Chapter 2:

2.1 Suggest the most appropriate generic software process model that might be used as a basis for managing the development of the following systems:

- A system to control antilock braking in a car
    - Safety Critical System
    - Plan Driven approach with requirements analysed carefully
    - Waterfall model most appropriate with formal transformations between different development stages.
- A virtual reality system to support software maintenance
    - Cutting edge and UI dependent for usability
    - Incremental or Spiral with some UI prototyping
    - Agile process may be used
- A university accounting system that replaces an existing system
    - Requirements fairly well known
    - Reuse based approach is appropriate
- An interactive travel planning system that helps users plan journeys with the lowest environmental impact
    - Complex UI which is stable and reliable
    - Incremental development approach most appropriate
    - System requirements will change as user experience is gained

2.2 Incremental software development could be very effectively used for customers who did not have a clear idea about the systems needed for their operations. Discuss

- Based on the idea of developing initial implementation, getting user feedback and evolving software through several versions.
- Either plan driven, agile or mixture.
- Customer can evaluate systems at a relatively early stage.

2.3 Consider the integration and configuration process model shown in Figure 2.3. Explain why it is essential to repeat the requirements engineering activity in the process.

- Firstly requirements spec involve proposing initial requirements for the system
    - Brief descriptions
- Refinement stage, reusable components and applications discovered for refinement.
    - Modified to reflect available components and system spec is redefined.

2.4 Suggest why it is important to make a distinction between developing the user requirements and developing system requirements in the requirements engineering process

1) User Requirements describe the system functions and features from the perspective of a user. These are usually abstract. System requirements provide a more detailed explanation of the procedure.

2) User requirements are written in plain and natural language. System requirements are more detailed with specific specifications that could be part of a contract.

3) User requirements start with gathering information, identifying important aspects, and understanding them.
System Requirements can only be implemented after the user requirements are understood and finalised.

2.5 Using an example, explain why the design activities of architectural design, database design, interface design and component design are interdependent

Design of architecture:

- System overall structure is defined.
- Discuss reason for interdependency

Design of Database:

- Define structure of database as representation.

Design of interface:

- Interface defined here.
- Ease of use

Component design and selection

- Reusable components matching requirements are defined here

All design activities above should be followed and integrated. Therefore these work together and can be called interdependent

2.6 Explain why software testing should always be incremental, staged activity. Are programmers the best people to test the programs that they have developed?

Software testing:

- Each program is tested for correct functionality
- Complete program divided into small modules, tested individually. (UNIT TESTING)
- Then whole system is tested with the programs tested above. (SYSTEM TESTING)
- Beta version released and end user checks (CUSTOMER TESTING)

Programmers are not the best persons

- Hard to find own errors.
- Best skills to develop but not to test.

2.7 Imagine that a government wants a software program that helps to keep track of the utilization of the country's vast mineral resources. Although the requirements put forward by the government were not very clear, a software company was tasked with the development of a prototype. The government found the prototype impressive, and asked it be extended to be the actual system that would be used. Discuss the pros and cons of taking this approach

Pros:

- Allows changes to prototype
- Customer sees prototype and modifications made before design is created
- Better implementation
- Satisfied customer

Cons:

- Additional cost for prototype creation
- Delays due to extra time creating prototypes

2.8 You have developed a prototype of a software system and your manager is very impressed by it. She proposes that it should be put into use as a production system, with new features added as required. This avoids the expense of system development and makes the systems immediately useful. Write a short report for your manager explaining why prototype systems should not normally be used as production systems

- Prototype anticipates changes required
- Requirements engineering prototype helps with elicitation and validation
- System design process: used to explore software solutions in UI development

- Minimal UI and not intuitive
- No error detection
- Vague error messages
- Not viewed as high quality product, only development aids.

2.9 Suggest two advantages and two disadvantages of the approach to process assessment and improvement that is embodied in the SEI's Capability Maturity framework

Adv:

- Focused on software engineering processes and practices used
- Led to significant improvements in capabilities

Disadv:

- Too much overhead in formal process improvement in small companies
- Maturity estimation with agile processes is difficult

2.10 Historically, the introduction of technology has caused profound changes in the labour market and, temporarily at least, displaced people from jobs. Discuss whether the introduction of extensive process automation is likely to have the same consequences for software engineers. IF you don't think it will, explain why not. If you think that it will reduce job opportunities, is it ethical for the engineers affected to passively or actively resist the introduction of this technology?

- Reduce human error in code creation
- Potential to produce similar or better software than conventionally produced software
- Cost reduction

- Standardised components used, increasing software reliability and cost reduction in future maintenance
- Automation assists software to address primary issues in the development process. (complexity, reliability, and productivity)

Chapter 3

3.1 At the end of their study program, students in a software engineering course are typically expected to complete a major project. Explain how the agile methodology may be very useful for the students to use in this case

- Team centered, assisting with interaction.
- Phase release, showing updates as they work
- Specification, design, and implementation are interleaved.
- No detailed system spec and design docs is minimized.

3.2 Explain how the principles underlying agile methods lead to the accelerated development and deployment of software.

- Individual and interactions over processes and tools
  - This means that the team can focus on the development of working software
- Working software over comprehensive documentation
  - Rather the programmer's time is focused on the development and testing of code.
- Customer collaboration over contract negotiation
  - This allows useful functionality to be developed and delivered earlier than would be possible if contracts were required.
- Responding to change over following a plan
  - There is significant overhead in changing plans to accommodate change and the inflexibility of a plan means that work may be done that is later discarded.

3.3 Extreme programming expresses user requirements as stories, with each story written on a card. Discuss the advantages and disadvantages of this approach to requirements description

Adv of stories:

- Represent real situations arising so systems support common user operations
- Easy for users to understand and critique stories
- Represent increments of functionality

Disadv:

- Liable to be incomplete
- Focus on functional requirements
- Representing system requirements such as performance and reliability are impossible
- Relationship between architecture and user stories is unclear.

3.4 In test-first development, tests are written before the code. Explain how the test suite may compromise the quality of the software system being developed.

- The test suite could become outdated by the time the actual code is written and not test every vulnerability that the code may have.
- If tests aren't reviewed, and further tests written, undetected bugs may be in system release.

3.5 Suggest four reasons why the productivity rate of programmers working as a pair might be more than half that of two programmers working individually

- Pair programming leads to continuous informal reviewing. Discovers bug quicker than individual testing.
- Informal sharing is implicit. Reducing need for documentation and time required for another's work load.
- Encourages refactoring – reducing costs of subsequent development and changes in the future can be done quicker
- People are likely to spend less time in fine-grain optimization. Both focus on essential features which are produced quicker.

3.6 Compare and contrast the Scrum approach to project management with conventional plan –based approaches. Your comparison should be based on the effectiveness of each approach for planning the allocation of people to projects, estimating the cost of projects, maintaining team cohesion, and managing changes in project team membership

	Scrum Approach:	**Conventional plan-based Approaches:**
Effectiveness applicability:	Best for small-medium sized projects.	Best for Large security and safety required projects.
Planning allocation of people to projects:	The whole team is involved in the project. No project managers. Planning is participated by everyone.	A manager will break down the work parts and assigns the parts to teams of members involved.
Estimating Cost on projects:	Easier to estimate costs as the project develops throughout the increments of software development, as the amount of time needed for development can be predicted easily.	Determined during initial planning phases. Cost is difficult to determine due to the amount of time needed and product feasibility being undeterminable.
Maintaining team cohesion:	Teams are always working together on the same increment and everything is visible to everyone. Frequent meetings held and tracks backlog of work	Teams are working on different parts of a system, therefore smaller teams work together as opposed to one large team working together, not knowing what other teams are doing
Managing changes in project team membership:	Lesser number of members in the project team due to all everybody working together.	Requires larger amounts of members to ensure the teams have sufficient members.

3.7 To reduce costs and the environmental impact of commuting, your company decides to close a number of offices and to provide support for staff to work from home. However, the senior management who introduce the policy are unaware that software is developed using Scrum. Explain how you could use technology to support Scrum in a distributed environment to make this possible. What problems are you likely to encounter using this approach?

Use of technology:

- Video conferencing
- Scrummaster should be located with development team to be aware of everyday problems
- Product owner should visit developers to establish good relationships with them
- Real time communication through informal communication, I.E. instant message and video calls.
- Continuous integration, all team members can be aware of state of product at any time
- Common development environment for all teams

Problems:

- Development requirement daily meetings but not possible in distributed environment

- Communication gap can occur between members
- Changes lead to slow the entire development of project
- Pair programming benefits detection and evaluation of errors.

3.8 Why is it necessary to introduce some methods and documentation from plan-based approaches when scaling agile methods to larger projects that are developed by distributed development teams?

1. Project Planning:
    a. Essential when developing software with larger teams to ensure right people are available when needed
    b. Ensure delivery schedules are aligned
2. Requirements analysis
    a. Important to decide how to distribute work across teams and ensure each team has some understanding of other teams' work
3. Design documentation:
    a. Import so teams develop independently with having access to software under development.
4. Risk management:
    a. Required to ensure all teams understand risks faced and organize their work to minimize these risks.
    b. Useful to cope with different delivery schedules.

3.9 Explain why agile methods may not work well in organizations that have teams with a wide range of skills and abilities and well-established processes.

Having to switch from their well-established process to a new process might set them back on time that they could use to develop.

3.10 One of the problems of having a user closely involved with a software development team is that they "go native." That is, they adopt the outlook of the development team and lose sight of the needs of their user colleagues. Suggest three ways how you might avoid this problem, and discuss the advantages and disadvantages of each approach.

1. Involve multiple users in the development team.
    a. Different perspectives
    b. Better coverage of user tasks
    c. Disadvantages are cost, and getting user engagement and conflicts.
2. Change the user who is involved with the team.
    a. multiple perspectives.
    b. Disadv. Each user takes time to be productive
3. Validate user suggestions with other user representatives
    a. Advantages are independent check on suggestions
    b. Disadv. Is slowing down development process to do checks.

Chapter 4

4.1 Identify and briefly describe the four types of requirements that may be defined for a computer-based system

1. User Requirements: which is the requirements that are statements in natural language plus diagrams of the services the system provides and its operational constraints.
2. System Requirements: A structured document setting out detailed description of the systems functions, services and operational constraints. Define what should be implemented. It may be part of a contract between client and contractor.
3. Functional Requirements: these are the statement of the services the system should provide, how the system should react to particular input and how the system should behave in particular situations.
4. Non-functional requirements: Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards etc often these are applied to the system as a whole rather than individual features or services.

4.2 Discover ambiguities or omissions in the following statement of the requirements for part of a drone system intended for search and recovery:
The drone, a quad chopper, will be very useful in search and recovery operations, especially in remote areas or in extreme weather conditions. It will click high-resolution images. It will fly according to a path preset by a ground operator, but will be able to avoid obstacles on its own, returning to its original path whenever possible. The drone will also be able to identify various objects and match them to the target it is looking for

- How will the drone determine when to take a picture
- Where will the pictures be stored
- What is the memory capabilities of the storage facility
- Will the drone be able to take pictures at night
- Can the ground operator change the path during flight
- What will happen if the drone cannot locate the original path after avoidance of obstacles
- Can more than one target be uploaded
- What will happen when the target is found
- Is a location saved on where the target is found
- How will the drone notify the ground operator that the target is found
- Can the ground operator request the drone to return before the preset path is complete

4.3 Rewrite the above description using the structured approach described in this chapter. Resolve the identified ambiguities in a sensible way

Function	Search and Recovery Drone
Description	Drone used during search and recovery to find the target on a set path
Input	Target image and set path
Source	Ground operator
Outputs	High-resolution images matched to target
Destination	
Action	The drone fly according to the preset path and take high-resolution images until the target is matched to the images taken
Requires	Preset path and target image
Precondition	Target image must match images taken
Post condition	Target is found
Side effects	Target is not found or matched

4.4 Write a set of Non-functional requirements for the drone system, setting out its expected safety and response time

- Drone travel range, when exceeded warning sent to operator
- Response time
- Feedback sent to operator in reasonable time specified.

4.5 Using the technique suggested here, where natural language descriptions are presented in a standard format, write plausible user requirements for the following functions.

1. An unattended petrol pump system that includes a credit card reader. The customer swipes the card through the reader, then specifies the amount of fuel required. The fuel is delivered and the customer's account debited.

Function	Issue gas
Description	Issue gas by deducting the payment from the user's credit card based on the monetary amount or amount of litres advised by the user
Input	Monetary amount or litres with credit card details
Source	Credit card and input details
Outputs	The required gas is issued
Destination	
Action	Gas filling station is at zero state User swipe credit card and advise either the monetary amount or litres System validate the user's credit card and credit availability Validation process should be returned within 5 seconds When the card validation is returned as success, the system indicates to the user to fill the car When the advised monetary amount or litres are reached the system stops with the filling action and advise the user to return the nozzle in the original state The system debits the credit card and issue a receipt. The receipt should be issued within 5-10 seconds The user interface should be easy to navigate and understand The display should output real-time the amount of litres and amount already pumped. The display should be bright during night time hours The nozzle should be able to sense when the fuel tank is overflowing and stop pumping
Requires	The monetary amount or litres and a credit card
Precondition	Credit card is valid and credit is available
Post condition	The requested gas is issued to the user
Side effects	None

The cash-dispensing function in a bank ATM.

Function	Issue cash
Description	Issue cash by validating the user's debit card and Pin and debiting the account by the specific amount advised by the user
Input	Monetary amount with debit card details
Source	Debit card and advised amount

Outputs	The advised amount of cash is issued to the user
Destination	Debit account with amount
Action	Initially the cash dispensing system is at zero state User swipes/inserts card and enter the secret pin associated with the card System validates the pin entered against the card If validation is successful the system request the amount to be withdrawn User advise the amount The system validate the amount against the account balance If the balance is sufficient, the cash is issued to the user. If the balance is insufficient the system display the amount available for withdrawal and allow the user to change the amount requested. The user can cancel the transaction at any time before the cash is issued When cash is issued successfully the bank balance is debited with the amount issued. System interface should be intuitive The system should alert the user audibly when transactions are completed and prompt the user when the card is left in the reader
Requires	An amount and a valid card
Precondition	Valid debit card and pin
Post condition	The cash is issued to the user
Side effects	None

In an internet banking system, a facility that allows customers to transfer funds from one account held with the bank to another account with the same bank

Function	Internal fund Transfer
Description	Transfer funds by deducting the amount from the user's account and input it into the account specified by the user
Input	Account number of receiver
Source	Transfer fund from source account to destination account for a specific amount
Outputs	The required funds are transferred to the destination account
Destination	Amount input and destination account
Action	Initially the fund transfer system is as zero state User select the fund transfer menu option User specify the from account from the list of available accounts User specify the amount to be transferred User specify the account to where the funds should be transferred System validate the destination account details specified System validate the specified amount against the balance available in the from account When validation is successful the funds is debited against the from account and credited to the destination account Notification sent to the user that the transaction was successful Notification sent to the receiver to notify of the transfer User logs out of the system and system returns to zero state
Requires	Sign on onto the system Destination account details Amount

Precondition	User have valid online banking log in details The destination account details are correct Available funds in the from account
Post condition	The required funds are transferred from the source to the destination account
Side effects	None

4.6 Suggest how an engineer responsible for drawing up a system requirements specification might keep track of the relationships between functional and non-functional requirements

- Keeping track is difficult because non functional are sometimes system level requirements rather than requirements specific to a single function(s).

Functional requirement	Related non-functional system requirement	Non-functional requirement
The system shall provide an operation which allows operators to open the release valve to vent steam into the atmosphere	Safety requirement: No release of steam shall be permitted if maintenance work is being carried out on any steam generation plan	Timing requirement: the valve must open completely within 2 seconds of the operator initiating the action

- Or a requirements verification document.

4.7 Using your knowledge of how an ATM is used, develop a set of use cases that could serve as a basis for understanding the requirements for an ATM system

Withdraw cash:

Actors	Customer, ATM, accounting system
Inputs	Customer's card, PIN, Bank account details
Outputs	Customer's card, Receipt, bank account details
Normal operation	Customer inputs his card into the machine He is prompted for a pin which is entered on the keypad If correct, he is presented with a menu of options The withdraw cash option is selected The customer is prompted with a request for the amount of cash required and enters the amount If there is sufficient funds in the account, the cash is dispensed, a receipt is printed and the account balance is updated. The card is returned to the customer who is prompted by the machine to take the card
Exception	Invalid card: Card is retained by machine. Customer is advised to seek advise Incorrect pin: customer is requested to rekey Pin. If incorrect after 3 attempts, card is retained by machine and customer is advised to seek advise Insufficient balance: Available balance is displayed to customer and allow customer to renter a valid amount

Display balance:

Actors	Customer, ATM, Accounting system
Inputs	Customer's card, PIN, Bank account details

Outputs	Customer's card, Receipt, bank account details
Normal operation	The customer authenticates using his card and associated PIN Customer select Display balance option The Current balance of their account is displayed on the screen Option is given to print the balance on a receipt Other menu options displayed to customer or terminate the transaction The card is returned to the customer
Exception	Invalid card: Card is retained by machine. Customer is advised to seek advise Incorrect pin: customer is requested to rekey Pin. If incorrect after 3 attempts, card is retained by machine and customer is advised to seek advise

Print statement:

Actors	Customer, ATM, Accounting system
Inputs	Customer's card, PIN, Bank account details
Outputs	Customer's card, Printed statement, bank account details
Normal operation	The customer authenticates using his card and associated PIN Customer select Print statement option The last five transaction on their account is printed Receipt with transaction is issued to the customer Other menu options displayed to customer or terminate the transaction The card is returned to the customer
Exception	Invalid card: Card is retained by machine. Customer is advised to seek advise Incorrect pin: customer is requested to rekey Pin. If incorrect after 3 attempts, card is retained by machine and customer is advised to seek advise

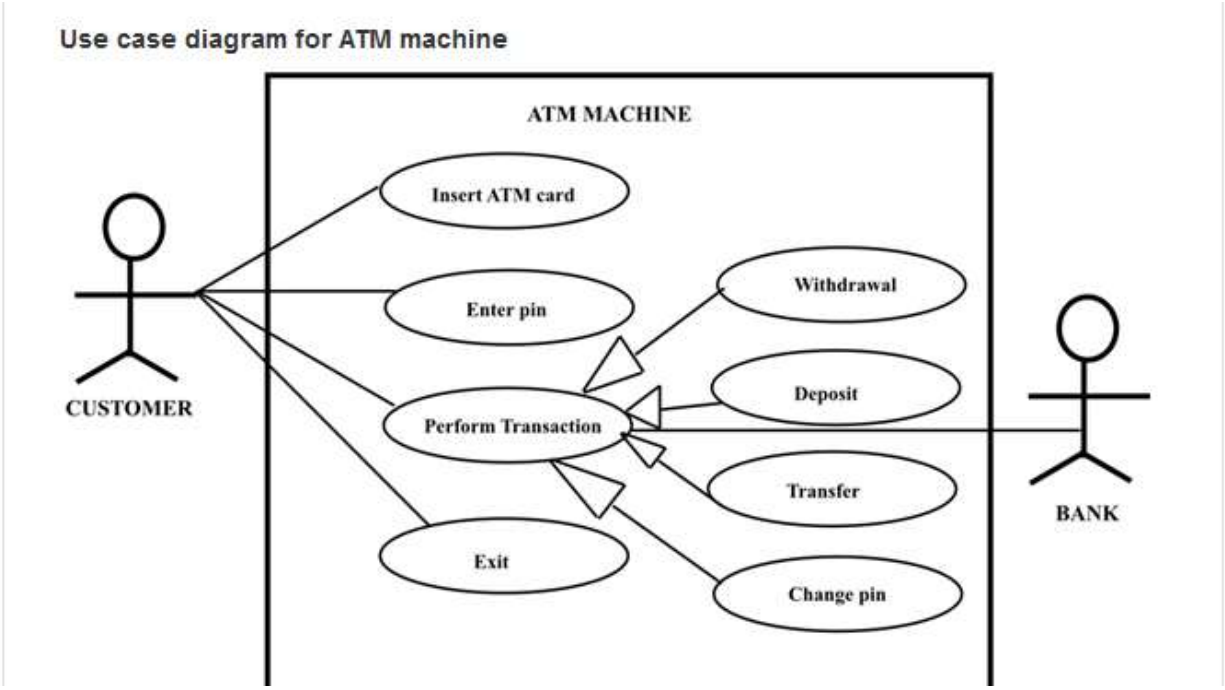
Change PIN

Actors	Customer, ATM, Accounting system
Inputs	Customer's card, PIN, Bank account details
Outputs	Customer's card, Printed statement, bank account details
Normal operation	The customer authenticates using his card and associated PIN Customer select Change PIN option He is prompted twice to input the new PIN System validate that the input are the same The new PIN number is encrypted and stored on the card Other menu options displayed to customer or terminate the transaction The card is returned to the customer
Exception	Invalid card: Card is retained by machine. Customer is advised to seek advise Incorrect pin: customer is requested to rekey Pin. If incorrect after 3 attempts, card is retained by machine and customer is advised to seek advise

	Pins do not match: The customer is prompted to repeat the process to reset the PIN

Deposit cash:

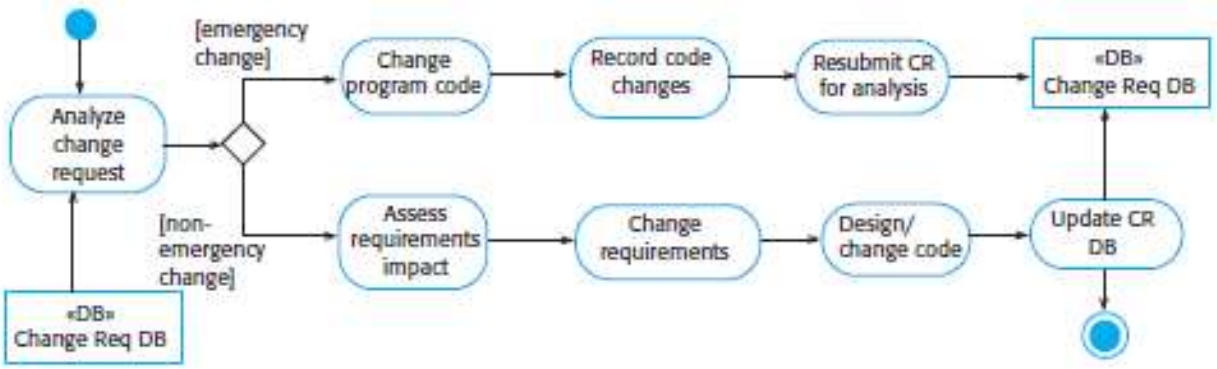
Actors	Customer, ATM, accounting system
Inputs	Customer's card, PIN, Bank account details
Outputs	Customer's card, Receipt, bank account details
Normal operation	Customer inputs his card into the machine He is prompted for a pin which is entered on the keypad If correct, he is presented with a menu of options The Cash Deposit option is selected The customer is prompted for the amount of cash to be deposited and inputs the amount. He is issued with a deposit envelope in which he should put the cash then return it back into the machine The account balance is updated with the amount deposited but marked as uncleared funds. Once the bank has verified the amount the funds are released as cleared funds. A receipt is issued The card is returned to the customer who is prompted by the machine to take the card
Exception	Invalid card: Card is retained by machine. Customer is advised to seek advise Incorrect pin: customer is requested to rekey Pin. If incorrect after 3 attempts, card is retained by machine and customer is advised to seek advise No cash deposited within 1 minute of envelope being issued, the transaction is terminated and card returned to the customer



4.8 To minimise mistakes during a requirements review, an organisation decides to allocate two scribes to document the review session. Explain how this can be done

- Record each defect mentioned and any suggestions for process improvement.
- Often the author plays this role, ensuring log is readable and understandable.

4.9 When emergency changes have to be made to systems, the system software may have to be modified before changes to the requirements have been approved. Suggest a model of a process for making these modifications that will ensure that the requirements document and the system implementation do not become inconsistent



4.10 You have taken a job with a software user who has contacted your previous employer to develop a system for them. You discover that your company's interpretation of the requirements is different from the interpretation taken by your previous employer. Discuss what you should do in such a situation. You know that the costs to your current employer will increase if the ambiguities are not resolved. However, you also have a responsibility of confidentiality to your previous employer

- When differences in requirement interpretation, the requirements validation is carried out or checked if it has been completed.
- It's the process of checking requirements correctly define the system requirements.

Chapter 5

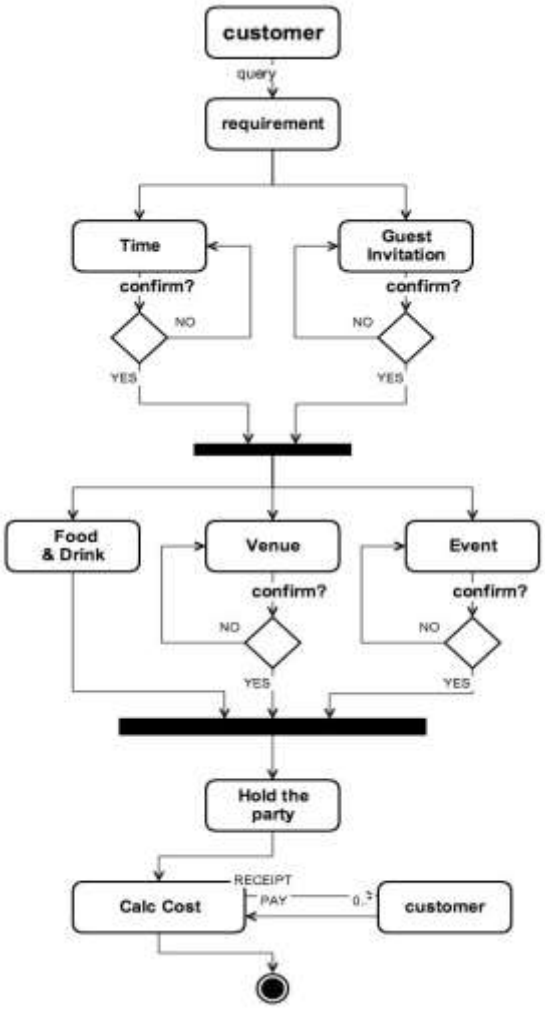
5.1 Scope creep can be defined as a continuous increase in the scope of a project that can significantly increase project cost. Explain how a proper model of the system context can help prevent scope creep

- Early Stage:
  - Define system boundaries
  - Decide what functionality should be included with processing and operations in mind.
  - Check possible overlaps in functionality with existing systems and decide where new functionality should be implemented.
- Later on:
  - Definition of context and dependencies that a system has on its environment.
  - Producing simple architectural model is first step in this activity.
- Context model
  - Shows that the environment includes several other automated systems.
  - UML activity diagrams may be used to show business processes in which systems are used.

5.2 The way in which a system boundary is defined and an appropriate context model is created may have serious implications on the complexity and cost of a project. Give two examples where this may be applicable

- Social and organizational concerns mean the position of a system boundary may be determined by non technical factors
- For example, a system boundary may be deliberately positioned so that the complete analysis process can be carried out on one site
- In developing the specifications you have to decide whether the system should focus exclusively on collecting information about consultations or whether it should also collect personal patient information (Mentcare example)
- The cost and complexity for the system to integrate to various other system will increase if this becomes an integrated Healthcare system reading data from various other systems and also providing data to those systems.

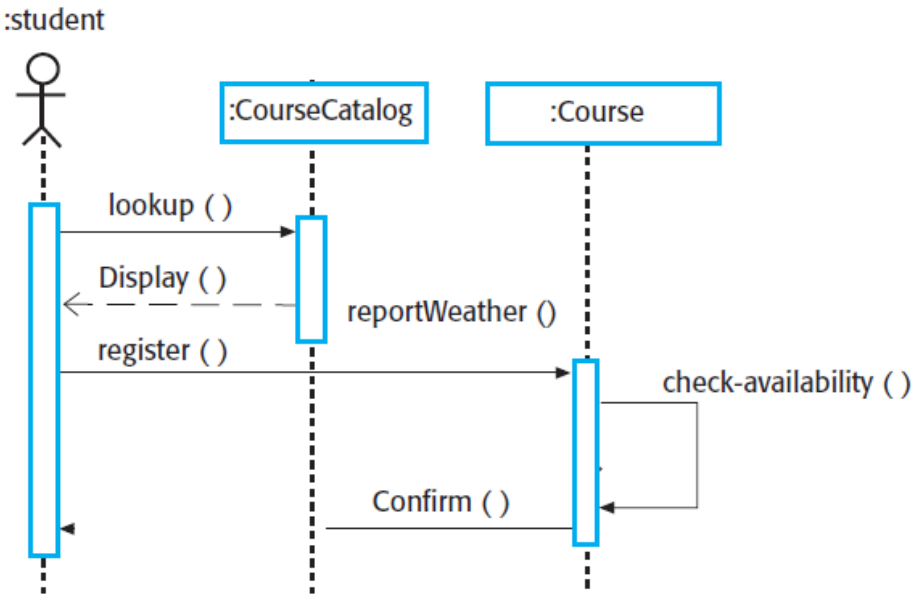
5.3 You have been asked to develop a system that will help with planning large-scale events and parties such as weddings, graduation celebrations and birthday parties. Using an activity diagram, model the process context for such a system that shows the activities involved in planning a party and the system elements that might be used at each stage.



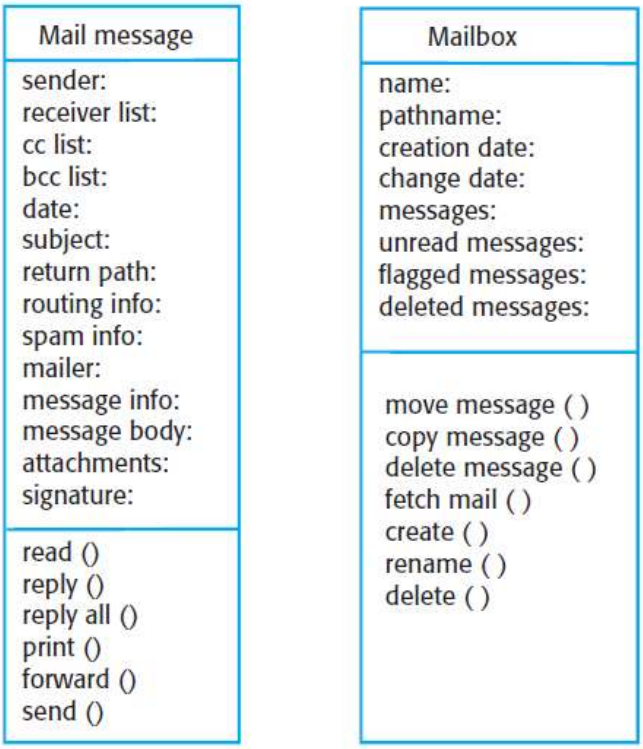
5.4 For the Mentcare system, propose a set of use cases that illustrates the interactions between a doctor, who sees patients and prescribes medicine and treatments, and the Mentcare system.

Actors	Doctor, Patient records system
Description	The doctor enters the patient's Unique Identification data into the Mentcare system The Patient records system transfer basic patient information to the doctor to view e.g. Personal information, diagnoses, previous recordings, treatment information The doctor records his session with the patient in the Mentcare system eg date, time, diagnoses The doctor records the treatment prescribed in the Mentcare system
Data	Patient's personal information, treatment summary
Stimulus	User command issued by the doctor
Response	Confirmation that the patient's record in the Mentcare system has been updated
Comments	The doctor must enter a valid unique Identification number to be able to access the patient's data The doctor must have appropriate security permissions to access the patient information

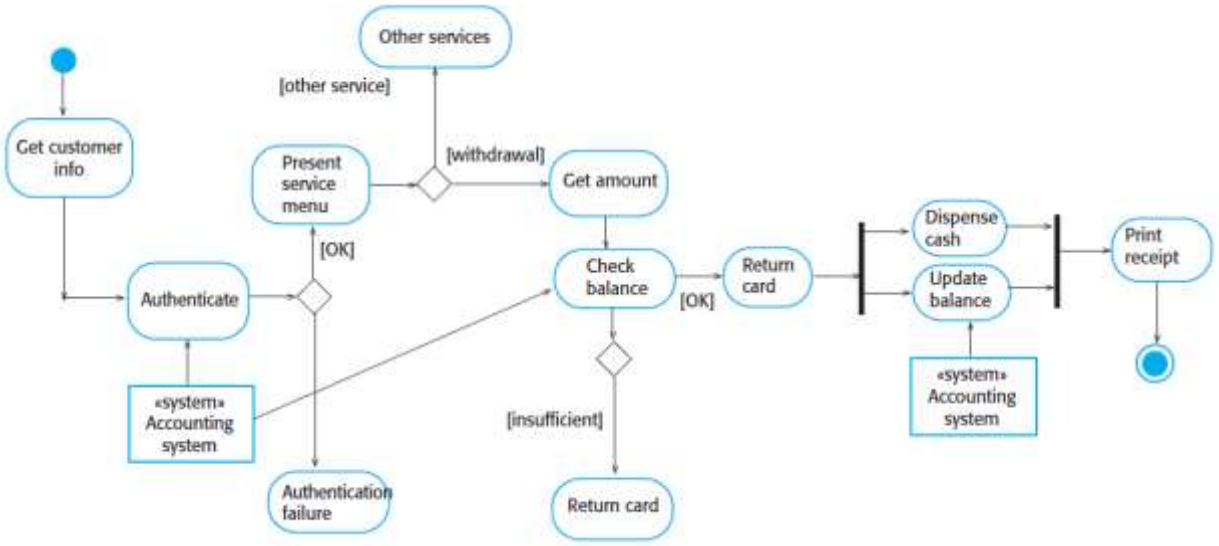
5.5 Develop a sequence diagram showing the interactions involved when a student registers for a course in a university. Courses may have limited enrolment, so the registration process must include checks that places are available. Assume that the student accesses and electronic course catalog to find out about available courses



5.6 Look carefully at how messages and mailboxes are represented in the email system that you use. Model the object classes that might be used in the system implementation to represent a mailbox and an email message

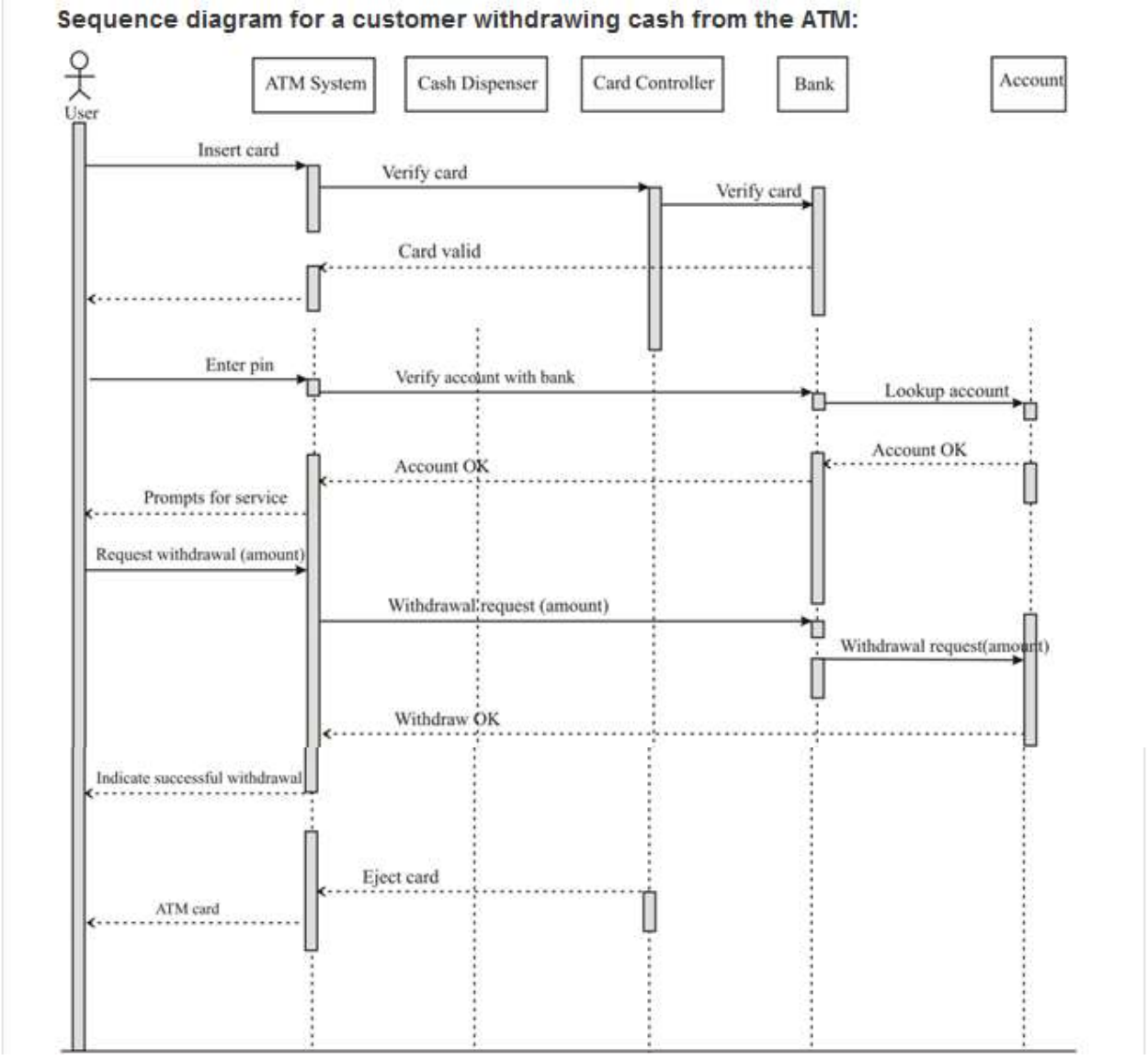


5.7 Based on your experience with a bank ATM, draw an activity diagram that models the data processing involved when a customer withdraws cash from the machine



5.8 Draw a sequence diagram for the same system. Explain why you might want to develop both activity and sequence diagrams when modelling the behaviour of a system.

Sequence diagram for a customer withdrawing cash from the ATM:



The models are used to explain the system in different perspectives

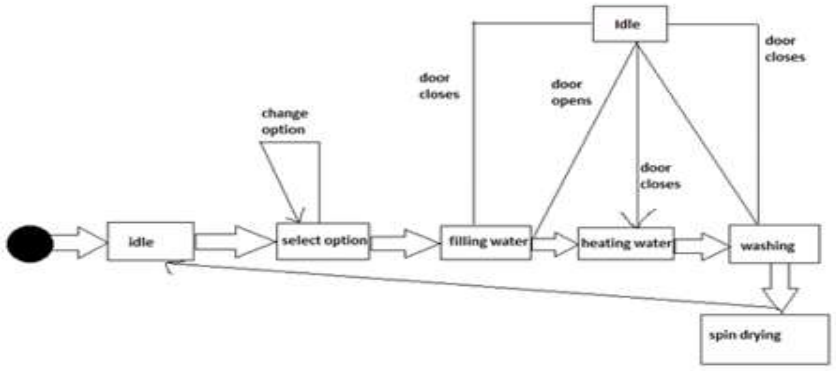
- An external perspective
- An interaction perspective
- A structural perspective
- A Behavioural perspective

- Activity diagram(structural perspective)
    - Activity diagram are used to define business process models
    - Program flows of activities are illustrated by activity diagram
- Sequence diagram (interaction perspective)
    - These diagrams are used to model interaction between the actors and the objects within the system
    - Sequence diagram are used to model the collaboration of objects based on time stamp

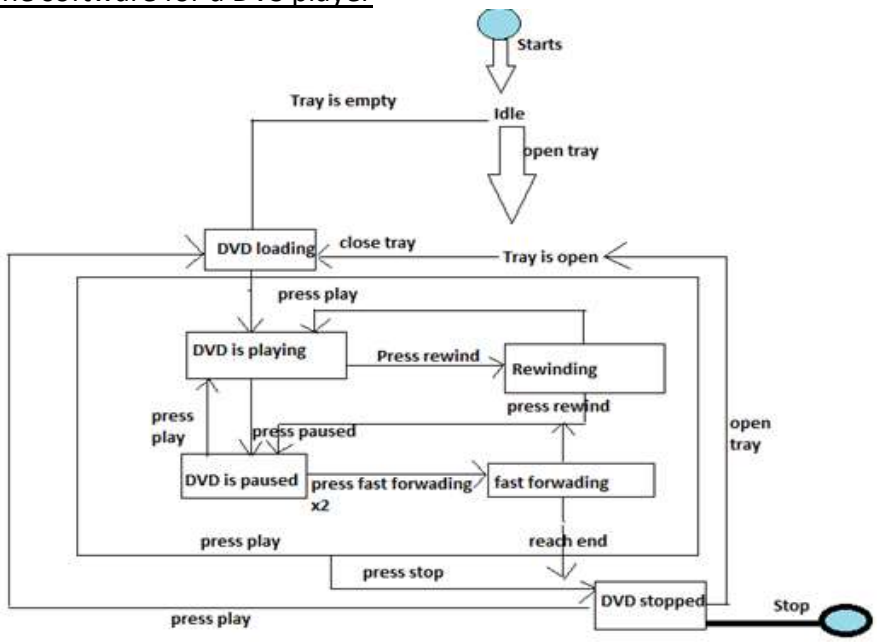
5.9 Draw state diagrams of the control software for:

1. An automatic washing machine that has different programs for different types of clothes

The state diagram for a washing machine is as given below:

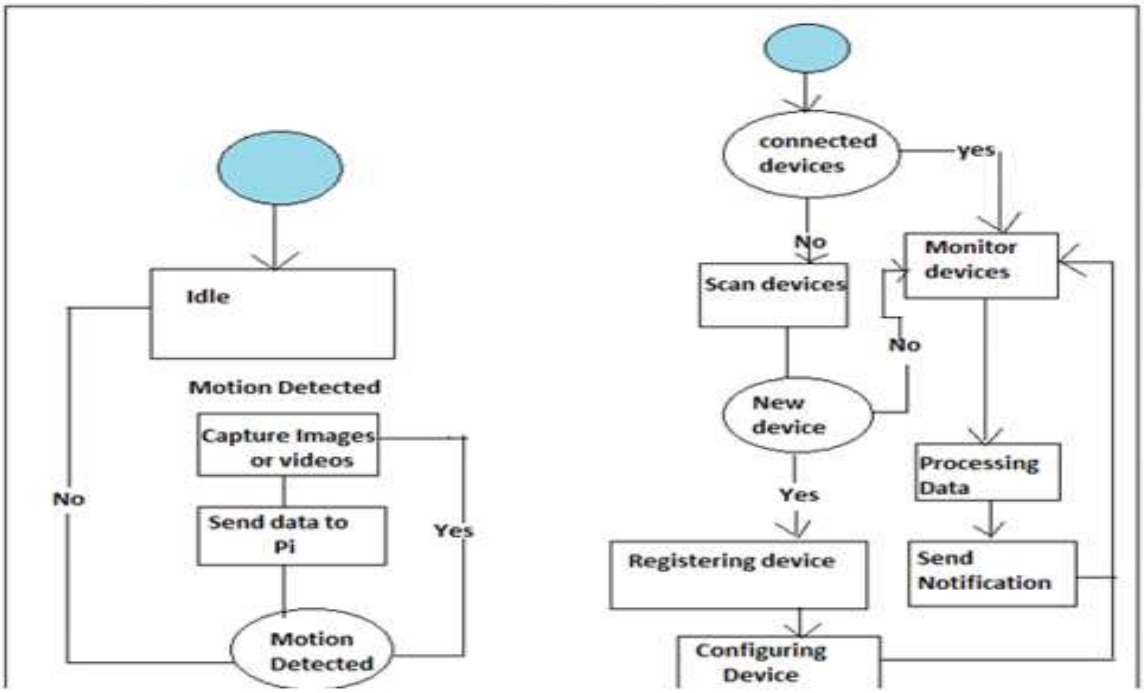


2. The software for a DVS player



3. The control software for the camera on your mobile phone. Ignore the flash if you have one on your phone

The state diagram for camera software of any mobile phone is as given below:



5.10 In principle, it is possible to generate working programs from a high-level model without manual intervention when using model-driven architectures. Discuss some of the current challenges that stand in the way of the existence of completely automated translation tools.

- A particularly difficult problem for automated model transformation is the need to link the concepts used in different CIMS.
- Off-the-shelf tool support is not availabe, so when MDA is introduced into an organisation, special-purpose translators may have to be created to make use of the facilities available in the local environment. Companies do not want to develop or maintain their own tools or to rely on small software companies for tool development.
- It doesn not always follow tha the abstractions that are useful for discussions are the right abstractions for implementation. You may decide to use a completely different implementation approach that is based on the reuse of off-the-shelf applciation systems.
- For most complex systems, implementation is not the major problem – requirements engineering, security and dependability, integration with legacy systems and testing are all more significant. Consequently the gains from the use of MDA are limited.
- The arguments for platform indepence are only valid for large, long-lifetime systems, where the platforms become obsolete during a system's lifetime. For software products and information systems that are developed fo rstandard platforms, such as Windows and Linux, the savings from the use of MDA are likely to be outweighed by the costs of its introduciton and tooling
- The widespread adoption of agile methods over the same period that MDA was evolving has diverted attention away from model-driven approaches.

Chapter 6

6.1 When describing a system, explain why you may have to start the design of the system architecture before the requirements specification is complete

- Architecture has an impact on the nonfunctional requirements and can influence the functional requirements too.
- Early stage of agile development:
    - Focus on designing an overall system architecture.
    - Incremental development is not usually successful.
    - Refactoring components in response to changes is relatively easy
    - Refactoring architecture is expensive because most system components will need to be modified.
- Architecture designed before requirements spec is complete because:
    - Allow manufacture of hardware by subcontractors and provide models for system costing.
    - you need to model the architecture to identify subsystems and associate the requirements with these subsystems.
- Main reasons for the above:
    - Means for structuring
    - Model for estimating system costing
    - Allow manufacture of hardware by subcontractors.

6.2 You have been asked to prepare and deliver a presentation to a nontechnical manager to justify the hiring of a system architect for a new project. Write a list of bullet points setting out the key points in your presentation in which you explain the importance of software architecture

Software architecture affects:

- Performance
- Robustness
- Distributability
- Maintainability

Advantages:

- Stakeholder communication
- System analysis
- Large-scale reuse

6.3 Performance and security may pose to be conflicting non-functional requirements when architecting software systems. Make an argument in support of this statement

Performance means few large components are used.

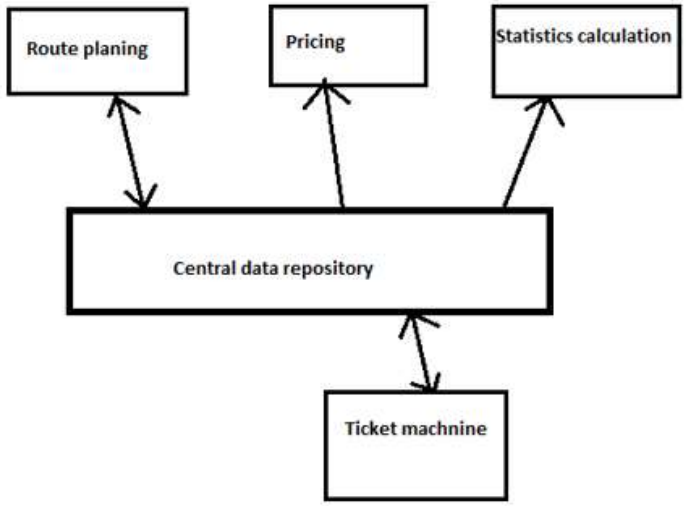
- Reducing component communication

Security means using a layered architecture, which required more than one component to implement security validation.

6.4 Draw diagrams showing a conceptual view and a process view of the architectures of the following systems:

1. A ticket machine used by passengers at a railway station

The system architecture for a ticket machine at a railway station is as given below:

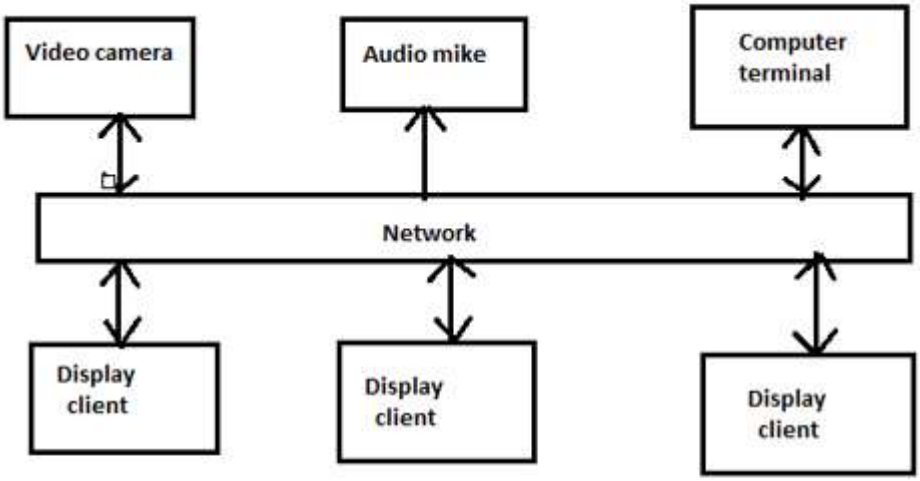


Explanation:

• When a user books a ticket, he/she visits the ticket machine.

2. A computer-controlled video conferencing system that allows video, audio and computer data to be visible to several participants at the same time

The system architecture for a video conferencing system is as given below:



Explanation:

• Initially, for starting the video conferencing user's needs the video camera, Audio mike and computer terminal on both sides.

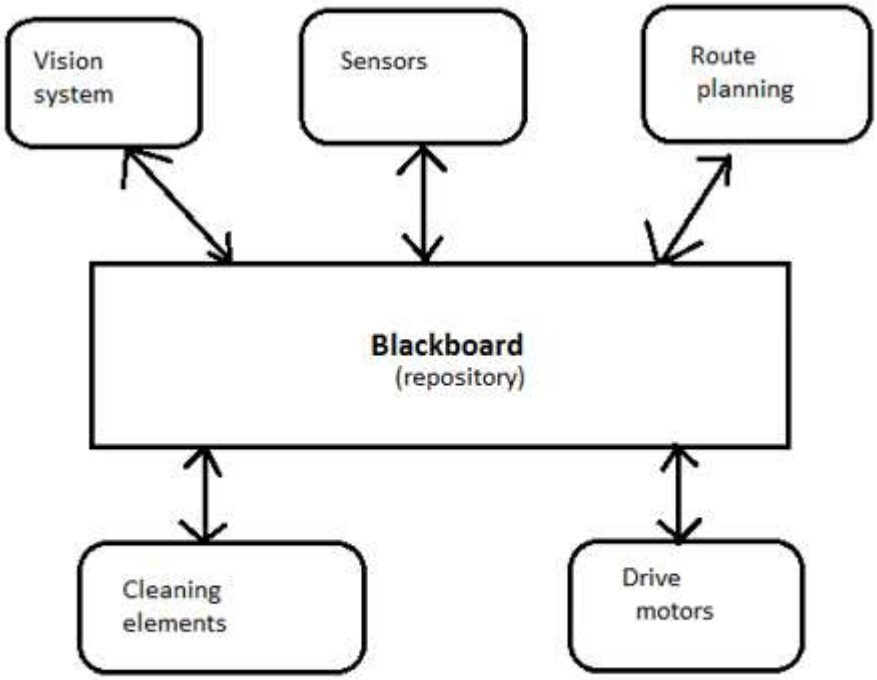
• Video camera and audio mike is required to both ends of the users to see each other and to listen their voice.

• A computer terminal is required to connect both video camera and audio mike.

• A display is required to the both end of the users to see each other.

3. A robot floor-cleaner that is intended to clean relatively clear spaces such as corridors. The cleaner must be able to sense walls and other obstructions.

The system architecture for a robot floor cleaner system is as given below:



Explanation:

- Initially, user requires a robot for cleaning the floor. User controls the robot through a blackboard (repository).
- When a user gives the start command to the robot, Robot is going to the vision system (where the robot sees the path where it works).
- After that the robot on the sensor to find out the route planning where the robot works.
- Robot collect the cleaning element for clean the floor.
- Robot stars floor cleaning and this all process of cleaning is monitored by the user on blackboard (repository).

6.5 A software sysem will be built to allow drones to autonomously herd cattle in farms. These drones can be remotely controlled by human operators. Explain how multiple architectural patterns can fit together to help build this kind of system

- Manages fundamental behaviors and data of the application
- Responds to requests and instructions
- Notifies observers in event driven systems when information changes.
- Could be a DB or any number of data structures or storage systems

- The view provides the UI element, rendering data from the model into a form suitable for UI
- Controller receives input and makes calls to model objects and view to perform various actions.
- These work together to create the 3 components of MVC

6.6 Suggest an architecture for a system (such as iTunes) that is used to sell and distribute music on the internet. What architectural patterns are the basis for your proposed architecture?

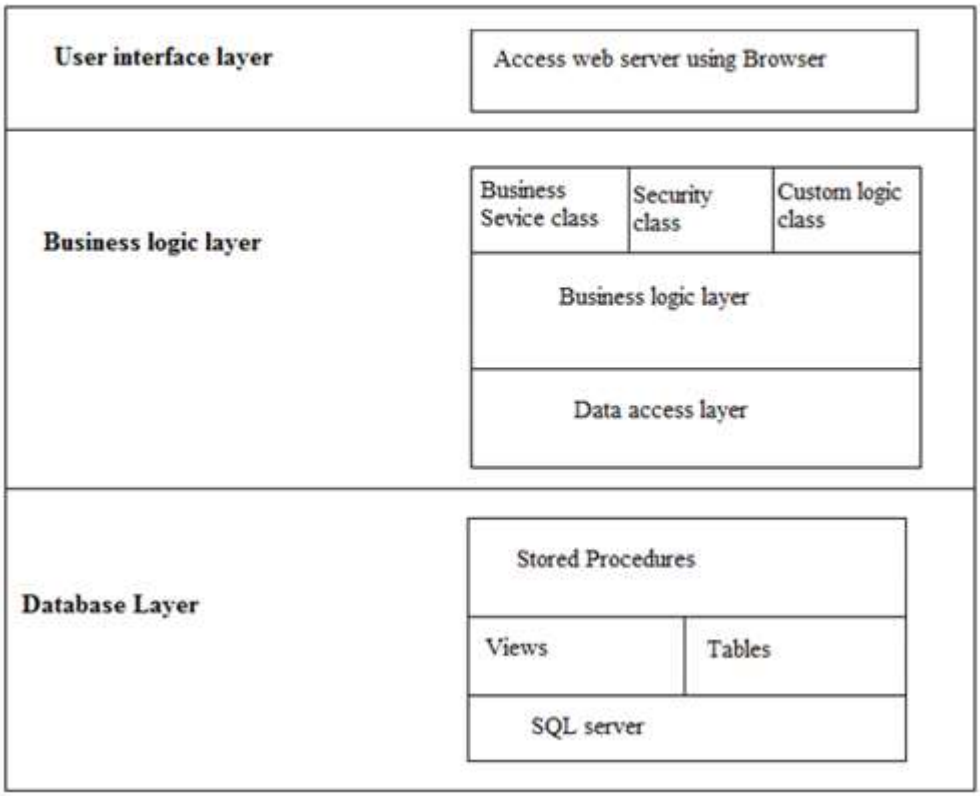
Client-server model.

- iTunes store music in a database where clients can search these tracks by music details via a web based interface.
- Music can be purchased and downloaded via a web based interface.

6.7 An information system is to be developed to maintain information about assets owned by a utility company such as buildings, vehicles, and equipment. It is intended that this will be updatable by staff working in the field using mobile device as new asset information becomes available. The company has several existing asset databases that should be integrated through this system. Design a layered architecture for this asset management system based on the generic information system architecture shown in figure 6.18

Asset management system

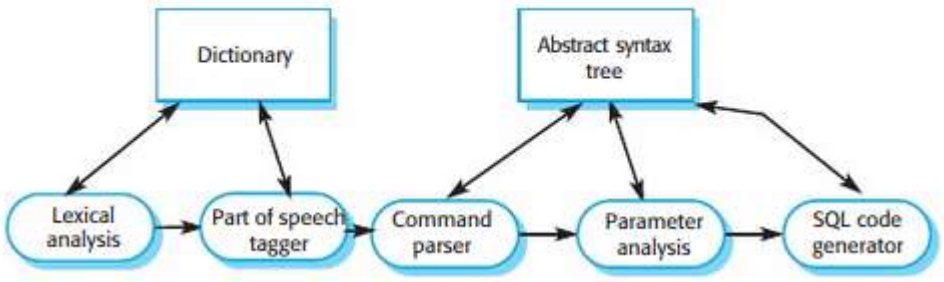
An asset management system has many layers which are related to security, set of rules, etc.
The diagram to show the layered architecture of asset management system is as given below:



There are three types of layers for architecture of an asset management system:

- User interface layer: user can access web services using the browser
- Business logic layer: user has some options. User goes to the business logic layer and access the data on the data access layer
- Database layer: user sees the stored procedures, different views, table and SQL server

6.8 Using the generic model of a language processing system presented here design the architecture of a system that accepts natural language commands and translates these into database queries in a language such as SQL.



6.9 Using the basic model of an information system, as presented in Figure 6.18, suggest the components that might be part of an information system that allows users to view box office events, available tickets and prices, and so eventually buy tickets.

Web browser

Login	Credential Authentication	Data validation	Event schedule manager

Event info manager	Security management	Data exchange

Transaction management Event database

First layer: UI

Second layer includes login components and obtain event schedules

Third layer implement security, maintain information

Fourth layer contains DB and transaction management

6.10 Should there be a separate profession of 'software architect' whose role is to work independently with a customer to design the software system architecture? A separate software company would then implement the system. What might be the difficulties of establishing such a profession?

The difficulties that may arise by the establishment of such a profession are:

1. Difficulty in the implementation
2. The company that would implement the system may find difficulty in understanding the architecture
3. Incompatibility of the architecture with the developing system
4. Architecture may not satisfy all the technical requirements
5. Re-work or redesign is needed sometimes, due to incompatibility or change in technical requirements