

Author: Diana Ciuperca

Date: 28/02/2022

Git repository: <https://git.fhict.nl/I505708/individual-assignment-semester-2.git>



Individual project

User requirement specifications

Table of Contents

Introduction	1
Functional Requirements	1
Non- functional Requirements	2
Use Cases – Desktop application	2

Introduction

The purpose of this User Requirement Specifications (URS) document is to outline the requirements for the development of an anime tracker application. The application will allow users to keep track of the anime they are watching, see detailed descriptions and receive notifications for new releases. They are able to create an account in order to discover new anime based on their preferences and customize their own profiles.

Functional Requirements

FR01: User should be able to log in to his account.

Account must already be in the system.

FR02: User should be able to add anime content.

FR03: User should be able to add manga content.

FR04: User should be able to add characters.

FR05: User should be able to see all the content that was added.

Non-functional requirements

NFR01 – Performance: The application shall be able to handle a large number of users and requests without significant slowdown or downtime. The application shall have a fast response time, with minimal latency and page load times, and be optimized for performance on both high-end and low-end devices.

NFR02 – Usability: The application shall have a user-friendly interface that is easy to navigate and understand for both novice and experienced users.

NFR03 – Reliability: The application shall be reliable, with minimal crashes or errors, and have a backup system in place to prevent data loss.

Use Cases - Desktop application

Use Case 1: Log into the application

Actor: Maintainer

Main Success Scenario:

1. Actor opens the application.
2. The system displays the *Login Page*.
3. Actor fills in their private details.
4. Actor confirms their details to proceed to their account.
5. The system opens the managing tab for the maintainers.

Extensions:

4a: Actor confirms the login without completing all the mandatory fields.

1. The system informs the actor that the data is wrong or the account doesn't exist.
2. The actor returns to MSS 3.

Use case 2: Add anime to the system

Actor: Maintainer

Prerequisites:

- Application is already open
- Actor is already logged in

Main Success Scenario:

1. Actor goes to the tab *Anime*.
2. The system displays the page to add new anime.
3. Actor fills in details about the new anime.
4. Actor confirms the details of the new anime's addition.
5. The system adds the anime to the database.
6. The system confirms to the actor the data is valid.

Extensions:

4a: Actor confirms the addition of new anime without completing all the mandatory fields or the data was in the wrong format.

1. The system cancels the addition and displays to the actor that mandatory fields are missing or are incorrect.
2. The actor returns to MSS 3.

Use case 3: Add manga to the system

Actor: Maintainer

Prerequisites:

- Application is already open
- Actor is already logged in

Main Success Scenario:

1. Actor goes to the tab *Manga*.
2. The system displays the page to add new manga.
3. Actor fills in details about the new manga.
4. Actor confirms the details of the new manga's addition.
5. The system adds the manga to the database.
6. The system confirms to the actor the data is valid.

Extensions:

4a: Actor confirms the addition of new manga without completing all the mandatory fields or the data was in the wrong format.

1. The system cancels the addition and displays to the actor that mandatory fields are missing or are incorrect.
2. The actor returns to MSS 2.

Use case 4: Add character to the system

Actor: Maintainer

Prerequisites:

- Application is already open
- Actor is already logged in

Main Success Scenario:

1. Actor goes to the tab *Characters*.
2. The system displays the page to add new characters.
3. Actor fills in details about the new character.
4. Actor connects the character to an already existing manga or anime.
5. Actor confirms the new character's addition.
6. The system confirms to the actor the data was valid.

Extensions:

5a: Actor confirms the addition of new character without completing all the mandatory fields or the data was in the wrong format.

1. The system cancels the addition and displays to the actor that mandatory fields are missing or are incorrect.
2. The actor returns to MSS 3.

5b: Actor confirms the addition of new character without connecting it to an already existing manga or anime.

1. The system cancels the addition and displays to the actor that a connection is mandatory.
2. The actor returns to MSS 3.

Use case 5: Display content

Actor: Maintainer

Prerequisites:

- Application is already open
- Actor is already logged in
- There is already data saved

Main Success Scenario:

1. Actor goes to the tab *All content*.
2. Actor filters the content by anime.
3. The system displays all anime content that was saved.
4. Actor filters the content by manga.
5. The system displays all manga content that was saved.
6. Actor filters the content by characters.
7. The system displays all characters content that were saved.

Extensions:

3a: There is no anime content saved.

1. The system displays that no anime content is saved.

5a: There is no manga content saved.

1. The system displays that no manga content is saved.

7a: There are no characters saved.

1. The system displays that there are no characters saved.