

# Ideation Document for Individual Assignment

## Final Version

**Tsvetislav Rangelov**

**Student Number:4401336**

- **Git Repository link:**

<https://git.fhict.nl/I478554/S2-individual-project.git>

- **Description**

The main theme of the project is a digital manga library, containing titles that the administrator adds via the extension application(WinForms), which is then displayed as the website's contents(.NET Core razor-pages). The main functionality is the 4 CRUD operations which revolve around manipulating the content on the website. Additionally, standardized techniques for state management, security and user control have been implemented as auxiliary features, which help provide better UX.

- **Website**

The website's design is very minimalistic and clean, with no unnecessary clutter and misplaced elements. The workflow is determined from whether the user is logged in or not. In the case that they are, there are 2 possibilities in terms of experience.

1. Normal User (no admin privileges) –

This is the type that every newly registered user is. This group of users can only view the content of the website, only to the pages that are accessible to them normally and this is the target type that the website is mainly made for. The operations that they can do are the following: See an overview of all the content via the library page, view a certain item and details about it, add it to their personal list of items, which is located in the user profile and remove an item that is already in their user list. Any removed item from their personal list will once again show up as available in the library page and can thus be added later on again, if the user so desires. The user can also upload their own profile picture and change the default one, which is provided to them upon their first time logging in.

2. Administrator User (has admin privileges) –

The administrator is a pre-registered user, whose credentials are fed straight to the database with no form required on the website. They have all of the normal user's privileges and can also do all of the operations of the normal user, however with a few additional possibilities. Namely, once an administrator views a certain item, they have the option to also completely delete that item from the website, which in turn also removes it from any user profile as well. Another option for them, is to update a selected item, which redirects the administrator to a page only accessible to them. Via this page, an item's data can be partially or fully modified. The form which allows the user to do so is also error handled, so no empty data or invalid data can be assigned to the object. The administrator also has exclusive rights of access to the WinForms extension app, which will be explained in the next part of this document.

The auxiliary features of the website are mainly related to security and error handling.

To summarize, I will enumerate them here and am available to questions regarding any of them:

- Input validation on all forms, for all types of users
- Password hashing and salting algorithms.
- State management via sessions and cookies.
- Error handling a logged out user trying to access a page with the back button on the browser. This essentially means that the session is checked each time any page is loaded, and if the session key is invalid or null, a redirect to a 404 not found page is returned.
- Full website responsiveness utilizing CSS media queries and also some html manipulation via the Razor View with the razor tag - @.
- Every component, or in this case, class, is tested to ensure happy flow and expected or encountered error situations at runtime. Since for this project I use a many-to-many relationship table in my database to handle multiple users having multiple items in their profile, I also implemented a simulated many-to-many relationship in my unit tests, which consists of 2 arrays which hold the user and manga data correspondingly and a List of key-value pairs (not the dictionary data structure) that hold the actual relationship between the 2 arrays. I used a List of key-value pairs so that the keys can be duplicates, which is not possible in the case of a Dictionary. Dependency Injection is of course, also applied to decouple the logic from the data classes and ensure a safe testing environment.

- **WinForms Extension**

The WinForms application is, as mentioned earlier, an Administrator only app. Its main purpose is for the admin to have a designated place to control database contents, such as details about certain manga in the website, revoke administrator privileges from another user or promote them to administrator status and also removing user accounts. The application also provides an additional entity, which is TV Series, as per the assignment's requirements, which also can only be controlled from the WinForms app. All of the CRUD functionality present in the website is also available to the administrator in the WinForms, with some additional functionality, like creating a whole new item, be it Series or Manga. The app is also completely integrated with the database and the core architecture itself, sharing the data fed from the database and also utilizing product and user logic using class libraries. This also applies to the website as well.

- **Final Thoughts**

This project ignited the inspiration and curiosity for me to start really investing a lot of time into learning how to program, both theoretically and applying the theory itself. Now, nearing its end, I have started learning Angular, due to the fact that I really began to enjoy web development and I wanted to learn a JavaScript framework. Thank you to both Tim and André.