**Software Requirements**

**Specification**

**for**

Course Registration System

Version 1.0

Prepared by: Team Innovators (Sreekanth kumar Vobilishetty, Akhila Yarlagadda, Piyusha Varshini Tirukovalluru, Meghana reddy Akkati)

University of North Texas

**Table of Contents**

1. Introduction .................................................................................................................................. 1.1 Purpose .................................................................................................................................. 3

1.2 Scope ..................................................................................................................................... 3

1.3 Definitions, Acronyms and Abbreviations ............................................................................ 3

1.4 References ............................................................................................................................. 3

2. Overall Description ................................................................................................................. 2.1 Product Perspective ............................................................................................................... 4

2.2 Product Features .................................................................................................................... 4

2.3 User Characteristics ............................................................................................................... 6

2.4 Design and Implementation Constraints ............................................................................... 6

2.5 Assumptions and Dependencies ............................................................................................ 7

3. System features…………………………………………………………….

3.1 Student features…………………………………………………………….7

3.1.1 Course Enrollment Feature…………………………………………………8

3.1.2 Finance Feature…………………………………………………………...8

3.1.3 Course Details Feature…………………………………………………….10

3.2 Admin Features…………………………………………………………….11

3.2.1 Course Edit Feature……………………………………………………….11

3.2.2 Finance Feature…………………………………………………………..11

3.2.3 Student Grades Feature……………………………………………………11

3.2.4 Enrollment Details Feature………………………………………………...11

4. External Interface Requirements................................................................................................ 4.1 User Interfaces ..................................................................................................................... 12

4.2 Hardware Interfaces ............................................................................................................ 13

4.3 Software Interfaces .............................................................................................................. 13

4.4 Communications Interfaces ................................................................................................. 14

5. Non-functional Requirements ................................................................................................. 14 5.1 Product Requirements ........................................................................................................14

5.2 Organizational Requirements .............................................................................................. 15

5.3 External Requirements ........................................................................................................ 15

6. Development Phases .................................................................................................................. 6.1 Development Phase I ........................................................................................................ 15

6.2 Development Phase II ............................................................................................................ 16

6.3 Development Phase III ........................................................................................................... 16

7 Other Requirements…………………………………………………………………………….16

**1. INTRODUCTION**

* 1. **Purpose**

We often see still many universities are following offline approach, paper method to register for classes. Many of us have filled a paper to register for courses and we did mess up doing so. We do think to eliminate the hassles we face while applying to courses. This motivation brings us to develop this project “COURSE REGISTRATION SYSTEM”.

COURSE REGISTRATION SYSTEM is an online website used for applying courses. Student login into website with valid credentials and can register to the classes. Admin has privileges to administer the courses.

**1.2 Scope**

The main scope of the project is to develop a website for course registration system. Benefits include save paper and fasten the process to register with few clicks. Institution can track student payments, course registration details such as how many registered into a class and dues of a student. Monitor whole courses from the website.

**1.3 Definitions, acronyms and abbreviations**

**JAVA** Java is a broadly useful programming language that is class-based, object-oriented (in spite of the fact that not it’s not a pure object oriented language and intended to have a couple of usage conditions as could reasonably be expected. It is proposed to give application engineers a chance to write once, run anywhere (WORA), implying that compiled Java code can keep running on all platforms without java being installed on the Operating system. [[1](https://en.wikipedia.org/wiki/Java_(programming_language))].

**Spring tool Suite** is an Eclipse-based development environment that is altered for creating Spring applications. It gives a prepared to-utilize condition to execute, troubleshoot, run, and deploy your Spring applications, including integrations for Pivotal tc Server, Pivotal Cloud Foundry, Git, Maven, AspectJ, and goes ahead top of the most recent Eclipse releases. [[2](https://spring.io/tools3/sts)].

**Application programming interface (API)** is an interface or protocol used for communication between a client and a server expected to improve the structure of client side programming. It has been portrayed as a "contract" between the client and the server, to such an extent that if the client makes a request in a particular format,[[3](https://en.wikipedia.org/wiki/Application_programming_interface)].

**UI** – User Interface

**JDK** – Java Development Kit

**OS** – Operating System

**API** – Application Programming Interface

**IDE** – Integrated Development Environment

**SQL** – Structured Query Language

**1.4** **References**

[1]<https://en.wikipedia.org/wiki/Java_(programming_language)>

[2]<https://spring.io/tools3/sts>

[3]<https://en.wikipedia.org/wiki/Application_programming_interface>

[4] <https://en.wikipedia.org/wiki/Base64>

**2. OVERALL DESCRIPTION**

**2.1 Product Perspective**

Current system is of offline based or many universities still in the nascent stage of developing course registration system or many features doesn’t present in the current system as they are in proposed systems

Courses

Contacts

Has

Department

enrolls

Admin

Student

**2.2 Product Features**

The Course Registration System mainly consists of two roles.

1. Student
2. Admin

**Student:**

When a student applies for a university and he is accepted in, he would be provided with a unique Student ID (Assumption). Once he has his ID, he is supposed to register for Course Registration System to enroll for courses where he enters all his details and creates a new password to register.

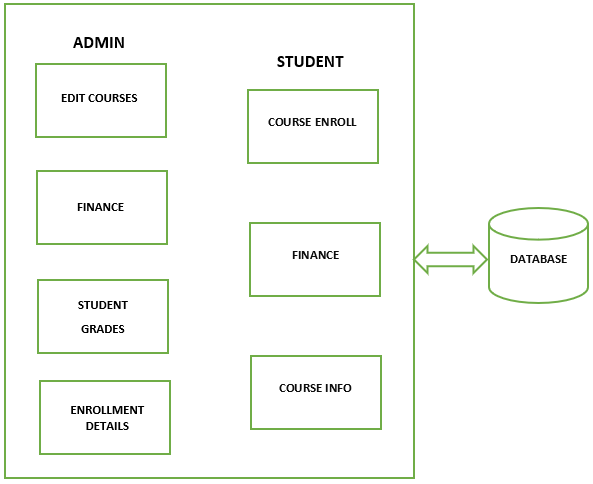
After the registration, student can login to our system with his student ID and newly created password. The three main functionalities at the student end are

* Course Enroll
* Finance
* View Course info

For each semester, he needs to enroll in at least 3 courses to be considered as a full-time student. In this site, he can enroll in courses, make payments for the registered courses, view his courses information like schedule, grades etc.

**Admin:**

The Course Registration System has an admin who has unique ID and password. Admin can edit course details such as professor, strength etc. Admin can also accept payments, post grades. Finally, admin can see all the details of students registered in particular class.



**2.3 User Classes and Characteristics**

User must have basic knowledge on browsing internet and websites. User must also have a mail id and should be familiar with it.

**2.4 Operating environment**

**2.4.1Software requirements:**

IDE: Spring tool suite

Language: Java, HTML, CSS, JS

Framework: Spring boot

Database: PostgreSQL.

**Hardware requirements:**

Operating System: Windows (any Microsoft supported version)

RAM:4GB

ROM: 5GB (min)

**2.5 Design and implementation constraints**

* User must always have a laptop or PC to access the website.
* Downtime of application required for migration of database.
* Downtime of application is required for updating software and server.
* This application is limited to development in Java related technologies.
* HTTP, SMTP protocols are required to use.

**2.6 Assumptions and Dependencies**

* Assume that student has user id to register himself/herself in the website.

**3. SYSTEM FEATURES**

This system has 3 main features to complete the course enrollment and related tasks.

1. User authentication features
2. Student features
3. Admin features

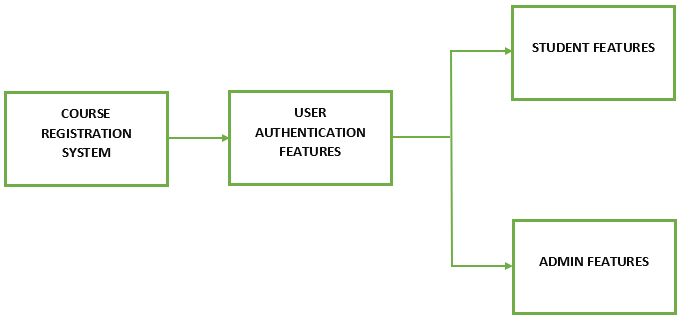


Figure 3.1 Feature Hierarchy

The above figure 3.1 shows the feature hierarchy for course registration system. User (Admin/Student) has to login to the site to access his course registration account. To log in, he is supposed to sign up in prior. As mentioned above, while using the course registration system in real time, user authentication features are the initial features used by the user as he needs to login/register. After logging in successfully, the remaining features can be accessed.

**3.1 Student features**

The student features include all the functionalities available to a student user. The basic functionalities for a student are

1. Course Enrollment
2. Finance
3. Course details

Student features also include the development of home page with all tabs available for a student. These tabs are common for all the UI pages student navigates to.

**3.1.1 Course Enrollment Feature**

Course Enrollment should basically provide the opportunity to enroll for a class which the student wants to take. This feature includes the following sub-features.

1. Add – Student should be able to add desired courses (which are available) to the shopping cart.
2. Drop – Student should be able to drop any of the courses which he is already enrolled in before certain deadline.
3. Swap – Student should also be allowed to swap an already enrolled course with some other course (available course) of his choice.
4. Enroll – After adding to the shopping cart, this is a final step to get registered. The student should be enrolled when he clicks ‘enroll’. If the class is full, the student must get added to the waiting list.

**3.1.2 Finance Feature**

Finance feature includes all the functionalities related to student’s payments. This includes the following sub-features.

1. Account Balance – This tab should show the total due amount of a student for all the courses.
2. Summary – This tab should show the brief summary of the due amount like course wise dues.
3. Make Payment – This tab should allow student to enter the desired amount and allow him to make the payment.
4. Payment History – This tab must show the student’s previous payment details.

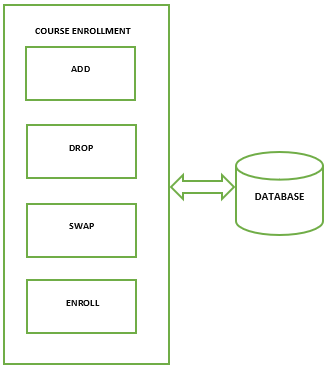


Figure 3.2 Course Enrollment Features

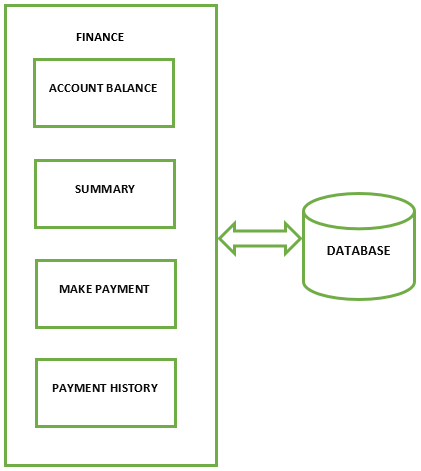


Figure 3.3 Finance feature

**3.1.3 Course Details Feature**

Course Details should show the information of the courses which the student is enrolled in. This includes the following sub-features.

1. Class Schedule – This tab should show the course details which the student is enrolled in. Course details include course id, course name, Room number, Building, Professor name, cost of the course.
2. View Grades – This tab should display the grades of the courses the student has already completed.
3. Mandatory courses – This tab should display all the mandatory courses in the student’s department and differentiate between the courses which student has already completed, student has currently enrolled and the student has not enrolled in.

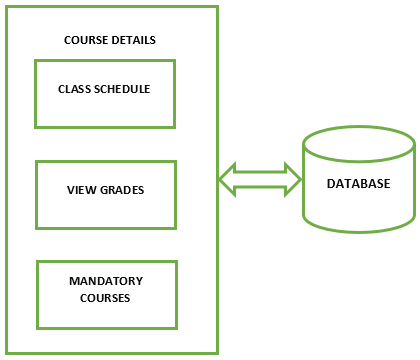


Figure 3.4 Course Details feature

**3.2 Admin Features**

The admin features include the functionalities available to an admin user. The basic functionalities for a student are

1. Course Edit
2. Finance
3. Student Grades
4. Enrollment Details

Admin features also include the development of home page with all tabs available for admin. These tabs are common for all the UI pages admin navigates to.

**3.2.1 Course Edit Feature**

The admin should be able to edit a course like updating course name, course ID, professor change, course timings, course cost etc. The admin should also be able to add new courses to the list.

**3.2.2 Finance Feature**

The admin should be able to edit the account balance of any student if in case of physical payments. The admin should also be able to view all students’ dues.

**3.2.3 Student Grades Feature**

The admin should be able to post grades of a student for the courses which he has completed in course registration portal. Admin can also edit the grades, mute and unmute them.

**3.2.4 Enrollment Details Feature**

The admin should be able to view all the students list in a specific course, filled courses, available courses.

**4. EXTERNAL INTERFACE REQUIREMENTS**

**4.1 User Interface**

The UI consists of many screens to make it user friendly. Initially, we have both login page where it allows the students and admin to login with their credentials and a signup page which allows students to create their account with the University.

Depending upon the role, the next page displays. If the logged in user is admin, then we have a page with the following tabs namely Search, edit course, Finance, Grades, Enrollment details, Contact US, Logout. Else if the logged in user is Student then the display page differs with the respective tabs such as Search, Enroll, my classes, Finance, Contact Us, Logout.

Each tab opens to a new page with the existing features of the previous page as well as new features of the respective tab. Each tab has a dropdown options of various other tabs which opens to another page with respective features.



***Search tab*** allows the course search related to the respective stream and semester.

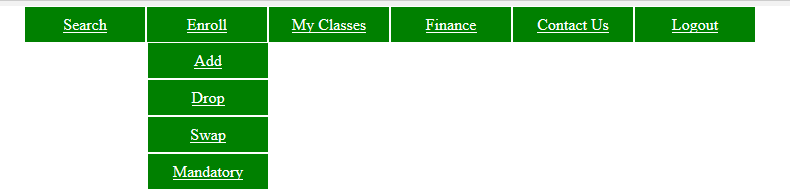
***Edit course tab*** allows the admin to edit the courses like the class strength and course instructor.

***Finance tab*** allows the admin to view the dues and payments made by the students.

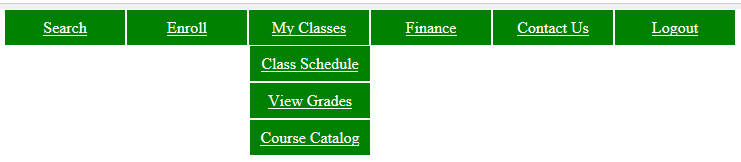
***Grades tab*** allows the admin to allot the grades to the students for their respective stream and semester.

***Enrollment details tab*** allows the admin to view the details of the students enrolled for the courses.

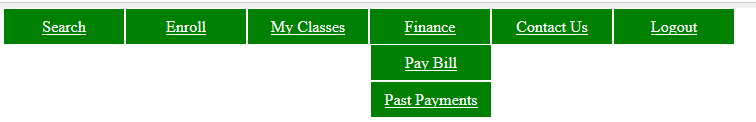
***Enroll tab*** has Add, Drop, Swap and Course Catalog options which opens to a new page having their own functionality.



***My classes tab*** has Class schedule, View grades and Mandatory courses tabs where Class schedule tab views the class schedule of the student, View grades tab displays the individual course grades of the student depending on the semester selected. Mandatory courses tab displays the mandate courses for the respective stream.



***Finance tab*** allows the users to make payments for the respective semester and courses. It also allows them to view the payment history for their account.



***Contact Us tab*** has the information about the University and admin.

***Logout tab*** logs the users out of the page or application.

**4.2 Hardware Interfaces**

The hardware interfaces include a web application which is operated using Windows OS and the interaction between the students and admin can be done via Email service which is done using SMTP protocol (Simple Mail transfer protocol).

Operating system: Windows (any Microsoft supported version)

RAM: 4 GB

ROM: Minimum 5 GB

**4.3 Software Interfaces**

The software interface acts as a connection between the web application and the local server (Apache Tomcat). The database we use here is PostgreSQL. It is a client- server database. All the emails that are sent will go to the local database. We develop this web application using HTML, CSS, Javascript and Java.

Operating System: Windows (any Microsoft supportive version)

Platform: Spring boot JDK framework

Application server: Apache Tomcat

IDE: Spring tool suite

Technologies Used: Java, HTML, CSS, Javascript

Database: PostgreSQL

**4.4 Communication Interfaces**

For communication interface, SMTP protocol is used to provide email services. HTTP protocol is used for the communication between the application server and the systems of the users across the web. The connection between the server and the database is done by using JDBC/ODBC protocol.

**5. NON - FUNCTIONAL REQUIREMENTS**

**5.1 Performance Requirements**

Performance mainly relies on response time, workload, scalability and platform.

***Response time:***

Our aim is to reduce the response time for all the pages in the website having it equal to 0.1sec and we are making sure that there is minimum amount of acceptable error rate.

***Workload:***

The workload is often described as the things that the users are likely to execute that particular day. There will be many request submitted by the user each time they use the website.

For example, a system may have external customers, internal staff providing data entry and batch processes such as backups. If the backup is not completed overnight then it may seriously disrupt the performance experienced by the users the next day.

The performance of the system is dependent on how the load is delivered to the system. For example it is easier to achieve faster response times for a system that receive a regular arrival of work throughout the day compared to one that receive burst of traffic. Hence it is very important to handle the workload to not let it hinder the performance.

We are aiming to support 100 users at a time and be able to access 6 pages/sec.

***Scalability:***

Scalability is specified as the increase in the system’s workload without affecting the performance of the application. The scalability depends on the system lifespan and maturity of the system.

We are planning to use an older system which has more accurately defined workloads rather than a new system so that scalability does not introduce any problems.

***Platform considerations:***

A platform is defined as the hardware and software which are used in the system for the application. Using appropriate hardware and software helps in the performance. For example, If a hardware is shared between other systems, then it’s efficiently is decreased. Hence we should make sure its processor and resources are solely used by the application itself.

**5.2 Safety Requirements**

Internet safety is a must while you are doing something on the Web. We are achieving the safety by not asking for any personal information, Bank information etc from the users. Also, making sure that after every process request, it reaches to a safe state within the response time.

In Power loss situations, we have to maintain the system consistent since we are dealing with the finances of making payments and enrolling for the courses.

High or low demand rates also are responsible for affecting the safety of the system.

There should be proper methods for Proof testing the system and its actions. Finally, the environmental hazards also need to kept in mind in order have safety requirement.

**5.3 Security Requirements**

Security is a primary consideration when choosing a Web service for all the applications. In our website we are striving to provide authentication, authorization, non-repudiation and data protection.

**Authentication**:  
Authentication ensures that each user whether am admin or student must be valid that is must be enrolled with the University having University provided credentials. We are achieving this by validating the existing details against the authorized one’s.

**Authorization:**  
Authorization means whether the service provider has granted access to the Web service to the requestor. It means that an admin and student both have different functionalities and only the authorized user (whether an admin or student) must be able to login to correct page and perform respective activities.

**5.4 Software Quality Attributes**

Some of the Software Quality Attributes we are trying to achieve are Accessibility,

Accountability, Availability, Durability, Integrity, Interoperability, reliability, Usability.

Since our application can be used by everyone who are enrolled with the university.

It is available around the clock every day.

It is durable supporting 100 users at a time and having response time 0.1 sec.

It is reliable since we are using efficient software and hardware.

# 6. DEVELOPMENT PHASES

Our development team has identified few major features for both the admin and student roles separately and implement them in the 3 phases of development cycle, namely user authentication features which is common for both the roles. Student features are features to enroll the courses, features to perform finance operations and features to view the course details. Admin features are features to edit the courses, features to finance operations, features to edit student grades, features to view the enrollment details.

## 6.1 Development phase – I

In this phase we develop the basic user authentication feature for both the student and admin, because security is primary concern for any website, here the student can access the website with there valid credentials. We also decided to implement the complete enrollment features of the student. The functional requirements which are implemented in this are:

* User Registration
* User Login
* Features to add the course
* Features to drop the course
* Features to swap the course
* Features to search for a course
* Features to finish enrolling a course
* Features to view the course catalogue

## 6.2 Development phase – II

In this phase we enhance the features of the student, here we intended to include the finance operations performed by the student to get enrolled into the courses. We also intended to include the complete features to view the course details. The functional requirements to be implemented in this phase are:

* Features to make payment for a course
* Features to ow the account balance
* Features to show the complete summary of the payment
* Transaction history
* Features to view the class schedule
* Features to view the grades to student
* Features to view the mandatory courses

## 6.3 Development phase – III

In this phase we completely concentrate on the side of admin. This phase will be intended to implement the complete features of the admin likely edit courses, finance, student grades and enrollment details. The functional requirements implemented in this phase are:

* Features to edit the courses
* Features to post the grades
* Features to confirm the payments and view the dues
* Display the course enrollment details

**7.OTHER REQUIREMENTS**

