Software Requirements Specification

for

Exam Management System

Version **1.0 Draft**

Prepared by

Emami, Chris
Goel, Shalini
Peixoto, Mario
Zhang, Xuezhu

March 9, 2016

Table of Contents

Revision History

Name	Date	Reason for Changes	Version
Draft	03/02	Initial document	1.0

1. Introduction

1.1. Purpose

The purpose of this document is to present a detailed description of the Exam Management System. It will explain the functions, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both users and developers of the system and will be submitted to Professor Xiao Chen as a requirement for the CS5391 Survey of Software Engineering on Spring 2016 semester.

1.2. Scope

The Exam Management System is a software system that will be used by the faculty members and students of the Computer Science department of Texas State University. The system will help the faculty members and students to track the student progress and determine when and if the student can graduate according to graduation requirements.

More specifically, the system is designed to allow the faculty members to login and create the core, programming, and communication exams. Once the exams are created, the faculty member can view the details of the registered students, edit and view each exam details and publish the results. The system is also designed to assist a faculty member to manage student record information. The system will allow faculty member to add, edit, view, and search the student record information, including the exam results. The system will also allow the faculty to view all students that have completed all the graduation requirements, therefore eligible to graduate.

The system is also designed for students, who can login, register or withdraw from an specific exam, and view its results for all exams.

1.3. Definitions, acronyms, and abbreviations

Term	Definition
Student	A student of the Texas State University with access to the Exam Management System.
Faculty	Professors from the Texas State University with access to the Exam Management System
Exam	Written Exam provided by the Texas State University and taken by the students as a requirement for graduation
UI	User Interface
HTML	Markup Language rendered by modern web browsers
CSS	Cascading Style Sheets used for definition web pages appearance
Javascript	Programming language interpreted by modern web browsers

1.4. References

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

1.5. Overview

The next chapter - Overall Description - of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the following chapter.

The third chapter - Specific Requirements - of this document is written primarily for the developers and describes in technical terms the details of the functionalities of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

2. Overall Description

2.1. Product Perspective

The Exam Management System is a new web-based application and does not depend or is a part of any other system. It has two different actors: the faculty member and the students. Both students and faculty members needs a compatible web browser (a list of supported browsers is provided on section 2.1.2) and internet access to access the system. The user interface is responsible for calling the backend server in order to correctly get information and handling that information to show on the screen. The backend server provides services and uses a database management system to save data.

2.1.1. User interfaces

Since the Exam Management System is a web-based application, the UI will be served in HTML, CSS and Javascript to be interpreted by a capable web-browser. Each user will see a different menu, after logging in, showing only the functions they are allowed to access. The UI should be designed in a way that a faculty member or a student should be able to use all functions after 1h of training.

2.1.2. Software interfaces

PostgreSQL v9.5.1 will be used as a Database Management System for the Exam Management System.

Compatible web-browsers: Chrome, Firefox and Safari.

2.2. Product Functions

The online exam management system can be accessed by Texas State faculty members and students. The system will assist in tracking the progress of the students and determine whether the student is ready to graduate or not.

The system will provide different functions for both faculty members and the students.This section outlines the use cases for faculty members and students pictured on Figure 1.

2.2.1. Shared use cases

Use case: Login
Actors: Faculty and Student
Brief description: The faculty or the student login to the system
Initial step-by-step description:
Before this use case can be initiated, the Faculty or the Student has already accessed the exam management system.

1. The system will show a login form with username and password
2. The faculty member or the student enters its username and password and click login
3. The system performs the login and redirects to a home page with a menu that has all functions accessible by the user according to its role.

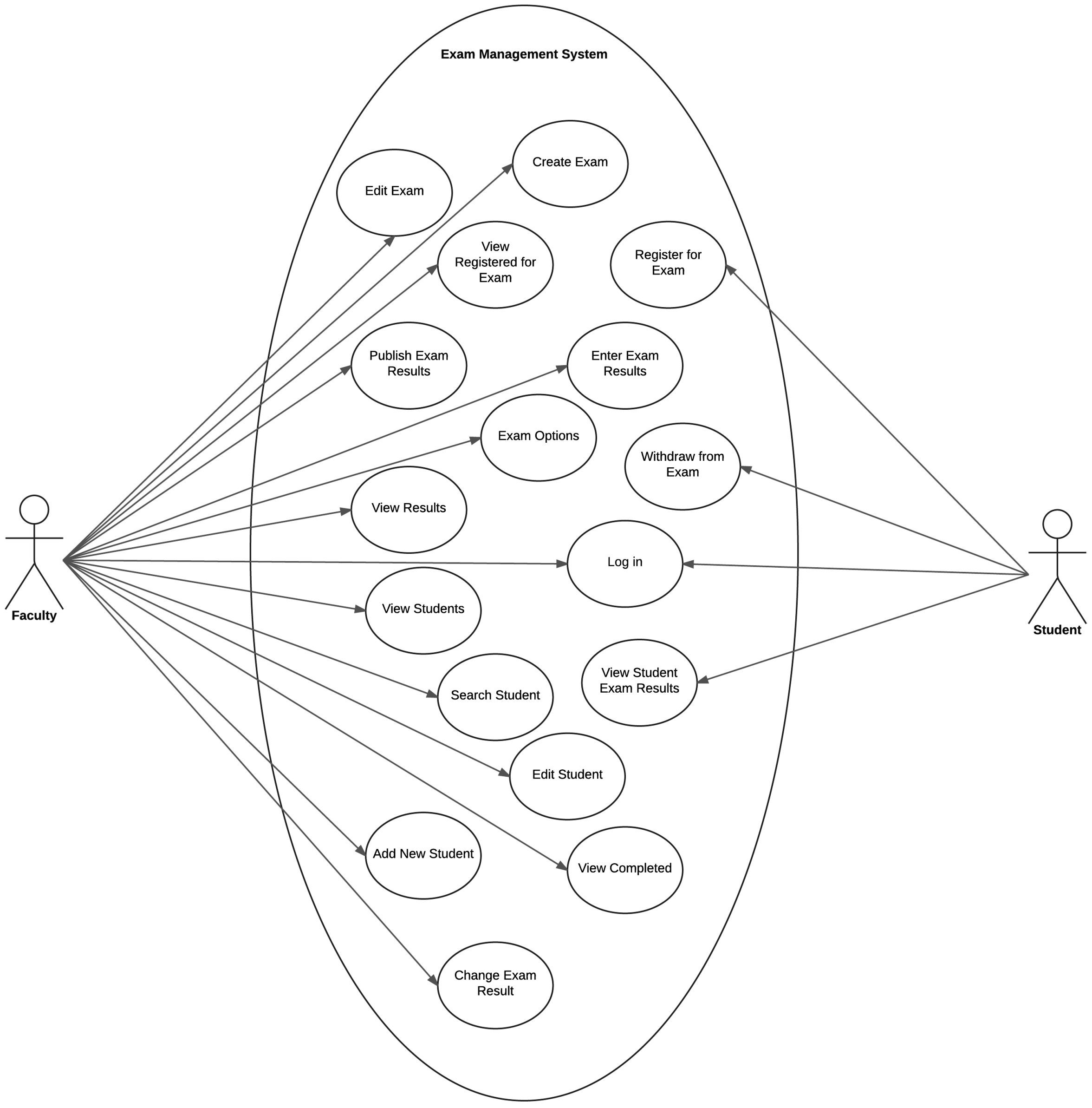

Fig. 1. Summary of use cases

2.2.2. Faculty use cases

Use case: Create Exam
Actors: Faculty
Brief description: The Faculty creates an exam
Initial step-by-step description:
Before this use case can be initiated, the Faculty has already logged in to the system.

1. The system show up a form with required data to create an exam:
   - Exam type: programming, communication or core
   - Exam date in format of yyyy-mm-dd
   - Exam start time in a 24 hr format of HH:MM:SS
   - Exam end time in a 24 hr format of HH:MM:SS

   - Semester (Fall, Spring)
   - Exam location
   - Registration deadline in a format of yyyy-mm-dd
2. The Faculty enters the required data to the form
3. The Faculty clicks on Create Exam button
4. The system creates the exam.

Use case: Exam Options
Actors: Faculty
Brief description: The Faculty lists all exams previously created.
Initial step-by-step description:
Before this use case can be initiated, the Faculty has already logged in to the system.

1. The system shows a list with all previously created exams
2. The Faculty can see the information of the exam including:
   - Exam type: programming, communication or core
   - Exam date in format of yyyy-mm-dd
   - Exam start time in a 24 hr format of HH:MM:SS
   - Exam end time in a 24 hr format of HH:MM:SS
   - Semester (Fall, Spring)
   - Exam location
   - Registration deadline in a format of yyyy-mm-dd
3. The Faculty can choose to perform an action from this list for any exam:
   - View Registered for Exam
   - Edit Exam
   - Enter Exam Results
   - View Exam Results
   - Publish Exam Results

Use case: View Registered for Exam
Actors: Faculty
Brief description: The Faculty views all students registered for a particular exam
Initial step-by-step description:
Before this use case can be initiated, the Faculty has already accessed Exam Options

1. The system shows a list with all students registered for a particular exam and the total number of students
2. The Faculty can see the information of the students including:
   - Username
   - Texas State ID
   - Last name
   - First name
   - Results (registered or not)
   - Comments

Use case: Edit Exam
Actors: Faculty
Brief description: The Faculty edits an exam
Initial step-by-step description:
Before this use case can be initiated, the Faculty has already accessed Exam Options

1. The system shows a form with the exam required data already filled up with the selected exam data:
    - Exam type: programming, communication or core
    - Exam date in format of yyyy-mm-dd
    - Exam start time in a 24 hr format of HH:MM:SS
    - Exam end time in a 24 hr format of HH:MM:SS
    - Semester (Fall, Spring)
    - Exam location
    - Registration deadline in a format of yyyy-mm-dd
2. The Faculty can edit any information
3. The Faculty clicks the Save button
4. The system saves the new data to the exam

Use case: Enter Exam Results
Actors: Faculty
Brief description: The Faculty enters results for a particular exam
Initial step-by-step description:
Before this use case can be initiated, the Faculty has already accessed Exam Options

1. The system shows a list of all students registered for a particular exam showing only the Texas State ID.
2. The Faculty can add the result for each student (noshow, passed, failed)
3. The Faculty clicks the Save button
4. The system saves the results for the exam

Use case: View Exam Results
Actors: Faculty
Brief description: The Faculty views the results for a particular exam
Initial step-by-step description:
Before this use case can be initiated, the Faculty has already accessed Exam Options

1. The system shows a list with all students registered for a particular exam showing the student information including:
    - Username
    - Texas State ID
    - Last name
    - First name
    - Result (noshow, passed, failed)
2. The system shows statistics including:
    - Total results
    - How many passed
    - How many failed
    - How many noshow

Use case: Publish Exam Results
Actors: Faculty
Brief description: The Faculty publishes the result for a particular exam
Initial step-by-step description:
Before this use case can be initiated, the Faculty has already accessed Exam Options

1. The system prompt for confirmation regarding whether the Faculty wants to publish the results or not

2. The Faculty chooses to publish the results
3. The system publishes the results by making it available to the students

Use case: View Students
Actors: Faculty
Brief description: The Faculty views all students
Initial step-by-step description:
Before this use case can be initiated, the Faculty has already logged into the system.

1. The system shows a list with all students registered on the system showingg information including:
   - Username
   - Texas State ID
   - Last name
   - First name
   - Major
   - Email
   - Result of programming exam
   - Result of communication exam
   - Result of each core exam
   - Exam history (exam name, exam date, exam result)

Use case: Add new Student
Actors: Faculty
Brief description: The Faculty adds a new student
Initial step-by-step description:
Before this use case can be initiated, the Faculty has already logged into the system.

1. The system shows a form with required data for student creation including:
   - Username
   - Texas State ID
   - Last name
   - First name
   - Major
   - Email
   - Result of programming exam
   - Result of communication exam
   - Result of each core exam
   - Local phone
   - Local address
   - Local city
   - Local state
   - Local zip
2. The Faculty fills up the form with the required data
3. The Faculty clicks on save student
4. The system creates the new student

Use case: Search Student
Actors: Faculty
Brief description: The Faculty searches for a student
Initial step-by-step description:
Before this use case can be initiated, the Faculty has already logged into the system.

1. The system shows a form with required data for student searching including:
   - Username
   - Texas State ID
   - Last name
   - First name
   - Major
   - Email
   - Result of programming exam
   - Result of communication exam
   - Result of each core exam
2. The Faculty fills up the form with the required data
3. The Faculty clicks on search
4. The system shows a list of students filtered by the search criteria

Use case: Edit Student
Actors: Faculty
Brief description: The Faculty edits student
Initial step-by-step description:
Before this use case can be initiated, the Faculty has already logged into the system.

1. The system shows a form with username and Texas State ID fields
2. The Faculty fills up either of the fields to search for the student
3. The system shows a form with the student data for editing including:
   - Last name
   - First name
   - Major
   - Email
   - Programming exam result
   - Communication exam result
   - Each core exam result
   - Local phone
   - Local address
   - Local city
   - Local state
   - Local zip
4. The Faculty edits the desired fields
5. The Faculty clicks the save button
6. The system saves the student data

Use case: Change Exam Result
Actors: Faculty
Brief description: The Faculty changes the result of exams for a particular student
Initial step-by-step description:
Before this use case can be initiated, the Faculty has already logged into the system.

1. The system shows a form with username and Texas State ID fields
2. The Faculty fills up either of the fields to search for the student
3. The system shows a form with the student exams results including:
   - Result of programming exam
   - Result of communication exam
   - Result of each core exam

4. The Faculty fills up the form with the desired data
5. The Faculty clicks on save results
6. The system saves the new results to the student record

Use case: View Completed
Actors: Faculty
Brief description: The Faculty views the students who have passed all exams
Initial step-by-step description:
Before this use case can be initiated, the Faculty has already logged into the system.

1. The system shows a list with all students that have passed programming exam, communication exam and each core exams. The results are displayed with the following information:
   - Username
   - Texas State ID
   - Last name
   - First name
   - Major
   - Email
   - Result of programming exam
   - Result of communication exam
   - Result of core exams

2.2.3. Student use cases

Use case: Register for Exam
Actors: Student
Brief description: The Student registers for an exam
Initial step-by-step description:
Before this use case can be initiated, the Student has already logged into the system.

1. The system shows all exams opened for registration
2. The Student choose the exam to register
3. The system shows a form with the required data for registration including:
   - Username
   - Texas State ID
   - Last name
   - First name
   - Major
   - Email
   - Local phone
   - Local address
   - Local city
   - Local state
   - Local zip
4. The Student clicks on the register button
5. The system registers the student on the desired exam and makes it available for the Faculty in the View Registered for Exam function

Use case: Withdraw from Exam
Actors: Student
Brief description: The Student withdraw from an exam
Initial step-by-step description:
Before this use case can be initiated, the Student has already logged into the system.

1. The system shows all exams the user is registered
2. The Student chooses the exam to withdrawn
3. The Student clicks on the Withdraw from Exam button
4. The system removed the student from the desired exam and removes the Student from the result on the View Registered for Exam function

Use case: View Student Exam Results
Actors: Student
Brief description: The Student views its results on the exams
Initial step-by-step description:
Before this use case can be initiated, the Student has already logged into the system.

1. The system shows exam results of the Student

2.3. User Classes and Characteristics

The users are faculty members and students from the Texas State University. They are supposed to have high education degrees and know how to use a web browser.

2.4. Constraints

The Student can only graduate if he/she has passed the programming, communication and core exams.

3. Specific Requirements

3.1. Functions

3.1.1. Shared Functions

3.1.1.1. Login

Function	Login
Description	The Faculty or the Student login to the system. The user enters username and password and submits the form. The system verifies the username and password. If the username and password is correct, it redirects the user to a home screen showing a list of functions the user is allowed to perform according to its role, otherwise it shows errors messages describing the error.
Inputs	Username, Password
Source	Username and password are input by the user
Outputs	Redirection to home screen, error messages
Destination	The Exam Management System API. The username and password are sent to the the API, that checks whether they are correct and login the user to the system or return error messages.
Pre-condition	The login page is displayed on the user's screen
Post-condition	The user is logged into the system and gets redirected to his home page or error messages are displayed on the user's screen.
Side-effects	None

3.1.2. Faculty Functions

3.1.2.1. Create Exam

Function	Create Exam
Description	The Faculty creates an exam. The Faculty enters exam data and clicks Create Exam button. The system validates the data and creates the exam or respond with error messages.
Inputs	Type (Programming, Communication or Core), Date (yyyy-mm-dd format), start time and end time (24hr format HH:MM:SS), Semester (Fall, Spring), Location, Registration deadline (yyyy-mm-dd format)
Source	The type, date, start time, end time, semester, location and registration deadline are input by the user.
Outputs	New exam identifier, redirection to home screen, error messages
Destination	The Exam Management System API. The exam data is sent to the API, that validates the data and save the exam to the database or return error messages.
Requires	Logged user with Faculty role.

Pre-condition	The user clicks the Create Exam function and the Create Exam page is displayed on the user's screen.
Post-condition	The exam is created and gets redirected to the home page or error messages are displayed on the user's screen.
Side-effects	None

3.1.2.2. Exam Options

Function	Exam Options
Description	The Faculty lists all previously created exams. The system shows a list of all previously created exams and a list of functions for each exam.
Inputs	None
Outputs	Information for each exam: Type (Programming, Communication or Core), Date (yyyy-mm-dd format), Start and end time (24hr format HH:MM:SS), Semester (Fall, Spring), Location, Registration deadline (yyyy-mm-dd format). List of functions: View Registered for Exam, Edit Exam, Enter Exam Results, Publish Exam Results
Destination	None
Requires	Logged user with Faculty role.
Pre-condition	The user clicks the Exam Options function and the Exam Options page is displayed on the user's screen.
Post-condition	None
Side-effects	None

3.1.2.3. View Registered for Exam

Function	View Registered for Exam
Description	The Faculty views all students registered for a particular exam. The system shows a list of students registered for a particular exam.
Inputs	Exam identifier
Source	The exam identifier from the database.
Outputs	Information for each student: Username, Texas State ID, Last name, First name, Results (registered or not), Comments
Destination	The Exam Management System API. The exam identifier is sent to the API and it returns the list of students registered.
Requires	Logged user with Faculty role.
Pre-condition	The Exam Options page is open and the user clicks on the View Registered for Exam button.
Post-condition	The View Registered for Exam page is displayed on user's screen with a list of students registered for it.
Side-effects	None

3.1.2.4. Edit Exam

Function	Edit Exam
Description	The Faculty edits an exam. The system shows the current exam information. The user edits the desired information. The user clicks the Save button. The system saves the changed data or respond with error messages.
Inputs	Exam identifier, Type (Programming, Communication or Core), Date (yyyy-mm-dd format), Start and end time (24hr format HH:MM:SS), Semester (Fall, Spring), Location, Registration Deadline (yyyy-mm-dd format)
Source	Exam type, date, start time, end time, semester, location, registration from the database or input by the user. Exam identifier from the database.
Outputs	Redirection to Exam Options, error messages
Destination	The Exam Management System API. The changed exam data is sent to the API, it validates the data and save to the database or return error messages.
Requires	Logged user with Faculty role.
Pre-condition	The Exam Options page is open and the user clicks on the Edit Exam button.
Post-condition	The exam gets updated with the new information or error messages are displayed.
Side-effects	None

3.1.2.5. Enter Exam Results

Function	Enter Exam Results
Description	The Faculty enters results for an exam. The system shows a list of students registered for the exam showing only the Texas State ID. The user enters the results for each student (noshow, passed, failed). The user clicks the Save button.
Inputs	Texas State ID, Exam identifier, exam result
Source	Exam result is input by the user. Texas State ID and exam identifier from the database.
Outputs	Redirection to Exam Options, error messages
Destination	The Exam Management System API. The results are sent to the API, that validates it and saves to the database or return error messages.
Requires	Logged user with Faculty role.
Pre-condition	The Exam Options page is open and the user clicks on the Enter Exam Results button.
Post-condition	The results are saved to the database or error messages are displayed.
Side-effects	None

3.1.2.6. View Exam Results

Function	View Exam Results
Description	The Faculty views results for an exam. The system shows a list of students registered for the exam and a summary of the exam results..
Inputs	Exam identifier
Source	Exam identifier from the database.
Outputs	Username, Texas State ID, Last name, First name, Result (noshow, passed, failed). Total Results, How many passed, How many failed, how many noshow.
Destination	The Exam Management System API. The exam identifier is sent to the API that returns a list of students with individual result and the summary.
Requires	Logged user with Faculty role.
Pre-condition	The Exam Options page is open and the user clicks on the View Exam Results button.
Post-condition	The student list with individual results and summary are displayed on the user's screen.
Side-effects	None

3.1.2.7. Publish Exam Results

Function	Publish Exam Results
Description	The Faculty publishes results for an exam. The system makes the results available to the students.
Inputs	Exam identifier
Source	Exam identifier from the database.
Outputs	None
Destination	The Exam Management System API. The exam identifier is sent to the API that makes the results available to the students.
Requires	Logged user with Faculty role.
Pre-condition	The Exam Options page is open and the user clicks on the Publish Exam Results button.
Post-condition	None
Side-effects	The results for the selected exam becomes available for the students.

3.1.2.8. View Students

Function	View Students
Description	The Faculty views all students. The system shows a list of all students in the system.
Inputs	None
Source	None

Outputs	Username, Texas State ID, Last name, First name, Major, Email, Result of programming exam, Result communication exam, Result of core exam, Exam history (exam name, exam date, exam result)
Destination	None
Requires	Logged user with Faculty role.
Pre-condition	None
Post-condition	The student list with all data is displayed on the user's screen.
Side-effects	None

3.1.2.9. Add new Student

Function	Add new Student
Description	The Faculty adds a new students. The Faculty enters student data and clicks Save student.
Inputs	Username, Texas State ID, Last name, First name, Major, Email, Result of Programming Exam, Result of Communication Exam, Result of Core Exams, Local phone, Local address, Local city, Local state, Local zip.
Source	Username, Texas State ID, Last name, First name, Major, Email, Result of Programming Exam, Result of Communication Exam, Result of Core Exams, Local phone, Local address, Local city, Local state, Local zip is input by the user.
Outputs	Student identifier.
Destination	The Exam Management System API. The student data is sent to the API that validates and saves the student to the database or return error messages.
Requires	Logged user with Faculty role.
Pre-condition	None
Post-condition	The new student is created to the database.
Side-effects	None

3.1.2.10. Search Student

Function	Search Student
Description	The Faculty searches for a student. The Faculty enters search data and clicks Search. The system shows a list of students in the database according to the search criteria.
Inputs	Username, Texas State ID, Last name, First name, Major, Email, Result of Programming Exam, Result of Communication Exam, Result of Core Exams.
Source	Username, Texas State ID, Last name, First name, Major, Email, Result of Programming Exam, Result of Communication Exam, Result of Core Exams are input by the user.

Outputs	Each student information: Username, Texas State ID, Last name, First name, Major, Email, Result of Programming Exam, Result of Communication Exam, Result of Core Exams.
Destination	The Exam Management System API. The search data is sent to the API that search for students using that search criteria and return the result.
Requires	Logged user with Faculty role.
Pre-condition	None
Post-condition	The student list in the database according to the search criteria is displayed on the user's screen.
Side-effects	None

3.1.2.11. Edit Student

Function	Edit Student
Description	The Faculty edits a student. The Faculty enters Username or Texas State ID and click search. The system loads the student information. The Faculty edits the student information except for username and texas state ID and clicks Save student.
Inputs	Username, Texas State ID, Last name, First name, Major, Email, Programming Exam Result, Communication Exam Result, Each core exam result, Local phone, Local address, Local city, Local state, Local zip
Source	Username, Texas State ID, Last name, First name, Major, Email, Programming Exam Result, Communication Exam Result, Each core exam result, Local phone, Local address, Local city, Local state, Local zip are input by the user.
Outputs	None
Destination	The Exam Management System API. The student data is sent to the API that validates and save it to the database or return error messages.
Requires	Logged user with Faculty role.
Pre-condition	None
Post-condition	The student gets updated with the new information or error messages are displayed.
Side-effects	None

3.1.2.12. Change Exam Result

Function	Change Exam Result
Description	The Faculty changes the result of exams for a student. The Faculty enters Username or Texas State ID and click search. The system loads the student information. The Faculty edits the result information clicks Save Results.
Inputs	Username, Texas State ID, Programming Exam Result, Communication Exam Result, Each core exam result.
Source	Username, Texas State ID, Programming Exam Result, Communication Exam Result, Each core exam result are input by the user.

Outputs	None
Destination	The Exam Management System API. The result data is sent to the API that validates and save it to the database or return error messages.
Requires	Logged user with Faculty role.
Pre-condition	None
Post-condition	The student results gets updated with the new information or error messages are displayed.
Side-effects	None

3.1.2.13. View Completed

Function	View Completed
Description	The Faculty views the students that have passed all exams. The system shows a list with all students that have passed all exams.
Inputs	None
Outputs	Information for each student: Username, Texas State ID, Last name, First name, Major, Email, Result of programming exam, Result of communication exam, Result of core exams
Destination	None
Requires	Logged user with Faculty role.
Pre-condition	None
Post-condition	The students that have passed all exams are displayed on the user’s screen.
Side-effects	None

3.1.3. Student Functions

3.1.3.1. Register For Exam

Function	Register for Exam
Description	The Student registers for an exam. The system shows all exams open for registration. The Student chooses the Exam to register and clicks on the Register button. The system registers the student on the desired exam and makes it available for Faculty.
Inputs	Student identifier, Exam identifier.
Source	Student identifier and exam identifier from the database
Outputs	None
Destination	The Exam Management System API. The student identifier and exam identifier is sent to the API that registers the user in the desired exam or return error messages.
Requires	Logged user with Student role and Exam open for registration.

Pre-condition	None
Post-condition	The student is registered for the exam or error messages are displayed.
Side-effects	The student registration becomes available for the Faculty.

3.1.3.2. Withdraw from Exam

Function	Withdraw from Exam
Description	The Student withdraws from an exam. The system shows all exams that the user is registered. The Student chooses the Exam to withdraw and clicks on the Withdraw button. The system withdraws the student from the desired exam and makes it available for Faculty.
Inputs	Student identifier, Exam identifier.
Source	Student identifier and exam identifier from the database
Outputs	None
Destination	The Exam Management System API. The student identifier and exam identifier is sent to the API that withdraws the user from the desired exam or return error messages.
Requires	Logged user with Student role and Student already registered for the Exam.
Pre-condition	None
Post-condition	The student is withdrawn from the exam or error messages are displayed.
Side-effects	The student withdrawn becomes available for the Faculty.

3.1.3.3. View Student Exam Results

Function	View Student Exam Results
Description	The Student views his results on all exams. The system shows exam results for the student.
Inputs	Student identifier.
Source	Student identifier from the database
Outputs	Result of programming exam, Result of communication exam, Result of core exams
Destination	None
Requires	Logged user with Student role.
Pre-condition	None
Post-condition	The exam results for the logged student are displayed on the user’s screen
Side-effects	None