Modeling and dimensioning of the resources in a delivery restaurant chain using queuing theory

Author: GUILLEM ONTIVEROS LÓPEZ

Academic Advisor: ESTEVE PALLARÈS SEGARRA

A Degree's thesis submitted in fulfillment of the requirements

for the degree in Science and Telecommunication Technologies Engineering

in the

Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona,
Universitat Politècnica de Catalunya

Revision history and approval record

Revision	Date	Purpose
0	17/09/2015	Document creation
1	23/11/2015	Document revision
2	10/01/2016	Document revision
3	23/03/2016	Document revision
4	10/04/2016	Document revision
5	27/04/2016	Document revision
6	13/05/2016	Document submission

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Guillem Ontiveros López	guillemontiveros@gmail.com
Esteve Pallarès Segarra	esteve.pallares@entel.upc.es

Written by: Guillem Ontiveros		Reviewed and approved by:	
Date	17/09/2015	Date	
Name	Guillem Ontiveros	Name	Esteve Pallarès
Position	Project Author	Position	Project Supervisor

Table of contents

List of Figures

1. Abstract

This project aims to design a tool that can help managers of delivery & take-away food business with detecting problems in the processes and in making management decisions related to the quality of service.

In the following pages you can see how, by using existing queuing theory concepts and the data taken in an actual delivery & take-away restaurant, the whole system is described and modeled analogously to a typical queuing theory problem, most often used in the network planning sector.

The board has approved for a first prototype to be developed in order to validate if the investment in developing an advanced system can be valuable for business objectives. This first prototype has been designed and developed by defining the most common management variables in such a business and with actual data as input that can help simulate and further understand how different factors can affect the key performance indicators of the quality of service of the business.

Finally, further development is proposed in order to improve the accuracy of the tool and integration with existing delivery & take-away front-end software, which the board will need to approve in order to go further with the development.

2. Introduction

2.1. Statement of purpose

The project has been carried out at Telemaki Tienda SL (Telemaki from now on) and at the department of Telematics Engineering in the Escola Tècnica Superior d'Enginyeria de Telecomunicacions de Barcelona (ETSETB) at Universitat Politècnica de Catalunya (UPC).

Telemaki is a Japanese delivery restaurant chain operating in the area of Barcelona that started in 2013. Nowadays, Telemaki has 5 restaurants, which make them the Japanese delivery restaurant chain leader in Catalonia. One of the main reasons that has made this quick expansion possible was realizing one thing: what clients in the delivery sector appreciate the most is quality of service, both in terms of total waiting time of delivery of an order and the compliment with the waiting time they are informed. This is why quality of service is one of the key marketing tools to foster customer's loyalty.

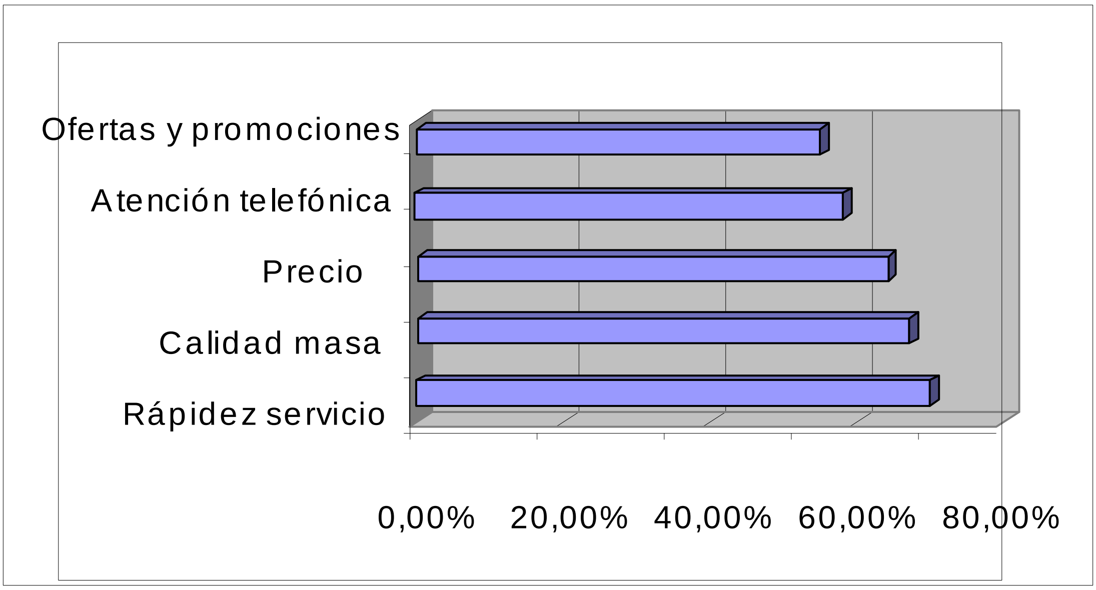

Figure 1. Most valued qualities by delivery "pizza" consumers (Source: "La Gaceta de los Negocios", 7.2.97.)

Therefore, over dimensioning those resources that allow for a faster service in the delivery sector (mainly human capital costs) doesn't have to be seen as a waste of resources, but as an investment in loyalty marketing, the most important type of marketing for such businesses (their potential customers are the people living in one specific area, not being able to attract customers from other areas without new openings).

Forgetting about the quality of food for a moment, it is easy to see that the shorter the service time of the orders, the greater the likeliness of your client's ordering again. However, in a low-cost business there is no expense that can remain uncontrolled; otherwise economic results can disappear in a blink of an eye. A low-cost company that is always over dimensioned in order to offer a great service will likely have trouble to achieve decent economic results.

Hence, we have identified the most important trade-off in the sector of delivery food: short-term economic results against long-term viability depending on how fast you can place your product at your client's home.

One of the greater problems regarding this trade-off is that demand is extremely concentrated at weekend nights. Moreover, for a same level of absolute demand, the quality of service can vary extremely depending on the statistics of the orders' arrivals (for a number N of orders arriving during a 3 hour night shift, the quality of service offered is obviously higher if the orders arrive uniformly distributed over the period rather than arriving in batches of size N/3 every hour). Thus, these businesses have developed the need to optimize the dimensioning of the human resources available at the stores at every time in order to guarantee a decent service without compromising economic results.

All of this looks very similar to a network planning and design problem, being Telemaki the operator of the service and its clients the subscribers. This is why we propose to solve the problem of dimensioning the restaurant human resources as a network problem, using queuing theory.

The company should know what average serving times they will have depending on the resources they use for every task in the restaurant. With this statistical data used as input to dimension the system, there should be enough information to program a simulator that could be used for:

a. Managers of the restaurant to adequately plan the human resources need to achieve a certain quality of service.
b. Workers of the restaurant to get information of the estimated service time at every moment, so that the information they provide to customers is as feasible as possible.

2.2. Requirements and specifications

Project requirements:

- The model has to be suitable for all the restaurants of the chain, regardless of their radium of action.
- It has to be compatible for all the possible states (the kitchen servers described will have different service rate depending on the people working at the moment).
- It should not be order dependant.
- The simulator should be easy programmable, so that it can work real-time without affecting the front-end software used at the restaurants.
- It should be scalable so that further improvements can be achieved

Project specifications:

- The simulator should have at least 10 minutes precision, the average deviation time at which a customer calls to ask for their order.

2.3. Report Structure and Work Plan

This chapter has started with the statement of purpose to define a problem that requires an engineering solution. Hence, goals, requirement and specifications have been explained to know exactly our objectives.

After this Introduction chapter, the following chapters explain the work and process done to reach a feasible solution.

- Chapter 2 - State of the art: this chapter describes the theories and techniques used to create the model and the suitable technologies for the simulator.
- Chapter 3 – Data acquisition: this chapter describes how data of every part of the system has been taken and defends the approximations that have been made.
- Chapter 4 - Modeling: in this chapter, with the data acquired in the previous chapter and the theories explained in the State of the Art, a dimensioning model is proposed
- Chapter 5- Simulator: in this chapter, its shown a real-time simulator implemented for the project.

- Chapter 6 - Budget: all this project has a cost and it is described in this chapter
- Chapter 7 - Conclusions and future development: final considerations of the development of the work and future lines are explained.

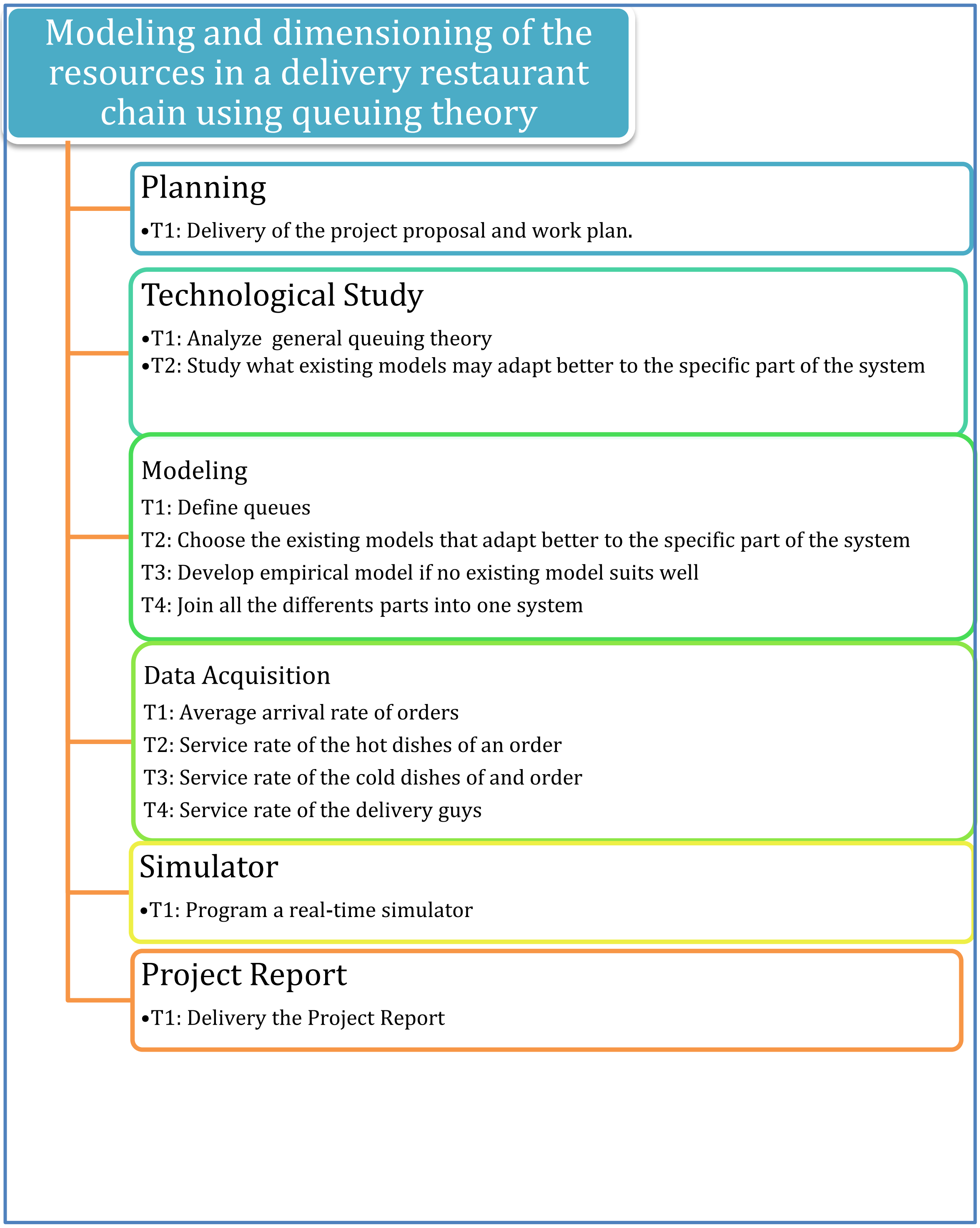

Figure 2. Work Plan Diagram

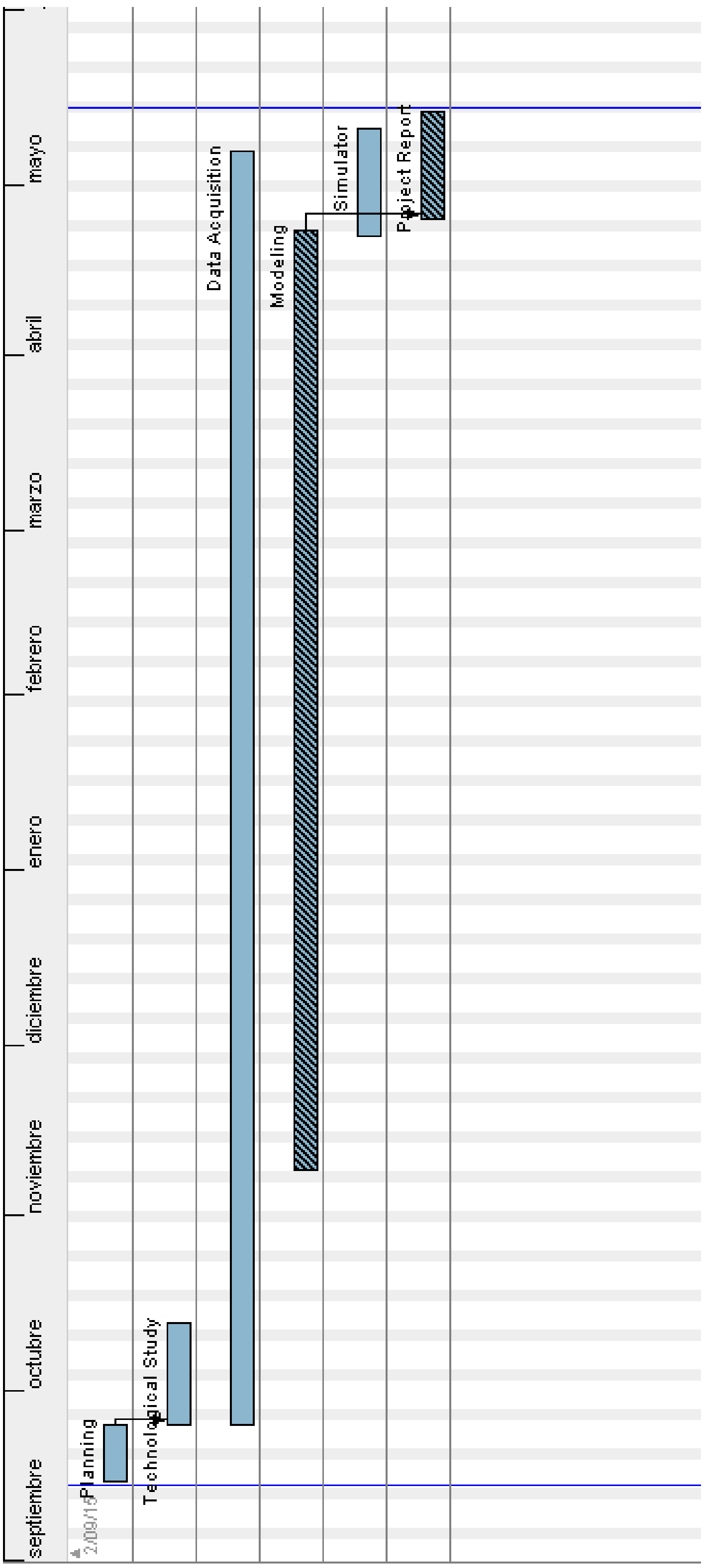

Figure 3. Updated Gantt Diagram

3. Technological Study: Queueing Theory

3.1. Introduction to queueing theory

Queueing Theory is a field of mathematical study dedicated to the understanding and modeling of waiting lines.

The purpose of queueing theory is to construct models so that the service waiting time of a client can be predicted, given the system characteristics and the current state of the system.

Throughout history, it has been strongly related with telecommunications, given that it originated alongside the telephone networks. Nowadays, it has a much broader area of application, since it is a very useful tool in business operations, at the time of deciding the resources that need to be deployed in order to provide services that comply with the quality standards of companies.

3.2. Notation

First of all, it is important to define the most common notation used in queuing theory.

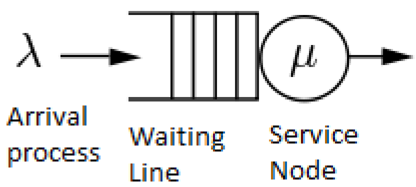

Figure 4. A queuing node

The most common one is that defined by Kendall [1], known as the Kendall's notation. With this notation, we can fully describe a queuing node by defining a few factors:

Those are:

1) A: distribution function of the arrival times
2) S: distribution function of the service times
3) m: the number of servers
4) K: capacity of the system, the maximum number of customers in the system including the ones being serviced

5) n: population size, number of sources of customers
6) D: the queue's discipline

Therefore, a queue can be described as A/S/m/K/n/D, once these factors are defined.

3.3. The arrival process

The arrival process of a queue is the model that describes how the customers arrive (e.g. singly or in groups) and how this arrivals are distributed over time (e.g. what is the probability distribution of time between successive arrivals, or the interarrival time distribution). Usually, it is assumed that the interarrival times are independent and have a common distribution. The most common processes are:

1) M: Markovian or memoryless - the arrivals follow a Poisson process. A Poisson stream of arrivals corresponds to arrivals at random, in which successive customers arrive after intervals which independently are exponentially distributed. Such processes can be modeled with just one variable λ, which represents the average rate of arrivals (arrivals/unitarian period of time).
   The probability of having x arrivals in a Unitarian period of time follows the next formula:
   $$P(x, \lambda) = \frac{\lambda^{x} * e^{-\lambda}}{x!} \text{ [3]}$$
2) D: Degenerate distribution - a deterministic arrival time.
3) G: General Distribution – Independent arrivals.

3.4. The service time distribution

The service time distribution gives us the distribution of the service time of clients. Just as the arrival process, it can follow several defined models, being the most common ones the following:

1) M: Markovian or memoryless - the service time is an exponentially distributed random variable.
2) D: Degenerate distribution - a deterministic service time.
3) G: General Distribution – Generic service time.

3.5. The number of servers

This is the number of servers that exist within a system. For example, in a call center with 10 operators, n would be 10.

3.6. The number of places in the system

The number of places of a system is the total number of clients that are allowed in a given system. That is, the total number of clients that can be served at a given time (which coincides with the number of servers) plus the total number of clients that are allowed in the queue or waiting line.

3.7. The calling population

The calling population is the total size of the calling source the system has. When omitted, it is considered infinite. This parameter is important because when the calling population is very limited, the system state will affect highly in the arrival rate (the more people are either being served or waiting, the less likely it is that the system will have new requests).

3.8. The queue's discipline

This is a fixed parameter that the service provider must define. It will give us the priority that certain clients have above others to start being served.

The most common Service Disciplines are the following:

1) FCFS: First Come First Served
2) LCFS: Last Come First Served
3) SIRO: Service In Random Order
4) Shortest job first

3.9. Customer's behavior of waiting

There are also some decisions that can be taken by the clients that affect the queueing model. Some examples of these are:

1) Balking: customers deciding not to join the queue if it is too long.
2) Jockeying: customers switch between queues if they think they will get served faster by so doing.

3) Reneging: customers leave the queue if they have waited too long for service.

3.10. Examples of queues

Depending on the value of the previous parameters, we can have many different queue nodes.

For example, we could have a node with Markovian arrival process, a general distribution as service distribution, 2 servers, a maximum capacity of 100 clients and population of 1000 clients served with LCFS priority. Such a queue would be noted as:

M/G/2/100/1000/LCFS

Each of the last three parameters can be obviated when the capacity can be considered infinite, the calling population infinite and the queue's discipline is FCFS.

Therefore, with generally distributed arrivals served with Markovian distribution, 1 server, capacity and calling population infinite and FCFS priority, the queueing node could be notated just as:

G/M/1

The problem of queuing modeling has been solved for certain types of queues, so the first thing one has to do when facing a queueing problem is trying to find the models or codes that best suit the parameters in the problem to solve, because many times the solution can be found immediately.

4. System definition

First of all, we are going to explain the way a Telemaki Restaurant works and define a model that simulates the processes that shape it.

4.1. Working modules

At a Telemaki restaurant, there are four clearly different working positions [2], each of which is a queuing node in itself:

1) **Customer service:** There are people whose work is to receive and take note of all the orders. These orders come from different sources, each of which is treated in a different way.
    a. Telephone: when customers call the restaurant directly, the worker manually introduces the order into the front-end software. The client is then informed of the estimated waiting time.
    b. Telemaki's web: when a customer orders via web, the order arrives directly to the front-end and needs only to be accepted, indicating the waiting time. There is no work to be done by the customer service employee.
    c. Telemaki's APP: when a customer orders via APP, the order arrives directly to the front-end and needs only to be accepted, indicating the waiting time. There is no work to be done by the customer service employee.
    d. Just-Eat & La Nevera Roja: When customers order through these platforms, orders arrive at the restaurant via a dataphone-like GPRS machine, informing the client of the waiting time at the time of accepting and is then manually passed to the front-end software, just as if the client was calling.

2) **The kitchen:** Once an order is into the front-end software, three tickets are printed , one with the hot dishes, another one with the cold dishes and the last one with all the items ordered in the customer service area, so that the order can be mounted as the food starts to go out of the kitchen. Therefore, we can identify two different type of workers in the kitchen, each of one being a queuing node:
    a. Hot dishes cooks
    b. Cold dishes cooks

    The order doesn't move to the next process until all hot dishes and cold dishes of the order have been prepared.

3) **Order preparation:** Once the food is outside the kitchen, the customer service workers put it in two bags, one with hot dishes and the other one with cold dishes. Once this is completed, the order is ready to be delivered.
4) **Delivery/Take-Away:** Depending on whether the clients want their order at home or are coming to the restaurant to pick it up, the process would end here – in the latter case - or would need a last step: delivery, which is carried out with motorbikes. Delivery is the last queuing node, with individual servers being each delivery guy and the total number of servers the number of delivery guys working at the time.

"For a client, the total service time would be **the difference between the time at which he has been informed of the waiting time until the delivery guy arrives at his home or until he takes the order away from the restaurant".**

4.2. System definition applying queuing theory

The system is defined by the distribution of order arrivals and by the four working modules.

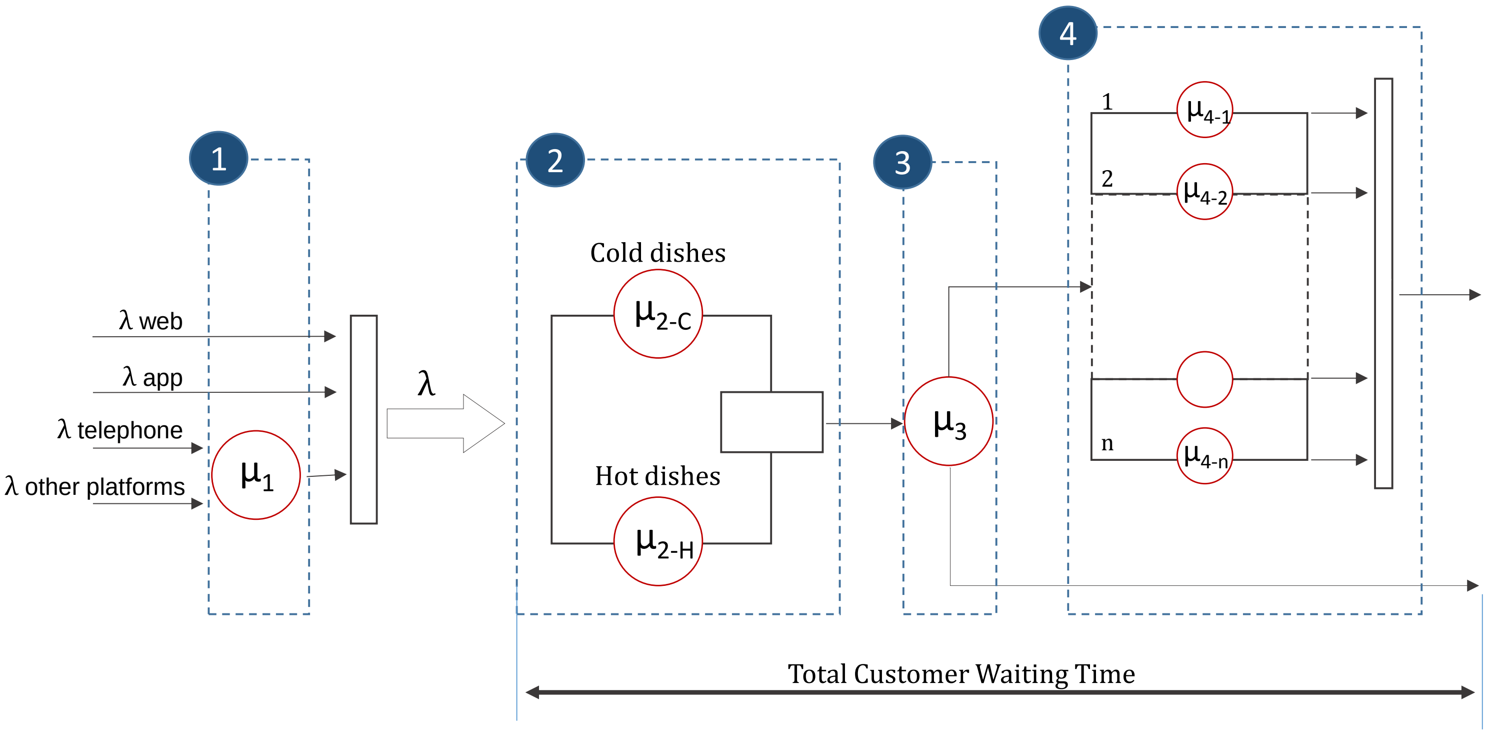

Figure 5. Complete system queuing scheme

Arrival of customers' orders:

The arrival of orders through each of the four channels available is a random variable A_i defined as a Poisson distribution $A_i \sim P(\lambda_i)$, with λ_i being the average arrival rate of orders that arrive through channel i, where i∈[telephone, web, app, other]. The Poisson distribution is an appropriate model for the arrivals because:

- The arrival of a customer order does not affect the probability that a second order will arrive. That is, order arrivals occur independently.
- The rate at which orders arrive is constant through the interval modeled.
- Two orders cannot arrive at exactly the same instant. Although this could be possible, the front-end software will always assign preference to one of the orders.
- The probability of arrival of an order during an interval is proportional to the length of the interval.

If $X \sim P(\lambda_x)$ and $Y \sim P(\lambda_y)$ are two *independent* random variables following a Poisson distribution with parameters λ_1, λ_2, then Z=X+Y is a Poisson distribution $P(\lambda_z)$ with $\lambda_z = \lambda_x + \lambda_y$.
Since the arrivals to each of the four channels is independent from the others, the total number of orders arriving to the restaurant is a random variable A following a Poisson distribution

$$A \sim P(\lambda), \text{ with } \lambda = \lambda_{telephone} + \lambda_{app} + \lambda_{web} + \lambda_{other}$$

where λ is the arrival rate of orders into the restaurant (orders/second).

1) **Customer service module:**

Only orders arriving via the telephone or other online channels different than the web or the app need some processing from the customer service workers, which are modeled as a queuing node.

It is fair to assume that both channels imply the same amount of processing work, and thus are modeled together as a single node. The service time t_1 in this node is assumed to be exponentially distributed with parameter μ_1, which is the service rate of the customer service worker in this node (orders/second). Orders exiting the node are automatically delivered to the front-end software, just as the other channels that do not need processing work. ω_1 is the waiting time since the order arrives until it is moved to the software.

The rate at which the orders exit the node and are delivered to the front-end software is then

$$\max\{\lambda_{telephone} + \lambda_{other}, \mu_1\}$$

Assuming that there is no saturation in the web and app channels, i.e. orders that arrive through these channels are automatically delivered to the front-end software, the average arrival rate of orders into the software will then be:

$$\lambda = \sum \max\{\lambda_{telephone} + \lambda_{other}, \mu_1\} + \lambda_{app} + \lambda_{web}$$

2) **Kitchen module:**

The exit process of the customer service module is the arrival process of the two kitchen nodes. Once an order is in the kitchen waiting line, it has to start to be processed, both in the hot dishes area and in the cold dishes area. An order is composed by a certain number of hot and cold dishes, and only exists the kitchen when *all* the dishes of the order are cooked.

Accordingly, the kitchen is modeled as two independent nodes:

a. <u>Cold dishes node</u>: This node processes the dishes sequentially, i.e. there is only one cold dish prepared at a time by the cooker. It is then modeled with parameter $\mu_{2\text{-}C}$, the average service rate of the cold dishes (dishes/second). $t_{2\text{-}C}$, is the service time of a dish (seconds to be cooked), and $\omega_{2\text{-}C}$ the waiting time of a dish to start to be cooked.

b. <u>Hot dishes node</u>: hot dish workers cook several dishes simultaneously, i.e. there is no sequentiality. $t_{2\text{-}H}$ is the service time of a dish (seconds to be cooked) and $\omega_{2\text{-}H}$ is the waiting time of a dish to start to be cooked.

If h is the number of hot dishes in an order and c is the number of cold dishes in an order, then the time for an order to be cooked is

$$t_2 = \max \{ h \times (\omega_{2\text{-}H} + t_{2\text{-}H}) , c \times (\omega_{2\text{-}C} + t_{2\text{-}C}) \}$$

3) **Order preparation module:**

Once all the dishes of an order are cooked, they need to be properly packed and the order needs to be prepared. Although this is a fairly quick and straightforward process, the fact that the workers who work on preparing the order are the same as the ones who work on customer service might lead to same waiting time too.

This will then be another queuing node of the system, with t_3 being the service time of the node and ω_3 the waiting time for an order to start being prepared.

4) Delivery – take-away module

Depending on whether the order is delivered to the customer's home or is picked up at the restaurant, there is one extra node in the system:

- Take-away: In this case, the order is finished at the exit of module #3. P(take-away) is the probability that an order is picked up by the customer at the restaurant.
- Delivery; P(delivery) is the probability that an order needs to be delivered to the customer's home. In this case, there is one last queuing node. There are n_d delivery workers, each modeled as an individual server. $\mu_{4\text{-}x}$ is the service rate of delivery worker, $t_{4\text{-}x}$ its service time, and ω_4 the waiting time for an order since it is prepared until it is picked up by a delivery worker.

  It is important to note that t_4 is the service time of a delivery worker in the queuing system, i.e. the time between the server that models its behavior gets busy until it is freed, which is the time between the delivery worker collects two consecutive orders to be delivered. It includes the round trip time (one-way trip to the client, the time spent with the client and the trip back to the restaurant), and the waiting time –if any- at the restaurant to collect the next order.

  The relevant measure for the customer is the total delivery time of an order, which is $t_D = \omega_4 + t_{trip}$, the time a ready order has to wait until it is picked up to be delivered and the one-way trip time.

Now that we have the whole restaurant system modeled, we can define the total service time for a client as the difference between the time at which he has been informed of the waiting time until the delivery guy arrives at his home or until the order is ready to be picked up at the restaurant. In our system, this is equivalent to:

Total customer service time = $\omega_1+t_1 + h\cdot(\omega_{2\text{-}H} + t_{2\text{-}H})+c\cdot(\omega_{2\text{-}C} + t_{2\text{-}C}) + \omega_3+t_3 + t_D$

where:

- ω_1 and t_1 are equal to 0 if the arrival channel is either the restaurant's web or app
- t_D is equal to 0 if the order is to be picked up at the restaurant by the client, i.e. with probability P(take-away)

4.3. Assumptions and Simplifications

In order to have a first simplified model of the whole system, we will assume a series of conditions that will make possible a first simpler analysis:

1) We will assume there is no **balking**, that is no one refuses to join the system because the queue is too long. There is indeed a certain percentage of balking, but there is neither actual data about it since the ERP software only considers orders that have actually entered the system nor is it a high percentage.
2) We will assume there is no **jockeying.** The fact is there are clients that do switch queues (they decide they will finally come to take it away since they consider the delivery waiting time too long or change their mind during the process). We will obviate it since when this happens, the order is switched in the ERP and the take-away ratio that we have already takes into consideration.
3) We will assume there is no **reneging,** that is that no one who has made an order decides to cancel it. We will do this for two reasons:
    a. There are a very low percentage of times when this happens: the company's policy is to take it to the client even if he decides to cancel it, with no cost.
    b. When it happens, it is only when the system is completely overloaded and when the client decides to leave the queue, the order has not started to be cooked; therefore it affects neither the kitchen system nor the delivery system.
4) We will assume **infinite calling population**. The typical number of orders that are placed in the busiest nights is 2 orders of magnitude below the total number of clients registered, and several more orders below the total population living in the homes we serve.
5) We will assume **infinite numbers of places in the system.** There is no policy marked that limits the number of people allowed in the system. There are indeed certain days where for overloading reasons orders are not accepted during a short period of time, but we will obviate this fact.
6) We will assume that each delivery guy takes one order per trip.
7) We will assume that the average ticket price is the same for delivery orders and take-away orders
8) We will assume that the ratio of orders per delivery area is constant.
9) All the data taken in the first prototype will be of the busiest shift of the week: **Friday Night**, since it is the one where more optimization can be done and being overloaded will affect the most number of orders.
10) We will assume that the average ticket price is constant during the whole shift and delivery area.
11) We will assume that all delivery guys work during the whole shift.

5. First Prototype of a restaurant model

A first prototype has been designed to simulate the processes that take place in a Telemaki restaurant.

5.1. <u>Implementation adjustments</u>

It is based on the previously defined model for a Telemaki's restaurant but, to facilitate its implementation and further analysis, it incorporates some modifications:

a) Customer service area is neglected

We have assumed that there is **neither waiting time nor service time** at the customer service area ($\omega_1 = t_1 = 0$), thus skipping a whole node of the system. The starting arrivals process is therefore the rate at which orders are entered into the front-end software (which is in fact the same time at which they are printed and join the kitchen queue). We do this for the following reasons:

1. Being the customer service area the one with the fewer problems and shortest service time, the moment at which the order is processed will not usually affect the whole service time of the system, since there is always queue at the kitchen.
2. Most importantly, we do this because the aim of the whole project is to determine which is the time that must be informed to every client at the time of ordering, and this is made once the client's orders has been processed, so the actual waiting time we are informing to a client is not the time between he decides to order and he gets the order home but the time between he has ordered and gets the order home.

b) Hot dishes, cold dishes and order preparation nodes are treated as a single queuing node: Kitchen & Preparation subsystem

As explained previously, there are two different kinds of dishes, each type cooked in a different area of the kitchen. Since we deliver full orders to clients, unlike a restaurant where the dishes have a certain order depending on the client's preferences, an order will not be ready until all the dishes have been cooked and packed.

The original idea was to find a model that described how everything was done based on some parameters. However, we have come across several difficulties that have made it an extremely complex model:

1) Some dishes need action from both the hot and the cold area, creating a big interdependence between both nodes.
2) Whereas the cold area works as a chain – every dish is cooked sequentially – the hot area way of working depends highly on the demand: since most dishes can be cooked simultaneously, the more work there is, the lesser the time between services.
3) Depending on how the service is running, an area can even be stopped in order to align it with the other area (if cold dishes are far behind, hot food may get cold while waiting).
4) Depending on how the service is running, a worker in an area may switch area in order to balance paths.
5) Every cold dish has a different preparation time, varying widely among the entire offer.
6) The workers who work on packing the orders for delivery are the same ones who work on customer service, so sometimes even when an order is cooked it needs to wait for these workers to be free to pack it, creating another interdependence, now with the arrival process.

All this led us to a model where every single cold dish had to be typified with a cooking time and a probability, where a complex demand-dependent model for hot dishes had to be found, and where orders with only hot or cold dishes had to be treated differently, with more statistical information as input. Furthermore, and the main reason we had to change our initial plan, is that the variance between services for all these statistical variables (ratio of only hot/cold dishes, type of cold dishes, amount of demand, etc.) was very high, making it almost impossible to be used in the final goal of the project, which is to be able to simulate shifts.

For all these reasons, we decided to use a more direct method, especially taking into account that this is a prototype that will need testing in order to finally integrate it with ERP software for real-time waiting estimation. Therefore, we will integrate the three different nodes (hot cooking, cold cooking and order preparation) into a Kitchen & Preparation subsystem, only considering the total service time of them.

5.2. Prototype system definition

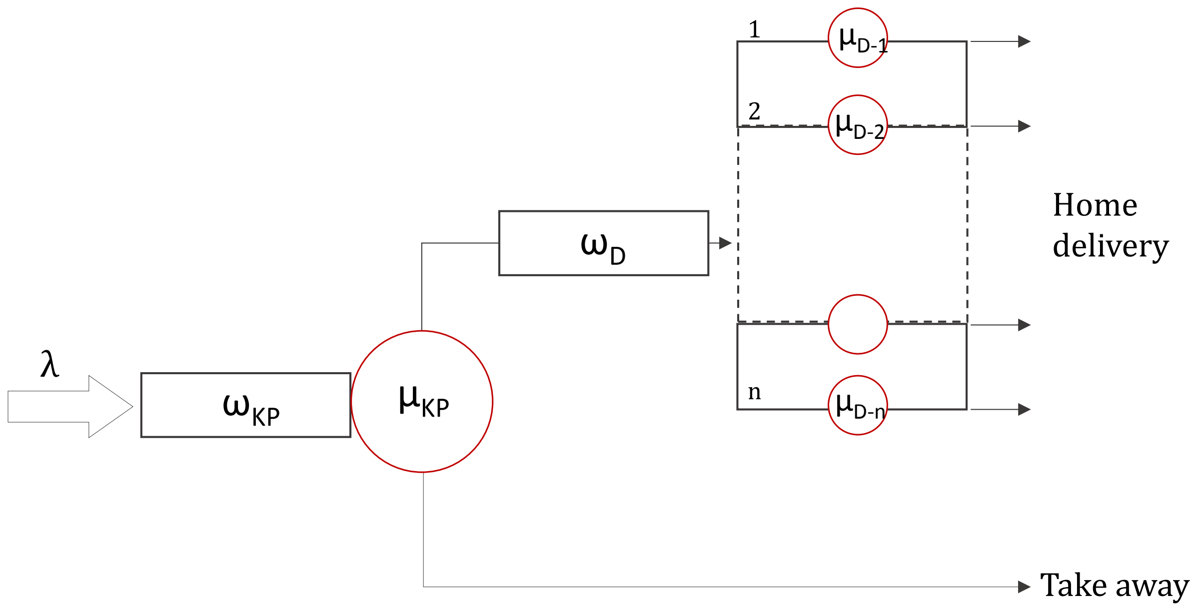

Figure 6. Prototype queuing scheme

Arrival of customers' orders:

As seen before, the total number of orders arriving to the restaurant is a random variable A following a Poisson distribution

$$A \sim P(\lambda), \text{ with } \lambda = \lambda_{telephone} + \lambda_{app} + \lambda_{web} + \lambda_{other}$$

where λ is the arrival rate of orders into the restaurant (orders/second).

Kitchen & Preparation subsystem:

The first queuing node in the system is the Kitchen & Preparation subsystem, so that λ is in fact the rate of order arrivals to this node. μ_{KP} is the service rate, t_{KP} the service time and ω_{KP} the waiting time of an order.

The Kitchen & Preparation subsystem processes one order at a time. Therefore, its behavior will be defined by its exit process: the time an order spends in the server.

It is clear that t_{KP} is not representative of the real service time of an order during the kitchen and preparation processes. However, since the simulated system allows for only one order at a time, the waiting time in the queue generated before the server is part of this real time.

When an order exits the K&P subsystem, it is considered to be ready for delivery.

- **Delivery – Take away:**

Since there can be a varying number of delivery guys working on a service, we will study each delivery guy as a different server, making the total node as N equal servers – we are obviating the human factor, obviously there are faster and slower ones.

Two different models have been proposed:

1) On one hand, we have used an empirical model using an APP called "Hellotracks". This APP holds a log of the GPS positions a delivery guy's mobile phone has been to. A file with these positions, associated with a time stamp can be extracted from this APP. Processing this file with Matlab – taking the time between the mobile phone leaves the shop coordinates until it gets back to it - we get samples of the service time for delivery. We would need to add an offset for the time that it takes a delivery guy to choose the next order and the few actions that must be done in the shop between orders.

2) On the other hand, we have developed an analytical model. Using a tool of the ERP system, we can have 2D histograms with the number of orders in each area of influence of a restaurant for a given period.
We have associated this histogram to the probability of having an order in this delivery area.
In order to know the route time to an order in a bin, we will use Google Maps to know the average time and a Normal Distribution with a given standard deviation inside that bin.
We will assume an adjustment factor, which will be variable, of time reduction due to the fact that delivery guys work on motorbikes and have shorter route times than cars, which is the time Google Maps inform.

   Finally, we will assume three different offset times:

   1) The time it takes a delivery since it arrives at the client's house until it can ride the motorbike again.
   2) The time it takes a delivery guy since he arrives in the shop from an order and can start looking at new orders (certain action that must be made on the Front Office Software).
   3) The time it takes a delivery guy since an order is ready until he is effectively on the motorbike.

6. Data acquisition

6.1. The arrival process

In order to find the statistics of the arrival process, we need to take data of the time between orders entering the system. This is relatively easy since all of the orders entered in it have a time stamp. We will take all the times stamps of orders at Friday Nights (20:00-23:30) in the period January, 2nd 2015 to May, 1st 2015, and from it calculate the time between orders. The data used can be seen in **Annex 1**.

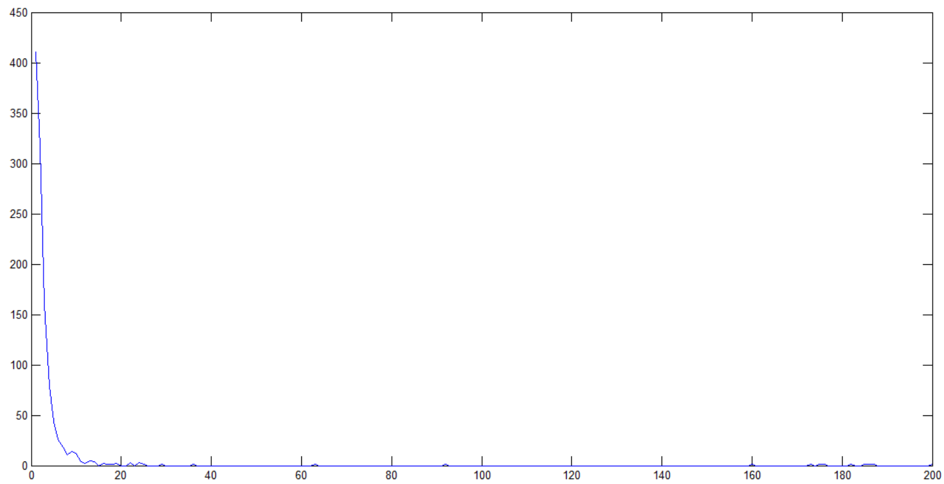

Figure 7. The arrival process

We have imported the data "Second between orders" from the Annex 1 into Matlab and created a histogram with 200 bins of the time between orders.

It can be easily seen that it follows an exponentially decreasing shape. It follows a Markovian distribution, which can be defined by one parameter: the order rate, λ.

We cannot assume that the order rate is constant during the whole shift, since there are very clear peak hours, as seen in the figure below:

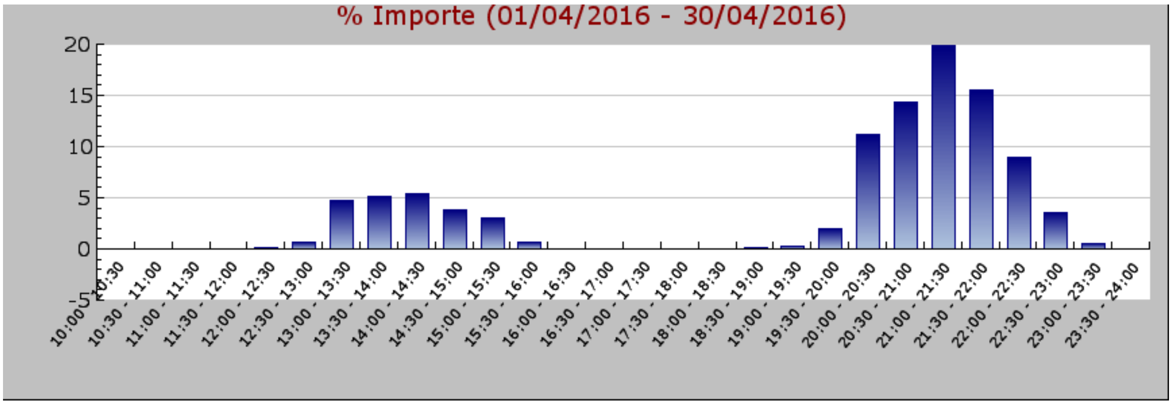

Figure 8. % of Revenue per hours

Therefore, we will model it using time frames of 30 minutes, each defined by its own order rate (λ); the next table shows the average time since the last order in each of the time frames, which is the inverse to λ.

As seen in the state of the art, when an arrival process follows a Poisson distribution, the probability of having an arrival in a Unitarian period of time follows the formula

$$P(x, \lambda) = \frac{\lambda^x * e^{-\lambda}}{x!}$$

Since λ takes such small values, $e^{-\lambda} \cong 1$, so:

$$P(1, \lambda) = \frac{\lambda^1 * 1}{1!} = \lambda$$

Time Frame Start	Time Frame End	Avg Time since last order	Λ	P(1,λ)
20:00	20:30	255	3,92E-03	3,92E-03
20:30	21:00	135	7,41E-03	7,41E-03
21:00	21:30	110	9,09E-03	9,09E-03
21:30	22:00	149	6,71E-03	6,71E-03
22:00	22:30	202	4,95E-03	4,95E-03
22:30	23:00	324	3,09E-03	3,09E-03
23:00	23:30	897	1,11E-03	1,11E-03

Figure 9. Arrivals model

6.2. The kitchen & Order Preparation

In order to model the black box behavior, we will take data of the exit process.

For this purpose, a field worker has spent 10 Friday Nights from March, 4th until May, 6th gathering data about it. We have taken a time stamp every order was ready for delivery or take-away, being that time when it has both come out of the kitchen, packed and put in the desk for ready orders.

In order that the time between ready orders can be assumed to represent the service time of these orders, an utmost condition must be fulfilled: the kitchen must be full at all times, this is guaranteed since data was taken in bursts, data acquisition was stopped when there were no pending tickets in the kitchen and resumed afterwards.

We have aggregated this data and pictured a histogram of it in figure 10.

Unluckily, such exit process does not apparently follow any known distribution and therefore, unlike previously done with the arrival process, we won't be able to use any known formula.

Therefore, we will need to develop an empirical model using the data measured. We will assign each bin a probability P_{bin} from the data histogram. For every order inside the K&P subsystem, a bin will be assigned with that probability and we will then uniformly distribute the service time inside that bin.

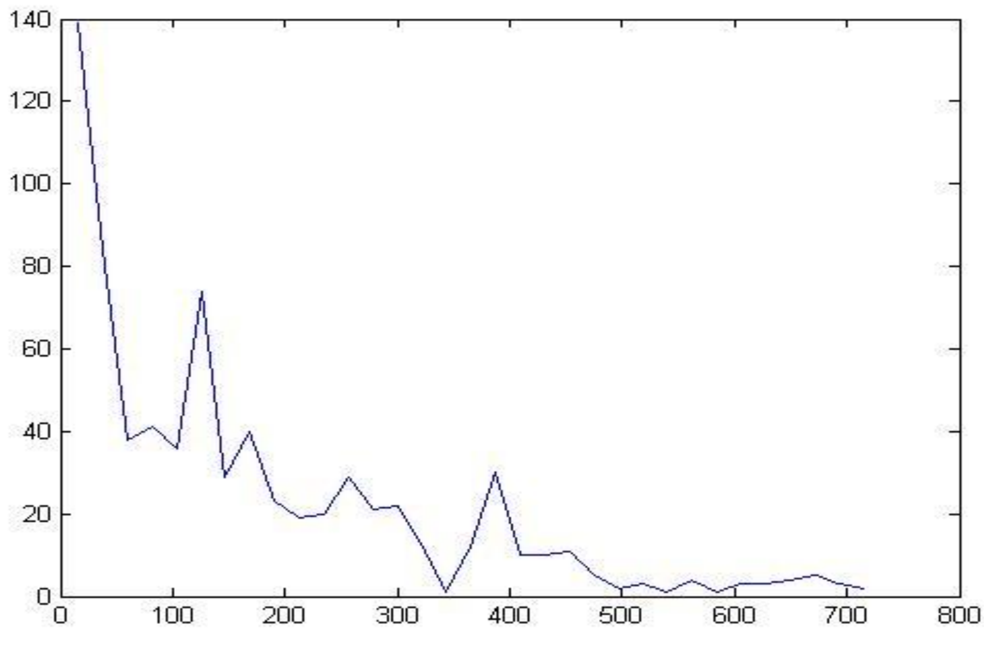

Figure 10. Seconds between orders ready for delivery or take-away histogram

A problem arises from such model, since we had seen that the service rate of the kitchen depended on the arrival rate due to the simultaneity allowed in the hot dishes area (there are no orders that are actually cooked in such little time as 30 seconds but they come out of the kitchen at this rate when the kitchen has a long queue), orders at the beginning of the shift are very likely to be assigned a very short service time which does not represent reality.

To solve this problem, we have decided to implement a new condition in the simulator. A minimum total waiting time at the K&P that will ensure that when an order enters the system and can be started to be cooked straight away, its total time at K&P system will look more like the real cooking time of an order. This waiting time will be extracted from the histogram, assuming that the third peak represents such orders.

6.3. Delivery

The last service node of the system will only affect certain orders, those which the clients choose want delivered at home and not to take-away.

Data of such a ratio is easily extracted from the order database; its statistics can be seen in Annex 2. A histogram of the delivery ratio is shown in the figure below of all the Friday Nights of the period January, 2nd 2015 to May, 1st 2015.

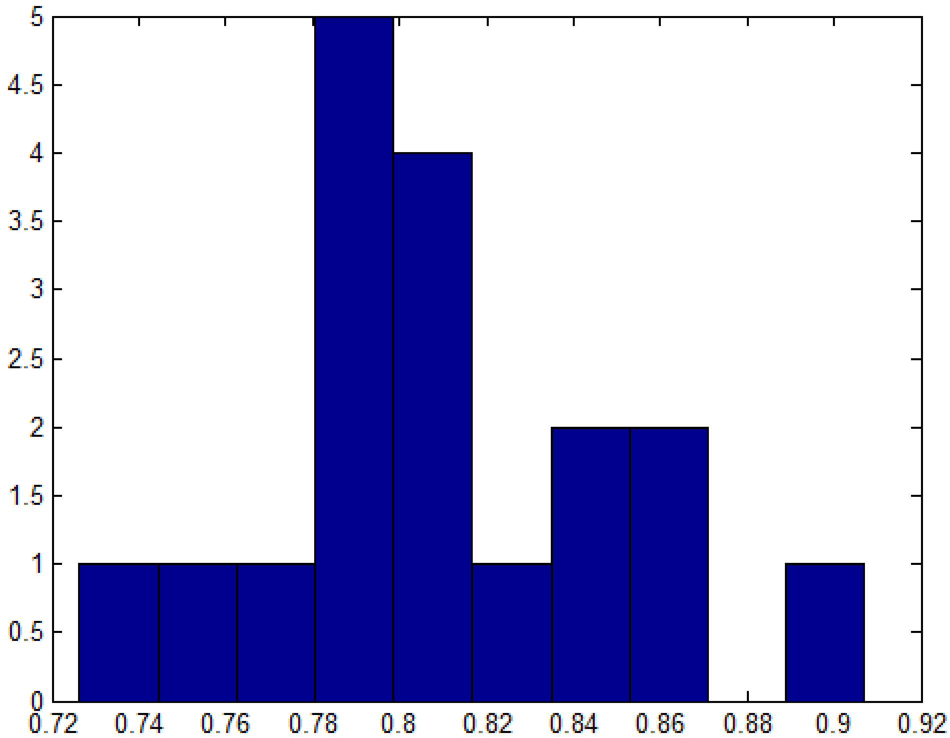

Figure 11. Delivery Ratio Histogram

On one hand, we have taken data from the ERP so as to be able to know which the probability of having an order in each delivery area is. A 2-D histogram of such data can be seen in Figure 12.

On the other hand, data from Google Maps has been taken to know the delivery time to the center of each area.

With this data and the model explained previously, the mean service time per delivery area is depicted in Figure 13. In this case, the adjustment factor was of 0.9

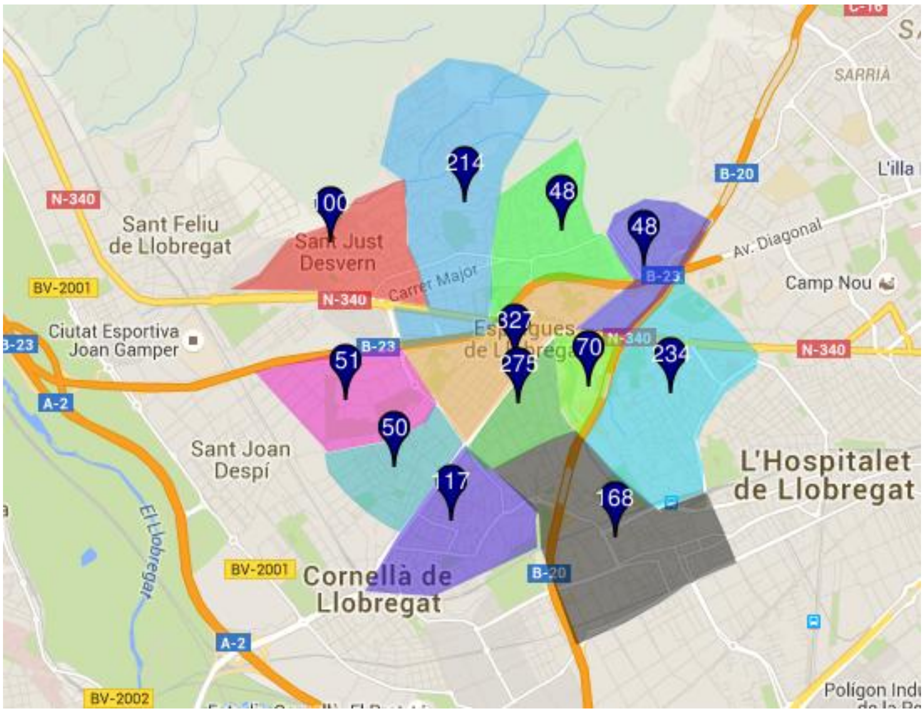

Figure 12. 2D-Histogram of orders per area

Zone	# orders	% orders	Central Adress	Route Time to Center	Client Offset	Store Offset 1	Store Offset 2	Service Time
1	275	16,16%	41.3698, 2.08545	180	120	60	60	564
2	70	4,11%	41.37385, 2.09215	300	120	60	60	780
3	327	19,21%	41.37246, 2.07957	300	120	60	60	780
4	51	3,00%	41.37212, 2.07258	360	120	60	60	888
5	50	2,94%	41.36684, 2.07390	300	120	60	60	780
6	117	6,87%	41.36194, 2.07890	360	120	60	60	888
7	168	9,87%	41.36076, 2.09952	480	120	60	60	1104
8	234	13,75%	41.36949, 2.10064	420	120	60	60	996
9	48	2,82%	41.38066, 2.09275	300	120	60	60	780
10	48	2,82%	41.38565, 2.08528	420	120	60	60	996
11	214	12,57%	41.38507, 2.07701	480	120	60	60	1104
12	100	5,88%	41.37981, 2.06975	420	120	60	60	996

Figure 13. Mean delivery service time per area

As for the model based on the "Hellotracks" APP, we have decided to leave it for further implementations, since some of the simplifications we have made go against the actual data that was taken with this method – basically the fact that a delivery guy takes one order per trip.
We can see, however, an example of the GPS data extracted from the APP "Hellotracks" in Annex 3.

7. Results analysis

Having programmed a simulator with Matlab (you can find the code of all of its functions in Annex 4), we will perform three differentiated kind of results analysis.

7.1. Simulations based on a random shift

The simulator is capable of generating random order arrivals based on historical statistics. As previously explained, using different order rates (λ) in 30 minute time frames the model generates a vector of arrivals based on a Poisson distribution.

We have simulated a shift from 20:00h to 23:30h based on these statistics. A total number of 68 orders arrived during the shift, of which 6 were take-away orders and 62 had to be delivered to the customer. They provided a total revenue to the business of 1768€.

The empirical cumulative distribution function of the waiting time for the orders is shown below:

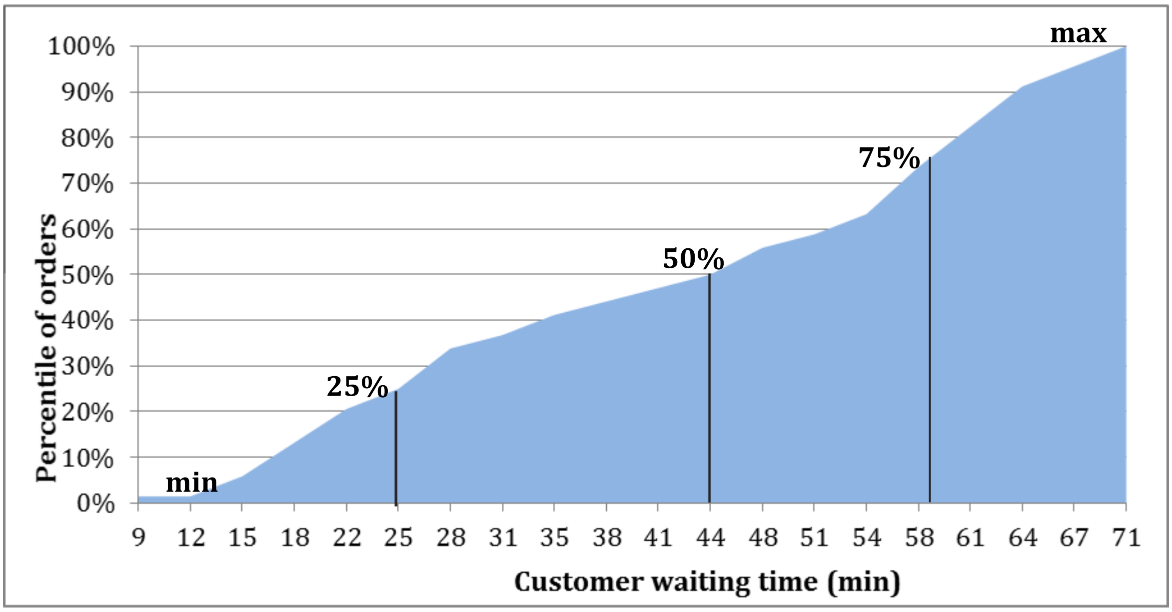

Figure 14. Empirical cumulative distribution function of the total customer waiting time.

In the previous graph we can see how much a customer is waiting to get his order as a percentile of all orders. The median waiting time in this simulation is 44 minutes, i.e. 50% of the orders are served in less than this time and the other 50% with more time.

An order is considered to have been served with bad quality when the customer has to wait more than 60 minutes to receive it. We can see that in this simulation this happens for almost 20% of the orders.

To better understand what is really going on in the system, we have plotted how the total time a customer has to wait depends on the different processes of the restaurant. Order numbers are ordered by arrival time (i.e. order #1 is the first to arrive and #68 the last).

As we already expected, the vast majority of the waiting time is driven by the time the order spends in the kitchen. It can also be observed that the time of delivery is independent from the arrival time of the order, since it is exclusively related to the client's location and completely independent of the system congestion.

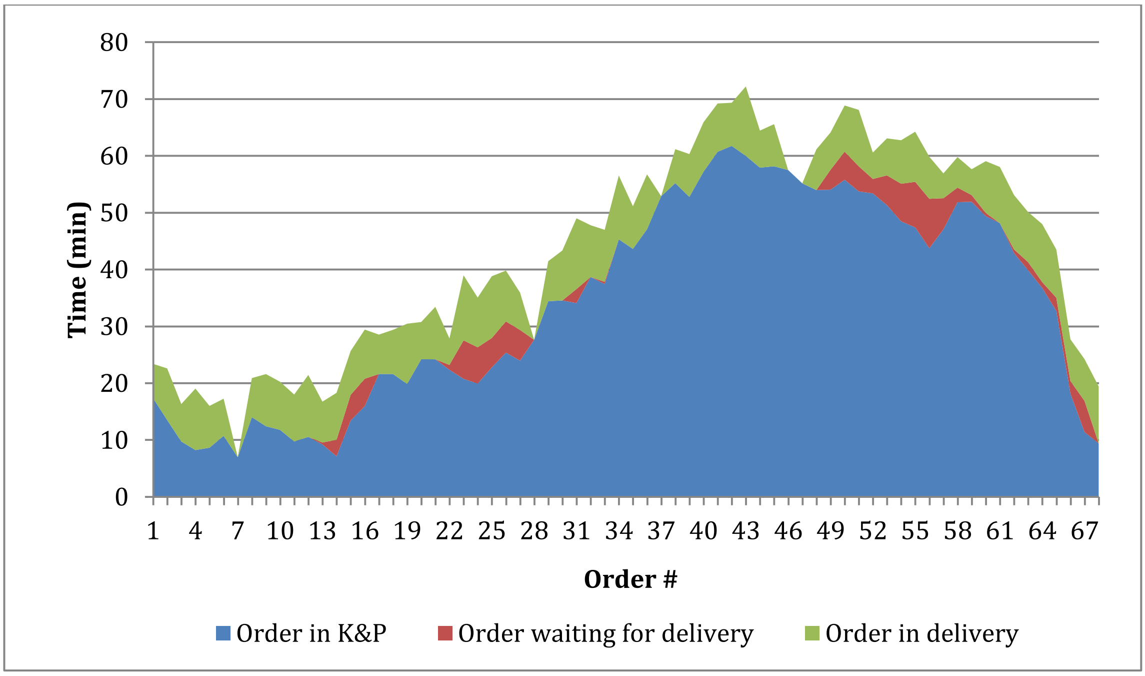

Figure 15. Total customer service time divided between the time spent in the Kitchen and preparation subsystem, waiting for delivery and during delivery (orders with no delivery time are those which are picked up by the client at the restaurant)

We also can see that some of the orders with a higher service time (e.g. orders #48 to #58) are waiting for delivery guys after being cooked, which increments the total waiting time for the customer. However, although adding some extra delivery workers during the period these orders arrive might help for these orders, the maximum waiting time would still remain the same (in order #42).

The total number of orders in the kitchen & Preparation subsystem over the whole shift is plotted below:

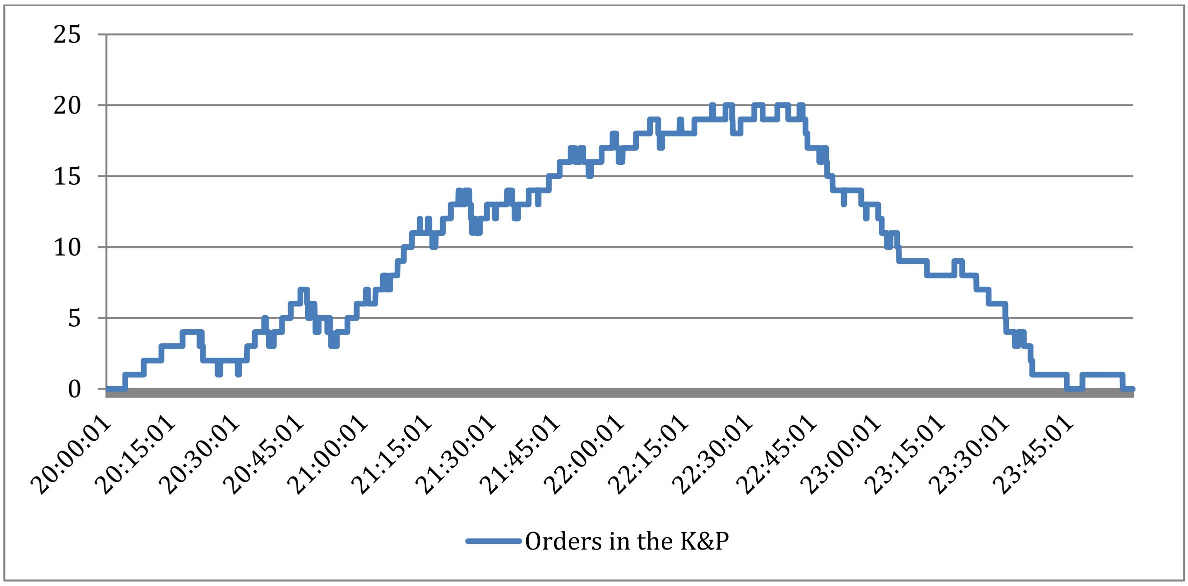

Figure 16. Total number of orders present in the K&P subsystem over the whole shift.

The 21:30-22:30h is the busiest period for the kitchen over the whole shift. The reason for this is that the 30 min time frame that registers the higher arrival is the 21:00-21:30h, generating a big workflow for the kitchen in the successive period.

One of the most useful applications of the prototype created is to analyze the needs of delivery workers over the whole shift to distribute these resources as efficiently as possible. This is why we have provided as an output of the prototype the number of unoccupied delivery workers over the shift. Although not being directly inverse, the number of free delivery workers is obviously inversely proportional to the number of orders waiting to be delivered, as can be clearly seen in figure 17.

Looking more closely to this graph, it is easily seen that there are some time periods during the shift where the number of free delivery workers is very high, especially at the beginning of the shift. As a rule of thumb, whenever Figure 17 shows a time frame in which the number of free delivery workers is 1 or above and there are no orders waiting for delivery, the system is over dimensioned in terms of delivery workers. Therefore, from this simulation we can conclude that:

- A number of delivery workers that vary over the shift is both cost efficient and easy to implement (in the sense that it is easy to differentiate the needs and dimensioning required for every shift)
- The number of delivery workers can and should be reduced during the 20:00-21:00h period – mostly due to the lower arrival rate at the beginning of the shift- and during the 22:00-23:00h period.
- In fact, this adjustment could come as a re-allocation of the resources in a more efficient way. For instance, the reduction of workers commented previously could be followed by an increase in the 21:00-22:00h and 23:00-23:45 periods, in which prepared orders have to wait for delivery workers to be free, increasing the customer waiting time and thus reducing quality. An increase in workers in these periods, especially in the later one, could drive significantly lower service times (as seen in figure 15).

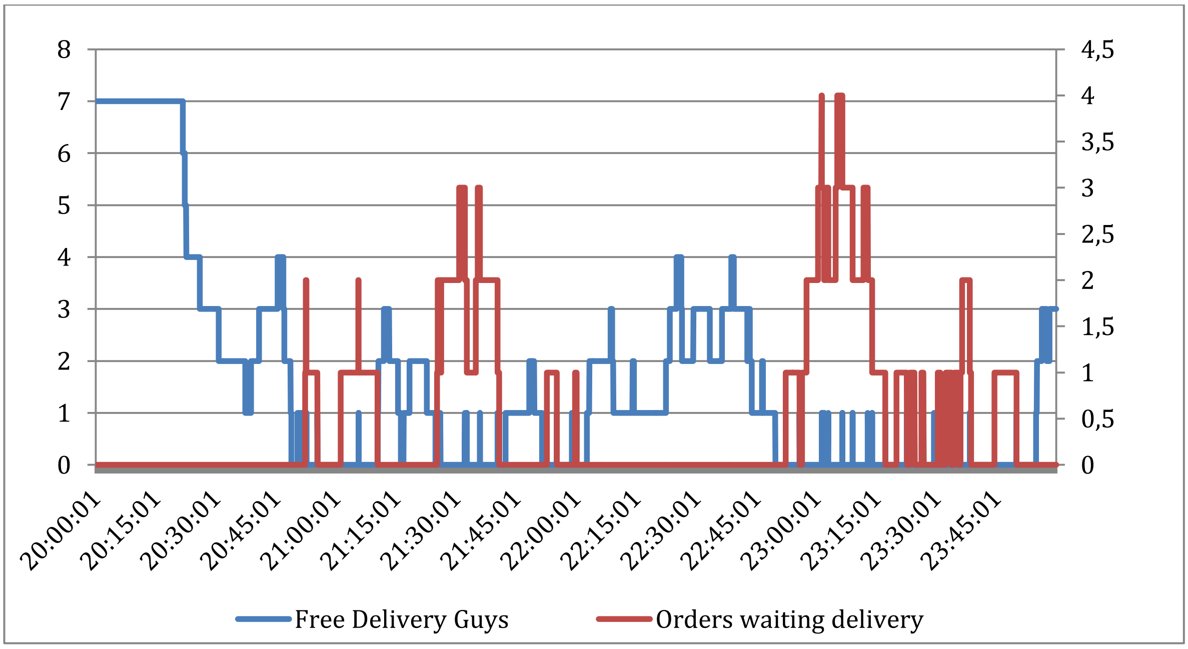

Figure 17. Left axis: Number of free delivery guys (i.e. waiting for more orders to deliver). Right axis: Number of prepared orders waiting for a delivery guy to process them.

Finally, another interesting result that can be obtained from this simulation is to clearly visualize in which service node the orders that are being processed by the restaurant are at every moment. It is no surprise that the closer to end of the shift, the bigger the percentage of orders that are in delivery (which will reach 100% after the last order is finished cooking).

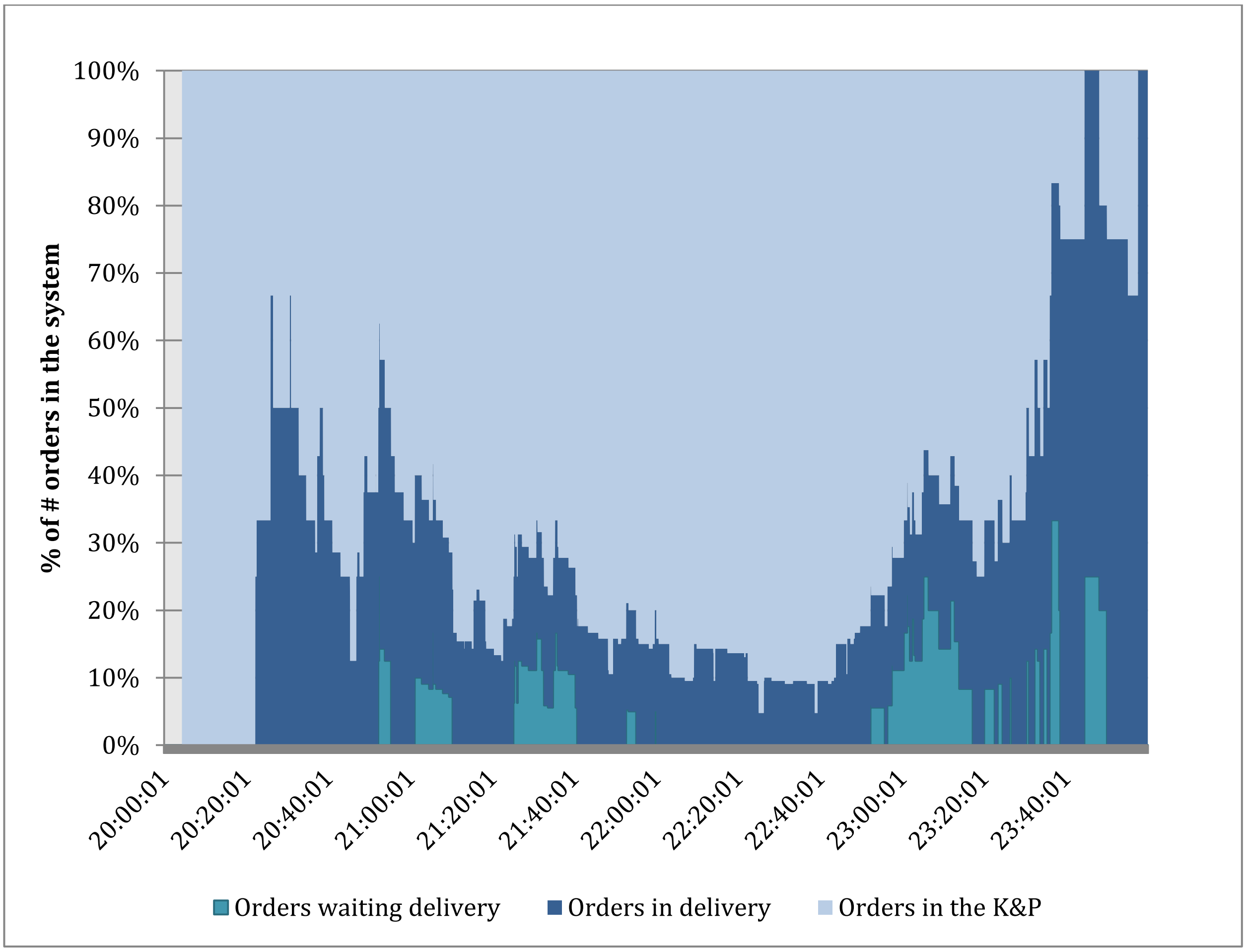

Figure 18. Distribution of the number of orders in the system between the K&P, waiting for delivery and in delivery.

7.2. Sensibilities

In order to depict the sort of general analysis about quality of services and resources variables that we can perform with the simulator, we have done three analyses.

1) We have studied how the empirical cumulative distribution function (ECDF) of total customer waiting time changes when shifting the delivery guys available at the shift. We have done this for several standard simulations (no stressed λ, 15% take-away ratio, traffic factor=0 and minimum waiting time in the kitchen queue of 360 seconds). The ECDF for delivery guys shifting from 5 to 13 can be seen in **Figure 19** below

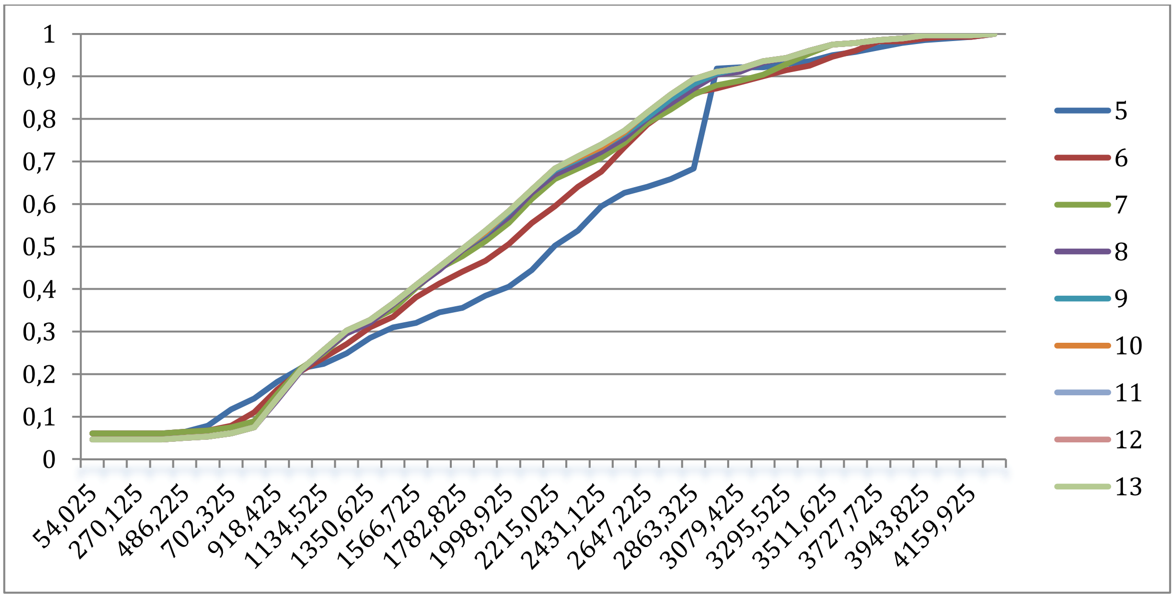

Figure 19. Empirical cumulative distribution function of the total customer waiting time

We can see how, despite adding new delivery guys has a very clear return on investment for the first increases (from 5 delivery guys until 7), there is a point where adding new delivery guys has no clear effect on the ECDF.
We can say from this Figure that the optimal amount of delivery guys would be around 7-8, which is the number fixed for this business unit.
We have been able to confirm that a decision that had been made through experience.

2) We have also studied how would affect to the quality of service if we had an unexpected work peak in a shift or the business unit had an increase in revenue. We have done this looking at the average service time for delivery orders when increasing λ in all the hourly frames by a certain factor and running several simulations under such conditions.

The result of such study can be seen in Figure 20, below.

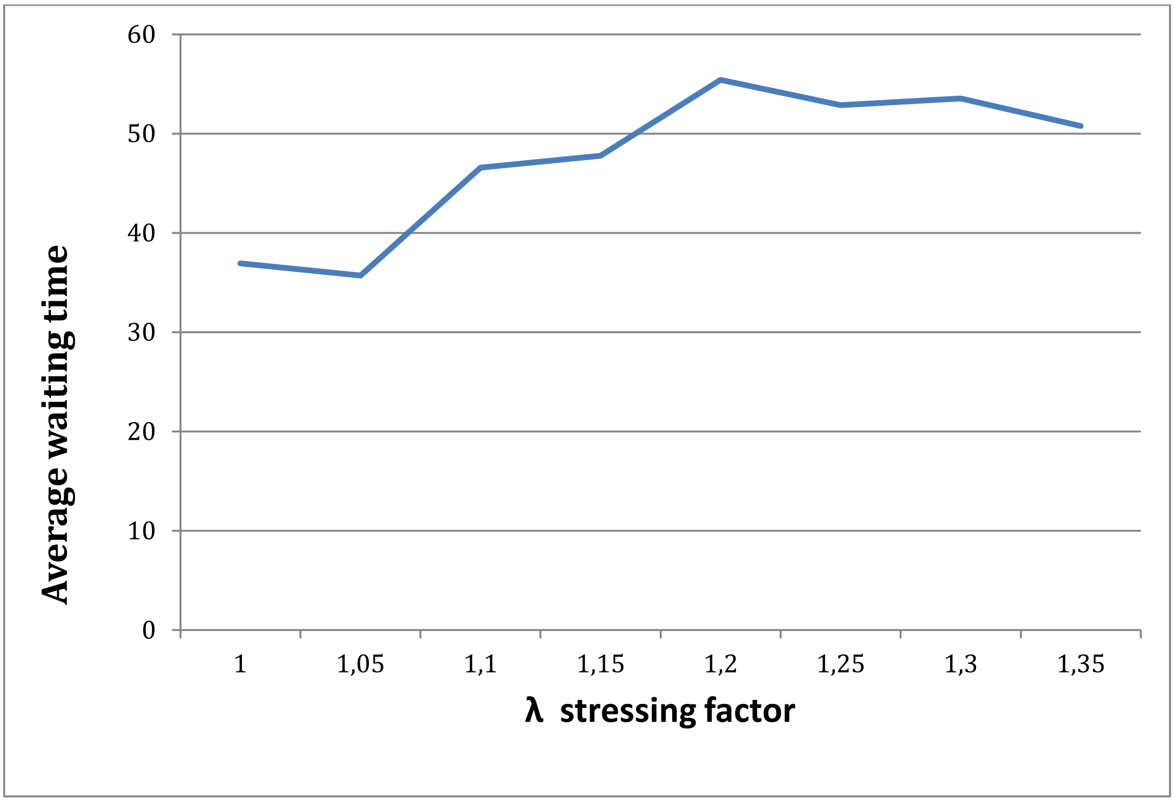

Figure 20. Average service time for orders when applying λ a stress factor

We can see how, given an important increase in revenue, the average waiting time for clients would be completely out of the company's standards, so we would need to readapt our resource deployment.

3) Finally, the last sensibility study we have performed consists on studying the exact effect of one of the strategic targets of the company: improving our take-away ratio, since take-away orders have higher margins. We have done this by running several simulations on different take-away ratio conditions, with the rest of the variables fixed.
The variable we have looked at is the % of unoccupied delivery guys during the whole shift. We believe that the average unoccupied ratio should always be between 25-30% because this is the only way to face work peaks without

affecting the quality of service. In the figure below, we can see how, as the take-away ratio increases, the unoccupied ratio does so. This is a source for saving a lot of resources without affecting quality of service. We should try and look for ways to incentive customers to come and pick their orders.

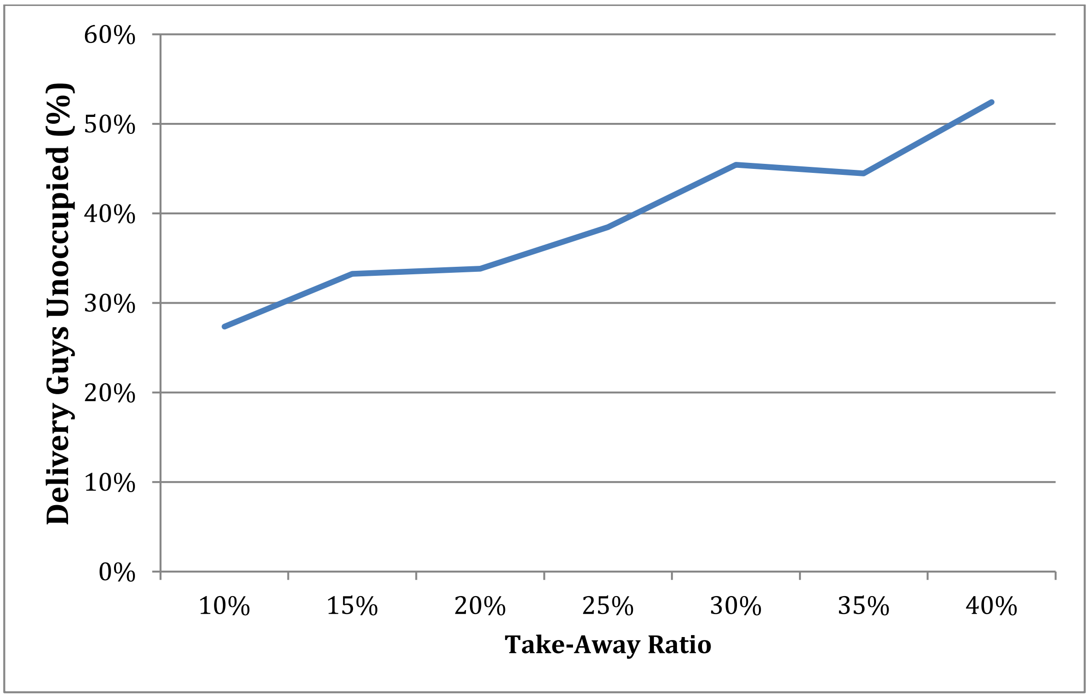

Figure 21. Percentage of unoccupied delivery guys against the Take-Away Ratio

8. Budget

Two different stages of the project implementation must be taken into consideration when analyzing its costs.

On one hand, for the study and characterization of the system and the development of the first prototype, the costs assumed will be mainly man hours, both direct costs – man hours that have actually been paid to existing workers or external people – and indirect cost- the hours I have myself committed to this project.

We will assume different man hour costs, depending on the functions being developed:

- Coordinator: 30€/h
    - The coordinator will coordinate all aspects of the project: including planning, developing the project documentation, the data taken and the software developed. The estimated dedication of such a position will be of 40 hours.
- Analyst: 15€/h
    - The analyst is the person in charge of exploiting the data from the ERP software and that obtained from the field worker. The estimated dedication of such position will be of 5 hours.
- Programmer: 25€/h
    - The programmer of this project will be the one responsible for implementing the proposed solutions with MATLAB. The estimated dedication of such position will be of 10 hours.
- Field worker: 7,5€/h
    - A field worker will be needed in order to take all the empirical data that cannot be registered with the existing ERP system. This will be done throughout 10 Friday Night shifts (20:00-23:30h).
    - The data taken from the delivery guys would carry no cost since it will be done during their normal shifts. The company has, however, offered a 2€/h plus (2,66€/h cost) to those delivery guys willing to cooperate with the project. This will be done in 5 3-hour shifts with a total of 6 delivery guys in the Esplugues shop.

Further development of the project, that is, actual integration in the ERP software, will require of an inside project coordinator and the company that owns the software. The duration of such an implementation will be of around 8 weeks, since it will affect the front-end and is a critical part for the company, no bugs can be allowed. The different costs of this part of the project are the following.

- Coordinator: 30€/h
    - A 2 hours weekly meeting will be scheduled during the duration of the project.
- ICSSolution: 2000€

The following table summarizes all the costs that need to be undertaken in the different stages:

	Unit Cost	Units	Total Cost
		Stage 1	
Coordinator	30,00 €	40	1.200,00 €
Analyst	15,00 €	5	75,00 €
Programmer	25,00 €	10	250,00 €
Field Worker	7,50 €	35	262,50 €
Delivery Guy	2,66 €	90	239,40 €
		Subtotal:	**2.026,90 €**
		Stage 2	
ICSSolution	2000 €	1	2.000,00 €
Coordinator	30 €	16	480,00 €
		Subtotal:	**2.480,00 €**
TOTAL:			**4.506,90 €**

Figure 22. Budget

9. Conclusions and Further Developments

9.1. Conclusions

From the results obtained with the simulations, the project itself and the personal experience of what a Telemaki shift is, we can extract a series of conclusions:

9.1.1. Project-related conclusions

We can overall conclude that we have developed a first prototype of what can be a very useful tool for delivery restaurants.

Despite the fact that it is a first prototype, it is already giving results that are consistent with the reality, both when simulating a random shift or a shift that comes from actual data.

9.1.2. Business-related conclusions

As for the business analysis that is intended with the final project, we can assure that with this first prototype we have been able to analytically corroborate things we have been doing by experience and confirm business strategic targets that could save costs.

We believe, then, that it is worth investing in further developing the system to be able to variabilize as many resources as possible to have deeper understanding of how everything works and by this, optimizing each and every of the factors, understanding their key relationship with the quality variables.

9.2. Further Developments

All the results and conclusions will be presented to the members of the board and in case that they are considered to be valuable for the business, an investment will be made in order to develop a complete system that will allow us to have an integral vision of the relationship between all the resources deployed at a business unit with the quality of service and provide us with greater accuracy and reliability.

The main lines that should be followed in order to further improve the system are the following:

1) Deepen in the kitchen system, at the dish level in cold area and simulating the way a hot cook works.
2) When two orders of the same delivery area are ready or close to be ready, the delivery guy waits and takes them both.

3) Take jockeying into consideration: people that are waiting for a delivery at home decide to come to the store to take it away.
4) Since kitchen output depends on input, the output pdf should depend on time as well.
5) Study how the average ticket price fluctuates with time.
6) Study how the average ticket price depends on delivery/take-away.
7) Try to simulate the fact that incidences occur with the service and that some orders need a second trip in order to fulfill it right.
8) Study how delivery area concentration fluctuates with time and day.
9) Model different starting working hours for delivery guys.
10) Study how the delivery/take-away ratio fluctuates with time.
11) The simulated model should allow for two disciplines concerning how the queue of this node is modeled:

    a. *FCFS discipline*: orders are served in the same sequence they are originated

    b. *Priority for take away orders*: when a take away order arrives, it automatically advances all the delivery orders in the Kitchen & Preparation queue. If there are other take away orders in the queue, it is set just behind the last one to have arrived. In case there aren't, the order is set in the first position of the queue and will enter the server once it is freed. There is no exclusion in this discipline, i.e. the order will always have to wait for the server to be free in order to occupy it.

12) Take data in all the business sites.
13) Integrate it with the front-end software, so that whenever an order enters the system, a waiting time is assigned to it.

10. Bibliography

[1] Queuing Theory and its Applications. A personal View – János Sztrik. University of Debrecen. Factulty of Informatics, Hungary.

[2] Telemaki: MOS (Manual Operativo de Sistemas) – Telemaki Tienda SL

[3] Queueing Systems. Volume 1: Theory

Annex 1: Example of Time Between arrivals data

Year	Month	Day	Date	Shift	Time	Time between orders	Seconds
2015	1	5	02/01/2015	2	19:49:12	0:00:00	
2015	1	5	02/01/2015	2	20:00:30	0:11:18	678
2015	1	5	02/01/2015	2	20:03:07	0:02:37	157
2015	1	5	02/01/2015	2	20:05:47	0:02:40	160
2015	1	5	02/01/2015	2	20:07:48	0:02:01	121
2015	1	5	02/01/2015	2	20:08:54	0:01:06	66
2015	1	5	02/01/2015	2	20:20:27	0:11:33	693
2015	1	5	02/01/2015	2	20:24:40	0:04:13	253
2015	1	5	02/01/2015	2	20:26:03	0:01:23	83
2015	1	5	02/01/2015	2	20:27:09	0:01:06	66
2015	1	5	02/01/2015	2	20:29:47	0:02:38	158
2015	1	5	02/01/2015	2	20:34:44	0:04:57	297
2015	1	5	02/01/2015	2	20:35:53	0:01:09	69
2015	1	5	02/01/2015	2	20:36:48	0:00:55	55
2015	1	5	02/01/2015	2	20:39:40	0:02:52	172
2015	1	5	02/01/2015	2	20:39:44	0:00:04	4
2015	1	5	02/01/2015	2	20:43:24	0:03:40	220
2015	1	5	02/01/2015	2	20:45:51	0:02:27	147
2015	1	5	02/01/2015	2	20:45:52	0:00:01	1
2015	1	5	02/01/2015	2	20:47:47	0:01:55	115
2015	1	5	02/01/2015	2	20:53:12	0:05:25	325
2015	1	5	02/01/2015	2	20:56:19	0:03:07	187
2015	1	5	02/01/2015	2	20:56:41	0:00:22	22
2015	1	5	02/01/2015	2	20:57:36	0:00:55	55
2015	1	5	02/01/2015	2	20:59:13	0:01:37	97
2015	1	5	02/01/2015	2	21:00:27	0:01:14	74
2015	1	5	02/01/2015	2	21:12:28	0:12:01	721
2015	1	5	02/01/2015	2	21:16:40	0:04:12	252
2015	1	5	02/01/2015	2	21:19:33	0:02:53	173
2015	1	5	02/01/2015	2	21:19:44	0:00:11	11
2015	1	5	02/01/2015	2	21:23:25	0:03:41	221
2015	1	5	02/01/2015	2	21:24:56	0:01:31	91
2015	1	5	02/01/2015	2	21:25:54	0:00:58	58
2015	1	5	02/01/2015	2	21:28:00	0:02:06	126
2015	1	5	02/01/2015	2	21:28:38	0:00:38	38
2015	1	5	02/01/2015	2	21:29:26	0:00:48	48
2015	1	5	02/01/2015	2	21:31:30	0:02:04	124
2015	1	5	02/01/2015	2	21:32:02	0:00:32	32
2015	1	5	02/01/2015	2	21:33:14	0:01:12	72
2015	1	5	02/01/2015	2	21:33:45	0:00:31	31
2015	1	5	02/01/2015	2	21:34:12	0:00:27	27

2015	1	5	02/01/2015	2	21:36:12	0:02:00	120
2015	1	5	02/01/2015	2	21:37:08	0:00:56	56
2015	1	5	02/01/2015	2	21:38:09	0:01:01	61
2015	1	5	02/01/2015	2	21:41:17	0:03:08	188
2015	1	5	02/01/2015	2	21:43:23	0:02:06	126
2015	1	5	02/01/2015	2	21:47:19	0:03:56	236
2015	1	5	02/01/2015	2	21:49:16	0:01:57	117
2015	1	5	02/01/2015	2	21:51:07	0:01:51	111
2015	1	5	02/01/2015	2	21:51:40	0:00:33	33
2015	1	5	02/01/2015	2	21:53:07	0:01:27	87
2015	1	5	02/01/2015	2	21:54:11	0:01:04	64
2015	1	5	02/01/2015	2	21:57:50	0:03:39	219
2015	1	5	02/01/2015	2	22:04:08	0:06:18	378
2015	1	5	02/01/2015	2	22:07:44	0:03:36	216
2015	1	5	02/01/2015	2	22:09:43	0:01:59	119
2015	1	5	02/01/2015	2	22:11:48	0:02:05	125
2015	1	5	02/01/2015	2	22:17:02	0:05:14	314
2015	1	5	02/01/2015	2	22:21:02	0:04:00	240
2015	1	5	02/01/2015	2	22:21:41	0:00:39	39
2015	1	5	02/01/2015	2	22:23:59	0:02:18	138
2015	1	5	02/01/2015	2	22:31:22	0:07:23	443
2015	1	5	02/01/2015	2	22:32:11	0:00:49	49
2015	1	5	02/01/2015	2	22:45:37	0:13:26	806
2015	1	5	02/01/2015	2	22:46:26	0:00:49	49
2015	1	5	02/01/2015	2	22:54:00	0:07:34	454
2015	1	5	02/01/2015	2	23:05:59	0:11:59	719

Annex 2: Delivery /Take-Away Ratio samples

Dia	Domicilio	Recoger en Local	Total	% Domicilio	% Local	TOTAL
02/01/2015	1.304,33 €	353,02 €	1.657,35 €	78,70%	21,30%	100,00%
09/01/2015	1.343,14 €	339,98 €	1.683,12 €	79,80%	20,20%	100,00%
16/01/2015	1.019,80 €	157,70 €	1.177,50 €	86,61%	13,39%	100,00%
23/01/2015	1.567,71 €	251,16 €	1.818,87 €	86,19%	13,81%	100,00%
30/01/2015	1.831,08 €	187,62 €	2.018,70 €	90,71%	9,29%	100,00%
06/02/2015	1.654,13 €	465,78 €	2.119,91 €	78,03%	21,97%	100,00%
13/02/2015	1.656,03 €	442,90 €	2.098,93 €	78,90%	21,10%	100,00%
20/02/2015	1.436,97 €	366,14 €	1.803,11 €	79,69%	20,31%	100,00%
27/02/2015	1.623,01 €	509,36 €	2.132,37 €	76,11%	23,89%	100,00%
06/03/2015	1.409,54 €	334,15 €	1.743,69 €	80,84%	19,16%	100,00%
13/03/2015	1.450,61 €	277,89 €	1.728,51 €	83,92%	16,08%	100,00%
20/03/2015	1.481,94 €	364,80 €	1.846,74 €	80,25%	19,75%	100,00%
27/03/2015	958,12 €	360,82 €	1.318,93 €	72,64%	27,36%	100,00%
03/04/2015	1.553,51 €	338,63 €	1.892,14 €	82,10%	17,90%	100,00%
10/04/2015	1.814,17 €	328,28 €	2.142,44 €	84,68%	15,32%	100,00%
17/04/2015	1.534,03 €	352,20 €	1.886,23 €	81,33%	18,67%	100,00%
24/04/2015	1.347,86 €	353,09 €	1.700,94 €	79,24%	20,76%	100,00%
01/05/2015	1.132,40 €	276,35 €	1.408,75 €	80,38%	19,62%	100,00%
AVERAGE	**1.451,02 €**	**336,66 €**	**1.787,68 €**	**81,12%**	**18,88%**	**100,00%**
DESV TIP	242,21 €	87,29 €	277,29 €	4,17%	4,17%	0,00%

Annex 3: Example of GPS sampling using "Hellotracks"

Type	time	latitude	Longitude	altitude (m)	speed (km/h)
T	14:06:50	41301910000	2007570000	75.0	10.8
T	14:06:53	41302060000	2007110000	79.0	61.2
T	14:06:57	41302060000	2007260000	85.0	0.0
T	14:08:07	41302110000	2006990000	46.0	21.6
T	14:08:11	41302040000	2006850000	48.0	7.2
T	14:08:14	41301960000	2006960000	47.0	10.8
T	14:08:22	41302050000	2007050000	44.0	10.8
T	14:08:36	41301910000	2007240000	52.0	7.2
T	14:08:40	41301800000	2007260000	20.0	10.8
T	14:08:48	41301740000	2007100000	15.0	21.6
T	14:09:06	41301950000	2007400000	15.0	3.6
T	14:09:25	41301780000	2007560000	15.0	3.6
T	14:09:47	41301860000	2006920000	49.0	14.4
T	14:10:23	41301670000	2006570000	34.0	7.2
T	14:10:26	41301740000	2006650000	45.0	0.0
T	14:10:29	41301780000	2006810000	45.0	0.0
T	14:10:40	41301780000	2006970000	39.0	7.2
T	14:11:07	41301710000	2006910000	40.0	0.0
T	14:12:47	41301770000	2007050000	41.0	10.8
T	14:13:59	41301880000	2007000000	52.0	14.4
T	14:14:18	41301930000	2007360000	51.0	18.0
T	14:14:24	41301900000	2007130000	51.0	7.2
T	14:14:28	41301960000	2007310000	50.0	7.2
T	14:14:32	41301990000	2007630000	50.0	7.2
T	14:14:36	41301930000	2007380000	48.0	7.2
T	14:14:43	41301860000	2007510000	53.0	10.8
T	14:14:46	41301890000	2007300000	54.0	3.6
T	14:16:40	41301980000	2007240000	47.0	0.0
T	14:17:25	41302150000	2006960000	56.0	54.0
T	14:17:29	41302160000	2006660000	56.0	61.2
T	14:17:33	41301580000	2006510000	54.0	79.2
T	14:17:37	41301570000	2006740000	53.0	72.0
T	14:17:41	41301750000	2006890000	55.0	25.2
T	14:17:45	41301790000	2007030000	57.0	57.6
T	14:17:49	41301630000	2007230000	57.0	97.2
T	14:17:52	41301380000	2007400000	56.0	97.2
T	14:17:56	41301070000	2007710000	53.0	136.8
T	14:17:59	41300750000	2008000000	59.0	126.0
T	14:18:03	41300450000	2008210000	54.0	154.8
T	14:18:07	41300220000	2008440000	56.0	97.2
T	14:18:11	41300090000	2008480000	54.0	32.4
T	14:18:14	41299800000	2008810000	49.0	144.0

T	14:18:18	41299370000	2009160000	46.0	216.0
T	14:18:22	41298820000	2009650000	54.0	226.8
T	14:18:23	41298600000	2009800000	51.0	176.4
T	14:18:27	41298160000	2010250000	51.0	165.6
T	14:18:30	41297810000	2010520000	58.0	205.2
T	14:18:33	41297210000	2011020000	58.0	280.8
T	14:18:43	41296610000	2011490000	57.0	252.0
T	14:18:46	41296120000	2011830000	54.0	194.4
T	14:18:50	41295590000	2012240000	51.0	234.0
T	14:18:54	41295170000	2012520000	50.0	180.0
T	14:18:58	41294760000	2012800000	49.0	198.0
T	14:19:03	41294190000	2013170000	47.0	226.8
T	14:19:07	41293560000	2013640000	44.0	237.6
T	14:19:10	41292950000	2013950000	44.0	208.8
T	14:19:14	41292440000	2014270000	46.0	205.2
T	14:19:17	41291830000	2014540000	46.0	226.8
T	14:19:20	41291250000	2014410000	47.0	208.8
T	14:19:23	41290900000	2013910000	48.0	187.2
T	14:19:26	41290690000	2013370000	49.0	176.4
T	14:19:30	41290490000	2012760000	53.0	180.0
T	14:19:34	41290430000	2012060000	52.0	183.6
T	14:19:37	41290370000	2011560000	52.0	144.0
T	14:19:41	41290220000	2011070000	51.0	144.0
T	14:19:44	41290600000	2010760000	51.0	154.8
T	14:19:48	41291040000	2010400000	53.0	176.4
T	14:19:52	41291550000	2010490000	55.0	180.0
T	14:20:03	41291860000	2011050000	57.0	183.6
T	14:20:07	41292380000	2011180000	55.0	201.6
T	14:20:11	41292980000	2011360000	54.0	205.2
T	14:20:14	41293460000	2011780000	55.0	208.8
T	14:20:18	41293740000	2012270000	56.0	169.2
T	14:20:23	41294160000	2013170000	54.0	252.0
T	14:20:27	41294930000	2014090000	53.0	327.6
T	14:20:31	41295510000	2014980000	51.0	324.0
T	14:20:35	41296180000	2015940000	52.0	345.6
T	14:20:38	41296770000	2016950000	53.0	367.2
T	14:20:42	41297400000	2018010000	51.0	378.0
T	14:20:45	41298100000	2019120000	52.0	385.2
T	14:20:48	41298850000	2020160000	57.0	381.6
T	14:20:52	41299500000	2020980000	56.0	338.4
T	14:20:55	41300370000	2021840000	58.0	367.2
T	14:20:59	41301230000	2022730000	57.0	374.4
T	14:21:02	41302060000	2023580000	55.0	381.6
T	14:21:06	41302930000	2024430000	55.0	385.2
T	14:21:09	41303820000	2025300000	56.0	396.0
T	14:21:13	41304700000	2026150000	56.0	388.8
T	14:21:23	41305510000	2026890000	55.0	374.4

T	14:21:27	41306360000	2027590000	63.0	338.4
T	14:21:30	41307200000	2028260000	57.0	349.2
T	14:21:34	41308100000	2028860000	59.0	360.0
T	14:21:37	41308940000	2029480000	63.0	345.6
T	14:21:43	41309740000	2030120000	69.0	334.8
T	14:21:47	41310510000	2030730000	65.0	316.8
T	14:21:50	41311300000	2031360000	64.0	313.2
T	14:21:54	41312130000	2031950000	60.0	334.8
T	14:21:57	41312980000	2032610000	55.0	356.4
T	14:22:00	41313790000	2033220000	53.0	334.8
T	14:22:04	41314620000	2033880000	57.0	360.0
T	14:22:08	41315520000	2034580000	61.0	349.2
T	14:22:11	41316310000	2035310000	59.0	338.4
T	14:22:14	41317090000	2036030000	54.0	363.6
T	14:22:18	41317890000	2036870000	55.0	374.4
T	14:22:22	41318680000	2037740000	56.0	370.8
T	14:22:25	41319440000	2038700000	55.0	378.0
T	14:22:29	41320160000	2039660000	58.0	342.0
T	14:22:33	41320830000	2040550000	56.0	324.0
T	14:22:43	41321410000	2041320000	56.0	295.2
T	14:22:47	41321970000	2042120000	56.0	280.8
T	14:22:50	41322180000	2042590000	52.0	194.4
T	14:22:54	41322750000	2043310000	53.0	288.0
T	14:22:58	41323370000	2044160000	52.0	327.6
T	14:23:03	41324040000	2045100000	54.0	345.6
T	14:23:07	41324730000	2046060000	52.0	363.6
T	14:23:10	41325430000	2047050000	51.0	360.0
T	14:23:14	41326200000	2048100000	56.0	374.4
T	14:23:18	41326890000	2049220000	58.0	378.0
T	14:23:22	41327520000	2050270000	58.0	363.6
T	14:23:24	41328040000	2051160000	58.0	396.0
T	14:23:29	41328450000	2052870000	55.0	392.4
T	14:23:33	41329020000	2054080000	56.0	385.2
T	14:23:36	41329720000	2055190000	58.0	388.8
T	14:23:40	41330380000	2056140000	62.0	345.6
T	14:23:43	41331260000	2056170000	64.0	309.6
T	14:23:47	41332060000	2055800000	60.0	302.4
T	14:23:50	41332760000	2055460000	60.0	262.8
T	14:23:53	41333390000	2055000000	66.0	255.6
T	14:24:03	41334100000	2054520000	64.0	273.6
T	14:24:06	41334830000	2054080000	66.0	284.4
T	14:24:10	41335560000	2053730000	63.0	277.2
T	14:24:14	41336330000	2053270000	66.0	277.2
T	14:24:18	41337010000	2053010000	68.0	234.0
T	14:24:23	41337560000	2052640000	67.0	208.8
T	14:24:26	41338180000	2052330000	64.0	270.0
T	14:24:29	41339050000	2052020000	63.0	306.0

T	14:24:32	41339850000	2051680000	63.0	298.8
T	14:24:36	41340270000	2050930000	61.0	252.0
T	14:24:39	41340620000	2050250000	59.0	226.8
T	14:24:43	41341110000	2049900000	63.0	208.8
T	14:24:46	41341810000	2049880000	57.0	241.2
T	14:24:50	41342140000	2049210000	60.0	201.6
T	14:24:54	41342900000	2049520000	55.0	288.0
T	14:24:57	41343740000	2049970000	59.0	334.8
T	14:25:01	41344620000	2050400000	59.0	327.6
T	14:25:04	41345460000	2050790000	60.0	295.2
T	14:25:08	41346090000	2051100000	62.0	223.2
T	14:25:12	41346710000	2051450000	65.0	223.2
T	14:25:23	41347230000	2051730000	66.0	216.0
T	14:25:26	41347910000	2052100000	67.0	295.2
T	14:25:30	41348700000	2052500000	63.0	306.0
T	14:25:34	41349460000	2052860000	60.0	298.8
T	14:25:38	41350350000	2053380000	62.0	338.4
T	14:25:43	41351060000	2053950000	62.0	306.0
T	14:25:47	41351680000	2054620000	63.0	288.0
T	14:25:51	41352040000	2055280000	66.0	212.4
T	14:25:54	41352380000	2056000000	62.0	226.8
T	14:25:57	41352620000	2056730000	61.0	208.8
T	14:26:01	41352810000	2057380000	61.0	180.0
T	14:26:04	41352960000	2057990000	61.0	176.4
T	14:26:07	41352960000	2058430000	60.0	108.0
T	14:26:11	41352670000	2058930000	61.0	205.2
T	14:26:14	41352100000	2059040000	63.0	208.8
T	14:26:17	41351950000	2058280000	65.0	180.0
T	14:26:20	41352510000	2057570000	66.0	226.8
T	14:26:23	41353080000	2056550000	53.0	298.8
T	14:26:27	41353460000	2055880000	53.0	219.6
T	14:26:31	41353880000	2055260000	57.0	212.4
T	14:26:43	41354260000	2054620000	50.0	226.8
T	14:26:46	41354740000	2053990000	55.0	255.6
T	14:26:50	41355320000	2053350000	60.0	277.2
T	14:26:53	41355990000	2052720000	57.0	295.2
T	14:26:57	41356640000	2052190000	59.0	298.8
T	14:27:03	41357360000	2051740000	63.0	306.0
T	14:27:07	41358140000	2051280000	64.0	306.0
T	14:27:10	41359220000	2051110000	71.0	385.2
T	14:27:14	41360270000	2051000000	66.0	356.4
T	14:27:17	41361240000	2050970000	66.0	352.8
T	14:27:21	41362260000	2050860000	67.0	367.2
T	14:27:24	41363250000	2050630000	67.0	360.0
T	14:27:28	41364110000	2050200000	67.0	349.2
T	14:27:32	41365060000	2049710000	67.0	363.6
T	14:27:35	41365910000	2049070000	67.0	327.6

T	14:27:38	41366690000	2048340000 66.0	345.6
T	14:27:41	41367400000	2047450000 67.0	342.0
T	14:27:45	41368060000	2046470000 70.0	352.8
T	14:27:48	41368540000	2045250000 69.0	360.0
T	14:27:52	41368920000	2044290000 70.0	259.2
T	14:28:03	41369270000	2043290000 71.0	320.4
T	14:28:06	41369670000	2042190000 72.0	334.8
T	14:28:10	41370100000	2041090000 80.0	324.0
T	14:28:13	41370490000	2040040000 79.0	316.8
T	14:28:16	41371030000	2039270000 81.0	291.6
T	14:28:23	41371870000	2039070000 82.0	291.6
T	14:28:27	41372610000	2039450000 80.0	306.0
T	14:28:28	41372970000	2039740000 79.0	306.0
T	14:28:33	41372850000	2041460000 70.0	334.8
T	14:28:36	41372800000	2042820000 71.0	360.0
T	14:28:40	41372830000	2044220000 66.0	374.4
T	14:28:44	41372810000	2045630000 66.0	388.8
T	14:28:47	41372730000	2047070000 69.0	392.4
T	14:28:51	41372680000	2048490000 69.0	385.2
T	14:28:54	41372600000	2049790000 69.0	345.6
T	14:28:58	41372680000	2050940000 67.0	302.4
T	14:29:02	41372840000	2051960000 72.0	295.2
T	14:29:05	41373030000	2053160000 76.0	320.4
T	14:29:09	41373280000	2054170000 75.0	259.2
T	14:29:12	41373470000	2055100000 82.0	266.4
T	14:29:22	41373550000	2055900000 87.0	237.6
T	14:29:26	41373860000	2057200000 90.0	309.6
T	14:29:29	41374140000	2058350000 83.0	338.4
T	14:29:32	41374540000	2059700000 89.0	360.0
T	14:29:35	41374810000	2061030000 92.0	370.8
T	14:29:42	41374940000	2062310000 95.0	367.2
T	14:29:45	41375070000	2063660000 93.0	360.0
T	14:29:48	41375180000	2064830000 100.0	316.8
T	14:29:52	41375320000	2066240000 106.0	381.6
T	14:29:56	41375460000	2067670000 113.0	378.0
T	14:29:59	41375500000	2069040000 116.0	363.6
T	14:30:03	41375550000	2070350000 121.0	345.6
T	14:30:06	41375590000	2071500000 128.0	284.4
T	14:30:10	41375580000	2072360000 133.0	212.4
T	14:30:13	41375580000	2073070000 133.0	187.2
T	14:30:17	41375350000	2073270000 125.0	118.8
T	14:30:20	41375520000	2073480000 128.0	108.0
T	14:30:24	41375780000	2073220000 128.0	133.2
T	14:30:28	41376300000	2072970000 130.0	226.8
T	14:30:31	41376830000	2072910000 130.0	180.0
T	14:30:42	41377270000	2072500000 129.0	201.6
T	14:30:45	41377870000	2072200000 129.0	226.8

T	14:30:49	41378530000	2072100000	132.0	248.4
T	14:30:52	41379250000	2072040000	136.0	244.8
T	14:30:55	41379030000	2072920000	141.0	169.2
T	14:31:02	41378930000	2073310000	120.0	64.8
T	14:31:25	41378920000	2073460000	140.0	28.8
T	14:31:28	41378720000	2073580000	137.0	68.4
T	14:31:32	41378190000	2073440000	145.0	151.2
T	14:31:36	41377880000	2073270000	144.0	115.2
T	14:31:39	41377700000	2073220000	143.0	79.2
T	14:31:47	41377720000	2073090000	145.0	10.8
T	14:31:51	41377650000	2073200000	150.0	7.2
T	14:31:58	41377670000	2073350000	153.0	0.0
T	14:34:01	41377430000	2073550000	99.0	3.6

Annex 4: Matlab Simulator

The code that runs the simulator can be found in the attached file MatlabSimulator.zip