Business Requirement Documentation

Brandon Petersen

CS499: Computer Science Capstone

Southern New Hampshire University

May 19, 2023

# CS 255 Business Requirements Document

## System Components and Design

### Purpose

The Inventory Control application aims to help businesses efficiently manage their inventory and keep track of stock levels, item locations, and other relevant information. It enables companies to maintain accurate records of their products, streamline their operations, and make informed decisions regarding purchasing, stocking, and selling inventory. The potential clients for an inventory tracking application can vary across different industries and sectors. With that said, the inventory tracking application has been designed with the following three industries in mind: retailers, e-commerce businesses, and warehouse and distribution centers. These clients are looking for an inventory tracking application allowing centralized inventory management, real-time tracking, barcode scanning, integration with their current POS, and a user-friendly interface and experience.

### System Background

When clients use the Control Your Inventory application, they expect an inventory tracking system to fulfill various needs and solve specific problems related to inventory management. These problems include accurate inventory management, streamlined operations, inventory optimization, inventory tracking and traceability, and integration with their current POS. The Control Your Inventory application must consist of several components to ensure the application fulfills these clients' expectations. For example, the application will need a database to store all the products and details, a user-friendly interface, the ability to scan a product's barcode, integration capabilities, and security and access control. The specific components and features required may vary based on the industry, business size, and clients' unique requirements.

### Objectives and Goals

When the Control Your Inventory application is completed, it will be able to accomplish the following tasks:

1. Accurate Inventory Tracking: The application should track inventory levels, locations, and statuses in real time.

Measurable Task: A system will need to be designed and implemented that updates inventory quantities immediately upon sales, purchases, transfers, or adjustments.

1. Barcode Scanning: The application will support barcode or QR code scanning for efficient and more accurate inventory tracking.

Measurable Task: Integrate barcode scanning devices or enable smartphone camera scanning functionality. Test the scanning process for accuracy and speed.

1. User-Friendly Interface: The application will have a straightforward, intuitive, and user-friendly interface for easy navigation and efficient inventory management.

Measurable Task: Develop and design an intuitive user interface with clear menus, buttons, and search functionalities. Conduct user testing to ensure usability and gather feedback for improvements.

1. Integration with other Systems: The application seamlessly integrates with other business systems to ensure data synchronization and streamline workflows.

Measurable Task: Establish integration capabilities with POS, e-commerce platforms, accounting software, or ERP systems. Test and validate data synchronization between the application and other methods.

1. Scalability and Performance: The application should be designed to handle increasing inventory volumes and provide optimal performance.

Measurable Task: Conduct load testing to determine the system's performance under high load conditions. Optimize the database queries and system architecture for scalability.

1. Security and Access Control: The application will have robust security measures to protect inventory data and control user access.

Measurable Task: Implement data encryption and access control mechanisms to ensure data security. Conduct security testing and penetration testing to identify and address vulnerabilities.

## Requirements

### Nonfunctional Requirements

#### Performance Requirements

The choice of environment for the Control Your Inventory Application will start as a mobile application. From there, it will branch out to a web-based application. Depending on which environment the user selects will depend on several factors, including specific requirements, client preferences, and the nature of the business.

1. Mobile Application: A mobile application will develop to run on Android and iOS and similar tablet devices. This will provide the user on-the-go access to their inventory information and functionality. In addition, it can leverage device-specific features such as barcode scanning and GPS to track inventory data.
2. Web-Based Application: A web-based application can be accessed through a web browser, offering flexibility and ease of access from various devices without requiring software installation. It allows for centralized management and updates; users can access the application from any location with internet access.

The client should choose the application platform based on what system aligns with their business needs, infrastructure, and preferences. Often, a web-based application offers greater flexibility and ease of maintenance, especially for multi-user access and updates.

The application speed should be designed to provide efficient and responsive performance, ensuring that users can perform tasks without significant delays. Responsiveness can vary based on numerous factors, such as the size of the database, network speed, and hardware capabilities.

The application maintenance and update frequency will vary based on the client's evolving business needs, bug and error fixes, security patches, and introducing new features. Software updates can be released periodically, such as every month or quarter, depending on the application. In addition, security updates will require immediate releases and updates.

#### Platform Constraints

The application will primarily run on the Android and iOS platforms for mobile architectures. In addition, a web-based application will be developed. A web-based application will run on the user's system regardless of their architecture and system OS. This will make the application accessible from Windows, macOS, Linux, and mobile devices.

The application will require various backend tools, like a database, to store and manage inventory-related data. There are primarily two types of databases that would be utilized for this type of application:

1. Relational Databases offer structured storage, robust query capabilities, and transaction support.
2. NoSQL Databases: These databases suit scenarios where flexible schema designs or scalability is a top priority.

#### Accuracy and Precision

Since this application is intended for businesses to use instead of individual use, multiple users will be accessing the application. The application will implement user authentication and authorization mechanisms to distinguish between users. The user must log in with credentials such as a username and password. The system will then verify the provided credentials against a user database or directory to authenticate the user. In addition, usernames and passwords will be case-sensitive. The passwords will use a hashing encryption algorithm to protect sensitive information from malicious attacks.

When it comes to this type of application, numerous issues and problems could potentially arise. As a result, the application must communicate when an error has occurred through a system notification. In addition, some types of errors may require help from an admin account. These errors include security breaches, unauthorized access attempts, inventory stockouts, and system performance issues.

Overall, the system will be designed with the appropriate logging and notification mechanisms to capture and report any issues that may arise. These reports can then be accessed by admin accounts, who can take the necessary action to address the issue, troubleshoot, or ensure the smooth operation of the inventory tracking application.

#### Adaptability

The Control Your Inventory will allow admin accounts the flexibility to manage users without requiring changes to the underlying codebase. This type of feature will be implemented by using a user management module or functionality within the application's settings page. The user manages functionality will allow an admin user to add, remove, or modify user accounts, roles, and permissions through a user-friendly interface in the settings tab. By providing an admin account with this type of functionality, the application will allow flexibility to the ever-changing personnel without needing to modify the codebase.

As platforms change and update, the application has been designed with modularity and compatibility. This will enable more straightforward adaptation to platform updates, such as OS upgrades or changes to the fundamental technology the application is being used on.

An IT admin is one type of admin account that can be added to the user role. An IT admin will need administrative access to the application and its associated systems. An IT admin can perform user management, system configuration, troubleshooting and maintenance, and security management. The IT admin will need to be granted the appropriate privileges while considering security protocols and ensuring that they have the necessary rights to perform their duties effectively without compromising the integrity and security of the overall system.

#### Security

Security must be a top priority in the software development lifecycle when designing and developing the application. There are numerous ways to ensure the application has adequate protection in place. For instance, when users log in, they typically need to provide their credentials. These credentials will be a username and a corresponding password. The application will then verify these credentials against stored user data to authenticate the user. Numerous practices will be in place when a user is being authenticated:

1. Password Security: users create strong passwords that combine uppercase and lowercase letters, numbers, and special characters. In addition, all passwords that are stored in the system will go through a hashing algorithm before being held to encrypt the passwords.
2. Account Lockouts: After several failed login attempts have been reached within a specific timeframe, the account will be temporarily locked, taking additional steps to unlock such as contacting the IT admin.
3. Password Reset: If the user forgets their password, there will be a mechanism in place for the user to reset their password.

If the application has a brute-force hacking attempt, the account will be temporarily locked to prevent unauthorized access. The lockout period can be determined based on security policies, and the user may need to go through additional steps to regain access, such as confirming their identity by contacting the system administrator for assistance.

Implementing strong security measures to protect user data and prevent unauthorized access is crucial. Regular security audits, updates, and adherence to security best practices will also be followed to ensure ongoing protection against potential vulnerabilities.

### Functional Requirements

1. The system shall validate user credentials when logging in.
2. The system shall provide a user management module to add, modify, and remove user accounts.
3. Using a hashing algorithm, the system shall encrypt the connection between the client and the server.
4. The system shall implement account lockout mechanisms to prevent brute-force hacking attempts.
5. The system shall provide a password reset mechanism for users who forget their passwords.
6. The system shall generate detailed logs of login attempts, including successful and failed attempts, to aid in auditing and security analysis.
7. The system shall notify the admin of critical errors or exceptions that impact the functionality or stability of the application.
8. The system shall provide configurable user roles and permissions to control access to different features and functionalities.
9. The system shall log user activities and actions for audit and accountability purposes.
10. The system will accurately track inventory levels, locations, and statuses in real time.
11. The system will seamlessly integrate with other business systems to ensure data synchronization and streamline workflows.
12. The system will support barcode or QR code scanning for efficient and accurate inventory tracking.
13. The system will allow users to see the location of items and add, modify, and delete inventory from the database.
14. The system will provide users who are logged on their mobile devices with stock tracking alerts when an item reaches a certain on-hand quantity.

These functional requirements outline the desired behaviors and capabilities of the inventory tracking system related to user authentication, security, and basic functionality.