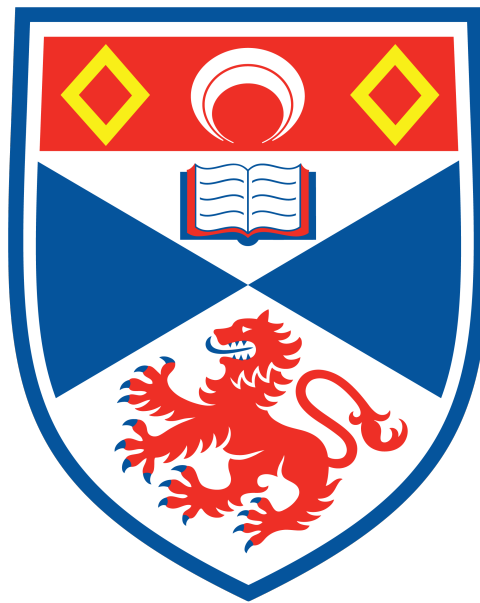


Report for CS5030 Assessment 3

Software Design, Modelling and Analysis

210016568



UNIVERSITY OF ST ANDREWS

CS5030 SOFTWARE ENGINEERING PRINCIPLES

Lecturer: Dharini Balasubramaniam

Word Count: 2011 words

Due date: November 16, 2021

CONTENTS

I Introduction	3
II System Requirements	4
i Specification of functional requirements	4
ii Specification of non-functional requirements	6
III Use Case Diagram and Specification	7
IV Ethical Considerations	8
V System Architecture	9
i System context diagram	9
ii System component diagram	10
iii System class diagram	11
iv System architecture analysis	12
VI System Behaviour	13
i Submit Feedback sequence diagram	13
ii Response Feedback sequence diagram	14
VII UML diagrams Merits and Limitations	15
i UML diagrams merits	15
ii UML diagrams limitations	15

I. INTRODUCTION

This requirement document is the output of the analysis for the an application to support feedback, response and actions between students and schools, I called it "Feedback Management System". We conducted a series of meeting with the customer to discuss their needs, and the The content of the conversation is organized as following 2 paragraphs:

Each school will appoint a member of administrative staff, who does not teach, supervise or advise students, to manage the system. All students and staff of the school are registered as users of the system and therefore, the application must interact with systems used by the University's Human Resources unit (for staff details) and Registry (for student details). In addition, the application should have information regarding role holders among staff and students, such as lecturers of modules, level coordinators, student representatives for different levels and technicians. The data contained in the application should be updated on a regular basis and reset each academic year.

Feedback can be provided by students or staff. Feedback is not anonymous at the point of submission, but it will be anonymous, unless requested otherwise, before it is passed on by the administrator to relevant stakeholders for response and action. Student representatives can submit feedback on behalf of other students. Feedback could relate to a specific module, a level of study (for example, Junior Honors or PGT) or a wider concern (for example, lab provision or school environment). A response from the role holder who receives feedback is expected within the specified time frame. Feedback from staff for students is passed on to the relevant level representatives to be disseminated to students. Actions can be associated with each feedback, to be completed within a set time and an update on actions taken is provided to the person, who provided the feedback. The application should enable anonymous summary reports to be generated for school management as well as 'you said, we did' reports for students. The system should be configurable, for example, to enable reminder notifications of upcoming response deadlines, set default response time frames and specify the period covered by reports.

(both two above paragraph are token from the pdf teacher provided)

Because of the width limitation of pdf, maybe the diagrams is not very clear, so I will also upload a pdf which just contains the original diagrams in the order in which the images appear in the report.

II. SYSTEM REQUIREMENTS

This part will illustrate this system's functional and non-functional requirement, to make sense, I have made some reasonable assumptions and definitions, which can be seen following:

- A response is based on the feedback received.
- A response includes a series of actions.
- This application should run on Windows, Linux, MacOS, Android, IOS devices.
- The application should use Microsoft Authenticator to log in.

i. Specification of functional requirements

ID	Functional requirement description	Priority	source	Dependencies
1	The system should create an account for every unique user in different roles	High	Specification	The role of users depends on 3, 4
2	The system should allow all the users to log in using their username and password in different interfaces	High	The need for all users	The username and password depend on 1
3	The system should get and set the staff details from HR system regularly	Medium	Specification	
4	The system should get and set the student details from Registry system regularly	Medium	Specification	
5	The system should reset data every academic year	Medium	Specification	Get and set data depends on 3, 4
6	The system should allow users to submit feedback	High	Specification	
7	The system should allow administrator to allocate feedback to stakeholders in anonymous or public way by setting a limited time	High	Specification	
8	The system should allow student representatives to submit feedback on behalf of other students	Medium	Specification	The feedback they allocated depends on 6
9	The system should allow feedback receiver respond to it by a series of actions	High	Specification	The feedback they respond depends on 6, 8
10	The system should send each action of the response to the submitter of feedback	High	Specification	
11	The system should distinguish the feedback from staff for student, and send it to relative representative	Medium	Specification	The role of representative depends on 1,4. The feedback depends on 6

12	The system should generate anonymous summary reports for school management	Low	Specification	The content in report based on the whole process of feedback-response-action and the period covered by report, so it depends on 6-11, 14
13	The system should generate 'you said, we did' reports for students	Low	Specification	The content in report based on the whole process of feedback-response-action and the period covered by report, so it depends on 6-11, 14
14	The administrative staff can configure the default response time frames and the period covered by reports	Low	Specification	This function depends on 9
15	The system should notify the feedback receiver in default response time frames before deadline	Low	Specification	The notification based on the time of submitting feedback, so it depends on 6
16	The system should be deployed into other universities	Medium	Specification	
17	The system should verify by Microsoft Authenticator when users are logging in	High	Assumption	This function depends on 2

ii. Specification of non-functional requirements

- **Speed:**
Reacts to user actions within 30 seconds and to scheduled system tasks (alerting the user some time before deadline) within one hour.
- **Size:**
The size of this application's installation package should not be bigger than 150MB.
- **Ease of use:**
The application should be able to be used quickly by high-educated or currently high-educated users in a short period of time.
- **Reliability:**
The system should be available at 90% of the time.
- **Robustness:**
The system should be restarted in 30 minutes, recover after reset data failure. In case of a problem, the system must be fixed into 10 hours.
- **Portability:**
The system should run on Windows, Linux, MacOS, Android 8 or newer, and IOS 11 or newer.
- **Localization:**
The system should support different languages for students from different countries.

III. USE CASE DIAGRAM AND SPECIFICATION

The use case diagram is Figure 1. To make the diagram more concise and at a high level of abstraction, I only described the main use cases.

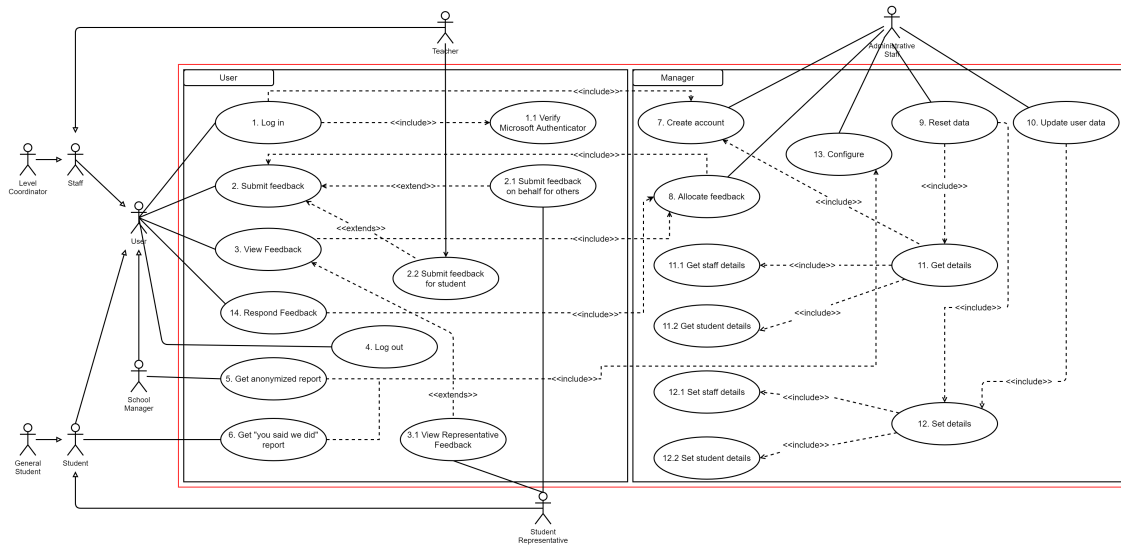


Figure 1: Use case diagram

Here I chose the Allocate Feedback use case to illustrate.

Use case name	Allocate Feedback
Use case ID	8
Goal in Context	The administrative staff need to allocate the feedback to stakeholders
Predconditions	The user has submitted a feedback
Primary Actors	The administrative staff
Main flow	1.The use case start when the user has logged in and submitted a feedback 2. the administrative staff send the feedback to stackholders
Postconditions	The feedback is expected to be responded by stackholders
Alternative flows	If the feedback is from teacher for student, the administrative send the feedback to student representative

IV. ETHICAL CONSIDERATIONS

The biggest ethical consideration in the system we will design should be the anonymity of the feedback. According to the specification, the system should keep the feedbackers anonymous until they receive a response. Therefore, when designing the system and implementing the software in code, the parameters should be passed very carefully to ensure complete anonymity.

In addition, a more obvious ethical consideration is the extent to which this system can get details in its interaction with Human Resources System and Registry System. To simplify the diagram, after fully considering the scenario, I fixed the details that the system designed for this report can get from HR and Registry systems to a few minimal information, which can be seen in the Details class and its subclasses in Figure 4 (system class diagram).

Further, in order to enforce the integrity of data, the use of the CASCADE keyword should be carefully considered in the design of the database, and appropriate use of foreign key constraints is required.

Finally, the integrity and recoverability of the data needs to be ensured as the data will be reset every academic year.

These are the ethical considerations that I can think of that might be present in the system I am designing.

V. SYSTEM ARCHITECTURE

Based on functional and non-functional requirements, I divided this system as five subsystems. different combination of them will be developed as two kinds of applications (can run in different OS including Windows, Linux, MacOS, Android and IOS), i.e., Manger App and User App in Figure 3.

In order to describe the system architecture at different levels of abstraction, I will use system context diagram in Figure 2, system component diagram in Figure 3, and system class diagram in Figure 4, separately.

i. System context diagram

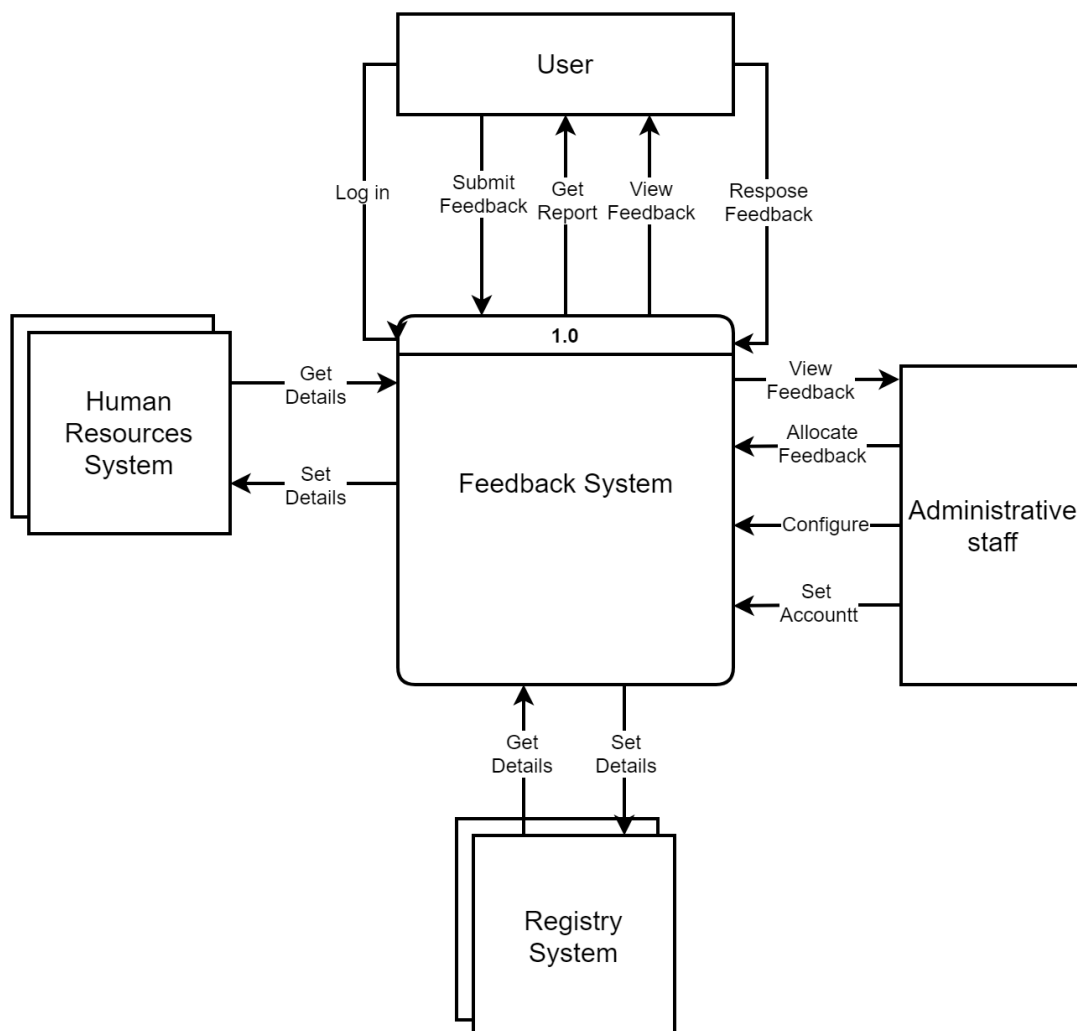


Figure 2: *System content diagram*

ii. System component diagram

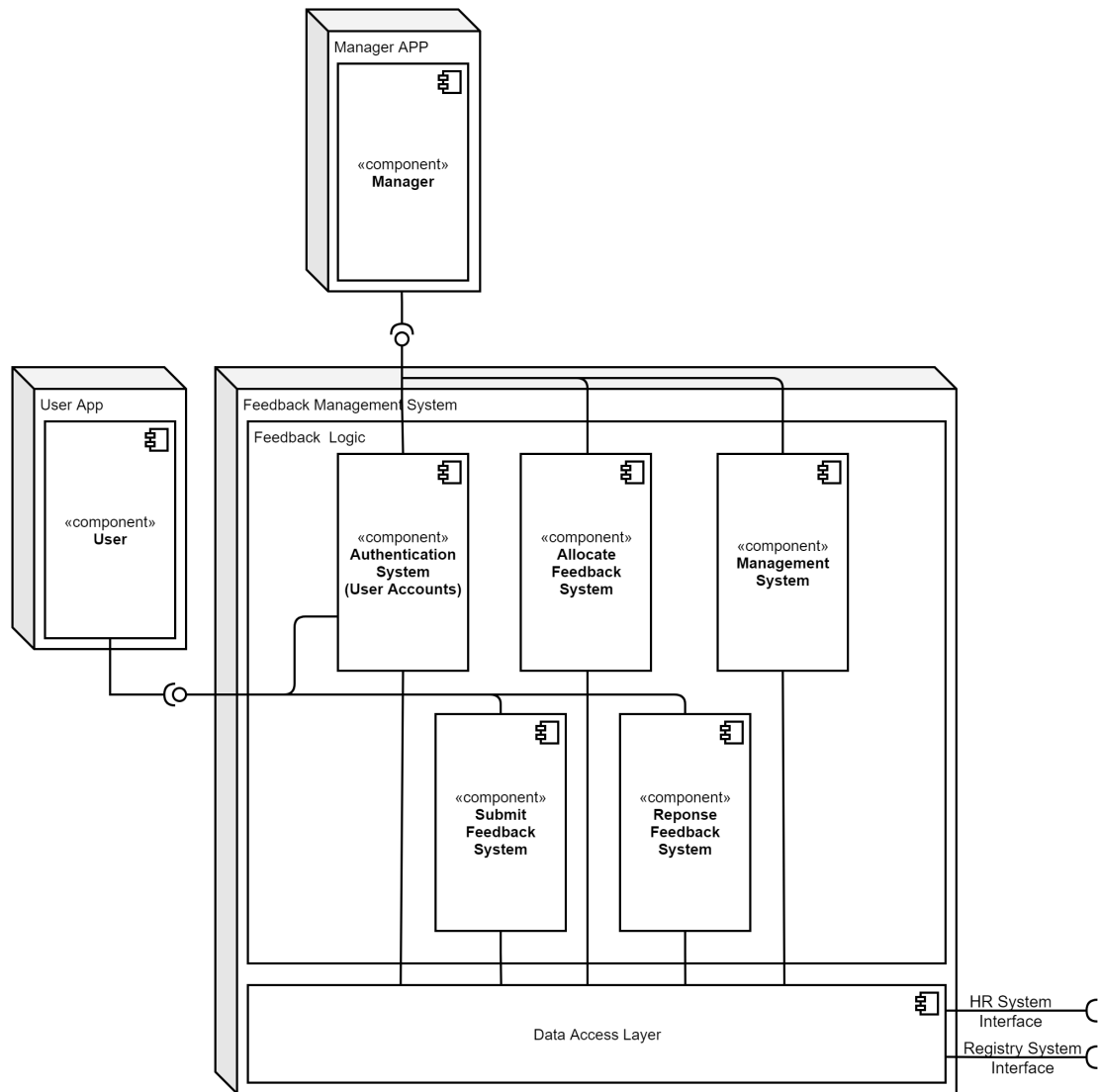


Figure 3: System Component diagram

iii. System class diagram

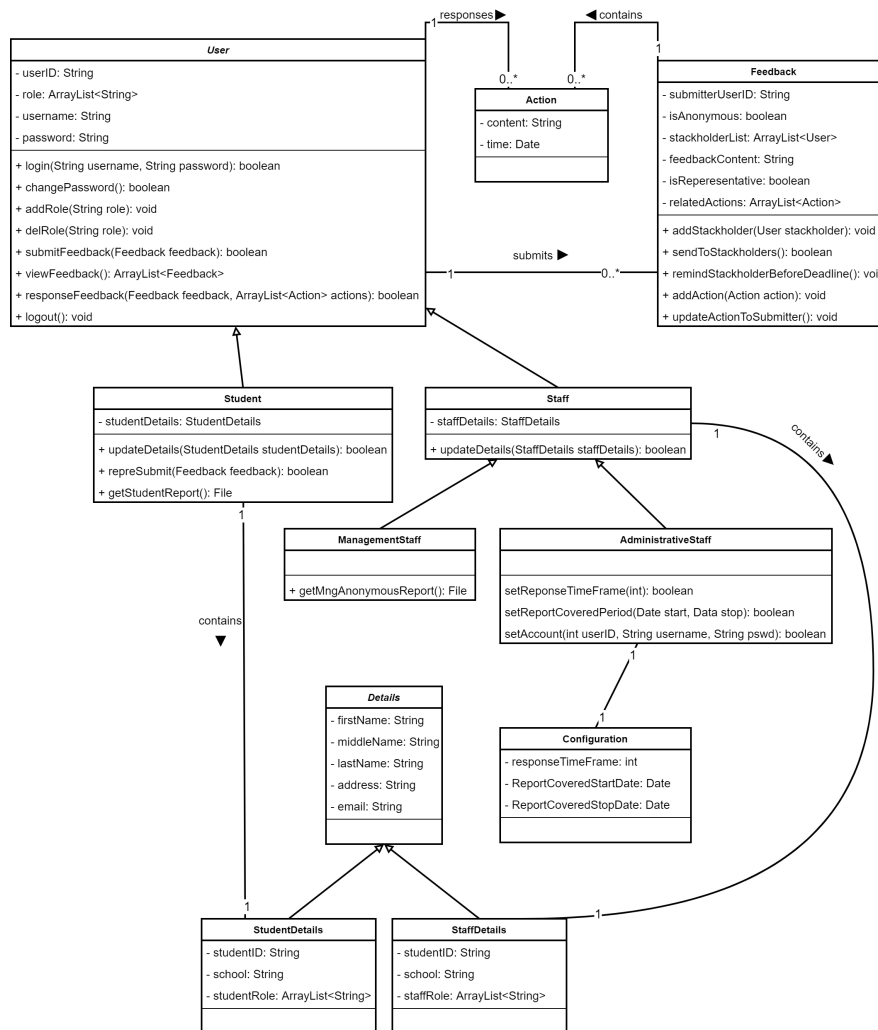


Figure 4: System Class diagram

iv. System architecture analysis

The system I designed divided it into several independent part, and use different combinations to suite different needs. I compared with other style of architectures, here are potential advantages and disadvantages.

Architecture advantages:

1. The system this assignment designed has three main data sources, the APP side, the database in the system, the external API from Human Resources system and Registry system. The data are independent of each other, which can safeguard the privacy and data security issues of users to a great extent.
2. The system is highly configurable and migratory, according to the non-functional requirements, it can easily adapt to different languages. So it can be easily deployed to other schools by simply providing external interfaces for staff and student details.

Architecture disadvantages:

1. The robustness of the system is poor. Although the interdependence of data solves the problem of data security, it means that the system relies on the APIs of other systems to obtain data, and it will be a big concern whether the system can operate normally if there is a problem in other systems.
2. Since the system requires the app to be developed on different operating systems, it requires a large development team to get this done. This will increase the cost of project development.

VI. SYSTEM BEHAVIOUR

In this part, I will illustrate users' behaviours in the system I designed. However, it is difficult to use only one diagram to illustrate, I will illustrate "Submit Feedback", "Response Action to Feedback" to present it.

i. Submit Feedback sequence diagram

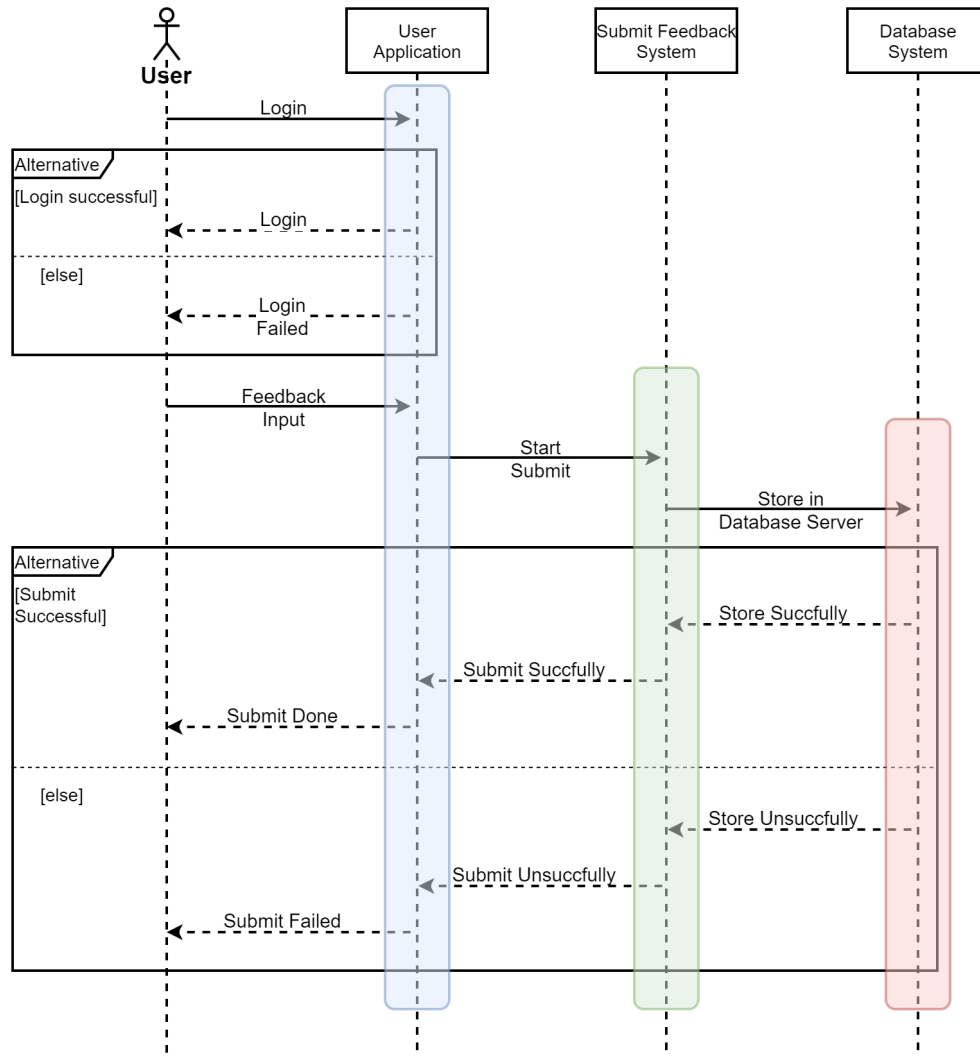


Figure 5: Submit Feedback Sequence Diagram

ii. Response Feedback sequence diagram

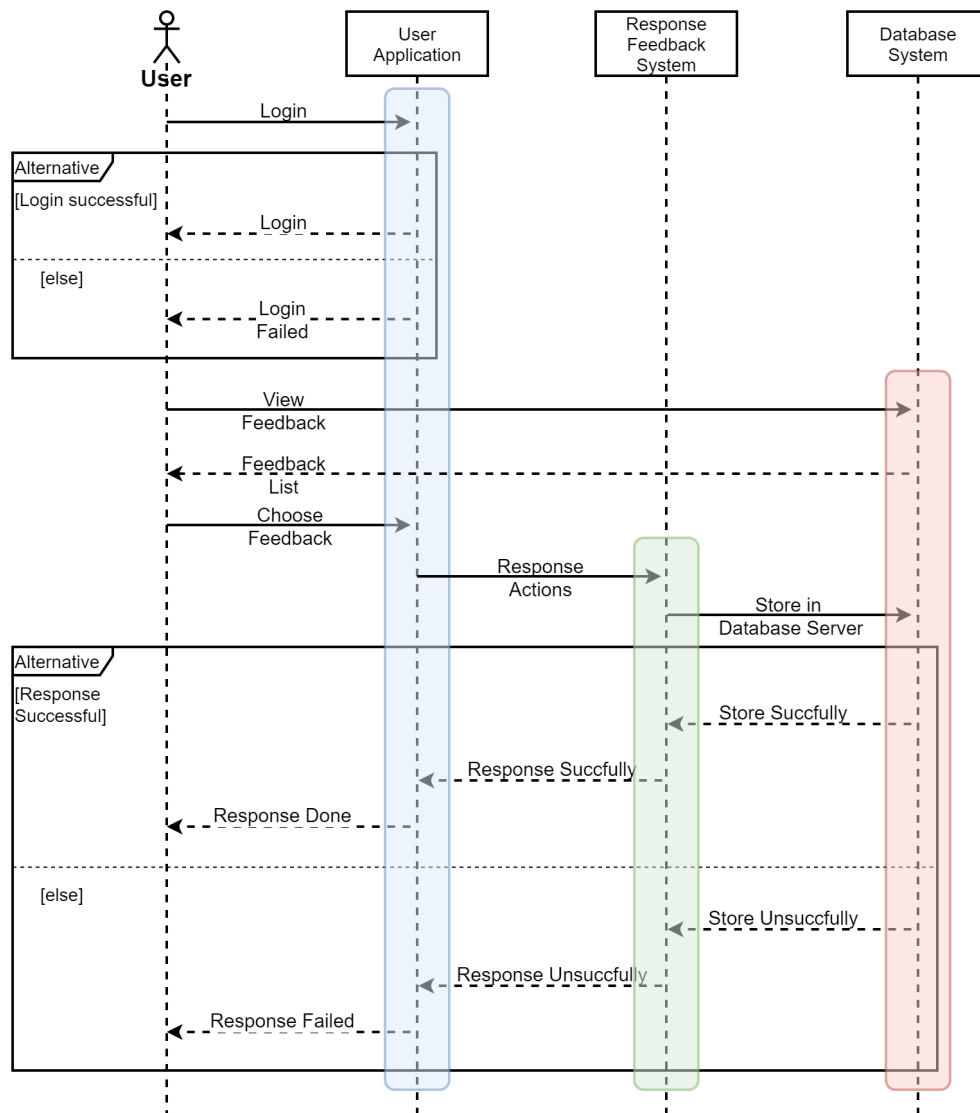


Figure 6: Submit Response Sequence Diagram

VII. UML DIAGRAMS MERITS AND LIMITATIONS

In this part, I will briefly discuss the merits and limitations of UML diagrams in the context of this experience of designing a system and the ideas in the lectures.

i. UML diagrams merits

1. UML diagrams are very concise and efficient for planning before the project starts, and it is easy to build out the interactions between different identity users and the system.
2. Compared to programming languages, UML diagrams are easier to get started and allows everyone to easily understand what is being expressed in diagrams.
3. UML diagrams are quite rich in variety and can be used to describe almost every phase of the software development process.
4. There are many free and lightweight UML diagrams tools that can be easily gotten.

ii. UML diagrams limitations

1. For beginners who do not have a lot of experience in software development projects, they only can understand the UML diagrams but cannot to draw them in the very first period of time.
2. There are different tools for drawing UML diagrams, and some of them may differ in their symbolic representation from the standard required by UML, which can cause some trouble to beginners.
3. UML diagrams have many different levels of abstraction of the system. In the process of drawing the diagram, it is easy to mix up the different levels of abstraction in wrong way, which leads to a very confusing result. On the other hand, there is often no way to determine a unique answer to the question of how to distinguish between different levels of abstraction and divide system into different parts, relying more on subjective feelings or intuition to make the division.

The third limitation was the most torturous part of this assignment for me. I asked different students about their views on the design system (without any academic misconduct taking place) and they all had different levels of abstraction, which kept me up at night. Although there are a thousand Hamlets in a thousand people, the huge difference in the level of abstraction made me doubt whether my design was reasonable and effective. It was only after reviewing my coursework and looking up the official UML standards[1] that I finally started to think that my design seemed to work.

REFERENCES

- [1] OBJECT MANAGEMENT GROUP. Unified modeling language, v2.5.1. Website, 2017. <https://www.omg.org/spec/UML/2.5.1/PDF>.