GNG2101

Design Project User and Product Manual

THE MATRIX

Submitted by:

Rayyan Faruquee, 300257795

Luka Flanigan, 300375708

Gurkan Emre Uslu, 300350916

Takudzwa Gwavava, 300388232

AbdulWahid Ahmed, 300331802

03 December 2024

University of Ottawa

Table of Contents

Table of Contents iii

List of Figures v

List of Tables vi

List of Acronyms and Glossary vii

1 Introduction 1

2 Overview 2

2.1 Cautions & Warnings 3

3 Getting Started 4

3.1 Configuration Considerations 4

3.2 User Access Considerations 4

3.3 Accessing/setting up the System 4

3.4 System Organization & Navigation 4

3.5 Exiting the System 6

4 Using the System 7

4.1 Detail Search 7

4.2 Check-in/Check-out 8

4.3 Asset add and remove 10

5 Troubleshooting & Support 11

5.1 Error Messages or Behaviors 11

5.2 Maintenance 11

5.3 Support 12

6 Product Documentation 13
6.1Detail Search 13
6.1.1 BOM (Bill of Materials) 13
6.1.2 Equipment list 13
6.1.3 Instructions 14
6.2Check-in/Check-out 16
6.2.1BOM 16
6.2.2Equipment List 17
6.2.3Instructions 17
6.3Asset add and remove 21
6.3.1BOM 21
6.3.2Equipment List 22
6.3.3Instructions 22
7Testing & Validation 25
8Conclusions and Recommendations for Future Work 31
9Bibliography 32
APPENDICES 34
APPENDIX I: Design Files 34

List of Figures

No table of figures entries found.

List of Tables

Table 1. Acronyms vii

Table 2. Glossary vii

Table 3. Referenced Documents 34

List of Acronyms and Glossary

Table 1. Acronyms

Acronym	Definition

Table 2. Glossary

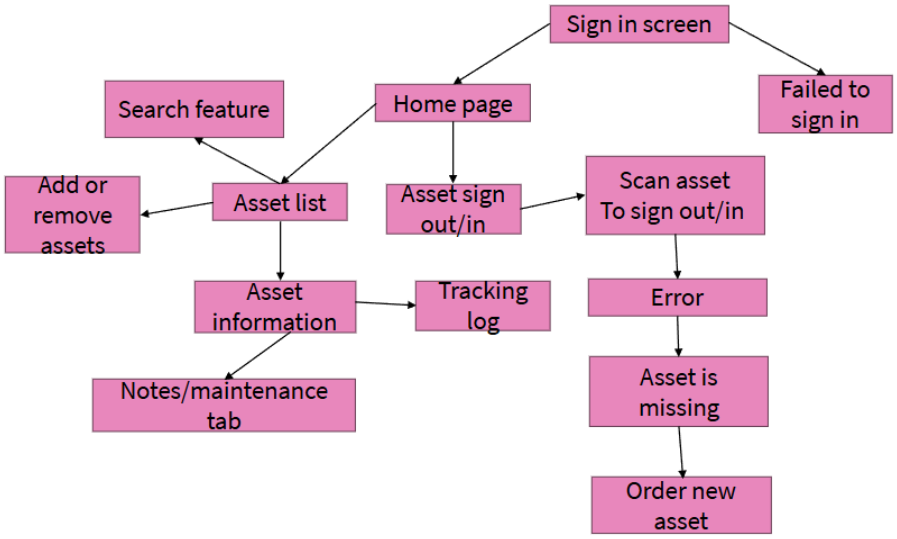
Term	Acronym	Definition

1 Introduction

This User and Product Manual (UPM) provides the information necessary for doctors, Care givers and system admins to effectively use the Matrix Asset Management System (MAMS) and for prototype documentation. The document will start by explaining all basic terminology and safety factors required for setting up and using the product followed by a detailed description of setting up the product; this will include a list of necessary applications and external hardware. The following sections will explain how to properly use the application including: how to sign into the system, how to sign in/out assets, how to add/remove assets and how to use the search engine. The following section will document the development stage of the project which includes all testing/prototype models as well as a bill of materials. This document aims to inform IT members on how to set up the system as well as troubleshooting for any possibles errors. Staff on how to properly operate the system. And any future developers aiming to expand on the work. For use of the MAMS requires a valid license to operate; using the system without a valid license will lead to a legal presute.

2 Overview

Bethany Children's Health Center is actively seeking to enhance its asset management processes by implementing a cheaper modern and efficient inventory system. The goal of this project is to develop a solution that enables staff to check out equipment seamlessly by scanning both a barcode on the equipment and their staff badge. This system not only aims to improve inventory control and accountability but also seeks to streamline workflow, ensuring that staff can focus on providing care without delays caused by inefficient equipment tracking. This is accomplished with a system that can be housed on-site, is simple to operate, and is low-cost.



This system is a Python-based code, with functions accessible in the UI depending on whether the user is a normal user or a system admin, with the ability to search for details, check-in or out assets, and add or remove assets from the system depending on the user's access authority.

2.1 Cautions & Warnings

Care should be taken not to point the scan gun at or near the eyes, as the scan gun emits a bright light while in use. It may cause damage to vision otherwise.

3 Getting Started

3.1 Configuration Considerations

The MAMS requires the use of a keyboard and scan-gun, in either a wired or wireless configuration. Output devices required are a monitor and a printer as a supplementary output device for printing labels corresponding to asset serial numbers.

3.2 User Access Considerations

This software will be used by two groups, the doctors, who will be the more frequent and larger group of users, and the system administrators, who will have about 3-4 in their group. Both groups will have access to the asset details UI, the search function and the check in and out function. Only the admins will have access to the add and remove asset UI.

3.3 Accessing/setting up the System

To activate the system, simply download the Python files and install them on the computer desktop. Run the code file "fileWriteToDatabase.py" to generate the Asset.txt file and Users.txt file. After initial installation, the assets can be imported into the by typing them into the asset text file, using the following order and inserting the $--$ break between the data: asset name, asset serial number, asset availability, check-out time (if applicable), check-out user (if applicable), asset notes, and file path for asset photo. For users, personnel information can be imported into the user text file in the following order, inserting the $--$ break between the data: first name, last name, employee number, whether or not they are an admin.

3.4 System Organization & Navigation

The asset inventory system is designed to streamline the management of assets by providing a user-friendly interface for tracking, checking in, and checking out items. The system consists of the following main components:

1. Main Menu (Home Page)

- Acts as the central hub for accessing all system features.
- Provides navigation paths to key functionalities, such as searching for assets, managing inventory, and performing check-in and check-out operations.

2. Navigation Paths

- Users can navigate the system using buttons and menus on the home page, which link to specific features.
- Each feature is accessible via a clear, intuitive navigation structure, ensuring ease of use for all users.

Main System Features

1. Search Interface

- Purpose: Allows users to search for assets by name or serial number.
- Navigation Path: From the home page, click the "Search" button to access the Search UI.
- Functionality:

    - Provides a search bar for entering keywords.
    - Displays matching results with details such as asset name, serial number, picture, and availability status.
    - Includes a "Details" button for viewing additional information about an asset.

2. Check In/Out Interface

- Purpose: Handles the borrowing (check-out) and returning (check-in) of assets.
- Navigation Path: From the home page, click the "Check In/Out" button to access this interface.
- Functionality:

    - Lists all assets with their details, including name, serial number, and availability.
    - Displays "Check Out" or "Check In" buttons based on the asset's current status.
    - Updates the availability status in real-time after actions are performed.

3. Asset Management

- Purpose: Enables administrators to add, edit, or remove assets from the inventory.
- Navigation Path: From the home page, click the "Manage Assets" button.
- Functionality:

    - Allows administrators to input details such as name, serial number, availability, and pictures.
    - Includes options to update or delete existing assets from the system.

4. Reporting and Logs

- Purpose: Provides an overview of asset usage, including check-in and check-out history.
- Navigation Path: From the home page, click the "Reports" button to view system logs.
- Functionality:

    - Displays logs of all asset transactions.
    - Includes filters for searching logs by asset name, user, or date.

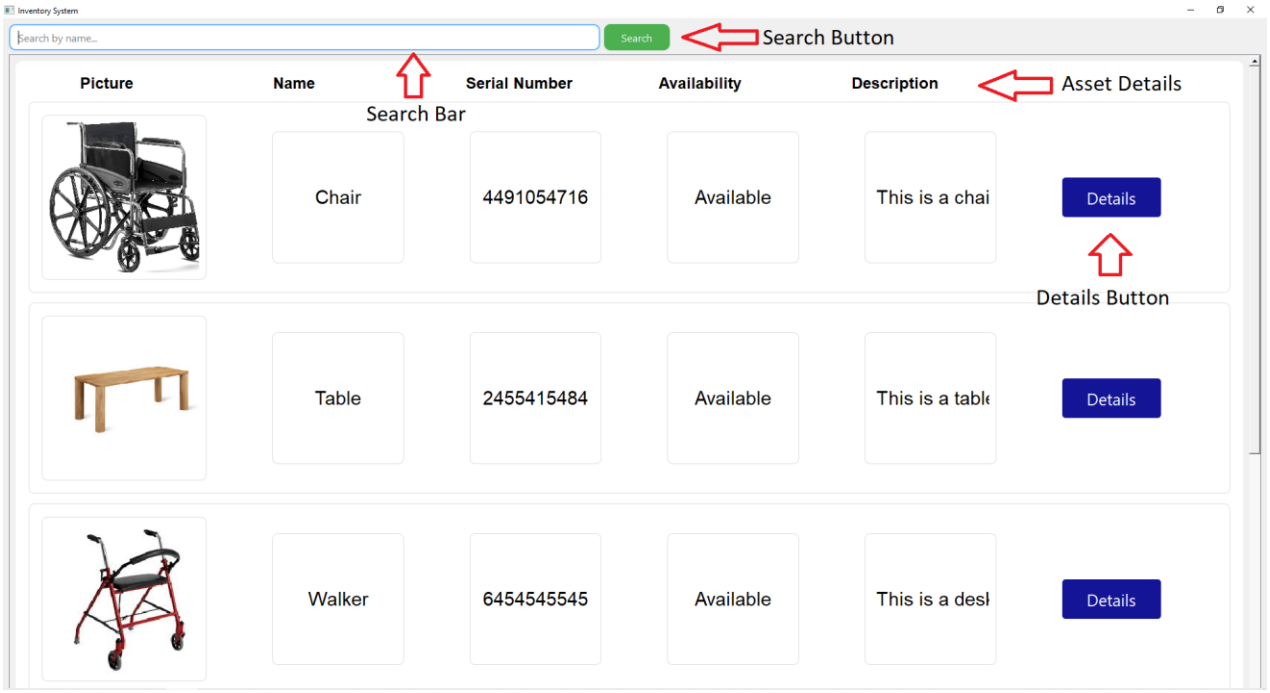
3.5 Exiting the System

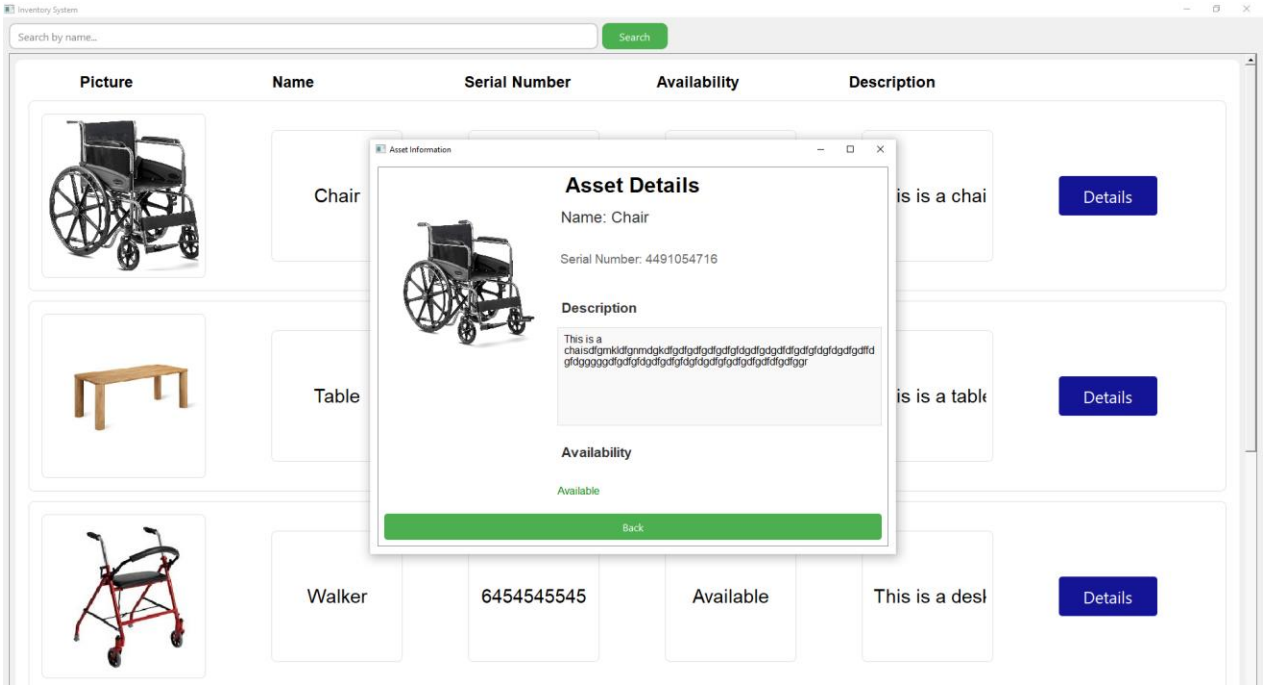
To shut down the system, the user must click on the red X button in the top right corner of the screen.

4 Using the System

The following sub-sections provide detailed, step-by-step instructions on how to use the various functions or features of the MAMS.

4.1 Detail Search





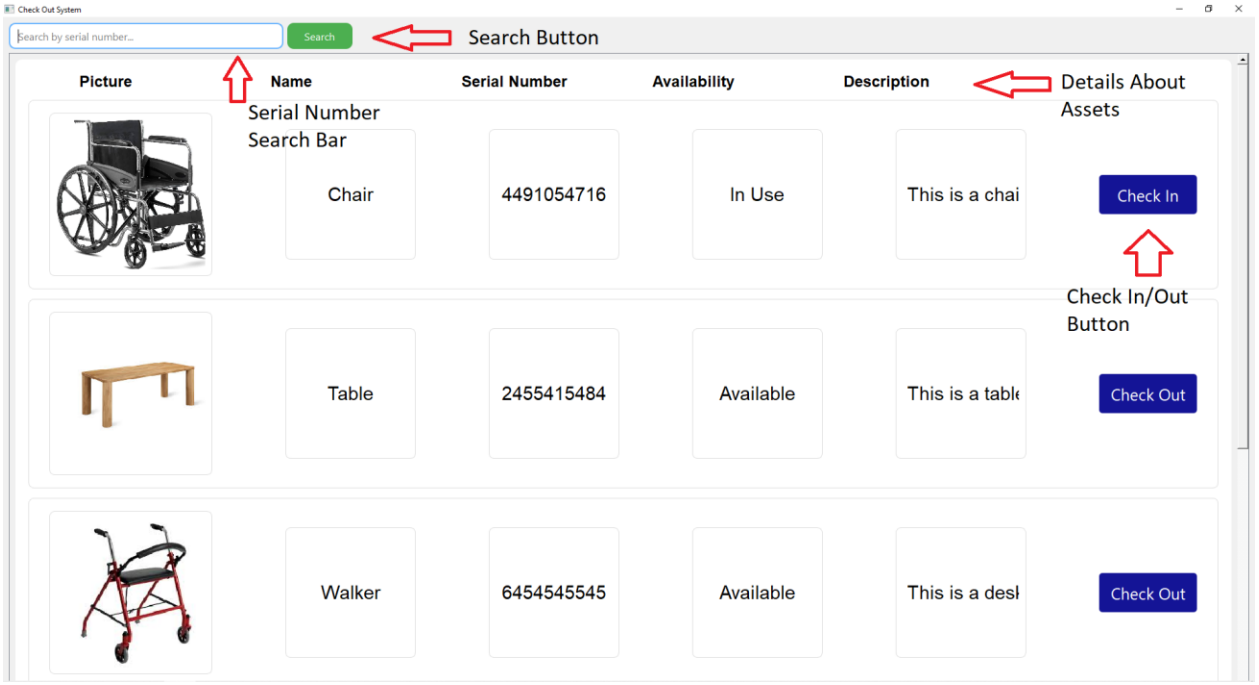
The Search UI is a critical part of the prototype system, allowing users to find and view asset details efficiently. This interface displays assets alongside their key information, such as pictures, names, serial numbers, availability status, and descriptions.

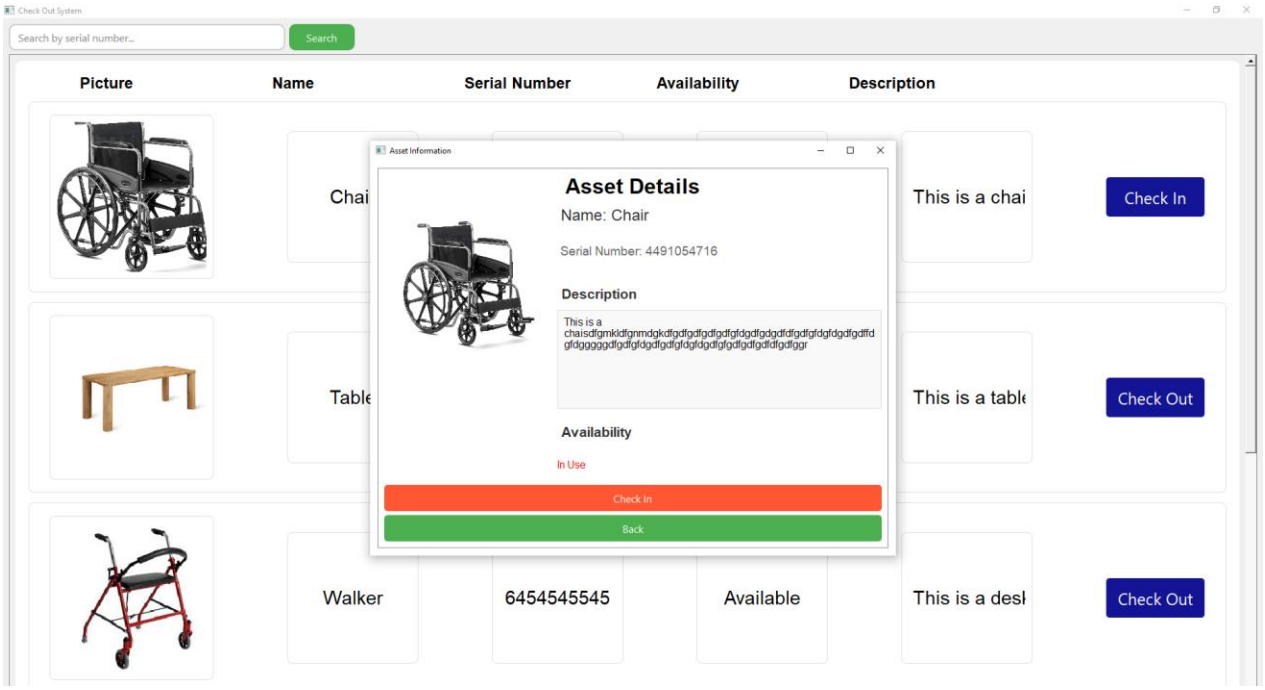
To use the search function, enter an asset's name into the search bar located at the top left corner of the interface, and then click on the search button. Any results that match the searched name will appear below on the screen.

To view asset details, click the button labelled “Details” located on the right-hand side of the asset row. A popup window will appear, providing a comprehensive view of the asset's specifications, notes, and any additional images. Click on the green button labelled “Back” to close the popup window and return to the main screen.

4.2 Check-in/Check-out

The Check In/Out UI is a core part of the system, managing the borrowing and returning of assets. It provides a detailed overview of all assets, their current availability, and allows users to update the system accordingly.

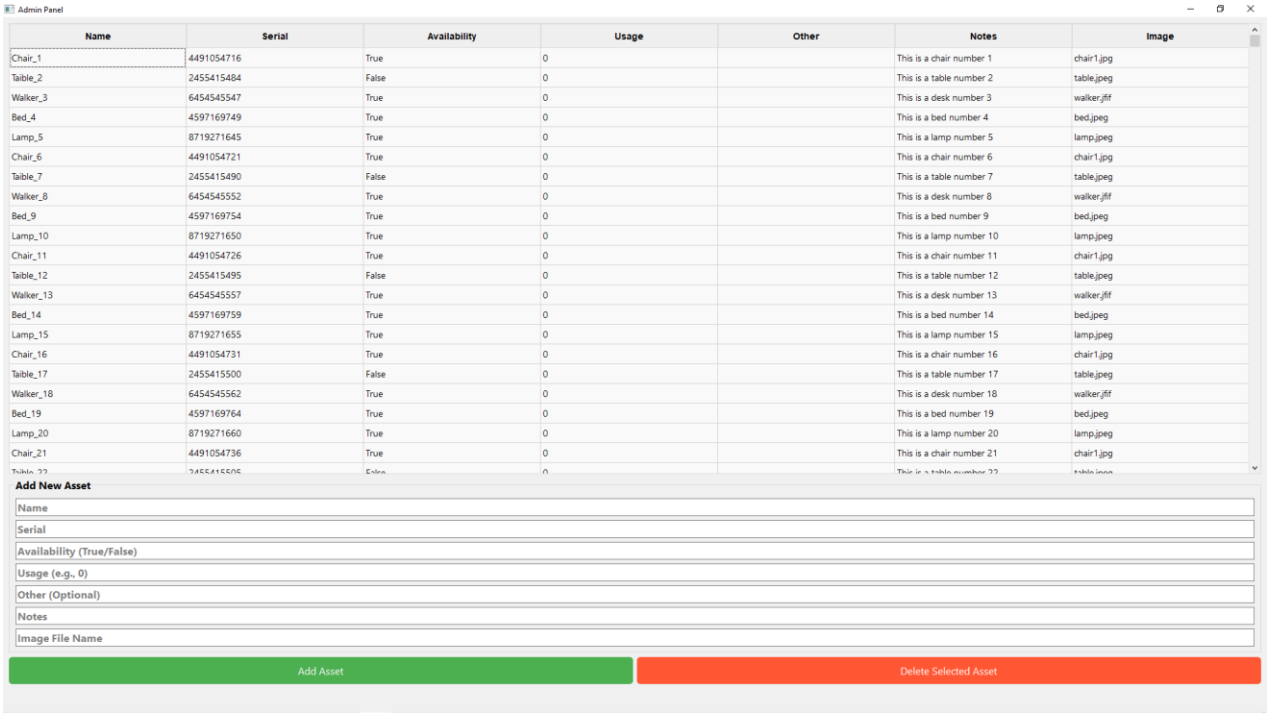

Check Out System
Search by serial number...
Search
Search Button
Picture
Name
Serial Number
Availability
Description
Details About Assets
Serial Number Search Bar
Chair
4491054716
In Use
This is a chai
Check In
Check In/Out Button
Table
2455415484
Available
This is a tabl
Check Out
Walker
6454545545
Available
This is a desl
Check Out


Check Out System
Search by serial number...
Search
Picture
Name
Serial Number
Availability
Description
Asset Information
Asset Details
Name: Chair
Serial Number: 4491054716
Description
This is a chaisdfgmkldfgnmdgkdfgdfgdfgdfgfdgdfgdgdfdfgdfgfdgfdgdfgdffd gfdgggggdfgdfgfdgdfgdfgfdgfdgdfgfgdfgdfdfgdfggr
Availability
In Use
Check In
Back
This is a chai
Check In
This is a tabl
Check Out
Walker
6454545545
Available
This is a desl
Check Out

To check-in or out an asset, either scan the barcode using the scan-gun, or type the serial number into the search bar in the top left corner and click on the search button. The system will display the requested asset with either a “Check-In” or “Check-Out” button on the right side of the asset row depending on whether the asset is available or not. When the button is clicked, a popup will appear with the asset information, with two buttons at the bottom. The green button labelled “Back” will return the user to the main screen, while the red button, labelled either “Check-In” or “Check-Out” will complete the action and change the availability of the asset in the system. If the red button was pressed, the popup will automatically close after 3 seconds and return to the main screen.

4.3 Asset add and remove

Accessible only by system admins, the asset add-and-remove UI is separate from to avoid tampering with. This page allows the admin to add new assets to the system or remove existing assets. To add a new asset, the admin needs to fill the information cells at the bottom of the page with the required information, as indicated within the cells. Once this is done, click on the green button in the bottom left corner of the screen labelled “Add Asset” to add the asset to the database. To remove an existing asset, click on the asset from the presented list in the centre of the page, then click on the red button in the bottom right corner of the screen labelled “Delete Asset” to relete the asset from the system.



5 Troubleshooting & Support

5.1 Error Messages or Behaviors

In the case of attempting to check out an asset, the asset may not appear when the asset barcode is scanned. If this occurs, attempt to scan the asset again. If the issue persists with that specific asset, contact the admin to verify whether that asset is registered in the system, and have them add the asset to the database if it does not exist.

If after checking out an asset and the system does not return to the starting screen, clear the search bar in the top left corner of the screen and click on the search button.

5.2 Maintenance

Regular maintenance of the system involves maintenance personnel reviewing the Asset.txt file and inspecting the layout for any irregularities, such as data truncating. Lines that have been truncated should be straightened out and formatted using the proper spacing characters ($--$) and order: asset name, asset serial number, asset availability, check-out time (if applicable), check-out user (if applicable), asset notes, and file path for asset photo

5.3 Support

In case system support is required, please contact either your institution's help desk or one of the code developers at 613-818-3689. When requesting assistance, please describe the issue and what messages are currently on screen.

6 Product Documentation

6.1Detail Search

6.0.1 BOM (Bill of Materials)

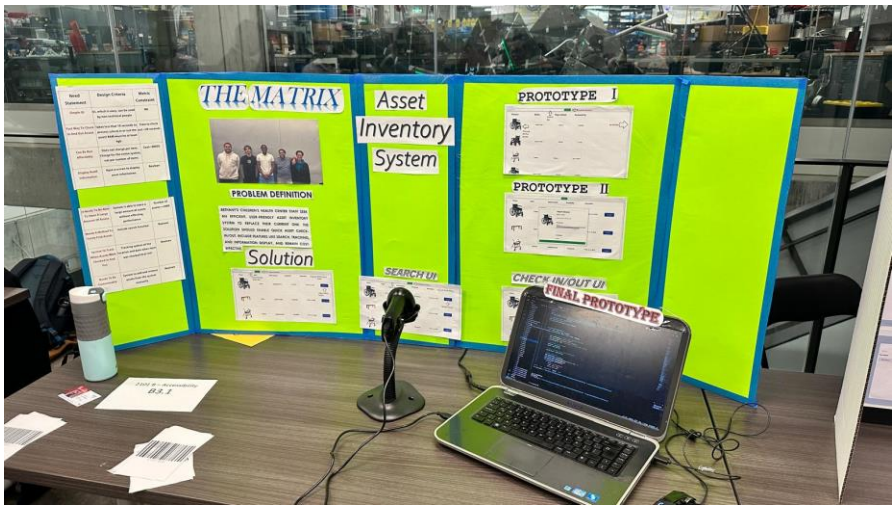
Items	Specifications	Quantity	Price ($)	Link
Spyder Python IDE	Python 3	1	0	https://www.spyder-ide.org/
Qt Designer	PyQt6	1	0	
Computer	Minimum: 8 Gb RAM Preferred: 16 Gb RAM	1		
Scanner		1	15	https://www.facebook.com/marketplace/item/1199374658171467/?ref=search&referral_code=null&referral_story_type=post

6.0.2 Equipment list

No additional equipment is required.

6.0.3 Instructions

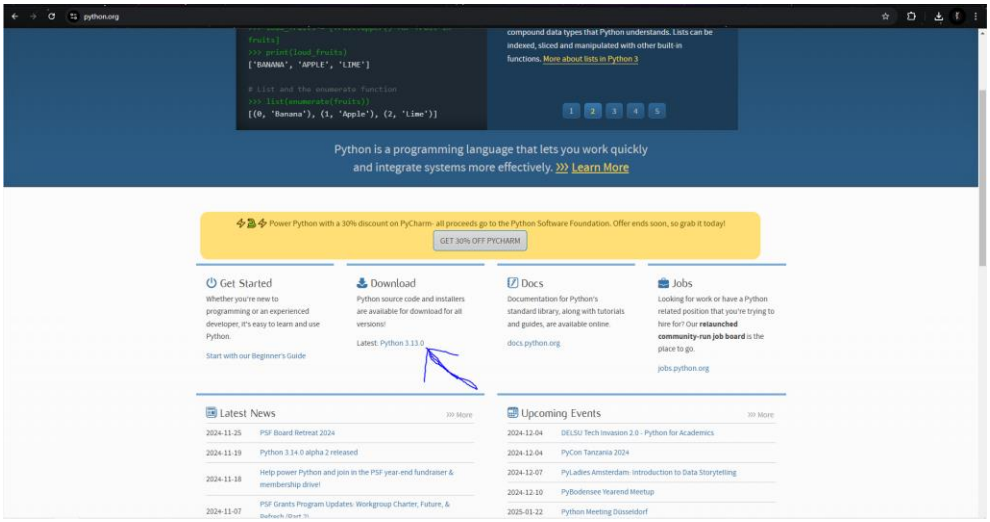
1. Setting Up the Environment



2. Installing Necessary Software

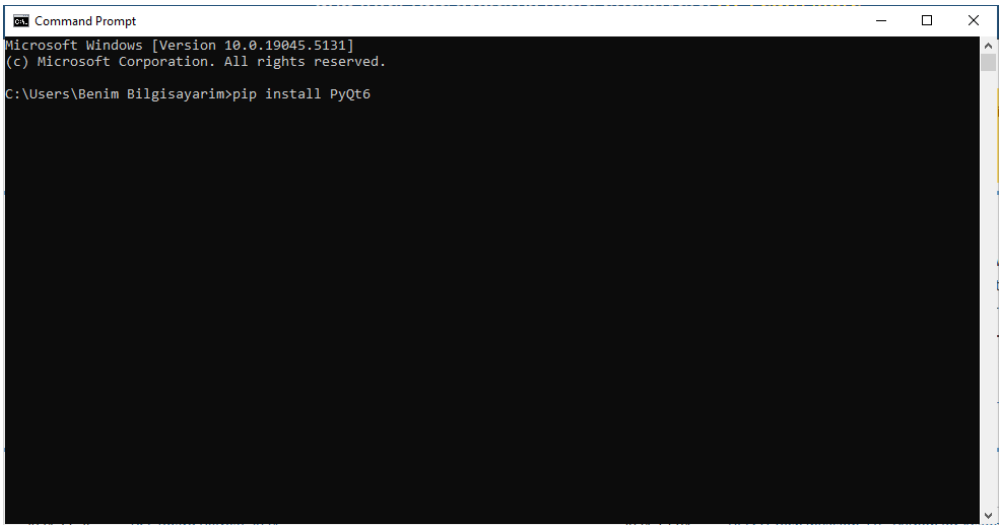
Install Python:

- Download and install Python from python.org.
- Add Python to the system PATH during installation.



Install PyQt6:

- Open a terminal or command prompt and run: pip install PyQt6



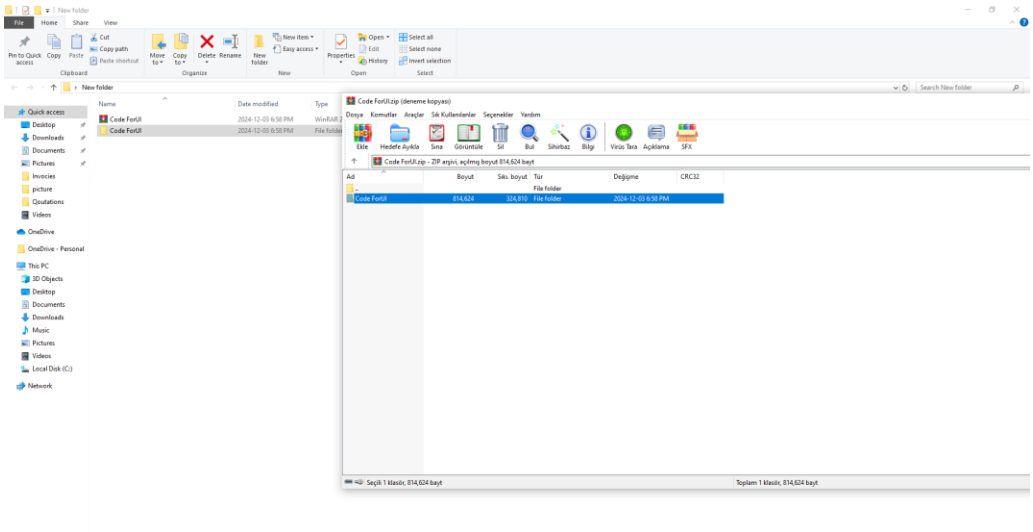
Install Additional Libraries (if needed): pip install -r requirements.txt

From the project folder, run:

3. Preparing the Files

- Extract the contents of the Code ForUI folder to a dedicated project directory on your computer.

- Open the Launcher.py script in your code editor to ensure it points to the correct MainWindow class (currently in test4.py).



Run Launcher.py to launch the GUI:

- Use the barcode scanner to scan asset serial numbers. The system should display asset details and availability status.

4. Testing the Prototype

- Verify that all asset data, including images and metadata, are displayed correctly in the GUI.
- Included Files for Reference

Python Scripts: For launching and managing the system.

Images: Asset images for demonstration.

Database Text Files: Assets.txt, Users.txt and __pycache__ file for preloaded data.

6.2Check-in/Check-out

6.2.1 BOM

Items	Specifications	Quantity	Price ($)	Link

Spyder Python IDE		1	0	https://www.spyder-ide.org/
Qt Designer		1	0	
Computer	Minimum: 8 Gb RAM Preferred: 16 Gb RAM	1		
Scanner		1	15	https://www.facebook.com/marketplace/item/1199374658171467/?ref=search&referral_code=null&referral_story_type=post

6.2.2 Equipment List

No additional equipment is required.

6.2.3 Instructions

1. Setting Up the Environment

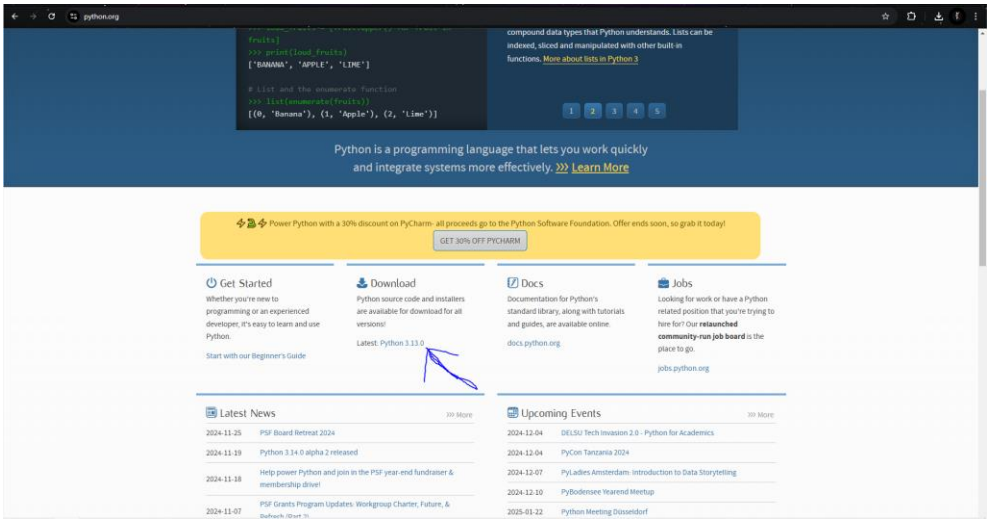
- Set up in an environment where internet connection is strong.

- The environment must be workable (free space in case there is maintenance needed)
- Only people with the know-how on how to use the system must work close to the environment

2. Installing Necessary Software

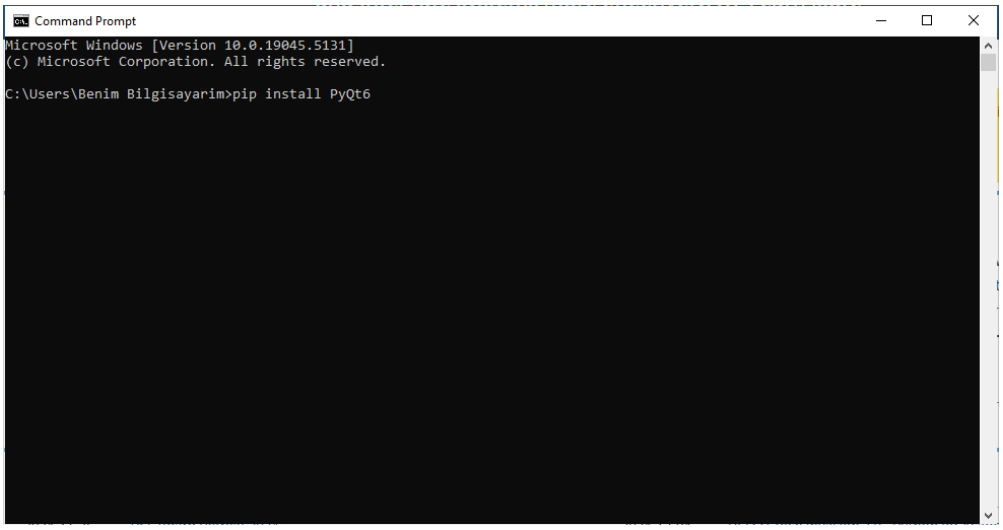
Install Python:

- Download and install Python from python.org.
- Add Python to the system PATH during installation.



Install PyQt6:

- Open a terminal or command prompt and run: pip install PyQt6

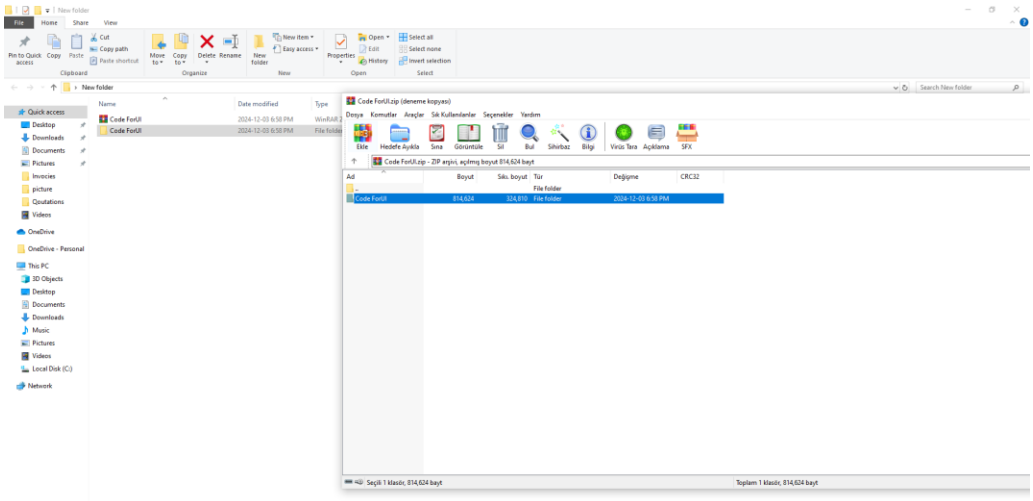


Install Additional Libraries (if needed): pip install -r requirements.txt

From the project folder, run:

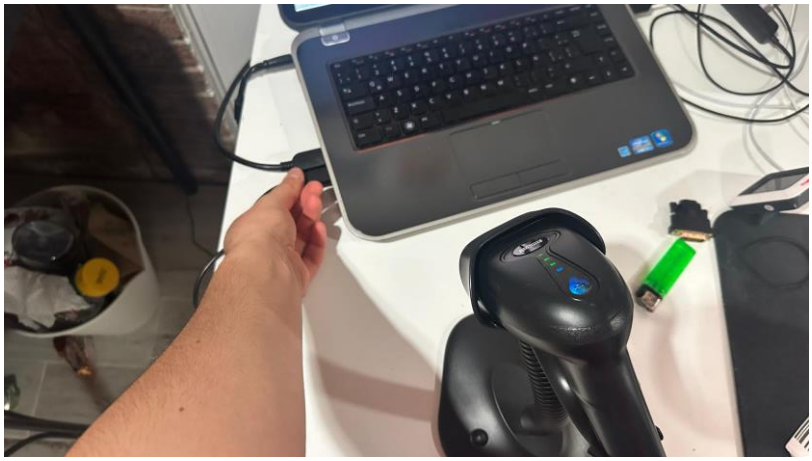
3. Preparing the Files

- Extract the contents of the Code ForUI folder to a dedicated project directory on your computer.
- Open the Launcher.py script in your code editor to ensure it points to the correct MainWindow class (currently in test4.py).



4. Configuring the Barcode Scanner

- Connect the USB Wireless Barcode Scanner to your computer.
- Test its functionality by scanning a sample barcode into any text editor to ensure proper input.
- Running the System



Run Launcher.py to launch the GUI:

- Use the barcode scanner to scan asset serial numbers. The system should display asset details and availability status.

7. Labeling the Assets

- Use a barcode label generator to print labels for each asset.
- Attach the labels to assets in easily accessible locations.

8. Testing the Prototype

- Scan a barcode to check the system's response.
- Verify that all asset data, including images and metadata, are displayed correctly in the GUI.
- Included Files for Reference

Python Scripts: For launching and managing the system.

Images: Asset images for demonstration.

Database Text Files: Assets.txt, Users.txt and __pycache__ file for preloaded data.

6.3Asset add and remove

6.3.1 BOM

Items	Specifications	Quantity	Price ($)	Link

Spyder Python IDE		1	0	https://www.spyder-ide.org/
Qt Designer		1	0	
Computer	Minimum: 8 Gb RAM Preferred: 16 Gb RAM	1		

6.3.2 Equipment List

No additional equipment is required.

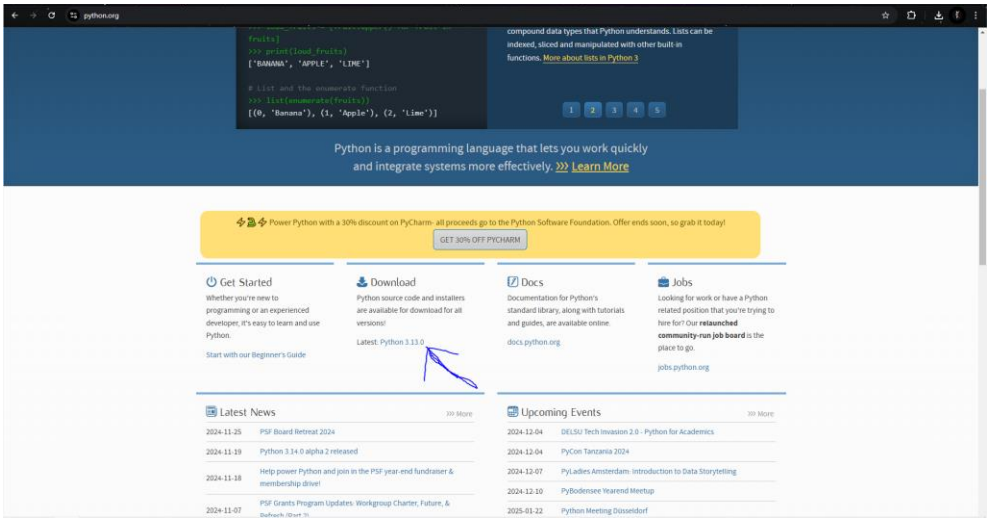
6.3.3 Instructions

1. Setting Up the Environment

2. Installing Necessary Software

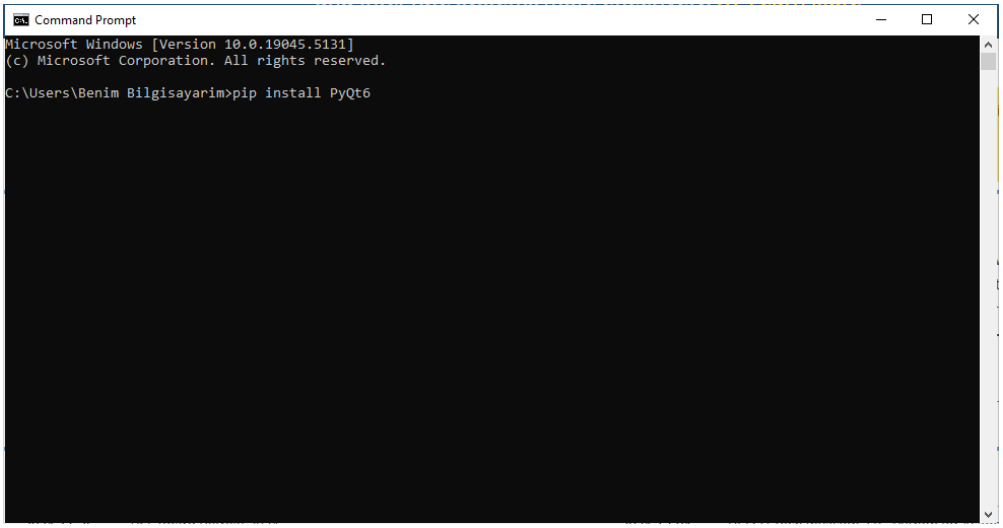
Install Python:

- Download and install Python from python.org.
- Add Python to the system PATH during installation.



Install PyQt6:

- Open a terminal or command prompt and run: pip install PyQt6

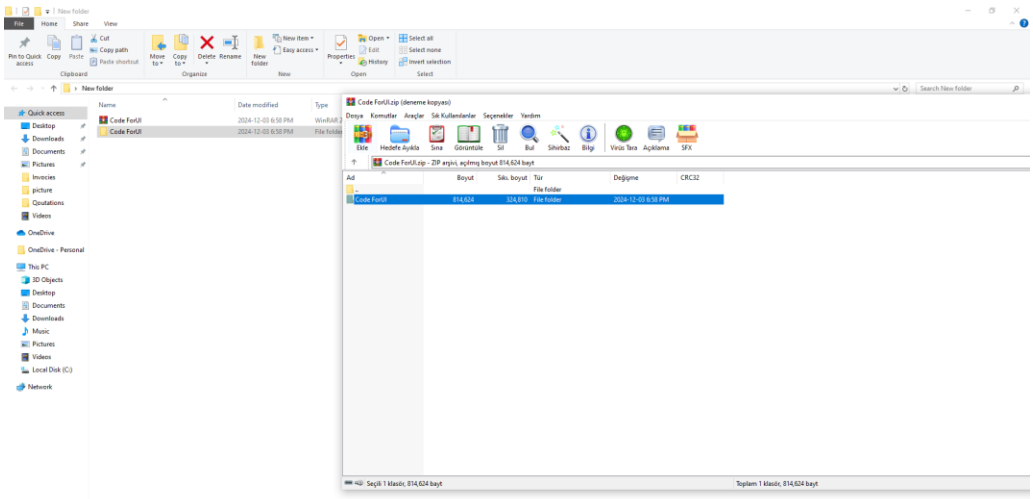


Install Additional Libraries (if needed): pip install -r requirements.txt

From the project folder, run:

3. Preparing the Files

- Extract the contents of the Code ForUI folder to a dedicated project directory on your computer.
- Open the Launcher.py script in your code editor to ensure it points to the correct MainWindow class (currently in test4.py).



Run Launcher.py to launch the GUI:

- Use the barcode scanner to scan asset serial numbers. The system should display asset details and availability status.

4. Testing the Prototype

- Verify that all asset data, including images and metadata, are displayed correctly in the GUI.
- Included Files for Reference

Python Scripts: For launching and managing the system.

Images: Asset images for demonstration.

Database Text Files: Assets.txt, Users.txt and __pycache__ file for preloaded data.

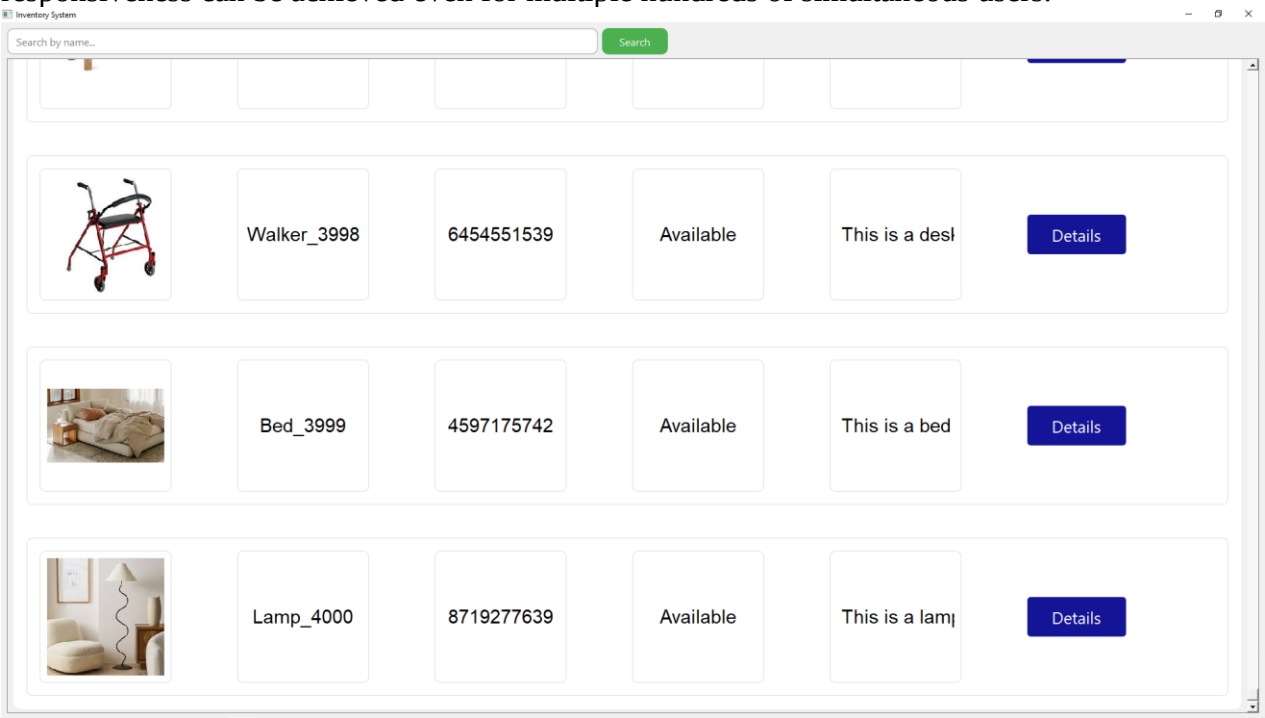
7 Testing & Validation

Table 1. Prototype testing plan

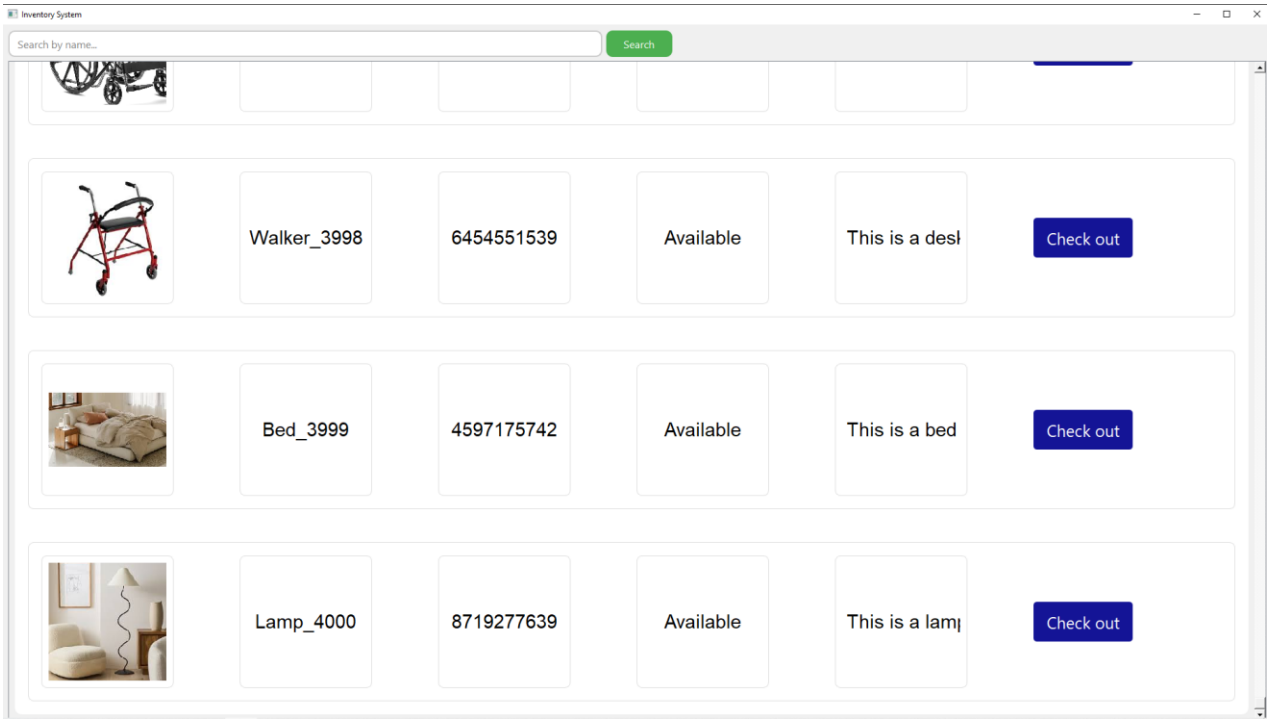
Test Number	Evaluation Criteria/Determine Measurables	Metrics	Test Description	Analysis Method	Results	Interpretation	Notes
1	Asset Search Feature (should less then 20 seconds)	Time: Seconds	How fast can the UI can search for assets	Search for different assets within a system containing 4000 assets	10 seconds	Pass	
2	UI Simplicity	Peoples approval	How easy it is for users to understand the UI	Asking peoples opinion on the UI	38 out of 40 people found it simple to use	Pass	
3	Check in and out speed (should be less then 20 seconds)	Time: Seconds	How fast can the user check in or out an asset	Checking out an asset within a system containing 4000 assets	10 seconds	Pass	
4	Affordiblity (should be less then 8000$ a year)	Money $	To see yearly cost of runnign the system	Calculate yearly cost of the system	5000$	Pass	
5	Storing large amounts of assets (should be at least 2500 assets)	Asset number	To see how many assets can the system handle	Add assets to the system until it becomes unstable	8000	Pass	
6	Add and remove assets		If assets can be added or removed from the system	Add and remove assets 10 times	10-Oct	Pass	
7	Display asset information		Does the UI display information about the assets	Check 100 asset infromation if they are correct	100/100	Pass	
8	Barcode Scanner Accuarcy	Number of correct scans	To see if the barcode scanner scans correct code	Scan 100 barcodes and check if it is displayed correctly	70/100	Fail	Upgrading the scanner and the label printer will improve the score on this topic

To test the database's capacity, we evaluated the system using 4,000 unique assets, a number significantly higher than the client's initial requirements, to assess its performance under a heavy load. The search UI consistently opened within 5 seconds, meeting the acceptable performance threshold for our use case. Similarly, the search functionality executed within 10 seconds, aligning with our performance criteria. The details feature operated efficiently, displaying accurate information within approximately 3 seconds, which is well within the acceptable range. Overall, the search UI performed as expected. With more advanced computing resources, this level of

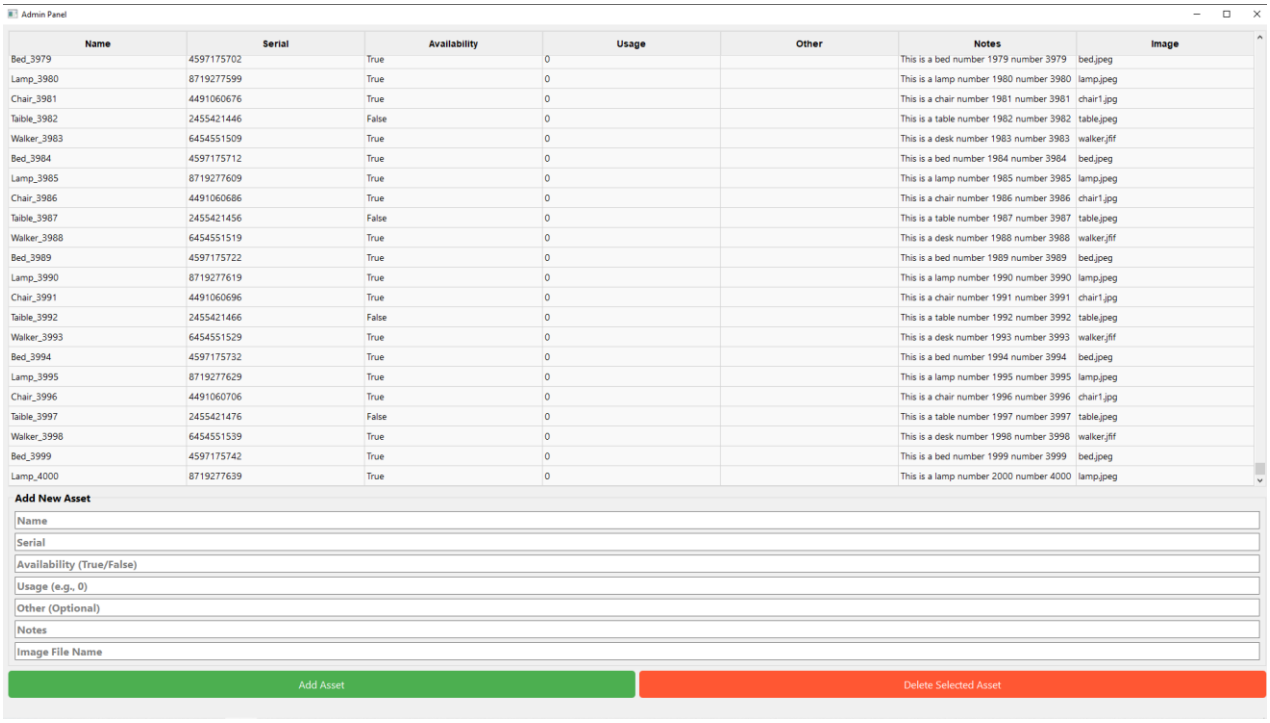
responsiveness can be achieved even for multiple hundreds of simultaneous users.



The check-in and check-out UI demonstrated performance consistent with the search UI, as both interfaces rely on nearly identical backend code and thus exhibit similar response times. Testing of the scanning feature on the prototype confirmed that whenever a barcode was successfully scanned, the correct asset information was displayed in the UI. However, occasional scanning errors were observed, attributable to the quality of the scanner and label printer. Improved hardware quality would likely reduce such malfunctions. Additionally, pressing the check-in or check-out button on the sub-UI consistently updated the asset availability status as expected.



The admin UI performs exceptionally well in handling 4,000 assets, even more efficiently than the other two UIs. This superior performance can be attributed to its simpler design and interface. The admin UI opens in less than 2 seconds, significantly exceeding the acceptable performance threshold. Adding an asset to the system, after entering the required information, is completed in under 1 second. Similarly, removing an asset is executed within the same time frame, ensuring fast and efficient operations.


Admin Panel
Name
Serial
Availability
Usage
Other
Notes
Image
Bed_3979 4597175702 True 0 This is a bed number 1979 number 3979 bed.jpeg
Lamp_3980 8719277599 True 0 This is a lamp number 1980 number 3980 lamp.jpeg
Chair_3981 4491060676 True 0 This is a chair number 1981 number 3981 chair1.jpg
Taible_3982 2455421446 False 0 This is a table number 1982 number 3982 table.jpeg
Walker_3983 6454551509 True 0 This is a desk number 1983 number 3983 walker.jfif
Bed_3984 4597175712 True 0 This is a bed number 1984 number 3984 bed.jpeg
Lamp_3985 8719277609 True 0 This is a lamp number 1985 number 3985 lamp.jpeg
Chair_3986 4491060686 True 0 This is a chair number 1986 number 3986 chair1.jpg
Taible_3987 2455421456 False 0 This is a table number 1987 number 3987 table.jpeg
Walker_3988 6454551519 True 0 This is a desk number 1988 number 3988 walker.jfif
Bed_3989 4597175722 True 0 This is a bed number 1989 number 3989 bed.jpeg
Lamp_3990 8719277619 True 0 This is a lamp number 1990 number 3990 lamp.jpeg
Chair_3991 4491060696 True 0 This is a chair number 1991 number 3991 chair1.jpg
Taible_3992 2455421466 False 0 This is a table number 1992 number 3992 table.jpeg
Walker_3993 6454551529 True 0 This is a desk number 1993 number 3993 walker.jfif
Bed_3994 4597175732 True 0 This is a bed number 1994 number 3994 bed.jpeg
Lamp_3995 8719277629 True 0 This is a lamp number 1995 number 3995 lamp.jpeg
Chair_3996 4491060706 True 0 This is a chair number 1996 number 3996 chair1.jpg
Taible_3997 2455421476 False 0 This is a table number 1997 number 3997 table.jpeg
Walker_3998 6454551539 True 0 This is a desk number 1998 number 3998 walker.jfif
Bed_3999 4597175742 True 0 This is a bed number 1999 number 3999 bed.jpeg
Lamp_4000 8719277639 True 0 This is a lamp number 2000 number 4000 lamp.jpeg
Add New Asset
Name
Serial
Availability (True/False)
Usage (e.g., 0)
Other (Optional)
Notes
Image File Name
Add Asset
Delete Selected Asset

8 Conclusions and Recommendations for Future Work

During the course of the term our group has learnt a lot of lessons that would have been greatly productive at the start of the term. Our group always knew that organization was key however we never truly knew to what extent. At the start of the term, we only planned 1 week in advance which led to a lot of disorganisations especially when mid-terms came around. If I were to recommend one key thing to another group It would be look plan further ahead, so you know what's coming up and have more time to react instead of facing the deadlines a few days before they're due. The second lesson we learnt was that using people's talents will greatly improve the quality of work. Durring the first few weeks of developing the prototype we quickly learnt that some people were much better at coding than others later we created a main development team based on the more talented coders while the other people were supported into a side coding team and mainly worked on the completion of the deliverables. Organizing a team is key to success and doing so sooner rather than later will greatly put future teams ahead of us.

If our team had a few more months to develop the project we would like to focus the attention on the UI of the system. Due to the lack of coding expectance we had to shift a large majority of our time on the basic functions in the system instead of towards the UI. In our conceptual designs one idea was to incorporate customizable UI so each staff member could have their personalized interface. This idea was meant to increase the user satisfaction and diversity the UI towards as many users as possible. The customizable UI would incorporate; customizable colors,

customizable sizes, an option to pin assets, an option to move the page setup and a customizable fast icon. These features would fall in the design for usability DFX and was part of our original design. The second feature we would have liked to add towards the UI would be to incorporate Nielsen's 10 heuristics for user interfaces; these 10 codes aim to maximize user satisfaction which is the mail goal for the design of the project and ties diffectly to our problem statement. Again, due to the lack of coding experience this idea was not further expanded on however would be something essential to implement if we were to develop our product for the market.

9 Bibliography

https://openai.com/
Lakhani, R. (2024). *Lecture notes.* GNG 2101: Introduction to Engineering Management. University of Ottawa. Brightspace.
Python Software Foundation. (2001). *Python*. Welcome to Python.org. Retrieved September 3, 2024, from https://www.python.org/

Python Software Foundation. (2024). *PyQt6 · PyPI*. PyPI. Retrieved October 3, 2024, from https://pypi.org/project/PyQt6/

Spyder Website Contributors. (2024). *Spyder*. Spyder | The Python IDE that scientists and data analysts deserve. Retrieved October 3, 2024, from https://www.spyder-ide.org/

APPENDICES

APPENDIX I: Design Files

This document serves as the capstone of a series of project deliverables and a code project. The deliverables (A-E) have been consolidated into 2 documents, while the Python code scripts have been consolidated into a single zip file.

Table 3. Referenced Documents

Document Name	Document Location and/or URL	Issuance Date
CodeForUI.zip	https://makerepo.com/takudzwa29/2230.gng-2101-the-matrix-asset-inventory-system	12-03-2024
PD A-D.docx	https://makerepo.com/takudzwa29/2230.gng-2101-the-matrix-asset-inventory-system	10-06-2024
PD E-H	https://makerepo.com/takudzwa29/2230.gng-2101-the-matrix-asset-inventory-system	11-28-2024