

# Software Design Document

---

## Section 1 - Project Description

### 1.1 Project

Source Control

Demonstration

Project

### 1.2 Description

This project was assigned to me to meet my honors requirement for the undergraduate honors research program and aims to teach me about source control and continuous integration which are key to modern DevOps. This project will also incorporate the various topics we cover in the Software Engineering I Course. The following is the instructions for the project:

“Source Control is a critical aspect to team-based software development. In SE I, students are introduced to the basic concepts of committing code and managing branches. Your independent project will be to take it to a higher level by leveraging GitHub actions, Discord, and Selenium. You will create a repository in GitHub to contain a simple website with one or more buttons to cause a simple state change. Once that is working, you will automate the inclusion of two code reviews (by team members other than the author of the code change) done sequentially or concurrently such that if a code review is not completed or was rejected, the code change is not carried forward in GitHub. After that, you will integrate Discord into the process so that a message goes out to a discord channel when a code review is being requested and the pull request it corresponds to. After that, you can create a Selenium automated test against the website. Once that works, you can integrate that test into the code change process within GitHub (to occur after the two code reviews) so that the test has to pass before the change can move forward.

All of the above is part of Continuous Integration which is key to modern DevOps.”

“Do the code review part first, then the Selenium part.

For the code review, you will set up a workflow within GitHub to have two code reviews on each commit into the repo. The "automation" being referred to here is the workflow itself which will require two different users (both different from the author of the commit) to perform an action in GitHub indicating a code review was performed. This is an action that GitHub should have some examples of. You are not automating the review itself, just the action as part of the commit workflow.

Similarly, for the Selenium test, you are inserting the execution of the Selenium test in the commit workflow. To do this, you will have to have a working webpage and then compose a Selenium test against it. An example might be that the default page is loaded by Selenium and then it initiates a menu or button selection that should take the user to a different page, and then it verifies that that new page is displayed. You can be more elaborate than that if you wish but that type of action is all that is necessary. Once you have this test working, integrate it into the GitHub workflow. This simulates regression testing. It will verify that every commit made will not break the page change. If it does, the commit will fail. The commit doesn't have to have anything to do with either the default page or the page the test navigates too. The test just confirms the page change still happens if the commit were to be finalized.”

# Software Design Document

---

## 1.3 Team Members

Nicholas Crump

## 1.3 Revision History

Date	Comment	Author
2/13/2025	Created the document, added the requirements, and the Architectural Design	Nicholas Crump
2/19/2025	Finished the Designs and Traceability Matrix	Nicholas Crump
2/26/2025	Added an additional Design & updated Matrix & Requirements	Nicholas Crump

# Software Design Document

---

## Contents

### [Section 1 - Project Description](#)

#### [1.1 Project](#)

#### [1.2 Description](#)

#### [1.3 Revision History](#)

### [Section 2 - Overview](#)

#### [2.1 Requirements](#)

##### [2.1.1 Traceability Matrix](#)

## Section 2 - Overview

### 2.1 Requirements

1. The solution must include a simple website
  - a. The solution must have a button that changes the state of the website
2. The solution must include GitHub
  - a. The solution must include two code reviews in the process
    - i. The solution must have two different actors that are not the author of the pull request to perform the code reviews
    - ii. The solution must have the code reviews be done sequentially or concurrently
    - iii. The solution must reject the code review so that the code is not carried to GitHub
3. The solution must incorporate GitHub Actions
  - a. The solution must incorporate Discord
    - i. The solution must send out a message to a discord channel once a code review is requested
    - ii. The solution must contain the pull request it corresponds to
4. The solution must incorporate Selenium
  - a. The solution must integrate a test that checks if the website has a state change if the two code reviews are successful
    - i. The solution must allow for the code to be carried to GitHub if the test succeeds

## 2.3.2 Traceability Matrix

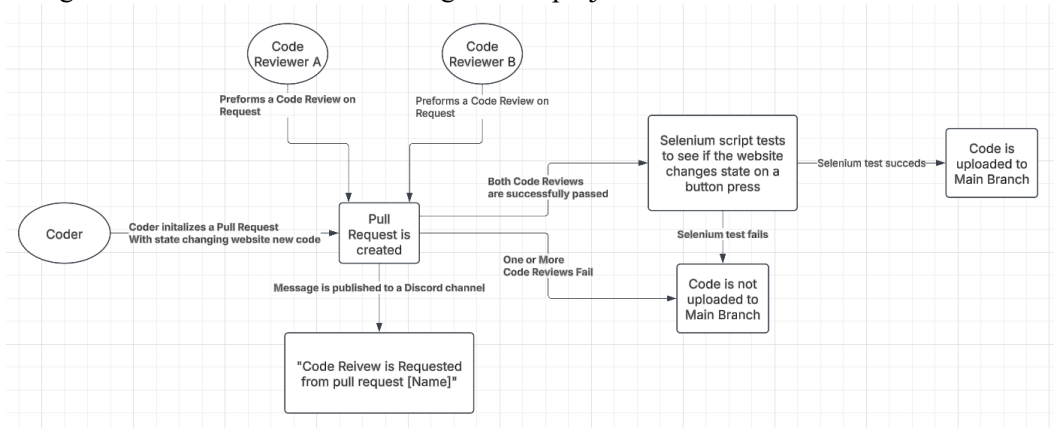
### SDD Module

SRS Requirements	Architectural Design	Website Design	Two Code Reviews
1	P	F	
1.a	P	F	
2	P		F
2.a	P		F
2.a.i	P		F
2.a.ii	P		F
2.a.iii	P		F
3	F		
3.a	F		
3.a.i	F		
3.a.ii	F		
4	F		
4.a	F		
4.a.i	F		

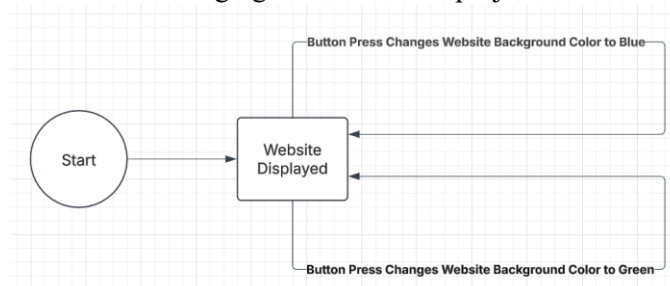
# Software Design Document

## Section 3 - Design

The following is the overall architectural design of the project and the flow:



The following is a design of the state changing website for the project:



The following is the design of the inclusion of two code review for the project:

