

EECS TA System

Requirements Specifications



teamBrevin

Brevin Simon

Course: CptS 322 - Software Engineering Principles I

Instructor: Sakire Arslan Ay

TABLE OF CONTENTS

I.	INTRODUCTION.....	3
II.	REQUIREMENTS SPECIFICATION	3
II.1.	CUSTOMER, USERS, AND STAKEHOLDERS	3
II.2.	USE CASES.....	3
II.3.	NON-FUNCTIONAL REQUIREMENTS.....	6
III.	USER INTERFACE REQUIREMENTS	6
IV.	REFERENCES.....	7
	APPENDIX: GRADING RUBRIC	ERROR! BOOKMARK NOT DEFINED.

I. Introduction

We would like to create an application where teacher assistants can apply for TA positions via an online form and the professors of the courses, they apply are able to review the forms and select TA's for their courses. The goal is to make the process easier for professors to match TA's with courses while meeting the criteria for each individual course. This application will essentially have two sides, one of the student teacher assistants to apply and manage their TA applications and one for the professors to manage the TA's for the courses they are instructing.

In Section II, we will analyze the project's starting requirements and go in depth on the roles behind the project such as specifying the stakeholders, the customer, and the users. While also constructing and specifying the use cases for the user's stories. This section will be concluded with the non-functional requirements that will illustrate the constraints that our design will have to follow to meet the quality needed for our application to be successful. In Section III, we will start to look at the functional requirements and specify exactly what the system should implement and what the user will see in comparison to this implementation.

Document Revision History

Rev 1.0 <10/14/20 > <Initial requirements document>

II. Requirements Specification

In this section you will describe the features, functions, and other specifications that are requirements for your product.

II.1. Customer, Users, and Stakeholders

The customer of the application is the school of EECS at WSU where every professor and teacher assistant under the EECS will benefit from the application as they will be working under this system. Thus, the stakeholders are the professors and the students who apply to be a TA. The user's will also be the professors and the students who apply to TA since they will be using the application. The professors need the ability to manage the TA's for all of their courses and the TA's need to be able to apply to TA for any of the courses they choose and they also will be able to manage the courses they have applied for.

II.2. Use Cases

From the student applying to be a TA, the user should have access to all open TA positions and can submit a form for which ever they choose.

From the professors teaching courses, the user should have access to the courses they are instructing and be able to see TA's applications for each of those courses. Here the user can select TA's for their courses.

Both the student and professors will have the ability to create an account and log-in. This account will allow them to enter the necessary data for their position (TA or professor).

Use case # 1

Name	Apply to TA for a Course
Users	Students
Rationale	The current system for professors to find and match TA's to courses is too tedious and can be made more automatic. The "Apply to TA for a Course" function will give the user a list of open TA positions and allow them to apply for any of the open positions. When applying they will receive a form that will allow them to fill out the required information.
Triggers	Once the student logs into their account this information will be available to them
Preconditions	Student creates account and/or logs-in
Actions	<ol style="list-style-type: none"> 1. When the student is logged in 2. The software knows that a student is logged in and responds by displaying open TA applications 3. Once the student selects a course, they are given a form to give the required information
Postconditions	Form is accepted/added to system (redirect Actions-Step-2-page)
Acceptance Tests	Confirm that the form the Student submitted has been added to the system
Iteration	Implemented in iteration 1

Use case # 2

Name	Add TA to Course
Users	Professors
Rationale	The current system for professors to find and match TA's to courses is awful and can be made automatic. The "Add TA to Course" function will allow the user to see the courses they are instructing and give the user the ability to add TA's to their courses by looking through the forms previously submitted.
Triggers	Once the professor is logged into their account list of their courses will be available to them
Preconditions	Professor creates account and/or logs-in
Actions	<ol style="list-style-type: none"> 1. When the professor is logged in 2. When the software knows the professor is logged in it will respond by displaying all the professor's courses, he/she is instructing 3. The user can select a course and display the TA forms for the TA's applying for that given course <p>Note: Base case it just displays the classes, only if specified will the user add the TA for his course in Alternative Paths</p>

Alternative Paths	<ol style="list-style-type: none"> 1. In Actions-step3 the user can select to add a TA for the course if he/she so decides 2. In Actions-step3 the user can select to delete a TA for the course if he/she so decides
Postconditions	Professor can view classes and if TA is added then add to system (redirect Actions-Step-2-page)
Acceptance Tests	Confirm the TA is added to course under classes taught by the user in the system
Iteration	Implemented in iteration 2

Use case # 3

Name	User Account Creation/Login
Users	Professors and Students
Rationale	The current system for adding TA's to courses is not feasible especially with the use of Email there can be a lot of problems. The "User Account Creation/Login" function will allow the user to create an account to meet their needs inside of the system. The user must choose whether they are creating account as a professor or as a student.
Triggers	Once user selects to "Create an Account" or to "Log-in"
Preconditions	Complete base/index of page (No log-in required)
Actions	<ol style="list-style-type: none"> 1. User is prompted to create an account as professor or as student or log-in 2. When user chooses to create an account, they are prompted for the information they need to complete account 3. System responds by giving them the appropriate homepage for their role (professor or student)
Alternative Paths	<ol style="list-style-type: none"> 1. In Actions-step1 if user already has an account, they can login and bypass Actions-step2 into Actions-step3
Postconditions	When account is created add to system and redirect to the given user's homepage i.e. Use Case #1 page or Use Case #2 page
Acceptance Tests	Confirm new user is added to the database
Iteration	Implemented in iteration 3

II.3. Non-Functional Requirements

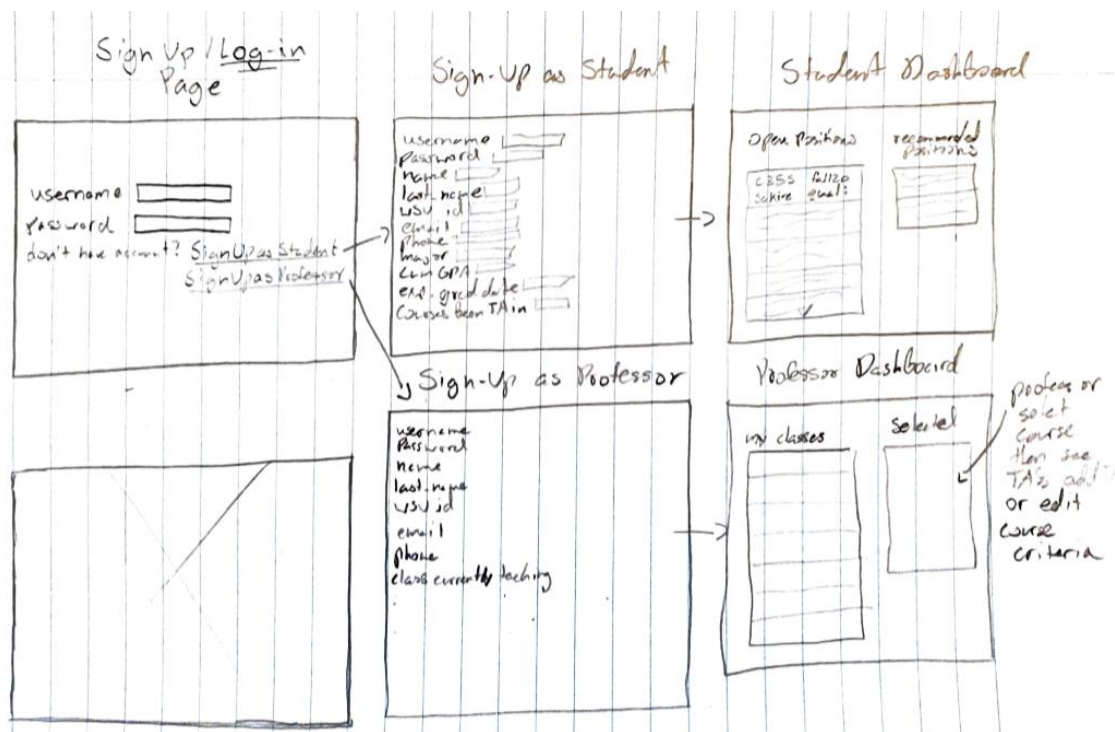
List the non-functional requirements in this section.

You may use the following template for non-functional requirements

1. Response Time: The system should add applications/TA's/accounts to the system quick and uneventfully
2. Security: The system account creation process i.e. username and password, must be safe and not have the possibility for leaks
3. Allowances for reusability: The user (professor or student) should be able to login to a pre-existing account to access their data without fail
4. Scalability: The system should have the ability to become very large scale without the possibility of occurring errors based on its growth
5. Environmental: The system interface should be able to adapt to smaller screens such as iOS/Android

III. User Interface

The user should be able to create an account for either professor or student and receive the corresponding form for their application. While their account is of the specified type being student or professor, they will get the corresponding dashboard in the system. For example, the TA's can view what courses they have applied for and the professor will be able to add TA's for whichever course he/she has selected. Implemented such that you can login without specifying the user role (professor or student).



IV. References

Eriksson, Ulf. "Functional vs Non-Functional Requirements - Understand the Difference." *ReQtest*, 28 July 2020, reqtest.com/requirements-blog/functional-vs-non-functional-requirements/.

Lynch, Warren. "All You Need to Know About Use Case Modeling." *Medium*, Medium, 22 Mar. 2019, medium.com/@warren2lynch/all-you-need-to-know-about-use-case-modeling-828756da3215.