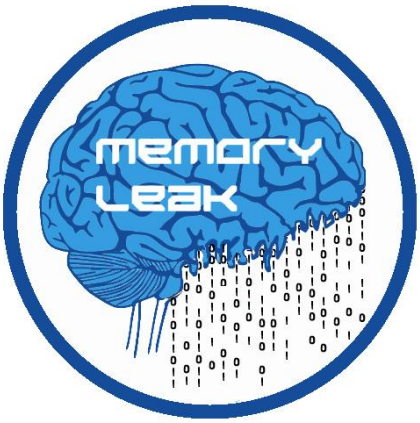
# Memory Leak
# Technical Documentation




Andrew Palmer, Christian Young, Kody Kimberl, and Michael Bauer

Capstone Project: Spring 2022

The University of West Florida

2/22/2022

CIS4595: Capstone Project

Dr. Bernd Owsnicki-Klewe

## Abstract

This work outlines the technical aspects that we will follow throughout the development of our project. These technical outlines include development and deployment aspects such as why and how we chose a particular tool or software. Additionally, we note numerous artifacts that define, test, and document the requirements for our project.

# Table of Contents


Abstract **2**

Table of Contents **3**

1 Team Process **4**

2 Requirements Elicitation Strategy **4**

3 Requirement Examples **5**

- 3.1 Trello Workflow 5
- 3.2 Use Case Example 6
- 3.3 Accessibility Testing 7
- 3.4 Completed Code 8

4 Development Platform **8**

5 Targeted Platforms **9**

6 Deployment **9**

# 1 Team Process

We will be utilizing the Agile development methodology, more specifically the SCRUM process. This model was chosen because each member of the team is familiar with it already from Software Engineering. It also works very well for a web development setting that contains many modular tasks. The overall characteristic of Agile that solidified our decision is that it prioritizes responding to changes and working software over extensive plans and documentation. While planning and documentation are important, we wanted to ensure that the ability for change and growth is present as we all continue to develop the idea together.

# 2 Requirements Elicitation Strategy

The primary users of our application include UWF students. Our team is composed of individuals that could be considered users of the platform. Everyone working on the project is a student, and two of the members are tutors. Thus, we have a great understanding of what we would like to see as users of the platform. Our team worked together to develop a detailed understanding of what would be required. Then, during the construction of some of our artifacts, we continued to expand upon the requirements and clarify them in more detail.

With respect to artifacts, we have numerous artifacts that will convey the planning, development, testing, and completion of the project. The first set of artifacts are our user stories and use cases, which convey many of the requirements for the core application. Additionally, we have set up these user stories in a Trello workflow that depicts the steps of our development process. We also are developing testing procedures for code, user acceptance, accessibility, and

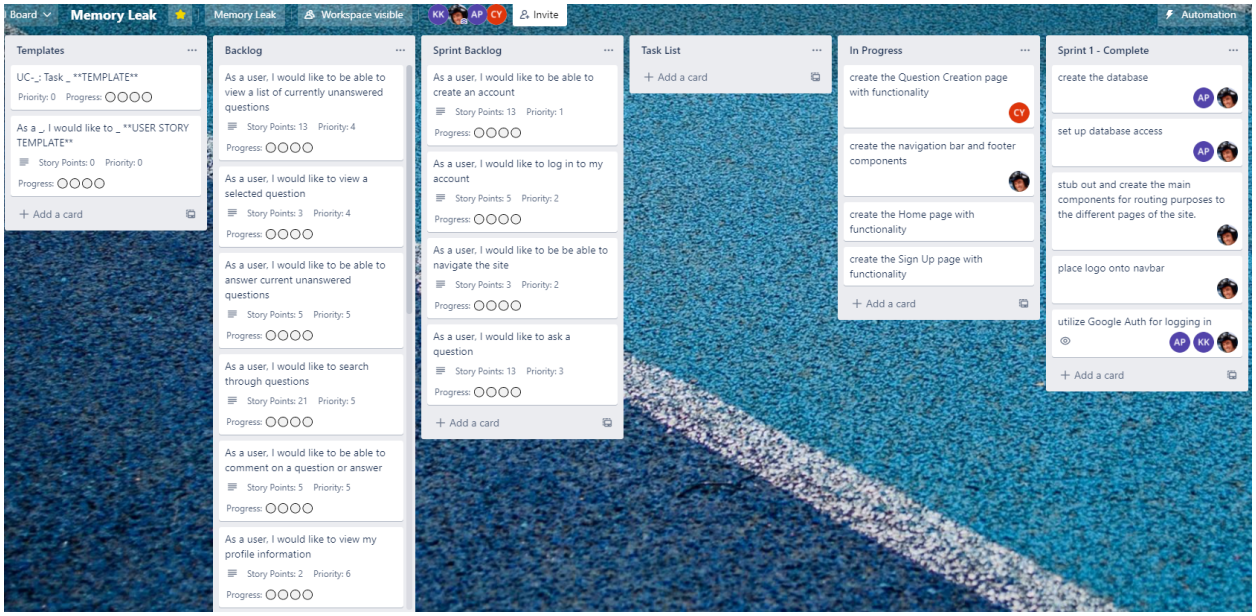
security in order to ensure that the requirements are being met. Lastly, we also plan to produce a completed web application that is deployed and accessible on the web.

Our requirements elicitation phase was completed prior to the start of Sprint 1. This deadline allowed us to start development with a much deeper understanding of what would be required.

# 3 Requirement Examples

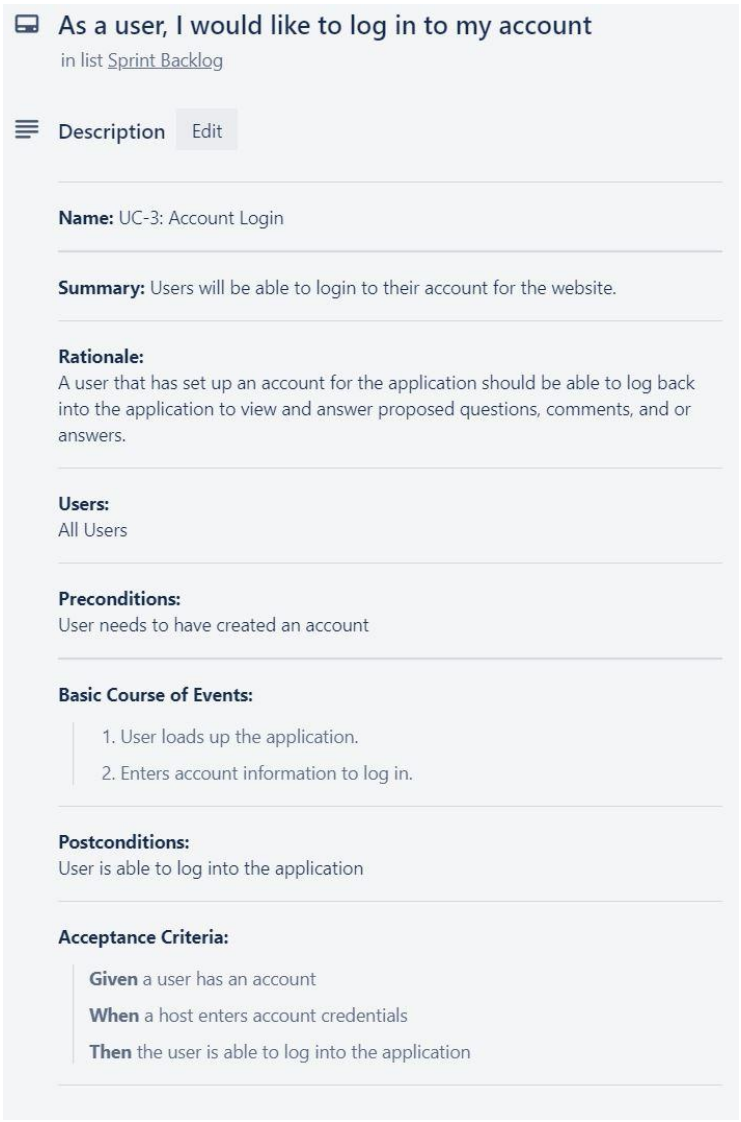
## 3.1 Trello Workflow

To ensure that the most pressing requirements are completed first, we have developed a workflow for the development process. The Trello workflow details the priorities and estimated effort for each user story along with the anticipated stories and tasks to be accomplished in each sprint.



## 3.2 Use Case Example

For many of the core requirements, we developed use cases to outline them in more detail. The following is what one of our completed use cases looks like. The entire set of user stories and use cases is comprised of 17 in total.



As a user, I would like to log in to my account

in list Sprint Backlog

Description Edit

**Name:** UC-3: Account Login

**Summary:** Users will be able to login to their account for the website.

**Rationale:**
A user that has set up an account for the application should be able to log back into the application to view and answer proposed questions, comments, and or answers.

**Users:**
All Users

**Preconditions:**
User needs to have created an account

**Basic Course of Events:**

1. User loads up the application.
2. Enters account information to log in.

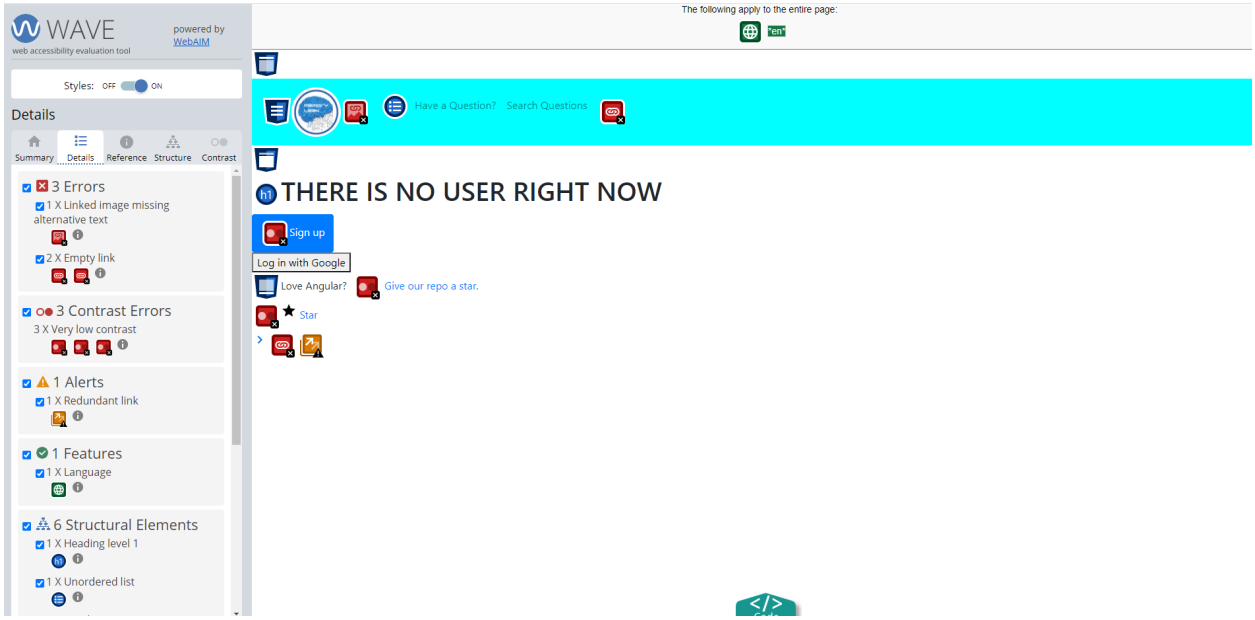
**Postconditions:**
User is able to log into the application

**Acceptance Criteria:**

**Given** a user has an account
**When** a host enters account credentials
**Then** the user is able to log into the application

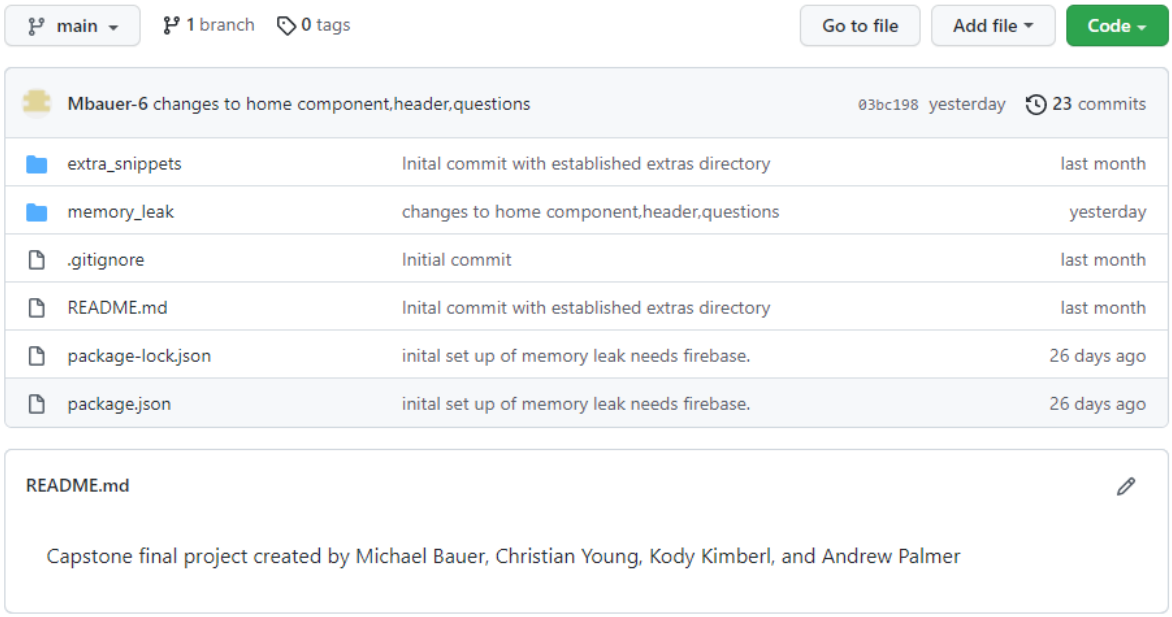
## 3.3 Accessibility Testing

One of our requirements for the project is that the project is accessible to as many users as possible. In addition to developing the application with ARIA in mind, we will be testing the site's accessibility using an extension called WAVE. This is the current accessibility test after Sprint 1, showing many things that need to be adjusted.



## 3.4 Completed Code

Another requirement of our project is a working final product. Each sprint, we will deploy a prototype that will reflect the new additions to the codebase. We will also have the completed set of code in github.




# 4 Development Platform

We are developing using the Angular framework in conjunction with node.js. Angular utilizes TypeScript, JavaScript, HTML, and CSS. Our team uses Visual Studio Code as our main IDE. Our application will rely on Firebase for hosting and Firestore for database persistence and access. For testing purposes we will be implementing the Jest testing framework and the WAVE tool for accessibility testing.

# 5 Targeted Platforms

The platform we have chosen to target is a web browser application. We chose this platform due the familiarity that the users have with web browsers. Using this platform, we are also able to develop mobile friendly versions by utilizing certain criteria in the code to adapt to screen sizes and what may or not be visible or accessed on those screen sizes. Since the development is utilizing a PaaS and a DBaaS, such as Google's Firestore and Firebase, the external components are being secured by those service providers.

# 6 Deployment

For our deployment process, we will deploy our application to Firebase for hosting. For version control and synchronization we are using a central GitHub repository. To access this repository we will use the GitHub command line interface.