Your Books Everywhere!

Analysis and Design Document

Student: Crisan Cristian-Razvan

**Group: 30238**

Table of Contents

1. Requirements Analysis 3

1.1 Assignment Specification 3

1.2 Functional Requirements 3

1.3 Non-functional Requirements 3

2. Use-Case Model 3

3. System Architectural Design 3

4. UML Sequence Diagrams 3

5. Class Design 3

6. Data Model 3

7. System Testing 3

8. Bibliography 3

1. Requirements Analysis

# Assignment Specification

Pentru aceasta aplicatie am avut de realizat un sistem de manageriat carti intr-o biblioteca sau o librarie. Aplicatia are 3 tiputi de utilizator: utilizatorul normal, care poate sa imprumute carti, angajati care pot sa aprobe imprumuturile de carti si sa adauge carti si administartorul retelei care poate sa modifice baza de date si sa adauge, modifice sau stearga utilizatori.

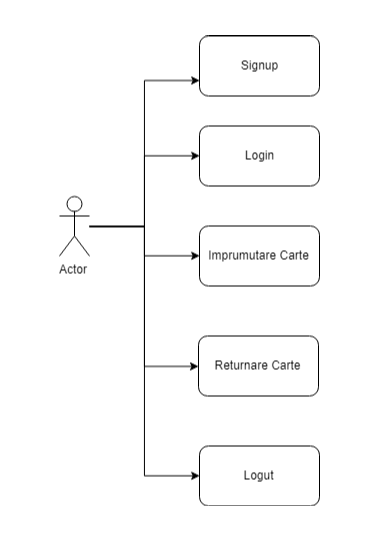
# Functional Requirements

Functiile cerute sunt: creearea de conturi pentru utilizatori, posibilitatea filtrarii cartilor dupa data, autor, titlu si gen, posibilitatea de a imprumuta o carte si generarea recomandarilor in mod dinamic.

# Non-functional Requirements

Aplicatia este una web, din aceasta cauza nu are nevoie de multe resurse pentru a functiona, a fost folosit framework-ul django si in acest mod s-a construti o aplicatie sigura.

2. Use-Case Model



Use-case: Signup/Login

Level: user-goal level

Primary actor: utilizatorul

Mai success scenario: utilizatorul introduce datele corecte si se logheaza sau isi face cont

Extensions: Datele introduse nu sunt corecte

Use-case: Imprumutare Carte

Level: user-goal level

Primary actor: utilizatorul

Mai success scenario: utilizatorul alege cartea pe care doreste sa o imprumute si o aduga in lista lui.

Extensions: deoarece se adauga intr-o lista, daca cartea nu este disponibila, nu exista cazuri alternativa

Use-case: Returnare Carte

Level: user-goal level

Primary actor: utilizatorul

Mai success scenario: utilizatorul returneaza cartea

Extensions: -

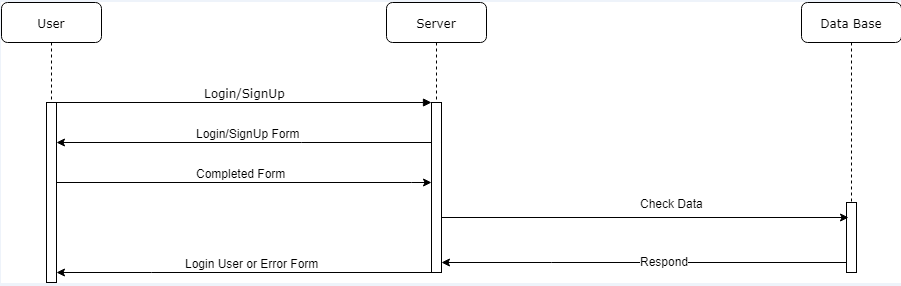
3. System Architectural Design

**3.1 Architectural Pattern Description**

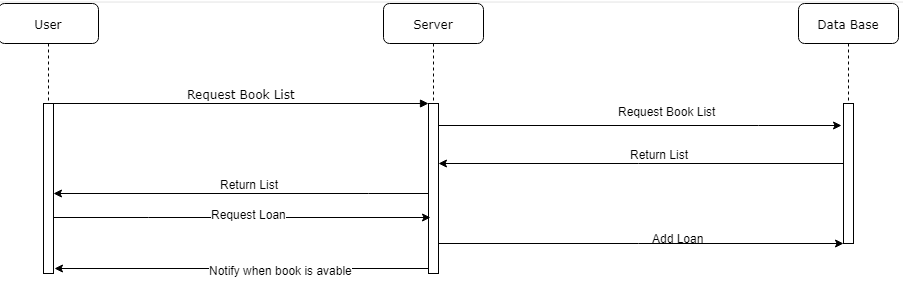
Aceasta aplicatie folosete framework-ul web Django scris in python. Deoarece aceasta este o aplicatie web, foloseste arhitectura client-server.

4. UML Sequence Diagrams

In urmatoarea flow-diagram am demostrat modul de utilizare a functioe de log-in sau de sign-up. Acesta se realizeaza astfel: utilizatorul anunta serverul ca dorese sa se intregistreze sau sa se logheze, acesta uramana sa primeasca un sablon pentru datele necesare. Dupa ce le completeaza, daca acestea sunt corecte, Se realizeaza operatiounea.



Pentru operatiunea de inchiriere a cratiilor, utilizatorul are nevoie de lista cartilor, din care is alege una, anunta serverul ca doreste sa o inchireze si este notificat cant cartea este in stoc.



5. Class Design

**5.1 Design Patterns Description**

Deoarece am folosit Django, structura codului este urmatoarea: acesta este impartit in mai multe “aplicatii”. Una pentru fiecare structura din baza de date: Book, BookLoan, Accout.

Aplicatia principala LibraryProject este impartita in uramtoarele componente: setari, url-rui, si vizualizari fiecare avand o functionalitate critica.

Setarile sunt folosite la pornirea aplicatiei pentru a creea mediul in care ruleaza aplicatia.

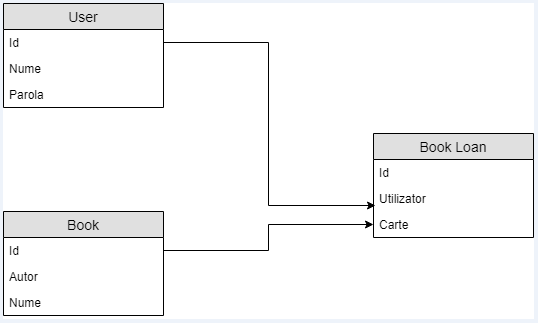
Modulul urls descrie redirectionare pentru fiecare pagina din aplicatia noastra, uramand ca aceasta desscriere sa fie implementata in views.

Fiecare “aplicatie” contine aceste module, plus froms, care descrie modul in care aceea “aplicatie” este stocata in baza de date.

Este folosit un observer care notifica utilizatorii care au cerut sa imprumute o carte cand aceasta carte este libera printr-un email.

6. Data Model

Datele sunt impartite in 3 tabele: unul cu utilizatori, unul cu carti si unul cu imprumuturi.



7. System Testing

Deoarece aplicatia este una web si tot ce returneaza sunt fisiere html, nu am putu sa teste aceste functionalitati. Singurul lucru testat a fost baza de date.

8. Bibliography

<https://www.djangoproject.com/>