**Software Requirements Specification**

for

trendAssist App

Version 1.0 approved

**Prepared by:**

Andre Manz

Cade Wall

Aaron Turner

Aaron Riggs

Xujia Wu

Mayur Bhakta

Flying Mongeese

February 5, 2018

Table of Contents

Revision History..........................................................................................................................3

1. Introduction.................................................................................................................................3

1.1 Purpose..................................................................................................................................3

1.2 Document Conventions.........................................................................................................3

1.3 Product Scope.....................................................................................................................3

1.4 References...........................................................................................................................3

2. Overall Description............................................................................................................................4

2.1 Product Perspective.............................................................................................................4

2.2 Product Functions...............................................................................................................4

2.3 User Classes and Characteristics.......................................................................................4

2.4 Operating Environment.......................................................................................................4

2.5 Design and Implementation Constraints.............................................................................4

2.6 User Documentation...........................................................................................................4

2.7 Assumptions and Dependencies.........................................................................................4

3. External Interface Requirements........................................................................................................5

3.1 User Interfaces....................................................................................................................5

3.2 Hardware Interfaces............................................................................................................6

3.3 Software Interfaces.............................................................................................................6

3.4 Communications Interfaces................................................................................................6

4. System Features.................................................................................................................................6

4.1 System Feature 1.................................................................................................................6

4.2 System Feature 2 (and so on)..............................................................................................7

5. Other Nonfunctional Requirements...................................................................................................7

5.1 Performance Requirements.................................................................................................7

5.2 Safety Requirements...........................................................................................................7

5.3 Security Requirements........................................................................................................7

5.4 Software Quality Attributes................................................................................................7

5.5 Business Rules....................................................................................................................7

6. Use Case Models……........................................................................................................................8

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| SRS FM | 2/13/2018 | Initial Setup | 1.0 |
|  |  |  |  |

# **1. Introduction**

## **1.1 Purpose**

## The purpose of this Software Requirements Specification document is to describe, in detail, the functionalities of trendAssist, a sales trend prediction software. This document will explain the purpose, features, constraints, interfaces, and functionality of trendAssist.

## **1.2 Document Conventions**

This document was written using the IEEE template for System Requirements Specification.

## **1.3 Product Scope**

trendAssist is a business finance and trend prediction program that allows the user to input data over sales made in a day; this data will then be submitted into a database containing financial sales data for every day of the week. It will also allow the user to input data in years past to increase the accuracy of sales predictions; the program will then use this information to generate a new estimation of revenue for that same day of the week in the future and describe requirements needed to hit specific profit margins. This will enable the user (owner or manager) to determine the optimal staffing needed on any given day to support the expected crowd and maximize profits. The most recent year’s data will be weighted higher than the previous year’s data to maintain accuracy and adjust for recent trends.

## **1.4 References**

IEEE 830

# 

# **2. Overall Description**

## **2.1 Product Perspective**

This program is a new, self-contained product that will be convenient for small business owners to make use of after inputting financial data by predicting potential future customer traffic based on this data on a given day in the future. This product will exist as a personal computer application.

## **2.2 Product Functions**

The primary functions of trendAssist are as follows:

- Input/Modify sales data

- Generate prediction of sales based on user inputted date

- Create data visualization

## **2.3 User Classes and Characteristics**

* Admin Account: authorization for total accessibility to view and modify all sales data
* Employee Account: authorization to input new sales data

## **2.4 Operating Environment**

The software will operate on Windows OS and mac OS.

This application will require the host computer to run:

- Java

- MySQL Server Database

- Windows/MacOS.

## **2.5 Design and Implementation Constraints**

The design and implementation of the program will be limited by the memory allocation ability of the programming languages used.

## **2.6 User Documentation**

Generic readme.txt file that includes basic instructions on how to navigate the program, input new data, view past data, generate new prediction graph, etc.

## 

## **2.7 Assumptions and Dependencies**

Constraints to the system:

- The personal computer must have a steady power source (External/Internal Power).

- The personal computer must have Input/Output capability.

- The personal computer must run a minimum of Windows OS 7/Mac OS 10.8.

- The personal computer will have enough storage space to host the mySQL database.

# **3. External Interface Requirements**

## **3.1 User Interfaces**

**SCREEN 1: Login Screen**

The opening screen will be comprised of a login screen that will take the user’s Username and Password.

Text Field 1: Username

Pre-existing account username stored in account mySQL database.

Text Field 2: Password

Password associated with pre-existing account.

Button 1: Sign In

Button to verify login credentials.

Once credentials have been verified, the main screen will have buttons for the main functions of the program.

**SCREEN 2: Main Screen**

Button 1: Generate Sale Prediction

This button will lead to a page that will prompt the user for a specific date. A small description will remind the user to only input a date that does not predate the current date.

Button 2: Modify Sales Data

This button will lead to another screen which will allow the user to select a specific date to modify the sales data. This is the screen that will be used to input new data after closing each night.

Button 3: Account Settings

This button will lead the user to a screen which will allow them to modify their login credentials.

Button 4: Create New Account

This button will lead the user, only if logged in as an admin, to a page where they will be able to enter details for a new account. This will create a restricted employee account where the new user is only authorized to utilize the input sales functionality of the program.

**SCREEN 3: Data Entry Screen**

Field 1: Date of Sales

Field 2: Sales Value

Button 1: Submit Data

**SCREEN 4: Account Creation Screen**

Field 1: Username

Field 2: Password

Field 3: Confirm Password

Button 1: Create My Account

## **3.2 Hardware Interfaces**

This program will be lightweight enough to store on a flash drive to maintain portability and be cross-platform with Windows and macOS devices alike to be as accessible as possible. It will not require the user to install on their machine but will be ran locally off the flash drive.

## **3.3 Software Interfaces**

Splitting the project up into 3 sections, frontend, security/encryption, and backend, the database will be a part of the backend that Python will be able to interface with and pass data to the frontend, written in Java to handle the data and number crunching. The database information will be fully secured and encrypted with HTTPS encryption.

Not fully discussed yet.

Databases: Excel, mySQL

Languages: Python, Java

## **3.4 Communications Interfaces**

TBD.

# **4. System Features**

## **4.1 System Feature 1**

4.1.1 Description and Priority

TBD.

4.1.2 Stimulus/Response Sequences

TBD.

4.1.3 Functional Requirements

TBD.

## **4.2 System Feature 2 (and so on)**

TBD.

# **5. Other Nonfunctional Requirements**

## **5.1 Performance Requirements**

Software will be able to run on devices that feature operating systems no older than Windows XP, the program will be able to run on computers that have 1GB of memory at a minimum. This software will be able to run on a device that has a processor that runs at 1GHz at a minimum.

## **5.2 Safety Requirements**

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

## **5.3 Security Requirements**

All customer data will be encrypted with AES-256. User accounts will be related to AES encryption keys to unlock the customers data when requested by an authorized user. The program will feature a login screen so that an authorized user may log in securely. The user account data will be stored with encrypted keys that will be checked against stored secure hashes, allowing access only from authorized users regardless of access to the physical device.

## **5.4 Software Quality Attributes**

The software is portable as well as able to be handled on multiple operating systems.

## **5.5 Business Rules**

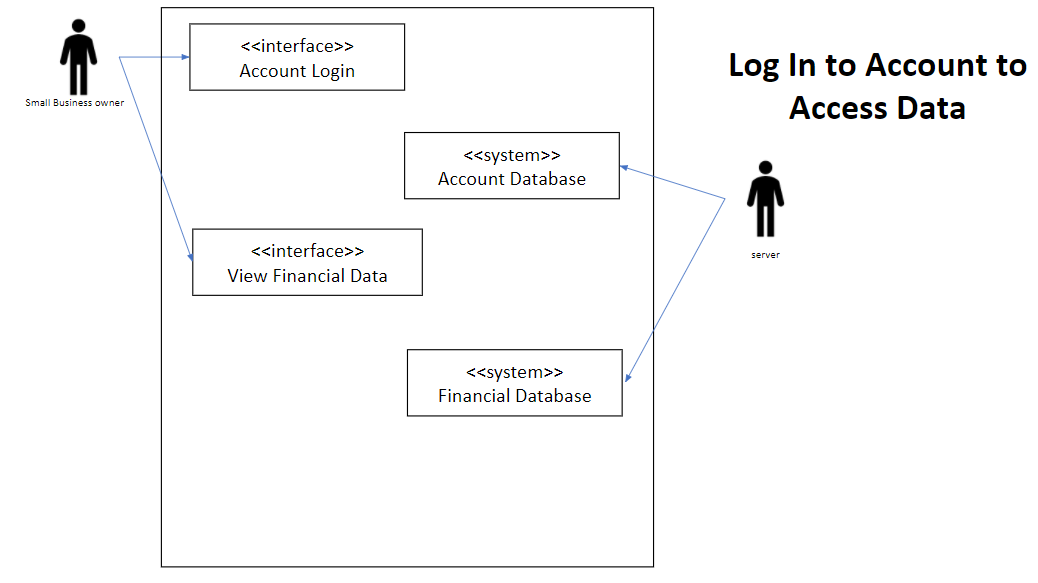
This software will be secure so that only authorised users within the approved organization will be able to use this software.

# **6. Use Case Models**

**Use Case Model**

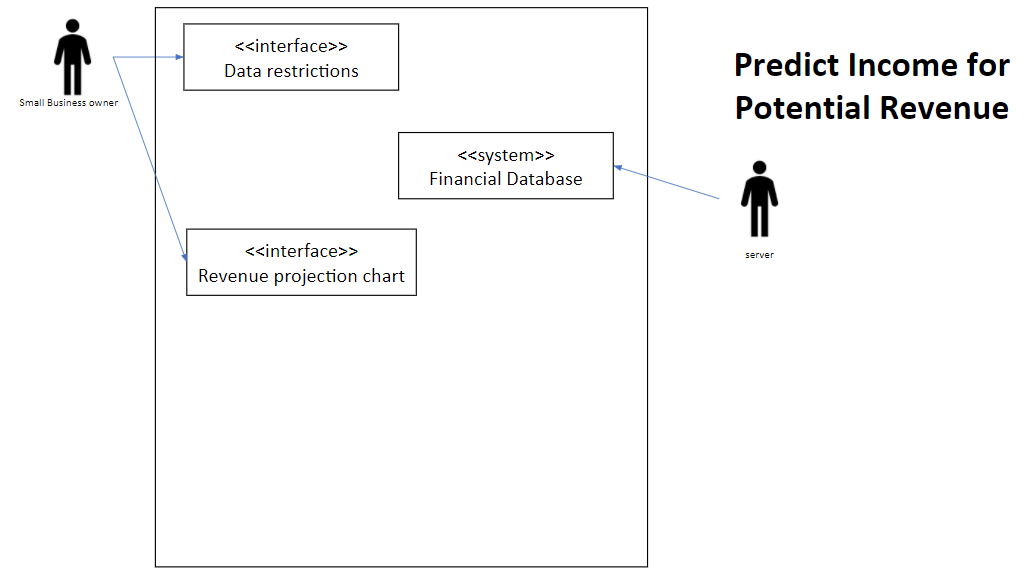
|  |  |
| --- | --- |
| **Title:** | **Log In to Account to Access Data** |
| **Description**: | As a small business owner, I want to be able to log into my account, so that I can easily access my data. |
| **System Under Design:** |  |
| **Primary Actor**: | Small business owner |
| **Participants**: | Small business owner |
| **Goal**: | Log in to account to be able to view all data inputted as well as predictions |
| **Following Use Cases**: |  |
| **Invariant**: |  |
| **Precondition**: | User shall log in successfully, and already have financial data inputted into database |
| **Success Postcondition**: | Ability to enter more data into database |

|  |  |
| --- | --- |
| **STEPS**:   1. User enters account ID and password. 2. User information is checked. 3. System displays all data of the business. 4. User is prompted to enter new data | **ALTERNATIVES**:   * If user account information is invalid, extend to Login Use Case |



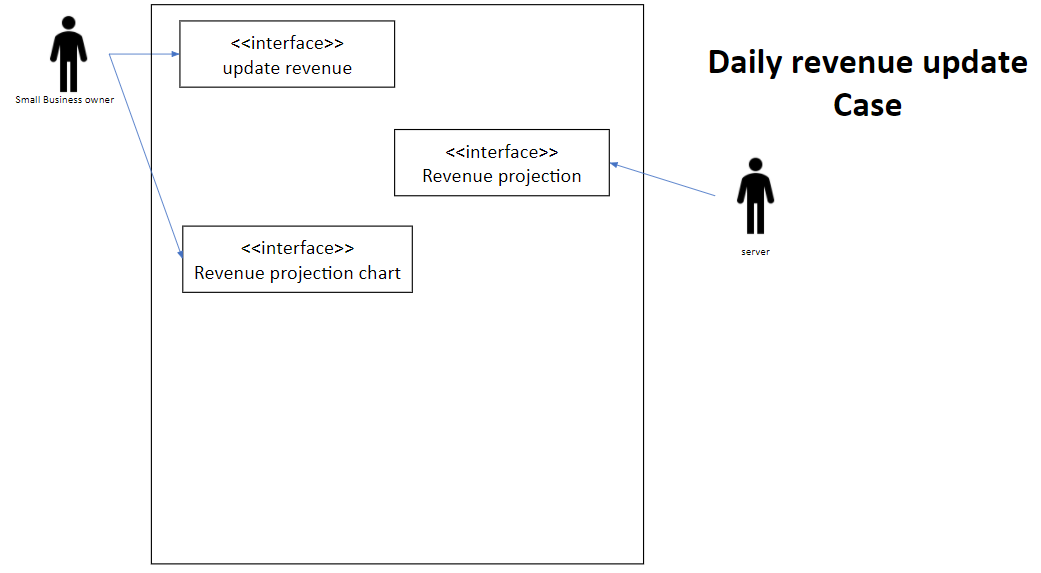
|  |  |
| --- | --- |
| **Title:** | **Predict Income for Potential Revenue** |
| **Description**: | As a small business owner, I want to be able to predict the amount of income on a certain day so that I can account for potential revenue |
| **System Under Design:** |  |
| **Primary Actor**: | Small business owner |
| **Participants**: | Small business owner |
| **Goal**: | Predict the amount of income on a certain day to account for potential revenue |
| **Following Use Cases**: |  |
| **Invariant**: |  |
| **Precondition**: | User shall log in successfully, and already have financial data inputted into database |
| **Success Postcondition**: |  |

|  |  |
| --- | --- |
| **STEPS**:   1. User enters account ID and password. 2. User information is checked. 3. System displays all data of the business. 4. User types a date. 5. System displays a predicted amount of revenue to be earned on that date based on that day in past years. | **ALTERNATIVES**:   * If user account information is invalid, extend to Login Use Case * If financial data for an inputted date isn’t found, prompt for data to be entered |



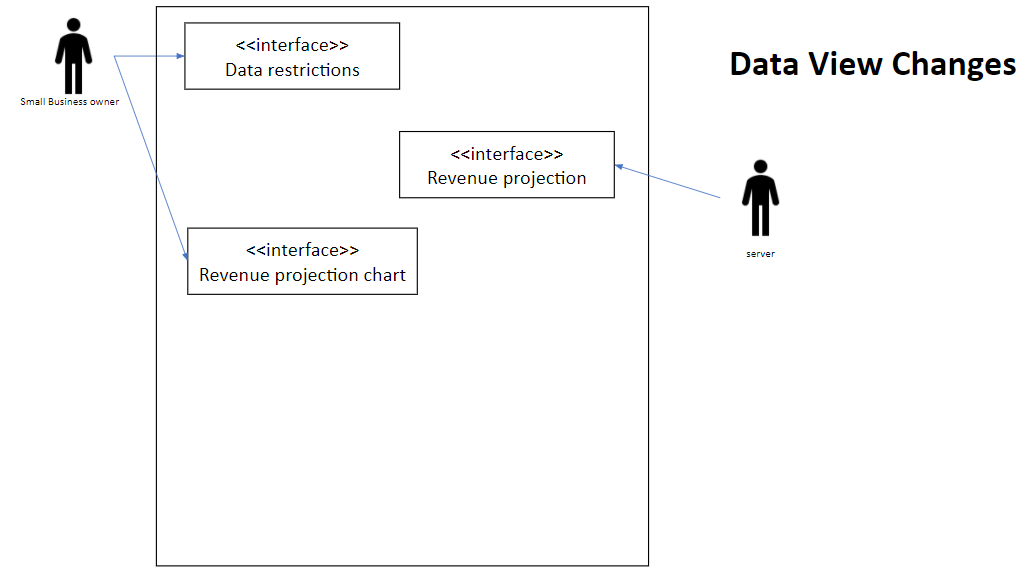
|  |  |
| --- | --- |
| **Title:** | **Revenue updating** |
| **Description**: | As a small business owner, I want to be able to input the total revenue made for the day into the database upon closing so that I can use that data to update the accuracy of the software |
| **System Under Design:** |  |
| **Primary Actor**: | **Business owner** |
| **Participants**: |  |
| **Goal**: | **To update the projections based on daily updates to revenue** |
| **Following Use Cases**: |  |
| **Invariant**: |  |
| **Precondition**: | **User shall log in successfully.** |
| **Success Postcondition**: | **Revenue predictions will be updated according to the data provided.** |

|  |  |
| --- | --- |
| **STEPS**:  1. User enters account ID and password.  2. User information is checked.  3. User request to update with today’s revenue.  4. System sends todays data to server.  5. System updates projections based on the new information.  6. System shows updated projections to user. | **ALTERNATIVES**:   * **If user account information is invalid,**   **extend to Login Use Case.** |



|  |  |
| --- | --- |
| **Title:** | **Data view changes** |
| **Description**: | As a small business owner, I want to be able to set certain parameters so that i can easily view specific data |
| **System Under Design:** |  |
| **Primary Actor**: | **Business owner** |
| **Participants**: |  |
| **Goal**: | **Show different projections based on user input** |
| **Following Use Cases**: |  |
| **Invariant**: |  |
| **Precondition**: | **User shall log in successfully.** |
| **Success Postcondition**: | **User will be shown the selected range of data.** |

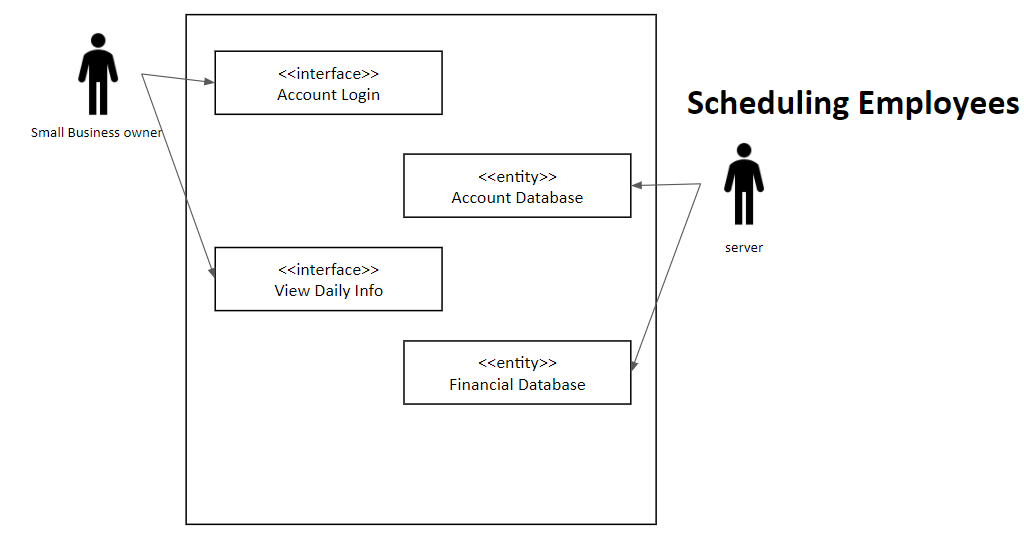
|  |  |
| --- | --- |
| **STEPS**:  1. User enters account ID and password.  2. User information is checked.  3. User selects parameters to view data under.  4. System selects the sections of data these parameters apply to.  5. System shows the user the data in the range. | **ALTERNATIVES**:   * **If user account information is invalid,**   **extend to Login Use Case.** |



|  |  |
| --- | --- |
| **Title:** | **Scheduling Employees** |
| **Description**: | **Business owner shall be able to see the number of customers received on a certain day, so that he/she can schedule enough employees.** |
| **System Under Design:** |  |
| **Primary Actor**: | **Business owner** |
| **Participants**: |  |
| **Goal**: | **Schedule employees accordingly.** |
| **Following Use Cases**: |  |
| **Invariant**: |  |
| **Precondition**: | **User shall log in successfully.** |
| **Success Postcondition**: | **User can see the number of customers received on a certain day.** |

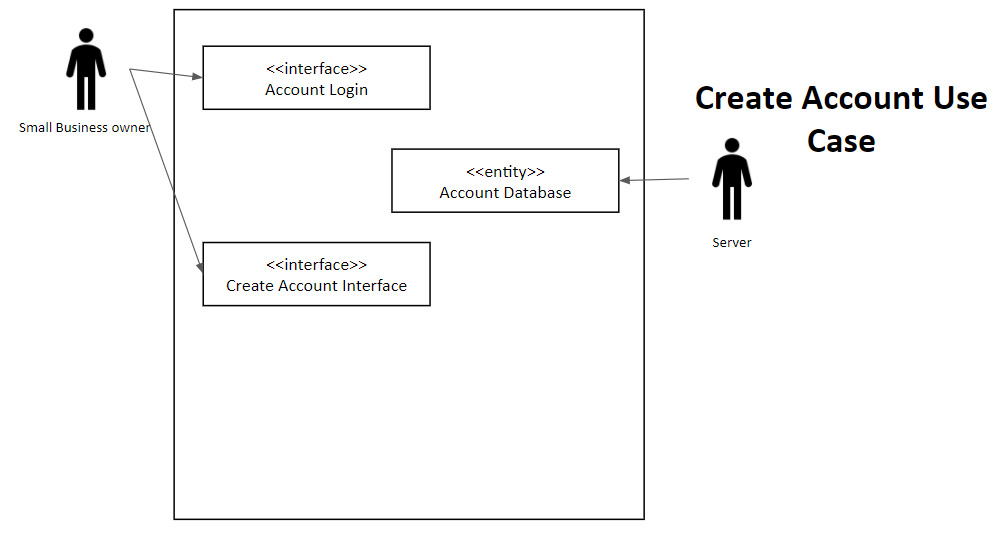
|  |  |
| --- | --- |
| **STEPS**:   1. User enters account ID and password. 2. User information is checked. 3. System display all data of the business. 4. User type a date. 5. Systems display information about date inputted. | **ALTERNATIVES**:   * **If user account information is invalid, extend to Login Use Case.** |

\



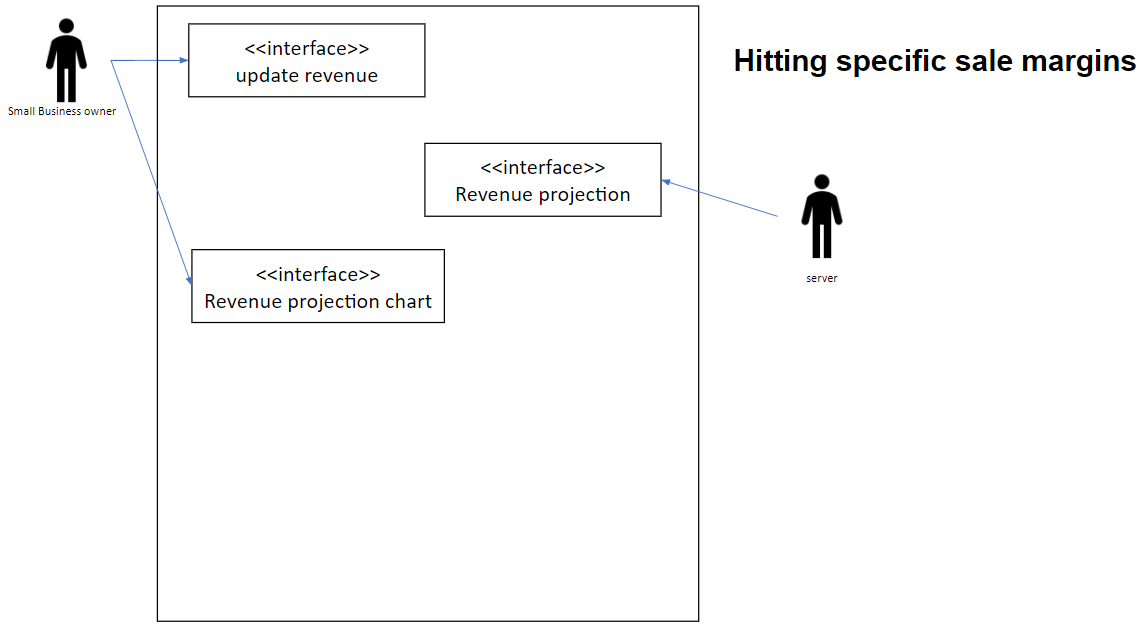
|  |  |
| --- | --- |
| **Title:** | **Account Creation Use Case** |
| **Description**: | **Business owner creates an account.** |
| **System Under Design:** |  |
| **Primary Actor**: | **Business owner.** |
| **Participants**: |  |
| **Goal**: | **User creates an account.** |
| **Following Use Cases**: |  |
| **Invariant**: |  |
| **Precondition**: | **User types a valid username and password.** |
| **Success Postcondition**: | **User successfully creates an account.** |

|  |  |
| --- | --- |
| **STEPS**:   1. System prompt user to create an account. 2. User type a valid username. 3. User types a valid password. 4. User creates an account successfully. | **ALTERNATIVES**:   * **If username already exists or it is invalid, then system prompt the user to enter a new username.** * **If password already exists or it is invalid, then system prompt the user to enter a new password.** |



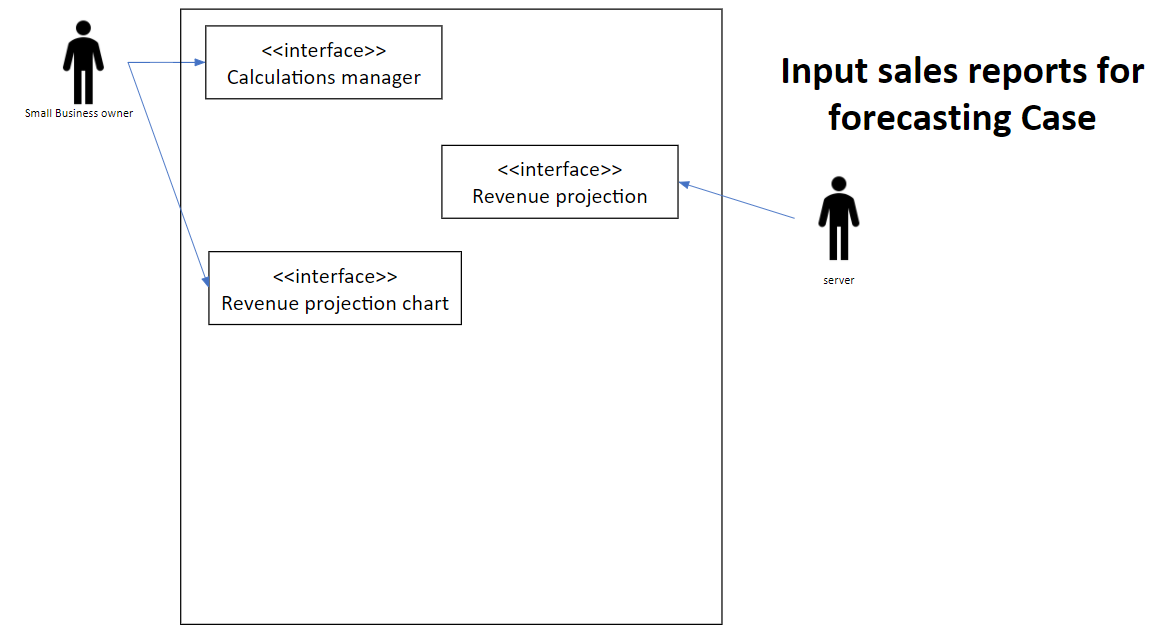
|  |  |
| --- | --- |
| **Title:** | **Hitting specific sale margins** |
| **Description**: | **The business owner wants to input history sales report and then can track the sale margins.** |
| **System Under Design:** |  |
| **Primary Actor**: | **Business owner** |
| **Participants**: |  |
| **Goal**: | **Find the specific sale margins** |
| **Following Use Cases**: |  |
| **Invariant**: |  |
| **Precondition**: | **User shall input valid sales report** |
| **Success Postcondition**: | **user can prepare and know which day to hit specific sale margins** |

|  |  |
| --- | --- |
| **STEPS**:   1. User enter account ID and password. 2. User information is checked. 3. System display all dates of the business. 4. User choose a date that wants to input. 5. User input that date’s sales report. 6. User chose the time to see the most requirement. | **ALTERNATIVES**:  **-If user input invalid data of the sales report, then system prompt the user to check the data they input.** |



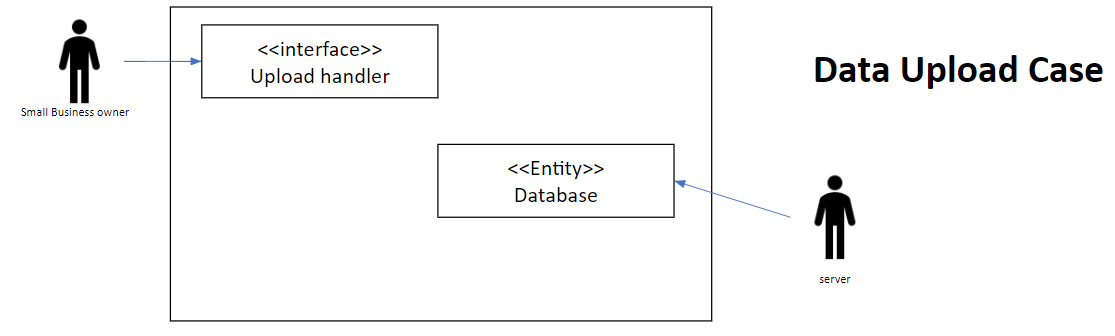
|  |  |
| --- | --- |
| **Title:** | **Input sales reports for forecasting** |
| **Description**: | As a small business owner, I want to use sales report from past years so that I can get an estimate prediction of future sales. |
| **System Under Design:** |  |
| **Primary Actor**: | Business Owner |
| **Participants**: |  |
| **Goal**: | Produce a prediction for future sales based on the data provided |
| **Following Use Cases**: |  |
| **Invariant**: |  |
| **Precondition**: | User has logged in and its sales reports have been uploaded |
| **Success Postcondition**: | The user can see the prediction that has been generated. |

|  |  |
| --- | --- |
| **STEPS**:   1. User logs in 2. After they have uploaded their data they select to do the prediction 3. They select what date(s) they would like to predict for 4. They continue | **ALTERNATIVES**:   * **If the user account information is not valid, extend to login use case** * **If there is no previous data extend to use case ID 21** |



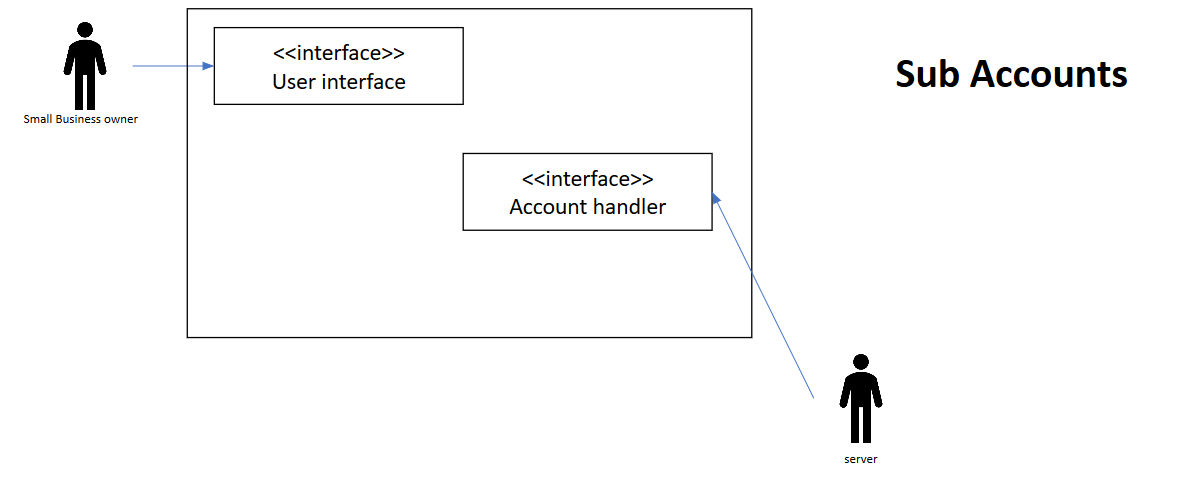
|  |  |
| --- | --- |
| **Title:** | **Data upload** |
| **Description**: | As a small business owner I want to be able to input sales reports from past years so they can be used in the application |
| **System Under Design:** |  |
| **Primary Actor**: | **Business Owner** |
| **Participants**: | **Business owner, server** |
| **Goal**: | **Uploading and saving the users data** |
| **Following Use Cases**: |  |
| **Invariant**: |  |
| **Precondition**: | **The user has an account and is logged in** |
| **Success Postcondition**: | **The information has successfully been saved** |

|  |  |
| --- | --- |
| **STEPS**:   1. The user logs in 2. The user selects to upload data to their account | **ALTERNATIVES**:  · **If the user account information is not valid, extend to login use case** |



|  |  |
| --- | --- |
| **Title:** | **Sub Accounts** |
| **Description**: | As a small business owner I want to be able to create sub accounts so that my employes can have restricted access |
| **System Under Design:** |  |
| **Primary Actor**: | **Business owner** |
| **Participants**: | **Business owner, server** |
| **Goal**: | **To have accessibility to information on multiple accounts provided the business owner allows** |
| **Following Use Cases**: |  |
| **Invariant**: |  |
| **Precondition**: | **The users account has been created** |

|  |  |
| --- | --- |
| **STEPS**:  1.The user logs in  2.The user selects to create a new account  3.User fills out the required information | **ALTERNATIVES**:   * **If the user account cannot login, extend to login use case** * **If the account cannot be created, contact admin for support** |



|  |  |
| --- | --- |
| **Title:** | **Input date into software to review projected sales** |
| **Description**: | As a small business owner, I want to input any date into the software so that I can review data over my company for that specific date. |
| **System Under Design:** |  |
| **Primary Actor**: | Small business owner |
| **Participants**: | Small business owner |
| **Goal**: | Input any date into the software so that I can review data over my company for that specific date. |
| **Following Use Cases**: |  |
| **Invariant**: |  |
| **Precondition**: | Data will have been inputted into the database. |
| **Success Postcondition**: | Predicted value over the specified date will be returned |

|  |  |
| --- | --- |
| **STEPS**:   1. User enters a specified date 2. Corresponding dates in the database will be evaluated. 3. System displays the predicted data for the inputted by the user. 4. User is prompted to enter new data. | **ALTERNATIVES**:   1. If the database has no corresponding data for the inputted date, user will be prompted to add the data to the database. 2. If the user inputs a date that has already passed, the system will display the database value for that given date. |

