Project Analysis

Jéssica Helena Soares de Pina

Fontys University of Applied Sciences

A diagram of a movie

AI-generated content may be incorrect.

Image 1: Funny wrong conceptual model

Table of Contents

[Functional and Non-functional requirements 4](#_Toc193493119)

[Functional Requirements 4](#_Toc193493120)

[Non-functional requirements 4](#_Toc193493121)

[Use Cases 6](#_Toc193493122)

[Use Case: User makes an account 6](#_Toc193493123)

[Stakeholder & Interests: 6](#_Toc193493124)

[Preconditions: 6](#_Toc193493125)

[Guarantees: 6](#_Toc193493126)

[Main success scenario: 6](#_Toc193493127)

[Extensions: 6](#_Toc193493128)

[Use Case: User Logs in 7](#_Toc193493129)

[Stakeholder & Interests: 7](#_Toc193493130)

[Preconditions: 7](#_Toc193493131)

[Guarantees: 7](#_Toc193493132)

[Main success scenario: 7](#_Toc193493133)

[Extensions: 7](#_Toc193493134)

[Use Case: User Logs out 8](#_Toc193493135)

[Stakeholder & Interests: 8](#_Toc193493136)

[Preconditions: 8](#_Toc193493137)

[Guarantees: 8](#_Toc193493138)

[Main success scenario: 8](#_Toc193493139)

[Extensions: 8](#_Toc193493140)

[Use Case: User adds a movie to the suggestion pool 9](#_Toc193493141)

[Stakeholder & Interests: 9](#_Toc193493142)

[Preconditions: 9](#_Toc193493143)

[Guarantees: 9](#_Toc193493144)

[Main success scenario: 9](#_Toc193493145)

[Extensions: 9](#_Toc193493146)

[Use Case: User votes for the movie to be in the list 10](#_Toc193493147)

[Stakeholder & Interests: 10](#_Toc193493148)

[Preconditions: 10](#_Toc193493149)

[Guarantees: 10](#_Toc193493150)

[Main success scenario: 10](#_Toc193493151)

[Extensions: 10](#_Toc193493152)

[Use Case: User views The List 12](#_Toc193493153)

[Stakeholder & Interests: 12](#_Toc193493154)

[Preconditions: 12](#_Toc193493155)

[Guarantees: 12](#_Toc193493156)

[Main success scenario: 12](#_Toc193493157)

[Extensions: 12](#_Toc193493158)

[Use Case: User confirms attendance 13](#_Toc193493159)

[Stakeholder & Interests: 13](#_Toc193493160)

[Preconditions: 13](#_Toc193493161)

[Guarantees: 13](#_Toc193493162)

[Main success scenario: 13](#_Toc193493163)

[Extensions: 13](#_Toc193493164)

[Use Case: User views library 14](#_Toc193493165)

[Stakeholder & Interests: 14](#_Toc193493166)

[Preconditions: 14](#_Toc193493167)

[Guarantees: 14](#_Toc193493168)

[Main success scenario: 14](#_Toc193493169)

[Extensions: 14](#_Toc193493170)

[Use Case: Admin user draws Movie 15](#_Toc193493171)

[Stakeholder & Interests: 15](#_Toc193493172)

[Preconditions: 15](#_Toc193493173)

[Guarantees: 15](#_Toc193493174)

[Main success scenario: 15](#_Toc193493175)

[Extensions: 15](#_Toc193493176)

[Conceptual Diagram 16](#_Toc193493177)

[Acceptance Testing (Incomplete) 17](#_Toc193493178)

# 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 one 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: Movie watcher
* 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: Movie watcher
* 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.

## Use Case: User Logs out

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

### Stakeholder & Interests:

* User logs out the movie night website
* System ensures session and user data are not lost in the process

### Preconditions:

* Users must have an account
* Users must be logged in

### Guarantees:

* System stores data of user in database
* System secures data of user

### Main success scenario:

* User presses “Log out” button
* System confirms log out is successful
* System redirects user to “Log in” page of movie night website

### Extensions:

#### No idea what could go wrong here really?

## Use Case: User adds a movie to the suggestion pool

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

### Stakeholder & Interests:

* User adds a movie to the list as a contender for the weekly draws
* System adds movie to the contender list successfully
* System ensures valid data is added to the list
* System prevents duplication

### Preconditions:

* User must have an account
* User must be logged in
* Movie is not yet added to The List

### Guarantees:

* System adds movie to the list
* System remembers which user added the movie to the contender list

### Main success scenario:

* User presses “Add movie” button
* System displays add movie page and form
* User fills in the form correctly
* User presses “Add movie!”
* System validates the data and adds the movie to suggestion pool
* System ensures suggestion is not a duplicate
* System confirms the movie is added successfully
* System refreshes page

### Extensions:

#### User attempts to add a movie already in The List or the contender list

* The system will inform user that the movie is already in “The List” or the contender list

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

* System highlights the fields that need to be corrected.

#### System fails to add movie

* Display error message

## Use Case: User votes for the movie to be in the list

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

### Stakeholder & Interests:

* User votes for the movie to be in the weekly drawn list
* System ensures votes are sent through
* System ensures the correct movies are displayed

### Preconditions:

* Users must be logged in

### Guarantees:

* System adds movie to The List
* System remembers votes and vetoes of other users
* System prevents user from voting on their own suggestions
* System prevents double votes from the same user on the same movie

### Main success scenario:

* User presses “Vote” button
* System provides overview of movies, with trailer, that are in the contender list
* User votes or vetoes said movie
* System registers user vote or veto
* System checks if movie is added by same user
* System remembers that user voted on the movie
* System confirms to user that the vote has been added
* System removes movies in which user has already voted
* System refreshes page

### Extensions:

#### System displays movies in which user has voted

* Buttons will be disabled for user. If voting was attempted error message will pop up saying “Vote already added”

#### System displays movies that user added themselves

* Buttons should be disabled for user. If voting is attempted, error message will pop up with “You added this yourself”

#### User attempts to vote without watching the trailer

* Pop up message with “Are you sure” will appear

#### User cannot add a vote

* System error “Movie not in vote list”
* Page reloads with updated list

## Use Case: User views The List

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

### Stakeholder & Interests:

* User has an overview of three categories of the list: Watched, not watched and currently watching
* System ensures the data is separated correctly

### Preconditions:

* Users must be logged in

### Guarantees:

* System displays data correctly

### Main success scenario:

* User presses “Overview” button
* System provides overview page of The List
* System provides the categories and separates data correctly

### Extensions:

#### System fails to display data

* Error message to inform there has been a problem
* Reload page

#### System blocks user from viewing The List

* System prompts user to log in with a textbox
* System redirects user to log in page

## Use Case: User confirms attendance

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

### Stakeholder & Interests:

* User confirms attendance to the movie night before the movie night
* System confirms attendance and adds movies added by specific user to the drawing pool
* System ensures the correct movies are added

### Preconditions:

* Users must be logged in

### Guarantees:

* System adds movie to the drawing pool
* System remembers which users will be on said movie night
* System shows which movies were added by which user

### Main success scenario:

* User presses “Calendar” button
* System calendar view of the next events
* User presses button “Confirm attendance”
* System confirms user attendance
* System adds movies added by user into the drawing pool
* System ensures movies are not in “Already Watched” or “Currently watching” lists
* System shows to the user operation is successful
* System displays which movies from the user were added to the pool
* System refreshes page

### Extensions:

#### System fails to confirm attendance

* Error message will pop up
* System asks user to try again

#### System fails to add movies to the pool

* Error message will pop up

## Use Case: User views library

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

### Stakeholder & Interests:

* User views all movies, series etc. that have been watched in movie nights
* System ensures display is correct and data is organized

### Preconditions:

* Users must be logged in

### Guarantees:

* System ensures data is well organized

### Main success scenario:

* User presses “Library” button
* System displays well organized page with movies
* System allows for user to press on movies for more details

### Extensions:

#### System does not show any movie

* Error message will pop up

#### System does not allow for more details to be displayed

* Error message will pop up

## Use Case: Admin user draws Movie

* Primary Actor: Projectionist
* Scope: Movie night website
* Level: Admin goal

### Stakeholder & Interests:

* Administrator user with proper permissions chooses a movie at random for the movie night.

### Preconditions:

* User is an administrator with proper permissions

### Guarantees:

* System ensures event can only happen on a Thursday,
* System ensures movies in the rolling pool are movies that movie watchers have marked attendance for,
* System ensures, upon confirmation, that the movie is added to the Watched list.

### Main success scenario:

* Administrator user “Projectionist” presses the “Roll!” button,
* System displays wheel of fortune with correct movies,
* Administrator, after checking the correctness of the pool, presses button to spin the wheel,
* System displays winner movie with confetti animation and audience clapping.

### Extensions:

#### Movie watcher attempts to spin the wheel

* Error message “Permission denied”.

#### No movies are displayed in the wheel of fortune

* Error message “No movies in Wheel” is displayed,
* Administrator will be able to check the list of movies for mistakes,
* Optionally Administrator can manually add the movies.

#### Wrong or already watched movies are displayed

* Warning will be displayed with “Roll again for x movie?”.

#### Administrator user cannot spin the wheel

* Error message “Permission denied” pops up.

# Conceptual Diagram

# A diagram of a diagram AI-generated content may be incorrect.

# Acceptance Testing (Incomplete)

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST TITLE** | | | | **PRIORITY** | **TEST CASE ID** | | | **TEST NUMBER** | | | | **TEST DATE** |
| Create Account | | | | 1 | 01 | | | 01 | | | | - |
| **TEST DESCRIPTION** | | | | | **TEST DESIGNED BY** | | | **TEST EXECUTED BY** | | | | **EXECUTION DATE** |
| User creates an account | | | | | Jéssica | | |  | | | | - |
|  | | | | | | | | | | | | |
| **TEST DESCRIPTION** | | | **TEST DEPENDENCIES** | | | **TEST CONDITIONS** | | | | **TEST CONTROL** | | |
| User creates an account | | |  | | |  | | | |  | | |
|  | | | | | | | | | | | | |
| **STEP ID** | **STEP DESCRIPTION** | **TEST DATE** | | **EXPECTED RESULTS** | | | **ACTUAL RESULTS** | | **PASS / FAIL** | | **ADDITIONAL NOTES** | |
| 1 | Press on “Sign up” |  | | Opens form to sign up | | |  | |  | |  | |
| 2 | Click on “Create username” field |  | | User creates a username for themselves | | |  | |  | |  | |
| 3 | Click on “Create Password” field |  | | User creates password | | |  | |  | |  | |
| 4 | Press on “Done” |  | | System confirms user is created. User can now log in. | | |  | |  | |  | |
| **TEST TITLE** | | | | | **PRIORITY** | **TEST CASE ID** | | | **TEST NUMBER** | | | | **TEST DATE** |
| Adding a movie | | | | | 1 | 03 | | | 03 | | | | - |
| **TEST DESCRIPTION** | | | | | | **TEST DESIGNED BY** | | | **TEST EXECUTED BY** | | | | **EXECUTION DATE** |
| User adding a movie | | | | | | Jéssica | | |  | | | | - |
|  | | | | | | | | | | | | | |
| **TEST DESCRIPTION** | | | | **TEST DEPENDENCIES** | | | **TEST CONDITIONS** | | | | **TEST CONTROL** | | |
| User attempts to add a movie | | | |  | | | User must be logged in | | | |  | | |
|  | | | | | | | | | | | | | |
| **STEP ID** | | **STEP DESCRIPTION** | **TEST DATE** | | **EXPECTED RESULTS** | | | **ACTUAL RESULTS** | | **PASS / FAIL** | | **ADDITIONAL NOTES** | |
| 1 | | Press on “Add Movie” |  | | Opens form | | |  | |  | |  | |
| 2 | | Fill in the form to add movie |  | | All fields are responsive | | |  | |  | |  | |
| 3 | | Press “Submit” |  | | System confirms through a window the submit is successful | | |  | |  | |  | |
|  | |  |  | |  | | |  | |  | |  | |