THE PROJECT TO CREATE A CLIENT-SERVER APPLICATION «LOBSTER»

Polyvanyi Volodymyr, Huseinov Rauf, PhD Teslenko Pavlo
Odessa national polytechnic university
Ukraine, Odesa
vladimirsbox2002@gmail.com

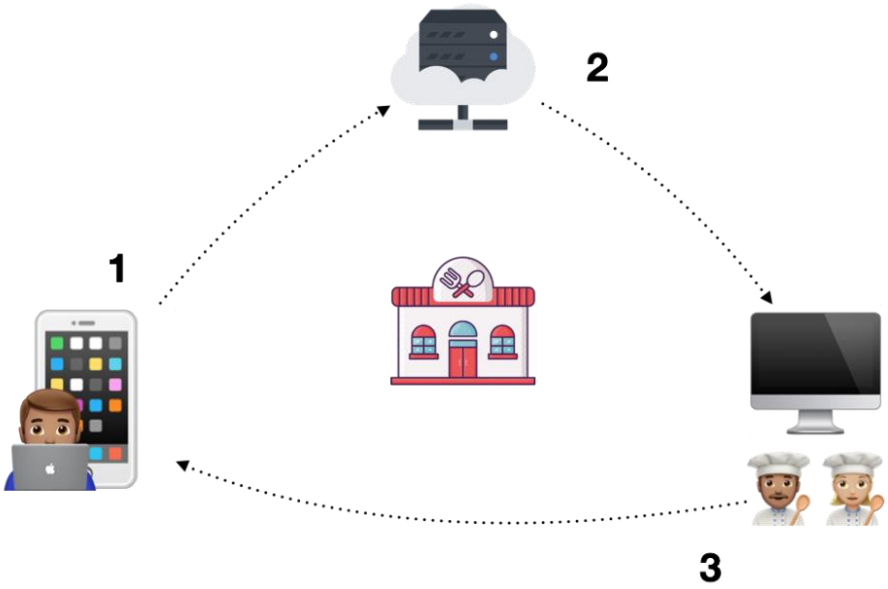
Presented the result of the projection of the information system that to relevant during a visit to the restaurant, cafe. Justified the use of IOS. Developed UML diagrams for understanding of business process of making order in cafes and as a result, a client-server application was created in the language Swift. It takes into account existing recommendations and developments and is focused on their technical implementation

Keywords: program project, IOS, Swift, café, platform

Introduction. We need to understand influence of Apple technologies and products. Because of this we developed informational system that to relevant during a visit to the restaurant. Swift was chosen like programming language because it is simpler and more comfortable of his analogs on IOS.

Objective. The purpose of the work is the development of a software complex for ordering dishes in restaurants / cafes. Such a complex should solve the main problems of running a restaurant business with tourists in mind.

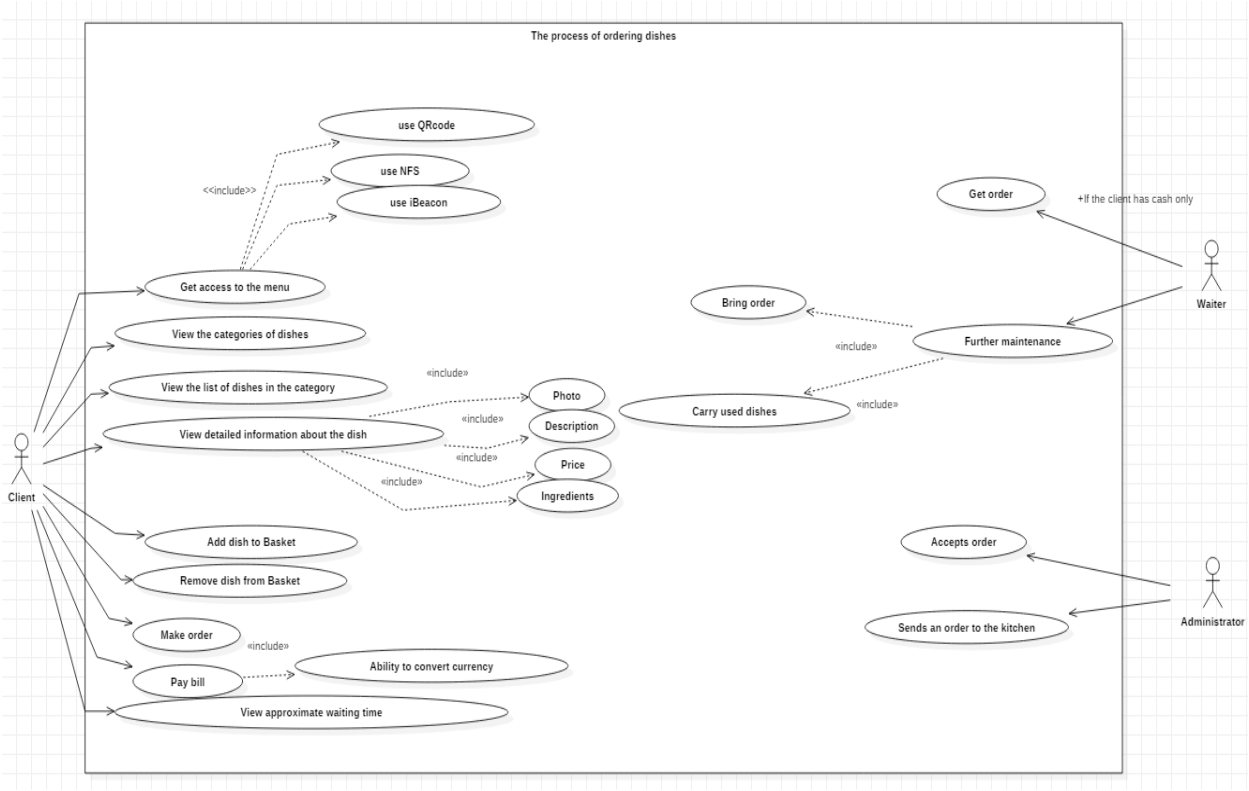
The purpose of our project is the development of software that would simplify the interaction of restaurants with tourists. User will have opportunity to make order beforehand, choose place in café or restaurant and choose even favorite dishes. The scheme of the program is shown in Picture 1.



Picture 1 - scheme of work of the program complex

For the first we need to understand what functional will be in our information system. The software package should consist of 4 systems: a client program installed on an iOS device; the server program responsible for the client's access to the restaurant menu; system that

determines the location of the client inside the restaurant / cafe; a client program installed on a computer in the kitchen to receive orders or other information. The simplest is a representation of a user's interaction with the system is use-case diagram that shows the relationship between the user and the different use cases in which the user is involved. The use-case diagram of the program is shown in Picture 2.



Picture 2– Use-case diagram

In the application there is access to a call of the waiter in unforeseen circumstances. When the waiter is called, the client application creates and sends a request to the server with the call of the waiter. The server processes this information, and passes the request to the client program installed on the computer in the kitchen.

The application for iOS is implemented in the programming language - Swift. Access to the menu will be realized at the restaurant's choice: using a QR code, an NFC sensor, or a Beacon sensor developed on a Bluetooth BLE system. Choosing the first option - QR-code, on each table will print a piece of paper which will be represented QR-code. It will store a hash in it, which will be recognized by the client application for iOS. The second option for accessing the menu is the NFC system. On the table will be installed special sensors NFC to obtain a hash. The third option, the Beacon system - the transfer of data between wireless devices - beacons - and devices that support Bluetooth LE.

To create the server, the Google App Engine cloud platform was studied and analyzed. The platform from Google has a number of advantages, the most important of which are the use of the PaaS model, the possibility of absolutely transparent expansion in case of increased

load, and easy integration with mobile devices. To write the API server used Google Endpoints - a set of tools for Internet services to which you can easily connect from mobile devices. When writing the application, the MVC programming pattern was applied, which effectively separated the application graphics from the business model. For stable and fast work, the concept of multithreading is implemented. The list of names, descriptions, prices of dishes, as well as descriptions of ingredients stored on the server in the form of a database, which makes it easy to update and adjust.

Our project aims to improve the service of cafes and restaurants. We want to provide exclusive content and exclusive possibilities. Information system will automate process of making order, book places and so on. We have chose IOS like the main platform because this platform need in applications like this and we have the biggest experience with this platform and programming languages which connecting with it. In future we plan expand the application by Android version and updates by both platforms. Also in planes use augmented reality for dishes with ARKit technologies.

REFERENCES

1. ISO/IEC 17799:2016. URL: https://docs.swift.org/swift-book/GuidedTour/GuidedTour.html
2. ISO/IEC 17799:2018. URL: https://www.oreilly.com/library/view/uml-20-in/0596007957/ch07.html
3. ISO/IES 15408-1:2014. URL:http://agilemodeling.com/style/useCase Diagram.htm
4. Carlos M. Icaza, The Swift Programming Language, 2016
5. Finn R. Program uncovers hidden connections in the literature. The Scientist. 1998;12

РАЗРАБОТКА МЕТОДИКИ СОЗДАНИЯ AR-ПУТЕВОДИТЕЛЕЙ

[1]Приходько Ирина, [2]Черниенко Екатерина, [3]Трояновская Юлия, [4]Droste Michael
[1,2,3]Одесский национальный политехнический университет;
[4]Берлинский университет техники и экономики HTW Berlin
[1,2,3]Украина, Одесса; [4]Федеративная Республика Германия, Берлин
ir.prikhodko@gmail.com

На примере AR-проекта “Look at the past” рассмотрены возможности существующих технологических решений по созданию мобильных приложений с применением технологии дополненной реальности. Обоснована актуальность повышения эффективности AR-приложений за счет разработки модуля распознавания зданий инвариантного к масштабу, сдвигу, повороту, освещенности и других помех, вызванных возможностями получения исходного изображения

Ключевые слова: дополненная реальность, распознавание изображений, Vuforia, Unity3D, AR-путеводитель