



Module Code & Module Title
CS5002NA Software Engineering

Assessment Weightage & Type
60% Individual Coursework

Year and Semester
2019-20 Autumn / 2019-20 Spring

Student Name: Girija Tamang

London Met ID: 18030995

College ID: NP05CP4S190007

Assignment Due Date: 05 June 2020

Assignment Submission Date: 50 June 2020

Submitted To: Mission Babu Sapkota

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Table of Contents

1. Introduction	1
2. Gantt Chart	3
3. Use Case Diagram	4
3.1 High Level Use Case Description.....	8
3.2 Expanded Use Case Description	11
4. Communication diagram.....	13
4.1 Steps for developing communication diagram.....	13
4.2 Collaboration diagram	16
5. Sequence diagram.....	17
5.1 Steps to draw a sequence diagram are:.....	17
5.2 Sequence Diagram	19
6. Class Diagram	20
6.1 Steps for developing class diagram	21
7. Further Development Process	24
8. Prototype	25
9. Conclusion.....	36
References.....	37

Tables of Figures

Figure 1: Gantt Chart.....	3
Figure 2: Use Case Diagram.....	7
Figure 3: Collaboration diagram of patient registration.....	16
Figure 4: Sequence diagram of patient registration.....	19
Figure 5: Class Diagram of Darpan Dental Home Application.....	23
Figure 6: Login Interface Design.	25
Figure 7: Feedback Interface Design.	26
Figure 8: Follow Up Reminder Interface Design.....	27
Figure 9: Staff Registration Interface Design.....	28
Figure 10: Set Reminder Interface Design.	29
Figure 11: De-register Staff Interface Design.....	29
Figure 12: Book Appointment Interface Design.....	30
Figure 13: Patient Registration Interface Design.....	31
Figure 14: Payment Interface Design.....	32
Figure 15: Generate Report Interface Design.	33
Figure 16: Record Checkup Report Interface Design.....	34
Figure 17: View Report Interface Design.....	35

Tables of Tables

Table 1: High level description of Login.....	8
Table 2: High level description of register patient.....	8
Table 3: High level description of register staff.....	8
Table 4: High level description of manage appointment.....	9
Table 5: High level description of manage report.	9
Table 6: High level description of manage profile.....	9
Table 7: High level description of terminates staff.	9
Table 8: High level description of send feedback.	10
Table 9: High level description of maintain payment details.....	10
Table 10: High level description of record checkup details.....	10
Table 11: High level description of make payment.	10
Table 12: Expanded Use Case Description of patient registration.	11
Table 13: Expanded Use Case Description of payment.	12

1. Introduction

In this course work I was given a scenario where I am a software engineer from a software development company. A firm named "Darpan Dental Home" hired me to provide a computerized information system to manage patient records and payment data. This clinic faces a lot of problems in holding the patient's records with their payment information. Hence, we need to reduce the issue by creating a computerized system that helps to handle the patient and their payment information.

This computerized system will allow the company to tackle the problems of record management. This application will allow them to be more accurate when it comes to bulk data recording and also increase their work effectiveness due to a strong recording system, and they will also have all the customer payment files and all the payment information that will help a lot in maintaining and handling better records that will directly boost the performance and profitability of the Dental Clinic and making it secure and more sustainable.

As a software engineer, we have chosen the RUP methodology for developing computerized system for Darpan Dental Home Clinic. RUP stands for "Rational Unified Process". This is an iterative software development process initially produced by Rational Software Corporation that IBM acquired in 2003 which offers a structured approach to assigning roles and responsibilities within a production organization with the goal of ensuring that the creation of high-quality software meets the needs of its end-users within a consistent timeline and budget (Rouse, 2007). The four phases of RUP are as follows:

Inception:

It is the initial step in the cycle of development. In this step the team ultimately decides the project's structure and basic concept. The development team determines whether the project is worth pursuing and what resources are required. It is almost like a project assessment.

Transition:

This is the final phase of the development process. The transition phase is the moment the product finishes, releases and delivers to customers. Final adjustments or updates are made based on feedback from end users.

Elaboration:

This is the second phase of the development process. The developers are taking a closer look at the project at this point to evaluate its design base and to evaluate the design in relation to the project. This stage is critical for the RUP because developers are evaluating the risks associated with changing the project scope or introducing new technologies along the way. It is also the stage where the team must determine whether or not to start building (development and coding).

Construction:

This is the third phase of the development process. At this stage, the project's production is finished. The program design is complete, and the source code is written down. At this point the program is checked to determine if the project has accomplished its goal set out in the initiation process. The program is successfully developed, written down, and tested.

Transition:

This is the final phase of the development process. The transition phase is the moment the product finishes, releases and delivers to customers. Final changes or improvements are made based on end-user feedback (Christensson, 2016).

2. Gantt Chart

As a software engineer, we have planned to develop the system using RUP methodology. So, we have developed a Gantt chart reflecting the RUP methodology of planning. The Gantt Chart is a project management tool that can be used to display the project and its achievements over time. It offers a visual guide to the start and end dates of the different operations, the status of those operations and whether those activities have been completed or not (Rouse, 2017).

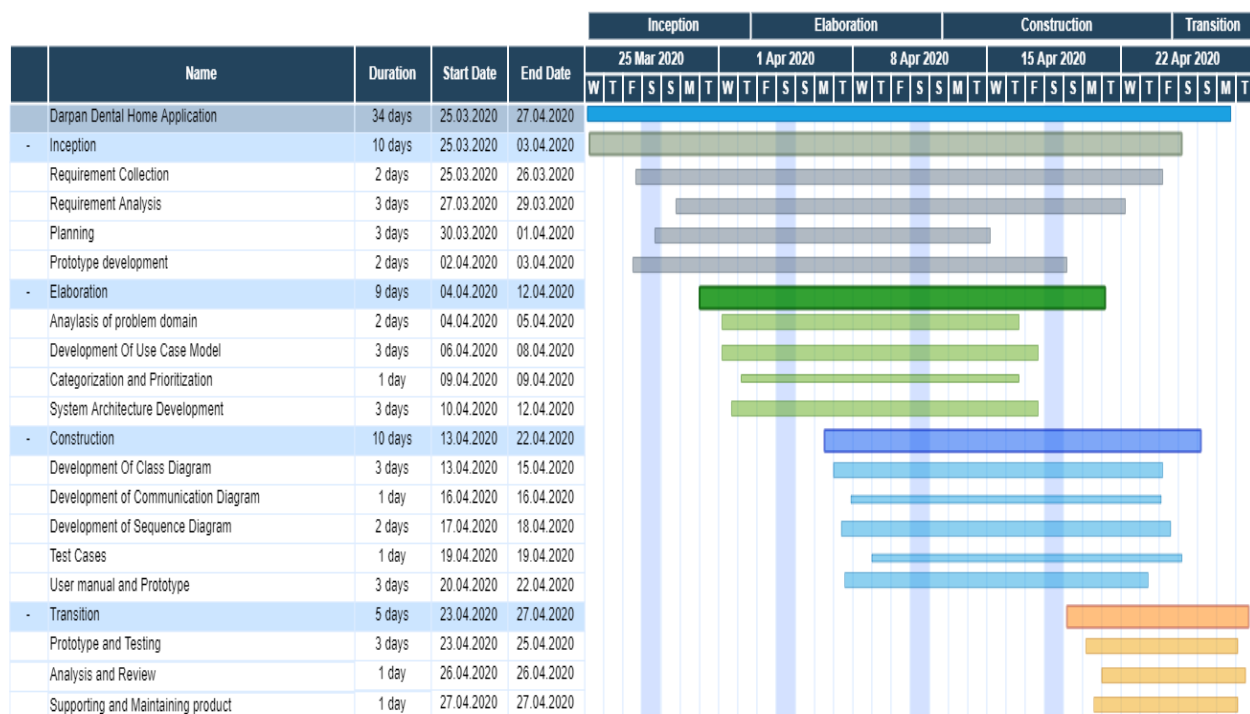


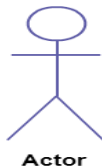
Figure 1: Gantt Chart

3. Use Case Diagram

Use case diagram is a type of behavioral UML diagram and is used for analysis of various structures. Use cases define the action anticipated (what), and not the exact process (how) to make it happen. They enable you to imagine the various types of roles within a system and how they communicate with the system. A core principle for use case modeling is that it allows one to build a program from the viewpoint of the end user. This is an efficient way of communicating system actions under the context of the user by defining all externally observable system behavior (Krishna, 2018). The use case diagram consists of four major parts or objects as follows.

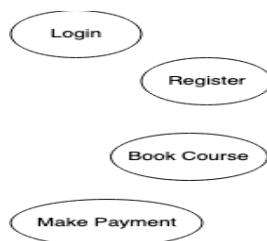
1. Actor

An actor is any kind of entity which plays a role in any given system. This may be a company, a person or an external process and is typically defined by a stickman.



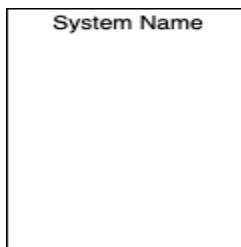
2. Use Case

Within the program, a use case represents a function or action. Drawn as an oval, it is called with the function.



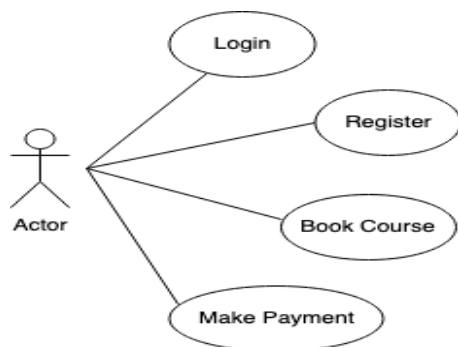
3. System

The project that we are developing is a system. System is used to describe the use case scope and is drawn as a rectangle where the name of the system is written inside the top of the rectangle that is system. All the function the system performs are placed inside the rectangle.



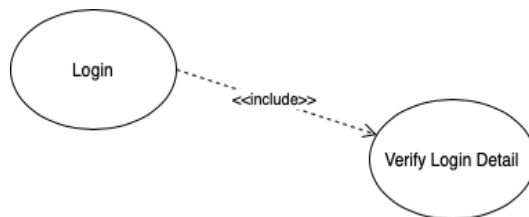
4. Relationship

As we know, by communicating with at least one of the Use Cases within the program, an individual uses the system to accomplish a purpose. For instance: clients can log in, register, book the service, make payments, etc. And to demonstrate the relationship, we draw the firm line between the Actor and the Use Case. It consists of three different types; included, excluded, and generalization.



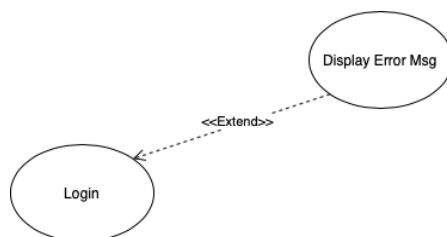
Include Relationship

In an include relationship, the included use case will be auto executed while the base use case is in execution. A dashed line with an arrow pointing to the included use case is drawn with << include >> as a label to show Include Relationship.



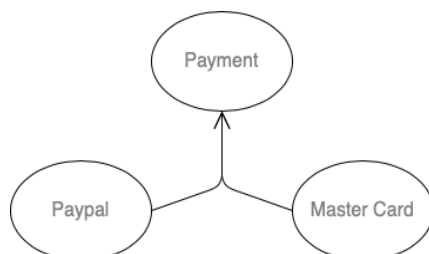
Extend Relationship

In an Extend relationship the case of extended use may or may not be executed while the case of base use is executed. The extended usage case can only be under certain conditions. A dashed line with an arrow pointing to the base use case is drawn with << extend >> as a label to show Extend Relationship.



Generalization Relationship

It shows the relationship between the parent and the child use case relationship. A strong line with an arrow from the child use case or actor to parent use case or actor is drawn to represent generalization relationship (Master2teach, 2019).



Use case diagram

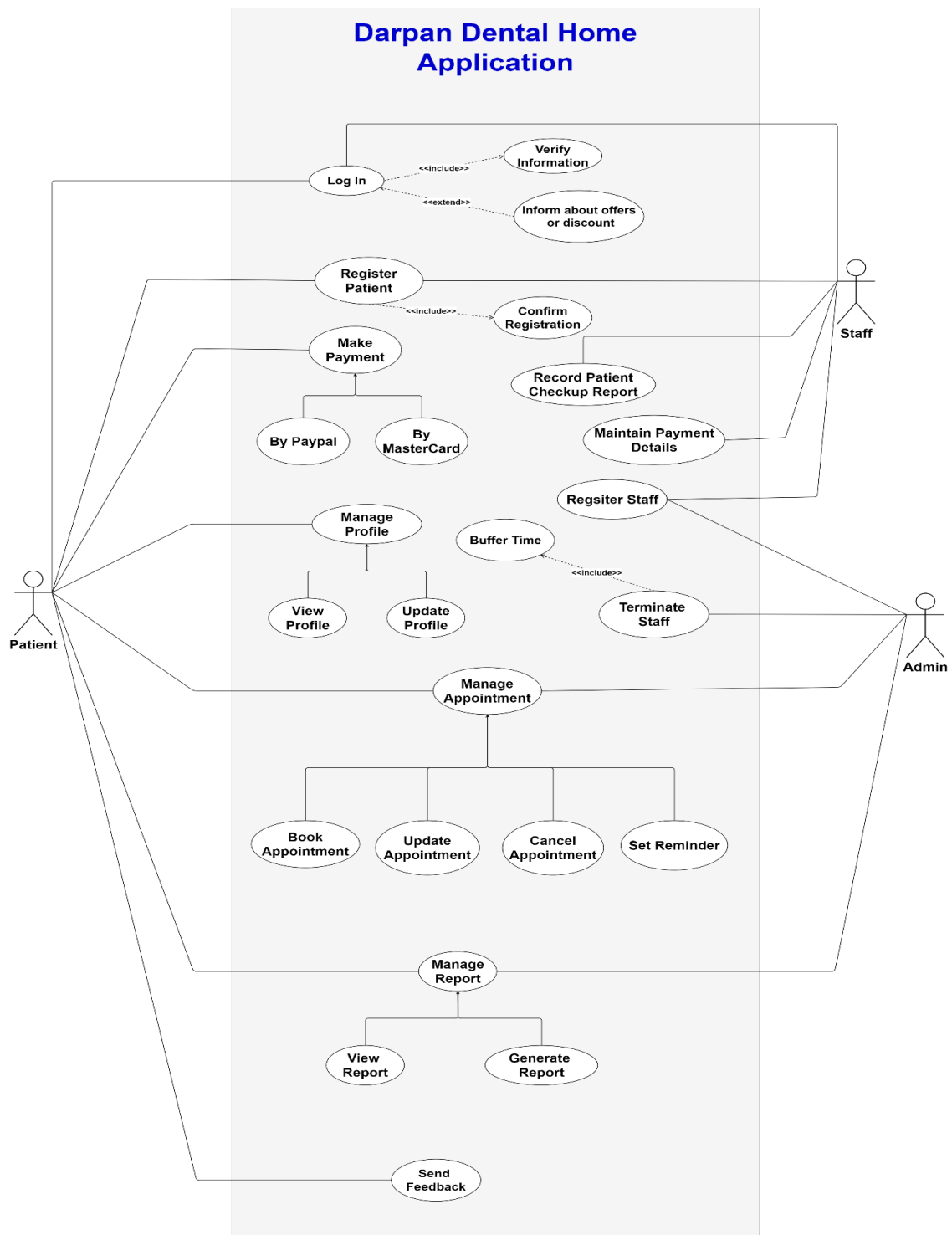


Figure 2: Use Case Diagram

3.1 High Level Use Case Description

A brief summary of each necessary process which is not detailed. Do not go through the interaction in detail.

Use Case:	Login
Actor:	Patient / Staff
Description:	Login is used for joining systems by giving username and password. He / She will be logged in the system if the username and password is correct, else the login failed message will appear.

Table 1: High level description of Login

Use Case:	Register Patient
Actor:	Patient / Staff
Description:	A new patient provides personal details and fills the registration form. A staff verifies the details and registers the patient.

Table 2: High level description of register patient.

Use Case:	Register Staff
Actor:	Staff/Admin
Description:	Staff provides the detailed information on the registration document. Admin verifies the details and registers the staff.

Table 3: High level description of register staff.

Use Case:	Manage Appointment
Actor:	Patient / Admin
Description:	Patients can book, update, cancel the appointment and set reminders whereas admin verifies the details and cancels the unwanted appointment.

Table 4: High level description of manage appointment.

Use Case:	Manage Report
Actor:	Patient / Admin
Description:	Patients view the report whereas admin generates the report of the patient.

Table 5: High level description of manage report.

Use Case:	Manage Profile
Actor:	Patient
Description:	Patients can view and update their profile.

Table 6: High level description of manage profile.

Use Case:	Terminate staff
Actor:	Admin
Description:	Admin de register the unwanted staff from system.

Table 7: High level description of terminates staff.

Use Case:	Send feedback
Actor:	Patient
Description:	Patients provide comments, suggestions, and other necessary feedback for the betterment of the system.

Table 8: High level description of send feedback.

Use Case:	Maintain Payment Details
Actor:	Staff
Description:	Staff checks and maintains the payment details of a patient in a system.

Table 9: High level description of maintain payment details.

Use Case:	Record Checkup Report
Actor:	Staff
Description:	Staff records and maintains the checkup details of a patient in a system.

Table 10: High level description of record checkup details.

Use Case:	Make Payment
Actor:	Patient
Description:	Patients pay payment through PayPal or MasterCard for appointment.

Table 11: High level description of make payment.

3.2 Expanded Use Case Description

Event occurs for patient registration

Patient	System
1. Clicks on the register button.	
	2. Display registration forms.
3. Inserts all the information required for registration.	
4. Clicks on the submit button.	
	5. Checks and validates all the information.
	6. Stores the details of the patient in the system.
7. Stores the details of the patient in the system.	

Table 12: Expanded Use Case Description of patient registration.

Alternatives:

Line 4: If any field is empty, error message is displayed, and use-case ends.

Line 5: If user input is not valid with the system, use case ends.

Event occurs for payment

Patient	System
1. Open payment page.	
	2. Displays the field of information to be filled for making payment.
3. Inserts all the required information like payment amount and its type.	
4. Clicks on the pay button.	
	5. Evaluates all the payment information is valid or not.
6. Gets confirm message for payment and patient confirms the message.	
7. Payment done message received.	

*Table 13: Expanded Use Case Description of payment.***Alternatives:**

Line 4: If any field is empty, error message is displayed, and use-case ends.

Line 6: If user denied the confirmation, payment cancel message is displayed, and use-case ends.

4. Communication diagram

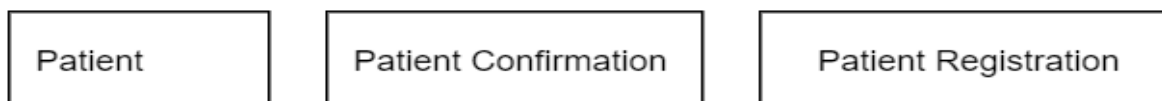
A communication diagram which is also called a collaboration diagram illustrates the relationships and interactions between software objects in the Unified Modeling Language (UML). These diagrams can be used to represent the dynamic behavior of a specific use case, and to define each object's function.

A collaborative diagram is similar to a flowchart that represents the functions, features and actions of individual objects as well as the overall operation of the system in real time. Objects are shown as rectangles with label names within. These object names are accompanied by colons and can be highlighted. The relationships between the objects are displayed as lines which connect the rectangles. Messages between objects are displayed as arrows which connect the related rectangles with the labels specifying the message sequence (JavaT, 2018).

4.1 Steps for developing communication diagram

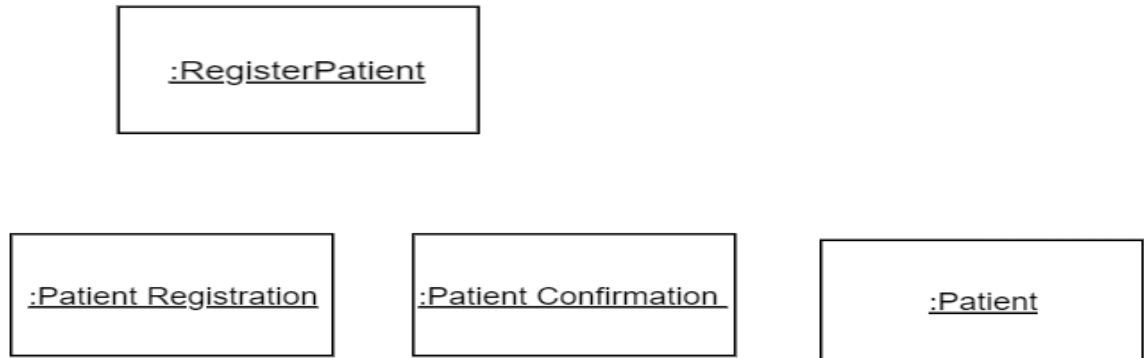
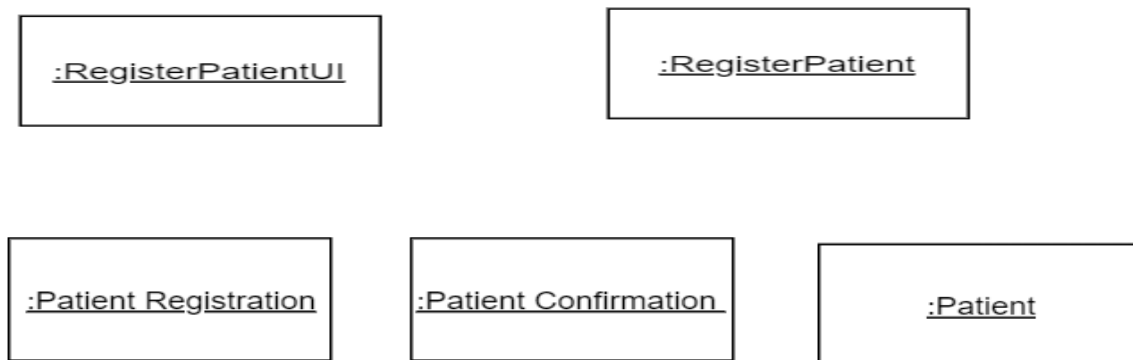
Step 1: Find the domain classes.

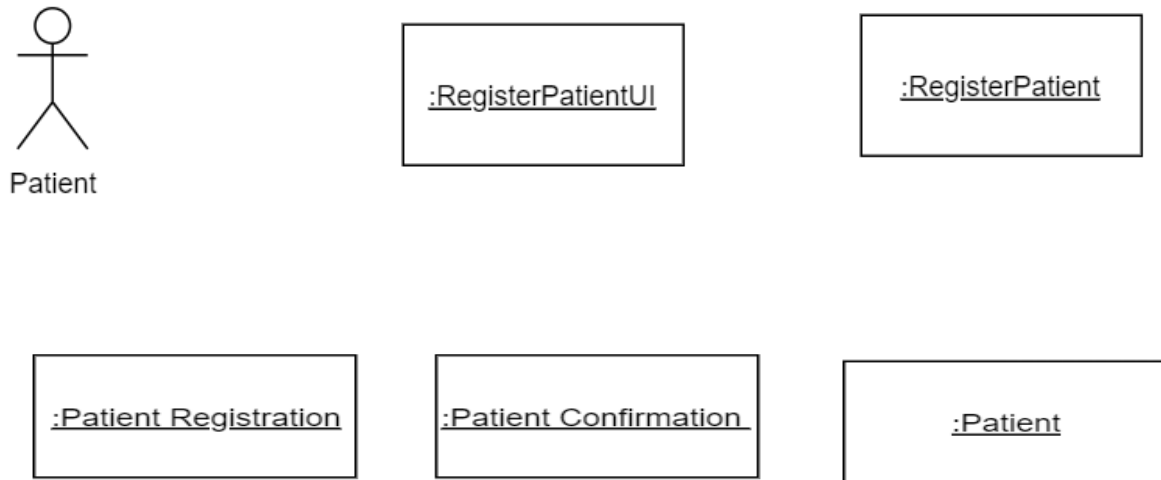
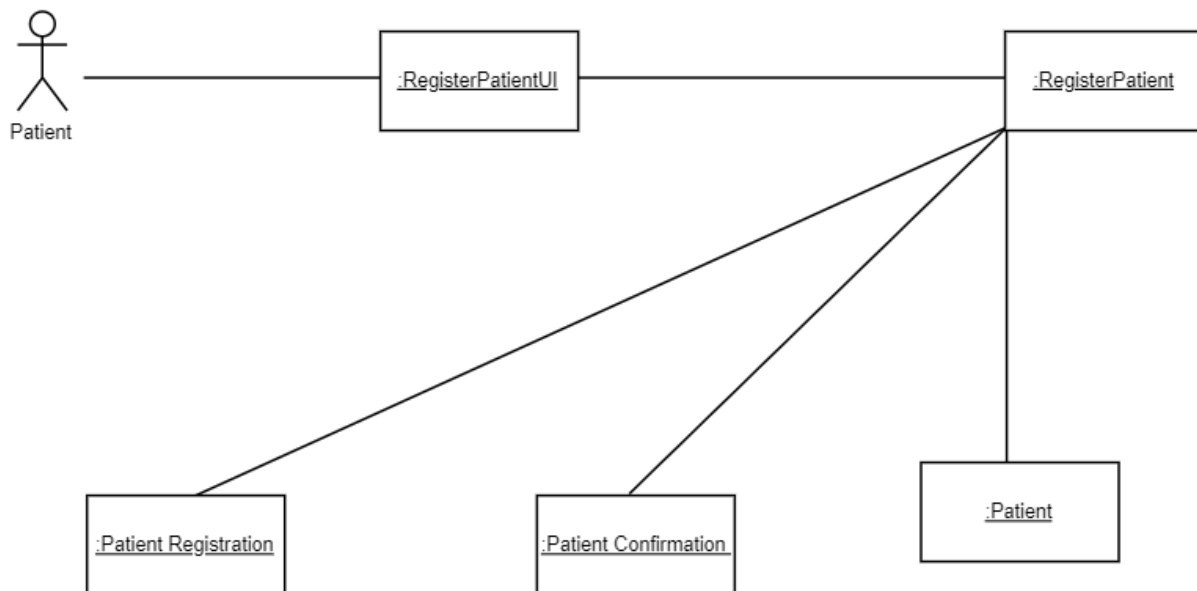
The domain classes for registration are Patient, Patient Confirmation, Registration



Step 2: Draw an object symbol for each of the domain classes.



Step 3: Add a control object.**Step 4: Draw Boundary object.**

Step 5: Draw Actor.**Step 6: Add associations**

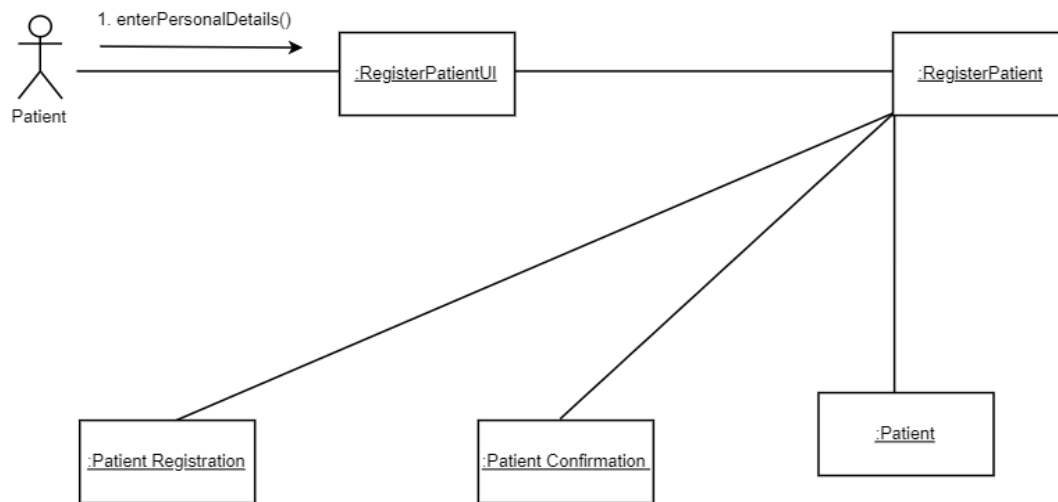
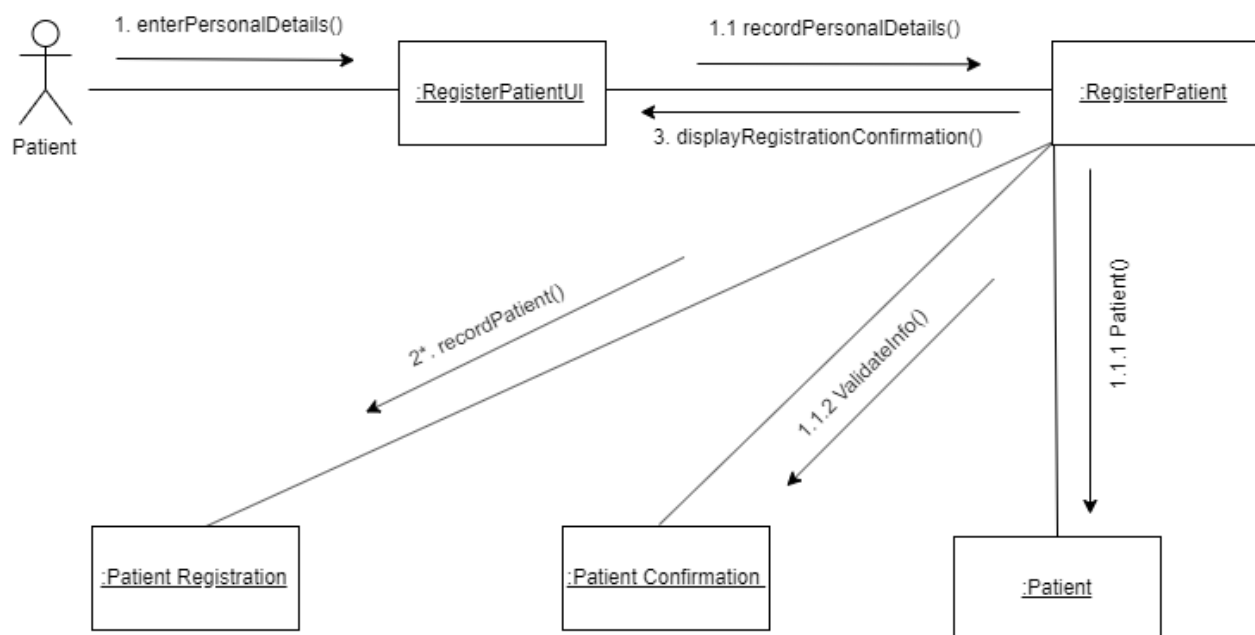
Step 7: Add message**4.2 Collaboration diagram**

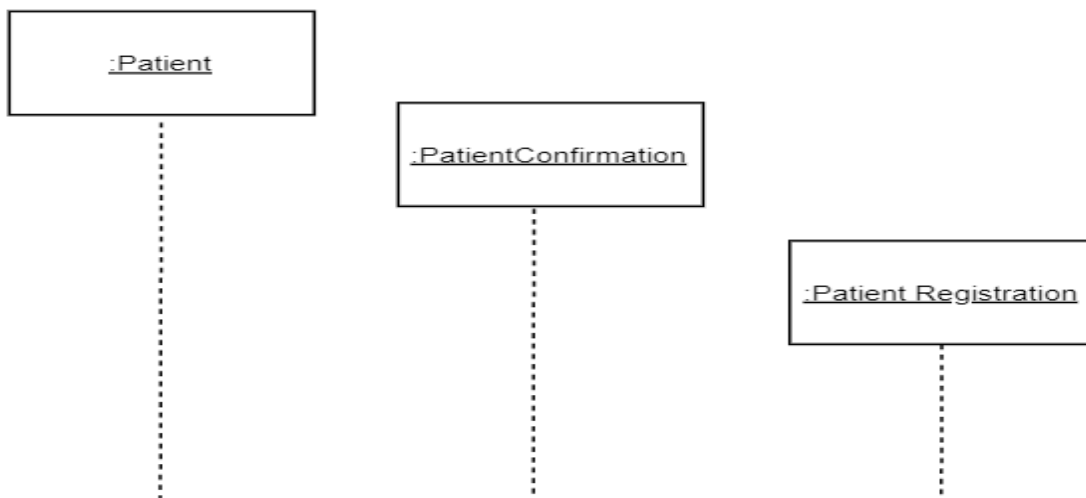
Figure 3: Collaboration diagram of patient registration.

5. Sequence diagram

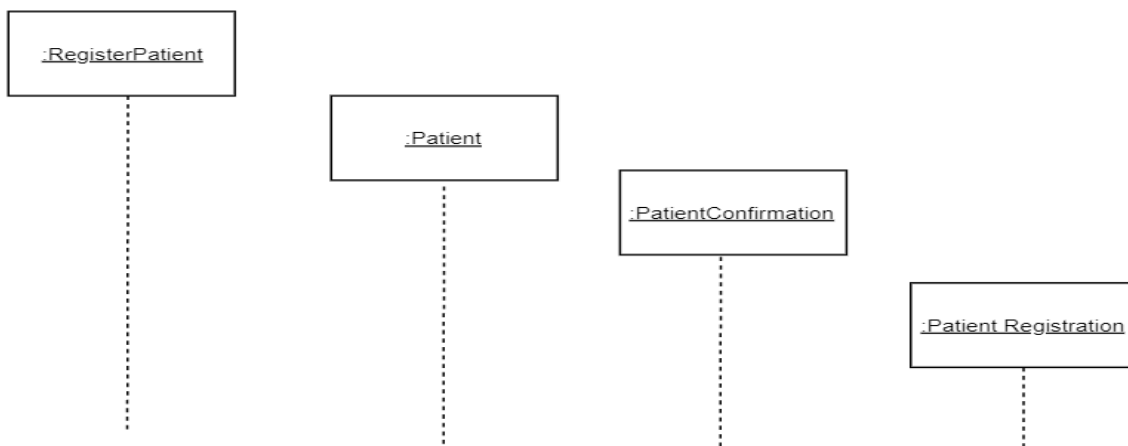
UML Sequence Diagrams are diagrams of interactions that describe how operations are performed. In the context of a collaboration they capture the interaction between objects. Sequence Diagrams are time based and visually show the order of the interaction by using the diagram's vertical axis to represent time which messages are sent and when (GeeksforGeeks, 2015).

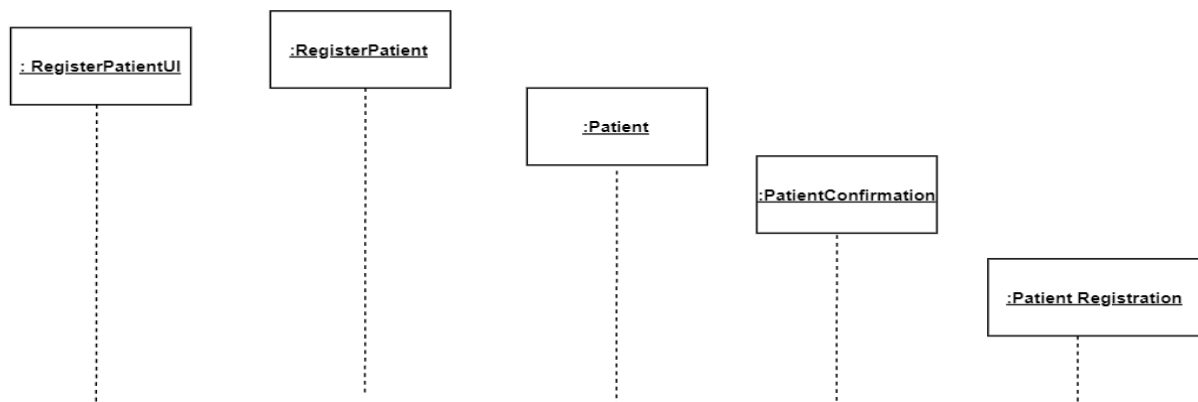
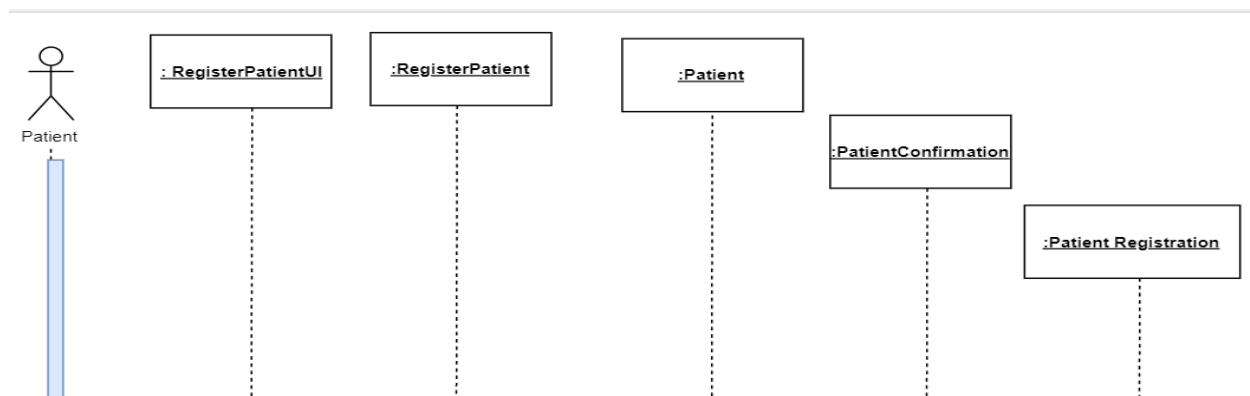
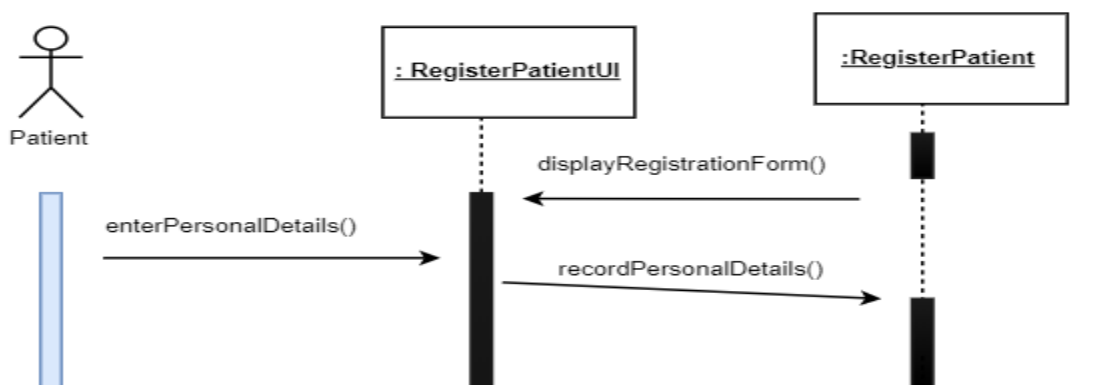
5.1 Steps to draw a sequence diagram are:

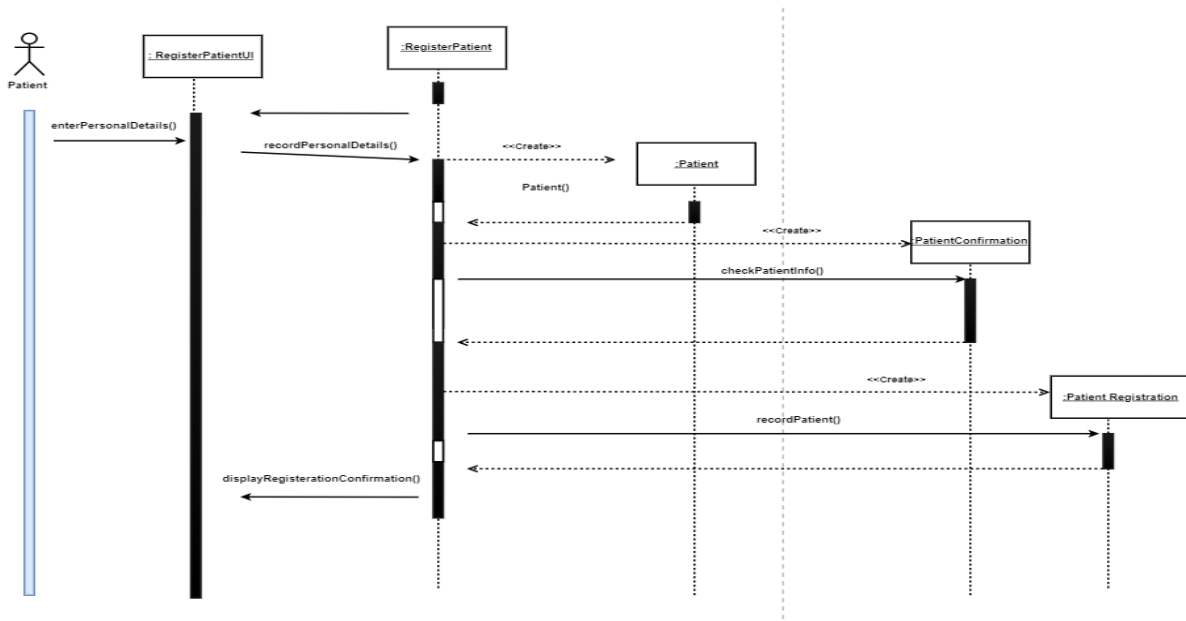
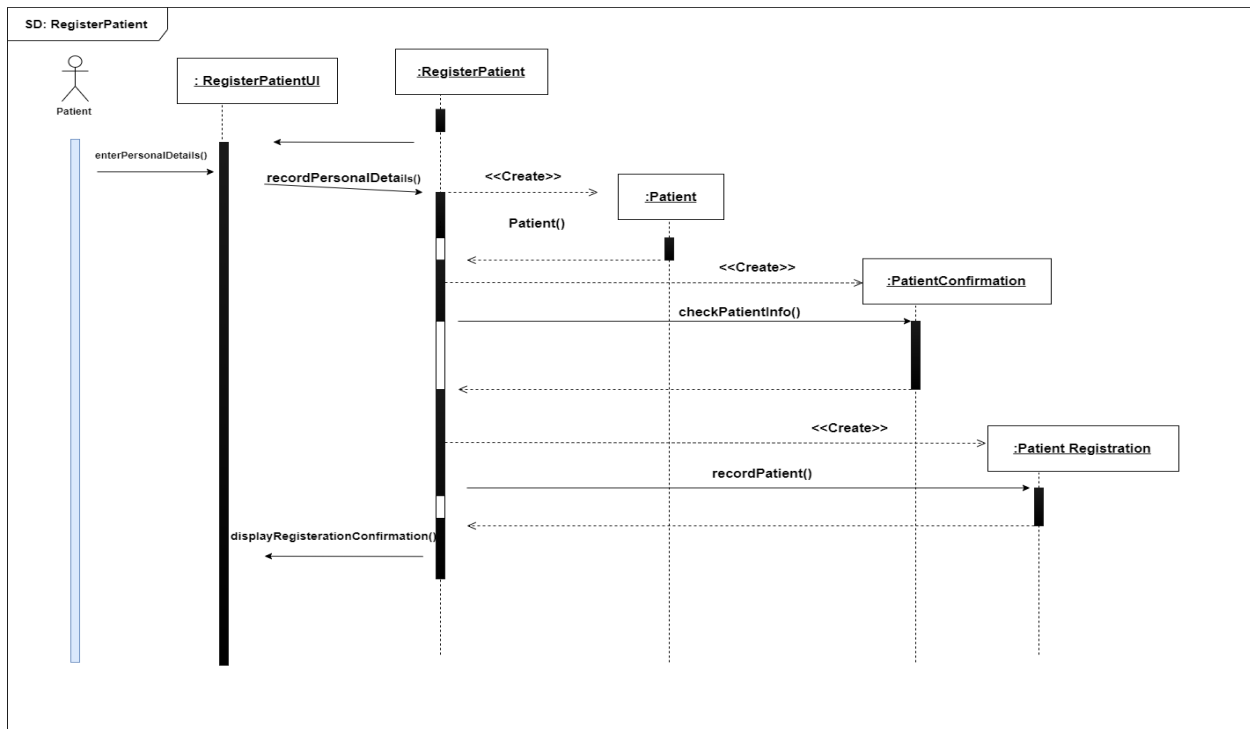
Step 1: Find the domain classes.



Step 2: Draw control Object Lifeline.



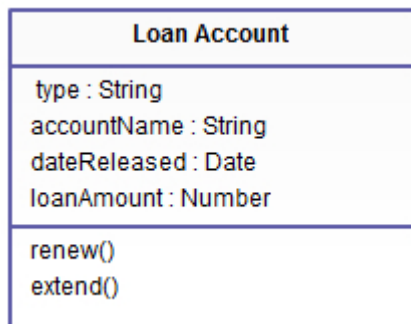
Step 3: Draw Boundary Object Lifeline.**Step 4: Draw Actor Lifeline.****Step 5: Adding Messages.**

Step 6: Draw Object Lifeline.**5.2 Sequence Diagram***Figure 4: Sequence diagram of patient registration.*

6. Class Diagram

Class Diagram is one of the essential UML software development diagrams that displays the classes of objects in the system and the relations between those classes. It is not used only to illustrate, explain, and report different aspects of a program, but also to create the software application's executable code.

The class diagram is represented by a rectangular box containing three compartments and is depicted on the class diagram. We specify the class name in the top container, we specify the attributes in the second or middle one, while the third or last one is for the methods or operations to be displayed (Tutorials Point, 2017). The example of class diagram representation is given below:



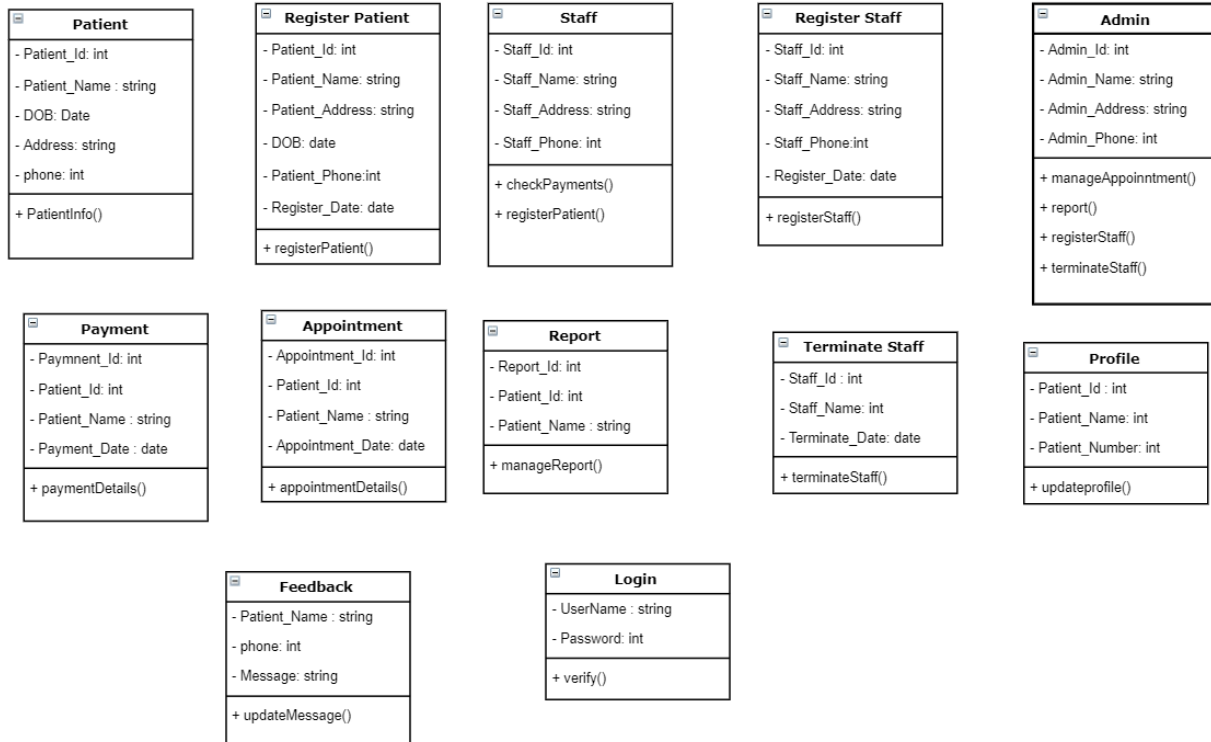
6.1 Steps for developing class diagram

Step 1: List out all the domain classes from each use case.

Use Case	Domain class
Register patient	Patient, Register Patient
Login	Login, Staff, Patient
Make Payment	Patient, Payment
Manage Profile	Patient, Profile
Manage Appointment	Patient, Appointment, Admin
Manage Report	Patient, Admin, Report
Record Patient Checkup Report	Staff, Maintain Details
Maintain Payment Details	Staff, Payment
Register staff	Admin, Staff, Register Staff
Terminates staff	Admin, Staff, Terminate Staff
Send Feedback	Patient, Send feedback

Step 2: Find the unique domain classes.

Domain classes
Admin, Staff, Patient, Register Patient, Register Staff, Terminate Staff, Report, Appointment, Send Feedback, Payment, Feedback, Profile, Login,

Step 3: Draw the Class Diagram of each domain class

Step 4: Showing Relations, Inheritance, Dependency Relationship and Adding Association Class.

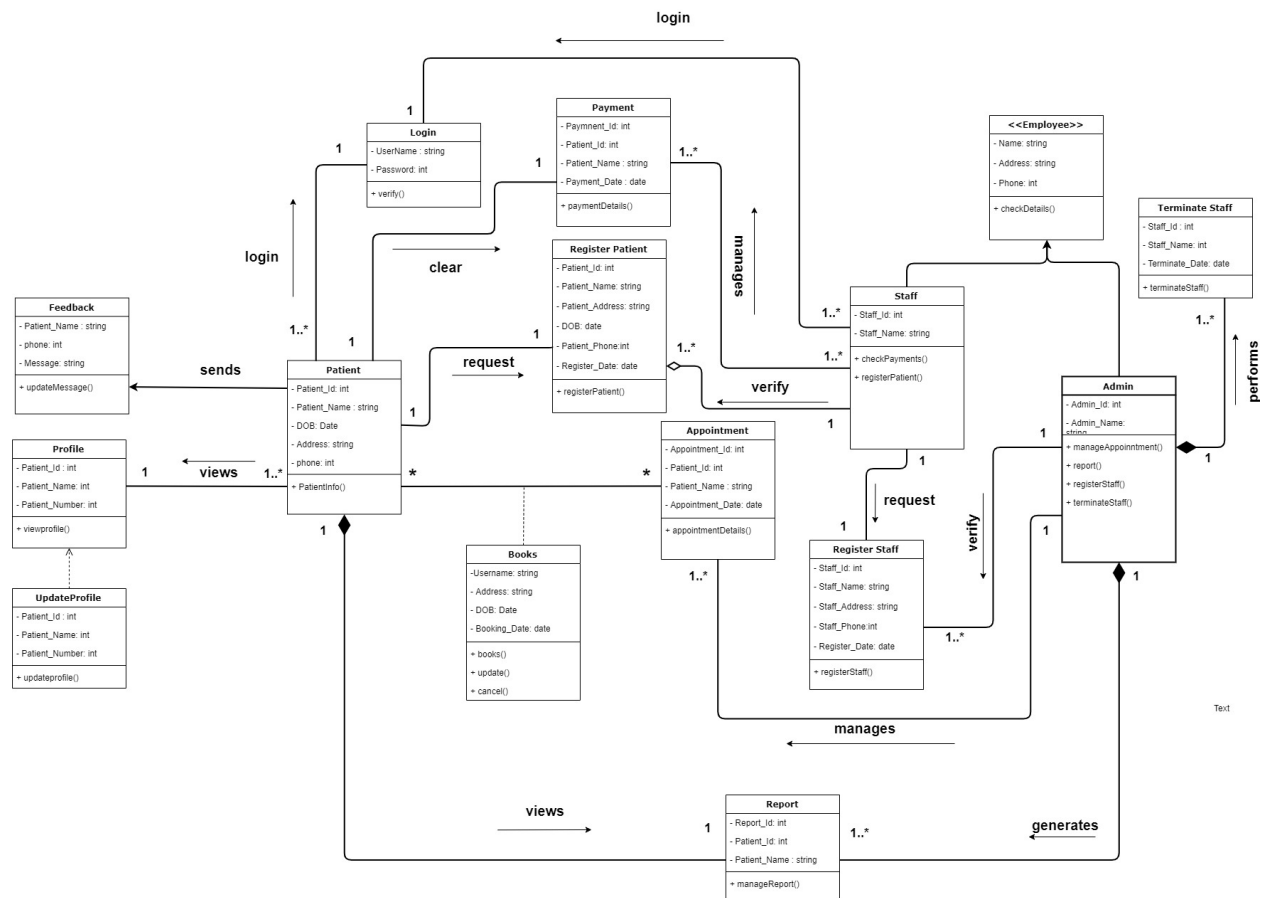


Figure 5: Class Diagram of Darpan Dental Home Application

7. Further Development Process

When designing any application, proper planning and management should be done. As a software engineer, for developing the “Darpan Dental Home Application” system the rational unified process methodology is used. The rational unified method is an iterative software development process where each lifecycle can be replicated for the desired outcome.

A Gantt chart showing the time flow of each step of the RUP methodology is generated. Following the RUP approaches we complete the inception phase. In the beginning of the elaboration phase, all the needed requirements are gathered, the gathered requirements are verified and validated, and planning is made for developing the system in next phases. However, every phase is further broken down into iterations as the program follows RUP methodology and adjustment or changing the requirements can be adopted. Moving towards the middle phase of elaboration, all processes such as the gathering of requirements, verification, modification of design, coding, testing begin simultaneously. As per planning, the developer starts developing the use case model and system, the planner modifies and verifies the changes, the designer updates the design as per the change, programmer starts developing the system whereas the system testing also starts. So, the further development process takes place like elaboration phases.

As planned in Gantt chart, in the ending phase of construction, the program design is complete, and the source code is written down. At this point the program is checked to determine if the project has accomplished its goal set out in the initiation process. In transition phases, Final changes or improvements are made based on end-user feedback. After completing all the phases of RUP methodology, the working system of Darpan Dental Home will be developed with all the features for recording the patient details and their payment types.

8. Prototype



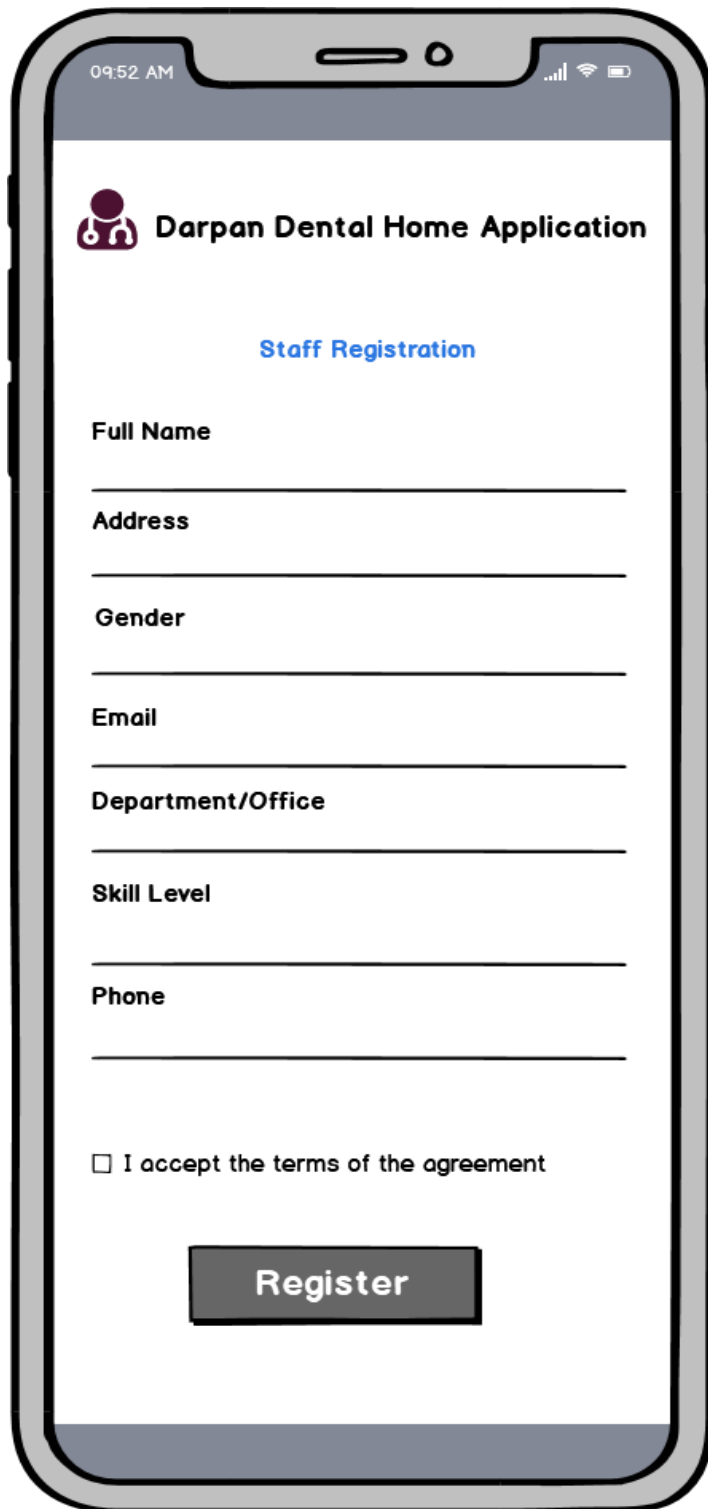
Figure 6: Login Interface Design.

The image shows a mobile application interface for 'Darpan Dental Home Application'. At the top, the status bar displays '09:52 AM' and signal icons. The app header includes a purple icon of two people and the title 'Darpan Dental Home Application'. Below the header, the word 'Feedback' is written in blue. The form consists of three input fields: 'Full Name', 'Email', and 'Message', each with a horizontal line for text entry. At the bottom, there is a dark grey rectangular button with the word 'Submit' in white text.

Figure 7: Feedback Interface Design.

The image shows a mobile application interface for 'Darpan Dental Home Application'. At the top, the status bar displays '09:52 AM' and signal icons. The app header includes a dental icon and the title 'Darpan Dental Home Application'. Below this, the section 'Follow Up Reminder' is highlighted in blue. The form contains a 'Patient Name' label followed by a horizontal line. A 'Date :' label is next to a date input field with slashes and a calendar icon. Below the date field are four checkboxes with labels: 'Check Up', 'Follow up', 'Appointment Date', and 'Renew Appointment'. An 'Offers and Discount' label is followed by another horizontal line. At the bottom, there is a large grey button labeled 'Send Reminder'.

Figure 8: Follow Up Reminder Interface Design.



The image shows a mobile application interface for 'Darpan Dental Home Application'. At the top, the status bar displays '09:52 AM' and signal icons. The app header includes a purple icon of two people and the title 'Darpan Dental Home Application'. Below this, the section 'Staff Registration' is highlighted in blue. The form consists of several text input fields: 'Full Name', 'Address', 'Gender', 'Email', 'Department/Office', 'Skill Level', and 'Phone'. Each field is followed by a horizontal line for text entry. At the bottom of the form, there is a checkbox labeled 'I accept the terms of the agreement'. Below the checkbox is a dark grey button with the text 'Register' in white.

Figure 9: Staff Registration Interface Design.

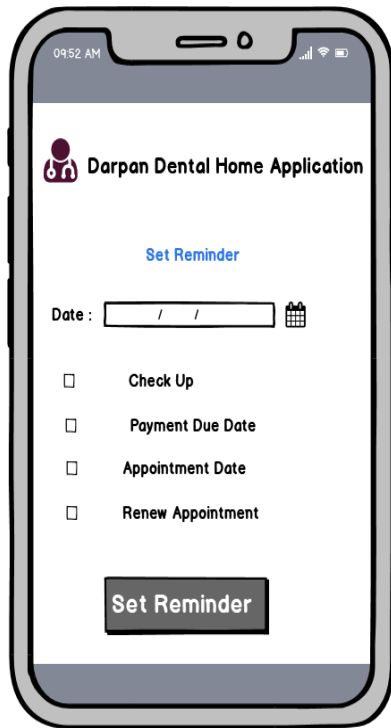


Figure 10: Set Reminder Interface Design.

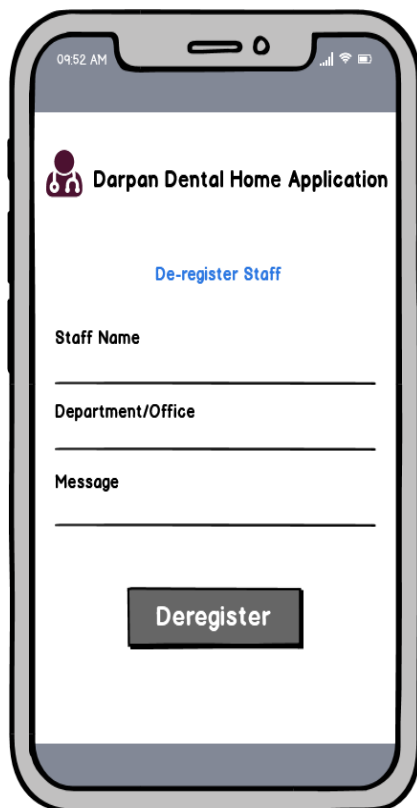


Figure 11: De-register Staff Interface Design.

The image shows a mobile application interface for 'Darpan Dental Home Application'. At the top, the status bar displays '09:52 AM' and signal icons. The app header includes a dental icon and the title 'Darpan Dental Home Application'. Below the header, the text 'Send an Request for Appointment' is centered. The form is divided into two sections: 'Doctors Information' and 'Patient Information'. The 'Doctors Information' section contains three input fields labeled 'Departments', 'Doctors', and 'Appointment Date'. The 'Patient Information' section contains five input fields labeled 'Full Name', 'Age', 'Gender', 'Address', and 'Phone'. Below these fields is an 'Email' label with an input field. At the bottom of the form is a large 'Book Appointment' button.

09:52 AM

Darpan Dental Home Application

Send an Request for Appointment

Doctors Information

Departments

Doctors

Appointment Date

Patient Information

Full Name

Age

Gender

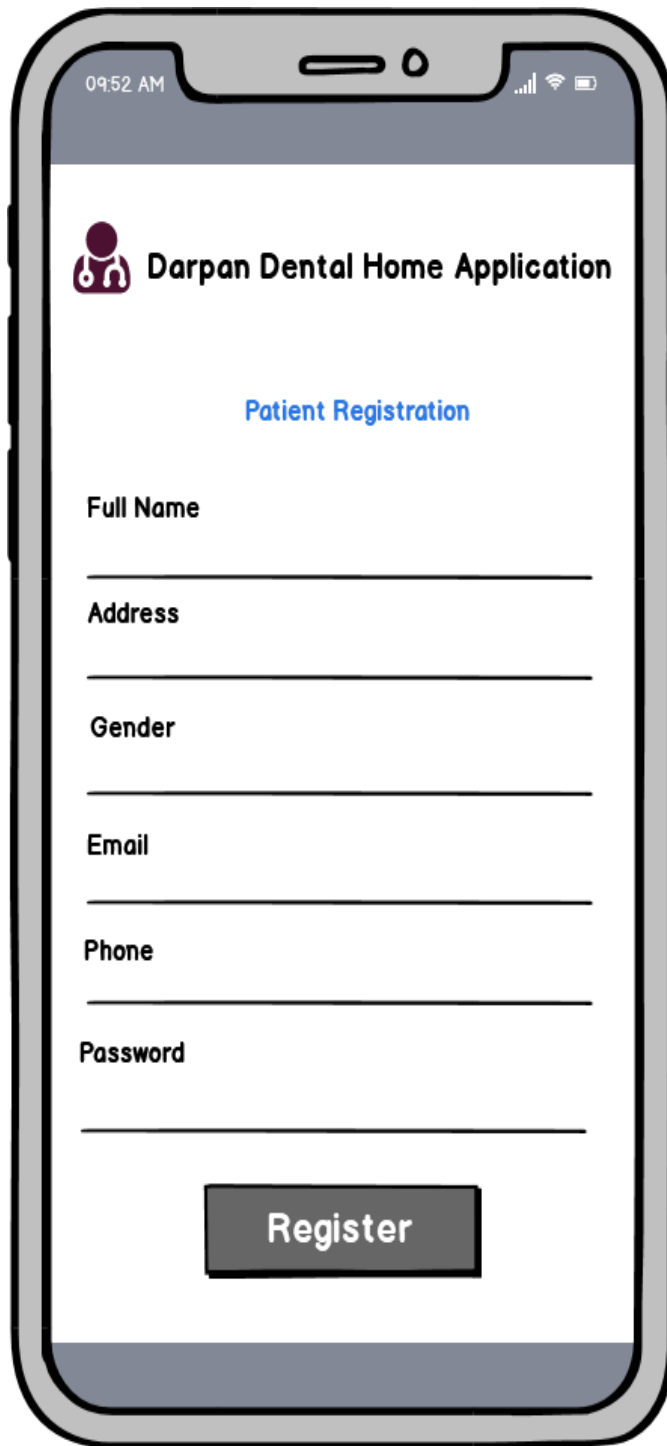
Address

Phone

Email

Book Appointment

Figure 12: Book Appointment Interface Design.



The image shows a mobile application interface for 'Darpan Dental Home Application'. At the top, there is a status bar with the time '09:52 AM' and icons for signal, Wi-Fi, and battery. Below the status bar is a header with a purple icon of a person and a dental chair, followed by the text 'Darpan Dental Home Application'. The main section is titled 'Patient Registration' in blue. It contains six input fields with labels: 'Full Name', 'Address', 'Gender', 'Email', 'Phone', and 'Password'. Each label is followed by a horizontal line for text entry. At the bottom of the form is a dark gray button with the text 'Register' in white.

Figure 13: Patient Registration Interface Design

The image shows a mobile application interface for a payment screen. At the top, the status bar displays '09:52 AM' and signal icons. The app header features a purple icon of two people and the text 'Darpan Dental Home Application'. Below this, the word 'Payment' is written in blue. The form contains several input fields: 'Full Name', 'Email', 'Date of register', 'Card Number', and 'Payment Amount'. Under 'Payment type', there are two radio buttons; the first is followed by a blue 'PayPal' logo, and the second is followed by a blue 'Credit Card' logo. Below the radio buttons is a checkbox labeled 'I accept the terms of the agreement'. At the bottom of the form is a large grey button with the text 'Payment'.


Figure 14: Payment Interface Design.

The image shows a mobile application interface for 'Darpan Dental Home Application'. At the top, the status bar displays '09:52 AM' and signal icons. The app header includes a purple icon of two people and the title 'Darpan Dental Home Application'. Below the header is a blue link labeled 'Generate Report'. The form contains the following fields: 'Report Type' (text input), 'Report Title' (text input), 'Date' (text input with a calendar icon), 'Patient Name' (text input), and 'Description' (text area with three lines). A large grey button labeled 'Generate' is positioned at the bottom of the form.


Figure 15: Generate Report Interface Design.

The image shows a mobile application interface for 'Darpan Dental Home Application'. The status bar at the top displays '09:52 AM' and signal icons. The app header includes a dental icon and the title 'Darpan Dental Home Application'. Below the header is a blue link labeled 'Record Checkup Report'. The form contains the following fields: 'Date' with a date picker icon, 'Patient Name', 'Report Type', 'Report Title', and 'Description'. At the bottom are 'Save' and 'Cancel' buttons.

09:52 AM

 **Darpan Dental Home Application**

[Record Checkup Report](#)

Date 

Patient Name

Report Type

Report Title

Description

Figure 16: Record Checkup Report Interface Design.

The image shows a mobile application interface for 'Darpan Dental Home Application'. The screen is titled 'View Report' in blue text. Below the title, there is a 'Date' field with a calendar icon, a 'Name' field, a 'Report Title' field, a 'Report Type' field, and a 'Report Details' field. At the bottom, there are two buttons: 'Save' and 'Exit'.

09:52 AM

Darpan Dental Home Application

[View Report](#)

Date / /

Name

Report Title

Report Type

Report Details

Figure 17: View Report Interface Design.

9. Conclusion

This coursework was related to creating a computerized information system to maintain records of customers and their payment details of Darpan Dental Home. The development phase began after determining the necessary requirements and specifications for Darpan Dental Home Application. All the required design and diagrams such as Use Case diagram, Collaboration diagram, Sequence diagram, Class diagram was developed according to the question requirement with the help of draw.io. All the other valuable documentation and writing were referred from the tutors and college's various study materials for developing this project.

This report involved the creation of a Gantt chart to show the timeline of the work and the history of the research I have done, and this also included the collaboration diagram and the class model that offers all the details of the system the user needs to understand. This project also provides all the clarification of how the use case model was used to construct the collaborative diagram and the sequence diagram. The models and diagrams for the app were carefully made without any errors and every detailed specification was studied and the specification and necessity that had to be met in the app were fulfilled.

Developing this coursework as per the scenario was a major challenge for me, I had faced several obstacles during the completion of this course work. With the help of module leader and tutor, all the questions and problems that emerged during the course work were easily tackled. I had done a lot of research and study for completing this coursework and app design. During this course, I got a chance to know more about Rational Unified Process (RUP) methodology which provided a very good understanding of the design and implementation to create any software for solving different problems. This coursework makes us adequate experience in designing software and provides us knowledge and understanding the value of necessary research and problem solving.

References

Christensson, P., 2016. "RUP Definition." *TechTerms*. [Online] Available at: <https://techterms.com/definition/rup#:~:text=RUP,implementation%2C%20testing%2C%20and%20deployment>. [Accessed 26 April 2020].

GeeksforGeeks, 2015. *Unified Modeling Language (UML) | Sequence Diagrams*. [Online] Available at: <https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/> [Accessed 6 may 2020].

JavaT, 2018. *UML Collaboration Diagram*. [Online] Available at: <https://www.javatpoint.com/uml-collaboration-diagram> [Accessed 2 may 2020].

Krishna, 2018. *Guru99*. [Online] Available at: <https://www.guru99.com/use-case-diagrams-example.html> [Accessed 28 April 2020].

Master2teach, 2019. *MASTER2TEACH STEP BY STEP GUIDE*. [Online] Available at: <https://master2teach.com/software-engineering/use-case-diagram/> [Accessed 29 April 2020].

Rouse, M., 2007. *TechTarget*. [Online] Available at: <https://searchsoftwarequality.techtarget.com/definition/Rational-Unified-Process> [Accessed 19 April 2020].

Rouse, M., 2017. *Tech Target*. [Online] Available at: <https://searchsoftwarequality.techtarget.com/definition/Gantt-chart> [Accessed 25 April 2020].

Tutorials Point, 2017. *UML - Class Diagram*. [Online]
Available at: https://www.tutorialspoint.com/uml/uml_class_diagram.htm
[Accessed 10 may 2020].