Cargo Project Final Documentation

June 06, 2021

- *Burak Bahir Günden* *#041701027*
- *Efe Ertekin* *#041701006*
- *Erdal Doğan* *#041701076*
- *Yasin Yılmaz* *#041701020*

Contents

1.Introduction

1.1 Goals and Objectives

This report summarizes the software requirements, planning, design, and implementation of a cargo tracking application. Such applications are adopted by many of the corporate logistics companies, if not all. Moreover, in the era of information, providing real-time information about the location and the situation of the packages have been an important business goal of the logistics firms. Competitive advantage of enabling users to access information about their packages, accurately estimating the arrival time, and early-detection of problems with opportunity to intervene cannot be ignored. Furthermore, having a detailed log of the company operations is very valuable for optimization purposes.

This project delivers a system where the sender and receiver can track the package during the logistics. Also, a user, regardless of being sender or receiver, can see the previous packages s/he transmitted in detail.

With our system, the business owner will have access to detailed logs about the operations of her business. This information will enable her to observe performance of certain aspects of the business, identify bottlenecks in the pipeline and optimize the procedure further.

1.2 Statement of Scope

Main users of the software can be divided into three categories; senders, receivers and the employees of the firm. Upon handing their packages to the cargo branch, the package is matched with the senders and receiver accounts. When they login to their accounts, they can see details about the situation and the location of the package.

Admins, employees of the cargo branches, update the status of the packages via their desktop applications. When a package reaches any facilities of the logistics company, such as distribution center, other branch, etc. that information is logged to the system and visible to sender and receiver.

1.3 Software Context

In this project, two applications have been developed. An admin panel for authorized employees to insert, view, update, and delete the package information, along with web-based application for public access to view package itinerary. Web-based application, as the name suggests, is easily accessible from any modern internet browser. Main goal of this particular application is to have it easily accessible for a wide range of users without any overhead, such as software installation, etc. On the other hand, a desktop application written with C# has been developed since there are a limited number of locations which will need to run the application. Other consideration of the admin panel was of course the security.

1.4 Major Constraints

Tracking application should be available on a wide range of devices, operating systems and platforms. We therefore decided to develop this application as a web app. Consequently, it will run on modern browsers without having to install anything.

Another major consideration was the security and access restrictions to the admin panel. To meet this criteria, we made this application only available on local computers located in the branches and distribution centers.

2.Usage Scenario

2.1 User Profiles

In the following table, actors in these systems are shown and described.

Customer	Customers can be the sender or receiver for a cargo. They can see their current cargo's status and check their old cargos.
Branch Employee	Branch employees can enter the shipment information to the database, update cargo location, and cargo's status.
Courier	Couriers can read the information of the shipment to be able to deliver the cargo by seeing the address of the receiver and update the cargo's status.

2.2 Use-Cases

In this project, there are 2 different systems. Web based side of the project is dedicated to customers' use, while the other, which is Windows Form Application, is dedicated to employees of the company. Each use-case will be detailed in 2.2.2.

Web-Side

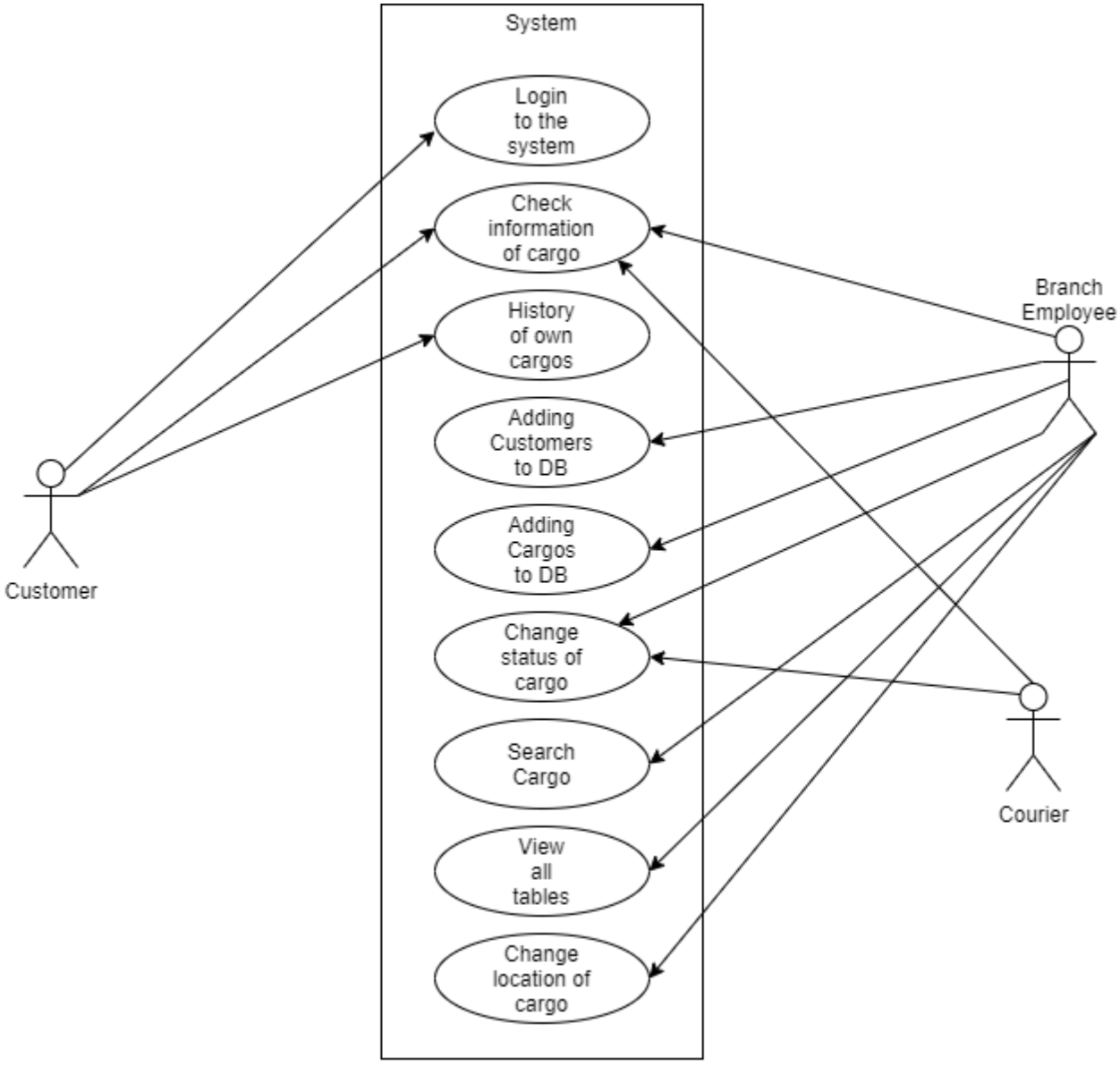
1. Log onto the web system.
2. Check the information of the shipment.
3. Check old sended or received cargos' information.

Form Application

1. Adding customers to the database.
2. Adding cargos to the database.
3. Changing the status of the cargo.

4. Changing the location of the cargo.
5. Searching for cargo.
6. Check customers' history.
7. View all tables in the database.

2.2.1 Use-Case Diagram



Figure#1 Use-Case Diagram

2.2.2 Use-Case Descriptions

Use-case:	Log onto the web system.
Primary Actor:	Customer
Goal:	Login to the system.
Scenario:	1.Open web page 2.Click login button 3.Fill username and password 4.Press sign-in
Expectations:	Customer is on the system.

Use-case:	Check the information of the shipment.
Primary Actor:	Customer, Branch Employee, Courier
Goal:	Check the information of the shipment.
Scenario:	1.Open web page 2.Search Cargo 3.Give cargo's ID 4.Press Enter
Expectations:	Customer checks his/her shipment information.

Use-case:	Check old sended or received cargos' information.
Primary Actor:	Customer, Branch Employee
Goal:	Seeing old cargos.
Scenario:	1.Open web page 2.Click login button 3.Fill username and password 4.Press sign-in 5.Press Account 6.Press Cargos
Expectations:	Customer is on the system.

Use-case:	Adding customers to the database.
Primary Actor:	Branch Employee
Goal:	Customer addition to the database.
Scenario:	1.Open form application. 2.Click Add Customer. 3.Fill the Name space. 4.Fill the Address space. 5.Fill the Mobile No space. 6.Fill the E-Mail space. 7.Choose Customer Type. 8.Click Add Customer.
Expectations:	Customer is added to the database.

Use-case:	Adding cargos to the database.
Primary Actor:	Branch Employee
Goal:	Cargo addition to the database.
Scenario:	1.Open form application. 2.Click Add Cargo. 3.Fill the Sender ID. 4.Fill the Receiver Address. 5.Select the Branch ID. 6.Select the Deliverer ID. 7.Select the Branch Employee ID. 8.Fill the Receiver ID. 9.Choose the Type. 10.Select the Payment_Type. 11.Fill the Cost. 12.Fill the Date Delivered. 13.Fill the Estimated Date. 14.Fill the Description. 15.Fill the Weight. 16.Click Add Cargo button.
Expectations:	Cargo is added to the database.

Use-case:	Changing the status of the cargo.
Primary Actor:	Branch Employee, Courier
Goal:	Change the “Delivered” status
Scenario:	1.Open form application. 2.Click Deliver Cargo. 3.Look at the Update Status part. 4.Fill the cargo ID 5.Select the new cargo status 6.Fill the date 7.Click the enter button
Expectations:	Cargo status is updated.

Use-case:	Changing the location of the cargo.
Primary Actor:	Branch Employee, Courier
Goal:	Login to the system.
Scenario:	1.Open form application. 2.Click Deliver Cargo. 3.Look at the Update Location. 4.Fill Cargo ID. 5.Select New Branch ID. 6.Fill the Date. 7.Click to the Enter.
Expectations:	Cargo’s location is updated.

Use-case:	Searching for cargo.
Primary Actor:	Branch Employee
Goal:	Cargo searching for any operation.
Scenario:	1.Open form application. 2.Click Search Cargo 3.Select/Fill the Cargo ID. 4.Click Enter.
Expectations:	Desired cargo was searched by an employee.

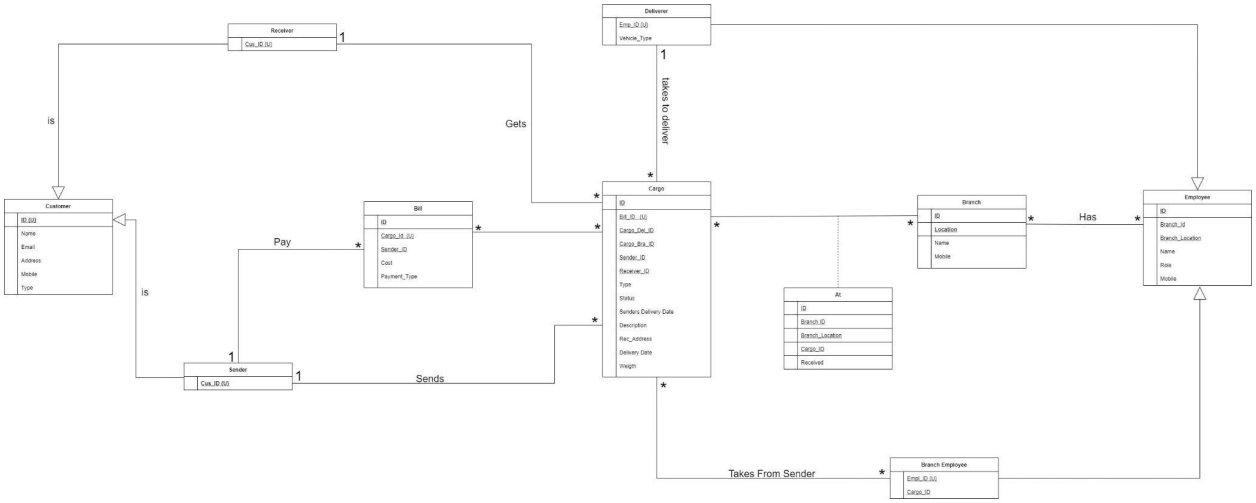
Use-case:	Check customers' history.
Primary Actor:	Branch Employee
Goal:	Customer's history will be shown.
Scenario:	1.Open form application. 2.Click Customer History. 3.Fill Customer ID. 4.Choose the Type. 5.Click Enter.
Expectations:	Customer's history is on the screen.

Use-case:	View all tables in the database.
Primary Actor:	Branch Employee
Goal:	All tables can be shown.
Scenario:	1.Open form application. 2.Click View Tables. 3.Choose desired table.
Expectations:	Desired table is shown.

3. 4+1 View

3.1 Logical View

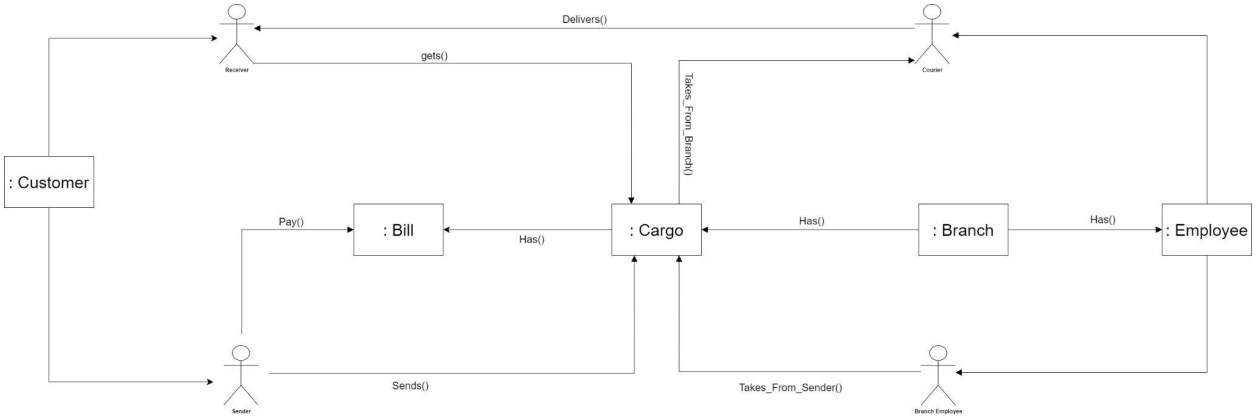
Class Diagram



High Resolution

Figure#2 Class Diagram

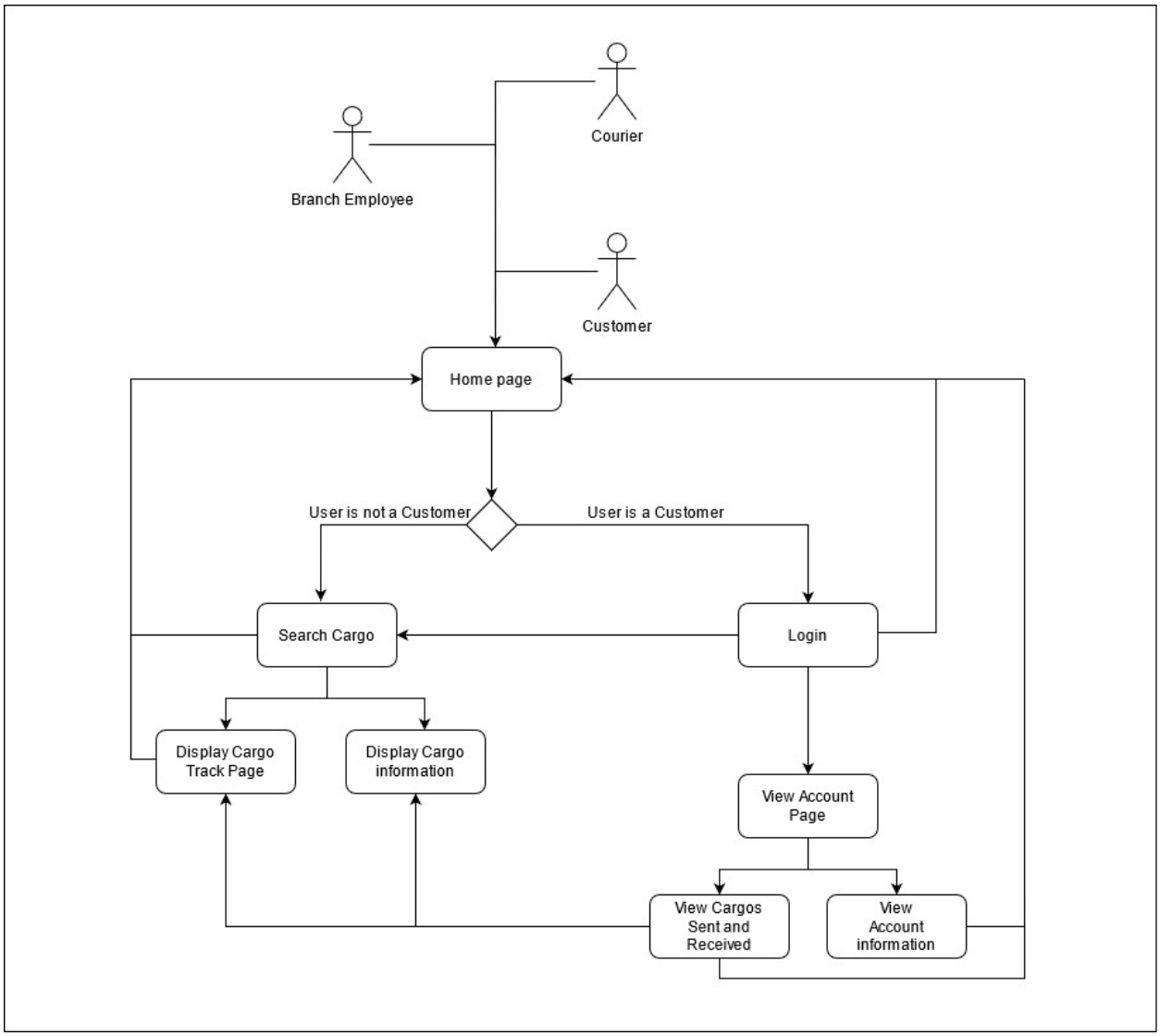
Communication Diagram



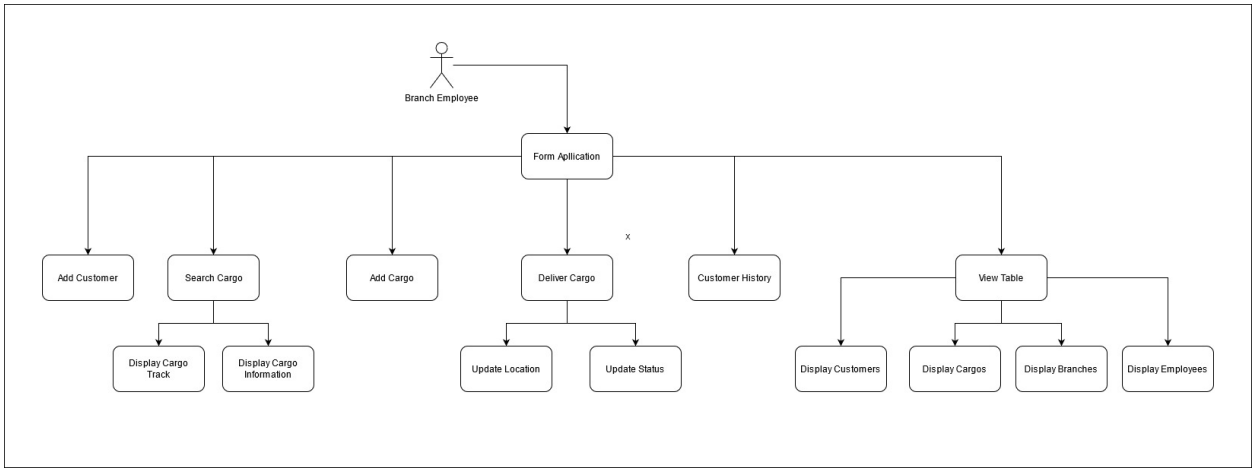
High Resolution

Figure#3 Communication Diagram

3.2 Process View



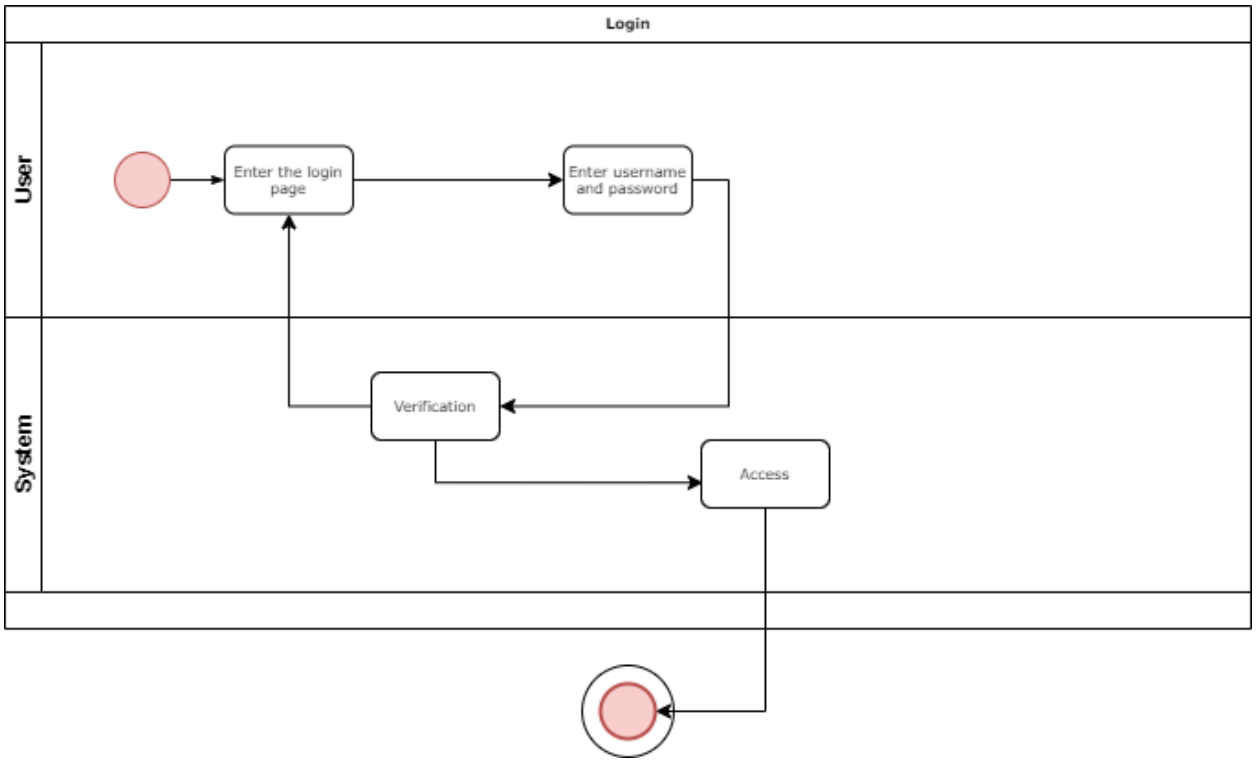
Figure#4 Process View for Web application



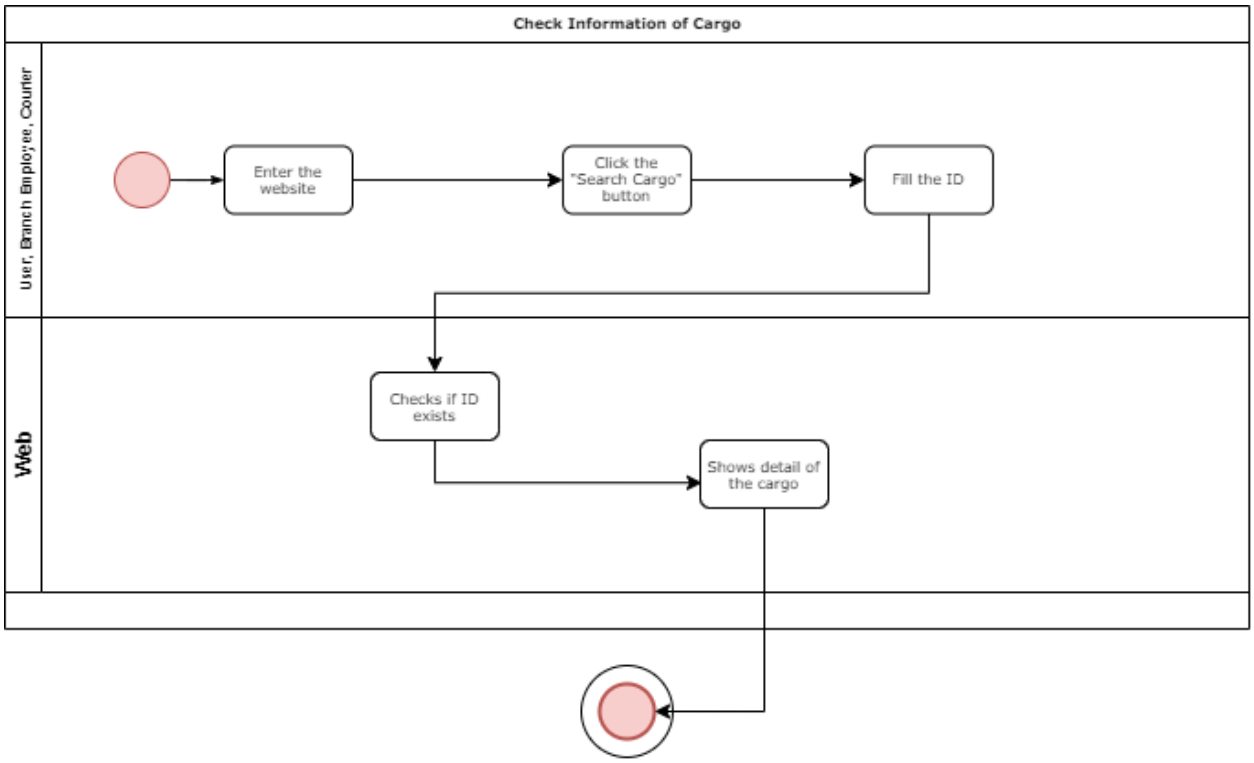
Click for better resolution

Figure#5 Process View for Desktop Application

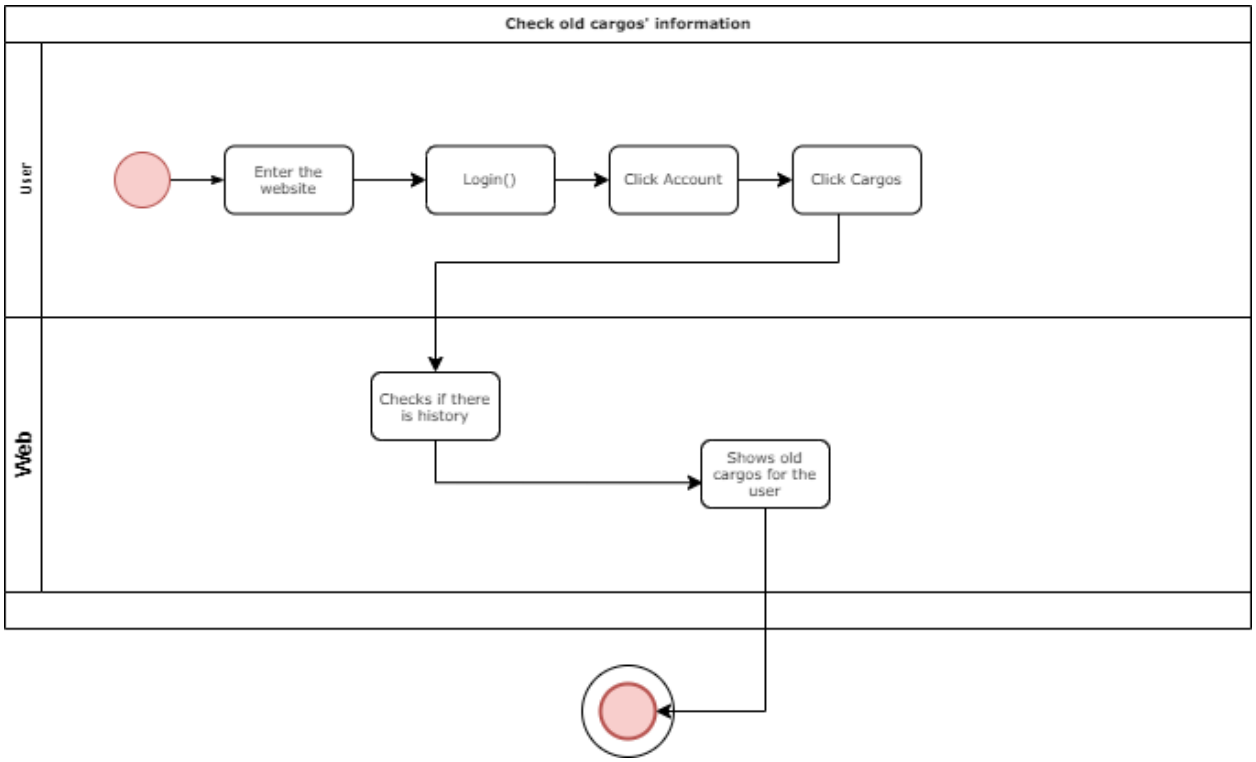
Activity and Sequence Diagrams



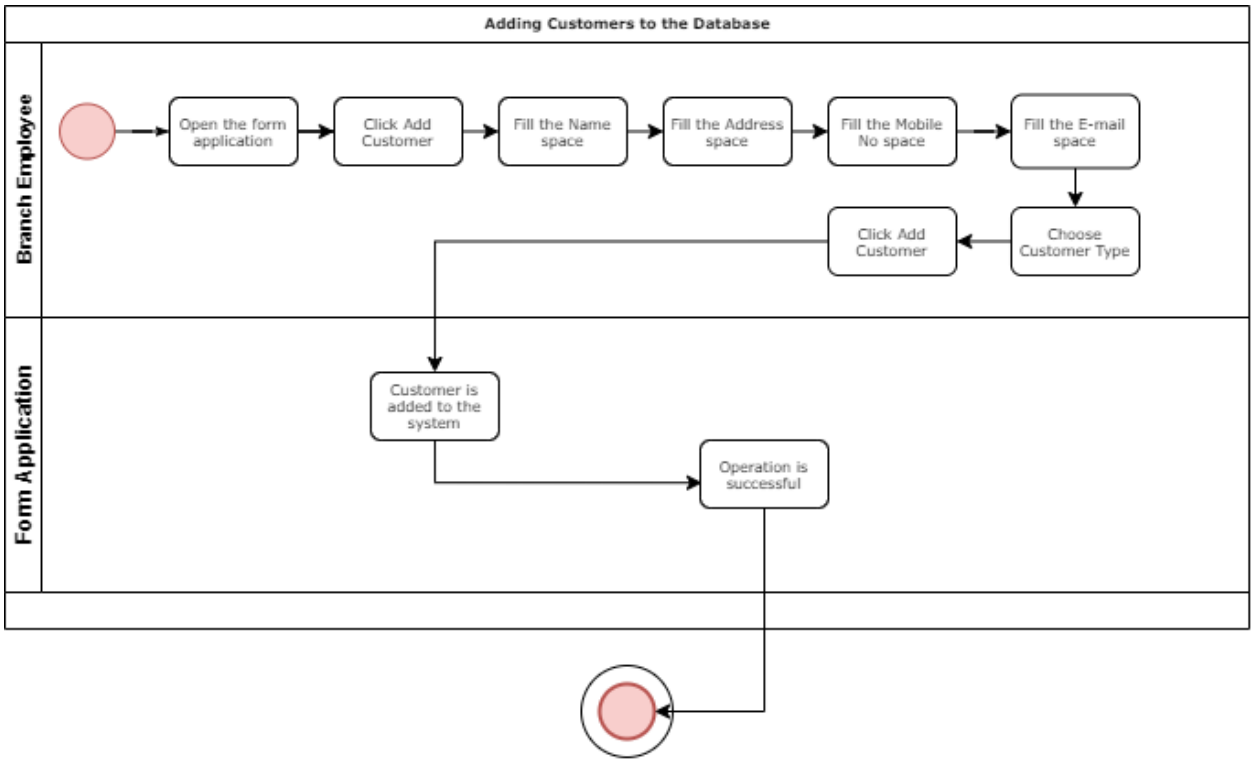
Figure#6 Login activity diagram



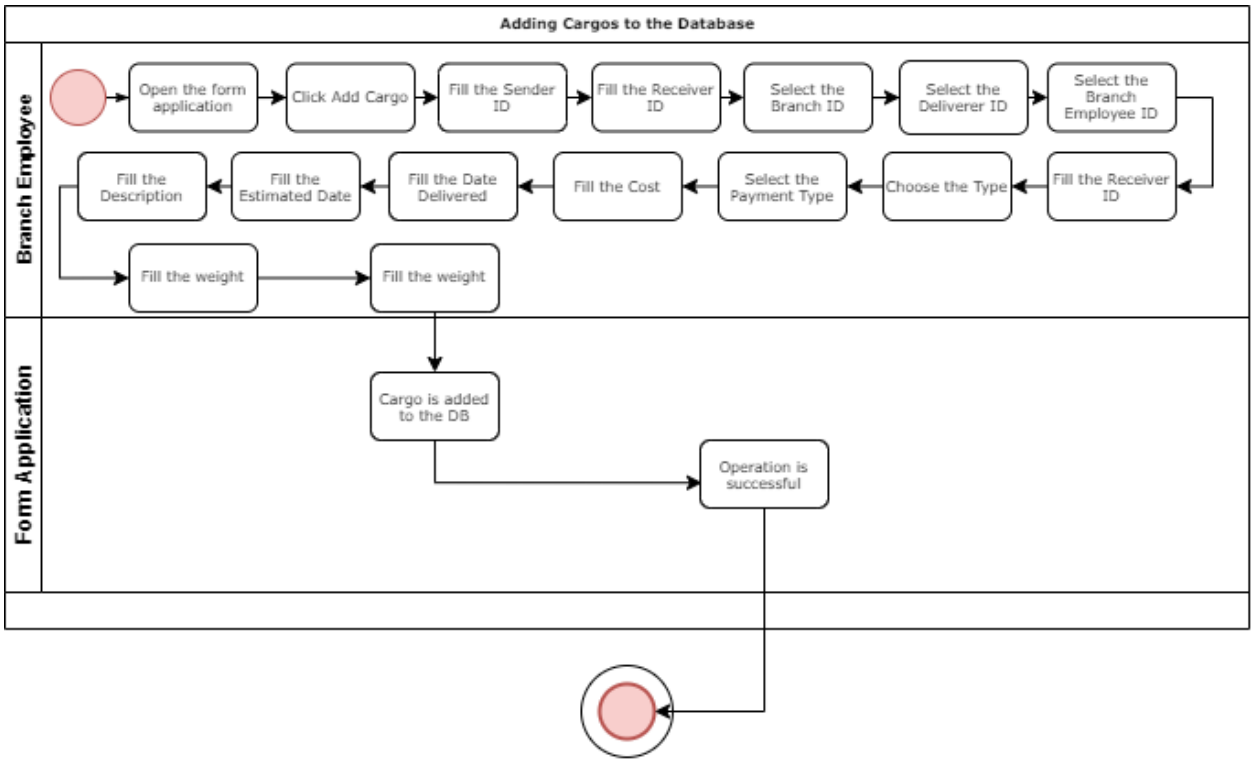
Figure#7 Check information of cargo activity diagram



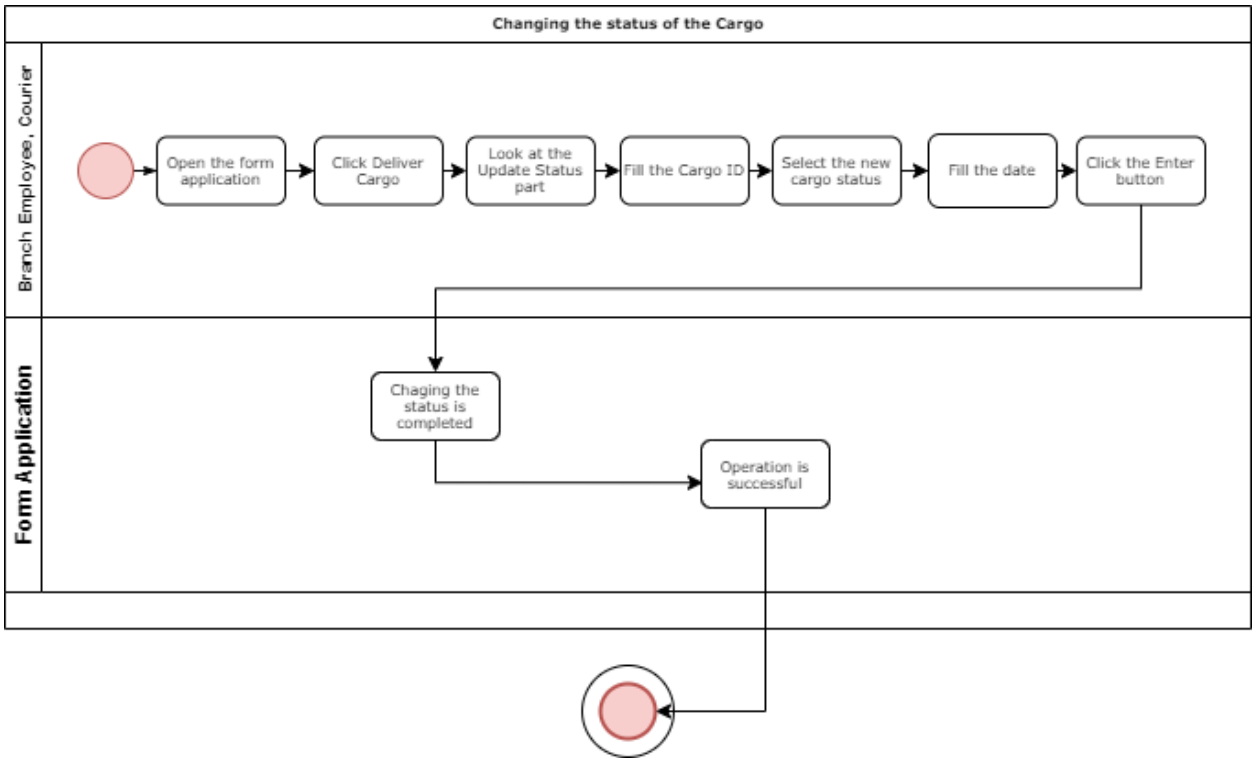
Figure#8 Check old cargos' information activity diagram



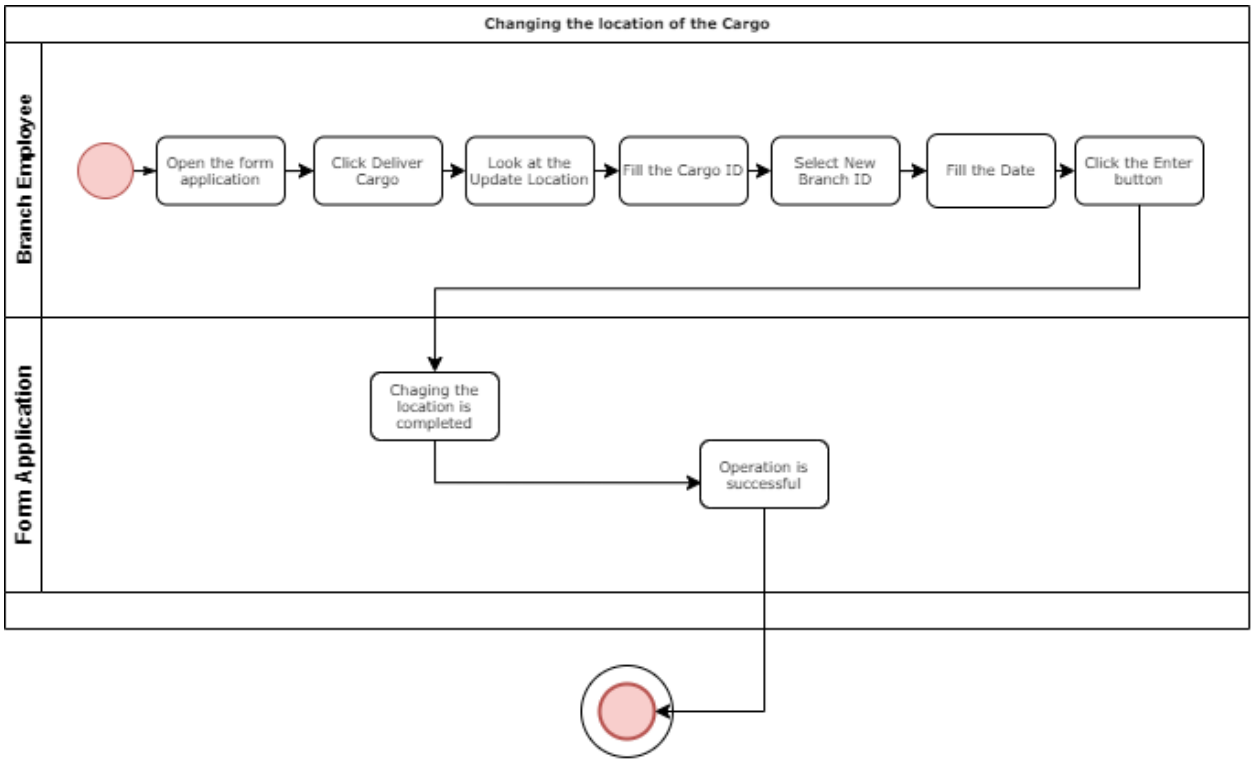
Figure#9 Adding customers to the database activity diagram



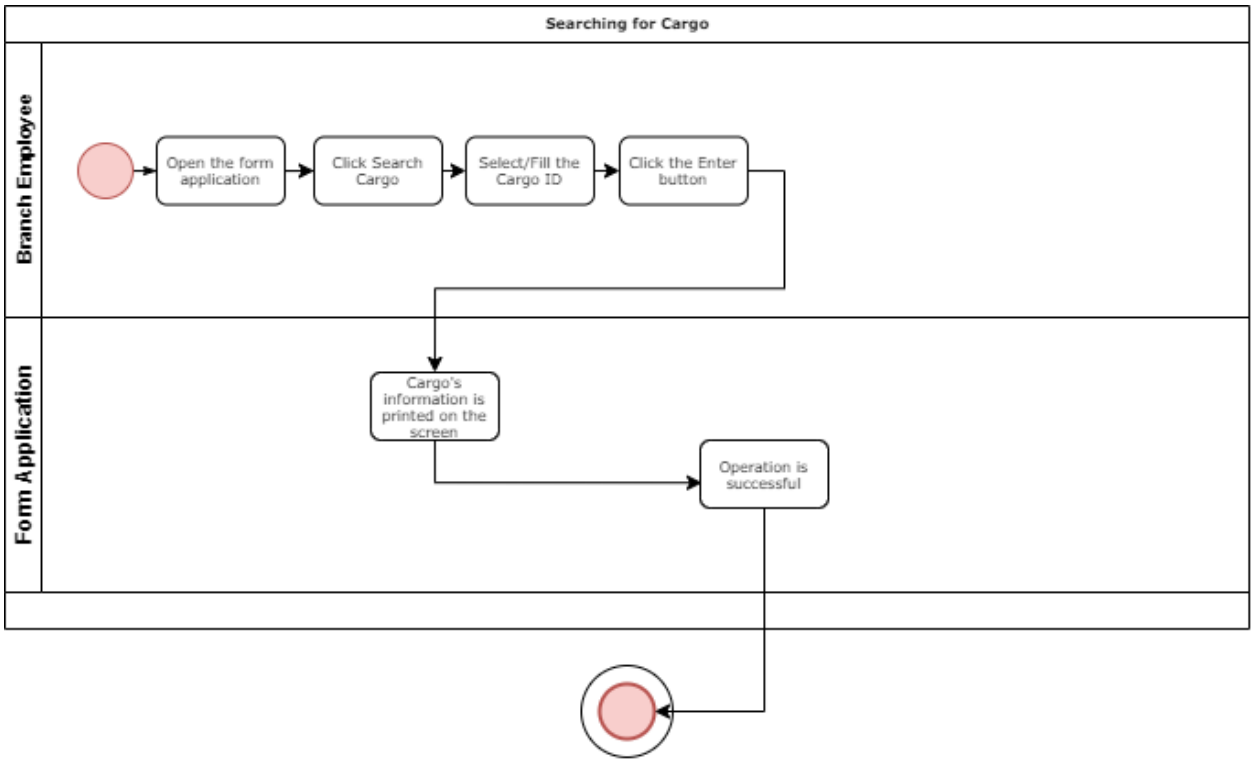
Figure#10 Adding cargos to the database activity diagram



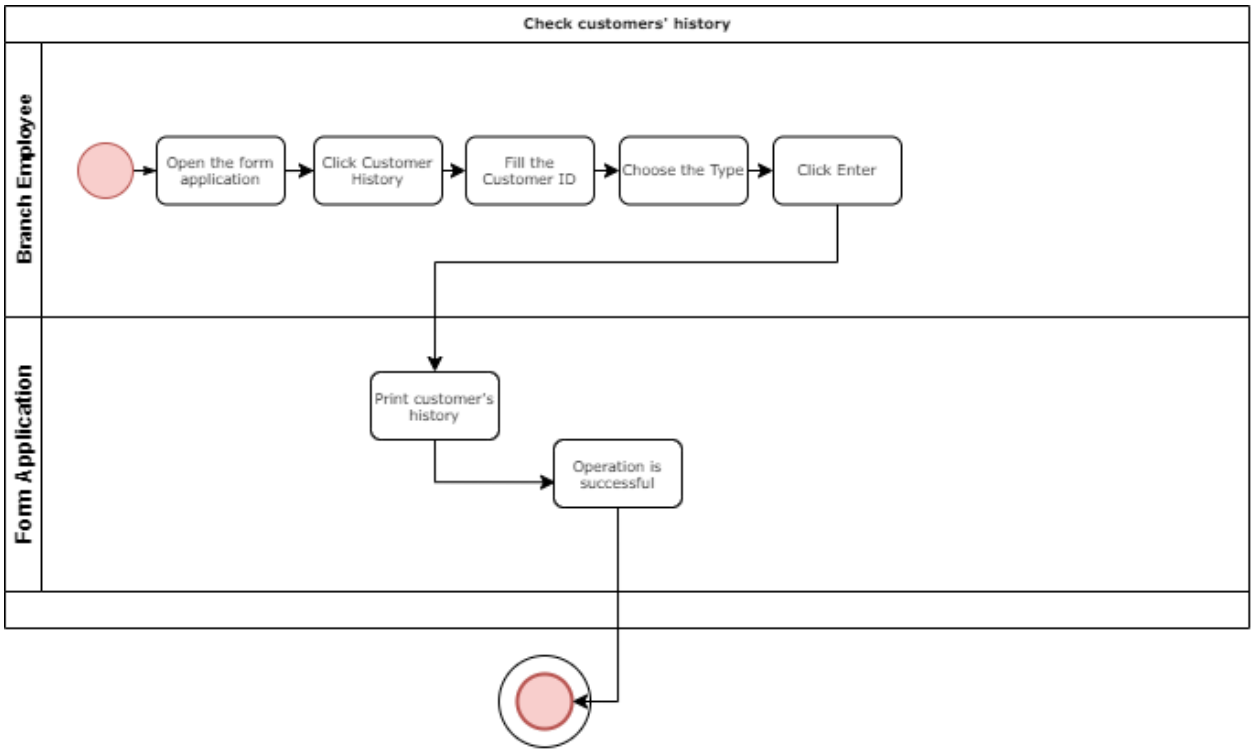
Figure#11 Changing the status of the cargo activity diagram



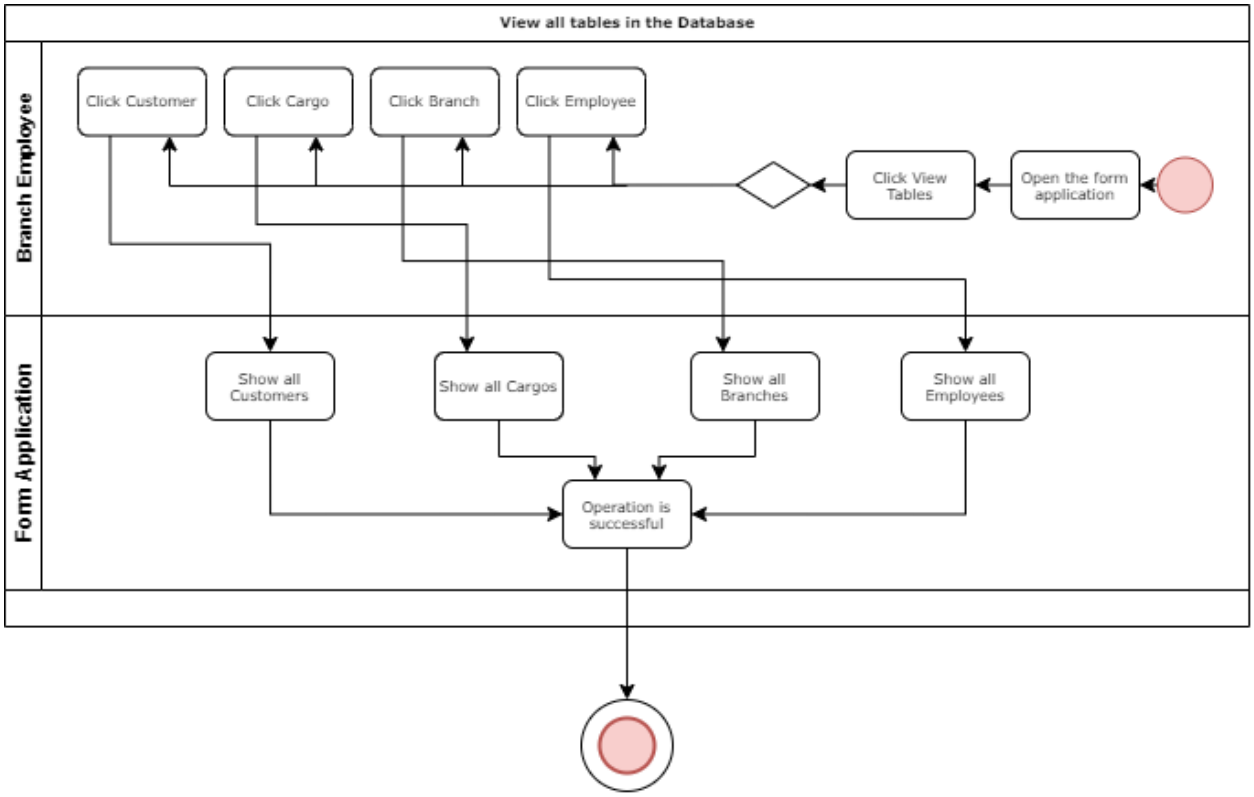
Figure#12 Changing the location of the cargo activity diagram



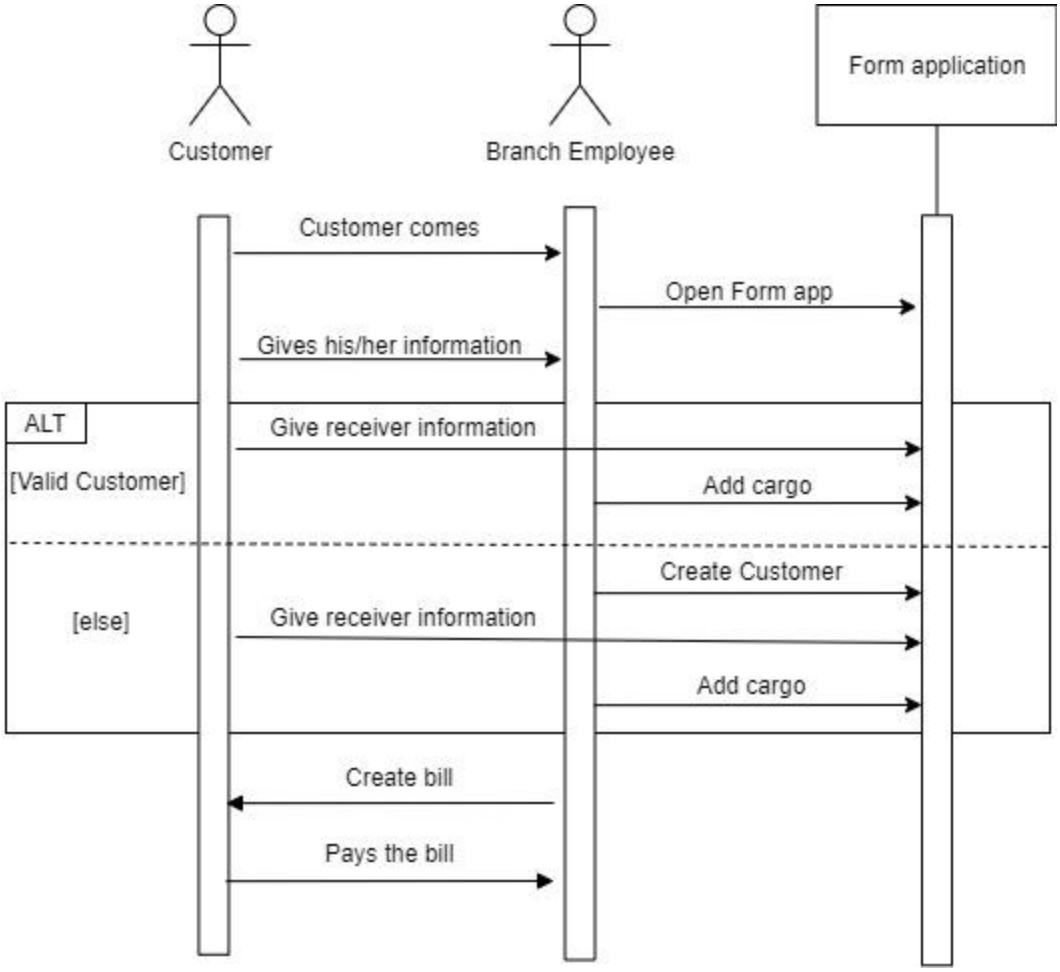
Figure#13 Searching for cargo activity diagram



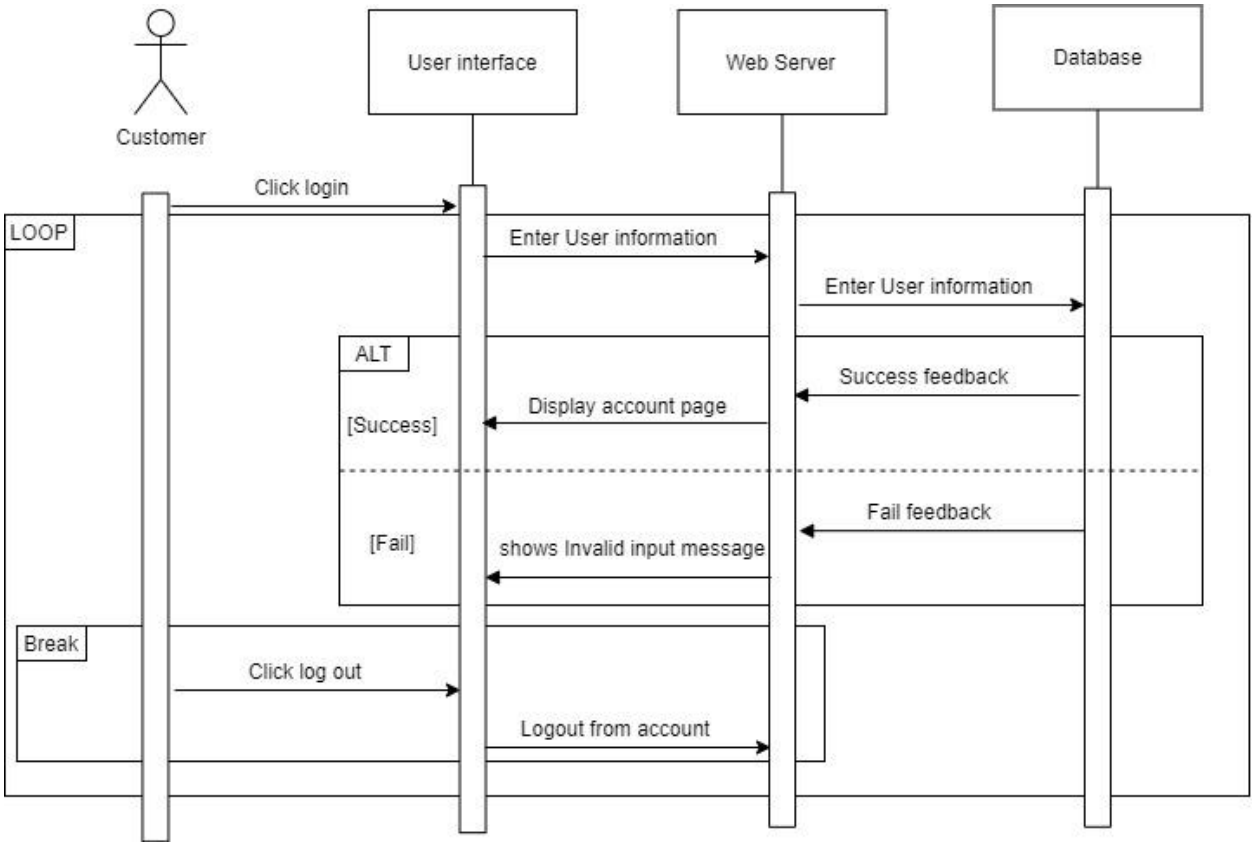
Figure#14 Check customers' history activity diagram



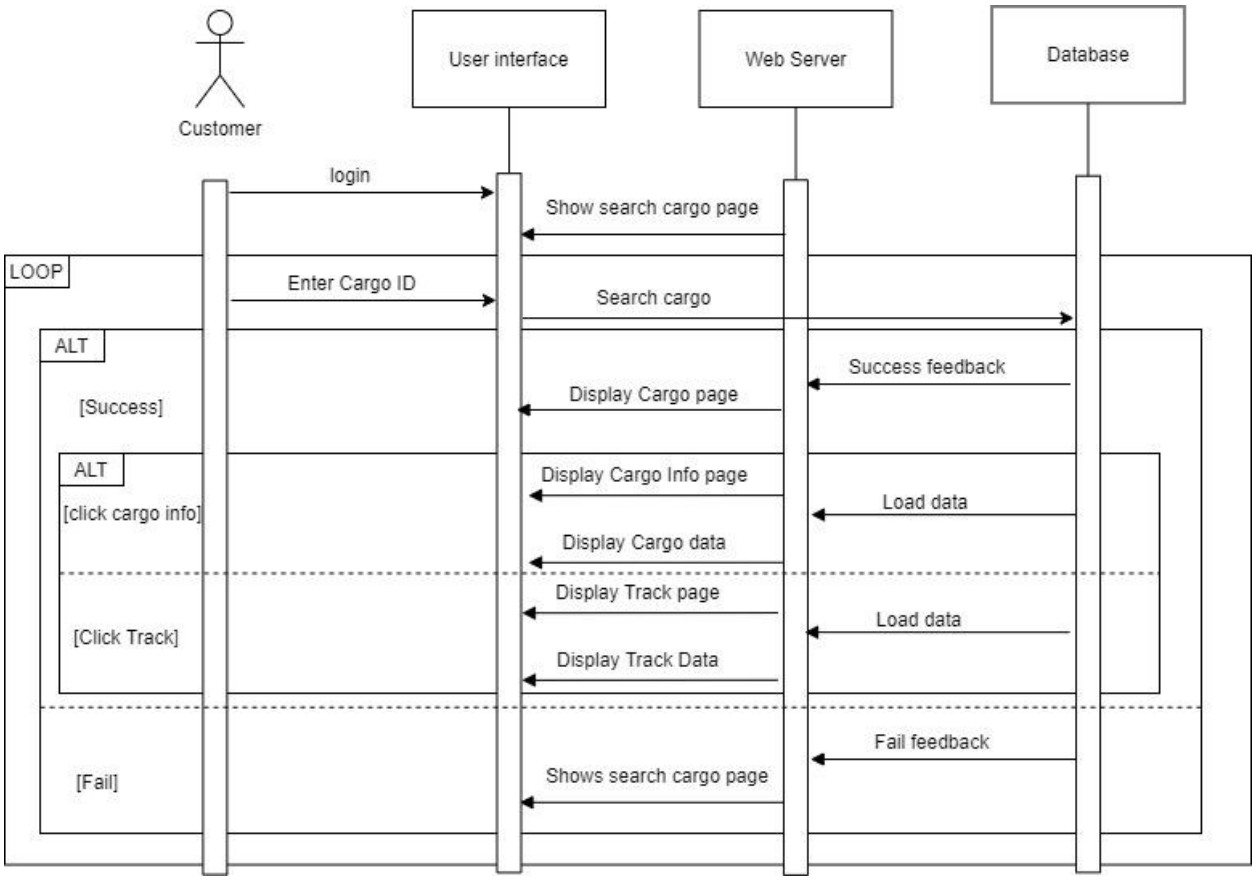
Figure#15 View all tables in the database activity diagram



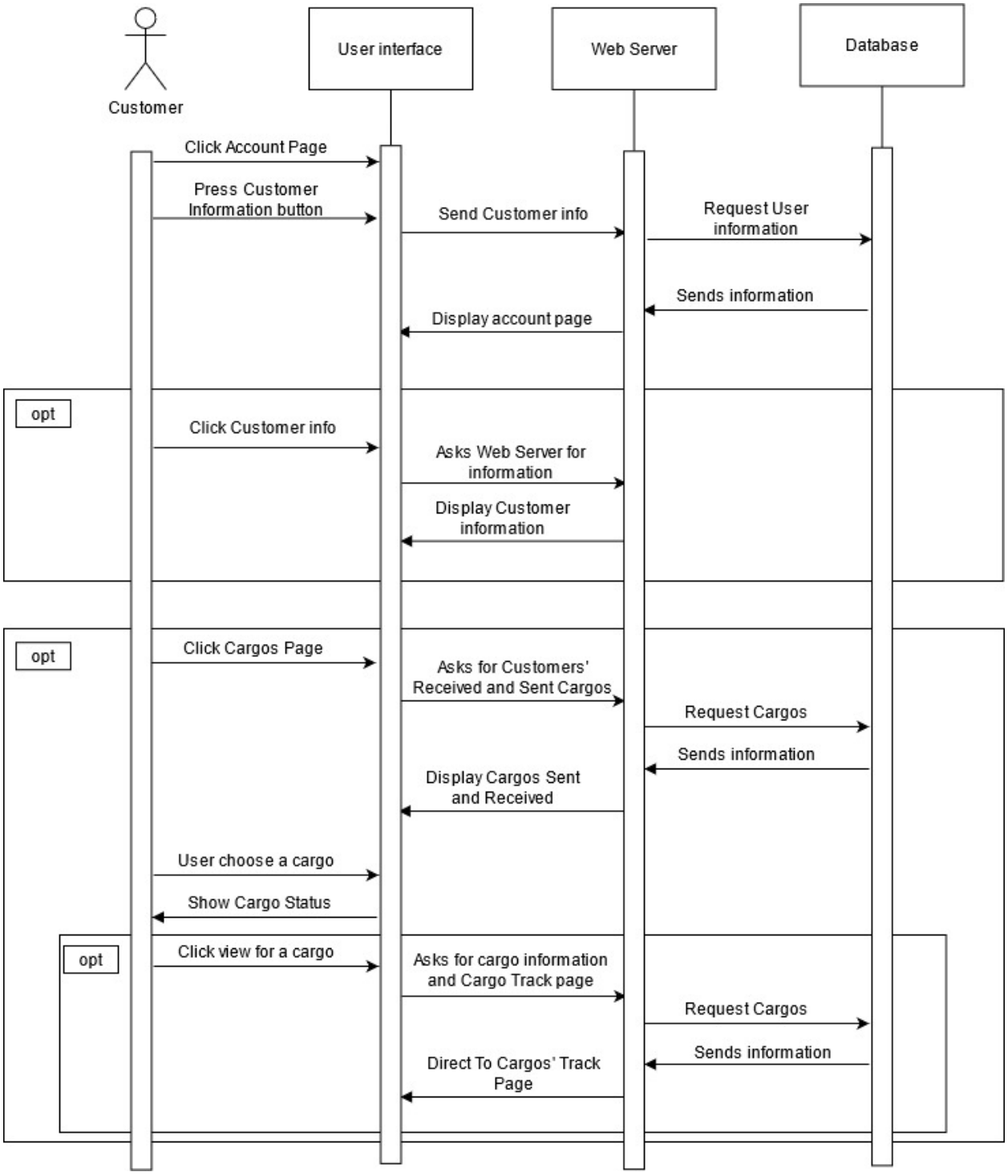
Figure#16 Sequence diagram of Add Cargo



Figure#17 Sequence diagram of login

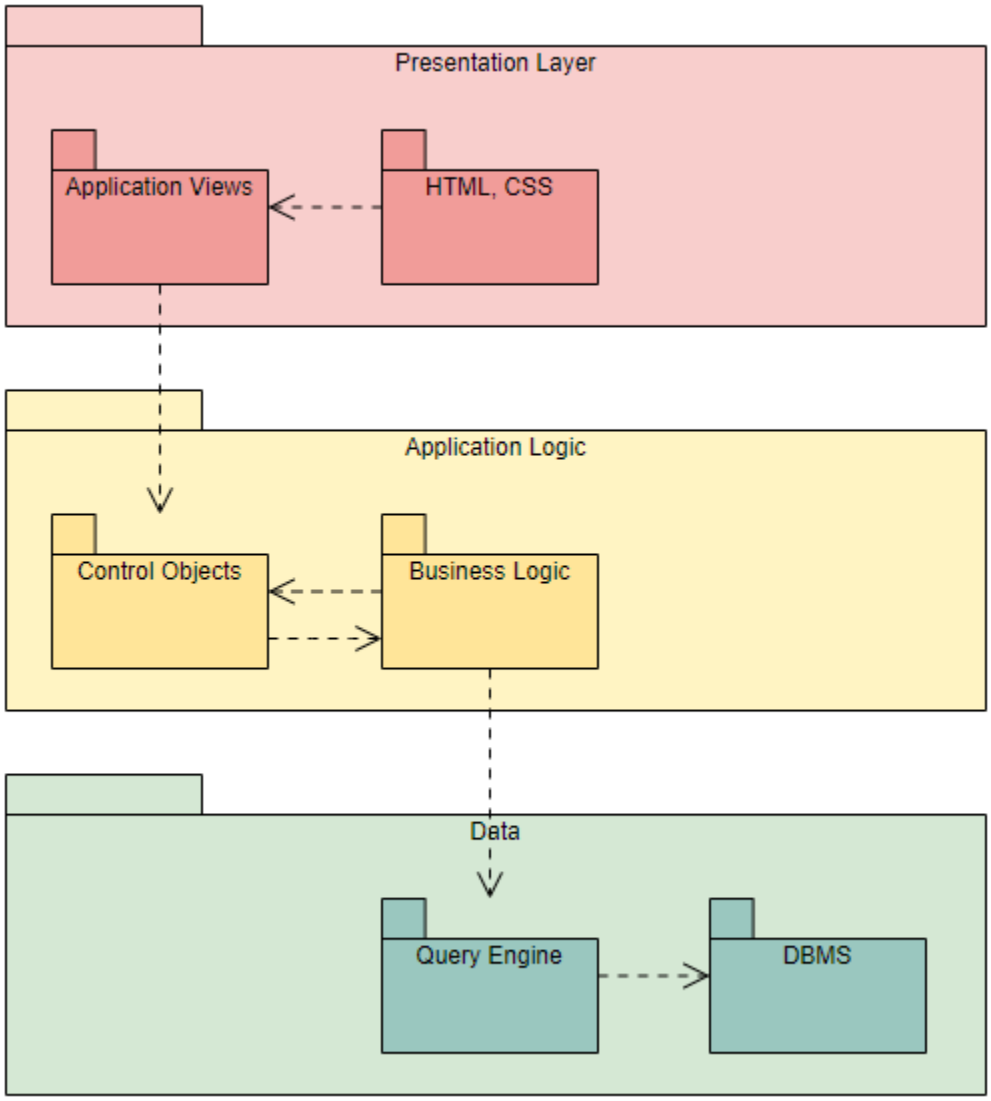


Figure#18 Sequence diagram of Search Cargo

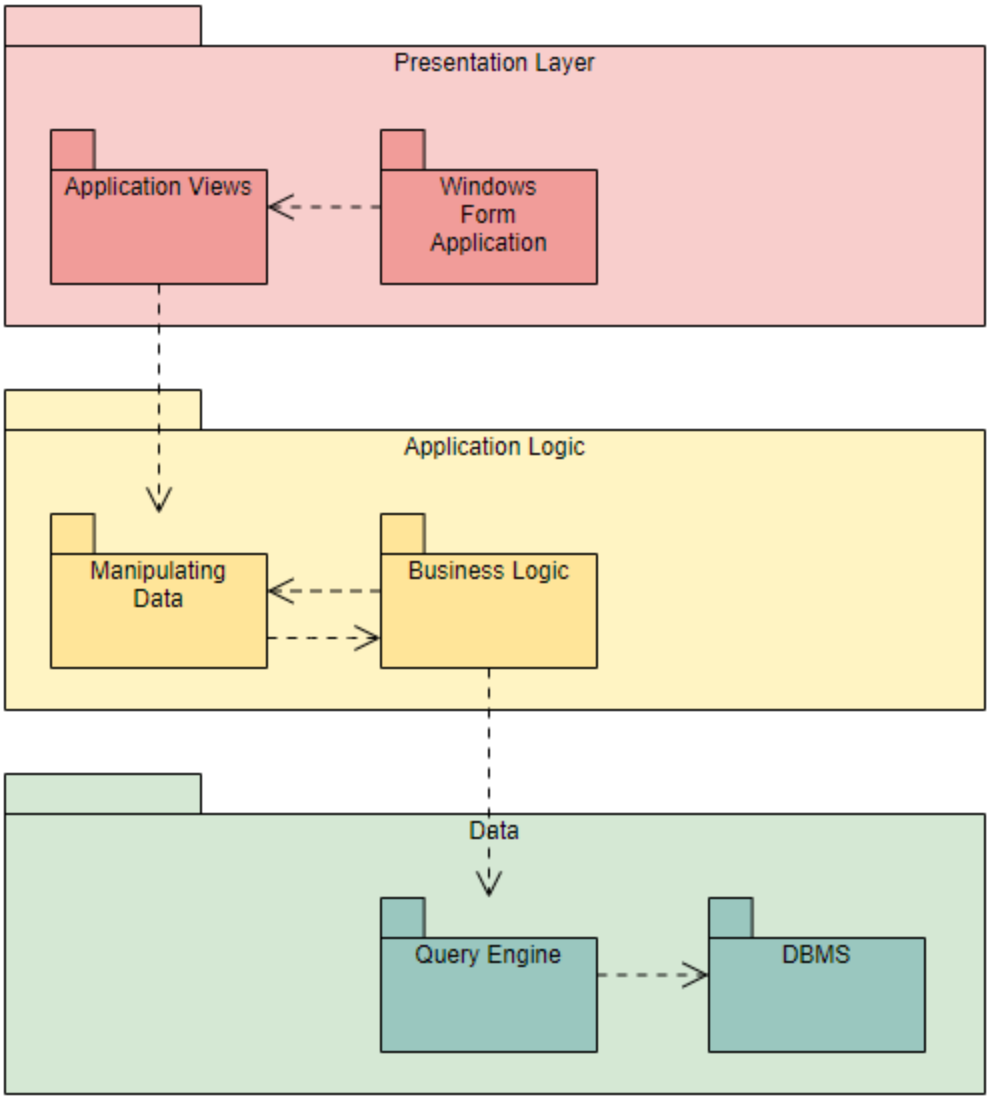


Figure#19 Sequence diagram of Account Page

3.3 Development View

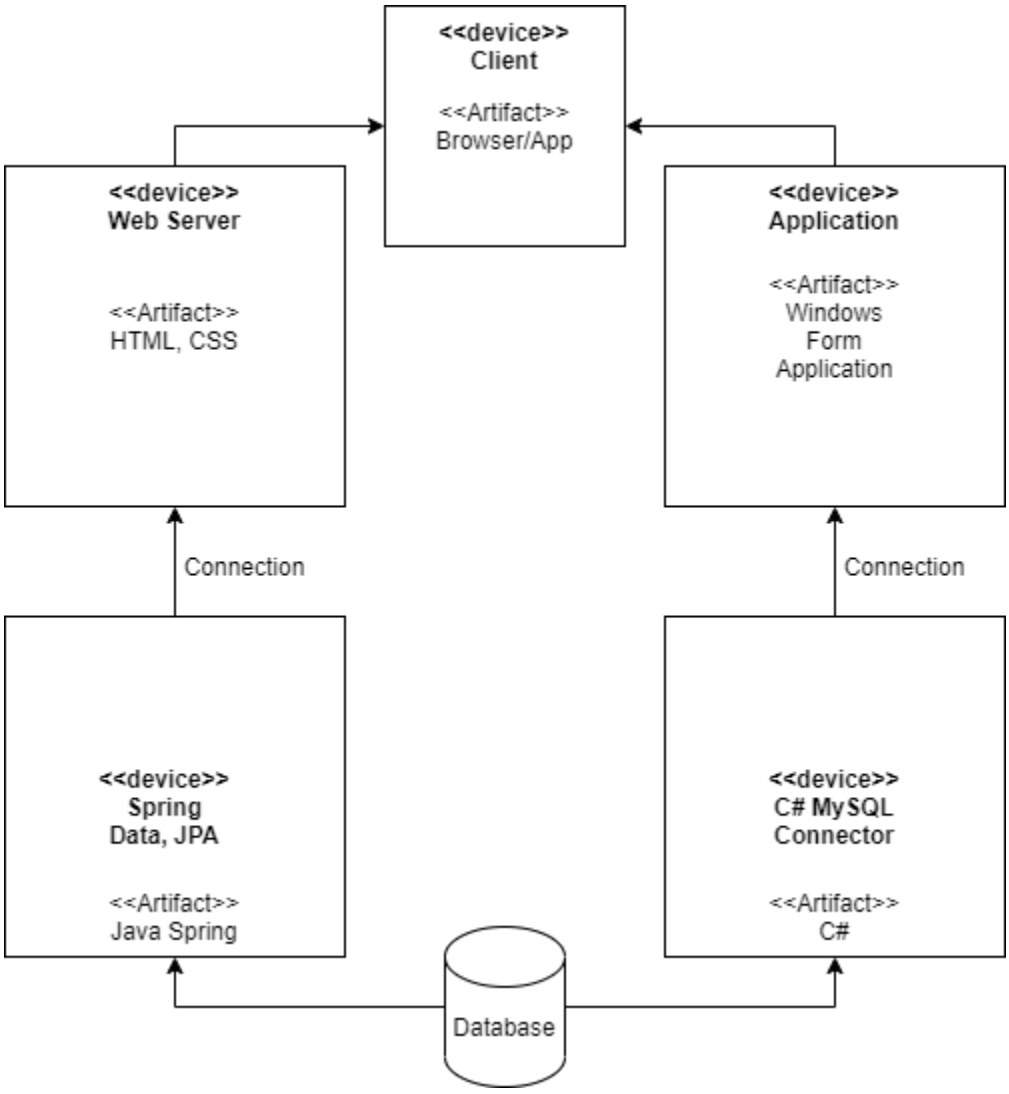


Figure#20 Web Package Diagram



Figure#21 Windows Form Application Package Diagram

3.4 Physical View



Figure#22 Physical View

4. Restrictions, Limitations, and Constraints

All non-functional requirements

- For the web, the back-end should be written with Spring Boot (Java).
- For the web, the front-end should be written with HTML, CSS, Js, and Bootstrap4.
- For the web, the API should be written with Spring Boot (Java).
- For the form app, DB connections should be written with C#.
- For the database, MySQL is the database management system.
- Spring Security for securing the user's data and login operation.
- Privacy of the customers' information must be ensured.
- Web and Form applications must work together without any problem.
- The website should be reachable from any device that is connected to the internet.

5. User Manual

In this section manuals for running both web and desktop apps are given.

To run the web project you can execute the jar file located in the target directory or if there is an IDE installed in the computer used, CargoApplication can be run via the ide. If there is no jar file you can generate it by running the maven install command. If you don't have maven installed in your machine you can just run ./mvnw install command but before running both commands, make sure that you are in the directory which has the pom.xml file. After running the application the application will be running on http://localhost:8080/

To run the desktop app there is a .exe file located in app/obj/Debug. By running this .exe file the application can be used.

Source codes of both applications are given below. For more, You can check it out via Github links.

6. Source Codes

Project's source codes are below here.

https://github.com/eertekin99/SE_Project