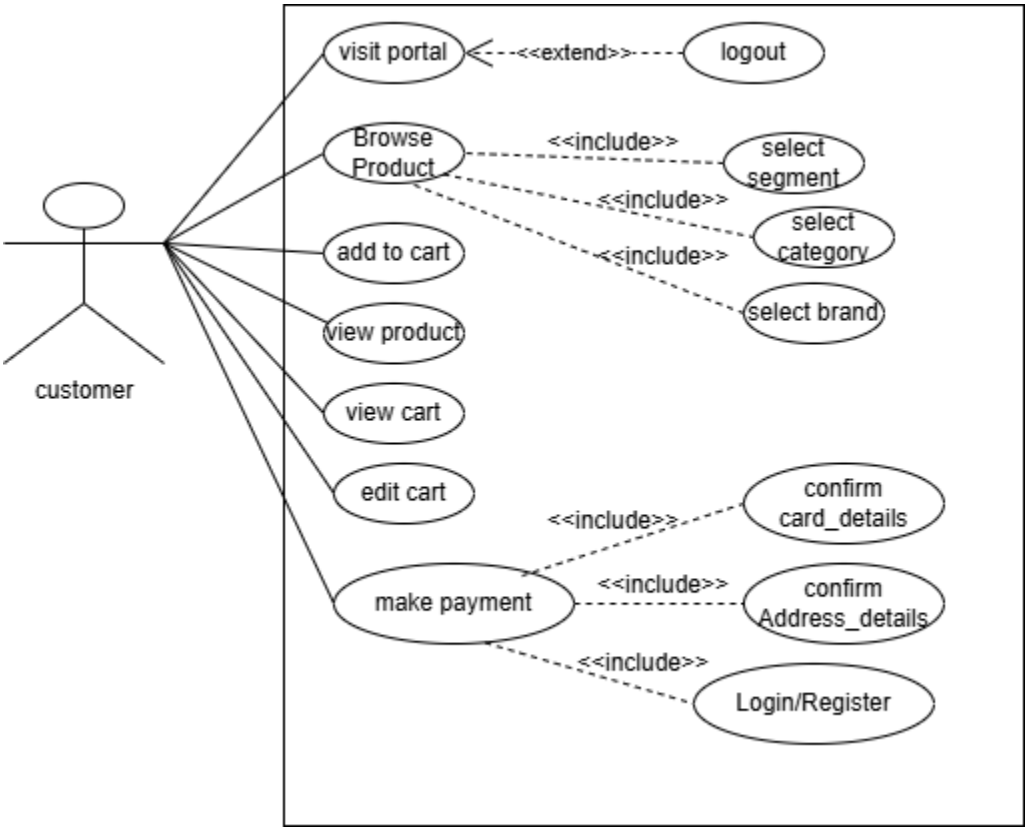
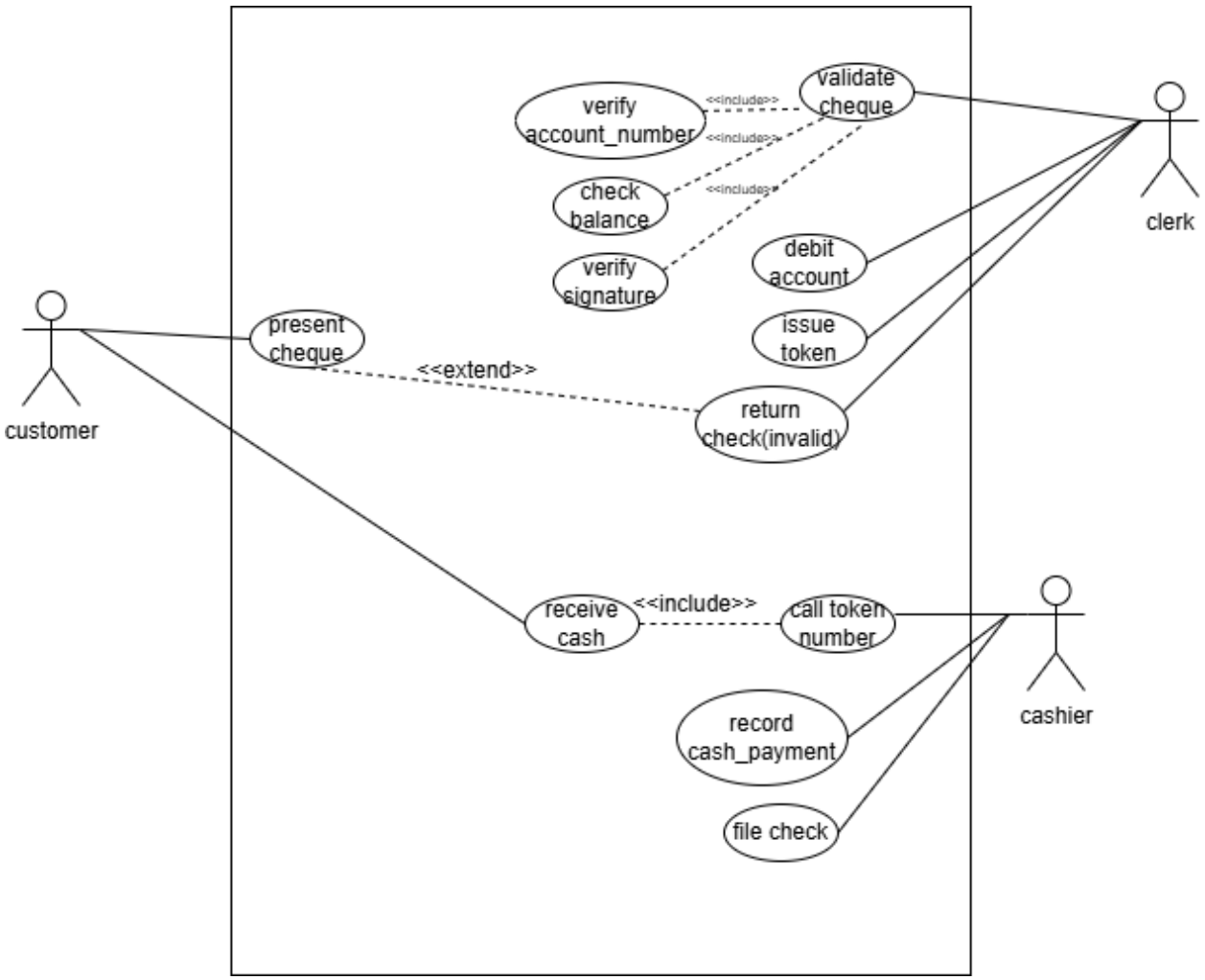
Q 1. Draw a use case diagram where A customer visits online shopping portal. A customer may buy item or just visit the page and logout. The customer can select a segment, then a category and brand to get different products in the desired band. The customer cam select product for purchasing. The process can be repeated for more items. Once the customer finishes selecting the product/s, the cart cane be viewed. If the customer wants to edit the final cart it can be done here. For final payment, the customer has to login to portal. If the customer is visiting for the first time, he must register with the site, else the customer use login page to proceed. Final cart is submitted for payment and card details and address details are to be confirmed with customer. Customer is confirmed with the shipment id and delivery if goods within 15 days.

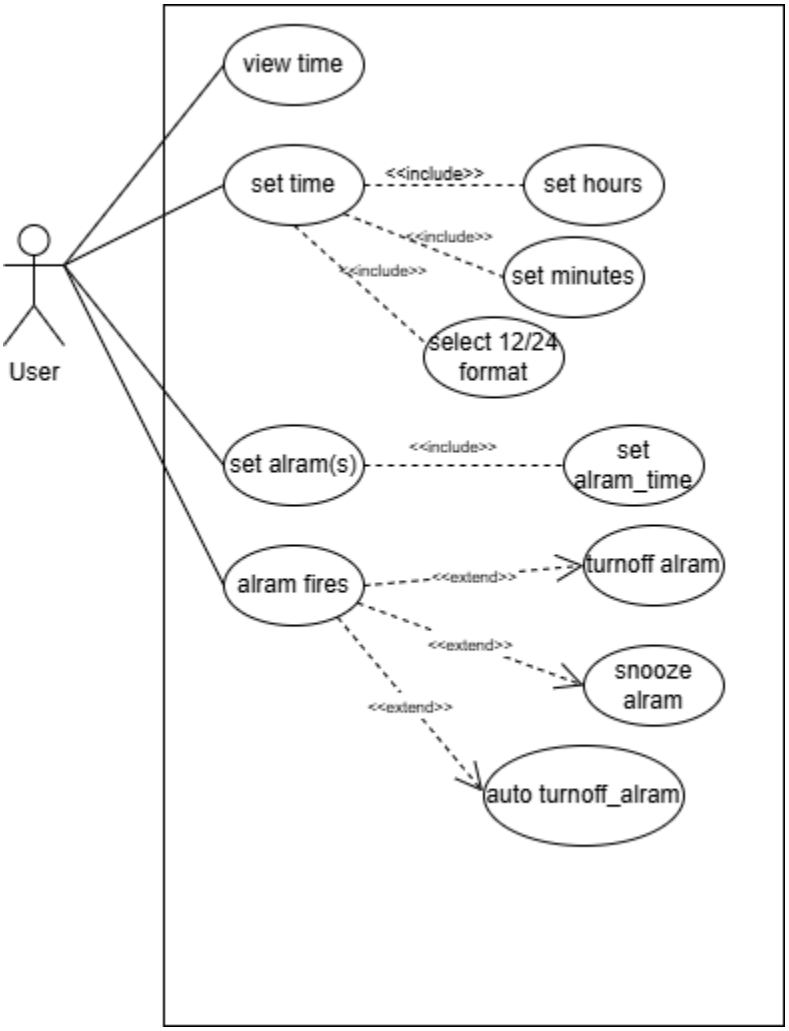


Q.2 A customer presents a cheque to a clerk. The clerk checks the ledger containing all account numbers and make sure whether the account number in the cheque is valid, whether adequate balance is there in the account to pay the cheque, and whether the signature is authentic. Having done these, the clerk gives the customer a token. The clerk also debits customer's account by the amount specified on the cheque. If cash cannot be paid due to an error in the cheque, the cheque is returned. The token number is written on the top of the cheque and it is passed on to the cashier. The cashier calls out the token number, takes the customer's signature, pays cash, enters cash paid in ledger called day book, and file the cheque.

Derive use-cases from the above scenario and model them into a Use-case diagram.



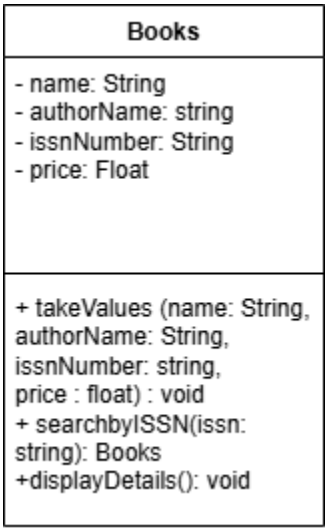
Q.3 Suppose you want to develop software for an alarm clock. The clock shows the time of day. Using buttons, the user can set the hours and minutes fields individually, and choose between 12 and 24-hour display. It is possible to set one or two alarms. When an alarm fires, it will sound some noise. The user can turn it off, or choose to snooze. It the user does not respond at all, the alarm turns off itself after 2 minutes. Snoozing means to turn off the sound, but the alarm will fire again after some minutes of delay. This snoozing time is pre adjusted. Draw use case for this system.



Q.4 Design a class diagram for class Books which has following attributes and operations:

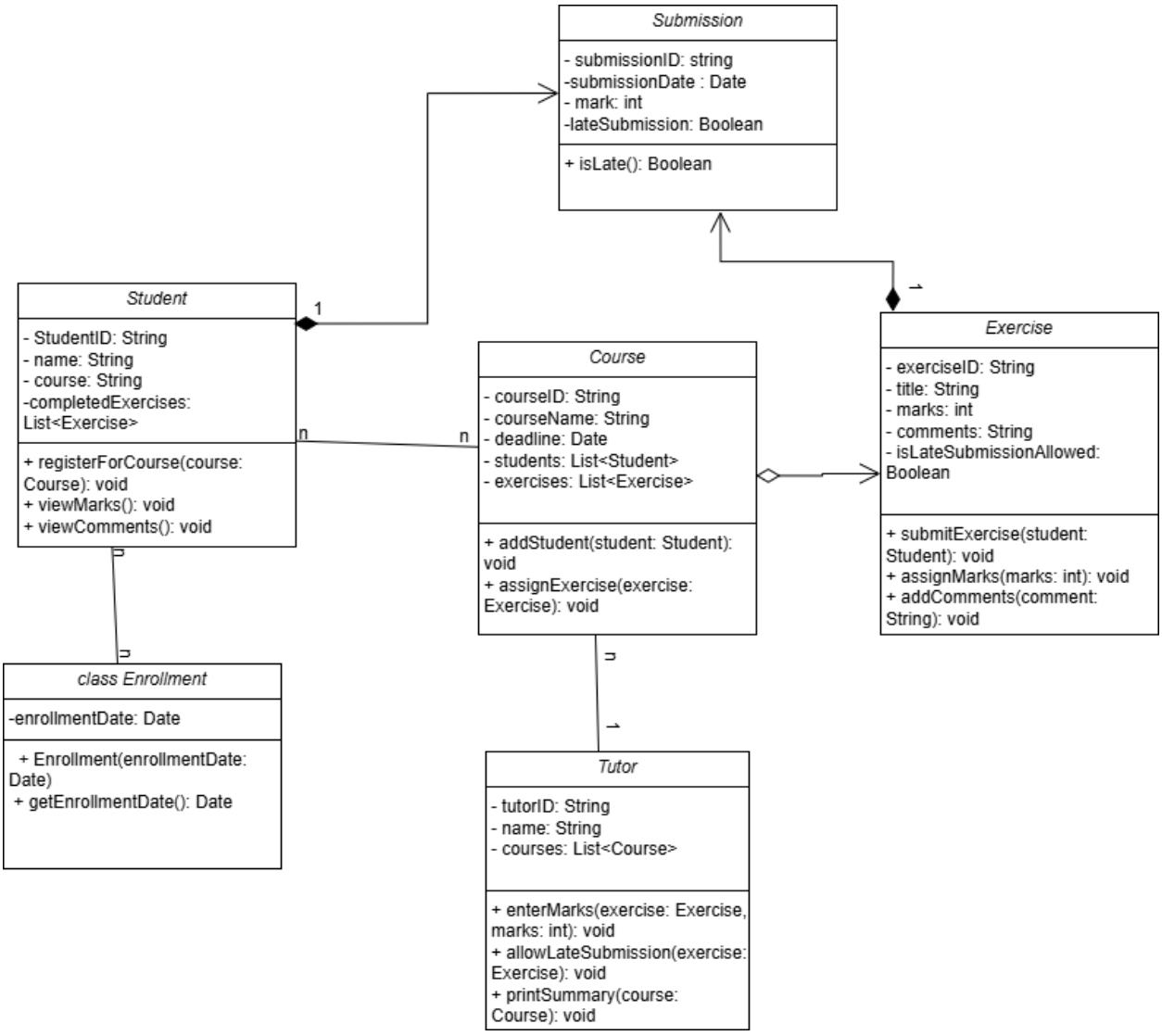
Attributes: name, Author Name, ISSN Number, Price

Operations: to take value for the above data, to search a book by its ISSN and to display the details of a book.

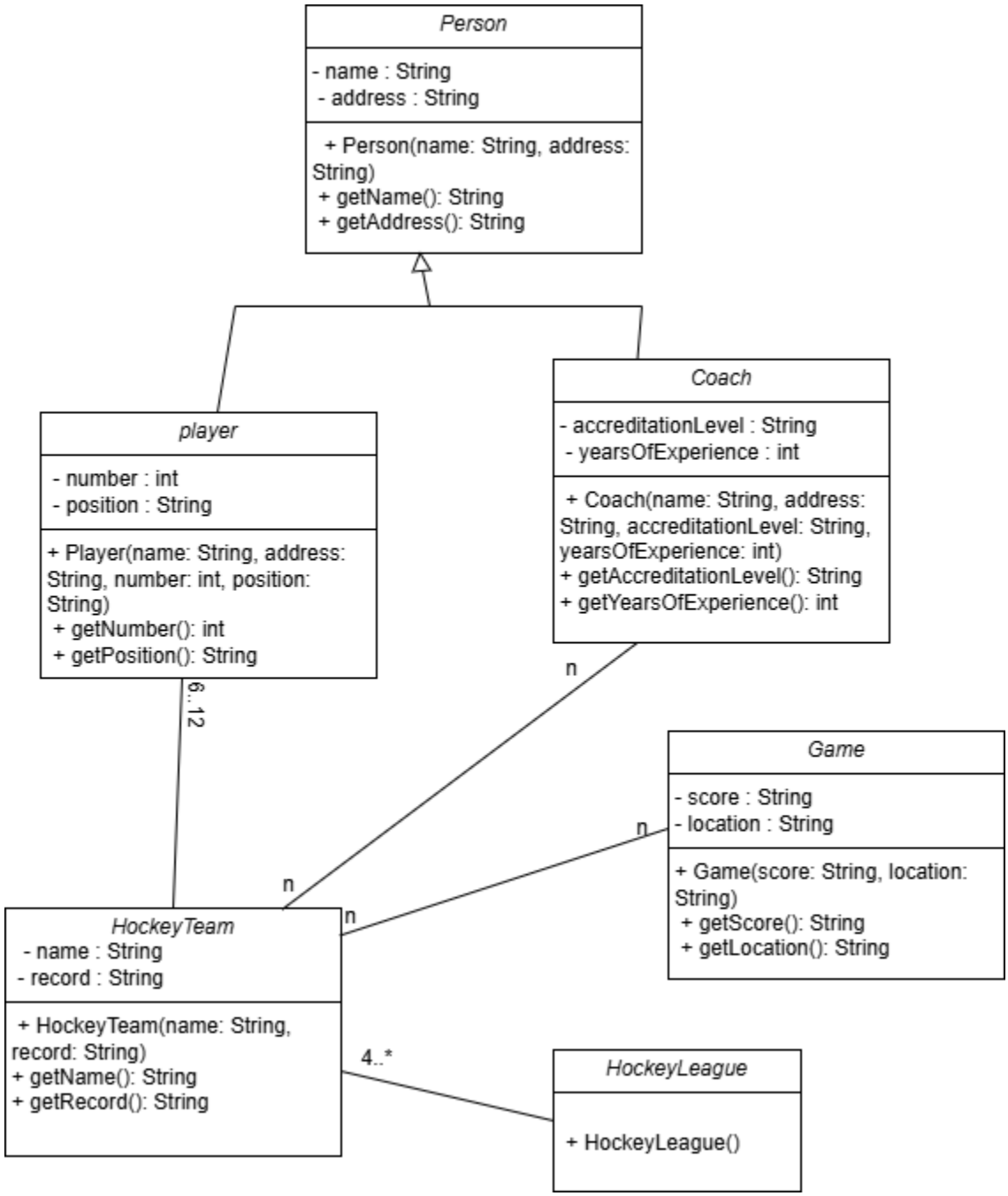


Q 5. A simple system is to be developed to support the management of exercises completed by students taking a course. Students first meet with course tutor to register for a course, and then during the course they submit a number of exercises. Every course has a certain deadline assigned by the course tutor. Tutors can allow an exercise to be submitted late. At any point, a student can find out from the system the marks they have received for any exercises already completed. A student shall also be able to view any comments made by the tutor on certain exercise. The course tutor can also enter a mark for an exercise, and print out a summary of the marks gained by all students on course.

Identify classes and draw a class diagram to model an efficient solution for the problem.

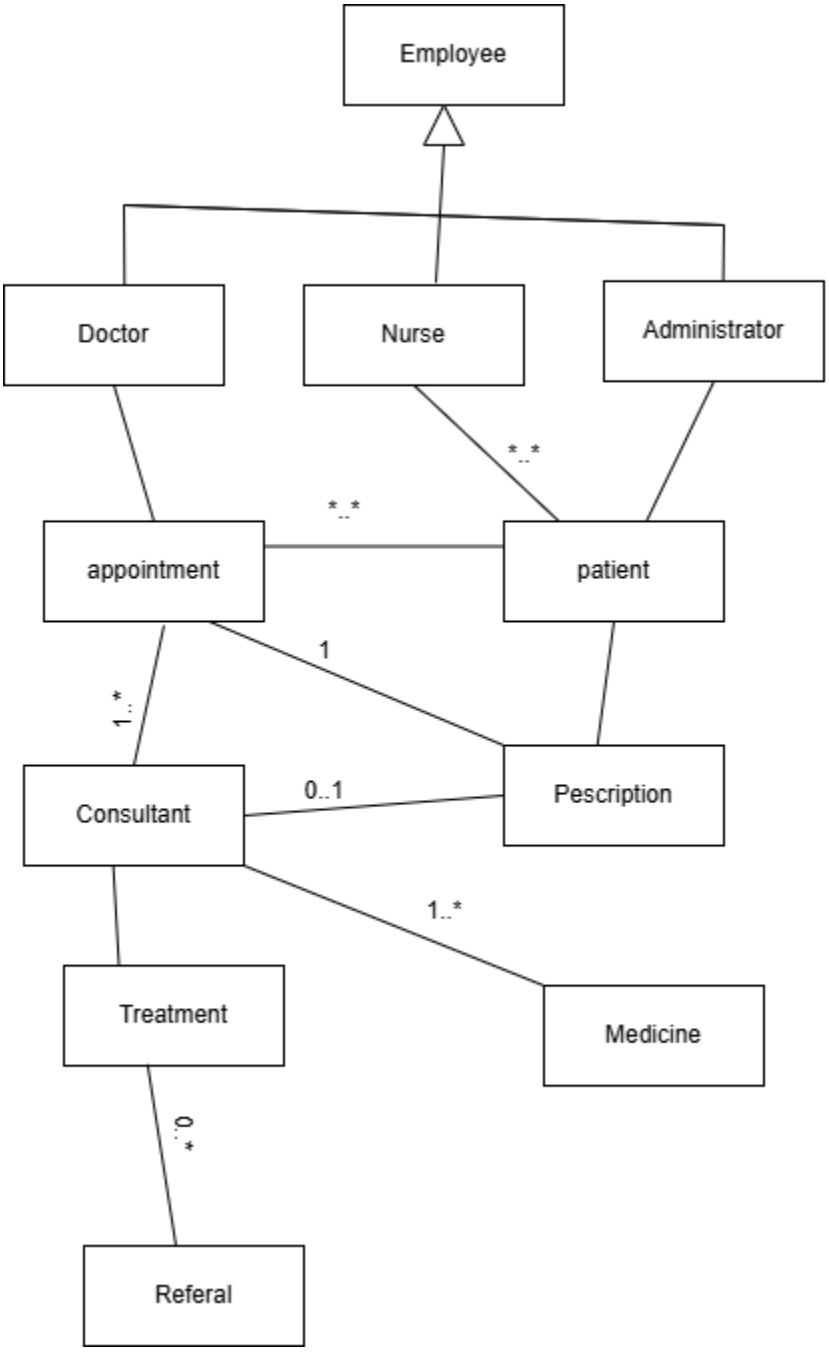


Q 6. Draw a UML class diagram representing the following elements from the problem domain for a hockey league. A hockey league is made up of at least four hockey teams. Each hockey team is composed of 6 to 12 players, and one player captains the team. A team has a name and a record. Players have a number and a position. Hockey team play games against each other. Each game has a score and a location. Teams are sometimes led by a coach. A coach has a level of accreditation and a number of years of experience, and can coach multiple teams. Coaches and players are people, and people have names and addresses. Draw a class diagram for this information, and be sure to label all associations with appropriate multiplicities.



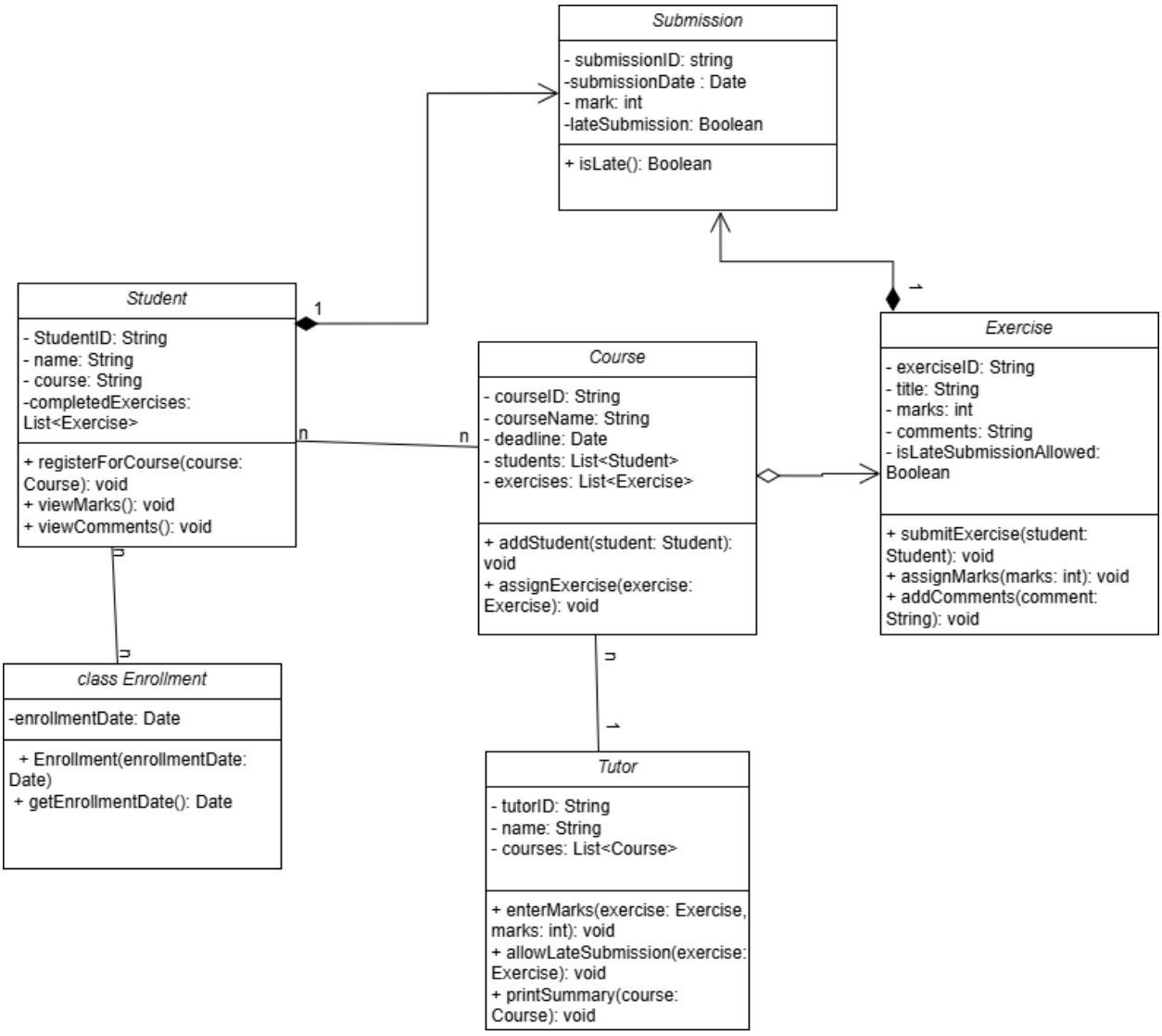
Q.7A health clinic provides medical services to patients in a small town. Five doctors and three nurses work at the clinic. They consult with patients, prescribe medicines and carry out minor medical treatments. Patients with more serious conditions are referred to specialists at the local hospital. A medical information system is being designed for the use in the clinic. The system will manage information about employees (doctors, nurses and administrator), patients and their contact details, appointments and consultations, medicines and prescriptions, treatments given, and referrals.

Produce a UML class diagram for use in constructing the system using an object-oriented programming language. Your diagram must include all applicable classes and relationships. There is no need to show the attributes and operations for each class.



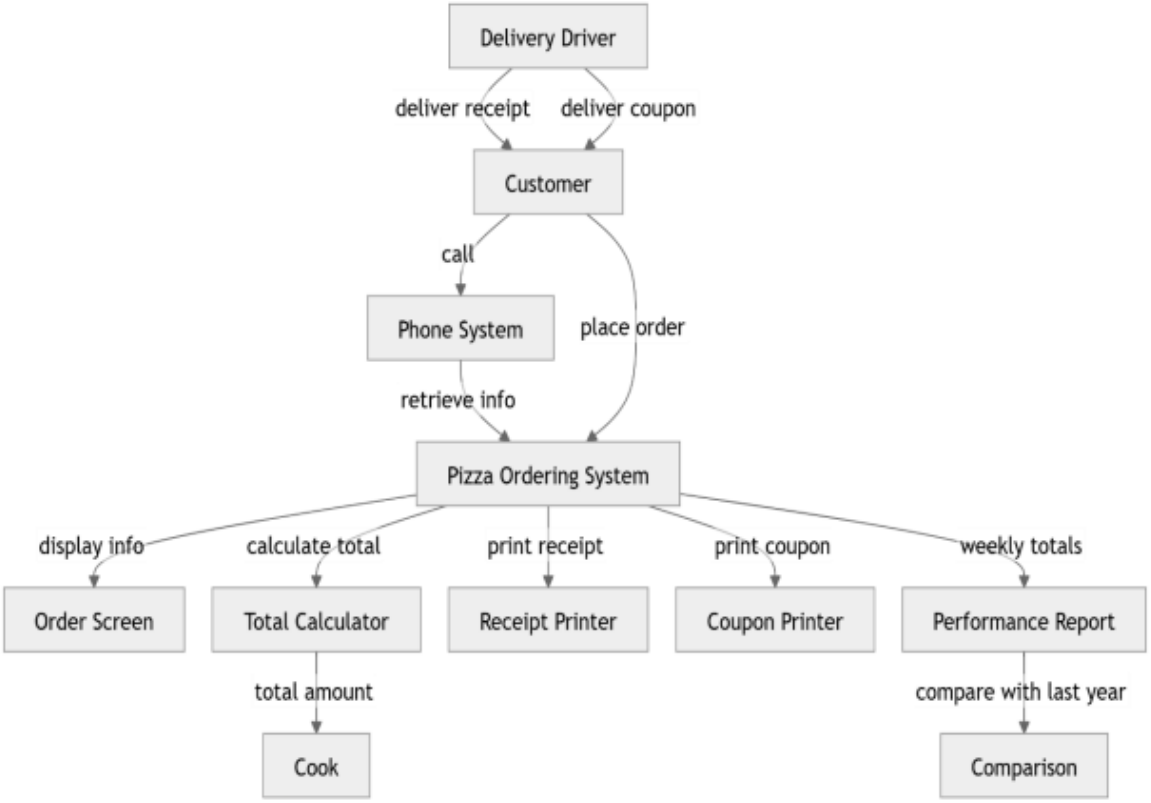
Q.8 A simple system is to be developed to support the management of exercises completed by students taking a course. Students first meet with course tutor to register for a course, and then during the course they submit a number of exercises. Every course has a certain deadline assigned by the course tutor. Tutors can allow an exercise to be submitted late. At any point, a student can find out from the system the marks they have received for any exercises already completed. A student shall also be able to view any comments made by the tutor on certain exercise. The course tutor can also enter a mark for an exercise, and print out a summary of the marks gained by all students on course.

Identify classes and draw a class diagram to model an efficient solution for the problem.



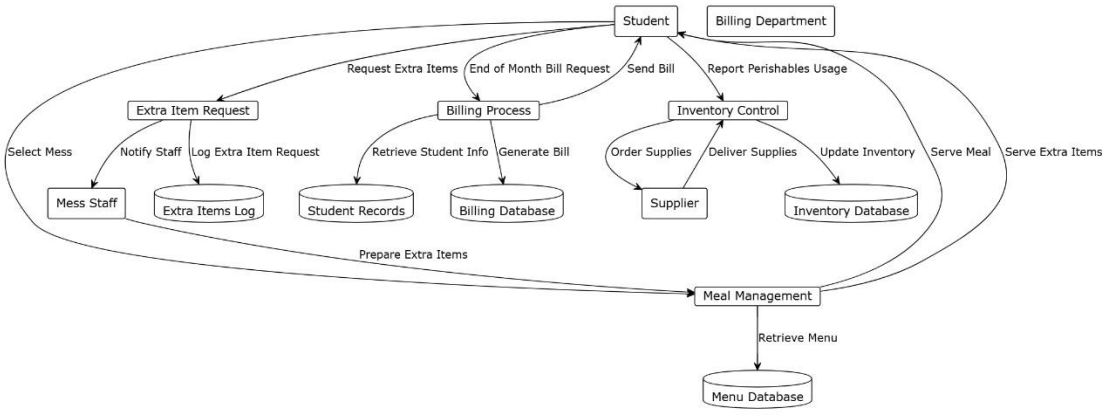
Q 9. Prepare level 1 DFD for the following Food Ordering System. [2020 fall]

KFC pizza wants to install a system to record orders for pizza and burger. When regular customer call KFC pizza on the phone, their phone number goes automatically into pizza system. The phone number invokes the name, address and last order date automatically on the screen. Once the order is taken, the total including tax and delivery is calculated. The order is given to the cook. A receipt is printed. Occasionally, special offers (coupons) is printed so the customer can get discount. Drivers who make deliveries gives customers a copy of the receipt and coupon (if any). Weekly totals are kept for comparison with last year's performance.



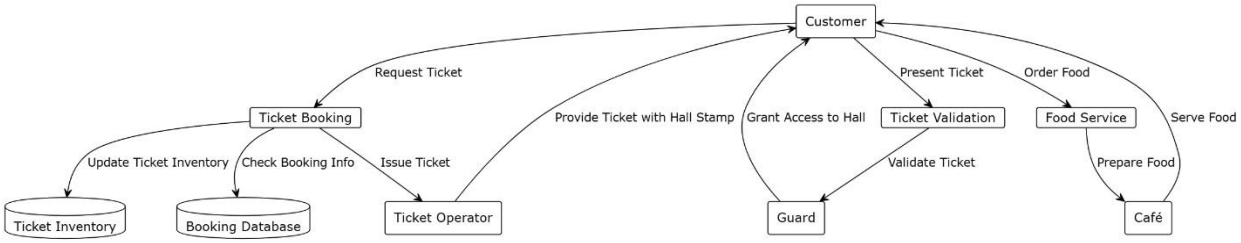
Q.10 Obtain **DFD** for the following Mess management system: [2014 Spring]

A hostel has 500 rooms and 4 messes. Currently, there are 1000 students in all in 2 seated rooms. They eat in any of the messes but can get rebate if they inform and do not eat for at least 4 consecutive days. Besides normal menu, extra items are also given to students when they ask for it. Such extras are entered in an extra book. At the end of the month, a bill is prepared based on the normal daily rate and extras and given to each student. System for stores issue and control is maintained for daily use of perishables and non-perishables items and order to vendor and suppliers are also maintained as well.

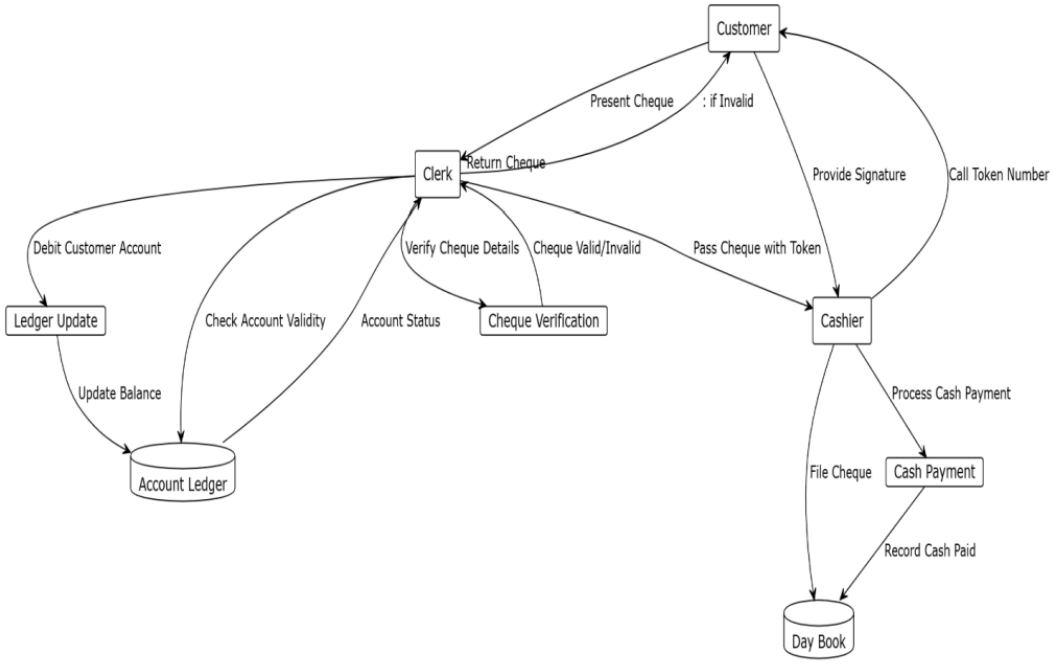


Q11. Obtain **1-level DFD** for Movie Management System: [2015 spring]

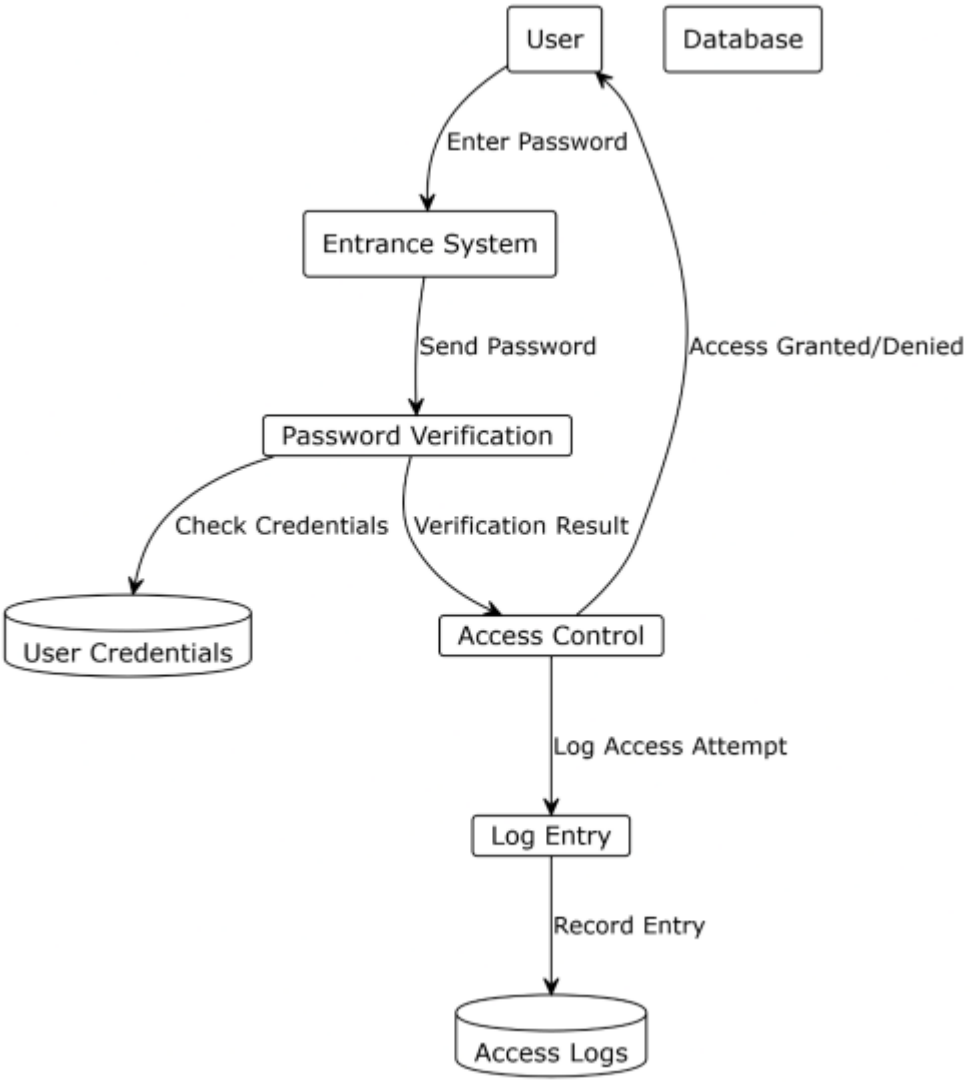
A customer can book a ticket from the Internet or can directly buy the ticket in the Movie-hall itself. There can be multiple halls within one movie theatre. The ticket operator provides a ticket with hall's stamp after checking the booking information to the customer. The guard in each hall validates the ticket and provides access to the customer inside the hall. There is also provision of complementary food item which the café will provide in the break time of the movie.



Q.12Obtain 1-level **DFD** for the following system of encashing cheque in a bank. A customer presents a cheque to a clerk. The clerk checks the ledger containing all account numbers and make sure whether the account number in the cheque is valid, whether adequate balance is there in the account to pay the cheque, and whether the clerk also debits customer's account by the amount specified on the cheque. If cash cannot be paid due to an error in the cheque, the cheque is returned. The token number is written on the top og the cheque and it is passed on to the cashier. The cashier calls out the token number, takes the customer's signature, pays cash, enters cash paid in ledger called day book, and file the cheque. [2016 spring, 2014 fall]

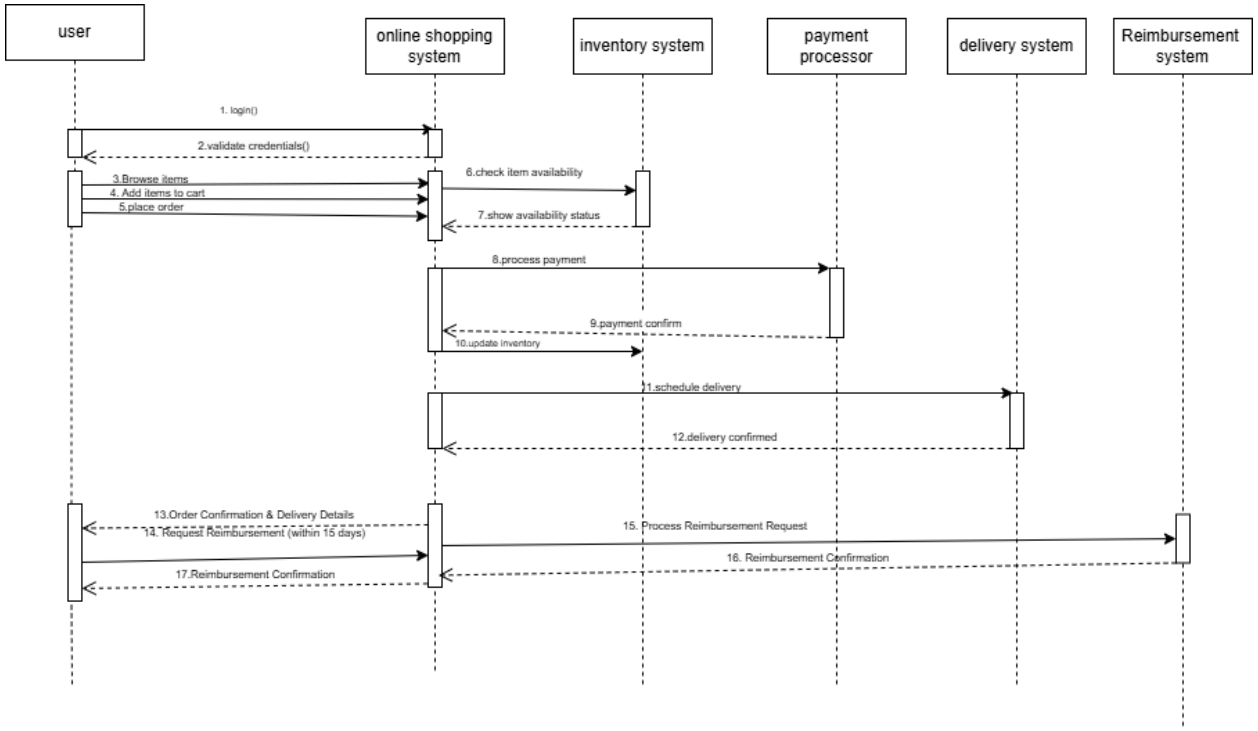


Q13. Draw the different levels of **DFD** for Safe Home System where any person can enter to the home on matching his/her password at the entrance door. [2018 fall]



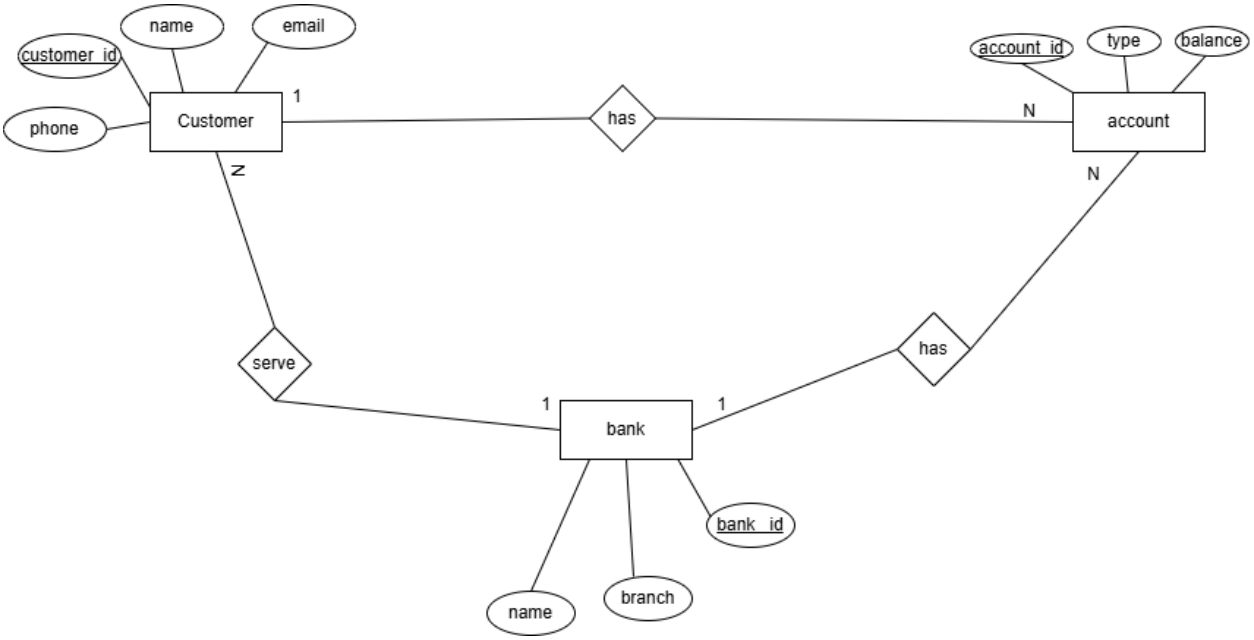
Q.14. Draw a sequence diagram of the following:

A valid user can login to the online shopping system. The user can order the items and as well as pay the Credit card. The items will be delivered later. The items within 15 days will be reimbursed.



Q16. Draw ERD for the following situation: [2014 Spring]

An accountant is a relationship between customer and bank. A customer has name. a bank has a branch. A customer may have several accountants of different types and balances.



Q17A restaurant uses an Information system that takes customer orders, send the order to the kitchen monitors: the goods sold and inventory and generates reports for management. List functional and Non-functional requirements for this Restaurant Information System.

• **Functional Requirements:** Order Management: Order should be possible from the customers' side. The order should reach the kitchen instantly. Inventory Management: Track how much ingredient is available and remind when the quantity is low. Update the quantity according to sales. Reporting: Daily, weekly, and monthly sales reports should be generated. Payment Processing: Support cash, card, online payment. Customer Notifications: Should notify the customers when their order is ready.

• **Non-Functional Requirements:** Performance: The system should not get lagged up during peak hours. Scalability: As the business is growing, increase the customer order. Usability: High usability for both the staff and customers. Availability: Maximally reduces downtime Security: Customer data and payment details are secured

.

Q 18. Why User interface design is important in software development? Referencing a mobile application for smart agriculture, describe user interface design issues.

User Interface (UI) design is critical in software development because it directly influences the user experience (UX). A well-designed UI ensures that the application is intuitive, visually appealing, and easy to use, which can lead to higher user satisfaction, better engagement, and increased productivity. Conversely, poor UI design can lead to frustration, errors, and even abandonment of the software. For mobile applications, where screen space and user attention are limited, a good UI is even more crucial to ensure seamless interactions.

Referencing a Mobile Application for Smart Agriculture

A mobile application for smart agriculture might include features like crop monitoring, irrigation scheduling, weather updates, pest control alerts, and soil health tracking. The success of such an app depends heavily on its UI design, as farmers with varying technical expertise need to use it efficiently.

UI Design Issues in a Smart Agriculture Mobile Application

- **Complex Navigation**
    - **Issue**: A poorly structured navigation system with too many nested menus or unclear labels can confuse users. For example, if a farmer struggles to find weather data or irrigation settings, the app becomes less useful.
    - **Solution**: Use simple and hierarchical navigation with self-explanatory icons and labels. Include a prominent search feature for quick access.
- **Overloaded Information Display**
    - **Issue**: Displaying too much data (e.g., multiple charts or tables for crop health) on a small mobile screen can overwhelm users.
    - **Solution**: Prioritize essential information, use collapsible sections, and employ visual cues like graphs or progress bars for clarity.
- **Low Accessibility for Non-Tech-Savvy Users**
    - **Issue**: Farmers in rural areas may lack advanced technological skills or literacy, making a complex interface challenging.
    - **Solution**: Use simple language, provide multi-language support, and include audio cues or tutorials for guidance.
- **Poor Responsiveness**
    - **Issue**: If the app lags or does not adapt well to different screen sizes, users may abandon it.

    - **Solution**: Ensure the app is responsive, lightweight, and optimized for low-end devices commonly used in rural areas.

- **Inconsistent Visual Design**

    - **Issue**: Inconsistent fonts, colors, or button designs can make the app feel unprofessional and harder to navigate.
    - **Solution**: Maintain a consistent design system with a harmonious color palette, legible fonts, and standardized button styles.

Q.19What is **Class Responsibility Collaborator** model? How do you develop CRC model? Explain with example. [2013 spring]

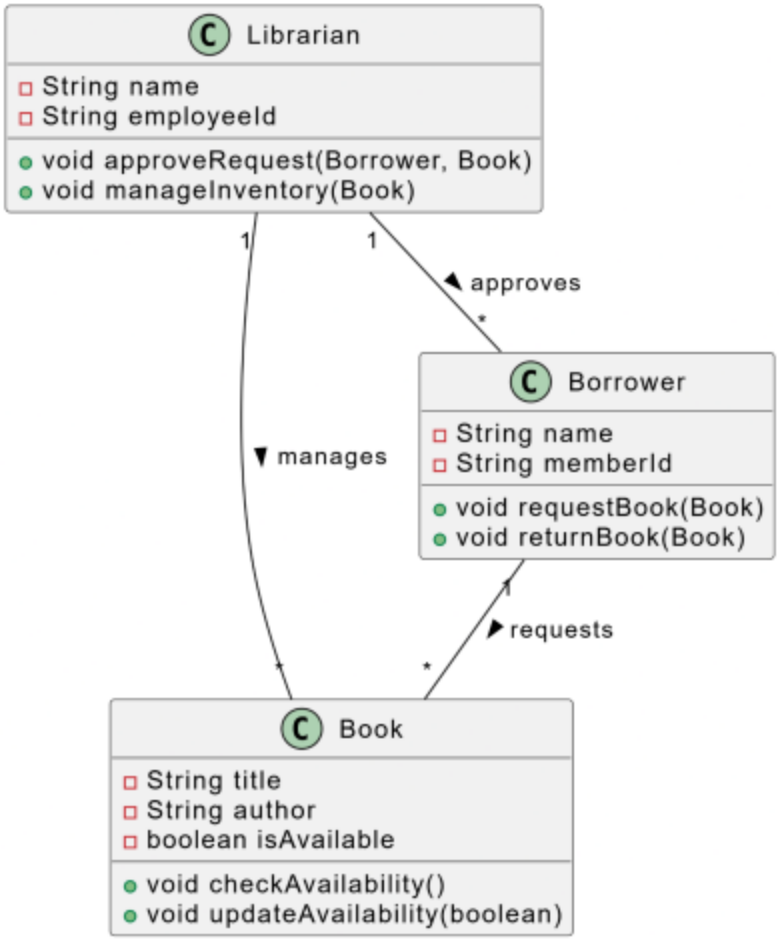
The **Class-Responsibility-Collaborator (CRC)** model is a conceptual tool used in object-oriented design to identify and describe classes, their responsibilities, and their collaborators. It helps in defining the structure of a software system and ensuring that each class has a focused purpose and clear interactions with other classes.

- **Class**: Represents a logical grouping of objects with similar attributes and behaviors.
- **Responsibility**: Refers to the tasks or duties that the class is responsible for performing.
- **Collaborator**: Refers to other classes that the given class interacts with to fulfill its responsibilities.

How to Develop a CRC Model

1. **Identify Classes**: Identify potential classes that the system will need based on the problem domain.
2. **Assign Responsibilities**: Determine the main responsibilities for each class. Responsibilities typically include knowing something (data) or doing something (actions).
3. **Define Collaborators**: Identify other classes that the class depends on to fulfill its responsibilities.
4. **Represent on CRC Cards**: Use index cards (physical or digital) to document each class. Each card contains:
    - The class name
    - Its responsibilities
    - Its collaborators
5. **Refine and Validate**: Review the CRC cards with the development team to refine class design and interactions.

Example: CRC


Librarian
String name
String employeeId
void approveRequest(Borrower, Book)
void manageInventory(Book)
1
1
approves
*
manages
Borrower
String name
String memberId
void requestBook(Book)
void returnBook(Book)
1
requests
*
*
Book
String title
String author
boolean isAvailable
void checkAvailability()
void updateAvailability(boolean)