INTERNATIONAL STUDENT ADMISSION SYSTEM DESIGNED AS A DYNAMIC WORKFLOW FOR WEB AND MOBILE

ABDULRAHMAN MOHAMMED AYASH YOUSEF

SESSION 2019/2020

FACULTY OF COMPUTING AND INFORMATICS MULTIMEDIA UNIVERSITY
FEBRUARY 2020

INTERNATIONAL STUDENT ADMISSION SYSTEM DESIGNED AS A DYNAMIC WORKFLOW FOR WEB AND MOBILE

BY

ABDULRAHMAN MOHAMMED AYASH YOUSEF

SESSION
2019/2020

THIS PROJECT REPORT
PREPARED FOR

FACULTY OF COMPUTING AND INFORMATICS
MULTIMEDIA UNIVERSITY
IN PARTIAL FULFILLMENT
FOR

BACHELOR OF COMPUTER SCIENCE (HONS)
WITH SPECIALIZATION IN SOFTWARE ENGINEERING

FACULTY OF COMPUTING AND INFORMATICS
MULTIMEDIA UNIVERSITY
FEBRUARY 2020

DECLARATION

I hereby declare that the work has been done by myself and no portion of the work contained in this thesis has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

_____

Abdulrahman Mohammed Ayash Yousef
Faculty of Computing & Informatics
Multimedia University
Date: 12: 02: 2020

ACKNOWLEDGMENT

I would like to express my utmost gratitude to my mother and to the memory of my father, to them I will forever be grateful. I would like to also express my deepest appreciation to everyone who supported me the throughout writing of this thesis. Namely, my project supervisor and my mentor Mr. Sharaf Al-Horani. Mr. Sharaf has been there for me since day one by helping me decide my project title even from the previous trimester until the last day of writing this thesis. Furthermore, a special thanks to Dr. Wan Noorshahida Binti Mohd Isa for her countless consultations throughout my degree; providing me with informed insights that resulted in confident decision making. Also, a very special thanks goes to my uncle Ahmed Rajab and my friend Abdulrahman Ahmad Al-Saud, without them I could have never even begin to think about achieving this dream. Last but definitely not least, all the love to my family and friends for their unlimited support and confidence in my ability to carry out this massive task.

ABSTRACT

Admission for international students is a big hurdle that's only a part of the journey. International students have to leave their whole lives behind them in order to attempt to pave their future. Considering the substantial screen time spent on mobile to contribute in the easement of admission, a mobile application for international students was developed as the first part of the system.

The lack of transparency and delay in response faced by students result in an unsettling fear because students tend to think that their application is ignored, therefore, they resolve to apply to other places that can potentially respond faster, ultimately universities are losing those potential international students.

Although, the process of admission itself is an integral element, the performance of the process is not being measured properly. To ensure proper adaptability and competitive relevance for universities, a website was developed for the management as the second part of the system. The website was designed to facilitate the creation of a dynamic workflow. As a result, the management has the complete freedom for future adjustments in the workflow. This design approach enables the management to control different process stages; this enables measuring and improving performance and efficiency. Also, the system has periodic statistics and indicators that provide integral data vital for the information needed to make knowledgeable decisions that result in meaningful improvements, ultimately, ensuring quality standards in performance are to be upheld.

Finally, to provide transparency throughout the experience the system will have an internal messaging subsystem.

Contents

List of figures

List of tables

Project Information

Project Name	International student admission system Designed as a dynamic workflow for web and mobile
Project Manager	Abdulrahman Mohammed Ayash Yousef
Project ID	1407
Supervisor **(MMU Faculty of Computing and Informatics)**	Mr. Sharaf Horani
Official Project Start Date	2nd July 2019
Planned Project End Date	13th February 2020

Abbreviation / glossary

FRS	Functional Requirement Specification
FR	Functional requirement
QR	Quality requirement
UC	Use case
UCS	Use case specification
ID	Identifier
UI	User interface
UML	Unified modelling language
FYP	Final year project

1. Introduction

1.1. Overview:

An international student admission system that consists of a website and a mobile application. The system is designed as a dynamic workflow. The website allows the admin to change the workflow on demand; promoting adaptability to any future adjustments to any stage in the workflow. The website also allows the staff to process student applications. The mobile app allows the student to process their application. Also, the student will never worry about waiting indefinitely without knowing the status of their application; by being able to check their application's progress throughout all stages of the workflow.

The system also has a simple messaging component between the student and staff therefore, it promotes transparency for the student. The messaging system promotes even more transparency in case the student needs to clarify important details. Finally, the system provides important statistics for the admin to help in monitoring the performance and progression of student applications.

1.2. Problem statement

In this digitalized era, most international students will be applying for universities online. The most pivotal information students normally lookup consists of Programmes, Fees, Student Visa requirement details, and the process needed to ensure a successful admission for their application. The inadequacy and lack of transparency in the currently implemented system(s) certainly demotivates the students. This results in universities losing a substantial number of potential students. The following list includes the characteristics that constitutes an inadequate admission system:

- No real dedicated mobile application is available for students. This is pivotal for students because just the thought of applying through a website on a desktop or having to deal with applying from a website using a phone is aggravating.
- No real explanation of the timeline in its entirety. Most of the time the students find themselves waiting indefinitely or missing the immediate next semester because of misunderstanding simple yet important details and deadlines.
- The current admission system(s) caused many international students to look for other institutions of learning due to the delay and lack of transparency. Thinking that they were ignored or their application has been rejected.
- Performance of the process of admission is not being measured properly. Which results in a process that is stagnant and no real way of improvement for the management.
- No real statistics and meaningful indicators that provide integral data is available for management resulting in low quality attempts for improvements.
- The confusion caused by the unorganized spread of pivotal information throughout a typical website.
- The confusion caused by the variety of systems in a typical online admission system; most systems start with simple forms and they turn into a typical exchange of emails down the line.

Developing a solution that solves the aforementioned problems by building a project that consists of an application for the student and a website for the management which will make the entire process measurable, transparent, adaptable, and to attempt to

ensure that no international student is ever demotivated, forced to push for the next semester, or is indefinitely waiting while applying.

1.3. Objectives

The purpose of this project is to develop a website and a mobile application that overcomes the weaknesses and shorting comings of current International student admission systems. The objectives of this project are as follows:

- To ensure that universities do not lose a substantial amount of potential international students by improving the transparency and response time.
- To provide a modifiable dynamic workflow for the management that would result in improvements at any stage of the process. Because if a certain stage is not meeting performance standards, the management can change the stage itself or improve the processes inside the stage.
- To Provide statistics on demand for the management. This will improve the performance because the management will be informed about indicators like the average process time for an application, throughput of individual staff, number of delays and root cause of delays in the process.
- To provide a simple messaging system for clarifications between student and staff. This would greatly improve transparency.
- To provide up to date application status to give students a peace of mind.

1.4. Goals

This project aims to design a practical system that benefits all relevant parties; by promoting productivity that in turn improves marketability of universities. Resulting in an increase in the number of admitted students. Some of the sub goals are as follows:

- To periodically improve on the workflow and the performance to ensure quality
- To simplify the admission process for students.
- To increase transparency throughout the admission process for students
- To improve communications by omitting the need for typical email exchanges.

1.5. Project scope

This project consists of developing an international student admission system as both website and mobile. The admin is able to modify the workflow and the staff is able to process the students' applications from the website. The mobile app is developed for Android mobile devices for students.

1.6. Project schedule

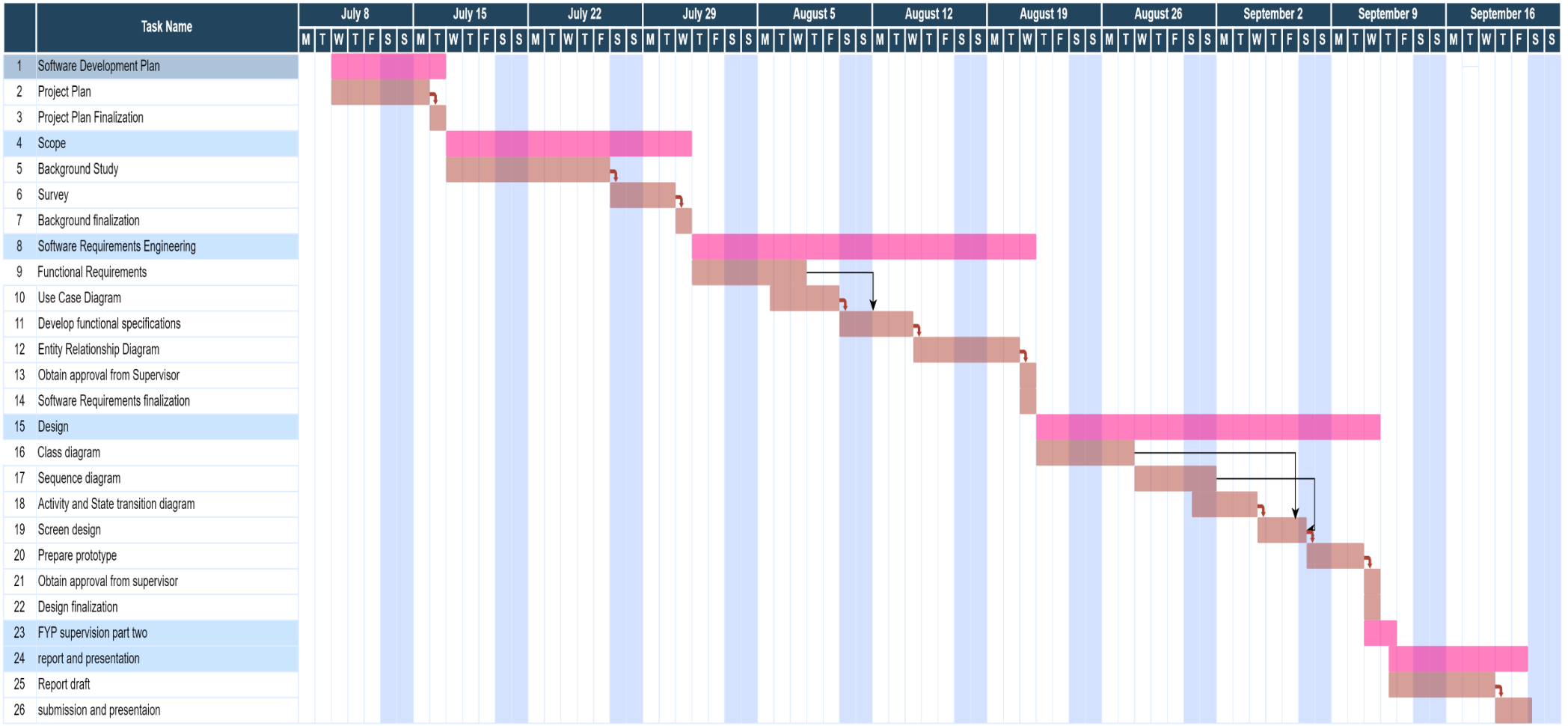


Figure 1.5-1 Gantt chart

1.6.1. Gantt chart description

1.6.1.1. Software development plan

The bigger picture for the entirety of the plan for final year project part (1) and project justification with some general goals are discussed.

1.6.1.2. Scope

Background studies on existing technologies, deployed websites, and deployed services that are relevant to the scope. A survey to have better views on information was discussed and deployed.

1.6.1.3. Software requirements engineering

Software requirements for the project is obtained. Additionally, using the requirements, a use case diagram and an entity relational diagram are sketched out.

Requirements were collected, depicted in diagrams and tables for further analysis and investigation. Resulting in, use case diagrams and ERD.

1.6.1.4. Design

Technical documents and diagrams were discussed to satisfy the requirements taken directly from the requirements analysis. Technical documents include sequence diagrams, activity diagrams, and state transition diagram.

1.6.1.5. FYP Supervision Part Two

The bigger picture for the entirety of the plan for final year project part (2) is discussed and depicted in a Gantt chart

1.6.1.6. Report

This thesis is formatted and put together to satisfy the required Interim Report that has all the documentation for the project stages.

1.6.2. Milestone Table

Table 1.6-1 Milestones

Project Plan	10th July 2019
Background Study	25th July 2019
Requirements documentation	1st August 2019
Use case diagram	4th August 2019
Entity relationship diagram	15th August 2019
Class diagram	22nd August 2019
Sequence diagram	26th August 2019
Activity and state diagram	29th August 2019
Screen Design	1st September 2019
Development of prototype	5th September 2019
Final report draft	13th September 2019
Report submission	16th September 2019
presentation	25th September 2019

2. Background study

2.1. Existing dynamic workflow systems

2.1.1. ProcessMaker

Process maker is a multi-platform service to simplify workflows. The service automates a workflow with integrated Business Process Management. Process Maker automates workflows that are form-based and approval-driven and it improves the way that information flows between people and systems. To create a process, you need to create a workflow, drag-and-drop your workflow as if you are drawing a simple flowchart, and add required forms to any node in the workflow. (processMaker, n.d.)

2.1.2. Activiti

Activiti is a lightweight, java-centric open-source business process model and notation (BPMN, n.d.) engine that supports process automation. Activiti offers a set of cloud native building blocks designed to run on distributed infrastructures. For an example, you can run a business campaign across all social media platforms. To simplify the idea, imagine someone posts on Facebook about your product. The moment this post is identified you can customize the processes that will be activated such as reply back or list this account as a supporter. (Activiti, n.d.)

2.1.3. Microsoft workflow

Microsoft Flow is cloud-based software that allows employees to create and automate workflows and tasks across multiple applications and services without help from developers. It provides the ability to customize a flow by adding one or more options and multiple actions for the same trigger. For an example, if someone tweets your company on twitter, you can set up a flow that follows the person on twitter, sends a reply, and adds the interaction to a premade email. (microsoft workflow, n.d.)

2.2. Existing international student admission systems in Malaysian universities

2.2.1. Asia Pacific University of Technology & Innovation (APU)

2.2.1.1. Intro

Asia Pacific University of Technology & Innovation (APU) located in Kuala Lumpur is amongst Malaysia's Highest Rated Private Universities. The system used by APU is almost entirely a manual email system that consists of three stages as their workflow (APU international students application, n.d.)

1. Submitting a Small online form (only part that's not through email)
2. Requesting of Offer letter

• Initiating an Application: The student has to email application forms and documents

• Confirming your admission: The student has to email an extensive amount of forms and student visa payment slips

3. Student visa processing

• Student Pass - Visa Processing: The management process the visa

• Visa Approval Letter: the student receives the visa approval letter

2.2.1.2. Conclusion (pros & cons)

Yes, they provide an online "initial" registration stage. But the manual system that follows is tedious to go through. Not only is it a waste of time, but it is also filled with human error and extended amounts of indefinite waiting; resulting in a horrible experience for international students.

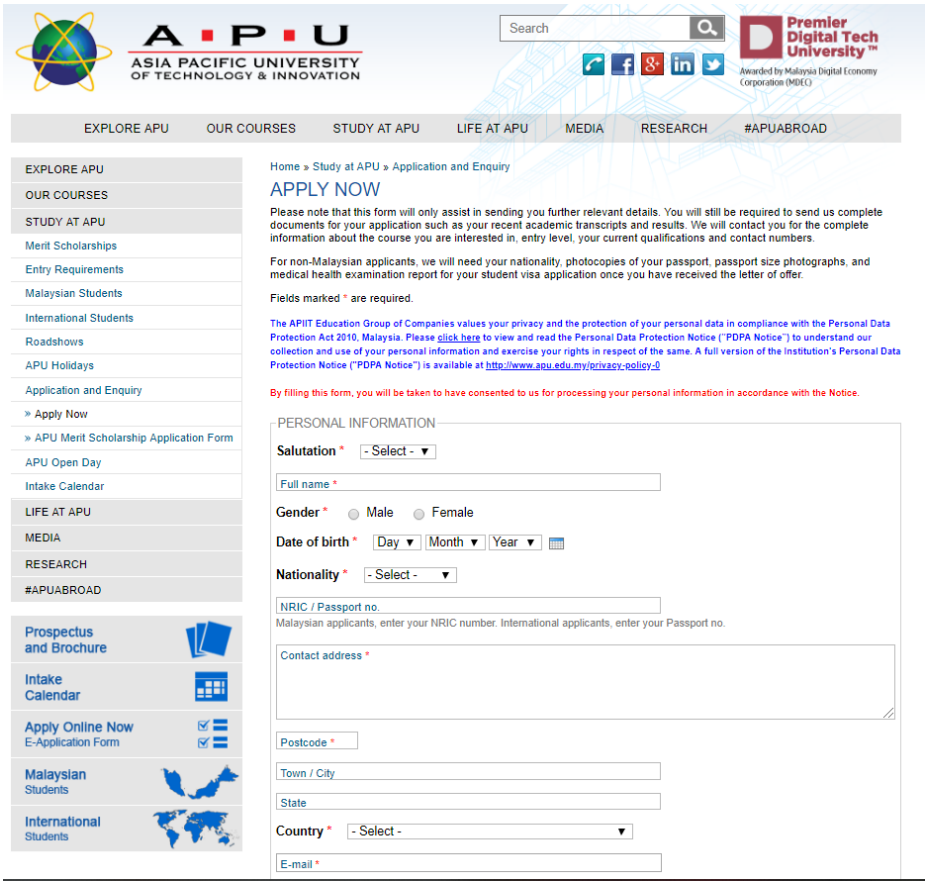


Figure 2.2-1 apu application and enquiry page

Step 2: Confirming your admission

You will receive the Letter of Offer if you meet the eligibility criteria. You can confirm your enrolment and to enable the University to submit an application for your Student Visa, the payments of RM 2,400.00 (USD 600.00) as ***Pre-Arrival Application Processing Fees*** is required immediately along with the documents stated below.

*(*Pre-Arrival Application Processing Fees include; Visa Processing Fees by Education Malaysia Global Services (EMGS) for ALL International students in Malaysia (and applicable GST charged by EMGS), APU/APIIT Application & Processing Fees, Airport assistance and transfer to the University and accommodation upon first arrival in Malaysia. Please take note that this Fee is non-refundable once the offer has been accepted.*)*

Together with your acceptance of the offer, please furnish us with the following for us to commence your Visa application:

- The Stage-1 Payment of Pre-Arrival Application Processing Fee of RM 2,400 (USD 600).
- 4 Passport-sized photographs with WHITE background (35mm x 45mm). The photos must meet the requirements set by EMGS.
- Copies of all pages of passport including blank pages. Passport must be valid for at least 24 months from the Intake Date.
- Academic and other certificates certified as a true copy by the institution issuing the document or a by a Commissioner of Oaths, Notary Public or lawyer. (Please ensure that your name and date of birth as indicated in the academic transcripts/certificates is identical to the details in your passport). If the language of your academic transcript/certificate is not in English, we will require you to submit the certified translated copies together with the original copies for visa application.
- **Health Declaration Form** signed by the student. You are advised to ensure that you are in good health prior to departure to Malaysia. APU recommends that all students attend the Pre-arrival Medical screening to ensure that they are in good health. You can download Health Declaration Form from the link below https://visa.educationmalaysia.gov.my/media/docs/Lampiran_B_-_Health_Dec...
- Please refer to the Medical Screening Guidelines here at https://visa.educationmalaysia.gov.my/catalog/category/view/id/126 for more details.
- IELTS/TOEFL/PTE/MUET English Certification Results.
- No Objection Certificate (NOC)/LOE - Only applicable to students from Oman, Sudan and Iran
- Any other requirements/documents as required by the University in support of your visa application.

Figure 2.2-2 APU list of forms under stage 2's "confirming your application"

2.2.2. Curtin University Sarawak

2.2.2.1. intro

Curtin University, located in Sarawak Malaysia, is the largest international campus of Curtin University, a university based in Perth, Western Australia. The system used by Curtain University is a step further because its less reliant on email. Although they don't really provide a mobile app, the steps and what they call as eStudent are both major components. They have two stages as their workflow. (Curtain university sarawak eStudent, n.d.)

1. Initial application through the website (filling and submitting forms and documents)
2. Follow up the progress through the website's eStudent system

2.2.2.2. Conclusion (pros & cons)

The eStudent system is a good system for providing progress feedback to students. Although there is no mobile application, this automation does a good job to a certain degree.

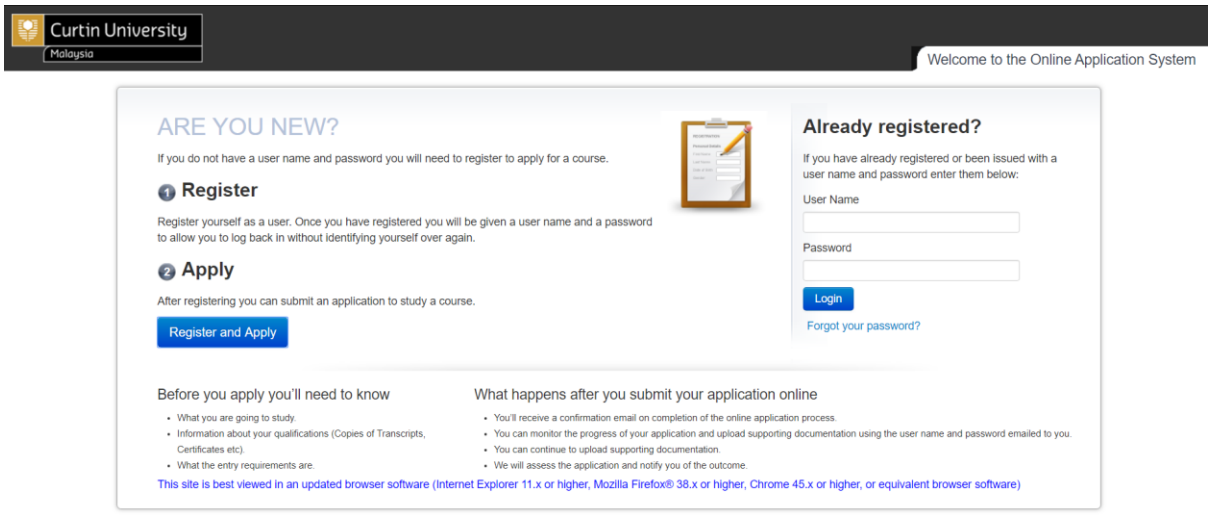


figure 2.2-3 curtain university online application login page

2.3. Summary on existing international student admission systems in Malaysian universities

Table 2.3-1 Existing Admission Systems

University	pros	cons
APU	Online "initial" registration	Mostly manual email system No Mobile application
Curtin University Sarawak	The eStudent system is automated to a certain degree Provides feedback for student	No Mobile application

2.4. Existing problems

The first problem is the lack of mobile applications for student admission systems. A university mobile application if it does in fact exist, besides the information available, it does not include a sufficient admission system. So, whether the student got interested from the website or the application, the student finds out that they have to go through most of the process in most cases through exchanging emails. The second problem is the lack of a reliable feedback system; the student has to wait indefinitely with promises like "X process takes Y amount of business days" with no real status feedback and that ruins the experience for the student.

2.5. Proposed solution

An international student admission system that consists of a website and a mobile application. The system is implemented as a dynamic workflow. The website allows the admin to change the workflow on demand; promoting adaptability to any future adjustments to any stage in the workflow. The website also allows the staff to process student applications. The mobile app allows the student to process their application. Also, it promotes transparency for the student. The student will never worry about waiting indefinitely without knowing the status of their application; by being able to check their application's progress throughout all stages of the workflow.

The system also has a simple messaging system between the student and staff. The messaging system promotes even more transparency in case the student needs to clarify important details. Finally, the system provides important statistics for the admin to help in monitoring the performance and progression of student applications.

2.6. Technological background

2.6.1. Android Application

Android apps are apps that run on devices that has Android operation systems. Typically developed using java or most recently Kotlin.

2.6.2. Web application

A web application is a software application that runs on virtually all devices through browsers. HTML, CSS (mainly bootstrap), JAVASCRIPT, is used as in the process of building websites.

2.6.3. Handlebars JS

Handlebars.js is a templating language derived from Mustache templating language (Mustache, n.d.) used as a logic less html template to help separating constant elements like headers and navigation bars from main content to organize the code and increase maintainability

2.6.4. NodeJS

Node.js is an open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside of a browser. Node.js lets developers use java script as a server-side scripting tool to produce dynamic web page content (Node JS, n.d.)

2.6.5. Express JS

Express JS is a web framework for NodeJS using node package manager (NPM) (Express JS, n.d.) (NPM, n.d.)

2.6.6. AngularJS

AngularJS is a JavaScript-based open-source front-end web framework mainly maintained by Google and by a community of individuals and corporations to address many of the challenges encountered in developing single-page applications. It aims to simplify both the development and the testing of such applications by providing a framework for client-side model–view–controller (MVC) and model–view–view model (MVVM) architectures, along with components commonly used in rich Internet applications. (Angular JS, n.d.)

2.6.7. Postgres

Postgres is a free and open-source relational database management system emphasizing extensibility and technical standards compliance. It is designed to handle a range of workloads, from single machines to data warehouses or Web services with many concurrent users. (Postegres, n.d.)

2.7. Project justification

Although. education is often quoted as a major contributing factor in any economy, and the fact that college is a very important step in building a student's future. It is surprising how admission is neglected by universities. It is the first process that a student does right after they get interested. Due to this fact, students must not go through a bad experience in their first process. Many universities are attempting to have an admission system. But, although it is 2019, they don't even have a mobile application

Based on an e-expectations trend report from Ruffalo-Noel-Levitz (RNL, n.d.) "The Mobile Browsing Behaviours and Expectations of College-Bound High School Students" back In 2012 "The proliferation of smartphones is transforming the online college search experience even further" (E-expectations trend, 2012). Also, based on an updated report back in 2018 "How digital engagement shapes the way high school juniors and seniors choose a college", "Colleges

may love the brand attention, but can lose great applicants when students find the process frustrating.”. (E-expectations trend, 2018).

Approaching a new endeavour like university has a lot of frustrations to international students. From the fear of not sufficiently researching the selected program, to figuring out financial details, to going through the admission process, and dealing with leaving their hometown. This study aims to at least soften one new international student frustration.

2.8. Tools

2.8.1. Kotlin

Android applications are recently being developed using Kotlin. Kotlin is a cross-platform, statically typed, general-purpose programming language with type inference. Kotlin is designed to interoperate fully with Java, and the JVM version of its standard library depends on the Java Class Library, but type inference allows its syntax to be more concise. Kotlin is sponsored by JetBrains and Google through the Kotlin Foundation. Kotlin has been Google’s preferred language for Android app development since 7 May 2019. (Kotlin, n.d.) (Kotlin wiki, n.d.)

2.8.2. Android Studio

The official integrated development environment used for Android development. The flexibility and version testability inside the IDE are both quite unique because it is possible to test on virtual devices or even test on your own physical android device. (Android studio, n.d.) (Android studio wiki, n.d.)

2.8.3. Visual Studio Code

Visual studio code is an open source editor developed by Microsoft. It has all features a developer wishes to have to help organize, debug, increase readability, and visual clues that makes managing a huge tool slightly less troublesome (Visual studio, n.d.)

2.9. Others

2.9.1. Visual paradigm

Visual paradigm is a diagramming tool. It supports UML 2.0 diagrams like class, activity, and sequence diagrams. Also, it has other system modelling diagrams like entity relationship diagrams, context diagram, etc. (visual paradim, n.d.)

2.9.2. Draw.io

Draw.io is an open source charting tool that enables the user to create UML diagrams of all kinds, entity relationship diagram, network diagram, flowcharts and more. (draw.io, n.d.)

2.10. Survey

A Survey conducted to obtain some insight of international student throughout the admission process. Answers from 31 respondents were collected and analysed by google forms. Some important conclusions from the survey were the following:

Only 19% of the students thought email is the ideal way of communication, where almost 40% thought mobile application was ideal. A staggering 77.4% of the students thought a direct messaging system will result in a much better experience for feedback. Their opinion on the overall experience on a scale from 1 to 5 was a tie between 2 and 3 on 25%.

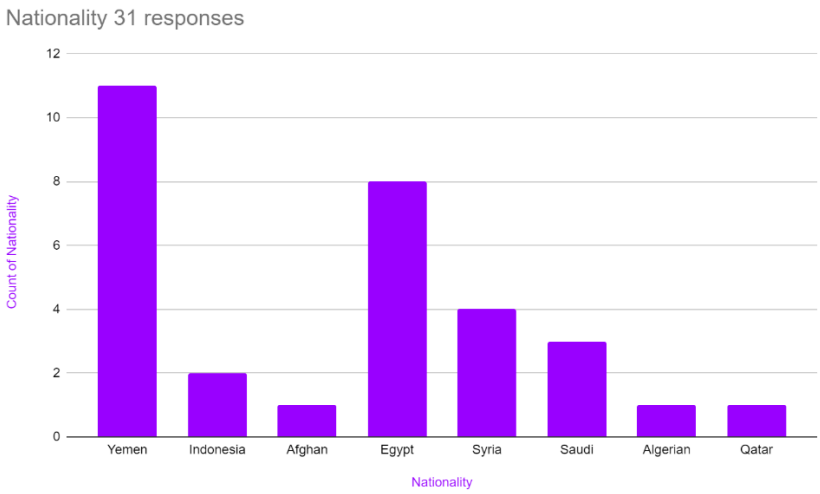


Figure 2.10-1 Respondents Nationality Chart

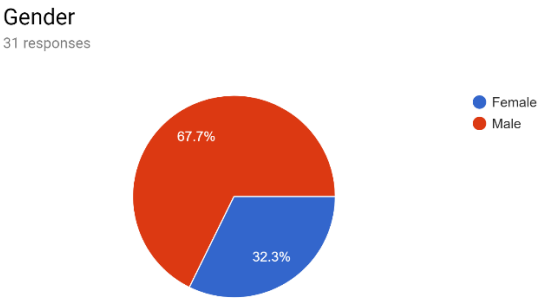


Figure 2.10-2 Respondents Gender Chart

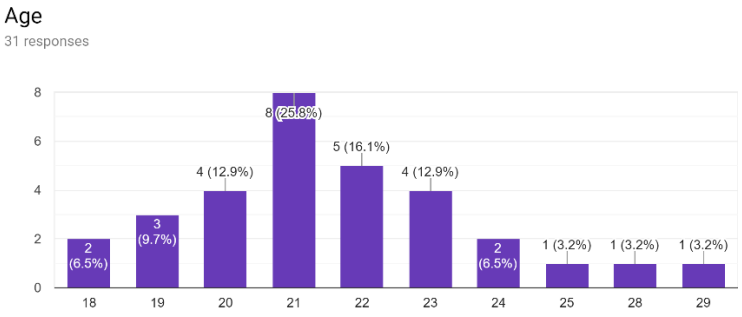


Figure 2.10-3 Respondents Age Chart

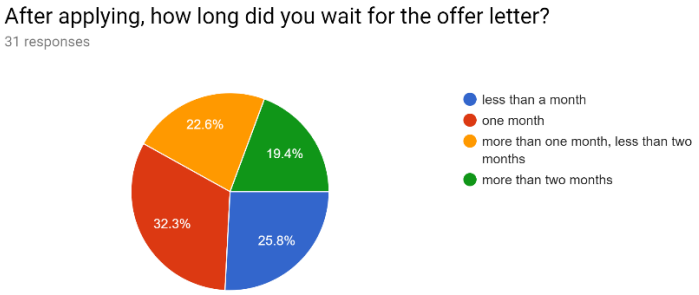


Figure 2.10-4 Offer letter waiting time Chart

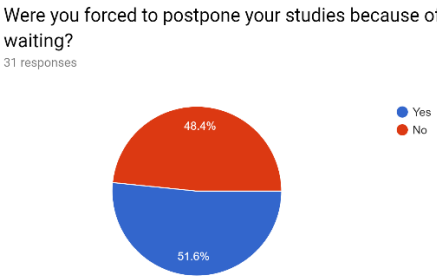


Figure 2.10-5 forced postponed applications Chart

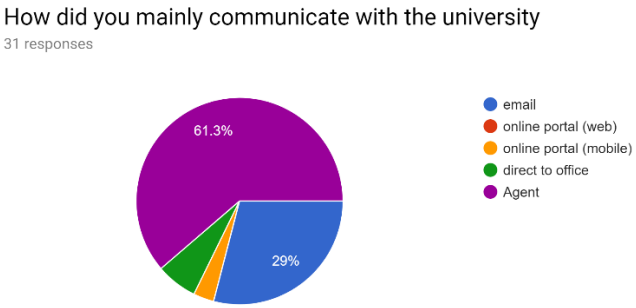


Figure 2.10-6 Main communication method Chart

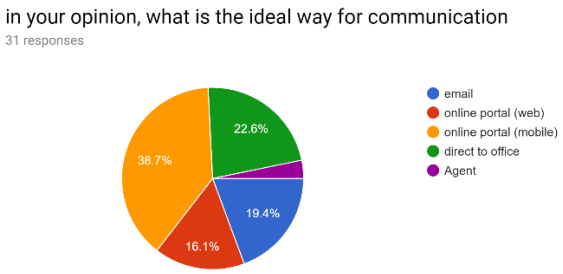


Figure 2.10-7 Students communication method preference Chart

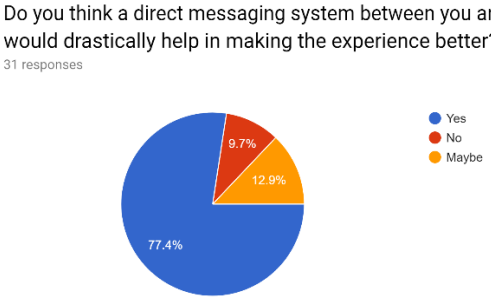


Figure 2.10-8 Messaging system approval Chart

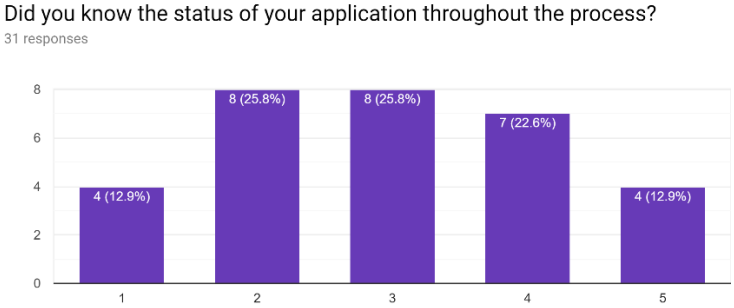


Figure 2.10-9 Application status feedback Chart

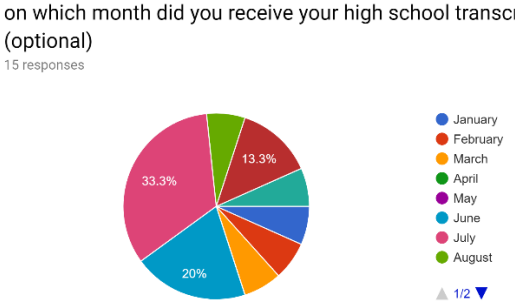


Figure 2.10-10 transcript receive month Chart

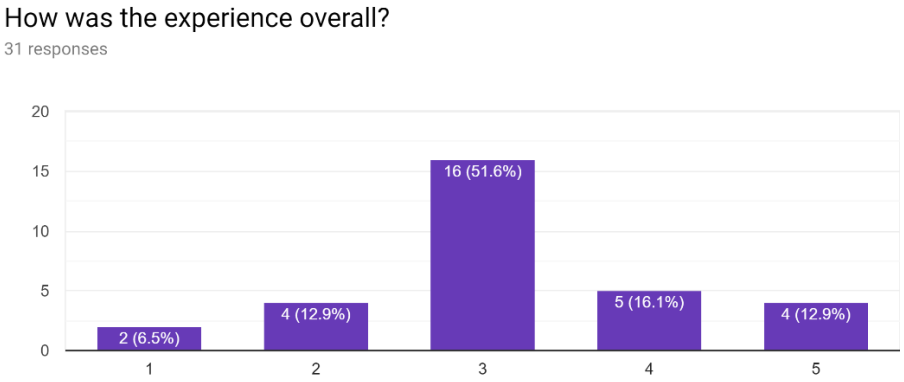


Figure 2.10-11 Overall Experience Chart

3. Requirement

3.1. functional requirements

A functional requirement defines a function of a system or software. A functional requirement consists of required inputs and conditions that trigger a certain behaviour which results in a specific output. Requirements are documented to describe the system's functional capabilities that must be met after development. For this project, the functional requirement is divided into three actors, Admin (web-based system), staff (web-based system), and student (mobile application).

3.1.1. Admin functional requirements

Table 3.1-1 Admin Functional Requirements

Section ID	Requirement Definition	Version
FRS 1.0	The admin shall be able to login and logout	1.0
FRS 2.0	The admin shall be able to retrieve password	1.0
FRS 3.0	The admin shall be able to register staff	1.0
FRS 4.0	The admin shall be able to view the system's operational statistics	1.2
FRS 5.0	The admin shall be able to view individual staff profiles	1.2
FRS 6.0	The admin shall be able to add a node	1.3
FRS 7.0	The admin shall be able to modify a node	1.3
FRS 8.0	The admin shall be able to delete a node	1.3
FRS 9.0	the admin shall be able to create a workflow	1.3
FRS 10.0	The admin shall be able to view their profile	1.3

Table 3.1-2 Admin Login and Logout

Requirement ID	Requirement Definition	Version
FR 1.1	The system shall allow the admin to log in	1.0
FR 1.2	The system shall allow the admin to log out	1.0
FR 1.3	If login was unsuccessful, the system shall display an error message for the admin	1.0
FR 1.4	If login was unsuccessful, the system shall prompt the admin to login again	
FR 1.5	If the login was successful, the system shall redirect the admin to main screen.	

Table 3.1-3 Admin Retrieve Password

Requirement ID	Requirement Definition	Version
FR 2.1	If the admin forgot the password, the system shall allow the admin to retrieve the password	
FR 2.2	the admin shall receive an email with their password	

Table 3.1-4 Admin Register Staff

Requirement ID	Requirement Definition	Version
FR 3.1	The system shall allow the admin to fill all required fields to register staff	1.0
FR 3.2	If registration was unsuccessful, the system shall display an error message for the Admin	1.0
FR 3.3	If registration was unsuccessful, the system shall prompt the Admin to register the staff again.	1.0
FR 3.4	If the registration was successful, the system shall display success message.	1.0
FR 3.5	If the registration was successful, the system shall create the staff profile.	1.0

Table 3.1-5 Admin View The System's Operational Statistics

Requirement ID	Requirement Definition	Version
FR 4.1	The system shall allow the admin to view system's statistics (e.g., number of registered students last week, over all average process time per node per day, current number of population, etc)	1.0

Table 3.1-6 Admin View The Individual Staff Profiles

Requirement ID	Requirement Definition	Version
FR 5.1	The system shall display the staff's details (name, email, picture, etc) to the admin	1.0
FR 5.2	the system shall allow the admin to view individual staff statistics such as (average process time per node, number of tasks currently taken on by staff, etc)	1.0

Table 3.1-7 Admin Add Node

Requirement ID	Requirement Definition	Version
FR 6.1	the system shall allow the admin to specify the name of the node	1.3
FR 6.2	the system shall allow the admin to select the workflow of the node	1.3
FR 6.3	the system shall allow the admin to specify the previous node(s) and the next node(s)	1.3
FR 6.4	the system shall allow the admin to specify the state of the node	1.2
FR 6.5	the system shall allow the admin to select the responsible staff(s)	1.3
FR 6.6	the system shall allow the admin to add tasks	1.4

Table 3.1-8 Admin Modify Node

Requirement ID	Requirement Definition	Version
FR 7.1	the system shall allow the admin to specify the name of the node	1.3
FR 7.2	The system shall allow the admin to modify tasks attached to the node	1.3
FR 7.3	the system shall allow the admin to modify the previous node(s) and the next node(s)	1.3
FR 7.4	the system shall allow the admin to modify the state of the node	1.2
FR 7.5	the system shall allow the admin to modify the responsible staff(s)	1.3

Table 3.1-9 Admin Delete Node

Requirement ID	Requirement Definition	Version
FR 8.1	the system shall allow the admin to select a node for deletion	1.0
FR 8.2	the system shall display a confirm message for the deletion	1.0
FR 8.3	If deletion was unsuccessful, the system shall display an error message for the admin	1.0
FR 8.4	If deletion was unsuccessful, the system shall prompt the admin to try again	1.0
FR 8.5	If the deletion was successful, the system shall redirect the admin to main screen.	1.0

Table 3.1-10 Admin Create Workflow

Requirement ID	Requirement Definition	Version
FR 9.1	the system shall allow the admin to specify the name of the workflow	1.0
FR 9.2	the system shall automatically pre-set the start node	1.0
FR 9.3	If creation was unsuccessful, the system shall display an error message for the admin	1.0
FR 9.4	If creation was unsuccessful, the system shall prompt the admin to try again	1.0
FR 9.5	If the creation was successful, the system shall redirect the admin to main screen.	1.0

Table 3.1-11 Admin View Profile

Requirement ID	Requirement Definition	Version
FR 10.1	The system shall allow the admin to change the password	1.0
FR 10.2	The system shall display the admin's details (name, email, picture, etc)	1.0

3.1.2. Staff functional requirements

Table 3.1-12 Staff Functional Requirements

Section ID	Requirement Definition	Version
FRS 11.0	The staff shall be able to login and logout	1.0
FRS 12.0	The staff shall be able to retrieve password	1.0
FRS 13.0	The staff shall be able to view individual student profiles	1.0
FRS 14.0	the staff shall be able to process forms	1.3
FRS 15.0	The staff shall be able to select a node	1.3
FRS 16.0	The staff shall be able to send and receive a message	1.3
FRS 17.0	The staff shall be able to view profile	1.0

Table 3.1-13 Staff Login and Logout

Requirement ID	Requirement Definition	Version
FR 9.1	The system shall allow the staff to log in	1.0
FR 9.2	The system shall allow the staff to log out	1.0
FR 9.3	If login was unsuccessful, the system shall display an error message for the staff.	1.0
FR 9.4	If login was unsuccessful, the system shall prompt the staff to login again	1.0
FR 9.5	If the login was successful, the system shall redirect the staff to main screen.	1.0

Table 3.1-14 Staff Retrieve Password

Requirement ID	Requirement Definition	Version
FR 12.1	If the staff forgot the password, the system shall allow the staff to retrieve the password	1.0
FR 12.2	the staff shall receive an email with their password	1.0

Table 3.1-15 Staff View Individual Student Profiles

Requirement ID	Requirement Definition	Version
FR 13.1	The system shall display the student's details (name, email, picture, status etc) to the staff	1.0
FR 13.2	the system shall allow the staff to start processing the student application	1.0

Table 3.1-16 Staff Process Forms

Requirement ID	Requirement Definition	Version
FR 14.1	The staff shall be able to approve the student's processed form	1.0
FR 14.2	The staff shall be able to reject the student's processed form	1.0

Table 3.1-17 Staff Select A Node

Requirement ID	Requirement Definition	Version
FR 15.1	The system shall allow the staff to view basic information of the node (name, status, etc)	1.0
FR 15.2	the system shall allow the staff to only select nodes assigned to them	1.0
FR 15.3	The system shall display the list of students under the selected node	1.0

Table 3.1-18 Staff Send and Receive Message

Requirement ID	Requirement Definition	Version
FR 16.1	the system shall allow the staff to send a message to a student in text format	1.0
FR 16.2	the system shall allow the staff to receive a message from a student in text format	1.0

Table 3.1-19 Staff View Profile

Requirement ID	Requirement Definition	Version
FR 17.1	The system shall allow the staff to change the password	1.0
FR 17.2	The system shall display the staff's details (name, email, picture, etc)	1.0

3.1.3. Student functional requirements

Table 3.1-20 Student Functional Requirements

Section ID	Requirement Definition	Version
FRS 18.0	The student shall be able to login, logout, and register	1.0
FRS 19.0	The student shall be able to retrieve password	1.0
FRS 20.0	The student shall be able to send/receive a message	1.3
FRS 21.0	The student shall be able to view profile	1.0

Table 3.1-21 Staff Login, Logout, And Register

Requirement ID	Requirement Definition	Version
FR 18.1	The system shall allow the user to register as student	1.0
FR 18.2	The system shall allow the student to log in	1.0
FR 18.3	The system shall allow the student to log out	1.0
FR 18.4	If login was unsuccessful, the system shall display an error message for the student.	1.0
FR 18.5	If login was unsuccessful, the system shall prompt the student to login again	1.0
FR 18.6	If registration was unsuccessful, the system shall display an error message for the student.	1.0

FR 18.7	If registration was unsuccessful, the system shall prompt the student to register again.	1.0
FR 18.8	If the login was successful, the system shall redirect the student to main screen.	1.0
FR 18.9	If the registration was successful, the system shall display success message.	1.0
FR 18.10	If the registration was successful, the system shall create the student profile.	1.0

Table 3.1-22 Student Retrieve Password

Requirement ID	Requirement Definition	Version
FR 19.1	If the student forgot the password, the system shall allow the student to retrieve the password	1.0
FR 19.2	the student shall receive an email with their password	1.0

Table 3.1-23 Student Send and Receive Message

Requirement ID	Requirement Definition	Version
FR 20.1	the system shall allow the student to send a message to a staff in text format	1.0
FR 20.2	the system shall allow the student to receive a message from a staff in text format	1.0
FR 20.2	the system shall allow the student to select the receiver staff from the responsible staff list	1.3

Table 3.1-24 Student View Profile

Requirement ID	Requirement Definition	Version
FR 21.1	The system shall allow the student to change the password	1.0
FR 21.2	The system shall display the student's details (name, email, picture, etc)	1.0

3.2. Quality requirements

3.2.1. Admin quality requirements

Table 3.2-1 Admin Quality Requirements

Requirement ID	Requirement Definition	Version
QR 1.0	The system shall only allow authorized admin access.	1.0
QR 2.0	the system shall provide the accurate statistics	1.0
QR 3.0	The system shall hide password at all times	1.0
QR 4.0	The system shall provide responsive web design	1.0
QR 5.0	The system shall provide a simple user interface design	1.0
QR 6.0	The system shall ensure a safe node addition, modification, and deletion	1.0

3.2.2. Staff quality requirements

Table 3.2-2 Staff Quality Requirements

Requirement ID	Requirement Definition	Version
QR 7.0	The system shall only allow authorized staff access.	1.0
QR 9.0	The system shall hide password at all times	1.0
QR 10.0	The system shall provide responsive web design	1.0
QR 11.0	The system shall provide a simple user interface design	1.0
QR 12.0	the system shall update the status correctly according to any updates done by the staff	1.0

3.2.3. Student quality requirements

Table 3.2-3 Student Quality Requirements

Requirement ID	Requirement Definition	Version
QR 14.0	The system shall only allow authorized student access.	1.0
QR 15.0	The system shall ensure that student can only access their information	1.0
QR 16.0	The system shall hide password at all times	1.0
QR 17.0	The system shall provide responsive user interface design	1.0
QR 18.0	The system shall support mobile devices with API level 16 and above.	1.0
QR 19.0	the system shall ensure that the student has filled all required fields	1.0
QR 20.0	the system shall ensure that the student's data is captured correctly	1.0
QR 22.0	The system shall provide a simple user interface design	1.0
QR 23.0	the system shall update the status correctly according to the current node the student is in	1.0

3.3. Use case specification

3.3.1. UCS-1 Admin's Use Case

Table 3.3-1 UCS-1 Admin's Use Case

Reference	name
UC-1	Login
UC-2	change password
UC-3	Retrieve password
UC-4	View Staff profile
UC-5	view profile
UC-6	add a node
UC-7	modify node
UC-8	delete node
UC-9	create workflow
UC-10	register staff
UC-11	View system statistics

Table 3.3-2 UC-1 Login

Attribute	Value
use case	Login
ID	UC-1
actor	admin
Description	The admin login to their account to start using the system
assumption	-
precondition	
Main scenario	1- The admin inputs ID and password in Login Screen.
	2- The admin clicks on "login" button
	3- the system will redirect the admin to the main screen
Alternative scenario (Unsuccessful)	1- The admin inputs ID and password in Login Screen.
	2- The admin clicks on "login" button
	3- The system displays "Error" message and prompt the admin to login again
postcondition	-

Table 3.3-3 UC-2 Change Password

Attribute	Value
use case	change password
ID	UC-2
actor	admin
Description	The admin changes their password
assumption	-
precondition	The admin is viewing their profile
Main scenario	1- The admin clicks on "change password" from their profile
	2- The system displays change password screen
	3- The admin fills in the required details such as old password and new password
	4- the admin clicks on "save changes"
	5- The system displays "success" message and redirects the admin to their profile
Alternative scenario (Unsuccessful)	1- The admin clicks on "change password" from their profile
	2- The system displays change password screen
	3- The admin fills in the required details such as old password and new password
	4- the admin clicks on "save changes"
	5- The system displays "error" message and prompts the admin to re-enter required details
postcondition	-

Table 3.3-4 UC-3 Retrieve Password

Attribute	Value
use case	Retrieve password
ID	UC-3
actor	admin
Description	The admin retrieves their password as a recovery procedure
assumption	The admin forgot their password
precondition	The admin has an ID and an email.
Main scenario	1- The admin clicks on "forgot password"
	2- The admin fills in the required details
	3- the admin clicks on "Retrieve password"
	4- The system receives an email with their password.
Alternative scenario (Unsuccessful)	1- The admin clicks on "forgot password"
	2- The admin fills in the required details
	3- the admin clicks on "Retrieve password"
	4- The system show "Error" message and prompt the admin to re-enter required details
postcondition	The admin receives an email containing their password

Table 3.3-5 UC-4 View Staff Profile

Attribute	Value
use case	View staff profile
ID	UC-4
actor	Admin
Description	The Admin views the staff profile to see their information (e.g., name, email, statistics, etc)
assumption	
precondition	The admin is viewing staff specific statistics
Main scenario	1- The admin clicks on the staff profile
	2- the system displays the staff information
Alternative scenario (Unsuccessful)	-
postcondition	-

Table 3.3-6 UC-5 View Profile

Attribute	Value
use case	View profile
ID	UC-5
actor	admin
Description	The admin views their profile to see their information (e.g., name, email, picture, etc)
assumption	-
precondition	The admin is logged in
Main scenario	1- The admin clicks on profile image
	2- the system displays the admin information
Alternative scenario	-
postcondition	-

Table 3.3-7 UC-6 Add Node

Attribute	Value
use case	add node
ID	UC-6
actor	admin
Description	The admin adds a node
assumption	-
precondition	The admin is logged in
Main scenario	1- the admin specifies the name of the node
	2- the admin specifies the workflow of the node
	3- the admin specifies the staff(s) responsible
	4- the admin specifies the task(s) and click add per task
	5- the admin specifies the state of the node
	6- the admin specifies the previous/next node(s)
	7- the admin clicks on add
	8- The system displays "success" message and redirects the admin to the main screen
Alternative scenario (unsuccessful)	1- the admin specifies the name of the node
	2- the admin specifies the workflow of the node
	3- the admin specifies the staff(s) responsible
	4- the admin specifies the task(s) and click add per task
	5- the admin specifies the status of the node
	6- the admin specifies the previous/next node(s)
	7- the admin clicks on add
	8- The system displays "error" message and prompts the admin to re-enter required details
postcondition	the node is added to the workflow

Table 3.3-8 UC-7 Modify Node

Attribute	Value
use case	modify node
ID	UC-7
actor	admin
Description	The admin modifies the node, the admin can modify the name, form, status, previous/next node, etc
assumption	the node exists in the system
precondition	The admin is logged in
Main scenario	1- the admin selects the node in the workflow
	2-the admin changes any of the name, tasks, state, previous/next node, etc
	3-the admin clicks "save changes"
	4- The system displays "success" message and redirects the admin to the main screen
Alternative scenario	1- the admin selects the node
	2-the admin changes any of the name, tasks, state, previous/next node, etc
	3-the admin clicks "save changes"
	4- The system displays "error" message and redirects the admin to the main screen
postcondition	the node's changes are saved

Table 3.3-9 : UC-8 Delete Node

Attribute	Value
use case	delete node
ID	UC-8
actor	admin
Description	The admin deletes any node
assumption	the node exists in the system
precondition	The admin is logged in
Main scenario	1-the admin specifies the workflow
	2- the admin selects the node
	3-the admin clicks on "delete"
	4- the system prompts the admin to confirm
	5- the admin confirms
	6- The system displays "success" message and redirects the admin to the main screen
postcondition	the node is deleted from the workflow

Table 3.3-10 UC-9 Create Workflow

Attribute	Value
use case	create workflow
ID	UC-9
actor	Admin
Description	The Admin creates a new workflow
assumption	the workflow wasn't created before
precondition	the admin is logged in
Main scenario	1- The admin clicks on create workflow
	2- the admin specifies the name and description
	3- the admin clicks on "create"
	4- The system displays "success" message and redirects the admin to the main screen
Alternative scenario (Unsuccessful)	1- The admin clicks on create workflow
	2- the admin specifies the name and description
	3- the admin clicks on "create"
	4- The system displays "error" message and prompts the admin to try again
postcondition	the workflow is added to the system

Table 3.3-11 UC-10 Register Staff

Attribute	Value
use case	register staff
ID	UC-10
actor	Admin
Description	The Admin registers a new staff
assumption	the staff wasn't registered before
precondition	the admin is logged in
Main scenario	1- The admin clicks on register staff
	2- the admin specifies the staff information (name, email)
	3- the admin clicks on "Register"
	4- The system displays "success" message and redirects the admin to the main screen
Alternative scenario (Unsuccessful)	1- The admin clicks on register staff
	2- the admin specifies the staff information (name, email, pw)
	3- the admin clicks on "Register"
	4- The system displays "error" message and prompts the admin to try again
postcondition	the Staff is added to the system and an email is sent to the staff

Table 3.3-12 UC-11 View System Statistics

Attribute	Value
use case	View system statistics
ID	UC-11
actor	admin
Description	The admin views system information (e.g., node, staff, student, etc)
assumption	-
precondition	The admin is logged in
Main scenario	1- The admin logs in
	2- system displays system statistics
Alternative scenario	-
postcondition	-

3.3.2. UCS-2 Staff's Use Case

Table 3.3-13 UCS-2 Staff's Use Case

Reference	name
UC-12	Login
UC-13	View student profile
UC-14	Retrieve password
UC-15	select a node
UC-16	change password
UC-17	view profile
UC-18	Approve/Reject
UC-19	send a message
UC-20	view message

Table 3.3-14 UC-12 Login

Attribute	Value
use case	Login
ID	UC-12
actor	Staff
Description	The Staff login to their account to start using the system
assumption	-
precondition	The Staff has already been registered in the system by the admin
Main scenario	1- The Staff inputs ID and password in Login Screen.
	2- The staff clicks on "login" button
	3- the system will redirect the staff to the main screen
Alternative scenario (Unsuccessful)	1- The Staff inputs ID and password in Login Screen.
	2- The staff clicks on "login" button
	3- The system displays "Error" message and prompt the staff to login again
postcondition	the staff is directed to main screen

Table 3.3-15 UC-13 View Student Profile

Attribute	Value
use case	View student profile
ID	UC-13
actor	Staff
Description	The staff selects the student profile to see their information (e.g., name, email, course, etc)
assumption	the student is registered in the system and belongs to the selected node
precondition	The staff is logged in
Main scenario	1- The staff clicks on the student profile
	2- the system displays the student information
Alternative scenario (Unsuccessful)	-
postcondition	-

Table 3.3-16 UC-14 Retrieve Password

Attribute	Value
use case	Retrieve password
ID	UC-14
actor	Staff
Description	The staff retrieves their password as a recovery procedure
assumption	The staff forgot their password
precondition	The staff has an ID and an email.
Main scenario	1- The staff clicks on "forgot password"
	2- The staff fills in the required details
	3- the staff clicks on "Retrieve password"
	4- The system sends an email with their password.
Alternative scenario (Unsuccessful)	1- The staff clicks on "forgot password"
	2- The staff fills in the required details
	3- the staff clicks on "Retrieve password"
	4- The system show "Error" message and prompt the staff to re-enter required details
postcondition	The staff receives an email containing their password

Table 3.3-17 UC-15 Select A Node

Attribute	Value
use case	select a node
ID	UC-15
actor	Staff
Description	The staff selects a node they want to process
assumption	
precondition	the staff is responsible for the node
Main scenario	1- The staff clicks on a node from the staff main screen
	2- The system displays the node's basic information (name, status, etc)
	3- the system displays the students who are under the selected node
Alternative scenario (Unsuccessful)	-
postcondition	-

Table 3.3-18 UC-16 Change Password

Attribute	Value
use case	change password
ID	UC-16
actor	Staff
Description	The staff changes their password
assumption	-
precondition	The staff is viewing their profile
Main scenario	1- The staff clicks on "change password" from their profile
	2- The system displays change password screen
	3- The staff fills in the required details such as old password and new password
	4- the staff clicks on "save changes"
	5- The system displays "success" message and redirects the staff to their profile
Alternative scenario (Unsuccessful)	1- The staff clicks on "change password" from their profile
	2- The system displays change password screen
	3- The staff fills in the required details such as old password and new password
	4- the staff clicks on "save changes"
	5- The system displays "error" message and prompts the staff to re-enter required details
postcondition	the password is changed

Table 3.3-19 UC-17 View Profile

Attribute	Value
use case	View profile
ID	UC-17
actor	Staff
Description	The staff views their profile to see their information (e.g., name, email, course, etc)
assumption	-
precondition	The staff is logged in
Main scenario	1- The staff clicks on profile image
	2- the system displays the staff information
Alternative scenario	-
postcondition	-

Table 3.3-20 UC-18 Approve/Reject

Attribute	Value
use case	approve/reject
ID	UC-18
actor	Staff
Description	The staff approves/rejects the student's processed form
assumption	There is a student that is awaiting approval
precondition	the staff is viewing the node screen (during process form)
Main scenario	1- The staff clicks on "approve" or "reject"
	2- The system displays "success" message and redirects the staff to main screen
Main scenario	1- The staff clicks on "reject"
	2- The system displays "success" message and redirects the staff to main screen
Alternative scenario (Unsuccessful)	1- The staff clicks on "approve" or "reject"
	2- The system displays "error" message and prompts the staff to retry again
postcondition	the system updates the student's status

Table 3.3-21 UC-19 Send Message

Attribute	Value
use case	send a message
ID	UC-19
actor	Staff
Description	The staff sends a message to the student
assumption	-
precondition	the staff is viewing the student
Main scenario	1- The staff types their message in the textbox
	2- the staff clicks on "send"
	3- The system displays "success" message and redirects the staff to student profile
Alternative scenario (Unsuccessful)	1- The staff types their message in the textbox
	2- the staff clicks on "send"
	3- The system displays "error" message and prompts the staff to retry again
postcondition	message is sent to the student

Table 3.3-22 : UC-20 View Message

Attribute	Value
use case	view message
ID	UC-20
actor	Staff
Description	The staff views a message received by a student
assumption	messages are available for viewing
precondition	the staff is inside the messages screen
	1- the staff clicks on messages
	2-the system displays messages
Main scenario	3- the staff clicks on the message
	4- the system displays the message content
postcondition	The content of the message is displayed

3.3.3. UCS-3 Student's Use Case

Table 3.3-23 UCS-3 Student's Use Case

Reference	name
UC-21	Register
UC-22	Login
UC-23	View profile
UC-24	Retrieve password
UC-25	change password
UC-26	send a message
UC-27	view message

Table 3.3-24 UC-21 Register

Attribute	Value
use case	Register
ID	UC-21
actor	Student
Description	The student registers an account to start using the system
assumption	-
precondition	The student is a new student
Main scenario	1- The student clicks on the "Register" button
	2- The system displays Registration screen
	3- the student navigates between the forms
	4- The student fills in the required details in each form
	5- The student clicks on "Submit" button
	6- The system displays "Success" message and redirects the student to Login Screen.
Alternative scenario (Unsuccessful)	1- The student clicks on the "Register" button
	2- The system displays Registration screen
	3- the student navigates between the forms
	4- The student fills in the required details in each form
	5- The student clicks on "Submit" button
	6- The system displays "Error" message and redirects the student to Registration Screen.
postcondition	The student receives an email containing both Registration ID and the student's profile is created

Table 3.3-25 UC-22 Login

Attribute	Value
use case	Login
ID	UC-22
actor	Student
Description	The student login to their account to start using the system
assumption	The student has already registered in the system
precondition	The student has to have the Registration ID and the correct password.
Main scenario	1- The student inputs Registration ID and password in Login Screen.
	2- The student clicks on "login" button
	3- the system will redirect the student to the main screen
Alternative scenario (Unsuccessful)	1- The student inputs Registration ID and password in Login Screen.
	2- The student clicks on "login" button
	3- The system displays "Error" message and prompt the student to login again
postcondition	-

Table 3.3-26 UC-23 View Profile

Attribute	Value
use case	View profile
ID	UC-23
actor	Student
Description	The student views their profile to see their information (e.g., name, email, course, etc)
assumption	-
precondition	The student is logged in
Main scenario	1- The student clicks on profile image
	2- the system displays the student information
Alternative scenario	-
postcondition	-

Table 3.3-27 UC-24 Retrieve Password

Attribute	Value
use case	Retrieve password
ID	UC-24
actor	Student
Description	The student retrieves their password as a recovery procedure
assumption	The student forgot their password
precondition	The student has a Registration ID and an email.
Main scenario	1- The student clicks on "forgot password"
	2- The student fills in the required details
	3- the student clicks on "Retrieve password"
	4- The system sends an email with their password
Alternative scenario (Unsuccessful)	1- The student clicks on "forgot password"
	2- The student fills in the required details
	3- the student clicks on "Retrieve password"
	4- The system show "Error" message and prompt the student to re-enter required details
postcondition	The student receives an email containing their password

Table 3.3-28 UC-25 Change Password

Attribute	Value
use case	change password
ID	UC-25
actor	Student
Description	The student changes their password
assumption	-
precondition	The student is viewing their profile
Main scenario	1- The student clicks on "change password" from their profile
	2- The system displays change password screen
	3- The student fills in the required details such as old password and new password
	4- the student clicks on "save changes"
	5- The system displays "success" message and redirects the student to their profile
Alternative scenario (Unsuccessful)	1- The student clicks on "change password" from their profile
	2- The system displays change password screen
	3- The student fills in the required details such as old password and new password
	4- the student clicks on "save changes"
	5- The system displays "error" message and prompts the student to re-enter required details
postcondition	-

Table 3.3-29 UC-26 Send Message

Attribute	Value
use case	send a message
ID	UC-26
actor	student
Description	The student sends a message to the student
assumption	-
precondition	the student must be inside the node's screen
	1- the student specifies the staff
Main scenario	2- The student types their message in the textbox
	3- the student clicks on "send"
	4- The system displays "success" message and redirects the student to node's screen
Alternative scenario (Unsuccessful)	1- the student specifies the staff
	2- The student types their message in the textbox
	3- the student clicks on "send"
	4- The system displays "error" message and prompts the student to retry again
postcondition	message is sent to the staff

Table 3.3-30 UC-27 View Message

Attribute	Value
use case	view message
ID	UC-27
actor	student
Description	The student views a message received by a staff
assumption	messages are available for viewing
precondition	the student is inside the messages screen
Main scenario	1- the student clicks on the message
	2- the system displays the message content
postcondition	The content of the message is displayed

3.4. Use case diagram

Use case diagrams are examples or scenarios that the user goes through to achieve desired functionalities. (Introduction to the Diagrams of UML 2.X, n.d.)

3.4.1. Admin Use Case diagram

Admin has the ability to login to be able to use the system. This scenario includes the verification of ID and Password entered. In the case of not remembering the password the admin can retrieve the password as a recovery procedure. Admin has the ability to view their profile too. The admin can also register staff in the system. Furthermore, the admin can create workflow and add/modify/delete nodes for workflows. Finally, the admin can view all sorts of system statistics to measure performance.

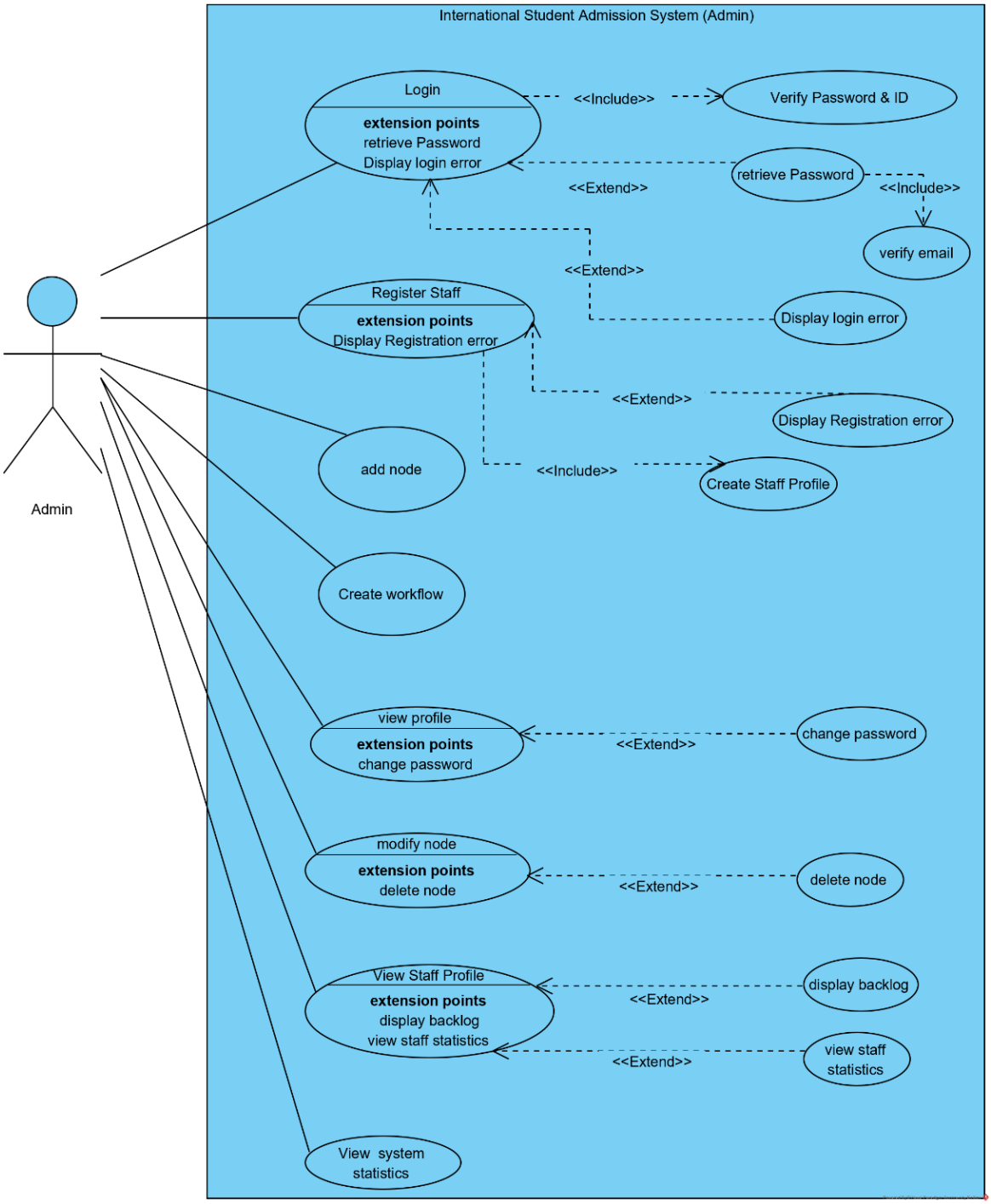


Figure 3.4-1 Admin Use Case diagram

3.4.2. Staff Use Case diagram

Staff has the ability to login to be able to use the system. This scenario includes the verification of ID and Password entered. In the case of not remembering the password the staff can retrieve the password as a recovery procedure. Staff has the ability to view their profile too. The Staff can also select nodes process its students with the approval and rejection of the application in the system. Finally, the staff can send and receive messages.

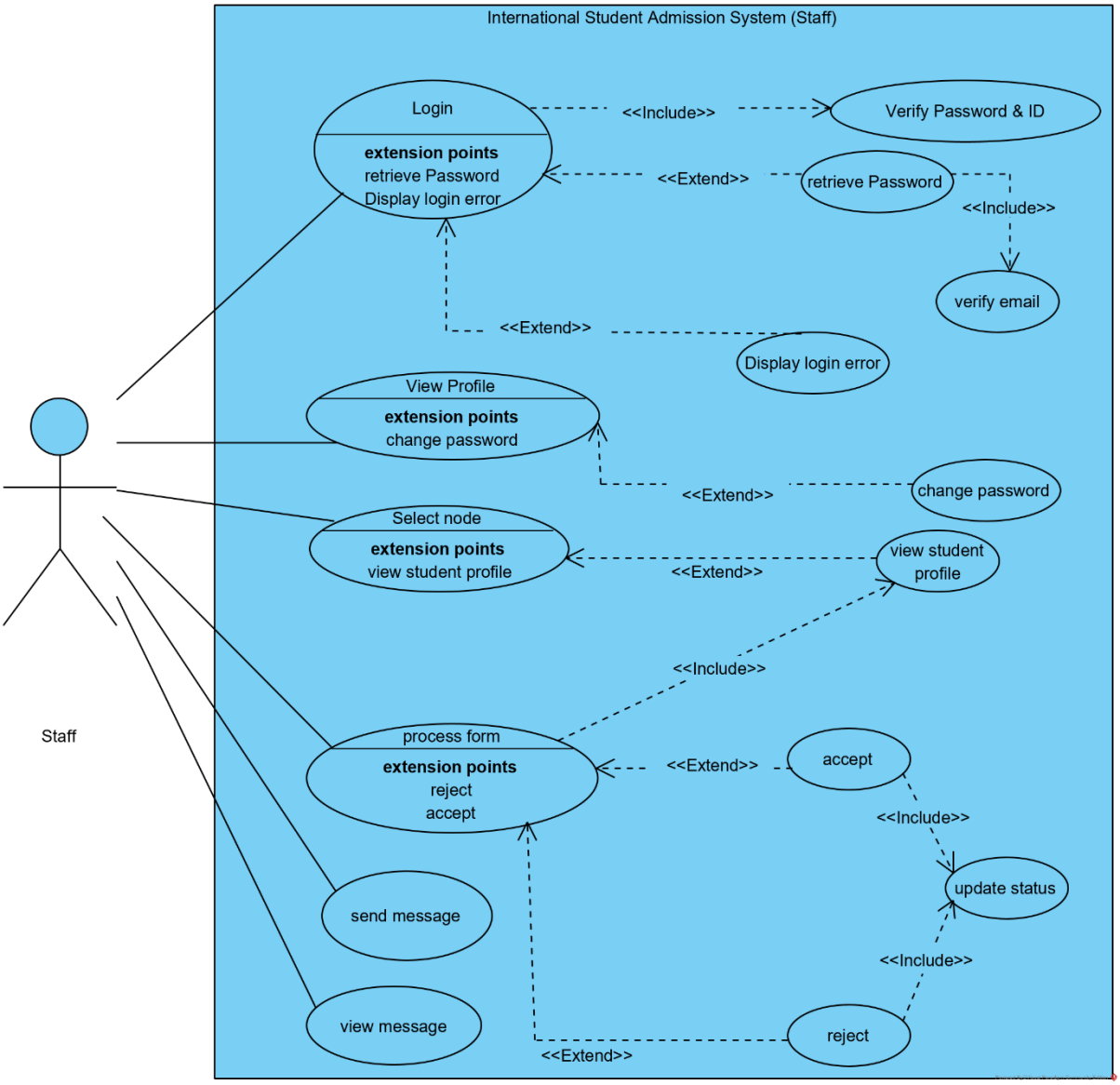


Figure 3.4-2 Staff Use Case diagram

3.4.3. Student Use Case diagram

Student has the ability to login to be able to use the system. This scenario includes the verification of ID and Password entered. In the case of not remembering the password the student can retrieve the password as a recovery procedure. Student has the ability to view their profile too. Finally, student can send and receive messages.

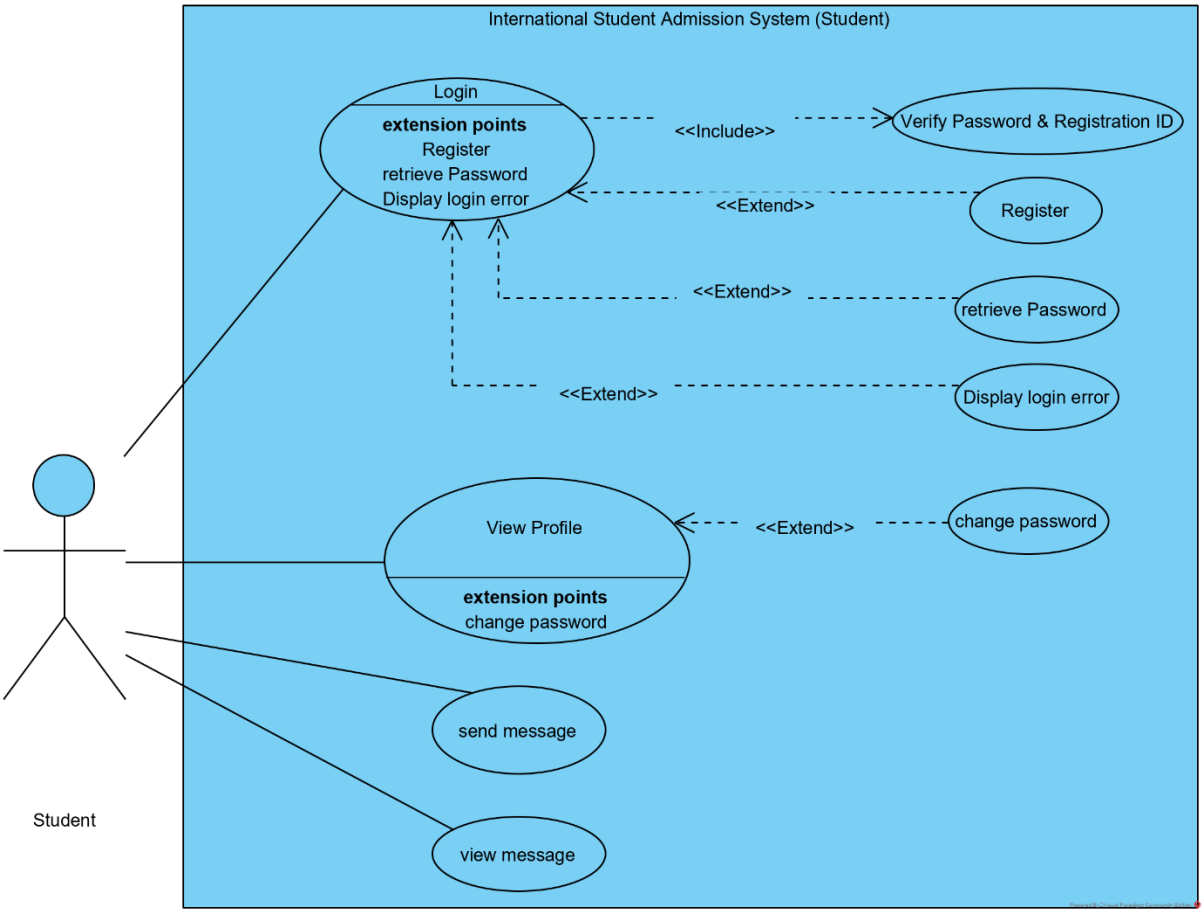


Figure 3.4-3 Student Use Case diagram

3.5. Context diagram

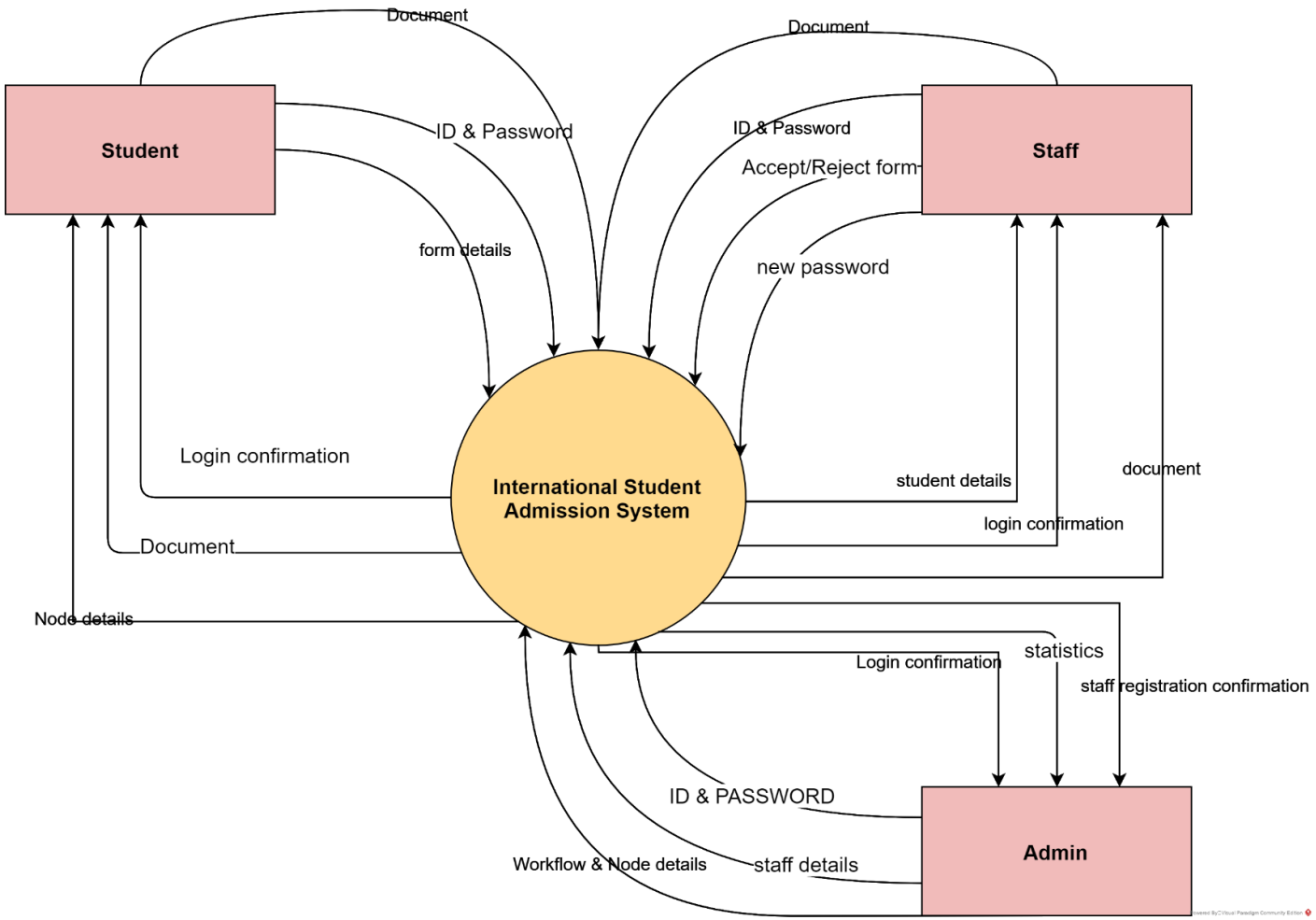


Figure 3.5-1 Context diagram

3.6. Entity relationship diagram

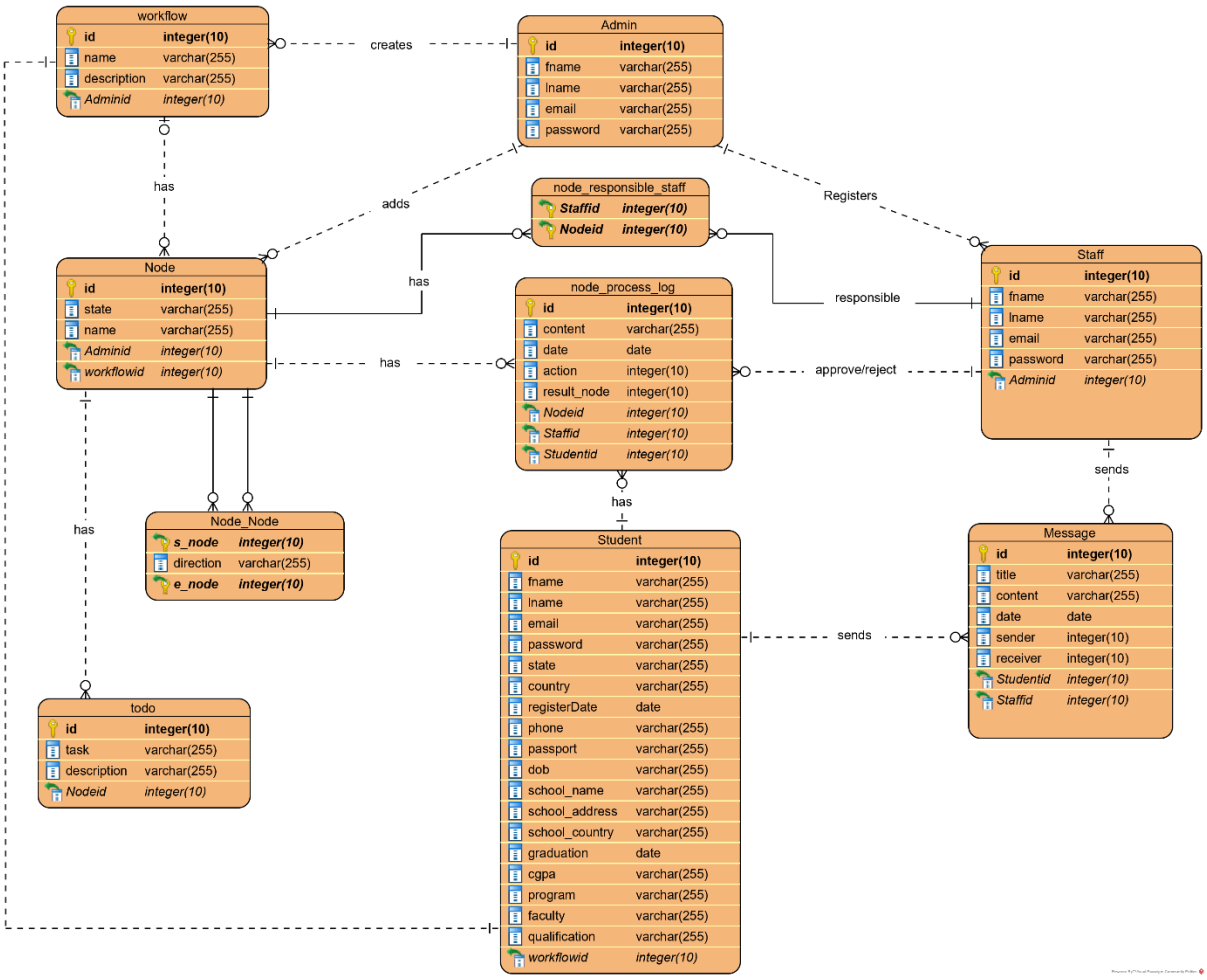


Figure 3.5-1 Entity relationship diagram

3.7. Data Dictionary

Table 3.7-1 Data dictionary

Entity Name	Column Name	Data Type	Length	Primary Key	Nullable	Unique
Admin	email	varchar	255	false	false	false
	fname	varchar	255	false	false	false
	id	integer	10	true	false	false
	lname	varchar	255	false	false	false
	password	varchar	255	false	false	false
Message	content	varchar	255	false	false	false
	date	date	0	false	false	false
	id	integer	10	true	false	false
	receiver	integer	10	false	false	false
	sender	integer	10	false	false	false
	Staffid	integer	10	false	false	false
	Studentid	integer	10	false	false	false
	title	varchar	255	false	false	false
Node	Adminid	integer	10	false	false	false
	id	integer	10	true	false	false
	name	varchar	255	false	false	false
	state	varchar	255	false	false	false
	workflowid	integer	10	false	false	false
Node_Node	direction	varchar	255	false	false	false
	e_node	integer	10	true	false	false
	s_node	integer	10	true	false	false

Entity Name	Column Name	Data Type	Length	Primary Key	Nullable	Unique
node_process_log	action	integer	10	false	false	false
	content	varchar	255	false	false	false
	date	date	0	false	false	false
	id	integer	10	true	false	false
	Nodeid	integer	10	false	false	false
	result_node	integer	10	false	false	false
	Staffid	integer	10	false	false	false
	Studentid	integer	10	false	false	false
node_responsible_staff	Nodeid	integer	10	true	false	false
	Staffid	integer	10	true	false	false
Staff	Adminid	integer	10	false	false	false
	email	varchar	255	false	false	false
	fname	varchar	255	false	false	false
	id	integer	10	true	false	false
	lname	varchar	255	false	false	false
	password	varchar	255	false	false	false
Student	cgpa	varchar	255	false	false	false
	country	varchar	255	false	false	false
	dob	varchar	255	false	false	false
	email	varchar	255	false	false	false
	faculty	varchar	255	false	false	false
	fname	varchar	255	false	false	false
		date	0	false	false	false

Entity Name	Column Name	Data Type	Length	Primary Key	Nullable	Unique
	graduation					
	id	integer	10	true	false	false
	lname	varchar	255	false	false	false
	passport	varchar	255	false	false	false
	password	varchar	255	false	false	false
	phone	varchar	255	false	false	false
	program	varchar	255	false	false	false
	qualification	varchar	255	false	false	false
	registerDate	date	0	false	false	false
	school_address	varchar	255	false	false	false
	school_country	varchar	255	false	false	false
	school_name	varchar	255	false	false	false
	state	varchar	255	false	false	false
	workflowid	integer	10	false	false	false
todo	description	varchar	255	false	false	false
	id	integer	10	true	false	false
	Nodeid	integer	10	false	false	false
	task	varchar	255	false	false	false
workflow	Adminid	integer	10	false	false	false
	description	varchar	255	false	false	false
	id	integer	10	true	false	false
	name	varchar	255	false	false	false

4. Design

4.1. Sequence diagram

Sequence diagram describes the sequential logic or particular scenario of use case(s), the events that external actors generate, and possible internal communications between the system's objects. (Introduction to the Diagrams of UML 2.X, n.d.)

4.1.1. General user

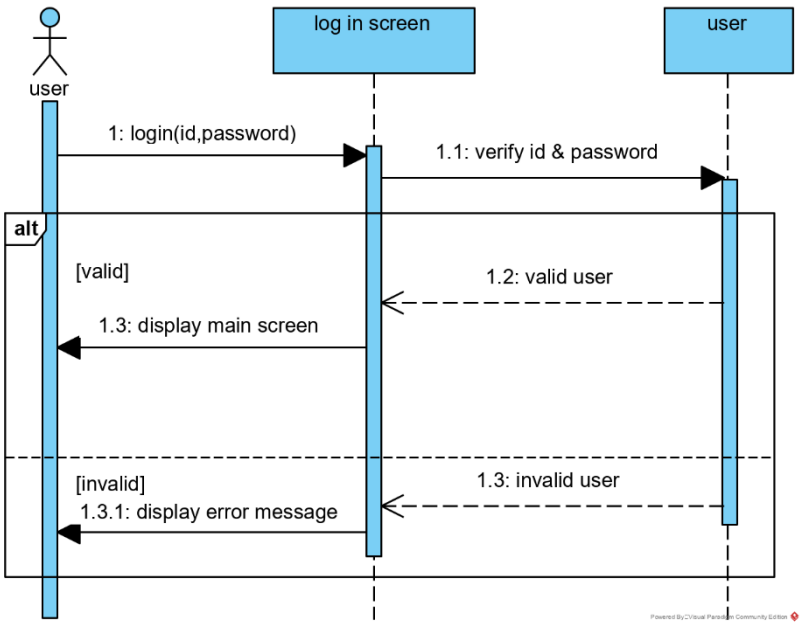


Figure 4.1-1 User Login

4.1.1.1. Login

User needs to be logged in to use the international student admission system. To login, the user will enter id and password and click "Login" button. If validation was successful, user will be redirected to main screen. However, if not valid, the system will display an error message.

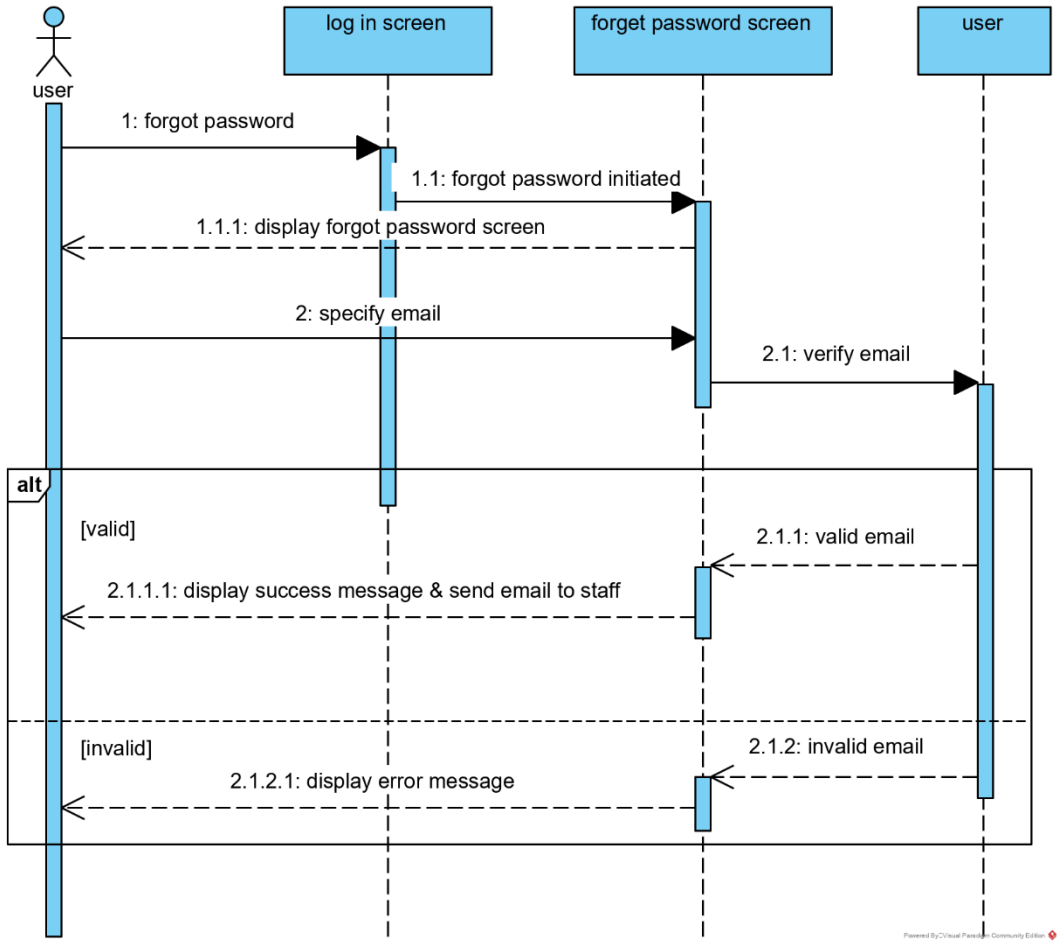


Figure 4.1-2 User Retrieve Password

4.1.1.2. Retrieve password

User needs to be logged in to use the international student admission system. When the user forgets the password, the user will click “forgot password” button. The system will display the forgot password screen. The user has to provide their registered email. If validation was successful, user will be sent an email for recovery. However, if not valid, the system will display an error message.

4.1.1.3. View profile & change password

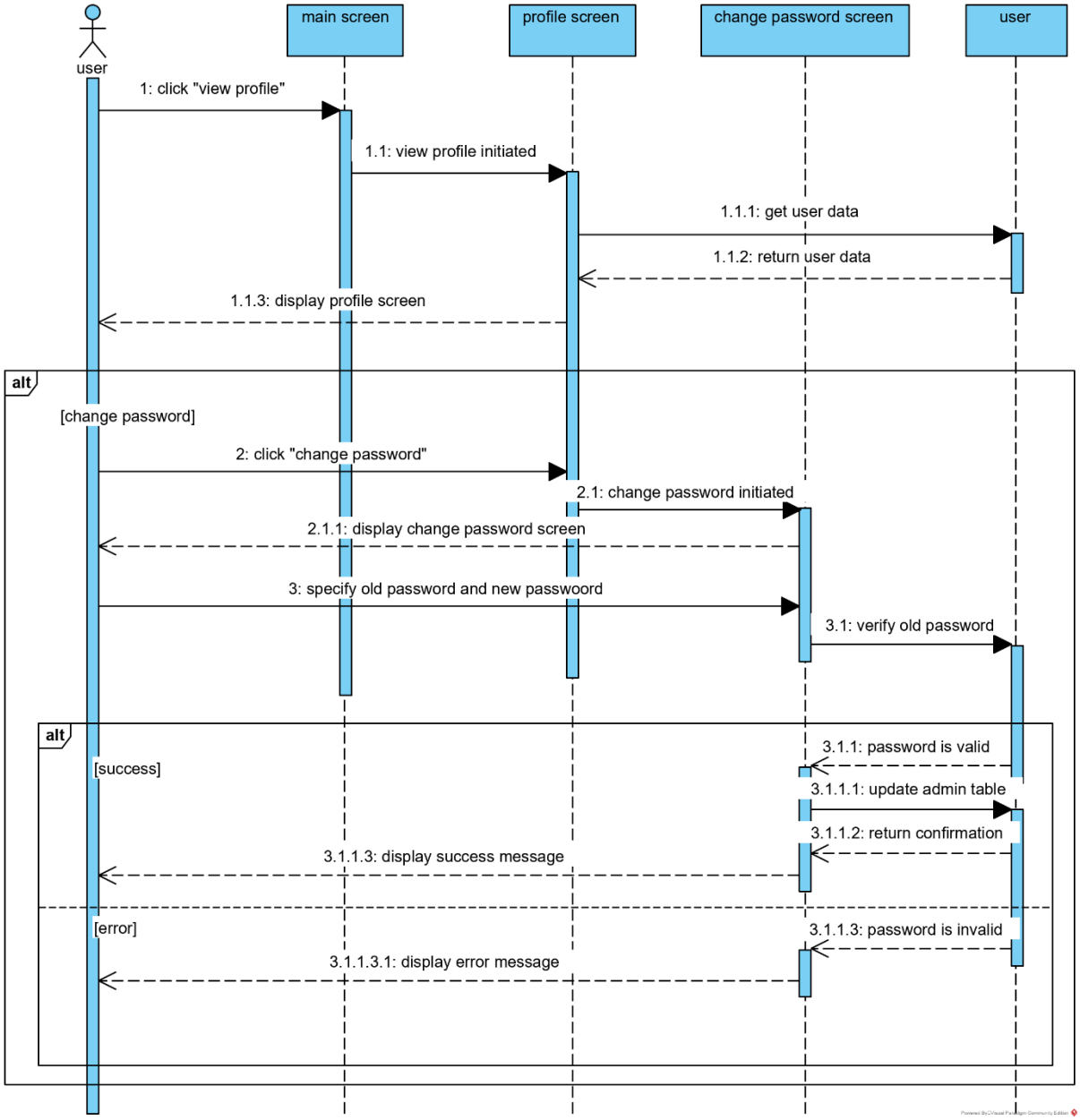


Figure 4.1-3 User View Profile & Change Password

When the user wants to view their profile, the user will click “view profile” button. The system will display user data. The user has the option to change their password. The user can opt to click on “change password” then provide old password and new password. If validation was successful, the user’s password will be updated in the user’s table and the system will display success message. However, if not valid, the system will display an error message.

4.1.2. Admin

4.1.2.1. Register Staff

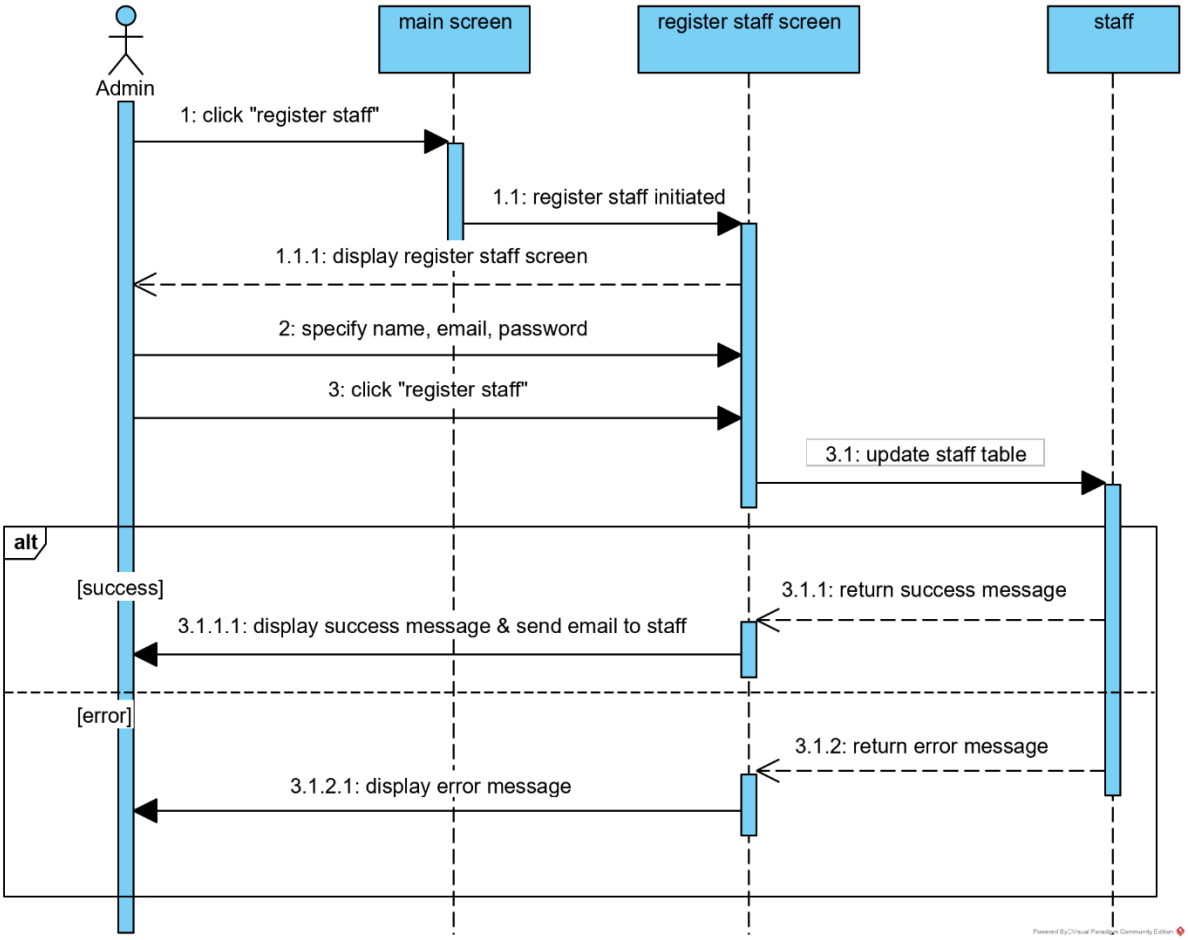


Figure 4.1-4 Admin Register Staff

Admin has to register staff in order for the staff to be inside the system. To do that, the admin clicks on “register staff” and specify the name and email. After updating the staff table to have a new staff, if validation was successful, system will display success message and also sends an email to staff. However, if not valid, the system will display an error message.

4.1.2.2. View staff profile

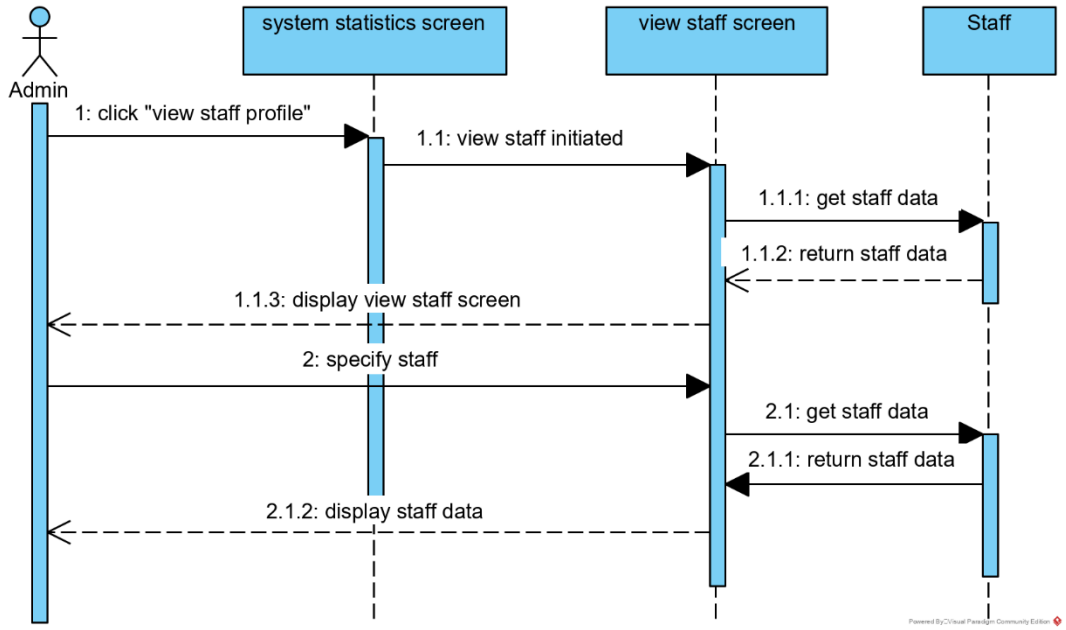


Figure 4.1-5 Admin View Staff Profile

Admin can view staff profile. To do that, the admin clicks on “view staff profile”. The system will display a list of all registered staff. The admin then proceeds to specify a staff. The system will display staff profile.

4.1.2.3. Create workflow

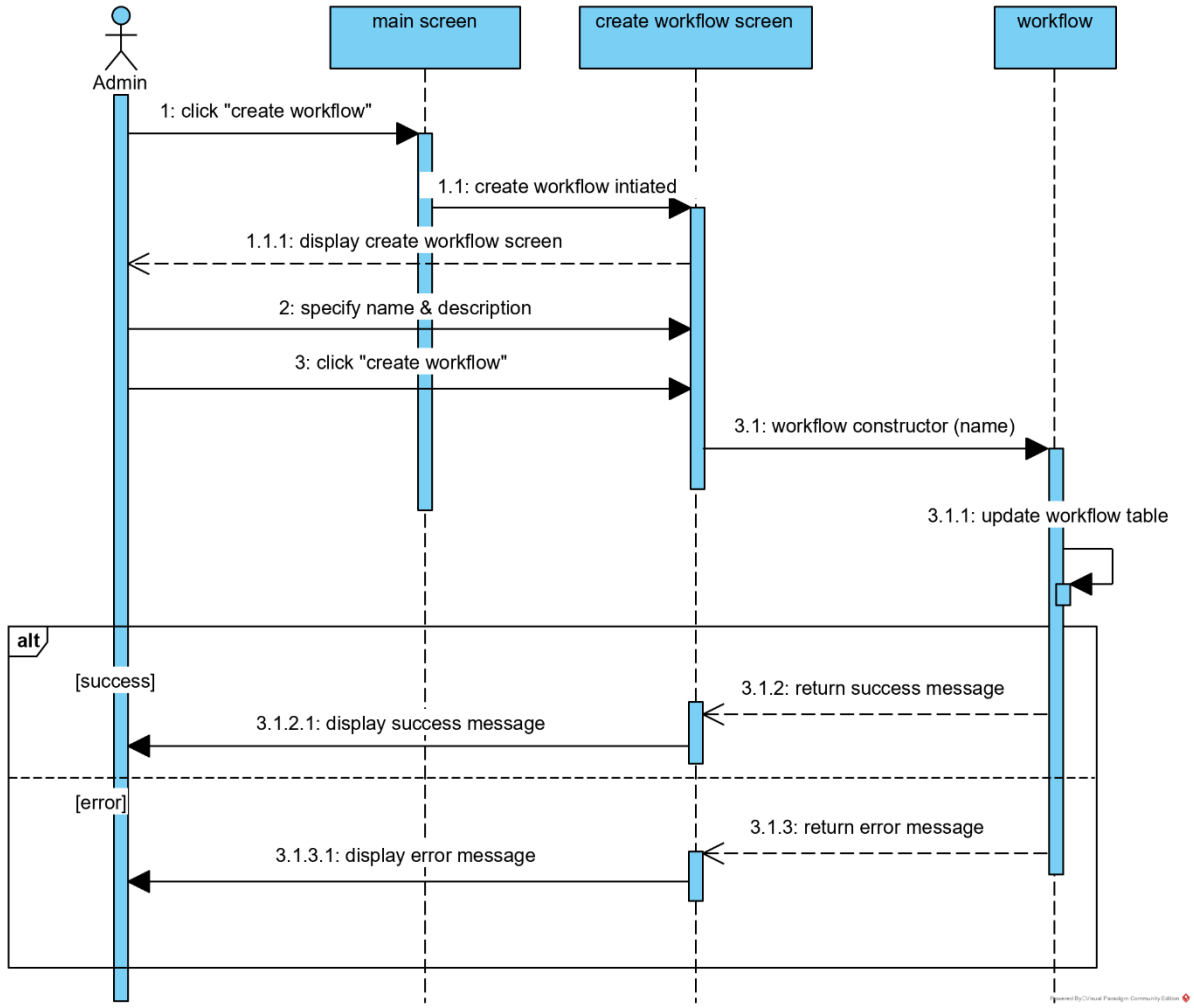


Figure 4.1-6 Admin Create Workflow

Admin can create workflow. To do that, the admin clicks on “create workflow”. The system will display create workflow screen. From there, the admin specifies the name and description of the workflow and clicks on create. If validation was successful, system will display success message. However, if not valid, the system will display an error message.

4.1.2.4. add node

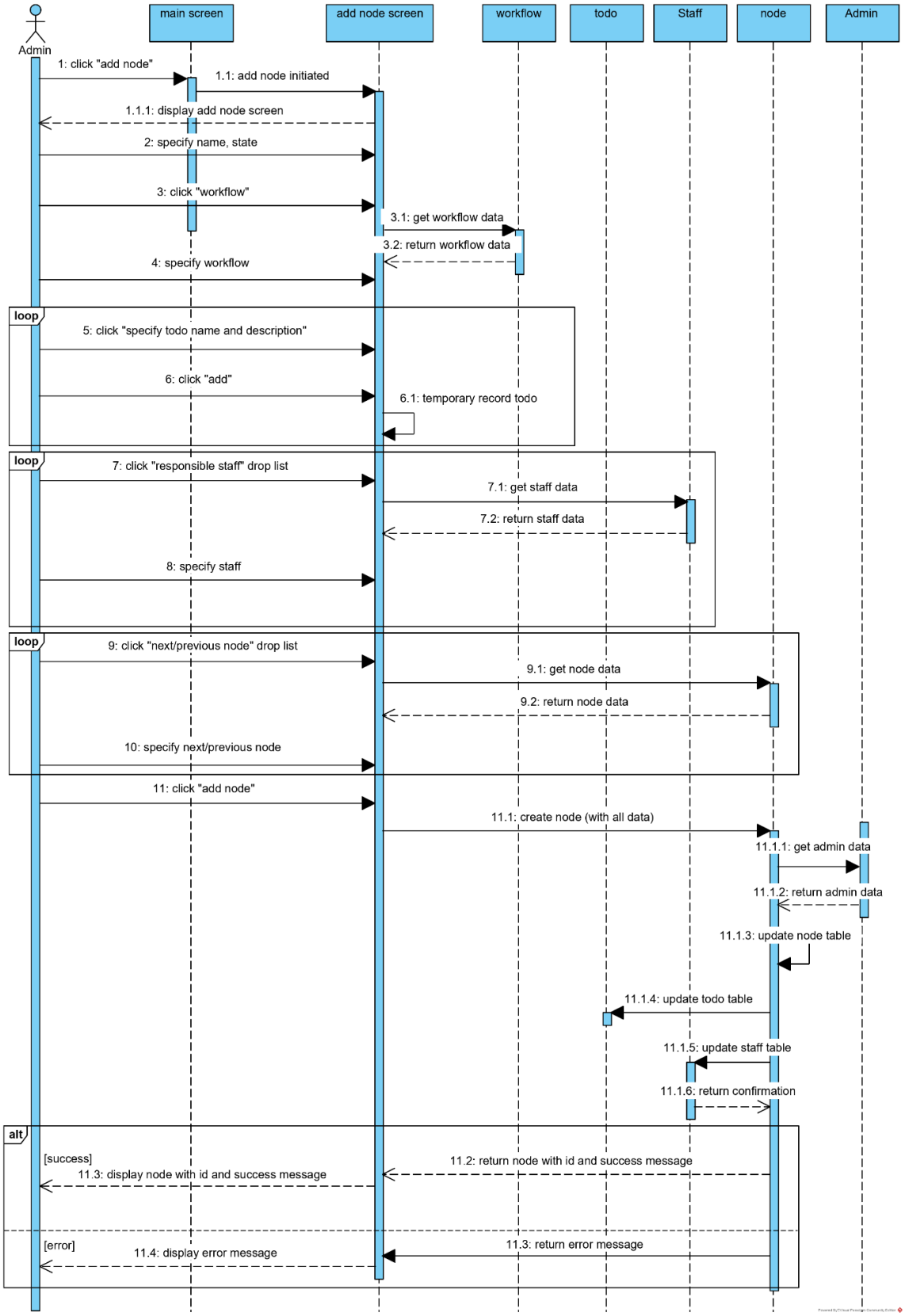


Figure 4.1-7 Admin Add Node

Admin can add a node to a workflow. To do that, the admin clicks on “add node”. The system will display create node screen. From there, the admin specifies the name, state, workflow, to-do list, list of responsible staff, list of previous nodes, and list of next nodes. Finally, clicks on add node. If validation was successful, the node table and the staff table will be updated and system will display success message. However, if not valid, the system will display an error message.

4.1.2.5. Modify node

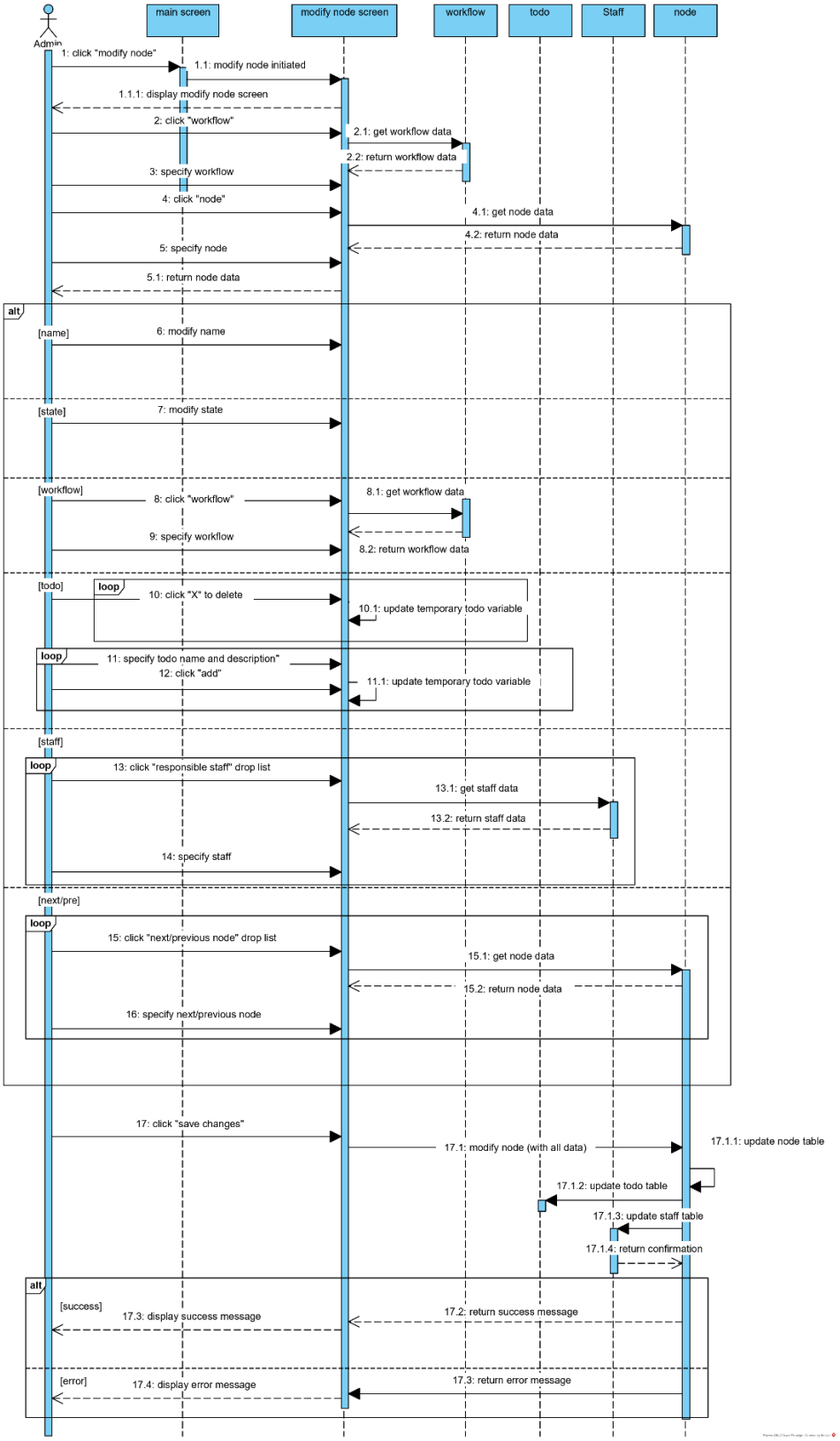


Figure 4.1-8 Admin Modify Node

Admin can modify a node to a workflow. To do that, the admin clicks on “modify node”. The system will display modify node screen. From there, the admin can modify the name, state, workflow, to-do list, list of responsible staff, list of previous nodes, and list of next nodes. Finally, clicks on save changes. If validation was successful, the node table and staff table will be updated and system will display success message. However, if not valid, the system will display an error message.

4.1.2.6. Delete node

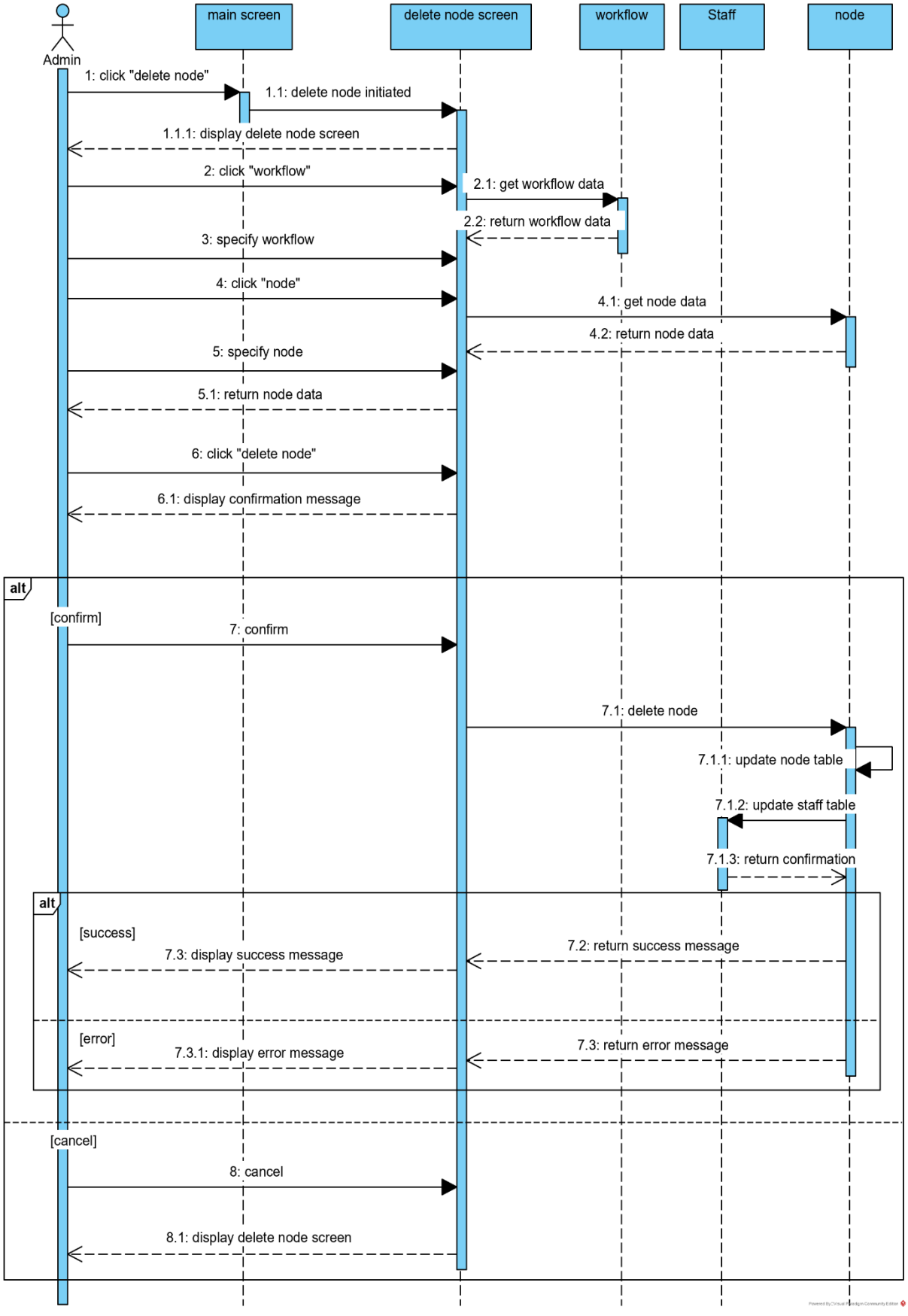


Figure 4.1-9 Admin Delete Node

Admin can delete a node. To do that, the admin clicks on “delete node”. The system will display the delete node screen. Inside there, the admin has to specify the workflow in which the node belongs to. Finally, the admin selects a node to delete it. The system will notify the admin because deleting can be a risky action. If and only if the admin confirms, the node will be deleted. If validation was successful, node table and staff table will be updated and system will display success message. However, if not valid, the system will display an error message.

4.1.3. Staff

4.1.3.1. Select a node & view student profile

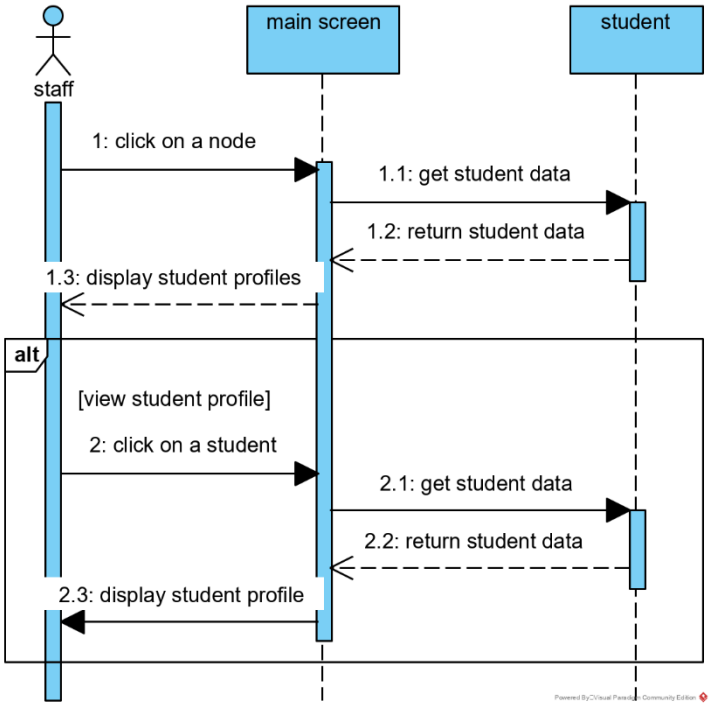


Figure 4.1-10 Staff Select A Node & View Student Profile

Staff has to select a node in order to start. To do that, the staff simply clicks on one of the available nodes. Only the nodes that this staff is responsible off will be clickable. The system will display the list of students under the selected node. The staff can view the student profile by clicking on the student.

4.1.3.2. Approve/Reject

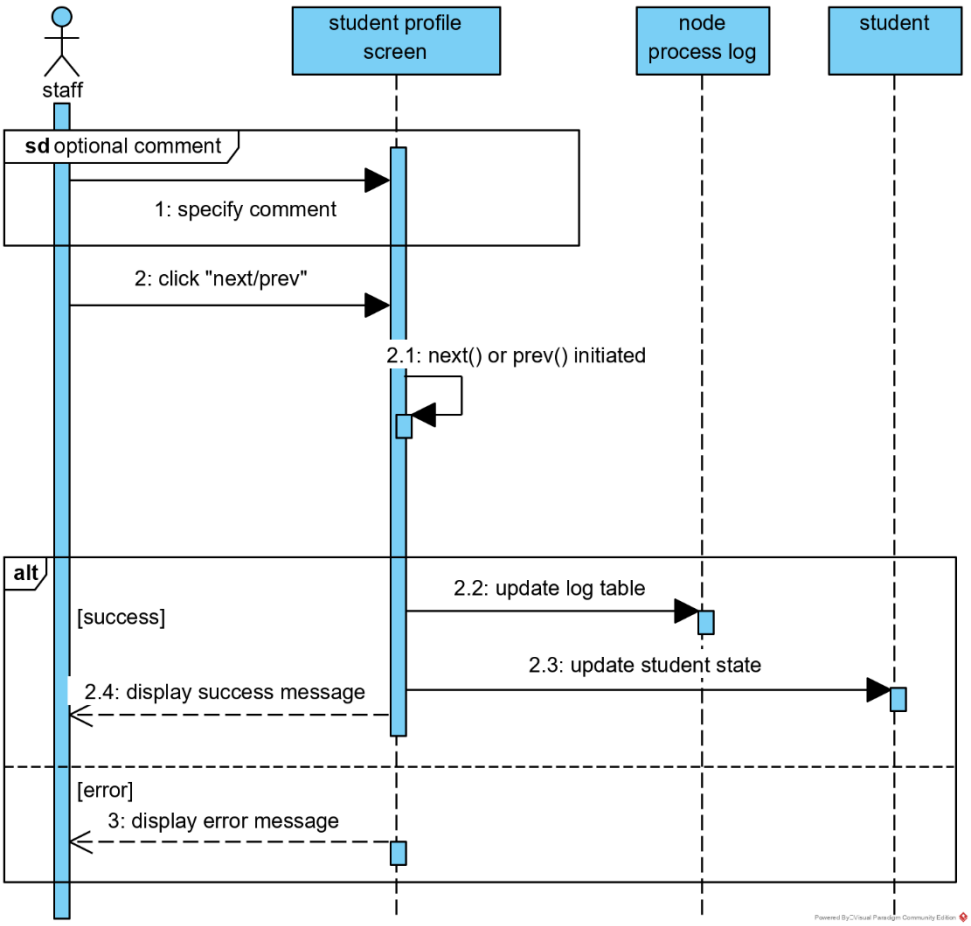


Figure 4.1-11 Staff Approve/Reject

Staff can approve/reject the student. To do that, the staff simply clicks on “next” or “prev” while viewing student profile. If validation was successful, system will display success message. However, if not valid, the system will display an error message.

4.1.3.3. Send message

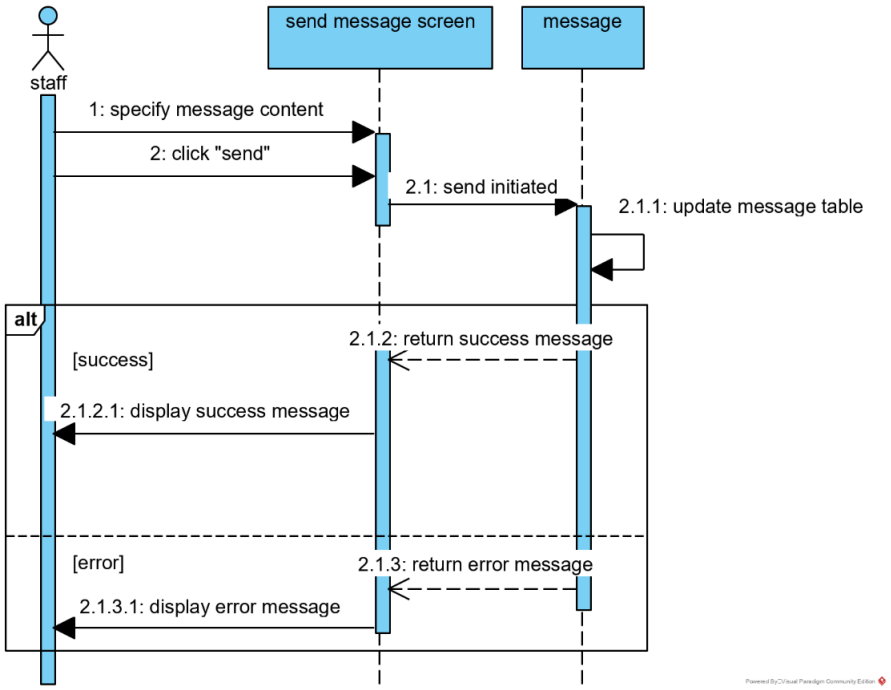


Figure 4.1-12 Staff Send Message

Staff can send a message. To do that, the staff specifies the content of the message and click "send". If validation was successful, system will display success message. However, if not valid, the system will display an error message.

4.1.3.4. Receive message (viewing received messages)

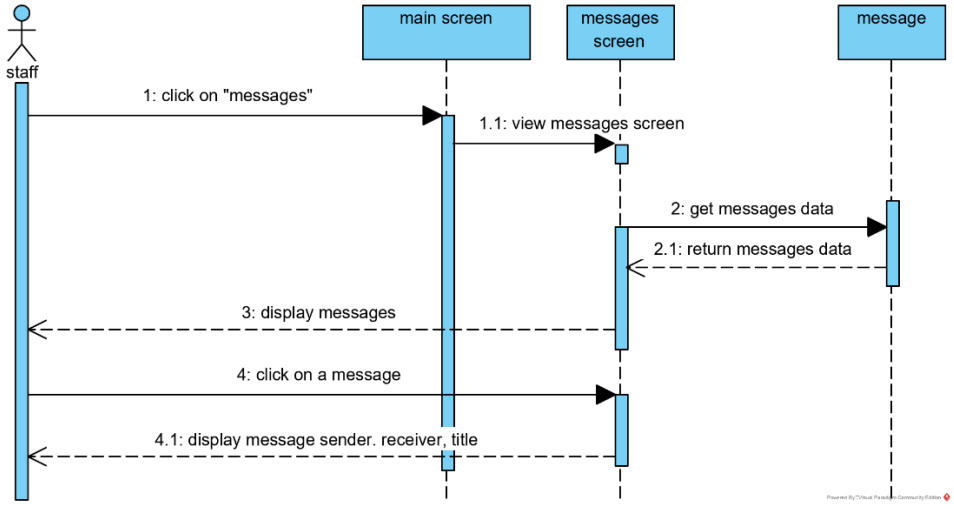


Figure 4.1-13 Staff Receive Message

Staff can view a message. To do that, the staff click "messages". The system will display all received messages including title, date, sender, and content.

4.1.4. Student

4.1.4.1. Register

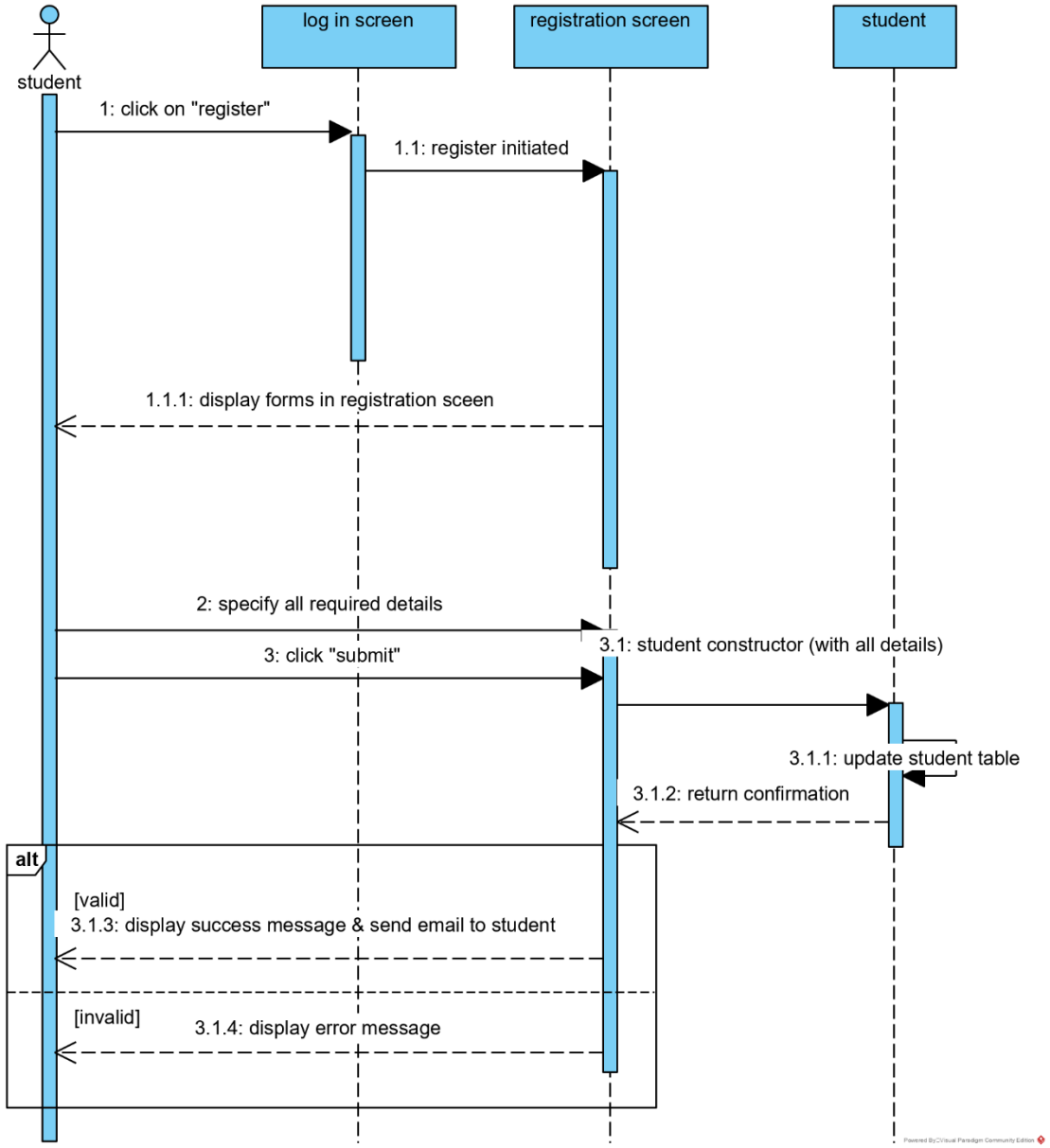


Figure 4.1-14 Student Register

Student has to register before being able to do anything in the system. To do that, the student clicks on “register”. The system will display registration screen. The student will specify all their data and click “submit”. If validation was successful, the student table will be updated and the system will display success message. However, if not valid, the system will display an error message.

4.1.4.2. Send message

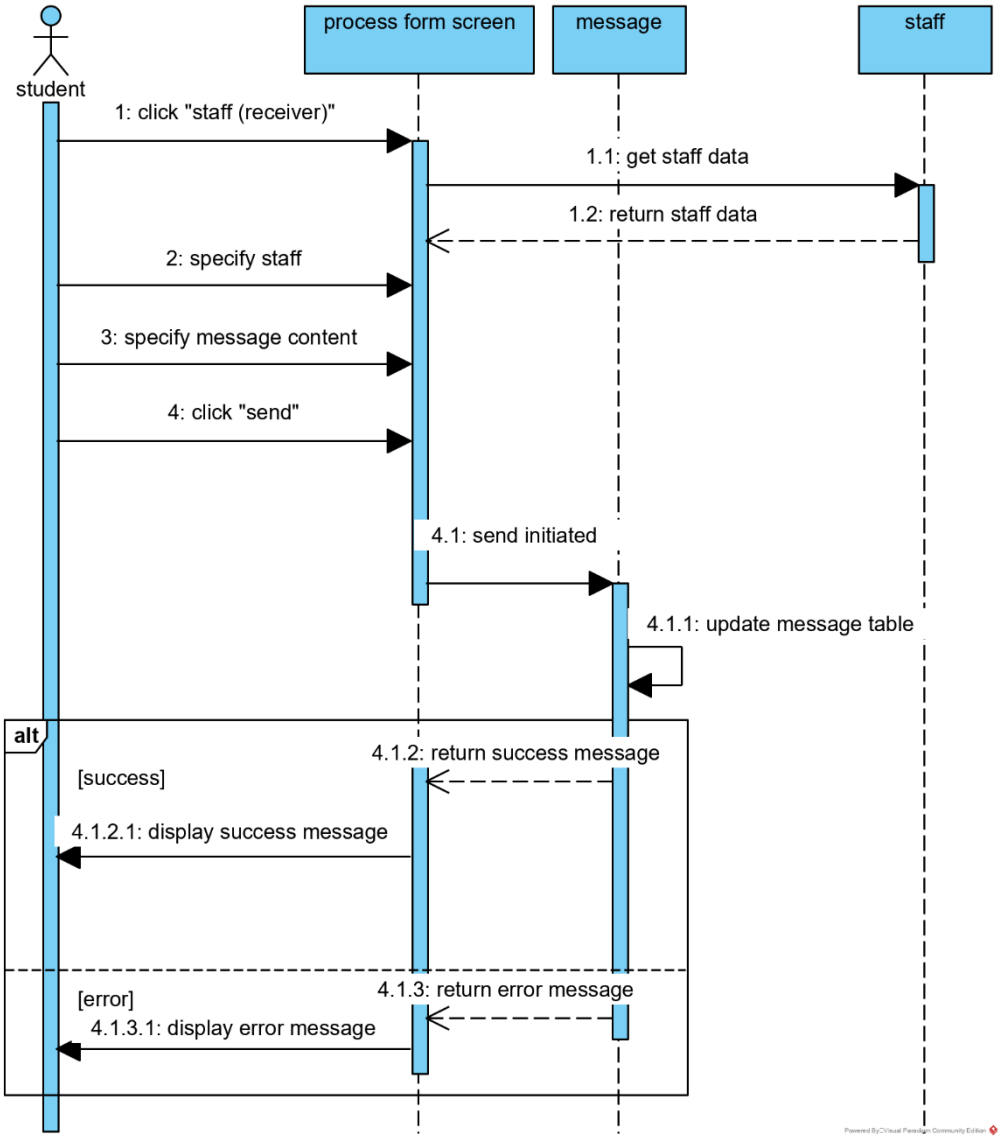


Figure 4.1-15 Student Send Message

Student can send a message. To do that, the student specifies the receiver from a list of staff responsible of the node. Then, the student specifies the content of the message and click “send”. If validation was successful, system will display success message. However, if not valid, the system will display an error message.

4.1.4.3. Receive message (viewing received messages)

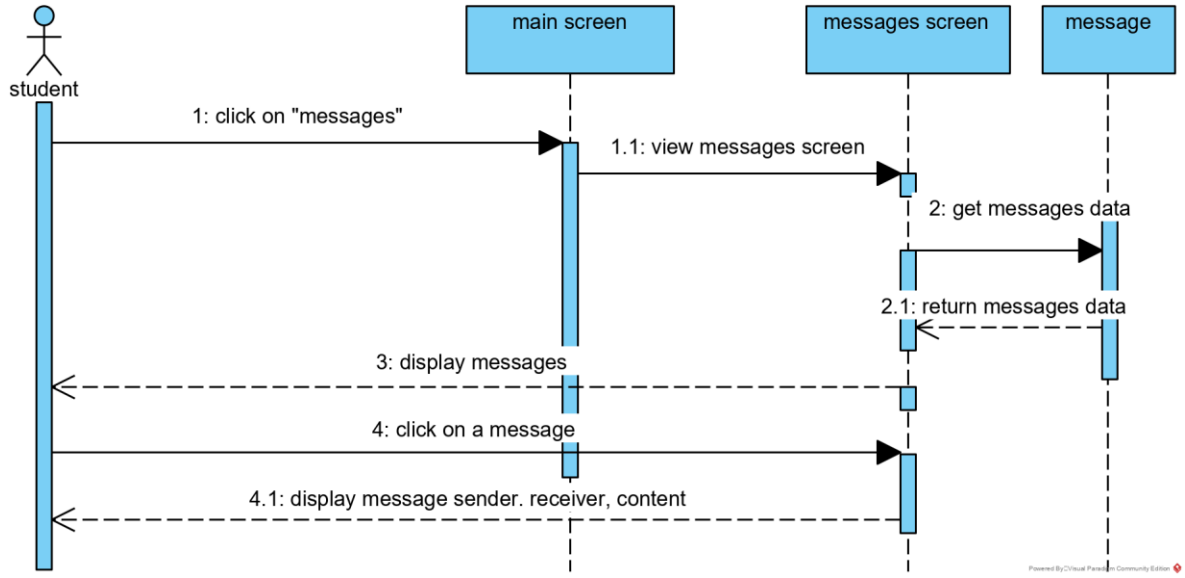


Figure 4.1-16 Student Receive Message

Student can view a message. To do that, the student click “messages”. The system will display all received messages including sender, title, date, and content

4.2. Activity diagram

Activity diagram describe the behaviour of a certain scenario in the system depicted as a flow. It is a sequential description of actions and flow of control in a system. (Introduction to the Diagrams of UML 2.X, n.d.)

4.2.1. General user

4.2.1.1. login

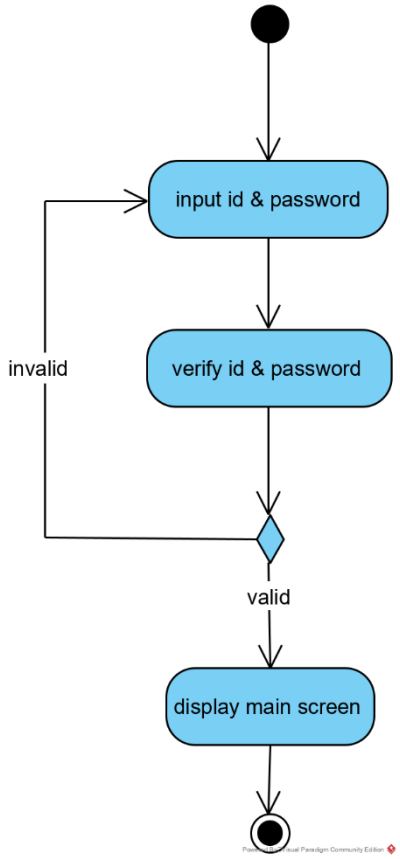


Figure 4.2-1 User Login

User needs to be logged in to use the international student admission system. To login, the user will enter id and password and click “Login” button. If validation was successful, user will be redirected to main screen. However, if not valid, the system will display an error message.

4.2.1.2. View profile & change password

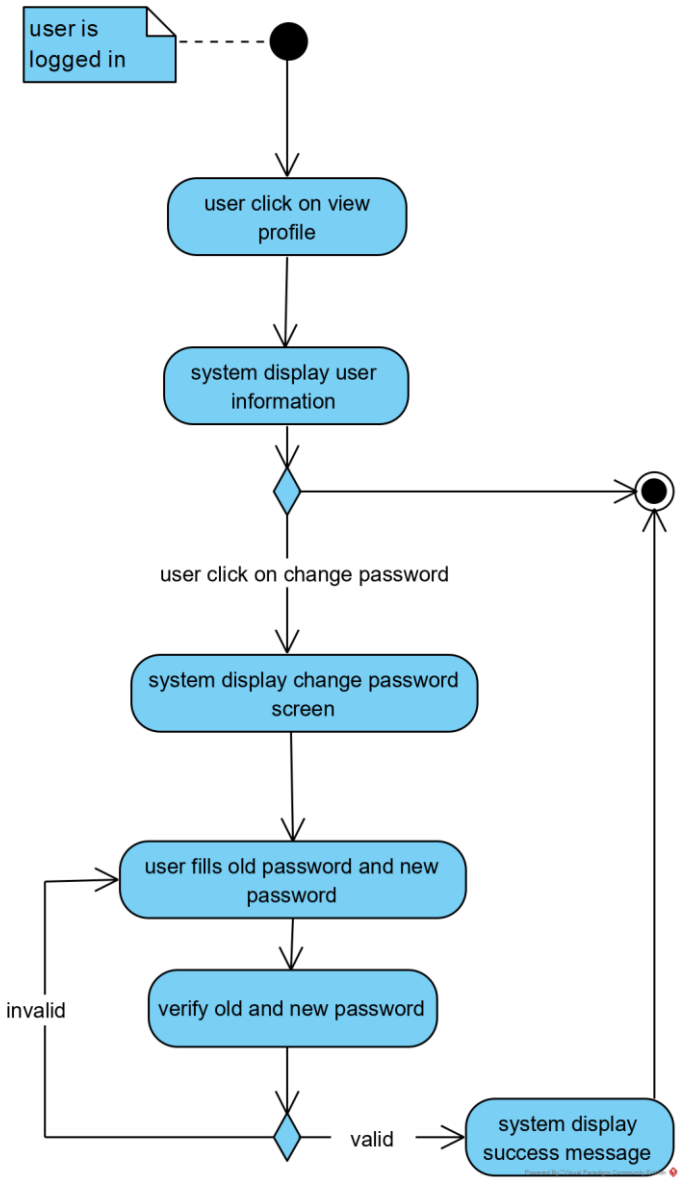


Figure 4.2-2 User View Profile & Change Password

When the user wants to view their profile, the user will click “view profile” button. The system will display user information. The user has the option to change their password. The user can opt to click on “change password” then provide old password and new password. If validation was successful, the system will display success message. However, if not valid, the system will display an error message.

4.2.1.3. Retrieve password

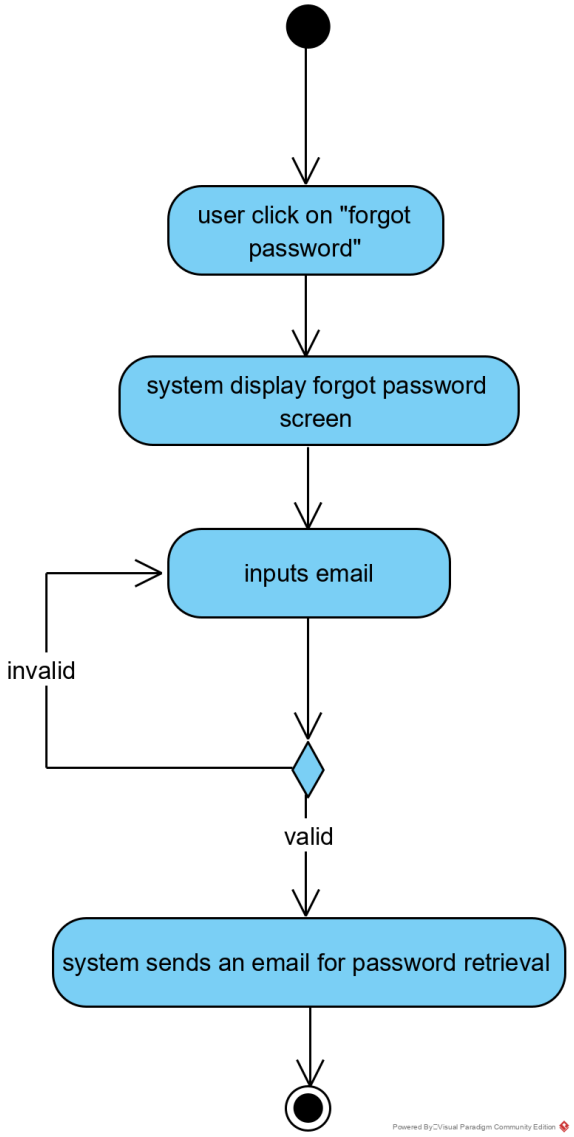


Figure 4.2-3 User Retrieve Password

When the user forgets the password, the user will click “forgot password”. The system will display the forgot password screen. The user has to provide their registered email. If validation was successful, user will be sent an email for recovery. However, if not valid, the system will display an error message

4.2.2. Admin

4.2.2.1. Register staff

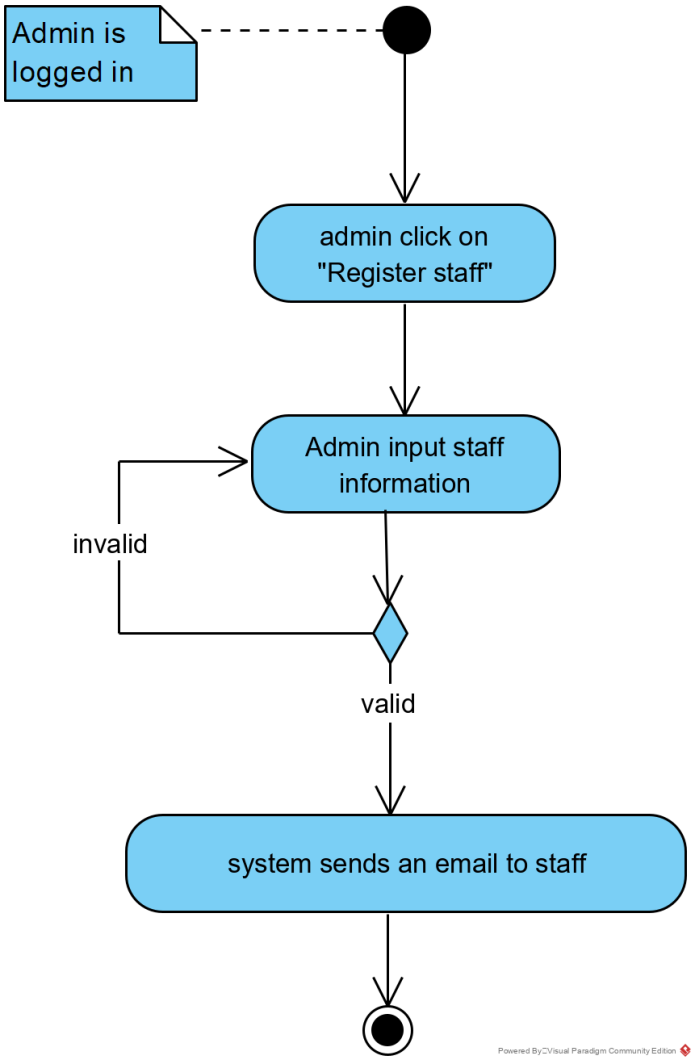


Figure 4.2-4 Admin Register Staff

the admin clicks on “register staff” and specify the name and email. If validation was successful, system will display success message and also sends an email to staff. However, if not valid, the system will display an error message.

4.2.2.2. Create workflow

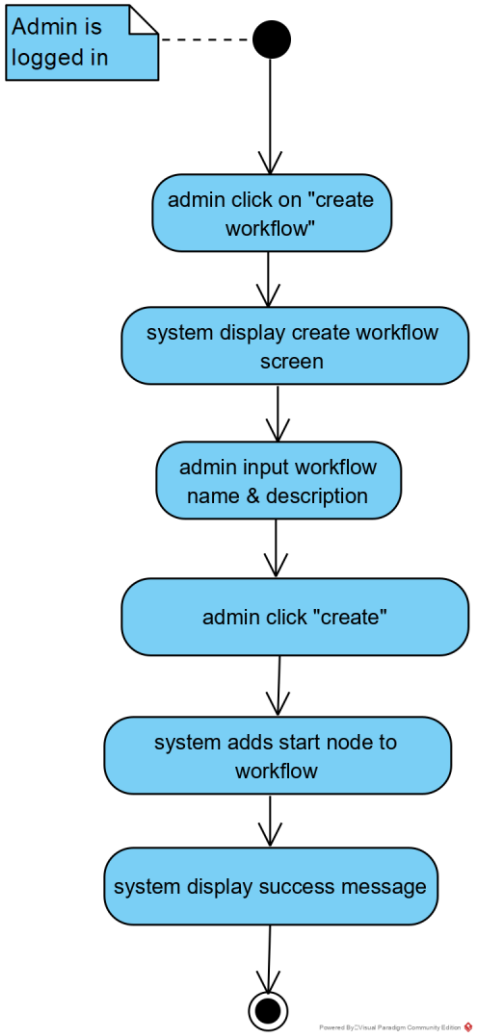


Figure 4.2-5 Admin Create Workflow

Admin can create workflow. To do that, the admin clicks on “create workflow”. The system will display create workflow screen. From there, the admin specifies the name and description of the workflow and clicks on create. If validation was successful, system will display success message. However, if not valid, the system will display an error message.

4.2.2.3. Add node

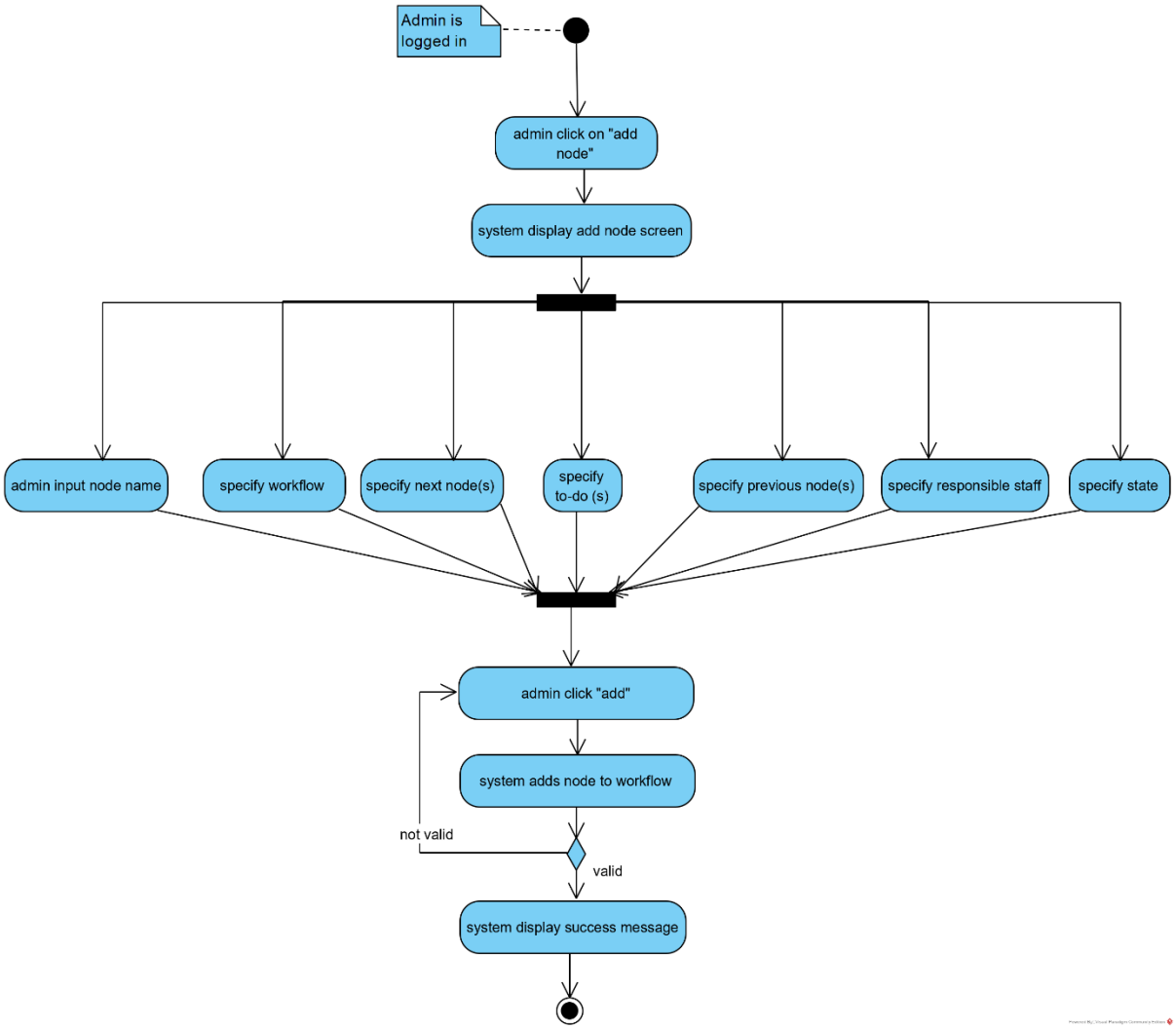


Figure 4.2-6 Admin Add Node

Admin can add a node to a workflow. To do that, the admin clicks on “add node”. The system will display create node screen. From there, the admin specifies the name, state, workflow, to-do list, list of responsible staff, list of previous nodes, and list of next nodes. Finally, clicks on add node. If validation was successful, system will display success message. However, if not valid, the system will display an error message.

4.2.2.4. Modify node

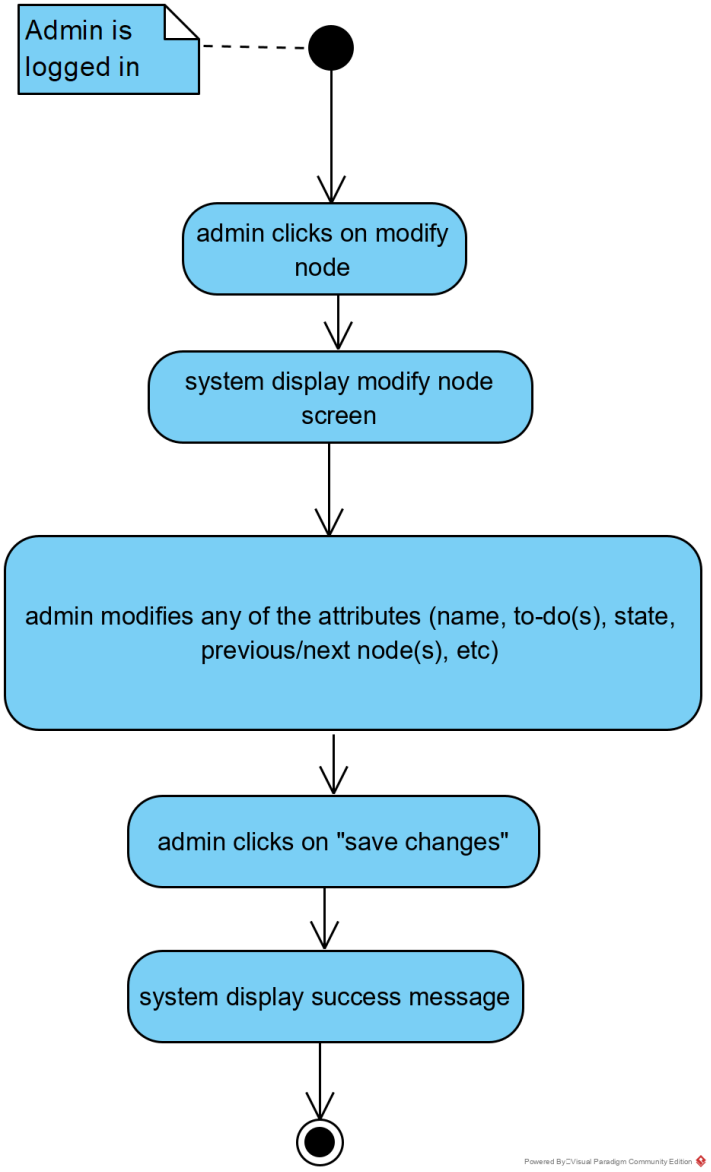


Figure 4.2-7 Admin Modify Node

Admin can modify a node to a workflow. To do that, the admin clicks on “modify node”. The system will display modify node screen. From there, the admin can modify the name, state, workflow, to-do list, list of responsible staff, list of previous nodes, and list of next nodes. Finally, clicks on save changes. If validation was successful, system will display success message. However, if not valid, the system will display an error message.

4.2.2.5. Delete node

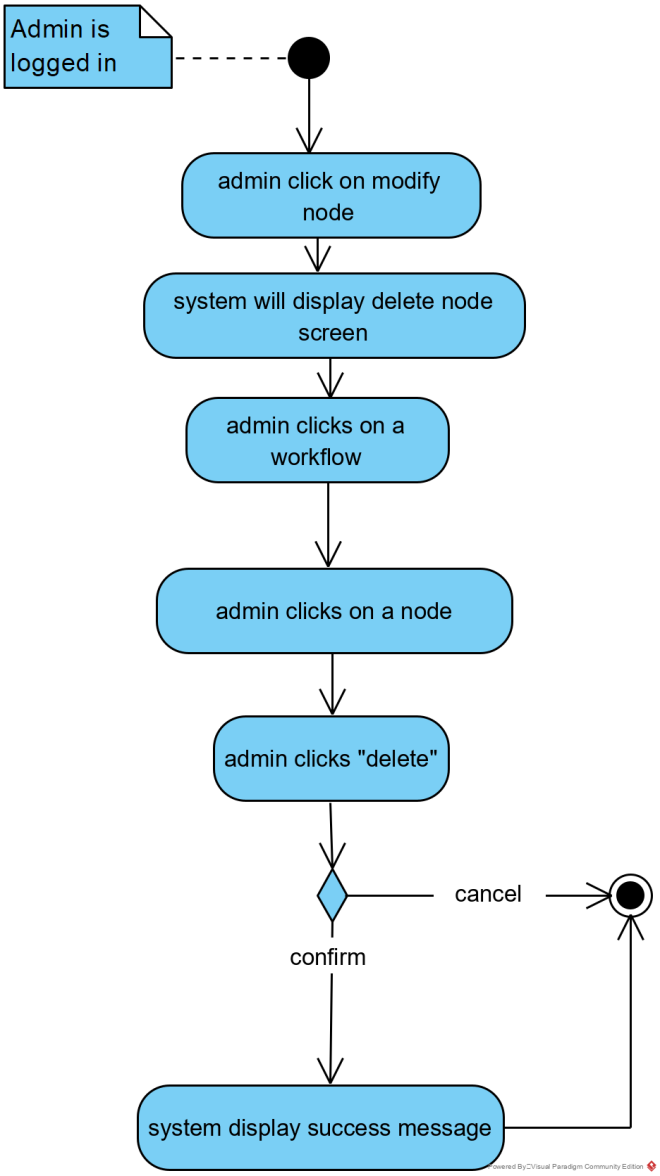


Figure 4.2-8 Admin Delete Node

Admin can modify a node to a workflow. To do that, the admin clicks on “modify node”. The system will display modify node screen. From there, the admin has to specify which node to delete, then clicks on “delete”. The system will ask for confirmation because it’s a risky action. If ok was clicked and If validation was successful, system will display success message. However, if not valid, the system will display an error message.

4.2.2.6. View system statistics & staff profile

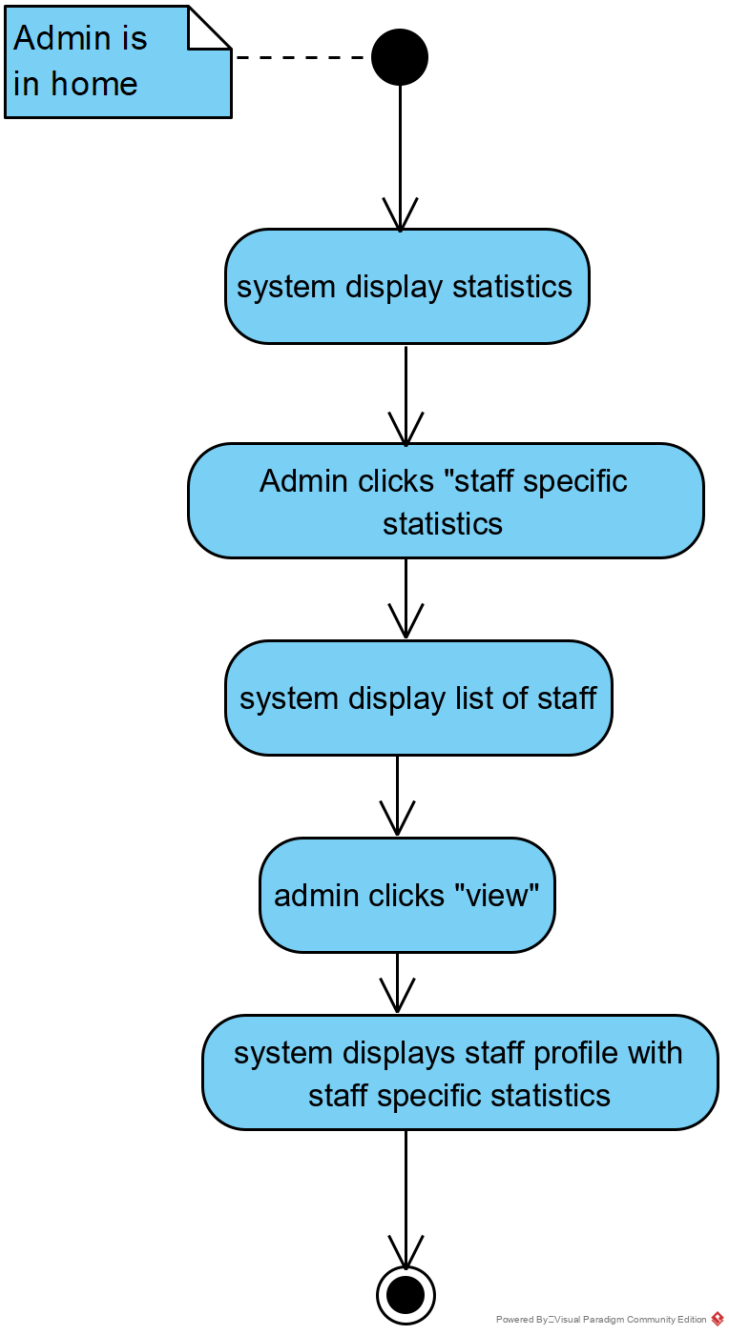


Figure 4.2-9 Admin View System Statistics & Staff Profile

Admin can view system statistics right from the home screen, after clicking on view staff specific statistics, the admin can specify a specific staff to view their profile and statistics.

4.2.3. Staff

4.2.3.1. Select node & view student profile

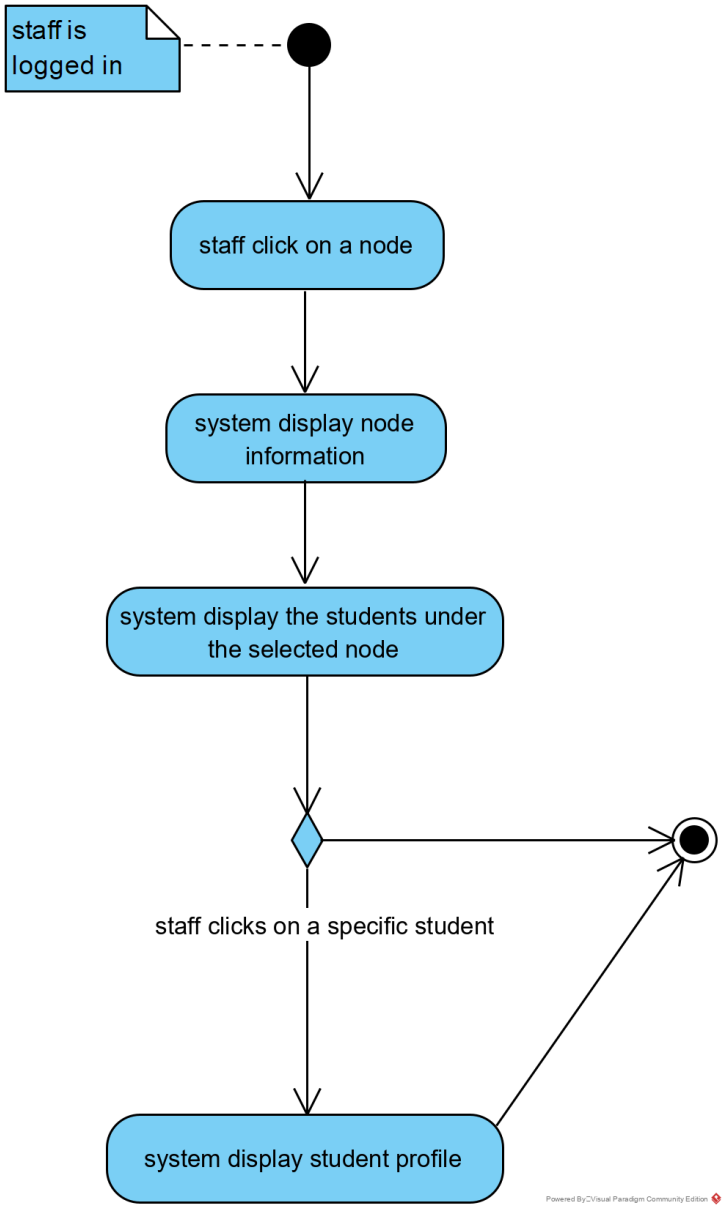


Figure 4.2-10 Admin View System Statistics & Staff Profile

staff selects any node that under their responsibility. The system will display both node information and all students under the selected node. Staff has the option to view a student's profile by clicking on it.

4.2.3.2. process student

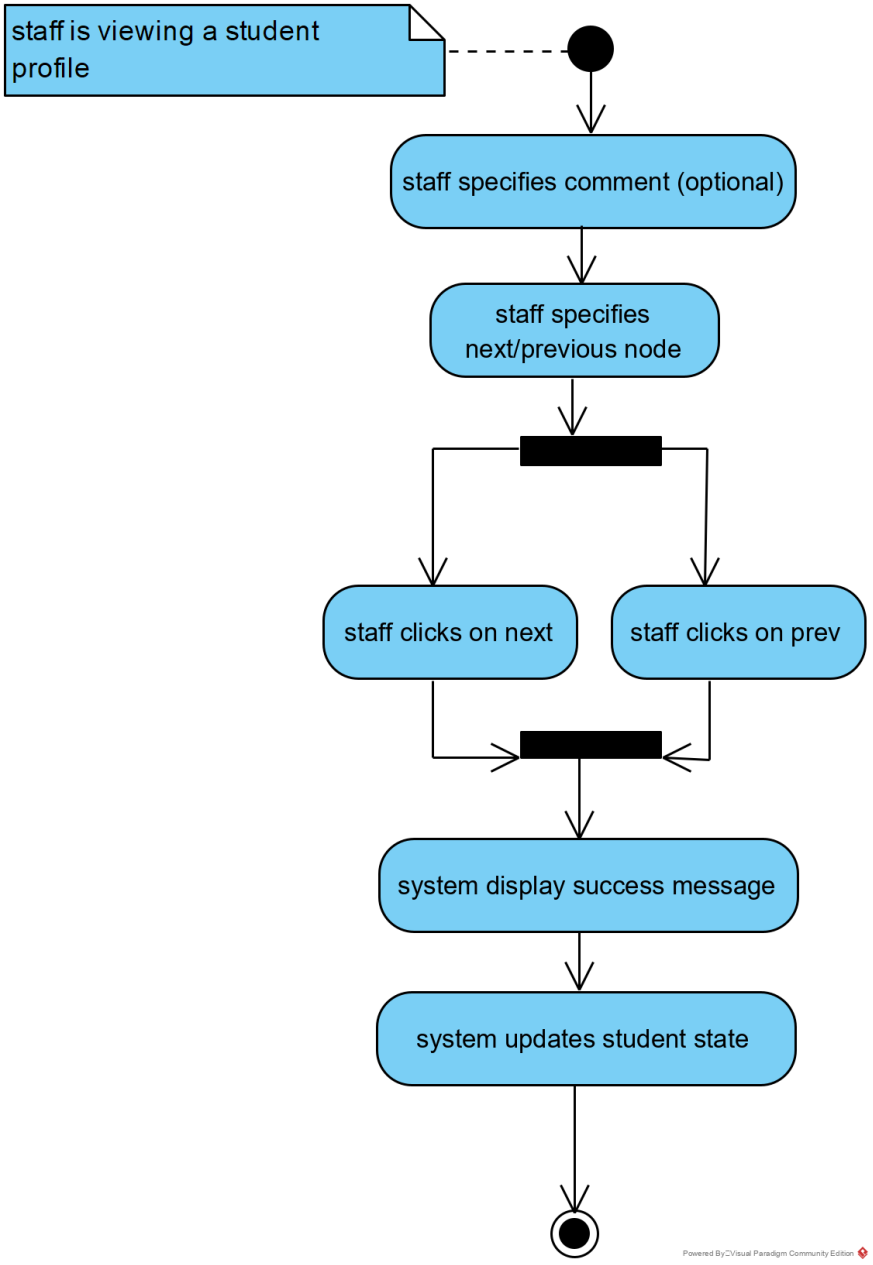


Figure 4.2-11 Staff Process Student

While the staff is viewing a student, the staff has an option to add a comment, after that the staff can click on either next or prev. When the staff clicks either, the system will display a message and will update the student state accordingly.

4.2.3.3. send message

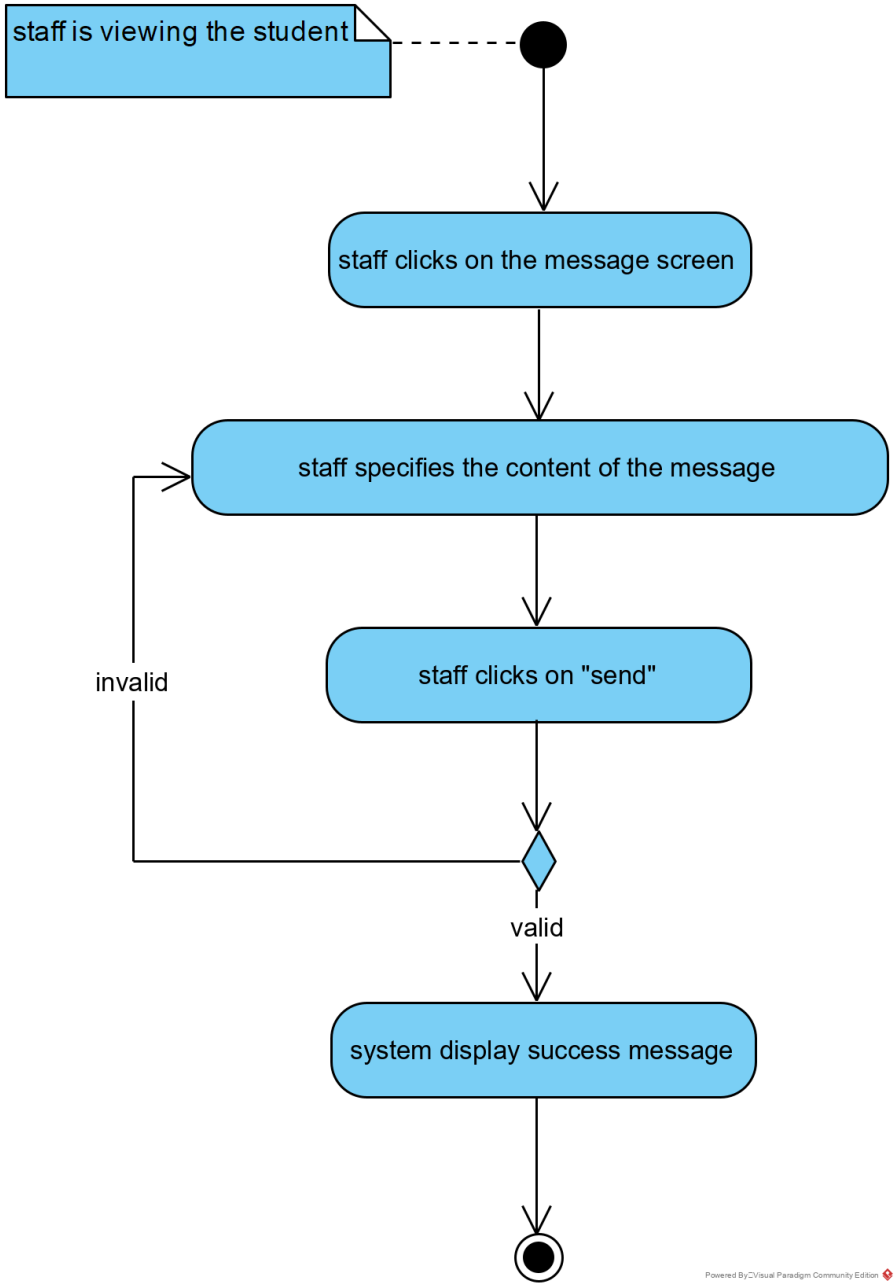


Figure 4.2-12 Staff Send Message

Staff clicks on the message screen. The system will just change focus to that portion of the screen. The staff will write the content of the message and then click send. If successfully validated the system will display success message. However, if it was not. The system will display error message.

4.2.4. Student

4.2.4.1. Register

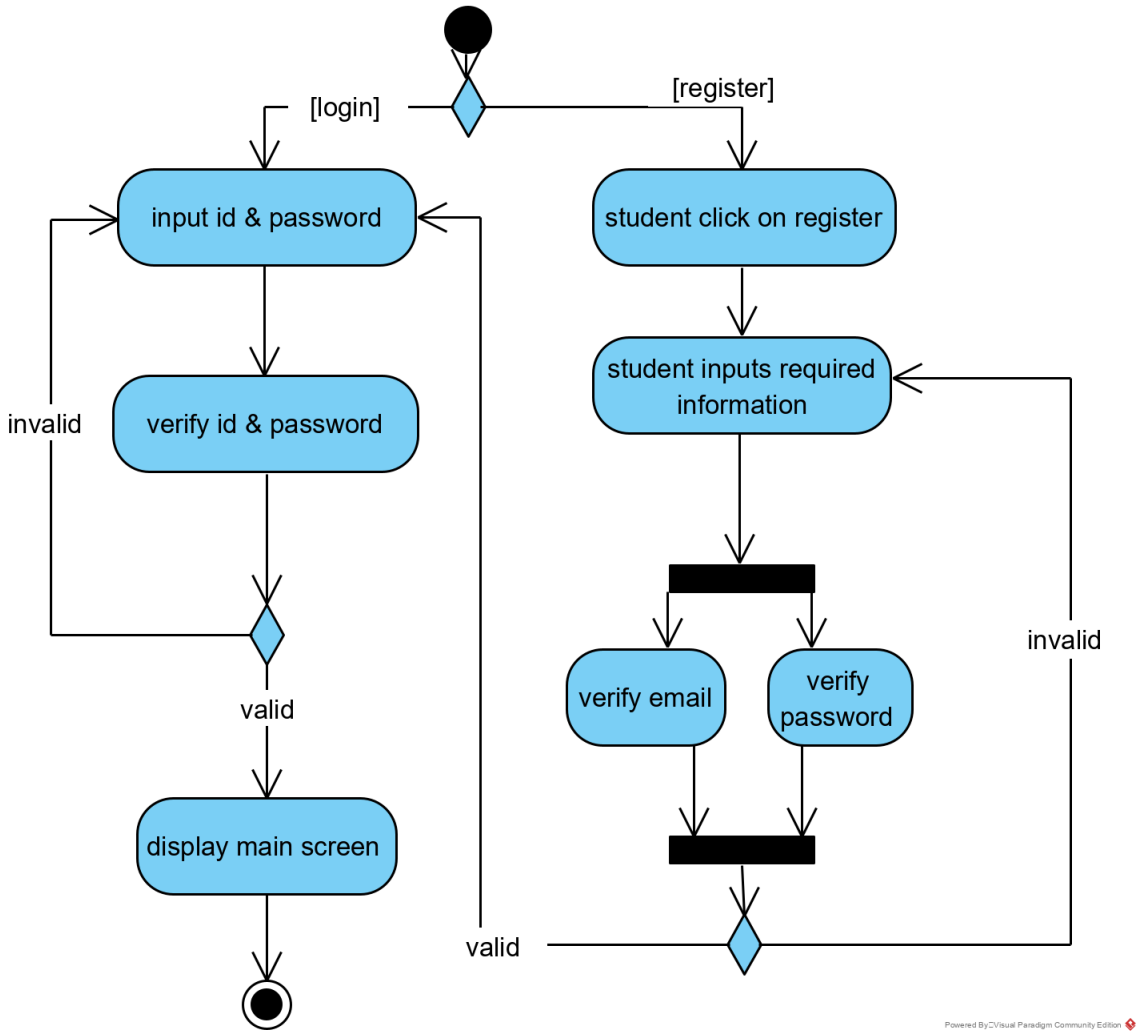


Figure 4.2-13 Student Register

Student has to register before being able to do anything in the system. To do that, the student clicks on “register”. The system will display registration screen. The student will specify all their data and click “submit”. If validation was successful, the student will be taken back to login. However, if not valid, the system will display an error message.

4.2.4.2. Send message

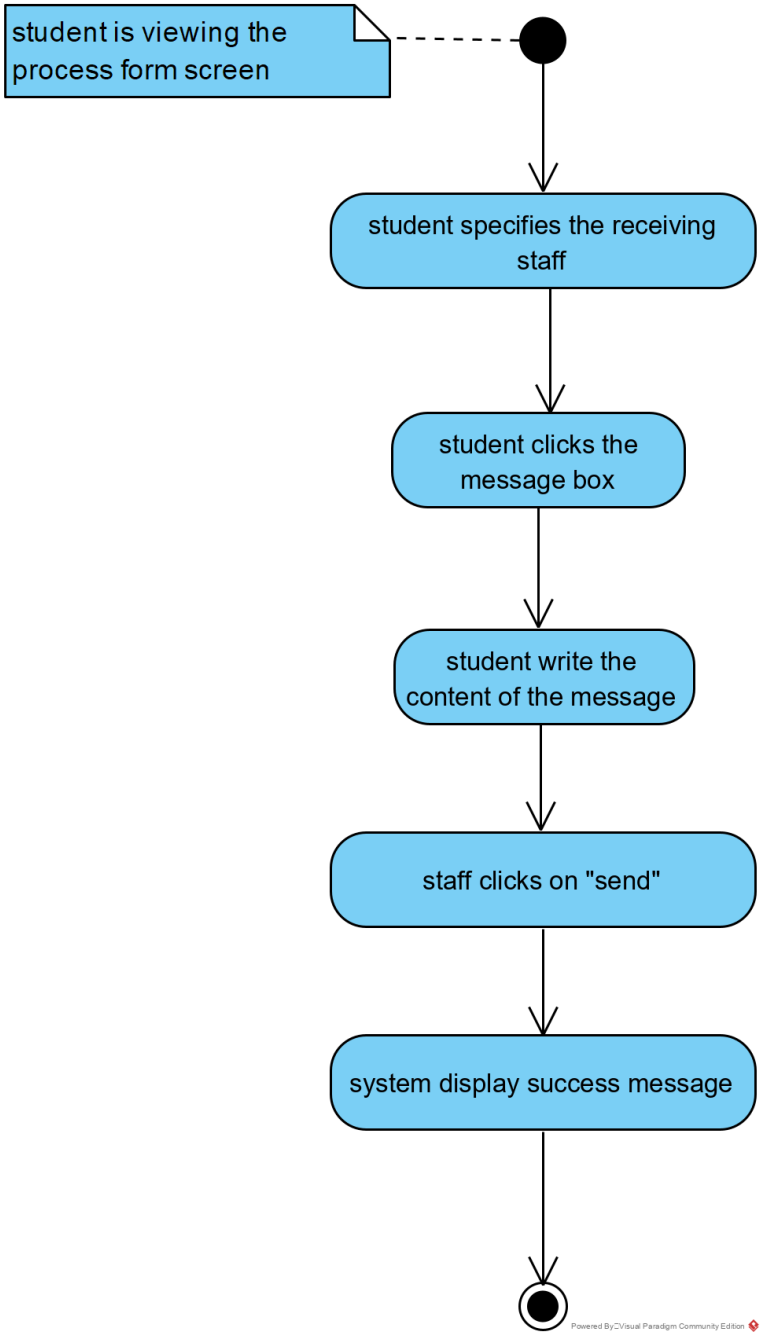


Figure 4.2-14 Student Send Message

The student can message staff. The student has to specify which staff. After that, the student specifies the content of the message and clicks send.

4.2.5. Staff and student (shared activity diagram)

4.2.5.1. View message

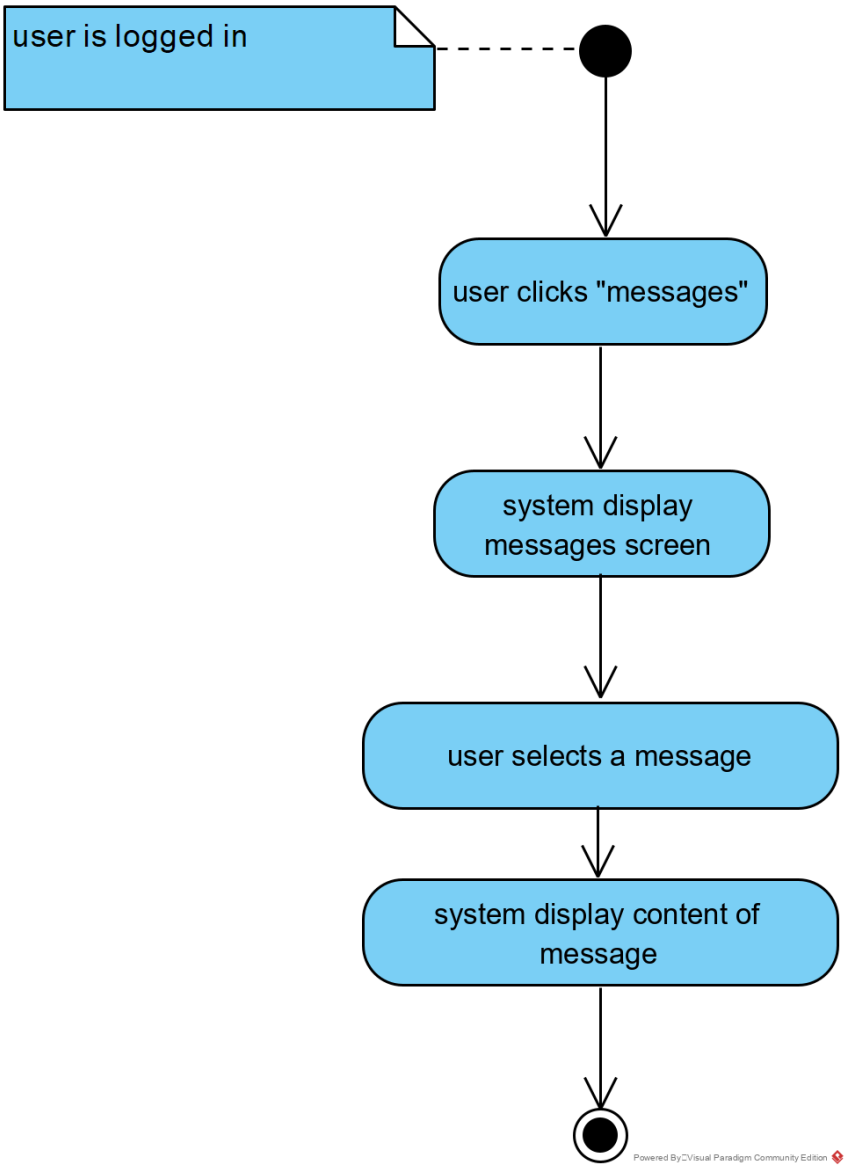


Figure 4.2-15 Student & Staff View Message

Both student and staff can view messages that they received. They click on messages from the main screen and the system will display the history of all received messages. Then they can select a message and the system will display the content of selected message.

4.3. State transition diagram

A state diagram is a type of behavioural diagram in the UML that shows transitions in system states or object states. A state transition contains present state and a next state(s) that are possible to transition to depending on the conditions that result into the change of state. (Introduction to the Diagrams of UML 2.X, n.d.)

4.3.1. Student status state diagram

The student state will be decided based on the node he is currently in. An example would be waiting “offer letter” approval. If approved the student will move to the next node if rejected the student will be brought back to previous node.

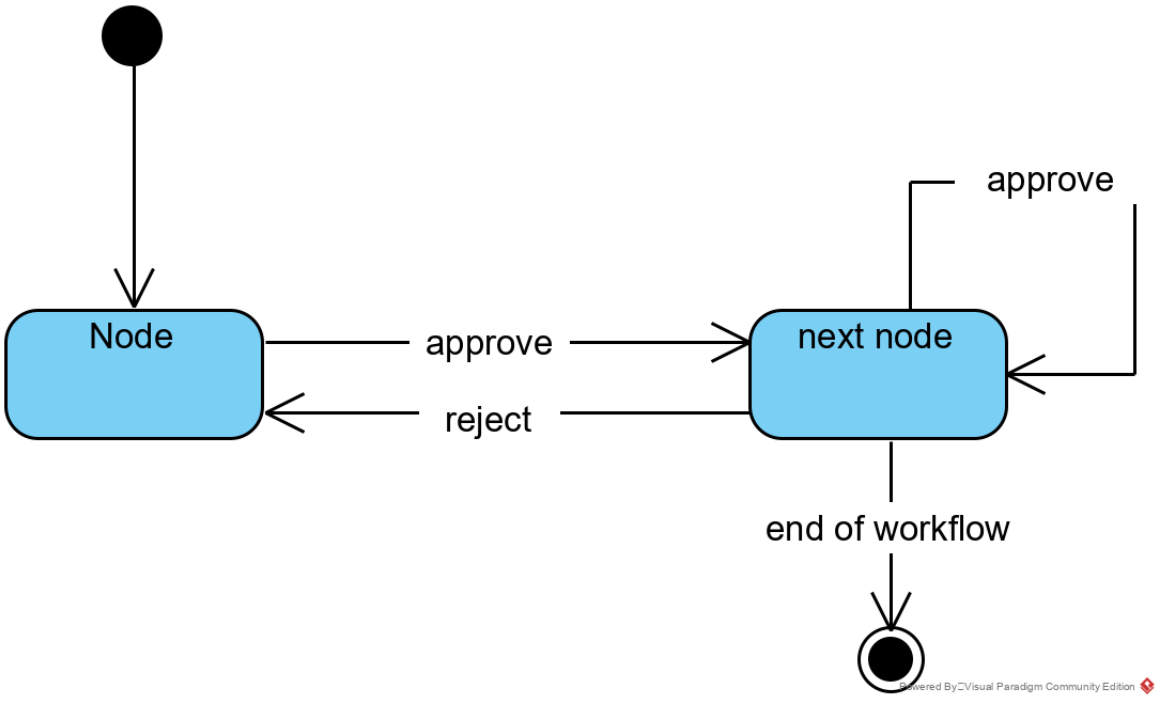


Figure 4.3-1 Student Status State Diagram

4.3.2. Admin system state diagram

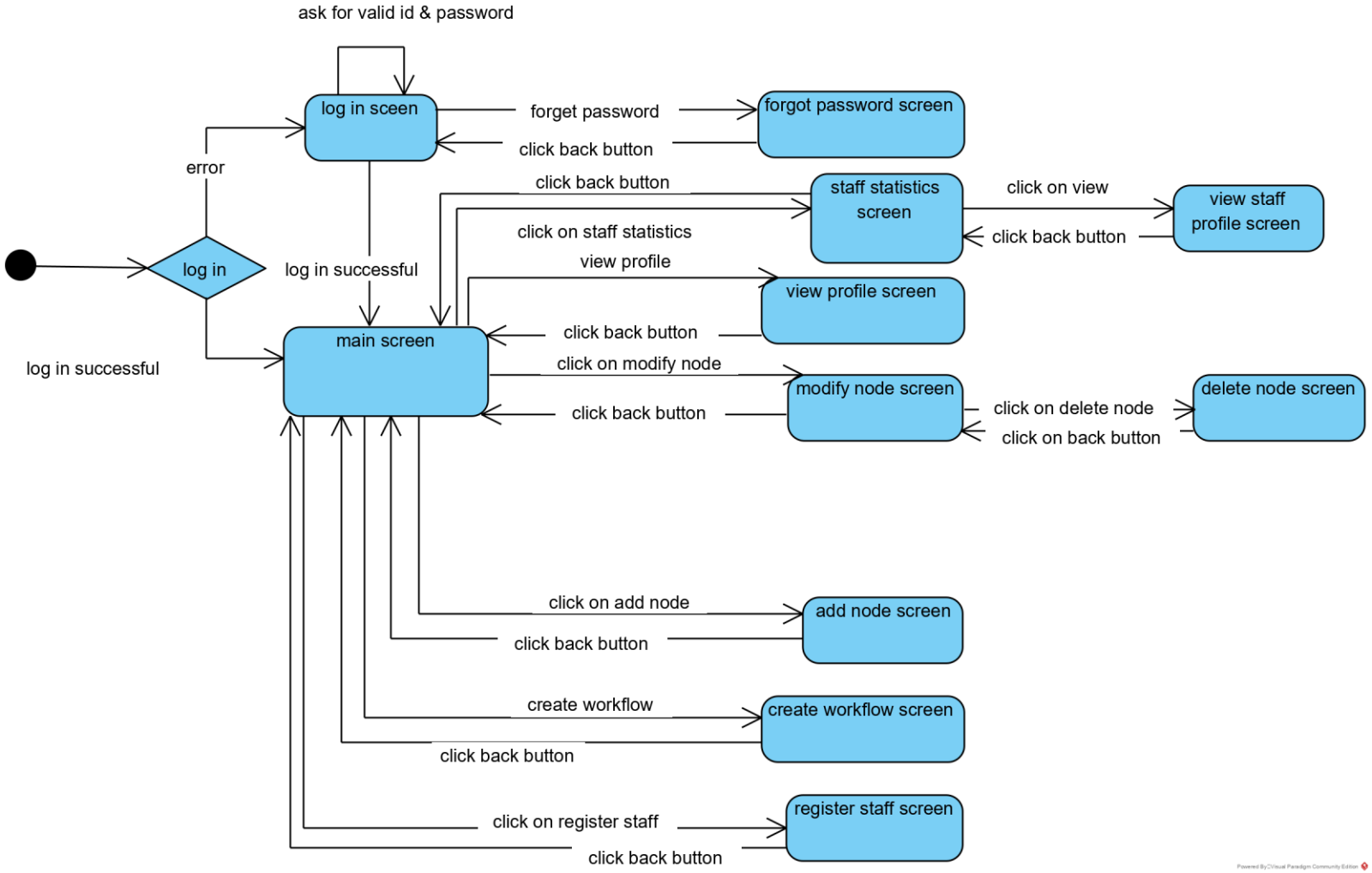


Figure 4.3-2 Admin System State Diagram

4.3.3. Staff system state diagram

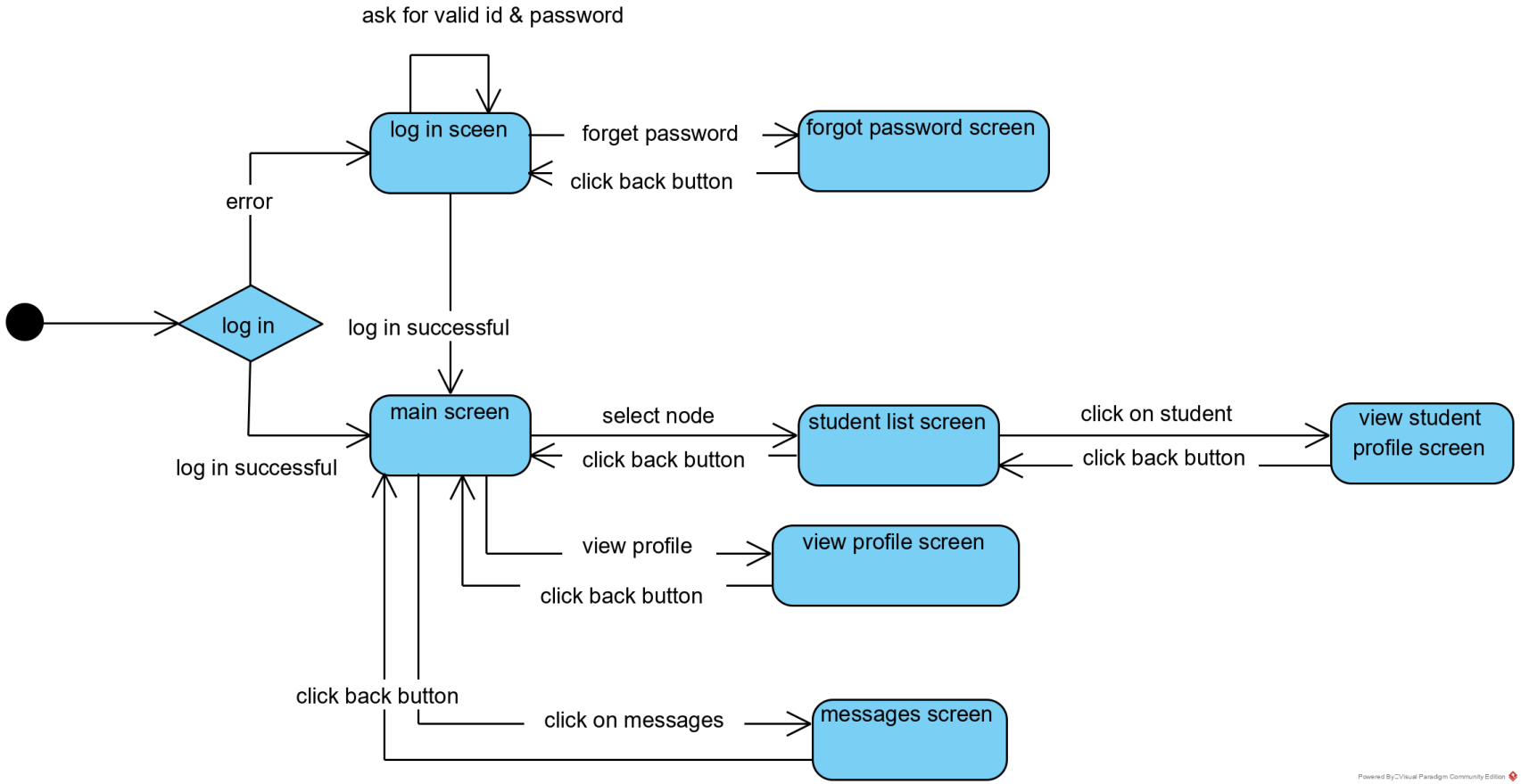


Figure 4.3-3 Staff System State Diagram

4.3.4. Student system state diagram

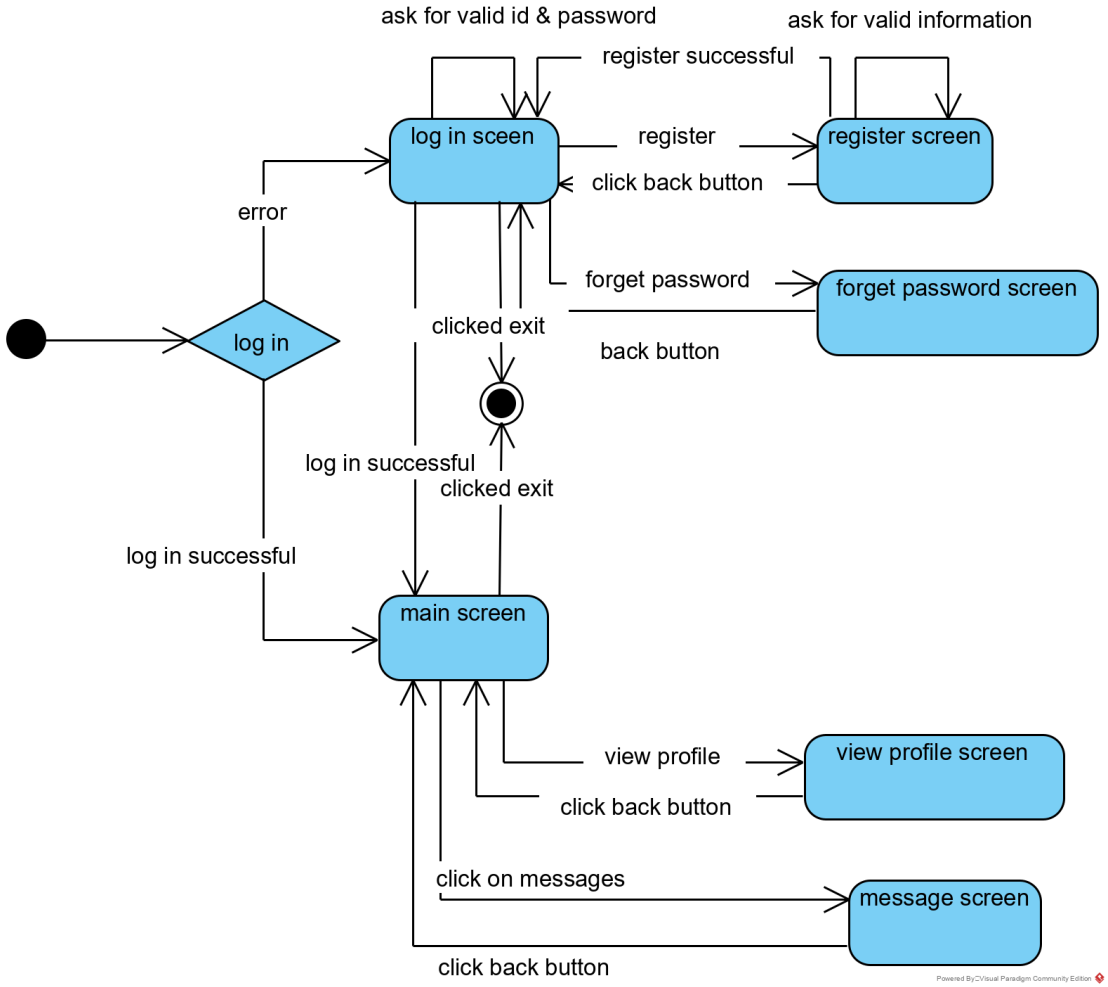


Figure 4.3-4 Student System State Diagram

5. Implementation

5.1. FYP part two Gantt char

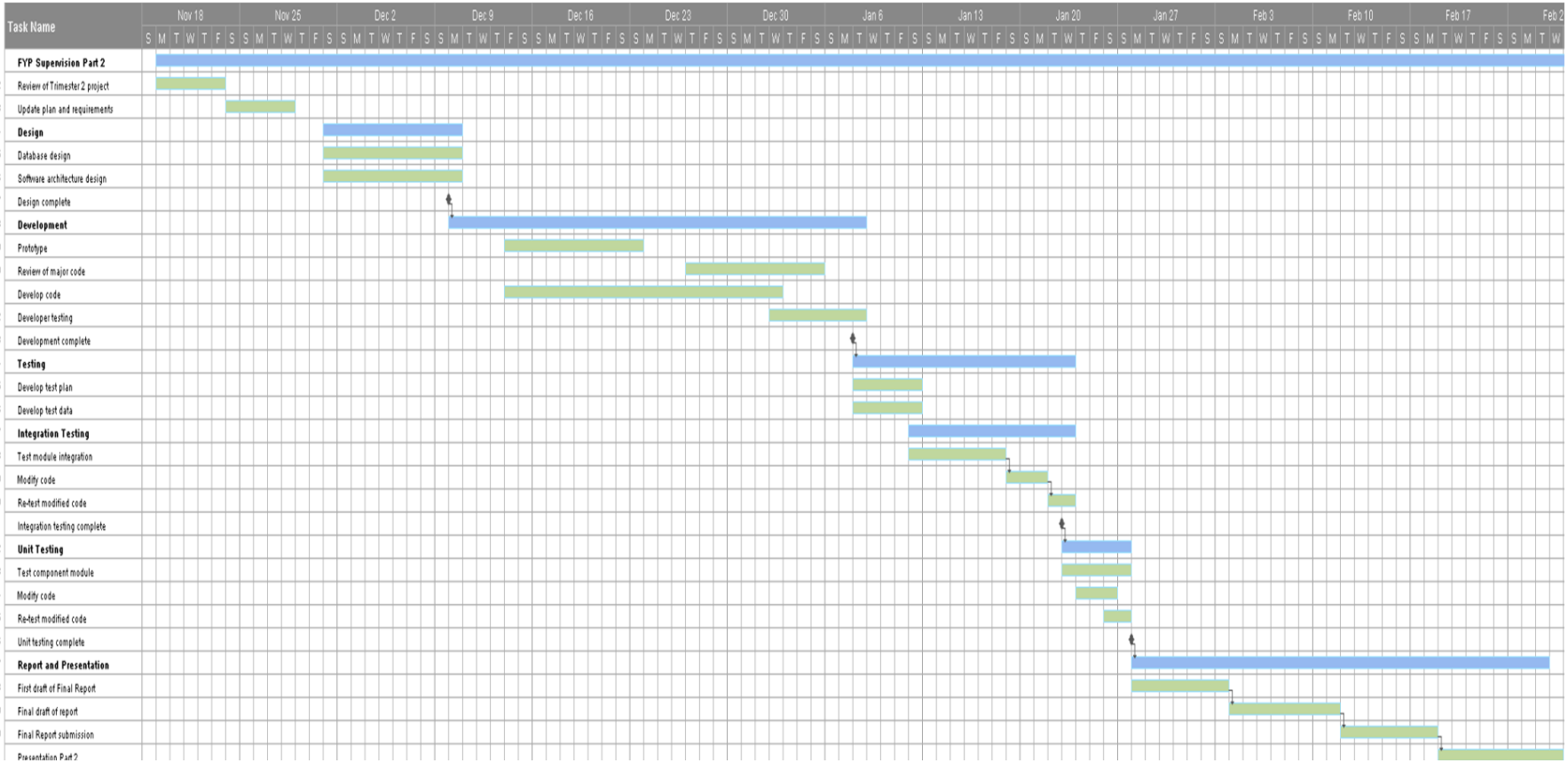


Figure 5.1-1 Gantt chart projection for FYP 2

5.2. Gantt chart brief description

5.2.1. FYP Supervision Part Two

First step of FYP 2 is started by reviewing this projected Gantt chart. Then, any feedback received from FYP1 is applied and discussed

5.2.2. Design

Implementing the design of ERD in a real database and starting to assemble the building blocks of the architecture.

5.2.3. Development

Real development is carried out. Development lasts until the end of the project implementation.

5.2.4. Testing

Testing the whole system for various results

5.2.5. Report

The Final Year Project Report that includes documentation for the whole project which includes the project plan, background studies, requirements, designs, implementations, testing and conclusion is produced.

5.2.6. Milestone table FYP 2

Table 5.2-1 Milestone table FYP 2

Review of Trimester 2 Project Work	19th November 2019
Update Plan and Requirement	25th November 2019
Database Design	4th December 2019
Update Prototype	16th December 2019
Review Major Modules	30th December 2019
Development of code	6th January 2020
Testing	30th January 2020
Final report Draft	10th February 2020
Final Report	11th February 2020
System Completion	13th February 2020
Presentation	18th February 2020

5.3. System Specifications

This project was being developed and tested on the following specifications:

Table 5.3-1 System Specifications

Processor	Intel® core™ i5-7300HQ CPU @2.50 GHz 2.50 GHz
RAM	8 GB 1333 MHz DDR3
Storage	1TB
GPU	GTX1050
Laptop Model	MSI GL62M 7RDX
Operating System	Windows 10, 64-bit
IDE	Visual Studio Code, Android Studio with emulator
Database	Postgres pgAdmin4
Browser	Chrome (Version 80.0.3987)

5.4. System highlights

Some of the screens inside the system will be expanded on and discussed in this part. The focus is the screens that are integral to the system.

5.4.1. Website

5.4.1.1. Login

Login screens are always an integral part of any system. It's the usual login screen with an identifier – an email in this case and password. Both the Admin and the staff can access the system through this Login screen.

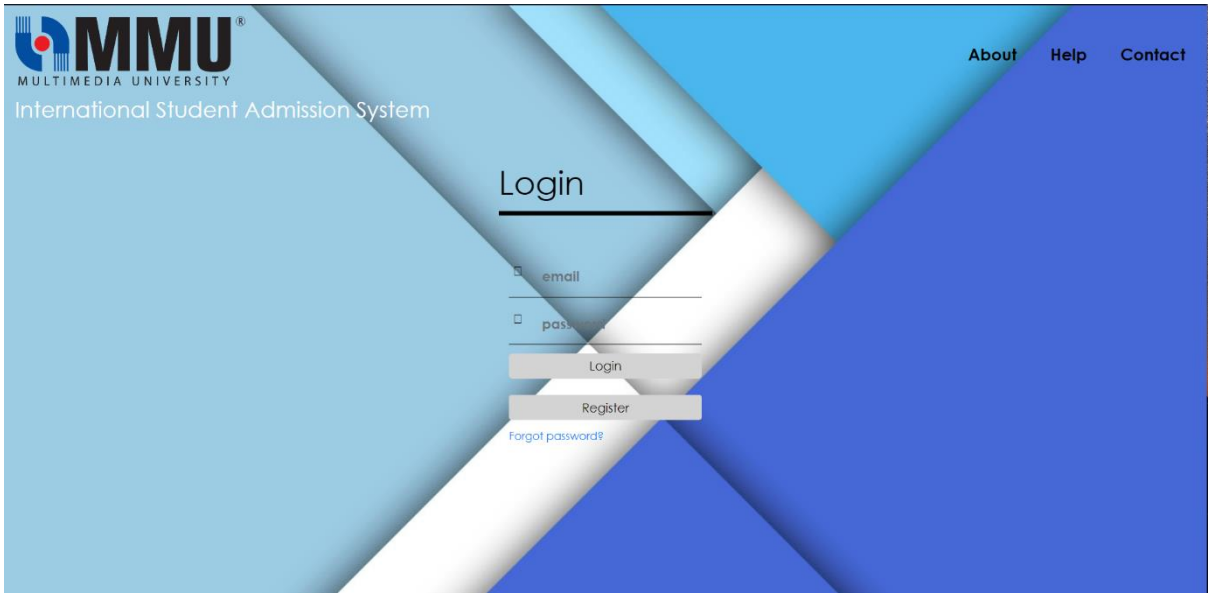


Figure 5.4-1 Login screen

5.4.1.2. Home

Home screens are always the centre of any system. Header, navigation, and accessibility of the whole system is structured here. Both the Admin and the staff have separate home screens

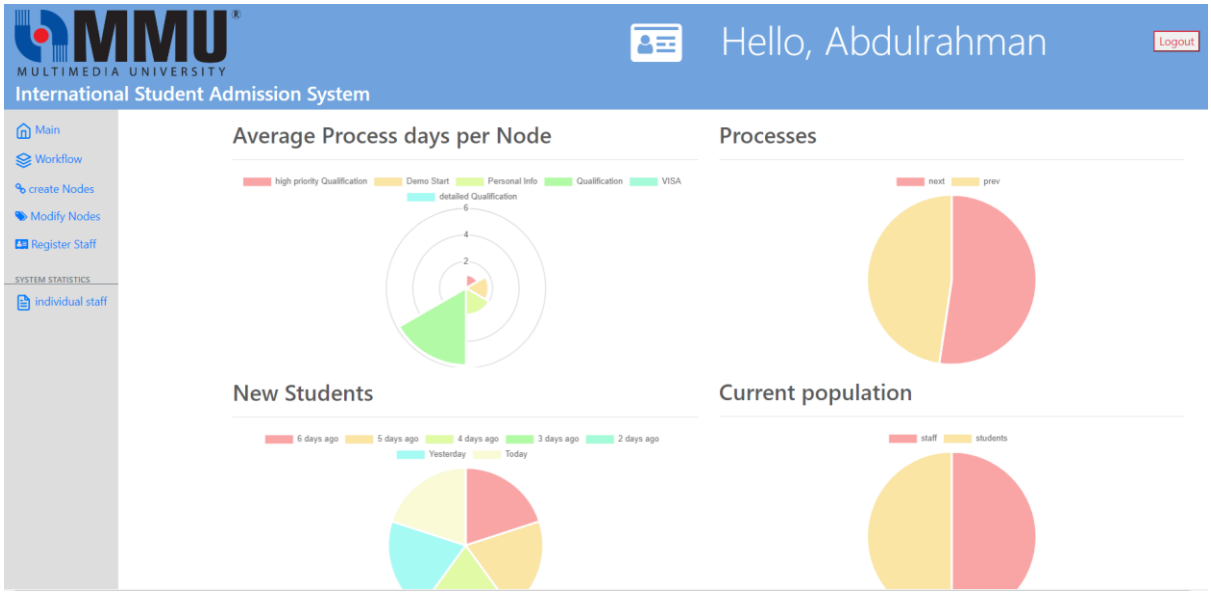


Figure 5.4-2 Admin home screen

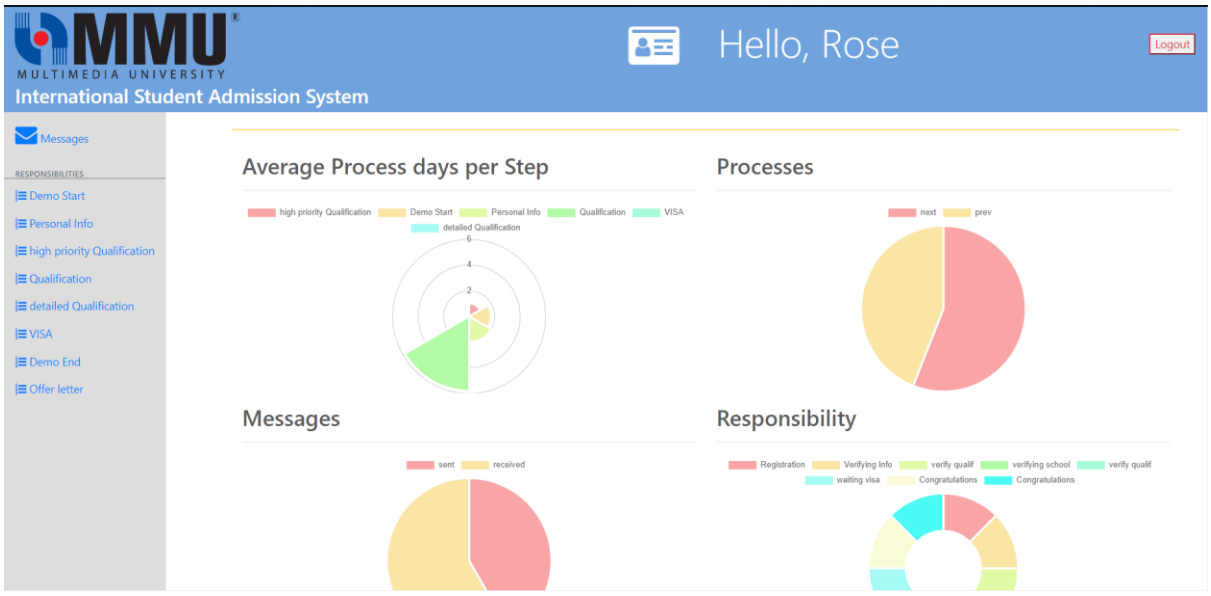


Figure 5.4-3 Staff home screen

5.4.1.3. Create workflow

The whole process starts by creating a workflow. A workflow is like a container in essence. This container can have any number of nodes in whatever order the admin wishes for it to be.

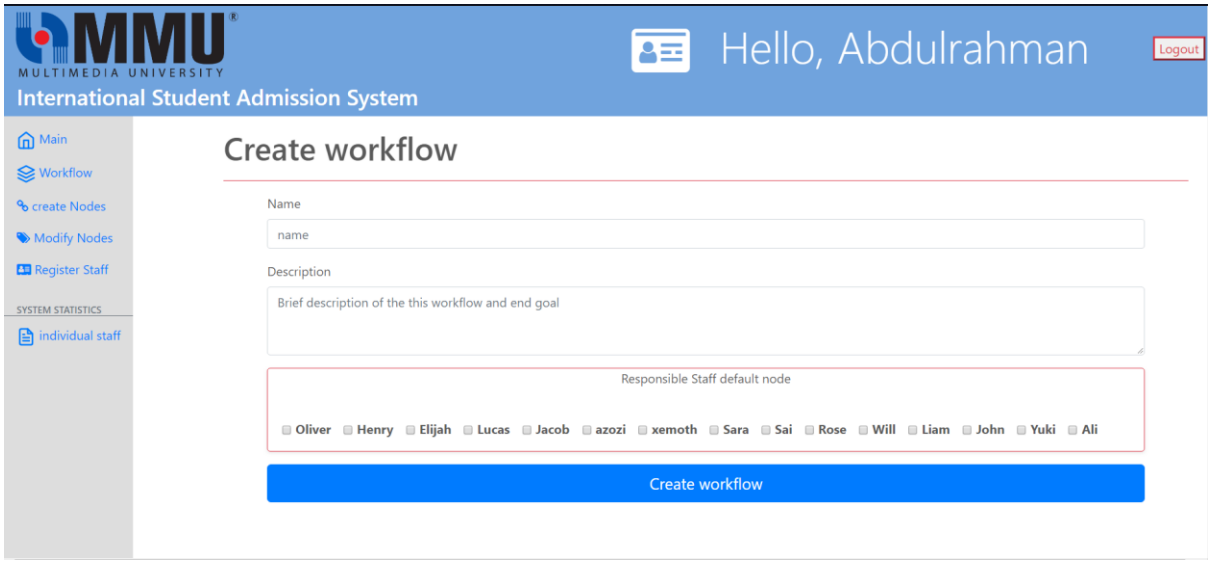


Figure 5.4-4 create workflow screen

5.4.1.4. Create node

A node is the building block for workflows. In order to design a workflow, an admin create nodes and organize them and decides the connectivity between them inside a workflow

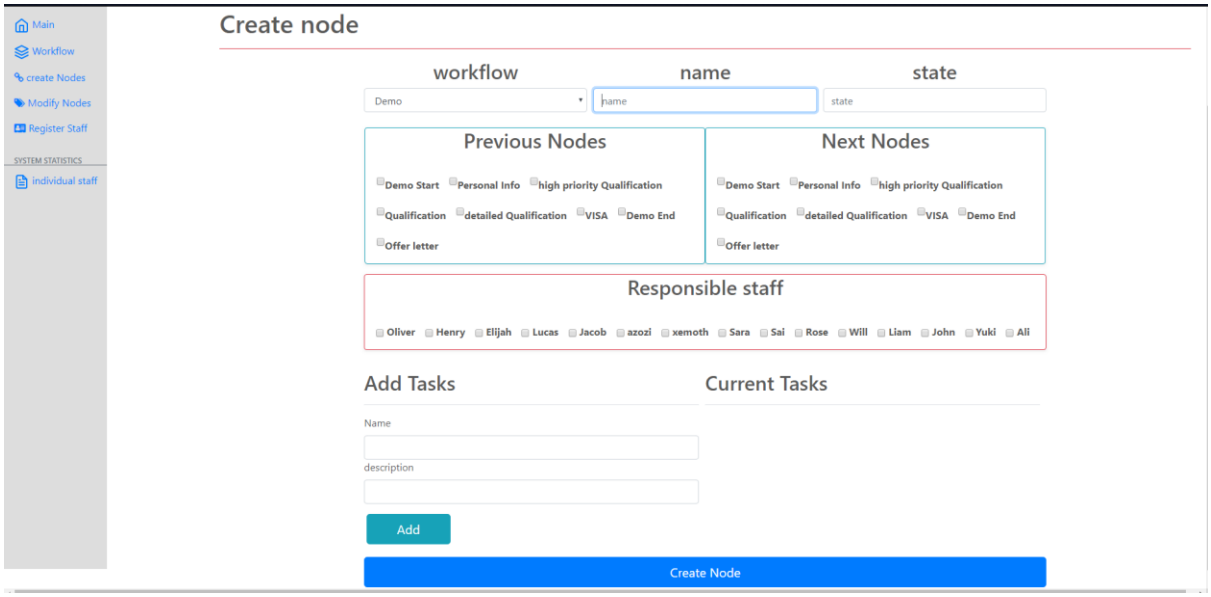


Figure 5.4-5 Create node screen

5.4.1.5. Modify node

A node is the building block for workflows. In order to help the admin in creating a proper workflow, an admin can modify nodes and organize them and decides the connectivity between them even after creation

5.4.1.6. Register Staff

In order for the staff to be inside the system, an admin has to register their data to the system

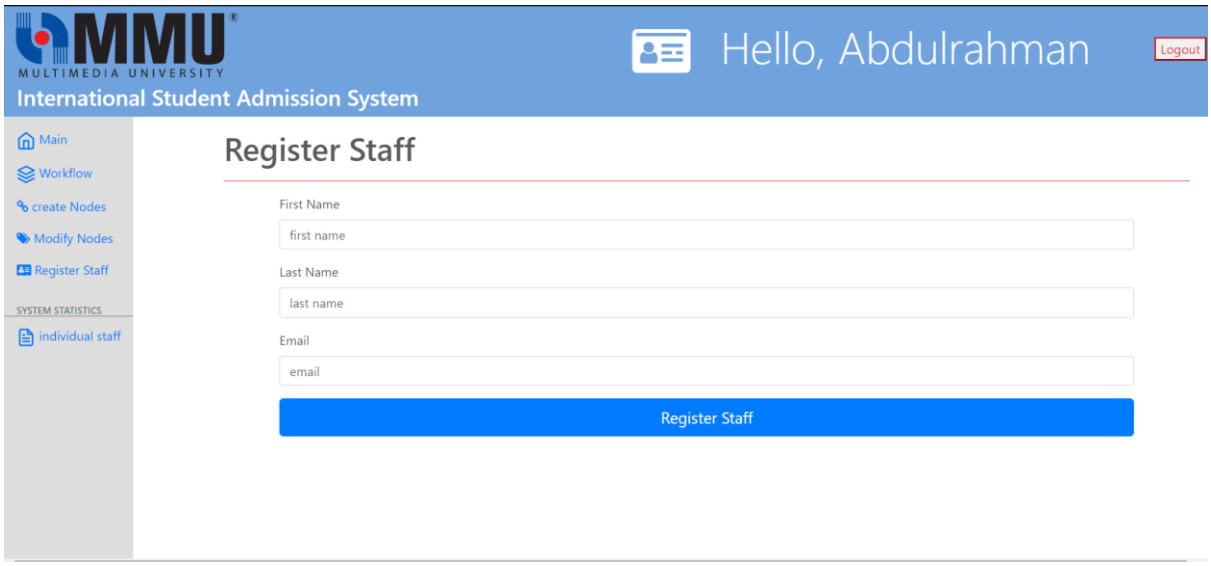


Figure 5.4-6 Register staff screen

5.4.1.7. Staff list

In order for the admin to view staff statistics, a list of all registered staff is displayed in this screen.

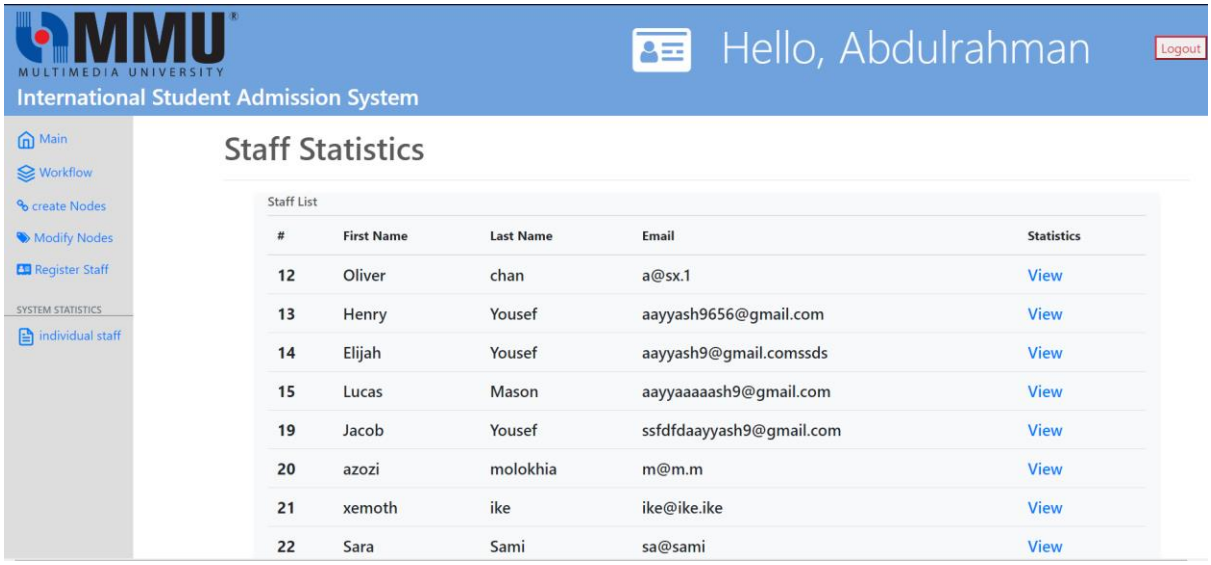


Figure 5.4-7 staff list screen

5.4.1.8. Individual staff statistics

In order for the admin to view all staff data and statistics, the admin clicks on view from the staff list and all data will be viewed for that specific staff

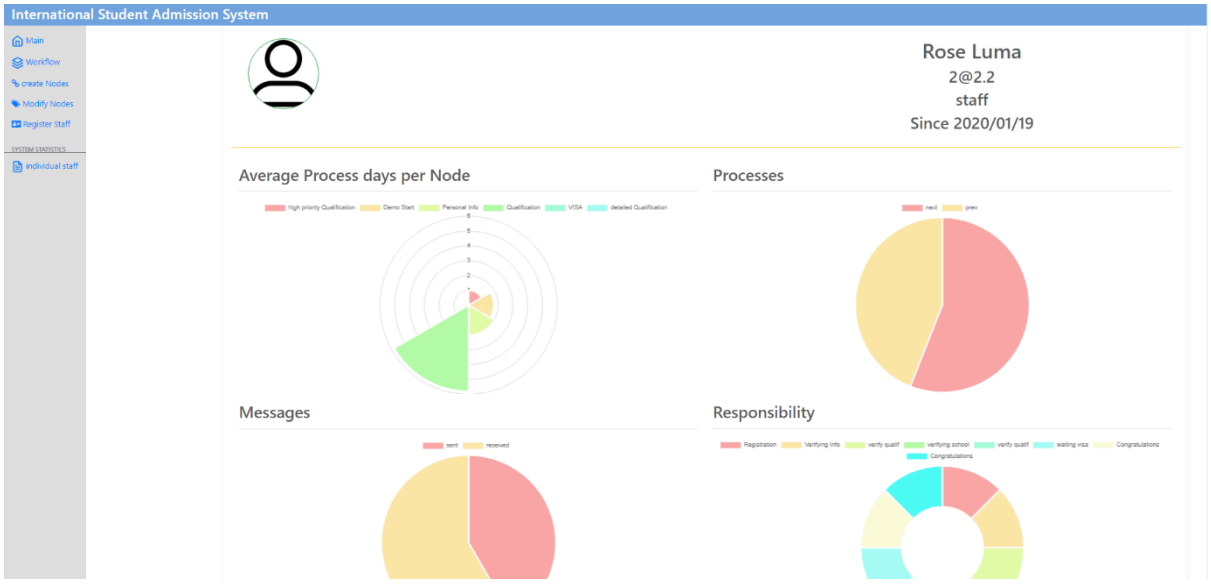


Figure 5.4-8 individual staff statistics screen

5.4.1.9. Messages

This is a basic inbox screen to facilitate the simple messaging system

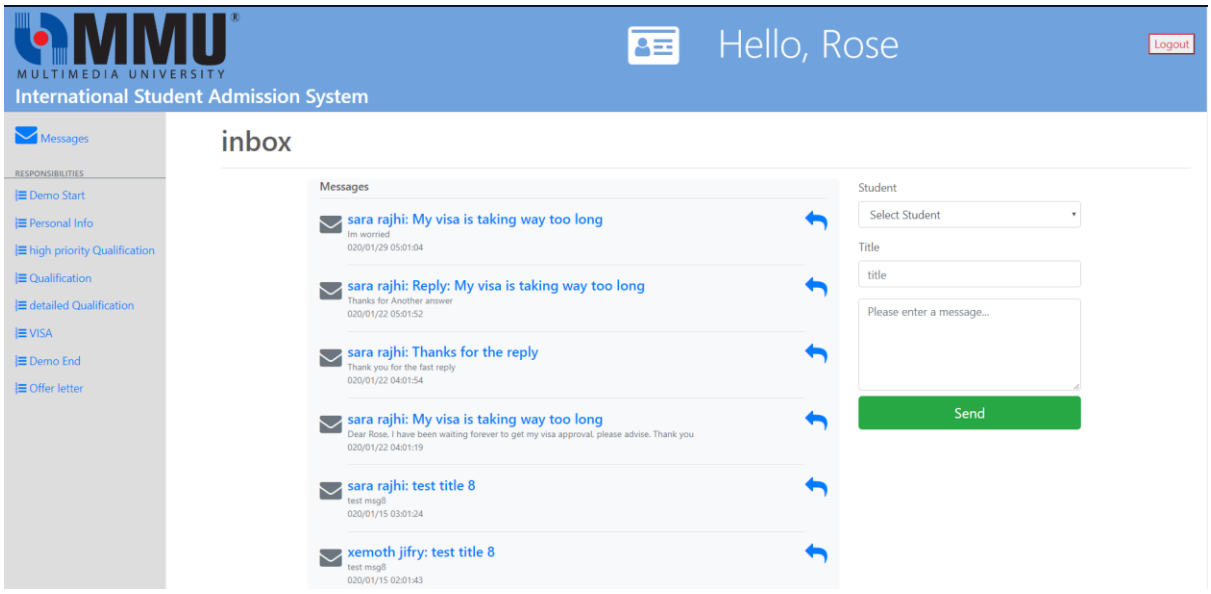


Figure 5.4-9 Messages screen

5.4.1.10. Student list

In order for the staff to view all student data they view any node that are under their responsibility

5.4.1.11. Student profile/process Student

This is the core. This screen is what makes traversing nodes and changing states possible. From viewing all available student's data to viewing the history of the student's application. The staff can make a decision whether to this student advances or not to the next node

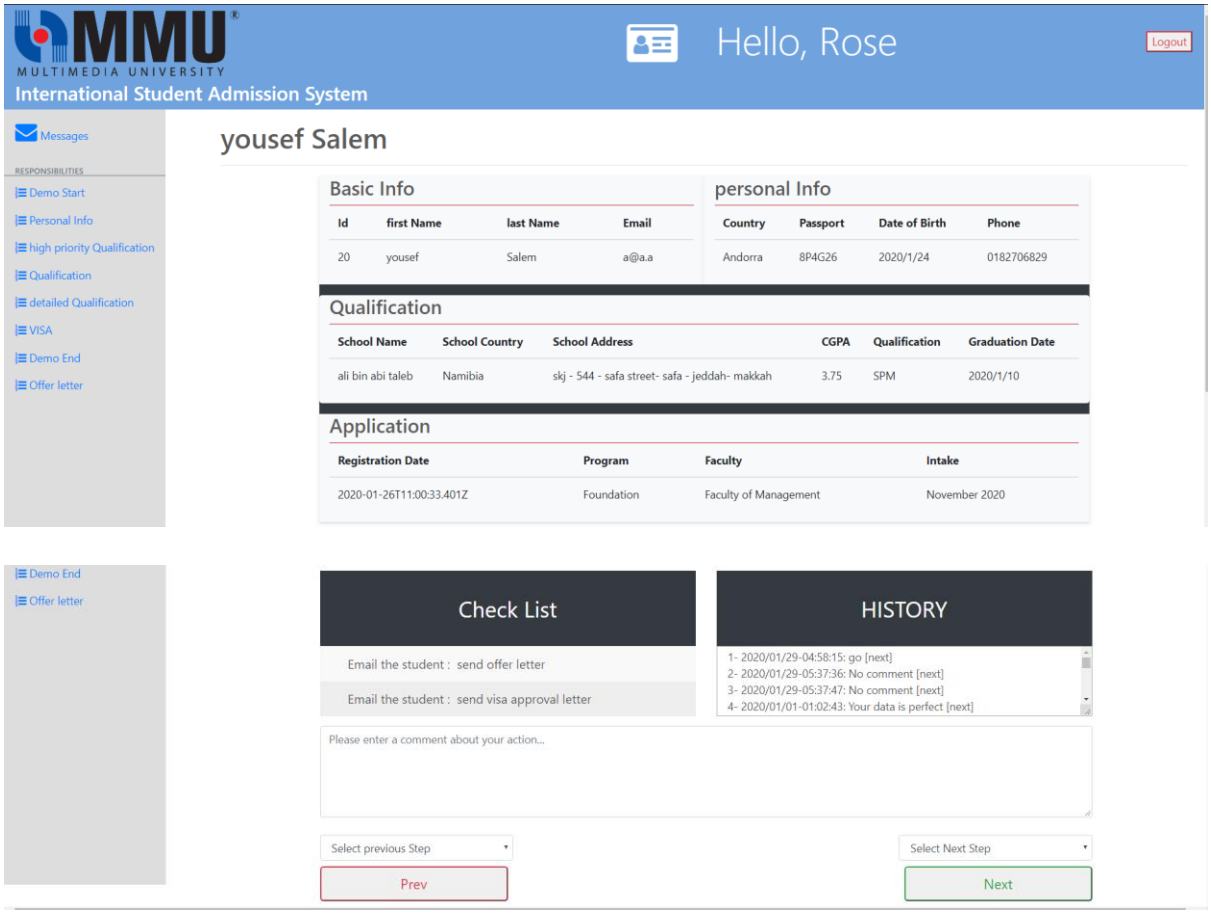


Figure 5.4-10 student profile and processing

5.4.2. Android

5.4.2.1. Login

Login screens are always an integral part of any system. It’s the usual login screen with an identifier – an email in this case and password. The student can access the system through this Login screen.

Figure 5.4-11 Student Login screen

5.4.2.2. Registration

5.4.2.3. Home

Home screens are always the centre of any system. Navigation, and accessibility of the whole system is structured here.

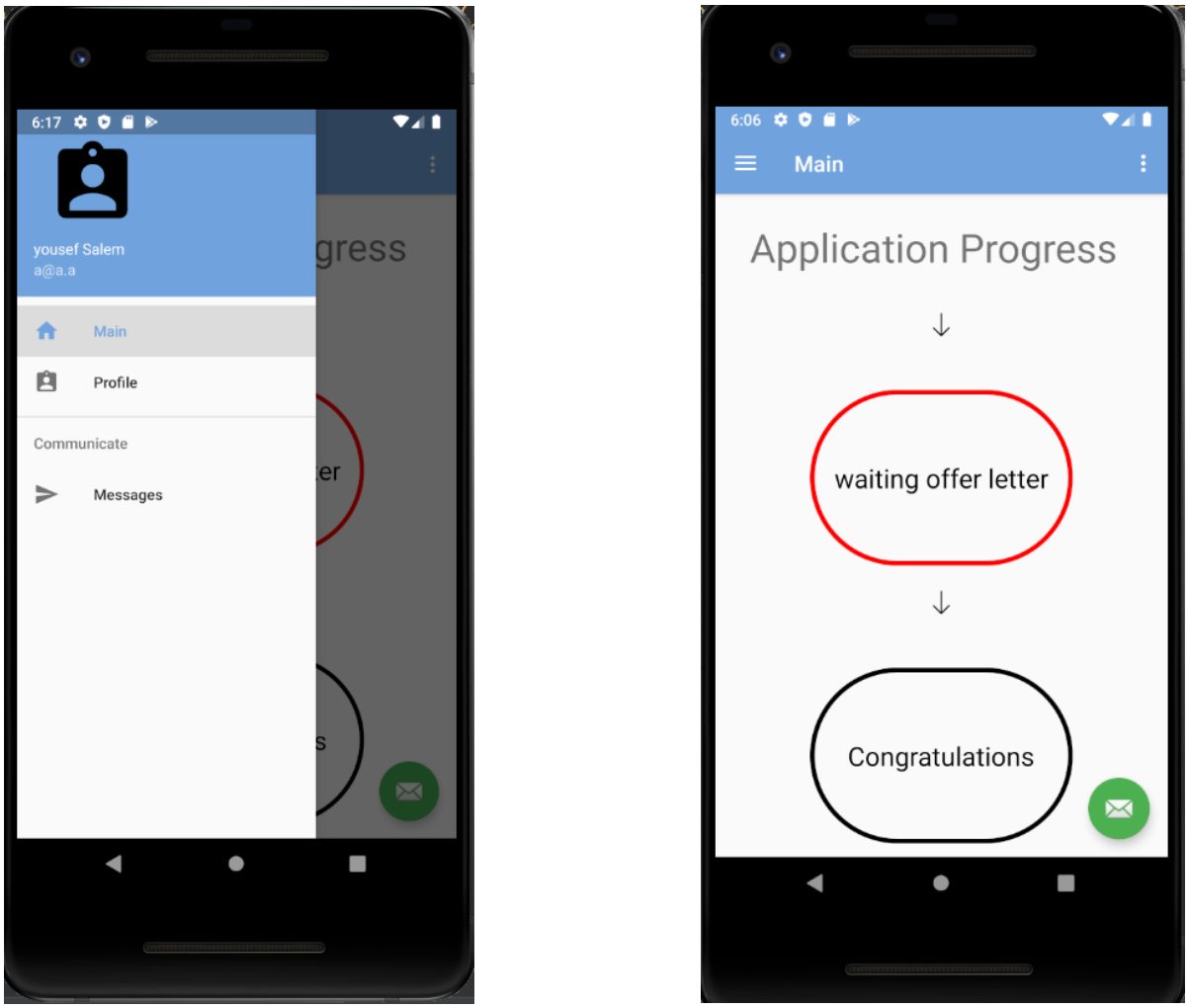


Figure 5.4-12 Student home screen

5.4.2.4. Messages

This is a basic inbox/send screen to facilitate the simple messaging system

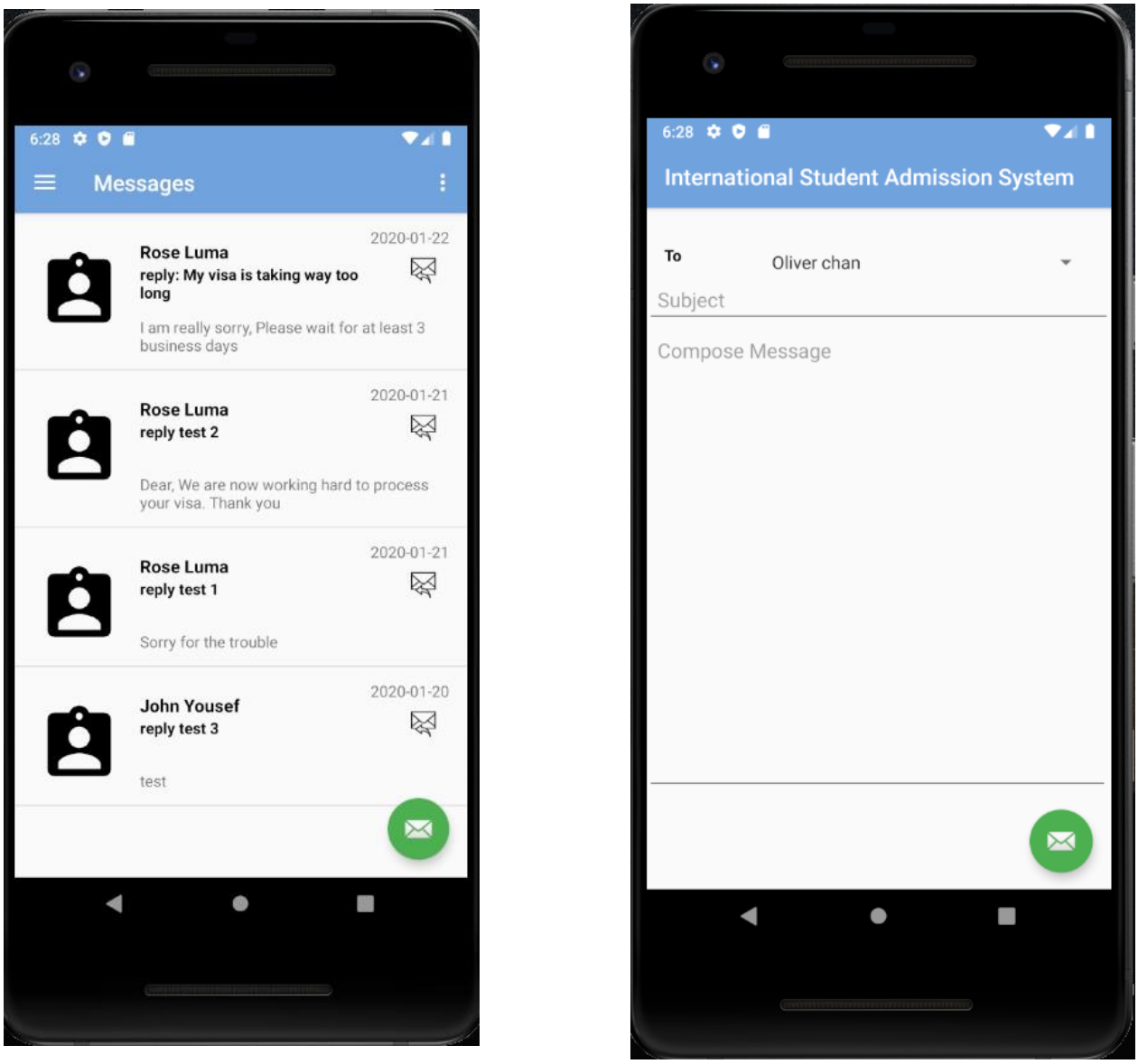


Figure 5.4-13 Student messages screen

5.4.2.5. Student profile

This screen is for the student to view their basic profile data with the ability to change their own password if they wish

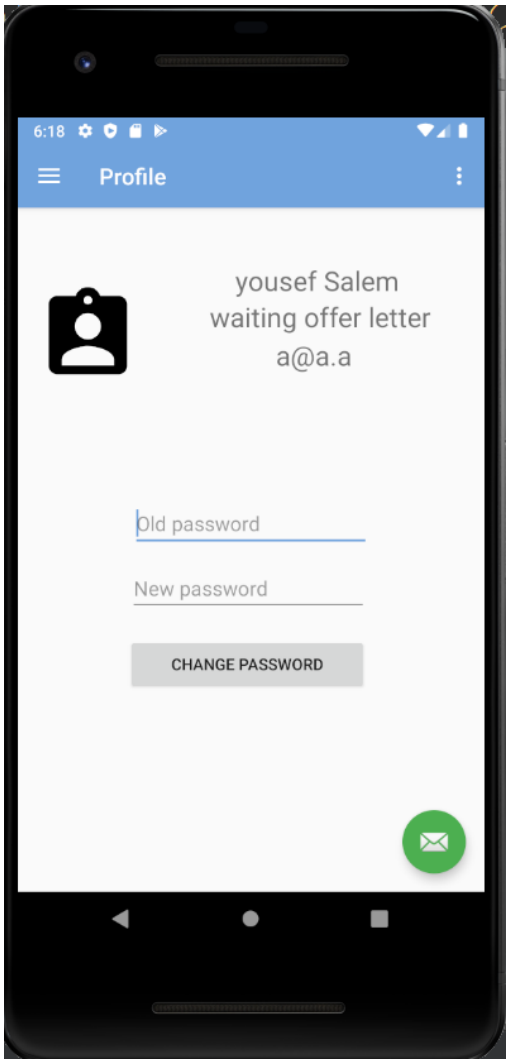


Figure 5.4-14 student profile screen

5.5. Algorithms

5.5.1. Admin system statistics (NodeJS)

```
app.get('/home/admin/chart',async function(req, res) {
    //prepare empty arrays
    nodesProcessed = []
    avg = []

    //Get relevant data
    allStudents = await db.getNumOfStudents();
    students = await db.getNumOfStudentsPerDayinOneWeek()
    actions = await db.getNumOfActionsPerWeek()
    if(actions.length == 1){
        actions[0].count = 0
        actions.push({action: null, count:0})
    }
    staff = await db.getNumOfStaff()
    arrivedNodes = await db.getArrivedNodes()
    processedNodes = await db.getProcessedNodes()

    //Set up average process time
    avgProcessTime = await setUpAverageProcessTime(arrivedNodes,processedNodes)

    //prepare data array and send it to Angular as JSON
    data = [students,actions,staff,avgProcessTime.avg,avgProcessTime.states,allStudents]
    res.json(data);
});
```

5.5.2. Admin system statistics (Angular)

```
app2.controller('chart', function($scope,$http){
    $http.get('/home/admin/chart').success(function(data) {
      //Receive all data
      $scope.students = data[0]
      $scope.actions = data[1]
      $scope.staff = data[2]
      $scope.avgProcessTime = data[3]
      $scope.nodesProcessed = data[4]
      $scope.allStudents = data[5]

      //Link with HTML Elements
      var avgProcessTime = document.getElementById("avgProcessTime").getContext("2d"
);
      var newStudents = document.getElementById("newStudents").getContext("2d");
      var processes = document.getElementById("processes").getContext("2d");
      var staff = document.getElementById("staff").getContext("2d");

      //Make the charts
      new Chart(avgProcessTime,
        getConfig('polarArea',$scope.nodesProcessed,$scope.avgProcessTime))

      new Chart(newStudents, getConfig('pie', ['6 days ago', '5 days ago', '4 days ago', '3 days
        ago','2 days ago', 'Yesterday','Today'],
        [$scope.students[0].count, $scope.students[1].count, $scope.students[2].count,
         cope.students[3].count, $scope.students[4].count, $scope.students[5].count,
        $scope.students[6].count ]))

      new Chart(processes, getConfig('pie', ['next','prev'], [$scope.actions[0].count,
         $scope.actions[1].count]))

        new Chart(staff, getConfig('pie', ['staff','students'], [$scope.staff.count,
          $scope.allStudents.count]) )
   })
})
```

5.5.3. Create workflow

```
exports.createWorkflow = async function(req,res){
  try {
        //database function returns true if transaction was successful
    if(await db.createWorkflow(req.body.workflowname,
        req.body.workflowdesc,req.user.id)){

        //Creation of default start
        start = req.body.workflowname + " " + "Start"
        workflow = await db.getLastWorkflow()
        await db.createNode(start,"Registration",workflow.id,req.user.id)
        //Receive ID of start node
        var node = await db.getLastNode()
        //Creation of default end node
        end = req.body.workflowname + " " + "End"
        await db.createNode(end,"Congratulations",workflow.id,req.user.id)
        //Creation of staff(s) responsible for default start
        if(req.body.staff && req.body.staff.constructor === Array){
          for(var i = 0; i < req.body.staff.length; i++){
            await db.createNodeStaff(node.id,req.body.staff[i])
          }
        }else if(req.body.staff){
          await db.createNodeStaff(node.id,req.body.staff)
        }
        //Render back workflow screen with a proper message on success
        return res.render('admin/workflow',
         {req: req, Message: "[workflow successfully created]"} )
    }else{
        //Render back workflow screen with a proper message on fail
        return res.render('admin/register-staff',
         {Message: "[workflow creation failed, check data]"} )
    }

  } catch (error) {
     //Render back workflow screen with a console error message in extreme cases
    res.redirect('/admin/workflow')
    console.log(error);
  } }
```

5.5.4. Create Node part 1

```
exports.createNode = async function(req,res){
  try {
    //Create node and save the return value for more complicated codes
    nodeCreationFlag = await db.createNode(req.body.nodename,req.body.nodestate
      ,req.body.workflow,req.user.id)

    //Get node that was created to use the auto generated serial ID
    var node = await db.getLastNode()

    //Create node(s) directions for next/prev for the node
    if(req.body.prev && req.body.prev.constructor === Array){
      for(var i = 0; i < req.body.prev.length; i++){
        await db.createNodeWithDirection(node.id,"prev",req.body.prev[i])
      }
    }else if(req.body.prev){
      await db.createNodeWithDirection(node.id,"prev",req.body.prev)
    }

    if(req.body.next && req.body.next.constructor === Array){
      for(var i = 0; i < req.body.next.length; i++){
        await db.createNodeWithDirection(node.id,"next",req.body.next[i])
      }
    }else if(req.body.next){
      await db.createNodeWithDirection(node.id,"next",req.body.next)
    }

    //Create responsible staff for the node
    if(req.body.staff && req.body.staff.constructor === Array){
      for(var i = 0; i < req.body.staff.length; i++){
        await db.createNodeStaff(node.id,req.body.staff[i])
      }
    }else if(req.body.staff){
      await db.createNodeStaff(node.id,req.body.staff)
    }
```

5.5.5. Create Node part 2

```
        //conver the tasks/Description JSON into an arrays for easier processing
        var tasks = JSON.parse(req.body.tasks);
        var description = JSON.parse(req.body.taskDescription);
        for(var i = 0; i < tasks.length; i++){
            await db.createTask(tasks[i],description[i],node.id)
        }


        //Render back create node screen with proper message
        if(nodeCreationFlag){
            res.render('admin/create-node',{req, Message: "[Node successfully Created]"})
        }else{
            res.render('admin/create-node',{req,Message: "[Node Creation failed, Check data]"})

        }

    } catch (error) {
        //Render back workflow screen with a console error message in extreme cases
        res.redirect('/admin/create-node')
        console.log(error);
    }
}
```

5.5.6. Process student (prev)

```
exports.processPrev = async function(req, res){
  try {
    //get selected state
    prev = JSON.parse(req.body.prev)
    result_node = prev.id
    newState = prev.state

    //update student state
    await db.updateStudentState(newState,req.body.student_id)

    //record the process wiith comment with an automatic current date
    //insert optional comment if comment doesn't return null
    if(req.body.comment){
      await db.createComment(req.body.comment,req.body.staff_id,
        req.body.student_id,req.body.node_id,'prev',result_node)
    }else{
      await db.createComment("No comment",req.body.staff_id,
      req.body.student_id,req.body.node_id,'prev',result_node)
    }

    //go back to process node
    res.redirect('/selectStudent/' + req.body.node_id)

  } catch (error) {
    //Render back to process screen with a console error message in extreme cases
    res.redirect('/selectStudent/' + req.body.node_id)
    console.log(error);
  }
}
```

5.5.7. Student Send Message (1)

```
class send_msg : AppCompatActivity(){
  //initialize and set up default values
   val database = db() lateinit var option: Spinner
  lateinit var title: EditText lateinit var content: EditText var resultID:Int = -1


   override fun onCreate(savedInstanceState: Bundle?) {
      super.onCreate(savedInstanceState)
      setContentView(R.layout.send_msg)
      //link with xml
      option = findViewById(R.id.staffNames) as Spinner
      title = findViewById(R.id.send_title) as EditText
      content = findViewById(R.id.send_content) as EditText
      //get all staff from database and get student data
      val staff = database.getStaff()
      val oldIntent  = intent.extras
      var student_id = oldIntent?.getInt("userID")
      var replyStaffId = oldIntent?.getInt("staff")
      var replyTitle = oldIntent?.getString("title")
      //Set up reply Links
      if(replyStaffId != null && replyTitle!= null){
         var replayStaff = db.newStaff(-1,"f","f")
         title.setText(replyTitle!!)
         resultID = replyStaffId
         for (s in staff){
            if(s.id == replyStaffId){
               replayStaff = s
            }
         }
         staff.remove(replayStaff)
         staff.add(0,replayStaff)
      }
```

5.5.8. Student Send Message (2)

```
//Prepare all staff as options in spinner
val staffNames = ArrayList<String>()
for (s in staff){
    staffNames.add(s.fname + " " + s.lname )
}
option.adapter = ArrayAdapter<String>
    (this,android.R.layout.simple_list_item_1,staffNames)
//set up on selection resultID
option.onItemSelectedListener = object : AdapterView.OnItemSelectedListener {
    override fun onNothingSelected(p0: AdapterView<*>?) {}
    override fun onItemSelected
    (p0: AdapterView<*>?, p1: View?, p2: Int, p3: Long) {
        resultID = staff.get(p2).id
    }
}
//Send button link with XML and set up
val fab: FloatingActionButton = findViewById(R.id.send)
fab.setOnClickListener {
    if(title.text.isNotBlank() && content.text.isNotBlank()){
        //database registration
        database.insertMsg(title.text.toString(), content.text.toString(),
                                student_id!!,resultID )
        println("Data is valid!")
        Toast.makeText(this,"Message Sent",Toast.LENGTH_LONG).show()
        finish()
    }else{
        Toast.makeText(this,"Please fill both subject and content of the message",
    Toast.LENGTH_SHORT).show()
    }
}   }}
```

6. Testing

For testing. Security testing and manual black box use case testing is used.

6.1. Test traceability Matrix

Only system-defining functions will be thoroughly tested. Since all of them are system-defining the risk is really high if any of them fail

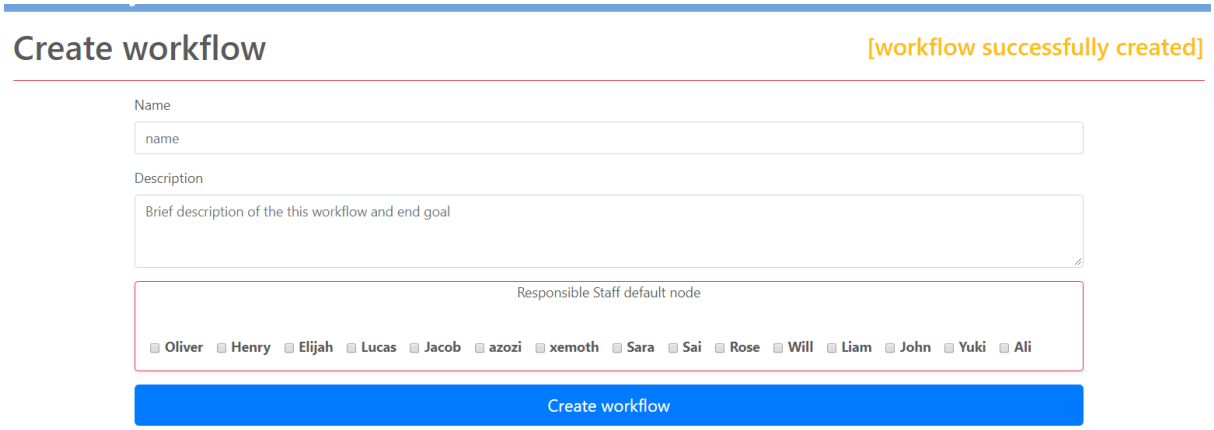
Table 6.1-1 Test traceability Matrix

Function ID	Name	Risk level	Description
F01	Create workflow	High	The ability to create a workflow
F02	Create node	High	The ability to create a node
F03	Process student	High	The ability to process a student
F04	Send message	Medium	The ability to send a message to a student

6.1.1. [F001] [Create workflow]

Table 6.1-2 test create workflow

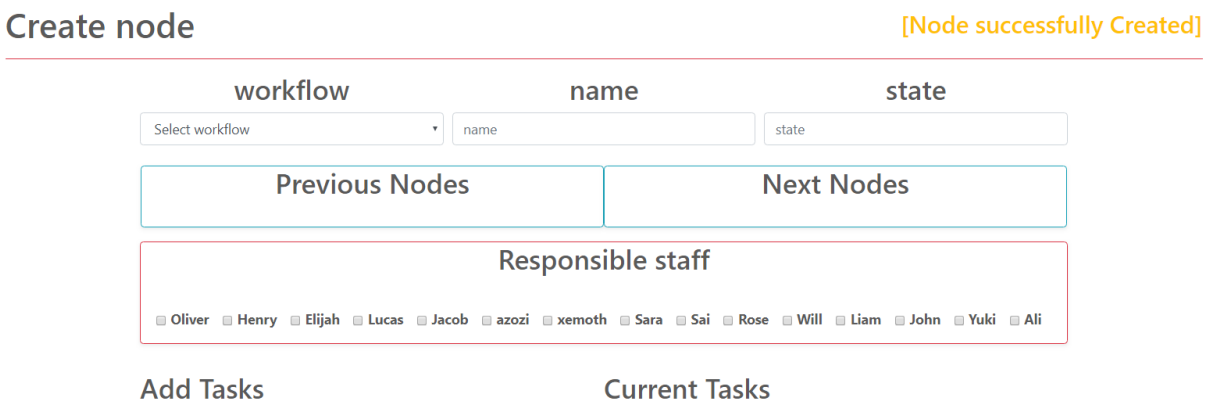
Use case ID	F001	
Use case	F001 create workflow	
Purpose	To allow admin to create the essence of the system, a workflow	
Actor	Admin	
Trigger	N/A	
Precondition	Admin is logged in	
Scenario name	Step	Action
Main flow	1	Fill workflow details including responsible staff
	2	Click create workflow
Rules	N/A	
Author	Abdulrahman	
Pass/fail	pass	
Picture		



6.1.2. [F002] [Create node]

Table 6.1-3 test create node

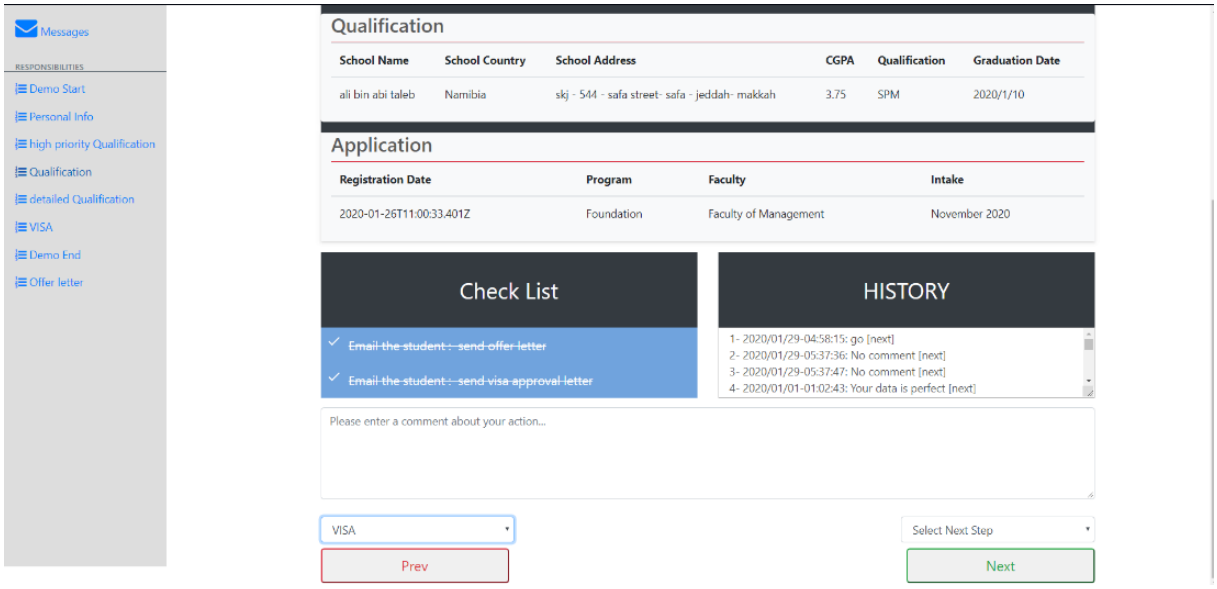
Use case ID	F002	
Use case	F002 create node	
Purpose	To allow admin to create the essence of the workflow, a node	
Actor	Admin	
Trigger	N/A	
Precondition	Admin is logged in	
Scenario name	Step	Action
Main flow	1	Fill node details including responsible staff, next/prev nodes. To do list
	2	Click create node
Rules	N/A	
Author	Abdulrahman	
Pass/fail	pass	
Picture		



6.1.3. [F003] [Process student]

Table 6.1-4 test process student

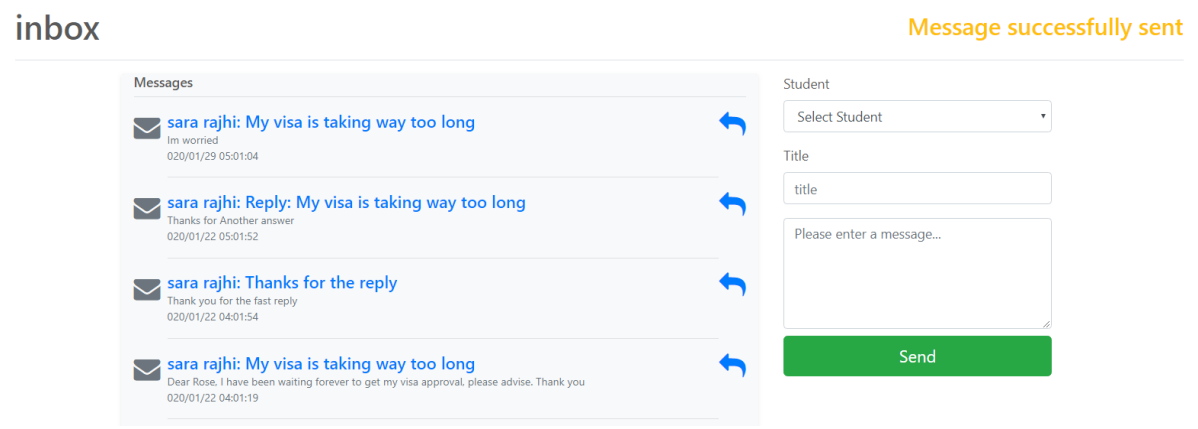
Use case ID	F003	
Use case	F003 process student	
Purpose	To allow staff to process students	
Actor	Staff	
Trigger	N/A	
Precondition	staff is logged-in student's state is the same as a node that's under the responsibility of a staff	
Scenario name	Step	Action
Main flow	1	Check all to-dos
	2	Click next/prev
Rules	N/A	
Post condition	Student state is updated accordingly. Systems stats are affected	
Author	Abdulrahman	
Pass/fail	pass	
Picture		

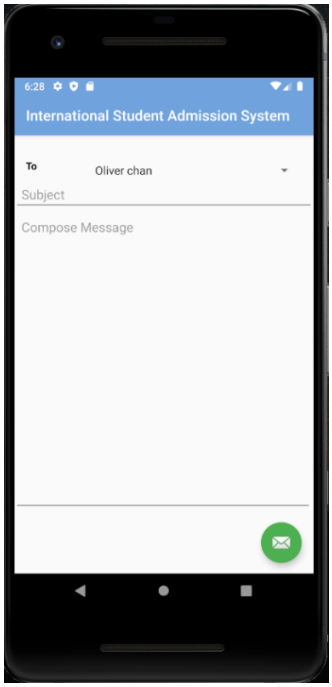


[F004] [Send Message]

Table 6.1-5 test send message

Use case ID	F004	
Use case	F004 Send Message	
Purpose	To allow exchange of messages between staff and students	
Actor	Staff	
Trigger	N/A	
Precondition	User is logged in	
Scenario name	Step	Action
Main flow	1	Fill message details
	2	Click details
Rules	N/A	
Author	Abdulrahman	
Pass/fail	Pass	
Picture web/android		





6.2. Security Testing

Security testing is testing that's done to check if the system adheres to certain security standards.

Table 6.2-1 Security Test

Test Scenario	Data delivery and database access is safe. Encryption is enforced to prevent any type of temperament during the transaction
Expected Result	Order details must be encrypted during delivery to database
Actual Result	Website is encrypted. No temperament allowed since no HIGH alerts were detected
Status	Pass

Using OWASP ZAP (Owasp, n.d.), an automated attack with ajax spider the result that the website is secured but there are alerts that of classes "medium", "low", and "information". Since none of these are HIGH the Website is not in real danger.

6.2.1. Security report

Here is the full table of the report and some examples of the medium warnings. The full report will be attached with the code as "Owasp.html"

Table 6.2-2 Security Risk Level

Risk Level	Number of Alerts
High	0
Medium	2
Low	6
Informational	2

6.2.1.1. Alert Detail

Table 6.2-3 Medium risk example 1

Medium (Medium)	X-Frame-Options Header Not Set
Description	X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks.
URL	http://localhost:3000/register
Method	GET
Parameter	X-Frame-Options
URL	http://localhost:3000
Method	GET
Parameter	X-Frame-Options
URL	http://localhost:3000/forgotpass
Method	GET
Parameter	X-Frame-Options
URL	http://localhost:3000/login
Method	GET
Parameter	X-Frame-Options
Instances	4
Solution	Most modern Web browsers support the X-Frame-Options HTTP header. Ensure it's set on all web pages returned by your site (if you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. ALLOW-FROM allows specific websites to frame the web page in supported web browsers).
Reference	http://blogs.msdn.com/b/ieinternals/archive/2010/03/30/combating-clickjacking-with-x-frame-options.aspx
CWE Id	16
WASC Id	15
Source ID	3

Table 6.2-4 Medium risk example 2

Medium (Medium)	CSP Scanner: Wildcard Directive
Description	The following directives either allow wildcard sources (or ancestors), are not defined, or are overly broadly defined: frame-ancestor
URL	http://localhost:3000/favicon.ico
Method	GET
Parameter	Content-Security-Policy
Evidence	default-src 'none'
Instances	1
Solution	Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.
Reference	http://www.w3.org/TR/CSP2/ http://www.w3.org/TR/CSP/ http://caniuse.com/#search=content+security+policy http://content-security-policy.com/ https://github.com/shapesecurity/salvation
CWE Id	16
WASC Id	15
Source ID	3

Table 6.2-5 Low risk example 1

Low (Medium)	Web Browser XSS Protection Not Enabled
Description	Web Browser XSS Protection is not enabled, or is disabled by the configuration of the 'X-XSS-Protection' HTTP response header on the web server
URL	http://localhost:3000
Method	GET
Parameter	X-XSS-Protection
URL	http://localhost:3000/favicon.ico
Method	GET
Parameter	X-XSS-Protection
URL	http://localhost:3000/login
Method	GET
Parameter	X-XSS-Protection
URL	http://localhost:3000/register
Method	GET
Parameter	X-XSS-Protection
Instances	5
Solution	Ensure that the web browser's XSS filter is enabled, by setting the X-XSS-Protection HTTP response header to '1'.
Reference	https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet https://www.veracode.com/blog/2014/03/guidelines-for-setting-security-headers/
CWE Id	933
WASC Id	14
Source ID	3

7. Conclusion

Admission for international students is a big hurdle that's only a part of the journey. International students have to leave their whole lives behind them in order to attempt to pave their future. To help the easement of going through the admission process a mobile application was developed.

Universities are losing students because the implemented systems lack transparency and proper performance indicators. The lack of data about the process and its stages hiders the quality decisions needed to have real improvements of the process. A website for the management was developed with dynamic workflow capabilities to solve the performance problems and to have proper adaptability.

The Internal messaging subsystem solves our transparency problem. Because even with providing application status updates, you need a modern and convenient communication approach.

8. Appendix A: Commercialization Proposal

COMMERCIALIZATION PROPOSAL

INTERNATIONAL STUDENT ADMISSION SYSTEM DESIGNED AS A DYNAMIC WORKFLOW FOR WEB AND MOBILE

ABDULRAHMAN MOHAMMED AYASH YOUSEF
FACULTY OF COMPUTING AND INFORMATICS
MULTIMEDIA UNIVERSITY

8.1. Executive Summary

The system is composed of two parts with separate platforms. A website for staff and admin and a mobile for the student. Uniqueness of the system is that the workflow is completely dynamic and depends completely on the admin's plan. Aided by the available statistics, Admins can modify the workflow to improve even the smallest details that results in delays anywhere in the process. It is truly a win-win solution for all parties involved. Since the solution is a general-purpose dynamic workflow, the system itself can be adopted by any type of "workflow-based" processes. The System was just implemented as an international student admission system as a choice. Changing the data captured by the mobile user can lead to huge changes in the context. This System is a base framework for this type of workflow- based processes. Potential competitors are any international student admission system for other universities. Target customers are universities who would want a better international admission system, another step can be industries than want their processes to have some sort of automation in the paperwork. The system is proposed to be sold as a complete package both mobile and web for 20000RM with an option to include IOS application per customer request with another 6000RM. The financial analysis conducted suggests that the total development and marketing would cost around 40000RM. Assuming that the system will interest at least two universities even without the IOS option, the generated revenue will break even with the total cost.

8.2. Objectives

- To ensure that universities do not lose a substantial amount of potential international students
- To provide a modifiable dynamic workflow for the management to help facilitate the changes to make the process faster, easier, and overall more pleasant.
- To provide statistics on demand for the management
- To provide a simple messaging system for clarifications between students and staff
- To provide up to date application status to give students a peace of mind

8.3. Market Analysis

8.3.1. Target customers

The system was developed for universities to service their international students and improve the process of admission. The potential customers are all universities all over the globe. At the current stage only Android application is available but with a customer's request an IOS version could be developed. In a larger scheme, companies, hospitals, any context that has some sort of workflow and a process that needs automation with analysis can be a potential customer.

8.3.2. Unique value proposition

The existing problem is a two-part problem. From the admission side, there are no metrics or analysis involved in the process so the admission often times have no idea why the process is slowed down or where to improve without conducting a thorough check up that halts the actual process and waste precious time. This system provides periodic analysis to help with monitoring and making sense of the bigger picture. The second problem is the lack of mobile applications for students as a whole or if an application exists it lacks the clarity of the process or even the openness of a direct internal messaging system.

8.4. Business Model

The system is proposed to be sold as a complete package both mobile and web for 20000RM with an option to include IOS application per customer request with another 6000RM.

8.4.1. Marketing

- Direct marketing: This will be conducted by meeting the international offices of universities and introduce and promote the system.
- Mass marketing This will be conducted by addressing the idea of a dynamic workflow with meaningful statistics to any party who needs such features to automate their processes

8.5. Milestones and key metrics

Activity	Time
Development (web/android)	completed
Development (IOS)	per request
Marketing	To be conducted
Selling the system	Assuming two customers, the cost will be covered

8.6. SWOT Analysis

8.6.1. Strength

The system is a complete package that is a win-win for all parties involved

8.6.2. Weakness

Only available in android, IOS development will take time per request

8.6.3. Opportunity

Every university has an international system. An improvement is always desirable.

8.6.4. Threat

The technologies used are all fast-evolving technologies, monitoring and providing support and improvements with updates and maintenance can be troublesome.

9. Appendix B

9.1. Meeting Logs (FYP1)

MULTIMEDIA UNIVERSITY

Faculty of Computing & Informatics
Final Year Project (FYP1) Meeting Log

MEETING DATE: 17/07/2019	**MEETING NO.:** 1
PROJECT ID: 1407	**SESSION :** 2019/2020
PROJECT TITLE : INTERNATIONAL STUDENT ADMISSION SYSTEM DESIGNED AS A DYNAMIC WORKFLOW FOR WEB AND MOBILE	
STUDENT ID : 1171300042	**STUDENT NAME:** Abdulrahman Yousef
SUPERVISOR : Mr. Sharaf Al-Horani	**CO- SUPERVISOR :** -

1. WORK DONE

Project Plan including gantt chart and what to write in the project. Some questions were asked such as: "what programming language would be better to use ?" and "what should we limit the project to ?".

2. WORK TO BE DONE

Merging the gantt chart with the milestones and the introduction section in a document.

3. PROBLEMS ENCOUNTERED

The goal was unclear, but became clearer once the plan was set.

4. COMMENTS

The student had some problems with the plan, and the goal wasn't clear for him, but he quickly caught up with the flow when a solution was suggested to him.

MR. SHARAF HORANI
Specialist 2
Faculty of Computing and Informatics
Multimedia University
Persiaran Multimedia, 63100 Cyberjaya
Selangor Darul Ehsan, Malaysia

.....
Supervisor's Signature

.....
Student's Signature

.....
Co-Supervisor's Signature

MULTIMEDIA UNIVERSITY

Faculty of Computing & Informatics
Final Year Project (FYP1) Meeting Log

MEETING DATE: 21/07/2019	**MEETING NO.:** 2
PROJECT ID: 1407	**SESSION :** 2019/2020
PROJECT TITLE : INTERNATIONAL STUDENT ADMISSION SYSTEM DESIGNED AS A DYNAMIC WORKFLOW FOR WEB AND MOBILE	
STUDENT ID : 1171300042	**STUDENT NAME:** Abdulrahman Yousef
SUPERVISOR : Mr. Sharaf Al-Horani	**CO- SUPERVISOR :** -

1. WORK DONE

The introduction and Gantt chart has been merged together, and part of chapter two has been written; Some requirements have been gathered and documented as general requirements to outline the system infrastructure.

Similar systems have been identified and a background study have been conducted on them.

2. WORK TO BE DONE

Specify generalized requirements, identify potential users, and split the specified requirements for each one of them, and to draw the first UML diagram the Use Case Diagrams for each user that has been identified.

Write about the findings about the similar systems found, and finalize background study writing.

3. PROBLEMS ENCOUNTERED

Not many programs were found that are similar to the project's idea, and the ones found, a description for them was also difficult to find and were tested to write a proper description about them.

Interviews took too much time and effort to gather the initial requirements.

4. COMMENTS

The project has been progressing at a good pace so far.

MR. SHAHAF HORANI
Specialist 2
Faculty of Computing and Informatics
Multimedia University
Persiaran Multimedia, 63100 Cyberjaya
Selangor Darul Ehsan, Malaysia

.....
Supervisor's Signature

.....
Student's Signature

.....
Co-Supervisor's Signature

NOTES:
1. Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
2. Weekly meeting logs (FYP1-Week 3 to Week 11) have to be submitted along with project report.

MULTIMEDIA UNIVERSITY

Faculty of Computing & Informatics
Final Year Project (FYP1) Meeting Log

MEETING DATE: 31/07/2019	**MEETING NO.:** 3
PROJECT ID: 1407	**SESSION :** 2019/2020
PROJECT TITLE : INTERNATIONAL STUDENT ADMISSION SYSTEM DESIGNED AS A DYNAMIC WORKFLOW FOR WEB AND MOBILE	
STUDENT ID : 1171300042	**STUDENT NAME:** Abdulrahman Yousef
SUPERVISOR : Mr. Sharaf Al-Horani	**CO- SUPERVISOR :** -

1. WORK DONE

the general requirements were revised by the supervisor to be rechecked, feed-back was given and changes are going to be made accordingly.

2. WORK TO BE DONE

specify requirements, draw sequence diagrams for each use case and write descriptions for them, along with system context diagram and database entity relationship diagram.

3. PROBLEMS ENCOUNTERED

The amount of requirements was overwhelming and took too much time to be written. I wasn't able to do all the tasks for this week, but that's fine. When any changes were made everything had to be updated and it's a bit difficult to keep up with everything.

4. COMMENTS

The background study was still not done, the student said that he will finish it by the end of the week. He had also sent me the initial requirements for revision, there were some mistakes, and he had inquiries, but everything was clarified and discussed at the end.

MR. SHARAF HORANI
Specialist 2
Faculty of Computing and Informatics
Multimedia University
Persiaran Multimedia, 63100 Cyberjaya
Selangor Darul Ehsan, Malaysia

.....
Supervisor's Signature

.....
Student's Signature

.....
Co-Supervisor's Signature

NOTES:
1. Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
2. Weekly meeting logs (FYP1-Week 3 to Week 11) have to be submitted along with project report.

MULTIMEDIA UNIVERSITY

Faculty of Computing & Informatics
Final Year Project (FYP1) Meeting Log

MEETING DATE: 7/08/2019	**MEETING NO.:** 4
PROJECT ID: 1407	**SESSION :** 2019/2020
PROJECT TITLE : INTERNATIONAL STUDENT ADMISSION SYSTEM DESIGNED AS A DYNAMIC WORKFLOW FOR WEB AND MOBILE	
STUDENT ID : 1171300042	**STUDENT NAME:** Abdulrahman Yousef
SUPERVISOR : Mr. Sharaf Al-Horani	**CO- SUPERVISOR :** -

1. WORK DONE

The requirements specifications are discussed with the supervisor, and a draft of the use case diagrams have been sent to the supervisor.

2. WORK TO BE DONE

finalize requirements, and finish sequence diagrams and use case diagram, then check with the supervisor for final confirmation on proceeding with the requirements.

The design part will be looked at slightly to begin early with it.

3. PROBLEMS ENCOUNTERED

No problems were encountered this week.

4. COMMENTS

The requirements' first draft was sent and checked.

MR. SHARAF HORANI
Specialist 2
Faculty of Computing and Informatics
Multimedia University
Persiaran Multimedia, 63100 Cyberjaya
Selangor Darul Ehsan, Malaysia

.....
Supervisor's Signature

.....
Student's Signature

.....
Co-Supervisor's Signature

NOTES:

1. Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
2. Weekly meeting logs (FYP1-Week 3 to Week 11) have to be submitted along with project report.
3. Log sheets are compulsory assessment criteria for FYP. Student who fails to meet the requirements of log sheets will not be allowed to submit FYP report.

MULTIMEDIA UNIVERSITY

Faculty of Computing & Informatics
Final Year Project (FYP1) Meeting Log

MEETING DATE: 14/08/2019	**MEETING NO.:** 5
PROJECT ID: 1407	**SESSION :** 2019/2020
PROJECT TITLE : INTERNATIONAL STUDENT ADMISSION SYSTEM DESIGNED AS A DYNAMIC WORKFLOW FOR WEB AND MOBILE	
STUDENT ID : 1171300042	**STUDENT NAME:** Abdulrahman Yousef
SUPERVISOR : Mr. Sharaf Al-Horani	**CO- SUPERVISOR :** -

1. WORK DONE

The first draft of function and quality requirements were written, use case diagrams and sequence diagrams were written for each use case specified from the requirements.

2. WORK TO BE DONE

finalize requirements, and finish sequence diagrams and use case diagram, then check with the supervisor for final confirmation on proceeding with the requirements.

3. PROBLEMS ENCOUNTERED

No problems were encountered this week.

4. COMMENTS

The drafts for requirements specifications were checked and discussed.

MR. SHARAF HORANI
Specialist 2
Faculty of Computing and Informatics
Multimedia University
Persiaran Multimedia, 63100 Cyberjaya
Selangor Darul Ehsan, Malaysia

.....
Supervisor's Signature

.....
Student's Signature

.....
Co-Supervisor's Signature

NOTES:

1. Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
2. Weekly meeting logs (FYP1-Week 3 to Week 11) have to be submitted along with project report.
3. Log sheets are compulsory assessment criteria for FYP. Student who fails to meet the requirements of log sheets will not be allowed to submit FYP report.

MULTIMEDIA UNIVERSITY

Faculty of Computing & Informatics
Final Year Project (FYP1) Meeting Log

MEETING DATE: 21/08/2019	**MEETING NO.:** 6
PROJECT ID: 1407	**SESSION :** 2019/2020
PROJECT TITLE : INTERNATIONAL STUDENT ADMISSION SYSTEM DESIGNED AS A DYNAMIC WORKFLOW FOR WEB AND MOBILE	
STUDENT ID : 1171300042	**STUDENT NAME:** Abdulrahman Yousef
SUPERVISOR : Mr. Sharaf Al-Horani	**CO- SUPERVISOR :** -

1. WORK DONE

The requirements specifications were finalized with the diagrams. The entity relationship diagram was also drawn.

2. WORK TO BE DONE

Start designing the system

3. PROBLEMS ENCOUNTERED

No problems were encountered this week.

4. COMMENTS

The student mentioned he will be busy with some outside interference for the next week and we may not be able to meet except when necessary.

MR. SHARAF HORANI
Specialist 2
Faculty of Computing and Informatics
Multimedia University
Persiaran Multimedia, 63100 Cyberjaya
Selangor Darul Ehsan, Malaysia

.....
Supervisor's Signature

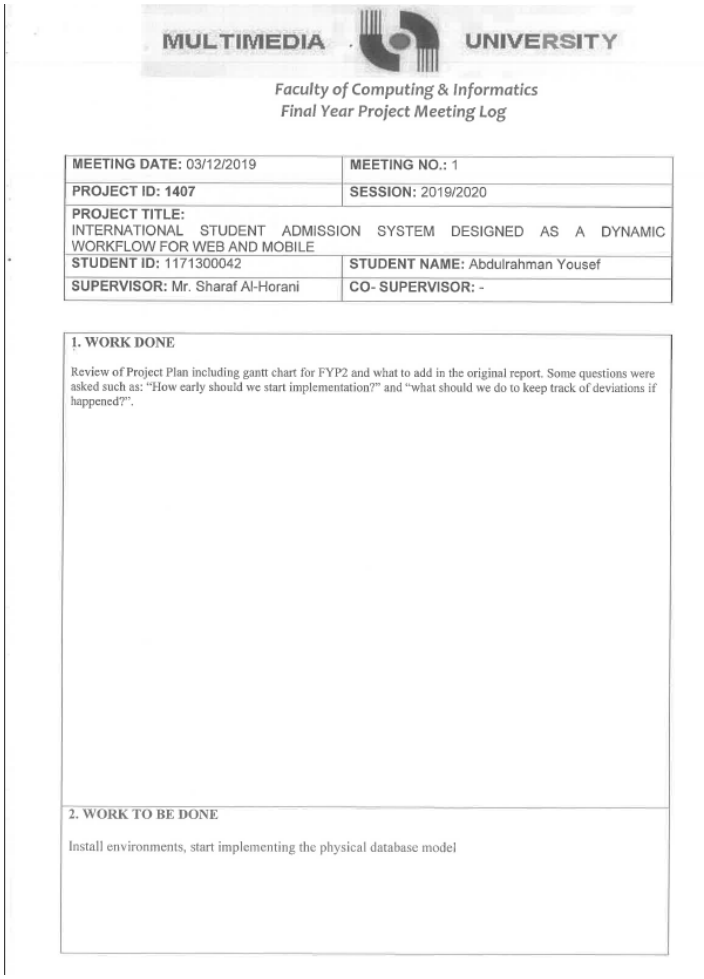
.....
Student's Signature

.....
Co-Supervisor's Signature

NOTES:

1. Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
2. Weekly meeting logs (FYP1-Week 3 to Week 11) have to be submitted along with project report.
3. Log sheets are compulsory assessment criteria for FYP. Student who fails to meet the requirements of log sheets will not be allowed to submit FYP report.

9.2. Meeting logs (FYP2)



MULTIMEDIA UNIVERSITY

Faculty of Computing & Informatics
Final Year Project Meeting Log

MEETING DATE: 03/12/2019	**MEETING NO.:** 1
PROJECT ID: 1407	**SESSION:** 2019/2020
PROJECT TITLE: INTERNATIONAL STUDENT ADMISSION SYSTEM DESIGNED AS A DYNAMIC WORKFLOW FOR WEB AND MOBILE	
STUDENT ID: 1171300042	**STUDENT NAME:** Abdulrahman Yousef
SUPERVISOR: Mr. Sharaf Al-Horani	**CO- SUPERVISOR:** -

1. WORK DONE

Review of Project Plan including gantt chart for FYP2 and what to add in the original report. Some questions were asked such as: "How early should we start implementation?" and "what should we do to keep track of deviations if happened?".

2. WORK TO BE DONE

Install environments, start implementing the physical database model

3. PROBLEMS ENCOUNTERED None
4. COMMENTS The student was fairly excited and eager to start realizing the design from FYP1

MR. SHARAF HORANI
Specialist 2
Faculty of Computing and Informatics
Multimedia University
Selangor Darul Ehsan, Malaysia

Supervisor's Signature

Student's Signature

Co-Supervisor's Signature

MULTIMEDIA UNIVERSITY

Faculty of Computing & Informatics
Final Year Project Meeting Log

MEETING DATE: 10/12/2019	**MEETING NO.:** 2
PROJECT ID: 1407	**SESSION:** 2019/2020
PROJECT TITLE: INTERNATIONAL STUDENT ADMISSION SYSTEM DESIGNED AS A DYNAMIC WORKFLOW FOR WEB AND MOBILE	
STUDENT ID: 1171300042	**STUDENT NAME:** Abdulrahman Yousef
SUPERVISOR: Mr. Sharaf Al-Horani	**CO- SUPERVISOR:** -

1. WORK DONE

Environments have been installed. Database have been created with all connectivity pk/fk mapping

2. WORK TO BE DONE

Connect Database with the back end. Start preparing login/Registration pages

3. PROBLEMS ENCOUNTERED
-
4. COMMENTS
The project has been progressing at a good pace so far.

AKMF HORANI
Specialist
Computing and Informatics
Multimedia University
63100 Cyberjaya
Selangor Darul Ehsan

Supervisor's Signature

Student's Signature

Co-Supervisor's Signature

MULTIMEDIA UNIVERSITY

Faculty of Computing & Informatics
Final Year Project Meeting Log

MEETING DATE: 17/12/2019	**MEETING NO.:** 3
PROJECT ID: 1407	**SESSION:** 2019/2020
PROJECT TITLE: INTERNATIONAL STUDENT ADMISSION SYSTEM DESIGNED AS A DYNAMIC WORKFLOW FOR WEB AND MOBILE	
STUDENT ID: 1171300042	**STUDENT NAME:** Abdulrahman Yousef
SUPERVISOR : Mr. Sharaf Al-Horani	**CO- SUPERVISOR:** -

1. WORK DONE

Database connection is established with backend. Registration and login design and implementation is halfway done

2. WORK TO BE DONE

Finish up registration and login. Make sure the password is encrypted. Start planning main functionality

3. PROBLEMS ENCOUNTERED

Database connection wasted a lot of time. The port, localhost, and some very postgres specific details were not quite clear with node postgres package. Also, exporting the database to be used throughout the node project was tricky

4. COMMENTS

Security issues were raised. All in all, the pace is quite good.

MR. SHARAF HORANI
Specialist 2
Faculty of Computing and Informatics
Multimedia University
Persiaran Multimedia, 63100 Cyberjaya
Selangor Darul Ehsan, Malaysia

Supervisor's Signature

Student's Signature

Co-Supervisor's Signature

MULTIMEDIA UNIVERSITY

Faculty of Computing & Informatics
Final Year Project Meeting Log

MEETING DATE: 24/12/2020	**MEETING NO.:** 4
PROJECT ID: 1407	**SESSION:** 2019/2020
PROJECT TITLE: INTERNATIONAL STUDENT ADMISSION SYSTEM DESIGNED AS A DYNAMIC WORKFLOW FOR WEB AND MOBILE	
STUDENT ID: 1171300042	**STUDENT NAME:** Abdulrahman Yousef
SUPERVISOR: Mr. Sharaf Al-Horani	**CO- SUPERVISOR:** -

1. WORK DONE

Login and registration are completely done. Security is enforced in the website with node BCRYPT package. Passport and routing are set up in node. The main functionality skeleton is built.

2. WORK TO BE DONE

Finish up main functionality.

3. PROBLEMS ENCOUNTERED

Sending and receiving messages from angular is a hustle. Dealing with this issue is hindering the process

4. COMMENTS

Angular is powerful but the connectivity from node to angular and client to angular to node is tricky and needs some research.

MR. SHARAF HORANI
Specialist 2
Faculty of Computing and Informatics
Multimedia University
Persiaran Multimedia, 63100 Cyberjaya
Selangor Darul Ehsan, Malaysia

Supervisor's Signature

Student's Signature

Co-Supervisor's Signature

MULTIMEDIA UNIVERSITY

Faculty of Computing & Informatics
Final Year Project Meeting Log

MEETING DATE: 14/01/2020	**MEETING NO.:** 5
PROJECT ID: 1407	**SESSION:** 2019/2020
PROJECT TITLE: INTERNATIONAL STUDENT ADMISSION SYSTEM DESIGNED AS A DYNAMIC WORKFLOW FOR WEB AND MOBILE	
STUDENT ID: 1171300042	**STUDENT NAME:** Abdulrahman Yousef
SUPERVISOR: Mr. Sharaf Al-Horani	**CO- SUPERVISOR:** -

1. WORK DONE

Main functionality is done. Angular research is done. More or less comfortable with PEAN full stack. Mobile Application development commenced. Login and registration with messaging system half way done

2. WORK TO BE DONE

Clean up the way the website looks. Finish up the messaging system from both ends

3. PROBLEMS ENCOUNTERED

No problems were encountered this week.

4. COMMENTS

Android studio is terrible. The student's time is seriously wasted while waiting for the simplest tasks because of laptop limitation.

Specialist 2
Faculty of Computing and Informatics
Multimedia University
Persiaran Multimedia, 63100 Cyberjaya
Selangor Darul Ehsan, Malaysia

Supervisor's Signature

Student's Signature

Co-Supervisor's Signature

MULTIMEDIA UNIVERSITY

Faculty of Computing & Informatics
Final Year Project Meeting Log

MEETING DATE: 05/02/2020	**MEETING NO.:** 6
PROJECT ID: 1407	**SESSION:** 2019/2020
PROJECT TITLE: INTERNATIONAL STUDENT ADMISSION SYSTEM DESIGNED AS A DYNAMIC WORKFLOW FOR WEB AND MOBILE	
STUDENT ID: 1171300042	**STUDENT NAME:** Abdulrahman Yousef
SUPERVISOR: Mr. Sharaf Al-Horani	**CO- SUPERVISOR:** -

1. WORK DONE

Mobile application is done. Testing and final report are both almost done. Finishing touches. Bootstrap was implemented in the website to make it look better.

2. WORK TO BE DONE

Finish up the report, get ready for turnit in and final testing

3. PROBLEMS ENCOUNTERED
No problems were encountered this week.
4. COMMENTS
The student has more or less finished their work.

MR. SHARAF HORANI
Specialist 2
Faculty of Computing and Informatics
Multimedia University
Persiaran Multimedia, 63100 Cyberjaya
Selangor Darul Ehsan, Malaysia

Supervisor's Signature

Student's Signature

Co-Supervisor's Signature

9.3. TurnItIn report

FYP2_comb bound copy report_Yousef

ORIGINALITY REPORT

%19	%4	%1	%18
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Multimedia University Student Paper	%14
2	Submitted to University of Hertfordshire Student Paper	%2
3	Submitted to University of Westminster Student Paper	%1
4	Submitted to Universidad Internacional de la Rioja Student Paper	%1
5	Submitted to Grand Canyon University Student Paper	%1
6	hdl.handle.net Internet Source	%1

10. References

Activiti. (n.d.). Retrieved from https://www.activiti.org/

Android studio. (n.d.). Retrieved from https://developer.android.com/about

Android studio wiki. (n.d.). Retrieved from https://en.wikipedia.org/wiki/Android_Studio

Angular JS. (n.d.). Retrieved from https://angular.io/

APU international students application. (n.d.). Retrieved from http://www.apu.edu.my/study-apu/international-students/international-students-application-procedures

BPMN. (n.d.). Retrieved from http://www.bpmn.org/

Curtain university sarawak eStudent. (n.d.). Retrieved from https://eapplication.curtin.edu.my/eStudentPROD/S1/eApplications/eAppLogin.aspx?f=%24S1.EAP.LOGIN.WEB&_ga=2.210803798.21713832.1567411515-1189240122.1565060260

draw.io. (n.d.). Retrieved from https://www.draw.io/

E-expectations trend. (2012). Retrieved from https://files.eric.ed.gov/fulltext/ED536678.pdf

E-expectations trend. (2018). Retrieved from http://learn.ruffalonl.com/rs/395-EOG-977/images/2018_RNL_E_Expectations_Report_no%20CTA.pdf

Express JS. (n.d.). Retrieved from https://en.wikipedia.org/wiki/Express.js

Introduction to the Diagrams of UML 2.X. (n.d.). Retrieved from http://www.agilemodeling.com/essays/umlDiagrams.htm

Kotlin. (n.d.). Retrieved from https://kotlinlang.org/

Kotlin wiki. (n.d.). Retrieved from https://en.wikipedia.org/wiki/Kotlin_(programming_language)

microsoft workflow. (n.d.). Retrieved from https://flow.microsoft.com/en-us/

Mustache. (n.d.). Retrieved from https://en.wikipedia.org/wiki/Mustache_(template_system)

Node JS. (n.d.). Retrieved from https://en.wikipedia.org/wiki/Node.js

NPM. (n.d.). Retrieved from https://en.wikipedia.org/wiki/Npm_(software)

Owasp. (n.d.). Retrieved from https://owasp.org/

Postegres. (n.d.). Retrieved from https://www.postgresql.org/

Postgres. (n.d.). Retrieved from https://www.postgresql.org/

processMaker. (n.d.). Retrieved from https://www.processmaker.com/

RNL. (n.d.). Retrieved from https://www.ruffalonl.com/about-ruffalo-noel-levitz/

RNL. (n.d.). *paper source*. Retrieved from https://files.eric.ed.gov/fulltext/ED536678.pdf

visual paradim. (n.d.). Retrieved from https://www.visual-paradigm.com/

Visual studio. (n.d.). Retrieved from https://code.visualstudio.com/

webstorm. (n.d.). Retrieved from https://www.jetbrains.com/webstorm/