

Use Case 1: Manage Player Profiles

Author: Nicolas Siebler

Characteristic Information

Goal in Context: The user is able to create a user profile as well as manage the profile and see relevant statistics.

Scope: Application

Level: Subfunction

Primary Actor: User

Stakeholders & Interests:

- ❖ User: Wants to be able to create, see, edit, change, and delete his profiles. Furthermore he want to see relevant information to each profile

Preconditions: User has started the application

Success Guarantee (Postconditions): The user has successfully created a profile, can see his information, and switch between profiles, and finally delete a profile.

Trigger: User navigates through the GUI and clicks on 'Manage Player Profile'

Main Success Scenario

1. XML file containing the profiles with their information is loaded in
2. User clicks on the "create new profile button"
3. The user fills out all the relevant fields
4. New account is created
5. All values are stored in the XML Player Profile file
6. User sees relevant information
7. User updates specific profile information
8. Profile information in the XML is updated
9. User switches between profiles
10. User deletes a profile
11. GUI shows next profile
12. Deleted profile is removed from the XML file

Extensions

- *a. At any time, System fails to find the XML file:
 - 1. User is notified
 - 2. System requests new location
 - 2.1. If player selects 'create new save file' instead, then system creates a new file
 - 2.2. If player selects a location, then file is moved back to predefined default spot
- 1. XML file is corrupt (different format than expected)
 - 1. User is notified
 - 2. System requests confirmation to delete old file and create new one
- 3. a. fields are not filled with valid inputs (e.g. text in numeric field)
 - 1. User is notified about incorrect input
 - 2. field is cleared
- 3. b. profile name already exists
 - 1. User is notified that another profile already is called like that
 - 2. field input for name is highlighted red
 - 3. System requests a different name
- 7. fields is not filled with valid input
 - 1. User is notified about incorrect input
 - 2. field is cleared
- 10. User deletes the only profile is deleted
 - 1. System notifies user that there are no other profiles
 - 2. Profile is not deleted

Technology & Data Variation List:

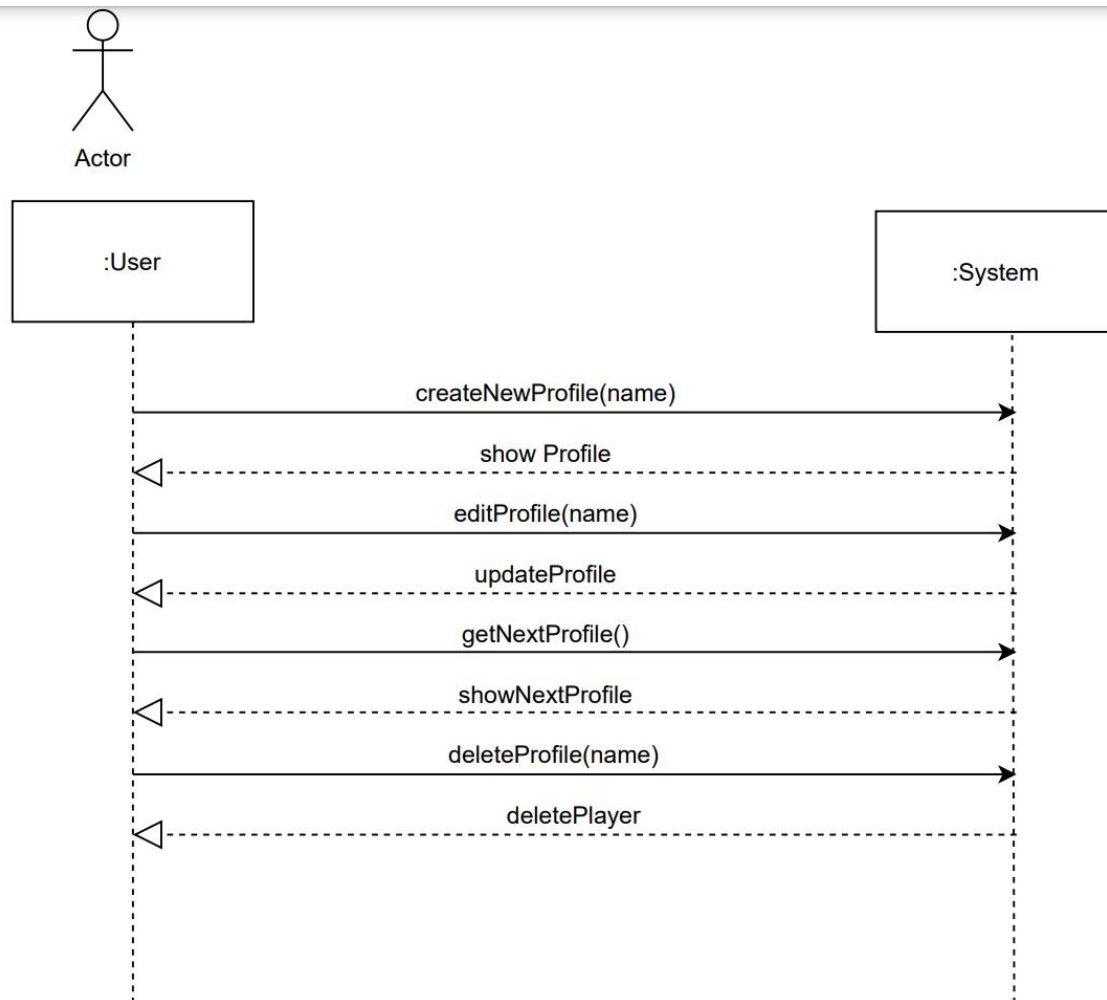
For the player Profile we exclusively use a xml file and store every value as a String (no attributes), to facilitate the usage of the data

Due Date

12.04.2021

System Sequence Diagram

Author: Nicolas Siebler



Operation Contracts

Author: Nicolas Siebler

Contract 1.1: createNewProfile

Operation: createNewProfile (name : String) : void

Cross References: UC 1: Manage Player Profiles

Preconditions:

- Player is in the profile menu
- Name was not occupied by another PlayerProfile associated with user's client

Postconditions:

- PlayerProfile p instance was created and associated with User's client
- P.name became *name*

Contract 1.2: editProfile

Operation: editProfile (name : String) : void

Cross References: UC 1: Manage Player Profiles

Preconditions:

- User edited PlayerProfile p
- Name was not occupied by another PlayerProfile associated with user's client

Postconditions:

- P.name became *name*

Contract 1.3: deleteProfile

Operation: deleteProfile () : void

Cross References: UC 1: Manage Player Profiles

Preconditions:

- User wants to delete PlayerProfile p

Postconditions:

- p has been disassociated from user's client

Contract 1.4: getNextProfile

Operation: getNextProfile () : void

Cross References: UC 1: Manage Player Profiles

Preconditions:

- User's next profile associated with his or her client is PlayerProfile instance p

Postconditions:

- Client.selectedProfile became p