Pet Shop and Pet Rescue Center

Analysis and Design Document

Student: Denisa-Maria Herlea

**Group: 30238**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 18/04/2023 | 1.0 | Creare document | Denisa-Maria Herlea |
| 30/04/2023 | 1.1 | Actualizare | Denisa-Maria Herlea |
|  |  |  |  |
|  |  |  |  |

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

*[Present the project specification]*

# Elaboration – Iteration 1.1

# Domain Model

Modelul de domeniu este o reprezentare abstractă a entităților importante în cadrul domeniului de aplicare a sistemului. Acesta poate fi utilizat pentru a defini clasele sistemului și pentru a ajuta la organizarea lor într-o manieră logică.

# Architectural Design

## Conceptual Architecture

Arhitectura sistemului este *client-server*, arhitectura implicită a framework-ului Django.

În această arhitectură, există două componente principale: serverul și clientul. Serverul reprezintă partea din aplicație care furnizează servicii, de exemplu, partea care procesează cereri și furnizează răspunsuri. Clientul este partea care utilizează aceste servicii și interacționează cu utilizatorul.

Serverul este reprezentat de aplicația Django, care procesează cererile HTTP primite de la client și furnizează răspunsurile corespunzătoare. Clientul este reprezentat de browser-ul web, care afișează paginile web generate de server și permite utilizatorilor să interacționeze cu aplicația. Folosirea arhitecturii client-server oferă o separare clară între partea de prezentare (interfața utilizatorului) și partea de logică a aplicației.

## Package Design

*[Create a package diagram]*

## Component and Deployment Diagrams

**Deployment Diagram:**

# 

**Component Diagram:**

O imagine care conține diagramă, linie, text, Plan

Descriere generată automat

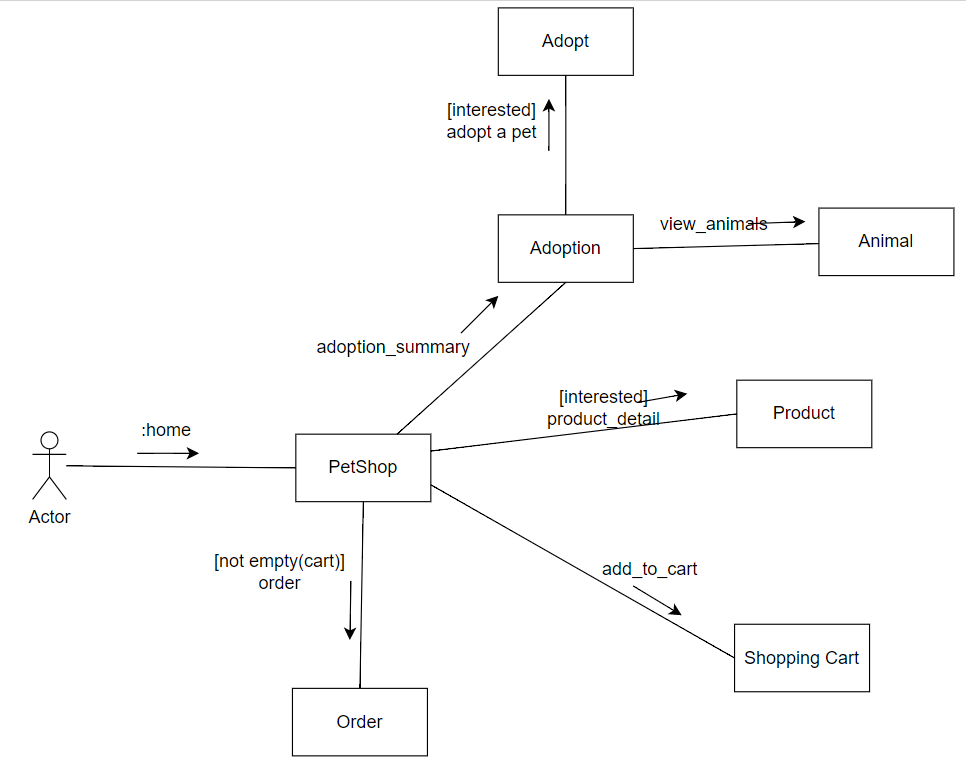
# Elaboration – Iteration 1.2

# Design Model

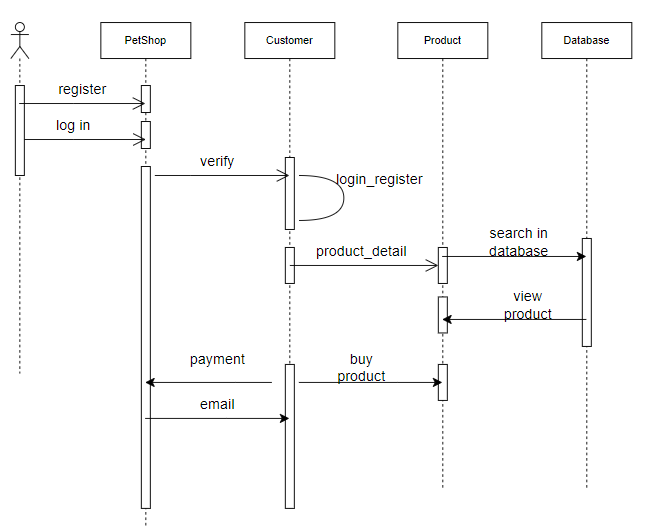
## Dynamic Behavior



**Communication Diagram**



**Sequence Diagram**

****

## Class Design

*[Create the UML class diagram; apply GoF patterns and motivate your choice]*

# Data Model

Doua dintre principalele modele de date sunt Category si Product.

**Category** contine campurile:

* *nume*
* *slug*

**Product** contine campurile:

* *category*
* *created\_by*
* *name*
* *description*
* *image*
* *slug*
* *price*
* *in\_stock*
* *created*
* *updated*
* *objects*
* *products*

Modelul de date specific adoptiei animalelor contine clasele Adoption si Animal cu urmatoarele campuri:

**Adoption**:

* *nume*
* *prenume*
* *adresa*
* *oras*
* *telefon*
* *created*
* *updated*

**Animal:**

* *added\_by*
* *name*
* *description*
* *image*
* *slug*
* *available*
* *created*
* *updated*
* *objects*
* *products*

In final, plasarea comenzii este realizata prin intermediul clasei **Order** cu urmatoarele campuri:

* *nume*
* *prenume*
* *adresa*
* *oras*
* *cod\_postal*
* *created*
* *updated*
* *total\_paid*

# 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

*[Describe how you applied integration testing and present the associated test case scenarios.]*

# Future improvements

Unele posibile îmbunătățiri viitoare pentru Petshop includ:

* Funcționalități de căutare și filtrare avansate: În prezent, utilizatorii pot filtra produsele după categorii, dar există oportunitatea de a îmbunătăți acest lucru prin adăugarea de filtre suplimentare, cum ar fi prețul, ratingul, dimensiunea, greutatea etc.
* Integrarea cu platforme de social media: Integrarea cu platforme de social media precum Facebook și Instagram ar face ca promovarea magazinului să fie mai ușoară și mai eficientă. Astfel, clienții pot vedea și distribui produsele pe care le-au achiziționat sau pe care le-au adăugat la lista de dorinte.
* Implementarea unui program de loialitate: Un program de loialitate ar face ca clienții fideli să fie recompensați cu puncte sau reduceri, ceea ce ar stimula cumpărăturile.
* Îmbunătățirea design-ului.

# Bibliography

<https://creately.com/diagram/example/hbscitcx3/sequence-diagram-for-ecommerce>

<https://www.researchgate.net/figure/e-Commerce-platform-deployment-diagram-refined-from-ACTIVE-designs_fig5_228673759>

<https://app.diagrams.net/>

<https://creately.com/diagram/example/gqz0nrua1/component-diagram-for-online-shopping-system-classic>

<https://www.uml-diagrams.org/examples/online-shopping-uml-communication-diagram-example.html>