Term Project

Requirements Specifications

SpaghettiCoders

Jimmy Zheng,

Hien Duong,

Peng-Yu Chu,

Eric Furukawa

Course: CptS 322 - Software Engineering Principles I

Instructor: Sakire Arslan Ay

Table of Contents

I. Introduction

The purpose of this project is to build a web application that will allow Professors to find teaching assistants. The objective of this project is to allow Professors who are looking for teaching assistants to be matched with students they are looking for. Students who are hoping to become teaching assistants will be able to post their information for Professors to find them. This project will allow us to apply software engineering principles. Section II will include the requirements specification, this section will describe the features, functions, and other specifications that are requirements for the products. Section III will include the user interface requirements; this section will have mockups and designs of what the finished product will look like. Section IV will include any references.

Document Revision History
Rev 1.0 2019-10-01 Initial Version

II. Requirements Specification

II.1. Customer, Users, and Stakeholders

The customer is our instructor of this course, she will tell us the specification and requirements for the product. The stakeholders are the developers, the customer, the user, the University. The users of our software will be the Professors and the students.

II.2. Use Cases

On the student page, a student user can:
1. Create a student account and set the account password (username should be the WSU email)
2. Login with username and password
3. Enter contact information (name, last name, WSU ID, email, phone)
4. Enter additional information (major, cumulative GPA, expected graduation date, whether s/he served as a TA before)
5. Enter course preferences for the TAship. A student can specify more than one course for her/his preferences. For each course the student should enter:
a. the course number (such as CptS 121)
b. the grade s/he earned when s/he took the course (e.g., the grade earned for CptS121)
c. the year and semester s/he took the course
d. the year and semester s/he applies for TAship
e. whether s/he served as a TA for this course before (yes/no)
Student users can add/remove courses to/from their list of preferred courses.
6. Save the entered information.

On the instructor page, a faculty user can:
1. Create an instructor account and set the account password (username should be the WSU email)
2. Login with username and password
3. Enter contact information (name, last name, WSU ID, email, phone)
4. Enter the courses s/he teaches currently. Of course, an instructor can teach more than one course and /or lab section during a semester. (If a particular course has more than one lab section, each of those section should be entered as a separate course. For example: CptS121-Lab1, CptS121-Lab2…etc.)

5. See the list of the students who applied for TAship. The list should show whether the students have already assigned to a course.
6. The instructor can select a student (who is not assigned to a course yet) and assign it to one of the courses/labs s/he teaches. S/he may also remove the current TA assignments for his courses/labs.

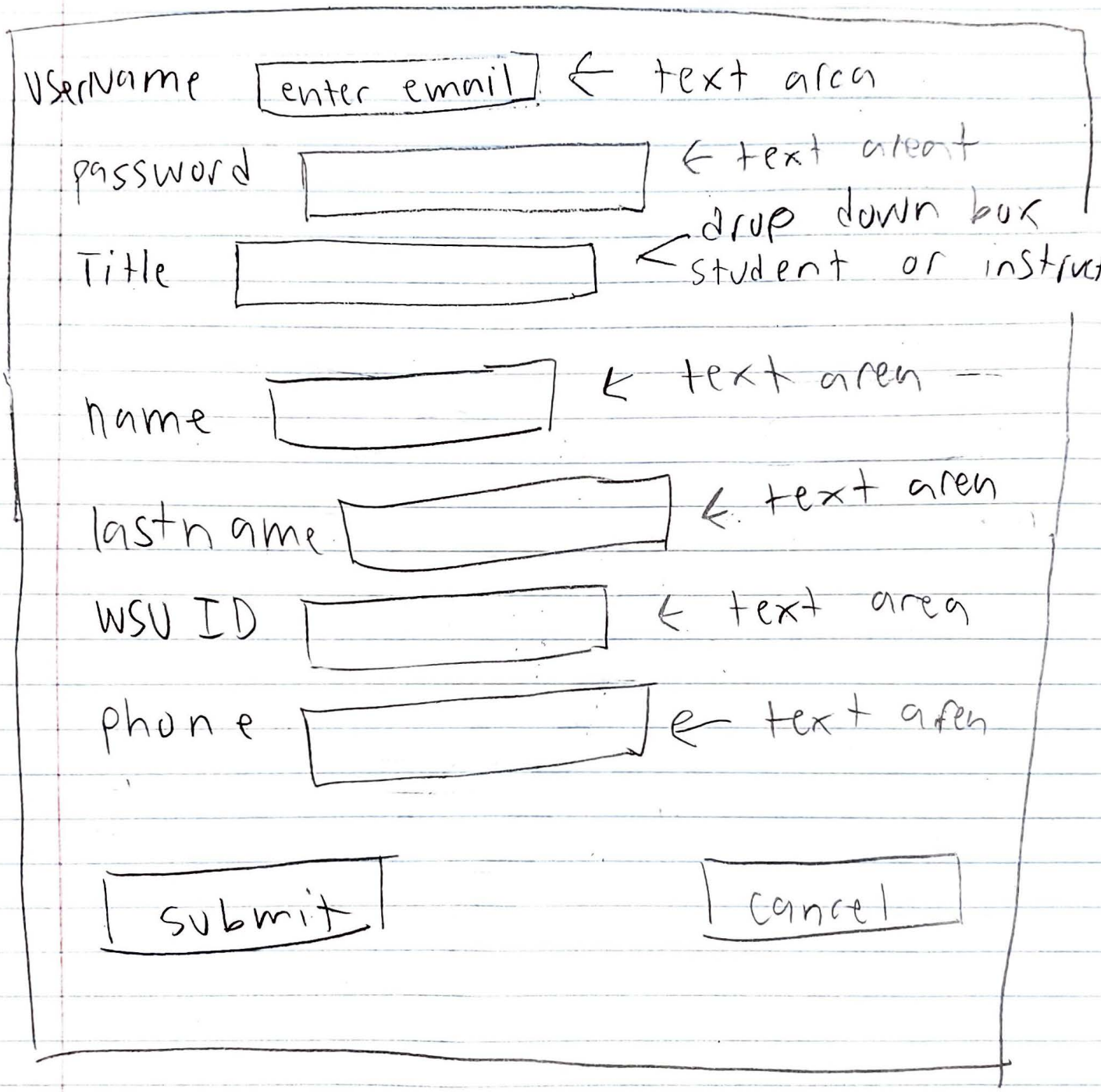
Use case # 1

Name	Create Instructor Account
Users	Instructor
Rationale	The user can create an instructor account and set the password for the account.
Triggers	The user selects “Create Instructor Account”
Preconditions	A website is loaded for the user to input data
Actions	1. The user indicates that the software is to perform the create in the website. 2. The software responds by requesting the inputs of the account and password information. 3. The user inputs the account and password text and indicate the information to submit. **4.** The software store the account information in the database.
Postconditions	The instructor account is created.
Acceptance Tests	Make sure that the instructor account is created correctly.
Iteration	1

Use case # 2

Name	Create Student Account
Users	Student
Rationale	The user can create a student account and set the password for the account.
Triggers	The user selects “Create Student Account”
Preconditions	A website is loaded for the user to input data
Actions	1. The user indicates that the software is to perform the create in the website. 2. The software responds by requesting the inputs of the account and password information. 3. The user inputs the account and password text and indicate the information to submit. 4. The software store the account information in the database.
Postconditions	The student account is created.
Acceptance Tests	Make sure that the student account is created correctly.
Iteration	1

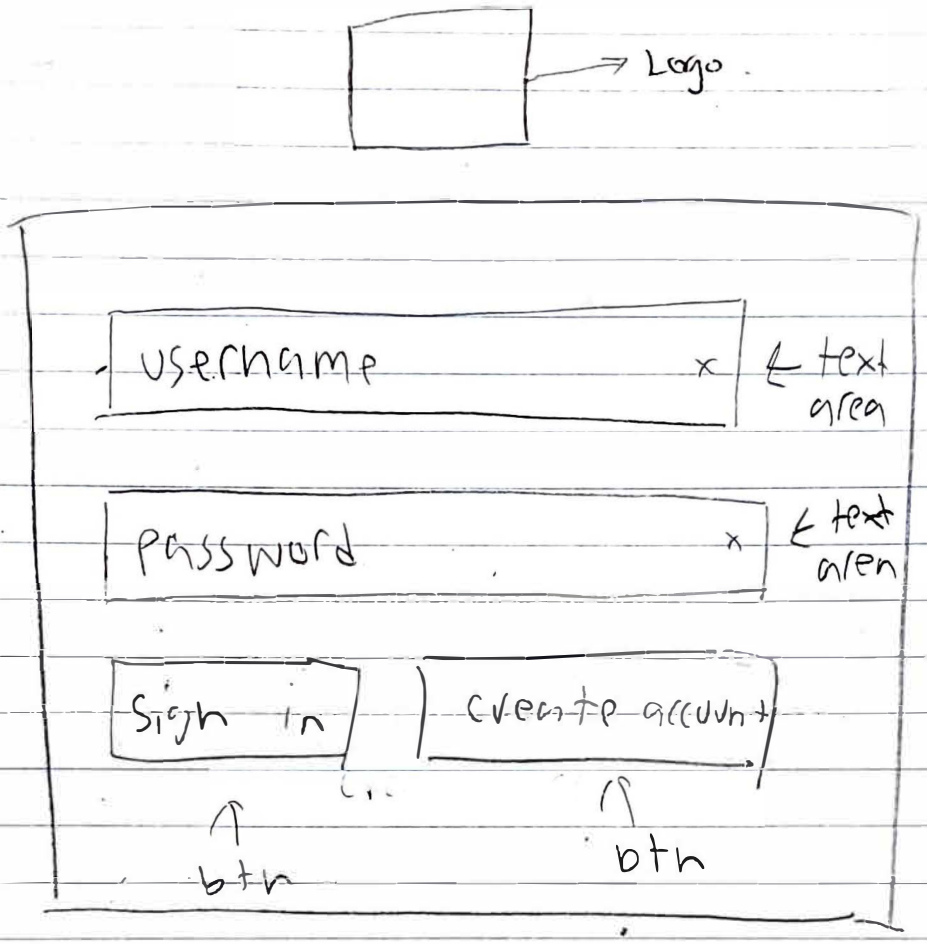
create account



Use case # 3

Name	Login Account
Users	All users
Rationale	The user login to the website with their own accounts and password
Triggers	The users select “Logic Account”.
Preconditions	Data are loaded for website for user to login.
Actions	1. The user indicates that the software is to perform login function in the website. 2. The software responds by requesting the inputs of the account and password information. 3. The user inputs the account and password text and indicate the information to submit. 4. The software search the information in database and log the users into the website.
Postconditions	The users are logged into the website
Acceptance Tests	Make sure the right accounts the users are logged in.
Iteration	1

Login page

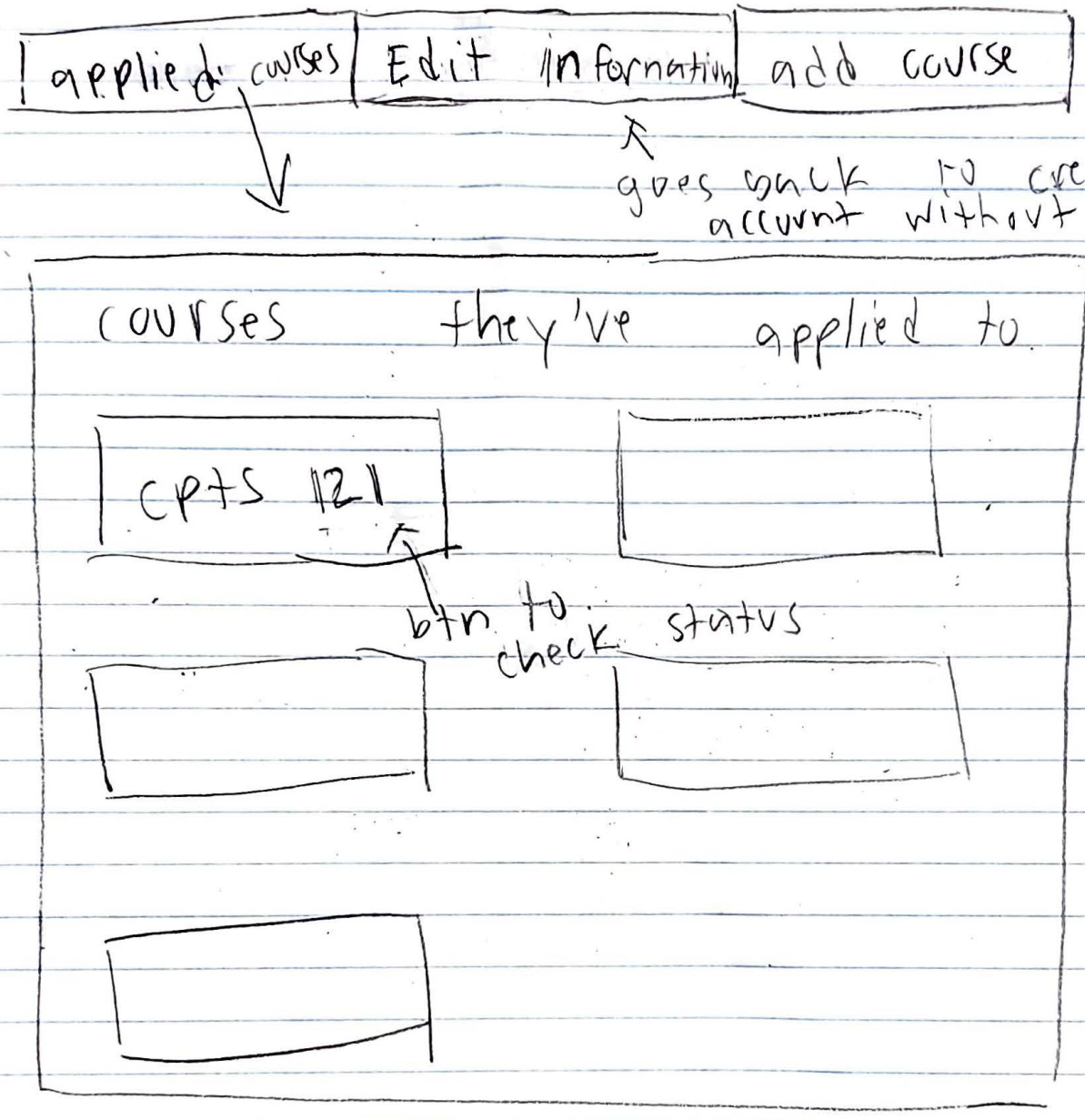

Logo
username
x
text area
password
x
text area
Sigh in
create account
btn
btn

Use case # 4

Name	Enter Course Preferences
Users	Students
Rationale	When students use the application, they will want to provide course preferences to ensure professors of those courses are the ones they are visible to.
Triggers	User selects "Change Class Preferences"
Preconditions	The student is logged in
Actions/Alternate Paths	1. The user enters a course name, grade received, date when daten, date when applied for TAship and whether previously TA'd for the class 2. The program adds the information entered to a list of preferred courses associated to the user 3. Prompt user to add another preferred course Alternate Path 1. Unless if the user specified to delete instead, in which the course selected will be removed
Postconditions	Selected course preference has been added or deleted
Acceptance Tests	Check that course preference has been successfully added or removed in the user's preferred list
Iteration	2

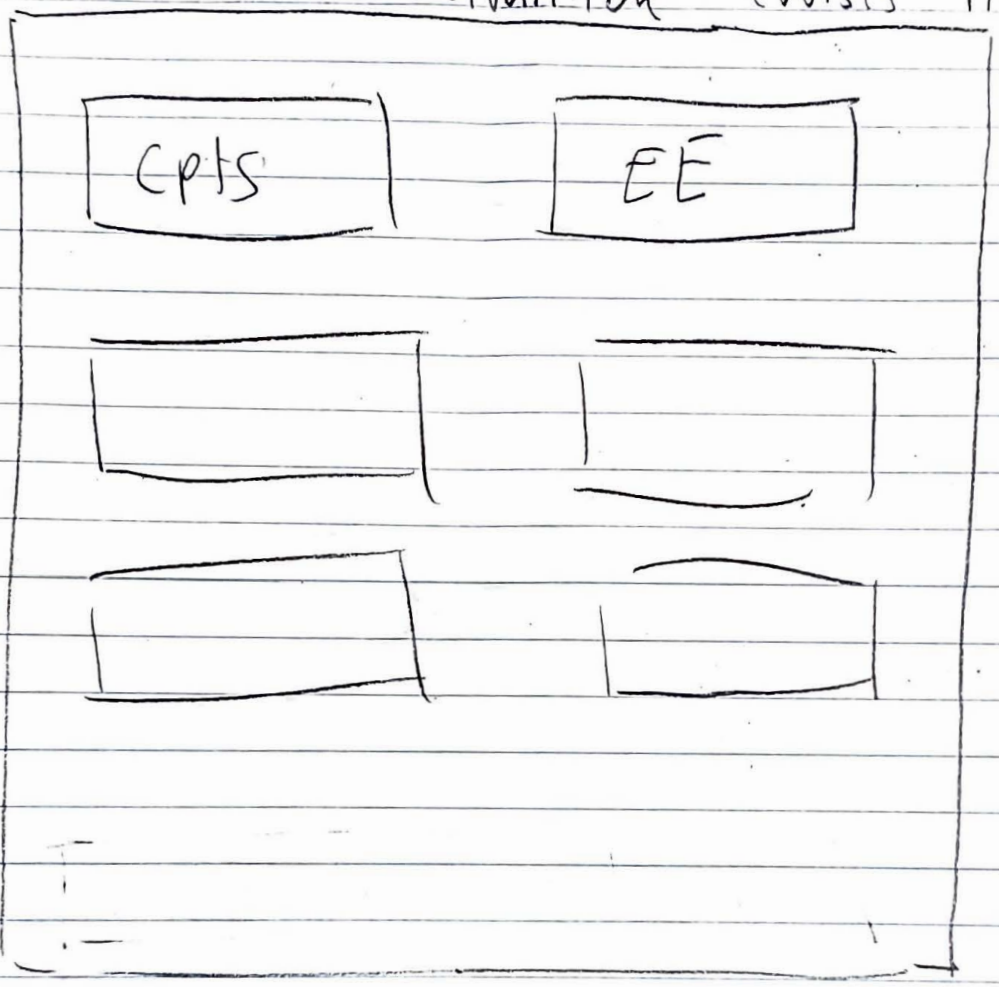
What Student Sees

HEADER LOGO



add course

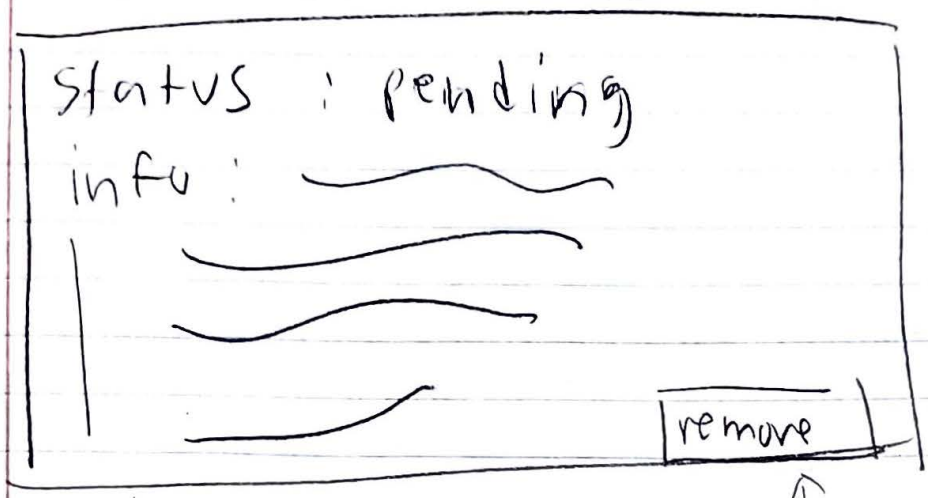
shows list of available courses profs need



Use case # 5

Name	Enter Contact Information
Users	All users
Rationale	Enter contact information (name, last name, WSU ID, email, phone)
Triggers	User selects “Enter Contact Information”
Preconditions	The user is logged in
Actions	1. The user enters their information (name, last name, WSU ID, email, phone)
Postconditions	The information is attached to the account
Acceptance Tests	Make sure the entered information is submitted correctly.
Iteration	2

check status

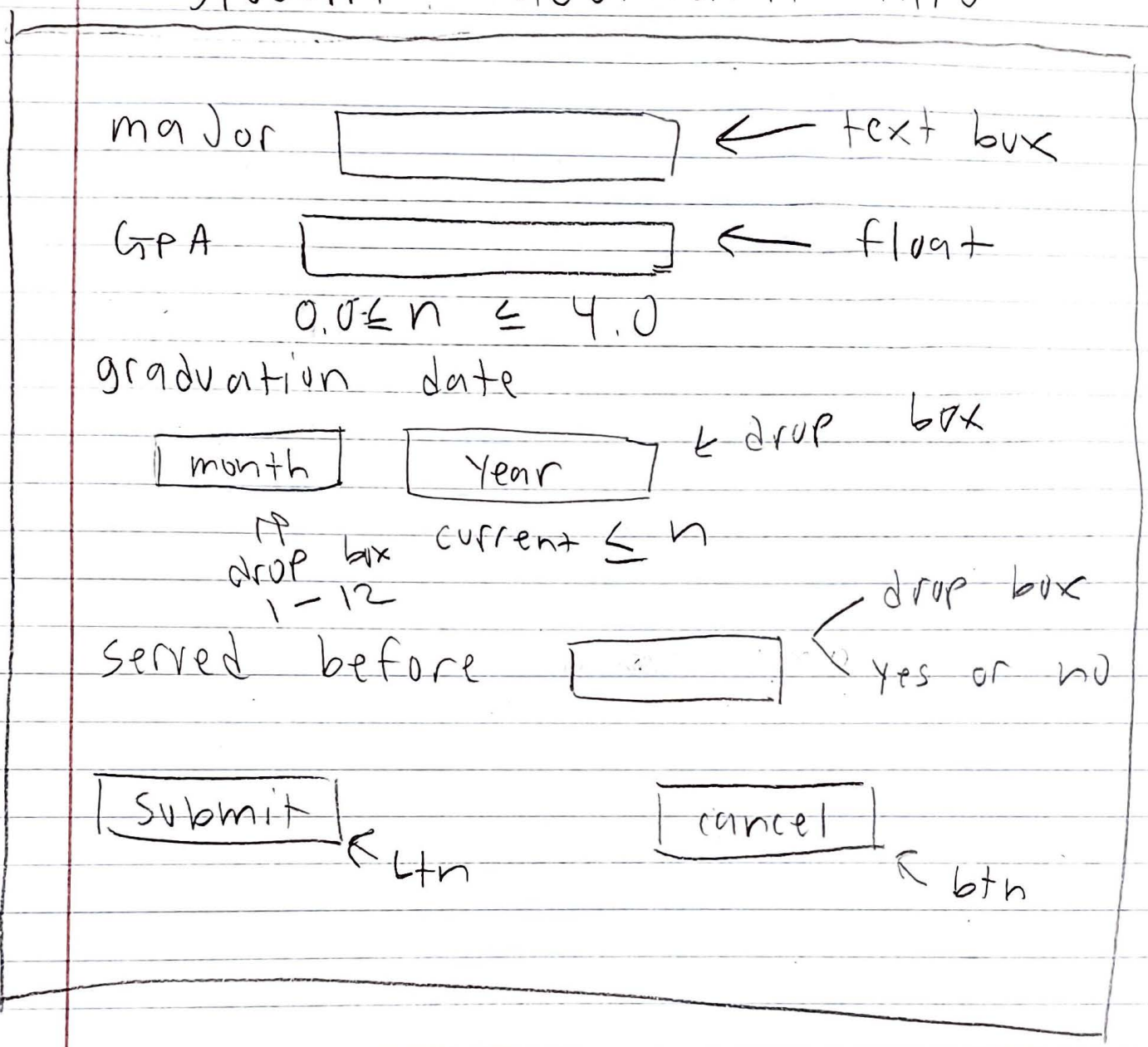


btn to remove curse

Use case # 6

Name	Enter Additional Information
Users	Students
Rationale	The users can enter more information into their profiles
Triggers	The users select “Enter Additional Information”
Preconditions	The information site is loaded and opened
Actions	1. The user indicates that the software is to perform the enter additional information. 2. The software responds by requesting the input on major, cumulative GPA, expected graduation date, whether s/he served as a TA before. 3. The user inputs the text of the information.
Postconditions	The information is attached to the account
Acceptance Tests	Make sure all the information are stored in the account correctly.
Iteration	2

Student additional info



Use case # 7

Name	Save Entered Information 
Users	All users
Rationale	Saves the entered information
Triggers	User selects “Enter Contact Information”
Preconditions	The user is logged in
Actions	User enters their information (name, last name, WSU ID, email, phone) 1. Users submit information 2. Software stores information with account
Postconditions	Information is saved
Acceptance Tests	Make sure that information is correctly saved to databse
Iteration	2

Use case # 8

Name	Enter Course and Lab Section Instructor Teaches 
Users	Instructor
Rationale	The Instructor should be able to enter courses that they teach and any corresponding lab sections associated with that course.
Triggers	Button to allow instructor to add courses
Preconditions	Logged in as professor
Actions	1. User clicks on “add course” button 2. User will be able to type and a list of courses will appear like a search 3. User can select what courses they teach 4. For the course selected they will be able to add lab sections to that course
Postconditions	Information saved to database
Acceptance Tests	Make sure that button works, and search works, and instructor will be able to add courses and labs correctly. And that the information is correctly saved to the database
Iteration	3

Course preference.

HEADER Logo.

My Course	Course preference	Edit Applicati

Course teaching: [CPTS 121] → drop down box

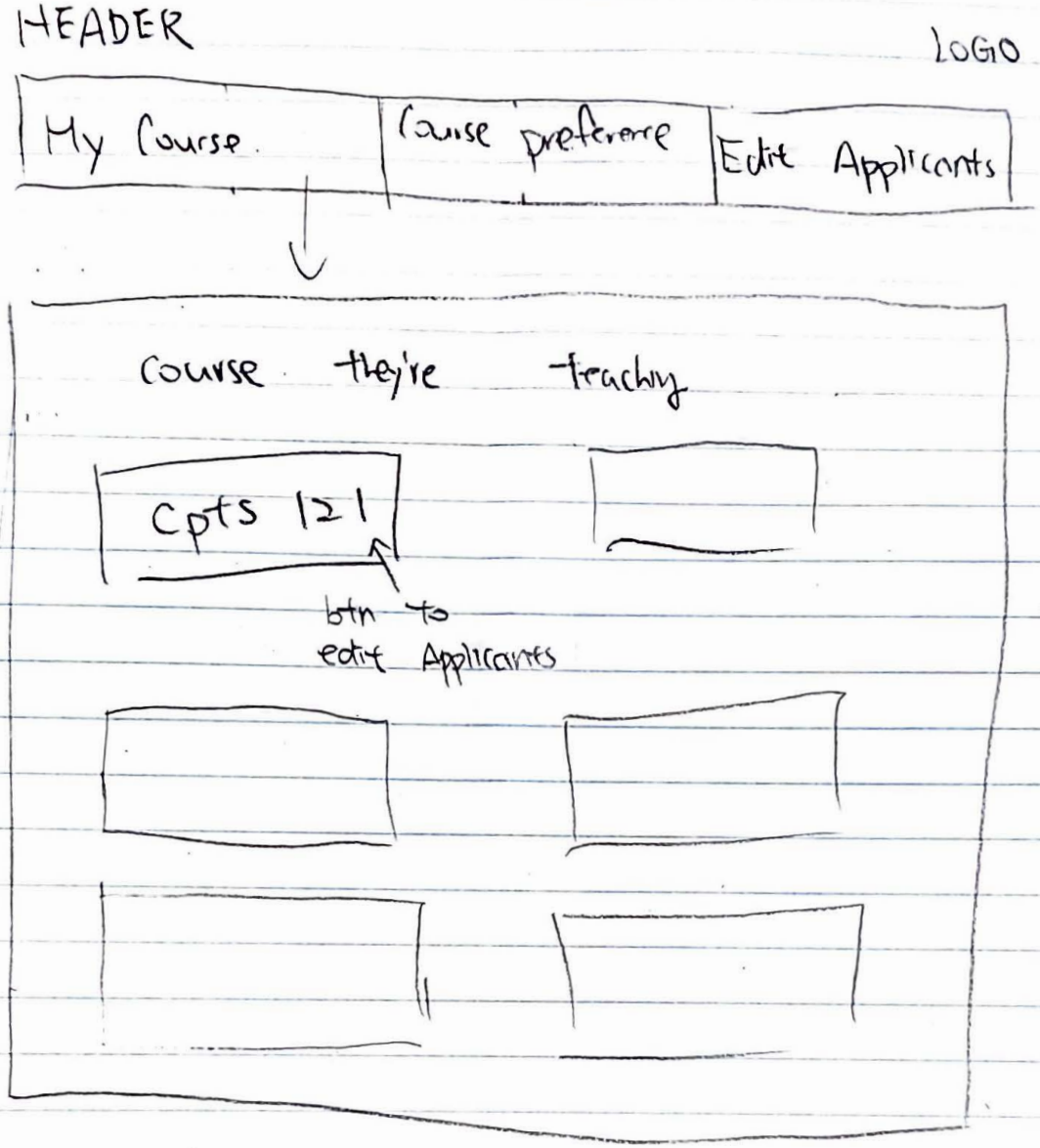
Lab section: [ ] → drop down box.

[Submit] [Cancel]

Use case # 9

Name	Display list of applicants per class
Users	Professor
Rationale	The instructor should be able to see an organized list of TA applicants (including if they have a TAship yet) for each of the classes they teach so choosing a TA is shorter and easier
Triggers	User selects a class to view applicants for
Preconditions	User is logged in as a professor
Actions	1. User clicks on a button for the specified class 2. Program gets TA data to display each under that class 3. Program displays data in specified sorting method
Postconditions	The application successfully displays all TA applicants for the specified course and if they have a TAship or not yet
Acceptance Tests	Make sure a professor account correctly displays applicants for their courses
Iteration	3

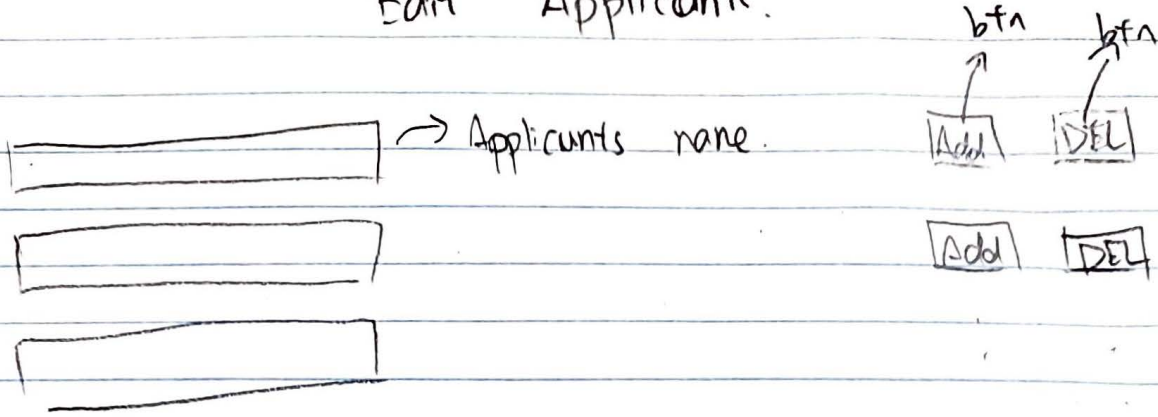
What Instructor sees



Use case # 10

Name	Choose/Edit applicants
Users	Instructors
Rationale	The user can select and edit applicants to the course that he/she is teaching.
Triggers	The user selects “Choose/Edit Applicants”
Preconditions	The list of the applicants of the course is shown.
Actions	1. The user select “Choose/Edit Applicant” 2. The software displays the list of the applicants of the course. 3. The user select applicant to the course. 4. The software added the selected applicant information to the course. 5. The user delete applicants to the course. 6. The software delete the selected applicant information from the course.
Postconditions	The applicants’ information is stored in the course.
Acceptance Tests	Make sure the applicants’ information are stored correctly to the course.
Iteration	3

Edit Applicants.



Submit

Cancel

II.3. Non-Functional Requirements

1. Reliability: Users should be able to consistently use the program without issue during any time outside maintenance or enhancement periods

2. Response Time: Page and data loading musnt take more than 20 seconds after the request is made by the user

3. Security: All user account information must be non-accessible to all other users or outside parties

4. Maintenance: Betterment of the program should not affect the uptime of the program by more than 24 hours a week after launch

5. Robustness: Application should support use on chrome, both Mac and PC

6. Scalability: Application should be able to support expansion to multiple schools and increased functions

7. Environment: Program must use python, javascript and flask/sqlalchemy for its implementation

8. Enhancement: Program enhancement should not step outside the boundaries of letting Professors make informed decisions on TAs and allowing TA applicants improve visibility to professors

9. Throughput: Program and backend should allow for at least 100 concurrent users (of which could hypothetically request at the same time)

10. Process: Program development must follow the agile SCRUM process with an assigned scrum master and professor oversight

11. Deployment: Application must have a final deployment by the specified project deadline

III. User Interface Requirements

User interface should be functionally based but also have a degree of cleanliness and visual appeal. Customers should not be confused on getting to specific pages.

IV. References

Google, google.com
Blackboard, learn.wsu.edu
MyWSU, my.wsu.edu
RateMyProfessor, Sakire Arslan, (Class Project)