**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 |
| **2.0** | 12/5/2024 | Moore |  | Final Edit |

**Table of Contents**

[**Table of Contents**](#_heading=h.30j0zll) **3**

[**List of Tables**](#_heading=h.1fob9te) **4**

[**List of Figures**](#_heading=h.3znysh7) **5**

[**1.**](#_heading=h.4d34og8) **Overview 6**

[**2.**](#_heading=h.2s8eyo1) **References and Definitions 7**

[2.1.](#_heading=h.17dp8vu) References 7

[2.2.](#_heading=h.3rdcrjn) Definitions 7

[**3.**](#_heading=h.lnxbz9) **User Interface (Web-App) 8**

[3.1.](#_heading=h.35nkun2) Frontend 8

[3.1.1.](#_heading=h.tyjcwt) Home 8

[3.1.2.](#_heading=h.1ksv4uv) Check In/out 8

[3.1.3.](#_heading=h.44sinio) Registration 9

[3.1.4.](#_heading=h.2jxsxqh) Sign In 9

[3.1.5.](#_heading=h.z337ya) Account 10

[3.2.](#_heading=h.z337ya) Backend 10

**4**[**.**](#_heading=h.lnxbz9) **User Interface (Mobile App) 11**

4[.1.](#_heading=h.35nkun2) Pages 11

4[.1.1.](#_heading=h.tyjcwt) Login Screen 11

4[.1.2.](#_heading=h.1ksv4uv) Registration Screen 12

4[.1.3.](#_heading=h.44sinio) Home Screen 12

4[.1.4.](#_heading=h.2jxsxqh) Check In/Out Page 13

4[.1.5.](#_heading=h.z337ya) FAQ page 13

4[.2.](#_heading=h.z337ya) Backend 13

[**5.**](#_heading=h.3dy6vkm) **Database Server 14**

[5.1.](#_heading=h.4i7ojhp) Database Access 14

[5.2.](#_heading=h.2xcytpi) Data Formats 14

[5.3.](#_heading=h.1ci93xb) Security and Data Consistency 14

**6**[**.**](#_heading=h.3dy6vkm) **Machine Learning Model 15**

6[.1.](#_heading=h.4i7ojhp) Pre-Trained Model 15

6[.2.](#_heading=h.2xcytpi) Dummy Data Sets15

6[.3.](#_heading=h.1ci93xb) Real Data Sets 15

6[.4.](#_heading=h.1ci93xb) Automated Function 15,16

**7**[**.**](#_heading=h.3dy6vkm) **Device Interface Software and Hardware Requirements 17**

7.1.Software Requirements 17

7.2[.](#_heading=h.1ci93xb) 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

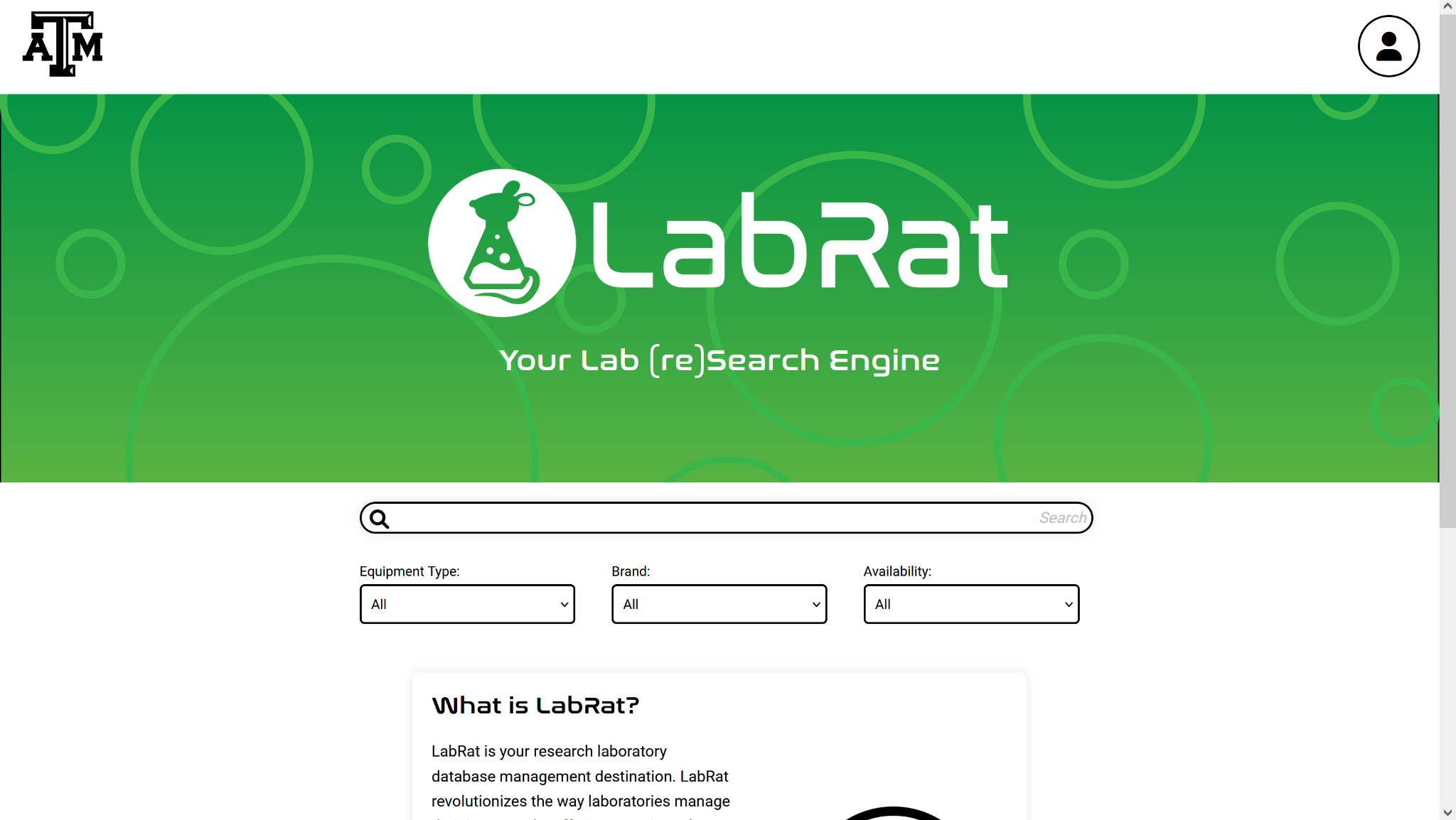
# 3. User Interface (Web-App)

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.



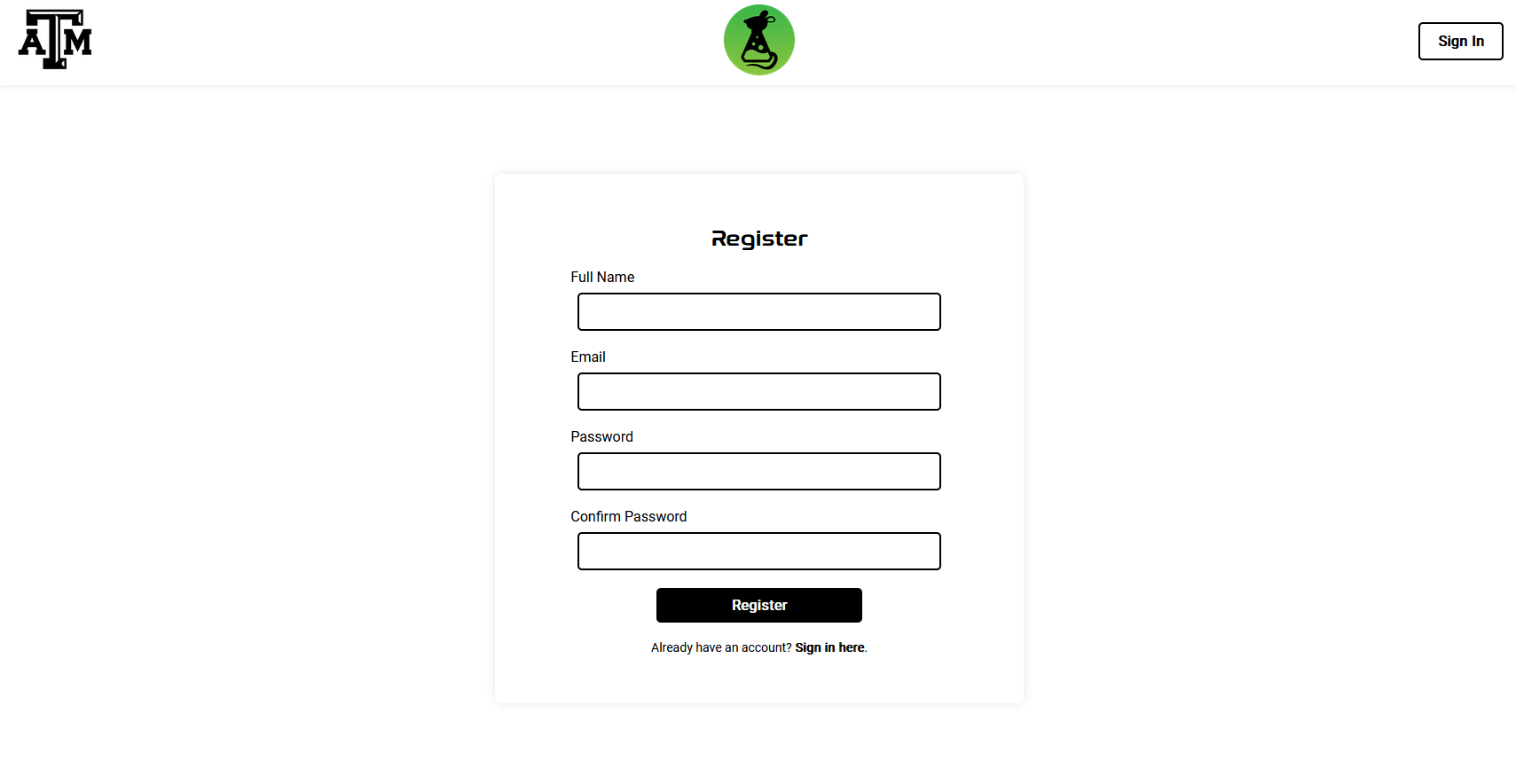
**Figure 1. Home Page of Website**

### 3.1.2. Check in/out

The checkout page 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

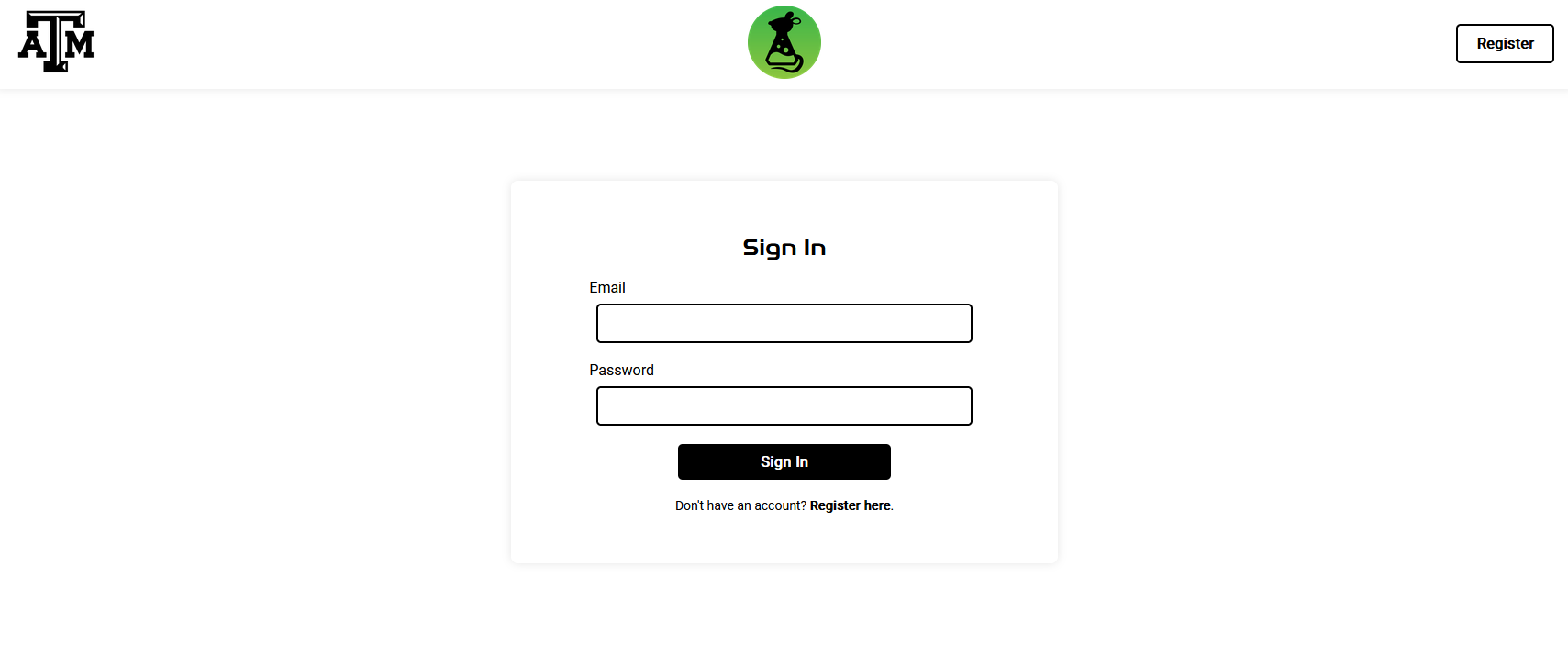
The registration page is already fully completed. It provides the user with a form asking for name, email, password, and a password confirmation. All this information is stored in the “users” table of our SQL database. All passwords are encrypted so that no plaintext passwords are stored to protect our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



**Figure 2. Register Page of Website**

### 3.1.4. Sign-in

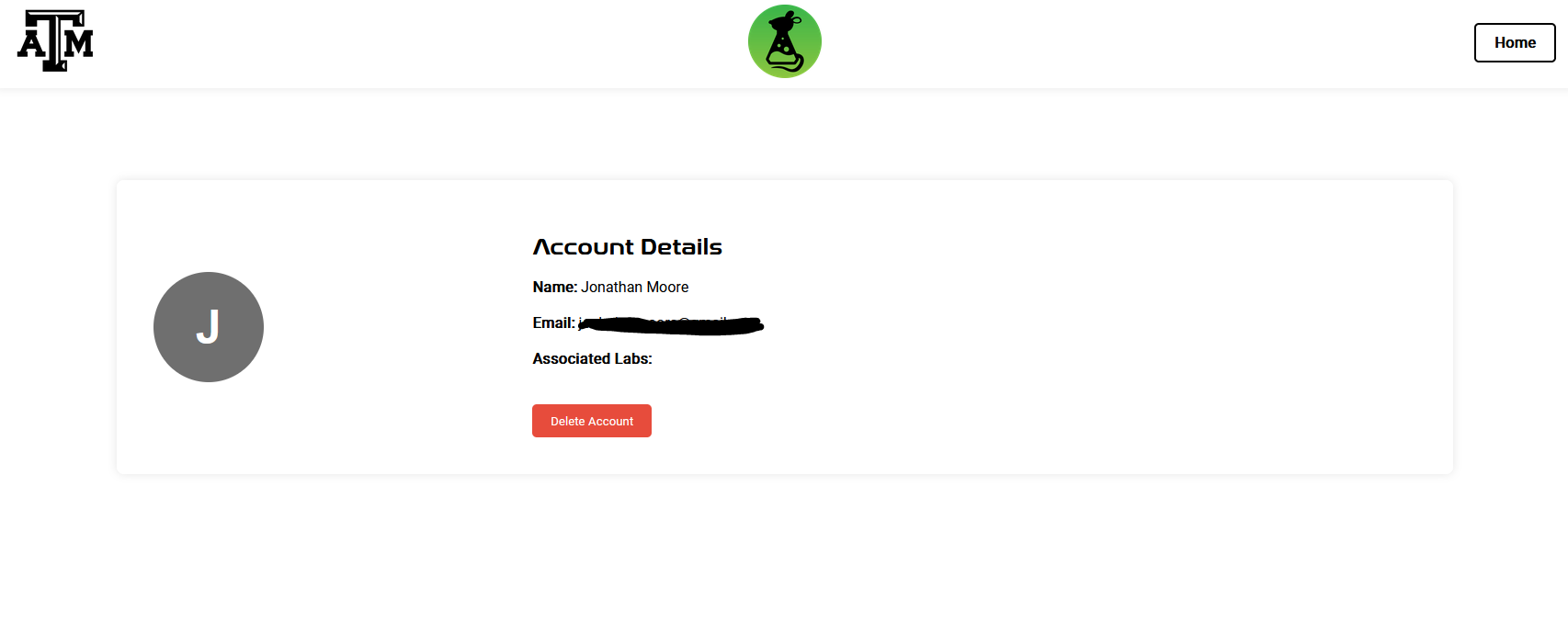
The sign-in page is already mostly completed. When a user types their email and password, a session begins under their id and is stored in their browser’s cookies. Upon return to the website, their sign-in will persist and they can return to their previous session. This can be ended by signing out from the home page.



**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.



**Figure 4. Account Page of Website**

## 3.2. Backend

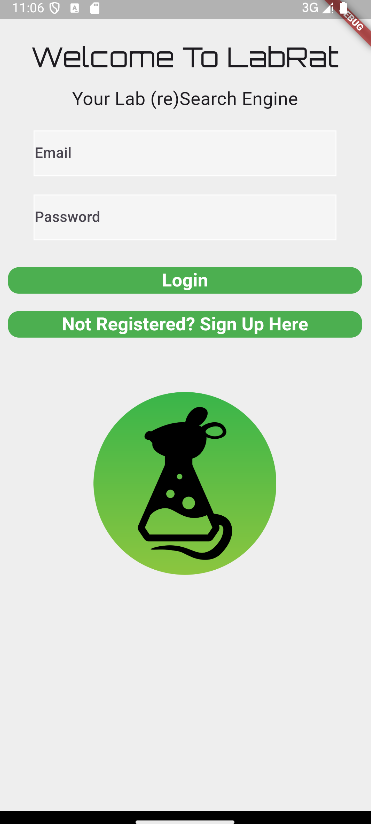
In order to accomplish the functionality of this website without the use of a framework (such as React, Vue, Angular), we are choosing to use Javascript and Node.js to host and operate the website. This choice was deliberately made by the team to make development move as quick as possible while still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

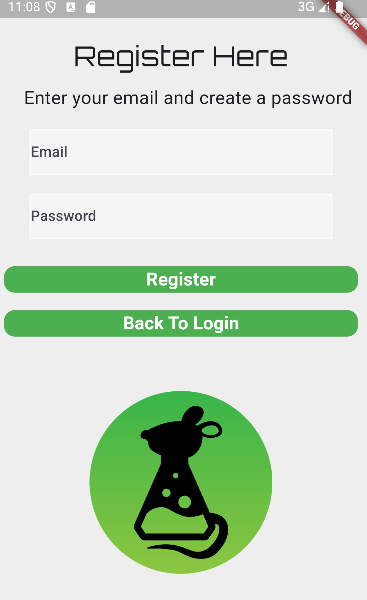
**4.1.1. Login Screen**

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**

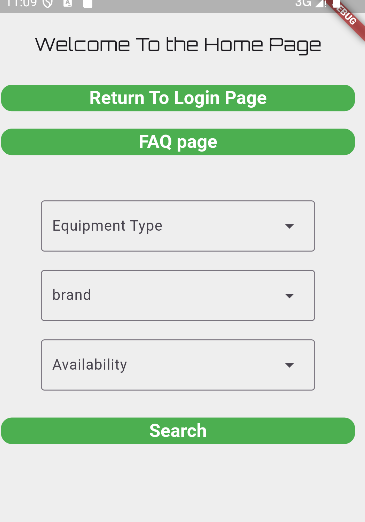
In this screen users will be able to register for an account. They will be prompted to enter their email and password, as well as password confirmation. User data will be stored in the SQL database “Users” table.



**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 allo w 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

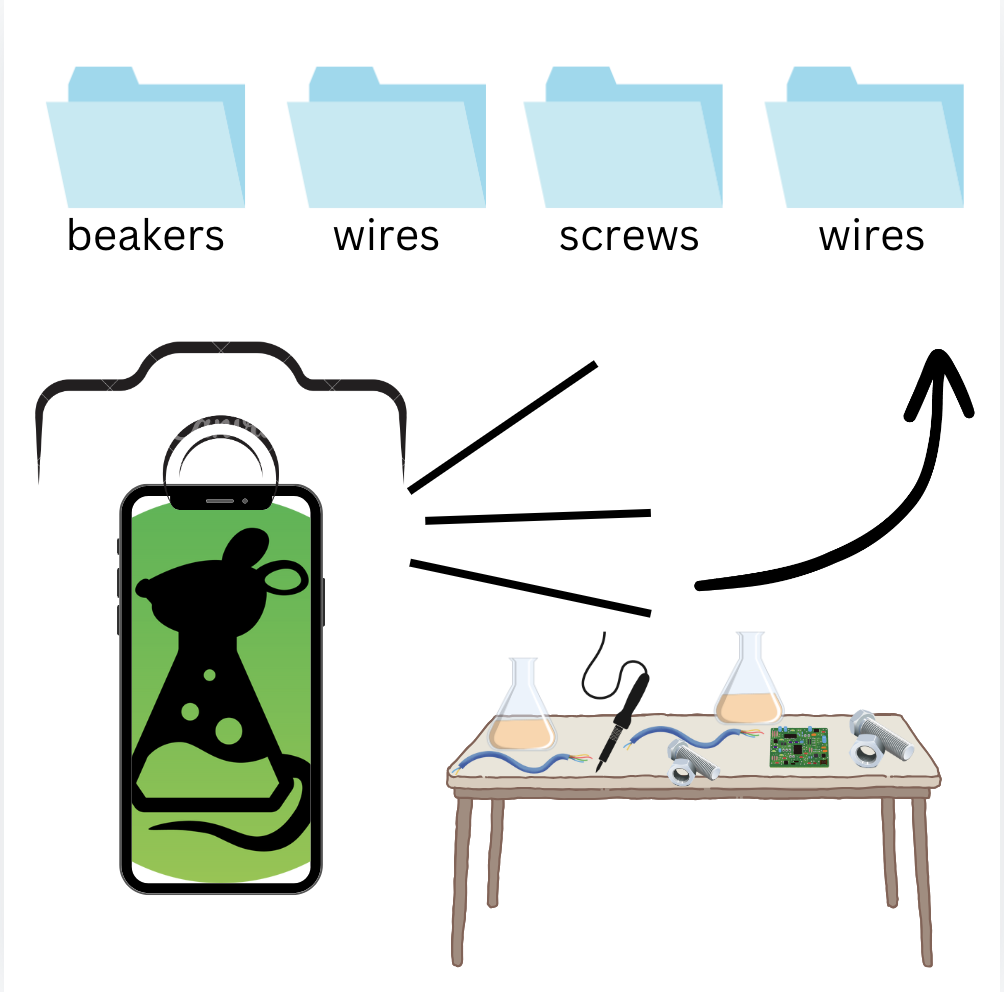
Having a standard data format helps keep consistent communication and proper data retrieval.

## 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.