**SOFTWARE DESIGN DOCUMENT**

**FOR**

**ANKA BUSINESS SUPPORT SERVICES**

**GITHUB LINK:**

[**https://github.com/Aurits/ANKA\_BUSINESS\_SUPPORT.git**](https://github.com/Aurits/ANKA_BUSINESS_SUPPORT.git)

**G-30**

**SUPERVISOR: MARY NSABAGWA**

|  |  |  |
| --- | --- | --- |
| **STUDENT NAME** | **REGISTRATION NUMBER** | **STUDENT NUMBER** |
| **MUGARURA KEVIN** | **21/U/09399/PS** | **2100709399** |
| **MUSIIMENTA CYNTHIA** | **21/U/05922/PS** | **2100705922** |
| **KIRABO JELLY ROLLINGS** | **21/U/08437/PS** | **2100708437** |
| **ALANDA AMBROSE** | **21/U/1392** | **2100701392** |
| **KATAMBA HARUN ARNOLD** | **21/U/07732/PS** | **2100707732** |

**TABLE OF CONTENTS**

[List of Tables iii](#_Toc113723866)

[List of Figures iv](#_Toc113723867)

[1. INTRODUCTION 1](#_Toc113723868)

[1.1 Purpose 1](#_Toc113723869)

[1.2 Scope 1](#_Toc113723870)

[1.3 Overview 1](#_Toc113723871)

[1.4 Definitions and Acronyms 2](#_Toc113723872)

[2. SYSTEM OVERVIEW 3](#_Toc113723873)

[3. SYSTEM ARCHITECTURE 4](#_Toc113723874)

[3.1 Architectural Design 4](#_Toc113723875)

[3.2 Decomposition Description 7](#_Toc113723876)

[3.3 Design Rationale 13](#_Toc113723877)

[4. DATA DESIGN 14](#_Toc113723878)

[4.1 Data Description 14](#_Toc113723879)

[4.2 Data Dictionary 18](#_Toc113723880)

[5. COMPONENT DESIGN 21](#_Toc113723881)

[6. HUMAN INTERFACE DESIGN 25](#_Toc113723882)

[6.1 Overview of User Interface 25](#_Toc113723883)

[6.2 Screen Images 27](#_Toc113723884)

# List of Tables

[Table 4.1. 1Database entities with respective attributes 14](#_Toc113723900)

[Table 4.1. 2Customer 16](#_Toc113723901)

[Table 4.1. 3Participant 16](#_Toc113723902)

[Table 4.1. 4ProductTable 16](#_Toc113723903)

[Table 4.1. 5PerformanceReport 16](#_Toc113723904)

[Table 4.1. 6BusinessAdministrator 17](#_Toc113723905)

[Table 4.1. 7Booking 17](#_Toc113723906)

[Table 4.1. 8ProductCatergory 17](#_Toc113723907)

[Table 4.2. 1Customer 18](#_Toc113723908)

[Table 4.2. 2Participant 18](#_Toc113723909)

[Table 4.2. 3Product 19](#_Toc113723910)

[Table 4.2. 4PerformanceReport 19](#_Toc113723911)

[Table 4.2. 5Administrator 19](#_Toc113723912)

[Table 4.2. 6Booking 20](#_Toc113723913)

[Table 4.2. 7ProductCatergory 20](#_Toc113723914)

[Table 5. 1Capture User Details 21](#_Toc113723915)

[Table 5. 2Data Request 22](#_Toc113723916)

[Table 5. 3Awarding 23](#_Toc113723917)

[Table 5. 4Booking 24](#_Toc113723918)

[Table 5. 5Report generation 25](#_Toc113723919)

# List of Figures

[figure 3.2. 1The use case diagram for user scenarios of the different users 7](#_Toc113723924)

[figure 3.2. 2Class diagram for the system 8](#_Toc113723925)

[figure 3.2. 3Object diagram for the system 8](#_Toc113723926)

[figure 3.2. 4Activity diagram for user login 9](#_Toc113723927)

[figure 3.2. 5Activity diagram for the administrator 10](#_Toc113723928)

[figure 3.2. 6Activity diagram for the customer 11](#_Toc113723929)

[figure 3.2. 7Activity diagram for the participant 12](#_Toc113723930)

[figure 4.1. 1Eerd for the entities of the system in database 15](#_Toc113723931)

[figure 6.2. 1Homepage of ANKA business support system website 27](file:///C:\Users\Victor%20Gundagh\Desktop\Recess%20documentation\Design%20document.docx#_Toc113723932)

[figure 6.2. 2Product section on the website 28](file:///C:\Users\Victor%20Gundagh\Desktop\Recess%20documentation\Design%20document.docx#_Toc113723933)

[figure 6.2. 3Command line Interface for participants to post products 28](file:///C:\Users\Victor%20Gundagh\Desktop\Recess%20documentation\Design%20document.docx#_Toc113723934)

[figure 6.2. 4Ordering section for customers to request products 28](file:///C:\Users\Victor%20Gundagh\Desktop\Recess%20documentation\Design%20document.docx#_Toc113723935)

[figure 6.2. 5login Screen for users to access the system via the web interface 28](file:///C:\Users\Victor%20Gundagh\Desktop\Recess%20documentation\Design%20document.docx#_Toc113723936)

[figure 6.2. 6Administrator Dashboard to view reports and award participants 28](file:///C:\Users\Victor%20Gundagh\Desktop\Recess%20documentation\Design%20document.docx#_Toc113723937)

# INTRODUCTION

## Purpose

This software design document describes the architecture and system design of Anka business support services. This SDD consists of different sections that explain in detail what the system does. It is intended to help the reader to understand the inner design and functioning of the system.

## Scope

ANKA Business Support Solutions is a system that supports participants to post their products via the command line interface built with JAVA and the products can be accessed via the web interface which is built by PHP by the customers. The posted products with their description are stored in the MYSQL database. The participants are awarded points, recognition and promotion depending on the performance of the products. The administrator logs into the system, and views reports generated from the competition.

## Overview

The main objective of the design document is to analyze and understand system in detail. In this way, the features comprise parts of feasible solution are identified and then documented.

During design phase, the focus shifts from “whether to how” i.e. we try to answer the question of how to build the system.

Activities used in the design document:

**Introduction (Section 1)**

This section describes the purpose of the SDD, the scope of the product which describes the description, goal and objectives of the product, the reference materials and the abbreviations used in this document.

**System overview (Section 2)**

This section gives a general description of the functionality, context and design of the ANKA business support services.

**System architecture (Section 3)**

This comprises of the architectural design which consists of the system diagram that shows major subsystems and data repositories and their interconnections, decomposition of the system and the design rationale which gives a reason for choosing the used design.

**Data design (section 4)**

This comprises of the data description that explains how the information domain of the system is transformed into data structures and the data dictionary that alphabetically lists the system entities.

**Component design (section 5)**

This describes what each component does in a more systematic way.

**Human interface design (section 6)**

This explains how the users will be able to use the system and also displays screenshots showing the interface from the user’s perspective.

## Definitions and Acronyms

PHP – Hypertext Preprocessor

MYSQL – My structured Query Language

SDD – Software Design Document

CLI – Command Line Interface

OO –Object Oriented

DBMS –Database Management System

OS – Operating System

CSS – Cascading Style Sheets

# SYSTEM OVERVIEW

ANKA Business Support Services is an application software which helps business owners to easily register merchants and helps customers purchase products of the merchants. The business software can be thought about like a utility. It reduces the time spent if the lob was done by humans. It has three kinds of users: participants, customers and administrators. It handles data ranging from text to long integers. It is secured by authorization techniques that reduce illegal usage. It performs basic business transactions which ensure that there is seamless collaboration between the software users.

The software uses a framework Laravel which is a PHP based. This framework in cooperates JavaScript, PHP, HTML and CSS. The framework handles the frontend of the software. The Laravel is configured to the MYSQL DBMS. It is a relational database management system that has databases that are used to store the software’s information.

The backend of the system is built with java which is an OO language. Java is used to build the command line interface which is majorly used by the participants. The java is majorly used to collect information which is then stored in the file by a cron job.

A cron job is Unix-linux kind of utility software inbuilt in most OS. The cron job is configured to run periodically i.e. after five minutes. The cron job is written in java and invoked after the set time. It is majorly used to save information from the text file to the database.

The system uses a server to accept client requests and save files. The software is configured to use local resources and no external networks are used.

# SYSTEM ARCHITECTURE

## Architectural Design

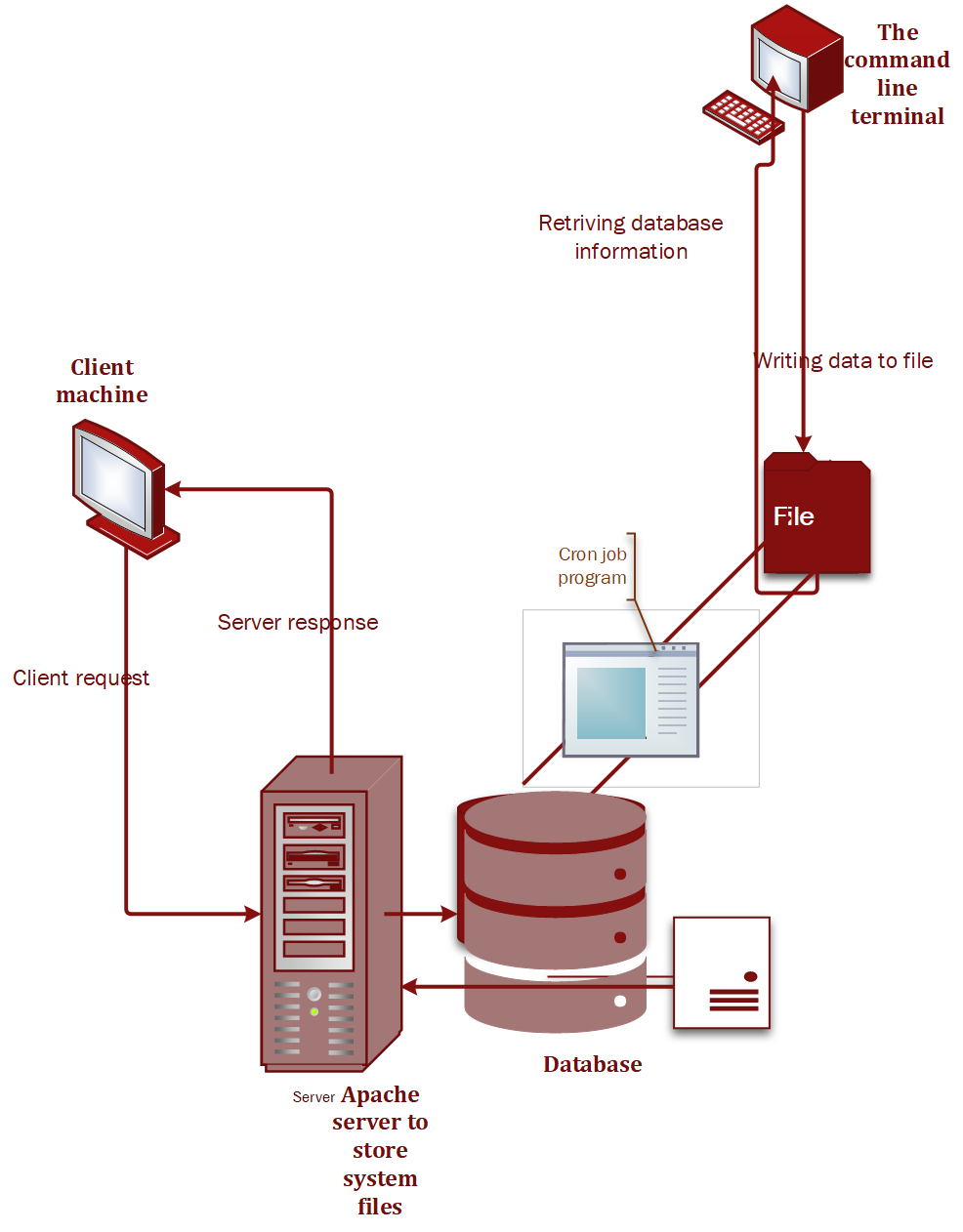


figure 3.1. 1The architectural design of the entire system

The system is made up of the Client-side computer, the Apache server, the Database, a Cron job program and another Administrator Computer which is operated via the command line. The command line is use to register products that are to be ordered for by the customers. The product name and description are first stored in the file while this data is periodically picked by the cron job program and stored in the database.

On the other end, the client can log in into the system as a customer and does sends request to the server that connects to the database to query information about the products and retrieved to the customer for booking.

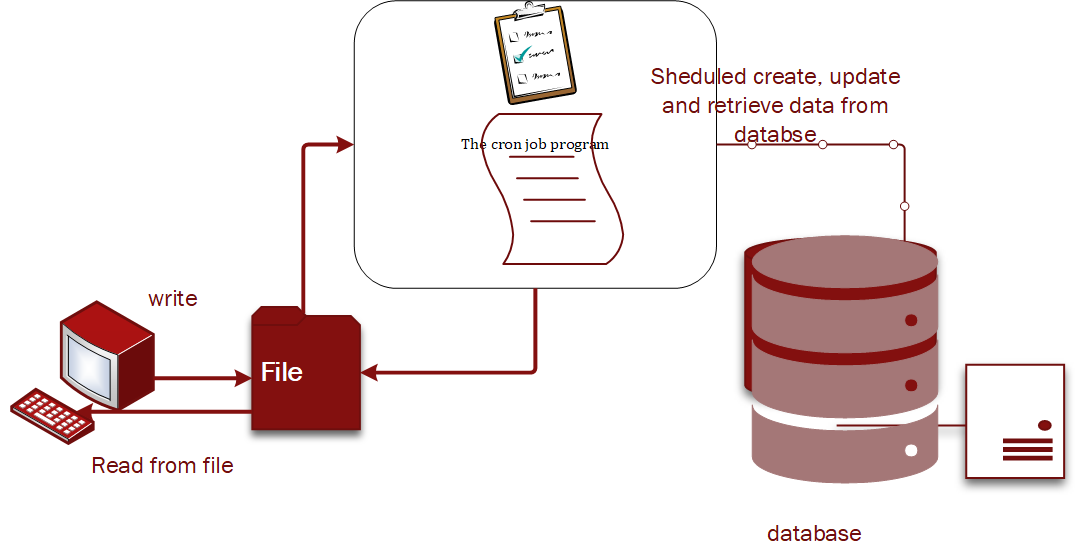


figure 3.1. 2The process of the cronjob that does fetch data to and from the file

There is also a cron job program that runs periodically in the system. When the participant does register a product through the terminal, the data is first stored into the fired where this program does pick it to store it into the database at scheduled intervals.

Similarly, when a participant wants to get information from the database about the products availability and any other related information like one’s performance, the request is first recorded on the file as the cron job program picks it and returns the required information to the participant at given interval. The time of request, recording and response is also stored in the database.

The system has the following components.

**Server**

This is where all the files for the system are stored and it does respond to client requests. It enables the client to search for the products from the database through the php script files that it stores. It therefore enables the client to access the information they need.

**Database**

The database is responsible to store the information about the products that are registered by the participants. It also stores the necessary information about the clients and the participants thus enables the administrators to track the participants present and their customers thus determine the best performing client.

**Command line**

This component is used to register the participants on the system and also use to register their products so that they can be accessed by the customers on the web interface for booking. It is used to enter data to the database through the file.

**Client**

This is the part of the system that the users use to access the system. They use the web interface. They send requests to the server that are responded by the server that at times fetches some information from the database and then returned to the user. It is used by the customers to book products, administrators to query the reports about the performance and also award the best performing participant that is shown on the homepage of the system website.

**File**

This is where data from the command line is first stored before being sent to the database by the cron job program, and also its where responses from the database to participant requests are first recorded before being viewed by the participant.

## Decomposition Description

For the ANKA business support services system , we follow an object-oriented approach to develop the system. The various objects work interactively with one another to enable the participants post the products on the system making the available for the customers to book.

**Use Case Diagram**

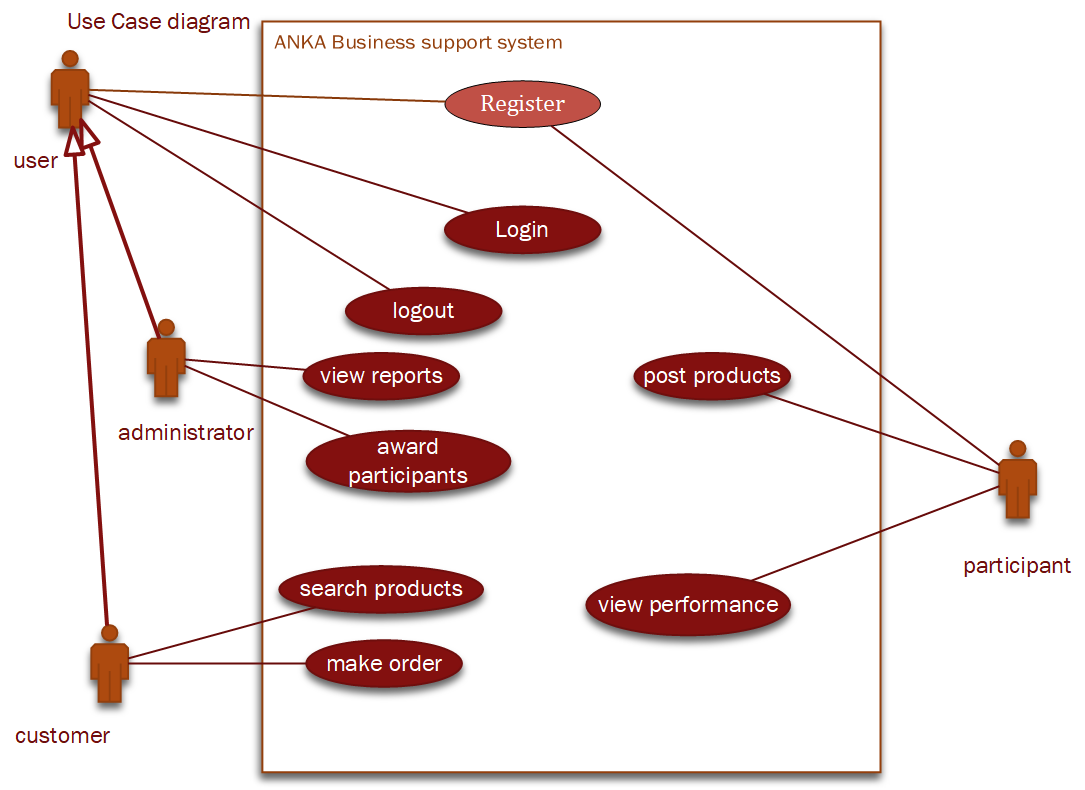


figure 3.2. 1The use case diagram for user scenarios of the different users

Figure 3.2.1 shows the detailed view of the system and how the different actors would interact with each other and with the system so that every user gets the required output from the system after they have satisfied the given conditions.

**Class Diagram**

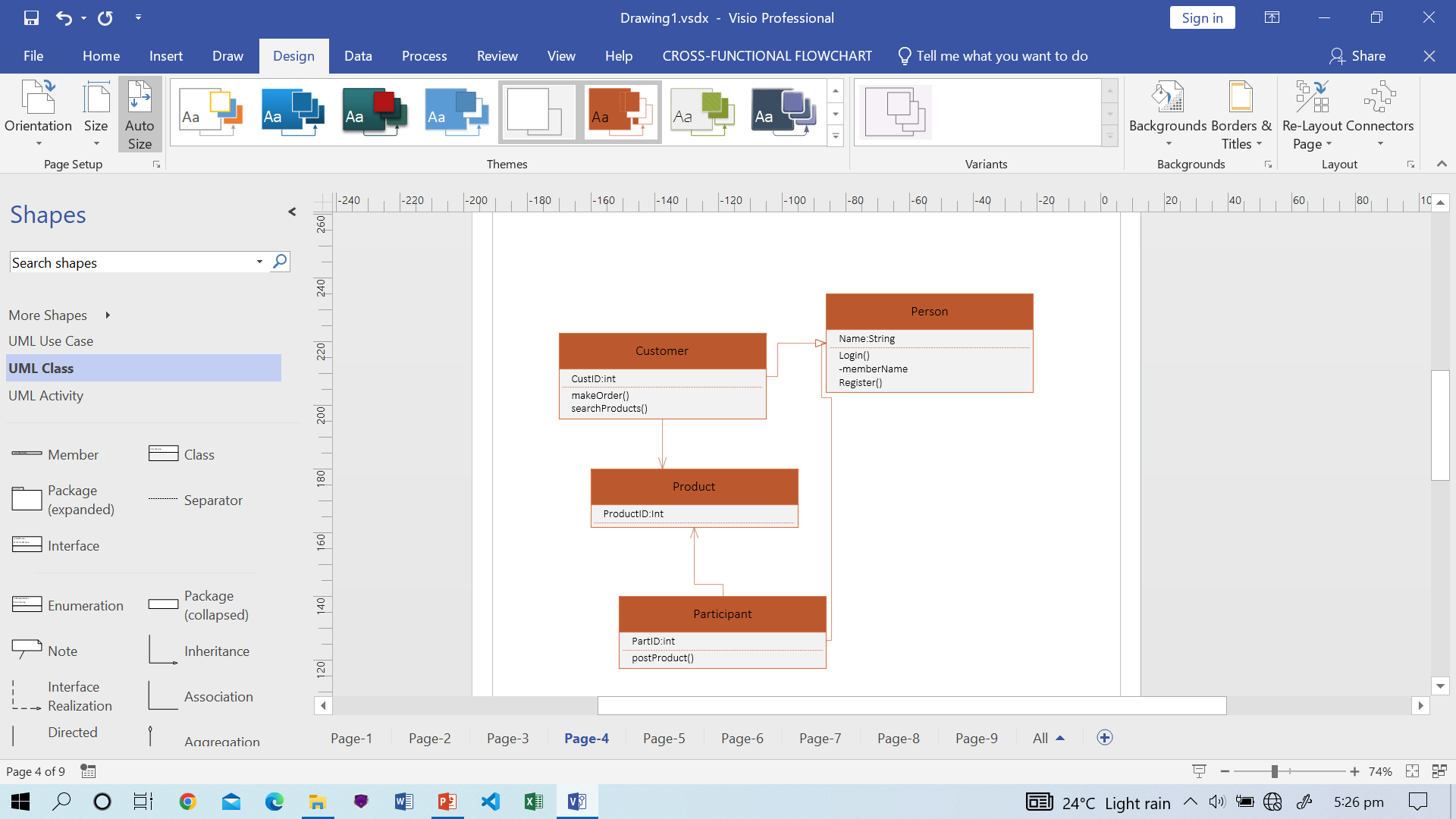


figure 3.2. 2Class diagram for the system

**Object Diagram**

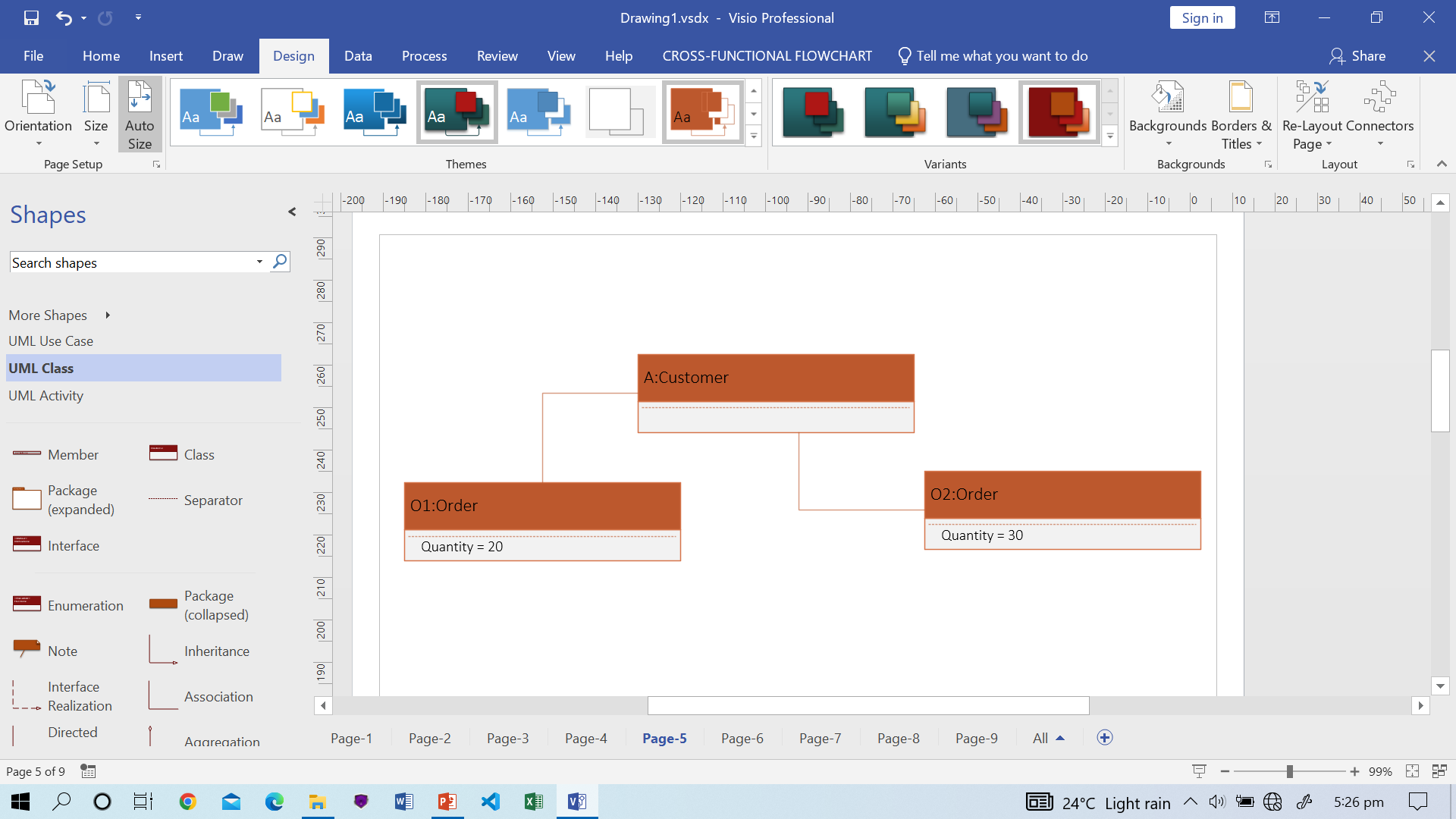


figure 3.2. 3Object diagram for the system

**Activity diagrams**

The activity diagrams show a detailed flow of the activities that are performed by the different users on the system so that each of them achieves a given goal from the system. There are different activity diagrams illustrating the flow of activities for the different users of the system.

**User login to the system activity diagram**

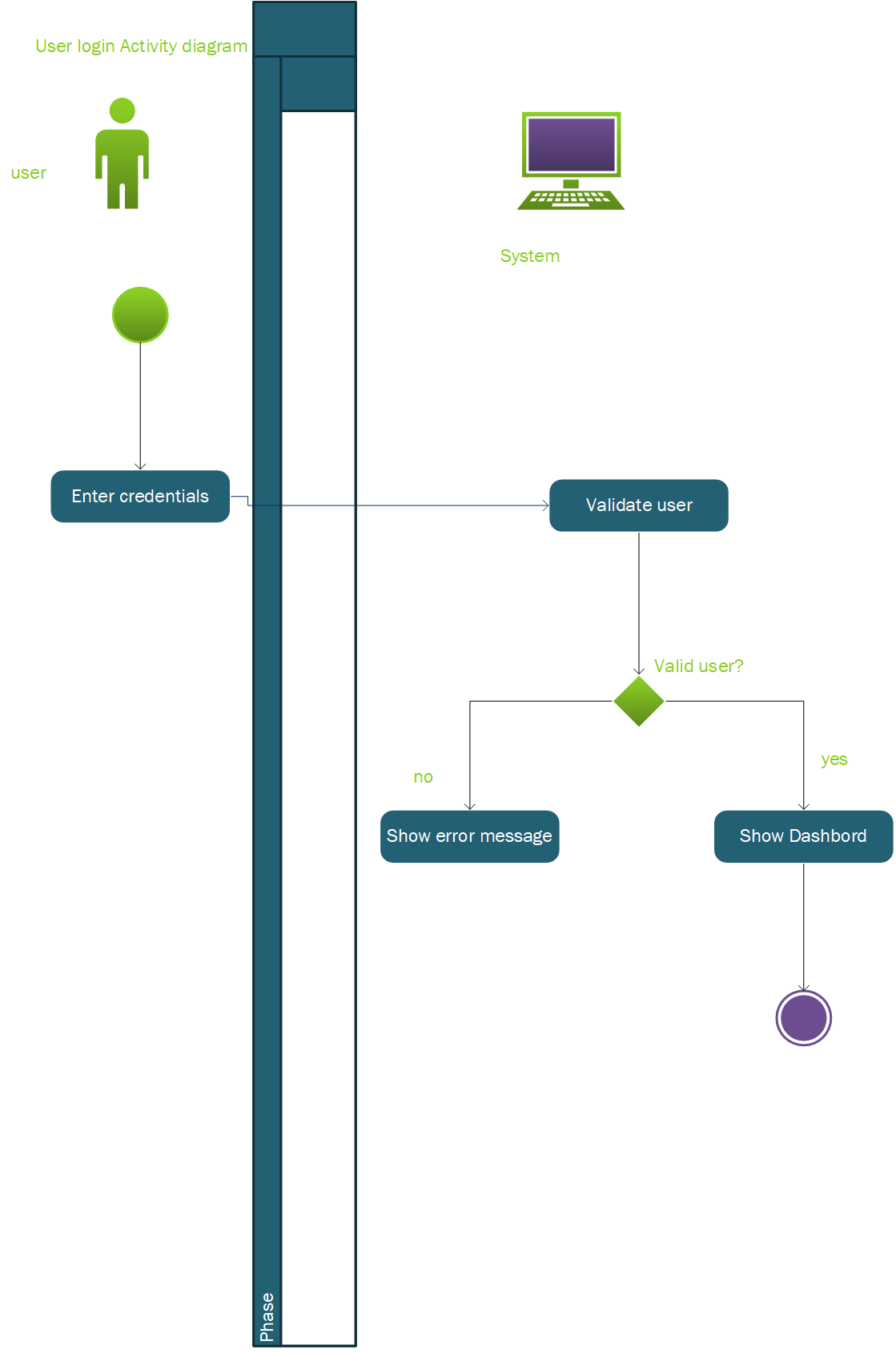


figure 3.2. 4Activity diagram for user login

**Administrator querying the database for best performing participant activity diagram**

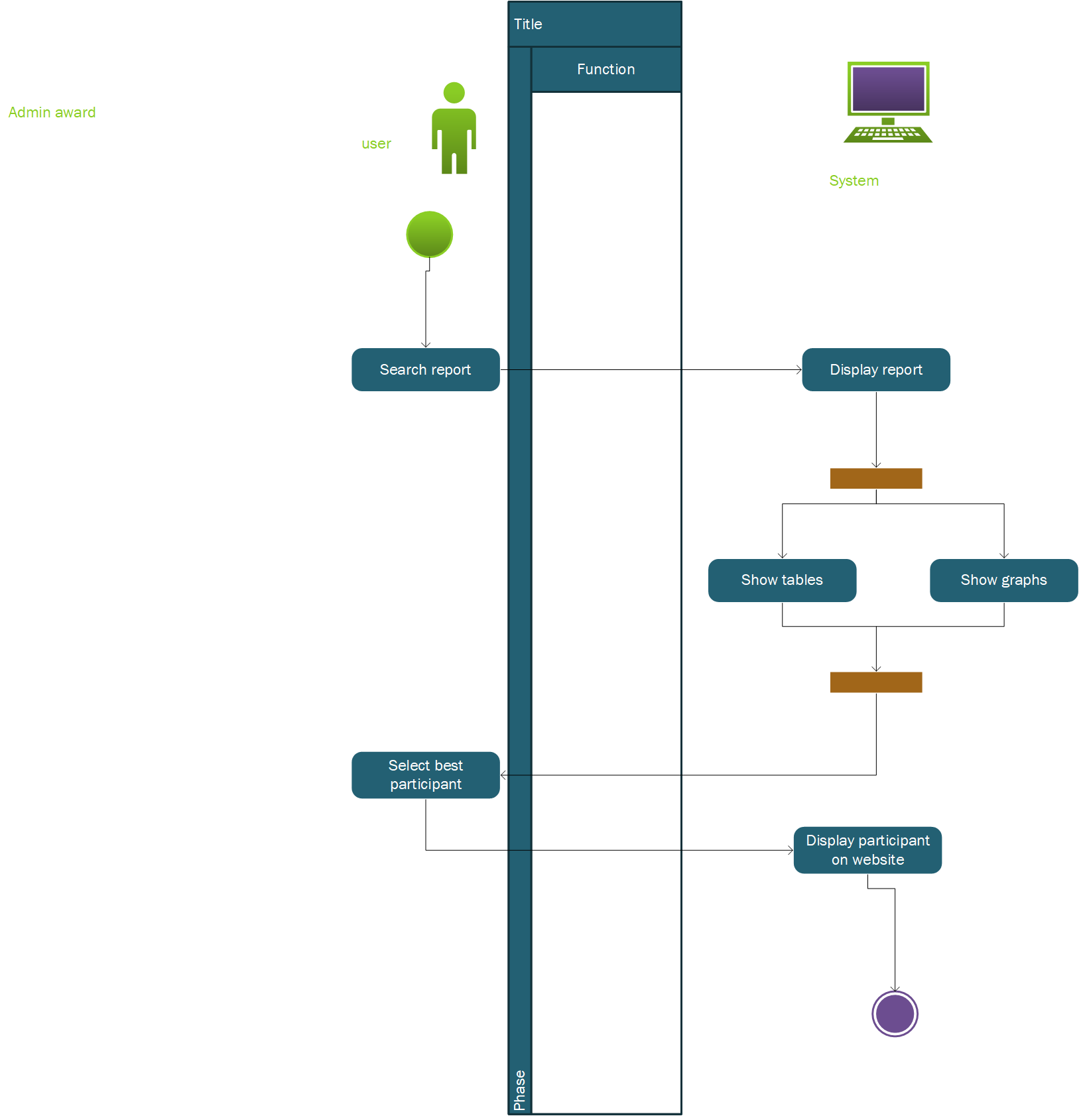


figure 3.2. 5Activity diagram for the administrator

**Customer making order for products on the system activity diagram**

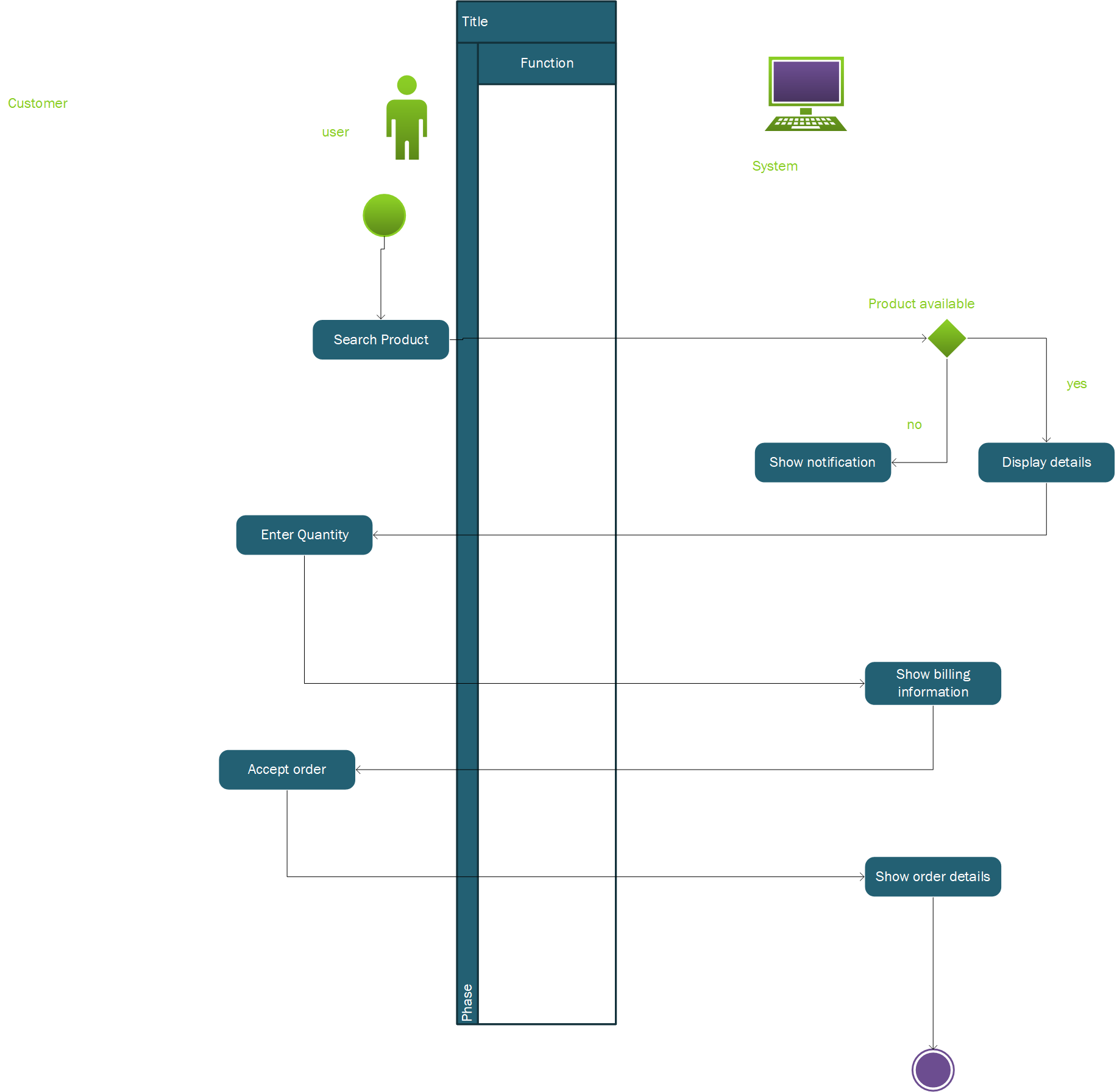


figure 3.2. 6Activity diagram for the customer

**Participant posting products to the website and viewing one’s performance activity diagram.**

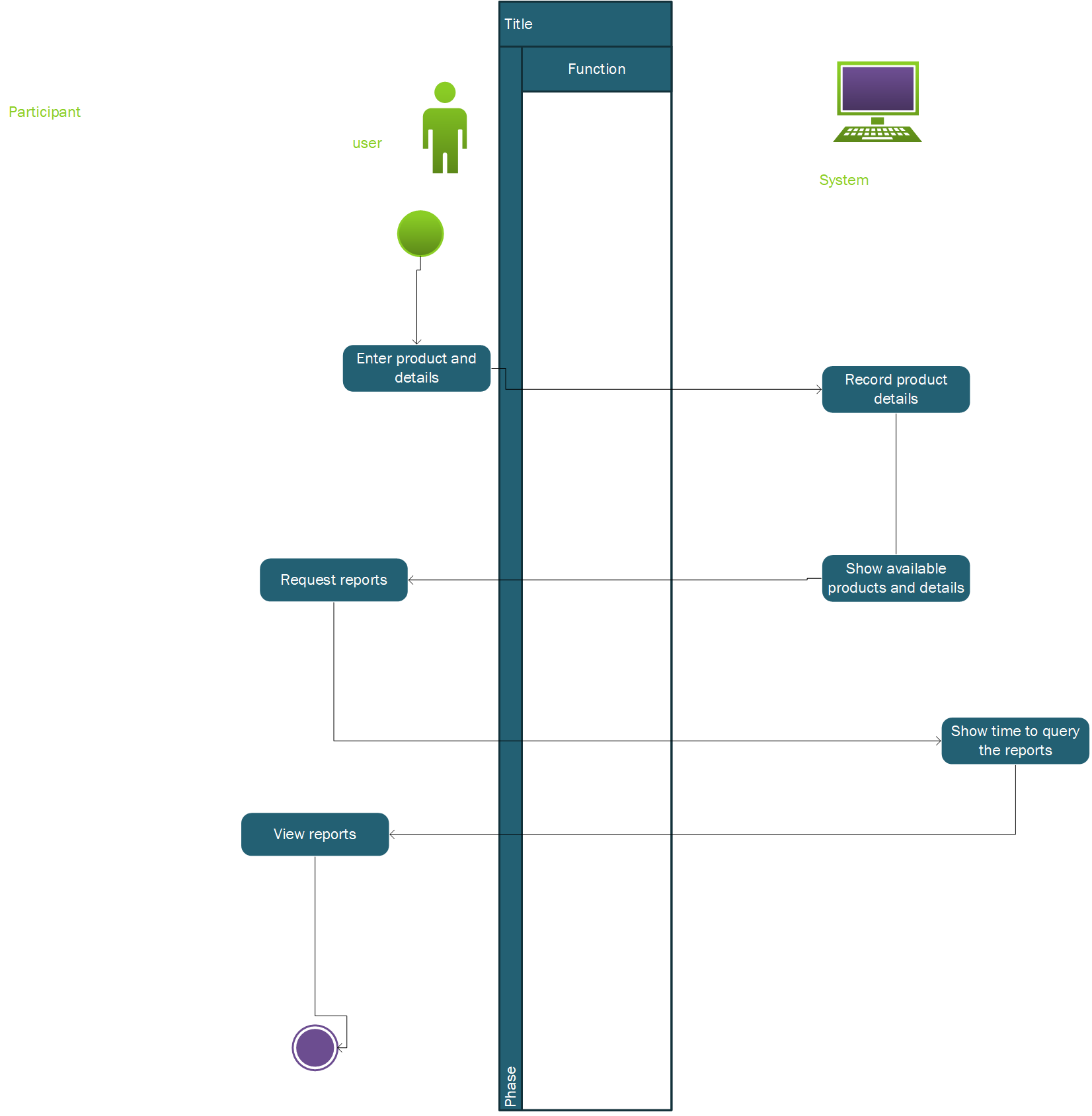


figure 3.2. 7Activity diagram for the participant

## Design Rationale

The system going to be designed with a central data repository, that stores all the data about the products posted by the participants and are accessed from the database by the client-side users. There is a file that stores data temporary from the participant command line and another program that continuously runs to fetch this data on the file and stores it in the database. This program also does fetch customer requests from the file and returns the response to the file where they are later read from the file by the client from the terminal. The object-oriented paradigm is considered for the project because it is easy to troubleshoot given the modularity used, reuse of the code through inheritance and effective through polymorphism thus easing the problem solving.

For structured programming, reusability of code ,readability and security would be a bit hard to implement thus choosing object-oriented approach for the development of the system thus making it secure, efficient and reliable for the users to attain their needs while using the system

# DATA DESIGN

## Data Description

This is where data attributes are explained.

|  |  |
| --- | --- |
| ENTITY | ATTRIBUTES |
| Customer | customerName |
|  | address |
|  | status |
| product | productName |
|  | productID |
|  | quantity |
|  | rate |
|  | description |
| participant | Password |
|  | participantID |
|  | participantName |
|  | dateOfBirth |
| perfomanceReport | participantName |
|  | awardName |
| businessAdministrator | password |
|  | logInTime |
|  | LogoutTime |
| booking | bookingNumber |
|  | productName |
|  | customerName |
| ProductCategory | categoryId |
|  | name |

Table 4.1. 1Database entities with respective attributes

The data design of the system is based on enhanced entity relational modeling approach which is represented by the EERD (Enhanced Entity Relational Diagram) below.

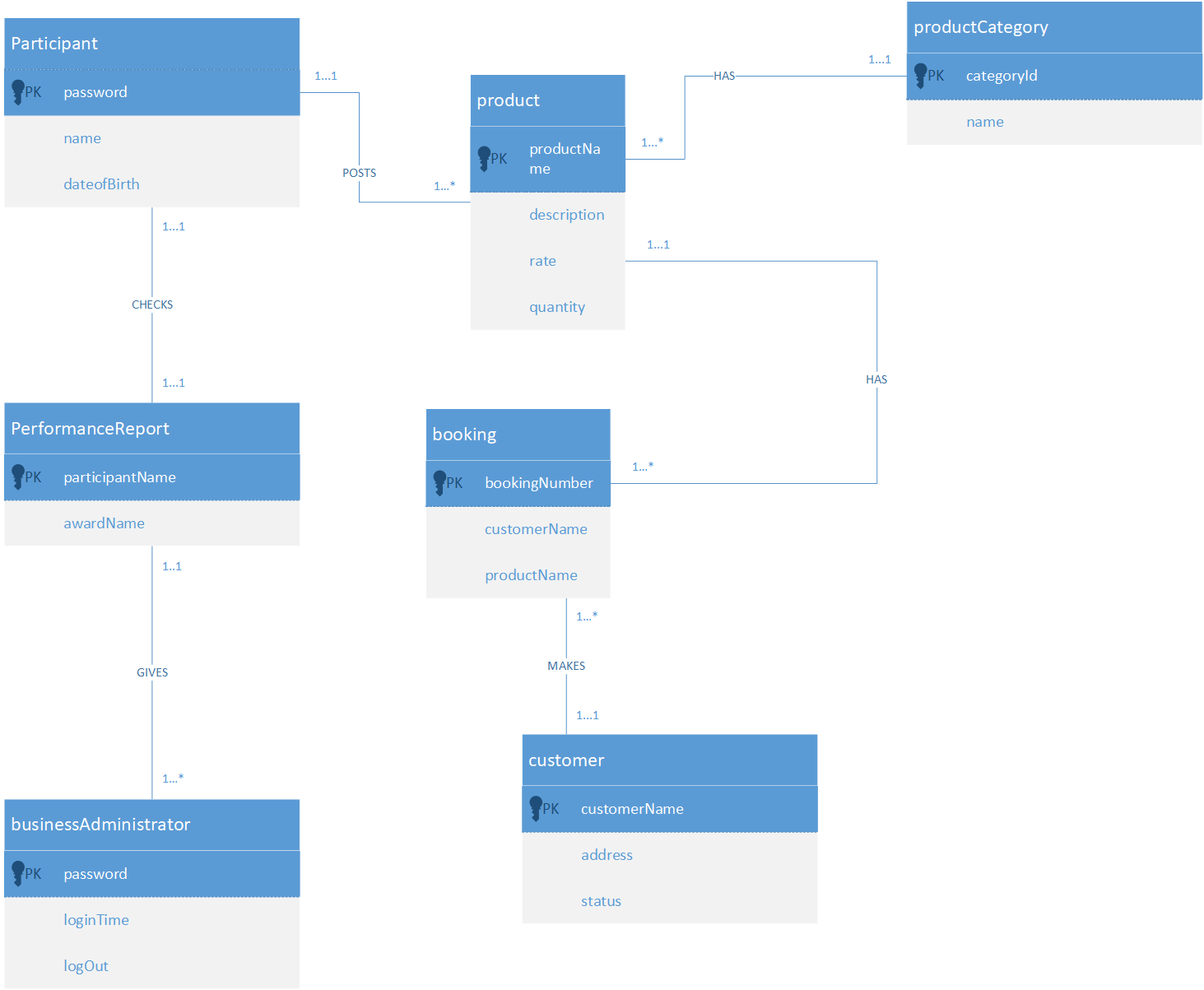


figure 4.1. 1Eerd for the entities of the system in database

**PHYSICAL DESIGN**

**Customer**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data type | size | key |
| customerName | varchar | 30 | Primary key |
| address | varchar | 30 |  |
| status | varchar | 40 |  |

Table 4.1. 2Customer

**Participant**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data type | Size | Key |
| ParticipantID | int | 10 | Primary key |
| password | varchar | 15 |  |
| participantName | varchar | 25 |  |
| dateOfBirth | date | 10 |  |

Table 4.1. 3Participant

**Product**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data type | size | key |
| productID | int | 10 | Primary key |
| productName | varchar | 30 |  |
| description | varchar | 250 |  |
| quantity | int | 8 |  |
| rate | int | 50 |  |

Table 4.1. 4ProductTable

**PerformanceReport**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data type | size | key |
| participantName | varchar | 30 | Primary key |
| awardName | varchar | 30 |  |

Table 4.1. 5PerformanceReport

**BusinessAdministrator**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data type | size | key |
| Password | varchar | 30 | Primary key |
| logInTime | date |  |  |
| logOutTime | date |  |  |

Table 4.1. 6BusinessAdministrator

**Booking**

|  |  |  |  |
| --- | --- | --- | --- |
| attribute | Data type | size | key |
| BookingNumber | Int | 10 | Primary key |
| ProductName | Varchar | 30 | Foreign key |
| CustomerName | varchar | 30 | Foreign key |

Table 4.1. 7Booking

**ProductCategory**

|  |  |  |  |
| --- | --- | --- | --- |
| attribute | Data type | size | Key |
| CategoryId | Int | 10 | Primary key |
| name | varchar | 30 |  |

Table 4.1. 8ProductCatergory

## Data Dictionary

**Customer**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data type | size | description |
| CustomerName(PK) | varchar | 30 | This unique attribute is used to identify the user by his name. |
| address | varchar | 30 | This attribute is used to identify the location of the customer |
| status | varchar | 40 | This attribute is used to identify whether the customer is a first time or old customer. |

Table 4.2. 1Customer

**Participant**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data type | Size | description |
| ParticipantID{pk} |  |  | This is a unique attribute used to identify the ID of the participant. |
| Password | varchar | 15 | This is an attribute used to identify the password of the participant. |
| participantName {fk} | varchar | 25 | This is an attribute used to identify the name of the participant |
| dateOfBirth | date | 10 | This attribute is used to identify the date of birth if the participant. |

Table 4.2. 2Participant

**Product**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data type | size | description |
| productID{pk} | int | 10 | This unique attribute is used to identify the id of the products |
| productName | varchar | 30 | This attribute is used to identify the name of the products. |
| description | varchar | 250 | This attribute is used to identify the description of the products |
| quantity | int | 8 | This attribute is used to describe the amount of available products. |
| rate | int | 50 | This attribute is used to identify the price rate of the products. |

Table 4.2. 3Product

**PerformanceReport**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data type | size | Description |
| participantName {pk} | varchar | 30 | The attribute is used to identify the participant name in the performance Report table |
| awardName | varchar | 30 | This attribute is used to identify the awardName in the table |

Table 4.2. 4PerformanceReport

**BusinessAdministrator**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data type | size | Description |
| Password {pk} | varchar | 30 | This attribute represents the password of the administrator. |
| logInTime | date |  | This attribute identifies the time at which the administrator logs in. |
| logOutTime | date |  | This attribute identifies the time at which the administrator logs out of the system |

Table 4.2. 5Administrator

**Booking**

|  |  |  |  |
| --- | --- | --- | --- |
| attribute | Data type | size | description |
| BookingNumber {pk} | Int | 10 | This attribute is used to identify the booking number which is identifies the booking. |
| ProductName {fk} | Varchar | 30 | This attribute is used to identify the name of the product. |
| CustomerName {fk} | varchar | 30 | this attribute is used to identify the name of the customer. |

Table 4.2. 6Booking

**ProductCategory**

|  |  |  |  |
| --- | --- | --- | --- |
| attribute | Data type | size | description |
| categoryId {pk} | Int | 10 | This attribute is used to identify the the category which by id. |
| categoryName | varchar | 30 | This attribute is used to identify the name of the category. |

Table 4.2. 7ProductCatergory

# COMPONENT DESIGN

**User Details**

|  |
| --- |
| **Start**  **If** command corresponds  Prompt user for name;  **If** name is not null  Prompt user for password;  **If** old user and details true  Login;  Access Homepage;  **If** old user and details false  Return an error message;  **If** user is new  Prompt user for name;  Prompt user for password;  Prompt user for product;  Prompt user for Date of Birth;  Register User;  Log in;  Access homepage;  **Stop.** |

Table 5. 1Capture User Details

**Data Request**

|  |
| --- |
| **Start**  **While**(true)  Cron job picks information from the file and inserts into database  If a participant makes a request for updates  Request is sent to file;  **If** Cron runs for next request  Time on request is shown;  Time on response is shown;  Time on response view is shown;  Information is then stored in Database;  **Stop** |

Table 5. 2Data Request

**Awarding**

|  |
| --- |
| **Start**  For every product bought  **If** customer buys one product  Participant is awarded one point;  **If** customer returns to buy another product  Participant is awarded 2 points;  **If** customer returns to buy more than one product  Participant is awarded 4 points;  Cumulative points awarded are summed up;  **For**  report generated on home page  **If** participant has highest cumulative points  Award is given to participant;  Award is stated in home screen;  **Stop** |

Table 5. 3Awarding

**Product**

|  |
| --- |
| **Start**  **For**  every product  Enter product details;  Show product quantity availability;  **If** product is bought  Available quantity reduces by 1 (-1);  Enter rate per time;  **Stop** |

Table 5. 4Booking

**Report Generation**

|  |
| --- |
| **Start**  **For**  every report generated  **If** report is generated in table form  **Return** tables;  **If** report is generated in graph form  **Return** graph;  **If** report is generated on dashboard  Show dashboard;  **Stop** |

Table 5. 5Report generation

# HUMAN INTERFACE DESIGN

## Overview of User Interface

ANKA Business Support Services will be used by the participants, customers and an administrator to carry out five main activities. The participants will: join the competition using their personal information, post their products and descriptions. The participant will be able do all of this using the CLI.

The customers will be able to: book for the products, browse and see participant information. The customer uses the web interface.

On the web interface, administrators can sign in and view the reports.

**Joining Competition.**

The user(participant) is also a merchant who is not registered in the software. Using their name, password of choice and date of birth the user is able sign up. He is prompted to enter the password only after he has entered his name. He does this after selecting the command corresponding to this functionality. The participant is continuously prompted as this step is crucial. The participant cannot access the other functionalities without completing this functionality.

The participant does not have to do this step every time he uses the CLI. A registered participant will use the log in function. During Log in, the user will be prompted for his name and password.

**Posting Products and respective Descriptions.**

The Participant will be able to post their products after signing up or logging in. The products can be only of one kind i.e. sells stationary only. Once the participant verifies the kinds of products, he then gives a detailed description which includes but not limited to available quantity and rate price. The participants may increase their products but they are required to add a description too. The participant can request for an update. This update will be show the next time he logs in.

**Booking Products**

The Users(customers) will be able to book products of interest. This will be on web interface which has all the participants. The web page offers many other functionalities as shall be discussed. The customer will use the description of the participants' products to identify the products. The customer can then select the product. The customers are prompted to enter their details(name, address and password) before the item is purchased. Once filled, the items can then be purchased. The returning customer will sign up. And their information will be loaded on the page. Once item is purchased, the item number will automatically decrease.

**Browse and view participant Information.**

The customers interested in the individual participants shall use the same page to access the participants details. The customer simply clicks on the participants tab which reveals all the details. The customers will be able to view the products sold by the particular participants. The viewed participants won’t know. The customer has constraints to some of this information for example they cannot see the date of birth. The customers can use this information to contact the participant.

**Signing in; Report viewing**

The reports generated by the software can be viewed by the administrator. He must sign in on the web interface using his name and password. His functionalities are limited to viewing since most of the work is done by the software. The administrator can view the best performing participant and award him too. The points earned by each of the individual participants are shown on the page.

## Screen Images

**Home Page**

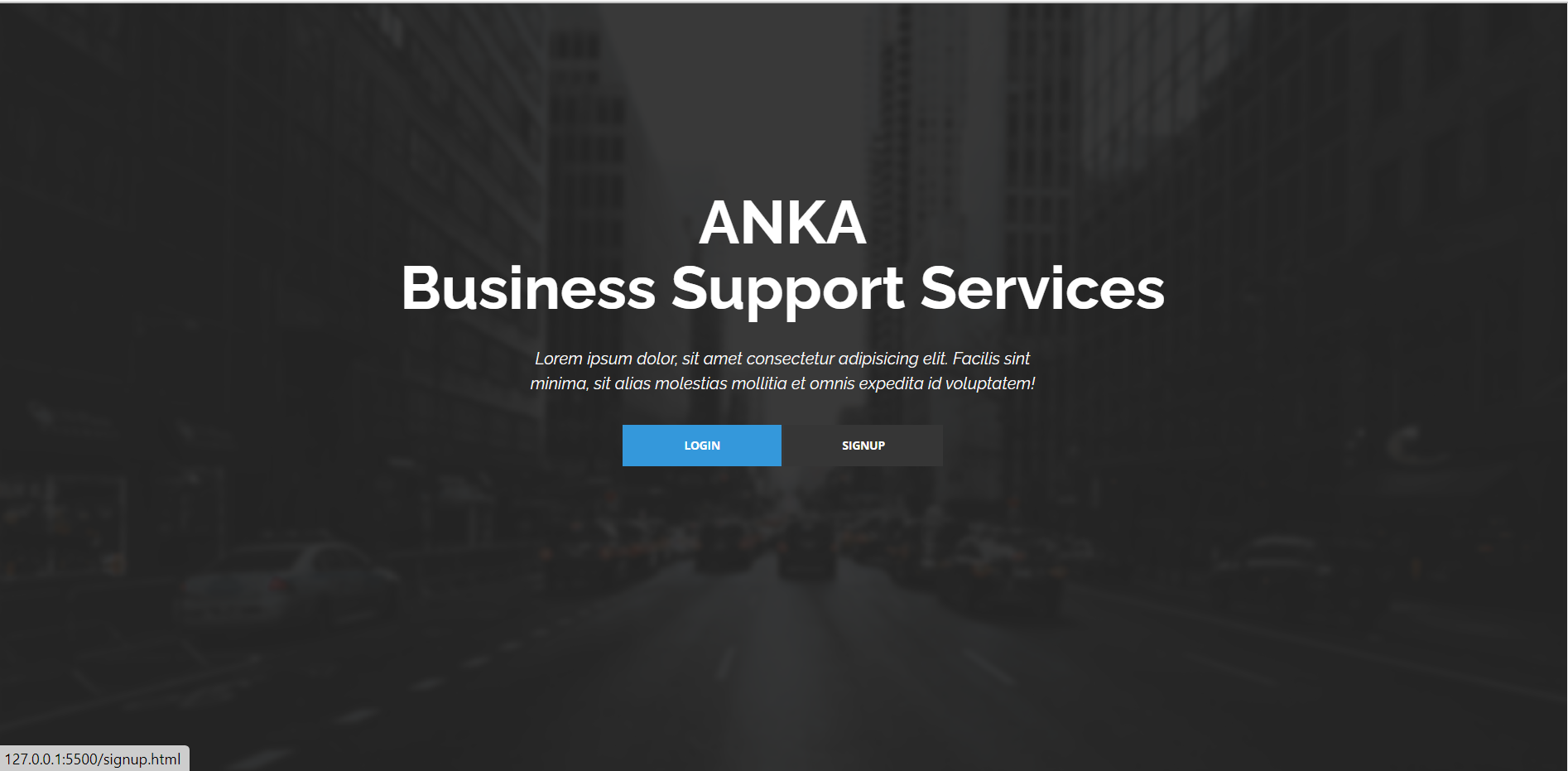
****

figure 6.2. 1Homepage of ANKA business support system website

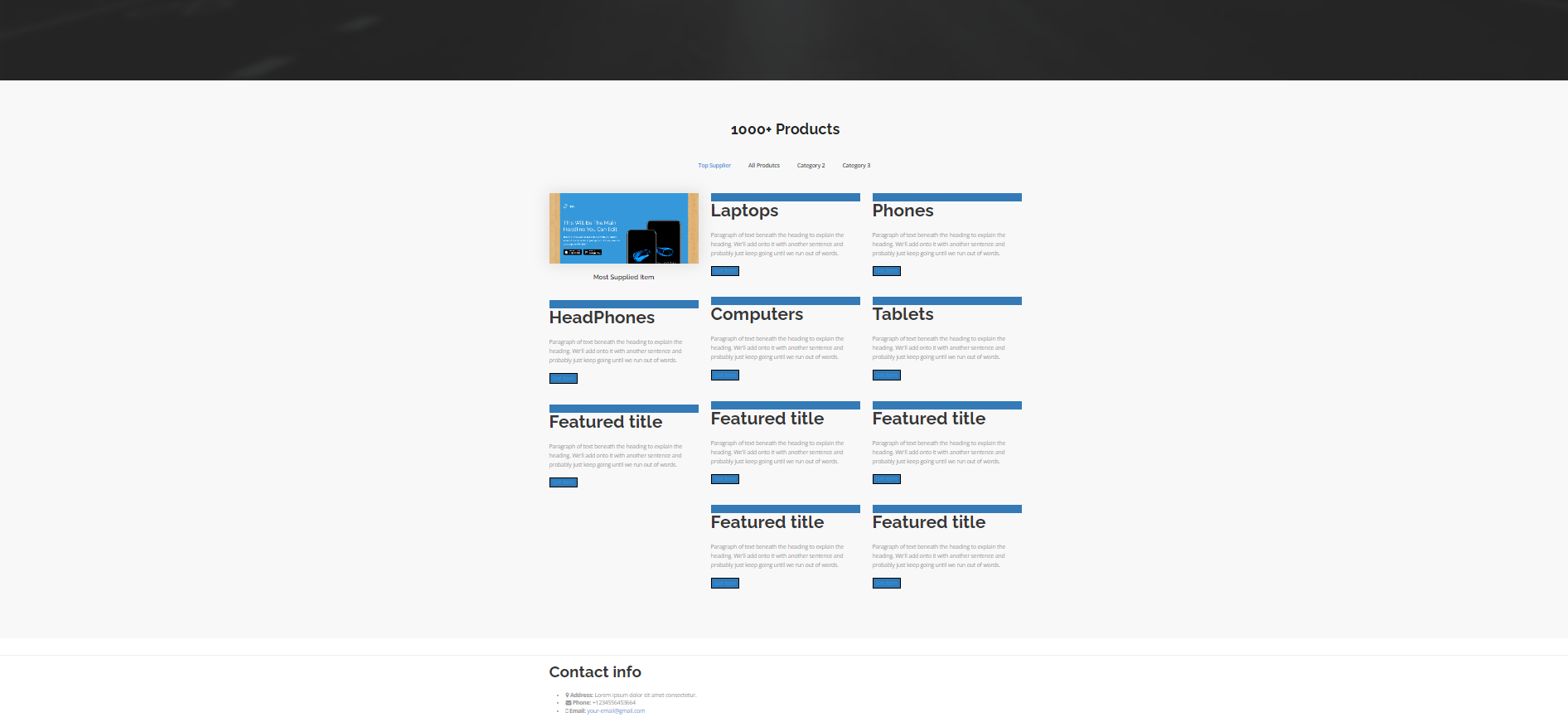
****

figure 6.2. 2Product section on the website

**Joining Competition, posting products and respective descriptions.**

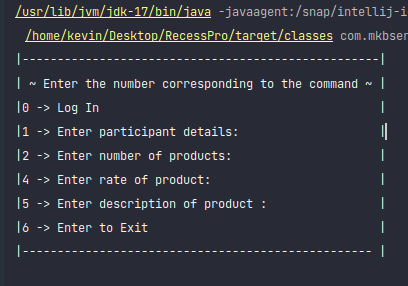
****

figure 6.2. 3Command line Interface for participants to post products

**Purchasing Products**

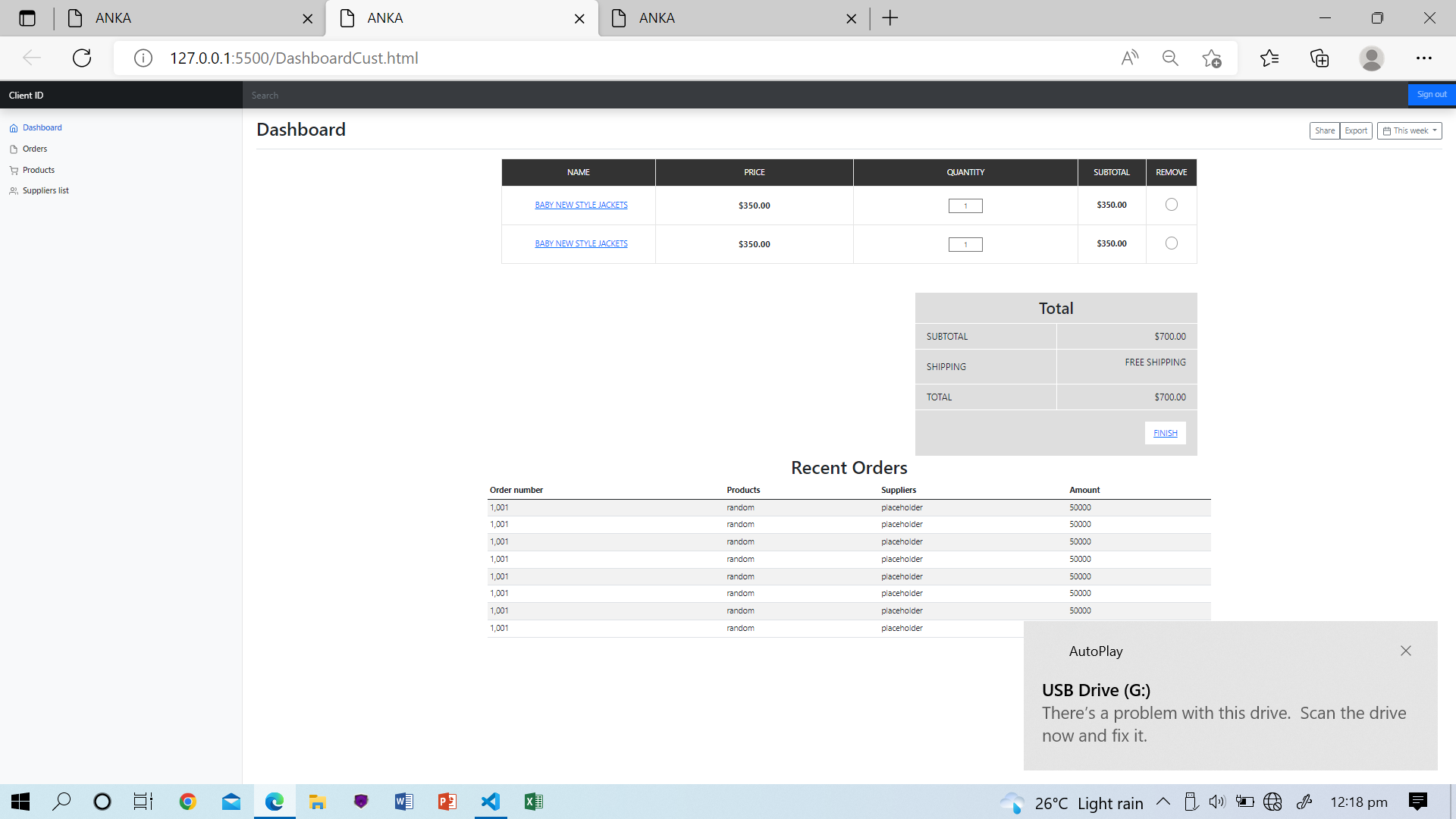
****

figure 6.2. 4Ordering section for customers to request products

**Logging in;**

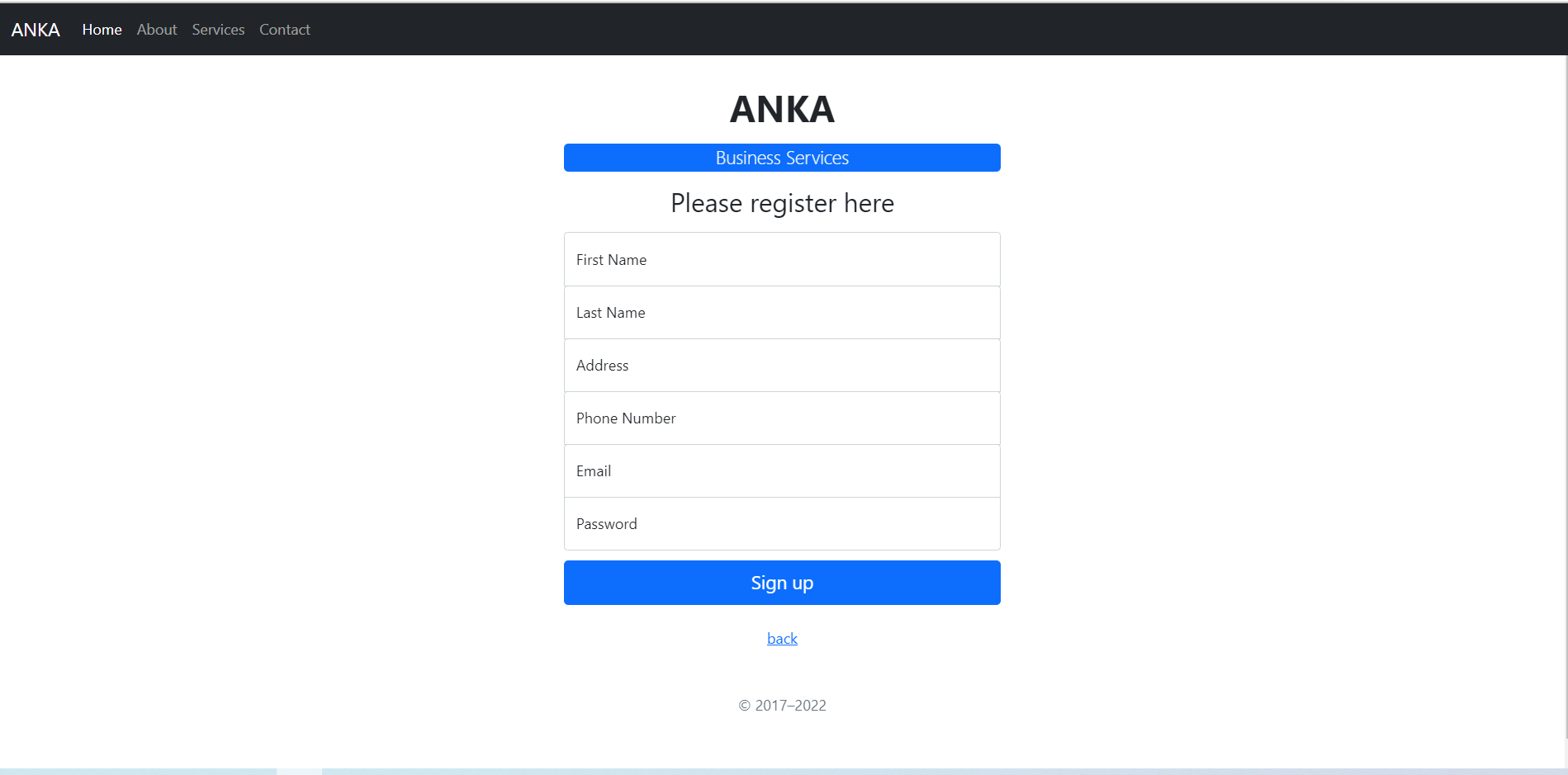
****

figure 6.2. 5login Screen for users to access the system via the web interface

**Report viewing**

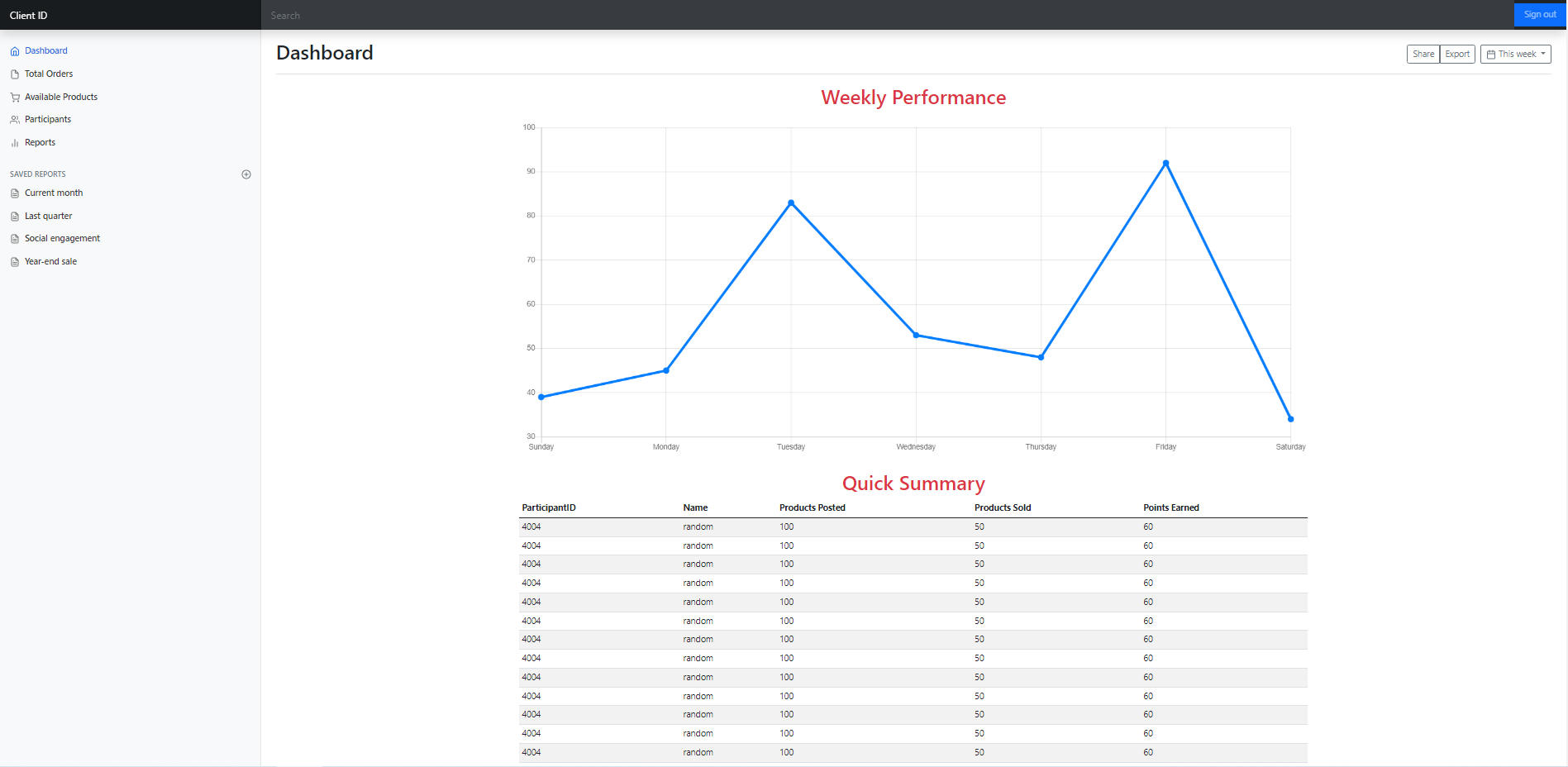
****

figure 6.2. 6Administrator Dashboard to view reports and award participants