Project Analysis

Table of Contents

[Functional and Non-functional requirements 3](#_Toc192598631)

[Functional Requirements 3](#_Toc192598632)

[Non-functional requirements 3](#_Toc192598633)

[Use Cases 4](#_Toc192598634)

[Use Case: User makes an account 4](#_Toc192598635)

[Interests: 4](#_Toc192598636)

[Preconditions: 4](#_Toc192598637)

[Guarantees: 4](#_Toc192598638)

[Main success scenario: 4](#_Toc192598639)

[Extensions: 4](#_Toc192598640)

[Conceptual Diagram 5](#_Toc192598641)

# Functional and Non-functional requirements

For this project the functional and non-functional requirements go as follows:

## Functional Requirements

* The user can log into the website,
* The user can view movies on an overview setting in the categories of: Watched, Not Watched in List and Currently Watching (Series),
* Series will feature episode tracking
* The user can add, and view movies called The List. Users cannot add movies that have been added already or is on the category of Watched,
* The user can add themselves to the attendance list of each week,
* Users can add option of “High interest” on 1 additional movie as it will affect attendance and thus removing the movie from the pool if no attendance is given by said user,
* The admin user is assigned to one user only called the Projectionist,
* Projectionist is the only user who can draw the movies and possibly change the drawing pool list as needed,
* The drawing pool list is edited as attendance changes. It means movies added by certain users will not be drawn if said user does not give attendance,
* The website will feature a wheel of fortune that gets the movies from the pool and selects a movie at random
* The drawn movie is added to the Watched list at the order of the Projectionist

## Non-functional requirements

* Website will be hosted and accessible,
* Website will feature sessions for weekly draws of movies,
* Website will store necessary data in its provided database,
* Website is secure, unauthorized users mustn’t be allowed into the website,
* Website is responsive and optimized,
* Website must not allow invalid input into the database.

# Use Cases

## Use Case: User makes an account

* Primary Actor: User
* Scope: Movie night website
* Level: User goal

### Stakeholder & Interests:

* User wants to make an account
* System adds User and makes sure no duplicates are made

### Preconditions:

* Users must provide username and password

### Guarantees:

* System adds the user
* System adds unique ID to user
* System prevents duplicate usernames

### Main success scenario:

* User presses “Join in” Button
* User is presented with “Provide Username and Password” form
* User fills the simple form
* User presses “Create account”
* System confirms account is created
* User is prompted to log in

### Extensions:

#### User attempts to log in without an account

* The system will prompt users to create an account.

#### User does not fill in the form correctly

* System highlights the fields that need to be corrected

#### User already created an account previously

* System asks users to log in instead

#### Username is not available

* System asks users to provide a different username or log in

## Use Case: User Logs in

* Primary Actor: User
* Scope: Movie night website
* Level: User goal

### Stakeholder & Interests:

* User logs into the movie night website and has access to personal content
* System authenticates user and ensures security

### Preconditions:

* Users must have an account

### Guarantees:

* System creates a session for the user
* System remembers and differentiates between different users
* System shows the correct information corresponding to logged in user

### Main success scenario:

* User presses “Log in” button
* System provides log in form
* User fills the simple form
* User presses “Log in”
* System validates the username and password
* System confirms User is logged in briefly
* System redirects user to “Overview” page of movie night website

### Extensions:

#### User attempts to log in without an account

* The system will prompt users to create an account.

#### User does not fill in the form correctly

* System highlights the fields that need to be corrected.

#### User inputs wrong password

* System asks user to retry.

#### Username does not exist

* System informs user that username does not exist and suggests user to create an account or retry.

# Conceptual Diagram