## Module Code & Module Title

## CC5002NA Software Engineering

## Assessment Weightage & Type

## 60% Individual Coursework

## Year and Semester

## 2019 Spring

**Student Name: Bikram Kumar Sharma**

**Group: C14**

**London Met ID: 18030830**

**College ID: np01cp4s190050**

**Assignment Due Date: 2020/06/05**

**Assignment Submission Date: 2020/06/05**

**Title (Where Required): Darpan Dental Home**

**Word Count (Where Required): 4000 words**

*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

## List of Figures

## List of Tables:

## Introduction:

As Software Engineering, the study will examine user requirements and then design, develop and test software applications that will meet those requirements. In this course we need to create a software for Darpan Dental Home (DDH) to provide a computerized information system to keep customer records and payment details using the Unified Software Development Process (USDP)/Rational Unified Process (RUP) with the Unified Modelling Language (UML) to perform the work.

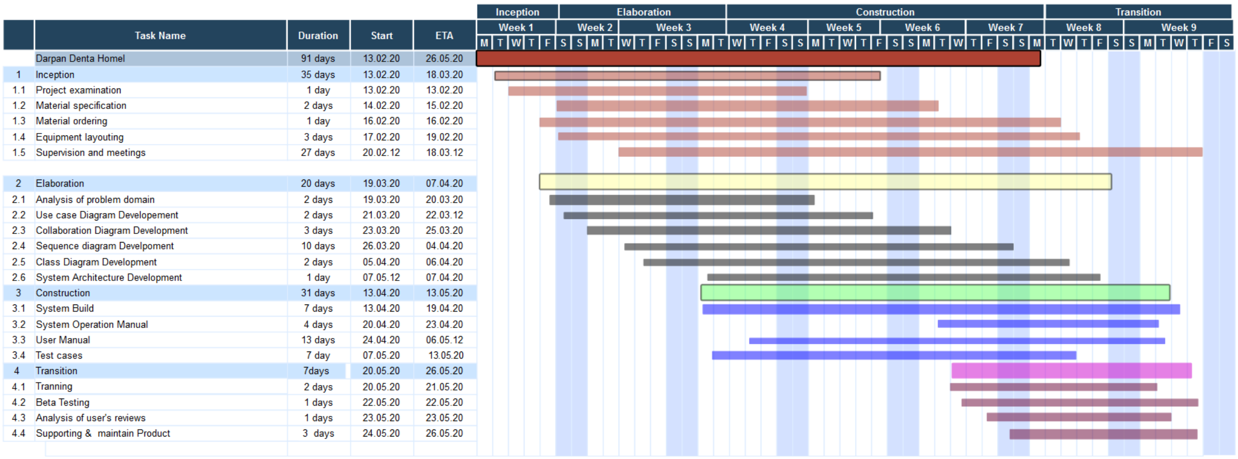
Rational Unified Process (RUP) is an agile methodology for the software development. Project is developed in RUP by means of a series of short, fixed-length iterations. Each iteration has its own activities in the analysis, design, implementation and testing of requirements. RUP's main goal is to build high-quality applications with evolving specifications in fixed cost and fixed time period regardless of whether it comes from the client or the project itself. RUP's main four stages are Conception, Elaboration, Development, and Transformation.

In initiation step team decides the scope, structure and basic project ideas such as: cost and time estimation, preparation, risk management, prototype scheduling tools, etc. The main task in the development phase is: problem domain analysis, development of Use Case Diagram, development of system architecture, and deciding whether or not to start construction phase, etc. The system is designed, developed and tested successfully in the development process and the outcome of this step is the delivery of goods to the consumer and the final phase is the transition phase. In the process of transition: the software is completed, released and shipped to the consumer, team manages all the bug and correct issues, or completed some advance features that were missing during the phases of initiation and development.

## 1. Gantt chart:

A Gantt chart, commonly used in project management, is one of the most popular and useful ways of showing activities (tasks or events) displayed against time. On the left of the chart is a list of the activities and along the top is a suitable time scale. Each activity is represented by a bar; the position and length of the bar reflects the start date, duration and end date of the activity (Anon., n.d.). In the graphical type known as the Gantt map, preparing a software development according to the approach chosen can be reflected. Gantt chart is nothing more than the planning pictorial representation where it shows tasks to be performed against time table.

We need to build the best software as a software developer with fixed costs in limited time. Here we used RUP methodology to do planning for a software creation. This is the type of Gantt cart in which we started this project with data 13-02-2020 and finished with data 26-05-2020, which is clearly reflected through this Gantt chart. We have done all the phases in particular fixed day to do best project. We include the date of all the action performed in the process. The Gantt chart of this system is given below:



|  | Task Name | Duration | Start | ETA |
|---|---|---|---|---|
|  | Darpan Denta Homel | 91 days | 13.02.20 | 26.05.20 |
| 1 | Inception | 35 days | 13.02.20 | 18.03.20 |
| 1.1 | Project examination | 1 day | 13.02.20 | 13.02.20 |
| 1.2 | Material specification | 2 days | 14.02.20 | 15.02.20 |
| 1.3 | Material ordering | 1 day | 16.02.20 | 16.02.20 |
| 1.4 | Equipment layouting | 3 days | 17.02.20 | 19.02.20 |
| 1.5 | Supervision and meetings | 27 days | 20.02.12 | 18.03.12 |
|  |  |  |  |  |
| 2 | Elaboration | 20 days | 19.03.20 | 07.04.20 |
| 2.1 | Analysis of problem domain | 2 days | 19.03.20 | 20.03.20 |
| 2.2 | Use case Diagram Developement | 2 days | 21.03.20 | 22.03.12 |
| 2.3 | Collaboration Diagram Development | 3 days | 23.03.20 | 25.03.20 |
| 2.4 | Sequence diagram Develpoment | 10 days | 26.03.20 | 04.04.20 |
| 2.5 | Class Diagram Development | 2 days | 05.04.20 | 06.04.20 |
| 2.6 | System Architecture Development | 1 day | 07.05.12 | 07.04.20 |
| 3 | Construction | 31 days | 13.04.20 | 13.05.20 |
| 3.1 | System Build | 7 days | 13.04.20 | 19.04.20 |
| 3.2 | System Operation Manual | 4 days | 20.04.20 | 23.04.20 |
| 3.3 | User Manual | 13 days | 24.04.20 | 06.05.12 |
| 3.4 | Test cases | 7 day | 07.05.20 | 13.05.20 |
| 4 | Transition | 7days | 20.05.20 | 26.05.20 |
| 4.1 | Tranning | 2 days | 20.05.20 | 21.05.20 |
| 4.2 | Beta Testing | 1 days | 22.05.20 | 22.05.20 |
| 4.3 | Analysis of user's reviews | 1 days | 23.05.20 | 23.05.20 |
| 4.4 | Supporting & maintain Product | 3 days | 24.05.20 | 26.05.20 |

Inception
Elaboration
Construction
Transition
Week 1
Week 2
Week 3
Week 4
Week 5
Week 6
Week 7
Week 8
Week 9
M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S

*Figure 1: figure of Gantt chart for DDH system by using RUP principle*

## 2. Use Case Model:

It is a model which shows how different types of users interact with the system such as, how it describe the goals of the users, how it interacts between the users and the system.

### 2.1. Use Case Diagram:

Use case diagram is a UML diagram type of behaviour, which is mostly used to evaluate various processes. Use Cases illustrates a system's functionality from the user's perspective. It is used both for modelling a current system and for modelling a proposed system. The main four different use case diagram objects are: System, Actors, Use Cases, and Relation.

#### 2.1.1 Actor:

Use case diagram is a UML diagram type of behaviour, which is mostly used to evaluate various processes. Use Cases illustrates a system's functionality from the user's perspective. It is used both for modelling a current system and for modelling a proposed system. The main four different use case diagram objects are: System, Actors, Use Cases, and Relation.

#### 2.1.2 Use case:

A use case asks for a feature or behaviour within the program. It may be Login, register, and so on, and it is drawn as an oval shape.

### 2.1.3 System:

The project that you are developing is a system. It may be application software, a website, etc and denoted by by:

### 2.1.4 Relationship:

We draw the solid line to the relationship between System Use case and Actor. It can be extension and inclusion, generalization and relationship.

Include occurs when the actor requests to system to do some task and system automatically perform another task, such as automatically checking password when we log in to any system. And Extend Occurs when the actor requests to system to perform any task and system may and may not call another task, such as when we log on to Facebook, error message may or may not take.

Generalization is also known as inheritance, which is used to explain the relationship between patents which cases of child use.

## 2.2. High Level Use Case Descriptions:

It is the short level description of all use cases or a short non-detailed description of each required process. The short description of all the required use cases are given below:

### 2.2.1. Book an Appointment:

| | |
|---|---|
| Use Case: | Book an Appointment |
| Actor: | Patient |
| Description: | A patient can be book an appointment for their treatment from online or by phone. They can pay the payment when they visit the hospital in the day of treatment. |

*Table 1: Description of book an appointment use case*

### 2.2.2. View their Profile:

| | |
|---|---|
| Use Case: | View their Profile |
| Actor: | Patient |
| Description: | A patient can view their profile on hospital's website only after the conformation of appointment registered. |

*Table 2: Description of view their profile use case*

### 2.2.3. View Repot:

| | |
|---|---|
| Use Case: | View Report |
| Actor: | Patient |
| Description: | A patient can view their report after their treatment to know what he/she have problem with their teeth. |

*Table 3: Description of view report use case*

### 2.2.4. Pay the payment:

| | |
|---|---|
| Use Case: | Pay the payment |
| Actor: | Patient |
| Description: | A patient can pay their payment after their treatment. It can be done either from cheque, Credit card or from Debit card after checking payment details. |

*Table 4: Description of pay the payment use case*

### 2.2.5. Can update and send feedback:

| | |
|---|---|
| Use Case: | Can update and send feedback |
| Actor: | Patient |
| Description: | A patient have right to update their profile in hospital's website and they can also leave feedback towards hospital problems. |

*Table 5: Description of can update and send feedback use case*

### 2.2.6. Cancel Appointment:

| | |
|---|---|
| Use Case: | Cancel Appointment |
| Actor: | Patient |
| Description: | A patient can cancel the appointment, if any problem occurs. It can be done as same as booking Appointment. |

*Table 6: Description of cancel appointment use case*

### 2.2.7. Add Staff:

| | |
|---|---|
| Use Case: | Add Staff |
| Actor: | Admin |
| Description: | Staff can be added by Admin. And also staff can be terminated by Admin. |

*Table 7: Description of add staff use case*

### 2.2.8. Terminate Contract:

| | |
|---|---|
| Use Case: | Terminate Contract |
| Actors: | Admin |
| Description: | Admin can terminate the staff either contract is finished or not. If contract is finish, in buffer time the staff can be de-register if they didn’t want to leave. |

*Table 8: Description of terminate contract use case*

### 2.2.9. Register a staff:

| | |
|---|---|
| Use Case: | Register a staff |
| Actors: | Admin, staffs (new) |
| Description: | Admin register a staff to run the hospital smoothly and to provide the necessary information to the users i.e. staffs, patient and also admin of the hospital who use this system. |

*Table 9: Description of register a staff use case*

### 2.2.10. Maintain record of Patient:

| | |
|---|---|
| Use Case: | Maintain record of Patient |
| Actors: | Staffs |
| Description: | The record of all patients, staffs and users of this system are maintain by the staffs to provide best facilities to them. If patient don’t visit the hospital or any offers comes then the system will provide flow-up remainder message to all patients. |

*Table 10: Description of maintain report of patient use case*

### 2.2.11. Generate report of Patient:

| | |
|---|---|
| Use Case: | Generate report of Patient |
| Actors: | Staffs |
| Description: | The final report of patients after their treatment is provided to patients. |

*Table 11: Description of Generate report of patient use case*

### 2.2.12. Register a patient:

| | |
|---|---|
| Use Case: | Register a Patient |
| Actors: | Staffs |
| Description: | After the conformation of appointment of patients, to add the details of patient's staffs registered patients in their system. |

*Table 12: Description of register a patient use case*

### 2.2.13. View Patient information:

| | |
|---|---|
| Use Case: | View Patients Information |
| Actors: | Staffs |
| Description: | Staffs can views the patient's information to give ideas or suggestions towards patient's health to patients. |

*Table 13: Description of view patient information use case*

### 2.2.14. Update record of Patient:

| | |
|---|---|
| Use Case: | Update record of patients |
| Actors: | Staffs |
| Description: | Staffs update the record of patients if patients updates their record. It helps to best communicate to patients. |

*Table 14: Description of update record of patient use case*

### 2.2.15. Setup Patient payment:

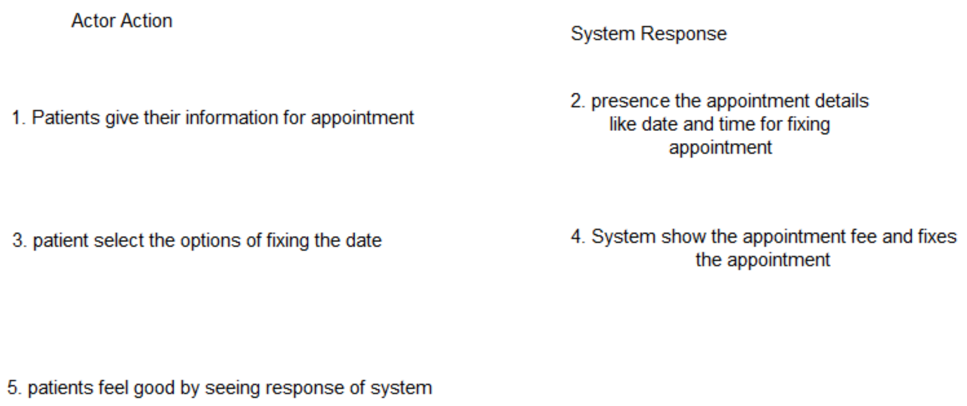
| | |
|---|---|
| Use Case: | Setup Patients payment |
| Actors: | Staffs |
| Description: | Staffs maintain the payment details of patients. It includes all the rate of all the test, medicine and appointment charge. |

*Table 15: Description of setup patient payment use case*

## 2.3. Expanded Use Case Descriptions:

Expanded use case definition is measures of two ways interaction. The communication between system and actors takes place in extended use case description. In this description, we write down each step for executing the use case. Here we will do expanded use case descriptions of just two usage cases, i.e. Book Appointment and Set Remainder The Expanded Use Case Descriptions are:

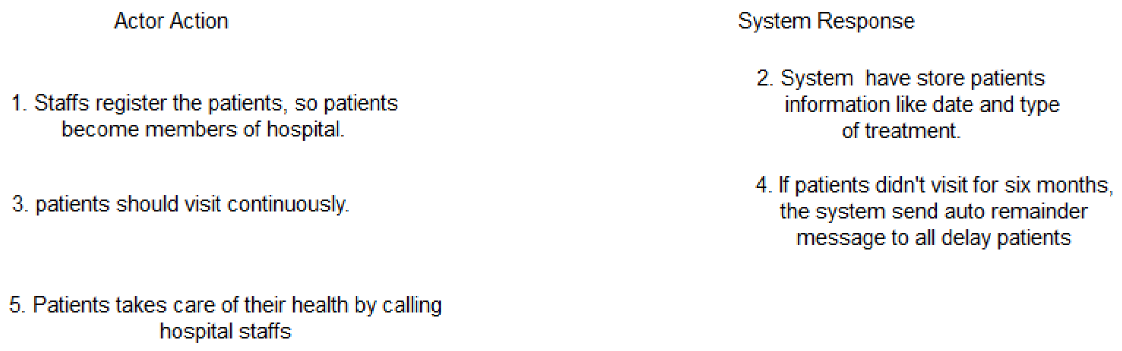
### 2.3.1. Book Appointment:



| Actor Action | System Response |
|---|---|
| 1. Patients give their information for appointment | 2. presence the appointment details like date and time for fixing appointment |
| 3. patient select the options of fixing the date | 4. System show the appointment fee and fixes the appointment |
| 5. patients feel good by seeing response of system | |

*Figure 2: description use case of book appointment use case*

### 2.3.2. Set remainder:



| Actor Action | System Response |
|---|---|
| 1. Staffs register the patients, so patients become members of hospital. | 2. System  have store patients information like date and type of treatment. |
| 3. patients should visit continuously. | 4. If patients didn't visit for six months, the system send auto remainder message to all delay patients |
| 5. Patients takes care of their health by calling hospital staffs | |

*Figure 3: description use case of set remainder use case*

The whole Use case Diagram of system is:

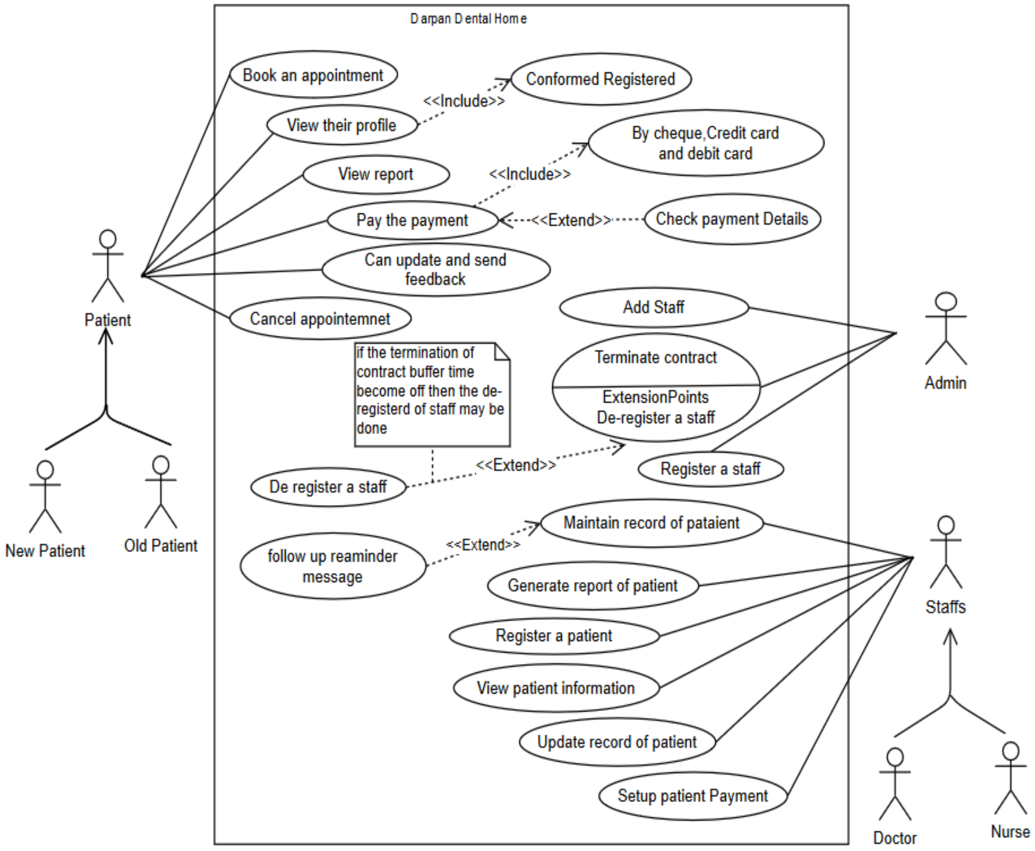


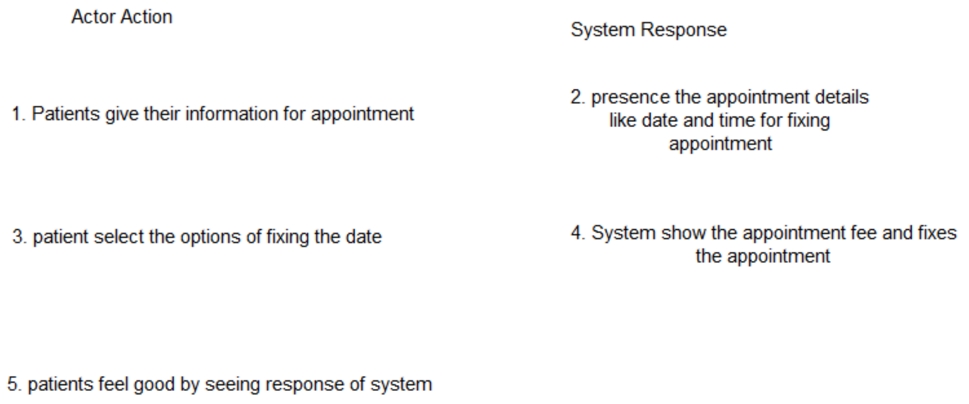
*Figure 4: Use case diagram of whole system*

# 3. Communication Diagram and Sequence Diagram:

## 3.1. Communication Diagram:

Also known as collaboration diagram, communication diagram. Collaboration diagram describes how objects interact. It displays the objects that participate in the interaction through their connections to each other and the message they send one to another. The design actually uses it to describe and explain the function of the object performing a specific flow of use case events. We are the main source of knowledge used for class accountability and system determination. Actually communication diagrams show the message flow between objects in an OO application and also imply the basic associations (relationships) between classes (Anon., n.d.).

As it is derived from case descriptions of high level use and from expanded case description of specific use case. Here we are going to create collaboration diagram of register a patient use case. The expanded use case description of Register a patients is:



| Actor Action | System Response |
|---|---|
| 1. Patients give their information for appointment | 2. presence the appointment details like date and time for fixing appointment |
| 3. patient select the options of fixing the date | 4. System show the appointment fee and fixes the appointment |
| 5. patients feel good by seeing response of system | |

*Figure 5: description use case of book appointment use case*

To draw communication diagram, there are many steps:

1. We need to find Domain Classes first. In order to find Domain classes, we need to find nouns from the definition and list those relevant for processing. Staffs, Patients, and Appointment may be the possible nouns of this use case.

2. We will draw an object symbol for each of the classes in the domain. The object symbol can be drawn in a box of rectangles followed by a column of groups of domains.

3. Control Object is used to manage object collaboration which gives effect to the Use case process.

4. To manage the Screen interaction for the use case is Boundary Object.
5. We need to draw an Actor interacting with the Boundary Object as an actor interacting with the system.
6. We have to add associations to give or show relation to each other between different objects.
7. And in last, we give message to the user interface.

The whole collaboration diagram for register a patient is given below:

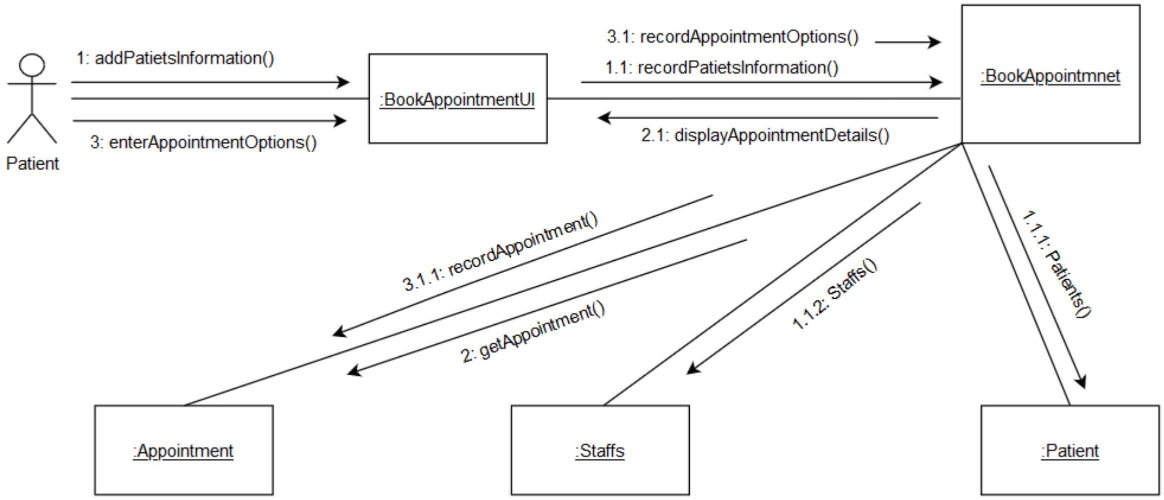

1: addPatietsInformation()
3: enterAppointmentOptions()
Patient
:BookAppointmentUI
3.1: recordAppointmentOptions()
1.1: recordPatietsInformation()
2.1: displayAppointmentDetails()
:BookAppointmnet
3.1.1: recordAppointment()
2: getAppointment()
1.1.2: Staffs()
1.1.1: Patients()
:Appointment
:Staffs
:Patient

*Figure 6: collaboration diagram of book appointment use case*

Here patients interact with Boundary Object by adding Patients information then the control object record patient Information. Control object get all the data from the Patients, staffs and Appointment class by which staff book an appointment and patients see the appointment date.

## 3.2. Sequence Diagram:

Sequence diagram is also known as Event Diagram. The UML diagram showing an interaction between objects arranged in a Diagram time sequence. It is a type of interaction diagram that shows details of how operations are performed as what message is sent and when, and it also draws case descriptions of the same high level use and case description of the same expanded use case from which we create collaboration diagram. This two diagram collaboration and sequence diagram are vice-versa means that they are connected to each other and one diagram can be obtained from another. Chain diagram is drawn in various levels of detail and in many phases of the creation life cycle, to satisfy specific objectives. Usually it uses to represent an interaction of object details that occurs for one use case or for one operation.

To draw Sequence diagram, there are many steps.

1. Dimension: Drawn in two dimensions, this diagram may be horizontal axis (Object Dimension) or may be vertically axis (Time Dimension). Sequence diagram is all about ordering not length, so we need to be careful about timing and ordering when drawing sequence diagram, but we don't just consider time period.
2. Actor: who or what is going to use our system is Actors.
3. Lifeline: The interaction represents an individual participant. They represent the various objects or parts that interact in the system during sequence with each other.
4. Activations: It is a thin rectangle box on a lifeline that represents the time that an object needs to complete a task. The top and the button of the rectangle box respectively represents the initiation time and completion time, if the activation box is longer than the task, and vice-versa.
5. Message: Specific communication between interaction lifelines means showing how objects interact with each other by sending and receiving messages. There

are many types of messages, some of which are: synchronous message, asynchronous message, message of return, etc.

6. Focus of Control: It defines times when the processing takes place inside the object during activation.
7. Control Object: It is used to manage object collaboration that gives effect to the use case process.
8. Boundary Object: It manages the interaction between the actor and the device and each case of usage implies at least one limit variable.
9. Guard: To have or build conditions, as if in programming situation, we use sequence diagram guards.
10. Alternative: in guard we represent when in diagram condition, whereas in sequence diagram alternative symbolizes a choice between two or more sequences of messages. It was represented by a large rectangle frame and specified inside the frame box by mentioning 'alt'.
11. Optional: It is similar to Guards where the fragment only executes when the condition supplied is true. Guard is used only for a message, whereas it is used as an option to model the process fragment and it is represented by a large rectangle frame and specified by mentioning 'Opt' inside the frame box, but if-else is used as an alternative, but optionally only if the same condition as Guard is present.
12. Parallel: If the condition fits, the message or fragment will be executed in parallel. It is represented as alternative and optional, but Par is used inside the frame box for notation and there is only one condition as an option and as an alternative many messages or fragments.
13. Loops: The fragment may be implemented several times and the diagram can be standard. Sequence diagram was improved by inserting fragment condition of loops and used to represent a repeated sequence and represented by the frame by mentioning 'Loop' within the frame.
14. Sequence Diagram: the entire sequence diagram is surrounded by the frame mentioning ‘SD:Use case name’ inside the frame.

The sequence diagram of Book Appointment is:

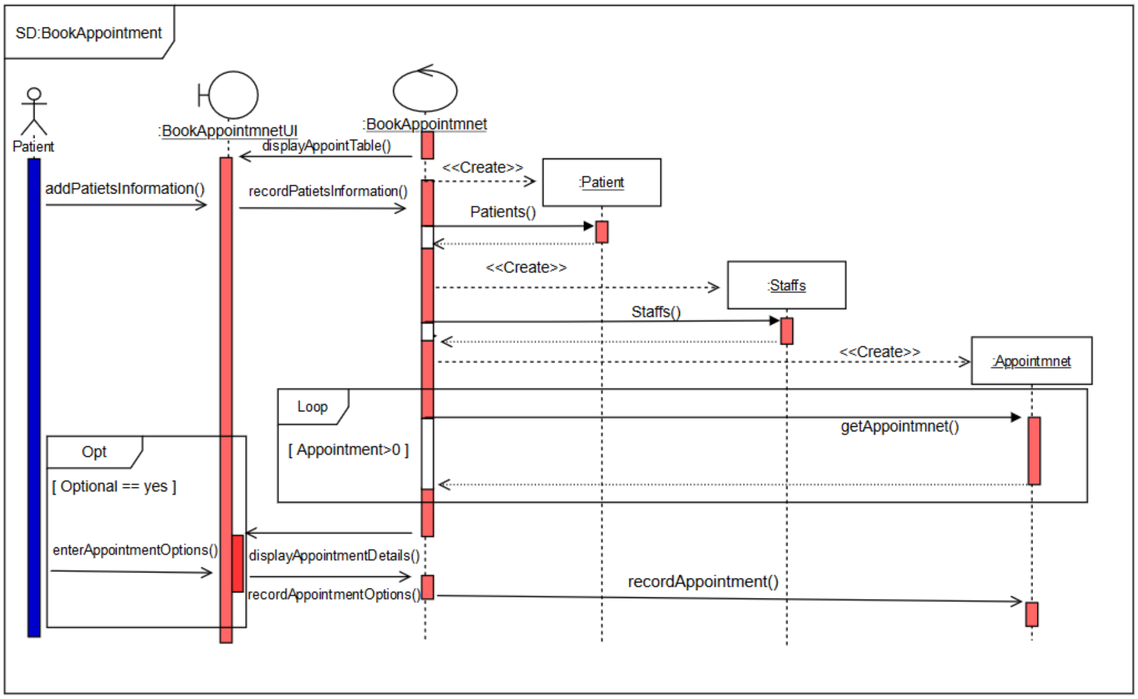


*Figure 7: sequence diagram of book appointment use case*

Here, in sequence diagram we use optional box because if the appointment details provided to patients by staffs means appointment is not proper for patients then the patients didn’t like appointment and they denied appointment and they become terminated from system, but option equal to yes, then patients see the appointment time which is fixed by staff again. And loop box is used because until the appointment is greater than 0 the appointment will occurs. In this we communicate in sequence diagram.

## 4. Class Diagram:

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects (Anon., n.d.). Class Diagram is an important UML diagram for software development that shows the classes of objects in the system and the relations between those classes. This is the key building block of the Oriented Modelling object used for general conceptual modelling and comprehensive modelling for computer code modelling. It is a static type structure diagram that describes a system structure b showing the classes of the system, the attributes and operations of each class and also the relationship between objects.

We can create class diagram by creating three rows in a table where we kept the name of the class, the attributes and the name or the operations of the methods. We can generate class diagram from an entire system's individual domain class. To construct a class diagram, we need to identify unique domain classes, draw the class diagram of and domain class, display various class diagram relationships and include inheritance relationships, association class and dependency relationships, association relationships, aggregation relationships, composition relationships.

Here we create class diagram by selecting unique domain classes from all use cases of DDH system i.e. Staffs, Patients, Admin, Hospital, Appointment, Report, Registration and Payment. Then we draw the different class diagram of domain class, and after that we start to show their relation. Between Staffs and patients there is association relation occurs in which we have made one association class also, Payment have also association relation between Staffs and Patients. Here Payment have association relation to Appointment, Composition relation to Report because report is fully depended on patients (if patients didn't come in hospital then report don't come) and again association relation to Registration. Here we have also created parents class with Admin and Staffs, which shows inheritance properties between them, staff have aggregation relation with Hospital because staffs can join another hospital even if this Hospital become close and they aren't depend on hospital. In this way we have drawn the class diagram of DDH system, which is given below:

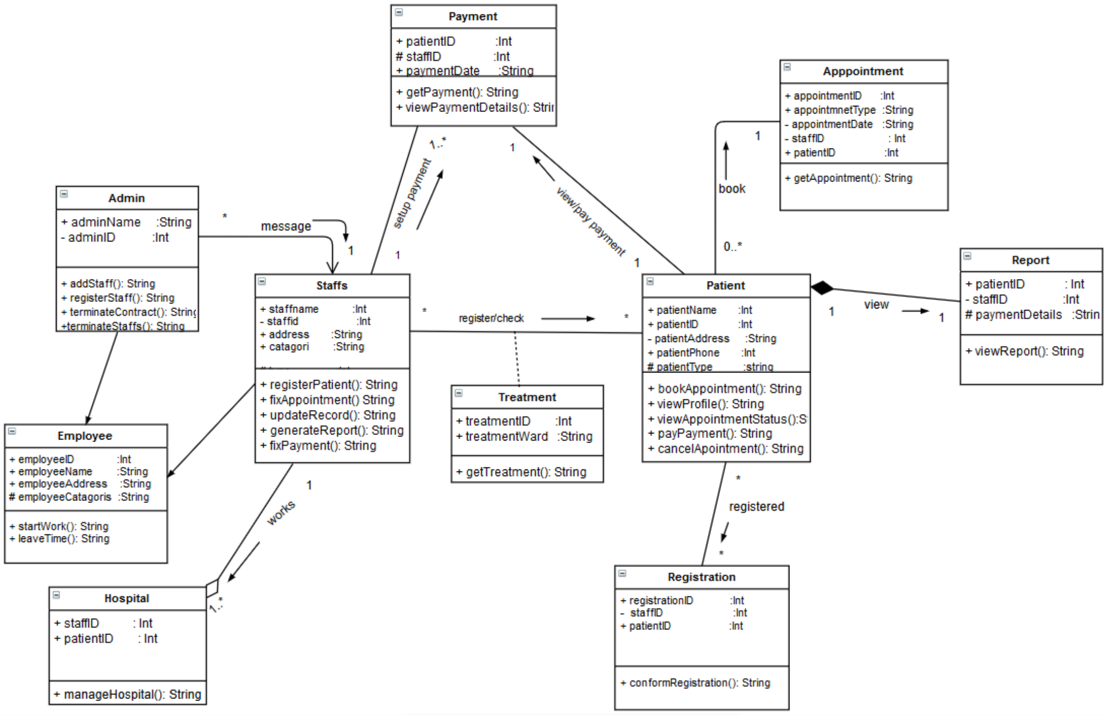


*Figure 8: class diagram for all DDH system*

## 5. Explain how you would move forward from this part of the development process. What techniques/product might be produce for the design? Explain briefly the purpose of each.

When we choose domain classes, object from each domain class in Elaboration phase in which we have created Use Case diagram, Collaboration diagram, Sequence diagram and Class diagram. When we are working in elaboration phase, the process of Inception phase is running back of elaboration phase for example risk management is running to decrease the probability of program failure, planning and prototype are also running in the back of elaboration phase. So we can't say directly that the system which is running in elaboration phase can't run in Inception phase. Also when the coding activity or modelling of different diagram are developing then the some of the task like testing and review activity is also doing. So their exit the working of all the process of all the phases.

As in different phase of RUP the different function is run, as:

Inception: It states the idea for the project. The development team determines whether the project is worth the effort and what resources are needed.

Elaboration: The architecture of the project and the required resources will be further assessed. Developers consider possible software applications, and the development-related costs.

Construction: The project is being developed and finalized. The software is designed, documented, and tested.

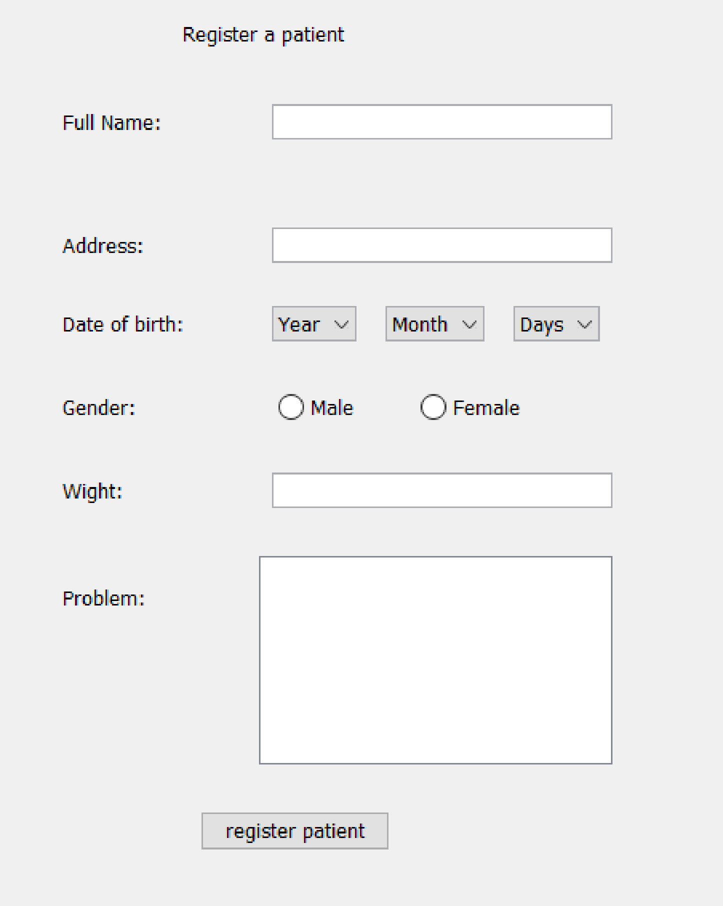
Transition: The software is distributed to the public. Final adjustments or updates shall be made based on end-user feedback (Anon., 2020).

As RUP is step by step mythology, in this way we can model the best software. At first we create the Use case diagram in which we mention different domain class and use cases. Then we create collaboration diagram through which we get relation between different use cases. After getting all, we again create class diagram which show the relation between all the use cases of system. In this way we create Gantt chart, how we work we can define in RUP. So RUP is very advance mythology of software development.

## 6. Prototype:

This cycle where develop to create an actual software model. The intention behind creating this model is to get user more deeply into the actual requirement. Prototyping helps a lot in getting the customer input. Software prototype process is: Identification of initial requirements, development of prototype, reviews and revision. The required prototype for DDT system is given below:

### 10.1. Register a patients:



Register a patient

Full Name:

Address:

Date of birth: Year Month Days

Gender: ○ Male ○ Female

Wight:

Problem:

register patient

*Figure 9: prototype of register a patient use case*

By this prototype, we can easily register a patients in easy way.

## 10.2. Generate report of patients:

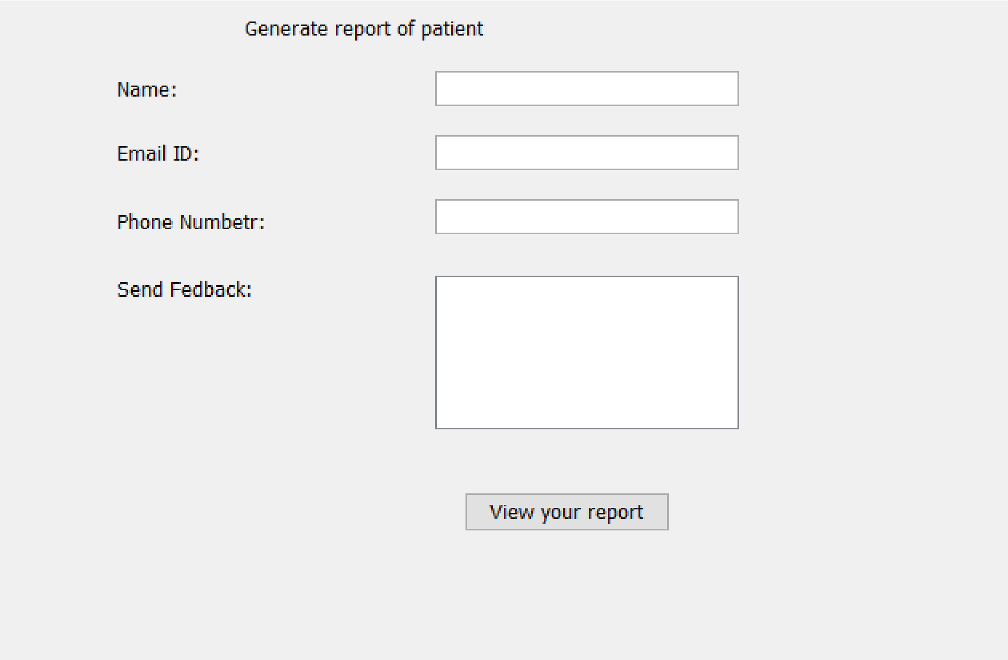
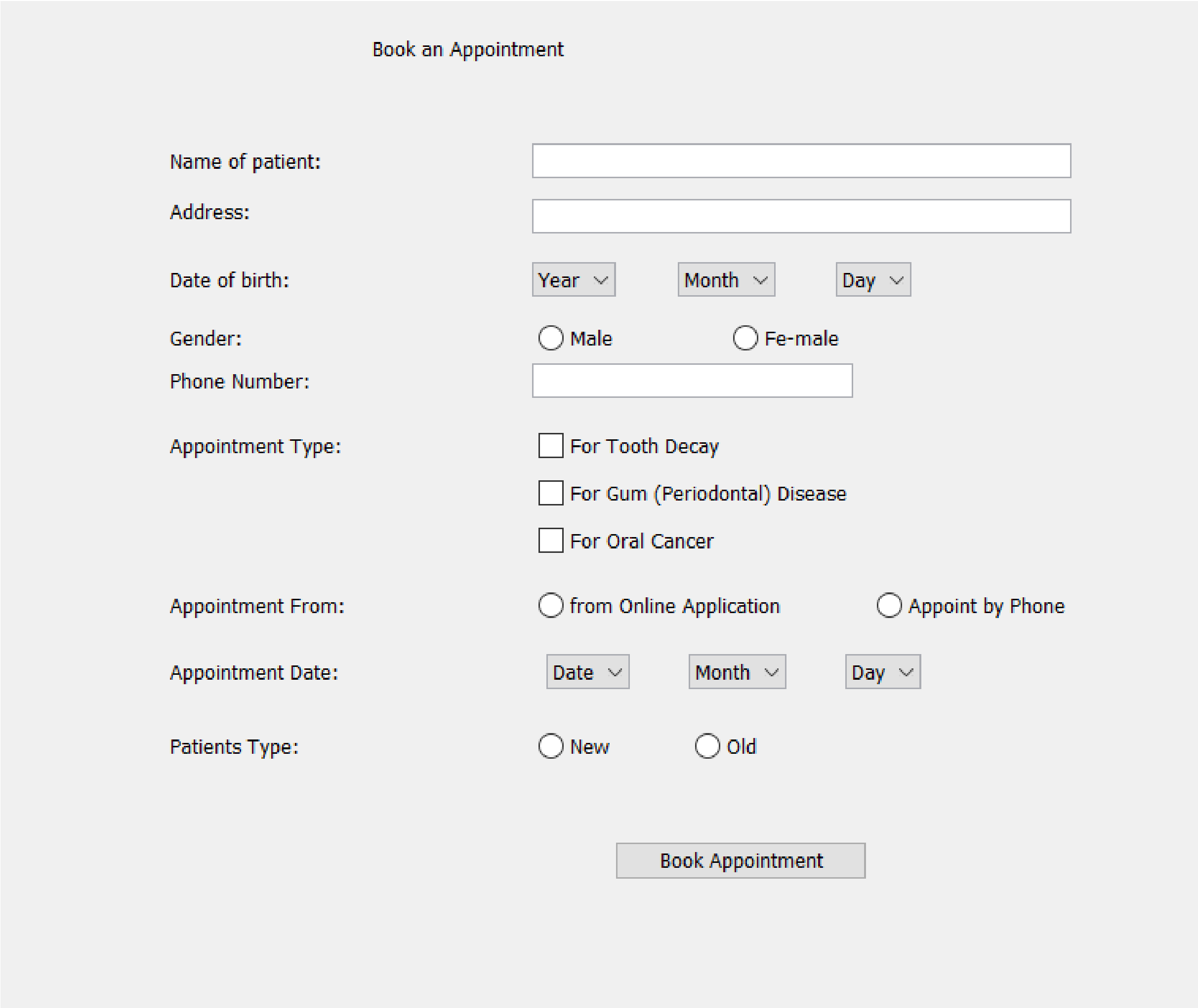


*Figure 10: prototype of generate report of a patient use case*

By this prototype, we can easily generate the report of patient in easy way.

## 10.3. Book Appointment:



Book an Appointment

Name of patient:

Address:

Date of birth: Year ⌄ Month ⌄ Day ⌄

Gender: ○ Male ○ Fe-male

Phone Number:

Appointment Type:
□ For Tooth Decay
□ For Gum (Periodontal) Disease
□ For Oral Cancer

Appointment From: ○ from Online Application ○ Appoint by Phone

Appointment Date: Date ⌄ Month ⌄ Day ⌄

Patients Type: ○ New ○ Old

*Figure 11: prototype of book appointment use case*

By this prototype, we can easily book an appointment for any office in easy way.

## 10.4. Payment of Patient:

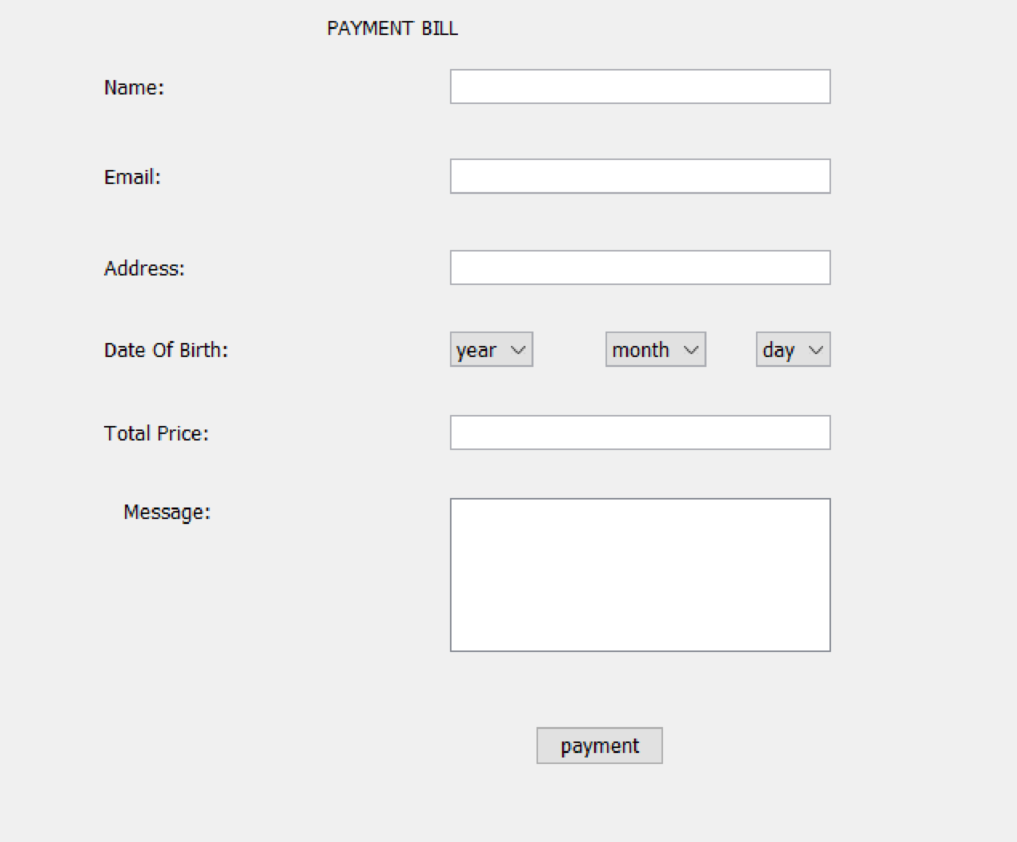
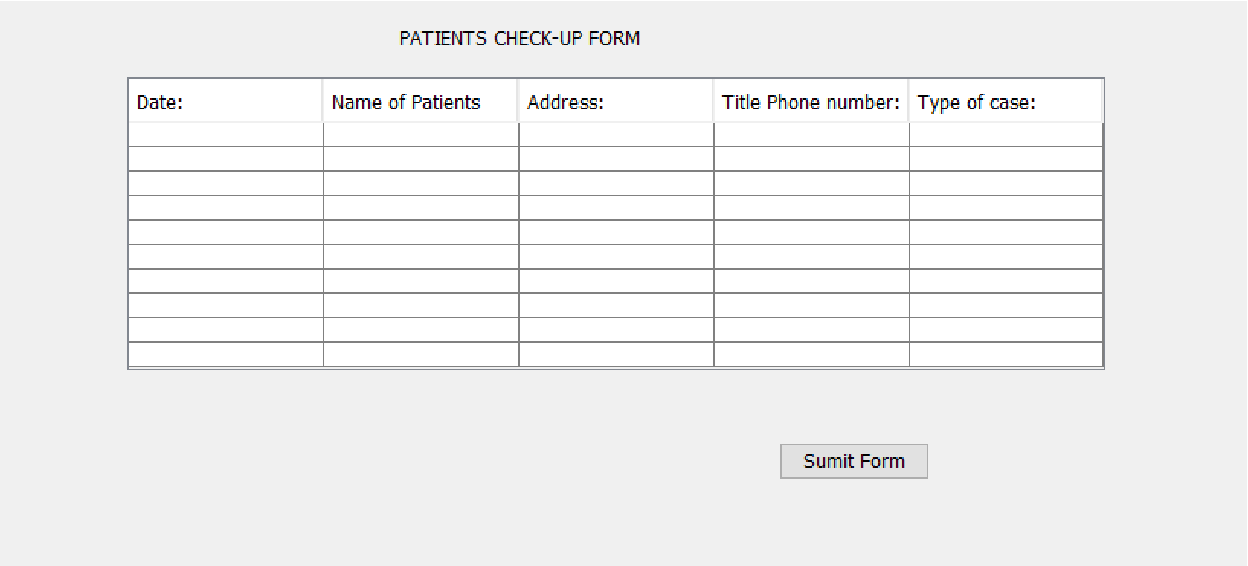


*Figure 12: prototype of payment of patient use case*

By this prototype, patients can easily payment their fee in easy way.

## 10.5. Records of patient check-up:



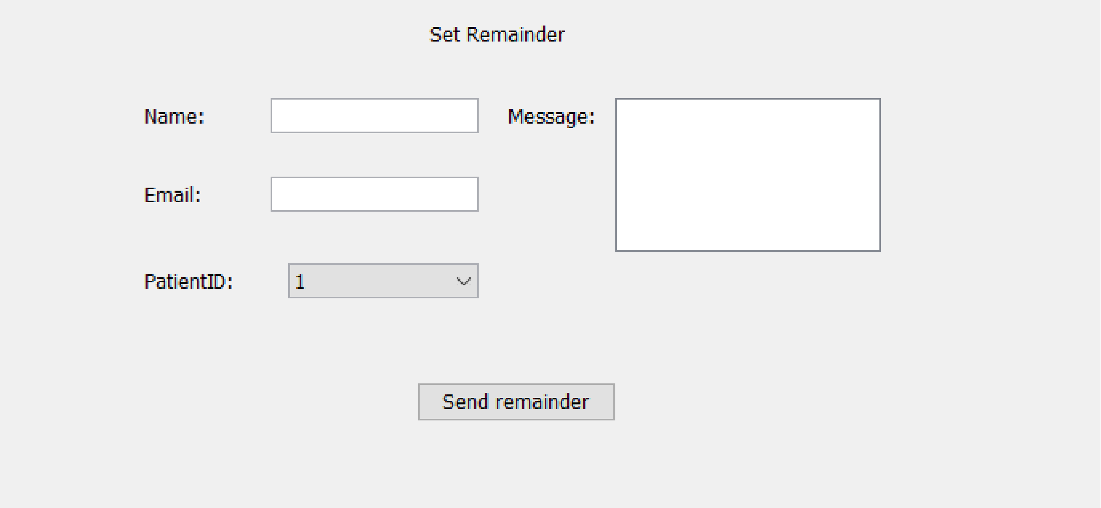
PATIENTS CHECK-UP FORM

| Date: | Name of Patients | Address: | Title Phone number: | Type of case: |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

*Figure 13: prototype of record of patient check-up use case*

By this prototype, we can easily check patient's record in easy way.

## 10.6. Set remainder:



Set Remainder

Name:

Email:

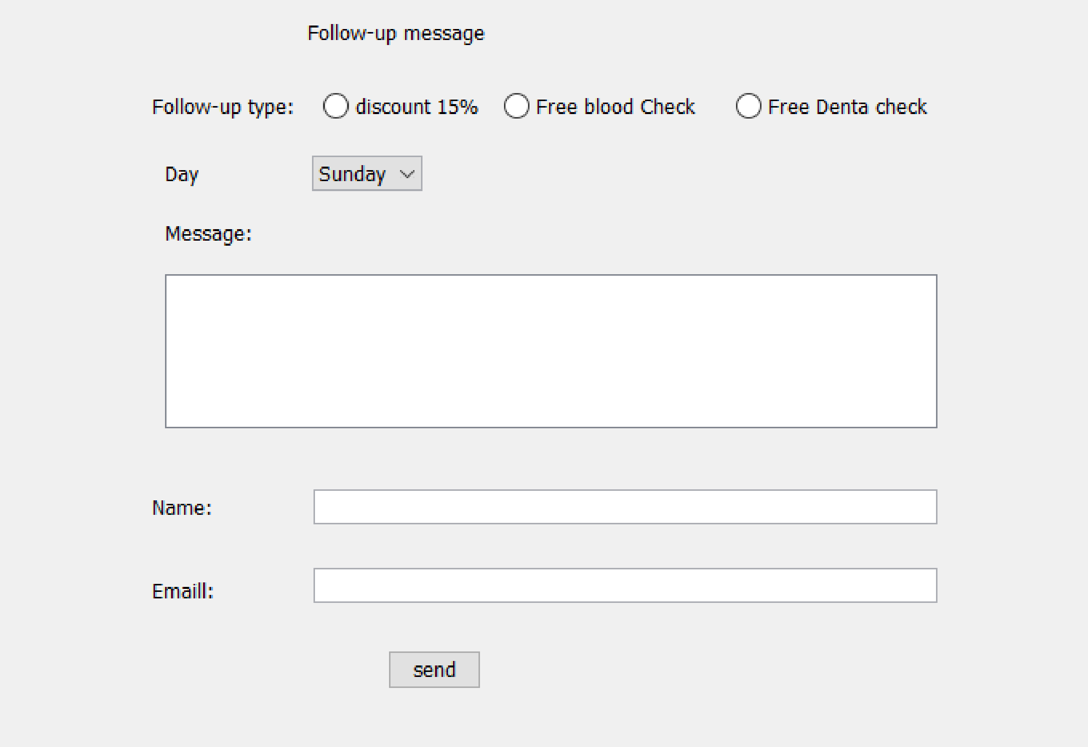
PatientID: 1

Message:

*Figure 14: prototype of set remainder use case*

By this prototype, we can easily set remainder to patients in easy way.

## 10.7. Follow up remainder:



Follow-up message

Follow-up type: ○ discount 15% ○ Free blood Check ○ Free Denta check

Day Sunday

Message:

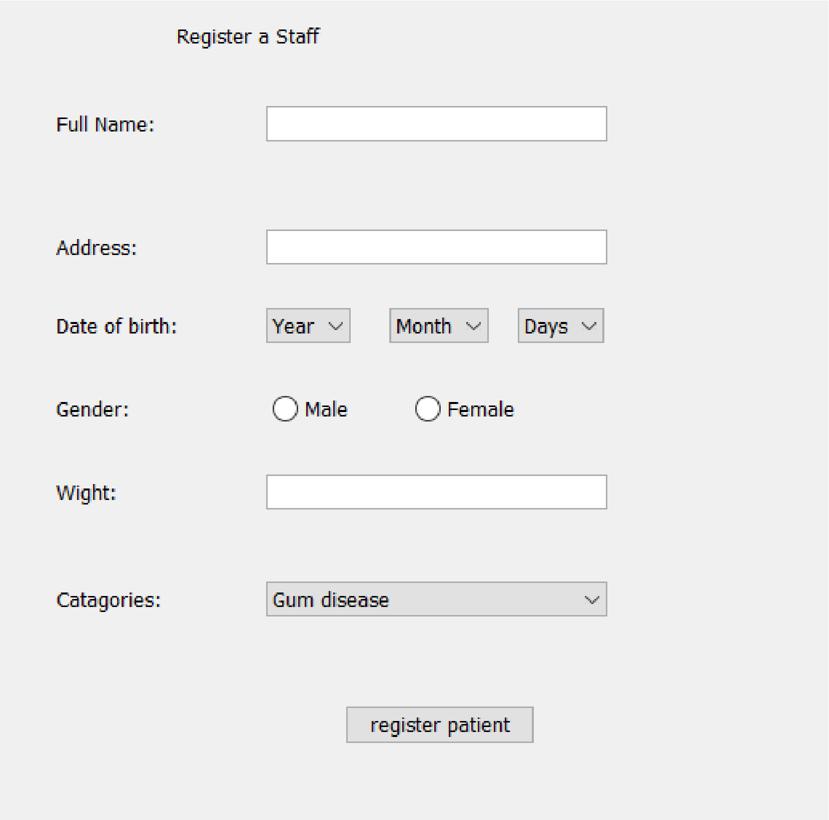
Name:

Emaill:

*Figure 15: prototype of follow up remainder use case*

By this prototype, we can easily follow-up remainder to patients in easy way.

## 10.8. Register a staff:



Register a Staff

Full Name:

Address:

Date of birth: Year Month Days

Gender: ○ Male ○ Female

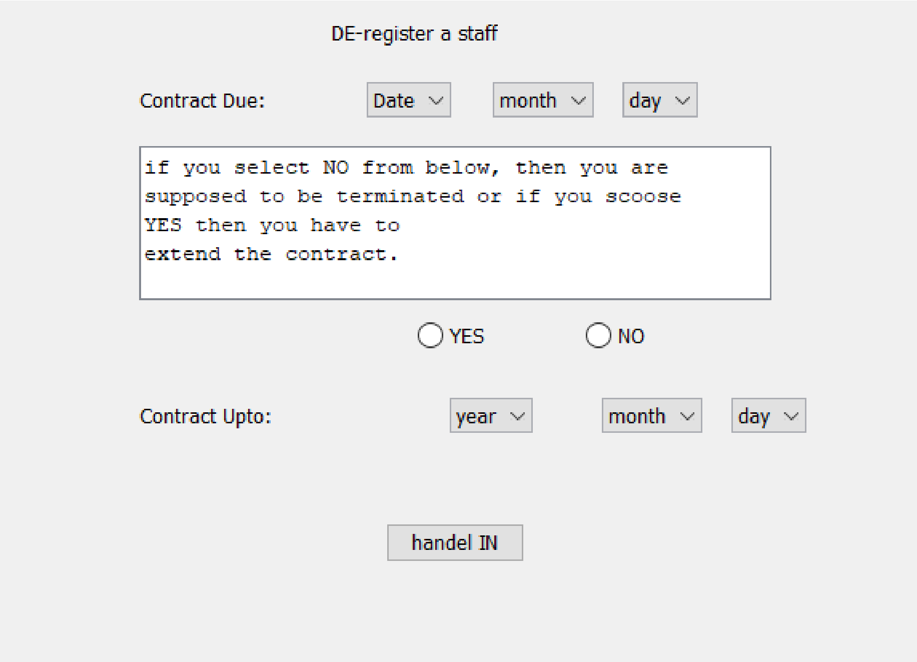
Wight:

Catagories: Gum disease

*Figure 16: prototype of register a staff use case*

By this prototype, we can easily register a staff in easy way.

## 10.9. De-register a staff:



if you select NO from below, then you are
supposed to be terminated or if you scoose
YES then you have to
extend the contract.

*Figure 17: prototype of de-register a staff use case*

By this prototype, we can easily de-register a staff in easy way.

## Conclusion:

As a Software Engineer, we have to create a computerized information system to maintain record of customer and their payment details for Dental Clinic named as Darpan Dental Home which are currently facing lots of problem in maintaining the record of their patients by using the Unified Software Development Process (USDP) / Rational Unified Process (RUP) with the Unified Modelling language (UML). By making the software from RUP principle, the developer as well as Clint might face the different problem for starting the development process. Here we can do every task simultaneously so that the system becomes complex at first, but day by day the system are becoming more interactive toward user.

This system used to provide object relation between use cases, but we fell very hard to draw the different diagram. By looking the coursework from upper side, we feel good to create, but after starting the coursework their arises many problems. but, but by taking help from teachers and friends and doing many research in books, internet and reading lecture and workshop slides I was able to create our scenario into many steps, which helps me to complete my coursework. After all a lots of challenges and difficulties were faced and many effort were needed to complete this coursework. As this module very importance for our futures, we give our best to complete good coursework even we fell hard in lockdown due to internet problem. Due to these all problem, we complete our coursework and summit within deadline.

## Bibliography and Reference:

### Bibliography

Anon., 2020. *RUP (Rational Unified Process) Definition.* [Online]
Available at: https://techterms.com/definition/rup
[Accessed 04 06 2020].
Anon., n.d. *UML 2 Communication Diagrams: An Agile Introduction.* [Online]
Available at: www.agilemodeling.com/artifacts/communicationDiagram.htm
[Accessed 04 06 2020].
Anon., n.d. *What is a Gantt Chart? Gantt Chart Software, Information ....* [Online]
Available at: https://www.gantt.com
[Accessed 04 06 2020].
Anon., n.d. *What is Class Diagram? - Visual Paradigm for UML.* [Online]
Available at: https://www.visual-paradigm.com/guide/uml-unified...
[Accessed 04 06 2020].