Documentation on Big U Management System

Made by: - Hendro Limanto

* Fredric Sidik
* Habibie Wikayan

**Executive Summary**

In this documentation, we are going to explain about a medium size university management system which is known as the Big U Management System. This system is ultimately aimed to help Big U University to perform its university management process, which focuses more on every activity regarding the students within the university.

The availability of this system itself is expected to bring many positive impacts for both the staffs and the students of Big U University. All the users of this system will soon be able to enjoy every single benefit that this system provides once it has been integrated to the university system and thus replacing the old and outdated version of the university management system.

With this system, students will soon be able to view every single details related to their university life without the need of coming directly to the university itself. For the staffs, a much better and fully automated system will help them to completely eliminate the hardships they experienced before when they were still using the old system.

Table of Contents

[1. Introduction 1](#_Toc328949683)

[2. Objectives 1](#_Toc328949684)

[3. Software Process Model 1](#_Toc328949685)

[3.1. Requirement analysis 1](#_Toc328949686)

[3.2. Use Case Diagram 3](#_Toc328949687)

[3.2.1. Create Project Use Case 3](#_Toc328949688)

[3.2.2. Modify Project Use Case 3](#_Toc328949689)

[3.2.3. Delete Project Use Case 4](#_Toc328949690)

[3.2.4. Add New Member Use Case 4](#_Toc328949691)

[3.2.5. Modify Member Use Case 5](#_Toc328949692)

[3.2.6. Delete Member Use Case 6](#_Toc328949693)

[3.2.7. Create Task Use Case 6](#_Toc328949694)

[3.2.8. Modify Task Use Case 7](#_Toc328949695)

[3.2.9. Delete Task Use Case 7](#_Toc328949696)

[3.2.10. Save a project 8](#_Toc328949697)

[3.2.11. Load a project 8](#_Toc328949698)

[3.2.12. View Progress Use Case 9](#_Toc328949699)

[3.2.13. Update Task Use Case 9](#_Toc328949700)

[3.2.14. Link Task Use Case 10](#_Toc328949701)

[3.2.15. Create Sub Task Use Case 11](#_Toc328949702)

[3.2.16. Assign Member Use Case 11](#_Toc328949703)

[3.2.17. Unassign Member Use Case 12](#_Toc328949704)

[3.3. Members Roles and Task Distribution 13](#_Toc328949705)

[3.4. Sequence Diagram 13](#_Toc328949706)

[3.4.1. Create Project 13](#_Toc328949707)

[3.4.2. Modify Project 14](#_Toc328949708)

[3.4.3. Delete Project 15](#_Toc328949709)

[3.4.4. Add New Member 15](#_Toc328949710)

[3.4.5. Modify Member 16](#_Toc328949711)

[3.4.6. Delete Member 17](#_Toc328949712)

[3.4.7. Create Task 17](#_Toc328949713)

[3.4.8. Modify Task 18](#_Toc328949714)

[3.4.9. Delete Task 19](#_Toc328949715)

[3.4.10. Save Project 19](#_Toc328949716)

[3.4.11. Load Project 20](#_Toc328949717)

[3.4.12. View Progress 20](#_Toc328949718)

[3.4.13. Update Task 21](#_Toc328949719)

[3.4.14. Link Task 22](#_Toc328949720)

[3.4.15. Create Sub Task 23](#_Toc328949722)

[3.4.16. Assign Member 23](#_Toc328949723)

[3.4.17. Unassign Member 24](#_Toc328949724)

[3.5. Tools and Environment 24](#_Toc328949725)

[3.6. Persistence Design 25](#_Toc328949726)

[3.7. Feasibility Studies 25](#_Toc328949727)

[3.7.1. Organizational Feasibility 25](#_Toc328949728)

[3.7.2. Resource Feasibility 26](#_Toc328949729)

[3.7.3. Schedule Feasibility 26](#_Toc328949730)

[3.8. System Scope Document 27](#_Toc328949731)

[3.9. Class Diagram 28](#_Toc328949732)

[4. Functionalities 29](#_Toc328949733)

[4.1. Iteration 1 29](#_Toc328949734)

[4.2. Iteration 2 29](#_Toc328949735)

[4.3. Iteration 3 29](#_Toc328949736)

[5. Test Plan 30](#_Toc328949737)

[6. Conclusion 30](#_Toc328949738)

# Introduction

This documentation is prepared to provide every single user and stakeholder of this new system with complete and clear details about the new system, which includes the software development methodology that we apply for developing this system along with the positive impacts that will be enjoyed by the users by using the functionalities provided by the system.

# Objectives

With the availability of clear requirements along with the successful identification on the stakeholders, our team has successfully developed Big U management system. Based upon our analysis, the stakeholders of this system are identified as follow:

* Students of Big U University
* Lecturers of Big U University
* Coordinators of Big U University
* Student advisors of Big U University
* Administrators in the Admission Office of Big U University
* Timetablers of Big U University
* Every single member of our team

These stakeholders identified above are those who are directly and indirectly affected with the creation of Big U management system.

This system itself is focusing more on helping the users of this system to perform their day to day tasks which are directly related to the students in the Big U University, while for the students, this system is aimed to help them to view the resources for their studies and some other important details via their mobile phone or computer, assuming web browser is already installed in those devices. With this system, the students will no longer need to go directly to the university just to view some trivial information. More details about the functionalities that this system provides will be explained further in greater details along with some diagrams in the next few sections.

# 3. Software Process Model

Out of those many software processes modeling available, our team has decided that Rational Unified Processing (RUP) process model will be the most appropriate for the development of Big U management system. Due to some constraints, we are only able to follow three from the original four phases of the RUP model. These three phases are inception, elaboration and construction, which exclude the last phase of RUP model, the transition phase.

The details on what we have performed and done in each of the phase will be described more in detail in the following three sub sections.

### Requirement analysis

From the collection of functional requirements provided in the project proposal, we broke it down further into total of 51 functionalities. For, the non-functional requirements, we have captured 4 main non-functional requirements.

All 51main functionalities that have to be provided within the system are as the following:

|  |  |
| --- | --- |
| No | Functionalities |
| 1 | Students are able to login |
| 2 | Students are able to modify their address |
| 3 | Students are able to view their marks and time tables |
| 4 | Students are able to view the contact details of their advisor |
| 5 | Students can enroll |
| 6 | Student can see their funding and fee status |
| 7 | Issue student cards with photo for the students |
| 8 | Determine the eligibility of students to graduate with honors or distinction |
| 9 | Exclude students |
| 10 | Generate list of students who are deemed at risk |
| 11 | Lecturers can get a list of enrolled students |
| 12 | Lecturers can enter marks for enrolled students at the end of the semester |
| 13 | Lecturers can write a subject guide for their subject |
| 14 | Student advisors will be automatically assigned to a student according to the student home department |
| 15 | Student advisors can view their list of student |
| 16 | Student advisors can view the progress of their student |
| 17 | Student advisors can view the result of their student |
| 18 | Student advisors can make notes on meetings with their student |
| 19 | Student advisors can upload any document related to their student |
| 20 | Student advisors can override normal course enrollment rules in special case |
| 21 | Advisor of a student can be changed to other advisor if the advisor leaves or in special circumstances |
| 22 | Coordinators can view their list of student |
| 23 | Coordinators can view the progress of their student |
| 24 | Coordinators can view the result of their student |
| 25 | Coordinators can make notes on meetings with their student |
| 26 | Coordinators can upload any document related to their student |
| 27 | Coordinators can write course structures |
| 28 | Coordinators can edit course structures |
| 29 | Coordinators can write subject guide |
| 30 | Coordinators can edit subject guide |
| 31 | Coordinators can approve subject guide |
| 32 | Coordinators can override normal course enrollment rule |
| 33 | Keep track of local student fees |
| 34 | Generate report for the tax department for each semester fee of the local student |
| 35 | Send invoice of each semester fee to the international student who has no scholarship |
| 36 | For international student who has scholarship, the invoice will be send to the body administering the scholarship |
| 37 | Administrator can see the students funding and fee status |
| 38 | Administrators can generate reports that show how much the university is earning and from what sources |
| 39 | A student who has not paid their fees by the cutoff date will be unenroled each semester |
| 40 | A student who paid their fees by the cutoff date will be unenroled each semester |
| 41 | Create timetable for each student |
| 42 | Store applicants for courses |
| 43 | Allow the selected officers to view application, rank and select applicants. |
| 44 | Send a letter to selected applicants. |
| 45 | Allow the scanning and storage of application form and accompanying document. |
| 46 | Create student record. |
| 47 | Timetabler can enter the number of staff allocated to each subject |
| 48 | Timetabler can enter the worktime of the staff |
| 49 | All data can be saved |
| 50 | Displays different interface for different users (staffs and students) |
| 51 | Students are able to login |

The next table below provides the information about the non-functional requirements:

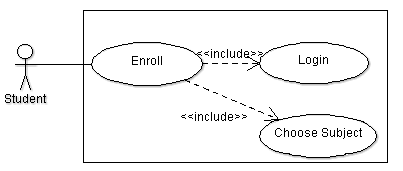
|  |  |
| --- | --- |
| No | Non-Functional Requirements |
| 1 | Login system |
| 2 | Platform independent |
| 3 | Small system footprint |
| 4 | Less than 1 hour of training time |
| 5 | Fast processing time |
| 6 | GUI based |

## Use Case Diagrams

The use case diagrams provided in this report will have some additional use cases which were not yet provided in the Software Requirement Specification. In the next few sub sections, we will provide the use cases of the functionalities which require the interaction between the users and the system.

## Enroll Use Case

**Diagram:**

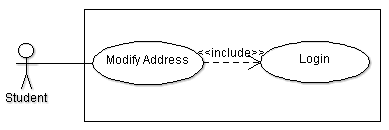


**Description:**

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:**

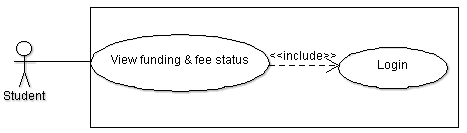


**Description:**

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:**

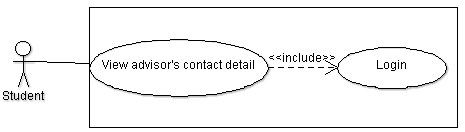


**Description:**

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:**

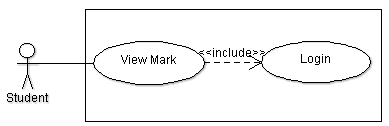


**Description:**

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.

## View Mark Use Case

**Diagram:**

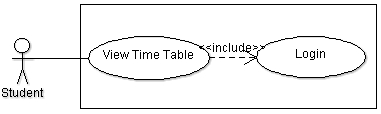


**Description:**

The student will be able to view the marks of the subjects that he/she has enrolled into once he/she has login to the system. Once the student’s identity has been verified and view mark functionality is triggered, he/she will then be able to view the marks he/she scored for the subjects

## View Time Table Use Case

**Diagram:**

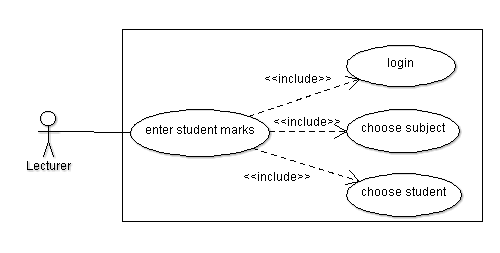


**Description:**

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

## Enter Student Marks Use Case

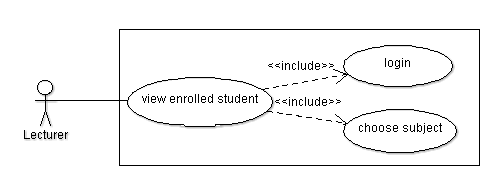
**Diagram:**



**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

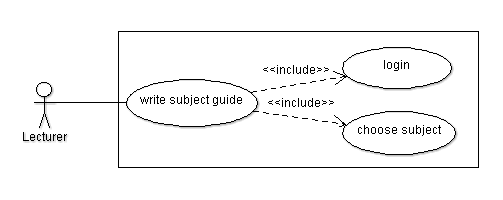
**Diagram:**

**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:**

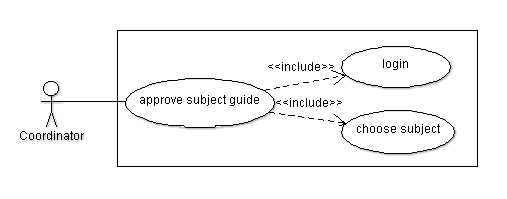


**Description:**

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:**

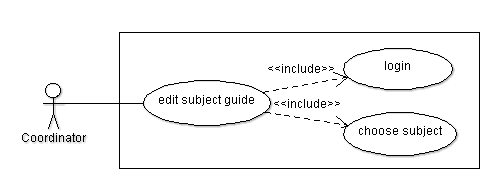


**Description:**

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

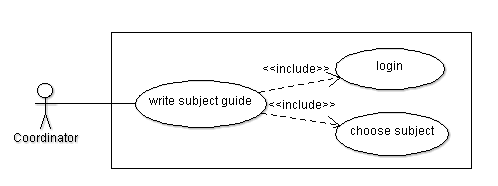
**Diagram:**



**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

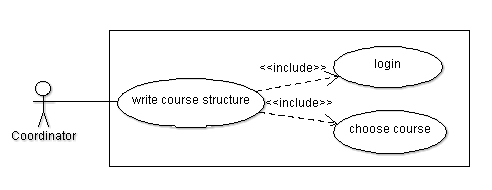
**Diagram:**

**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

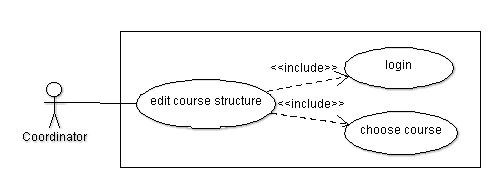
**Diagram:**



**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

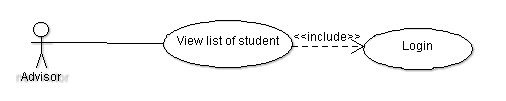
**Diagram:**

**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:**

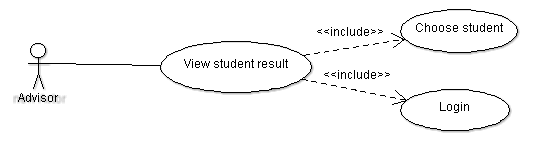


**Description:**

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:**

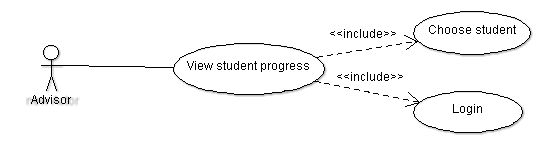


**Description:**

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.