Use Cases

**FilmSo**

### A platform that allows you to keep track of all the movies you have seen and/or want to see

### **Ileana Aguilar**

**Contents**

[General characteristics](#_wvmzsykhhn51) 3

[Admission](#_oy6txkoxs3ru) 4

[Purpose of the document](#_qao6tnsm4bfg) 4

[Document scope](#_idrvil2ytkhb) 4

[Characteristics of the document](#_l14oplbif78z) 4

[Identified use cases](#_azzeavfb3604) 4

[2.1 Use cases](#_gi5ix2xtn6z0) 5

[2.1.1 Use Case 1: System Administration - Administrator](#_r9tnxf4tbbwh) 5

[2.1.1.1 Use Case 1.1: Manage a user’s profile](#_hrmk97a2yz0r) 5

[2.1.1.1.1 User’s security profile](#_f915jgqfxfkl) 5

[2.1.1.1.2 Authentication and Authorization](#_q9l3lqxlhpfd) 6

[2.1.1.2.1 Add a film record to the database](#_892l4b4azukj) 6

[2.1.1.2.2 Delete a record from the database](#_j9hbqmyifklz) 6

[2.1.1.2.3 Update a record from the database](#_6hyr46h8nyt4) 6

[2.1.1.2.4 Show a record](#_lpojp21kgl1k) 6

[2.1.2 Use Case 2: System User](#_sbg6auas9ijk) 6

[2.1.2.1 Use Case 2.1: Adding a film to user profile](#_4k47qm6yxobt) 6

[2.1.1.2.1 Rate a film](#_og4as7hh0k8q) 7

[2.1.1.2.2 Review or comment a film](#_8zbvizyiv1pz) 7

[2.1.1.2.3 Mark as watched or want to watch](#_djgy3nxp12jh) 7

[2.1.1.2.4 Delete a film](#_u11je7ps89ka) 7

[2.2 Users](#_b1g8dvmg0nlp) 8

[Table 1.](#_jzskg6tg518w) 8

[2.3 Business Entities](#_avizegwb7bqy) 8

[Table 2.](#_tmmj31m55v2x) 8

[3. Description of use cases](#_sllcx2jgkoln) 9

[3.1. Use Case 1: <System administration - administrator>](#_ep1ixqfpayt) 9

[3.1.1. Use case 1.1: <Manage user profile>](#_roud4ynnpvb) 9

[3.1.1. Use case 1.1: <Record data>](#_fbehc727p8ra) 10

[3.2. Use Case 2: <System user>](#_xqr4j6jfyge0) 13

[3.2.1. Use case 2.1: <Adding a film to user profile>](#_xjzgb6e1ywxy) 13

[3.2.2. Use case 2.2: <Sign up>](#_7p6dlmmm4unm) 16

[3.2.3. Use case 2.3: <Sign in>](#_d14klp8cv3yl) 17

[3.2.4. Use case 2.4: <Sign out>](#_zg8amjcg9qs7) 17

[3.2.5. Use case 2.5: <View statistics>](#_9s25sv2hyx4z) 18

[4. User interface](#_jla31ep0iipv) 19

[Bibliography:](#_ydnj3j9ihf0w) 22

#### 

#### **General characteristics**

FilmSo is an entertainment platform that helps users to keep a faithful record of all those movies that they have seen during their life or those that they will record over time as well as having a list of those that they want to see in the future.

The purpose of this app is for movie buffs to stop worrying about their movie viewing history and to feel confident that a great movie description is at hand.

The following document provides a functional specification for the proposed application in the form of a description of application use cases and the scenarios defined for them.

Proposed use cases will be used to design an application, create its implementation, and conduct a testing process. Defined scenarios for identified use cases will be used to illustrate the system’s interaction with its potential users.

### **Admission**

Use cases describe the behavior of the application/system and the sequence of events performed by the system or system actors that interact with it in order to perform a specific task.

#### Purpose of the document

The purpose of the following document is to classify and describe all use cases related to the proposed application.

#### Document scope

A complete set of use cases allows you to define the entire system. Use cases determine how the system will help an actor to accomplish a specific task or activity string.

A use case describes a situation where an actor uses the system in a typical way to achieve the intended result. A description of the use case helps you design the appropriate set of interactions to meet a specific task.

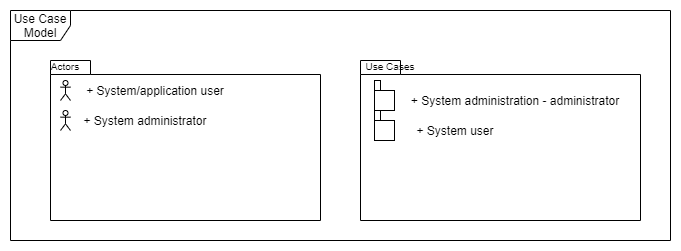
#### Characteristics of the document

The following document is divided into two main parts. The first one identifies the usage cases for the system you create by providing a brief description of each use case and lists the actors the system, along with their brief characteristics. The second part describes the scenarios for each identified use case - it contains a normal and alternative event flow, prerequisites and end conditions, and expected UI behavior.

### **Identified use cases**

The following part shows a conceptual model of use cases, consisting of a set of actors and their assigned use cases:

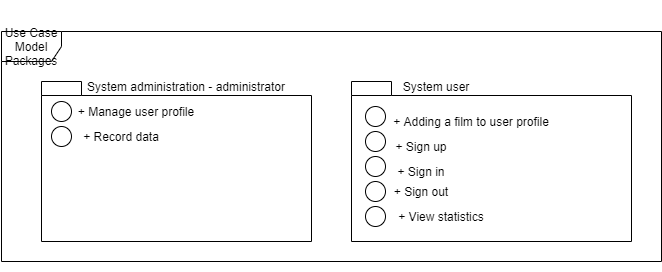
* The actor’s pack includes the following service actors:
  + System/application user.
  + System administrator.
* The “Use Cases” package contains service use cases grouped into two sub-packages:
  + System administration - administrator.
  + System user.



#### 2.1 Use cases

FilmSo identified two groups of use cases, grouped by two functional packages:

1. System administration - administrator.
2. System user.



#### 2.1.1 Use Case 1: System Administration - Administrator

The “Administrator” system administration package includes the use case responsible for managing access control at the security administrator level for an existing system.

#### 2.1.1.1 Use Case 1.1: Manage a user’s profile

Use case in which the administrator is responsible for collecting information from the user's account such as name, surname, email, and password. It allows you to link the registry, login, and logout. Also, the information that corresponds to each user with their movie records and statistics.

##### **2.1.1.1.1 User’s security profile**

Use case that allows you to manage a user’s safety profile by assigning role to the user, deleting account, defining security restrictions for the user.

##### **2.1.1.1.2 Authentication and Authorization**

Use case that allows you to log in, authenticate, and authorize the users.

2.1.1.2 Use Case 1.2: Record data

Use case where all information related to user records is accessed (declared below). The administrator uses this information to analyze and create statistics.

##### **2.1.1.2.1 Add a film record to the database**

Use case in which information is added/created with INSERT or stored in the database regarding the new publication or user registration. You can create as many records as you like.

##### **2.1.1.2.2 Delete a record from the database**

Use case where the information stored in the database is deleted from the user's registry with DELETE (destroy). You can delete the movie you have already watched or want to watch.

##### **2.1.1.2.3 Update a record from the database**

Use case where it is modified with UPDATE in the database. Related to the modification of the film registration. You can change the review, the rating, whether you liked it or not.

##### **2.1.1.2.4 Show a record**

Use case where the information stored in the database is read with SELECT (get). Only the records published by the user can be shown to the user.

#### 2.1.2 Use Case 2: System User

#### 2.1.2.1 Use Case 2.1: Adding a film to user profile

Use case that allows the user to put into practice our main functionality of the web application which is to add a movie to the system since the whole system is based on these movie records.

The user finds the button to add a movie to his record and goes to see all the information that he can add along with the name, year of release of the movie, among other information.

The user must confirm the action by clicking the add / save button to send the data to the database and confirm the action.

##### **2.1.1.2.1 Rate a film**

Use case where it is allowed to give a rating to the registered film based on a 5-star scale.

##### **2.1.1.2.2 Review or comment a film**

Use case in which the user can write a small text where he talks about the film, the plot, what he liked, what he didn't like. This is more personal and descriptive. It can be optional.

##### **2.1.1.2.3 Mark as watched or want to watch**

Use case where the icon is selected if the movie was watched or the watchlist icon is chosen, which means that the user has not yet seen the movie but wants to.

##### **2.1.1.2.4 Delete a film**

Use case that allows deleting the created movie record from the database. In case of being deleted, the user will stop seeing this specific record in his "diary" and if he wants to add it again he will have to do it manually.

2.1.2.2 Use Case 2.2: Sign up

Use case to allow the user to register in our database and thus give them permissions so that they can use FilmSo's functionalities. The user can create an account to be able to access all the features and with it to be able to have a unique registry per user since all the information is gathered without being confused.

The user registers his name, his email, and creates a password for his account. From here you can go to the next stage which is to start the session but without having to return to it at another time.

We must verify that the user does not exist to create a new one and if it exists, notify him that there is already a user with that email.

2.1.2.3 Use Case 2.3: Sign in

Use case where the user now, after registering, can enter the platform or service of our web application to view and use the functions. Here you should check that the user is actually registered in our database and make sure that the data matches. The login should be fast, with the least type and time of failures. The start screen and its functions will be displayed.

This is only available only when the user has previously created an account and the administrator can verify it.

2.1.2.3 Use Case 2.3: Sign out

Use case in which the user decides to end the session in which he is using our web application, so he exits his session and disconnects, ceasing to see his movie records and statistics. You need to log in again to be able to see them again.

The user must stop seeing the main interface of our website and start to see the template to login or create an account.

2.1.2.3 Use Case 2.3: View statistics

Use case where data analyzed by us is provided in which the user can corroborate which film genre he has seen the most based on all his records. You can also see his average rating of the total of them, as well as how many movies he has seen per week, per month and per year.

These statistics can only be seen in an external section to that of adding a film since the information collected is one that already has a record of them. The user will see an additional button in their profile to be able to view the statistics and also return to the beginning. Statistics cannot be modified by the user.

### **2.2 Users**

FilmSo identified two actors: system/application user and system administrator.

#### *Table 1.*

|  |  |
| --- | --- |
| System user | Description |
| System/application user | The system user represents the person using the application/system for the mere purpose of the app. They have access policies defined (user’s security profile). |
| System administrator | The security administrator represents the person responsible for managing the security of the IT system (or individual application), with a particular focus on access control management and usage control, also in control of the database. |

### **2.3 Business Entities**

FilmSo identifies one business unit: system user list.

#### *Table 2.*

|  |  |
| --- | --- |
| Business entity | Description |
| UserList | List of the system users for whom system access rights are defined - security profile. |

### **3. Description of use cases**

The following chapter describes the scenarios for identified FilmSo use cases.

#### **3.1. Use Case 1: <System administration - administrator>**

The “Administrator” system administration package includes use cases responsible for managing access control at the security administrator level for the system and managing the database.

#### 3.1.1. Use case 1.1: <Manage user profile>

Use case that allows you to manage the users profile with the data given by the user.

|  |
| --- |
| Use Case <User security profile> |
| Types: |
| Users |
| * System administrator * System/application user |
| Prerequisites: |
| * The user is listed as an application/system user - sign up, sign in and sign out |
| Typical step order: |
| 1. View the list of users wanting to sign up 2. Approving and give user role 3. Do role-role allocation <assign features to a role> 4. Definition of security restrictions for role-function relationships |
| Final conditions: |
| * The set of functions assigned to a role is a non-empty set |
| Alternate step order: |
|  |
| Comments: |
|  |

|  |
| --- |
| Use Case <Authentication and Authorization> |
| Types: |
| Users |
| * System administrator * System/application user |
| Prerequisites: |
| * The user is listed as an application/system user – you have an account on the system. |
| Typical step order: |
| 1. User sign in. 2. Confirm username, email and password between database and user input. 3. Confirm functions available depending on the role. |
| Final conditions: |
| * User sign in correctly to the system. |
| Alternate step order: |
|  |
| Comments: |
|  |

#### 

#### 3.1.1. Use case 1.1: <Record data>

A use case that allows you to apply the basic function of the model CRUD in the database.

|  |
| --- |
| Use Case <Add a film record to the database> |
| Types: |
| Users |
| * System administrator * System/application user |
| Prerequisites: |
| * The user is listed as an application/system user – you have an account on the system. * The user is signed in. |
| Typical step order: |
| 1. User does the process to add film. 2. Information added to database through INSERT method in a query. 3. Store data in the database. |
| Final conditions: |
| * User adds a new entry of a film. * Database updated. |
| Alternate step order: |
|  |
| Comments: |
|  |

|  |
| --- |
| Use Case <Delete a record from the database> |
| Types: |
| Users |
| * System administrator * System/application user |
| Prerequisites: |
| * The user is listed as an application/system user – you have an account on the system. * The user is signed in. |
| Typical step order: |
| 1. User does the process to delete film. 2. Information deleted from the database through DELETE(destroy) method in a query. 3. Remove data from the database. |
| Final conditions: |
| * User removes the entry of a film. * Database updated. |
| Alternate step order: |
|  |
| Comments: |
|  |

|  |
| --- |
| Use Case <Update a record from the database> |
| Types: |
| Users |
| * System administrator * System/application user |
| Prerequisites: |
| * The user is listed as an application/system user – you have an account on the system. * The user is signed in. |
| Typical step order: |
| 1. User does the process to update film. 2. Information updated from the database through UPDATE method in a query. 3. Modify data from the database. |
| Final conditions: |
| * User updates the entry of a film. * Database updated. |
| Alternate step order: |
|  |
| Comments: |
|  |

|  |
| --- |
| Use Case <Show a record> |
| Types: |
| Users |
| * System administrator * System/application user |
| Prerequisites: |
| * The user is listed as an application/system user – you have an account on the system. * The user is signed in. |
| Typical step order: |
| 1. User selects a film. 2. Information showed from the database through SELECT (get) method in a query. 3. Show data from the database. |
| Final conditions: |
| * User selects the entry of a film. * Data shown. |
| Alternate step order: |
|  |
| Comments: |
| The user can only see the records published by themselves. |

#### **3.2. Use Case 2: <System user>**

The “System user” package includes use cases that are responsible for managing how the user interacts with the FilmSo application.

#### 3.2.1. Use case 2.1: <Adding a film to user profile>

A use case that allows you to manage roles at the application designer level (local level). This should be in relation to other use cases:

|  |
| --- |
| Use Case <Rate a film> |
| Types: |
| Users |
| * System user |
| Prerequisites: |
| * There is a non-empty set of roles defined for the application. * There is a non-empty set of functions defined for the application. * The user is listed as an application/system user – you have an account on the system. * The user is signed in. |
| Typical step order: |
| 1. User clicks a button -add film- 2. Selects the number of stars to rate the film (in a scale of 0 to 5 stars) that he/she is adding. 3. Add data to the database. |
| Final conditions: |
| * The data about the rating is updated. |
| Alternate step order: |
| 1. Select film already posted. 2. The user clicks the -edit button-. 3. Selects the number of stars to rate the film (in a scale of 0 to 5 stars) that he/she is adding. 4. Update data from the database. |
| Comments: |
| The user can select any number of stars and also not select any. |

|  |
| --- |
| Use Case <Review or comment a film> |
| Types: |
| Users |
| * System user. |
| Prerequisites: |
| * There is a non-empty set of roles defined for the application. * There is a non-empty set of functions defined for the application. * The user is listed as an application/system user – you have an account on the system. * The user is signed in. |
| Typical step order: |
| 1. User clicks a button -add film- 2. Clicks the review/notes rectangle and fill it. 3. Add data to the database. |
| Final conditions: |
| * The data about the review/notes is updated. |
| Alternate step order: |
| 1. Select film already posted. 2. The user clicks the -edit button-. 3. Clicks the review/notes rectangle and fill it. 4. Update data from the database. |
| Comments: |
| It is not obligatory to fill it. |

|  |
| --- |
| Use Case <Mark as watched or want to watch> |
| Types: |
| Users |
| * System user. |
| Prerequisites: |
| * There is a non-empty set of roles defined for the application. * There is a non-empty set of functions defined for the application. * The user is listed as an application/system user – you have an account on the system. * The user is signed in. |
| Typical step order: |
| 1. User clicks a button -add film- 2. Selects the button of the clock if it is -want to watch- once the user adds the film automatically is added as watched (if they don’t select the want to watch clock). 3. Add data to the database. |
| Final conditions: |
| * The data about the watched or want to watch is updated. |
| Alternate step order: |
| 1. Select film already posted. 2. The user clicks the -edit button-. 3. Click the want to watch clock button (to select or remove it) 4. Update data from the database. |
| Comments: |
|  |

|  |
| --- |
| Use Case <Delete a film> |
| Types: |
| Users |
| * System user |
| Prerequisites: |
| * There is a non-empty set of roles defined for the application. * There is a non-empty set of functions defined for the application. * The user is listed as an application/system user – you have an account on the system. * The user is signed in. * The film must have been registered before. |
| Typical step order: |
| 1. User clicks a button -add film- 2. Clicks delete button. 3. Confirm action. 4. Delete data from the database. |
| Final conditions: |
| * The data about the film is deleted or removed. |
| Alternate step order: |
| 1. Select film already posted. 2. The user clicks the -edit button-. 3. Click the delete button. 4. Confirm action. 5. Delete data from the database. |
| Comments: |
|  |

#### 3.2.2. Use case 2.2: <Sign up>

|  |
| --- |
| Use Case <Sign up> |
| Types: |
| Users |
| * System user |
| Prerequisites: |
| * The user is NOT listed as an application/system user – you do not have an account on the system. |
| Typical step order: |
| 1. The user enters to the web page link. 2. Click the -sign up- button in the upper right side. 3. Enters the email address. 4. Enters the username. 5. Enters the password. 6. Clicks -Sign up- button from the bottom. |
| Final conditions: |
| * The user creates an account. * User’s data is now added to the database. |
| Alternate step order: |
|  |
| Comments: |
|  |

#### 3.2.3. Use case 2.3: <Sign in>

|  |
| --- |
| Use Case <Sign in> |
| Types: |
| Users |
| * System user |
| Prerequisites: |
| * There is a non-empty set of roles defined for the application. * There is a non-empty set of functions defined for the application. * The user is listed as an application/system user – you have an account on the system. |
| Typical step order: |
| 1. The user enters to the web page link. 2. Enters the username or email. 3. Enters the password. 4. Clicks -Sign in- button. |
| Final conditions: |
| * The user can enter the platform of our web application to view and use the functions. * The main page and its functions will be displayed. |
| Alternate step order: |
|  |
| Comments: |
|  |

#### 3.2.4. Use case 2.4: <Sign out>

|  |
| --- |
| Use Case <Sign out> |
| Types: |
| Users |
| * System user |
| Prerequisites: |
| * The user is listed as an application/system user – you have an account on the system. * The user is signed in. |
| Typical step order: |
| 1. The user selects the upper right button with the initials of the user. 2. Clicks sign out. |
| Final conditions: |
| * The user is log out from the session of the system. * The user stops seeing his/her movie records and statistics. |
| Alternate step order: |
|  |
| Comments: |
|  |

#### 3.2.5. Use case 2.5: <View statistics>

|  |
| --- |
| Use Case <View statistics> |
| Types: |
| Users |
| * System user |
| Prerequisites: |
| * There is a non-empty set of roles defined for the application. * There is a non-empty set of functions defined for the application. * The user is listed as an application/system user – you have an account on the system. * The user is signed in. * The user must have registered in various films. |
| Typical step order: |
| 1. The user goes to profile. 2. Click button - statistics. |
| Final conditions: |
| * The statistics are shown. |
| Alternate step order: |
|  |
| Comments: |
| The statistics are only shown if there are films in the data of the user. |

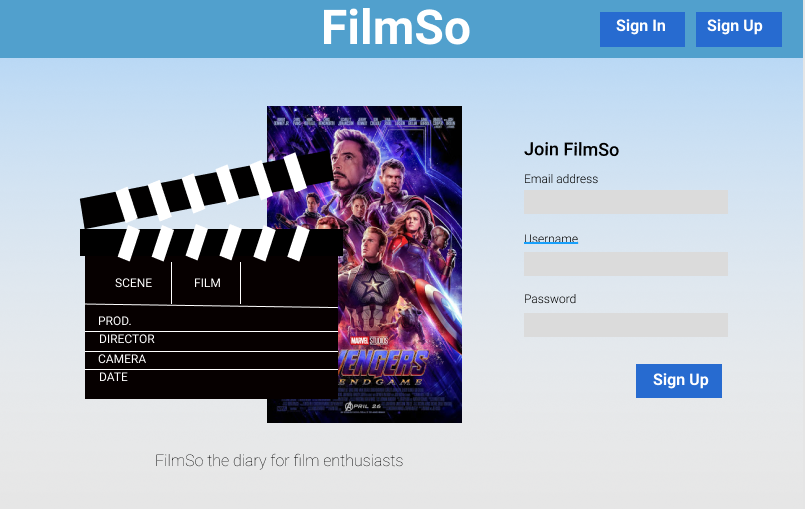
### **4. User interface (Original draft)**

The following is an example UI window for our application that helps the user to interact with our platform. Here you can see the functions that FilmSo has and what makes the application an application for movie enthusiasts.

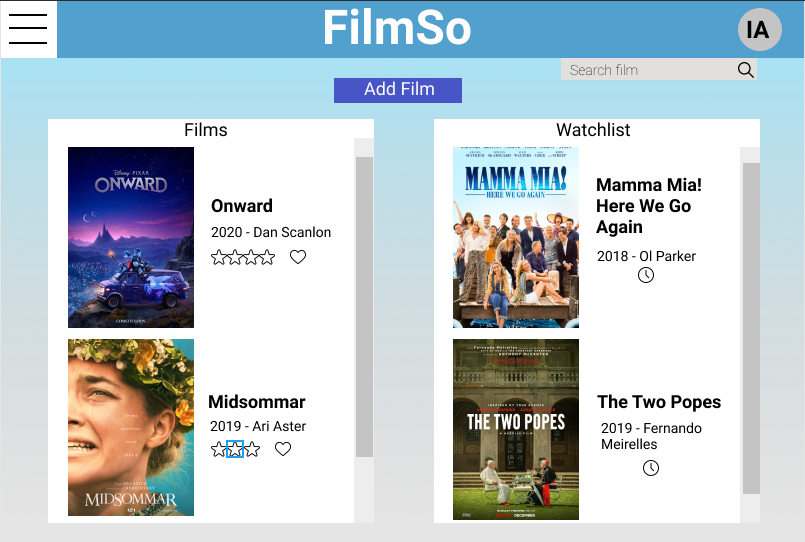
Login/Sign In application window:



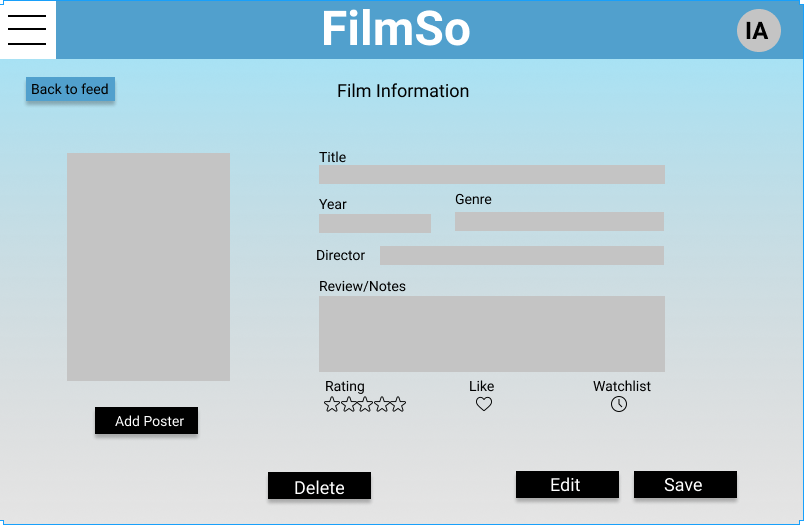
Sign Up application window:



Main application window:



Add film application window:



Profile application window:



Statistics/profile application window:



\*The plots here are only for illustrative purposes.

#### **Bibliography:**

|  |  |
| --- | --- |
| Source | Description |
| Rational Software Corporation (2002). “Guidelines: Use Case Model”. Web site: https://sceweb.uhcl.edu/helm/RationalUnifiedProcess/process/modguide/md\_ucmod.htm | Brief explanation of the use case model and how to make the diagram. |