Use Cases

Spital Farmacie

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | UC-1 SignUp | | |
| Primary actor | Angajat Spital | Secondary actors | Sistemul Spitalului |
| Description | Un angajat al spitatlului isi va crea un cont nou pentru accesarea sistemului de prescriptii. Va aveqa posiblitatea de a alege daca este angajat pe sectie(doctor) sau angajat al farmaciei | | |
| Trigger | Un anagajat nou trebuie sa intre in sistemul electronic al spitalului | | |
| Preconditions | PRE1:Sa fie angajat al spitalului  PRE2:Sa introduca datele despre sine si username si parola | | |
| Postconditions | POST1:Angajatul o sa aiba un cont nou pentru inregistrarea in sistem. | | |
| Normal flow | 1.Angajatul apasa pe butonul “Sign-up” si se deschide o fereastra noua  2.Angajatul introduce datele despre el si isi alege un username si o parola.  3.Alege daca este angajat la farmacie sau sectie  4.Apasa pe un buton si se creeaza cont cu succes.  5.Se intoarce la pagina de login | | |
| Alternative flows |  | | |
| Exceptions | E.1 Date eronate.  1.Utilizatorul uita sa introduca un camp, sau introduce gresit unele Campuri  2.Se afiseaza un mesaj de eroare  E.2 Exista cont deja  1.Exista deja un utilizatorul cu acelasi username, sau exista deja un cont pentru persoana respectiva. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | LogIn | | |
| Primary actor | Angajat Spital | Secondary actors | Sistemul spitalului |
| Description | Un angajat intra in contul sau personal pentru a efectua actiuni in sistemul electronic al spitalului | | |
| Trigger | Un angajat doreste sa efectueze o actiune in sistemul electronic al spitalului si trebuie sa se logheze. | | |
| Preconditions | PRE1: Angajatul trebuie sa aiba un cont | | |
| Postconditions | POST1: Se va deschide o fereastra corespunzatoare tipului de angajat | | |
| Normal flow | 1.Angajatul introduce userName si parola corespunzatoare contului  2.Afiseaza un mesaj de informare si de bine-venit.  3. Se va deschide o fereastra corespunzatoare tipului de angajat | | |
| Alternative flows |  | | |
| Exceptions | E.1 Date eronate.  1.Utilizatorul si parola introduse sunt gresite  2.Se afiseaza un mesaj de eroare | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | Honor Prescriptions | | |
| Primary actor | Operator Farmacie | Secondary actors |  |
| Description | Un farmacist doreste sa indeplineasca comanda pentru un client. Ii va da medicamentele dupa care va inregistra comanda ca fiind terminata in sistemul electronic al spitalului. | | |
| Trigger | Clientul a venit sa isi ia medicamentele prescrise. | | |
| Preconditions | PRE1.Angajatul este angajat ca farmacist. | | |
| Postconditions | POST1. Reteta va fi stearsa din terminalul farmaciei | | |
| Normal flow | 1.Utilizatorul selecteaza reteta  2.Apasa pe un buton pentru a indeplini comanda  3.Comanda nu mai este in terminal. | | |
| Alternative flows |  | | |
| Exceptions | E1.Nu mai sunt medicamente pe stoc.  1.Farmacia nu mai dispune de cantiatea necesara de medicamente.  2.Se afiseaza un mesaj de eroare.  3.Ramane neschimbata lista si comanda nu se efectueaza. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | View Prescriptions | | |
| Primary actor | Operator Farmacie | Secondary actors |  |
| Description | Pe pagina principala a farmacistului se pot vedea intr-un tabel toate retele care sunt valabile | | |
| Trigger | Farmacistul vrea sa verfice retele disponibile | | |
| Preconditions | PRE1.Angajatul este angajat ca farmacist. | | |
| Postconditions | POST1:Lista de comezni este afisata | | |
| Normal flow | 1.Farmacistul deschide interfata dupa ce se logheaza  2.Se afiseaza toate comenzile | | |
| Alternative flows |  | | |
| Exceptions |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | Create new Prescription | | |
| Primary actor | Operator sectie | Secondary actors | Operator farmacie |
| Description | Un doctor doreste sa inregistreze o reteta pentru un nou pacient, astfel incat sa o vada operatorii de la farmacie. Spune ce medicamente si cantitatea dorita, dupa care intra | | |
| Trigger | Un doctor doreste sa creeze o noua reteta. | | |
| Preconditions | PRE1.Angajatul este angajat ca doctor. | | |
| Postconditions | POST1.Reteta este inclusa in terminalul farmaciei | | |
| Normal flow | 1.Doctorul vede lista de medicamente  2.Selecteaza medicamentul din lista  3.Insereaza cantitatea  4.Apasa butonul de creare a retetei.  5.Se valideaza datele.  6.Se afiseaza un mesaj corespunzator  7.Comanda este acum in sistem, vizibila si de catre cei din farmacie | | |
| Alternative flows |  | | |
| Exceptions | E1.Reteta este goala  1.Nu este selectat niciun medicament cu o cantitate pozitiva  2.Se afiseaza un mesaj de eroare  E2.Cantitate negativa  1.Cantitatea introdusa pentru un medicament este negativa.  2.Se afiseaza un mesaj de eroare | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | View and Manage Medication Stock | | |
| Primary actor | Administrator | Secondary actors | Sistemul spitalului |
| Description | Administratorul intra pe pagina sa, de unde poate vedea in timp real stock-ul de medicamente si il poate modifica. | | |
| Trigger | Administratorul doreste sa vada situatia medicamentelor | | |
| Preconditions | PRE1.Angajatul este logat ca administrator | | |
| Postconditions | POST1.Se afiseaza lista cu medicamente si optiuniule de adaugare/stergere | | |
| Normal flow | 1.Administratorul este pe pagina sa dupa ce se logheaza  2.Administratorul vede tot stock-ul de medicamente | | |
| Alternative flows |  | | |
| Exceptions |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | Add medication | | |
| Primary actor | Administrator | Secondary actors | Sistemul spitalului |
| Description | Administratorul adauga un medicament nou, precizand numele si cantitatea medicamentului. | | |
| Trigger | Administratorul doreste sa adauge un medicament nou(O noua comanda de medicamente a ajuns) | | |
| Preconditions | PRE1.Angajatul este logat ca administrator | | |
| Postconditions | POST1:Medicamentul este adaugat in sistem si se actualizeaza lista cu medicamente disponibile de pe pagina doctorului | | |
| Normal flow | 1.Doctorul introduce numele noului medicament  2.Doctorul introduce cantitatea  3.Se valideaza cantitatea  4.Se adauga in sistem medicamentul  5.Se actualizeaza lista | | |
| Alternative flows | - | | |
| Exceptions | E1.Cantitate negativa  1.Cantitatea introdusa pentru un medicament este negativa.  2.Se afiseaza un mesaj de eroare | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | Delete Medication | | |
| Primary actor | Administrator | Secondary actors | Sistemul spitatlului |
| Description | Aministratorul sterge un medicament, selectand-ul din lista si apasand delete. | | |
| Trigger | Administratorul doreste sa stearga un medicametn(lipsteste de la ultima inventariere) | | |
| Preconditions | PRE1.Angajatul este logat ca administrator | | |
| Postconditions | POST1:Medicamentul este sters din sistem si se actualizeaza lista cu medicamente disponibile de pe pagina doctorului | | |
| Normal flow | 1.Administratorul selecteaza medicamentul pe care vrea sa il stearga.  2.Apasa pe butonul delete  3.Se afiseaza un mesaj de confirmare | | |
| Alternative flows | 1.Selectare multipla de medicamente sa fie sterse | | |
| Exceptions | E1.Apasare buton fara medicamente selectate  1.Se apasa butonul de delete fara medicamente selectate.  2.Se afiseaza un mesaj de eroare | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | Update Existing Medication | | |
| Primary actor | Administrator | Secondary actors | Sistemul medical |
| Description | Administratorul apasa pe un medicament din lista si introduce noua cantiate. Apasa pe butonul de update si se salveaza modificarea in sistem. | | |
| Trigger | Administratorul doreste sa modifice stock-ul(In urma unui inventar de ex.) | | |
| Preconditions | PRE1.Angajatul este logat ca administrator | | |
| Postconditions | POST1:Medicamentul este actualizat in sistem si se actualizeaza lista cu medicamente disponibile de pe pagina doctorului. | | |
| Normal flow | 1.Administratorul selecteaza medicamentul pe care vrea sa il updateze.  2.Introduce noua cantitate  3.Se valideaza cantitatea  4.Se afiseaza un mesaj de confirmare | | |
| Alternative flows | - | | |
| Exceptions | E1.Cantitate negativa  1.Cantitatea introdusa pentru un medicament este negativa.  2.Se afiseaza un mesaj de eroare | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 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 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)