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benefit that this system provides once it has been integrated to the university system and thus replacing the old and outdated version of the university management system.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](#_Toc330918124)

[2. Objectives 1](#_Toc330918125)

[3. Software Process Model 2](#_Toc330918126)

[3.1. Requirement analysis 2](#_Toc330918127)

[3.2. Use Case Diagrams 4](#_Toc330918128)

[3.2.1. Enroll Use Case 5](#_Toc330918129)

[3.2.2. Modify Address Use Case 5](#_Toc330918130)

[3.2.3. View Funding & Fee Status Use Case 6](#_Toc330918131)

[3.2.4. View Advisor Contact Detail Use Case 6](#_Toc330918132)

[3.2.5. View Mark Use Case 7](#_Toc330918133)

[3.2.6. View Time Table Use Case 7](#_Toc330918134)

[3.2.7. Enter Student Marks Use Case 8](#_Toc330918135)

[3.2.8. View Enrolled Student Use Case 8](#_Toc330918136)

[3.2.9. Write Subject Guide Use Case (Lecturer) 9](#_Toc330918137)

[3.2.10. Approve Subject Guide Use Case 9](#_Toc330918138)

[3.2.11. Edit Subject Guide Use Case 10](#_Toc330918139)

[3.2.12. Write Subject Guide Use Case (Coordinator) 10](#_Toc330918140)

[3.2.13. Write Course Structure Use Case 11](#_Toc330918141)

[3.2.14. Edit Course Structure Use Case 11](#_Toc330918142)

[3.2.15. Upload Student’s Document Use Case (Coordinator) 12](#_Toc330918143)

[3.2.16. View List of Student Use Case (Coordinator) 12](#_Toc330918144)

[3.2.17. View Student Result Use Case (Coordinator) 13](#_Toc330918145)

[3.2.18. View Student Progress Use Case (Coordinator) 13](#_Toc330918146)

[3.2.19. Write Meeting Note Use Case (Coordinator) 14](#_Toc330918147)

[3.2.20. Override Enrollment Rules (Coordinator) 14](#_Toc330918148)

[3.2.21. View List of Student Use Case (Advisor) 15](#_Toc330918149)

[3.2.22. View Student Result Use Case (Advisor) 15](#_Toc330918150)

[3.2.23. View Student Progress Use Case (Advisor) 16](#_Toc330918151)

[3.2.24. Upload Student’s Document Use Case (Advisor) 16](#_Toc330918152)

[3.2.25. Override Enrollment Rules (Advisor) 17](#_Toc330918153)

[3.2.26. Write Meeting Note Use Case (Advisor) 17](#_Toc330918154)

[3.2.27. Enter Number of Staff Use Case 18](#_Toc330918155)

[3.2.28. Enter Work Time 18](#_Toc330918156)

[3.2.29. View Students Funding & Fee Status Use Case 19](#_Toc330918157)

[3.2.30. Generate Report Use Case 19](#_Toc330918158)

[3.3. Persistent Design 20](#_Toc330918159)

[3.4. Sequence Diagram 21](#_Toc330918160)

[3.4.1. Enroll Sequence Diagram 21](#_Toc330918161)

[3.4.2. Modify Address Sequence Diagram 22](#_Toc330918162)

[3.4.3. View Funding & Fee Status Sequence Diagram 23](#_Toc330918163)

[3.4.4. View Advisor Contact Detail Sequence Diagram 24](#_Toc330918164)

[3.4.5. View Mark Sequence Diagram 25](#_Toc330918165)

[3.4.6. View Time Table Sequence Diagram 26](#_Toc330918166)

[3.4.7. Enter Student Marks Sequence Diagram 27](#_Toc330918167)

[3.4.8. View Enrolled Student Sequence Diagram 28](#_Toc330918168)

[3.4.9. Write subject guide Sequence Diagram 29](#_Toc330918169)

[3.4.10. Approve Subject Guide Sequence Diagram 30](#_Toc330918170)

[3.4.11. Edit Subject Guide Sequence Diagram 31](#_Toc330918171)

[3.4.12. Write Subject Guide Sequence Diagram 32](#_Toc330918172)

[3.4.13. Write Course Structure Sequence Diagram 33](#_Toc330918173)

[3.4.14. Edit Course Structure Sequence Diagram 34](#_Toc330918174)

[3.4.15. Upload Student’s Document Sequence Diagram ( Coordinator) 35](#_Toc330918175)

[3.4.16. View List of Student Sequence Diagram (Coordinator) 36](#_Toc330918176)

[3.4.17. View Student Result Sequence Diagram (Coordinator) 37](#_Toc330918177)

[3.4.18. View Student Progress Sequence Diagram (Coordinator) 38](#_Toc330918178)

[3.4.19. Write Meeting Note Sequence Diagram (Coordinator) 39](#_Toc330918179)

[3.4.20. Override Enrollment Rules (Coordinator) 40](#_Toc330918180)

[3.4.21. View List of Student Sequence Diagram (Advisor) 41](#_Toc330918181)

[3.4.22. View Student Result Sequence Diagram (Advisor) 42](#_Toc330918182)

[3.4.23. View Student Progress Sequence Diagram (Advisor) 43](#_Toc330918183)

[3.4.24. Upload Student’s Document Sequence Diagram (Advisor) 44](#_Toc330918184)

[3.4.25. Override Enrollment Rules (Advisor) 45](#_Toc330918185)

[3.4.26. Write Meeting Note Sequence Diagram (Advisor) 46](#_Toc330918186)

[3.4.27. Enter Number of Staff Sequence Diagram 47](#_Toc330918187)

[3.4.28. Enter Work Time Sequence Diagram 48](#_Toc330918188)

[3.4.29. View Students Funding & Fee Status Sequence Diagram 49](#_Toc330918189)

[3.4.30. Generate Report Sequence Diagram 50](#_Toc330918190)

[3.5. Deployment Diagram 51](#_Toc330918191)

[3.6. Members Roles and Task Distribution 51](#_Toc330918192)

[3.7. Tools and Environment 53](#_Toc330918193)

[3.8. Feasibility Studies 53](#_Toc330918194)

[3.8.1. Organizational Feasibility 53](#_Toc330918195)

[3.8.2. Resource Feasibility 53](#_Toc330918196)

[3.8.3. Schedule Feasibility 54](#_Toc330918197)

[3.9. System Scope Document 54](#_Toc330918198)

[3.10. Class Diagram 56](#_Toc330918199)

[4. Functionalities 57](#_Toc330918200)

[4.1. Iteration 1 57](#_Toc330918201)

[4.2. Iteration 2 58](#_Toc330918202)

[4.3. Iteration 3 58](#_Toc330918203)

[5. Conclusion 59](#_Toc330918204)

# 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:

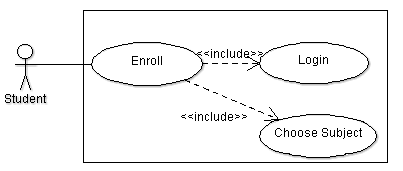
|  |  |
| --- | --- |
| No | Non-Functional Requirements |
| 1 | Login system |
| 2 | Platform independent |
| 3 | Small system footprint |
| 4 | Less than 1 hour of training time |
| 5 | Fast processing time |
| 6 | GUI based |

## 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:**

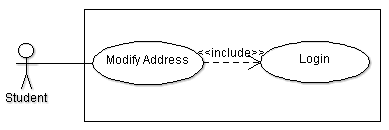


**Description:**

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:**

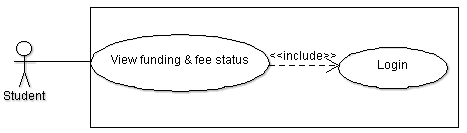


**Description:**

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:**

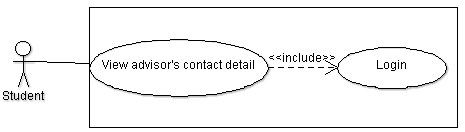


**Description:**

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:**

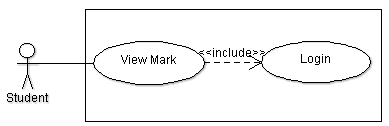


**Description:**

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:**

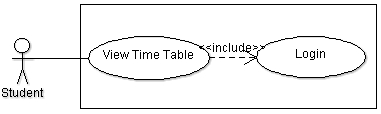


**Description:**

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:**

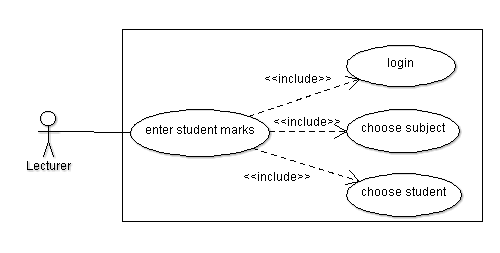


**Description:**

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

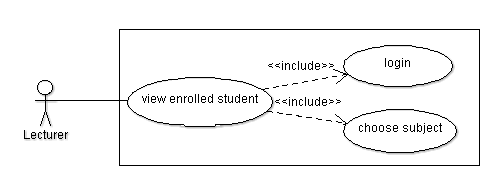
**Diagram:**



**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

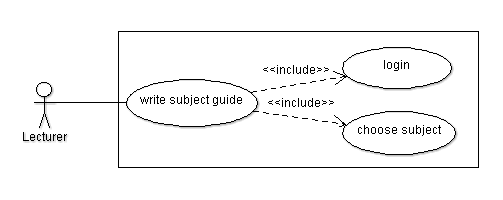
**Diagram:**

**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 (Lecturer)

**Diagram:**

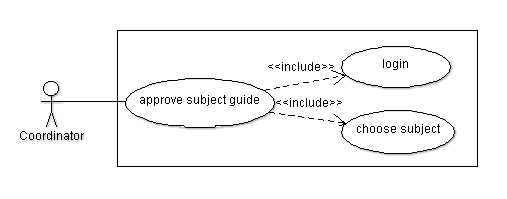


**Description:**

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:**

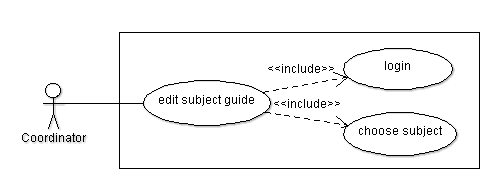


**Description:**

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:**

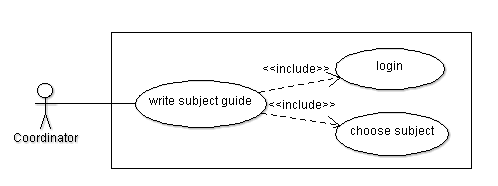


**Description:**

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 (Coordinator)

**Diagram:**

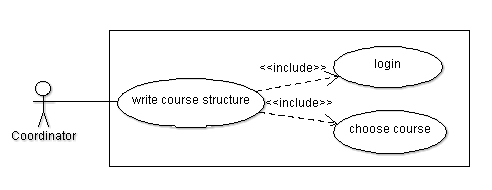


**Description:**

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

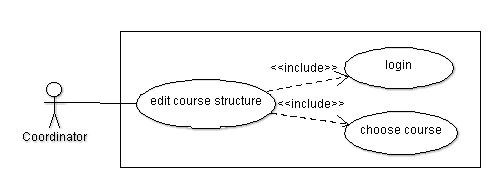
**Diagram:**



**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

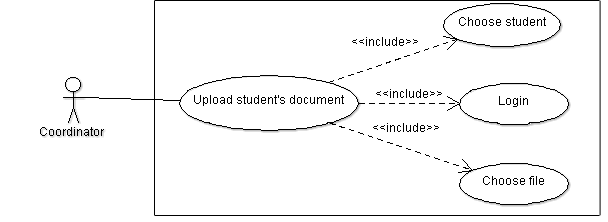
**Diagram:**

**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.

## Upload Student’s Document Use Case (Coordinator)

**Diagram:**

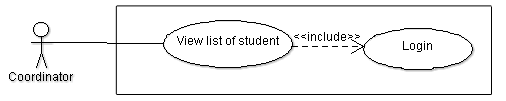


**Description:**

The coordinator will be able to upload his/her student’s document once he/she has login to the system. The coordinator must select one student then choose the file of the selected student to be uploaded.

## View List of Student Use Case (Coordinator)

**Diagram:**

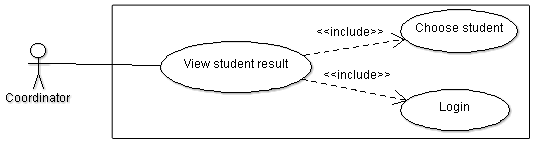


**Description:**

The coordinator 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 (Coordinator)

**Diagram:**

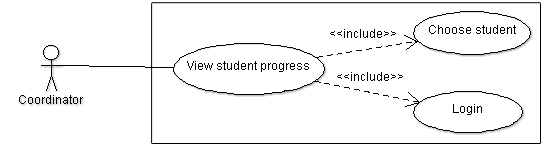


**Description:**

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

## View Student Progress Use Case (Coordinator)

**Diagram:**

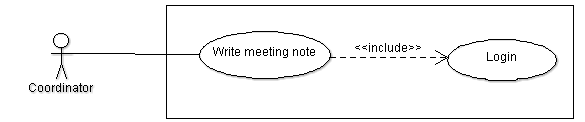


**Description:**

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

## Write Meeting Note Use Case (Coordinator)

**Diagram:**

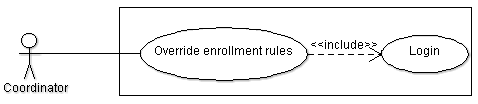


**Description:**

The coordinator can write a meeting note after the meeting with his/her student. But of course the coordinator needs to login to the system first.

## Override Enrollment Rules (Coordinator)

**Diagram:**

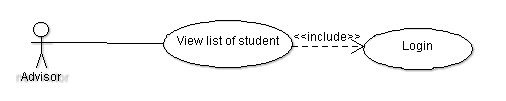
****

**Description:**

The coordinator must login to the system first before he/she can override enrollment rules. Once his identity has been verified, he is now able to override the enrollment rules.

## View List of Student Use Case (Advisor)

**Diagram:**

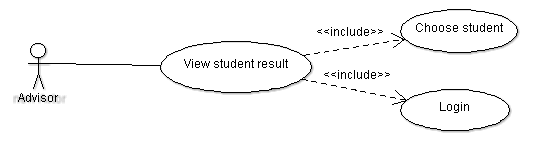


**Description:**

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 (Advisor)

**Diagram:**

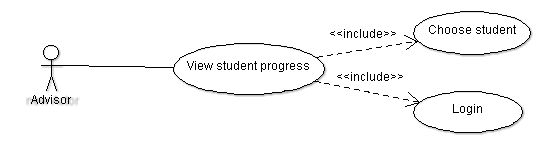


**Description:**

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 (Advisor)

**Diagram:**

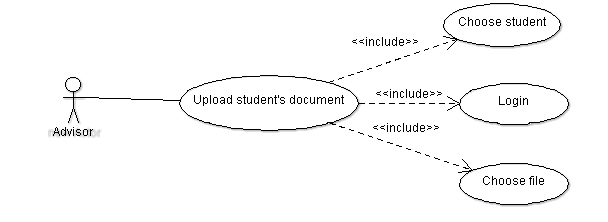


**Description:**

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

## Upload Student’s Document Use Case (Advisor)

**Diagram:**

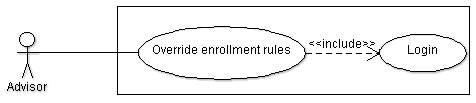


**Description:**

The student advisor will be able to upload his/her student’s document once he/she has login to the system. The advisor must select one student then choose the file of the selected student to be uploaded.

## Override Enrollment Rules (Advisor)

**Diagram:**

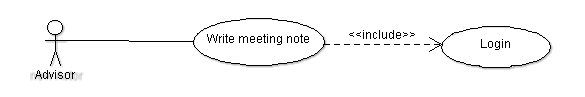


**Description:**

The student advisor must login to the system first before he/she can override enrollment rules. Once his identity has been verified, he is now able to override the enrollment rules.

## Write Meeting Note Use Case (Advisor)

**Diagram:**

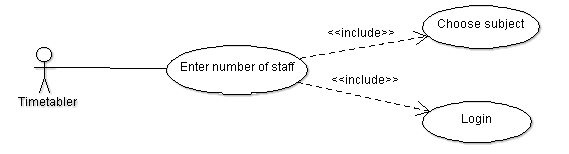


**Description:**

The student advisor can write a meeting note after the meeting with his/her student. But of course the advisor needs to login to the system first.

## Enter Number of Staff Use Case

**Diagram:**

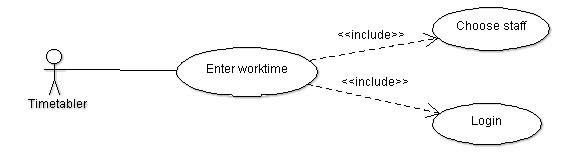


**Description:**

The timetabler will be able to enter the number of staff of a subject once he/she has login to the system. The timetabler will choose the subject to specify to what subject that he/she want to add the staff.

## Enter Work Time

**Diagram:**

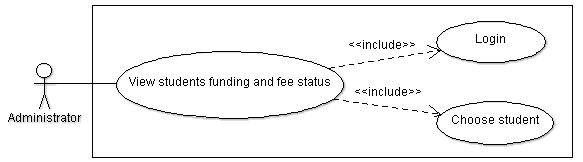


**Description:**

The timetabler will be able to enter the work time for the specified staff once he/she login to the system.

## View Students Funding & Fee Status Use Case

**Diagram:**

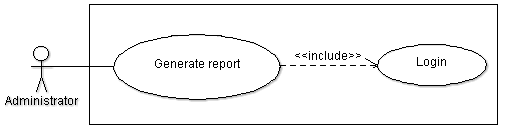


**Description:**

The administrator will be able to view the students funding and fee status once he/she has login to the system. Once the administrator’s identity has been verified he/she will then be required to choose the student that he/she wants to view the status on.

## Generate Report Use Case

**Diagram:**



**Description:**

The administrator will be able to generate report once he/she has login to the system. Once the administrator’s identity has been verified he is now allowed to write the report.

## Persistent Design

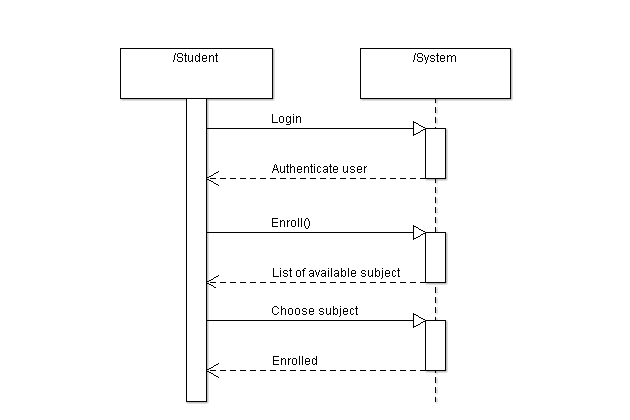
In developing this system, our team has already decided that we are going to implement a database server to keep and process all the data traffic. Every data which is requested by the user will be retrieved from the database server, passed on to application server. The data from application server is then passed on to the web server which will then send the HTML reply back to the browser in the users’ computer showing the requested information.

When the users write data into the database, the data will be sent to the application server. The data will then be processed and saved directly into the database server.

## Sequence Diagram

## Enroll Sequence Diagram

**Diagram:**

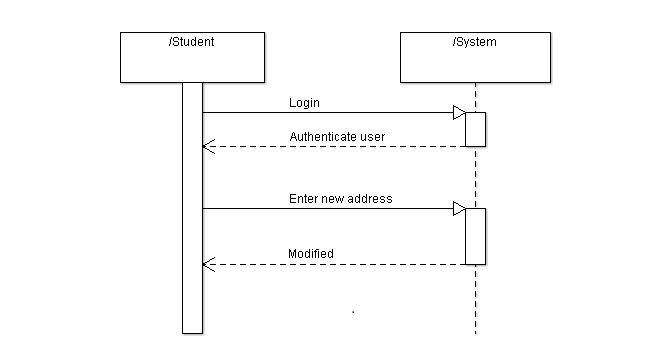


**Description**

The student will firstly login to the system and the system will verify the user. Once the student login successfully, he/she is now allowed to select the enroll button and the list of available subject will be displayed. The student can choose the subject to be enrolled. Lastly the system will enroll the chosen subject.

## Modify Address Sequence Diagram

**Diagram:**

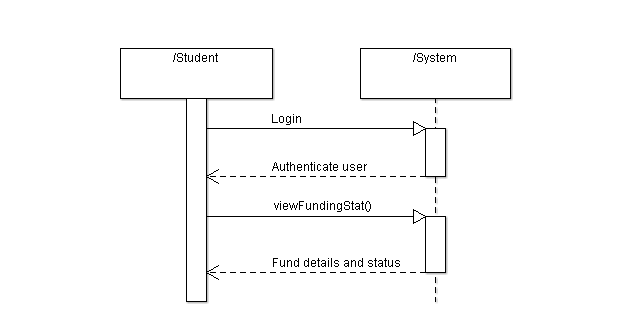


**Description**

The student will firstly login to the system and the system will verify the user. Once the student login successfully, he/she is now allowed to modify the address. Once it is done, the system will automatically modifying it.

## View Funding & Fee Status Sequence Diagram

**Diagram:**

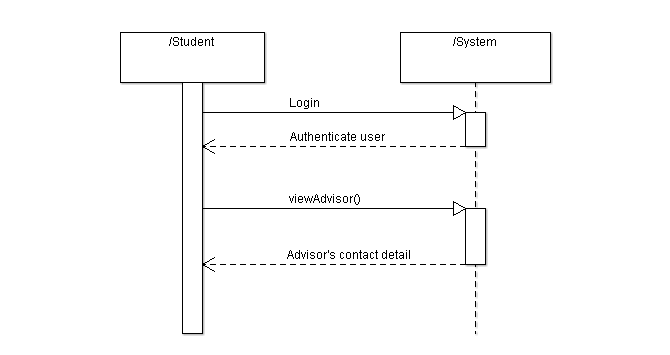


**Description**

The student will firstly login to the system and the system will verify the user. Once the student login successfully, he/she is now allowed to select the view funding status button and then the fund details and the fee status will be displayed.

## View Advisor Contact Detail Sequence Diagram

**Diagram:**

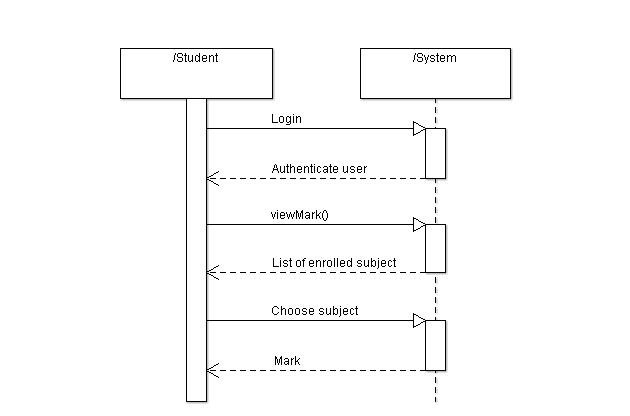


**Description**

The student will firstly login to the system and the system will verify the user. Once the student login successfully, he/she is now allowed to select the view advisor button and then the contact detail of the advisor will be displayed.

## View Mark Sequence Diagram

**Diagram:**

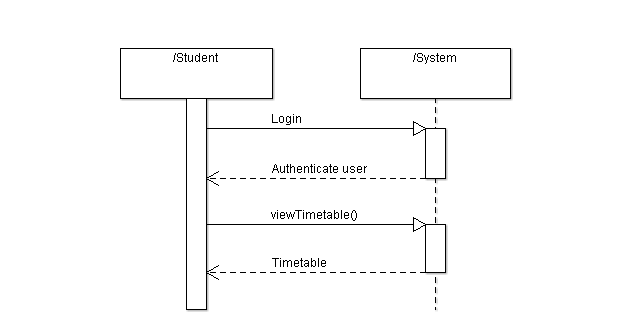


**Description**

The student will firstly login to the system and the system will verify the user. Once the student login successfully, he/she is now allowed to select the view mark button and the list of enrolled subject will be displayed. Now the student can choose once subject then the system will display the mark of the chosen subject.

## View Time Table Sequence Diagram

**Diagram:**

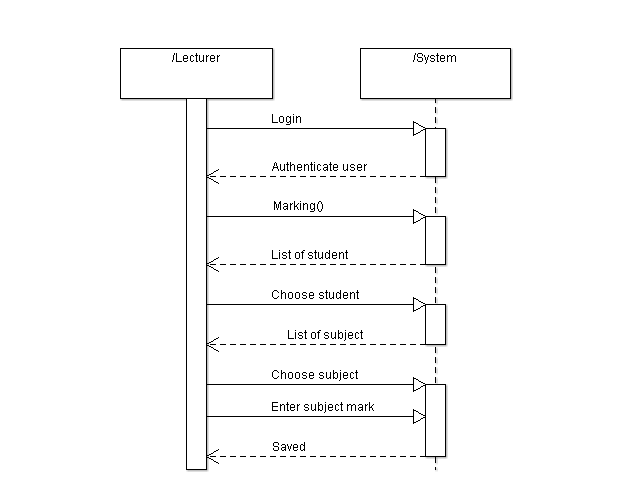


**Description**

The student will firstly login to the system and the system will verify the user. Once the student login successfully, he/she is now allowed to select the view timetable button and then his/her timetable will be displayed.

## Enter Student Marks Sequence Diagram

**Diagram:**

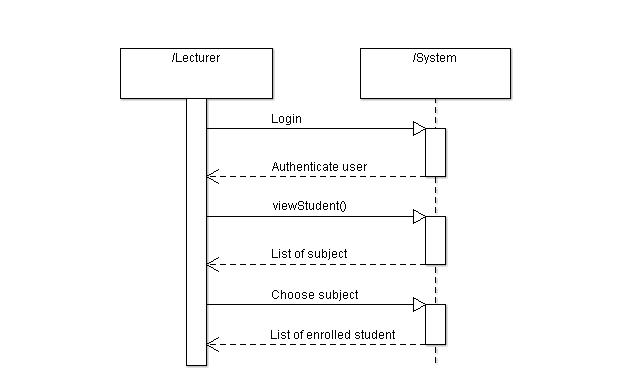


**Description**

The lecturer will firstly login to the system and the system will verify the user. Once the lecturer login successfully, he/she is now allowed to select the marking button and the list of student will be displayed. The lecturer can choose one of the given student then the list of subject enrolled by that student will be displayed. Now he/she can choose one subject and then enter the mark for the selected subject. Once it is done, the system will automatically save it.

## View Enrolled Student Sequence Diagram

**Diagram:**

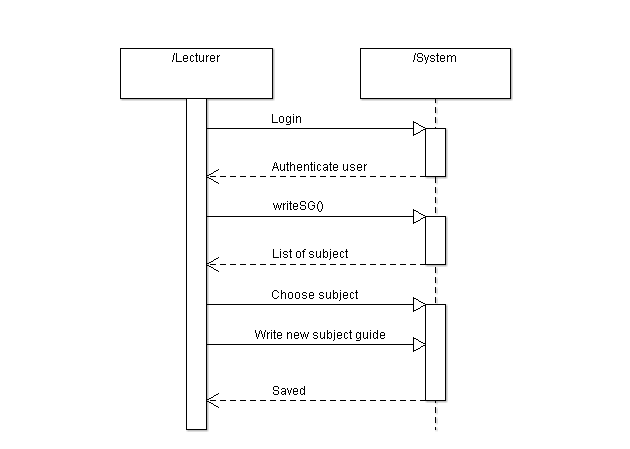


**Description:**

The lecturer will firstly login to the system and the system will verify the user. Once the lecturer login successfully, he/she is now allowed to select the view student button and the list of subject will be displayed. Now the lecturer can choose one of the given subjects then the list of student enrolled in this subject will be displayed.

## Write subject guide Sequence Diagram

**Diagram:**

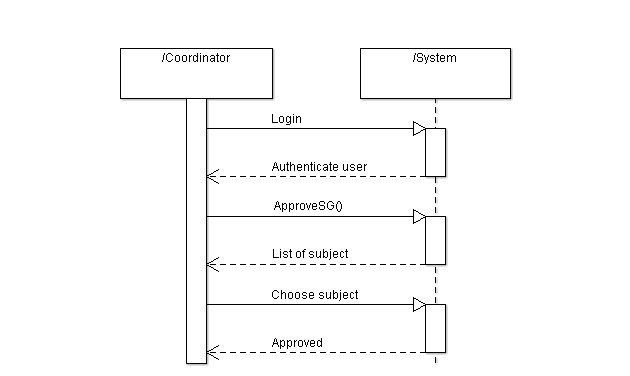


**Description:**

The lecturer will firstly login to the system and the system will verify the user. Once the lecturer login successfully, he/she is now allowed to select the write subject guide button and the list of subject will be displayed. The lecturer can choose one of the given subject then write a new subject guide for the selected subject. Once it is done, the system will automatically save it.

## Approve Subject Guide Sequence Diagram

**Diagram:**

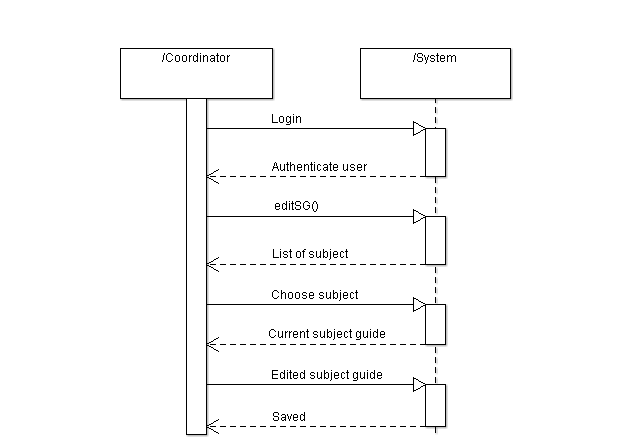


**Description:**

The coordinator will firstly login to the system and the system will verify the user. Once the coordinator login successfully, he/she is now allowed to select the approve subject guide button and the list of subject will be displayed. Now the coordinator can choose one subject to be approved then the system will approve it.

## Edit Subject Guide Sequence Diagram

**Diagram:**

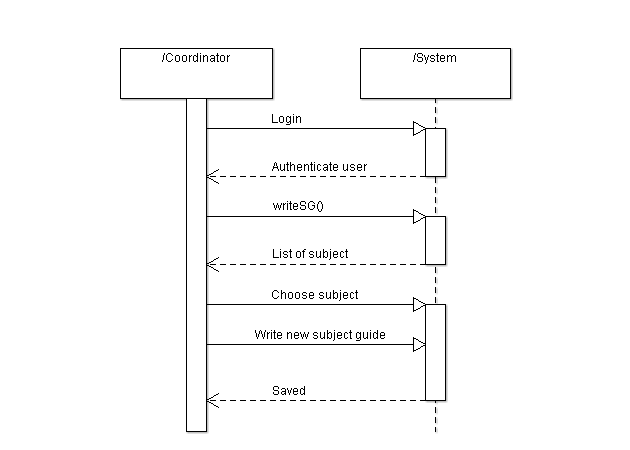


**Description:**

The coordinator will firstly login to the system and the system will verify the user. Once the coordinator login successfully, he/she is now allowed to select the edit subject guide button and the list of subject will be displayed. The coordinator can choose one subject to be edited then the system will display the current subject guide of the chosen subject. Now he/she can modify the subject guide. Once it is done, the system will automatically save it.

## Write Subject Guide Sequence Diagram

**Diagram:**

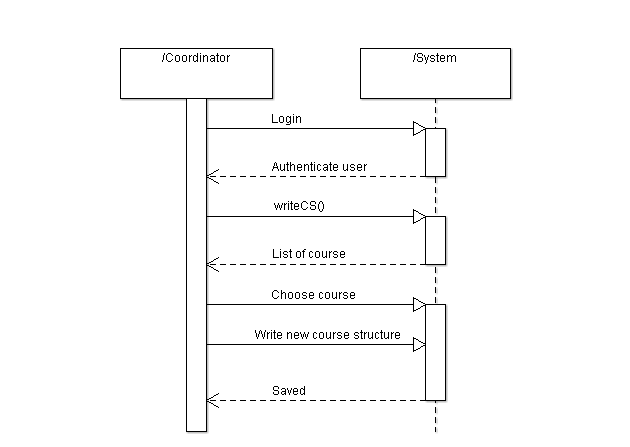


**Description:**

The coordinator will firstly login to the system and the system will verify the user. Once the coordinator login successfully, he/she is now allowed to select the write subject guide button and the list of subject will be displayed. The coordinator can choose one of the given subject then write a new subject guide for the selected subject. Once it is done, the system will automatically save it.

## Write Course Structure Sequence Diagram

**Diagram:**

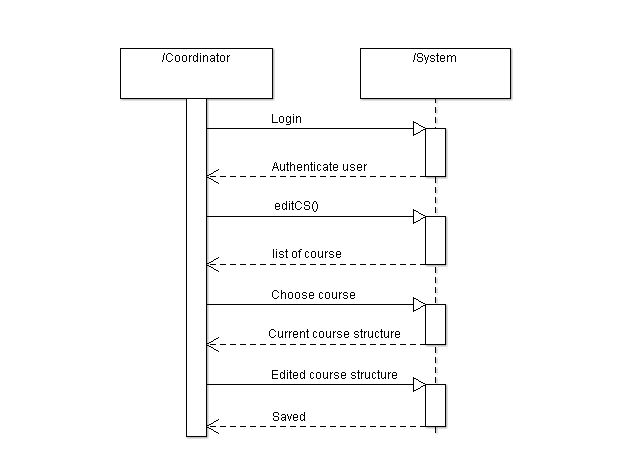


**Description:**

The coordinator will firstly login to the system and the system will verify the user. Once the coordinator login successfully, he/she is now allowed to select the write course structure button and the list of course will be displayed. The coordinator can choose one of the given course then write a new course structure for the selected course. Once it is done, the system will automatically save it.

## Edit Course Structure Sequence Diagram

**Diagram:**

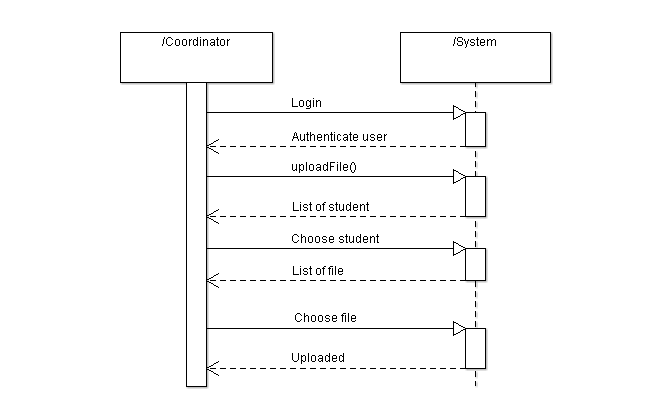


**Description:**

The coordinator will firstly login to the system and the system will verify the user. Once the coordinator login successfully, he/she is now allowed to select the edit course structure button and the list of course will be displayed. The coordinator can choose one course to be edited then the system will display the current course structure of the chosen course. Now he/she can modify the course structure. Once it is done, the system will automatically save it.

## Upload Student’s Document Sequence Diagram ( Coordinator)

**Diagram:**

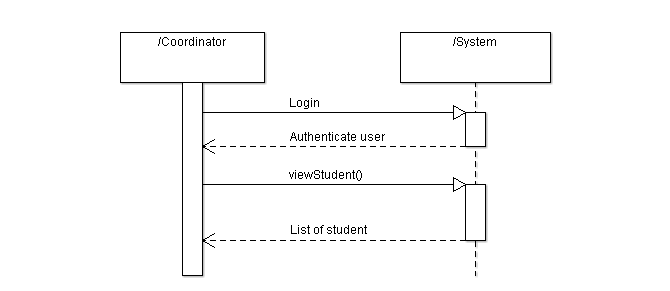


**Description:**

The coordinator will firstly login to the system and the system will verify the user. Once the coordinator login successfully, he/she is now allowed to select the upload file button and the list of student will be displayed. Now the coordinator can choose one of the listed student and the system show the list of file that belong to the selected student. Lastly the coordinator can choose the file to be uploaded. Once it is done, the system will upload the selected file.

## View List of Student Sequence Diagram (Coordinator)

**Diagram:**

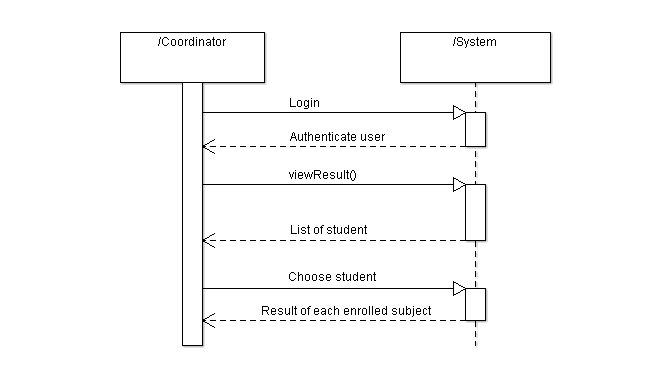


**Description:**

The coordinator will firstly login to the system and the system will verify the user. Once the coordinator login successfully, he/she is now allowed to select the view student button and the list of student will be displayed.

## View Student Result Sequence Diagram (Coordinator)

**Diagram:**

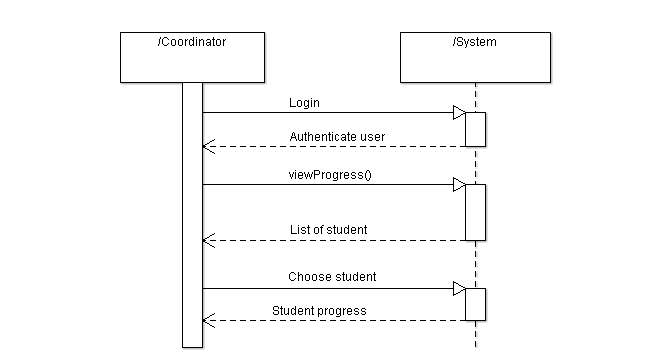


**Description:**

The coordinator will firstly login to the system and the system will verify the user. Once the coordinator login successfully, he/she is now allowed to select the view result button and the list of student will be displayed. Now the coordinator can choose one of the listed student then the system will show the result of each subject enrolled by the student.

## View Student Progress Sequence Diagram (Coordinator)

**Diagram:**

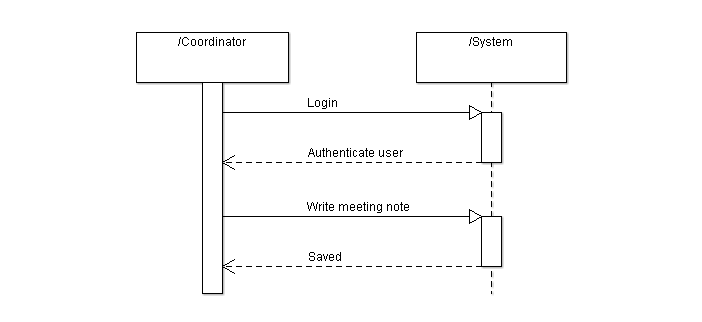


**Description:**

The coordinator will firstly login to the system and the system will verify the user. Once the coordinator login successfully, he/she is now allowed to select the view progress button and the list of student will be displayed. Now the coordinator can choose one of the listed student then the system will show the progression of the selected student.

## Write Meeting Note Sequence Diagram (Coordinator)

**Diagram:**

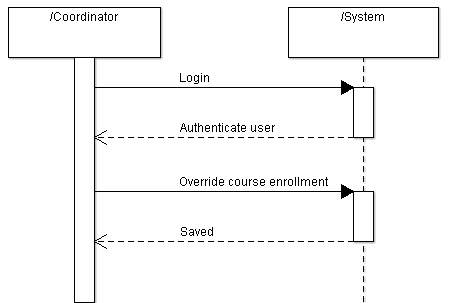


**Description:**

The coordinator will firstly login to the system and the system will verify the user. If the login is successful, he/she is now allowed to write the meeting note. Once it is done, the system will automatically save it.

## Override Enrollment Rules (Coordinator)

**Diagram:**

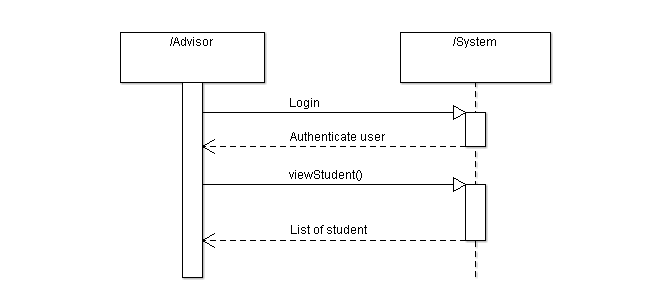
****

**Description:**

The coordinator will firstly login to the system and the system will verify the user. If the login is successful, he/she is now allowed to override course enrollment. Once it is done, the system will automatically save it.

## View List of Student Sequence Diagram (Advisor)

**Diagram:**

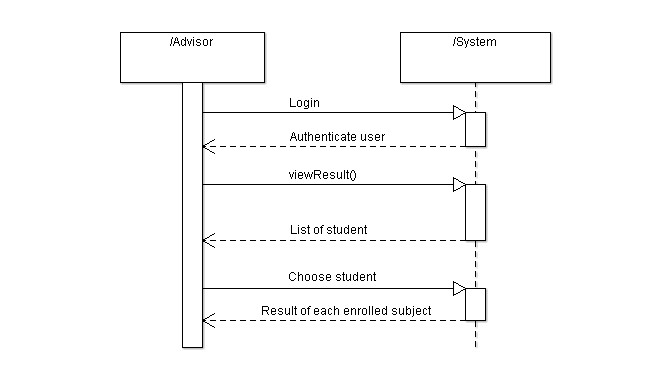


**Description:**

The advisor will firstly login to the system and the system will verify the user. Once the advisor login successfully, he/she is now allowed to select the view student button and the list of student will be displayed.

## View Student Result Sequence Diagram (Advisor)

**Diagram:**

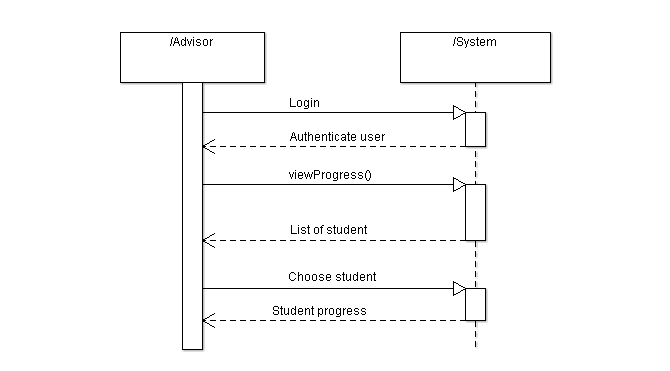


**Description:**

The advisor will firstly login to the system and the system will verify the user. Once the advisor login successfully, he/she is now allowed to select the view result button and the list of student will be displayed. Now the advisor can choose one of the listed students then the system will show the result of each subject enrolled by the student.

## View Student Progress Sequence Diagram (Advisor)

**Diagram:**

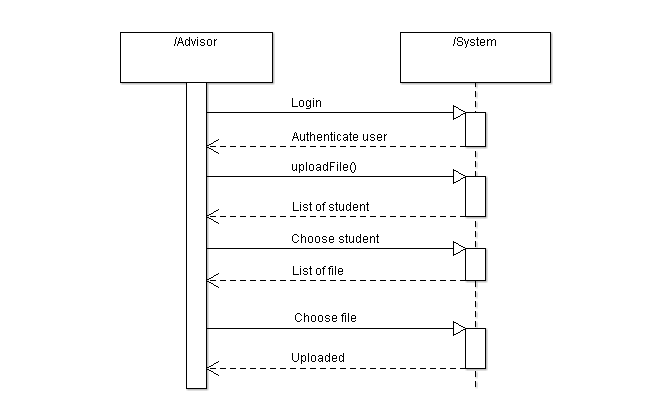


**Description:**

The advisor will firstly login to the system and the system will verify the user. Once the advisor login successfully, he/she is now allowed to select the view progress button and the list of student will be displayed. Now the advisor can choose one of the listed student then the system will show the progression of the selected student.

## Upload Student’s Document Sequence Diagram (Advisor)

**Diagram:**

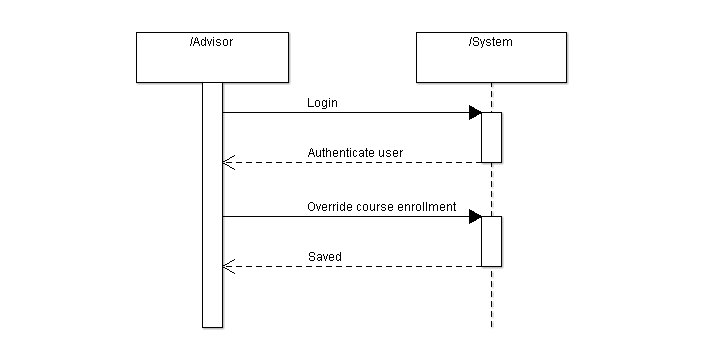


**Description:**

The advisor will firstly login to the system and the system will verify the user. Once the advisor login successfully, he/she is now allowed to select the upload file button and the list of student will be displayed. Now the advisor can choose one of the listed student and the system show the list of file that belongs to the selected student. Lastly the advisor can choose the file to be uploaded. Once it is done, the system will upload the selected file.

## Override Enrollment Rules (Advisor)

**Diagram:**

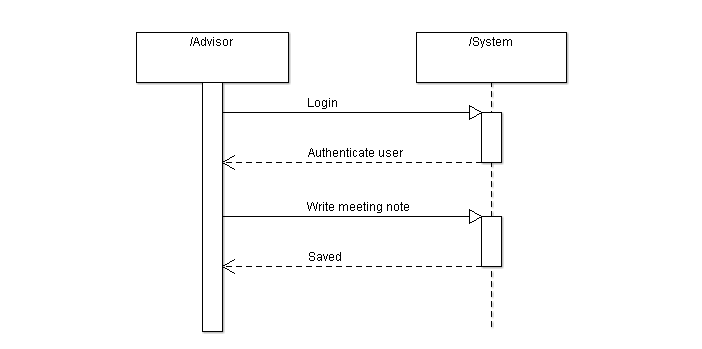
****

**Description:**

The advisor will firstly login to the system and the system will verify the user. Once the advisor login successfully, he/she is now allowed to override the course enrollment rules. Once it is done, the system will save the changes made.

## Write Meeting Note Sequence Diagram (Advisor)

**Diagram:**

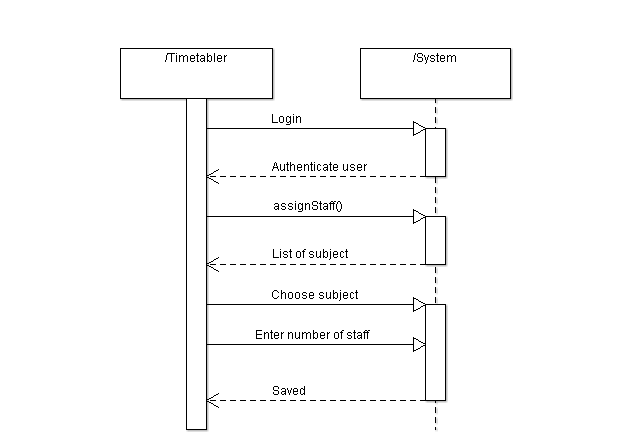


**Description:**

The advisor will firstly login to the system and the system will verify the user. If the login is successful, he/she is now allowed to write the meeting note. Once it is done, the system will automatically save it.

## Enter Number of Staff Sequence Diagram

**Diagram:**

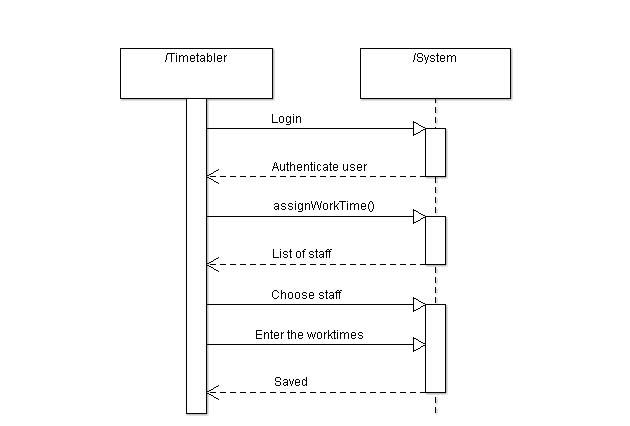


**Description:**

The timetabler will firstly login to the system and the system will verify the user. Once the timetabler login successfully, he/she is now allowed to select the assign staff button and the list of subject will be displayed. The timetabler can choose one subject then enter the number of staff to be assigned to that subject. Once it is done, the system will automatically save it.

## Enter Work Time Sequence Diagram

**Diagram:**

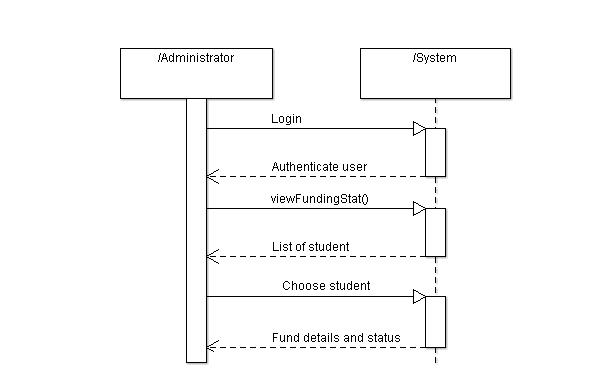


**Description:**

The timetabler will firstly login to the system and the system will verify the user. Once the timetabler login successfully, he/she is now allowed to select the assign work time button and the list of staff will be displayed. The timetabler can choose one staff then enter the worktime for a particular staff. Once it is done, the system will automatically save it.

## View Students Funding & Fee Status Sequence Diagram

**Diagram:**

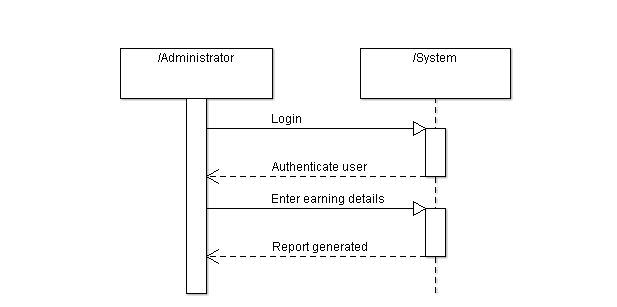
****

**Description:**

The administrator will firstly login to the system and the system will verify the user. Once the administrator login successfully, he/she is now allowed to select the view students funding button and the list of students will be displayed. The administrator will then choose one particular student to view his/her fees and funding status.

## Generate Report Sequence Diagram

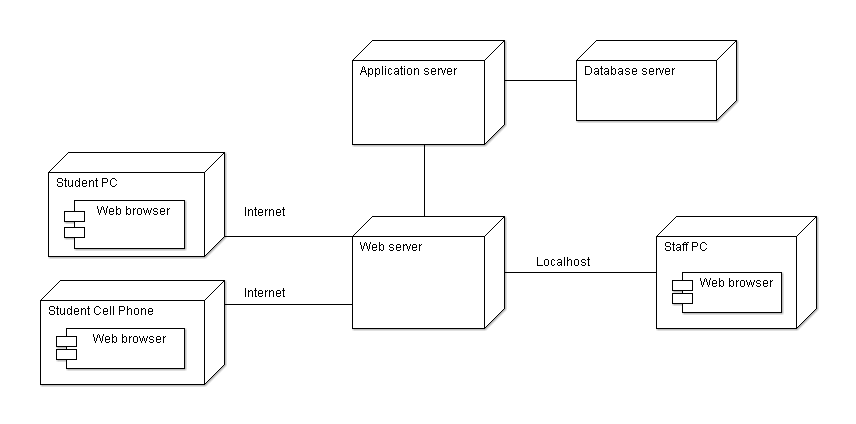
**Diagram:**

****

**Description:**

The administrator will firstly login to the system and the system will verify the user. Once the administrator has login successfully, he/she is now allowed to select the generate report button and the system will prompt the administrator to enter the earning details. Once finish with the input, the report will be automatically generated by the system.

## Deployment Diagram



There are three hardware that are connected to the web server, they are student PC, student cell phone, and staff PC. The web browser is installed in the client’s device. The student devices are connected to the web server via internet, while the staff pc connected to the web server through local network. The client will first send data to the web server and then the data will be sent directly to the application server to be processed. After finished processing the data, the application server will retrieve or save the data from or to the database server according to the user’s request.

## Members Roles and Task Distribution

|  |  |  |
| --- | --- | --- |
| Name | Role | Task |
| Hendro | Manager, Analysts, Designers, Documenter | * Login function * Design student interface * View advisors contact detail function (Student) * Check students eligibility function * Assign advisor function * View student progress function (Advisor) * View student progress function (Coordinator) * Upload document function (Advisor) * Upload document function (Coordinator) * Write subject guide function (Lecturer) * View enrolled student (Lecturer) * Assign mark function (Lecturer) |
| Habibie | Systems Architect, Analysts, Designers, Documenter | * Approve subject guide function (Coordinator) * Design staff interface * Modify address function (Student) * View mark and timetable function (Student) * View student function (Advisor) * View student function (Coordinator) * View result function (Advisor) * View result function (Coordinator) * Write course structure function (Coordinator) * Edit course structure function (Coordinator) * View funding and fee status function (Administrator) * View funding and fee status function (Student) * Generate earning report (Administrator) |
| Fredric | Requirement Engineer, Analysts, Designers | * Enroll function (Student) * Write note function (Advisor) * Write note function (Coordinator) * Override course enrollment rules function (Advisor) * Override course enrollment rules function (Coordinator) * Allocate staff function (Timetabler) * Assign staffs worktime function (Timetabler) * Write subject guide function (Coordinator) * Edit subject guide function (Coordinator) * Create timetable function * Store applicant function * Create student record function * Save function |

## Tools and Environment

After the analysis on both functional and non-functional requirements, our team continues with the analysis on the tools and equipment to be used. Based upon the project proposal, out team has come to an agreement on the programming language that is going to be used is Java, compiled and coded using NetBeans IDE 7.0.1 and ArgoUML for developing all the diagrams which are provided in this report such as Use Case diagram, Sequence diagram, Class diagram and Deployment diagram.

For the environment, as we all know that Java program is platform independent, our team is pretty confident that Big U Management System will work perfectly fine in every type of OS environment.

## Feasibility Studies

## Organizational Feasibility

The Big U Management System is a system that will help simplify all the works that were previously done manually. This includes the works of students, lecturers, student advisors, coordinators, timetablers and administrators. The system will make all their works faster and more efficient. The system is also very easy to use and thus, the training time can be very short for even computer illiterate users.

The only drawback for the system implementation is that the university doesn’t have the proper network infrastructure for a web server that is connected to the internet. This can cost a considerable amount to implement an internet connected web server.

## Resource Feasibility

The Big U Management System development does not face any limitations in term of the resource. The budget is more than enough to develop the system as it doesn’t require much of resources. However, to implement the system, quite a large amount of money needs to be spent for buying a new domain for the web server and new equipment to implement an internet connected web server. The new system will also need a more powerful web server, application server, and database server to avoid overloading.

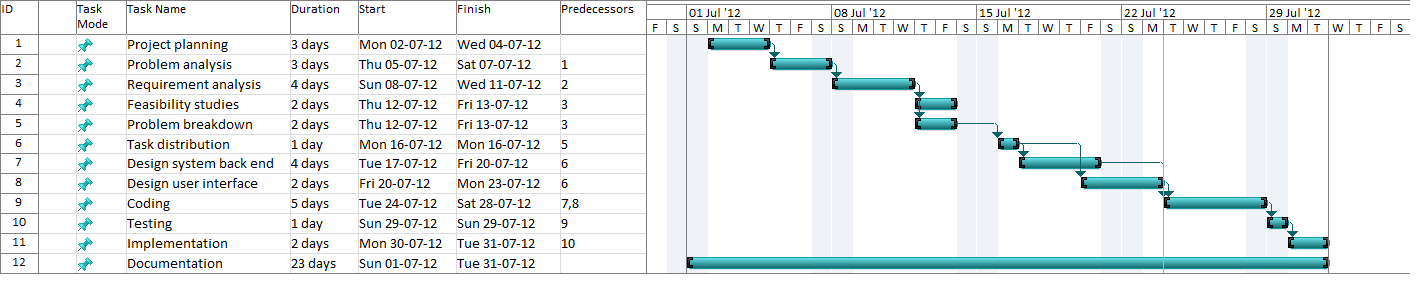
To build and implement the system itself, the time required is not really that long. During the implementation phase, all the processes done using the old system will be halted for a while as the time is going to be used to replace the old system with the new system.

In terms of development tools, it is not really a problem as our team already owns the required IDE and other additional tools. Our team also has our own computer to develop the new system.

## Schedule Feasibility

This project has been given 1 month for its completion. The amount of time given is more than enough to complete the system. However, more time is needed to implement the system as some changes needed to be made to the current equipment.

The Gantt chart below will display more in detail about our scheduling:



## System Scope Document

System Scope Document

Big U Management System

**Problem Description**

After viewing the project proposal from our client, we have pointed out that the main problem faced by Big U University is regarding the need of a new and better system to replace the old one which is no longer able to cope with the currently growing size of students. Furthermore, a more automated and better solution is expected to overcome the current problem.

**Anticipated Business Benefits**

The main business benefit that can be anticipated from implementing Big U Management System is that this system is expected to be able to help both the students and staffs in Big U University with their daily activities, mostly those related with the academic matters. With the presence of this system, a lot of works which were previously must be done manually can now be done using the system which is far more convenient and simple. Some benefits that are worth mentioning for using Big U Management System are as the following:

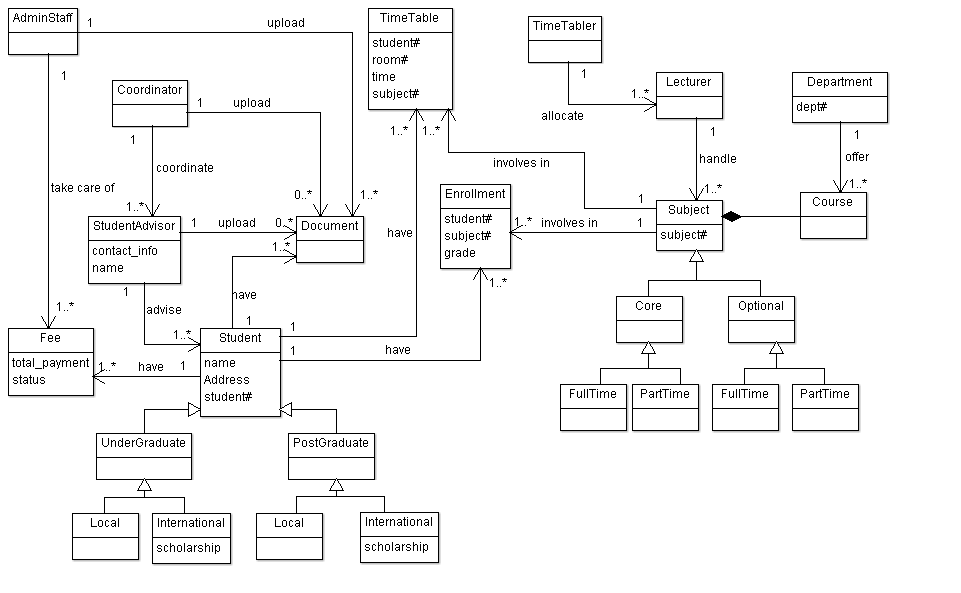
* Easy to use and understand
* Simple and clear GUI
* Centralized database server
* No special application is required (only web browser)

**System Capabilities**

Big U Management System provides a lot of useful capabilities for both the students and staffs of Big U University. Those capabilities are summarized as the following:

* Enroll, modify address, view funding and fee status, view advisor contact detail, view mark and view time table for Students
* Enter student marks, write subject guide and view enrolled student for Lecturers
* Approve subject guide, edit subject guide, write subject guide, write course structure, edit course structure for Coordinators
* Upload student's document, view list of student, view student result, view student progress, write meeting note and override enrollment rules for both Coordinators and Student Advisors

## Class Diagram



The diagram above shows the entities of the system. There are 13 different entities in the system, not including its subcategories. Student has 2 seperate categories, undergraduate student and postgraduate student. Each category is then seperated to 2 more subcategories, local student and international student. Subject has 2 seperate categories, core subject and optional subject. Each category is then seperated to 2 more subcategories, full time subject and part time subject.

The relation of each entities is also shown in the above diagram. Each administrative staff takes care of at least one fee. Each fee can only be handled by one administrative staff. Each student has at least one fee. Each fee can only be associatied with one student. Each student advisor is advising at least one student. Each student can only be advised by one student advisor. Each coordinator coordiantes at least one student advisor. Each student coordinator can only be coordinated by one coordinator. Each student has at least one document that is related to them. Each document can only be related to one student. Each student advisor can upload more than one document related to their student. Each coordinator can upload more than one document related to the student that is advised by their student advisor. Each administrative staff uploads at least one document that is related to a student. Each document can only be uploaded by one student advisor, coordinator, or administrative staff. Each student has at least one enrollment. Each enrollment can only be associated with one student. Each subject is involved in at least one enrollment. Each enrollment can only be associated with one subject. Each student has at least one timetable and each timetable can only be associated with one student. Each subject is involved in at least one timetable and each timetable is associated with one subject. Each timetabler allocates at least one lecturer. Each lecturer can only be allocated by one timetabler. Each lecturer handles at least one subject and each subject dan only be handled by one lecturer. A course has at least one subject. Each department offers at least one subject and each subject is offered only by one department.

The diagram also shows the data that describes each entitiy. Some entities have no data to describe them. A fee is described by the total payment and the status of the fee. All students is described by their name, address, contact information, and student#. For an international student, they are described by one more data, scholarship. A student advisor is described by their name and contact information. Each timetable is described by student#, room#, subject#, and time. A subject is decribed by subject# and each department is desribed by the department#.

# Functionalities

## Iteration 1

|  |  |
| --- | --- |
| Function | Description |
| Login function | Added the feature to login |
| View advisors contact detail function (Student) | Added the feature to view advisor contact detail |
| Modify address function (Student) | Added the feature to modify address |
| View mark and timetable function (Student) | Added the feature to view mark and timetable |
| View funding and fee status function (Student) | Added the feature to view fee and funding status |
| Enroll function (Student) | Added the feature to enroll |
| Save function | Added the feature to save |
| Design student interface | Design process for student interface |
| Design staff interface | Design process for staff interface |
| Write subject guide function (Lecturer) | Added the feature to write subject guide |
| View enrolled student (Lecturer) | Added the feature to view enrolled student |
| Assign mark function (Lecturer) | Added the feature to assign mark |

## Iteration 2

|  |  |
| --- | --- |
| Function | Description |
| Create student record function | Added the feature to create student record |
| Check students eligibility function | Added the feature to check student eligibility |
| View student progress function (Advisor) | Added the feature to view student progress |
| View student progress function (Coordinator) | Added the feature to view student progress |
| Upload document function (Advisor) | Added the feature to upload document |
| Upload document function (Coordinator) | Added the feature to upload document |
| View student function (Advisor) | Added the feature to view student |
| View student function (Coordinator) | Added the feature to view student |
| View result function (Advisor) | Added the feature to view result |
| View result function (Coordinator) | Added the feature to view result |
| Write note function (Advisor) | Added the feature to write note |
| Write note function (Coordinator) | Added the feature to write note |

## Iteration 3

|  |  |
| --- | --- |
| Function | Description |
| Assign advisor function | Added the feature to assign advisor |
| Override course enrollment rules function (Coordinator) | Added the feature to override course enrollment rules |
| Override course enrollment rules function (Advisor) | Added the feature to override course enrollment rules |
| Write course structure function (Coordinator) | Added the feature to write course structure |
| Edit course structure function (Coordinator) | Added the feature to write course structure |
| Approve subject guide function (Coordinator) | Added the feature to approve subject guide |
| Write subject guide function (Coordinator) | Added the feature to write subject guide |
| Edit subject guide function (Coordinator) | Added the feature to edit subject guide |
| View funding and fee status function (Administrator) | Added the feature to view student’s fee and funding status |
| Generate earning report (Administrator) | Added the feature to generate earning report |
| Create timetable function | Added the feature to create time table |
| Store applicant function | Added the feature to store the applicant |
| Allocate staff function (Timetabler) | Added the feature to allocate staff |
| Assign staffs work time function (Timetabler) | Added the feature to assign staffs work time |

# Conclusion

Empowered with every single functionalities provided by our system, we believe that the staffs and students of Big U University will no longer face any difficulties that they faced when they were still using the old system. They will now be able to finish their tasks in a more automated way and with a much fewer time compared to before.