ID Scanner Design Document

Clarkson University CS350

Spring 2025

Delara, Hunter, Samantha, Soliat, Nick

Contents

Project Overview & Objective:

Project Overview:

The ID Scanner Project aims to provide a fast, and easy scan-in/out system for the MakerSpace, replacing the existing inefficient Google Form with a simple hardware solution. Currently, visitors are required to fill out a Google Form that includes the machines that they intend to use during their visit. Additionally, if a student has not been to the MakerSpace before, they are also required to fill out a short waiver in the form of another Google Form. With our system, students will be able to simply scan their ID cards at an RFID scanner. If they are new, the screen will prompt them to fill out a waiver, which will then be stored in a database that will remember they have completed it. Otherwise, scanning alone is sufficient for entry. On the way out, students will scan their ID again, and the screen will prompt them to select what machines they used during their visit. This will be achieved through a custom interface rather than a Google Form to improve simplicity, and will also capture more accurate usage data, as some students may change their mind about which machines they want to use after signing in.

Objective:

- Improve the Makerspace scan-in process by using an RFID scanner included with a touchscreen interface.
- Improve the accuracy of machine usage records.
- Enhance user experience by integrating waiver submission directly into the system.
- Provide data export and analytics

Identification of Modules

This project consists of four modules.

I. **Hardware Interface Module**
   A. **Function**: Communicate directly with the RFID scanner and touchscreen
   B. **Responsibility**: Detect RFID scans and pass scanned ID information to the next module (UI or backend API)

II. **User Interface (Front-end) Module**

A. **Function**: Display a simple form with checkboxes for equipment selection and additional prompts for new users

B. **Responsibility**:

1. For new users: Prompt to complete waiver
2. For returning users: Confirm entry
3. Scan-out: Prompt for machine selection

III. **Back-end & Database Module**

A. **Function**: Store student & machine usage information

B. **Database Management System**: SQLite

C. **Data Handling**: Create an API endpoint that receives scan-in data and user details

D. **Export Data**: Implement a function that queries the database and formats the data as CSV

IV. **Analytics & Admin Dashboard**

A. **Function**: Provide visual insights

B. **Considerations**:

1. Secure admin access (authentication & authorization) to view statistics
2. Allow admins to generate reports and monitor real-time activity (if feasible)

Module Interaction (Integration Strategy)

I. **Hardware Event**

A. When an RFID card is scanned, the RFID module sends the data to the Raspberry Pi

II. **Software Trigger**

A. A Python script running on the Pi reads the card’s ID

B. The system launches the appropriate UI depending on if the student is new or returning

III. Data Collection

A. The UI gathers inputs and sends the data to the back-end via an API call

IV. Data Storage

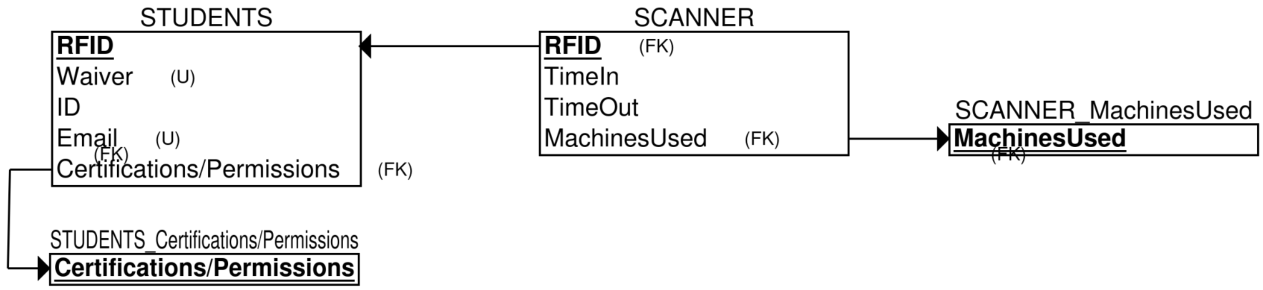
A. The back-end module writes the data to the database

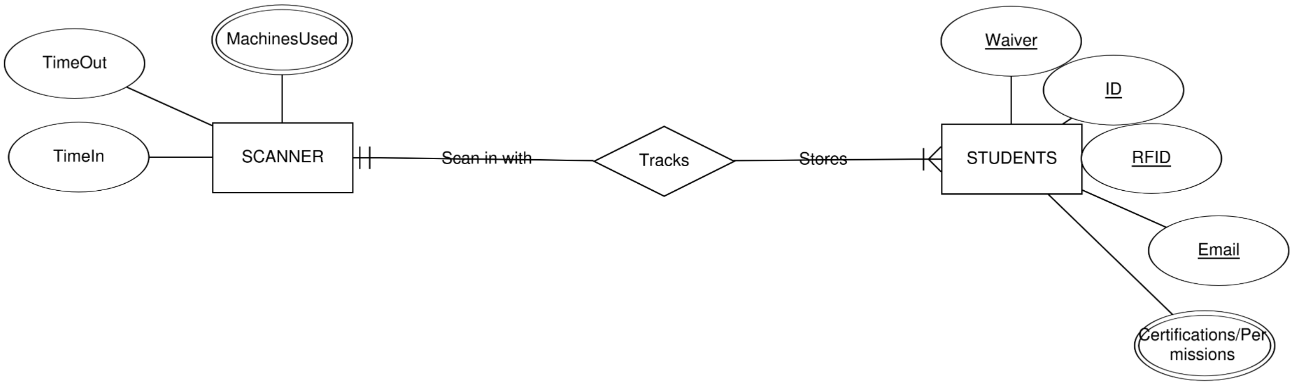
V. Data Export & Analytics

A. An export function allows CSV generation from the stored data

B. Optionally, an analytics dashboard can be launched for real-time monitoring

Database Schema



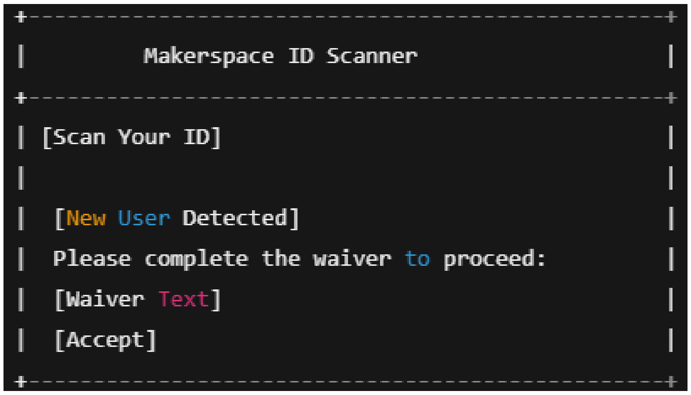


UI & Interaction Diagrams

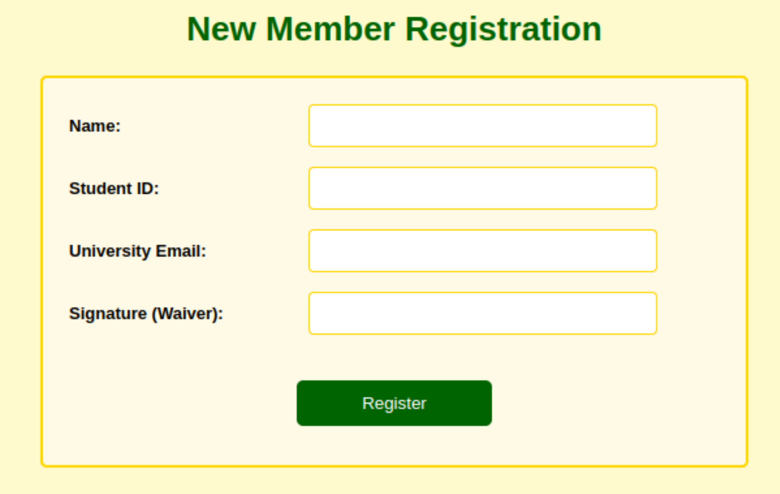
User Screens

New User Example:

1. Scan ID → "Sign Waiver" form (name, email, checkbox)
2. Register → Access granted.



```
+--------------------------------------------------+
|         Makerspace ID Scanner                    |
+--------------------------------------------------+
| [Scan Your ID]                                   |
|                                                  |
|  [New User Detected]                             |
|  Please complete the waiver to proceed:          |
|  [Waiver Text]                                   |
|  [Accept]                                        |
+--------------------------------------------------+
```



New Member Registration

Name:

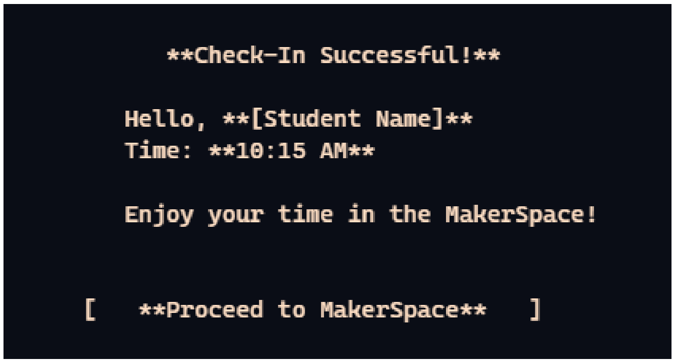
Student ID:

University Email:

Signature (Waiver):

Returning User Example:

1. Scan ID → "Welcome back!" → Log entry time.
2. Scan-out → Select machines → Submit.

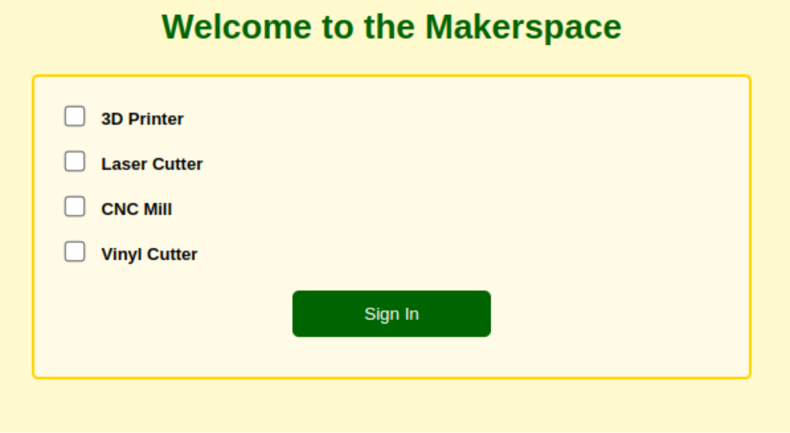


```
        **Check-In Successful!**

     Hello, **[Student Name]**
     Time: **10:15 AM**

     Enjoy your time in the MakerSpace!


  [   **Proceed to MakerSpace**    ]
```



Welcome to the Makerspace

☐ 3D Printer

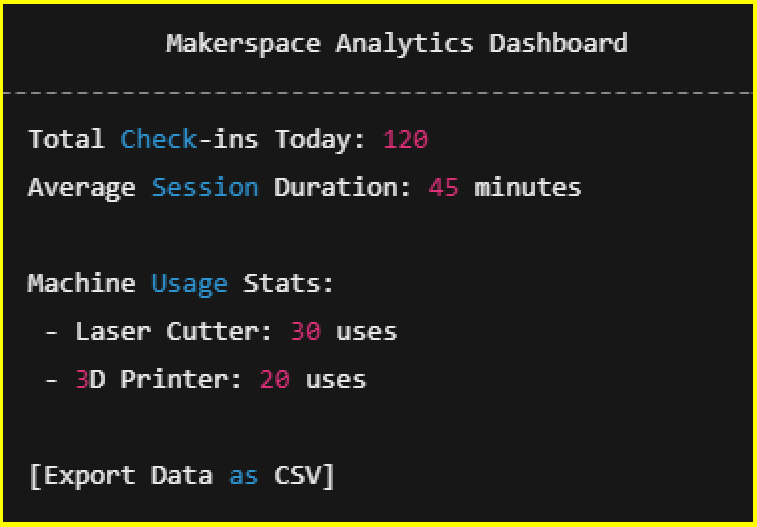
☐ Laser Cutter

☐ CNC Mill

☐ Vinyl Cutter

Admin Screens

Analytics Dashboard (Example)



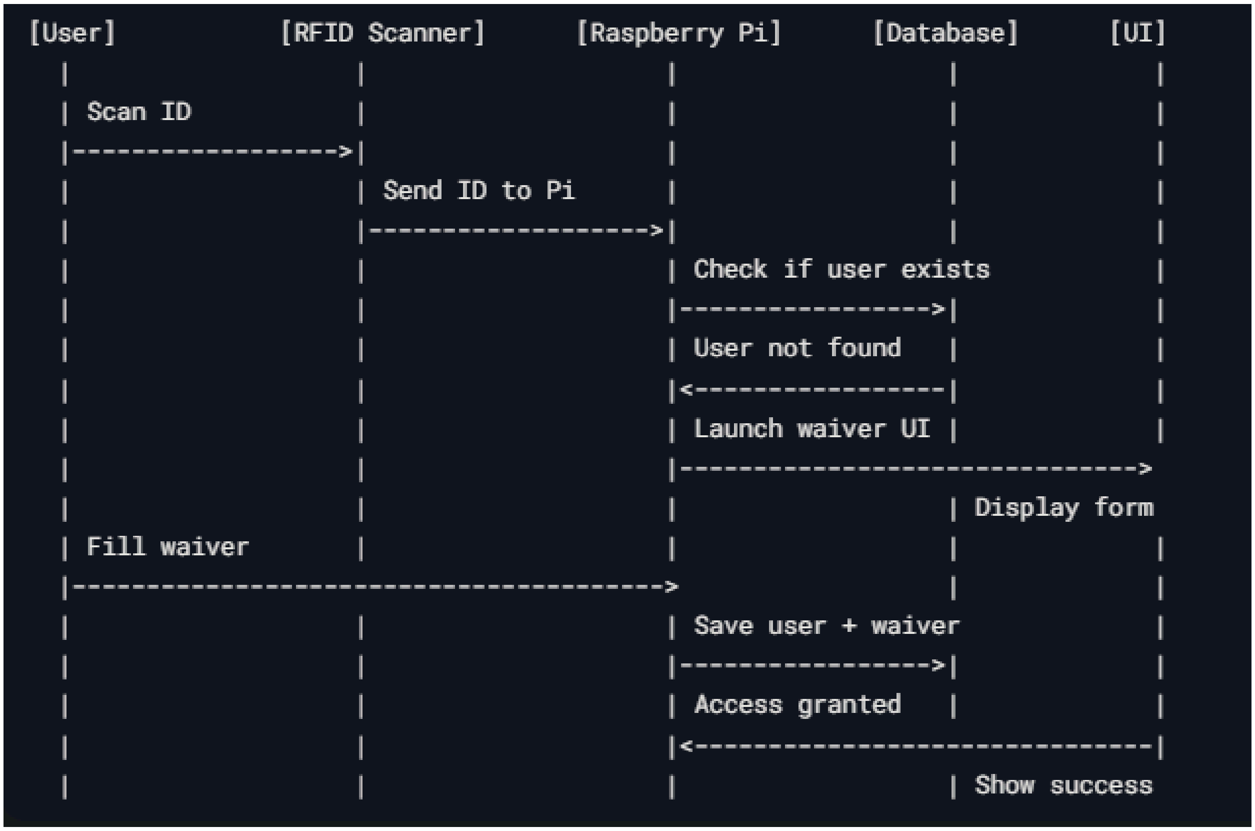
```
          Makerspace Analytics Dashboard
-------------------------------------------------
 Total Check-ins Today: 120
 Average Session Duration: 45 minutes

 Machine Usage Stats:
  - Laser Cutter: 30 uses
  - 3D Printer: 20 uses

 [Export Data as CSV]
```

Sequence Diagrams

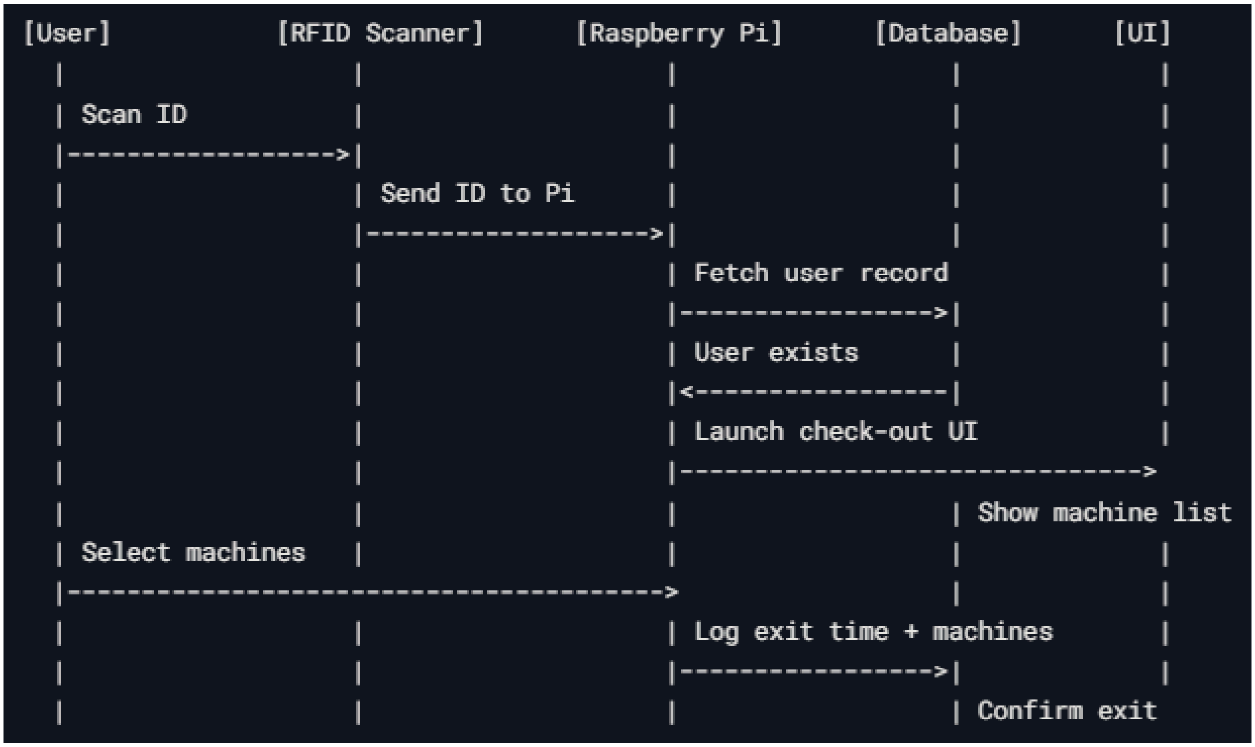
New User:



```
[User]           [RFID Scanner]      [Raspberry Pi]      [Database]      [UI]
  |                   |                    |                  |             |
  | Scan ID           |                    |                  |             |
  |------------------>|                    |                  |             |
  |                   | Send ID to Pi      |                  |             |
  |                   |------------------->|                  |             |
  |                   |                    | Check if user exists           |
  |                   |                    |----------------->|             |
  |                   |                    | User not found   |             |
  |                   |                    |<-----------------|             |
  |                   |                    | Launch waiver UI |             |
  |                   |                    |-------------------------------->
  |                   |                    |                  | Display form
  | Fill waiver       |                    |                  |             |
  |---------------------------------------->                  |             |
  |                   |                    | Save user + waiver             |
  |                   |                    |----------------->|             |
  |                   |                    | Access granted   |             |
  |                   |                    |<-------------------------------|
  |                   |                    |                  | Show success
```

Returning User:

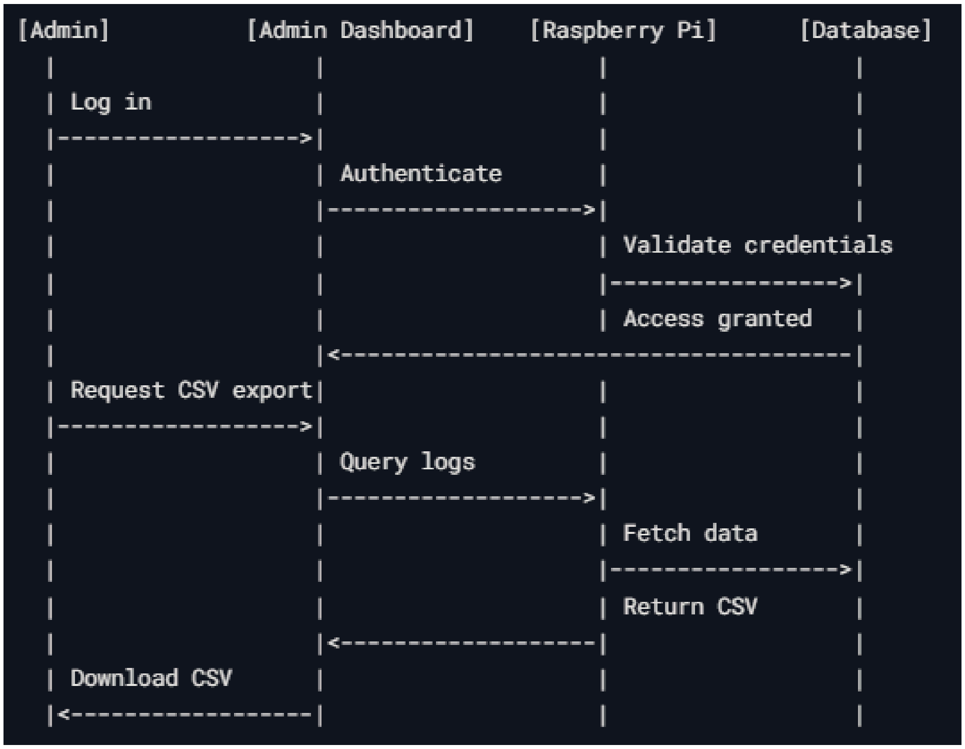
(Show interactions when scanning out as there is not much interaction when scanning in)



```
[User]           [RFID Scanner]      [Raspberry Pi]      [Database]      [UI]
  |                   |                    |                  |             |
  | Scan ID           |                    |                  |             |
  |------------------>|                    |                  |             |
  |                   | Send ID to Pi      |                  |             |
  |                   |------------------->|                  |             |
  |                   |                    | Fetch user record              |
  |                   |                    |----------------->|             |
  |                   |                    | User exists      |             |
  |                   |                    |<-----------------|             |
  |                   |                    | Launch check-out UI            |
  |                   |                    |------------------------------->
  |                   |                    |                  | Show machine list
  | Select machines   |                    |                  |             |
  |---------------------------------------->                  |             |
  |                   |                    | Log exit time + machines       |
  |                   |                    |----------------->|             |
  |                   |                    |                  | Confirm exit
```

Admin:

(Shows interactions when an admin exports data)



```
[Admin]          [Admin Dashboard]    [Raspberry Pi]      [Database]
  |                   |                    |                  |
  | Log in            |                    |                  |
  |------------------>|                    |                  |
  |                   | Authenticate       |                  |
  |                   |------------------->|                  |
  |                   |                    | Validate credentials
  |                   |                    |----------------->|
  |                   |                    | Access granted   |
  |                   |<--------------------------------------|
  | Request CSV export|                    |                  |
  |------------------>|                    |                  |
  |                   | Query logs         |                  |
  |                   |------------------->|                  |
  |                   |                    | Fetch data       |
  |                   |                    |----------------->|
  |                   |                    | Return CSV       |
  |                   |<-------------------|                  |
  | Download CSV      |                    |                  |
  |<------------------|                    |                  |
```

Identification of Technologies

I. **Programming Languages**
   A. Python for hardware interaction, API, and backend logic
   B. HTML/CSS for frontend

II. **Frameworks**
   A. Flask for backend API and web serving
   B. Optional: Bootstrap for enhanced UI design

III. **Database**
   A. SQLite

IV. **Hardware & Library**
   A. MFRC522 Python library for interfacing with RFID reader
   B. Raspberry Pi

V. **Additional Tools**
   A. GitHub for version control and collaboration

Security & Access Control

For the web-based user UI, security will be implemented using a multi-layered approach. We will combine RFID-based authentication with role-based access control (RBAC) to ensure that only authorized users can check in and out or manage data. Regular users scan their ID to check in/out, with new users prompted to complete a waiver before access. Admins have additional permissions to manage data. Secure communication via HTTPS, input validation, and session timeouts prevent unauthorized access. Data is encrypted, and audit logs track all activity for security monitoring. It is best to stay away from admin UI because it would be messier and make it less secure to be able to access the data, outside people could SSH into the Pi and copy the data onto a random computer.

Connections to Other Systems

If time permits, we plan to develop an additional front-end webpage that connects to our database to display live Makerspace usage data. This feature would include a simple “crowd meter” built with HTML/CSS, to display accurate live crowd activity. This is not our primary focus though, as we are prioritizing giving MakerSpace mentors a more efficient sign-in process.