Information Management and Systems Engineering 052400 – 2022S

Laboratory Inventory System

Philipp Tschandl (a00642137)

*Note: The project was **adapted** from my own DBS project in 2021 WS.*

Milestone 1

When describing your application domain, be precise about how things are expected to work. Make sure that the things described in the application domain and/or use cases are actually working with your model. For example, if it is tracked what products a user visits in a shop, this must be explained in the outline.

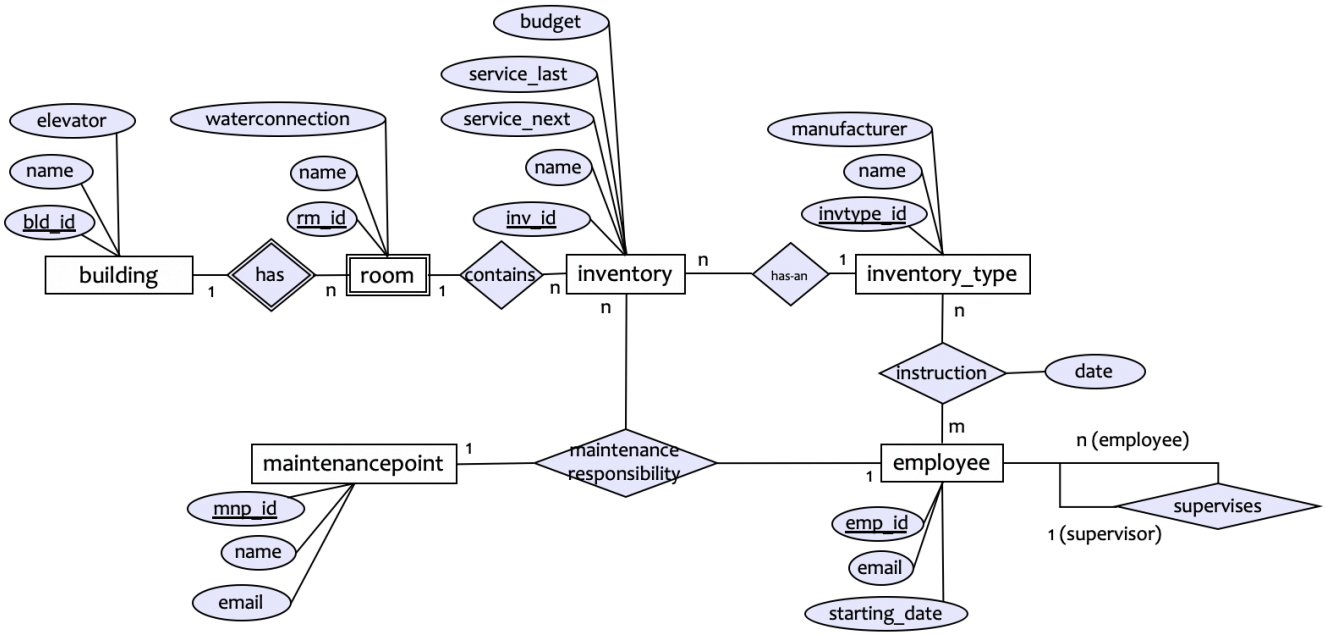
Goal

The goal of this project is to design an implement a laboratory inventory management system for the dermatopathology laboratory at the vienna general hospital ("AKH").

This laboratory is embedded within the AKH, which has multiple **buildings** of which some may not have an elevator installed – which could be important for delivering certain devices.
Buildings consist of **rooms**, which are identified by a unique identifier code, but also can have a generic name, and a specified amount or no water connections.
Each room has fixed **inventory** items assigned to it, such as tables, chairs, but also (medical) devices. Any inventory has to follow preventive maintenance routines which are called "service" in this implementation, thus any inventory item has information on the next planned service date, but also optional information about the last. Since services have to be budgeted and paid for, every inventory item has an assigned budget at acquisition, which gets reduced at every service – this is to inform management whether an item is reaching its end-of-life.
Every inventory has a specific **inventory_type**. A laboratory needs to make sure there are employees who know how to handle the inventory. Since specific inventory types can be present multiple times, e.g. 10 staining machines of type X from manufacturer Y, this information is not stored on every inventory item but a generic inventory_type.
The laboratory has **employees**, where the relevant information for this application are a unique identifier, their current email (which could change in case their name changes), and when they started working in the laboratory. An employee can be supervised by another, and can supervise others.
Employees can have had an ***instruction*** on how to use a specific inventory_type, where the date of instruction should be stored in the application.
Employees of a laboratory cannot do maintenance ("services") of an inventory themselves, even if they have an instruction on how to use them. Instead, there are **maintenancepoints** who are either within the AKH or external companies to do this. These maintenance points have unique identifiers, a name, and a contact email.
As there can be different maintenance contracts for specific inventory items – even of the same type – the information on the relevant maintenancepoint is connected to an inventory item as ***maintenance responsibility***. To distribute the burden on service management, there are also employees named within this connection.

Entity-Relationship model (Chen notation)

Strong entity (min. 4 entities)
Weak entity (min. 1 weak entity)
Unary relationship (at least 1 recursive relationship)
Binary relationship (at least one 1:n & one n:m rel., optional one 1:1 rel.)
Attributes (at least 2 attributes per entity, excl. key attributes)
Key attributes



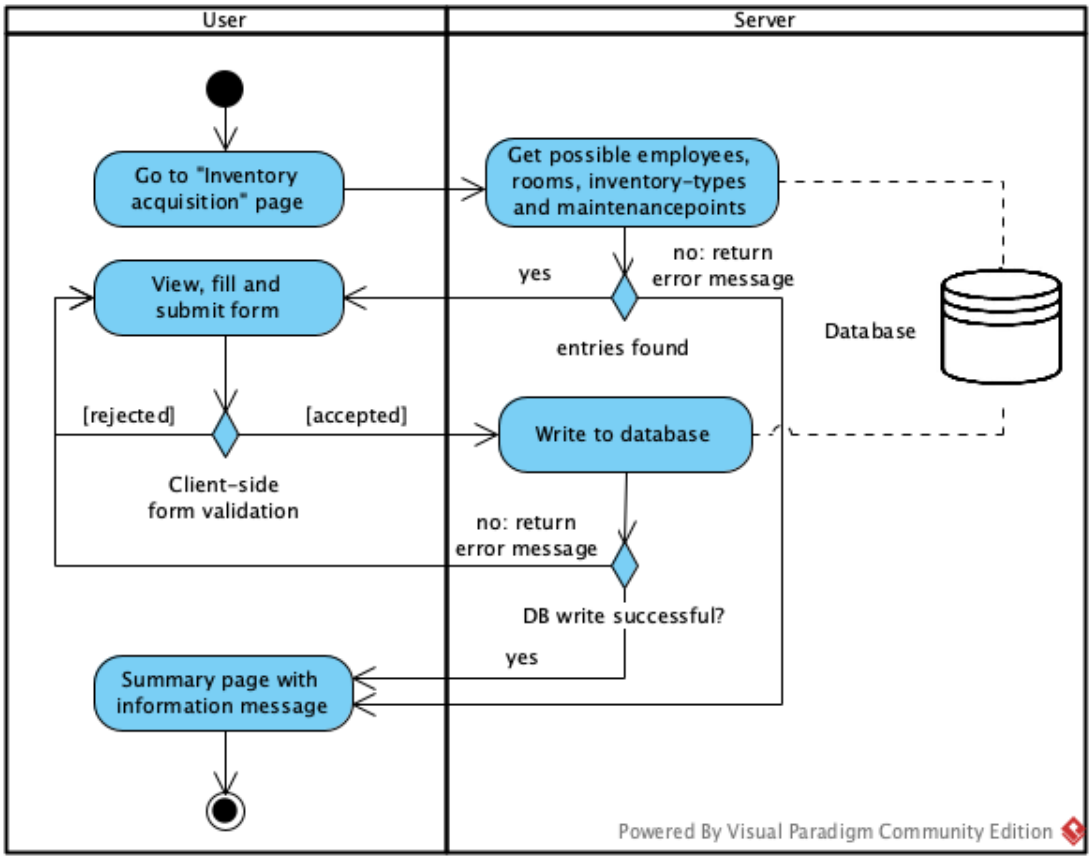
Use-Cases

Out of the five required use cases, two use cases have to be clearly indicated as the main use cases of your business. Please make sure that each main use case creates some sort of data (meaning it inserts/updates into the database).
Examples: A library could have two of the following use cases as its main use cases "renting a book", "returning a book", "register a new book", "register a new user", "add book to user favourites", … and whatever you can think of that creates some sort of data.

- *Detailed textual description A use case description has to include at least the following elements: objective, short description, precondition(s) (e.g. successful execution of other use cases or a specific system state), expected execution (set of activities), postcondition(s) in case of success and error. Verify that each mentioned data element is actually present in the model (i.e. do not refer to information that is not part of 1.1.2)*
- *Graphical representation of dynamics: Illustrate your use cases using only "Activity diagrams" (see https://en.wikipedia.org/wiki/Activity_diagram) or "BPMN diagrams" (see http://www.bpmn.org/).*

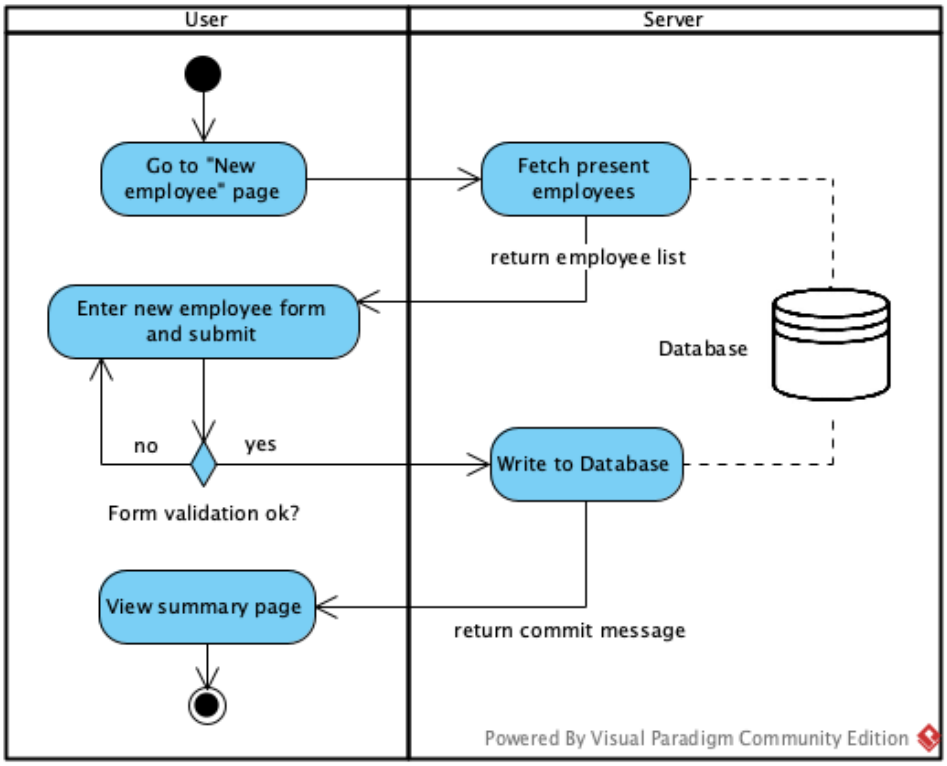
Use case 1 – Inventory acquisition (MAIN)

Objective	Register a new inventory in the laboratory
Description	When the laboratory acquires a new device/inventory, this use case should enable the user to add it to the database. Within the activity, the user should already assign necessary relationships and initial values.
Preconditions	- At least one room present - At least one inventory-type present - At least one maintenancepoint present - At least one employee present
Expected Execution	1. User navigates to device acquisition page 2. Device-types and Rooms are prefetched from the database and returned to the form elements. If they are not present end process and go to step 6. 3. User enters necessary information in the submission form and submits 4. Client-side validation – if ther is any error go back to step 3 and provide user information 5. Server tries to write data – if there is any error go back to step 3 and provide user information. 6. Summary page without form but information message
Postcondition (Success)	- Changes written to the database - Success message to the user
Postcondition (Error)	- No changes written to the database - Error message to the user



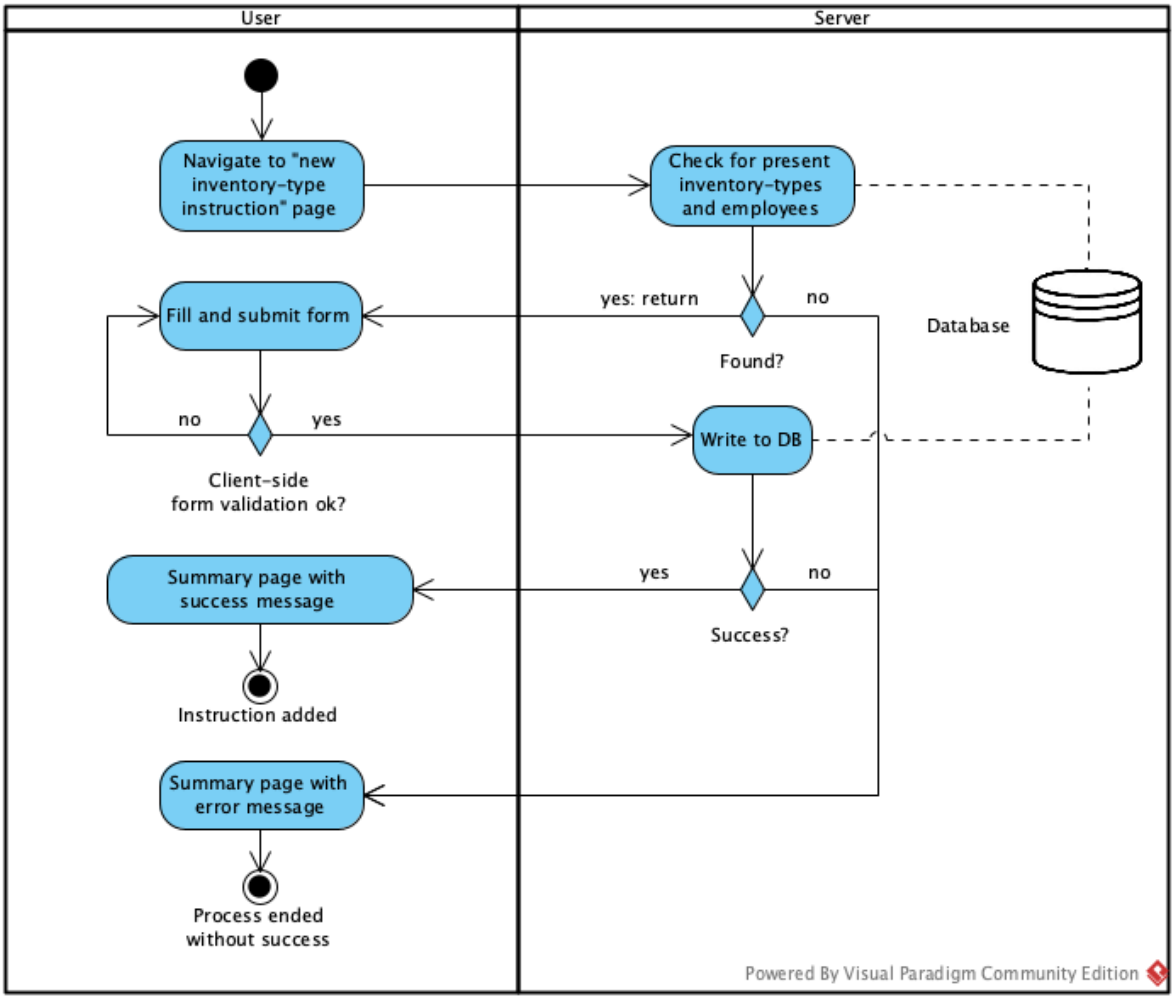
Use case 2 – New employee (MAIN)

Objective	Register a new employee
Description	When a laboratory employs a new person, they need to be registered. Existing employees (fetched from the DB) can be enteres as “supervising” employee.
Precondition	None
Expected Execution	A user enters employee information on a dedicated employee-registration page.
Postcondition (Success)	- A new employee is present in the database - Success message to the user
Postcondition (Error)	- No changes written to the database - Error message to the user



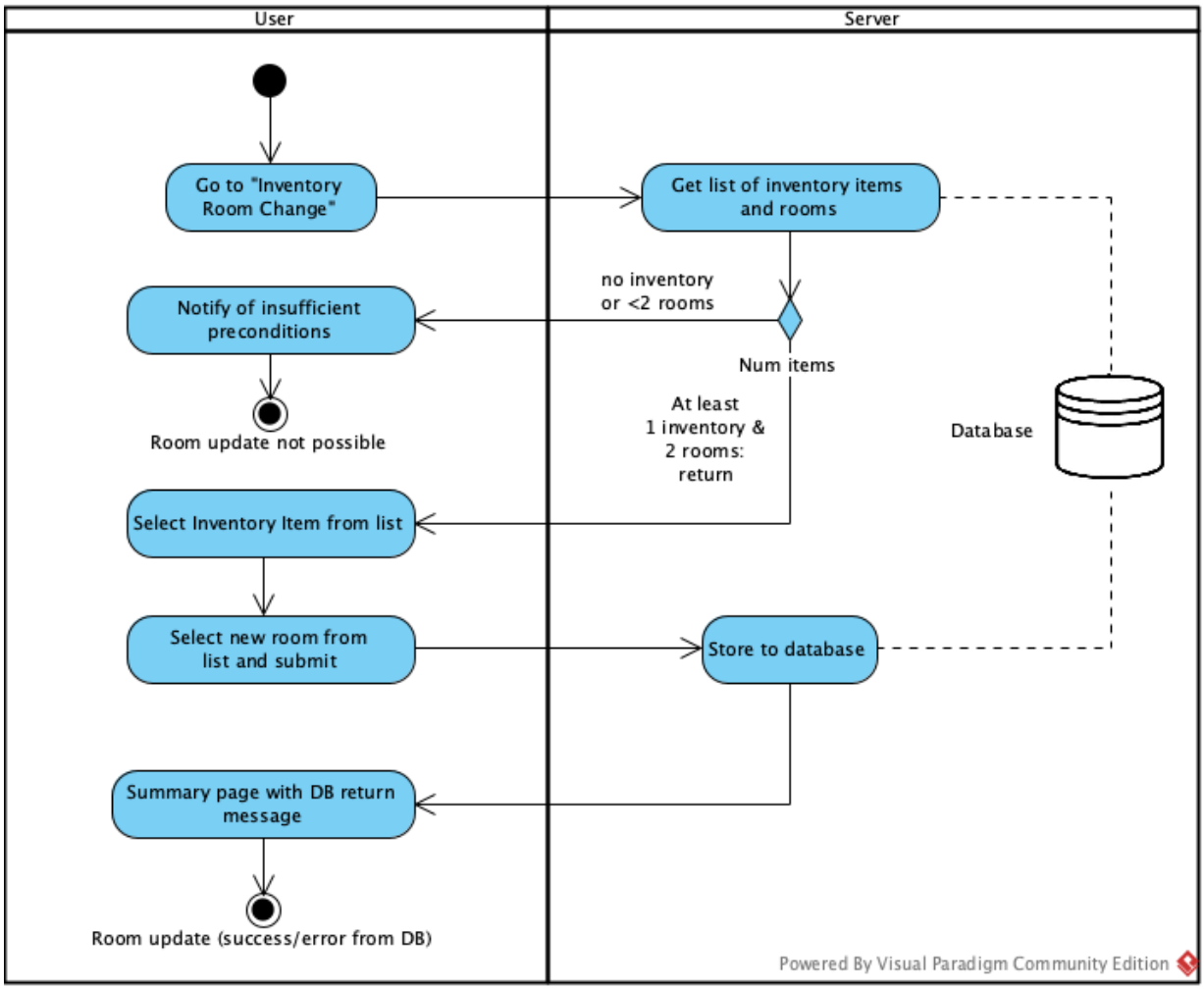
Use case 3 – New inventory-type instruction

Objective	Record a inventory-type instruction for an employee
Description	An instruction can be entered, where an employee who already has received an instruction can be entered as the instructor. When the administrator-user is entered it is assumed that the manufacturer of a device has served as the instructor.
Precondition	- At least one inventory type present - At least one employee present
Expected Execution	A user enters inventory-type instruction information on a dedicated page.
Postcondition (Success)	- A new inventory-type instruction is present in the database - Success message to the user
Postcondition (Error)	- No changes written to the database - Error message to the user



Use case 4 – Inventory room change

Objective	Change room of an inventory item
Description	Inventory items should be allowed to change room (german: “Umverortung”), but this needs to be documented in the laboratory database to be able to find them lateron.
Precondition	- At least one inventory present - At least two rooms present
Expected Execution	A user selects an item from a list of inventories, and on the following page selects the new room.
Postcondition (Success)	- Inventory item is located in a new room
Postcondition (Error)	- Inventory item is located in the original room



Reporting

Define two reports that provide some useful insights into how your company's main business is evolving. Please make sure that both reports …

- *… include at least 3 different entities (remember that joining/bridging tables resulting from m:n relations are no entities!).*
- *… define a field to be sorted on.*
- *… define a field to be filtered on.*
- *… include data written b y your main use cases.*

When describing your application domain, be precise about how things are expected to work. Make sure that the things described in the application domain and/or use cases are actually working with your model. For example, if it is tracked what products a user visits in a shop, this must be explained in the outline.

Report 1 – Maintenance plan

Outline	Looking at the whole laboratory, which *devices* are in need of maintenance in the next year, and who are the relevant *employees* responsible for it, and which are the respective *maintenance_point* contacts? This report could serve as a mailing template to be automatically sent as a reminder to responsible employees.
Use Entities	employee, inventory, maintenance_point
Filtered by	next_maintenance is in next calendar year
Sorted by	employee id, maintenance_point id
Data written	Device acquisition

Report 2 – Room inventory

Outline	Which inventory is – or should be – present in room X, and who are the employees able to handle them (i.e. who have received an *instruction* for it). This report should be used as a printout-template to pin on the door of room X
Use Entities	room, inventory, employee, (instruction)
Filtered by	selected room
Sorted by	inventory_type name
Data written	Device acquisition

Work procotol

Date	Start	Time	Task	Activity	Result	Responsibility	Remarks
2022-03-09	20:00:00	02:00:00	Project Setup	PHPStorm Project init, GitLab repository init; docker-compose.yml draft; php&mariadb&mongo&nginx images and tags; db drivers in php Dockerfile	- Basic project setup	P. Tschandl	Assess which IDE to use: PHPStorm; Assess which stack to use without needing ORM. Trying to solve db driver situation of php within a Dockerfile with mysql/mariadb (PDO vs. Mysqli) and mongodb (pecl install?) – not solved today
2022-03-14	16:00:00	00:45:00	Project Setup	initial docker-compose setup	- docker-compose.yml	P. Tschandl	Research methods for CI/CD of docker-compose
2022-03-14	16:45:00	00:45:00	M1 document	Review project ideas, guidelines and M1 definitions; group member contact	- M1 draft	P. Tschandl	Multiple channels for group member contact not successful
2022-03-15	20:00:00	01:15:00	GitLab CI/CD	trying to get dind on univie git01lab.cs.univie.ac.at/	- gitlab-ci.yml	P. Tschandl	Continue research on CI/CD with docker-compose, including testing (pytest? Python docker image?). Possible issue being insufficient access to docker socket, thus stopping this for now
2022-03-15	22:00:00	01:59:59	DB Driver	Mongodb & mysql in php-docker	- Dockerfile - docker-compose.yml	P. Tschandl	getting drivers to work (pecl vs. Apt-get vs. …) and make template queries; mysqli vs. pdo_mysql; pdo_mysql.so not found errors; mongodb ssl requirements
2022-03-16	00:00:00	01:45:00	DB Driver	Mongodb & mysql in php-docker	- Dockerfile - docker-compose.yml	P. Tschandl	getting drivers to work (pecl vs. Apt-get vs. …) and make template queries; mysqli vs. pdo_mysql; pdo_mysql.so not found errors; mongodb ssl requirements
2022-03-19	09:00:00	03:20:00	M1 document	Review sample material; adapt design from DBS, formalize use-cases and reports	- Reality Domain - ER Diagram - Report drafts - Work protocol update	P. Tschandl	
2022-03-20	10:30:00	01:52:00	M1 document	Use-case elaboration	- Use case 1-3	P. Tschandl	Find appropriate graphical design and software; diagrams.net vs. Visualparadigm

2022-03-20	13:20:00	01:05:00	M1 document	Use-case elaboration	- Use case 4 - Work protocol update	P. Tschandl	Change initial idea to new „inventory room change“
2022-03-20	14:30:00	02:00:00	SSL connection	Use nginx docker with self-signed certificate	- README.md: SSL part - docker-compose.yml updates - nginx/ssl-params-cipherlist-eu.conf - nginx/defult.conf → nginx/dev-self-signed.conf - gitignore: no crt/key commits	P. Tschandl	Follow tutorials on https://www.digitalocean.com/community/tutorials/how-to-create-a-self-signed-ssl-certificate-for-nginx-in-ubuntu-20-04-1 and ciphertext.euhttps://cipherlist.eu/. Research on whether to to use certbot/letsencrypt, reading https://certbot.eff.org/glossary#website-thats-already-online will skip using certbot for dev environment, omit HSTS and dhparams
2022-03-21	11:00:00	00:15:00	SSL connection	Use rootless docker	- README.md: Instructions for enabling privileged port on rootless docker	P. Tschandl	
2022-03-26	21:00:00	01:15:00	DB Filling	DDL script of mariaDB schema	- reset_database.sql	P. Tschandl	
2022-03-27	10:45:00	00:45:00	M1 document	Diagram design	- Added DB to graphs	P. Tschandl	Reviewed other software (ADONIS, diagrams.net) as BPMN alternatives. No satisfying complete solution found, thus reverting to VisualParadigm CE