LabMGF Warehouse Manager

Tutors: Mirko Reguzzoni, Lorenzo Rossi

Contents

Specifications 2

Objective 2

Specification 2

Specification Analysis 2

Implementation 3

Security 3

Database 4

Back and Front Ends 5

Performed Tests 5

Conclusions and Comments 5

Bibliography 6

Database 6

Back-End 6

Tomcat 6

Jakarta 6

Front-End 6

Bootstrap 6

Bootstrap Table 6

FullCalendar 6

Alessandro Tischer, 10661939

Specifications

Objective

The purpose of the project is to create an all-in-one tool that simplifies the management of the equipment in the LabMGF warehouse of related documentation and sharing.

It will be used by the following user types:

- Basic users: will be able to make requests to reserve a list of instruments and accessories, view the status of their requests and download documents related to approved requests, report malfunctions or defects of instruments or accessories.
- Superusers: will be able to add, remove, edit and link equipment and documentation, view user reports.
- Administrators: will be able to change user roles, approve, reject or mark as pending user requests.

Specification

The project was developed as a website. It is therefore divided in three parts:

- The Back-End, which was developed using Jakarta Servlet 10.0.
- The Front-End, which was developed using HTML and JavaScript.
- The Database, which was developed using MySQL 8.
- The Web Server, which is managed by Tomcat 10.

Specification Analysis

The project uses the following external libraries:

- Back-end:
  - MySQL-connector-j for database connection
  - Jakarta mail for mail management
  - Google gson for Json management (used in client communication and for reading config files)
  - bcrypt for secure password management.
- Front-end:
  - Bootstrap 5 for style management
  - Bootstrap Table for table management
  - FullCalendar for calendar management.
  - iconsvg.xyz for icons selection

Implementation

GitHub Repository

https://github.com/AlexTischer/LabMGFWarehouseManager/

Demo Video

https://github.com/AlexTischer/LabMGFWarehouseManager/raw/master/Deliverables/LabMGF%20Warehouse%20Manager.mp4

(Video file will be downloaded)

Security

- There are three filters in order to access any resource:
  - UserChecker: this will check that either the resource is public, or the user is authenticated.
  - SuperUserChecker: for any action or resource that needs superuser's clearance, this filter will block any basic user request.
  - AdminChecker: for any action or resource that needs superuser's clearance, this filter will block any basic user or superuser request.
- Since the service will be used in local network (or possibly via VPN), it doesn't implement any https strategy.
- All passwords are stored as hashed strings and cannot be decrypted by anyone.
- No sensitive information is ever sent to the client.

Database

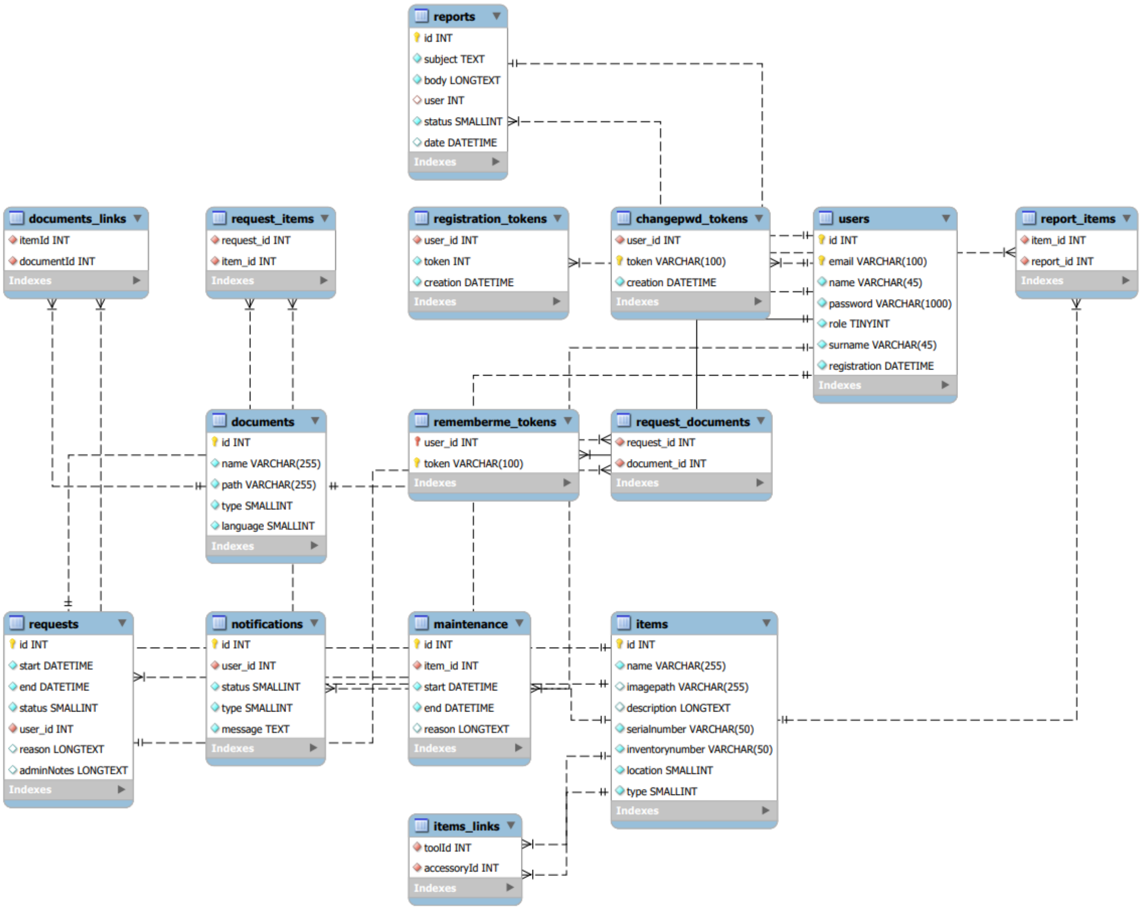

Figure 1: EER SQL Diagram

Back and Front Ends

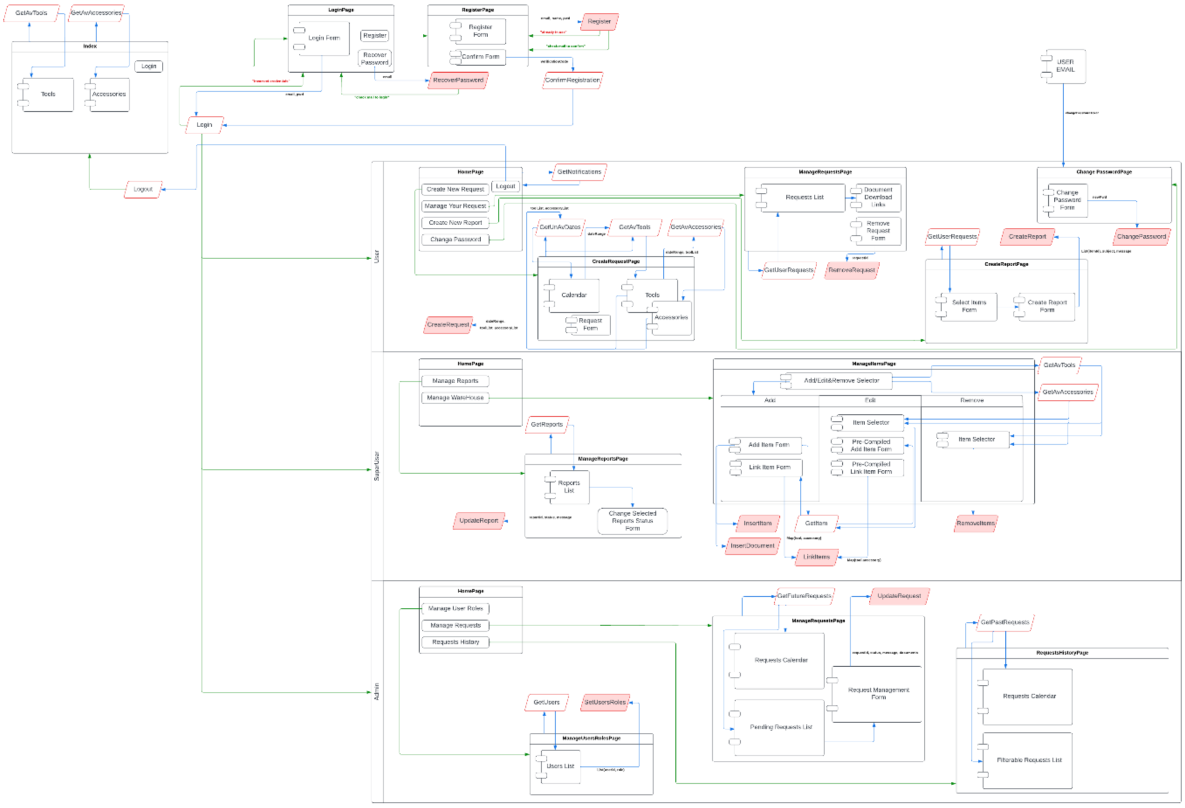

Figure 2: WebService IFML

Performed Tests

- Every possible action has been tested manually and systematically with scripted tests.
- Exceptions are caught and solved or sent as an intelligible message to the user.

Conclusions and Comments

- This project required a great amount of time, but I reached the goal I was assigned, and I can be proud of my work.
- I worked mostly autonomously, and I was complimented for this by my tutors.
- My tutors were extremely available and clear with their requests from the very beginning.

Bibliography

Database

https://www.mysql.com/

Back-End

Tomcat
https://tomcat.apache.org/index.html

Jakarta
https://jakarta.ee/

Front-End

Bootstrap
https://getbootstrap.com/

Bootstrap Table
https://bootstrap-table.com/

FullCalendar
https://fullcalendar.io/docs/