
Software Requirements Specification

for

APC Payroll

Version 1.0

Prepared by

Group Name: SemiByte

James Alfafara
Jethro Rae Garcia
Alexandra Noynay
Aliyah Kristie Lopez
Chase Villarroel

2021-140144
2021-140428
2021-140392
2021-140947
2021-140032

jsalfafara@student.apc.edu.ph
jbgarcia@student.apc.edu.ph
asnoynay@student.apc.edu.ph
aclopez@student.apc.edu.ph
cgvillarroel@student.apc.edu.ph

Instructor: Mr. Manuel Sebastian Sanchez

Course: SOFTDEV

Date: October 2024

Contents

CONTENTS.....	I
---------------	---

REVISIONS	II
1 INTRODUCTION	1
1.1 DOCUMENT PURPOSE	1
1.2 PRODUCT SCOPE.....	1
1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW.....	1
1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS	2
1.5 DOCUMENT CONVENTIONS.....	2
1.6 REFERENCES AND ACKNOWLEDGMENTS.....	2
2 OVERALL DESCRIPTION.....	3
2.1 PRODUCT OVERVIEW	3
2.2 PRODUCT FUNCTIONALITY	3
2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS	3
2.4 ASSUMPTIONS AND DEPENDENCIES	4
3 SPECIFIC REQUIREMENTS	4
3.1 EXTERNAL INTERFACE REQUIREMENTS.....	4
3.2 FUNCTIONAL REQUIREMENTS	5
3.3 USE CASE MODEL	5
4 OTHER NON-FUNCTIONAL REQUIREMENTS	6
4.1 PERFORMANCE REQUIREMENTS	6
4.2 SAFETY AND SECURITY REQUIREMENTS	7
4.3 SOFTWARE QUALITY ATTRIBUTES	7
5 OTHER REQUIREMENTS.....	7
APPENDIX A – DATA DICTIONARY.....	8
APPENDIX B - GROUP LOG	9

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft v.01	SemiByte	Draft fundamental work document.	09/25/24
Draft v.02	SemiByte	Polishing and narrowing down specifications	--

1 Introduction

<TO DO: Please provide a brief introduction to your project and a brief overview of what the reader will find in this section.>

1.1 Document Purpose

This document outlines the software requirements for the Asia Pacific College (APC) Payroll System, specifically detailing the features and functionalities included in the current release. The purpose of this Software Requirements Specification (SRS) is to define the system's capabilities, user interactions, and operational constraints. It serves as a guide for developers, testers, and stakeholders by providing a comprehensive understanding of what the payroll system must accomplish. This includes processing employee salaries, calculating deductions, generating pay slips, and maintaining records of payroll-related data.

The scope of the APC Payroll System covered by this SRS focuses on key payroll functionalities. It includes core subsystems such as employee information management, salary computation, tax and benefit deductions, and the generation of reports for payroll administration.

1.2 Product Scope

The APC Payroll System is a software solution designed to automate and streamline payroll processes at Asia Pacific College. Its primary purpose is to accurately compute employee salaries, apply deductions for taxes and benefits, and generate pay slips. The system also maintains detailed records of employee earnings, tax contributions, and other payroll-related data, ensuring compliance with internal policies and government regulations. By automating payroll functions, the system minimizes human error, reduces the time spent on manual calculations, and simplifies reporting tasks for payroll administrators.

1.3 Intended Audience and Document Overview

This Software Requirements Specification (SRS) document is intended for a variety of readers, each with a different focus and use for the information contained within. Primarily, this document is targeted at the client, who is the Asia Pacific College (APC) administration, as well as the professor overseeing the project. The client will use this document to ensure that the system's requirements align with their needs for the payroll process, while the professor will review it for academic completeness and technical accuracy.

For developers, the SRS outlines detailed functional and non-functional requirements that will guide the design and implementation of the system. Testers will reference the document to define test cases that ensure all features meet the specified criteria. Project manager and documentation writers can use the SRS to ensure that milestones, deliverables, and system documentation align with the client's expectations.

1.4 Definitions, Acronyms and Abbreviations

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.

TO DO: Please provide a list of all abbreviations and acronyms used in this document sorted in alphabetical order.> (In need of more content to add in the list)

1.5 Document Conventions

<In general this document follows the IEEE formatting requirements. Use Arial font size 11, or 12 throughout the document for text. Use italics for comments. Document text should be single spaced and maintain the 1" margins found in this template. For Section and Subsection titles please follow the template.

TO DO: Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. Sometimes, it is useful to divide this section to several sections, e.g., Formatting Conventions, Naming Conventions, etc.>

1.6 References and Acknowledgments

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. >

2 Overall Description

2.1 Product Overview

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. In this part, make sure to include a simple diagram that shows the major components of the overall system, subsystem interconnections, and external interface. In this section it is crucial that you will be creative and provide as much information as possible.

TO DO: Provide at least one paragraph describing product perspective. Provide a general diagram that will illustrate how your product interacts with the environment and in what context it is being used. This is not a formal diagram, but rather something that is used to illustrate the product at a high level.>

2.2 Product Functionality

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary is needed here. These can be at the level given in the project description.>

TO DO:

- 1. Provide a bulleted list of all the major functions of the system*

2.3 Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software). You can be creative here to some degree. At a minimum, you need to identify that you must use the COMET method for software design and the UML modeling language. Make sure you provide references for both. >

2.4 Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project.

TO DO: Provide a short list of some major assumptions that might significantly affect your design.>

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. For your project, you only need to be concerned with the main thermostat (not the mobile app) and can use the graphic from the project description as the basis for your user interface..

TO DO: Provide the graphic for the thermostat user interface and provide a basic description as to how users will interact (e.g. touch screen, menus, etc.).>

3.1.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware. You are not

required to specify what protocols you will be using to communicate with the hardware, but it will be usually included in this part as well.

TO DO: Please provide a short description of the different hardware interfaces. This can simply be a list of the devices you must interact with at this point. For sensors, you can assume that each has a basic “read” interface to get the current values. Each sensor in this system uses English units.>

3.1.3 Software Interfaces

<Describe the connections between this product and other specific software components (in your case, just the mobile app that can send commands).>

3.2 Functional Requirements

< Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform. This section is the direct continuation of section 2.2 where you have specified the general functional requirements. Here, you should list in detail the different product functions.

3.2.1 F1: The system shall ...

3.2.2 <Functional Requirement or Feature #2>

...

3.3 Use Case Model

TO DO: Provide a use case diagram that will encapsulate the entire system and all actors.

3.3.1 Use Case #1 (use case name and unique identifier – e.g. U1)

TO DO: Provide a specification for each use case diagram

Author – Identify team member who wrote this use case

Purpose - What is the basic objective of the use-case. What is it trying to achieve?

Requirements Traceability – Identify all requirements traced to this use case

Priority - What is the priority. Low, Medium, High. Importance of this use case being completed and functioning properly when system is depolyed

Preconditions - Any condition that must be satisfied before the use case begins

Post conditions - The conditions that will be satisfied after the use case successfully completes

Actors – Actors (human, system, devices, etc.) that trigger the use case to execute or provide input to the use case

Extends – If this is an extension use case, identify which use case(s) it extends

Flow of Events

1. Basic Flow - flow of events normally executed in the use-case
2. Alternative Flow - a secondary flow of events due to infrequent conditions
3. Exceptions - Exceptions that may happen during the execution of the use case

Includes (other use case IDs)

Notes/Issues - Any relevant notes or issues that need to be resolved

3.3.2 Use Case #2

...

4 Other Non-functional Requirements

4.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.

TODO: Provide performance requirements based on the information you collected from the client/professor. For example, you can say “P1. The secondary heater will be engaged if the desired temperature is not reached within 10 seconds”>

4.2 Safety and Security Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied. Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements.

TODO:

- Provide safety/security requirements based on your interview with the client - again you may need to be somewhat creative here. At the least, you should have some security for the mobile connection.

4.3 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.

TODO: Use subsections (e.g., 4.3.1 Reliability, 4.3.2 Adaptability, etc...) provide requirements related to the different software quality attributes. Base the information you include in these subsections on the material you have learned in the class. Make sure, that you do not just write "This software shall be maintainable..." Indicate how you plan to achieve it, & etc... Do not forget to include such attributes as the design for change (e.g. adapting for different sensors and heating/AC units, etc.). Please note that you need to include **at least 2** quality attributes. You can Google for examples that may pertain to your system.>

5 Other Requirements

<This section is **Optional**. Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A – Data Dictionary/ERD

<Data dictionary is used to track all the different variables, states and functional requirements that you described in your document. Make sure to include the complete list of all constants, state variables (and their possible states), inputs and outputs in a table. In the table, include the description of these items as well as all related operations and requirements.>

Appendix B - Group Log

<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist in determining the effort put forth to produce this document>

Appendix C – Test Plan/Test Cases