

Household Items Inventory Application

*Tracking Large or Expensive Household
Items for Milestone Decisions*

Milestone Decisions



08-FA25-SP26-MD-MOB

Ryann Lacks, Willow Zuvich, Justin Simm

9-26-25

TABLE OF CONTENTS

I.	Introduction	3
II.	System Requirements Specification	3
II.1.	Use Cases	3
II.2.	Functional Requirements	6
II.2.1.	User Management	6
II.2.2.	View History and Auditing	6
II.2.3.	Assignment and Tracking	7
II.2.4.	Inventory Management	7
II.3.	Non-Functional Requirements	8
III.	System Evolution	9
IV.	Glossary	9

I. Introduction

Milestone, located in Moscow, Idaho, is an organization dedicated to supporting adults with intellectual disabilities by providing resources that promote healthy and independent living. To meet state reporting requirements, Milestone identified a need for a more efficient way to track and manage their large item purchases.

In response, our team will develop a web application designed for use by Milestone staff, including both administrators and supervisors. This application will provide a streamlined system for recording, organizing, and maintaining an inventory of purchased items, ensuring accurate tracking and compliance with state regulations.

II. System Requirements Specification

II.1. Use Cases



Figure 1: Use case diagram

Use Case 1: Register New User

Actors	Administrator, New user
Description	Staff member registers for an account to access the inventory system.
Pre-condition	The new user is on the registration page for the application.
Post-condition	The user is either approved or denied their registration and sent an email.
Main Flow:	<ul style="list-style-type: none">-User enters their email and password-User selects "Create Account"-User is sent a verification email-User confirms email-Admin Approves the new account-User can now log in
Alternate Flows	<ul style="list-style-type: none">-Email verification fails -> User may resend email-Admin rejects new accounts -> New account is canceled
Related FRs	<ul style="list-style-type: none">• FR-01, FR-02, FR-07

Use Case 2: Assign Items to Households or Residents

Actors	Admin, Supervisor
Description	A user links purchased items to households or individual residents to track expenses.
Pre-condition	User is logged into the system and on the household management page.
Post-condition	The records for updated households and/or residents are updated in the system and reflected in the inventory.
Main Flow:	<ul style="list-style-type: none">- The user selects the option to assign an item- The user selects the item(s) to add- The items are added to the inventory of the resident or household
Alternate Flows	<ul style="list-style-type: none">- The item is already assigned to a resident or household -> The user is prompted to remove the item from the previous assignment
Related FRs	FR-04, FR-09, FR-10

Use Case 3: Add Household Items

Actors	Admin, Supervisor
Description	A user adds a new inventory item
Pre-condition	The user is logged in.
Post-condition	A new item is added to the database.
Main Flow:	<ul style="list-style-type: none">- The user selects the option to add a new item- The user enters details such as cost or condition- The user saves the new item- The item is added to the database

Alternate Flows	- Required information is missing -> The user will be prompted to add the missing information
Related FRs	FR-03, FR-04, FR-06, FR-08, FR-12, FR-13

Use Case 4: Delete Household Items

Actors	Admin, Supervisor
Description	A user removes an inventory item
Pre-condition	The user is logged in and on the items page.
Post-condition	A new item is archived until the buffer time runs out.
Main Flow:	<ul style="list-style-type: none"> - The user selects the option to delete an item - The user confirms deletion - The item is archived for up to 1 year before permanent deletion
Alternate Flows	<ul style="list-style-type: none"> - The user cancels deletion -> The action is canceled, and the item remains in the system
Related FRs	FR-03

Use Case 5: Edit Household Items

Actors	Admin, Supervisor
Description	A user edits or attaches metadata (cost, condition, image) to an item
Pre-condition	The user is logged in and on the items page.
Post-condition	The new item information is saved and updated in the database.
Main Flow:	<ul style="list-style-type: none"> - The user selects the option to edit an item - The user alters the cost, condition, description or photo - The user saves the item - The new changes are reflected in the database
Alternate Flows	<ul style="list-style-type: none"> - The user cancels the operation -> The action is canceled, and the item remains unchanged.
Related FRs	FR-03, FR-04, FR-06, FR-08, FR-12, FR-13

Use Case 6: Audit and Approve Users

Actors	Administrator, New user
Description	Administrator
Pre-condition	Admin approves or denies new user registrations.
Post-condition	Admin is logged into the system and on the new user validation page
Main Flow:	<ul style="list-style-type: none"> - The admin selects the pending account - The admin approves the new account - The new account is added to the system
Alternate Flows	<ul style="list-style-type: none"> - The admin rejects the new account -> The new account is thrown out
Related FRs	FR-07

Use Case 7: View Item History

Actors	Administrator, Supervisor
Description	User checks the log of changes made to an item (who edited, when, and what was changed).
Pre-condition	The user is logged in and on the item page.
Post-condition	NA
Main Flow:	<ul style="list-style-type: none">- The user selects the option to view the log- The view is displayed
Alternate Flows	NA
Related FRs	FR-05, FR-11

II.2. Functional Requirements

II.2.1. User Management

FR-01: User Registration

Description	The system shall allow users to register using a valid email and password.
Source	Client request for secure staff access.
Priority	Level 0.

FR-02: Confirmation Email

Description	The system shall send a confirmation email after successful registration.
Source	Client need for account verification.
Priority	Level 2.

FR-07: Admin Approval of Users

Description	The system shall allow administrators to approve or deny newly registered users before granting access.
Source	Milestone's requirement for controlled staff access.
Priority	Level 0.

II.2.2. View History and Auditing

FR-05: Viewing past logs/logging data

Description	The system shall allow users to view the logging history of items.
Source	Milestone need for state reporting
Priority	Level 1.

II.2.3. Assignment and Tracking

FR-04: Assign Items to Residents/Households

Description	The system shall allow users to assign an item to a resident and/or household.
Source	Milestone requirement for state reporting
Priority	Level 0.

FR-09: Household Attachment

Description	The system shall allow users to attach items within the system to a certain household.
Source	Client need for household – level tracking.
Priority	Level 0.

FR-10: Resident Attachment

Description	The system shall allow users to attach items to a specific resident.
Source	Client need for individual-level tracking.
Priority	Level 0.

II.2.4. Inventory Management

FR-03: Add and Delete Items

Description	The system shall allow users to add and delete items from the database.
Source	Client need for inventory tracking.
Priority	Level 0.

FR-06: Edit Item Information

Description	The system shall allow users to describe and edit information about an item.
Source	Client need for flexibility.
Priority	Level 1.

FR-08: Attach Images

Description	The system shall allow users to add an image to items.
Source	Client request for easier identification.

Priority	Level 1.
----------	----------

FR-12: Attach Condition

Description	The system shall allow users to attach or edit condition notes for each item.
Source	Client requirement for item maintenance.
Priority	Level 1.

FR-13: Attach Cost

Description	The system shall allow users to add cost information to each item for reporting purposes.
Source	Client need for financial reporting.
Priority	Level 0.

II.3. Non-Functional Requirements

- **NFR-01: Performance**
The system shall load the dashboard page in under 2 seconds under normal load conditions.
- **NFR-02: Scalability**
The system shall support up to 500 concurrent users without performance degradation.
- **NFR-03: Availability**
The system shall be available 99.5% of the time during business hours (8 AM–6 PM).
- **NFR-04: Security**
The system shall not expose protected information to unauthorized parties and shall encrypt sensitive data (passwords, resident details).
- **NFR-05: Maintainability**
The system shall be developed following coding standards (C# ASP.NET best practices) to ensure maintainability.
- **NFR-06: Data Integrity**
The system shall prevent duplicate item entries in the database.
- **NFR-07: Storage Capacity**
The system shall support storage of at least 5000 items without performance degradation.

III. System Evolution

This project is designed to remain a web application, ensuring accessibility across platforms without the need for specialized hardware or client-side installations. The system will be developed under a budget constraint of \$100, which limits reliance on costly third-party services.

The application is scoped to manage no more than 500 items in its database and no more than 500 concurrent users. These limits reduce the need for large-scale infrastructure and allow for a simpler hosting solution. However, if future requirements push the application beyond these thresholds, additional scaling considerations such as cloud hosting upgrades, database optimization, or caching solutions may be required.

A baseline level of user security will be maintained, particularly for account registration and authentication. While sufficient for the current scope, evolving security needs (e.g., multi-factor authentication, stronger encryption, compliance with regulations) may require future updates.

Anticipated change points and risks include:

- Hosting limitations: Current low-cost hosting may not support future scaling beyond the 500-user or 500-item thresholds.
- Budget constraint: With only \$100 allocated, there is little flexibility for unexpected costs such as licensing fees, performance upgrades, or third-party service requirements.
- Security evolution: As threats evolve, maintaining even a small user base securely may require enhancements beyond the original design scope.
- Changing user needs: If user demand exceeds assumptions (more users, larger datasets, or new functionality), the system's current design may require substantial rework.

By acknowledging these assumptions and risks early, the design avoids premature optimization while remaining open to future adjustments if project scope or user needs change.

IV. Glossary

Use Case: A description of how users interact with a system for defined purposes.