## **Workout Planner use cases**

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC-1:Login User | | |
| Primary actor | User | Secondary actors |  |
| Description | A user logs in as a regular User | | |
| Trigger | A user indicates that they want to log in | | |
| Preconditions | - | | |
| Postconditions | POST-1. User credentials are valid. | | |
| Normal flow | 1.0. User logs in with existent account  1. User clicks on login  2. The app shows a field for username and one for password  3. The user enters the credentials.  4. The app validates credentials. If they are valid user is logged in. | | |
| Alternative flows | 1.1. User creates account  1. User clicks on login, then signup.  2. The app shows a field for username and one for password  3. The user enters the new credentials.  4. The app validates the username. If it is unique the user is logged in and the credentials are stored. | | |
| Exceptions | 1.0.E1. Credentials are not valid  1. The app lets the user know credentials are not valid and restarts the use case.  1.0.E2. Username already exists  1. The app lets the user know the username is not valid and restarts the use case. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC-2: User creates exercise | | |
| Primary actor | User | Secondary actors |  |
| Description | User creates new exercise. | | |
| Trigger | User indicates that they want to add an exercise. | | |
| Preconditions | PRE-1: User is logged in | | |
| Postconditions | POST-1: Exercise is created. | | |
| Normal flow | 2.0. User creates new exercise  1. User clicks on add exercise button  2. The app shows a field for the exercise name and description.  3. User enters name and description.  4. The app saves the new exercise | | |
| Alternative flows | - | | |
| Exceptions | - | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC-3: User updates exercise | | |
| Primary actor | User | Secondary actors |  |
| Description | User updates existing exercise. | | |
| Trigger | User indicates that they want to update existing exercise. | | |
| Preconditions | PRE-1: User is logged in  PRE-2: Exercise exists | | |
| Postconditions | POST-1: Exercise is updated. | | |
| Normal flow | 3.0. User updates exercise  1. User clicks on an existing exercise and on the edit button  2. The app shows a field for the exercise name and description filled with the current information.  3. User modifies name and description.  4. The app saves the new exercise | | |
| Alternative flows | - | | |
| Exceptions | - | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC-4: User deletes exercise | | |
| Primary actor | User | Secondary actors |  |
| Description | User deletes exercise. | | |
| Trigger | User indicates that they want to delete existing exercise. | | |
| Preconditions | PRE-1: User is logged in  PRE-2: Exercise exists | | |
| Postconditions | POST-1: Exercise is deleted | | |
| Normal flow | 4.0. User deletes exercise  1. User clicks on an existing exercise and on the delete button  2. The app shows a warning and asks the user to confirm.  3. User clicks the confirm button.  4. The app deletes the exercise | | |
| Alternative flows | 4.1. User doesn’t confirm delete  1. User clicks on an existing exercise and on the delete button  2. The app shows a warning and asks the user to confirm.  3. User clicks away.  4. The app does not delete the exercise | | |
| Exceptions | - | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC-5: User creates workout | | |
| Primary actor | User | Secondary actors |  |
| Description | User creates workout. | | |
| Trigger | User indicates that they want to create new workout. | | |
| Preconditions | PRE-1: User is logged in | | |
| Postconditions | POST-1: Workout is created. | | |
| Normal flow | 5.0. User creates new workout  1. User clicks on new workout button  2. The app shows on a line a list of exercises and empty fields for number of reps and number of sets.  3. User enters the data.  4. The app shows another list of exercises and empty fields.  5. User clicks on save button  6. The app creates the new workout if the fields are not empty. | | |
| Alternative flows | - | | |
| Exceptions | 5.0.E1. Fields are empty  1. The app signals that there are empty fields and restarts the use case. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC-6: User updates workout | | |
| Primary actor | User | Secondary actors |  |
| Description | User updates workout. | | |
| Trigger | User indicates that they want to update existing workout. | | |
| Preconditions | PRE-1: User is logged in  PRE-2: Workout exits. | | |
| Postconditions | POST-1: Workout is updated. | | |
| Normal flow | 6.0. User updates workout  1. User clicks on a workout and on the edit button.  2. The app shows the list of exercises in the workout as editable fields and an empty line at the bottom for a new exercise.  3. User enters or modifies the data and clicks on the save button.  7. The app modifies new workout if the fields are not empty. | | |
| Alternative flows | - | | |
| Exceptions | 6.0.E1. Fields are empty  1. The app signals that there are empty fields and restarts the use case. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC-7: User deletes workout | | |
| Primary actor | User | Secondary actors |  |
| Description | User deletes workout. | | |
| Trigger | User indicates that they want to delete existing workout. | | |
| Preconditions | PRE-1: User is logged in  PRE-2: Workout exits. | | |
| Postconditions | POST-1: Workout is deleted. | | |
| Normal flow | 6.0. User deletes workout  1. User clicks on a workout and on the delete button.  2. The app shows a warning and asks the user to confirm.  3. User clicks the confirm button.  4. The app deletes the exercise | | |
| Alternative flows | 6.1 User does not confirm  1. User clicks on a workout and on the delete button.  2. The app shows a warning and asks the user to confirm.  3. User clicks away.  4. The app does not delete the exercise. | | |
| Exceptions | - | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC-7: Add workout to day | | |
| Primary actor | User | Secondary actors |  |
| Description | User adds an existing workout to a certain day. | | |
| Trigger | User indicates that they want to add workout to a day. | | |
| Preconditions | PRE-1: User is logged in  PRE-2: Workout exits. | | |
| Postconditions | POST-1: Workout is added to given day. | | |
| Normal flow | 7.0. User adds workout to day  1. User clicks on workout and on add to day button  2. The app shows a date picker.  3. User picks date.  4. The app saves date for workout. | | |
| Alternative flows | 7.1 User does not select day  1. User clicks on workout and on add to day button  2. The app shows a date picker.  3. User clicks away.  4. The app doesn't add workout to day. | | |
| Exceptions | - | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC-8: View workouts by day | | |
| Primary actor | User | Secondary actors |  |
| Description | User sees planned workouts on a selected day. | | |
| Trigger | User indicates that they want to see workouts by day. | | |
| Preconditions | PRE-1: User is logged in | | |
| Postconditions | POST-1: Workouts of given day are shown. | | |
| Normal flow | 8.0. User views workouts by day  1. User clicks on view by day button  2. The app shows a date picker and by default the workouts for the current date.  3. User picks date.  4. The app shows workouts for given date. | | |
| Alternative flows | 7.1 User does not select day  1. User clicks on view by day button  2. The app shows a date picker and by default the workouts for the current date.  3. User clicks away.  4. The app shows workouts for the current date. | | |
| Exceptions | - | | |