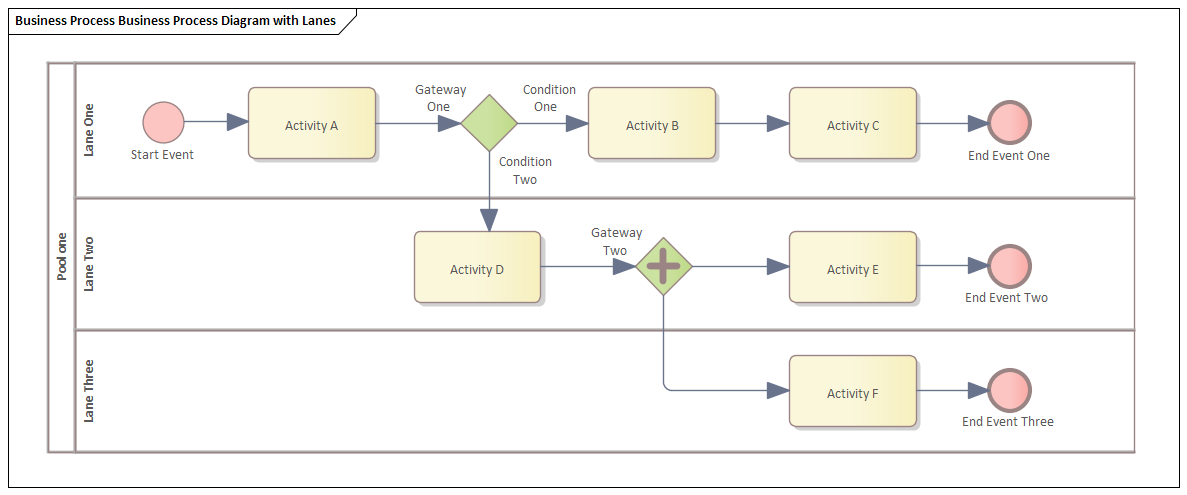
**1. Business Process Model and Notation (BPMN) Diagram**

**Definition**: BPMN diagrams are used to model the steps of a business process from start to finish. They provide a standard way to visualize internal business procedures with a graphical notation and are widely used to improve efficiency, identify bottlenecks, and align processes.

**Example**:

* **Scenario**: A customer placing an online order.
* **Diagram Elements**: Start Event, Task (e.g., “Enter Order”), Gateway (e.g., “Payment Confirmation”), End Event.
* **Flow**:
  + Customer starts the order (Start Event).
  + Fills in order details (Task).
  + Proceeds to payment (Task).
  + Payment is verified (Gateway).
  + If payment is successful, order confirmation is sent (End Event).
  + If payment fails, the customer is asked to retry (Gateway loops back to payment).



What are the four categories of BPMN?

BPMN symbols fall under four main areas: flow objects, connecting objects, artifacts, and swimlanes.

**Flow objects**

BPMN flow symbols are the elements that connect and form the process flow. They include the following:

Events - An event represents something that happens in the process. received” event could be represented by a pile of coins or a dollar bill.

Activities- Activities appear in a BPMN diagram as a rounded rectangle. They represent the parts of the process that require action

**Connecting objects**

These are the connection lines and arrows that symbolize how objects relate to each other and how the process flows from one activity to the next. Here are some of the different connecting objects:

**Artifacts**

Artifacts provide supporting information about your business processes. They allow you to describe elements of the process and help to categorize and organize tasks.

**Swimlanes**

Swimlanes are a key part of a BPMN diagram’s structure. They’re typically horizontal, rectangular boxes that separate different stages of the process.

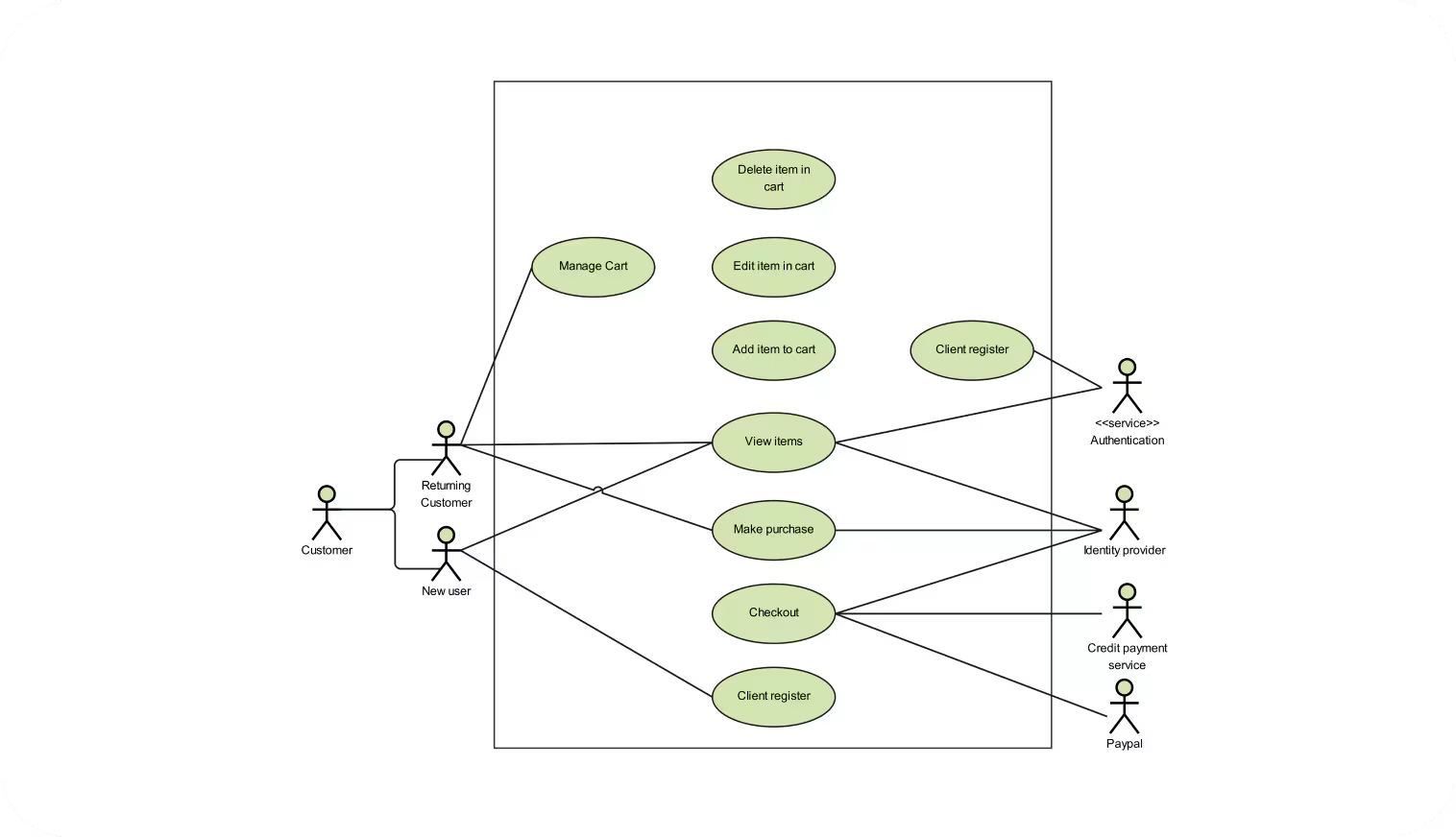
**2. Use Case Diagram**

**Definition**: Use Case Diagrams depict the interactions between users (actors) and the system to achieve a goal. They provide a high-level view of what the system will do and who will interact with it.

**Example**:

* **Scenario**: An e-commerce website.
* **Actors**: Customer, Admin.
* **Use Cases**: “Search for Products,” “Place Order,” “Manage Inventory.”
* **Diagram**:
  + Customer -> (Search for Products).
  + Customer -> (Place Order).
  + Admin -> (Manage Inventory).
  + Relationships: Actors interact with their respective use cases.

AXPNKCKNDCDK CKJ CEJKEDHJ JD



Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system.

Note, that UML 2.0 to 2.4 specifications also described use case diagram as a specialization of a class diagram, and class diagram is a structure diagram.

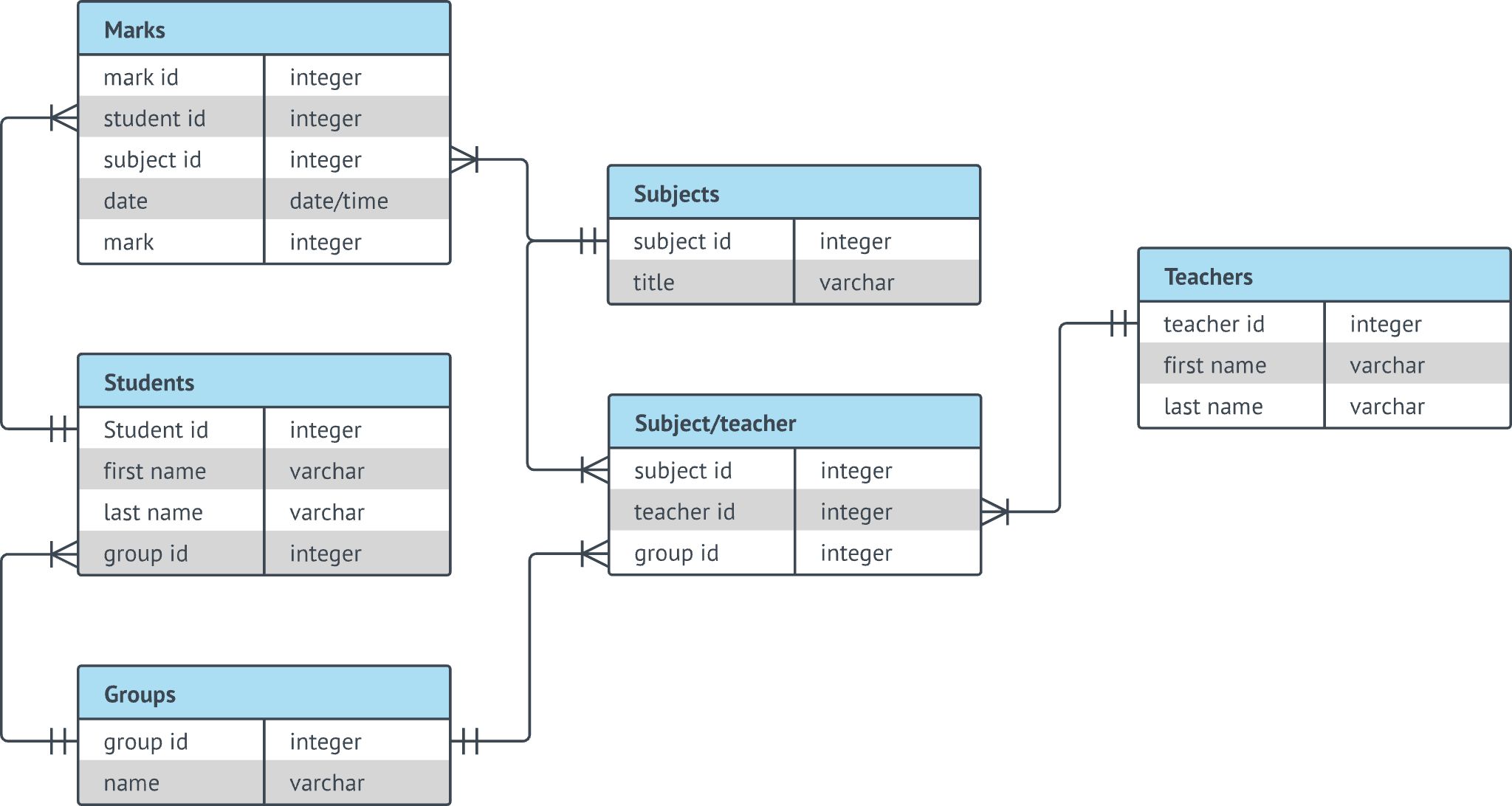
Use case diagrams are in fact twofold - they are both behavior diagrams, because they describe behavior of the system, and they are also structure diagrams - as a special case of class diagrams where classifiers are restricted to be either actors or use cases related to each other with associations.

**3. Entity-Relationship (ER) Diagram**

**Definition**: ER Diagrams show the structure of a database by illustrating entities, their attributes, and the relationships between them. An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system.

**Example**:

* **Scenario**: A university database.
* **Entities**: Student, Course, Enrollment.
* **Attributes**: Student (ID, Name, Age), Course (ID, Title, Credits), Enrollment (Student ID, Course ID, Date).
* **Relationships**:
  + Student “enrolls in” Course (many-to-many).
  + ER Diagram links Student and Course via Enrollment.



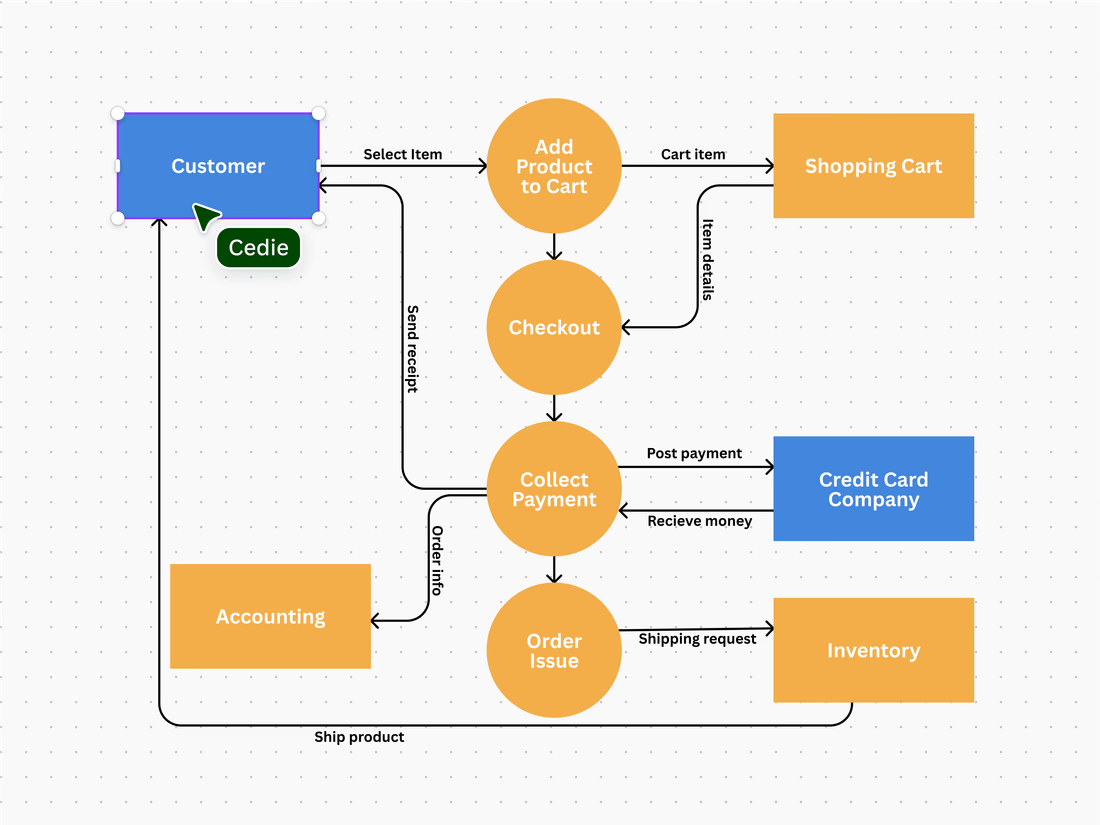
ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.

**4. Data Flow Diagram (DFD)**

**Definition**: DFDs represent how data moves through a system, illustrating processes, data stores, data flows, and external entities. They help identify data inputs, outputs, and processes within a system.

**Example**:

* **Scenario**: A payroll system.
* **Processes**: “Calculate Pay,” “Generate Payslip.”
* **Data Stores**: Employee Records, Payslip Archive.
* **External Entities**: Employee, Bank.
* **Data Flows**:
  + Employee provides work hours (input).
  + System calculates pay (Process).
  + Payslip is generated and stored (Data Store).
  + Payslip is sent to Bank (Output).

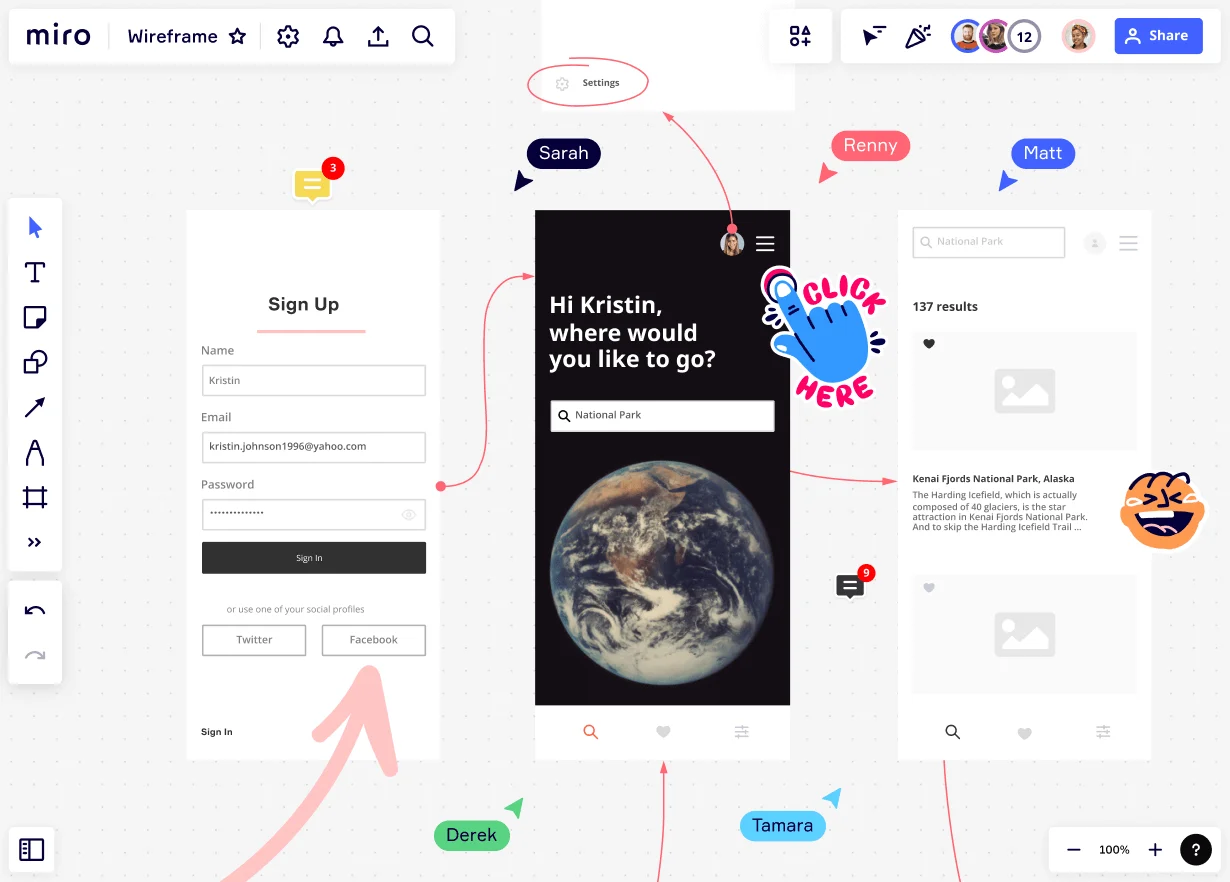


**5. Wireframe or Mockup Diagram**

**Definition**: Wireframes are low-fidelity, simplified layouts of web or app interfaces. They focus on functionality and structure rather than visual design.

**Example**:

* **Scenario**: A new landing page for a website.
* **Elements**: Header, Navigation Bar, Main Content Area, Footer.
* **Wireframe**:
  + Placeholder for logo and navigation at the top.
  + Main content section with text boxes and image placeholders.
  + Footer with contact details and social media links.



A wireframe is a visual diagram that outlines the skeletal framework of a website, app, or other digital product. Sometimes known as a page schematic or screen blueprint, it demonstrates how elements relate to each other and how they’re structured. By using a

wireframe tool

to build a blueprint, designers can create consistent layouts that meet user needs.

Wireframing is a top-level process.

User experience (UX) designers

often use it to map out the design and layout of their work without going into too much detail. It’s the first stage of the design process before it is fleshed out to add more detail. As such, a wireframe primarily focuses on functionalities and intended behaviors, but not color schemes or final stylistic choices.

**6. Swimlane Diagram (Cross-Functional Flowchart)**

**Definition**: Swimlane diagrams split tasks into lanes to represent different actors, departments, or systems, showing responsibilities and interactions in a process.

**Example**:

* **Scenario**: Processing a loan application.
* **Lanes**: Customer, Loan Officer, Underwriting Department, Legal Department.
* **Flow**:
  + Customer submits application (Customer Lane).
  + Loan Officer reviews application (Loan Officer Lane).
  + Application is evaluated by Underwriting (Underwriting Lane).
  + Contract is prepared and signed (Legal Lane).

A swim lane diagram, sometimes called a cross-functional diagram, is a process

flowchart that provides richer information on who does what. It can also be

expanded to show times—when tasks are done and how long they take. As

seen in a swim lane diagram of ARC Community Services’ intake process

(FIGURE 4), the visual metaphor is a swimming pool, with each participant in the

process assigned to “lanes.”

The steps to create a swim lane diagram follow:

1. Focus on a specific process, and put the title of your diagram on top.

2. Enumerate the people involved in this process and assign them to rows, typically

beginning with the customer on the top row.

3. Create the process flowchart, drawing processes and decisions made, as well as

arrows that indicate the process flow.

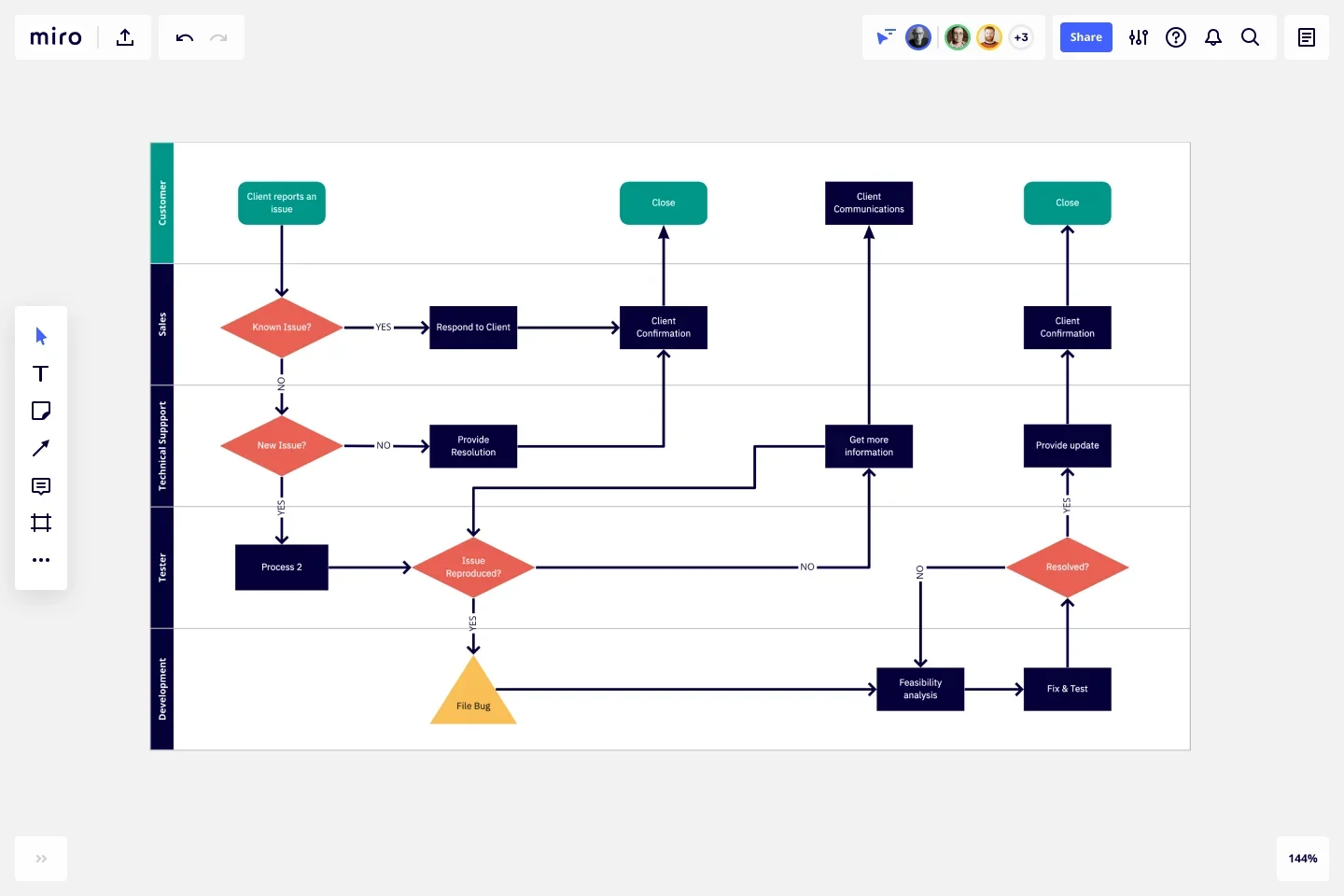
4. If the diagram is too complex, break it up into its components. As seen in

FIGURE 4, the diagram indicates phases or sub-processes (i.e., pre-intake,

intake, pre-treatment, treatment, etc.).

5. If possible, indicate times for each node on the bottom of the diagram. Compute

cumulative totals—the total time elapsed—within each phase.



**7. Gantt Chart**

**Definition**: Gantt charts are project management tools that illustrate the start and finish dates of elements within a project. They help in scheduling, tracking progress, and resource allocation.

**Example**:

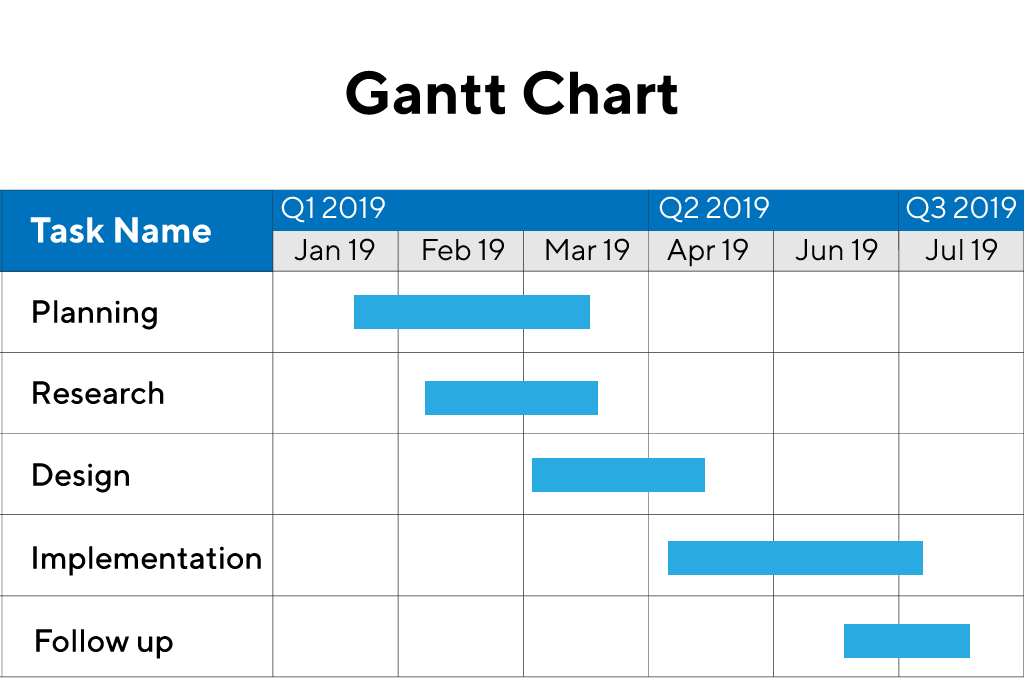
* **Scenario**: Software Development Project.
* **Tasks**: Requirements Gathering, Design, Development, Testing, Deployment.
* **Timeline**:
  + Requirements (Weeks 1-2).
  + Design (Weeks 3-4).
  + Development (Weeks 5-8).
  + Testing (Weeks 9-10).
  + Deployment (Week 11).
* Each task is represented as a bar, aligned with the timeline.

A Gantt chart, or harmonogram, is a bar chart that graphically illustrates a schedule for planning, coordinating, and tracking specific tasks related to a single project.

Even a century after its introduction, the Gantt chart continues to be an important tool in project and program management. Described in 1999 as “one of the most widely used management tools for project scheduling and control.”

The anatomy is straightforward: Tasks appear on the vertical axis, and time intervals, or activity duration, appear along the horizontal axis.

These charts are commonly used by teams in construction, consulting, marketing, manufacturing, HR, software development, and event planning.

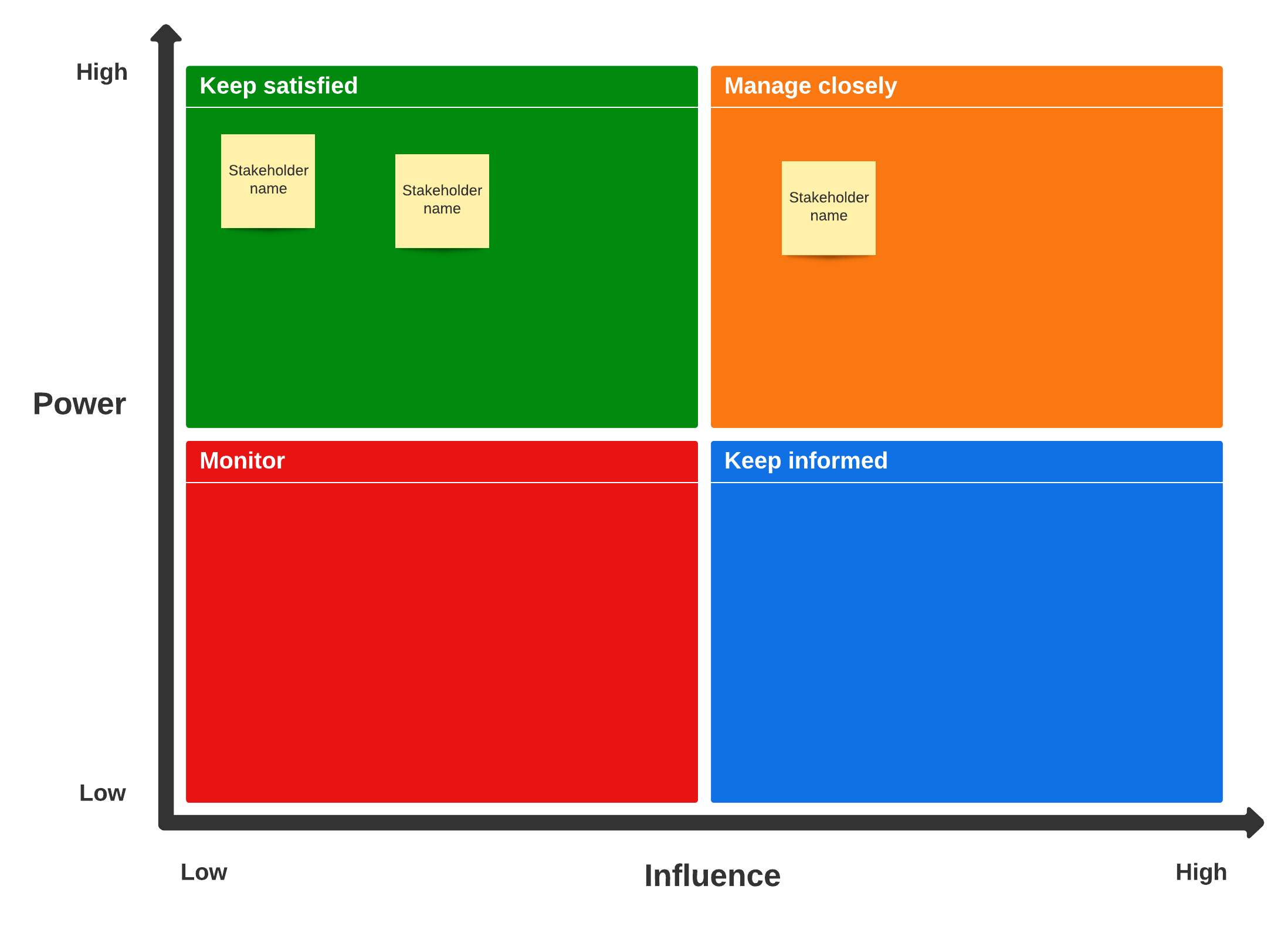


**8. Stakeholder Map**

**Definition**: A Stakeholder Map identifies all the key stakeholders of a project and their interests, influence, and involvement. It helps in understanding who to communicate with and manage expectations.

**Example**:

* **Scenario**: Launching a new product.
* **Stakeholders**: Customers, Marketing Team, Sales Team, Product Development, Investors.
* **Map**:
  + Customers: High interest, low influence.
  + Investors: High interest, high influence.
  + Product Development: High influence, high interest.



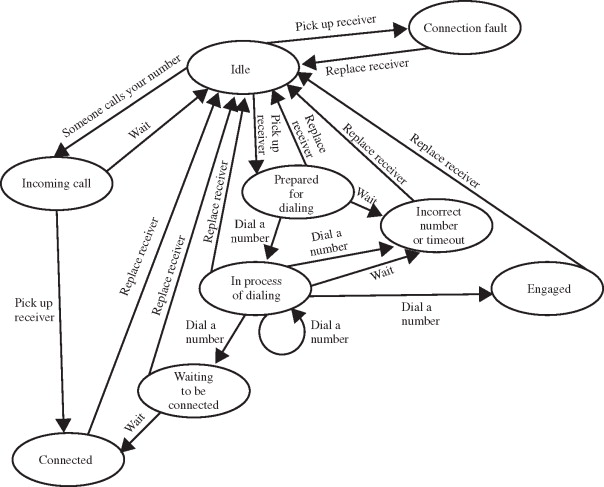
**9. State Diagram**

**Definition**: State diagrams represent the various states of an object and the transitions between these states, typically used in system design to model dynamic behavior.

**Example**:

* **Scenario**: A user account system.
* **States**: Created, Active, Suspended, Closed.
* **Transitions**:
  + Account is created (Created -> Active).
  + Account is suspended due to inactivity (Active -> Suspended).
  + Account is closed upon request (Active/Suspended -> Closed).

A state diagram (also known as a state machine or statechart diagram) is an illustration of all the possible behavioral states a software system component may exhibit and the various state changes it's predicted to undergo over the course of its operations.

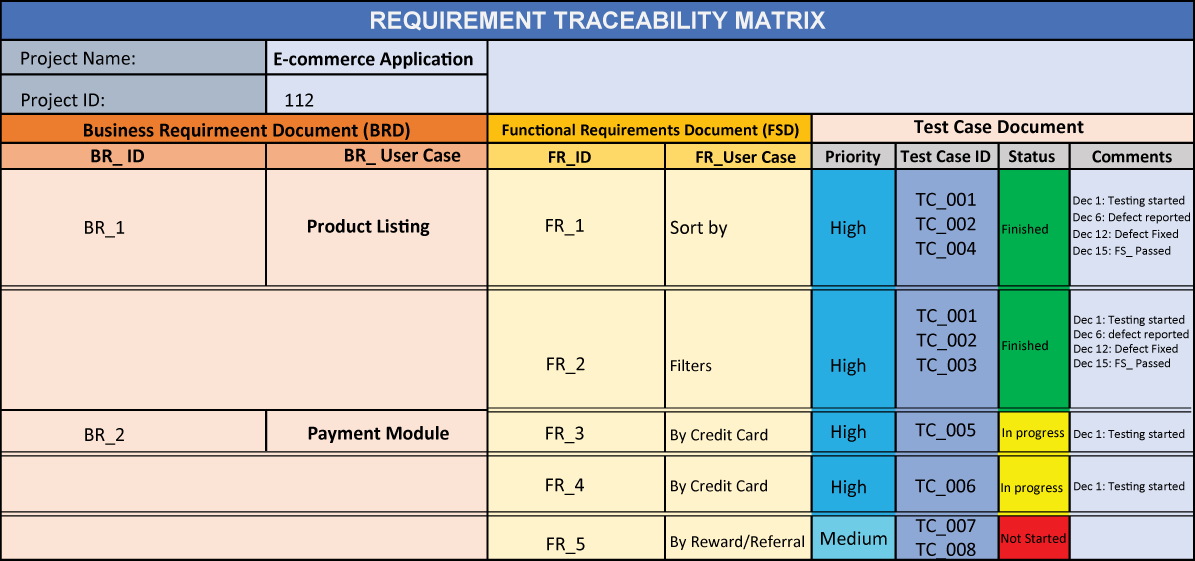


**10. Requirement Traceability Matrix (RTM)**

**Definition**: RTMs link requirements to their corresponding test cases, ensuring each requirement is tested and tracked throughout the development lifecycle.

**Example**:

* **Scenario**: A new feature in an app.
* **Requirements**: “User can reset password.”
* **Test Cases**: “Test password reset via email,” “Test reset link expiry.”
* **Matrix**:
  + Maps Requirement ID to Test Case IDs.
  + Tracks status (Pass/Fail).



The Requirements Traceability Matrix (RTM) is a tool or a document that helps project managers establish and track the project progress. It helps monitor deliveries by providing a digital thread for each demand from the beginning to the end of the project. This guarantees that the scope and deliverables of the project are in line with the baseline.

Traceability Matrix is used in different phases of the project lifecycle, such as

The Requirements Analysis and Specification phase

Design Analysis and Specification phase

Risk Analysis phase

Source Code Analysis, Unit Testing & Integration Testing phase

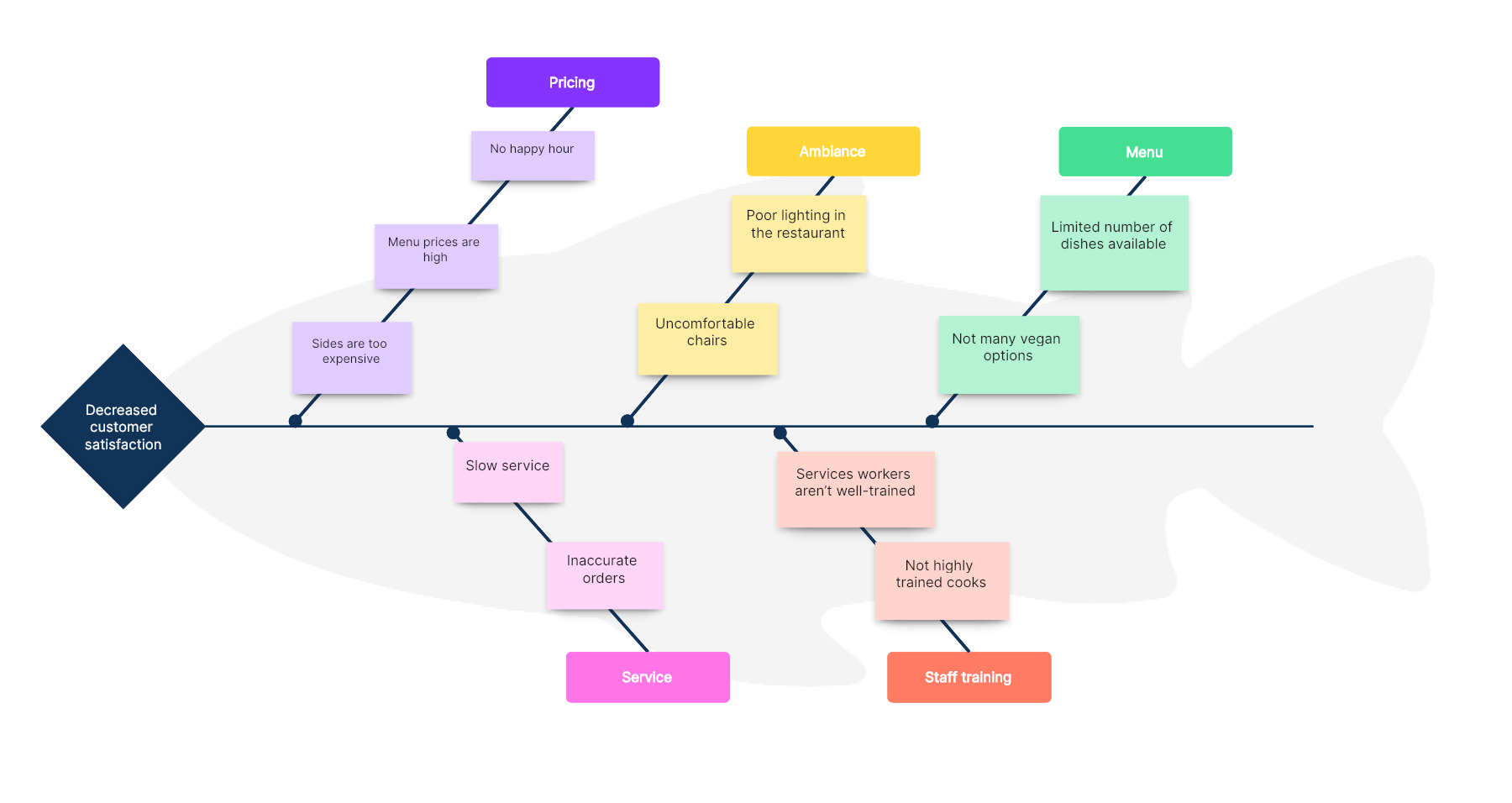
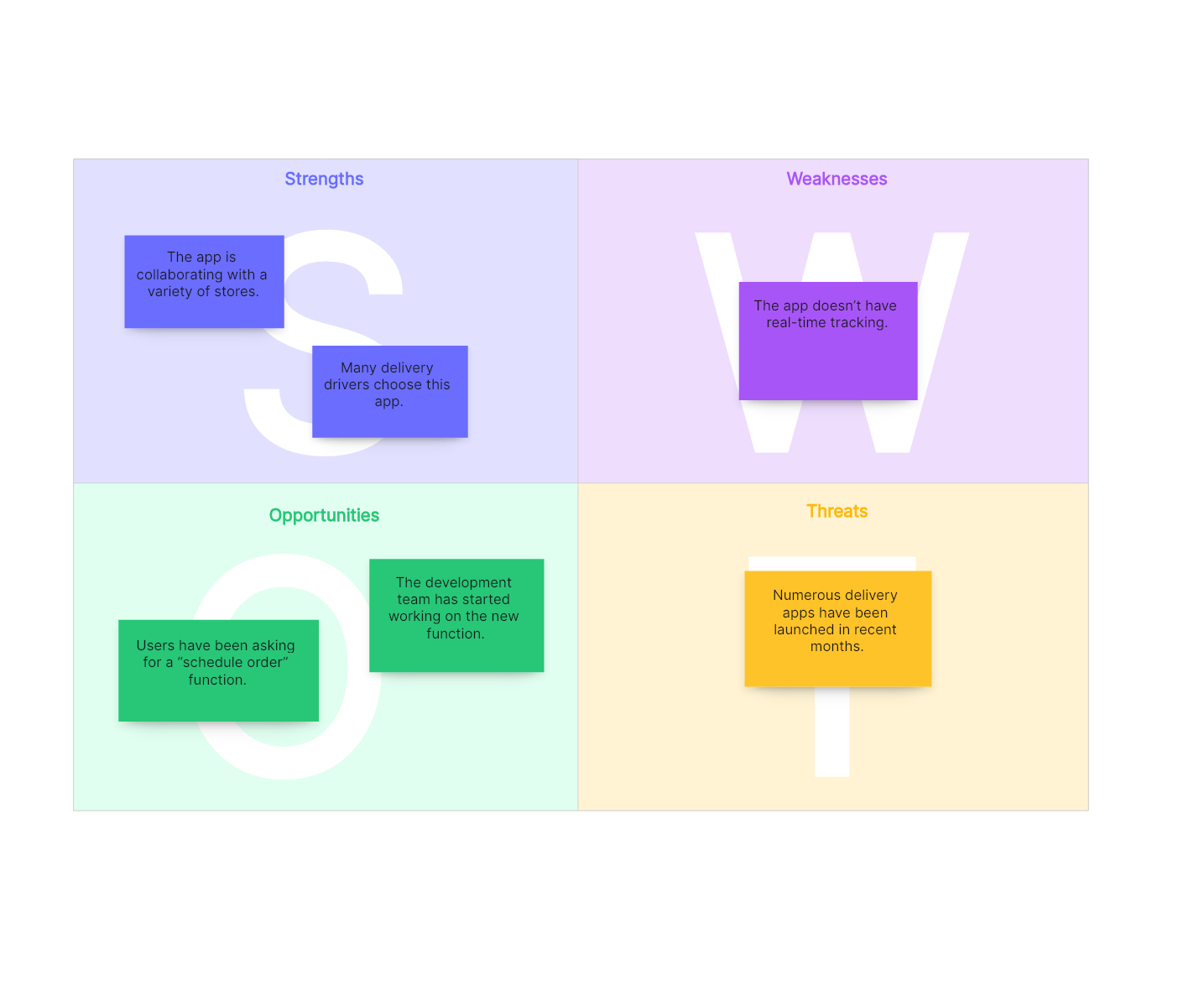
Validation – System Testing, Functional Testing phase

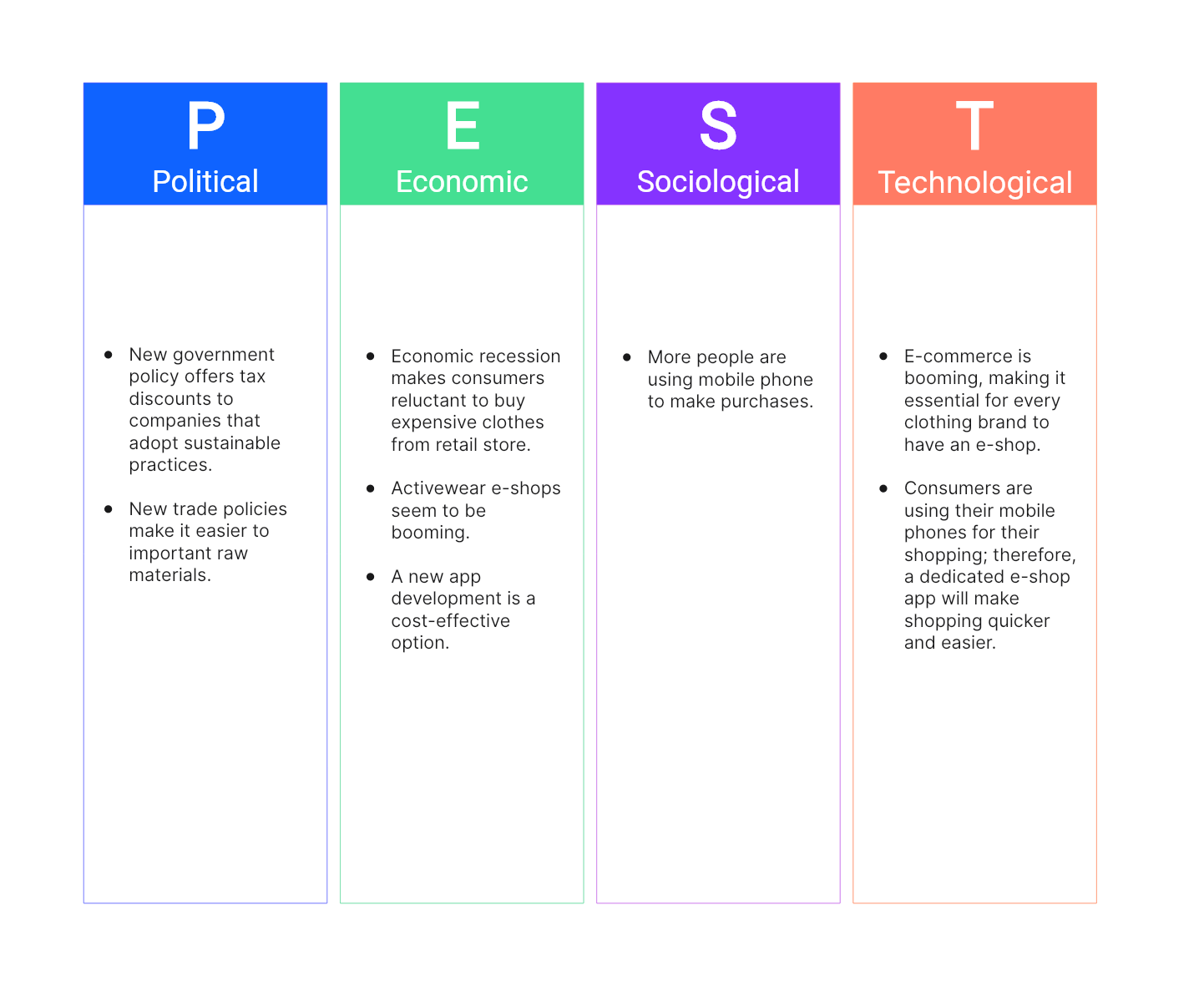
**11. Gap Analysis Diagram**

**Definition**: Gap Analysis identifies the gap between current performance and desired goals, providing a basis for improvement plans.

**Example**:

* **Scenario**: Analyzing software performance.
* **Current State**: 70% uptime.
* **Desired State**: 99.9% uptime.
* **Gap**: 29.9%.
* **Action Plan**: Upgrade servers, implement better monitoring tools.





**12. Activity Diagram**

**Definition**: Activity diagrams represent the flow of activities in a system or process, often used to model workflows, operations, or use cases.

**Example**:

* **Scenario**: User registration process.
* **Activities**: “Fill Registration Form,” “Submit Form,” “Validate Data,” “Send Confirmation Email.”
* **Flow**:
  + User fills the form (Start).
  + Form is submitted (Transition).
  + Data is validated (Decision).
  + If data is valid, a confirmation email is sent (End).

Activity diagrams show the workflow from a start point to the finish point detailing the many decision paths that exist in the progression of events contained in the activity

