Understanding Requirements

Software Engineering

The Problems with our Requirements Practices

We have trouble understanding the requirements that we do acquire from the customer

We often record requirements in a disorganized manner

We spend far too little time verifying what we do record

We allow change to control us, rather than establishing mechanisms to control change

Most importantly, we fail to establish a solid foundation for the system or software that the user wants built

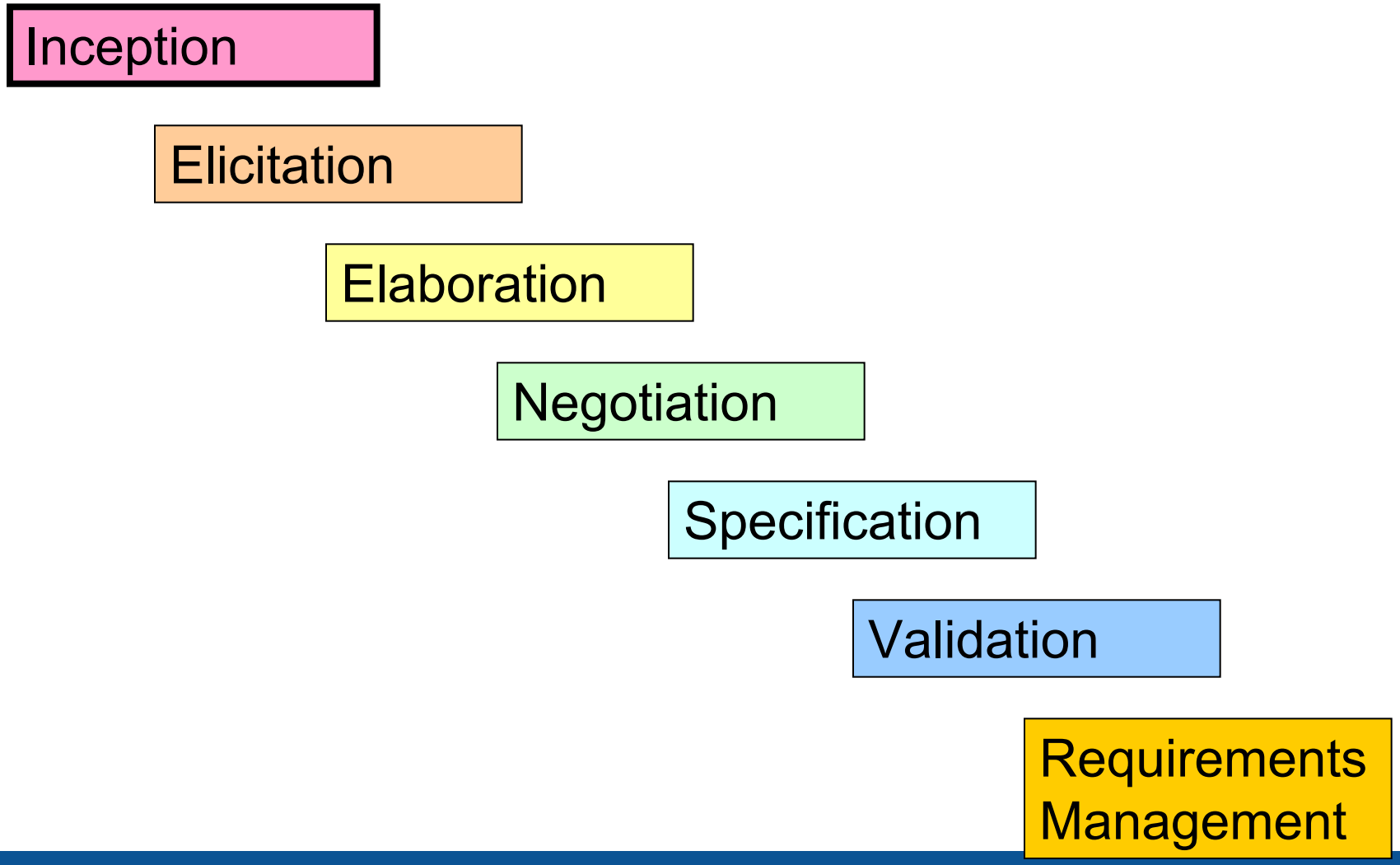
A Solution – Requirements Engineering

- Begins during the communication activity and continues into the modeling activity

- Builds a bridge from the system requirements into software design and construction

- Allows the requirements engineer to examine
  the context of the software work to be performed
  the specific needs that design and construction must address
  the priorities that guide the order in which work is to be completed
  the information, function, and behavior that will have a profound impact on the resultant design

Seven Distinct Steps – Requirements Engg Tasks



Inception Task

During inception, the requirements engineer asks a set of questions to establish…

A basic understanding of the problem
The people who want a solution
The nature of the solution that is desired
The effectiveness of preliminary communication and collaboration between the customer and the developer

Through these questions, the requirements engineer needs to…

Identify the stakeholders
Recognize multiple viewpoints
Work toward collaboration
Break the ice and initiate the communication

First set of Questions

These questions focus on the customer, other stakeholders, the overall goals, and the benefits

Who is behind the request for this work?

Who will use the solution?

What will be the economic benefit of a successful solution?

Is there another source for the solution that you need?

Next Set of Questions

These questions enable the requirements engineer to gain a better understanding of the problem and allow the customer to voice his or her perceptions about a solution

How would you characterize "good" output that would be generated by a successful solution?

What problem(s) will this solution address?

Can you show me (or describe) the business environment in which the solution will be used?

Will special performance issues or constraints affect the way the solution is approached?

Final Set of Questions

These questions focus on the effectiveness of the communication activity itself

Are you the right person to answer these questions? Are your answers "official"?

Are my questions relevant to the problem that you have?

Am I asking too many questions?

Can anyone else provide additional information?

Should I be asking you anything else?

Exercise

Campus Information Access Kiosk

- Both podium-high and desk-high terminals located throughout the campus in all classroom buildings, admin buildings, labs, and dormitories
- Hand/Palm-login and logout (seamlessly)
- Voice input
- Optional audio/visual or just visual output
- Immediate access to all campus information plus
  - E-mail
  - Cell phone voice messaging

Case Study – Phase – I – Retail Application

Problem Statement: Easy shop wants to automate the system of purchase of items by customers and billing process as Phase I. The automation involves maintenance of customers, purchase of items by customer and billing of items. Customers can visit any of the retail outlets of Easy Shop and purchase items. Customers can be regular or privileged customers. Customers who are regular visitors to the store are eligible for discount on the bill amount. The privileged customers are given membership cards (Platinum, Gold and Silver). Such customers are eligible for gifts based on the type of membership card. The Billing staff does the billing and delivery of items to the customer. The bill calculation involves the logic of computation of the bill depending on customer type. The customer can pay the bill through credit card or cash. In the former case, two percent processing charge is applicable. VAT % is also applicable on the final bill amount.

The store wants to initially pilot the system where purchase is done by one customer for one item.

Case Study – Course Registration System

Situation: A Course Registration System needs to be developed for an engineering college. The college wants an automated system to replace its manual system for the purpose of registration of students to branches and calculation of fees for each year. The engineering college provides graduation courses in various branches of engineering.

The system will be used by the admin staff to register students admitted to the college to the branches at the time of joining the college and also to calculate the yearly fees for the students. The student has to register every year for the next academic year. The Admin takes care of the yearly registration of the students and the calculation of yearly fees. The system needs to be authenticated with a login id and password.

Registration of a student to a branch is based on the qualifying exam marks and the entrance counseling. For every branch, a yearly branch fee is applicable. Discounts are given to the branch fees of the first year based on the qualifying exam marks. There is a registration fees also applicable to the first year students. Students can opt to be a day scholar or hostelite. Yearly bus fees are applicable for all the day scholars based on the distance of travel. Yearly hostel fees are applicable for all the hostelites. Yearly infrastructure fees and library fees are also applicable to all the students. Admin calculates the yearly college fees for each student and the college fees include all the fees specified earlier based on the type of student. Admin will provide a printed receipt of the fees to the students once the annual college fees have been paid.

At the time of registration, student has to provide the permanent address and in case the student is opting to be a day scholar, he/she has to provide the residential address also.

Assumption:

1. Decision of the branch of study a student is allocated, is not within the scope of this case study

Elicitation

Eliciting requirements is difficult because of

Problems of scope

Boundaries of the system ill-defined or specifying too much technical detail rather than overall system objectives

Problems of understanding

What is wanted, what the problem domain is, and what the computing environment can handle

Problems of volatility

The requirements change over time

Elicitation may be accomplished through two activities

Collaborative requirements gathering

Quality function deployment

Collaborative Requirement Gathering

Meetings are conducted and attended by both software engineers, customers, and other interested stakeholders

Rules for preparation and participation are established

An agenda is suggested that is formal enough to cover all important points but informal enough to encourage the free flow of ideas

A "facilitator" (customer, developer, or outsider) controls the meeting

The goal is to identify the problem, propose elements of the solution, negotiate different approaches, and specify a preliminary set of solution requirements

Quality Function Development

Emphasizes an understanding of what is valuable to the customer.

Normal requirements: These requirements are the objectives and goals stated for a product or system during meetings with the customer

Expected requirements: These requirements are implicit to the product or system and may be so fundamental that the customer does not explicitly state them

Exciting requirements: These requirements are for features that go beyond the customer's expectations and prove to be very satisfying when present

Elaboration

During elaboration, the software engineer takes the information obtained during inception and elicitation and begins to expand and refine it

Elaboration focuses on developing a refined technical model of software functions, features, and constraints

It is an analysis modeling task

- Use cases are developed
- Domain classes are identified along with their attributes and relationships
- State machine diagrams are used to capture the life on an object

The end result is an analysis model that defines the functional, informational, and behavioral domains of the problem

Elements of Analysis Model

Scenario-based elements

Describe the system from the user's point of view using scenarios that are depicted in use cases and activity diagrams

Class-based elements

Identify the domain classes for the objects manipulated by the actors, the attributes of these classes, and how they interact with one another; they utilize class diagrams to do this

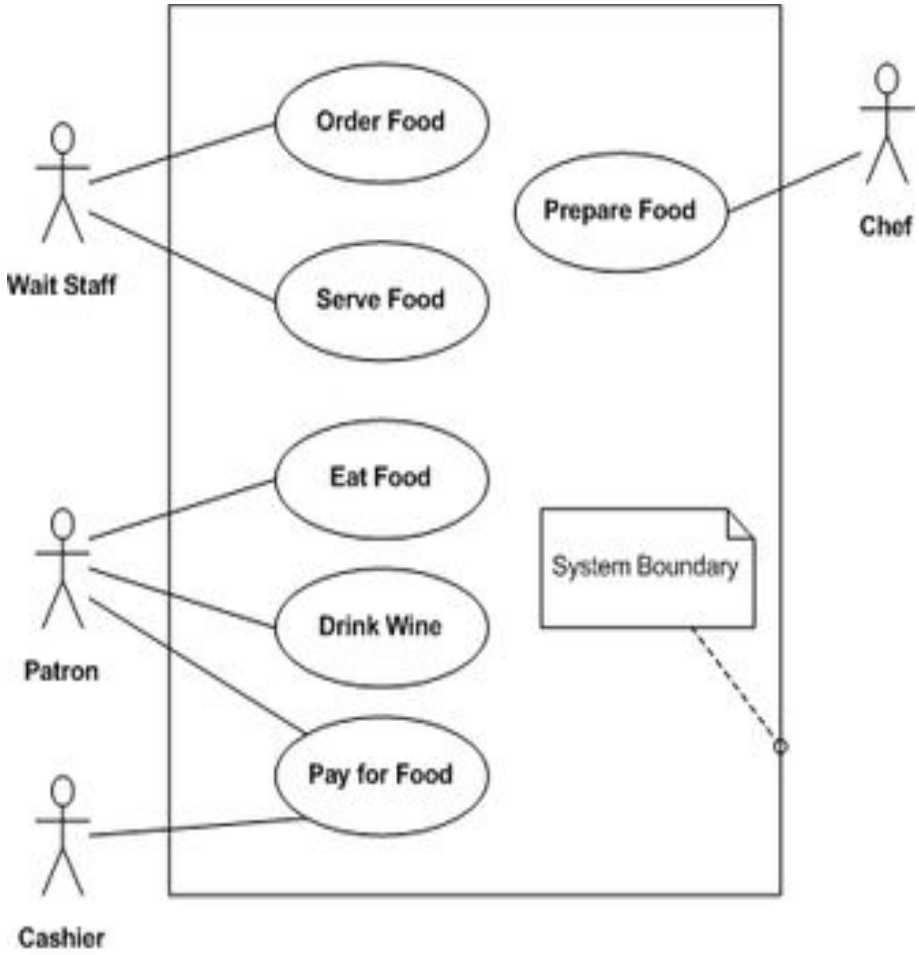
Behavioral elements

Use state diagrams to represent the state of the system, the events that cause the system to change state, and the actions that are taken as a result of a particular event; can also be applied to each class in the system

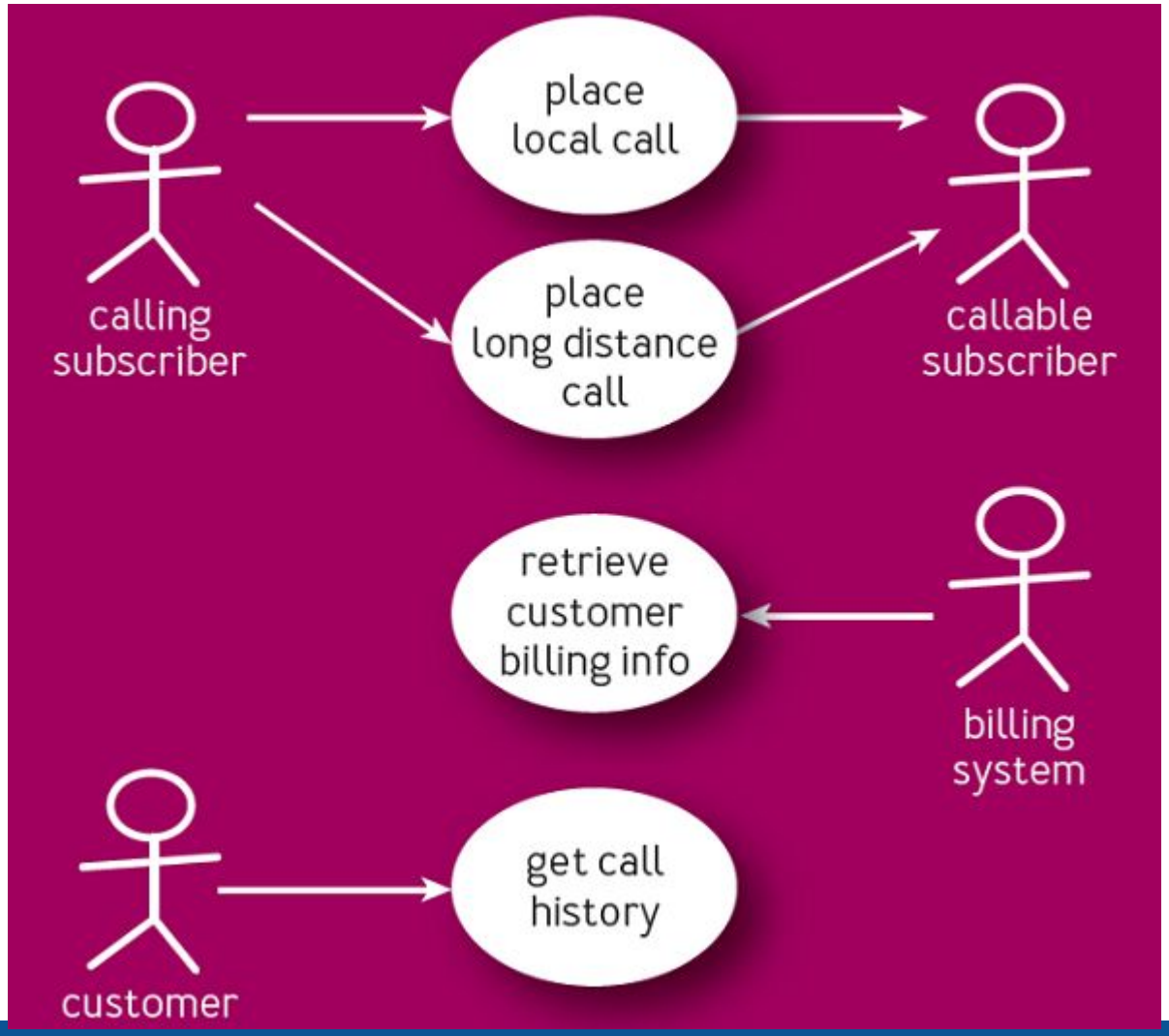
Flow-oriented elements

Use data flow diagrams to show the input data that comes into a system, what functions are applied to that data to do transformations, and what resulting output data are produced

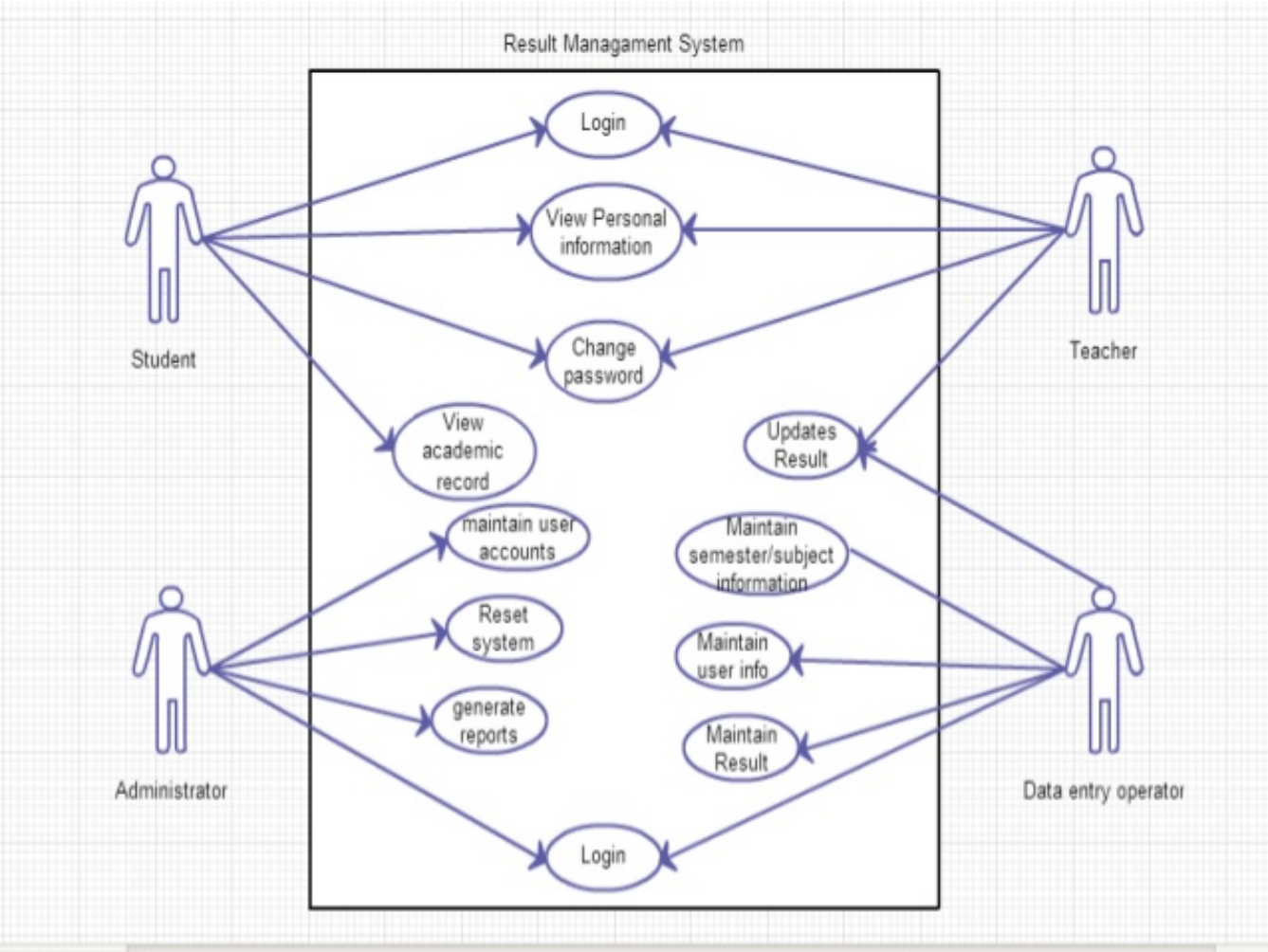
Restaurant Use Case Diagram



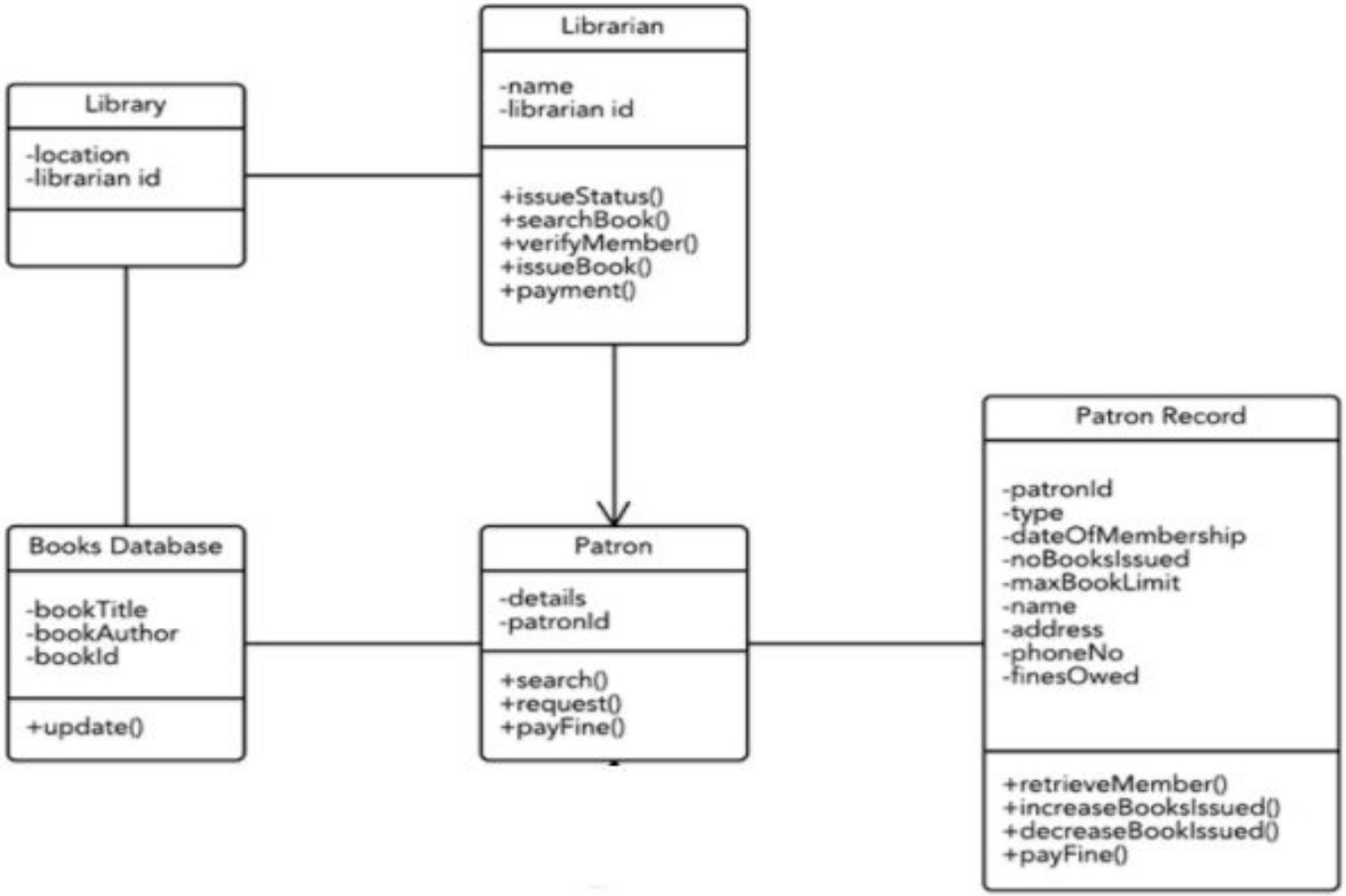
Use Case for Simple Telephone System



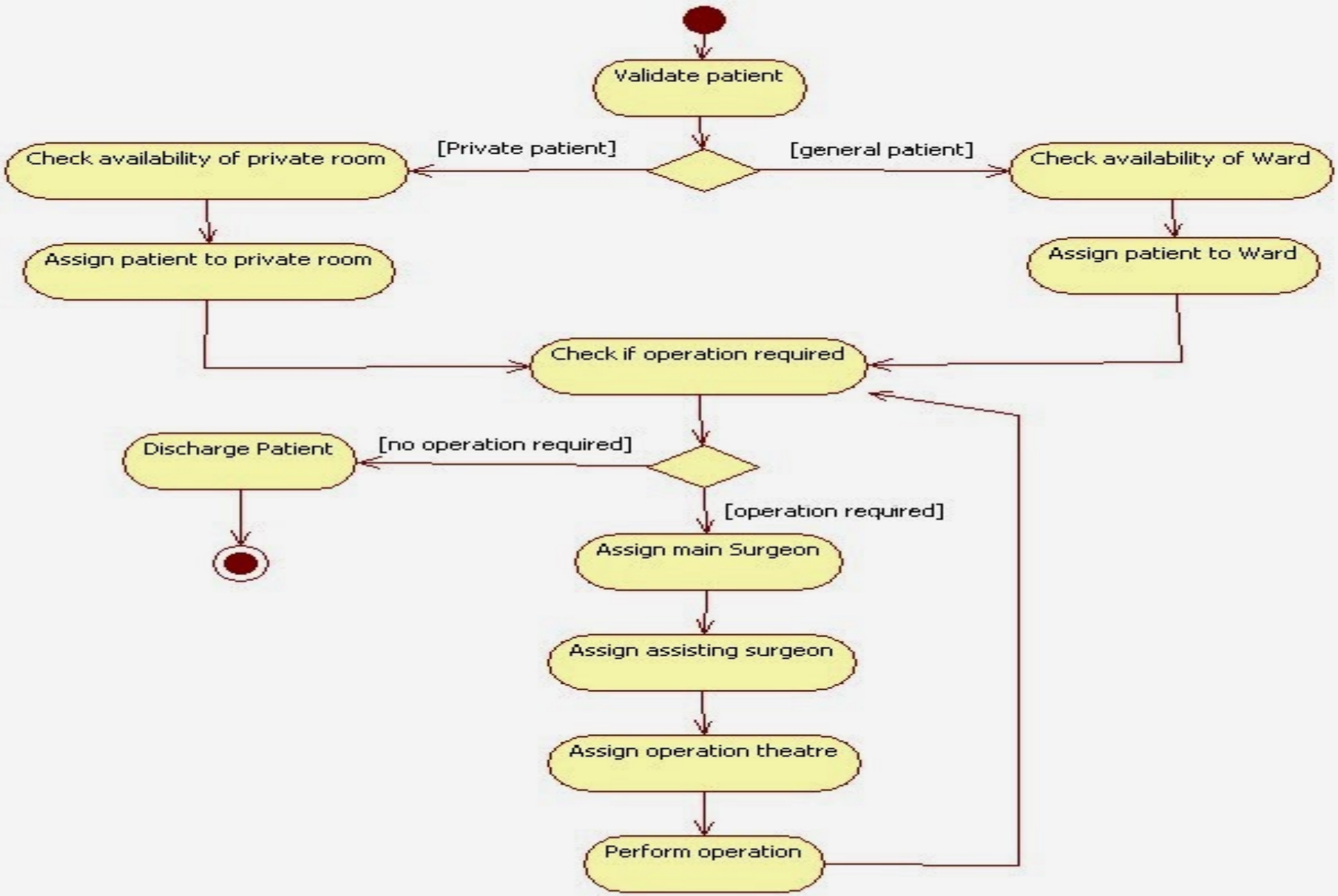
Use Case for Result Management System

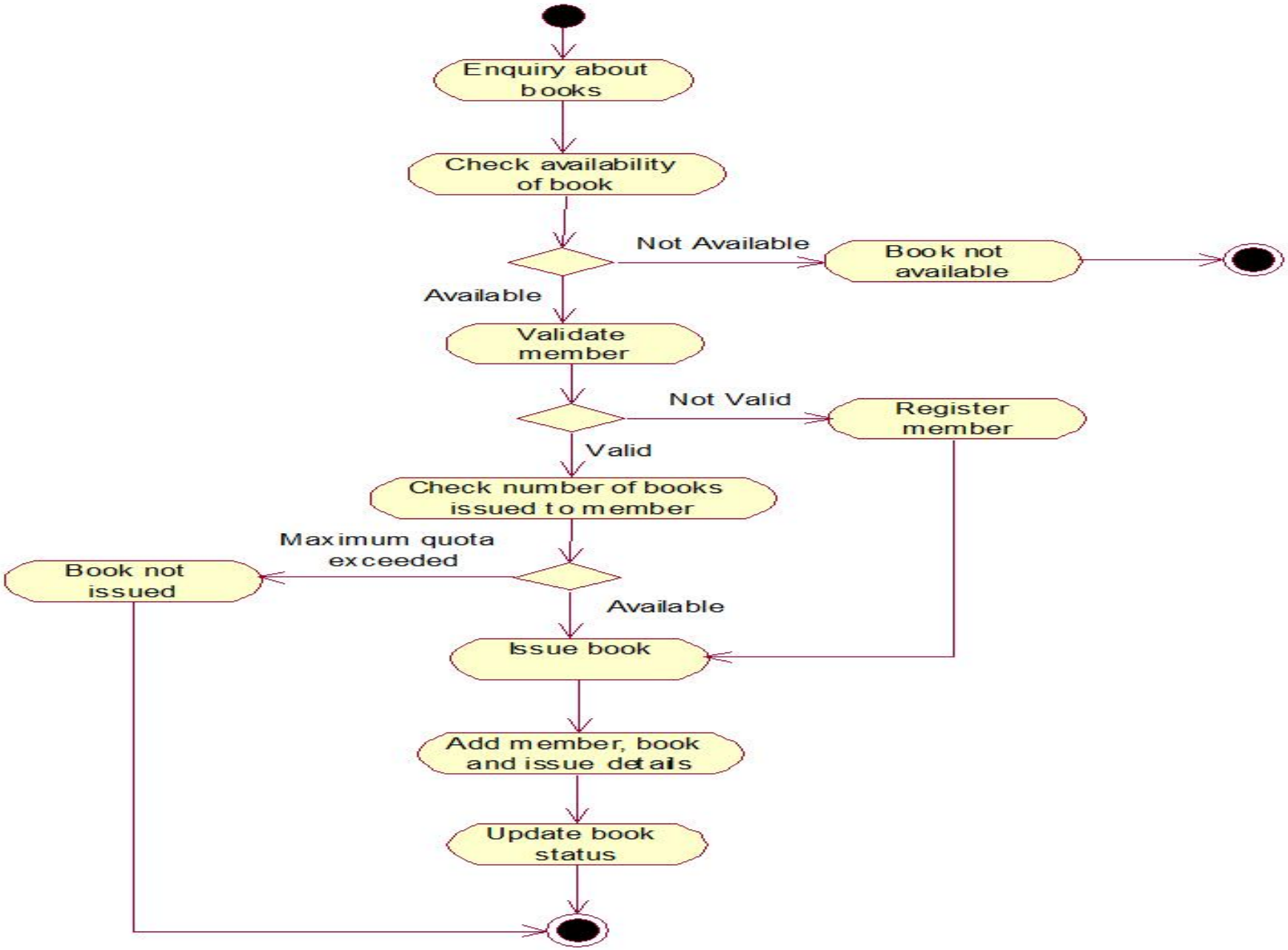


Class Diagram for Library Information System



State Diagram




Enquiry about books
Check availability of book
Not Available
Book not available
Available
Validate member
Not Valid
Register member
Valid
Check number of books issued to member
Maximum quota exceeded
Book not issued
Available
Issue book
Add member, book and issue details
Update book status

Negotiation

During negotiation, the software engineer reconciles the conflicts between what the customer wants and what can be achieved given limited business resources

Requirements are ranked (i.e., prioritized) by the customers, users, and other stakeholders

Risks associated with each requirement are identified and analyzed

Specification

A specification is the final work product produced by the requirements engineer

It is normally in the form of a software requirements specification

It serves as the foundation for subsequent software engineering activities

It describes the function and performance of a computer-based system and the constraints that will govern its development

It formalizes the informational, functional, and behavioral requirements of the proposed software in both a graphical and textual format

Validation

During validation, the work products produced as a result of requirements engineering are assessed for quality

The specification is examined to ensure that

- All software requirements have been stated unambiguously
- Inconsistencies, omissions, and errors have been detected and corrected
- The work products conform to the standards established for the process, the project, and the product

The formal technical review serves as the primary requirements validation mechanism

- Members include software engineers, customers, users, and other stakeholders

Requirement Management

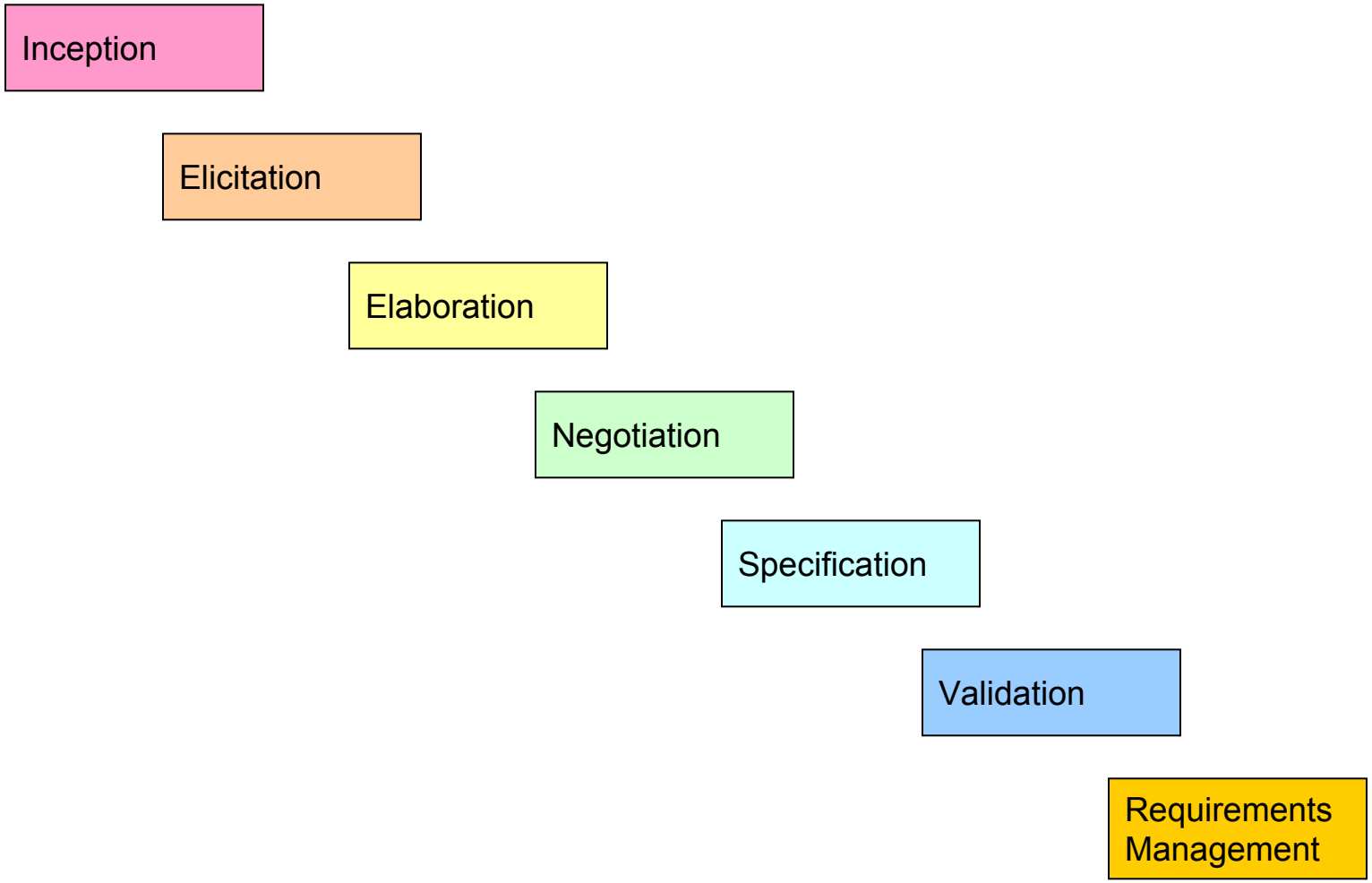
During requirements management, the project team performs a set of activities to identify, control, and track requirements and changes to the requirements at any time as the project proceeds

Each requirement is assigned a unique identifier

The requirements are then placed into one or more traceability tables

These tables may be stored in a database that relate features, sources, dependencies, subsystems, and interfaces to the requirements

A requirements traceability table is also placed at the end of the software requirements specification


Inception
Elicitation
Elaboration
Negotiation
Specification
Validation
Requirements Management

Scenarios	Requirements elicitation technique
Interrogative conversations with Managers, Cashiers, Clerks, and other staff for arriving at the requirements for automating transactions	
Formal and planned requirement discussions in a conference room conducted among managers of diversified branches facilitated by an anchor	
Survey form circulated amongst the users (account holders) who visit the bank, to ease their interactions with the bank	
Analysis for understanding mode of transactions – Checks, Cash, DD, MT, Gold, etc	
Ethnographers deployed for understanding the users interaction with bank officials	

Thank You