Unit 5: Analysis

Performing Requirements Determination

What is Requirements Determination? A requirement is a vital feature of a new system which may include processing or capturing of data, controlling the activities of business, producing information and supporting the management. Requirements determination involves studying the existing system and gathering details to find out what are the requirements, how it works, and where improvements should be made.

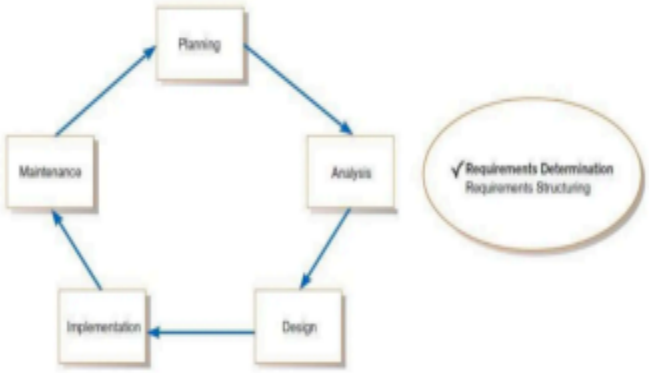


FIGURE 6-1
Systems development life cycle with analysis phase highlighted

This step gathers user requirements through methods like interviews, surveys, and observation. Accurate requirements gathering ensures the final system meets user needs and expectations.

Requirements Anticipation

It predicts the characteristics of a system based on previous experience which include certain problems or features and requirements for a new system. It can lead to analysis of areas that would otherwise go unnoticed by inexperienced analysts. But if shortcuts are taken and bias is introduced in conducting the investigation, then requirement Anticipation can be half baked.

Requirements Investigation

- It is studying the current system and documenting its features for further
- analysis.
- It is at the heart of system analysis where analyst documenting and
- describing system features using fact-finding techniques, prototyping, and computer assisted tools

Requirements Specifications

- It includes the analysis of data which determine the requirement
- specification, description of features for new system, and specifying what
- information requirements will be provided.
- It includes analysis of factual data, identification of essential requirements,
- and selection of Requirement-fulfillment strategies

Traditional for Requirements Determination

Traditional methods of requirements determination are well-established techniques that have been used for many years. They are often structured and rely on formal processes.

1. Interviews

Interviews are a key method for gathering information from users, stakeholders, or subject matter experts. They can be conducted in two main formats: structured and unstructured. In structured interviews, predefined questions are asked in a specific sequence to maintain consistency, allowing for easier comparison of responses across different participants. This approach works well when the desired information is clear, and responses need to be standardized. In contrast, unstructured interviews rely on open-ended questions, allowing participants to express their thoughts more freely. This format is ideal for gaining in-depth insights and uncovering unexpected issues or ideas, especially in the early stages of project planning.

2. Questionnaires and Surveys

Questionnaires and surveys are useful for gathering input from a larger audience, helping analysts identify common themes and trends across a broad group. Standardized questionnaires include fixed questions that can be distributed to many participants, allowing for easy aggregation and analysis of responses. This is especially helpful in organizations with many users, where gaining an overview of general opinions or concerns is beneficial. On the other hand, custom questionnaires are tailored to specific project requirements, enabling the gathering of more targeted insights based on unique needs or issues. This flexibility helps address specialized areas, making custom surveys valuable when a deeper understanding of specific aspects is required.

3. Document Analysis

Document analysis involves reviewing existing documentation, such as user manuals, workflow diagrams, reports, and system logs, to understand the current system and its processes. This technique allows analysts to gain foundational insights into the system without directly involving users, often highlighting workflow patterns, process bottlenecks, or system redundancies. Document analysis serves as a background study, helping analysts identify areas that need further investigation. It's especially useful in understanding historical data and established procedures, providing context that complements information gathered through interviews or surveys.

4. Observation

Observation is a method where analysts watch users perform their tasks to understand how a system or process works in real-life settings. Passive observation involves observing users without interference, allowing the analyst to see natural behaviors, interactions, and task flows. This type of observation can reveal inefficiencies or issues that users may not explicitly mention. In active observation, analysts engage with users by asking questions or clarifying actions as they work. This interactive approach provides a more detailed view of specific tasks and challenges users face, giving analysts a deeper understanding of user experiences and pain points.

5. Focus Groups

Focus groups bring multiple stakeholders or users together in a group discussion format, allowing them to share their insights, opinions, and expectations regarding a system. The group setting encourages participants to build on each other's ideas, which can uncover new perspectives and lead to a richer understanding of user needs. Focus groups are particularly valuable for gathering diverse viewpoints, as they allow for open discussion and feedback that might not emerge in one-on-one interviews. This collaborative environment often fosters creative solutions and provides a more holistic view of user requirements.

Contemporary Methods

Contemporary methods incorporate modern techniques and tools to enhance the process of requirements determination. These methods often involve more collaboration and iterative processes. Contemporary methods in requirements gathering utilize modern techniques to enhance collaboration, flexibility, and ongoing feedback from users and stakeholders. These approaches ensure that requirements are aligned with user needs and expectations through iterative engagement, promoting the development of a more user-centered system.

1. Prototyping

Prototyping involves creating preliminary versions or mock-ups of the system to allow users to interact with and provide feedback on its functionality and design. This early-stage model gives users a tangible sense of the system, helping them clarify their needs and expectations. Feedback gathered through prototyping enables developers to make adjustments before moving to full-scale development, reducing the risk of costly changes later.

2. Joint Application Development (JAD)

Joint Application Development (JAD) is a structured approach that brings together users, developers, and other key stakeholders in facilitated sessions to collaboratively define system requirements. This approach encourages open dialogue and a shared understanding of objectives, reducing the risk of miscommunication and misalignment. JAD sessions enable stakeholders to reach a consensus on requirements and create a foundation for a system that reflects the perspectives and needs of all involved.

3. Use Case Analysis

Use Case Analysis is a technique that involves identifying and detailing the interactions between users and the system, focusing on various scenarios that represent user goals and actions. By breaking down these interactions, developers gain a clear understanding of the system's required functionality. Use cases help clarify system requirements from the user's perspective, ensuring the system supports specific tasks and workflows.

4. User Stories

User stories are simple, informal descriptions of system features written from the end-user's perspective. These short statements outline what the user wants to accomplish and why, without going into technical detail. User stories make it easy to capture specific needs in a straightforward way, allowing developers to design features that directly address user expectations and support essential functionalities.

5. Agile Methods

Agile methods embrace iterative and incremental approaches to gathering and refining requirements throughout the development process. Rather than collecting all requirements upfront, Agile teams work in short cycles, continuously adapting and evolving requirements based on feedback from users and stakeholders. This flexibility enables the team to respond to changing needs and ensures the final system aligns closely with user requirements.

6. Workshops

Workshops are interactive sessions that bring together stakeholders to discuss and document system requirements. These sessions provide a structured environment where participants can collectively explore ideas, address questions, and establish a shared understanding of objectives. Workshops encourage collaboration and thorough documentation, helping to build consensus on requirements early in the project.

7. Brainstorming

Brainstorming sessions are group discussions aimed at generating ideas and gathering requirements through open and creative dialogue. These sessions allow stakeholders to explore various ideas, share perspectives, and identify potential requirements in a collaborative setting. Brainstorming fosters creativity, often uncovering unique insights and requirements that might not emerge in more formalized settings.

8. Personas

Personas are fictional characters created to represent different types of users who will interact with the system. Each persona is based on a specific user profile, including attributes like goals, behaviors, and challenges. By focusing on these personas, teams can better understand diverse user needs, ensuring that requirements are tailored to the expectations of each user group. Personas help guide design and functionality, ensuring the system serves a range of users effectively.

These contemporary methods prioritize user involvement, iterative feedback, and a focus on practical user needs, enhancing the overall process of requirements determination for modern systems.

Radical Methods

Radical methods for requirements determination utilize unconventional and innovative approaches, prioritizing creativity, rapid feedback, and adaptability. These methods engage stakeholders in unique ways, encouraging exploration and collaboration to gather valuable insights for system development.

1. Extreme Programming (XP)

Extreme Programming (XP) focuses on continuous feedback, simplicity, and close customer collaboration throughout the development process. XP emphasizes short development cycles, known as iterations, where user requirements are constantly refined based on user feedback. This approach prioritizes direct involvement from users, fostering an environment where requirements can be adjusted frequently to better meet user needs and ensure high-quality functionality.

2. Design Thinking

Design Thinking is a user-centered methodology that relies on empathy and understanding user needs to guide the design of solutions. Through iterative prototyping, teams explore various solutions, testing ideas with users to gain insights and refine requirements based on real-world feedback. This approach is particularly effective for tackling complex or ambiguous problems, as it combines creativity with structured problem-solving to deliver a system that resonates with users.

3. Lean Startup

The Lean Startup method emphasizes building a minimum viable product (MVP) and iterating based on user feedback. This approach minimizes upfront investment, enabling teams to test core ideas quickly and make data-driven adjustments as needed. By focusing on delivering essential features and gathering immediate user feedback, Lean Startup ensures that resources are allocated effectively and that the product aligns with user expectations from an early stage.

4. Storyboarding

Storyboarding involves creating visual representations of user interactions and scenarios through sketches or illustrations. By mapping out user journeys, storyboarding helps developers and stakeholders visualize the steps users will take within the system. This visual storytelling approach clarifies complex workflows, highlights potential challenges, and fosters a shared understanding of requirements among all team members, ensuring that the system supports intuitive user experiences.

5. Contextual Inquiry

Contextual Inquiry is a technique that involves observing and engaging with users in their actual environment to understand their tasks, workflows, and challenges. By immersing themselves in the user's context, analysts gain insights into real-world needs and pain points that might be overlooked in traditional settings. This method provides a deeper understanding of user behavior, ensuring that requirements align closely with actual work practices and environmental factors.

6. Hackathons

Hackathons are intensive, collaborative events where developers, designers, and stakeholders come together to rapidly prototype solutions. In these sessions, teams work under tight time constraints to brainstorm ideas, develop features, and test solutions. Hackathons foster a spirit of creativity and experimentation, allowing participants to explore new ideas and gather feedback in a short timeframe. This approach is ideal for projects that benefit from innovation and rapid prototyping.

7. Gamification

Gamification applies game design elements, such as points, levels, and rewards, to the requirements gathering process, making it more engaging and motivating for stakeholders. By transforming requirement gathering into a more interactive and enjoyable activity, gamification encourages active participation from users and stakeholders. This approach can lead to a more thorough exploration of requirements, as participants feel motivated to contribute their ideas and insights.

8. Crowdsourcing

Crowdsourcing leverages a large network of people (the "crowd") to gather ideas and requirements, drawing on the diverse perspectives of a broader audience. By opening up the requirements gathering process to a wider group, organizations can access unique insights, innovative ideas, and alternative solutions that may not surface in traditional approaches. Crowdsourcing can be particularly valuable for systems that serve a large and varied user base, as it provides a comprehensive view of different user needs and preference

These radical methods focus on user-centered innovation and adaptability, ensuring that requirements reflect real-world needs and are responsive to user feedback, often leading to more dynamic and well-rounded systems.

Comparing the Methods

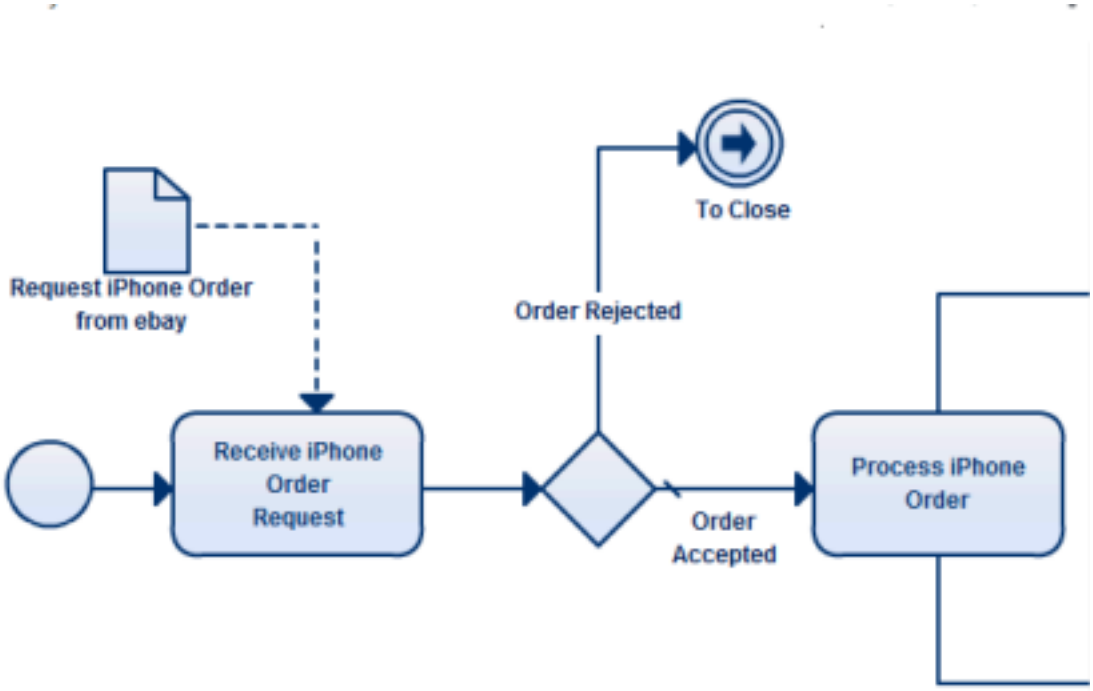
Aspect	Traditional Methods	Contemporary Methods	Radical Methods
Approach	Structured, formal, linear	Flexible, collaborative, iterative	Innovative, creative, adaptive
Key Techniques	Interviews, questionnaires, document analysis, observation	Prototyping, JAD, use case analysis, Agile methods	Extreme Programming (XP), Design Thinking, Lean Startup, Hackathons
Strengths	Clear documentation, reliable processes	Continuous feedback, iterative development	Rapid iterations, user involvement, creativity
Weaknesses	Time-consuming, rigid, less adaptable	Can be resource-intensive, less formal documentation	Lack of formal documentation, scope creep, resource-heavy
Flexibility	Low flexibility, slow to adapt to changes	Medium flexibility, adaptable to evolving needs	High flexibility, quick to adapt to changing requirements
Collaboration	Limited collaboration, often between analysts and users	High collaboration among stakeholders	High collaboration, often with diverse and creative teams
Feedback Cycle	Slow, typically gathered after large tasks are completed	Frequent and ongoing feedback	Real-time feedback, rapid changes based on feedback
Documentation	Extensive, detailed, and formal	Moderate documentation, often less formal	Minimal documentation, relies on prototypes or rapid testing

Process Modeling

Process modeling is the graphical representation of business processes or workflows. Like a flow chart, individual steps of the process are drawn out so there is an end-to-end overview of the tasks in the process within the context of the business environment.

A process model allows visualization of business processes so organizations can better understand their internal business procedures so that they can be managed and made more efficient. This is usually an agile exercise for continuous improvement.

Process modeling is a vital component of process automation, as a process model needs to be created first to define tasks and optimize the workflow before it is automated.



Data Flow Diagram

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually "say" things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That's why DFDs remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or systems.

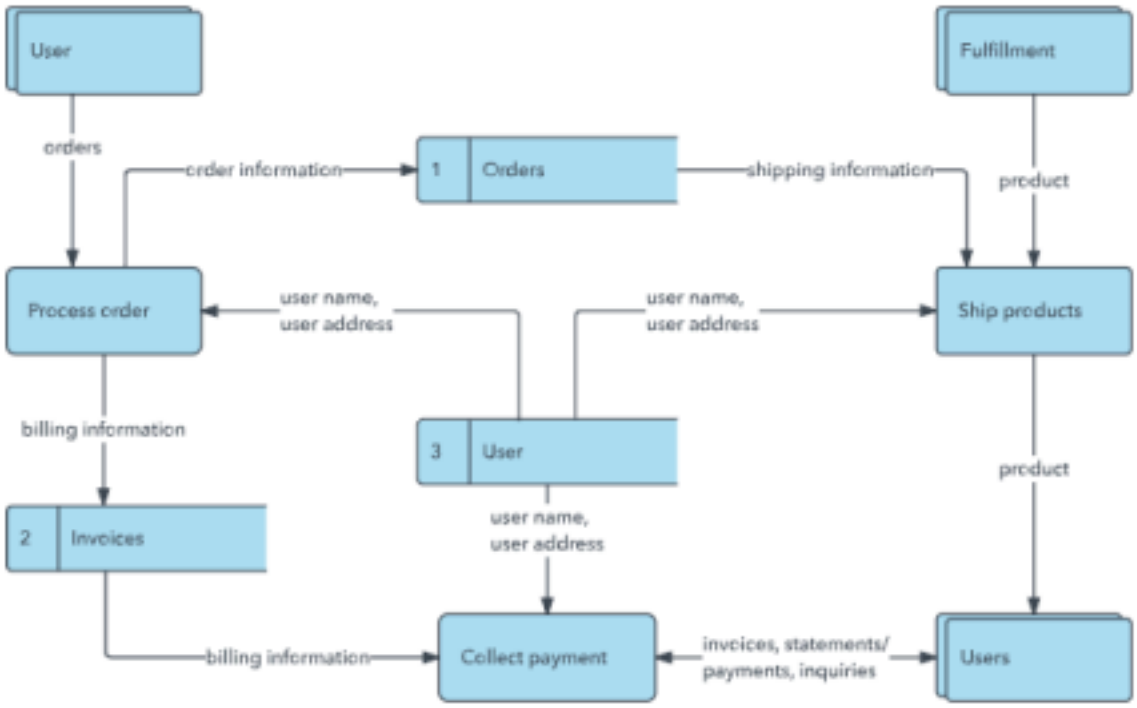
Using any convention's DFD rules or guidelines, the symbols depict the four components of data flow diagrams.

External entity:
an outside system that sends or receives data, communicating with the system being diagrammed. They are the sources and destinations of information entering or leaving the system. They might be an outside organization or person, a computer system or a business system. They are also known as terminators, sources and sinks or actors. They are typically drawn on the edges of the diagram.

Process:
any process that changes the data, producing an output. It might perform computations, or sort data based on logic, or direct the data flow based on business rules. A short label is used to describe the process, such as "Submit payment.

Data store:
files or repositories that hold information for later use, such as a database table or a membership form. Each data store receives a simple label, such as "Orders."

Data flow
: the route that data takes between the external entities, processes and data stores. It portrays the interface between the other components and is shown with arrows, typically labeled with a short data name, like "Billing details."



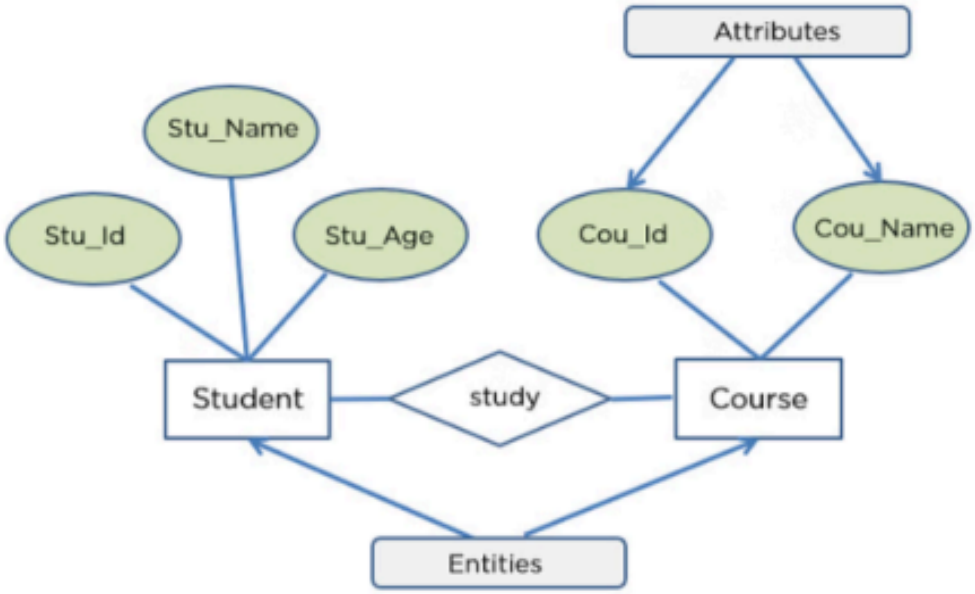
Conceptual Data Modeling and the E-R Model

The conceptual data modeling step (ER approach) involves the classification of entities and attributes first, then identification of generalization hierarchies and other abstractions, and finally the definition of all relationships among entities.

Conceptual data modeling is a crucial phase in database design that focuses on identifying and organizing high-level relationships and attributes of data entities within an organization or system. This modeling technique aims to capture the essence of business requirements and the underlying structure of data without delving into specific implementation details. At its core, conceptual data modeling uses Entity-Relationship (E-R) diagrams to visually represent entities as distinct objects or concepts, attributes as properties describing these entities, and relationships as connections or associations between entities. By creating a conceptual data model, stakeholders and database designers can gain a comprehensive understanding of the data landscape, including key entities and their interrelationships.

ER model

The Entity Relational Model is a model for identifying entities to be represented in the database and representation of how those entities are related. The ER data model specifies enterprise schema that represents the overall logical structure of a database graphically.
The Entity Relationship Diagram explains the relationship among the entities present in the database. ER models are used to model real-world objects like a person, a car, or a company and the relation between these real-world objects. In short, the ER Diagram is the structural format of the database.



SuperType and SubType in Data Modeling:

At times, few entities in a data model may share some common properties (attributes) within themselves apart from having one or more distinct attributes. Based on the attributes, these entities are categorized as Supertype and Subtype entities.

Supertype is an entity type that has got relationship (parent to child relationship) with one or more subtypes and it contains attributes that are common to its subtypes.

Subtypes are subgroups of the supertype entity and have unique attributes, but they will be different from each subtype.Supertypes and Subtypes are parent and child entities respectively and the primary keys of supertype and subtype are always identical.
E.g. People, Bank Account, Insurance, Asset, Liability, Credit Card.

What are business rules?

A business works with processes, which consist of interrelated activities. The functions of purchasing, stock, logistics, finance, sales, and marketing departments, for example, are part of the process of providing customers with a product.

Within these processes, there are some rules to be followed during the execution of activities. They help define HOW the operations will be performed, WHO will perform them, WHEN, WHERE, and WHY, according to BPM CBOK.
Business rules can be said to be limits imposed on operations so that they respect company policies and objectives. In general, business rules should:

- Have only one function, being indivisible and simple.
- Be complete, with beginning, middle, and end.
- Allow for measuring and tracking.
- Comply with current legislation.
- Be updated and always revised.
- Reflect the organization's policy and values.
- Be understandable for employees and stakeholders.

Packaged Conceptual Data Models.

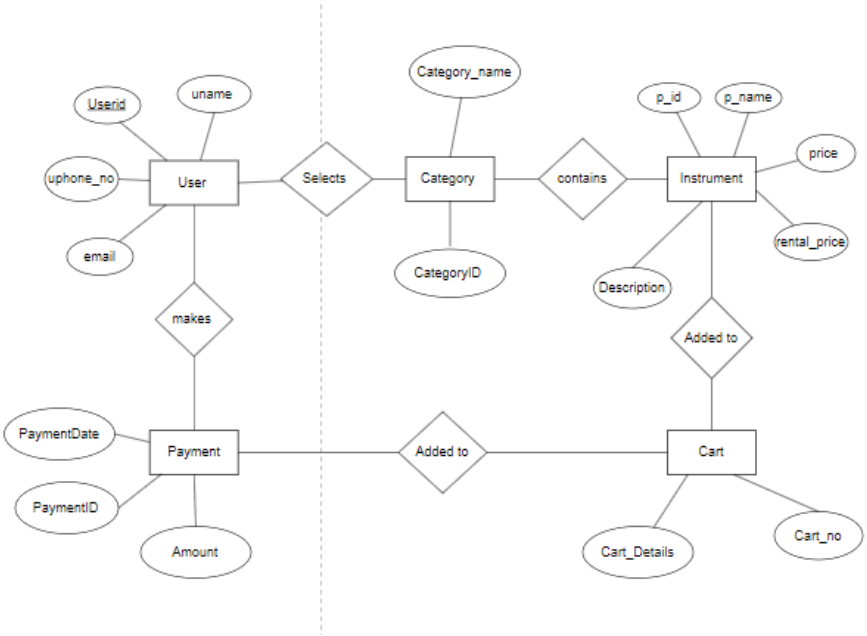
There are three different types of data models – conceptual, logical and physical. Each data model has a specific purpose, which is primarily defined by the level of operational detail. Conceptual data models are built at the first stage of the data modeling process. They provide a summary-level perspective, omitting finer details in favor of a more readily digestible format.

As the name suggests, conceptual data modeling is most relevant at the conceptual stage, when an organization drafts a rough plan with the intention to work out the finer details later. Usually created by data architects and business stakeholders, conceptual data models give stakeholders an easily digestible snapshot of the relevant concepts or entities and the relationships between them. By communicating the model in a way that is relevant to stakeholders who aren't necessarily tech- and/or detail-oriented, modelers are more likely to get support for their projects. The erwin platform was built with fostering this sort of collaboration in mind.

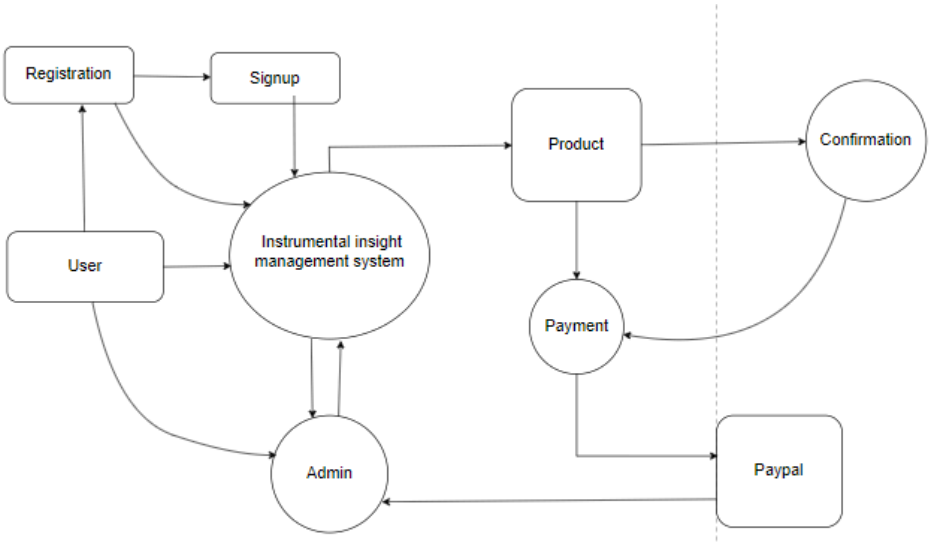
The aim of a conceptual data model is to provide a data-centric perspective of the organization by documenting how different business entities relate to one another. This is often achieved via entity relationship diagrams (ERD) and/or object-role models (ORM). Unlike logical and physical
data models, conceptual data models are technology- and application-independent. This means they are untethered from the reality and context of systems and processes currently in place.

Assignments given in classroom

Er diagram



Data flow diagram



D1 Data flow Diagram