Post-Kickoff

X

Project management report

LITHOZ®

Table of contents

Responsibilities

1.

Cserich Philipp *[Project Leader]*

"Application of user interface practices and assurance of a good user experience."

UI - Design

UX - Design

2.

Boigner Thomas *[Project Member]*

"3D preview of the printed layers and whole parts."

3D Viewer

3.

Maurutschek Fabian *[Project Member]*

"Quality assurance through git hub actions, pull request and unit tests."

Testing

4.

Siegl Bernhard *[Project Member]*

"Connection to the 3d printers and handling of the web requests with MQTT."

Web Request Handling

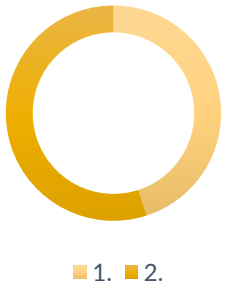
Printer Connectivity

MQTT

Definition of Done

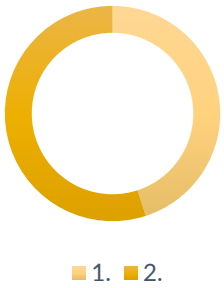
Basic Components:

Implemented UI	45%
Implemented functionality	55%



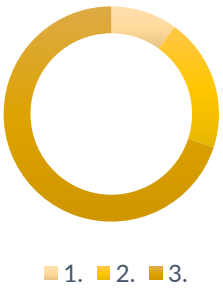
Form Components:

Implemented UI	45%
Implemented functionality	55%



Test Setup:

Imported testing environment	10%
Wrote plain tests	20%
Wrote tests for Basic- & Form- Components	70%

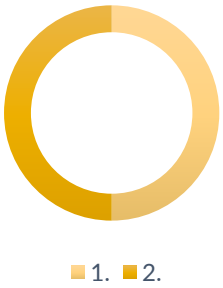


Panel & Layout components:

Implemented UI	35%
Implemented panel routing	20%
Implemented responsiveness	20%
Wrote testcases	25%

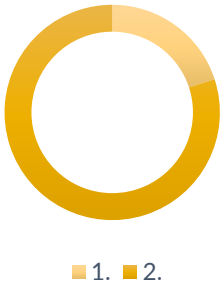
HTTP implementation:

- HTTP Get – for all required fields **50%**
- HTTP Post – for all required fields **50%**



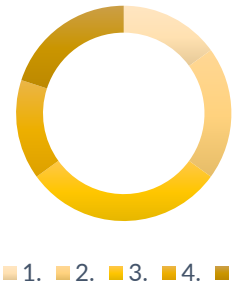
MQTT implementation:

- MQTT Connectivity/Setup **20%**
- MQTT implementation **80%**



Prototype:

- Finishing UI **15%**
- Responsiveness **20%**
- Server connectivity (MQTT/HTTP) **30%**
- 3D Viewer **15%**
- Panel Routing **20%**



Requirements definition

Definition

Lithoz Web interface will **replace** the current desktop user interface used to **control** the Lithoz **3D printers**. The big improvements from the switch to a browser-based user interface will be, that it can be used on **all devices**, independent of their screen size. This will enable the customers of Lithoz to control their printer from **everywhere** on the world. The Web interface also allows you to **monitor** the printing process through a camera in the 3D printer and the 3D viewer that will **show a preview** of the printed parts.

User Stories

- *As a company* who bought several printers,
  I want to be able to control the printer's behaviour from everywhere in the world.
- *As a company* who bought a Lithoz 3D printer,
  I want to control it from the Display on the printer.
- *As a company* who bought a Lithoz 3D printer,
  I want to view all parts of the currently printed Object
- *As a company* who bought a Lithoz 3D printer,
  I want to modify the printing process in Realtime

Other Requirements

The design and layout of the Webinterface must **resemble** the design and layout of the **original** interface from Lithoz. But not like the interface of Lithoz the Webinterface must be responsive and should look good on all devices including smartphones.

Deadlines

- **29.09.2022** Project setup and basic components
- **26.10.2022** Form components and test setup
- **21.11.2022** Hand in post-kickoff
- **30.11.2022** Panel and Layout components
- **28.12.2022** HTTP and MQTT implementation
- **30.01.2023** Quality loop
- **16.01.2023** Prototype finished
- **31.01.2023** Finishing web user interface
- **17.02.2023** Pre study finished
- **03.03.2023** Status report supplied
- **23.03.2023** Rollout completed

SWOT

S

Strengths

General team experience - We worked as a team before and could gain valuable experiences through it. We know the strengths and weaknesses of each other and know how to efficiently work together.

Time management – Through our past projects we could improve our time management skills and learn how to work efficiently.

W

Weaknesses

Lack of experience with front end development – None of the team members has experience with large-scale front-end development projects. That means that we have to spend many hours to learn front end technologies such as Vue.js.

O

Opportunity

Technological evolution – Software frameworks and libraries are always evolving and providing us with new opportunities to improve the Lithoz Web Interface. There are new frameworks getting released and already existing ones are getting updated every day.

T

Threats

Backend changes – The Backend, that controls the 3d printers, is currently being developed by Lithoz. That means, it is possible, that there are changes to the API that we must adapt to.

School stress – Due to our school we have to work on other projects and learn for exams, which takes away a lot of time that we could spend on developing the Lithoz Web Interface.

Resource storage

Local Storage

Code is written on our own device locally before it is pushed to GitHub. Also, some of the documents are saved locally while we are still working on them before they are transferred to other locations, where all the team members have access to them.

GitHub

Git is our main version management system and Github is our main platform to share and manage our code. Furthermore, its issues are used as a ticket system in which we define all

Discord

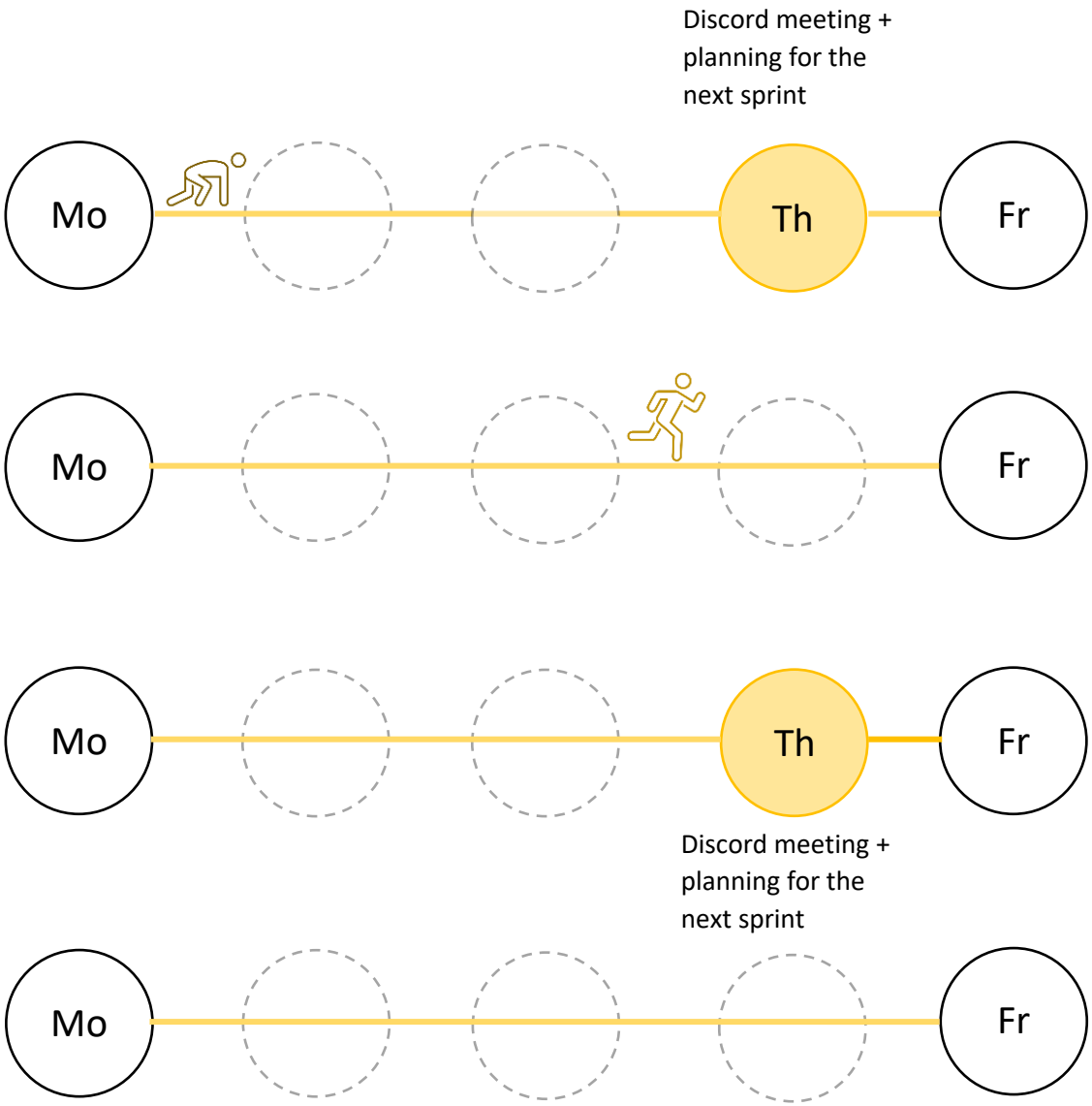
Discord is mainly used for communication purposes, but also some of the resources such as links to the most important websites and the meeting protocols are distri buted via Discord.

Clockify

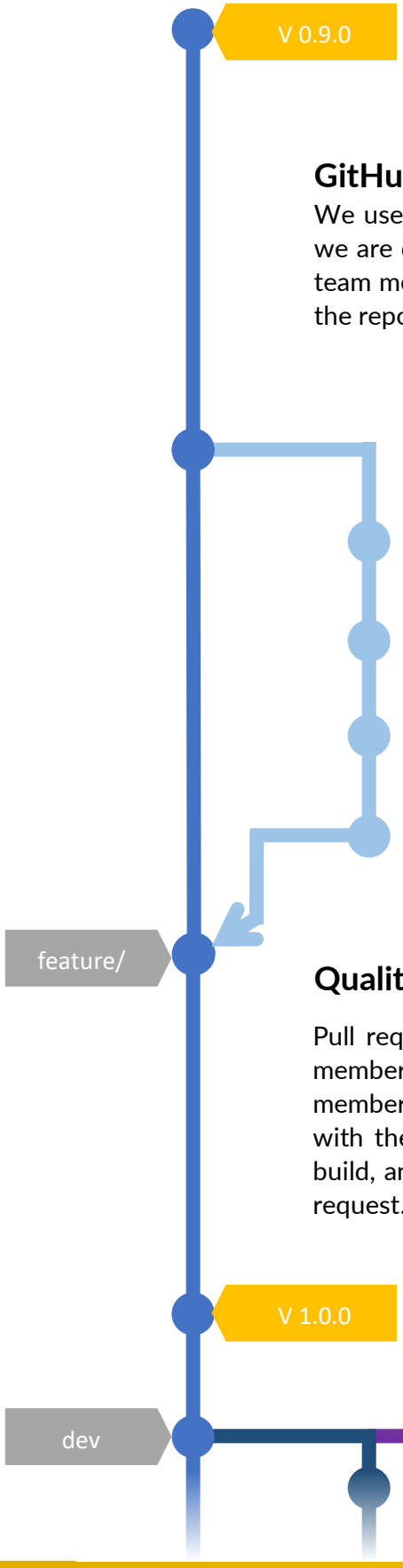
We use Clockify to track and manage the time we work on the project. It helps us to keep an overview of the time we worked on the Webinterface and makes it easy to share our time records.

Sprints

The sprint review with the Lithoz employees is scheduled on every second Thursday, where we plan and define the content of the following Sprint. Every Sprint has a duration of two weeks.



Version management approach



GitHub

We use Git for version management and push our code once we are done with coding to our private GitHub repository. All team members as well as the Lithoz employees have access to the repository.

Branches

We use three kinds of branches, on the main branch all the stable releases can be found, the development branch contains the merged feature branches and resembles the progress towards the finished product that we make, and the feature branches are where we work on our assigned tasks.

Quality assurance

Pull request are used to gain feedback from the other team members and need at least one approval of one of the team members. The project is also being build and all the tests are run with the use of GitHub actions. The project must be able to build, and all the tests must pass in to be able to merge a pull request. We use a template for all our pull requests.

Collaboration approach

Meetings

The project team has informal meetings on discord with the Lithoz employees **every second week** and formal meetings at the Lithoz office when a **milestone** is reached.

Code Collaboration

The Lithoz employees have access to our private GitHub repository, can review our code and create issues if they want to request changes. GitHub issues are also used by the team members to organize who does what.

Everyday Communication between the Team members and Lithoz

If there are any questions or something is not working the team members and Lithoz can communicate with one another via e-mail, discord direct messages and GitHub Issues.

Meeting protocols

Meeting protocol template

The meeting protocol template contains the following points:

Title

The Title of the protocol contains if the meeting was held on discord or was a physical meeting at the Lithoz office and the date the meeting took place.

Start/End time

Start and end time document at which time the meeting stated and ended.

Participants

Lists all the participants of the meeting. It is divided into the participants from the project team as well as the participants from the Lithoz employees.

Topics

Topics that we want to talk about, usually defined before the meeting.

Findings

Findings describes solutions that we come up with during the meeting. That could be ways to implement something or information from the Lithoz employees.

Next-Steps

Documents the goals that we set ourselves to reach until the next meeting.

Physical Meeting Protocol - 28.09.2022

Start: 13:20

End: 14:30

Participants

- Reiner Bachleitner
- Michael Gollner
- Richard Gradischnegg

Team

- Philipp Cserich
- Thomas Boigner
- Fabian Maurutschek
- Bernhard Siegl

Showcase - New's

- Model Viewer Pop-Up
- Button size
- formularcomponents -> Coat
- Option Menu -> Toggle Parameters
- Fine Tuning with Rows etc.

Issues

- Scaling and buttons are cut off
- generally scaleability
- formularcomponent: when something changed from the default value - put it on
- bold

Features to try

- css grid/columns
- Focus on
- full hd grid layout
- expand collapse formularcomponents

Others

- 4.11.2022 - Intern Presentation / Quality Gateway

We get

- backend connection overview / API
- MQQT to get a status (GET)
- HTTP to execute (POST)
- remote connectivity to a host in their network for exploring
- (we can load everything as a json)
- centralize the HTTP requests (because they could change)

Next up

- Three Discord Meetings till Quality Gateway to ensure prgress Thursday 4pm every week
- till 4th November(expect first week)

Discord Meeting Protocol - 13/10/2022

Start: 16:00

End: 16:30

Participants

- Reiner Bachleitner
- Michael Gollner
- Richard Gradischnegg

Team

- Philipp Cserich
- Thomas Boigner
- Fabian Maurutschek
- Bernhard Siegl

Showcase - New`s

- new Buttons
- Expert View Coating

Issues

- Expert View Coating not binded yet to the Switch

Discussing

- Any Desk TCP Tunnel connection

Discord Meeting Protocol - 20.10.2022

Start 16:00

End 16:30

Participants

- Reiner Bachleitner
- Michael Gollner
- Richard Gradischnegg

Team

- Philipp Cserich
- Thomas Boigner
- Fabian Maurutschek
- Bernhard Siegl

Predefined topics

- Responsiveness (Lithoz 1 & 2 Größem)
- HTTP fetch (Fabian)
- Expertview view
- 3D viewer start Three.js
- Icons richtig eingefügt
- Size of all panels is now the same (full page components)
- Routing gefixt

Issues

- Numpad bug

Discussing

- mqtt explorer showcase
- Vue Axios/Http requests
- tcp tunnel to remote pc
- Check which mqtt confluence pages we have
- sdl loader

Outcome

- reworking requests
- lock svg

Discord Meeting Protocol - 27.10.2022

Start: 16:00

End: 16:30

Participants

- Reiner Bachleitner
- Roland Fischer
- Richard Gradischnegg
- Michael Gollner

Team

- Philipp Cserich
- Thomas Boigner
- Fabian Maurutschek
- Bernhard Siegl

Topics

- Bugfixes
- 3D Viewer toggleable
- Fabian auswahl job
- MQTT problems
- http endpoint explanations

Findings

- Multiple tcp tunnels are a premium feature of AnyDesk
- AnyDesk Version lower than 7.1 allows multiple tcp tunnels => try to downgrade
- Responses of the Http requests => new Issue
- opening a job should load the job settings in the tune parameters panel
- commands => implement all from the pdf - especially: light on/off
- projects/{id} should load the lists of the tune parameters panel

Next-Steps

- New Issues describe the goals until next Wednesday

Discord Meeting Protocol - 10.11.2022

Start: 4:00pm

End: 4:45pm

Participants

- Reiner Bachleitner
- Roland Fischer
- Richard Gradischnegg

Team

- Philipp Cserich
- Thomas Boigner
- Fabian Maurutschek
- Bernhard Siegl

Topics

going trough

- tune parameters
- default panel
- coating
- pull request to remove/work arround vuetify -> denied from us
- how to get layerview-pictures -> sill in progress

Findings

- Subscribe/Unsubscribe optimizable -> subscribe to all at once
- reset button with boolean
- use pull request tool to format code etc.
- 6102 - to "machine"
- 6101 - to "data"
- How do we get the machine name from the api? -> not implemented
- How can we find the Stop button? -> reset will transform into a stop button (abort)
- How do I know if im in the run state or loading state? -> running
- Where should we display the time on the interface? -> not at all, but possibly to display the time from the machine

Next-Steps

- mqtt implementation
- fixing few tiny flaws
- **2:30pm next thursday testing physically**

Legal documents

2 

Before the project started the project team as well as Lithoz representatives had to sign the cooperation agreement. It defines who is the rights holder of the product, who is allowed to use it, when the project starts and when it ends, what the rights of the project team are and in which way the school is allowed to help the project team.

Additionally, the project members had to sign a legal clarification where they agreed upon all legal conditions concerning the writing process of the diploma thesis.

Tech Stack

Environment: Vite

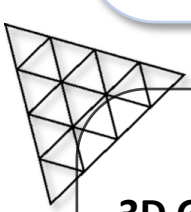
Vite is used together with node to bundle Script files and Script
libraries. It runs the server and provides Plugin Support.

Design Library: Vuetify

Vuetify provides us with various design elements, such as buttons,
text fields, sliders, and grid components.

Java Script Framework: Vue.js

Vue is the basis for our single page application. It is used for dynamic property binding and templating HTML Elements.



3D Graphics: Three.js

The Lithoz Web Interface includes a 3d viewer to display a 3d model of the parts that are currently printed.

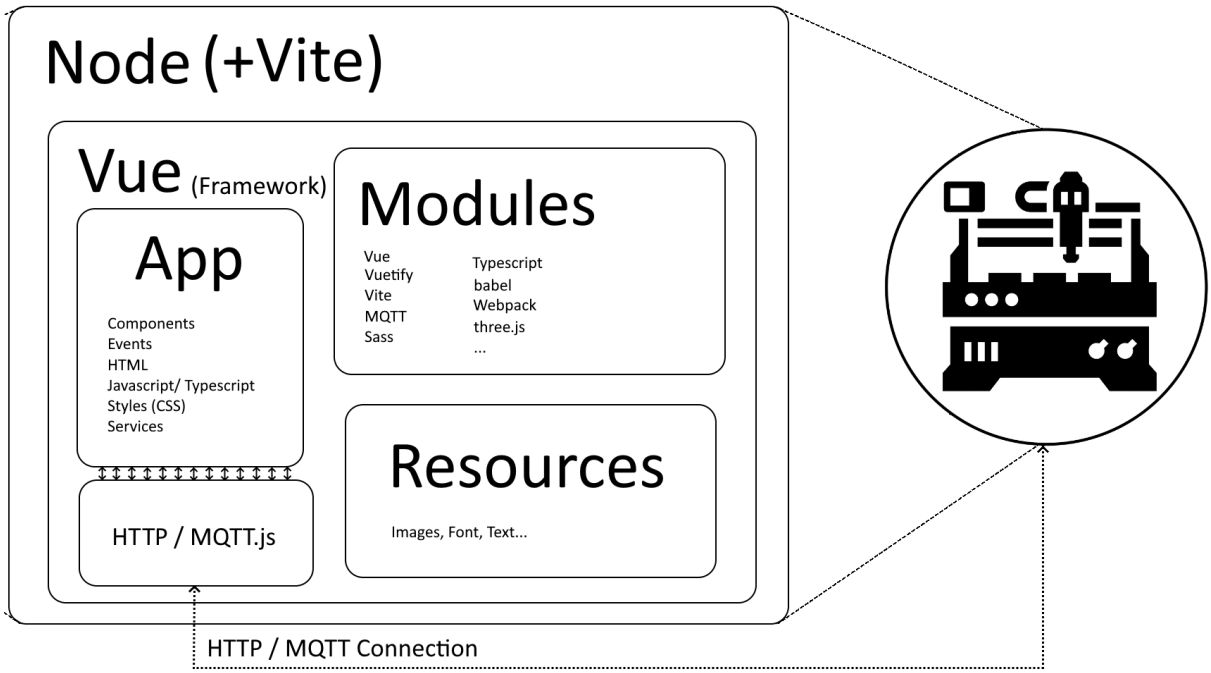
Machine to Machine communication: MQTT.js

MQTT connects the Lithoz Web Interface to the 3d printer. We use Mqtt.js as a MQTT library to establish a connection to the
3d printer and send data from and to it.

Programming Language: TypeScript

TypeScript is a programming language that adds Type support to JavaScript, giving you better tooling with an own compiler.

Architecture



Node

The WebHC architecture is based on Node.js, which is an open-source, cross-platform JavaScript runtime environment.

Modules

Within are some Node-Modules installed that are distinguish in Libraries and Frameworks. Our primary module and framework is Vue.js, which is the basis for the Vuetify-library, it provides prebuild components, code and style classes. Additionally, we have installed Typescript, Sass, and Babel for coding with certain plugins for Vue and Typescript compatibility. In the background are Vite and Webpack used for server improvements and MQTT.JS for communication with the printer.

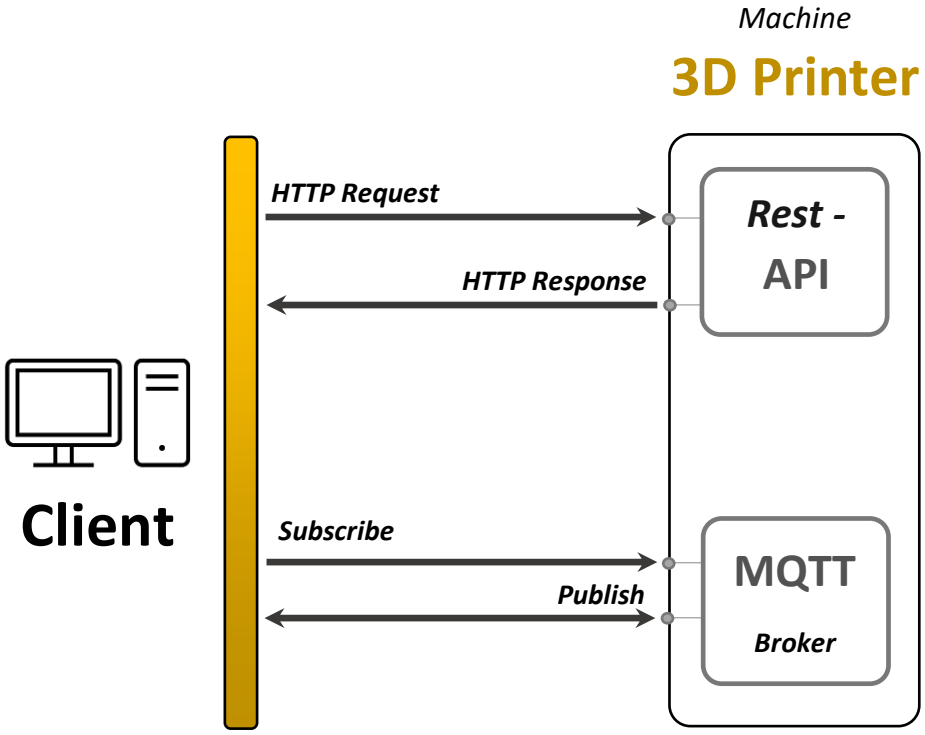
Resources

Lithoz provided us for the creation of the Web-Interface images, icons, text, and styles. These are stored as resources in the project. Due to resource-type we make use of those differently. Images, icons and styles are used within the HTML, so all resources are automatically sent from the server and used by the clients. For internationalization (i18n) the server requests from the client which language it prefers and sends then the correct language file back.

App

We use Single-File-Components in Composition-Code-Style for developing the app. With MQTT.js we create a typescript handler which provides real time information and updates some parts of the application. HTTP is used for basic information gathering from the machine user controlled.

Request Handling / Connections:



Interaction will be established through **HTTP** and **MQTT** connections.

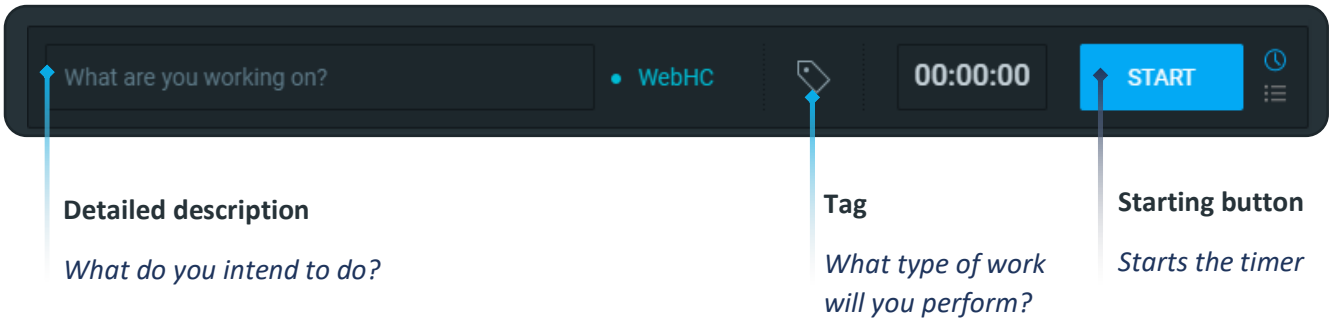
While the API requests are used to call Events, MQTT is responsible to update the displayed information in real-time.

(The visualization above is an extremely simplified version of the actual connectivity process)

Time records

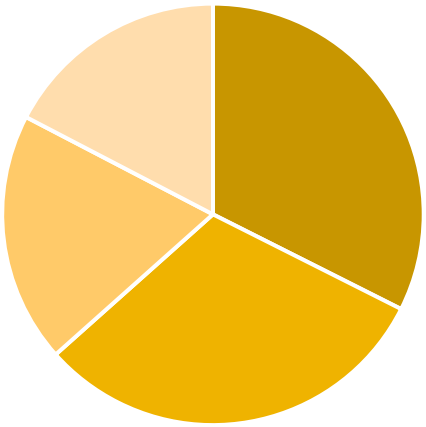
We track our Time by using a tool called **“Clockify”.**

Before the tracking process can start, we must describe what we will work on and then choose a tag to identify in what category this activity belongs.



The Clockify project for our team provides 4 different working tags.

- **Learn**
  Includes: reading documentation, inform about solutions & any type
- **Code**
  Is used for any feature implementation or adaptation
- **Test**
  Every feature and every component need to be tested.
  Despite it also being a coding task, we are differentiating it from the feature implementation
- **Other**
  This tag can be used for any type of work that cannot be described by the 3 existing tags. For example, writing Docker files, setting up the GitHub environment and adding pull request/issue templates .....



User	Time *(h)*
Fabian Maurutschek	90:39
Thomas Boigner	86:39
Cserich Philipp	53:45
Bernhard Siegl	48:32
Total	279:35

Kanban

Our Project team uses **GitHub Projects** to keep track of changes and the working progress.

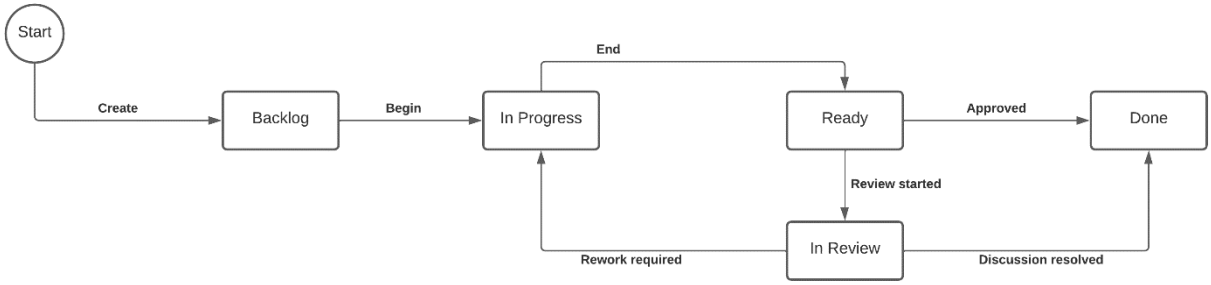
Since we use **GitHub** for version management, it is the best option to use its built-in Kanban tool.

	GitHub Projects	Trello	Jira
GitHub integration *(Issues & Pull Requests)*	✓	✓	✓
Free to use	✓	✓	X
Assigning groups	X	X	✓
Automated card movement *(On closed PRs)*	✓	X	✓
Export feature *(To pdf…)*	X	✓	✓

Board structure:

Person – Issue relations:

Controller	Person in charge of the issue
Reviewer	Person who reviews the issue and decides whether it is finished.



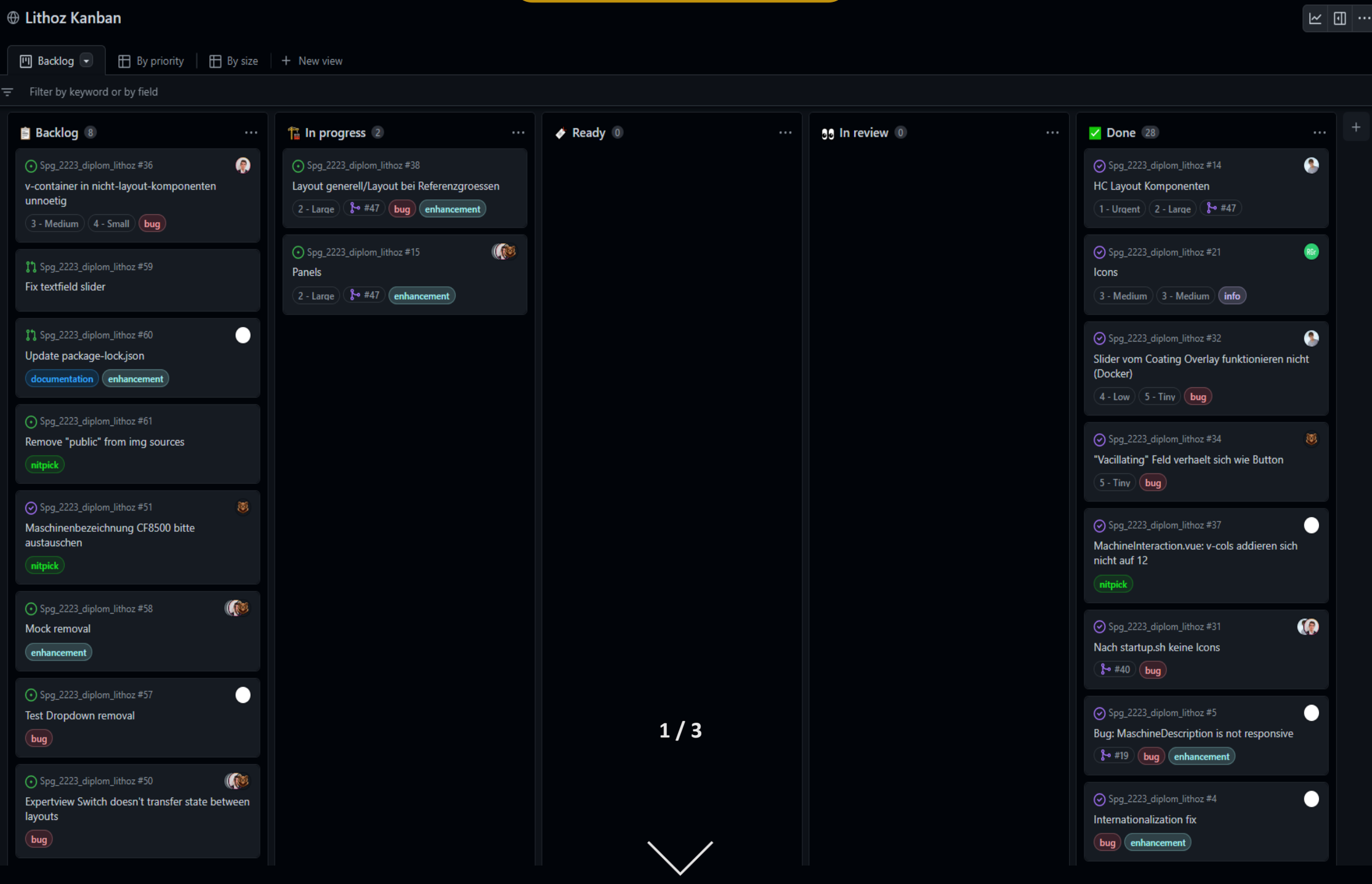
The GitHub Projects board has 5 separate stages of work.

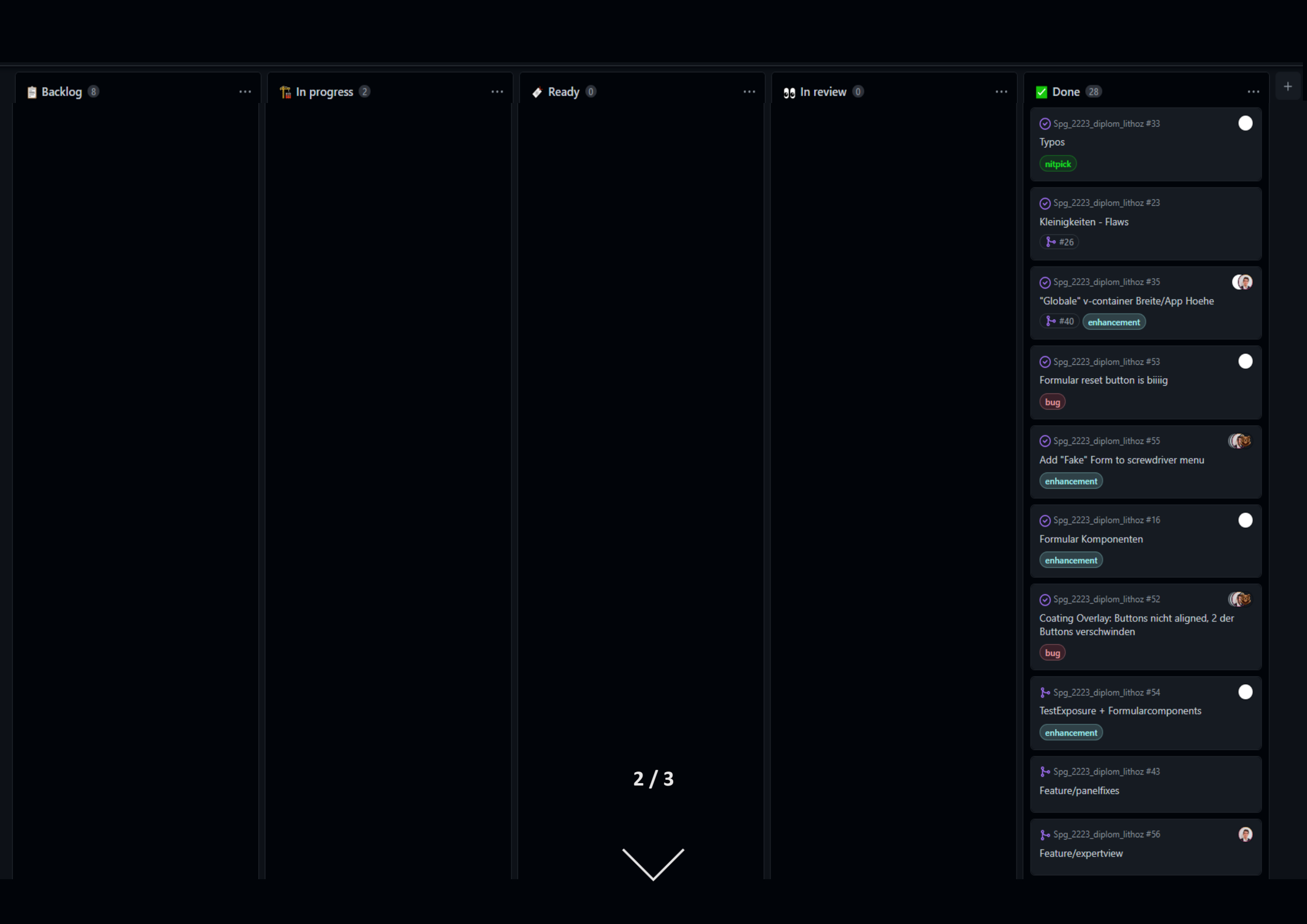
1. **Backlog**	Issues that nobody is working on
2. **In Progress**	Issues that are incomplete, but somebody is currently working on its implementation.
3. **Ready**	Issues that are finished in the controllers' eyes
4. **In Review**	Reviewer has an idea for a better solution but it has not been discussed yet
5. **Done**	Issue has been approved by the Reviewer and is officially done

The issues can move between stages in the following ways:

Create –	An issue enters the Backlog
Begin –	An issue moves from Backlog to In Progress
End –	An issue moves from In Progress to Ready
Review started -	An issue moves from Ready to In Review
Approved –	An issue moves from Ready to Done
Discussion resolved -	An issue moves from In Review to Done
Rework required -	An issue falls back from Review to Ready

Kanban – Snapshot: *22.10.2022*




Backlog 8
In progress 2
Ready 0
In review 0
Done 28
Spg_2223_diplom_lithoz #33
Typos
nitpick
Spg_2223_diplom_lithoz #23
Kleinigkeiten - Flaws
#26
Spg_2223_diplom_lithoz #35
"Globale" v-container Breite/App Hoehe
#40
enhancement
Spg_2223_diplom_lithoz #53
Formular reset button is biiiig
bug
Spg_2223_diplom_lithoz #55
Add "Fake" Form to screwdriver menu
enhancement
Spg_2223_diplom_lithoz #16
Formular Komponenten
enhancement
Spg_2223_diplom_lithoz #52
Coating Overlay: Buttons nicht aligned, 2 der Buttons verschwinden
bug
Spg_2223_diplom_lithoz #54
TestExposure + Formularcomponents
enhancement
Spg_2223_diplom_lithoz #43
Feature/panelfixes
Spg_2223_diplom_lithoz #56
Feature/expertview
2 / 3


Backlog 8
In progress 2
Ready 0
In review 0
Done 28
Spg_2223_diplom_lithoz #47
Feature/btn scaling
enhancement
Spg_2223_diplom_lithoz #49
Feature/formating fix
Spg_2223_diplom_lithoz #42
Bugfix/routing
Spg_2223_diplom_lithoz #41
Squash formularcomponents
enhancement
Spg_2223_diplom_lithoz #48
Feature/modelviewer scaling
Spg_2223_diplom_lithoz #29
Feature/formular components
enhancement
Spg_2223_diplom_lithoz #30
Feature/routing
enhancement
Spg_2223_diplom_lithoz #40
Feature/dockerfile fix
bug
Spg_2223_diplom_lithoz #39
Feature/tune parameters
Spg_2223_diplom_lithoz #46
Feature/expert view
3 / 3

Current Status

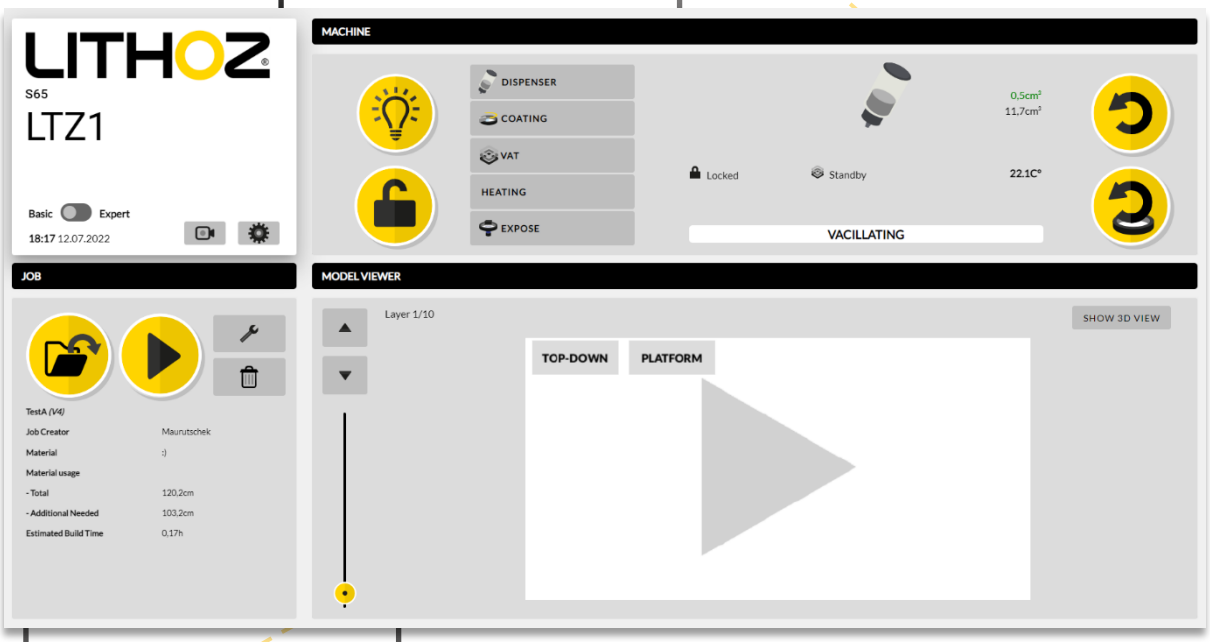
Machine Description

Includes Machine details, daytime …

Machine Interaction

Basic Printer Commands & panel reference buttons

01_Default Screen

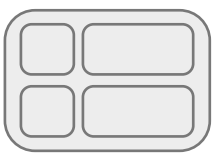


Job

Starting, importing, or configuring a job

Model Viewer

Views all Layers of the current object in the 3D Printer - *Realtime*

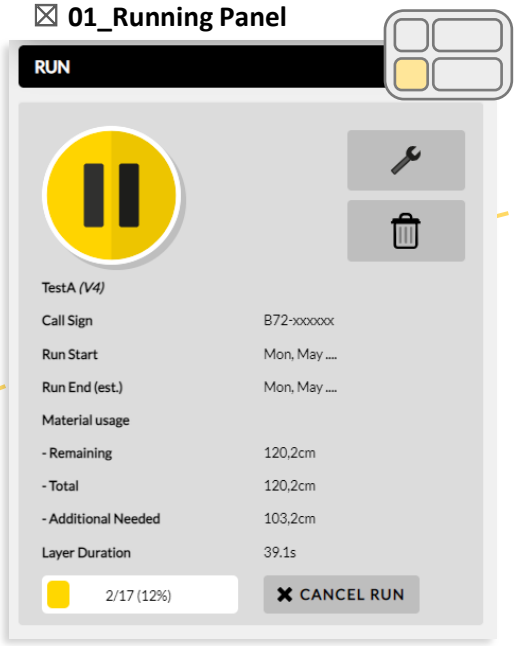


Every Panel except the Machine Description is interchangeable.

The position of the panels will be shown through a yellow marking in the graphic seen on the left.

Changing Panels:

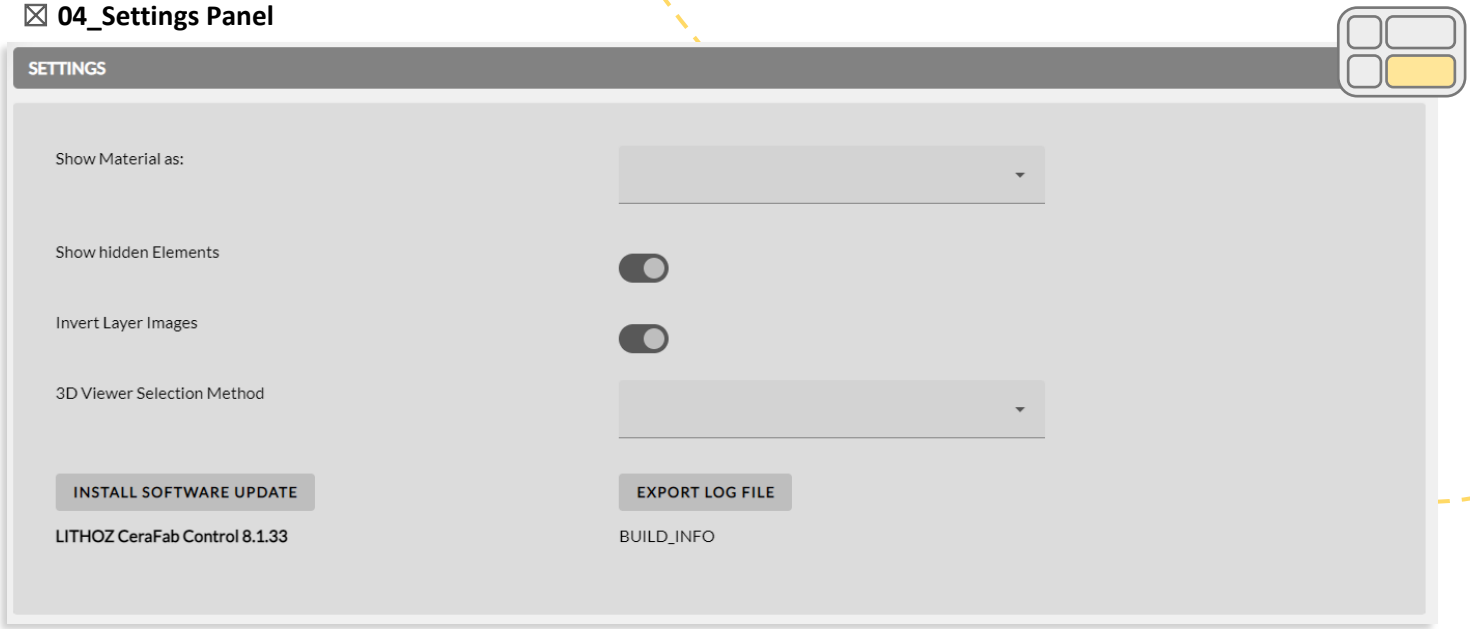
☒ 01_Running Panel



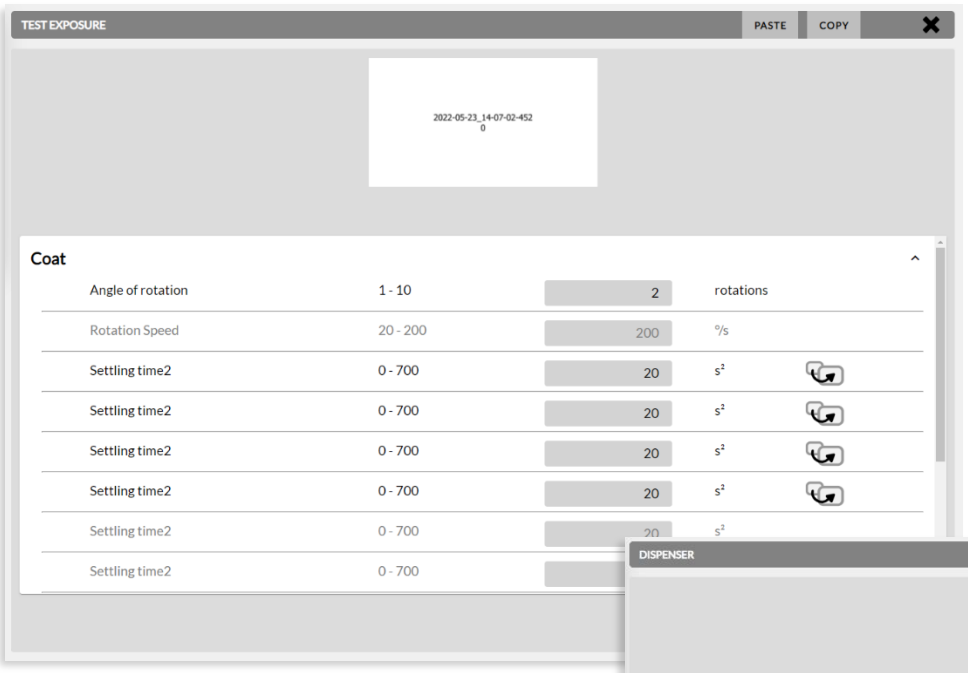
☒ 02_Vat Panel

☒ 03_Heating Panel

☒ 04_Settings Panel



☒ 05_Test-Exposure Panel



☒ 06_Dispenser Panel

☒ 07_Coating Panel