No Kill Inventory

Programmer’s Manual

Spring 2024

**Vision Statement**

No Kill Inventory is a simple web app designed to help No Kill Louisville keep track of their inventory of pet food that they collect. It allows them to see this information at a glance and add or remove food as needed.

**Introduction**

The inventory system is designed specifically for No Kill Louisville. It provides an easy way to track inventory as it goes in and out, which allows them to have more accurate records and make decisions based on that.

**Component Overview**

* Frontend
  + UI – The user interface seen in the web browser by users.
* Backend
  + Razor Pages – Individual webpages, includes all their functionality and visuals.
  + Database – SQLite database that holds all the data input to the inventory.

**Tools Overview**

* C# - The primary programming language for the backend and frontend functionality.
* HTML – Language that defines objects and elements of the UI.
* CSS – Language that creates the visual style for HTML elements.
* EF Core – Object relational mapper used to interact with the database using C#.
* SQLite – The type of database chosen for the project. All in 1 file and library, no server needed.
* .NET 8 – The platform, runtime, and framework used for the project.
* Blazor – Frontend web framework the project is built on.

**Project Repository**

* Software
  + <https://github.com/RiggsJaM/No-Kill-Louisville-Inventory-System-CapstoneF23>
* Test Cases
  + See NoKill-Inventory/No-Kill-Inventory/Testing for test cases.
* Documentation
  + See README and the Documents folder for documentation.
* Test Platform Description
  + Front end testing done manually by the developers.
  + Backend and database done by combination of manual testing and BUnit + MSTest.
* Test Scripts
  + See NoKill-Inventory/No-Kill-Inventory/Testing.

**Installation For New Install**

* See README.md

**Installation For New Platform**

* See README.md

**Further Development Statement**

Future improvements to the Inventory System would include integration of QR codes to scan items instead of having to manually input them, as well as more front-end improvements. This would make it easier for users to input items and clean up some of the UI, as many of the manual elements could be hidden until needed.