Software Requirements Specification

Version 1.0

June 21, 2020

Sun Kids’ system

Basel Bairkdar

Mohammad Hozifa Abbara

# Table of Contents

[Table of Contents i](#_Toc77487619)

[List of Figures ii](#_Toc77487620)

[1.0. Introduction 1](#_Toc77487621)

[1.1. Purpose 1](#_Toc77487622)

[1.2. Scope of Project](#_Toc77487623) 2

[1.3. Glossary](#_Toc77487624) 2

[1.4. References 2](#_Toc77487625)

[1.5. Overview of Document](#_Toc77487626) 2

[2.0. Overall Description](#_Toc77487627) 3

[2.1 System Environment](#_Toc77487628) 3

[2.2 Functional Requirements Specification](#_Toc77487629) 3

[2.2.1 Cashier Use Cases](#_Toc77487636) 3

[Use case: Register a Kid](#_Toc77487637) 4

[Use case: Login a Kid](#_Toc77487638) 5

[Use case: Logout a Kid](#_Toc77487639) 5

[Use case: Buy a Card](#_Toc77487640) 6

[Use case: Pay the Bill](#_Toc77487641) 6

[2.3 Non-Functional Requirements](#_Toc77487649) 8

[3.0. Requirements Specification](#_Toc77487650) 9

[3.1 Functional Requirements](#_Toc77487652) 9

[3.1.1 Register a Kid](#_Toc77487653) 9

[3.1.2 Login a Kid](#_Toc77487654) 9

[3.1.3 Logout a Kid](#_Toc77487655) 9

[3.1.4 Buy a Card](#_Toc77487656) 10

[3.1.5 Pay the Bill](#_Toc77487657) 10

[3.2 Detailed Non-Functional Requirements](#_Toc77487665) 11

[3.2.1 Logical Structure of the Data](#_Toc77487666) 11

[3.2.2 Security](#_Toc77487667) 12

# List of Figures

[Figure 1 - System Environment](#_Toc77487669) 3

[Figure 2 - Cashier Use Cases](#_Toc77487670) 4

[Figure 3 - Logical Structure of the Article Manager Data](#_Toc77487672) 11

# 1.0. Introduction

## 1.1. Purpose

The purpose of this document is to present a detailed description of the Sun Kids’ System. It will explain the purpose and features of the system, what the system will do, the constraints under which it must operate. This document is intended for both the stakeholders and the developers.

## 1.2. Scope of Project

This software system will be a local System for a local center in which the cashier can easily manage daily work at the center.

More precisely, the software allows the user to automate the process of kids’ entrance and exit allowing the parents to buy subscription cards with a specific amount of hours.

## 1.3. Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Cashier | The employee who uses the software and store desired info. |
| Parent | The person who brings the kid to the center, brings kid home and buys cards. |
| Entrance | The process of storing the kids’ entrance info. |
| Exit | The process of storing the kids’ exit info. |
| Card | A Subscription in the center with a certain amount of hours. |
| Software Requirements Specification | A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document. |
| Stakeholder | Any person with an interest in the project who is not a developer. |
| User | Cashier. |

## 1.4. References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

## 1.5. Overview of Document

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

# 2.0. Overall Description

## 2.1 System Environment

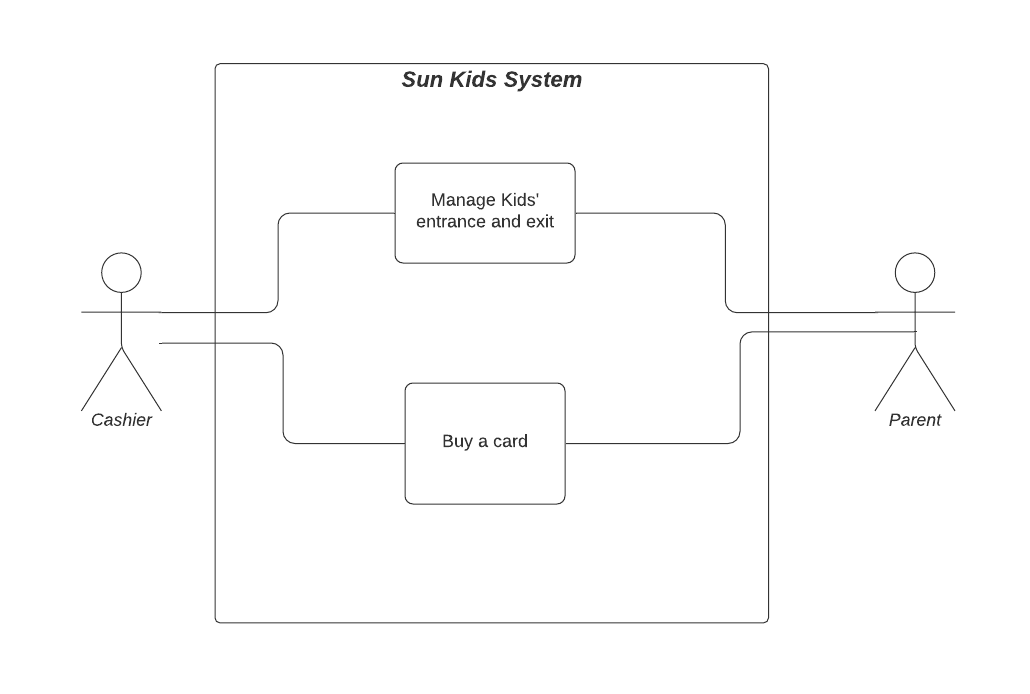


Figure 1 - System Environment

The System has two active actors.

The parent triggers the system, while the cashier handles storing required info.

## 2.2 Functional Requirements Specification

This section outlines the use cases for the Cashier.

### **2.2.1 Cashier Use Cases**

The Cashier has the following sets of use cases:

Register kid

Cashier

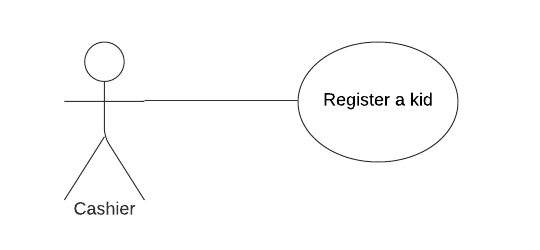
Manage kid’s entrance and exit

Buy a card

Figure 2 - Cashier Use Cases

#### Use case: Register a kid

**Diagram:**

****

**Brief Description**

The Cashier stores a kid’s info from the parent.

**Initial Step-By-Step Description**

Before receiving a kid in the center, all of his info must be registered for once.

1. The Cashier selects to *Add/Update kid’s info*.
2. The system presents a choice of adding or updating.
3. The Cashier chooses to add or to update.
4. If the Cashier is updating a kid’s info, the system presents a list of kids to choose from and presents a grid filling in with the information; else the system presents a blank grid.
5. The Cashier fills in the information and submits the form.
6. The system verifies the information and returns the Cashier to the main page.

**Xref:** Section 3.1.1, Register a Kid.

**Manage kids' entrance and exit use cases**

#### Use case: Login Kid

**Diagram:**

Cashier

Login Kid

**Brief Description**

The Cashier logs in a kid to the center.

**Initial Step-By-Step Description**

1. The Cashier selects to *log in a kid*.
2. The system presents a list of kids to choose from.
3. The Cashier chooses to log in a certain kid.
4. The system verifies the information and returns the Cashier to the main page.

**Xref:** Section 3.1.2, Login a Kid.

#### Use case: Logout Kid

**Diagram:**

Cashier

Logout Kid

**Brief Description**

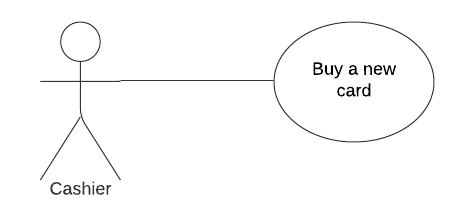
The Cashier indicate that a kid is leaving the center.

**Initial Step-By-Step Description**

1. The Cashier selects to *log out kid*.
2. The system presents a list of logged in kids.
3. The system presents the information of the chosen kid to verify the parent’s identity.
4. The Cashier logs out the kid.

**Xref:** Section 3.1.3 Logout a Kid.

#### Use case: Buy a Card

****

**Diagram:**

**Brief Description**

The Parent asks to buy a card so the cashier stores the kid’s info.

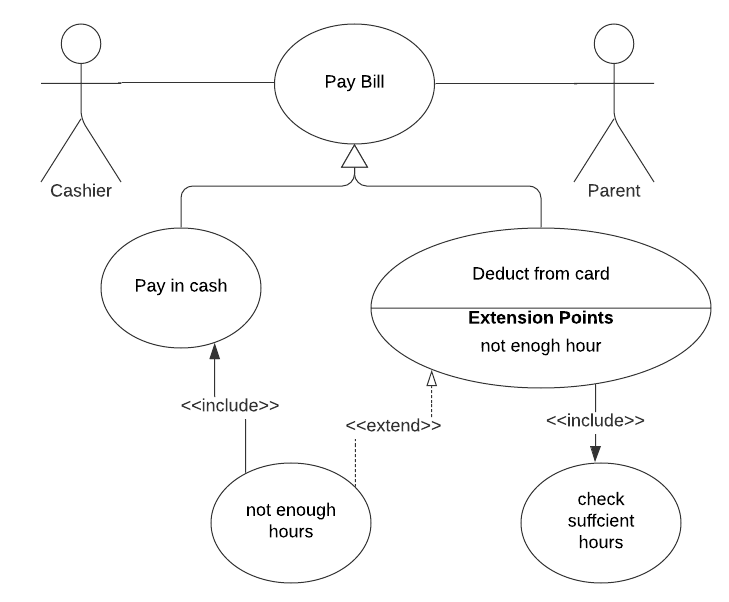
**Initial Step-By-Step Description**

1. The Cashier selects to *add a card* for a specific kid.
2. The system presents a list of registered kids.
3. The Cashier chooses the desired kid.
4. The system returns the Cashier to the main page.

**Xref:** Section 3.1.4 Buy a Card.

Use case: Pay The Bill

This use case is included in the *Logout Kid* use case.

**Diagram:**

**Brief Description**

The Cashier clicks on bill details once a kid is logged out.

**Initial Step-By-Step Description**

1. The Cashier selects to *Show Details*.
2. The system provides the entrance time and the amount of time the kid spent in the center.
3. The parent chooses to pay in cash or by deducting from a card; however, if the card does not have the required amount of hours, the parent can pay in cash.
4. The cashier marks the bill as paid.
5. The system returns the Cashier to the main page.

**Xref:** Section 3.1.5 Pay the Bill.

## 2.3 Non-Functional Requirements

The system will be run on the Cashier PC providing security for the

Kids’ data while having a high processing speed.

# 3.0. Requirements Specification

## 3.1 Functional Requirements

The Logical Structure of the Data is contained in Section 3.3.1.

### 3.1.1 Register a Kid

|  |  |
| --- | --- |
| **Use Case Name** | Register a Kid. |
| **XRef** | Section 2.2.1, Register a Kid. |
| **Trigger** | The parent asks to register his kid. |
| **Precondition** | Kid is not registered before. |
| **Basic Path** | 1. The Parent asks to register or updates his kid info. 2. The Cashier selects *register a kid.* 3. The Cashier store the required info (full name – father’s name – mother’s name – home address – school – birthdate – father’s phone – mother’s phone – home phone) |
| **Alternative Paths** | In step 2, if the parent chose to update info:  3- The Cashier stores the updated info. |
| **Postcondition** | None. |
| **Exception Paths** | None. |
| **Other** | None. |

### 3.1.2 Login a Kid

|  |  |
| --- | --- |
| **Use Case Name** | Login a Kid |
| **XRef** | Section 2.2.1, Login a Kid. |
| **Trigger** | The Parent brings the kid to the center. |
| **Precondition** | The Kid’s info are already registered in the system. |
| **Basic Path** | 1. The Parent brings the kid to the center. 2. The Cashier selects *Login a Kid.* 3. The Cashier store the required info (entrance time) |
| **Alternative Paths** | None. |
| **Postcondition** | None. |
| **Exception Paths** | Kid’s info are not in the system which leads to registration. |
| **Other** | None |

### 3.1.3 Logout a Kid

|  |  |
| --- | --- |
| **Use Case Name** | Logout a Kid |
| **XRef** | Section 2.2.1, Logout a Kid |
| **Trigger** | The Parent comes to bring the kid back home. |
| **Precondition** | The kid is already logged in. |
| **Basic Path** | 1. The parent comes to logout his kid. 2. The Cashier selects *Logout a Kid*. 3. The Cashier stores the required info (exit time) |
| **Alternative Paths** | None. |
| **Postcondition** | Paying the bill. |
| **Exception Paths** | None. |
| **Other** | None. |

### 3.1.4 Buy a Card

|  |  |
| --- | --- |
| **Use Case Name** | Buy a Card. |
| **XRef** | Section 2.2.1, Buy a Card. |
| **Trigger** | The Parent asks to dedicate a subscription card. |
| **Precondition** | The card is for a registered kid. |
| **Basic Path** | 1. The Parent demand a card. 2. The cashier stores the card required info (Kid’s record – amount of hours) |
| **Alternative Paths** | None. |
| **Postcondition** | None. |
| **Exception Paths** | None. |
| **Other** | None. |

### 3.1.5 Pay the Bill

|  |  |
| --- | --- |
| **Use Case Name** | Pay the Bill. |
| **XRef** | Sec 2.2.1 Pay the Bill. |
| **Trigger** | The Parent comes to bring his kid back home. |
| **Precondition** | The Parent logs out his kid. |
| **Basic Path** | 1. The Cashier logs out a kid. 2. The System generates the bill based on entrance time and amount of time spent in the center. 3. The parent pays in cash. 4. The Cashier marks the bill as paid. |
| **Alternative Paths** | In step 3, if the parent chose to pay by a card   1. The Cashier selects *deduct from card.* |
| **Postcondition** | None. |
| **Exception Paths** | If the Parent chose to pay by a card and the card does not have enough hours left, the Parent can pay in cash. |
| **Other** | None. |

## 3.2 Detailed Non-Functional Requirements

### **3.2.1 Logical Structure of the Data**

The logical structure of the data to be stored in the internal Article Manager database is given below.

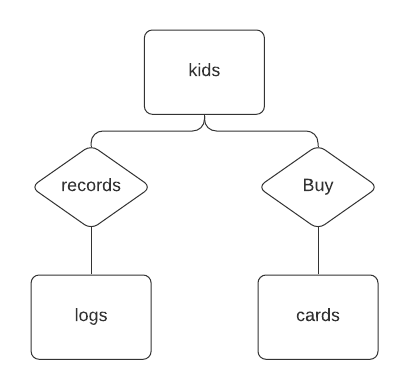


Figure 3 - Logical Structure of the Article Manager Data

The data descriptions of each of these data entities is as follows:

**Kids Data Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Full name | Text |  |  |
| Father’s name | Text |  |  |
| Mother’s name | Text |  |  |
| Father’s phone | Text |  | Can be changed |
| Mother’s phone | Text |  | Can be changed |
| Address | Text |  | Can be changed |
| School | Text | Current school | Can be changed |

**Logs Data Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Kid | Kid’s record |  |  |
| Entrance time | Time |  |  |
| Exit time | Time |  |  |
| Hours spent | Integer |  | Rounded up. |
| Bill | Integer |  |  |

**Cards Data Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Kid | Kid’s record |  |  |
| Hours left | integer |  |  |

### **3.2.2 Security**

The data will be stored encrypted and locally so it can be retrieved by a specific key.