**Inventory Application Proposal**

**By Michael Reynolds**

**Application Goals**

The Inventory application’s main objectives will be tracking items in a warehouse, utilizing a QR/SKU code system for scanning items into a view to organize with. Keeping track of what’s in each container, as well as different types and shapes of objects in the warehouse by a square foot-age system. Providing intuitive views such as warehouses in a grid display, items within the warehouses and add and remove functionality, where the user will be able to quickly manage items. Utilizing various databases and localized data to achieve persistence and providing push notifications to keep the end-user updated on inventory as well as lack-of inventory. Small unit management will be considered, and other storage solutions such as refrigerated units and restaurant FIFO/Health code will also be strongly considered.

**Target Audience**

This application will be targeting Small to Medium sized businesses, where inventory is not micro-manageable but also fleet management can be abstracted away from warehouse management. Specifically, this application will be utilized by end-users who are warehouse workers and managers, who need a compact digital view of their warehouse that sums up their inventory and ensures they can integrate their management into other parts of business operations. Although it should be mentioned overall that potential users for this application will include warehouse managers, restaurant owners, and individual storage unit owners. Each of these users have storage management needs, but on various scales and of various sizes. Assumedly warehouse managers will be much more interested in an organized and thoroughly tracked system, where many employees will be at play, and so having a system for visualizing live storage and ensuring correct placement/accountability is vital.

**Screens & Features**

There will be five screens, the first being a dashboard for quick accession and panels that allow a general summarization of user data. The second screen will be a warehouse(s) grid view, this will contain a grid of card panels that show registered storage units/warehouses and quick data regarding their capacity and categorization. The third view will be a specific warehouse view, this will contain a list of inventories, as well as a 3D view of the warehouse using simple box objects to help visualize inventory placement. The fourth view will be a login/register screen with basic input fields and buttons. The final view will be a profile view, with many dropdowns for settings/login, localization, logout function, as well as billing, product help resources and more.

**Functional Requirements & UI**

For functional requirements and UI, we’ll be strictly following the Model-View-Controller pattern or the MVC pattern. For the dashboard screen, we’ll be connecting to data sources such as warehouse data, inventory quantity, and use that to display into the view of the panels/quick accession views found on the dashboard activity view. Users will be able to navigate through the app via these panels. This same functionality can be achieved through the grid view.

The users will be able to login on the login/register screen. This view will contain UI elements such as input fields and buttons, where the user can then input their username and passwords. This will then authenticate through the appropriate controllers and models.

In the specific warehouse view, we’ll be using 3D models on a isometric view of the warehouses/storage unit spaces, our list view will be filled with per-item row data which will be interactable by the user to change quantity/item info/item image. In the 3D view, the end-user will be able to move object by grid and with appropriate width, height, length, they’ll be able to move objects and see the results happen in real time in this view.

For the user profile view, they will be able to interact with drop down menus, most of these options will be in radio format, where the user can change user data found locally on the device, for things like localization, application theme, and settings for accessibility like font size and color blindness. This will serve as a form, storing preferences locally.