Sistem informatic pentru gestiunea

inspectoratului de poliție

Student:Cocoș Andreescu Teodora

Student: Drăgan Diana

Student: Cîrhoată Oana Andreea

Anul:IV

Semestrul:I

Cuprins ii

Istoricul reviziilor ii

1. Introducere 1

1.1 Scop 1

1.2 Convenții privind documentul 1

1.3 Sugestii de citire 1

1.4 Scopul produsului 1

1.5 Referințe 1

2. Descriere generală 2

2.1 Perspectiva produsului 2

2.2 Funcțiile produsului 2

2.3 Clasele utilizator și caracteristici 2

2.4 Mediu de funcționare 2

2.5 Constrângeri de proiectare și implementare 2

2.6 Documentația utilizatorului 2

2.7 Ipoteze și dependențe 3

3. Cerințe privind interfața externă 3

3.1 Interfețe utilizator 3

3.2 Interfețe hardware 3

3.3 Interfețe software 3

4. Caracteristicile sistemului 4

4.1Cerințe funcționale 4

5. Alte cerințe non funcționale 4

5.1 Cerințe de siguranță 5

5.2 Cerințe de securitate 5

6. Alte cerințe 5

Anexa A: Glosar 5

Istoricul Reviziilor

|  |  |  |  |
| --- | --- | --- | --- |
| **Nume** | **Data** | **S-a modificat** | **Versiune** |
|  |  |  |  |
|  |  |  |  |

# Introducere

## Scop

Acest document descrie cerințele pentru un sistem informatic ce gestionează activitatea unui magazin online.

## Convenții privind documentul

Font:Arial

Dimensiune:11

Documentul este structurat pe capitole iar fiecare capitol conține unul sau mai multe subcapitole.

Limba utilizată pentru realizarea documentului este limba română.

## Sugestii de citire

Acest document este destinat dezvoltatorilor,managerilor de proiecte,utilizatorilor.

În primul capitol veți găsi scopul și referințele utilizate la realizarea produsului.

În subcapitolul 2.2 sunt enumerate funcțiile realizate de produs și o scurtă descriere a fiecărei funcții.

## Scopul produsului

Sistemul realizat va fi un produs software va asista clienții magazinului în realizarea următoarelor acțiuni specifice:

a)creare cont

b)vizualizare informații produs

c)agaugă în coș

d)șterge produsul adăugat

## Referințe

# Descriere generală

## Perspectiva produsului

Produsul actual este independent și autonom,astfel încât nu depinde de alte produse software exterioare produsului.

## Funcțiile produsului

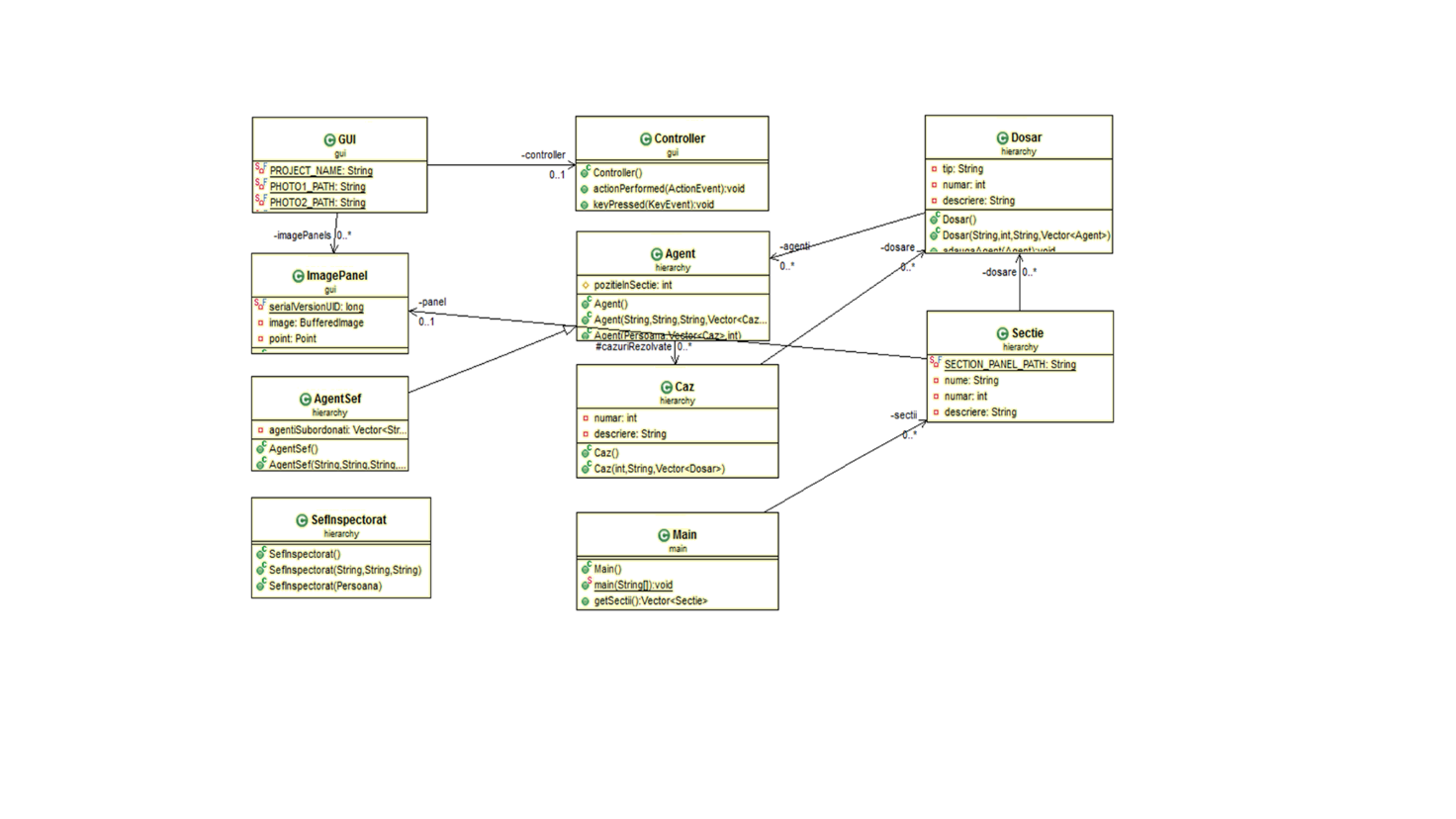
### *Aplicația va avea o interfață grafică simplă și intuitivă*.

## Clasele utilizatori și caracteristici

Utilizatorul tipic al acestui program este considerat a nu avea nici un fel de experiență în dezvoltarea de software și,prin urmare,interfața cu utilizatorul trebuie să fie ușor de folosit.Acest utilizator este de obicei un potențial client al magazinului.

*Implementarea software-ului va respecta conceptul de Mode-View-Controller (MVC)*.

Acest concept este foarte răspândit în programarea web, în MVC, modelul conține datele și regulile business; view conține elemente din interfața utilizator (texte, input-uri ale formularelor etc); controller-ul genstionează comunicația dintre model și view.



## Mediu de funcționare

Proiectul va fi realizat utilizând CodeIgniter și un editor de coduri, de exemplu Sublime. Baza de date va fi locală (localhost).

## Constrângeri de proiectare și implementare

Caracteristicile computer-ului pentru a rula aplicația:

-Procesor: 1GHz sau mai rapid

-Memorie:cel puțin 128 Mb

-Spațiu pe hard disk:minimum 10 Mb

Aplicația va necesita o comunicare pe internet deja stabilită înainte de a rula produsul.

## Documentația utilizatorului

Acest document va servi și utilizatorilor.

## Ipoteze și dependențe

Această subsectie cuprinde acele caracteristici care pot influența proiectarea sistemului.

Aplicația va rula pe orice tip de windows. Pe sistemul de operare trebuie sa fie instalat CodeIgniter, Xampp și eventual Sublime.

# Cerințe privind interfața externă

## Interfețe utilizator

Vor exista 2 interfețe utilizator:

1. una pentru utilizatorul avansat(admin)
2. una pentru utilizatorul de bază care va avea acces limitat (client)

## Interfețe hardware

Nu se aplică.

## Interfețe software

Produse software utilizate pentru funcționare:

1. CodeIgniter
2. Sublime
3. Xampp

Produse software utilizate pentru dezvoltare:

1. CodeIgniter
2. Sublime
3. Xampp

# Caracteristicile sistemului

## Cerință de sistem 1

### *Sistemul va oferi posibilitatea clienților de a se loga prin crearea unui cont.*

### *Aplicația afișează pagina de home, iar după ce au ales un produs clienții trebuie să intre în cont.*

### *Se crează un cont .*

### *Înregistrarea se realizează pe bază de user și parolă.*

## Cerință de sistem 2

### *Pot fi adăugate noi produse cu specificațiile corespunzătoare.*

## Cerință de system 3

### Clientul realizează următoarele operații:

#### Se loghează.

#### Poate căuta informații despre un anumit produs.

#### Poate realiza o comandă.

#### Se deconectează.

## Cerință de sistem 4

### Operațiile care pot fi realizate de administrator:

#### Se conectează.

#### Adaugă produse.

#### Șterge conturi.

#### Se deconectează.

# Alte cerințe non functionale

## Cerințe de siguranță

Nu se aplică.

## Cerințe de securitate

*Finalizarea comenzii se face pe bază de username și o parolă. După o înregistrare reușită, utilizatorul poate plasa comanda.*

# Alte cerințe

Anexa A: Coding style

## Identarea liniei

### *Utilizați un rând de 4 spații,* Se recomandă păstrarea liniilor la o lungime de aproximativ 75-85 de caractere pentru o mai bună citire a codului.

## Structuri de control

### *Acestea includ if, pentru, while, switch, etc. Declarațiile de control ar trebui să aibă un spațiu între cuvântul cheie de control și paranteza de deschidere, pentru a le distinge de apelurile funcționale.*

Exemplu:

if ((condition1) || (condition2)) {

action1;

}elseif ((condition3) && (condition4)) {

action2;

}else {

default action;

}

Exemplu:

switch (condition) {

case 1:

action1;

break;

case 2:

action2;

break;

default:

defaultaction;

break;

}

## Apeluri funcționale

### *Funcțiile ar trebui să fie numite fără spații între numele funcției, paranteza de deschidere și primul parametru; spațiile între virgule și fiecare parametru și nici un spațiu între ultimul parametru, paranteza de închidere și punct și virgulă.*

Exemplu:

$var = foo($bar, $baz, $quux);

## Definiții ale funcțiilor

### *Declarațiile funcțiilor urmează "stilul BSD / Allman".*

function fooFunction($arg1, $arg2 = '') {

if (condition) {

statement;

}

return $val;

}

### *Comentarii-Comentariile despre stilul C (/ \* \* /) și comentariile standard C ++ (//) sunt amândouă. Utilizarea comentariilor stilului Perl / shell (#) este descurajată.*

### *PHP Code Tags-Utilizați întotdeauna <? Php?> Pentru a delimita codul PHP*