Prof. Christopher Howard

CMIS 495 7980

‘SRS’

**Software Requirement Specification**

**For**

**[Project Name]**

*Version 0.1*

*03/14/2017*

*Prepared by:*

Ian Spooner

Wendy Velasquez Ebanks

Elvin Petrosy

Justin Mullins

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Version Number** | **Signature** |
| Ian Spooner | 03/14/17 | 0.1 | Ian Spooner |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Document Approval**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Version Signed** | **Signature** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of contents

**Revision History 1**

**Document Approval 1**

**Table of Contents 2**

1. Introduction 3

1.1 Purpose

1.2 Scope

1.3 Definitions

1.4 References

1.5 Documents

2. General Description 4

2.1 Product Perspective

2.2 Product Functions

2.3 User Characteristics

2.4 General Constraints

2.5 Assumptions and Dependencies

3. Specific Requirements 5

3.1 External Interface Requirements

*3.1.1 User Interfaces*

*3.1.2 Hardware Interfaces*

*3.1.3 Software Interfaces*

*3.1.4 Communications Interfaces*

3.2 Functional Requirements

*3.2.1 Built-In Calendar*

*3.2.2 Payment Acceptance and Guest Entry*

3.3 Use Cases

*3.3.1 Use Case #1*

*3.3.2 Use Case #2*

3.4 Classes and Objects

*3.4.1 Calendar Class*

*3.4.2 Guest Class*

3.5 Non-Functional Requirements

*3.5.1 Performance*

*3.5.3 Availability*

*3.5.4 Security*

*3.5.6 Portability*

3.6 Inverse Requirements

3.7 Design Constraints

3.8 Logical Database Requirements

4. Analysis Models 10

4.1 Architectural Context Diagrams

4.2 Data Flow Diagrams (DFD)

*4.3.1 Level 0*

*4.3.2 Level 1*

4.3 Entity Relationship Diagram (ERD)

4.4 State-Transition Diagrams (STD)

**1. Introduction**

1.1 Purpose

The purpose of this document is to provide the specifications development to design a timeclock system. Users shall be able to ‘clock in’ and ‘clock out’ in order to record the time and date, and hours worked. The system will function as an ‘electronic time sheet’ of sorts. This document will outline the specifications utilized in designing the system and is developed for the development management.

1.2 Scope

This document covers the design of the system and its components.

1.3 Definitions

- The system will hereafter be referred to as the [project name] or the Product.

- Developer, Designer, and Engineer shall all hereafter refer to components of the software development team named at the beginning of the document, or the team as a whole.

- User shall refer to the hourly employee needing to track time.

- Superuser shall refer to the administrative user performing supervisory functions.

-Shift shall refer to a tracked period of time.

- Punch In shall refer to the user entering credentials in order to start a shift.

-Punch Out shall refer to the user entering in order to stop time tracking for the current shift.

1.4 References

- IEEE Standard 1233-98: Guide for Developing System Requirements Specification

- IEEE Standard 1058-1998: Standard for Software Project Management Plans

1.5 Documents

Documents included:

* Architectural context diagram (ACD)
* Level 0 Data Flow Diagram (DFD)
* Level 1 Data Flow Diagram
* Entity Relationship Diagram
* Activity Diagram
* Use Case Diagram
* State Diagram

**2. General Description**

2.1 Product Perspective

2.2 Product Functions

2.3 User Characteristics

2.4 General Constraints

2.5 Assumptions and Dependencies

**3. Specific Requirements**

3.1 External Interface Requirements

This section is intended to specify the requirements for interaction between hardware, software, and the user.

3.1.1 User Interfaces

3.1.2 Hardware Interfaces

3.1.3 Software Interfaces

3.1.4 Communications Interfaces

3.2 Functional Requirements

3.2.1 Built-In Calendar

3.2.2 Payment Acceptance and Guest Entry

3.3 Use Cases

3.3.1 Use Case #1 – Use case name.

3.3.2 Use Case #2 – Non-Guaranteed Reservation is Dropped.

3.4 Classes and Objects

3.4.1 Class 1

* + 1. Class 2

3.5 Non-Functional Requirements

3.5.1 Performance

System shall perform with no delay, being based on simplicity; there should never be user interruption.

3.5.3 Availability

The system should always be available to the user, and shall rest in its most accessible state.

3.5.4 Security

3.5.6 Portability

3.6 Inverse Requirements

3.7 Design Constraints

.

3.8 Logical Database Requirements

**4. Analysis Models**

4.1 Architectural Context Diagrams

4.2 Data Flow Diagrams (DFD)

**4.2.1 DFD Level 0**

**4.2.2 DFD Level 1**

4.3 Entity Relationship Diagram (ERD)

4.4 State-Transition Diagrams (STD)