# 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 | B1:vizualizare lista carti | | |
| Primary actor | Abonat | Secondary actors | Sitemul bibliotecii |
| Description | Abonatul dupa ce se logheaza poate viziona lista de carti. | | |
| Trigger | Abonatul vrea sa vizualizeze lista de carti. | | |
| Preconditions | Pre 1:Abonatul este autentificat. | | |
| Postconditions | None | | |
| Normal flow | 1.0 Vizualizeaza lista de carti  1.Abonatul deschide aplicatia   2.Abonatul de autentifica  3.Abonatului ii apare lista cu carti | | |
| Alternative flows | * 1. Vizualizare lista carti si imprumutare carte   1.Face exact ce a facut in 1.0 2.Abonatul cauta o anumita carte  3.Abonatul gaseste cartea  4.Abonatul o imprumuta  5.Simtemul actualizaeza lista cartilor 6.Toata lumea vede lista de carti actualizata | | |
| Exceptions | 1.1E1  Nu exista cartea cautata(vezi iteratia 2,acolo o sa detaliez)  1.2E2  Abonatului nu ii este permis sa imprumute cartea | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | B2:Imprumutare carte | | |
| Primary actor | Abonat | Secondary actors | Sitemul bibliotecii, bibliotecarul |
| Description | Abonatul dupa ce se logheaza poate viziona lista de carti,apoi poate alege unu.Daca n-are intarzieri la returnarea altor carti,el va putea imprumuta cartea. | | |
| Trigger | Abonatul vrea sa imprumute o carte. | | |
| Preconditions | Pre 1:Abonatul este autentificat.  Pre 2:Abonatul n-are intarziere la returnarea cartilor | | |
| Postconditions | Post 1:Imprumutul s-a inregistrat si iar lista cartilor disponibile s-a actualizat. | | |
| Normal flow | 1.0 Imprumuatare carte  1.Abonatul deschide aplicatia   2.Abonatul de autentifica  3.Abonatului ii apare lista cu carti  4.Abonatul poate cauta o carte anume  5.Cartea este disponibila  6.Abonatului i se aproba imprumutul  7.Sistemul a actualizat lista cartilor disponibile  8Abonatul iese din aplicatie | | |
| Alternative flows | 1.1Imprumutul nu se mai realizeaza ,abonatul doar vizualizand lista de carti  1.Abonatul deschide aplicatia   2.Abonatul de autentifica  3.Abonatului ii apare lista cu carti  4.Abonatul se razgandeste si iese din aplicatie | | |
| Exceptions | 1.0E1  Nu exista cartea cautata  .Dupa ce acesta cauta carte,sistemul afiseaza mesajul care-I indica ca caertea nu mai este disponibila  1.2E2  Abonatului nu ii este permis sa imprumute cartea  Aplicatia anunta bibliotecarul care abonatul respectiv nu este la zi cu returnarrea cartilor ,apoi bibliotecarul invalideaza imprumutul | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | B3:Returnare carte | | |
| Primary actor | Bibliotecar | Secondary actors | Sitemul bibliotecii, abonatul |
| Description | Abonatul retunreaza carte dupa ce a terminat-o de citit.Acesta i-o inmaneaza bibliotecarului ,iar acesta o trece in sistem ,iar sistemul actualizeaza lista de carti.La final bibliotecarul si ceilati abonati vad lista cartilor actualiazata. | | |
| Trigger | Abonatul vrea sa retuneze cartea. | | |
| Preconditions | Pre 1:Bibliotecarul este autentificat. | | |
| Postconditions | Post 1:Returnarra s-a inregistrat si iar lista cartilor disponibile s-a actualizat. | | |
| Normal flow | 1.0 Returnare carte  1.Bibliotecarul deschide aplicatia   2.Bibliotecarul de autentifica  3.Bibliotecarul o trece in sitem ca si returnata  4.Sistemul actualizeaza lista cartilor | | |
| Alternative flows | None | | |
| Exceptions | None | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | B4: Manage lista carti | | |
| Primary actor | Admin | Secondary actors | Sitemul bibliotecii |
| Description | Adminul dupa ce se logheaza poate viziona lista apoi o poate modifica.Acesta poate aduga carti,sterge carti sau poate actualiza informatii desre anumite carti. | | |
| Trigger | Adminul vrea sa modifice lista cartilor. | | |
| Preconditions | Pre 1:Adminul este autentificat. | | |
| Postconditions | Post 1: Lista cartilor s-a actualizat. | | |
| Normal flow | 1.0 Actualizare carte  1.Adminul deschide aplicatia   2.Adminul de autentifica  3.Adminului ii apare lista cu carti  4.Adminul modifica informatiile despre o anumita carte.  5.Lista de carti este actualizata | | |
| Alternative flows | 1.1Adaugare carte  1Adminul deschide aplicatia   2.Adminul de autentifica  3.Adminului ii apare lista cu carti  4.Adminul adauga o carte noua..  5.Lista de carti este actualizata 1.2 Stergere carte  1.Adminul deschide aplicatia   2.Adminul de autentifica  3.Adminului ii apare lista cu carti  4.Adminul sterge din lista o anumita carte  5.Lista de carti este actualizata | | |
| Exceptions | None | | |

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