# Requirement guide for class project

[Use cases](https://en.wikipedia.org/wiki/Use_case) will be used as requirements for this project.

## Template

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

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC 1: Rezervare loc spectacol | | |
| Primary actor | Spectator | Secondary actors |  |
| Description | Un spectator accesează terminalul de rezervare locuri pentru reprezentația din ziua curentă, vizualizează configurația actuală a sălii, își introduce datele solicitate și alege locurile pe care dorește să le rezerve. | | |
| Trigger | Un spectator dorește să rezerve unul sau mai multe locuri. | | |
| Preconditions | PRE 1: Terminalul deține configurația actuală a sălii și a locurilor disponibile | | |
| Postconditions | POST 1: Terminalul memorează locurile ocupate în urma rezervării  POST 2: Terminalul afișează locurile ocupate și în celelalte instanțe ale sale | | |
| Normal flow | 1. Spectatorul accesează terminalul de rezervări  2. Sistemul afișează toate locurile din sală, incluzând pentru fiecare poziția, numărul, prețul aferent și starea actuală, doar spre vizualizare.  3. Spectatorul optează pentru rezervarea de locuri  4. Sistemul afișează un formular în care îi solicită spectatorului datele personale (de ex. nume, prenume, email, nr. de telefon, CNP etc.)  5. Spectatorul completează datele cerute de sistem și le trimite.  6. Sistemul afișează toate locurile din sală, incluzând pentru fiecare poziția, numărul, prețul aferent și starea actuală.  7. Spectatorul selectează locul/ locurile pe care intenționează să le rezerve.  8. Sistemul memorează locul/ locurile selectate.  9. Spectatorul confirmă rezervarea locurilor selectate prin declanșarea unui buton de confirmare.  10. Sistemul memorează locurile selectate, actualizează locurile disponibile și afișează un mesaj de confirmare spectatorului. | | |
| Alternative flows | Nu există | | |
| Exceptions | 1.E1. Spectatorul introduce date personale invalide  6a. Sistemul detectează introducerea unor date personale eronate (nr. de telefon, email, CNP, etc.) și atenționează spectatorul asupra acelor câmpuri cu valori invalide.  6a1. Spectatorul reintroduce datele (corecte).  Se revine la 6.  1.E2. Spectatorul selectează locuri deja rezervate  8a. Sistemul atenționează spectatorul asupra faptului că încearcă să selecteze locuri deja rezervate.  8a1. Spectatorul selectează alte locuri (libere).  Se revine la 8.  1.E3. Spectatorul încearcă să rezerve locuri la o reprezentație înainte/ după o anumită oră (înainte de deschidere/ după ora începerii spectacolului)  4a. Sistemul atenționează spectatorul asupra faptului că încearcă să selecteze locuri într-un interval de timp inadecvat, afișând detalii legate de motivul invocat.  8a1. Spectatorul ia la cunoștință detaliile.  Se revine la 2. | | |

|  |  |
| --- | --- |
| ID and name | UC 2: Vizualizare locuri spectacol |

|  |  |  |  |
| --- | --- | --- | --- |
| Primary actor | Spectator | Secondary actors |  |
| Description | Un spectator accesează terminalul de rezervare locuri pentru reprezentația din ziua curentă și vizualizează configurația actuală a sălii | | |
| Trigger | Un spectator dorește să vizualizeze locuri (disponibile și rezervate) de la reprezentația din ziua curentă. | | |
| Preconditions | PRE 1: Terminalul deține configurația actuală a sălii și a locurilor disponibile | | |
| Postconditions | Nu există | | |
| Normal flow | 1. Spectatorul accesează terminalul  2. Sistemul afișează toate locurile din sală, incluzând pentru fiecare poziția, numărul, prețul aferent și starea actuală, doar spre vizualizare. | | |
| Alternative flows | Nu există | | |
| Exceptions | Nu există | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC 3: Votare spectacol următor | | |
| Primary actor | Spectator | Secondary actors |  |
| Description | Un spectator accesează terminalul de vot spectacole, selectează dintr-o listă de spectacole acel/ acele spectacole pe care și le dorește și apoi confirmă selecția făcută | | |
| Trigger | Un spectator dorește să voteze un spectacol | | |
| Preconditions | PRE 1: Terminalul deține o listă cu posibile spectacole ce pot fi votate | | |
| Postconditions | POST 1: Terminalul memorează voturile acordate de spectator | | |
| Normal flow | 1. Spectatorul accesează terminalul de vot spectacole  2. Sistemul afișează toate spectacolele disponibile pentru votare  3. Spectatorul vizualizează lista și selectează un spectacol pentru a-l vota  4. Sistemul afișează spectacolul selectat  5. Spectatorul apasă un buton de confirmare vot  6. Sistemul afișează un mesaj de confirmare | | |
| Alternative flows | * 1. Votare mai multe spectacole  1. Spectatorul dorește să voteze mai multe spectacole 2. Se revine la pasul 3. din Normal flow | | |
| Exceptions | 1.E1. Spectatorul votează de două ori același spectacol  6a. Sistemul detectează încercarea de votare multiplă a unui spectacol de către un spectator și atenționează spectatorul asupra acestei erori.  6a1. Spectatorul accesează un buton de confirmare.  Se revine la 3.  1.E2. Nu există spectacole disponibile pentru a fi votate  8a. Sistemul atenționează spectatorul asupra faptului că nu există momentan spectacole ce pot fi votate.  8a1. Spectatorul accesează un buton de confirmare. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC 4: Sugerare spectacol următor | | |
| Primary actor | Spectator | Secondary actors |  |
| Description | Un spectator accesează terminalul de sugestii spectacole, introduce datele unui spectacol pe care dorește să îl sugereze și confirmă introducerea făcută. | | |
| Trigger | Un spectator dorește să sugereze un spectacol | | |
| Preconditions | Nu există | | |
| Postconditions | POST 1: Terminalul memorează detaliile spectacolului introdus | | |
| Normal flow | 1. Spectatorul accesează terminalul de sugestii spectacole  2. Sistemul afișează un formular cu datele necesare introducerii unui spectacol sugerat  3. Spectatorul introduce datele cerute de terminal (nume spectacol, autor, an, sugestii etc.) și apasă un buton de confirmare  4. Sistemul confirmă introducerea făcută. | | |
| Alternative flows | * 1. Votare mai multe spectacole  1. Spectatorul dorește să sugereze mai multe spectacole 2. Se revine la pasul 3. din Normal flow | | |
| Exceptions | 1.E1. Spectatorul dorește să introducă de două ori același spectacol  6a. Sistemul detectează încercarea de sugestie multiplă a unui spectacol de către un spectator și atenționează spectatorul asupra acestei erori.  6a1. Spectatorul accesează un buton de confirmare.  Se revine la 3. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC 5: Introducerea unui nou spectacol | | |
| Primary actor | Regizor | Secondary actors |  |
| Description | Un regizor accesează terminalul de gestionare spectacole, vizualizează spectacolele existente și cele propuse, introduce datele necesare pentru un anumit spectacol și confirmă introducerile făcute. | | |
| Trigger | Un regizor dorește să adauge detaliile pentru un spectacol | | |
| Preconditions | PRE 1: Terminalul are memorate sugestii de spectacole ce pot fi gestionate de regizor | | |
| Postconditions | POST 1: Terminalul memorează detaliile introduse de regizor. | | |
| Normal flow | 1. Regizorul accesează terminalul de gestiune spectacole.  2. Sistemul afișează o listă cu spectacolele înregistrate deja în sistem și una cu cele sugerate de spectatori, precum și un buton de adăugare spectacol nou.  3. Regizorul selectează adăugarea unui nou spectacol.  4. Sistemul afișează un formular cu datele necesare pentru un spectacol (titlu, autor, actori, roluri, data spectacolului etc.)  5. Regizorul completează datele aferente formularului.  6. Sistemul adaugă spectacolul în lista de spectacole înregistrate. | | |
| Alternative flows | * 1. Introducerea unui nou spectacol din lista celor sugerate  1. Sistemul completează automat formularul afișat la pasul 4. cu datele introduse de spectator pentru un spectacol sugerat 2. Se revine la pasul 5. din Normal flow | | |
| Exceptions | 1.E1. Spectatorul dorește să introducă de două ori același spectacol  6a. Sistemul detectează încercarea de sugestie multiplă a unui spectacol de către un spectator și atenționează spectatorul asupra acestei erori.  6a1. Spectatorul accesează un buton de confirmare.  Se revine la 3. | | |

Planificare iterații:

1. Inițializare terminal + vizualizare configurație sală
   1. Domain locuri
   2. Domain rezervare
   3. Repository locuri
   4. Repository rezervare
   5. Service locuri
   6. Service rezervare
   7. Controller locuri + rezervare
2. Rezervare locuri libere
   1. Domain spectator
   2. Repository spectator
   3. Service spectator
   4. Controller spectator
3. Sincronizare locuri libere între terminale + validări cazuri excepții
   1. Observer/ Terminals synchronization
   2. Exception manager

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name |  | | |
| Primary actor |  | Secondary actors |  |
| Description |  | | |
| Trigger |  | | |
| Preconditions |  | | |
| Postconditions |  | | |
| Normal flow |  | | |
| 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.

## Examples

For a hypothetical *Cafeteria Ordering System*[[1]](#footnote-1):

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-1: Order a Meal | | |
| **Primary actor** | Patron | **Secondary actors** | Cafeteria Inventory System |
| **Description** | A Patron accesses the Cafeteria Ordering System from either the corporate intranet or external Internet, views the menu for a specific date, selects food items, and places an order for a meal to be picked up in the cafeteria or delivered to a specified location within a specified 15-minute time window. | | |
| **Trigger** | A Patron indicates that he wants to order a meal. | | |
| **Preconditions** | PRE-1. Patron is logged into COS.  PRE-2. Patron is registered for meal payments by payroll deduction. | | |
| **Postconditions** | POST-1. Meal order is stored in COS with a status of “Accepted.”  POST-2. Inventory of available food items is updated to reflect items in this order.  POST-3. Remaining delivery capacity for the requested time window is updated. | | |
| **Normal flow** | **1.0 Order a Single Meal**   1. Patron asks to view menu for a specific date. (see 1.0.E1, 1.0.E2) 2. COS displays menu of available food items and the daily special. 3. Patron selects one or more food items from menu. (see 1.1) 4. Patron indicates that meal order is complete. (see 1.2) 5. COS displays ordered menu items, individual prices, and total price, including taxes and delivery charge. 6. Patron either confirms meal order (continue normal flow) or requests to modify meal order (return to step 2). 7. COS displays available delivery times for the delivery date. 8. Patron selects a delivery time and specifies the delivery location. 9. Patron specifies payment method. 10. COS confirms acceptance of the order. 11. COS sends Patron an email message confirming order details, price, and delivery instructions. 12. COS stores order, sends food item information to Cafeteria Inventory System, and updates available delivery times. | | |
| **Alternative flows** | **1.1 Order multiple identical meals**   1. Patron requests a specified number of identical meals. (see 1.1.E1) 2. Return to step 4 of normal flow.   **1.2 Order multiple meals**   1. Patron asks to order another meal. 2. Return to step 1 of normal flow. | | |
| **Exceptions** | **1.0.E1 Requested date is today and current time is after today’s order cutoff time**  1. COS informs Patron that it’s too late to place an order for today.  2a. If Patron cancels the meal ordering process, then COS terminates use case.  2b. Else if Patron requests another date, then COS restarts use case.  **1.0.E2 No delivery times left**  1. COS informs Patron that no delivery times are available for the meal date.  2a. If Patron cancels the meal ordering process, then COS terminates use case.  2b. Else if Patron requests to pick the order up at the cafeteria, then continue with normal flow, but skip steps 7 and 8.  **1.1.E1 Insufficient inventory to fulfill multiple meal order**  1. COS informs Patron of the maximum number of identical meals he can order, based on current available inventory.  2a. If Patron modifies number of meals ordered, then return to step 4 of normal flow.  2b. Else if Patron cancels the meal ordering process, then COS terminates use case. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-5 Register for Payroll Deduction | | |
| **Primary actor** | Patron | **Secondary actors** | Payroll System |
| **Description** | Cafeteria patrons who use the COS and have meals delivered must be registered for payroll deduction. For noncash purchases made through the COS, the cafeteria will issue a payment request to the Payroll System, which will deduct the meal costs from the next scheduled employee payday direct deposit. | | |
| **Trigger** | Patron requests to register for payroll deduction, or Patron says yes when COS asks if he wants to register. | | |
| **Preconditions** | PRE-1. Patron is logged into COS. | | |
| **Postconditions** | POST-1. Patron is registered for payroll deduction. | | |
| **Normal flow** | **5.0 Register for Payroll Deduction**   1. COS asks Payroll System if Patron is eligible to register for payroll deduction. 2. Payroll System confirms that Patron is eligible to register for payroll deduction. 3. COS asks Patron to confirm his desire to register for payroll deduction. 4. If so, COS asks Payroll System to establish payroll deduction for Patron. 5. Payroll System confirms that payroll deduction is established. 6. COS informs Patron that payroll deduction is established. | | |
| **Alternative flows** | None | | |
| **Exceptions** | 5.0.E1 Patron is not a full time employee.  5.0.E2 Patron is already enrolled for payroll deduction. | | |

## Extra credit step: Traceability

For this extra step, you will add traceability information for each use case by adding a new field to the template:

|  |  |
| --- | --- |
| Method-level traces | <fully.qualified.ClassName>#<methodName>  ... |

Any method that implements the functionality described in the normal flow, alternative flow or exceptions should be included in this field. This means that the method that is initially executed and any methods of any classes that the work is delegated to should be included.

Examples for previous use cases:

UC-1:

|  |  |
| --- | --- |
| Method-level traces | my.company.ordering.MenuWidget#dateClicked  my.company.ordering.MenuWidget#completeOrder  my.company.ordering.InventoryInterface#checkInventory  ... |

UC-5:

|  |  |
| --- | --- |
| Method-level traces | my.company.payroll.PayrollInterface#checkEligibility  my.company.payroll.RegistrationForm#confirm  ... |

1. Spectatorul accesează terminalul

2. Sistemul afișează toate locurile din sală, incluzând pentru fiecare poziția, numărul, prețul aferent și starea actuală, doar spre vizualizare.

3. Spectatorul optează pentru rezervarea de locuri

4. Sistemul afișează un formular în care îi solicită spectatorului datele personale (de ex. nume, prenume, email, nr. de telefon, CNP etc.)

5. Spectatorul completează datele cerute de sistem

6. Sistemul afișează toate locurile din sală, incluzând pentru fiecare poziția, numărul, prețul aferent și starea actuală.

7. Spectatorul selectează locul/ locurile pe care intenționează să le rezerve.

8. Sistemul memorează locul/ locurile selectate.

9. Spectatorul confirmă rezervarea locurilor selectate prin declanșarea unui buton de confirmare.

10. Sistemul memorează locurile selectate, actualizează locurile disponibile și afișează un mesaj de confirmare spectatorului.

1. Spectatorul accesează terminalul

2. Sistemul afișează toate locurile din sală, incluzând pentru fiecare poziția, numărul, prețul aferent și starea actuală, doar spre vizualizare.

3. Spectatorul optează pentru rezervarea de locuri

4. Sistemul afișează un formular în care îi solicită spectatorului datele personale (de ex. nume, prenume, email, nr. de telefon, CNP etc.)

5. Spectatorul completează datele cerute de sistem

6a. Sistemul detectează introducerea unor date personale eronate (nr. de telefon, email, CNP, etc.) și atenționează spectatorul asupra acelor câmpuri cu valori invalide.

6a1. Spectatorul reintroduce datele (corecte).

Se revine la 6.

7. Spectatorul selectează locul/ locurile pe care intenționează să le rezerve.

8a. Sistemul atenționează spectatorul asupra faptului că încearcă să selecteze locuri deja rezervate.

8a1. Spectatorul selectează alte locuri (libere).

Se revine la 8.

9. Spectatorul confirmă rezervarea locurilor selectate prin declanșarea unui buton de confirmare.

10. Sistemul memorează locurile selectate, actualizează locurile disponibile și afișează un mesaj de confirmare spectatorului.

1. Examples adapted from Wiegers, K. E. & Beatty, J. (2013) Software requirements . 3rd ed. Redmond, WA: Microsoft Press. [↑](#footnote-ref-1)