## **Impartire pe iteratii**

Iteratia 1: Domain, Repository, Service, DB setup, UI LogIn/Register

Iteratia 2: UC-1, UC-2, UC-3, UC-4, UC-9, UI Client

Iteratia 3: UC-5, UC-6, UC-7, UC-8, UI Admin

## **Use case template**

An adaptation of the standard Cockburn template will be used. The template and examples follow:

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC-1: Log In | | |
| Primary actor | User | Secondary actors | Sistem |
| Description | Un user (client sau admin) introduce username-ul si parola, daca aceste date sunt valide si existente, User-ul va acesa platforma, daca nu exceptiile vor fi afisate in ErrorBox-uri, | | |
| Trigger | Un user apasa butonul de login cu datele introduse | | |
| Preconditions | - | | |
| Postconditions | POST-1: User is logged  POST-2: Window closes  POST-3: UserWindow is opened | | |
| Normal flow | 1. User-ul introduce username-ul si parola 2. User-ul apasa butonul de log in 3. Fereastra de log in se inchide si se deschide fereastra de rezervare locuri | | |
| Alternative flows | 1. Daca User-ul este admin se va deschide o alta fereastra, de unde acesta introduce spectacole noi | | |
| Exceptions | E1. Username inexistent  UC-1 nu se opreste!  E2. Parola gresita  UC-1 nu se opreste!  E3. Date lipsa din chestionar | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC-2: Register | | |
| Primary actor | User | Secondary actors | Sistem |
| Description | Un utilizator isi poate creea un cont introducand datele sale: - nume - username - parola - email Dupa creare, fereastra se inchide si se deschide UserWindow | | |
| Trigger | Un utilizator apasa butonul de register cu datele introduse corect | | |
| Preconditions | PRE-1. Sa nu existe un alt User cu acelasi username  PRE-2. Sa nu se creeze un User cu aceiasi adresa de email a altuia | | |
| Postconditions | POST-1: User is created and logged  POST-2: Window closes  POST-3: UserWindow is opened | | |
| Normal flow | 1. Introduce numele 2. Introduce un username 3. Introduce o parola 4. Introduce un email 5. Apasa butonul de register 6. User-ul nou este creat 7. Fereastra se inchide si se deschide cea a User-ului | | |
| Alternative flows | - | | |
| Exceptions | E1. Username existent (recompletarea datelor)  E2. Email existent (recompletarea datelor)  E3. Date incomplete/invalide | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC-3: Selectare/Deselectare Locuri | | |
| Primary actor | User  (isAdmin = False) | Secondary actors | Sistem |
| Description | Un utilizator poate selecta unul sau mai multe locuri care sunt disponibile. Locurile necolorate sunt disponibile, cele rosii ocupate iar cele albastru deschis selectate de user. Pentru a selecta un loc se va apasa pe locul dorit si pentru a-l deselecta se apasa pe unul deja selectat. | | |
| Trigger | Click pe loc disponibil/selectat | | |
| Preconditions | PRE-1. User logged  PRE-2. Locuri libere existente | | |
| Postconditions | POST-1. Bilet creat pt fiecare loc selectat  POST-2. Locurile devin albastre (se selecteaza) | | |
| Normal flow | 1. Se selecteaza un loc sau mai multe 2. Se da click pe un loc selectat si se poate deselecta | | |
| Alternative flows | - | | |
| Exceptions | E1. Locul este ocupat (se va afisa un ErrorBox)  E2. Locul selectat se ocupa intre timp (se va afisa un InfoBox) | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC-4: Comanda Locuri | | |
| Primary actor | User | Secondary actors | Sistem |
| Description | Dupa selectarea a minim un loc, User-ul poate plasa comanda de locuri si acesta vor fi asociate lui pe un Bilet sau mai multe (in functie de cate locuri selecteaza) | | |
| Trigger | Click buton Comanda | | |
| Preconditions | PRE-1: Locuri selectate disponibile  PRE-2: User logged | | |
| Postconditions | POST-1: Bilet generat pt fiecare loc selectat  POST-2: Locurile devin rosii (se ocupa) | | |
| Normal flow | 1. Apasarea buton comanda 2. Crearea Biletelor 3. Ocuparea Locurilor | | |
| Alternative flows | - | | |
| Exceptions | E1. Niciun loc selectat -> ErrorBox | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-5: Vizualizare Spectacole | | |
| **Primary actor** | User | **Secondary actors** | Sistem |
| **Description** | Un User cu drepturi de admin, poate vizualiza toate spectacolele introduse in baza de date. Spectacolele vor fi sortate dupa data in care vor fi redate crescator, de la 3 zile anterioare datii curente | | |
| **Trigger** | - | | |
| **Preconditions** | PRE-1. User Admin loggat | | |
| **Postconditions** | POST-1. Spectacolele existente afisate | | |
| **Normal flow** | 1. Se iau din repo spectacolele 2. Se sorteaza crescator dupa data 3. Se afiseaza | | |
| **Alternative flows** | 1. Daca nu exista spectacole, se va afisa un mesaj cu “nu exista spectacole” | | |
| **Exceptions** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-6: Adauga Spectacole | | |
| **Primary actor** | User | **Secondary actors** | Sistem |
| **Description** | Un User cu drepturi de admin, poate adauga spectacole pentru urmatoarele zile, cu conditia ca in ziua respectiva sa nu existe alte spectacole | | |
| **Trigger** | Click buton adauga | | |
| **Preconditions** | PRE-1. User Admin loggat  PRE-2: Data spectacolului sa fie libera | | |
| **Postconditions** | POST-1. Spectacolele afisate se actualizeaza  POST-2. Spectacolul nou este adaugat | | |
| **Normal flow** | * + - 1. Se completeaza titlu pret, data, descriere       2. Se creaza Spectacolul       3. Este introdus in DB si in fereastra | | |
| **Alternative flows** | - | | |
| **Exceptions** | E1. Data introdusa este ocupata de alt spectacol  E2. Datele introduse sunt incomplete | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-7: Modifica Spectacol | | |
| **Primary actor** | User | **Secondary actors** | Sistem |
| **Description** | Un User cu drepturi de admin, poate modifica un spectacol introducand o data in care deja exista un spectacol si scimband ori toate atributele ori doar unele. | | |
| **Trigger** | Click buton modifica | | |
| **Preconditions** | PRE-1. User Admin loggat  PRE-2: Data spectacolului sa fie existe in Repo Spectacol | | |
| **Postconditions** | POST-1. Spectacolele afisate se actualizeaza  POST-2. Spectacolul din data selectata este modificat | | |
| **Normal flow** | * + - 1. Se completeaza titlu pret, data, descriere       2. Se modifica Spectacolul       3. Este actualizata in DB si in fereastra | | |
| **Alternative flows** | - | | |
| **Exceptions** | E1. Data introdusa este libera de alt spectacol | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-8: Sterge Spectacole | | |
| **Primary actor** | User | **Secondary actors** | Sistem |
| **Description** | Un User cu drepturi de admin, poate sterge spectacole selecatandu-le in view. | | |
| **Trigger** | Click buton delete + spectacol selectat | | |
| **Preconditions** | PRE-1. User Admin loggat  PRE-2: Un spectacol selectat | | |
| **Postconditions** | POST-1. Spectacolele afisate se actualizeaza  POST-2. Spectacolul sters nu va mai exista nici in DB | | |
| **Normal flow** | * + - 1. Se selecteaza un spectacol       2. Se apasa delete       3. Se va sterge din view si db | | |
| **Alternative flows** | - | | |
| **Exceptions** | E1. Nu exista spectacol selectat | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-9: Log Out | | |
| **Primary actor** | User | **Secondary actors** | Sistem |
| **Description** | User-ul fie in AdminPanel fie in UserPanel are un buton de log out care il va deloga si ii va inchide fereastra si o va redeschide pe cea de log in / register | | |
| **Trigger** | Click buton log out | | |
| **Preconditions** | PRE-1. User Admin loggat | | |
| **Postconditions** | POST-1. UserWindow/AdminPanel se inchide  POST-2. Utilizatorul delogat | | |
| **Normal flow** | * + - 1. Se apasa butonul de logout       2. Se inchide fereastra curenta       3. Se redeschide cea de login | | |
| **Alternative flows** | - | | |
| **Exceptions** | - | | |

Descriptions of template fields:

* **ID and name:** Title should be descriptive and should usually begin with a verb, e.g. order, calculate, input, etc. ID can have any format but must be unique among all use cases.
* **Primary actor:** Person that wishes to accomplish a goal through the use of the system. Only a single primary actor per use case.
* **Secondary actors:** Actors that have an interest in the completion of the goal but that do not directly interact with the system.
* **Description:** Concise description of the purpose of the use case.
* **Trigger:** Condition internal or external to the system that prompts the use case to start.
* **Preconditions:** Conditions that must be true before the use case starts. Each should be labeled with an ID unique to the use case.
* **Postconditions:** Conditions that must be true after the use case ends normally. Each should be labeled with an ID unique to the use case.
* **Normal flow:** Detailed step-by-step description of the logical flow of the use case. It should describe an explicit two way interaction, with the system prompting for input and the actor responding accordingly. Each step should be numbered.
* **Alternative flows:** Flows that achieve the same goal as the normal flow but are expected to be less common or lower priority.
* **Exceptions:** Conditions that result in the normal flow ending prematurely due to an unrecoverable condition in the system. The condition that causes the flow should be clearly stated, as should be any other decisions that the actor must make in this situation.