Software Requirements Specification

for

Student Management System

Version 2.0

Prepared by Akshay U Prabhu (15IT203)

National Institute of Technology, Surathkal

20-Feb-2018

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Student Mangement System	7-Jan-2018	Created basic SRS	1.0
Student Mangement System	20-Feb-2018	Updated functional and non-functional requirements	2.0

1. Introduction

1.1 Purpose

This project gives all the services that must be provided to a student over the internet to find course details provided by that administrator of the college.This project contains data regarding student personal details which can be updated by the student and viewed by the administrator. It provides the detailed information about the course details given at the college. It provides the visibility in the design and provides information needed for software support.

1.2 Document Conventions

Main topics are bolded in heading (whole numbered) followed by subtopics (decimal numbered) and bullets. All acronyms have been introduced with their full names, followed by the acronym in the parenthesis.

1.3 Intended Audience and Reading Suggestions

The intended audience for this Student Management System is the internal guides(teachers), administrator of the college/dept and students of the organization.

1.4 Product Scope

Online Student Management System(SMS) is developing for general purpose and used to replace old paper work system. The objective is to build upon the existing information system in order to efficiently provide student information to students and school administration. This increase in efficiency of administration, provide details about the courses,make announcements and results. It provides a mechanism to edit the student information form which makes the system flexible.

1.5 References

[1] Software Requirements Specification for LNMIIT's STUDENT MANAGEMENT PORTAL Version 1.0 retrieved from
https://www.slideshare.net/SumanSaurabh9/srs-for-student-database-management-system

[2] The reference for this SRS template is retrieved from
https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc

[3] SRM University. (2012 - 2013). CS0411 - Software Engineering Lab - Laboratory Manual. Retrieved from
http://www.srmuniv.ac.in/sites/default/files/files/SOFTWARE%20Engineering%20LAB-CS0411.pdf

2. Overall Description

2.1 Product Perspective

Student Management System is capable of managing each and every data regarding student, announcements etc. Student Management System helps us in managing in an extremely efficient way.
This Student Management System works in an efficient manner. We have three users in this project. Admin, Teacher and Student. Admin can maintain the details of students,create the announcements. Teacher can make announcements for their course and add students to the course. Student can edit their personal details and can view the announcements.

The SMS will work with the following hardware interfaces:

- ***Hard disk** : The database connectivity requires a hardware configuration with a fast database system running on high rpm hard disk permitting complete data redundancy and backup systems to support the primary goal of reliability.*
- *The system must interface with the standard output device, keyboard and mouse to interact with this software.*

The SMS will work with the following software interfaces (both system and user interfaces (UI)):

- ***Back End** : Node server..*
- ***Front End** : Any compatible Browser.*

2.2 Product Functions

The Student Management system allows users to perform a series of actions to aid in managing the courses and administration of the students process. These functions include the following:

- *Complete details of the student can be stored and retrieved.*
- *Admin can see all the student's details and also export to excel sheets.*
- *The student can view all his details, course details and the announcements made by the admin or the teacher.*
- *The students can also update their details.*
- *The Teacher can add students to the course or remove them.They can also make announcements to a particular course.*

2.3 User Classes and Characteristics

The users of the system are students, teachers and the administrators who maintain the system. The users are assumed to have basic knowledge of the computers and Internet browsing. The administrators of the system to have more knowledge of the internals of the system and is able to rectify the small problems that may arise due to disk crashes, power failures and other catastrophes to maintain the system. The proper user interface, users manual, online help and the guide to install and maintain the system must be sufficient to educate the users on how to use the system without any problems.

2.4 Operating Environment

This SMS application has chosen the following elements of the environment for managing the students online:

- *Linux or Windows based Operating System (OS) will be used to house the application.*
- *The language chosen for the deployment of the application would be JavaScript on the server side as well as on the client side*
- *The UI will be handled by HTML along with JS and CSS.*
- *The database which holds student,teacher and admin information and course details is deployed in a NoSQL Database.*

2.5 Design and Implementation Constraints

The following are the constraints laid before utilizing the SMS application in full scale:

- *The Student Management System software is designed in such a way that the user can easily interact with the screen because of GUI.*
- *The information of all the users must be stored in a database that is accessible by the Online Student Information Management System.*
- *The university Information security systems must be compatible with the Internet applications.*
- *The Online Student Information Management System is connected to the university computer and is running all 24 hours a day.*
- *The users must have their correct usernames and passwords to enter into the Online Student Management System.*
- ***Reliability requirements** : Data redundancy and use of special characters must be avoided.*
- ***Safety and Security considerations** : The user must always exit the application after logging out of their account normally.*
- *Programming language requirements : The developers and programmers must have experience with JavaScript and NoSQL Database.*

2.6 User Documentation

N/A

2.7 Assumptions and Dependencies

- *The users have sufficient knowledge of computers.*
- *The University computer should have Internet connection and Internet server capabilities.*
- *The users know the English language, as the user interface will be provided in English*
- *The product can access the university student database*

3. External Interface Requirements

3.1 User Interfaces

All pages of the system are following a consistent theme and clear structure. The occurrence of errors should be minimized through the use of checkboxes, radio buttons and scroll down in order

to reduce the amount of text input from user. JavaScript implement in HTML in order to provide a Data Check before submission. HTML Tables to display information to give a clear structure that easy to understand by user. Error message should be located beside the error input which clearly highlight and tell user how to solve it. If system error, it should provide the contact methods. The page should display the project process in different colour to clearly reflect the various states that student done. Each level of user will have its own interface and privilege to mange and modify the project information such as supervisor able to monitor/manage his student progress and make comment on it, student can change his detail, view the progress, submit project idea. The System should provide a feedback form for all users to give comments or asking questions. It should provide a FAQ to minimize the workload of system administrator.

3.2 Hardware Interfaces

3.2.1.1 Server Side

The web application will be hosted on one of the department's Linux system using node server and connecting to one of the school Database server. The web server is listening on port 3000.

3.2.1.2 Client Side

The system is a web based application; clients are requiring using a modern web browser such as Mozilla Firebox 1.5, Internet Explorer 6 and Enable Cookies. The computer must have an Internet connection in order to be able to access the system.

3.3 Software Interfaces

The following software interfaces are used in the deployment of the Student Management System :
NoSQL Database, Javascript, NodeJS, Twitter Bootstrap etc.
The Database should have the following specifications:

Name of the Interface	*Description or Purpose*
Type System	*Dynamic, Static*
Architecture	*Object Oriented Model*
Software License	*Proprietary*
Operating System	*Windows, Mac OS X, Linux, Unix*
Max DB Size	*Unlimited*

3.3.1.1 Server Side

The UOP already has the required software to host a Javascript web application. A NodeJS server will accept all requests from the client. A development database will be hosted locally (using MySQL); the production database is hosted centrally.

3.3.1.2 Client Side

An OS is capable of running a modern web browser which supports HTML version 3.2 or higher.

3.4 Communications Interfaces

The communications functions required by this product are LAN connection within the whole company so that the Admin, employee, and customer can interact with each other. We use TCP/IP protocol.

4. System Features

This section outlines the low level details (implementation) of every system function of the Student Management System application. It sets the priorities of each function so that the developers would have a lucid picture of what will be accomplished first and what functions will be available to review first.

4.1 Login

4.1.1 Description and Priority

There are 3 types of logins in this application.
Student login, Teacher login and Admin login.
The users have to enter their respective credentials to login.

4.1.2 Stimulus/Response Sequences

After the user logs in, they are redirected to their corresponding home pages, I.e if the user is an admin he will be redirected to admin home page, similarly teacher will be redirected to teacher home page and student will be redirected to student home page.

4.1.3 Functional Requirements

REQ-1: An input box will be provided to enter the username and the password of the user.
REQ-2: The password entered should be seen as a list of '*****'.
REQ-3: If either the username or password is incorrect , a popup should be shown saying 'either the username or password incorrect'.
REQ-4: If both of them match to atleast one of the user in the database then they should be redirected to their respective home pages

4.2 Update Details

4.1.1 Description and Priority

Admin, Student and Teacher can update their respective details.However Admin can update details of anyone (even the student and the teacher)

4.1.2 Stimulus/Response Sequences

After the user updates the details, the details in the database will also get updated.

4.1.3 Functional Requirements

REQ-1: An input box will be provided to enter the username and the password of the user.
REQ-2: The password entered should be seen as a list of '*****'.
REQ-3: If either the username or password is incorrect , a popup should be shown saying 'either the username or password incorrect'.

REQ-4: If both of them match to atleast one of the user in the database then they should be redirected to their respective home pages

4.3 Add Students to the Course

4.1.1 Description and Priority

Upon the successful login of the admin or the teacher, they can add the students to a particular course.If the logged in user is an admin he can add any student to any course in the college or department.If the logged in user is a teacher then he/she can add students to only those courses that they are alloted to.

4.1.2 Stimulus/Response Sequences

After the teacher or the admin adds the student to the course, the student can view the course details of the course which he is added to. The student will also be able to get all the announcements made by the admin or the teacher in that particular subject or course.

4.1.3 Functional Requirements

REQ-1: In the Admin home page a link named **Student** will be provided which on clicked redirects the admin to the student subpage.
REQ-2: In the student subpage a list of names of the students will be provided which upon clicked the admin can edit the students details
REQ-3: A drop down menu will be provided which has a list of courses in the department along with a Add Course button.
REQ-4: The admin can select a course from the dropdown menu and click on the Add Course button provided.
REQ-5: In the Teacher home page a list of the courses that he/she is alloted is displayed in the form of button.
REQ-6: Upon clicking on the course name buttons, he/she will be redirected to the corresponding course page.
REQ-7: In the course page, a input box will be provided to enter the name of the student to be added.

4.4 Make Announcements

4.1.1 Description and Priority

Upon the successful login of the admin or the teacher, they can make announcements to a particular course.If the logged in user is an admin he can make announcements to any course in the college or department.If the logged in user is a teacher then he/she can make announcements to only those courses that they are alloted to.

4.1.2 Stimulus/Response Sequences

After the teacher or the admin makes the announcements to the course, the student can view the announcement details which is added to his course.

4.1.3 Functional Requirements

REQ-1: In the Admin home page a link named **Announcement** will be provided which on clicked redirects the admin to the announcement subpage.
REQ-2: In the announcement subpage a list of all the announcements made will be displayed which upon clicked the admin can edit the announcement details.

REQ-3: A create button is provided on clicking which will open a Input box.
REQ-4: The user should enter the details of the announcement and then click on create button.
REQ-5: The announcement will be displayed to all the students who have taken the corresponding course.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The event management system requires a good browser which can render JavaScript with good speed. The application also requires a good internet connection. Browsers such as Google Chrome, Mozilla Firefox, IE8 or Microsoft Edge are recommended. The performance of the system depends on various factors such as the network and server traffic. Server load can dramatically increase with more and more users using the application, thereby increasing the number of database queries.

5.2 Safety Requirements

The user must make sure that he/she has set a strong password in order to prevent his/her account from getting compromised. The system can enforce this by specifying constraints on the password such as setting the minimum password length, using numbers and special characters in the password,etc.

5.3 Security Requirements

- *To prevent unauthorized access to the application and in order to prevent false data manipulation, only users who have registered can add new events.*
- *Only admin has the control over the entire database, so that no one else can change the database.*

5.4 Software Quality Attributes

- ***Correctness** : The information displayed about the events must be correct.*
- ***Maintainability**: The web application should be easy to extend and maintain.*
- ***Portability**: The website should be accessible on all kinds of devices.*
- ***Usability**: Interface should be user friendly with respect to the media device through which it is being accessed.*

5.5 Business Rules

- *The admin has the control over the entire database.*
- *The teacher can add students to the course which he/she is taking or alloted to.*
- *The student can only edit his/her profile and see the announcements by the teachers.*

6. Other Requirements

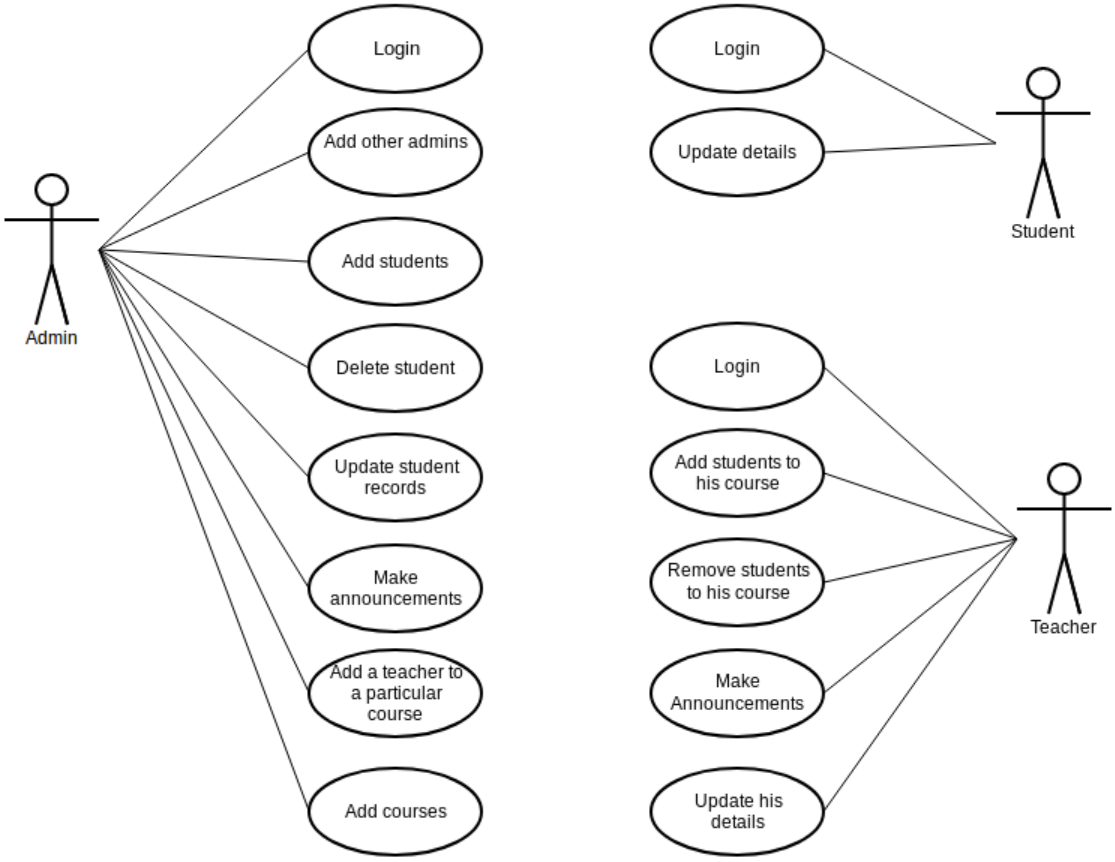
<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

N/A

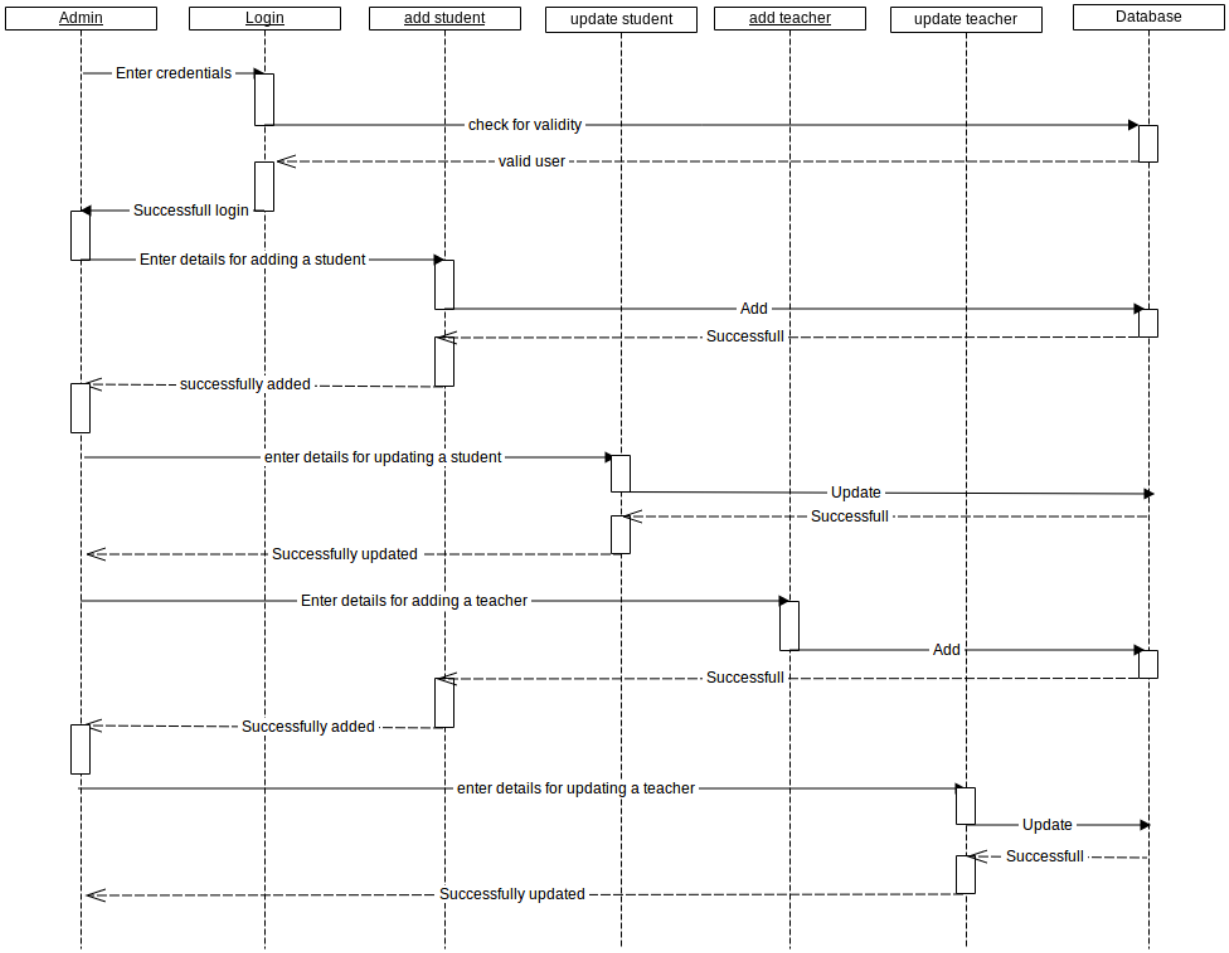
Appendix B: Analysis Models

USE CASE DIAGRAM:

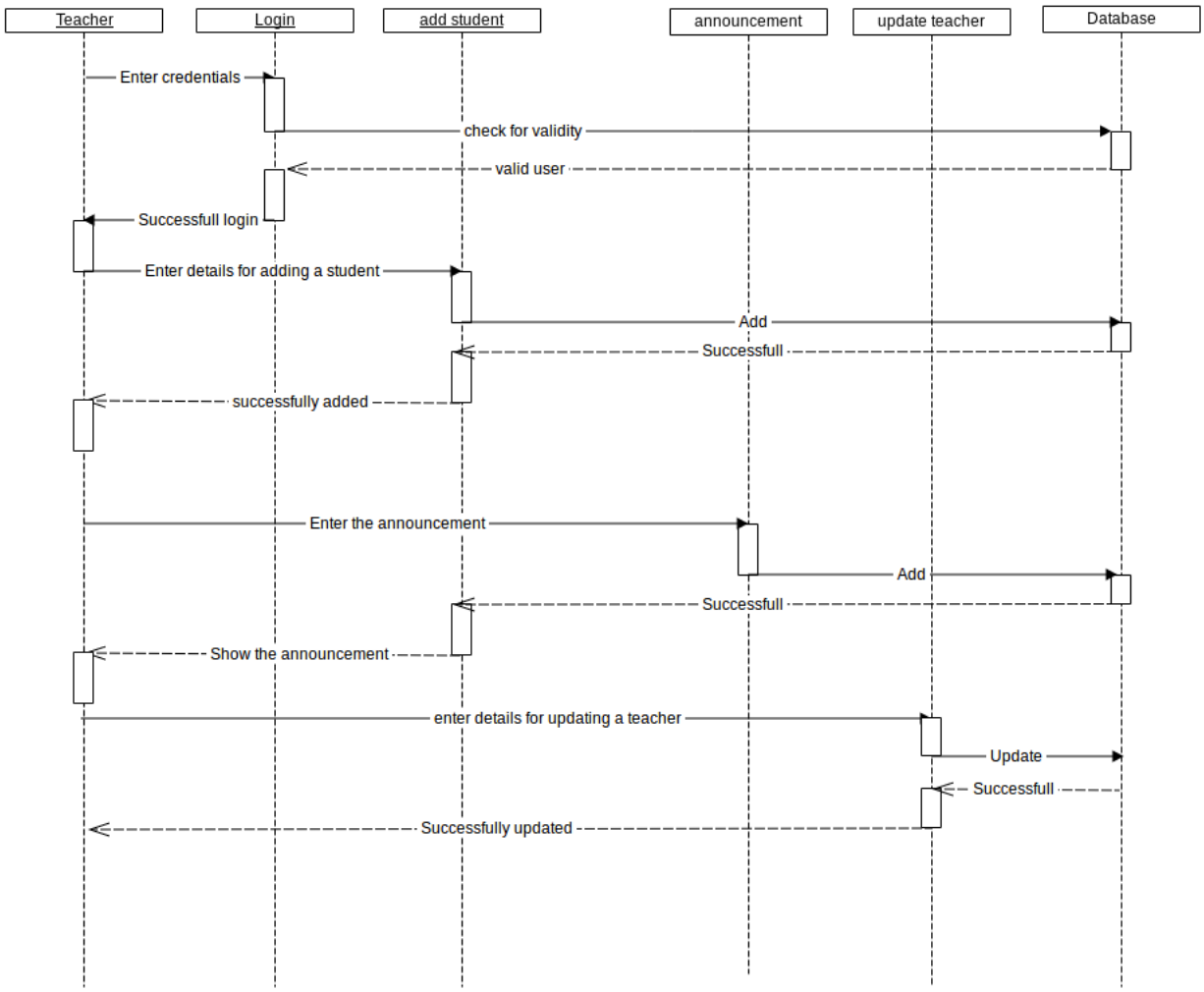


SEQUENCE DIAGRAM:

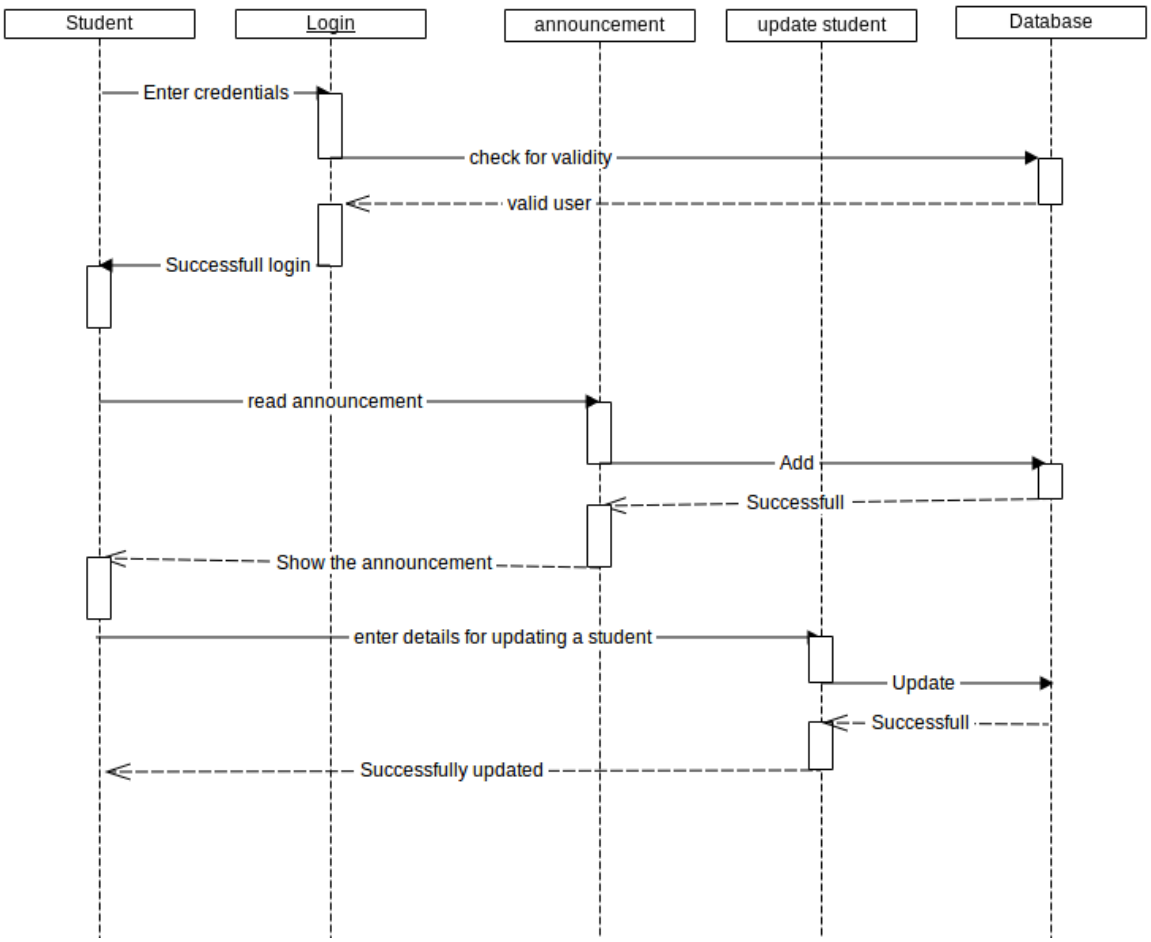
For Administrator:


Admin
Login
add student
update student
add teacher
update teacher
Database
Enter credentials
check for validity
valid user
Successfull login
Enter details for adding a student
Add
Successfull
successfully added
enter details for updating a student
Update
Successfull
Successfully updated
Enter details for adding a teacher
Add
Successfull
Successfully added
enter details for updating a teacher
Update
Successfull
Successfully updated

For Teacher:

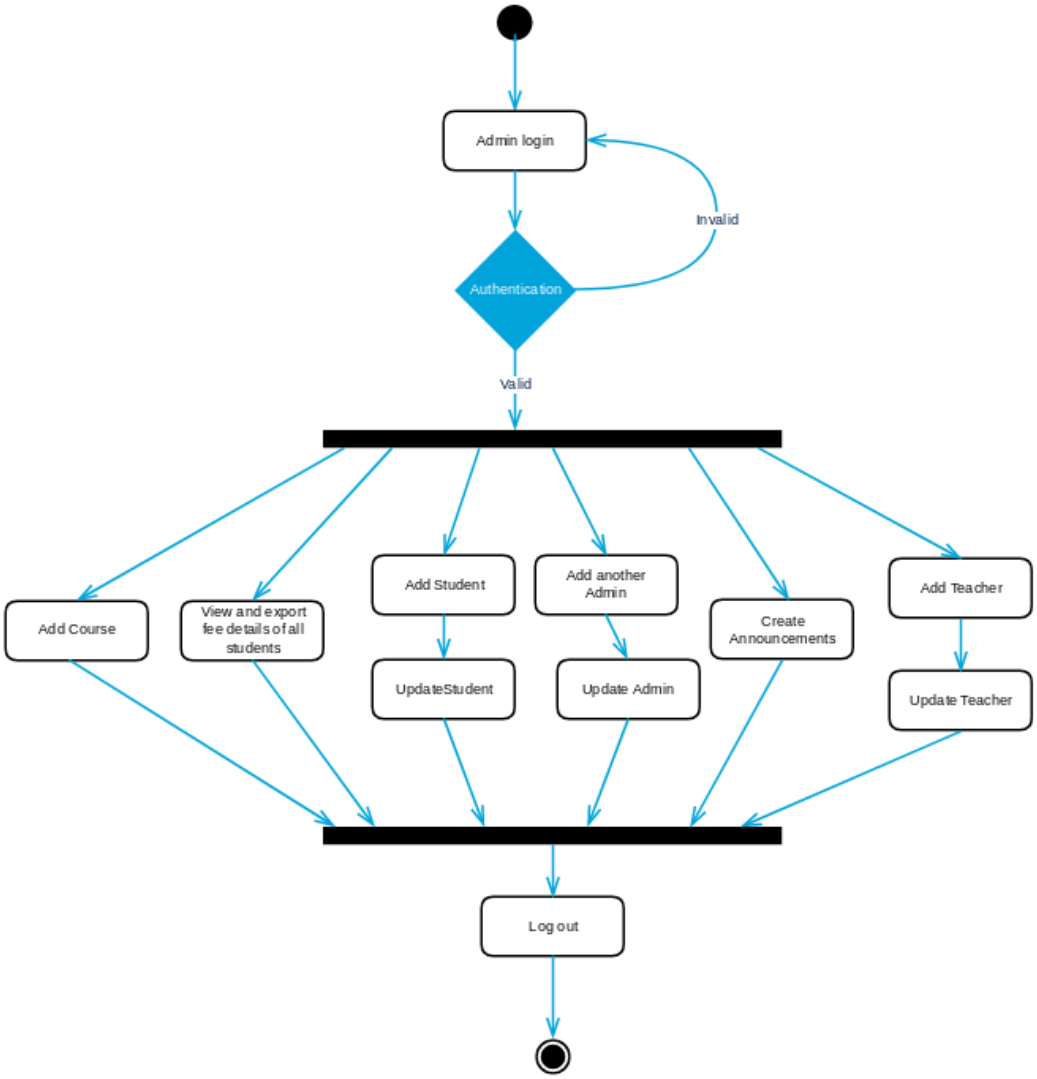

Teacher
Login
add student
announcement
update teacher
Database
Enter credentials
check for validity
valid user
Successfull login
Enter details for adding a student
Add
Successfull
successfully added
Enter the announcement
Add
Successfull
Show the announcement
enter details for updating a teacher
Update
Successfull
Successfully updated

For Student:

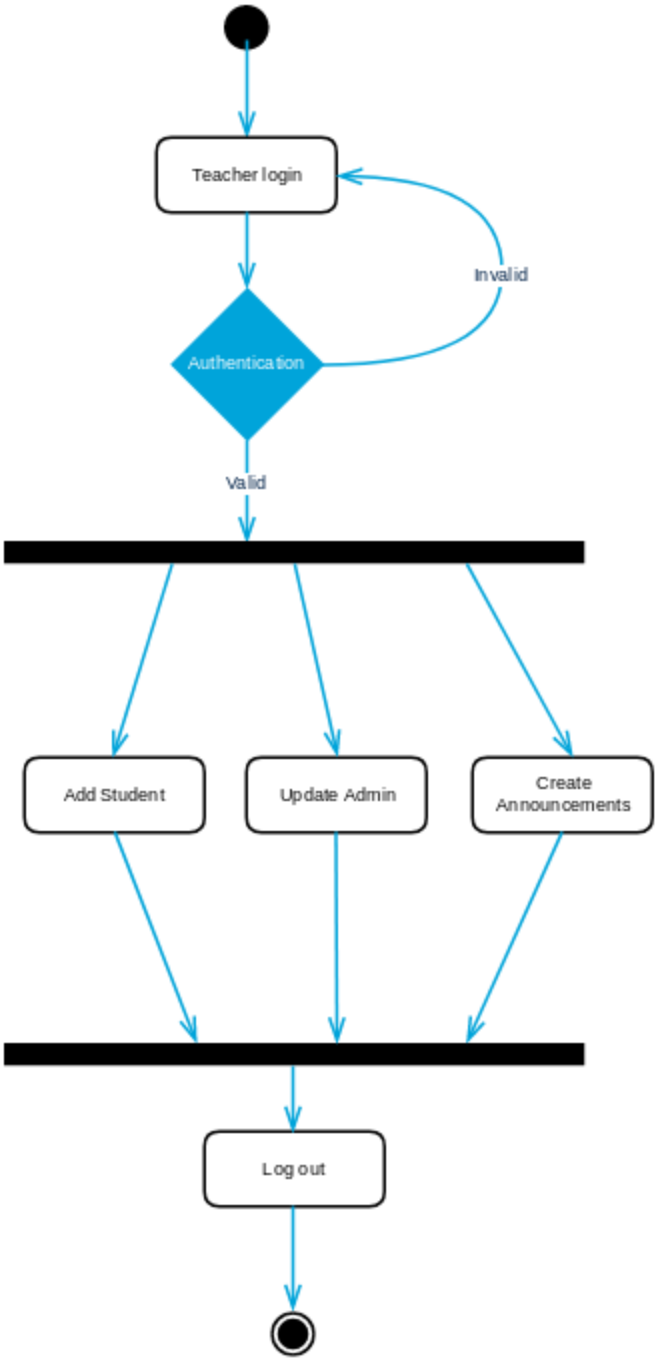

Student
Login
announcement
update student
Database
Enter credentials
check for validity
valid user
Successfull login
read announcement
Add
Successfull
Show the announcement
enter details for updating a student
Update
Successfull
Successfully updated

ACTIVITY DIAGRAM:

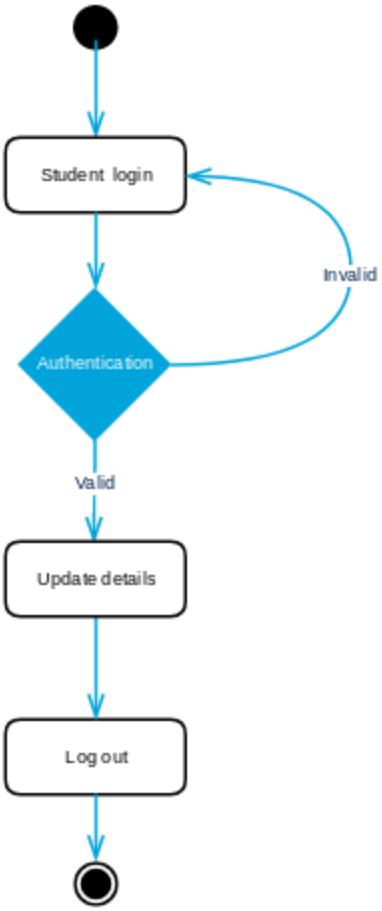
For Admin



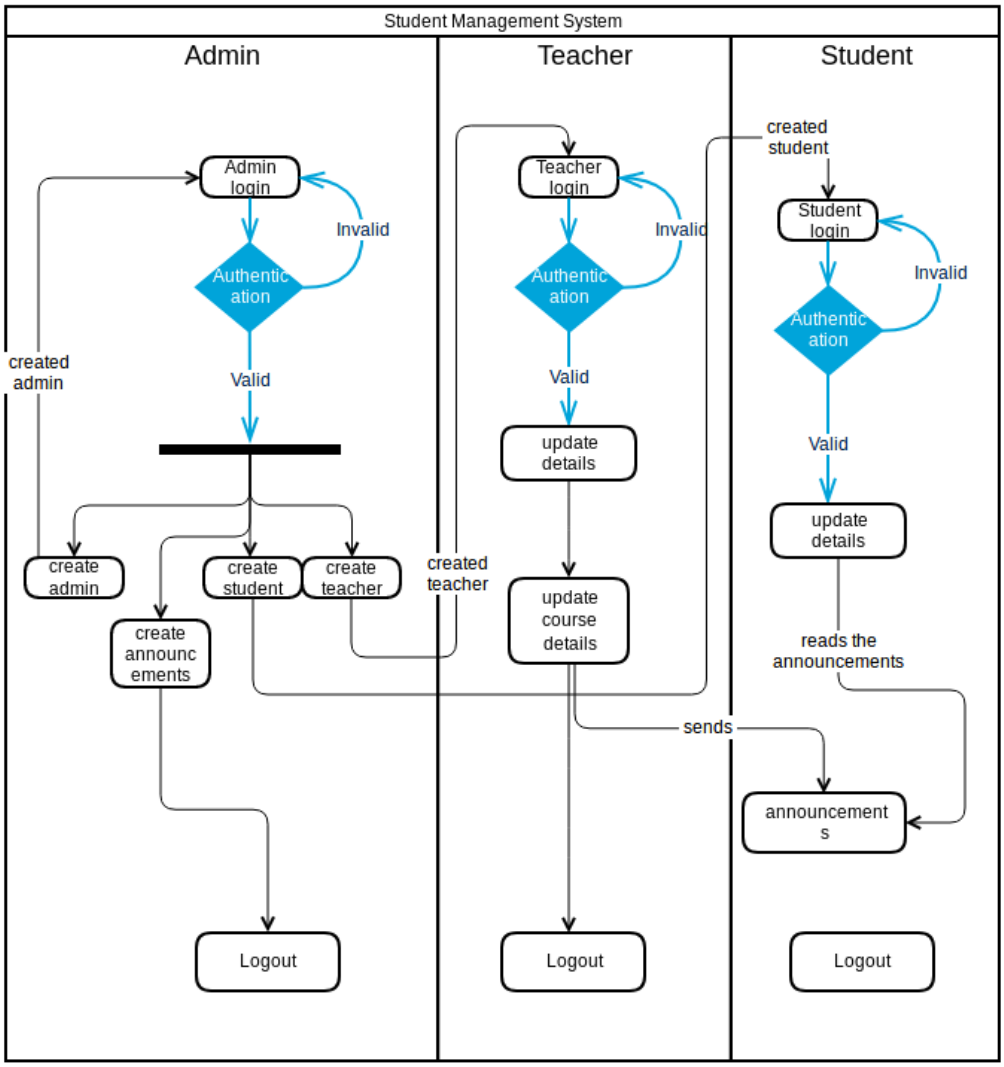
For Teacher:


Teacher login
Invalid
Authentication
Valid
Add Student
Update Admin
Create Announcements
Log out

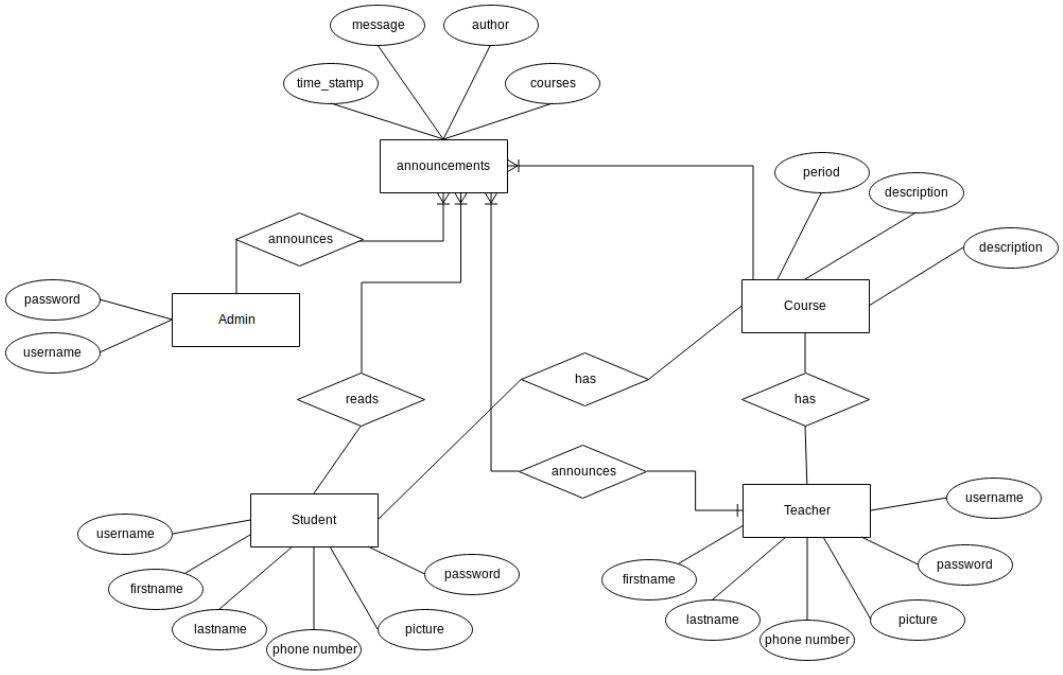
For Student:


Student login
Invalid
Authentication
Valid
Update details
Log out

SWIMLANE DIAGRAM:



ER DIAGRAM:



Appendix C: To Be Determined List

N/A