# WSU TA Management

## Design Document

10/23/2019

Version 1.0

**Team Kappa**

Joshua Green, Zach Penn, Mykhailo Bykhovtsev, Rodney Walton

Course: CptS 322 - Software Engineering Principles I

Instructor: Sakire Arslan Ay

# TABLE OF CONTENTS

# I. Introduction

The purpose of providing this design document is to track the progress, functionality, and details regarding the project (WSU TA Management). The information presented should provide a thorough understanding of how the system is being designed and what needs to be done. Even with no prior knowledge, someone new should be able to read through and continue the design of the system by themselves. This document will constantly be checked back on and revised, following and guiding us to the very end of the project. Main revisions will be done for each of the three iterations, and will include documentation of any changes.

The project presented in this document is a tool to enable teachers to easily find willing assistants for their classes. The final version will allow students to maintain a profile containing their contact information and course interests, and instructors to select a teaching assistant from those students.

Section II of the design document, Architecture Design, describes the software's architecture and components in great depth. This includes how the different pieces of the system interact, architectural patterns, and diagrams to help explain the design.

Section III of the design document, Design Details, goes deeper into the system. Looking at the subsystems, this section will explain how each of them will work and communicate, as well as provide information on the organization and inner workings of the software.

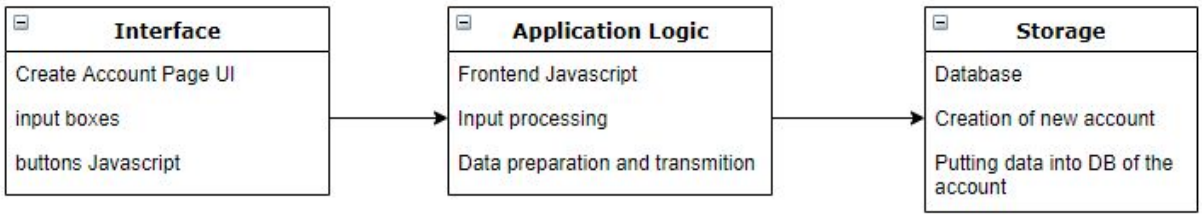
## Document Revision History

Rev.1 10/23/2019 Initial Version

# II. Architecture Design

## II.1. Overview

- The architecture pattern that we decided to go with is the Three-Tier Architectural Pattern because it is best suited for Websites and in our case.
- The subsystems currently have high coupling. Which we plan to reduce in future in iteration 2. But our coherence is high because of how our functions perform similar tasks.



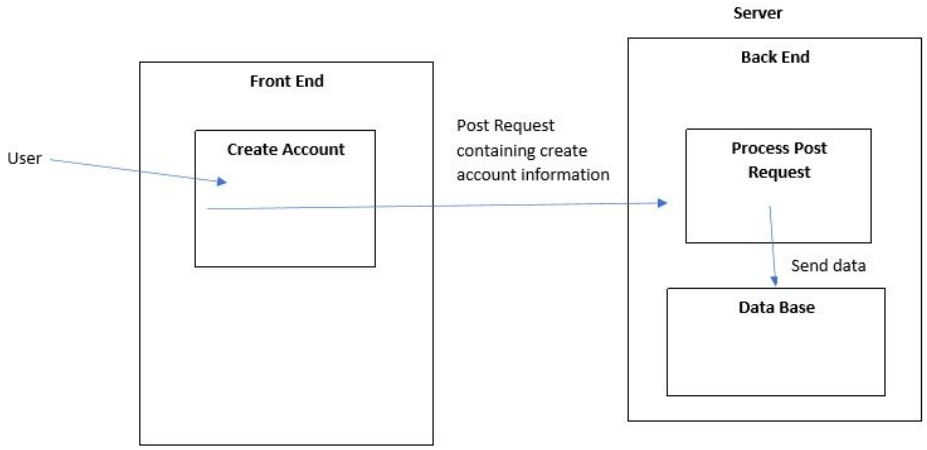

-
- Interface component is responsible to provide an interface with user. The application logic component is currently our Frontend Javascript code which processes the input and prepares and transmits data to the Storage(backend) component. In future Application Logic will also include Page generation by generating full page by grabbing specific page and adding header, footer and menu. The storage component of our architecture is our backend that handles

queries to the database. Currently it can receive information sent by application logic component and put it properly for storage into the database. It also includes password encryption.

# III. Design Details

## III.1. Subsystem Design

- Our current design of interface component is an HTML page for Create User Account which we try and keep generic for both Student and instructor by having them input common information such as first name, last name, WSU ID, email, phone number, password. Then we have an option to choose the account type. There is a submit button and a cancel button. By identifying the common input data that both users have we were able to figure out that we can have a single page for creating an account. The application logic system uses Jquery to find forms and grab input from them. Data are being prepared and sent to the right route for storage subsystem to receive. It is designed to check account type and based on it send it to the right route. Our backend is technically both application logic system and storage. Because the process of post requests is handled on the backend and addition of data into the database in handled there too. It is designed to use a special routes that are based on the account type. After it receives data it processes it and puts it into the database.
- The data that Database contains are of 3 types: Users, Applications, Classes. Users is a table that contains user information, which depends on user type: Student, Instructor. It is generic table where Instructor has less data than Student typically. Applications and Classes are to be worked in future. The data that we send from Frontend(Application logic system) are in JSON format which are the inputted user data. The data that we receive from Backend are also in JSON format which contains same inputted user data and status if we succeeded or not.



## I.1. Data design

We currently have 3 tables named User, Classes, and Applications. The Users table contains all users on the site, students and otherwise. The Classes table contains all the classes entered. The Applications table contains all of the applications submitted by students.

**Applications Attributes**

- Class id
- Student ID
- Student Name
- Student Last Name

**Classes Attributes**

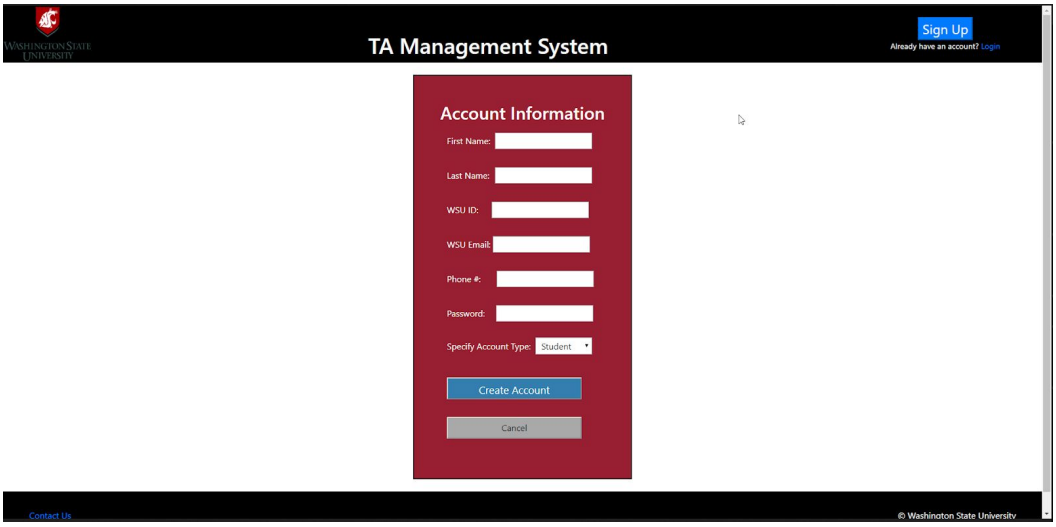
- Teacher ID
- Class Title

**User Attributes**

- User ID
- Student Name
- Student Last Name
- Teach or student boolean
- Password
- Email
- Phone
- Major
- GPA
- Experience

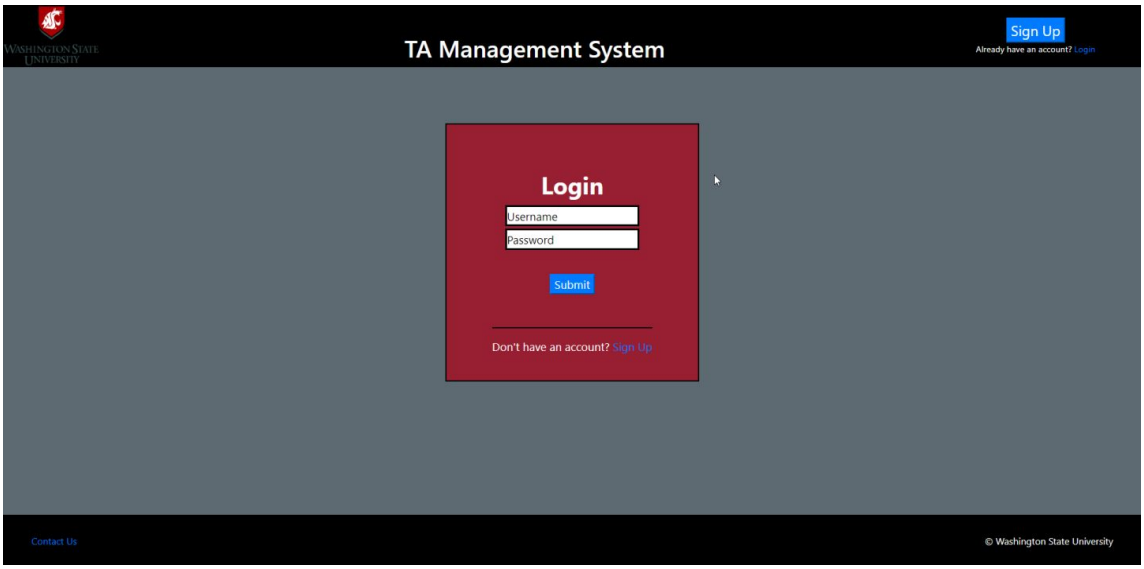
## I.2. User Interface Design

So far we've built drafts of a few of the use cases, along with a mock header and footer. First we've built the create a profile feature from the first use case:



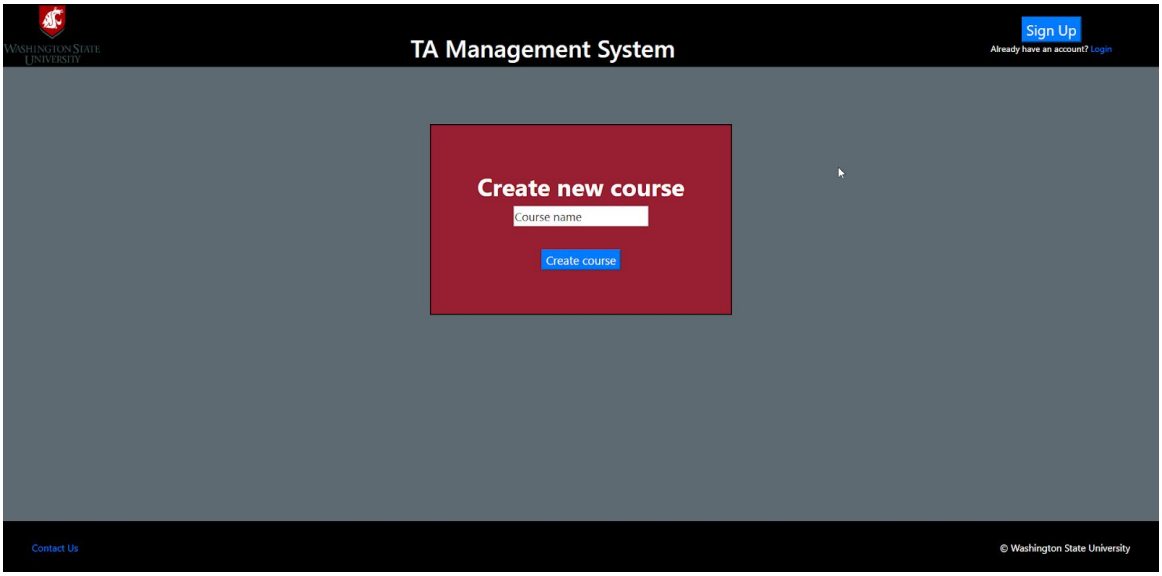

It allows the user to input relevant information regarding account creation and asking if they're a TA or a student. You can also see the start of our footer and header, which is currently more or less a place holder but will later grow to fit other links and will have better styling.

Next we've built the login feature from use case 2:

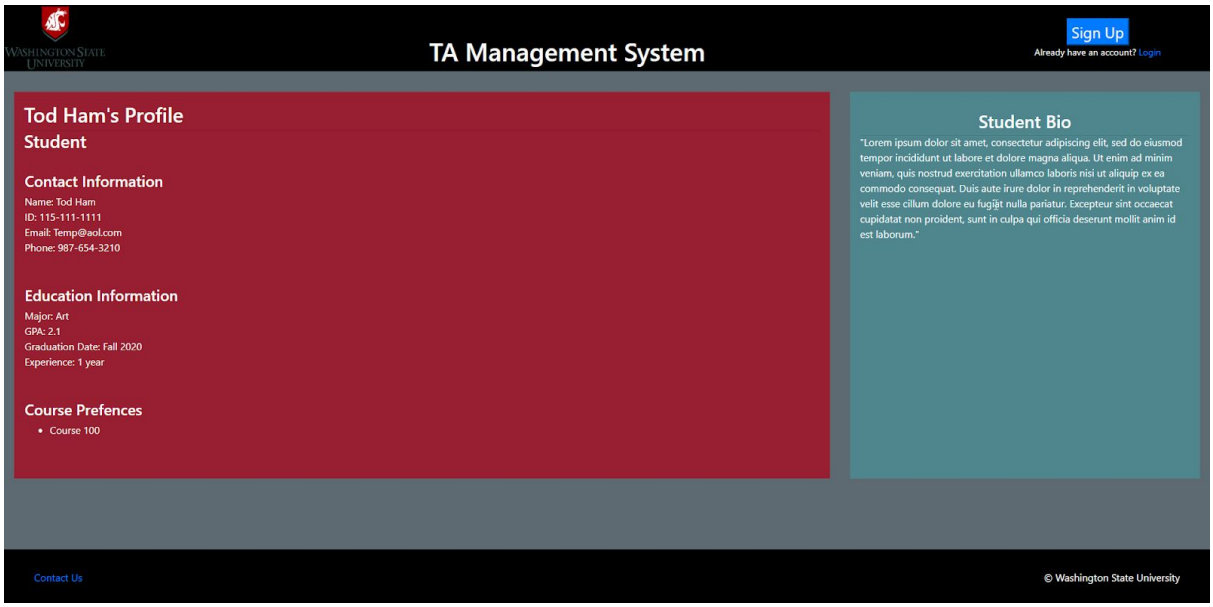


Takes in username and password and checks if valid, then logs them in. Also gives the user the option to click the sign up link on the bottom of the login box,  if the don't already have an account.

We've also created create new course feature from use case 6:




Allows the user(only teacher in this case) be able to add courses available that students can later apply as TA for. Finally we built the display profile feature from use case 3:




Displays users name, contact information, education information, and course preferences. Along with whether they're are TA and a bio.

## IV. Progress Report

The main difficulties of the project so far were more organizational rather than technical. While we haven't had many problems trying to create what we wanted, we did have some trouble deciding what we did want in specific places. This includes how we split up the work, and how it will come together. We had one person design the header/footer for our website, but there was difficulty trying to figure out how we are adding those to all of our pages. The features we wanted to implement for iteration 1 went smoothly and as planned. The only feature implemented fully (in frontend and backend) was the create profile feature. Overall in the frontend, the create profile, add class, login, and view profile are all we needed for iteration 1. Regarding these, there is still missing functionality that is yet to be implemented, but wasn't needed for iteration 1. There weren't really any tests we needed for this iteration, other than making sure the pages loaded correctly and the functionality (of create profile) worked as planned. For iteration 2, this section will include information on all the specific features implemented and how we went about testing the system.