

User Requirements Specification (URS) for RSS Hamster



| | | |
|---------|---|------------|
| Date | : | 16/05/2024 |
| Version | : | 2.0 |
| State | : | Finished |
| Author | : | Nuno Dias |

Version history

| Version | Date | Author(s) | Changes | State |
|---------|------------|-----------|--|----------|
| 1.0 | 04-04-2024 | Nuno Dias | First Draft | Finished |
| 2.0 | 16-05-2024 | Nuno Dias | Expanded on original doc Isolated Use Cases to other doc Corrected grammar | Finished |

Glossary

| | |
|-----------|---|
| RSS | Really Simple Syndication |
| Algorithm | Coded Instructions that create a desired effect |

Introduction

The User Requirements Specifications (URS) outlines the functional and non-functional requirements for the development of a project, an RSS Feed aggregator and viewer. These will serve to guide the project so that both the clients requirements and expectations are met.

The project itself will be a tool to allow individuals to centralise their news, entertainment and other website notifications so they don't need to visit thirty different websites to keep up with the web-o-verse.

The project will include a WinForms application, mainly for use by Admins to help them manage and moderate the content, as well as a WebApp to be deployed to the Web for any user to use. These requirements will cover both Applications in one way or another.

Functional Requirements

FR.1 Feeds

- FR. 1.1. The system will allow administrators to create, view, update, and delete Feeds from the database (CRUD).
- FR. 1.2. By selecting a Feed a User will be able to view its comments, ratings and information that might help him decide if he is interested in following the feed or not. These include, but are not limited to, categories, popularity and similarity to content the user enjoys (determined by an algorithm).
- FR. 1.3. There will be a record of all removed feeds.

FR.2 Users

- FR. 2.1. The system will allow users to register, view their profile, update information, and delete their account (CRUD).
- FR. 2.2. Users will be able to follow Feeds they want to keep up with and add Feeds at their discretion.
- FR. 2.3. Users will have a simple Profile they can view and edit

FR.3 Admins

- FR. 3.1. Admins will be able to create and delete other admins accounts (CRUD).
- FR. 3.2. Admins will be able to view and manage all Feeds and their details.
- FR. 3.3. Admins will be able to view all Users and manage them.
- FR. 3.4. Admins will be able to hide or remove certain feeds.

FR.4 Search and Filters

- FR. 4.1. Users will be able to search all feeds
- FR. 4.2. Users will be able to select a subset of Feeds based on filters such as, but not limited to, categories, algorithmic recommendations, ratings and popularity

FR.5 Algorithms

- FR. 5.1. Users will be able receive a recommendation of Feeds based on their past activities
- FR. 5.2. There will be multiple Algorithms to select from and therefore lists of recommendations for the user to browse through.

FR.6 Reporting and Analytics

- FR. 6.1. The system will generate analytical reports for admins.
- FR. 6.2. Reports will include feed popularity, overall user activity, banned users and more.
- FR. 6.3. The system will support export capabilities for sharing reports in various formats (e.g. PDF, JSON).

Non-Functional Requirements

NFR.1 User Friendly

- NFR.1. The system will have an intuitive and user-friendly interface to facilitate ease of use and navigation for administrators and users.
- NFR.2. Features such as search filters, sorting options, and contextual help will be provided to make the experience more intuitive.

NFR.2 Scalability

- NFR.1. The system architecture will be designed to accommodate future growth and scalability requirements, including increased data volume and user concurrency.
- NFR.2. Modular and extensible code will be utilised to support future updates and swapping of modules with minimal effect on the rest of the code.

NFR.3 Reliability

- NFR.1. Through error handling
- NFR.2. Multiple data validation check at multiple points through the program

NFR.4 Security

- NFR.1. The system will implement role-based access control to control sensitive information and functionalities based on user roles. This will help the App restrict unauthorised access and ensure that users will only have access to the corresponding resources.
- NFR.2. The system will implement user authentication mechanisms (e.g., username/password, multi-factor authentication). This will help to verify the identity of users and add another layer of security to the system.

NFR.5 Performance

- NFR.1. The system shall be capable of handling concurrent user interactions and database transactions without significant performance loss.
- NFR.2. Response times for common operations (e.g., record retrieval, profile updates) should be within acceptable limits.

Use Cases

Web Application

Use Cases relating to use of the Web Application.

Mainly relate to non-Admin Users.

Use Case 1: User Registration

Use Case ID: UC-001

Description: Allow users to register a new account and log in to the application.

Primary Actor: User

Preconditions: - User has access to the Web Application.

Postconditions: - User is registered.

Main Flow:

1. User navigates to the register page.
2. User enters required information (name, username, email, password).
3. User submits the registration form.
4. System validates the information and creates a new account.
5. System sends user to login page.

Alternate Flow:

3a. If the email is already in use, the system prompts the user to use a different email.

3b. If the password does not meet security requirements, the system prompts the user to choose a stronger password.

Use Case 2: User Login

Use Case ID: UC-002

Description: Allow users to log in to the application.

Primary Actor: User

Preconditions: - User has access to the Web Application.

Postconditions: - User is logged in.

Main Flow:

1. User navigates to the login page.
2. User enters required information (email, password).
3. User submits the login form.
4. System validates the information and logs in with the account.

Alternate Flow:

4a. If the email does not match any existing account, the system prompts the user to use a different email.

4b. If the email matches an account but the password does not, the system prompts the user to try again.

Use Case 3: Search for Feeds

Use Case ID: UC-003

Description: Allow users to search for feeds by name.

Primary Actor: User

Preconditions: - User has access to the Web Application.

Postconditions: - Search results are displayed based on the user's input.

Main Flow:

1. User enters feed name (or part there off) in search bar.
2. User submits the search.
3. System processes the search and retrieves matching feeds.
4. System displays the search results.

Alternate Flow:

3a. If no feeds match the criteria, the system displays a “no results found” message.

Use Case 4: Filter Feeds

Use Case ID: UC-004

Description: Allow users to filter searches.

Primary Actor: User

Preconditions: - User has access to the Web Application.

Postconditions: - Search results are displayed based on the user’s input.

Main Flow:

1. User navigates to ‘All Feeds’ page.
2. User selects a number of filters from available options.
3. User submits the search.
4. System processes the search and retrieves matching feeds.
5. System displays the search results.

Alternate Flow:

3a. If no feeds match the criteria, the system displays a “no results found” message.

Use Case 5: Generate Recommendations

Use Case ID: UC-005

Description: Give Users a list of recommended feeds.

Primary Actor: User

Preconditions: - User is logged in.

Postconditions: - Recommended results are displayed based on user data.

Main Flow:

1. User navigates to ‘All Feeds’ page.

2. User selects 'Generate Recommendations'.
3. System processes the users data and retrieves a list of feeds.
4. System displays the recommended feeds results.

Alternate Flow:

3a. If the system does not have enough information on the user a "not enough data" message is displayed.

Use Case 6: View Feed Details

Use Case ID: UC-006

Description: Allow users to view detailed information about a specific feed.

Primary Actor: User

Preconditions: - User has performed a search or navigated through feed listings.

Postconditions: - Detailed feed information is displayed.

Main Flow:

1. User selects a feed from the search results or listings.
 2. System retrieves detailed information about the selected feed.
 3. System displays the feed details (title, last updated date, items, additional content, comments, reviews).
-

Use Case 7: Write and Rate a Feed

Use Case ID: UC-007

Description: Allow users to write reviews and rate feeds.

Primary Actor: User

Preconditions: - User is logged in. - User is viewing the details of a feed.

Postconditions: - The review and rating are saved and displayed.

Main Flow:

1. User navigates to the review section of the feed details page.
2. User writes a review and selects a rating.

3. User submits the review and rating.
4. System saves the review and rating.
5. System updates the feed's overall rating based on the new submission.
6. System displays the new review and rating.

Alternate Flow:

3a. If the review content is inappropriate, the system flags the review and prompts the user to revise it.

WinForms Application

Use Cases relating to use of the WinForms Application.

Mainly relate to Admin Users.

Use Case 8: User Login

Use Case ID: UC-008

Description: Allow users to log in to the application.

Primary Actor: User

Preconditions: - User has access to the WinForms Application.

Postconditions: - User is logged in.

Main Flow:

1. User enters required information (email, password).
2. User submits the login form.
3. System validates the information and logs in with the account.

Alternate Flow:

3a. If the email does not match any existing account, the system prompts the user to use a different email.

3b. If the email matches an account but the password does not, the system prompts the user to try again.

Use Case 9: Manage Feed Database (Admin)

Use Case ID: UC-009

Description: Allow administrators to manage the feed database.

Primary Actor: Admin

Preconditions: - Admin is logged in.

Postconditions: - Changes to the feed database are saved and updated.

Main Flow:

1. Admin logs in and navigates to the feeds form.
2. Admin selects an option to add, edit, or delete a feed.
3. Admin enters or updates feed information (title, director, cast, synopsis, release date, genre).
4. Admin submits the changes.
5. System validates and saves the changes.
6. System updates the feed database and displays a confirmation message.

Alternate Flow:

4a. If the feed information is incomplete or incorrect, the system prompts the admin to correct the errors.

Use Case 10: Manage User Accounts (Admin)

Use Case ID: UC-010

Description: Allow administrators to manage user accounts.

Primary Actor: Admin

Preconditions: - Admin is logged in.

Postconditions: - User accounts are updated as necessary.

Main Flow:

1. Admin logs in and navigates to the user management section.
2. Admin selects an option to view, edit, or delete a user account.
3. Admin makes the necessary changes (e.g., reset password, update email, deactivate account).
4. Admin submits the changes.
5. System validates and saves the changes.
6. System displays a confirmation message.

Alternate Flow: 4a. If the changes are invalid (e.g., email already in use), the system prompts the admin to correct the errors.

Use Case 11: Moderate User Content (Admin)

Use Case ID: UC-011

Description: Allow administrators to moderate user reviews and ratings.

Primary Actor: Admin

Preconditions: - Admin is logged in.

Postconditions: - User-generated content is moderated and updated.

Main Flow:

1. Admin logs in and navigates to the reviews section.
 2. Admin reviews content.
 3. Admin decides to edit or delete the content.
 4. Admin submits the moderation decision.
 5. System updates the status of the content based on the admin's decision.
 6. System displays a confirmation message.
-