Analysis and Design Project Documentation

Simplified Patient Care Clinic System (PCCS)

INFO 620: INFORMATION SYSTEMS ANALYSIS & DESIGNWINTER 2020

Shahrar Nizam, Sophearith Rin, Sm Kabir, Bledar Noka

Table of Contents

INTRODUCTION

The purpose of this project is to produce a specification of analysis and design for Patient Care Clinic, based on UML and other non-UML specifications techniques. A local clinic has requested to create a simple system through which can be able to keep track of patients, appointments, generate reports, invoices and medical histories such symptoms and treatments. This project will assume that the clinic is operated by a few members of staff such as doctors, nurses, receptionists, admin and will provide minimum functionalities for keeping track of everyday tasks. The project is divided in four chapters with each chapter including sections and subsections.

The first chapter will be focused on the system analysis which will include definition of the scope of the project, the functional and non-functional requirements of the system, the use case model of the system and its descriptions along with the class model. The use case model will provide the blueprints of how the system will work and what functions each user (actor) will have access through the help of "stories" requirements document. Use case descriptions will follow the use case model which will help understand most important primary uses cases, their extend and included use cases. Another important subsection of chapter one will include class model and its definitions. Class model will be conducted based on the TCM model to show the analysis process our group followed.

At the end of chapter one there will be a discussion section which will show how each decision was taken step by step to conduct these uses cases and important clarifications will be provided.
The second chapter of the project will provide the system design aspect of the system which will include sequence diagrams for the chosen use case along with validations and discussions of the diagrams. Class diagrams will be completed in this chapter showing all the attributes, operations and navigability along with the validation and discussion section for the diagram.
The third chapter of the project will focus on the physical design of the system which will show the Entity-Relationship Diagram and the relational database schema. Our group has yet decided whether at this stage of the project we will add component diagrams and/or deployment diagrams.

The fourth chapter will be a summary of the entire project by evaluating the process our group chose to complete this project. This includes all the decisions that our group made throughout the entire project such as: splitting individual work, challenges faced during the project, time-management and group meetings. It will also include a summary of modeling tools our group used to complete this project.

CHAPTER 1: System Analysis

(1) Problem Statement

A small local clinic would like to create a simplified patient care clinic system (PCCS) for the operation of the clinic.

(a) Context and Importance of the system

It is important for a healthcare provider like a small local clinic, to be able to record their patients and staff details. The ability for clinic to track patient's information is essential to keep up with their conditions and medical histories such as symptoms, treatments and other clinical services. The clinic should be able to schedule appointments for their patients and generate invoices for each visit. Doctors should be able to retrieve patient's medical records for proper diagnosis. It is also important for the clinic to be able to keep track of the payment specifically handling the payment with their health insurance.

(b) Overall goals of the system

The overall goals of the system are to record new patients, staff, schedule appointments, keep track of patient's condition, medical histories, generate reports, invoices and payments.

(c) Scope of the project

IN-Scope:

PCCS will include patients, staff, appointments, medical records and invoice management systems.

OUT-Scope:

PCCS will not be dealing with the process of claiming and contacting insurance company. It will not include fixed and non-fixed costs such as rent, office-related expenses, hardware expenses such as monitors, and computers used by the staff of the clinic. It will not include staff salaries and other tax-related costs.

(2) Requirements

2.1 Functional Requirements

1. Manage patient information
    a. Add, update, view and delete patients.
2. Manage staff information (Doctors, Nurses and Receptionists)
    a. Add, update, view and delete staff.
3. Book patient appointments
    a. Create, update, view and cancel appointments.
    b. Keep track of show and no-show appointments.
    c. View doctor's availability
4. Manage patient's visit information
    a. Create, update, view and delete patient's visit info.
    b. Record patient's medical conditions and history.
    c. Record patient's vital signs.
    d. Record patient's tests, screenings and shots.
    e. Record patient's symptoms, diagnosis and treatment.
    f. Record patient's prescription and references.
    g. Attach external reports to visit info.
5. Generate patient's medical history report
    a. Report must include all past visit information.
    b. Report should be generated based on applied filters.
6. Manage clinic services
    a. Add, update, view and delete clinic services.
    b. Update service fees.
7. Manage visit invoice
    a. Create, view and delete invoice.
    b. System must generate fees based on visit services.
    c. System must perform total calculation.
8. Print patient visit report
    a. Report must include visit info.
    b. Report must include a copy of the invoice.
9. Provide separate login for doctors, nurses, admin and receptionist
    a. Every staff member should have different access based on role.
    b. Admin must have all access.

2.2 Data Requirements

1. Patients: Each patient record must contain patient ID, name, date of birth, gender, social security, address, contact info, registration date and notes.
2. Staff: Each staff record must contain staff id, name, date of birth, gender, contact info, joining date, active status, staff type and notes.
3. Appointments: Each appointment record should contain appointment id, name, staff id, contact info, show-up status, date of booking, date and time of appointment.
4. Visits: Each visit record should contain a visit id, patient id, appointment id, staff id, visit time, medical conditions, medical history, vital signs, symptoms, diagnosis, tests, screenings, shots, treatment, prescription and references.
5. Clinic Services: Each service record should contain a service id, name and fee.
6. Invoice: Each invoice record should contain invoice id, visit id, patient id, staff id, social security, service fees, tax and total.

2.3 Business Rules and Logic

1. Staff should not be able to register, update and delete staff.
2. Staff should not be able to delete patients, medical conditions and visit records.
3. Patients cannot see a doctor without booking an appointment.
4. Appointments with no-show status should not have a connected visit.
5. If a patient is more than 10 minutes late, they will have to book a new appointment.
6. Two appointments cannot be scheduled at the same time for one doctor
7. Appointments can only be scheduled for working hours
8. No visit should not be more than 40 minutes.
9. The Clinic's base fee is applied to every visit. Additional fees are added based on services such as tests, screenings and shots.
10. Only doctors and nurses can manage visit information.
11. Only doctors and nurses can generate medical history reports.
12. Insurance is able to recognize a patient based on social security.
13. Invoices cannot be updated once created. Any updates should require new invoice creation.

2.4 Non-Functional Requirements

1. Performance:
    a. All requests must respond in less than 1 second.
    b. System must launch in less than 10 seconds.
    c. Search results must not exceed more than 20 records per page.
2. Availability:
    a. System must be available 365 days a year with minimal downtime (0.15% downtime and 99.85% uptime).
    b. Must be accessible from the clinic only and based on the personnel credentials.
    c. Forms must time out without saving if the user is inactive for more than 15 minutes.
3. Scalability:
    a. System must not slow down when all the users are online at the same time.
    b. System must not slow down with more users and data.
4. Security:
    a. System must correctly identify and authenticate users at login.
    b. Data must be protected from external attacks.
    c. Deletion must remove all data completely from the system.
    d. Users must reset their password every 90 days.
    e. Users must be locked out if a wrong password is entered more than 5 times.
    f. Patient's medical records should only be visible to doctors and nurses.
5. Maintainability:
    a. System must be flexible to updates and changes.
    b. Admin must be able to debug errors in the system and perform routine maintenance.
    c. System should be well documented.
6. Usability:
    a. System must be a web-based application accessible through any web browser compatible with Windows and Mac OS.
    b. Interface must be simple and allow basic functionalities to be learned in one day.
    c. Icons without texts must display a description when pointed at.
    d. Errors must provide proper explanation and solution.
7. Reliability:
    a. System must back-up every 1 hour.
    b. Disaster recovery must complete within 1 hour.

c. There must be more than one back-up system at all times.
d. More than one person should not be able to edit a record at the same time.
e. Data redundancy must be removed.
f. Data must be consistent system-wide.

2.5 Other Important Assumptions

1. There is only one insurance company and insurance is claimed by emailing them the invoice.
2. Working hours of the clinic is between 9am-6pm Monday-Friday.
3. All doctors are able to take appointments during working hours.
4. The system will serve only one clinic with 3-4 doctors, 6-7 nurses and 2-3 receptionists and 1 system administrator.
5. The clinic is not an urgent care but a primary care with limited services.
6. The system is a 3-tier web based application.
7. The system will be used by users with basic computer proficiency.
8. The internet connection in the clinic is at least 10mbps, stable and available 24/7.
9. Staff will manipulate patient's data with honest intentions at the clinic's best interest.
10. Old medical records are not transferred into the new system.

(3) Use Case Model

3.1 "Stories" Requirements Document

a. As a receptionist I would like to manage patient's personal records.
b. As a receptionist I would like to schedule appointments.
c. As a receptionist I would like to confirm an appointment when a patient shows up.
d. As a receptionist I would like to cancel appointments upon the patient's or doctor's request.
e. As a receptionist I would like to create invoices to bill patients.
f. As a receptionist I would like to give patients a report about their visit.
g. As a medical professional I would like to record information about the patient's visit.
h. As a medical professional I would like to assign different clinical services to the patient.
i. As a medical professional I would like to include any external medical documents to the patient's visit information.
j. As a medical professional I would like to see patient's past medical history.
k. As a medical professional I would like to give patients a report about their visit.
l. As an admin I would like to manage staff's personal information.
m. As an admin I would like to keep a record of all the clinical services and their prices.
n. As an admin I would like to manage all the user access.

o. As an admin I would like to help receptionists process patient's personal records.
p. As an admin I would like to help medical professionals process patient's visit information.
q. As an admin I would like to help receptionists process invoices.

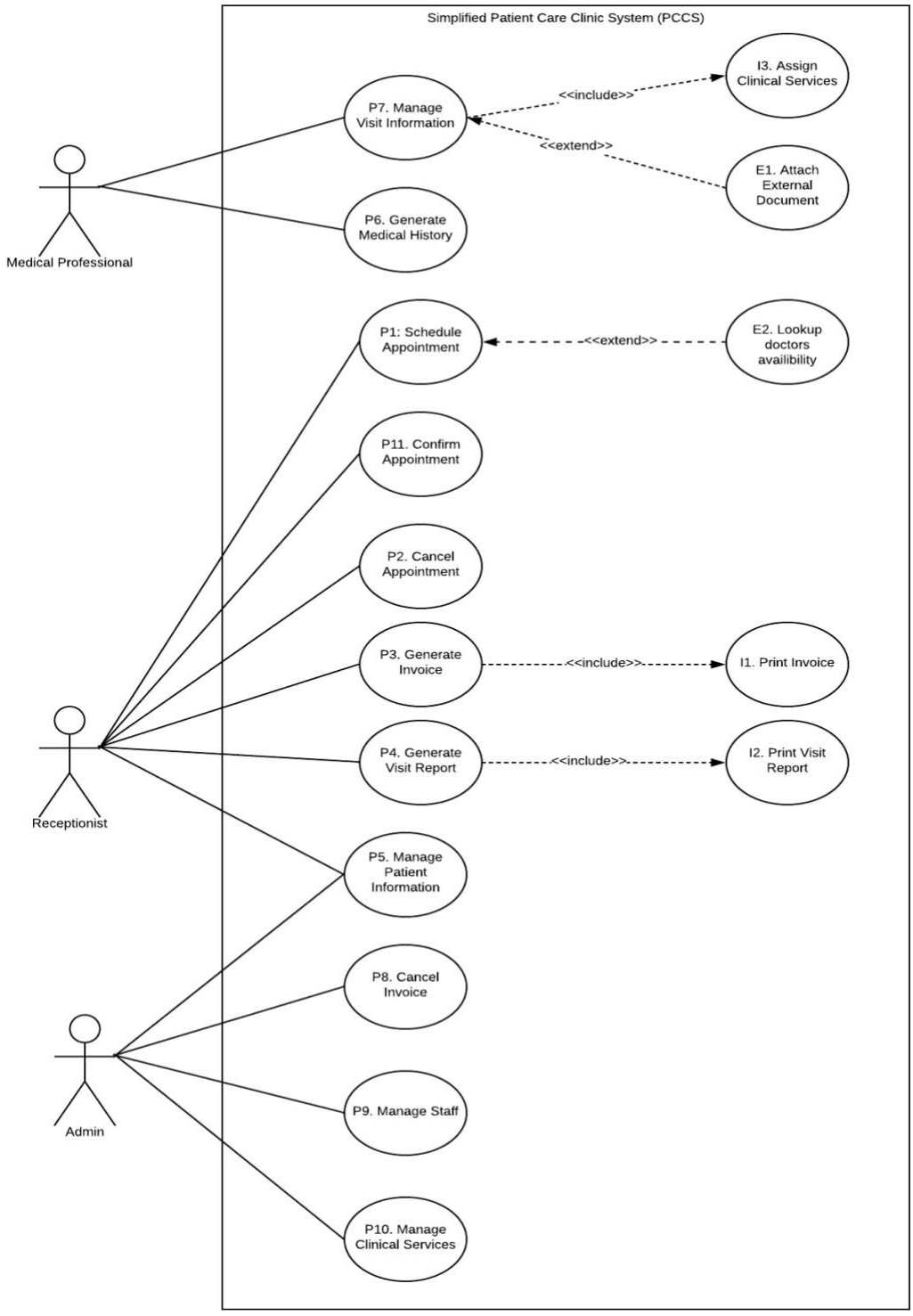
3.2 Actors and Their Goals

Receptionist (primary actor) – A receptionist is a staff member who performs daily administrative tasks for the clinic.

Medical Professional (primary actor) – A medical professional can be a doctor or a nurse. Their goal is to diagnose patients' and provide treatments.

Admin (secondary actor) –An admin helps maintain the clinical system and supports staff members to complete their tasks in the system.

3.3 Use case diagram



3.4 Use Case Overview

For our system we have recognized 16 use cases and 3 primary actors. When a patient call to book an appointment, the receptionist schedules the appointment. They can choose to lookup the doctor's availability to schedule the appointment. When a patient shows up for the scheduled appointment, the receptionist will confirm the appointment in order to track show status as per requirement. The receptionist can also cancel appointments upon doctor's or patient's request. A medical professional is a doctor or a nurse. Once the patient goes in to see the doctor, the doctor can manage the patient's visit information which involves tasks such as entering, updating or deleting a patient's vital signs, medical conditions, medical history, symptoms, diagnosis, treatments, tests, prescriptions and references. At the end of the visit, the medical professional will have to assign the clinical services (such as clinic fee, test fee, screening fee) that were used by the patient for billing purposes. They can also attach external documents (such as reports from an external hospital) that the patient brings to the patient's visit information. The medical professionals can also generate a patient's medical history which is a report containing all of the patient's previous visits. Once the visit is over, the receptionist can generate an invoice to be emailed to the insurance company and to provide a hard copy to the patient. They can also generate a visit report so that the patient can take home a printed report with details about their visit. The receptionist can also add, update or delete a patient from the system. All of these are part of the manage patient information use case. The admin also has access to managing patient's information. The admin is the only one that can cancel an invoice if an invoice was generated by mistake by the receptionist. The admin is the only one that can manage staff's information such as adding, updating or deleting a staff member from the system. They can also add, update or delete clinical services upon the clinic's request.

3.5 Use case descriptions

USE CASE DESCRIPTION		
Use Case #	P1	
Use Case Name	Schedule Appointment	
Actor	Receptionist	
Goal (1 phrase)	To schedule appointments for the patients to see a doctor	
Overview and scope	When a patient calls to book an appointment with a doctor, the receptionist will enter an appointment with a name, phone number and email address for a particular doctor at a specific date and time. Looking up a doctor's schedule or canceling the appointment is outside of this scope.	
Level	Primary	
Preconditions	The requested doctor must be registered in the system. The doctor must be available for the requested date and time of the appointment The patient should not already have a scheduled appointment	
Postconditions in words (write in passive and past tense)	An appointment has been scheduled for the patient to come and see a doctor.	
Trigger	Patient calls to book an appointment	
Included Use Cases	Lookup doctor’s schedule	
Extending Use Cases		
MAIN SUCCESSFUL	***Actor Action***	***System Action***

SCENARIO in numbered sequence	1. Receptionist logs in	2. System provides access
	3. Receptionist opens appointment scheduler	4. System displays all appointments
	5. Receptionist requests to add new appointment	5. System provides appointment form
	6. Receptionist Enters patient's name, email, phone, appointment date, time and appointed doctor	
	7. Receptionist schedules appointment	8. System prompts appointment scheduled
OTHER SUCCESSFUL SCENARIOS	***Step***	***Branching Action***
UNSUCCESSFUL SCENARIOS	***Conditions***	***Actions***
	1b. Receptionist provides wrong username or password	2b. System denies access
	7b. Receptionist does not enter one of the required information	8b. System prompts required fields must be complete to proceed
	7c. Receptionist entered a time outside clinic's working hours	8c. System prompts to enter a valid time.
	7d. Receptionist entered a time when the doctor already has an appointment	8d. System prompts appointment collides with another appointment
Priority in scheduling	1st Priority	
Frequency	Will be performed every time a patient wants to see a doctor	
Business rules and data logic	Multiple appointments cannot be scheduled for the same doctor at the same date and time All fields must be completed to schedule an appointment	

	An appointment must be scheduled to see only one doctor
Other non-functional requirements	System must respond in less than 1 second and not keep the patient waiting. Appointment information must be backed up.
Superordinates	
Developer	Shahrar Nizam
Creation date and last modified date	Created on 02/19/20

USE CASE DESCRIPTION	
Use Case #	I5
Use Case Name	Lookup doctor's schedule
Actor	Receptionist
Goal (1 phrase)	To see the doctors availability for an appointment.
Overview and scope	When a patient requests to book an appointment with a specific doctor, the receptionist can view the doctor's next available date and time before scheduling the appointment.
Level	Extended
Preconditions	The requested doctor must be registered in the system
Postconditions in words (write in passive and past tense)	The receptionist knows when the doctor is next available for an appointment or if the doctor is available at the patient's requested date and time.
Trigger	Receptionist would like to know about doctor's availability

Included Use Cases		
Extending Use Cases		
MAIN SUCCESSFUL SCENARIO in numbered sequence	***Actor Action***	***System Action***
	1. Receptionist opens the appointment scheduler	2. System displays all appointments
	3. Receptionist request's doctor's availability	4. System provides list of doctors
	5. Receptionist selects required doctor	6. System displays all appointments for required doctor
OTHER SUCCESSFUL SCENARIOS	***Step***	***Branching Action***
UNSUCCESSFUL SCENARIOS	***Conditions***	***Actions***
	5b. Required doctor does not exist	6b. Request admin to add doctor to the system.
Priority in scheduling	2nd priority	
Frequency	Whenever a receptionist needs to know a doctors availability for scheduling appointments	
Business rules and data logic	Schedule will not be available if doctor is not in the system System must display all appointments with show and no-show status.	
Other non-functional requirements	System must perform a search in less than 1 second and not keep the patient waiting on the phone.	

Superordinates	Schedule Appointment
Developer	Shahrar Nizam
Creation date and last modified date	Created on 03/03/20

USE CASE DESCRIPTION	
Use Case #	P7
Use Case Name	Manage Visit Information
Actor	Medical Professional
Goal (1 phrase)	To record all procedures and diagnosis during the patient's visit.
Overview and scope	When a patient comes in, the doctor will record all symptoms. They will then suggest a procedure that might resolve the issues and write them in the visit notes. Whether the procedure works or not will also be recorded.
Level	Primary
Preconditions	
Postconditions in words (write in passive and past tense)	Once the patient arrives the doctor will write down everything that happened during that visit. A visit object is created with patient symptoms, medical history, diagnosis, possible treatments and any prescribed medications.
Trigger	Patient arrives for their appointment
Included Use Cases	Assign Clinical services
Extending Use Cases	Attach external documents

MAIN SUCCESSFUL SCENARIO in numbered sequence	***Actor Action***	***System Action***
	1. Medical professional logs into the system	2. System provides access
	3. Doctor searched for patient in the system	4. System displays all of patient's information
	5. Doctor starts a new visit report for that date.	5. System provides empty form for the doctor to fill out as the appointment goes on
	6. Doctor enters the patient's symptoms and all procedures performed and any medications they need.	7. System saves documents for future reference.
OTHER SUCCESSFUL SCENARIOS	***Step***	***Branching Action***
UNSUCCESSFUL SCENARIOS	***Conditions***	***Actions***
	1b. Doctor provides wrong username or password	2b. System denies access
	6b. Doctor cannot find the patients in the system	7b. System prompts unable to find patient.
Priority in scheduling	1st Priority	
Frequency	Will be performed every time a doctor will see a patient	
Business rules and data logic	Only one patient can be seen at a time. The doctor can fill in as much or as little as they see fit into the report. A patient does not need a prior history to be seen. They can start their medical history for the first time with the doctor.	
Other non-functional requirements	System must respond in less than 1 second and not keep the doctor waiting. Patient information must be backed up. Patient information must be kept secure and only the patient and the doctor may have access to the information in order to	

	avoid HIPAA violations.
Superordinates	
Developer	Sm Kabir
Creation date and last modified date	Created on 02/19/20

USE CASE DESCRIPTION	
Use Case #	I3
Use Case Name	Assign Clinical Services
Actor	Medical Professional
Goal (1 phrase)	To Assign Clinical Services
Overview and scope	When a doctor assigns specific services to a patient in hopes of resolving the patient's illness. The doctor may assign multiple services in order to get more information about the patient and their symptoms or to even cure them. Actually performing the procedure is out of the scope.
Level	Include
Preconditions	The patient must be seen by a medical professional first and have permission from their doctor to have the services.
Postconditions in words (write in passive and past tense)	The patient had the services done by a medical professional and the results were inputted into the visit report.
Trigger	The Need for more information on the patient's condition or the service will resolve their issues.
Included Use Cases	

Extending Use Cases		
MAIN SUCCESSFUL SCENARIO in numbered sequence	***Actor Action***	***System Action***
	1. Doctor opens list of medical services offered	2. System provides access
	3. Doctor schedules the patient for all necessary services needed for the patient	4. System displays patient's scheduled services information and results.
OTHER SUCCESSFUL SCENARIOS	***Step***	***Branching Action***
	1. Procedure shows the problem.	2. No further services are needed.
UNSUCCESSFUL SCENARIOS	***Conditions***	***Actions***
	1b. Doctor request a service	2b. The requested service is unavailable at the time.
Priority in scheduling	3rd Priority	
Frequency	Everytime the doctor needs more information on the patient or their symptoms.	
Business rules and data logic	Medical services will be assigned when the doctor needs more information on a patient. The data must be human readable and placed into the patient records and history.	
Other non-functional requirements	System must respond in less than 1 second and not keep the doctor waiting. Patient information must be backed up. Patient information must be kept secure and only the patient and the doctor may have access to the information in order to avoid HIPAA violations.	
Superordinates	Manage Visit Information	

Developer	SM Kabir
Creation date and last modified date	Created on 02/19/20

USE CASE DESCRIPTION		
Use Case #	E1	
Use Case Name	Attach External Documents	
Actor	Medical Professional	
Goal (1 phrase)	Upload any outside documents that the patient brings in that may be important to the patient's medical history.	
Overview and scope	The patient may bring in documents from previous doctors that may be important for us to know. The doctor can scan in the document and upload it to the patient's medical history.	
Level	Extended	
Preconditions	The patient must have a valid and necessary document about their health in order to be uploaded into the system.	
Postconditions in words (write in passive and past tense)	The patient came in with symptoms that was looked at by a different doctor and the patient brought with them all the documents from their previous doctor. These documents are important and should be saved in the patient's medical history to help resolve the patient's current issues.	
Trigger	Patient has important documents from prior doctors that are relevant to her current case.	
Included Use Cases		
Extending Use Cases		
MAIN SUCCESSFUL	***Actor Action***	***System Action***

SCENARIO in numbered sequence	1. Scans documents	2. Displays the scanned document
	3. Doctor uploads the document to the patient's file	4. System saves the document in proper format
	6. Doctor saves the document.	7. System saves documents in the server for future reference and can be viewed on any computer with the server access.
OTHER SUCCESSFUL SCENARIOS	***Step***	***Branching Action***
UNSUCCESSFUL SCENARIOS	***Conditions***	***Actions***
	1b. The document is too big	2b. System denies upload
	6b. The scanner is broken or is not connected to the computer.	7b. No files can be uploaded.
Priority in scheduling	3rd Priority	
Frequency	Will be performed every time there are important outside documents about a patient.	
Business rules and data logic	Upload as many documents as needed The files should be in a common format so any computer view it.	
Other non-functional requirements	System must respond in less than 1 second and not keep the doctor waiting. Patient information must be backed up. Patient information must be kept secure and only the patient and the doctor may have access to the information in order to avoid HIPAA violations. Scans should be clear and readable.	
Superordinates	Manage visit information.	
Developer	Sm Kabir	

Creation date and last modified date	Created on 02/19/20

USE CASE DESCRIPTION		
Use Case #	P9	
Use Case Name	Manage Staff	
Actor	Admin	
Goal (1 phrase)	To maintain all staff's details including create, update and delete.	
Overview and scope	When a new staff comes in, the admin will create a new record. The admin also updates staff records if there are any changes to their details such as contact details or address. Admin is also able to delete old staff records if they stop working in the clinic or remove any duplicate record.	
Level	Primary	
Preconditions		
Postconditions in words (write in passive and past tense)	A record will be created for the new staff. A record will be updated if there are any changes to the current staff's details. A record will be deleted if there is any duplicate record or non-existing staff.	
Trigger	New staff joins the clinic or changes in current staff's details or staff stops working or duplicate staff record	
Included Use Cases		
Extending Use Cases		
MAIN SUCCESSFUL SCENARIO in numbered sequence	***Actor Action***	***System Action***
	1. Admin logs in	2. System provides access

	3. Admin requests to create a new staff record	4. System provides the form to fill in for the new staff
	5. Admin submit the form	5. System create a new record
	6. Admin requests to update current staff record	7. System displays current staff record
	8. Admin fill in new details to staff record and submit the form	9. System update a new record
	10. Admin request to delete a record	11. System displays all staff records
	12. Admin select a record to delete	13. System prompt for delete confirmation
	14. Admin confirm the delete	15. System delete the record
OTHER SUCCESSFUL SCENARIOS	***Step***	***Branching Action***
UNSUCCESSFUL SCENARIOS	***Conditions***	***Actions***
	1b. Admin provides wrong username or password	2b. System denies access
	3b. Admin does not enter one of the required information	4b. System prompts all fields must be complete to proceed
Priority in scheduling	1st Priority	
Frequency	Will be performed every time a new staff join the clinic or changes to their details or stop working at the clinic	
Business rules and data logic	All fields must be completed to create a new record All current staff must have a record in the database All current staff must notify the admin within 7 days of changes to their details Admin must delete staff record who stops working in the clinic	

Other non-functional requirements	System must respond in less than 2 second Staff details must be backed up All staff details must be encrypted
Superordinates	
Developer	Sophearith Rin
Creation date and last modified date	Created on 02/20/20

USE CASE DESCRIPTION	
Use Case #	P10
Use Case Name	Manage Clinical Services
Actor	Admin
Goal (1 phrase)	To manage all clinical services including create, update and delete.
Overview and scope	When there is a new clinical service available to the clinic, the admin will create a new service record. The admin also updates clinical service records if there are any changes to the rates. Admin is also able to delete clinical services that are no longer available in the clinic.
Level	Primary
Preconditions	
Postconditions in words (write in passive and past tense)	A record will be created for the new clinical service. A record will be updated for the new rates . A record will be deleted for the clinical service that is no longer available.
Trigger	New equipment is brought into the clinic that can perform medical services or changes in the rate for the current services or current services that are no longer available.

Included Use Cases		
Extending Use Cases		
MAIN SUCCESSFUL SCENARIO in numbered sequence	***Actor Action***	***System Action***
	1. Admin logs in	2. System provides access
	3. Admin requests to create a new clinical service record	4. System provides the form to fill in for the service
	5. Admin submit the form	5. System create a new record
	6. Admin requests to update current clinical service record	7. System displays current clinical service record
	8. Admin fill in new details to clinical service record and submit the form	9. System update a new record
	10. Admin request to delete a record	11. System displays all clinical service records
	12. Admin select a record to delete	13. System prompt for delete confirmation
	14. Admin confirm the delete	15. System delete the record
OTHER SUCCESSFUL SCENARIOS	***Step***	***Branching Action***
UNSUCCESSFUL SCENARIOS	***Conditions***	***Actions***
	1b. Admin provides wrong username or password	2b. System denies access
	3b. Admin does not enter one of the required information	4b. System prompts all fields must be complete to proceed
Priority in scheduling	1st Priority	

Frequency	Will be performed every time a clinical service is available or changes to current rate or service no longer available
Business rules and data logic	All fields must be completed to create a new record All available clinical services must have a record in the database All changes in the clinical service rate must be updated instantly Admin must delete clinical service records that are no available in the clinic
Other non-functional requirements	System must respond in less than 2 second Clinical service details must be backed up All staff clinical services record must be encrypted
Superordinates	
Developer	Sophearith Rin
Creation date and last modified date	Created on 02/20/20

USE CASE DESCRIPTION	
Use Case #	P3
Use Case Name	Generate Invoice
Actor	Receptionist
Goal (1 phrase)	To create invoices for the patient visit
Overview and scope	When a patient is done with their visit, the receptionist will create an invoice to bill the insurance company that represents the patient. Sending the invoice to the insurance companies for billing is out of scope for our system, therefore is not part of our use case diagram.
Level	Primary

Preconditions	1. Receptionist is logged into the system 2. The patient has shown the insurance card to the receptionist for validation. 3. The patient visit is finished.	
Postconditions in words (write in passive and past tense)	After the visit is conducted with one of the medical professionals, the receptionist will generate the invoice for the insurance company that represents the patient and also print a copy for the patient.	
Trigger	Payment is recorded and a copy of the receipt is saved.	
Included Use Cases	Print Invoice	
Extending Use Cases		
MAIN SUCCESSFUL SCENARIO in numbered sequence	***Actor Action***	***System Action***
	1. Receptionist logs in	2. System provides access
	3. Receptionist requests to view patient social security and insurance provider ID.	
	5. Receptionist inputs patient insurance provider information	6. Invoice generated will include invoice Id, visit Id, patient Id, staff Id, social security, service fees, tax and total.
	7. 6. Receptionist prints a copy of the invoice for the patient.	7. Invoice is sent to the insurance provider via email.
OTHER SUCCESSFUL SCENARIOS	***Step***	***Branching Action***
	Conditions	***Actions***

UNSUCCESSFUL SCENARIOS	1b. Receptionist provides wrong username or password	2b. System denies access
	6b. Receptionist does not enter one of the required information	7b. System prompts all fields must be complete to proceed
Priority in scheduling	1st Priority	
Frequency	Will be performed every time after a visit is conducted.	
Business rules and data logic	Insurance is able to recognize a patient based on social security. Invoices cannot be updated once created. Any updates should require new invoice creation.	
Other non-functional requirements	Data must be consistent system wide. More than one person should not be able to edit a record at the same time.	
Developer	BLEDAR NOKA	
Creation date and last modified date	Created on 02/23/20	

USE CASE DESCRIPTION		
Use Case #	I1	
Use Case Name	Print Invoice	
Actor	Receptionist	
Goal (1 phrase)	To generate a copy of the receipt for the patient.	
Overview and scope	After inputting all actions needed to record the visit, receptionist accepts form of payment and prints receipt.	
Level	Include	
Preconditions	1. Receptionist is logged into the system 2. The patient visit is finished.	
Postconditions in words (write in passive and past tense)	After the visit is conducted with one of the medical professionals, the receptionist will print a copy of the receipt and give it to the patient.	
Trigger	Payment is recorded and a copy of the receipt is saved.	
Included Use Cases	None	
Extending Use Cases	None	
MAIN SUCCESSFUL SCENARIO in numbered sequence	***Actor Action***	***System Action***
	1. Receptionist requests to print the receipt.	2. Invoice is printed.
	3. Receptionist hands the copy to the patient.	

OTHER SUCCESSFUL SCENARIOS	***Step***	***Branching Action***
UNSUCCESSFUL SCENARIOS	***Conditions***	***Actions***
	1b. Receptionist does not request to print the receipt.	2b. Invoice is not printed.
	3b. Receptionist does not give a copy to the patient.	
Priority in scheduling	Included	
Frequency	Will be performed every time after a visit is conducted.	
Business rules and data logic	A copy must be generated for the patient after the visit. Invoices cannot be updated once created. Any updates should require new invoice creation.	
Other non-functional requirements	The receptionist must have access to a desktop.	
Superordinates	None	
Developer	BLEDAR NOKA	
Creation date and last modified date	Created on 02/23/20	

3.7 Discussion

Modelling a software is an essential process to develop a good system. It acts as a blueprint and roadmap to develop a software that generally can be used by all types of user. There are many different modelling methods that exist in the software developing world and each of them serves their own purpose. Use case model and class model in this case is a head start to develop a system. Both of these models include all the necessary processes and steps that indicate the interaction between the users and the system. However, two of these models are not sufficient to indicate all the important aspects of the system. Alternative solution to this case is to introduce a few more modelling techniques such as sequence diagram, activity diagram, state diagram and database schema. With the combination of all these modelling techniques, it would draw out a clear and full picture of the system.

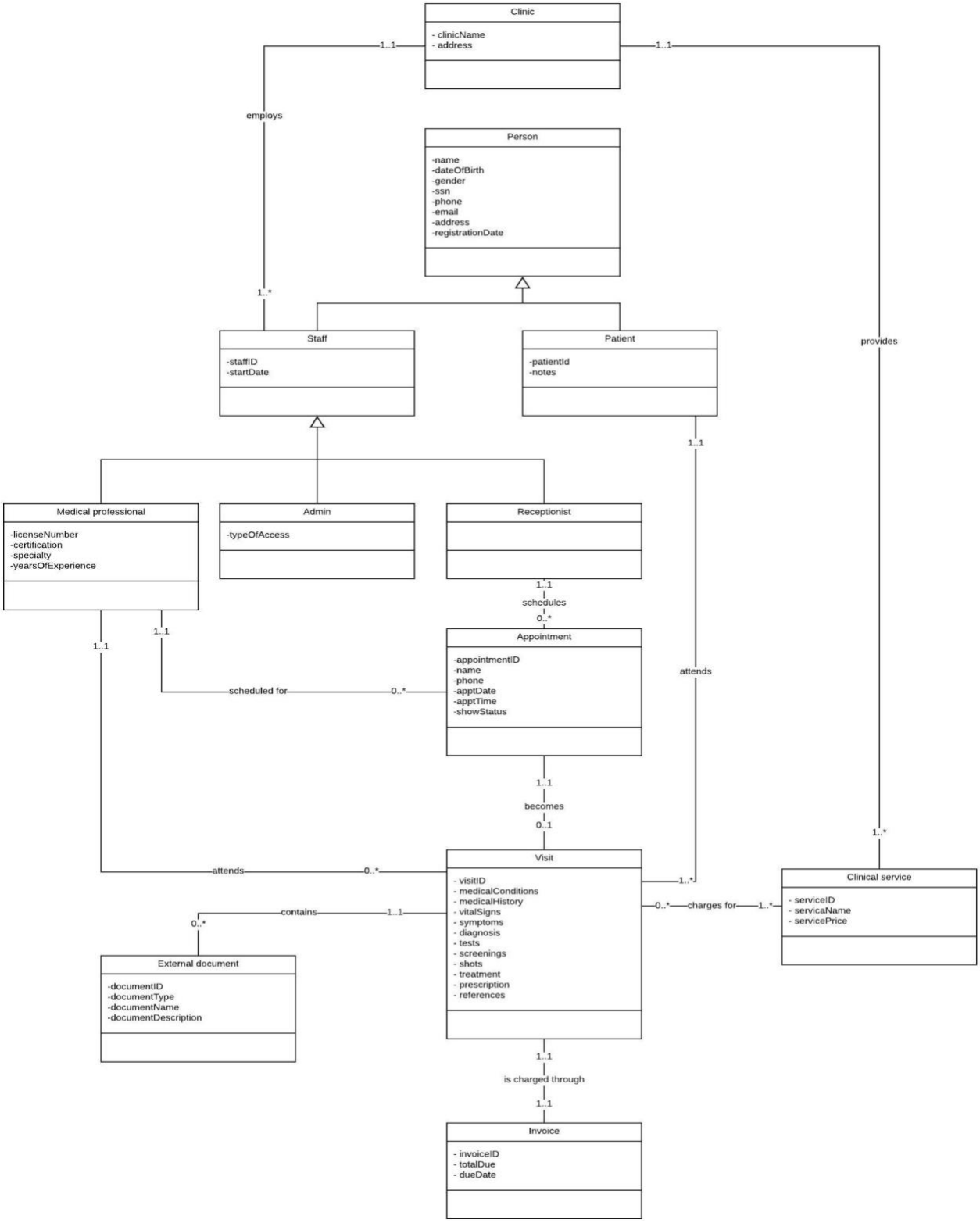
(4) Class Model

4.1 TCM Table

Noun	Class Elimination Rules Applied (Step N2)	Class Categories Applied (Step N3)	Class
Clinic	NO	Physical things (CC3)	Clinic
Staff		Roles of People (CC1)	Person
Medical professional	NO	Roles of People (CC1)	Person, Staff
Patients	NO	Roles of People (CC1)	Person
Admin	NO	Roles of People (CC1)	Person, Staff
Visit	NO	Event (CC5)	Visit
Medical history	Attribute (CER7)		Attr: Visit
Appointment	NO	Event (CC5)	Appointment
Invoices	NO	Physical thing (CC3)	Invoice
clinical services	NO	Physical thing (CC3)	Medical service
Users	Redundant (CER1)		
Diagnosis	Attribute (CER7)		Attr: Visit
Treatments	Attribute (CER7)		Attr: Visit

Symptoms	Attribute (CER7)		Attr: Visit
Doctors	Redundant (CER1)		
Nurses	Redundant (CER1)		
Employees	Redundant (CER1)		
Receptionist	NO	Roles of People (CC1)	Person, Staff
Visit Report	Derived (CER9)		
Documents	Redundant (CER1)		
Test	Attribute (CER7)		Attr: Medical services
Patient visit history	Derived (CER9)		
Prescription	Attribute (CER7)		Attr: Visit
Insurance	Irrelevant (CER2)		
Show status	Values (CER8)		
External document		Physical thing (CC3)	External Document
Doctors Schedule	Derived (CER9)		

4.2 Class Diagram



4.3 Selected Class Definitions

All classes are intuitive.

4.4 Selection Association Definitions

All classes are intuitive.

4.5 Discussion

A clinic employs one or more staff and provides at least one clinical service. A staff member is a person who can be a medical professional, an admin or a receptionist. When a patient (who is also a person) calls to make an appointment with a medical professional, the receptionist looks at scheduled appointments for the medical professional and suggests a date and time when a new appointment can be scheduled. If the patient agrees, the receptionist schedules an appointment. A receptionist can schedule no appointments at all or multiple appointments and every appointment is scheduled for exactly one medical professional. When a patient shows up for the appointment, the receptionist confirms the appointment and the appointment becomes a visit. An appointment must be made for a visit therefore every visit is affiliated with exactly one appointment. But an appointment may or may not become a visit depending on whether the patient showed up or not. The visit is attended by a single medical professional and a single patient. A medical professional can attend no visits at all or attend multiple visits. A patient should have attended at least one visit and can attend multiple visits. Every visit may contain zero or many external documents but every external document is recorded at a particular visit. Every visit charges for one or many clinical services and a clinical service may be assigned to no visits or multiple visits. The payment for the visit is charged through an invoice. Every visit can have only one invoice and every invoice must be from a single visit.

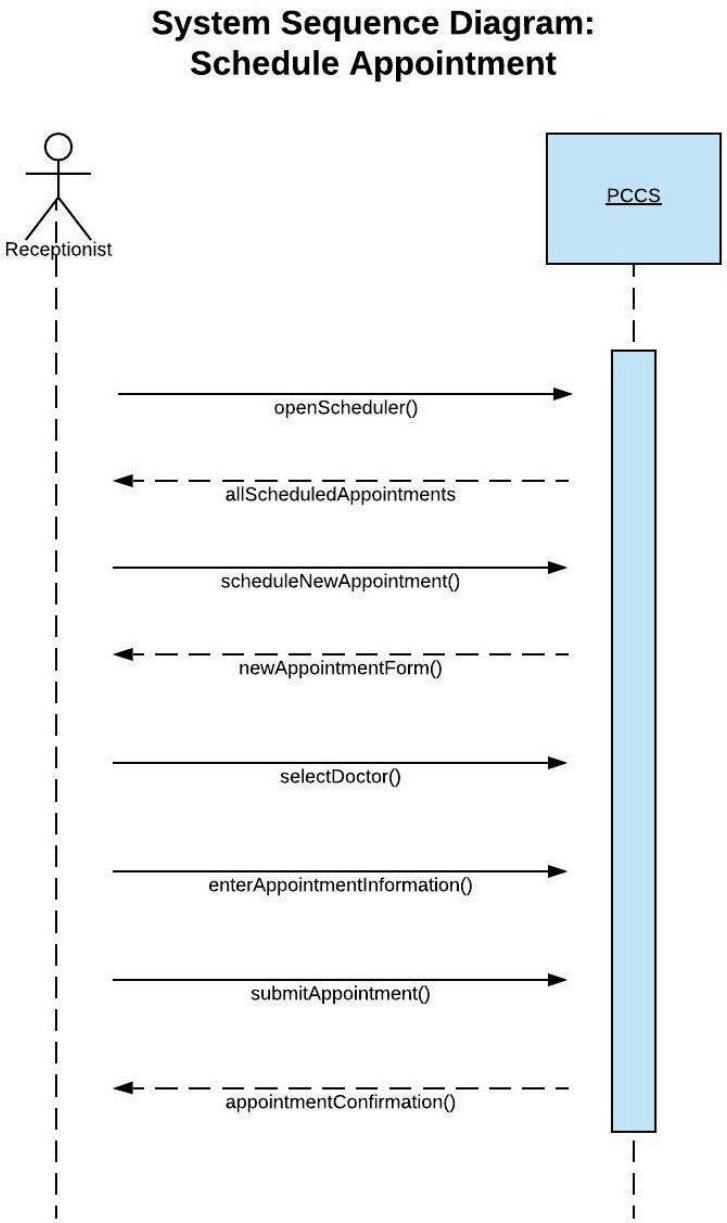
The class diagram was updated as the roles of the classes became more and more clear throughout the project phases. For example, earlier versions of the class diagram showed an inappropriate association between patient class and visit class. Also, we removed the association between the clinic and person class. After some suggestions and group discussions we realized the mistake and made changes thus, the final version was chosen to represent more closely the goal of the system.

CHAPTER 2 System Design

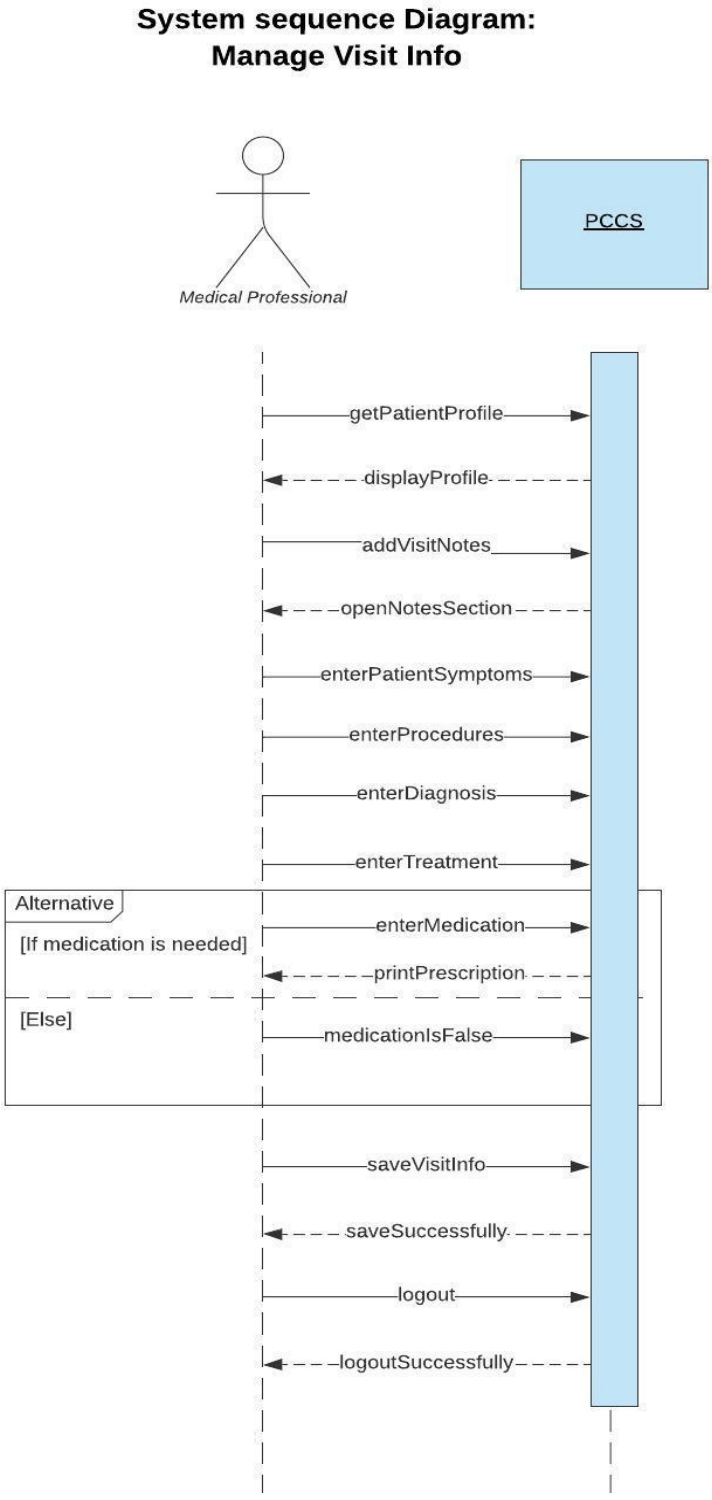
(5) Sequence Diagrams

5.1 System Sequence Diagrams

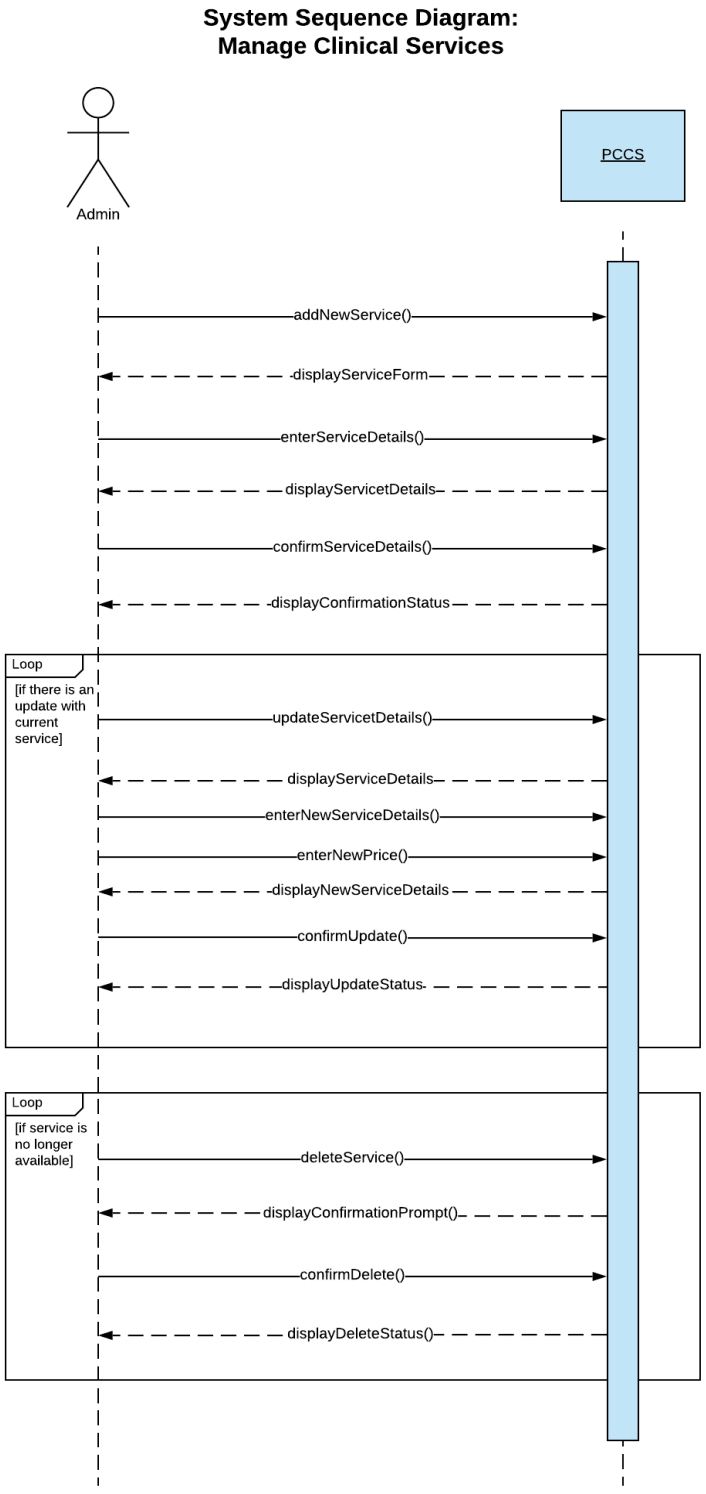
5.1.1 Use Case - Schedule Appointment - Shahrar Nizam



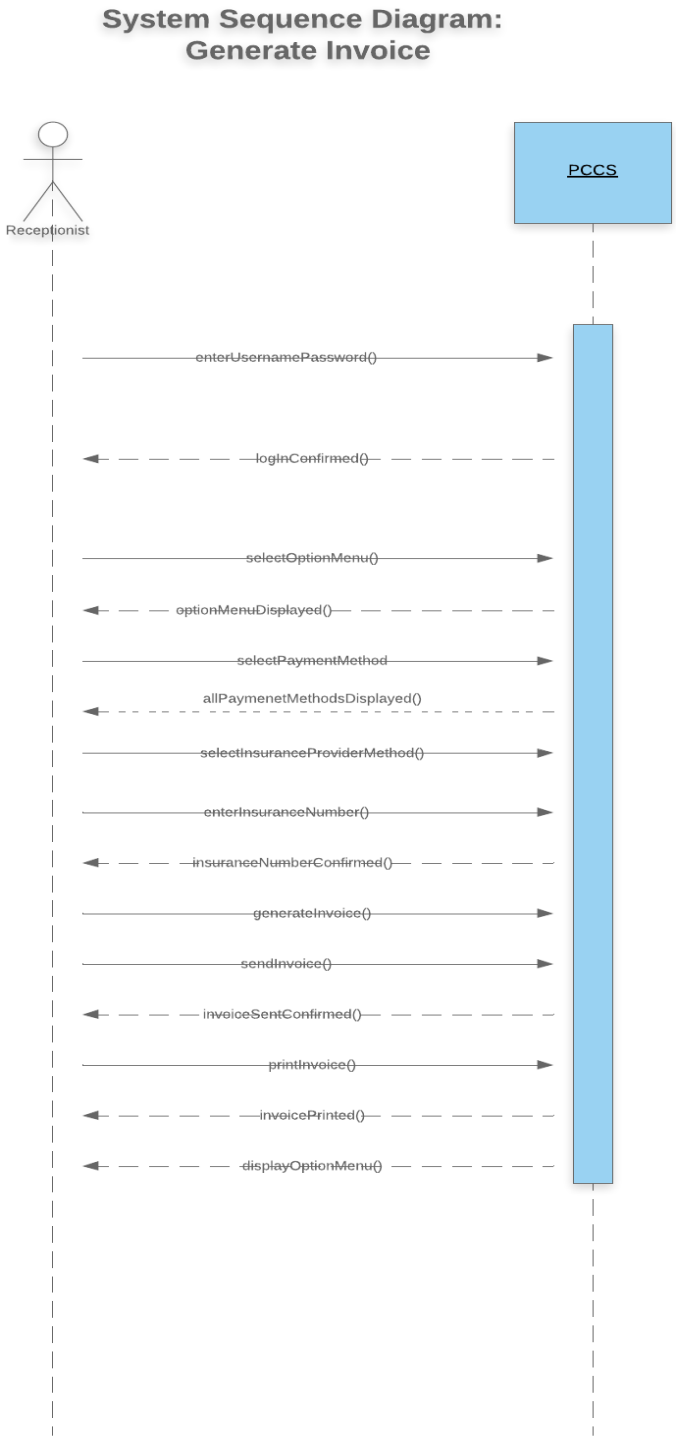
5.1.2 Use Case - Manage Visit Info - SM Kabir



5.1.3 Use Case - Manage Clinical Services - Sophearith Rin

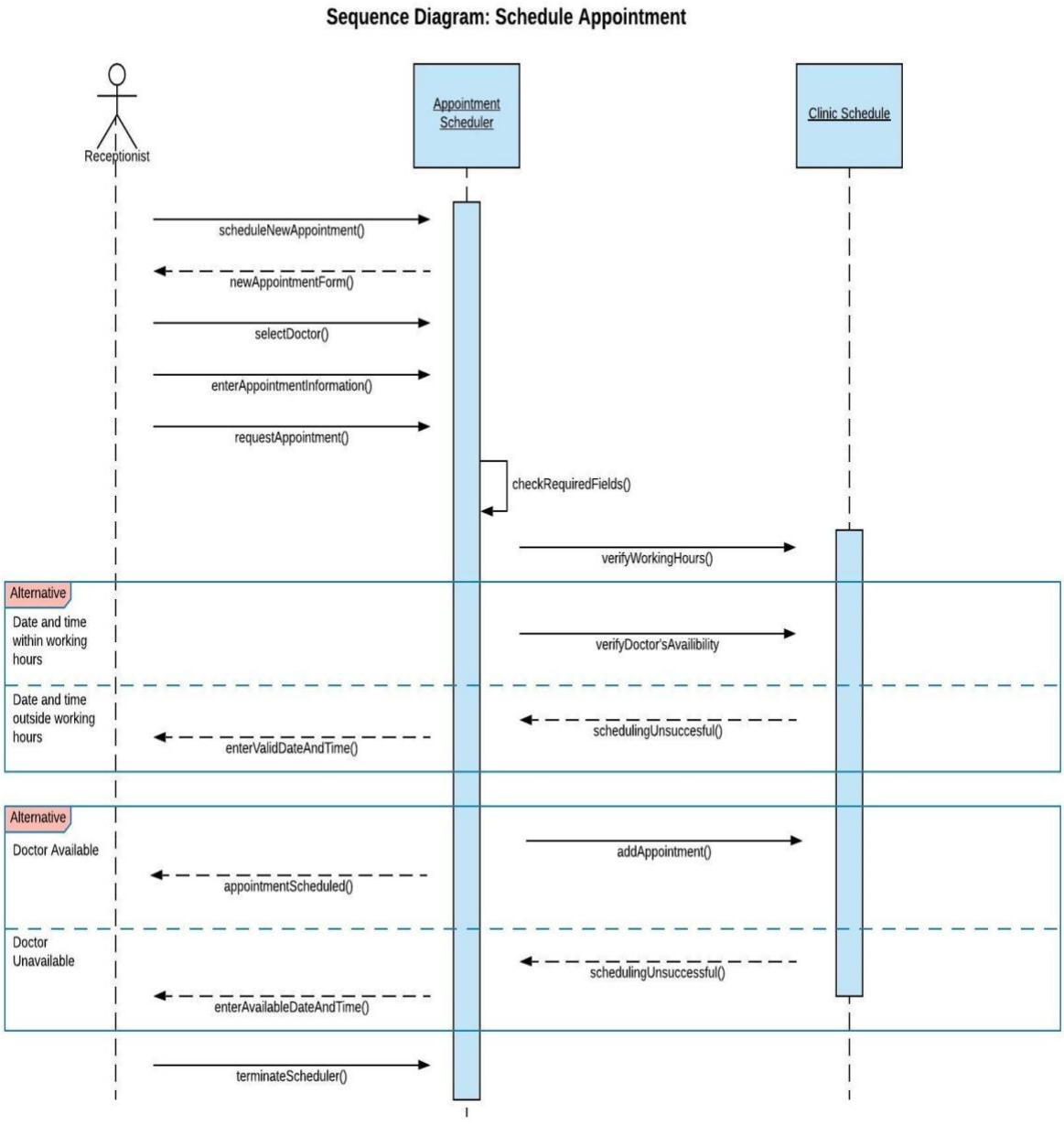


5.1.4 Use Case - Generate Invoice - Bledar Noka

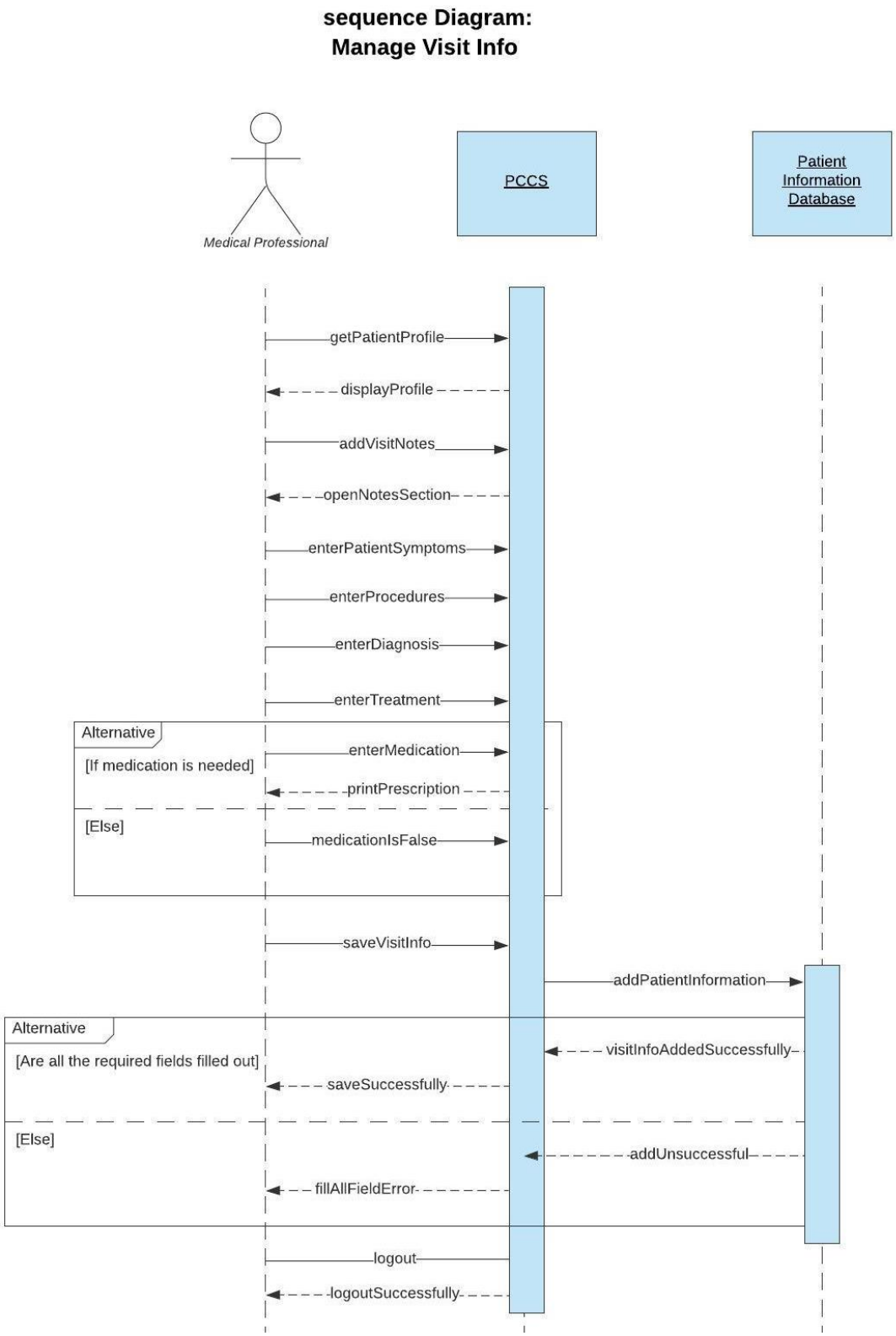


5.2 Sequence Diagrams

5.2.1 Use Case - Schedule Appointment - Shahrar Nizam

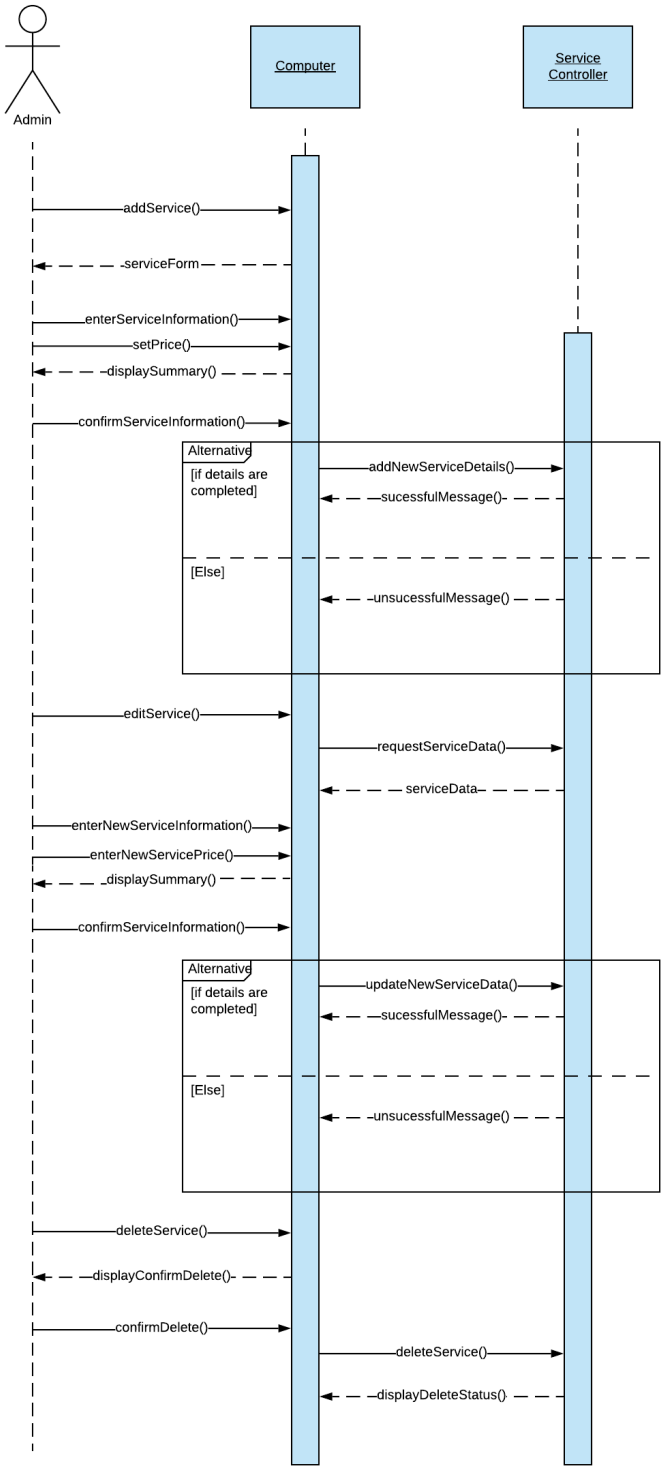


5.2.2 Use Case - Manage Visit Info - SM Kabir

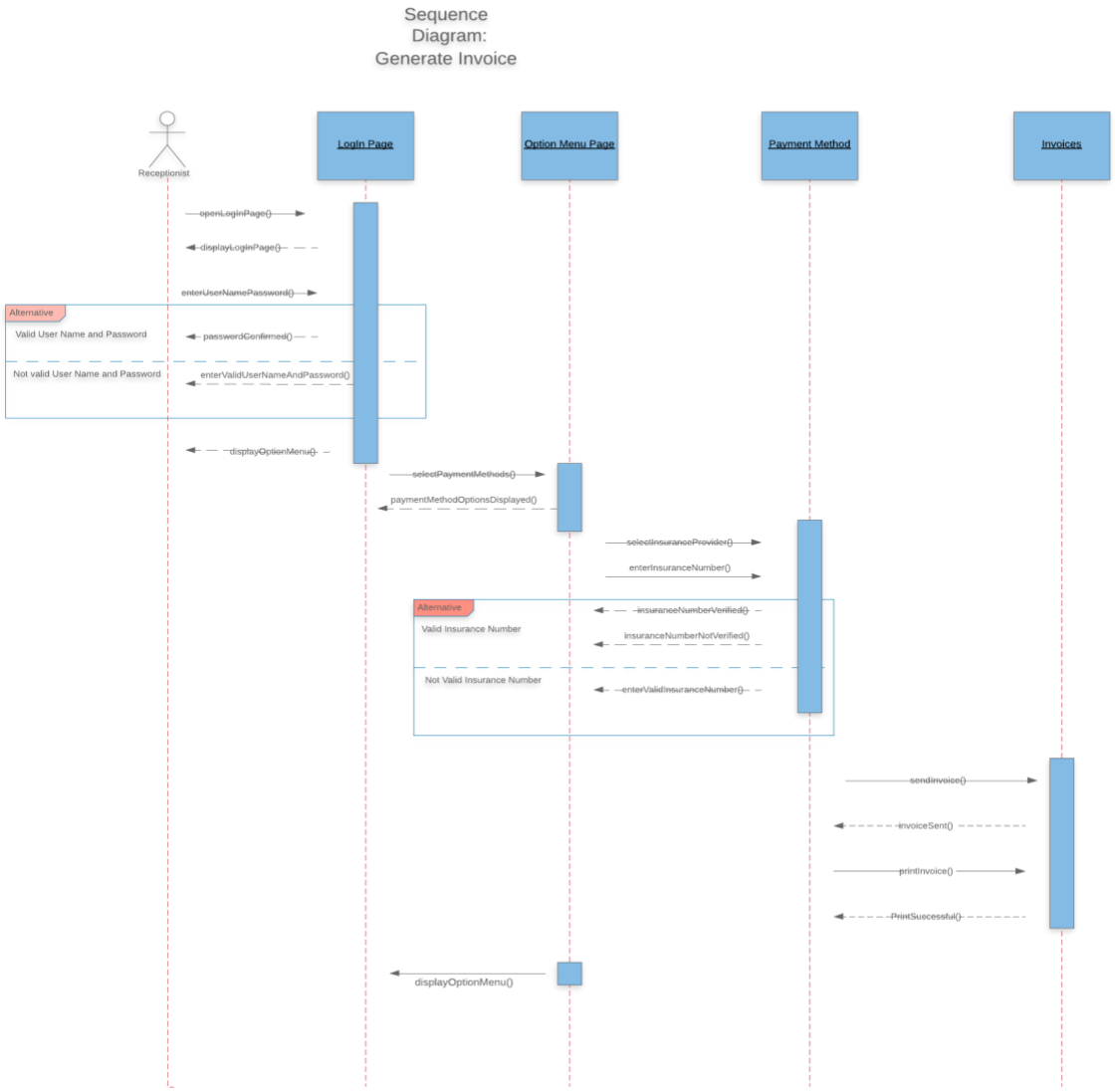


5.2.3 Use Case - Manage Clinical Services - Sophearith Rin

Sequence Diagram: Manage Clinical Services



5.2.4 Use Case - Generate Invoice - Bledar Noka



The above diagram illustrates an invoice being generated through the system. The receptionist accesses the login page where the username and password credentials are required to be inserted to validate the user. If credentials are not valid, the system requires the receptionist to enter them again, if valid the system proceeds onto the next page, displaying the optional menu page. The receptionist selects the payment method option which is through the insurance provider. Then, input the patient's insurance number. If the insurance number is not valid, the system will require to re-enter the number again, and if verified, the system will proceed onto the next step of sending the invoice to the insurance provider and also printing a copy for the patient. The system will then display back to the options menu.

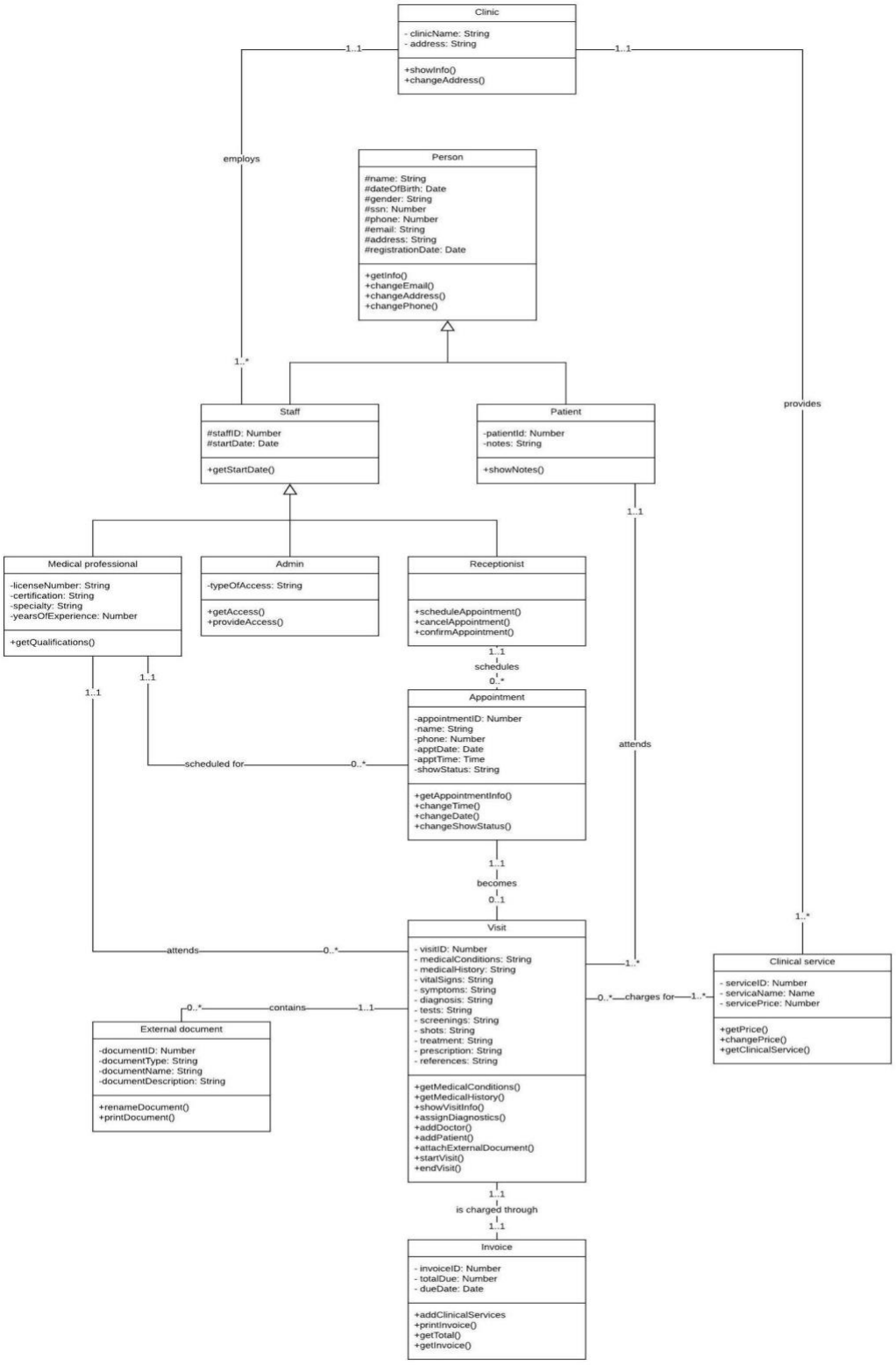
5.3 Validation and Discussion of Sequence Diagrams

The system sequence diagrams illustrate how a certain type of user (actor) uses the system in reference to a particular use case depicted in the use case diagram. The illustration presents a black box on the system side and only represents the requests to the system in solid lines and responses from the system in dotted lines. The bars on the system side represent the time the system remains active.

The system sequence diagrams go in-depth into the use case diagrams and present a white box view into the system. It captures the communications inside the system that the user is not exposed to. The alternative box shows what would happen in certain scenarios passed or failed. The communications inside the option box show communications that are optional.

(6) Design Class Diagram

6.1 Design Class Diagram



6.2 Validation and Discussion of Design Class Diagram

The Design Class Diagram shown above includes three parts: the class in the first part, attributes in the second part and methods in the third part. Invoice class and external document class is derived from the visit class. However, the clinical service class is derived from both the clinic class and visit class. The clinic can provide staff to help patients but also provide clinical services and this is done after the patient has finished the visit with a medical professional. Staff and patient classes are derived from the person's class. However, the medical professional, admin and receptionist classes are all derived from the staff class. The appointment class has a 1-1 (only one) relationship with the receptionist because only the receptionist can schedule an appointment. Whereas, only an appointment can show a patient's visit that is done by the medical professionals.

CHAPTER 3 Physical Design

7.1 Relational Schemas

- Clinic (clinicName, address)
- Medical Professional (staffID, name, dateOfBirth, gender, ssn, phone,email,address, registrationDate, startDate, licenseNumber, certification, specialty, yearsOfexperience, clinicName)
- Admin (staffID, name, dateOfBirth, gender, ssn, phone,email,address, registrationDate, startDate, typeOfAccess, clinicName)
- Receptionist (staffID, name, dateOfBirth, gender, ssn, phone,email,address, registrationDate, startDate, clinicName)
- Patient (patientID, name, dateOfBirth, gender, ssn, phone,email,address, registrationDate, notes)
- Appointment (appointmentID, name, phone, apptDate, apptTime, showStatus, doctor, receptionist)
- Visit (visitID, medicalCondition, medicalHistory, vitalSigns, symptoms, diagnosis, tests, screenings, shots, treatments, prescriptions, references, doctor, patient, appointmentID)
- Clinical Services (serviceID, serviceName, servicePrice, clinicName)
- Clinical Charges (visitID, serviceID)
- Invoice (invoiceID, totalDue, dueDate, visitID)
- External Document (documentID, documentType, documentName, documentDescription, visitID)

CHAPTER 4 Evaluation of Analysis & Design

8.1 Evaluation of the Project

When our group first met, we decided to go with a simplified patient care clinic system (PCCS). At first it looked like a straightforward project, but it turned out to be much more complex and challenging than we anticipated. Despite this, we still managed to complete our project proposal. This was achieved after many group meetings, group discussions, and thanks to the decision that we took in researching other patient care systems available in Philadelphia. One of the members of the team, Shahrar, also spoke with an individual who works at a patient care clinic. This helped the team understand the requirements of the project. After understanding the requirements of the system, our group managed to split the work individually and completed the project proposal relatively quickly. We completed the rest of the chapters of this project week by week, after we would learn a new concept in class, such as the new diagrams needed for the stages of the project.

If we had more time, we would have conducted an ERD diagram as it would have helped in doing the relational database schema faster.

Even though we spent many weeks and had many group meetings together, we all would have liked to have more time to produce a better product. However, we are confident and happy with the current product delivered.

8.2 Evaluation of the UML and Tool

Our group enjoyed working with the UML diagrams on this project. At first, we faced some problems in understanding and conducting our Use Case diagram but after a consultation with the professor and a group meeting, we made some changes and refined it into the current working model.

Also, we struggled in understanding the difference between the Domain and Design Class Model, but after reading the lecture notes carefully, the Dr. Song paper and a few YouTube videos, we managed to understand and draw them successfully. We believe that having a great understanding of the UML diagrams and being able to draw them successfully is a skill that is most needed and rewarded in the object oriented design field.

Appendix

9.1 Lessons Learned

Shahrar Nizam

During this project, I was able to learn the process of analyzing a system and taking a conceptual idea to an implementable design. At first, the concept of UML seemed quite simple as drawing out relationships, processes and objects did not seem very hard but as we got involved in actually analyzing and designing the concepts, it seemed to be a lot harder than expected. It is not actually as easy to put thoughts into formal diagrams. The constraints and protocols for UML made the task more challenging as it took away the freedom to express theoretical concepts. But in the end, I realized that it was these protocols that helped filter out things that could have complicated or confused the concept more. Understanding of object-oriented design was the key to success in this project. I have learned how to design use case diagrams, class diagrams, sequence diagrams and more importantly the fundamental idea that these diagrams present. Working in a group it was also clear that UML is subjective and there is no right or wrong way to design the same concept as everyone had a different way to present the same things. This project pushed me to think critically and in-depth about the interactions inside the system and between its users. In the future, knowledge learned from this project will help me analyze system designs and clarify requirements with clients and other stakeholders.

SM Kabir

While working on this project I have learned how complex it is to design software. At first, the idea of just drawing out what to do seemed very simple and easy to do. Once we started the process it got very complex. When we come up with one great idea it also brought up many questions on how good that idea truly is. Even though some of these protocols seem tedious and unnecessary at first but in the end, it all made sense and truly helped us. These protocols are here to give us guidance and to make sure the project we are the best it can be. One of the biggest surprises for me throughout this process was how many different ways you can go about this. I always thought that with big projects such as this one, there is only one right way to go about it and everything has to be done properly. But, as we worked together we all came up with different ideas and they all seemed doable. There is no one way to make a project work. Just because there are protocols to follow does not mean it is a linear process. This idea will help me in the real world just as much it has done for me during this project.

Sophearith Rin

At the end of this project, I have learned that to build a good system is not all about coding but also consider all the other aspects as well such as design, usability, and flow. To effectively design a system, proper modeling technique is essential to determine how the user interacts with the system and how the system should behave. During the project, I have learned that the diagram is the most effective way to design a system and there is no one correct answer to a

diagram. Each person has their own way of doing things. Working as a team, I have learned that coming together and gathering all our ideas to design a system is very effective to find the best solution to the system. I have always thought that different diagram design would result in a different outcome. But as we work throughout this project, I have learned that different diagrams can also mean a better process and design to get the same thing done, possibly at a lower cost and shorter time. Teamwork is a very important factor during the making of this documentation. The idea of the importance of teamwork will motivate me to among the team members for future classes and workplaces.

BLEDAR NOKA

There were so many lessons learned throughout this project. I had never had any prior experience with object-oriented design concepts; therefore, this project was quite challenging and also very interesting as it introduced me to new concepts and models.
At first, the concept of UML seemed straightforward and intuitive, however, whenever I would think of drawing a diagram (use-case, class, design, system or sequence diagram) it would become very messy because I was faced with the challenge of putting thoughts into drawing those diagrams.
However, it should be said that there is no right or wrong way in drawing a diagram. Despite the protocols that UML includes, it does not have to limit the thinking process when drawing diagrams.
Another lesson learned was the importance of the group in a project such as this one. Our group was composed of individuals with different experiences and coming from different backgrounds, which proved to be a great advantage as we were able to help each other in understanding various concepts. It should be said that the members of this group were outstanding with their contribution and their help. I learned that having great members when working on a project is crucial in concluding a great project such as this one.

9.2 Division of the work among team members

Sections of the Project	**Members contribution**
Title page, table of contents	Sophearith Rin
Introduction	Bledar Noka
1.0 Problem statement (Goal, Importance, In-scope, Out-scope)	Sophearith Rin, Bledar Noka

2.1 Functional Requirements	Shahrar Nizam
2.2 Data Requirements	Sm Kabir
2.3 Business Rules and Logic	Shahrar Nizam
2.4 Non-Functional Requirements	Shahrar Nizam
2.5 Other Important Assumptions	Shahrar Nizam
3.1 “Stories” Requirements Document	Shahrar Nizam, Bledar Noka, Sophearith Rin, SM Kabir
3.2 Actors and Their Goals	Bledar Noka
3.3. Use Case diagram	Shahrar Nizam, Bledar Noka, Sophearith Rin, SM Kabir
3.4 Overview section of all use cases	Shahrar Nizam
3.5 Use case descriptions	Shahrar Nizam, Bledar Noka, Sophearith Rin, SM Kabir
3.7 Discussion	Sophearith Rin
4.1 TCM Table	Shahrar Nizam, Bledar Noka, Sophearith Rin, SM Kabir
4.2 The class diagram	Shahrar Nizam, Bledar Noka, Sophearith Rin, SM Kabir
4.3 Selected Class definitions	N/A
4.4 Selected Association definitions	N/A
4.5 Discussion	Shahrar Nizam, Bledar Noka
5.1 System Sequence Diagram	Shahrar Nizam, Bledar Noka, Sophearith Rin, SM Kabir

5.2 Sequence Diagram	Shahrar Nizam, Bledar Noka, Sophearith Rin, SM Kabir
5.3 Validation and Discussion of Sequence Diagrams	Shahrar Nizam
6.1 Design Class Diagram	Shahrar Nizam, Bledar Noka, Sophearith Rin, SM Kabir
6.2 Validation and Discussion	Bledar Noka
7.1 Relational Database Schema	SM Kabir, Shahrar Nizam
8.1 Evaluation of Project	Bledar Noka
8.2 Evaluation of UML and Tool	Bledar Noka
9.1 Lesson learned	Shahrar Nizam, Bledar Noka, Sophearith Rin, SM Kabir
9.2 Division of the work among team members	Shahrar Nizam, Bledar Noka, Sophearith Rin, SM Kabir
9.3 Unsolved Problems	Bledar Noka

9.3 Unsolved problems

In our use case diagram, we did not include a secondary actor. For example, we could have included a lab or pharmacy which could have shown where the patient receives the prescriptions or the test results from.