

UNIT₁: OVERVIEW OF SYSTEM ANALYSIS AND DESIGN

1. Introduction to system and its components and examples.
2. System Analysis and Design: Case Study: Online Shopping System.
3. Types of Information system.
 - a. Transaction Processing system(TSP)
 - b. Management Information system.
 - c. Decision Support System
 - d. Expert System
4. System Development Life Cycle(SDLC)
5. System Analysis and design tools : Prototyping ,Joint Application Design.

1. Introduction to System

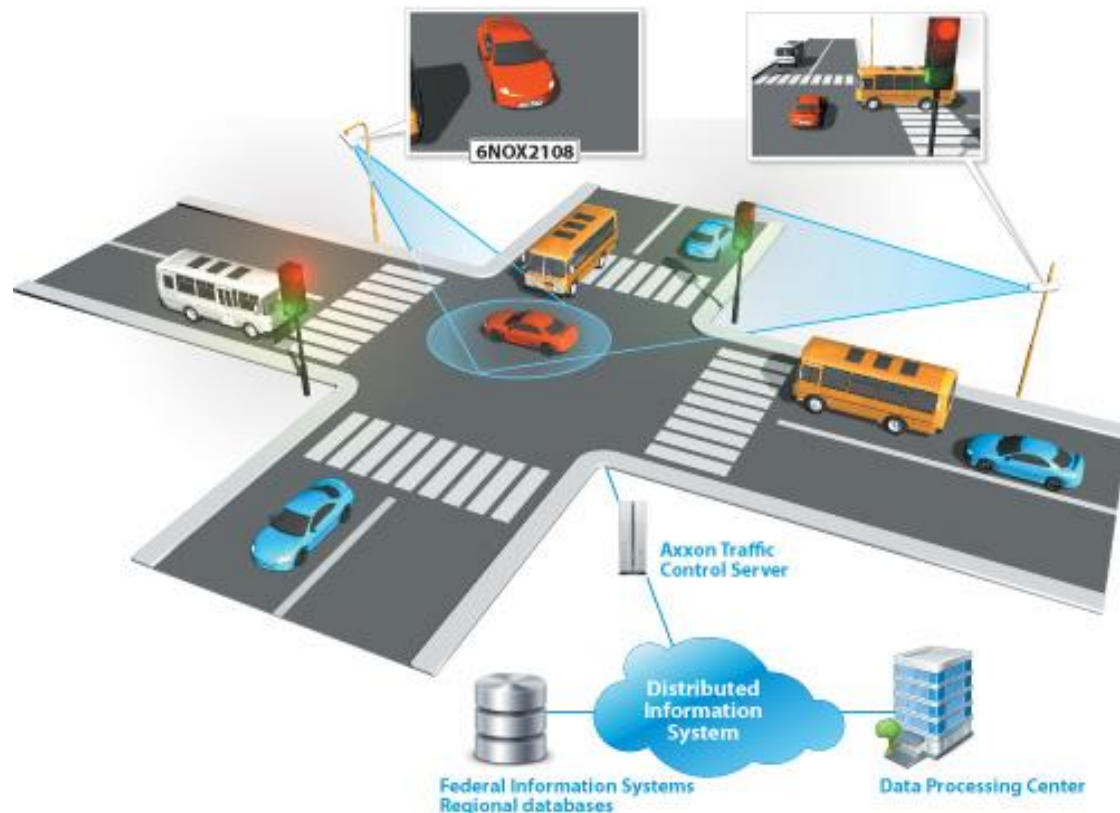
- A system is a well organized set of interconnected components or parts that work together to achieve a specific purpose or goal.
- System is not a randomly arranged set, It is arranged with some logic governed by rules, regulations, principles and policies and Systems are created to solve problems.
- One can think of the systems approach as an organized way of dealing with a problem.

Note:

By understanding how different parts of the system work together, we can make improvements, fix problems, and create well-organized, user-friendly, and reliable systems that make our lives easier and more productive.

❖ Traffic Management System:

- In a big city, there are traffic lights, road signs, cameras, and sensors all over to manage the flow of vehicles and pedestrians.
- The system's goal is to regulate traffic, prevent accidents, and keep everyone safe on the roads.



❖ Automated Irrigation System:

- In agriculture, farmers may use automated irrigation systems. These systems have sensors that measure soil moisture levels. When the soil becomes too dry, the system automatically turns on the water to keep the plants healthy. This way, the farmer doesn't have to manually water each plant.



❖ College Management System:

- In a college ,an entities could be teachers, students, classrooms, subjects, schedules/routines, grades, and more to manage. A College management system brings all these elements together, providing tools for:
- ❖ **Efficient Administration:** Administrative tasks such as student admissions, attendance tracking, fee management, and timetable scheduling etc.
- ❖ **Enhanced Communication:** Facilitates communication between administrators, teachers, students, and parents through announcements, notifications, emails, and messaging features.
- ❖ **Student Information Management:** Provides a centralized database for student information, including personal details, academic records, attendance, and assessment results. Enables quick retrieval of student information and historical data.
- ❖ **Examination and Grading:** Automates exam scheduling and result processing, saving time and reducing errors. Facilitates efficient grading and report card generation.
- ❖ **Online Fee Payment and Financial Management:** Simplifies fee payment processes by allowing online payments and generating digital receipts.



School Administrator

Dashboard

Messages

Students

Teachers

Parents

Classes

Attendance

Subjects

Marks

Exams

Events

Time Table

Import History

Notify

Payment

Dashboard

Home / Dashboard

177

Students

More info

20

Teacher

More info

102

Parents

More info

14

Class

More info

Activities

Event Exam Holidays

October 2017

today < >

month week day

Mon	Tue	Wed	Thu	Fri	Sat	Sun
25	26	27	28	29	30	1
					Weekend leave for class	Weekend leave for class
2	3	4	5	6	7	8
9 seminar					Weekend leave for class	Standard 3 - English Weekend leave for class
9	10	11	12	13	14	15
10 Annual Sports Event 10 sports	Standard 1 - Mathemat Standard 4 Winter Exam				Weekend leave for class	Weekend leave for class
16	17	18	19	20	21	22
Remedial Test	End Term				Weekend leave for class	Weekend leave for class
23	24	25	26	27	28	29
	Winter Exams 10 Cultural Fest				Weekend leave for class	Weekend leave for class
30	31	1	2	3	4	5

Exams

Date	Exam
24-10-2017 TO 31-10-2017	Winter Exams
17-10-2017 TO 19-10-2017	End Term
16-10-2017 TO 16-10-2017	Remedial Test
10-10-2017 TO 10-10-2017	Standard 1 - Mathematics
10-10-2017 TO 0000-00-00	Standard 4 Winter Exam
08-10-2017 TO 0000-00-00	Standard 3 - English

❖ Government System:

- A government system can be seen as a complex and large-scale system that involves various interconnected components, processes, and actors working together to manage a country or a political entity.

❑ Components of Government System:

- ❖ **Legislative Branch:** Law making branch of a government ,Makes laws and policies.
- ❖ **Executive Branch:** responsible for implementing laws and manages government operations. It is headed/directed by the president, prime minister, or a similar position, depending on the country's political structure.
- ❖ **Judicial Branch:** Interprets laws and ensures justice. It consists of courts and judges who resolve legal disputes and ensure that the laws are followed correctly.
- ❖ **Bureaucracy:** Bureaucracy refers to the administrative where government officials, public servants, and administrative staff who help carry out the day-to-day functions of the government. They work in various ministries and departments, such as education, health, and finance to implements laws, policies, and programs.
- ❖ **Public Services Commission:** The Public Service Commission (Lok Sewa Aayog) is a constitutional body in Nepal that is responsible for recruiting, selecting, and appointing civil servants for various government positions.
- ❖ **Election Systems:** The election system is the process by which citizens choose their representatives. It involves voting, ballot counting, and ensuring a fair and transparent electoral process.
- ❖ **Infrastructure Development:** Supports transportation, utilities, and communication.
- ❖ **Taxation:** Collects funds for public programs.
- ❖ **Foreign Relations:** Manages international affairs.
- These components collectively form the foundation of a government's role and responsibilities within a society.

❑ What is System Analysis and Design?

- Well, System analysis and design is composed of two things:

1. System Analysis.

- It is a method of studying a existing system /evaluating a particular business situation ,identifying problems and figure out what needs to be improved or created.
- In simple words ,it is a method of finding the pros and cons of an existing system for the betterment.

2. System Design:

- It is a creation of a detailed blueprint of how the new system, software, or process will work over the existing system demerits.
- This includes deciding what features it will have, how different parts will work together, and how data will be organized.

❖ Online Shopping System:

- Imagine you want to create an online shopping website. The system here would include various components such as,
 - ❑ User Interface (UI) Design: Creating visually appealing and consistent layouts. It involves using a standardized arrangement of elements such as headers, navigation menus, content sections, footers, fonts, colors, and spacing. The goal of a consistent layout is to provide a user-friendly experience for visitors as they navigate different parts of the website.
 - ❑ The User Experience(UX) Design:
 - ✓ User Experience Design in E-commerce refers to the process of creating a website that is user-friendly, intuitive, responsive and enjoyable for online shoppers.
 - ✓ It aims to enhance the overall user experience, making it easier for visitors to find products, make purchases, and have a positive interaction with the website.
 - ✓ The ultimate goal is to achieve the customer satisfaction, and enhance the business growth.

❑ Key functionalities to remember while making a User Experience Design.

- **Information Architecture:** Organizing the website's content and product categories in a intuitive manner, helping users find what they are looking for quickly and efficiently.
- **Search and Navigation:** Implementing a powerful and user-friendly search functionality, as well as clear navigation menus and filters to help users refine their search and discover relevant products easily.
- **Product Presentation:** Showcasing products with high-quality images, accurate descriptions, and customer reviews to provide users with the necessary information to make informed decisions.
- **Mobile Responsiveness:** Ensuring the website is optimized for various devices and screen sizes, especially for mobile users, as mobile shopping continues to grow.
- **Optimizing Checkout Process:** minimizing the number of steps required for a purchase to reduce cart abandonment/reject rates.
- **Security and Trust:** (e.g., secure payment options, SSL certificates) to build trust with customers.
- **Digital Footprint:** Using customer data and behavior and buying patterns, the system suggesting relevant products, showcasing offers, and displaying recently viewed items.
- **Load Times:** Optimizing the website's performance to ensure fast loading times and smooth navigation, reducing the risk of system lagging in the site due to slow speeds.
- **Accessibility:** Making the website accessible to all users, including those with disabilities, by following accessibility guidelines and providing alternative means of accessing content.
- **Customer Support:** Offering multiple channels for customer support, such as live chat, email, or phone, to assist users with any queries or issues they may encounter.

□ A Database Design:

- The role of a database in building an E-commerce website is critical as it serves as the backbone of the entire system. A well-designed and efficient database is essential for managing, storing, and retrieving data related to products, customers, orders, payments, and other aspects of the E-commerce platform.

Here are the key functions/ guidelines of a database in building an E-commerce website:

- **Product Catalog Management:** The database stores all information about the products available for sale on the website, including product names, descriptions, prices, images, and stock levels. It allows for easy management and updating of the product catalog.
- **Customer Information:** The database stores customer data, such as names, addresses, contact information, and order history. This information is crucial for processing orders, and handling customer support.
- **Order Processing and Management:** When a customer places an order, the database stores all the relevant details, including the items purchased, quantities, prices, and delivery information. It enables efficient order processing and tracking.
- **Inventory Management:** The database tracks the stock levels of products in the E-commerce store. It helps prevent overselling and allows administrators to manage inventory effectively.
- **User Accounts and Authentication:** For E-commerce websites with user accounts, the database stores account information, such as usernames, passwords and account settings. It enables user authentication and authorization for secure access to personalized features.

- **Shopping Cart Management:** The database manages the shopping cart information, including items added, quantities, and prices, as customers browse the website and add products to their cart.
- **Search and Filtering:** Product search functionality relies on the database to quickly retrieve relevant products based on search queries and filters applied by users.
- **Reviews and Ratings:** If the website allows customers to leave reviews and ratings, the database stores and manages this user-generated content to display it on product pages.
- **Promotions and Discounts:** The database may store information about ongoing promotions, discounts, and coupon codes. It ensures that customers receive the appropriate discounts during the checkout process.
- **Analytics and Reporting:** By storing data related to user interactions, purchases, and other events, the database enables generating insights through analytics and reporting tools. This data helps in understanding customer behavior and making data-driven decisions.
- **Security and Backup:** Regular backups are essential to prevent data loss in case of hardware failures or other unforeseen issues.

Note: To build an efficient database for an E-commerce website, it's crucial to consider factors such as data modeling, database schema design, normalization, indexing and database optimization techniques.

❑ A payment gateway process

- Payment gateways are essential components of E-commerce websites that facilitate secure and seamless online transactions between customers and businesses.
- Here are some popular payment gateway methods and examples used in developing E-commerce websites:
 - ❖ **Credit/Debit Card Payments.**
 - ❖ **Digital Wallets** (Int. PayPal, Google Pay, Apple Pay)(Nat. Khalti, E-Sewa, IME Pay etc)
 - ❖ **Cash on Delivery (COD):** Example: Local courier services . In COD, customers pay in cash when the order is delivered to their doorstep. It is a popular option in regions with limited online payment options or where customers prefer to pay after receiving the goods.
 - ❖ **Cryptocurrency Payments:** Some E-commerce platforms accept payments in cryptocurrencies like Bitcoin, Ethereum, or other digital currencies.
 - ❖ **Recurring Payments and Subscriptions:** E-commerce websites that offer subscription-based services or recurring billing can use payment gateways that support automatic recurring payments.

❑ Shipping and delivering products

- ❑ Providing a smooth and efficient shipping process is essential to meet customer expectations and ensure a positive shopping experience.
- ❑ Here's a breakdown of the key considerations and steps involved in managing shipping and delivery on an E-commerce website:

❖ Shipping Options and Carriers:

- Choose reliable shipping carriers based on factors like cost, speed, and coverage.
- ✓ **FedEx Express:** FedEx offers international shipping services from Nepal to various countries, known for its fast and reliable deliveries.



- **Gorkha Express:** A private courier and cargo company in Nepal offering domestic and international shipping services.
- **Sundar Yatayat:** Another private courier and cargo company that provides shipping services within Nepal and to select international destinations.
- **Mero Express:** A courier and logistics company in Nepal offering a range of domestic and international shipping options.
- **Aramex:** Aramex operates in Nepal and offers international shipping services to various destinations around the world.
- **DHL Express:** DHL is a well-known international courier company that operates in Nepal, providing global shipping and logistics solutions.

Note: All these parts work together to create a smooth online shopping experience for customers.

❏ Other Factors:

❖ Shipping Costs:

- Define your shipping cost structure based on the factors like package weight, dimensions, destination, and shipping method.
- Decide whether to offer free shipping and flat-rate shipping, based on real-time carrier calculations.

❖ Shipping Zones and Restrictions:

- Define shipping zones based on geographic regions you'll serve. Set up shipping restrictions (e.g., certain products cannot be shipped to certain countries) and specify any shipping rules or limitations.

❖ Packaging and Labeling:

- Choose appropriate packaging materials to ensure products are well-protected during transit.
- Generate and print shipping labels with tracking information for each order.



❖ Tracking and Notifications:

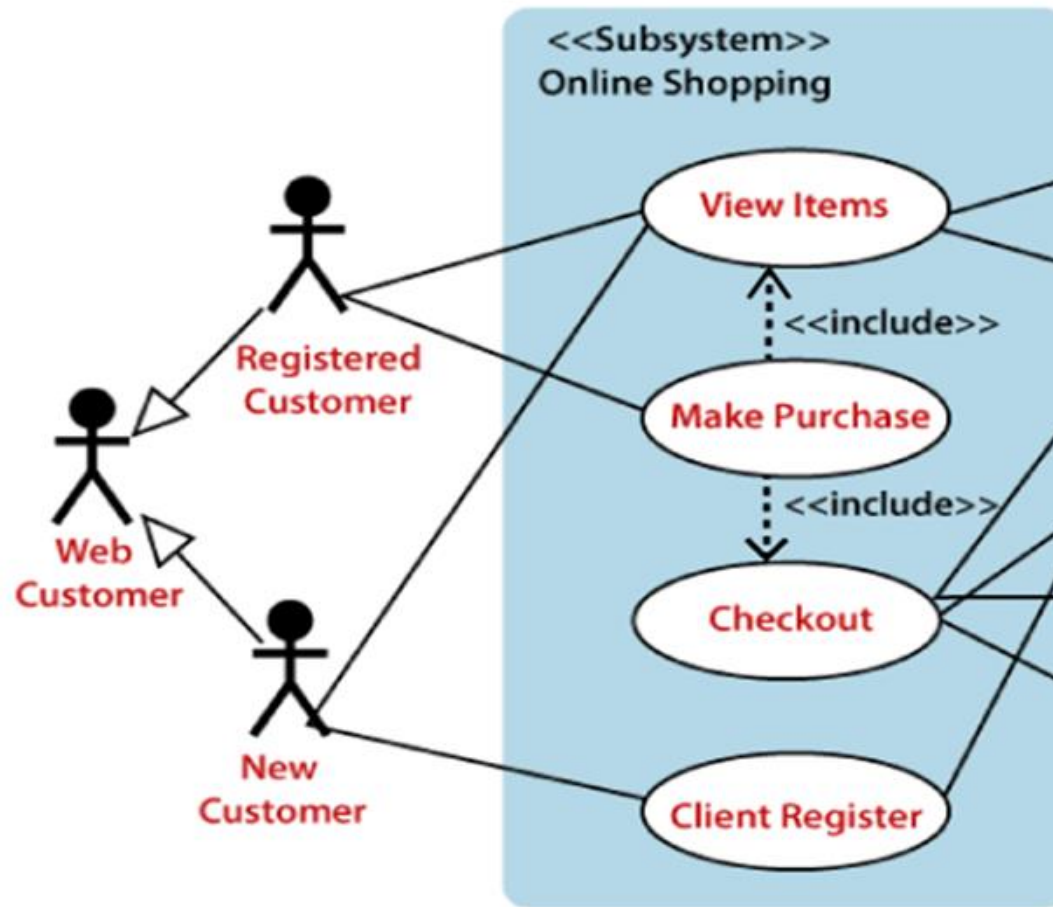
- Send automated email notifications at key stages of the shipping process, such as order confirmation, shipment tracking, and delivery confirmation to the clients.

❖ Returns and Exchanges Policy:

- Establish a clear returns and exchanges policy, including instructions for customers to initiate returns and how to handle refunds or exchanges.

Note: By effectively managing shipping and delivery processes, you can enhance customer satisfaction, reduce cart abandonment, and build trust with your audience, ultimately contributing to the success of your E-commerce website.

Online Shopping System:



❑ The Systems Components

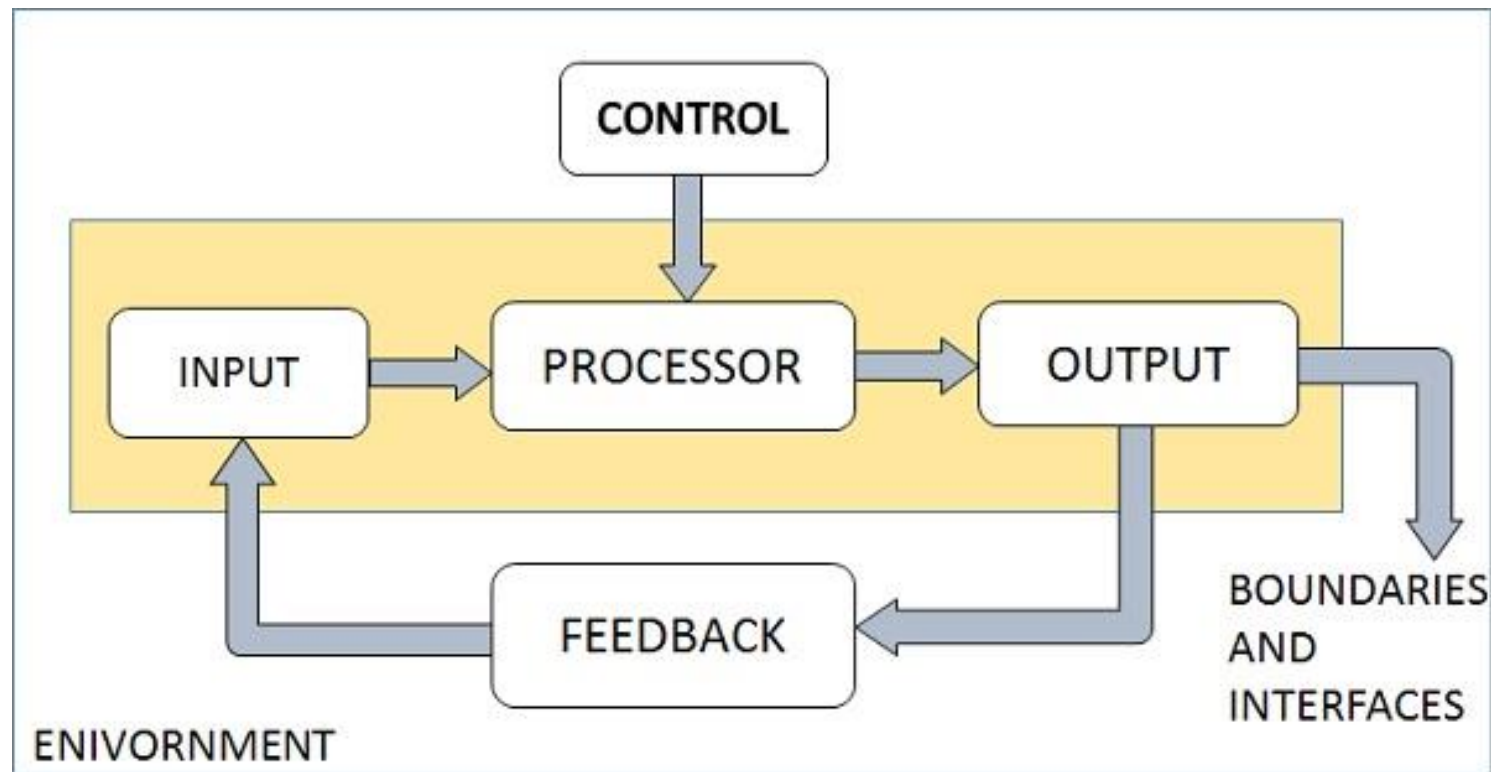


Fig: Elements/ Components of System

- In the context of system analysis and design for an E-commerce website, the concepts of input, processing, control, output, feedback, environment, and boundaries are crucial to creating a successful online shopping platform. Let's explore each of these components with reference to an E-commerce website:
- ❖ **Input:** Input in the context of an E-commerce website refers to the data or information that users provide to the system. This data includes various elements such as user registration details, shipping address, payment information, product search queries, and product reviews. Understanding the input is essential for the E-commerce website's design, as it determines how users interact with the platform and what information the system needs to process for effective online shopping.
- ❖ **Processing:** In the context of an E-commerce website, processing includes tasks such as product catalog management, order processing, inventory tracking, payment processing, and handling user accounts. The processing components ensure that users can find products easily, add items to their shopping carts, and complete transactions securely.
- ❖ **Control:** Control mechanisms in the context of an E-commerce website refer to managing the flow of processes, error handling, and ensuring the website operates as intended.
- ❖ **Output:** Output in the context of an E-commerce website represents the information and results presented to users based on their input and the system's processing. This includes product listings, shopping cart summaries, order confirmations, and delivery status updates. The output must be well-designed, user-friendly, and provide relevant information to enhance the overall shopping experience.

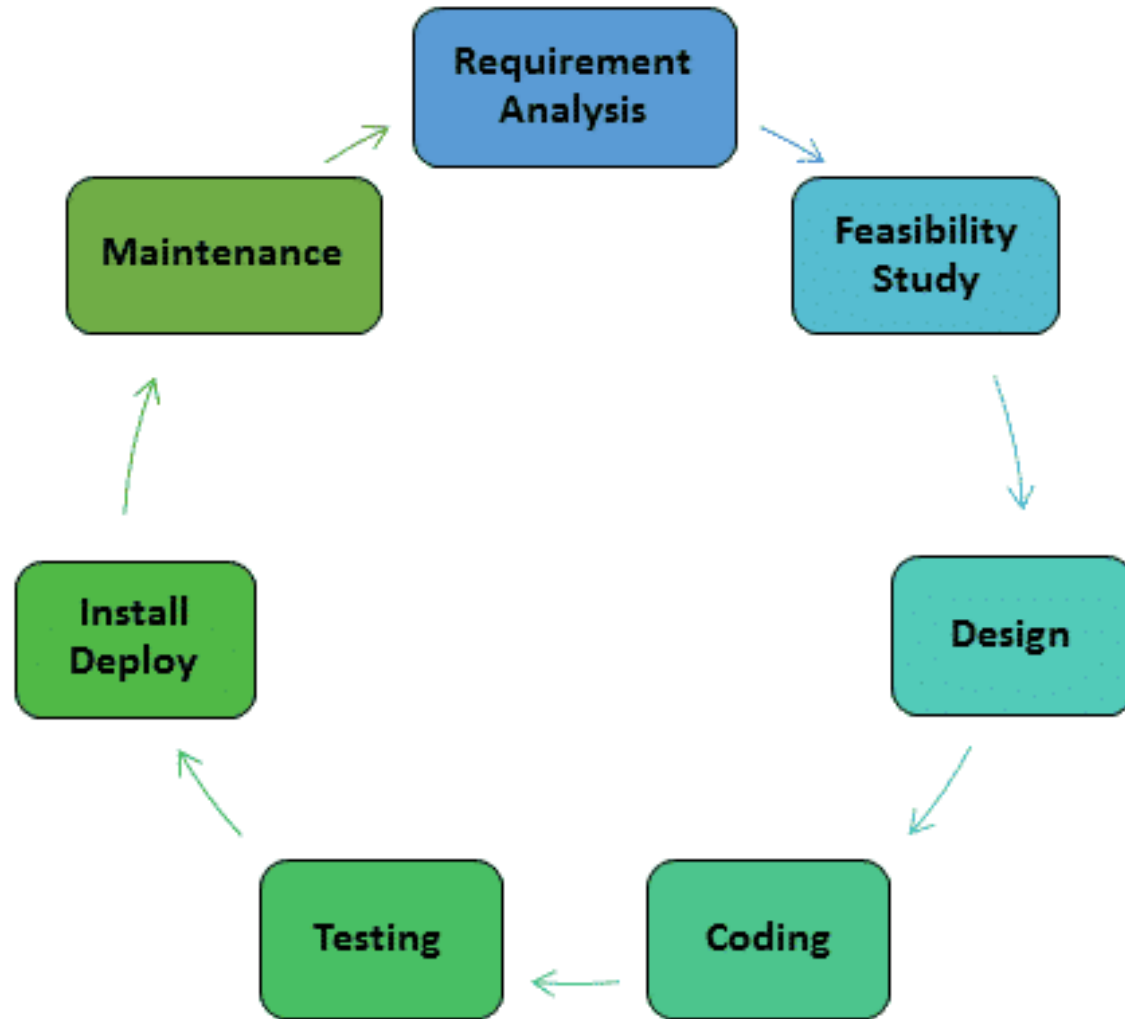
- ❖ **Feedback:** Feedback in the context of an E-commerce website involves information received from users about their experiences with the platform.
 - Feedback can be in the form of product ratings, customer reviews, and user satisfaction surveys. E-commerce websites collect and analyze this feedback to understand customer preferences, identify areas for improvement, and make necessary adjustments to enhance user satisfaction.
 - Feedback provides the control in a dynamic system. There are two types of feedback,
 - ✓ Positive feedback encourages the performance of the system.
 - ✓ Negative feedback is informational in nature that provides the controller with information for action.
- ❖ **Environment:** The environment in the context of an E-commerce website includes all **external factors** that can influence or interact with the system. This encompasses various elements such as web browsers, internet connection speeds, device types (desktop, mobile, tablet), and external payment gateways. Understanding the environment is vital for ensuring the website's compatibility and responsiveness across different devices and platforms.
- ❖ **Boundaries:** Boundaries in the context of an E-commerce website define the scope and limits of the system's functionalities. E-commerce websites may have boundaries related to the types of products sold, supported payment methods, and geographic regions for shipping.

Note: In system analysis and design for an E-commerce website, carefully analyzing and designing these components is crucial to providing a smooth and satisfying shopping experience for users. By considering input, processing, control, output, feedback, environment, and boundaries, developers can create an efficient and user-friendly E-commerce website that meets customer needs and drives business success.

❏ Software Development Life Cycle (SDLC)

- Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality software's.
- It is also called as Software Development Process.
- The SDLC aims to produce a high-quality software that meets customer expectations, reaches completion within times and cost estimates.
- SDLC provides a well-structured flow of phases(7 phases) that help an organization to quickly produce high-quality software which is well-tested and ready for production use.
- The Software Development Life Cycle (SDLC) refers to a methodology with clearly defined processes for creating high-quality software. in detail, the SDLC methodology focuses on the following phases of software development:
 - Requirement analysis
 - Planning
 - Software design such as architectural design
 - Software development
 - Testing
 - Deployment

Phases of SDLC



❖ Phase1: Requirement Gathering & Planning

- ➔ It is the first phase of SDLC in which all the necessary information's including features and functionalities of a software is collected from the clients to develop the software as per their expectation.
- ➔ The main aim of this phase is to collect the details information of each requirement of the customer so that the developers will clearly understand what they are developing and how to fulfill the customers requirement.
- ➔ During this phase, Business analyst and Project Manager set up a meeting with the customer to gather all the information like:
 - ❖ What the customer wants to build,
 - ❖ Who will be the end-user,
 - ❖ What is the purpose of the product and
 - ❖ What is the future scope of that software are discussed. Before building a product a core understanding or knowledge of the product is very important.
- ➔ In the Planning phase, project Managers evaluate the necessity of the project. This includes calculating labor and material costs, creating a timetable with target goals, and creating the project's teams and leadership structure.
- ➔ Any ambiguities/risks must be resolved in this phase only.
- ➔ This phase gives the clear picture that what we are going to build.

Phase 2: Feasibility Analysis

➔ As the name suggests ,feasibility study is a study to reveal whether a project is worth it or not. It is conducted to find answers to the following questions.

- Do we have required resources and technologies to build the project ?
- Do we receive profit from the project ? Or Whether the project is worth the investment or not?

It is the second phase of SDLC in which an organization study about the cost(including designing, developing and maintenance cost), time duration and benefits of the software, known as software requirement Specification(SRS) document.

There are mainly five types of feasibilities checks:

1. Economic: Feasibility

- ➔ In economic feasibility study , cost and benefit of the project is analyze and it ensures can we complete the project within the budget or not?
- ➔ In this feasibility study , a detailed analysis is carried out to know what will be the cost of the project including hardware and software resources required ,design and development cost and so on.

2. Legal:

- ➔ In legal feasibility study we investigate whether the project is legal or not?
- ➔ This includes analyzing barriers of legal implementation of project, such as license, copyright etc. and should follow the cyber ethics. Overall it can be said that Legal Feasibility Study is study to know if proposed project conform legal and ethical requirements.

3. Operation feasibility:

- ➔ In operational feasibility study , we examine whether the project satisfies the requirements identified in the requirement analysis phase.
- ➔ Can we create operations and produce a product which is expected by the client ?

4. Technical Feasibility:

- In technical feasibility ,we check whether we have technical resources like hardware and software to develop the project.
- This feasibility study also analyzes technical skills and capabilities of technical team ,existing technology can be used or not and Up-gradation is easy or not for chosen technology etc.

5. Scheduling Feasibility:

- ➔ In scheduling feasibility study, we estimate the time necessary to complete the project.
- ➔ Decide that the project can be completed within the given schedule/time or not.
- ➔ After the feasibility study ,the project may be accepted ,accepted with modifications or rejected.

(It means a project after feasibility study may be accepted if a company is getting benefits , accepted with modifications means with a certain adjustment the way company wants or completely rejected if company is not getting benefits from that project.)

#Importance of Feasibility Study:

- ➔ Get a clear-cut idea whether the project is likely to be successful before allocating budget , man power and time.
- ➔ Improves project teams focus.
- ➔ Provides valuable information for a “go/no-go” decision.
- ➔ Identifies a valid reason to undertake the project .
- ➔ Enhances the success rate by evaluating multiple parameters. Etc.

In essence, feasibility analysis serves as a foundation for well-informed decision-making, effective resource management, risk reduction, and alignment with organizational objectives. It enables organizations to embark on projects with a higher probability of success and to allocate resources wisely, ultimately leading to more successful and sustainable outcomes.

Phase 3: Design

- ❖ It is the third phase in which designers starts working on logical designing of the software.
- ❖ It is a diagrammatic(DFD,ERD,UC etc) representation of an entire system functionality.
- ❖ in software development, the logical design is like creating a blueprint or a detailed map for how the software will work. It's not about writing the actual code yet, but rather deciding how different parts of the software will interact with each other.
- ❖ **Some aspects of the design include:**
 - **Architecture** – Specifies programming language, industry practices, overall design, and use of any templates.
 - **User Interface** – Defines the ways customers interact with the software, and how the software responds to input.
 - **Platforms** – Defines the platforms on which the software will run, such as Apple, Android, Windows version, Linux, or even gaming consoles.
 - **Security** – Defines the measures taken to secure the application, and may include SSL traffic encryption, password protection, and secure storage of user credentials.

(SSL (Secure Sockets Layer)/TLS (Transport Layer Security)**encrypts communications between a client and server, primarily web browsers and web sites/applications.**)

- **Database Design** - The designer determines what data must be stored and how the data elements interrelate. **etc.**

Finally, this phase provides a prototype of the final product and all it includes is design of everything which has to be coded.

Phase 4: Software Development

- ❖ The development stage is the part where developers actually write code and build the application according to the earlier design documents and outlined specifications.
- ❖ The Software design is translated into source code. All the designed components of the software are implemented in this phase.
- ❖ In the coding phase, tasks are divided into units or modules and assigned to the various developers.
- ❖ It is the longest phase of the Software Development Life Cycle process.
- ❖ In this phase, Developer needs to follow certain predefined coding guidelines. They also need to use programming tools like compiler, interpreters, debugger to generate and implement the code. Different high level programming languages such as C, C++, Pascal, Java and PHP are used for coding. The programming language is chosen with respect to the type of software being developed.

Phase 5: Testing the Product

- ❖ It's critical to test an application before making it available to users. Once the software development is completed, then it is sent to the QA and Testing Team. The testing team starts testing the functionality of the entire system. It must be tested to make sure that there aren't any bugs and that the end-user experience will not negatively be affected at any point.
- ❖ Whenever a bug is found, then the software is resent to the coders to fix it and then overall software is retested.
- ❖ Depending on the skill of the developers, the complexity of the software, and the requirements for the end-user, testing can either be an extremely short phase or take a very long time but ultimately bugs or defects that need to be tracked, fixed, and later retested. This leads to a higher user satisfaction and a better usage rate.
- ❖ This process continues until the software is bug-free, stable, and working according to the business needs of that system.

Phase 6: Deployment

- After overall testing of the software and after checking that is bug free, then the software is launched and available for users .This can be as simple as a download link on the company website, or It could also be downloading an application on a smartphone.
- Even after deployment of the software ,if any bugs or errors are still found then the software is re-evaluated by the maintenance team and then it is re-deployed with New Version.

Phase 7: Maintenance

- Once the system is deployed, and customers start using the developed system, following 3 activities belongs to the maintenance depart.
- Bug fixing
 - User's Feedback
 - **Upgrade and Enhancement** – Upgrading the application to the newer versions of the Software and **Adding some new features into the existing software.**

❖ Pyramid Diagram of Organizational levels of Management and information Systems.

→ A typical organization is divided into three levels of Management: Operational, middle(Tactical) and Strategic(Senior) Levels. The information requirements for users at each level differ. Towards that end, there are number of information systems that support each level in an organization.



1. Operational Management Level

- ❖ This Level of employees are responsible for managing the day-to-day activities of the organization to ensure products and services are running smoothly and efficiently.

❑ Hospital System:

- **Nurse:** Provides patient care, administers medication, assists with medical procedures, and maintains patient records.
- **Medical Technologist:** Perform a variety of medical laboratory tests, such as blood tests, urine tests, and other body fluid analyses, to assist in diagnosing, treating, and preventing diseases as well as Prepare and distribute detailed test reports to medical staff, including doctors and nurses, in a timely manner for patient care planning.
- **Ward Clerk:** Manages patient records, schedules appointments, and assists with administrative tasks on the hospital floor.
- **Orderly:** Assists with patient transportation, maintains cleanliness of patient areas, and supports nursing staff.
- **Radiology Technician:** Operates imaging equipment to perform X-rays, MRIs, and other diagnostic procedures.

Note: Users at this level use make *structured decisions*. This means that they defined processes, rules, and procedures, that guides them while making decisions or having processes. These decisions are *repetitive and routine*.

❑ Banking System:

- ❖ **Teller:** Provides customer service at the front counter, handles deposits, withdrawals, and other basic transactions.
- ❖ **Customer Service Representative:** Assists customers with inquiries, account management, and problem resolution.
- ❖ **Bank Clerk:** Performs administrative tasks such as record-keeping, data entry, and document management.
- ❖ **Cashier:** Manages cash transactions, balances cash drawers, and ensures accuracy in financial transactions.
- ❖ **Mortgage Specialist:** Assists customers with mortgage(Collateral) applications, provides information about mortgage options, and coordinates the mortgage process.

❑ Software Company System:

- ❖ **Software Developer:** Designs, codes, and tests software applications, contributing to the development of new features and enhancements.
- ❖ **QA Tester:** Evaluates software for bugs, defects, and performance issues to ensure its quality before release.
- ❖ **Technical Support Engineer:** Assists customers with technical inquiries, troubleshoots software problems, and provides solutions.
- ❖ **UI/UX Designer:** Creates user interfaces and experiences, designing layouts, graphics, and interactions for software products.
- ❖ **Documentation Specialist:** Creates user manuals, technical documentation, and guides to help users understand and utilize software effectively.
- ❖ **Data Entry Operator:** Inputs and maintains data within software systems, ensuring accuracy and integrity of information.

2.Tactical Management(Middle Management)

- ❖ The employees of this level of organization level includes : *Project Managers, heads of departments(HOD), supervisors, leaders etc.* The users at this level usually oversee and guides the activities of the users at the operational management level.
- ❖ For instance, In a software company, tactical-level employees are responsible for translating strategic goals into actionable plans, managing projects, teams, and resources, and ensuring the successful execution of software development and delivery. These individuals play a crucial role in bridging the gap between high-level strategy and day-to-day operations.
- ❖ This level of employs makes Semi-structured decisions that fall between completely *unstructured decisions and fully structured decisions.*
- ❖ In semi-structured decisions, some aspects of the decision-making process are well-defined and follow established procedures and rules/routines while other aspects are situation oriented means circumstances surrounding them are uncertain, unique, complex , and non-repetitive situations.

3.Strategic Management(Senior Management)

- ❖ This is the most senior level in an organization. The Company owner, CEO, Founders at this level make unstructured decisions. Senior level managers are concerned with the long-term planning of the organization. They use information from tactical managers and external data to guide them when making unstructured decisions.
- ❖ This level of employs makes tactical (Unstructured) decisions which are completely situation oriented rather than a routine means circumstances surrounding them are uncertain, unique, complex , and non-repetitive situations.
- ❖ Means, selection of appropriate/optimal solution based on the current situations and scenario of system to achieve a target/goal.
- ❑ Example of decisions:
 - ❖ Decides entrance or exit from markets.
 - ❖ Approve capital budget/financials decisions.
 - ❖ Decides long-term goals etc.
 - ❖ Choosing a vendors.
 - ❖ Establishing partnerships to enhance the company's position in the industry.

Note: In summary, operational management deals with the day-to-day operations, tactical management involves achieving specific goals, and strategic management focuses on the long-term direction and vision of an organization. Each level of management plays a crucial role in ensuring the company's success and growth.

Information System

- ❖ An information system (IS) is a set of interconnected components that work together to collect, process, store, and distribute information to support decision-making, coordination, control, analysis, and visualization in an organization or enterprise.
- ❖ It encompasses both the technology and human elements involved in managing and using information.

Key components of an information system include:

- ❖ **Data:** Raw facts and figures that are collected and stored for processing.
- ❖ **Hardware:** Physical devices, such as computers, servers, networking equipment, and storage devices, that handle data processing and storage.
- ❖ **Software:** Programs and applications that are used to process, manipulate, and analyze data.
- ❖ **Networks:** Communication channels that enable the exchange of data and information between different components of the information system.
- ❖ **Procedures:** The rules, policies, and protocols that govern how data is collected, processed, and used within the system.
- ❖ **People:** Individuals who interact with the information system, including users, administrators, developers, and support staff.

☐ Types of Information System

- ❖ Transaction Processing system(TPS)
- ❖ Management Information system(MIS)
- ❖ Decision Support System(DSS)
- ❖ Expert System(ES)

1. Transaction Processing System (TPS)

- A Transaction Processing System (TPS) is an information system designed to manage and process day-to-day business transactions of an organization.
- There are two types of TPS: **Batch transaction processing** and **Real time processing**.
- ❑ **Batch transaction processing** : In this approach, data is collected over a period of time and then processed all at once. Batch processing is typically used when immediate processing is not required, and data can be accumulated and processed periodically, such as daily, hourly, weekly or monthly.
- ❑ **Examples of BTP include:**
 - ❖ **Pay cheques and other forms of paper output.**
 - ❖ **Payroll Processing:** Companies often process employee payrolls in batches. Employee time and attendance data are collected over a specific period, and then payroll calculations, including salary, taxes, and deductions, are performed for all employees in a single batch.
 - ❖ **Inventory Management:** Retailers and wholesalers use batch processing to manage their inventory. They might process orders received during the day in a batch, updating stock levels and generating reports on sales and stock availability.

- ❖ Whereas, *Real time processing* also known as online processing, involves the immediate processing of transactions as they occur.
- ❖ In this approach, data is processed and updated in real-time, providing instant results.
- ❖ Real-time processing is essential when immediate responses or actions are required based on the input data.

❖ **Examples:**

- **Online Payments:** When you make a purchase using a credit or debit card online, the payment gateway processes the transaction in real-time, verifying the card details and ensuring there are sufficient funds for the purchase.
- **Traffic Management:** Smart traffic management systems process real-time data from sensors, cameras, and GPS devices to monitor traffic conditions and adjust traffic signals or routes in response to current congestion levels.
- **ATM Withdrawals:** When you withdraw money from an ATM, the transaction is processed in real-time to ensure your account has the necessary balance and to dispense the requested amount immediately.
- **Online Customer Support:** Chatbots and customer support systems often provide real-time responses to customer inquiries, assisting with immediate solutions or routing inquiries to human agents.
- **Social Media Interactions:** Social media platforms process real-time interactions such as likes, comments, and shares instantly, reflecting these actions to users and their networks in real-time.

❑ Decision Support System

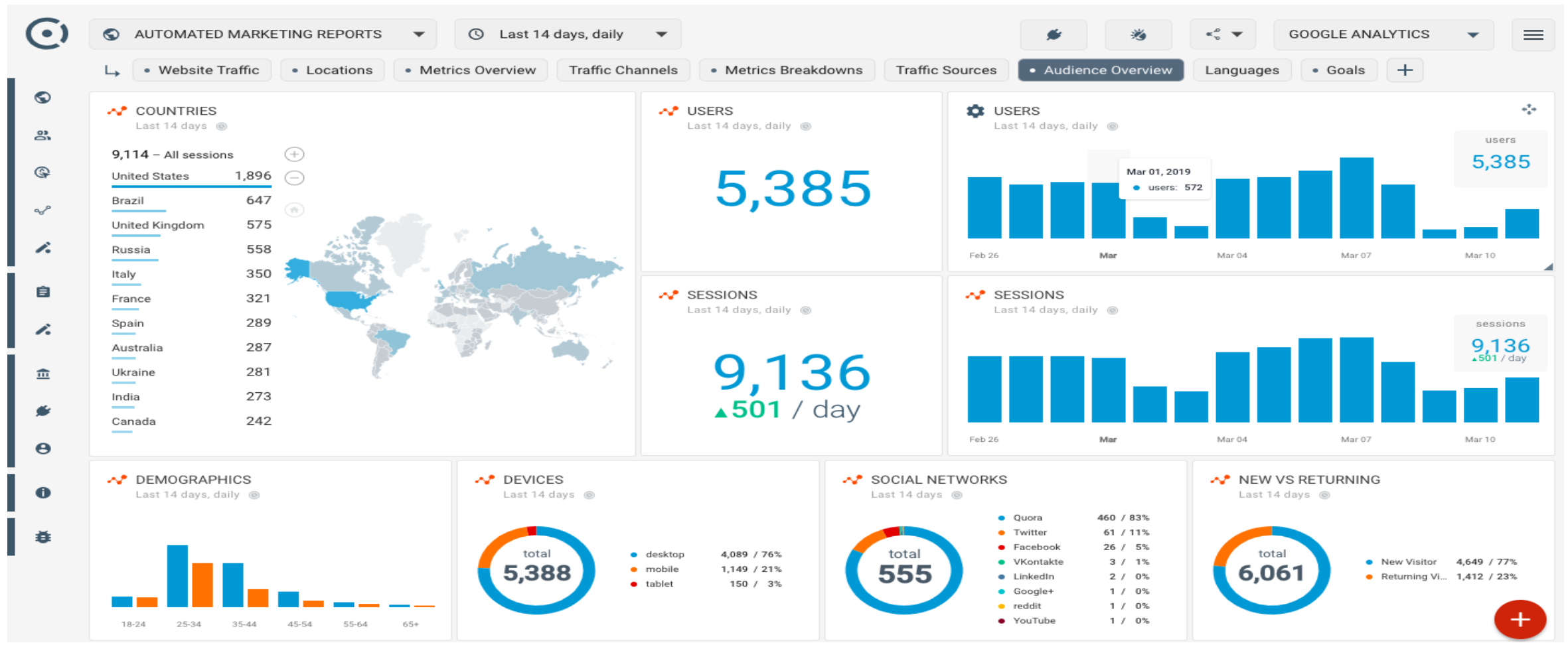
- ❖ A Decision Support System (DSS) is an information system that assists users in making complex decisions by providing relevant data, models, and tools for analysis.
- ❖ DSS helps individuals, managers, and organizations explore various alternatives, understand potential outcomes, and make informed decisions based on data-driven insights.
- ❖ These systems are designed to support decision-making at different levels of an organization, from operational to strategic decision-making.

❑ Here are some practical applications of DSS:

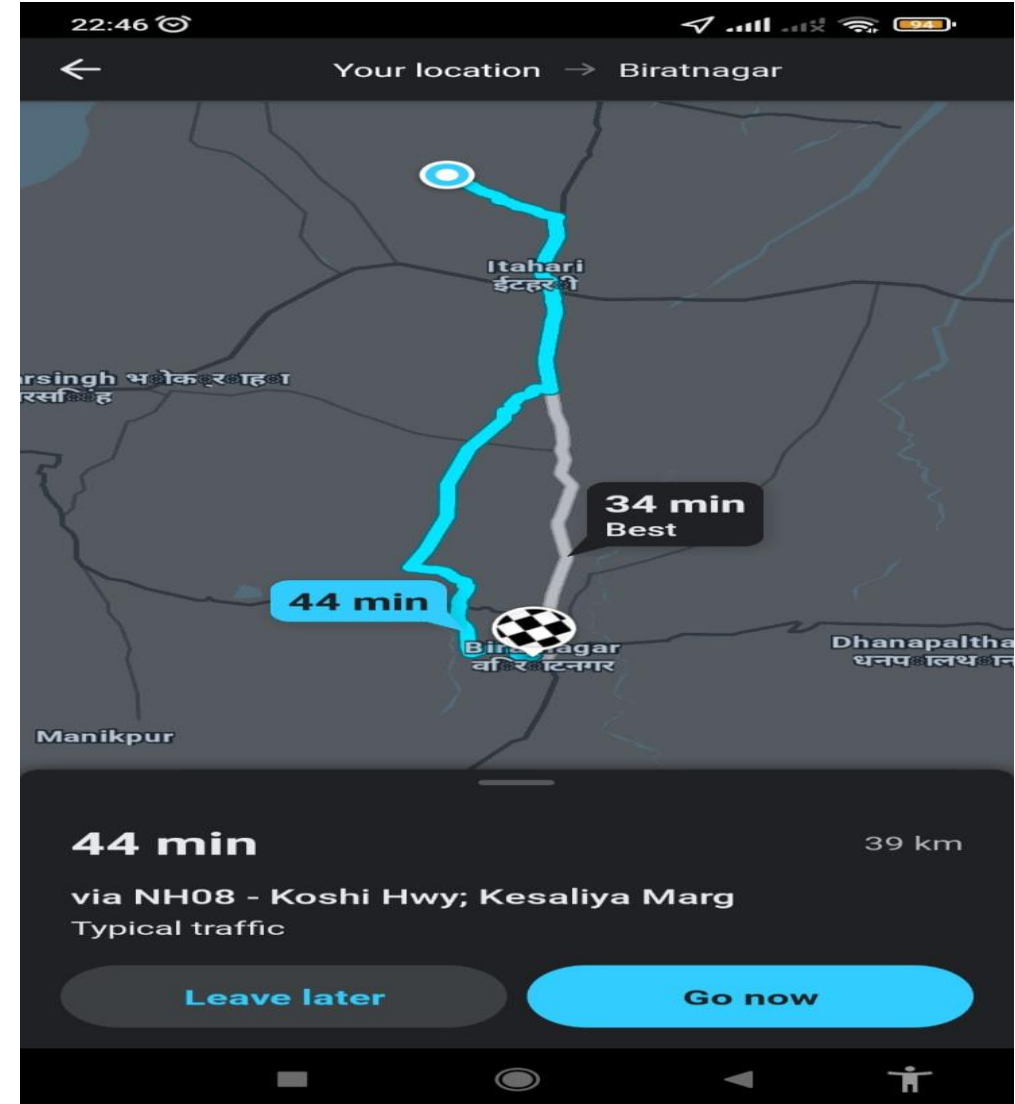
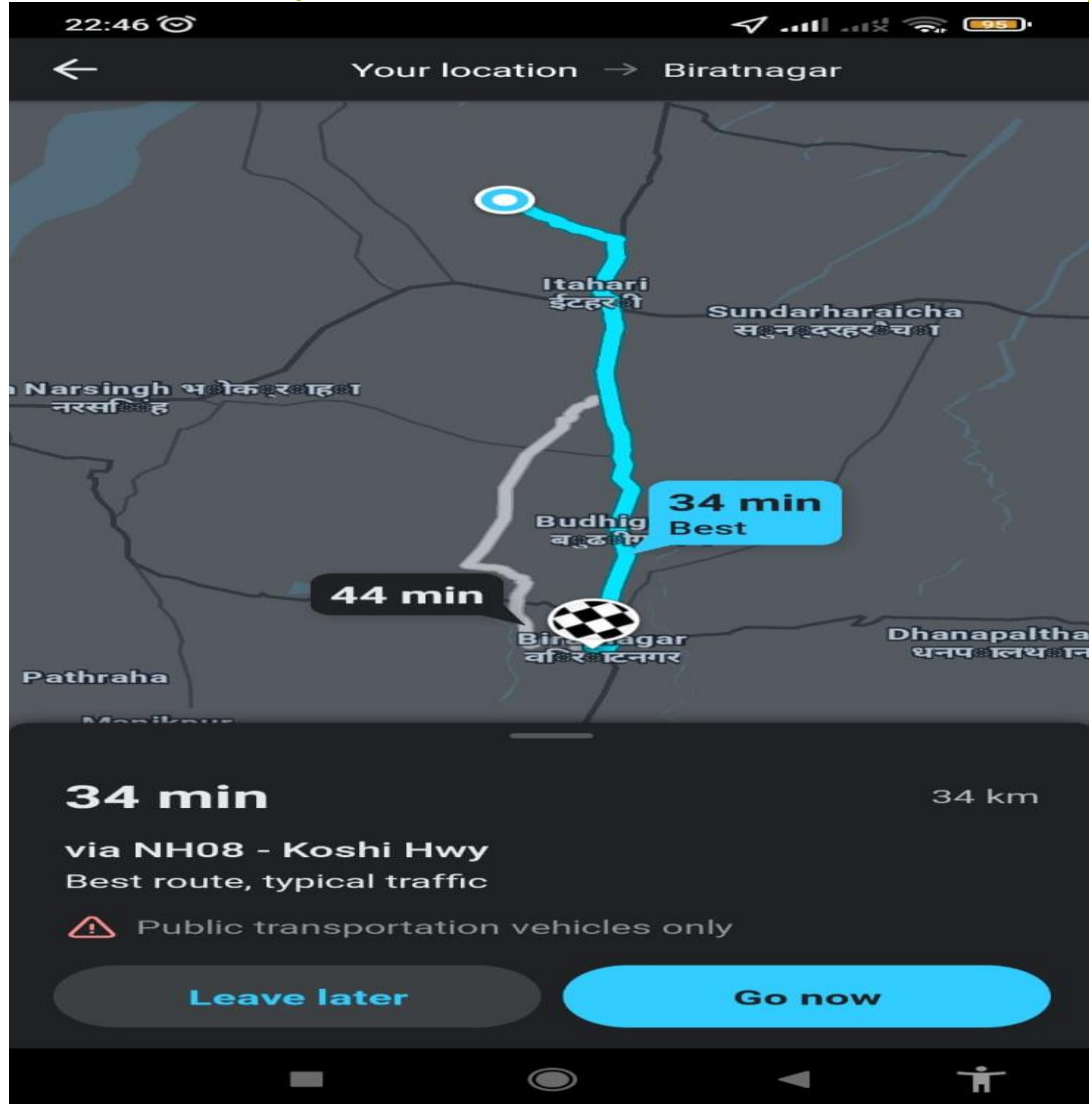
- **Business Planning and Strategy:** DSS helps businesses in strategic planning, budgeting, and forecasting. By analyzing data and market trends, DSS assists in making informed decisions about resource allocation, product development, and market entry.
- **Healthcare and Medical Diagnosis:** DSS aids healthcare professionals in diagnosing and treating patients. By analyzing patient data it suggests potential diagnoses and treatment options.
- **Marketing and Customer Relationship Management (CRM):** DSS helps marketers understand consumer behavior, markets, and optimize marketing strategies. It uses data from customer interactions, surveys, and social media to provide data-driven marketing insights.
- **Tourism and Travel Planning:** DSS supports tourists in planning trips, suggesting , and recommending attractions based on their preferences and budget.

Examples:

- **Google Analytics:** A web analytics service that provides website and app owners with data and insights on user behavior, traffic sources, and site performance, helping them make data-driven decisions to improve their online presence.

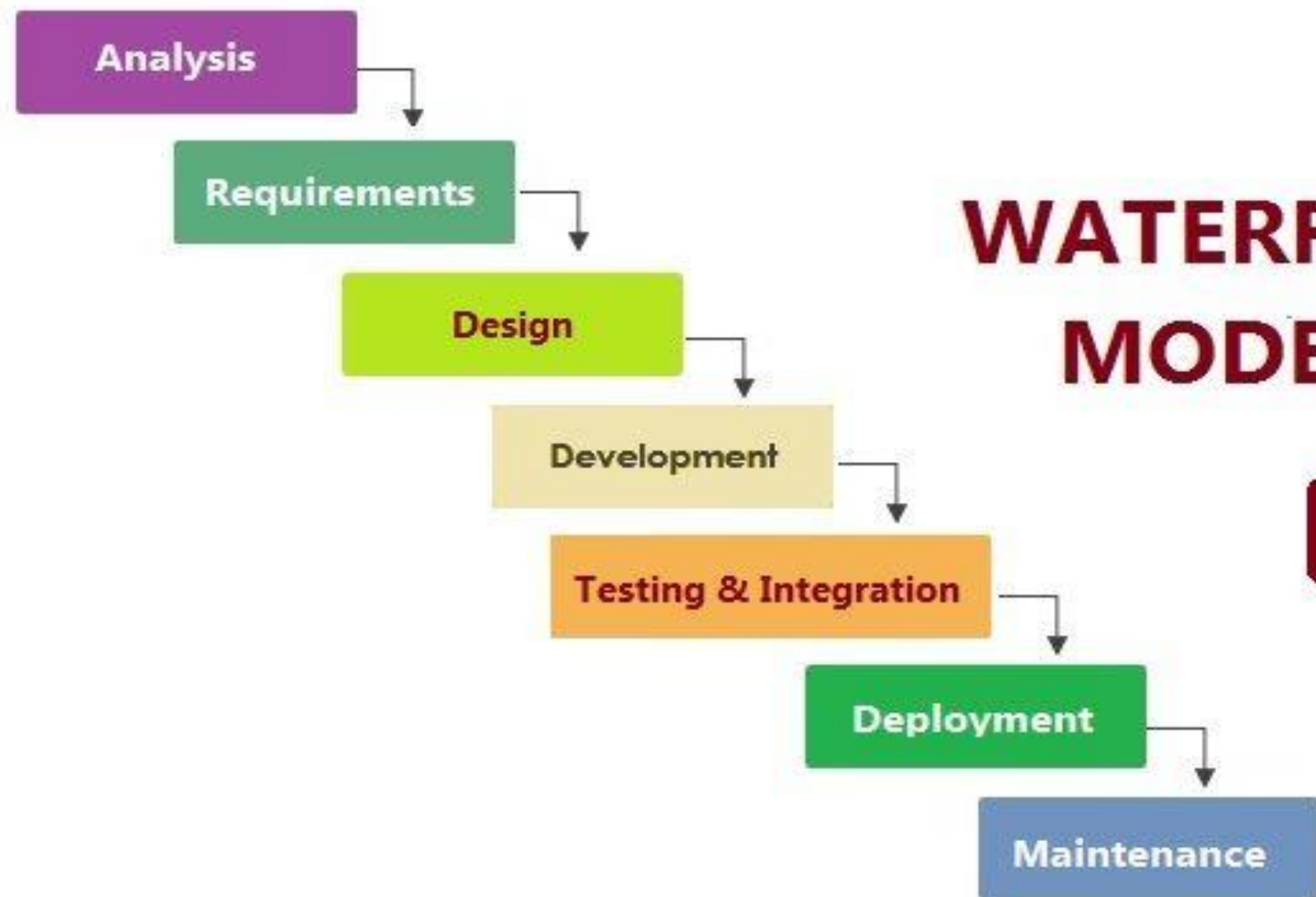


- **Waze:** A navigation app that utilizes real-time traffic data to help drivers find the best routes, enabling users to make informed decisions on their travel plans. It also provides driving directions, live traffic and road conditions update.



Management Information System

- ***Inventory Management System:*** An inventory management system is a management information system that tracks and manages inventory levels, ensuring businesses have the right amount of products in stock.
- ***Health Information System:*** A health information system is a management information system used in healthcare settings to manage patient records, appointments, medical history, and other health-related data.
- ***School Management System:*** A school management system is a management information system used in educational institutions to manage student records, class schedules, grades, and other administrative tasks.
- ***Hotel Management System:*** A hotel management system is a management information system used in the hospitality industry to manage room bookings, guest information, and hotel operations.
- ***Project Management System:*** A project management system is a management information system that helps teams plan, track, and collaborate on projects, ensuring they are completed on time and within budget.



WATERFALL MODEL

pmvidya.com

2. Spiral Model

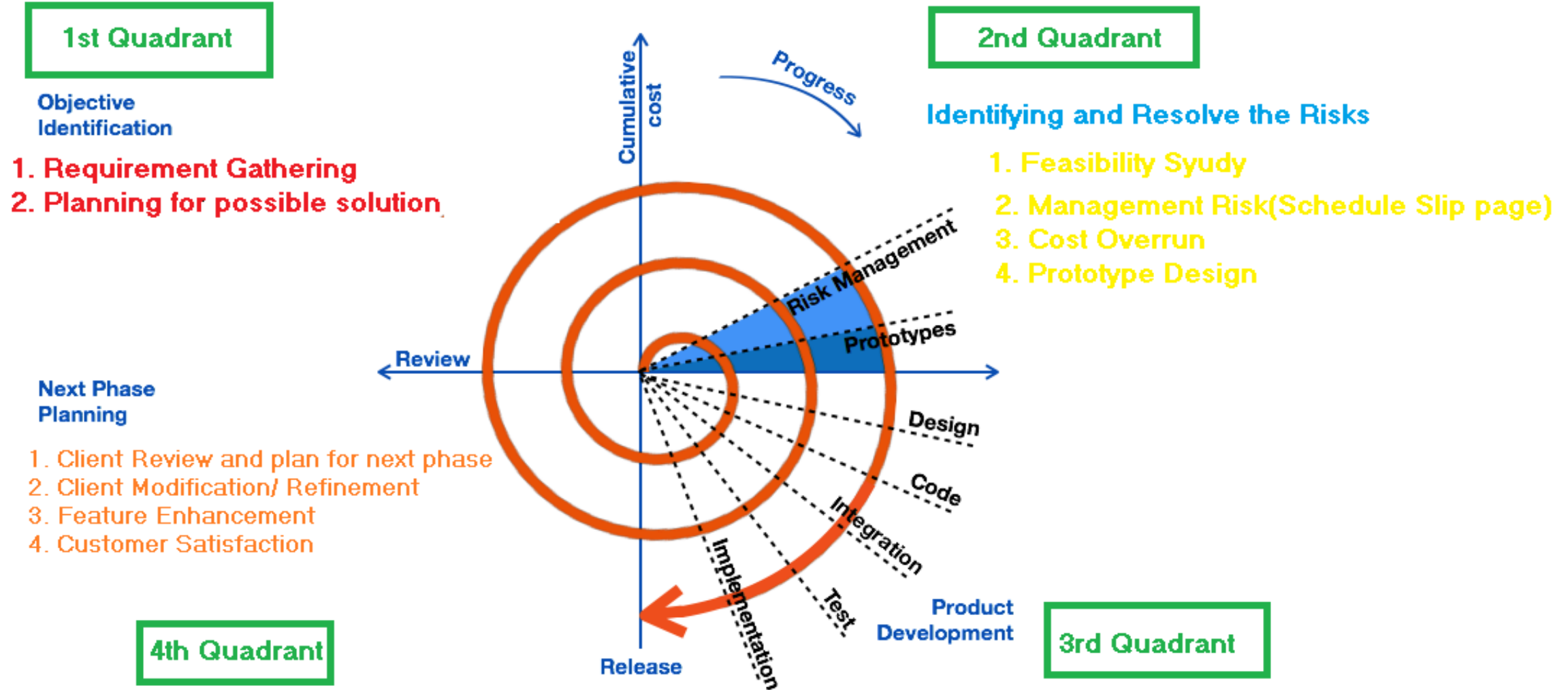


Fig: Spiral Model

❏ KEY POINTS

- ❖ It was introduced by Barry Boehm in 1986 as a response to the limitations of traditional linear approaches like the Waterfall model.
- ❖ The Spiral Model aims to address the challenges posed by changing requirements, evolving technology, and the inherent risks associated with software development.
- ❖ Organizations that have *complex and critical projects* with high levels of uncertainty, changing requirements, and significant risks may find the Spiral Model to be a suitable choice.
- ❖ Radius of spiral is directly proportional to the cost.
- ❖ Angular dimension = Progress.
- ❖ It is a meta Model.
- Uses classical waterfall model.
- Takes feedback in each spiral(Iterative).
- Use of Prototype Model.
- ❖ Used to handle large, complex and risky Projects.
- ❖ Used by ISRO(Indian Space Research Organization) , NASA and Multinational Companies.
- ❖ High level of Satisfaction can be achieved because spiral model continuously revolves in angular dimension.

❑ Key advantages of the Spiral Model include:

- ❖ **Risk Management:** The model provides a systematic approach to identifying, assessing, and managing project risks, which can lead to better risk mitigation strategies and ultimately a more successful project.
- ❖ **Flexibility:** The iterative nature of the model allows for changes to be incorporated at various stages, making it well-suited for projects with evolving requirements or uncertain environments.
- ❖ **User Involvement:** Regular user feedback is integral to the process, ensuring that the software aligns with user needs and expectations.
- ❖ **Quality Focus:** The iterative cycles allow for continuous evaluation and improvement of the software's quality.

❑ Disadvantages of the Spiral Model include:

- ❖ **Complexity:** The risk analysis and management aspects can make the model more complex and resource-intensive.
- ❖ **Documentation:** The model's iterative nature may require careful documentation and communication to maintain a clear project trajectory.
- ❖ **Resource Management:** The flexibility of the model can sometimes lead to scope creep and potential challenges in resource management.

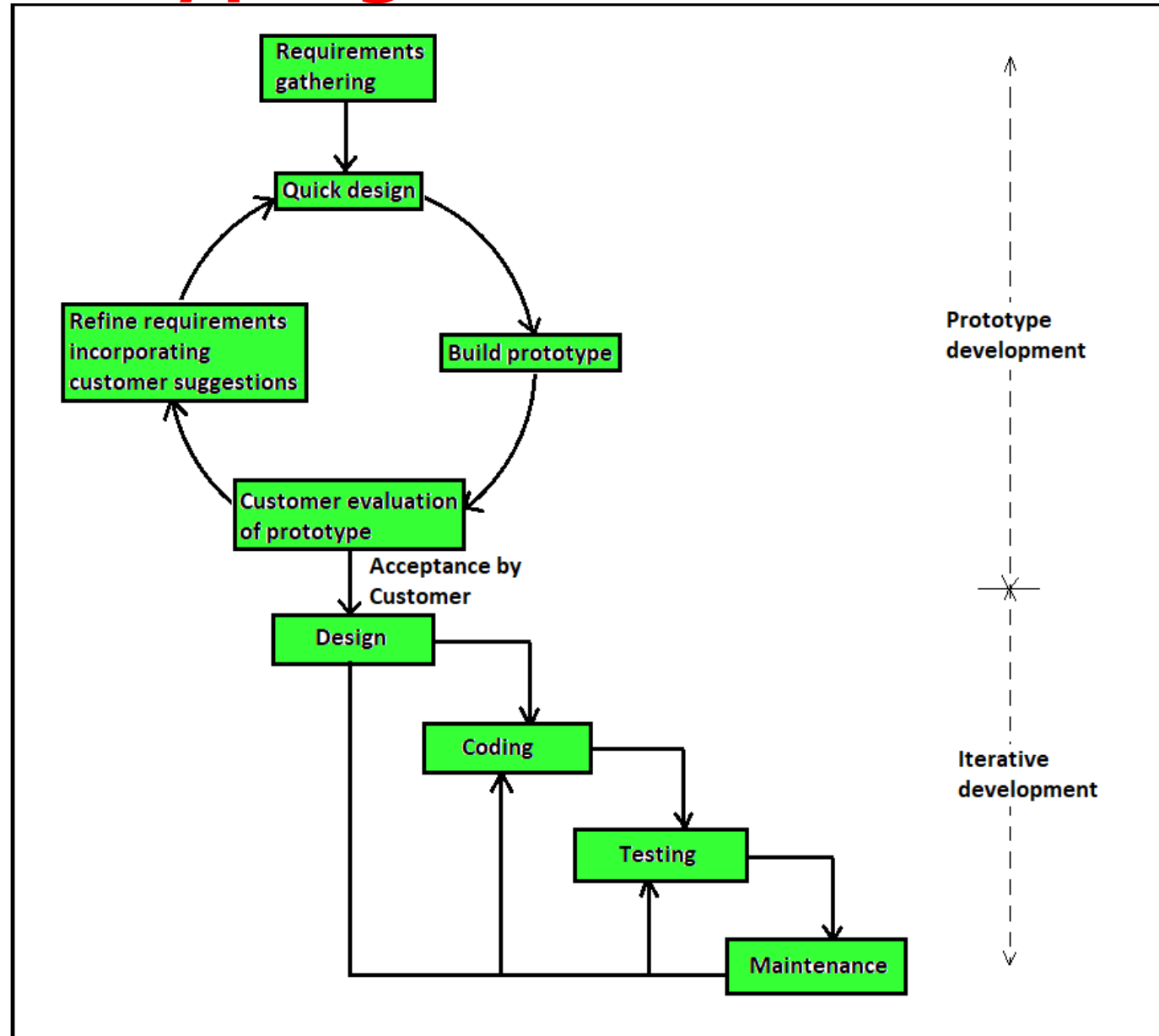
Note:

- The Spiral Model is particularly well-suited for large, complex projects where risks and uncertainties are significant.
- The Spiral Model is still used in modern software development practices, although it's not as commonly used as some other methodologies like Agile or Scrum. The choice of which development model to use depends on the nature of the project, the organization's preferences, and the specific needs of the development team and stakeholders.
- However, many organizations have shifted toward more lightweight and Agile methodologies in recent years due to their emphasis on rapid iterations, customer collaboration, and quicker delivery of working software.

3. Prototyping Model(Throwaway model)

- ❖ The Prototyping Model is a software development process model that focuses on building a preliminary version of the software, **called a prototype**, to better understand and refine the requirements and design before proceeding with full-scale development it as per client requirements, until a final acceptance product is achieved.
- ❖ It's particularly useful when the requirements are not well-defined.
- ❖ A prototype is a working model of software with some limited functionality.
- ❖ A prototype always holds the exact logic used in the actual software application because the prototype is a ***dummy or toy model***.

1. Prototyping Model(throwaway model)



❑ Here's an in-depth explanation of the Prototyping Model:

- ❖ **Requirements Gathering:** In this initial phase, the development team collaborates with stakeholders to gather the initial requirements of the software. These requirements may be vague, incomplete, or subject to change. The goal is to understand the general objectives and functionality that the software should have.
- ❖ **Quick Design and Prototyping:** Based on the gathered requirements, a preliminary design of the software is created. The design doesn't need to be comprehensive; it's focused on creating a prototype that demonstrates the core features and functionalities. The prototype is a basic version of the software that provides a tangible representation of the proposed system.
- ❖ **Prototype Development:** The prototype is developed rapidly, often using tools and technologies that enable quick development. The focus is on implementing the key features and functionalities identified in the requirements. The prototype might lack full functionality, and some parts might be simulated or stubbed out for demonstration purposes.
- ❖ **Prototype Evaluation:** Once the prototype is ready, it's shared with stakeholders, including users, for evaluation. This evaluation phase is crucial as it allows stakeholders to interact with the prototype and provide feedback. Feedback can reveal misunderstandings, missing features, or requirements that need further clarification.
- ❖ **Refinement and Iteration:** Based on the feedback received during the evaluation phase, the development team refines the prototype. This could involve adding new features, modifying existing ones, or clarifying requirements. The prototype is iteratively improved and enhanced based on stakeholder input.
- ❖ **Final Product Development:** After several iterations of prototyping, and as the requirements become clearer and more stable, the development team transitions from prototype development to full-scale development. The lessons learned from the prototype phase help in building a more accurate and aligned final product.
- ❖ **Testing and Deployment:** The final product undergoes testing to ensure that it meets the quality standards and fulfills the refined requirements. Once testing is complete, the software is deployed to the production environment for actual use.

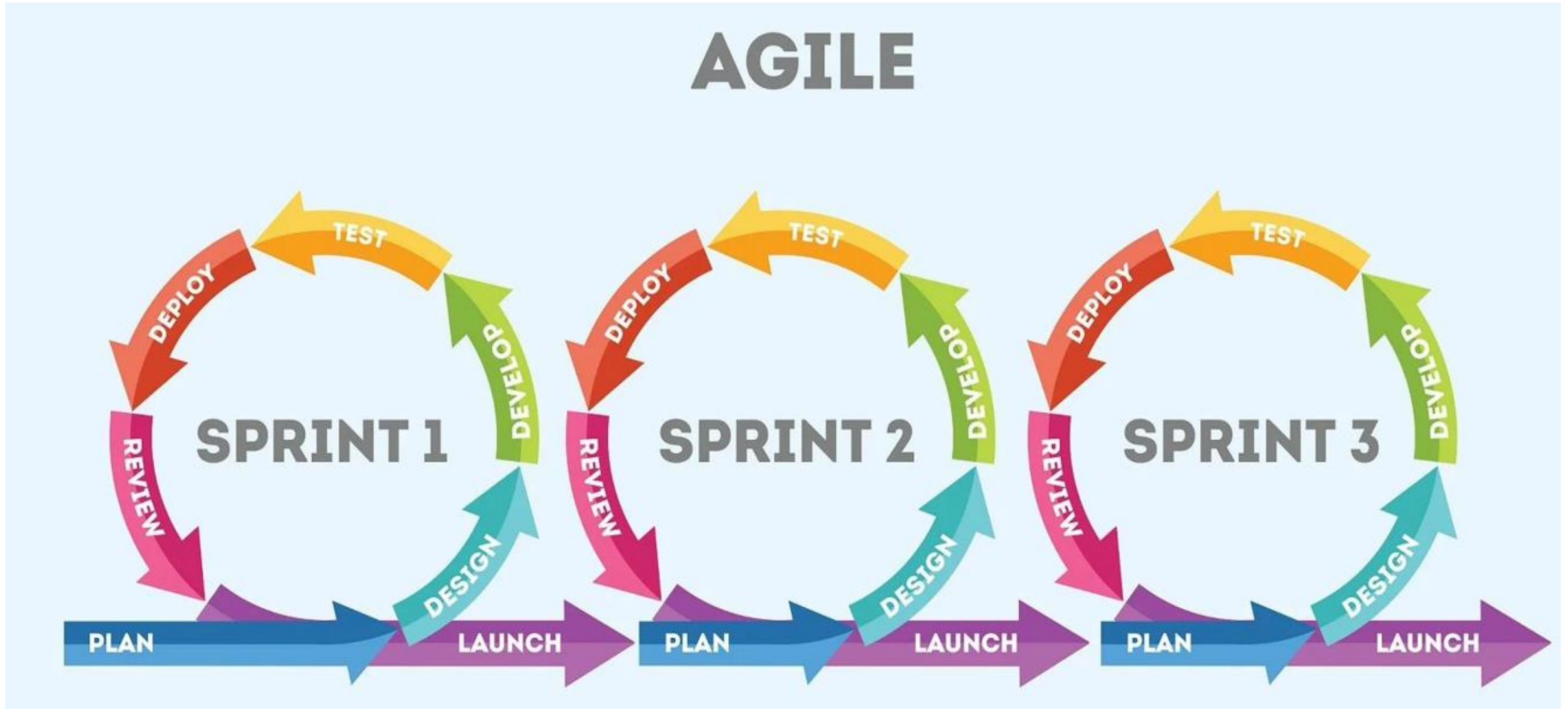
❑ Key advantages of the Prototyping Model include:

- **Early Feedback:** Stakeholders get a hands-on experience with the prototype early in the process, allowing for rapid feedback and validation of requirements.
- **Requirement Refinement:** The iterative nature of the model allows for continuous refinement of requirements based on user interactions with the prototype.
- **Reduced Risk:** The prototype serves as a risk reduction mechanism by identifying potential issues and misunderstandings early in the project.
- **User Involvement:** User involvement in the evaluation phase ensures that the software aligns with user expectations and needs.
- **Reusability:** Developed prototype can be used later for any similar projects.

❑ Disadvantages of the Prototyping Model include:

- ❖ Prototyping may be a slower and time taking process.
- ❖ Poor documentation due to the changes in the requirement.
- ❖ Regular meeting are vital to keep the project on time.
- ❖ Increase in cost of development.
- ❖ **Scope Creep(project's requirements tend to increase over time.):** Frequent changes during the prototyping phase can lead to scope creep if not managed effectively.
- ❖ **Incomplete Features:** Due to the focus on rapid development, some features may not be fully implemented in the prototype.

4. Agile or Scrum Methodology



❖ KEY POINTS

- ❖ The Agile methodology is a Iterative project management approach that involves breaking the project into phases called **Sprint** and emphasizes continuous collaboration and improvement.
- ❖ It is an approach to divide the big project into smaller chunks and develop parallely to deployed in market quickly.
- ❖ Each sprint typically lasts 2 to 4 weeks.
- ❖ Each sprint is intend to develop a **module** of an application not an entire application.
- ❖ Consumable output in each sprint.
- ❖ Latest Model/ Methodology: Used by major Companies like
 - ❑ Facebook
 - ❑ Amazon
 - ❑ Google for quick update and frequent changes.

- ❖ Less documentation oriented rather it focuses on quick development.
- ❖ Each iteration involves a team working through SDLC phases.

❑ Agile methodologies vs. traditional approaches

- So, how do these Agile methodologies compare to traditional approaches to project management? Let's highlight the Waterfall approach as an example.
- When working with this traditional methodology, teams would follow a strictly linear sequence: requirements gathering, design, build, test, deliver. They are required to complete one phase before moving on to the next one. Changes are difficult to incorporate once a stage is completed and customer interactions are limited. As a result, Waterfall suits projects with fixed guidelines and minimal changes.
- By comparison, Agile methodologies are far more fluid in nature. Every Agile framework emphasizes a degree of adaptability, breaking projects into phases and embracing changing requirements. Through iterations and incremental efforts, they incorporate collaboration and customer feedback, leading to continuous improvement.

Advantages and Disadvantages of Agile Methodology



❏ Topics to cover

- ❖ Stakeholders Theory
- ❖ System Analyst (Introduction, Roles & Responsibilities, Skills)
- ❖ Project Manager Roles and Responsibility
- ❖ Managing the information system(IS) project

□ Stakeholders Theory

- Stakeholders are individuals, groups, or entities that have an interest, influence, or investment in a project, organization, or system.
- Stakeholders have varying levels of interest and involvement, and their perspectives, needs, and expectations should be considered when making decisions or implementing initiatives.
- Stakeholders associated with a system can vary depending on the nature of the system, the organization, and the project. However, here is a general list of stakeholders that are commonly associated with systems:

1. Internal Stakeholders:

- Employees at various levels *within an organization*, including:
 - ❖ System Analysts: A systems analyst is a employee who analyzes, maintains, improves and designs systems for an organization to solve business problems.
 - ❖ Business Managers: Individuals responsible for the overall business strategy, decision-making, and ensuring that the system aligns with organizational goals.
 - ❖ Project Managers: Individuals responsible for planning, organizing, and managing the project's timelines, budgets, resources requirement(including human resource, raw materials etc).

- ❖ **Developers/Programmers:** Individuals responsible for writing code and implementing the system according to the specifications provided.
- ❖ **Designers:** Individuals who create the user interface (UI) and user experience (UX) design of the system, ensuring it is user-friendly and visually appealing.
- ❖ **Quality Assurance/Testers:** Individuals who test the system to identify and report any defects, ensuring that it functions correctly and meets the specified requirements.
- ❖ **Database Administrators (DBAs):** Professionals responsible for designing, maintaining, and securing the system's database infrastructure.
- ❖ **Network Administrators:** Individuals responsible for managing the network infrastructure that supports the system's communication and connectivity.
- ❖ **Security Specialists:** Individuals who ensure the system's security by identifying vulnerabilities and implementing measures to protect against data breaches and cyberattacks.

- ❖ **Procurement Team:** A group of professionals, dedicated to finding and acquiring goods and services(including hardware, software requirement) for an organization at the best possible price.
- ❖ **Training Specialists:** Individuals who design and deliver training programs for end-users to effectively use the system.
- ❖ **Support and Helpdesk Staff:** Individuals who provide technical support, troubleshooting, and assistance to end-users in case they encounter issues with the system.

2. External Stakeholders:

- ❖ **Customers/Clients:** Customers or clients are important stakeholders who use the system's services or products.
- ❖ **Vendors/Suppliers:** vendors or suppliers provide products, services, or integrations that contribute to the system's functionality.
- ❖ **Partners and Collaborators:**
- ❖ **Competitors:** Rival organizations or entities that might be affected by the organization's activities or decisions.
- ❖ **Media and Public Relations:** Media outlets(Broadcasting Channels), journalists, and public relations professionals who shape public perception and opinion.
- ❖ **Advocacy Groups:** It may be Non-governmental organizations (NGOs), community groups, or associations that advocate for specific causes or interests.

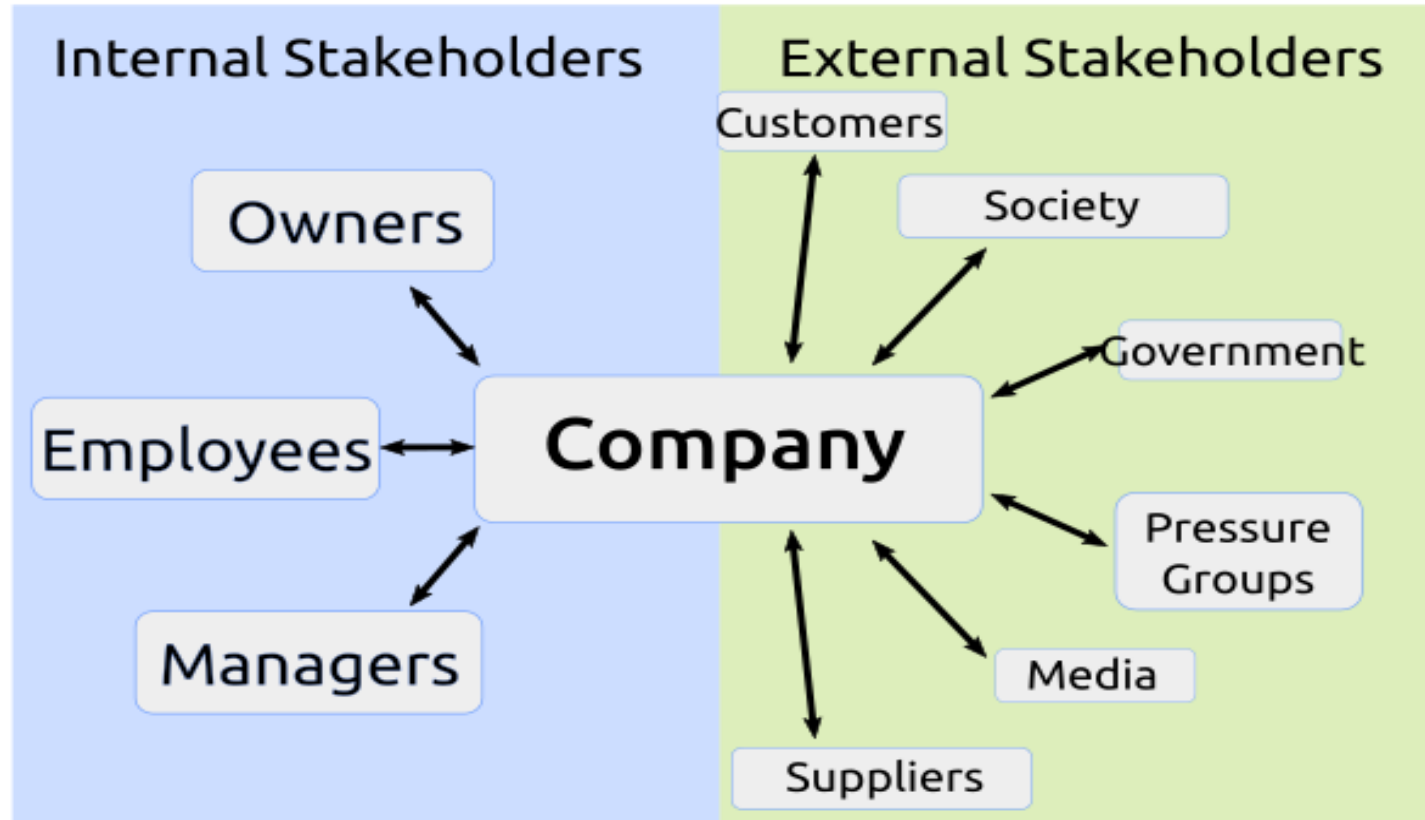
❖ **Investors/Shareholders:**

Individuals or groups who have a financial interest in the organization's success.

or

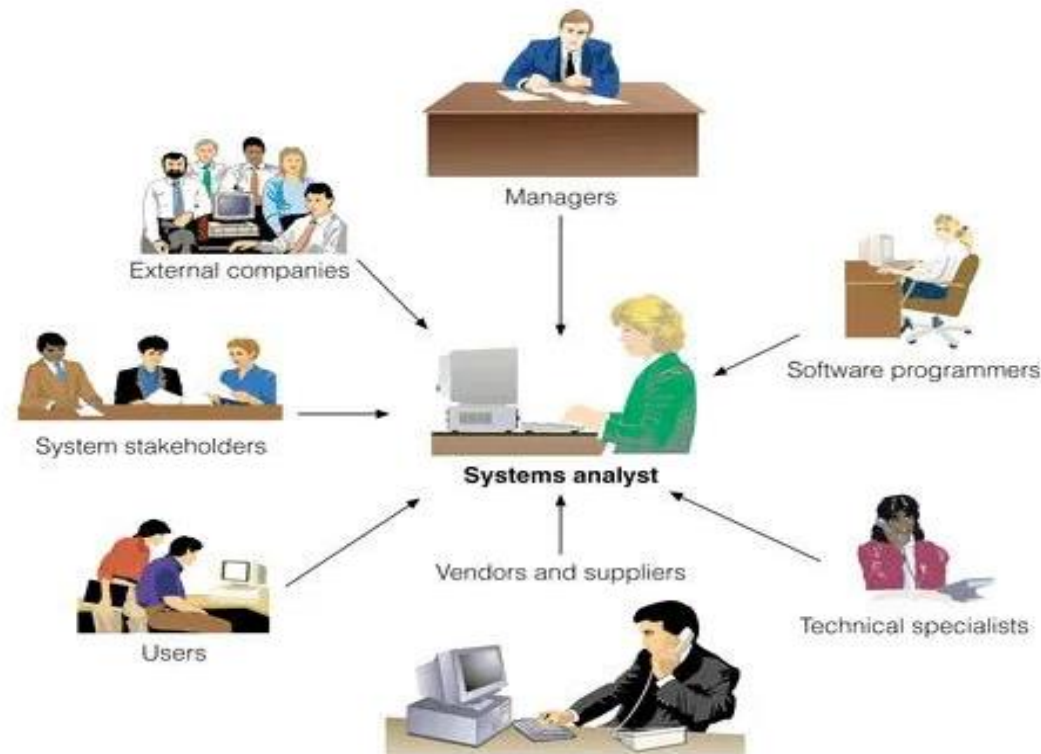
Individuals, groups, or institutions that provide financial support, loans, or investments to the organization.

Stakeholder Theory



❑ System Analyst (Introduction, Roles & Responsibilities, Skills)

- A systems analyst is a **employee** who analyzes, maintains, improves and designs information systems for an organization to solve business problems using information technology.
- A systems analyst is an IT professional who works on organization to ensure that systems, infrastructures are functioning as effectively and efficiently as possible.



❑ Roles & Responsibilities:

- ❖ Work closely with stakeholders to understand their needs and objectives.
- ❖ Analyze system requirements, both functional and non-functional.

❑ Functional Requirements:

- Functional requirements can be most simply defined as: *Something the system must do*. If the system does not meet a functional requirement it will fail.

Examples of Functional Requirements:(In the Context of E-commerce Site)

- **User Registration:**
 - Users must be able to create accounts with a unique username and password.
 - The system must send a confirmation email upon successful registration.
- **Order Processing:**
 - Users should be able to add products to their cart, view the cart, and complete the checkout process.
 - The system must calculate the total cost, apply discounts, and update inventory levels.
- **Search Functionality:**
 - The system must allow users to search for products using keywords.
 - Search results should be displayed in a relevant and organized manner.

❖ User Permissions:

- Admin should have access to additional features like *user management and content editing*.

❖ Reporting and Analytics:

- The system must generate sales reports for a specified date range.
- Reports should be downloadable in various formats, such as PDF or Excel.

❖ Non-Functional Requirements:

- Non-functional requirements specify the qualities, constraints, and characteristics that define how the system operates, performs, and behaves beyond its core functionalities. Or in simple words it is a *Requirements that describe how the system works*.
- These requirements focus on aspects such as performance, security, usability, and reliability.
- **Examples of Non-Functional Requirements:**

1. **Performance:**

1. The system should handle a minimum of 1000 concurrent users without experiencing performance degradation.
2. Page load times should not exceed 3 seconds.

2. **Security:**

1. User passwords must be stored securely using encryption.
2. The system must have access control methods to restrict unauthorized access to sensitive data.

- ❖ **Usability:** Usability refers to how easy and efficient a product is to use, allowing users to achieve their tasks effectively while providing a satisfying and enjoyable experience.
- ❖ **Reliability:** A reliable product is one that can be trusted to work consistently and deliver accurate results, ensuring minimal downtime, disruptions, or unexpected behaviors for users.
- ❖ **Scalability:** Scalability refers to a product or system's ability to handle increased userloads, user demands, or data volume in the future.
 - New servers should be added without disrupting service.
- ❖ **Compatibility:** *The system should be compatible with major web browsers (Chrome, Firefox, Safari, Edge) and devices (desktop, tablet, mobile).*

❖ **Maintainability:** Maintainability refers to how easily a product or system can be updated, modified, repaired, or enhanced over its lifecycle. A maintainable product is designed and developed in a way that allows developers to make changes efficiently, fix issues, and introduce new features without causing undue complications or disruptions to the existing functionality.

Note: Remember that both functional and non-functional requirements are essential for delivering a successful system that meets user needs while satisfying performance, security, and other critical aspects. Properly defining and prioritizing these requirements ensures that the system's development aligns with stakeholder expectations.

❖ **System Design:**

- Design the system architecture, data flow, and user interfaces to meet the requirements.
- Consider factors like usability, scalability, and security in the design.

❖ **Feasibility Analysis:**

- Assess the technical, economic, operational, and organizational feasibility of proposed systems.
- Provide recommendations based on the analysis to aid decision-making.

❖ **Collaboration and Communication:**

- Act as a bridge between business stakeholders and the development team.
- Facilitate clear communication and understanding between both parties.
- Ensure that the technical team understands the business requirements accurately.

❖ **Risk Management:**

- Identify potential risks and challenges associated with the project.
- Develop risk mitigation strategies and contingency plans.

❖ **Testing and Quality Assurance:**

- Collaborate with the testing team to ensure thorough testing and bug resolution.

❖ **Documentation:**

- Create comprehensive documentation including requirement specifications, design documents, and user manuals.

❖ User Training and Support:

- Provide training to end-users on how to use the system effectively.
- Offer ongoing support and troubleshooting as needed.

❖ Change Management:

- *It refers to the systematic process of planning, implementing, and controlling changes to a project or system. System analysts play a critical role in ensuring that changes to project requirements, scope, or functionalities are managed effectively to minimize disruptions and maintain project success.*

☐ Skills:

❖ **Technical Proficiency:**

- *Understanding of software development processes, methodologies, and programming languages.*
- *Familiarity with database management systems, networking concepts, and system architecture.*

❖ **Analytical and Problem-Solving Skills:**

- *Ability to analyze complex business problems and translate them into technical solutions.*
- *Identifying root causes of issues and proposing effective solutions.*

❖ **Communication Skills:**

- *Strong written and verbal communication skills to interact with technical and non-technical stakeholders.*
- *Ability to convey technical concepts to non-technical individuals and vice versa.*

- **Interpersonal Skills:**
 - *Effective collaboration and teamwork with developers, designers, business users, and other stakeholders.*
- **Critical Thinking:**
 - *Capacity to evaluate pros and cons of different solution options and make informed decisions.*
- **Time Management:**
 - *Ability to manage multiple tasks and priorities within project timelines.*
- **Domain Knowledge:**
 - *Understanding of the specific industry or domain in which the system will be implemented or launched.*
- **Adaptability:**
 - *Flexibility to handle changing requirements and project dynamics.*
- **Quality Focus:**
 - *Commitment to delivering a high-quality product that meets user needs.*
- **Overall, a system analyst serves as a critical link between the business and technical aspects of a project, ensuring that the final information system aligns with organizational goals and requirements.**

❖ Project Manager

- *A project manager is responsible for planning, executing, monitoring, and closing projects while ensuring that they are completed on time, within scope, and within budget. Their role involves coordinating and leading teams, managing resources, and ensuring effective communication among stakeholders. Here are the key roles and responsibilities of a project manager:*

❖ **Project Planning:**

- *Define project goals, scope, outlines tasks, timelines, resource requirements and deliverables in collaboration with stakeholders and Identify potential risks and develop strategies to mitigate them.*

❖ **Team Leadership:**

- *Assemble project teams, assign roles and responsibilities, and ensure team members understand their tasks.*
- *Motivate and inspire team members to achieve project objectives.*
- *Develop a collaborative and positive team environment.*

❖ **Resource Management:**

- *Allocate resources effectively to tasks based on skillsets, availability, and project needs.*
- *Manage resource conflicts and ensure teams have the necessary tools and support.*

- **Communication Management:**

- *Maintain clear and consistent communication with stakeholders, team members, and sponsors.*
- *Facilitate regular status meetings, update reports, and progress reviews.*

- **Risk Management:**

- *Continuously monitor project risks and proactively address potential issues.*
- *Develop risk mitigation plans and execute them as needed.*

- **Budget Management:**

- *Develop and manage project budgets, tracking expenses and financial constraints.*
- *Approve expenditures and allocate resources efficiently.*

- **Scope Management:**

- *Define project scope and ensure that changes are documented, assessed, and approved through proper channels.*
- *Prevent scope creep(uncontrolled growth/change) and manage any changes effectively.*

- **Quality Assurance:**

- *Ensure that project deliverables meet the required quality standards and align with stakeholder expectations.*
- *Implement quality control processes and conduct regular reviews.*

- **Time Management:**

- *Monitor project timelines and milestones, making adjustments as necessary to stay on track.*
- *Address delays promptly and communicate revised schedules.*

- **Stakeholder Management:**

- *Identify project stakeholders and their interests, needs, and concerns.*
- *Engage stakeholders through effective communication and manage their expectations.*

❏ **Project Manager Skills:**

❖ Leadership Skills:

- *Ability to guide and inspire teams, set direction, and make critical decisions.*

❖ Communication Skills:

- *Strong verbal and written communication to effectively convey information to diverse audiences.*

❖ Problem-Solving Abilities:

- *Skill in identifying issues, analyzing root causes, and developing effective solutions.*

❖ Time and Resource Management:

- *Capacity to manage project timelines, prioritize tasks, and allocate resources efficiently.*

❖ Negotiation and Conflict Resolution:

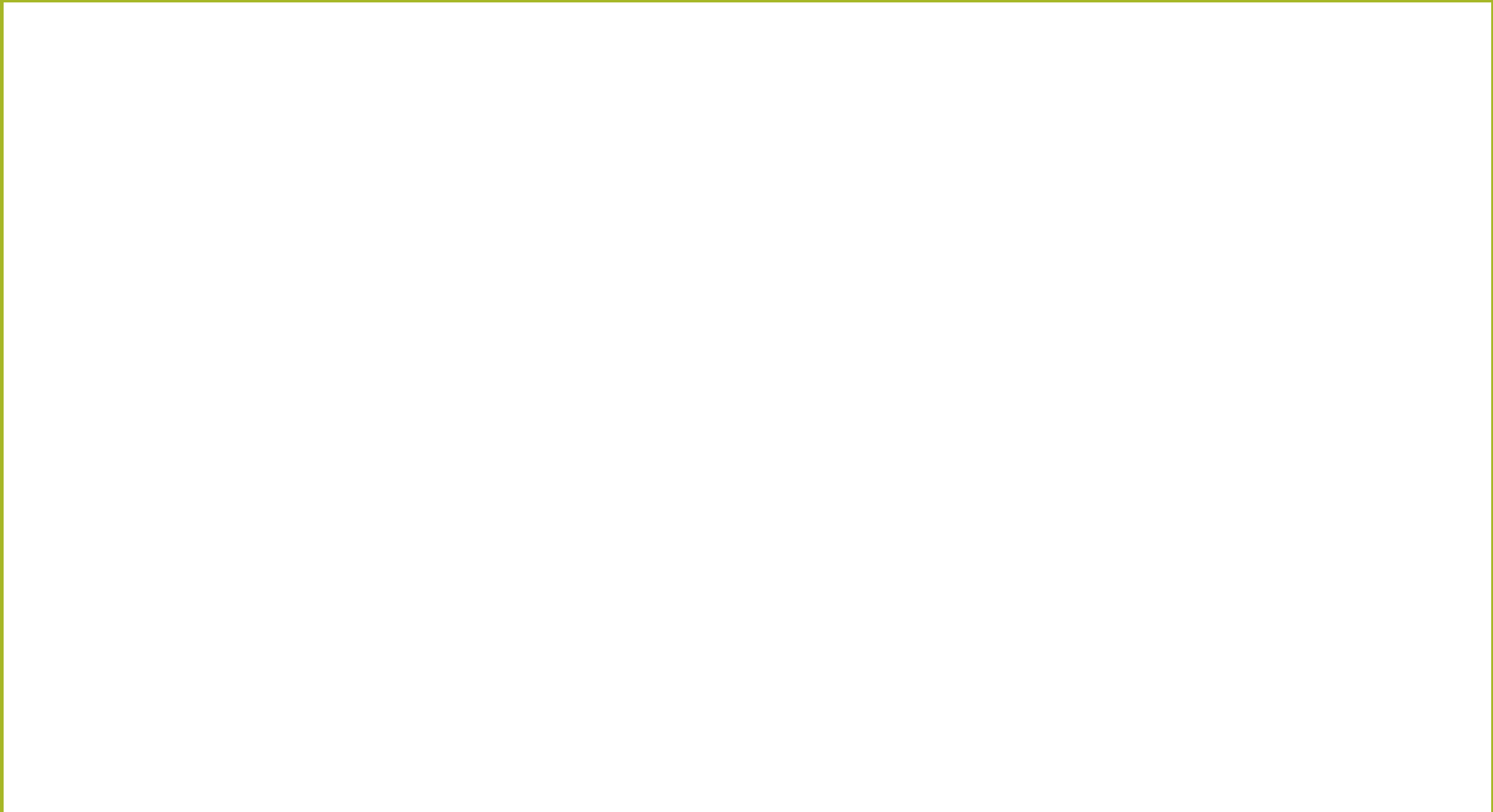
- *Proficiency in resolving conflicts and negotiating with stakeholders to achieve goal.*

- **Risk Management:**
 - *Ability to identify, assess, and mitigate project risks effectively.*
- **Budgeting and Financial Management:**
 - *Knowledge of budgeting principles and practices to manage project finances.*
- **Adaptability and Flexibility:**
 - *Readiness to adjust plans and strategies based on changing project circumstances.*
- **Technical Understanding:**
 - *Familiarity with the technical aspects of the project domain or industry.*
- **Interpersonal Skills:**
 - *Strong relationship-building skills to foster collaboration and trust.*
- **Analytical Thinking:**
 - *Capacity to analyze complex situations and make data-driven decisions.*

- **Organizational Skills:**
 - *Ability to organize tasks, documents, and resources in a structured manner.*
- **Customer Focus:**
 - *Dedication to meeting stakeholder needs and delivering value.*
- **Conflict Management:**
 - *Skill in resolving conflicts constructively and maintaining a positive team environment.*
- **Decision-Making:**
 - *Capacity to make informed decisions for the project's benefit.*
- **Influence and Persuasion:**
 - *Ability to influence stakeholders and gain support for project initiatives.*
- A successful project manager balances these roles, responsibilities, and skills to lead projects to successful completion while ensuring stakeholder satisfaction and project outcomes that align with organizational goals

Roles of a Project Manager





#Managing the information system project

- In this chapter, we focus on the role of a project manager of an information systems project throughout the SDLC.
- As the name suggests, project managers are in charge of projects from initiation to close, making sure the work gets done efficiently and satisfactorily.
- The duties of a project manager include managing resources, keeping the client requirements in check, coordinating with the team and making sure that the outcomes are delivered on time.
- Project management is arguably the most important aspect of an information systems development project because "Project Managers play the lead role in planning, executing, monitoring, controlling, and closing projects. They're expected to deliver a project on time, within the budget.
- Effective project management helps to ensure that systems development projects meet customer expectations and are delivered within budget and time constraints.

#What Are the Responsibilities of a Project Manager?

- The general project manager duties/responsibilities includes:

1. Plan and Develop the Project Idea

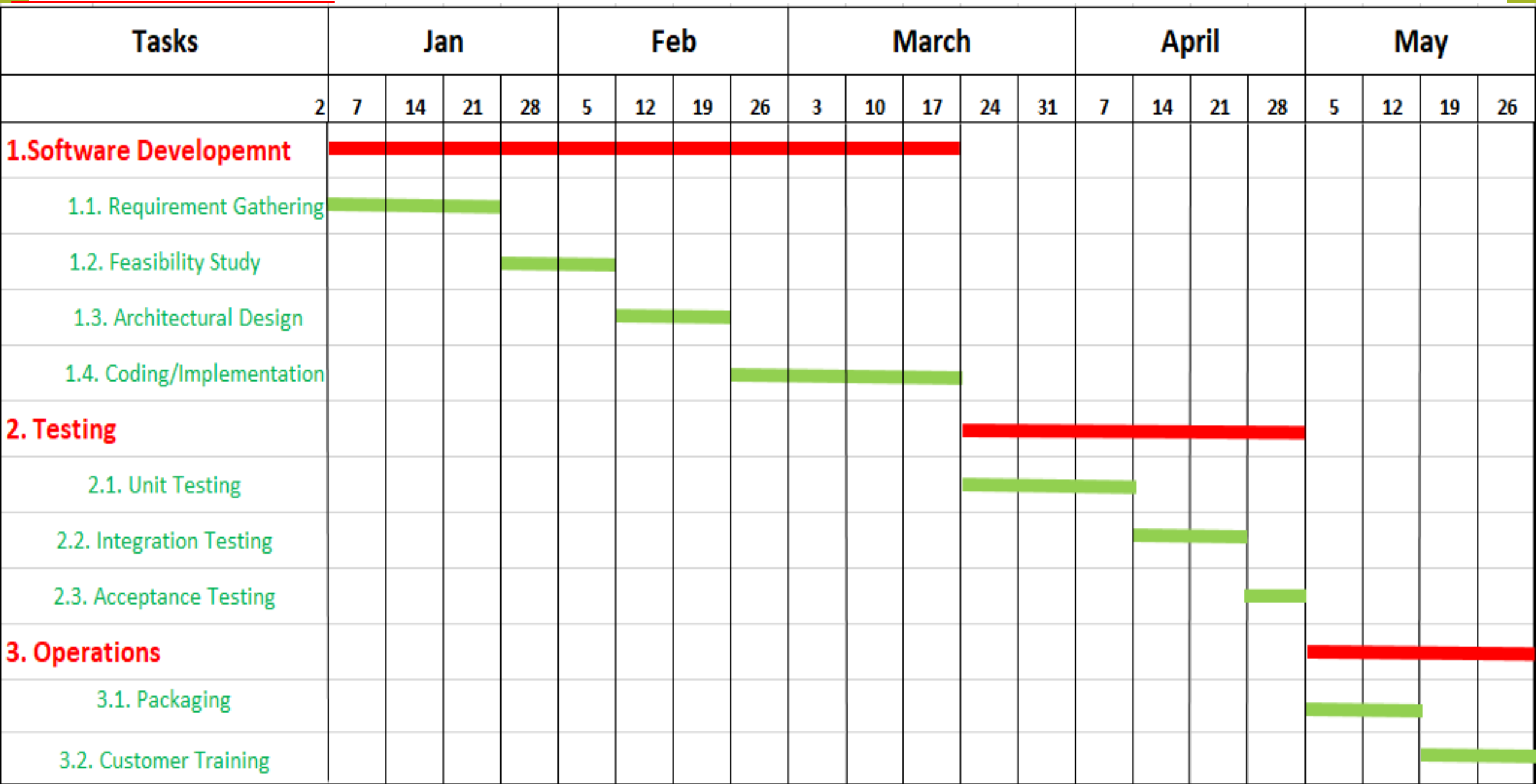
- Every project starts as an idea. It's a project manager's job to work with internal stakeholders and external clients to define that concept and create a process to bring it to fruition(success/completion).
- This includes setting and managing client expectations, developing a detailed project plan, defining the scope of the project and assigning team members to specific tasks.
- It often requires you to make numerous assumptions about the availability of resources such as hardware, software, and personnel and all related resources and manpower to achieve the goal.
- During this activity, you should reach agreement on the following questions:
 - What problem or opportunity does the project address(complexity of the project)?
 - What are the quantifiable results to be achieved?
 - What needs to be done to accomplish the project ?
 - How will success be measured?
 - How will we know when we are finished?

1.1. Dividing the project into manageable tasks.

- This is a critical activity during the project planning process. Here, you must divide the entire project into manageable tasks and then logically order them to ensure a smooth evolution between tasks.
- A work breakdown for these activities is represented in a Gantt chart.
- A Gantt(**Generalized Activity Normalization Time Table**) chart is a graphical representation of a project that shows each task as a horizontal bar whose length is proportional to its time for completion.

Note: Different colors, shades, or shapes can be used to highlight each kind of task.

Gantt Chart



1.2. Determining project standards and procedures

- During this activity, you will identify how various deliverables(output/product) are produced by using best possible SDLC Model and tested by you and your project's testing and quality assurance team.
- The team must decide
 - Which tools to use,
 - Which SDLC methods will be used,
 - Documentation styles (e.g., type fonts and margins for user manuals(The User Manual **contains all essential information for the user to make full use of the information system**)
 - How team members will report the status of their assigned activities, and terminology.
- Setting project standards and procedures for work acceptance is a way to ensure the development of a high-quality system. Also, it is much easier to train new team members when clear standards are in place.

2. Create and Lead Your Dream Team

Project managers are accountable for every aspect of the project, including leading a team capable of meeting or exceeding client expectations for their vision. Hence creating a capable team is the major concern to achieve the goals.

Successful project managers assemble and manage these individuals to make a fine-tuned project team.

If the team needs guidance, training or coaching, it's a project manager's responsibility to set them up for success.

In order to build and maintain a dynamic team mentality, a project manager must be able to keep open and honest communication, form working relationships and motivate anyone who needs it.

"There is no other way than leading by example. If you are doing your part correctly, always supporting your team, and having a fair and healthy approach with them, motivation should never be a problem."

— Dragan Hrgić, Remade

3. Monitor Project Progress and Set Deadlines

- Organization(maintaining the order and structure of the system) and follow-through are a big part of a project manager's job.
- From creating an accurate timeline of project completion to ensuring tasks are finished within the given deadlines of the assignment.
- The project manager must remain aware of how the project is progressing.
- The project manager also anticipates(predict) delays that may occur on the client side and apprises(inform) the team of any changes in the client's needs.

4. Solve Issues That Arise

- During every project, issues arise that need to be solved.
- Risks might arise from the use of new technology, prospective users resistance to change/frequent modification, availability of critical resources, team member inexperience with technology or the business idea.
- Hence, as a project manager you should continually try to identify and assess project risk.
- The project manager is also responsible to anticipate(predict) any potential hiccups(interruptions) before they happen.
- Adaptability and problem solving are key to keeping control of a project

→ Addressing the risks and solving the risks are the major concern of a project manager else it can seriously affect the timeline and budget.

"You have to go in expecting that things won't be as you had planned, and things won't be as easy as first expected. Goals, conditions, and circumstances will change."

— Kalila Lang

5. Manage the Capital(Money)

- Budget management is another primary project manager duty. These professionals make sure that the project gets done without excessive spending. A good project manager has mastered the art of cost efficiency.
- Project managers also must be transparent and realistic about the cost so clients are aware from the beginning how much they are likely to spend.

6. Ensuring customer satisfaction

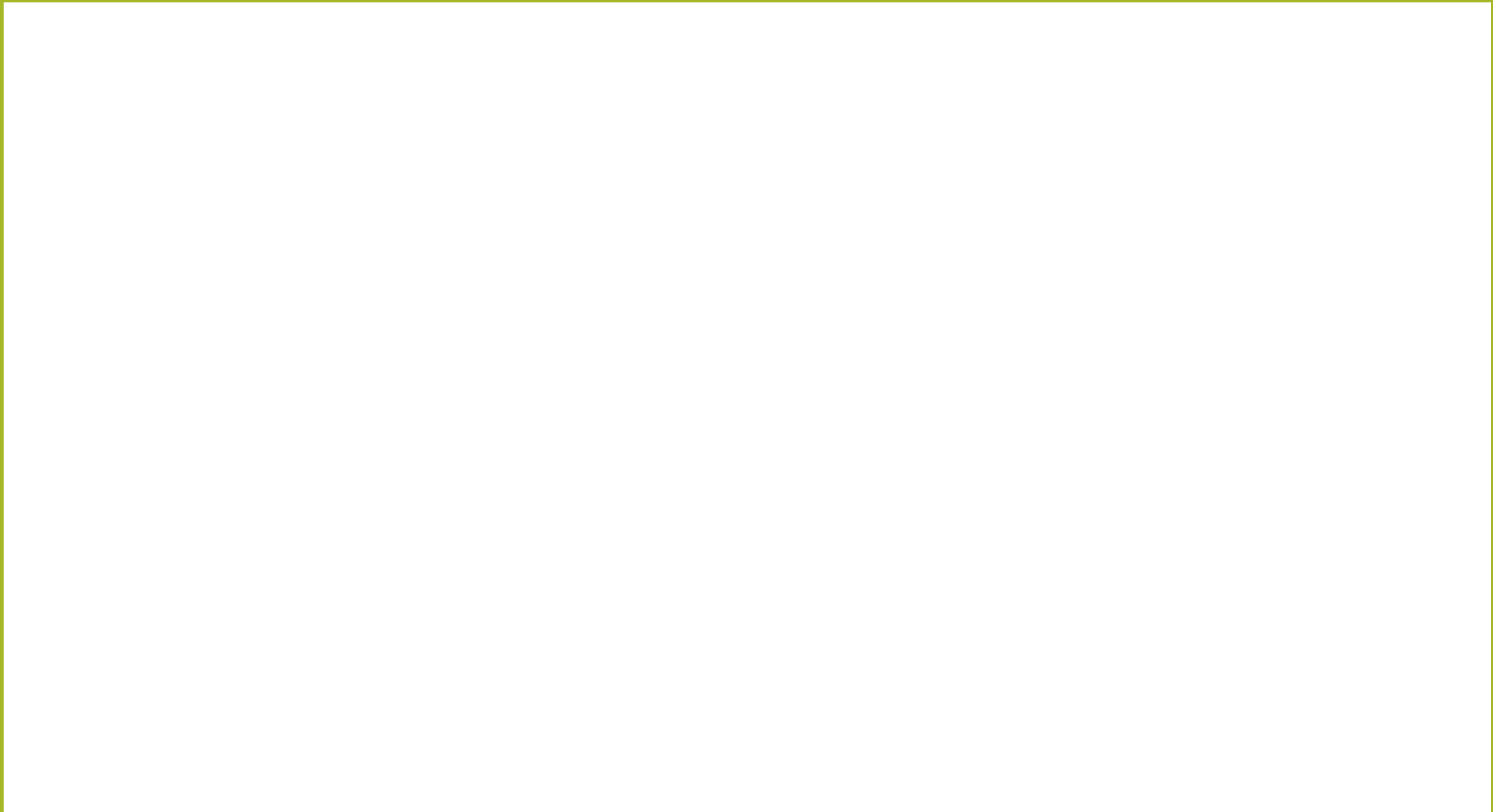
- In the end, a project is only a success if the customer is happy. One of the key responsibilities of every project manager is to minimize uncertainty, avoid any unwanted surprises, and involve their clients in the project as much as is reasonably possible. Good project managers know how to maintain effective communication and keep the company's clients up-to-date.

7.Ensuring Client communication and feedback

- Project managers have the closest relationship with clients or anyone who works on a project. Because of this, it is important that they keep open lines of communication for updates and feedback.
- The periodic updates and the feedback keeps the project on track and If any issues or changes arise in the timeline of a project, the project manager is in charge of keeping the stakeholders up to date.

8. Managing reports and necessary documentation

- Finally, experienced project managers know how essential final reports and proper documentation are. Good project managers can present comprehensive reports documenting that all project requirements were fulfilled, as well as the projects' history, including what was done, who was involved, and what could be done better in the future.
- After a project is finished, the project manager is responsible for evaluating its efficiency and effectiveness.
- With the data they've tracked throughout the process, they can begin to identify shortcomings(issues) and plan for ways to fix similar issues in the future.
- This is also an opportunity to highlight what went right, including building camaraderie(togetherness/closeness) and rewarding team members who excelled during the project.



❑ Managing the Information System Project

- Managing an information system project involves a series of steps and best practices to ensure that the project is completed successfully. Here is a general overview of how to manage an information system project:

1. Project Initiation:

a. Define the project's objectives and scope.

- ❖ Clearly state the specific goals, outcomes, or results that the project is intended to produce. These objectives should be **(SMART)**: specific, measurable, achievable, relevant, and time-bound.
- By defining the project's objectives and scope clearly and comprehensively, you create a roadmap for the project team to follow. This helps prevent misunderstandings, manage expectations, and ensure that the project stays on track to achieve its intended outcomes.

b. Identify key stakeholders and their roles.

- Stakeholders are individuals, groups, or entities that have an interest, influence, or investment in a project, organization, or system.
- There are two categories of stakeholder's. i. Internal and External.
- The actions and perspective of stakeholders influence the operations and decision making of an organizations.

1. Internal Stakeholders:

- Employees at various levels *within an organization*, including:
 - ❖ **System Analysts:** A systems analyst is a employee who analyzes, maintains, improves and designs systems for an organization to solve business problems.
 - ❖ **Business Managers:** Individuals responsible for the overall business strategy, decision-making, market trends and behavior, customers expectation and satisfaction etc.
 - ❖ **Project Managers:** Individuals responsible for everything required from beginning to completion of project i.e planning, organizing, and managing the project's timelines, budgets, resources requirement(including human resource, raw materials etc). Everything to complete the project.

- ❖ **Developers/Programmers:** Individuals responsible for writing code and implementing the system according to the *specifications provided*.
- ❖ **Designers:** Individuals who create the user interface (UI) and user experience (UX) design of the system, ensuring it is user-friendly and visually appealing.
- ❖ **Quality Assurance/Testers:** Individuals who test the system to identify and report any defects, ensuring that it functions correctly and meets the specified requirements.
- ❖ **Database Administrators (DBAs):** Professionals responsible for designing, maintaining, and securing the system's database infrastructure.
- ❖ **Network Administrators:** Individuals responsible *for managing the network infrastructure* that supports the system's communication and connectivity.
- ❖ **Security Specialists:** Individuals who ensure the system's security by identifying vulnerabilities and implementing measures to protect against data breaches and cyberattacks.

- ❖ **Procurement Team:** A group of professionals, dedicated to finding and acquiring goods and services(including hardware, software requirement) for an organization at the best possible price.
- ❖ **Training Specialists:** Individuals who design and deliver training programs for end-users to effectively use the system.
- ❖ **Support and Helpdesk Staff:** Individuals who provide technical support, troubleshooting, and assistance to end-users in case they encounter issues with the system.

2. External Stakeholders:

- ❖ **Customers/Clients:** Customers or clients are important stakeholders who use the system's services or products.
- ❖ **Vendors/Suppliers:** vendors or suppliers provide products, services, or integrations that contribute to the system's functionality.
- ❖ **Partners and Collaborators:**
- ❖ **Competitors:** Rival organizations or entities that might be affected by the organization's activities or decisions.
- ❖ **Media and Public Relations:** Media outlets(Broadcasting Channels), journalists, and public relations professionals who shape public perception and opinion.
- ❖ **Advocacy Groups:** It may be Non-governmental organizations (NGOs), community groups, or associations that advocate for specific causes or interests.

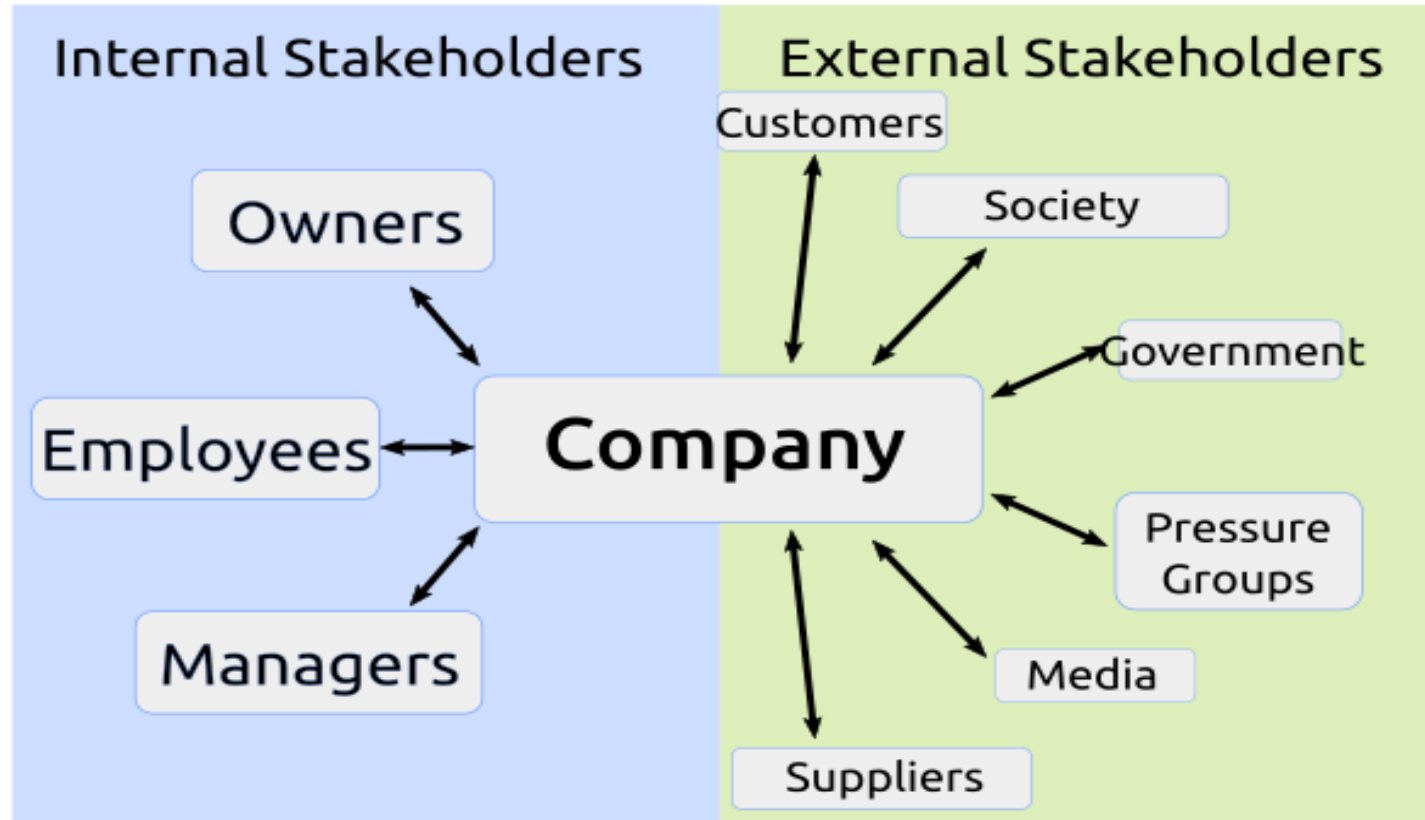
❖ **Investors/Shareholders:**

- **Individuals or groups who have a financial interest in the organization's success.**

or

Individuals, groups, or institutions that provide financial support, loans, or investments to the organization.

Stakeholder Theory



c. Establish a project team with the necessary skills and expertise.

- By establishing a project team with the necessary skills and expertise, you ensure that the project has the talent and capabilities required to meet its objectives.
- A well-rounded and skilled team is better equipped to handle challenges, make informed decisions, and deliver high-quality results.
- In order to build and maintain a dynamic team mentality, a project manager must be able to keep open and honest communication, form working relationships and motivate anyone who needs it.

d. Develop a project charter that outlines the project's purpose, goals, and constraints.

- The project charter serves as a *foundational document that provides direction* and context for the *project*, ensuring that everyone involved understands *its purpose, objectives, and the boundaries* within which *it must operate*. It is a critical reference point throughout the project's lifecycle.

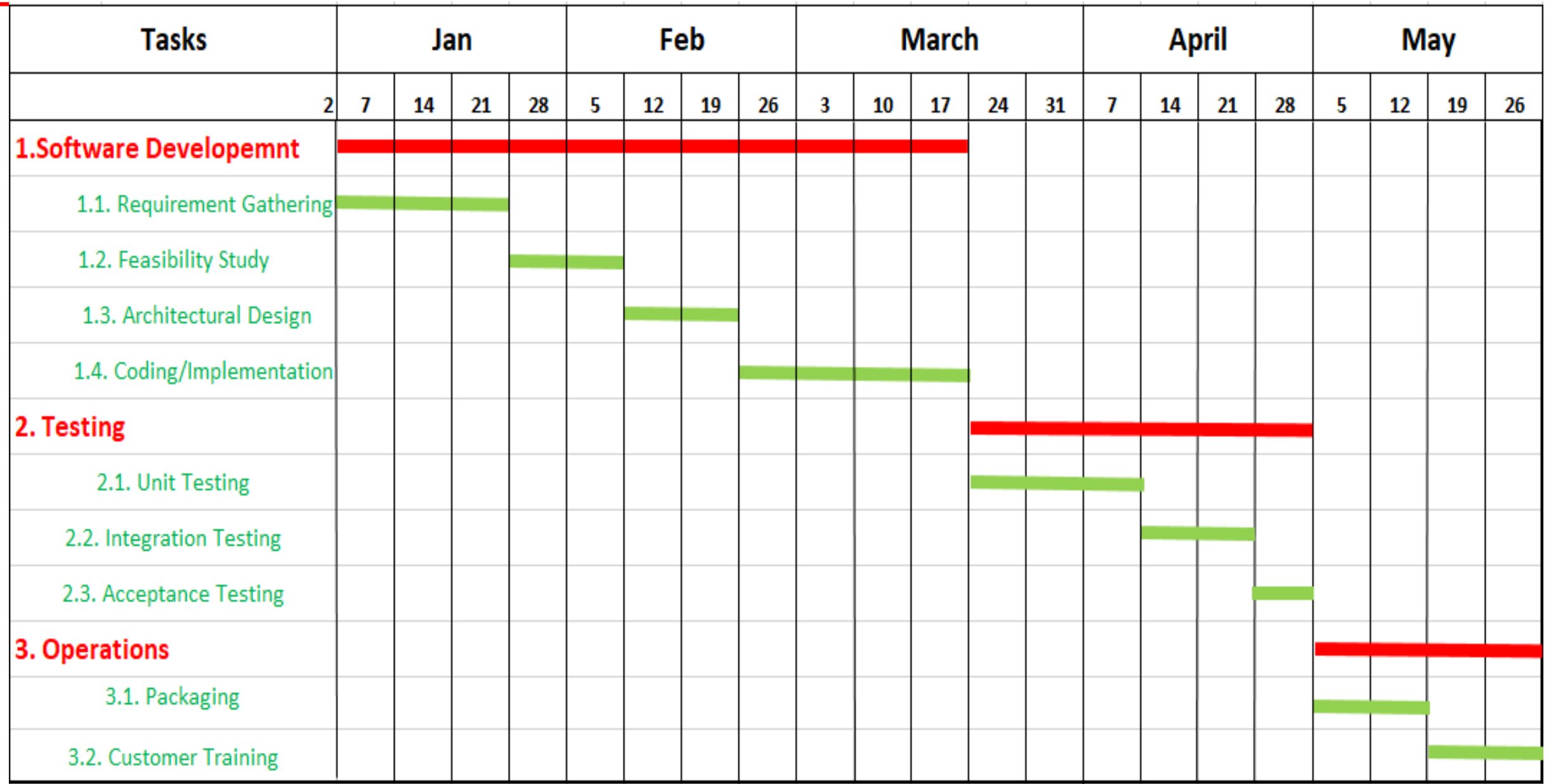
2. Planning:

a. Create a detailed project plan that includes tasks, timelines, and dependencies.

- This is a critical activity during the project planning process. Here, you **must divide the entire project into manageable tasks** and then logically order them to ensure a smooth evolution between tasks.
- A work breakdown for these activities is represented in a Gantt chart.
- A Gantt(**Generalized Activity Normalization Time Table**) chart is a graphical representation of a project that shows each task as a horizontal bar whose length is proportional to its time for completion.

Note: Different colors, shades, or shapes can be used to highlight each kind of task.

Gantt Chart



b. Allocate resources and budget for the project.

- Budget management is another primary project manager duty. *Project manager make sure that the project gets done without excessive spending.*
- A good project manager has mastered the art of cost efficiency.

c. Develop a risk management plan to identify and mitigate potential issues.

☐ *Identify Risks:*

- Begin by systematically identifying all potential risks that could affect the project. Risks can be anything that might hinder the project's progress, such as technical challenges, resource constraints, changing requirements, or external factors like market shifts or regulatory changes.

☐ *Assess Risks:*

- Once identified, assess the risks to determine their likelihood and potential impact on the project.

❑ **Prioritize Risks:**

- Prioritize risks based on their significance. High-priority risks are those with the highest potential impact on the project's success.
- Example: A risk that could lead to a significant project delay or budget overrun might be considered high-priority.

❑ **Mitigation Strategies:**

- Develop mitigation strategies for each high-priority risk. These strategies are proactive measures to reduce the likelihood or impact of a risk or to have a plan in place to address it if it occurs.

C. Identify and engage with vendors or suppliers if needed.

- Engaging vendors can provide valuable support but requires effective vendor management to ensure alignment with project goals and quality standards.

3. Requirements Gathering:

- Requirement gathering is a critical phase in managing an information system project. It involves the process of collecting, documenting, and understanding the needs, expectations, and specifications of the project's stakeholders.
- Effective requirement gathering is essential in ensuring that the resulting information system meets the needs of the stakeholders, aligns with the project's objectives, and minimizes the risk

4. System Design:

- Create a system architecture and design based on the gathered requirements.
- Develop data models, user interface designs, and system flow diagrams.
- Review the design with stakeholders and make necessary adjustments.

5. Development and Implementation:

- Build the system based on the approved design.
- Conduct thorough testing, including *unit testing, integration testing, and user acceptance testing*.
- Train end-users and provide documentation.
- Deploy the system in a controlled manner.

6. Monitoring and Control:

- Monitor project progress against the plan.
- Address issues and risks as they arise.
- Ensure that quality standards are maintained.
- Communicate regularly with stakeholders about project status.

7. Quality Assurance:

- It is the **quality control processes** to ensure that the system meets the defined standards and requirements.
- It involves setting standards, implementing processes, and conducting **continuous improvement** activities to prevent defects and **maintain consistent** quality throughout a project or production cycle.
- Conduct regular reviews and testing to identify and address defects or issues.
- ❑ In summary, Quality Assurance is a systematic approach to ensuring that products or services meet defined quality standards and customer expectations.

8. Managing reports and necessary documentation

- Finally, experienced project managers know how essential final reports and proper documentation are.
- Good project managers can present comprehensive reports documenting that all project requirements were fulfilled, as well as the projects' history, including what was done, who was involved, and what could be done better in the future.
- After a project is finished, the project manager is responsible for evaluating its efficiency and effectiveness as well.

Note: This is also an opportunity to highlight what went right, including building camaraderie(togetherness/closeness) and rewarding team members who excelled during the project.

9. Post-Implementation Review:

- It includes:
- Monitor the system's performance and gather feedback from users.
- Address any additional issues or enhancements that arise after deployment.

10. Ongoing Maintenance and Support:

- Establish a maintenance plan and support structure to ensure the long-term health of the system.
- Continuously evaluate and update the system as needed to meet changing requirements and technology advancements.
- Throughout the project, effective communication, risk management, and stakeholder engagement are critical for success. Regularly assess progress and make adjustments as necessary to keep the project on track and aligned with its goals.

❑ Identifying and selecting system development project in system analysis and design?

- Identifying and selecting a system development project in system analysis and design is a critical step that requires careful consideration and planning. Here are some steps to help you through this process:

1. Understand the Business Needs:

- Begin by understanding the business or organizational needs. Meet with stakeholders, such as business owners, managers, and end-users, to identify their pain points and requirements. What problems need solving? What goals should the new system achieve?

2. Feasibility Study:

- Conduct a feasibility study to assess whether the proposed project is viable. Consider technical, operational, economic, legal, and scheduling aspects. This study will help you determine if the project is worth pursuing.

3. Project Selection Criteria:

Define clear criteria for selecting projects. These criteria might include strategic alignment, return on investment (ROI), technical feasibility, regulatory compliance, and market demand. Assign weights to each criterion based on their importance

4. Generate Project Ideas:

- Brainstorm and gather project ideas from stakeholders and team members. Encourage creativity and consider both short-term and long-term goals.

5. Evaluate Project Ideas:

- Use the established criteria to evaluate each project idea. Score them based on how well they meet the criteria. Eliminate projects that don't meet the minimum threshold.

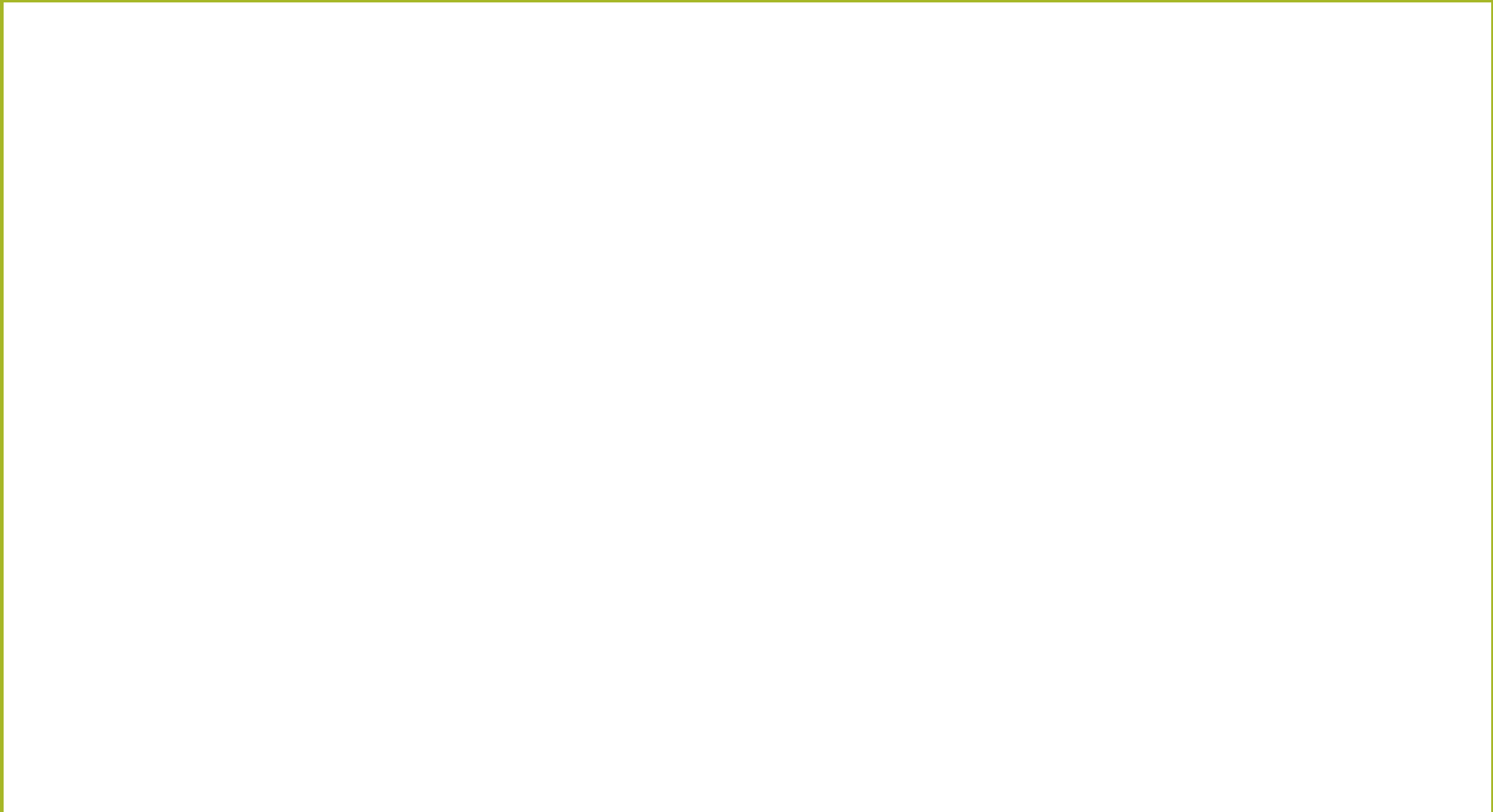
6. Cost-Benefit Analysis:

- Conduct a cost-benefit analysis for the remaining project ideas. Estimate the costs associated with development, implementation, and maintenance and compare them to the expected benefits.

7. Risk Assessment:

- Identify potential risks associated with each project idea, such as technical challenges, resource constraints, and market uncertainties. Assess the impact and likelihood of these risks.

- **Prioritization and Selection:**
 - Rank the project ideas based on their evaluation scores, cost-benefit analysis, and risk assessments. Select the project that aligns best with the organization's strategic goals and has the highest overall score.
- **Create a Project Charter:**
 - Once a project is selected, create a project charter. This document outlines the project's scope, objectives, stakeholders, budget, timeline, and key deliverables. It serves as a roadmap for the project.
- **Engage Stakeholders:**
 - Involve key stakeholders in the decision-making process and keep them informed throughout the project's lifecycle. Their input and support are crucial for project success.
- **Obtain Approval:**
 - Seek approval from senior management or relevant decision-makers to proceed with the selected project. Present the project charter and the rationale behind the selection.
- **Allocate Resources:**
 - Allocate the necessary resources, including personnel, budget, and technology, to kickstart the project.
- **Project Planning:**
 - Begin detailed project planning, including defining requirements, designing the system, and creating a project schedule.
- **Iterative Development:**
 - Implement an iterative development process, such as Agile or Scrum, to continuously gather feedback and make necessary adjustments throughout the project's lifecycle.
- **Monitoring and Control:**
 - Continuously monitor the project's progress, budget, and risks. Implement control measures to address any deviations from the plan.
- **Testing and Deployment:**
 - Conduct thorough testing to ensure the system meets the specified requirements. Once testing is successful, deploy the system into production.
- **Post-Implementation Review:**
 - After deployment, conduct a post-implementation review to evaluate the system's performance and its impact on the organization. Gather feedback from end-users and make any necessary improvements.
- Remember that selecting the right system development project is crucial for achieving business objectives and avoiding wasted resources. It's essential to involve stakeholders, conduct thorough analysis, and follow a structured decision-making process.



UNIT2:Modeling Tools for Systems Analyst

- System Analyst (Introduction, Roles, and Skills)
- Data Flow Diagram (Level 0, Level 1, and Level 2)
- E-R Diagram
- Context Diagram
- CASE tools

DFD(Data flow diagram)

- ❖ DFD stands for “Data Flow Diagram”. It is also known as “Bubble chart” through which we can represent the flow of data graphically in an information system.
 - ❖ By using DFD we can easily understand the overall functionality of a system because diagram represents the incoming data flow ,outgoing data flow and stored data in a graphical form.
 - ❖ It describes how data is processed in a system in terms of input and produce output.
 - ❖ A DFD model uses different notations or symbol to represent flow of data.
1. **External Entity** : Entities are source and destination of information data. Entities are represented by a rectangles with their respective names.

1.Entities

Examples of entities:

- **Person:** Employee, Student, Patient, customer
- **Place:** Class room , Building
- **Object:** Machine, product, and Car
- **Event:** Sale, Registration, Renewal
- **Concept:** Account, Course

2. Process or Bubble - Activities and action taken on the data are represented by Circle or Round-edged rectangles.

3. Data Storage - There are two variants of data storage - it can either be represented as a rectangle with absence of both smaller sides or as an open-sided rectangle with only one side missing.

4. Data Flow - Movement of data is shown by pointed arrows. Data movement is shown from the base of arrow as its source towards head of the arrow as destination.



❑ RULES OF DFD:

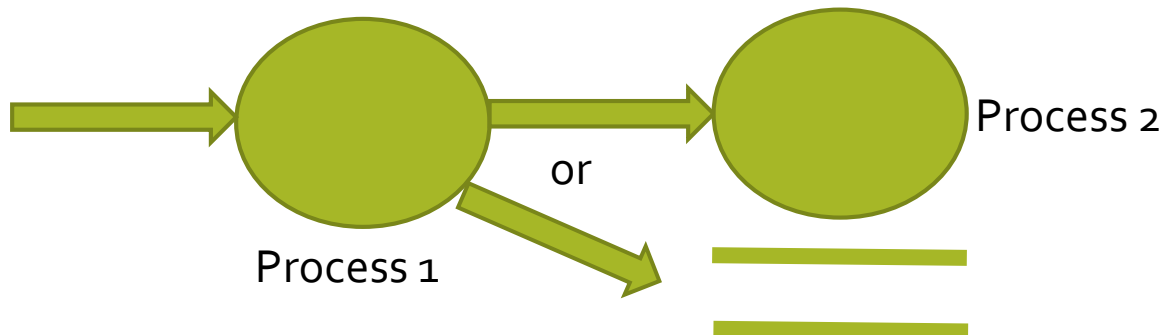
1. Each process should have at least one input and one output.



2. Each data store should have at least *one data flow in* and *one data flow out*.

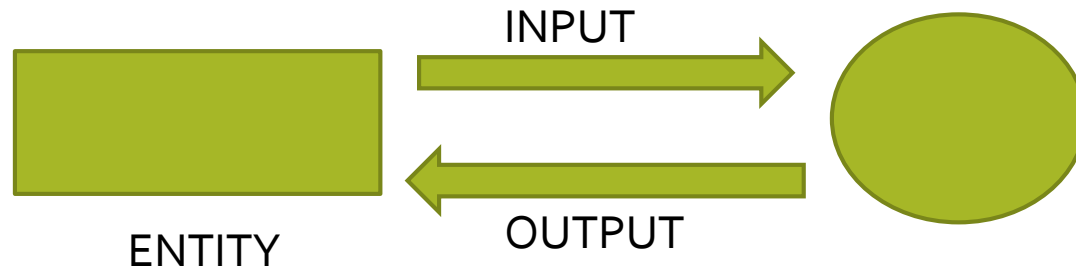


3. All process in a DFD go to either another process or data store.



❑ RULES OF DFD:

4. All the external entities must be connected through a process and entity *can be act as a INPUT* to the Process as well as *can receive information or desire output after processing*.

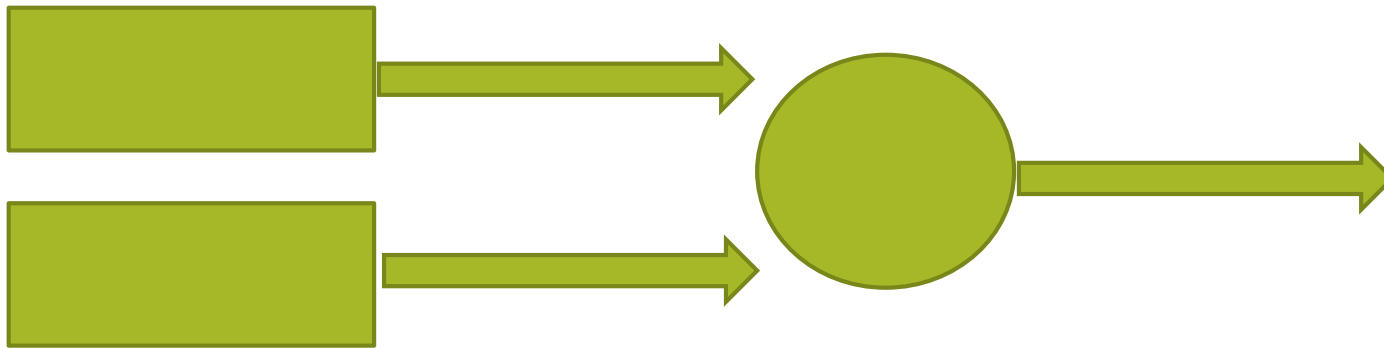


❑ LEVELS DFD(Data flow diagram)

1.LEVEL 0 DFD:

→It is a diagram which provides the entire systems data flows and processing with a single process (bubble) is called as *context level DFD*.

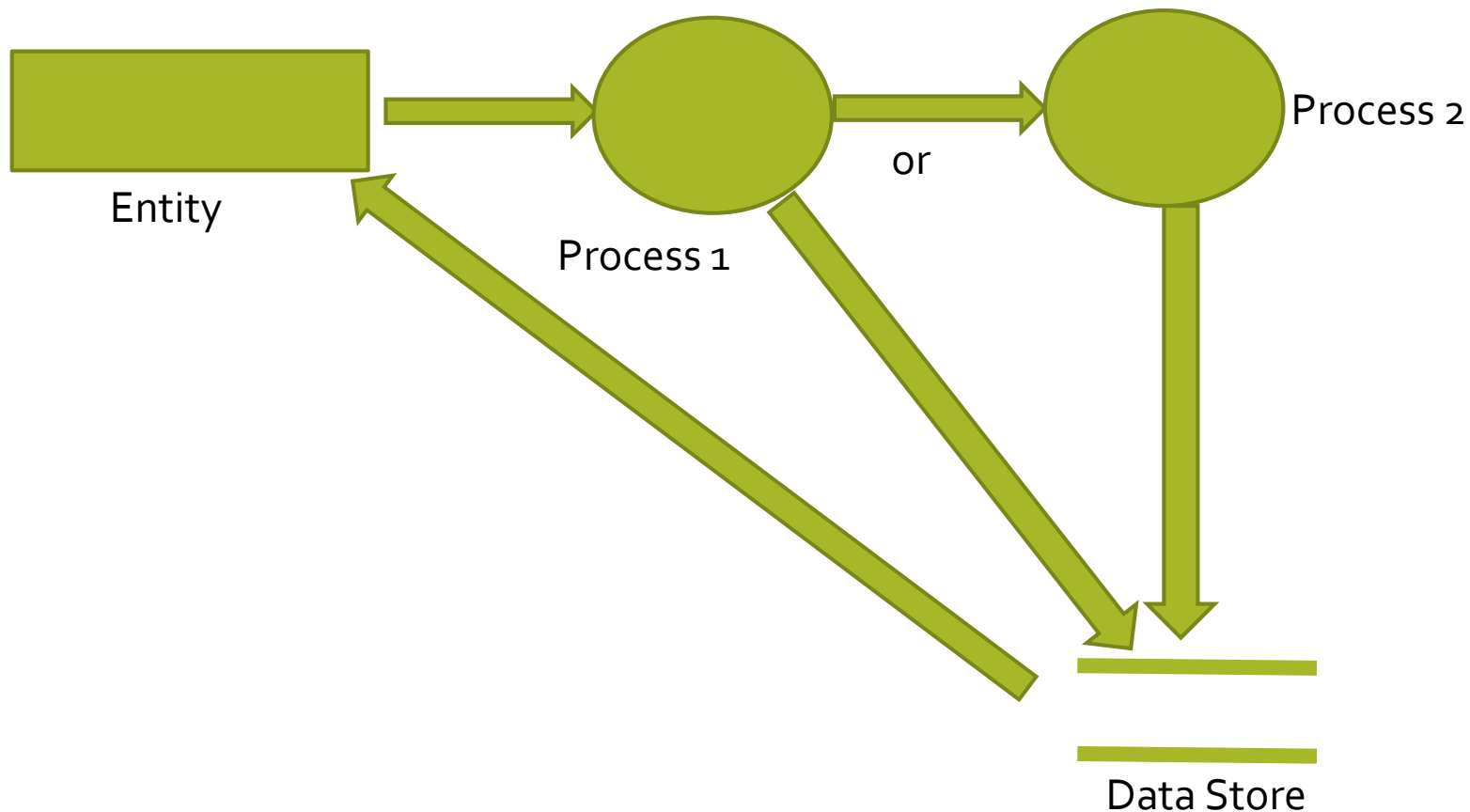
Note: In context level DFD we do not show any data store. Data stores are added at level 1.



LEVELS DFD(Data flow diagram)

2 . LEVEL 1 DFD:

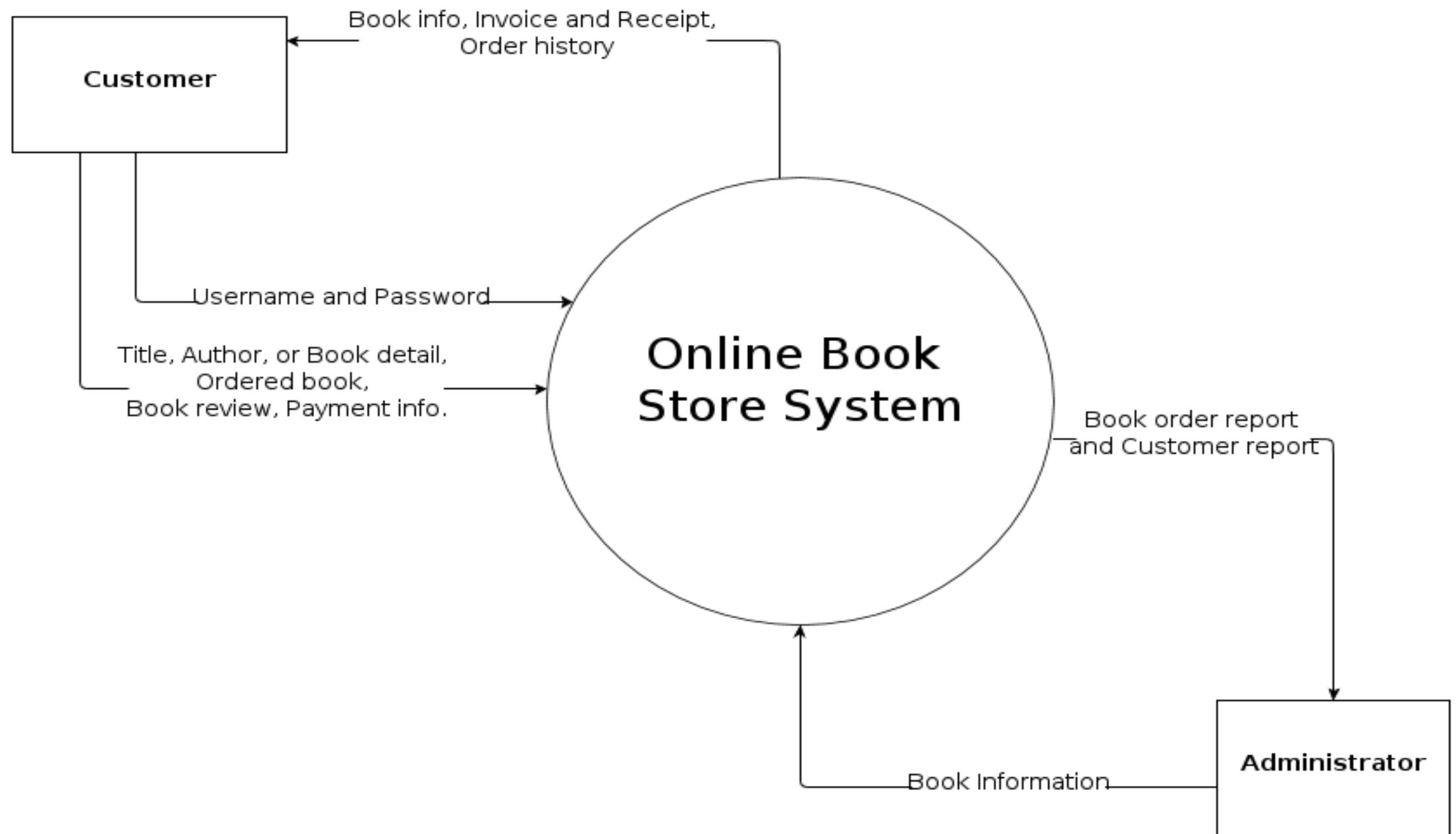
→ This is the *more detailed version of the previous level DFD* that includes the database and various important units.



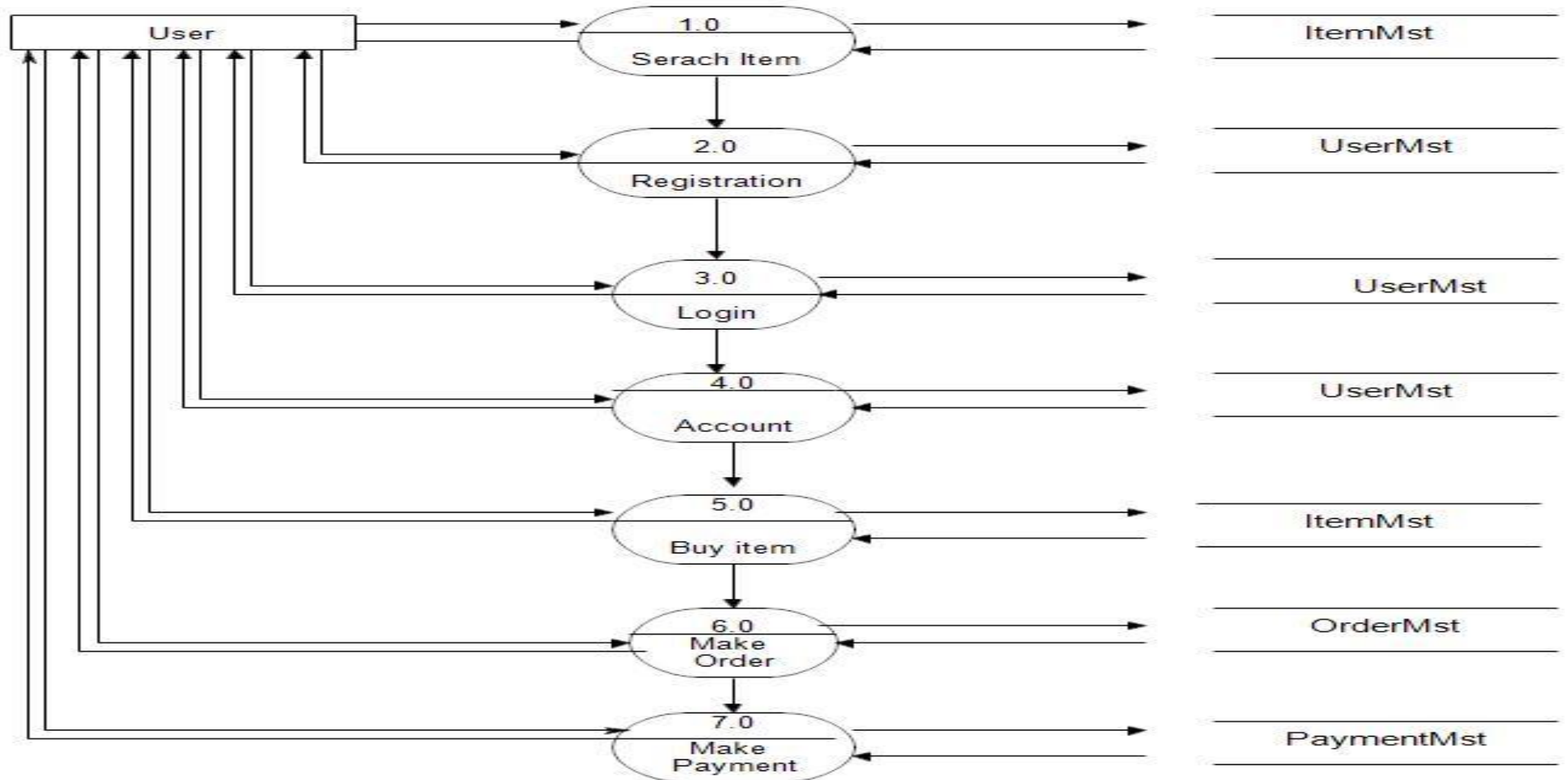
❑ LEVELS DFD(Data flow diagram)

3. LEVEL 2 DFD:

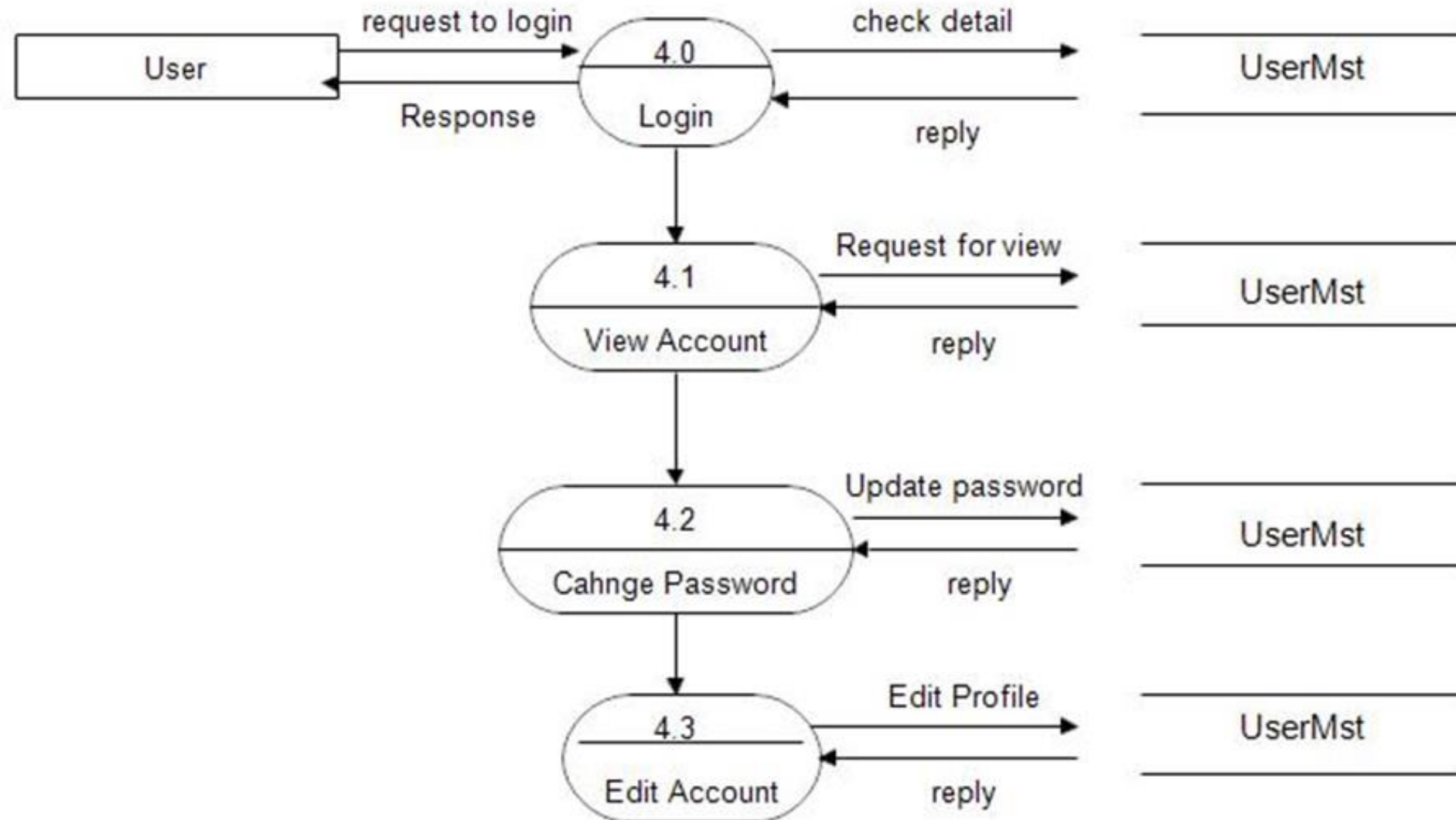
- At this level, DFD shows how data flows inside the modules mentioned in Level 1.
- Higher level DFDs can be transformed into more specific lower level DFDs with deeper level of understanding unless the desired level of specification is achieved.
- Levels in DFD are numbered 0, 1, 2 or beyond. Here, we will see primarily three levels in the data flow diagram, which are: 0-level DFD, 1-level DFD, and 2-level DFD.



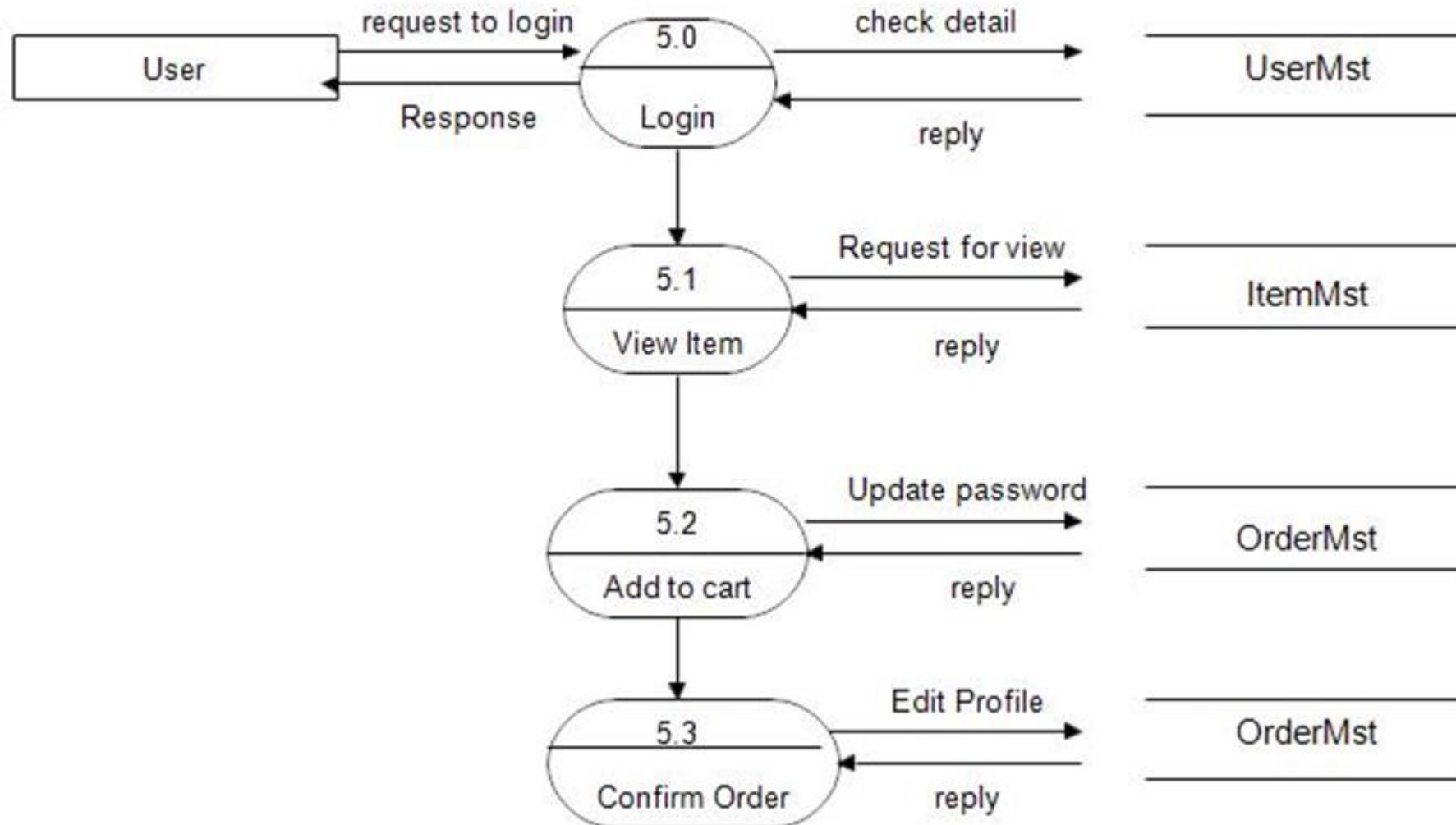
1st Level User side DFD



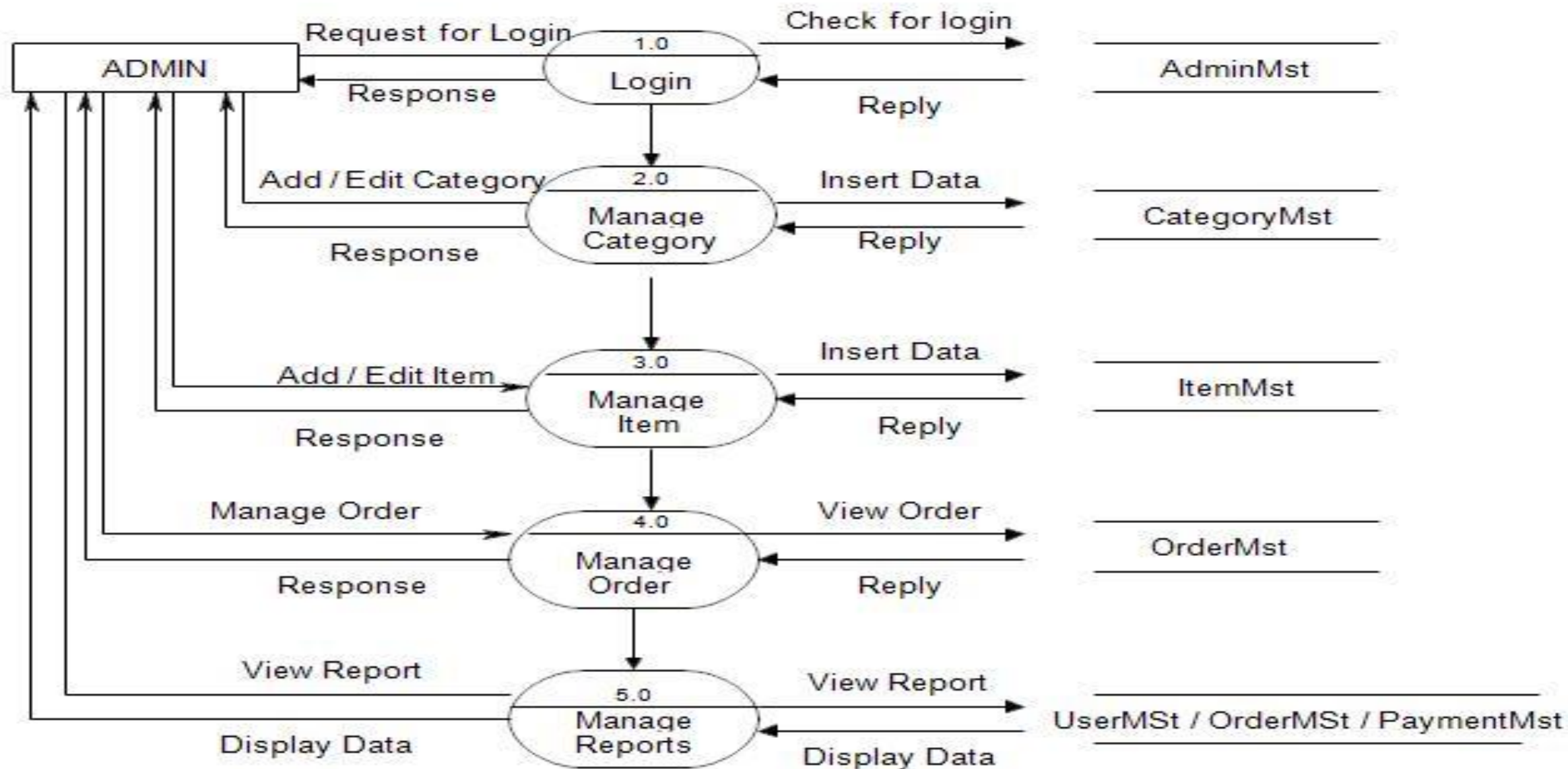
2nd Level User DFD - (4.0)



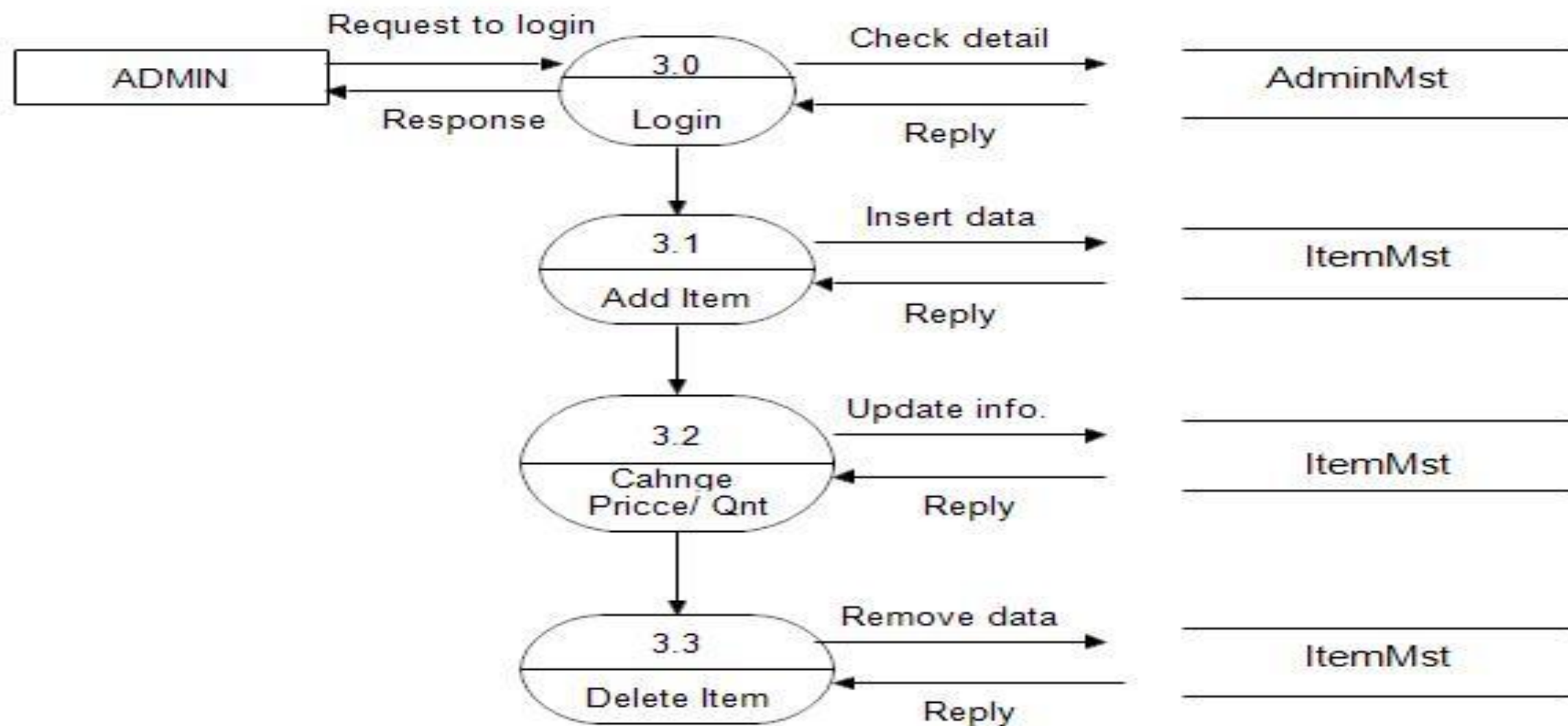
2nd Level User DFD - (5.0)



Admin Side DFD - 1st Level



2nd Level Admin DFD - (3.0)



Library Management System (LMS)

1. 0th Level DFD

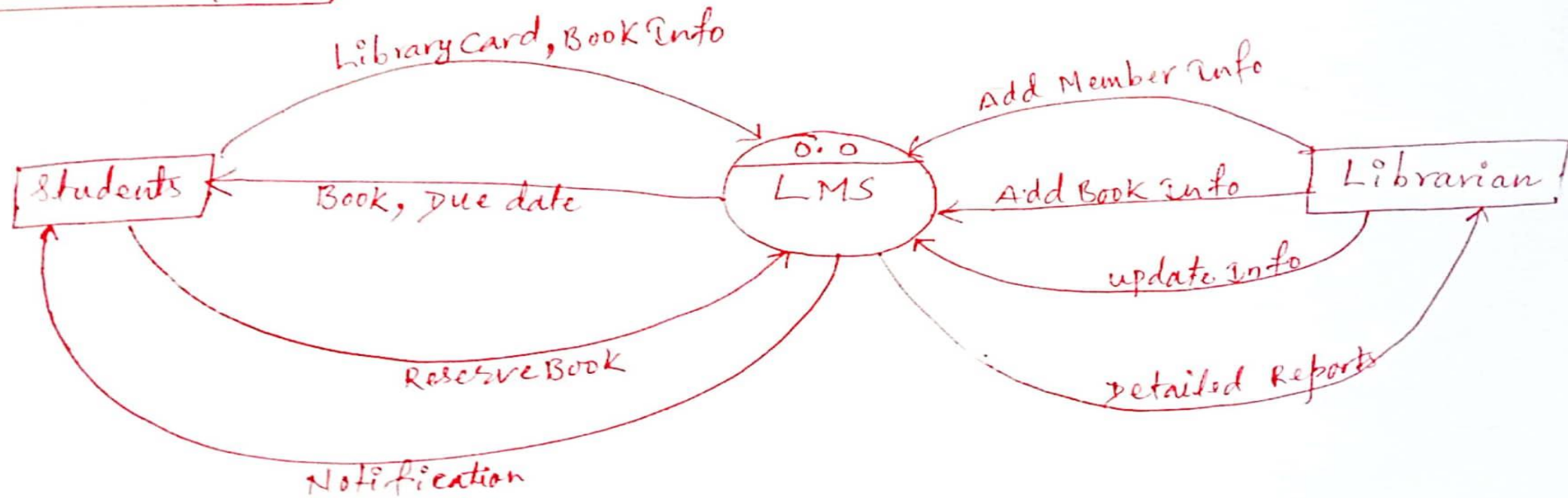


Fig: LEVEL 0 DFD / Context level DFD.

Library Management System (LMS)

2. 1st level DFD

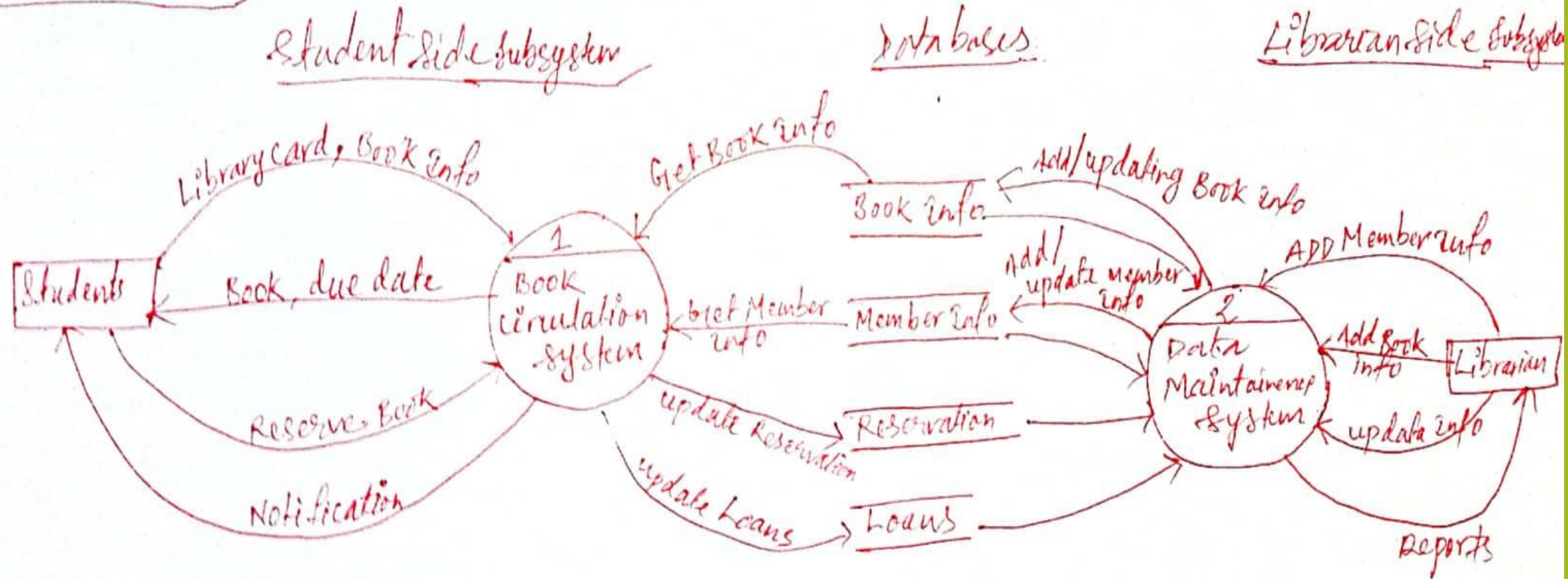
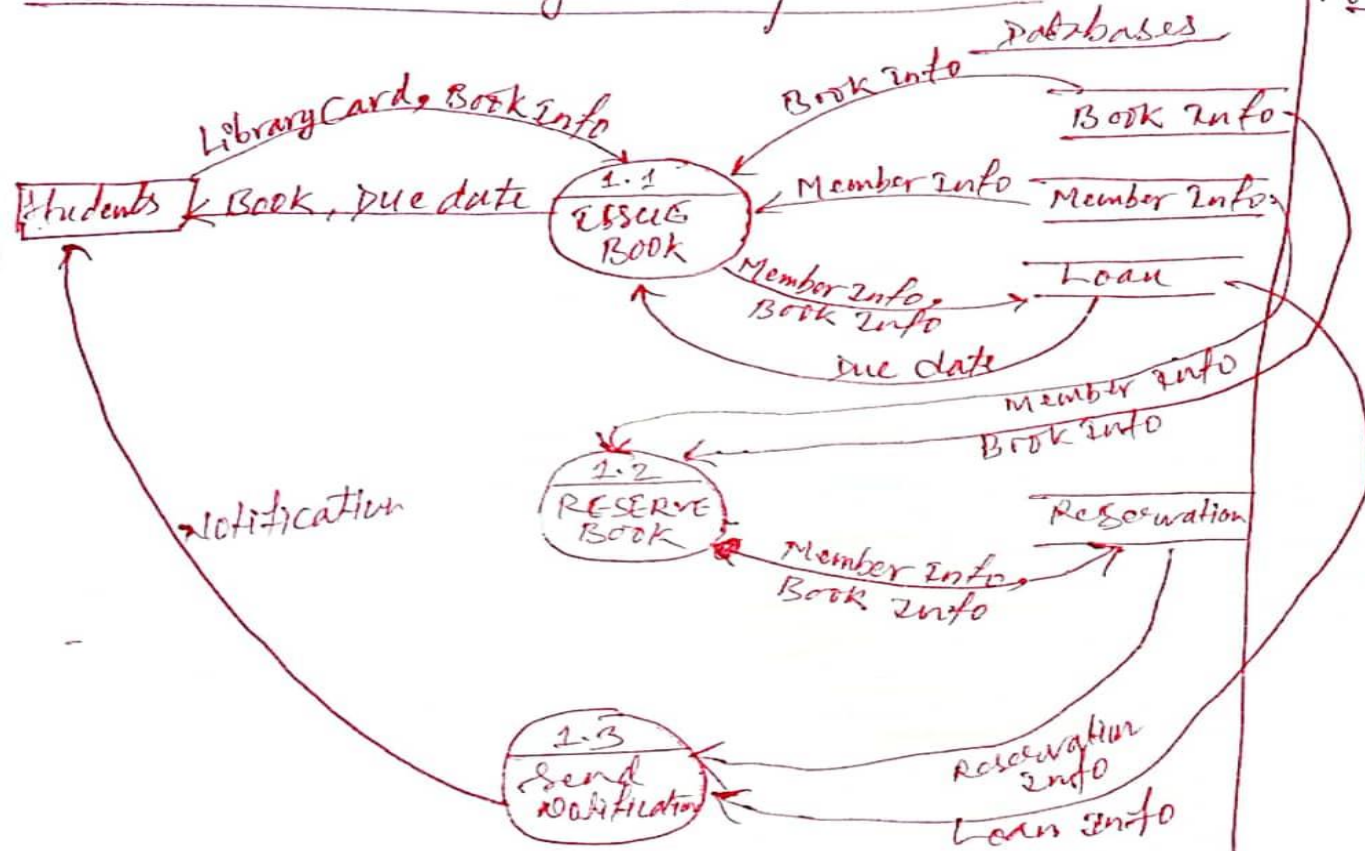


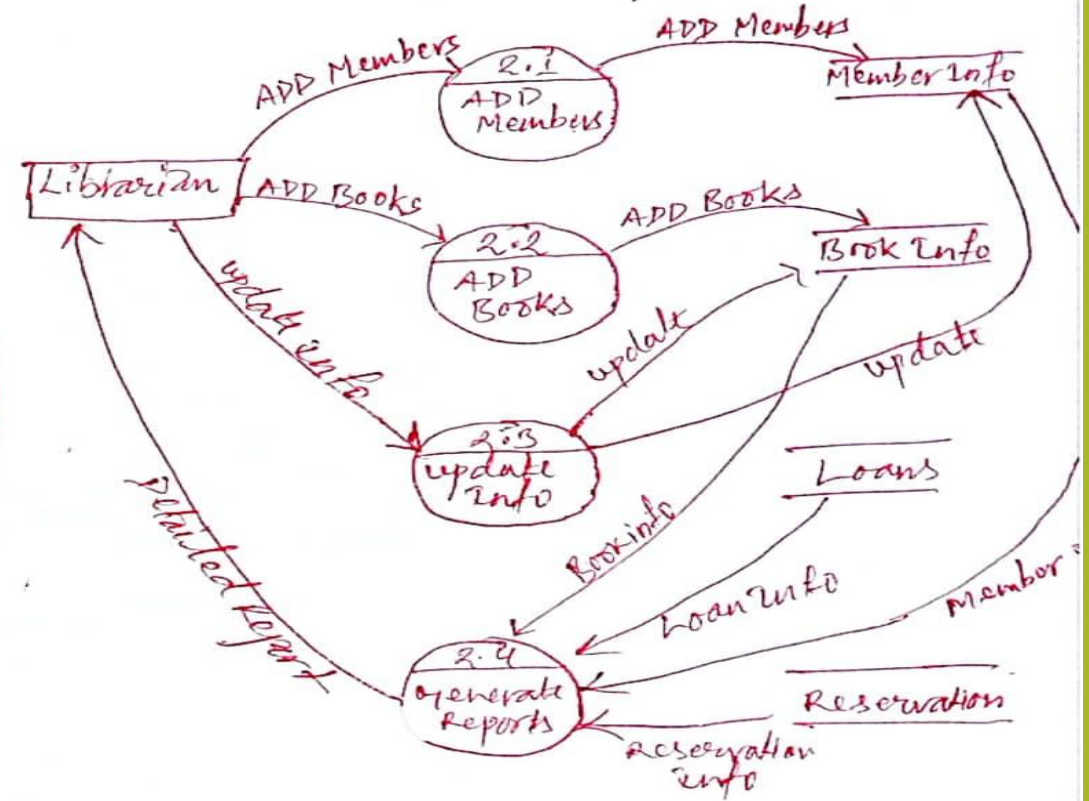
fig: level 1 DFD of Library Management System (LMS)

Library Management System (LMS) - LEVEL 2 DFD

1. Student Side Subsystem subprocesses: —



2. Librarian Side Subsystem subprocesses: —



1.What is ER Diagram?

- ➔ **ER Diagram** stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases.
- ➔ ER diagrams are created based on three basic concepts: **entities**, **attributes** and **relationships**.
- ➔ ER Diagrams contain different symbols that **use rectangles to represent entities**, **ovals to define attributes** and **diamond shapes to represent relationships**.

2.Purpose of ERD

- The system analyst gains a better understanding of the entities of an organization and their relationship through the step of constructing the ERD.
- It is used for *data modeling* in the context of system analysis and design.
- Data modeling typically involves the following steps:
- **Identify Entities**: Determine the main objects or entities within the domain you're modeling. For example, in a university system, entities could be students, courses, professors, etc.
- **Define Attributes**: Specify the characteristics or properties of each entity. For a student, attributes might include name, ID, and birthdate.
- **Establish Relationships**: Describe how entities are related to each other. Students might be enrolled in courses, creating a relationship between the two.

- **Normalize Data:** In relational databases, data normalization is the process of organizing data to minimize redundancy and improve data integrity.
- **Create Diagrams:** Visual representations, such as entity-relationship diagrams, help communicate the structure and relationships of the data.
- **Apply Constraints:** Define rules and constraints that ensure data accuracy and consistency.

For example, a student's birthdate should be within a certain range.i.e. a student's birthdate should be realistic, like not in the future or too far in the past.

E.g. every student must have a unique student ID, so there's no confusion between students.

3.Components of the ER Diagram.

- This model is based on three basic concepts:

1.Entities

Examples of entities:

- **Person:** Employee, Student, Patient, customer
- **Place:** Class room , Building
- **Object:** Machine, product, and Car
- **Event:** Sale, Registration, Renewal
- **Concept:** Account, Course

2.Attributes

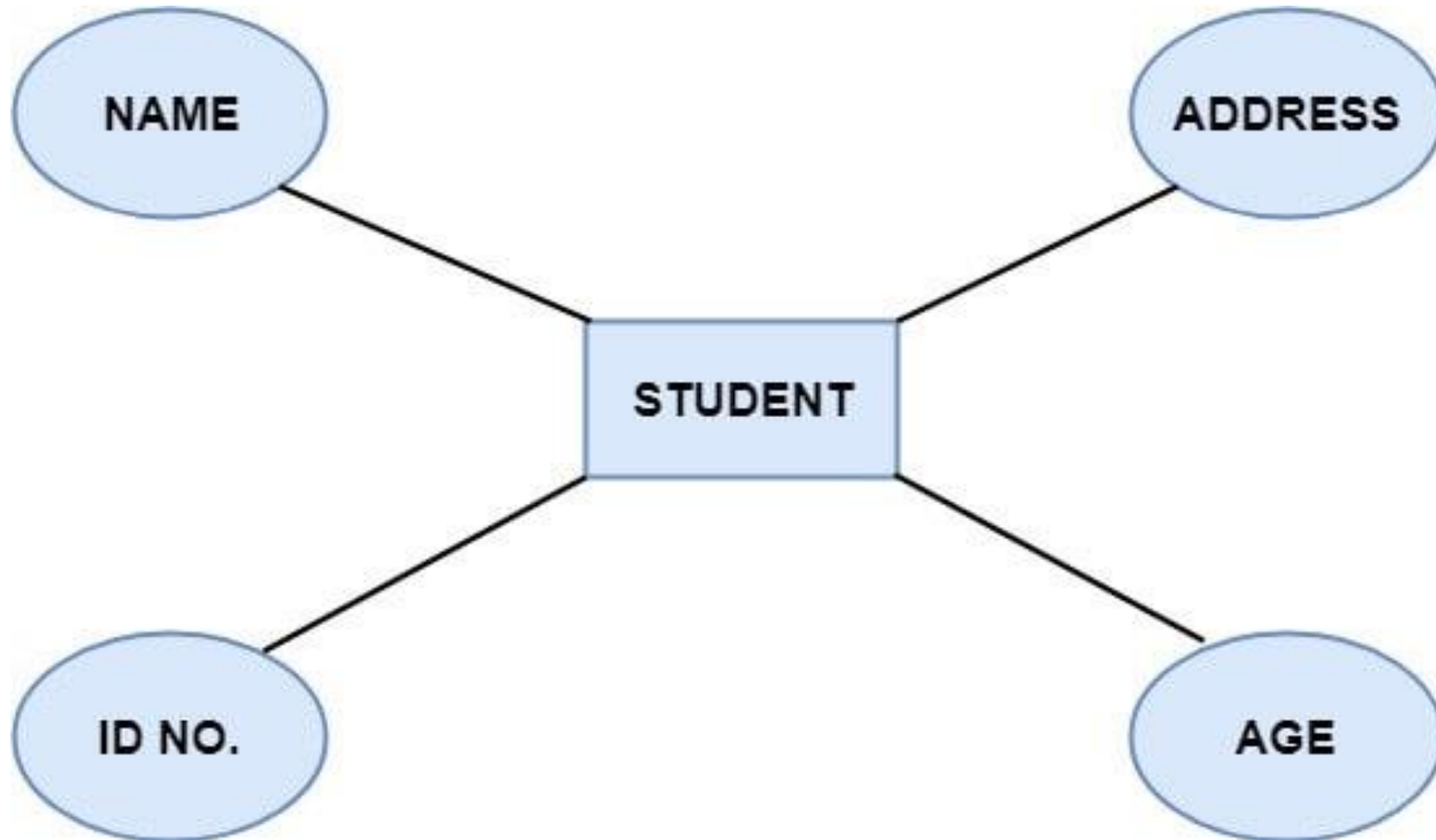
❖Attributes are the properties of an entities.

For example, a student entity may have name, class, and age as attributes.

❖ There exists a domain or range of values that can be assigned to attributes.

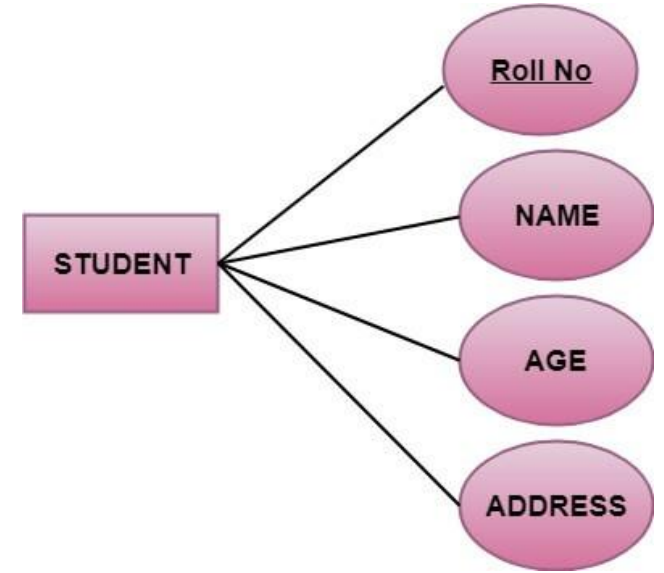
For example, A student's name cannot be a numeric value. It has to be alphabetic.

A student's age cannot be negative, etc



❑ There are four types of Attributes:

- ❖ Key attribute
- ❖ Composite attribute
- ❖ Single-valued attribute
- ❖ Multi-valued attribute
- ❖ Derived attribute

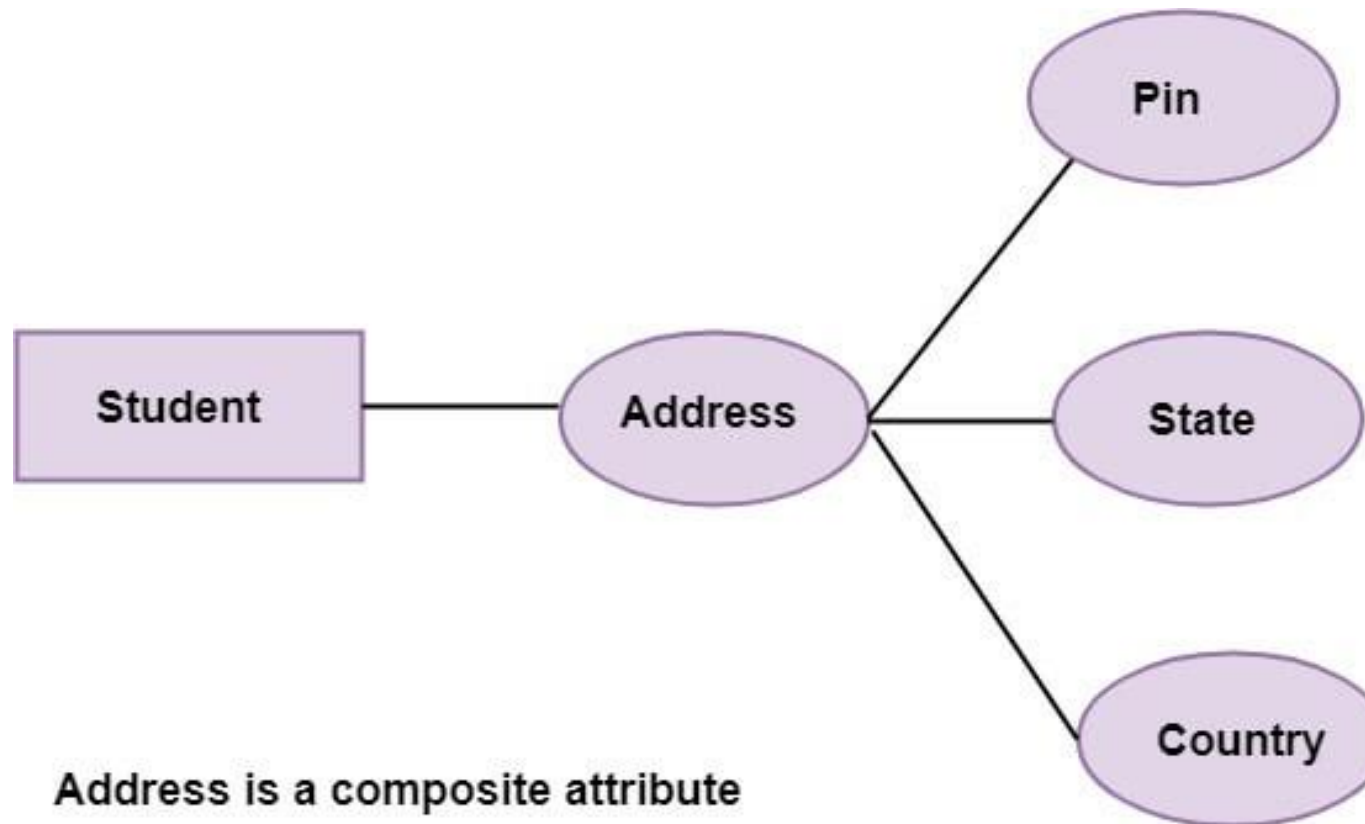


1.Key attribute: Key is an attribute that uniquely identifies an entity among the entity set.

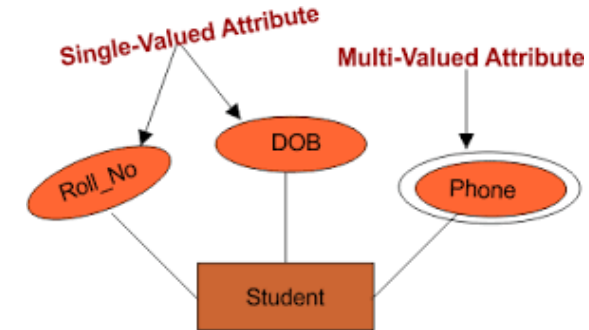
For example, the **Roll No** of a student makes him/her identifiable uniquely among students.

2. Composite attribute: An attribute that is a combination of other attributes is called a composite attribute.

For example, In **student** entity, the **student address** is a composite attribute as an address is composed of other characteristics such as **pin code, state, country**.



3. Single-valued attribute: Single-valued attribute contain a single value.

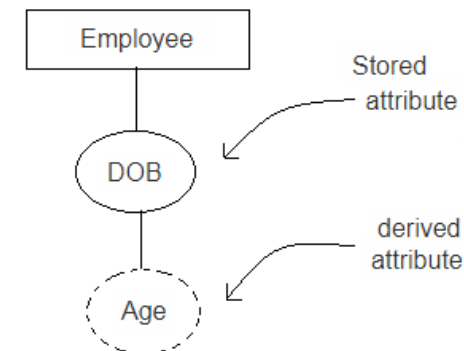


4. Multi-valued Attribute: If an attribute can have more than one value, it is known as a multi-valued attribute. Multi-valued attributes are depicted by the double ellipse.

For example, a person can have more than one **phone number, email-address**, etc.

5. Derived attribute: Derived attributes are the attribute that does not exist in the physical database, but their values are derived from other attributes present in the database.

For example, **age** can be derived from **date of birth**. In the ER diagram, Derived attributes are depicted by the dashed ellipse.



3. Relationships

- The association among entities is known as relationship.
- Relationships are represented by the diamond-shaped box.

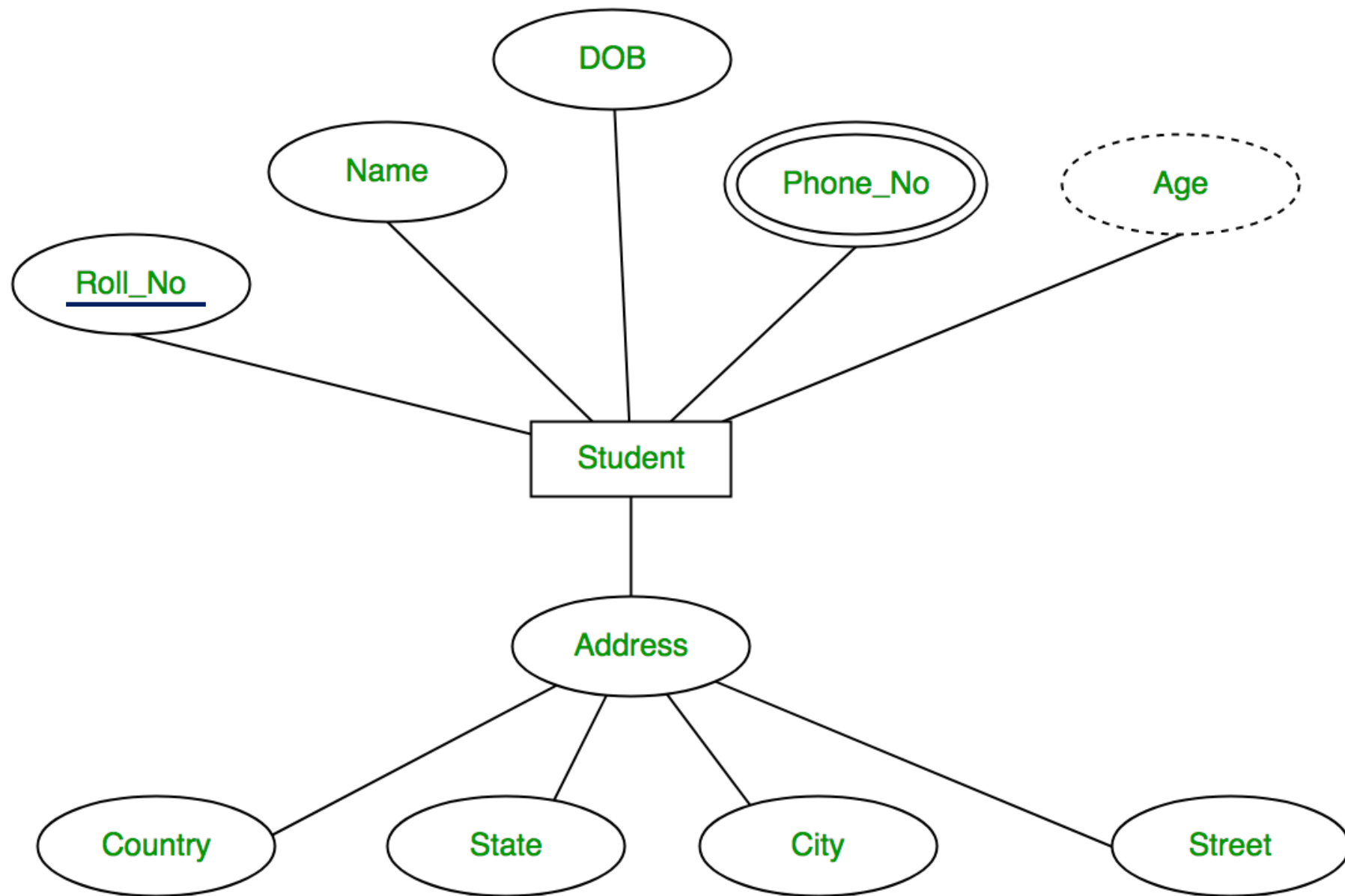
For example, An employee works at a department,

A student enrolls in a course.

Here, *Works_at* and *Enrolls* are called relationships between the entities.



Fig: Relationships in ERD



❑ Degree of a relationship set

- The number of participating entities in a relationship describes the degree of the relationship. The three most common relationships in E-R models are:
- Unary (degree1)
- Binary (degree2)
- Ternary (degree3)

1. Unary relationship: This is also called recursive relationships. A types of relationship that involves only *one entity* type or the relationship between the of *same entity* type.

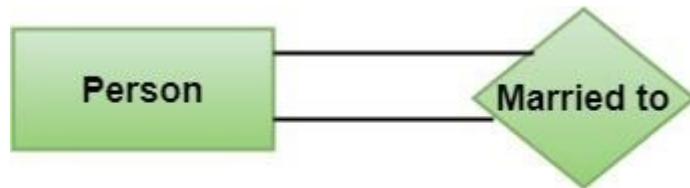
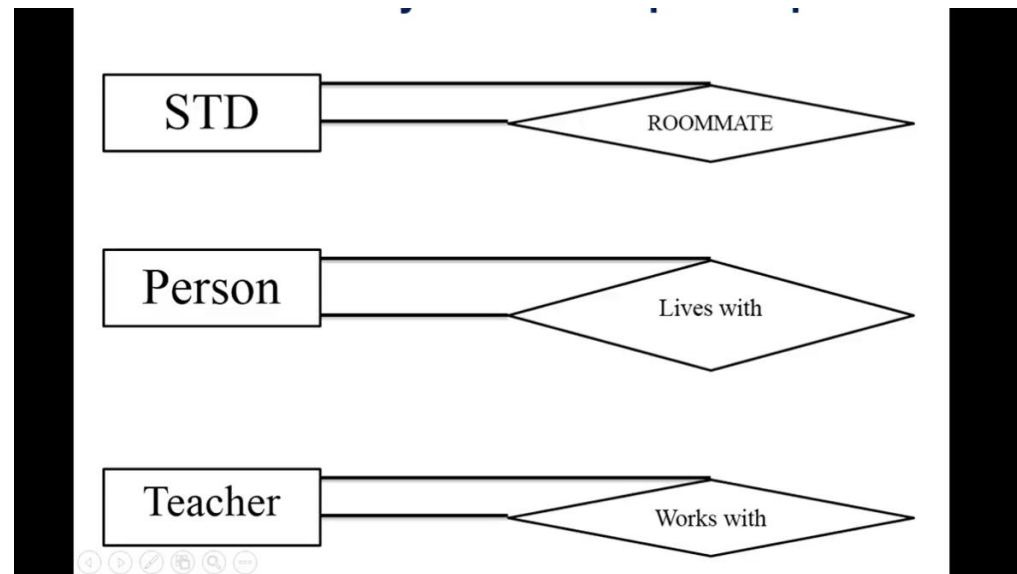


Fig: Unary Relationship



2. Binary relationship: It is a relationship between the instances of two entity types. For example, the Teacher teaches the subject.



Fig: Binary Relationship

3. Ternary relationship: It is a relationship amongst instances of *three entity types*. In fig, the relationships "may have" provide the association of three entities, *i.e. TEACHER, STUDENT, and SUBJECT*. A Ternary relationship exists when there are three types of entity and we call them a degree of relationship is 3.

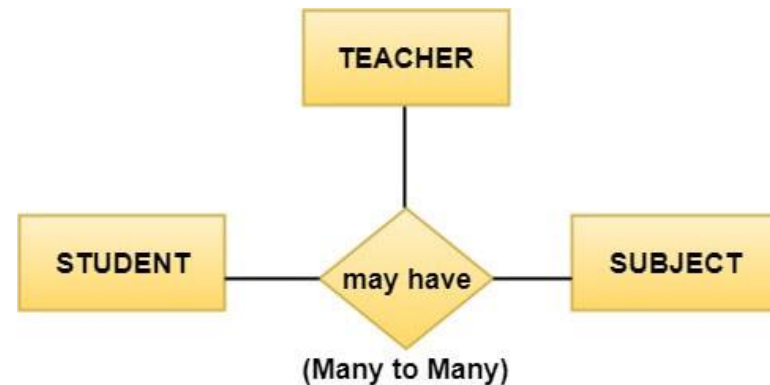


Fig: Ternary Relationship

Types of Cardinalities

1. One to One: Let us assume that each **student** has only one **student ID**, and each **student ID** is assigned to only **one person**. So, the relationship will be one to one.



Beginnersbook.com

2. One to many: When a single instance of an entity is associated with more than one instances of another entity then it is called one to many relationships.

For example, a **client(Customer)** can place many **orders**; a **order** cannot be placed by many customers.



3. Many to One:

For example - many students can study in a single college, but a student cannot study in many colleges at the same time.



4. Many to Many:

For example, the students can be assigned to many projects, and a projects can be assigned to many students.



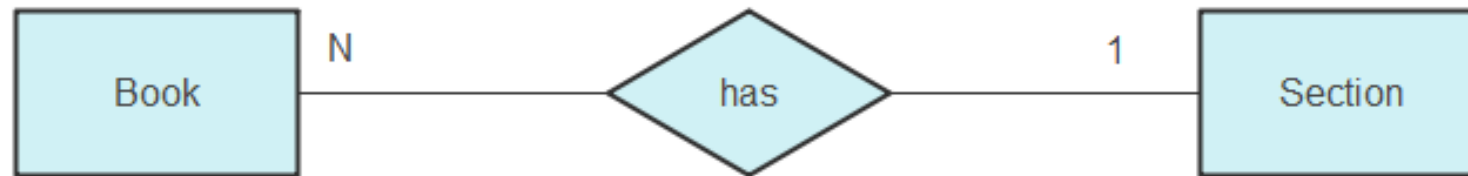
one-to-one (1:1)



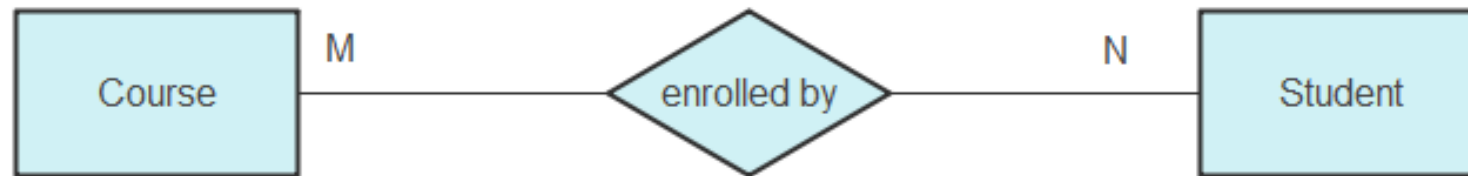
one-to-many (1:N)



many-to-one (N:1)

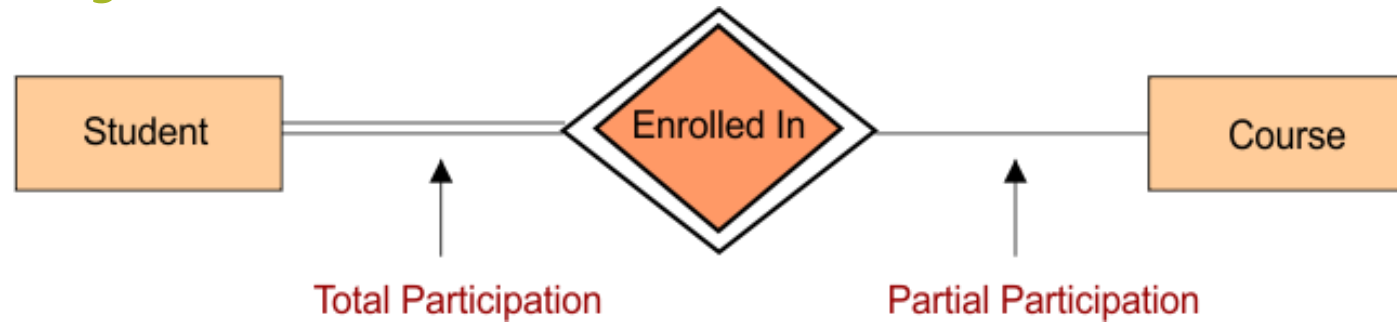


many-to-many (M:N)



#Participation Constraint:

1. Total Participation – Each entity in the entity set **must participate** in the relationship. If each **student** must enroll in a **course**, the participation of **student** will be total. Total participation is shown by **double line** in ER diagram.



2. Partial Participation – The entity in the entity set **may or may NOT participate** in the relationship. If some **courses** are not enrolled by any of the **student**, the participation of course will be partial.

E.g. Every Student is guided by a professor(Total Participation)

Not every professor guides a student(Partial Participation)



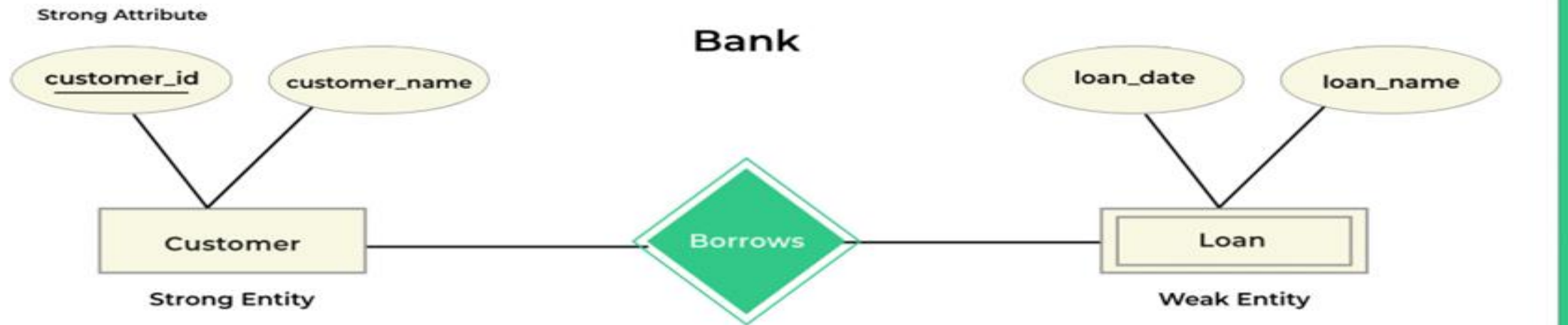
#Weak Entities

- An entity type should have a key attribute which uniquely identifies each entity in the entity set, but there exists some entity type for which key attribute(primary key) can't be defined. These are called ***Weak Entity type***.
- The ***double diamond*** symbol is used for representing the **relationship between a strong entity and a weak entity** which is known as identifying relationship.

Note – Weak entity always has total participation but Strong entity may not have total participation.

All forms of **loan** belongs to a certain **customer** but a **customer** may or may not borrow any form of the **loan**.

Weak Entity



Initialisation

Strong Entities	Customer
Weak Entities	Loan
Strong Attributes	customer_id
Partial Key	loan_name

== Total Participation

— Partial Participation

Weak Entity

Weak Relationship

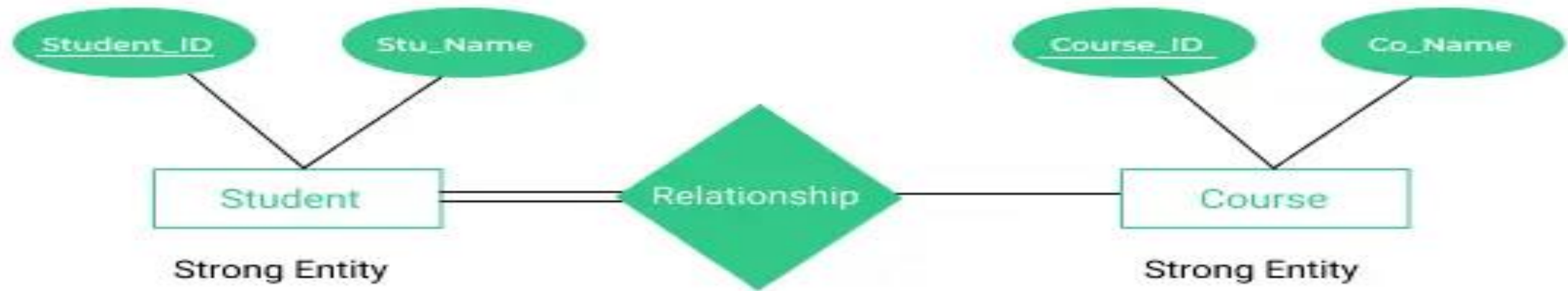
Strong Entity

Attribute

2.Strong entity

- ❖ A Strong entity is nothing but an entity set having a *primary key* attribute or a table that consists of a primary key column.
- ❖ The relationship between two strong entities is represented by a single diamond.
- ❑ Consider the ER diagram which consists of two entities **student** and **course**:
 - ❖ **Student entity is a strong entity** because it consists of a **primary key** called student id which is enough for accessing each record uniquely.
 - ❖ In the same way, the **course** entity contains of course ID attribute which is capable of uniquely accessing each row.

Strong Entity



Strong Entity Student, Course

Strong Attribute Student_ID,
Course_ID



Relationship



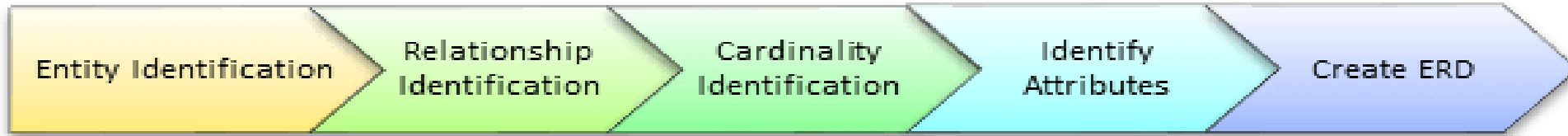
Entity



Attribute

#How to Create an Entity Relationship Diagram (ERD)

→ Following are the steps to create an ER Diagram:

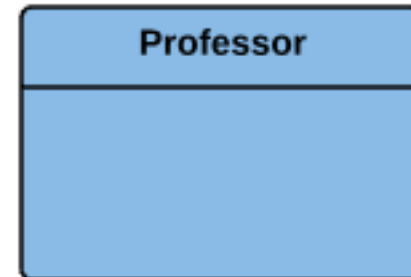
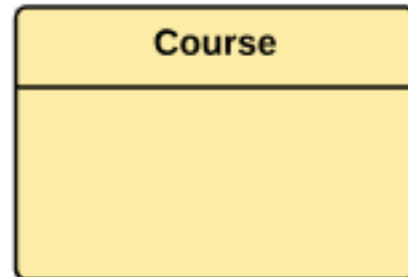
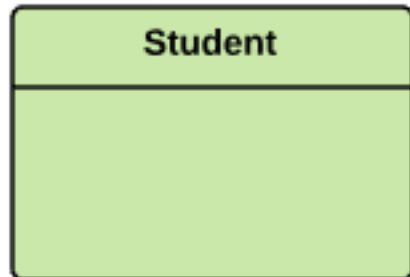


In a university, a Student enrolls in Courses. A student must be assigned to at least one or more Courses. Each course is taught by a single Professor. To maintain instruction quality, a Professor can deliver only one course.

Step 1) Entity Identification

We have three entities

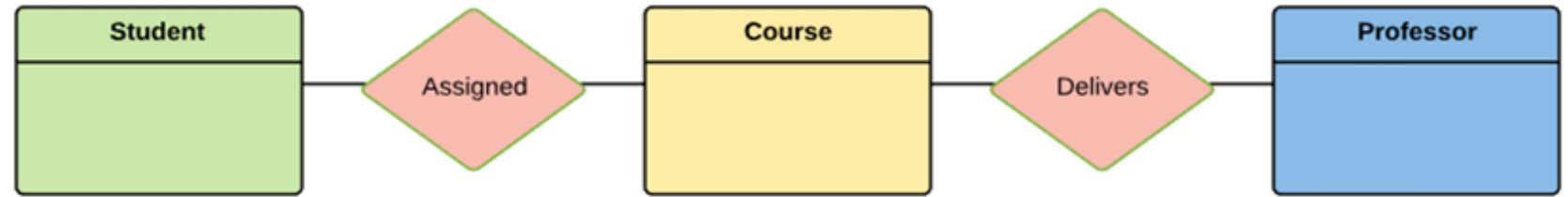
- Student
- Course
- Professor



Step 2) Relationship Identification

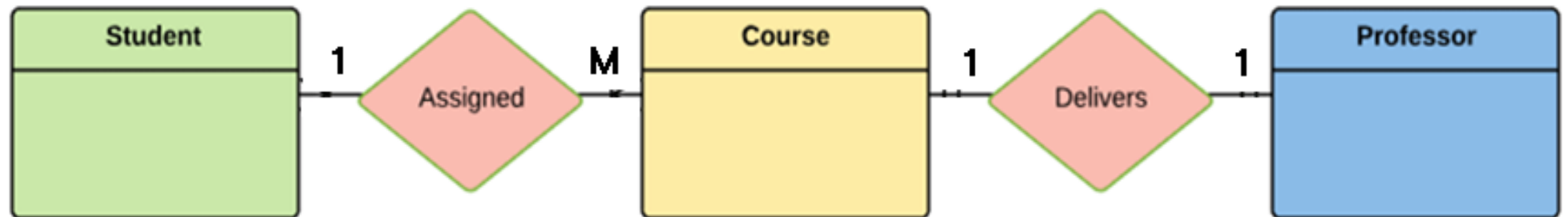
We have the following two relationships

- The **student** is **assigned** a **course**
- **Professor** **delivers** a **course**



Step 3) Cardinality Identification

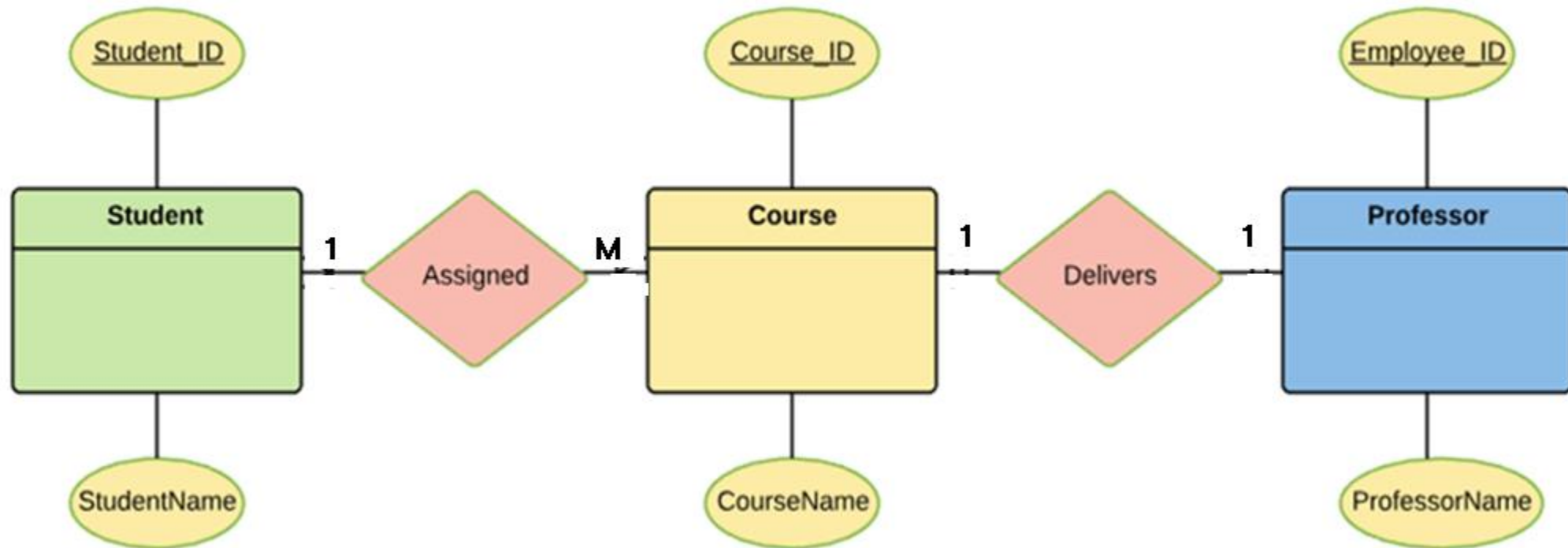
- For the problem statement we know that,
- A student can be assigned **multiple** courses
- A Professor can deliver only **one** course



Step 4) Identify Attributes

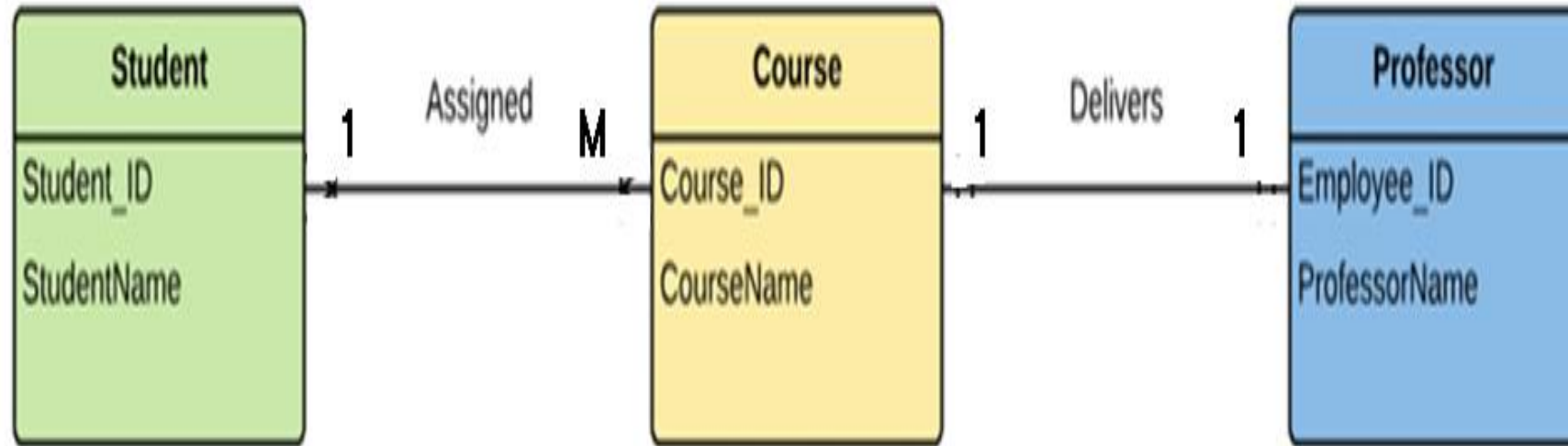
- Once the mapping is done, identify the primary Keys. If a unique key is not readily available, create one.

Entity	Primary Key	Attribute
Student	Student_ID	StudentName
Professor	Employee_ID	ProfessorName
Course	Course_ID	Course Name



Step 5) Create the ERD Diagram

- A more modern representation of Entity Relationship Diagram Example







❖ Modeling Tools

❖ *Decision Tree:*

- A decision tree is a widely used and data *analysis tool* that helps in *making decisions or predictions* by visually representing a series of choices/conditions and their potential consequences/output.
- Decision tree is a graphical representation that presents conditions and actions sequentially.
- The size of the tree will be depend upon the number of conditions and actions.
- Each condition is expressed in two ways: True/False or Yes/No.
- Decision trees are commonly used in operation research, decision analysis, & descriptive means for calculating conditional probabilities.
- Decision trees are used in various fields, including finance, healthcare, marketing, and more.

❑ Components of a Decision Tree:

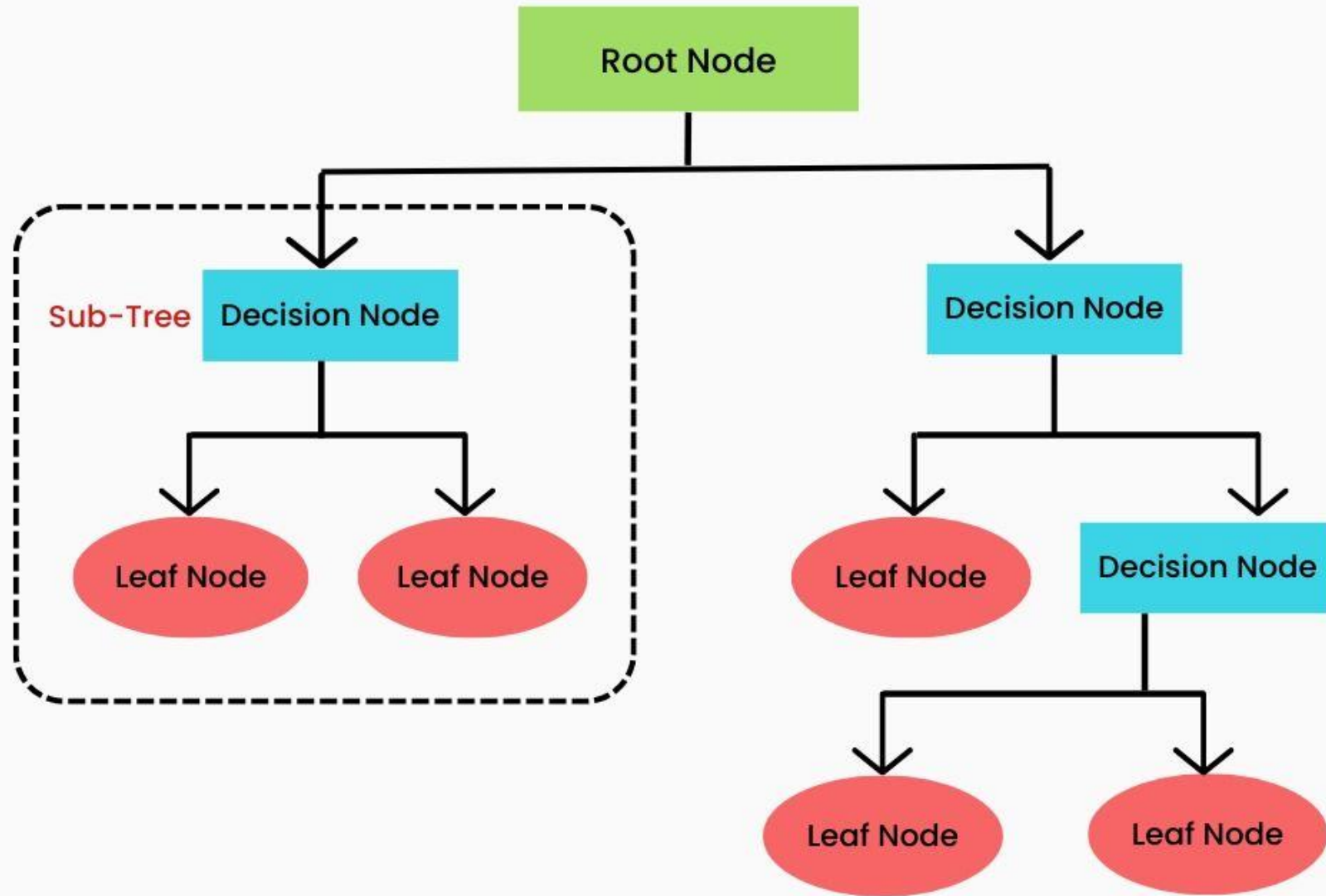
- A decision tree consists of the following components:

1. **Root Node:** The starting point of the tree, representing the initial decision or question.
2. **Internal Nodes:** These nodes represent decisions based on specific features or attributes. It is also called decision node.
3. **Branches:** The edges connecting nodes, showing the possible outcomes of decisions.
4. **Leaf Nodes:** The end nodes that represent final decisions or outcomes.
5. **Attributes:** Characteristics or features used to make decisions at internal nodes.

❖ Decision Tree and Decision Table:

- A decision tree consists of the following components:

1. **Root Node:** The starting point of the tree, representing the initial decision or question.
2. **Internal Nodes:** These nodes represent decisions based on specific features or attributes. It is also called decision node.
3. **Branches:** The edges connecting nodes, showing the possible outcomes of decisions.
4. **Leaf Nodes:** The end nodes that represent final decisions or outcomes.
5. **Attributes:** Characteristics or features used to make decisions at internal nodes.



❑ Decision Table:

- A decision table is a brief visual representation for specifying which actions to perform depending on given conditions.

❑ Components of a Decision Table:

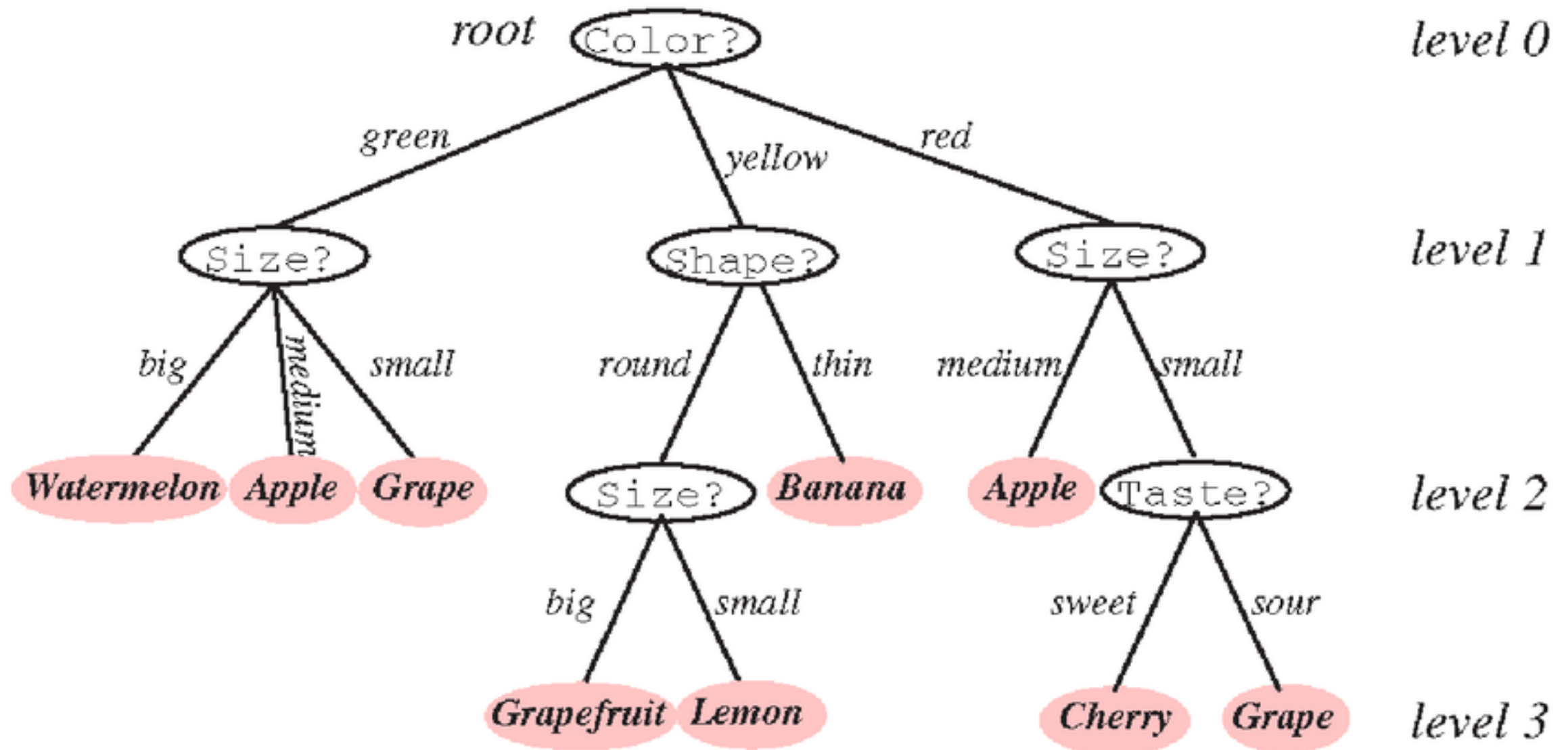
- **Conditions (Inputs):** Conditions represent the factors or variables that affect a decision or outcome. They are typically *Boolean values (true or false)* or *categorical variables*.
- **Actions (Outputs):** Actions represent the possible outcomes or actions that result from a combination of conditions being met.
- **Rules:** Each row in the decision table represents a specific combination of conditions and the corresponding action(s) that should be taken when those conditions are met. These rows are also called rules.

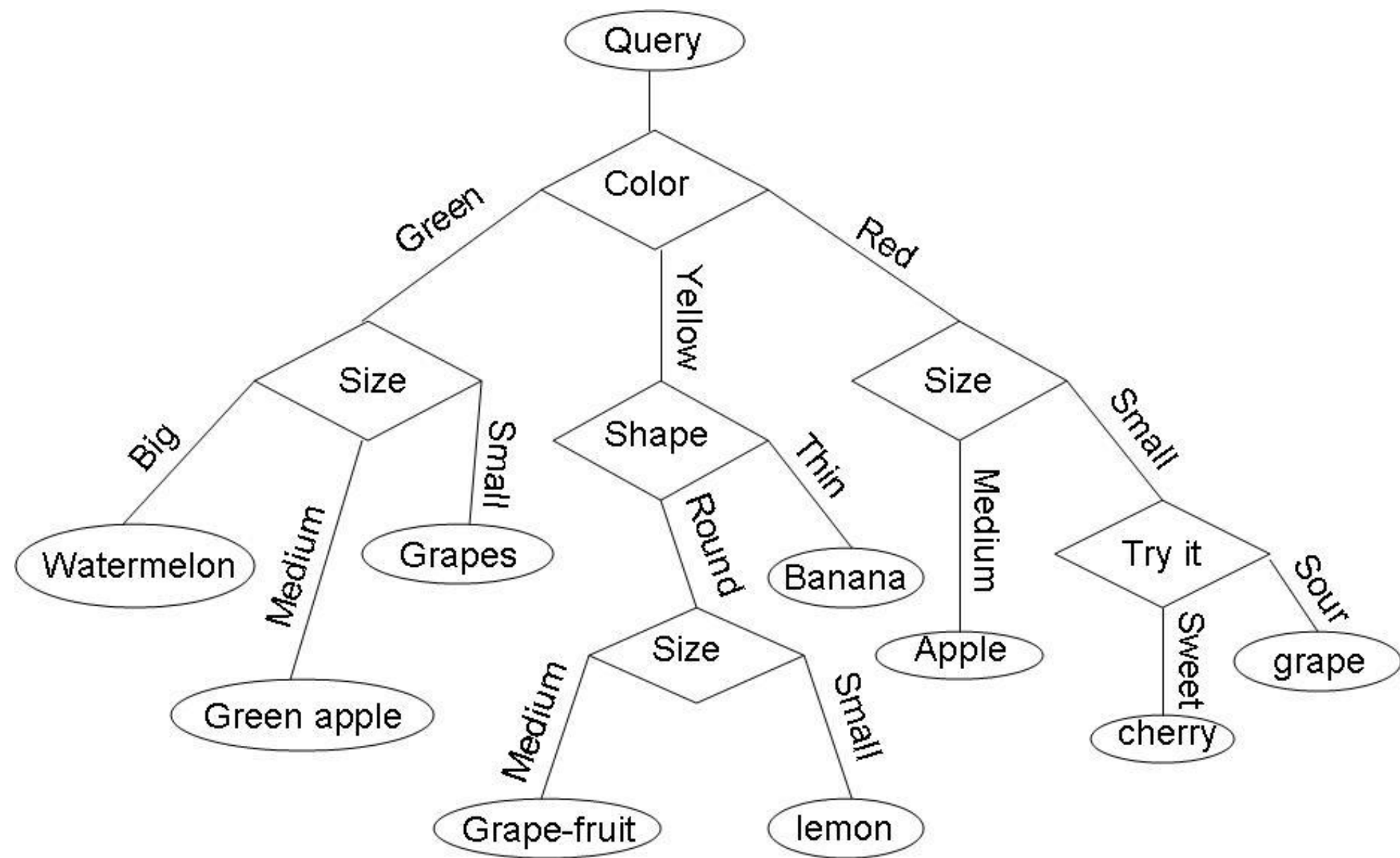
Scenario: Discount Offer Decision

In this decision table, we are deciding whether to offer a discount to a customer based on their purchase amount and loyalty status.

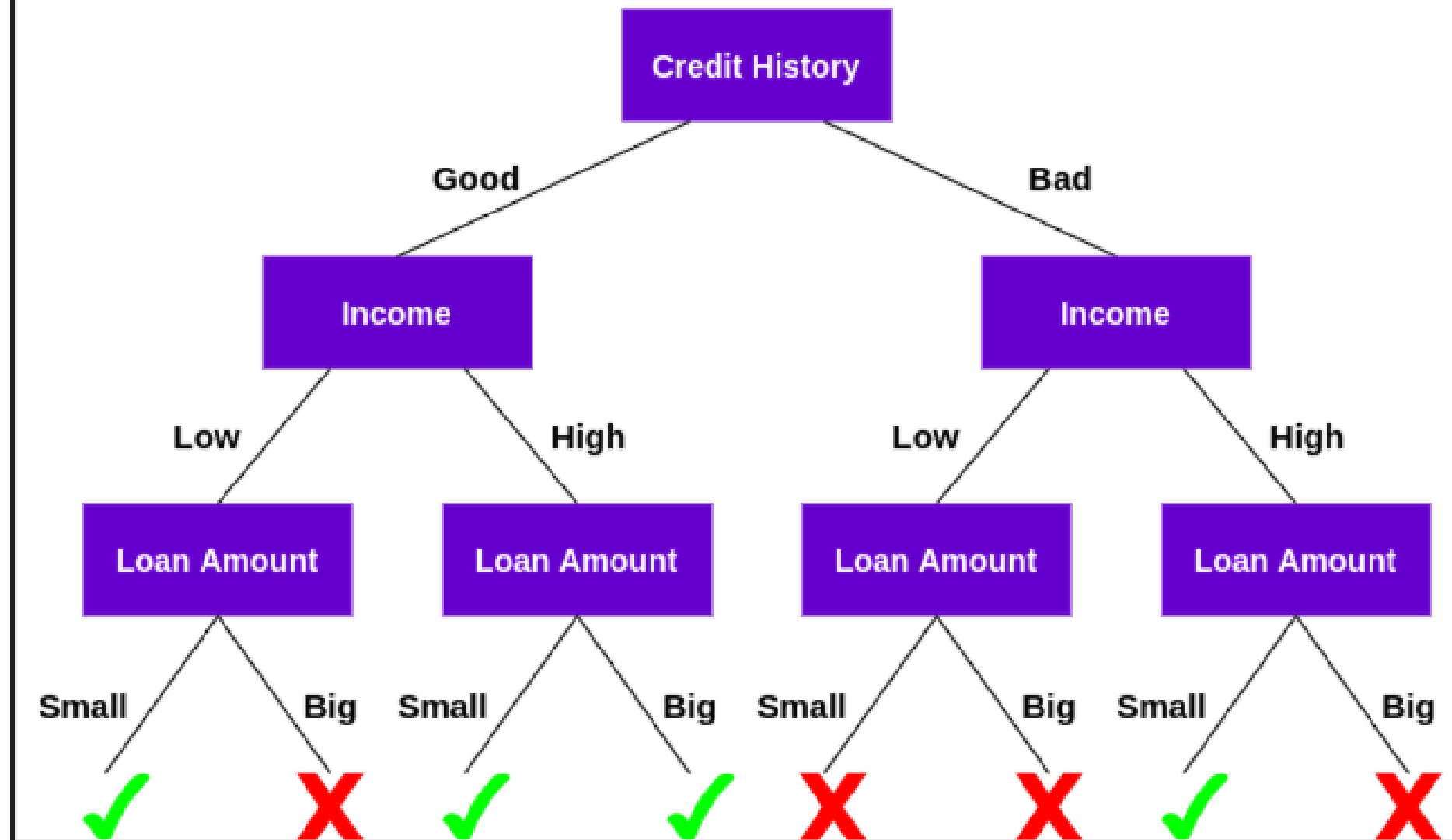
Condition 1 (Purchase Amount)	Condition 2 (Customer Loyalty)	Action (Discount Offer Decision)
\$100 or more	Loyal	Offer a 10% discount
\$100 or more	Not Loyal	Offer a 5% discount
Less than \$100	Loyal	Offer a 5% discount
Less than \$100	Not Loyal	Do not offer a discount

1. Create a decision tree to identify different fruits based on the actions: color ,shape and size.

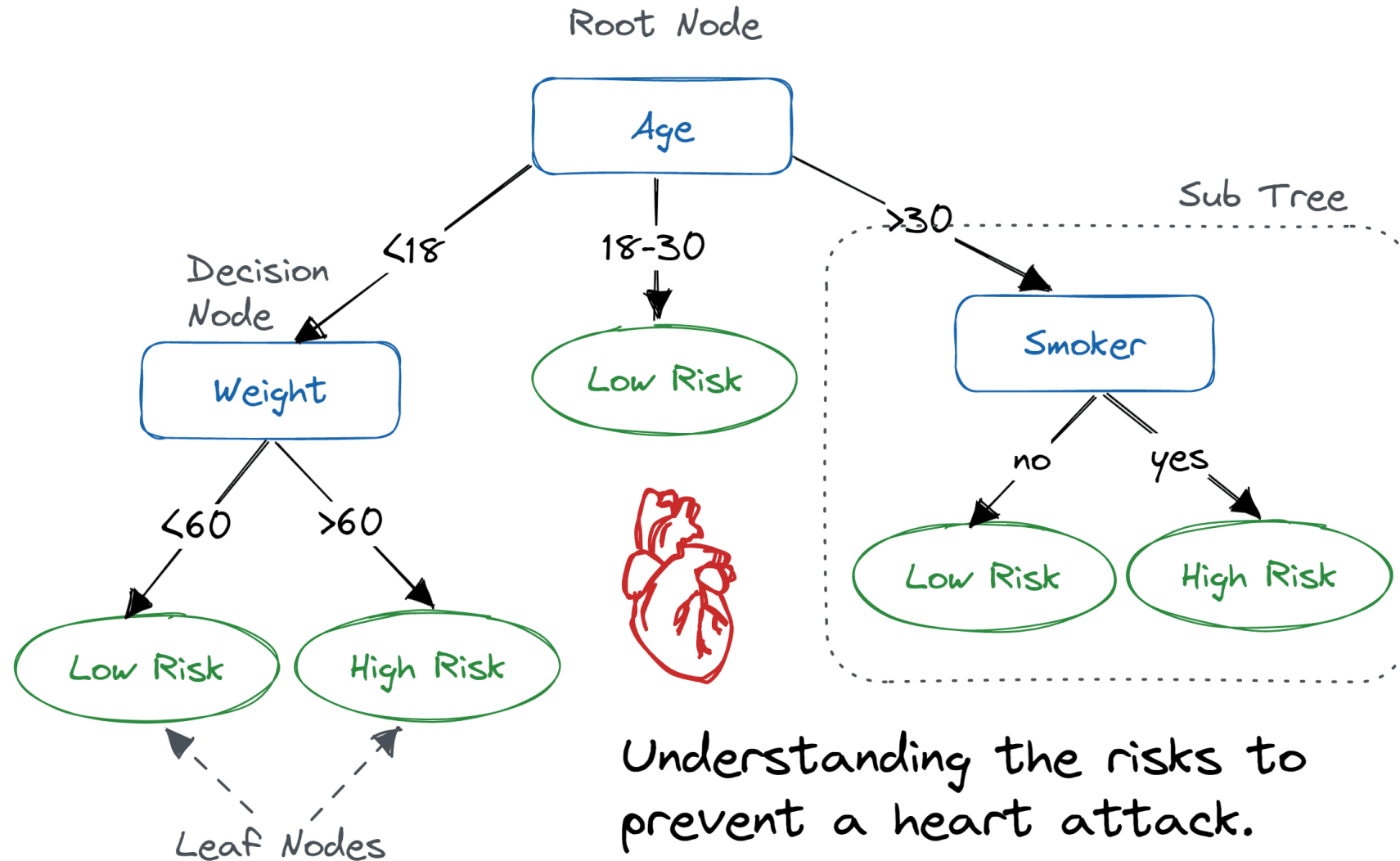




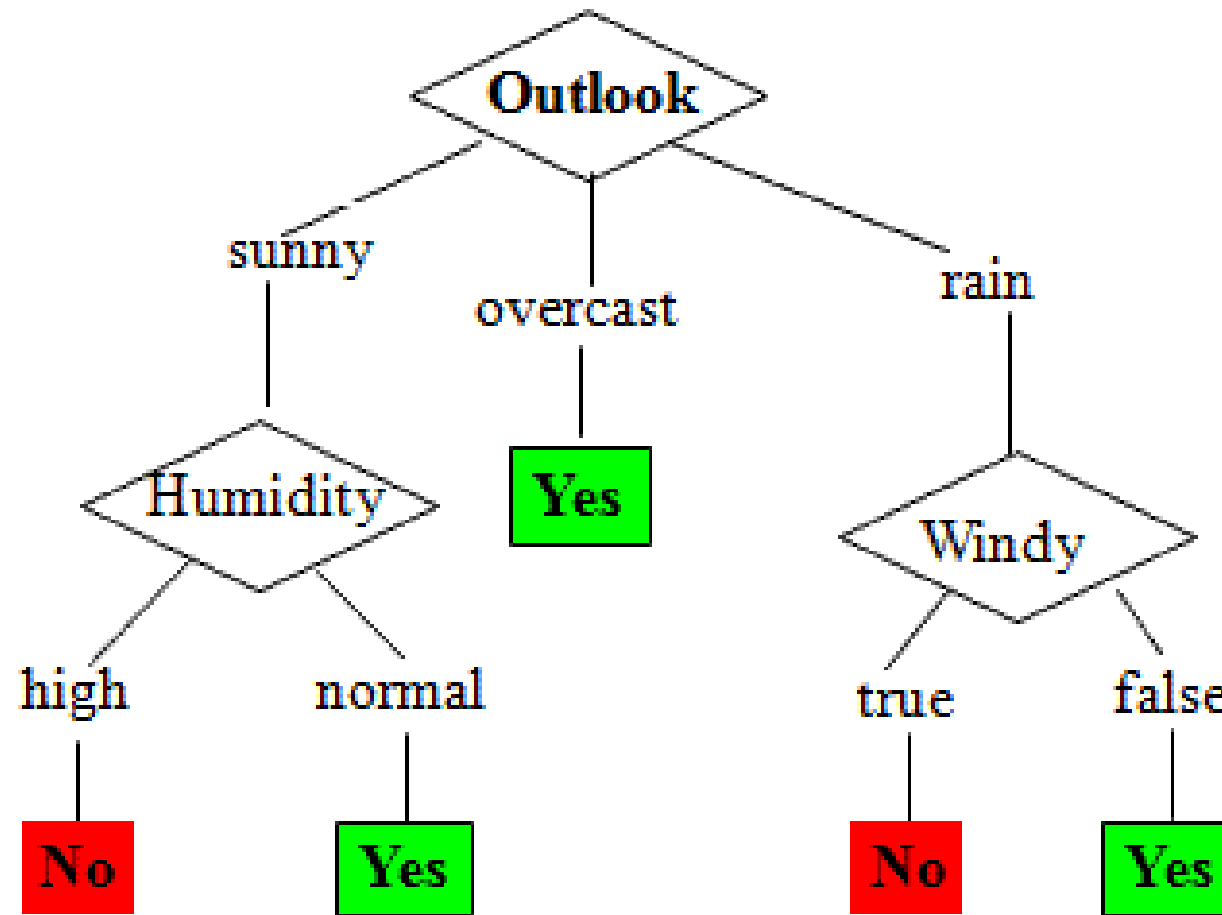
Decision Tree for Loan Approval



2. Create a decision tree to understand the risk of Heart Attack based on the conditions: **Age** , **weight** and **smoking habit**.



- ❑ A decision tree to determine whether to play football or not based on the outlook or weather condition.



Age > 30

No

Yes

Lot of Junk Food?

Exercise?

Yes

No

No

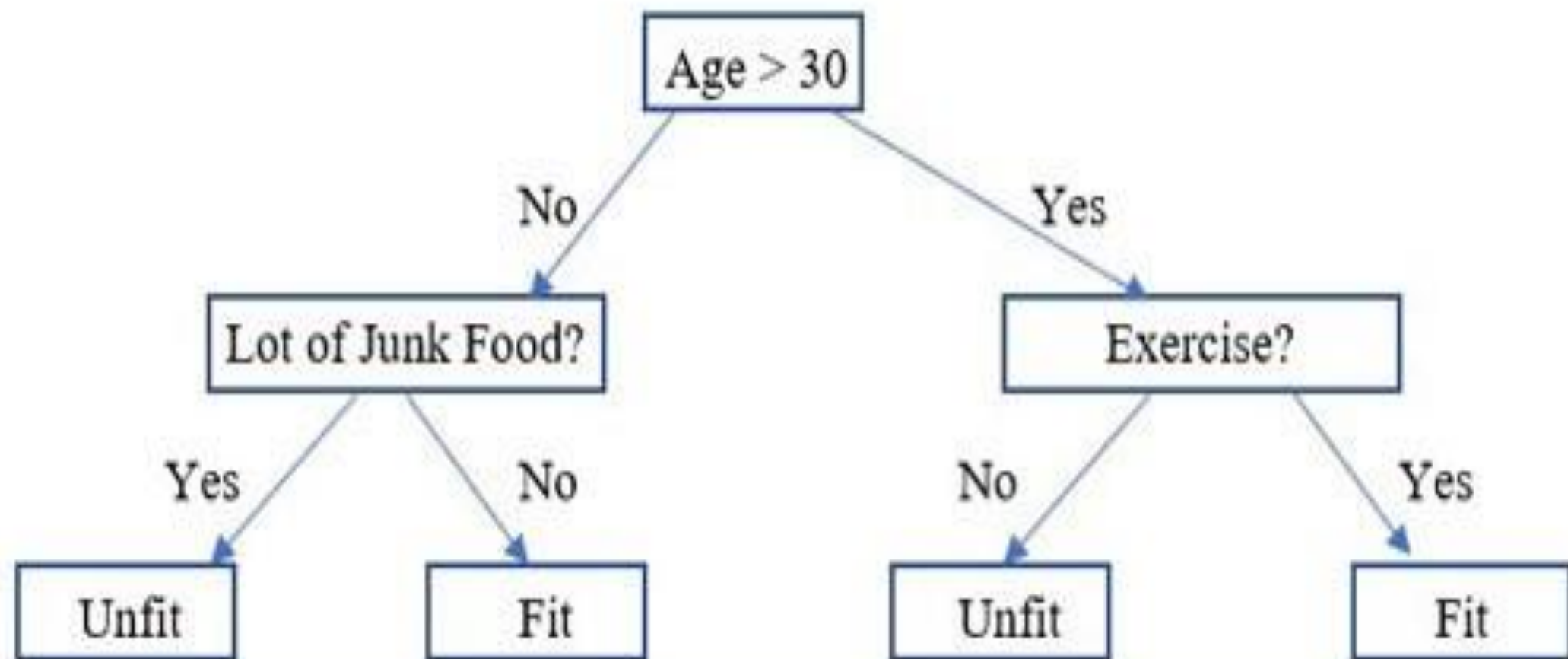
Yes

Unfit

Fit

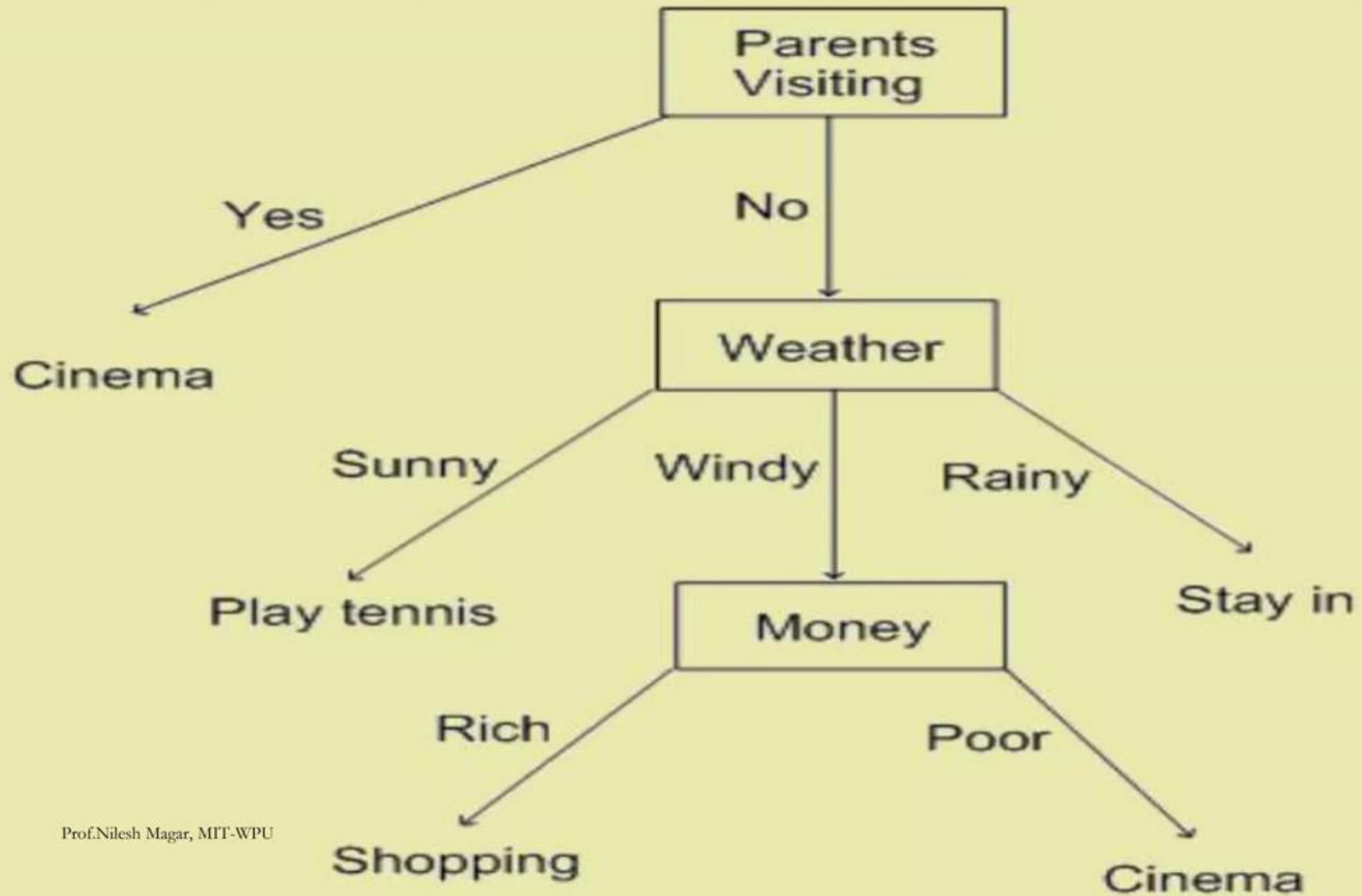
Unfit

Fit



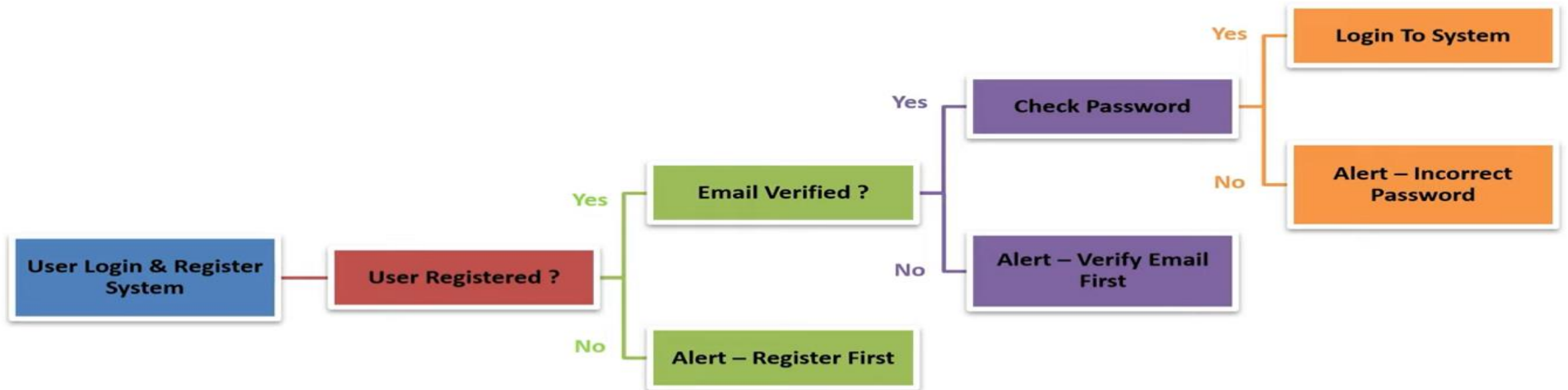
EXAMPLE.-

- Imagine you only ever do four things at the weekend: **go shopping, watch a movie, play tennis or just stay in.**
- What you do depends on three things: the **weather** (windy, rainy or sunny); **how much money you have** (rich or poor) and **whether your parents are visiting.**
- You say to your yourself:
 - 1) **if my parents are visiting, we'll go to the cinema.**
 - 2) **If they're not visiting and it's sunny, then I'll play tennis,**
 - 3) **If it's windy, and I'm rich, then I'll go shopping.**
 - 4) **If they're not visiting, it's windy and I'm poor, then I will go to the cinema.**
 - 5) **If they're not visiting and it's rainy, then I'll stay in.**



- ❑ **Design a decision tree of user login and Register System:**
1. If user is Registered , Email verified , password matched then Login to the system.
 2. If User is not registered then register first.
 3. If User is registered but email not verified then Alert –Verify Email first.
 4. If User is registered , email verified but password does not match then Alert –Incorrect Password.

Decision Tree



Decision Table

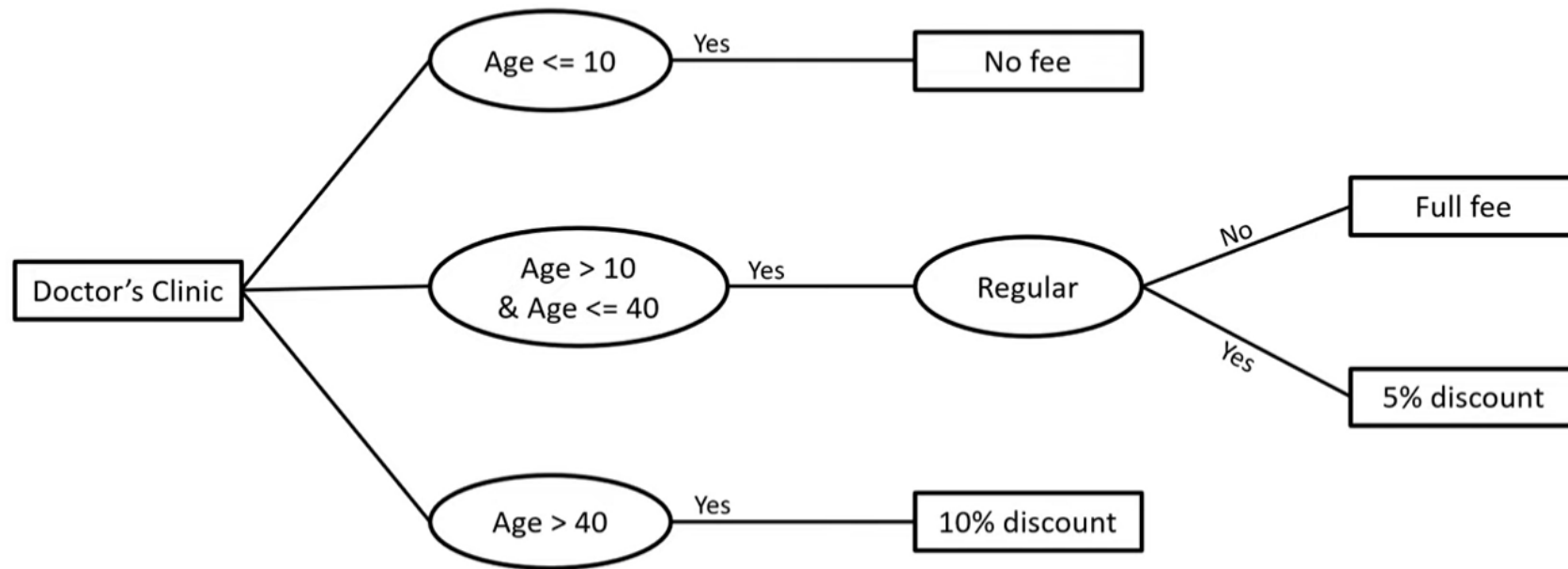
User Login & Register System

Condition	Rule 1	Rule 2	Rule 3	Rule 4
Username	F	T	F	T
Password	F	F	T	T
Action	-	-	-	-
Output	Error	Error	Error	Login

❑ Construct a decision tree and table of Doctor's Clinic based on the following conditions and actions.

1. If the patient's age ≤ 10 , doctor will not charge any amount.
2. If patient's age > 10 & Age ≤ 40 then, if patient is regular then give 5% discount else charge full fee.
3. If Age > 40 then give 10% straight away.

Decision Tree



Decision Table

Doctor's Clinic

Condition	Rule 1	Rule 2	Rule 3	Rule 4
Age \leq 10	Yes	-	-	-
Age $>$ 10 & Age \leq 40	-	-	Yes	Yes
Age $>$ 40	-	Yes	-	-
Regular	-	-	Yes	No
Action				
Fee	No	Yes	Yes	Yes
Discount	-	10%	5%	No