Software Requirements Specification and Design document

for

Lab Map

Version 1.2 approved

Prepared by Raymond Quach, Noah Thornton, Saul Romero, and

Bryan Smith

Please Hire Me

May 19, 2023

Revision History

Name	Date	Reason For Changes	Version
Raymond Q., Noah T., Saul R., Bryan S.	2/19/2023	Initial Proposal	1.0
Raymond Q., Noah T., Saul R., Bryan S.	04/02/2023	Midproject Modifications for SRS	1.1
Raymond Q., Noah T., Saul R., Bryan S.	5/19/2023	Final SRS Modifications	1.2

1. Introduction

GitHub link: https://github.com/nthornton00/CS370Project/tree/CS370Project_ver2

1.1 Purpose

The program's purpose is to create an organizational tool to help manage and maintain a lab that is testing hardware. The program will help visualize which employee, batch of devices from a lot tray(s), testing hardware (motherboards), what project it is currently working on, and testing peripherals are stationed at which testing setup/tester PC in the lab. As per client, every batch of devices, testing hardware, tester PC, and peripheral are all ID'd. A separate menu will pop up when you select a tester and request to see more details via a button to see extra details about the tester's setup. Details can include the staff member appointed to the tester, the project that the tester is being used for, and how many devices are in the batch at the tester. Data of every aspect of the lab covered in this software will be saved to a local SQL database.

1.2 About our project team

Please Hire Me is a small group of four senior computer scientists at California State University San Marcos (CSUSM) looking to provide a user-friendly solution to facilitate organized lab sessions. Our Lab Map product will be the first project developed by Please Hire Me. Each senior computer scientist can be contacted through the following email addresses:

Noah Thornton – thorn073@csusm.edu
Bryan Smith – smith829@csusm.edu
Saul Romero – romer381@csusm.edu
Raymond Quach – quach028@csusm.edu

2. Statement of Work

2.1 Communication

Communication between Please Hire Me's software team will be conducted mainly through our team Discord server. This Discord server will be used to discuss solutions to potential issues that arise while working on the Lab Map project, project updates, deadlines, and potential improvements that can be implemented. Bi- weekly in-person meetings with all team members will be conducted to address developments within the project and to review deadlines to ensure what is to be expected. Any concerns from CSUSM and stakeholders shall be addressed through in-person meetings or email.

2.2 Dependencies and Constraints

This project will be dependent on a local SQL database to store and retrieve lab information. This project will also be coded in Java, allowing it to run on most systems.

The primary constraint for this project is time, as it is due by Week 15 of the 2023 CSUSM Spring semester.

2.3 Design, Development, and Implementation Methods

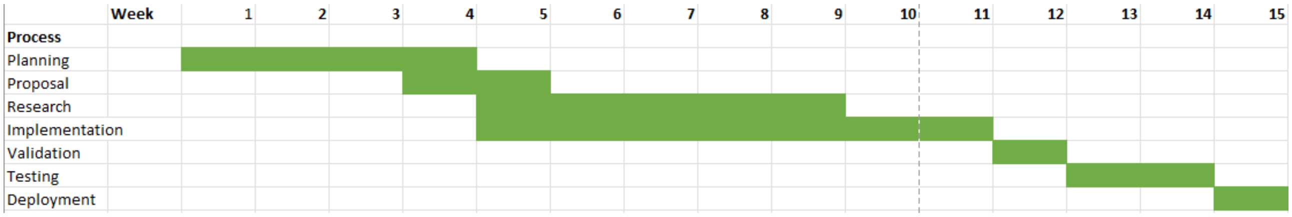
For this project, GitHub will be utilized to allow team members to push and pull project code as necessary while not relying on a specific integrated development environment (IDE). GitHub is a vision control system that allows code to be managed and recovered when needed. An incremental development model will be followed to complete this project. An outline with all desired specifications will be produced and agreed upon with the customer. Once agreed upon, the project will begin development. An initial version will be produced and presented to the acquirer for feedback. Subsequent versions of the project will be developed based on the feedback received from each version. This model accommodates the customer the best while allowing Please Hire Me to quickly deliver a useful, working product. Extensive testing will be done on all functions and databases implemented and will be documented. CSUSM will be the intellectual property owner of the design documentation.

2.4 Change Management

Any changes in scope, requirements, technology, and the contract should be submitted to Please Hire Me, where each submitted change will be documented and reviewed to determine the costs and schedule accommodation necessary. CSUSM will be responsible for the cost of any information not originally provided that creates significant costs or schedule changes. Conversely, in the event Please Hire Me provides information not originally provided that creates significant costs or schedule changes, Please Hire Me will be fully responsible for all costs associated with such changes.

3. Timeline

.



	Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Process																
Planning		🟩	🟩	🟩	🟩											
Proposal					🟩	🟩										
Research						🟩	🟩	🟩	🟩	🟩						
Implementation						🟩	🟩	🟩	🟩	🟩	🟩	🟩				
Validation													🟩			
Testing														🟩	🟩	
Deployment																🟩

4. Overall Description

**Note*: Due to time constraints and the overall priority given to features, some features were not able to be implemented in the final version of this project. To specify which features have not been implemented, a note just like this one will be included underneath each section if there is a feature described in that section that is not in the final version of the project.

The program (as demonstrated in the project showcase) obtains information from already made databases and represents the environment of a hypothetical organization. All of the remaining functionalities work as intended and represent the entire set of tasks that a non-privileged user can perform.

4.1 Product Perspective

This will be a self-contained product. Its primary function will be to serve as an organization tool for hardware labs where the use of each piece of hardware equipment will be documented (as described in section 3.1. The program will allow the user (lab administrator) to create different lab regions each of which will have different lab rooms.* When selecting a lab, a map of the lab will be displayed which will show all the equipment/testers as well as information about their details. Additional information and functions will be given for testers where individual users (workers) will have the ability to see maintenance details as well as submit maintenance requests. Users will also be able to claim a tester for their use. The lab map will also display information about shifts and staffing including their roles and who is working on-site/remotely.

**Note: Due to time constraints and priority of features implemented, the following features were not able to be implemented:*

- Administrator account privileges
- Lab room creation

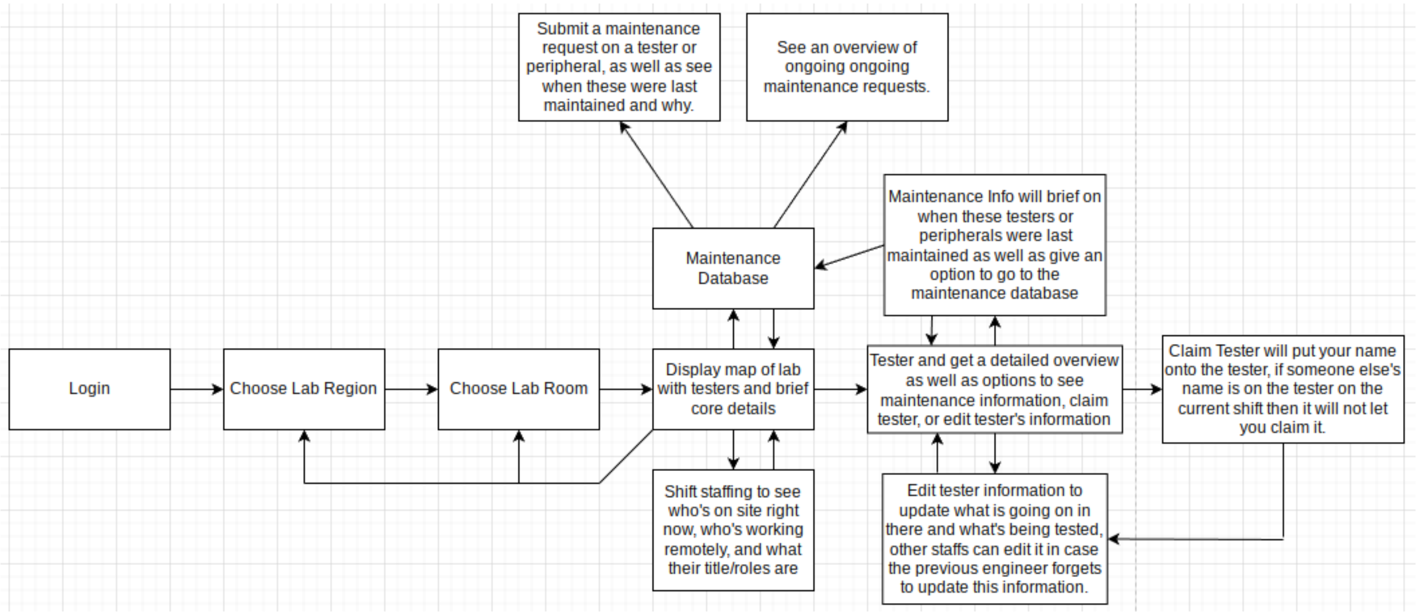
4.2 Product Functions

- Create a lab region*
- Create a room *
- Access a lab (where the lab map will be displayed)
  - Claim a tester
  - Display tester information
  - Edit tester information
  - See maintenance info / submit maintenance request.
  - Display information about shift staffing

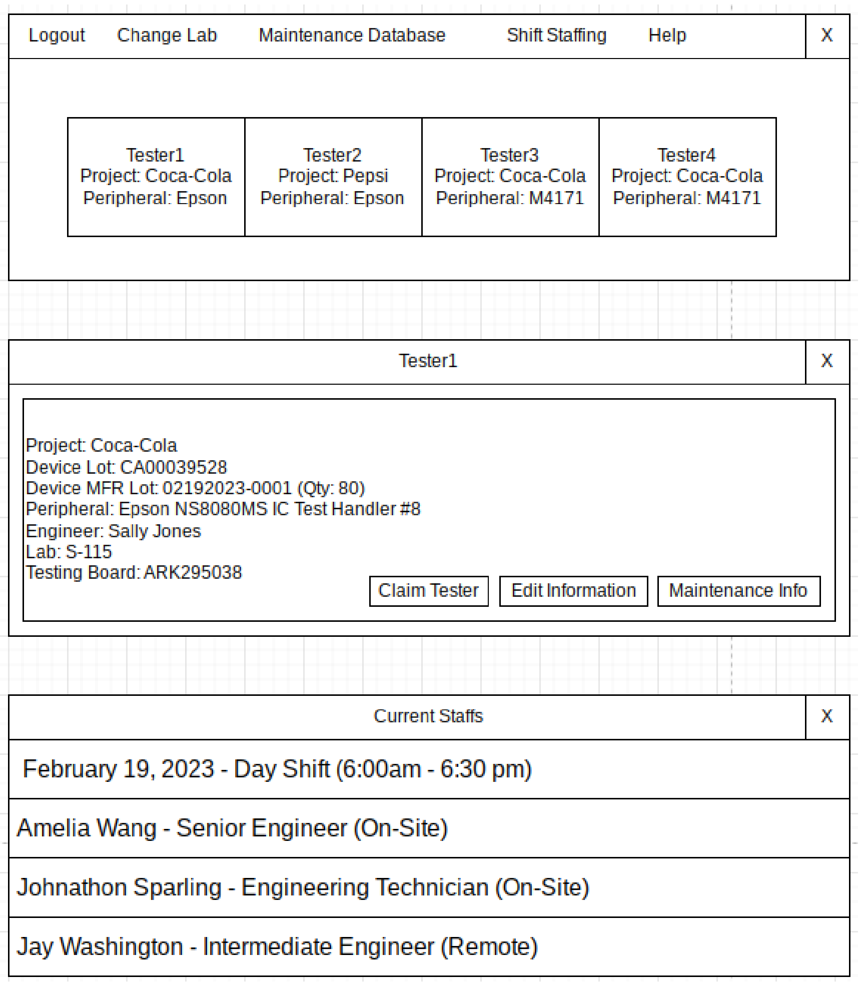
**Note: Due to time constraints and priority of features implemented, the following features were not able to be implemented:*

- Lab region creation
- Lab room creation

Planned Flowchart of Program


Submit a maintenance request on a tester or peripheral, as well as see when these were last maintained and why.
See an overview of ongoing ongoing maintenance requests.
Maintenance Info will brief on when these testers or peripherals were last maintained as well as give an option to go to the maintenance database
Maintenance Database
Login
Choose Lab Region
Choose Lab Room
Display map of lab with testers and brief core details
Tester and get a detailed overview as well as options to see maintenance information, claim tester, or edit tester's information
Claim Tester will put your name onto the tester, if someone else's name is on the tester on the current shift then it will not let you claim it.
Shift staffing to see who's on site right now, who's working remotely, and what their title/roles are
Edit tester information to update what is going on in there and what's being tested, other staffs can edit it in case the previous engineer forgets to update this information.

Mockup GUI of the program



Tester1
Project: Coca-Cola
Peripheral: Epson

Tester2
Project: Pepsi
Peripheral: Epson

Tester3
Project: Coca-Cola
Peripheral: M4171

Tester4
Project: Coca-Cola
Peripheral: M4171

Tester1

Project: Coca-Cola
Device Lot: CA00039528
Device MFR Lot: 02192023-0001 (Qty: 80)
Peripheral: Epson NS8080MS IC Test Handler #8
Engineer: Sally Jones
Lab: S-115
Testing Board: ARK295038

Current Staffs

February 19, 2023 - Day Shift (6:00am - 6:30 pm)

Amelia Wang - Senior Engineer (On-Site)

Johnathon Sparling - Engineering Technician (On-Site)

Jay Washington - Intermediate Engineer (Remote)

4.3 Technical requirements

- The application will display a digital map with an accurate representation of where each device is on each floor. The map will ideally be interactive upon multiple improvements so users can interact with testers displayed on the map. This can be built using Java.

- An SQL database will be used to store information about each tester and related information in the lab.

- The application will provide user authentication to make sure only authorized persons have access to the system. This can be achieved using hashing algorithms and storing hashes in a database.*

- The device will need to have JRE installed in order to run the software. Therefore, the hardware and the Operating System must be compatible with the most recent version of JRE. There must also be sufficient storage to store the application and any associated files. The device must also have access to the local database.

**Note: Due to time constraints and priority of features implemented, the following features were not able to be implemented:*

- Password hashing

4.4 Operating Environment

This program will be portable to use on a variety of hardware and operating systems. The client company is expected to work with desktop PCs that use both Linux based operating systems, such as Red Hat and CentOS, as well as Windows 7 and 10 operating systems. The hardware of each tester PC may highly vary, but this will be operatable in low end PCs such as laptops in order to maximize portability.

4.5 Design and Implementation Constraints

As per client’s projects are highly confidential, as we, the developers, do not have access to any of the data. We are required to make the software highly flexible and highly modular to ensure the demands are met by the client. The client will also be assumed to maintain the software to incorporate it with any databases that they wish to use.

4.6 User Documentation

The README file will be delivered with and serve as the primary source of information for users, as it will contain the majority, if not all, of the information needed to use this software.

4.7 Assumptions and Dependencies

The databases will be assumed to be in SQL formats. If this does change then we will include it in our README.txt in our software zip file.

5. External Interface Requirements

5.1 User Interfaces

The main software components of this project include a basic login screen that will prompt the user to enter a username and password and a map screen which will display a list of testers. These testers will be displayed as boxes adjacent to each other, with the name of each tester inside each box. A device tester screen will allow users to view and modify tester details. This screen is shown as a simple list of information about each tester. Some other components include an account screen, shift staffing information screen, and a maintenance screen that can be used to request maintenance on a tester. These screens will follow a similar design layout to the screens described above.

5.2 Hardware Interfaces

The software will be coded in Java, allowing it to run on all systems compatible with the Java Runtime Environment. This software is a desktop application and will only be tested and optimized for desktops. A keyboard may be necessary, although not required, to input tester information into text boxes within the program. A mouse will be necessary to browse through the application.

5.3 Software Interfaces

This software will be connected to a local SQL database. All information put into the software will be stored in this database. All accounts, maintenance, regions, staff shifts, and testers will be stored. All testers will have the following details: name, project associated with tester, a unique peripheral id, device id, manufacturing lot number, device quantity, staff assigned to the tester, and the respective staff’s role. The user will also be able to modify the following tester details: devices and peripheral.

5.4 Communications Interfaces

The primary form of communication in the system is the interaction between Java and the MySQL database.

6. System Features

6.1 Logging in

6.1.1 Description and Priority

High Priority. Users/Administrators will have the option to log in with their respective credentials (username, password).

6.1.2 Stimulus/Response Sequences

User runs program->System displays a login page->user enters a username and password into text fields.

6.1.3 Functional Requirements

1. Read input from “Username” and “Password” entry fields in the GUI.
2. Compute the hashing for the password. This will be done within java code.*
3. Access the database and compare the username and password hash entries.
4. If successful grant the user access to the lab region, otherwise prompt them to reenter credentials.

.

**Note: Due to time constraints and priority of features implemented, the following features were not able to be implemented:*

- Password hashing

6.2 Choose a Lab Region

6.2.1 Description and Priority

Users will be able to select a lab region based on their respective location. Each lab region **will** access data specific to said region. This feature is a moderate priority (5) feature, as the primary usage will start at the local level.

6.2.2 Stimulus/Response Sequences

User clicks the change region drop down menu and selects region of choice.-> The screen will now display a map of the devices from that region in their corresponding location if the user were to enter the lab and facing upwards from the bottom from the map perspective. After the region is selected, the user can complete actions and manage information under the new region.

6.2.3 Functional Requirements

By selecting the lab region, it will connect to the local MySQL database to retrieve appropriate data. If incorrect information is returned, the situation will be handled individually.

6.3 Maintenance

6.3.1 Description and Priority

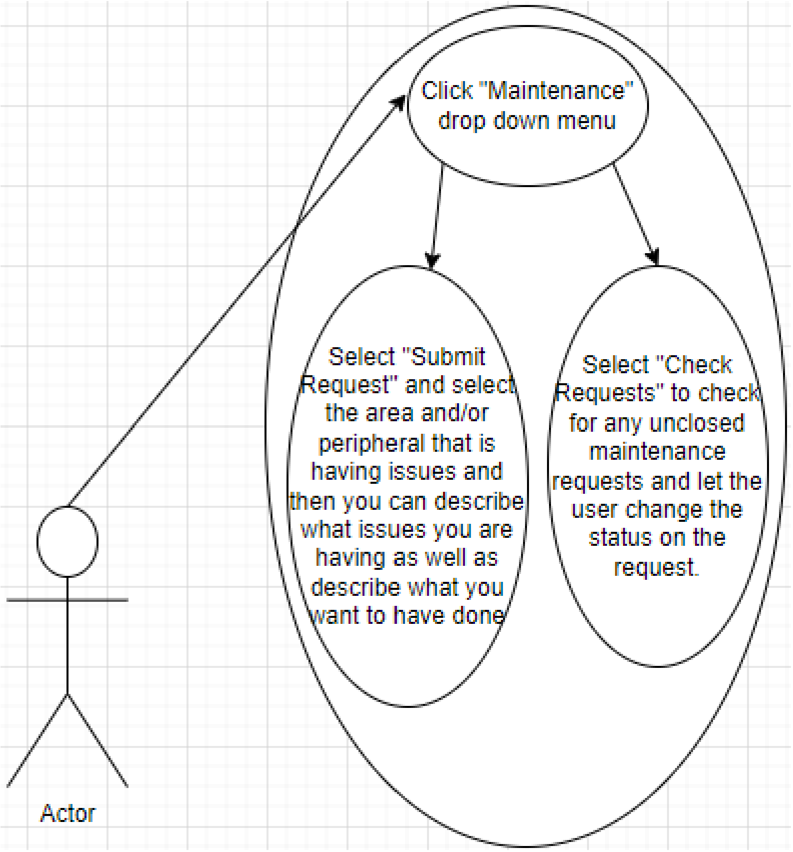
Users will be able to select testers from a menu of available testers at each region. This will be a high priority (7) feature. This allows the user to submit a maintenance ticket on a peripheral or tester if there is something wrong. The user can describe the issue and ask maintenance to diagnose and attempt to solve the issue.

6.3.2 Stimulus/Response Sequences

By clicking the maintenance drop down menu will allow the user to submit a work order for a specified tester or peripheral and let the user describe the issue they are having. The user will also have the option to check for any unclosed work orders to check, when they were submitted, and allow the user to modify the status.

6.3.3 Functional Requirements

REQ-1: Must be connected to the local MySQL database
REQ-2: Event handler will detect user interactions as they click on the listed devices/testers
REQ-3: Relevant information from the chosen tester will be obtained from the database and displayed in the GUI.


Click "Maintenance" drop down menu
Select "Submit Request" and select the area and/or peripheral that is having issues and then you can describe what issues you are having as well as describe what you want to have done
Select "Check Requests" to check for any unclosed maintenance requests and let the user change the status on the request.
Actor

6.4 Choose a Lab Tester

6.4.1 Description and Priority

Users will be able to select testers from a menu of available testers in each region. This will be a high priority (9) feature, as all information relevant to the tester will be shown.

6.4.2 Stimulus/Response Sequences

By clicking a tester, a menu will pop up with current tester details (Tester name, project name, peripheral id, device id, manufacturing lot, device quantity, staff attending to the tester, and the respective staff's role.

6.4.3 Functional Requirements

REQ-1: Must be connected to the local MySQL database
REQ-2: Event handler will detect user interactions as they click on the listed devices/testers
REQ-3: Relevant information from the chosen tester will be obtained from the database and displayed in the GUI.

6.5 Claim/ Modify a Lab Tester

6.5.1 Description and Priority

Users will be able to modify testers after selecting them from the lab map. This will be a high priority (9) feature; information for testers may need to be modified.

6.5.2 Stimulus/Response Sequences

Users can locate the “Modify Tester” button on the bottom right section of the application window. Upon selecting this option, users can add devices and peripherals to an existing tester. To confirm these changes, the user must select the “Modify Details” button on the bottom right of the pop-up window. Users can also claim the tester for use with the “Claim Tester” button to the left of the “Modify Details” button.

6.5.3 Functional Requirements

REQ-1: Must be connected to the local MySQL database
REQ-2: Ability to read user input from the “New Devices” text box and “New Peripheral” drop down box.
REQ-3: Once the “Modify Details” button is clicked, the program must be able to connect to the database and update the new information. The

updated information must also be immediately retrieved and displayed on the GUI.

6.6 Provide “Help” information

6.6.1 Description and Priority

A “Help” tab will be provided which will display a detailed manual of how to use the program. This will be of low priority (1) as even though it is important to educate the user about the proper use of the software, this is not necessary for the actual functionality of the program.

6.6.2 Stimulus/Response Sequences

User clicks on the “Help Tab”->A manual or about will be displayed.

6.6.3 Functional Requirements

REQ-1: Must be connected to the local SQL database
REQ-2: Retrieve LONGSTRING information and display it on the GUI

6.7 User/ Administrator Accounts Tab*

6.7.1 Description and Priority

Medium priority (7). Users and administrators can see their name, role, and project. Administrators will also have the option of creating new user accounts.

6.7.2 Stimulus/Response Sequences

Users and Administrators click the “Account” button on the top left of the main lab map window. A drop-down menu will appear with two options: “Information” and “Log Out”. Upon clicking “Information,” a pop-up menu will appear, detailing the account’s name, role, and project. Administrators will have another button located at the bottom of the pop-up menu labelled “Create Account.” This will allow new user accounts to be made.

6.7.3 Functional Requirements

REQ-1: Must be connected to the local SQL database

**Features are only available for users. Administration accounts have not been implemented due to time and priority constraints.*

6.8 Staffing

6.8.1 Description and Priority

Information about present shifts and present staff will be displayed when clicking on the "Shift Staffing" option.

6.8.2 Stimulus/Response Sequences

User clicks on "Shift Staffing" option->A list of shifts and staffing will be displayed.

6.8.3 Functional Requirements

TBD

7. Other Nonfunctional Requirements

7.1 Performance Requirements

- This software is a self-contained product that will perform all functions as expected.
- Since the software is coded in Java, it should run efficiently on all major platforms.
- The software is expected to be run on desktops. Mobile platforms may not be supported.

7.2 Safety Requirements

- Electrical Hazards: Without electronic access, the software will be unusable.
- Service Failure Hazards: In the event of a database malfunction, data may be irretrievable and/or inaccurate until fixes are implemented

7.3 Security Requirements

- Technicians and Administrators will have different login credentials. Passwords will be hashed to allow for greater security.
- Naturally, all databases are at risk due to outside attacks, such as SQL Injections. All data entered in text boxes within the Lab Map software will be verified to protect the database's data from being leaked.

7.4 Software Quality Attributes

- Correctness is one of the most important characteristics for this project. The information being retrieved from the database and displaying correctly to the user is critical for the system to work.
- Portability is important to this project. It will run on all systems compatible with JRE.
- Software will be reliable. It will not crash and shutdown from expected user interaction.
  - Each feature will be tested at least 10 times to ensure reliability.
- Software will emphasize ease of use. A help drop down bar will be included in the software to give users access to the software's manual.
- Software can be modified for each unique lab setting. Software can be reused for all lab experiments to track equipment.

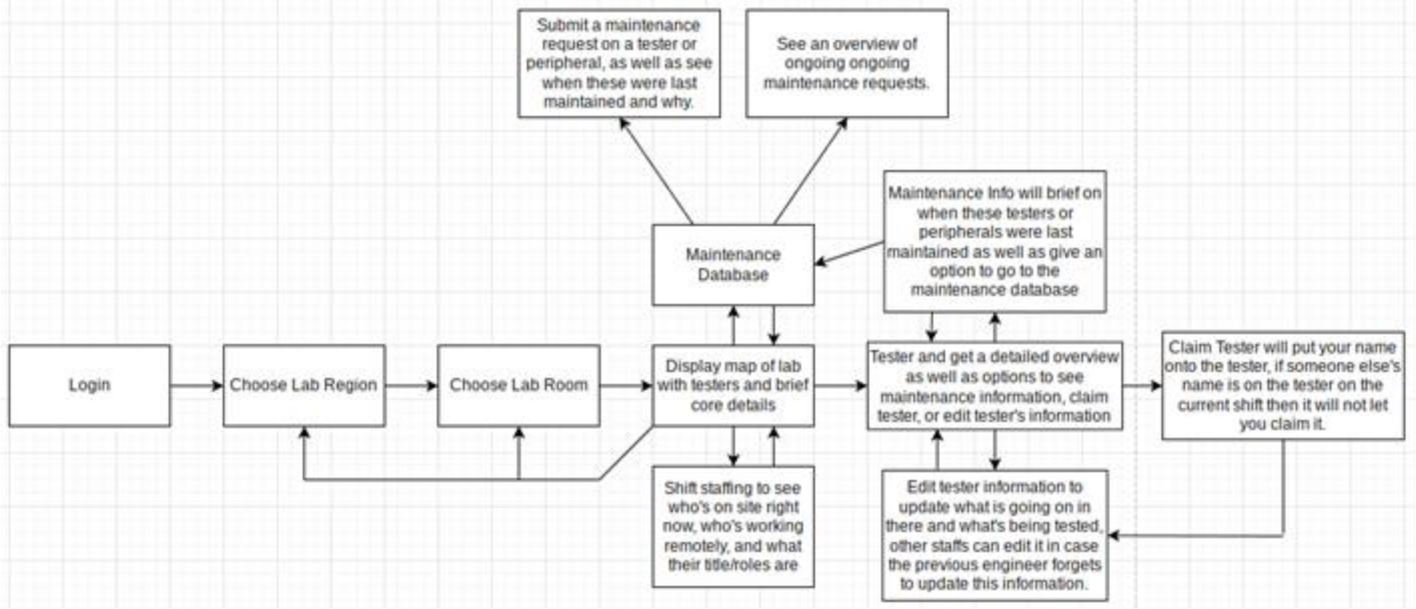
7.5 Business Rules

- Employees will be able to login, view account info, view testers and tester information, claim and modify testers, request maintenance, and view staffing information.
- Admins will be able to do all of the above as well as creating new users.

8. System Architecture

8.1 Architectural Design

Develop a modular program structure and explain the relationships between the modules to achieve the complete functionality of the system. This is a high-level overview of how the responsibilities of the system were partitioned and then assigned to subsystems. Identify each high-level subsystem and the roles or responsibilities assigned to it. Describe how these subsystems collaborate with each other in order to achieve the desired functionality. Don't go into too much detail about the individual subsystems. The main purpose is to gain a general understanding of how and why the system was decomposed, and how the individual parts work together. Provide a diagram showing the major subsystems and data repositories and their interconnections. Describe the diagram if required.


Submit a maintenance request on a tester or peripheral, as well as see when these were last maintained and why.
See an overview of ongoing ongoing maintenance requests.
Maintenance Info will brief on when these testers or peripherals were last maintained as well as give an option to go to the maintenance database
Maintenance Database
Login
Choose Lab Region
Choose Lab Room
Display map of lab with testers and brief core details
Tester and get a detailed overview as well as options to see maintenance information, claim tester, or edit tester's information
Claim Tester will put your name onto the tester, if someone else's name is on the tester on the current shift then it will not let you claim it.
Shift staffing to see who's on site right now, who's working remotely, and what their title/roles are
Edit tester information to update what is going on in there and what's being tested, other staffs can edit it in case the previous engineer forgets to update this information.

8.2 Decomposition Design

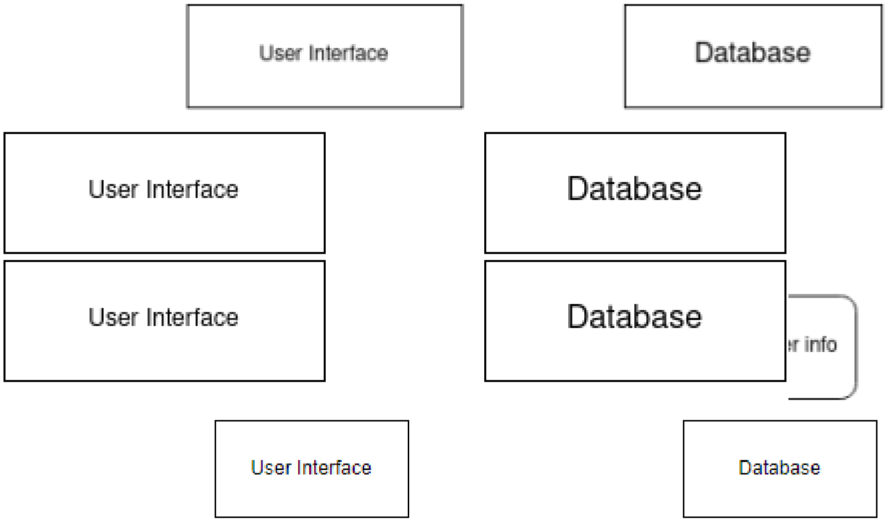
Provide a decomposition of the subsystems in the architectural design. Supplement with text as needed. You may choose to give a functional description or an object- oriented (OO) description. For a functional description, put top -level data flow diagram (DFD) and structural decomposition diagrams. For an OO description, put subsystem model, object diagrams, generalization hierarchy diagram(s) (if any), aggregation hierarchy diagram(s) (if any), interface specifications, and sequence diagrams here.

Login*

*Password hashing is not implemented due to time and priority constraints.

Choose a lab region

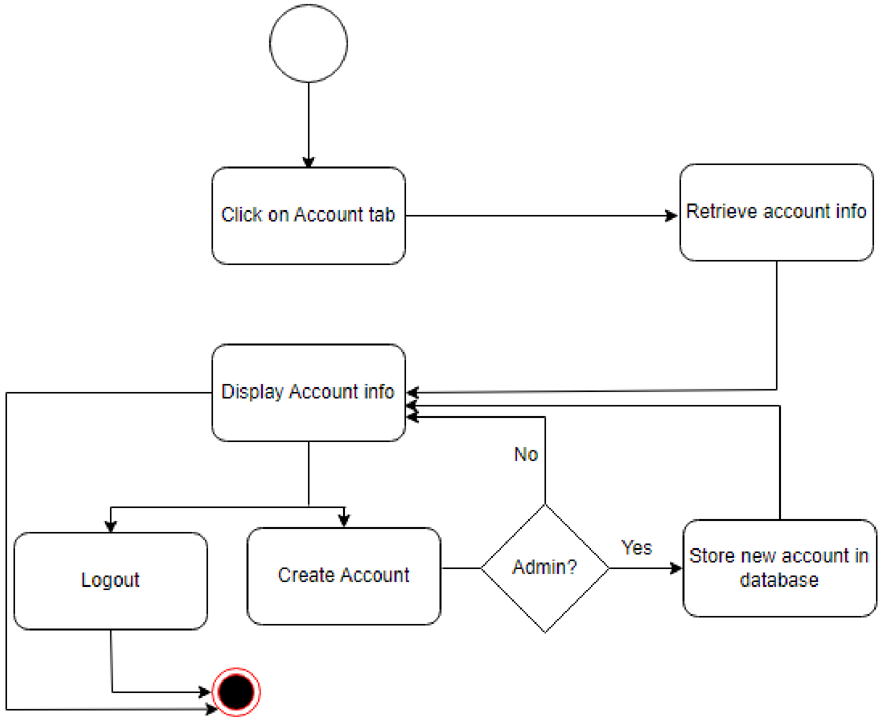
Claim/Modify a lab tester



Display Help information

Staffing

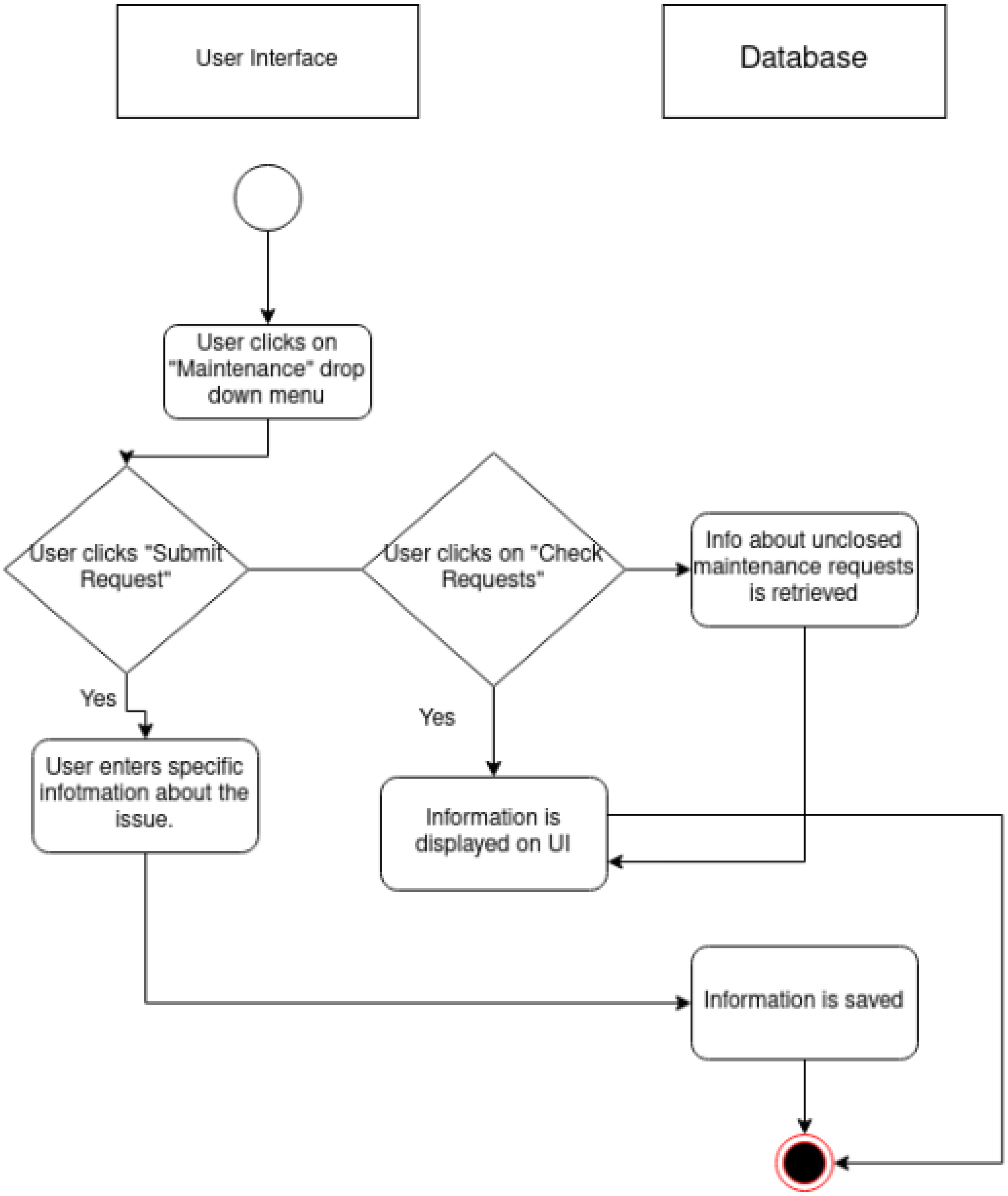
Account Information/Options



**Note: Due to time constraints and priority of features implemented, the following features were not able to be implemented:*

- Administrator functionalities

Maintenance


User Interface
Database
User clicks on "Maintenance" drop down menu
User clicks "Submit Request"
User clicks on "Check Requests"
Info about unclosed maintenance requests is retrieved
Yes
Yes
User enters specific infotmation about the issue.
Information is displayed on UI
Information is saved

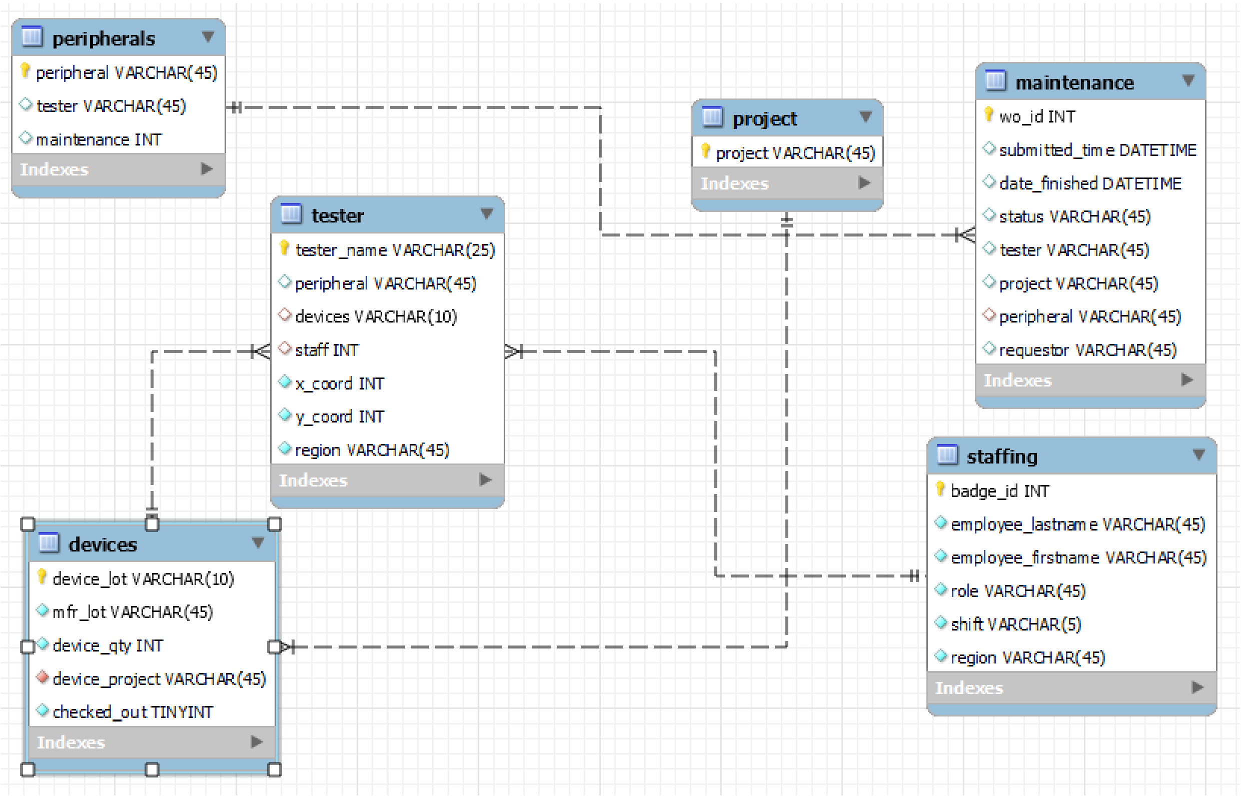
8.3 Design Rationale

Discuss the rationale for selecting the architecture described in 6.1 including critical issues and trade/offs that were considered. You may discuss other architectures that were considered, provided that you explain why you didn't choose them.

A database is essential to this project. All information will flow in and out of one database as it interacts with user inputs from the GUI. Acquired by Oracle, MySQL Workbench was chosen because it is a free, open-source tool backed by its reliability. It is also one of the most popular databases due to its ease of use, scalability, and data encryption.

9. Data Design (only applicable if your project has a database component)

9.1 Entity relationship design



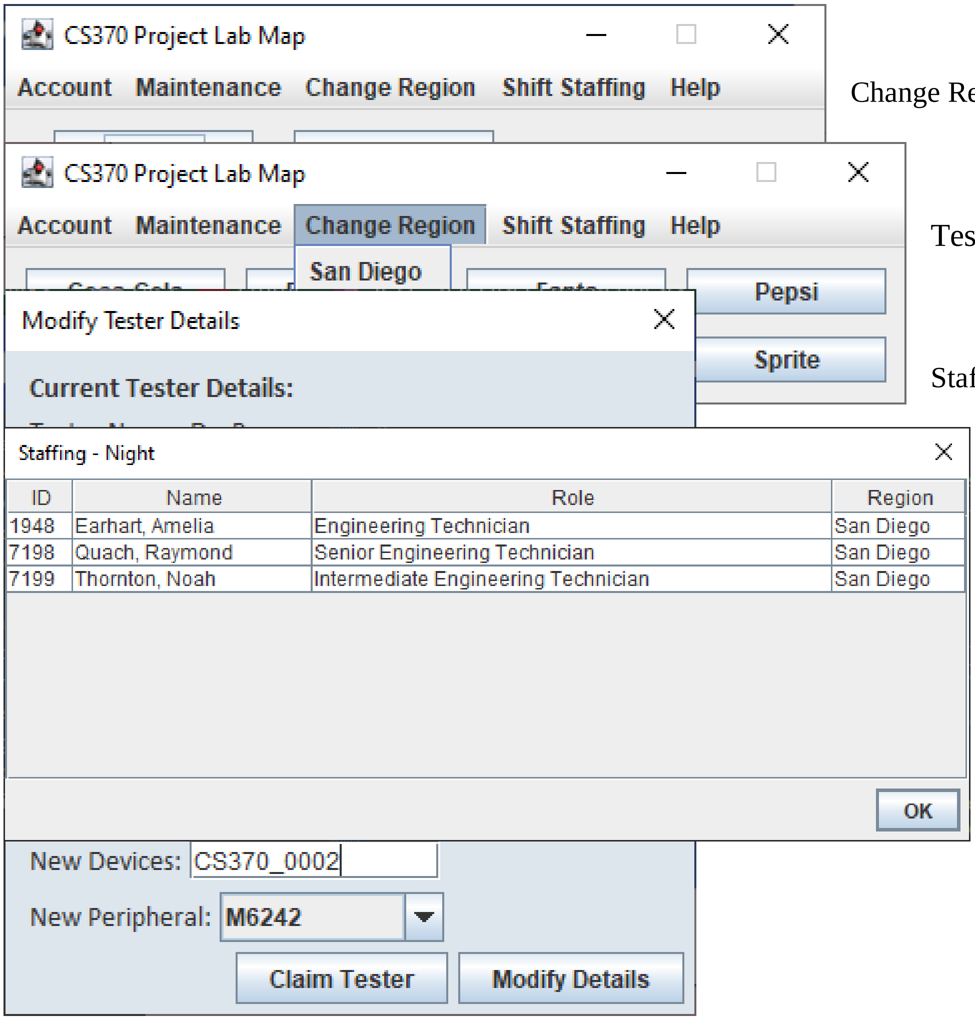
9.2 Data dictionary

- *Badge ID* – The user's associated badge ID, unique to each employee.
- *Checked Out* – If a device lot is checked out at a tester or not.
- *Device Lot/Devices* – Usually interchangeable in terms, split from manufacturing lot for testing organizational purposes. These should never be at two testers at a time.
- *Device Qty* – The number of devices in a device lot.
- *Manufacturing Lot (mfr_lot)* – The lot which the devices came from the fabrication.
- *Peripheral* – Machines and methods of testing the device in different conditions.
- *Requestor* – Person who submitted the maintenance request.
- *Role* – The employee's role in the company.
- *Shift* – Employee's assigned shift, either night or day shift.
- *Status* – The status of whether a maintenance is active, finished, or not yet started.
- *Tester* – The PC and hardware utilized to test the device.
- *Tester X Coordinate (x_coord)* – Tester's location on the x-coordinate on the GUI
- *Tester Y Coordinate (y_coord)* – Tester's location on the y-coordinate on the GUI
- *Work Order ID (wo_id)* – The ticket attached to a maintenance request.

Appendix A: Screen Images

Display screenshots showing the interface from the user's perspective. These can be hand- drawn or you can use an automated drawing tool. Just make them as accurate as possible. (Graph paper works well.). Also

provide a discussion of screen objects and actions associated with those objects.
Main Menu:



Change Region Menu Option:

Tester Detail:

Staff Shift Window:

ID	Name	Role	Region
1948	Earhart, Amelia	Engineering Technician	San Diego
7198	Quach, Raymond	Senior Engineering Technician	San Diego
7199	Thornton, Noah	Intermediate Engineering Technician	San Diego

New Devices: CS370_0002

New Peripheral: M6242