**WSU Wants You as TA**

**Requirements Specifications**



**Team Bad Namers**

Jeonghyeon Woo

Kadir Nour

Blake McCollough

Course: CptS 322 - Software Engineering Principles I

Instructor: Sakire Arslan Ay

**TABLE OF CONTENTS**

I. Introduction.. 4

II. Requirements Specification.. 4

II.1. Customer, Users, and Stakeholders. 4

II.2. User Stories. 4

II.3. Non-Functional Requirements. 5

III. User Interface Requirements.. 5

# **I. Introduction**

The requested program shall match TA applicants to available TA jobs of their preference. Since the tasks are too complex to be done manually, our client wanted it be automated. Our client requested three essential functions. First, the program must have the applicant complete a survey. The survey is comprised of personal information like name, contact information and a selection of courses the student wants to TA for. Second, the program needs to provide instructors data of applicants. The data may be trimmed and organized for instructors’ convenience. Third, the program needs to notify applicants their assignments. The assignments would be displayed when the instructor hires the student. Along with these main functions, there may need to create additional functions that are for user API, data management, and backend system.

# **II. Requirements Specification**

* The application should first and foremost ask for a username and password. If the user does not have an account, they will press a “create account” button that redirects them to an appropriate account creation page
* If it’s the user’s first time using the application, contact/academic information should be provided as requirements before any other application features are used. The user may change information at any time via “change information” button. (function 1)
* If the user has indicated they are an instructor, they should have the ability to search for a course and view students who have requested to be a TA for the specified course. Instructors can assign TA positions to students or remove them. A student should not be a TA for more than one course. (function 2/3)
* If the user has indicated they are a student, they will be able to create/add to a list of courses they would prefer to TA. Course lectures may be specified, but not individual labs (the instructor will deal with what lab they get). Students can look at their preference list and see which labs they have been assigned to. Students should not be able to drop TA positions by themselves.

## **II.1. Customer, Users, and Stakeholders**

WSU, our client, ordered a software/web that manages the TA assignment system of the campus. The main user of this program will be the TA applicant and teaching faculty members. However, all students of the campus are potential TA applicants so the program needs to be accessible by every student of WSU.

## **II.2. Use Cases**

**Use case #1**

|  |  |
| --- | --- |
| Name | Create Account |
| Users | Students and Instructors |
| Rationale | Students and instructors who do not have an account can make one. This should ideally be their WSU username. |
| Triggers | The user presses “create account” and may provide a username, password, and teacher/student |
| Preconditions | Should be a student or instructor |
| Actions | 1. User requests to create an account 2. Application redirects to provide username, password, and teacher/student 3. User enters a unique username, a password, and indicates if they are a student or instructor 4. If username is not unique, tell the user to choose another username 5. Username, password, and student/instructor are initiated and logs in |
| Postconditions | The requested username will be accepted as a valid login |
| Acceptance Tests | Makes sure user can log in successfully with their username |
| Iteration | 1 |

**Use case #2**

|  |  |
| --- | --- |
| Name | Sign in |
| Users | Students and Instructors |
| Rationale | Students and instructors can log in using their specific username and password. This is to differentiate between who is using the application. |
| Triggers | User indicates their unique username and password associated with their username. |
| Preconditions | Should be a student or instructor who have already created an account |
| Actions | 1. User indicates their unique username to system 2. User indicates their associated password to system 3. System will check if the given password is acceptable 4. If password is correct, the system will redirect to the user’s application 5. If password is incorrect, tell the user that the password is incorrect |
| Postconditions | The user will be redirected to the application |
| Acceptance Tests | The correct application is provided with the user’s input |
| Iteration | 1 |

**Use case #3**

|  |  |
| --- | --- |
| Name | Enter Information |
| Users | Students and Instructors |
| Rationale | Students and instructors should be able to have their contact and academic information stored in a database associated with their username. This is to help with filtering and let students/instructors find the appropriate contact information. |
| Triggers | User will press “change information” button, or if it’s the user’s first time logging in without saving their information |
| Preconditions | Should be a student or instructor |
| Actions | 1. If the user is a student, they will indicate their first/last name, WSU ID, email, phone number, major, GPA, graduation year, and if they’ve been a TA before 2. If the user is an instructor, they will indicate their first/last name, WSU ID, email, phone number 3. User will press save button, which saves the indicated information to the associated username |
| Postconditions | Provided information is saved to the user’s username |
| Acceptance Tests | The information is successfully linked with the associated username |
| Iteration | 2 |

**Use case #4**

|  |  |
| --- | --- |
| Name | Show current status |
| Users | Students |
| Rationale | Students must be able to check their current status: preferred courses, assigned courses, etc. |
| Triggers | The user press “current status button” |
| Preconditions | The user must be signed in as student (TA applicant) |
| Actions | 1. User may go back to the main page or start a new application. 2. User will see their preferred courses 3. User will see their assignments 4. User will see any additional infos 5. User will remain in the page unless any special action (action 1) is taken |
| Postconditions | The user remain in the page if there is no special action |
| Acceptance Tests | Make sure the contents are visible to the user |
| Iteration | 2 |

**Use case #5**

|  |  |
| --- | --- |
| Name | Start application |
| Users | Students |
| Rationale | Students will complete a survey to apply for TA. This helps put on paper what classes the students is looking at as well as giving instructors information is the student’s qualifications. |
| Triggers | The user presses “start application” and provides the appropriate information regarding the courses that interest them. |
| Preconditions | Should be a student and have information stored with their username. Student should also have passed the course they plan on becoming a TA for. |
| Actions | 1. User at any point may cancel their course preference application 2. User will specify the course number they are interested in TA’ing 3. User will specify the grade they received for the specified course 4. User will specify the semester they passed the course 5. User will specify the semester they plan on TA’ing 6. User will specify if they’ve been a TA for the specific course 7. User will press save button and the indicated information will be linked with their username |
| Postconditions | The system should add the course preference to the user’s data |
| Acceptance Tests | The course preference is added to the user’s data |
| Iteration | 3 |

**Use case #6**

|  |  |
| --- | --- |
| Name | Courses Instructing |
| Users | Instructors |
| Rationale | Instructors will specify the courses they are instructing that they are accepting TA’s for. This will be an organization tool for the instructor so they can easily see who is or is requesting to be a TA for a course they instruct. Each lab should be listed as a different course. |
| Triggers | The user presses “Instructing Course” and chooses which course they are accepting TA’s for |
| Preconditions | Should be an instructor and have information stored with their username |
| Actions | 1. User should specify which course lecture they are instructing, as well as every lab associated with that course 2. For whichever course lecture they select, a list of students who have chosen this as their preference will be displayed. Students already assigned to a TA position will be highlighted 3. For whichever course lab they select, the assigned TA will be displayed. If no TA has been assigned, the system will indicate this 4. User may be able to assign any particular student for the preference list to a specified lab section. 5. User may remove a student from a TA position at any time by pressing “remove” under their course |
| Postconditions | Updates to the list will be adjusted as necessary |
| Acceptance Tests | Makes sure student’s addition or removal from a TA position is successfully saved |
| Iteration | 3 |

## **II.3. Non-Functional Requirements**

1. Database password storage: The system shall not store user passwords in their raw form
2. Data integrity: The system shall not divulge any user information
3. Platform access: The system shall be compatible with all major web browsers: Chrome, Firefox, IE/Edge, Safari, etc.
4. Response time: Loading the system shall not take longer than 10 seconds
5. Scalability: The system shall be scalable to accommodate unrestricted growth depending on the number of instructors in need of a TA

# **III. User Interface Requirements**



The main page will show the courses that are currently in need of a TA. There will be buttons for signing in and to create an account. When sign in is clicked, the hidden form for signing in will be visible. When the create account button is clicked, the hidden form for creating an account will be shown. It will not be possible for both forms to be visible at the same time.

After signing in, the user can change their contact information or work on their TA applications. When the user starts working on their TA application, there will be 2 cases. If the user is a TA applicant, the user will be redirected to the page showing their current status. In the page, the applicant can see their current status and start a new application. If the user is faculty member, the user will be redirected to the page showing the list of courses that the faculty member is managing. The faculty member can assign or dismiss any TA in this stage.