Software Requirements Specification

(SRS)

for

Paycheck Checker

2.0

Prepared By: Rafael Deluna

Company Name: Meme Matrix

Date Created: 10/16/2020

**Table of Contents**

1. Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions, acronyms, and abbreviations

1.4 References

1.5 Overview

2. Overall Description

2.1 Product perspective

2.2 Product functions

2.3 User characteristics

2.4 Constraints

2.5 Assumptions and dependencies

3. Specific Requirements

3.1 External Interface Requirements

3.2 Functional Requirements

3.3 Performance Requirements

3.4 Logical Database Requirements

3.5 Design Constraints

3.6 Software System Attributes

3.7 Organizing the specific requirements

Appendixes

Index

1 Introduction (Section 1 of the SRS)

This is an introduction of the SRS and it will provide an overview of the entire SRS. It will include the following:

1. Purpose;
2. Scope;
3. Definitions, acronyms, and abbreviations;
4. References;
5. Overview;

1.1 Purpose (Section 1.1 of the SRS)

1. The purpose of this SRS document is that it will serve as a basis for the entire project. The project will be a payroll system that will be used by a specific client. The SRS will lay down the framework of the development process. The SRS describes the software functional and nonfunctional requirements of the Paycheck Checker. The SRS also provides critical information to a team that is working on the project such as development, quality assurance, operations, and maintenance.

b) The intended audience for this SRS will be the members of the team. They will make sure that the payroll system will be correctly functioning through implementation and verification. It is also designed for a client/customer who is looking for a payroll system to be added to their company.

1.2 Scope (Section 1.2 of the SRS)

1. The software product that will be produced by name is JFoenix.

b) JFoenix serves as a JavaFX tutorial as well as a JavaFXML tutorial. It will allow you to perform any task that is within the JavaFX and the JavaFXML libraries. It does not cater to any other computer programming language.

c) The benefits, objectives, and goals of JFoenix is that this particular software will allow you to create buttons and tree tables for your projects. You can also create input fields and boxes as well as popups. However, the main draw of JFoenix is that it can be used for animation and material design.

1.3 Definitions, acronyms, and abbreviations (Section 1.3 of the SRS)

customer: The person, or persons, who pay for the product and usually (not necessarily) decide the requirements.

user: The person, or persons, who operate or interact directly with the product.

1.4 References (Section 1.4 of the SRS)

1. Deluna, Rafael. Paycheck Checker Use Case Document

2. Deluna, Rafael. Paycheck Checker Vision and Scope Document

1.5 Overview (Section 1.5 of the SRS)

1. The rest of the SRS will contain more information about the product that is going to be sold. It will include product perspective and product functions. It will explain how the user will interact with the product. The SRS will tell you about the different interfaces that will be used as well as the design constraints. Finally, the SRS includes information about specific requirements that are involved with the software that is going to be used to create the product.

b) The SRS is broken up into three parts. The first part introduces the SRS. It also states its purpose and scope. The second part describes the overall product. It describes its function, what the user is allowed to do, and the design constraints behind it. The third part details the specific requirements that the developer must follow to make sure that the customer is satisfied with how the product operates.

2 Overall Description (Section 2 of the SRS)

This section of the SRS will define the general factors that affect the product and its requirements. This section also provides a background for the requirements. It will include the following:

1. Product perspective;
2. Product functions;
3. User characteristics;
4. Constraints;
5. Assumptions and dependencies;
6. Apportioning of requirements;

2.1 Product Perspective (Section 2.1 of the SRS)

The Paycheck Checker is a payroll system that can help small businesses and big businesses alike. It allows a manger or the person in charge of the payroll system to add, modify, and delete an employee’s name, their personal information, their benefits, their taxes, their gross income, and of course their net pay. The Paycheck Checker also contains a database so it would be easier to use. This section will also include the following:

a) System interfaces;

b) User interfaces;

c) Hardware interfaces;

d) Software interfaces;

e) Communications interfaces;

f) Memory;

g) Operations;

h) Site adaptation requirements.

2.1.1 System Interfaces

The software that we are using is called JFoenix. JFoenix was used to create login page. It was also used to create multiple tree tables for several components like benefits, employees, gross income, taxes, and net pay.

2.1.2 User Interfaces

1. login menu - Allows the user to log in and access their information.

graphical user interface (GUI) - A GUI is a system of interactive visual components for computer software.

natural language interface (nli) - A type of human computer interface where linguistic phenomena such as verbs, phrases, and clauses act as UI controls selecting, adding, deleting, and modifying different types of date in software applications.

form based interface (fbi) - Allows the user to interact with the application or product by selecting a value out of all the possible values, and by entering text into the fields that accept it.

b) A user will see a login menu and they will have to enter their id/username + their password to gain access to all of the information that they are allowed to see.

The use of a GUI will help perform the add, modify, and delete functions in the JFXDrawer.

The nli includes adding, deleting, and modifying.

adding: entering a new value from the database.

deleting: removing a value from the database.

modifying: making a slight change to one of the values from the database.

The fbi will allow the user to enter in a value or text into any field that they choose but that specific field has to accept that value or text. For example, gross income, taxes, and net pay will accept doubles or ints. On the other hand, benefits and personal information will only accept strings.

2.1.3 Hardware Interfaces

There are no hardware interfaces that have been identified so therefore, it is non-applicable.

2.1.4 Software Interfaces

SI-1: Database- The payroll system heavily relies on the database as well as the source code in order for it to be fully functional. The database contains plenty of information including employee names, their personal information, their benefits, their taxes, their gross income, and finally their net pay.

SI-1.1: Allows the manager or whoever is in charge of payroll to add, modify, and delete all the data that is contained within the database.

SI-1.2: Allows the employee to view only the information that pertains to them.

2.1.5 Communication Interfaces

CI-1: The system will let the user know if a id/username or a password is entered incorrectly.

CI-2: The system will let the user know if they have successfully logged in.

CI-3: The system will let the user know if they have successfully logged out.

CI-4: The system will let the user know if an action/ event is performed successfully.

CI-5: The system will let the user know if an action/event is performed unsuccessfully.

CI-6: The system will let the user know if the user entered an incorrect value into a certain field.

2.1.6 Memory Constraints

I don’t necessarily think that we ran into any memory constraints.

2.1.7 Operations

1. The user initiated operations are secure logins, searching for an employee, print out a check, creating an account by creating an id/username along with a password, recovering that same password if it’s lost, opening up an employee window, confirming if an employee has been added, adding data, deleting data, and modifying data.

b) All operations will be attended.

c) Sorting was used because there was different types of information that was sorted into their own category. Aggregation was used because the database combines multiple pieces of data. Analysis was used because the use of tree tables collects the data, therefore, the data is presented in an organized way.

d) Recovering a loss password is a backup/recovery operation that we used.

2.2 Product Functions (Section 2.2 of the SRS)

PF-1: Printing out a check

PF-2: Searching for employees

PF-3: Opening up an employee window

PF-4: Adding, deleting and modifying employee information

PF-5: Adding, deleting, and modifying an employee’s benefits

PF-6: Adding, deleting, and modifying an employee’s taxes

PF-7: Adding, deleting, and modifying an employee’s gross income

PF-8: Adding, deleting, and modifying an employee’s net pay.

PF-9: Determining whether a part time employee is qualified to purchase health care, dental benefits, optical benefits, and vision benefits.

2.3 User Characteristics (Section 2.3 of the SRS)

The educational level of the user should be that the user should at least have graduated college already. They must also have a degree in banking. The user must be an expert in the field of banking.

2.4 Constraints (Section 2.4 of the SRS)

There are not really a lot of constraints when it comes to the payroll system. One constraint is to make sure the payroll system functions properly so that it can fulfill the client’s requirements and needs. Another constraint is security related. The client have to let the developers know who is in charge of the payroll system, whether it’s a manager or a specific individual. The client must also let them know which person is allowed to see or not allowed to see certain types of information.

2.5 Assumptions and Dependencies (Section 2.5 of the SRS)

There will not be any changes to the design constraints so the SRS will have very little modification.

2.6 Apportioning of Requirements (Section 2.6 of the SRS)

There are no requirements that will be delayed because a future version of the system has not even been thought up yet.

3 Specific Requirements (Section 3 of the SRS)

This section of the SRS will contain all of the software requirements to a sufficient level of detail to enable designers to design a system to meet those requirements and testers should also test the system to make sure that all requirements are met. The requirements should a include a description of the stimulus(input) and the response(output). It should also include all functions performed by the system in response to an input or output.

3.1 External Interfaces (Section 3.1 of the SRS)

1. JFoenix acts as a tutorial for people who want to learn the language of Java. It aided in the creation of the product because it made the GUI meaning that it created the login menu. JFoenix also contained tree tables so that it could store the information that was within the database in an organized way.

b) The login menu acts as an entrance to certain types of information when the user logs in successfully by entering in their correct username/id and password.

c) The GUI creates the login menu.

d) The nli is a human computer interface where the UI will control the different changes made to the information in the database.

e) The fbi allows you to enter in new information to the database.

3.2 Functional Requirements (Section 3.2 of the SRS)

1. The system shall let the user know if they have logged in successfully.

b) The system shall let the user know if they have entered a incorrect username/id and password. An error message will pop up. This is how we dealt with error handling.

c) The system shall let the user know if they entered in incorrect information into the database. For example, if they entered in a string into the gross income field of the database.

d) The system shall let the user know if an action is successfully performed on the database meaning a piece of information has been added, deleted, or modified.

e) The system shall let the user see the information that they are allowed to see.

f) The system shall let the user find any employee by just typing in their id/username.

3.3 Performance Requirements (Section 3.3 of the SRS)

PE-1: The source code for the payroll system runs very fast so it is pretty efficient. The same thing could be said about the responses to the database queries.

The user will not have to wait a long time for any of the transactions to be completed.

3.4 Logical Database Requirements (Section 3.4 of the SRS)

1. The different types of information that will be added to the database are an employee’s name, their personal information, their benefits, their taxes, their gross income, and their net pay.

b) The database will be used when there is going to be a change that is made to the database.

c) The manager and their employees will have access to the database but they will only see certain information that they should see.

d) The data entities that I listed in part a all relate to one specific employee.

e) One integrity constraint is that a primary key, in this case being the social security number, cannot have a null value, otherwise there will be an error in the system. Another integrity constraint is that if a foreign key in Table A refers to a primary key of Table B, then every value of the foreign key in Table A must be null or available in Table B.

f) The data retention requirements are that the data will be kept in the company’s database forever.The data of any employee will not be deleted. It does not matter what an employee’s status is. An employee’s status will also be included. The different types of statuses include: active, terminated, quit, deceased, and disabled.

Section 3.5 Design Constraints (Section 3.5 of the SRS)

Our group has not run into any design constraints.

Section 3.6 Software System Attributes (Section 3.6 of the SRS)

Here are a list of software system attributes:

a) reliability

b) availability

c) security

d) maintainability

e) portability

3.6.1 Reliability

The reliability of the software system can be tested by acceptance testing, verification, and validation.

3.6.2 Availability

Our payroll system should be fully operational at all times.

3.6.3 Security

For security, we are going to use a password encryption. We are using mysql to store all of our information as well as changing information. A part-time or a full time option is also included to determine the current status of an employee.

There will also be 3 main levels of security:

Level 1: Employee based. An employee would be able to look up their own information and they would be able to print out their own information regarding their deductions, taxes, and/or their salary.

Level 2: This will also be employee based. This will enable company employees to look up information about another employee if they choose to do so. They will gain access to an employee’s title, their specific room number, their phone as well as their email contact.

Level 3: HR based. This will allow Human Resources (HR) to make any modifications and/or edits to all types of information that an employee has.

3.6.4 Maintainability

The interfaces that have been used during this project are system interfaces, software interfaces, and user interfaces. Our group has also used software modularity because we made a class for every type of information that will go into the database. Our group used a pretty efficient algorithm because the time it takes for the source code to run is very fast.

3.6.5 Portability

The proven portable language that our group used is Java. The compiler that we used is Eclipse and the operating systems that we used are most likely a Mac OS X or Windows.

3.7 Organizing the specific requirements (Section 3.7 of the SRS)

3.7.1 User Class

A general user would be an employee meaning that they would only have access to the information that pertains to them. A power user would be the manager because they would have access to all of the information of his/her employees.

3.7.2 Features

The features of a payroll system are accuracy, deductions, record-keeping and streamlining. A payroll system needs to be accurate so that you can have the correct number of working hours for all your employees. Most payroll systems need complex operations in order to accommodate an employee’s tax and benefits needs. A payroll system is used to keep detailed and accurate information where they are stored in a main database. Streamlining can help a payroll system and eliminate busy work. Employees can access their information, print their own checks, and also request their own time off. There is a new feature that will be added for the modified version: not only can an employee print their own check, they can also view their own check. Finally, employees can also view an itemized deductions list of all the taxes and benefits that they are paying.