Research Labs Inventory

Jonathan Moore
Evan Ross
Lizzett Tapia
Sumedha Bhattacharyaa

Concept of Operations

REVISION – Draft
10 September 2024

Concept of Operations for Research Lab Inventory

Team <55>

Approved by:

Project Leader	Date
Prof. Kalafatis	Date
T/A	Date

Change Record

Rev.	Date	Originator	Approvals	Description
1.0	9/10/2024	Moore,Ross, Tapia, Bhattacharyaa		Initial Submission
2.0	9/26/2024	Moore		Format Changes

Table of Contents

Table of Contents III
List of Tables IV
List of Figures IV
1. Executive Summary 6
2. Introduction 7
2.1. Background.....7
2.2. Overview.....7
2.3. Referenced Documents and Standards.....8
3. Operating Concept 9
3.1. Scope.....9
3.2. Operational Description and Constraints.....9
3.3. System Description.....9, 10
3.4. Modes of Operations.....10
3.5. Users.....10
3.6. Support.....10
4. Scenario(s) 11
4.1. Check-Out as Student.....11
4.2. Check-In as Student.....11
4.3. Monitor Students as Staff.....11
4.4. Add/Manage Inventory as Staff.....11
5. Analysis 12
5.1. Summary of Proposed Improvements.....12
5.2. Disadvantages and Limitations.....12
5.3. Alternatives.....12
5.4. Impact.....12

List of Tables

Table 1: Referenced Documents……………………………………….8

List of Figures

Figure 1: Visual Functional Diagram……………………………..7

Figure 2: Landing Page of Web-app………………...……………...9

Figure 3: System Layout Description………………...…………10

1. Executive Summary

An important aspect of a research lab is the ability to track inventory. This is important for many reasons such as the ability to know when an item inventory is low, tracking who is responsible for what items, and knowing what items are available. Our project at its core is to develop a website, along with a smartphone app, that allows users to track and make changes to the Inventory in research labs. Users will need to login using their credentials, and they will either be categorized as a student or a staff member based on initial registration. Students will be able to check out and check in items in the lab, and the inventory will adjust accordingly. Staff members will be able to make larger changes to the inventory, allowing for the addition and deletion of large amounts of components. Our Inventory data will be stored on a database that will be linked to both the app and the website, so that changes can be made from both. Our smart phone application will also have a machine learning component where users will be able to scan an item and have it automatically identified. After scanning an item, the app will open the page where users can either check out or check in that specific item. This will make tracking components in laboratories easy and intuitive.

2. Introduction

The current lab inventory system existing is inefficient and disorganized. Lab members often go to check things out, and are met with the issue of not knowing how many things they checked out, and how much stock the lab has left of items. The current system has a person manually check things out to members, without much of a record. As a consequence, the lab lacks cohesive organization, and its members face challenges in maintaining an efficient workflow. We aim to tackle this simple logistical issue with a highly methodized Lab Inventory Tracking system.

2.1. Background

This Lab Inventory Tracking system aims to replace the manual process of logging items being checked in or out. The system comprises both a website and app that will have a live tracking method that can show what's checked out by whom. The app will also have a Machine Learning component that allows users to use their device's camera to locate items' locations. This will enhance the current system's efficiency as this system can update the log faster than it is being done manually. Replacing the manually updated log, with a live-updated one, allows users to also know the stock and status of items they may need.

2.2. Overview

To check-out an item, a user can direct themselves to either the app or website. Through either, they can add however many items they need to a cart. They will then be prompted to check a box that agrees that they will return items in the condition needed (if it's an item that needs to be returned). The app version will allow users to take a picture of all the items they're checking back in. The machine learning model will then automatically log those items as being checked back in. This process can be shown visually in **Figure 1**.

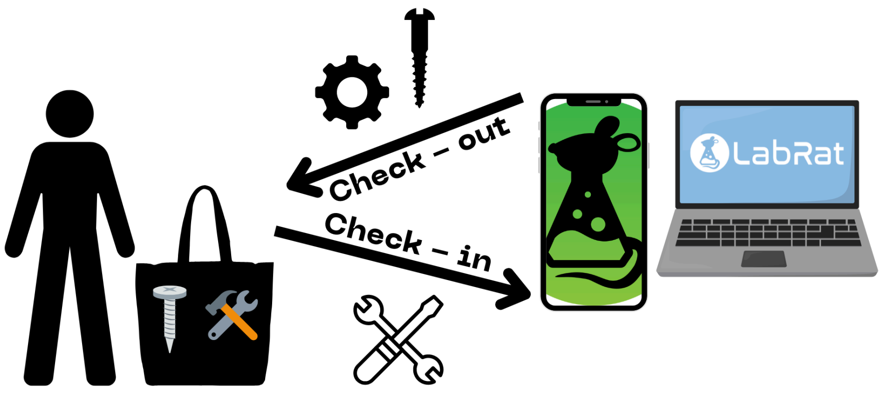


Figure 1. Visual Diagram of how the app and website will work synonymously for checking items in and out.

2.3. Referenced Documents and Standards

Table 1. Referenced Documents

Document Name	Revision/Release Date	Publisher
Google’s Machine Learning Crash Course	2023	Google Machine Learning Education
Building a Python Image Recognition System	2024	Cloudinary
OpenCV Library	4.10.0	OpenCV
Flutter document	2017	Flutter

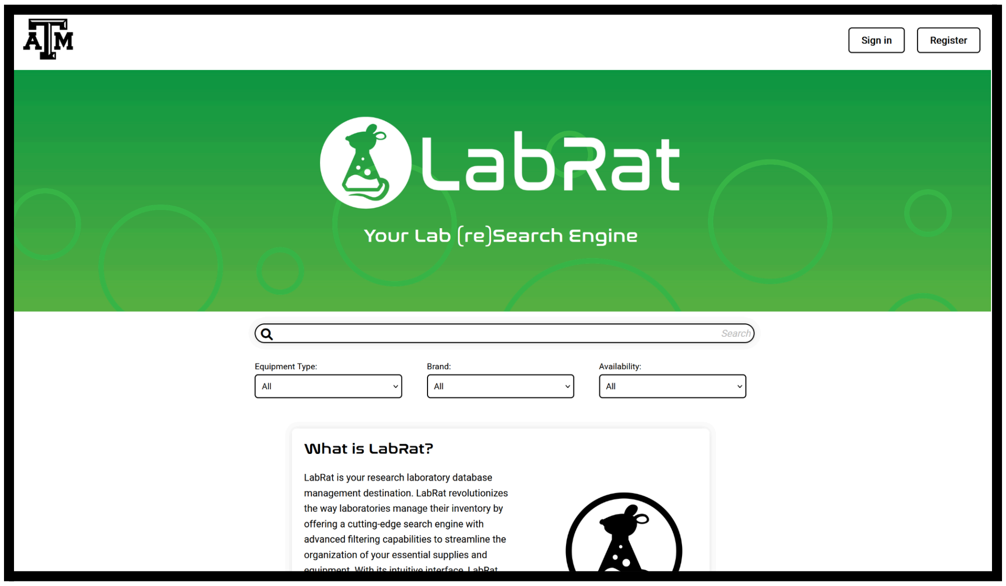
3. Operating Concept

3.1. Scope

The scope of our project is centered around people at Texas A&M. The intent of this project is for research students and faculty to use the application to find items in their associated lab. We will include functionality for other labs at Texas A&M to register their inventory and users, so any Texas A&M labs that aren't included in the initial design can make use of our product. Theoretically we can add functionality for labs outside of Texas A&M to use our services, but it was a deliberate decision to limit our scope to people at this university.

3.2. Operational Description and Constraints

Our project has two interfaces, a native mobile application and a web-based application. From the web-app, users will be able to log in and access all inventory items in the database for their assigned lab. If a user has access to multiple labs on campus, then they will be able to switch access between each for individual access.



What is LabRat?

LabRat is your research laboratory database management destination. LabRat revolutionizes the way laboratories manage their inventory by offering a cutting-edge search engine with advanced filtering capabilities to streamline the organization of your essential supplies and

Figure 2: Landing Page for Web-app

From the mobile-app, users will be given the same abilities as the web-app in addition to an item-finder. If a user cannot find more of a certain item, they can take an image of what they want to find more of and the machine learning model will help the user. Assuming that there are no barcodes or QR codes on the items, items can be found in the laboratory using this model.

3.3. System Description

The project is divided into four subsystems. These consist of the database, the web-app, the mobile application, and a machine learning model. Below is a visual representation of how each part interacts:

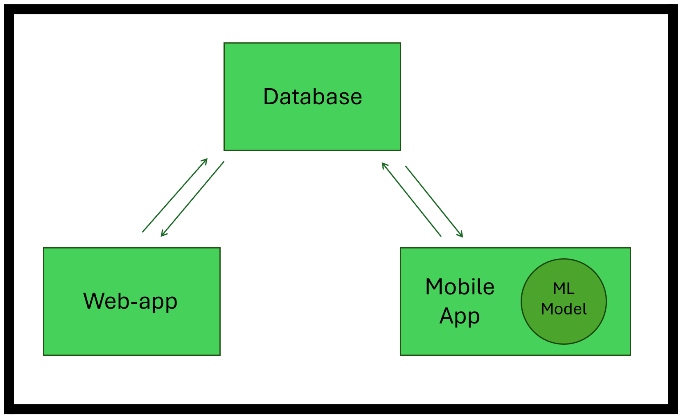


Figure 3: System Layout Description

As you can see, all parts of the project connect back to the database. Whether a user is operating on the web-app side or the mobile-app side, the database will be able to receive requests from both and update all information dynamically. For example, if changes to the database were made from the mobile application, then the database should populate the pages of the web-app automatically. The machine learning model is only accessed from the mobile application, and any time the camera is used to find items or check in/out those items.

3.4. Modes of Operations

The modes of operation for the software will be via android smartphone or computer. The camera functionality will only be able to be accessed through the app on the smartphone. It can be used in any environment where the device being used is functional. It is intended to be used in research labs or similar facilities requiring storage (Libraries, warehouses, ect.).

3.5. Users

As of now, there are only two defined users, but we have the ability to add new permission levels for new types of users later in development. Currently, we have “student” and “staff” roles. Students have the ability to check into their lab and look at all the inventory of their respective lab(s). They can check-out any items they need, see which items are available, and see how much of each the lab has for their own reference. Staff users have slightly more abilities. They are able to see the recent login times of other users and see who the most recent borrower of any item was. This will make tracking students’ usage easier for staff.

3.6. Support

Our website along with the smartphone application will have a FAQ page. This will be the main form of support. Any questions not included in the FAQ can be emailed to the team member whose subsection the question pertains to.

4. Scenarios

4.1. Check-out as Student

As a student, users should be able to login to observe all available tools in the lab, see which ones are available, locate where the item is stored, and check-out the item they need.

4.2. Check-in as Student

As a student, users should be able to login to return any items to the lab. Upon returning, the student should be able to take a picture of the item, use the machine learning model to recognize which item it is, and change the status of the item to “checked-in” so anyone else can use it

4.3. Monitor Students as Staff

As a member of the lab’s staff, users should be able see the recent login times of their students and monitor the most recent borrower of tools from the lab. If something is damaged, then staff members should be able to find the person responsible for damage.

4.4. Add/Manage Inventory as Staff

As a staff member, I should be able to add new items to the inventory of my lab, edit the attributes of any of the items, remove any unnecessary items, and deactivate items that I deem unusable.

5. Analysis

5.1. Summary of Proposed Improvements

This research lab inventory tracker will be able to provide improvement for the current traditional inventory management process. With the use of the website and application, the user will be able to have real-time updates of such inventory. This helps reduce overstocks or stockouts. The automatic inventory updates help minimize manual work that could eventually lead to human error. By having one central database server, there is a reduction in inconsistency, as staff and students are both accessing the same data.

5.2. Disadvantages and Limitations

While the proposed inventory tracker system offers various improvements, it also comes with disadvantages and limitations. There is a possibility of running into server issues that can cause data loss or website downtime. If the server becomes unavailable, the user will not be able to access the inventory data, which can cause delay in tracking or order processing. The machine learning model has the possibility of not being as accurate. As the user uploads a picture of the inventory, the machine learning model could accidentally end up confusing a piece of inventory for another, and therefore updating the wrong data. Another issue could be that item locations are not correct due to human error, such as item misplacement.

5.3. Alternatives

Some alternative solutions for this inventory management include using Excel or Google Sheets. This allows for manual data entry without having to depend on a database server. Using spreadsheets is free and is very simple to use, but not dynamic for a website management system.

5.4. Impact

A website and application inventory tracker has environmental positive impacts. There is a clear reduction in paper consumption, as there is no need for any physical records. By being able to avoid over stockings, laboratories are able to reduce overall waste. Overall, laboratories are able to reduce human error and improve productivity. In return, more reliable information can be provided to users.

Research Labs Inventory

Jonathan Moore
Evan Ross
Lizzett Tapia
Sumedha Bhattacharyaa

Functional System Requirements

REVISION – Draft
26 September 2024

# Functional System Requirements
for
Research Lab Inventory

Prepared by: Team <55>

Author Date

Approved by:

Project Leader Date

John Lusher, P.E. Date

T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
1.0	9/26/2024	Moore,Ross, Tapia, Bhattacharyaa		Initial Submission

Table of Contents

Table of Contents 3
List of Tables 4
List of Figures 5
1. Introduction 6
1.1. Purpose and Scope 6
1.2. Responsibility and Change Authority 6
2. Applicable and Reference Documents 7
2.1. Applicable Documents 7
2.2. Reference Documents 7
2.3. Order of Precedence 7
3. Requirements 8
3.1. System Definition 8
3.2. Requirements 9
3.2.1. Functional Requirements 9
3.2.2. User Interface Requirements 9
3.2.3. Performance Requirements 9, 10
4. Support Requirements 11
Appendix A Acronyms and Abbreviations 12
Appendix B Definition of Terms 13

List of Tables

Table 1. Reference Documents

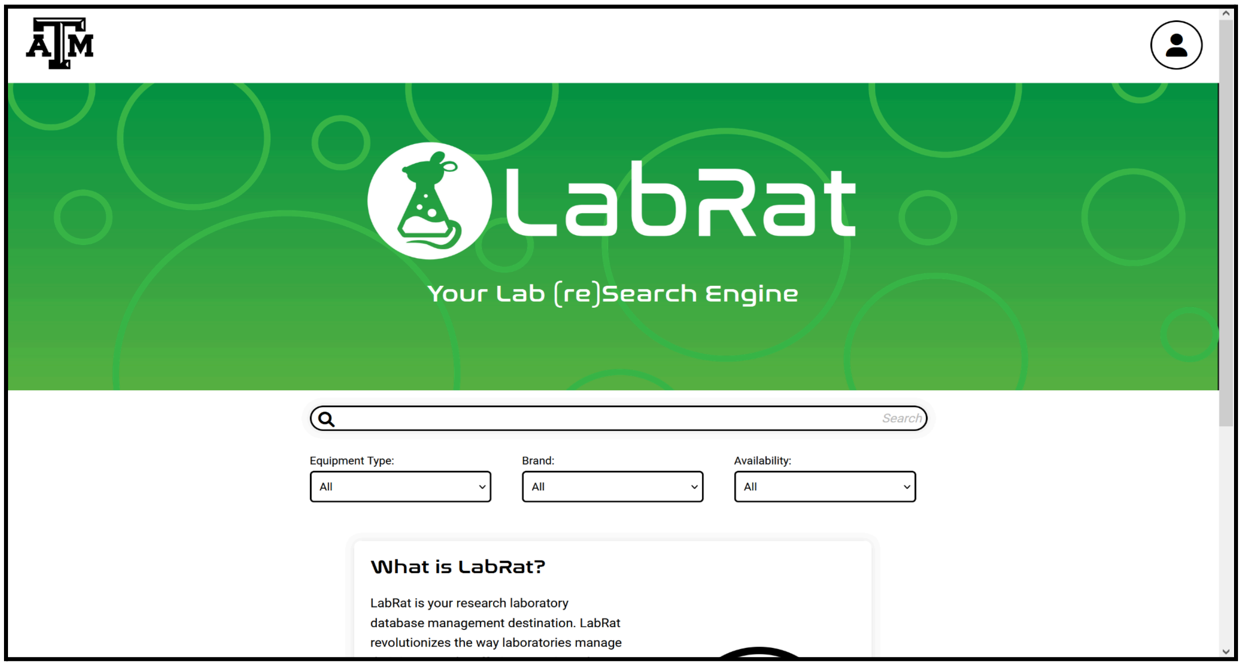
List of Figures

Figure 1. Home Page of Website **1**
Figure 2. Block Diagram of System **4**

1. Introduction

1.1. Purpose and Scope

The purpose of this Research Lab Inventory management system is to create an efficient system for tracking the inventory of research labs at Texas A&M University. This project will aim to develop a website and smartphone application where users can check in and check out lab items, which will be categorized using photo recognition machine learning. Staff members will be able to make large changes to their inventory, while students can select certain items for personal use. Ultimately, this project is designed to make the lives of researchers at this university easier with an intuitive system of organization.



LabRat is your research laboratory database management destination. LabRat revolutionizes the way laboratories manage

Figure 1. Home Page of Website

1.2. Responsibility and Change Authority

Each team member is responsible for their own subsystem meeting such requirements. Any changes to be done shall be discussed among each other and the project's sponsor, Shima Hasanpour. Since each subsystem acts separately, each team member is solely responsible for their work.

2. Applicable and Reference Documents

2.1. Applicable Documents

There are no applicable documents for this project. All instructions and requirements are given directly from our sponsor by word of mouth during our weekly meetings with her.

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.

Document Name	**Revision/Release Date**	**Publisher**
Google's Machine Learning Crash Course	2023	Google Machine Learning Education
Building a Python Image Recognition System	2024	Cloudinary
OpenCV Library	4.10.0 / 4 June 2024	OpenCV
Flutter document	2017	Flutter

Table 1. Reference Documents

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

This section defines the minimum requirements that the development item(s) must meet. The requirements and constraints that apply to performance, design, interoperability, reliability, etc., of the system, are covered.

3.1. System Definition

The Research Lab Inventory Tracker is designed to efficiently streamline the process of tracking and managing inventory at any Texas A&M University laboratory. The objective is to create a website and a smartphone application where students, professors, and administrators can log in, track inventory, and update inventory records. The system integrates a machine learning model into the smartphone application. It will allow users to scan and upload images in order to locate items, as well as help out with the check-in and check-out process. The inventory data is stored in a database server that syncs with both the website and application.

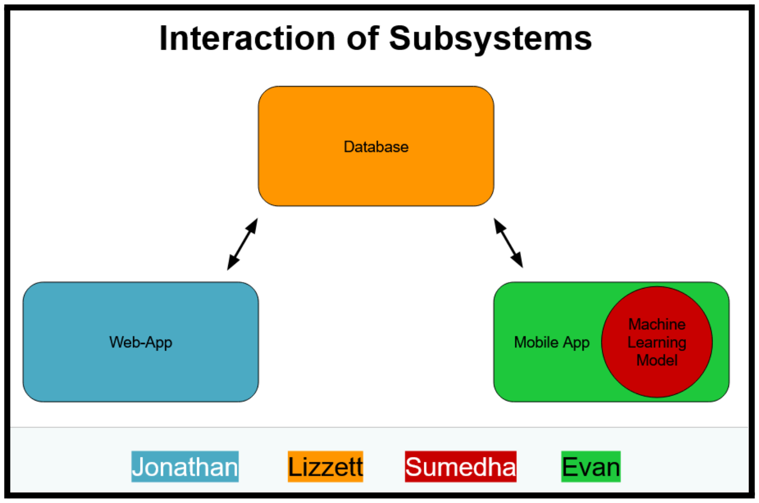


Figure 2. Block Diagram of System

The block diagram provides an overview of the subsystems in the Research Lab Inventory Tracker, as well as their interconnections. The block diagram consists of the database, the website, the mobile application, and the embedded machine learning model. The database is the main repository where all inventory data is stored. It contains information on the users, such as their usernames and email addresses. It also contains item information, such as item location, quantity, and its supplier. The database serves as the backbone of this overall system. It is what keeps track of all the data and ensures there is consistency between these interfaces. The website and mobile application interface allow users to interact with the system. Students will be able to check-out and check-in items, as well as see the location and quantity of the items. Staff members will have more of an administrative control, as they will be the ones to process any new orders. The machine learning component will enable users to scan and immediately identify, and locate, items through image recognition.

3.2. Requirements

3.2.1. Functional Requirements

3.2.1.1. Must Allow User Login

The website, as well as the mobile app, shall allow users to register for accounts and login via the account credentials (email and password). User information will be stored in the created database.

Rationale: It is important to know which user is using the software and checking out items in case of a damaged or missing item.

3.2.1.2. Inventory Management

Both students and staff shall be able to view the current inventory details, such as the item name, item description, location, and availability. When students check-in and check-out items, the inventory data should automatically update. These updates will be able to be seen on both the website and mobile application. The system should generate alerts when stock gets low for any item.

3.2.1.3. Concurrent User Load

The System should be able to support the load of at least 100 users at once without performance quality issues.

Rationale: It's likely that every lab member should use this system in the case that it's implemented, so load support is required.

3.2.2. User Interface Requirements

3.2.2.1. Data Display and Access

The user interface must present all the inventory data stored within the database in a clear and understandable manner. Items in the database will be displayed in a table with important information, such as item name, description, quantity, unit, location, and supplier.

3.2.3. Performance Requirements

3.2.3.1. Machine Learning Item Categorization

The system should accurately categorize items at least 93% of the time by the end of the training period for the machine learning model.

Rationale: This is necessary for machine learning to be used as a replacement for manually entering items to check them back in.

3.2.3.2. Item Scanning and Recognition

The system should accurately recognize items as separate entities.

Rationale: This distinction is necessary to prevent items being grouped together when there’s many items in one scan/picture.

4. Support Requirements

For use of the website application the user is required to have a computer with Google Chrome, Mozilla Firefox, or Safari as the browser. For the mobile application the user is required to have a mobile phone with the Android operating system using API 16 (Android 4.1) or above. If the user wishes to use the machine learning component of the mobile application, the user's phone must also have a functional camera.

Appendix A: Acronyms and Abbreviations

GUI	Graphical User-Interface
ORM	Object-relational Mapping
API	Application Programming Interface
IDE	Integrated Development Environment
ML	Machine Learning
API	Application Programming Interface

Appendix B: Definition of Terms

GUI	Visual interface allowing users to interact with electronic devices
ORM	A technique used in programming allowing for the connection between object-oriented programming languages to relational databases
API	A set of rules and protocols that allow software programs to communicate with each other
IDE	A software application that aids programmers in developing code
ML	An ML system allows computers to learn without directly programming anything

Research Labs Inventory

Jonathan Moore
Evan Ross
Lizzett Tapia
Sumedha Bhattacharyaa

INTERFACE CONTROL DOCUMENT

REVISION – Draft
26 September 2024

Interface Control Document
for
Research Lab Inventory

Prepared by: Team <55>

____
Author Date

Approved by:

____
Project Leader Date

____
John Lusher II, P.E. Date

____
T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
1.0	9/26/2024	Moore,Ross, Tapia, Bhattacharyaa		Initial Submission

Table of Contents

Table of Contents 3
List of Tables 4
List of Figures 5
1. Overview 6
2. References and Definitions 7
2.1. References 7
2.2. Definitions 7
3. User Interface (Web-App) 8
3.1. Frontend 8
3.1.1. Home 8
3.1.2. Check In/out 8
3.1.3. Registration 9
3.1.4. Sign In 9
3.1.5. Account 10
3.2. Backend 10
4. User Interface (Mobile App) 11
4.1. Pages 11
4.1.1. Login Screen 11
4.1.2. Registration Screen 12
4.1.3. Home Screen 12
4.1.4. Check In/Out Page 13
4.1.5. FAQ page 13
4.2. Backend 13
5. Database Server 14
5.1. Database Access 14
5.2. Data Formats 14
5.3. Security and Data Consistency 14
6. Machine Learning Model 15
6.1. Pre-Trained Model 15
6.2. Dummy Data Sets 15
6.3. Real Data Sets 15
6.4. Automated Function 15,16
7. Device Interface Software and Hardware Requirements 17
7.1. Software Requirements 17
7.2. Hardware Requirements 17

List of Tables

No table of figures entries found.

List of Figures

Figure 1. Home Page of Website
Figure 2. Register Page of Website
Figure 3. Sign-in Page of Website
Figure 4. Account Page of Website
Figure 5. Login Screen of Mobile App
Figure 6. Registration Screen of Mobile App
Figure 7. Home Screen of Mobile App

1. Overview

This Interface Control Document (ICD) will outline the communication and interaction between the research lab inventory tracking software, its connected systems, and its users. It will detail how our software is designed to facilitate these interactions to guarantee that the system operates effectively and efficiently. This document will also specify the technical and functional requirements for the final integration of the system, providing a clear guide for integration and operational compatibility. The goal is to support the deployment of our solution to Texas A&M University's issues with lab organization.

2. References and Definitions

2.1. References

Google's Machine Learning Crash Course
2023
Google Machine Learning Education

Building a Python Image Recognition System
2024
Cloudinary

OpenCV Library
4.10.0 / 4 June 2024
OpenCV

Flutter Documentation
2017
Flutter

2.2. Definitions

API	Application Programming Interface
SQL	Structured Query Language

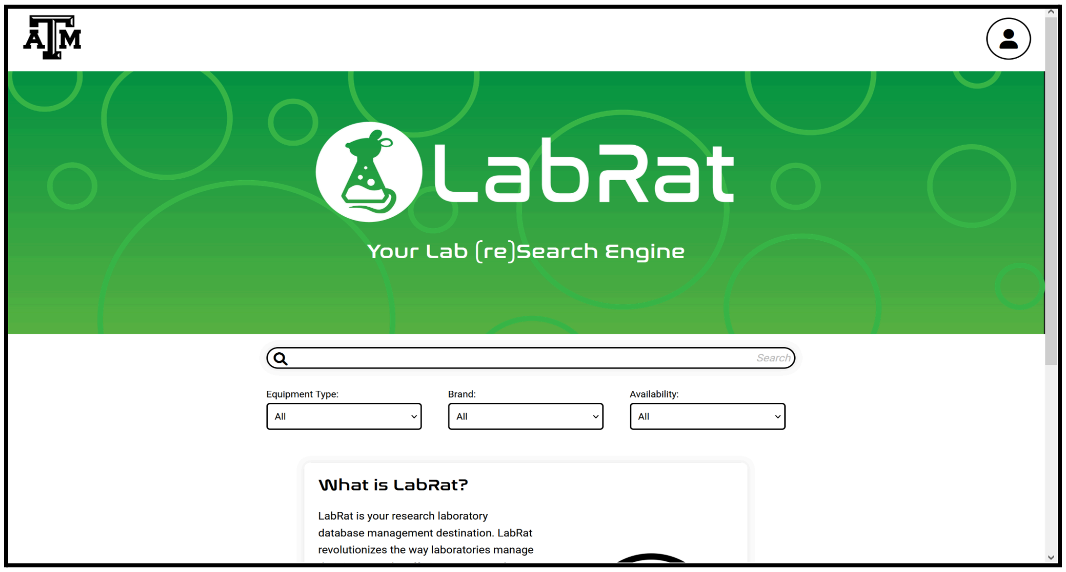
3. User Interface (Web-App)

The UI of the web application is designed to streamline the inventory management process in Texas A&M University research labs by offering an intuitive and efficient user experience. The layout will help enable users to quickly search for equipment, filter results based on specific criteria, and access relevant information about lab inventory. The interface is built for ease of navigation, allowing users to perform actions such as signing in, registering, and managing their accounts with minimal effort. Each element of the UI is aimed at simplifying inventory tracking and enhancing the overall efficiency of lab operations. The design will be further elaborated upon in dedicated sections for each functionality.

3.1. Frontend

3.1.1. Home

The home page is designed as the home for users. Most of the functionality will be built into this page, including the search functionality. When users sign in, their initial replaces the person icon on the top right corner to signify a session is active for that user. After signing in, the page will populate with a table of the current inventory of the database so the user can check-out/check-in items. A cart feature is also planned. In this case, a cart appears within the dashboard bar above when items have been added to the user's desired list of items.



LabRat is your research laboratory database management destination. LabRat revolutionizes the way laboratories manage

Figure 1. Home Page of Website

3.1.2. Check in/out

The checkout and checkin pages will be added to finalize any changes to the user's desired items. On this page, users will have to check a box agreeing to our terms of use, so that any items borrowed from the lab will be protected under an agreement.

3.1.3. Registration

The registration page is already fully completed. It provides the user with a form asking for name, email, password, and a password confirmation. All this information is stored in the “users” table of our SQL database. All passwords are encrypted so that no plaintext passwords are stored to protect our users.

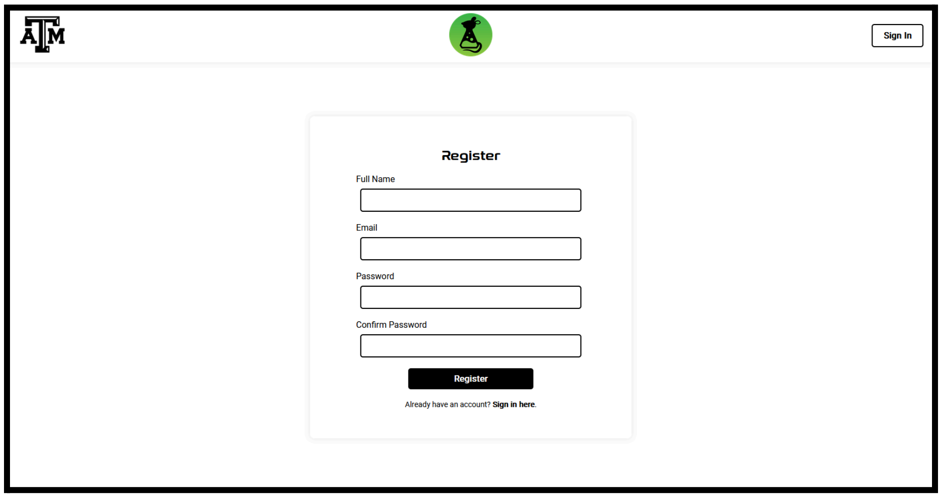


Figure 2. Register Page of Website

3.1.4. Sign-in

The sign-in page is already mostly completed. When a user types their email and password, a session begins under their id and is stored in their browser’s cookies. Upon return to the website, their sign-in will persist and they can return to their previous session. This can be ended by signing out from the home page. A forgotten password feature is planned. In order to accomplish this, we will have to use an email API to send temporary passwords to our users so they can reset their passwords. This is for later in the project.

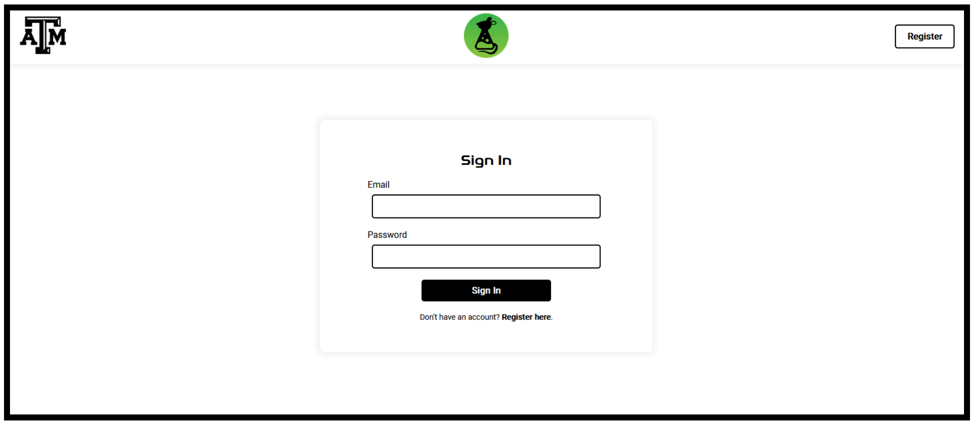
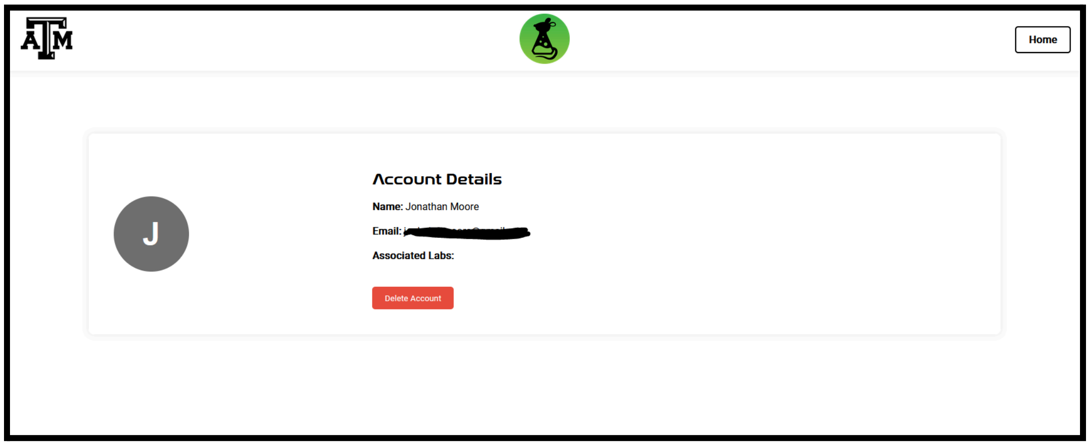


Figure 3. Sign-in Page of Website

3.1.5. Account

The account page gives users the ability to change their information. They have the option to change their name, email and password, but to change their affiliated TAMU labs, it has to be approved by a professor or administrator from that lab. This will also be accomplished through an email API.



Account Details
Name: Jonathan Moore
Email: [redacted]
Associated Labs:

Figure 4. Account Page of Website

3.2. Backend

In order to accomplish the functionality of this website without the use of a framework (such as React, Vue, Angular), we are choosing to use Javascript and Node.js to host and operate the website. This choice was deliberately made by the team to make development move as quick as possible while still providing great results.

4. User Interface (Mobile App)

The purpose of the mobile app is to give users the ability to easily and intuitively track, manage, and make changes to the inventory of a Texas A&M University research lab from their Android smartphone. The mobile app is designed to have similar functionality to the website, as well as similar pages and navigation between them. This is so that someone familiar with the website is able to easily navigate the mobile app counterpart, and vice versa. One unique aspect of the Mobile app is that it will allow users to use their camera to scan and identify items in the lab. Once an item is scanned, the user will be sent to the check in/out page for that item. This will be achieved through a machine learning model that is trained to recognize the items in the lab.

4.1. Pages

4.1.1. Login Screen

This is the first screen that users see when they open the mobile app. They will be given the option to enter their email and password to sign in if they already have an account. The password will be obscured for security. If they do not have an account yet there will be a button which brings users to the registration page.

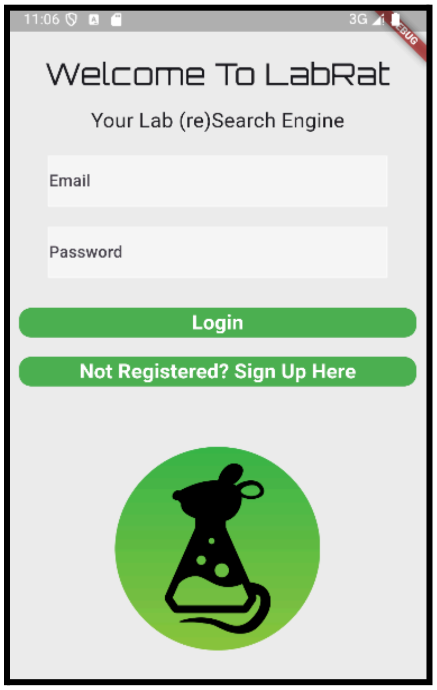


Figure 5. Login Screen of Mobile App

4.1.2. Registration Screen

In this screen users will be able to register for an account. They will be prompted to enter their email and password, as well as password confirmation. User data will be stored in the SQL database “Users” table.

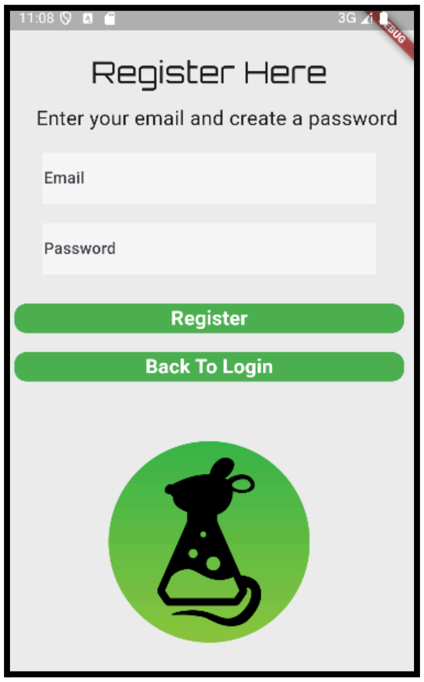


Figure 6. Registration Screen of Mobile App

4.1.3. Home Screen

The home screen will be the main page in the mobile application. It is where users will be able to search for items either through a search bar, or through the drop down menus on the page. It will allow users to see what Items are available to be checked-out/checked-in. When a user selects an Item from the home screen they will be sent to the check out/in page. This is also the page where users will be able to open their camera to scan an item. When a user scans an item they will be sent to the corresponding check in/out page.

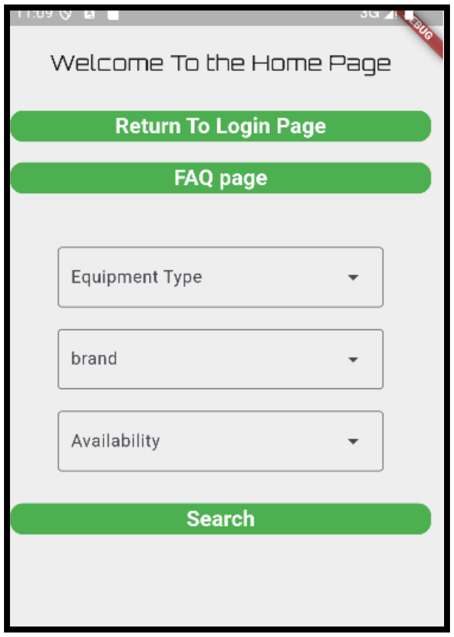


Figure 7. Home Screen of Mobile App

4.1.4. Check Out/In page

This is the page that will allow users to check out/in items. After a user selects an item from the home page they will be sent to this page, which will provide a brief description of the item as well as the ability to agree to the terms of use and check out/in the item.

4.1.5. FAQ page

This page will provide answers to frequently asked questions, such as how to create an account and check in/out items.

4.2. Backend

The Mobile application UI is created using Android Studio along with the plugin Flutter. Flutter uses the dart coding language, which is specifically designed for mobile app development. A Flutter plugin called SQFLITE is used to connect the application to a temporary SQLite database.

5. Database Server

The database is the central point for both data storage and data retrieval. It helps all the subsystems work together and interact with user data and inventory information.

5.1. Database Access

Access to the database will depend on the user's role, whether they are a student or staff. Students will have a more restricted access, as they can only modify check-in and check-out records. On the other hand, staff members will be able to add or remove items.

5.2. Data Formats

Each data entity in the database will have a defined structure. For example, the *Users* data table will have:

- UserID
- FirstName
- LastName
- Email
- Role
- Username
- Hashed-Password

The *Items* data table will have:

- ItemID
- Name
- Description
- CategoryID
- Quantity
- Unit
- Location
- Supplier

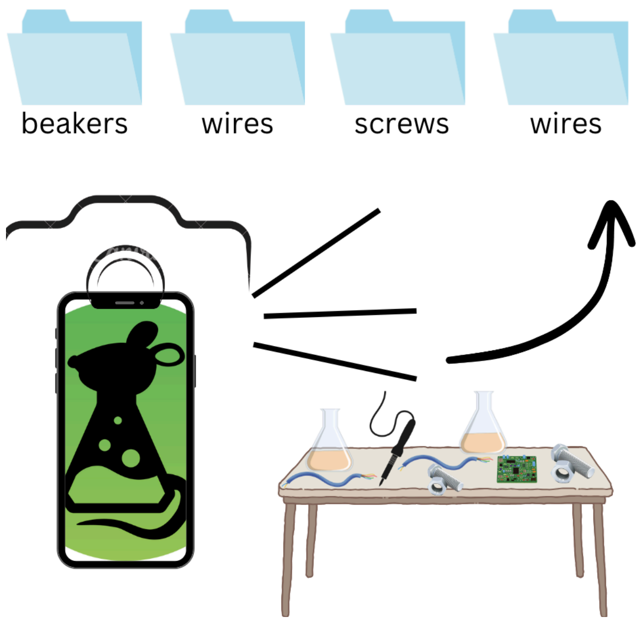
Having a standard data format helps keep consistent communication and proper data retrieval.

5.3. Security and Data Consistency

Data that gets exchanged between the website/mobile app and the database will be encrypted. This helps make sure that confidential data, such as login credentials, cannot be intercepted. Only authorized users will have access to the data. Audit logs will be beneficial, as it allows for traceability in the case that there are any outages or data breaches.

6. Machine Learning Model

The purpose of the ML Model is to allow users to use the app component on their mobile devices to scan all their items, and have them automatically checked-in. This includes identifying how many items they're checking in, as well as the category of each. The overall goal is to increase the efficiency of labs at TAMU, and this is a component of that increase. This also aims to enhance the experience of all app users.



6.1. Pre-trained model

A pretrained model optimized for image classification purposes. The chosen model is called ResNet50 and was made from TensorFlow - an open source machine learning platform.

6.2. Dummy Data Sets

The dummy data sets are the initial data that will be used to train the model specifically with common items in a lab. There's different resolutions, sizes, and formats of images in every sub-category.

6.3. Real Data Sets

Eventually, pictures of actual items from labs will be acquired in order to further train it with the exact items it will be witnessing through user application.

6.4. Automated Function

An automated function will be implemented in the Python code of the ML Model, to enhance the speed of training. This function will allow for the model to run through all the images in a folder, so that the path to each image doesn't have to be manually entered every time.

7. Device Interface Software and Hardware Requirements

7.1. Software Requirements

7.1.1 Software Requirements of Web Application

The website will be compatible with the latest versions of Google Chrome, Mozilla Firefox, and Safari.

7.1.2 Software Requirements of Mobile Application

The device used to run the mobile application must use the Android operating system. As stated in the Flutter Documentation, Android API 16 (Android 4.1) or above is required to run the mobile application.

7.2. Hardware Requirements

7.2.1 Hardware Requirements of Application Server

The server must be equipped with storage to accommodate for large amounts of data for inventory.

7.2.2 Hardware Requirements of Mobile Application

Though not required, the mobile device must have a functional camera if the user wishes to utilize the machine learning aspect of the mobile application.

Execution Plan

Key
🟦 Web App
🟧 Data Server
🟩 Mobile App
🟥 ML Model
🟩 Completed
🟨 In Progress
▨ Not Started
🟥 Behind Schedule

	8/20/24	8/27/24	9/3/24	09/10/24	09/17/24	09/24/24	10/01/24	10/08/24	10/15/24	10/22/24	10/29/24	11/5/24
Basic Page Layouts	🟨	🟩	🟩	🟩	🟩	🟩						
Backend for each Page		🟨	🟨	🟩	🟩	🟩						
Search Engine Functionality							▨	▨				
Check-in and check-out for items								▨	▨	▨		
Web-app Validation											▨	▨
SQL Database Created	🟨	🟨	🟨	🟨	🟩	🟩						
Getting a working server						▨	▨					
Populate database							▨	▨				
Relationships Between Data								▨	▨			
Database Server Validation										▨	▨	▨
Create App and Functional Simulation		🟨	🟩									
Create UI for each page			🟨	🟩	🟩	🟩						
Add functionality using SQLite database					🟨	🟩						
Search Database and Check in + out Functionality						▨	▨	▨	▨			
Mobile App Validation										▨	▨	▨
Watch Youtube Videos on ML/Nueral Networks	🟩											
Choose ML Model and Finalize Code		🟩	🟩	🟩	🟩							
Finalize Dummy Data Sets and Create an Automation Function						🟨	🟨					
Create an Automation Function					▨	▨	▨					
ML Model Validation								▨	▨	▨	▨	▨
ConOps Project Report	DUE SEP 15											
FSR, ICD, Milestones, Plan	DUE SEPT 26											
Project Introduction Presentations	PRESENT ON SEPT 24											
Project Update Presentations	PRESENT ON OCT 14 - 23											
Final Presentations	PRESENT ON NOV 11-20											
Final Report	DUE DEC 5											

Validation Plan

Key
🟦 Web App
🟧 Data Server
🟩 Mobile App
🟥 ML Model

Test Name	Success Criteria
Input Testing (Happy path)	Accept changes in db for valid input fields for registration, sign-in, etc.
Input Testing (Unhappy path)	Do not make changes in db for invalid input fields for registration, sign-in, etc.
Page route testing	Ensure that all page routing acts as expected (kicking unauthorized users, sign-out upon deletion of account, etc.)
Cart functionality (Happy path)	Cart for check-in/check-out should allow users to select as many ites as they want and perform a collective check-out all at once.
Cart functionality (Unhappy path)	Cart shouldn't break when stress tested. Should adapt to the limits of what is possible from user.
Permission level testing	Users of different permission levels are restricted to only their given abilities.
API functionality (email, ORM)	Validate API interaction and test inputs
Functional Testing	All create, read, update, and delete operations work as expected.
Data Integrity Testing	No data corruption or any sort of inconsistent data entry is allowed.
Performance Testing	The database can handle large datasets without significant slowdowns.
Backup and Recovery Testing	After any failure encounters, data can be restored from backup.
Application UI functionality	App User interface acts as expected, It can open, navigate pages, close, has no bugs
Database Test (Users)	Application can successfully add and remove users from database
Database Test (Components)	Application can successfully add and remove components from database
User Permissions	Ensure that Students and Staff members have different permissions and abiities
Image Recognition Accuracy	The model correctly identifies/categorizes items with at least 95% accuracy.
Response Time (Latency)	The time in between inputting an image and then returning the result shouldn't exceed 3 seconds.
False Positive Rate	The model should have a false positive rate of less than 4% when identifying items.
Usability	It should be easy for users to capture a picture and scan their items in less than ~1 min,
Handling of Multiple items in One Image	The model should be able to identify multiple items in a single image with at least 90% accuracy for each object.

Validation Plan Continued

Key
Web App
Data Server
Mobile App
ML Model

Test Name	Methodology	Status	Responsible Engineer (s)
Input Testing (Happy path)	Using Playwright to test forms. All inputs are hand picked to stress test bounds of valid inputs.	Untested	Jonathan Moore
Input Testing (Unhappy path)	Using Playwright to test forms. All inputs are hand picked to stress test using invalid inputs.	Untested	Jonathan Moore
Page route testing	Using Playwright for UI regression testing.	Untested	Jonathan Moore
Cart functionality (Happy path)	Playwright for regression testing for the display of the cart.	Unitested	Jonathan Moore
Cart functionality (Unhappy path)	Playwright for regression testing for the display of the cart, ensuring it doesn't break.	Untested	Jonathan Moore
Permission level testing	Manual testing. This requires changing the permission levels of a current user multiple times. This is not possible using Playwright, so I have to alter manually.	Untested	Jonathan Moore
API functionality (email, ORM)	Using Postman, I will test all API inputs/outputs	Untested	Jonathan Moore
Functional Testing	Manually test each operation, such as creating a new inventory item and checking if it appears in the table.	Untested	Lizzett Tapia
Data Integrity Testing	Manually test by inserting different records with any sort of mising or invalid data to see if it gets rejected.	Untested	Lizzett Tapia
Performance Testing	Read data, using the select statement, on large datasets and measure the response time.	Untested	Lizzett Tapia
Backup and Recovery Testing	Set up an automated backup system using MySQL.	Untested	Lizzett Tapia
Application UI functionality	Test application within the Android Studio simulated phone	Untested	Evan Ross
Database Test (Users)	Test by adding and removing users through the mobile app, test invalid users as well (such as invalid email) to ensure user has valid credentials.	Untested	Evan Ross
Database Test (Components)	Test by adding cononents to a the database through the mobile app, test both large and small amounts at once.	Untested	Evan Ross
User Permissions	Test by creating users with different attributes and test manually to ensure that they have the correct permission.	Untested	Evan Ross
Image Recognition Accuracy	Compile dummy datatset with unknown data set (distractor items) and run the test data sets through. Record the percentage of correct identifications.	Untested	Sumedha Bhattacharyaa
Response Time (Latency)	Record time in between uploading an image and getting a result. Test across different internets, and image sizes.	Untested	Sumedha Bhattacharyaa
False Positive Rate	Create dataset with items that don't belong in the lab. Test the system's ability to avoid classifying these items. The calculate the percentage of how many times it incorrectly categorized something.	Untested	Sumedha Bhattacharyaa
Usability	Test with a sample group of 'users', and then measure the average time taken to capture an image and receive a result.	Untested	Sumedha Bhattacharyaa
Handling of Multiple items in One Image	Make dataset with images that contain more than one lab item. The test the model's ability to identify all items in the image and record the accuracy.	Untested	Sumedha Bhattacharyaa