a 

Republic of Cameroon

\*\*\*

Peace-Work-Fatherland

\*\*\*

IAI Cameroon

Email: contact@iaicameroun.com

République du Cameroun

\*\*\*

Paix-Travail-Patrie

\*\*\*

IAI Cameroun

Email: contact@iaicameroun.com

REPORT

**Group Members:**

-ONANA JACQUES MARIE 80%

-NOUTCHOGUE KAMLA 20%

**BA2B**

Academic Year 2024-2025

Tutor: Mr. FOMEKONG

**JAVA**

**PROGRAMMING:**

**REPORT ON THE ANALYSIS OF AN OUTDOOR GEAR MANAGEMENT SYSTEM**

**PREFACE**

The project aims to design and implement a simplified outdoor gear management stock app. The application has some features including user management, product management and order management. The project was developed using java and utilize a database management system to store and retrieve data.

The objective of this project was to demonstrate an understanding of object-oriented programming concepts, system analysis, database design and software development. The project was complete within a one-week timeframe.

The project scope includes the following features:

-Employee management: create, read, update and delete (**CRUD**)employees. -Product management: **CRUD** products -Order management: Create and manage orders.

This document is focus on the analysis that was done from the requirement gathering to the implementation of the project.

Table des matières

[INTRODUCTION 4](#_Toc190873774)

[INTRODUCTION TO THE ANALYSIS METHOD USE 5](#_Toc190873775)

[Presentation Of The Analysis Methodology 5](#_Toc190873776)

[THE ANALYSIS PHASE OF AN OUTDOOR GEAR MANAGEMENT APP 9](#_Toc190873777)

[A-Dynamic Diagrams 9](#_Toc190873778)

[1.USE CASE DIAGRAM 9](#_Toc190873779)

[2. ACTIVITY DIAGRAM 10](#_Toc190873780)

[3. SEQUENCE DIAGRAM 15](#_Toc190873781)

[4. COMMUNICATION DIAGRAM 19](#_Toc190873782)

[B-Static Diagrams 22](#_Toc190873783)

[1.CLASS DIAGRAM 22](#_Toc190873784)

[2.Package Diagram 23](#_Toc190873785)

[Component Of The System 24](#_Toc190873786)

[Database: 24](#_Toc190873787)

[IDE: 24](#_Toc190873788)

[USER GUIDE 25](#_Toc190873789)

# INTRODUCTION

Outdoor gear equipment refers to the various tools, devices and apparatus used for outdoor activities such as **camping, hiking, climbing**  and **other adventure sports.**

Our project was base on building an outdoor gear stock management app which is a software designed to help outdoor gear retailer, distributors and manufacturers manage their inventory of outdoor gear and equipment.

Our application features involve:

**-**Creation and deletion of employee Accounts in the database.

**-**The addition, update and deletion of product in the database.

**-**The creation of customer accounts int the database.

# INTRODUCTION TO THE ANALYSIS METHOD USE

To make our analysis we have choose **UML(**Unified Modeling Language**)** and the **2TUP** methodology which offers much to developers seeking for a user-centered approach and a wide scope in design.

## Presentation Of The Analysis Methodology

a. UML

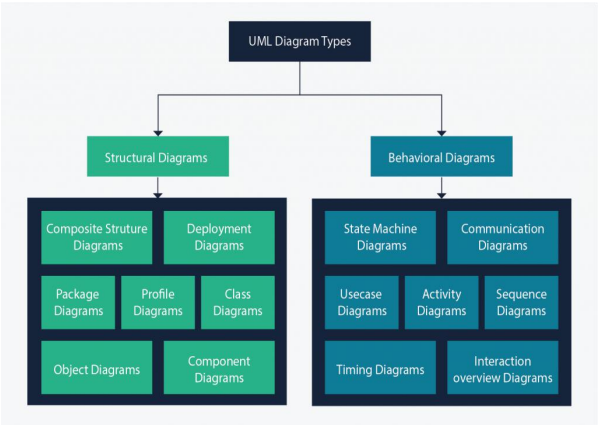
**UML(**Unified Modeling Language**)** is a standard notation for the modeling of real world objects as a first step in developing an object oriented design methodology. Its notation is derived from and unifies the notation of three object t-oriented design and analysis methodologies: Grady Booch's methodology for describing a set of objects and their relationships, James Rumbaugh's Object-Modelling Technique (OMT), Ivar Jacobson's approach which includes a use case methodology. Other ideas also contributed to UML, which was the result of a work effort by Booch, Rumbaugh, Jacobson, and others to combine their ideas, working under the sponsorship of Rational Software. UML captures information about the static and dynamic view of a system. UML 2.5 comprises of 14 diagrams which represent the different views of a system. The 14 diagrams can be subdivided into two, Static or structural and Dynamic diagrams. These diagrams include;

i. Static Or Structural Diagrams

* Class diagram;
* Object diagram;
* Component diagram;
* Deployment diagram;
* Composite Structure diagram;
* Package diagram;
* Profile diagram;

ii. Dynamic Or Behavioural Diagrams

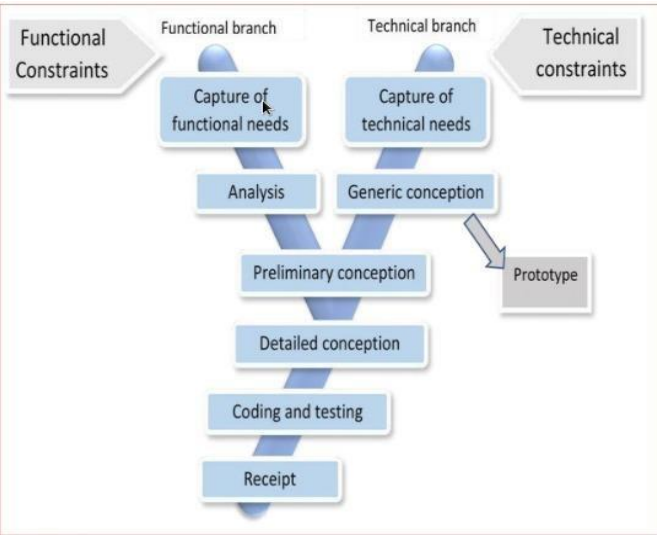
* Use case diagrams
* Activity diagram
* State machine diagram;
* Sequence diagram
* Communication diagram
* Global Interaction diagram
* Timing Diagram;



*Figure 1: UML Diagrams*

b. The Two Track Unified Process(2TUP)

2TUP is a unified process which is built on UML and has an objective to bring solution to constraints of functional and technical changes imposed on information systems by strengthening controls on development capacities. It proposes a Y-sharped development life cycle that separates the functional aspect from the technical aspects, and the merging of these two forms the implementation aspect. 2TUP distinguishes therefore two branches: the functional and technical branches, the combination of the result of these two branches forms the third: the realization branch where we realize our system. The diagram below illustrates the



*Figure 2: 2TUP life cycle*

i. The Left Branch(functional branch)

It captures the functional needs of a system. This ensures the production of software that meets the needs/requirements of the user. The analysis here consists of studying precisely the functional specification in order to obtain an idea of what the system is going to realize, and its result does not depend on any technology.

ii. The Right Branch(Technical branch)

The technical branch enumerates the technical needs and proposes a generic design validated by a prototype. The technical needs include constraints and choices related to the conception of the system, the tools and equipment as well as the integration constraint with the existing system condition.

iii. The Middle Branch(Realization or Implmentation branch)

In this branch, we study the preliminary conception, detailed conception, and documentation of the system. The realization branch supports the following:

**Preliminary conception:** This is the most sensitive step of 2TUP as it is the confluence of the functional and technical branch. It is completed when the deployment model, the operating model, the logical model, interphases and the software configuration model are defined. We have the following diagrams:

* + Component Diagram ;
  + Deployment Diagram ;
  + Package Diagram ;
  + Composite Structure Diagram ;

**Detailed conception:** This is the detailed design of each feature of the system. We have the

Following diagrams:

* Class;
* Object;
* Sequence;
* Timing Diagram;

**Coding and testing**: This is the phase where we program the designed features and test the coded features.

**The recipe**: Also known as the deliverables is the validation phase of the functions of the developed system.

# THE ANALYSIS PHASE OF AN OUTDOOR GEAR MANAGEMENT APP

After a brief Introduction to the methodology that was use in the analysis of our project lets move to the analysis phase that provide the different models of our system. Lets starts with the dynamic view of our system:

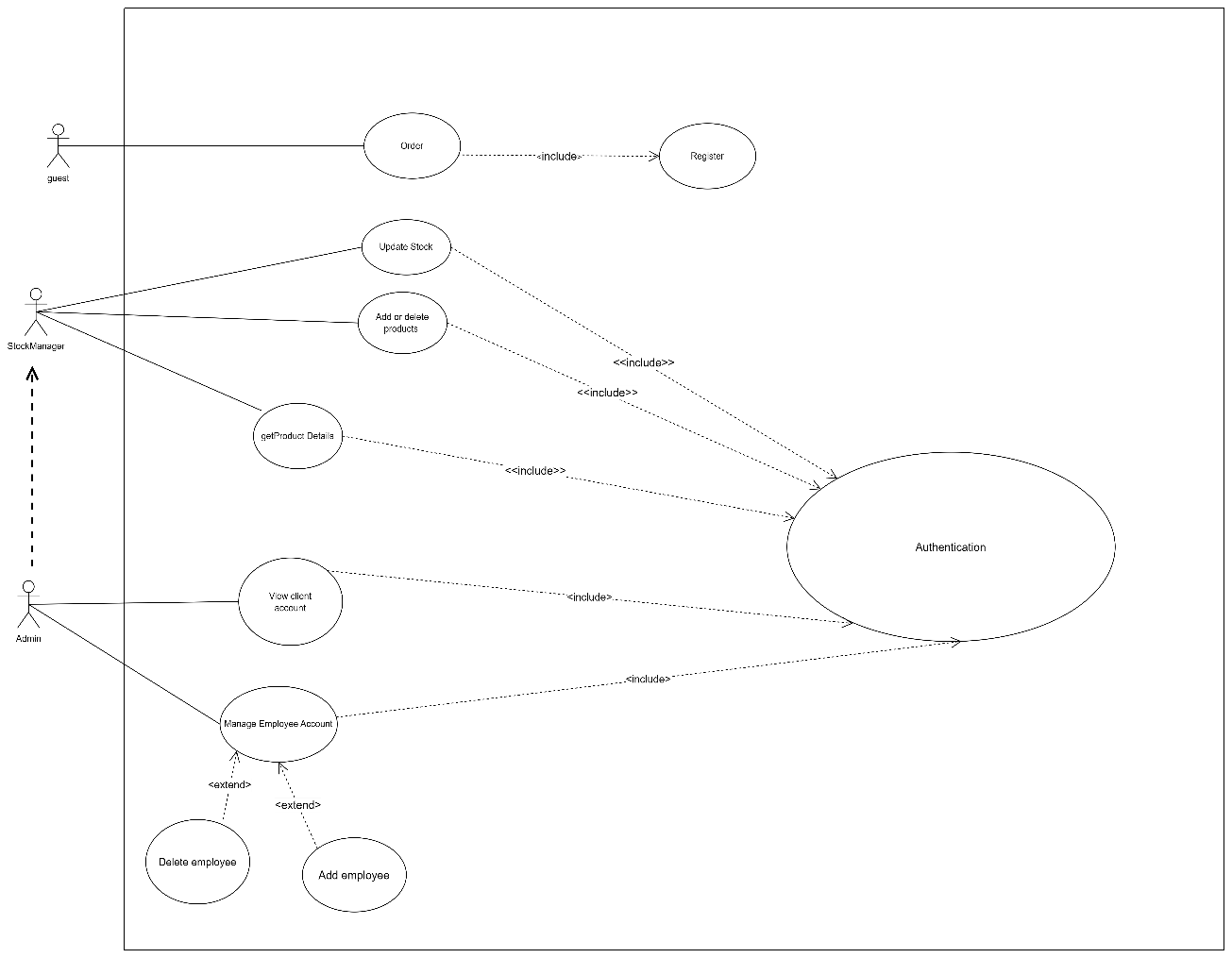
## A-Dynamic Diagrams

### 1.USE CASE DIAGRAM

A use case diagram shows the functionalities of a system, their inter-dependencies and how they relate with actors of the system. A use case is a specification of behavior. The main objectives of the use case diagram are:

* Provide a high-level view of the system.
* Identify the functions of the system.

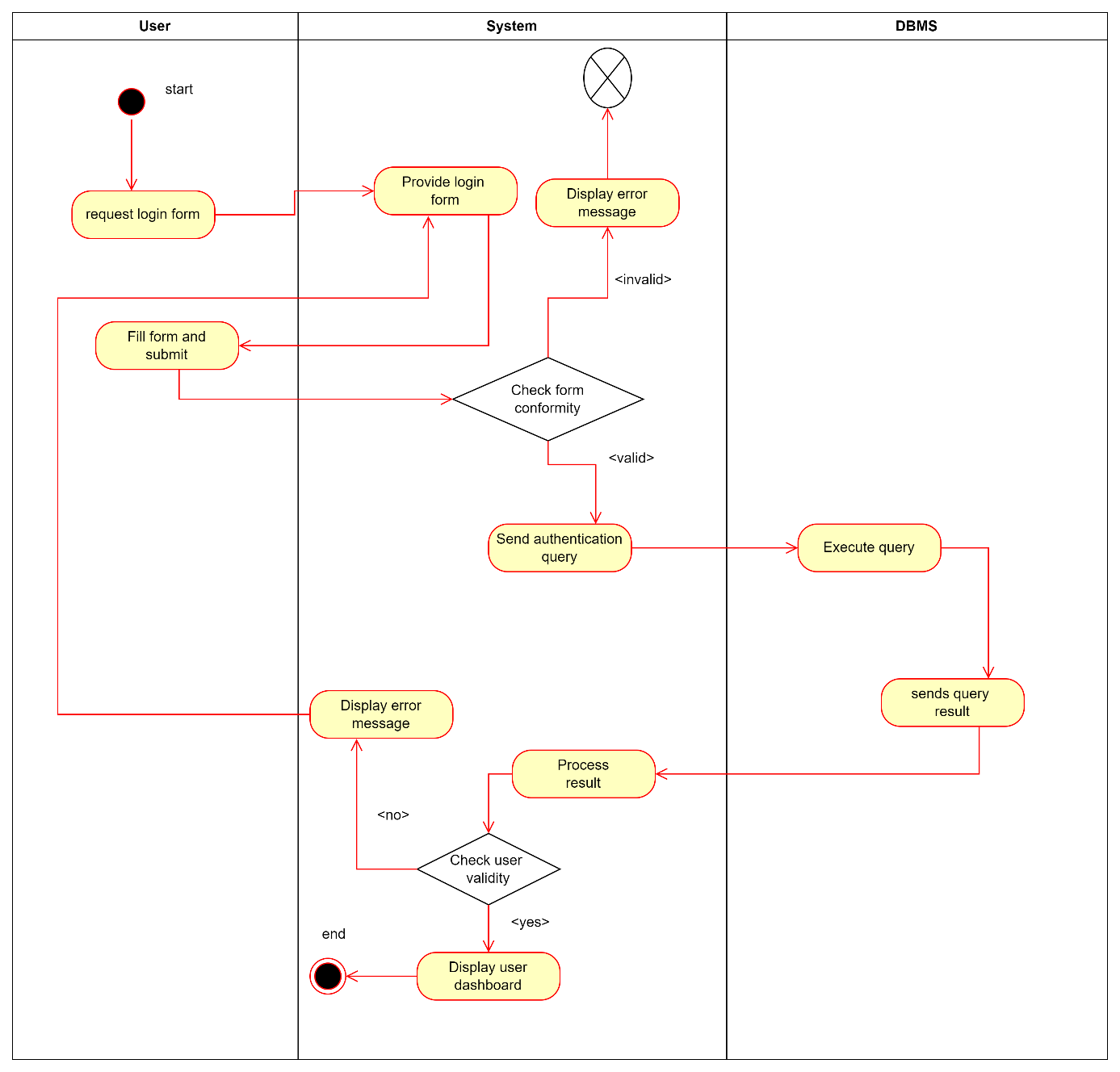
Use case diagrams are completed with a textual description of each use case that is intended to define the use case in greater details.



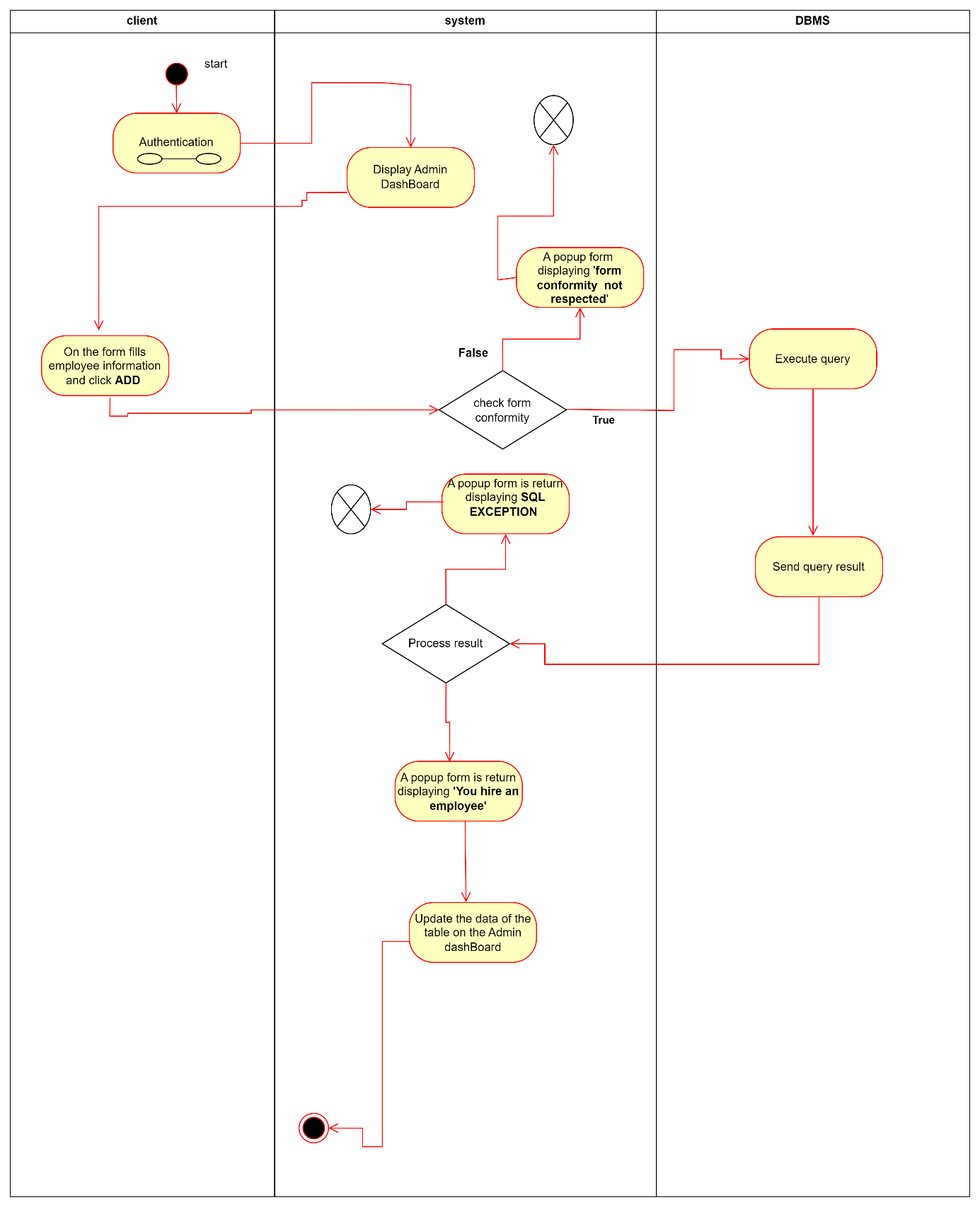
*Figure 3: General Use Case Diagram Of An OutDoor Gear Management App*

### 2. ACTIVITY DIAGRAM

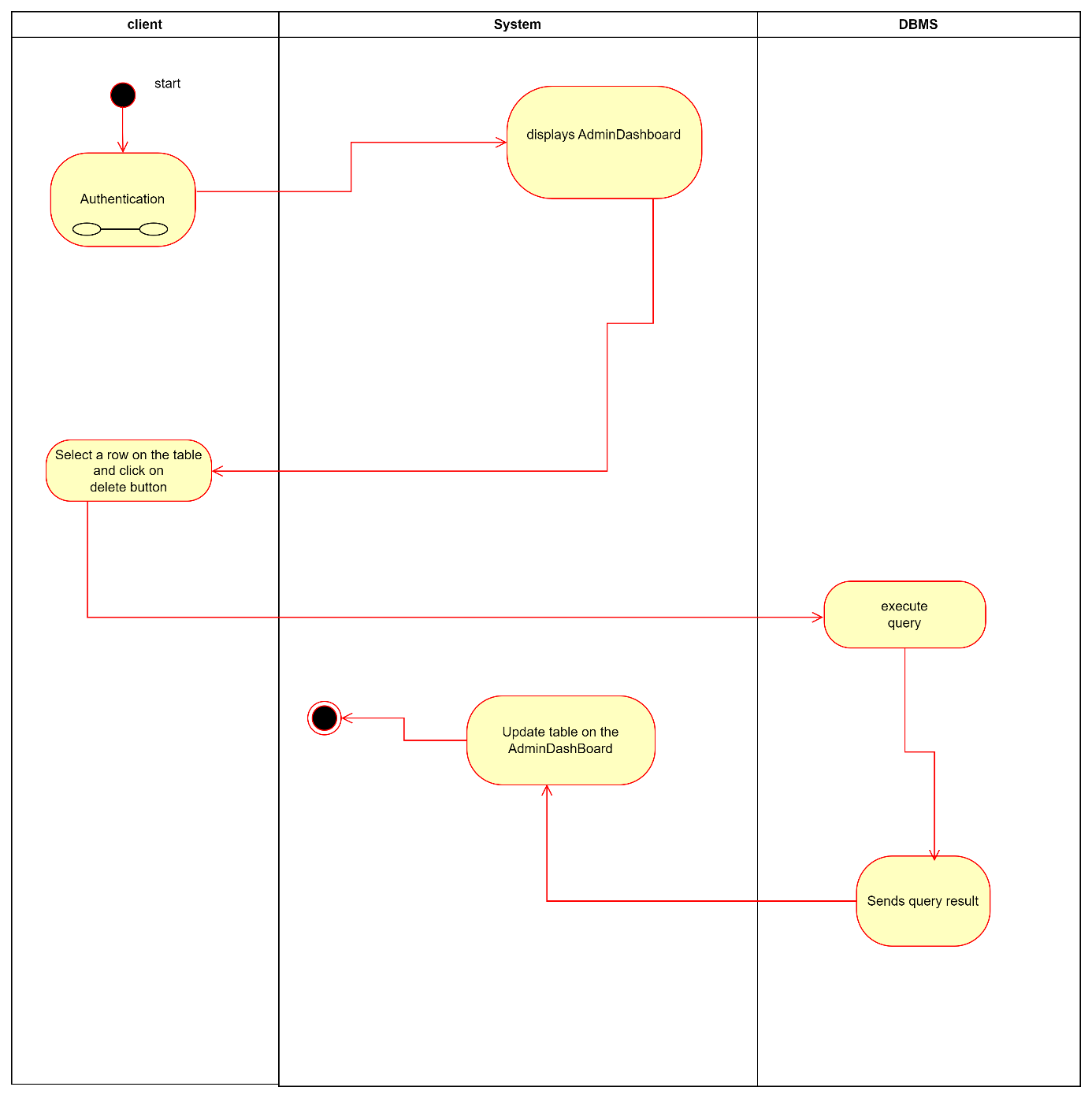
An activity diagram is a graphical representation of workflows that show the steps needed in the realization of a process; showing the details from a start point to an end point through all decisions and actions that can possible be performed. Activity diagrams are intended to model both the computational and organizational process. They flow can be sequential, branched or concurrent. Below is an activity diagram formalism.



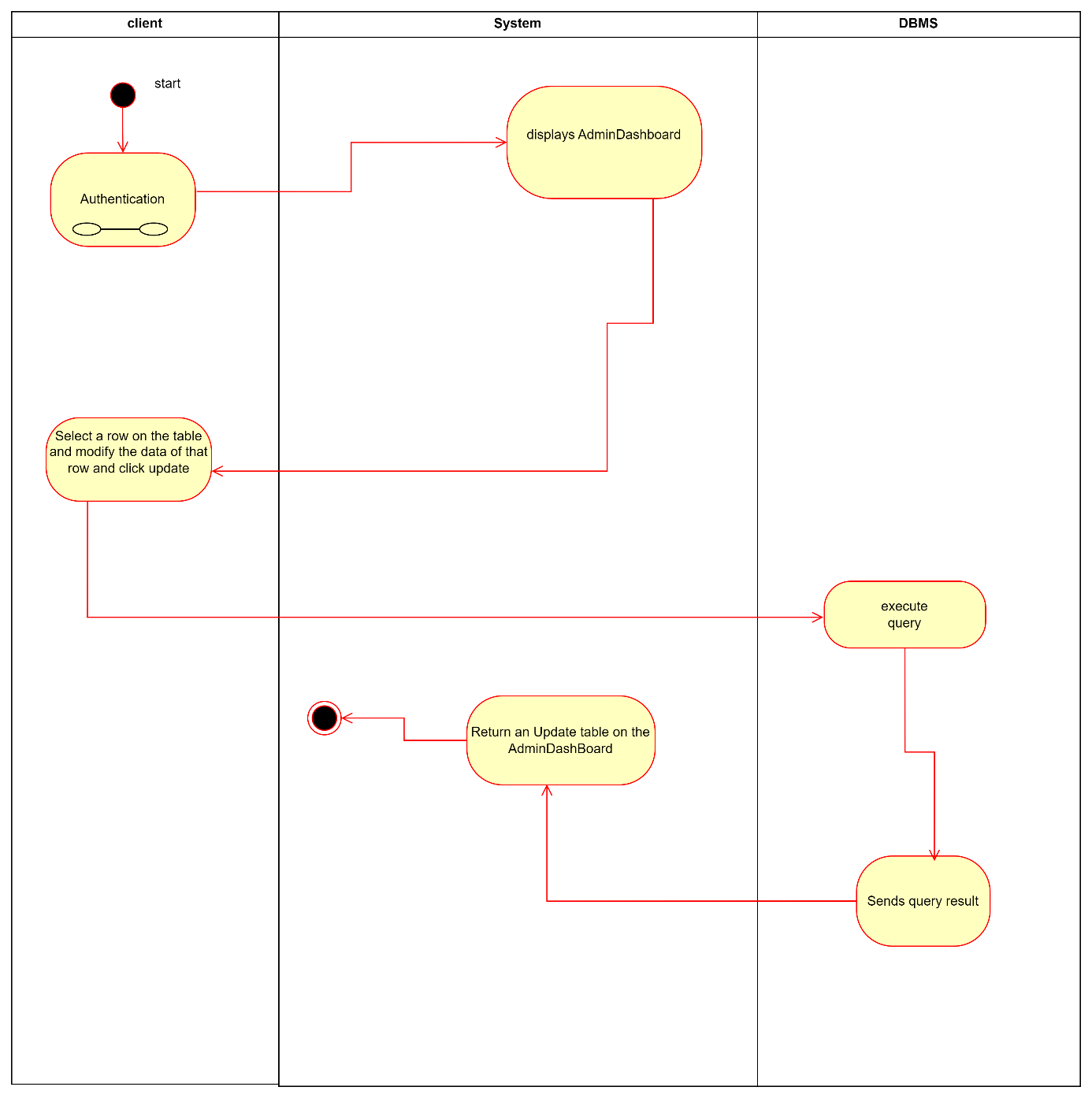
*Figure 4: Activity Diagram Of Authentiction*



*Figure 5: Activity Diagram of Employee Account creation and Addition Of Product*



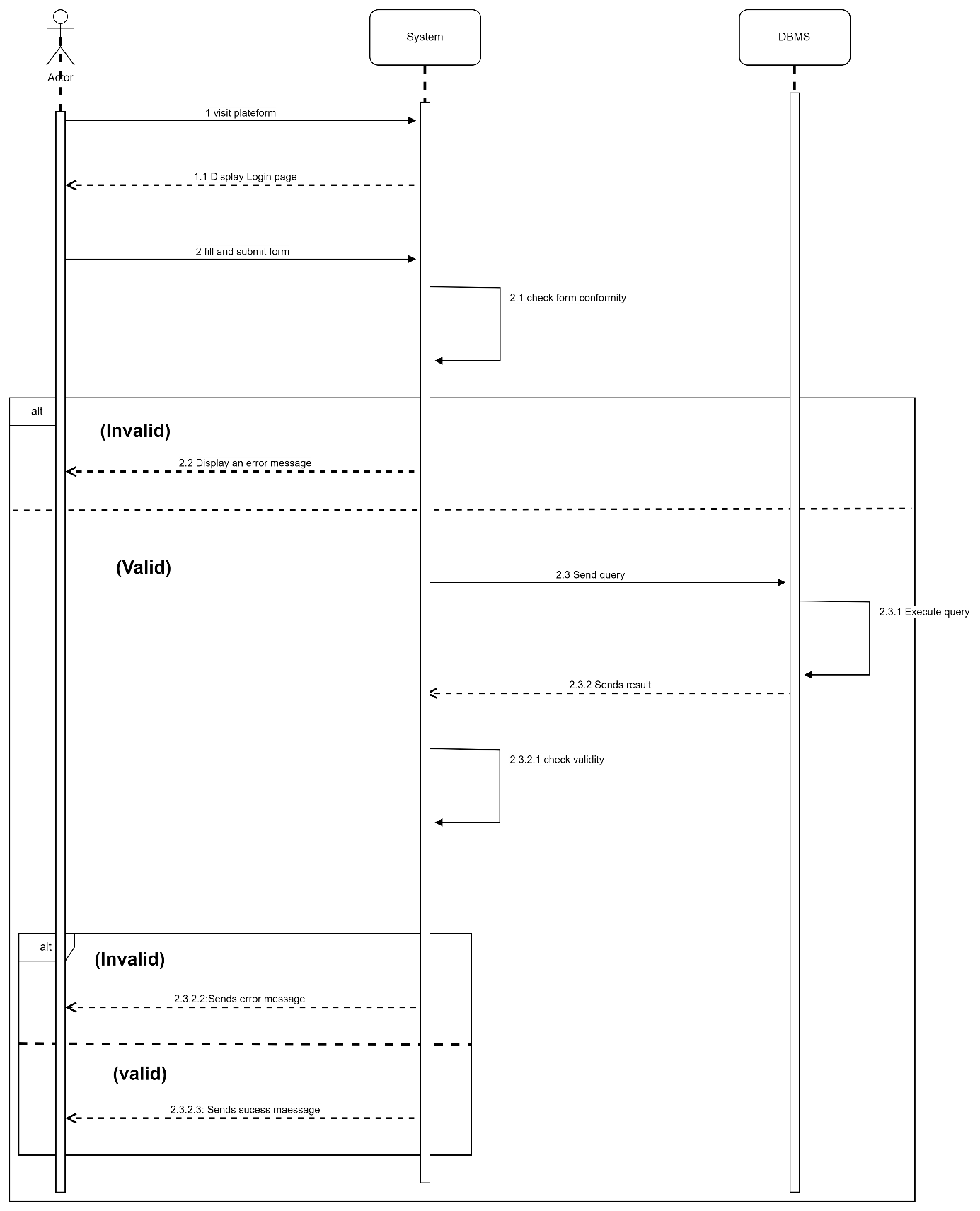
*Figure 6: Activity Diagram Of the deletion of Employee Account and Product Details*

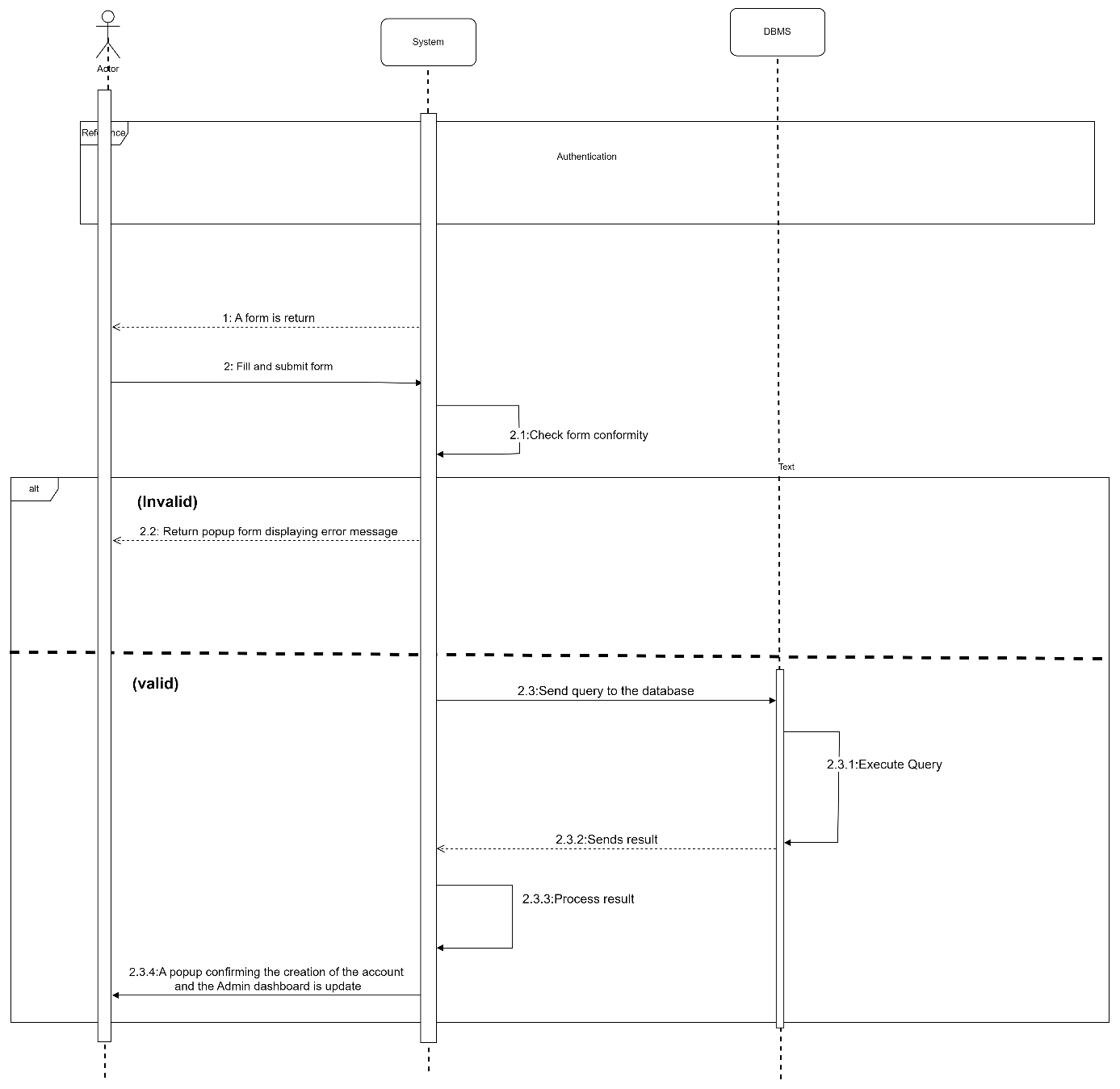


*Figure 7: Activity Diagram Of Update product information*

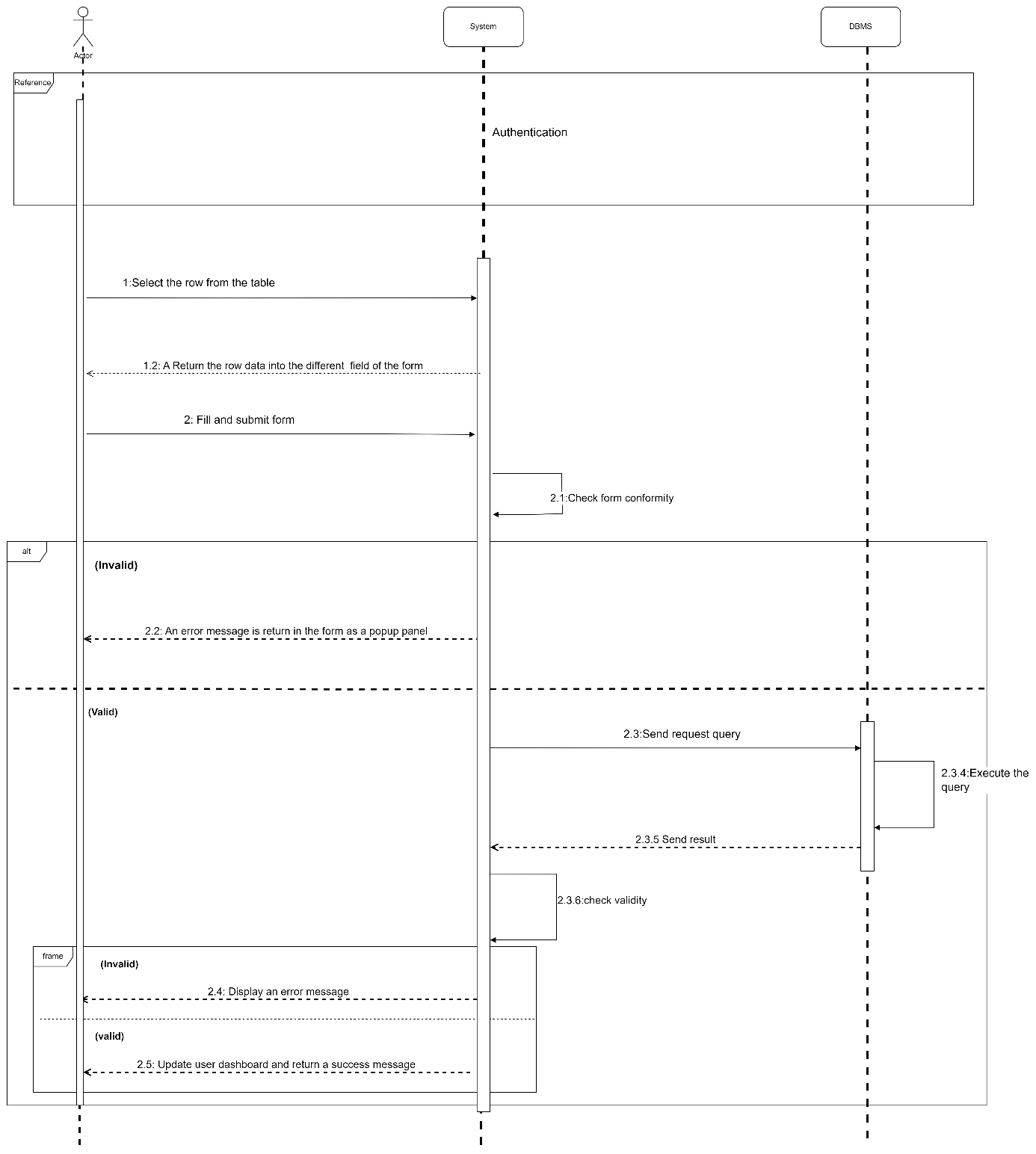
### 3. SEQUENCE DIAGRAM

A sequence diagram is an interaction diagram which represents the flow of message between elements in a system, it is termed as an event diagram. It portrays the communication between any two lifelines as a time-ordered sequence of events.

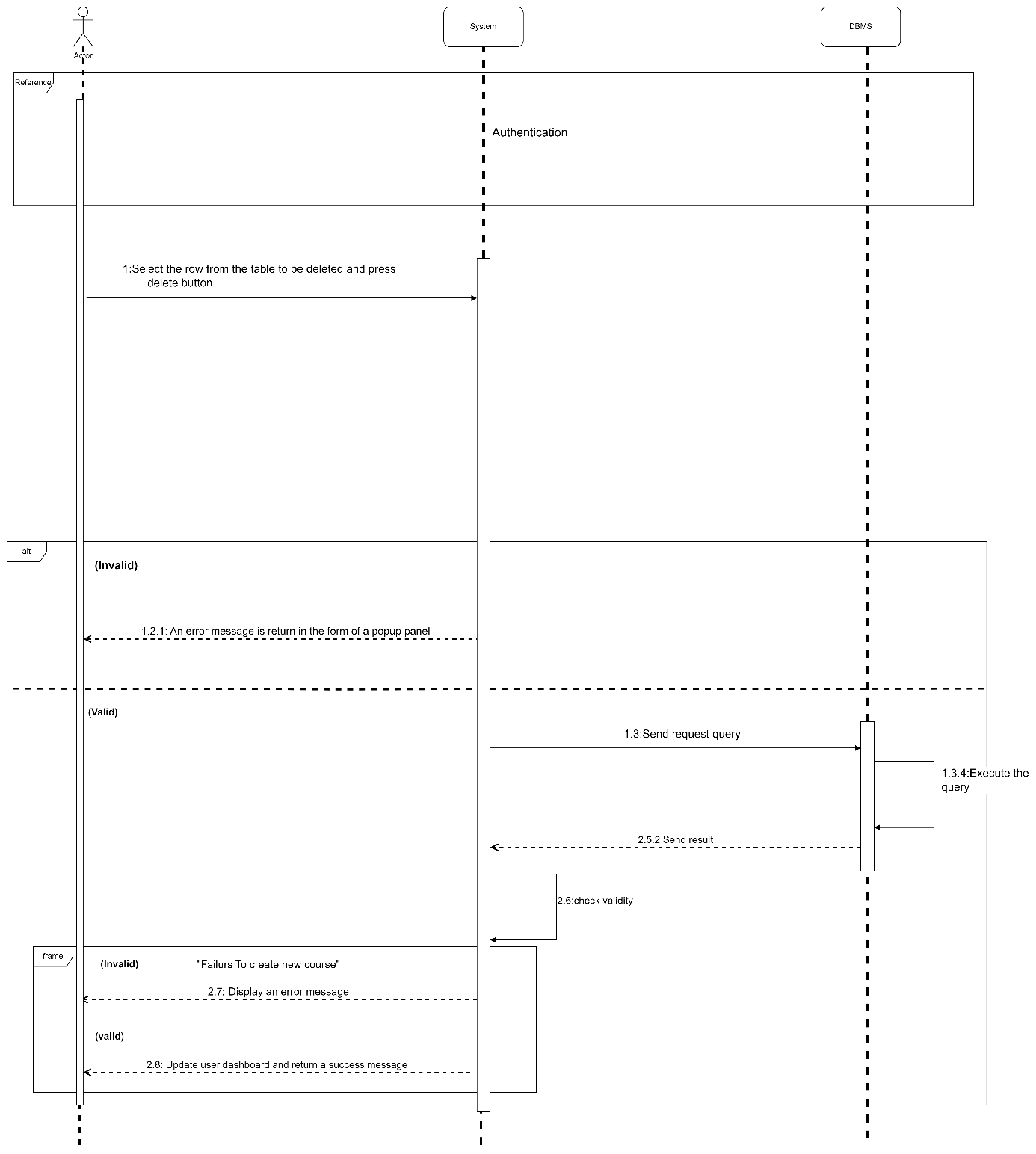
*Figure 8: Sequence Diagram Of Authentication*



*Figure 9: Sequence Diagram Of Employee Account Creation*



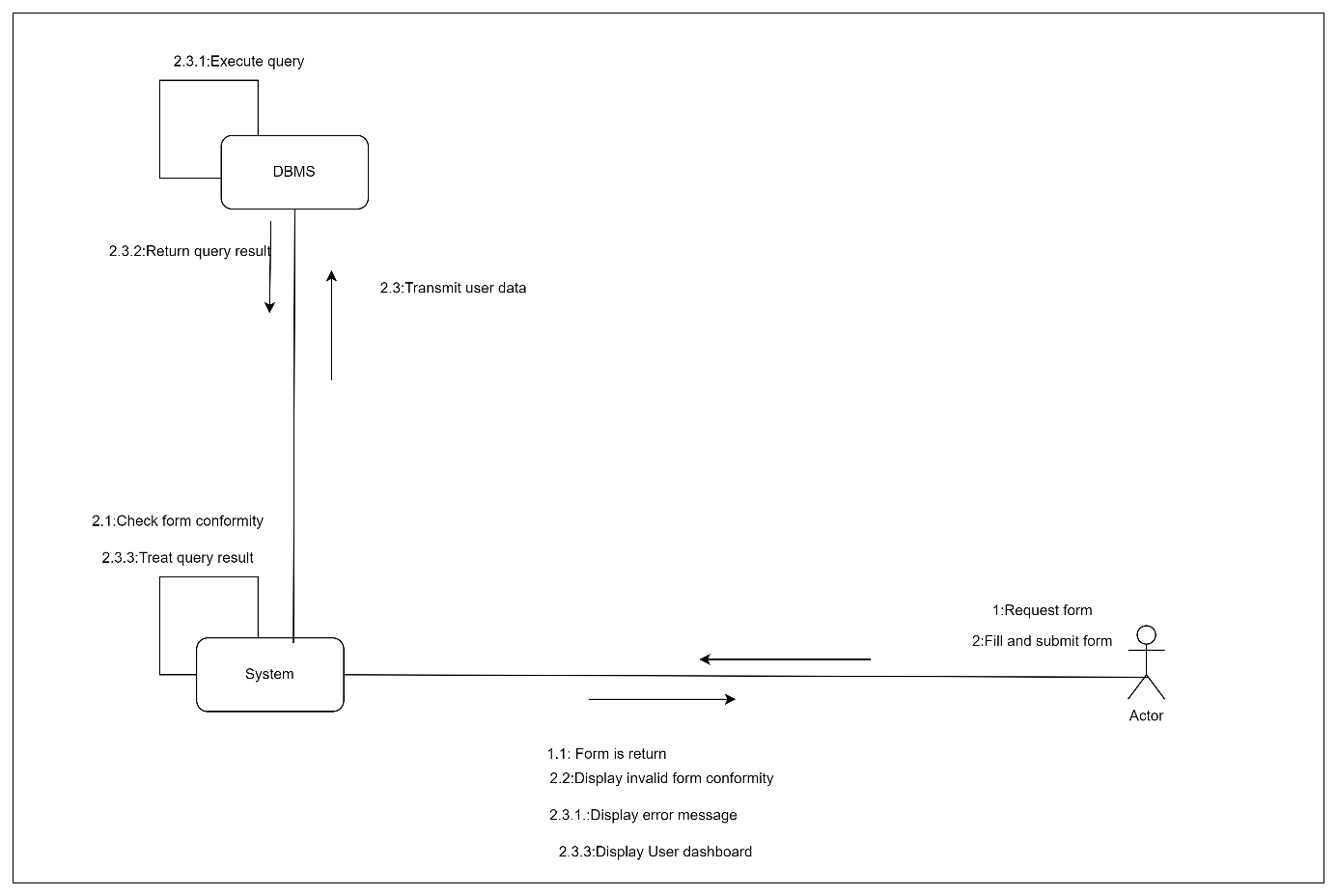
*Figure 10: Sequence Diagram Of Updating Product Details*



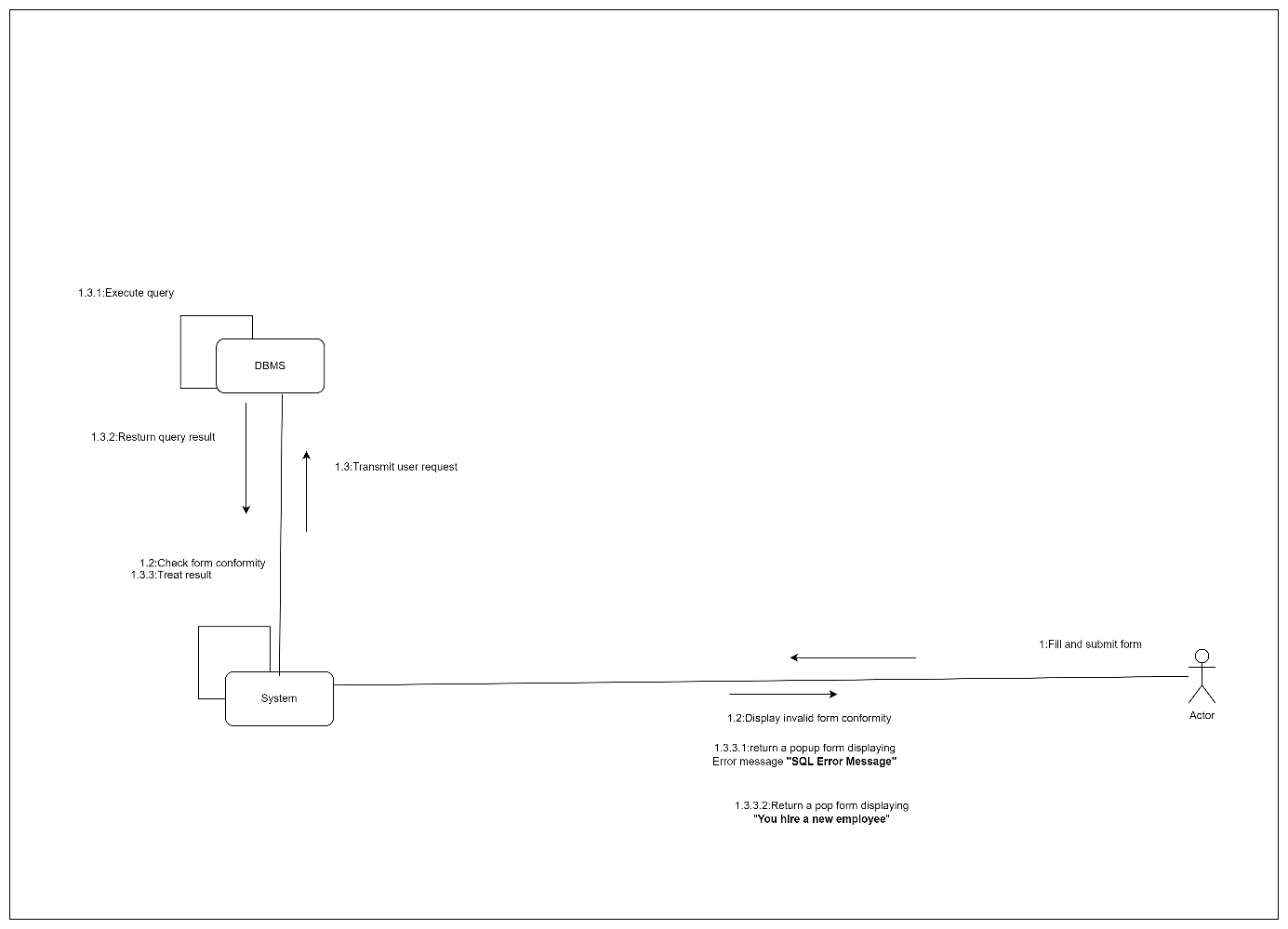
*Figure 11: Sequence Diagram Of Employee Account Deletion*

### 4. COMMUNICATION DIAGRAM

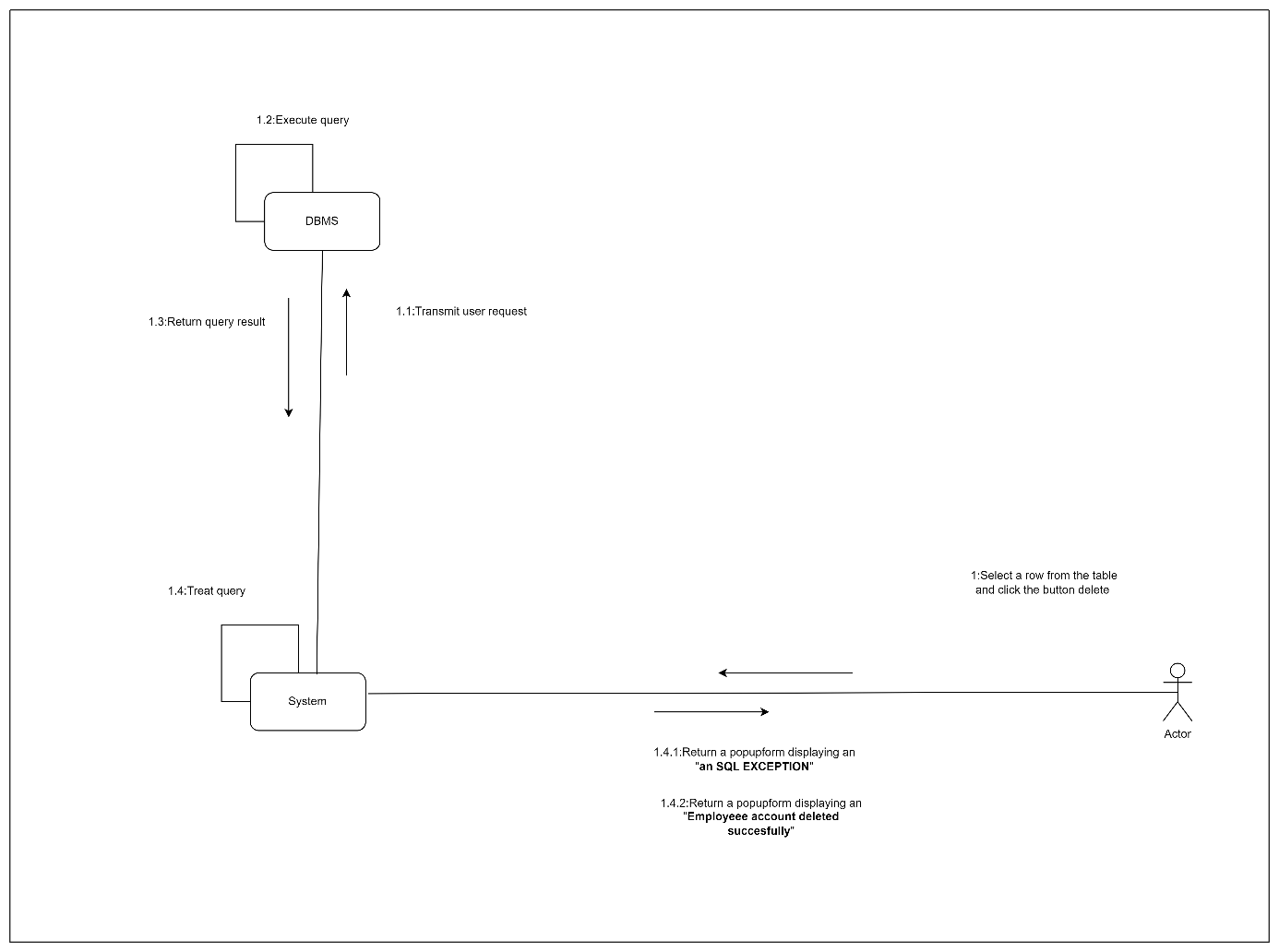
It is a diagram which is used to show the relationship between the actors of a system, both the sequence and the communication diagrams represent the same information but differently. Instead of showing the flow of message. It depicts the architecture of the object residing in the system as it is based on object-oriented programming.



*Figure 12: Communication Diagram Of Authentication*



*Figure 13: Communication Diagram Of Employee Account Creation*

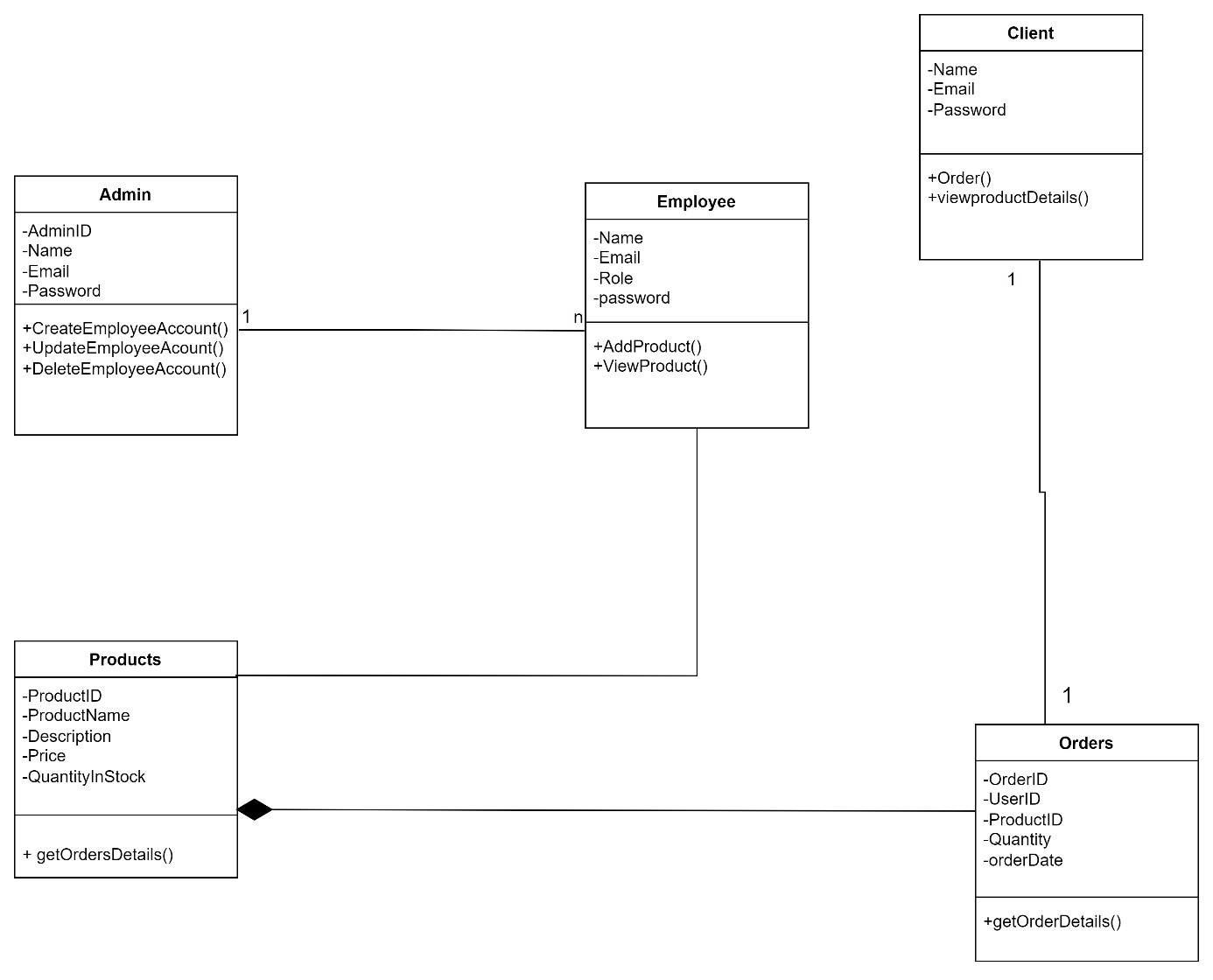


*Figure 14: Communication Diagram Of Employee Account Deletion and Product Details Deletion*

## B-Static Diagrams

### 1.CLASS DIAGRAM

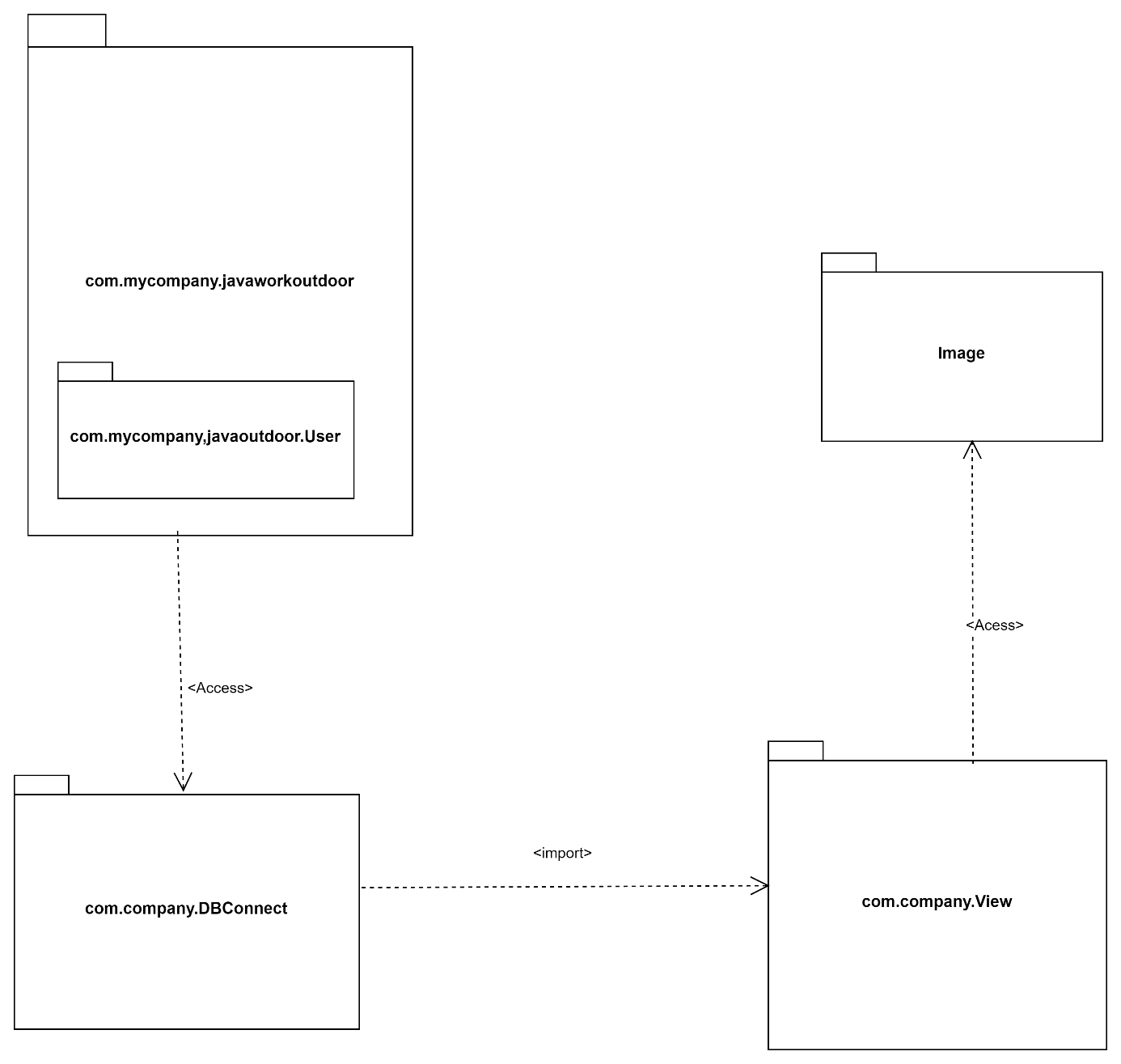
A class diagram is a static diagram. It represents the static view of an application. class diagram is not only used for visualizing, describing and documenting different aspect of the system but also for constructing executable code of the software application. Class diagram describes the attribute and operation of a class and also constraints imposed on the system. It purpose is to model the static view of an application.



*Figure 15: Class Diagram Of The OutDoor Gear Management App*

### 2.Package Diagram

A class diagram is a static diagram. It represents the static view of an application. class diagram is not only used for visualizing, describing and documenting different aspect of the system but also for constructing executable code of the software application. Class diagram describes the attribute and operation of a class and also constraints imposed on the system. It purpose is to model the static view of an application.

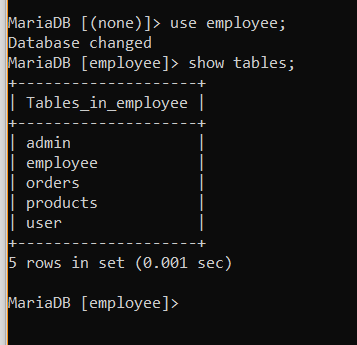


*Figure 16: Package Diagram Of The Out Door Gear Management App*

# Component Of The System

## Database:

* For this System the DBMS(Data Base Management System) use is MYSQL.
* The name of the database is **employee** it has 5 tables.



*Figure 17: Picture Of tables of the employees database*

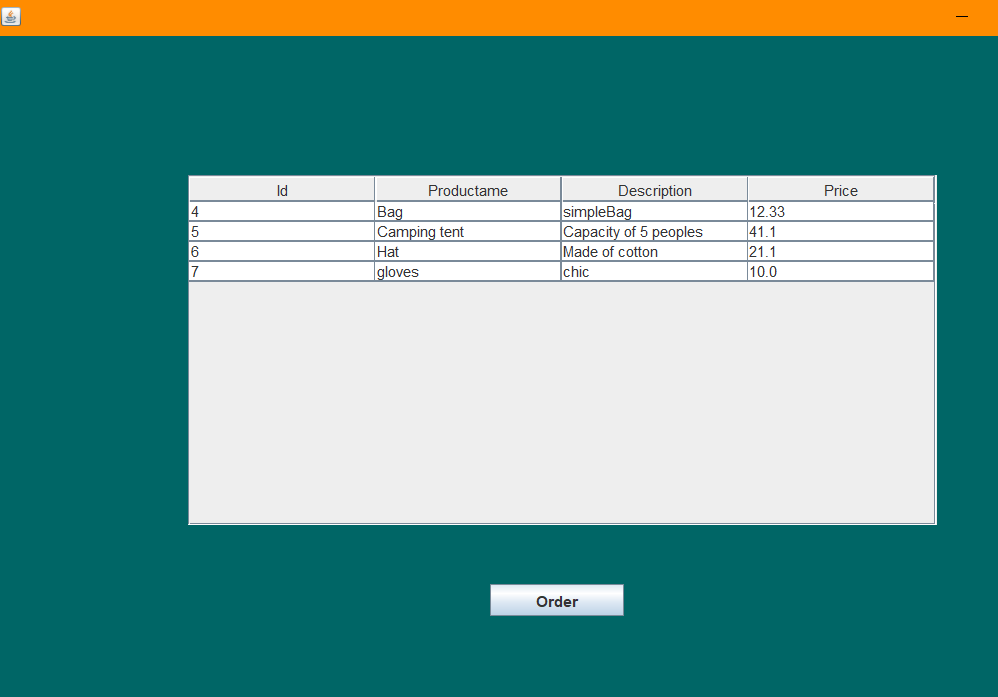
## IDE:

* Stands for Integrated development environment.
* The one use to implement this system was **Net bean**  version 20.

# USER GUIDE

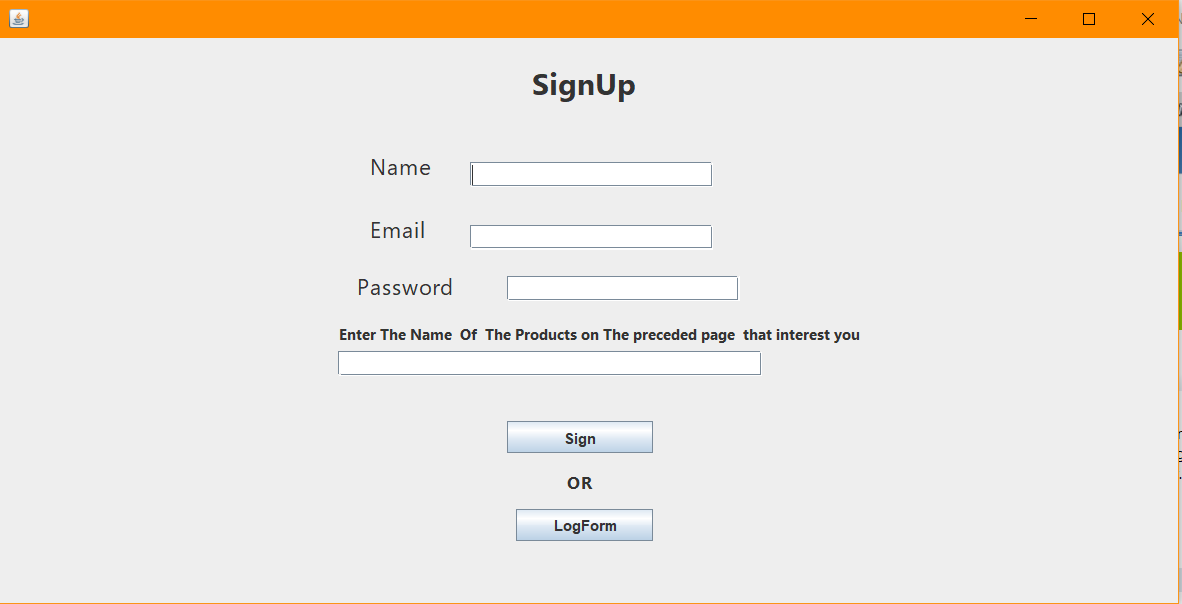
1.Access the Application

Run the program and a default home page will be display



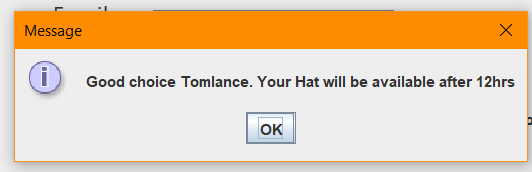
*Figure 18: Default Home page*

-Click on order button and signup page will be display



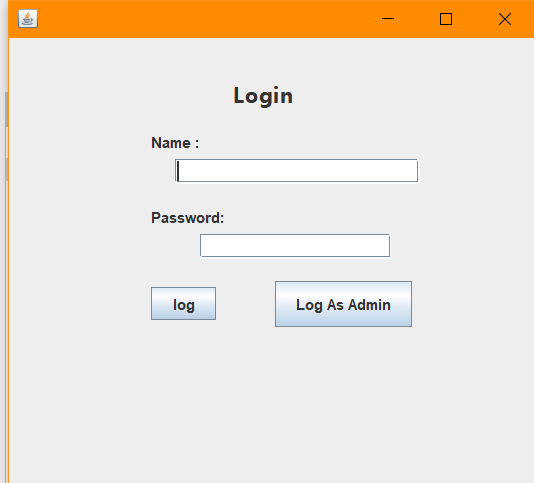
*Figure 19: SignUp page*

If you field up the different field according to their context a popup form will be return notifying you see image below



*Figure 20: JOption Pannel Validating the sign button functionality*

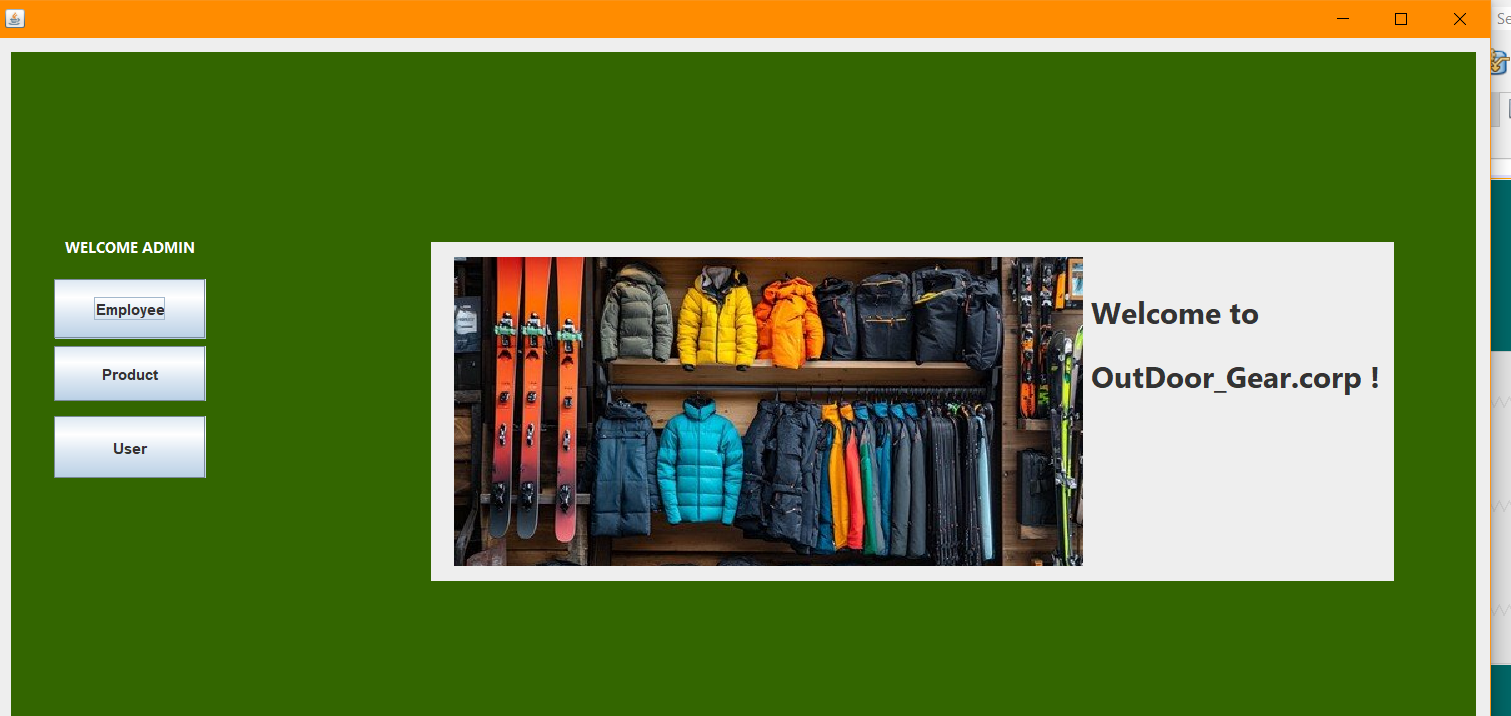
-But if you click on the **LogForm** button a form is return characterize by 2 textfields and 2 buttons having values **Log**  and  **Log As Admin** respectively as see below



*Figure 21: Login page*

**Log As Admin**

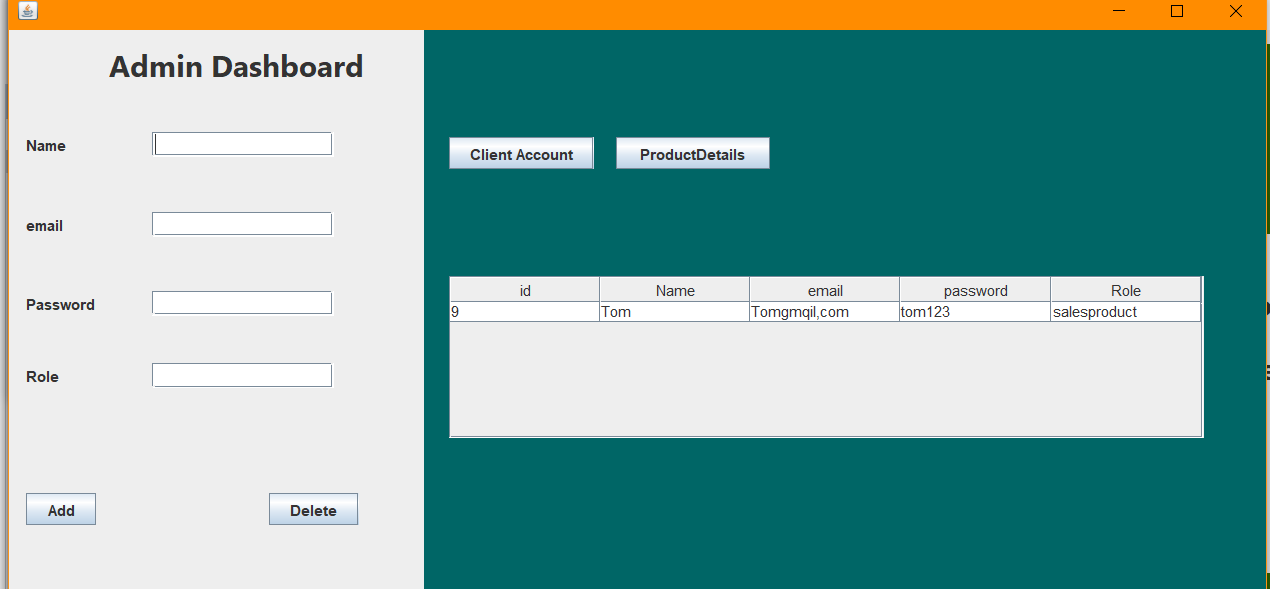
-To log as the Admin enter the following credentials: name “ **Jacques**”, password “**12345**”. And click on the button **Log As Admin.** The admin home page will be return.



*Figure 22: Admin Home page*

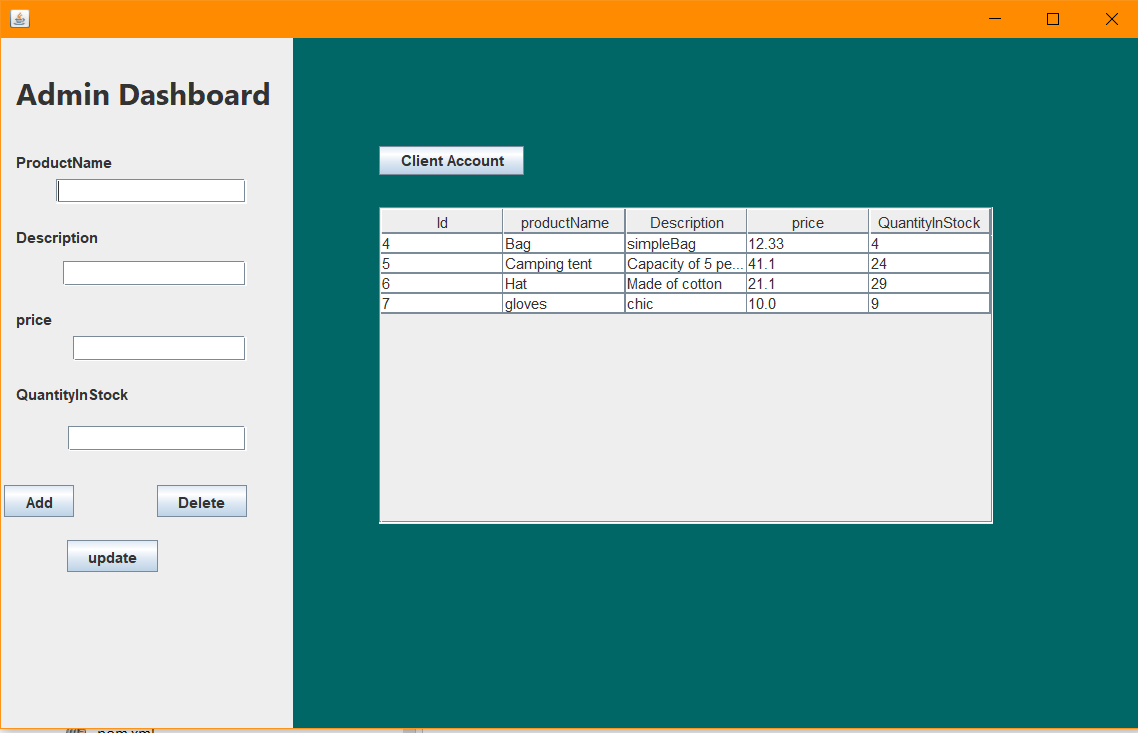
-From this page you can perform the following actions:

* Clicking on the **Employee** button a new page will be return where you will be able to add or delete your personnels.



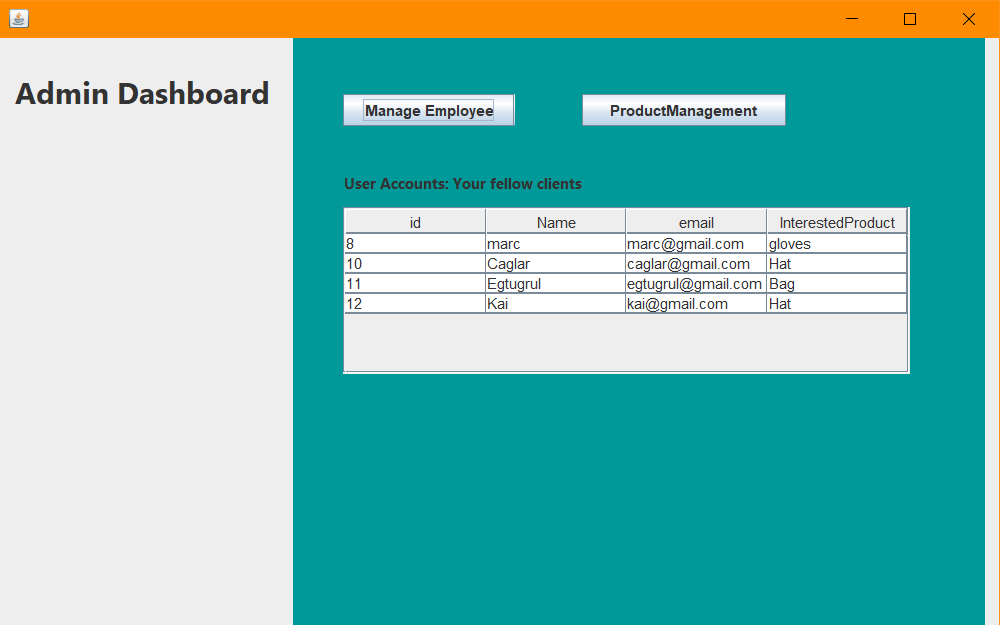
*Figure 23: Admin Dashboard employee management page*

* Clicking on the **Product** button on the welcome Admin page or the **ProductDetails** button on the dashboard page you will be return a page which offers product management functionality



*Figure 24: Product management page*

* Clicking on the **client Account**  button you will be directed to a new page that displays clients information



**Log Button**

If you provide the right credentials and click on the log button you will be return page that offers product management functionalities only.

