Banca Online

Analysis and Design Document

Student: Crisan Cristian-Razvan

**Group: 30238**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <x.x> | <details> | <name> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

I. Project Specification 4

II. Elaboration – Iteration 1.1 4

1. Domain Model 4

2. Architectural Design 4

2.1 Conceptual Architecture 4

2.2 Package Design 4

2.3 Component and Deployment Diagrams 4

III. Elaboration – Iteration 1.2 4

1. Design Model 4

1.1 Dynamic Behavior 4

1.2 Class Design 4

2. Data Model 4

3. Unit Testing 4

IV. Elaboration – Iteration 2 4

1. Architectural Design Refinement 4

2. Design Model Refinement 4

V. Construction and Transition 5

1. System Testing 5

2. Future improvements 5

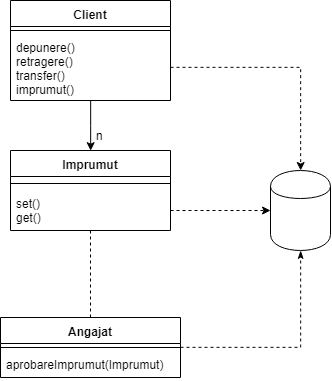
VI. Bibliography 5

# Project Specification

Proiectul are la baza o aplicatie bancara, aceasta fiind una web. Utilizatorul normal poata sa isi vada situatia financiara, sa aplice pentru credite, sa plateasca facturi, etc. Un alt tip de utilizator este angajatul bancii care poate sa aprobe/respinga solicitarea de credit pentru utilizatori si are acces la datele fiecarui utilizator. Ultimul tip de cont este cel de administrator care poate sa modifice datele fiecarui utilizator si sa adauge conturi de angajat.

# Elaboration – Iteration 1.1

# Domain Model



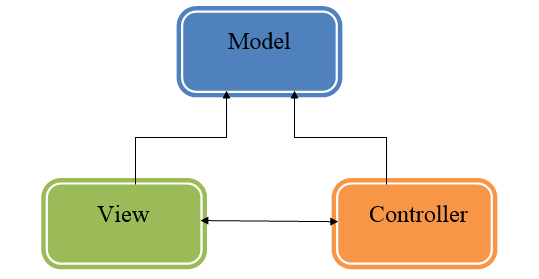
# Architectural Design

## Conceptual Architecture

Aceasta aplicatie folosete framework-ul web Django scris in python, acesta are la baza pattern-ul Model-View-Controller, redenumit sub forma Model-View Template.

Patternul Model-View-Controller, acesta imparte aplicatia in trei parti interconectate, pentru a putea separa reprezentarea interna a informatiilor.

## 2.2 Package Design



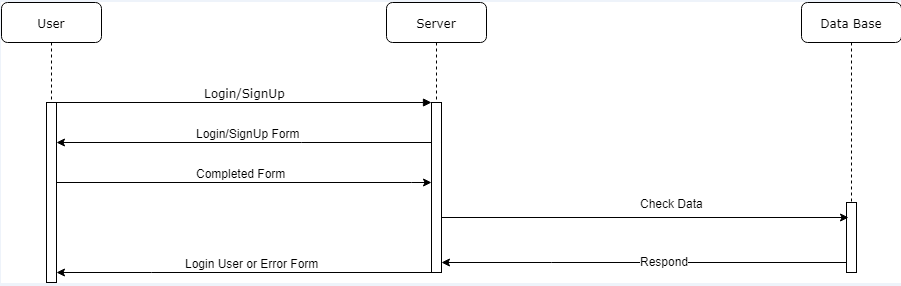
## Component and Deployment Diagrams

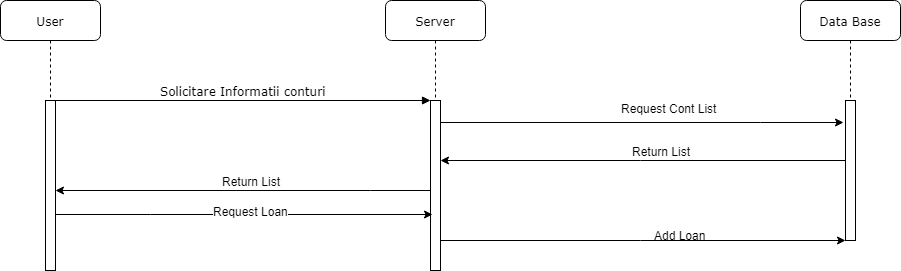
# 

# Elaboration – Iteration 1.2

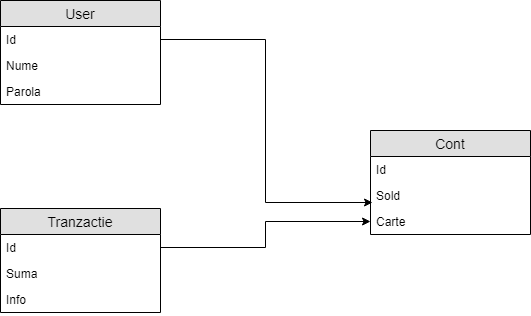
# Design Model

## Dynamic Behavior





# Data Model



# Unit Testing

*[Present the used testing methods and the associated test case scenarios.]*

# Elaboration – Iteration 2

# Architectural Design Refinement

*[Refine the architectural design: conceptual architecture, package design (consider package design principles), component and deployment diagrams. Motivate the changes that have been made.]*

# Design Model Refinement

## *[Refine the UML class diagram by applying class design principles and GRASP; motivate your choices. Deliver the updated class diagrams.]*

# Construction and Transition

# System Testing

Testele se vor efectua pentru fiecare operatiune in parte atat pentur utilizatorul normal care poate sa: scoata bani, sa transfere bani, sa plateasca facturi sau alte servicii, iar pentru angajatii se vor testa functionalitatiile de aprobare a creditelor si de depunere a banilor in contrurile utilizatorilor.

# Future improvements

Se pot adauga multe inbunatatiri, ca de exaplu: posibilitatea de a cumpara actiuni la bursa, de a avea mai gestiona mai multe conturi in mai multe valute.

# Bibliography

1. <https://docs.djangoproject.com/en/2.2/>
2. <https://docs.python.org/3/>