A black and white logo

Description automatically generated with low confidence

L I Q U I D

Not the videogame-library you asked for, but the videogame-library you will need

Table of Contents

[The idea 3](#_Toc73516460)

[Origin 3](#_Toc73516461)

[Functionality 3](#_Toc73516462)

[Features 4](#_Toc73516463)

[Use-Case 5](#_Toc73516464)

[Image of Use-Case-Diagram 5](#_Toc73516465)

[UI-concept: Flowchart 6](#_Toc73516466)

[Image of the Flowchart: 6](#_Toc73516467)

[Structure: Class-diagram, created files 7](#_Toc73516468)

[Image of class-diagram 7](#_Toc73516469)

[Database ERM 8](#_Toc73516470)

[Image of database-ERD 8](#_Toc73516471)

[Image of database-ERM 8](#_Toc73516472)

[Wireframe of all pages 9](#_Toc73516473)

[Login page 9](#_Toc73516474)

[Register page 9](#_Toc73516475)

[Store page 10](#_Toc73516476)

[Library page 10](#_Toc73516477)

[Profile page 11](#_Toc73516478)

[Used technologies 12](#_Toc73516479)

[Java 15.0.2 12](#_Toc73516480)

[JavaFX 12](#_Toc73516481)

[Simple-json 12](#_Toc73516482)

[JDBC 12](#_Toc73516483)

[phpMyAdmin 13](#_Toc73516484)

[Conclusion 13](#_Toc73516485)

[Stuff to learn 14](#_Toc73516486)

[MySQL & Java (for the database) 14](#_Toc73516487)

[Images & JavaFX (logo & thumbnails of games) 14](#_Toc73516488)

[Images & Java (upload for pfp) 14](#_Toc73516489)

[Workplan (Accuracy: 2h): 15](#_Toc73516490)

[The Gantt-Diagram 15](#_Toc73516491)

[First week: 15](#_Toc73516492)

[Second week 15](#_Toc73516493)

[Third week 16](#_Toc73516494)

[Fourth week 16](#_Toc73516495)

# The idea

## Origin

The ideas origin was a combination of different ideas:

3-4 Weeks before the project started, I started a project called Z-i-fy, which is a music player application (Does not have the music player function yet, but unnecessary functions, like changing the background, by getting the most used colour in an image and much more). This project was made with mostly PHP, but also python (4 img -> 1 for the “algorithm”).

I wanted to create something similar and had the new UI of Steam in my mind, so my first idea was created: A game library

The second idea came from a project idea Diego told everyone in the beginning: A store.

And if you fuse the two together you have a small system, where the user can buy video games in the store and then view them in their game library.

And as every store needs a profile to change the password, the third idea was born: A user profile where the user can edit everything, but their set log in name.

The fourth idea came to my mind, as I was thinking about the store. A community store, where users themselves can sell items from their inventory to others.

## Functionality

As you can probably guess by reading the features, the main activities a user can do in this application are viewing bought video games, “buying” video games, editing username & password, (switching between viewing modes in library), (selling and buying items in the community store).

The activities mentioned in parenthesis are only going to be implemented into the program if I have enough time to finish them. Fully. That means, I will focus on the store and library features first and only then start the work on the viewing modes in the library. In case even more time to finish the project is available, I will be creating the community store.

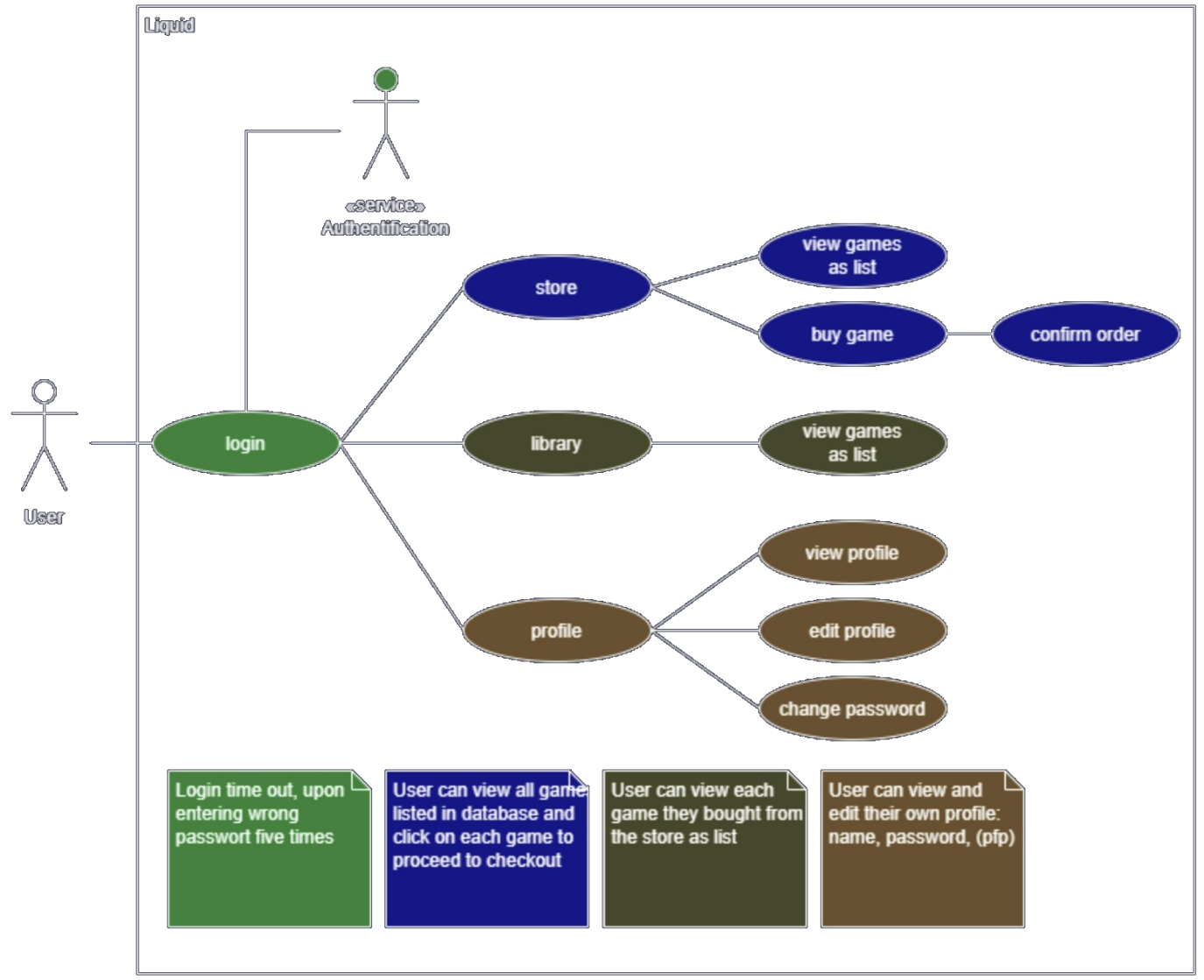
And as an old farmer once said: “It ain’t much, but it’s honest work”.

## Features

* Library
  + All games listed
  + Two modes (Second mode only if enough time available)
    - A list with all the videogames bought via the shop
    - (All the videogames bought via the store as clickable thumbnails)
* Shop
  + List of buyable games in database
  + Search field
    - Search by name
  + Filter (Second and third only if enough time available)
    - By price → Set max. price
    - (By category → checkbox)
    - (By OS → checkbox)
* User profile
  + Login system
    - Set username (used for login, not editable)
    - Set username (used for display, editable)
    - Set password (used for verification, editable)
  + (Profile picture)
    - Custom user upload (used for display, editable)
    - Downscaled .png image (128 x 128) to save space
  + (Inventory)
    - List (or thumbnails) of items bought from the community store
* (Community store) (Whole community store (& inventory) only if enough time available)
  + Buyable items
    - user → user,
    - transaction fees
    - thumbnails

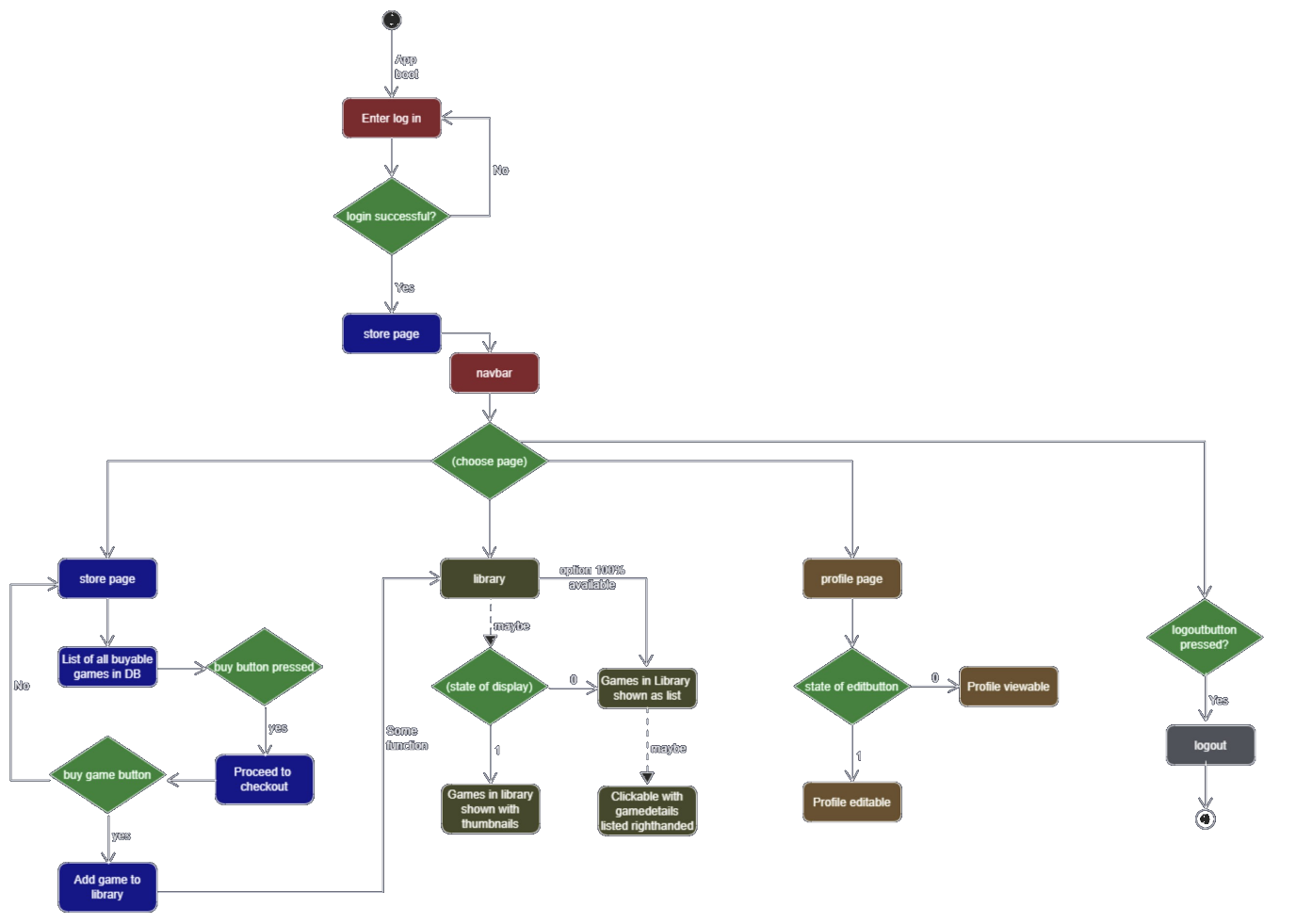
# Use-Case

## Image of Use-Case-Diagram



# UI-concept: Flowchart

## Image of the Flowchart:

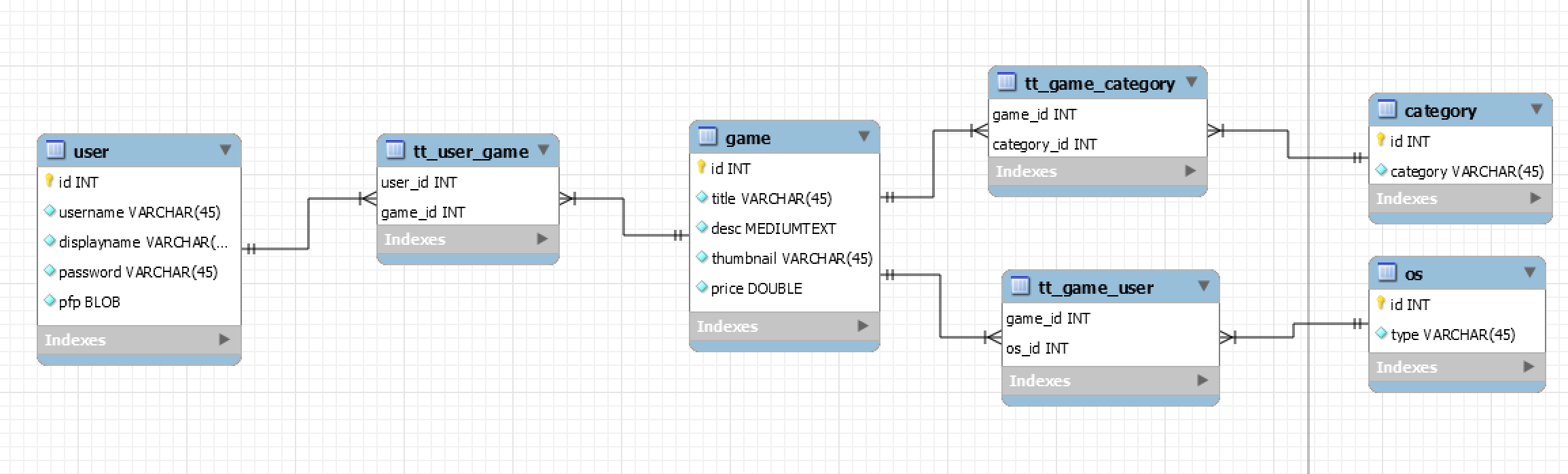


# Structure: Class-diagram, created files

## Image of class-diagram

# Database ERM

## Image of database-ERM



## How I will work with the database

Store page:

The videogames on the store page will be shown, by an SQL-query to the games table and joins to the tt\_game\_category/os tables.

Library page:

Same as the above, except the join for the category and os will be replaced by a join to the tt\_user\_game table, to check if a user really bought the game.

Profile page:

The profile page is a bit easier, due to no joins I will have to use. It only has to show the pfp, user and display names and last but not least the “password-change-feature”.

The catch is an insert statement, if the user wants to change his password. But also this is pretty easy. The challenge here is to make it secure and prevent the database from injections.

Log in page:

Here I am going to use a simple Query to the user table, to check if the entered username and password are correct

Register page:

A small insert statement into the user table. (Maybe adding a token table later on)

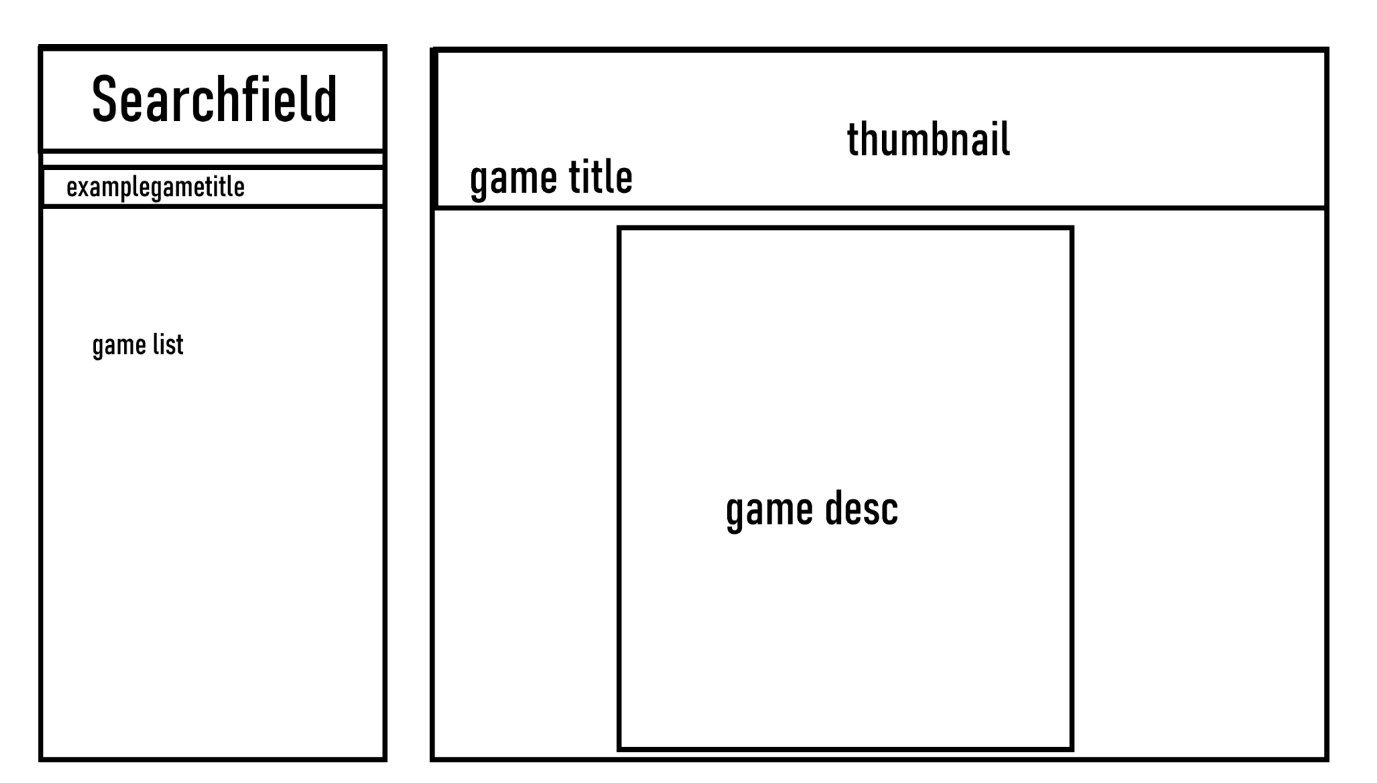
# Wireframe of all pages

## Login page

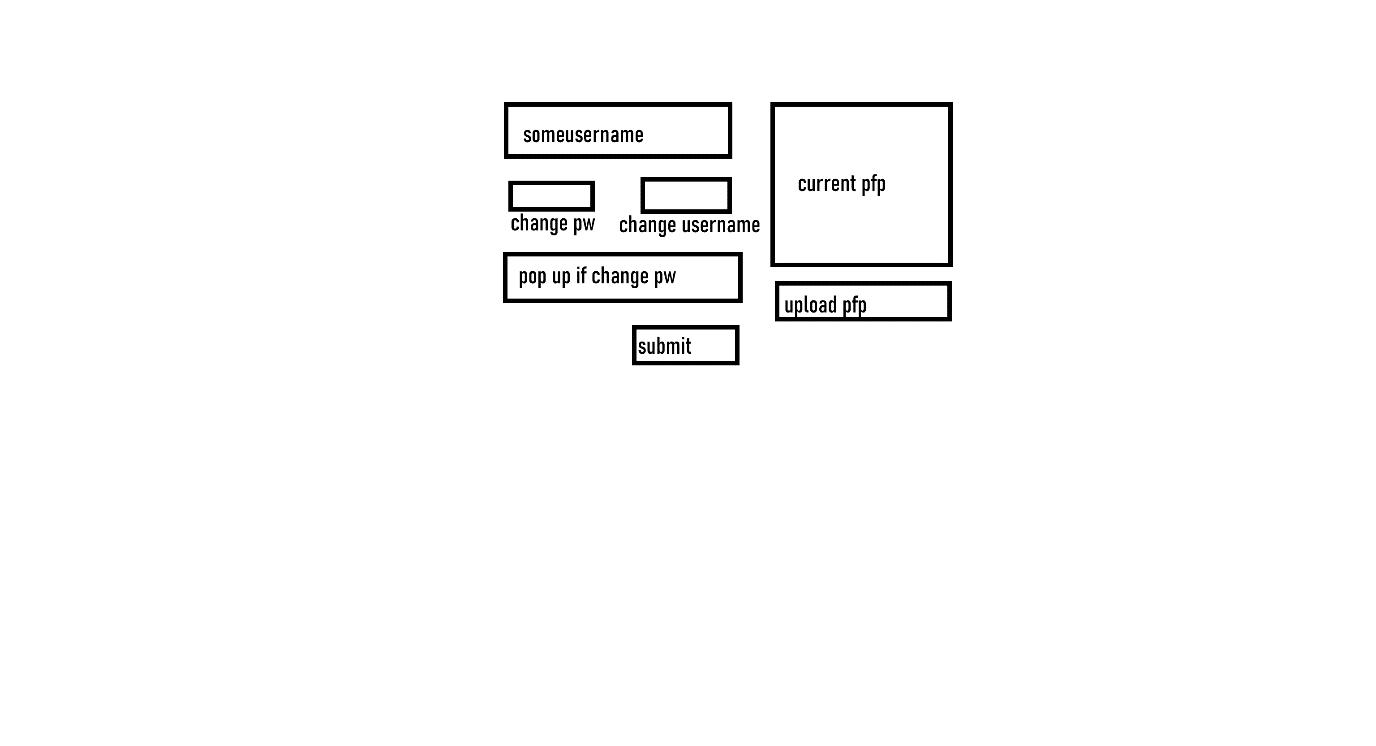
## Register page

## Store page

Library page



## Profile page



# Used technologies

## Java 15.0.2

Java is the programming language I will be using during this project.

Everything in this project will be written in Java and *JavaFX* (excluding the .css and .fxml files).

## JavaFX

Along to Java itself, I also will be going to use lots of JavaFX.

JavaFX is an additional library for Java to create visual applications for all operating systems (Assuming a working Java version is installed). It is going to be the front end of my application and will be used to show everything the user needs to see.

It is the interface between the user and the back end, which means it also will be used by the user to enter certain things like his login credentials and/or to change his displayable username & password.

In the user’s perspective, the front-end is the most important thing. That is why I am going to try my best in its creation, keeping it simple and good looking, yet functional at the same time.

## Simple-json

Simple-json, also is an additional library for Java.

The library’s name already gives away all its functionality. It is for reading and writing from and to json files and of course, it’s simple.

It has lots of built-in functions I can use to convert json objects into “Java-readable” stuff.

I do not even know ifI am going to use any form of json files in my project, but I added the library to my project, just in case I need it.

## JDBC

JDBC is an additional library I have never used before. I don’t really know what to write here, as I have no idea, what its functions are, nor how to use it.

But it is needed to for a connection to a database, so I am going to use it.

## phpMyAdmin

phpMyAdmin is the web-interface of MySQL or something like that.

I do not know its purpose or what exactly it is, but I will use it to directly see changes made by my code. (In my opinion it is easier to use than a console interface-database-management-system like MariaDB)

## Conclusion

By now, I am pretty used to Java, JavaFX and simple-json, which means I am probably going to finish this project, without the thumbnails for each game, due to me having to save them as a blob into the database.

Speaking of databases: I have not tried connecting to a database with Java (yet). JDBC will be the only new technology, I will have to learn from scratch, for the completion of this project, but I don’t think this is going to cause too much trouble.

The rest should not be a problem, because I have *“lots”* of experience with MySQL and PHP.

# Stuff to learn

## MySQL & Java (for the database)

As I already know how to work with MySQL, I only have to learn JDBC, the library for Java to connect the application to a database.

I do not think this will be a problem to learn, due to it being simple (At least the stuff I’ve read about it sounded simple).

It is pretty much the same, with all the functions you need to call, so you can access the database. The hardest part is going to be the “bringing-it-to-the-front-end-part”. And even that is pretty simple to do.

## Images & JavaFX (logo & thumbnails of games)

One of the harder things I need to do is the blob to image conversion in Java, to make the logos, profile pictures and thumbnails work. I barely know how to do this in PHP and am pretty new to the bit64 conversion.

It already was hard to make it work in PHP and I hope I will manage do it faster and better in this Java project.

## Images & Java (upload for pfp)

Right now, I do not know, if I need to convert any image to bit64, to upload it into the database or if I can just upload it as a blob, but that is not the big problem here.

The problem with the profile pictures will be the downscaling of them.

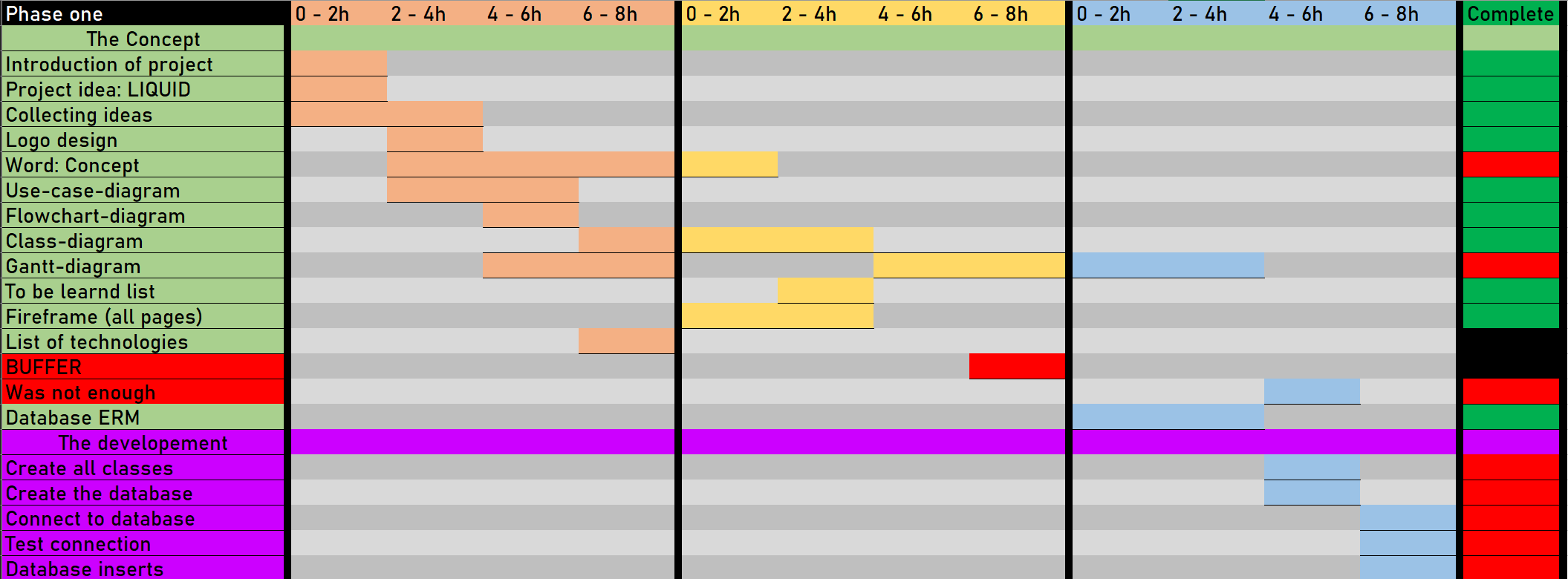
To save lots of memory, I need to downscale the pictures to at least 256p x 256p (It will probably be 128p x 128p).

So first I need to find a library, which can do that and also find a way to do it fast.

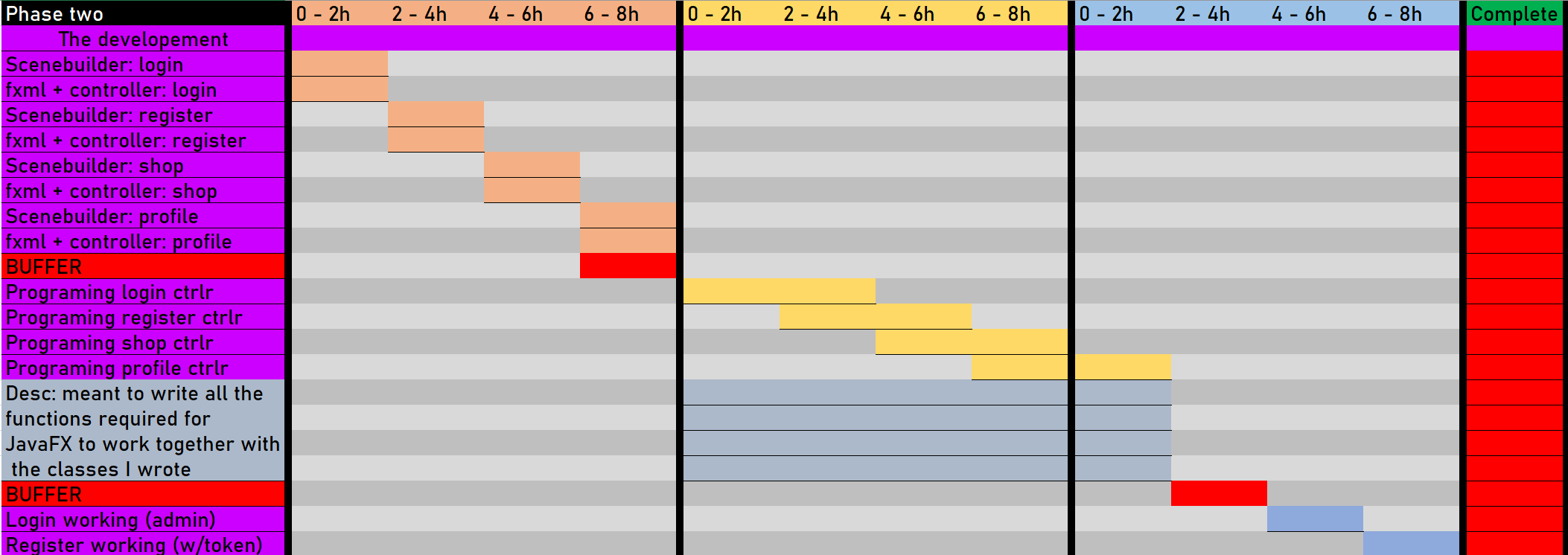
# Workplan (Accuracy: 2h):

## The Gantt-Diagram

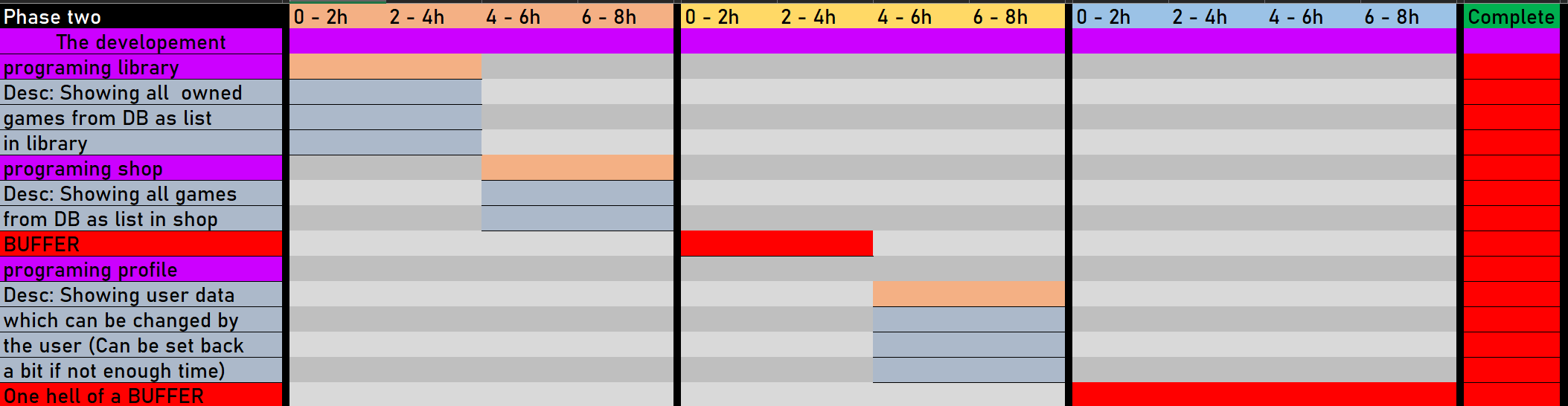
### First week:



### Second week



### Third week



### Fourth week

