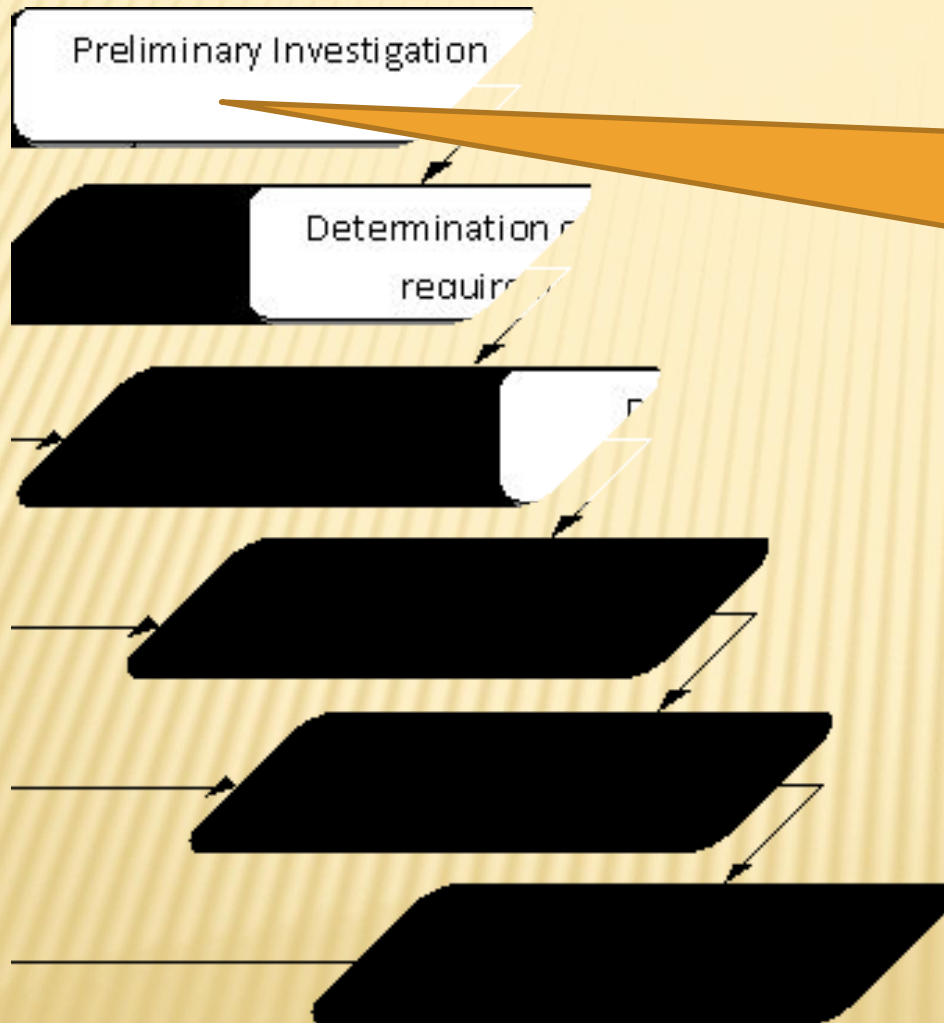# PURPOSE OF DIAGRAMS

Preliminary Investigation

Determination requir

1. CLARIFY PROJECT REQUEST
2. PERFORME FESIBILITY ANALYSIS
3. APPROVAL

STUDENTS

- They are more concerned with Project Completion and Learning how to develop software.

OWNER OF THE COMPANY

- Owner of the company more concerns with Profit Maximization and Cost Minimization.
- Cost benefit analysis is the main task under economic feasibility.
- There should be try to make profit by identifying problem of organization and constructing solution in the form of software.

3

THERFORE STUDENT SHOULD IDENTIFY ..

- Management needs
- User needs

ORGANIZATION CHART
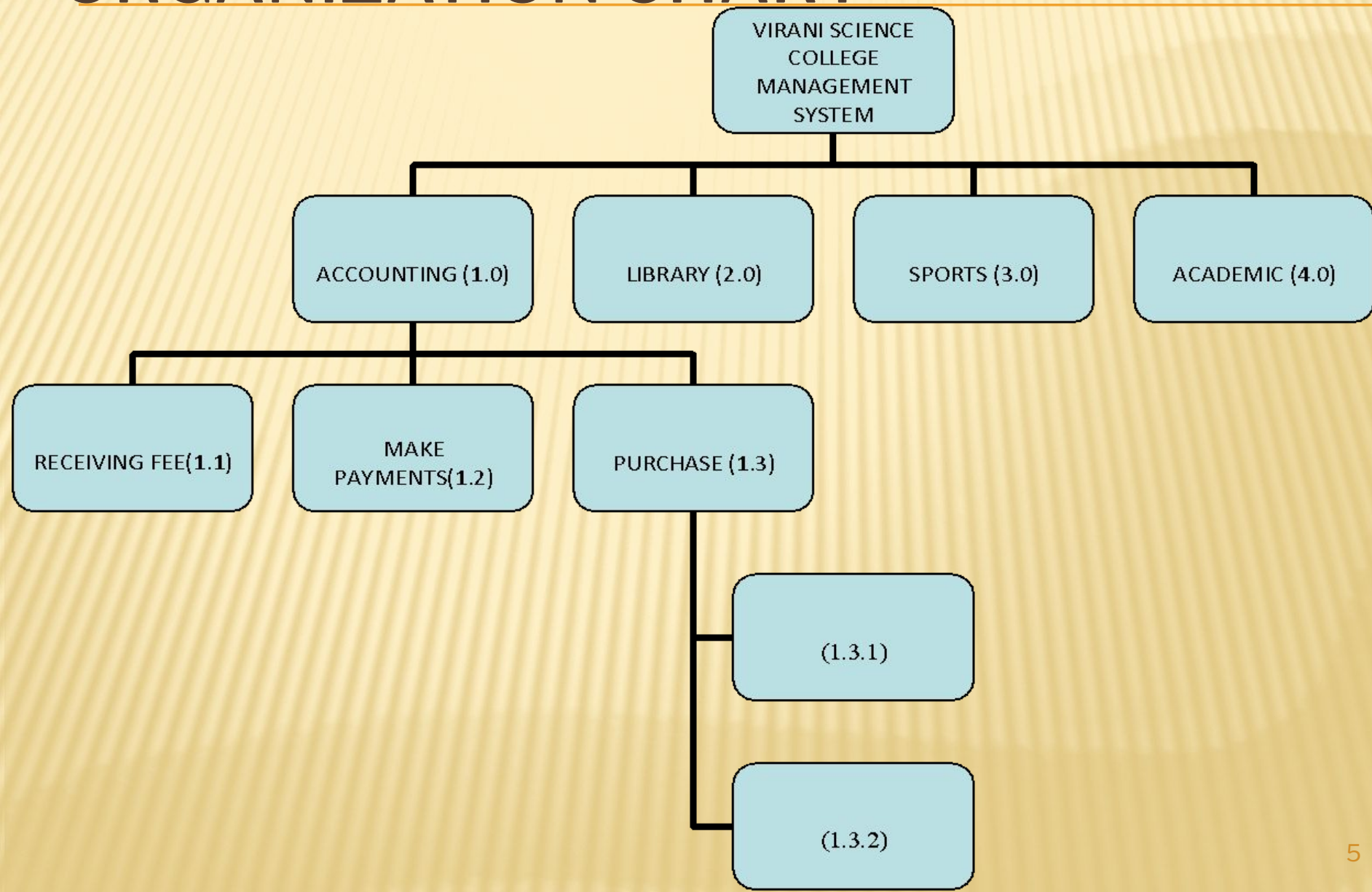
ORGANIZATION

- System can be divided in to ..
  - Subsystems
  - Components
  - Modules
- Module comprised of ..
  - Programs
  - Data Structures (Files)

PURPOSE OF DIAGRAM

Preliminary Investigation

Determination of requirements

1. APPLY FACT FINDING TECHNIQUES
2. DRAW USE CASE DIAGRAM - VERIFY WITH CUSTOMER AND END USER

1. DRAW DFD
2. DESIGN ALGORITHM , DATA STRUCTURE (DATABASE)
3. USER INTERFACE FLOW DIAGRAM
4. INTERFACE DESIGNING AND SO ON

WHAT IS A USE-CASE

- A use-case captures some user visible function
- This may be a large or small function
  - Depends on the level of detail in your modeling effort
- A use-case achieves a discrete goal for the user
- Examples
  - Format a document
  - Request an elevator
- How are the use cases found (captured or elicited)?

USE-CASE DIAGRAMS (POST)

POST: Point of Sale Terminal
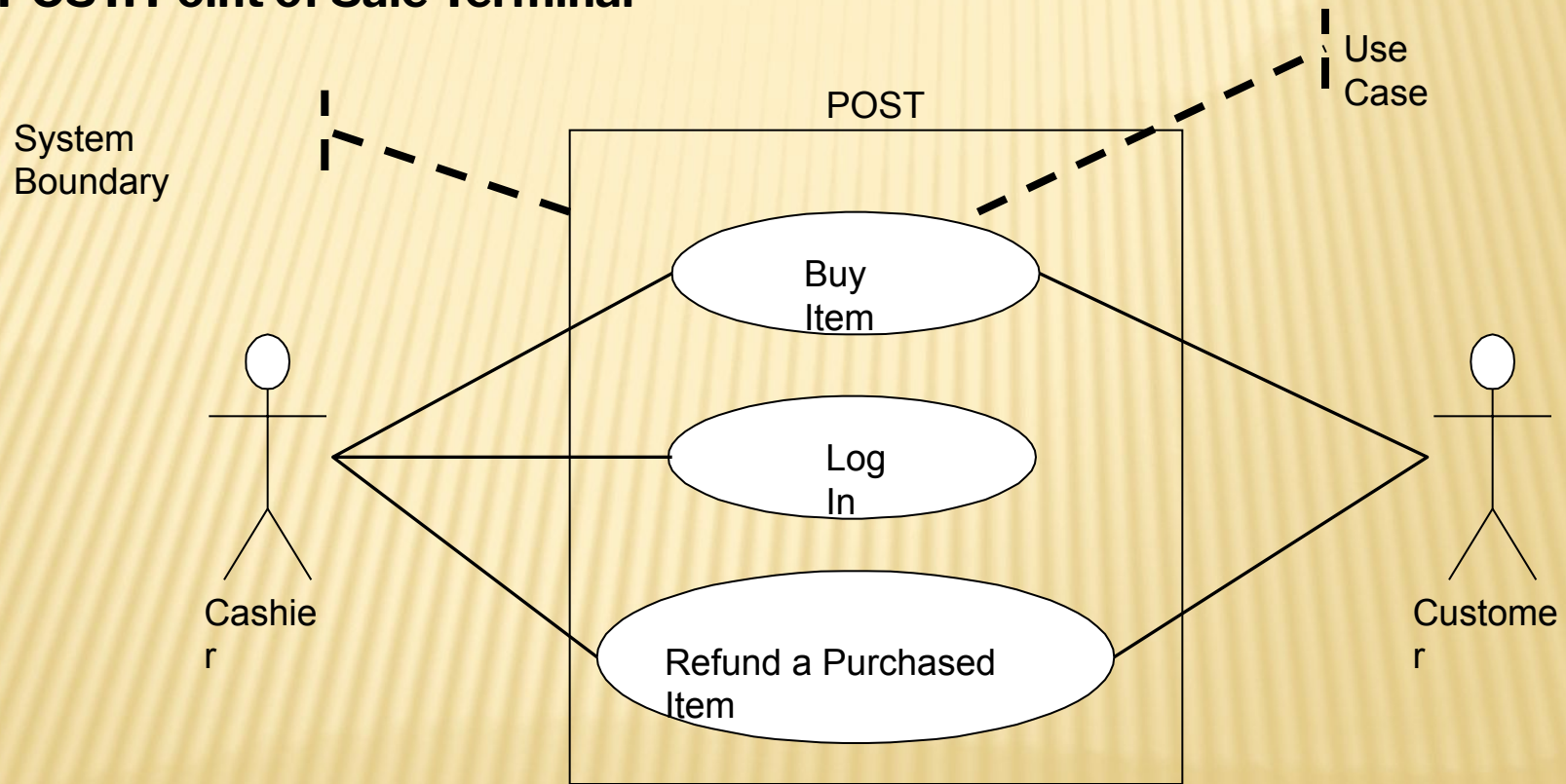
Adapted from Larman "Applying UML and Patterns"

WE WILL COVER

- What is a use-case
  - Use-case versus user interaction
- Use-Case diagrams
  - The constructs in the use-case diagrams
- Capturing the use-case
  - High-level use-case
  - Extended use-case

USER GOALS VERSUS USER INTERACTIONS

- Consider the following when formatting a document
  - Define a style
  - Change a style
  - Copy a style from one document to the next
  - versus
    - Format a document
    - Ensure consistent formatting of two documents
- The latter is a user goal
  - Something the user wants to achieve
- The former are user interactions
  - Things the user does to the system to achieve the goal

GOALS AND INTERACTIONS

- There is a place for both goals and interactions
- Understand what the system shall do
  - Capture the user goals
- Understand how the user will achieve the goals
  - Capture user interactions
  - Sequences of user interactions
- Thus, start with the user goals and then refine the user goals into several (many) user interactions

INCLUDES AND EXTENDS

- Includes
  - You have a piece of behavior that is similar across many use cases
  - Break this out as a separate use-case and let the other ones "include" it
  - Examples include
    - Valuation
    - Validate user interaction
    - Sanity check on sensor inputs
    - Check for proper authorization

- Extends
  - A use-case is similar to another one but does a little bit more
  - Put the normal behavior in one use-case and the exceptional behavior somewhere else
    - Capture the normal behavior
    - Try to figure out what can go wrong in each step
    - Capture the exceptional cases in separate use-cases
 - Makes it a **lot** easier to understand

INCLUDES

- You have a piece of behavior that is similar across many use cases
- Break this out as a separate use-case and let the other ones "include" it
- Examples include
  - Valuation
  - Validate user interaction
  - Check for proper authorization

EXTENDS

- A use-case is similar to another one but does a little bit more
- Put the normal behavior in one use-case and the <u>exceptional behavior</u> somewhere else
  - Capture the normal behavior
  - Try to figure out what can go wrong in each step
  - Capture the exceptional cases in separate use-cases
- Makes it a lot easier to understand

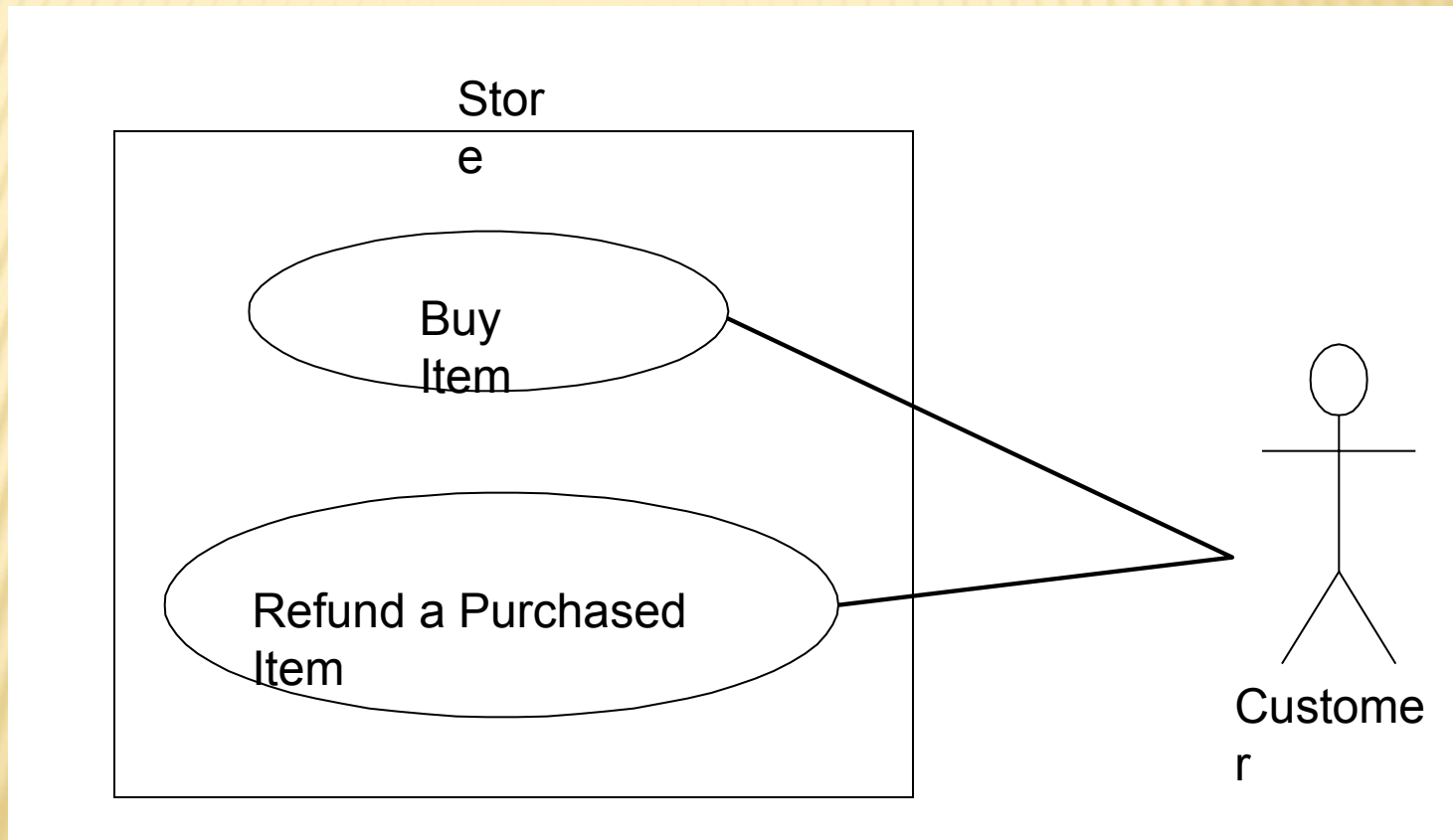# SETTING THE SYSTEM BOUNDARY

- The system boundary will affect your actors and use-cases
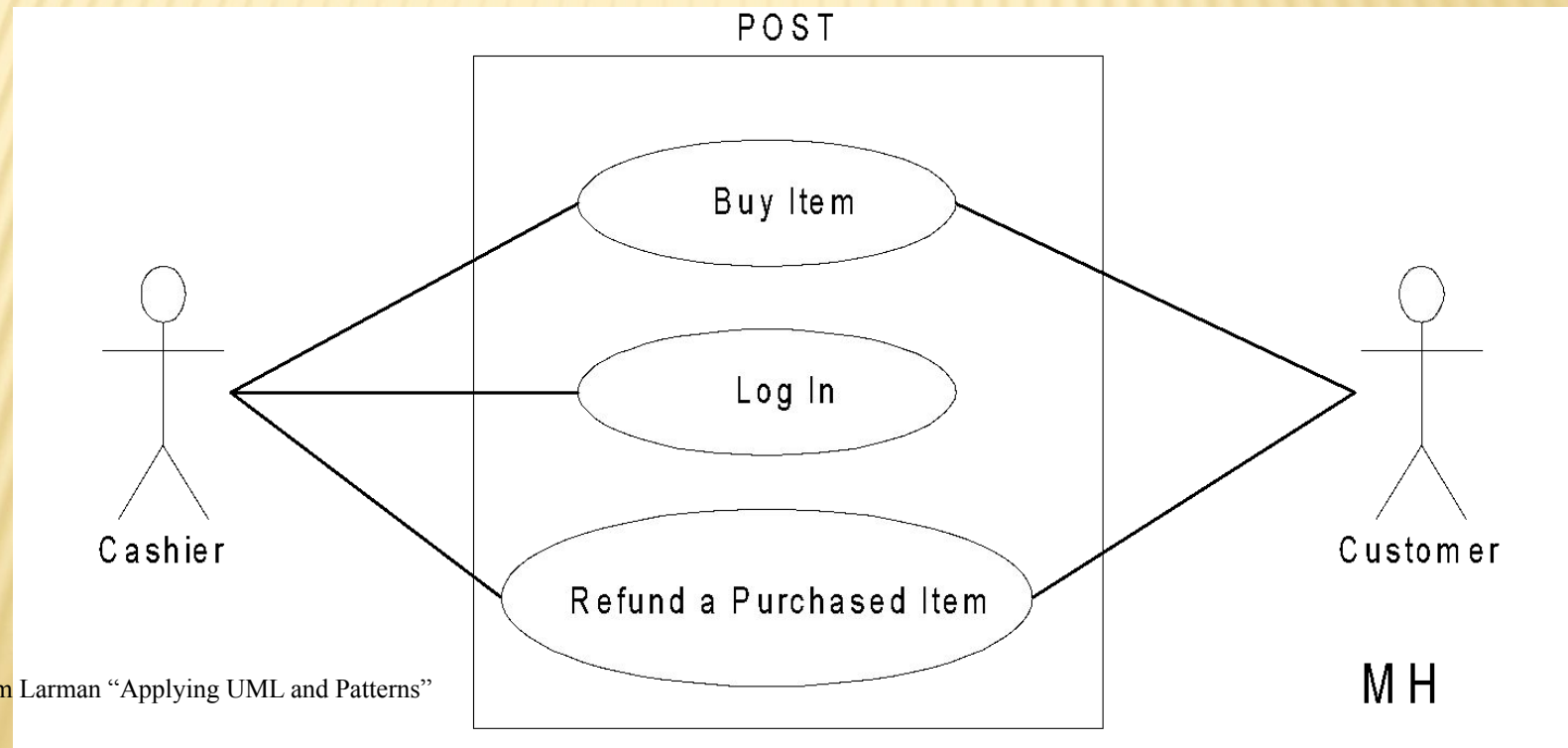
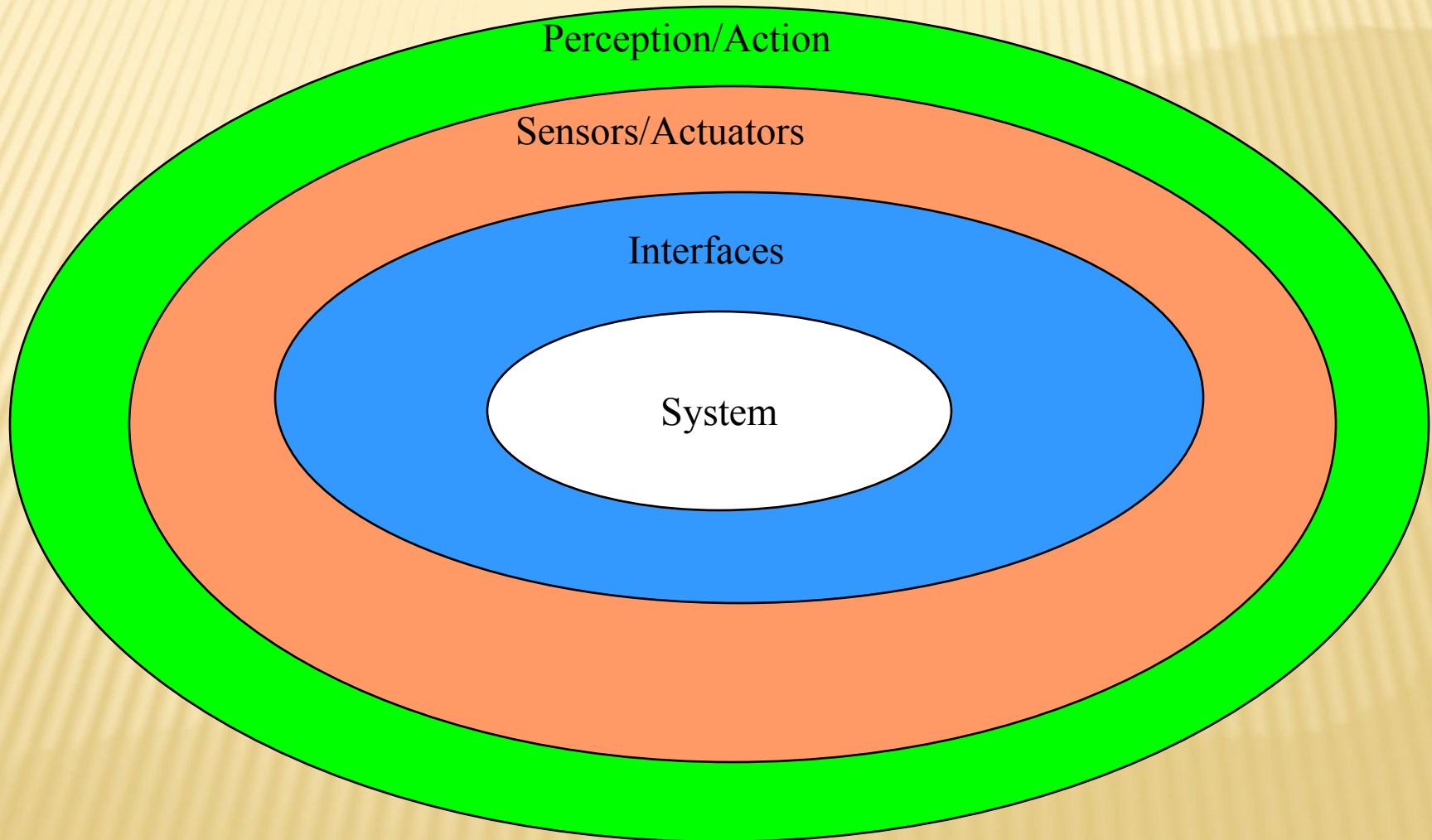Adapted from Larman "Applying UML and Patterns"

A DIFFERENT BOUNDARY

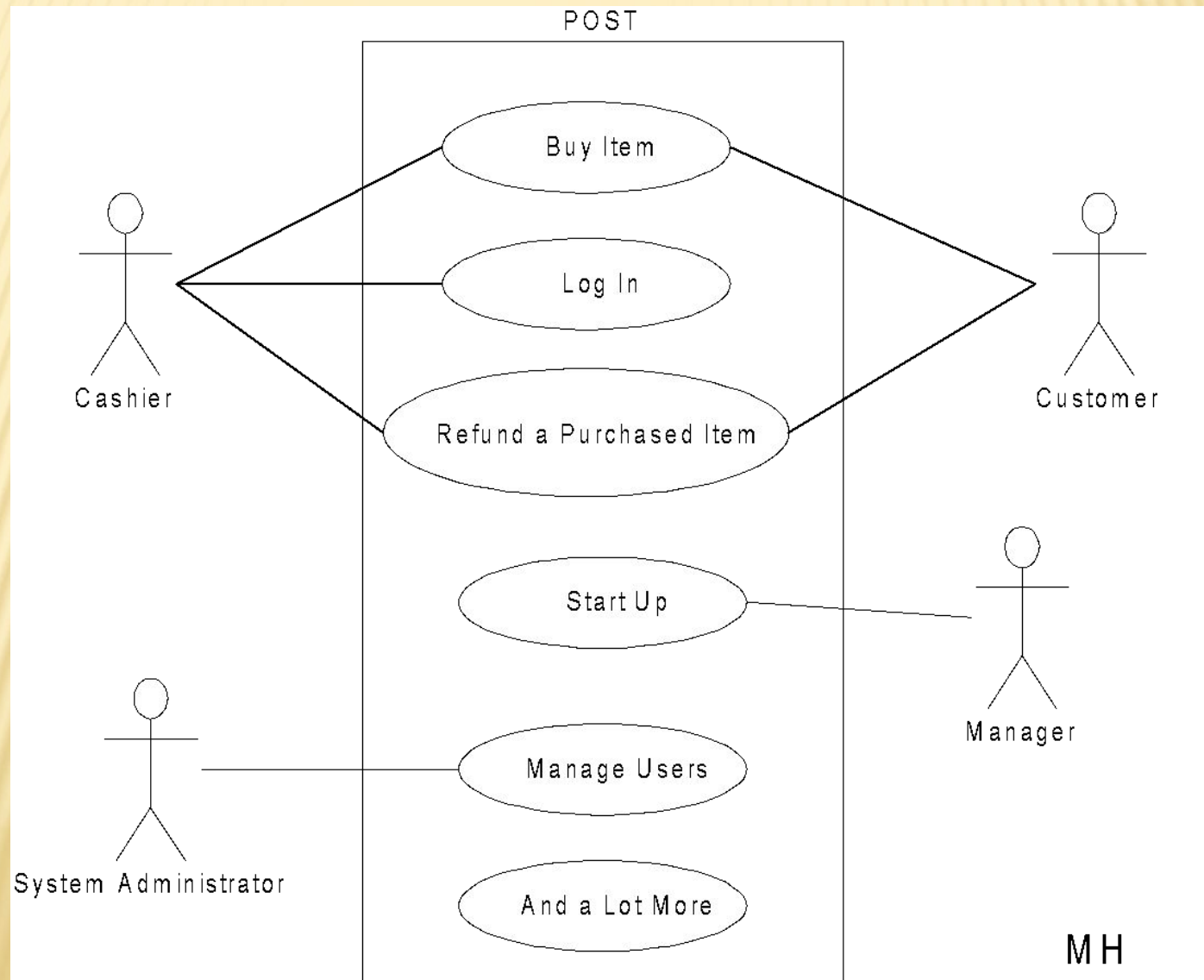- Let us view the whole store as our system

EMBEDDED SYSTEM "ONION SKIN"

*

Perception/Action

Sensors/Actuators

Interfaces

System

PARTIAL POST

POST USE-CASE

Use case: **Buy Item**

Actors: Customer (initiator), Cashier

Type: Primary

Description: The Customer arrives at the checkout with items to purchase. The Cashier records the purchase items and collects a payment. On completion the Customer leaves with the items

POST EXPANDED USE-CASE

Use case: **Buy Item**

Actors: Customer (initiator), Cashier

Type: Primary and essential

Description: The Customer arrives at the checkout with items to purchase. The Cashier records the purchase items and collects a payment. On completion the Customer leaves with the items.

Cross Ref.:   Requirements XX, YY, and ZZ

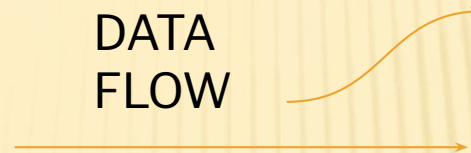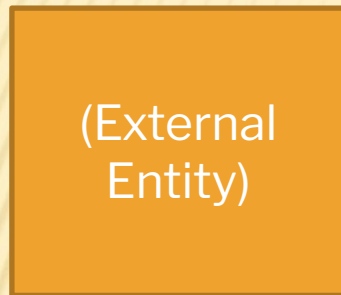**Use-Cases:** Cashier must have completed the *Log In* use-case

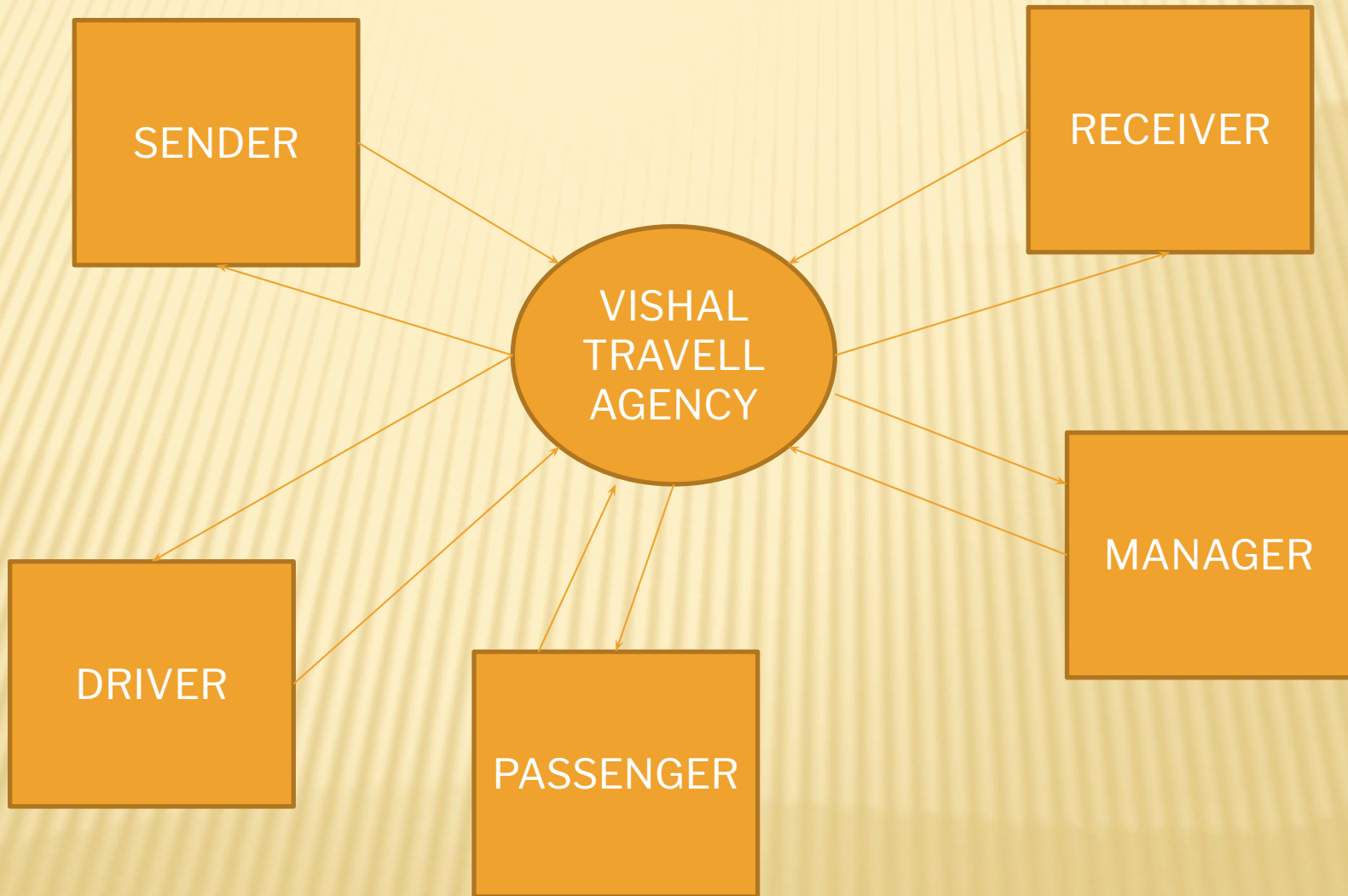# TYPICAL COURSE OF EVENTS

Actor Action
1. This use-case begins when a user arrives at the checkout
2. The cashier records purchase items
3. The cashier collects payment
4. The user leaves with items

DATA FLOW DIAGRAM

DFD SYMBOLS

(External Entity)

DATA FLOW

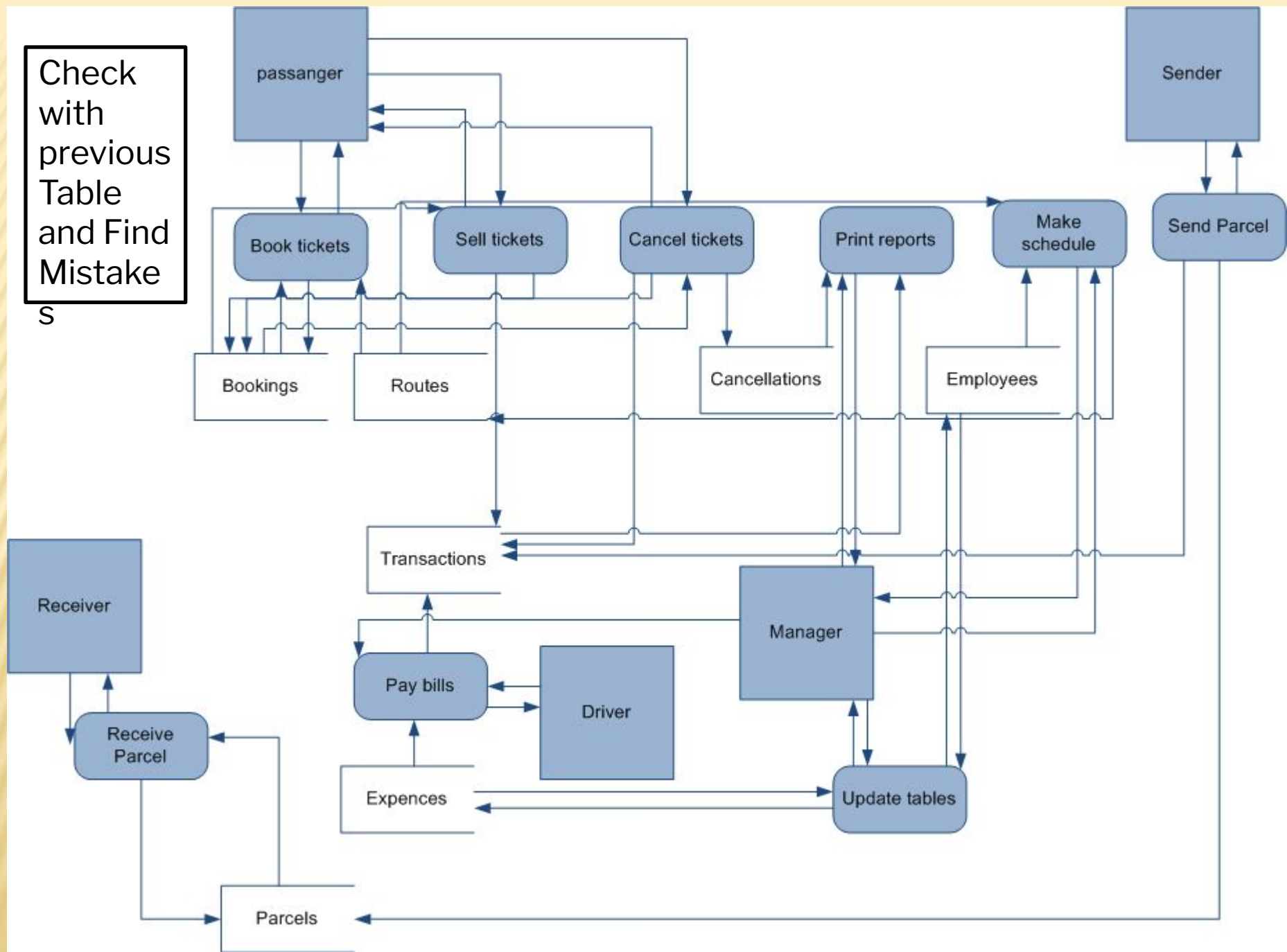PROCESS

PROCESS

DATA STORE

TEMPORARY DATA STORE

CONTEXT DIAGRAM

PREPARE FOLLOWING LIST

PROCESS	DATA STORE	EXTERNAL ENTITY
1. BOOK TICKET	A. ROUTES	a. SENDER
2. SELL TICKET	B. BOOKING	b. RECEIVER
3. CANCEL TICKET	C. CANCLELLATION	c. PASSANGER
4. PRINT REPORT	D. EXPENSES	d. DRIVER
5. MAKE SCHEDULE	E. EMPLOYEES	e. MANAGER
6. SEND PARCLE	F. TRANSACTIONS	
7. PAY BILLS	G. PARCELS	
8. RECEIVE PARCEL		
9. UPDATE DATABASES		

PREPARE FOLLOWING BEFORE DRAWING

NO.	PROCESS	INPUT FROM		OUTPUT TO	
		DATA STORE	EXTERNAL SOURCE	DATA SOTRE	EXTERNAL DESTINATION
1	BOOK TICKET	B,A	Passenger	B	Passenger
2	SELL TICKET	B,A	Passenger	B,F	Passenger
3	CANCEL TICKET	B	Passenger	B,C,F	Passenger
4	PRINT REPORT	B,C,F	Maneger	------	Manager
5	MAKE SCHEDULE	A,E	Manager	A	Manager
6	SEND PARCLE	A	Sender	G,F	Sender
7	PAY BILLS	D	Driver	F	Driver
8	RECEIVE PARCEL	G	Receiver	G	Receiver
9	UPDATE DATABASES	D,E	Manager	D,E	Managerr

Check with previous Table and Find Mistakes

USER INTERFACE FLOW DIAGRAM