**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**

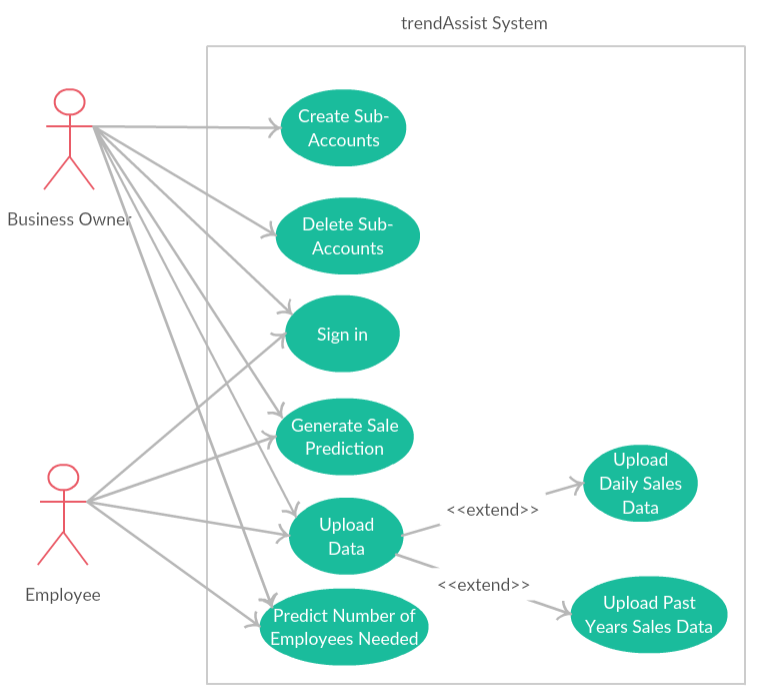
The software is portable as well as able to be handled on multiple operating systems.

## **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**

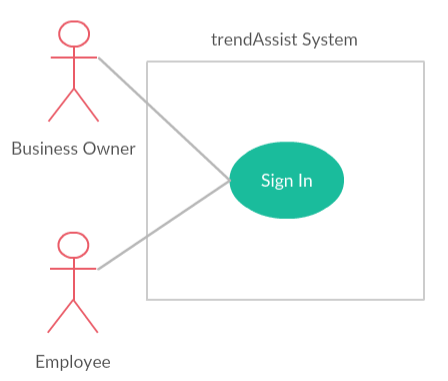
**Overall System Use Case Diagram**



**6.1 Sign in**

|  |  |
| --- | --- |
| **Title:** | **Sign in** |
| **Description**: | Existing user tries to login in the application. |
| **Actors**: | Business owner, employee, application. |
| **Precondition**: | User is not logged in but have an existing account. |
| **STEPS**:   1. User starts application. 2. User enters username and password. 3. User information is checked against the database for validation. 4. If valid, user is able to get in the application. | **ALTERNATIVES**:   * If user account information is invalid a message will be prompted showing the error. |
| **Postcondition:** | User successfully logged in. |

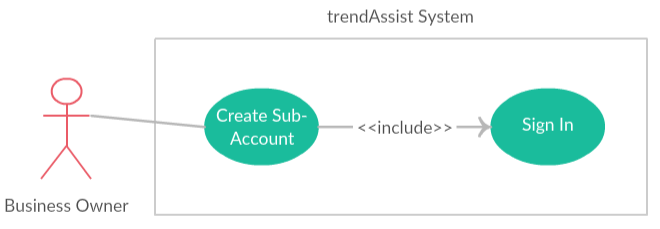
**6.1.1 Sign in Use Case Diagram**



**6.2 Create Sub-Account**

|  |  |
| --- | --- |
| **Title:** | **Sub account creation** |
| **Description**: | Business owner user tries to create account for employee. |
| **Actors**: | Business owner, application. |
| **Precondition**: | User is the business owner and is not logged in but have an existing account. |
| **STEPS**:   1. Business owner starts application. 2. Business owner enters username and password. 3. Business owner information is checked against the database for validation. 4. If valid, business owner is able to get in the application. 5. Business owner selects to create account for employee. 6. Business owner enter employees’ username and password. 7. Employees’ username is checked if it already exists in the database. 8. If not, then employee account is successfully created. | **ALTERNATIVES**:   * If employee username already exist in the database then a message will be prompted to enter a new username. |
| **Postcondition:** | Business owner successfully creates new account for employee. |

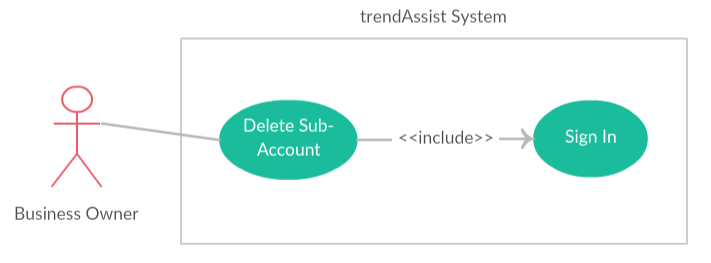
**6.2.1 Create Sub-Account Use Case Diagram**



**6.3 Delete Sub-Account**

|  |  |
| --- | --- |
| **Title:** | **Delete employee account from system.** |
| **Description**: | Business owner wants to delete account information of an employee that no longer works with them |
| **Actors**: | Business owner, application. |
| **Precondition**: | User is the business owner and has an existing account. |
| **STEPS**:   1. User star application. 2. User enter username and password. 3. User information is checked against the database for validation. 4. If valid, user is able to get in the application. 5. User select account settings. 6. System displays all employee’ account. 7. User types account username he wants to delete, and press delete. 8. System deletes selected account from the database and displays a message for successful deletion. | **ALTERNATIVES**:   * If user account information is invalid extend to Sign in use case. |
| **Postcondition:** | User successfully deletes account of an employee that no longer works with them. |

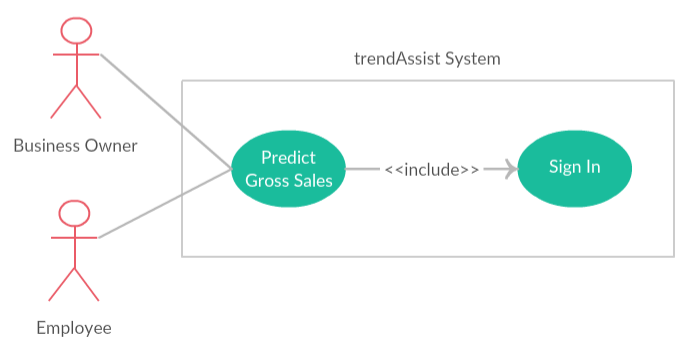
**6.3.1 Delete Sub-Account Use Case Diagram**



**6.4 Predict Gross Sales**

|  |  |
| --- | --- |
| **Title:** | **Predict gross sales** |
| **Description**: | Business owner or employee wants to predict amount of income for a date range to have an idea of potential revenue. |
| **Actors**: | Business owner, employee, application. |
| **Precondition**: | User has an existing account |
| **STEPS**:   1. User starts application. 2. User enters username and password. 3. User information is checked against the database for validation. 4. If valid, user is able to get in the application. 5. User selects to generate sale prediction. 6. User enters a date range. 7. System displays a predicted amount of revenue to be earned for that date range. | **ALTERNATIVES**:   * If user account information is invalid extend to Sign in use case. * If financial data for an inputted date isn’t found, prompt for data to be entered |
| **Postcondition:** | System provides an accurate prediction to user. |

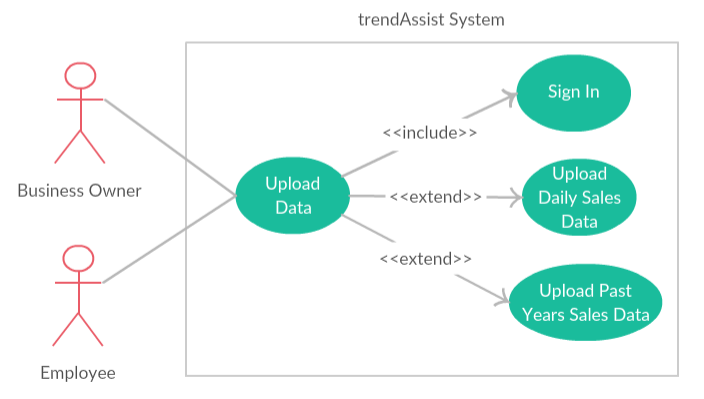
**6.4.1 Predict Gross Sales Use Case Diagram**



**6.5 Upload Data**

|  |  |
| --- | --- |
| **Title:** | **Upload data** |
| **Description**: | Business owner or employee wants to upload sales data in the database to improve the accuracy of the software. |
| **Actors**: | Business owner, employee, application. |
| **Precondition**: | User has an existing account |
| **STEPS**:   1. User starts application. 2. User enters username and password. 3. User information is checked against the database for validation. 4. If valid, user is able to get in the application. 5. User selects to modify sales data. 6. User then either select to input sales data for the day or input sales data for past years. 7. User input sales data. 8. System stores data in the database. | **ALTERNATIVES**:   * If user account information is invalid extend to Sign in use case. * If user selects to input sales data for the day extend to Upload daily sales use case. * If user selects to input sales data for past years extend to Upload past year sales use case. |
| **Postcondition:** | User successfully store/update sales data in the database. |

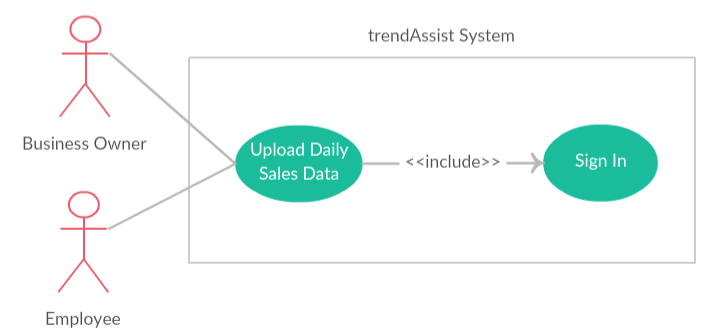
**6.5.1 Upload Data Use Case Diagram**



**6.6 Upload Daily Sales Data**

|  |  |
| --- | --- |
| **Title:** | **Upload daily sales data** |
| **Description**: | Business owner or employee wants to upload daily sales data in the database to improve the accuracy of the software. |
| **Actors**: | Business owner, employee, application. |
| **Precondition**: | User has an existing account |
| **STEPS**:   1. User starts application. 2. User enters username and password. 3. User information is checked against the database for validation. 4. If valid, user is able to get in the application. 5. User selects to modify sales data. 6. User then select to input sales data for the day. 7. User input data manually by typing the date and gross sales for the day. 8. System stores data in the database. | **ALTERNATIVES**:   * If user account information is invalid extend to Sign in use case. |
| **Postcondition:** | User successfully store/update daily sales data in the database. |

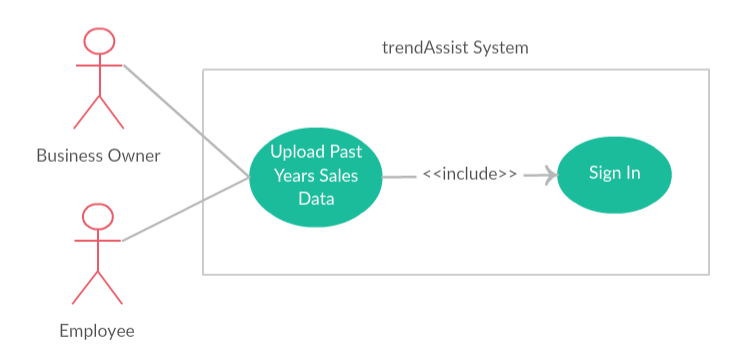
**6.6.1 Upload Daily Sales Data Use Case Diagram**



**6.7 Upload Past Years Sales Data**

|  |  |
| --- | --- |
| **Title:** | **Upload past years sales data** |
| **Description**: | Business owner or employee wants to upload past years sales data in the database to improve the accuracy of the software. |
| **Actors**: | Business owner, employee, application. |
| **Precondition**: | User has an existing account |
| **STEPS**:   1. User starts application. 2. User enters username and password. 3. User information is checked against the database for validation. 4. If valid, user is able to get in the application. 5. User selects to modify sales data. 6. User then select to input sales data for the day. 7. User input data manually by typing the date and gross sales for the day. 8. System stores data in the database. | **ALTERNATIVES**:   * If user account information is invalid extend to Sign in use case. |
| **Postcondition:** | User successfully store/update past years sales data in the database. |

**6.7.1 Upload Past Years Sales Data Use Case Diagram**



**6.8 Predict Number of Employees Needed**

|  |  |
| --- | --- |
| **Title:** | **Predict number of employees needed** |
| **Description**: | Business owner or employee wants to get an estimated number of employees needed based on sales made on previous year for any given date. |
| **Actors**: | Business owner, employee, application. |
| **Precondition**: | User has an existing account |
| **STEPS**:   1. User starts application. 2. User enters username and password. 3. User information is checked against the database for validation. 4. If valid, user is able to get in the application. 5. User selects to generate sale prediction. 6. User enters a date range. 7. System displays a predicted amount of revenue to be earned for that date range. 8. User select generate number of employees required. 9. System calculates number of employees required using an algorithm. 10. System displays number of employees needed for the given date. | **ALTERNATIVES**:   * If user account information is invalid extend to Sign in use case. |
| **Postcondition:** | User successfully store/update daily sales data in the database. |

**6.8.1 Predict Number of Employees Needed Use Case Diagram**

