**Bouncy Sports Equipment’s Inc**

**Requirements Analysis Document**

Dmytro Poliak

Student Number: 100443432

Date: 4.11.2024

1. **Introduction**

The purpose of this project is to develop a software for Bouncy Sports Equipment’s Inc. that manages in-store customer count and controls entry barriers to enforce occupancy limits during the Covid-19 pandemic. This system helps keep the store safe by making sure the number of customers never goes over the maximum limit.

**System Scope**

The system is designed to manage customer flow by utilizing sensors, barriers, and card readers within the store environment. It will track the number of customers inside the store, open/close entry barriers, and allow store personnel to bypass the barriers without impacting customer count. The central computer will serve as the core control point, receiving signals from sensors, managing barriers, and enabling store managers to monitor and adjust the maximum capacity limit.

**Context and Background**

This project, developed for Bouncy Sports Equipment’s Inc., addresses the need to control customer flow in-store, ensuring that occupancy stays within safe limits as per public health guidelines. The system will simulate core functionality using Java, focusing on counting customers, managing entry barriers, and allowing special access for store personnel. Designed as a console-based application, it runs entirely in the command line, with no database or graphical interface. This setup allows us to focus on applying software design patterns and object-oriented principles.

**Objectives and Success Criteria**

The primary objectives of this project are:

* Customer Count Management: Accurately track the number of customers inside the store using input from entry and exit sensors.
* Barrier Control: Control entry barriers to restrict or allow customer entry based on the current customer count relative to the limit.
* Store Personnel Access: Enable personnel to access the store without affecting the customer count.
* Manager Console: Provide store managers with a console to view customer count and adjust the in-store limit.

**Success Criteria**

The project will be considered successful if it meets the following criteria:

* The system reliably updates and maintains the correct customer count based on input.
* Barriers operate accurately according to the store’s occupancy rules, opening and closing at appropriate times.
* Personnel can bypass the barriers without impacting the customer count.
* Store managers have access to real-time customer count data and can adjust the occupancy limit effectively.

1. **Proposed System**

This section outlines the requirements gathered for the new system and presents an analysis model to define its functionality and structure.

The system will track and control the number of customers inside Bouncy Sports Equipment’s Inc. stores. It will automatically count people entering and exiting, manage entry barriers, and allow access for store staff. This setup runs in real time and simulates people-counting sensors, entry barriers, and a manager console through a console-based simulation. Main parts of the system handle customer counting, barrier control, store personnel access, and admin adjustments, creating a flexible and easy way to manage the store.

* 1. **Functional Requirements**

Functional requirements describe the high-level functionality of the system. These requirements focus on managing the in-store customer count, controlling barriers at entry points, handling store personnel access, and providing administrative functionality for the store manager.

1. Customer Counting:

* Detect customer entries and exits
* Maintain a real-time count of customers inside the store
* Update the count accurately as customers enter or exit

1. Barrier Control:

* Control entry barriers based on the current customer count relative to the maximum capacity
* Close entry barriers when the store reaches the customer limit
* Display a message indicating restricted entry due to capacity when the barrier is closed, and maximum number of customers is reached
* Reopen entry barriers automatically when the customer count drops below the maximum capacity

1. Store Personnel Access:

* Allow store personnel to bypass the barrier using a card reader
* Ensure that personnel entries and exits do not affect the customer count
* Temporarily open the barrier to allow personnel entry or exit and then close it after its use

1. Administrative Console for Store Manager:

* Provide the store manager with real-time access to the current customer count
* Allow the store manager to set or adjust the maximum customer limit
* Automatically adjust barrier status in response to changes in the maximum limit set by the manager
* Allow store manager manage/view active employees in the store
  1. **Nonfunctional Requirements**

Nonfunctional requirements describe user-level requirements that are not directly related to functionality.

1. Usability:

* The system should have a simple command-line interface, ensuring that it is easy to navigate and use by both store personnel and the store manager
* Commands should be intuitive, requiring minimal instructions for basic operations

1. Performance:

* The system should process input with minimal delay to ensure that customer counts and barrier actions are updated in real-time
* The response time for commands related to barrier control or customer counting should be under one second to avoid any noticeable delay

1. Supportability:

* The code should be modular and easily extendable, allowing for future adaptations or changes, such as modifying barrier control logic or adjusting entry/exit configurations
* The system should be built with clear separation of components, allowing developers to maintain and update specific modules independently

1. Reliability:

* The system should consistently track customer entries and exits without any error to ensure accurate in-store customer counts
* Barrier control must operate reliably, opening or closing based on real-time customer count and updated limits set by the store manager
* The system must allow store personnel to bypass the barrier reliably without affecting the customer count

1. Implementation:

* The system will be developed using Java and should operate entirely through a command-line interface, without requiring a database or graphical UI
* All in-memory data will be reset upon restarting the program, as no persistent storage is required

1. Interface:

* The interface will be text-based, providing prompts for the user to input commands and displaying system status updates and actions as plain text in the console
* The store manager interface should include clear instructions for viewing the customer count and adjusting the customer capacity limit

1. Security:

* The system should have basic access control to ensure that only authorized personnel can adjust the maximum customer limit. This may be simulated with a simple authorization prompt or password check for manager-level commands
* Store personnel should have unique identifiers (such as card IDs) to prevent unauthorized entries or exits from affecting the customer count
* All input validation should be handled carefully to avoid unintended actions, ensuring that only recognized commands trigger system actions
  1. **System Models**

This section provides a functional specification of the system, including typical user scenarios, a use case model, an object model with class diagrams, and dynamic models to represent system behavior over time.

* + 1. **Scenarios**

1.

A screenshot of a computer entry

Description automatically generated

2.

A screenshot of a computer error

Description automatically generated

3.

A screenshot of a computer entry

Description automatically generated

4.

A screenshot of a computer error

Description automatically generated

5.

A screenshot of a computer

Description automatically generated

* + 1. **Use Case Model**

A diagram of access control system

Description automatically generated

* + 1. **Object Model**

The Object Model provides a high-level view of the system’s main user-visible objects and their relationships. This model helps outline the essential components that interact within the system from the user’s perspective, without focusing on implementation specifics. The following key objects have been identified for the access control system at Bouncy Sports Equipment Inc.

Key Objects

1. Customer: Represents a person entering or exiting the store. This object interacts with the Sensor and Barrier to facilitate entry or exit actions.
2. StoreManager: Manages the store’s customer limit by setting and adjusting the maximum allowed capacity. The manager also has visibility over the current customer count.
3. StorePersonnel: Represents store staff who need to bypass the customer counter when entering or exiting. This object interacts with the Barrier using an authorization mechanism.
4. Barrier: Controls access to the store based on the number of customers. It can be in states like Open, Closed, or Temporarily Opened (for personnel).
5. PeopleCounter: Tracks the number of customers in the store and checks if the limit has been reached, triggering actions on the Barrier when necessary.
6. Sensor: Detects entries and exits, incrementing or decrementing the customer count accordingly.

Relationships

* Customer interacts with Barrier: When a customer enters or exits, the Barrier responds based on the current count.
* StoreManager manages PeopleCounter: The StoreManager can set the maxCapacity on the PeopleCounter.
* StorePersonnel bypasses Barrier: Store staff use an authorization mechanism to enter without impacting the customer count.
* PeopleCounter controls Barrier: The PeopleCounter monitors the customer count and signals the Barrier to open or close based on capacity.
* PeopleCounter connected to Sensor: The Sensor updates the PeopleCounter on customer movements.
  + - 1. **Class Diagram**

A diagram of a computer

Description automatically generated

* + 1. **Dynamic Model**

Activity diagram

Example of Entering the store (Same for Exit, but we decrease Customer Count)

A diagram of a system

Description automatically generated

Exit diagram

A diagram of a flowchart

Description automatically generated

Activity diagram from the program(barrier) perspective

A diagram of a system

Description automatically generated

Barrier State Diagram

A diagram of a closed process

Description automatically generated

Sequence Diagram

A diagram of a computer

Description automatically generated