

**BIT304 FINAL YEAR PROJECT I**

**ITERATION BREAKDOWN**

**Ayo Magang: Web Application for Internship in Denpasar**

Rivaldo Bagus Soepardhy (E1700882)

Submitted to the

FACULTY OF COMPUTING AND DIGITAL TECHNOLOGY

(SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY)

In partial fulfilment of the requirements

For the degree of

Bachelor of Information Technology (HONS)

HELP UNIVERSITY

JULY - AUGUST 2020

Chapter 4: ITERATIONS 6

**4.1 Introduction 6**

**4.2 Iteration 1 6**

4.2.1 Introduction 6

4.2.2 Purpose 6

**4.2.3 Context 6**

**4.2.4 Schedule of Iteration Workflows 6**

4.2.5 Iteration Schedule Breakdown 7

4.2.6 Resource Summary 7

4.2.7 Evaluation Criteria 7

4.2.8 Analysis and Design Artifacts 8

4.2.9 Implementation and Testing 8

4.2.10 Iteration Review and Evaluation 10

**4.3 Iteration 2 11**

**4.4 Iteration 3 11**

**4.1 Iteration 1**

4.1.1 Introduction

The goal of doing iteration is to ensure the developed use cases can be working properly without any interference or debug that might slow down our information system of natural disaster. At this moment, we have managed to build a prototype of our system along with 3 functionality there are register user, login user and login all admin. It all work properly and fortunately can be done on time based on our Gantt chart plan.

4.1.2 Purpose

The result of our iteration plan based on the waterfall with prototyping methodology is in accordance with what we have expected. From the feasibility study phase, we have done the project background, project aims and objectives, resource plan, interview and observation. Through the iteration process, we can obtain the complete information not only by holding a meeting with our mentor in the college but also from the disaster agency that really help us in determining the requirement of system. Then, in detailed analysis phase, we have finished the risk management, develop methodology and techniques, proposal, proposal submission, literature review, requirement gathering, and interim presentation. Through the iteration of this phase, we can analyze the risk of system openly from different point of view of our team member, and we are able to choose the appropriate methodology and technique to satisfy the user expectation and also admin. Continued with the detailed design phase, we have done system design, database design and interface of website. From this iteration, we have learned to make a user friendly interface and design specification of our project properly. In construction phase, we have done the iteration for the coding process and prototype. In our prototype, we have three use cases that are already worked such as register of user, login user and login admin. Through the iteration, we have learned to manage our time to code the project and decreases the debug that emerge in unexpected time. In testing phase, we need to ensure that our prototype can show the basic functionality and performance of the future system. So the iteration plan has been implemented to know whether the system already pass the criteria of validation and verification which means that, all the use cases that implemented has worked correctly and has met all the requirement from user and admin. In system delivery and operation & maintenance phase, we have not implemented the iteration plan because these work is to be done in our final project 2 of BIT 305. Therefore the iteration that has been implemented is started from the feasibility of study until testing phase.

**4.1.3 Context**

The implementation of iteration plan to our project is really helping us to determine the appropriate requirement both of design and specification of our information system. Began with feasibility study until the testing phase where we check the correctness of each functionality that we provide in our prototype, there are register for user, login for user and login for all admin. The login admin has separated into 4 category, there are login of super admin, login of province admin, login of natural disaster admin, and login of command post admin. The differences of admin is made based on the interview and observation that we conduct during the feasibility study phase with one of coordinator in disaster recovery agency, BPBD Bali. It is because, each of admin has their own functionality that support each other in dealing with the disaster recovery problem. Through the iteration plan, not only to decide the requirement, but also we can improve our skill in coding the project. Eventhough this is only the basic functionality of register and login, but through the iteration, we can specified the requirement of login and register form design based on user information that is needed and also admin identity that is needed because both of them has specific functionality that differ to one another.

**4.1.4 Schedule of Iteration Workflows**

|  |  |  |  |
| --- | --- | --- | --- |
| **Workflow** | **Start Date** | **End Date** | **Duration (days)** |
| **Use Case 1: Register User** | November, 24th 2017 | December, 4th 2017 | 10 days |
| Analysis | November, 24th 2017 | December, 4th 2017 | 10 days |
| Design | November, 24th 2017 | December, 4th 2017 | 10 days |
| Implementation | November, 24th 2017 | December, 4th 2017 | 10 days |

|  |  |  |  |
| --- | --- | --- | --- |
| Testing | November, 24th 2017 | December, 4th 2017 | 10 days |
| **Use Case 2: Login User** | November, 24th 2017 | December, 4th 2017 | 10 days |
| Analysis | November, 24th 2017 | December, 4th 2017 | 10 days |
| Design | November, 24th 2017 | December, 4th 2017 | 10 days |
| Implementation | November, 24th 2017 | December, 4th 2017 | 10 days |
| Testing | November, 24th 2017 | December, 4th 2017 | 10 days |
| **Use Case 3: Login Admin** | November, 24th 2017 | December, 4th 2017 | 10 days |
| Analysis | November, 24th 2017 | December, 4th 2017 | 10 days |
| Design | November, 24th 2017 | December, 4th 2017 | 10 days |
| Implementation | November, 24th 2017 | December, 4th 2017 | 10 days |
| Testing | November, 24th 2017 | December, 4th 2017 | 10 days |

*Table 3.6.4.1 Iteration Workflow Schedule*

There is no slip date when develop all these use cases and we can provide it all in our prototype on time. This is because we have done the iteration in feasibility study that useful to gather the full information from the mentor in college also from the disaster agency’s delegation.

4.1.5 Iteration Schedule Breakdown

|  |  |  |  |
| --- | --- | --- | --- |
| **Task Name** | **Start** | **Finish** | **Assigned To** |
| Use Case 1: Login All Admin | November 24, 2017 | December 4, 2017 | Tommy,Daniya,Sinthia |
| **1.1 Analysis** | November 24, 2017 | December 4, 2017 | Daniya,Sinthia |
| 1.1.1 Analysis of login form requirement for admin | November 24, 2017 | December 4, 2017 | Daniya,Sinthia |
| **1.2 Design** | November 24, 2017 | December 4, 2017 | Tommy |

|  |  |  |  |
| --- | --- | --- | --- |
| 1.2.1 Design of login form | November 24, 2017 | December 4, 2017 | Tommy |
| **1.3 Implementation** | November 24, 2017 | December 4, 2017 | Tommy |
| 1.3.1  Implementation of login form in website | November 24, 2017 | December 4, 2017 | Tommy |
| **1.4 Testing** | November 24, 2017 | December 4, 2017 | Sinthia |
| 1..4.1 Testing of login of all admin | November 24, 2017 | December 4, 2017 | Sinthia |

|  |  |  |  |
| --- | --- | --- | --- |
| **Task Name** | **Start** | **Finish** | **Assigned To** |
| Use Case 1: Login User | November 24, 2017 | December 4, 2017 | Tommy,Daniya,Sinthia |
| **1.1 Analysis** | November 24, 2017 | December 4, 2017 | Daniya,Sinthia |
| 1.1.1 Analysis of login form requirement for admin | November 24, 2017 | December 4, 2017 | Daniya,Sinthia |
| **1.2 Design** | November 24, 2017 | December 4, 2017 | Tommy |
| 1.2.1 Design of login form | November 24, 2017 | December 4, 2017 | Tommy |
| **1.3 Implementation** | November 24, 2017 | December 4, 2017 | Tommy |
| 1.3.1 Implementation of login form in website | November 24, 2017 | December 4, 2017 | Tommy |
| **1.4 Testing** | November 24, 2017 | December 4, 2017 | Daniya |
| 1..4.1 Testing of login of user | November 24, 2017 | December 4, 2017 | Daniya |

|  |  |  |  |
| --- | --- | --- | --- |
| **Task Name** | **Start** | **Finish** | **Assigned To** |
| Use Case 1: Register User | November 24, 2017 | December 4, 2017 | Tommy,Daniya,Sinthia |
| **1.1 Analysis** | November 24, 2017 | December 4, 2017 | Tommy, Sinthia |
| 1.1.1 Analysis of register requirement for user | November 24, 2017 | December 4, 2017 | Tommy,Sinthia |
| **1.2 Design** | November 24, 2017 | December 4, 2017 | Daniya |
| 1.2.1 Design of register form | November 24, 2017 | December 4, 2017 | Daniya |
| **1.3 Implementation** | November 24, 2017 | December 4, 2017 | Tommy |
| 1.3.1 Implementation of register form in website | November 24, 2017 | December 4, 2017 | Tommy |
| **1.4 Testing** | November 24, 2017 | December 4, 2017 | Sinthia |
| 1..4.1 Testing of register user | November 24, 2017 | December 4, 2017 | Sinthia |

*Table 3.6.5.1 Iteration Plan Task Breakdown by Workflow*

4.1.6 Resource Summary

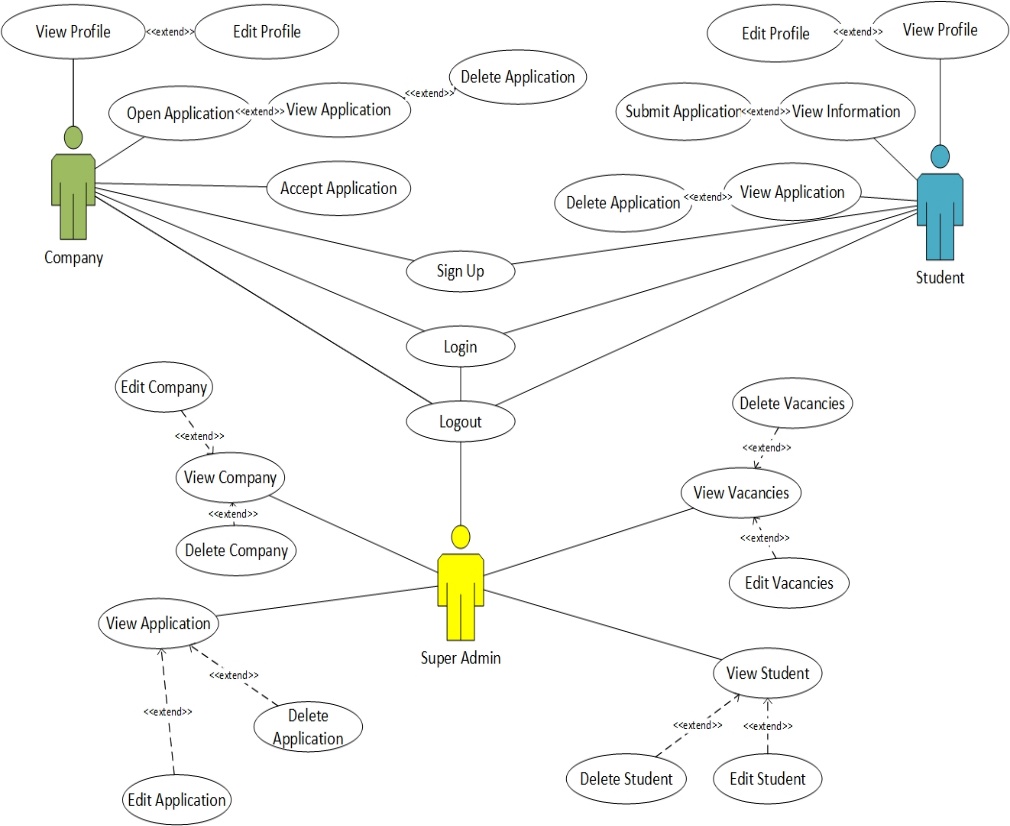
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Resources | Usage | Quantity | Cost |
| Hardware | | | |  |
| 1 | Computer (MacOs Sierra version 10.12.6, Windows 8,  Windows 10,) | Development of web application and testing | 3 computers | Rp0 (Existing computers) |
| Software | | | |  |
| 2 | Sublime Text 3, XAMPP and Adobe Dreamweaver | Development of front end and back end of natural disaster information system | 3 platform | Rp0 ( Open source) |
| 3 | Browser (Chrome, Safari) | Conduct testing to ensure there is no bug in the website and design specification of user and admin has met. | 2 platform | Rp0 |
| Human Resources | | | |  |
| 3 | Team Member | Conduct testing of prototype system | 1 member | Rp0 (Include in team  project) |

*Table 3.6.6.1 Iteration Workflow Schedule*

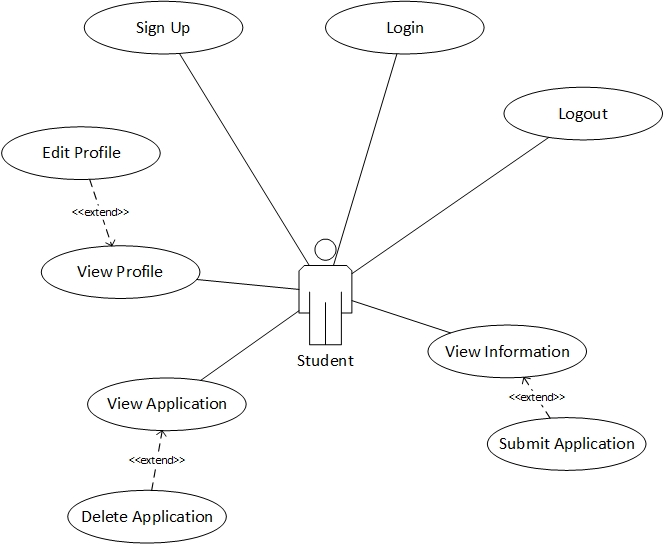
4.1.7 Evaluation Criteria

Based on the methodology of waterfall with prototyping we implement, we have several phase from feasibility study to operation and maintenance stage. In feasibility study phase, we have a criteria to have a clear understanding of specification that we should be performed in our system. Therefore we are able to build up the system and also documentation from the same understanding and miscommunication between team is able to be decreased. The result of the iteration is project background, project aims, objectives, resource plan, interview and observation can be done on time as the milestone we set. In detailed analysis phase, the result of deliverable is risk management, methodology, technique used to develop the system, literature review and requirement gathering. All deliverable has been achieved and finished on time by team member. In detailed design phase, the iteration is implemented to define the appropriate design for user-friendly interface of system and its specification fulfil the user and admin expectations. Fortunately, all design has been completed on the specified time frame based on our project schedule. In construction phase, we have set three use cases that we want to implement in our prototype. There are register and login for user, and login for all admin. All these use cases has been implemented as the criteria of our prototype system. Continued with the testing phase, the criteria of iteration of this phase is when all the use cases implemented and design of website has been fulfilled what we have expected. In testing phase, we have done validation and verification of system and finally can achieve the criteria of testing phase where all use cases has been worked properly. For the system delivery phase and operation and maintenance, we will do it as the future work of BIT 305 or Final Project 2. The evaluation criteria for iteration of both phases, whether our system has fulfilled the expectation of user and admin especially from the disaster recovery agency and operation and maintenance of system can be operated easily by the user and secured the system from the unexpected risk that might attack.

4.1.8 Analysis and Design Artifacts



*Figure 3.1 Full Use Case Diagram*



*Figure 3.1 Use Case Diagram with actor Sudent (Worked by Rivaldo*

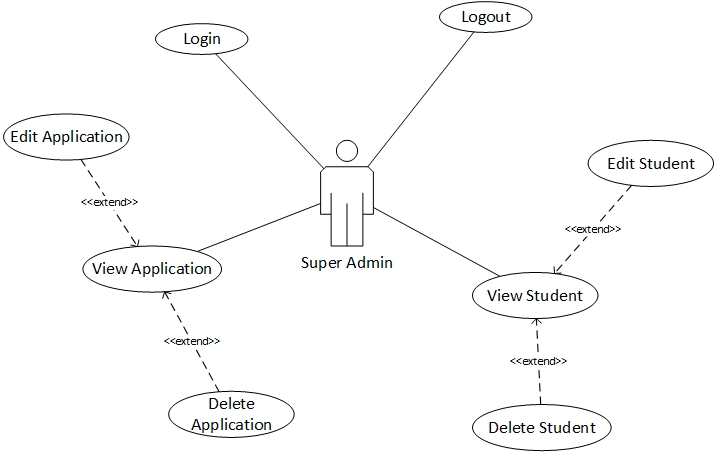


Figure 3.2 Use Case Diagram with Actor Super Admin (Partial & Worked By Aldo)

### 3.3.2 High-Level Use Cases

**High-Level Use Case with Actor Student**

|  |  |
| --- | --- |
| Use Case 1 | Sign up |
| Goal in Context | Student can register to the system |
| Primary Actors  Secondary Actors | Student  - |
| Description | Student can register to the system by providing email address, password, and other personal information for user’s profile. |

Table 3.1 High Level Use-Case of Sign up with actor Student

|  |  |
| --- | --- |
| Use Case 2 | Login |
| Goal in Context | Student can login to the application |
| Primary Actors  Secondary Actors | Student  - |
| Description | Student have to enter valid username and password to login to the website application. |

Table 3.2 High Level Use-Case of Login with actor Student

|  |  |
| --- | --- |
| Use Case 3 | Logout |
| Goal in Context | Student end their session from the application, redirected to the guest homepage. |
| Primary Actors  Secondary Actors | Student  - |
| Description | Student wants to end their session of using the application, by clicking the log out button. After that, student will be redirected to the guest homepage. |

Table 3.3 High Level Use-Case of Logout with actor Student

|  |  |
| --- | --- |
| Use Case 4 | View Profile |
| Goal in Context | Student can see his/her profile page. |
| Primary Actors  Secondary Actors | Student  - |
| Description | Student can see his/her profile page, containing their profile pictures, personal information such as names, birthdates, school, course, etc. |

Table 3.4 High Level Use-Case of View Profile with actor Student

|  |  |
| --- | --- |
| Use Case 5 | Edit Profile |
| Goal in Context | Student can change his/her profile details. |
| Primary Actors  Secondary Actors | Student  - |
| Description | Student can change his / her profile details, such as profile picture, name, birthdate, etc. Student |

Table 3.5 High Level Use-Case of Edit Profile with actor Student

|  |  |
| --- | --- |
| Use Case 6 | View Information (Dashboard) |
| Goal in Context | Student can view information about vacancy of academic internships. |
| Primary Actors  Secondary Actors | Student  - |
| Description | Student access and view information about vacancy of academic internships. Students can find the information according to their preferences, and filter (or sort) them. |

Table 3.6 High Level Use-Case of View Information (Dashboard**)** with actor Student

|  |  |
| --- | --- |
| Use Case 7 | View Application |
| Goal in Context | Student can view lists of his/her application submitted to the internship. |
| Primary Actors  Secondary Actors | Student  - |
| Description | Student can view lists of his/her application submitted to the internships of the companies. Student can see the details and status of his/her application. |

Table 3.7 High Level Use-Case of View Applicationwith actor Student

|  |  |
| --- | --- |
| Use Case 8 | Submit Application |
| Goal in Context | Student can apply (submit) for the academic internship. |
| Primary Actors  Secondary Actors | Student  - |
| Description | Student can apply (submit) for the academic internship that is already chosen by him/her. Student will be needed to fill certain form or following certain steps according to the system or companies’ policy. |

*Table 3.8 High Level Use-Case of Submit Application with actor Student*

|  |  |
| --- | --- |
| Use Case 9 | Delete Application |
| Goal in Context | Student can cancel (remove) the application that has already submitted. |
| Primary Actors  Secondary Actors | Student  - |
| Description | Student can cancel (remove) his/her application submitted to the selected company. This will need time for verification as a prevention from unauthorized access or human error to make sure that this is a conscience decision. Removing the application means immediate cancellation to the application. |

*Table 3.9 High Level Use-Case of Delete Application with actor Student*

**High-Level Use Case with Actor Super Admin**

***(Partial Use Case developed individually by Rivaldo)***

|  |  |
| --- | --- |
| Use Case 10 | Login |
| Goal in Context | Super Admin can login to the application |
| Primary Actors  Secondary Actors | Super Admin  - |
| Description | Super Admin have to enter valid username and password to login to the website application. |

Table 3.10 High Level Use-Case of Login with actor Super Admin

|  |  |
| --- | --- |
| Use Case 11 | Logout |
| Goal in Context | Super Admin end their session from the application, redirected to the login page. |
| Primary Actors  Secondary Actors | Super Admin - |
| Description | Super Admin wants to end their session of using the application, by clicking the log out button. After that, student will be redirected to the login page. |

Table 3.11 High Level Use-Case of Logout with actor Super Admin

|  |  |
| --- | --- |
| Use Case 12 | View Student |
| Goal in Context | Super Admin can see lists of registered Student object. |
| Primary Actors  Secondary Actors | Super Admin  - |
| Description | Super Admin can see lists of registered Student Object in the system. |

Table 3.12 High Level Use-Case of View Student with actor Super Admin

|  |  |
| --- | --- |
| Use Case 13 | Edit Student |
| Goal in Context | Super Admin can change Student’s personal detail. |
| Primary Actors  Secondary Actors | Super Admin  - |
| Description | Super Admin can Student’s personal detail such as profile picture, name, birthdate, etc. This is done in respond to request, error / failures happen in the front-end as Student cannot change on his/her own. |

Table 3.13 High Level Use-Case of Edit Student with actor Super Admin

|  |  |
| --- | --- |
| Use Case 14 | Delete Student |
| Goal in Context | Super Admin remove (delete) Student object |
| Primary Actors  Secondary Actors | Super Admin  - |
| Description | Super Admin can remove (delete) Student object registered in system. This is done in respond to request, errors / failures happen as some account may come error or invalid, requiring new registered account. This also necessary to delete old unused accounts. |

Table 3.14 High Level Use-Case of Delete Student with actor Super Admin

|  |  |
| --- | --- |
| Use Case 15 | View Application |
| Goal in Context | Super Admin can see lists of registered Application objects (From Student). |
| Primary Actors  Secondary Actors | Super Admin  - |
| Description | Super Admin can see lists of registered Application objects (from Student) in the system. |

Table 3.15 High Level Use-Case of View Application with actor Super Admin

|  |  |
| --- | --- |
| Use Case 16 | Edit Application |
| Goal in Context | Super Admin can change Student’s Application details. |
| Primary Actors  Secondary Actors | Super Admin  - |
| Description | Super Admin can Student’s Application details |

Table 3.16 High Level Use-Case of Edit Application with actor Super Admin

|  |  |
| --- | --- |
| Use Case 17 | Delete Application |
| Goal in Context | Super Admin remove (delete) Application object. |
| Primary Actors  Secondary Actors | Super Admin  - |
| Description | Super Admin can remove (delete) Application object registered in system. This is done in respond to request, errors / failures happen as some account may come error or invalid, requiring new submission of Application Object. This action can also be performed by a request from Student for Application deletion. |

Table 3.17 High Level Use-Case of Delete Application with actor Super Admin

### 3.3.3 Expanded Use Cases

**Expanded Use Case with Actor Student**

|  |  |
| --- | --- |
| **Use Case Name** | **Sign Up** |
| **Goal in Context** | Student can register to the system. |
| **Primary Actor**  **Secondary Actor** | Student  - |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. Begin with Student visiting the website. Student clicks sign up. | 1. System display sign up form |
| 1. Student fills the form with necessary details. Student clicks sign up button. | 1. System validates and registering the Student object with filled form. |
|  | 1. System will display “Success”, then redirect to login page. |
| **Alternatives Course of Events** | |
| 1. If Student fails to sign because of invalid data registered, Student will be asked to re-enter the sign up form with valid data (valid birth date, valid name, valid student ID from the school). | |

Table 3.18 Expanded Use-Case of Sign Up with actor Student

|  |  |
| --- | --- |
| **Use Case Name** | **Login** |
| **Goal in Context** | Student can login to the website application |
| **Primary Actor**  **Secondary Actor** | Student  - |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. Begin with student accessing the login page | 1. System displays the login form. |
| 1. Student fills out the login form by entering the default username and password. Student clicks the login button. | 1. System checks the data that already filled in the login form to match the data in the database. |
|  | 1. System will display the dashboard page, if Student is a valid user. |
| **Alternatives Course of Events** | |
| 1. If the user input invalid username or password, the system will display a notification. | |

Table 3.19 Expanded Use-Case of Login with actor Student

|  |  |
| --- | --- |
| **Use Case Name** | **Logout** |
| **Goal in Context** | Student can end the session (log out) from the website. |
| **Primary Actor**  **Secondary Actor** | Student  - |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. Begin with Student clicks Log out Button. | 1. System is processing by terminating the session. Sessions made by last login will be logged. |
|  | 1. System will redirect Student to guest homepage. |
| **Alternatives Course of Events** | |
| - | |

Table 3.20 Expanded Use-Case of Log out with actor Student

|  |  |
| --- | --- |
| **Use Case Name** | **View Profile** |
| **Goal in Context** | Student can view him/her profile page. |
| **Primary Actor**  **Secondary Actor** | Student  - |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. Begin with Student click “My Profile” | 1. System displays My Profile page. |
| **Alternatives Course of Events** | |
| - | |

Table 3.21 Expanded Use-Case of View Profile with actor Student

|  |  |
| --- | --- |
| **Use Case Name** | **Edit Profile** |
| **Goal in Context** | Student can change details on My Profile page. |
| **Primary Actor**  **Secondary Actor** | Student  - |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. Begin with Student click My Profile page. | 1. System displays My Profile page. |
| 1. Student click “Edit Profile” | 1. System displays the edit form. |
| 1. Student change the details according to his/her preferences. After that, Student click “Apply” | 1. System displays confirmation message before submitting the changes. |
| 1. Student confirms by click “Yes” to change the details. | 1. System submits changes to the database. |
|  | 1. The system will display notification (success or fail) of the change result. |
| **Alternatives Course of Events** | |
| 1. Changes will not be submitted if Student cancelled the confirmation. 2. Changes will be aborted if the data inputted was invalid. | |

Table 3.22 Expanded Use-Case of Edit Profile with actor Student

|  |  |
| --- | --- |
| **Use Case Name** | **View Information** |
| **Goal in Context** | Student can view the information of academic internship. |
| **Primary Actor**  **Secondary Actor** | Student  - |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. Begin with Student clicking “Home” (Dashboard” | 1. System displays information about academic internship immediately within Home (Dashboard). |
| **Alternatives Course of Events** | |
| - | |

Table 3.23 Expanded Use-Case of View Information with actor Student

|  |  |
| --- | --- |
| **Use Case Name** | **Submit Application** |
| **Goal in Context** | Student applies (submits) application to the preferred company for his/her academic internship. |
| **Primary Actor**  **Secondary Actor** | Student  - |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. Begins with Student click View Information | 1. System displays information about academic internship immediately within Home (Dashboard). |
| 1. Student then clicks the preferred vacancy / company. | 1. System displays the detail of the vacancy and the company. |
| 1. Student then clicks “Submit Application”. | 1. System displays forms for the application. |
| 1. Student fills the form with necessary data. Clicks “Apply Submit” | 1. System displays confirmation before submitting the application. |
| 1. Student confirmed by Clicking ‘Yes” | 1. Systems accept the application and save it to database. Company will be notified. |
|  | 1. System will display result (“Success” or “Fail”) with a notification to Student. |
| **Alternatives Course of Events** | |
| 1. The use case will be cancelled if Student choose “No” | |

Table 3.24 Expanded Use-Case of Submit Application with actor Student

|  |  |
| --- | --- |
| **Use Case Name** | **Delete Application** |
| **Goal in Context** | Student can remove (delete) submitted application by requesting it. |
| **Primary Actor**  **Secondary Actor** | Student  Super Admin |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. Begin with Student click “View Application” | 1. System displays list of application submitted by Student. |
| 1. Student choose application to be removed by clicking “Delete” onto it. | 1. System displays confirmation before deleting the application. |
| 1. Student confirmed by Clicking “Yes” | 1. System saves the request to the database. Super Admin will be notified. |
|  | 1. System will display result (“Success” or “Fail”) with a notification to Student. |
| 1. Super Admin gets the notification, then go to View Application. | 1. System displays all of the lists of Application. |
| 1. By the reference ID (ApplicationID) of the request, Super Admin deletes the application by clicking “Delete” onto it. | 1. System displays confirmation before deleting the application. |
| 1. Super Admin confirmed by clicking “Yes” | 1. System remove the selected application from the database. Student will be notified. |
|  | 1. System will display result (“Success” or “Fail”) with a notification to Student. |
| **Alternatives Course of Events** | |
| 1. The use case will be cancelled if Student choose “No” 2. The use case will be cancelled if Super Admin choose “No” | |

Table 3.25 Expanded Use-Case of Delete Application with actor Student

**Expanded Use Case with Actor Super Admin**

***(Partial Use Case developed individually by Rivaldo)***

|  |  |
| --- | --- |
| **Use Case Name** | **Login** |
| **Goal in Context** | Super Admin can login to the website application |
| **Primary Actor**  **Secondary Actor** | Super Admin  - |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. Begin with Super Admin accessing the login page | 1. System displays the login form. |
| 1. Super Admin fills out the login form by entering the default username and password. Super Admin clicks the login button. | 1. System checks the data that already filled in the login form to match the data in the database. |
|  | 1. System will display the dashboard page, if Super Admin is a valid user. |
| **Alternatives Course of Events** | |
| 1. If the user input invalid username or password, the system will display a notification. | |

Table 3.26 Expanded Use-Case of Login with actor Super Admin

|  |  |
| --- | --- |
| **Use Case Name** | **Logout** |
| **Goal in Context** | Super Admin can end the session (log out) from the website. |
| **Primary Actor**  **Secondary Actor** | Super Admin - |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. Begins with Super Admin clicks Log out Button. | 1. System is processing by terminating the session. Sessions made by last login will be logged. |
|  | 1. System will redirect Super Admin to login page. |
| **Alternatives Course of Events** | |
| - | |

Table 3.27 Expanded Use-Case of Log out with actor Super Admin

|  |  |
| --- | --- |
| **Use Case Name** | **View Student** |
| **Goal in Context** | Super Admin can see lists of registered Student Object. |
| **Primary Actor**  **Secondary Actor** | Super Admin  - |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. Begin with Super Admin clicks “View Student” | 1. System displays list of registered Student Object. |
| **Alternatives Course of Events** | |
| - | |

Table 3.28 Expanded Use-Case of View Student with actor Super Admin

|  |  |
| --- | --- |
| **Use Case Name** | **Edit Student** |
| **Goal in Context** | Super Admin can change details on Student Object. |
| **Primary Actor**  **Secondary Actor** | Super Admin  - |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. Begin with Super Admin clicks “View Student” | 1. System displays list of registered Student Object. |
| 1. Super Admin clicks “Edit” on selected Student Object. | 1. System displays form for the Student Object. |
| 1. Super Admin change the details according to requests. Super Admin click “Apply” | 1. System displays confirmation message before submitting the changes. |
| 1. Super Admin confirms by click “Yes” to change the details. | 1. System submits changes to the database. |
|  | 1. The system will display notification (success or fail) of the change result. |
| **Alternatives Course of Events** | |
| 1. Changes will not be submitted if Super Admin cancelled the confirmation. 2. Changes will be aborted if the data inputted was invalid. | |

Table 3.29 Expanded Use-Case of Edit Student with actor Super Admin

|  |  |
| --- | --- |
| **Use Case Name** | **Delete Student** |
| **Goal in Context** | Super Admin can remove (delete) Student Object from the system. |
| **Primary Actor**  **Secondary Actor** | Super Admin  - |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
|  |  |
| 1. Begin with Super Admin gets notification for deletion or decision of policy. Super Admin clicks “View Student” | 1. System displays list of registered Student Object. |
| 1. Super Admin clicks “Delete” on selected Student Object (as requested by notification, or as decided). | 1. System displays confirmation before deleting Student Object. |
| 1. Super Admin confirmed by clicking “Yes” | 1. System remove the selected Student Object from the database. |
|  | 1. The system will display notification of the deletion result. |
| **Alternatives Course of Events** | |
| 1. Process will be aborted if Super Admin choosed “No” | |

Table 3.30 Expanded Use-Case of Delete Student with actor Super Admin

|  |  |
| --- | --- |
| **Use Case Name** | **View Application** |
| **Goal in Context** | Super Admin can see lists of registered Application Object. |
| **Primary Actor**  **Secondary Actor** | Super Admin  - |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. Begin with Super Admin clicks “View Application” | 1. System displays list of registered Application Object. |
| **Alternatives Course of Events** | |
| - | |

Table 3.31 Expanded Use-Case of View Application with actor Super Admin

|  |  |
| --- | --- |
| **Use Case Name** | **Edit Application** |
| **Goal in Context** | Super Admin can change details on Application Object. |
| **Primary Actor**  **Secondary Actor** | Super Admin  - |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. Begin with Super Admin clicks “View Application” | 1. System displays list of registered Application Object. |
| 1. Super Admin clicks “Edit” on selected Application Object. | 1. System displays form for the Application Object. |
| 1. Super Admin change the details according to requests. Super Admin click “Apply” | 1. System displays confirmation message before submitting the changes. |
| 1. Super Admin confirms by click “Yes” to change the details. | 1. System submits changes to the database. |
|  | 1. The system will display notification (success or fail) of the change result. |
| **Alternatives Course of Events** | |
| 1. Changes will not be submitted if Super Admin cancelled the confirmation. 2. Changes will be aborted if the data inputted was invalid. | |

Table 3.32 Expanded Use-Case of Edit Application with actor Super Admin

|  |  |
| --- | --- |
| **Use Case Name** | **Delete Application** |
| **Goal in Context** | Super Admin can remove (delete) Application Object from the system. |
| **Primary Actor**  **Secondary Actor** | Super Admin  Student |
| **Typical Course of Events** | |
| **Actor Action** | **System Response** |
| 1. Begin with Super Admin gets notification for deletion from Student or decision of policy. Super Admin clicks “View Application” | 1. System displays list of registered Application Object. |
| 1. Super Admin clicks “Delete” on selected Application Object (as requested by notification, or as decided). | 1. System displays confirmation before deleting the application. |
| 1. Super Admin confirmed by clicking “Yes” | 1. System remove the selected Application Object from the database. |
|  | 1. The system will display notification of the deletion result. |
|  | 1. System will notify Student for the deletion result. |
| **Alternatives Course of Events** | |
| 1. Process will be aborted if Super Admin choosed “No” | |

Table 3.33 Expanded Use-Case of Delete Student with actor Super Admin

4.2.9 Implementation and Testing 8

4.2.10 Iteration Review and Evaluation 10

Iteration of each phase except for system delivery and operation & maintenance has been met the specified time frame we set. Iteration plan has helped us in determining the requirement and specification of system. Started from the feasibility study to testing phase, the result has been in accordance with our expectation. Eventhough there is some bug that emerge during the testing phase iteration but fortunately we can overcome it and finished the protoype on time.

**4.3 Iteration 2 11**

**4.4 Iteration 3 11**