<Project Name>

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 04/11/2019 | 1.0 | Add Problem Statement, Product Position Statement, Stakeholder Summary, User Summary and User Environment. | Nguyễn Lý Nhật Phương |
| 05/11/2019 | 1.1 | Add Summary of Key Stakeholder or User Needs. |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 3

1.1 References 3

2. Positioning 3

2.1 Problem Statement 3

2.2 Product Position Statement 3

3. Stakeholder and User Descriptions 3

3.1 Stakeholder Summary 3

3.2 User Summary 3

3.3 User Environment 3

3.4 Summary of Key Stakeholder or User Needs 3

3.5 Alternatives and Competition 3

4. Product Overview 3

4.1 Product Perspective 3

4.2 Assumptions and Dependencies 3

5. Product Features 3

6. Other Product Requirements 3

# Introduction

The purpose of this document is to collect, analyze and define high-level needs and features of the convenience store sales management software. It focuses on the capabilities needed by the stakeholders, and the target users, and why these needs exist. The details of how the convenience store sales management software fulfils these needs are detailed in the use-case and supplementary specifications.

## References

Collegiate sports paging system at sceweb.uhcl.edu

# Positioning

## Problem Statement

|  |  |
| --- | --- |
| The problem of | The difficulty of manage a large amount of product in the convenience store. |
| affects | The convenience store employees, store manager, the storage staffs and the customers. |
| the impact of which is | A slow and costly process combined with dissatisfied all the employees, staffs and manager. Sometimes can annoyed the customer. |
| a successful solution would be | Help the employees, the staffs, the manager manage the store more effectively and can attract more customers. |

## Product Position Statement

|  |  |
| --- | --- |
| For | The convenience store employees, the storage staffs and the store manager |
| Who | Work at the convenience store |
| The Convenience Store Sale Management Software | Is a tool |
| That | Help the employees track the inventory of the store, help the manager with statistical report |
| Unlike | Just use a manual cash register and paper report to manage the inventory |
| Our product | Provide the up-to-date information about the inventory of the store for the employees and storage staffs, statistical reports for the manager. |

# Stakeholder and User Descriptions

## Stakeholder Summary

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Responsibilities** |
| IT Executive | The development team | Responsible for the software development. |
| Store employee | Store employees | Ensures that the system will meet the needs of the store employees, who stand at the store and make the bill for the customer. |
| Storage staff | Storage staffs | Ensures that the system will meet the needs of the storage staffs, who import product and manage the inventory of the store. |
| Store manager | Store manager | Ensures that the system will meet the needs of the store manager, who has to manage the sale of the store. |

## User Summary

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Responsibilities** | **Stakeholder** |
| Store employee | Store employees | Queries for the information of the product in the store, calculate the bill for customer, sell the products. | self-represented |
| Storage staff | Storage staffs | Import and export products information. | self-represented |
| Store manager | Store manager | Queries for statistical reports, for products information and employees information | self-represented |

## User Environment

The conv*enience* Store User Community is a small community that demands the accuracy, fast and all-time work that a store management software can provide.

The store manager and storage staffs are educated, computer literate but not all the store employees are. The difficulty to read the product information and calculate the bill need to be low so that all the store employees can do.

The initial release of Conv*enience* Store Sale Management will be limited to small conv*enience* which just have one store and it can be used on the computer of the store which install the software. If the respond are well, the software will be designed to be expandable and can be use in the chain of conv*enience* store.

## Summary of Key Stakeholder or User Needs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Need** | **Priority** | **Concerns** | **Current Solution** | **Proposed Solutions** | |
| Know about the inventory of the conv*enience* | High | Check the inventory by paper is slow and inefficient. | Current store employees and storage staffs have to check the inventory by paper to know the amount of each product in the store. At this point, it very hard to track if the amount of product left is more or less to get the supplier in time so that not dissatisfied the customer. | | Employees and staffs would like to have the product data stored on the computer and can update each time the customer buy something easily. |
| Make report about the sale. | Medium | Long delay to get reports, don’t have the action in time when it need. | The store manager has to calculate the sale of the store each week, month, year by himself/herself and it very slow. He/she can’t know if the store has any problem soon to make the action. | | The manager would like to have a software to calculate and make report for him/her fast at the end of a week, a month, a year. |

## Alternatives and Competition

Nowadays, there are a lot of software support sales management . That is software freshales, hubspot, insightly, …

But All three software list above exist strengths and weaknesses by the stakeholder or end user.

|  |  |  |  |
| --- | --- | --- | --- |
| **Features** | **Freshsales** | **Hubspot** | **Insightly** |
| Unlimited users |  |  | X |
| Unlimited records |  | X | X |
| Lead Scoring |  | X | X |
| Integrations |  |  |  |
| Mobility |  |  |  |
| Support |  | X | X |
| Upgrade fee /month | X | X | X |

# Product Overview

[This section provides a high level view of the product capabilities, interfaces to other applications, and system configurations. This section usually consists of two subsections, as follows:

• Product perspective

• Assumptions and dependencies]

## Product Perspective

[This subsection of the **Vision** document puts the product in perspective to other related products and the user’s environment. If the product is independent and totally self-contained, state it here. If the product is a component of a larger system, then this subsection needs to relate how these systems interact and needs to identify the relevant interfaces between the systems. One easy way to display the major components of the larger system, interconnections, and external interfaces is with a block diagram.]

## Assumptions and Dependencies

[List each factor that affects the features stated in the **Vision** document. List assumptions that, if changed, will alter the **Vision** document. For example, an assumption may state that a specific operating system will be available for the hardware designated for the software product. If the operating system is not available, the **Vision** document will need to change.]

# Product Features

[List and briefly describe the product features. Features are the high-level capabilities of the system that are necessary to deliver benefits to the users. Each feature is an externally desired service that typically requires a series of inputs to achieve the desired result. For example, a feature of a problem tracking system might be the ability to provide trending reports. As the use-case model takes shape, update the description to refer to the use cases.

Because the **Vision** document is reviewed by a wide variety of involved personnel, the level of detail needs to be general enough for everyone to understand. However, enough detail must be available to provide the team with the information they need to create a use-case model.

To effectively manage application complexity, we recommend for any new system, or an increment to an existing system, capabilities be abstracted to a high enough level so 25-99 features result. These features provide the fundamental basis for product definition, scope management, and project management. Each feature will be expanded in greater detail in the use-case model.

Throughout this section, each feature will be externally perceivable by users, operators, or other external systems. These features should include a description of functionality and any relevant usability issues that must be addressed. The following guidelines apply:

• Avoid design. Keep feature descriptions at a general level. Focus on capabilities needed and why (not how) they should be implemented.

• If you are using the Rational RequisitePro toolkit, all need to be selected as requirements of type for easy reference and tracking.]

# Non-Functional Requirements

[Provide non-functional requirements that globally affect the product features descrived in the previous section.

At a high level, list applicable standards, hardware, or platform requirements; performance requirements; and environmental requirements.

Define the quality ranges for performance, robustness, fault tolerance, usability, and similar characteristics that are not captured in the Feature Set.

Note any design constraints, external constraints, or other dependencies.

Define any specific documentation requirements, including user manuals, online help, installation, labeling, and packaging requirements.

Define the priority of these other product requirements. Include, if useful, attributes such as stability, benefit, effort, and risk.]