HOSPITAL MANAGEMENT SYSTEM

OOSE LAB FILE

HMS automation refers to use of computers, associated peripheral media such as Disks, Printer, Optical media etc. and utilization of computer based products and services in the performance of all type of Institutions functions and operations. Computers are capable of introducing a great deal of automation in operations, functions since they are electronic, programmable and are capable of control over the process being performed.

SUBMITTED BY:-

VARUN GUPTA	***SUSHIL MEHRA***
C123(C2)-04496402710	***C123(C2)-043***

INDEX

Sr. No.	Title	Date	Remarks

OBJECTIVES

The objective of this computerized product is to automate the HOSPITAL MANAGEMENT SYSTEM [HMS]. HMS automation refers to use of computers, associated peripheral media such as Disks, Printer, Optical media etc. and utilization of computer based products and services in the performance of all type of Institutions functions and operations. Computers are capable of introducing a great deal of automation in operations, functions since they are electronic, programmable and are capable of control over the process being performed.

IDENTIFICATION OF NEED

Existing System Details:-

In Hospital Management System, presently all Patient and staff management operations are being done manually. Various Books and Registers are maintained for entries about patient's and staff enquiry, registration, and fees submission. Final report preparation is very cumbersome and time consuming, as even for a single Record, several books have to be referred, in all immediate updating, validation, and reporting is just too large.

This result in unnecessary delay in various operation of organization and could be detrimental to the progress.

How Computerization Is Useful: -

The system has been felt to be computerization to make it more relevant and a real life project with minimal errors. It is therefore important to computerize it because there are many limitations in the manual system, which is more time consuming with many loopholes and the chance of errors, are apparent. Therefore a computerized system having an organized database cut redundancy and eases data access.

By this system, it becomes easier to get the updated information about any student such as fee deposited, different course offered and also for the management to have updated information of their students regarding fees, results etc. We can get a formatted and specialized printout of the output system, which is easy to keep and to be read to get the desired information. Due to the high capacity of computerized system, the user can store vast amount of information with easy access without time loss.

The diligent character of computers makes the system perfect, as it doesn’t suffer from human traits of tiredness and lack concentration. Thus by computerization this project software provides considerable help to the user by providing number checks to user so that he may get required feedback about the database. Computerization of this system would be time saving and the efficient handling of the database would prove to be an indispensable gift to the user in long run.

FEASIBILITY STUDY

The applications of technology to the Hospital Management allow producers to supply new and flexible services that are cost-competitive with conventional mass, standardized and rigidly packaged options. Technology gives Institute the flexibility to react to Patient demands.

Economical Feasibility:-

Our Software's Hospital Management System is to provide properties of all sizes with an affordable, effective, user-friendly Hospital Management System. The products are designed to control costs and resources, therefore improving long-term profitability and efficiency.

Our software is an affordable Hospital Management System software package that caters to small to medium-sized properties. It effectively manages Patient and staff Records, Registration Process and Staff records. A reliable answer to tracking availability and property statistics.

Operational Feasibility:-

Project is not rejected simply because of operational unfeasibility but such consideration is likely to critically affect the nature and scope of the eventual recommendations. As we know the users have very little knowledge of computer, a user-friendly environment will be required. The system should be GUI based. This goal can be met by using the Visual Basic as Front end and Oracle as Back end. A little consideration and training may enable the user to handle this package.

For Hospital Management System apart from other facts as per operational feasibility is concerned we need to see to it that the system to be developed should be user-friendly so that staff personnel's who are not computer literate find it easy to work with. As the office staff does most of the work, so they might have to be trained.

-

PROBLEM STATEMENT

The hospital needs an online system whereby it can accommodate the patients comfortably and avoid any confusion to the doctors regarding their work. There should be a system where the patients are categorized under insurance policy and non policy holders and the system helps the hospital management to claim the bills from concerned insurance companies.

The software system should be useful to record patient's details along with the compliant. It also should record the inpatient details and outpatient details and arrange the appointment of doctors. Should also provide the management reports like schedules, appointments of doctors, inpatients, insurances and discharges. And also generate bills dynamically for the discharged patients etc.

The administrative user should create new users and change their passwords. He shold be able to add the doctor's information as well as new doctor's details. Also can add information related to inpatients, outpatients, rooms availability, billing and insurance information. The administrator can view the management reports.

The doctors can change their own passwords. He can view his own appointments and information of inpatients and outpatients for any day.

Manager is another person who manages the activities in the hospital. He can add a new doctor to the list. He can also add new room information. He can view the information of doctors, rooms and patients.

An accountant can add the information related to patient insurances and view all the reports. He can view the details of inpatients, outpatients and discharged patient's information. He also collects the bill amount from the patient and enters it into the system.

SRS(Software requirements specification)

1 INTRODUCTION

1.1 PURPOSE

The Software Requirement Specification document describes requirements and functionality of the system. This document follows the IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1998). The intended addressees for this SRS are project examiners (PIEAS) and supervisor (PIEAS).

Requirement Analysis is the first technical step in the software process . A general statement of software scope is refined into a concrete pattern that becomes the basis for all software engineering activities that follows.

1.1.1 INTRODUCTION TO HOSPITAL MANAGEMENT SYSTEM

This document describes the requirement specification for the software HMS for hospital's medical department. HMS will mainly compose of four modules.

- Doctors
- Patients
- Laboratory
- Medical store

HMS for the trust will be a high performance database system designed purposely for keeping records of patients, doctors, laboratory test results for each patient and medicine stock in medical store and medicine prescribed to each patient.

Doctors section deals with the hiring and record keeping of experienced doctors and physicians of different specialties. It keeps a record of each and every doctor and also that of every checkup of various patients. Patient section deals with the registration and checkups of different patients.

The laboratory reports are generated related to tests done, their results and doctor s remarks and precautions.

The acquiring of medicine system keeps a record of all the acquiring whether purchases or donations, and also that of medicine stock in the medical store. The medical store s stock is updated when new medicine is purchased or when it is prescribed and delivered to the patients.

1.1.2 CURRENT SYSTEM

It is an OPD hospital dealing with following major departments.

- Reception registration and record office Doctor s examination rooms
- Skin Clinic
- Laboratory
- X-ray
- ECG
- Operation theatre (Eye) Recovery rooms Injection rooms Dispensary

They maintained almost all of their records in local registers and cards which had to be replaced by new ones due to shortage of space and hectic maintenance procedures.

1.1.3 PROBLEMS WITH THE EXISTING SYSTEM

Since all the information is maintained in the registers so searching a piece of data from those registers is definitely a time consuming and hectic task.

Every time a patient visits a doctor, after the visits he has to first come back to the reception for entries of the treatment given, diagnosis and remarks, which of course wastes the time of the patients as well as this is extra work. On the other hand the doctor can directly make these entries himself if the system is computerized and if there is a PC in his office.

All entries are made manually which is of course a lengthy process. This again results in people standing and waiting for their turns. The main problem with the system is that the same data is being entered in the system again and again at different places by different people. This might result in the inconsistency of data because the data is being entered redundantly at many places, and inconsistency of the system can cause serious problems.

The medicines are to be properly documented and a proper DB would ensure that the usage of each of the medicines is according to the patients who have requested or what the doctor has ordered.

The medicines are to be properly documented and a proper DB would ensure that the usage of each of the medicines is according to the patients who have requested or what the doctor has ordered.

There is a lot data redundancy. Data has to be copied for use by different departments.

Data updating is a major hurdle in such an environment where there are many people that are to be serviced and to take care that no record is tempered.

Security provision is a serious issue relating to such a huge organization to make it work efficiently. Records related to each patient need to be kept safe.

Extracting useful information from the current system is a very difficult task.

A close analysis makes it very certain that the existing system is time consuming. There are greater chances of inconsistency of data and the frustration of the patients. So it can be said that the existing system is not very reliable and needs to be changed to give the patients and the doctors and the staff the higher degree of satisfaction.

Keeping in view all the problems mentioned above, we hereby propose, a new and a computerized system of hospital's medical wing, to meet all the requirements of patients as well as the staff.

1.2 SCOPE

The HMS for Hospital's is an application running in Windows environment. The clients of the application are the receptionists, doctors, laboratory attendants, medical store attendants and the system administrator. The hospital wants to automate the whole system and get rid of the manual procedures.

The product will provide following functions Maintain Database

- Patient Registration Forms
- Laboratory Report Generation
- Security Management

The targeted project takes in following aspects:

1.2.1 USER-FRIENDLY ENVIRONMENT

User-friendly graphical interfaces are provided so that the user of the system can interact with the system and enter data into the database. The interfaces get input from the user and save it to the database.

1.2.2 MINIMAL TIME CONSUMPTION

As dealing with a large number of patients, hospital has great need to make its system less time consuming. Presently, they are maintaining almost all of their records in local registers, which are to be replaced by new ones due to shortage of space and hectic maintenance procedures.

1.2.3 MAINTAINING THE DATABASE

The daily entries are recorded in the database. These are then accessed according to the requirements.

1.3 OVERVIEW

Rest of the document is structured as follows:

- Overall description of the project including product perspective, functions, user characteristics, constraints, assumptions and dependencies.
- Specific requirements (functional and non functional) for the development of software.

2 OVERALL DESCRIPTIONS

The section describes the common factors that affect the product and its requirements. It comprises of the following sub sections.

- Product Perspective
- Product Functions
- User Characteristics
- Constraints
- Assumptions and Dependencies

2.1 PRODUCT PERSPECTIVE

The project is to provide the hospital (OPD), under the supervision of Hospital’s, with a system with the help of which it can get rid of the manual procedures.

HMS will be designed to provide the Foundation with easy access to critical information there by enabling the management to take better medical stock s acquiring and can generate reports required by the hospital on daily and monthly basis. This HMS is a high performance database system designed specifically for keeping records for patients, doctors, laboratory test results for each patient, medicine stock in medical store and medicine prescribed to each patient.

2.1.1 USER INTERFACES

User Interfaces are provided so that the user of the system can interact with the system and enter data into the database. The interfaces that are provided are to get input from the user and save it to the database. These interfaces will help in report generation

2.1.2 SOFTWARE INTERFACE

Following software interfaces will be required for the system.

- Web server (IIS)
- Microsoft SQL server
- Windows Operating System
- Browser Software (Internet Explorer 6.0 or above)

2.1.3 COMMUNICATIONS INTERFACES

HMS will require an internal protocol for communication. As the protocols are automatically mounted on the windows operating system, there is no need to install them separately. The protocols needed are

- ➢ HTTP
- ➢ TCP/IP

2.1.4 MEMORY CONSTRAINTS

- ➢ **PRIMARY MEMORY**

  The primary memory required at the client end will be very small. As only internet explorer is the component that will be running at client side, estimated memory required will be 21,036 K Bytes. Recommended primary memory is 128-256 MB or more.

- ➢ **SECONDARY MEMORY**

  The required software on the server includes o Internet Explorer

  - o Visual Studio
  - o SQL Server 2000

  According to task manager, estimated memory for Internet Explorer is 16,132 Kbytes, for Visual Studio 2 MB, and for SQL Server 2000 estimation came out to be 13 MB. Memory required for the database will be decided after the deployment of the system.

2.2 PRODUCT FUNCTIONS

2.2.1 MAINTAIN DATABASE

The daily entries are recorded in the database. These are then accessed according to the requirements.

2.2.2 PATIENT REGISTRATION FORMS

To enter patients to the database there are Patient Registration Forms. The receptionist enters the data through these entry forms.

2.2.3 LABORATORY REPORT GENERATION

This is the important functionality of the system. It includes generating reports on every patient s laboratory test and its result.

2.2.4 SECURITY MANAGEMENT

For the security of the data there is a login system. Five logins are created. 1) Reception. 2) Doctor. 3) Administrator. 4) Medical Store. 5) Laboratory. The receptionist is only allowed to register the patients and doctors by using entry forms. The Doctor is allowed to access and update every patient s checkup data, prescribe medicine and see the Laboratory test reports. The Administrator has full rights to the whole system. The Medical Store attendant can only update the medicine record delivered to each patient upon the prescriptions of the doctor. The Laboratory attendant can only access and update the patients test results.

2.3 USER CHARACTERISTICS

The users of the system and their characteristics are as follows:

2.3.1 DOCTOR

The Doctor will access and update every patient s checkup data, prescribe medicine and see the Laboratory test reports.

2.3.2 RECEPTIONIST

The receptionist handles various enquiries about the patient's registration as well as Doctor Consultation and Scheduling, Doctor Consultancy Fees and Time Allocation [2].

- Doctor visit schedule
- Doctor Appointment Scheduling
- Enquiry of Patient
- Find History of Patient Enquired

2.3.3 System Administrator

System administrator will inspect the overall performance of the system. He will have the rights to grant permissions.

2.3.4 Medical store Attendant

He will be responsible for updating data about medical store which includes record keeping of medicine stock.

2.3.5 Lab Attendant

He will update data about the patient's tests and their results.

2.4 Constraints

- The user must be affiliated with Hospital. No user can login to the system without authentication.
- Every user must know how to use a personal computer.
- A monthly report of all processes must be generated and sent to the chairman of Hospital.

2.5 Assumptions and Dependencies

- The daily reports of the processes will be generated by applying queries on the database.
- The user must know how to interact with the system.
- Database administrator will be responsible for manipulating with the database. He will manage the fields of the database and make entries in the tables.

3 SPECIFIC REQUIREMENTS

The use case diagrams and the non functional requirements of system are as follows.

3.1 FUNCTIONAL REQUIREMENTS (FR)

The stake holders (actors) of system and their use case modeling are as follows .

- Doctor
- Receptionist
- Database Administrator
- Medical Store Attendant
- Lab Attendant

3.1.1 FR FOR THE DOCTOR

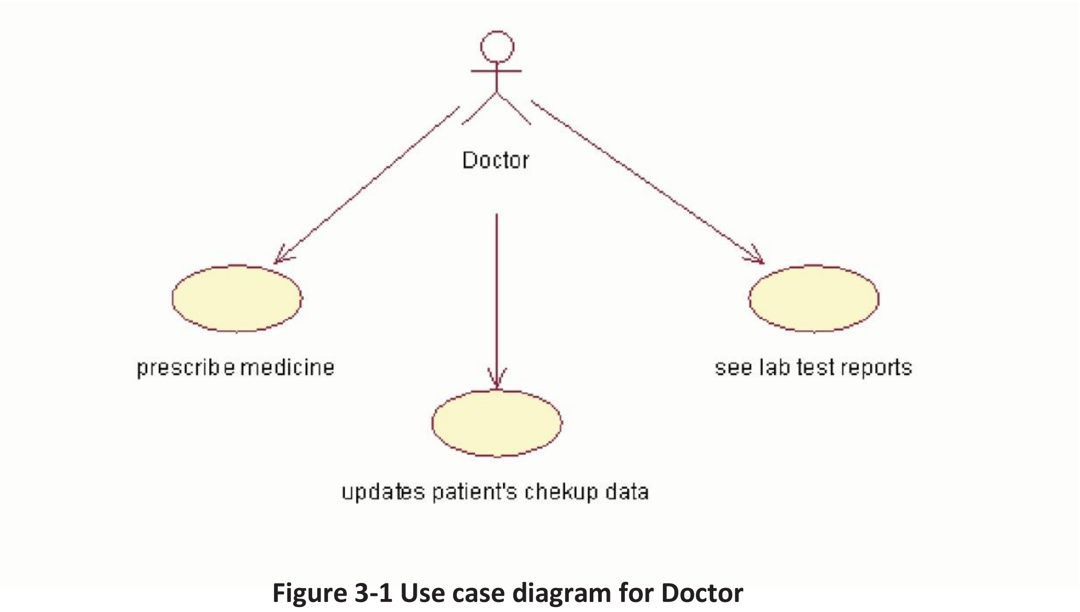


Figure 3-1 Use case diagram for Doctor

1. PRESCRIBE MEDICINE

   The doctor will prescribe the medicine to the patient which will be stored in the patient record update form.

2. UPDATES PATIENT S CHECKUP DATA

   This activity will let the doctor to update the patient s checkup information in the patient record.

3. SEE LAB REPORTS

   In this activity, the doctor will see the reports about the patient s tests and makes analysis from the report.

3.1.2 FR FOR THE RECEPTIONIST

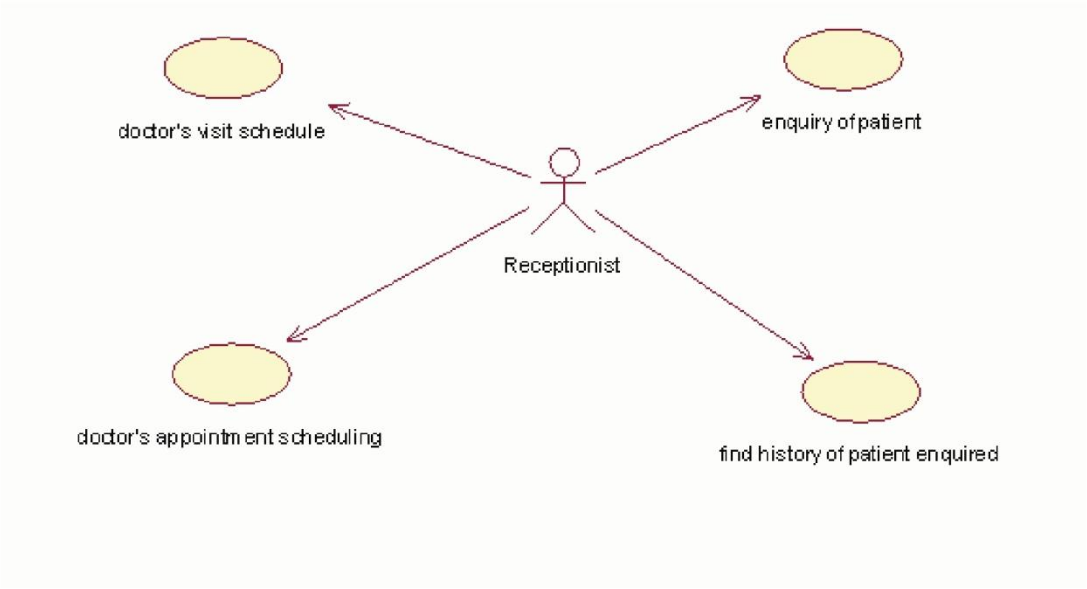

doctor's visit schedule
enquiry of patient
Receptionist
doctor's appointment scheduling
find history of patient enquired

Figure 3-2 Use Case Diagram for Receptionist

1. DOCTOR S APPOINTMENT SCHEDULING

   The receptionist will keep the record of patient s appointments with the doctors.

2. DOCTOR S VISIT SCHEDULE

   He will have the updated information about all the doctors schedule visiting or permanent.

3. Enquiry of Patient

   The receptionist will handle the various enquiries about the patient s registration.

Find History Of Patient Enquired

This activity will help receptionist to find the history of the patients who were enquired for their registration information.

3.1.3 FR for the System Administrator

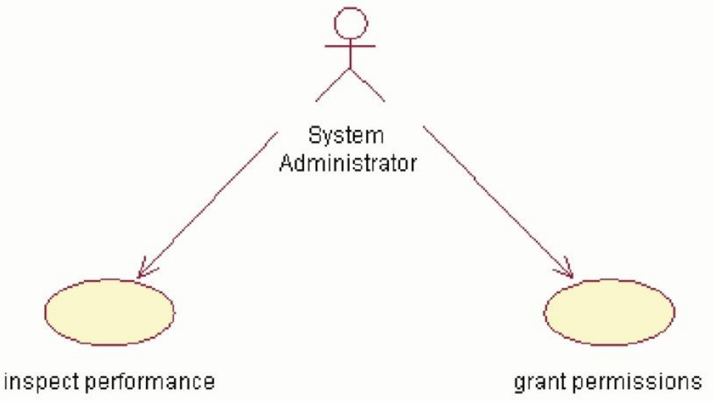


Figure 3-3 Use Case Diagram for System Administrator

1. Inspect Performance

   The system administrator will be responsible for the overall inspection of the system.

2. Grant Permissions

   The System Administrator has the rights to grant permissions to the respective users of the system.

3.1.4 FR for Medical Store Attendant

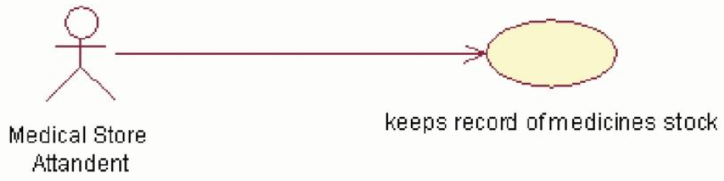


Figure 3-4 Use Case Diagram for Medical Store Attendant

1. KEEPS RECORD OF MEDICINE STOCK

The medical store attendant will keeps the record of the acquiring of the medicines whether purchase or donations, and also that of medicine stock in the medical store. The medical store s stock will be updated when new medicine is purchased or when it is prescribed and delivered to the patients.

3.1.5 FR FOR LAB ATTENDANT

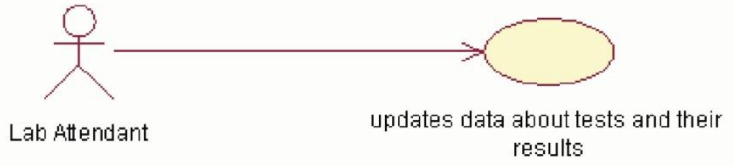


Figure 3-5 Use Case Diagram for Lab Attendants

1. UPDATES TEST S DATA

The Lab attendant will update the patient s tests data and their result in the database in order to generate reports. The reports generated will be related to the tests done, their results doctor s remarks and precautions.

3.2 Non Functional Requirements

The non-functional attributes of the project are illustrated under various sections below:

3.2.1 Maintainability

The system is developed in such a manner that its functionality can be enhanced to support further development in the system.

3.2.2 Usability

The system is designed to accept only registered user. Only five users are defined. 1) Reception. 2) Doctor (by name). 3) Administrator. 4) Medical Store. 5) Laboratory. The receptionist is only allowed to register the patients and doctors by using entry forms. The Doctor is allowed to access and update every patient s checkup data, prescribe medicine and see the *Laboratory test reports. The Administrator has full rights to the whole system. The Medical Store attendant can only update the medicine record delivered to each patient upon the prescriptions of the doctor. The Laboratory attendant can only access and update the patients test results.

3.2.3 Security

Security is provided in the system in a way that the different users will not be able to access everything in the system. The database will not be within everyone s reach.

3.2.4 Reliability

The system shall be reliable. If the server crashes, the data will not be lost because a backup will be maintained.

4 REFERENCES

[1] Outsource 2 India (O2I), “Software Services”, Bangalore, India,

URL: http://www.outsource2india.com/software/RequirementAnalysis.asp [2] XO Infotech Limited, “Hospital Management System”,

URL: http://www.xoinfotech.com/xohms2.htm#recep

[3] Simon Bennett, Steve McRobb and Ray Farmer, “*Object Oriented Systems*

Analysis and Design using UML”, 2/E, Mc Graw Hill, pp.134 – 144 (2001)

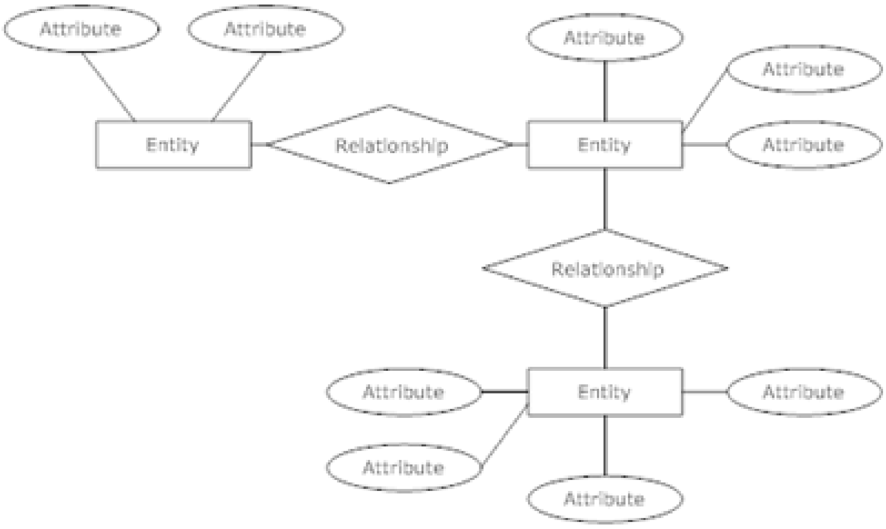
[4] “*IEEE Recommended Practice for Software Requirements Specifications*”, (IEEE

Std 830, 1998)

ENTITY RELATIONSHIP DIAGRAM (ER)

What are Entity Relationship Diagrams?

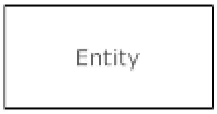
Entity Relationship Diagrams (ERDs) illustrate the logical structure of databases.



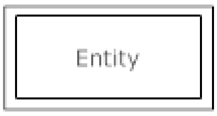
Entity Relationship Diagram Notations

Entity

An entity is an object or concept about which you want to store information.

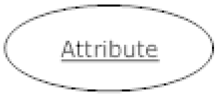


Weak Entity
A weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.



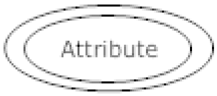
Key attribute

A key attribute is the unique, distinguishing characteristic of the entity. For example, an employee's social security number might be the employee's key attribute.



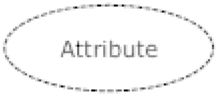
Multivalued attribute

A multivalued attribute can have more than one value. For example, an employee entity can have multiple skill values.



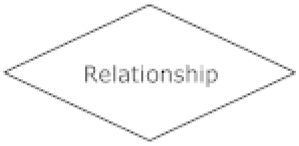
Derived attribute

A derived attribute is based on another attribute. For example, an employee's monthly salary is based on the employee's annual salary.



Relationships

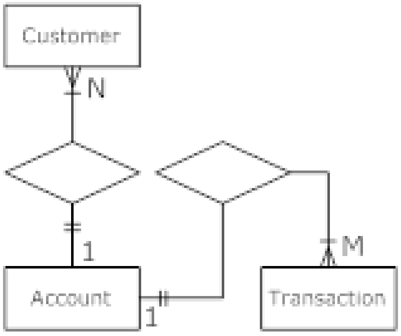
Relationships illustrate how two entities share information in the database structure.



Cardinality

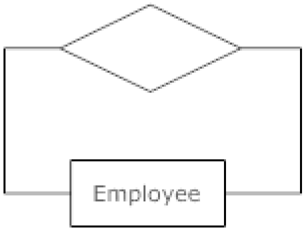
Cardinality specifies how many instances of an entity relate to one instance of another entity.

Ordinality is also closely linked to cardinality. While cardinality specifies the occurences of a relationship, ordinality describes the relationship as either mandatory or optional. In other words, cardinality specifies the maximum number of relationships and ordinality specifies the absolute minimum number of relationships.



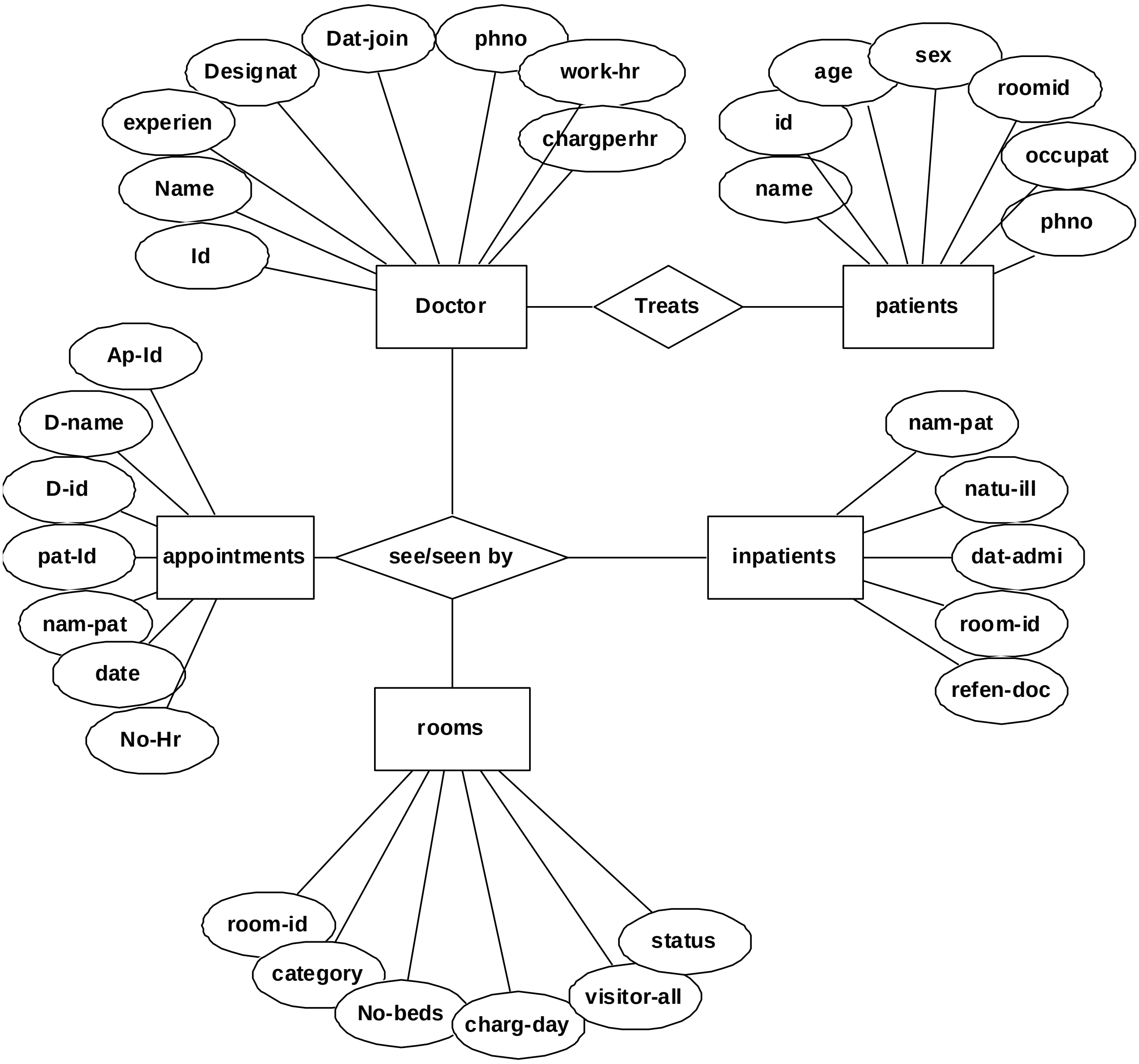
Recursive relationship

In some cases, entities can be self-linked. For example, employees can supervise other employees.



ENTITY RELATIONSHIP DIAGRAM

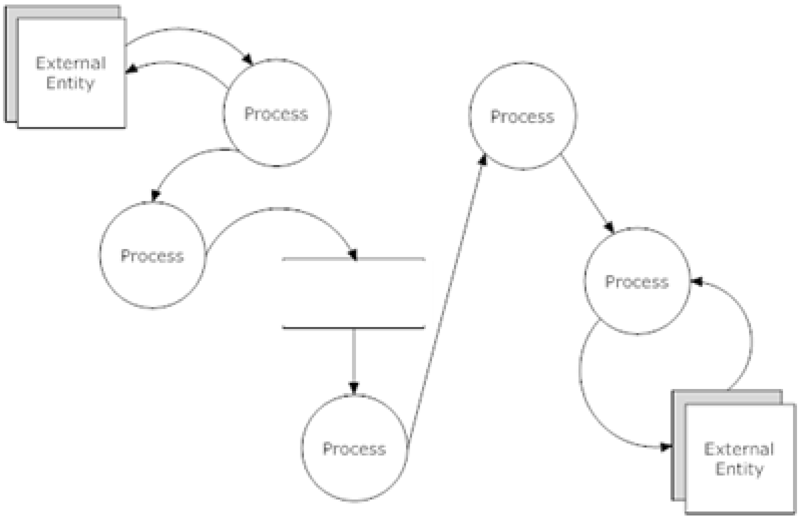
FOR HOSPITAL MANAGEMENT SYSTEM



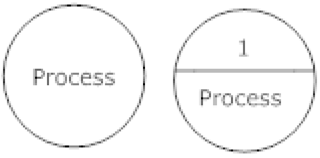
DATA FLOW DIAGRAMS (DFD)

What are Data Flow Diagrams?

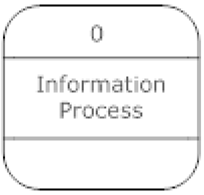
Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs.



Process
A process transforms incoming data flow into outgoing data flow.



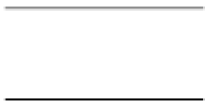
Yourdon and Coad Process Notations



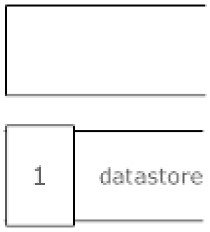
Gane and Sarson Process Notation

Datastore Notations

Datastores are repositories of data in the system. They are sometimes also referred to as files.



Yourdon and Coad Datastore Notations

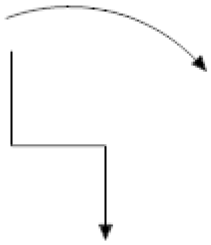


Gane and Sarson Datastore Notations

Dataflow Notations

Dataflow

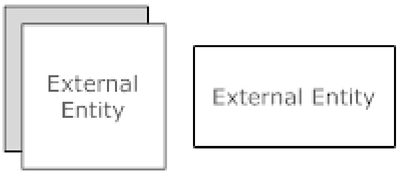
Dataflows are pipelines through which packets of information flow. Label the arrows with the name of the data that moves through it.



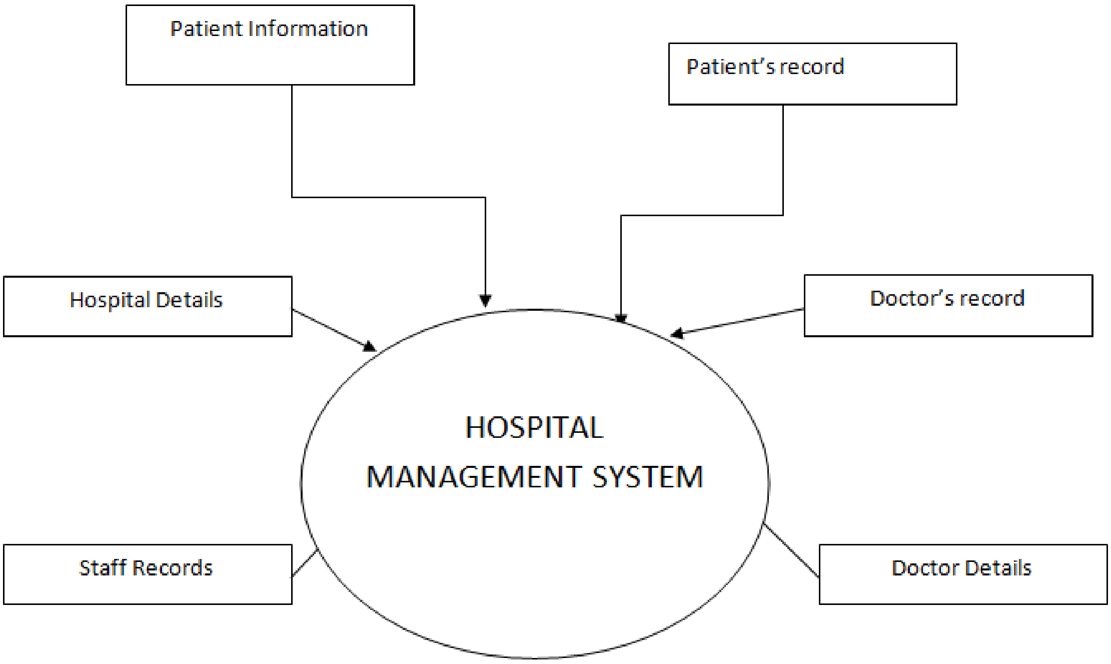
External Entity Notations

External Entity

External entities are objects outside the system, with which the system communicates. External entities are sources and destinations of the system's inputs and outputs.



0 level DFD/ context diagram



Level 1 DFD

USE CASE DIAGRAMS

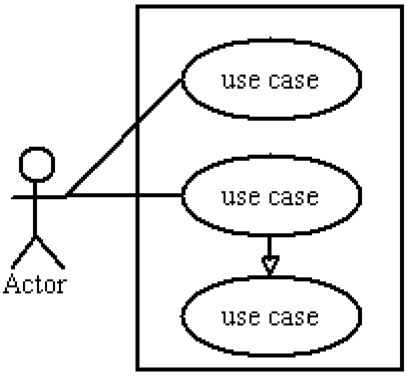
What is a UML Use Case Diagram?

Use case diagrams model the functionality of a system using actors and use cases. Use cases are services or functions provided by the system to its users.

Basic Use Case Diagram Symbols and Notations

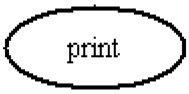
System

Draw your system's boundaries using a rectangle that contains use cases. Place actors outside the system's boundaries.



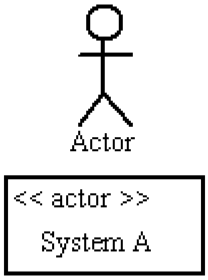
Use Case

Draw use cases using ovals. Label with ovals with verbs that represent the system's functions.

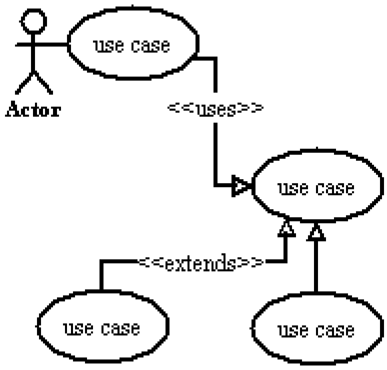


Actors

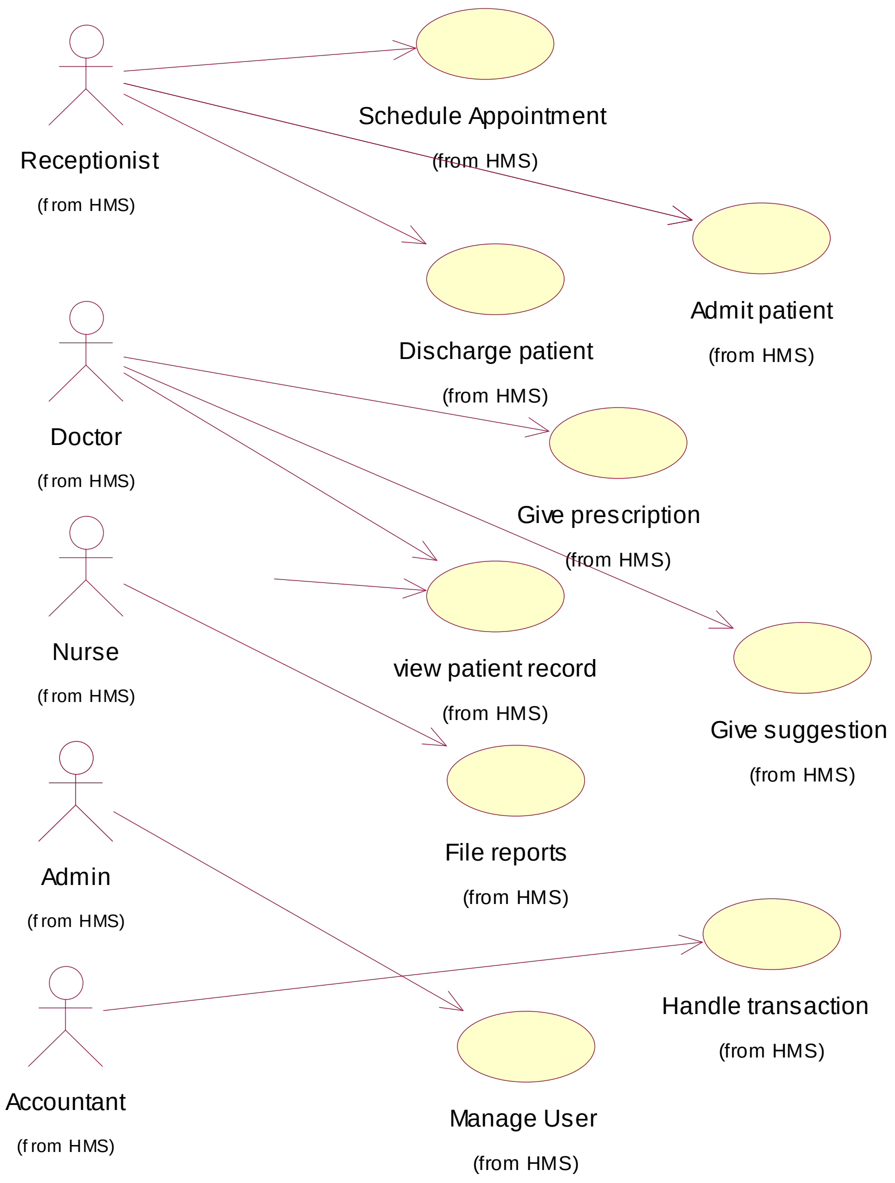
Actors are the users of a system. When one system is the actor of another system, label the actor system with the actor stereotype.



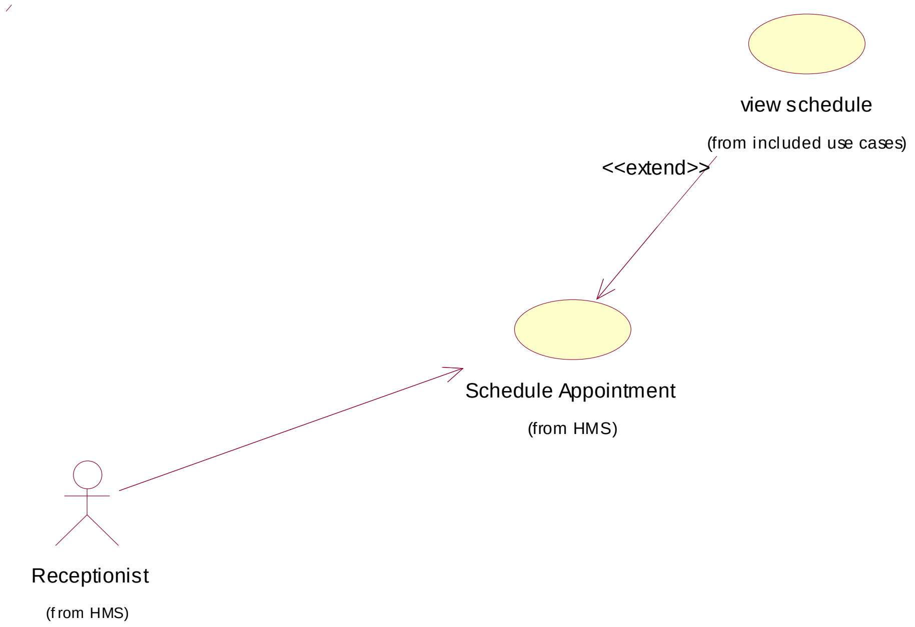
Relationships
Illustrate relationships between an actor and a use case with a simple line. For relationships among use cases, use arrows labeled either "uses" or "extends." A "uses" relationship indicates that one use case is needed by another in order to perform a task. An "extends" relationship indicates alternative options under a certain use case.



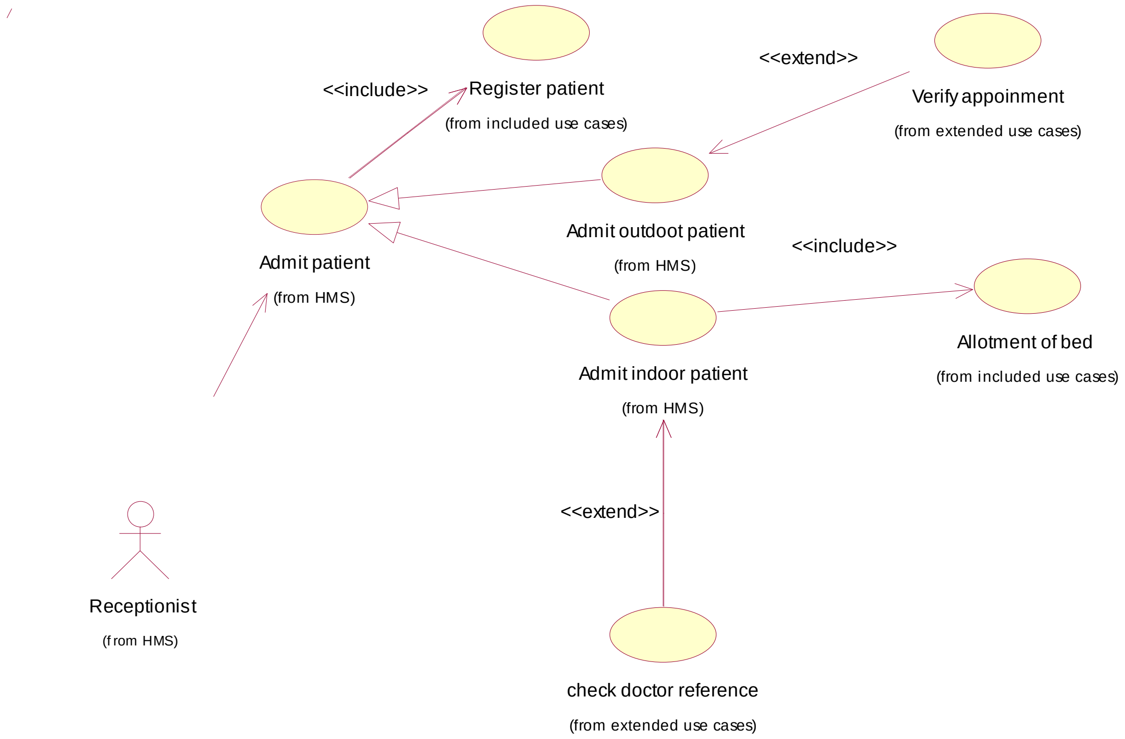
EXTENDED USE CASE DIAGRAM FOR HOSPITAL MANAGEMENT SYSTEM



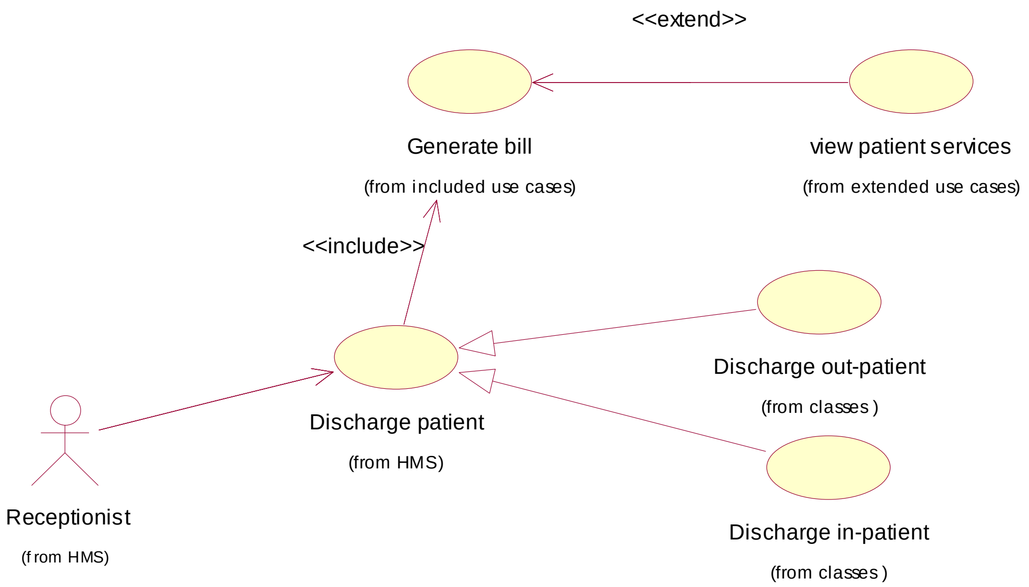
Schedule Appointment



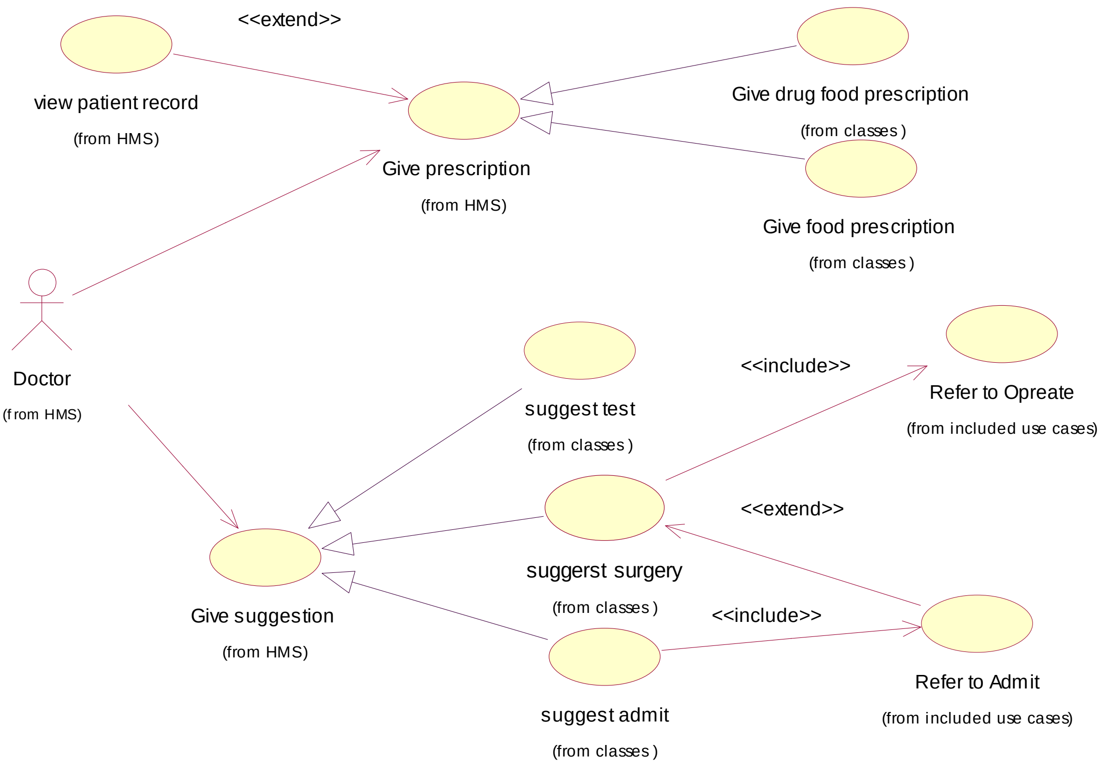
Admit patient



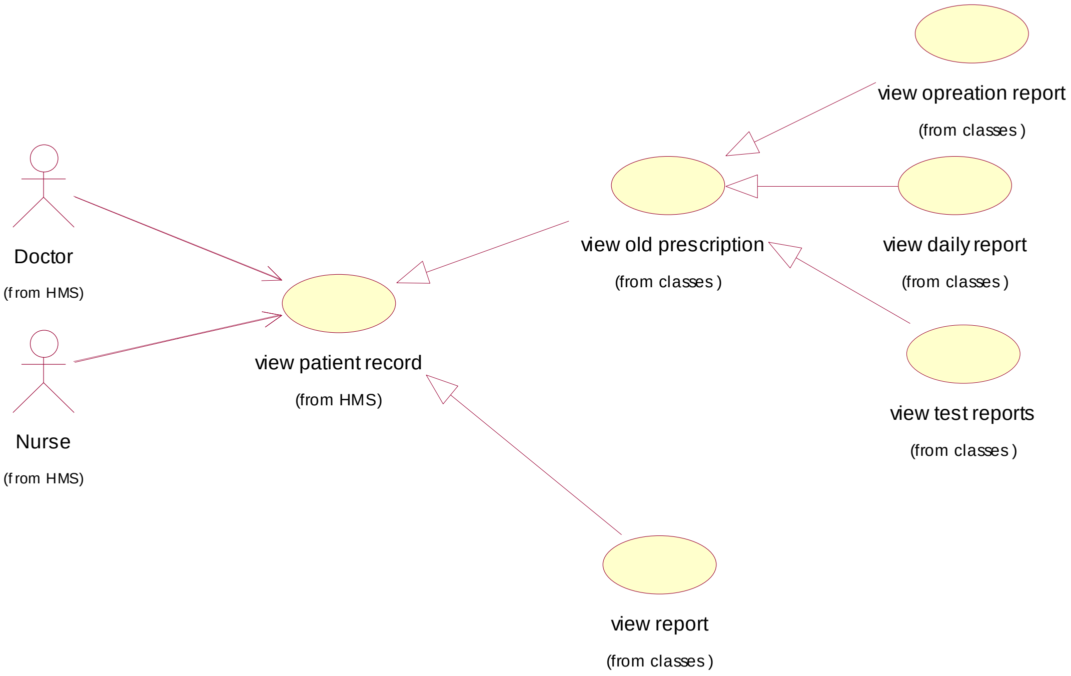
Discharge patient



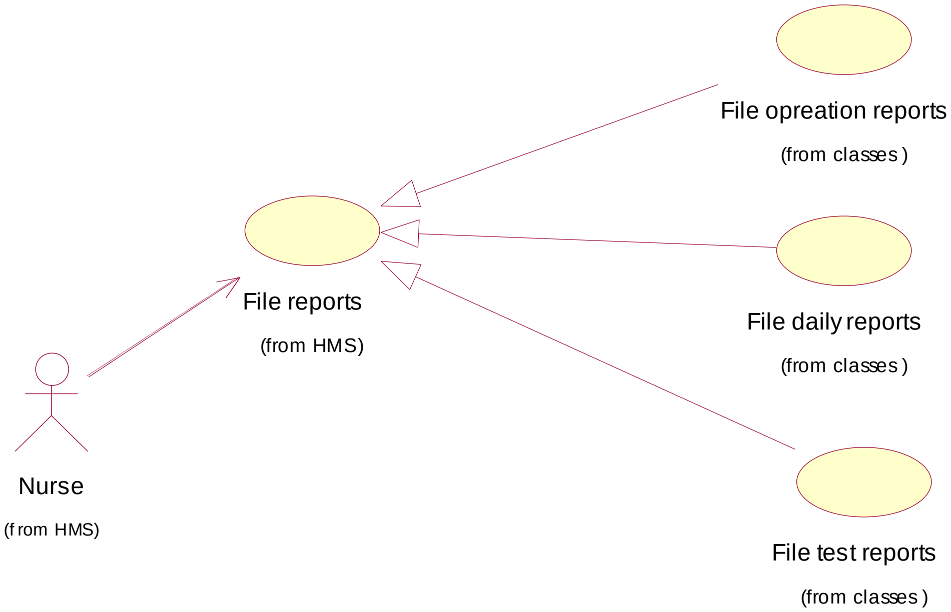
Give prescription & suggestion



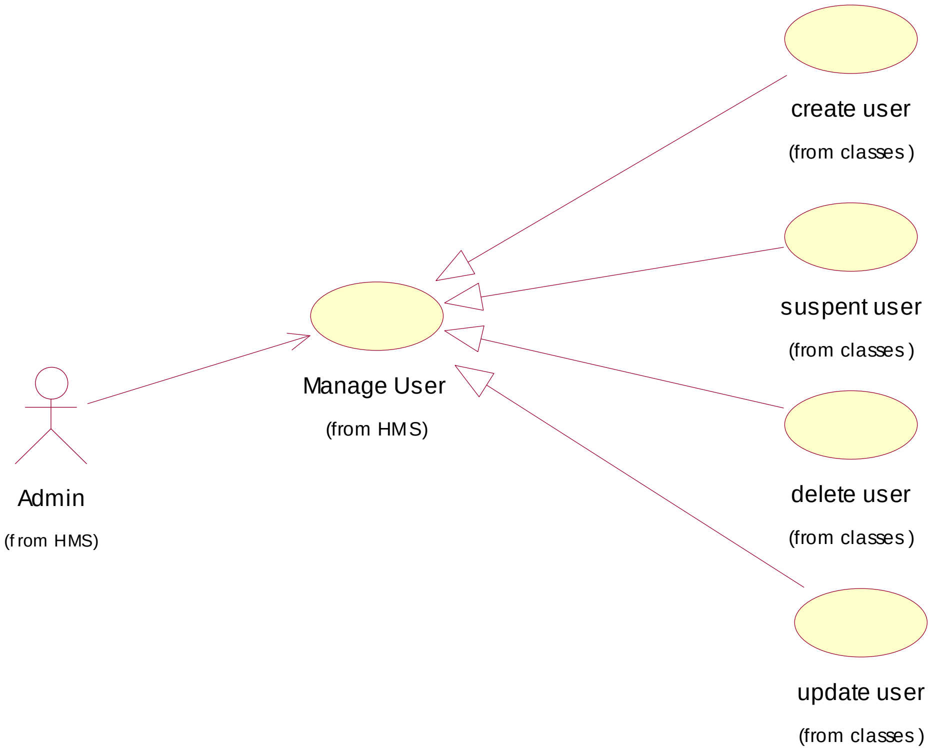
View patient record



File reports



Manage User


create user
(from classes)
suspent user
(from classes)
Manage User
(from HMS)
delete user
(from classes)
Admin
(from HMS)
update user
(from classes)

ACTIVITY DIAGRAMS

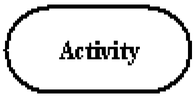
What is a UML Activity Diagram?

An activity diagram illustrates the dynamic nature of a system by modeling the flow of control from activity to activity. An activity represents an operation on some class in the system that results in a change in the state of the system. Typically, activity diagrams are used to model workflow or business processes and internal operation. Because an activity diagram is a special kind of statechart diagram, it uses some of the same modeling conventions.

Basic Activity Diagram Symbols and Notations

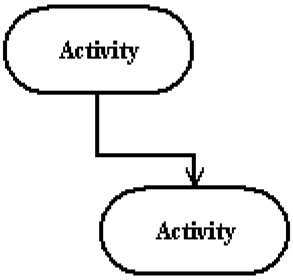
Action states

Action states represent the noninterruptible actions of objects.



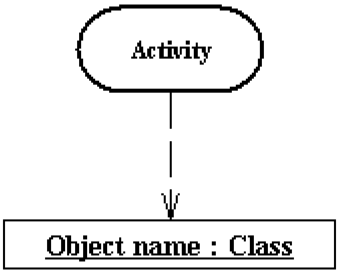
Action Flow

Action flow arrows illustrate the relationships among action states.



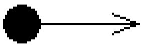
Object Flow

Object flow refers to the creation and modification of objects by activities. An object flow arrow from an action to an object means that the action creates or influences the object. An object flow arrow from an object to an action indicates that the action state uses the object.



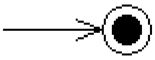
Initial State

A filled circle followed by an arrow represents the initial action state.



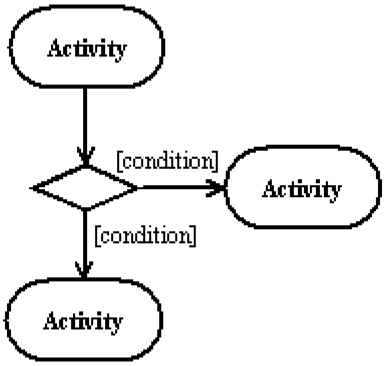
Final State

An arrow pointing to a filled circle nested inside another circle represents the final action state.



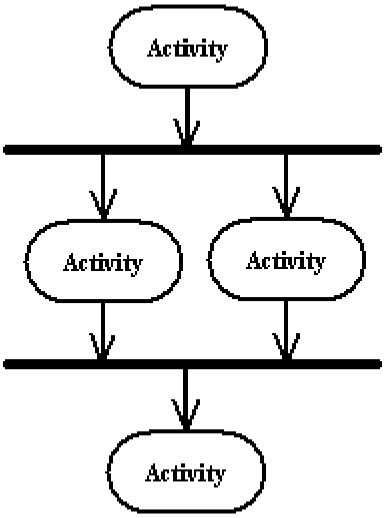
Branching

A diamond represents a decision with alternate paths. The outgoing alternates should be labeled with a condition or guard expression. You can also label one of the paths "else."



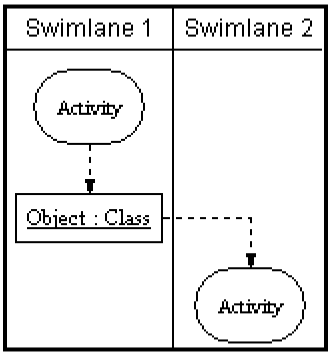
Synchronization

A synchronization bar helps illustrate parallel transitions. Synchronization is also called forking and joining.

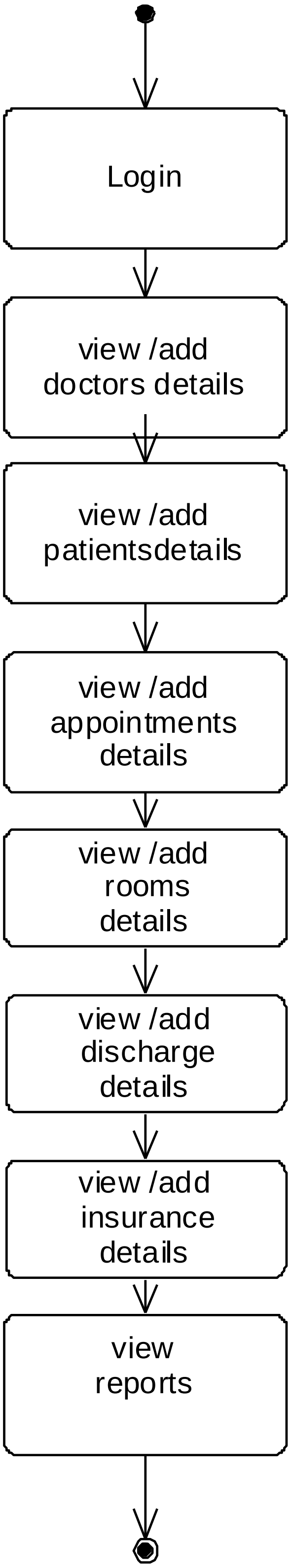


Swimlanes

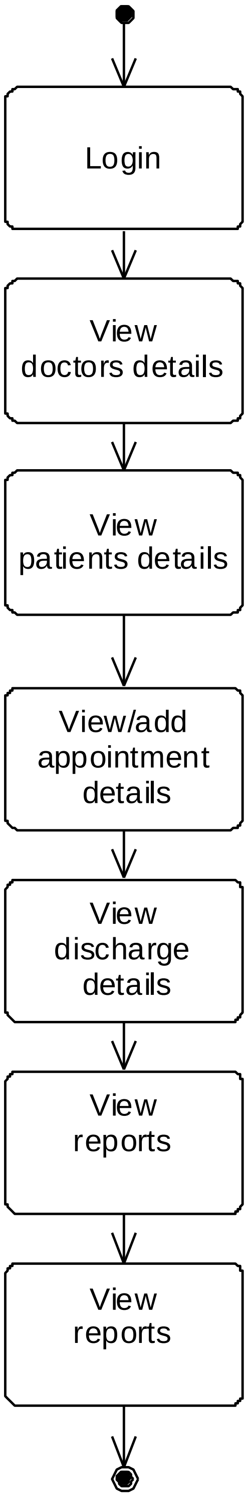
Swimlanes group related activities into one column.



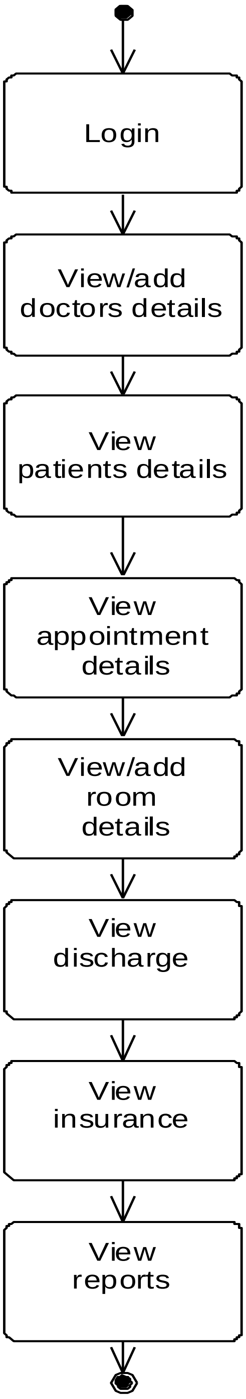
Activity diagram for Administrator:


Login
view /add doctors details
view /add patientsdetails
view /add appointments details
view /add rooms details
view /add discharge details
view /add insurance details
view reports

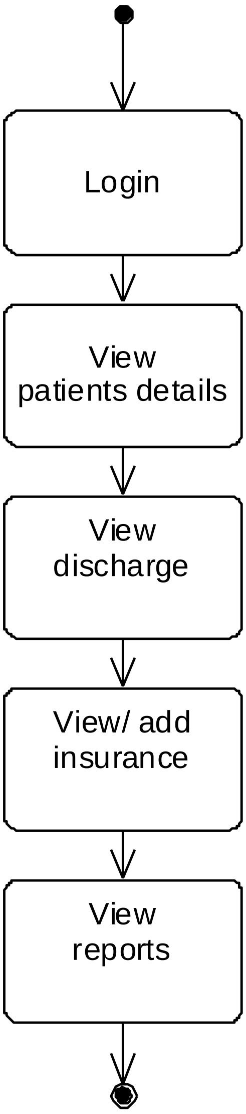
Activity diagram for Doctor:



Activity diagram for nurse:



Activity diagram for Accountant:



STATECHART DIAGRAMS

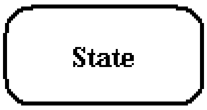
What is a UML Statechart Diagram?

A statechart diagram shows the behavior of classes in response to external stimuli. This diagram models the dynamic flow of control from state to state within a system.

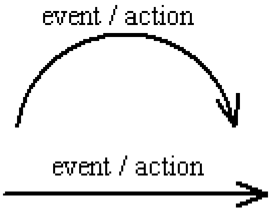
Basic Statechart Diagram Symbols and Notations

States

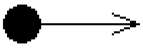
States represent situations during the life of an object. You can easily illustrate a state in SmartDraw by using a rectangle with rounded corners.



Transition
A solid arrow represents the path between different states of an object. Label the transition with the event that triggered it and the action that results from it.

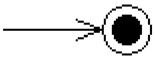


Initial State
A filled circle followed by an arrow represents the object's initial state.
Learn how to rotate objects.



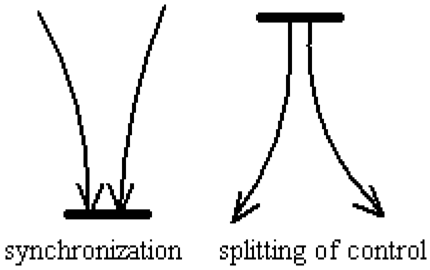
Final State

An arrow pointing to a filled circle nested inside another circle represents the object's final state.

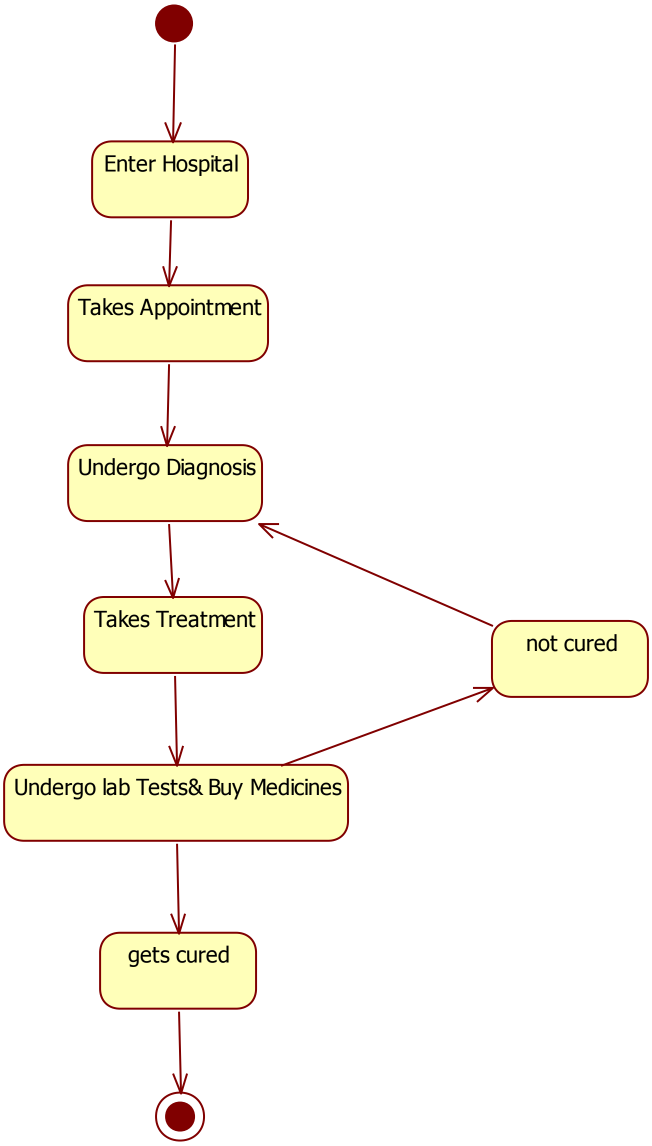


Synchronization and Splitting of Control

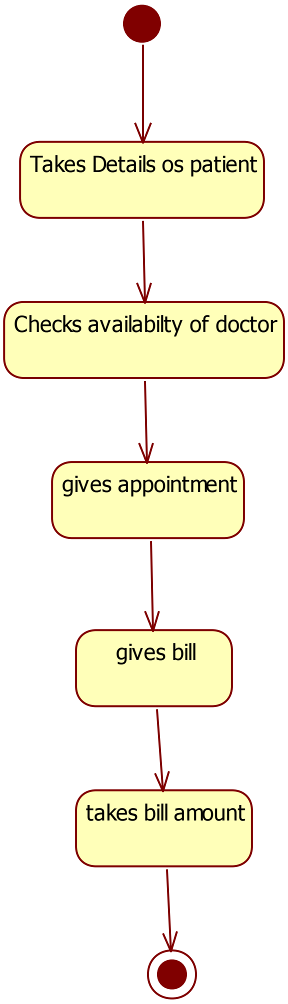
A short heavy bar with two transitions entering it represents a synchronization of control. A short heavy bar with two transitions leaving it represents a splitting of control that creates multiple states.



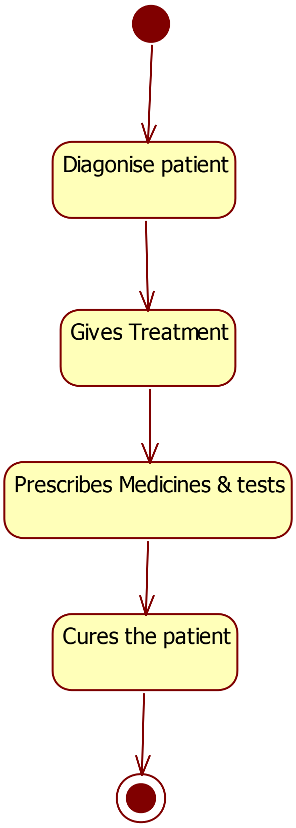
PATIENT:


Enter Hospital
Takes Appointment
Undergo Diagnosis
Takes Treatment
not cured
Undergo lab Tests& Buy Medicines
gets cured

RECPTIONIST:



DOCTOR:


Diagonise patient
Gives Treatment
Prescribes Medicines & tests
Cures the patient

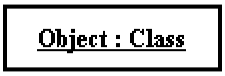
SEQUENCE DIAGRAM

What is a UML Sequence Diagram?

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time.

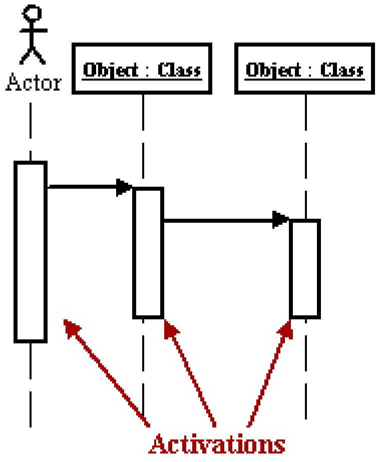
Basic Sequence Diagram Symbols and Notations

Class roles
Class roles describe the way an object will behave in context. Use the UML object symbol to illustrate class roles, but don't list object attributes.

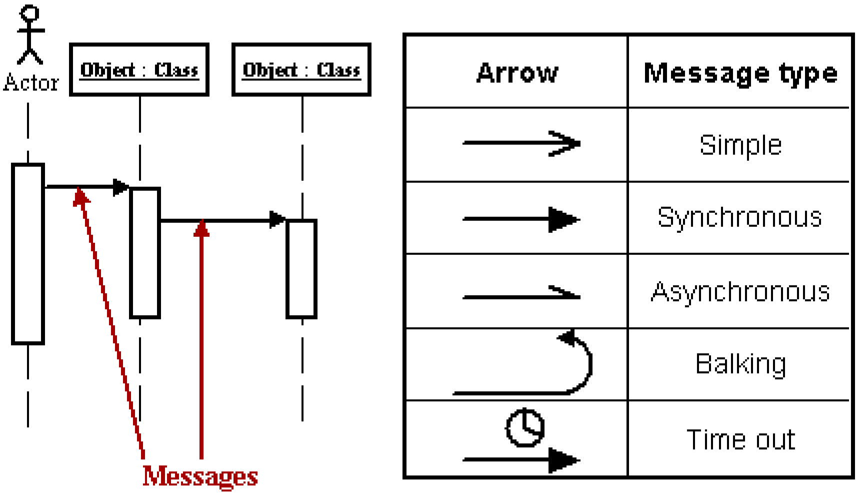


Activation

Activation boxes represent the time an object needs to complete a task.



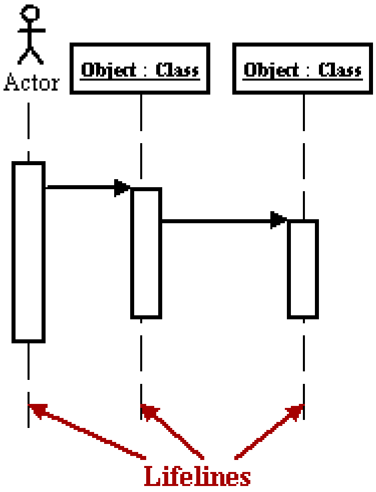
Messages
Messages are arrows that represent communication between objects. Use half-arrowed lines to represent asynchronous messages. Asynchronous messages are sent from an object that will not wait for a response from the receiver before continuing its tasks.



Arrow	Message type
→	Simple
▶	Synchronous
⇀	Asynchronous
↩	Balking
⏲ ▶	Time out

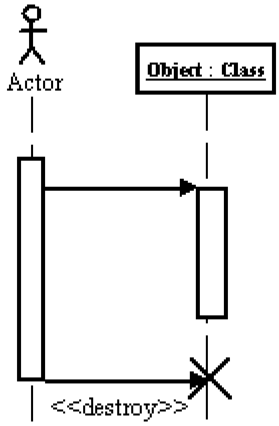
Various message types for Sequence and Collaboration diagrams

Lifelines
Lifelines are vertical dashed lines that indicate the object's presence over time.



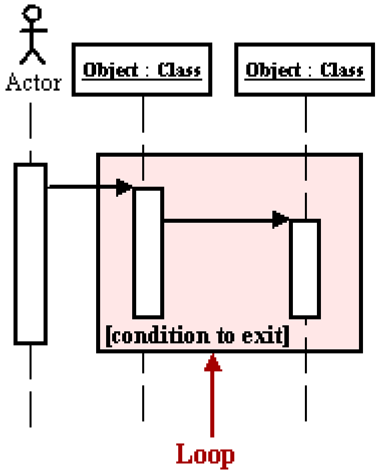
Destroying Objects

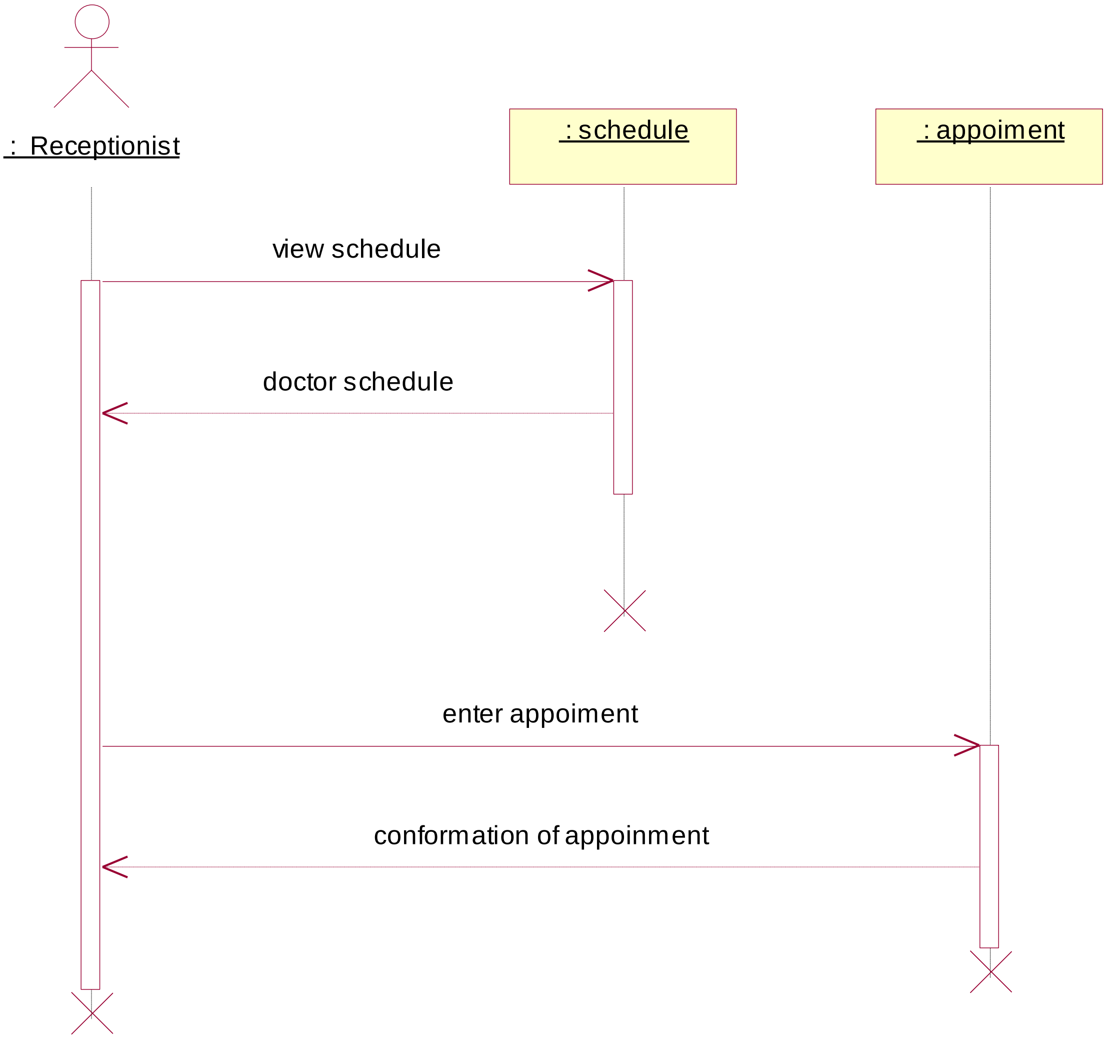
Objects can be terminated early using an arrow labeled "<< destroy >>" that points to an X.



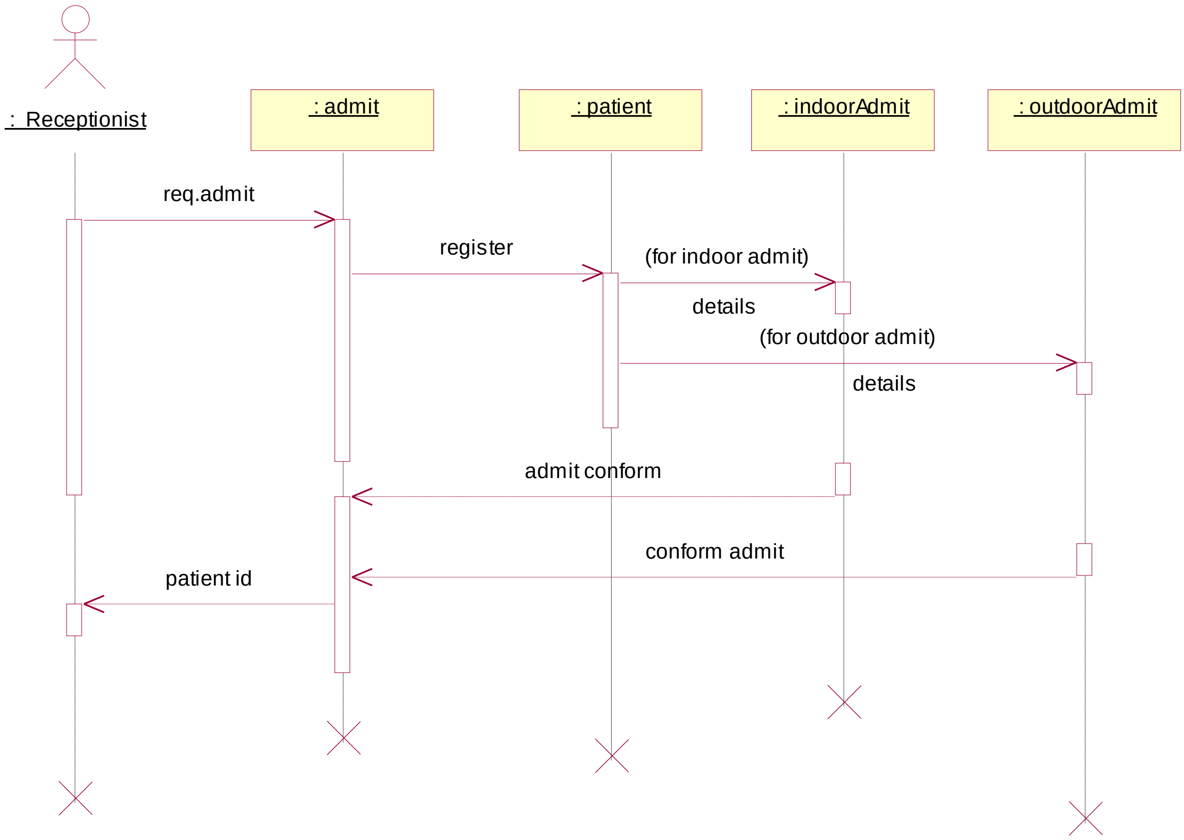
Loops

A repetition or loop within a sequence diagram is depicted as a rectangle. Place the condition for exiting the loop at the bottom left corner in square brackets [

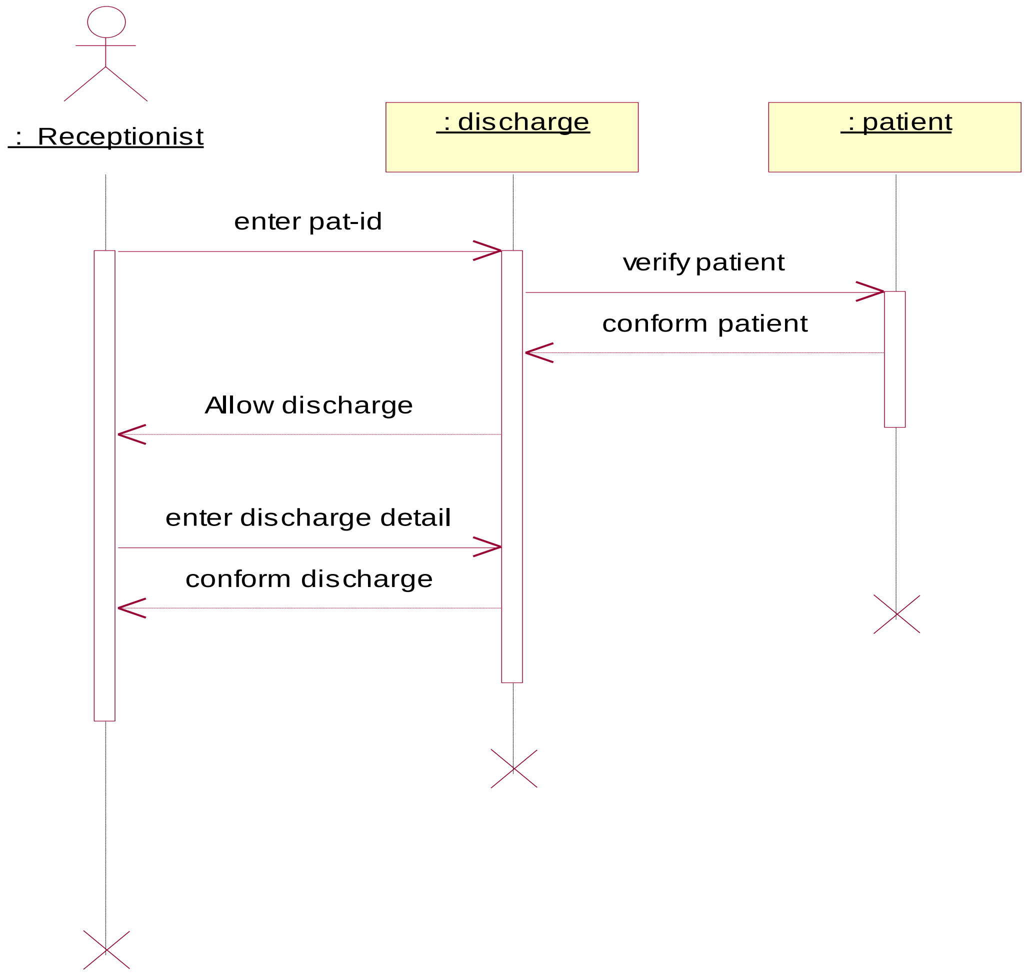



: Receptionist
: schedule
: appoiment
view schedule
doctor schedule
enter appoiment
conformation of appoinment

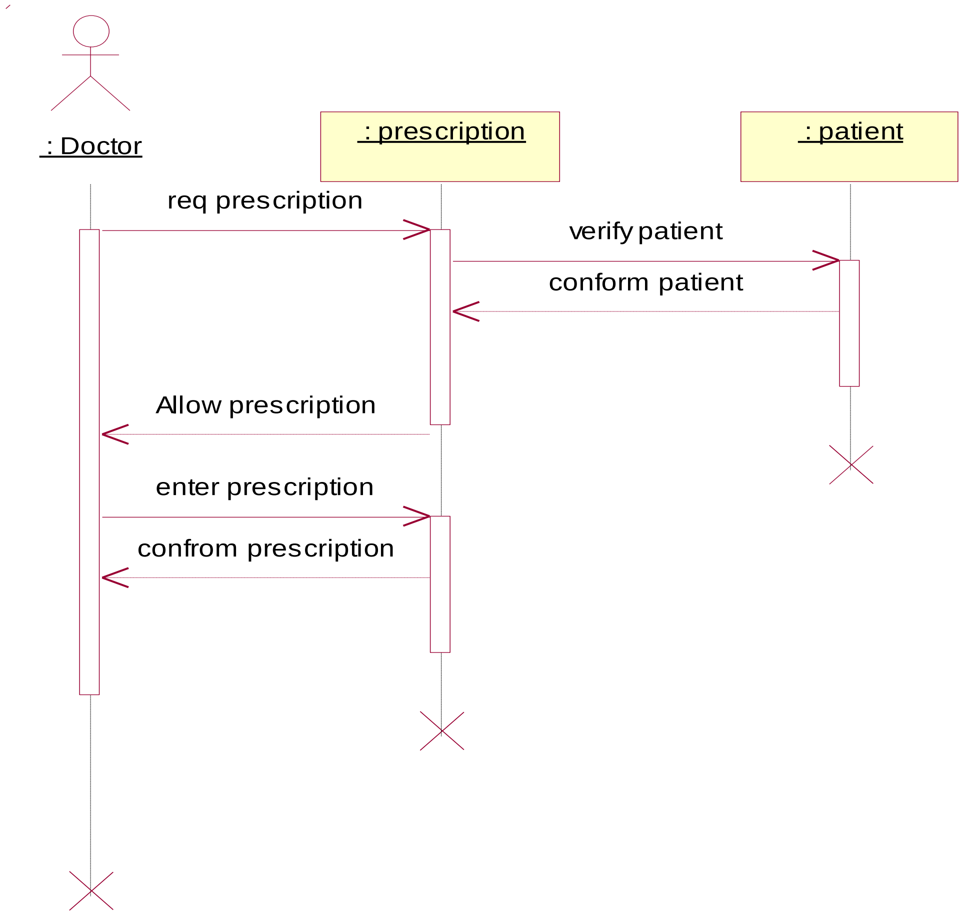
Admit patient



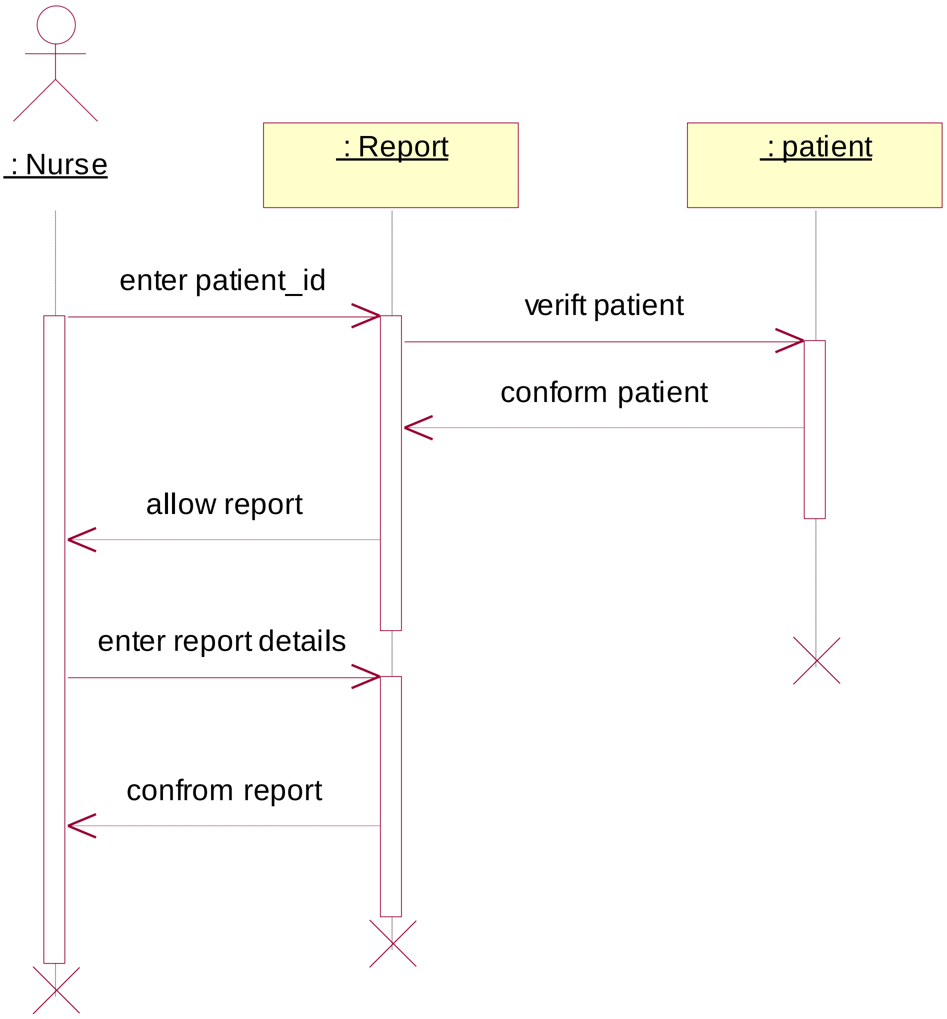
Discharge patient



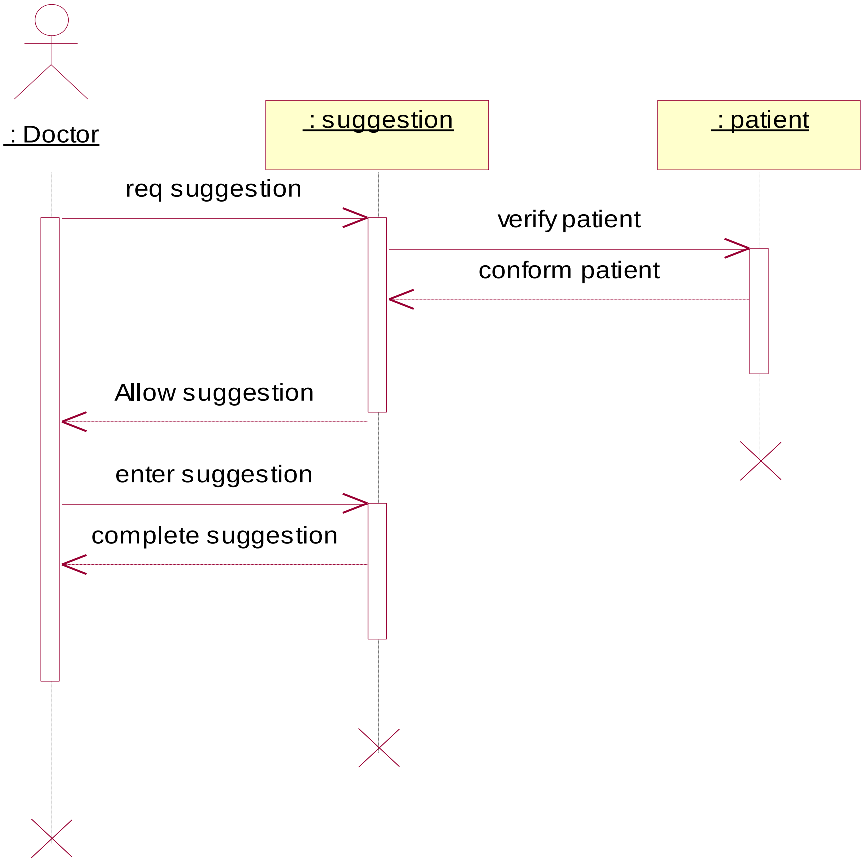
Give prescription



File reports



Give suggestion



COLLABORATION DIAGRAMS

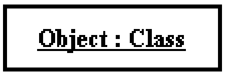
What is a UML Collaboration Diagram?

A collaboration diagram describes interactions among objects in terms of sequenced messages. Collaboration diagrams represent a combination of information taken from class, sequence, and use case diagrams describing both the static structure and dynamic behavior of a system.

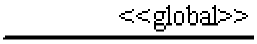
Basic Collaboration Diagram Symbols and Notations

Class roles

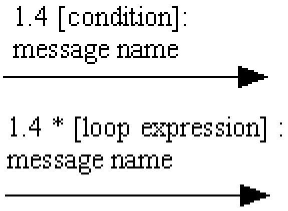
Class roles describe how objects behave. Use the UML object symbol to illustrate class roles, but don't list object attributes.



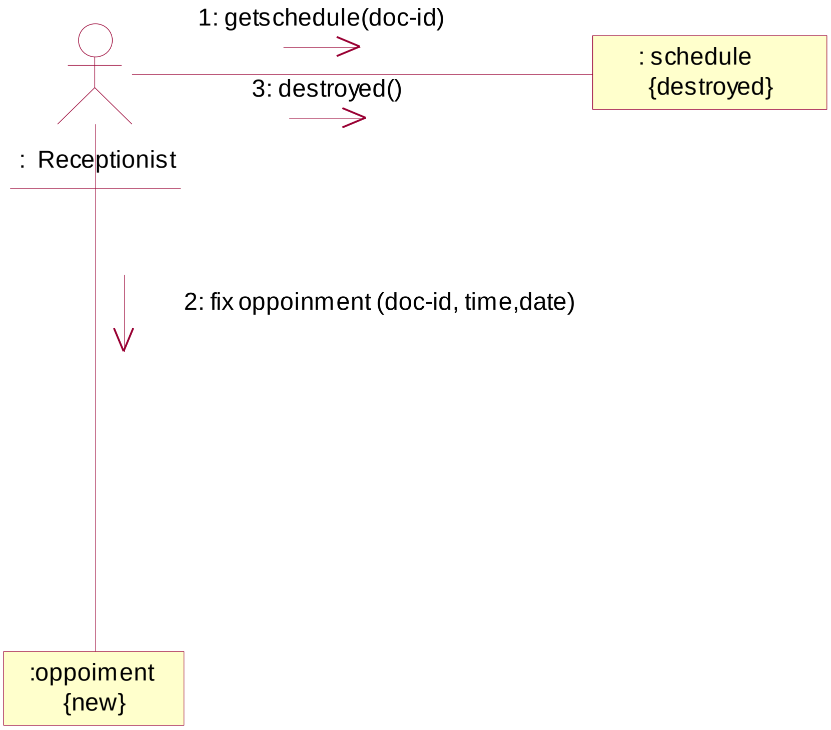
Association roles
Association roles describe how an association will behave given a particular situation. You can draw association roles using simple lines labeled with stereotypes.



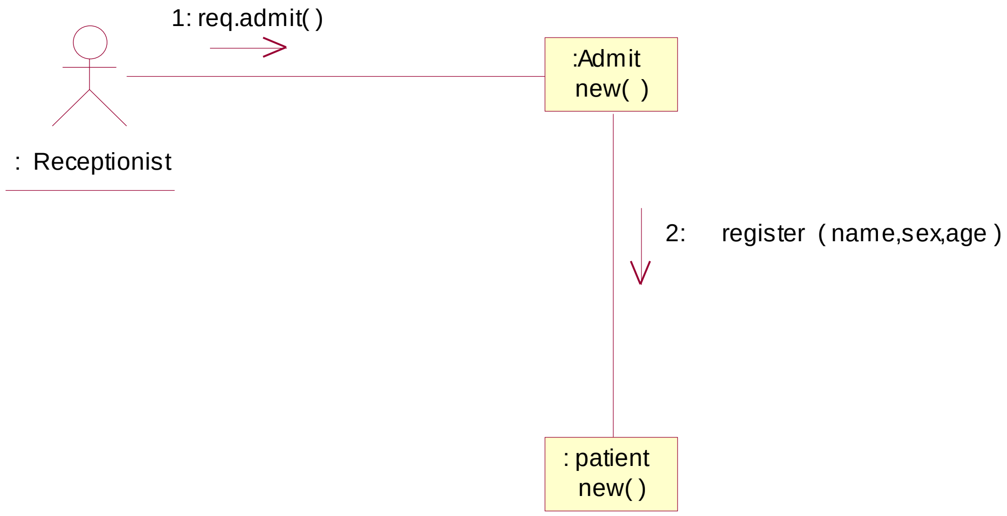
Messages
Unlike sequence diagrams, collaboration diagrams do not have an explicit way to denote time and instead number messages in order of execution. Sequence numbering can become nested using the Dewey decimal system. For example, nested messages under the first message are labeled 1.1, 1.2, 1.3, and so on. The a condition for a message is usually placed in square brackets immediately following the sequence number. Use a * after the sequence number to indicate a loop.



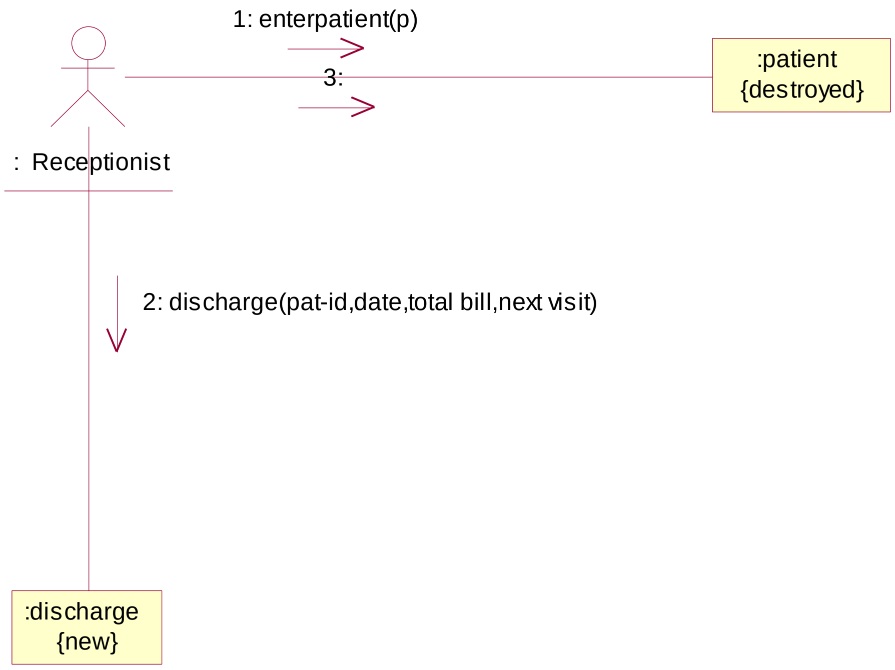
Schedule Appointment



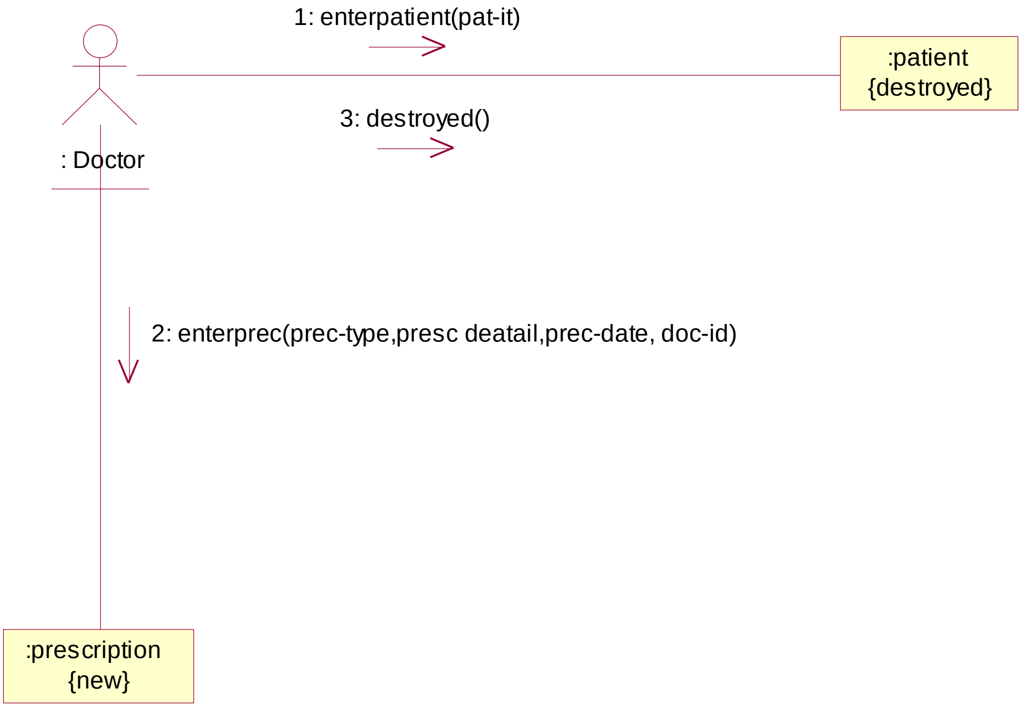
Admit patient



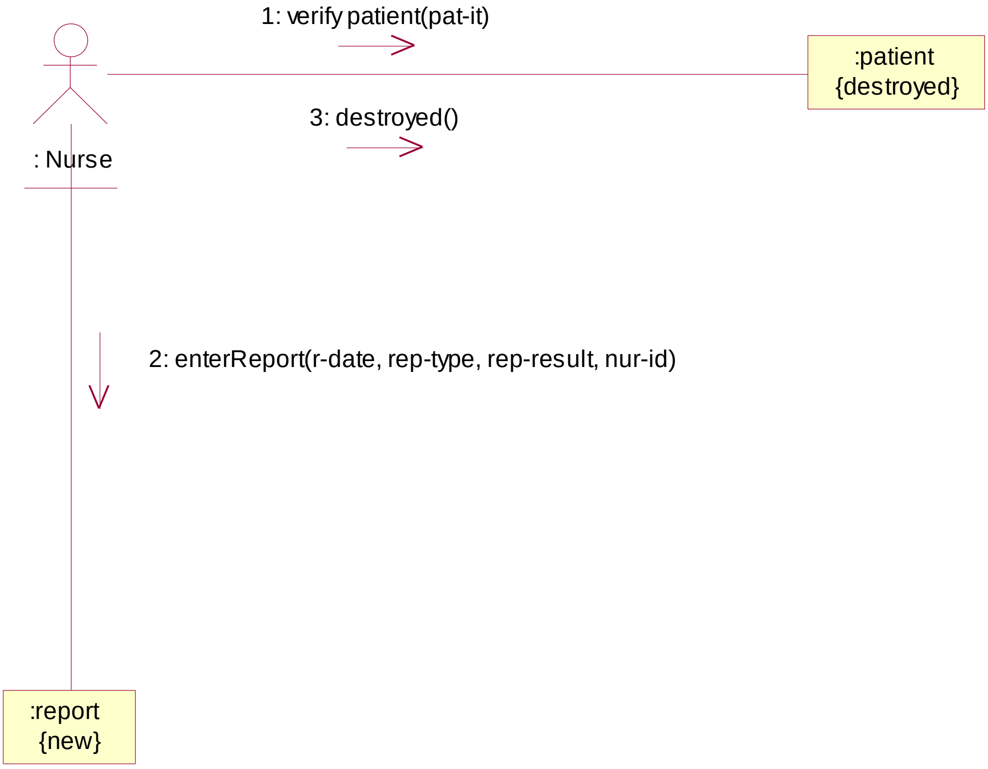
Discharge patient



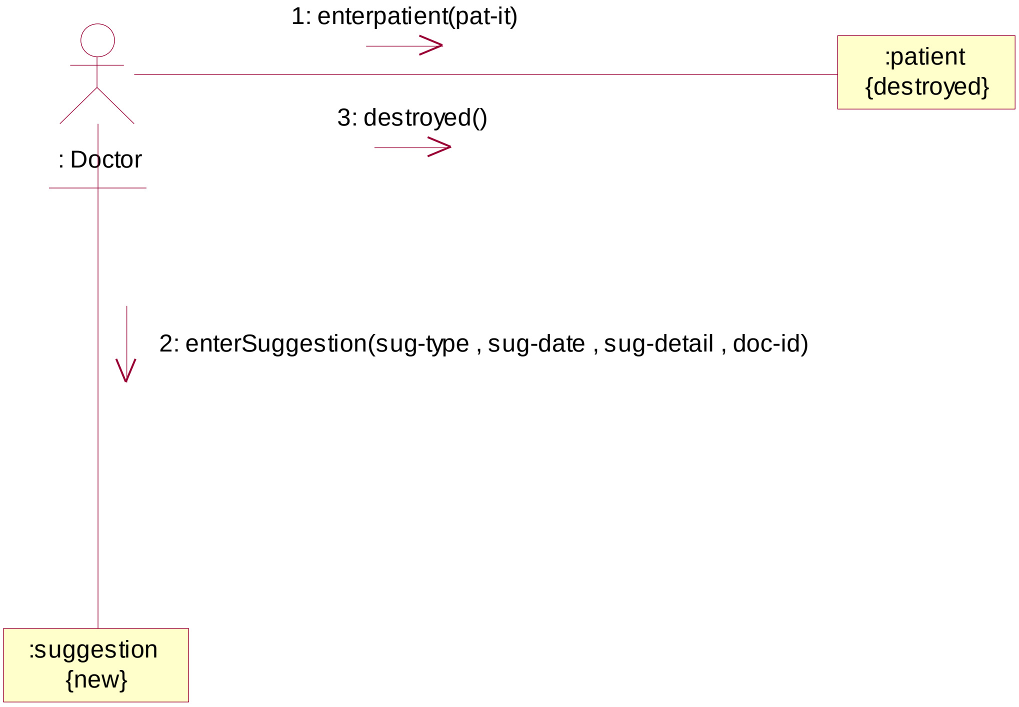
Give prescription



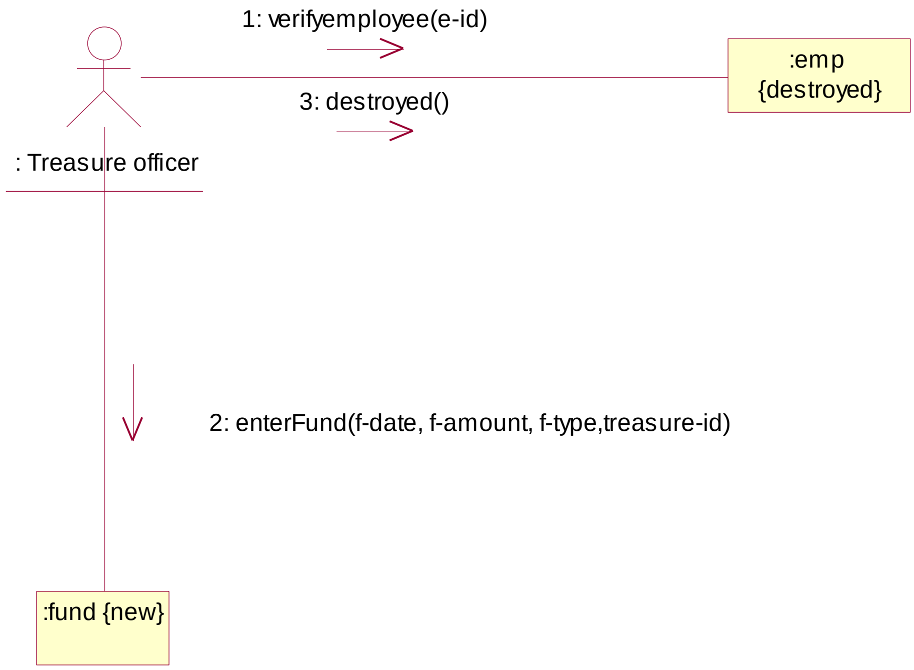
File reports


1: verify patient(pat-it)
:patient
{destroyed}
3: destroyed()
: Nurse
2: enterReport(r-date, rep-type, rep-result, nur-id)
:report
{new}

Give suggestion



Issue funds


1: verifyemployee(e-id)
:emp {destroyed}
3: destroyed()
: Treasure officer
2: enterFund(f-date, f-amount, f-type,treasure-id)
:fund {new}

CLASS DIAGRAM

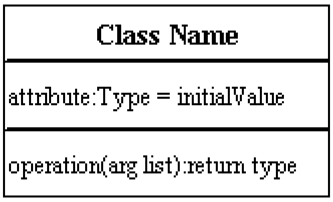
What is a UML Class Diagram?

Class diagrams are the backbone of almost every object-oriented method including UML. They describe the static structure of a system.

Basic Class Diagram Symbols and Notations

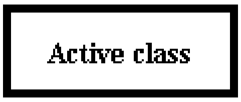
Classes represent an abstraction of entities with common characteristics. Associations represent the relationships between classes.

Illustrate classes with rectangles divided into compartments. Place the name of the class in the first partition (centered, bolded, and capitalized), list the attributes in the second partition, and write operations into the third.

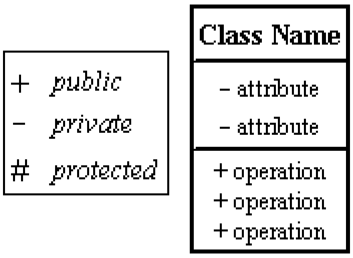

Class Name
attribute:Type = initialValue
operation(arg list):return type

Active Class

Active classes initiate and control the flow of activity, while passive classes store data and serve other classes. Illustrate active classes with a thicker border.

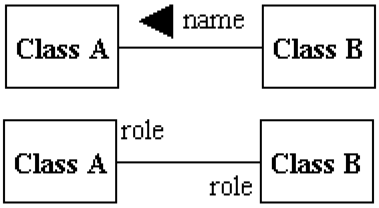


Visibility
Use visibility markers to signify who can access the information contained within a class. Private visibility hides information from anything outside the class partition. Public visibility allows all other classes to view the marked information. Protected visibility allows child classes to access information they inherited from a parent class. Learn how to edit text.



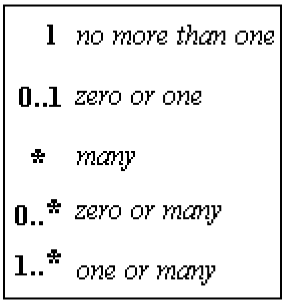
Associations
Associations represent static relationships between classes. Place association names above, on, or below the association line. Use a filled arrow to indicate the direction of the

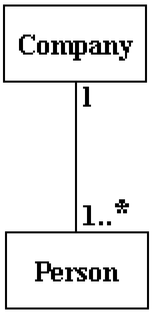
relationship. Place roles near the end of an association. Roles represent the way the two classes see each other.
Note: It's uncommon to name both the association and the class roles.



Multiplicity (Cardinality)

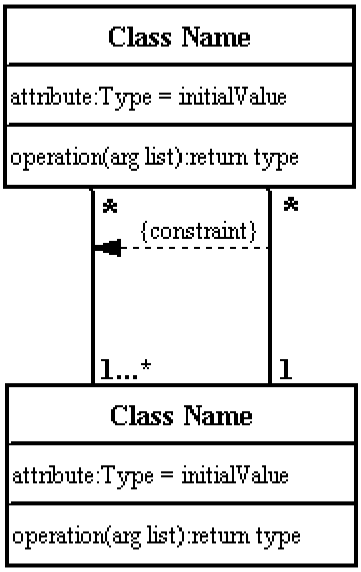
Place multiplicity notations near the ends of an association. These symbols indicate the number of instances of one class linked to one instance of the other class. For example, one company will have one or more employees, but each employee works for one company only.

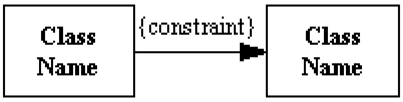




Constraint

Place constraints inside curly braces {}.

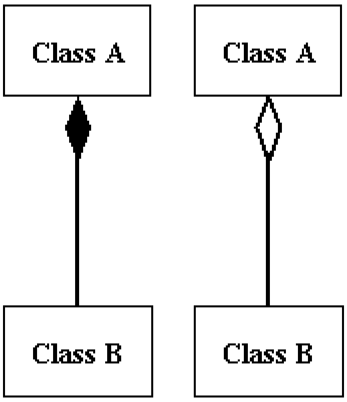

Class Name
attribute:Type = initialValue
operation(arg list):return type
*
{constraint}
*
1...*
1
Class Name
attribute:Type = initialValue
operation(arg list):return type



Simple Constraint

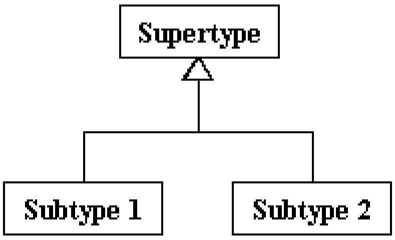
Composition and Aggregation

Composition is a special type of aggregation that denotes a strong ownership between Class A, the whole, and Class B, its part. Illustrate composition with a filled diamond. Use a hollow diamond to represent a simple aggregation relationship, in which the "whole" class plays a more important role than the "part" class, but the two classes are not dependent on each other. The diamond end in both a composition and aggregation relationship points toward the "whole" class or the aggregate.



Generalization

Generalization is another name for inheritance or an "is a" relationship. It refers to a relationship between two classes where one class is a specialized version of another. For example, Honda is a type of car. So the class Honda would have a generalization relationship with the class car.



In real life coding examples, the difference between inheritance and aggregation can be confusing. If you have an aggregation relationship, the aggregate (the whole) can access only the PUBLIC functions of the part class. On the other hand, inheritance allows the inheriting class to access both the PUBLIC and PROTECTED functions of the superclass.