Smart Inventory

Michael Chen
Oriana Martin
Michael Stallbaumer

Functional System Requirements

REVISION – 1
20 February 2023

FUNCTIONAL SYSTEM REQUIREMENTS
FOR
Smart Inventory

PREPARED BY:

Author Date

APPROVED BY:

Project Leader Date

John Lusher, P.E. Date

T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
1	02/20/2023	Michael Stallbaumer	Oriana Martin	Draft Release

Table of Contents

List of Tables

List of Figures

1. Introduction

1.1. Purpose and Scope

The smart inventory system will be an application that intends to replace the existing inventory management system used by the capstone lab. This shall be done through the use of a machine learning (ML) model that will perform image recognition on lab equipment. The ML shall have an accuracy of at least 90%. This ML system will be used to aid TAs in identifying equipment and informing them of information relevant to the identified equipment. The application will have a database that will store all of the relevant information for students using the application and the equipment that is inventoried. The student information shall include their permissions, what equipment they are and are not allowed to check out. The application shall have a user interface (UI) that will inform students if they do not have the required permissions to check out a piece of equipment as well as direct them to the process of gaining the permissions to check the equipment out. Students shall be able to request equipment through the application before arriving at the lab and TAs shall be able to verify or deny these requests as well as check-ins. The application shall also notify users of changes to their stored information. The application should also have different UI for users depending on their status, being TAs and professors or students. The prototype will be tested on a small sample size of users and items before being implemented in the capstone lab. Figure 1 shows a conceptual programming flow chart of the proposed application in the CONOPS. The verification requirements for the project are contained in a separate Verification and Validation Plan.

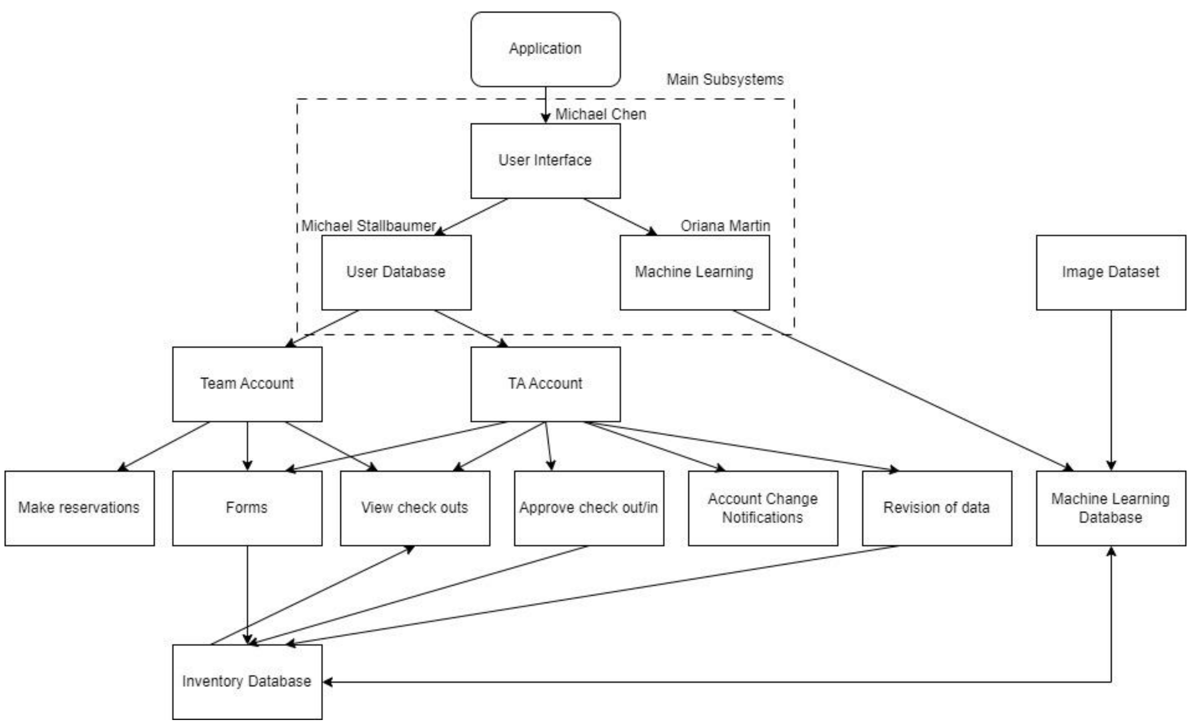

Figure 1. Smart Inventory Conceptual Programming Flow Chart

1.2. Responsibility and Change Authority

Our team leader, Michael Stallbaumer, is responsible for validating that subsystem requirements are made. Changes to the design specifications and requirements are to be approved by both the team leader, Michael Stallbaumer, and the project sponsor, Eric Robles. Each team member is also responsible for their own progress on the development of their subsystems as well as verifying that the systems fulfill the design requirements. Subsystem delegation can be seen in the following table.

Table 1. Subsystem Responsibilities

Subsystem	**Responsible Party**
Application UI	Michael Chen
Machine Learning Model	Oriana Martin
Database Management	Michael Stallbaumer

2. Applicable and Reference Documents

2.1. Applicable Documents

The following documents, of the exact issue and revision shown, form a part of this specification to the extent specified herein:

Table 2. List of Applicable Documents

Document Number	Revision/Release Date	Document Title
IEEE 0-8186-3570-3	04/19-23/1993	Data quality requirements analysis and modeling
IEEE 0-8186-3570-3	04/19-23/1993	A language multidatabase system communication protocol
IEEE 0162-8828	03/31/2008	Framework for Performance Evaluation of Face, Text, and Vehicle Detection and Tracking in Video: Data, Metrics, and Protocol

2.2. Reference Documents

The following documents are reference documents utilized in the development of this specification. These documents do not form a part of this specification and are not controlled by their reference herein.

Table 3. List of Reference Documents

Document Number	Revision/Release Date	Document Title
IEEE 978-1-4673-0242-5	07/14-17/2012	App interface study on how to improve user experience
IEEE 978-1-5386-5457-6	10/28-31/2018	Machine Learning for Object Labeling
IEEE 978-1-6654-2317-5	07/19/2021	Designing an Inventory Database Software Suitable for Small Business

2.3. Order of Precedence

In the event of a conflict between the text of this specification and an applicable document cited herein, the text of this specification takes precedence without any exceptions.

All specifications, standards, exhibits, drawings or other documents that are invoked as “applicable” in this specification are incorporated as cited. All documents that are referred to within an applicable report are considered to be for guidance and information only, except ICDs that have their relevant documents considered to be incorporated as cited.

3. Requirements

3.1. System Definition

The Smart Inventory system will be an application comprising three main subsystems, the application UI, ML model, and database management. These three systems will be closely interconnected.

The application UI will be the method of control for the overall application, sending commands to both the database management and the ML model subsystems. The UI shall have a login that should differentiate between student and TA accounts through communicating with the database management, determining the UI that will appear for the application. The student UI will allow students to create equipment check out requests.The student UI will also have the ability to view personal user information, including permissions and currently checked out equipment and unverified requests. The TA UI will allow the TA to see and edit all user and equipment information as well as approving and denying check outs and check-ins. The TA UI should also have an option for revising the databases incase of error and pushing notifications to accounts. The TA UI shall include access to the ML image recognition system which shall inform the user of the equipment's database information.

The database management system will comprise three other subsystems, a user database, inventory database, and database management functions. The user database will contain each user's login information, a user ID, administration status, check out permissions, name, team number, contact information, and pending checkout requests. The inventory database will contain each equipment's inventory ID, amount in the inventory, storage location, category, amount available for check out, required permissions, and a list of users that currently have the equipment checked out. Both databases will be editable by administrative accounts. These databases will also both be controlled by the data management functions. These functions will be created so that navigating and editing the databases can be handled by the UI using clear functions.

The Machine Learning system will comprise an image labeling API and a label database. Using the image labeling API, it will be possible to take an image of a component and extract information regarding the items in the image. This portion of the application will only be accessible to TAs in order to aid in the classification of physical inventory items. For the TAs this action will run in the background so that it does not interfere with other operations of the application. For the dataset that the ML model is trained on, it must contain both images taken of the inventory items as well as supplemental images acquired through online dataset websites. The database that will be used to label the images will contain the most up-to-date information regarding the category of the items. In order for this to be accurate, the database will check the inventory database to obtain the most current labels.

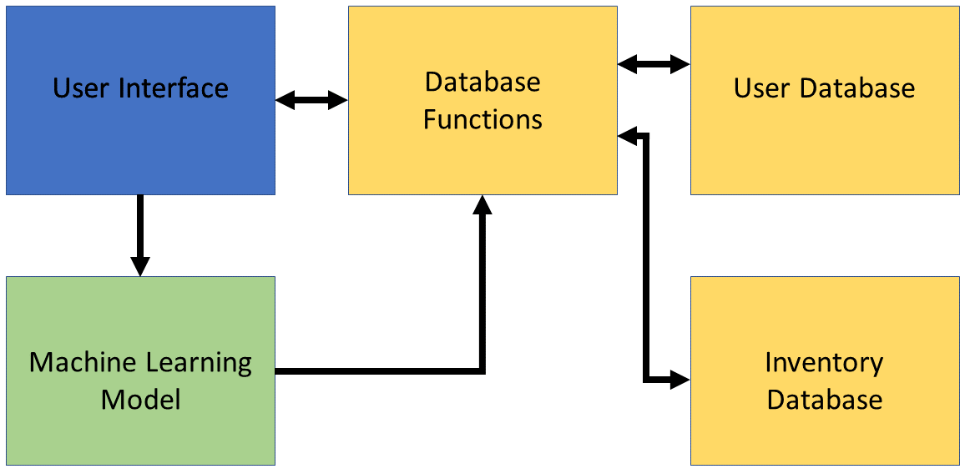

Figure 2. Block Diagram of System

The UI subsystem will control the rest of the system. It will do this by communicating with both the ML model and the database functions. The UI will send commands to the ML model to activate the image recognition based on user input to the UI. When the ML model identifies an item it will send the equipment ID to the database functions which will then send the equipment information stored in the inventory database to the UI to be displayed. The UI will also call functions that are defined in the database functions subsystem which will read, add, change, or remove information stored in the databases.

3.2. Characteristics

3.2.1. Functional / Performance Requirements

3.2.1.1. ML Model Accuracy

The ML model shall correctly identify equipment at least 90% of the time as requested by the project sponsor.

> *Rationale: This is a set requirement by the sponsor. Inaccurate identification of equipment can lead to equipment being stored in the wrong locations and or incorrectly tracking the status of equipment.*

3.2.1.2. Equipment classification

The application shall be able to classify equipment by its category.

Rationale: The intent of the application is to aid in inventory management. The sponsor requests that the application shall inform administrative users of the category of equipment so that it will be stored in the correct location.

3.2.1.3. Secure Accounts

The application shall have secure user accounts for each user.

Rationale: Accounts must be secure to ensure that the correct user has the equipment checked out to them, making them responsible for the return of the equipment. This will also ensure that users can only check out equipment that they have the permissions to use.

3.2.1.4. Reservations

Users will be able to schedule equipment reservations before arriving at the lab if they have the correct permissions.

Rationale: Time spent checking items out will be reduced as TAs need only to approve or deny reservation requests and users will be aware of equipment availability and restrictions before arriving at the lab.

3.2.1.5. Restrictions

The application will handle equipment restrictions. This will entail preventing users from checking out equipment they do not have the permissions for and redirecting them to information on how to acquire the needed permissions.

Rationale: With the application handling the permissions and redirecting users to information about gaining them, users will have a clear understanding of why they cannot use equipment if they lack the permissions as well as how to gain them.

3.2.1.6. Validation

TAs will validate check outs and check-ins of equipment through the application.

Rationale: This will help mitigate any errors the application may have in checking permissions or identifying equipment. With this check in place, it is much less likely that the database will contain incorrect information.

3.2.2. Software Characteristics

3.2.2.1. Inputs

The users should be able to input account data and receive access to the system. This is true for both the student and TA accounts. For the ML model the user should be able to take an image and have the model read it.

Rationale: These are requirements set by the sponsor in order for users to be able to utilize the application

3.2.2.2. Make Equipment Reservations

The users should be able to be able to access information on the inventory bases in order to request the use of lab equipment. This request will then be sent to the TA accounts for approval.

Rationale: This is a requirement set by the sponsor to be able to manage reservations on the application

3.2.2.3. Manage Restrictions

If a user attempts to check out equipment that they do not have the proper permissions for, the application will reject the reservation and direct the user as to how to gain the permissions they do not have.

Rationale: This is a requirement by the sponsor as users should only be able to reserve equipment that they have access to.

3.2.3. Failure Propagation

The Smart Inventory application will have failure protection through user interaction and automated checks.

3.2.3.1. Failure Detection, Isolation, and Recovery (FDIR)

3.2.3.1.1 Built In Test (BIT)

The Machine Learning Model will have both a training dataset as well as a testing dataset. With the testing dataset, it will be possible to see if there are any errors with image labeling.

3.2.3.1.1.1 BIT Critical Fault Detection

The BIT for machine learning should be able to detect a critical fault 90 percent of the time.

Rationale: The machine learning model is required by the sponsor to be 90 percent accurate.

3.2.3.1.1.2 BIT False Alarms

The BIT should have a false alarm rate of less than 10 percent.

Rationale: This is a requirement by the sponsor to achieve high accuracy in the ML model

3.2.3.1.1.3 BIT Log

The BIT shall save the results of a test into a log that will be accessible by TAs for retrieval and maintenance of the databases.

Rationale: This is a requirement by the sponsor so that it is possible to to track the accuracy of the ML model.

3.2.3.1.2 Isolation and Recovery

The machine learning model should provide for fault isolation and recovery by resetting upon the result of the BIT in order to protect the other subsystems.

Rationale: This is good practice to have in order to protect all facets of the system from fault propagation

4. Support Requirements

4.1.1. Android Device

The Smart Inventory system will require the use of an Android smart device. This device, likely a smartphone, will require a camera and to be running Android Marshmallow 6.0 or newer. This device will also require the ability to connect to the internet.

Rationale: The application will be implemented on Android and requires a camera to perform the ML image recognition. The application also requires internet connectivity to read and write to the database.

4.1.2. Internet Connectivity

The Smart Inventory system will require the users to have access to the internet while accessing the application.

Rationale: Having WiFi at the location of implementation will ensure that the application can be accessed by users.

Appendix A: Acronyms and Abbreviations

ML	Machine Learning
API	Application Programming Interface
BIT	Built-In Test
GUI	Graphical User Interface

Appendix B: Definition of Terms

Database	A set of structured, digital data that is stored and accessed digitally.
Graphical User Interface	A visual form of a user interface that uses elements such as icons and menus for interacting with the user.
Image recognition	The process of identifying an object through previously images of the object.
Inventory	A list of all items that are contained within some classification.
Machine Learning	The use of a program that is able to learn from data that it receives through algorithms and data patterns.
User Interface	Some means by which a user interacts with software.