Software Engineering
Lab Assignment

Sub Code: CSE-229

Start Date: 08-01-2021

End Date: 09-04-2021

Vivek Kumar Ahirwar

191112419

CSE - 3

Department:

Computer Science and Engineering

Under Guidance of:

Prof. Gagan Vishwakarma,

Mr. Kapil Kumar Soni

Contents

ASSIGNMENT-1 (SRS)

SOFTWARE REQUIREMENT SPECIFICATION (SRS)
Document of Drowsiness Detector

08-01-2021

Introduction:

1. Purpose:

The Software Requirements Specification (SRS) will provide a detailed description of the requirements for *Drowsiness Detector*. This SRS will allow for a complete understanding of what is to be expected of the system to be constructed. A clear understanding of the system and its functionality will allow for the correct software to be developed for the end-user and will be used for the development of the future stages of the project. It will provide the foundation for the project. From the recorded specifications, the *Drowsiness Detector* can be designed, constructed, and finally tested.

2. Definition, Acronyms and Abbreviations:

1) SRS – Software Requirements Specification.

Overall Description:

1. Product Description:

This project used to detect whether students in an online class is drowsy or present in class or not. This Project is based on the fact that nowadays in online classes most of the students generally feel drowsy and they sometimes sleep during classes. Our software uses a camera of their device and whenever a student sleeps or gets off class it plays an alarm and after some time closes the meeting application.

2. Product Functions:

The system will perform the following functions:

I. Real-time image processing using a webcam.
II. Face Detection and Face Landmark Detection.
III. Detection of person's drowsiness with use of EAR ratio.
IV. Play alarm when the person is drowsy.
V. If the person is drowsy for some duration, then online meeting apps will be closed, if the student is not present in front of the screen, then turn off online meeting apps.

3. Product Implementations:

There are several other implementations of our product other than using it in the online class are as follow: -

a) Online Classes:

Students nowadays are attending online classes and while attending lots of time they go to sleep. The teacher will not able to know that a particular student is sleeping or not.

b) Office Employments:

Office employee also sometimes feels drowsy while doing continuous work. If our product will be applied in offices then we can have a watch on employees and the boss can warn them.

c) Drowsy Driving:

A 2014 AAA Traffic Safety Foundation study found that 37 per cent of driver's report having fallen asleep behind the wheel at some point in their lives. An estimated 21 per cent of fatal crashes, 13 per cent of crashes resulting in severe injury and 6 per cent of all crashes, involve a drowsy driver.

External Interface Requirements:

1. User Interface:

First of all, the system will capture the video of the student i.e., the system will capture the image frames continuously. After that, the system will detect the face, mouth and eyes of the student. The face landmark model is stored in the database and the face and eyes are compared with this model. If the system detects drowsiness for more than a set period then an alarm will alert the student. If the system detects the student drowsy for a longer period, then it will automatically close the meeting application.

2. Hardware Interface:

a) System Specification:

i. Intel® Core™ i5 processor.
ii. RAM Required: 4 GB, Disk space: 2 to 3.
iii. Operating systems: Windows® 10, macOS, and Linux.
iv. Working webcam.

b) Software Specification:

i. Python 3.6.0
ii. PyCharm
iii. E-Speak

Optional: Microsoft Teams, ZOOM, Google Meet etc.

c) Python Library Dependencies:

i. OpenCV-python

ii. Dlib
iii. SciPy
iv. Play sound
v. Os
vi. Time
vii. Distance

System Requirements:

1. Functional Requirements:

a) Calculating Real-Time Video:
The system should capture frames at 30 frames per second.

b) Processing the frames:
   a) Captured frames will be pre-processed i.e., converted into greyscale.
   b) This system will be responsible for finding facial, mouth and eye landmarks.

c) Capturing no. of Frames in which eyes are closed:
The system determines if the student is drowsy based upon whether these frames exceed the threshold value we set.

d) Generating Results/Output (Alarm):
Based on the output, the system will decide whether the student is drowsy and generate an alarm.

e) Closing Online Meeting Applications (MS Teams, Zoom, Google Meet):
Based on the output, the system will decide whether the student is drowsy and close the meeting apps.

2. Non-Functional Requirements:

a) Security:
The system will start on start-up and hence the student will have no control of the system other than turning the alarm off.

b) Portability:
It works on various meeting/classroom applications like MS Teams, Zoom Meeting app, Google Meet etc.

c) Usability:
The system is easy to learn and very efficient to use.

d) Cost Effective:
System is cost-effective.

Feasibility analysis & Report:

1. Technical Feasibility:

The system being developed is economic concerning student attendance. It is cost-effective in the sense that it has eliminated the bunking of the class.

The system is very responsive as it captures and processes 30 frames per second. It produces the real-time status of the student. The result obtained contains fewer errors and are highly accurate as the data are required.

2. Economic feasibility:

The technical requirement for the system is very low and economic and it does not use any other specific Hardware as the only Webcam is required.

3. Behavioural Feasibility:

The system working is quite easy to use and learn due to its simplicity. The user requires no special training for operating the system.

ASSIGNMENT- 2

Types of SRS diagrams and their brief description

22-01-2021

Types of SRS diagrams and their brief description:

Different types of diagrams required in an SRS document are -

1. Data Flow Diagram

The data flow diagram is a graphical representation of the flow of data in an information system. It is capable of depicting incoming data flow, outgoing data flow and stored data. The DFD does not mention anything about how data flows through the system.

2. E-R Diagram:

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. 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.

3. Flow Charts

A flowchart is a diagrammatic representation of the sequence of logical steps of a program. Flowcharts use simple geometric shapes to depict processes and arrows to show relationships and process/data flow.

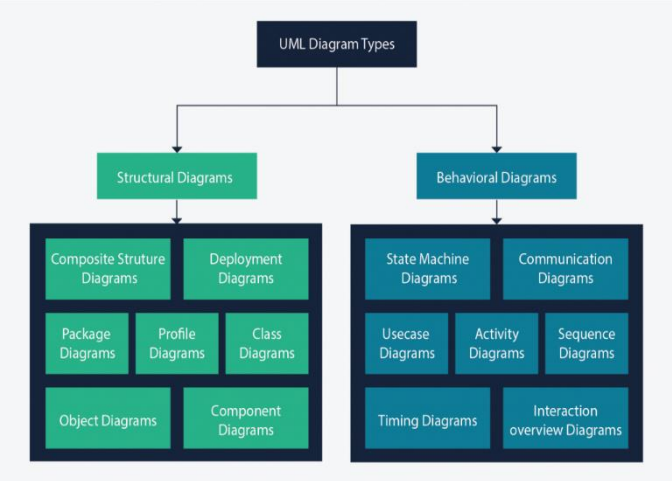
4. UML (Unified Modeling Language) Diagrams:

UML, short for Unified Modelling Language, is a standardized modelling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing, and documenting the artefacts of software systems, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part of developing object-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. There are *14 UML diagram types* to help you model these behaviours.

Types of UML Diagrams

There are two main categories; **structure diagrams** and **behavioural diagrams**. Click on the links to learn more about a specific diagram type.

- Structure Diagrams
  - Class Diagram
  - Component Diagram
  - Deployment Diagram
  - Object Diagram
  - Package Diagram
  - Profile Diagram
  - Composite Structure Diagram
- Behavioural Diagrams
  - Use Case Diagram
  - Activity Diagram
  - State Machine Diagram
  - Sequence Diagram
  - Communication Diagram
  - Interaction Overview Diagram
  - Timing Diagram



There are different types of diagrams that can be added to an SRS document. Five of them are discussed below:

a) Class Diagram:

The class diagram is a central modelling technique that runs through nearly all object-oriented methods. This diagram describes the types of objects in the system and various kinds of static relationships which exist between them.

b) Use Case Diagram:

A use-case model describes a system's functional requirements in terms of use cases. It is a model of the system's intended functionality (use cases) and its environment (actors). Use cases enable you to relate what you need from a system to how the system delivers on those needs.

c) Activity Diagram:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. It describes the flow of control of the target system, such as exploring complex business rules and operations, describing the use case also the business process. In the Unified Modelling Language, activity diagrams are intended to model both computational and organizational processes (i.e., workflows).

d) State Diagram:

A state diagram is a type of diagram used in UML to describe the behaviour of systems which is based on the concept of state diagrams by David Harel. State diagrams depict the permitted states and transitions as well as the events that affect these transitions. It helps to visualize the entire lifecycle of objects and thus help to provide a better understanding of state-based systems.

e) Sequence Diagram:

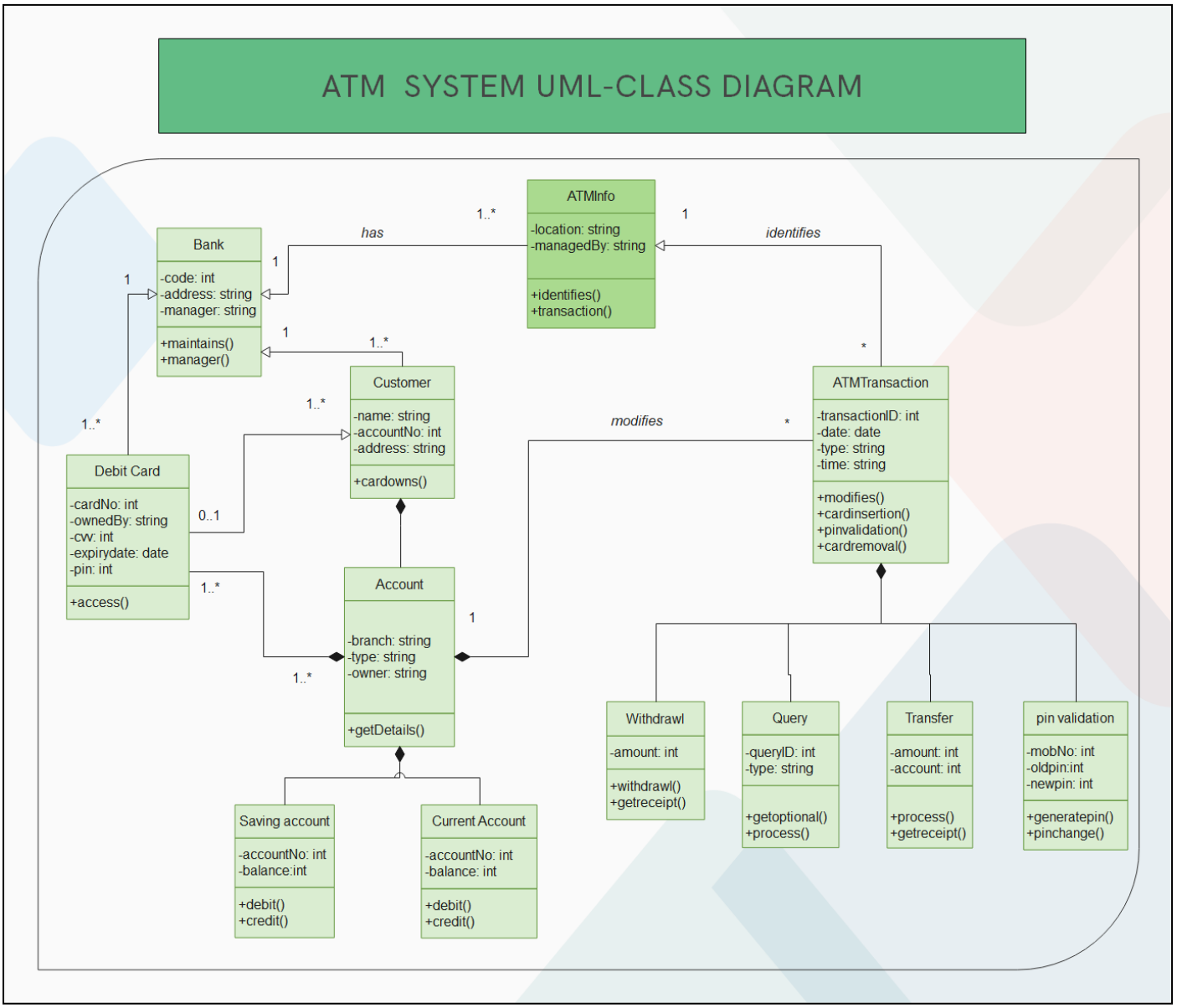
The Sequence diagrams in UML show how objects interact with each other and the order in which those interactions occur. It's important to note that they show the interactions for a particular scenario. The processes are represented vertically and interactions are shown as arrows.

ASSIGNMENT- 3 & 4

ATM UML Diagrams

1. Class Diagram:

a) Diagram:



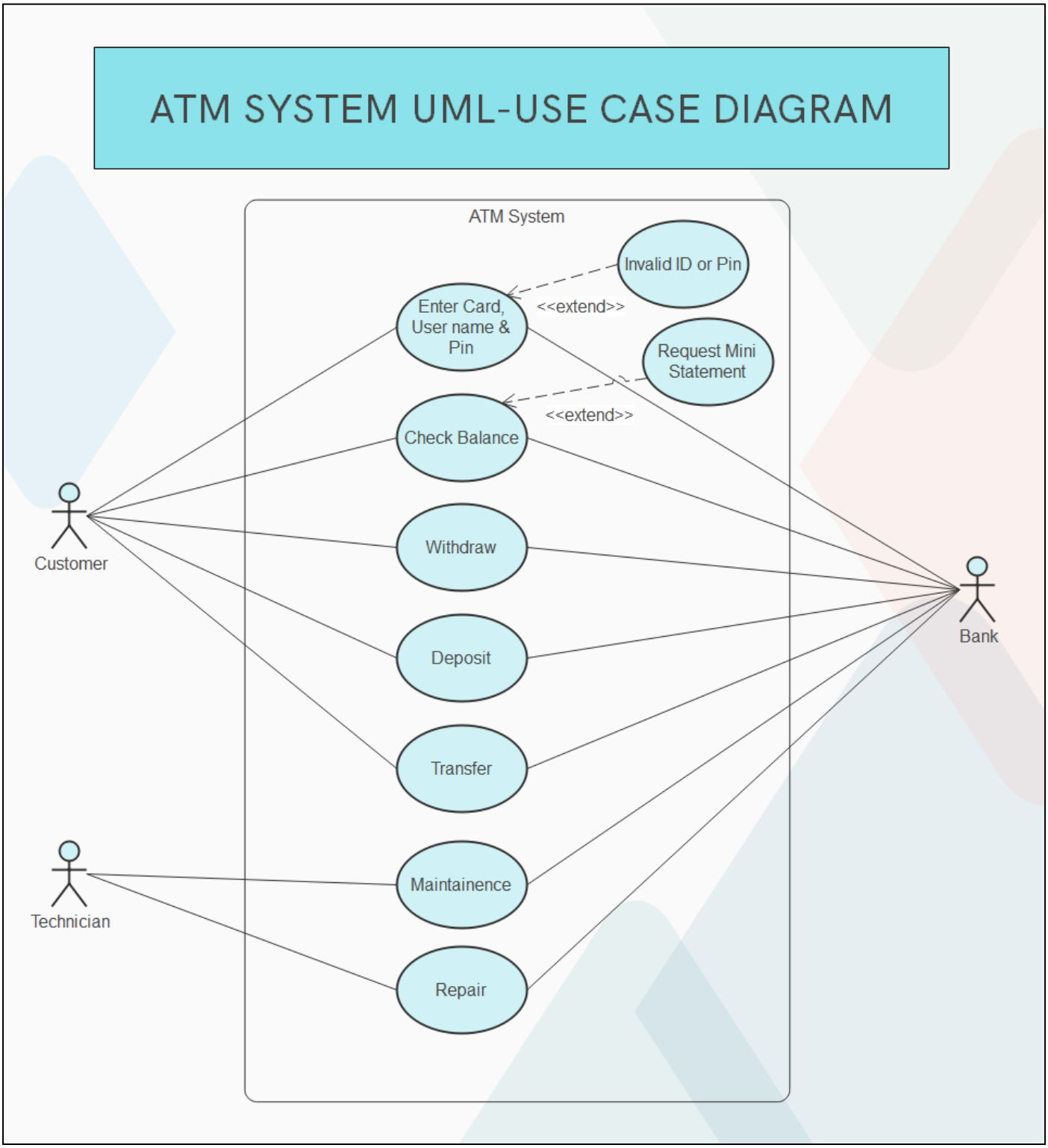
b) Description:

The Above Class Diagram of the ATM system includes all the Classes involved in The ATM and Banking System, namely: Bank, ATM Info, Debit Card, Customer, Account, ATM Transaction and their further subclasses have been shown in the diagram; with their respective attributes and methods; the relationships between different classes have been appropriately shown.

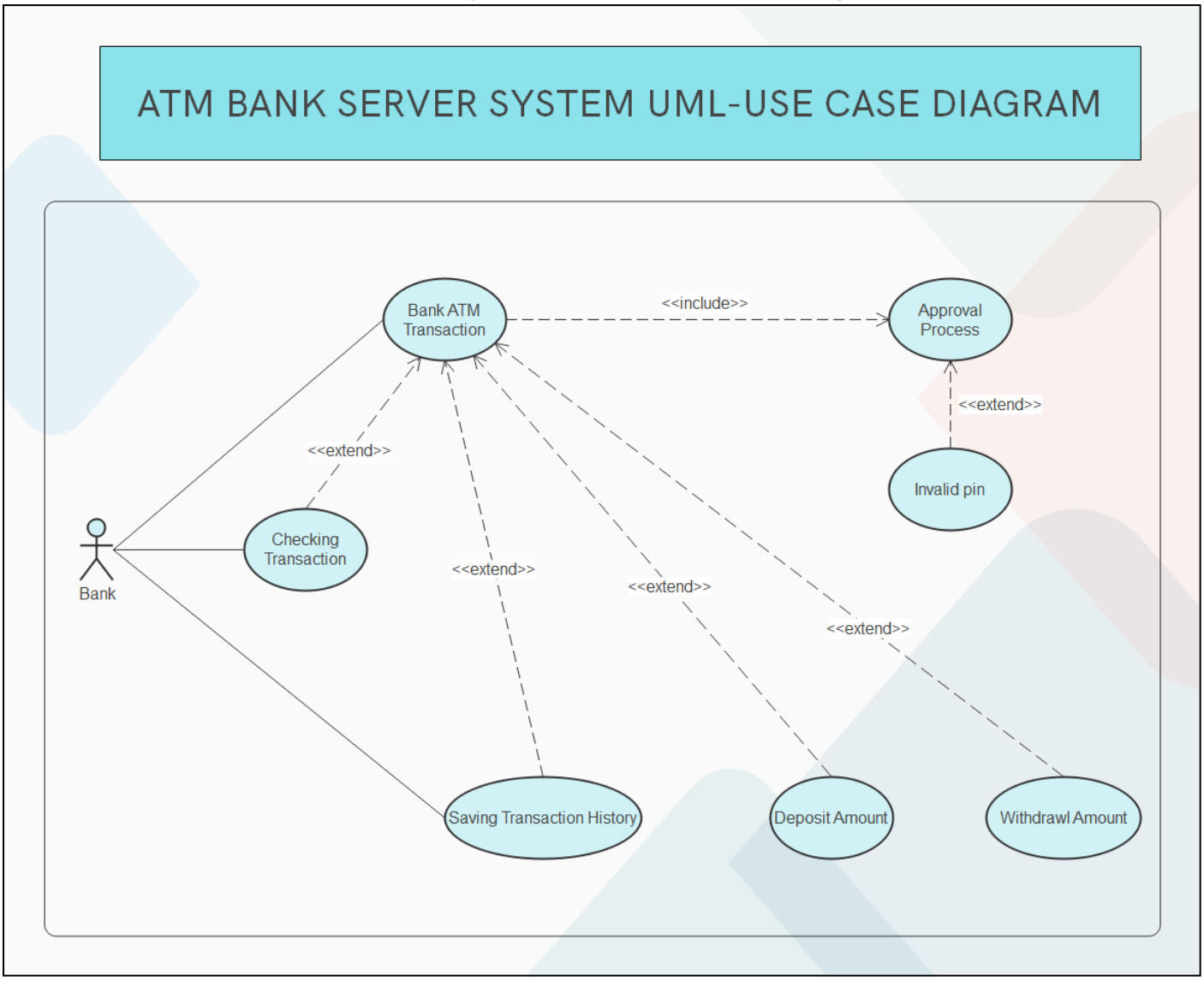
2. Use case Diagram:

a) Diagram:

i. ATM System UML – Use Case Diagram:



ii. ATM Bank Server System UML – Use Case Diagram:

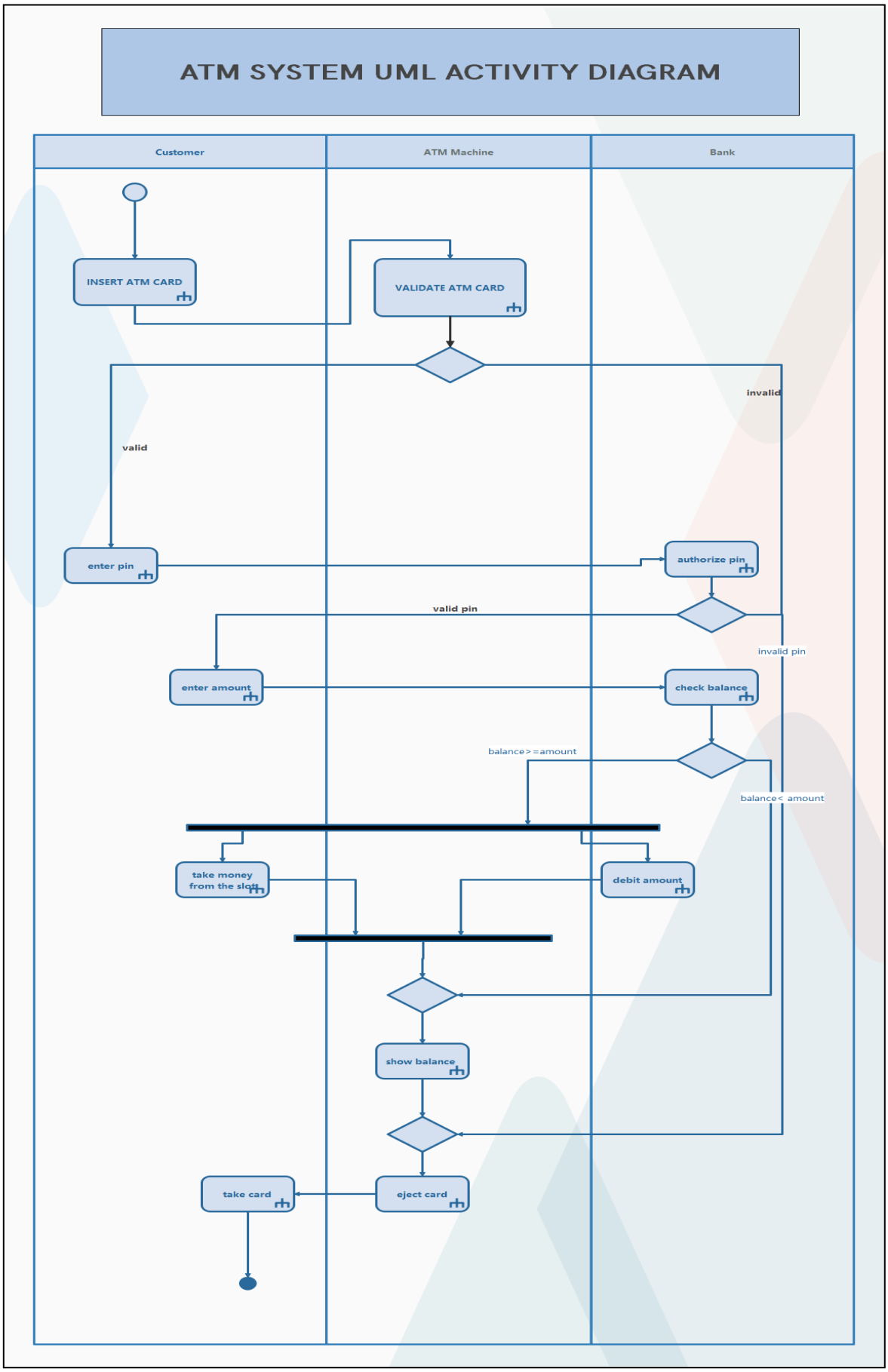


b) Description:

In the Above Diagram, the Customer (actor) uses a bank ATM to Check Balances of his/her bank accounts, Deposit Funds, Withdraw Cash and/or Transfer Funds (use cases). ATM Technician (actor) provides Maintenance and Repairs. All these use cases also involve Bank (actor) whether it is related to customer transactions or ATM servicing. All the possible Use Cases of the ATM system and Bank Server System are shown in the above diagrams.

3. Activity Diagram:

a) Diagram:



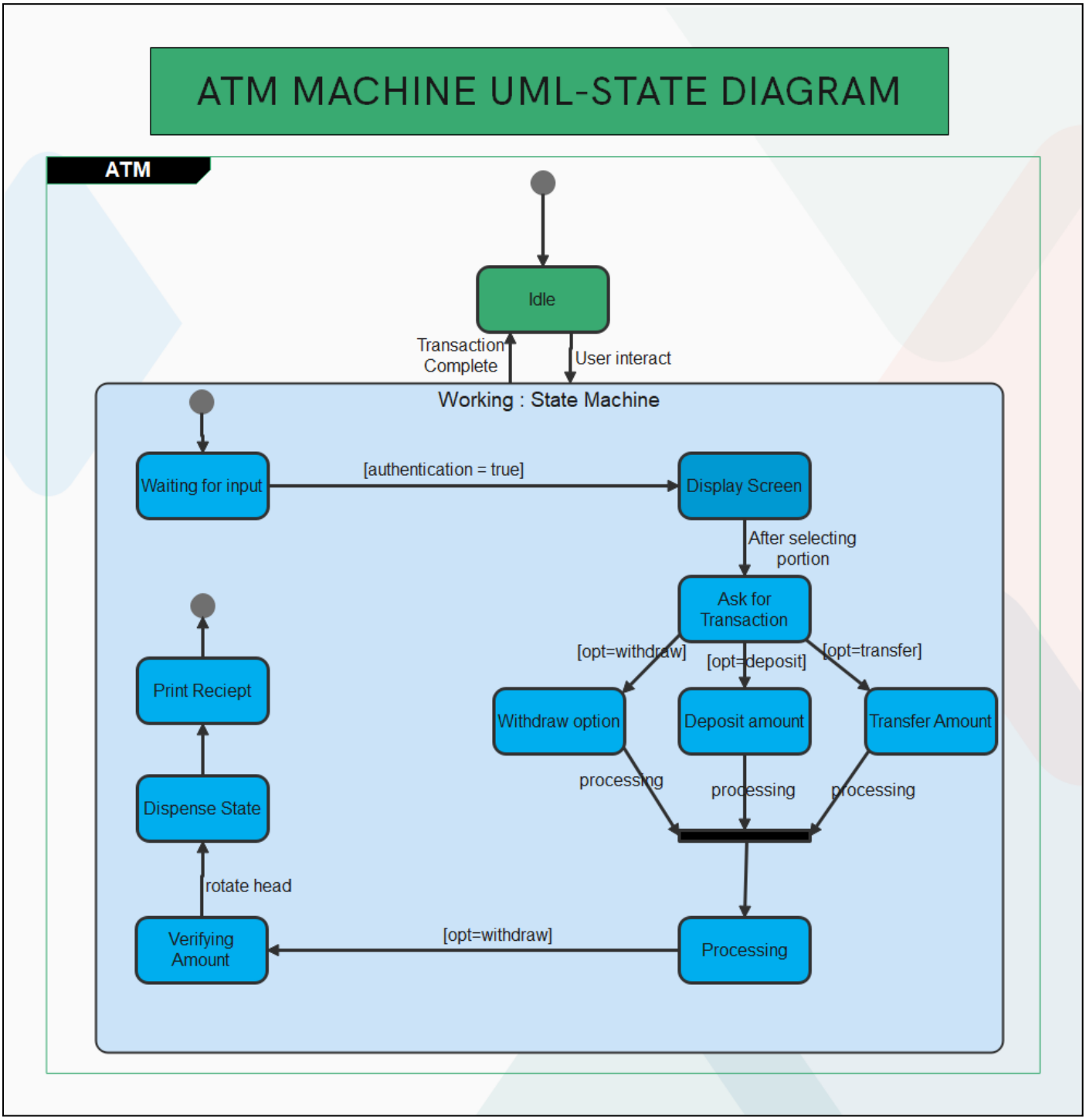
b) Description:

In the above, the flow of control of all the possible Activities of the ATM System is shown. As evident from the fig, At the start, the User Inserts the ATM card and enter the pin; the pin is validated; if the pin is valid, Then the ATM provides further transactions; or else the user is prompted about the pin is invalid, and is further asked to re-enter the pin. Now say, the user enters the correct PIN; So Now, he/she is again asked to enter the amount, and on entering the amount; again, the Account Balance is being checked and only if the Balance is greater than the amount being entered, the machine provides further options; or else it again notifies the user about the Amount being too high. Now if the user has sufficient balance, he/she is prompted about various options like cash withdrawal, account balance check, etc, and the user any of the options as per his / her requirements. and after the user's job being done, he/she ejects the ATM Card and the process ends.

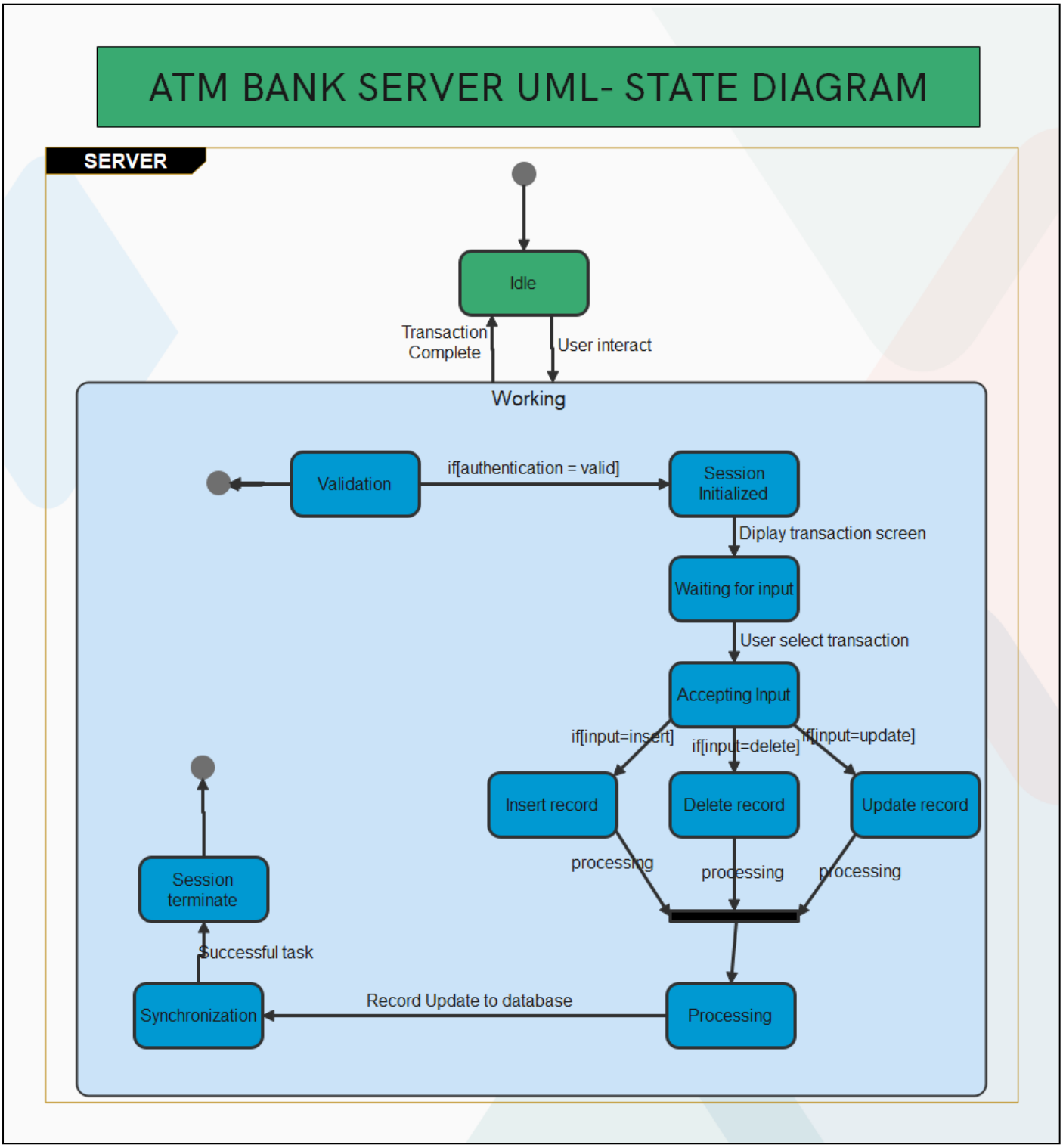
4. State Diagram:

a) Diagram:

i. ATM Machine UML – State Diagram:



ii. ATM Bank Server UML – State Diagram:

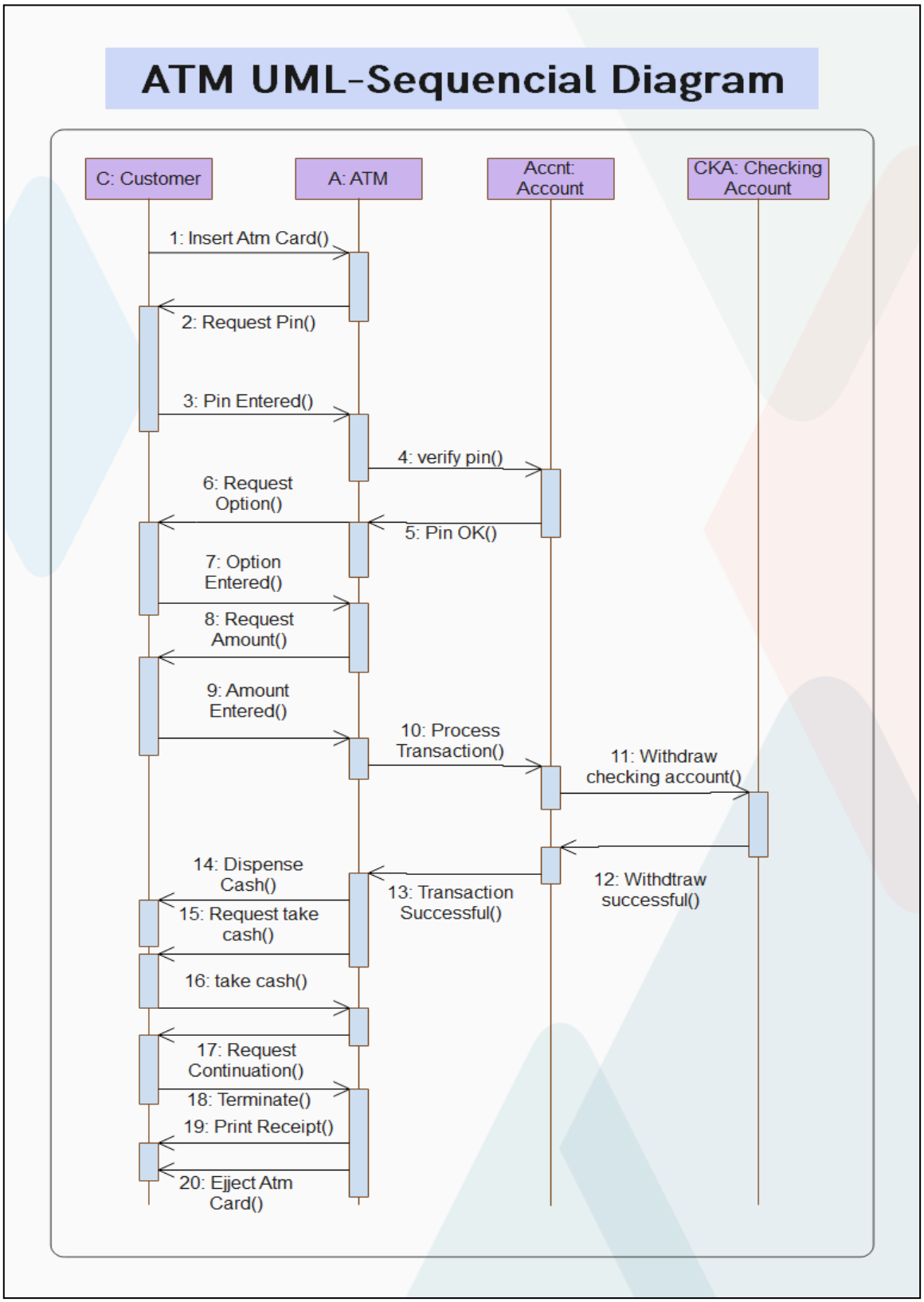


b) Description:

These diagrams in many ways, resembles the activity diagram, in many aspects of the Behavioural Aspects of the way an ATM works; here we talk in terms of states and define all the activities as STATE TRANSITIONS between an initial and final state.

5. Sequence Diagram:

a) Diagram



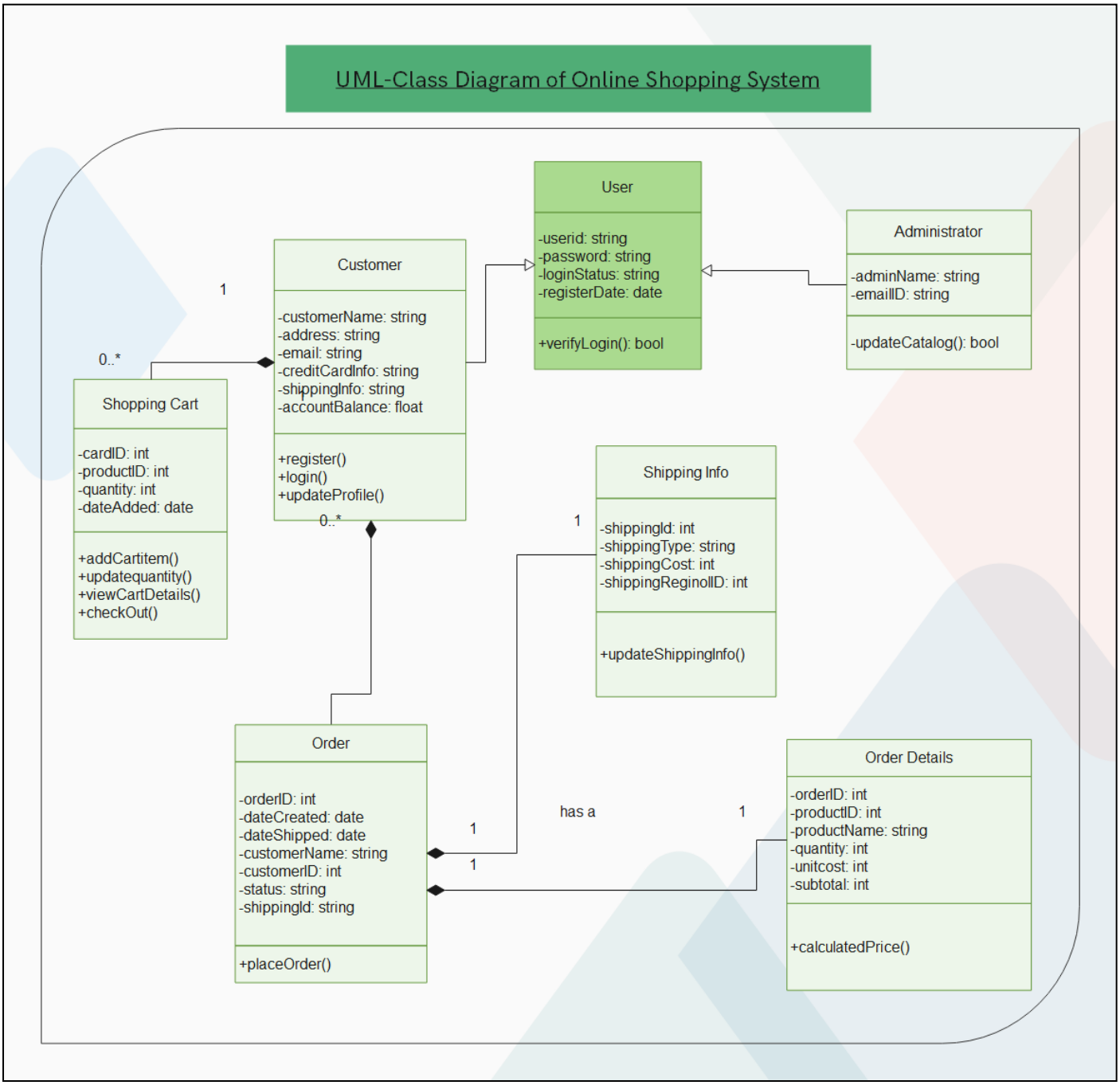
b) Description

As evident from the figure, all the sequences of operations: inserting ATM card, PIN verify, deposit, withdrawal, Balance check, etc are well described sequentially via the sequence diagram.

Online Shopping System UML Diagrams

1. Class Diagram:

a) Diagram:

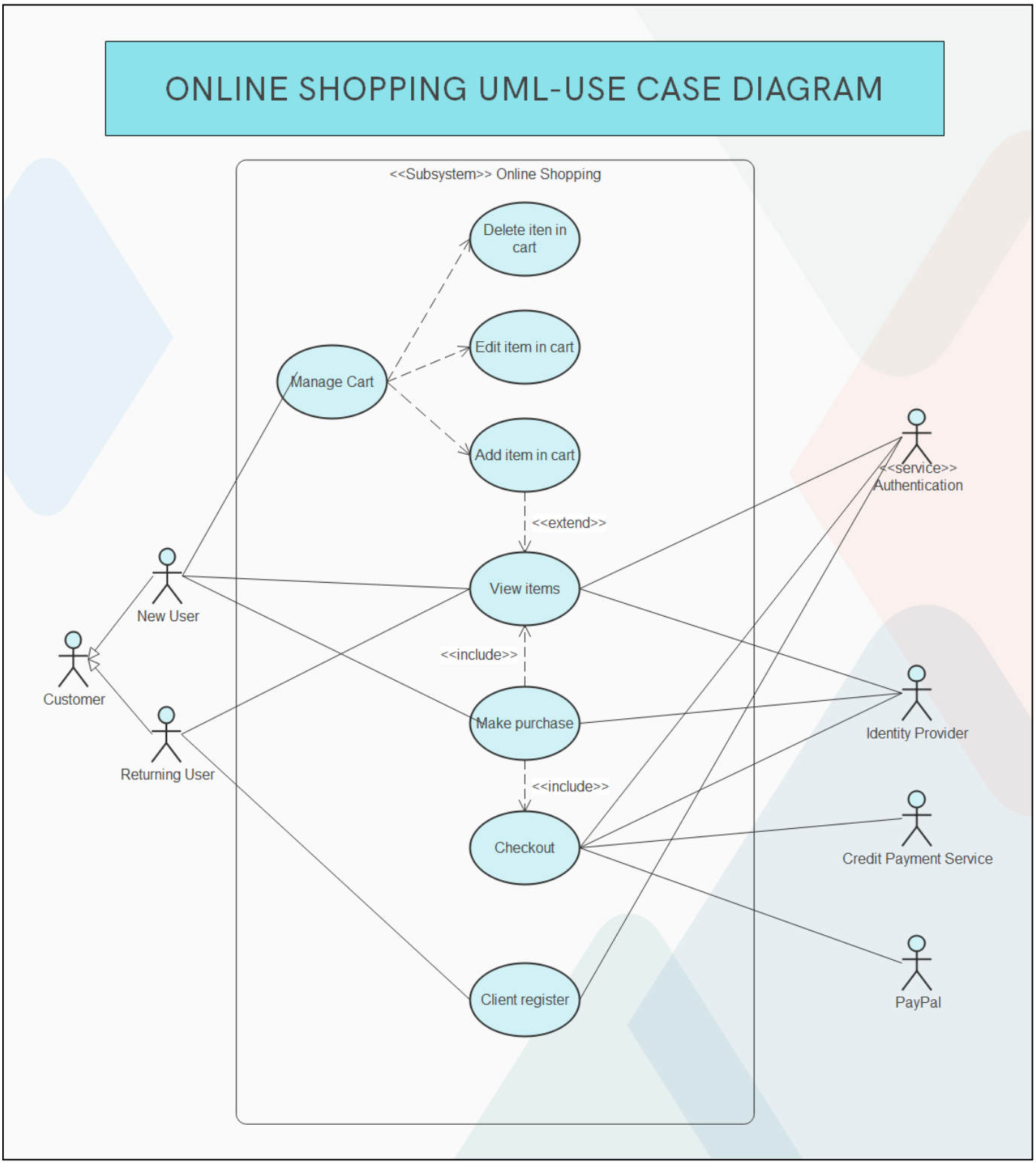


b) Description:

The above Class diagram for the Online Shopping System shows all the associated classes and their attributes & methods related to the System. The Classes of online Shopping System includes Customer, User, Administrator, Shopping Cart, Order, Shipping Info, Order Details. All the attributes and method of classes are well described in the diagram.

2. Use case Diagram:

a) Diagram:



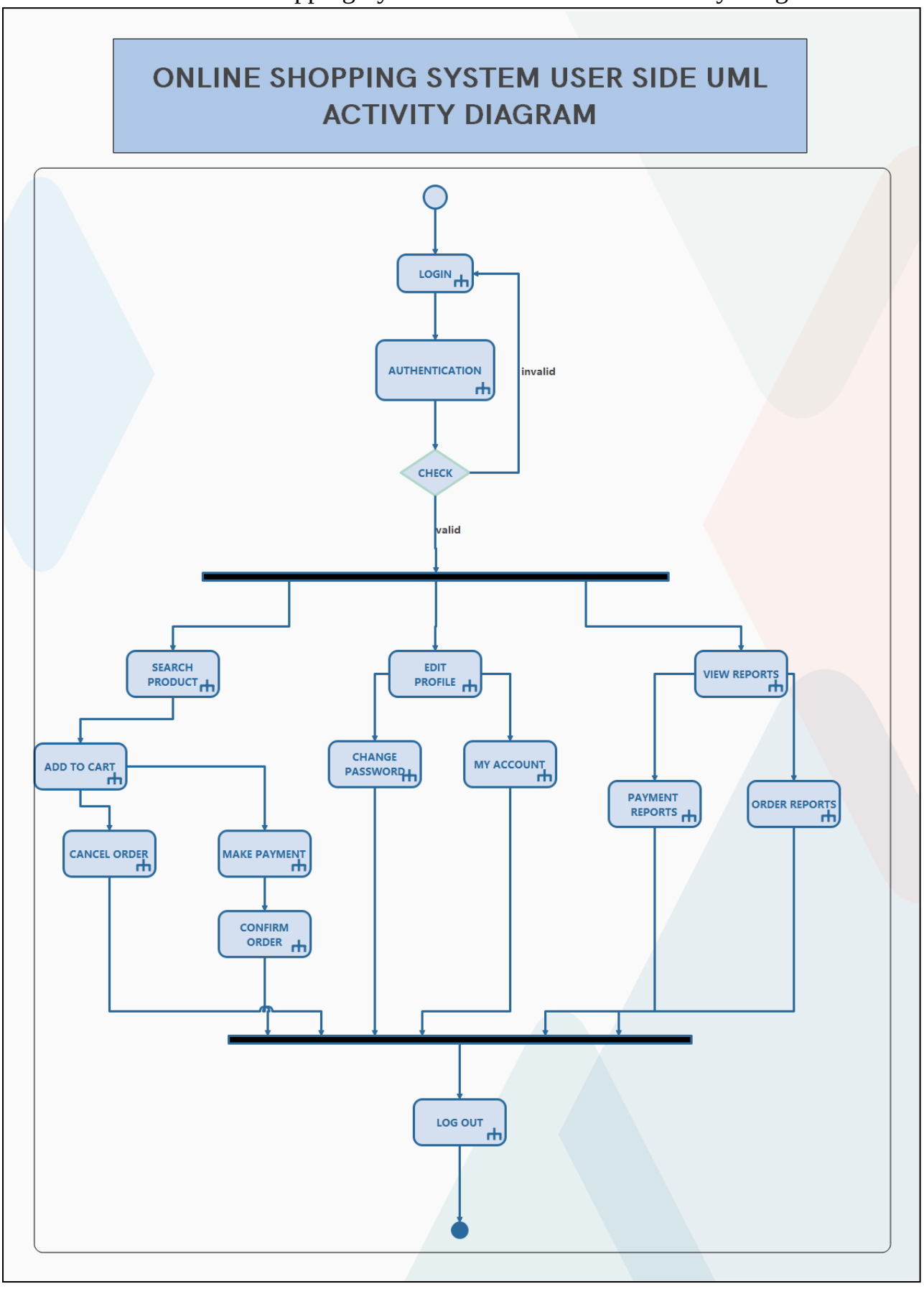
b) Description:

The above use case diagram shows the interaction of the customer with the online shopping system site. The customer (actor) uses some website to make online purchases. The Top-Level Use Cases are View Items, Make Purchase, Client Register, Manage Cart. A detailed description of all use cases is shown in the diagram.

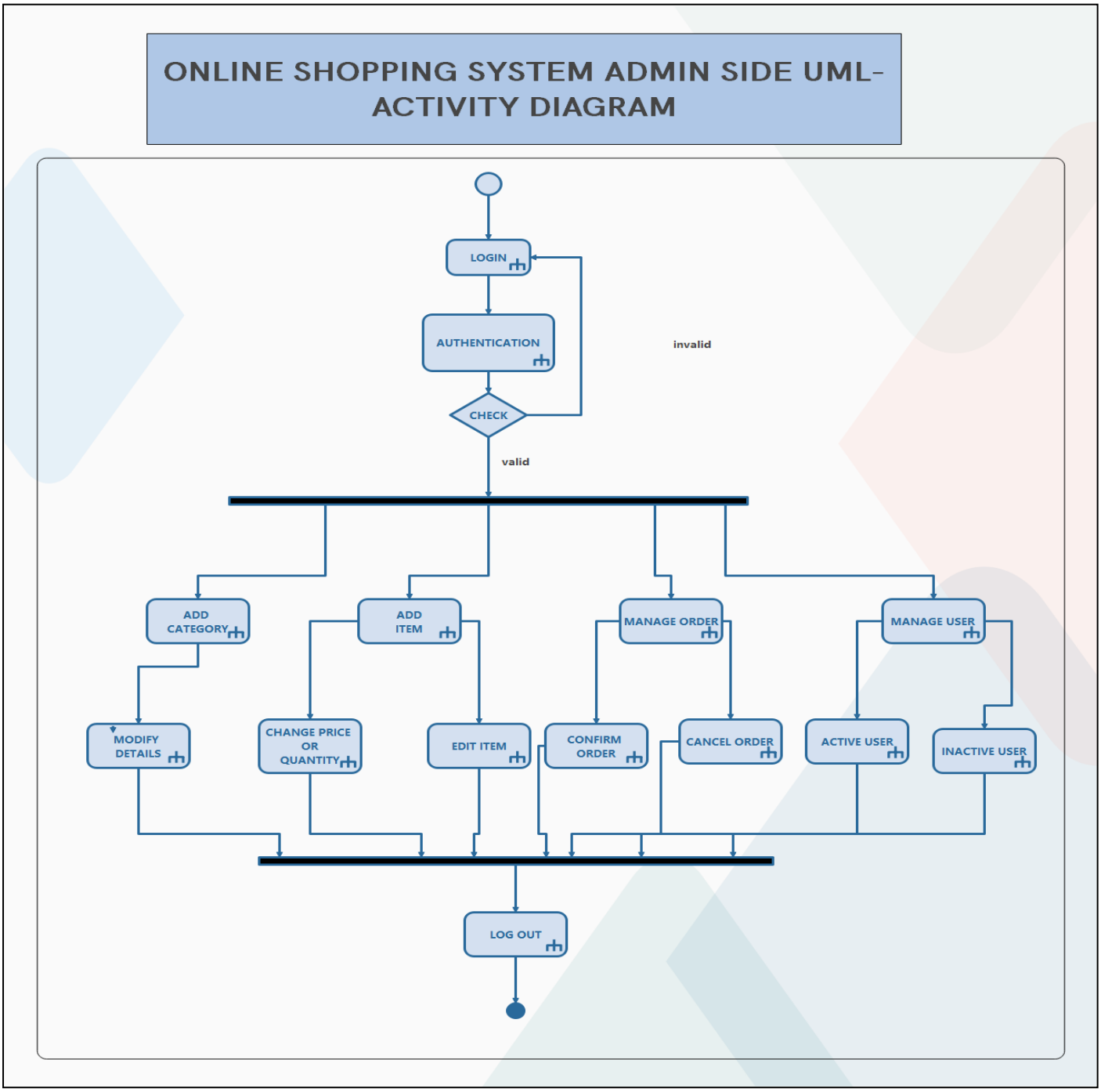
3. Activity Diagram:

a) Diagram:

i. Online Shopping System User Side UML - Activity Diagram:



ii. Online Shopping System Admin Side UML - Activity Diagram:



b) Description:

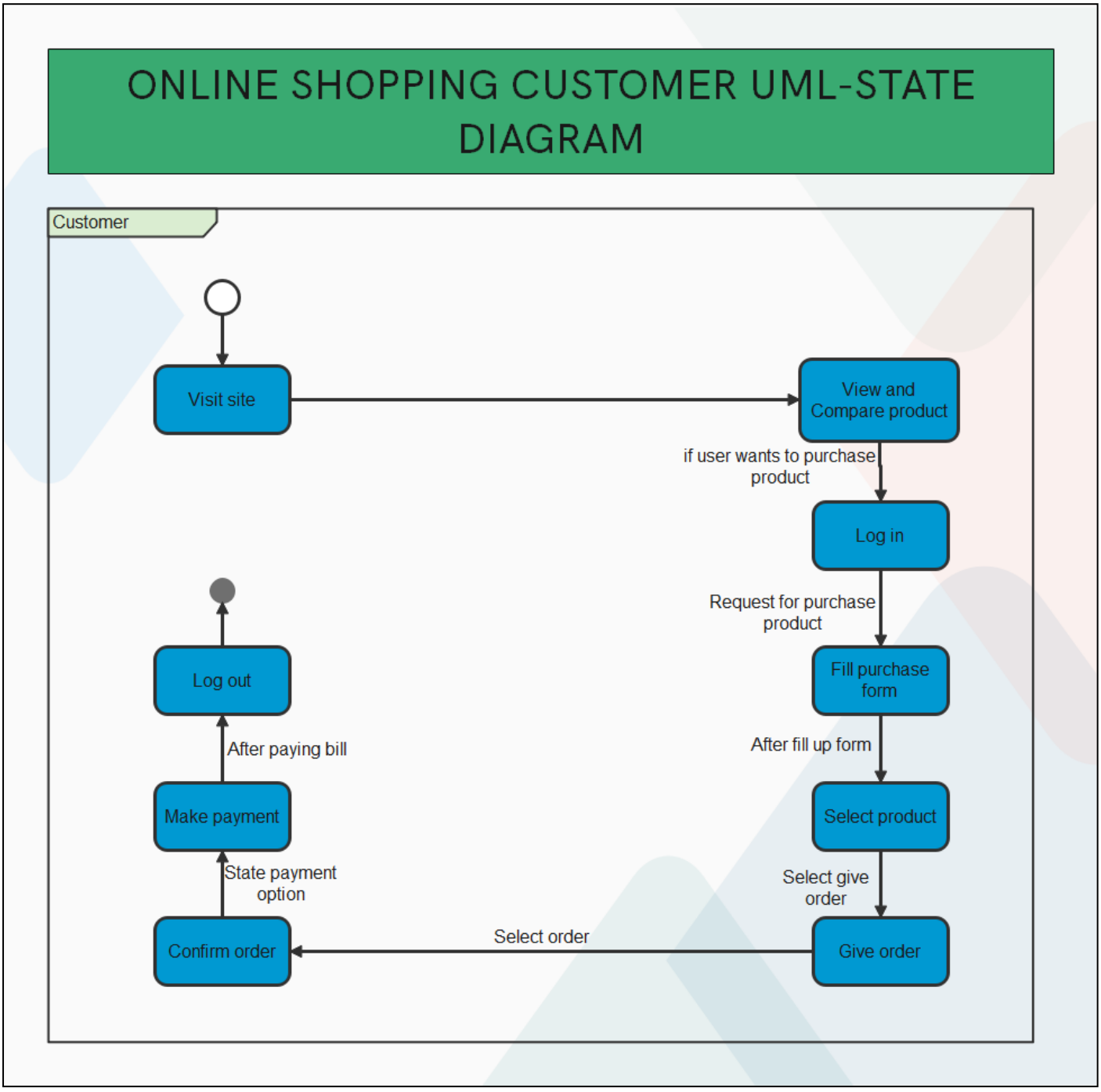
The User firstly Logins and registers himself/herself on the Online Shopping site and then User can perform various activities like the searching product, adding it to the cart, checkout, edit profile, view reports etc. These various activities are shown in the above user side activity diagram

The Admin can perform various tasks like adding categories of items, modifying the details, etc. The admin can add additional items and change the price and quantities, and can also manage the orders placed by choosing to confirm them or cancel them. Admin also has the powers to manage the users and activate/ inactivate the user. These various activities are shown in the above admin side activity diagram

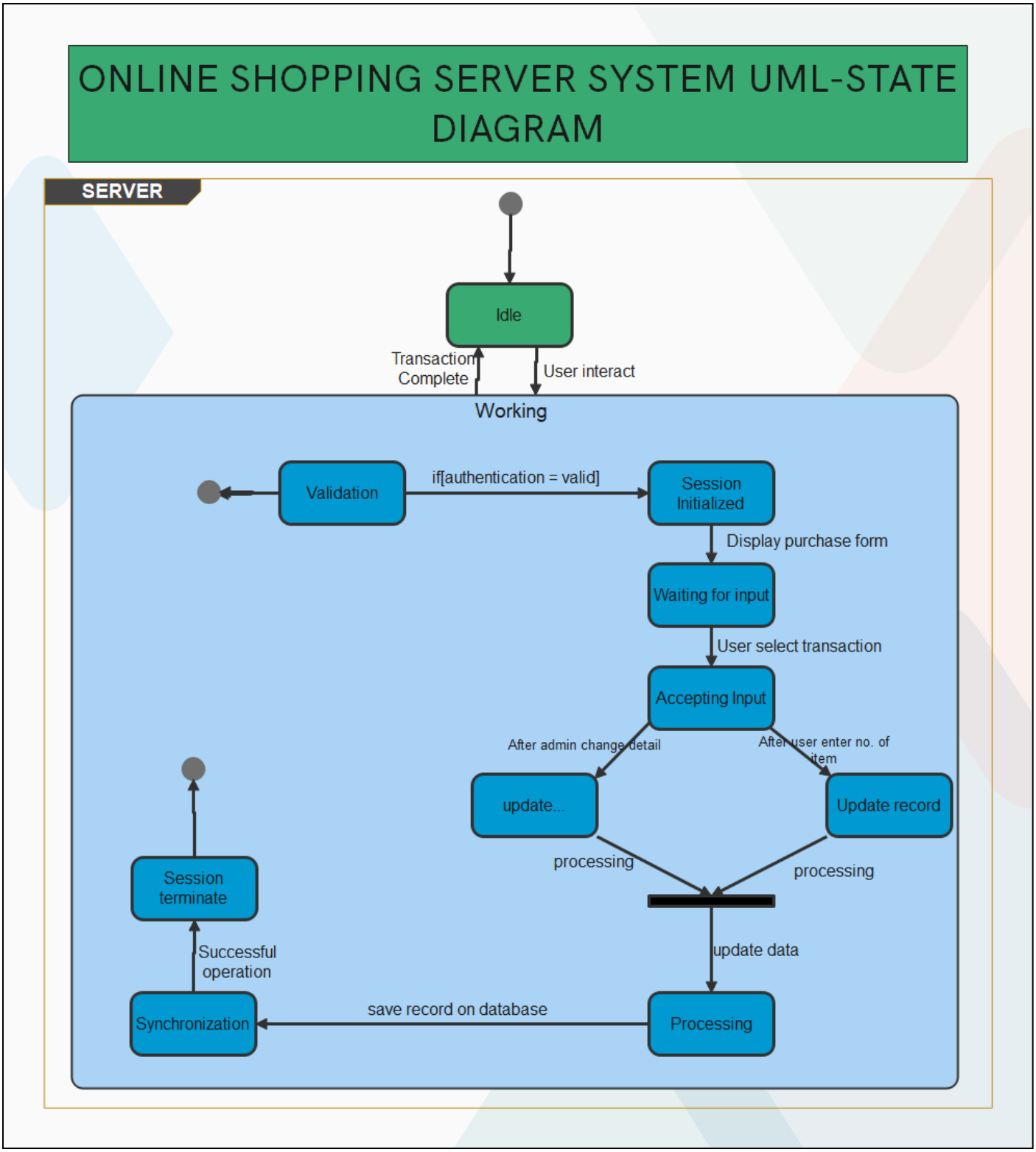
4. State Diagram:

a) Diagram:

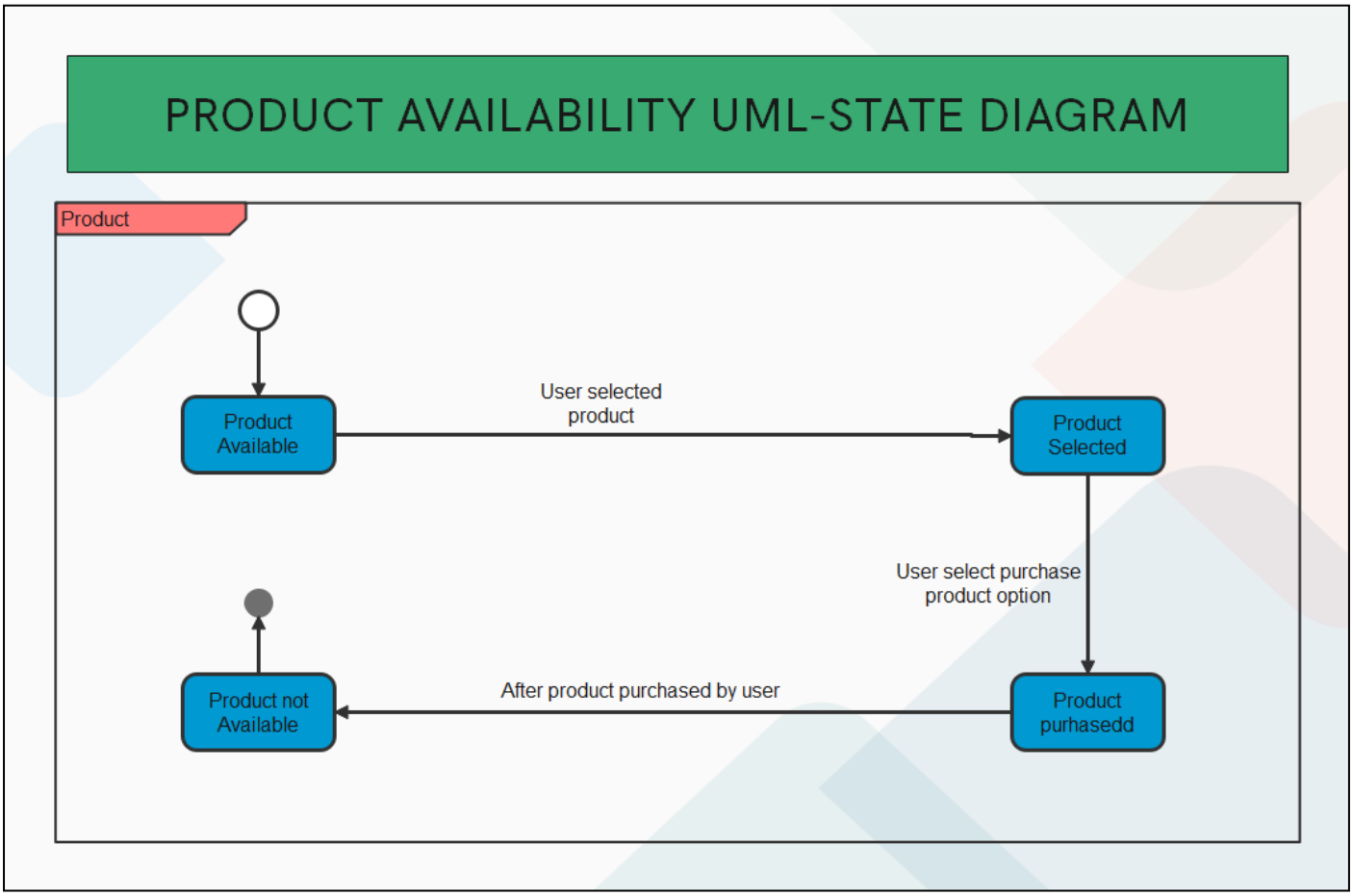
i. Online Shopping Customer UML – State Diagram:



ii. Online Shopping Customer UML – State Diagram:



iii. Product Availability UML – State Diagram:



b) Description:

The above State Diagrams explains all the transitions and activities intertwined between the finite states possible in an online shopping system.

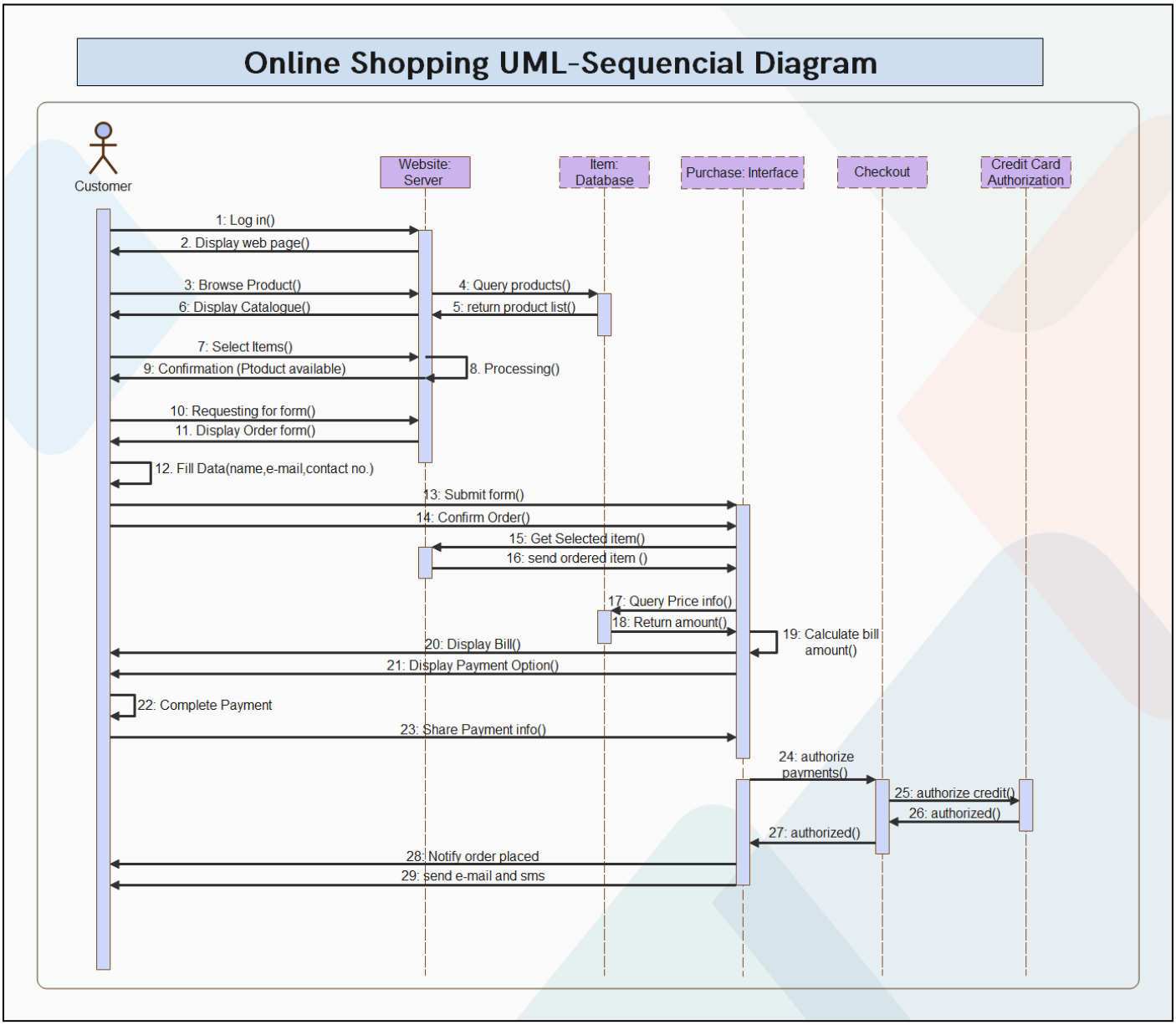
The User makes transitions to the Register state upon visiting the site for the first time. If he/she already has an account, he/she will transit to Login state; and can go to Search Product State by clicking the search option. If the desired product found, the user goes to the View Item state; if the user likes the item, he/she can add it to the cart and can even decide to buy it or cancel the item, as shown by the respective states. If chooses to buy, the user can make payment and go for the Checkout state.

The online shopping server system and product availability state diagrams also show the respective states of the server and product availability which includes authentication, processing, updating record, synchronization, and checking the availability of product etc.

All these states are well described in the above diagrams

5. Sequence Diagram:

a) Diagram:



b) Description:

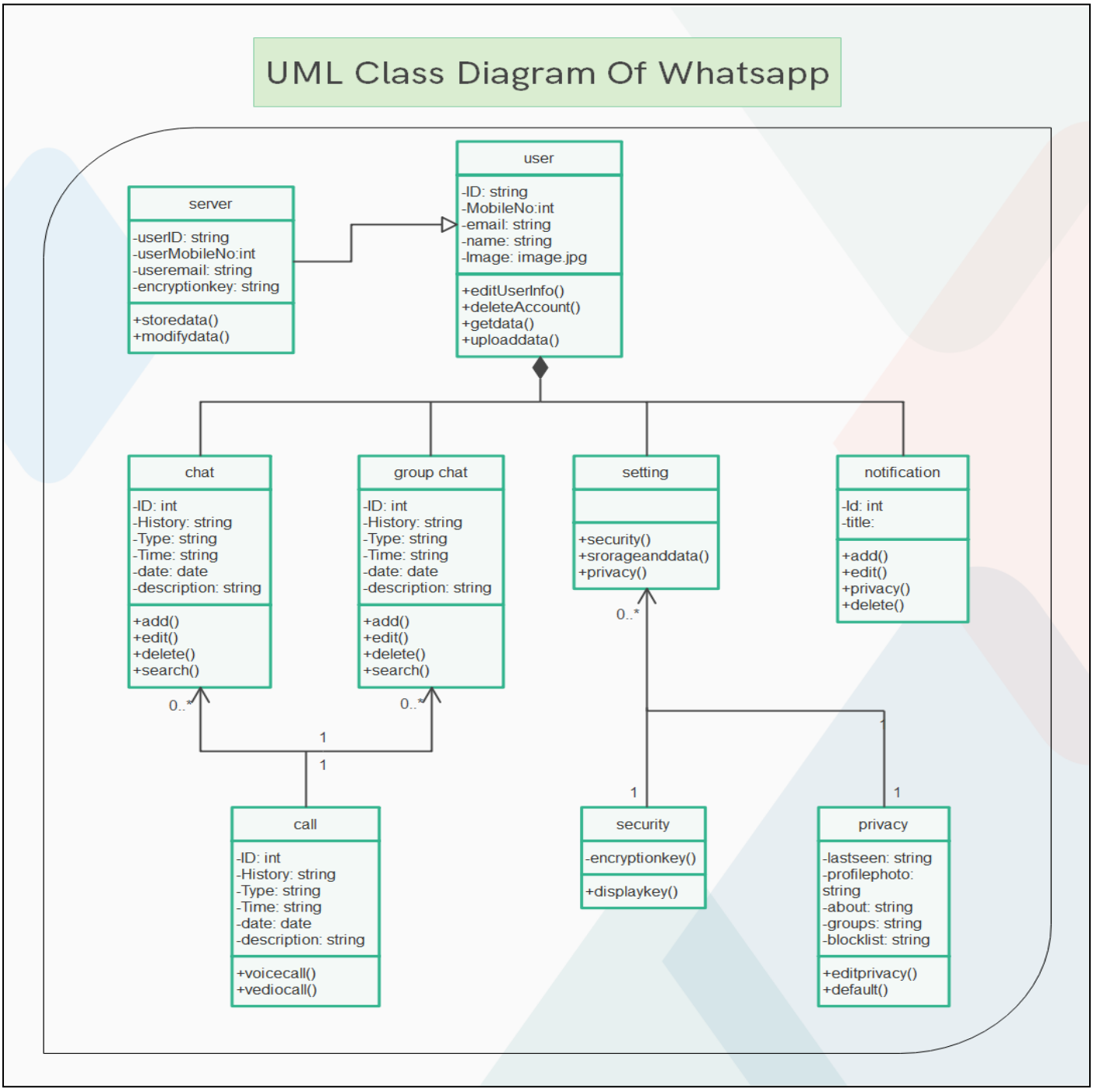
The Sequence Diagram depicts the sequence of activities for the online shopping system sequentially.

After logging in, the user can browse, select and add an item to the cart, remove an item from the cart or choose to display the cart. If the user goes for buying an item, he/she has to specify the payment method and do the payment. After successful payment, his / her order gets placed, and then afterwards the user can review and/or update his order status.

WhatsApp UML Diagrams

1. Class Diagram:

a) Diagram:

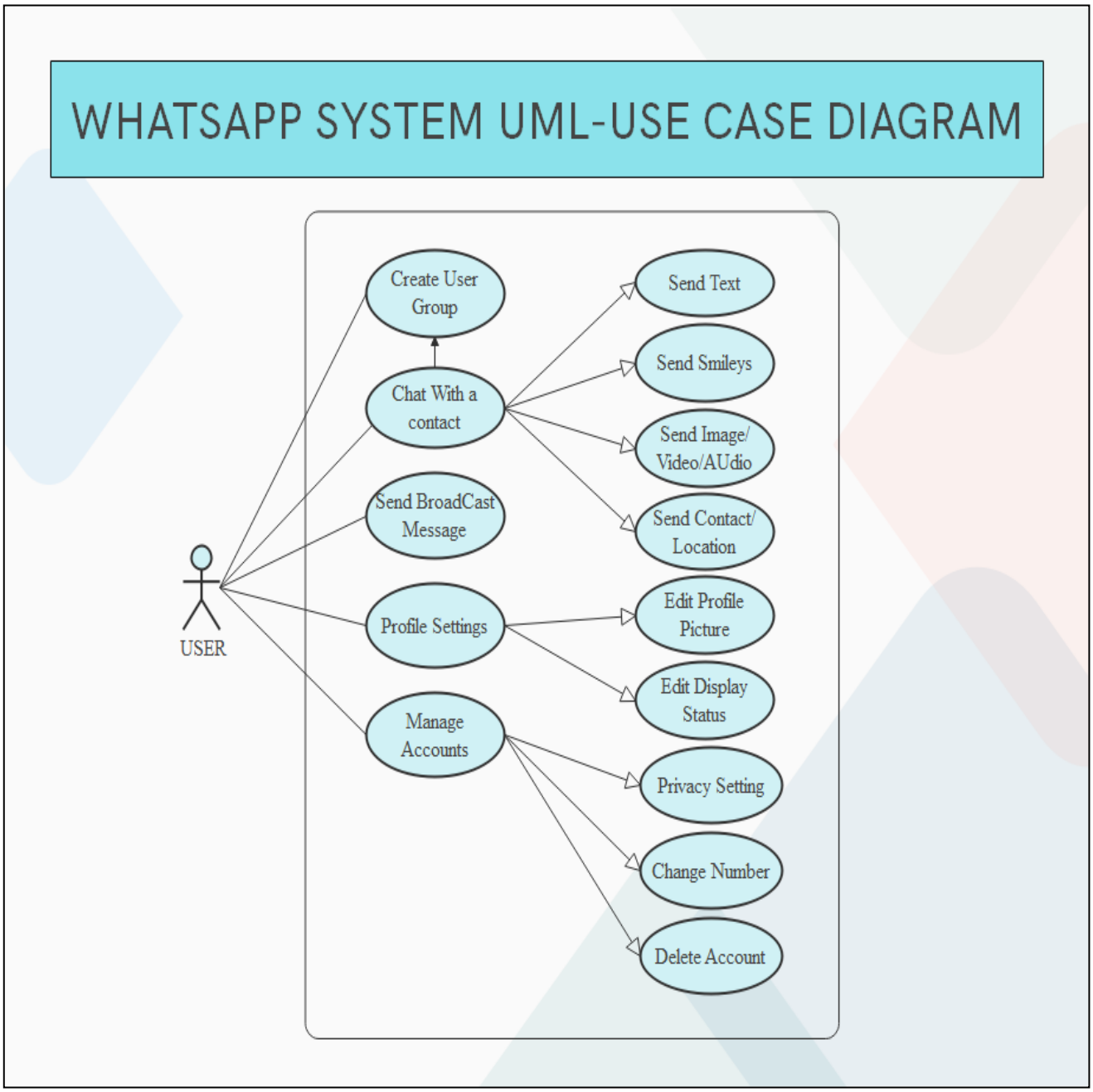


b) Description:

The above Class Diagram of the WhatsApp System gives us a structural view of the System. Various Classes and their associated attributes and methods have been portrayed in the diagram. Classes: User, Server, chat, group chat, call, setting, notification, security, privacy, etc. All the Relationships between the classes also have been shown appropriately.

2. Use case Diagram:

a) Diagram:

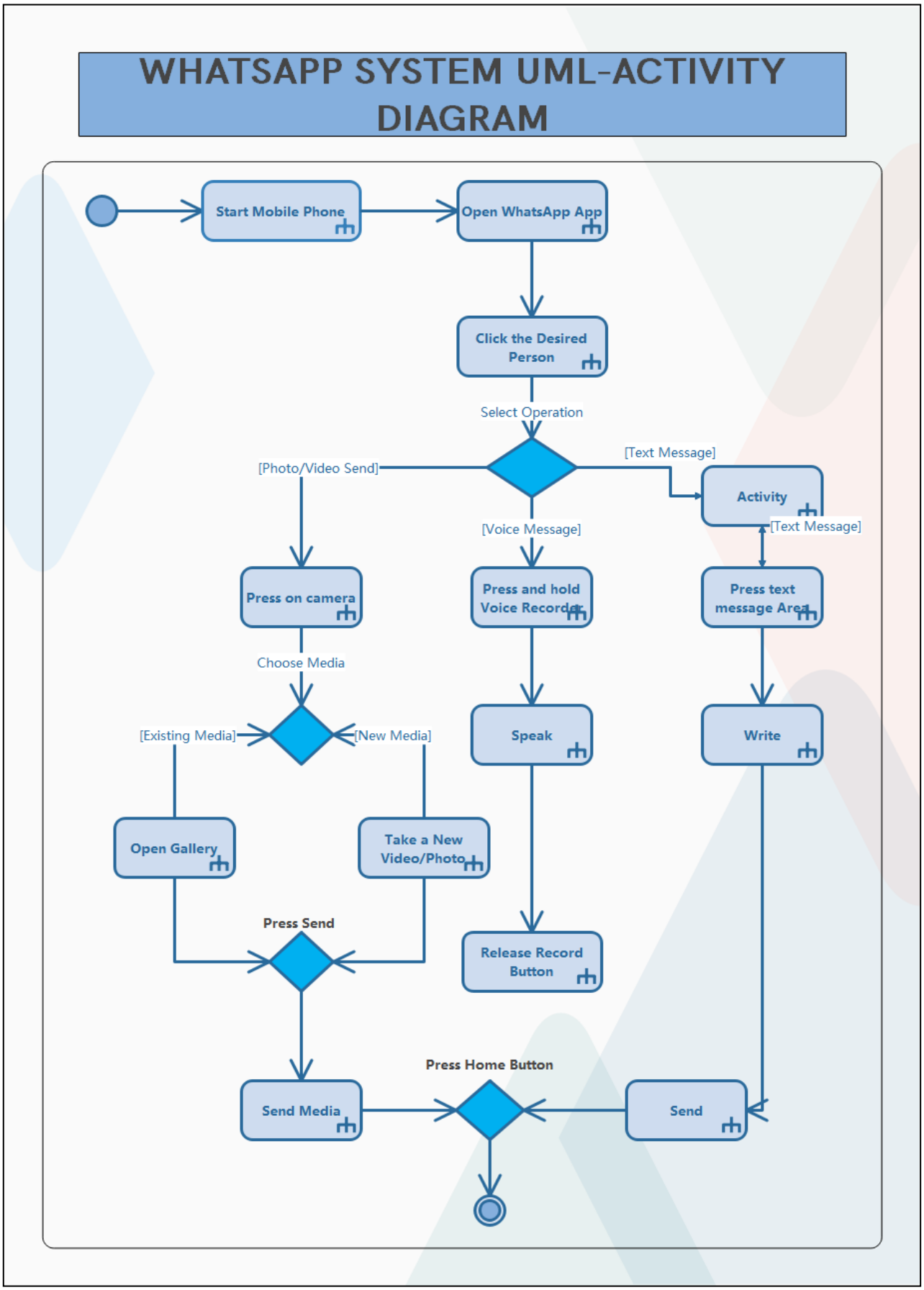


b) Description:

The Use Case Diagram shows all the possible scenarios that can occur in the interaction of the User and WhatsApp System. It shows how the user interacts with the system and his relationship with different entities of the system as per the situation. The user can broadly perform activities like: Managing account, profile settings, chatting with a contact, group creation, sending the broadcast message, etc. In the case of Chatting, the user share smileys, text message, audio/video/images, or contact location; In the case of Profile settings, the user can edit his / her profile and/or display status. While in the Manage Accounts case, the user can manage his / her privacy, change Number and/or delete account.

3. Activity Diagram:

a) Diagram:



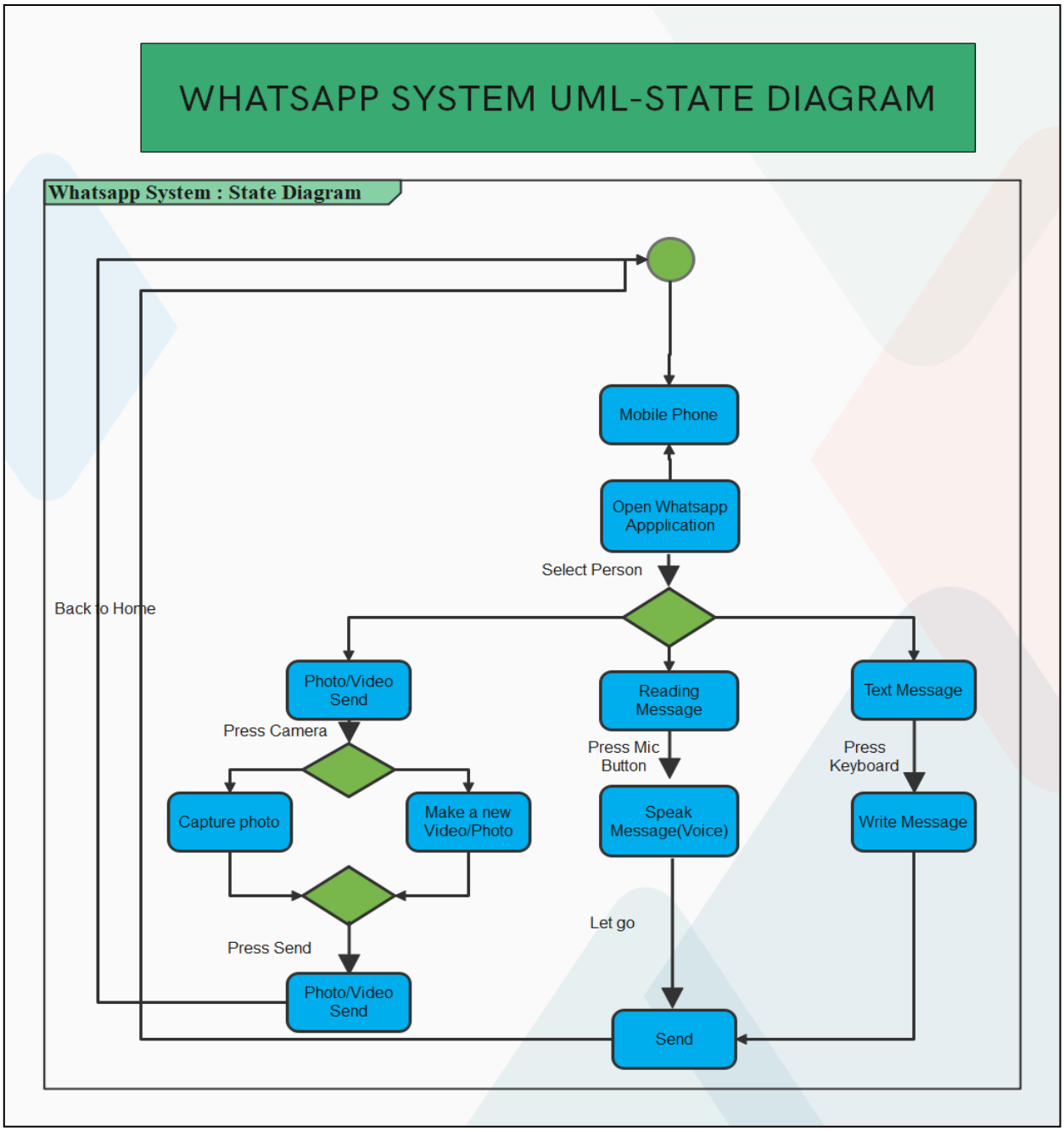
b) Description

The Above Activity Diagram shows all the activities and behavioural aspects of the working of The WHATSAPP system.

It Depicts the control flow right from starting to use WhatsApp to exiting the application. When a user downloads WhatsApp for the first time, he/she need to sign up and verify an OTP sent to his mobile number which is linked to WhatsApp. Now signed up, the user can effectively use WhatsApp further. WhatsApp allows the user to do messaging and voice calls; the process of message and voice call can be combined into one activity while voice message and text message can be combined into another activity. Now after performing the Chatting, the user may continue to use WhatsApp or exit the app!

4. State Diagram:

a) Diagram:



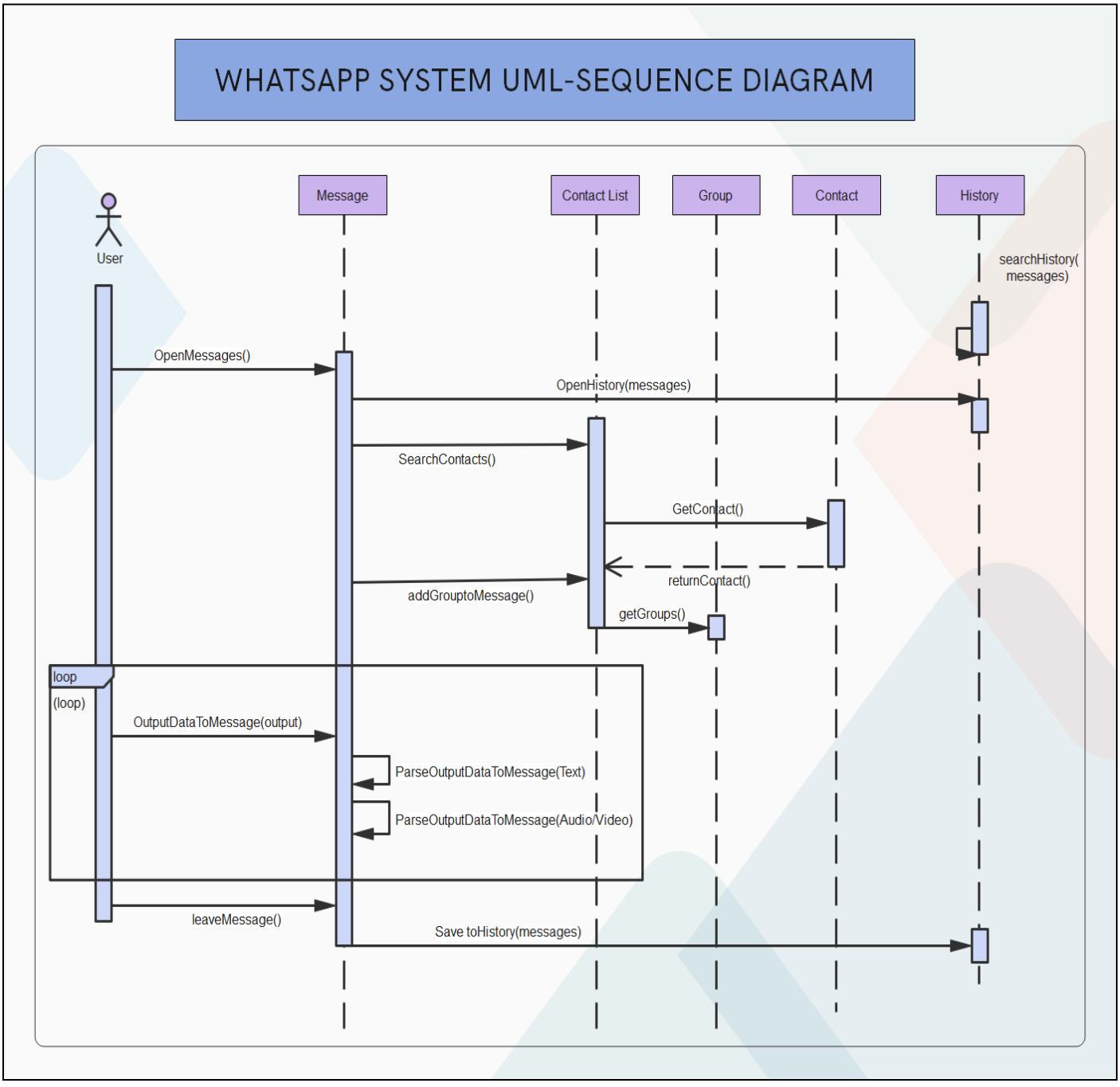
b) Description:

The State Diagram of WhatsApp shows the various functionalities of the application as finite State Transitions between initial and final states.

We make the first transition as soon as we start using WhatsApp via mobile (Android/iOS) / Windows. When we open WhatsApp, we move to the Open WhatsApp State. Now the user is supposed to select a contact for chatting; after the selection, the user can do various things like sending text/photo/videos, reading the selected person's messages etc; all the activities associated with these tasks like capturing photo / making a new video, recording a voice message, writing a text message, etc have been detected and represented in the above diagram. After making the concerned decision, we can go to the Send State; and when done with sending; we can revert to the initial state, and continue the process or take a halt.

5. Sequence Diagram:

a) Diagram:



b) Description

The Above Diagram depicts the sequence of operations and functions occurring inside the WhatsApp system sequentially.

Dining Philosophers Problem Activity Diagram

a) Problem Statement:

Construct an activity diagram for the following scenario There are 5 philosophers and 5 forks around a circular table. Each philosopher has access to 2 forks, one on either side. Each fork is shared by 2 philosophers. Each fork may be either on the table or in use by one philosopher. A philosopher must have 2 forks to eat. The Problem can be depicted as follows:

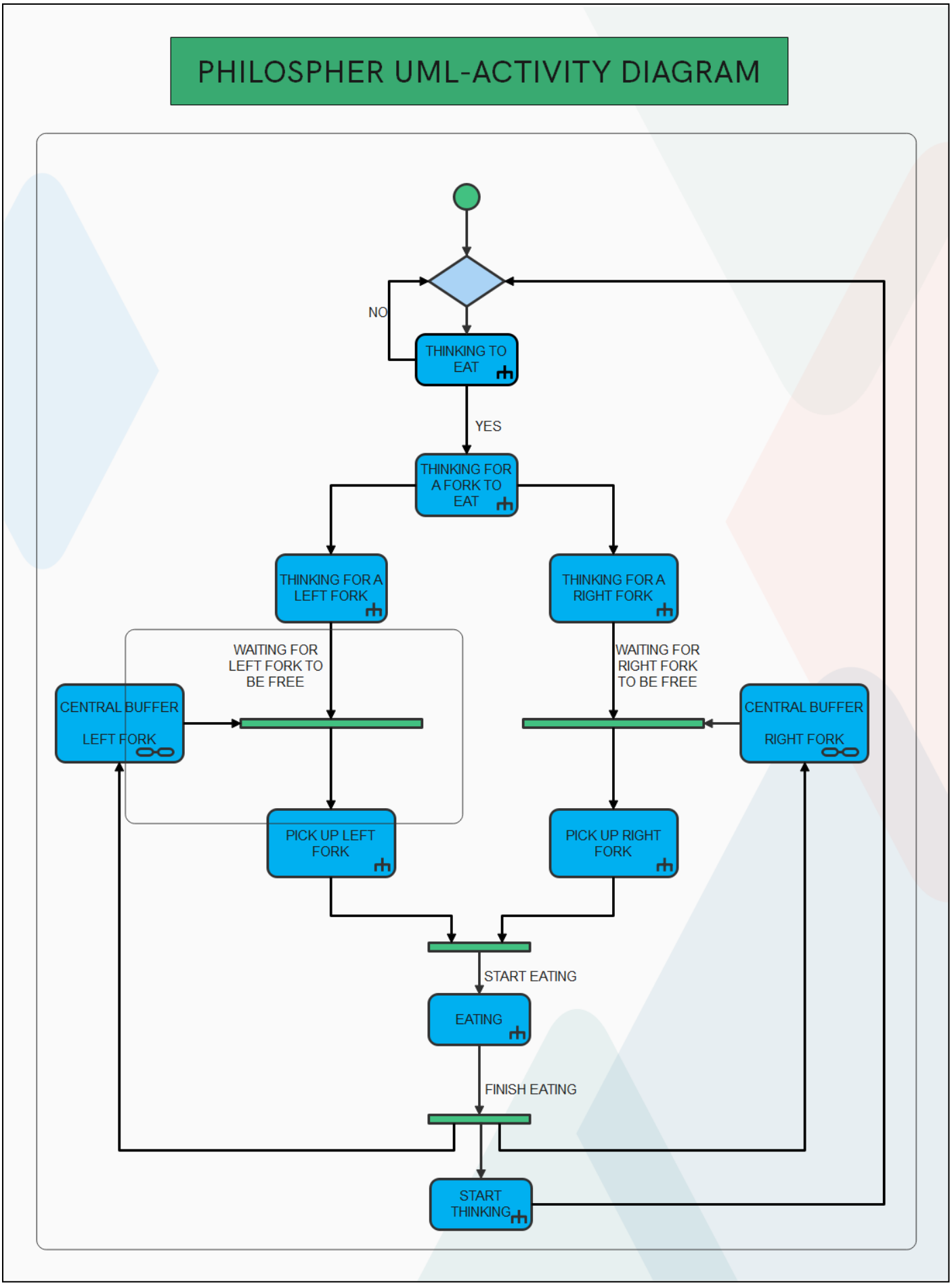
b) Description:

The dining philosopher's problem is the classical problem of synchronization which says that 5 philosophers are sitting around a circular table and their job is to think and eat alternatively. A bowl of noodles is placed at the centre of the table along with five chopsticks for each of the philosophers. To eat a philosopher needs both their right and a left chopstick. A philosopher can only eat if both immediate left and right chopsticks of the philosopher is available. In case if both immediate left and right chopsticks of the philosopher are not available then the philosopher puts down their (either left or right) chopstick and starts thinking again. The dining philosopher demonstrates a large class of concurrency control problems hence it's a classic synchronization problem.

The below figure shows the activity diagram for the dining philosophers' problem. We have shown only one philosopher, and for others, it will be the same by making copies and connecting them with the common forks.

In the below figure we can identify 2 characteristics of the Philosophers, i.e., THINK and EAT. In the THINK phase, the philosopher will Not eat but only Think, while in the EAT phase, he eats. So, it can be inferred that a Philosopher can eat only if both his adjacent fellows are thinking as then he can use the adjacent forks to do so. So, in this way, all the philosophers can eat following the think – eat pattern as depicted in the above diagram. Hence, we have proposed a solution through an activity diagram to the dining philosopher's problem.

c) Diagram


PHILOSPHER UML-ACTIVITY DIAGRAM
NO
THINKING TO EAT
YES
THINKING FOR A FORK TO EAT
THINKING FOR A LEFT FORK
THINKING FOR A RIGHT FORK
WAITING FOR LEFT FORK TO BE FREE
WAITING FOR RIGHT FORK TO BE FREE
CENTRAL BUFFER LEFT FORK
CENTRAL BUFFER RIGHT FORK
PICK UP LEFT FORK
PICK UP RIGHT FORK
START EATING
EATING
FINISH EATING
START THINKING

Social Networking Website Activity Diagram

a) Problem Statement:

Construct an activity diagram for the following modules for a Social Networking Website:

1. Friend Request Module
2. Messenger Module
3. Friend Search Module
4. Post Activity Module
5. Share Activity Module

b) Description

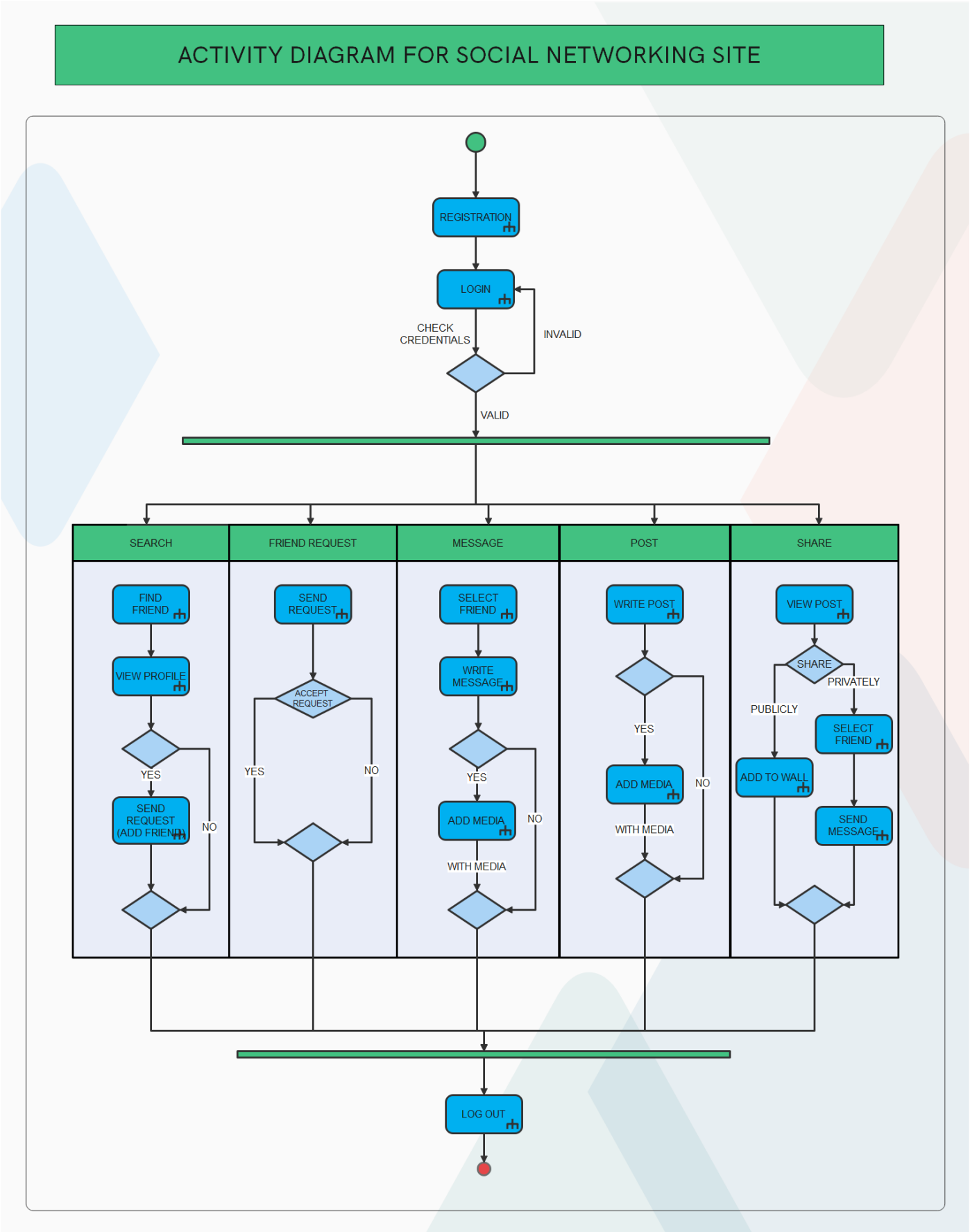
The below diagram depicts all the activities regarding Social Networking Site like Facebook. In the diagram the activities are divided into 5 modules i.e., Friend Request Module, Messenger Module, Friend Search Module, Post Activity Module, Share Activity Module briefly, these modules are described in the diagram.

Firstly, the user needs to login into his account with his ID and opens up his account.

Then he/she can do any activity from any of the modules like searching for a friend, viewing profile, sending friend request; sending message/media, writing a post and share the post, viewing a post, etc. At last user can logout from the session

All these activities are properly divided into 5 modules as stated above and very well described.

c) Diagram


ACTIVITY DIAGRAM FOR SOCIAL NETWORKING SITE
REGISTRATION
LOGIN
CHECK CREDENTIALS
INVALID
VALID
SEARCH
FRIEND REQUEST
MESSAGE
POST
SHARE
FIND FRIEND
VIEW PROFILE
YES
SEND REQUEST (ADD FRIEND)
NO
SEND REQUEST
ACCEPT REQUEST
YES
NO
SELECT FRIEND
WRITE MESSAGE
YES
ADD MEDIA
NO
WITH MEDIA
WRITE POST
YES
ADD MEDIA
NO
WITH MEDIA
VIEW POST
SHARE
PUBLICLY
PRIVATELY
ADD TO WALL
SELECT FRIEND
SEND MESSAGE
LOG OUT

References

[1] GeeksforGeeks, "Software Requirement Specification (SRS) Format - GeeksforGeeks," [Online]. Available: https://www.geeksforgeeks.org/software-requirement-specification-srs-format/.

[2] R. S. Pressman, Software engineering: a practitioner's approach., Palgrave Macmillan, 2005.

[3] V. Paradigm, "What is Unified Modeling Language (UML)?," [Online]. Available: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/.

[4] Creately, "UML Diagram Types Guide: Learn About All Types of UML Diagrams with Examples," [Online]. Available: https://creately.com/blog/diagrams/uml-diagram-types-examples/.

[5] Javatpoint, "Software Engineering Data Flow Diagram - javatpoint," [Online]. Available: https://www.javatpoint.com/software-engineering-data-flow-diagrams.

[6] Javatpoint, "The Dining Philosophers Problem - javatpoint," [Online]. Available: https://www.javatpoint.com/os-dining-philosophers-problem.

[7] uml-diagrams.org, "Bank ATM UML Diagram Examples," [Online]. Available: https://www.uml-diagrams.org/examples/bank-atm-example.html.

[8] W. EdrawMax, "UML Diagrams drawn using Wondershare EdrawMax Software," [Online]. Available: https://www.edrawsoft.com/edraw-max/.