Software Requirements Specification

for

4K Learning Management Software

Prepared by: Regina Andes, Austin Pritchett,

Hiep Nguyen, Andres Olvera

Created on April 6, 2020

Table of Contents

Table of Contents ii

1. Introduction 1

1.1 Purpose 1

1.2 Team Organization 1

1.3 Software Development Lifecycle 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Features and Functions 2

2.3 User Classes and Characteristics 3

2.4 Operating Environment 4

2.5 Design and Implementation Constraints 4

2.6 Assumptions and Dependencies 4

3. System Features 4

3.1 Student User 4

3.2 Faculty User 5

3.3 Administrator 6

4. External Interface Requirements 7

4.1 User Interfaces 7

4.2 Software Interface 8

4.3 Other Requirements 9

Appendix A: Glossary 9

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is for us to show our Learning Management Software (LMS) functionalities and demonstrate our skills as software engineers to create an operating LMS using python. The SRS document should allow our client to understand the LMS software with its technical and non-technical aspect.

1.2 Team Organization

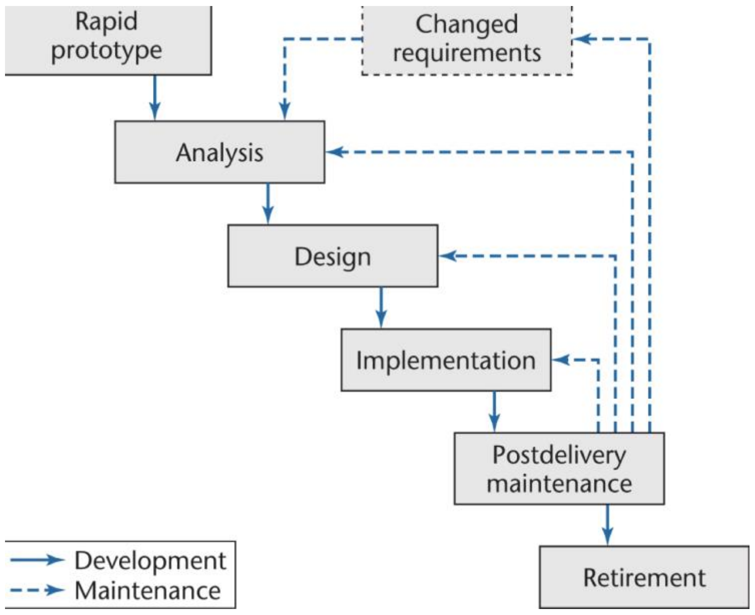
The approach we chose to tackle this project was the Democratic team organization. It suited us well due to our team consisting of undergraduate students. We have skill sets that are about the similar level. Our tasks were divided into two and decisions were made as a collective team.

Main Programmers: Hiep Nguyen and Austin Pritchett

SRS Documentation: Regina Andes and Andres Olvera

1.3 Software Development Lifecycle

We based our Learning Management Software's development life cycle to be a Rapid Prototyping Model. We wanted to get a working Graphical User Interface and log in feature as quickly as we can be interactive and work on any issues that arise within our prototype.



2. Overall Description

2.1 Product Perspective

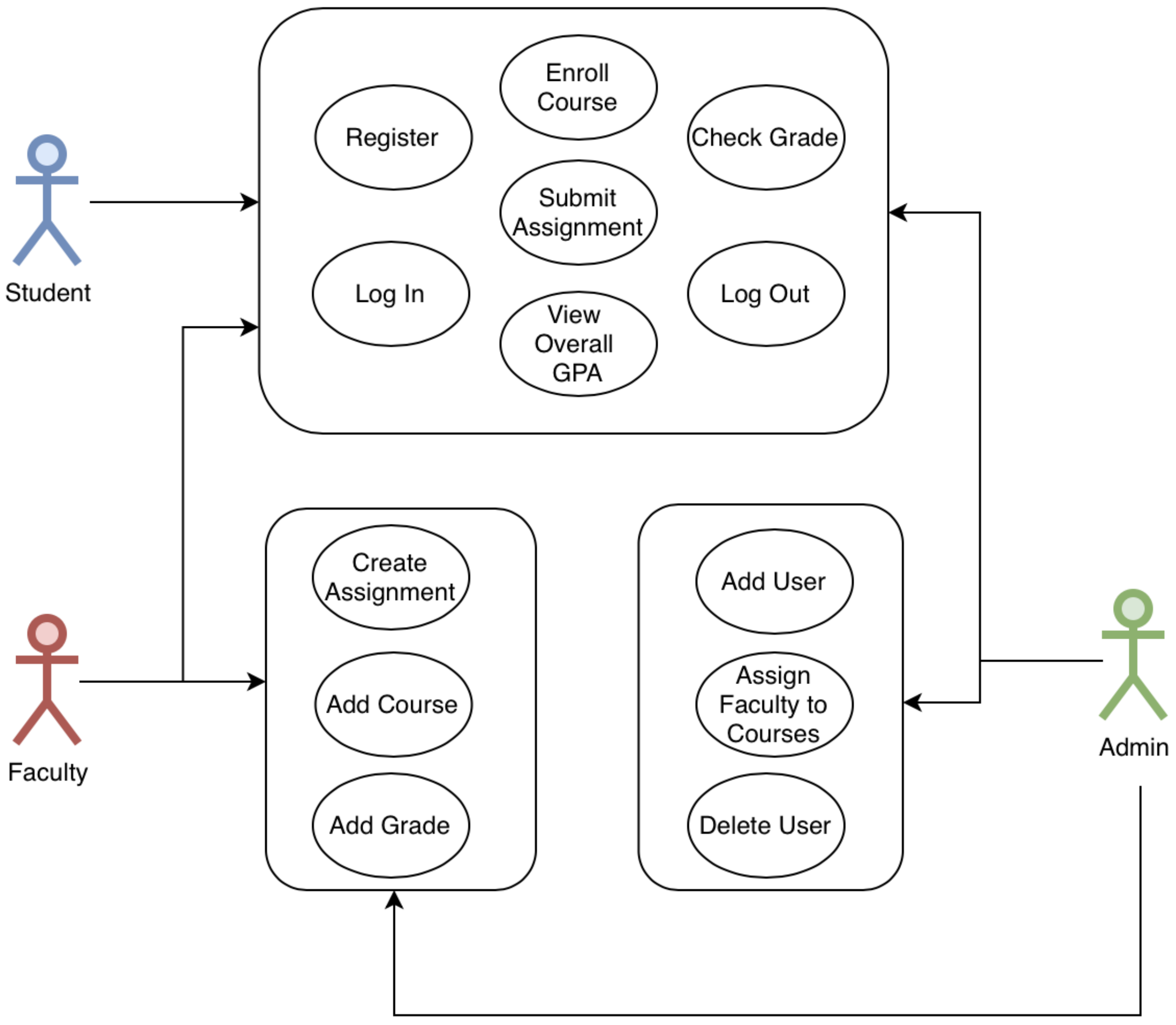
The goal of the Learning Management Software is to provide alternative selection for institutions, educators and their students. A software is needed to manage faculty and student information, store and calculate grades, upload assignments, provide resources and track student progress. This software is modeled after the University of Houston-Downtown preferred learning module: Blackboard, which our team used as a foundation of starting this LMS.

2.2 Product Features and Functions

The Learning Management Software product features and functions is listed below:

- **Registration:**
  1) User can create an account by clicking the Register option in the front page
  2) User enters information: username and password
- **User Log In:**
  1) User can log in by clicking Login option in the front page
  2) User enters information: username and password
- **User Log Out:**
  1) User gets automatically logged off after exiting program
- **Add Grade:**
  1) Faculty user can input student grades in the system
  2) Student user won't be allowed to edit this feature
- **Add Course:**
  1) Faculty user and Admin can input courses for students in the system.
- **View Overall GPA / Check Grade:**
  1) All users can view overall GPA and check student grades.

2.3 User Classes and Characteristics



The system will have three types of users: student, faculty and administrator.

Student users will only have certain limited privileges. Currently enrolled students can register, log in, log out, enroll courses, submit assignment, check grade and view overall GPA.

Faculty users also have limited privileges. Faculty users will consist of lecturers and professors. Those users can do everything student users can. Additional use cases they can create assignments, add grades and add courses.

Administrators will have full control of the system privileges. They are allowed to do both student and faculty privileges and have extra privileges such as assigning faculty to courses, add students or faculty users and delete users.

2.4 Operating Environment

Our source code is planned on being written in Python, it allows the software to be used on a variety of systems. Data will be stored on a centrally located server. We used all kinds of operating systems like Linux, Windows, MacOs for the project. Compatibility was essential.

2.5 Design and Implementation Constraints

Being that our team was more knowledgeable in languages such as C++ and Python. Our constraints were limited to using a Python IDE to compile our working code together. With limited knowledge and time constraints on web-based programming, we couldn't implement this prototype into a website.

The team found it challenging to mend the code as a whole, so we had to compile each code separately.

2.6 Assumptions and Dependencies

In order to work on this LMS project as a team, we relied on using the Github platform to submit content and Google Drive/Google Docs to work on the SRS documentation. These were reliable websites that kept our Democratic team in accountable. Our team also used GroupMe and Zoom to communicate with one another while we couldn't physically get together

3. System Features

3.1 Student User

3.1.1 Description and Priority

Low Priority. Allows the student user to view the classes they are currently enrolled in, as well as their past classes by semester. The user has no ability to change anything about this list, they can only read the information.

3.1.2 Stimulus/Response Sequences

After logging into the LMS, students can select a link to view their current courses. The course names, instructors, grades, and overall GPA will be displayed on this page.

3.1.3 Functional Requirements

REQ-1: GUI Functionality

REQ-2: Valid user login information

ERR-1: If login invalid, either register or enter correct information

REQ-3: Must be enrolled in at least one class

3.2 Faculty User

3.2.1 Description and Priority

Medium Priority. Allows the professor to view their enrolled students by course reference number and create assignments for each class. After each assignment's deadline, the professors will enter grades for each student across courses.

3.2.2 Stimulus/Response Sequences

After logging into the LMS as faculty, professors will be shown a list of the courses they are teaching. For each course, the professors can view and edit the grades of every enrolled student as well as create assignments, quizzes, and tests.

3.2.3 Functional Requirements

REQ-1: GUI Functionality

REQ-2: Valid educator login information

ERR-1: If login invalid, either register as a professor or enter correct password

REQ-3: Must teach at least one class in the given semester. Otherwise they can only view information for past semesters.

3.3 Administrator

3.3.1 Description and Priority

High Priority. Administrators have full access in terms of who can register an account to the LMS, the number of students allowed to enroll for a course, as well as which professors will be teaching the course.

3.3.2 Stimulus/Response Sequences

After logging into the LMS, administrators can view every detail for any registered student or educator on the system. Admins can delete accounts, in the case of students leaving the university, as well as remove students from a course, if no attendance or expulsion. Prior to the start of every semester, administrators also create each course and assign professors.

3.3.3 Functional Requirements

REQ-1: GUI Functionality

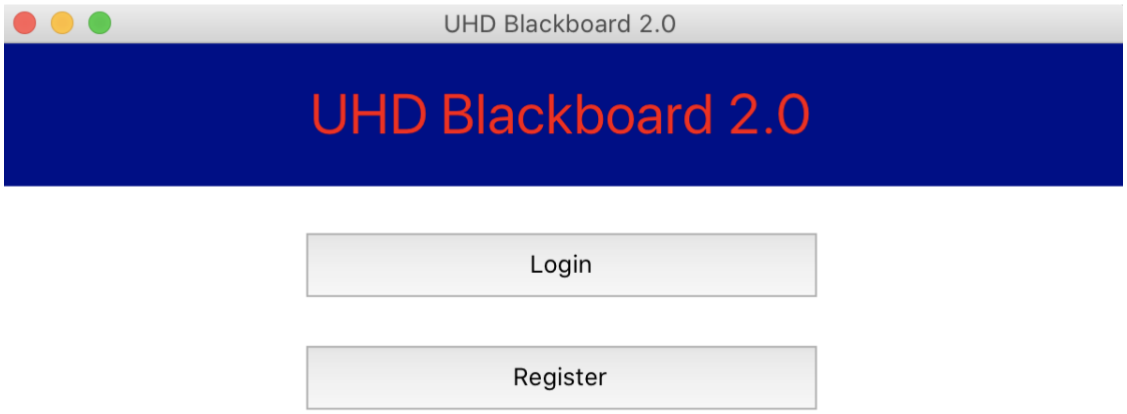
REQ-2: Valid admin login information

ERR-1: If login invalid, enter correct information or contact school

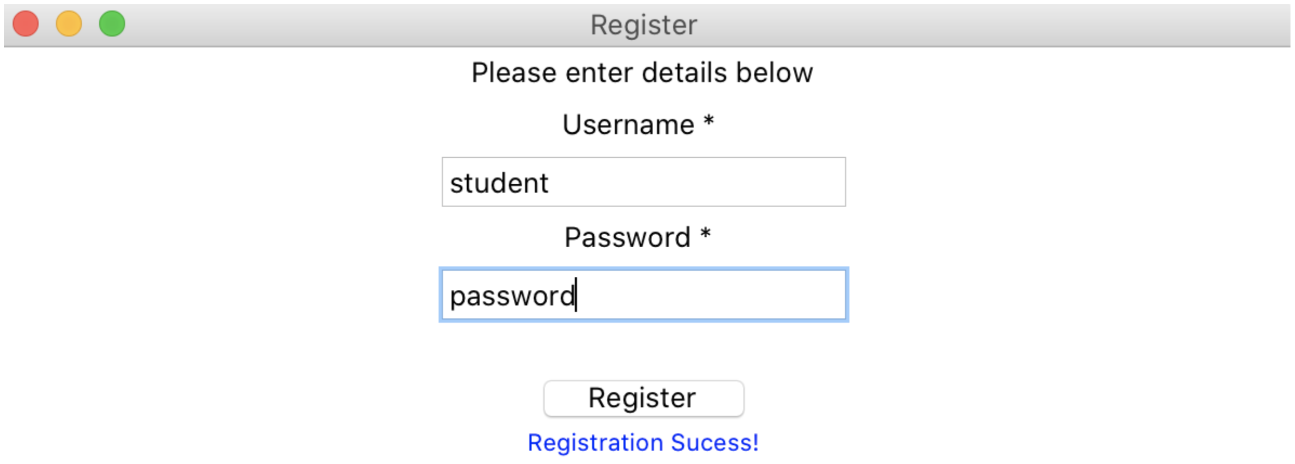
4. External Interface Requirements

4.1 User Interfaces

GUI for the UHD Blackboard 2.0:



Registration: user creates a username and password



Please enter details below

Username *

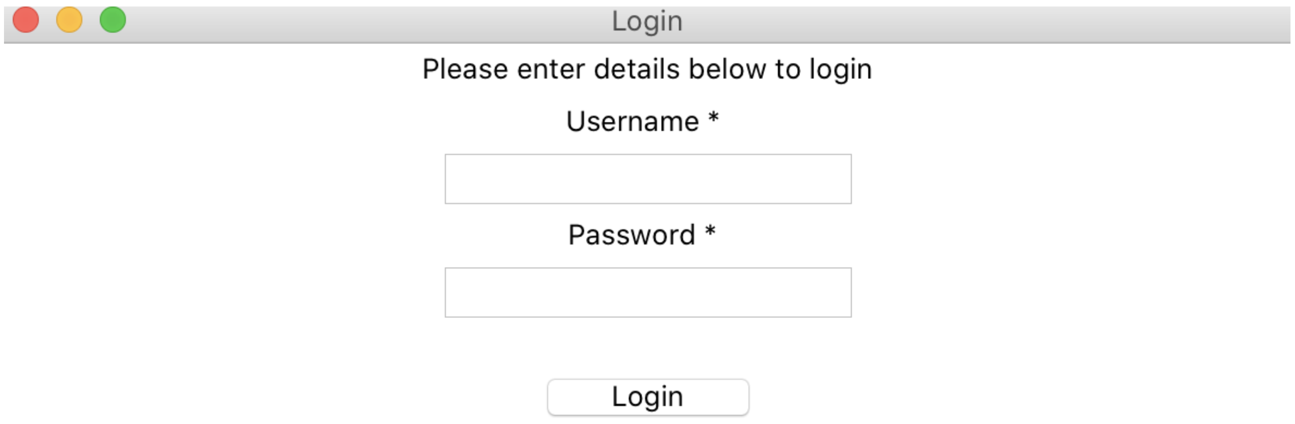
student

Password *

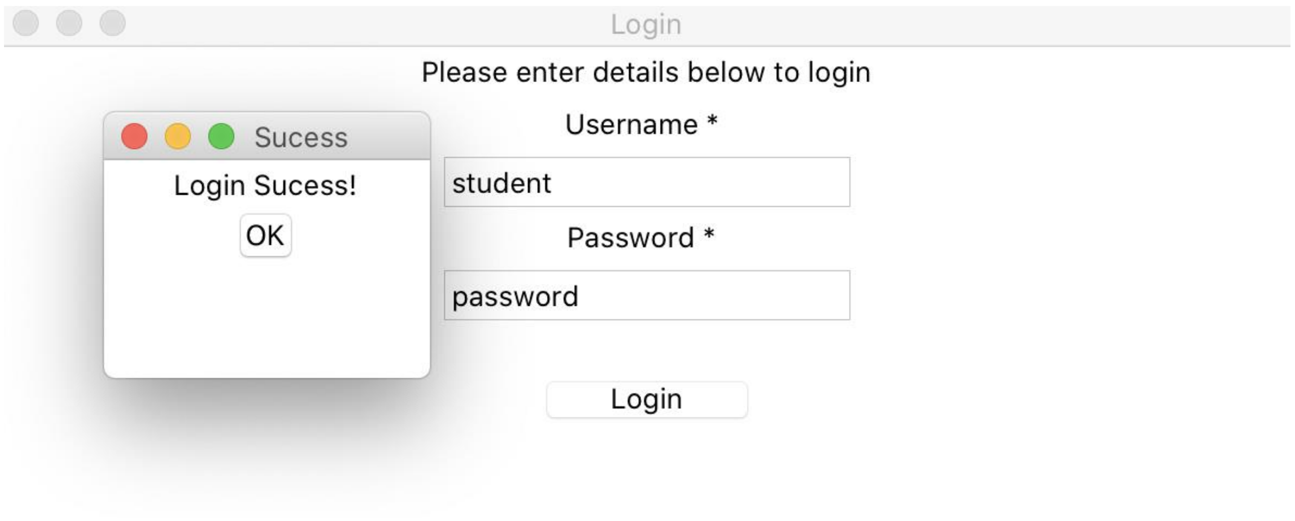
password

Registration Sucess!

System prompting user to enter login information below:



After logging in - system creates a successful login notification:



Username: student

Password: password

Login Sucess!

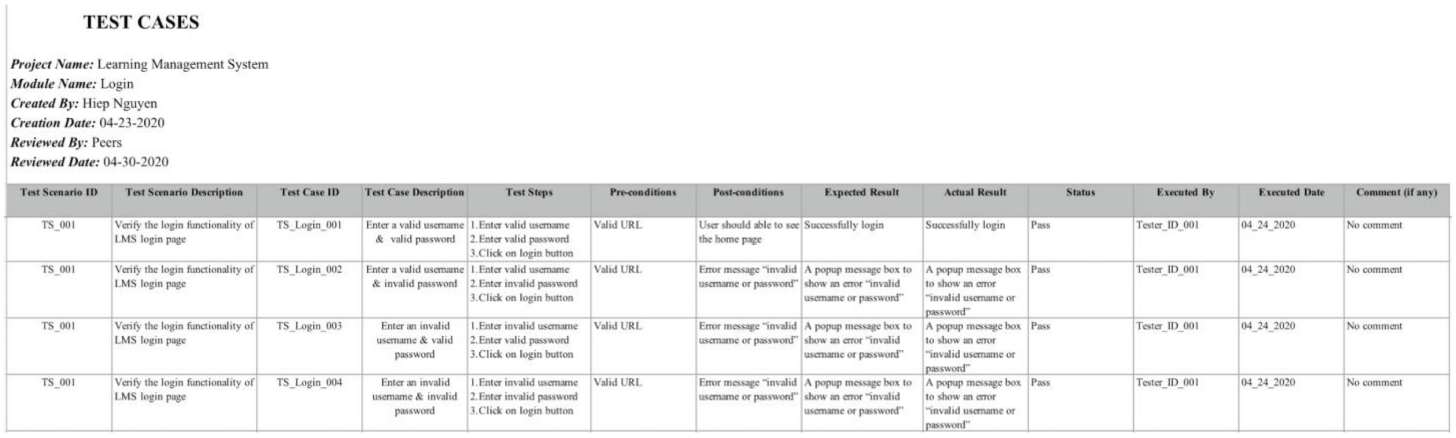
Ideally, after logging in with a type of account (student, educator, or administrator), users will be presented with the proper information discussed in the system features list.

4.2 Software Interface

Tkinter python package was used to program user interface above this document. This graphical user interface (GUI) toolkit allowed our software developers to create a separate window for the Learning Management System page.

4.3 Other Requirements

One of our software developers performed test cases to validate the login functionality of the LMS. The team reviewed the test cases.



TEST CASES

Project Name: Learning Management System
Module Name: Login
Created By: Hiep Nguyen
Creation Date: 04-23-2020
Reviewed By: Peers
Reviewed Date: 04-30-2020

Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Test Steps	Pre-conditions	Post-conditions	Expected Result	Actual Result	Status	Executed By	Executed Date	Comment (if any)
TS_001	Verify the login functionality of LMS login page	TS_Login_001	Enter a valid username & valid password	1.Enter valid username 2.Enter valid password 3.Click on login button	Valid URL	User should able to see the home page	Successfully login	Successfully login	Pass	Tester_ID_001	04_24_2020	No comment
TS_001	Verify the login functionality of LMS login page	TS_Login_002	Enter a valid username & invalid password	1.Enter valid username 2.Enter invalid password 3.Click on login button	Valid URL	Error message "invalid username or password"	A popup message box to show an error "invalid username or password"	A popup message box to show an error "invalid username or password"	Pass	Tester_ID_001	04_24_2020	No comment
TS_001	Verify the login functionality of LMS login page	TS_Login_003	Enter an invalid username & valid password	1.Enter invalid username 2.Enter valid password 3.Click on login button	Valid URL	Error message "invalid username or password"	A popup message box to show an error "invalid username or password"	A popup message box to show an error "invalid username or password"	Pass	Tester_ID_001	04_24_2020	No comment
TS_001	Verify the login functionality of LMS login page	TS_Login_004	Enter an invalid username & invalid password	1.Enter invalid username 2.Enter invalid password 3.Click on login button	Valid URL	Error message "invalid username or password"	A popup message box to show an error "invalid username or password"	A popup message box to show an error "invalid username or password"	Pass	Tester_ID_001	04_24_2020	No comment

Appendix A: Glossary

- **Github:**
  Hosting for software development version control using Git.

- **Graphical User Interface (GUI):**
  System of interactive visual components for computer software.

- **Learning Management Software (LMS):**
  Software application for the administration, documentation, tracking, reporting, and delivery of educational courses, training programs, or learning and development programs.

- **Python:**
  Interpreted, high-level, general-purpose programming language.

- **Software Requirement Specification (SRS):**
  Documentation of a software system to be developed with its functional and non-functional requirements.

- **Test Cases:**
  set of actions executed to verify a particular feature or functionality of the software.

- **Tkinter:**
  Python binding Tk GUI toolkit.