

## Designing a Courier Optimized Route Mobile Application

Luiza Cecilia MAZARE, Simona Vasilica OPREA  
Department of Economic Informatics and Cybernetics  
The Bucharest University of Economic Studies, Romania  
[luiza.mazare@gmail.com](mailto:luiza.mazare@gmail.com), [simona.oprea@csie.ase.ro](mailto:simona.oprea@csie.ase.ro)

*Nowadays, most activities have had encountered changes due to the technology progress process. There is an obvious preference for the use of applications that help people with the everyday tasks, such as: getting a cab, booking a ticket at the cinema or even waiting for the online shopping to arrive. In this paper, we propose a mobile application for courier activities with optimization route that improves the performance.*

**Keyword:** *courier optimization route, mobile application design*

### 1 Introduction

Online shopping was a new yet weird concept in the beginning, but after some time it became the first choice for most of the people. The idea of online shopping was applied for all sorts of products, such as: furniture, clothes, books, cosmetics, flowers and much more items that we use every day. The work schedule most people have nowadays doesn't leave so much time left for going to stores so online shopping is a very good alternative. This process can be done through a mobile application or through a web application. In the end, the result is the same through both of these options: the items are home delivered without any effort from the buyer and being less time consuming.

Regarding online shopping, we also need to mention a very important actor: the courier companies. These companies take the products or the packages from the sender and deliver them to the receiver. This entire process helps the way that the desired product gets from the warehouse to the receiver in almost no time, even maybe in the very same day. The arrival of the order is influenced by the location where the destination point is and sometimes by the shopper options.

Given these aspects, the courier companies started to grow more and more. The payment for the ordered items can be done either online or to the courier. Also, an important aspect is that courier companies

deliver packages from one person to another.

The main benefit of this entire process is that the delivery is way less time consuming, giving the opportunity to place orders with emergency status (this means being delivered very fast). The competition between courier companies is at the level of delivery speed. Therefore, it is intended to minimize the time the customer has to wait in order to receive the package.

The entire process described above can be improved by a mobile application. Most people nowadays own a smartphone or even a tablet, and most of the activities that we could only do from standing at the computer or at the laptop can also be accessed from the phone. Mobile applications are widespread because they offer, through a user-friendly interface, benefits within certain domains. An application for a courier company helps the customer to take a new step towards the easy delivery of packages.

### 2 Similar applications

At the moment, the courier companies are on the rise, being the preferred method in the case of people who want to move a particular object from one area to another. Every company that works in this activity field has a web page or mobile application, through which is connected with the potential customers, providing relevant information about the services they offer.

A very popular application is the one from FAN Courier. It provides the ability to monitor order status using AWB Tracking technology, generating an estimated order price, dynamics map viewing, having a user-friendly simple interface, but very comprehensive.

A similar application is the AfterShip Package Tracker. This application main purpose is keeping track of the location where the order placed by the client is located. This application is an important support for other popular application used in this domain. The features they use consist in: notifications sent by the application, data synchronization, no ads and it can be used free. The process of using this application consists in typing the order number, then the application will identify the package. It also has the ability to scan the barcode.

A well-known courier company is FedEx. It has an application that implements order tracking, notifications sent to the client through the app, and the ability of using it after the creating of an account. An interesting functionality is the implementation of digital signature. The application also has the ability of scanning the barcode, estimate the price and the time the package will reach its destination.

The 17TRACK application also aims to monitor your orders. It provides support for international courier companies. Its main functionalities are: code scanning, changing the language in which the application can be used, etc. The application works on mobile devices and also on tablets. Once the status of an order

has changed, the client is notified. A very important feature is that the application synchronizes its data in cloud.

Another courier application is CourierManager that can be used in correlation with the web page. Its functionalities are related to the physical track of couriers through a map, generating information about their previous experience. The application is connected to a portable printer and it can monitor the amount of money that has been paid. Google Navigator and Google Maps applications are used to generate routes. It also allows the display of order's content.

### 3 Designing the application

In order to access the application, the client needs to create an account. Through it, the user can access all the functionalities offered by the computer system. Account creation is done by filling in a form. Subsequently, when placing an order, it is necessary to fill in a form. Once the order has been sent, it can be reviewed and its status can be tracked. Also, the saving of an order is marked by receiving a confirmation email. The order is picked up by the courier. His account is created by the administrator and grants him access to all the functionalities within the application. The courier takes the order to the warehouse to be sorted and delivered to the warehouse situated in the delivery sector. From that point the order will be taken over by another courier who is commissioned to deliver the package to the consignee. The application's functionalities are presented in Figure 1.

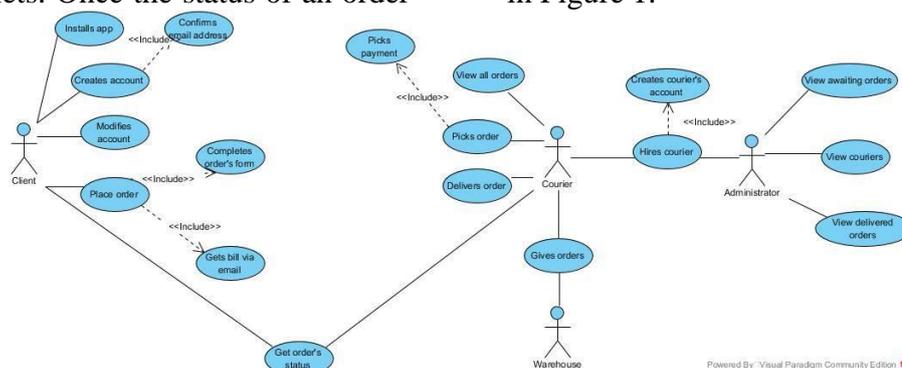


Fig. 1. Use case diagram

The most important cases in the application are: placing and delivering an order. Figures 2 and 3 illustrates the two use cases. The customer needs an account.

Then the order is registered and appears as an awaiting order for the couriers in a particular sector. The client receives a confirmation email.

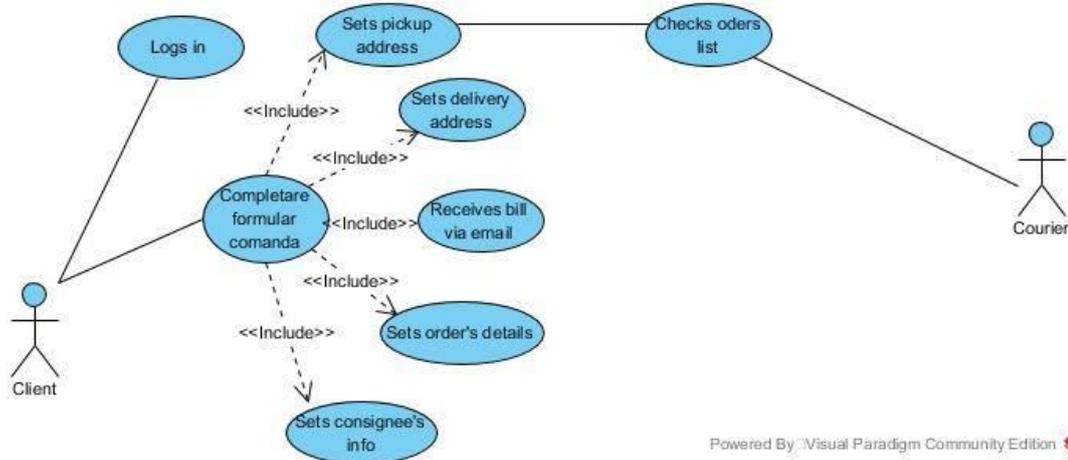


Fig. 2. Placing an order

In Figure 3 is illustrated the case of picking an order by courier and delivering it to the warehouse. The courier has to be logged in

the application. The courier needs to go to the picking point and take the package along with the payment.

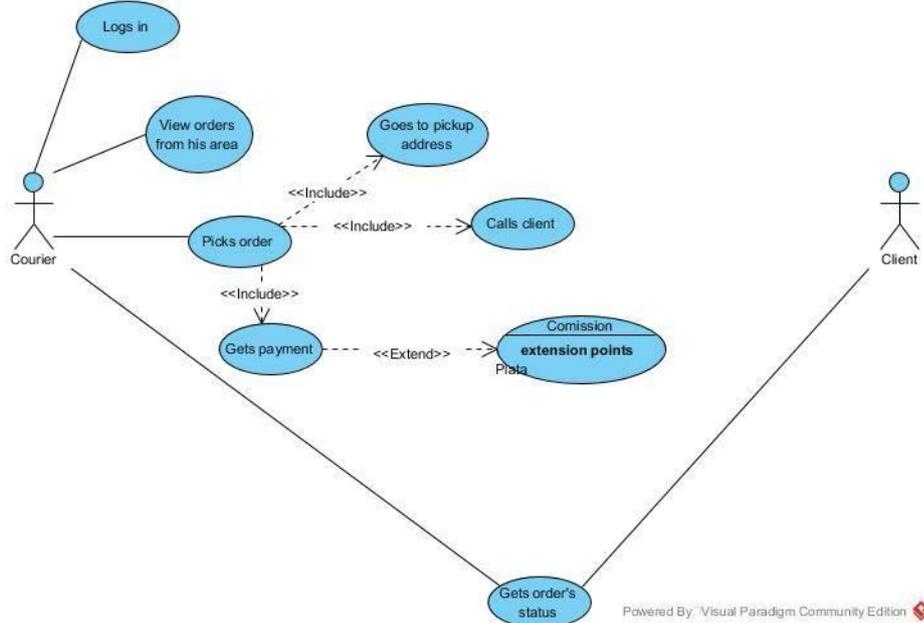


Fig. 3. Picking up an order

The creation of an account only applies to clients and couriers, administrators having their accounts created previously. The user opens the application, accesses the client menu and completes the registration form. If the fields from the form are not validated then the form cannot be submitted. Once the account has been successfully created, the email address needs to be confirmed.

After this step, the account can be used without any other restrictions. If the email address is not confirmed, the account cannot be accessed. The process of creating an account for the employee starts when the courier is employed. The administrator creates the account for the courier. In Figure 4, the activity diagram is shown.

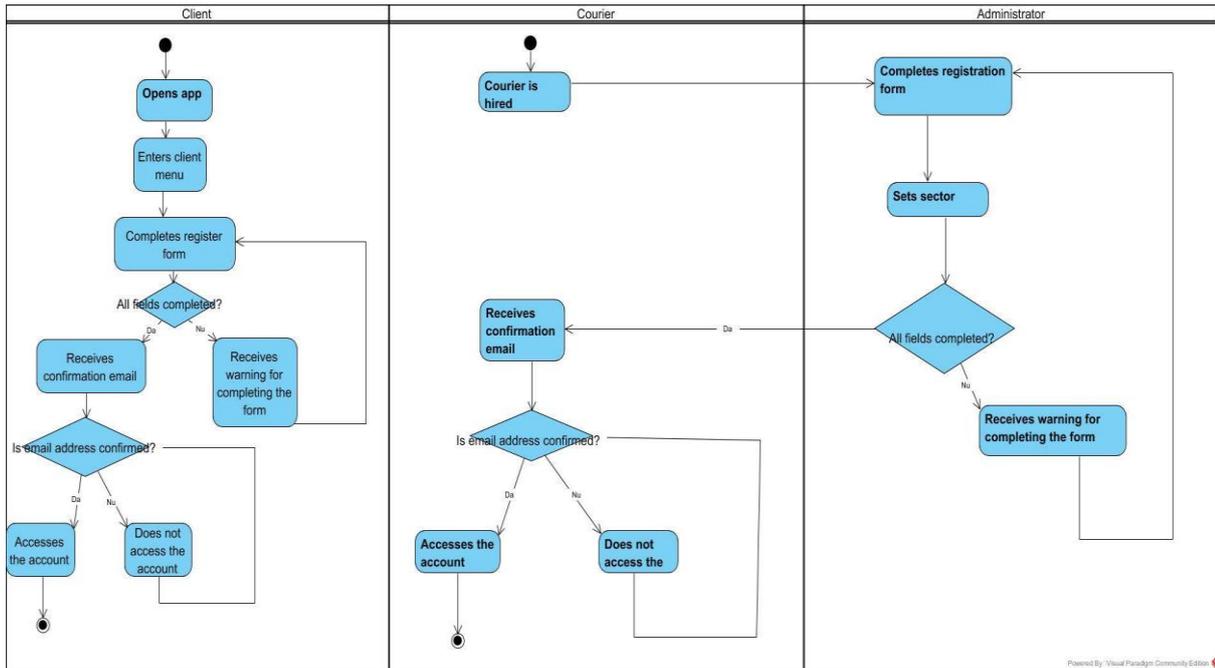


Fig. 4. Activity diagram for the creation of the account

Figure 5 shows the process of picking an order. This involves client-side authentication.

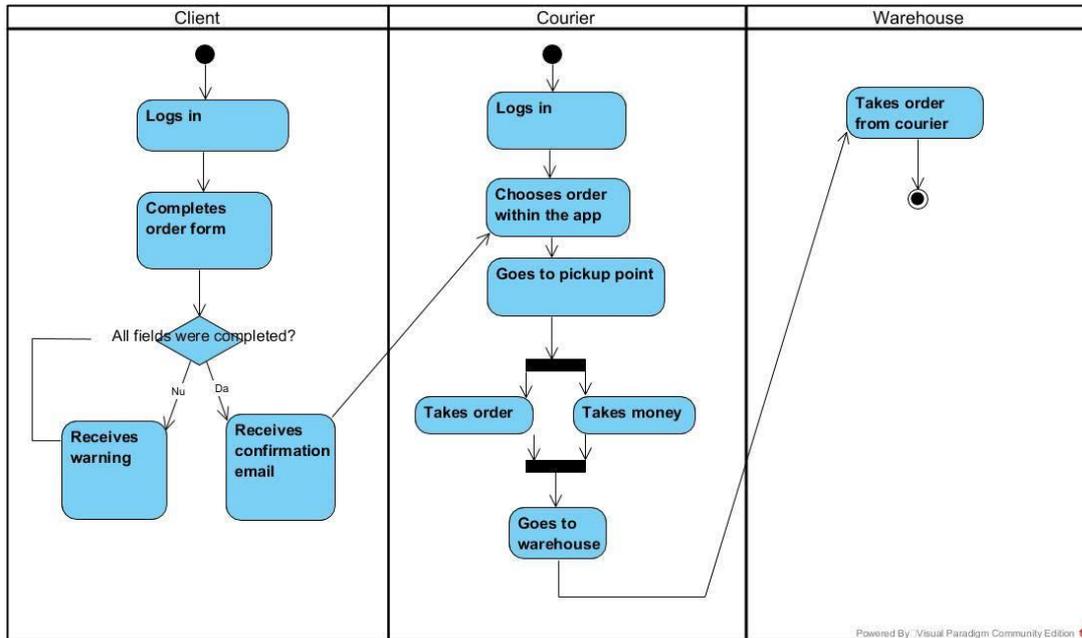
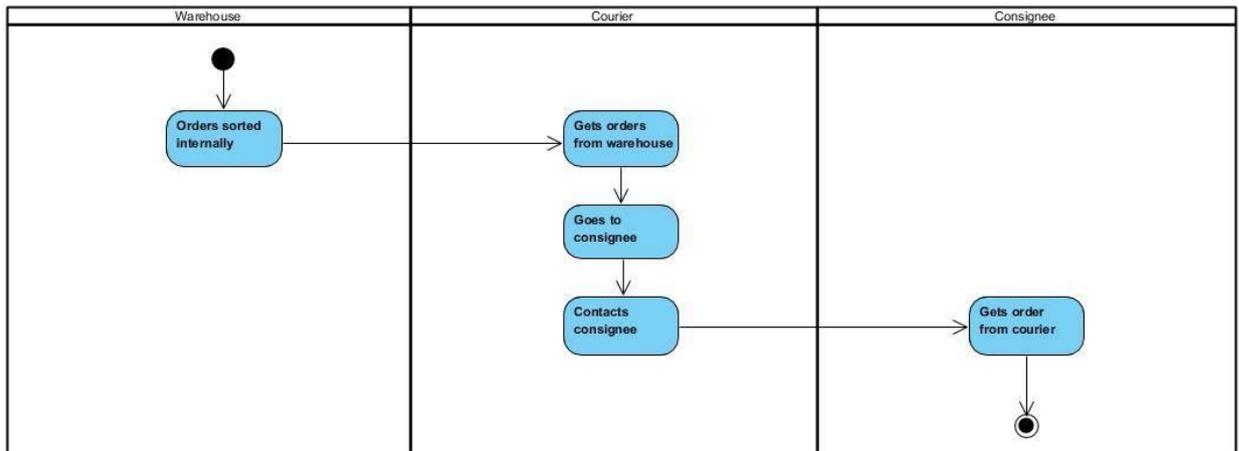


Fig. 5. Picking up an order

Following internal sorting within the warehouse, the order is picked up by the courier. It goes to the destination, contacts

the recipient and takes over the command in the possession of the courier (as in Figure 6).



**Fig. 6.** Delivering an order

The BPMN process diagram illustrated in Figure 7 represents the way the application works in the client-specific interface. This process is executable and basically shows

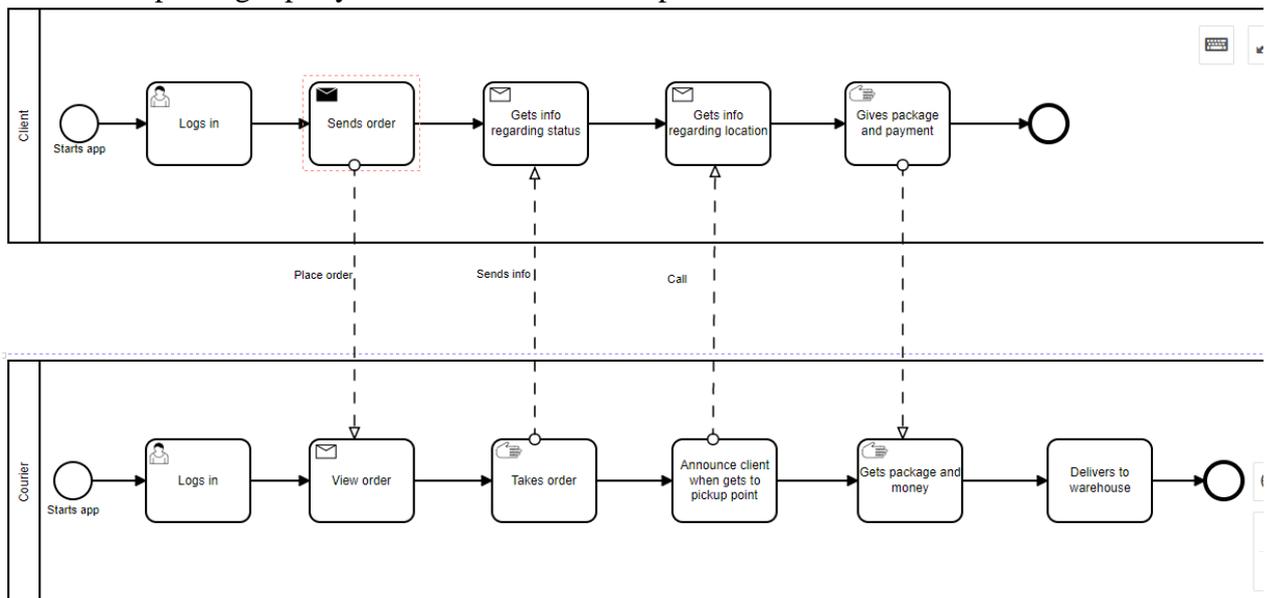
the main thread of actions a client is naturally pursuing when using the application.



**Fig. 7.** Process diagram

The diagram illustrated in Figure 8 is a BPMN collaboration diagram. It deals with the process of placement of an order by the client and picking up by the courier. It

highlights the exchanges of information between the two entities and the way they communicate with each other during this process.



**Fig. 8.** Collaboration BPMN diagram for placing an order

An important process within the application is the delivery of an order. To

illustrate it, we use a BPMN collaboration chart (as in Figure 9).

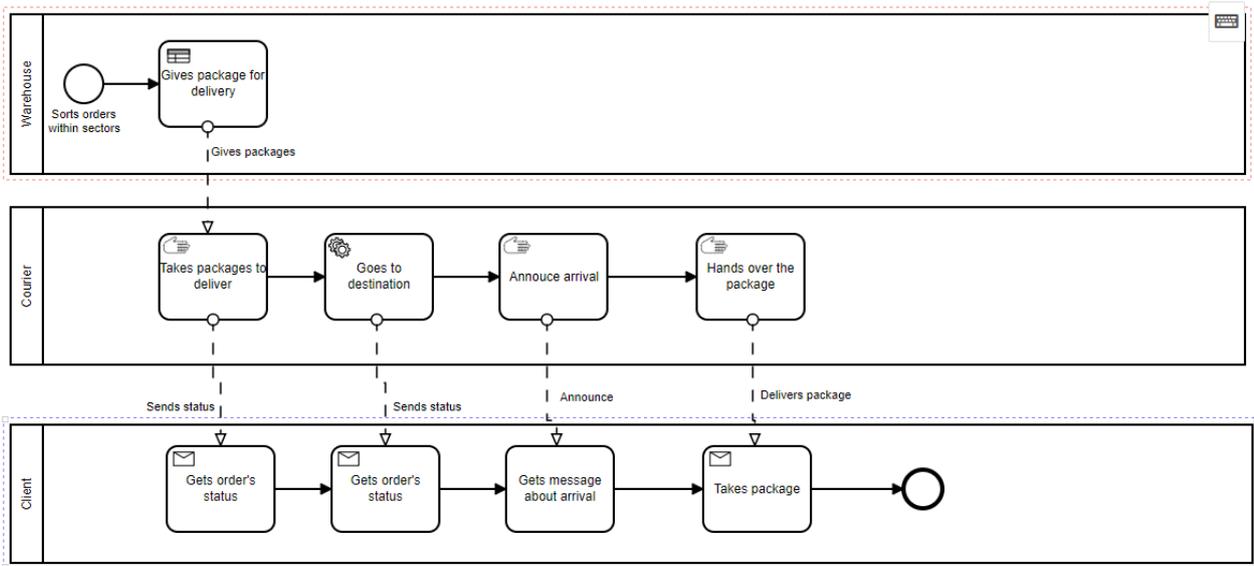


Fig. 9. Collaboration diagram for delivering an order

The Database schema is represented in Figure 10.

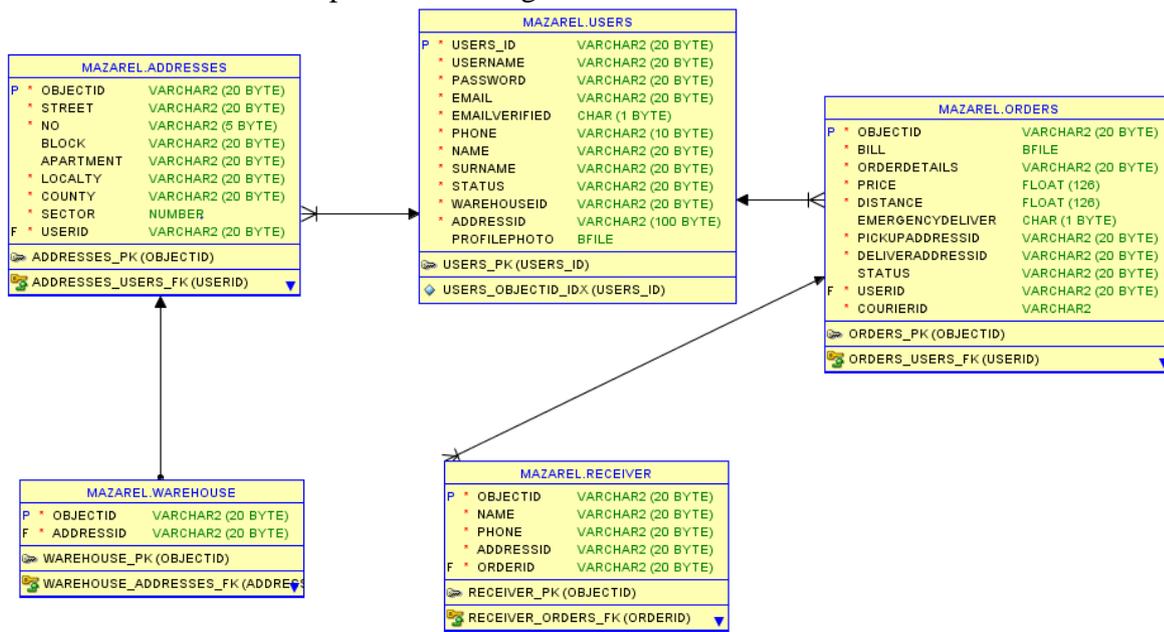


Fig. 10. Database schema

#### 4 Software technologies

Parse server is an open version that supports any type of infrastructure running on Node.js. The Parse server has some essential features, such as:

- Uses a MongoDB database and does not depend on a database hosted in Parse;
- Provides the ability to migrate any existing application to its own infrastructure;
- The built-in application based on a Parse server can be tested and developed locally using Node.

-The server does not depend on a backend hosted in Parse [2].

The databases used in the Parse server are MongoDB and Postgres. For security of the application, it is not necessary to use a client key. The server itself includes client key, REST API, .NET and JavaScript. Each application has a unique id through which the application security process is performed. Also, the read-only key is available to the developer. By enabling it, the server will perform all reading operations, but will not be able to perform

any write operation. This type of key is used in the dashboard.

*MongoDB.* It is a NoSQL database. NoSQL concept includes a huge variety of technologies that have emerged due to demand for the development of modern applications [3]. Application developers process very large volumes of rapidly changing data such as: structured, unstructured, semi-structured, polymorphic.

*Android Studio.* The structure of a project in Android Studio includes Android modules, bookstores and Google apps. The language used in Android Studio is Java. The most important files in a project made in Android Studio are: Manifest file, Java files and res files. The Manifest file expresses permissions that allow the application to access various pre-installed applications on that device. Java files represent the place where all the functions and functionality of the application are implemented, representing the behavior of the application during its use. Files that do not contain code are stored in res files. These are XML files that help add visual elements to the application, such as buttons, colors, images, etc. [4]. Android Studio can be integrated with Google Services, a way to make it easier to deploy features like Google Maps, Google Direction.

*Google play services.* Google Play Services can use the latest and most popular APIs without the need for support on your device. When versions of Google Play are undergoing improvements, they are automatically distributed and delivered via Android SDK Manager [5].

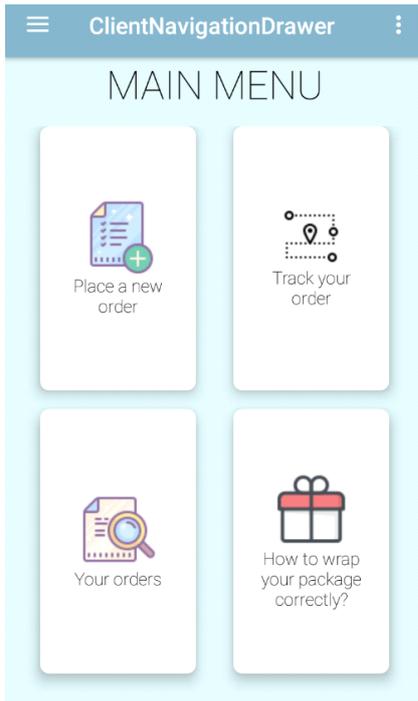
*Google Maps.* The direction arrow is intended to find the route from a starting point to an arrival point. Through it, one can get directions by various means of transport, such as walking, driving, public transport or pedal. At the same time, one can specify destinations, stop points in the route as text, longitudinally or latitude coordinate, or location IDs [6]. Through this API, optimal routes are returned, and the first factor behind the optimization of a route is the time spent on the route. Also, in

the process of calculating the optimal route, there are other important factors that account for, such as the distance and the number of curves.

*The travelling salesman problem.* Generating an optimized route for delivering and taking orders, it is an important step in completing the menu for the courier interface. To implement this functionality, we use the Google Maps API. By querying the database, we took over all the addresses to which a courier has to arrive within a day and created a link. A JSON file is found when accessing the address. The link should follow the format provided by Google Maps. Within the JSON file, the addresses are stored in an optimized order. Also, the directions the driver must follow to reach the destination points are included. With Intent objects, the link must be sent to the Google Maps app. Then, it was necessary to create another address that respects a different format to be compatible with the map application. So, we obtained the addresses from JSON file first, then stored them inside a list we used to create the new link. This method has ensured that the addresses will remain in the same optimized order. The goal is to find the shortest route between various destinations where the salesman must arrive using the algorithm for the travelling salesman problem [7].

## 5 Application interface depiction

The four major functionalities of the application are: placing an order, tracking the status of the placed order, order history, and how to package parcels as in Figure 11.

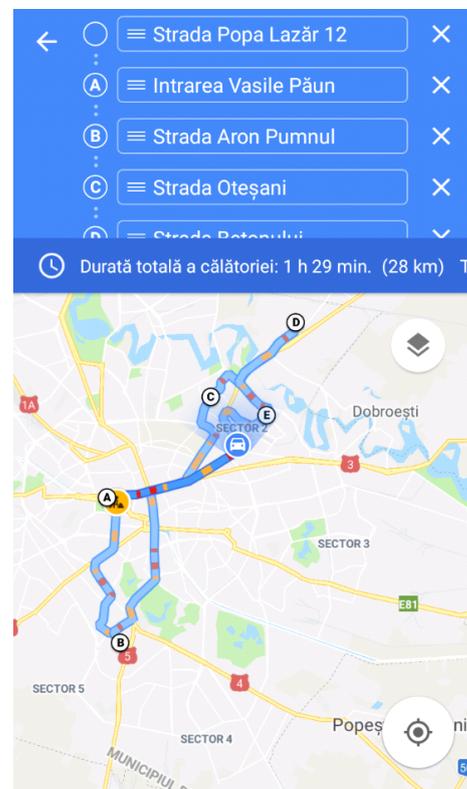


**Fig. 11.** Client interface menu

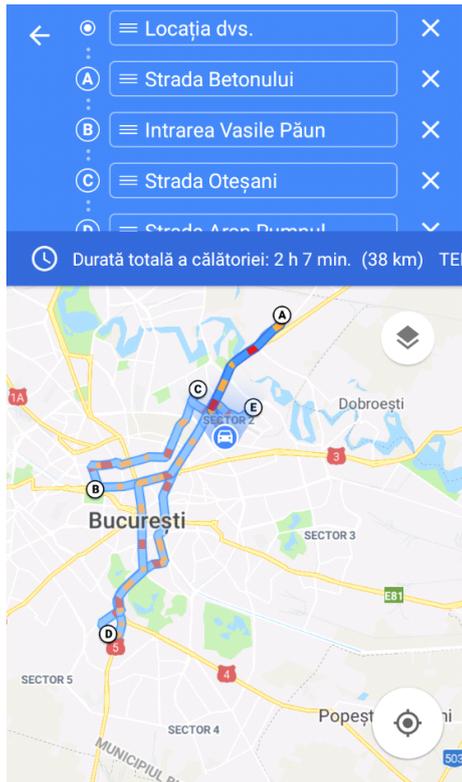
The order placement form gives the client the possibility to enter a new address or use one of the saved ones so far. Once the user sends commands from or to a specific address, it is automatically saved in the database. It must then enter information about the recipient, such as: name, first name, telephone number. It also requires entering information about the subject you are sending, such as the number of pounds or envelopes, weight, length, width, height. These fields contain validations, and it is not possible to send very large or heavy objects. The price is calculated dynamically, based on the weight and size you entered. The user may opt for an emergency delivery, but it will cause an extra charge. It is not acceptable to send it to the server if the data are not complete or valid and the required fields are left blank. Errors are generated on the fields where the entered data did not meet the required requirements. Once the form is sent to the server, the customer receives the invoice by email. This is in pdf format and contains the important information in the order, plus the unique QR code of the order. The invoice must be printed and glued on the package to be scanned later

by the courier.

The basic functionality in the courier interface is the way the route he has to take is optimized. This route must be optimized in terms of both distance and time. The route is made up of all the addresses he needs to reach, and if additional addresses are added, the route is reordered to include the new added point. Figures 12 and 13 illustrate two routes, one optimized, and the other non-optimized, in which the addresses are given in a random order. The first criteria to be considered in the optimization process is the distance traveled, observing that in the optimized route, the distance is 28 kilometers. In the non-optimized route, the distance traveled is 38 kilometers, a major difference in the resources that the company should allocate to the same transport. Also, the time difference is approximately one hour between the two routes. It can also be seen that addressing is done in a different order within the two routes.



**Fig. 12.** Optimized route



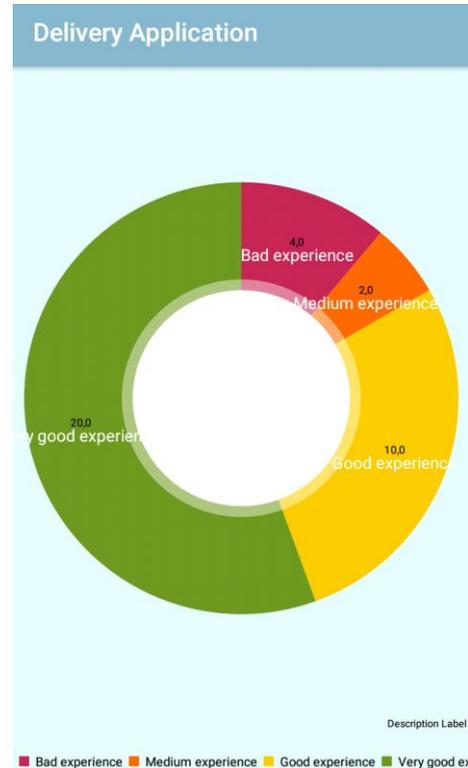
**Fig. 13.** Non-optimized route

The interface for administrators is completely different from the ones described above, both in terms of design and functionalities. The menu is based on three windows that illustrate the stage of the entire business activity. The first window lists all company employees, along with contact details. This enables them to be contacted if it is necessary. In the second window are all commands that have not been retrieved at that time. In the third window are all commands that have been delivered to the recipient. This includes the reference of all orders and the way they are delivered.

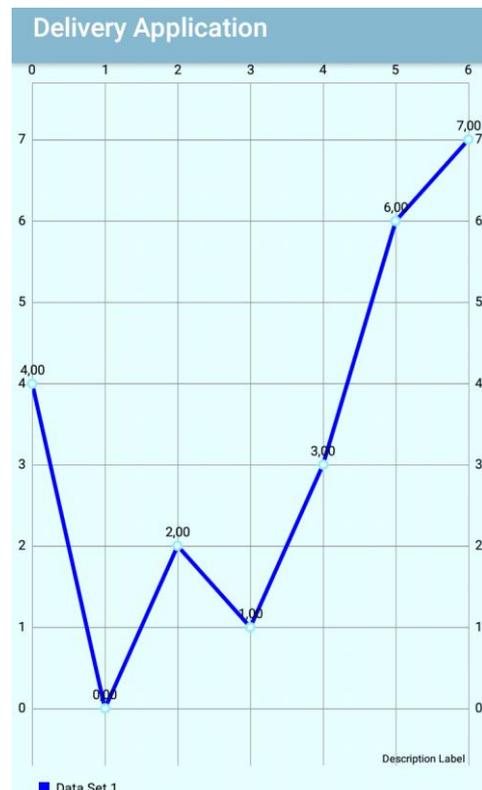
The application also generates graphs that help explain the company's performance from a financial and productivity point of view. The first graph refers to the ratings obtained from the feedback process. These are illustrated as in Figure 14. In this way, conclusions can be drawn on how customers perceive the services offered by the company.

The second graph illustrates the number of orders delivered during the last week (Figure 15). This way one can see how

much the company's productivity increased or decreased, as the case may be, from one day to the next, or the level of demand for services offered by the company.



**Fig. 14.** Feedback chart



**Fig. 15.** Last week deliveries

## 6 Conclusions and future works

In this paper, we propose a mobile application for carrier activities with optimization route using algorithm for the travelling salesman problem that improves the performance. By the optimized routs, both distances and transporting time were reduced.

## References

- [1] Anca Andreescu-Dezvoltarea sistemelor software pentru managementul afacerilor, București, Editura ASE, 2010.
- [2] Parse Server guide, 2010. Available online at: <http://docs.parseplatform.org>, accessed June 2017;
- [3] MongoDB guide, 2012. Available online at: <https://www.mongodb.com>, accessed June 2017;
- [4] Android Studio guide, 2012. Available online at: <https://developer.android.com>, accessed June 2017;
- [5] Erik Hellman-*Android Programming: Pushing the Limits*, Editura Wiley, 2015.
- [6] Google Maps Platform guide, 2012. Available online at: <https://developers.google.com>, accessed June 2017;
- [7] Hector Ortega-Arranz, Diego R. Llanos, Arturo Gonzales-Escriano-*The Shortest-Path Problem: Analysis and Comparison of Methods*, Editura Morgan&Claypool, 2014.



**Luiza-Cecilia MAZĂRE** (b. January 6, 1996) is a third year student at the University of Economic Studies from Bucharest, Faculty of Cybernetics, Statistics and Economic Studies. She has graduated Theoretic High-School “Grigore Moisil” from Tulcea in 2015 with specialization in Mathematics-Informatics, advanced English. She is familiar with Java, C# and Microsoft Dynamics 365.



**Simona-Vasilica OPREA** (b. July 14, 1978) is an Assistant at the Economic Informatics Department at the Faculty of Cybernetics, Statistics and Economic Informatics from the Bucharest University of Economic Studies of Bucharest. She has graduated the Polytechnic University in 2001, holds a Master Diploma in Infrastructure Management Program, Yokohama National University, Japan in 2007, a PhD diploma in Electrical Engineering in 2009 and a PhD diploma in Economic Informatics in 2017. She is the author of over 50 articles.