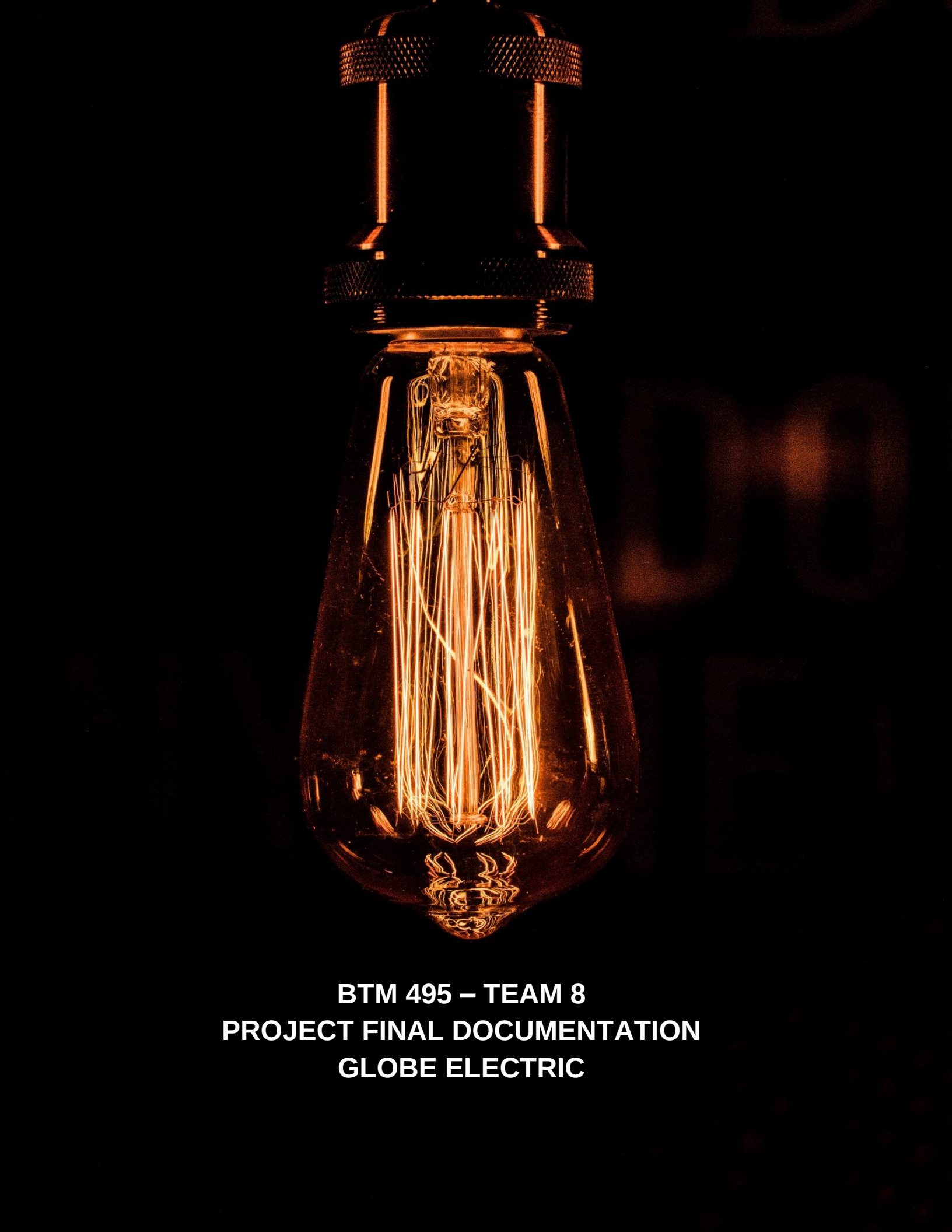


BTM 495 – TEAM 8
PROJECT FINAL DOCUMENTATION
GLOBE ELECTRIC

Globe Electric Development Report

Presented to Professor: Hossein Azarpanah
Information Systems Design and Implementation
BTM 495 Section AA
Team 8

John Molson School of Business
Concordia University

Submitted by:
Team - 8
Sima Sukkar - 40090653
Alex Juneau - 40085739
Luis Aguilar - 40045618
Abigail Estrada - 40077578
Amritjot Singh Nagi - 40037989
Steven Robinson - 40093733
Fabrizio Capparelli - 40097997

April 21st, 2021

Executive Summary

Globe Electric is a lighting and electric products manufacturer and wholesaler, founded in Montreal in 1932. From manufacturing light bulbs and other simple appliances, Globe Electric is now at the forefront of smart home technology. Describing themselves as the "creative energy company", Globe prides themselves on developing and designing innovative new lighting products that delight consumers with their ease of use and connectivity features. Now, Globe Electric products are sold at major national retailers such as Costco and Best Buy as well as the e-commerce giant, Amazon.

However, the advent of digital commerce did bring along additional challenges that are difficult to deal with. Having to manage a vast network of retail clients, all with unique contracts and terms leads to a high number of deductions. A deduction can occur for multiple reasons, such as when the client either enters a promotional period where they are selling their products for a discount, or if the client believes Globe has violated part of the terms in their agreement. While some deductions are mutually agreed upon, most of them come as a surprise to Globe and require further investigation. Such an example of these are short shipments, where the client claims that Globe did not send them the right quantity of a certain product. Globe Electric estimates that 1000 of these deductions are made every month and can range in costs from a few hundred to thousands of dollars of lost revenues.

After completing an analysis of the claims system, we concluded that 3 points in this process need urgent improvement. The submission of new claims into the system, the validation of a claim as well as the initiation of a dispute. The current claims system is hindered with miscommunication and information silos which prevent Globe from managing these claims effectively. Therefore, we set out to develop a prototype for an application that we believe will optimize this process by providing a clean and simple to use interface that will give Globe a better overview of ongoing claims and disputes in order to allocate resources accordingly.

Throughout the development of this prototype, we made ample use of diagramming tools to create highly detailed and accurate functional, structural, and behavioral models of our planned application. Creating clear class and methods design illustrations, we were able to compile a thorough package and class diagram. Finally, we designed the first iteration for the Human Computer Interface layer, detailing how users will interact with this application. A deployment diagram was also created to outline where each layer will be located on the cloud. This has led to our prototype build for the application that meets or exceeds the business needs of Globe Electric. Details of the physical architecture plan are located on a GitHub link.

The purpose of this report is to provide documentation of the development of this prototype along with an extensive list of requirements needed to deploy this application into Globe's current claims system. Details of specific training and communication activities are included in our proposed implementation plan, to serve as a guideline for Globe throughout the process.

Table of Contents

Proposal

BTM 481 Project Details

In Fall 2020, 3 members of our team completed our analysis on Globe Electric. Globe Electric is a lighting and electric products manufacturer and wholesaler, founded in Montreal in 1932. From manufacturing light bulbs and other simple appliances, Globe Electric is now at the forefront of smart home technology. Describing themselves as the "creative energy company", Globe prides themselves on developing and designing innovative new lighting products that delight consumers with their ease of use and connectivity features. Now, Globe Electric products are sold at major national retailers such as Costco and Best Buy as well as the e-commerce giant, Amazon.

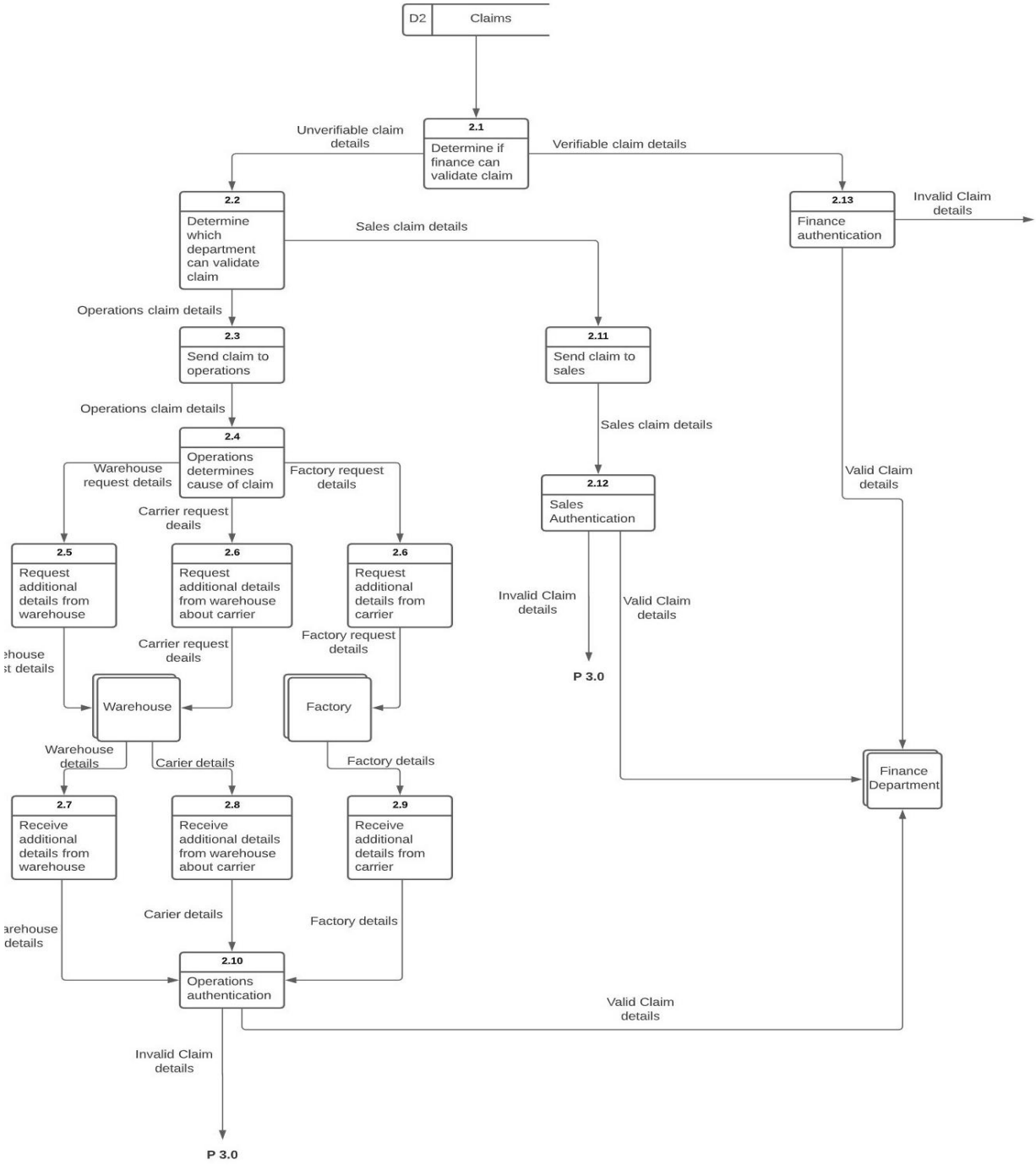
However, the advent of digital commerce did bring along additional challenges that are difficult to deal with. Having to manage a vast network of retail clients, all with unique contracts and terms leads to a high number of deductions. A deduction can occur for multiple reasons, such as when the client either enters a promotional period where they are selling their products for a discount, or if the client believes Globe has violated part of the terms in their agreement. While some deductions are mutually agreed upon, most of them come as a surprise to Globe and require further investigation. Such an example of these are short shipments, where the client claims that Globe did not send them the right quantity of a certain product. Globe Electric estimates that 1000 of these deductions are made every month and can range in costs from a few hundred to thousands of dollars of lost revenues.

Our project will be focusing on creating an application that will optimize the existing processes within subsystem 2 (Validate Claim). While looking through the deductions process, we narrowed down what we believe to be a major hindrance to the process. Whenever a client files a claim, it is often followed with a deduction code. Because clients use different systems to file these claims, these deductions codes will often be different despite representing the same deduction. We found that integrating a deduction library into Globe's current system would allow for automatic translation of customer deduction codes into a unified library, where deductions can clearly be labeled and sent to the appropriate department for review. We also wanted to optimize the flow of

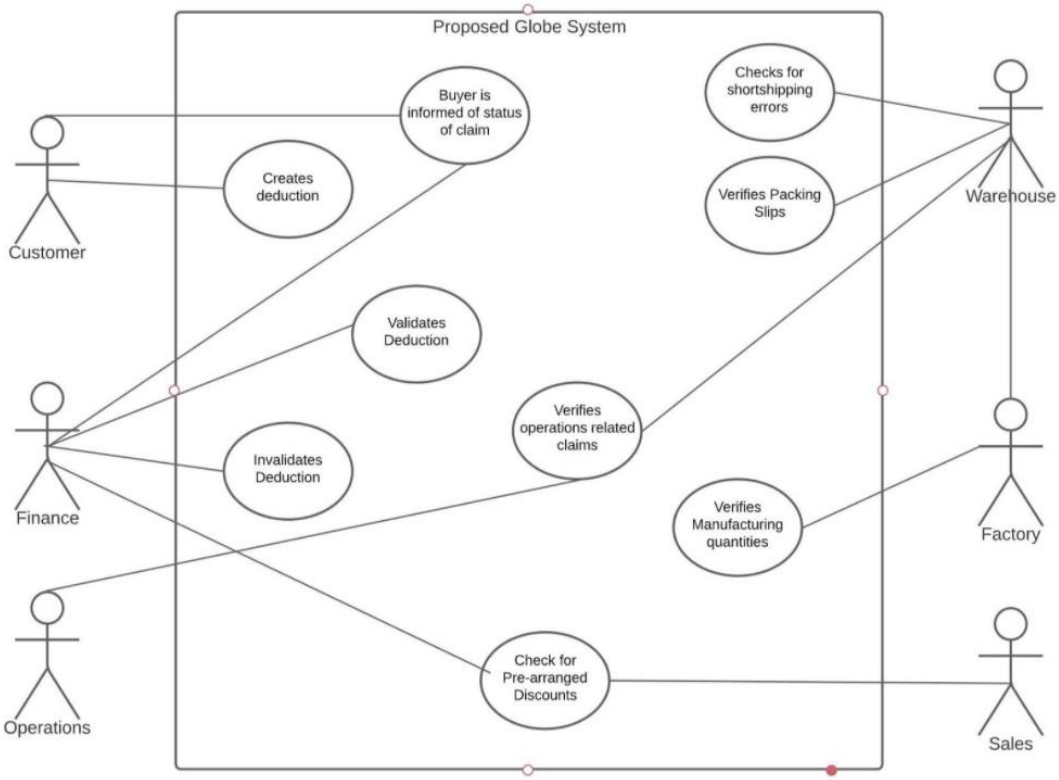
documents in order to facilitate the process of claim validation, which often requires documents such as the Bill of Lading or packing slips from the Operations department.

We want to implement a platform that can assist with not only the identification and assignment of claims, but also an invaluable financial analysis tool. This tool could aid the finance department with identifying sources of revenue loss, but to also negotiate future contracts based on data gathered from past deductions. This tool could not only provide data but also generate quarterly reports for individual departments, highlighting potential inefficiencies. Additionally, our project could classify claims based on the departments involved and then pair supporting documents such as packing slips or bills of lading to the claim automatically for easier validation and decision making. Furthermore, to assist with processing small claims which Globe historically always accepts, the tool can be configured to automatically accept claims that fit specific criteria, such as low dollar value. Through data gathered, we can generate reports highlighting specific products with high numbers of claims, or specific suppliers that are generating high amounts of claims as well. This can allow management to identify issues and make decisions.

Subsystem DFD



Use Case Diagram



BTM 481 Narrative

Globe Electric is an electronics and lighting manufacturer that supplies products to major retailers. When Globe onboards new customers to sell their products, a Customer Master Sheet (CMS) is created for complete written communication between the business and its customers. This sheet includes all of the terms, penalties and conditions for deductions, or amounts taken off an invoice for a variety of reasons. For example, if Globe does not send the required amount of a product to the store, they will receive a deduction in the invoice for that previously specified amount. After a deduction is received, Globe goes through an internal process to authenticate a claim.

To authenticate a claim, the Finance department must first determine the nature of the issue. Depending on the code assigned to the deduction by the customer, the finance department will determine if Operations or Sales departments are involved. A dispute involving Sales most likely involves a deduction made by the customer for a sale or promotional event, and finance must confirm if the event was approved by the Sales team or not. Sales does not communicate to Finance for every promotion period, which leads to deductions being made without finance knowing the

reason for such deduction. If a deduction code indicates it is a logistical issue such as short shipments, Finance must communicate with operations in order to retrieve documents to prove the shipment was accurate. According to the Finance department, the majority of deductions are made for short shipments which require documents from the carrier, like a bill of lading. However, due to the sheer volume of short shipment disputes, it is extremely difficult for both Finance and Operations to stay up to date.

If Globe accepts the claim, the invoice remains unchanged, and no further actions are taken. If the claim is deemed invalid, the customer is notified, and an extra payment amount is requested. In this process a back and forth between globe and the customer can happen until a settlement is reached and agreed upon. At this point the claim is closed and the process is over.

Due to the large amounts of claims that come in per year, approximately 12,0000, it is a very strenuous process to fight each one. Additionally, each customer uses different coding schemes to label their deductions. For example, Home Depot will use a different code than Costco for wrong quantity shipments or other identical deductions. Therefore, departments must constantly go back to the customer's documentation in order to properly match the codes to their deductions. Finally, many deductions have a prescribed time period where they can be contested, due to the large volume of deductions not all of them can be handled in time.

Semester of Completion: Fall 2020

Professor of 481 Section: Dr. Suchit Ahuja

Status of 481 Client: Although the client was extremely generous with their time in our previous project, the client has stated that they would no longer be available to assist us with this next phase of the project.

Project Proposal

Project Sponsor :

Globe electric is a consumer electronics company that designs and manufactures a myriad of light bulbs and smart connected fixtures. Founded in 1932 by Jack weinstein it has grown from a small business in Montreal to a global player in lighting products around the world. It does most of its business by selling in bulk to retailers around the world such as Walmart, Costco and Amazon.

Business Need :

With the advent of new sales channels come the headaches that accompany them. Globe is now selling to thousands of retailers and has started to experience problems with delivering their products. Clients in this case send a form for compensation on their invoice. Issues that can arise to customers vary from short shipments, where less product is delivered to the customer than promised, to sales deductions that allow retailers to expose their products on shelves and during expos. All the incoming claims need to be validated before finance can modify their invoices. This is where the problem lies. Too many claims are being received by Globe and they cannot validate all of them. This is caused by claims not being received in a standardized fashion. All clients have different internal codes they use to identify the different types of claims. Employees at Globe are spending their time contacting clients to figure out what the type of claim is so they can find the supporting documents to validate. Globe is in need of a software that can translate codes and provide uniform information to all finance employees so they can validate claims.more efficiently.

Business Requirements :

Globe requires the piece of software to be able to translate incoming claims into standardized documents that can be easily read by the finance team. Furthermore, they are looking for software that can create reports to keep track of claims and address them with the clients as needed. The software cannot implement changes to clients' ways of sending claims and must be usable on their existing hardware.

Business Value :

With Globe receiving upwards of 12000 claims per year any lost claim is money lost. Implementing an application that will be able to speed up the validation process will allow Globe the chance to recuperate their lost income and not lose money to expired claims. Furthermore, it will allow Globe to handle future increases in business effectively by being prepared to handle an increase in claims through more efficient claim processing.

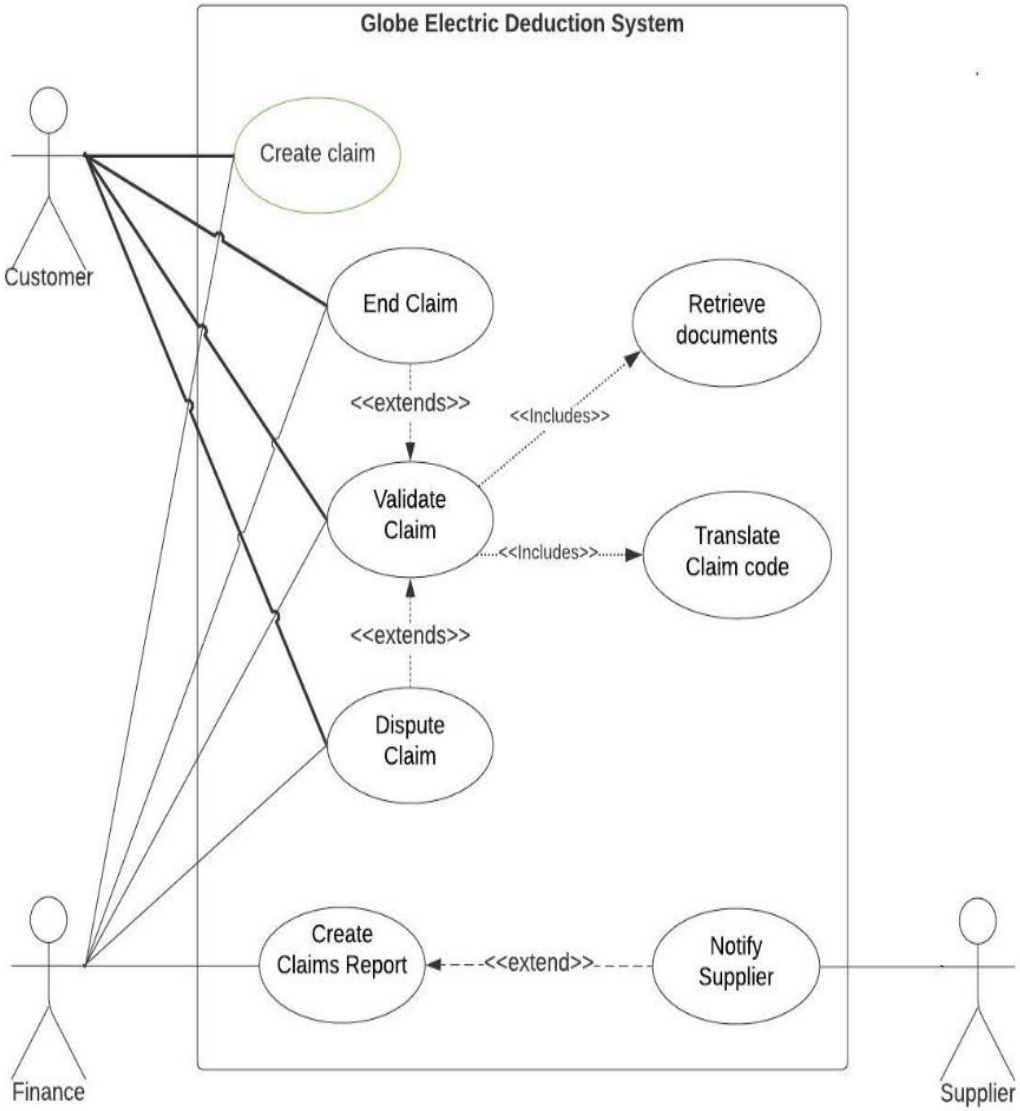
Special Issues or Constraints.

Aside from special requirements for the application to be easily integratable into Microsoft SharePoint, there are no additional special issues or constraints for this project.

Project Documentation

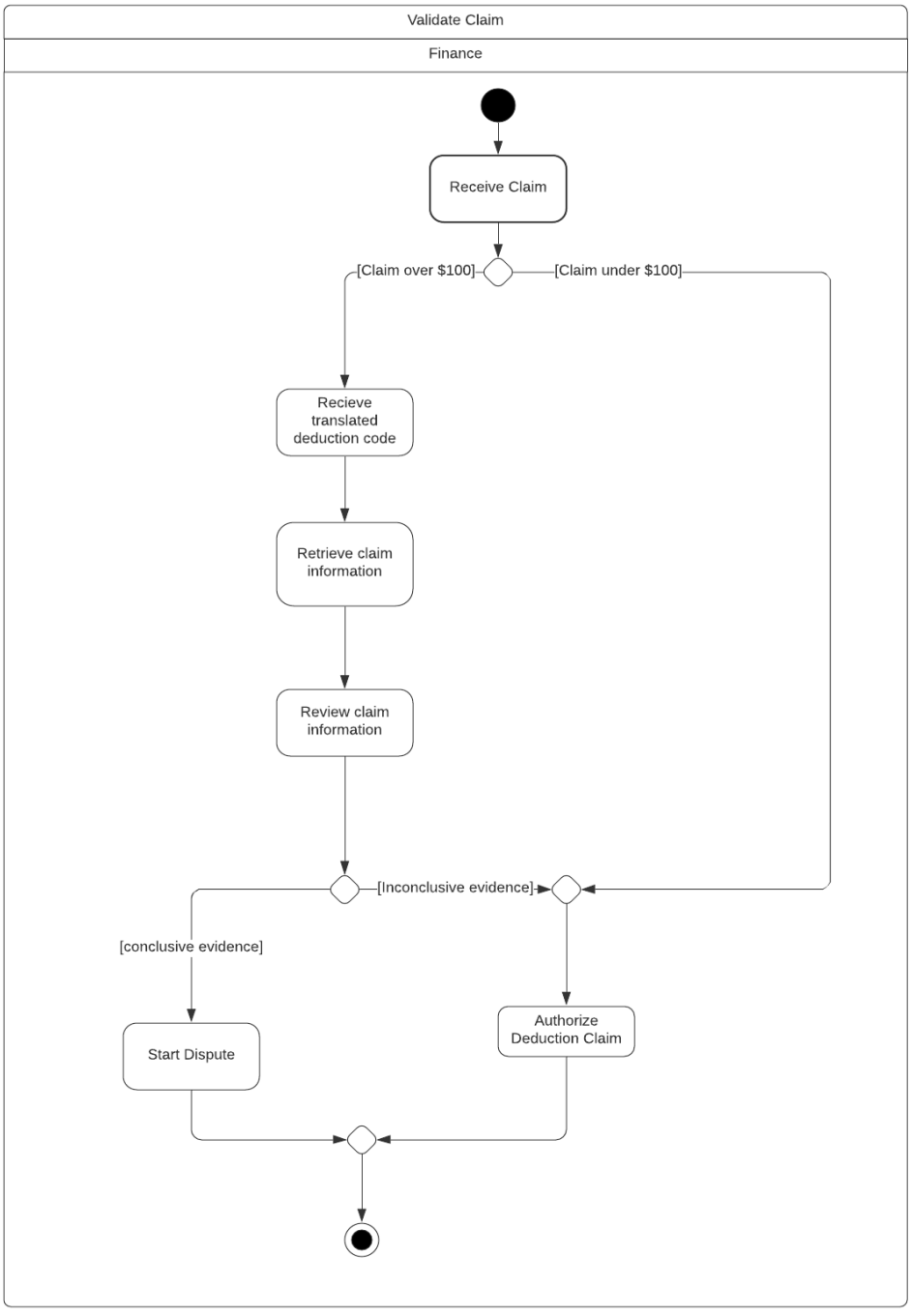
Functional Modeling

Use Case Diagram 2.0



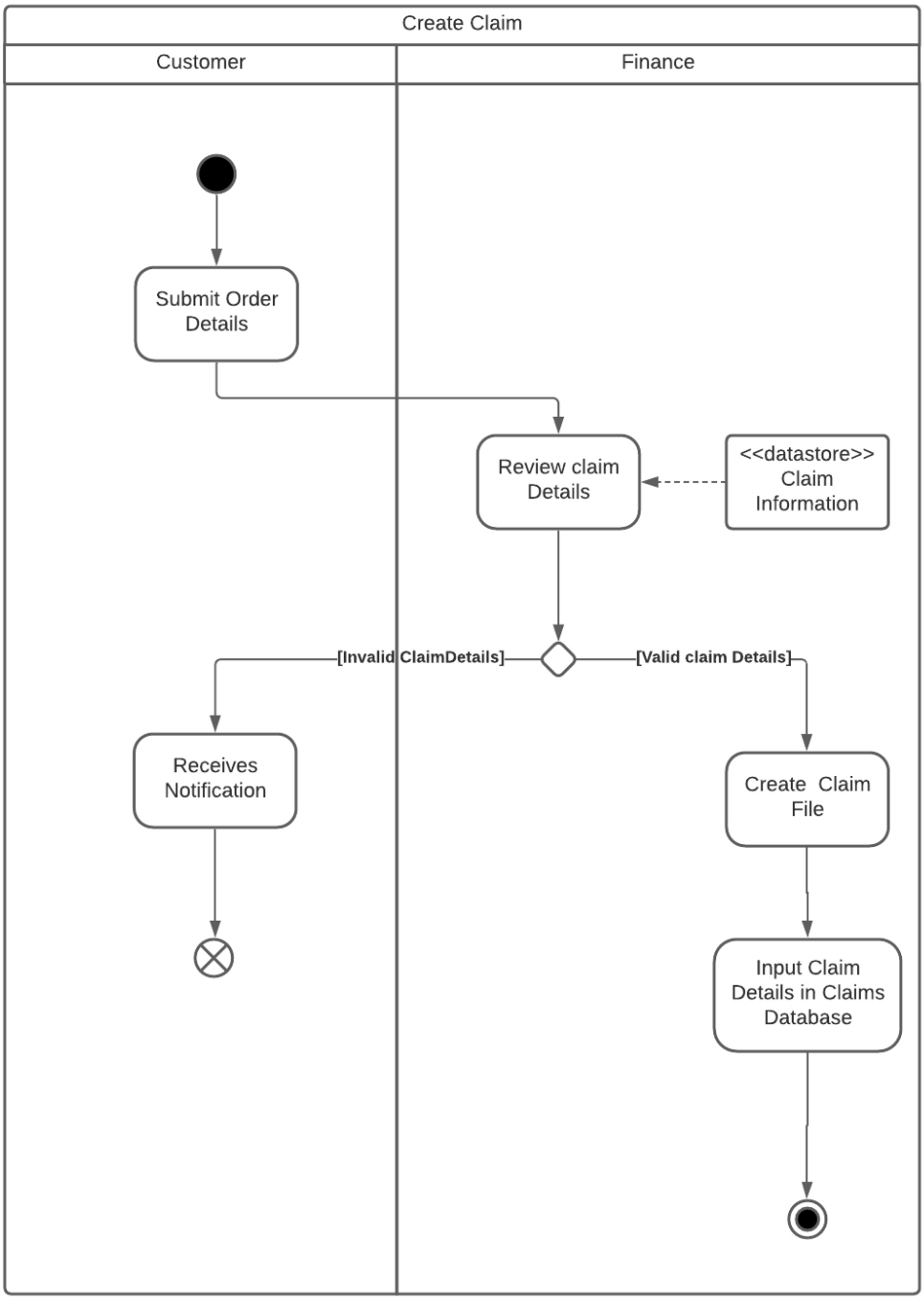
Activity Diagrams

Validate Claim 2.0



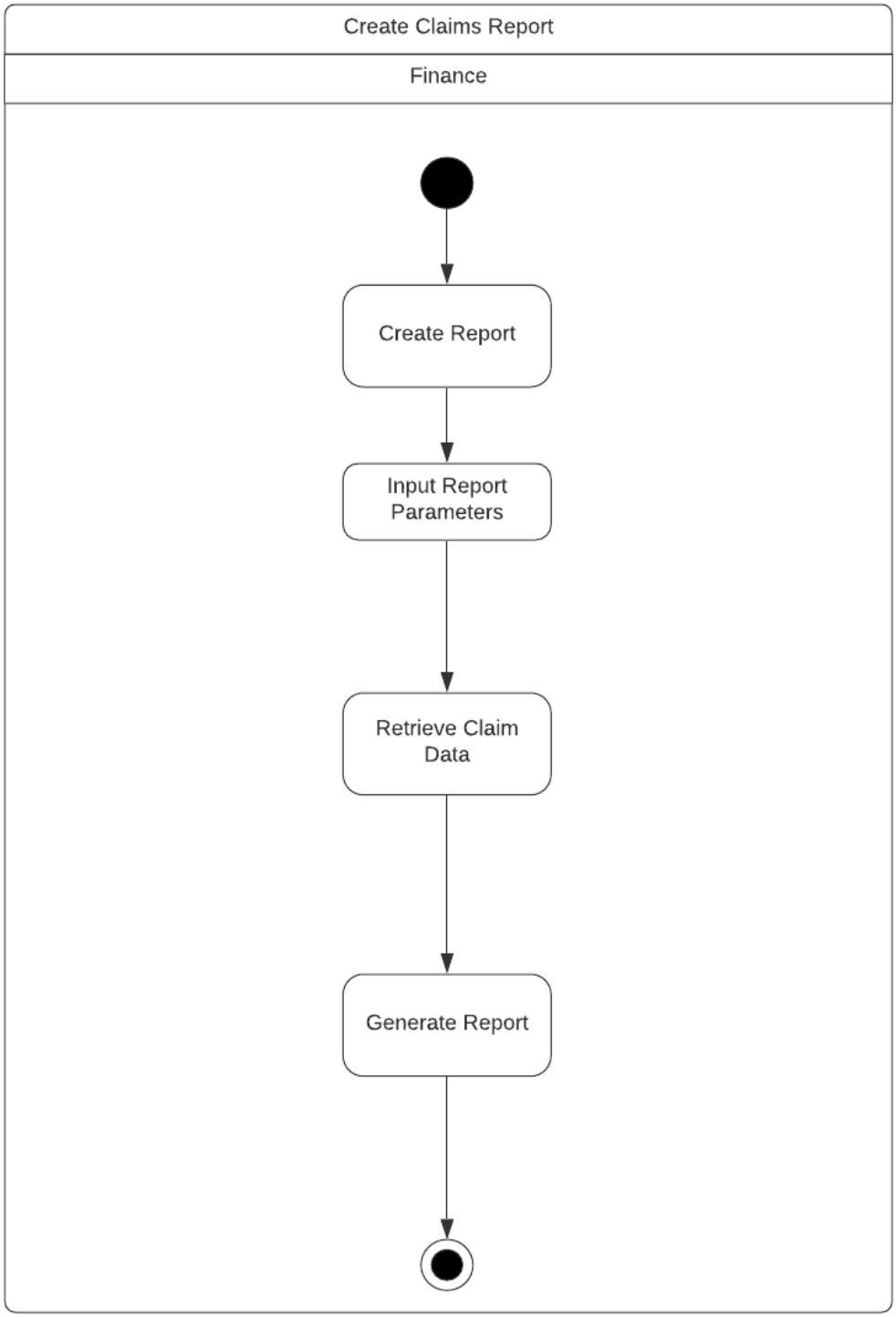
Use Case Name: Validate Claim	**ID: 2**	**Importance Level:** High
Primary Actor: Finance	**Use Case Type:** Detail, Essential	
Stakeholders and Interests: Finance - To verify legitimacy of claims recieved by customers and reduce sunken costs related to deductions		
Brief Description: This use case describes how the Finance department verifies claims recieved by customers. The system translates the claim code which is an encrypted custom code that customers have for all types of deductions. After translating the code, finance can determine if the claim is legitimate or not, and can notify the customer. The Finance department can also attach files to the database.		
Trigger: Claim is recieved	**Type:**External	
Relationships: Association:Finance Include: Extend: Dispute Claim, Pay Balance **Generalization:**		
Normal Flow of Events: 1. Finance receives claim If the claim is over $100, proceed to translate deduction code. 2. Deduction code on the order is translated 3. Finance checks claims database for order information 4. Finance reviews the order 5. Finance finds conclusive evidence that the claim is invalid, Finance starts dispute. Execute dispute use-case		
SubFlows 1.a If the claim is under $100, authorize the deduction claim. 2a. Finance does not find conclusive evidence to dispute the claim, proceeds to authrorize the deduction.		
Alternate/Exceptional Flows:		

Create Claim 2.0



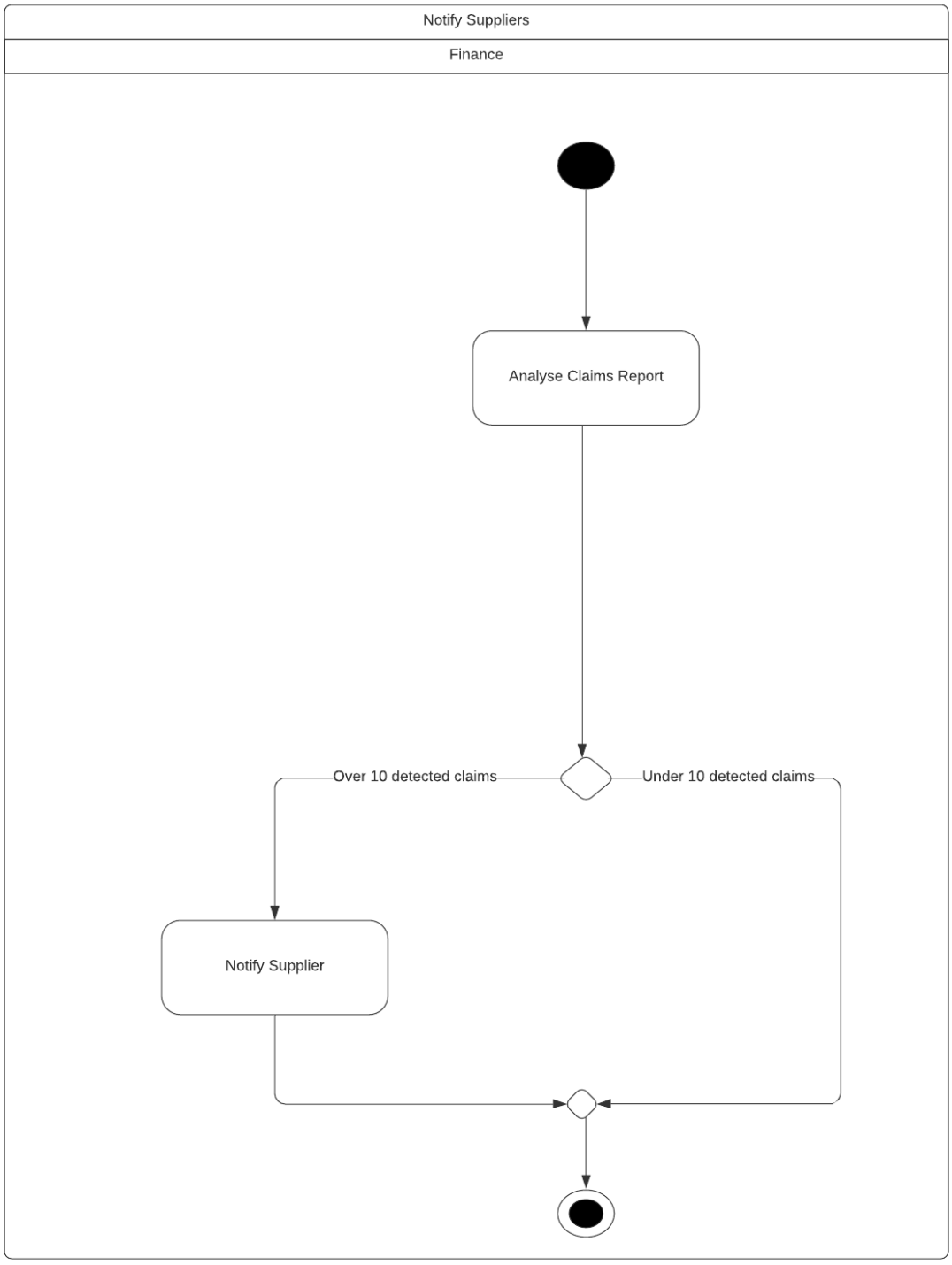
Use Case Name: Create Claim	**ID:** 1	**Importance Level:** High
Primary Actor: Finance	**Use Case Type:** Detail, Essential	
Stakeholders and Interests: Finance – wants to ensure claim details are correct Customer – wants to claim his deduction		
Brief Description: This use case describes how claim details are reviewed and claims are created in the system.		
Trigger: Customer sends claim details to Global Electric. **Type:** External		
Relationships: Association: Finance, Customer Include: - Extend: - Generalization: -		
Normal Flow of Events: 1.Claim details are received from the customer. 2.The claim details are reviewed by the finance department. 3.If the claims details are invalid, the customer is notified. 4. If the claims details are valid, finance department creates a claim file. 5. The finance department inputs claim details into to the claim file.		
SubFlows: -		
Alternate/Exceptional Flows: -		

Create Report 2.0



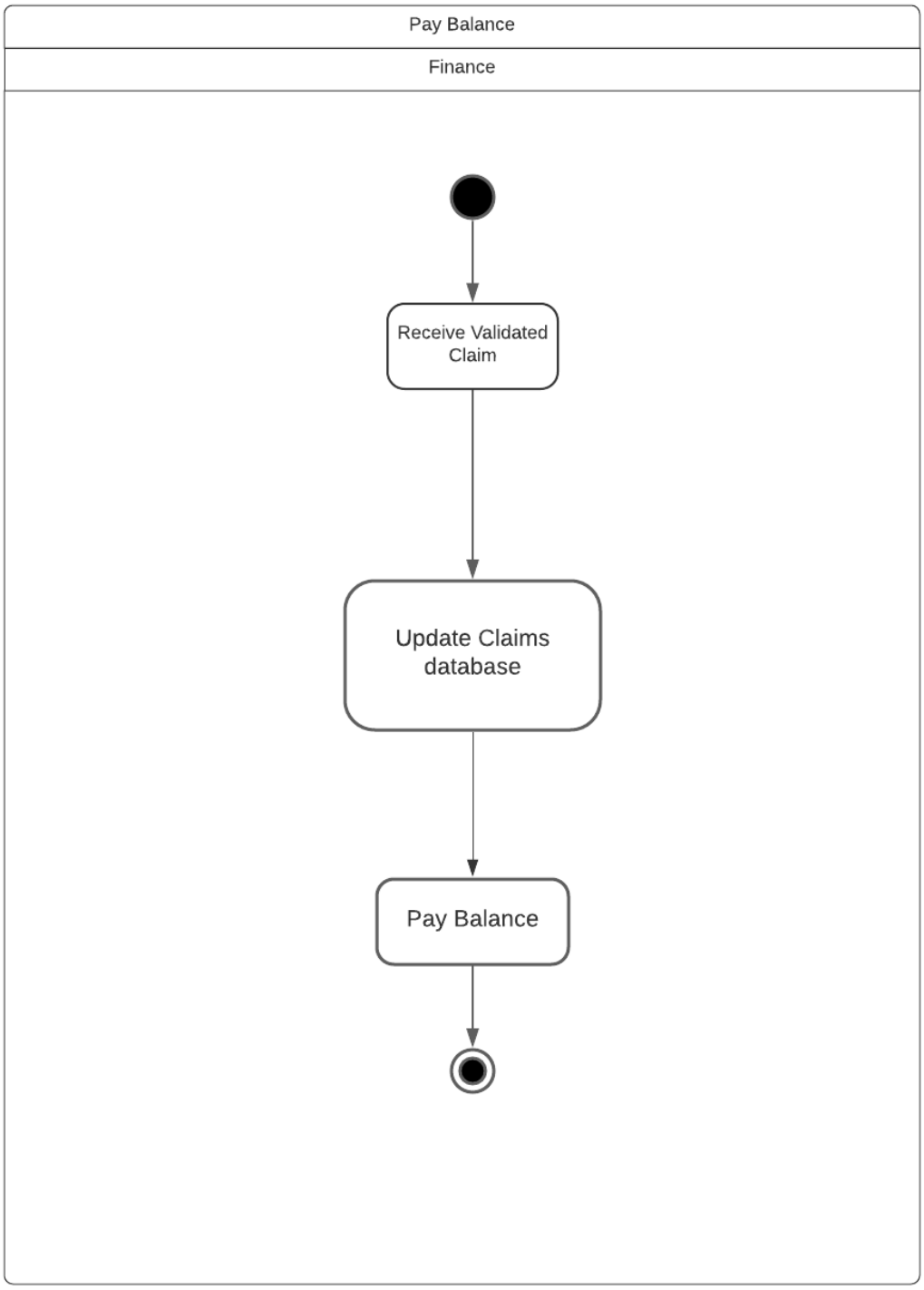
Use Case Name: Create Claims Report	**ID: 5**	**Importance Level:** Medium
Primary Actor: Finance	**Use Case Type:** Detail, Essential	
Stakeholders and Interests: Finance - To generate reports based on claims recieved within a specific time period or from a specific department ---------- Supplier: Receives the report		
Brief Description: This use case describes how the Finance department can create reports for the amount of claims recieved within a time period, to make predictive budgeting and accounting decisions as well as identify quality control issues with certain suppliers and manufacturers.		
Trigger: Report time and settings are input by Finance **Type:**Temporal		
Relationships: Association:Finance Include: Extend: Generalization: Create Report		
Normal Flow of Events: 1. Finance department starts to create a report 2. Finance inputs specified time period and/or department and/or customer parameters for creating the report 3. All claim data corresponding to the parameters is retrieved from the Claims database		
SubFlows		
Alternate/Exceptional Flows:		

Notify Suppliers 2.0



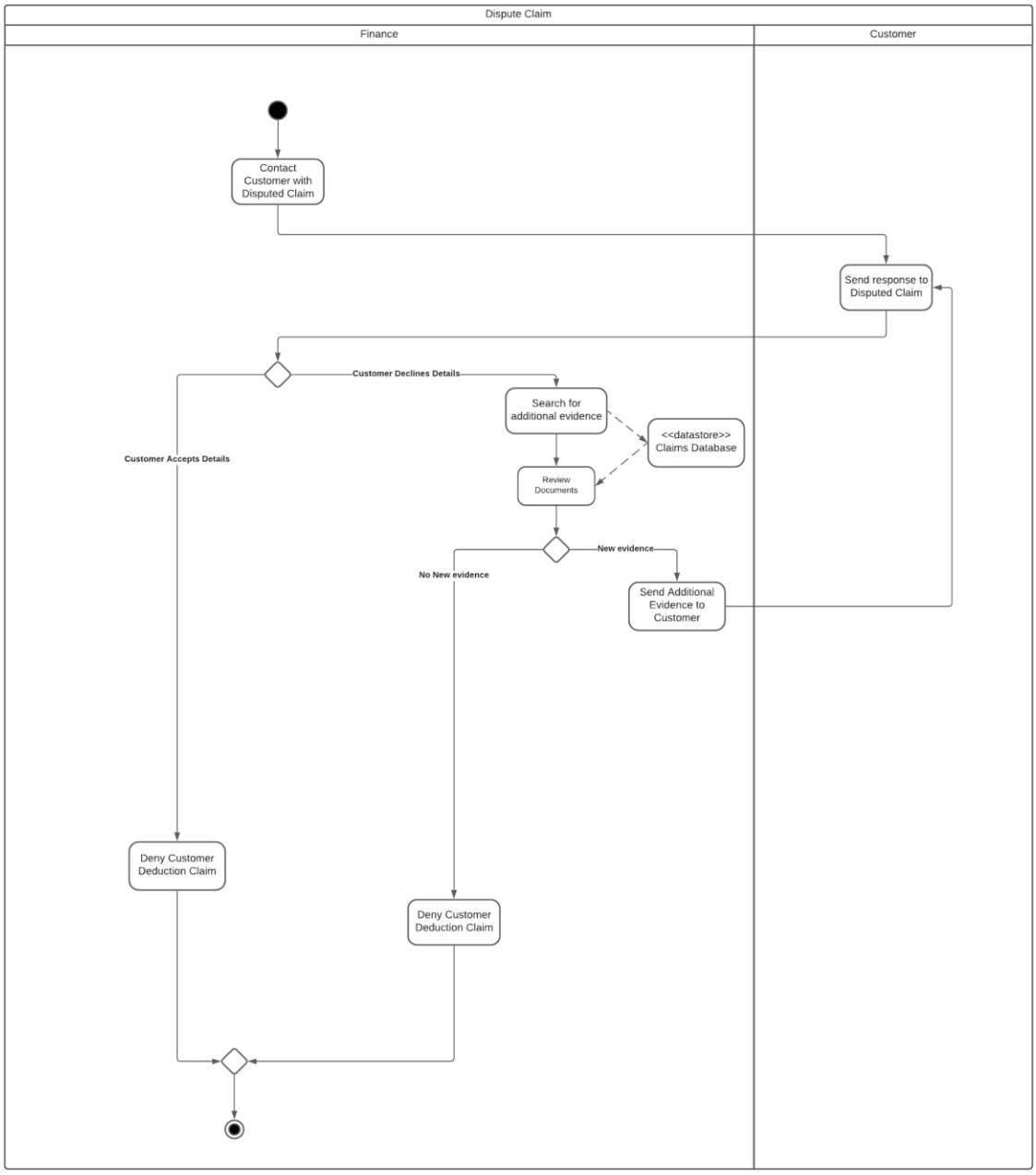
Use Case Name: Notify Supplier	**ID: 7**	**Importance Level:** Low
Primary Actor: Finance	**Use Case Type:** Detail, Essential	
Stakeholders and Interests: Finance Suppliers (they are notified if there's more than 10 claims)		
Brief Description: The end of month report analysis includes the number of identical claims suppliers have. If a supplier has 10 or more claims, then they get notified. If they have 10 or less claims, then the process remains the same and the extend wouldn't be necessary.		
Trigger: Supplier has 10 or more claims in a month. **Type:** Temporal		
Relationships: Association: Include: Extend: "Create Claims Report" Generalization:		
Normal Flow of Events: 1.Finance analyses report generated by the system from the create claims report use case. 2.If over 10 detected claims, Finance chooses to send report to suppliers.		
SubFlows: 2a. If under 10 claims are detected, no action is taken.		
Alternate/Exceptional Flows:		

End Claim 2.0


Pay Balance
Finance
Receive Validated Claim
Update Claims database
Pay Balance

Use Case Name: End Claim	**ID: 4**	**Importance Level:** High
Primary Actor: Finance	**Use Case Type:** Detail, Essential	
Stakeholders and Interests: Finance Customer		
Brief Description: After a claim is validated and accepted Finance notifies department to make necessary changes.		
Trigger: Claim is accepted in the Validate Claim Phase **Type:** External		
Relationships: Association: Include: Extend: “Validate Claim” Generalization:		
Normal Flow of Events: 1.Validated claim is recieved. 2.Update claims database with status of claim. 3.Pay claim balance		
SubFlows:		
Alternate/Exceptional Flows:.		

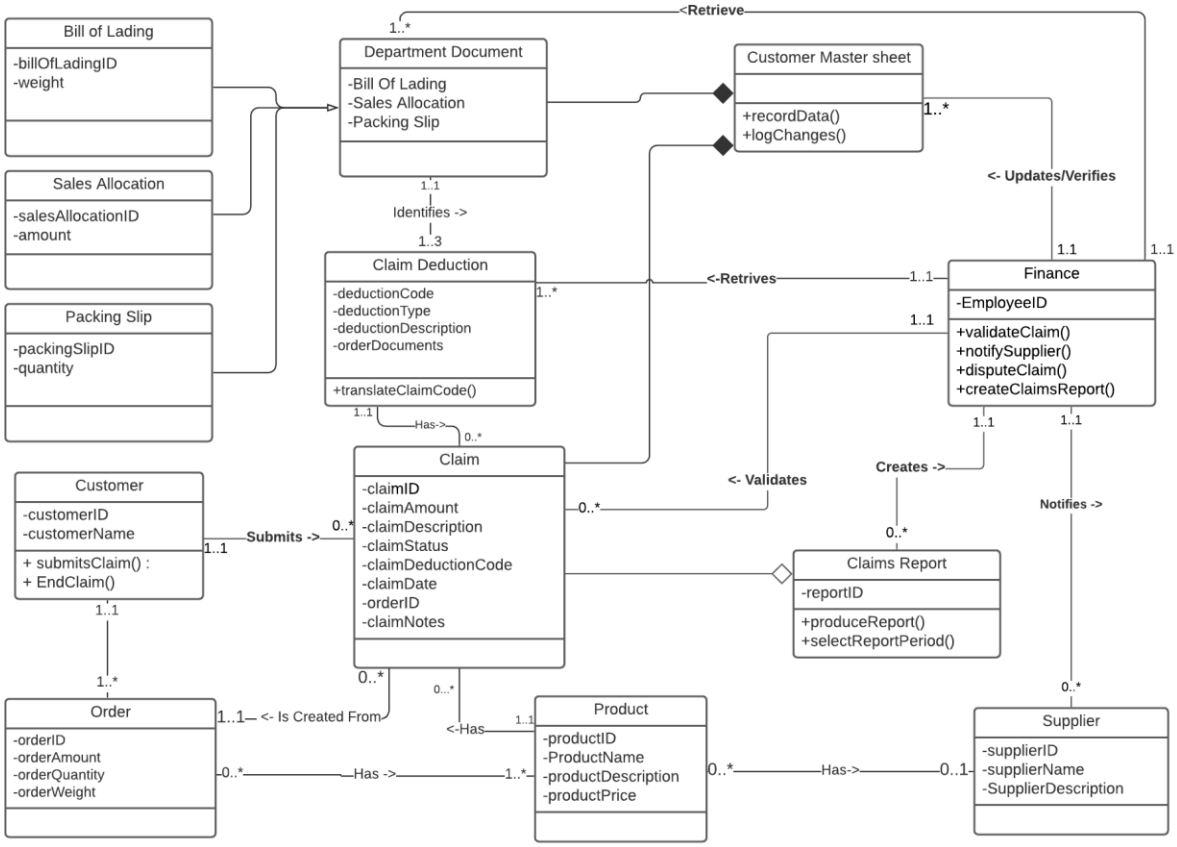
Dispute Claim 2.0



Use Case Name: Dispute Claim	ID: 3	Importance Level: High
Primary Actor: Finance	Use Case Type: Detail, Essential	
Stakeholders and Interests: Finance Department-wants to disprove deduction claim and receive their refund Customer-wants to prove their deduction claim is valid		
Brief Description: This use case describes the back-and-forth between the finance department and customer when disputing a claim.		
Trigger: Finance find that the customer's deduction claim is invalid Type:External		
Relationships: Association: Finance, Customer Include: Extend: Validate Claim Generalization:		
Normal Flow of Events: 1.Finance contacts customer disputing their deduction claim. 2.If the customer accepts the dispute The S-1: Deny Customer Deduction Claim. 3.If the customer declines the dispute The S-2: Search for Additional Evidence is performed.		
SubFlows: **S-1:**Deny Customer Deduction Claim. 1.Customer accepts the details of the disputed claim, finance proceeds to deny the customer's deduction claim. **S-2:**Search for Additional Evidence 1.Finance will search for additional evidence from the order master sheet. 2.New evidence will be sent to the customer. 2a. When there is no new evidence the customer is denied and has to pay the balance.		
Alternate/Exceptional Flows:		

Structural Modeling

Class Diagram 2.0



CRC Cards 2.0

Front:		
Class Name: Claim Deduction	**ID: 1**	**Type:** Concrete, Domain
Description This class contains all information pertaining to a specific claim.		**Associated Use Cases:** Validate Claim
Responsibilities Validate Claim	**Collaborators**	

Back:
Attributes: -deductioncode- int -deductiontype- string -deductiondescription - string -orderdocuments - string
Relationships: **Generalization (a-kind-of):** **Aggregation (has-parts):** **Other Associations:** Claim

Front:		
Class Name: Customer	**ID:** 3	**Type:Concrete, Domain**
Description: This class contains information regarding customer who does not interact with the system directly but rather submits a email containing the claim information for the finance department to create the claim request in there system.		**Associated Use Cases:** Submit Claim
Responsibilities Submits Claim	**Collaborators** Claims	

Back:
Attributes: customerID - int customerName - string
Relationships: **Generalization (a-kind-of):** **Aggregation (has-parts):** **Other Associations:** Claim

Front:		
Class Name: Finance	**ID:** 2	**Type:** Concrete, Domain
Description This class manages the claim validation process and updates the customer master sheet accordingly.		**Associated Use Cases:** Validate Claim, Dispute claim, Create Claims report
Responsibilities Reviews Claim Generates email to notify supplier	**Collaborators** Claim Supplier	

Back:
Attributes: EmployeeID - int
Relationships: **Generalization (a-kind-of):** **Aggregation (has-parts):** **Other Associations:** Claims report, Customer Master sheet, Supplier, Claim type, Claim

Front:		
Class Name: Supplier	**ID:** 4	**Type:** Concrete, Domain
Description This class contains information about the supplier who does not interact with our system but rather will get notfied by the finance department if our system detects that they have been associated with reoccurring claims fromt the claims report.		**Associated Use Cases:** notify supplier
Responsibilities	**Collaborators**	

Back:
Attributes: -supplierID - int -supplierName - string -SupplierDescription - string
Relationships: **Generalization (a-kind-of):** **Aggregation (has-parts):** **Other Associations:** Finance

Front:		
Class Name: Claim	**ID:** 5	**Type: Concrete, domain**
Description This class contains information about the claim, more importantly the deduction code which will need to be translated in order to find the deduction type of the claim.		**Associated Use Cases:** Submit Claim, validate claim
Responsibilities	**Collaborators**	

Back:
Attributes: -claimId : int -claimAmount : double -claimDescription : string -claimStatus :string -claimDeductionCode : string -claimDate : string -orderID : int -claimNotes : string
Relationships: **Generalization (a-kind-of):** **Aggregation (has-parts):** **Other Associations:** finance, claim deduction, customer

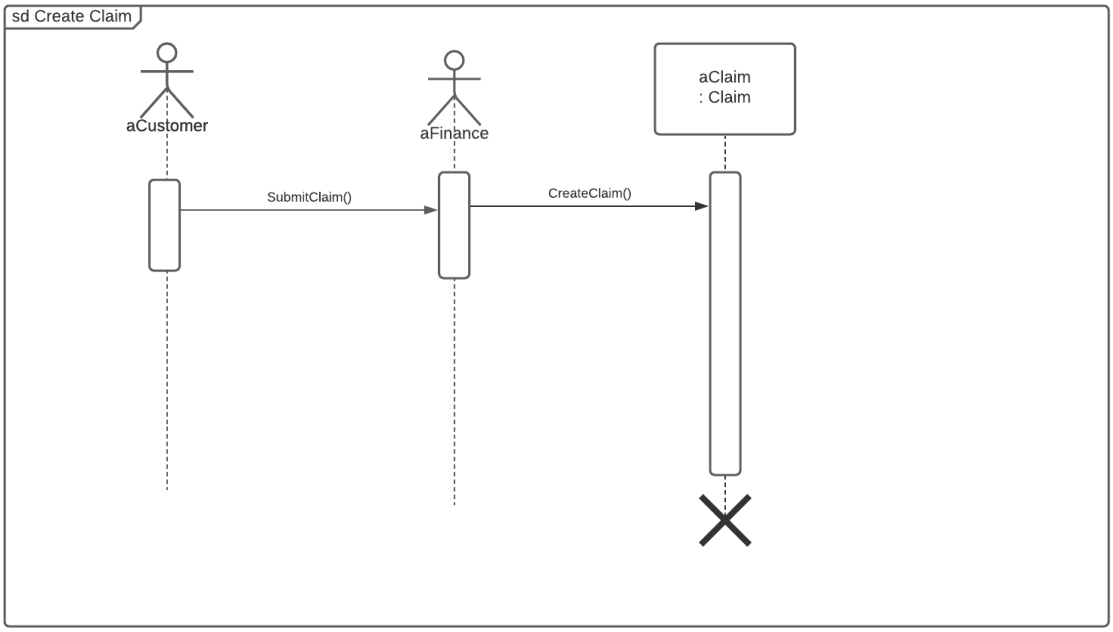
Front:		
Class Name: Customer Master Sheet	**ID:** 6	**Type: Concrete, domain**
Description The customer master sheet holds all the department documents that would be needed by finance in order to validate a claim.		**Associated Use Cases:** Create Claim, validate claim
Responsibilities	**Collaborators** Finance	

Back:
Attributes:
Relationships: **Generalization (a-kind-of):** **Aggregation (has-parts):** department documents **Other Associations:** Finance

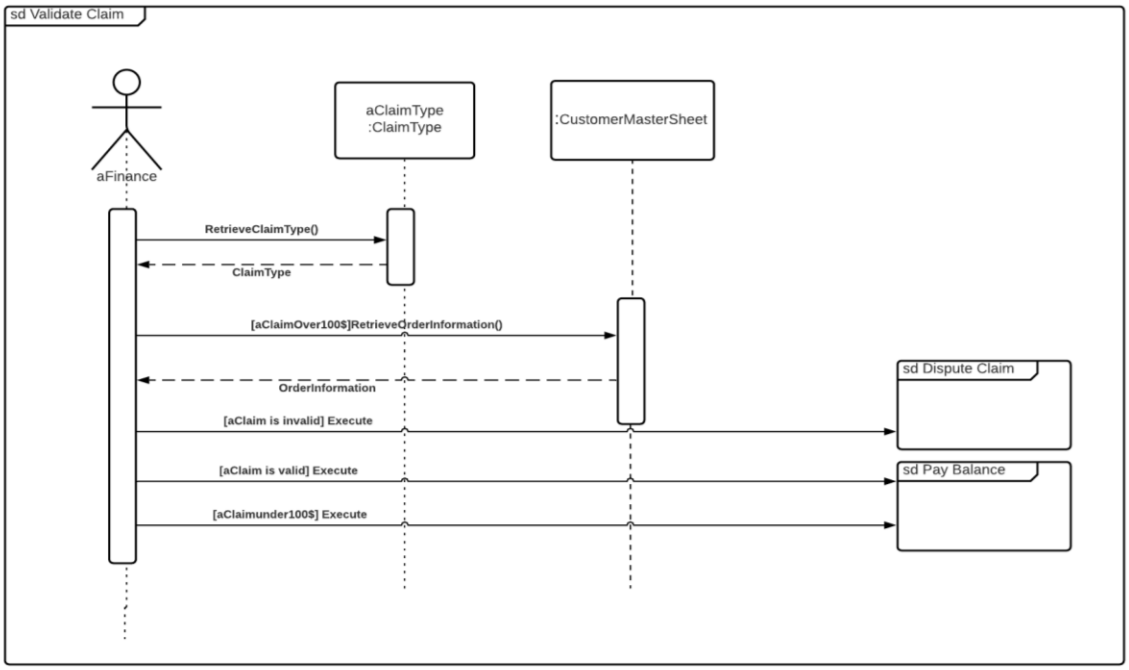
Behavioral Modeling

Sequence Diagrams

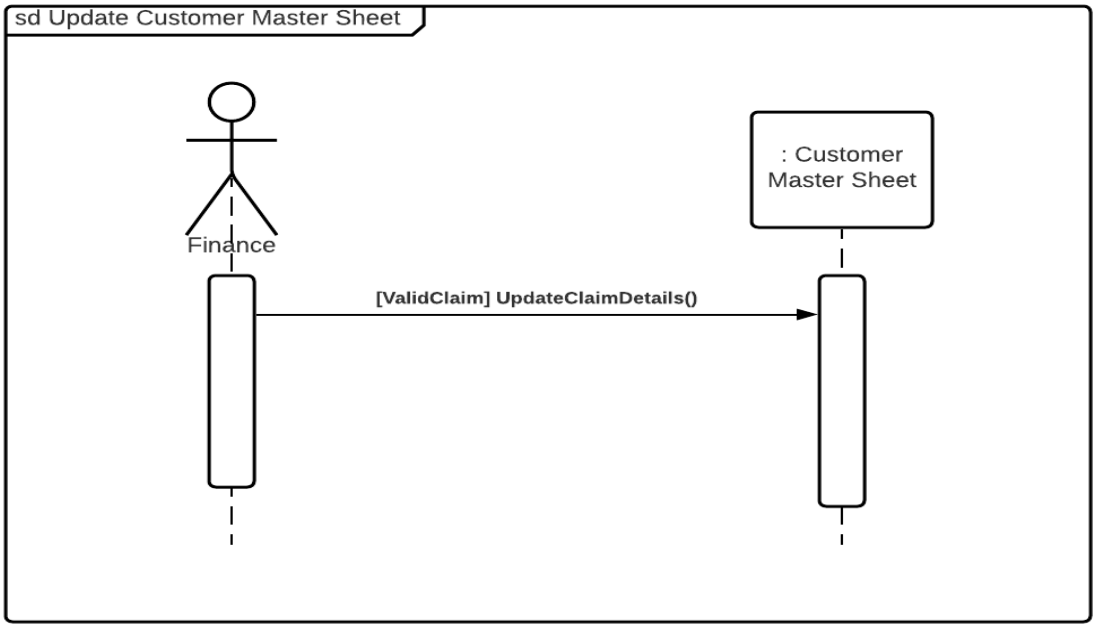
Create Claim 1.0



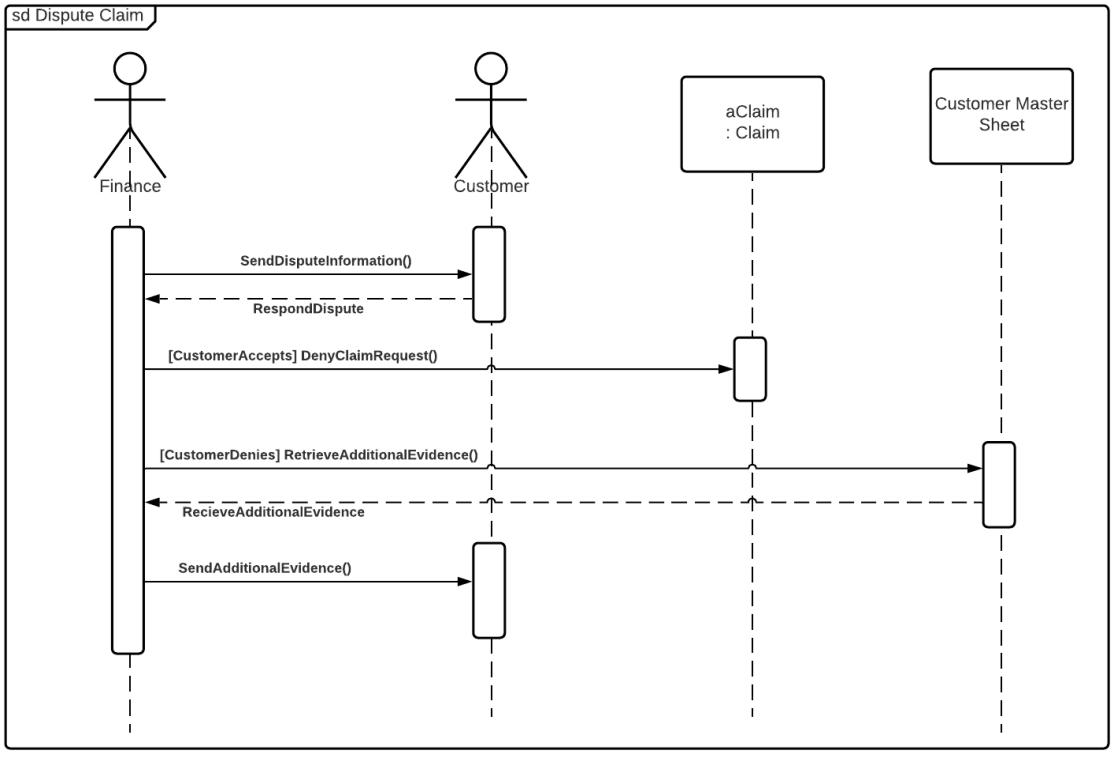
Validate Claim 1.0



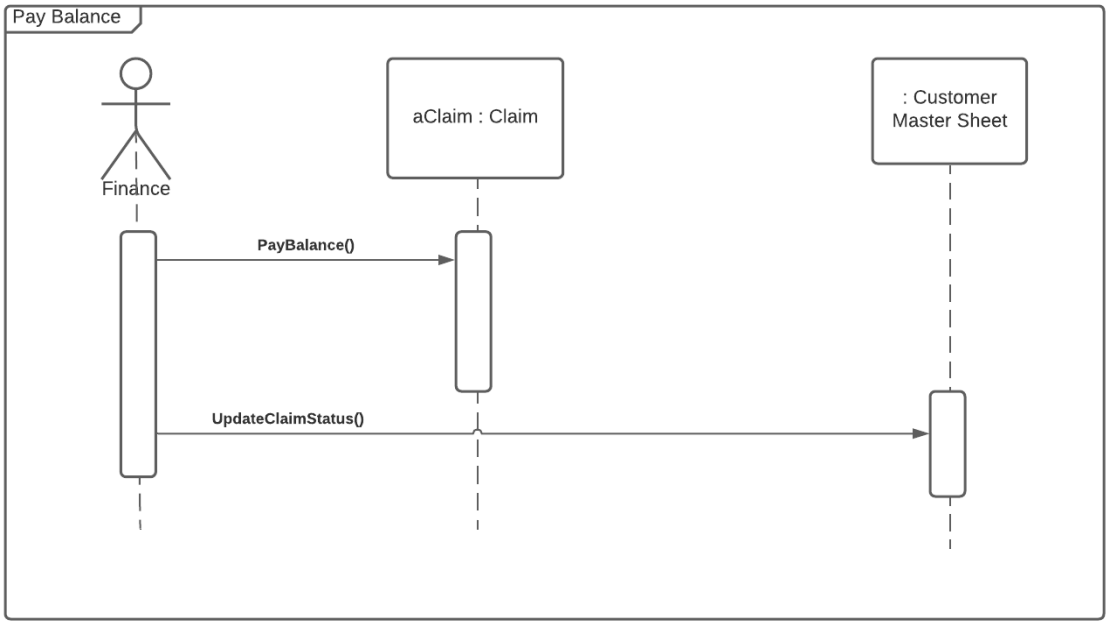
Update Customer Master Sheet 1.0



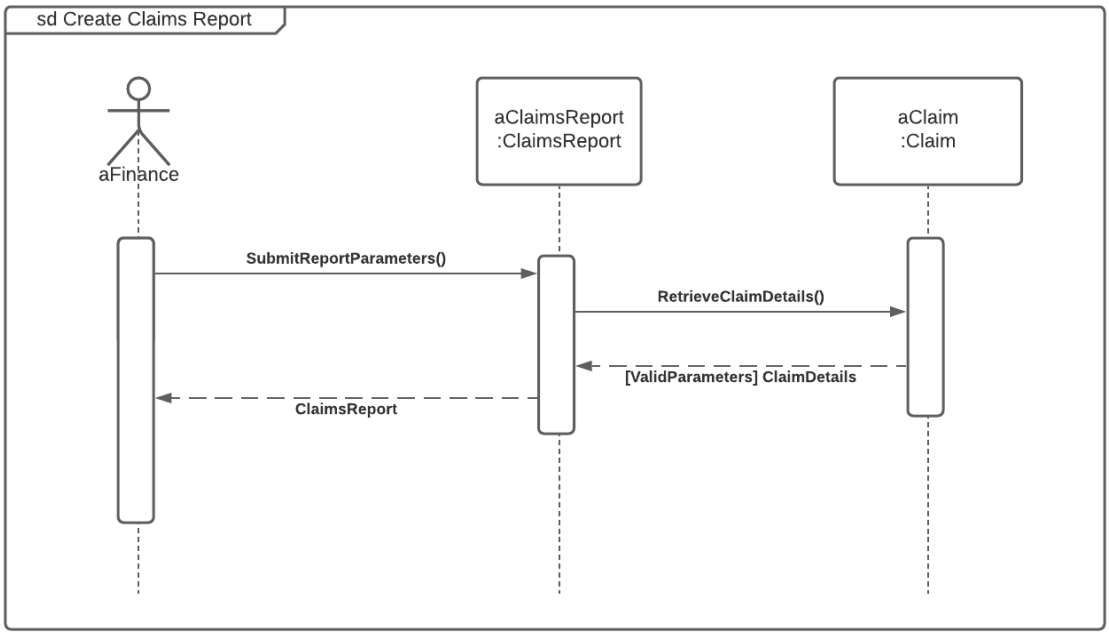
Dispute Claim 1.0



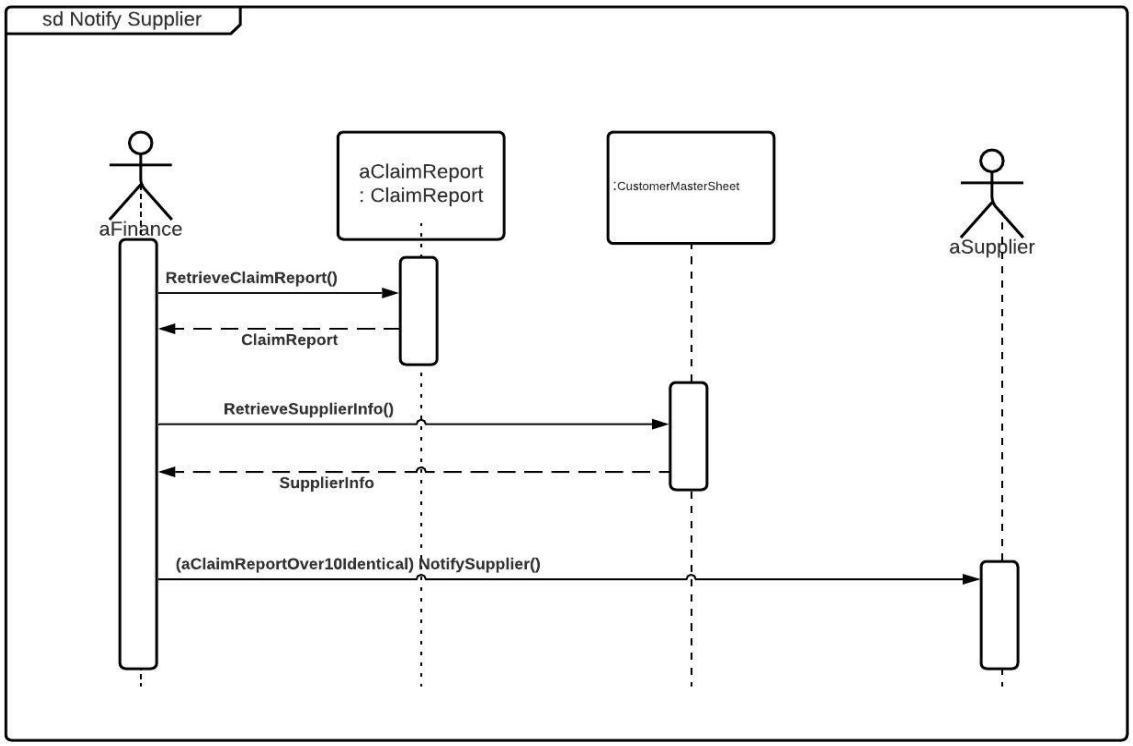
Pay Balance 1.0



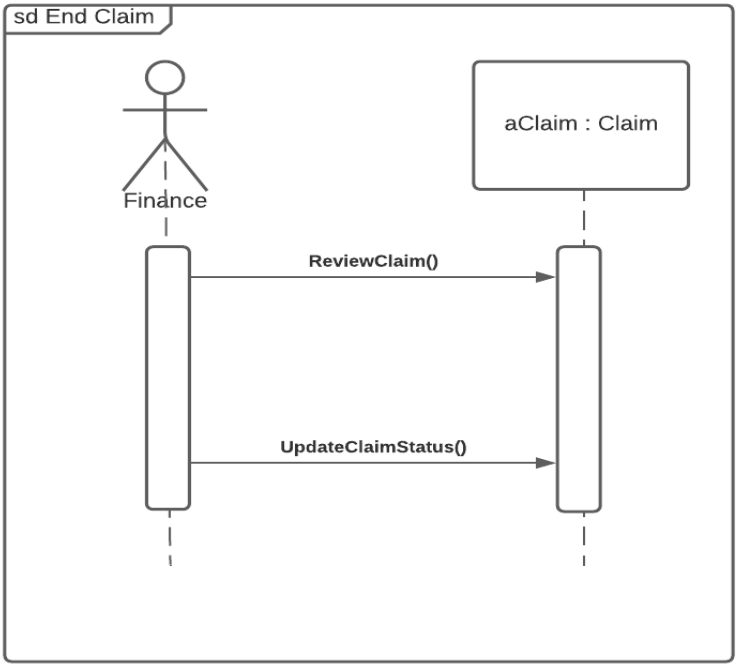
Create Claims Report 1.0



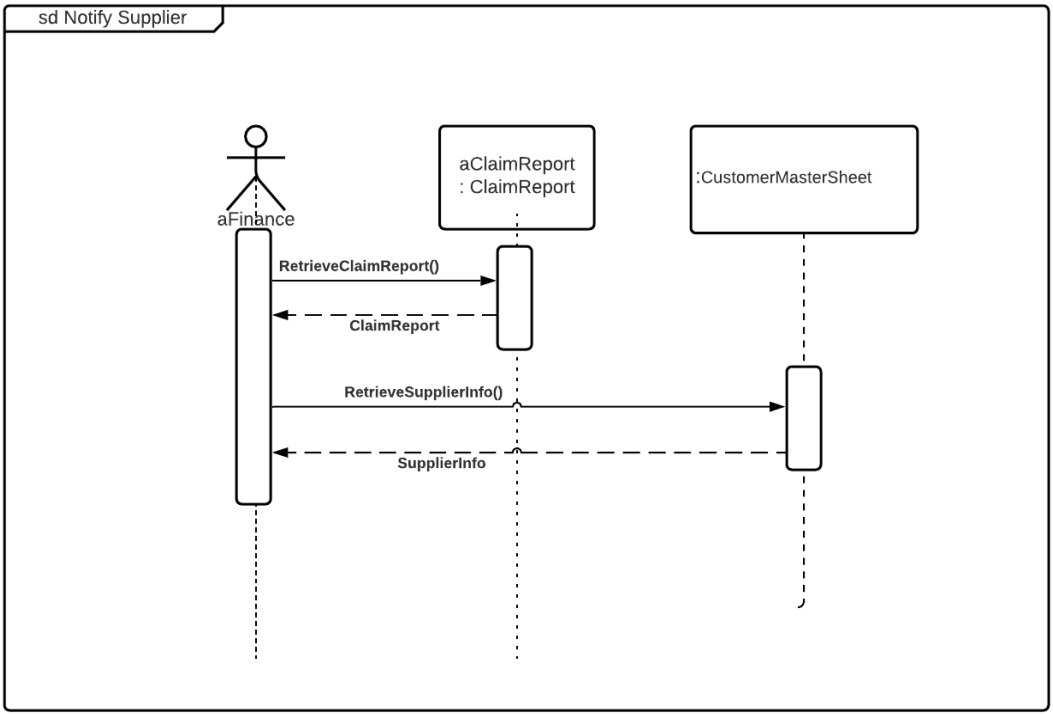
Notify Supplier 1.0



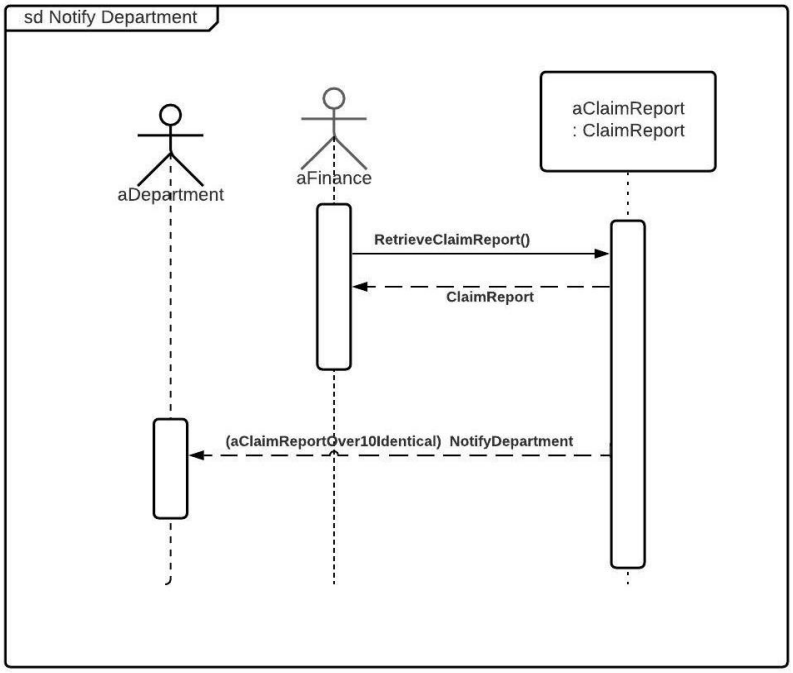
End Claim 1.0



Notify Supplier 1.0

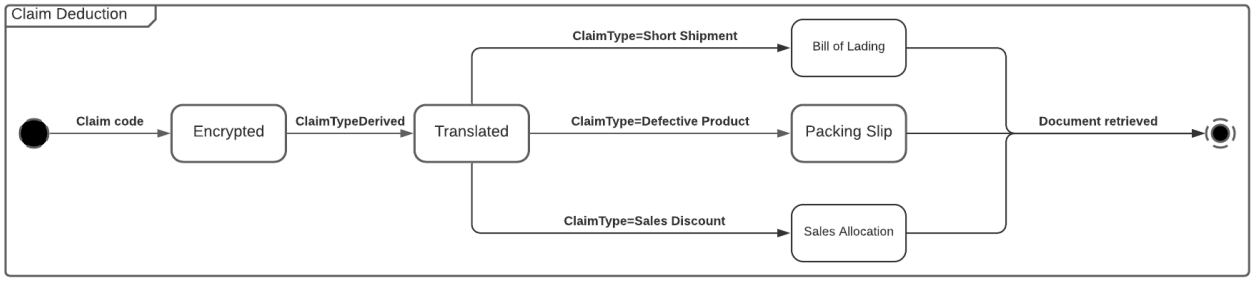


Notify Department 1.0

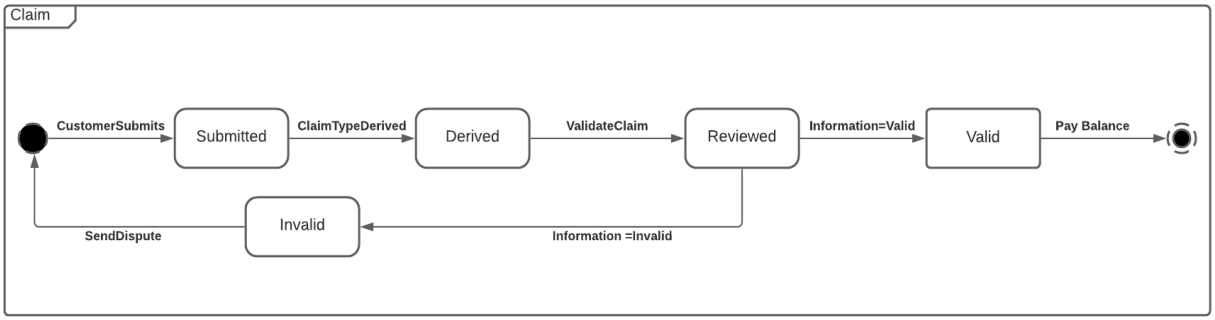


Behavior Diagrams

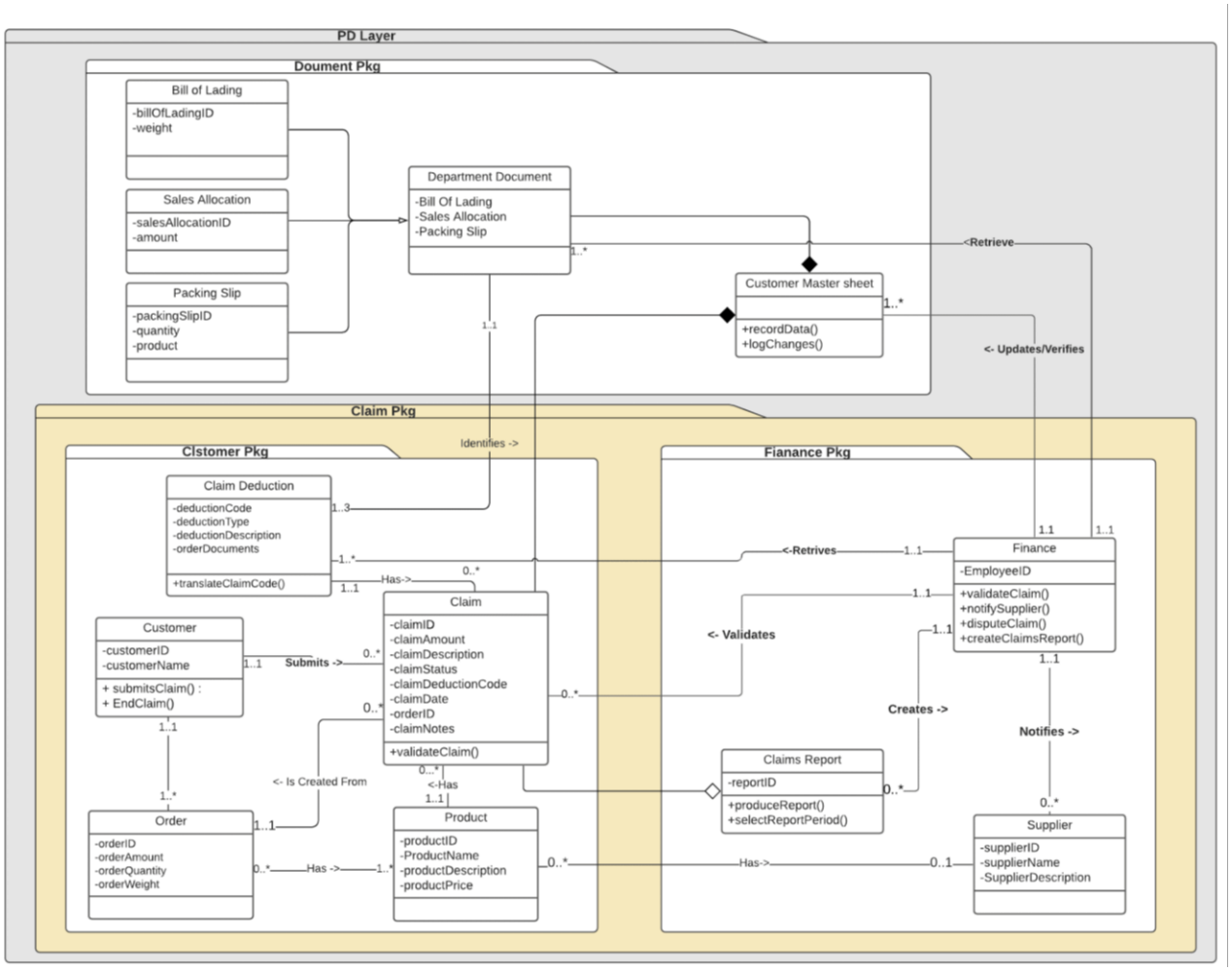
Claims Deduction 2.0



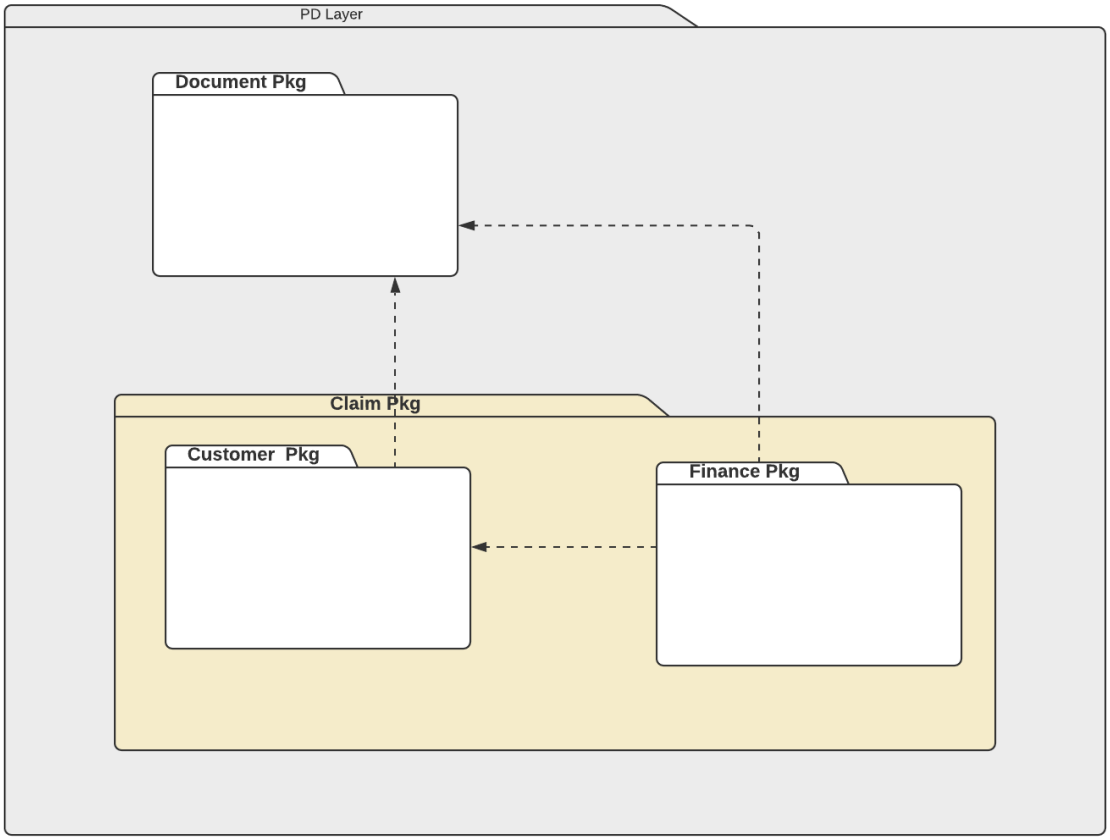
Claims 2.0



Package Diagram 2.0



Refined Package Diagram 2.0



Class and Method Design

Method Form 1

Method Name: CreateClaim	**Class Name:** Customer	**ID:** 101
Contract ID: 201	**Programmer:** Alex. J	**Date Due:** 04/04/2021
Programming Language: **Visual Basic** • **Smalltalk** • **C#** • **Java**		
Triggers/Events:		

Arguments Received: **Data Type:**	**Notes:**
ClaimCode: Varchar(10)	The code that encrypts the type of claim that must be submitted.
ClaimDate: Date	The date at which the claim is submitted.
ClaimDescription: Varchar(150)	Brief description of the issue.
Order_id: Int	The order id of the order of which the claim was needed for.
ClaimAmount:Double	The amount that is requested from the claim.

Messages Sent & Arguments Passed: **ClassName.MethodName:**	**Argument** **Data Type:**	**Notes:**
CreateClaim.new()		Inserts the data into the Mysql Database
load_data()		Loads the data in the data grid from the Mysql Database
dbConn.Open()		Opens the connection to MySql Database
dbConn.Close()		Closes the Connection to the MySql Database
cmd.ExecuteNonQuery()		Executes MySql query in Visual studio
textBox1.Clear()		Clears the text box
textBox2.Clear()		Clears the text box
textBox3.Clear()		Clears the text box
order_ID_txt.Clear()		Clears the text box
textBox4.Clear();		Clears the text box

Argument Returned: **Data Type:**	**Notes:**

Algorithm Specification:
Create new Claim
private void submit_btn_Click(object sender, EventArgs e)
{

```
        if (textBox1.Text != "##########" && textBox2.Text != "$0.00" &&
textBox3.Text != "Please add a description here." && textBox4.Text != "YYYY-
MM-DD")
        {
           MySqlCommand cmd = new MySqlCommand("Insert into
claim(Claim_Amount,Claim_Description,Claim_Deduction_Code,Claim_Date,
Order_ID)Values('" + textBox2.Text + "','" + textBox3.Text + "','" +
textBox1.Text + "','" + textBox4.Text + "','" + order_ID_txt.Text + "')", dbConn);
        dbConn.Open();
        try
        { cmd.ExecuteNonQuery();
           MessageBox.Show("The claim has been submitted successfully.");

           textBox1.Clear();
           textBox2.Clear();
           textBox3.Clear();
           textBox4.Clear();
           order_ID_txt.Clear();


        }

        catch (Exception ex)
        {
           MessageBox.Show("Error!" + "\n" + "The data entered is invalid. " +
"Please check your data inputs," + "\n\n" + ex.Message ) ;

        }
        dbConn.Close();
         load_data();
        }
        else
        {

           MessageBox.Show("Error!" + "\n" + "Either the data entered is invalid
or required data fields have been left empty.");

        }

      }
```

Misc.Notes:

Method Name: submitClaim	**Class Name:** Customer	**ID:** 101
Clients (Consumers): Claim		
Associated Use Cases: Create Claim		

Description of Responsibilities: Implement the necessary behavior to receive order details and submit the claim to the finance department.
Arguments Received: anOrder: Order
Type of Value Returned: void
Pre-Conditions: none
Post-Conditions: none

Method Form 2

Method Name: translateClaimCode	**Class Name:** Claim	**ID:** 103
Contract ID: 203	**Programmer:** Amrit. N	**Date Due:** 04/04/2021
Programming Language: **Visual Basic** • **Smalltalk** • **C#** • **Java**		
Triggers/Events: submitClaim		

Arguments Received: **Data Type:**	**Notes:**
ClaimCode: Varchar(10)	The Claim Code

Messages Sent & Arguments Passed: **ClassName.MethodName:**	**Argument Data Type:**	**Notes:**
TranslateClaimCode()		Translates the Claim code
dbConn.open()		Opens the connection to MySql Database
dbConn.Close()		Closes the Connection to the MySql Database
Load_data()		Loads the data in the data grid from the Mysql Database
Load_DeductionType()		Loads the deduction type and the corresponding order document in the data grid from the Mysql Database
SDA.SelectCommand.ExecuteNonQuery()		Executes Mysql query in Visual studio

Argument Returned: **Data Type:**	**Notes:**
isValid:Boolean	If the provided claim code is associated with a valid deduction type, then ClaimStatus will be changed to validated, and the associated order document will be displayed, otherwise it will be set to false and an error message will be displayed.

Algorithm Specification:
Translates the Claim Code

```
private void trans_btn_Click(object sender, EventArgs e)

    {
      if (textBox1.Text != "##########")
      {

        dbConn.Open();
        DataTable D = new DataTable();
        dbConn.Close();
```

```
            if
(double.Parse(validateGridView.CurrentRow.Cells[1].Value.ToString()) < 100 )
            { MessageBox.Show("As the claim is less than $100, it will be simply
paid to the customer.");
              dbConn.Open();
              MySqlDataAdapter SDA = new MySqlDataAdapter("UPDATE
Claim SET Claim_Status = 'To be Paid' WHERE Claim_Deduction_Code ='" +
textBox1.Text + "'", dbConn);
              DataTable DATA = new DataTable();
              SDA.SelectCommand.ExecuteNonQuery();
              dbConn.Close();
              Load_data();
              textBox1.Text = "##########";
              textBox1.ForeColor = Color.DarkGray;
             }

            else {

            try
               {


            Load_DeductionType();

            dbConn.Open();
              MySqlDataAdapter SDA = new MySqlDataAdapter("UPDATE
Claim SET Claim_Status = 'Validated' WHERE Claim_Deduction_Code ='" +
textBox1.Text + "'", dbConn);
              DataTable DATA = new DataTable();
              SDA.SelectCommand.ExecuteNonQuery();
              dbConn.Close();


              Load_data();

              // MessageBox.Show("The claim code has been translated. Please
check the table to see what documents you need to complete the validation
process");
              MessageBox.Show("The Deduction Code for claim has been
translated successfully." +"\n" + "The deduction type is " +
outputgrd.CurrentRow.Cells[1].Value.ToString() + ". " + "\n" + "Please find the
associated " + outputgrd.CurrentRow.Cells[2].Value.ToString() + " to complete
the validation process.");
              }

              catch (Exception)
              {
                 MessageBox.Show("Claim Deduction Code is not valid." +"\n" +
"Please check the Claim Deduction Code." );
;              }


           }
         }
         else
```

```
            {
                MessageBox.Show("Please enter a valid Claim Deduction Code");
            }

        }
```

Misc.Notes:

Method Name: translateClaimCode	**Class Name:** Deduction Type	**ID:** 103
Clients (Consumers): Finance		
Associated Use Cases: Validate Claim, Translate Claim code		
Description of Responsibilities: This method will translate incoming foreign deduction codes into familiar codes.		
Arguments Received:		
Type of Value Returned: int		
Pre-Conditions:		
Post-Conditions:		

Method Form 3

Method Name: disputeClaim	**Class Name:** Finance	**ID:** 102
Contract ID: 202	**Programmer:** Alex. J	**Date Due:** 04/04/2021
Programming Language: **Visual Basic • Smalltalk • C# Java**		
Triggers/Events: Finance deems a claim to be invalid after verification.		

Arguments Received: **Data Type:**	**Notes:**
ClaimCode: Varchar(10)	The Claim Code
ClaimNote:Varchar(150)	Description about why the claim is disputed or denied

Messages Sent & Arguments Passed: **ClassName.MethodName:**	**Argument** **Data Type:**	**Notes:**
Dispute.Claim()		Changes the status of the claim in the database and shows a message.
dbConn.open()		Opens the connection to MySql Database
dbConn.Close()		Closes the Connection to the MySql Database
SDA.SelectCommand.ExecuteNonQuery()		Executes MySql query in Visual studio
cmd.ExecuteNonQuery()		Executes MySql query in Visual studio
load_data()		Loads the data in the data grid from the Mysql Database

Argument Returned: **Data Type:**	**Notes:**
Void	

Algorithm Specification:

Dispute a Claim

```
 private void dispute_btn_Click(object sender, EventArgs e)
      {
         if (Discodetxt.Text != "##########" && Disputebox.Text != "Please add
a note here.")
          {
             dbConn.Open();
             MySqlDataAdapter SDA = new MySqlDataAdapter("UPDATE Claim
SET Claim_Status = 'Disputed' WHERE Claim_ID ='" + Discodetxt.Text + "'",
dbConn);
              DataTable DATA = new DataTable();
              SDA.SelectCommand.ExecuteNonQuery();
              dbConn.Close();
```

```
            MySqlCommand cmd = new MySqlCommand("UPDATE Claim SET
Claim_Notes = '" + Disputebox.Text + "' WHERE Claim_ID ='" +
Discodetxt.Text + "'", dbConn);
            dbConn.Open();
            cmd.ExecuteNonQuery();
            dbConn.Close();

            load_data();

            MessageBox.Show("The customer has been sent the imformation about
the claim dispute via email.");
            Discodetxt.Text = "##########";
            Disputebox.Text = "Please add a note here.";
            Discodetxt.ForeColor = Color.DarkGray;
            Disputebox.ForeColor = Color.DarkGray;

        }
            else
        {
            MessageBox.Show("Error!" + "\n" + "Either the data entered is invalid
or required data fields have been left empty");
            Discodetxt.Text = "##########";
            Discodetxt.ForeColor = Color.DarkGray;
            Disputebox.Text = "Please add a note here.";
            Disputebox.ForeColor = Color.DarkGray;
        }

        }
```

Misc.Note

Method Name: disputeClaim	**Class Name:** Finance	**ID:** 102
Clients (Consumers): Customer		
Associated Use Cases: Dispute Claim		
Description of Responsibilities: Implement the necessary behavior to notify the customer about the invalid claim and retrieve additional evidence from the customer mastersheet.		

Arguments Received: aClaim: ClaimID
Type of Value Returned: Void
Pre-Conditions: aClaim:Claim.invalid
Post-Conditions: aClaim:ClaimStatus.invalid

Method Form 4

Method Name: EndClaim	**Class Name:** Finance	**ID:** 100
Contract ID: 200	**Programmer:** Amrit. N	**Date Due:** 04/04/2021
Programming Language:		
Visual Basic • **Smalltalk** ▪ **C#** • **Java**		
Triggers/Events: A claim has been either validated, disputed or accepted and must be resolved or paid.		

Arguments Received: Data Type:	**Notes:**
ClaimCode: Varchar(10)	The Claim Code
ClaimNote:Varchar(150)	Description about why the claim is disputed or denied

Messages Sent & Arguments Passed: ClassName.MethodName:	**Argument Data Type:**	**Notes:**
End.Claim()		
dbConn.open()		Opens the connection to MySql Database
dbConn.Close()		Closes the Connection to the MySql Database
SDA.SelectCommand.ExecuteNonQuery()		Executes MySql query in Visual studio
cmd.ExecuteNonQuery()		Executes MySql query in Visual studio
load_data()		Loads the data in the data grid from the Mysql Database

Argument Returned: Data Type:	**Notes:**
Void	

Algorithm Specification:

```
private void End_btn_Click(object sender, EventArgs e)
      {
         if (Endtxt.Text != "##########" && Endnotetxt.Text != "Please add a
note here.")
         {
            dbConn.Open();
            MySqlDataAdapter SDA = new MySqlDataAdapter("UPDATE Claim
SET Claim_Status = 'Resolved' WHERE Claim_ID ='" + Endtxt.Text + "'",
dbConn);
            DataTable DATA = new DataTable();
            SDA.SelectCommand.ExecuteNonQuery();
            dbConn.Close();
```

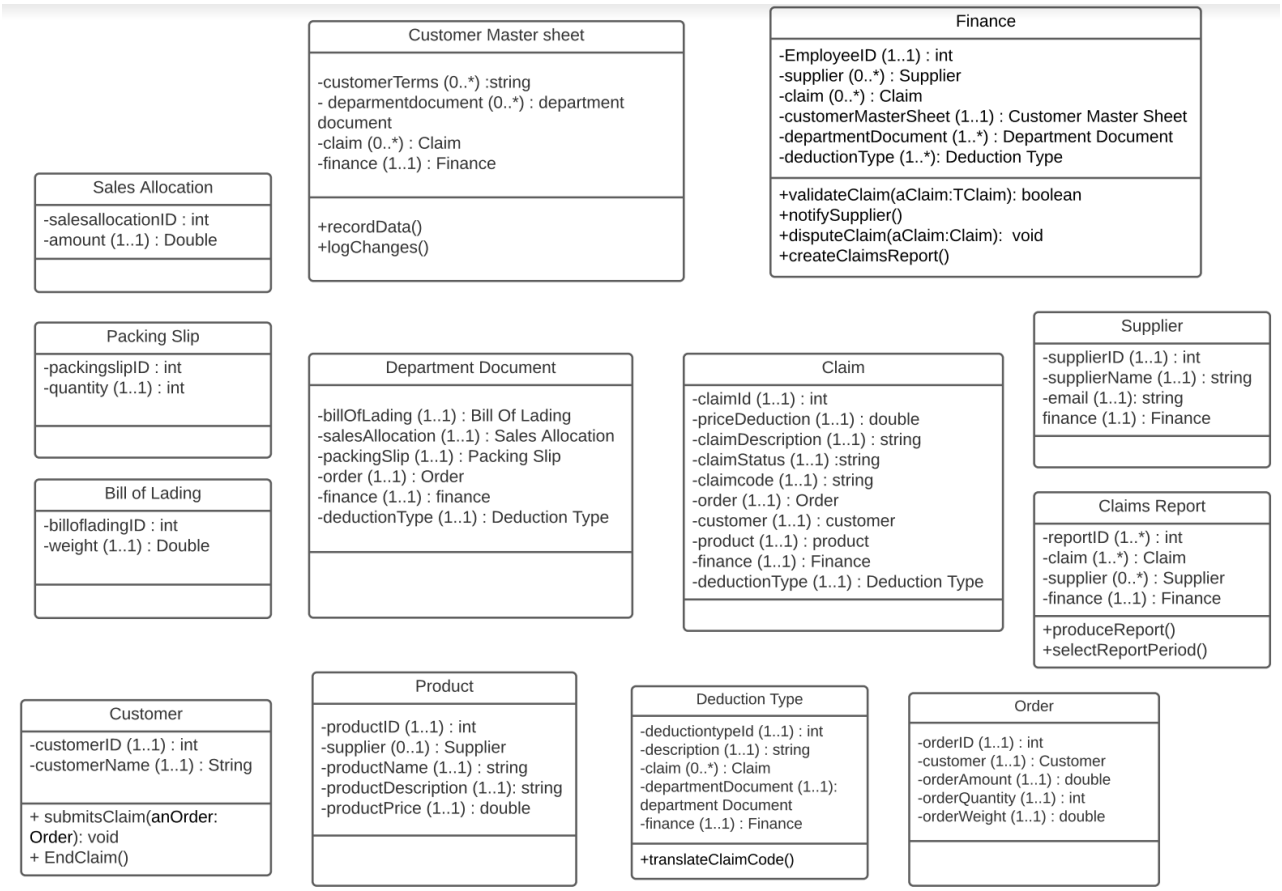
```
            MySqlCommand cmd = new MySqlCommand("UPDATE Claim SET
Claim_Notes = '" + Endnotetxt.Text + "' WHERE Claim_ID ='" + Endtxt.Text +
"'", dbConn);
            dbConn.Open();
            cmd.ExecuteNonQuery();
            dbConn.Close();

            load_data();

            MessageBox.Show("Claim " + Endtxt.Text + " has been resolved.");
            Endtxt.Text = "##########";
            Endnotetxt.Text = "Please add a note here.";
            Endtxt.ForeColor = Color.DarkGray;
            Endnotetxt.ForeColor = Color.DarkGray;

        }
        else
        {
            MessageBox.Show("Error!" + "\n" + "Either the data entered is invalid
or required data fields have been left empty.");
            Endtxt.Text = "##########";
            Endtxt.ForeColor = Color.DarkGray;
            Endnotetxt.Text = "Please add a note here.";
            Endnotetxt.ForeColor = Color.DarkGray;
        }

    }
```

Misc.Notes:

Method Name: EndClaim	**Class Name:** Finance	**ID:** 100
Clients (Consumers): Claim		
Associated Use Cases: EndClaim		
Description of Responsibilities: Implement the necessary behavior to verify the claims received by the finance department to evaluate if the claims received are legitimate or not.		
Arguments Received: aClaim:Claim		

Type of Value Returned: boolean
Pre-Conditions: aClaim:Claim > $100
Post-Conditions: aClaimStatus:Claim = valid aClaimStatus:Claim = invalid aClaimStatus:Claim = in progress

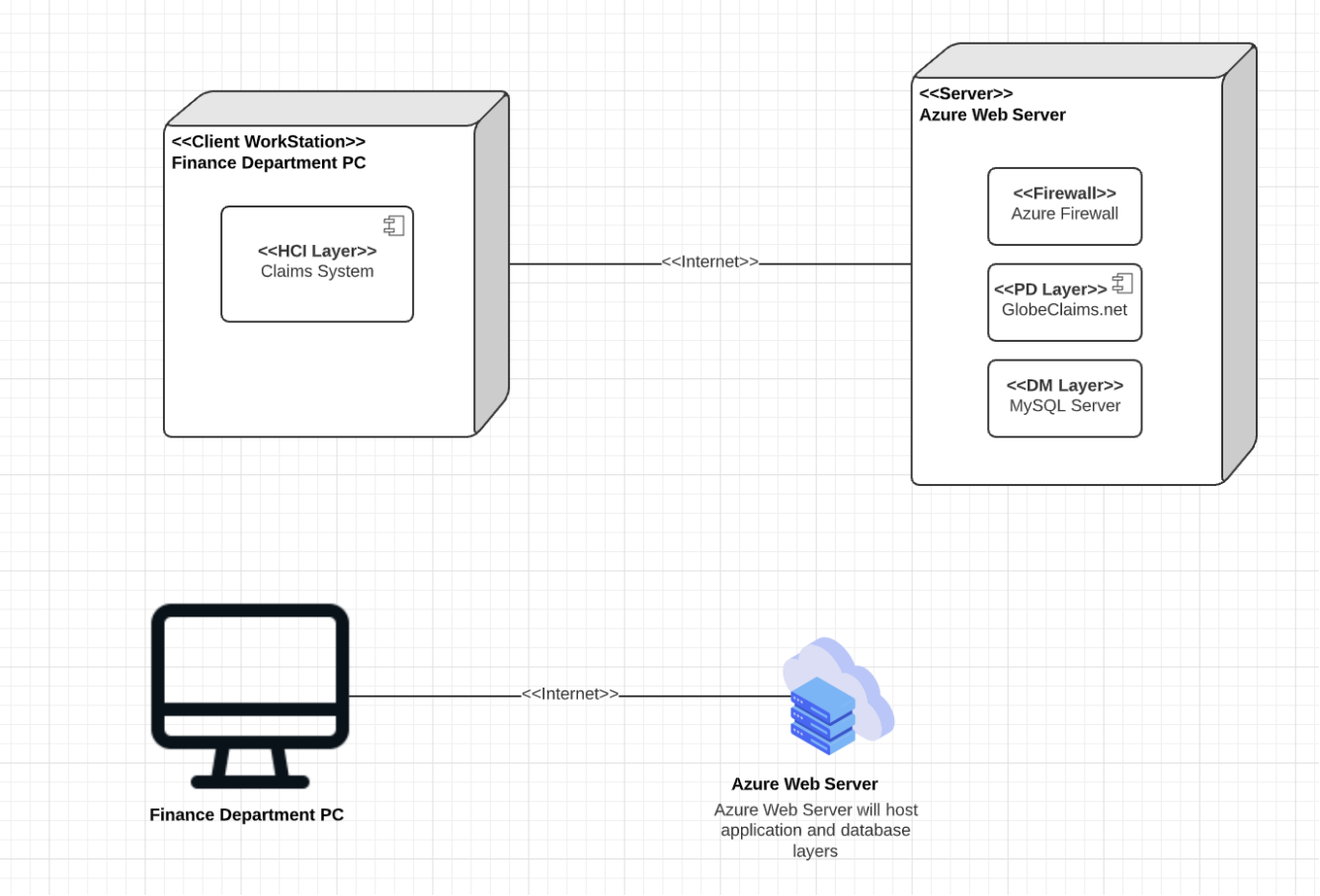
Implementation Diagram 2.0



HCI Design

For this project, we used the digital product design platform Figma, to design the HCI layer for this application. A demonstration of the interface can be found here: **https://www.figma.com/file/tr5rnjDw964HmxYU8vysAJ/HCI-Design?node-id=0%3A**

Deployment Diagram



Prototype

GitHub Link

https://github.com/Bhoemoth/BTM-495-Globe-Electric-Claim-Deduction-Prototype-

Data Management

For the data management portion of the prototype we decided to use mySQL because we already had experience using it in the past. It was easy for us to create the schema needed for the prototype as it mainly included data needed for the claim and claim deduction. The main function that we needed from SQL was the data set that would translate the claim code. When operating the prototype, the claim code entered in SQL would already exist because when a customer of Globe submits their claim code it will always be the same code for that specific deduction type. The idea was that once that claim code is entered, we would use sql to reference the dataset that has that specific code and then display the corresponding deduction type. Another benefit to using SQL is that we were able to connect it to visual studio while using C#. Our programming code revolved largely around running queries through mysql to display the data. For instance, when a new claim is submitted into the prototype, the data in my sql would be directly updated with the new claim and then that updated dataset is what would be displayed in the prototype. This was the same principle for the validating, disputing and end claim of the prototype where the data being displayed in the prototype is directly from mySQL. Any edits made to these data sets would instantly be updated synchronously between the prototype and SQL.

Physical Architecture

Deployment Recommendations

In order to ensure a smooth implementation of this program into Globe's current workflow, it is crucial that this application can seamlessly integrate into the current SharePoint and Dynamics ERP system. As Globe's current IT team is already familiar with Azure cloud services, it is highly recommended that this application be deployed using Azure web services in order to ensure compatibility and smooth integration. With Microsoft Azure, it will be possible to host the entire application, from the HCI layer to the application layer as well as the database layer all within the cloud. Therefore, we strongly recommend that Globe takes advantage of the cloud to offer unparalleled accessibility and customizability.

Hardware/Software

Because of the serverless nature of this application, there should be no additional requirement to procure additional hardware, as any device with Internet connectivity should be able to access the application.

Construction

This application was built using the Visual Studio development environment to develop in the C# language. As many team members were already familiar with this environment, the choice to use C# and the .NET framework was clear. The backend database was built using MySQL Server as the type of data required to process would benefit greatly from a relational database structure.

Installation and operation:

We wish to have our application be available online and remotely. As such, there will be no installation of additional hardware or software necessary. Everything will be available from a

web browser. Users will have a username and password to enter and will not need to install any application on their computer. After they have signed in to the application they will be greeted with a menu that will allow them to choose between validating a claim or disputing a claim providing supporting documents for them. The claims that appear in the system have already been translated. Once the documents have been attached relating to the claims the validation process is finished and the user can move on to the next one.

User testing and training

In order to ensure that the application is operating correctly there needs to be a testing phase. The testing phase will run for 5 days. It will be run parallel to existing operations. During this time every 10th claim submitted, users will manually check the claim in order to make sure it is correct. If an error is discovered users can contact us in order to update the system and correct the mistake. All previous claims will have to be checked. If no errors are discovered after 50 claims the system will be considered correct. In order to make sure overall efficiency remains higher than the status quo proper training will have to be done. During the training phase we will use teleconferencing software to provide live seminars and train the finance department on how to use the application. We will also have a forum made to answer any questions and provide an interactive learning experience. Furthermore, we will have weekly meetings set up for the first month in order to iron out any possible bugs in the application. During these meetings we will also discuss the efficacy of the application and whether or not it has satisfied all the conditions and needs listed earlier. If any modifications are required they will be made during that time frame.

Appendix

Team Management

Team Contract – Team 8

1 Meeting Coordinates

Following a discussion between the team members where everyone gave 2 to 3 preferences, the following meeting coordinates have been determined:

Virtual Location	Day	Time	Duration	Status
Zoom	Sunday	11 am	1 to 2 hours	Mandatory
Zoom	Monday	12 pm	1 to 2 hours	Mandatory
Zoom	Thursday	11 am	1 to 2 hours	Optional

2 Administrative Responsibilities

2.1 Team leader

The team leader will be a rotational task. Depending on the component of the project we are working on, the role of the team leader will be assumed by the person that is the most knowledgeable in that field. For example, when we will meet potential clients, the team leader role will be the team member that has the best public relation skills. The leader will be responsible for creating and maintaining an open and welcoming environment during team meetings. The role of the leader will be to maintain a productive working climate in the team during the meetings and support the cohesion between the team members. The leader will assign tasks at the end of every

meeting with mutual agreement among the team. The leader will make sure that the deliverables and deadlines are respected so that a colleague waiting for the part of another will not be penalized. A failure to fulfill his/her functions will result in:

- Offense 1-2 - Verbal warning on behalf of the team
- Offense 3-4 - Repeat offenses will lead to the leader being replaced and loss of marks on peer evaluation at the discretion of every individual team member

The team decided that we don't want a designated leader for specific time periods. We would rather appoint a person every week

2.2 Scribe (Secretary)

Alex Juneau will be our designated scribe throughout the whole semester. Before the meetings, the scribe will have to prepare the agenda to make sure that we can make the most out of every meeting. Its function will also be to manage the minutes and take detailed notes of what is said during the meetings and what the team has decided. At the end of a gathering, the secretary will send to the team members a summary of what has been discussed during the meeting. The scribe will be responsible for reminding every member of the deliverables for the next meeting/class. The scribe will document what was assigned to each member during the team meeting. A failure to fulfill his/her functions will result in:

- Offense 1-2 - Verbal warning on behalf of the team
- Offense 3-4 - Repeat offenses will lead to scribe being replaced

2.3 Members

The team members will be treated equally no matter their ethnicity, gender, or sexual orientation. When discussing a controversial subject, the team will proceed to a vote where the majority will win. Each team member will be given one vote in order to respect a democratic deliberation. It will be the responsibility of every team member to help create and maintain an

open and welcoming environment during team meetings. If they decide to sign it, the members will have to comply with what is written in the contract.

The team members are going to have to be prepared for every meeting they will attend. The team members also need to complete their tasks on time. Failure to adhere to the responsibilities and deadlines will result in:

- Offense 1-2 - Verbal warning on behalf of the team
- Offense 3-4 - Loss of marks on peer evaluation at the discretion of every individual team member

**The team will attempt to solve the issue internally before escalating it to the professor.

3 Specialization Responsibilities

Knowledge Area	Lead Team Member	Support Team Member
KA1 - Information Systems Analysis and Design	Sima Sukkar	Amritjot Singh Nagi
KA2 - Database Management	Fabrizio Capparelli	Steven Robinson
KA3 - Human-Computer Interaction (HCI - UI/UX)	Alex Juneau	Abigail Estrada
KA4 - Programming and Development	Amritjot Singh Nagi	Steven Robinson
KA5 - Project Management	Luis Aguilar	Alex Juneau

4 Performance at team meetings

4.1 Attendance and punctuality

The attendance at the meetings will be taken very seriously and is going to be shown in the minutes. A team member will need a very good reason to miss a meeting and the reason will have to be validated by the team leader. Team members will attend the meetings virtually over a video call. The team is required to be ready on time when a meeting starts so they will have to show up earlier than the scheduled time in order to be fully prepared. Anybody joining the meeting more than 15 minutes after the start of a meeting will be considered late. The team member must inform the team in advance if he/she will be joining the team meeting late. In the case of an absence, the team member will have to notify the team leader at least 48 hours prior to the meeting. The maximum number of absences allowed is 3. Absences without a valid reason will result in:

- Offense 1-2: Verbal warning on behalf of the team and completion of the assigned tasks
- Offense 3 or more: 10% deduction on peer evaluation at the discretion of every individual team member for every additional offense

4.2 Participation in discussions and task handling

The scribe will send the agenda 24 hours prior to the meetings and the team members will have to be prepared to discuss the agenda before showing up to the meeting. The team will work some components of the project in smaller groups and others in bigger groups so mutual respect will be required from all the members. The tasks and their weights are going to be equally divided between the team members. The members will be responsible to meet their deadlines and deliver quality work. Team members will be responsible for asking questions to the team and to have a mutual agreement with the team on what exactly needs to be done. The team will assess the overall quality of work once it has been completed on a weekly basis. Low quality work submitted by a team member will result in:

- Offense 1 or more: Verbal warning on behalf of the team and resubmission of work within 48 hours
- Repeat offenses will result in a loss of marks on peer evaluation at the discretion of every individual team member.

**The team will attempt to solve the issue internally before escalating it to the professor.

5 Medium of Communication

The following mediums of communication will be used to ensure effective communication:

- Facebook Messenger/Microsoft Teams- General chat and discussions
- Moodle Forum/Microsoft Teams - Discuss the confusions we have as a team with the professor
- Zoom/Microsoft Teams - Remote meetings
- Google Drive - share various files and deliverables related to the project
- Google Docs/ Google Sheets - Collaborate to work on deliverables

6 Technology Tools

- Modeling tool: Lucid Chart
- Programming tool: Visual Studio and MySql
- Implementation platform: Microsoft SharePoint

7 Guidelines to improve meeting effectiveness.

- A peer evaluation system on Moodle will be created to provide feedback to the other team members and report possible laziness.
- The team leader will be responsible to motivate the team at every meeting so that the team stays focused throughout the project.
- A voting system will be established to solve conflicts between the team members. Every team member would be expected to accept the decision agreed upon by the majority of the team.
- Rude, criticizing, and domineering behavior will have certain consequences.
- An open environment will be created during team meetings where new creative ideas will be welcomed.
- Every 3 meetings, the team will have a discussion over how the team is performing and if anyone has any concerns or comments.

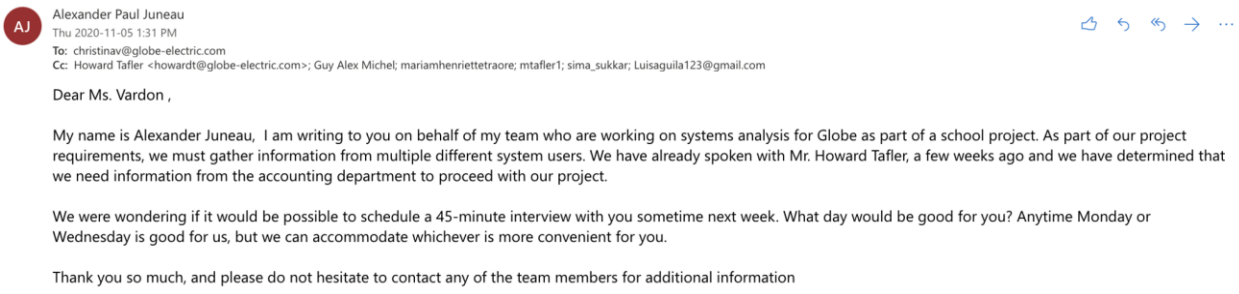
Date of Signature

Name	ID#	Signature
Alexander Juneau	40085739	AJ
Luis Aguilar	40045618	LA
Steven Robinson	40093733	SR
Fabrizio Capparelli	40097997	FC
Sima Sukkar	40090653	SS
Amritjot Singh Nagi	40037989	ASN
Abigail Estrada	40077578	AE

Software Used

1. Visual Studio
2. MySQL Database
3. Figma
4. LucidChart
5. Google Docs
6. Google Slides

Client Communication



AJ

Alexander Paul Juneau
Thu 2020-11-05 1:31 PM
To: christinav@globe-electric.com
Cc: Howard Tafler <howardt@globe-electric.com>; Guy Alex Michel; mariamhenriettetraore; mtafler1; sima_sukkar; Luisaguila123@gmail.com

Dear Ms. Vardon ,

My name is Alexander Juneau, I am writing to you on behalf of my team who are working on systems analysis for Globe as part of a school project. As part of our project requirements, we must gather information from multiple different system users. We have already spoken with Mr. Howard Tafler, a few weeks ago and we have determined that we need information from the accounting department to proceed with our project.

We were wondering if it would be possible to schedule a 45-minute interview with you sometime next week. What day would be good for you? Anytime Monday or Wednesday is good for us, but we can accommodate whichever is more convenient for you.

Thank you so much, and please do not hesitate to contact any of the team members for additional information

Meeting Minutes

Week#	Week of	Total Duration	Location	Leader	Attendance	Scribe
1	Jan 18th 2021	2 hours	Zoom	Steven Robinson	All members	Fabrizio Capparelli
2	Jan 25th 2021	3 hours	Zoom	Alexander Juneau	All members	Luis Aguilar
3	Feb 1st 2021	5 hours	Zoom	Sima Sukkar	All members	Abigail Estrada
4	Feb 8th 2021	4.5 hours	Zoom	Steven Robinson	All members	Fabrizio Capparelli
5	Feb 15th 2021	5.5 hours	Zoom	Amrit Singh	All members	Alexander Juneau
6	Feb 22nd 2021	6 hours	Zoom	Luis Aguilar	All members	Sima Sukkar
7	March 8th 2021	4 hours	Zoom	Abigail Estrada	All members	Steven Robinson
8	March 15th 2021	7 hours	Zoom	Fabrizio Capparelli	All members	Amrit Singh
9	March 22nd 2021	5 hours	Zoom	Alexander Juneau	All members	Luis Aguilar
10	March 29th 2021	6 hours	Zoom	Sima Sukkar	All members	Abigail Estrada
11	April 12th 2021	3 hours	Zoom	Steven Robinson	All members	Fabrizio Capparelli
12	April 19th 2021	4 hours	Zoom	Amrit Singh	All members	Alexander Juneau

Week #	Agenda
1	I) Team introduction II) Discussion about clients III) Decisions on roles and responsibilities (specialists' groups) IIII) Project proposal and team contract tasks
2	I) Discuss use-cases and activity diagrams II) Split tasks
3	I) Discuss structural models II) Discuss functional modelling workshop takeaways III) Split tasks for use-cases and activity diagrams
4	I) Discuss what happened in the status meeting with prof II) Split tasks for upcoming submissions III) Look over sequence diagrams
5	I) CRC Cards II) Class diagrams roles and responsibilities III) Discuss corrections needed based on prof's comments
6	I) Split Behavioural modelling tasks II) Discuss workshop takeaways III)Prototype platform discussion
7	I) HCI design discussion and Q&A II) HCI roles allocation III) HCI workshop IIII) Team Feedback reflection was discussed.
8	I)Team Feedback reflection was addressed again and solutions we're provided. II) Package Diagram and Class Method designs we're reviewed before submitting. III) Article Reviews q&a discussion between specialists III) Prototype follow up discussion
9	I) Physical Architecture Design ideas were discussed II) Expectations for the design we're also expressed
10	I) Deployment Diagram discussion II) Prototype follow up III) Detecting coding error for prototype
11	I) Finalize Prototype II) Splitting tasks remaining III)Assign roles for final report review
12	I) Prepare PowerPoint II) Practice presentation III) Review Report

Previous Iterations

Use Case Diagram 1.0