**Software Requirements Specifications**

**Student Help Desk**

Version 1.0 Approved

Prepared By

Laraib Ishtiaq FA17-BSE-096

Umair Ansar FA17-BSE-069

Abdul Rafay FA17-BSE-092

Zunaira Ali FA17-BSE-076

COMSATS University Islamabad, Wah Campus

December 09, 2019

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Revision | Description | Author |
| December 20, 2019 | 1.0 | Initial | Laraib Ishtiaq  Zunaira Ali  Abdul Rafay  Umair Ansar |

Table of Contents

[**Revision History 2**](#_Toc27413558)

[**1 Introduction 4**](#_Toc27413559)

[1.1 Introduction 4](#_Toc27413560)

[1.2 Purpose 4](#_Toc27413561)

[1.3 Scope 4](#_Toc27413562)

[1.4 Definitions, Acronyms and abbreviations 4](#_Toc27413563)

[1.5 References 4](#_Toc27413564)

[1.6 Assumptions and dependencies 4](#_Toc27413565)

[**2 Use Case Model Survey 4**](#_Toc27413566)

[**3 Actor Survey 8**](#_Toc27413568)

[**4 Requirements 9**](#_Toc27413569)

[4.1 Functional Requirements 9](#_Toc27413570)

[4.2 Non-Functional Requirements 9](#_Toc27413571)

[**5 User Documentation and Help System Requirements 9**](#_Toc27413572)

[**6 Design Constraints 9**](#_Toc27413573)

[6.1 Implementation language 9](#_Toc27413574)

[6.2 Tool used 9](#_Toc27413575)

[**7 Purchased Components 10**](#_Toc27413576)

[**8 Software Interfaces 10**](#_Toc27413577)

[**9 Licensing and Security Requirements 10**](#_Toc27413578)

[**10 Legal, Copyright and Other Notices 10**](#_Toc27413579)

[**11 Applicable Standards 10**](#_Toc27413580)

[**12 Glossary 10**](#_Toc27413581)

# 1 Introduction

## 1.1 Introduction

The main purpose of the system Student Help Desk is to automate the management of student’s daily needs efficiently. To build a system that can guide freshers and any other student for their upcoming courses and to view many needed documents i.e. assignments, refences books in easy way and short time.

## 1.2 Purpose

It provides a single point of contact for students to gain assistance by seniors and fellows, get answers to study related questions, and solve known problems. This system will manage the student queries and offer them with the best solutions.

## 1.3 Scope

The scope of this system extends over Comsats university, Wah Campus and its key stakeholder students so that they can have easy access to all the course guidelines and books related to their curriculum uploaded by any fellow students without surfing internet.

## 1.4 Definitions, Acronyms and abbreviations

PHP Hypertext Pre-processor

HTML Hypertext Mark-up Language

CSS Cascading Style Sheets

## 1.5 References

[1] SANTA ANA COLLEGE: Student Help Desk - Technical Support, url:

“<https://www.sac.edu/StudentServices/s4s-program/Pages/student-help-desk.aspx>”

[2] New Man University Help Desk, url: “<https://newmanu.edu/campus-life/student-services/i-t-help-desk>”

[3] Creatly Online Software for prototype and diagrams, url:” <https://creately.com/>”

## 1.6 Assumptions and dependencies

It is assumed that the student is familiar with the fundamentals of computer.

Students have an active internet connection.

Student has an operating system which supports internet browsing.

Removeable media should be scanned for viruses before use.

Only authorized students will be allowed access to system.

User have the enough knowledge of university system and all its collaborations with the students.

Product will be always used in those systems that have enough resources as required by this system

# 2 Use Case Model Survey

Table 2.1: Partial Specification of the Student’s help desk “Register Student” use case.

|  |  |
| --- | --- |
| **ID:** | UC\_1 Register Student |
| **Created By:** | Laraib Date created 9/12/19 |
| **Primary Actor:** | New student |
| **Description:** | The new student in order to get access to the system, to upload and view documents will request for registration by entering its name or registration number. The system either registers the new student or declines the request if student does not have a valid registration number from domain. |
| **Trigger:** | New student indicates that he wants to register himself for getting access to system. |
| **Preconditions:** | PRE-1. New Student’s record has been registered in university’s SE department.  PRE-2. Students database is online. |
| **Postconditions:** | POST-1. Student’s request is stored in database.  POST-2. Request was sent to the admin for acceptance or denial. |
| **Normal Flows:** | **1.0 Student’s registration in database online**  1. New Student enters his name and registration number and sends a request for registration.  2. System states the new student as registered student, if request is accepted.  3. Information about student is stored into database and he is then redirected to login page. |
| **Alternative Flows:** | **1.1 Student login into system**  1. New Student enters his name and registration number and sends a request for registration.  2. System states the new student as registered student, if request is accepted.  3. Information about student is stored into database and he is then redirected to login page. |
| **Exceptions:** | N-A |
| **Priority** | High |
| **Frequency of Use:** | Approximately 150 times per week by a student. |
| **Business Rules:** | N-A |
| **Other Information:** | The system must be able to store record of students in an orderly manner. |
| **Assumptions:** | Database for student’s record is assumed to be online. |

Table 2.2: Partial Specification of the Student’s help desk “Update records” use case.

|  |  |
| --- | --- |
| **ID:** | UC\_2 Update Records |
| **Created By:** | Umair Date created 9/12/19 |
| **Primary Actor:** | Registered Student |
| **Description:** | The registered student requests to update his profile information or any document which he uploaded. |
| **Trigger:** | Registered student indicates that he wants to update the information he entered before or any document which he uploaded before. |
| **Preconditions:** | PRE-1. Student’s identity has been authenticated.  PRE-2. Student is authorized to request an update.  PRE-3. Student is logged into the system.  PRE-4. Student’s record database is online. |
| **Postconditions:** | POST-1. Updation request is stored in database.  POST-2. Request was sent to the admin for further processing. |
| **Normal Flows:** | **2.0 Updating student’s record.**  1. Registered student specifies the desired update input.  2. System lists all the information present in database related to that student.  3. System gives registered student the option to view and update the information.  4. System stores the request and updates the database. |
| **Alternative Flows:** | **2.0 Updating student’s record manually.**  1. Registered student specifies the desired update information to admin.  2. Admin inputs the record in database.  3. Admin tells student about successful Updation. |
| **Exceptions:** | N-A |
| **Priority** | Medium |
| **Frequency of Use:** | Approximately 5 times per week by a student. |
| **Business Rules:** | N-A |
| **Other Information:** | The system must be able to update record of students without any delay. |
| **Assumptions:** | Database for student’s record is assumed to be online. |

Table 2.3: Partial Specification of the Student’s help desk “View Uploaded documents” use case.

|  |  |
| --- | --- |
| **ID:** | UC\_3 View uploaded documents |
| **Created By:** | Rafay Date created 9/12/19 |
| **Primary Actor:** | Registered Student |
| **Description:** | The registered student can view any of the file, past papers, assignments and quizzes solutions which are uploaded by him or any other user. |
| **Trigger:** | Registered student indicates that he wants to view the uploaded documents. |
| **Preconditions:** | PRE-1. Student’s identity has been authenticated.  PRE-2. Student is authorized to view the documents.  PRE-3. Student is logged into the system. |
| **Postconditions:** | POST-1. There should be one or more documents present in system to view.  POST-2. Documents will appear on screen. |
| **Normal Flows:** | **2.0 Viewing uploaded documents.**  1. Registered student specifies the desired file or category of file to view.  2. System lists all the desired documents present in database.  3. Registered Student selects a specific document.  4. System gives registered student the option to view and download the document.  5. Registered student then views or downloads it according to need. |
| **Alternative Flows:** | N-A |
| **Exceptions:** | N-A |
| **Priority** | High |
| **Frequency of Use:** | Approximately 200 times per week by a student. |
| **Business Rules:** | N-A |
| **Other Information:** | The system must be able to store all documents and in all file types. |
| **Assumptions:** | Desired file is assumed to be uploaded before. |

Table 2.4: Partial Specification of the Student’s help desk “Delete any record” use case.

|  |  |
| --- | --- |
| **ID:** | UC\_4 Delete any record |
| **Created By:** | Zunaira Date created 9/12/19 |
| **Primary Actor:** | Registered Student |
| **Description:** | The registered student can delete any of the file, past papers, assignments and quizzes solutions which are uploaded by him. |
| **Trigger:** | Registered student indicates that he wants to delete the uploaded documents. |
| **Preconditions:** | PRE-1. Student’s identity has been authenticated.  PRE-2. Student is authorized to delete the documents  PRE-3. Student has already uploaded a file  PRE-4. Student is logged into the system. |
| **Postconditions:** | POST-1. There should be one or more documents uploaded by him to delete.  POST-2. Documents will appear on screen. |
| **Normal Flows:** | **4.0 Deleting uploaded documents.**  1. Registered student specifies the desired file to delete.  2. System lists all the desired documents present in database.  3. Registered student selects a specific document.  4. System gives registered student the option to view and delete the document.  5. Registered student then deletes it. |
| **Alternative Flows:** | **2.0 Deleting student’s record manually.**  1. Registered student specifies the record to be deleted to admin.  2. Admin deletes the record from database.  3. Admin tells student about successful deletion. |
| **Exceptions:** | N-A |
| **Priority** | Low |
| **Frequency of Use:** | Approximately 10 times per week by a student. |
| **Business Rules:** | N-A |
| **Other Information:** | The system must be able to store all documents and in all file types. |
| **Assumptions:** | Desired file is assumed to be uploaded before. |

Table 2.5: Partial Specification of the Student’s help desk “View all records” use case.

View All Records

|  |  |  |
| --- | --- | --- |
| Description | List of Actors | Diagram |
| The admin can view all the registered student’s records. He is given the access to these records and he can delete a record if needed. | Admin |  |

Table 2.6: Partial Specification od the Student’s help desk “upload assignments, quizzes solutions” use case.

Upload Assignments, quizzes solutions

|  |  |  |
| --- | --- | --- |
| Description | List of Actors | Diagram |
| The Registered Students can upload assignments, quizzes solutions. He will login to the system, then insert the specific file from his local directory into the system and every other registered student will then be able to view or download that file. | Registered Students |  |

Table 2.7: Partial Specification od the Student’s help desk “upload past papers, books, files” use case.

Upload Past papers, books and files

|  |  |  |
| --- | --- | --- |
| Description | List of Actors | Diagram |
| The Registered Students can upload past papers, books or files. He will login to the system, then insert the specific file from his local directory into the system and every other registered student will then be able to view or download that file. | Registered Students |  |

# 3 Actor Survey

Table 3.1 List of Actors

|  |  |
| --- | --- |
| Actor Name | Description |
| New Student | 1- New student can register to the system to avail all the facilities provided by the system. |
| Admin | 1- Admin can register new students.  2- Admin can view student’s records.  3- Admin can view all documents.  4- Admin can maintain student records.  5- Admin can update student’s records.  6- Admin can delete any students record. |
| Registered Student | 1- Registered student can upload and view past papers.  2- Registered student can upload and view assignments.  3- Registered student can upload and view quizzes.  4- Registered student can upload and view notes.  5- Registered student can update his files.  6- Registered student can delete his file. |

# 4 Requirements

## 4.1 Functional Requirements

* FR1: Registered student can upload and view past papers.
* FR2: Registered student can upload and view assignments.
* FR3: Registered student can upload and view quizzes.
* F-R4: Registered student can upload and view notes.
* FR5: Registered student can update his files.
* FR6: Registered student can delete his files.
* FR7: Admin can register new students.
* FR8: Admin can maintain student records.
* FR9: Admin can update student’s records.
* FR10: Admin can delete any students record.

## 4.2 Non-Functional Requirements

* NFR1: Performance of system should be fast and accurate.
* NFR2: Should be able to handle large amount of data
* NFR3: System shall be operational 24 hours a day and 7 days a week.
* NFR4: System shall provide 100% access reliability.
* NFR5: Changes (new additions, password changes, database changes) must be verified once per day.

# 5 User Documentation and Help System Requirements

[1] See attached document “Prototype”

# 6 Design Constraints

## 6.1 Implementation language

HTML

CSS

JAVA Script

PHP

## 6.2 Tool used

Laravel Framework

# 7 Purchased Components

N/A

# 8 Software Interfaces

[2] See attached document “Prototype”

# 9 Licensing and Security Requirements

N/A

# 10 Legal, Copyright and Other Notices

N/A

# 11 Applicable Standards

This document follows IEEE-830 Standard.

# 12 Glossary

Laravel Framework: Laravel is a free, open-source PHP web framework, created by Taylor Otwell and intended for the development of web applications following the model–view–controller architectural pattern and based on Symfony.

JavaScript - A programming language designed by Sun Microsystems, in conjunction with Netscape,

that can be integrated into standard HTML pages.

JavaScript - A programming language designed by Sun Microsystems, in conjunction with Netscape,

that can be integrated into standard HTML pages.

JavaScript: A programming language designed by Sun Microsystems, in conjunction with Netscape, that can be integrated into standard HTML Pages.

PHP: An HTML-embedded Web Scripting language.