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 5 seconds) queries the printers to fetch their status. The list of printers is updated every 5 minutes, checking if any new device was added. 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, rooms, 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.

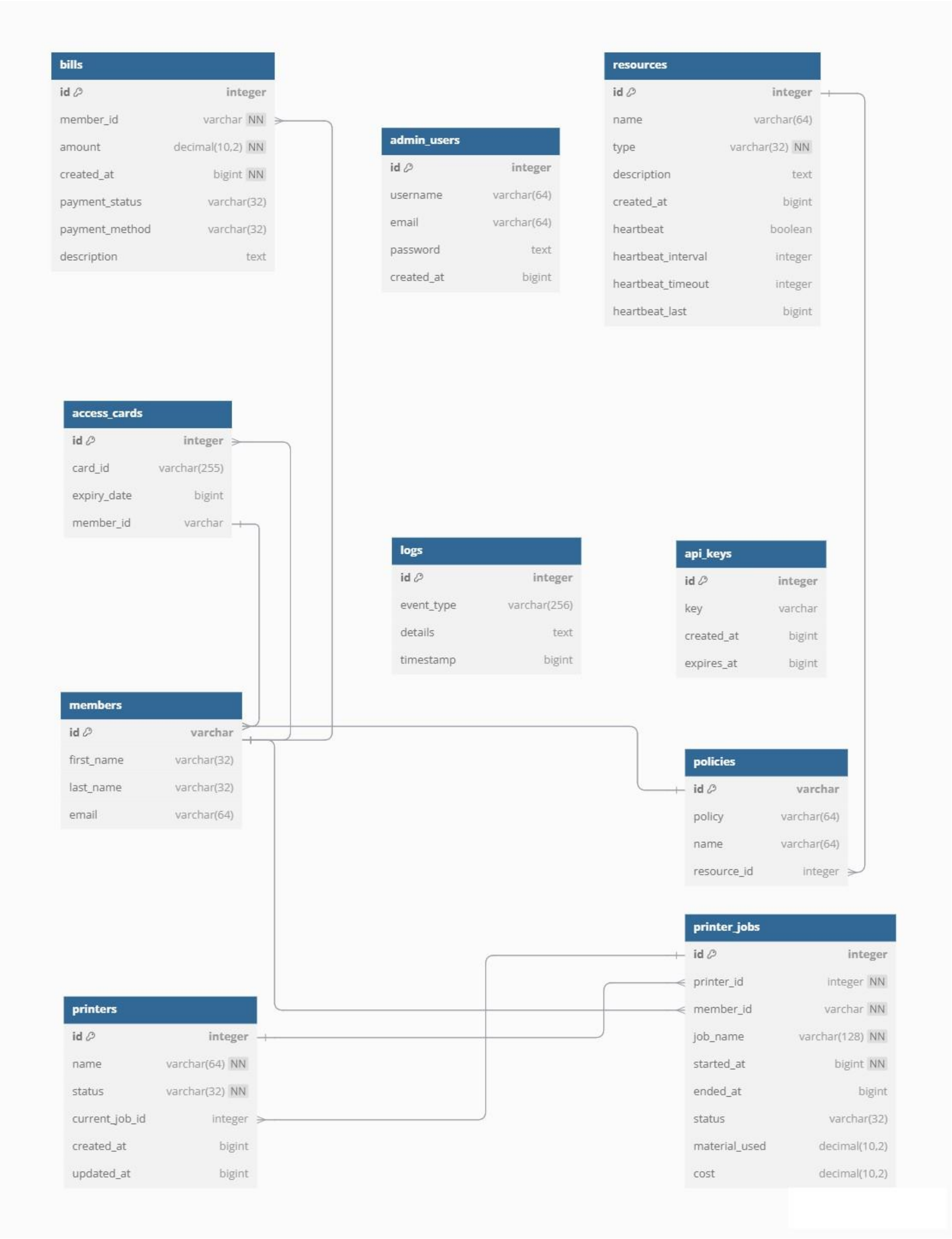
2 Detailed Data Model

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`.



bills		
id 🔑	integer	
member_id	varchar	NN
amount	decimal(10,2)	NN
created_at	bigint	NN
payment_status	varchar(32)	
payment_method	varchar(32)	
description	text	

admin_users	
id 🔑	integer
username	varchar(64)
email	varchar(64)
password	text
created_at	bigint

resources		
id 🔑	integer	
name	varchar(64)	
type	varchar(32)	NN
description	text	
created_at	bigint	
heartbeat	boolean	
heartbeat_interval	integer	
heartbeat_timeout	integer	
heartbeat_last	bigint	

access_cards	
id 🔑	integer
card_id	varchar(255)
expiry_date	bigint
member_id	varchar

logs	
id 🔑	integer
event_type	varchar(256)
details	text
timestamp	bigint

api_keys	
id 🔑	integer
key	varchar
created_at	bigint
expires_at	bigint

members	
id 🔑	varchar
first_name	varchar(32)
last_name	varchar(32)
email	varchar(64)

policies	
id 🔑	varchar
policy	varchar(64)
name	varchar(64)
resource_id	integer

printer_jobs		
id 🔑	integer	
printer_id	integer	NN
member_id	varchar	NN
job_name	varchar(128)	NN
started_at	bigint	NN
ended_at	bigint	
status	varchar(32)	
material_used	decimal(10,2)	
cost	decimal(10,2)	

printers		
id 🔑	integer	
name	varchar(64)	NN
status	varchar(32)	NN
current_job_id	integer	
created_at	bigint	
updated_at	bigint	

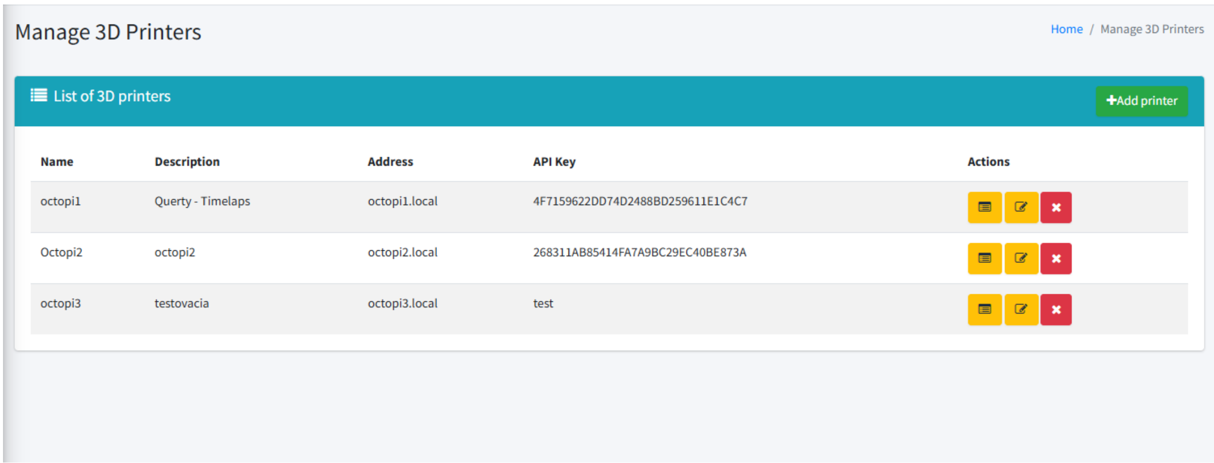
Img. 1: Illustration of the Database Diagram

3 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.

3.1 Printer List Interface

The main interface displays a responsive list of printers. Printers can be added or deleted, and the data shown can be edited.

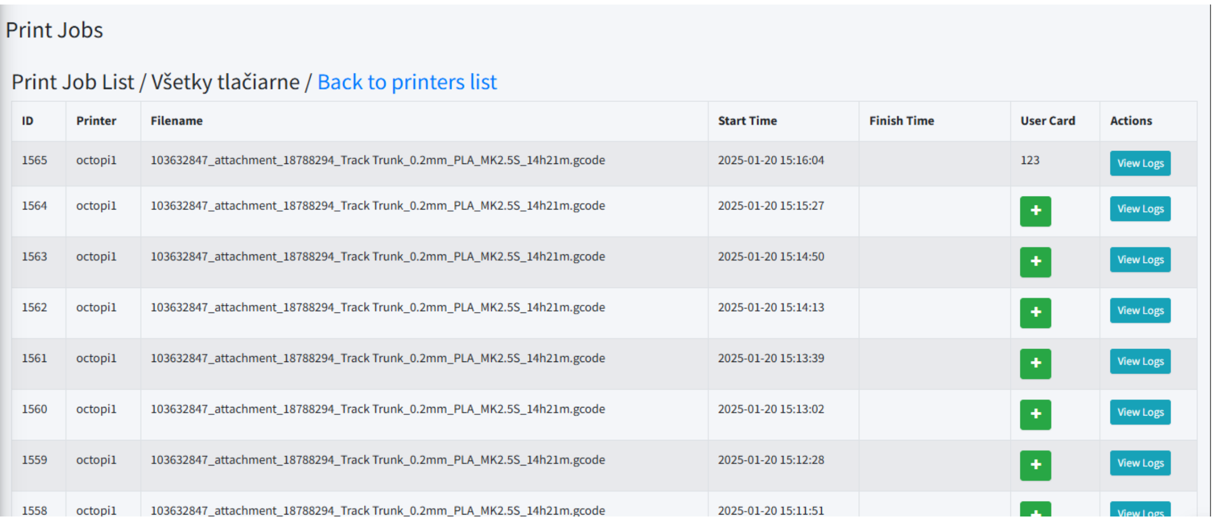


Name	Description	Address	API Key	Actions
octopi1	Querty - Timelaps	octopi1.local	4F7159622DD74D2488BD259611E1C4C7	▤ ✎ ✖
Octopi2	octopi2	octopi2.local	268311AB85414FA7A9BC29EC40BE873A	▤ ✎ ✖
octopi3	testovacia	octopi3.local	test	▤ ✎ ✖

Img. 2: Illustration Showing Printer’s list interface

3.2 Printer Jobs

Each printer's job is clearly displayed with its ID, printer name, filename (of what is printing), start and finish time of printing, username – if there is no user it gives ability to add user.
Logs of each job can be displayed.



Print Jobs

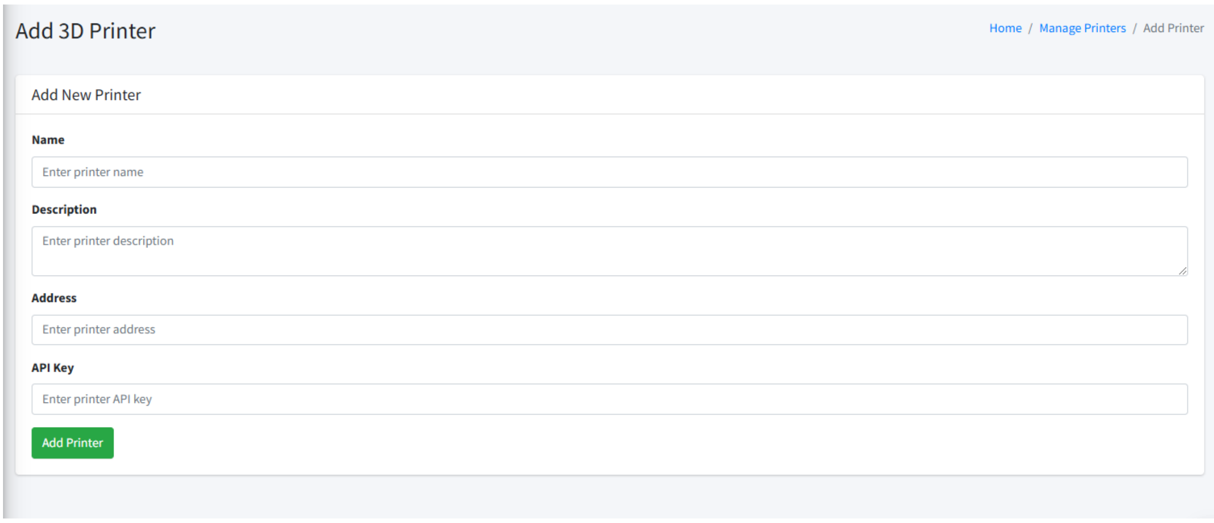
Print Job List / Všetky tlačiarne / Back to printers list

ID	Printer	Filename	Start Time	Finish Time	User Card	Actions
1565	octopi1	103632847_attachment_18788294_Track Trunk_0.2mm_PLA_MK2.5S_14h21m.gcode	2025-01-20 15:16:04		123	View Logs
1564	octopi1	103632847_attachment_18788294_Track Trunk_0.2mm_PLA_MK2.5S_14h21m.gcode	2025-01-20 15:15:27		+	View Logs
1563	octopi1	103632847_attachment_18788294_Track Trunk_0.2mm_PLA_MK2.5S_14h21m.gcode	2025-01-20 15:14:50		+	View Logs
1562	octopi1	103632847_attachment_18788294_Track Trunk_0.2mm_PLA_MK2.5S_14h21m.gcode	2025-01-20 15:14:13		+	View Logs
1561	octopi1	103632847_attachment_18788294_Track Trunk_0.2mm_PLA_MK2.5S_14h21m.gcode	2025-01-20 15:13:39		+	View Logs
1560	octopi1	103632847_attachment_18788294_Track Trunk_0.2mm_PLA_MK2.5S_14h21m.gcode	2025-01-20 15:13:02		+	View Logs
1559	octopi1	103632847_attachment_18788294_Track Trunk_0.2mm_PLA_MK2.5S_14h21m.gcode	2025-01-20 15:12:28		+	View Logs
1558	octopi1	103632847_attachment_18788294_Track Trunk_0.2mm_PLA_MK2.5S_14h21m.gcode	2025-01-20 15:11:51		+	View Logs

Img. 3: Illustration Showing Print jobs

3.3 Add Printer

To add new printer, new page displays to type in printer name, description, address, and API key. When successful, display message confirming successful action.



Add 3D Printer

Home / Manage Printers / Add Printer

Add New Printer

Name

Enter printer name

Description

Enter printer description

Address

Enter printer address

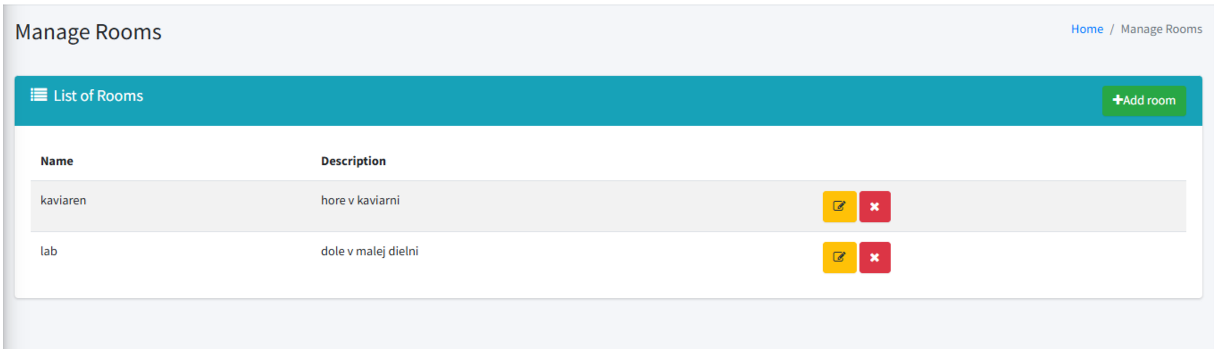
API Key

Enter printer API key

Img. 4: Illustration of Add Printer page

3.4 Rooms List Interface

Displays a responsive list of rooms. Rooms data can be edited, new room can be added and also deleted. User can manage the list of printer in every room.



Manage Rooms

Home / Manage Rooms

List of Rooms

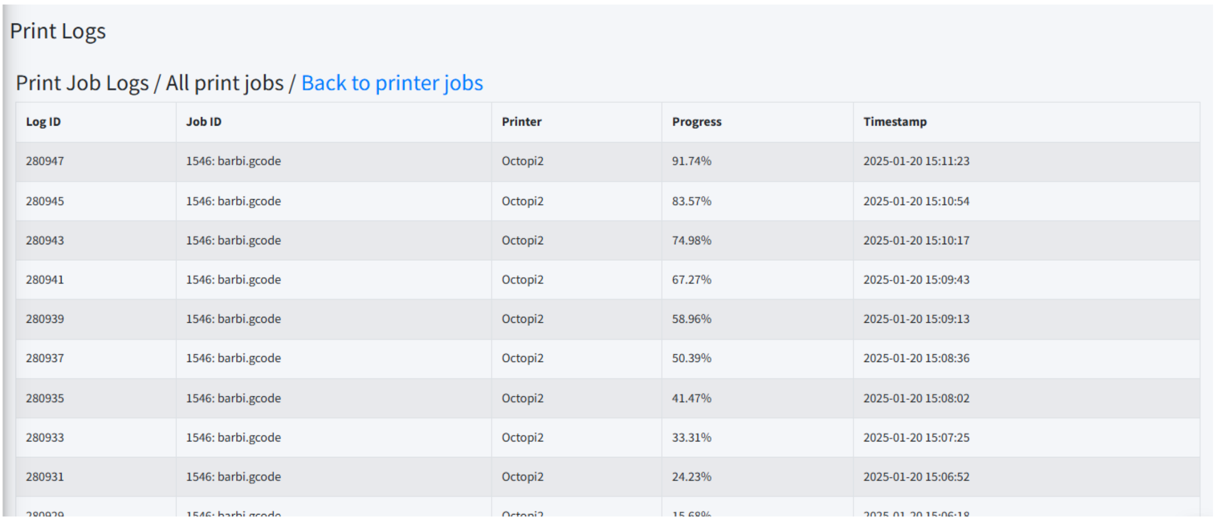
+Add room

Name	Description	
kaviaren	hore v kaviarni	✎ ✖
lab	dole v malej dielni	✎ ✖

Img. 5: Illustration of Rooms list page

3.5 Print Logs page

Simple, easy to read list of all print logs (showing most important data as progress) for print jobs.



Print Logs

Print Job Logs / All print jobs / Back to printer jobs

Log ID	Job ID	Printer	Progress	Timestamp
280947	1546: barbi.gcode	Octopi2	91.74%	2025-01-20 15:11:23
280945	1546: barbi.gcode	Octopi2	83.57%	2025-01-20 15:10:54
280943	1546: barbi.gcode	Octopi2	74.98%	2025-01-20 15:10:17
280941	1546: barbi.gcode	Octopi2	67.27%	2025-01-20 15:09:43
280939	1546: barbi.gcode	Octopi2	58.96%	2025-01-20 15:09:13
280937	1546: barbi.gcode	Octopi2	50.39%	2025-01-20 15:08:36
280935	1546: barbi.gcode	Octopi2	41.47%	2025-01-20 15:08:02
280933	1546: barbi.gcode	Octopi2	33.31%	2025-01-20 15:07:25
280931	1546: barbi.gcode	Octopi2	24.23%	2025-01-20 15:06:52
280929	1546: barbi.gcode	Octopi2	15.68%	2025-01-20 15:06:18

Img. 6: Illustration of Print Logs

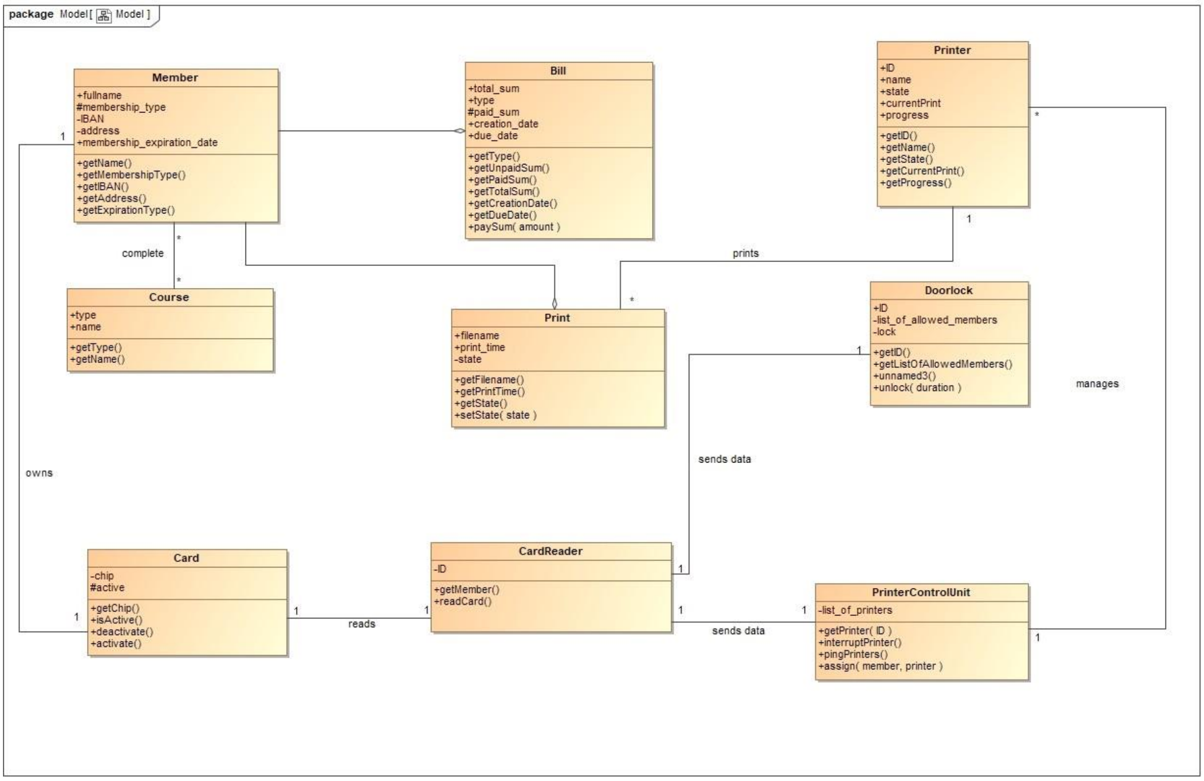
4 Implementation Design

4.1 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.



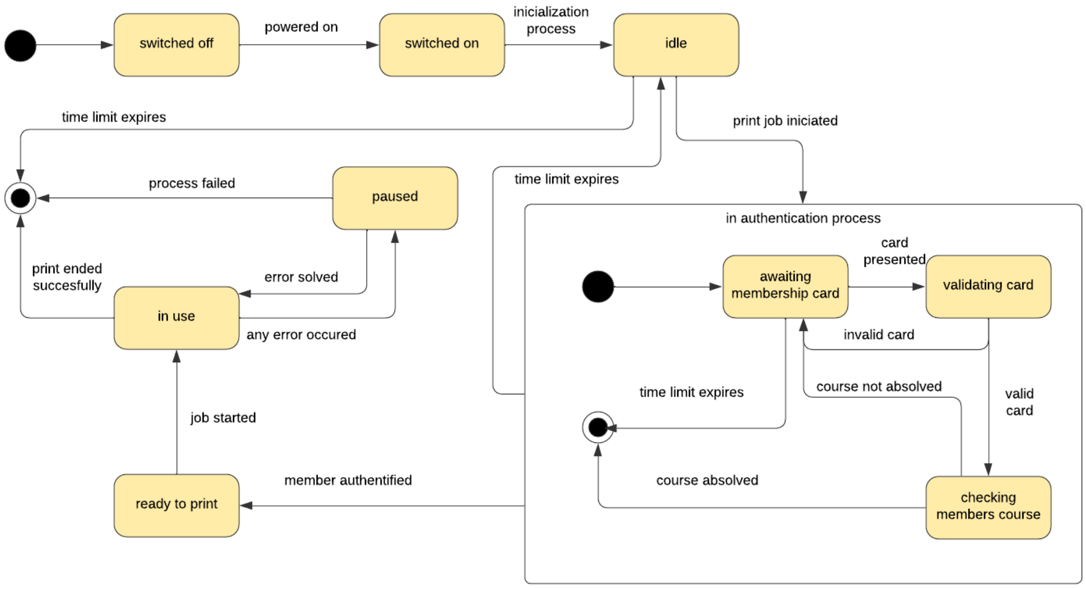
Img. 8: UML Class Diagram

4.2 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.



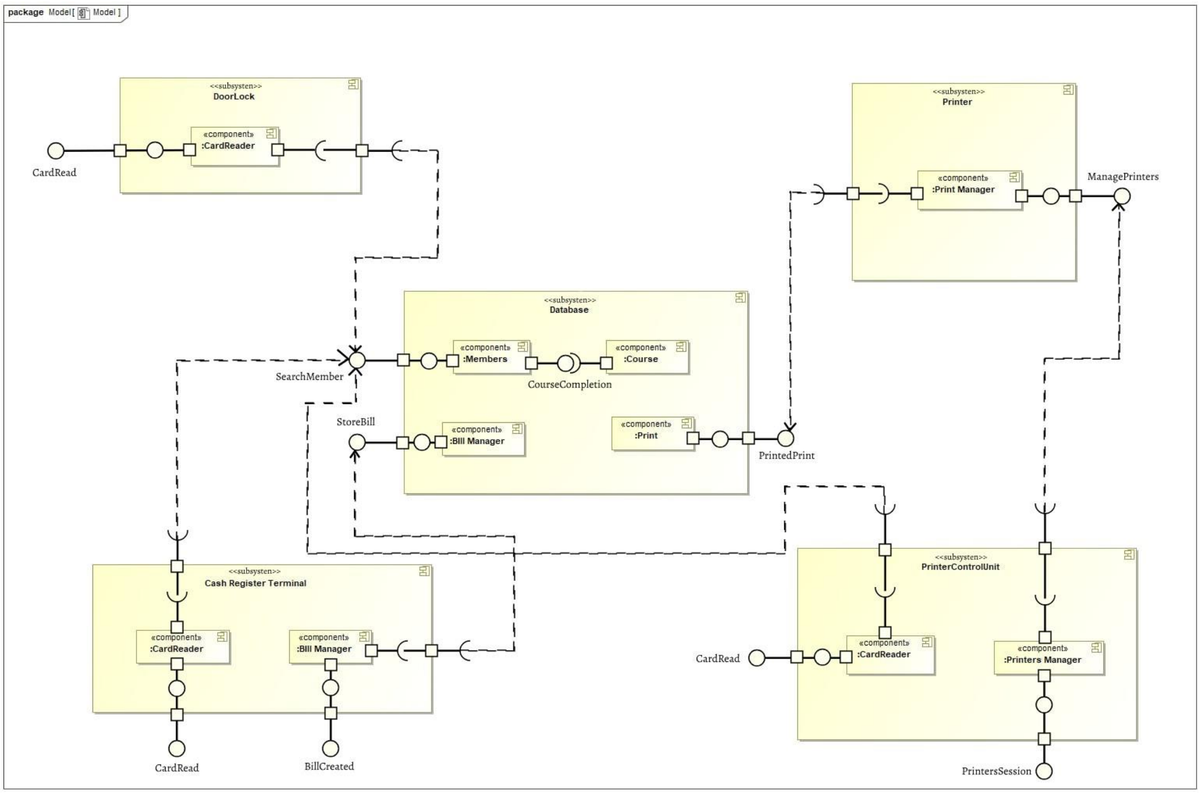
Img. 9: State Diagram of a 3D Printer

4.3 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. 10: Component Diagram of the System Architecture

5 Implement Plan

5.1 External Interfaces

5.1.1 3D Printers

Setup Octoprint: Install Octoprint on Raspberry for each 3D printer.

API Integration: Develop HTTP REST API endpoints for communication between printers and the Otello system.

Status Monitoring: Implement periodic status checks and job claiming mechanism.

5.1.2 Database Usage

Database Setup: Configure MySQL database for storing printer configurations, print job records, member data, and access rights.

Data Synchronization: Integrate Airtable for additional data synchronization.

5.1.3 Membership and Machine Access

Fabman.io Integration: Use REST API to verify user access to machines via RFID card scans.

Access Control: Implement access control mechanisms and duration settings for machine usage.

5.1.4 Payment Processing

PapayaPOS Integration: Develop endpoints for communication between LabPOS terminal and Otello system.

Transaction Handling: Implement payment processing and data storage in Airtable.

5.1.5 Membership Management

Woocommerce Integration: Synchronize memberships purchased or renewed through Woocommerce with Fabman.io via Make.com.

5.1.6 User Interface

UI Development: Design and develop user interfaces for monitoring printer statuses and managing print jobs.

5.2 Implementation Design

Member Management: Implement classes for managing members, memberships, and access control.

Printing Services: Develop classes for handling print jobs, printer control, and billing.

Printer Lifecycle: Implement state transitions for printer power states, user authentication, printing, and error handling.

Subsystems: Develop components for DoorLock, Database, Printer, Cash Register Terminal, and PrinterControlUnit.

Interactions: Define interactions between subsystems for card reading, user validation, billing, and print job processing.

5.3 User Interface Design

5.3.1 Printer List Interface

Responsive Design: Create a responsive list of printers with dynamic adjustments based on screen width.

Status Display: Implement clear status displays for each printer.

5.3.2 Printer Statuses

State Representation: Develop visual representations for printer states such as Printing, Paused, Finished, and Operational.

5.3.3 Adding a printer

User Interaction: Implement button interactions and give admin abilities such as adding printers.

5.3.4 Error Handling and Feedback

User Feedback: Provide clear messages and feedback for system states and errors.