VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : COMPUTER SCIENCE AND ENGINEERING

NAME OF THE LABORATORY : SE LAB

Name: Lokesh Naidu Roll No: 1602-22-733-022 Page No :_____

Experiment – II: Data Flow Diagrams

1. Problem Statement: To design data-flow diagram for software requirements

SOFTWARE REQUIREMENTS: Ms Word.

Data flow diagram (DFD)

A Data Flow Diagram (DFD) is a visual tool that represents information flow within a system. Like a picture, a clear DFD effectively communicates system requirements graphically. It illustrates how data enters, changes, is stored, and leaves the system, defining its scope and boundaries. DFDs serve as a communication tool, especially between analysts and system stakeholders, often initiating system redesign. They typically start with a high-level "context diagram" (Level 0) and are refined through lower-level diagrams (Level 1, 2, and sometimes 3), showing decomposed functions. The detail level for each function is dependent on its complexity, making the DFD a flexible tool.

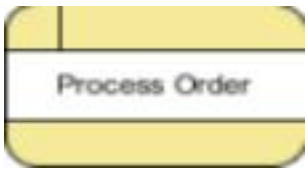
Diagram Notations :

External Entity

An external entity can represent a human, system or subsystem. It is where certain data comes from or goes to. It is external to the system we study, in terms of the business process. For this reason, people used to draw external entities on the edge of a diagram.

Process

A process is a business activity or function where the manipulation and transformation of data takes place. A process can be decomposed to finer level of details, for representing how data is being processed within the process



Data Store

A data store represents the storage of persistent data required and/or produced by the process. Here are some examples of data stores: membership forms, database table, etc.

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

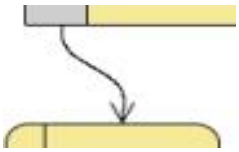
DEPARTMENT OF : COMPUTER SCIENCE AND ENGINEERING

NAME OF THE LABORATORY : SE LAB

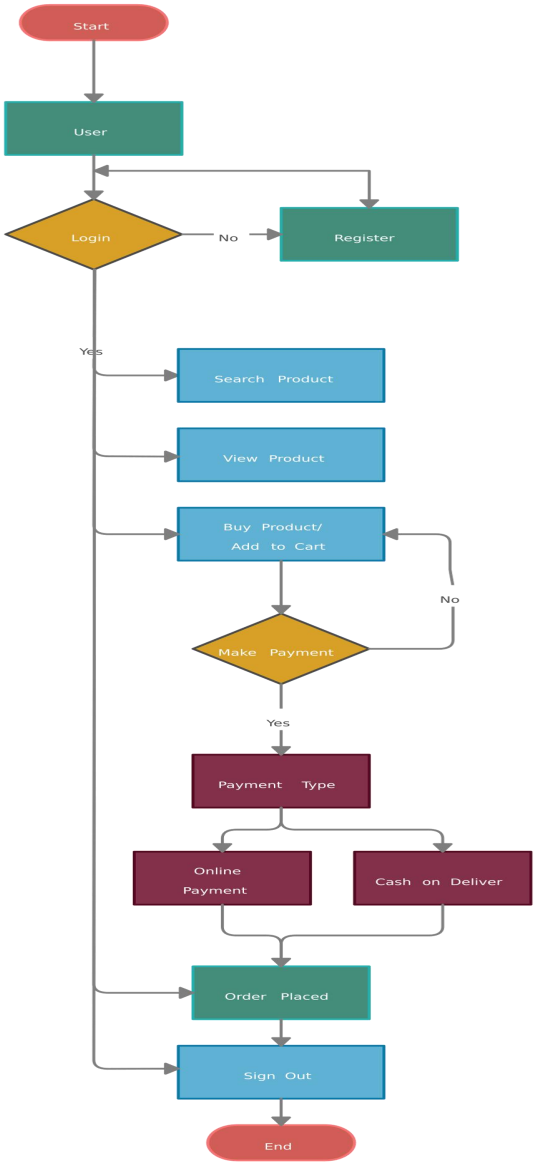
Name: Lokesh Naidu Roll No: 1602-22-733-022 Page No :_____

Data Flow

A data flow represents the flow of information, with its direction represented by an arrow head that shows at the end(s) of flow connector.



DFD for B2B E commerce Management System:



VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University**)**

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : COMPUTER SCIENCE AND ENGINEERING

NAME OF THE LABORATORY : SE LAB

Name: Lokesh Naidu Roll No: 1602-22-733-022 Page No :_____

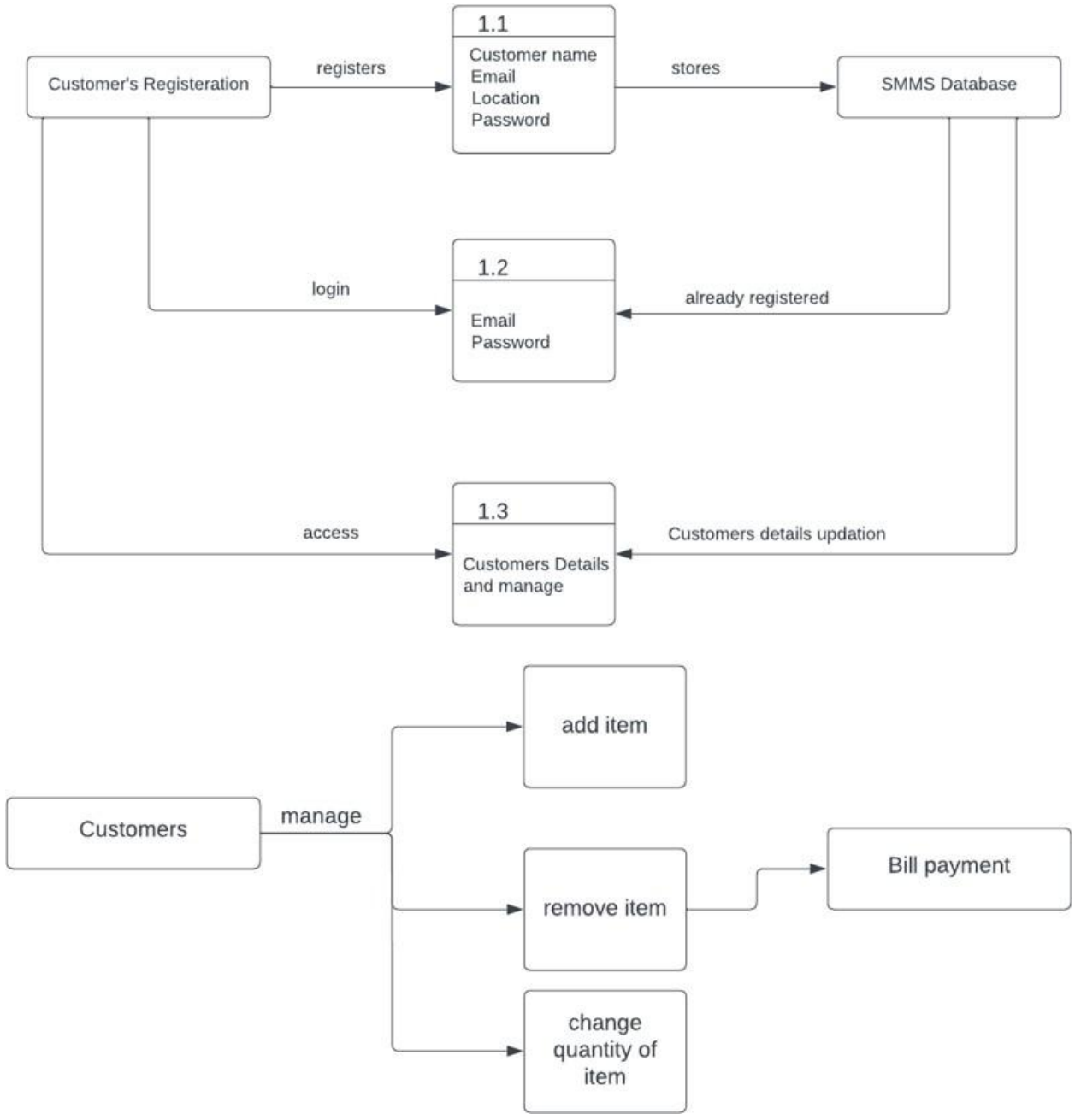


Figure 2.3: Level 2

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : COMPUTER SCIENCE AND ENGINEERING

NAME OF THE LABORATORY : SE LAB

Name: Lokesh Naidu Roll No: 1602-22-733-022 Page No :_____

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : COMPUTER SCIENCE AND ENGINEERING

NAME OF THE LABORATORY : SE LAB

Name: Lokesh Naidu Roll No: 1602-22-733-022 Page No :_____

Experiment – III: Functional Decomposition and Structuring

1. Problem Statement: To design functional decomposition diagram for software requirements

SOFTWARE REQUIREMENTS: Ms Word.

Theory

Decomposition

Decomposition is the process of breaking down complex entities, such as processes, technology, or business problems, into smaller, more manageable sub-parts. This process continues until a discreet and understandable structure is achieved. It represents a fundamental analytical technique frequently used by business analysts. Common items for decomposition include systems, processes, goals, and requirements. A key characteristic is that the sub-components should fully describe the parent component. Hierarchical diagrams, such as Functional Decomposition Diagrams, are common outputs.

Types of Decomposition

There are three main types of decomposition:

1. **Structural Decomposition:** This focuses on the "what," including physical components, logical objects, and data elements. It uses nouns and adjectives for descriptions.
2. **Behavioural/Functional Decomposition:** This centers on the "how" and "when," considering actions, processes, and controls. Verbs and adverbs are used for descriptions. This is a core technique in business analysis.
3. **Goal Decomposition:** This addresses the "why," breaking down strategic, project, or personal goals into specific objectives.

Cohesion and Coupling

- **Cohesion:** This measures the similarity of functions within a group. High cohesion is desirable within a sub-group.
- **Coupling:** This measures the interdependence between functions. Low coupling is desired between sub-groups.
- **Effective Decomposition:** An effective decomposition strives for high cohesion and coupling within sub-units, and low cohesion and coupling between them.

Why Use Decomposition?

Decomposition assists in several areas:

- It facilitates the analysis of complex processes.
- It aids in defining project scope.
- It assists in eliciting and analyzing requirements.

- It translates high-level goals into concrete requirements.
- It helps understand the functionality of undocumented systems.
- It contributes to strategic planning.

Examples of Use:

- Decomposition can break down HR into sub-functions like hiring and termination.
- System functions, such as login, can be decomposed into user ID and password validation.
- Project goals can be broken down into measurable objectives.

Steps for Functional Decomposition

1. **Plan:** Define the focus area, the level of detail required, and the documentation method.
2. **Identify Main Components:** Collaborate with stakeholders to identify primary components.
3. **Identify Sub-Components:** Break down main components into smaller parts.
4. **Iterate:** Review components for further decomposition if necessary.
5. **Check Completeness:** Ensure all relevant aspects are captured and correctly grouped.
6. **Refine:** Repeat the process until all stakeholders are satisfied.
7. **Document:** If necessary, provide consistent documentation for each component.

Documentation Methods

- Functional Decomposition Diagrams provide a visual representation.
- Outlines offer a structured textual approach.
- Tables can be used for a tabular format of the decomposition.

Advantages

- Decomposition is intuitive and easy to understand.
- It helps to identify duplicate activities.
- It simplifies complex systems into manageable pieces.

Disadvantages

- The approach can be internally focused, potentially overlooking external factors.
- There is a risk of over-detailing the components.
- It may be possible to miss crucial components during the decomposition.
- Confusion may arise between functional diagrams and organizational charts.

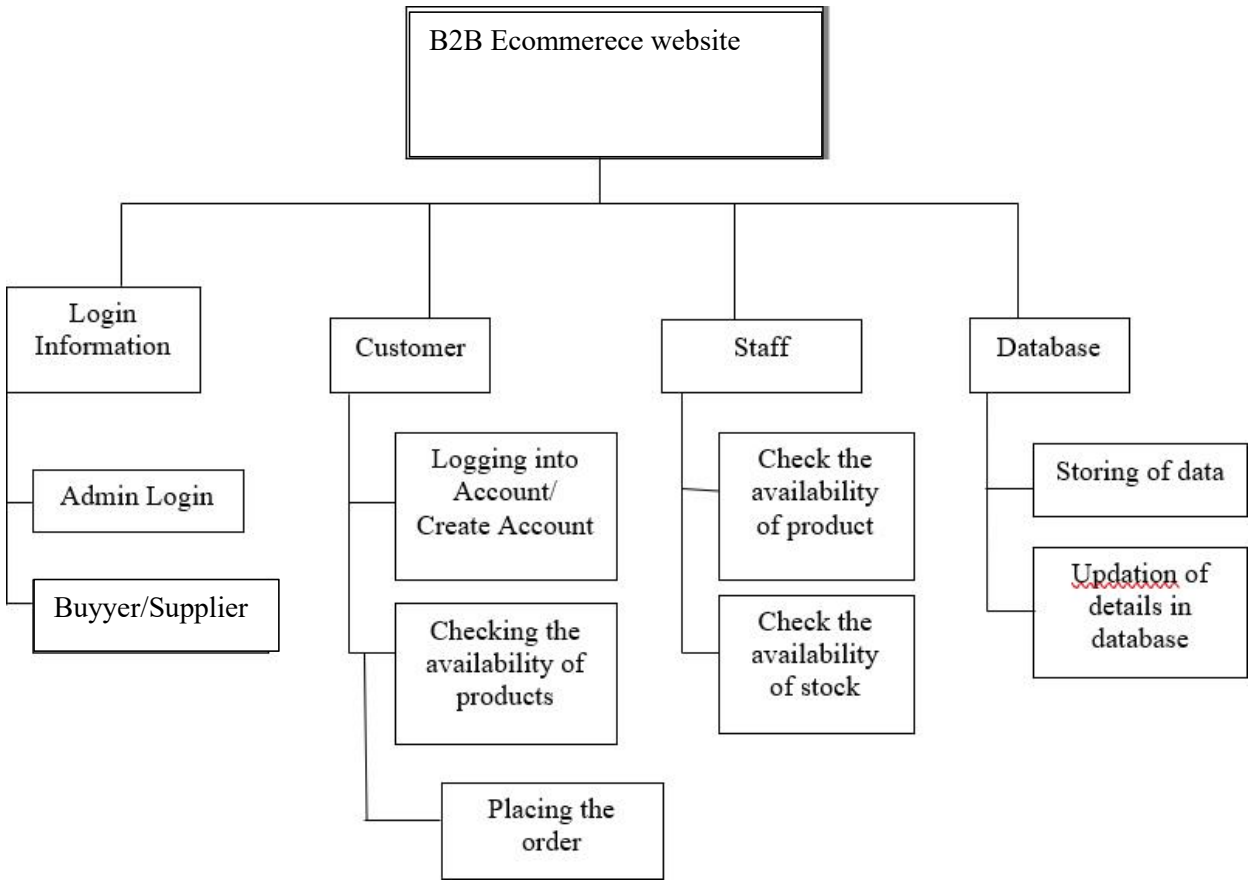


Figure 3.1:Functional Decomposition for B2BEC

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : COMPUTER SCIENCE AND ENGINEERING

NAME OF THE LABORATORY : SE LAB

Name: Lokesh Naidu Roll No: 1602-22-733-022 Page No :_____

Experiment – IV: Use Case Diagram

1. Problem Statement: To design Use Case diagram for software requirements of B2B E commerce Management System

SOFTWARE REQUIREMENTS: Rational Rose

Theory

Use case diagram is a behavioural **UML diagram type** and frequently used to analyze various systems. They enable you to visualize the different types of roles in a system and how those roles interact with the system.

Importance of Use Case Diagrams

As mentioned before use case diagram are used to gather a usage requirement of a system. Depending on your requirement you can use that data in different ways. Below are few ways to use them.

- **To identify functions and how roles interact with them** – The primary purpose of use case diagrams.
- **For a high level view of the system** – Especially useful when presenting to managers or stakeholders. You can highlight the roles that interact with the system and the functionality provided by the system without going deep into inner workings of the system.
- **To identify internal and external factors** – This might sound simple but in large complex projects a system can be identified as an external role in another use case.

Use Case Diagram Objects:

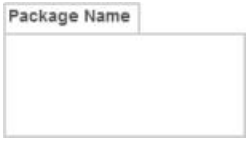
Use case diagrams consist of **four** main objects:

- **Actor:** Represents an entity (person, organization, or external system) that performs a role within the system. They are typically represented as a stick figure.

- **Use Case**: Depicts a specific function or action within the system, visualized as an oval shape with a name that describes the function.

- **System:** This element is used to define the scope of the use case. It is drawn as a rectangle, and is optional but helpful when visualizing larger systems.

- **Package:** An optional element that is useful in complex diagrams. Similar to class diagrams, packages group together related use cases. They are represented with a folder-like image.



Relationships in Use Case Diagrams:

Five types of relationships exist within use case diagrams:

- **Association:** A connection between an actor and a use case.
- **Generalization (Actor):** Represents inheritance between actors.
- **Extend:** A relationship between two use cases, where one use case extends the behavior of another conditionally.
- **Include**: A relationship between two use cases, where one use case always includes the behavior of another.
- **Generalization (Use Case):** Represents inheritance between use cases.

Creating a Use Case Diagram: Example Using an Airline Reservation System (ARS):

1. **Identifying Actors:** Actors are entities interacting with the system. In an ARS, the primary actor is a passenger, with further distinctions like "new passenger" and "existing passenger."
2. **Identifying Use Cases:** Determine what actors need from the system. In an ARS, passengers need to check availability, reserve, and cancel tickets, among other functions. These are potential use cases. Top-level use cases should provide a complete set of functions.
3. **Include Relationship:** Look for common functionality that can be reused across multiple use cases. Extract these common actions into separate use cases and connect with the "include" relationship.
4. **Generalize Actors and Use Cases:** Generalize actors when they have similar interactions and then special actions unique to the them. This allows inheritance of functionalities. A similar method is used for generalizing use cases, like "Make Payment", to be more specific such as "Pay by Credit Card", "Pay by Cash", etc.
5. **Optional Functions and Extension**: Use the "extend" relationship for optional or supplementary functions. The base use case should always function independently, even if the extended use case is not called.

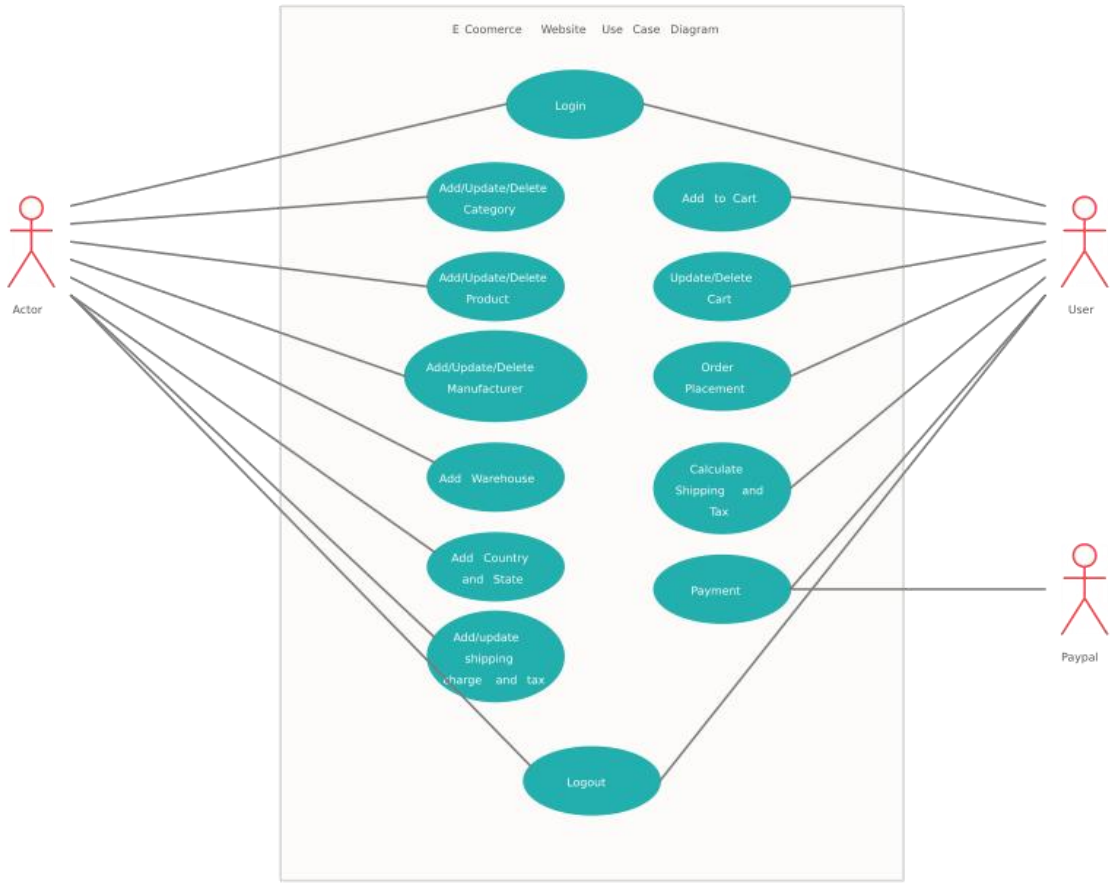


Fig 4.1
Use case Diagram for B2BEC

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : COMPUTER SCIENCE AND ENGINEERING

NAME OF THE LABORATORY : SE LAB

Name: Lokesh Naidu Roll No: 1602-22-733-022 Page No :_____

Fig:
Use Case Diagram

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : COMPUTER SCIENCE AND ENGINEERING

NAME OF THE LABORATORY : SE LAB

Name: Lokesh Naidu Roll No: 1602-22-733-022 Page No :_____

Experiment – V: Interaction Diagrams

1. Sequence Diagram

Problem Statement: To draw Interaction diagram of B2B Ecommerce Management System

System Requirements:

SOFTWARE: Rational rose

Theory:

DEFINITION:

· A sequence diagram is a graphical view of a scenario that shows object interaction in a time-based sequence¾what happens first, what happens next.

· Sequence diagrams establish the roles of objects and help provide essential information to determine class responsibilities and interfaces.

· This type of diagram is best used during early analysis phases in design because they are simple and easy to comprehend.

· Sequence diagrams are normally associated with use cases.

· Sequence diagrams are closely related to collaboration diagrams and both are alternate representations of an interaction.

· There are two main differences between sequence and collaboration diagrams:

· Sequence diagrams show time-based object interaction

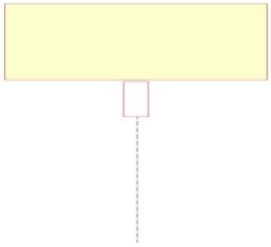
· While collaboration diagrams show how objects associate with each other.

CONTENTS:

A sequence diagram has two dimensions: typically, vertical placement represents time and horizontal placement represents different objects.

The following tools located on the sequence diagram toolbox enable we to model sequence diagrams:

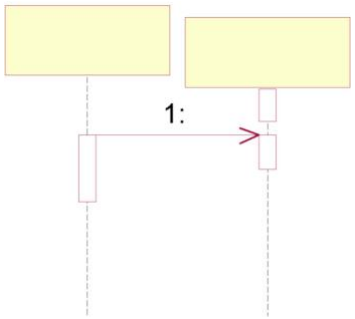
· Object: An object has state, behavior, and identity. The structure and behavior of similar objects are defined in their common class. Each object in a diagram indicates some instance of a class. An object that is not named is referred to as a class instance.



If we use the same name for several object icons appearing in the same collaboration or activity diagram, they are assumed to represent the same object; otherwise, each object icon represents a distinct object. Object icons appearing in different diagrams denote different objects, even if their names are identical. Objects can be named three different ways: object name, object name and class, or just by the class name itself.

. Message Icons: A message icon represents the communication between objects indicating that an action will follow. The message icon is a horizontal, solid arrow connecting two lifelines together. The following message icons show three different ways a message icon can

appear: message icon only, message icon with sequence number, and message icon with sequence number and message label.



Each message icon represents a message passed between two objects, and indicates the direction of message is going.

A message icon in a collaboration diagram can represent multiple messages. A message icon in a sequence diagram represents exactly one message. · Focus of Control: Focus of Control (FOC) is an advanced notational technique that enhances sequence diagrams. It shows the period of time during which an object is performing an action, either directly or through an underlying procedure.FOC is portrayed through narrow rectangles that adorn lifelines. The length of an FOC indicates the amount of time it takes for a message to be performed. When we move a message vertically, each dependent message will move vertically as well. Also, we can move a FOC vertically off of the source FOC to make it detached and independent. An illustration of a sequence diagram with FOC notation follows:

· Message to Self: A Message to Self is a tool that sends a message from one object back to the same object. It does not involve other objects because the message returns to the same object. The sender of a message is the same as the receiver.

The following example shows a Message to Self labeled, 2. Turn on Light, in a sequence diagram:

· Note: A note captures the assumptions and decisions applied during analysis and design. Notes may contain any information, including plain text, fragments of code, or references to other documents. Notes are also used as a means of linking diagrams. A note holds an unlimited amount of text and can be sized accordingly.

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : COMPUTER SCIENCE AND ENGINEERING

NAME OF THE LABORATORY : SE LAB

Name: Lokesh Naidu Roll No: 1602-22-733-022 Page No :_____

Notes behave like labels. They are available on all diagram toolboxes, but they are not considered part of the model. Notes may be deleted like any other item on a diagram.
·Note Anchor: A note anchor connects a note to the element that it affects. To draw a note anchor, place a note on the diagram and connect the note to an element with the note anchor icon.

DIAGRAM:

e

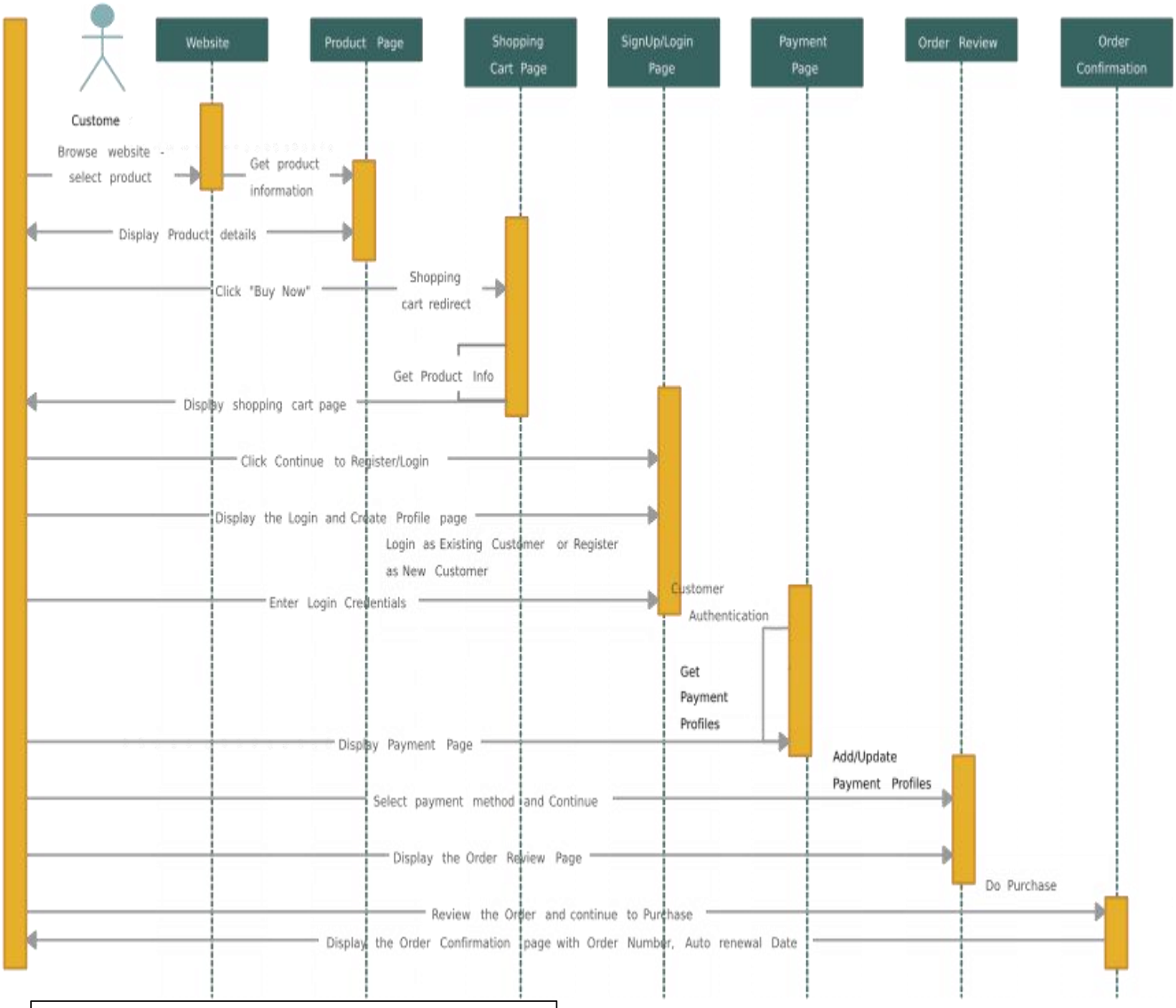


Fig 5.1
Sequence diagram for B2BEC

Figure 5.1 Sequence Diagrams

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad – 500 031.

DEPARTMENT OF : COMPUTER SCIENCE AND ENGINEERING

NAME OF THE LABORATORY : SE LAB

Name: Lokesh Naidu Roll No: 1602-22-733-022 Page No :_____

2. Collaboration Diagram

Problem Statement: To draw Collaboration diagram of B2B E commerce Management System

System Requirements:

SOFTWARE: Rational rose

Theory:

Collaboration and sequence diagrams are two ways to represent interactions. Collaboration diagrams show the order of messages that implement an operation or transaction and focus on the object relationships. Sequence diagrams display object interaction in a time-based order. Both show objects, links, and messages, and can include class instances. Collaboration diagrams visualize interactions between objects within a model's logical package. The Create Collaboration Diagram command generates a diagram from sequence diagram data and vice versa.

Collaboration Diagram Contents:

- **Object:** Objects, like class instances, have state, behavior, and identity and are represented with an underlined name. They can be named in three ways: object name, object name and class, or class name alone. Multiple instances of the same class can be shown using a special icon, and their concurrency and persistence can be shown as adornments.

- **Link:** A link represents an instance of an association between two objects and supports messages between the objects. The visibility of one object to another can be displayed as an adornment on the link, with options for global, parameter, field, or local visibility.
- **Message/Event:** A message is communication that triggers an event between two objects and conveys information. Messages are shown as arrows and can have synchronization types such as simple, synchronous, balking, timeout, asynchronous, procedure call, or return. A single message icon in a collaboration diagram can represent multiple messages, and the type can be modified through the message specification. Collaboration diagrams have a message-to-self icon for self-referencing messages.
    - The collaboration diagram has a forward arrow for sending message from client to supplier and a reverse arrow for sending message from supplier to client.
- **Message Icons:** Message icons are horizontal, solid arrows that link objects and indicates the direction of message flow, and a message icon in a collaboration diagram can represent multiple messages. They can include sequence numbers, message labels, and operation names and their synchronization can be modified.

You can enhance a message by attaching a script to the message. The script will remain aligned with the message regardless of any changes in the position of the message. Scripts are attached to a message by clicking Edit > Attach Script

Changing the Synchronization Type

The default message synchronization icon is "simple." To change the synchronization type, select a type from the synchronization field in the message specification. The following synchronization types are supported:

· Simple
· Synchronous
· Balking
· Timeout
· Asynchronous
· Procedure Call
· Return

DIAGRAM:

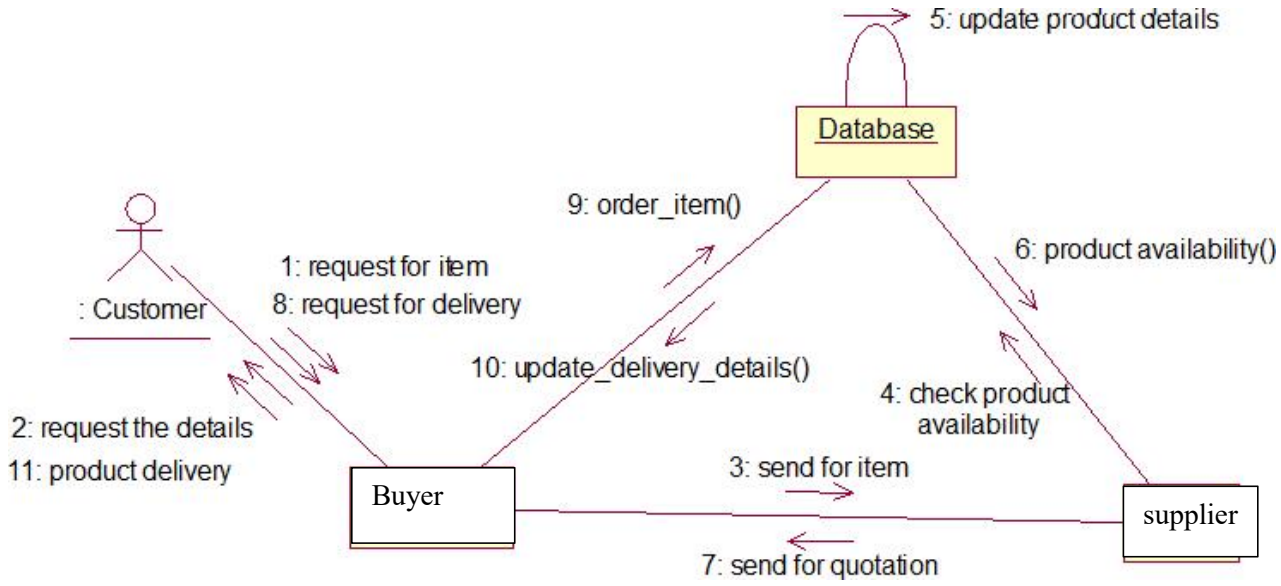


Figure 5.2 Collaboration Diagram for B2BEC