Software Requirements Specification

for

User Registration System

Version 1.0 approved

Prepared by Adam E. Svatek

University of Arizona Global Campus

April 15, 2024

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Adam E. Svatek | 4/15/2024 | Initial version of SRS. | 1.0 |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this software requirements system (SRS) is to identify and describe the requirements of the software for the User Registration System. This SRS document if for release 1.0 of the User Registration System. The scope of the User Registration System is to allow users to create an account and register for college courses.

## Document Conventions

This document consists of five sections that include Introduction, Overall Description, External Interface Requirements, System Features, Other Nonfunctional Requirements, and Other Requirements. The titles of the sections and subsections will be in bold font. The titles of the sections will be in a larger font.

## Intended Audience and Reading Suggestions

The intended audience of this document is the writers, software designers, software developers, software testers, web page architects, project managers, subject matter experts, college office staff, and other parties included in this project. It is suggested the intended audience reads this SRS in its entirety beginning with the table of contents and ending with the appendices.

## Product Scope

The scope of this product is to provide the users with a user registration system. They will be able to create a user account. The users will be able to register for courses, cancel enrollments, and place their name on a waiting list for classes that are full. The college using the User Registration System will provide students and potential students with a means of registering for classes online.

This section describes WHY we’re creating this software.

## References

None.

# Overall Description

## Product Perspective

The User Registration System will be a new, self-contained product.

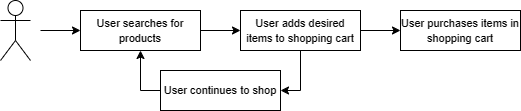
## Product Functions

The major functions that this program will perform include providing the user the ability to create a unique username, create a profile, register for classes, cancel enrollment, and add their name to a waiting list of classes that are full. Table 2-1 shows the user interface when they search for classes, enroll, and then log out.

* Create an account.
  + User will be able to create a unique account.
  + No two user names can be the same.
  + User creates a password of 8 characters or more.
  + User will add their name, phone number, and email address.
* Users will be able to log out and log back in after creating an account.
  + User must enter the correct password for their username to login.
* Register for classes.
  + Users can select one of three semesters (spring, summer, and fall classes) to register for classes.
  + Users can register for classes in the semester they select.
  + If a class is book, users can put their name on a waiting list to be notified if someone cancels their enrollment.
* Cancel enrollment.
  + Users can cancel their enrollments.

**Diagram

Description automatically generated**



**Figure 2-1:**

## User Classes and Characteristics

The three classes of users for the User Registration System will be student, advisor, and administrator. Students will have the characteristics of being able to create their own accounts, add and edit their information, search for classes, register for classes, and cancel enrollments. Advisors will be able to register students for classes, cancel enrollments, print out rosters for classes, and access other student information. Administrators will have the ability to add and remove all types of users, add or edit their information, and access all user information except passwords.

## Operating Environment

The User Registration System will run on currently existing Apple and Microsoft operating systems on desktop computers and laptops. The User Registration System will also run on current operating systems for iPhone, Android, and Google mobile devices and tablets.

## Design and Implementation Constraints

The User Registration System will be created adhering to the policies of the university. Functions of the software will be performed as scheduled by the development team.

## User Documentation

The User Registration System will have a built-in tutorial to help students create their own student account. It will also have a tutorial describing how to register for classes, view their enrollments, and cancel enrollments.

## Assumptions and Dependencies

It is assumed that personal computers and laptops will run on the current Microsoft operating system, Apple computers and laptops will run on the current Apple iOS, iPhones and iPads will run on the current Apple iOS, and other mobile devices will run on either the current Android or Google operating systems.

# External Interface Requirements

## User Interfaces

The screens for the User Registration System will have a solid background with a simple header consisting of defined drop-down menus. The font on all screens will be Arial font. Buttons will have solid boarders and use Arial font. Keyboard shortcuts will not be used in the initial version. Error messages will be displayed in simple popups. The User Registration will be 508 compliant.

## Hardware Interfaces

The User Registration System will communicate with the monitors, keyboards, and mouses for computers. It will communicate with the touchscreen displays of mobile devices and tablets. It will communicate with a keyboard or mouse if connected to a mobile device or tablet. The User Registration System will communicate with printers.

## Software Interfaces

The User Registration System will connect to databases including a database of users, database with details of the university’s courses, student enrollments, schedule of courses, and course member rosters. A student can access a list of courses which will interface with a database of the course information. As the student decides to sign up for classes, the system will access a database with the schedule for the upcoming semesters. They can enroll or cancel enrollments, which the software would interact with a database of student enrollments. As the student enrolls for a class, their name will be added to a course roster.

The software will interface whichever operating system that the user’s device runs on, i.e. iOS, Microsoft OS, and Google OS.

## Communications Interfaces

The User Registration System will require e-mail, web browsers, and internet network server communication. Students can choose to receive e-mail notification of their course enrollments and reminders of course start dates that are approaching. The software on the user’s device will communicate with the system with private and public key encryption.

# System Use Cases

Diagram

Description automatically generated

## Create User Login

1. **Objective** – User creates their username, password, and user information.
2. **Priority** – High
3. **Source** – Student. Joe Johnson (End user)
4. **Actors** - Students
5. **Flow of Events** 
   1. **Basic Flow** – User enters a username and password to create login.
   2. **Alternative Flow(s)** - None
   3. **Exception Flow(s)** – None
6. **Includes** – Potential user name is checked against list of users.
7. **Preconditions** – If username is not the same as any existing username, the student is able to create their account.
8. **Post conditions** – The user creates their account. They are able to search for classes, register for classes, or log out.
9. **Notes/Issues** – None.

## Withdraw money from ATM (U2)

1. **Objective** – The customer is withdrawing money from the ATM and the system will debit the customer’s account.
2. **Priority** – High
3. **Source** – Carl Gnome (marketing)
4. **Actors** – Customer, central bank computer
5. **Flow of Events** 
   1. **Basic Flow**
      1. Customer chooses the checking option on the ATM
      2. Customer chooses the amount of money needed
      3. Customer confirms the choice
      4. System validates the amount
      5. System asks central bank computer to debit the customer’s account
      6. System issues money to the user
   2. **Alternative Flow 1** – At step 5.1.4 the amount is not a multiple of $20
      1. An error message is displayed telling the customer they must use multiple of $20.
      2. Return to step 5.1.2
   3. **Alternative Flow 2** – At any step the user presses “cancel”
      1. System returns to the main menu
   4. **Alternative Flow 3** - At step 5.1.5 bank computer returns a failed status, “insufficient funds”
      1. An error message is shown to the user
      2. Return to step 5.1.2
   5. **Exception Flow 1** –
      1. Database is locked due to backup in progress. System executes use case U5
6. **Includes**
   1. U5 – Exception occurs
7. **Preconditions** – User is logged in
8. **Post conditions** – Money has been returned to the user and their account balance has been updated.
9. **Notes/Issues** - None

## Deposit money into ATM (U3)

….

# Other Nonfunctional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

You may not have any.

# System Requirements Chart

*< Include a* ***table*** *in this section with the following columns:*

***ID*** *– Unique requirement ID*

***Priority*** *– Priority of this requirement*

***Type*** *– Functional(F) or Non-functional(NF)*

***Source*** *– Who is most interested in this requirement (John Smith – Customer). For this project you can make it up, in reality you’ll want to capture this as you capture the requirements.*

***Contained in Use Case(s****) – Which use cases reference this requirement or which use cases when executed will perform this requirement. There may be a few functional requirements without a use-case and the non-functional requirements generally will NOT be part of a use-case (so put N/A).*

***Description*** *– The description of the requirement. “The system shall …. “*

*>*

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

**Don’t do any of these for CS421 SRS. You will create these models during the high level design deliverable.**

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>

List here any open questions or things you know still need to be done to the SRS, but haven’t been addressed yet. (It’s okay to have things like that, especially in this CS421 project because we don’t have time to do everything.)