Facultatea de Automatică şi Calculatoare, Iași

StDelivery

Membri

Mănăstireanu Dany-Andrei (1306B)

Stratulat Ștefănel-Constantin (1306B)

Zălincă Claudiu-Șerban (1310A)

- Proiect la disciplina Ingineria Programării –

Cuprins

Documentul specificațiilor cerințelor (SRS)

Table of Contents

1 Introduction

1.1 Purpose

StDelivery is an application that facilitates the purchase of food. Given that it is easier to order food online than to go to a restaurant, the application contains a variant menu from which the customer can choose, to make a decent meal. The customer can create an account in the application, so that his contact data is saved, and the delivery of the order is easier, without always filling in the personal contact data. Therefore, the application contains: a menu for authentication and account creation, a page with products that you can add to the shopping cart, a page that contains the shopping cart, with the possibility of removing some products, and a last page for completing the data personal and sending the order. Details about these can be found in the following chapters. This document is made for a preliminary version of the application, in which the customer receives an email with the ordered products.

1.2 Document Conventions

This document follows the IEEE standard formatting for software development. The standard defines a regular formatting this document follows including writing to be done in third-person, passive voice as well as readable and grammatically correct text.

1.3 Intended Audience and Reading Suggestions

This document is not intended for the end user because it provides a detailed specification of how the software is to be implemented. Since a user needs information on how to use the application to order some food, instead of how to implement an application, this document is more geared towards testers and mostly the developers of the mobile applications. The document starts off with an overview of the functions and specifications for this application in section 2, then moves on the describe the requirements for interfacing with external hardware and software in section 3. Section 4 describes the application functions in great detail and section 5 lists various requirements the application must adhere to after completion. It is suggested that all the audiences of this document start with section 2 first, to get a general idea of the software requirements.

Testers should next read sections 5.1 through 5.4 (performance, safety and software quality attributes). Next, a tester should read section 3.1 (user interfaces) followed by all of section 4 (system features).

Reading the document in this order will give the tester an idea of what to expect in the interface at first glance, and then they may test all the individual functions to make sure they adhere to the specifications. After reading section 2, mobile applications developers should read the remaining sections in order because this document was designed specifically for the purpose of developing the app. The develop needs to get an overall idea in section 2 of the app. Section 4 is the most important because it describes all the functions of the app in great detail and it will help with making decisions in writing actual code for the application.

1.4 Product Scope

StDelivery is a food delivery app developed on a C# platform. The objective of the application is to go through the list of products and build a shopping cart that the user can order. This shopping cart can be modified later, in the final verification stage. The customer can request the sending of the order by manually or automatically filling in the contact details. The autocomplete is done based on the details of the user account. Also, based on the data in the account, he will receive an email with the order made.

The application can run on any mobile device running an Android version at least 8.0.

2 Overall Description

2.1 Product Perspective

The StDelivery application is a standalone application, developed from scratch. The idea of developing this application came by making an analysis of the world we live in. We can see that people are in a hurry and do not have time to cook. Also, people have become more individualistic and prefer to stay home to watch a show and eat, instead of going to a restaurant with friends. Therefore, the application meets the needs of these people by ordering food online. The application ensures that it contains the most consumed products in the world today (fast food, pizza, pasta, soup, salad and desserts). The application is user-friendly, both by the fact that it is easy to use and by the fact that it ensures the protection of the client's contact data. The account password is not saved in plain text, but a key is saved, resulting from the application of an SHA encryption algorithm.

StDelivery is implemented as a C # application on the Xamarin platform. The graphical interfaces are made using XML and the Activity classes related to the manipulation of each graphical window, will contain an OnCreate function that will be responsible for the manipulation of all current elements, by attaching the callback functions to the interaction elements. These callback functions contain the logic of the current window, and decisions to move to another window, by instantiating another Activity class.

Also, for the user's authentication and contact data, SQL databases are used, manipulated through the SQLite library, and for sending the email related to the command, the SMTP protocol is used, manipulated through the Mail library in System.Net. The latter will be implemented in DLLs.

2.2 Product Functions

The main functions that StDelivery must perform are the following:

- Create login account: the ability to quickly create a user account and quick login; contact data is stored in an SQL database and password encryption is ensured.
- Building the shopping cart: the customer has at his disposal a wide variety of products, which he can easily add to the shopping cart, which he can modify later.
- Order completion: contact details are filled in automatically or manually, depending on the customer's request and after requesting the order, an email with the order details is sent.

2.3 User Classes and Characteristics

StDlivery must be designed for ease of use so that any class of user who contains an Android device can use it without encountering difficulties. The graphical interface is very intuitive. By using intuitive graphic elements, it is very clear how the user creates an account, logs in, builds the shopping cart and completes the data so that he can order food.

The only class of users that would be more difficult would be the elderly, who do not have an email to receive the order or people who do not have an Android mobile device.

2.4 Design and Implementation Constraints

Because the application is developed as a project within the Programming Engineering course of the Technical Faculty of Automation and Computers, the only impediment encountered is the development time, having only 13 weeks, but some of the principles we had to use were presented later, during those weeks.

Therefore, the application will be summarized only at the prototype level, which offers the possibility to build a shopping cart from a rich list of products. Sending the order is limited to sending an email with the requested products. The application does not yet aim to produce food and deliver it. This functionality will be developed later, as our professional development.

2.5 User Documentation

The documentation related to the application, class diagrams, activities and sequences will be attached for a better understanding of the application. Also, the application will be associated with a help that will describe the functionality of each class

2.6 Assumptions and Dependencies

The application will depend on the Xamarin.Android platform and the main functionalities it uses. The available graphic elements will be taken over and manipulated accordingly. Their handling is done using the appropriate packages, taken from NuGet. We also use some libraries such as SQLite to manipulate the SQL database and Net.Mail to manipulate the SMTP protocol in order to send emails.

3 External Interface Requirements

3.1 User Interfaces

The graphical interface will contain elements provided by the Xamarin platform for C #. Therefore, the interface is very simple and intuitive, containing simple elements. The user is greeted by an authentication page. If he does not have an account, he has the possibility to create one. In this case, the user fills in a series of entries regarding personal data and then, by pressing the end button, he can authenticate, see the list of products.

The product list contains a series of defined categories of images of product types, arranged in the form of a matrix. By selecting a category, a list of products is displayed (name, ingredients, price and button to add to cart). When added to the cart, the user can change his mind by selecting the revoke option. After selecting all the products, the final list of products is displayed, which the user can modify by deleting some products. The products are displayed, a label with the final price and the order completion button, which redirects to the last page, which contains the order submission form. There is a combo-box, which offers the possibility of manual or automatic data completion. All data and checkboxes must be completed in order to complete the order.

At the same time, the selection of the main actions generates guidance messages for the user (incomplete data or the order sent successfully). There is also the possibility to navigate back between pages. The interface is simple and intuitive, with an emphasis on functionality.

3.2 Hardware Interfaces

The main hardware device required is a mobile device running Android at least 8.0 (mobile phone or tablets). It must have a display with a functional touch screen that coherently runs all the basic functionalities of Android.

3.3 Software Interfaces

The main software feature required to run the application is the Android operating system at least 8.0, which offers the functionalities related to pressing buttons, navigating between windows, scrolling in the list and writing text. The operating system must be able to communicate with an SQL database (for users) and manipulate Internet protocols such as SMTP. The latter will be encapsulated in DLLs, which the application must be able to use properly. In fact, the data is manipulated internally by the application, either by the parameters of the functions or by global variables (current user and shopping cart) and design templates are used for a better design of the communication between the components.

3.4 Communications Interfaces

For communication with the database in order to manipulate user accounts, the SQLite library is used, which offers all the CRUD facilities. For user security, a SHA password encryption algorithm is used. At the same time, the SMTP internet protocol is used to send emails, manipulated with the Net.Mail library. With this library the connection is made by authentication to the SMTP server and an email is built and sent. To use this protocol, use the Gmail server, with a specially created email, for which the security elements have been deactivated.

4 System Features

Our system features are covered in-depth in separate appendices for scenarios and diagrams of classes, activities, sequences. This is also a very usefull help file, containing the entire design of the project.

5 Other Nonfunctional Requirements

5.1 Performance Requirements

The application must run on any mobile system with Android at least 8.0. No memory or processor requirements are required. The application is simple and does not require many resources.

5.2 Safety Requirements

No requirements are imposed to ensure user safety. The application cannot affect the end user in any way. The only concern is that he does not order too much food.

5.3 Security Requirements

The application ensures the confidentiality and protection of customer data. These data will not be shared but will be used only for the purpose of the application. At the same time, the requested data are only the contact ones, which does not imply in any way the violation of the client's privacy. At the same time, the encryption of the client's user password is ensured.

5.4 Software Quality Attributes

The application must be robust, intuitive and error-free to ensure a pleasant user experience. Also, design templates and DLLs are used, which makes the application easy to develop and modify. Classes are used that contain public and private methods that can be easily understood and tested

5.5 Business Rules

It is the policy of the development team to follow all the rules established by the University.

6 Other Requirements

Appendix A: Glossary

N/A

Appendix B: Analysis Models

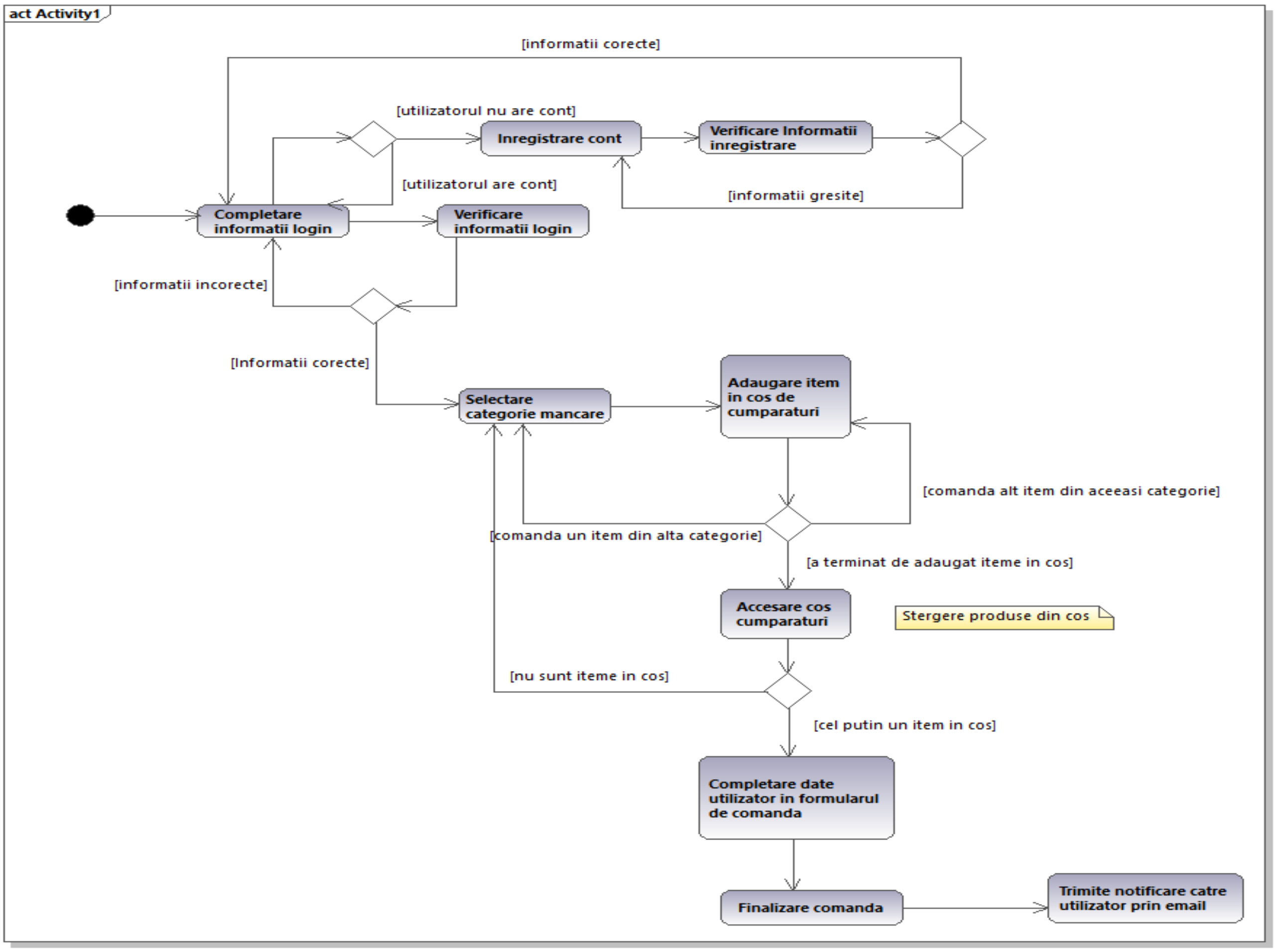
Related to the project, UML diagrams of classes, activities and sequences are attached for a better understanding of the design.

Appendix C: To Be Determined List

Further details can be found in the help file attached to the project, which contains the project structure and how to use it.

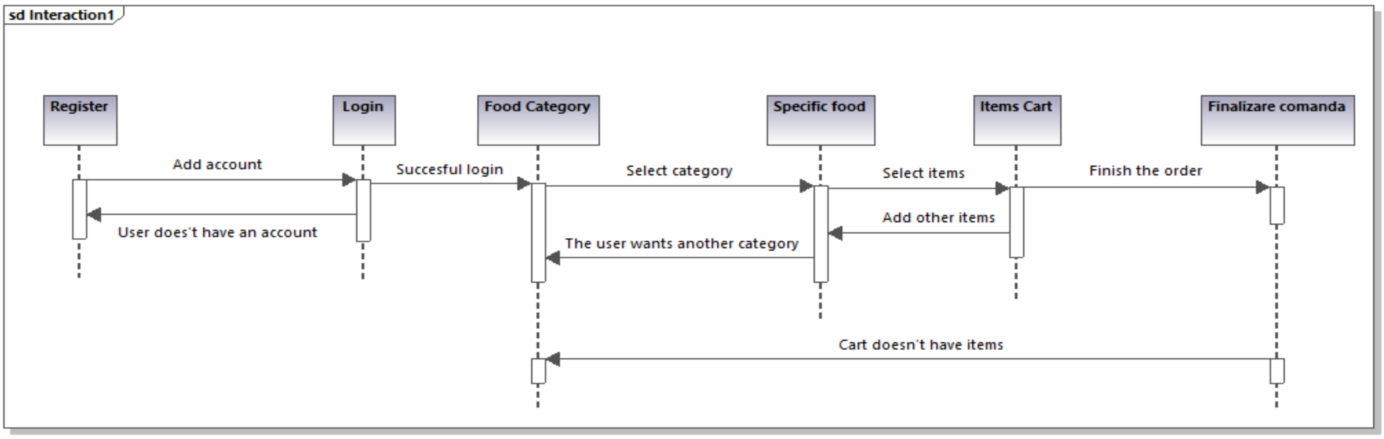
Diagrame UML

- **Diagrama de activități**


act Activity1
[informatii corecte]
[utilizatorul nu are cont]
Inregistrare cont
Verificare Informatii inregistrare
[utilizatorul are cont]
[informatii gresite]
Completare informatii login
Verificare informatii login
[informatii incorecte]
[Informatii corecte]
Selectare categorie mancare
Adaugare item in cos de cumparaturi
[comanda alt item din aceeasi categorie]
[comanda un item din alta categorie]
[a terminat de adaugat iteme in cos]
Accesare cos cumparaturi
Stergere produse din cos
[nu sunt iteme in cos]
[cel putin un item in cos]
Completare date utilizator in formularul de comanda
Finalizare comanda
Trimite notificare catre utilizator prin email

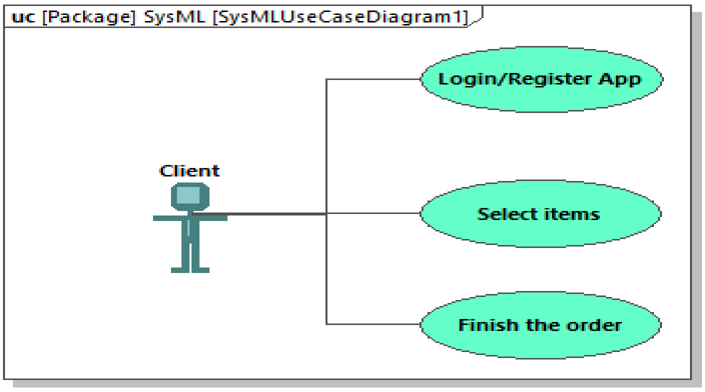
Generated by UModel www.altova.com

- **Diagrama de secvențe**



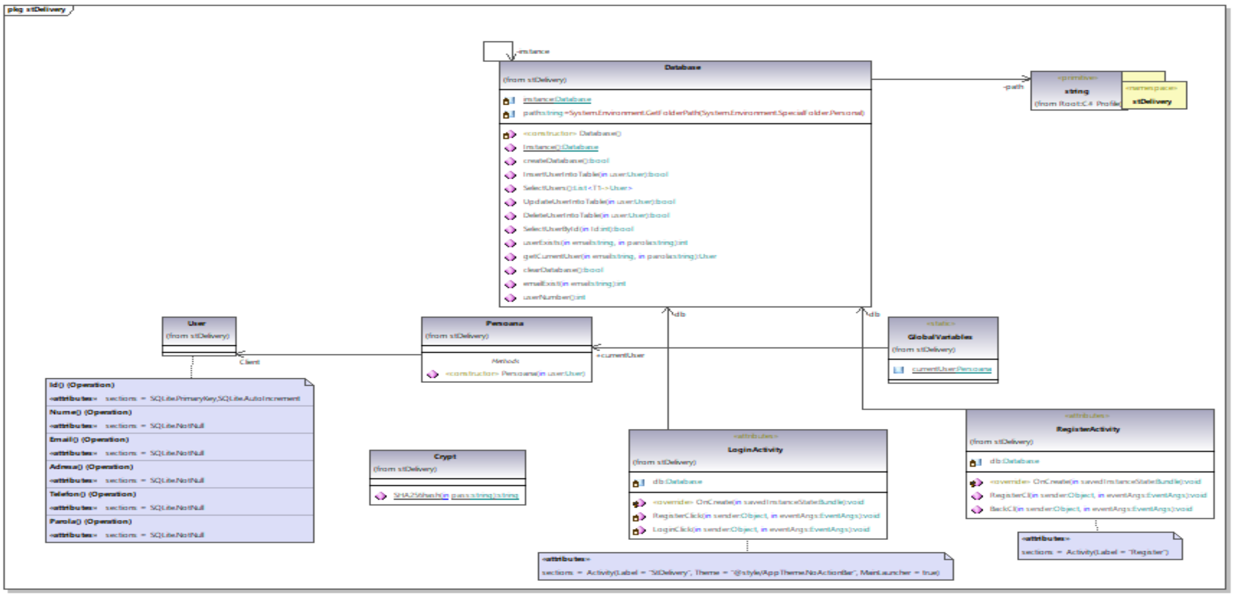
Generated by UModel www.altova.com

- **Diagrama Use-Case**

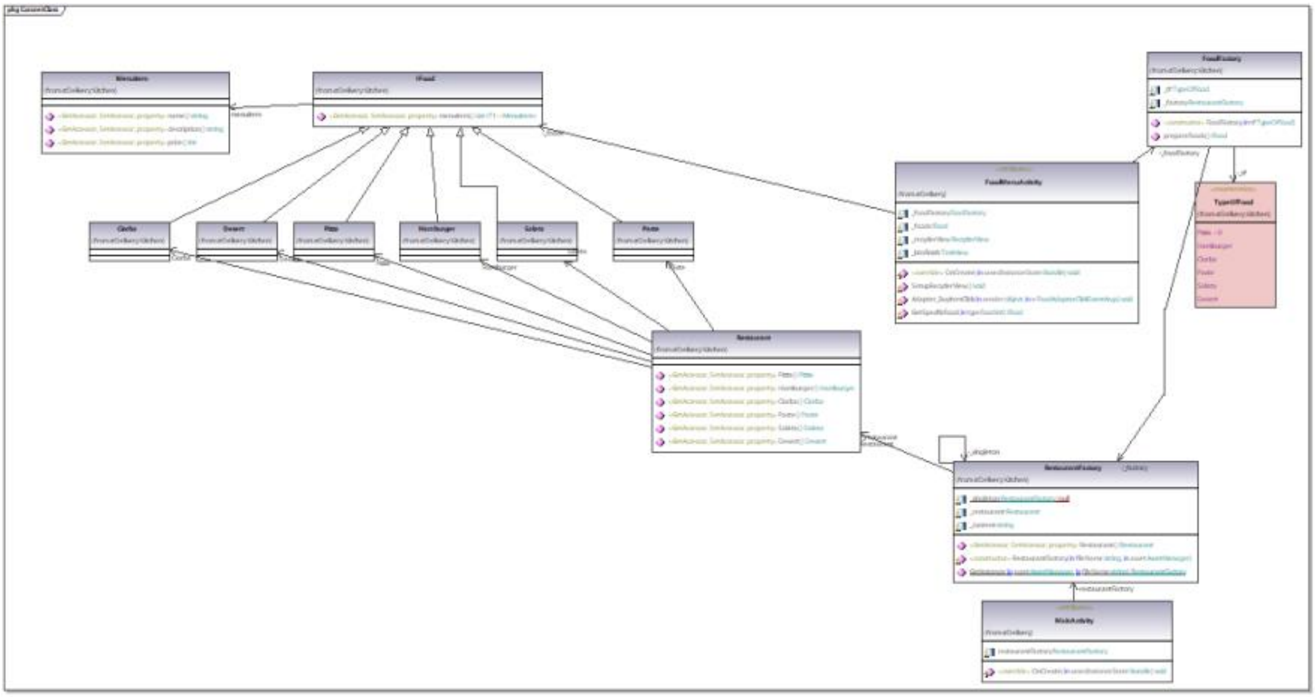


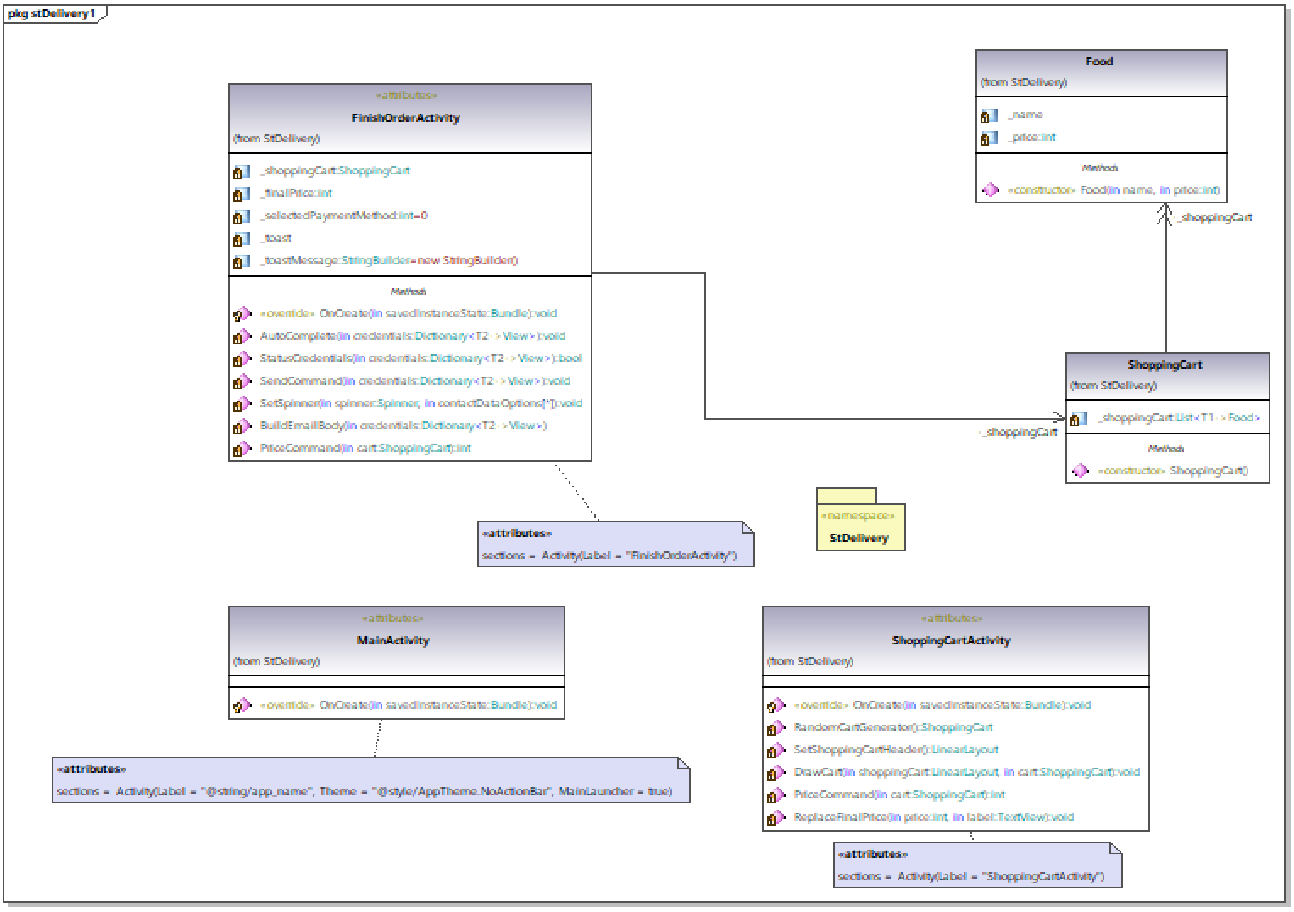
Generated by UModel www.altova.com

- **Diagrama de clase**


pkg stDelivery
-instance
Database
(from stDelivery)
instance:Database
path:string =System.Environment.GetFolderPath(System.Environment.SpecialFolder.Personal)
«constructor» Database()
Instance():Database
createDatabase():bool
InsertUserIntoTable(in user:User):bool
SelectUsers():List<T1->User>
UpdateUserIntoTable(in user:User):bool
DeleteUserIntoTable(in user:User):bool
SelectUserById(in Id:int):bool
userExists(in email:string, in parola:string):int
getCurrentUser(in email:string, in parola:string):User
clearDatabase():bool
emailExist(in email:string):int
userNumber():int
-path
«primitive» string (from Root:C# Profile)
«namespace» stDelivery
User
(from stDelivery)
Client
Id() (Operation)
«attributes» sections = SQLite.PrimaryKey,SQLite.AutoIncrement
Nume() (Operation)
«attributes» sections = SQLite.NotNull
Email() (Operation)
«attributes» sections = SQLite.NotNull
Adresa() (Operation)
«attributes» sections = SQLite.NotNull
Telefon() (Operation)
«attributes» sections = SQLite.NotNull
Parola() (Operation)
«attributes» sections = SQLite.NotNull
Persoana
(from stDelivery)
Methods
«constructor» Persoana(in user:User)
+currentUser
-db
-db
«static» GlobalVariables
(from stDelivery)
currentUser:Persoana
Crypt
(from stDelivery)
SHA256hash(in pass:string):string
«attributes» LoginActivity
(from stDelivery)
db:Database
«override» OnCreate(in savedInstanceState:Bundle):void
RegisterClick(in sender:Object, in eventArgs:EventArgs):void
LoginClick(in sender:Object, in eventArgs:EventArgs):void
«attributes» sections = Activity(Label = "StDelivery", Theme = "@style/AppTheme.NoActionBar", MainLauncher = true)
«attributes» RegisterActivity
(from stDelivery)
db:Database
«override» OnCreate(in savedInstanceState:Bundle):void
RegisterCl(in sender:Object, in eventArgs:EventArgs):void
BackCl(in sender:Object, in eventArgs:EventArgs):void
«attributes» sections = Activity(Label = "Register")

Generated by UModel www.altova.com



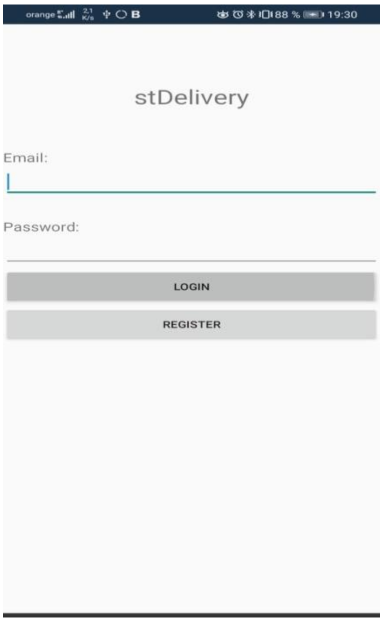

pkg stDelivery1
«attributes»
FinishOrderActivity
(from StDelivery)
_shoppingCart:ShoppingCart
_finalPrice:int
_selectedPaymentMethod:int=0
_toast
_toastMessage:StringBuilder=new StringBuilder()
Methods
«override» OnCreate(in savedInstanceState:Bundle):void
AutoComplete(in credentials:Dictionary<T2 -> View>):void
StatusCredentials(in credentials:Dictionary<T2 -> View>):bool
SendCommand(in credentials:Dictionary<T2 -> View>):void
SetSpinner(in spinner:Spinner, in contactDataOptions[*]):void
BuildEmailBody(in credentials:Dictionary<T2 -> View>)
PriceCommand(in cart:ShoppingCart):int
Food
(from StDelivery)
_name
_price:int
Methods
«constructor» Food(in name, in price:int)
_shoppingCart
ShoppingCart
(from StDelivery)
_shoppingCart:List<T1 -> Food>
Methods
«constructor» ShoppingCart()
-_shoppingCart
«attributes»
sections = Activity(Label = "FinishOrderActivity")
«namespace»
StDelivery
«attributes»
MainActivity
(from StDelivery)
«override» OnCreate(in savedInstanceState:Bundle):void
«attributes»
sections = Activity(Label = "@string/app_name", Theme = "@style/AppTheme.NoActionBar", MainLauncher = true)
«attributes»
ShoppingCartActivity
(from StDelivery)
«override» OnCreate(in savedInstanceState:Bundle):void
RandomCartGenerator():ShoppingCart
SetShoppingCartHeader():LinearLayout
DrawCart(in shoppingCart:LinearLayout, in cart:ShoppingCart):void
PriceCommand(in cart:ShoppingCart):int
ReplaceFinalPrice(in price:int, in label:TextView):void
«attributes»
sections = Activity(Label = "ShoppingCartActivity")

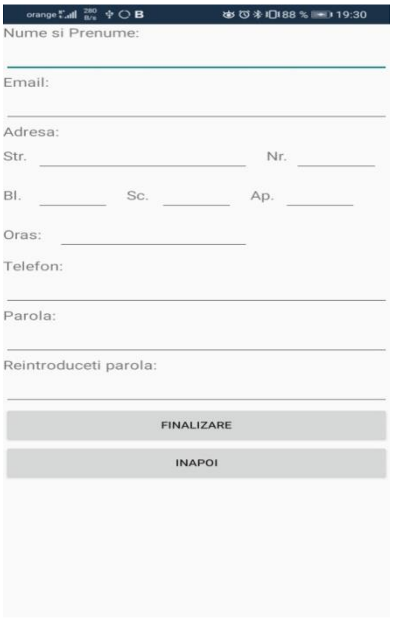
Generated by UModel www.altova.com

Modul de utilizare a programului

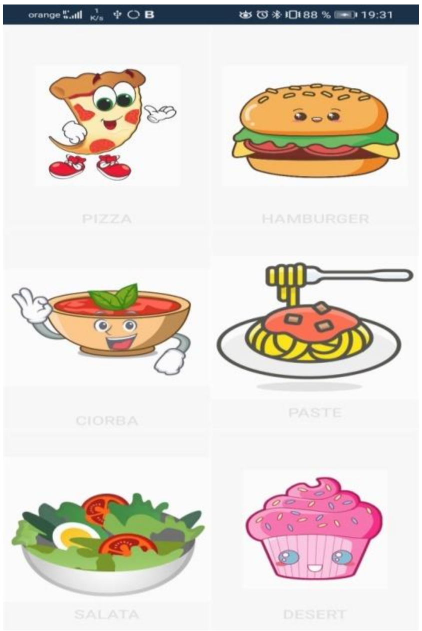
Aplicația StDelivery are o interfață grafică foarte intuitivă și simplă, astfel încât utilizatorul să înțeleagă ce acțiuni trebuie sa facă astfel încât să obțină rezultatul dorit.

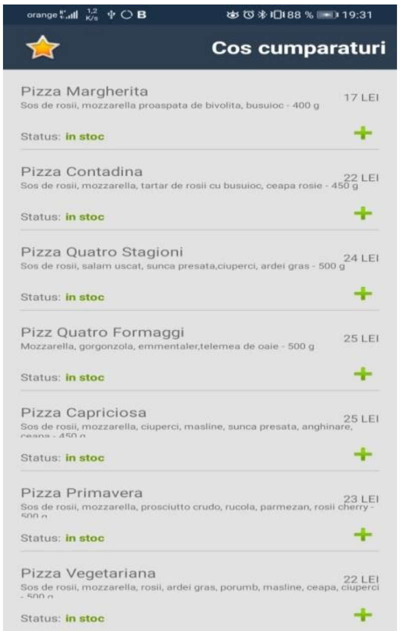
Aplicația se deschide cu o pagină de autentificare. Utilizatorul trebuie să introducă datele de autentificare pentru a putea accesa aplicația (lista de cumpărături). Acesta trebuie să introducă email-ul și parola pentru a se autentifica. În cazul în care acesta nu deține un cont de utilizator, acesta are opțiunea de a crea un cont, selectând opțiunea Register. După ce acesta introduce toate informațiile cerute și selectează opțiunea de finalizare, are un cont creat și se poate autentifica în aplicație.





După ce logarea se realizează cu succes, utilizatorul are acces la lista cu categoriile de produse. Acesta poate selecta oricare dintre aceste categorii, urmând să se afișeze o listă cu toate produsele aferente categoriei selectate. Parcugând lista de produse, utilizatorul poate alege un produs și prin apăsare, apare un meniu de unde acesta selectează dacă dorește introducerea produsul în coș, sau s-a răzgândit.





Pizza Margherita 17 LEI
Sos de rosii, mozzarella proaspata de bivolita, busuioc - 400 g
Status: in stoc +

Pizza Contadina 22 LEI
Sos de rosii, mozzarella, tartar de rosii cu busuioc, ceapa rosie - 450 g
Status: in stoc +

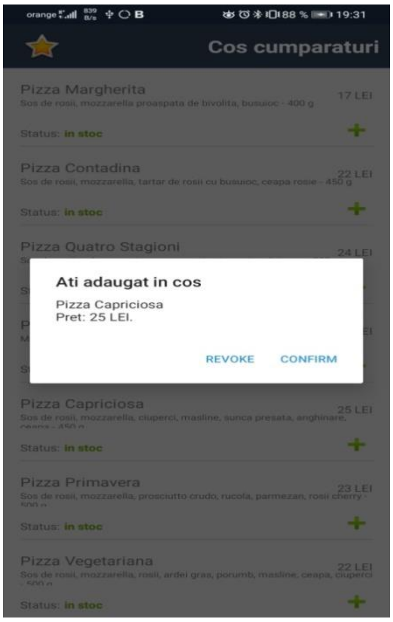
Pizza Quatro Stagioni 24 LEI
Sos de rosii, salam uscat, sunca presata,ciuperci, ardei gras - 500 g
Status: in stoc +

Pizz Quatro Formaggi 25 LEI
Mozzarella, gorgonzola, emmentaler,telemea de oaie - 500 g
Status: in stoc +

Pizza Capriciosa 25 LEI
Sos de rosii, mozzarella, ciuperci, masline, sunca presata, anghinare, ceapa - 450 g
Status: in stoc +

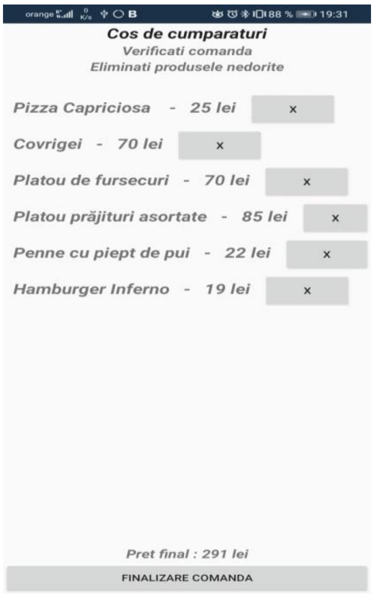
Pizza Primavera 23 LEI
Sos de rosii, mozzarella, prosciutto crudo, rucola, parmezan, rosii cherry - 500 g
Status: in stoc +

Pizza Vegetariana 22 LEI
Sos de rosii, mozzarella, rosii, ardei gras, porumb, masline, ceapa, ciuperci - 500 g
Status: in stoc +



Ati adaugat in cos
Pizza Capriciosa
Pret: 25 LEI.

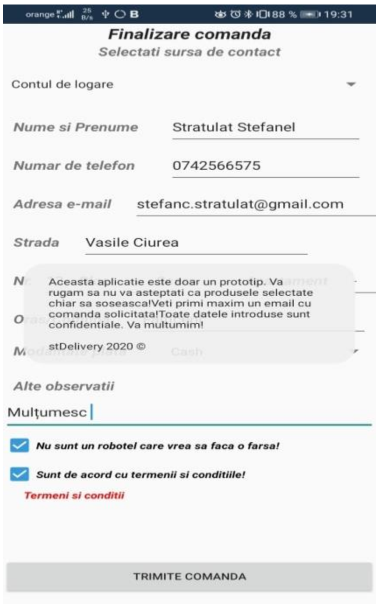
După ce utilizatorul a parcurs categoriile de interes și a selectat produsele dorite, utilizatorul poate selecta opțiunea de vizualizare a coșului de cumpărături, unde găsește toate produsele pe care le-a selectat. Aici, acesta poate șterge din coșul de cumpărături, produsele pe care nu le mai dorește. Acesta poate viziona atât produsele selectate, cât și prețul total al comenzii. Utilizatorul poate selecta opțiunea de finalizare a comenzii, fiind redirecționat spre pagina finală. Pe această pagină, utilizatorul trebuie să introducă datele de contact pentru finalizarea comenzii. Această acțiune se poate face manul, utilizatorul introducând manual toate datele, sau automat, pe baza datelor de autentificare. Selectarea acestei opțiuni se face pe baza unui combobox. După ce utilizatorul introduce toate datele, trebuie să bifeze cele două opțiuni care confirmă faptul că nu este robot și că este de acord cu termenii și condițiile, pe care le poate citi, apăsând textul roșu de pe ecran.



Cos de cumparaturi
Verificati comanda
Eliminati produsele nedorite

Pizza Capriciosa - 25 lei ×
Covrigei - 70 lei ×
Platou de fursecuri - 70 lei ×
Platou prăjituri asortate - 85 lei ×
Penne cu piept de pui - 22 lei ×
Hamburger Inferno - 19 lei ×

Pret final : 291 lei



Finalizare comanda
Selectati sursa de contact

Contul de logare

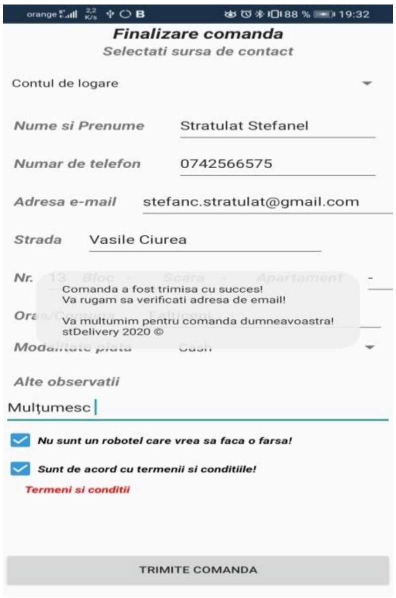
Nume si Prenume: Stratulat Stefanel
Numar de telefon: 0742566575
Adresa e-mail: stefanc.stratulat@gmail.com
Strada: Vasile Ciurea

Aceasta aplicatie este doar un prototip. Va rugam sa nu va asteptati ca produsele selectate chiar sa soseasca!Veti primi maxim un email cu comanda solicitata!Toate datele introduse sunt confidentiale. Va multumim!

stDelivery 2020 ©

Alte observatii: Mulțumesc

☑ Nu sunt un robotel care vrea sa faca o farsa!
☑ Sunt de acord cu termenii si conditiile!
Termeni si conditii



Finalizare comanda
Selectati sursa de contact

Contul de logare

Nume si Prenume: Stratulat Stefanel
Numar de telefon: 0742566575
Adresa e-mail: stefanc.stratulat@gmail.com
Strada: Vasile Ciurea
Nr. 13

Comanda a fost trimisa cu succes!
Va rugam sa verificati adresa de email!

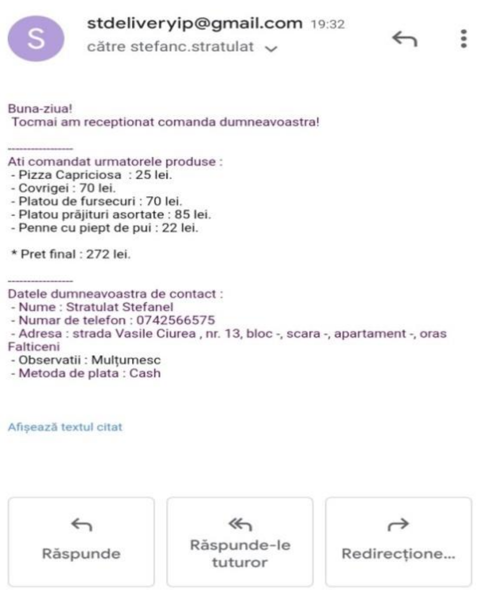
Va multumim pentru comanda dumneavoastra!
stDelivery 2020 ©

Alte observatii: Mulțumesc

☑ Nu sunt un robotel care vrea sa faca o farsa!
☑ Sunt de acord cu termenii si conditiile!
Termeni si conditii

Toate acțiunile de pe această pagină sunt însoțite de informații care apar sub forma de message-box.

Toate datele trebuie să fie introduse și check-box-urile selectate pentru ca utilizatorul să poată trimite comanda. Dacă toate datele sunt introduse și sunt valide, utilizatorul selectează opțiunea de trimitere a comenzii prin apăsarea butonului „Trimite Comanda”. La apăsarea acestui buton, utilizatorul va primi un mesaj de confirmare și un email cu produsele din coșul de cumpărături și datele de contact.



stdeliveryip@gmail.com 19:32
către stefanc.stratulat

Buna-ziua!
Tocmai am receptionat comanda dumneavoastra!

Ati comandat urmatorele produse :
- Pizza Capriciosa : 25 lei.
- Covrigei : 70 lei.
- Platou de fursecuri : 70 lei.
- Platou prăjituri asortate : 85 lei.
- Penne cu piept de pui : 22 lei.

* Pret final : 272 lei.

Datele dumneavoastra de contact :
- Nume : Stratulat Stefanel
- Numar de telefon : 0742566575
- Adresa : strada Vasile Ciurea , nr. 13, bloc -, scara -, apartament -, oras Falticeni
- Observatii : Mulțumesc
- Metoda de plata : Cash

Afișează textul citat