EXPERIMENT-1

AIM: To make a feasibility report on Fingerprint-based ATM system.

OBJECTIVE

The objective of this study is to develop a system which is used for ATM security applications. The need of developing the fingerprint-based ATM system is analyzed along with problems in the existing system, the various hardware and software requirements, their cost analysis, and thus determine the feasibility of the system. The system's limitations are also analyzed and checked whether they overweigh the advantages of the system.

PROBLEMS IN THE EXISTING SYSTEM

- The traditional ATM systems are susceptible to a number of different types of frauds like card skimming, card trapping, etc.
- With the current advanced technology, the ATM cards can be replicated easily.
- The PINs can also be traced and captured.
- But since the fingerprints are biological features and are unique for every human, it is nearly impossible to bypass a fingerprint-based system.

DESCRIPTION

As the name suggests, the fingerprint-based ATM system uses the fingerprint of the user to authenticate and verify the user. Since fingerprints are unique for all the humans (even the twins can't have same fingerprints), the system is very robust and prone to different types of ATM frauds. Along with the PIN, the user will also need to verify the fingerprint in order to proceed with any transaction. There will be two layers of security:

- The smartcard is used for the first layer of mutual authentication when a user requests transaction.
- Biometric authentication is the second layer.

The high-level process is mentioned below:

- The fingerprint image is encrypted via 3D map as soon as it is captured.
- It is then is transmitted to the central server via symmetric algorithm.
- The encryption keys are extracted from the random pixels distribution in a raw image of fingerprint.
- After this, the minutiae matching is performed at the central server.
- The successful minutiae matching at last verifies the claimed user.

COST ANALYSIS

Fingerprint Scanner	:	Rs. 15,000
Database Management	:	Rs. 30,000
Maintenance	:	Rs. 25,000
Windows 10 Operating System	:	Rs. 8,000
.NET Framework	:	Rs. 5,000
Web Server	:	Rs. 20,000
DB Architect	:	Rs. 50,000

Total Cost- Rs. 1,53,000

LIMITATIONS

The various limitations of the system are:

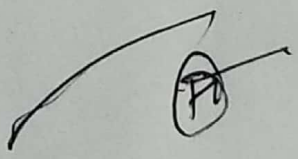
- Misidentification
- False Acceptance due to some fault in the scanner or the system
- False Rejection due to dry, wet or dirty hands
- Design issues in Biometric systems

FUTURE SCOPE

- Future work will focus on the study of stable features (as part of encryption key) of fingerprint image, which may help to set up a fingerprint matching dictionary so that to narrow down the workload of fingerprint matching in a large database.
- The time complexity can be improved.
- The false acceptance and false rejection rates due to a number of factors can be minimized.

CONCLUSION

A biometric-based ATM system is a great addition to the traditional smartcard based ATM system. The possession (smartcard) together with the claimed user's fingerprint increases the security and robustness of the system and is thus essential in a transaction.



Software Requirements Specification

For

CampusPro!

Institute Management System

Version 1.0 approved

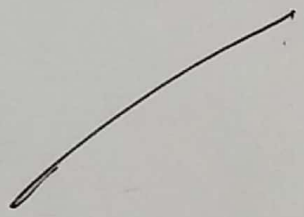
Prepared by:

BIPIN LALA
AASHITA ARORA

Bharati Vidyapeeth's College of Engineering

25st January 2020

Table of Contents



1. Introduction

1.1 Purpose

The main purpose of our software is to ease out the various processes involved in running and managing a institute. This can include tasks like managing the record of students, teachers and staff, their financial records, generating report cards of students as well managing library and issuing of books. This document serves as the unambiguous guide for the developers of this software system.

1.2 Document Conventions

The document only covers the requirement specification for the CampusPro! Institute Management System. This document does not provide any references to the other component of the CampusPro! Institute Management System. All the external interfaces and the dependencies are also identified in this document.

1.3 Intended Audience and Reading Suggestions

Our project helps in easing the managing of various processes involved in running an Institute.

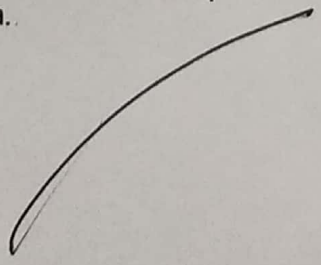
1.4 Product Scope

The overall scope of the feasibility study was to provide sufficient information to allow a decision to be made as to whether the Institute Management System project should proceed and so, its relative priority in the context of the other existing Institute Management System.

2. Overall Description

2.1 Product Perspective

Institute Management System is a software developed to ease out the managing and the running of institute. By keeping all the records in an easy to access database, we increase the speed and efficiency of performing tasks. Also as all the data will be stored in a centralized database, it will be much easier to update the records while maintaining consistency and security of data.



2.2 Product Functions

The main purpose of our software is to ease out the various processes involved in running and managing a institute. This can include tasks like managing the record of students, teachers and staff, their financial records, generating report cards of students as well managing library and issuing of books. As all the student records will be stored centrally, the student marks and can be updated and a report card can be generated easily. All the books issued by the student can also be saved accordingly. Even financial data like fees paid can be managed easily.

2.3 User Classes and Characteristics

This software is developed such that total appearance of the product to make it more user friendly. The operator will be provided with login id and password. General users with basic computer skills can use this software.

2.4 Operating Environment

Any popular operating system (Windows, linux, macOS) can be used for running the software. SQL: – Structured Query Language for handling the Database Related Query will be used.

2.5 Design and Implementation Constraints

The major constraint will be the size of database which will be ever increasing. A large number of records are to be stored in the centralized relational database. So speed of accessing data also needs to be maintained.

3. External Interface Requirements

3.1 User Interfaces

This software is designed in such way that total appearance of the product to make it more user friendly. The operator will be provided with loginid and password. General users with basic computer skills can use this software.

3.2 Hardware Interfaces

The capability of the computer depends on the performance of the software. The software can take any number of input provided the database size is large enough. This would depend on the available memory space.

3.3 Software Interfaces

The user and administrators interact with the system through a web-portal, A new user should be able to register on the web-portal in order to log in and manage the Account information. An administrator should also be able to log in to the web-portal where he/she can administer the system by for instance editing details.

3.4 Communications Interfaces

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems for both the mobile application and the web portal.

4. System Features

This project is to maintain the User details, student details, teachers and staff details. Various tasks involving such information can be performed easily within seconds.

- **Administration module**
- **Teacher module**
- **Student related app**

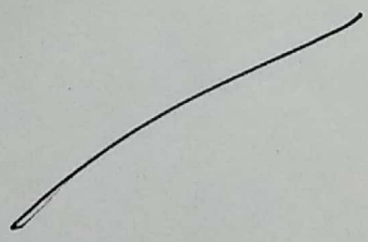
5. Other Nonfunctional Requirements

5.1 Performance Requirements

The requirements in this section provide a detailed specification of the user interaction with the software and measurements placed on the system performance to help the developers understand the intent and make suitable design choices. For large dataset, it can handle operation in more optimized way.

5.2 Safety Requirements

Safety requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. So proper safeguards or actions are taken to prevent misuse of product.



5.3 Security Requirements

Since this software will be hosted on cloud server, all the user data will be kept on the cloud server. Product should be able to protect privacy of user data. Workspace of the user should only be accessed through user own credentials and any other user should not be able to access to the user private data.
Applications designed with security in mind are safer than those where security is an afterthought. Traditionally security issues are first considered during the Design phase of the Software Development.

5.4 Portability Requirements

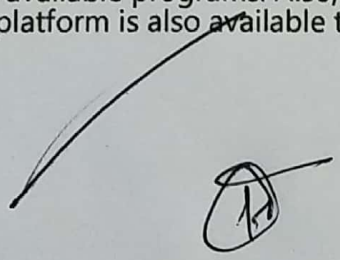
As explained in the performance requirements section, software should be lightweight so that it can run on a machine with slow internet connection. To make the web application lightweight, simple libraries and tools should be used at developing phase. Such as using JavaScript and HTML5 instead of Apache Flex. Portability also means running on most number of different platform without an additional effort. To achieve this, web application should be developed by using the common technologies and tools

5.5 Software Quality Attributes

Additional quality characteristics for the product are introduced will be important to either the customers or the developers. These are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability.

6. Other Requirements

The project works on various available programs. Also, it can be used to link new open source GitHub projects. This platform is also available to work on SAAS and other different web-portals.



Page No.
Date

EXPERIMENT-3

AIM - To draw class diagram of order processing system

THEORY - The UML class diagram is a graphical notation used to construct and visualize object oriented system. A class diagram in UML is a type of static diagram that describes structure of a system by showing :

- Classes
- attributes
- operations
- and relationship among objects

CLASS NOTATION -

Class Name - The name of the class that appears in first partition

Class Attributes - Shown in second partition, they map onto other member variables

Class operations - Shown in 3rd partition, they

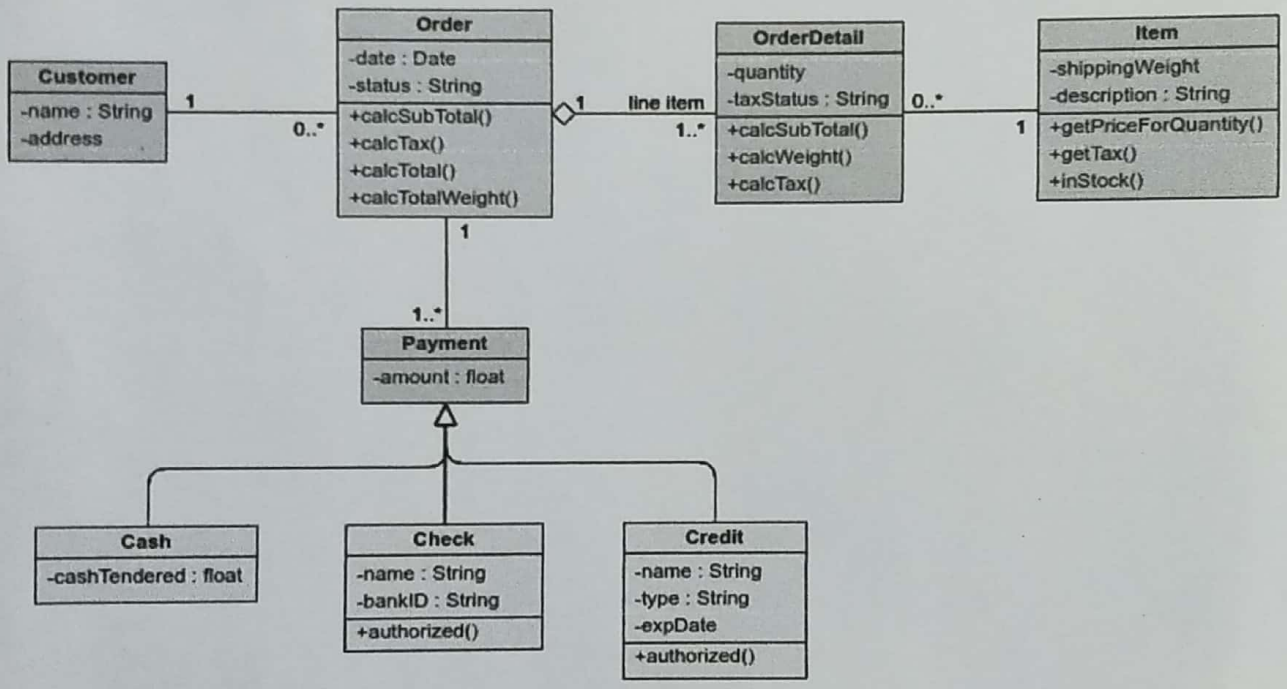
Page No.
Date

are services the class provides. The return type of a method is shown after colon at end of method signature.

RESULT- The class diagram was drawn & analysed

CLASS DIAGRAM

AIM – To draw class diagram of ORDER PROCESSING SYSTEM



Page No.
Date

EXPERIMENT-4

AIM - To draw a use case diagram for canteen Automation system

THEORY - A UML use case diagram is the primary form of system requirements for a new program underdeveloped. In UML a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system.

use case diagrams are twofold - They are both behaviour diagrams because they describe behaviour of system and are destructive diagrams as a special case of class diagram where classes are restricted to be either actors or use cases related to each other.

PURPOSE OF USE CASE DIAGRAM

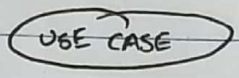
- Specify context of system
- Capture requirements of system
- Validate systems architecture
- Drive implementation & generate test cases.

Page No. _____
Date

NOTATION

Actor- Actor is someone which interacts with the use case. Actor provides inputs to the system and gets outputs from system

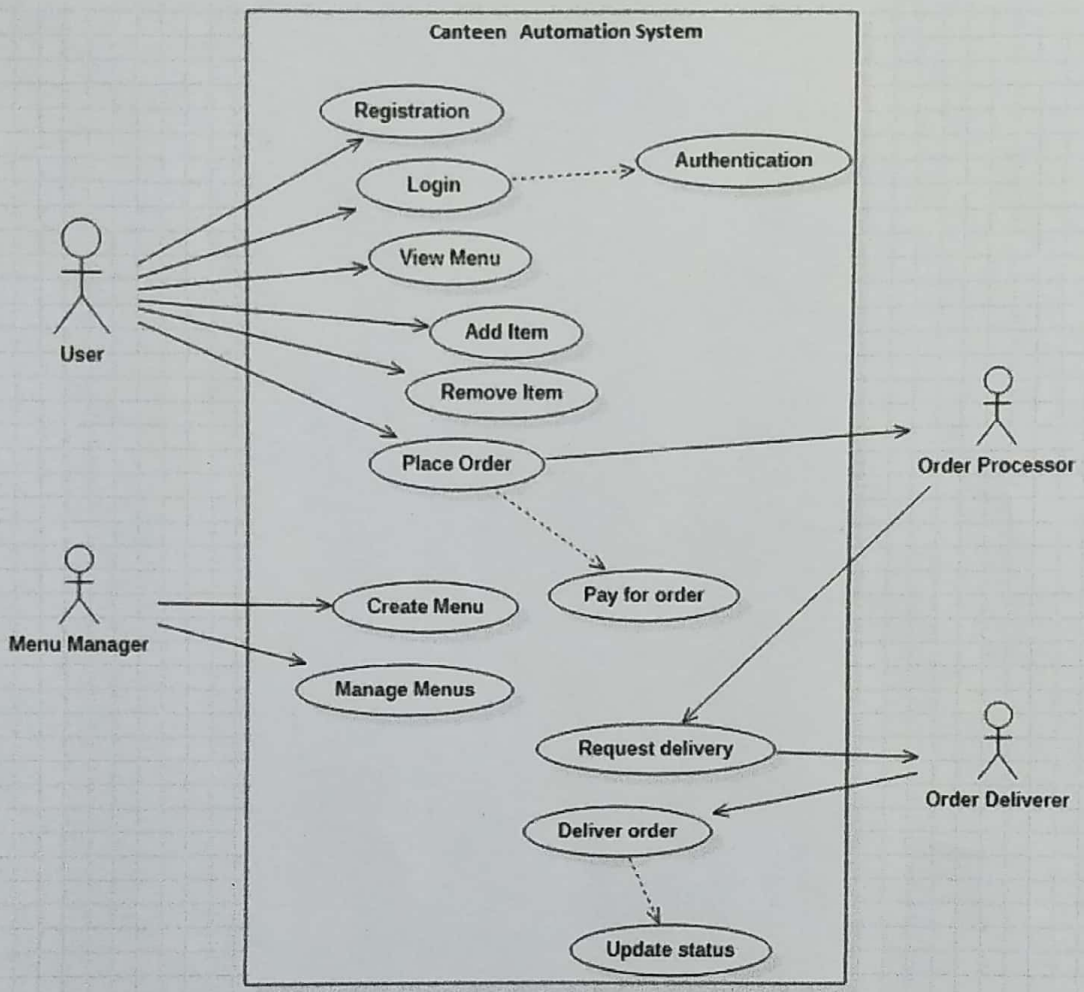
Use case - It is a system function which can be manual or automated



CONCLUSION - The use case diagram was implemented.

USECASE DIAGRAM

Aim: Create a Use Case Diagram for Canteen Automation System.



Page No.
Date

EXPERIMENT-5

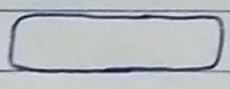
AIM - To make an activity diagram for order processing

THEORY - The unified Modelling Language (UML) includes several subsets of diagrams including structure diagrams, interaction diagrams and behaviour diagrams - Activity diagrams, along with use case and state machine diagrams are considered behavior diagrams because they describe what must happen in the system being modeled.

BENEFITS - Activity diagrams present a large no. of benefits to the users.

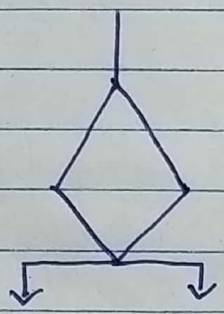
- Demonstrate logic of an algorithm
- Simplify and improve any process of clarifying complicated use cases
- Model software architecture elements such as method function and operation.

BASIC COMPONENTS -

ACTION -  Symbolised with round edged rectangles.

wherein the users or software perform a given task

DECISION NODE - A conditional branch in the flow that is represented by a diagram.



Single input but two or more outputs

CONTROL FLOW - Connections that show the flow

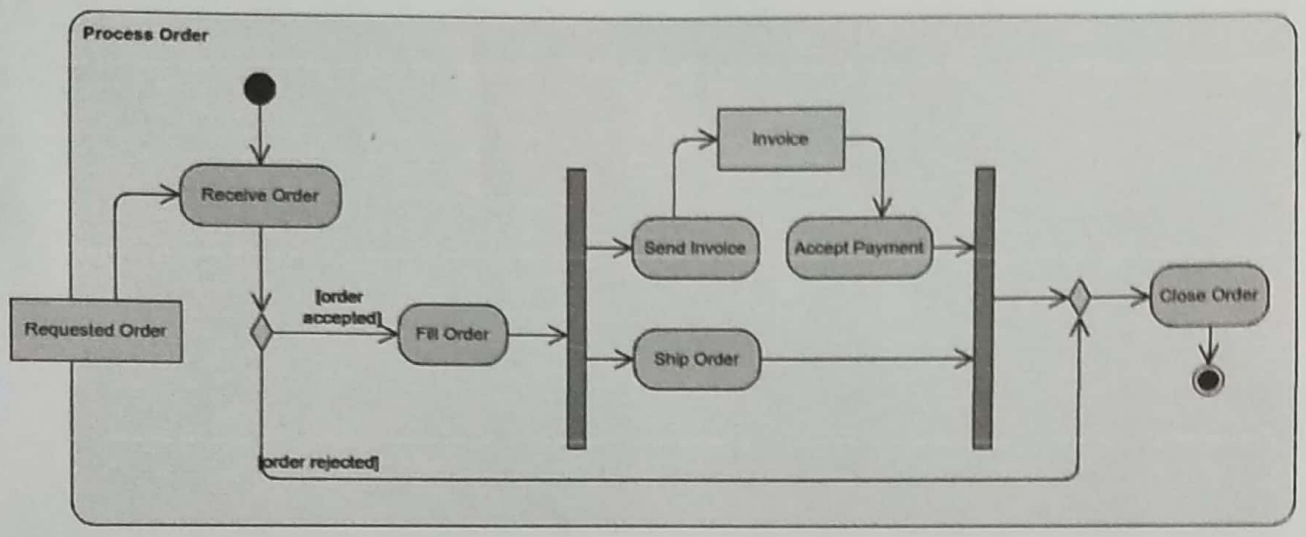
START NODE - Symbolizes beginning of an activity

represented by black circle

END NODE - Represents the final step in the activity represented by outlined black circle.

ACTIVITY DIAGRAM

AIM – To draw activity diagram of ORDER PROCESSING SYSTEM



Page No.
Date

EXPERIMENT-6

AIM- To make a state chart diagram

THEORY- A state chart diagram describes a state machine. It describes different states of a component in a system. These states are specific to a component/object of the system.

State chart diagram is used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by system. These diagrams describe the flow of control from one state to another.

PURPOSE OF STATE CHART DIAGRAM

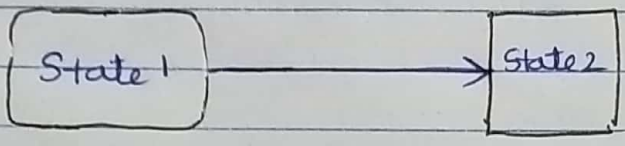
- To model dynamic aspect of a system
- To model the life time of a reactive system
- To describe different states of an object during its lifetime

BASIC COMPONENTS

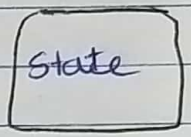
1.) Initial State- Represents the initial state of the system

Page No. _____
Date

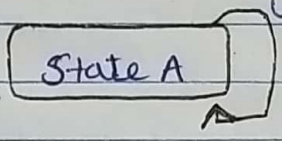
2.) Transitions - Arrow is labelled with event that causes the change in state.



3.) State - Represents the circumstances of an object at an instant of time



4.) Self Transition -

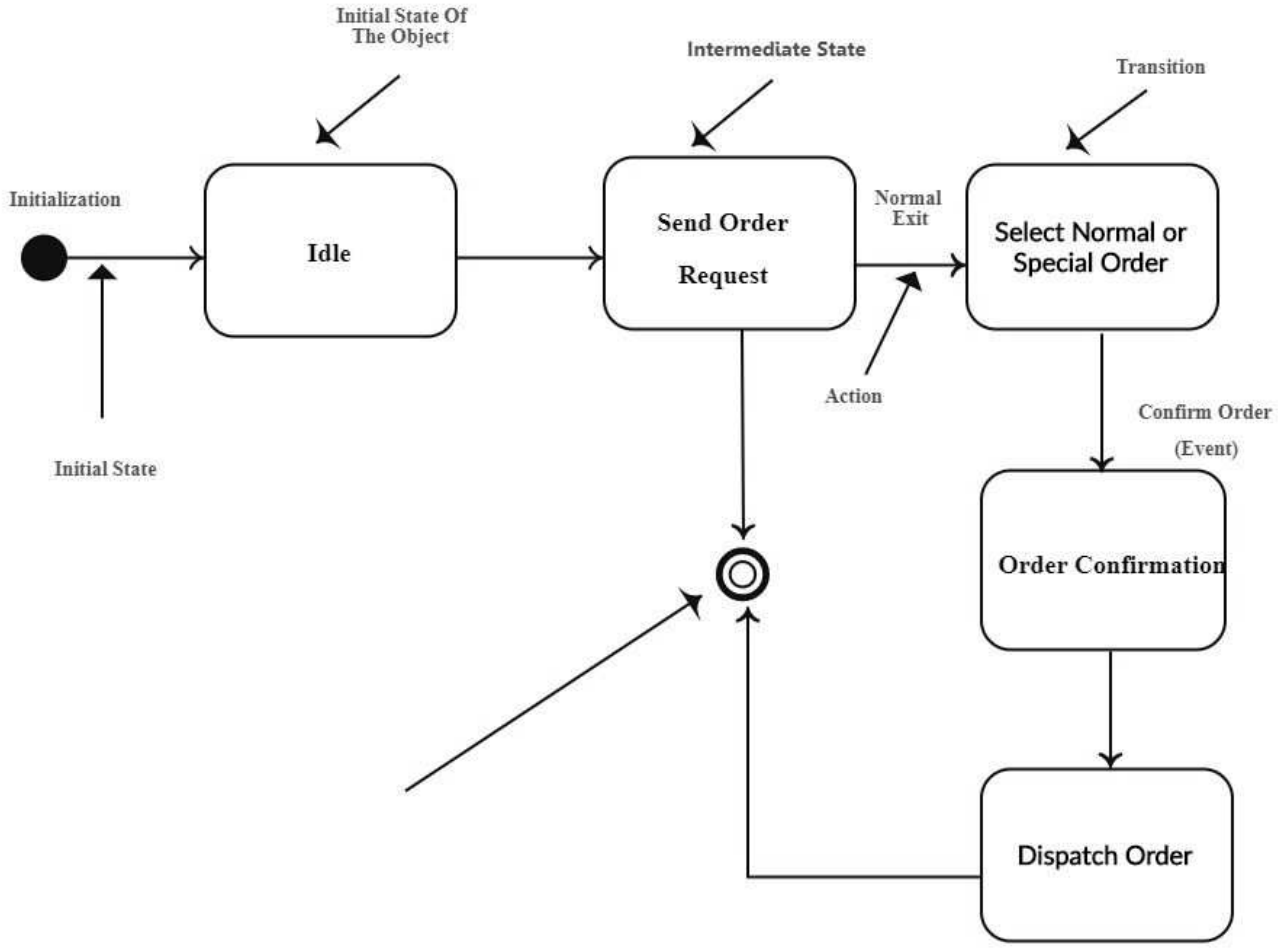


5.) Final State - Represents the final state of the system.

CONCLUSION - State chart diagram has been implemented

EXPERIMENT - 6

AIM - To draw a State Chart Diagram for Order Processing System



State Chart Diagram for Order Processing System

Page No.
Date

EXPERIMENT - 7

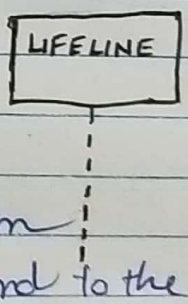
AIM - To make a Sequence Diagram

THEORY - Sequence diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence diagrams describe how and in what order the objects in a system function.

BASIC NOTATIONS -

- **Actors** - Represents the typed role where it interacts with the system and its objects.

- **Lifeline** - A lifeline is a named element which depicts an individual participant in a sequence diagram. A lifeline portrays objects internal to the system while actors are used to depict objects external to the system.



Page No.
Date

Messages - Messages appear in a sequential order in a lifeline and represented using arrows

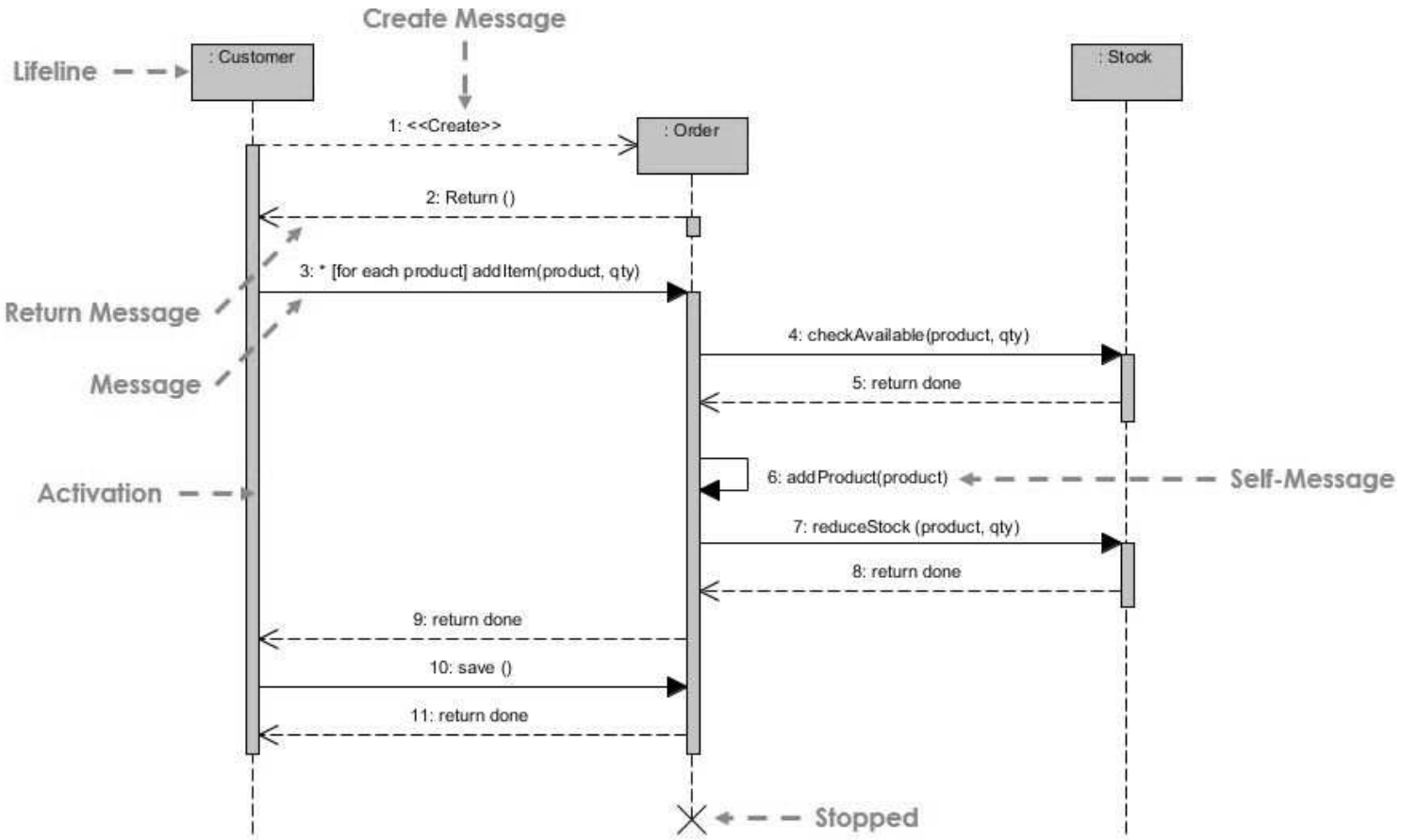
Synchronous messages - waits for a reply before the interaction can move forward.

Asynchronous messages - does not wait for a reply.

RESULT - Sequence diagram has been implemented.

EXPERIMENT - 7

AIM - To draw a Sequence Diagram for Order Processing System



Sequence Diagram for Order Processing System

Page No.
Date

EXPERIMENT-8

AIM - To make a collaboration diagram

THEORY - Collaboration diagram is a type of interaction diagram. In this diagram the method call sequence is indicated by some numbering technique.
The number indicates how the methods are called one after the another.

Method calls are similar to that of a sequence diagram. However, difference being the sequence diagram does not describe the object organization, whereas the collaboration diagram show the object organization.

If time sequence is important, the sequence diagram is used.
If organization is required, then collaboration diagram is used.

Collaboration diagrams are used to describe the structural organization of the objects taking part in the interaction.

NOTATIONS

Object - An object in the collaboration is named and has

its class specified.

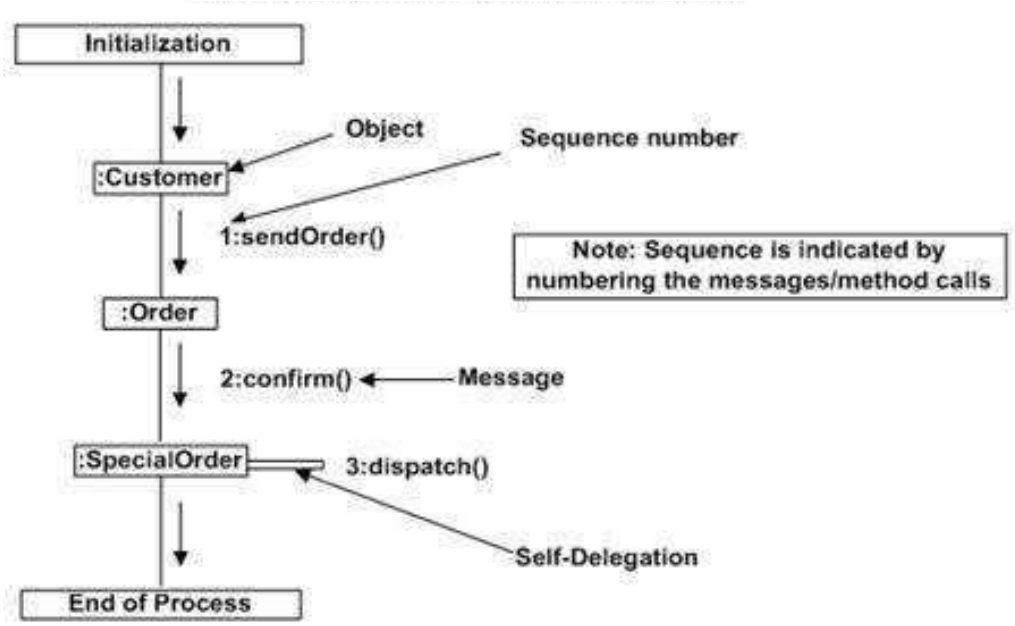
Actors - An actor in the collaboration diagram as the invoker of the interaction.

LINKS - In collaboration diagram a link is a relationship among objects across which messages can be sent.

CONCLUSION - The collaboration diagram was implemented.

EXPERIMENT - 8

AIM - To draw a Collaboration Diagram for Order Processing System



Collaboration Diagram for Order Processing System