



Vanier College

Computer Science and Technology Department

System Development

420-436-VA

Deliverable 3

Orange Team

Jericho Nieva

Philip Dubé

Daniel Levitin (Team Leader)

Client: FavoriteDesign stretch ceilings

3/9/2023

We, the orange team, certify that this project is our own work.

Code learned and used from our eCommerce class will be used for features of our project.

I, Philip Dubé, student ID#2145451, confirm that I have contributed to this deliverable.



I, Daniel Levitin, student ID# 2162749, confirm that I have contributed to this deliverable.



I, Jericho Nieva, student ID# 6161653, confirm that I have contributed to this deliverable.



Table of contents

Table of contents	2
Executive Overview	3
Summary of the Client	4
The Business Problem	5
Narrative description	6
Appendix I - Flowchart	7
Appendix II - UML	9
Appendix III - Use Cases	15
Appendix IV - Class Diagram	33
Appendix V - Client's Tables	35
Project Records	35
Address Records	36
Mediography	36

Executive Overview

Although our client's business problem remains unchanged, we have devised solutions to address their challenge. To solve the problem of our client, which is manually writing down information on paper, we are creating a database application where he will be able to manipulate his records in ways that would be impossible in a notebook.

This document outlines the flow and the procedures that need to be followed by the admin to ensure a smooth and efficient use of the website. It highlights in detail the processes such as; logging into the application, adding, editing and deleting records, like projects, addresses and expenses.

Following these processes, there are several appendices that provide critical documentation for the technical aspect of this project. Appendix 1 includes the system diagram for the entire information system. Appendix 2 contains the UML diagram, while appendix 3 contains the filled out use case templates that represent use cases found in the system diagram. Furthermore, right before appendix 5, which contains the already existing tables and forms used by our client, there is the 4th appendix, which includes the class diagram for the entire system.

Summary of the Client

Alexander Levitin, Daniel's dad, is a man from Ukraine who studied mechanical engineering in University, and eventually went into renovation, and more specifically stretch ceilings, which he temporarily went back to school for with the other owner to get proper certification. He has the computer knowledge that an average middle-aged person has, meaning that he knows the basics when it comes to navigating computers, phones, and other electronic devices. However, he doesn't have much knowledge when it comes to coding and when it comes to more advanced stuff such as the developer tool for websites.

Alexander Garbovnik, the other owner and a good friend of Daniel's dad, studied music and plays guitar and piano. He eventually went into renovations, and along with the other owner, went back to school temporarily to get proper certification for making stretched ceilings. He has similar knowledge in terms of technology as Alexander Levitin.

The Business Problem

Our client has a problem with how they keep a record of their clients, as they currently are doing physically through paper and pencil/pen. This is inefficient for adding/removing/editing information, looking for a specific record, and doesn't allow to filter the records, at least not in a scalable way. It also makes it unrealistic to create a backup, which is an important risk to consider for a business that plans to exist for a long time. Furthermore, sending data or records to someone would involve them either recopying them, scanning them with a printer, or taking pictures, so the communication between the two owners isn't very efficient. With an online Database Management System, communication about certain clients becomes very easy, convenient, and the owner who doesn't currently have the booklet of records doesn't need to wait for the other owner to send it.

Another problem encountered by our client is that they don't have a landing page/poster for their different services that they offer, which makes it difficult to advertise in newspapers, online ads, etc. So, one solution is to create a nice and compact landing page, where clients get all of the important information, while also getting sold on the product as much as possible with the limited words.

Narrative description

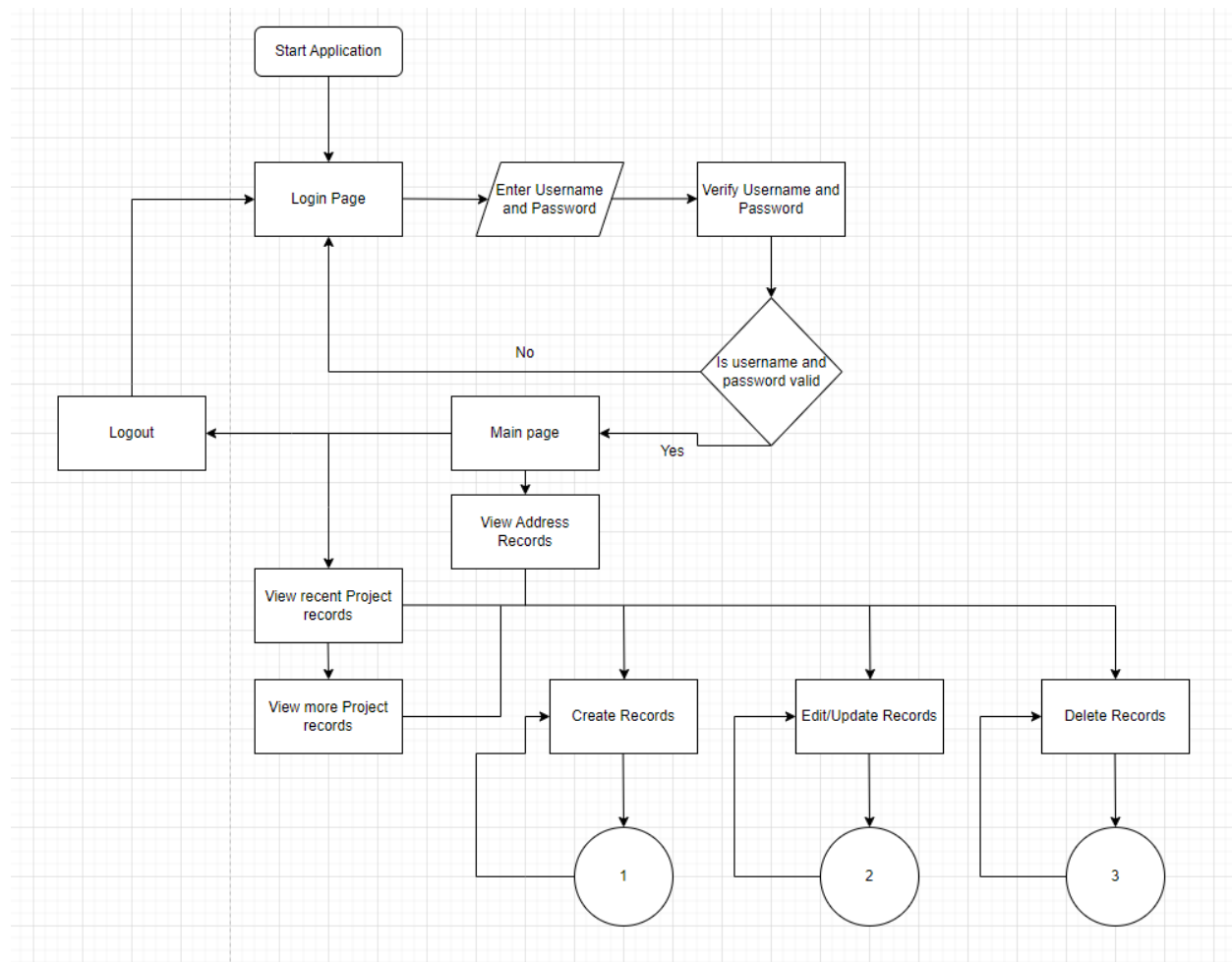
Information is written manually on paper by our client, the owner of the business Favorite Design. Records are recorded into two different tables which have different uses and purposes.

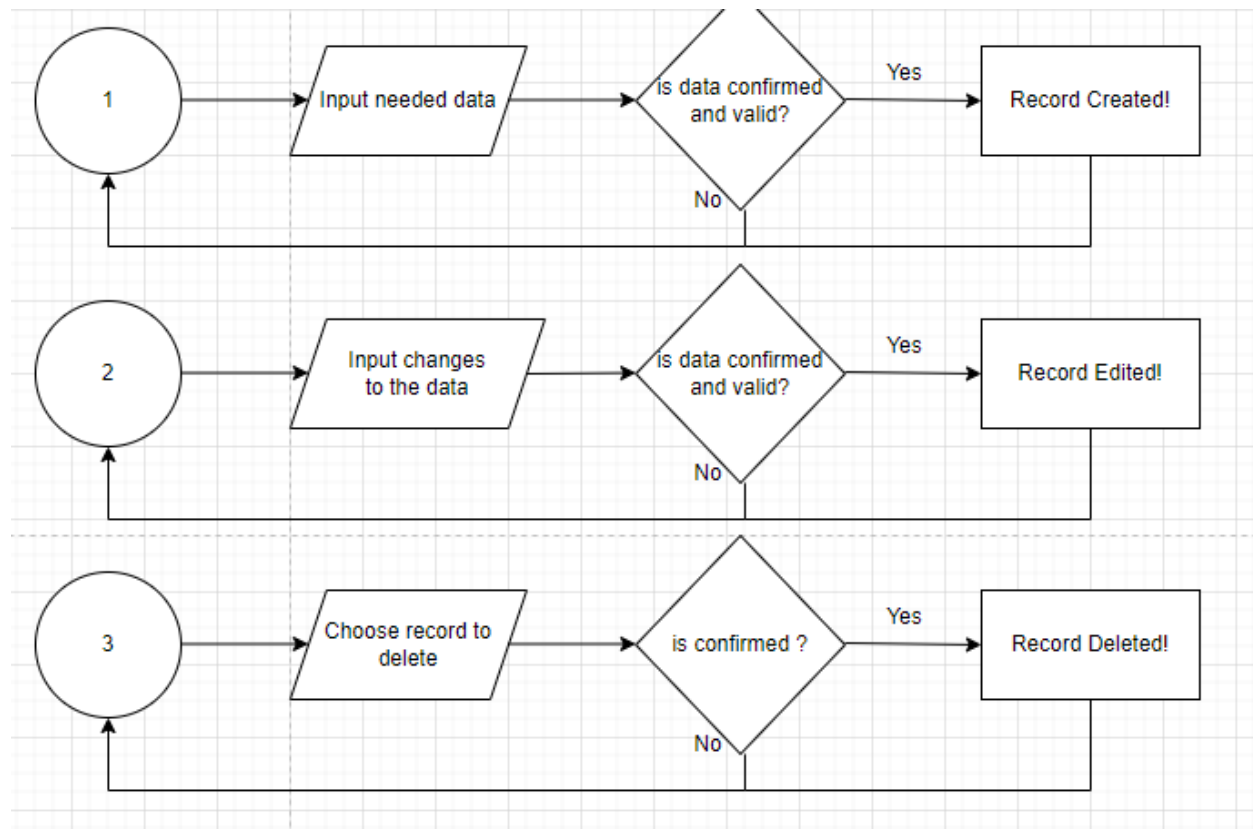
Firstly, one of the tables is used for any information relating to money. Records can either be relating to suppliers or clients. Information regarding clients includes their names, details about the project, details about the payments like the date and the method (cash/check/e-transfer), balances, exhausts/vents for bathrooms and any extra costs. On the other hand, records regarding suppliers include their names, the purchases and expenses.

Secondly, the other table is used for information regarding his clients' addresses and trips. Information recorded in this table includes the type of service (estimation/installation/other services, usually for a previous client such as leakage or light changes), the name of the client, the start and end date of the service, as well as their address and the distance in kilometer for a 2-way trip.

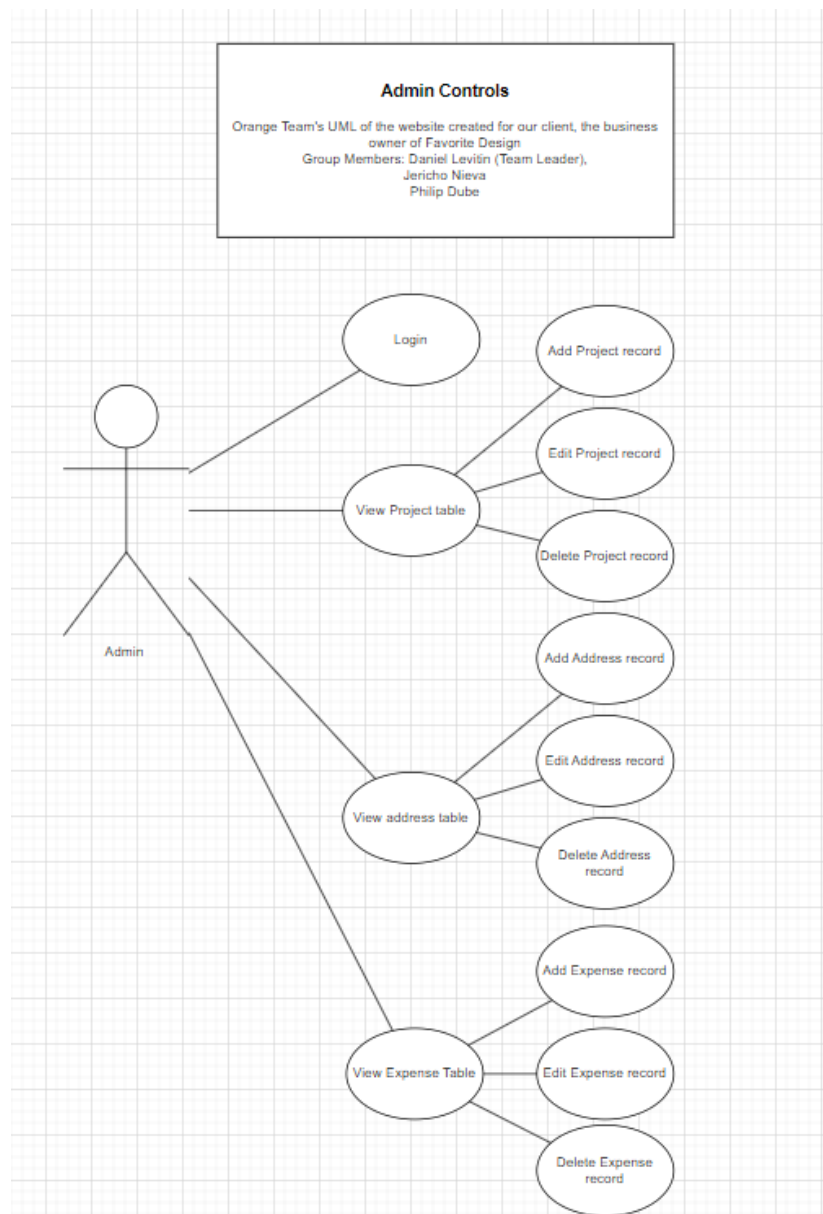
Part of running a stretched ceiling business is ensuring information is well-recorded and properly organized. The owner stores records in an easily accessible way through paper, which is his preferred method. His tables contain certain pieces of information which tailor to their use cases. For example, storing anything relating to money in the first table and anything relating to trips and addresses in the second one.

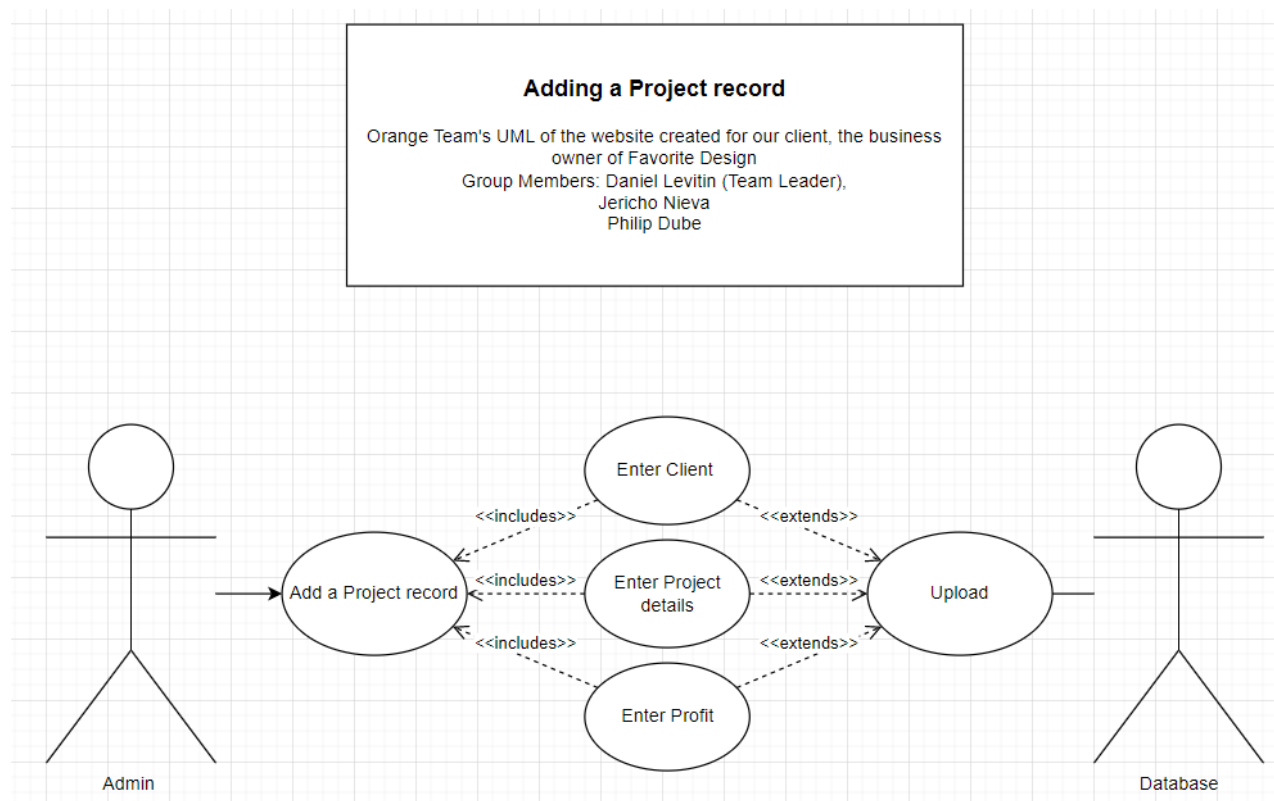
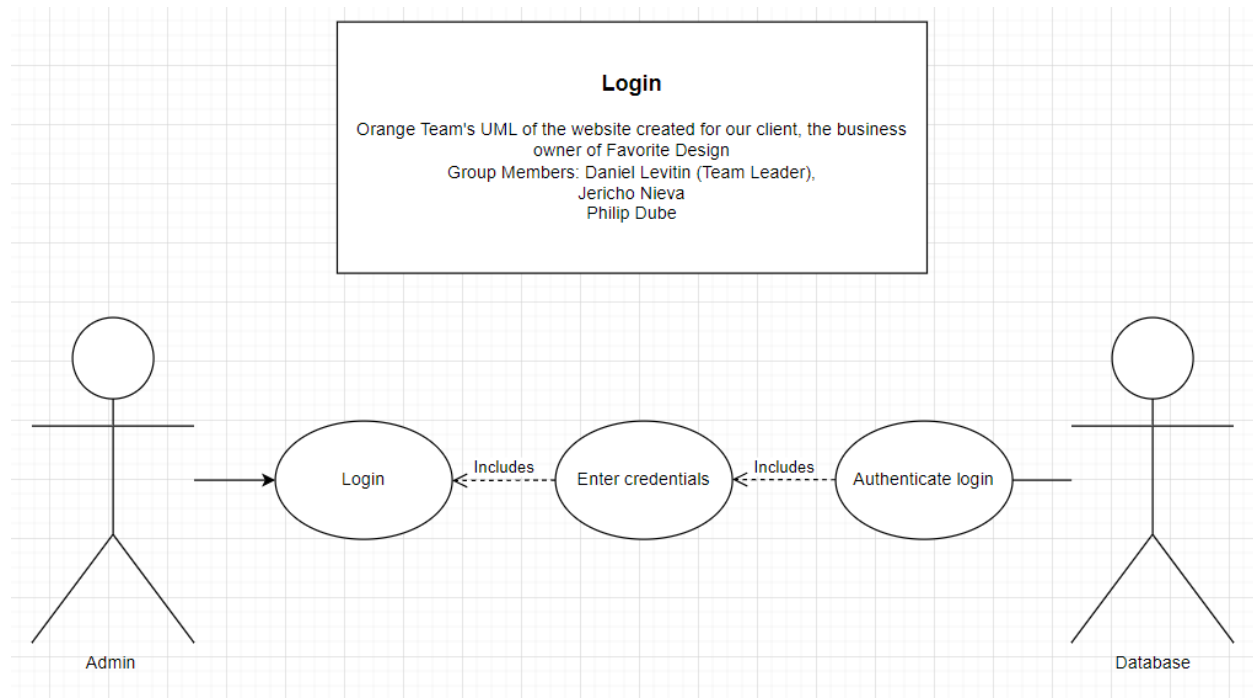
Appendix I - Flowchart

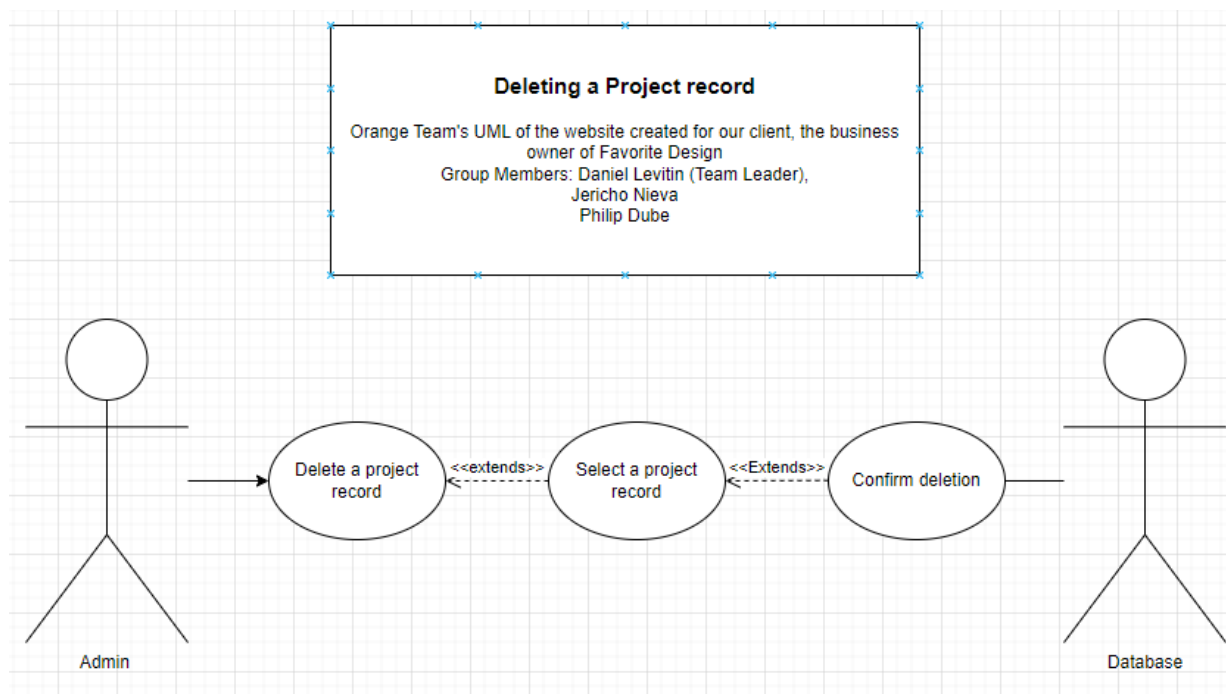
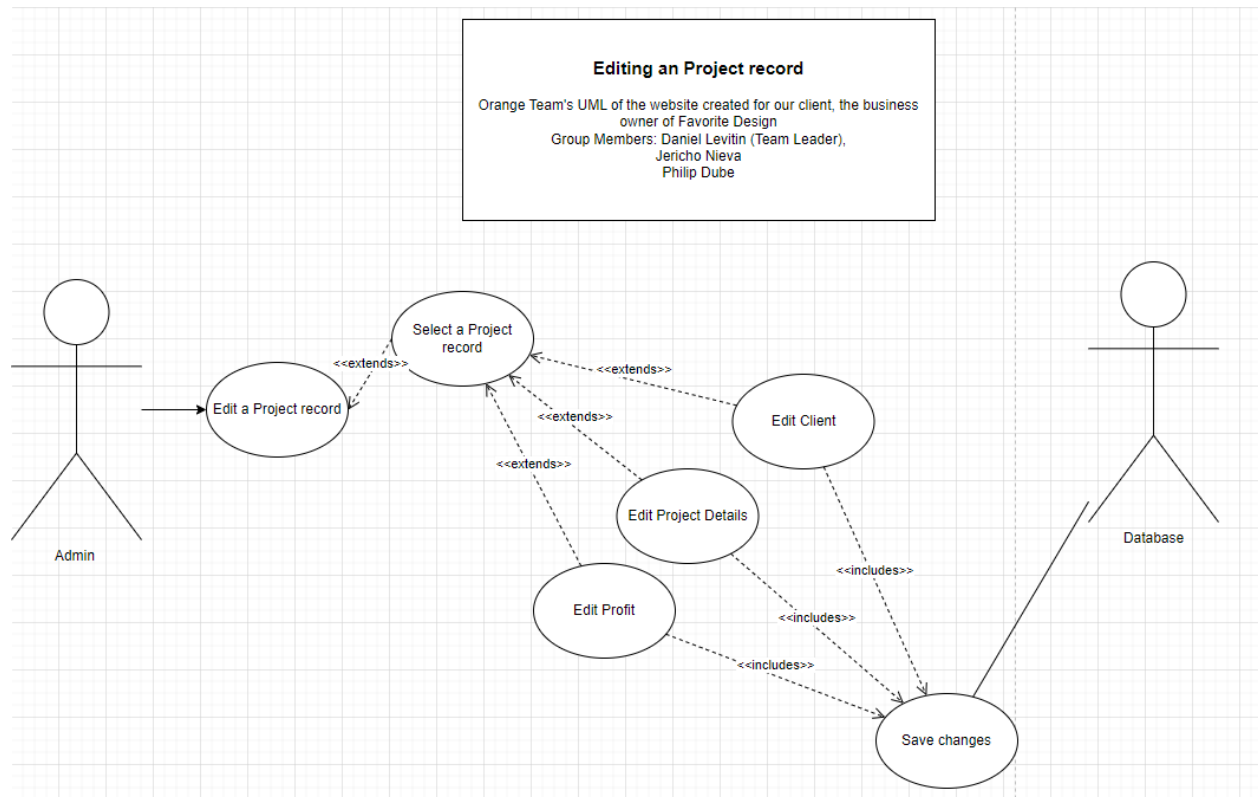


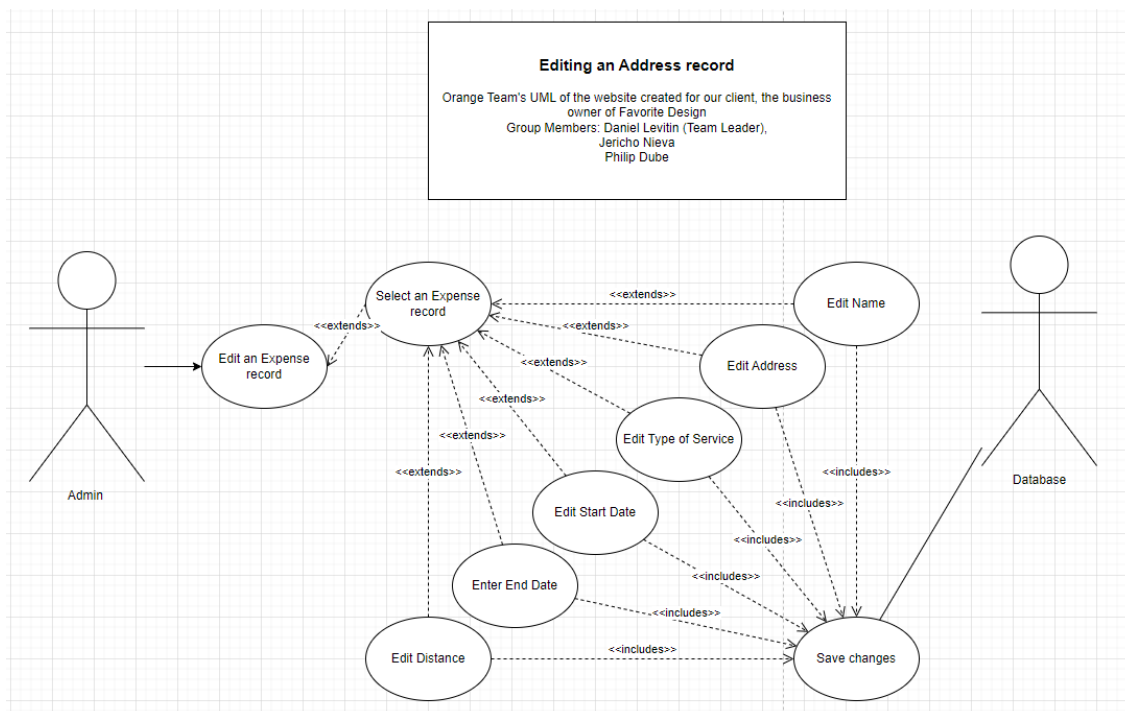
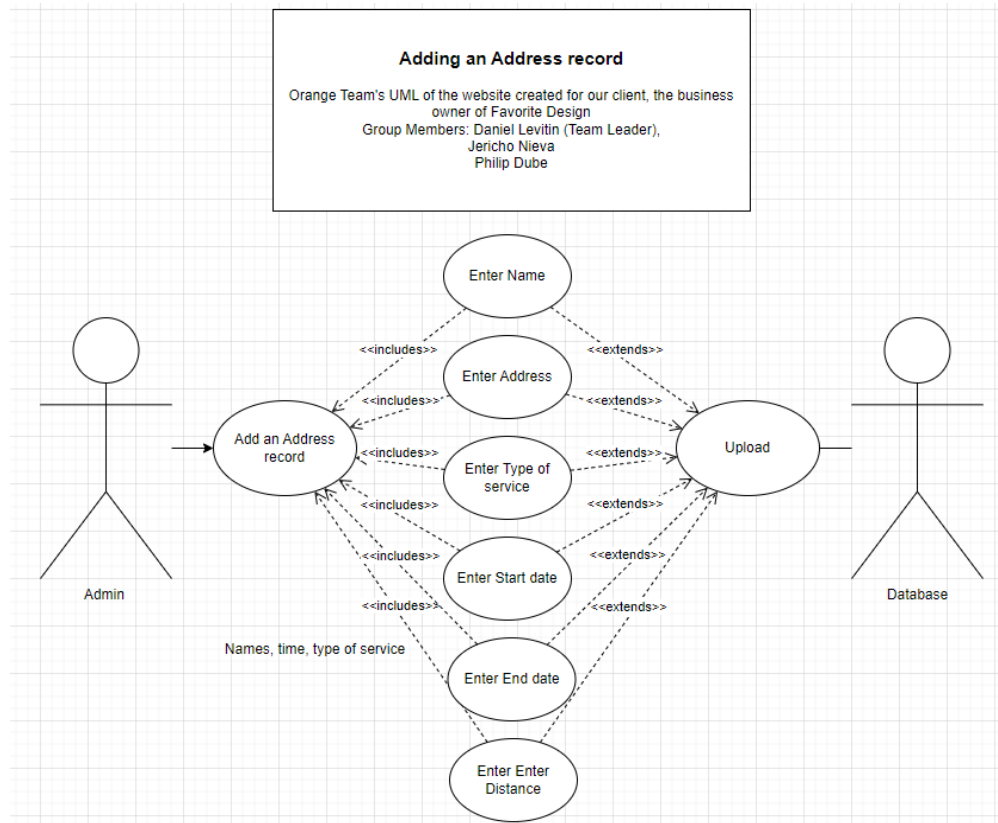


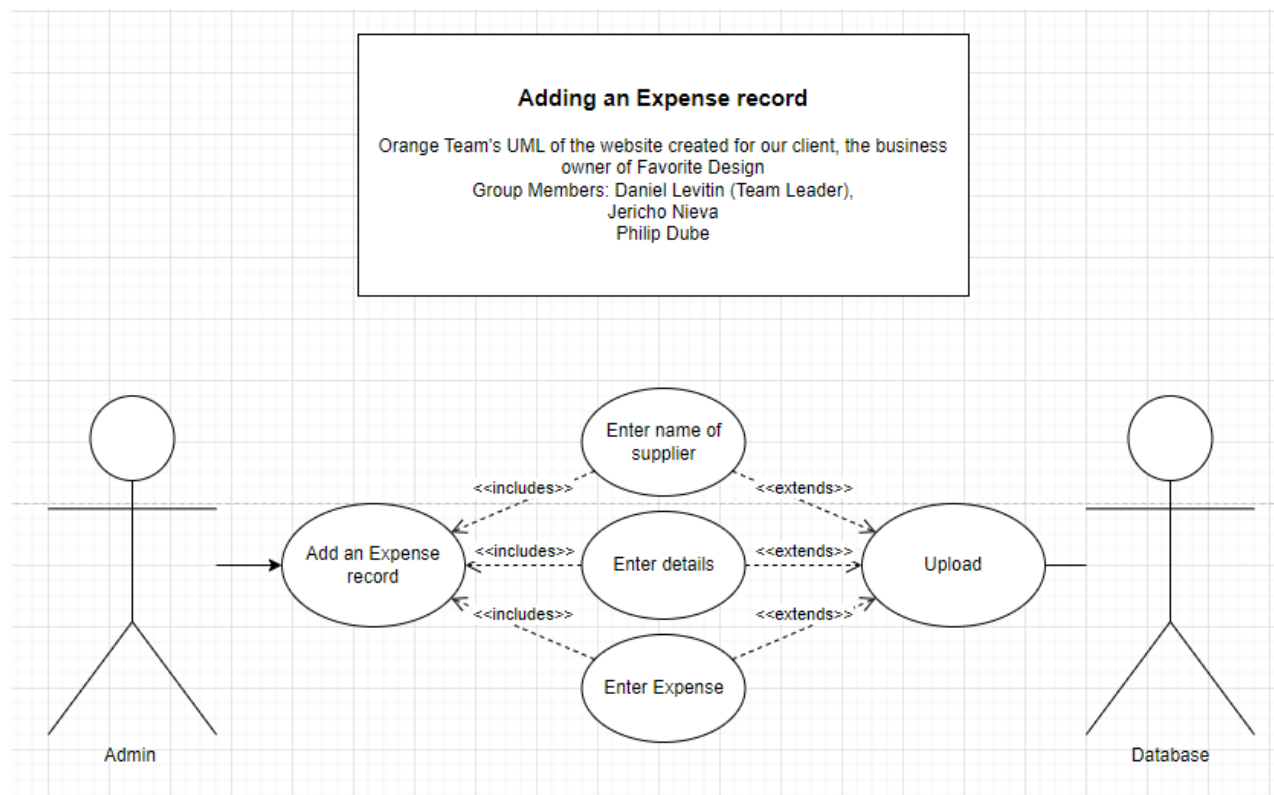
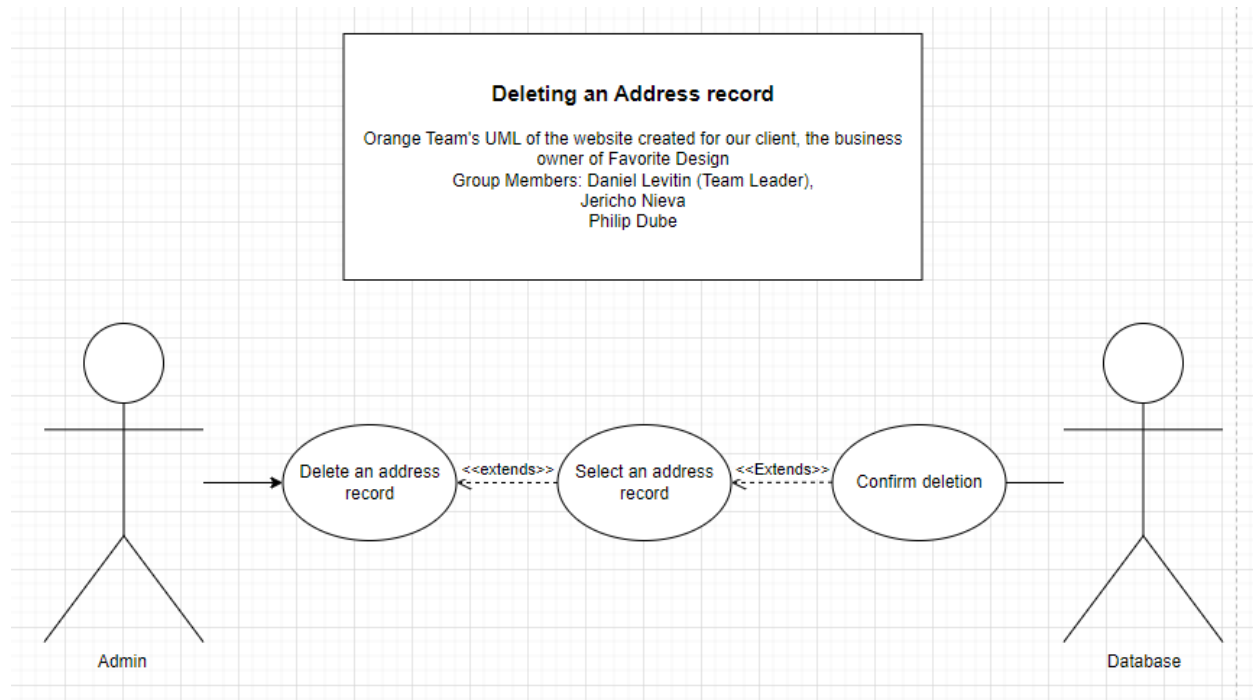
Appendix II - UML

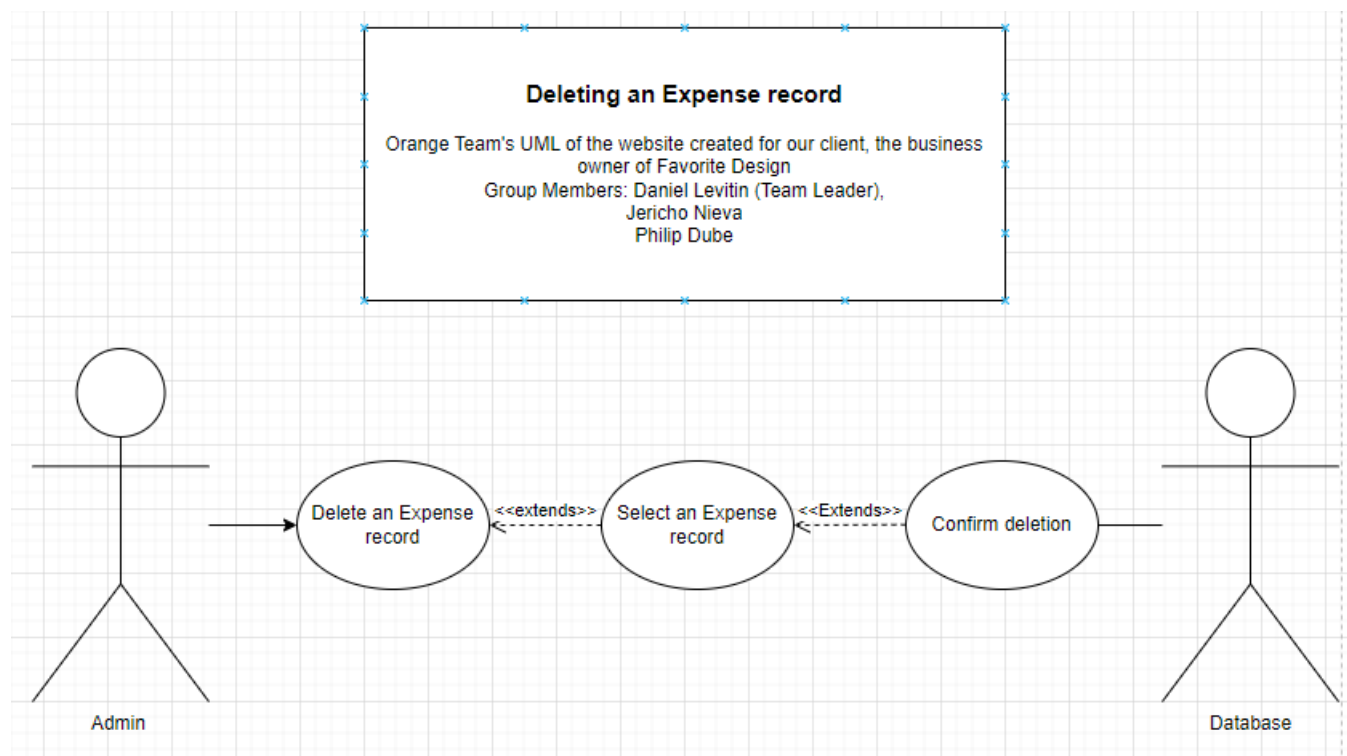
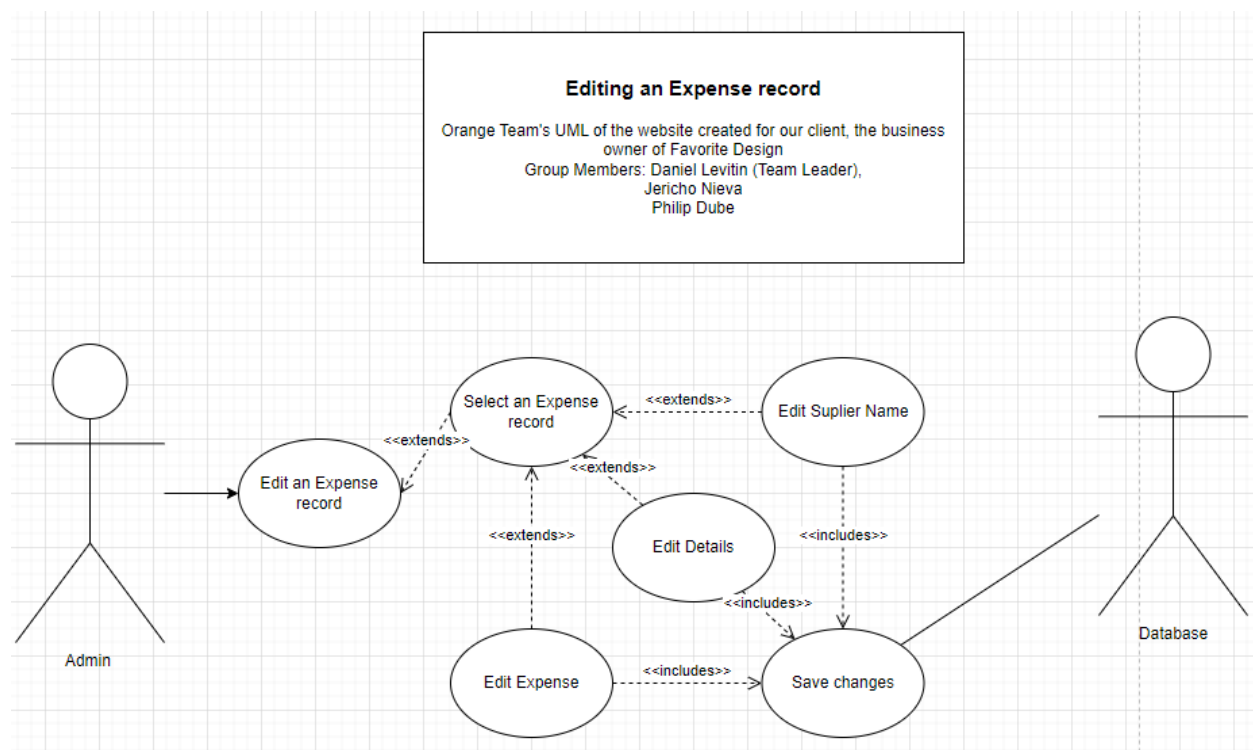












Appendix III - Use Cases

Use Case ID	FDH-AppStart		
Use Case Name	Starting Application		
Created By:	Daniel Levitin	Last Updated By:	Daniel Levitin
Date Created:	03/08/2023	Last Revision Date:	03/09/2023
Actors:	Owners of FavoriteDesign Hackers		
Description:	This use case is for starting the website		
Trigger:	User starts the website		
Preconditions:	1- User accesses the website		
PostConditions	1- User stays on the website		
Normal Flow	1. Owner opens the website 2. Owner is properly connected to the required network 3. The login page appears and the system starts as intended		

Alternative Flows	<ol style="list-style-type: none">1. Owner/hacker tries changing the link to access a login-required view of the website2. Website redirects the user to the login page
Exceptions:	<p>2a. If owner isn't properly connected to the required network</p> <ol style="list-style-type: none">1. Error appears on screen2. Give instructions on fixing network issues
Includes:	N/A
Frequency of Use:	Upon every use of the software
Special Requirements:	N/A
Assumptions:	The user knows the link to the website
Notes and Issues:	N/A

Use Case ID	FDH-Login		
Use Case Name	Logging In		
Created By:	Daniel Levitin	Last Updated By:	N/A
Date Created:	03/09/2023	Last Revision Date:	N/A
Actors:	Owners of FavoriteDesign Hackers		
Description:	This use case is for logging into the account		
Trigger:	User enters information		
Preconditions:	1- User successfully accessed the website		
PostConditions	1- User doesn't leave their session		
Normal Flow	1. Owner enters login information 2. Home page is displayed 3. Owner has all permissions		
Alternative Flows	1. User enters wrong login information 2. User gets redirected to the login page and an error message pops up		

Exceptions:	2a. If database connection has an issue 1. Error appears on screen 2. Give instructions on fixing database connectivity
Includes:	FDH-AppStart
Frequency of Use:	Upon every use of the software
Special Requirements:	N/A
Assumptions:	The user knows or has access to the login information
Notes and Issues:	Hacker getting through this step is to be avoided at ALL cost

Use Case ID	FD-Logout		
Use Case Name	Logging Out		
Created By:	Daniel Levitin	Last Updated By:	N/A
Date Created:	03/09/2023	Last Revision Date:	N/A
Actors:	Owners of FavoriteDesign		
Description:	This use case is for logging out of the account		
Trigger:	User presses the "Logout" button		
Preconditions:	1- User was previously logged in		
PostConditions	N/A		
Normal Flow	1. Owner presses "Logout" button 2. Owner confirms logout		
Alternative Flows	1. Owner presses "Logout" button 2. Owner cancels the logout 3. Owner keeps using the application		
Exceptions:			

Includes:	FD-HomePage FD-FullHistory FD-Trips
Frequency of Use:	Upon every use of the software
Special Requirements:	N/A
Assumptions:	The owner wants to logout after every session for security
Notes and Issues:	N/A

Use Case ID	FD-HomePage		
Use Case Name	Home Page Functionality		
Created By:	Daniel Levitin	Last Updated By:	N/A
Date Created:	03/09/2023	Last Revision Date:	N/A
Actors:	Owners of FavoriteDesign		
Description:	This use case is for accessing the most needed data		
Trigger:	User successfully logged in		
Preconditions:	1- User entered the correct login information		
PostConditions	1- User doesn't log out		
Normal Flow	1. Owner looks at recent client/supplier records 2. Owner scrolls 3. Owner presses "Load More" and looks at more records 4. Owner logs out		
Alternative Flows	a) 1. Owner presses "Add Record" 2. Owner enters needed information 3. Owner presses "Create Record" 4. Owner logs out ----- b)		

	<ol style="list-style-type: none"> 1. Owner presses "Edit a Record" 2. Owner chooses a record to edit 3. Owner enters new updated information 4. Owner presses "Edit Record" 5. Owner logs out <hr/> <p>c)</p> <ol style="list-style-type: none"> 1. Owner presses "Delete a Record" 2. Owner chooses record to delete 3. Owner presses "Confirm and Delete" 4. Owner logs out
Exceptions:	<p>Global) If database connection has an issue</p> <ol style="list-style-type: none"> 4. Error appears on screen 5. Give instructions on fixing database connectivity
Includes:	FD-FullHistory, FD-Trips, FD-Logout, FD-AddRecord, FD-EditRecord, FD-DeleteRecord
Frequency of Use:	Most of the software's purpose
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A

Use Case ID	FD-FullHistory		
Use Case Name	Full Page Functionality		
Created By:	Daniel Levitin	Last Updated By:	N/A
Date Created:	03/09/2023	Last Revision Date:	N/A
Actors:	Owners of FavoriteDesign		
Description:	This use case is for accessing older records		
Trigger:	User Activated the "Full History" view		
Preconditions:	1- There are still more client/supplier records to be loaded		
PostConditions	1- User stays on the "Full History" view		
Normal Flow	1. Owner looks at older client/supplier records 2. Owner scrolls/filters/searches through the older records 4. Owner logs out		
Alternative Flows	a) 1. Owner presses "Add Record" 2. Owner enters needed information 3. Owner presses "Create Record" 4. Owner logs out -----		

	<p>b)</p> <ol style="list-style-type: none"> 1. Owner presses "Edit a Record" 2. Owner chooses a record to edit 3. Owner enters new updated information 4. Owner presses "Edit Record" 5. Owner logs out <hr/> <p>c)</p> <ol style="list-style-type: none"> 1. Owner presses "Delete a Record" 2. Owner chooses record to delete 3. Owner presses "Confirm and Delete" 4. Owner logs out
Exceptions:	<p>Global) If database connection has an issue</p> <ol style="list-style-type: none"> 6. Error appears on screen 7. Give instructions on fixing database connectivity
Includes:	FD-Trips, FD-HomePage, FD-Logout, FD-AddRecord, FD-EditRecord, FD-DeleteRecord
Frequency of Use:	Rarely
Special Requirements:	N/A
Assumptions:	That the owner wants to be able to modify records from this page
Notes and Issues:	N/A

Use Case ID	FD-Trips		
Use Case Name	Trips Page Functionality		
Created By:	Daniel Levitin	Last Updated By:	N/A
Date Created:	03/09/2023	Last Revision Date:	N/A
Actors:	Owners of FavoriteDesign		
Description:	This use case is for accessing the trips table		
Trigger:	User Activated the "Trips Table" view		
Preconditions:	1- There are trips that exist		
PostConditions	1- User stays on the "Trips Table" view		
Normal Flow	1. Owner looks at trip records 2. Owner scrolls through the records 3. Owner presses "Load More" and looks at more records 3. Owner logs out		
Alternative Flows	a) 5. Owner presses "Add Record" 6. Owner enters needed information 7. Owner presses "Create Record" 8. Owner logs out -----		

	<p>b)</p> <ol style="list-style-type: none"> 6. Owner presses "Edit a Record" 7. Owner chooses a record to edit 8. Owner enters new updated information 9. Owner presses "Edit Record" 10. Owner logs out <hr/> <p>c)</p> <ol style="list-style-type: none"> 5. Owner presses "Delete a Record" 6. Owner chooses record to delete 7. Owner presses "Confirm and Delete" 8. Owner logs out
Exceptions:	<p>Global) If database connection has an issue</p> <ol style="list-style-type: none"> 8. Error appears on screen 9. Give instructions on fixing database connectivity
Includes:	FD-HomePage, FD-Logout, FD-AddRecord, FD-EditRecord, FD-DeleteRecord
Frequency of Use:	Rarely
Special Requirements:	N/A
Assumptions:	That the owner wants to be able to modify records from this page
Notes and Issues:	N/A

Use Case ID	FD-AddRecord		
Use Case Name	Adding a Record		
Created By:	Daniel Levitin	Last Updated By:	N/A
Date Created:	03/09/2023	Last Revision Date:	N/A
Actors:	Owners of FavoriteDesign		
Description:	This use case is for adding a record to a table		
Trigger:	User clicked the "Add Record" button from one of the views including it		
Preconditions:	1- The database exists		
PostConditions	N/A		
Normal Flow	1. Owner presses "Add a Record" 2. Owner inputs the required information 3. Owner presses the "Add Record" button 3. Owner logs out		
Alternative Flows	1. Owner presses "Add Record" 2. Owner enters needed information 3. Owner presses "Cancel"		
Exceptions:	Global) If database connection has an issue		

	10. Error appears on screen 11. Give instructions on fixing database connectivity
Includes:	N/A
Frequency of Use:	Often, whenever a trip or client/supplier record needs to be added
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A

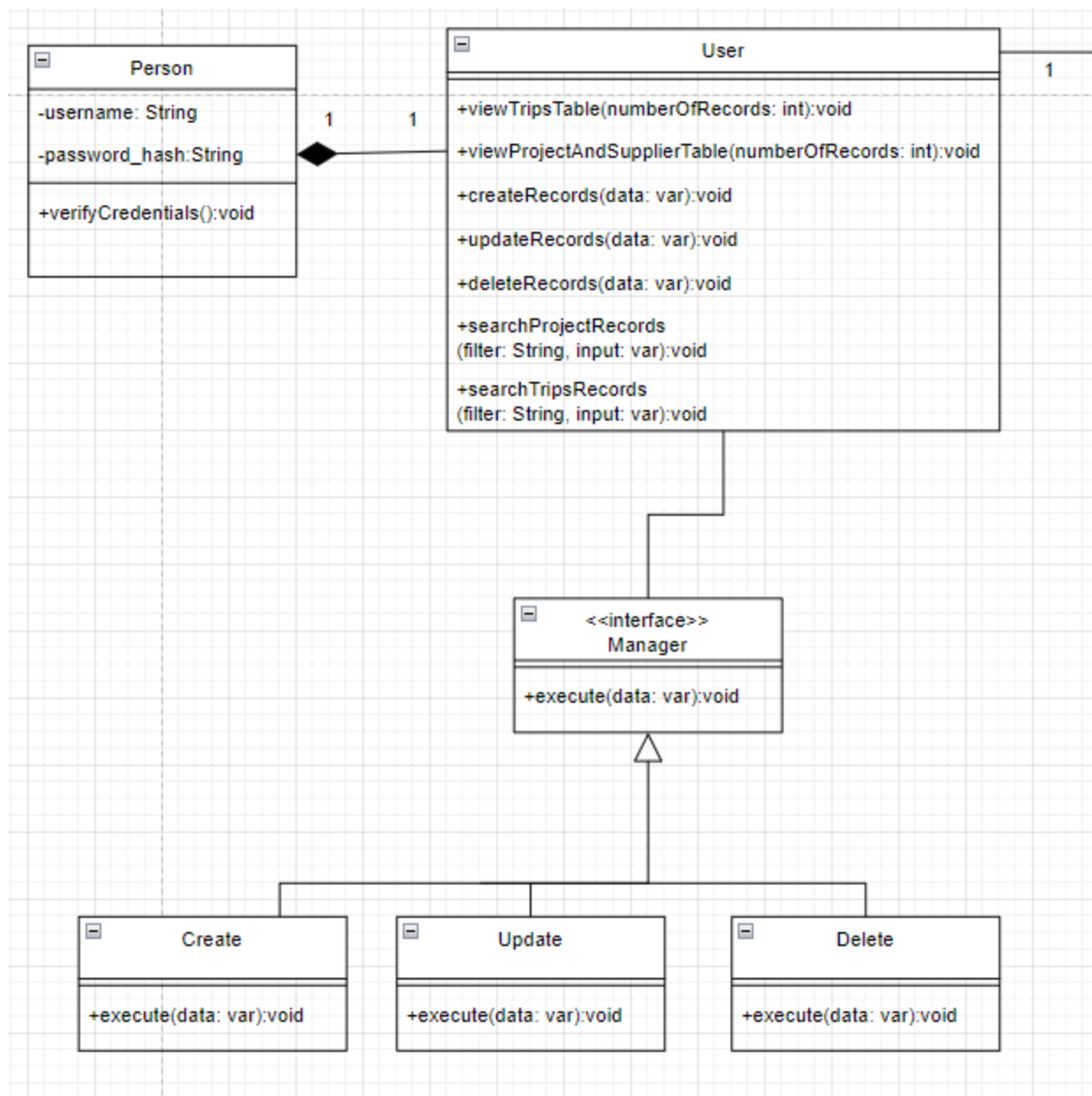
Use Case ID	FD-EditRecord		
Use Case Name	Editing a Record		
Created By:	Daniel Levitin	Last Updated By:	N/A
Date Created:	03/09/2023	Last Revision Date:	N/A
Actors:	Owners of FavoriteDesign		
Description:	This use case is for editing a record in a table		
Trigger:	User clicked the "Edit Record" button from one of the views including it		
Preconditions:	1- There are records in the database		
PostConditions	N/A		
Normal Flow	1. Owner presses "Edit a Record" 2. Owner chooses the record to edit 3. Owner inputs the required information 4. Owner presses the "Edit Record" button 5. Owner logs out		
Alternative Flows	1. Owner presses "Edit Record" 2. Owner chooses a record to edit 3. Owner enters needed information 4. Owner presses "Cancel"		

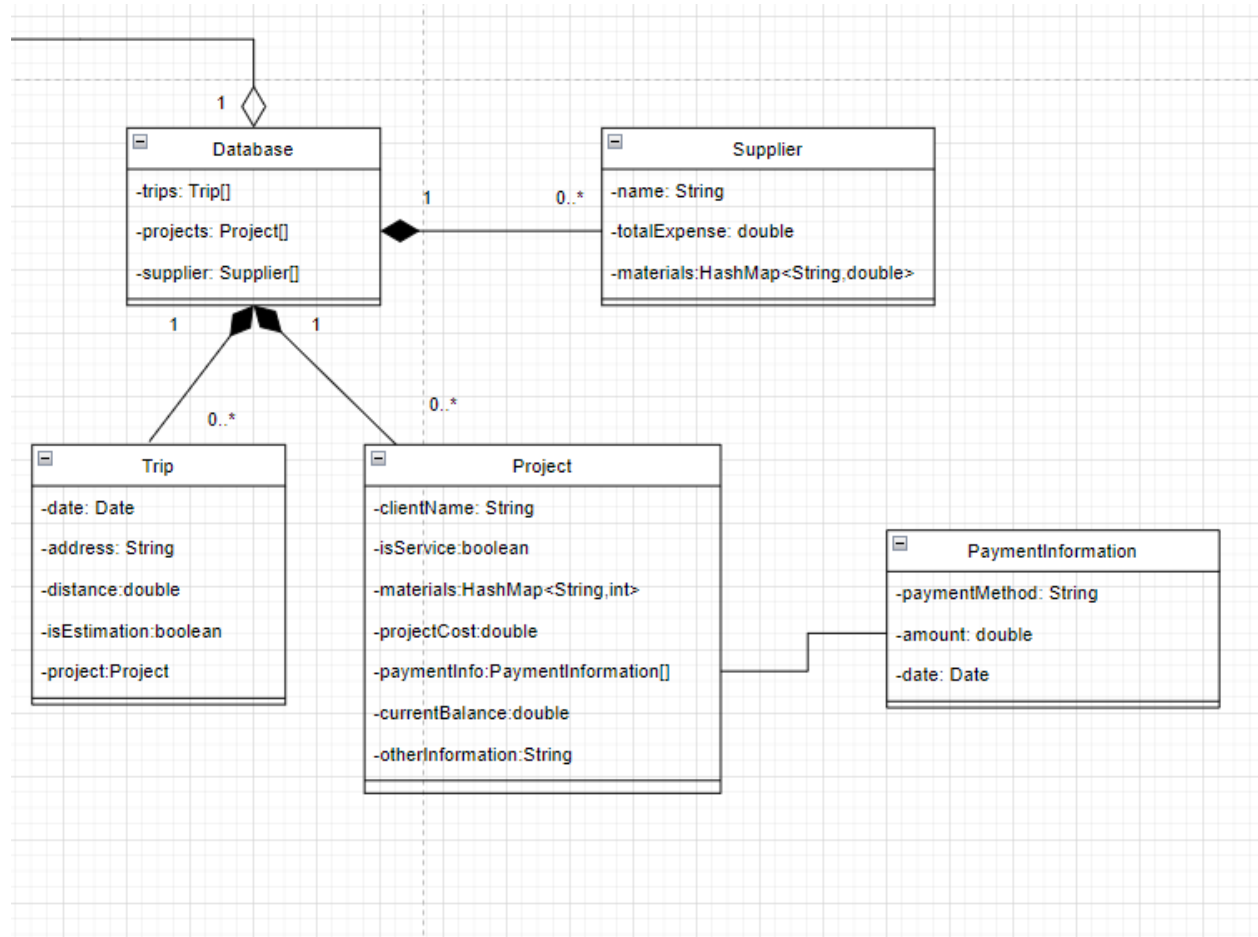
Exceptions:	Global) If database connection has an issue 12. Error appears on screen 13. Give instructions on fixing database connectivity
Includes:	N/A
Frequency of Use:	Somewhat often, used to update client balance
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A

Use Case ID	FD-DeleteRecord		
Use Case Name	Deleting a Record		
Created By:	Daniel Levitin	Last Updated By:	N/A
Date Created:	03/09/2023	Last Revision Date:	N/A
Actors:	Owners of FavoriteDesign		
Description:	This use case is for deleting a record in a table		
Trigger:	User clicked the "Delete Record" button from one of the views including it		
Preconditions:	1- There are records in the database		
PostConditions	N/A		
Normal Flow	1. Owner presses "Delete a Record" 2. Owner chooses the record to delete 3. Owner presses the "Confirm Delete" button 4. Owner logs out		
Alternative Flows	1. Owner presses "Delete Record" 2. Owner chooses a record to delete 3. Owner presses "Cancel"		
Exceptions:	Global) If database connection has an issue		

	14. Error appears on screen 15. Give instructions on fixing database connectivity
Includes:	N/A
Frequency of Use:	Almost never, a very niche case where a customer cancels after the beginning of a project.
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A

Appendix IV - Class Diagram





Appendix V - Client's Tables

Project Records

Name | Project Details | Profit | Expenses

The image shows a handwritten table with four columns: 'Name', 'Проект' (Project), 'Прибыль' (Profit), and 'Расход' (Expense). The table is filled with handwritten entries, some of which are highlighted in yellow. The handwriting is in Cyrillic script. The table appears to be a record of various projects and their associated costs and profits.

Address Records

Name and Time	Address	Distance(in Km)
---------------	---------	-----------------

Name	Time	Address	Km
1.
2.
3.
4.
5.
6.
7.
8.
9.
10.
11.
12.
13.
14.
15.
16.
17.
18.
19.
20.

Mediography

Stretch Ceilings Market Trend, Size, Share | Report [2028]. (2021, February). Business Research Insights. Retrieved February 23, 2023, from <https://www.businessresearchinsights.com/market-reports/stretch-ceilings-sales-market-100067>

(n.d.). Stretch Ceilings and Wall Decor in Montreal. Favorite Design. Retrieved February 23, 2023, from <https://www.favoritedesign.ca/en/>