Software Requirements Specification

for

Sales Forecaster

Version 0.1 approved

Prepared by Hassan Mebed

University of Surrey

March 1, 2019

Table of Contents

Table of Contents ii
Revision History ii
1. Introduction 1
1.1 Purpose 1
1.2 Product Scope 1
1.3 Definitions, Acronyms and Abbreviations 1
1.4 References 2
1.5 Overview 2
2. Overall Description 3
2.1 Product Perspective 3
2.2 Product Functions 3
2.3 User Characteristics 3
2.4 Constraints 3
2.5 Assumptions and Dependencies 4
3. Specific Requirements 4
3.1 External Interface Requirements 4
3.1.1 User Interfaces 4
3.1.2 Hardware Interfaces 4
3.1.3 Software Interfaces 4
3.1.4 Communications Interfaces 5
3.2 Functional Requirements 5
3.2.1 System Feature 1: Security for its users 5
3.2.2 System Feature 2: Management of product sales data 5
3.2.3 System Feature 3: Generation and management of quarterly sales forecasts 6
3.2.4 System Feature 4: Comparison of actual product’s sales vs. its forecasted sales 7
3.3 Non-Functional Requirements 8
3.4 Design Constraints 8

Revision History

Name	Date	Reason(s) for Changes	Version

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is to describe a sales forecasting system utilized for predicting quarterly sales for any product/service based on its previous sales. Additionally, it will explain key factors that complement and define the available features of this system. This document is intended to be presented to academics inspecting the development process of the first version of this sales forecasting system. It was written in accordance with the IEEE Recommended Practice for Software Requirements Specifications 830-1998 [1].

1.2 Product Scope

The "Sales Forecaster" helps sales people, business owners, etc. to manage each of their products'/services' sales data and generate quarterly sales forecasts for them.
It can:

- ✓ Take in a product's/service's sales data stored on a local database, and use it to generate a basic quarterly sales forecast.
- ✓ Allow the user to add and remove his/her products/services to/from the database as well as update the sales data of the current stored ones.
- ✓ Provide the user the options of storing any generated sales forecast to the database and manually editing these forecasts.
- ✓ Compare the actual sales data of a product/service with its previously generated forecasts.

1.3 Definitions, Acronyms and Abbreviations

- **Time series**:
  "A sequence of numerical data points in successive order." [2]

- **Mean Absolute Percentage Error (MAPE)**:
  "A statistical measure of how accurate a forecast system is. It measures this accuracy as a percentage, and can be calculated as the average absolute percent error for each time period minus actual values divided by actual values." [3]

- **ARIMA**: Autoregressive Integrated Moving Average.
  It is "A statistical analysis model that uses time series data to either better understand the data set or to predict future trends." [4]

1.4 References

[1] K. Wiegers, "srs_template-ieee.doc", *Web.cs.dal.ca*, 2011. [Online]. Available: https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc. [Accessed: 01- Mar- 2019].

[2] W. Kenton, "Time Series", *Investopedia*, 2018. [Online]. Available: https://www.investopedia.com/terms/t/timeseries.asp. [Accessed: 03- Mar- 2019].

[3] "Mean absolute percentage error (MAPE)", *Statistics How To*, 2017. [Online]. Available: https://www.statisticshowto.datasciencecentral.com/mean-absolute-percentage-error-mape/. [Accessed: 03- Mar- 2019].

[4] J. Chen, "Autoregressive Integrated Moving Average - ARIMA", *Investopedia*, 2018. [Online]. Available: https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp. [Accessed: 03- Mar- 2019].

[5] Simplilearn, *Time Series Analysis - 2* | *Time Series in R* | *ARIMA Model Forecasting* | *Data Science* | *Simplilearn*. 2018. [Streaming video]. Available: https://www.youtube.com/watch?v=Y5T3ZEMZZKs. [Accessed: 04- Mar- 2019].

[6] T. Ylinen, J. Nissi, T. Ala-Risku and J. Småros, "Measuring Forecast Accuracy: The Complete Guide", *RELEX Solutions*. [Online]. Available: https://www.relexsolutions.com/measuring-forecast-accuracy/. [Accessed: 06- Mar- 2019].

1.5 Overview

The rest of this document contains two more sections:
Section two dives deeper into the system's functionality and its use of a database. Furthermore, it describes the different users that might interact with the system and lays out the system's implementation constraints. Lastly, it mentions the assumptions made about the system's development.

Section three details the different requirements that the software must accomplish and the different system interfaces. Finally, it discusses design constraints caused by external factors that will affect the system.

2. Overall Description

2.1 Product Perspective

The system will involve a java-based application that is connected to a local SQL database server on the same computer (local database for now); the java application will interface all of the system's functions to the users, which involves accessing and manipulating data on the database.

2.2 Product Functions

The system's core features for its users are:

1. Security for its users and their respective product/service sales data.
2. Management of product/service sales data.
3. Generation and management of quarterly sales forecasts.
4. Comparison of actual product/service sales data against its previously forecasted data.

2.3 User Characteristics

There is only one type of user class anticipated to use this system. This user class entails sales people working for a company, a business owner keeping track of his/her product's/service's sales, marketing individuals, and other similar individuals concerned with sales data.

2.4 Constraints

1. Ability of the operating system to connect the java application to the local SQL database.
2. Java application will be constrained by the size of the database.
3. Accuracy of sales forecast results will depend on the time series analysis model used in the java application and the integrity of the sales data inputted by the user. The time series model adopted for this application is an ARIMA model [5].

4. The overall design of the system must take into consideration security of product/service data i.e. implement privileged access to each product/service stored on database as well as take measures to encrypt data being sent to and read from the database.

2.5 Assumptions and Dependencies

- The java application will be developed using Java SE 8.
- The UI of the java application will be developed using JavaFX.
- The SQL database will be developed using MariaDB 10.1.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

U1 On system start-up, system will attempt to connect to the local database. If it is unable to, then an error message will be sent to the users to notify them, and when users confirm that they have seen the error message, system will exit.

U2 When any new window, such as a window for editing sales or forecasts data, is opened by the user at any time from the main GUI window, then system should deny the user access to the main GUI window until the new window is closed.

U3 There shall be a top menu bar in the main window of the java application where user can find options to exit, logout, and access help for the current window as well as view the "About" info for this java application.

3.1.2 Hardware Interfaces

The java application and SQL database do not have specific hardware interfaces.

3.1.3 Software Interfaces

S1 The java application will communicate with the local SQL database to send and retrieve user, product/service sales and forecasts data.

3.1.4 Communications Interfaces

Communication between the java application and SQL database is handled by the MySQL Java Connector and the underlying operating sytem.

3.2 Functional Requirements

3.2.1 System Feature 1: Security for its users

F1.1 There shall exist a login system that will verify the current user through requiring user id and password on start-up before continuing to access the rest of the system's features.

F1.2 If user is not registered on the system's database, then he/she will be able to register through the system by inputting a first and last name, user id and password.

F1.3 Each individual user should have access to his/her own stored products/services data only.

F1.4 Logged-in user should be able to logout of the system at any point.

3.2.2 System Feature 2: Management of product sales data

F2.1 User should be able to add a new product/service to the database and continually add sales data to it.

F2.1.1 User cannot add two or more products/services with the same exact name.

F2.1.2 User cannot enter multiple sales records with the same year.

F2.2 On successful system start-up, after user is logged in, he/she should be able to select which of their stored products/services they would like to access from database for system to work with.

F2.3 Once product/service is selected and accessed, user should be able to view its sales data in a table and graph format.

F2.4 User should be able to edit/delete a stored product's/ service's sales data.

F2.5 User should be able to remove any of his/her products/services from the database.

F2.5.1 If a product/service is removed from the database, then all of its saved sales and generated sales forecasts should be removed as well.

3.2.3 System Feature 3: Generation and management of quarterly sales forecasts

F3.1 Given that the selected product/service has at least three years of complete sales history and no gaps between all recorded years of sales, the system should be able to generate an accurate sales forecast for its current year's unknown quarter sales or for its next year's quarterly sales, depending on what the user chooses (user can choose both options). See figure 1 for an example.

Figure 1:

Year	Q1	Q2	Q3	Q4
…	…	…	…	…
2002	344,879	347,304	340,556	298,893
2003	343,942	346,115	? (To be predicted)	? (To be predicted)

Or:

Year	Q1	Q2	Q3	Q4
…	…	…	…	…
2002	344,879	347,304	340,556	298,893
2003	343,942	346,115	341,368 (can still be unknown)	312,432 (can still be unknown)
2004	? (To be predicted)	? (To be predicted)	? (To be predicted)	? (To be predicted)

F3.1.2 There is really no such thing as an acceptable accuracy level for time series forecasting in this case, as it is impossible to know for sure whether a product's/service's trend of demand will remain constant throughout the years, however, one can observe a pattern of forecasting accuracy for an individual product's/service's sales over the years and try to identify what inputs and changes can improve the accuracy [6].

F3.2 User should be able to save a generated sales forecast to the database.

F3.3 Upon successful login and product/service selection, user should be able to view all saved generated sales forecast for the selected product/service in table format.

F3.4 User should be able to manually edit a saved generated sales forecast.

F3.5 User should be able to delete a saved generated sales forecast from the database.

3.2.4 System Feature 4: Comparison of actual product's sales vs. its forecasted sales

F4.1 Upon successful login and product/service selection, user should be able to compare the product's/service's current sales data with its saved generated sales forecasts through viewing a generated graph plotting the two sets of data against each other, and system shall additionally return the mean absolute percentage error (MAPE) of the forecasted data in the graph.

F4.1.1 If there exist multiple saved forecasts for the same year of product sales, then the system shall ask the user which one he/she wants to select for the comparison process.

F4.1.2 User should be able to override the generated comparison graph with a new one at any time, in case he/she makes changes to the sales and/or forecasted data afterwards.

3.3 Non-Functional Requirements

NF1 Database should be able to store at least 50 years of product sales and forecasting data for each product.

NF2 User should receive error message if java application is unable to send or retrieve data from database.

NF3 System should validate every input taken in from the user accordingly before attempting any operation with it, and send an error message if an input cannot be taken in and why.

3.4 Design Constraints

D1 Size of java application should not exceed 10 MB