FII Documents Flow

**Echipa:**

Bejan Octavian

Chilaboc Ecaterina

Glodeanu Irina

Macariu Camelia

Manolache Ecaterina

Rusu Gabriel

Stolnicu Simona

Ciofu Vlad

Grosu Alexandru

Hardon Elena

Ursachi Georgiana

**Grupa:** 2A6

**Cuprins:**

1.Descriere ……………………………………………………3

2.Domeniu …………………………………………………….3

3.Actionari/Interese …………………………………………...4

4.Actori&Obiective …………………………………………...4

5.Data Layer…………………………………………………...4

6.Scenarii de utilizare………………………………………….5

1. **Descriere:**

Aplicatia presupune un sistem de management al documentelor necesare studentilor de la Facultatea de Informatica Iasi. Sistemul va permite atat inregistrarea cererilor legate de tipuri de adeverinte necesare dar si un sistem de notificare legat de statusul cererii (email, SMS)

Acestia pot face cereri pentru anumite documente iar aplicatia manipuleaza datele lor in asa fel incat procesul este automatizat din punct de vedere al inforatiilor utilizatorului (se iau automat din baza de date). Date aditionale necesare vor fi introduse de user si prelucrate in vederea intocmirii documentelor.

Bazele de date tin informatii necesare despre toate entitatile aplicatiei (Studenti, Persoane din Consiliul de Administratie, Documente) . Aceste date sunt accesate prin request-uri generate de partea de Front-End prin metode specifice.

Aplicatia pune constrangeri din punctul de vedere al permisiunilor studentilor in a face o cerere care nu i se cuvine (de exemplu, daca e la taxa, nu va primi acces la o cerere adresata celor care sunt la buget; daca are deja emis un document care inca mai este in vigoare nu i se permite sa mai faca o cerere de acelasi tip pana expira...)

1. **Domeniu**

Se vor descrie scenariile de utilizare ale aplicatiei atunci cand este accesata de un student care doreste sa obtina un document , cat si de personalului de administratie pentru elibrarea lor.

**3.Actionari/Interese**

-Student : isi doreste sa obtina repede si usor documentul dorit

-Secretara : completarea, verificarea si emiterea mai rapida a documentelor cerute

-Aplicatie : sa functioneze eficient, rapid si sa fie stabila

-Document : sa ajunga rapid si usor la destinatar

-Administrator : sa faca mentenanta cat mai rar si sa merga aplicatia

**4.Actori & Obiective**

-Student : primirea documentului

-Secretara : eliberarea documentului

-Aplicatie : gestionarea interactiunii dintre studenti si personalul de administratie

-Document : rezultatul final

-Administrator : sa administreze aplicatia cat mai corect

**5. Data Layer**

Datele despre studenti sunt stocate in tabela IStud care ne vor ajuta atat pentru logare cat si pentru popularea cu date a cererilor pentru documente.

Tabela in care tinem informatiile despre personalul de administratie (IAdministrare) ofera date despre userii cu anumite permisiuni aditionale : validarea/respingerea unei cereri, trimitere de notificari ...

Tabelele care tin de documente sunt TipuriDocument si RequestHistory in care vom avea informatii atat despre documentele disponibile cat si despre cererile efectuate.

**6.Scenarii de utilizare**

**6.1.Logarea student**

**6.1.1. Obiectiv/Context**

Intrarea studentului in aplicatie pe baza unui username si a unei parole.(contul de fenrir)

**6.1.2 Scenariu/Pasi**

1. Userul introduce date care vor fi comparate cu anumite campuri existente din tabela IStud si daca exista un match atunci se trece la verificarea parolei.

2. Daca si parola se potriveste contului aferent atunci logarea a avut succes.

**6.1.3 Extensii**

3a. Daca datele introduse nu sunt valide (fie username-ul, fie parola) atunci studentului i se cere sa reintroduca datele.

**6.2. Alegere document necesar de catre student**

**6.2.1. Obiectiv/Context**

Selectarea dintr-un meniu a documentului dorit.

**6.2.2 Scenariu/Pasi**

1. Cand alege documentul, este accesata tabela TipuriDocument. Din aceasta se vor lua informatii cu privire la fiecare tip de document.

2. In functie de campul “format” din aceasta se vor extrage din tabela IStud informatiile necesare pentru autocompletarea documentului in cauza.

**6.3. Completare campuri aditionale pentru cerere**

**6.3.1. Obiectiv/Context**

Introducerea unor noi informatii necesare pentru completarea tuturor campurilor

**6.3.2 Scenariu/Pasi**

1. Dupa selectarea cererii dorite (vezi punctul 6.2) , in cazul in care mai este nevoie de informatii aditionale, studentul va introduce acele date in campuri speciale.

2. In functie de campul “format”din tabela TipuriDocument se va verifca corectitudinea datelor introduse.

3. Aceste date sunt mai apoi stocate pentru a fi luate mai tarziu cand se vor genera documentele.

**6.3.3 Extensii**

3a. Daca datele speciale introduse nu respecta acel “format” se va afisa un mesaj de eroare si se va face un request pentru reintroducerea datelor (in format corect).

**6.4. Trimiterea cererii**

**6.4.1. Obiectiv/Context**

Trimitrea datelor catre partea administrativa si oferirea accesului la datele necesare completarii documentului.

**6.4.2 Scenariu/Pasi**

1. Apasarea butonului de “Send” si acctionarea unui ActionListener care determina stocarea temporara a datelor.

**6.5. Logare pe partea administrativa**

**6.5.1. Obiectiv/Context**

Intrarea unei persoane din administratie in aplicatie pe baza unui username unic si a unei parole.

**6.5.2 Scenariu/Pasi**

1. Userul introduce date care vor fi comparate cu anumite campuri existente din tabela IAdministrare si daca exista un match atunci se trece la verificarea parolei.

2. Daca si parola se potriveste contului aferent atunci logarea a avut succes.

**6.5.3 Extensii**

3a. Daca datele introduse nu sunt valide (fie username-ul, fie parola) atunci utilizatorului i se cere sa reintroduca datele.

**6.6. Generarea documentului**

**6.6.1. Obiectiv/Context**

Inserarea in template-ul tipului de document a datelor puse a dispozitie de catre utilizator.

**6.6.2 Scenariu/Pasi**

1. Preluarea template-ului stocat.

2. Inserarea datelor in campul corect.

**6.7. Alegerea unei categorii de cereri pentru aprobare**

**6.7.1. Obiectiv/Context**

Selectarea documentelor de aprobat specifice unei categorii.

**6.7.2 Scenariu/Pasi**

1. Dupa actionarea unui buton se va afisa lista documentelor care trebuiesc aprobate de catre persoana din administratie.

2. Din aceasta lista se vor marca cererile acceptate si sterge cererile eronate.

**6.8. Trimitere notificare**

**6.8.1. Obiectiv/Context**

Trimiterea unui mesaj text prin retea.

**6.8.2 Scenariu/Pasi**

1. Generarea unui mesaj specific in functie de caz (succes/esec).

2. Trimitere catre student.

**6.9. Vizualizare istoric pe luni**

**6.9.1. Obiectiv/Context**

Vizualizare de catre un user(personalul de administratie sau student) a istoricului cererilor in functie de luna selectata.

**6.9.2 Scenariu/Pasi**

**6.9.2.1 In cazul Studentului**

1. Daca logarea a avut succes atunci studentul poate alege sa vizualizeze lista cererilor trimise (stocate in tabela RequestHistory) intr-o anumita luna (selectata prin intermediul numarului din cadrul unui an) si stadiul lor (accepted/rejected/pending).

2. Dupa selectarea lunii, se vor face fetch-uri in baza de date RequestHistory filtrand dupa luna.

3. Rezultatele vor fi transmise pentru display pe ecran.

**6.9.2.2 In cazul persoanei din administratie**

1. Daca logarea a avut succes atunci aceasta persoana poate alege sa vizualizeze lista cererilor trimise (stocate intr-o baza de date) intr-o anumita luna (selectata prin intermediul numarului din cadrul unui an) si stadiul lor (accepted/rejected/pending) sau sa introduca numele unui student pentru a-i vedea istoricul.

2. Dupa selectarea lunii/numelui de student, se vor face fetch-uri in baza de date RequestHistory respectiv IStud filtrand dupa luna/numele studentului.

3. Rezultatele vor fi transmise pentru display pe ecran.

**6.9.3 Extensii**

3a. In cazul in care persoana din administratie introduce date invalide sau inexistente, rezultatul fetch-ului va fi nul .