Camping Manager

Analysis and Design Document

Student:Cîmpan Roxana-Adriana

**Group:30233**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 31/mai/17 | 2.0 | Domain Model  Architectural Design  Component and Deployment diagrams | Cîmpan Roxana-Adriana |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

Proiectul implementat are doua tipuri de utilizatori. Fiecare dintre acestia au propriile operatii pe care le pot efectua. Prin umare prezint cei doi utilizatori:

1. -**user regular** (**user normal**)

-poate sa rezerve casute din incinta camping-ului(pentru rezervare va fi necesar sa completeze anumite campuri, de exemplu: nume, prenume, adresa, numar de telefon, adresa de mail).

-poate sa vizualizeze fotografiile disponibile intr-un album photo

-poate sa lase comentarii, observatii, reclamatii, sugestii sau aprecieri

-poate sa vizualizeze casutele care sunt disponibile

-poate sa vizualizeze perioada in care este ocupata o anumita casuta

1. -**administrator**:

-poate sa vizualizeze comentariile, observatiile, reclamatiile, sugestiile sau aprecerile

-poate sa vizualizeze anumite rapoarte despre casuta

-poate sa stearga anumite rezervari

# Architectural Design

## Conceptual Architecture

Una dintre arhitecturile pe care doresc sa o utilizez: 3-tier. Aceasta este o arhitectura client-server. Aceasta arhitectura se bazeaza pe:



1.Presentation tier

2.Bussines Logic tier

3.Data tier

* Presentation tier=UI layer

-reprezinta nivelul superior al aplicatiei

-mai précis, este layerul la care utilizatori pot avea acces direct (pagina WEB sau GUI-in cazul proiectului de fata)

* Bussines Logic tier

-controleaza functionalitatea unei aplicatii

* Data tier

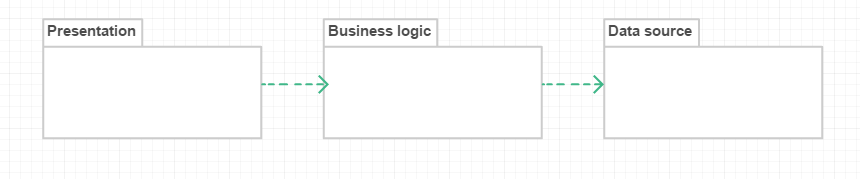
-include mecanisme de persistenta

* Fisiere partajate
* Servere da baze de date

-layer-ul de acces incapsuleaza mecanismele de persistenta

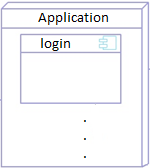
* Codul pentru fiecare layer terbuie sa fie in fisiere separate care pot fi mentinute separate
* Fiecare layer poate cantine numai cod care face parte din acel layer
* bussines logic se poate afla numai in Bussines layer
* presentation logic se poate afla numai in Presentation layer
* data acces logic se poate afla numai in Data Acces Layer

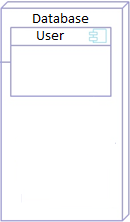
## Package Design

**

## Component and Deployment Diagrams

# 



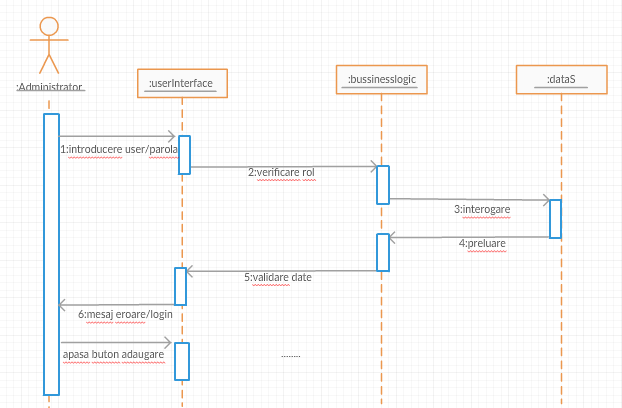




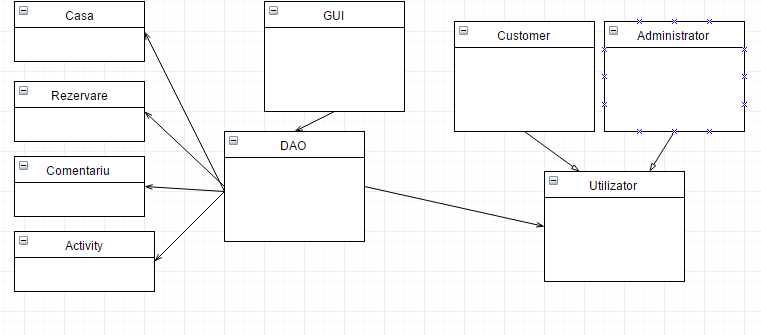
# Elaboration – Iteration 1.2

# Design Model

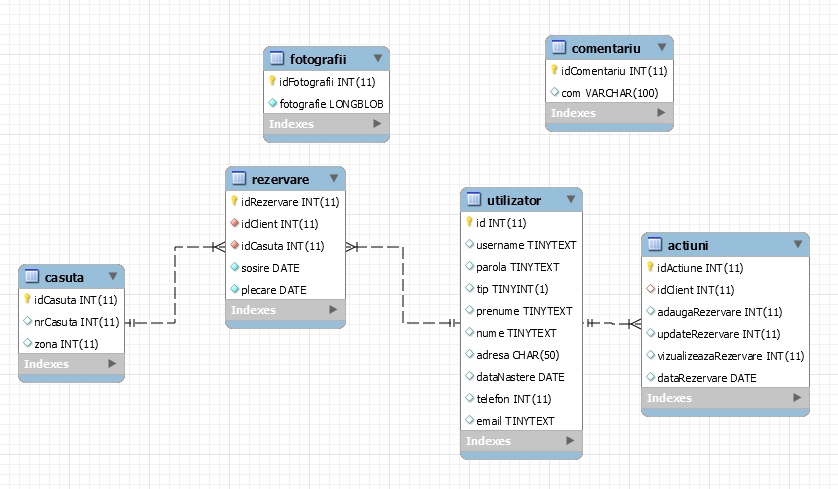
## Dynamic Behavior



## Class Design



# Data Model



# 3.Unit Testing

# 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.]*

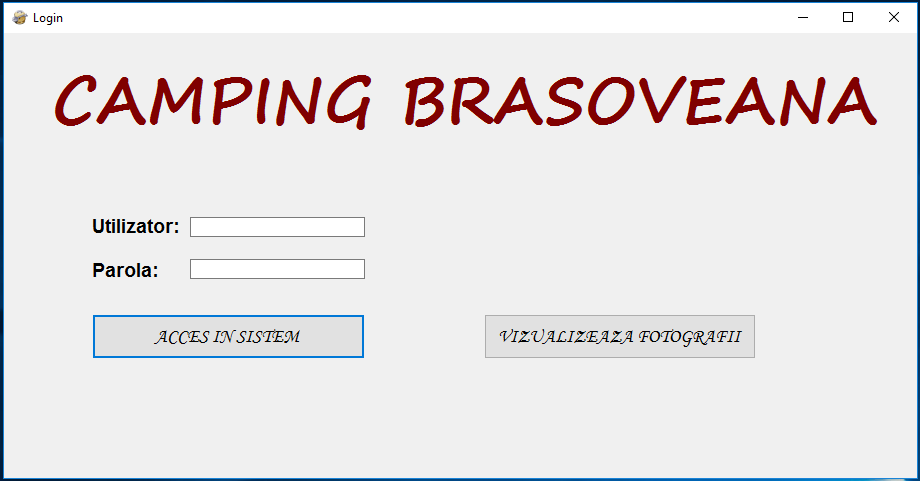
# Design Model Refinement

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

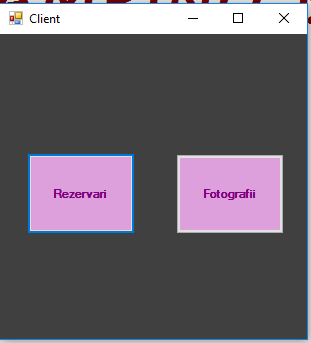
# Construction and Transition

# System Testing

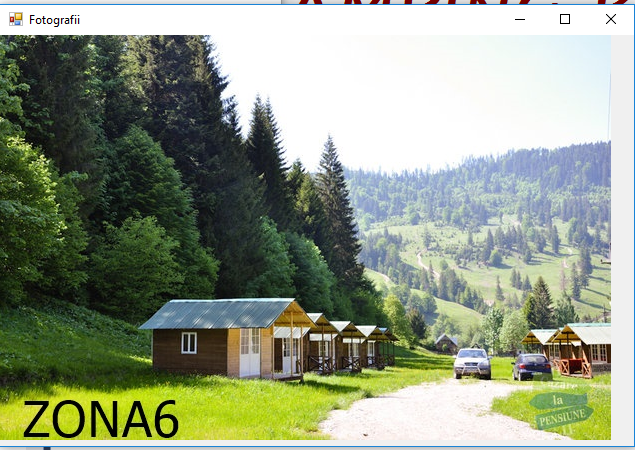
LOGARE:



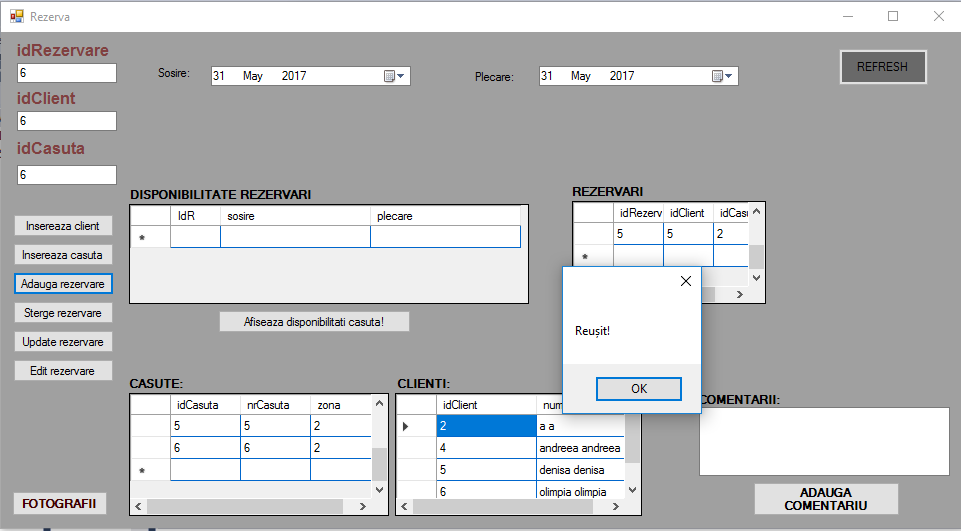
MENIU CLIENT:



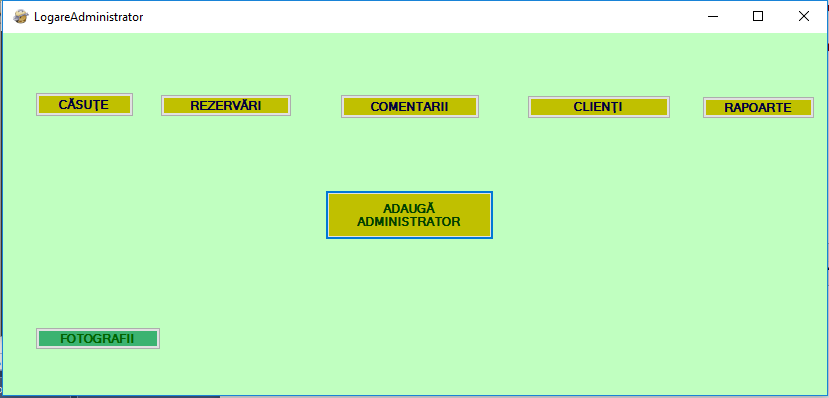
GALERIE:



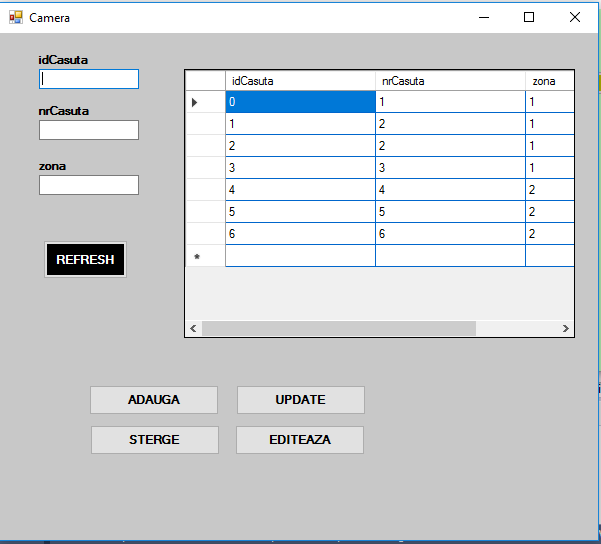
EFECTUARE REZERVARE:



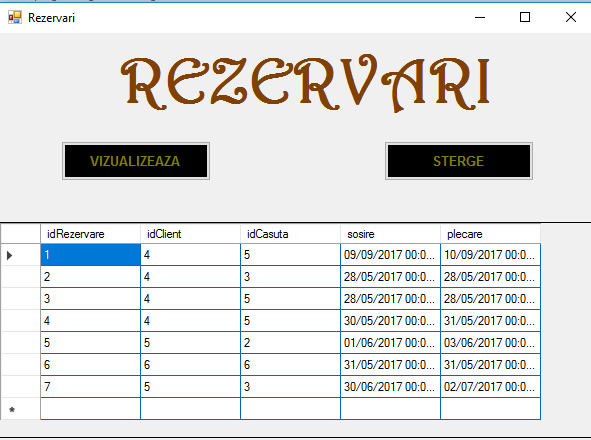
MENIU ADMINISTRATOR:



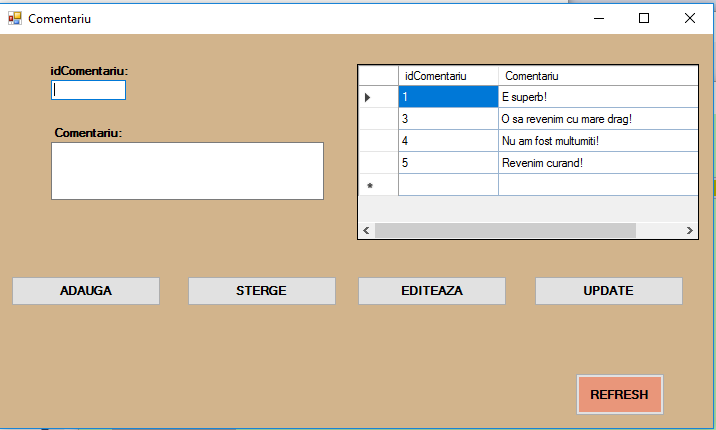
CRUD CASUTA:



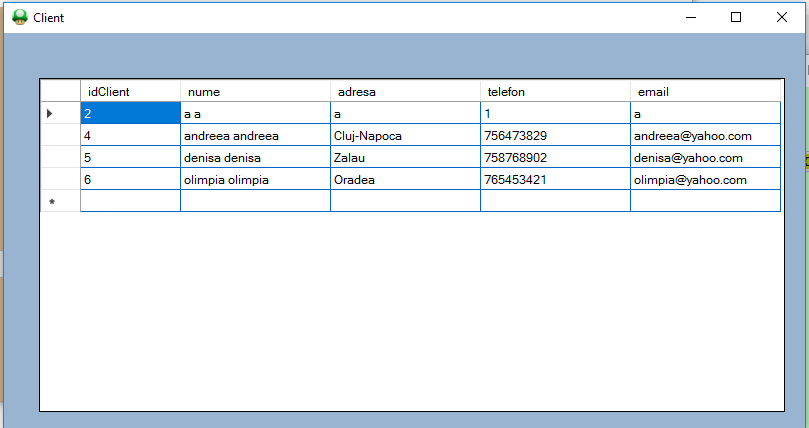
ADMINSTRATOR-REZERVARI



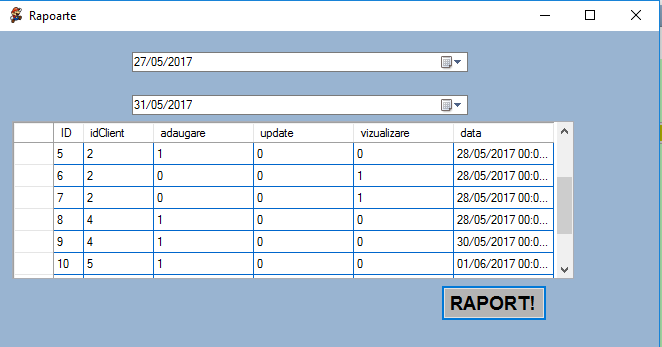
COMENTARII:



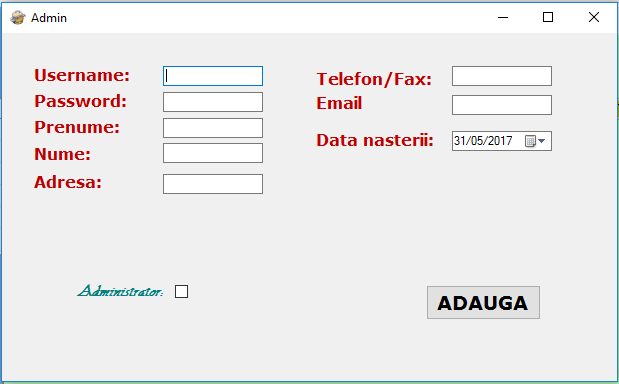
CLIENTI:



RAPOARTE:



ADAUGARE ADMINISTRATOR:



# Future improvements

Proiectul poate avea multiple dezvoltari ulterioare. Spre exemplu:

-implementarea aplicatiei – WEB

-crearea unor meniuri care sa contina preturi

-trimiterea unor e-mail-uri care sa notifice clientii cu o zi inainte de cazare

-reducerea pretului (calculare automata) pentru client fideli

# Bibliography

<http://www.tonymarston.net/php-mysql/3-tier-architecture.html#3-tier>

<https://www.techopedia.com/definition/24649/three-tier-architecture>

<https://www.google.ro/search?q=model+view+controller+package+design&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjFtcnCkY7TAhWMhSwKHfuwBgEQ_AUIBigB&biw=1366&bih=613#imgrc=_>