|  |  |  |  |
| --- | --- | --- | --- |
| **User Task** | | Movie Management | |
| Purpose (Goal) | | Users manage movies and manipulate the corresponding data of performers and movies. | |
| Frequency | | Users are able to manage their collection at any time and as often as they like. | |
| Actors | | User who wants to manage movies | |
|  | | | |
| **Sub-tasks:** | | | **Example of solution:** |
| **1** | **Describe a movie**  Create and describe a movie with typical data like its title, releases, production locations, runtime, filming locations, and performers. Or change an existing description. Undo unwanted changes. View movies (possibly sorted and/or filtered) and movie details. Or search for Movies. | | Provide default values wherever possible, implemented in *Change Movie Detail Data*, *List Movies*, *Create Movie*, *Sort/Filter Movies*, *Undo Movie Changes*, *Show Movie Details*, *Search*, *Save Movie Detail Data Changes*, *Link Existing Performer* and *Link New Performer* |
| **1ap** | **Remove movie**  **Problems:** Removing all movies a certain performer participates in might result in performers not associated with movies. | | Ensure the consistency of performers and movies when movies are removed, implemented in function *R*emove Movie |
| **1bp** | **Remove the link between a performer and movie**  **Problems:** Removing a link can result in a performer not associated with any movies. | | Ensure the consistency of performers and movies when links are removed, implemented in function *Unlink Performer* |
| **2** | **Describe a performer**  Create and describe a performer with typical data like name, biography, occupation, date of birth, and movies s/he is featuring in. Or change an existing description. Undo unwanted changes. Or view performers (possibly filtered and sorted) and performer details. | | Provide default values wherever possible, implemented in *Change Performer Detail Data*, *List Performers*, *Sort/Filter Performers*, *Undo Performer Changes*, *Show Performer Details*, *Search*, *Save Performer Detail Data Changes*, *Link New Movie* and *Link Existing Movie*. |
| **2ap** | **Link performer to movie**  **Problems:** Creating a performer without linking her/him to a movie leads to performers not associated with any movies. | | Ensure the consistency of performers and movies when creating a performer, implemented in function Create Performer |
| **2bp** | **Remove the link between a performer and movie**  **Problems:** Removing a link can result in a performer not associated with any movies. | | Ensure the consistency of performers and movies when links are removed, implemented in function *Unlink Movie* |
| **2c** | **Remove performer**  Removes a performer. | | Implemented in function Remove Performer |
| **3** | **Sort movies / performers**  Movies can be sorted by title or by rating (descending or ascending).  Performers can be sorted by first name or by last name.  **Problems:** How to break ties (equal ratings) | | Implemented via the functions *Sort Movies* and *Sort Performers*  Function does not work in case of less than two movies/performers. Performer first name/last name ties can be resolved similarly by looking at the lexicographical ordering of other attributes. |
| **4** | **Search for movie/performer:**  Search for specific performer/movie.  **Problem**: Movies/Performers do not exist. In this case an empty query is returned and converted to the string “No performers to display" or "No movies to display". | | Implemented via the *Search Performer* and *Search Movie* functions. |