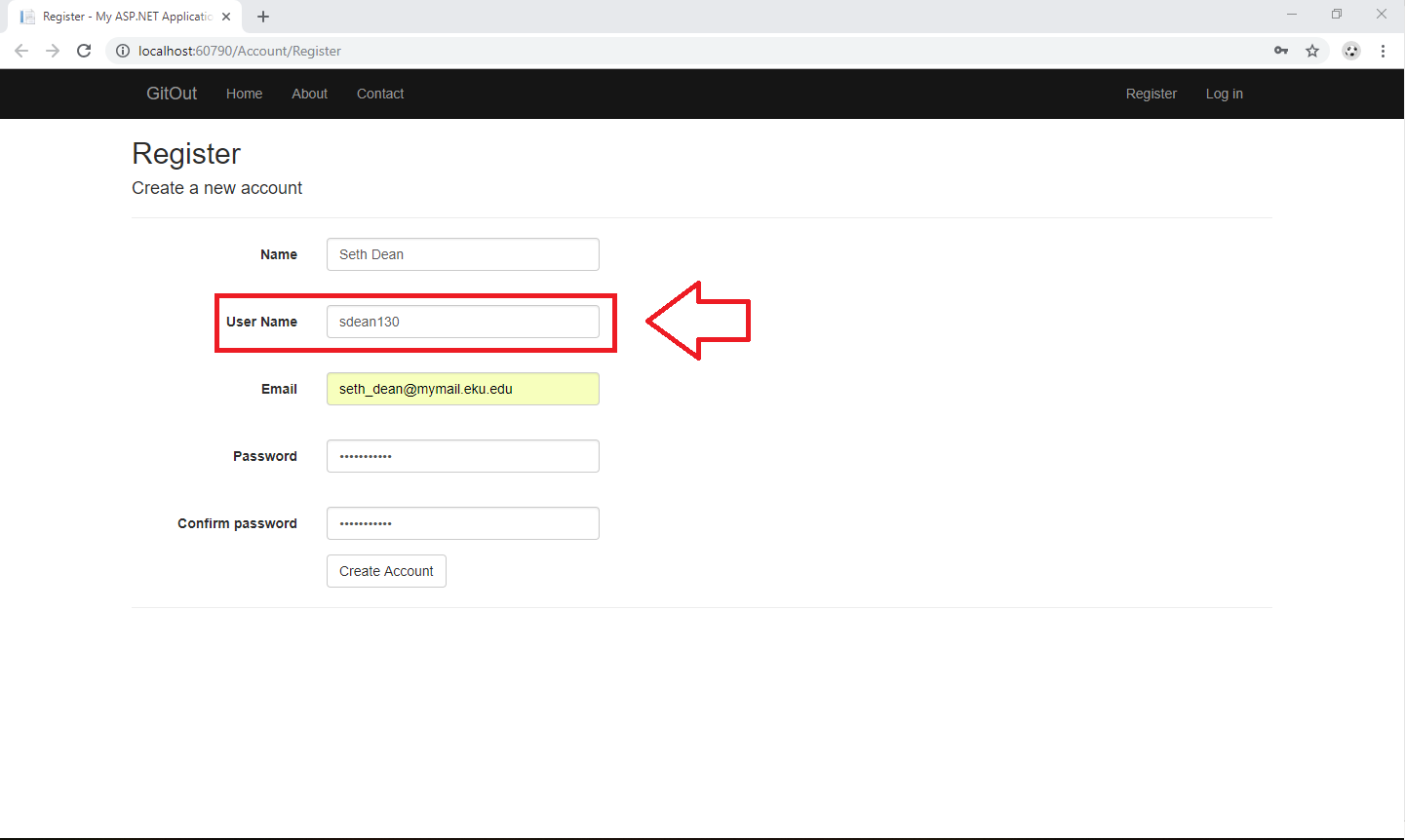
R1.1. The user shall create an account by clicking the register button



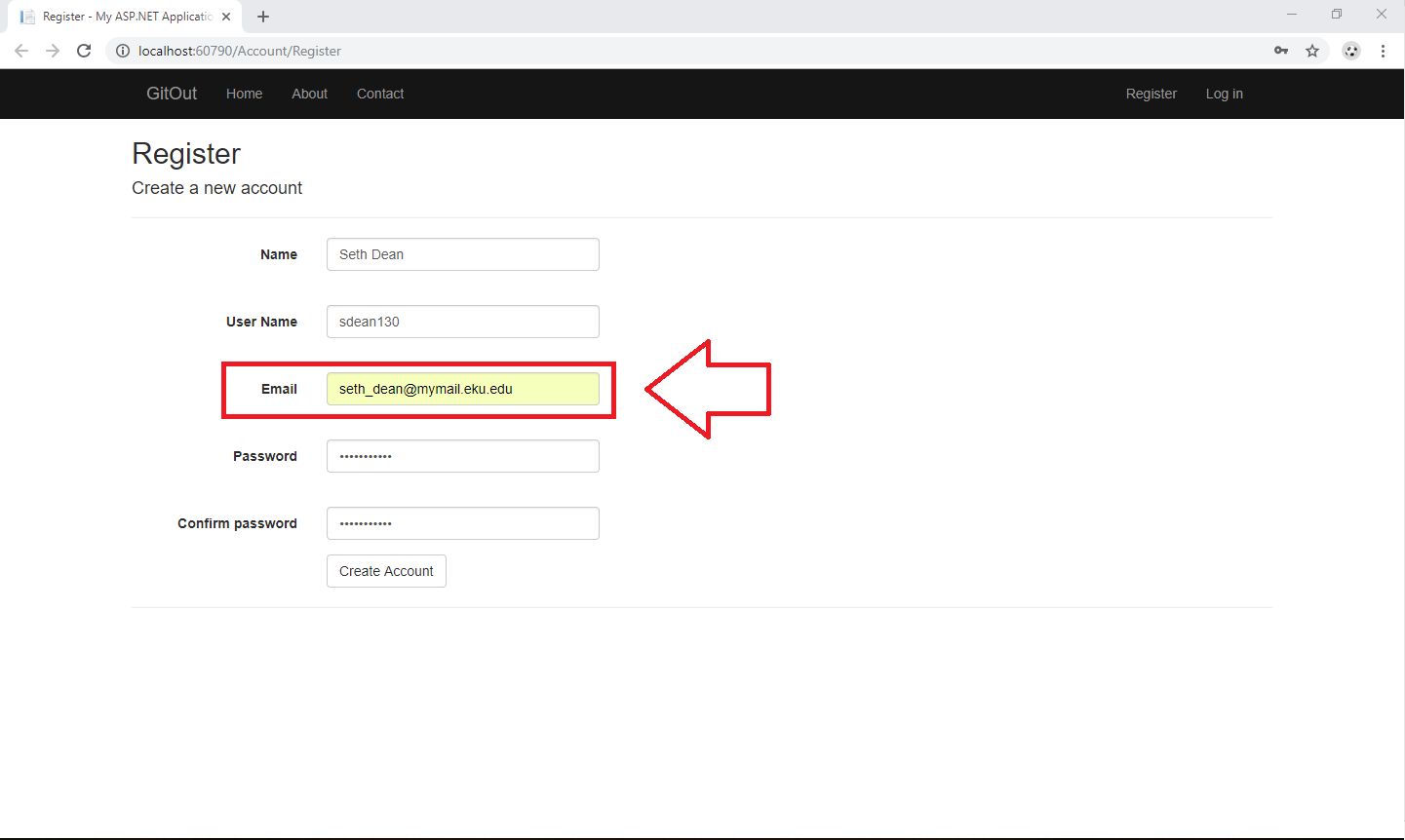
R1.2. The user shall enter a name



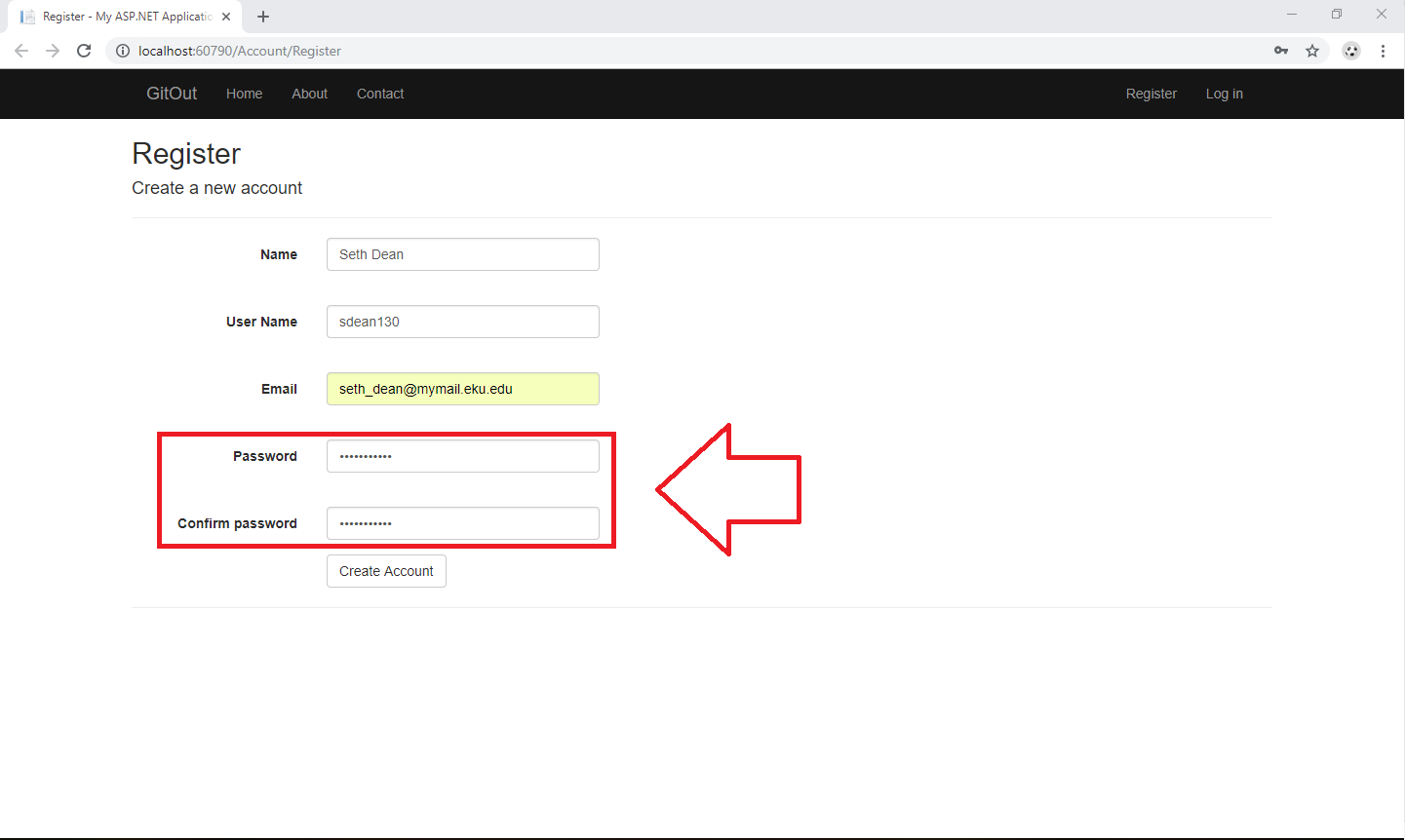
R1.3. The user shall enter a username



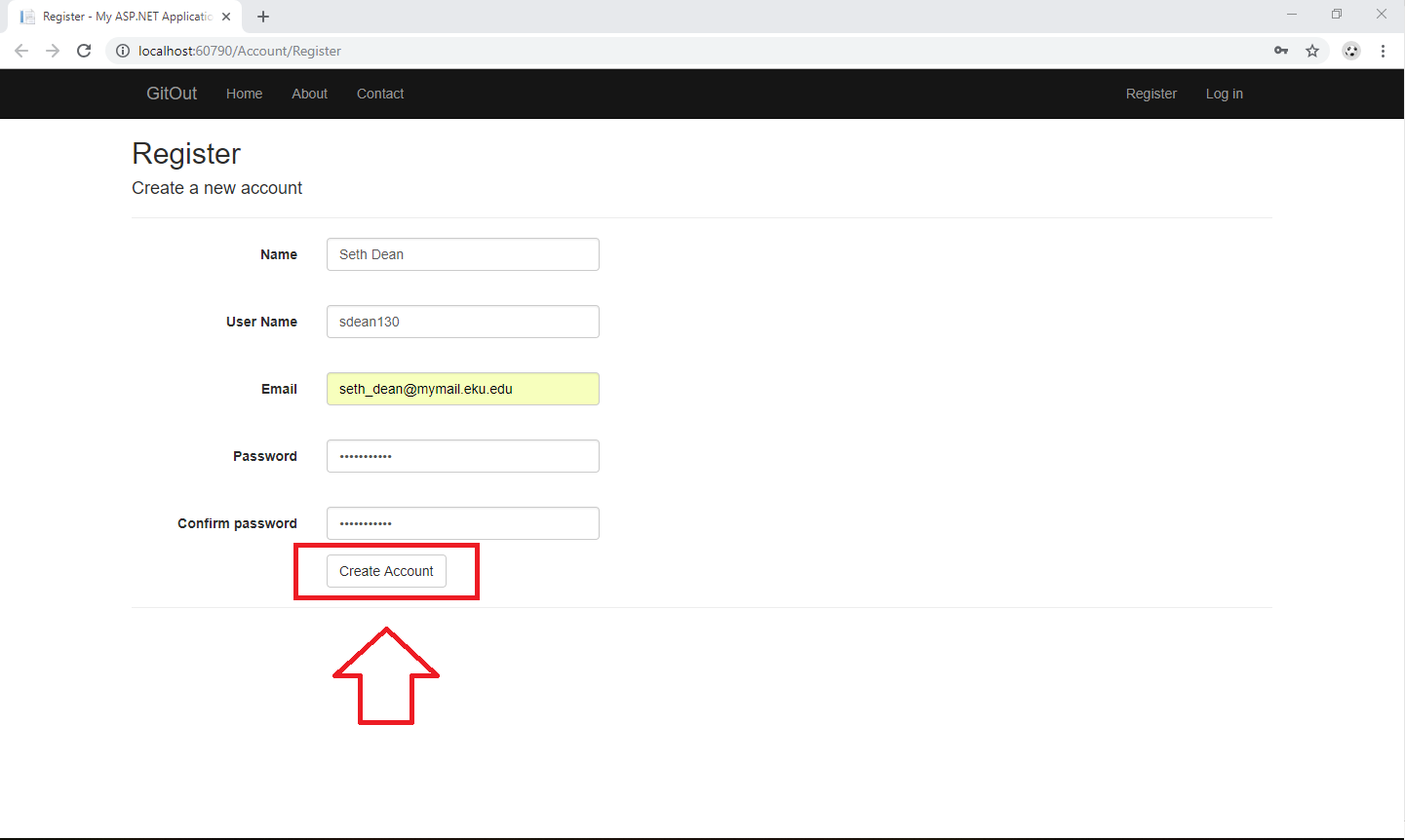
R1.4. The user shall enter an email address



R1.5. The user shall enter a password

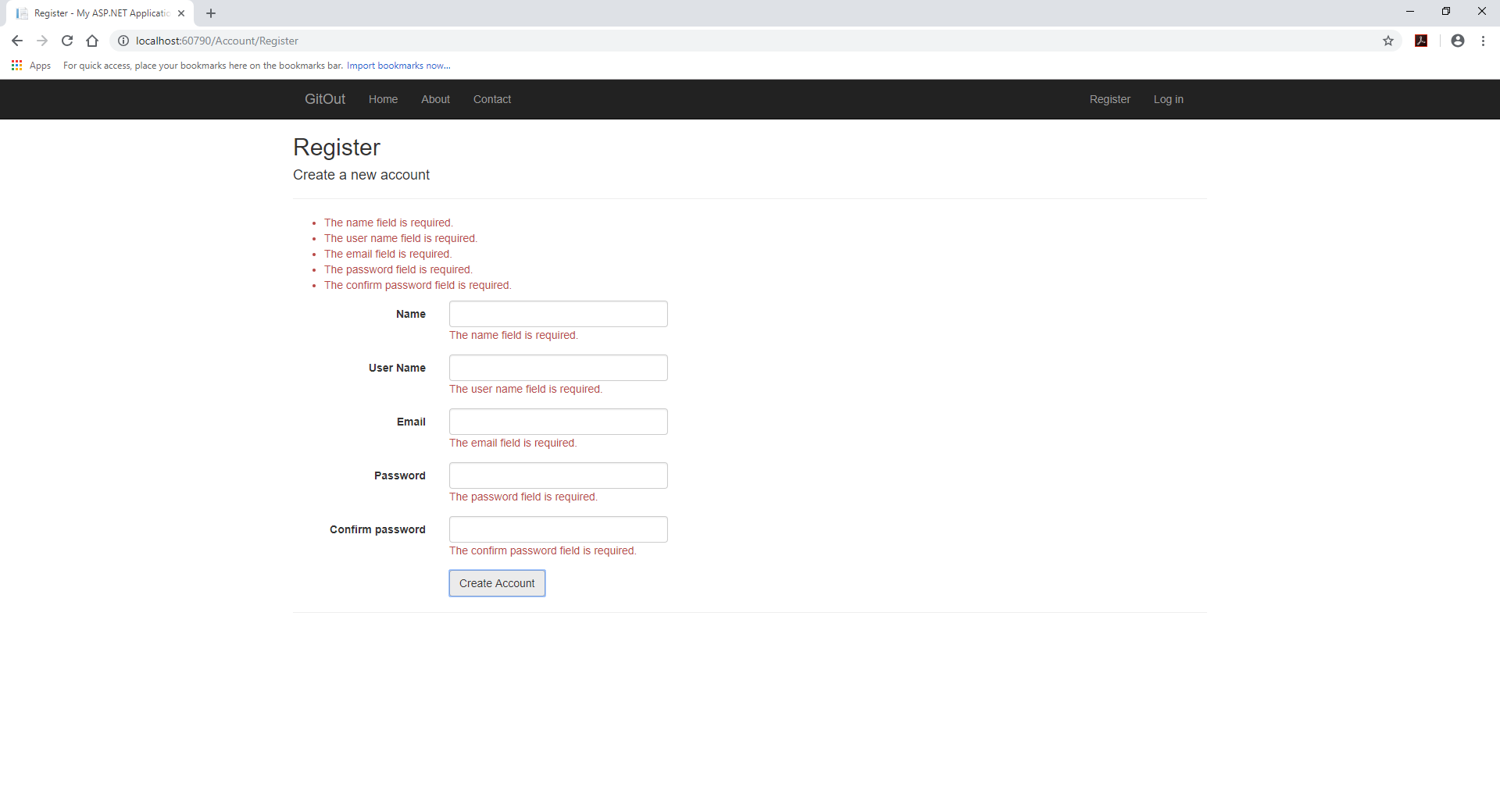


R1.6. The user shall click the create account button



R1.6.1. The system checks username availability

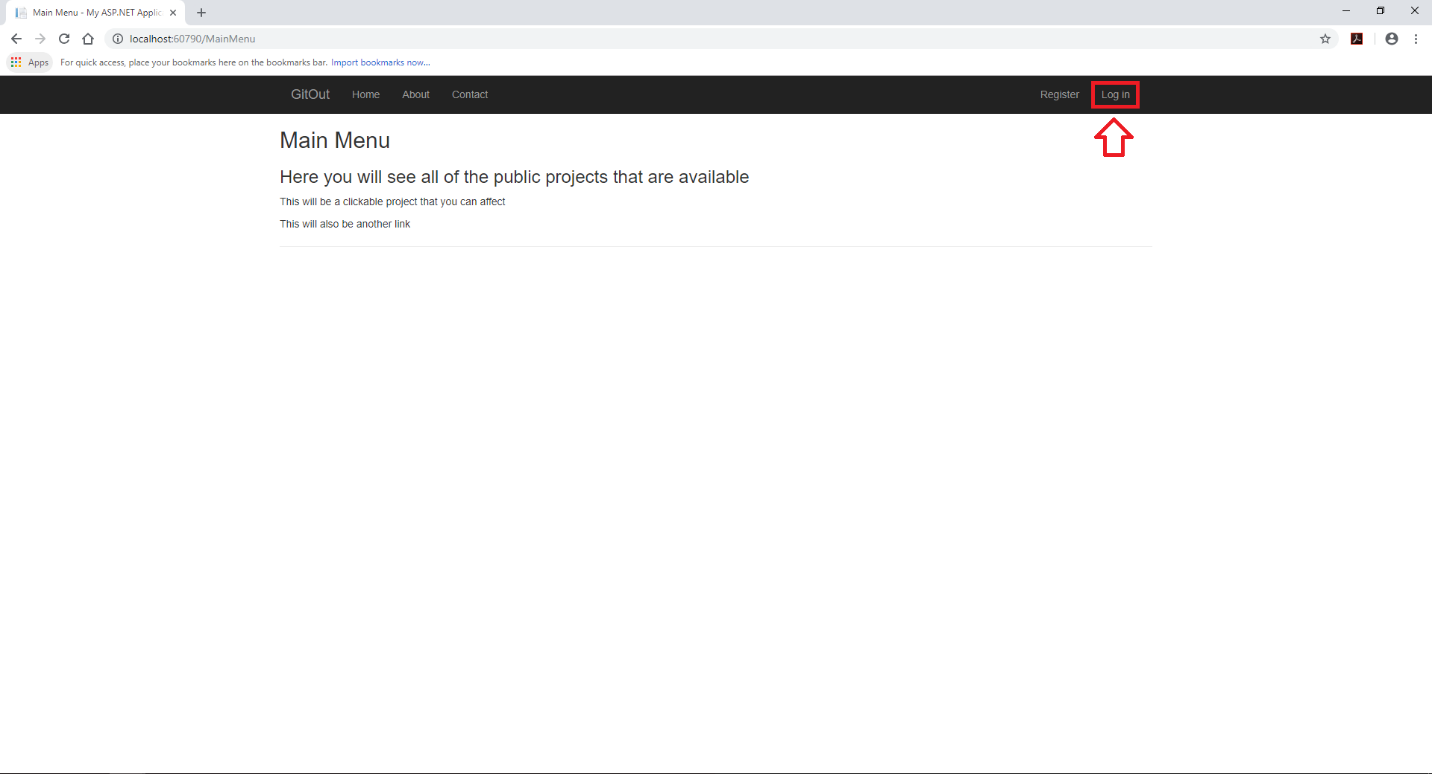
R1.6.2. The system shall return the registration screen with an error if there are incomplete fields



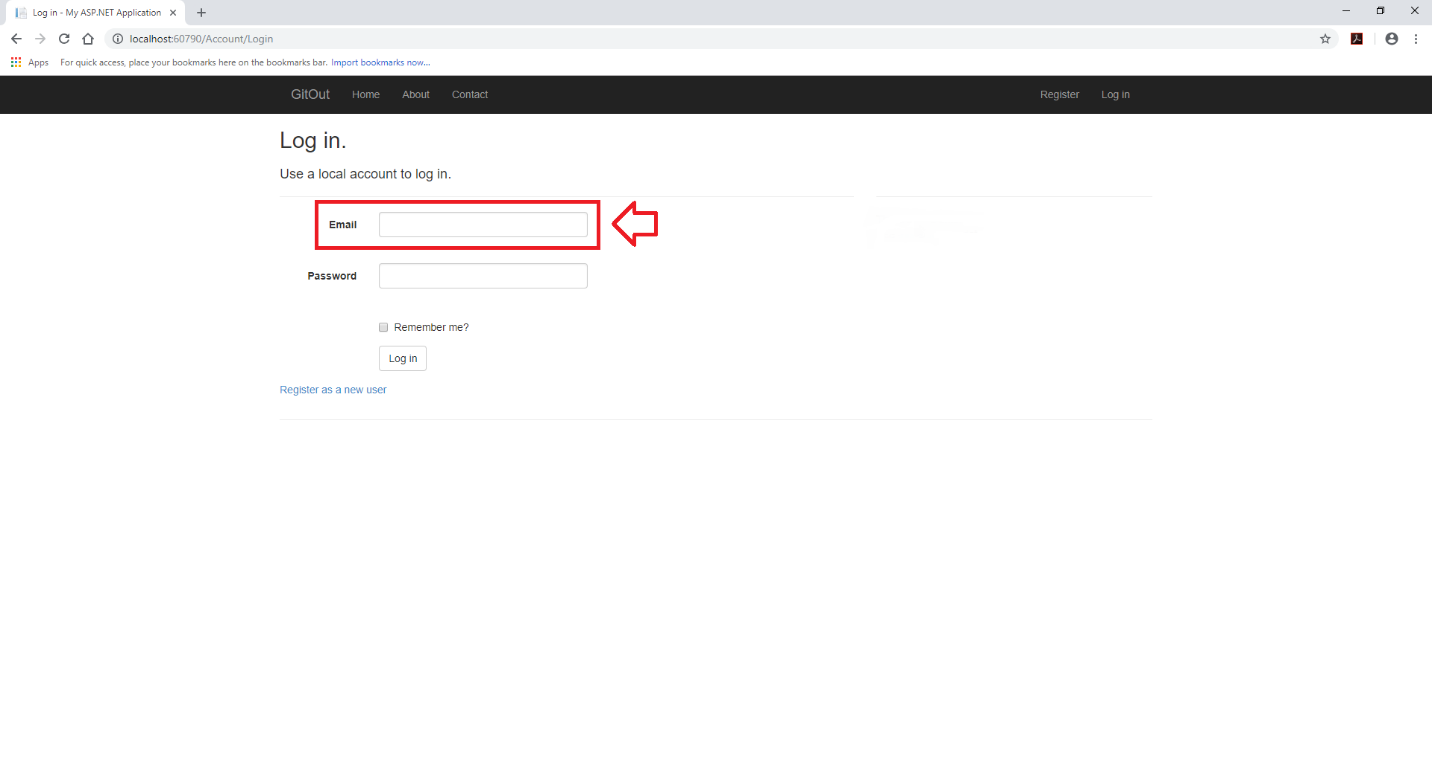
R1.6.3. The system shall return an error if the username has been taken and will send the user back to the same screen (R1.2)

R2. Log in to newly created account

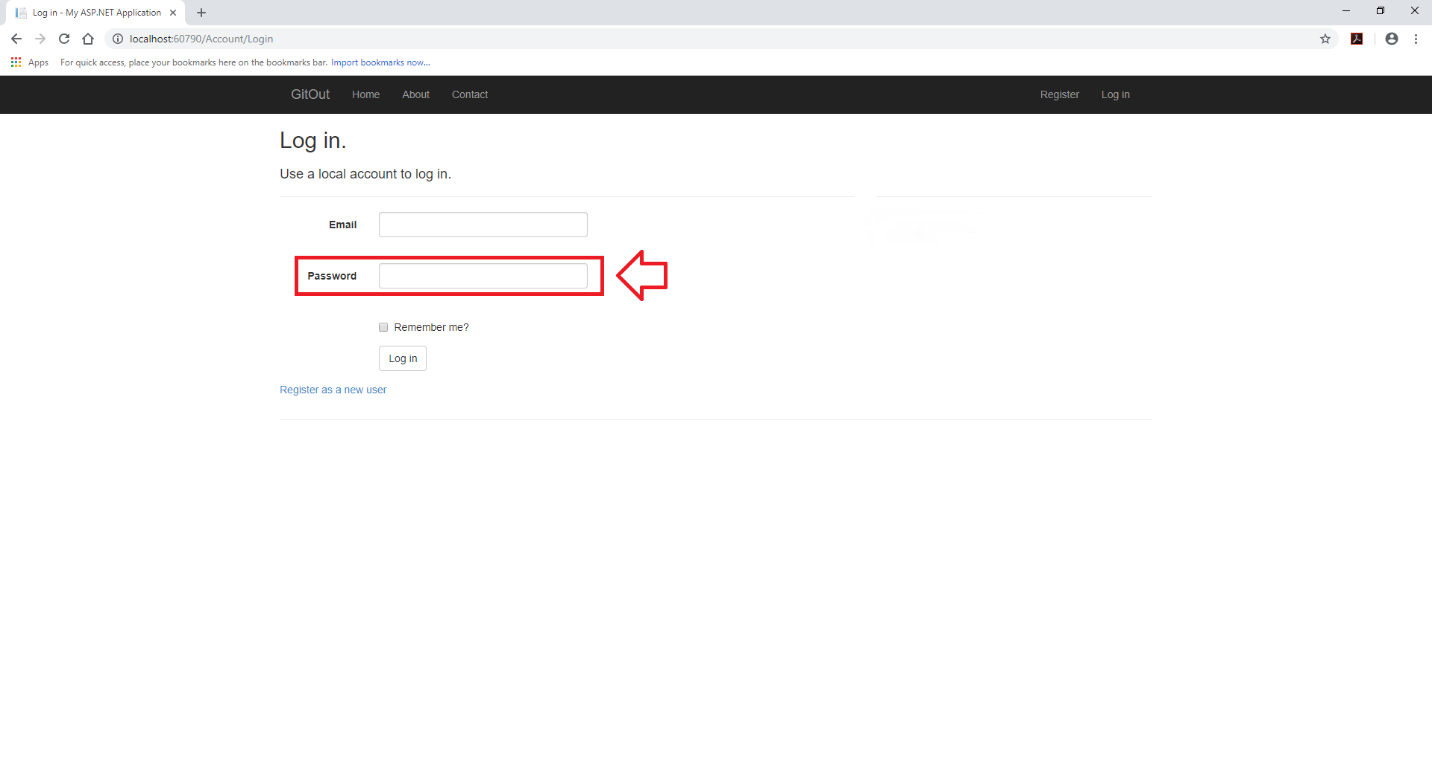
R2.1. The user shall click the login button



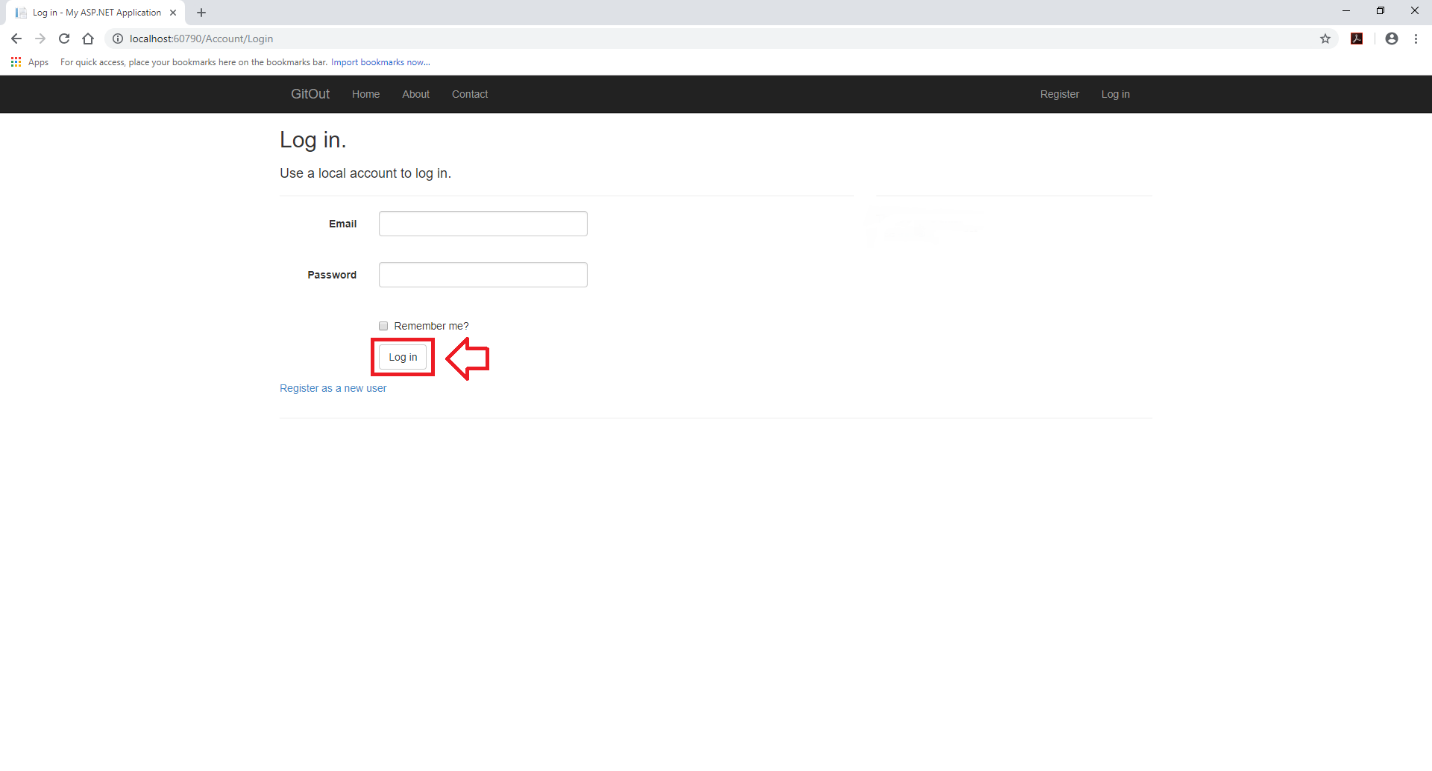
R2.2. The user shall enter either their email address



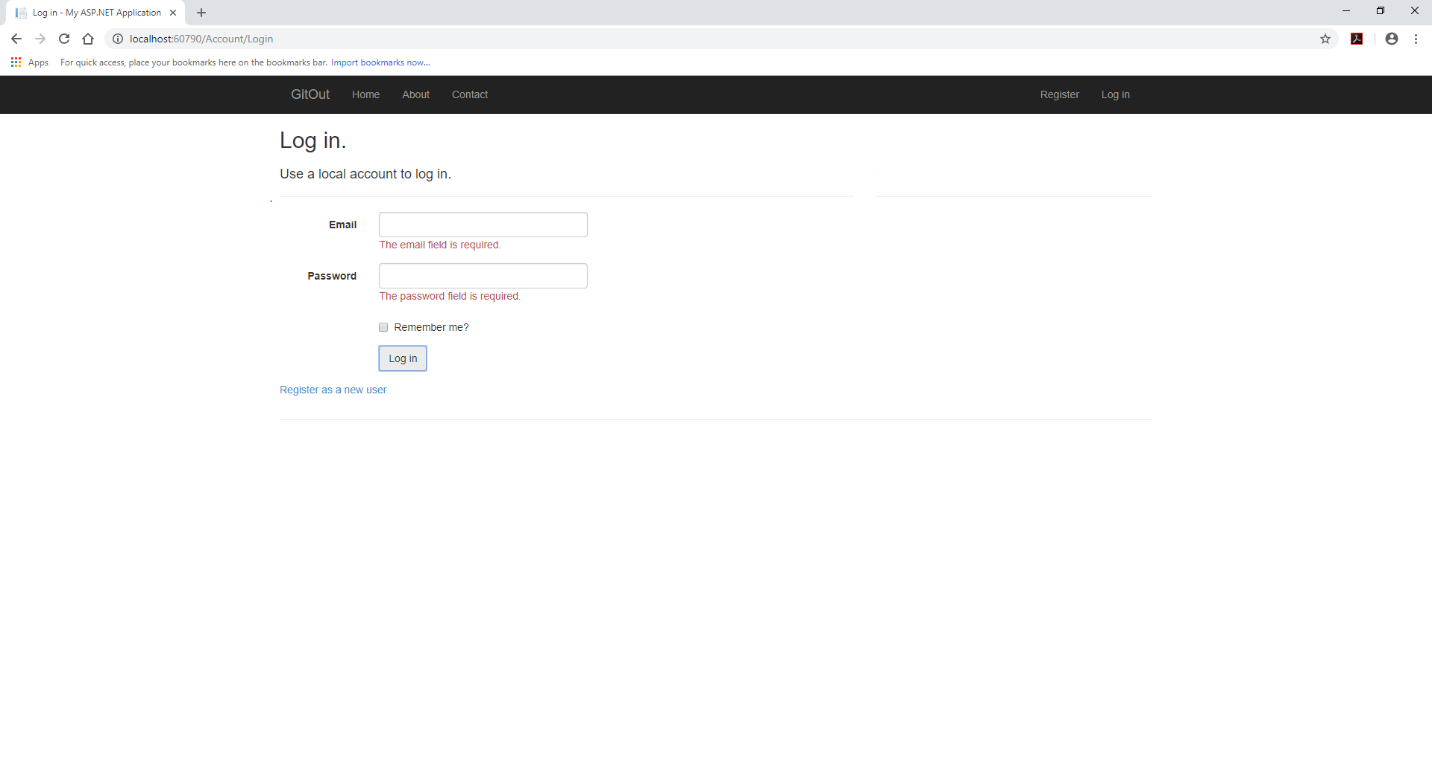
R2.3. The user shall enter their password



R2.4. The user shall click the login button



R2.4.1. The system will return an error if information is missing



R2.4.2. The system should check if this is account exists

R2.4.3. The system shall log the user in if it does

R2.4.4. The system shall return the user back to R2 with an error message if it

does not

R3. Creating a new project

R3.1. User shall click the new project button

R3.2. User shall name the project

R3.3. The user will choose the level of security that they want

R3.3.1. The system will give them the choice between public and private projects

R3.4. The user will then click the create project button

R4. Uploading files to Git

R4.1. User clicks on the project that they wish to change/update

R4.2. User shall upload their changes that they made to the project

R4.2.1. The system shall let the user know if they can access the project

R4.2.2. The system will allow the change to the project space to go through if they do have access

R4.2.3. The system will return an error if they don’t have access to the project

R4.3. The user shall be given the option to upload more changes they have to project

R4.4. The user shall be able to exit and work on other projects

R5. Downloading projects

R5.1. The user shall have the option to look at both past projects that they have made as well as others

R5.2. The user shall click on the download button if they wish to download the current project to their computer (R4.2.1 - R4.2.3 for permission)

R5.3. The user shall be able to work on the project

R6. Logging out of account

R6.1. The user shall click on their account name

R6.1.1. The system will display a drop-down list of options for the user

R6.2. The user shall navigate to the logout button

R6.3. The user shall click the logout button to log out of their account

## Non-Functional Requirements

NR1. At least 1 capital, 1 lowercase, 1 number, and at least 8 characters long when signing in

NR2. Comments on uploads

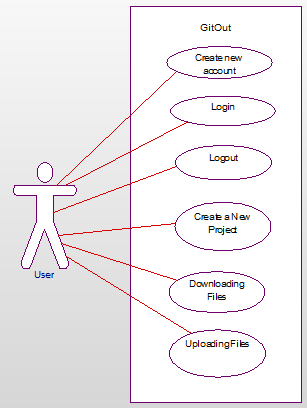
NR3. Security Questions when logging in

NR4. Time-stamp for upload

NR5. Search function

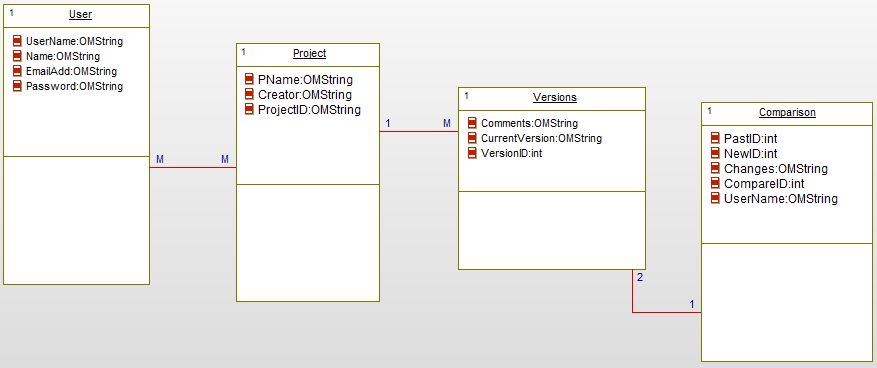
NR6. Listing project

# Use Case Diagram:



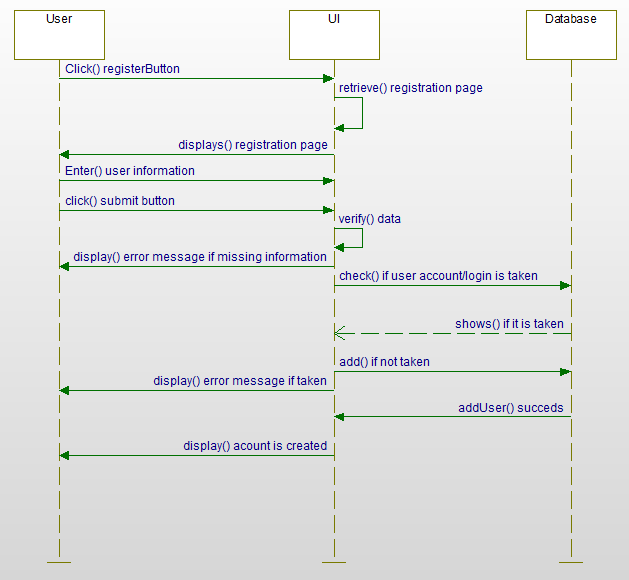
These six use cases are the most important actions that the user can interact with. The first three options that the user has when accessing GitOut are; Create a new account, login and logout. The other three options that they are giving are what make the website perform; Create a New project, upload files and download files. The registration will ask them for their information (Name, email address, telephone number, etc.) and then will allow the user to log into the website. Once the use has logged into the system, they are greeted with the main menu which will have other people’s projects. From here they can either create a new project, upload to an existing project or downloading a project. Creating a project will give them the option to name said project, add any files that they are currently storing in the system and create the project. They also have the option to upload to an existing project by clicking on the project they wish to upload to, click the upload button, choose the files and upload to the latest version. Download does the opposite and downloads to the user’s computer. The last option that they have is to log off which can be done through the button or leaving the page.

# Domain Diagram:



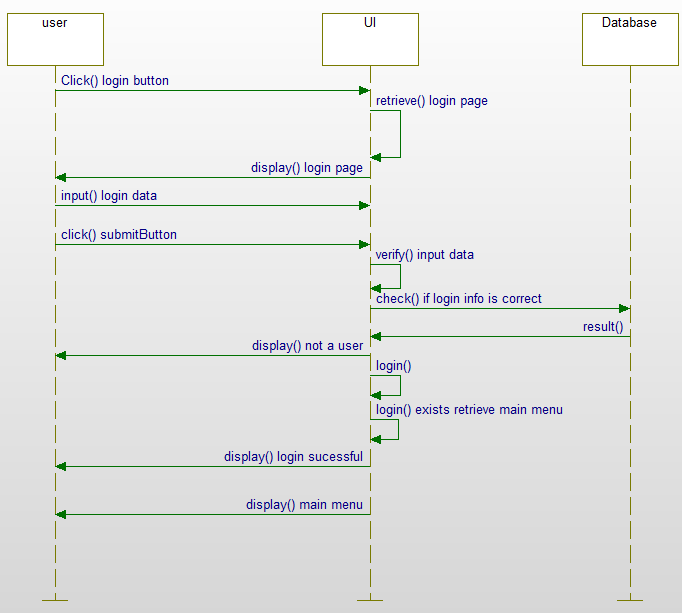
# Sequence Diagram

## Register



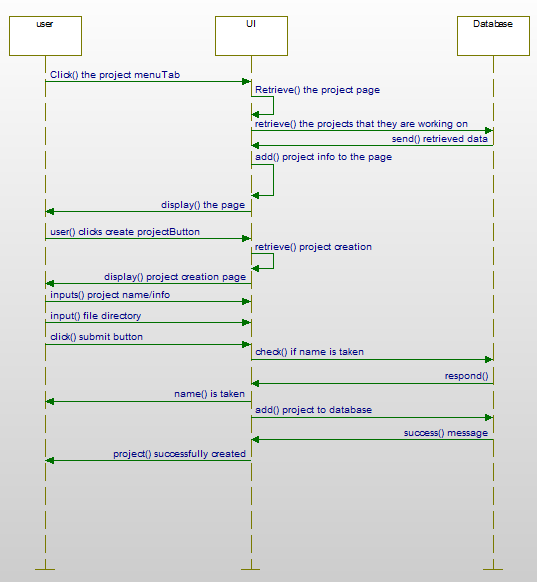
The user clicks the registration button, which will then display a page in which the user can input information. They will input their own information, it will then check if the user has an account already, which prompt an error message if any information is missing. The system will then check if the account has been taken, in which it will display an error message if there already is one. If this is new information, the system will add it to the database and prompt the user that their account has been created.

## Login



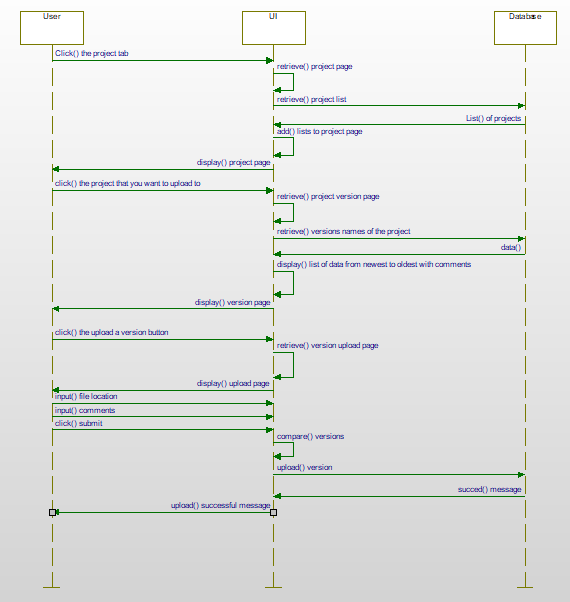
The user will be able to click the login button on the main menu, which will then grab the page and display it to the user. User will input their account information (Username/email address, password) and click the submit button. The system will then verify that this person is a user and will return whether they are a user. It will log them in if the are and display the main menu with their user name.

## Create A Project



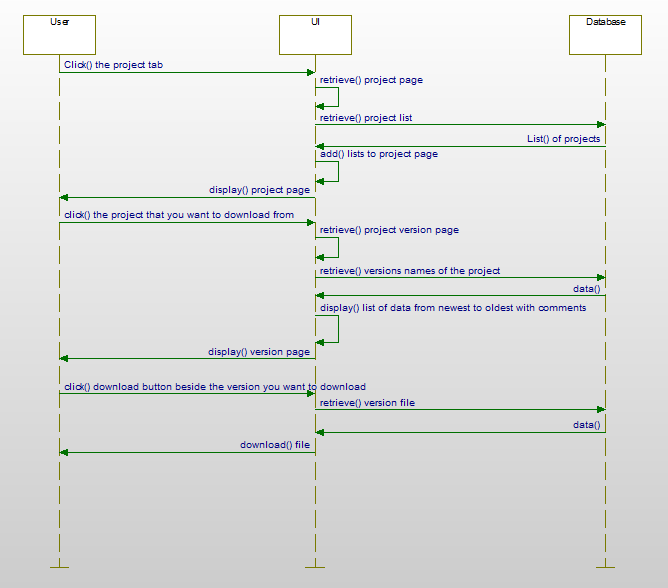
Here the user can click on the projects menu tab when they have successfully logged into the system. It will display all their projects that they are working on or have created. They then can click the create a project button, which will display a new page where they can input the project name and the information for it as well as where they want to save it. The user will then click the submit button and check if this project name already exists. It will return to the input information page to let the user know that this project has been taken. If not, it will save it to the database under their name and let the user know that it was created successfully.

## Upload



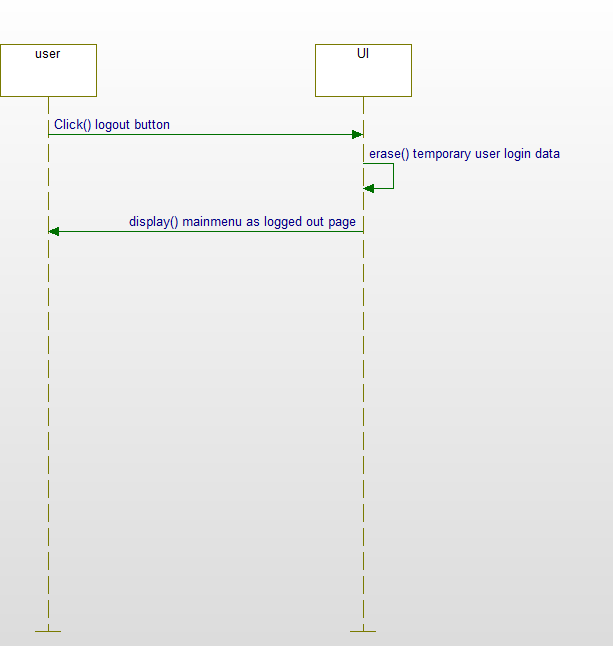
The user clicks the project tab which retrieves their projects and displays it to them. This will allow the user to click on the upload a version so that it will add to the current project that they are looking at. They can then add comments to the upload in which it will compare versions to make sure that information has been added or not.

## Download



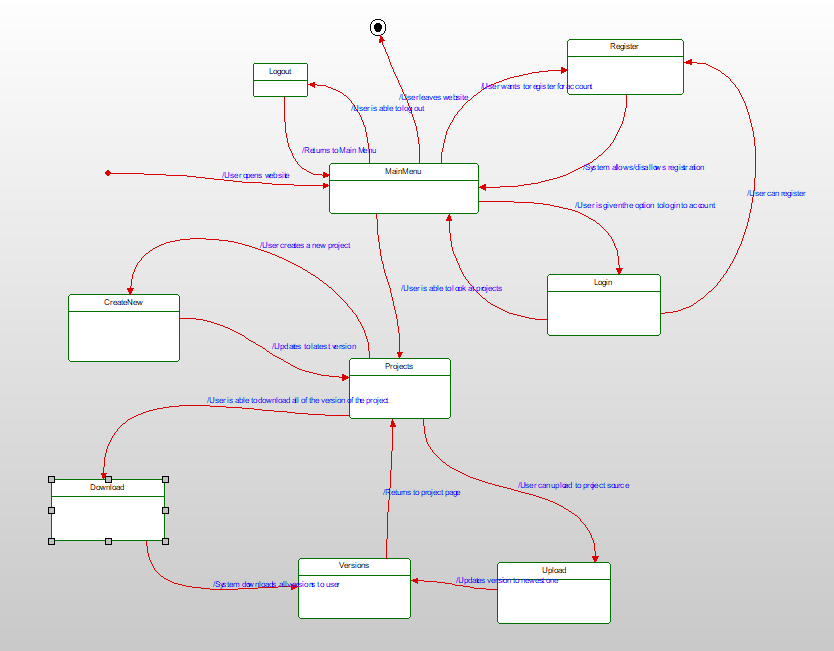
This portion will display the projects that they can download including public and private versions. It will retrieve the data from the project that they are currently on and copy it to the user’s machine.

## Logout



This portion shows the user log out process with the user being able to leave the website for it to logout or through them clicking the logout button.

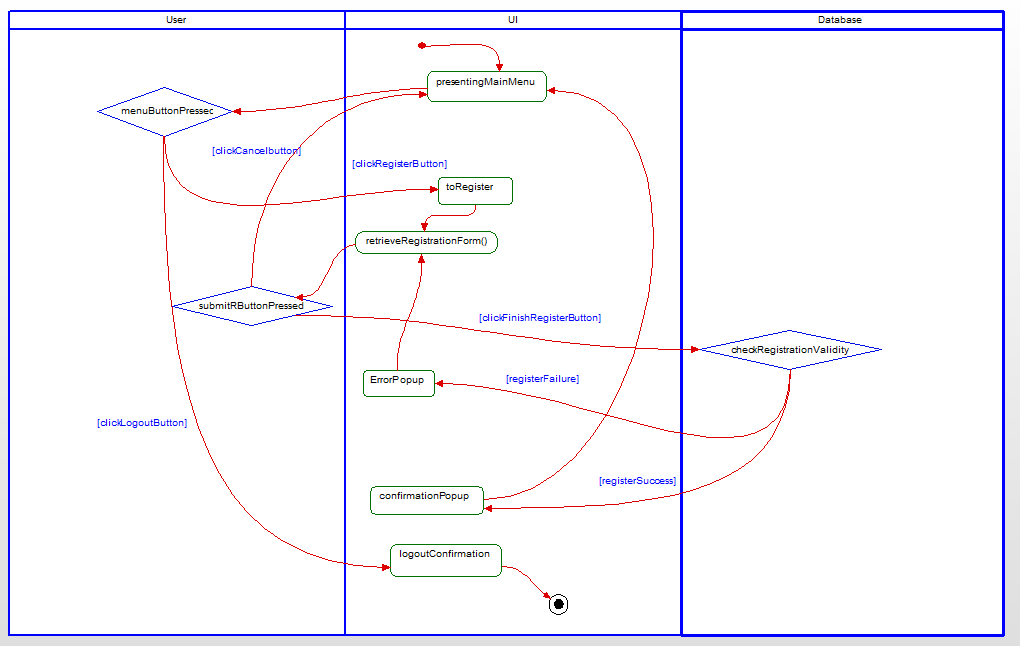
# State Diagram



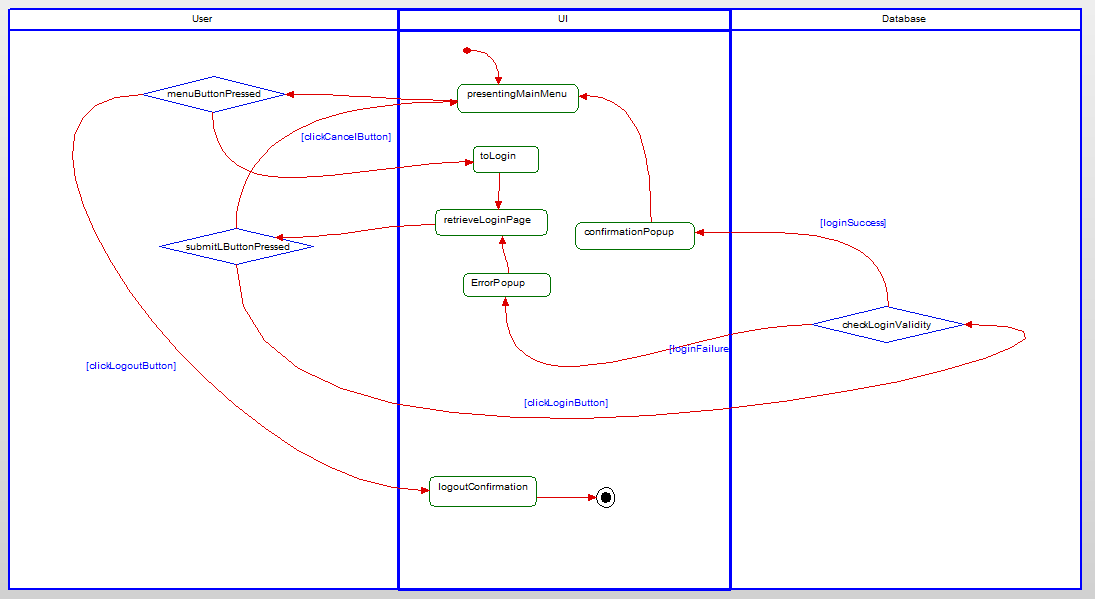
In this state diagram, we can that when the website is opened, it loads to the main menu page. From this point the user is allowed register an account with the system, log into the system or log out of the system. All these states will return the user back to the main menu with their action being completed. The user can also access the projects that they have created, or one that other users have created. From this state the user can create, upload or download projects. Creating a new one will return the user back to the project page. Both downloading and uploading the project will access the versions state of the system and either upload to the system or download from the system.

# Activity Diagram

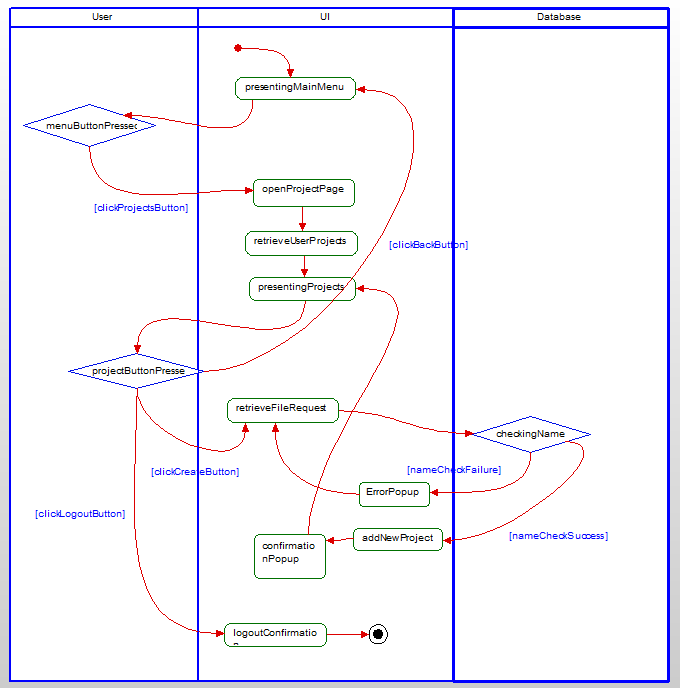
## Register



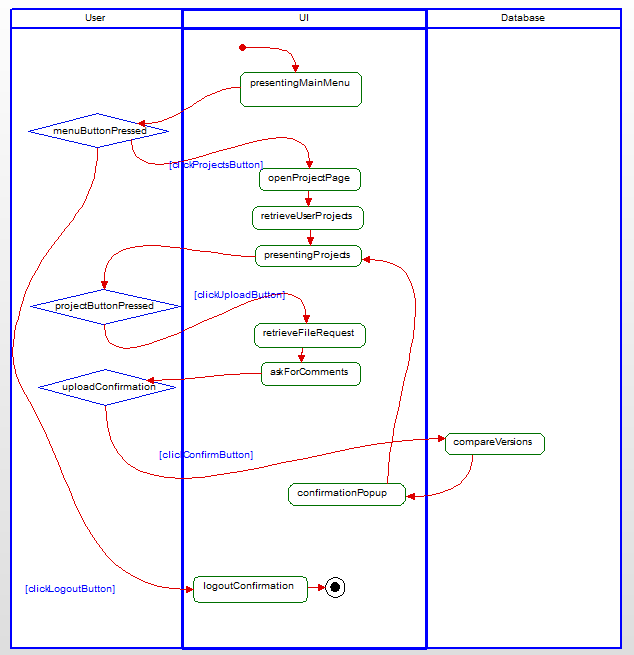
## Login



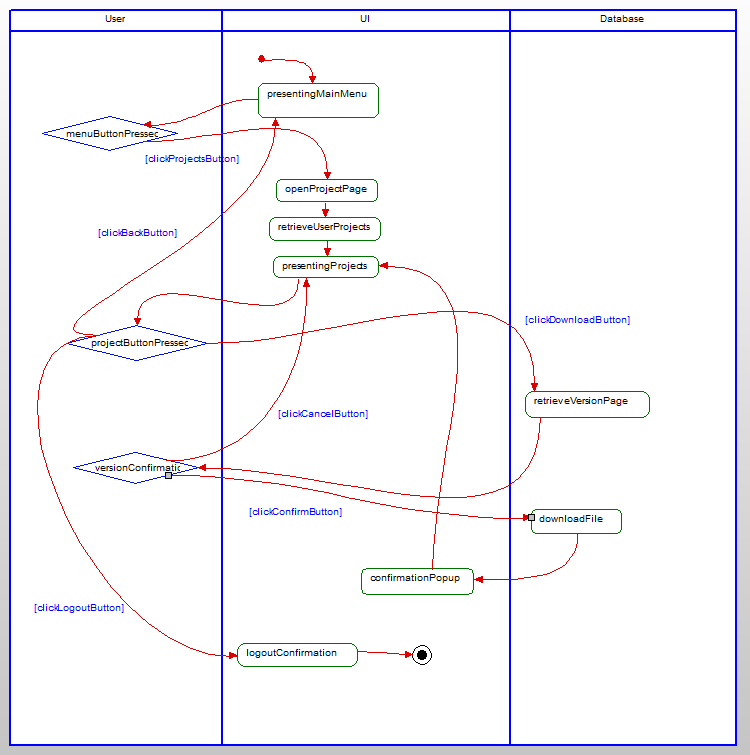
## Create A Project



## Upload



## Download



# Database Design

## ER Schema



## Table Schema



# Conclusion

The goal of this project was to create a system that improves upon the GitLab website and allows the users to easily access projects. This would allow more users to store their source code and other projects on this site and basically call for an ease of access all around

# Data Dictionary:

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Field Length | Description |
| UserName | String | 20 | Name displayed for user |
| Name | String | 20 | Name of user |
| EmailAdd | String | 50 | Email address of user |
| Password | String | 15 | Password for user |
| PName | String | 20 | Name of created project |
| Creator | String | 20 | User that created the project |
| ProjectID | Int | 50 | ID given to the project |
| Comments | String | 100 | Descriptions of user interactions |
| CurrentVersion | String | 20 | Latest version of the project |
| VersionID | Int | 20 | ID of the version |
| PastID | Int | 20 | ID before the latest version |
| CurrentID | Int | 20 | ID of the current project |
| Changes | String | 50 | Changes that have occured in the project |
| CompareID | Int | 20 | ID of the comparison between the PastID and currentID |