Software Requirement Specification

Big U Management System

Team members: - Hendro Limanto

- Habibie Wikayan

- Fredric Sidik

Table of Contents

[1. Purpose 1](#_Toc330769073)

[1.1. Introduction 1](#_Toc330769074)

[1.2. Scope 1](#_Toc330769075)

[1.3. Document Overview 1](#_Toc330769076)

[2. Overall Description 2](#_Toc330769077)

[2.1. Functional Requirement 2](#_Toc330769078)

[2.1.1. Enroll Use Case 4](#_Toc330769079)

[2.1.2. Modify Address Use Case 4](#_Toc330769080)

[2.1.3. View Funding & Fee Status Use Case 5](#_Toc330769081)

[2.1.4. View Advisor Contact Detail Use Case 6](#_Toc330769082)

[2.1.5. Enter Student Marks Use Case 6](#_Toc330769083)

[2.1.6. View Enrolled Student Use Case 7](#_Toc330769084)

[2.1.7. Write subject guide Use Case 7](#_Toc330769085)

[2.1.8. Approve Subject Guide Use Case 8](#_Toc330769086)

[2.1.9. Edit Subject Guide Use Case 8](#_Toc330769087)

[2.1.10. Write Subject Guide Use Case 9](#_Toc330769088)

[2.1.11. Write Course Structure Use Case 10](#_Toc330769089)

[2.1.12. Edit Course Structure Use Case 10](#_Toc330769090)

[2.1.13. View List of Student Use Case 11](#_Toc330769091)

[2.1.14. View Student Result Use Case 11](#_Toc330769092)

[2.1.15. View Student Progress Use Case 12](#_Toc330769093)

[2.2. Non-Functional Requirement 12](#_Toc330769094)

# Purpose

## Introduction

This Software Requirement Specification provides description of all functions and non-functional requirements of the Big U Management System. This document will help to identify if the proposed system meets all the requirements. The expected audience of this document is Mr. Johnny Derp and all the staff that will use the system.

## Scope

The Big U Management System is created to help all the process of university management easier. The system will help the students, lecturers, student advisors, coordinators, administrators, and timetablers to do all their works that needs to be done manually before. All the works that can be done through the system will be shown in detail later on in this document. All the data of the system will be stored inside a centralized database server.

## Document Overview

In this document, we provide the diagrams along with the descriptions of the basic design on the functionalities that this system is going to provide for the users. These functionalities can be seen on the use case diagrams that are provided below in this report. The interaction between the users and system itself will also be illustrated more clearly with the help of sequence diagrams. Furthermore, the non-functional features that this system going to have will be described later on in the last few sections of this document.

# Overall Description

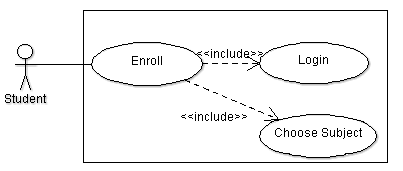
## Functional Requirement

Big U Management system is a system that will help Big U university staffs to properly perform the management process on everything which are related to students’ needs in their education side along with the management of their payments and invoices. Based on the system proposal, our team has developed the system to have functionalities that are required by the university. Here is the list of functionalities that this system is expected to have:

* Students are able to login
* Students are able to modify their address
* Students are able to view their marks and time tables
* Students are able to view the contact details of their advisor
* Students can enroll
* Student can see their funding and fee status
* Issue student cards with photo for the students
* Determine the eligibility of students to graduate with honors or distinction
* Exclude students
* Generate list of students who are deemed at risk
* Lecturers can get a list of enrolled students
* Lecturers can enter marks for enrolled students at the end of the semester
* Lecturers can write a subject guide for their subject
* Student advisors will be automatically assigned to a student according to the student home department
* Student advisors can view their list of student
* Student advisors can view the progress of their student
* Student advisors can view the result of their student
* Student advisors can make notes on meetings with their student
* Student advisors can upload any document related to their student
* Student advisors can override normal course enrollment rules in special case
* Advisor of a student can be changed to other advisor if the advisor leaves or in special circumstances
* Coordinators can view their list of student
* Coordinators can view the progress of their student
* Coordinators can view the result of their student
* Coordinators can make notes on meetings with their student
* Coordinators can upload any document related to their student
* Coordinators can write course structures
* Coordinators can edit course structures
* Coordinators can write subject guide
* Coordinators can edit subject guide
* Coordinators can approve subject guide
* Coordinators can override normal course enrollment rule
* Keep track of local student fees
* Generate report for the tax department for each semester fee of the local student
* Send invoice of each semester fee to the international student who has no scholarship
* For international student who has scholarship, the invoice will be send to the body administering the scholarship
* Administrator can see the students funding and fee status
* Administrators can generate reports that show how much the university is earning and from what sources
* A student who has not paid their fees by the cutoff date will be unenroled each semester
* A student who paid their fees by the cutoff date will be unenroled each semester
* Create timetable for each student
* Store applicants for courses
* Allow the selected officers to view application, rank and select applicants.
* Send a letter to selected applicants.
* Allow the scanning and storage of application form and accompanying document.
* Create student record.
* Timetabler can enter the number of staff allocated to each subject
* Timetabler can enter the worktime of the staff
* All data can be saved
* Displays different interface for different users (staffs and students)

## Enroll Use Case

**Diagram:**

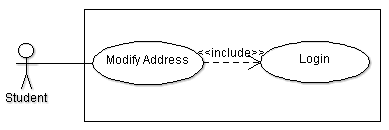


**Description:**

The student will be able to enroll once he/she has login to the system. Once the student’s identity has been verified and enroll functionality is triggered, he/she will then be required to choose the subject(s) to be enrolled into.

## Modify Address Use Case

**Diagram:**

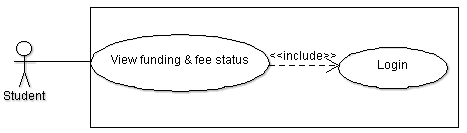


**Description:**

The student will be able to modify his/her address once he/she has login to the system. Once the student’s identity has been verified and modify address functionality is triggered, he/she will then be able to enter the details of the new address.

## View Funding & Fee Status Use Case

**Diagram:**

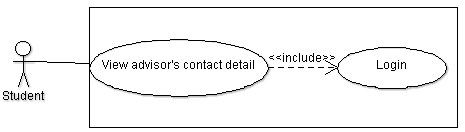


**Description:**

The student will be able to view the funding and fee status once he/she has login to the system. Once the student’s identity has been verified and view funding and fee status functionality is triggered, he/she will then be able to view his/her own funding details

## View Advisor Contact Detail Use Case

**Diagram:**

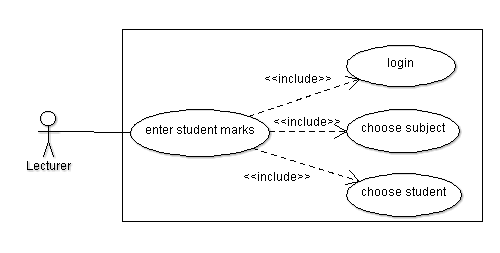


**Description:**

The student will be able to view the contact detail of the advisor who has been assigned to that him/her once he/she has login to the system. Once the student’s identity has been verified and view advisor contact detail functionality is triggered, he/she will then be able to view his/her advisor’s contact details.

## Enter Student Marks Use Case

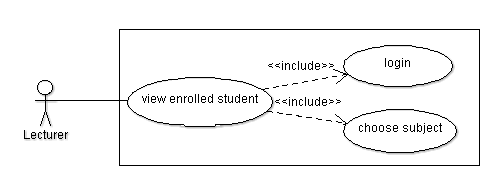
**Diagram:**



**Description:**

The lecturer must login to the system first before he/she can enter the student marks. When the lecturer wants to enter the student marks, he/she needs to choose the subject that he/she wanted to give marks for. After that, the lecturer must choose the student who he/she wants to give the marks to.

## View Enrolled Student Use Case

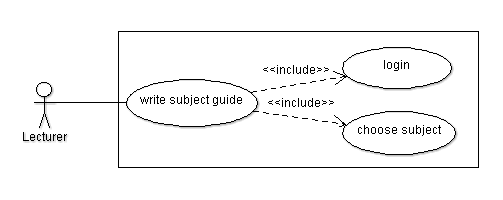
**Diagram:**

**Description:**

The lecturer must login to the system first before he/she can ciew the enrolled student. When the lecturer wants to view the enrolled students, he/she needs to choose the subject he/she wants to see the list of students enroll in that particular subject.

## Write subject guide Use Case

**Diagram:**

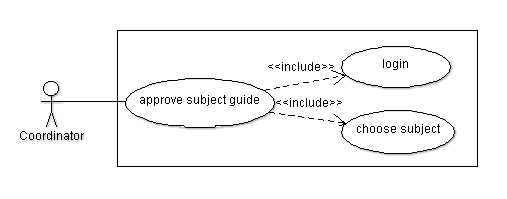


**Description:**

The lecturer must login to the system first before he/she can write a subject guide. When the lecturer wanted to write a subject guide, he/she needs to choose which subject he/she wants to write its subject guide

## Approve Subject Guide Use Case

**Diagram:**

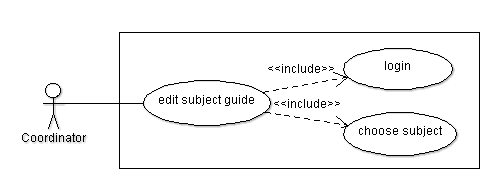


**Description:**

The coordinator must login to the system first before he/she can approve a subject guide. When the coordinator wanted to approve a subject guide, he/she needs to choose which subject he/she wants to approve its subject guide.

## Edit Subject Guide Use Case

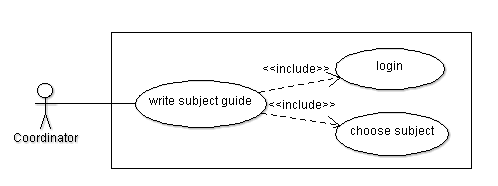
**Diagram:**



**Description:**

The coordinator must login to the system first before he/she can edit a subject guide. When the coordinator wanted to edit a subject guide, he/she needs to choose which subject he/she wants to edit its subject guide.

## Write Subject Guide Use Case

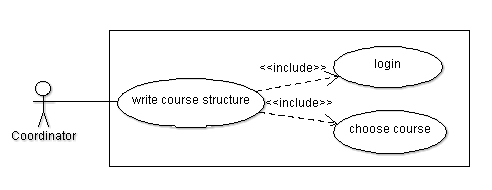
**Diagram:**

**Description:**

The coordinator must login to the system first before he/she can write a subject guide. When the coordinator wanted to write a subject guide, he/she needs to choose which subject he/she wants to write its subject guide.

## Write Course Structure Use Case

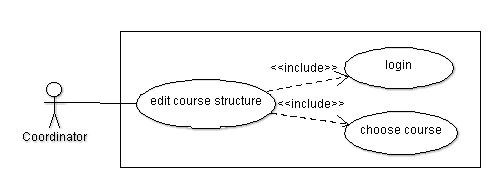
**Diagram:**



**Description:**

The coordinator must login to the system first before he/she can write a course structure. When the coordinator wanted to write a course structure, he/she needs to choose which course he/she wants to write its course structure.

## Edit Course Structure Use Case

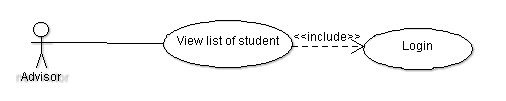
**Diagram:**

**Description:**

The coordinator must login to the system first before he/she can edit a course structure. When the coordinator wanted to edit a course structure, he/she needs to choose which course he/she wants to edit its course structure.

## View List of Student Use Case

**Diagram:**

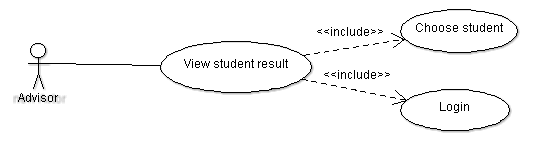


**Description:**

The student advisor will be able to view the list of student that belongs to him/her once he/she has login to the system.

## View Student Result Use Case

**Diagram:**

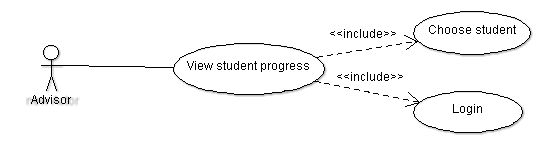


**Description:**

The student advisor will be able to view the result of the student once he/she has login to the system. The advisor needs to choose the student first before he/she can see their result. This will help the advisor for further evaluation for his/her student.

## View Student Progress Use Case

**Diagram:**



**Description:**

The student advisor is able to see the progress of his/her students. But first they need to login to the system and then pick one student which he/she would like to see the progress.

## Non-Functional Requirement

Aside from all the functionalities of the Big U management system stated above, our system also has several benefits which are not directly related to the functionalities. First of all, our system implements the login system to identify the users. Next, our system is platform independent or in other words, it is capable to work in any kind of OS background. Also, the Big U management system also has a very fast processing time, which takes less than 2 seconds for moving from one page to another page. Furthermore, as the system is developed in GUI based format, the users will just need to click the selection that they want and the page requested will be directly shown on the screen.

As an addition, the system’s footprint can be considered to be relatively small and therefore allowing computers with low storage capacity to install and use this system. To complement all those, the usage of this system is actually very simple and the interfaces are very straight forward. With this kind user friendly system, our team is very confident that the training time for using this system is expected to take less than 1 hour.