

# HAWASSA UNIVERSITY



**FACULTY OF INFORMATICS**  
**DEPARTMENT OF COMPUTER SCIENCE**  
**SOFTWARE ENGINEERING PROJECT**  
**TITLE : STUDENT CLINIC SYSTEM**

## GROUP 5

KALEAB TESFAYE	1801/14
NATHAN ABAYNEH	2403/14
GETAHUN DERJE	1463/14
EYOB HAILU	1249/14
VENUS HAILEMESKEL	3064/14

**Submitted To: Mr. Belilgn**

**Submission Date: February 7 , 2024 , 2024**

## **Part 2 : System analysis**

### **2.1 DATA COLLECTION TOOLS**

The term "data collection tools" describes the equipment, procedures, or software used to collect, store, and organize data for analysis or study. These tools are intended to make the process of gathering data more efficient and accurate while maintaining consistency and accuracy in the collection of pertinent data. Data gathering instruments can vary widely across various industries, covering conventional techniques like questionnaires and interviews as well as contemporary innovations like wearables, sensors, and customized software programs.

**Surveys and Questionnaires:** These organized methods of gathering data are intended to obtain information from people or groups. A written or digital format including a preset set of questions is the standard structure for surveys. Open-ended questions that enable respondents to offer detailed responses or closed-ended questions with predetermined response alternatives are both common types of questionnaires. These resources are helpful for gathering information on a variety of subjects, such as preferences, experiences, opinions, and actions, both quantitatively and qualitatively. Scalable questionnaires and surveys allow researchers to get information from huge groups of people and provide a methodical way to understand different points of view.

**Interviews:** In an interview, a participant or group of participants and a researcher communicate directly and actively. By allowing for in-depth topic research, this qualitative data collection method offers a greater knowledge of attitudes, opinions, and experiences. Structured, semi-structured, and unstructured interviews are all possible. While semi-structured and unstructured interviews offer greater flexibility and enable the conversation to flow naturally, structured interviews follow to a specified list of questions. Developing interaction with participants, getting more detailed insights, and explaining answers are all made possible by this useful tool.

**Observation:** In observation, activities, events, or processes are carefully observed and documented in their natural environments without the researcher's intervention or influence. Researching environmental phenomena, organizational procedures, and human behavior are common uses for this qualitative data collection method. Depending on the degree of researcher participation, observations may be participant or non-participant. The researcher may classify observed actions using established coding methods. With its unique perspective that other methods can miss, this instrument offers direct, real-time insights into contextual elements, social dynamics, and behavioral patterns.

In order to conduct research and make decisions, data collection tools are essential tools that facilitate the systematic collection of information needed to provide helpful insights. Questionnaires and surveys offer systematic methods for gathering both qualitative and quantitative data, making them effective instruments for understanding beliefs, inclinations, and actions on a broad scale. Direct conversation during an interview allows for in-depth investigation and delves into complex viewpoints and experiences for researchers. On the other hand, observation offers up-to-date knowledge on actions and procedures in unchanged environments. By using these tools, a thorough approach to data collecting is ensured, allowing researchers, analysts, and decision-makers to make well-informed decisions based on a solid body of data. They can also spot patterns and develop appropriate conclusions.

## **2.2 SYSTEM REQUIREMENT**

The fundamental element of the entire software development lifecycle is the Software Requirements Specification. It facilitates the development of a common understanding of the project's goals and specifications across customers, including clients, testers, and developers. This clarity is necessary for effective project management, collaboration, and the creation of software that meets up to user expectations.

System requirements are the qualities, both functional and nonfunctional, that a software system needs to have in order to meet the requirements.

### **2.2.1 Functional Requirements**

Functional requirements are requirements that list the attributes and functionalities a software system requires to satisfy its consumers. These specifications outline the functionality and behavior of the system with a focus on what the program must be able to perform. The following list is the functional requirement for our project .

- The system should allow the doctor to login
- The system should allow the doctor to access patient data
- The system should allow the doctor to generate report
- The system should allow the doctor to write a prescription for patients
- The system shouldn't allow except the doctor to access patient medical history
- The system should allow the doctor to write a referral to another hospital
- The system should allow the doctor to generate printable report , referrals and prescriptions
- The system should allow the doctor to update patients medical information
- The system should allow the reception to login
- The system shouldn't allow the receptionist to access patients medical information
- The system should allow the reception to register new patient to the system
- The system should allow the doctor to give a report information for missed class during treatment
- The system should allow the pharmacist to login to the system
- The system should allow the pharmacist to search for available medication in store
- The system should allow the pharmacist to generate report in order to track stock level

- The system should allow the pharmacist to manage medical drug information (add new medical drug , to update an already available drug information )
- The system should allow the admin to login to the system
- The system should allow the admin to validate when new user is created (doctor , pharmacist and receptionist )
- The system should allow the admin to deactivate user account

## **Non functional requirements**

Non-functional requirements are aspects of a software system that describe its characteristics and qualities, rather than specific behaviors or features. These requirements focus on attributes such as performance, reliability, security, and usability

- The system should be scalable to accommodate an increasing number of users and data.
- The system should be available for use during specified operating hours, with minimal downtime for maintenance or unexpected issues.
- Access to sensitive information and system functionalities should be restricted based on user roles and permissions.
- The user interface should maintain a consistent design and layout for a seamless user experience.
- The system should be intuitive, requiring minimal training for users to navigate and perform tasks effectively.
- The system should handle an increasing number of users and data without significant degradation in performance.
- The system should work seamlessly across different operating systems.
- The system's code should be well-documented and structured to facilitate future maintenance and updates.

These non-functional requirements contribute to the overall success and effectiveness of the student clinic system by addressing critical aspects beyond specific features. They guide the development and evaluation of the system in terms of performance, reliability, security, usability, and other important attributes.

## **2.3 ANALYSIS MODELS**

UML (Unified Modeling Language) is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems.

UML plays an important role in defining different perspectives of a system. These perspectives are design , Implementation ,process and deployment.

The center is the Use Case view which connects all these four. A Use Case represents the functionality of the system. Hence, other perspectives are connected with use case.

- **Design** of a system consists of classes, interfaces, and collaboration. UML provides class diagram, object diagram to support this.
- **Implementation** defines the components assembled together to make a complete physical system. UML component diagram is used to support the implementation perspective.
- **Process** defines the flow of the system. Hence, the same elements as used in Design are also used to support this perspective.
- **Deployment** represents the physical nodes of the system that forms the hardware. UML deployment diagram is used to support this perspective.

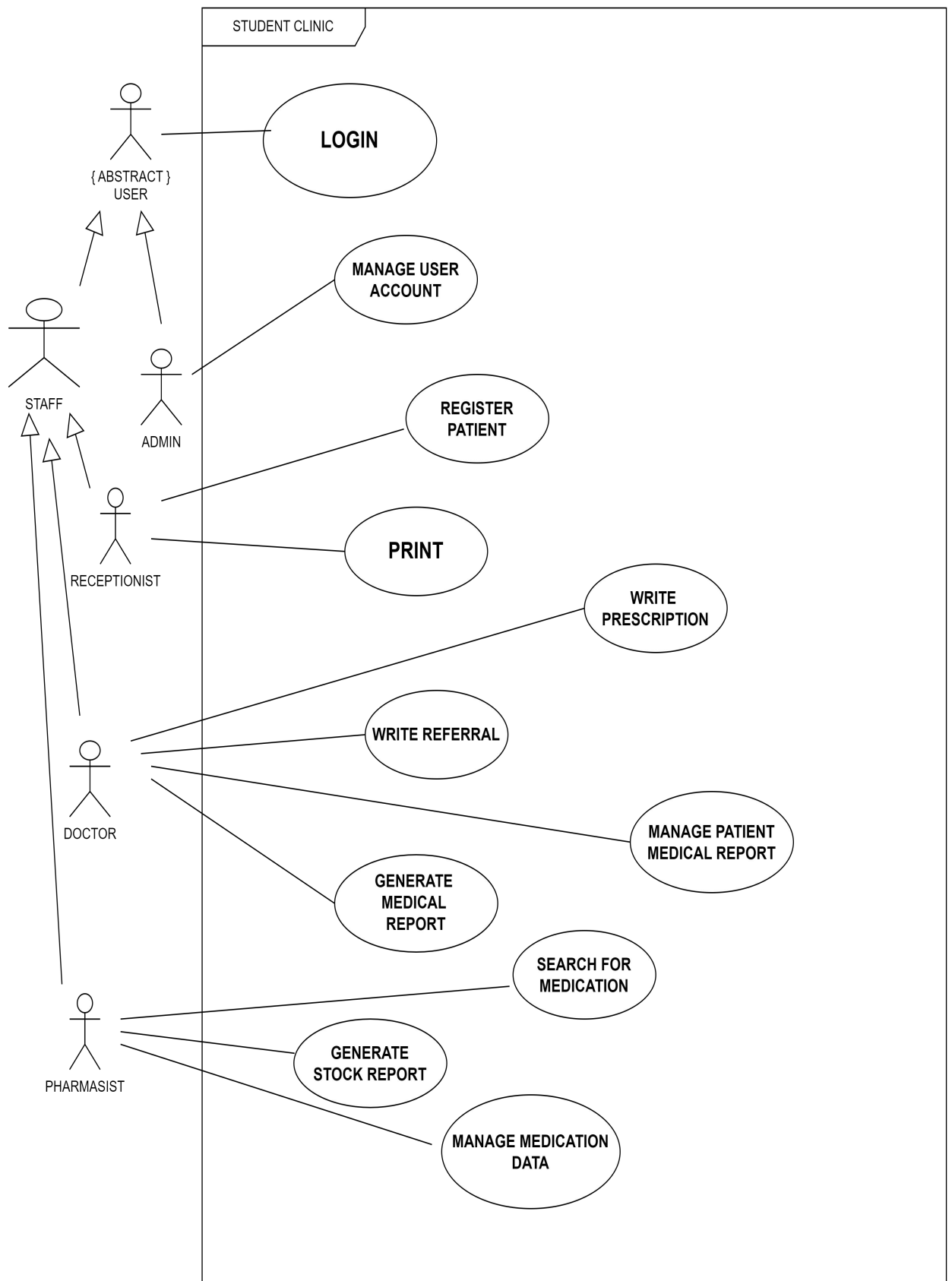
This portion of our project documentation will examine the three main categories of Unified Modeling Language (UML) diagrams that we will use to illustrate the functionality of our system. These diagrams are effective tools for demonstrating the dynamic features of our student clinic system. An overview of the system's interactions, showing the numerous roles and their related functionality, will be provided using the Use Case Diagram. The sequential representation of these interactions throughout time, which captures the dynamic flow of events, will be provided by the Sequence Diagram. The Activity Diagram, which offers information about the actions and decision points that characterize the functioning of our system, will, in the end, offer a complete visual representation of the workflow within particular processes.

### **2.3.2 System Use-Case Modeling**

To model a system, the most important aspect is to capture the dynamic behavior. Dynamic behavior means the behavior of the system when it is running/operating. Only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior. In UML, there are five diagrams available to model the dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature, there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. Use case diagrams consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application.

The Use Case Diagram for our student clinic system serves as a visual representation of the various interactions and relationships between system users (actors) and the core functionalities offered by the system. This diagram provides a high-level overview of the specific roles within the clinic, such as healthcare providers, administrators, and patients, and illustrates how these actors interact with the system to accomplish different tasks. By outlining these key user-system interactions, the Use Case Diagram offers a simplified yet comprehensive snapshot of the primary functionalities and user roles within our project, setting the stage for a more in-depth exploration of the system's dynamic behavior



## Usecase description

A generally recognized guideline for the length of use case descriptions is approx. 1–2 pages per use case. The following presents a structured approach for the description of use cases

- Name
- Short description
- Precondition: prerequisite for successful execution
- Postcondition: system state after successful execution
- Error situations: errors relevant to the problem domain
- System state on the occurrence of an error
- Actors that communicate with the use case
- Trigger: events which initiate/start the use case
- Standard process: individual steps to be taken
- Alternative processes: deviations from the standard process

### **1) Login**

Name	login
Short description	Users login using their username and id
Precondition	Users are registered into the system
Post condition	Users are logged in
Error situations	User entered incorrect username or password
System state on the occurrence of an error	Users are not logged in
Actors that communicate with the use case	Admin , doctor and pharmacist
Trigger	User requires to log in into the system
Standard process	<ul style="list-style-type: none"><li>● The actor initiates the login process by navigating to the login page.</li><li>● The actor provides their unique identification (ID) and corresponding password.</li><li>● The system conducts a validation check on the entered credentials to ensure accuracy.</li><li>● With valid credentials, the actor gains access to the system and is successfully logged in.</li><li>● The login use case concludes</li></ul>
Alternative processes	<ul style="list-style-type: none"><li>● The actor initiates the login process by navigating to the login page.</li><li>● The actor provides their unique identification (ID) and corresponding password.</li><li>● In the case of invalid credentials (incorrect ID or password), the system prompts an error message.</li><li>● The actor receives a notification about the invalid credentials and is redirected to the login page.</li><li>● Once the correct credentials are provided, the system proceeds with the standard login process, as described in the main flow.</li><li>● The actor is logged into the system with the correct credentials.</li><li>● The use case concludes</li></ul>

Name	Manage user account
Short description	Admin validates account created by the user
Precondition	The admin is logged into the system
Post condition	New user is created
Error situations	User is not a staff member
System state on the occurrence of an error	user not created
Actors that communicate with the use case	admin
Trigger	New user trying to sign in
Standard process	<ul style="list-style-type: none"> <li>● The user will enter id and password</li> <li>● The admin will give the valid user a verification code</li> <li>● The user then will complete the process by using the verification code</li> </ul>
Alternative processes	<ul style="list-style-type: none"> <li>● The user is not valid</li> <li>● Verification code will not be given</li> </ul>

Name	Register patient
Short description	The receptionist will create a record for new patient
Precondition	The receptionist is logged into the system The patient should be student of the campus
Post condition	New patient record is created
Error situations	User is not a student of the campus
System state on the occurrence of an error	Record is not created
Actors that communicate with the use case	receptionist
Trigger	When the receptionist try to create a new record
Standard process	<ul style="list-style-type: none"> <li>● The receptionist will enter student id</li> <li>● A new record is created</li> </ul>
Alternative processes	<ul style="list-style-type: none"> <li>● The record will not be created</li> </ul>

Name	Medical report
Short description	Doctors generate medical reports of the student
Precondition	The doctors is logged into the system
Post condition	Medical report about the student is generated
Error situations	Inaccurate information
System state on the occurrence of an error	Misleading report is generated
Actors that communicate with the use case	Doctor
Trigger	Doctor requires to generate medical report
Standard process	<ul style="list-style-type: none"> <li>● The doctor initiates the process by identifying the student for whom the medical information report needs to be generated.</li> <li>● The doctor accesses medical information database to retrieve the student's health information.</li> <li>● Report is generated</li> </ul>
Alternative processes	Faulty records are reviewed



Name	print
Short description	The receptionist will print the printable files
Precondition	The receptionist is logged into the system Printable files must be available
Post condition	The file is in printed form
Error situations	Insufficient resource
System state on the occurrence of an error	The file is not printed
Actors that communicate with the use case	receptionist
Trigger	When the receptionist try to print
Standard process	<ul style="list-style-type: none"> <li>● The receptionist will selected the file to be printed</li> <li>● The receptionist will enter the print command</li> <li>● The file will be printed</li> </ul>
Alternative processes	<ul style="list-style-type: none"> <li>● The receptionist will selected the file to be printed</li> <li>● The receptionist will enter the print command</li> <li>● Insufficient resource message will be displayed</li> <li>● The receptionist will provide the required resource</li> <li>● The receptionist will enter the print command again</li> <li>● The file will be printed</li> </ul>

Name	Write prescription
Short description	The doctor will write a prescription for the patient
Precondition	The doctor is logged into the system
Post condition	Prescription is written
Error situations	System crash
System state on the occurrence of an error	Prescription is not written
Actors that communicate with the use case	Doctor
Trigger	When the doctor try to write a prescription
Standard process	<ul style="list-style-type: none"> <li>● The doctor will enter hospital name</li> <li>● A referral is created</li> <li>● The referral is sent to the receptionist</li> </ul>
Alternative processes	Manually handled

Name	Write referral
Short description	The doctor will write a referral for the patient
Precondition	The doctor is logged into the system
Post condition	referral is written
Error situations	Invalid medication or dosage is written
System state on the occurrence of an error	Prescription is not written
Actors that communicate with the use case	Doctor
Trigger	When the doctor try to write a prescription
Standard process	<ul style="list-style-type: none"> <li>● The doctor will enter dosage and medication information</li> <li>● A prescription is created</li> </ul>

	<ul style="list-style-type: none"> <li>● Medical record is updated</li> <li>● The prescription is sent to the pharmacist</li> </ul>
Alternative processes	<ul style="list-style-type: none"> <li>● The doctor will enter dosage and medication information</li> <li>● The system will give the doctor a chance to edit the prescription</li> <li>● The corrected prescription will be sent to the pharmacist</li> </ul>

Name	Manage patient data
Short description	The doctor will update patient medical record
Precondition	The doctor is logged into the system
Post condition	Record is updated
Error situations	The doctor entered faulty information
System state on the occurrence of an error	Faulty data is recorded
Actors that communicate with the use case	doctor
Trigger	When the doctor try to update a record
Standard process	<ul style="list-style-type: none"> <li>● The doctor will enter id of the patient</li> <li>● The doctor will update patient data</li> <li>● The doctor will conform</li> <li>● Medical record is updated</li> </ul>
Alternative processes	<ul style="list-style-type: none"> <li>● The doctor will enter id of the patient</li> <li>● The system will give the doctor a chance to edit the medical record</li> <li>● The corrected data will be written to the database</li> </ul>

Name	Search for medication
Short description	The pharmacist will search for medication
Precondition	The pharmacist is logged into the system
Post condition	The medication is founded
Error situations	If the medication record doesn't exist
System state on the occurrence of an error	Drug does not exist
Actors that communicate with the use case	pharmacist
Trigger	When the pharmacist search for a drug
Standard process	<ul style="list-style-type: none"> <li>● The pharmacist will enter name of medication</li> <li>● The pharmacist will update the stock</li> <li>● stock record is updated</li> </ul>
Alternative processes	<ul style="list-style-type: none"> <li>● The pharmacist will enter name of medication</li> <li>● The medication is not found</li> <li>● The prescription is forwarded to the receptionist to be printed</li> </ul>

Name	Generate stock report
Short description	Pharmacists generate stock reports of the medications
Precondition	The pharmacist is logged into the system
Post condition	Stock report about the drugs is generated
Error situations	Inaccurate information
System state on the occurrence of an error	Misleading report is generated
Actors that communicate with the use case	pharmacist
Trigger	pharmacist requires to generate stock report
Standard process	<ul style="list-style-type: none"> <li>● The pharmacist initiates the process by identifying the criterion in which the report needs to be generated.</li> <li>● The pharmacist access medication information database to retrieve the medication stock information.</li> <li>● Report is generated</li> </ul>
Alternative processes	Faulty records are reviewed

Name	Manage medication data
Short description	The pharmacist will update medication stock record
Precondition	The pharmacist is logged into the system
Post condition	Record is updated
Error situations	The pharmacist entered faulty information
System state on the occurrence of an error	Faulty data is recorded
Actors that communicate with the use case	pharmacist
Trigger	When the pharmacist try to update a record
Standard process	<ul style="list-style-type: none"> <li>● The pharmacist will enter name of the drug</li> <li>● The pharmacist will update medication stock data</li> <li>● The pharmacist will confirm</li> <li>● Stock record is updated</li> </ul>
Alternative processes	<ul style="list-style-type: none"> <li>● The doctor will enter id of the patient</li> <li>● The system will give the pharmacist a chance to edit the stock record</li> <li>● The corrected data will be written to the database</li> </ul>

### 2.3.3 Sequence Diagram

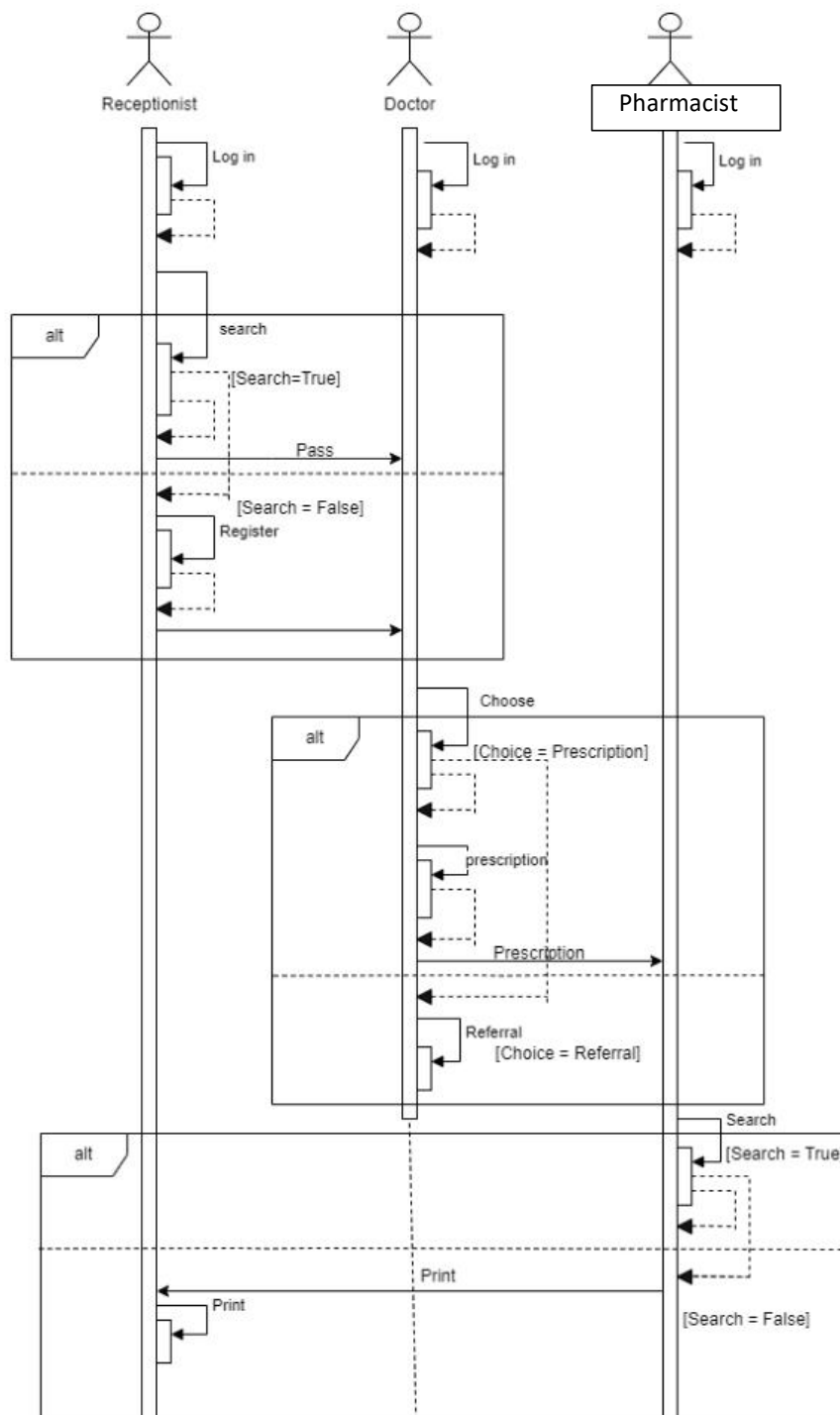
The sequence diagram captures the time sequence of the message flow from one object to another and the collaboration diagram describes the organization of objects in a system taking part in the message flow. Sequence diagrams are used to capture the order of messages flowing from one object to another

Following things are to be identified clearly before drawing the interaction diagram

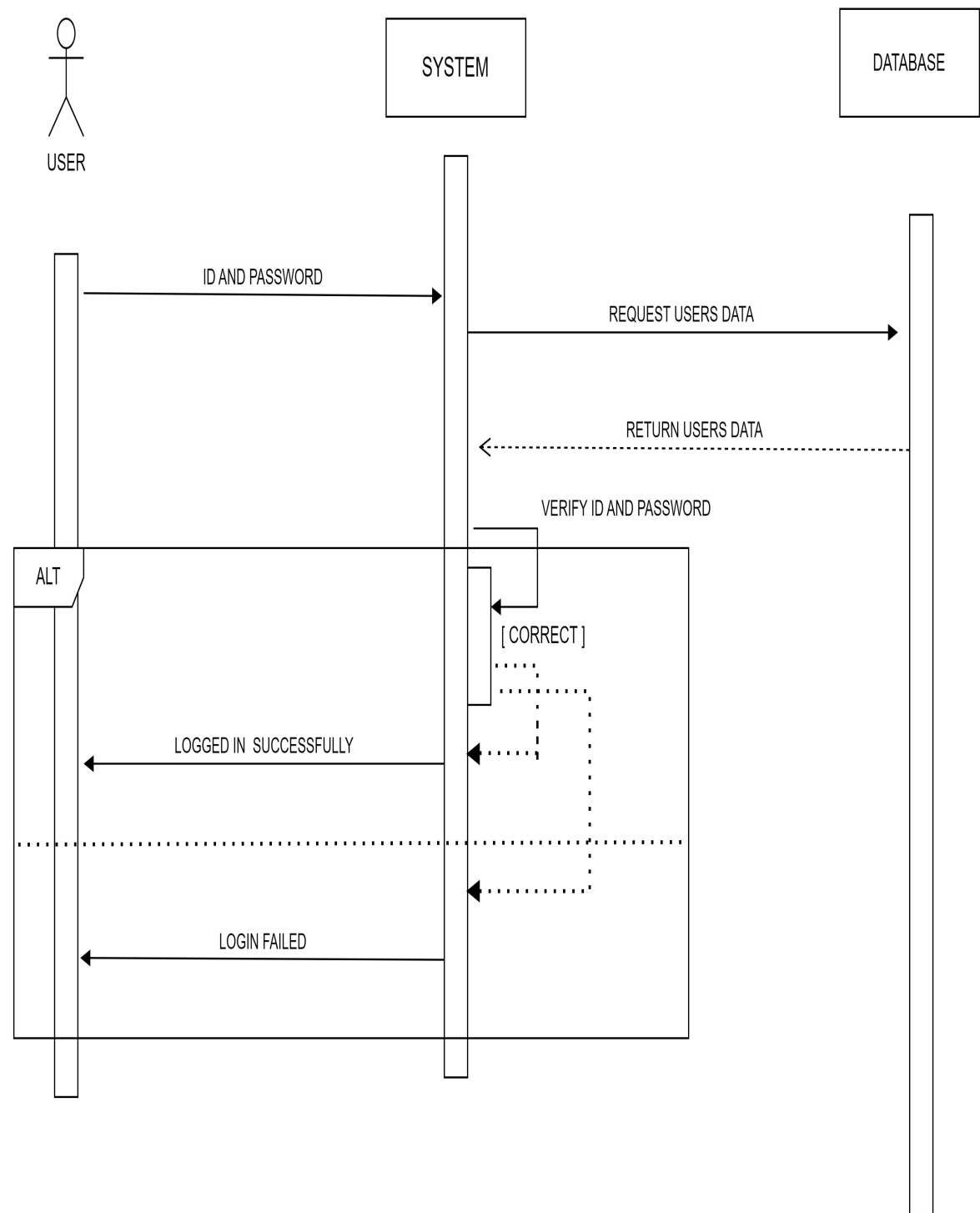
- Objects taking part in the interaction.
- Message flows among the objects.
- The sequence in which the messages are flowing.
- Object organization.

In our project there are three actors that participate in the sequential diagram

Actors	flow of messages and actions
Receptionist	<ul style="list-style-type: none"><li>● A receptionist should enter the system by using their username and password</li><li>● The receptionist will have the ability search for the patient using their ID</li><li>● The receptionist will let the patient see the doctor if the patient record is available.</li><li>● If not , the patient's medical record will be updated.</li></ul>
Doctor	<ul style="list-style-type: none"><li>● A doctor should enter the system by using their username and password</li><li>● The doctor will have the ability to choose from<ul style="list-style-type: none"><li>A) Write prescription</li><li>B) Write Referral to better health center</li></ul></li><li>● If the choice is alternative a the doctor will have the ability to write a prescription and forward it to the pharmacist</li><li>● Else if the choice is b the doctor will be able to write a referral and forward it to the receptionist to be printed.</li></ul>
Pharmacist	<ul style="list-style-type: none"><li>● A Pharmacist should enter the system by using their username and password</li><li>● The pharmacist will have the ability to search drugs in store that are prescribed by the doctor</li><li>● If the drug is available in store the patient will take the drug</li><li>● Else the prescription will be forwarded to the receptionist to be printed to buy the medication from external pharmacy</li></ul>



**LOGIN**



### **2.3.4 Activity Diagram**

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The basic purposes of activity diagrams is similar to the above diagrams. It captures the dynamic behavior of the system.

Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

The purpose of an activity diagram can be described as –

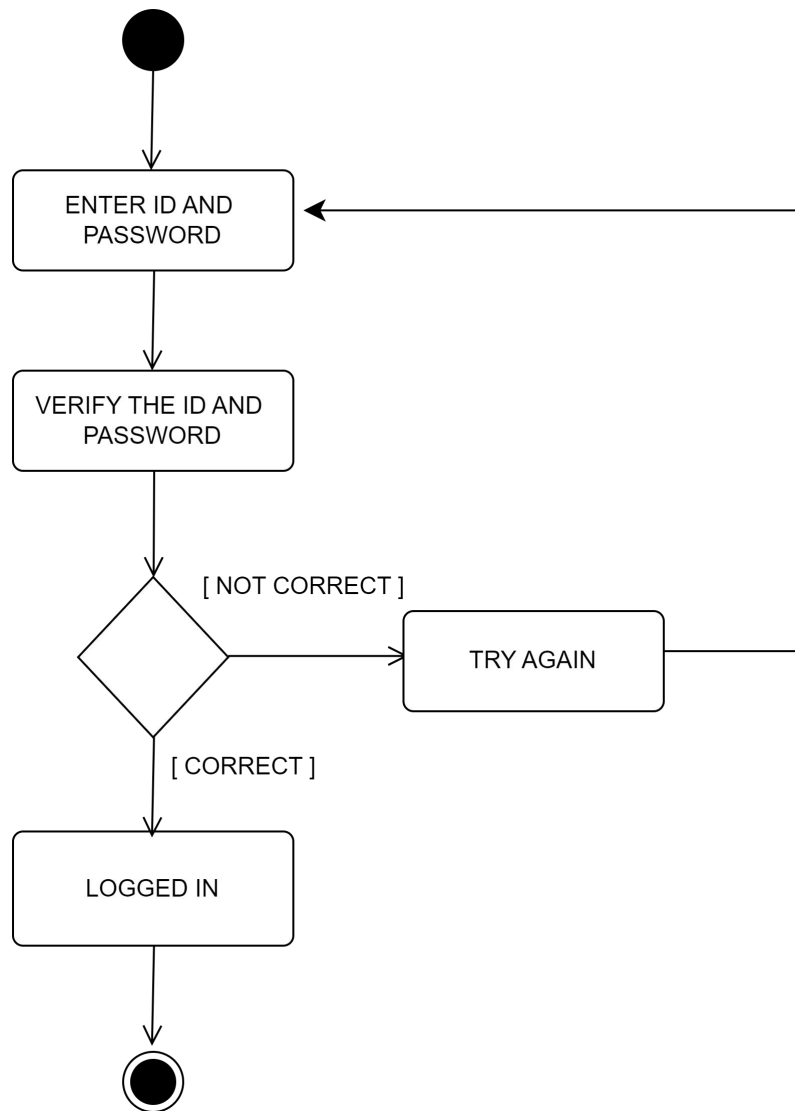
Draw the activity flow of a system.

Describe the sequence from one activity to another.

Describe the parallel, branched and concurrent flow of the system.

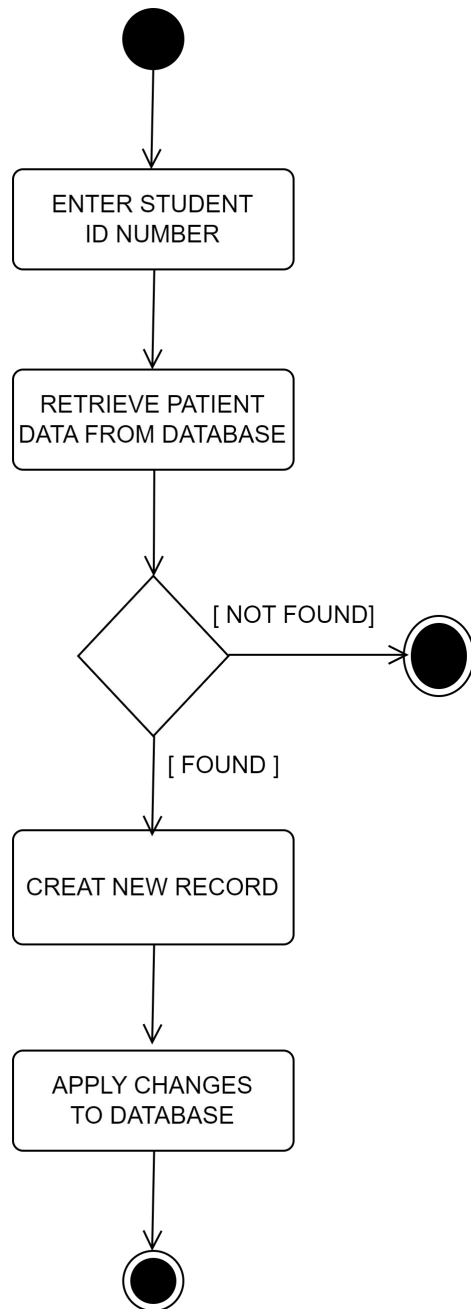
Activity	Flow
Login	<ul style="list-style-type: none"><li>● Enter id and password</li><li>● Verify the id and password</li><li>● If the id and password is correct the user will be logged in</li><li>● Else the user will try again</li></ul>
Registration	<ul style="list-style-type: none"><li>● Enter id</li><li>● The system will retrieve the from the database</li><li>● Id the id is valid a new record will be created for the student</li><li>● Else terminate the process</li></ul>
Medical information management	<ul style="list-style-type: none"><li>● Enter id</li><li>● Retrieve patient data from the database</li><li>● Apply changes to the record</li><li>● Confirm changes</li></ul>
Issue medication	<ul style="list-style-type: none"><li>● Enter drug name</li><li>● Retrieve data from the database</li><li>● if the drug is available the stock will be updated</li><li>● Else the prescription will be sent to the receptionist</li></ul>

## 1) LOGIN

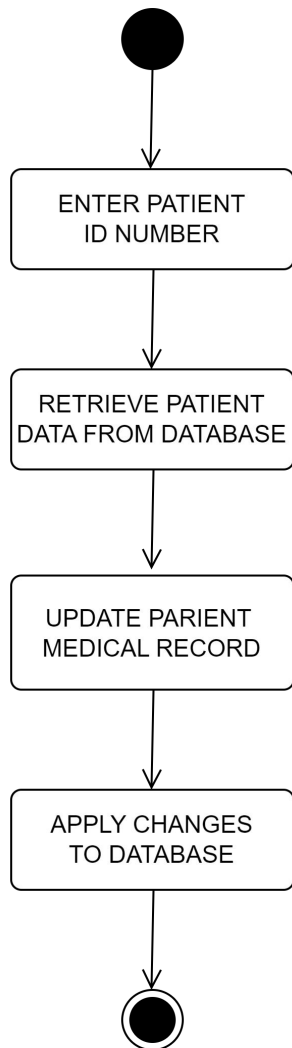




## 2) REGISTRATION



### 3) Medical information management



#### 4 Issue medication

