

---

# **SOFTWARE REQUIREMENTS SPECIFICATION**

**for**

**Automatic Timetable Generator**

**Version 1.0 approved**

**Hex Group**

**June 4, 2019**

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Purpose . . . . .	5
1.2	Document Conventions and Terminology . . . . .	5
1.3	Intended Audience and Reading Suggestions . . . . .	5
1.4	Project Scope . . . . .	6
1.5	References . . . . .	7
<b>2</b>	<b>Overall Description</b>	<b>8</b>
2.1	Product Perspective . . . . .	8
2.2	Product Functions . . . . .	8
2.3	User Classes and Characteristics . . . . .	9
2.4	Operating Environment . . . . .	10
2.5	Design and Implementation Constraints . . . . .	10
2.5.1	Programming / Scripting Language . . . . .	10
2.5.2	Web development tool . . . . .	11
2.5.3	Database and Technology . . . . .	11
2.5.4	System Development . . . . .	11
2.5.5	Database Development . . . . .	11
2.5.6	Testing . . . . .	12
2.5.7	Unit Testing . . . . .	12
2.5.8	Link or Integration Testing . . . . .	12
<b>3</b>	<b>External Interface Requirements</b>	<b>13</b>
3.1	User Interfaces . . . . .	13
3.2	Software Interfaces . . . . .	13
3.3	Communications Interfaces . . . . .	13
<b>4</b>	<b>System Features</b>	<b>14</b>
4.0.1	Work Flow modeling . . . . .	17
4.0.2	Description and Priority . . . . .	19
4.0.3	Stimulus/Response Sequences . . . . .	19
4.0.4	Functional Requirements . . . . .	19
<b>5</b>	<b>Other Nonfunctional Requirements</b>	<b>21</b>
5.1	Performance Requirements . . . . .	21
5.2	Safety Requirements . . . . .	21
5.3	Security Requirements . . . . .	21
5.4	Software Quality Attributes . . . . .	21

5.5	Business Rules . . . . .	22
<b>6</b>	<b>Other Requirements</b>	<b>23</b>
6.1	Appendix A: Glossary . . . . .	23
6.2	Appendix B: Analysis Models . . . . .	23

## Revision History

Date	Author	Reason For Changes	Version
18-03-2019	Kumbong Hermann N	First Draft	1.0
5-05-2019	Akola Mbey Denis	Second Draft	1.1

# 1 Introduction

## 1.1 Purpose

The aim of this document is to delineate the software requirements specification for an automatic timetable generation software. The initial version of the software will be aimed at scheduling courses at the university level and will serve as a course completion requirement for the COE 356 Software engineering course at KNUST. Future versions(not described in this document) will generalize the scheduling problem to cater for scheduling in non academic environments like hospitals, jobs etc.

## 1.2 Document Conventions and Terminology

This document follows MLA Format. Bold-faced text has been used to emphasize section and sub-section headings. Highlighting is to point out words in the glossary and italicized text is used to label and recognize diagrams. In addition the following jargon is used throughout the text:

Term	Meaning
GUI	The graphical user interface is a form of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, instead of text-based user interfaces, typed command labels or text navigation.

## 1.3 Intended Audience and Reading Suggestions

The primary audience for this document consists of all group members of Hex(project manager, developers, section heads and analysts) together with the project supervisor. The information provided could also be of benefit to product users and the prospective users on whose suggestions this document heavily relies. The document includes but is not limited to: an overall description of the the project, a discussion on the interface requirements, an analysis of system features and a description of other non functional requirements of the system. The document follows the sequence just described and is meant to be followed in that order. However, a busy reader may omit the last section on non functional requirements. The section on external interface requirements is directed primarily at the front end department while the overall description of the project will most suit the needs of a prospective client or user. It is the duty of the project manager to peruse this document and to enforce its usage and distribution of tasks described.

## 1.4 Project Scope

The current goal of this project is to develop a software for automatically scheduling courses in the university milieu. Currently most universities use a manual or a semi manual system for generating timetables. Most software available works for high schools and primary schools and is not easily adapted to the university setting. The process which can last for even weeks necessitates a lot of effort in resolving clashes and adjusting the timetable to meet specific institutional needs or those of lectures and students. Our objectives in designing our own system to solve this problem are as follows:

- Since at this moment we are unable to determine with certainty if the process can be completely automated, our goal is to automate the process of generating a timetable as much as possible providing the user of the system with little manual work to complete the timetable.
- To reduce the effort and time that are currently expended in constructing a timetable by atleast 70 %.
- To generate schedules that can
- To design a system that can be customized to meet the specific scheduling needs of any institution.
- To develop a simple and intuitive user oriented system that requires as little user training as possible.
- To provide an solution for maintaining, publishing and sharing timetables to users.

In order to achieve this,the software system must cover the following features and functions: An **administrative** section which includes the following:

- Manage student profiles
- manage lecturer's profiles
- Manage the username,password,and change password.
- Manage the add,or drop class
- Manage the add,edit and delete class
- Creation of Master Timetable
- Accept changes to made by lecturers and students.

A **lecturer's** section which includes the following:

- View and print their own timetable
- View and print the master timetable for one semester

- Query on the class availability
- Booking the class
- Creation of the lecturer's timetable

A **Student** Section which includes the following:

- View and print the timetable for a semester
- Change password
- Creation of student's timetable
- Request for change in timetable schedules

## 1.5 References

1. IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

## 2 Overall Description

### 2.1 Product Perspective

The Automatic Timetable Generation System is a closed source system comprising of 3 main components. The three components are as follows:

The Desktop platform is a stand alone program, which is a component of a larger timetable management system. The desktop component allows the user(administrator) to create and maintain a new timetable or to maintain a timetable already generated using the web component. This component is synchronized with the web app component which we describe next. The web app component provides a similar role to the desktop component. In addition to this it provides the means for managing user accounts. The mobile app component is meant for the end users of the timetable. (students, lecturers or other stakeholders). It provides them with a means of accessing their schedules and keeps them aware of any updates. It also gives them the opportunity to request any changes in their schedules from the administrator.

### 2.2 Product Functions

The timetable management system alongside its subsystems shall perform the following features:

- Provide the administrator with a means of creating a timetable template that satisfies the particular requirements(constraints) of his/her institution.
- Provide the administrator with a feature to automatically generate a timetable based on constraints they impose.
- Allow the administrator to apply manual changes to the timetable.
- **When applying manual changes to the timetable, provide the administrator with real time alerts and information when constraints are not met.** (This feature is prime in importance)
- Provide the administrator on hints or information on available resources or solutions to resolving clashes.
- End users (students and lecturers) should have a module to provide the administrator with information required to generate schedules for them.



- End users (students and lecturers) should have a module to request changes in their schedules from the administrator.
- Provide the administrator or other stakeholders with an easy interface to import their data or to enter new data required for timetable generation and management.
- End users should have a module that informs them of their timetable and any changes that have been made to it. (This could be through the mobile or the web app component)

## 2.3 User Classes and Characteristics

The users of the system can be broadly classified into two major categories. Users can fall into only one category only or in both categories. However a user can only have one type of account. In a case where a user can be in anyone category, they have to create an account for each new category. The categories are as follows:

### 1. **Administrators**

Administrators consist of anyone who are involved in the timetable generation process. More succinctly, administrators have the following privileges and roles:

- Can initiate the process of creating a new timetable.
- Has access to and can modify any timetable generated.
- Publishes the timetable and is responsible for responding to user requests and performing updates on the timetable.
- Fills the database with relevant information about resources such as (courses, lecturers, class size)
- Creates the atomic timetable template (for example is the timetable from monday to saturday or just weekdays, the number of working hours and all related information.)
- Decides on and implements scheduling and constraints and enforces that they are strictly followed.
- Assigns priorities to different users, tasks, resources.

### 2. **End Users**

End users are the consumers of the timetable schedule. For instance a lecturer could have a schedule, a student may have a schedule, a particular class group have a schedule and so on. (Due to elective courses a student's schedule is not necessarily the same as that of their class. The discrepancy increases in a liberal art system). The end users have the following roles and privileges with respect to the system:

- Can view their personal schedules or those of any group they belong to.
- Can request changes to their schedules or that of any group they belong to (it is left to the administrator to decide whether or not to implement such changes.)

## 2.4 Operating Environment

Our system is cross platform and does not dep on the hardware or architecture of the host system The web component is browser independent. In particular we support the particular operating systems.

### 1. Mobile

- a) Android OS
- b) ios

### 2. Desktop

- a) Windows 7 and above
- b) Mac
- c) Linux
- d) Solaris

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

## 2.5 Design and Implementation Constraints

**Hardware requirements** In order to develop a good application program, it is very important to choose the correct hardware, software and technology. Below are some explanations of the hardware, software and technology chosen as development tools for the Timetable Management System

- A processor of good processing speed  $2.5MHz$  or more
- 128MB DDR RAM ,256 MB is recommended
- 10GB hard-disk space or higher
- 56 Kbps Modem
- Keyboard and mouse

### 2.5.1 Programming / Scripting Language

- **Flask,a Python Framework** Python was created in 1991 by Guido Van Rossum Flask is web development framework based on Python and it can be interspersed within Hypertext Markup Language (HTML), which makes developing dynamic websites more accessible. Python Flask was selected to develop the Timetable

Management System because it is a server side, cross-platform technology. Server-side actually refers to the fact that everything Python Flask does occurs on the server instead of the client's site. Its cross-platform means that Python Flask runs on most operating systems, including Windows, UNIX, and Macintosh. Besides that, when it comes to developing dynamic websites, Python Flask is better, faster and easier to learn than the alternatives. Of course, the main reason for Python Flask being chosen to develop the website is it comes at no cost. It is open sourced.

### 2.5.2 Web development tool

- **Visual Studio Code, Atom, Sublime Text** Visual Studio Code, Atom and Sublime Text were chosen as the text editors to develop the website as well since the researcher is more familiar with using the text editors to develop a websites

### 2.5.3 Database and Technology

- **MySQL** MySQL is a database management system (DBMS) for relational databases (therefore, MySQL is an RDBMS), a database being a collection of interrelated data, be it text, numbers, or binary files, that are stored and kept organized by the DBMS MySQL was selected to develop the database for this web based system because like Python Flask, MySQL offers excellent performance, portability and reliability, with moderate learning curve at little to no cost because MySQL is the world's most popular open source database. Besides that, another reason for it being chosen is Python Flask has good support for MySQL.

### 2.5.4 System Development

The objective of the development phase is to convert the deliverables of the design phase into a complete system. Most activities in the development phase addresses the computer programs that make up the system, but this phase also puts in place the hardware, software, and communications environment for the system and other important elements of the overall system. The activities in the development phase translate the system design produced in the design phase into a working system. The development phase includes activities for developing the system, testing the system, and to ensure the system functions satisfy the functional process requirements. At the end of this phase, the system will be ready for the activities of the testing phase. The Timetable Management System web base was developed after the system analysis and system design phases. After gathering data from the system analysis stage and designing the web base, the development is divided into three main sessions which a

### 2.5.5 Database Development

The Timetable Management System used a relational database in its database implementation because it can support multiple tables that store each item only once, thus

significantly reducing storage place. The database was created and using MySQL, which is a GUI Client that works alongside MySQL Database server.

### **2.5.6 Testing**

The objective of this testing phase is to prove that the developed system (Timetable Management System) satisfies the requirements defined earlier. Several types of tests will be conducted in this phase. Testing is an important phase of system development because it can ensure the system matches the specifications. Besides that, testing also ensures that the system functions in the correct and proper manner with the minimum amount of errors.

### **2.5.7 Unit Testing**

Unit testing reveals syntax and semantic errors from the smallest programming unit. In this thesis, unit testing is used to test each individual web page. Errors that are found in a particular page of the website are thoroughly debugged and removed before starting to develop another web page. Due to the dynamic nature of testing, there is no proper testing documentation created.

### **2.5.8 Link or Integration Testing**

When each webpage of a particular Section in the Timetable Management System passed the unit testing, integration test was carried out to ensure that pages are linked in the correct flow and integrate properly into the entire website. Integration testing was mainly conducted in “Administrator Module” Section “Lecturer Module” and “Student Module” Section. All the buttons, hyperlinks and navigation bars were tested

## 3 External Interface Requirements

### 3.1 User Interfaces

The product presents to the user a friendly user interface. The GUI provides fields for the user to enter the data He/she wishes to schedule . In our case,the examinations officer can enter data comprising the classrooms,courses,and the names of the lecturers.

### 3.2 Software Interfaces

Developing end

- Python 3.6 -Python is fast, secure, and reliable.
- MySQL server -Database connectivity and management

**Client End**

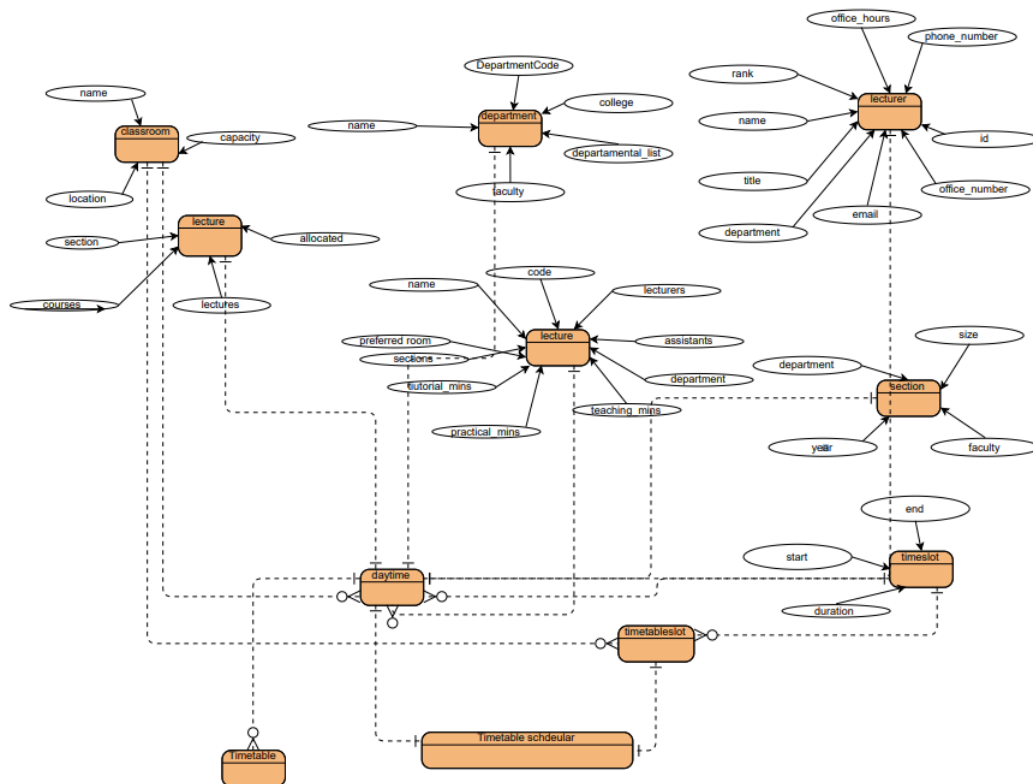
- OS -Window 7 or higher -Very friendly and common OS
- MySQL server -Database connectivity.

### 3.3 Communications Interfaces

- NIC (Network Interface Card) - It is a computer hardware component that allows a computer to connect to a network. NICs may be used for both wired and wireless connections.
- CAT 5 network cable- for high signal integrity
- TCP/IP protocol- Internet service provider to access and share information over the Internet
- Ethernet Communications Interface- Ethernet is a frame-based computer network technology for local area networks (LANs)
- Ubiquitous, easy to set up and easy to use. Low cost and high data transmission rates.

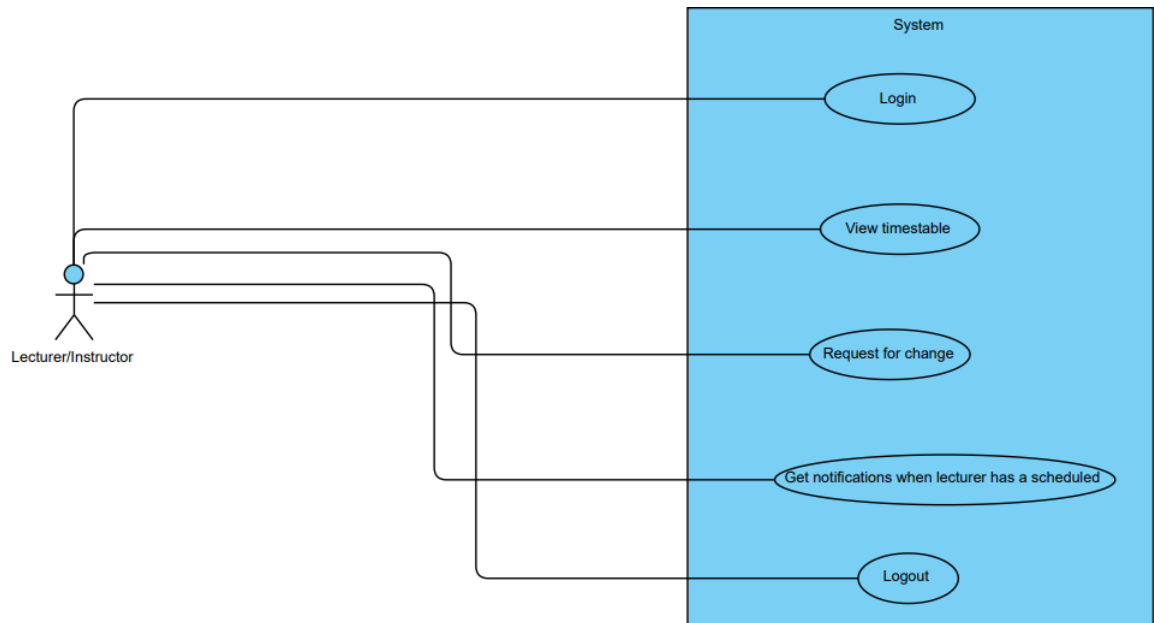
## 4 System Features

The Software product is made up of a number of UML's ,use case diagrams and a number of object classes. UML use case diagrams are used to describe the main processes and functionality of the Timetable Management System. The purpose of having use case diagram is to identify the scope of the system. Three use case diagrams have been created for timetable system: one for the lecturer, students and the administrator



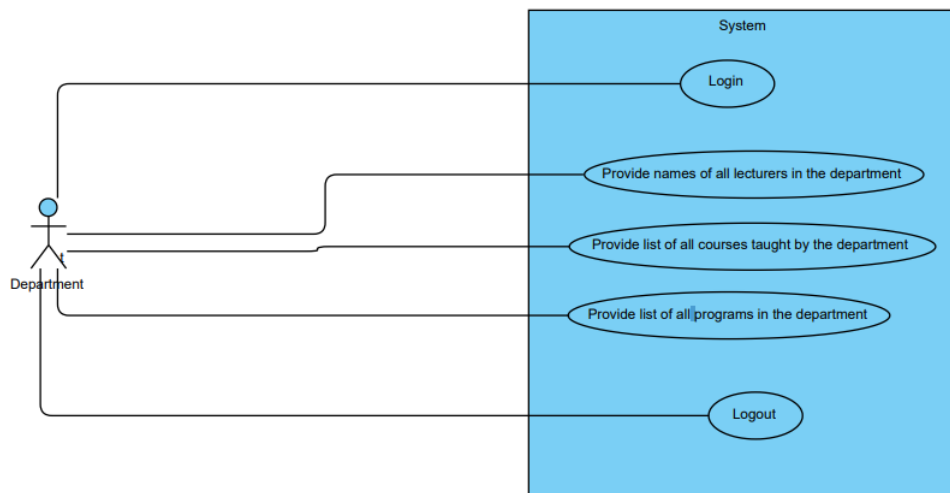
ER

diagram for the System Database Design

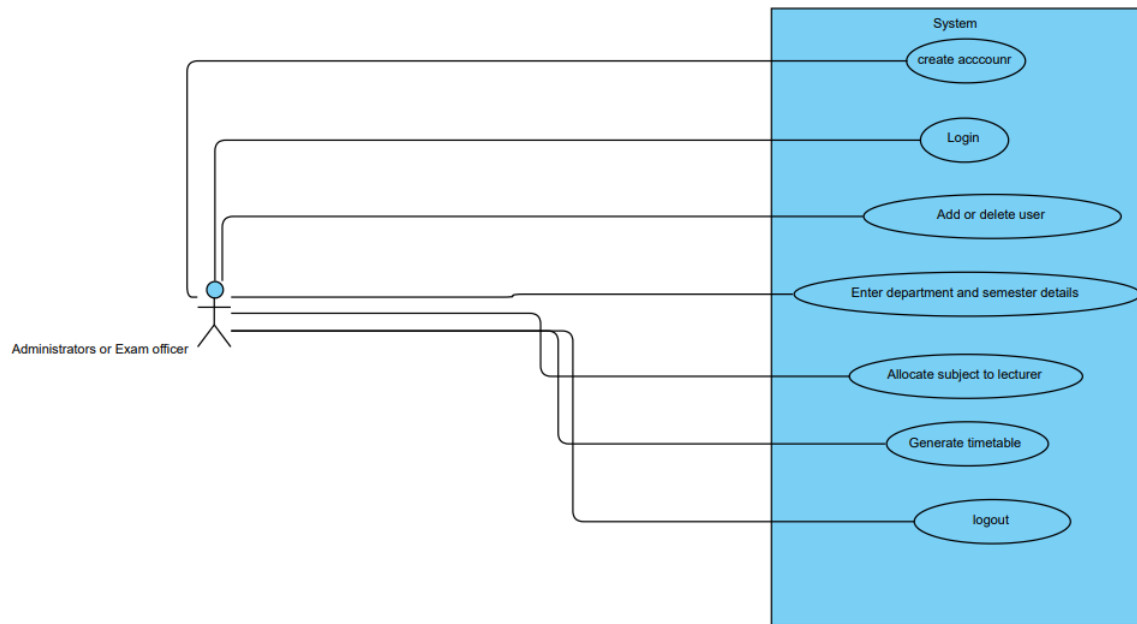


Lecturer use case diagram

Lecturer can log-in to Timetable Management System, using their usernames and passwords. System displays the main menu if log-in is successful. If the username and password are not accepted, system displays a message indicating that the username or password is invalid. Once the lecturer logs in, he or she can perform the processes (use cases) like view class, view timetable, view master timetable, inquiry class available, class booking and change password.

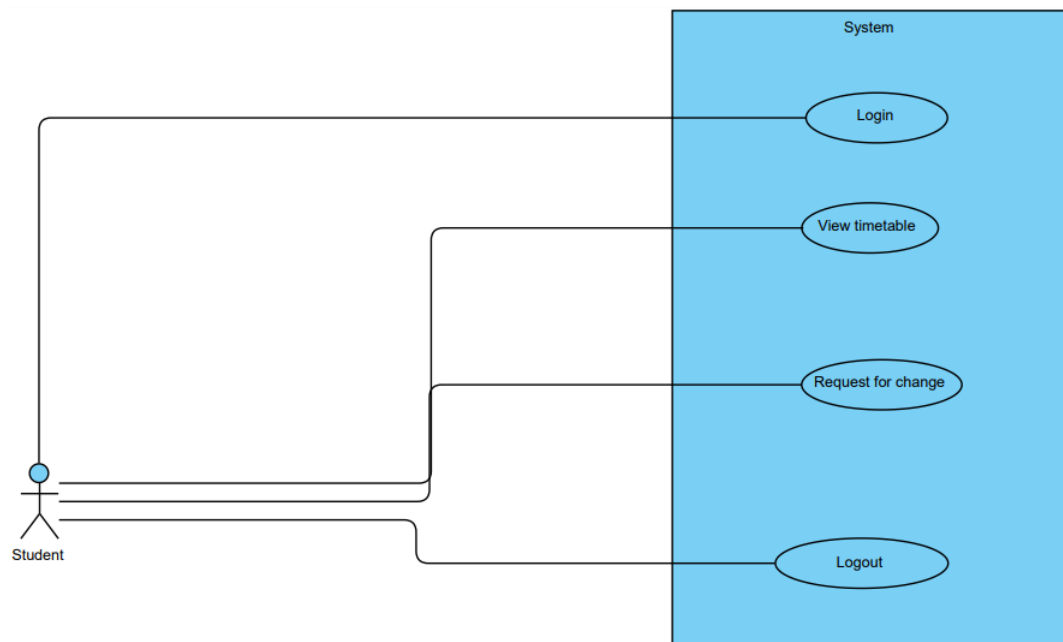


Department use case diagram



Administrator/Exam officer use case diagram

The administrator does not need to register, as his or her username and password is fixed in the database. The administrator needs to log in to Timetable Management System in order to manage the system. Besides logging in, the seven main use cases for the administrator are student registration, lecturer registration, add class, edit class, delete class, add courses , edit and delete course.



Student use case diagram



Student can log-in to Timetable Management System, using their usernames and passwords. System displays the main menu if log-in is successful. If the username and password are not accepted, system displays a message indicating that the username or password is invalid. Once the student logs in, he or she can perform the processes (use cases) like add subjects, drop subject, view timetable, view registration slip and can change password

Table 4.1: Lecturer use case diagram description.

Use case	Description
Login	It allows the lecture to have access to the system by logging in with a username and password.
View	It allows the lecturer to view his /her lecture schedules on the timetable.
Request for change	It allows the lecturer to request for change in lecture times where necessary.
Notification	This facility notifies the lecturer anytime he/she has a lecture.
Logout	It allows a lecturer to logout of the system.

Table 4.2: Administrator use case diagram description.

Use case	Description
Create Account	This facility allows the administrator to create new users.
Login	It allows the administrator must login with username and password.
Add/delete	It allows the administrator to add or delete users (i.e lecturers).
Enter department and semester details	It allows the admin to enter a department details (Name,course, semester).
Generate timetable	This feature allows the administrator to generate the timetable.
Allocate courses	It allows the administrator to assign courses to lecturers.
Logout	It allows a administrator to logout of the system.

Table 4.3: Department use case diagram description.

Use case	Description
Login	This facility allows the department to login with username and password.
Provisions of Courses,Programs and Lecturers in department	It allows the department to add the list of courses, programs and lecturers.
Logout	It allows a administrator to logout of the system.

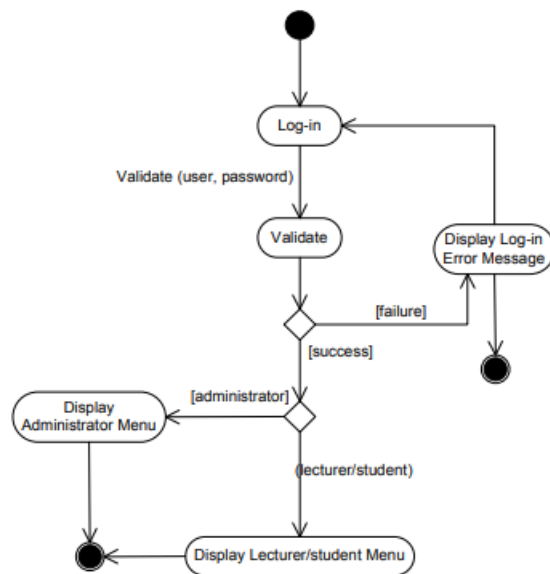
#### 4.0.1 Work Flow modeling

Activity diagrams are used here to model the flow between the different components. An activity diagram is needed because the researcher wants to model the workflow of a

Table 4.4: Student use case diagram description.

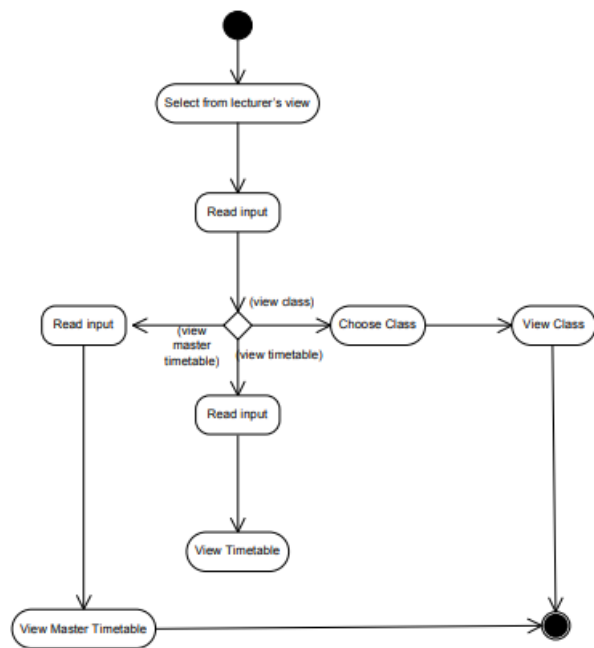
Use case	Description
Login	Student cant login with username and password generated by the system
View	It allows the student to to view the lectures timetable
Request for change	It allows the student to to request for a change in course schedules
Logout	It allows a student to logout of the system.

use case, and it can show the paths within the use case as well as other use cases. With activity diagrams, the researcher will be able to illustrate where functionality exists in the system and how the functionality coordinates with the functionality of other pieces of the system. The researcher has developed ten activity diagrams for this system. A brief description will be given to each of the activity diagram in the following pages.



Login activity diagram

The figure above shows the activity diagram for Log-in. First, the lecturers, students and administrator need to log in using the username and password that was created during registration. The system will validate the username and password. If the password or username is invalid, an error message will be displayed and the lecturer or student or administrator can try to log in again. If log in is successful, the system will identify the user as a lecturer, student or an administrator. If the person logs in as administrator, the administrator's menu page will be displayed; else the lecturer or student menu's page will be displayed.



Activity diagram for View Class,  
Timetable or Master Timetable

The figure above shows the activity diagram for View class, timetable and master timetable for lecturers. The lecturers can click on the view class or timetable or master timetable menus and the specific information will be loaded. If the lecturers want to print their timetable or their master timetable they can click on the button “PRINT”

#### 4.0.2 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on relative scale from a low of 1 to a high of 9).>

#### 4.0.3 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

#### 4.0.4 Functional Requirements

1. Some of the functional requirements of our software product are:
  - Login process/authentication for the administrator, students and lecturers. The administrators can generate the timetable and make changes when new

changes arrive. The lecturers can be authenticated to view their lecture scheduled and request for change in schedules where necessary. They are also notified periodically about their lecture schedule from time to time. The students are also authenticated to have access to their class timetables but they can request for a change when they feel that their class schedules are not favorable.

- There are 5 Modules for the System
  - a) The Department details
  - b) Instructors/Lecturers details
  - c) The Time Table Allocation details
  - d) The Courses details
- Some of the mandatory constraints on the system are:
  - a) No two lectures can be taken by the same lecturer except for combined classes.
  - b) The minimum time for a lecture should be at least one hour and at most four hours.
  - c) A classroom can be occupied by only one class at a time except for combined lectures where two or more classes can together for a lecture.
  - d) A classroom cannot be assigned to a class unless the class size is less than the capacity of the classroom.
  - e) There must be no colliding lectures for a particular class. No two lectures should be taken by a class unless they are elective courses where students choose to take one course or the other.

## 5 Other Nonfunctional Requirements

### 5.1 Performance Requirements

- Response time-The system will give responses within 1 second after checking the patient information and other information.
- User interface- User interface screen will response within 5 seconds.
- conformity –The system must conform to the Microsoft accessibility

### 5.2 Safety Requirements

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

### 5.3 Security Requirements

All the administrative and data entry operators have unique logins so system can understand who is login in to system right now no intruders allowed except system administrative nobody cannot change record and valuable data.

### 5.4 Software Quality Attributes

- **Availability** : The system shall be available all the time.
- **Correctness**: A bug free software which fulfill the correct need/requirements of the client
- **Maintainability**: The ability to maintain ,modify information and update fix problems of the system
- **Usability**: Software can be used again and again without distortion.
- **Accessibility**: Administrator and many other users can access the system but the access level is controlled for each user according to their work scope.

- **Accuracy:** The reliability on the information/output. Can depend/be sure of the outcome
- **Stability :** The system outcome/output won't change time to time. Same output will be given always for a given input.

## 5.5 Business Rules

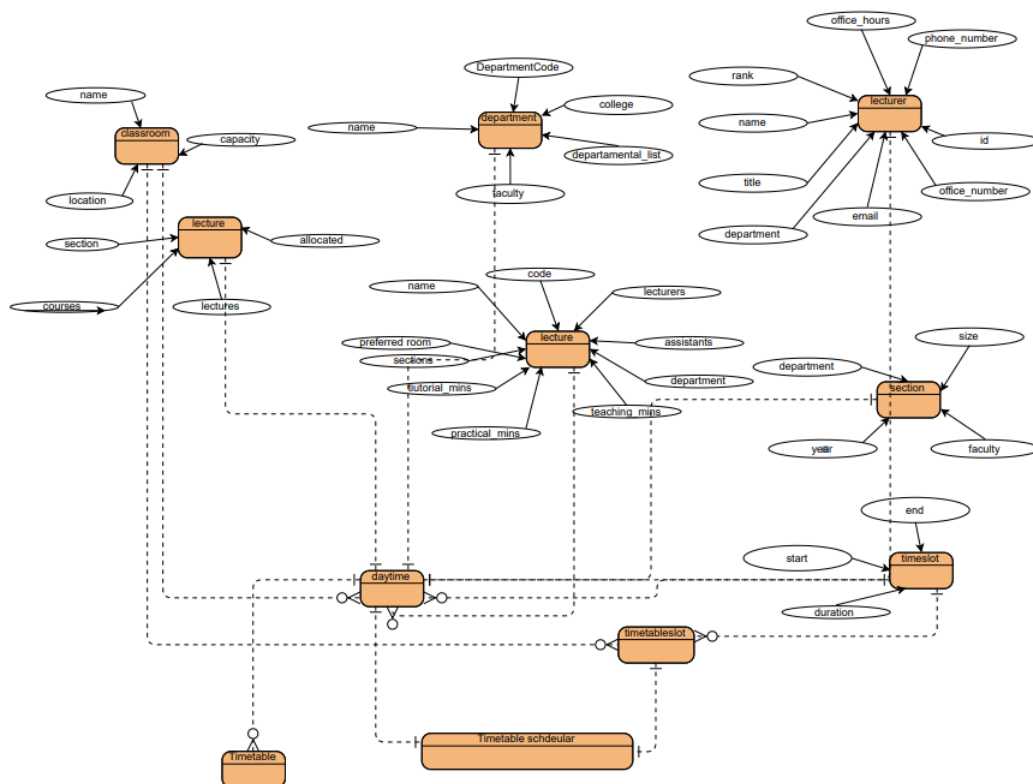
- Want take the responsibility of failures due to hardware malfunctioning.
- Warranty period of maintaining the software would be one year.
- Additional payments will be analysed and charged for further maintenance
- If any error occur due to a user's improper use. Warranty will not be allocated to it.
- No money back returns for the software.
- Trust bond placement should be done before designing and coding. An advance or an agreement

## 6 Other Requirements

### 6.1 Appendix A: Glossary

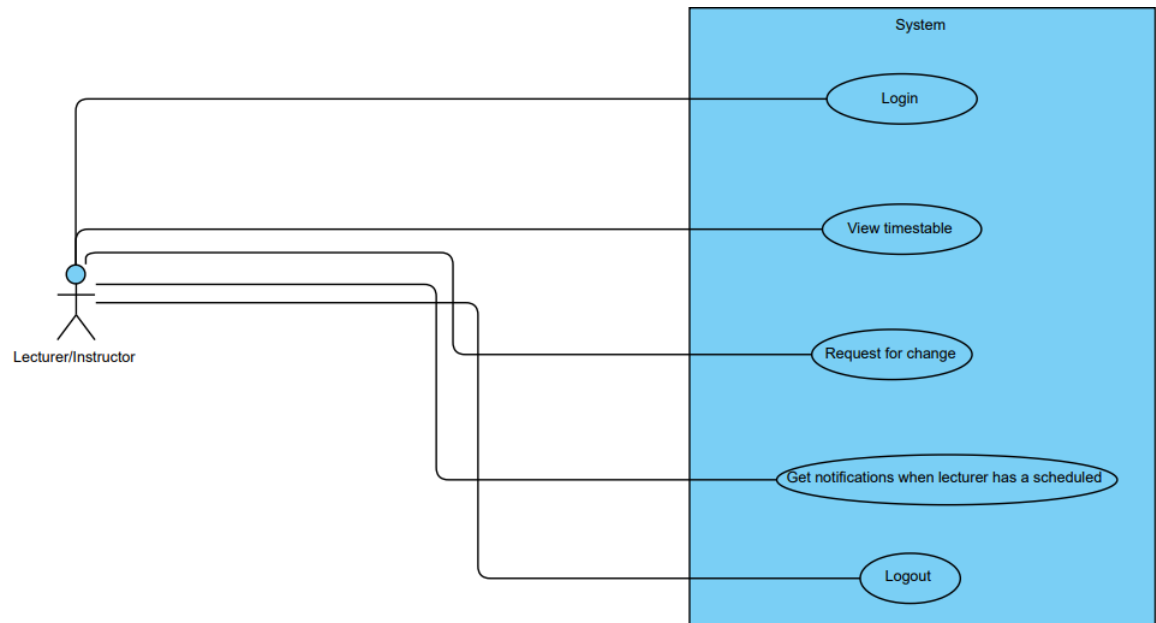
No glossary terms available at this time.

### 6.2 Appendix B: Analysis Models

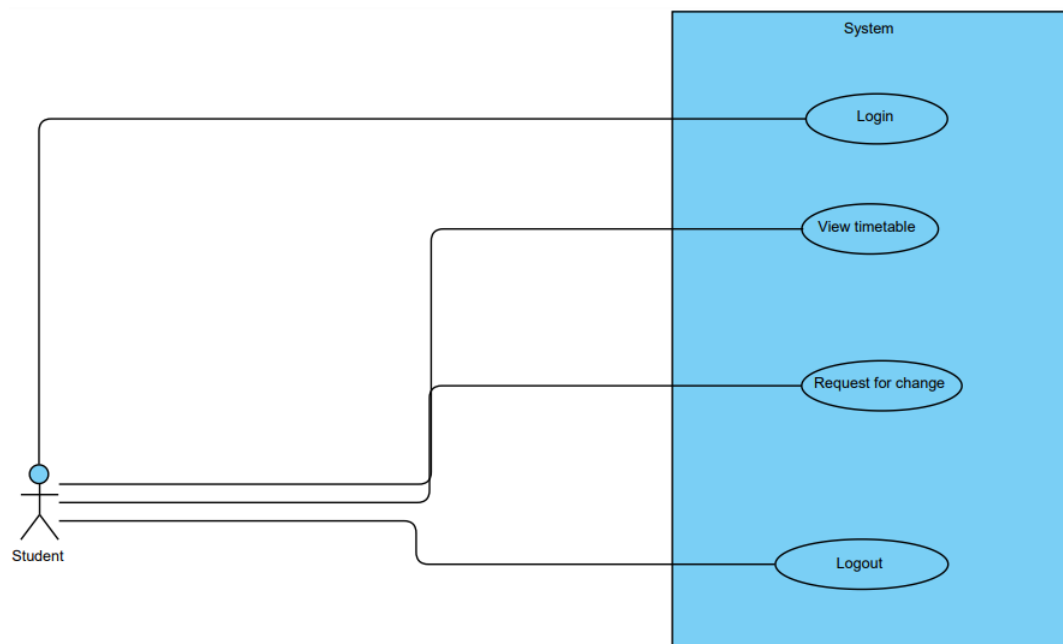


ER

diagram for the System Database Design

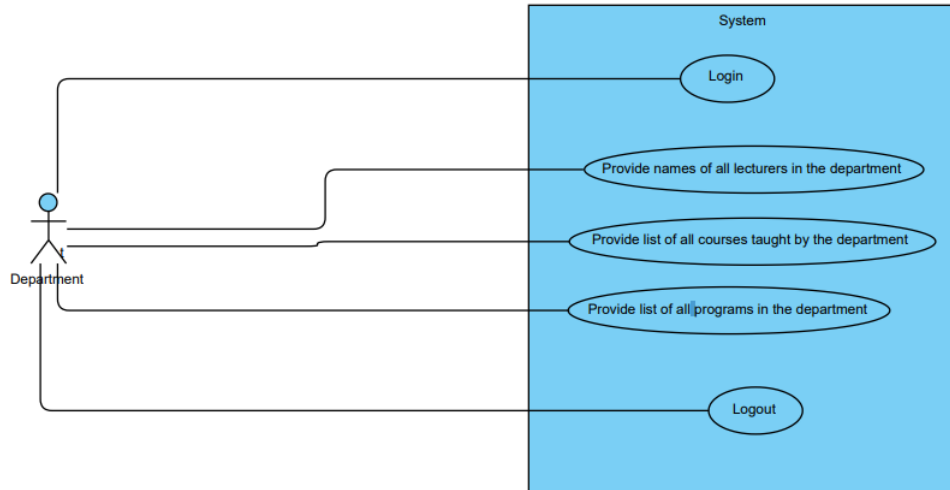


Lecturer use case diagram

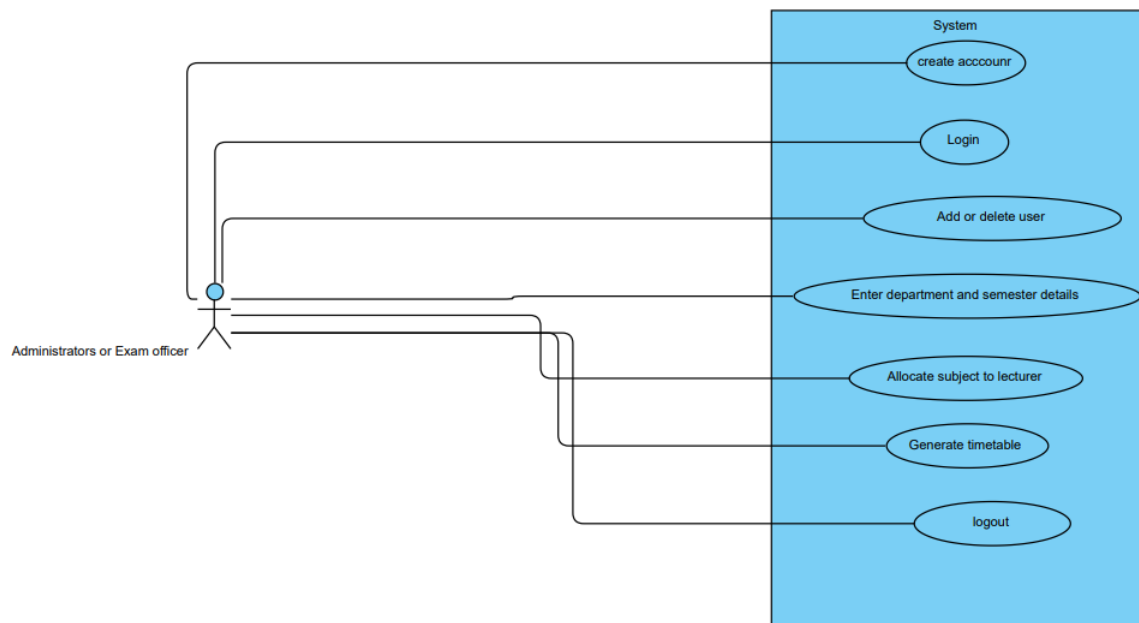


Student use case diagram





Department use case diagram



Administrator/Exam officer use case diagram