Lab.cafe

Design

Benovicsová Lucia, Forgáč Adam, Mihál Lukáš, Vančo Michal

Content

1 Detailed specification of external interfaces

The machine, access, and member management system at **Lab.cafe** integrates multiple applications, devices, and file formats to ensure seamless functionality and efficient data management.

1.1 3D Printers

3D printers communicate with the Otello system through the HTTP REST API provided by Octoprint. Each printer is equipped with a Raspberry Pi Zero 2W running Octoprint, enabling print monitoring and retrieval of active job information. Otello periodically (every 30 seconds) queries the printers to fetch their status. Users must "claim" the print job within Otello within 5 minutes of starting a new print; otherwise, the print is automatically stopped. A centralized database logs all print jobs for monthly summaries and user billing purposes.

1.2 Database Usage

The system utilizes a **MySQL database** to store printer configurations, print job records, member data, and access rights. Additional data, such as consumption transactions, is synchronized with **Airtable**.

1.3 Membership and Machine Access

For membership and machine access management, the system integrates with **Fabman.io**, which uses REST API to verify user access to specific machines. When an RFID card is scanned at a Fabman Bridge device, Fabman checks user permissions and, if authorized, activates the machine for a specified duration.

1.4 Payment Processing

In processing payments at the bar, the system uses **PapayaPOS** in conjunction with the LabPOS terminal. A member scans their RFID card at the LabPOS terminal, which calls an endpoint in the Otello system to identify the active member. PapayaPOS then communicates with Otello via a simulated hotel API to retrieve active member data. Upon transaction confirmation, PapayaPOS sends the payment details back to Otello, where they are stored in **Airtable**.

1.5 Membership Management

Regarding memberships, the system integrates **Woocommerce** and **Fabman.io** via **Make.com**. Memberships purchased or renewed through Woocommerce are automatically synchronized with Fabman.io, ensuring members have access to machines and facilities.

1.6 User Interface

The user interface of the Otello system allows standard users to monitor printer statuses and manage print jobs. Administrators have access to an extended range of functions,

including adding new printers, members, and configurations. Device or member configurations can also be managed in bulk through JSON file uploads. Additionally, the system supports exporting and importing print job data in CSV format.

2 Detailed Data Model

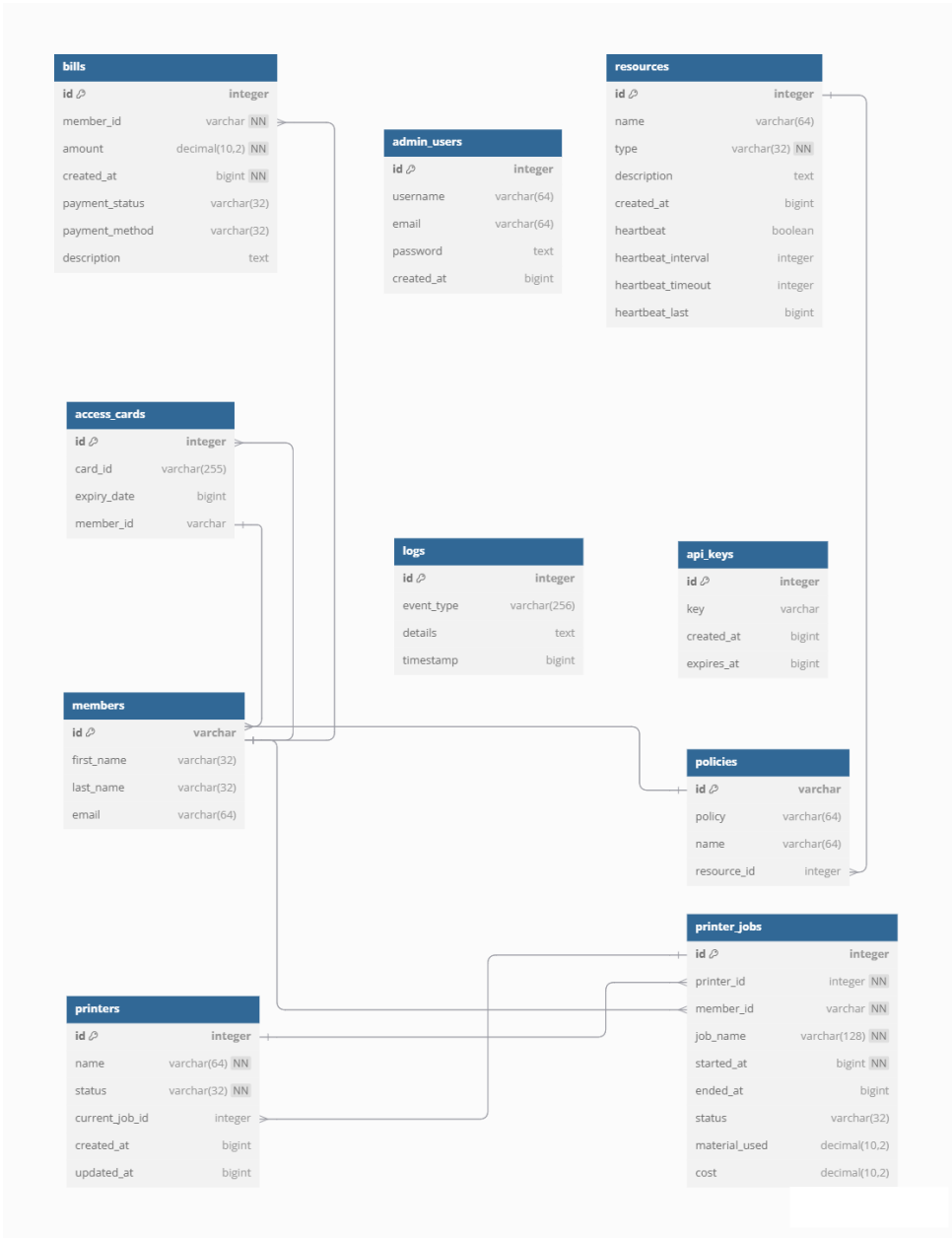
The data model represents the system's core entities and their relationships, focusing on managing members, access control, billing, and resource usage.

Access Cards store membership card details linked to **Members**, who are the primary users of the system. Members can have associated **Bills**, tracking charges for services like 3D printing. **Resources**, such as printers or doors, are controlled by **Policies**, which define access and usage rules.

Printers store the status of 3D printers and their current jobs, while **Printer Jobs** track individual tasks initiated by members, linking printers to specific users.

Admin Users manage the system and indirectly interact with entities such as bills or policies. **Logs** record system events, offering insights into activities like resource usage or errors. **API Keys** secure system endpoints, facilitating authenticated communication.

This model ensures modularity and scalability, with clear relationships between entities where necessary, while allowing flexibility for independent tables like `logs` and `api_keys`.



Img. 1: Illustration of the Database Diagram

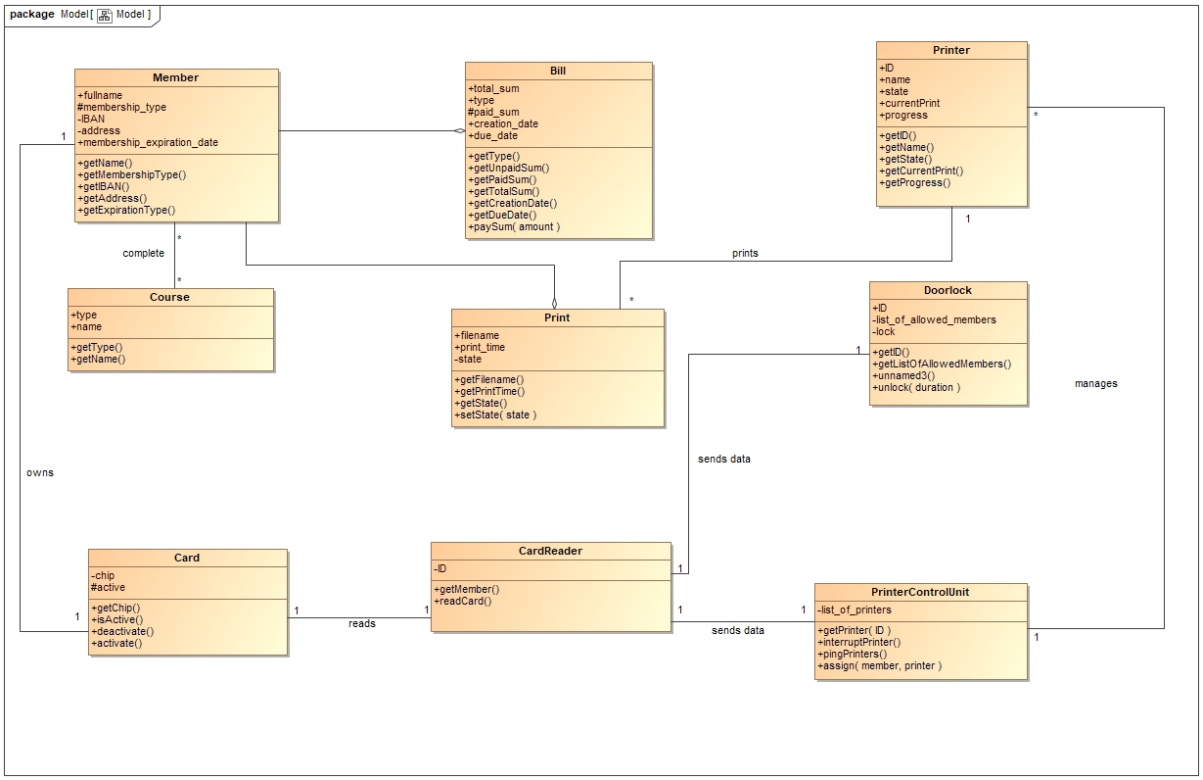
3 Implementation Design

3.1 Class Diagram

This UML class diagram represents a system for managing members, printing services, billing, and access control. The **Member** class stores user details, membership types, and expiration information. Members can enroll in **Courses**, receive **Bills** for services like printing, and access secure areas via **Cards**.

CardReaders read cards to verify members, while **Doorlocks** grant access based on permissions. The printing system includes **Print jobs**, which are managed by **Printers** under the control of a **PrinterControlUnit**, responsible for job assignment, printer monitoring, and interruptions. Printers process jobs and track their progress, and Bills handle payments for services used.

Overall, the system ensures secure access, organized printing workflows, and efficient member management for environments like labs, libraries, or co-working spaces.



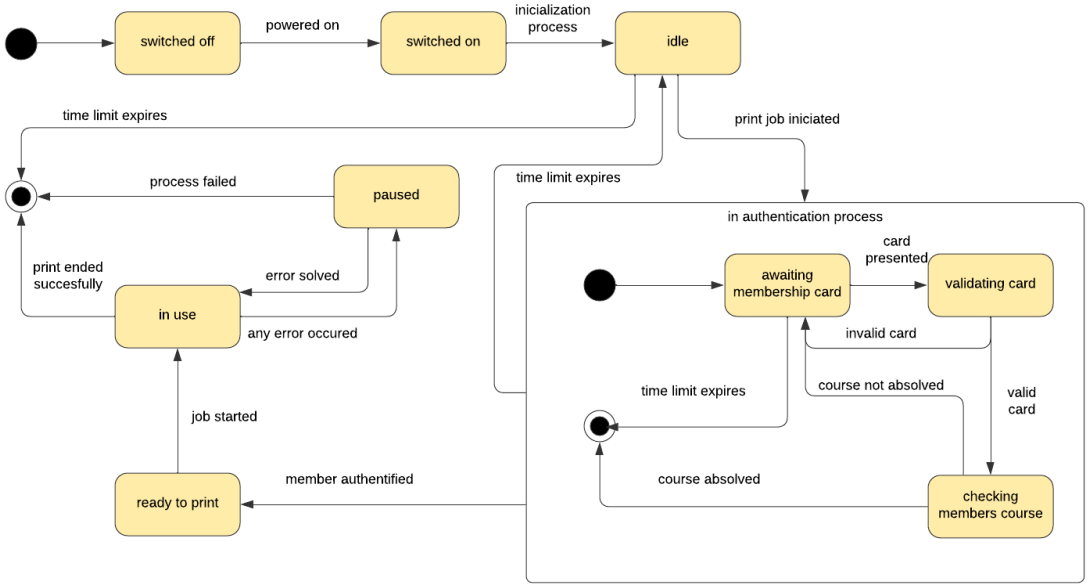
Img. 2: UML Class Diagram

3.2 State Diagram

This state diagram illustrates the lifecycle of a 3D printer, covering power states, user authentication, printing, and error handling.

The printer starts in the **switched off** state and transitions to **idle** after powering on and initializing. When a print job is initiated, it enters the **authentication process**, where the user must present a valid membership card. Upon validation and prerequisite checks, the printer moves to **ready to print** and then **in use** during the job.

If the print completes successfully, it returns to **idle**. Errors move it to a **paused** state until resolved, after which it resumes printing or returns to **idle**. If time limits expire or validation fails, the process ends in a terminal state, ensuring secure and efficient operation.



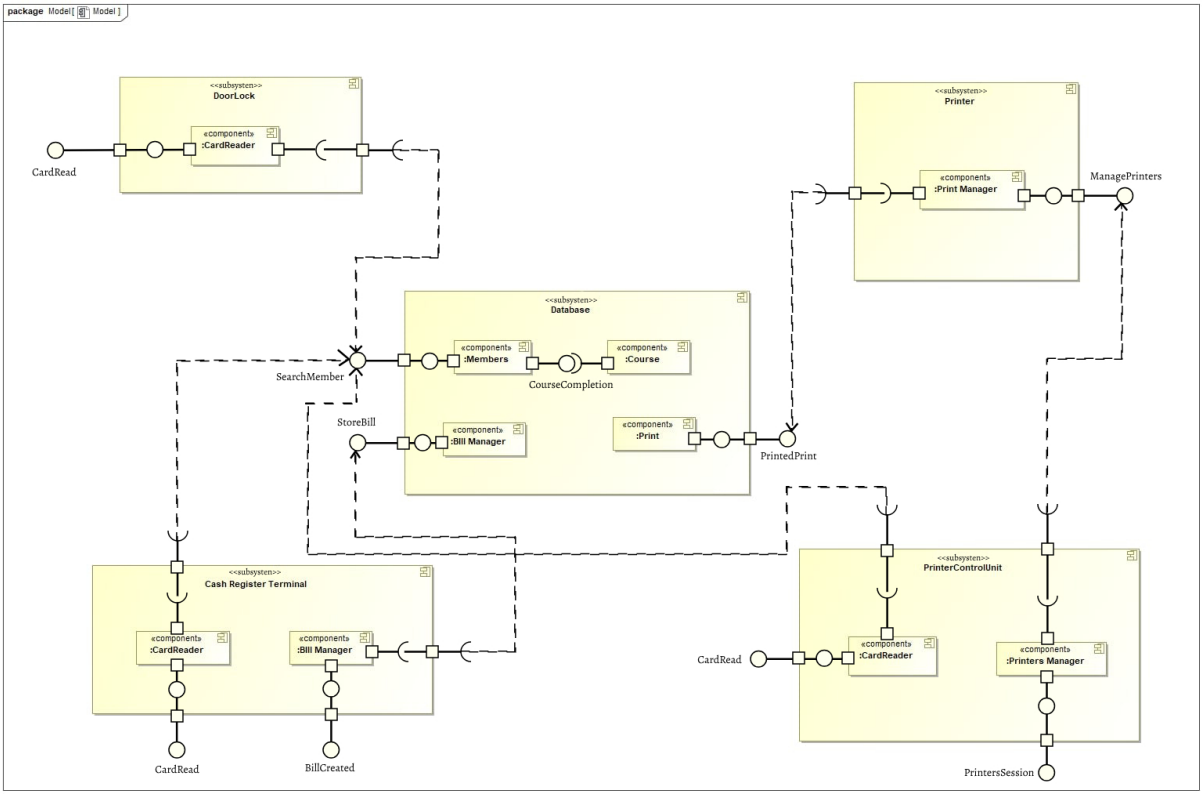
Img. 3: State Diagram of a 3D Printer

3.3 Component Diagram

This component diagram depicts the architecture of a system managing printing services, billing, and access control. The system includes several subsystems, such as **DoorLock**, **Database**, **Printer**, **Cash Register Terminal**, and **PrinterControlUnit**, each containing specific components.

The **DoorLock** subsystem uses a **CardReader** to authenticate user access, while the **Database** manages key components like **Members**, **Courses**, **Bills**, and **Print jobs**, ensuring user data, course completions, and billing are stored and retrieved. The **Printer** subsystem contains a **Print Manager** for handling printing operations. The **Cash Register Terminal** includes a **CardReader** and a **Bill Manager** for payment processing.

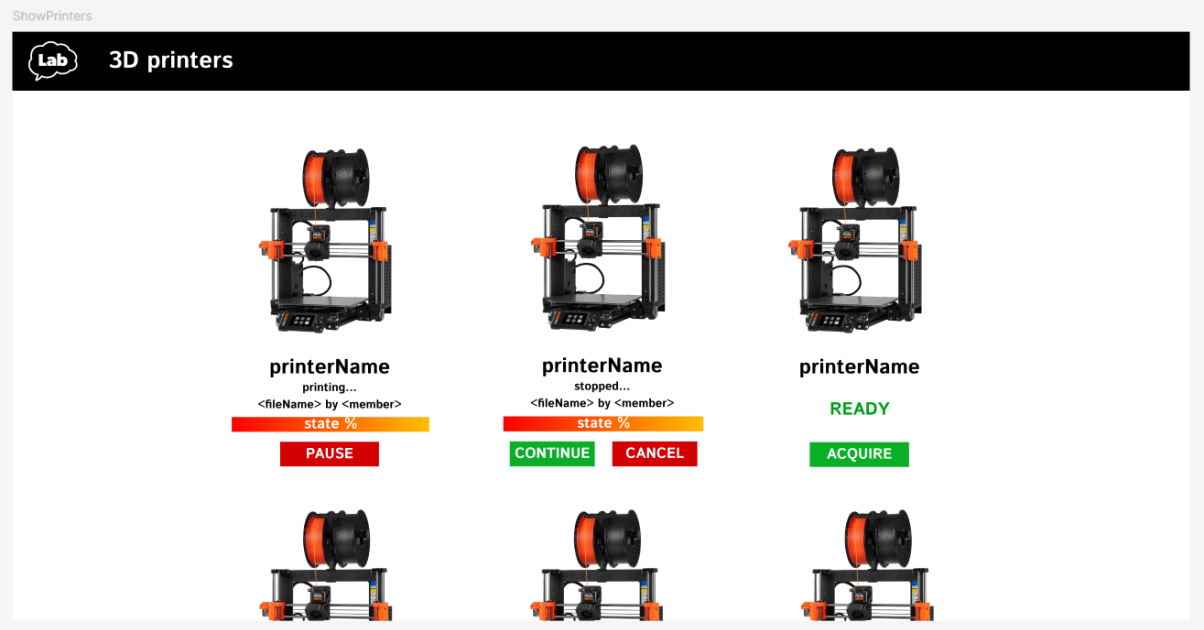
Finally, the **PrinterControlUnit** coordinates printers via its **Printers Manager** and handles user sessions through its **CardReader**. Interactions between subsystems are marked by dashed connectors, showing workflows like card reading, user validation, course completion checks, billing, and print job processing. This architecture ensures secure user access, efficient printing, and proper billing in an integrated environment.



Img. 4: Component Diagram of the System Architecture

4 User Interface Design

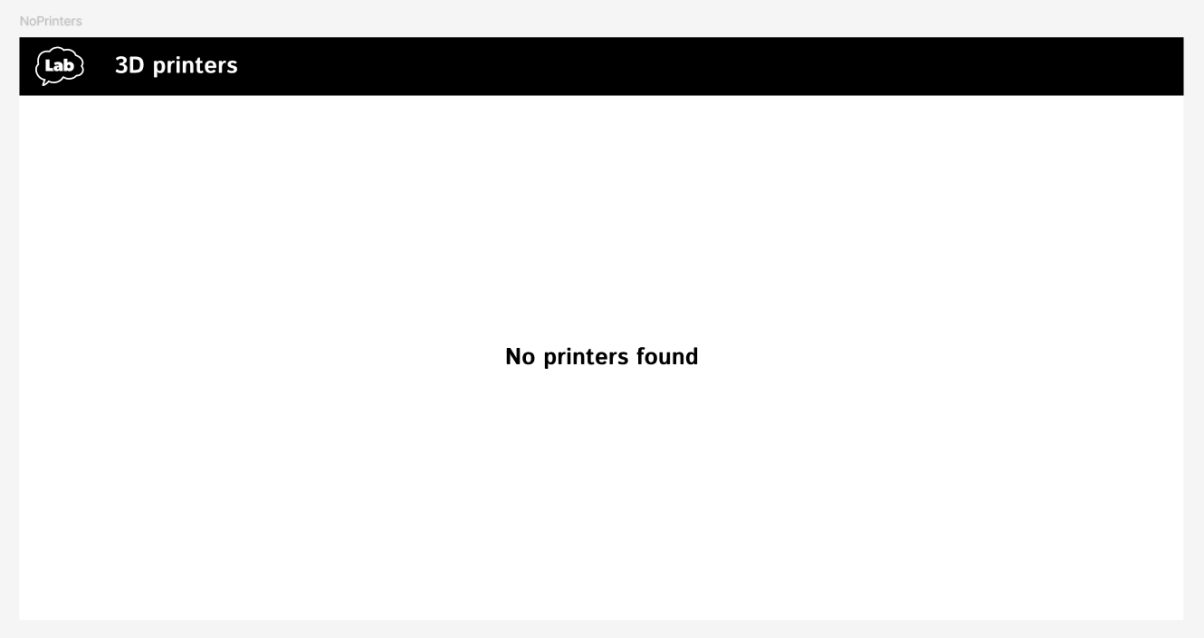
The graphical user interface (GUI) is designed specifically for managing 3D printers. This interface operates on a standalone microcomputer using PHP, providing an interactive display of all printers retrieved from the Otello system. It periodically fetches the latest printer statuses, ensuring real-time updates and evaluations of their condition.



Img. 5: Example of the Printer List Interface

4.1 Printer List Interface

The main interface displays a responsive list of printers, with the number of printers in a row dynamically adjusting based on the device's screen width. If no active printers are detected, the interface shows only a header accompanied by the message *"No printers found"*.



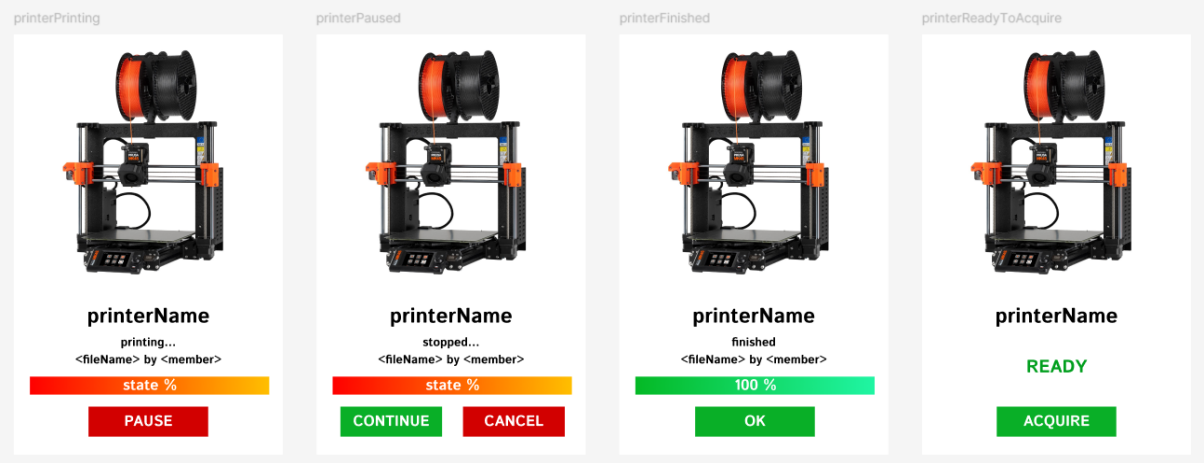
Img. 6: Illustration Showing No Printers Found

4.2 Printer Statuses

Each printer's status is clearly displayed, with possible states including:

- **Printing**: The printer is actively processing a job.
- **Paused**: The printer is temporarily stopped.
- **Finished**: The print job is complete and ready for collection.
- **Ready to Acquire**: The printer is available for a new job.

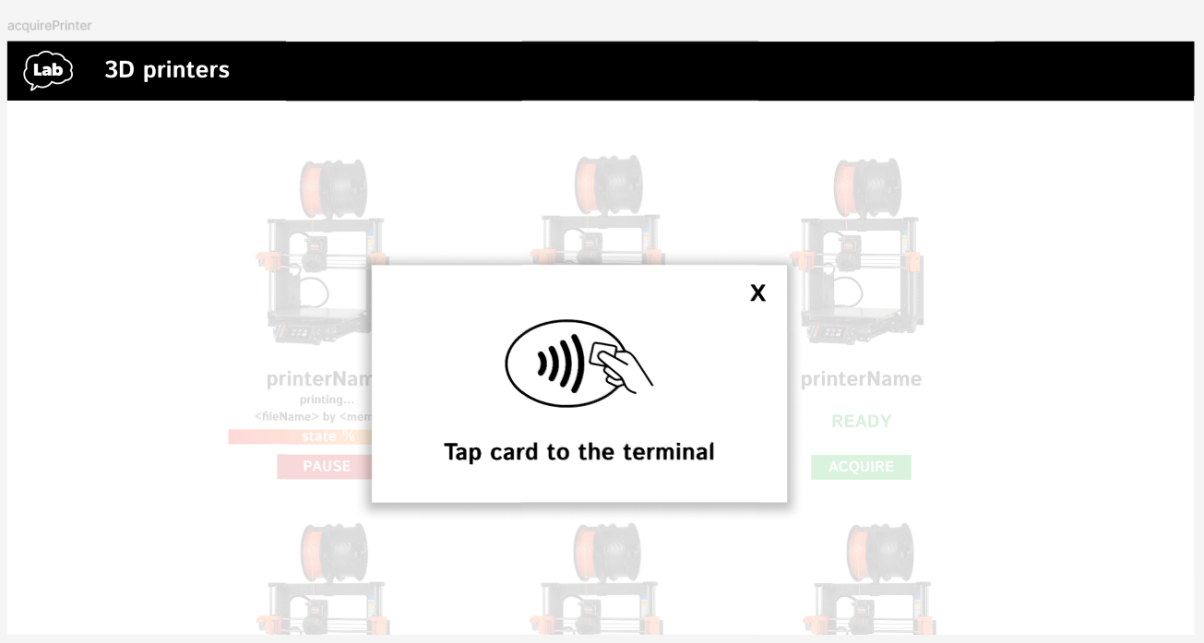
Figures illustrating these statuses are included for reference. The visual representation of each printer reflects its actual type and appearance, ensuring intuitive recognition by users.



Img. 7: Illustration of Printer States

4.3 Acquiring a Printer

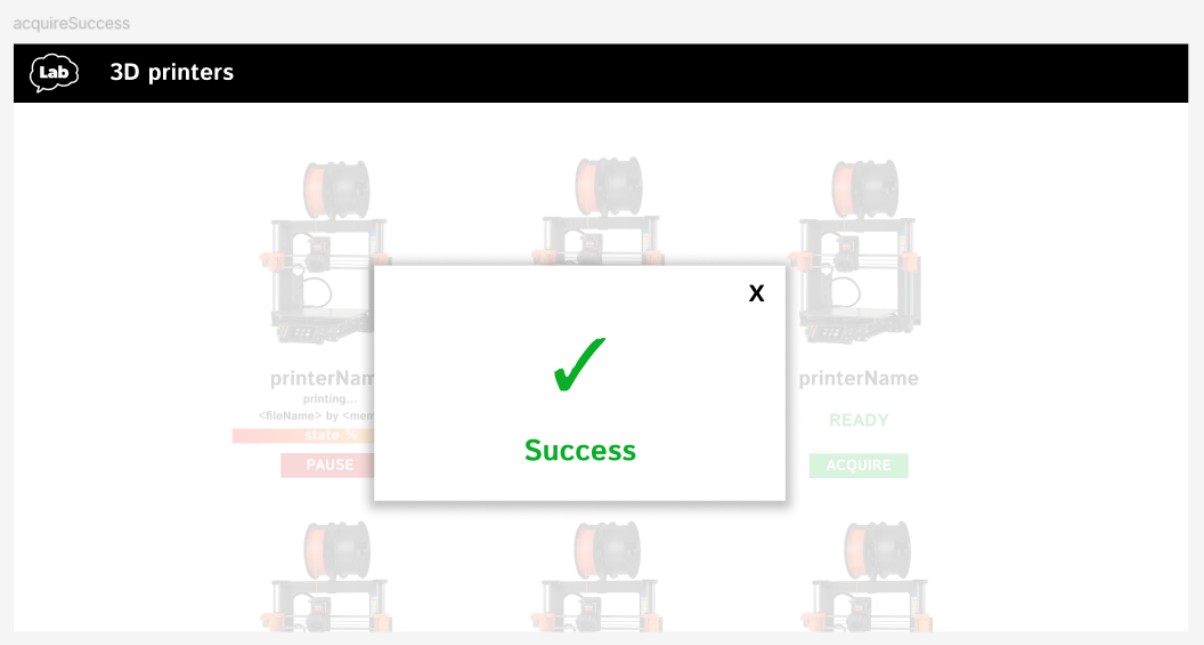
To acquire a printer, users interact with a corresponding button in the interface. Clicking this button prompts a dialog window instructing users to tap their membership card. The result of the card tap is evaluated as either a *success* or *failure*, after which the printer's status is updated accordingly in the interface.



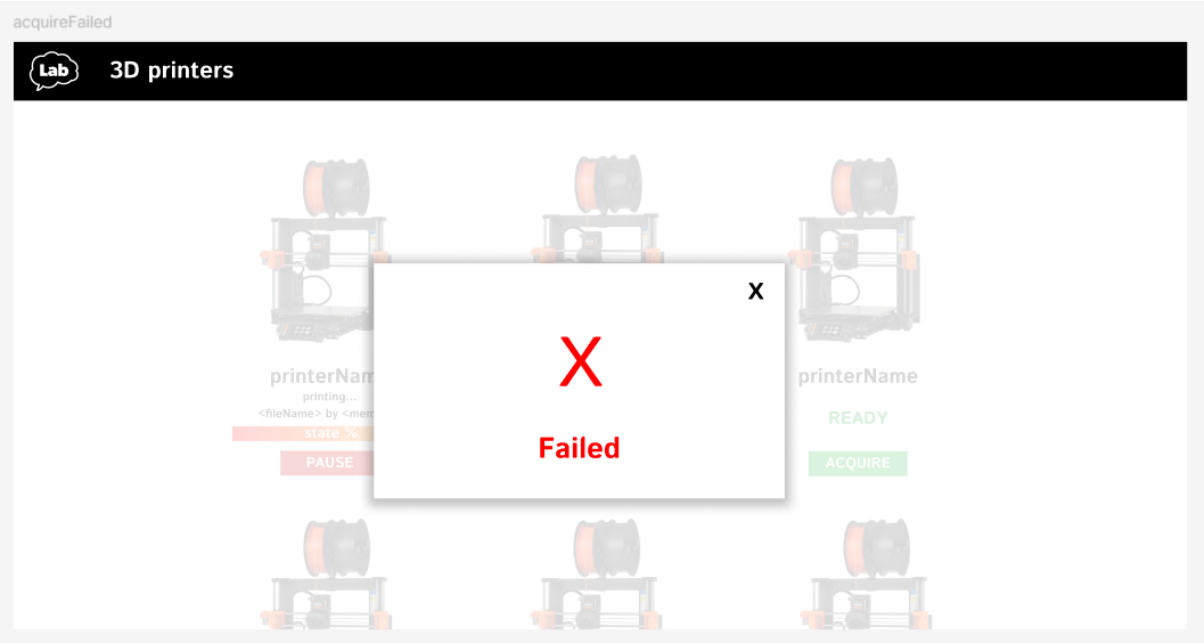
Img. 8: Tap membership card

4.4 Error Handling and Feedback

Clear messages and feedback ensure users are informed of the system's state at all times. Any errors encountered during operations, such as failed card taps or system malfunctions, are displayed prominently to assist in resolution.



Img. 9: Illustration of Successful Card Tap Status



Img. 10: Illustration of Failed Card Tap Status