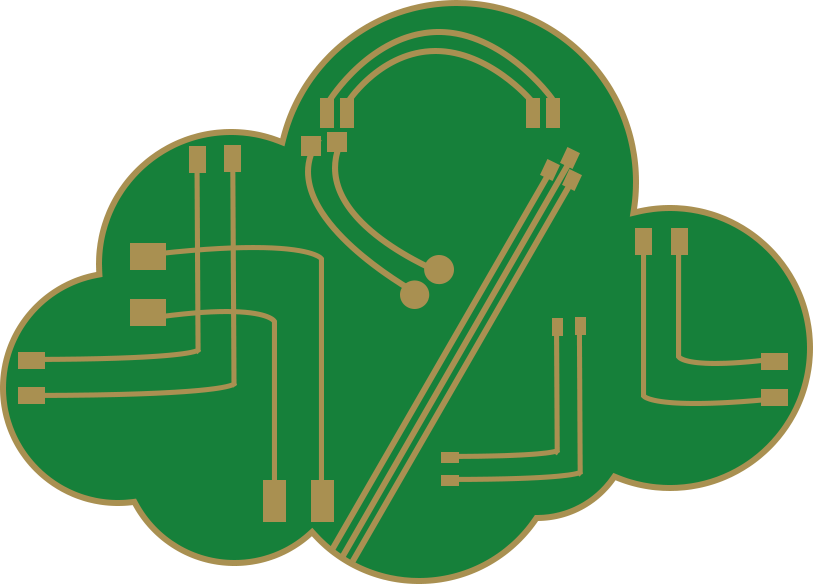
Cloud 9

CS 499 - Spring 2020

2/18/2020

## Team Members:

-Benjamin Ellis-

-Alex Reel-

-Camron Savage-

-Alex Tanner-

-Patrick Yoder-

-Yankai Zhao-

# Test Automation

## Customer:

Ben Fox - Trissential

Edited Using Google Docs

## Table of Contents

[**Introduction**](#_b126qr74nuoc) **2**

[**Project Overview**](#_s7fzmq1f6t79) **3**

[**Development and Target Environments**](#_yc52az210aa1) **4**

[**System Model**](#_l6jfn0vq7kfj) **4**

[**User Interaction**](#_ewxi7avqapwk) **5**

[**Functional Requirements**](#_dqn9dqxaz3hn) **6**

[**Nonfunctional Requirements**](#_3x4b6qifgnmf) **8**

[**Feasibility**](#_p94o83yintq7) **8**

[**Conclusions**](#_hff2wx7ooq6p) **8**

[**Appendices**](#_8m3fsdce48jh) **9**

# Introduction

In the Fall of 2019, Trissential tasked a group of students to create a web application that would help train and/or hire quality assurance personnel. The result was an e-commerce website that allowed for the assigning of “bugs” to different users. The associated user could then log in and run through their own test cases with the intention of finding the non-standard behavior assigned to them. Our job this semester is to improve and expand upon this idea. The most important goal is to improve the styling across the site while also adding other standard e-commerce features and assignable bugs to go along with them.

The purpose of this document is to outline our plans for the final project. This document is for use by the instructor and customer to track our initial plans and estimations for the coming months. Content of this document includes the following:

* An overview for background information and factors that influenced the project’s direction.
* A system model to outline the components of the website and how they relate to each other.
* A user interaction section to display the ways in which users can interface with the website.
* Functional and nonfunctional requirements that display the changes and updates we intend to make to the application.
* A feasibility section to display the minimum requested version of the project and the ideal version of the project.

# Project Overview

This application will be an extension of Trissential’s existing web application. It will provide Trissential with a more comprehensive means to train quality assurance personnel. The current system has the foundation of our planned application, but lacks proper visual design and an extensive suite of bugs which can be enabled for training purposes. Our implementation will improve upon these issues in order to provide a more complete application.

In building off the existing web application, we will be able to implement a fleet of features that the original development team did not. Among these features are many optional bugs which can be enabled for specific users to train Trissential QA personnel in testing and testing protocol. By creating an environment where bugs can be added to user accounts, QA personnel will be able to train their testing abilities to find the applied bugs. Additionally, a visual overhaul is necessary to improve the design of the web application by making it more professional and consistent with Trissential’s parent corporation. Beyond these features, we will implement wish lists and other crucial e-commerce features. Our expansion of Trissential’s web application will focus adding these features in a manner so that they can be used reliably, and consistently, to train users.

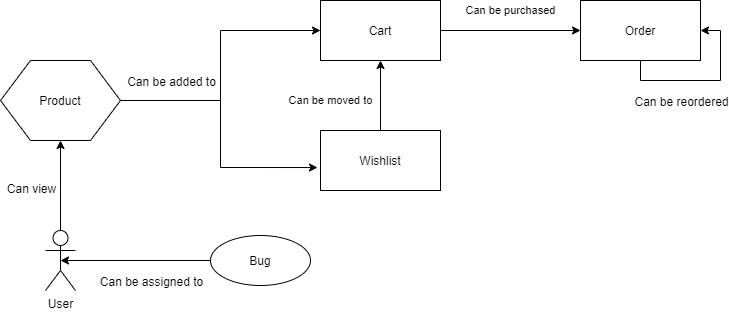
# Development and Target Environments

The existing project was designed for use on a LAMP stack (Linux, Apache, MySql, PHP). Considering the customer’s requirements of polish and feature additions, the LAMP stack design will remain. An additional requirement is that the PHP server the application runs on must include an email sending service.

To run the application, a Linux system is recommended, though most of the team is using Windows with PHP compatible ide’s for development. In either case, PHP and MySql must be installed. The application needs to be configured to connect to the MySql server but database creation and configuration is handled by the application upon first startup. The email relay service is necessary to validate new users. The exact service to use depends on the configuration of the server, so it must be decided on a case by case basis.

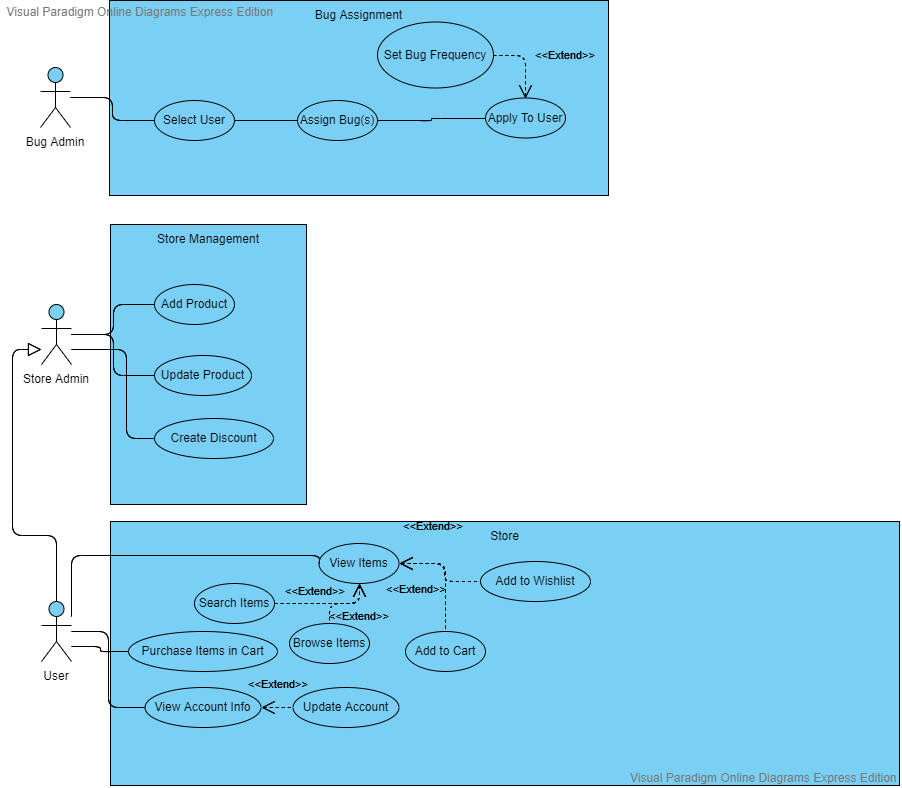
# System Model

Here we will display a high-level view of the major areas of our project. This model will include a combination of the existing project functionality and that which we plan to include in the coming months. A diagram of this system can be seen below:



This diagram displays how the primary areas of the application interact and how the website as a whole will flow. We see that the main innovation of this project, the bugs, can be assigned to a user. That user can view products, add them to their cart or wishlist (they are also able to move products from the wishlist to the cart), and place an order of the products in their cart. Previous orders can also be replicated and ordered again. Additionally, all of the listed functionality can potentially be influenced by any bugs assigned to the user.

# User Interaction



The diagram above is the actions that can be taken from the perspective of each user type. A bug admin can log in to the system, select a user, and assign bug(s) to that user. An optional feature in this process is the ability to set a frequency for different bugs, allowing for the simulation of intermittent issues. Store admins have all interactions that normal users do, along with the ability to create or update products or create discounts. Lastly, standard users are able to view their account information and update it. They can also look through the product catalog, add products to their cart or wishlist, and purchase items in their cart. The actions of users and store admins can also be influenced by any bugs assigned to them by bug admins.

# Functional Requirements

There are three primary areas of required development: visual styling improvements on the existing website, ecommerce feature additions, and the addition of new bugs to coincide with the newly added ecommerce features. No features were given any priority over others. However, the priority has been placed on ensuring any features we implement have assignable bugs to go with them, rather than adding more ecommerce features and no new bugs. The details of what features and bugs to add were left open-ended. However, we were given a number of ideas to help kickstart development.

Most requirements for the project are functional ones, as the base of the project has already been built by a preceding group. Almost all stories given to us by the customer were in expanding the existing functionality of the project

* Confirm the current condition of the “past orders” functionality.
  + Finish work on this area if it is not already completed.
  + Add a resubmit order button that copies the products and product quantities into the user’s cart.
* Add a page for user account information.
  + Allow users to set multiple addresses on this page.
* Enforce character limits on form fields where appropriate.
* Add email support to notify users of completed orders.
* Add wishlist functionality that users can save and use to add items to their cart.
  + Allow wishlists to be marked as public or private.
  + Add a page to search for public wishlists.
* Allow for subscription based/automatically recurring orders.
* Add promotion codes and store-wide discounts.
* Display a dynamic image on the store’s front page.
* Allow for adding to cart on the search page, rather than having to enter individual product pages.
* Allow changing between thumbnail and a compacted list view when searching for products.
* Allow for the setting of a free shipping price point.
* Allow assigned bugs to be set as intermittent, so an admin can set how often the bug can occur.
* Add multiple levels of users.
  + Bug Admin: Assigns bugs to users and can add new bugs.
  + Store Admin: Can add new products, update products, set discounts.
  + User: Can Order items and create wishlists.
* Add new bugs.
  + A bug that places incorrect items into the cart when resubmit order is pressed.
  + A bug that removes functionality of some number of links in the application or changes the links to incorrect urls.
  + A bug that has sets a product limit in the cart.
  + A bug that makes search bars case-sensitive.
  + A bug that makes a special screen effect occur when a cart price reaches over $1,000,000.

# Nonfunctional Requirements

* Improve website aesthetics, styling, etc.
  + Make the ui cleaner and easier to read.
  + Adhere to the company color scheme of purple, black, and white.
* Project should be built for use on a LAMP (Linux, Apache, MySql, PHP) stack
* Ensure that all form fields have unique IDs for easy development in the future.

# Feasibility

Due to the flexible nature of this application’s requirements, development of the highest priority features would result in a successful deployment. Redesigning the applications interface to meet customer expectations and expanding the implementation of the bug system within the application are the highest priority requirements for this project. In that sense, the bare bones version of the program would be a version with an updated visual design and some number of bugs implemented. The enhanced version of the program would include more optional bugs and a cluster of other features asked for by the customer. Among these, are wishlist implementation, email support, and promotional code support. To that end, the bare bones version of the program is undoubtedly achievable. While the enhanced full version of the application has numerous additional requirements, we are confident that it can still be completed by the deadline.

# Conclusions

When complete, this project will allow an admin to load bugs onto a normal user’s profile, in order to train that user in writing tests which account for bugs and other edge cases. The base of the project is already complete, and now it is our team’s turn to expand upon this base. The highest priorities for our customer were in website aesthetics, creating a usable wishlist, and then heavily expanding on the bugs that were available to load onto a user’s profile. With continuous work, we see few obstacles in the way of reaching the requirements of our customer.

# Appendices

Here is a link to our project website, which contains developer notes and information as the project progresses: <https://github.com/GreenAlex96/StorefrontTestingApp/wiki>

## Authors and Word Counts:

**Benjamin Ellis (350 words)-** Requirements, models, feasibility

**Alex Reel (1081 words)-** Content, requirements, models

**Alex Tanner (100 words)-** Editing, finalization

**Patrick Yoder (50 words)-** Document structure, planning