Use case – restul cazurilor de utilizare

**ID and Name:** Vizualizare lista de bug-uri actualizata

**Primary Actor:** Verificator

**Secondary Actor:** Programator

**Description:** Verificatorul primeste in orice moment de accesare al aplicatiei lista de bug-uri ale programatorilor actualizata

**Trigger:** Codul cel putin unui programator din sistem contine bug-uri

**Preconditions:** A fost incarcat cel putin un cod cu bug-uri

**Postconditions:** Verificatorul poate decide daca/cand va trimite bug-ul inapoi programatorului pentru a fi rezolvat.

**Normal Flow:**

1. Programatorul trimite munca/codul verificatorului
2. Sistemul de verificare evalueaza codul si gaseste unul sau mai multe bug-uri
3. Verificatorul va vedea lista de bug-uri din cod si poate decide cum sa actioneze
4. Selectand un bug, verificatorul decide daca il va intregistra programatorului pentru rezolvare
5. Daca se intampla pasul anterior, programatorul vede bug-ul in lista personala de bug-uri si va incerca sa il rezolve

**Alternative Flows:** None

**Exceptions:** Lista de bug-urinu este actualizata corect(nu toti verificatorii vad lista actuala de bug-uri)

**ID and Name:** Autentificare

**Primary Actor:** Programator

**Secondary Actor:** Verificator

**Description:** Un verificator/programator se logheaza in aplicatia de urmarire de bug-uri

**Trigger:** Un programator doreste sa isi inregistreze un cod unui verificator pentru a fi rezolvat./Un verificator doreste sa rezolve bug-urile unui cod trimis de un programator.

**Preconditions:** Programatorul/Verificatorul trebuie sa aiba un cont in aplicatie, format dintr-un username si o parola.

**Postconditions:** Aplicatia gestioneaza informatia de logare si directioneaza utilizatorul in interfata dorita (este programator sau verificator).

**Normal Flow:**

1. Un verificator se logheaza si intra in meniul destinat verificatorilor.

**Alternative Flows:**

1. Programatorul se logheaza si intra in meniul destinat programatorului.

**Exceptions:** Un cont dispare/ nu mai functioneaza/ a fost sters sau modificat in mod eronat.

Un programator ajunge in meniul predestinat verificatorilor si vice-versa.

**ID and Name:** Vizualizare bug-uri personale de rezolvat

**Primary Actor:** Programator

**Secondary Actor:** Verificator

**Description:** Un programator vede bug-urile aflate in programul/programele lui.

**Trigger:** Un programator a avut cel putin un cod care contine bug-uri

**Preconditions:** Verificatorul inregistreaza in munca programatorului unul sau mai multe bug-uri.

**Postconditions:** Programatorul poate selecta un bug din lista si sa anunte astfel verificatorul cum ca bug-ul ar fi fost rezolvat

**Normal Flow:**

1. Programatorul se inregistreaza in aplicatie si vede lista de bug-uri personale.
2. Daca a avut cel putin un cod evaluat cu bug-uri dde catre verificator, acesta va aparea in lista de bug-uri personale.
3. Programatorul selecteaza un bug pe care il va rezolva si va anunta la sfarsit verificatorul cum ca a incercat rezolvarea bug-ului
4. Verificatorul poate reevalua codul cu bug-uri, caz in care se repeta primii 3 pasi.
5. Bug-ul rezolvat va aparea intr-un mod diferit in lista de bug-uri personale ale programatorului, astfel programatorul va stii ca bug-ul este rezolvat.

**Alternative Flows:** None.

**Exceptions:** Unele bug-uri personale nu apar in lista/ apar ca fiind rezolvate in mod eronat.

**ID and Name:** Notificare eliminare Bug

**Primary Actor:** Verificator

**Secondary Actor:** Programator

**Description:** Verificatorul anunta programatorul ca un bug este rezolvat in totalitate.

**Trigger:** Un cod scris de un programator este evaluat cel putin o data ca fiind cu bug-uri.

**Preconditions:** Cel putin un cod al unui programator a continut bug-uri.

**Postconditions:** Programatorul primeste notificarea si vede intr-un mod specific ca bug-ul este rezolvat (in lista de bug-uri personale).

**Normal Flow:**

1. Programatorul isi inregistreaza codul, iar verificatorul inregistreaza in cadrul codului unul sau mai multe bug-uri.
2. Programatorul incearca sa rezolve bug-ul pana cand verificatorul ii valideaza munca.
3. Verificatorul notifica programatorul ca bug-ul e rezolvat
4. Programatorul vede notificare in lista de bug-uri si poate alege daca vrea sa elimine aceasta notificare din lista.

**Alternative Flows:** None.

**Exceptions:** Notificarea bug-ului rezolvat este eronata, programatorul rezolvand mai repede acest bug sau verificatorul evaluand corect in mod eronat un anume cod.

**ID and Name:** Confirmare de eliminare a bug-ului

**Primary Actor:** Verificator

**Secondary Actor:** Programator

**Description:** Verificatorul realizeaza ca unul sau mai multe bug-uri au fost rezolvate.

**Trigger:** Un cod scris de un programator este evaluat cel putin o data ca fiind cu bug-uri.

**Preconditions:** Cel putin un cod al unui programator a continut bug-uri.

**Postconditions:** Verificatorul poate decide daca/cand va anunta ca programatorul a reusit rezolvarea bug-ului.

**Normal Flow:**

1. Programatorul trimite un cod fara bug-uri/curatat de bug-uri.
2. Verificatorul vede ca acest cod este “bug-free” si ii apare in mod specific ca fiind un cod curat.
3. Verificatorul poate decide daca trimite notificare programatorului cum ca bug-ul este eliminat, astfel bug-ul nu va mai fi luat in considerare de verificator si va disparea din lista de bug-uri.

**Alternative Flows:** None.

**Exceptions:** -

**ID and Name:** Trimiterea modificarilor programatorului

**Primary Actor:** Verificator

**Secondary Actor:** Programator

**Description:** Verificatorul trimite programatorului statusul unui cod (“bug-free” sau cu bug-uri).

**Trigger:** Un cod scris este evaluat de catre un verificator

**Preconditions:** A fost trimis cel putin un cod de catre un programator.

**Postconditions:** Programatorul primeste un status de rezolvare a bug-ului sau un status cum ca a fost gasit cel putin un bug.

**Normal Flow:**

1. Verificatorul evalueaza munca unui programator.
2. Verificatorul trimite notificare programatorului ca respectivul cod este “bug-free”.

**Alternative Flows:**

1. Verificatorul evalueaza munca unui programator.
2. Verificatorul trimite notificare programatorului ca respectivul cod contine cel putin un bug.
3. Programatorul vede bug-ul in lista personala si incearca sa rezolve acest bug.
4. Se poate repeta acest scenariu pana la ajungerea scenariului principal.

**Exceptions:** -