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

**Table of Contents**

[**Table of Contents**](#_heading=h.1fob9te) **3**

[**List of Tables**](#_heading=h.3znysh7) **4**

[**List of Figures**](#_heading=h.2et92p0) **5**

[**1.**](#_heading=h.tyjcwt) **Introduction 6**

[1.1.](#_heading=h.3dy6vkm) Purpose and Scope 6

[1.2.](#_heading=h.4d34og8) Responsibility and Change Authority 6

[**2.**](#_heading=h.2s8eyo1) **Applicable and Reference Documents 7**

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

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

[2.3.](#_heading=h.26in1rg) Order of Precedence 7

[**3.**](#_heading=h.35nkun2) **Requirements 8**

[3.1.](#_heading=h.1ksv4uv) System Definition 8

[3.2.](#_heading=h.2jxsxqh) Requirements 9

[3.2.1.](#_heading=h.z337ya) Functional Requirements 9

[3.2.2.](#_heading=h.3j2qqm3) User Interface Requirements 9

[3.2.3.](#_heading=h.1y810tw) Performance Requirements 9, 10

[**4.**](#_heading=h.2bn6wsx) **Support Requirements 11**

[**Appendix A Acronyms and Abbreviations 1**](#_heading=h.qsh70q)**2**

[**Appendix B Definition of Terms 1**](#_heading=h.1pxezwc)**3**

**List of Tables**

**Table 1. Reference Documents**

**List of Figures**

[**Figure 1.**](#_heading=h.1t3h5sf) **Home Page of Website** [**1**](#_heading=h.1t3h5sf)

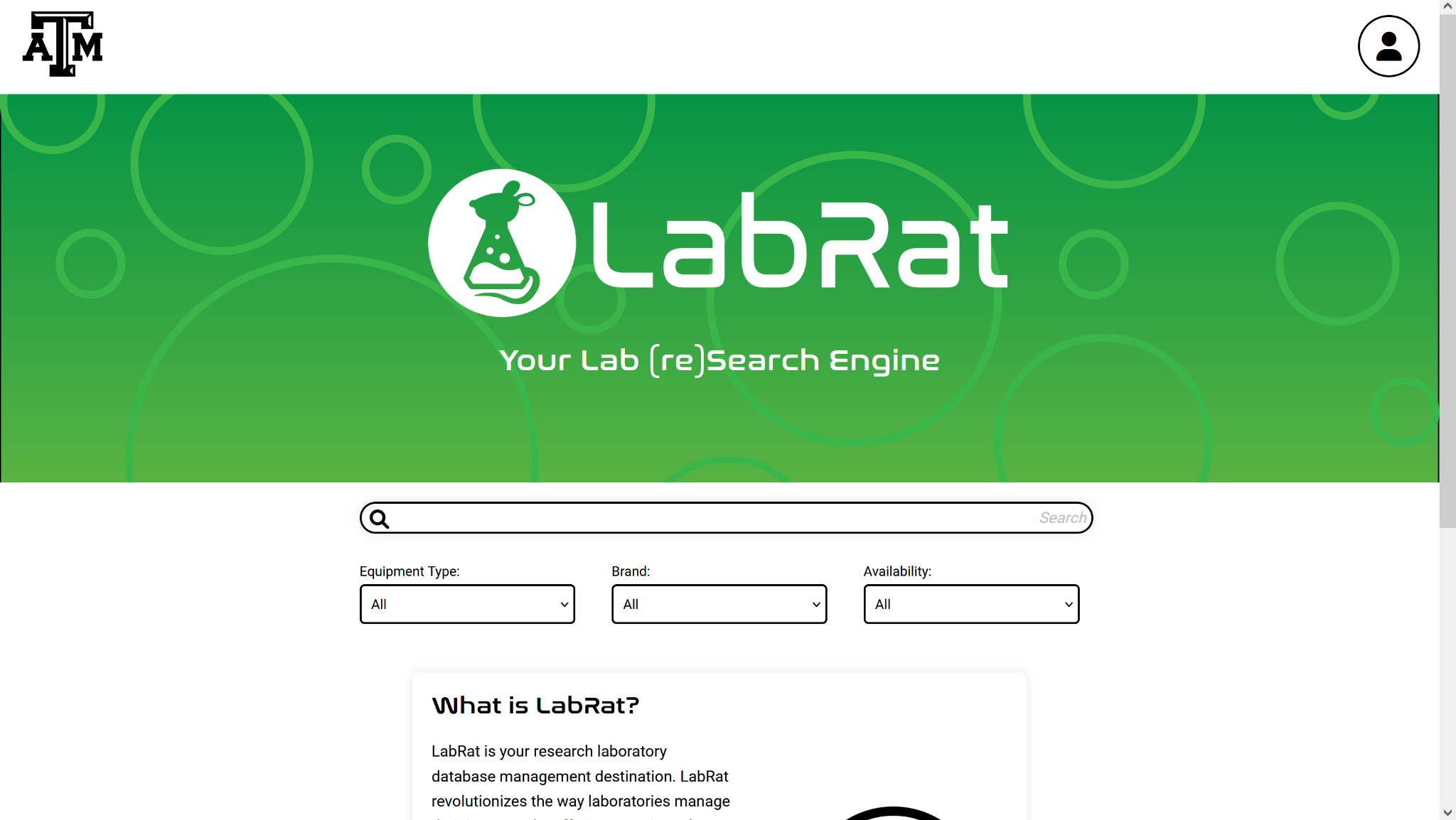
[**Figure 2. Block Diagram of System 4**](#_heading=h.44sinio)

# 

# 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.



**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.

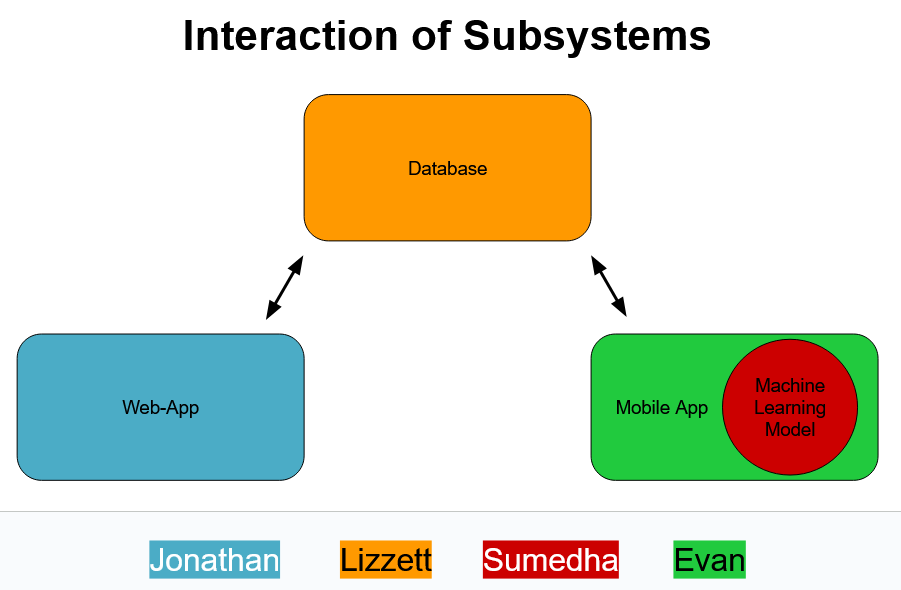
All specifications, standards, exhibits, drawings or other documents that are invoked as “applicable” in this specification are incorporated as cited. All documents that are referred to within an applicable report are considered to be for guidance and information only, except ICDs that have their relevant 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](#_heading=h.1y810tw)

#### 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