# 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 |  | | |
| 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.

# 5.Urmarire bug-uri

O firma producatoare de software pune la dispozitia programatorilor si verificatorilor sai un sistem

prin care acestia pot sa comunice electronic. Astfel, fiecare dintre angajatii mentionati are la dispoziție

un terminal prin care:

• verificatorul poate înregistra un bug, dându-i o denumire si o descriere; imediat dupa

înregistrarea bug-ului, toti programatorii vad lista bug-urilor actualizata cu obiectul nou

introdus;

• programatorul vizualizeaza lista bug-urilor; de asemenea, programatorul poate selecta un bug

din lista si poate declansa un buton prin care declara ca bug-ul a fost eliminat, caz în care bugul

este scos din lista tuturor programatorilor.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-1: Logare | | |
| **Primary actor** | Verificator  Programator  Administrator | **Secondary actors** |  |
| **Description** | Un angajat al firmei(verificator/programator/administratoor) se logheaza la sistemul de urmarire al bug-urilor existente intr-o aplicatie | | |
| **Trigger** | Un angajat are de efectuat o sarcina si are nevoie sa se logheze in sistem | | |
| **Preconditions** | PRE-1. Angajatul are un cont de pe care sa se poata loga | | |
| **Postconditions** | POST-1. Angajatul primeste un mesaj ca autentificarea s-a realizat cu succes | | |
| **Normal flow** | 1. **Logare cu succes** 2. Angajatul se logheaza in sistem 3. Sistemul afiseaza un mesaj prin care informeaza utilizatorul ca autentificarea a fost realizata cu succes | | |
| **Alternative flows** | **1.1 Angajatul nu are un cont**  1a.Verificatorul/Programatorul nu are un cont cu care sa se poata loga in sistem  1a.1.Administratorul ii creaza un cont  Se revine la punctul 1 | | |
| **Exceptions** | **1.0.E1 Angajatul introduce credentiale eronate**  2a.Sistemul afiseaza un mesaj de eroare prin care informeaza utilizatorul ca datele introduse sunt invalide  2a.1.Utilizatorul reintroduce datele de autentificare  Se revine la punctul 2 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-2 Inregistrare bug | | |
| **Primary actor** | Verificator | **Secondary actors** |  |
| **Description** | Verificatorul detecteaza un bug in aplicatie si il inregistreaza in sistem, urmand sa fie informat ca bug-ul a fost inregistrat cu succes si ca urmeaza sa fie preluat de un programator | | |
| **Trigger** | Verificatorul doreste sa inregistreze un nou bug | | |
| **Preconditions** | PRE-1. Verificatorul este logat in sistem | | |
| **Postconditions** | POST-1. Verificatorul este informat ca bug-ul a fost inregistrat cu succes | | |
| **Normal flow** | **2.0 Inregistrare bug**  1.Verificatorul detecteaza un nou bug intr-o aplicatie si il inregistreaza in sistem  2.Sistemul afiseaza un mesaj prin care informeaza verificatorul ca bug-ul a fost inregistrat cu succes in sistem | | |
| **Alternative flows** | None | | |
| **Exceptions** | **2.0.E1 Bug-ul exista deja in sistem**  2a.Sistemul afiseaza un mesaj de eroare prin care informeaza verificatorul ca bug-ul introdus este deja inregistrat in sistem | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-3 Vizualizare bug-uri | | |
| **Primary actor** | Programator | **Secondary actors** | Verificator |
| **Description** | Programatorul cere sistemului lista cu bug-urile detectate de catre verificator, iar sistemul afiseaza lista cu bug-urile inregistrate de catre verificator | | |
| **Trigger** | Programatorul doreste sa vizualizeze lista cu toate bug-urile existente | | |
| **Preconditions** | PRE-1. Programatorul este logat in sistem | | |
| **Postconditions** | POST-1. Programatorul primeste lista cu bug-urile inregistrate in sistem | | |
| **Normal flow** | **3.0 Vizualizare bug-uri**  1. Programatorul cere sistemului lista cu bug-urile detectate de catre verificator  2.Sistemul afiseaza lista cu bug-urile inregistrate de catre verificator | | |
| **Alternative flows** | None | | |
| **Exceptions** | **3.0.E1 Nu exista bug-uri in sistem momentan**  2a.Sistemul afiseaza un mesaj de eroare prin care informeaza programatorul ca nu exista bug-uri in sistem | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-4 Eliminare bug | | |
| **Primary actor** | Programator | **Secondary actors** |  |
| **Description** | Programatorul rezolva un bug si il elimina din lista de bug-uri, iar sistemul ii afiseaza un mesaj prin care informeaza programatorul ca bug-ul a fost eliminat din lista | | |
| **Trigger** | Programatorul doreste sa elimine un bug din lista | | |
| **Preconditions** | PRE-1. Programatorul este logat in sistem | | |
| **Postconditions** | POST-1. Programatorul primeste un mesaj ca bug-ul a fost eliminat din lista | | |
| **Normal flow** | **4.0 Vizualizare bug-uri**  1.Programatorul rezolva un bug si il elimina din lista de bug-uri  2.Sistemul ii afiseaza un mesaj prin care informeaza programatorul ca bug-ul a fost eliminat din lista | | |
| **Alternative flows** | None | | |
| **Exceptions** | **4.0.E1 Bug-ul nu s-a putut elimina**  2a.Sistemul afiseaza un mesaj de eroare prin care informeaza programatorul ca nu s-a putut elimina bug-ul din cauza unei erori tehnice | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-5 Adaugare angajat | | |
| **Primary actor** | Administrator | **Secondary actors** |  |
| **Description** | Firma angajeaza o noua persoana in echipa iar administratorul ii creaza un cont, in functie de pozitia pe care o ocupa(verificator sau programator) | | |
| **Trigger** | Administratorul doreste sa creeze un cont unui angajat | | |
| **Preconditions** | PRE-1. Administratorull este logat in sistem | | |
| **Postconditions** | POST-1. Administratorul primeste un mesaj conform caruia contul a fost creat cu succes | | |
| **Normal flow** | **5.0 Adaugare angajat**  1.Administratorul doreste sa creeze un cont nou  2.Sistemul deschide o fereastra in care administratorul trebuie sa completeze datele noului cont  3.Administratorul introduce datele necesare  4.Sistemul informeaza administratorul ca noul cont a fost creat cu succes si poate fi utilizat | | |
| **Alternative flows** | None | | |
| **Exceptions** | **5.0.E1 Administratorul introduce date eronate**  3a.Sistemul afiseaza un mesaj de eroare prin care informeaza administratorul ca datele introduse nu sunt valide  Se revine la pasul 3 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-6 Eliminare angajat | | |
| **Primary actor** | Administrator | **Secondary actors** |  |
| **Description** | Un angajat pleaca de la firma iar administratorul ii sterge contul din baza de date | | |
| **Trigger** | Administratorul doreste sa stearga contul unui fost angajat | | |
| **Preconditions** | PRE-1. Administratorull este logat in sistem | | |
| **Postconditions** | POST-1. Administratorul primeste un mesaj conform caruia contul a fost sters cu succes | | |
| **Normal flow** | **6.0 Eliminare angajat**  1.Administratorul doreste sa stearga un cont  2.Sistemul deschide o fereastra in care administratorul trebuie sa caute contul pe care doreste sa il stearga  3.Administratorul sterge contul  4.Sistemul informeaza administratorul ca a fost eliminat cu succes contul | | |
| **Alternative flows** | None | | |
| **Exceptions** | **6.0.E1 Contul nu s-a putut elimina**  3a. Sistemul afiseaza un mesaj de eroare prin care informeaza administratorul ca nu s-a putut elimina contul din cauza unei erori tehnice  Se revine la pasul 3  **6.0.E2 Contul nu exista**  2a. Sistemul afiseaza un mesaj de eroare prin care informeaza administratorul ca nu exista contul cautat  Se revine la pasul 2 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-7 Modificare angajat | | |
| **Primary actor** | Administrator | **Secondary actors** |  |
| **Description** | Un angajat isi modifica anumite date personale iar administratorul inregistreaza aceste modificari in sistem | | |
| **Trigger** | Administratorul doreste sa modifice contul unui angajat | | |
| **Preconditions** | PRE-1. Administratorull este logat in sistem | | |
| **Postconditions** | POST-1. Administratorul primeste un mesaj conform caruia contul a fostmodificat cu succes | | |
| **Normal flow** | **7.0 Modificare angajat**  1.Administratorul doreste sa modifice un cont  2.Sistemul deschide o fereastra in care administratorul trebuie sa completeze noile date ale contului  3.Administratorul introduce datele necesare  4.Sistemul informeaza administratorul ca acel cont a fost modificat cu succes | | |
| **Alternative flows** | None | | |
| **Exceptions** | **7.0.E1 Administratorul introduce date eronate**  3a.Sistemul afiseaza un mesaj de eroare prin care informeaza administratorul ca datele introduse nu sunt valide  Se revine la pasul 3 | | |