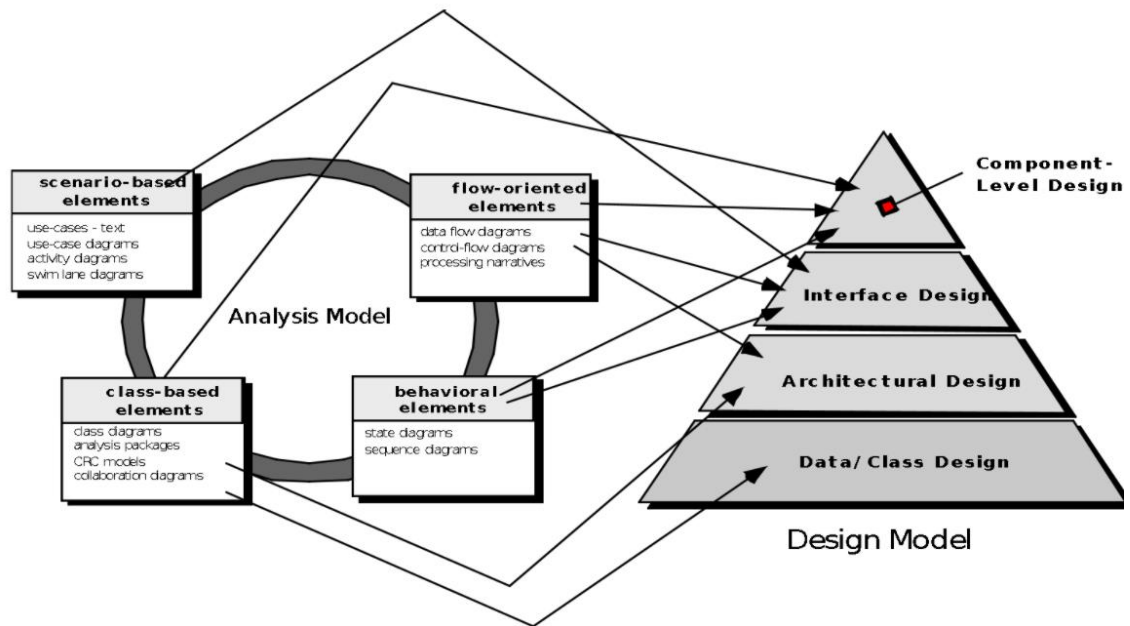


UNIT – III

1. Explain with a neat diagram the flow of Analysis model into design model.

Ans:

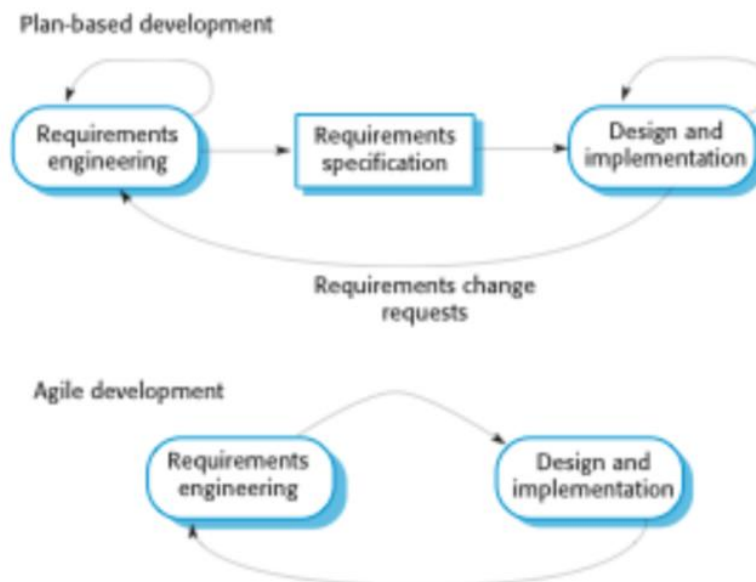
Analysis Model -> Design Model



- ✧ The data/class design transforms class models into design class realizations and the requisite data structures required to implement the software. The objects and relationships defined in the CRC diagram and the detailed data content depicted by class attributes and other notation provide the basis for the data design action.
- ✧ The architectural design defines the relationship between major structural elements of the software, the architectural styles and design patterns that can be used to achieve the requirements.
- ✧ The interface design describes how the software communicates with systems that interoperate with it, and with humans who use it. An interface implies a flow of information (e.g., data and/or control) and a specific type of behavior.
- ✧ The component-level design transforms structural elements of the software architecture into a procedural description of software components.

2. Compare and contrast in between Agile method and iterative development process.

Ans:



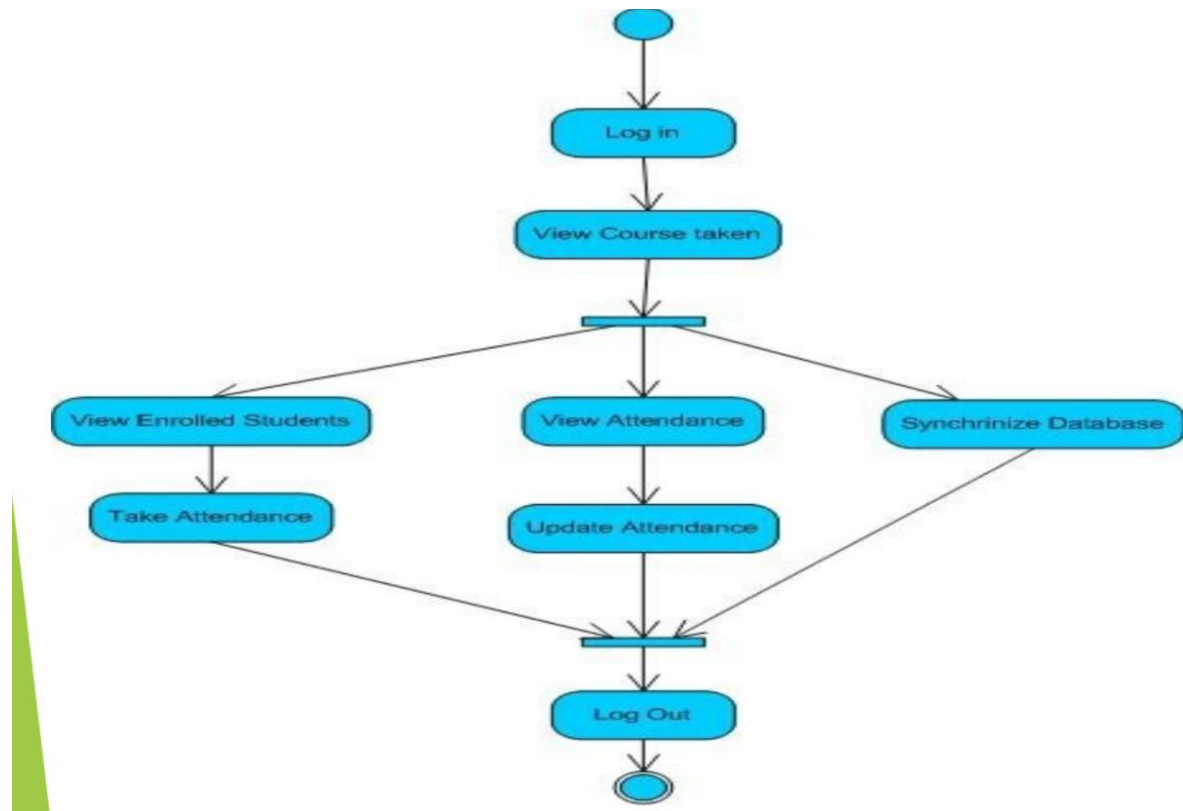
Technical, human, organizational issues

- Most projects include elements of plan-driven and agile processes. Deciding on the balance depends on:
 - Is it important to have a very detailed specification and design before moving to implementation? If so, you probably need to use a plan-driven approach.
 - Is an incremental delivery strategy, where you deliver the software to customers and get rapid feedback from them, realistic? If so, consider using agile methods.
 - How large is the system that is being developed? Agile methods are most effective when the system can be developed with a small co-located team who can communicate informally. This may not be possible for large systems that require larger development teams so a plan-driven approach may have to be used.

- What type of system is being developed?
 - Plan-driven approaches may be required for systems that require a lot of analysis before implementation (e.g. real-time system with complex timing requirements).
 - What is the expected system lifetime?
 - Long-lifetime systems may require more design documentation to communicate the original intentions of the system developers to the support team.
 - What technologies are available to support system development?
 - Agile methods rely on good tools to keep track of an evolving design
 - How is the development team organized?
 - If the development team is distributed or if part of the development is being outsourced, then you may need to develop design documents to communicate across the development teams.
-
- Are there cultural or organizational issues that may affect the system development?
 - Traditional engineering organizations have a culture of plan-based development, as this is the norm in engineering.
 - How good are the designers and programmers in the development team?
 - It is sometimes argued that agile methods require higher skill levels than plan-based approaches in which programmers simply translate a detailed design into code
 - Is the system subject to external regulation?
 - If a system has to be approved by an external regulator (e.g. the FAA approve software that is critical to the operation of an aircraft) then you will probably be required to produce detailed documentation as part of the system safety case.

3. Draw an activity diagram (process perspective) for Attendance Management system of college.

Ans:



4. Explain different perspectives to develop various software design models.

Ans:

- ✧ An external perspective, where you model the context or environment of the system.
- ✧ An interaction perspective, where you model the interactions between a system and its environment, or between the components of a system.
- ✧ A structural perspective, where you model the organization of a system structure of data flow.
- ✧ A behavioral perspective, where you model the dynamic behavior of the system and how it responds to events.

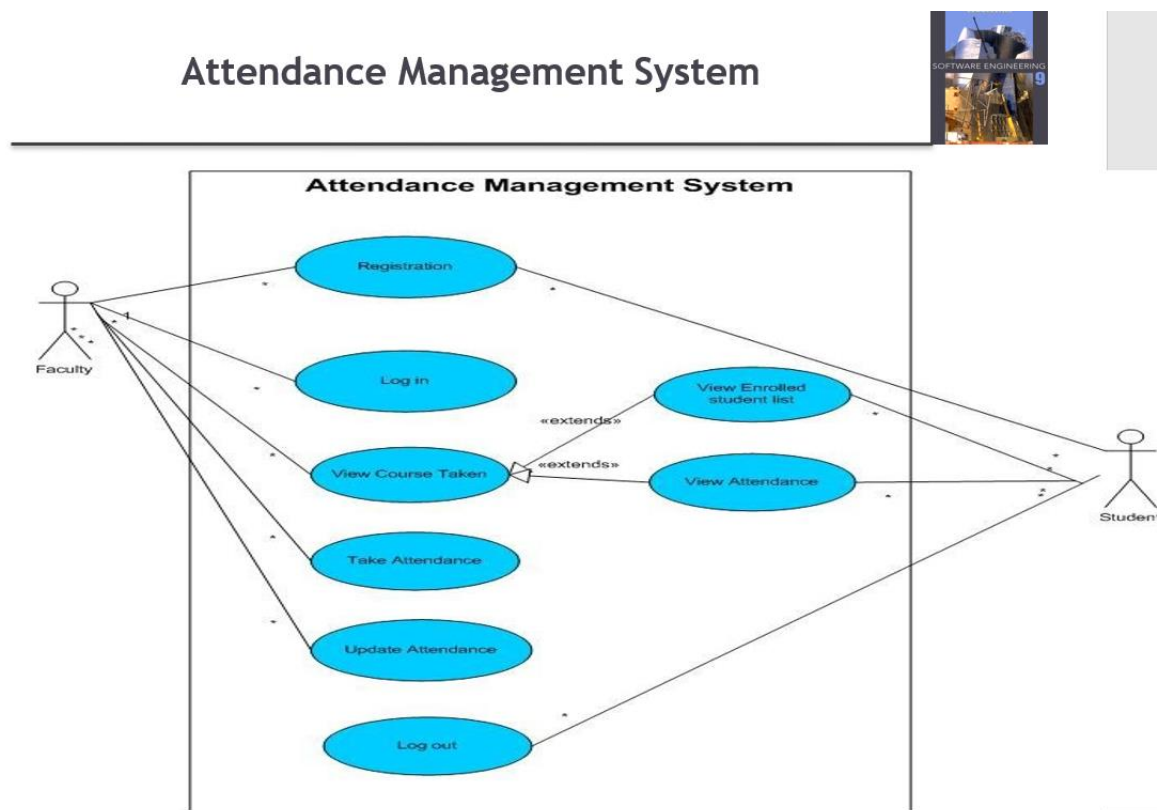
Examples:

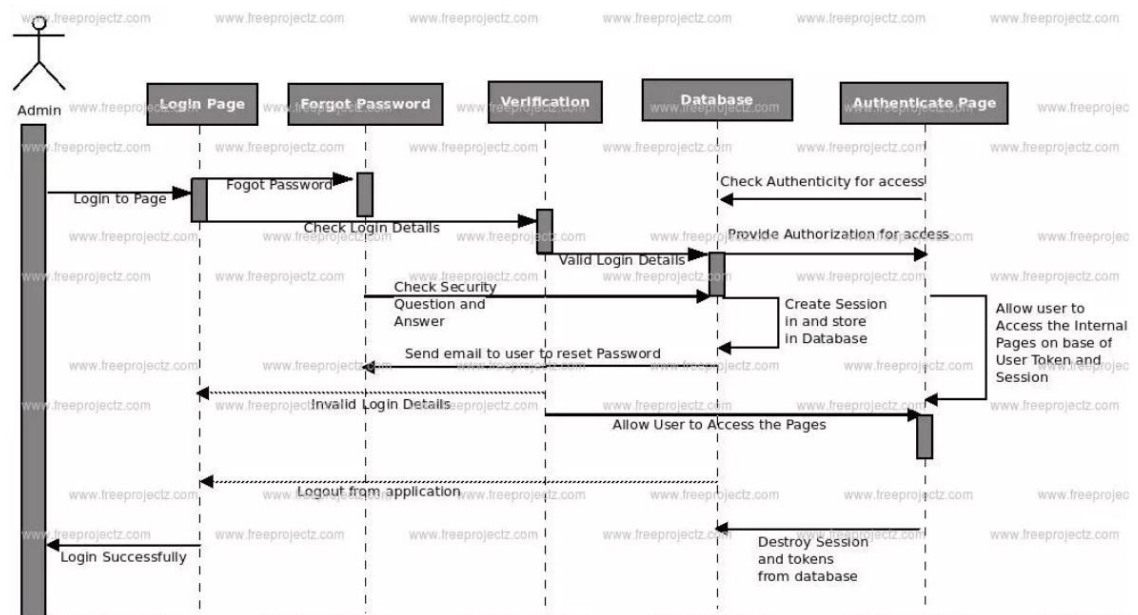
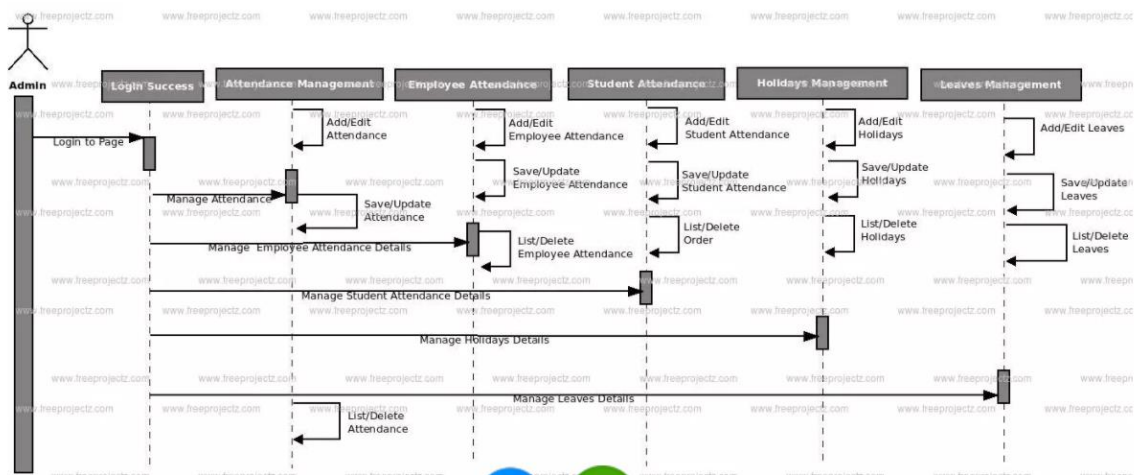
- i. External perspective: Context model.
- ii. Interaction perspective: Use case.
- iii. Structural perspective: Structural activity.
- iv. Behavioural perspective: Behaviour state model.

5. Design a use-case diagram and sequence diagram for Attendance Management System of college.

Ans:

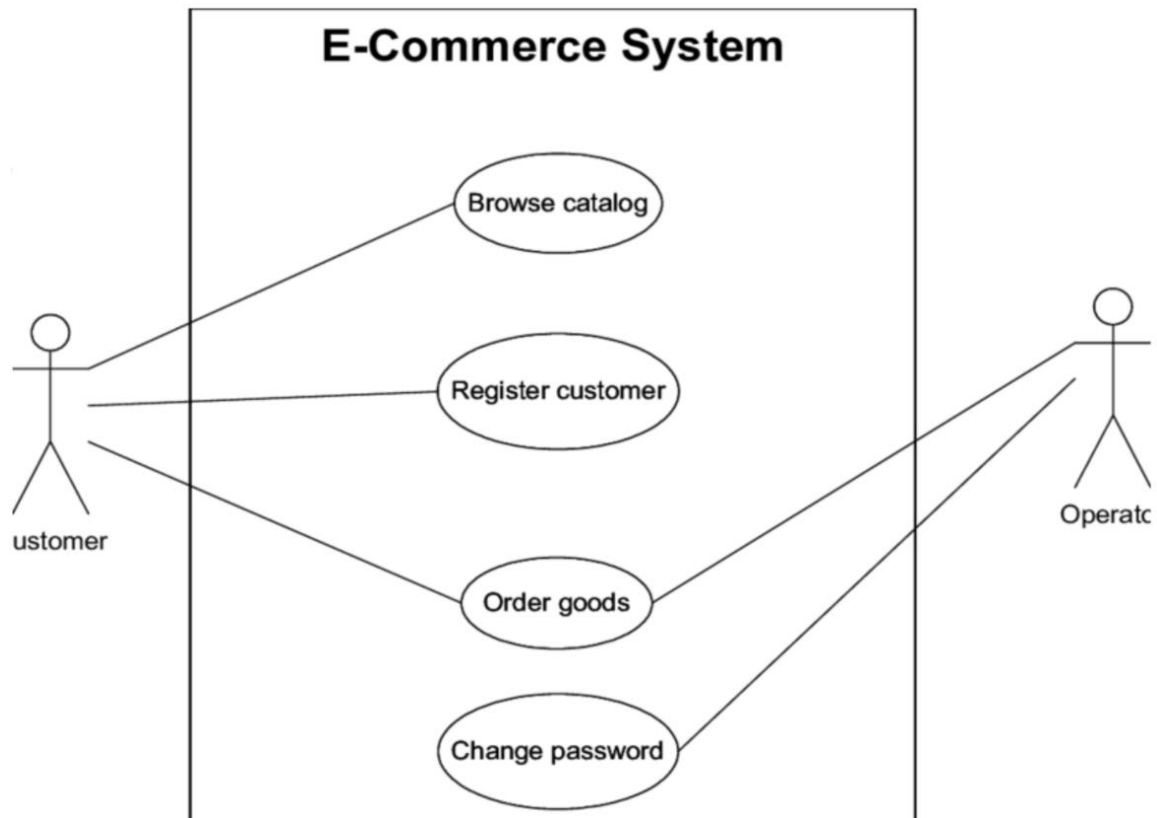
USE-CASE DIAGRAM





6. Develop a Use-case diagram for E-commerce system with tabular description.

Ans:



Use case name:	Browse Catalogue
Goal:	To explore the lists of goods available from the system.
Actor(s):	Customer
Preconditions:	The customer has access to the internet.
Main flow of events:	<ol style="list-style-type: none"> 1. The customer enters the e-commerce website. 2. The customer selects the Browse Catalogue section. 3. The system displays lists of products to the customer. 4. The customer browses the catalogue for a particular product. 5. The customer finds the product.
Alternate flows:	User cannot find product he/she wanted. Use case ends.
Post conditions:	The product is found.

Use case name:	Register customer
Goal:	To register a customer identity with the system.
Actor(s):	Customer
Preconditions:	The customer has access to the website. The customer has not registered before.
Main flow of events:	<ol style="list-style-type: none"> 1. The customer enters the Register Customer section. 2. The system displays the new customer registration form. 3. The customer provides registration details. 4. The customer submits the registration form. 5. The system updates its registration data information.
Alternate flows:	The customer has already registered. Use case ends.
Post conditions:	The customer is registered and the details are saved to a database.

Use case name:	Order goods
Goal:	To order goods from the system.
Actor(s):	Customer Operator
Preconditions:	The customer is registered to order goods. The customer has entered registration details e.g. user name and password (the customer is logged on to the ordering section).
Main flow of events:	<ol style="list-style-type: none"> 1. The customer enters Order Goods section. 2. The system displays the customer's account detail. 3. For each product that the customer wishes to order, the customer enters its identity. 4. The customer provides delivery details. 5. The system calculates and displays the price of the goods ordered. 6. The customer submits payment details. 7. The system confirms the result of transaction. 8. The operator collects the detail of the order. 9. The operator processes the order.
Alternate flows:	
Post conditions:	The order and its detail are entered on the system and the order is processed.

Use case name:	Change Password
Goal:	Change the password for the login
Actor(s):	Operator
Preconditions:	The operator is logged in.
Main flow of events:	<ol style="list-style-type: none"> 1. The operator enters his/her current password 2. The system validates the password. 3. The operator enters a new password, twice, as prompted by the system. 4. The operator confirms the change. 5. The system saves the new password.
Alternate flows:	A1. If the operator's old password is incorrect, an error message should be displayed and the password should not be changed. A2. If the two entries of the new passwords do not match, the operator is prompted to re-enter them.
Post conditions:	The password of the operator is changed and updated in the database.

7. List and explain the principles of Agile methods. Give its applicability and problems involved.

Ans:

The principles of agile methods

Principle	Description
Customer involvement	Customers should be closely involved throughout the development process. Their role is provide and prioritize new system requirements and to evaluate the iterations of the system.
Incremental delivery	The software is developed in increments with the customer specifying the requirements to be included in each increment.
People not process	The skills of the development team should be recognized and exploited. Team members should be left to develop their own ways of working without prescriptive processes.
Embrace change	Expect the system requirements to change and so design the system to accommodate these changes.
Maintain simplicity	Focus on simplicity in both the software being developed and in the development process. Wherever possible, actively work to eliminate complexity from the system.

Agile method applicability

- Product development where a software company is developing a small or medium-sized product for sale.
- Custom system development within an organization, where there is a clear commitment from the customer to become involved in the development process and where there are not a lot of external rules and regulations that affect the software.
- Because of their focus on small, tightly-integrated teams, there are problems in scaling agile methods to large systems.

Problems with agile methods

- It can be difficult to keep the interest of customers who are involved in the process.
- Team members may be unsuited to the intense involvement that characterizes agile methods.
- Prioritizing changes can be difficult where there are multiple stakeholders.
- Maintaining simplicity requires extra work.
- Contracts may be a problem as with other approaches to iterative development , Many organizations especially large companies have spent years to changing their culture

8. Analyse the agile method to develop story card, 3 task cards and 2 test cards for credit card balance checking system in banking app. Stakeholders for a given system are
- i. Credit card holder
 - ii. Bank
 - iii. Bank employees

Ans:

Story Card

As a credit card holder, I want to view my statement balance, so that I can pay the balance due

- Display statement balance upon authentication
- Display Total Balance
- Show "Payment Due Date" and "Minimum Payment Due"
- Display Error message if service not responding/ timeout

Task Cards

1.

Display Statement balance upon Authentication

Authentication is necessary to check that the person trying to access the system is a valid user.

The user enters the password to access the system. The system then checks whether the entered password is correct or not.

Then the system verifies that the user has registered and has a valid account.

If he/she is not a valid user then the system asks them to verify their details.

2.

Show Payment Due Date and Minimum payment due

While checking the statement balance, the user can also check due date for the payment and the minimum balance that user should pay within the due date. This all information that is displayed on the app is added by bank employees.

3.

Display error message if not responding or time out

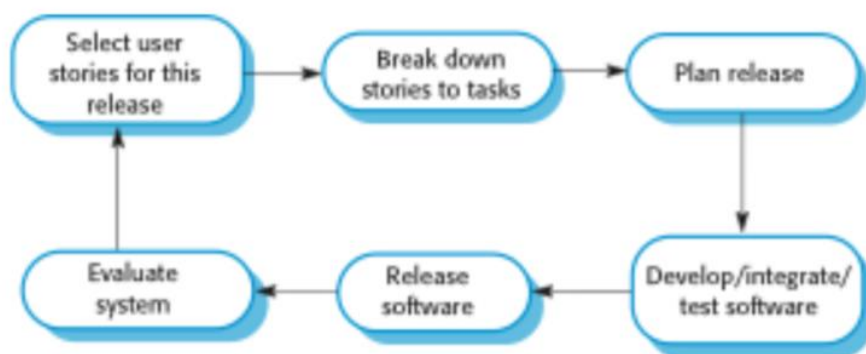
While the user is continuing with payment process, if there occurs a error or disturbance like software failure, service not found, hardware failure or it exceeds the time, then it should display a error with appropriate message.

Test Cards

Input:1	User Credit Card Number, username, password.
Input:2	Amount To be credited.
Test case1	Testing credit card is valid or invalid.
Test case2	Testing user name is correct password is invalid
Test case3	Testing password is correct user name is invalid
Test case4	Whether Minimum balance of amount is available or not.
Test case5	Once payment is done, its successfully updated or not
Test case6	Network connectivity, server availability,response time.
Output:	Successful amount payment or Error caused by system.

9. Briefly discuss the extreme programming release cycle with a neat diagram.

Ans:



In an XP process, customers are intimately involved in specifying and prioritizing system requirements. The requirements are not specified as lists of required system functions. Rather, the system customer is part of the development team and discusses scenarios with other team members. Together, they develop a 'story card' that encapsulates the customer needs. The development team then aims to implement that scenario in a future release of the software. An example of a story card for the mental

Figure 3.4 Extreme programming practices

health care patient management system is shown in Figure 3.5. This is a short description of a scenario for prescribing medication for a patient.

The story cards are the main inputs to the XP planning process or the 'planning game'. Once the story cards have been developed, the development team breaks these down into tasks (Figure 3.6) and estimates the effort and resources required for implementing each task. This usually involves discussions with the customer to refine the requirements. The customer then prioritizes the stories for implementation, choosing those stories that can be used immediately to deliver useful business support. The intention is to identify useful functionality that can be implemented in about two weeks, when the next release of the system is made available to the customer.

Of course, as requirements change, the unimplemented stories change or may be discarded. If changes are required for a system that has already been delivered, new story cards are developed and again, the customer decides whether these changes should have priority over new functionality.

10. Explain the practices followed in extreme programming.

Ans:

Extreme programming practices (a)

Principle or practice	Description
Incremental planning	Requirements are recorded on story cards and the stories to be included in a release are determined by the time available and their relative priority. The developers break these stories into development 'Tasks'.
Small releases	The minimal useful set of functionality that provides business value is developed first. Releases of the system are frequent and incrementally add functionality to the first release.
Simple design	Enough design is carried out to meet the current requirements and no more.
Test-first development	An automated unit test framework is used to write tests for a new piece of functionality before that functionality itself is implemented.
Refactoring	All developers are expected to refactor the code continuously as soon as possible code improvements are found. This keeps the code simple and maintainable.

Extreme programming practices (b)

Pair programming	Developers work in pairs, checking each other's work and providing the support to always do a good job.
Collective ownership	The pairs of developers work on all areas of the system, so that no islands of expertise develop and all the developers take responsibility for all of the code. Anyone can change anything.
Continuous integration	As soon as the work on a task is complete, it is integrated into the whole system. After any such integration, all the unit tests in the system must pass.
Sustainable pace	Large amounts of overtime are not considered acceptable as the net effect is often to reduce code quality and medium term productivity
On-site customer	A representative of the end-user of the system (the customer) should be available full time for the use of the XP team. In an extreme programming process, the customer is a member of the development team and is responsible for bringing system requirements to the team for implementation.

Chapter 3 Agile software development

19

11. Analyse the agile method to develop story card, 3 task cards and 2 test cards for website development of college. Stakeholders for a given system are
- College
 - Student
 - Faculty

Ans:

Story Card

It is very important to have website for a college.

Faculty members get the updates about the schedule of lectures, certain events etc.

Students get the update about Calendar of events, examinations, results, revaluations, or some circulars etc.

It contains brief information about all the faculties of all departments.

It also contains information regarding college, departments that can help freshers to have information about the college.

Task cards

1.

'Useful' Home page

By word 'useful', it means the students/faculties should get the required information really quick on just few clicks.

The website should be made simple by offering best design.

The website should have news about the calendar of events, latest update of college.

Website should have responsive design which fits all desktop.

Website should have well-organised navigation.

2.

Admission enquiry page

Admission enquiry page should have all the information such as Department names, fees for certain categories (CET, Comed-K) etc.

It should have a welcome page for students and it should have a brochure where students can download the details about the college.

3.

‘What’s new’ about the college

The college website should contain information page where in new accomplishments of college can be displayed.

The accomplishments of college students in different aspects and field should be visible for the users.

It should display the upcoming events taking place in the college.

Test Cards

1.

Input: The user enters the website link to open the website

Test case: Tests whether the link is correct or invalid

Output: Opens the college website if successful or display error.

2.

Admission enquiry page

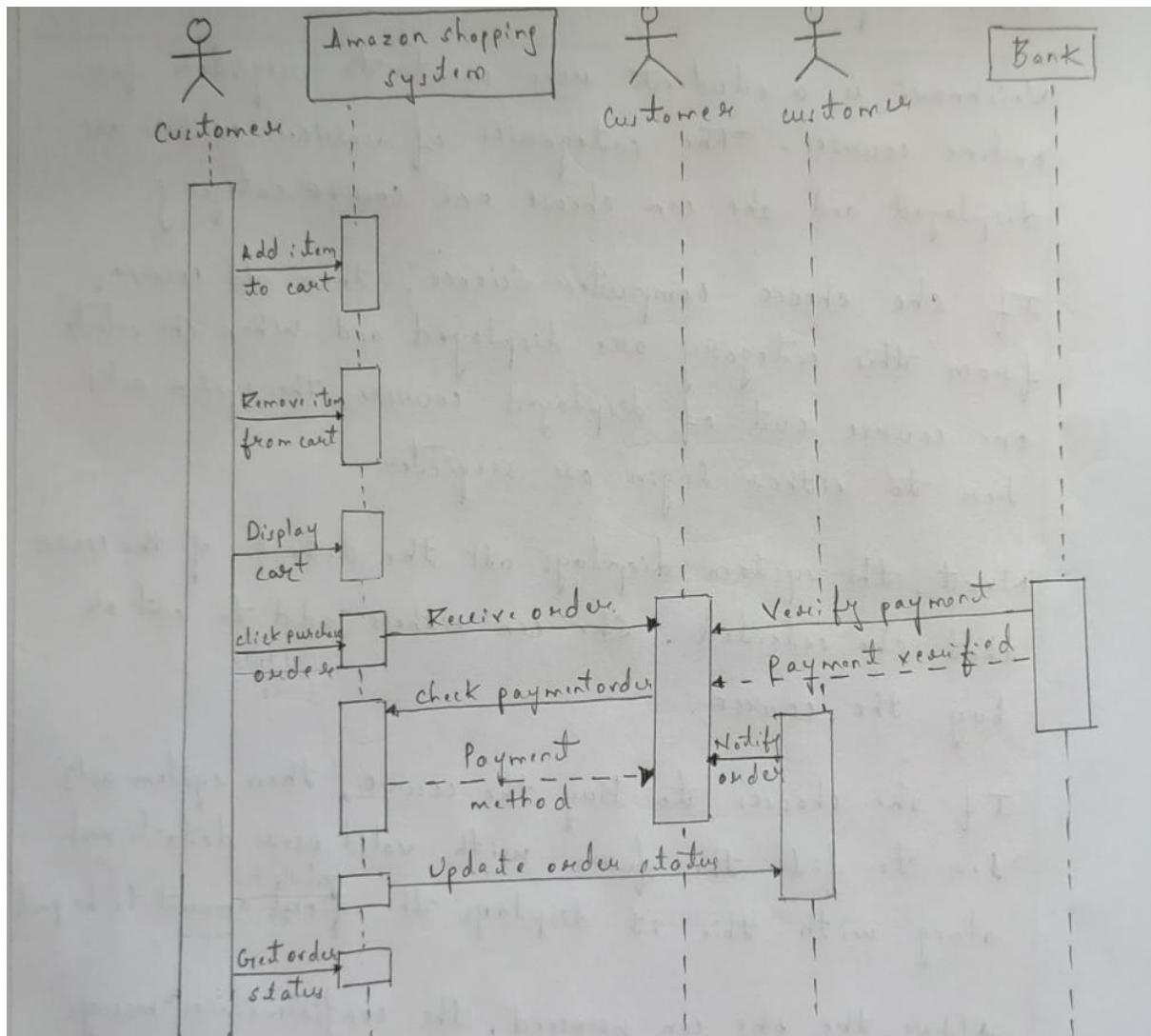
Input: User enters the details in the given form.

Test case: Tests whether the form is completely filled in proper format or not.

Output: The brochure containing college details is mailed to the user if successful or display error.

12. Draw Sequence diagram for make payment use case in Amazon E-commerce web application.

Ans:



13. What are the quality guidelines and quality attributes of software engineering design process?

Ans:

Quality Guidelines



- ✧ A design should exhibit an architecture that (1) has been created using recognizable architectural styles or patterns, (2) is composed of components that exhibit good design characteristics and (3) can be implemented in an evolutionary fashion.
- ✧ A design should be modular.
- ✧ A design should contain distinct representations of data, architecture, interfaces, and components.
- ✧ A design should lead to data structures that are appropriate for the classes to be implemented and are drawn from recognizable data patterns.

Quality Guidelines continued



- ✧ A design should lead to components that exhibit independent functional characteristics.
- ✧ A design should lead to interfaces that reduce the complexity of connections between components and with the external environment.
- ✧ A design should be represented using a notation that effectively communicates its meaning.

Quality Attributes



✧ Functionality

✧ Usability

✧ Reliability

✧ Performance

✧ Supportability

-
- ✧ Functionality is assessed by evaluating the feature set and capabilities of the program, the generality of the functions that are delivered, and the security of the overall system.
 - ✧ Usability is assessed by considering human factors , overall aesthetics, consistency, and documentation.
 - ✧ Reliability is evaluated by measuring the frequency and severity of failure, the accuracy of output results, the mean-time-to-failure (MTTF), the ability to recover from failure, and the predictability of the program.
 - ✧ Performance is measured by considering processing speed, response time, resource consumption, throughput, and efficiency.
 - ✧ Supportability combines the ability to extend the program (extensibility), adaptability, serviceability—these three attributes represent a more common term, maintainability—and in addition, testability, compatibility, configurability .the ease with which a system can be installed, and the ease with which problems can be localized.

14. Explain Context model with an example.

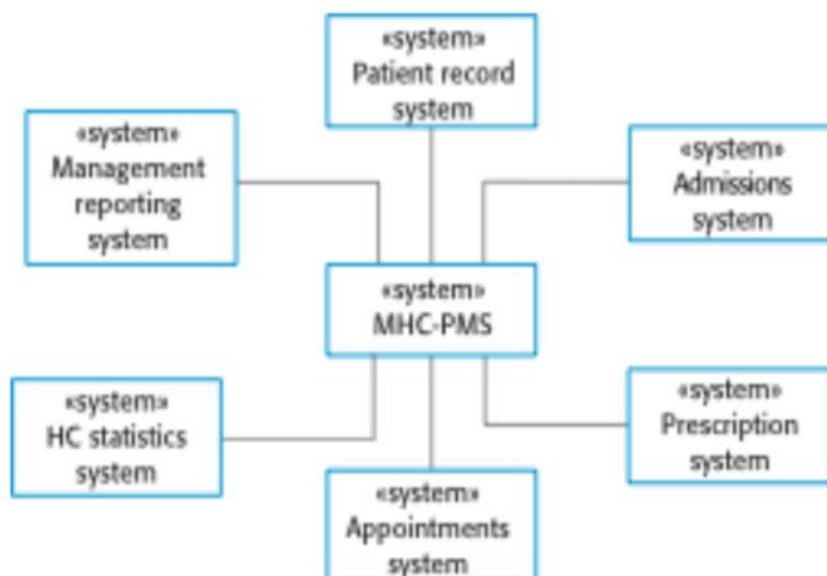
Ans:

Context models



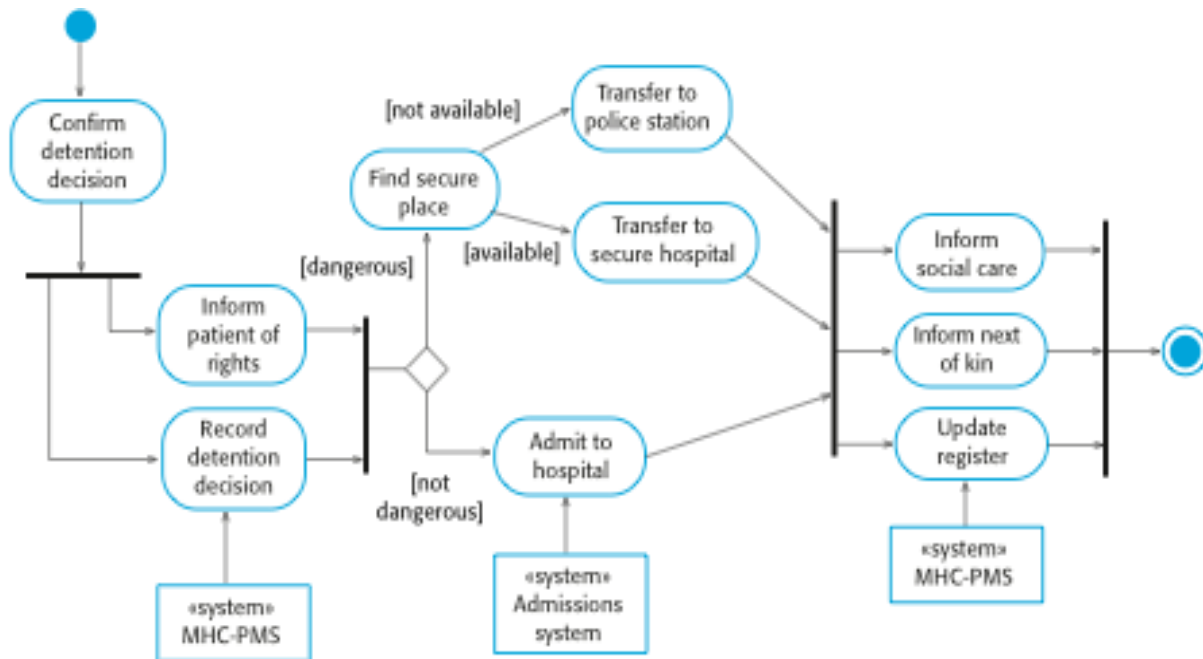
- ✧ Context models are used to illustrate the operational context of a system - they show what lies outside the system boundaries.
- ✧ Social and organisational concerns may affect the decision on where to position system boundaries.
- ✧ Architectural models show the system and its relationship with other systems.
- ✧ Context models simply show the other systems in the environment, not how the system being developed is used in that environment.

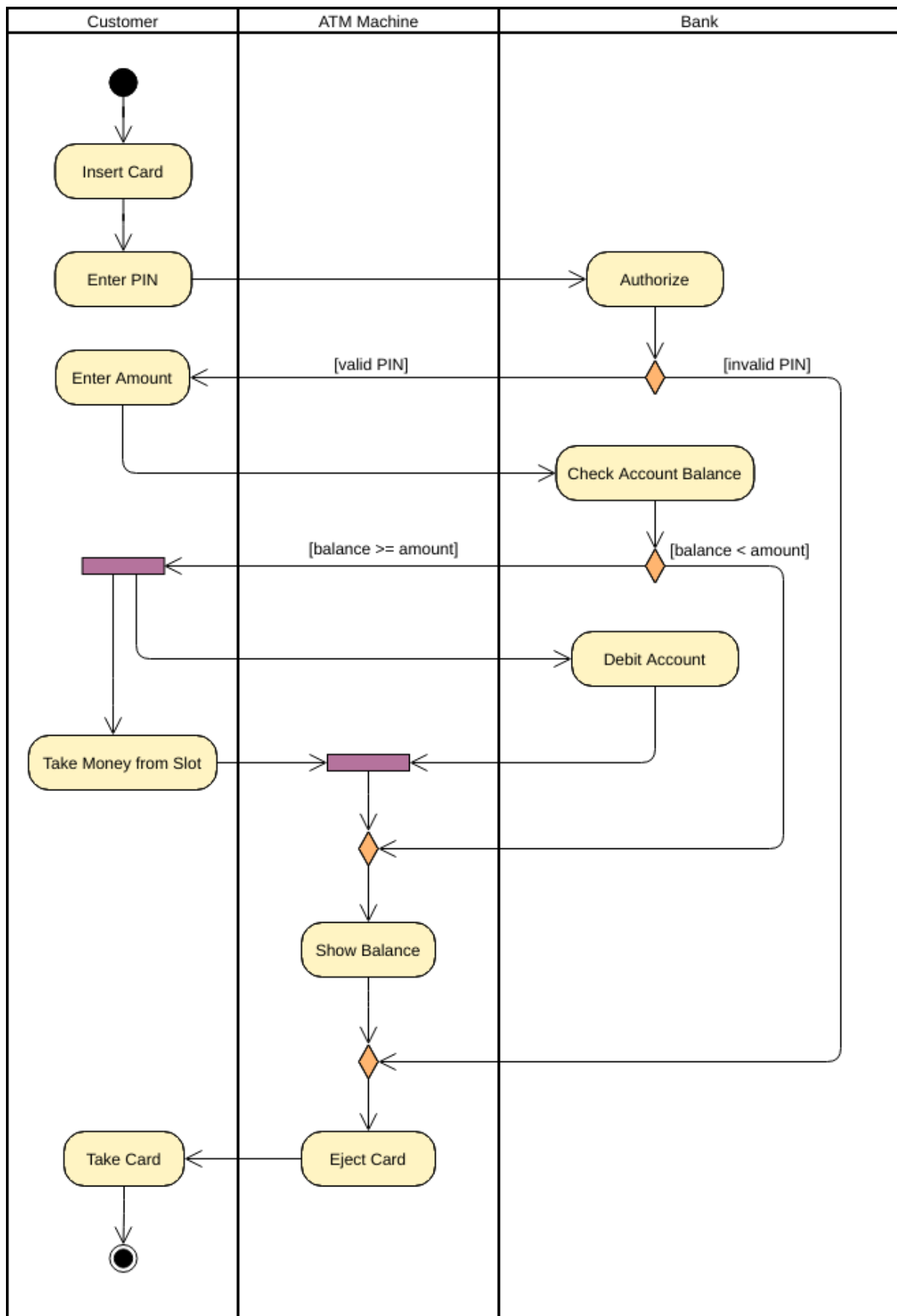
The context of the MHC-PMS

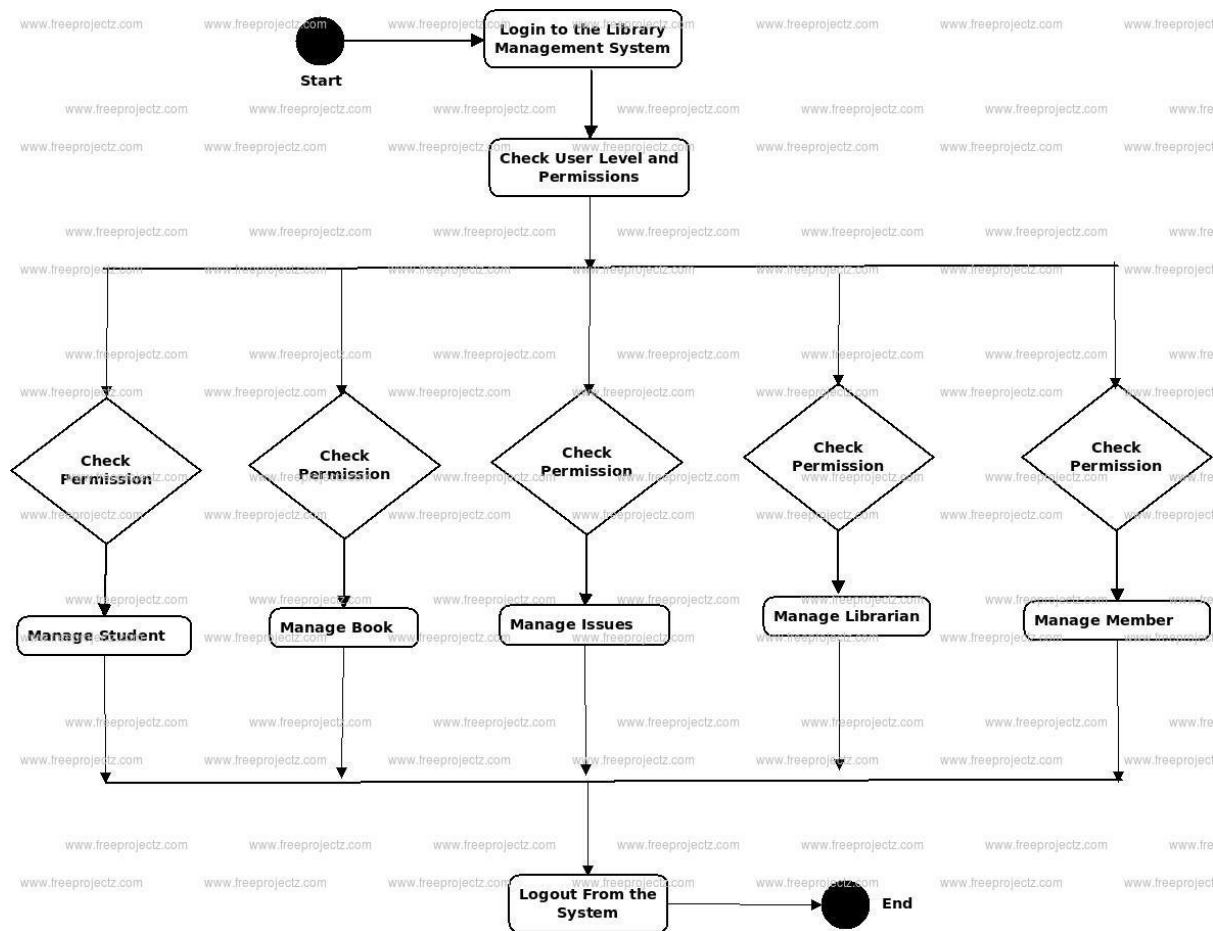


15. Define process model and design a process model (activity diagram) for MHC-PMS involuntary detention, Bank ATM, Library system.

Ans: Process models reveal how the system being developed is used in broader business processes.



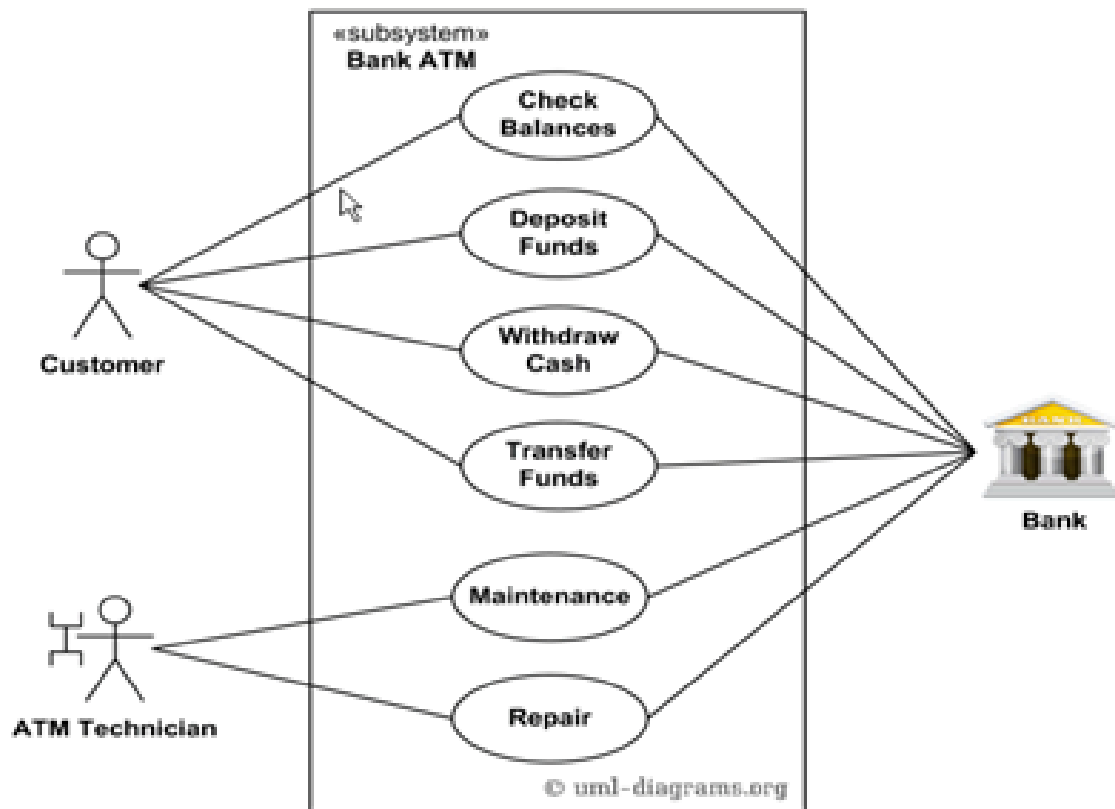




Activity Diagram for Library Management System

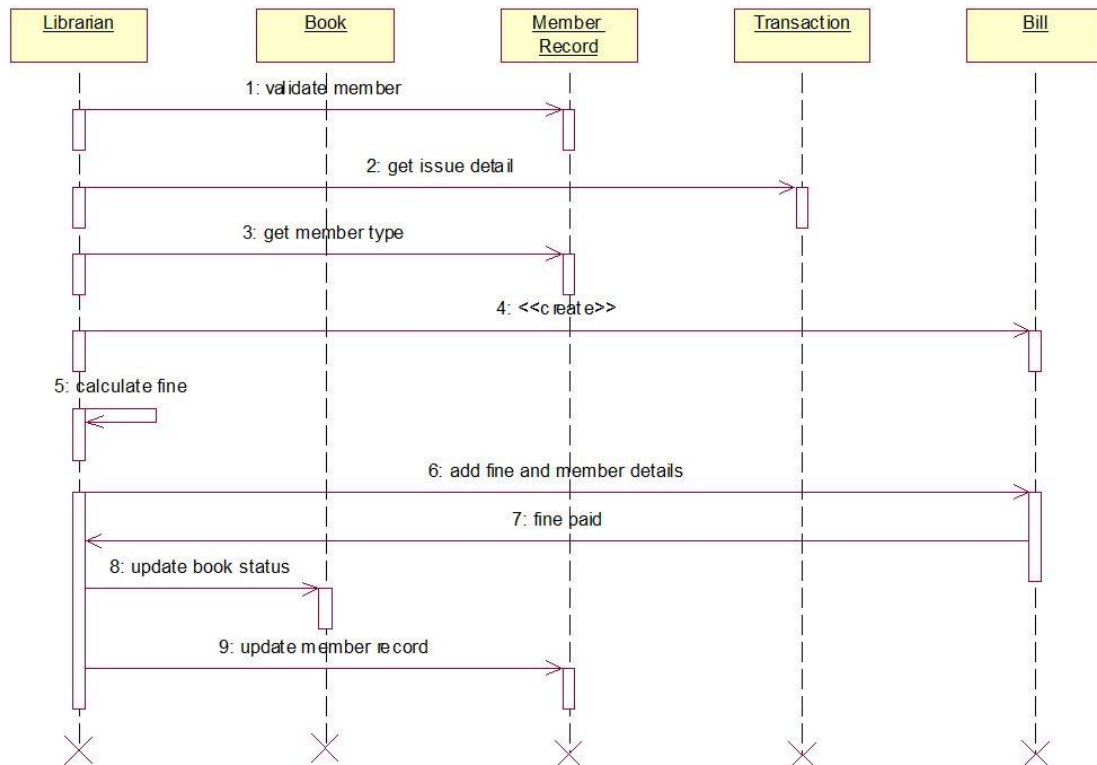
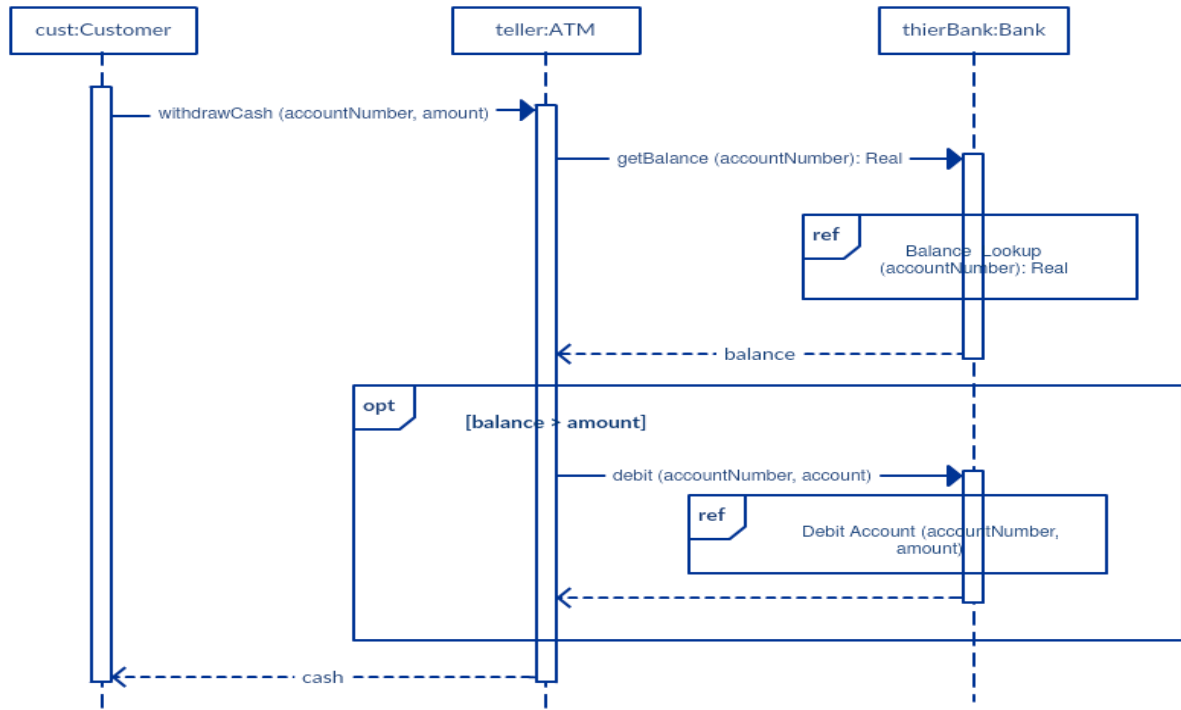
16. Develop a set of Use Cases that would serve as bases for understanding the requirement for an ATM system. Note: Actors: Bank, Customer, ATM

Ans:



17. Design a sequence diagram for ATM withdrawal system and library system.

Ans:



18. What are Agile methods? State Agile manifesto.

Ans: Agile methods are incremental development methods in which the increments are small, typically, new release of the system is created and made available to customers every two or three weeks.

Agile manifesto

- *We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:*
 - *Individuals and interactions over processes and tools*
 - *Working software over comprehensive documentation*
 - *Customer collaboration over contract negotiation*
 - *Responding to change over following a plan*
- *That is, while there is value in the items on the right, we value the items on the left more.*

19. Discuss the advantages of pair programming.

Ans:

Advantages of pair programming

- It supports the idea of collective ownership and responsibility for the system.
 - Individuals are not held responsible for problems with the code. Instead, the team has collective responsibility for resolving these problems.
- It acts as an informal review process because each line of code is looked at by at least two people.
- It helps support refactoring, which is a process of software improvement.
 - Where pair programming and collective ownership are used, others benefit immediately from the refactoring so they are likely to support the process.

