

**BIT304 FINAL YEAR PROJECT I**

**REQUIREMENT ANALYSIS**

**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

## 3.1 Introduction

Requirements analysis, as it relates to software projects, is the process of studying, determining and documenting user needs and expectations of the software system to be designed that solves a particular problem (Catanio, 2006). It is imperative to have requirement analysis in our project for the purpose of requirements and development course, helping to identify cases, features that are need to be developed during the project.

In this chapter, we will discuss the requirement summary, functional requirement, and non-functional requirement:

* The requirement summary contains the summarization of system requirements, explaining about the users, functions, and input-output of the system.
* The functional requirement defines the functional of the system explained in the form of use case diagram, while the detail explanation will be in high-level use case and expanded use case.
* The non-functional requirement consists of technical requirement, usability requirement, reliability requirement, and security requirement of the system.

## 3.2 Requirement Summary

Ayo Magang is a web-based application for providing information of vacancy of academic internship for vocational high school and college students. The web is developed using PHP Native and Bootstrap with responsive-design. The main purpose of this web application is to provide easy way for students to find information about vacancy of academic internship on existing companies, helping students to help desired place to exercise with. This will be very helpful for student with low networking skill, or low soft-skill that does not have friends or connection with people. This is also an opportunity for companies to gain benefit in branding by opening chances for academic internship, creating mutual business relationships with many education institute, whether by providing perspective employee through the academic internship, or by doing a joint project between both parties.

This website application has three users:

* The student, who wants to find information of vacancy for academic internship.
* The company, who opens vacancy for academic internship and accepting students’ application.
* Super Admin. The master user that controls and over watch everything inside the system, having the basic CRUD to perform on all users.

There are some main function provided by the website application that can be used by the various user, those features are as follows:

* View Information

Students can see various information of academic internships opening in the website. Students can filter or sort the information by their preferred course or location of the companies.

* Open Application (Internship Vacancy)

Companies can sign up to the website, open an application for academic internship to recruit students. Companies can describe what they are working on, what kind of project or jobs handled by company, and requirements for the qualification for whether the internship or the real job.

* Submit Application (Applying to the company)

Students can sign up to the website, find the preferred company, and then apply or submit an application for the academic internship on the company.

## 3.3 Functional Requirement

Functional Requirement defines the functional of the system, divided as follow:

1. Student
2. Sign Up

This function will let student to register to the system. User will need to provide email address and password for the login. Then after a successful register, system will prompt user to provide username and other details regarding to the user profile. This will be needed as a basic information for system to recommend companies that are correlated with the course (profile) filled by user, easier way to help student find information for their internship. This stage also needs agreement of user about the privacy policy regarding asking and storing sensitive data of the users (Name, Birthdate, etc.)

1. Login

This function will let student to log in to the system. User will be redirected straight to dashboard to see vacancy or opening for academic internship from companies existing in the system.

1. View Profile

This function will let student to view their profile in the system. Student can see their profile picture, username, and other personal information in the system.

1. Edit Profile

This function will let student to perform changes on their profile details, user change their personal information, customize profile with CV, resume or profile pictures.

1. View Information (Dashboard / Search)

This function will let student to search, find for available academic internships. This is an indirect form of Dashboard. In the Dashboard, student can directly scrolls for the information. Alternatively, if user wants to filter or sort the information based on certain category (course, location, or profile), user just have to click “Search” button and try to find the desired company by keywords, location, courses, or recommendation judging from your profile.

1. View Application

This function will let student to view list of application applied by students to their selected companies.

1. Submit Application

This function will let student to apply (submit) an application for their academic internship to the desired company. Student will be required to fill certain data for the need for the recruitment.

1. Delete Application

This function will let student to cancel (delete) an existing application for the internship. This allows student to cancel their application if there is any unexpected event happen in the future, or because the student has already accepted by other company, which means that the student must cancel other application that has already submitted to other company. This will not action as an absolute action, but as a request of deletion, that later will be performed by Super Admin

1. Logout

This function will let student to left from the website (exit / logout). After the logout, student will be redirected to the guest homepage of the website.

1. Super Admin (Full explanation, Partial work)
2. Login

This function will let Super Admin to log in to the system. Student will be redirected straight to dashboard, containing general information and basic CRUD Operations overall system.

1. Logout

This function will let Super Admin to log out from the system. After that, Super Admin will be redirected to the login page.

1. View Student

This function will let Super Admin to see lists of registered Student objects in the system.

1. Edit Student

This function will let Super Admin to perform changes on Student object’s detail, such as personal information, or password. This is a useful feature if there is an unexpected error or failure, creating anomaly caused by changes performed by Students during profile editing. This option can overwrite the changes.

1. Delete Student

This function will let Super Admin to delete an existing Student object from the system. This will be useful if there’s any error or failure happen in the Student Object, causing the user object to be invalid, meaning Super Admin can also help by reset the data, and user can re-register their selves. This can also used to delete old accounts (or spam).

1. View Application

This function will let Super Admin to see lists of registered Application (from Student) objects in the system.

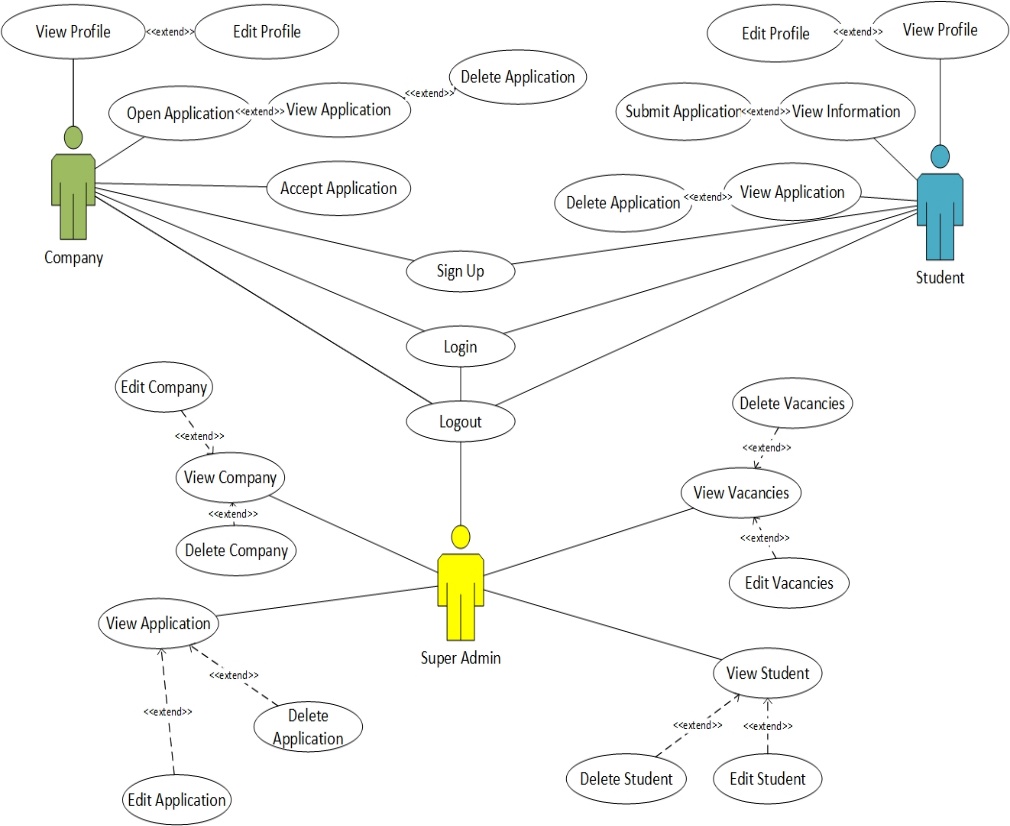
1. Edit Application

This function will let Super Admin to perform changes on Application object’s details.

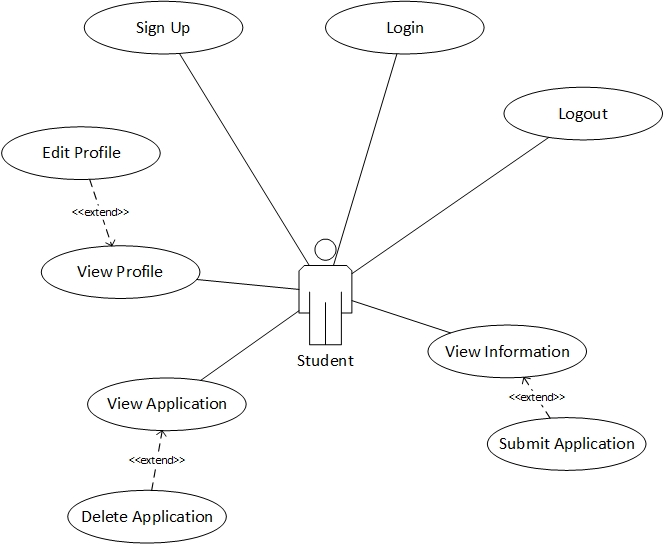
1. Delete Application

This function will let Super Admin to delete an existing Application object (from student) from the system. This will be useful if there’s any error or failure happen in the Application Object, or if the Application is invalid because of breaking policy. This can also be an action based on deletion request by Student.

### 3.3.1 Use Case Diagram



*Figure 3.1 Full Use Case Diagram*



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

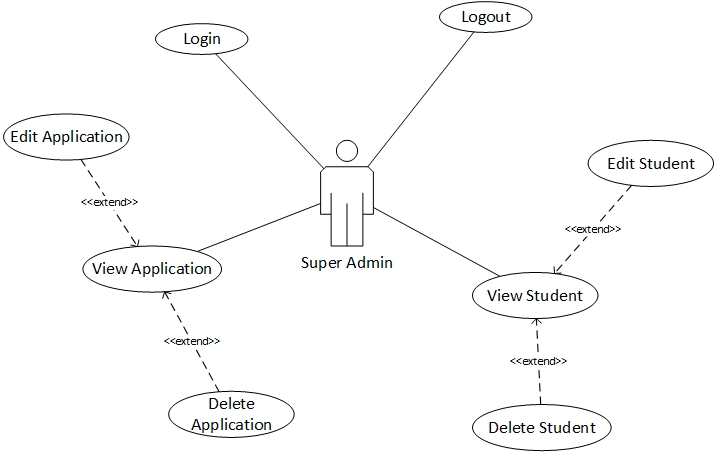


Figure 3.2 Use Case Diagram with Actor Super Admin (Partial & Worked By Aldo)

### 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

## 3.4 Non-Functional Requirements

### 3.4.1 Technical Requirements

To conduct development, testing, and implementation optimally, there are some technical requirement to be fulfilled. The requirements will be based on technical standpoint.

1. Technical Requirement for Developer

In developing the website application, we will use PHP Native as programming language and HTML, CSS, and Bootstrap as the companion for the design on user interface. Here are some technical requirement on software and hardware:

* Software:
* Operating System

PHP-based website application can be installed in various operating system. Linux, Mac OS, and Microsoft Windows can be used to develop the website application. It is recommended to use Linux Ubuntu 18.04 or higher, Mac OS 10.12 Sierra or higher, and Microsoft Windows 8.1 or higher.

* Text Editor

We used numbers of text editors, such as Brackets (primarily), Visual Studio Codes, or even Notepad. Brackets and Visual Studio support various of programming language, such as C, C++, C#, Python Java, PHP, etc. We choose Brackets and Visual Studio Code because for its flexibility and versatility.

* Server

We used XAMPP as our server for the system. In XAMPP, we used Apache for the website host and phpMyAdmin for the database host.

* Browser

We used Google Chrome, Mozilla Firefox, and newly developed Microsoft Edge based on Chromium engine. We do this to test compatibility and scalability on each browser.

* Hardware
* Processor

The minimum requirement for CPU will be sufficient for 3rd Gen Core i3 series (i3-3xxx), or AMD A8-6600K Richland Gen.

* RAM

4GB of RAM will the lowest threshold for the minimum. It is recommended to get 8GB or more for better experience in developing the website, as bigger memory will store more work for the development.

* Storage

It is recommended to have 6 GB of free storage for the sake of space convenient. It is also recommended to use SSD instead of HDD, as it will greatly increase system speed and response, better experience in developing.

* Network Interface

A Fast Ethernet or Wireless Connection with standard protocol (802.11 a/b/g/n) will suffice the need for the development to be connected to internet. Gigabit Ethernet and WLAN with 802.11 ac protocol will be recommended for low latency and fast update for development with local network or internet.

1. Technical Requirement for User

It is obvious that user will need to meet at least the minimum requirement for a good experience of using the website application. Those requirements are as follows:

* Software:
* Operating System

Linux, Mac OS, and Microsoft Windows can be used to access the website application. It is recommended to use Linux Ubuntu 18.04 or higher, Mac OS 10.12 Sierra or higher, and Microsoft Windows 7 or higher.

* Browser

User can choose different browsers such as Google Chrome, Mozilla Firefox, and newly developed Microsoft Edge based on Chromium engine. In order for the website to run properly, make sure to check updates regularly for your browser for the sake of compatibility, performance, and security.

* Hardware
* Processor

The minimum requirement for CPU will be sufficient for 3rd Gen Core i3 series (i3-3xxx), or AMD A8-6600K Richland Gen.

* RAM

4GB of RAM will the lowest threshold for the minimum. It is recommended to get 8GB or more for better experience in running the website.

* Storage

It is recommended to have 2 GB of free storage for the sake of space convenient (caching, etc.). It is also recommended to use SSD instead of HDD, as it will greatly increase system speed and response, better experience in surfing the website application.

* Network Interface

A Fast Ethernet or Wireless Connection with standard protocol (802.11 a/b/g/n) will suffice the need for the accessing the website. Gigabit Ethernet and WLAN with 802.11 ac protocol will be recommended for low latency and fast access.

### 3.4.2 Usability Requirements

Usability requirements is all about how the application will be used. In essence, easy and simple is the target of the design of the website. This will have impact of how User Interface and User Experience (UI / UX) affect user. We want a simple design that is easy to understand, and easy to use. All the icons, images, texts, and other property will be aligned and configured properly. We will CSS and Bootstrap as our companion for designing the website. We also want the design to be responsive, means that the website can be opened in any screen size, and each screen size will have its own adaption of User Interface and User Experience.

### 3.4.3 Reliability Requirements

Reliability requirements in its core means that the website we developed must be reliable. User can trust the website to be robust, smooth, compatible, and easy to use. We also make sure the website to be as light as possible, but at the same time, the application will do its full function without burden the system. Imperatively, the website should give a bright and clear of information that will be displayed, so user can easily understand the information that is trying to be delivered.

### 3.4.4 Maintainability Requirements

Changes, patches, updates, or even reboot (re-initiate) in a system is an unavoidable event. In essence, we also need to think about the future development of the application. During time, technology will be developed better, design mood changes, trending will be different from time to time. With this many changes in the future, we must be ready for adaptation and improvement. Also at the same time, we do not lose the authentic image that we have on the website. Maintainability a very serious subject to be planned and developed to the future.

We can start to do this by doing surveys, feedbacks, Q&As, closed and open betas to public. With this, we can get much more feedback for the better things for development. We can also do an open-source involving developer and selected end-user to do further more testing and evaluation regarding the development of the website application. We can detect unexpected errors, bugs, failures, holes, vulnerabilities, and places for improvement. This will ensure the website application will always be up to date, in order to maintain to get the best performance.