Mahmudul Hasan
IT-21019

Answer to the question number: 1

(i) Product backlog for these user stories given bellow:

User Story 1: "As a user, I want to log in securely so that I can access my account."

Design & Setup

- Design the log in UI (mockups & wireframes).
- Set up authentication routes in the backend.

Development

- Implement front-end log in from (HTML, CSS, JS/React)
- Validate user input (email, password etc)
- Implement API endpoint for authentication.
- Integrate password hashing & storage
- Implement session management & token-based authentication.
- Set up Multi-factor Auth. (MFA)

Testing & Security:

·Deployment & Documentation.

User story 2: "As a user, I want to search for products by category to find items easi

Design & Setup:

- Design the search UI
- Define database schema for categories and products.

Development:

- Implement the front-end search bare & filters options.
- Develop API endpoint for product se
- Optimize database queries for efficient category-based search
- Implement autocomplete suggestions.

Testing:

- Test search functionality with

different categories.

- Measure response time for large dataset.

Development & Documentation:

- Deploy search feature to the testing environment
- Write API documentation for the search functionality.

(ii) Priority Assignment:

User Story 1 (Log in Securely) → High Priority
→ Essential for user access, security, and account management.
→ Needs to be implemented first before personalized features

User Story 2 (Search by category) → Medium Priority
→ Important for usability but can follow after login implementation

Sprint Distribution:

→ Sprint 1: Implement & Test the login feature

→ Sprint 2: Develop and test the search functionality.

(iii) The development team uses a Scrum Board to track the progress of task

To Do	In Progress	Done
Design login UI	Develop frontend login form	Validate user
Set up authentication routes	Implement API endpoint for authentication	Implement password hashing
Define database schema for search	Optimize database queries	Test search categories

As tasks progress, they move from To Do → In Progress → Done

Answere to the question no. 2

Spiral Model:

The Spiral Model explicitly focuses on risk analysis and mitigation at each phase. It divides development into iterative cycles, where each cycle includes risk assessment, prototyping, and validation before moving forward. This ensure that high-risk components are addressed early in development.

Adaptability:

Since it incorporates feedback and prototyping in every iteration, the Spiral model allows for changes based on evolving requirements

Agile Methodology

Agile manages risk by breaking develop into short sprints, each delivering a work product increment.

Frequent feedback from the client reduc the risk of building something that do meet their needs. Continous integration testing reduce technical risks.

Adaptability

Agile thrives in environments with uncer and evolving requirements. Since each sprint invokes reassessing priorities, i allows for rapid changes in features or scope.

Extreme Programming

(i) Focus on rapid development with continuous testing and refactoring to manage risks.

(ii) Adaptable to frequent changes but requires constant client involvement.

@ Works best for small teams with rapidly changing needs but may struggle with large, high-risk projects

S. best methodology for this project since the project involves both high risks and evolving requirement, a Hybrid Approach combining. Spiral and Agile is idea.

Answer to the question no: 3

Comparison within different models

Methodology	Flexibility	Customer Collaboration	Risk management	Best for
Water fall	High	low	low	well-defined, strict deadlines
Agile	medium	High	Medium High	Continuous Customer involvement
RAD	low	Very high	medium	Small, fast moving Project and frequent change
Spiral	Medium	Medium	High	High risk project, needed both risk management and adaptability

Best methodology for each project:

For Project A (well-defined & strict deadline): Best fit model is "Waterfall" Since the project has fixed requirements,

and a strict timeline, Waterfall ensures structured planning, clear milestones, and predictable delivery. But if you considered risk management then "Spiral" would be the best alternative.

Project B (Evolving requirements, uncertain timeline)
Best fit model is "Agile" Since agile allows fore frequent iterations, continous feedback, and adaptability to changing customers need. But if need rapid development with increase customer collaboration then "XP" would be the best alternative.

Ans to the question no 9

Software Engineering Ethics & Professional Responsibility.

Key Ethical Principle :-

(i) Public safety and Interest.
(ii) Integrity & Honesty.
(iii) Privacy and Security.
(iv) Fairness & Non-Discrimination
(v) Professional Competence
(vi) Accountability.

ACM/IEEE code of Ethics in Decission-making:

(i) Prioritize Public good
(ii) Honesty and Fairness
(iii) Privacy & Security
(iv) Professional integrity
(v) Continuous Learning

The ACM/IEEE code of Ethics

helps software engineers make ethical choices by emphasizing public good, honesty, privacy, security and continuous learning. It promotes fairness, transparency and accountability in software development.

Ans. to the question no 5:

Functional and Non-functional Requirements for an Airport Reservation System:

Functional Requirements:

(i) Users Registration and Autentication:
(ii) Flight Search and Booking
(iii) Payment Processing.
(iv) Booking Confirmation and notification.
(v) Ticket cancellation and Refunds.

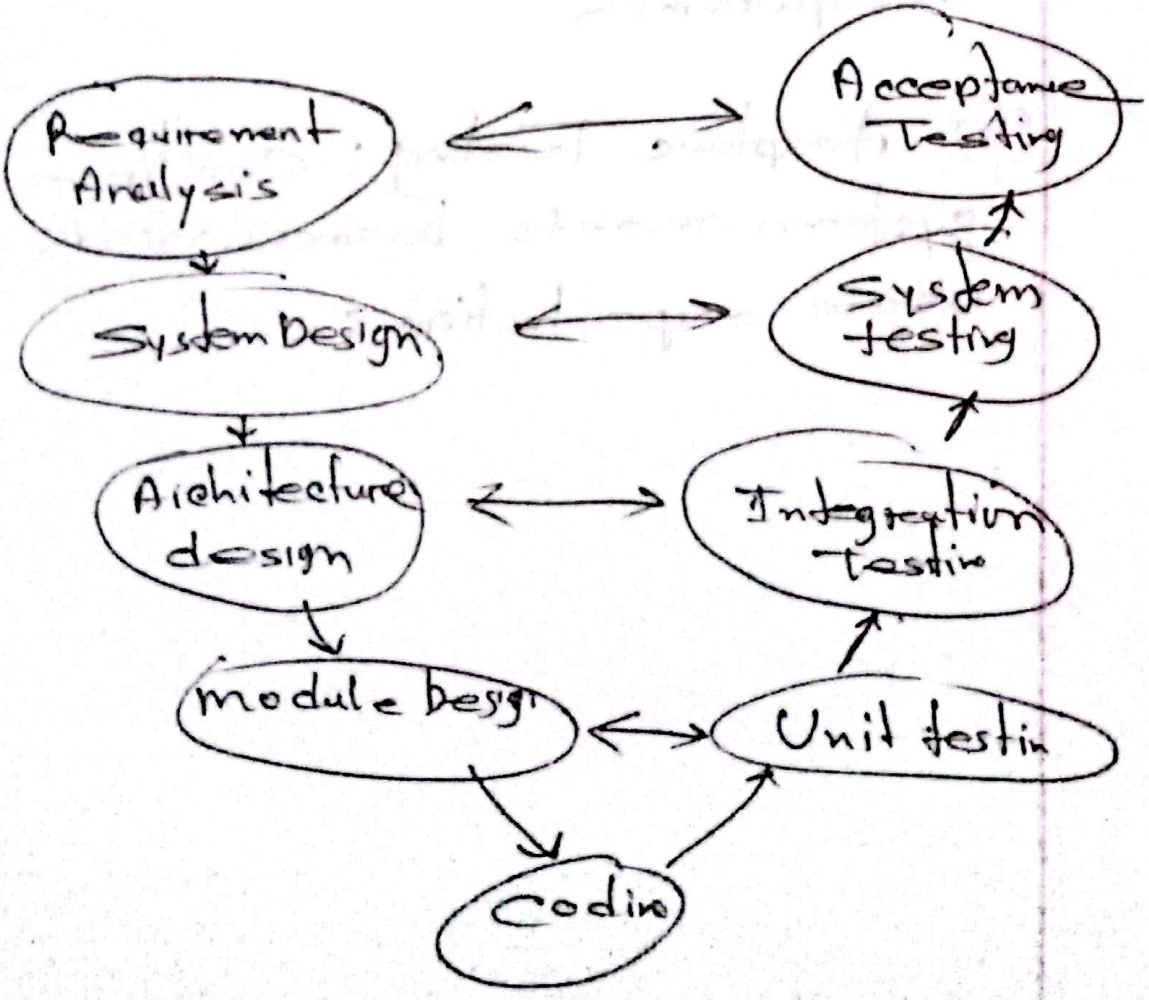
Non-functional Requirements:

(i) Performance and Scalability
(ii) Security and Data Privacy
(iii) Availability and Reliability
(iv) Usability and Accessibility
(v) Maintainability and Upgradibility

By addressing both functional and non-functional requirements the Airport Reservation System can ensure a secure, scalable, and user friendly experience for passengers while maintaing high reliability and efficiency.

Answer to the question no- 6

The V-model (Verification and Validation Model) is sequential software development model where each development phase has a corresponding testing phase. The model follows a "V" shape representing the relationship between development and testing/validation.



Testing Phase:

(i) Unit Testing: Verifies each module independently for correctness

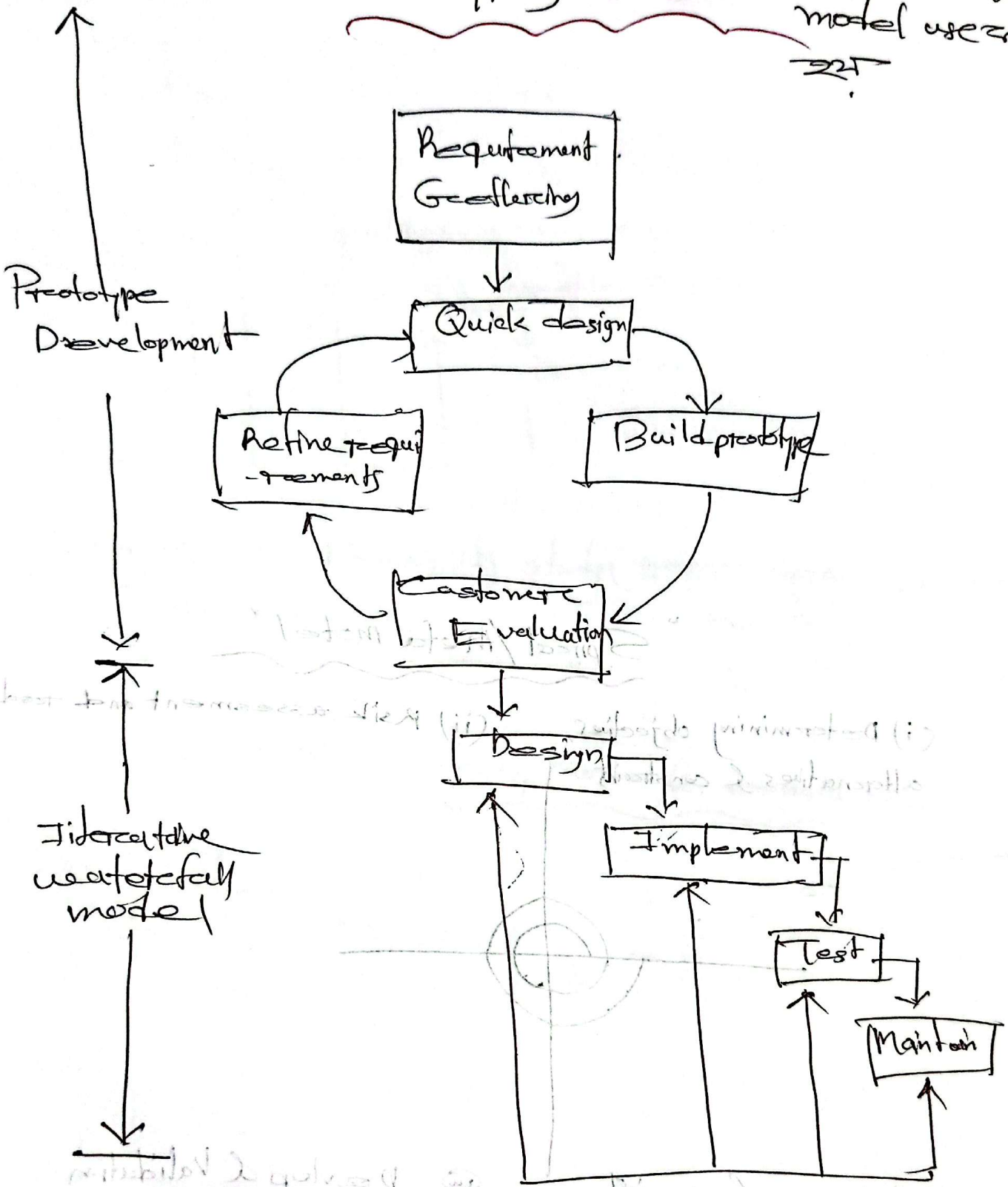
(ii) Integration Testing: Ensures modules work together as expected

(iii) System Testing: Validates the complete system against design requirements.

(iv) Acceptance Testing: Confirm the system meets business needs and users expectations.

(7)

"Prototyping Model" iterative waterfall model use करते हैं।



When use?

→ Customer not clear with idea
→ Rapid prototype → new requirement
→ Throw-away प्रोटो → time-consuming
→ Demo working model → Poor documentation

Mahmudul Hasan
IT-21019

Answers to the question no 8

The process improvement cycle focuses on continuously enhancing software development processes to improve quality, efficiency and performances. It typically follows a structured approach like PDCA (Plan-Do-Check-Act)

Key stages in Process Improvement:

1. Assessment: Evaluate the current process to identify inefficiencies or bottlenecks.
2. Analysis: Investigate the root causes of problems and determine where improvements are needed.

3. Implementation: Apply changes or improvements to the process.

4. Evaluation: Measure the result and impact of the changes.

5. Standardization: If successful, standardize the improvements and make them part of the regular process.

Commonly used Process Metrics:

(1) Defect Density: Measures the number of defects per unit of software.

(2) Cycle time: The time taken to complet a task.

Mahmudul Hasan
IT-21019

3. Velocity: In Agile development, velocity measures the amount of work.

Ans. to the question no- 9

The capability Maturity Model (cmm) provides a framework for assessing and improving software development process.

(i) ~~Level 1:~~ Initial: The focus is on building the basic awareness of the need for improved processes. Organization at this stage face a high level of risk and inefficiency due to lack of structure.

(ii) Managed: Basic project management process are established. The organizati

Starts to repeat successful practices, focusing on ensuring projects are completed on time and within budget.

(iii) Defined: Process are well defined and standardized across the organization. There is a focus on continuous process improvement, with practices for quality assurance.

(iv) Quantitatively Managed: The organization begins using quantitative data and metrics to monitor and control process. Performance is measured and statistical techniques are applied to predict future outcomes and improve process performance.

Mahmudul Hasan
IT-21019

(v) Optimizing: Continous improvement by optimizing processe based on data and feedback.

Answere to the question no 10

Core principle of Agile software Development.

1. Customer Collaboration Over contract Negotiation.
2. Responding to change Over Following a Plan
3. Individuals and interaction over Processes and Tools
4. Working software Over Comprehensive Documentation.
5. Continous Delivery of valuable Software.

1. Application in different environment:

(i) Small projects
(ii) Large projects
(iii) Startups
(iv) Regulated environment.

Benifits:

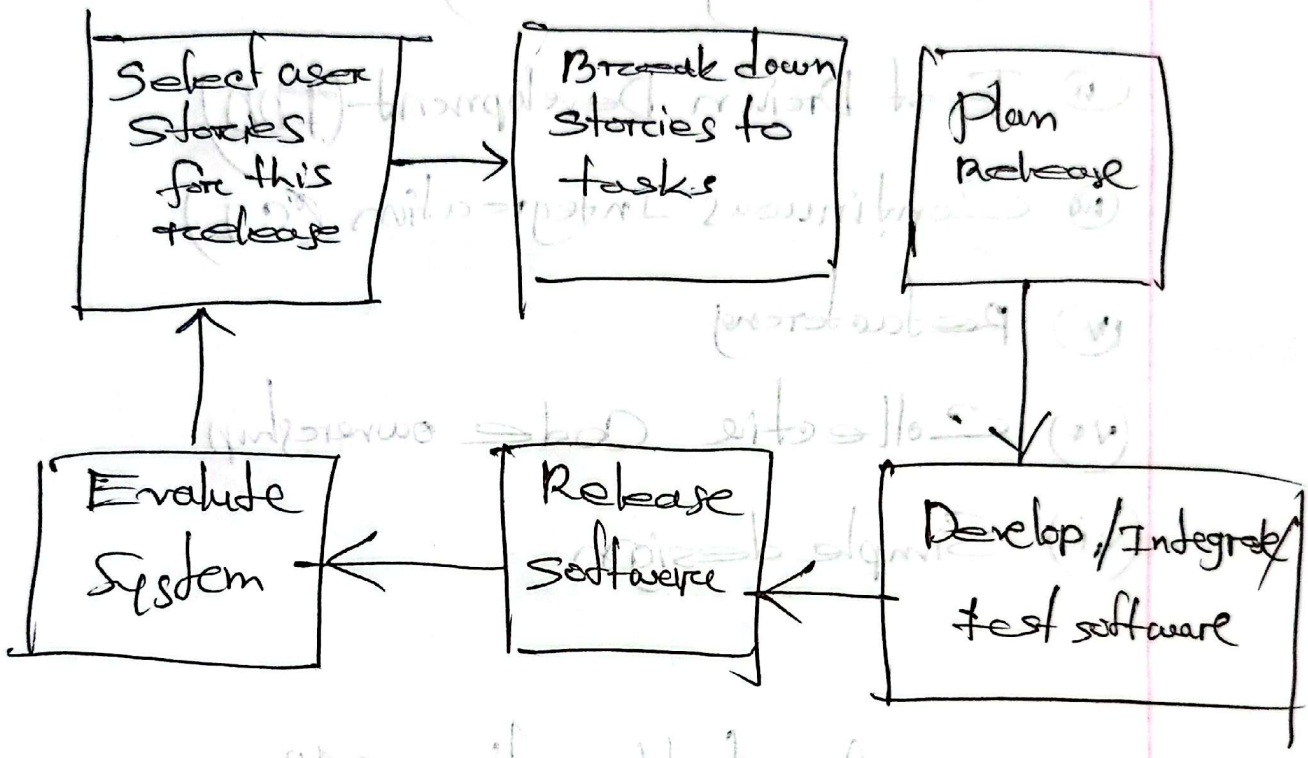
(i) Customer satisfaction
(ii) Flexibility
(iii) Faster Delivery

Challanges of Agile:

(i) Scope Creep
(ii) Coordination
(iii) Documentation

Mahmudul Hasan
IT-21019

Answer to the question no 11



(i) Panning : Initial & iteration planning define features and deliverables.

(ii) Development: Code is developed in short iteration

(iii) Continuous Testing: Automated tests ensure functionality and quality

(iv) Customer feed back

(v) Release

Influential XP Programming Practice:

(i) Paire Programming:

(ii) Test Driven Development (TDD)

(iii) Continuous Integration (CI).

(iv) Refactoring

(v) Collective Code ownership

(vi) Simple design.

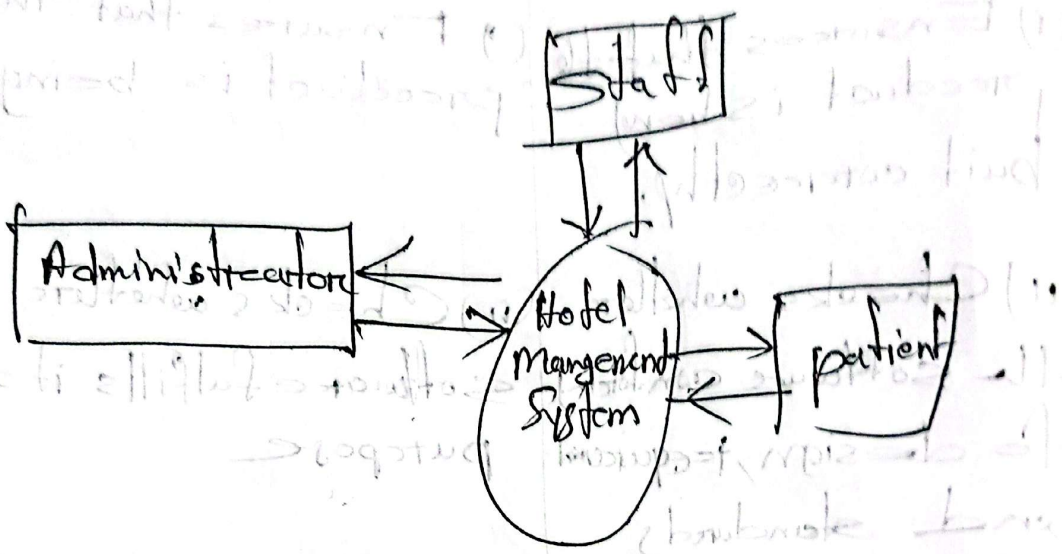
Ans. to the question no 13

Testing: Testing is the process of evaluating a system or its components to identify errors, ensure quality, and verify that it meets specified requirements.

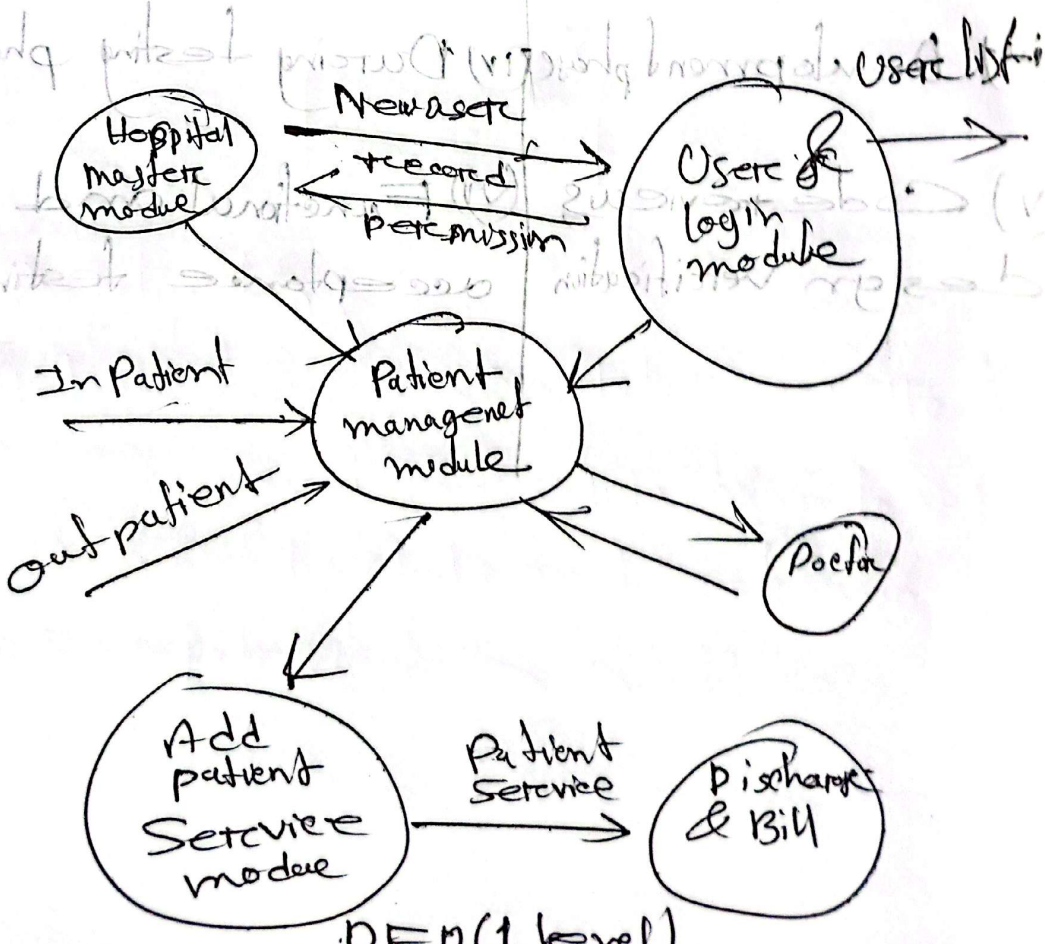
Mahmudul Hasan
IT- 21019

Verification	Validation
(i) Ensures that the product is being built correctly.	(i) Ensures that the right product is being built
(ii) Checks whether the software conforms to design, requirement and standards	(ii) Checks whether the software fulfills its intended purpose
(iii) Static process	(iii) Dynamic process
(iv) Development phase	(iv) During testing phase
(v) Code reviews, design verification	(vi) Functional and user acceptance testing

Answer to the question no 15

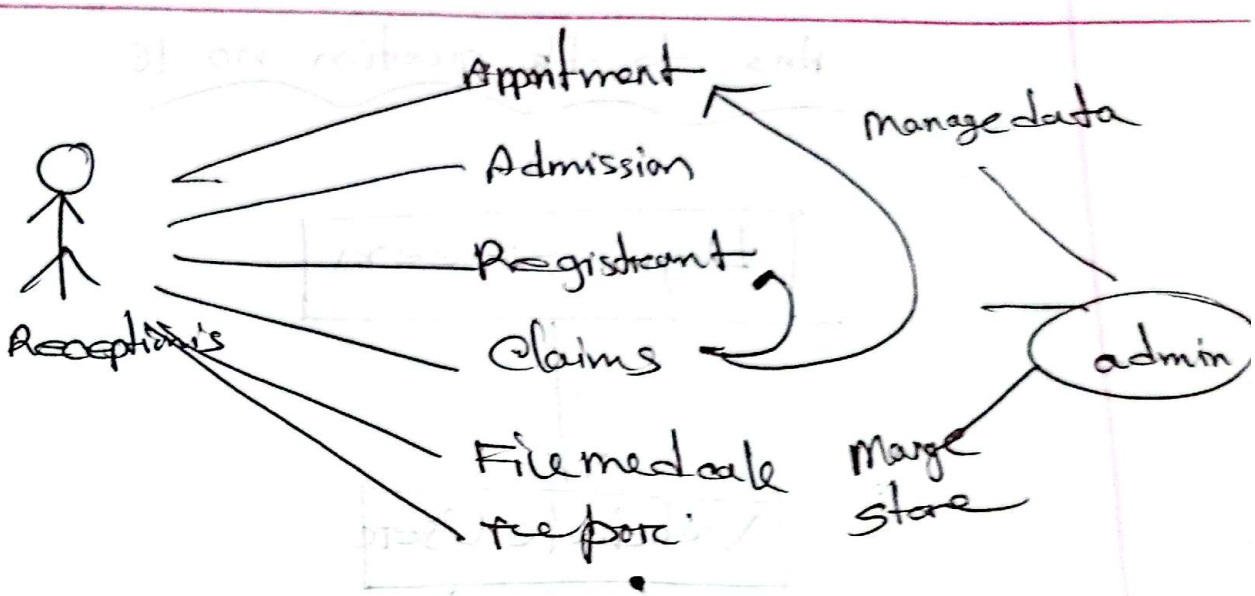


DFD (0 level)



DFD (1 level)

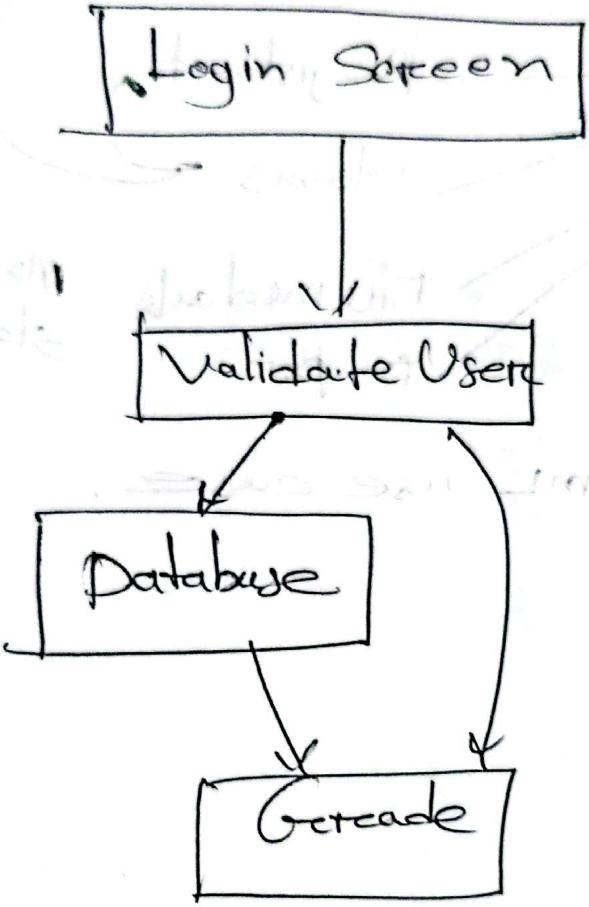
Mahmudul Hasan
IT-2019



UML use case.

A.

Ans. to the question no 16



Relationship :

(i) login Scree → ValidateUsere: The loginScreen depends on ValidateUsere to validate credentials and retrieve data

(ii) ValidateUser → Database.

Mahmudul Hasan
IT-21019

Ans. to the question no. 17

QA: It is process-oriented and proactive, focusing on preventing defects by improving processes and methodologies throughout the SDLC

QC: It is product-oriented and reactive, concentrating on identifying and fixing defects in the final product through testing and inspections.

Quality Assurance	Quality Control
(i) Process-oriented	(i) Product-oriented
(ii) Proactive	(ii) Reactive
(iii) process model, checklist, guideline, and methodologie	(iii) Test cases, bug reports test execution and reviews
(iv) Done During SDLC	(iv) after Development.

Ans. to the question no. 18

Quality Assurance (QA) is not just about finding bugs early, but also about ensuring that the software meets the required quality standards at each stage of development, optimizing the entire development process, and preventing defects rather than just detecting them. QA involves the

Role of QA in SDLC

(i) Requirement analysis: QA ensures that requires are clear, complete and testable, identifying any ambiguities or inconsistencies early.

(ii) Design: QA reviews design documents to ensure they meet requirements, are testable and address usability, security and performance.

Mahmudul Hasan
IT-21019

(iii) Coding: Prepares test cases, performs unit tests, and ensures adherence to coding standards and specification.

(iv) Testing: Conducts various types of testing

(v) Deployment

(vi) Maintainance: QA continues regression testing and supports updates, ensuring stability and addressing any new issues.

Ans. to the question no. 19

The RAD model is a software development methodology that emphasizes rapid prototyping quick iteration, and active user participation to deliver software solution quickly and efficiently

Key phase:

(i) Requirement planning
(ii) User Design
(iii) Construction
(iv) Cutover

Principles

(i) Prototyping
(ii) Iterative Development
(iii) User Involvement
(iv) Time-boxing
(v) Modularization

Advantages:

(i) Fastest Delivery
(ii) User Satisfaction
(iii) Flexibility:
(iv) Cost - Effectiveness