

**BHARATI VIDYAPEETH’S**

**INSTITUTE OF COMPUTER APPLICATIONS & MANAGEMENT**

(Affiliated to Guru Gobind Singh Indraprastha University, Approved by AICTE, New Delhi)

**Object Oriented Software Engineering**

**(MCA-164)**

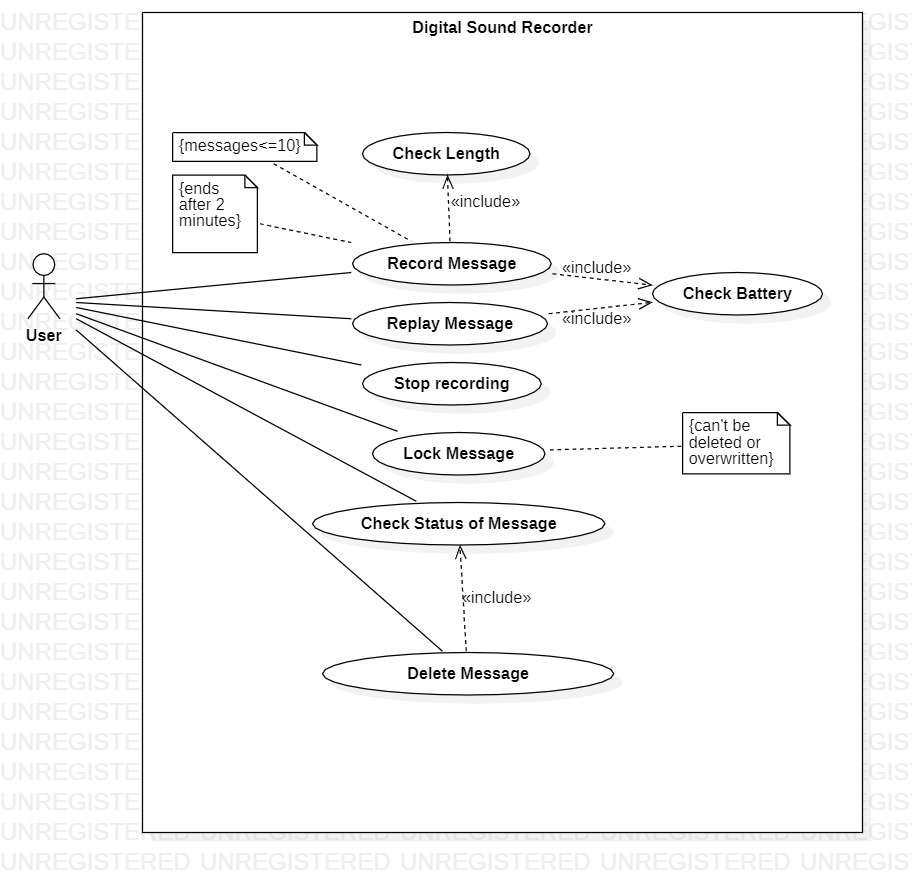
**Practical File**

**INDEX**

|  |  |
| --- | --- |
| **S. No.** | **Problem Description** |
| AP1 | Create a Use Case Diagram for a Digital Sound Recorder with the following main features:   * The recorder stores up to 10 messages * Each message is max. 2 minutes long * The user can record message * Recording of a message ends after 2 minutes or when the user stops recording * Recording destroys the original message at chosen slot * Sufficient level of battery is checked before recording message * Message of a given slot can be replayed * Sufficient level of battery is checked before replaying message * Messages can be locked/unlocked * Locked messages cannot be deleted or over-written by recording to the same slot * User uses LCD display and buttons to interact with recorder |
| AP2 | Formulate use case descriptions for 5 major use cases identified in the digital sound recorder |
| AP3 | As the head of Information Technology at Acme Airlines, you are tasked with building a new booking system to replace the existing system. Acme needs a new system to allow employees to book ticket electronically and check the status of the booked tickets.  The new system will be state of the art and will have a Windows-based desktop interface to allow employees to enter booking details and view employee details. The system will run on individual employee desktops throughout the entire company. For reasons of security and auditing, employees can access only their data. The employees would only be able to view their information, and for any changes they would have to send an  E-Mail to the administrator. Administrator would only have the right to make any changes in the records.  For the cost being a factor Acme wants to use their old server with the existing database. The system will retain information on all employees in the company.  The administrator maintains employee information. He is responsible for adding new employees, deleting employees and changing all employee information such as name, address, and paycheck generation, as well as running administrative reports |
| AP4 | 1. Consider the following use-case of a travel agency.  Use-Case Name: Ticket Purchasing  Description:  1. The use-case begins when the customer calls the travel agency to ask it to issue a ticket that (s)he has booked.  2. The travel agency operator asks the customer to give his/her booking number.  3. The customer gives the booking number.  4. The operator types in the booking number and the flight reservation system displays the details of the reservation made.  5. The operator asks the customer to confirm the details of the reservation made.  5. The customer confirms the reservation made.  6. The operator asks the customer for a credit card number.  7. The customer gives his/her credit card number.  8. The operator types in the customer's credit card number and when the system confirms that the credit card transaction has been authorised (s)he asks the system to print the tickets, the details of the flights, and an invoice.  9. When the system confirms that the requested items have been printed, the operator informs the customer that the tickets have been issued.   Think of at least two alternative courses of events for this use-case. Describe the alternative courses as separate use-cases. Modify the description of the original use-case to make evident where exactly these alternative use-cases may be called and under which conditions. Create a use-case diagram to illustrate the relationships between the alternative use-cases and the use-case given. |
|  | ADVANCED PROBLEMS |
| AQ1 | Create a use case diagram for a restaurant ordering and billing system? |
| AQ2 | Create a use case diagram for any project carried out by you? |
|  | WEEK 2 |
| BP1 | Create a Class Diagram for a Digital Sound Recorder with the following main features:   * The recorder stores up to 10 messages * Each message is max. 2 minutes long * The user can record message * Recording of a message ends after 2 minutes or when the user stops recording * Recording destroys the original message at chosen slot * Sufficient level of battery is checked before recording message * Message of a given slot can be replayed * Sufficient level of battery is checked before replaying message * Messages can be locked/unlocked * Locked messages cannot be deleted or over-written by recording to the same slot * User uses LCD display and buttons to interact with recorder |
| BP2 | Forward engineer the model obtained in BP1 to automatically generate equivalent Java stub Code? |
|  | ADVANCED PROBLEMS |
| BQ1 | Create a class diagram for your project for which you created use case model in AQ2? |
| BQ2 | A reactive real-time object (eg: any embedded real time system) wants to communicate an event to an interactive non-real time object (eg: user-interface or system monitoring or data logging)? |
|  | WEEK 3 |
| CP1 | Purpose: Describe major services (functionality) provided by a hospital's reception.  Summary: Hospital Management System is a large system including several subsystems or modules providing variety of functions. Hospital Reception subsystem or module supports some of the many job duties of hospital receptionist. Receptionist schedules patient's appointments and admission to the hospital, collects information from patient upon patient's arrival and/or by phone. For the patient that will stay in the hospital ("inpatient") she or he should have a bed allotted in a ward. Receptionist may also receive and process patient’s payments, record them and provide receipts. She also is responsible to file insurance claims and generate medical reports.   1. Make the use case diagram 2. Provide description for atleast 04 use cases 3. Make the class diagram 4. Generate Java stub code 5. Create Object Diagram |
| CP2 | Create an object diagram for a Digital Sound Recorder with the following main features:   * The recorder stores up to 10 messages * Each message is max. 2 minutes long * The user can record message * Recording of a message ends after 2 minutes or when the user stops recording * Recording destroys the original message at chosen slot * Sufficient level of battery is checked before recording message * Message of a given slot can be replayed * Sufficient level of battery is checked before replaying message * Messages can be locked/unlocked * Locked messages cannot be deleted or over-written by recording to the same slot   User uses LCD display and buttons to interact with recorder |
| CP3 | Create a package diagram for the Digital Sound Recorder in AP1 |
|  | ADVANCED PROBLEMS |
| CQ1 | Create an object diagram for your project? |
| CQ2 | Create a corresponding package diagram for your project? |
|  | WEEK 4 |
| DP1 | Create an activity diagram for record message use case of AP1? |
| DP2 | Create an activity diagram for each basic use case of the project made by you in AQ2? |
|  | WEEK 5 |
| EP1 | Create a sequence diagram for the playback message use case of AP1? |
| EP2 | Create a sequence diagram for each basic use case of the project made by you in AQ2? |
| EP3 | Create a state diagram for a BANK ATM machine? |
| EP4 | Create a state diagram for each basic use case of the project made by you in AQ2? |
| FP1 | Create a communication diagram for the playback message use case of AP1? |
| FP2 | Create a communication diagram for each basic use case of the project made by you in AQ2? |
| GP1 | Create a component diagram for the Hospital Reception Subsystem? |
| GP2 | Create a component diagram for your project? |
| HP1 | Create a deployment diagram for the Hospital Reception Subsystem? |
| HP2 | Create a deployment diagram for your project? |
|  | PERT /GANTT CHART |
| IP1 | Create a PERT Chart on Powerpoint for the following   |  |  |  |  | | --- | --- | --- | --- | | Activity | Description | Predecessor | Estimated Time (hours) | | A | Drive Home | None | 0.5 | | B | Wash Clothes | A | 4.0 | | C | Pack | B | 0.5 | | D | Go to Bank | A | 1.0 | | E | Pay Bill | D | 0.5 | | F | Pack Car | C,E | 0.5 | | G | Drive to Bus | F | 0.5 | |
| IP2 | |  |  |  |  | | --- | --- | --- | --- | | Task | Prec Task | Description | Time (in hrs) | | A | None | Decide on Date for the Party | 1 | | B | A | Book a bouncy Castle | 1 | | C | A | Send Invitations | 5 | | D | C | Receive Replies | 7 | | E | D | Buy Toys and Balloons | 1 | | F | D | Buy Food | 3 | | G | E | Blow up Balloons | 2 | | H | F | Make Food | 1 | | I | H, G | Decorate | 1 | | J | B | Get Bouncy Castle | 1 | | K | J, I | Have Party | 2 | | L | K | CleanUp | 4 | | M | K | Send Back Bouncy Castle | 1 | | N | M | Donate Unwanted Gifts | 3 |  * + label hours 0 to 30 across the horizontal axis   + Mark a review stage at hour 14 to monitor the progress   + Assume and illustrate that tasks A, B, C and D have been completed at hour 14   + State which tasks are ahead and which tasks are behind schedule   Make a Gantt chart for the above? |

**AP1** Create a Use Case Diagram for a Digital Sound Recorder with the following main features:

* The recorder stores up to 10 messages
* Each message is max. 2 minutes long
* The user can record message
* Recording of a message ends after 2 minutes or when the user stops recording
* Recording destroys the original message at chosen slot
* Sufficient level of battery is checked before recording message
* Message of a given slot can be replayed
* Sufficient level of battery is checked before replaying message
* Messages can be locked/unlocked
* Locked messages cannot be deleted or over-written by recording to the same slot
* User uses LCD display and buttons to interact with recorder



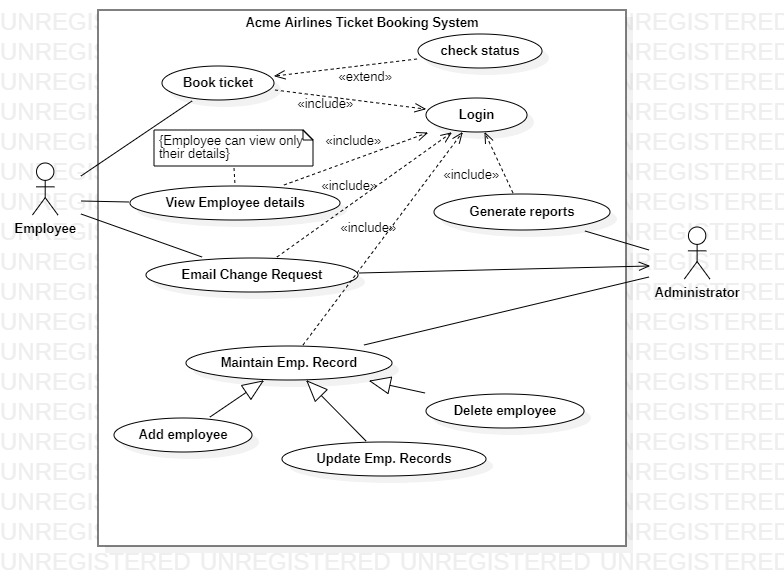
**AP3** As the head of Information Technology at Acme Airlines, you are tasked with building a new booking system to replace the existing system. Acme needs a new system to allow employees to book ticket electronically and check the status of the booked tickets.

The new system will be state of the art and will have a Windows-based desktop interface to allow employees to enter booking details and view employee details. The system will run on individual employee desktops throughout the entire company. For reasons of security and auditing, employees can access only their data. The employees would only be able to view their information, and for any changes they would have to send an

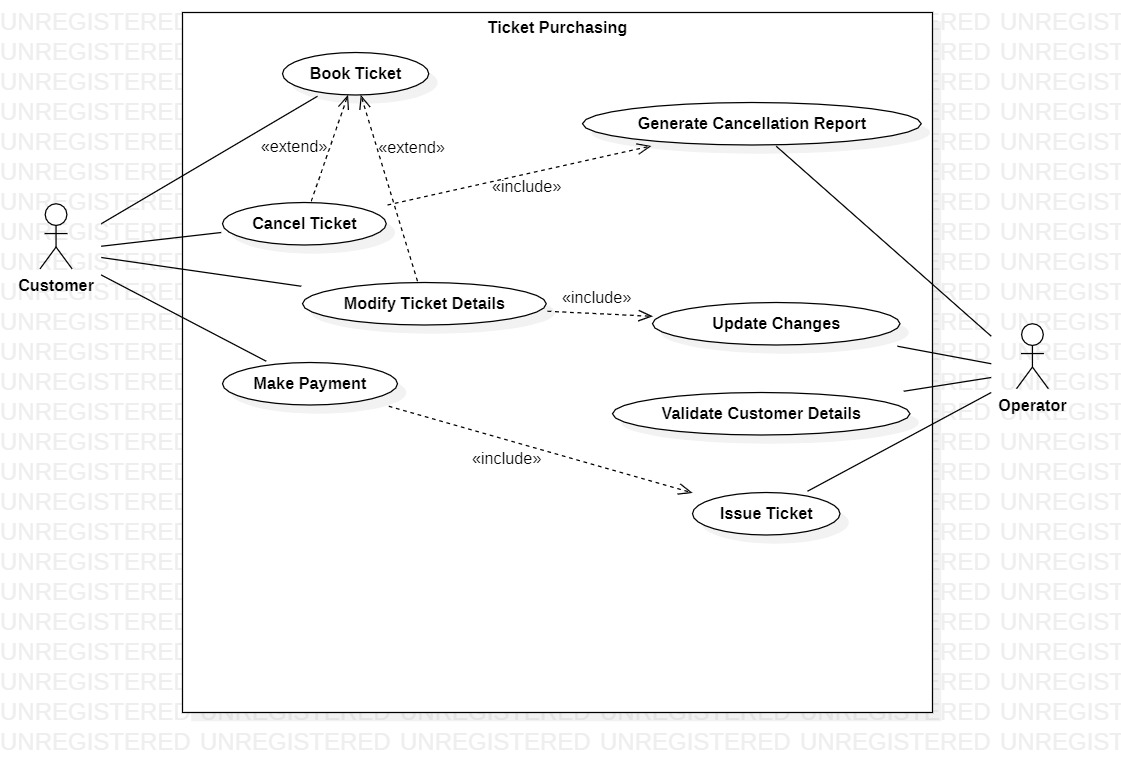
E-Mail to the administrator. Administrator would only have the right to make any changes in the records.

For the cost being a factor Acme wants to use their old server with the existing database. The system will retain information on all employees in the company.

The administrator maintains employee information. He is responsible for adding new employees, deleting employees and changing all employee information such as name, address, and paycheck generation, as well as running administrative reports



**AP4:** Consider the following use-case of a travel agency.   
Use-Case Name: Ticket Purchasing  
Description:   
1. The use-case begins when the customer calls the travel agency to ask it to issue a ticket that (s)he has booked.   
2. The travel agency operator asks the customer to give his/her booking number.   
3. The customer gives the booking number.   
4. The operator types in the booking number and the flight reservation system displays the details of the reservation made.   
5. The operator asks the customer to confirm the details of the reservation made.   
6. The customer confirms the reservation made.   
7. The operator asks the customer for a credit card number.   
8. The customer gives his/her credit card number.   
9. The operator types in the customer's credit card number and when the system confirms that the credit card transaction has been authorised (s)he asks the system to print the tickets, the details of the flights, and an invoice.   
10. When the system confirms that the requested items have been printed, the operator informs the customer that the tickets have been issued.   
  
Think of at least two alternative courses of events for this use-case. Describe the alternative courses as separate use-cases. Modify the description of the original use-case to make evident where exactly these alternative use-cases may be called and under which conditions. Create a use-case diagram to illustrate the relationships between the alternative use-cases and the use-case given.



**AQ2** Create a use case diagram for any project carried out by you?

FALCON (a chess engine)

INTRODUCTION: A chess engine is simply a software program that plays and analyzes chess. The word “engine” simply refers to a kind of high-powered program that does a lot of searching and processing—similar to a search engine.

Problem Definition:

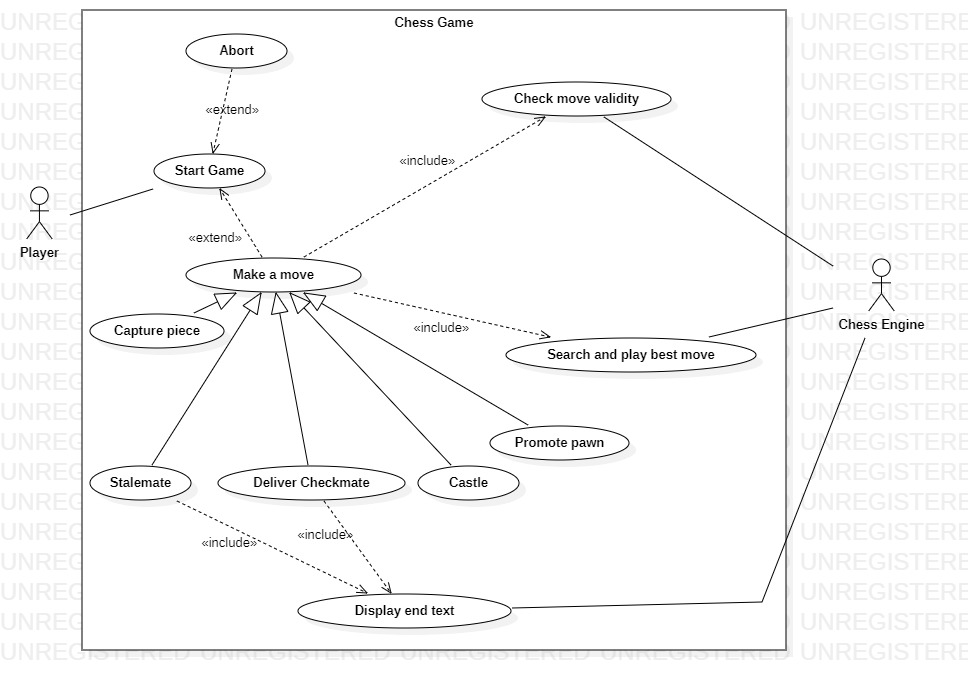
Falcon is aimed to be a Python enabled chess engine that automates and streamlines the conventional task of analysing the chess move played in a given position during the chess game.

MODULE DESCRIPTION

The various modules of the system are detailed below:

(Describe each briefly)

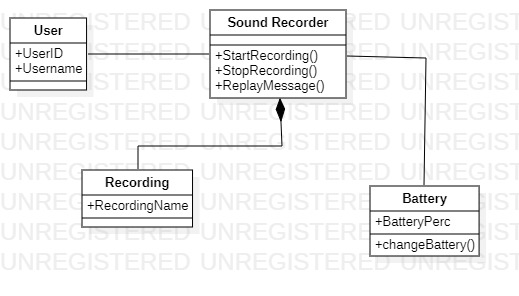
* ChessMain.py: contains main function, responsible for rendering board, game state, pieces, highlighting squares, end text
* ChessEngine.py: Contains 2-D list for drawing board, Move class, GameState Class, functions for making a move, getting all possible moves, all valid moves(each piece moves: pawn, rook, bishop, knight, king and queen), undo moves, castle moves(King side castle and Queen side castle), function for getting chess notation, to check if king is in check, if square is under attack, if en passant move is possible
* ChessAI.py: functions to generate and make best moves by the help of Nega Max, Alpha-Beta Pruning Algorithm (Upto Depth=2).



BP1 Create a Class Diagram for a Digital Sound Recorder with the following main features:

* The recorder stores up to 10 messages
* Each message is max. 2 minutes long
* The user can record message
* Recording of a message ends after 2 minutes or when the user stops recording
* Recording destroys the original message at chosen slot
* Sufficient level of battery is checked before recording message
* Message of a given slot can be replayed
* Sufficient level of battery is checked before replaying message
* Messages can be locked/unlocked
* Locked messages cannot be deleted or over-written by recording to the same slot

User uses LCD display and buttons to interact with recorder



BP2 Forward engineer the model obtained in BP1 to automatically generate equivalent Java stub Code?

import java.util.\*;

/\*\*

\*

\*/

public class Sound Recorder {

/\*\*

\* Default constructor

\*/

public Sound Recorder() {

}

/\*\*

\*

\*/

public void StartRecording() {

// TODO implement here

}

/\*\*

\*

\*/

public void StopRecording() {

// TODO implement here

}

/\*\*

\*

\*/

public void ReplayMessage() {

// TODO implement here

}

}

import java.util.\*;

/\*\*

\*

\*/

public class Recording {

/\*\*

\* Default constructor

\*/

public Recording() {

}

/\*\*

\*

\*/

public void RecordingName;

}

import java.util.\*;

/\*\*

\*

\*/

public class User {

/\*\*

\* Default constructor

\*/

public User() {

}

/\*\*

\*

\*/

public void UserID;

/\*\*

\*

\*/

public void Username;

}

import java.util.\*;

/\*\*

\*

\*/

public class Battery {

/\*\*

\* Default constructor

\*/

public Battery() {

}

/\*\*

\*

\*/

public void BatteryPerc;

/\*\*

\*

\*/

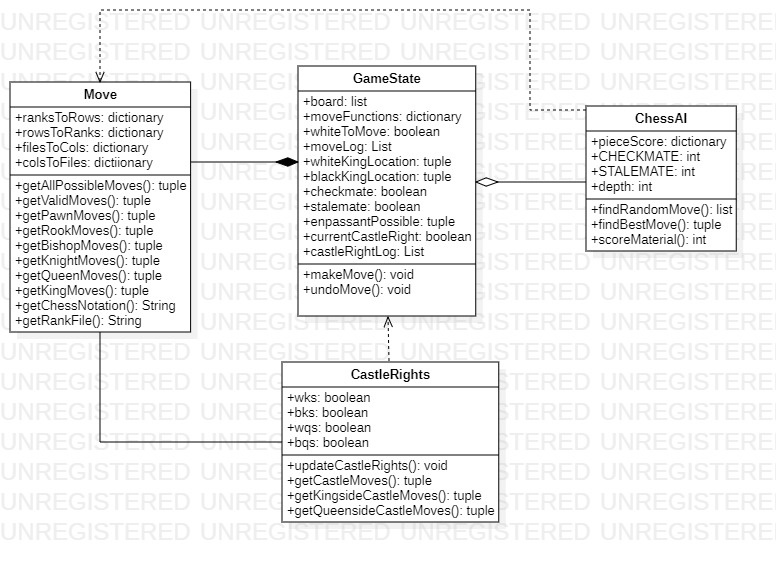
public void changeBattery() {

// TODO implement here

}

}

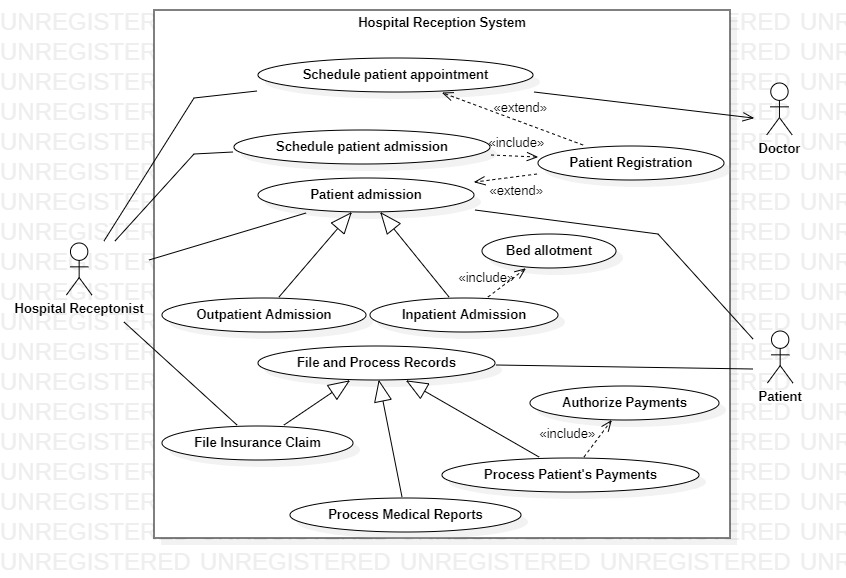
BQ1 Create a class diagram for your project for which you created use case model in AQ2?



**CP1** **Purpose:** Describe major services (functionality) provided by a hospital's reception.

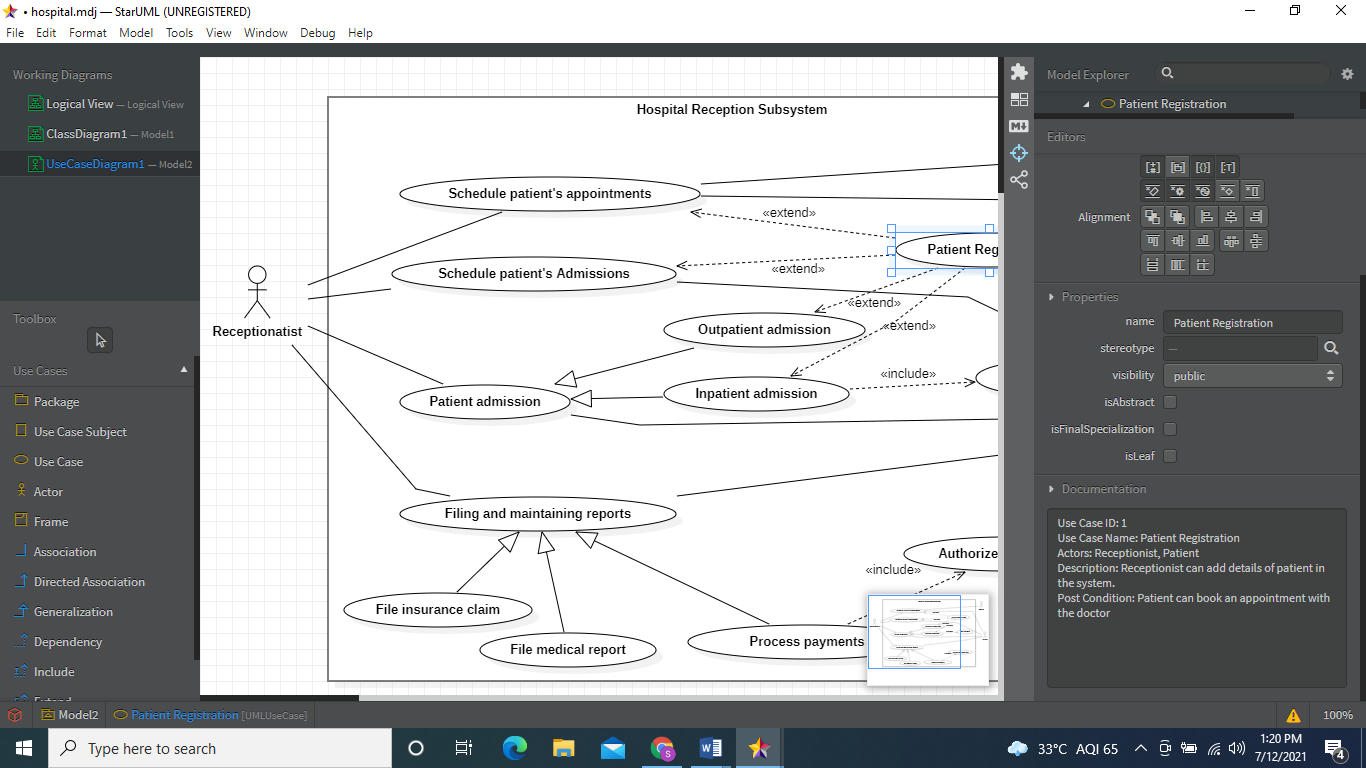
**Summary**: **Hospital Management System** is a large system including several subsystems or modules providing variety of functions. **Hospital Reception** subsystem or module supports some of the many job duties of hospital receptionist. Receptionist schedules patient's appointments and admission to the hospital, collects information from patient upon patient's arrival and/or by phone. For the patient that will stay in the hospital ("inpatient") she or he should have a bed allotted in a ward. Receptionist may also receive and process patient’s payments, record them and provide receipts. She also is responsible to file insurance claims and generate medical reports.

1. Make the use case diagram
2. Provide description for at least 04 use cases
3. Make the class diagram
4. Generate Java stub code

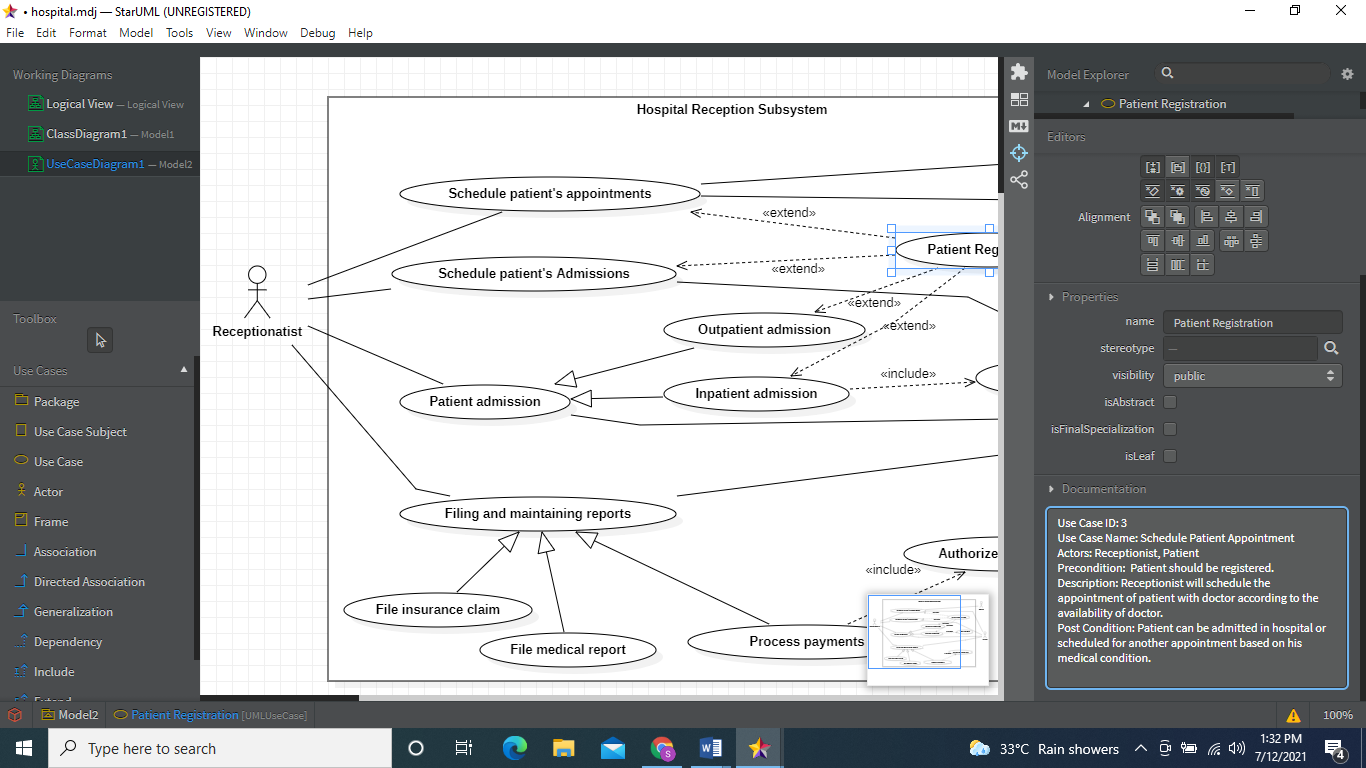


**Documentation:**

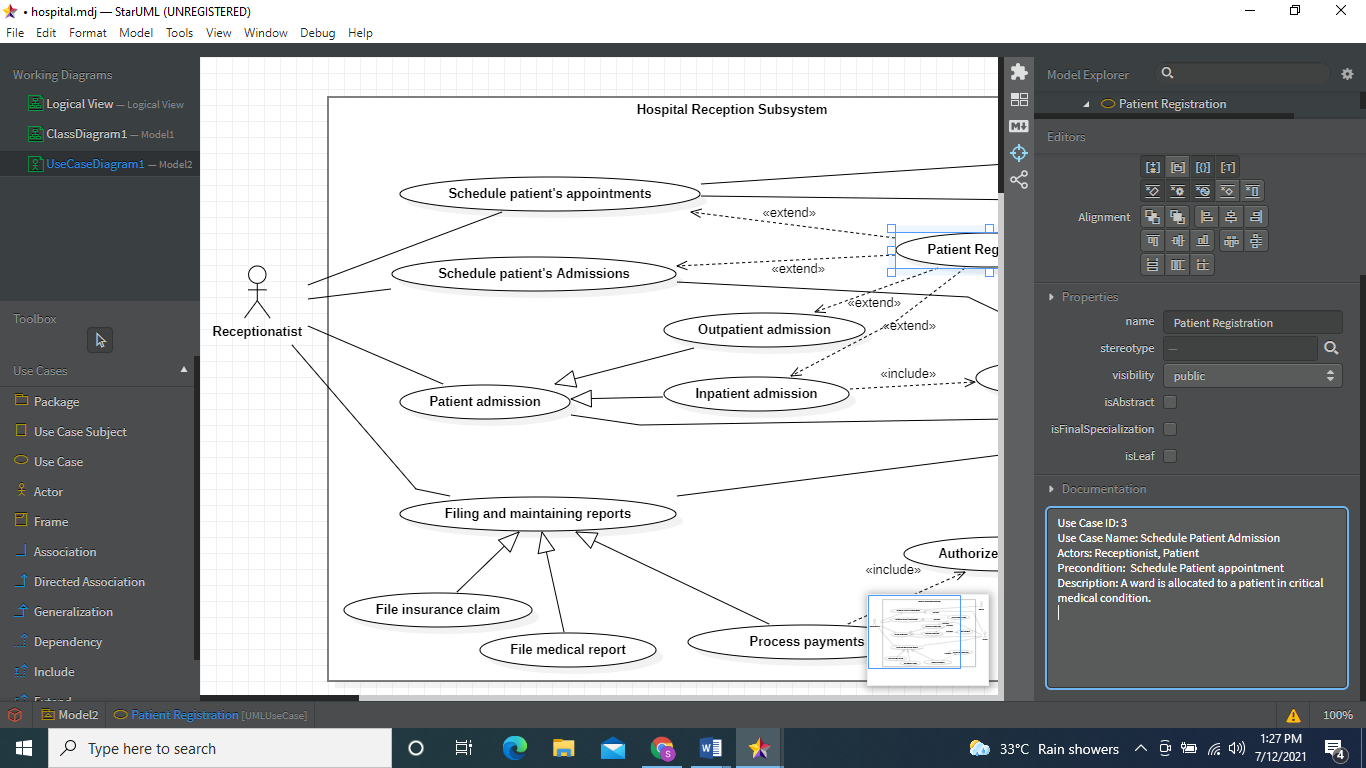
**Patient Registration**



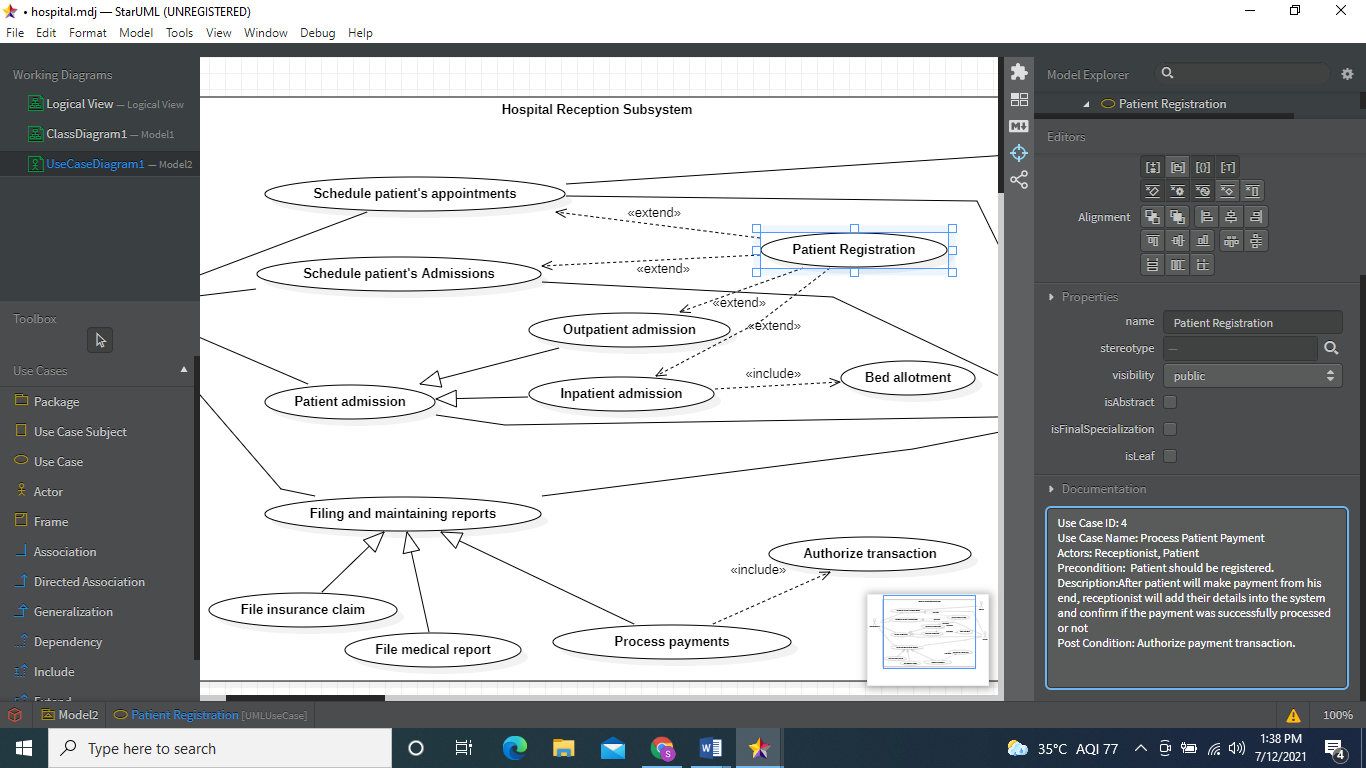
**Schedule Patient Appointment**



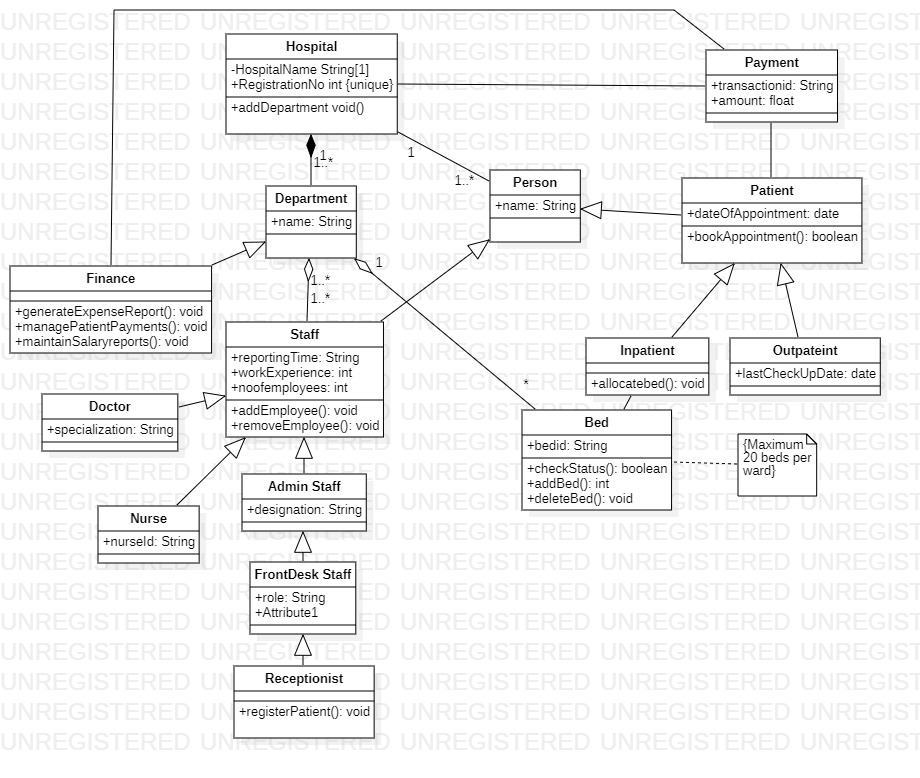
**Schedule Patient Admission**



**Process Patient Payment**



Hospital Class Diagram:



**JAVA STUB CODE:**

**//Hospital.java**

import java.util.\*;

/\*\*

\*

\*/

public class Hospital {

/\*\*

\* Default constructor

\*/

public Hospital() {

}

/\*\*

\*

\*/

private void HospitalName String[1];

/\*\*

\*

\*/

public void RegistrationNo int;

/\*\*

\*

\*/

public void Attribute2;

/\*\*

\*

\*/

public void addDepartment void() {

// TODO implement here

}

}

**//Department.java**

import java.util.\*;

/\*\*

\*

\*/

public class Department {

/\*\*

\* Default constructor

\*/

public Department() {

}

/\*\*

\*

\*/

public String name;

}

**//Staff.java**

import java.util.\*;

/\*\*

\*

\*/

public class Staff extends Person {

/\*\*

\* Default constructor

\*/

public Staff() {

}

/\*\*

\*

\*/

public String reportingTime;

/\*\*

\*

\*/

public int workExperience;

/\*\*

\*

\*/

public int noofemployees;

/\*\*

\* @return

\*/

public void addEmployee() {

// TODO implement here

return null;

}

/\*\*

\* @return

\*/

public void removeEmployee() {

// TODO implement here

return null;

}

}

**//Person.java**

import java.util.\*;

/\*\*

\*

\*/

public class Person {

/\*\*

\* Default constructor

\*/

public Person() {

}

/\*\*

\*

\*/

public String name;

}

**//Patient.java**

import java.util.\*;

/\*\*

\*

\*/

public class Patient extends Person {

/\*\*

\* Default constructor

\*/

public Patient() {

}

/\*\*

\*

\*/

public date dateOfAppointment;

/\*\*

\* @return

\*/

public boolean bookAppointment() {

// TODO implement here

return false;

}

}

**//Inpatient.java**

import java.util.\*;

/\*\*

\*

\*/

public class Inpatient extends Patient {

/\*\*

\* Default constructor

\*/

public Inpatient() {

}

/\*\*

\* @return

\*/

public void allocatebed() {

// TODO implement here

return null;

}

}

**//Outpatient.java**

import java.util.\*;

/\*\*

\*

\*/

public class Outpateint extends Patient {

/\*\*

\* Default constructor

\*/

public Outpateint() {

}

/\*\*

\*

\*/

public date lastCheckUpDate;

}

**//Doctor.java**

import java.util.\*;

/\*\*

\*

\*/

public class Doctor extends Staff {

/\*\*

\* Default constructor

\*/

public Doctor() {

}

/\*\*

\*

\*/

public String specialization;

}

**//Nurse.java**

import java.util.\*;

/\*\*

\*

\*/

public class Nurse extends Staff {

/\*\*

\* Default constructor

\*/

public Nurse() {

}

/\*\*

\*

\*/

public String nurseId;

}

**//Admin Staff.java**

import java.util.\*;

/\*\*

\*

\*/

public class Admin Staff extends Staff {

/\*\*

\* Default constructor

\*/

public Admin Staff() {

}

/\*\*

\*

\*/

public String designation;

}

**//FrontDesk Staff.java**

import java.util.\*;

/\*\*

\*

\*/

public class FrontDesk Staff extends Admin Staff {

/\*\*

\* Default constructor

\*/

public FrontDesk Staff() {

}

/\*\*

\*

\*/

public String role;

/\*\*

\*

\*/

public void Attribute1;

}

**//Receptionist.java**

import java.util.\*;

/\*\*

\*

\*/

public class Receptionist extends FrontDesk Staff {

/\*\*

\* Default constructor

\*/

public Receptionist() {

}

/\*\*

\* @return

\*/

public void registerPatient() {

// TODO implement here

return null;

}

}

**//Bed.java**

import java.util.\*;

/\*\*

\*

\*/

public class Bed {

/\*\*

\* Default constructor

\*/

public Bed() {

}

/\*\*

\*

\*/

public String bedid;

/\*\*

\* @return

\*/

public boolean checkStatus() {

// TODO implement here

return false;

}

/\*\*

\* @return

\*/

public int addBed() {

// TODO implement here

return 0;

}

/\*\*

\* @return

\*/

public void deleteBed() {

// TODO implement here

return null;

}

}

**//Finance.java**

import java.util.\*;

/\*\*

\*

\*/

public class Finance extends Staff {

/\*\*

\* Default constructor

\*/

public Finance() {

}

/\*\*

\* @return

\*/

public void generateExpenseReport() {

// TODO implement here

return null;

}

/\*\*

\* @return

\*/

public void managePatientPayments() {

// TODO implement here

return null;

}

/\*\*

\* @return

\*/

public void maintainSalaryreports() {

// TODO implement here

return null;

}

}

**//Payment.java**

import java.util.\*;

/\*\*

\*

\*/

public class Payment {

/\*\*

\* Default constructor

\*/

public Payment() {

}

/\*\*

\*

\*/

public String transactionid;

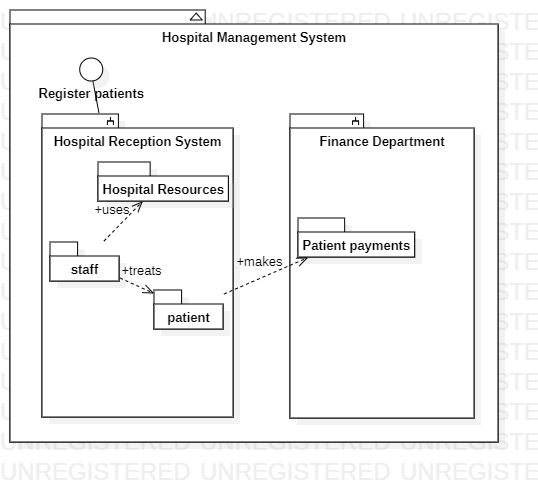
/\*\*

\*

\*/

public float amount;

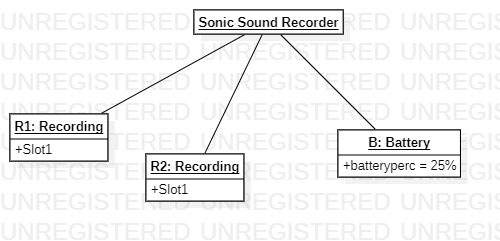
}



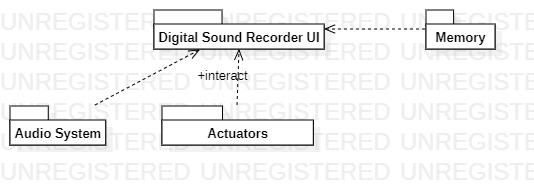
CP2 Create an object diagram for a Digital Sound Recorder with the following main features:

* The recorder stores up to 10 messages
* Each message is max. 2 minutes long
* The user can record message
* Recording of a message ends after 2 minutes or when the user stops recording
* Recording destroys the original message at chosen slot
* Sufficient level of battery is checked before recording message
* Message of a given slot can be replayed
* Sufficient level of battery is checked before replaying message
* Messages can be locked/unlocked
* Locked messages cannot be deleted or over-written by recording to the same slot

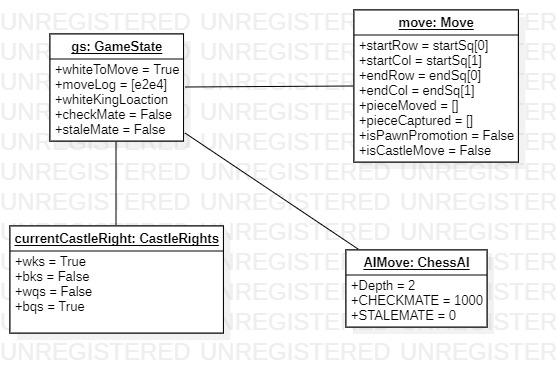
User uses LCD display and buttons to interact with recorder



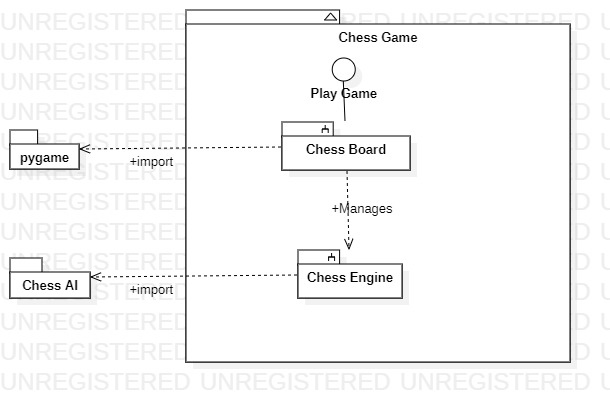
CP3 Create a package diagram for the Digital Sound Recorder in AP1

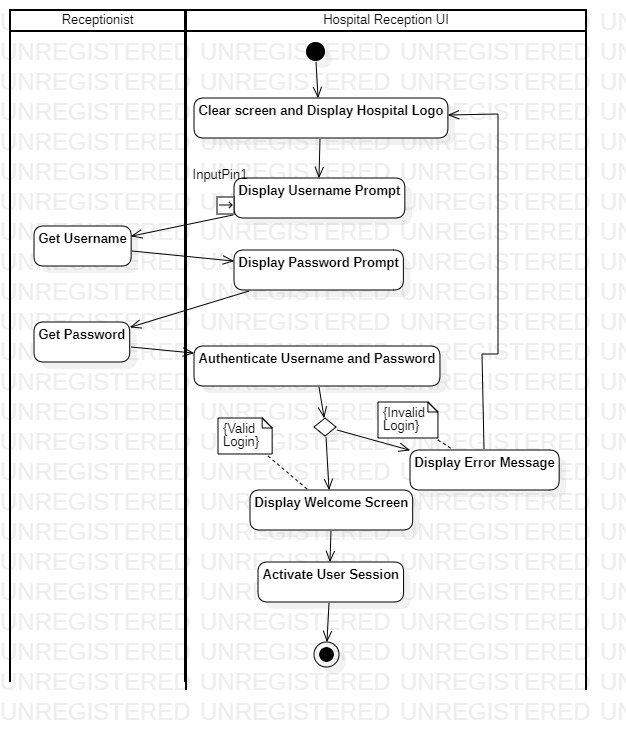


CQ1 Create an object diagram for your project?

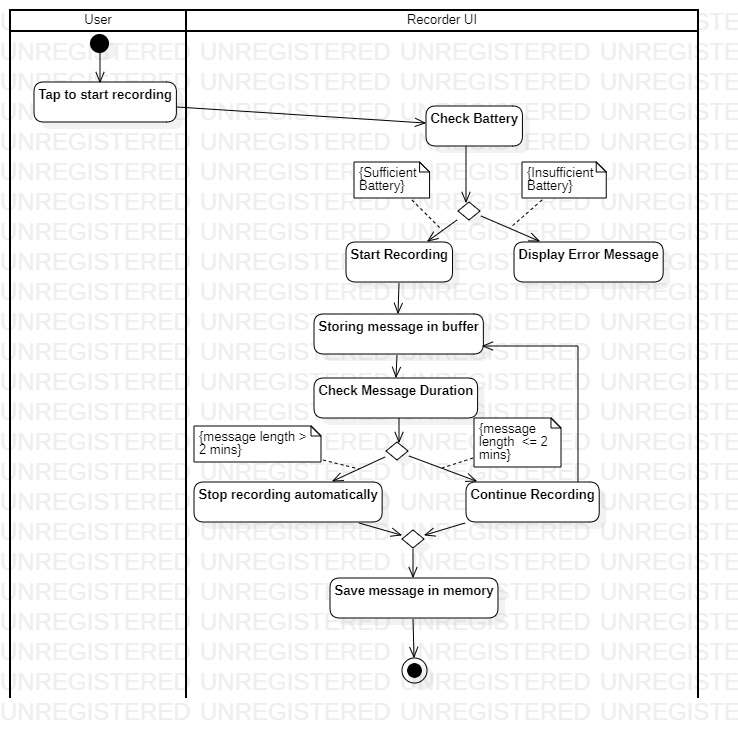


CQ2 Create a corresponding package diagram for your project?



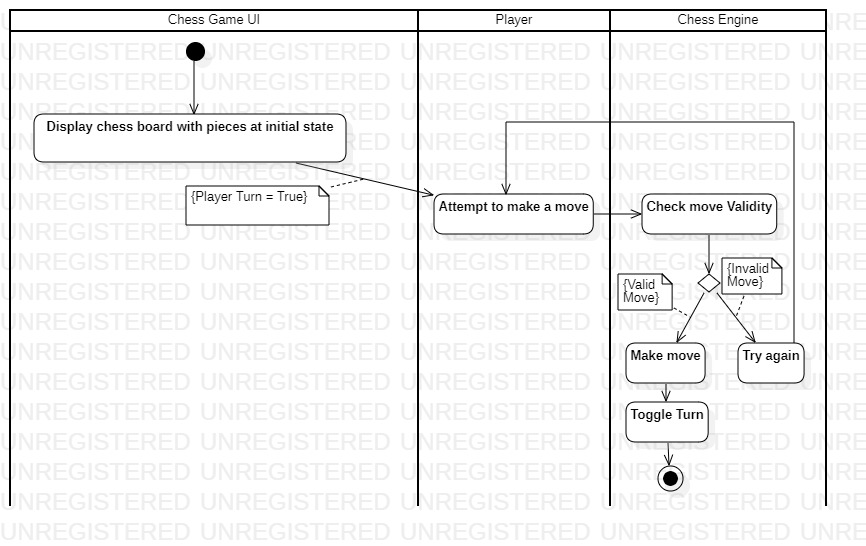


DP1 Create an activity diagram for record message use case of AP1?

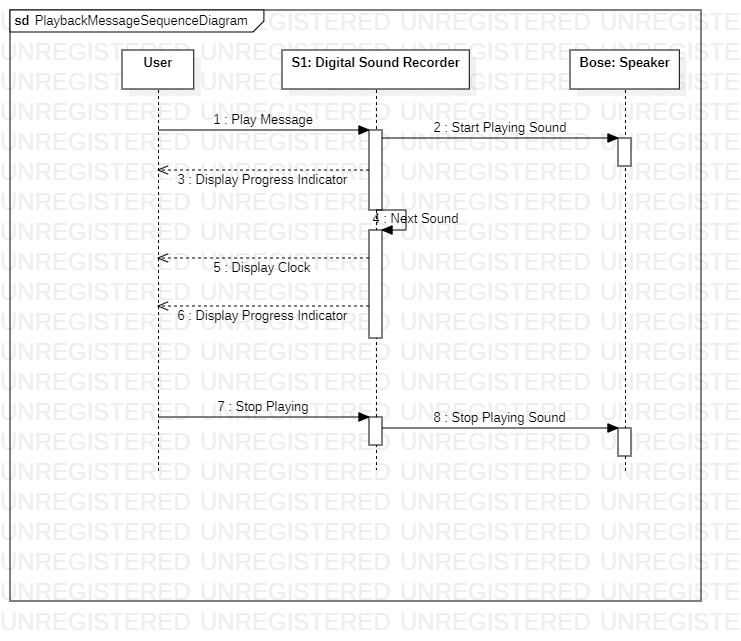


DP2 Create an activity diagram for each basic use case of the project made by you in AQ2?

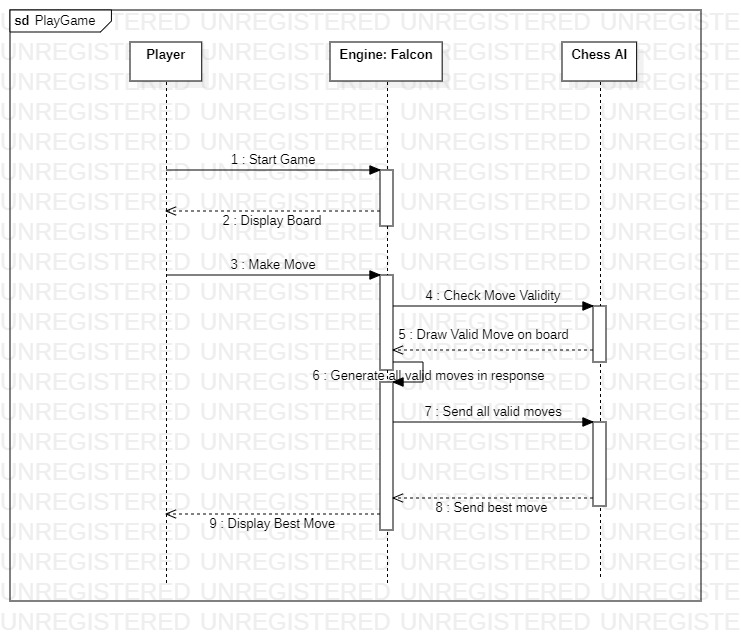
Actitvity Diagram for Make Move Use Case:-



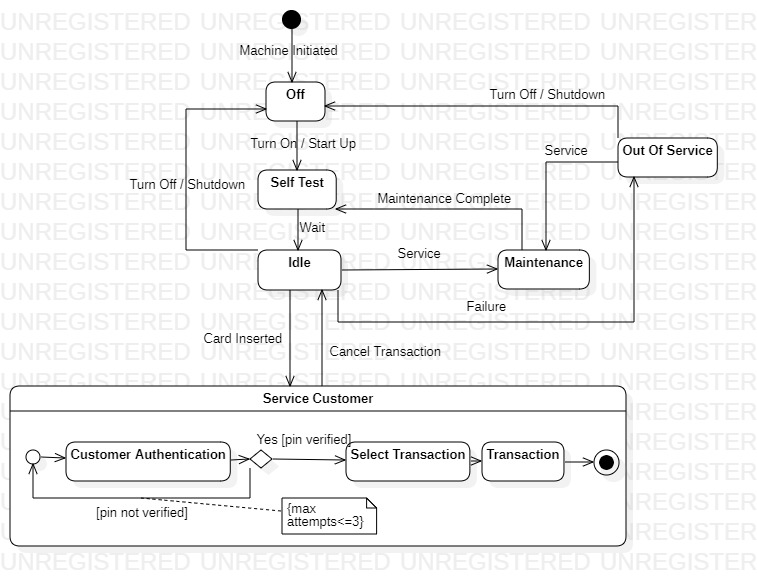
EP1 Create a sequence diagram for the playback message use case of AP1?



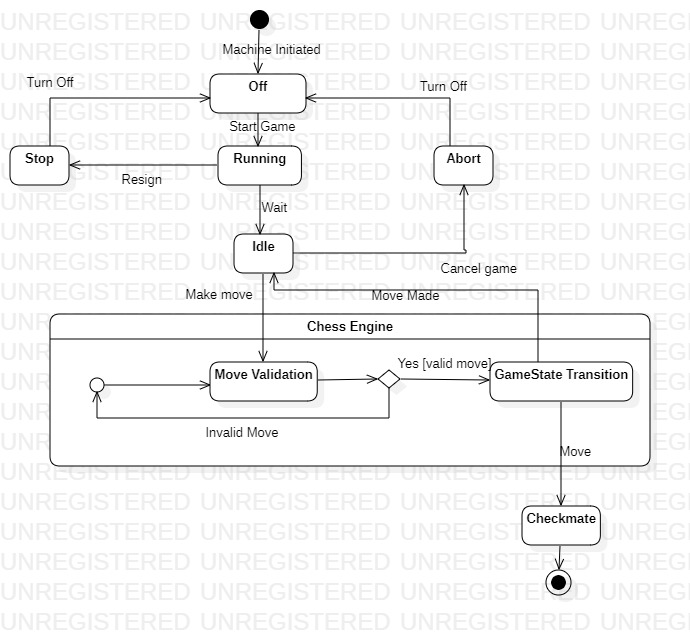
EP2 Create a sequence diagram for each basic use case of the project made by you in AQ2?



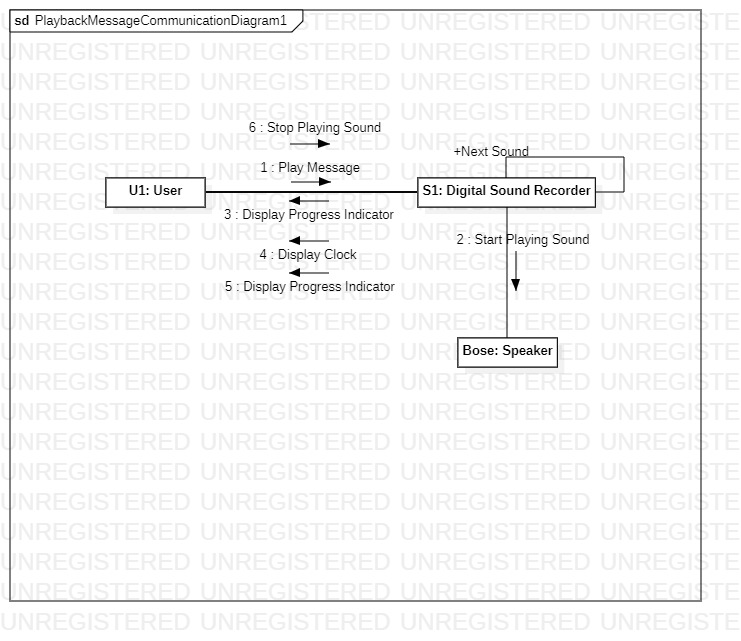
EP3 Create a state diagram for a BANK ATM machine?



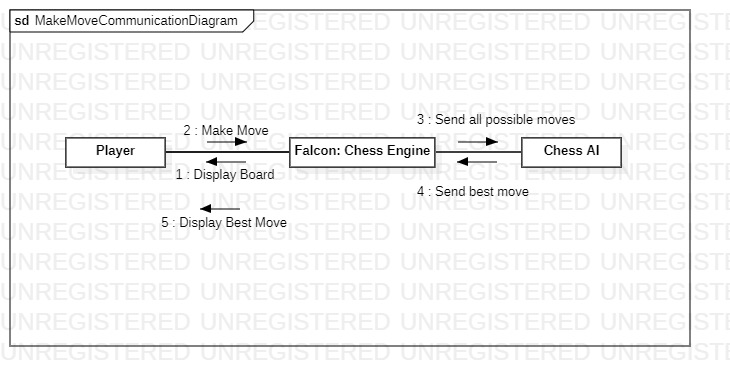
EP4 Create a state diagram for each basic use case of the project made by you in AQ2?



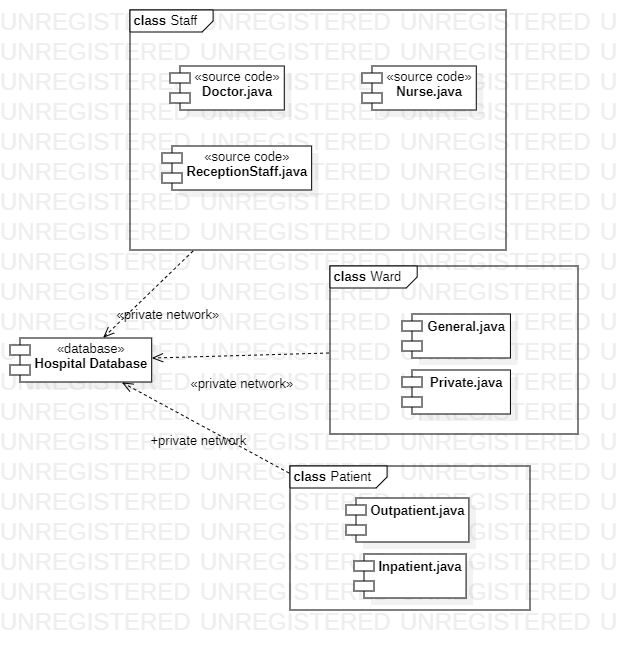
FP1 Create a communication diagram for the playback message use case of AP1?



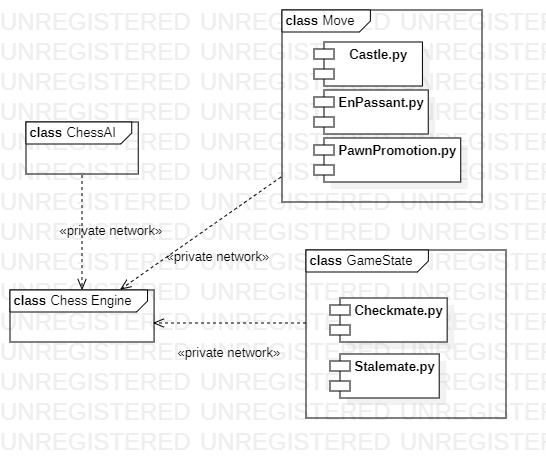
FP2 Create a communication diagram for each basic use case of the project made by you in AQ2?



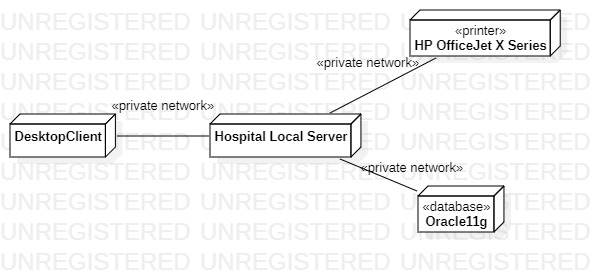
GP1 Create a component diagram for the Hospital Reception Subsystem?



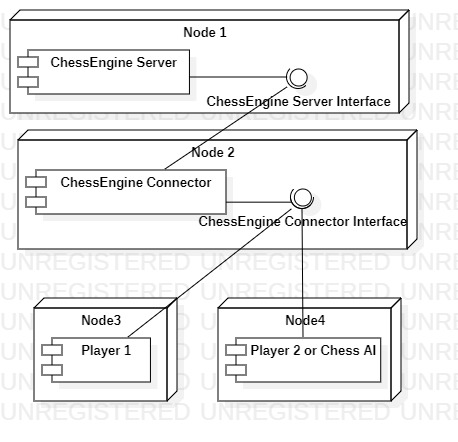
GP2 Create a component diagram for your project?



HP1 Create a deployment diagram for the Hospital Reception Subsystem?



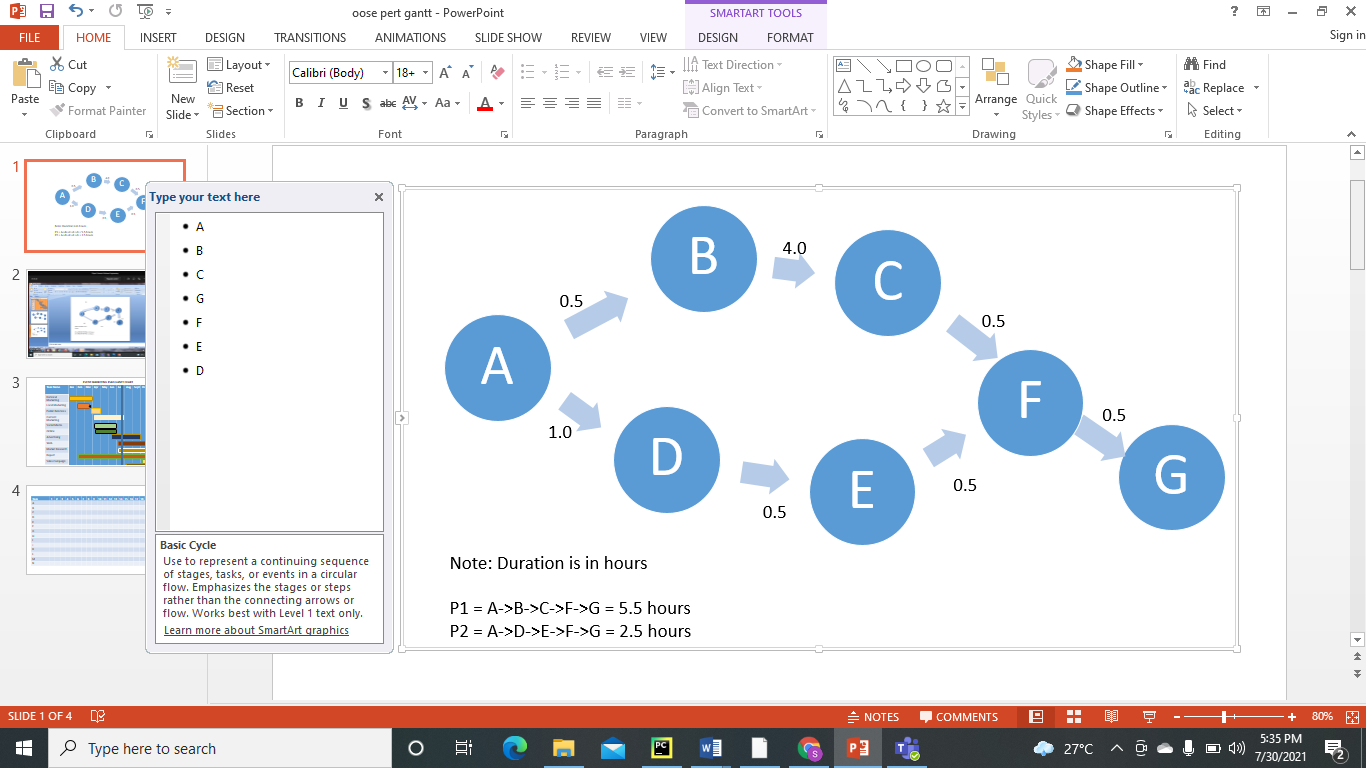
HP2 Create a deployment diagram for your project?



IP1 Create a PERT Chart on Powerpoint for the following

|  |  |  |  |
| --- | --- | --- | --- |
| Activity | Description | Predecessor | Estimated Time (hours) |
| A | Drive Home | None | 0.5 |
| B | Wash Clothes | A | 4.0 |
| C | Pack | B | 0.5 |
| D | Go to Bank | A | 1.0 |
| E | Pay Bill | D | 0.5 |
| F | Pack Car | C,E | 0.5 |
| G | Drive to Bus | F | 0.5 |

**PERT CHART**



|  |  |  |  |
| --- | --- | --- | --- |
| Task | Prec Task | Description | Time (in hrs) |
| A | None | Decide on Date for the Party | 1 |
| B | A | Book a bouncy Castle | 1 |
| C | A | Send Invitations | 5 |
| D | C | Receive Replies | 7 |
| E | D | Buy Toys and Balloons | 1 |
| F | D | Buy Food | 3 |
| G | E | Blow up Balloons | 2 |
| H | F | Make Food | 1 |
| I | H, G | Decorate | 1 |
| J | B | Get Bouncy Castle | 1 |
| K | J, I | Have Party | 2 |
| L | K | Clean Up | 4 |
| M | K | Send Back Bouncy Castle | 1 |
| N | M | Donate Unwanted Gifts | 3 |

IP2

* + label hours 0 to 30 across the horizontal axis
  + Mark a review stage at hour 14 to monitor the progress
  + Assume and illustrate that tasks A, B, C and D have been completed at hour 14
  + State which tasks are ahead and which tasks are behind schedule

Make a Gantt chart for the above?

