Team 18 (Super Project X 3000)

1. **Alice Johnson**
   * **Student ID**: 123456
   * **Email**: alice.johnson@csulb.edu
2. **David Lee**
   * **Student ID**: 789012
   * **Email**: david.lee@csulb.edu
3. **Emily Rodriguez**
   * **Student ID**: 345678
   * **Email**: emily.rodriguez@csulb.edu
4. **Michael Patel**
   * **Student ID**: 567890
   * **Email**: michael.patel@csulb.edu
5. **Sophia Kim**
   * **Student ID**: 234567
   * **Email**: sophia.kim@ csulb.edu
6. **Daniel Nguyen**
   * **Student ID**: 901234
   * **Email**: daniel.nguyen@ csulb.edu

## Preface

(Describe the purpose of this document, its expected readership and its version history, including a rationale for the creation of a new version and a summary of the changes made in each version.)

|  |  |  |
| --- | --- | --- |
| Version | Date | Changes |
| 1.0 | 2/31/XXXX | Initial Version |
| 1.1 | 33th of Augtember, YYYY | Added Glossary |
| 2.0 | [Whenever] | Fixed version date notation, … |

### Purpose

This document serves as a comprehensive guide for the development and comprehension of the software project entitled “Super Project X 3000.”

### Audience

This document is intended for the following stakeholders: project managers, developers, testers, and all other individuals involved in the project lifecycle.

## Introduction

(Introduce the software project, its goals, and the problem it aims to solve)

### Project Overview

“ProjectX” is a comprehensive web-based inventory management system tailored specifically for small businesses. Its primary functions encompass the seamless tracking of inventory, efficient order management, and the generation of insightful reports.

### Project Goals

* Improve inventory accuracy.
* Enhance order fulfillment efficiency.
* Provide real-time reporting capabilities.

## Glossary

(Define key terms and acronyms used throughout your document, unless they are commonly known to each possible stakeholder (e.g., “Cell phone”) AND used with their common meaning. Do not expect your stakeholders to be experts. If in doubt, define a term.)

* **Inventory**: The stock of products available for sale.
* **SKU**: Stock Keeping Unit, a unique identifier for each product.
* **API**: Application Programming Interface.

## User Requirements and Use Cases

(Outline what the system must do from the user's perspective. User stories need to use the format discussed in class and on our slides. Use cases provide detailed scenarios of system interactions.)

### User Stories

(A collection of user stories that apply to the project.)

1. As a registered user, I want to log in securely so that I cannot be held responsible for someone else’s actions.
2. As an inventory manager, I need to add new products to the inventory so that the database matches the physical inventory.
3. As a sales manager, I want to generate sales reports by date range so that the upper management can be kept up to date on the company’s performance.

### Use Case: Adding a New Product

|  |  |
| --- | --- |
| Identifier | UC-2 Add new Product to Inventory |
| Purpose | Update inventory with new product name and quantity |
| Requirements | User Story #2 |
| Development Risks | None |
| Pre-conditions | Inventory manager is logged in and on inventory management page |
| Post-conditions | New product is added to inventory |

**Table 1: Typical Course of Action**

|  |  |  |
| --- | --- | --- |
| Seq# | Actor’s Action | System’s Response |
| 1 | Inventory manager selects “Add new product” |  |
| 2 | Enters product details (name, SKU, quantity) |  |
| 3 | Clicks “save” | Runs plausibility checks on submitted information |
|  |  | Checks are successful |
|  |  | Saves submitted information in inventory database |
|  |  |  |

**Table 2: Alternate Course of Action**

|  |  |  |
| --- | --- | --- |
| Seq# | Actor’s Action | System’s Response |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Table 3: Exceptional Course of Action**

|  |  |  |
| --- | --- | --- |
| Seq# | Actor’s Action | System’s Response |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## System Architecture

(Describe the high-level design of the software. List the components, name the architectural pattern you’ve used and provide a diagram.)

### Components

1. **Frontend**: Web-based user interface (built with React).
2. **Backend**: RESTful API (built with Node.js and Express).
3. **Database**: MySQL for data storage.
4. **Authentication**: OAuth 2.0 for user authentication.

### Architectural Pattern

We used the Pipe-and-Filter pattern for our system. Below is a diagram that shows how we applied it to our system.

