Virtualization of Customer Flow and Corporate TV Management

Vítor Carvalho, Vítor Rodrigues, Paulo Ferreira
Instituto Superior Técnico
vitor.carvalho@ist.utl.pt, vitor.rodrigues@inm.pt, paulo.ferreira@inesc-id.pt

October 2012

Keywords: Mobility, Smartphone, Queue Management, Waiting Lines, Mobile Corporate TV, Profile Matching

Abstract: The queue management systems evolved until their current state, where they haven’t keep up with the new technological trends, providing nothing else but, a variety of services with different tickets and displays presenting information of each service queue, as well as some distraction to customers with multimedia content called Corporate TV. Benefiting from the fact that mobile phones are becoming increasingly used in the everyday life, and with the mass emergence of smartphones, it becomes possible to use its characteristics and its performance on the field queue management systems to request queue status through the customer’s smartphone, or even user specific information such as incoming turn notifications, instead of constantly checking the screens available. Another advantage about the use of smartphones in this field is that we can now entertain users with personalized video content.

This paper presents a solution to take advantage of these same benefits in current mobile devices, reducing and improving the waiting period, as well as the service care. It guarantees mobility with a customized service and with the possibility of multiple calls avoiding the risk of collision for the same user. Queue management algorithms are used to improve customer flow as well as avoid these collisions. It is described the proposed architecture to create the queue management system that supports the desired requirements. Finally it is presented the evaluation of the solution through performance and usability tests, as well as a conclusion where further development features for the future are pointed out.

1 INTRODUCTION

A major concern in customer flow systems is the call duration. With an efficient service the user is more satisfied, but we should not overlook the waiting time to be served, because it is much more important than the service time. As noted in [1], "Products are Consumed, services are experienced" therefore, we should try to reduce this time, or at least provide a wait as pleasant as possible.

When a customer goes to an organization with multiple services, he will get as many tickets as he needs from the different services we wants. However there is a recurring problem. There is the possibility of being called for two services at the same time. Besides the fact that we must always pay attention to the numbers of all the services of our interest, to try and understand which one will be called first.

Another problem of the current systems is ticket printing. With thermal paper tickets, those can not be recycled later. Besides the environment, there is still the cost associated with the printing of the tickets and the maintenance of their printers.

The tickets are usually called through simple displays or televisions indicating queue status and some more useful information, as well as multimedia content also called Corporate TV/Digital Signage. The problem of such systems is that all users are forced to watch the same content, which, in some cases, is not the appropriated.

The main purpose of the solution presented in this paper is to explore the technological improvement that have been noted in current queue management systems and thus integrate it with a mobile device, in order to make these systems more user-friendly and personal. Through the use of smartphones the customers own mobile device will notify him of its turn in a particular organization. With this type of integration, there’s the possibility to present a lot of information to the user, such as waiting time, number of tickets ahead, estimated time of service, etc.. It can also be provided personalized Corporate TV content to the user through the custom user profile provided.

Finally, another goal of this solution is to solve the problem of service collision of two different tickets of the same customer. With the presented solution, there is no need to manage the tickets or the time it will take each one to be called, because the system takes care of this management for him.

The solution proposed in this paper is a queue management system where a customer can get multiple tickets for multiple services simultaneously, without having the worry of constantly checking which service will be called first and without collisions in
those calls.

It also aims to ease the ticket transfer to a mobile device without printing it on paper. Thus the customer ceases to depend on the fixed displays, placed in waiting rooms, to know when it’s his turn. Moreover, the customer can get some ticket information, such as the queue status for the services that he wants and the order in which they will be served. (Figure 1. Get ticket example).

Still through the mobile device, the customer will be able to watch Corporate TV content, customized according to his personal profile. A Profile Matching algorithm, will choose videos to be shown to the customer.

This paper is structured as follows: Next section (2) presents some concepts and current queue management systems available. Section 3 describes the components of the proposed solution and its main features. Next, there is the evaluation of the solution in section 4, where the results of different tests are analysed. Finally, in section 5, future work is presented to complement the solution.

2 STATE OF THE ART

2.1 Q-Matic

Q-Matic queue management solutions are divided into two classes, Linear Queueing, figure 2 and Virtual Queueing, figure 3. On Linear Queueing systems the customer waits in a physical queue to be served. On the other hand, in Virtual Queueing the customer uses a ticket to define his position on a particular queue.

2.1.1 Linear Queueing

Linear Queues can be divided into three different categories.

SQSSP  Single Queue Single Service Point, which is the most basic form of queuing. Still used today in locations with few customers, or even before requesting a Virtual Queueing system ticket.

MQMSP  Multiple Queue Multiple Service Points, mainly used in areas with a large influx of customers, like supermarkets, or airport check-ins.

SQMSP  Single Queue Multiple Service Points, where only one queue is formed in order to ensure attendance order by arrival time, but do not allow for service separation.

2.1.2 Virtual Queueing

The Virtual Queueing Systems may be used as SQSSP or SQMSP scenarios, in which the customer receives a ticket and waits to be called. The calling order will be ensured, because all tickets are served sequentially. In Virtual Queueing there is the possibility of using a customers buffer before they are served. This buffer is a place where they move on a pre-call.
2.2 TI-2000

TI-2000 offers two product lines in the ticket based queue management systems. The QSystem[3] and the QSystem Lite[4]. The Lite version, narrower than the first, allows a maximum of 4 different services and 8 desks, which can serve one or more services. For service information, simple LED displays can be used or even a TV with the Corporate videos or TV advertising. The QSystem is a more complete solution in order to allow a greater number of services and desks. It also enables service appointments, ticket transfer between services, ticket priorities and statistic reports in real time.

2.3 Newvision

Another queue management system is presented by Newvision [5]. It presents two different solutions, for different scenarios. First of all, a more light version of the system, called Smart Inline (figure 4), where there is only one basic ticket dispenser, with 8 services available, up to 16 counters, a LED display panel to inform customers of their turn and the physical terminals at the desks. Organizations which require a higher degree of complexity, use the Corporate Inline[6] solution. It works as a client/server system where the peripheral server controls the tickets dispensers, displays and call terminals. Communication is established through TCP/IP with the server. It should also be noticed the Corporate TV presented in the Inline solution, as it has some similarities with the solution presented in this paper. The system allows some rules, thereby providing corporate videos/images according to the customers profile during the call. However, in this scenario the customer is distracted during the call, rather than during the waiting period.

2.4 Electronic Solution of paper tickets replacement

To solve the problem of paper tickets, the Federal University of Paraná presents an alternative[7] which is a system composed of a fixed station and several wireless modules which communicate through radio frequency. When the user arrives at the location, rather than receiving a paper ticket, it is assigned to a radio frequency module. The module signals the customers turn with a flashing LED. However, such system has some disadvantages. The notification is visual only, without any beep. Besides, it requires the use of more equipment than a person usually carries.

2.5 Qminder

Another solution which struggles against paper tickets is presented by Qminder[8]. As the previous one, it also allows an active waiting while the user waits his turn. Through a smartphone the user can request a virtual ticket for a particular service in a particular organization and then wait his turn. Communication with the system is done through the Internet and is open to any user. After requesting a ticket, the mobile application displays it on the screen, as we can see in figure 5, with information of the number of people waiting and when is the estimated time of service. When called, the customer is alerted by the application, to inform it’s his turn. This system is also designed to integrate with existing QMatic solutions for queue management service. The middleware communicates with the existing systems through freely available scripts by QMatic. However, this application have some problems regarding it’s navigation. After we get a ticket the application locks, where the only option available is to cancel the ticket.
2.6 Scheduling Algorithms

On the algorithm developed by Zhao and Ramamrithan[9], the problem is defined by a set of \( n \) tasks and a system with \( r \) resources. A task may require a number of resources. Mapping our problem with this scenario, we assume a task is a customer, a service is a resource and as such, a customer may need to use more than one service. This algorithm stops serving our problem, because it focuses on deadlines for a task and shared resources. The time restriction needed is only to avoid calling collision on the different departments.

Karger, Stein and Wein[10] present a set of scheduling algorithms for a generic problem defined as follows: tasks assigned to machines, subjected to restrictions, in order to optimize a certain function. To map out this problem into our context, we have customers as tasks which are attended in services represented by machines, subjected to restrictions of collision avoidance and to be served as soon as possible in all services. One of the described scenarios is called Open Shop, where several machines will perform operations related to a task. Continuing the mapping with our problem, operations may be different services that a customer needs. A restriction of this scenario, which becomes obvious in our context, is the impossibility of two operations being performed simultaneously on different machines. For this problem the algorithm presented is List Scheduling, known as a greedy algorithm. This algorithm was chosen for use in this work, because it presents a solution for the problem, without using too many restrictions not applicable in this context.

2.7 Profile Matching Algorithms

Wassermann and Zimmermann [11] present an algorithm based on statistical data for adaptation of user interfaces in different environments. Although the algorithm is interesting, in this context does not apply because there is no need to use statistical data. Only videos according to user preferences will be transmitted.

Another solution, in this case, based on a description logic language, by Cail, Calvanese, Colucci, Noia and Donini [12], is the definition of profiles through their own language. These profiles are quite complete because they include several properties. One of the characteristics of these profiles is the associated weight of a property. This kind of specificity is not necessary in our context, since the objective is to fill a short list of user interests, without too many options and less intrusive. We could implement the algorithm with a fixed weight in all properties, but these weights are then transformed into penalties for carrying out the calculations. As interest levels aren’t our objective, the penalties in the algorithm do not matter because they make the algorithm more complex than necessary.

3 ARCHITECTURE

In this section is introduced the proposed solution architecture. Firstly, the equations used for the main algorithm of multiple tickets, for the same user, in order to avoid service collision, are presented. After is explained the algorithm used for matching the customers profile according to the Corporate TV videos. Finally, the various components which compose the final solution are specified.

3.1 Problem Formulation

3.1.1 Efficient ticket management

The problem of tickets efficient management is the guarantee of a call to be made as soon as possible and with the assurance of no collision in the serving of more than one service. Thus, for a given set of services the user requests, it is necessary to sort them by the least waiting time \( T_e \). It is also necessary to ensure that the \( T_e \) of each service, added to the average serving time \( T_m \) of that same service and to the average moving time between the two services \( T_d \), does not collide with the \( T_e \) of the next service.

\[
T_e = \frac{T_m \times N_s}{N_b} 
\]

\[
\frac{T_m \times N_s}{N_b} + T_m + T_d < \left( \frac{T_m \times N_s}{N_b} \right)_{Next} 
\]

- \( T_m \) - Average serving time of a specific service
- \( T_d \) - Average moving time between services
- \( N_s \) - Number of tickets ahead of a specific service
- \( N_b \) - Number of desks available for a service

For example, on a services set of A,B,C, with waiting times in this exact order:

\[
T_e A < T_e B < T_e C \]

it is necessary to guarantee that for A,B:
\[
\frac{T_mA \times N_lA}{N_pA} + T_mA + T_dAB < \frac{T_mB \times N_lB}{N_pB} \quad (4)
\]

and for B, C:

\[
\frac{T_mB \times N_lB}{N_pB} + T_mB + T_dBC < \frac{T_mC \times N_lC}{N_pC} \quad (5)
\]

In case these rules are not followed, then one of the tickets that is colliding is delayed one or more positions as follows:

Assuming service A collides with service C, it is necessary do get a \(N_lA'\) and a \(N_lC'\) which follow these next rules:

\[
\frac{N_lA' \times T_mA}{N_pA} > \frac{T_mC \times N_lC}{N_pC} + T_mC + T_dCA \quad (6)
\]

\[
\frac{N_lC' \times T_mC}{N_pB} > \frac{T_mA \times N_lA}{N_pA} + T_mA + T_dAC \quad (7)
\]

It can be assumed that \(T_mC\) equals \(T_dAC\).

After obtaining \(N_lA'\) and \(N_lC'\) is then chosen the one that results in the less \(T_c\). This way we optimize the customers waiting period in the organization.

### 3.1.2 Profile Matching

Regarding the profile matching problem for Corporate TV content, it is used an algorithm based on score. If a video has one associated theme and if it belongs to the customer’s interest list, then the video earns one point.

- A customer has a list of interests - \(LI_\text{u}\)
- A video has a list of themes - \(LT_v\)
- Both lists belong to a set of themes - \(CT\)

\[
LI_\text{u} \subseteq CT \land LT_v \subseteq CT \quad (8)
\]

- A video \(v\) belongs to a set of videos - \(CV\)

\[
v \in CV \quad (9)
\]

For a given user with his \(LT_\text{u}\), we make the intersection of \(LT_v\) of each video \(v\). Then, the number of results in each intersection indicates the score of that particular video. Should this score exceeds a certain value \(X\) and the video becomes recommended for the customer and is inserted on his videos of interest list (VI).

\[
\forall v \in CV, if |LT_v \cap LI_\text{u}| > X \rightarrow v \in VI \quad (10)
\]

In addition to the interest list there are two values that help choosing the videos for the customers. Those are age and gender. With these properties the scoring system is not used. Both properties are exclusionary. In the case of age, it is assigned a range of recommended ages to the video. If the user does not belong in this range then this video is not considered for the next phase of scoring. Regarding the gender, if the video is classified to the opposite sex of the customer, it will not be considered. If there is a video that should not be differentiated by gender or age, must be rated for both genders and should have an age range large enough to be accepted.

### 3.2 Major Components of a Queue Management System

A queue management system is composed of several software and hardware components. For a better understanding of the architecture implemented, these components are presented in this section.

- Caller - A caller is the component which call the tickets. It is a physical device or an application used by employees of the organization.
- Ticket Dispenser - The ticket dispenser is responsible for providing the tickets to the customers. It can be a roll of pre-printed tickets or a ticket printer.
- Display - A display is a device which presents the status of the service queues. These displays may consist of LED arrays or general purpose screens (LCDs, plasma, etc.). In some cases, these last, displays also provide information and/or entertainment content to customers.

### 3.3 Technical Architecture

This section describes the components developed in this solution and the specification of their main functionality. In figure [8] can be seen these exact components.

#### 3.3.1 Server

Starting with the main component of the entire system, the server is a web application developed in Java, hosted at a Tomcat[13] application server with database in MySQL[14]. The server is responsible for the business logic of the entire system, as well as the operations for customized Corporate TV. The server is
Figure 6: Solution Architecture

also responsible for the notifications, both for callers, as for Displays. These notifications are made through a publish-subscribe protocol, where the server is the publisher and other entities the subscribers. The reason for this protocol’s choice relates to the fact that the server does not need to have information about each specific display, or caller, to make notifications.

3.3.2 Broker

To implement the notifications a Broker is needed. The server sends a notification to this broker with a message topic. Then, the broker makes the distribution of that notification to the subscribers of that topic. The chosen broker was Mosquitto[15]. This broker was selected because it is open source and supports the MQTT protocol[16]. This protocol is very lightweight and is designed for applications where its use does not require a major code modification.

3.3.3 Caller

The next component developed was a web application in GWT[17] to perform as a Caller. This application calls the server web services and is registered in the message broker.

3.3.4 Ticket Dispenser

The ticket dispenser was designed as an application to be placed in a touch screen kiosk. It is used by the customers to request tickets for the needed services. If the customer wants a paper ticket it is printed, but one of the goals is the virtualization of tickets, integrating this system with a mobile phone. Thus, it is possible to import the ticket to a mobile device with our client application installed. This import is made through a QRCode presented at the kiosk and read by the mobile application.

When the ticket is printed on paper, the kiosk allows the customer to check its status (estimated time for service, number of tickets ahead and last called ticket) by entering the ticket identifier at the kiosk. Besides the basic functionality of getting tickets, it is also possible to obtain the status of each service. The ticket dispenser was developed in Smart GWT[18] to be presented at an HTML page on the kiosk.

3.3.5 Displays

Fixed Display A screen for all customers to watch simultaneously with queue information. This display provides information of the available service queues, as well as some Corporate TV with generic content. The display has been implemented as a web application with Smart GWT, to be watched on a desktop computer through an HTML page. Receiving call and recall notifications is done by registering in the same message broker mentioned above. It was built, also in the HTML page, a player for the Corporate TV component.

Mobile Display It was created a mobile application that indicates, among other things, information relating to all queues, as well as specific information about the customers ticket. The customer can also watch the video contents of Corporate TV at the mobile device, with the difference that it is customized to himself. The application was developed for the Android 2.1 Operating System[19]. It calls the operations of the ticket system and the Corporate TV services in the application server.

3.3.6 Multimedia Server

Streaming Server For the Desktop displays, video contents are transmitted continuously and without any differentiation. For this scenario is used a simple streaming server, Windows Media Services[20], on a machine with Windows Media Server 2008, which transmits continuous streaming of pre-defined videos. The Desktop Display application receives this stream and displays it on the screen.

Video Server In the mobile application it is possible to watch appropriate content to the customers pro-
file. Through the algorithm presented above, the more appropriate videos will be chosen. The video files are hosted on a Web server where they are available for download/streaming. The container chosen was the MP4 with video codec H264 and audio codec AAC. This choice lies in the Android devices on the market decoding compatibility.

4 EVALUATION

Several points of the presented solution were evaluated. For a better understanding of these points and the results of these assessments, they were separated and are presented in the following subsections.

4.1 Server Performance

For this review we tested the algorithms execution timings, it was used a development laptop computer, with an Intel i5 @ 2.50GHz, 8GB of RAM installed and Windows 7 Home Premium 64-bit Operating System. As the Server component have two main algorithms, these were evaluated separately.

4.1.1 Ticket Sort

On this test, the set up was 5 different available services, served over 8 service counters. 100 customers which requested approximately 300 tickets uniformly distributed. On the ticket sorting, the following times were measured: 635ms, 520ms, 356ms, 391ms and remained close to 400ms. Conflict resolution was also decreasing with values around 16ms, 12ms, 9ms and remained below 10ms. With figure's 7 graph, we conclude the algorithm used to avoid collision of services is quite efficient, even with a lot of tickets in the queue.

4.1.2 Profile Matching

This evaluation consisted on the definition of 17 types of different video Interests/Themes. Then, 100 videos were added assigning to them these same themes. On the mobile device a customer profile was defined with the following characteristics: age 25; Male; and 6 different Interests. In the six different measurements made, the expended time values were around 100ms, with a maximum of 125ms and a minimum of 82ms. Through the graph on figure 8 we observe the video selection time is quite stable and the values presented, fast enough. Thereby avoiding a long waiting period for the corporate tv start.

4.2 Mobile Display Performance

The mobile equipment used in this evaluation was a Samsung Galaxy W (I8150), of personal use, with Android 2.3.6 (Gingerbread) operating system, running the (most relevant) following applications in background: Avast! Mobile Security; WhatsApp; AccuWeather; Email; Maps; The device was connected to the laptop used in the previous evaluation, with an USB cable. CPU and RAM usage measurements were made as follows:

- Through the adb application\footnote{Android Debug Bridge - command line tool that allows to communicate with emulators and physical devices connected to the computer} using the command "adb shell top -m 10" to display the CPU usage of all processes running on the device.
- Through the creation of a service, running in background, on the application, that calls a function of CPU usage measurement and prints it on the debug console.
- Through the mobile device own Task Manager, which displays information of the RAM used by the application, and CPU usage.

\begin{figure}[h]
\centering
\includegraphics[width=\columnwidth]{sort_timings.png}
\caption{Sort timings and conflict resolution}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\columnwidth]{video_selection_times.png}
\caption{Video selection timings}
\end{figure}
• Through DDMS\textsuperscript{[4]} using the information from the application’s memory stack.

4.2.1 CPU usage

Several measurements were made. These measurements can be found in the graph of figure 9. In the first case, in idle state, at the main menu, the values measured were around 0.04% and 0.05% of CPU usage. Next, a call to a webservice reached 1.5% usage, decreasing after it to values between 0% and 0.33%. Watching multimedia content have been also tested. Firstly, with videos in H.264/MP4 and secondly in VP8/WebM. The CPU usage percentage measured was quite similar in both cases with values between 0.5% and 2%. However, in VP8, reproduction took a lot more time to start and, in some tests, with vertical lines on the image. Moreover, this container and codec have never been considered for this solution because it is not supported on mobile devices with any version less than Android 2.3.3.

Figure 9: CPU usage

4.2.2 RAM usage

RAM usage values were measured in the previous scenarios, and in a few more specific cases. The values shown on figure’s graph, were obtained in the application’s stack memory readings in DDMS. Once again, the first scenario measurement was performed in idle state. In this case the value measured was 2.827 MB. With the tests evolution were analysed some ups and downs in the used memory, never exceeding 3.575 MB. The total value of the stack was going up depending on the scenario but did not exceed 8MB.

Although these are not critical values, for more limited devices, with short RAM available, can lower its performance. However we believe these values can be reduced, because a large portion of memory in use is due to the pictures used for the application design.

Figure 10: RAM usage

4.2.3 Usability testing

The usability tests consist in defining a set of tasks to be performed by a group of users, to assess the interface of the mobile application. Six tasks were defined to be performed by users. Through the graph of figure 11, the conclusions drawn from this evaluation with users is that the average error for each task is less than 1. Showing that the application is easy to use for customers. Regarding the average execution times, they were acceptable, as they were all below 30sec, with the exception of Task number 5, which is understandable since it was necessary to fill out a personal profile.

Figure 11: Average execution time and error

5 CONCLUSIONS

In this paper a queue management solution was proposed, intended to address three main points in the current ticketing systems. Firstly, it allows the use of multiple tickets without collision between the services needed. Secondly, de-materialize the paper
tickets, importing them to current smartphones, thus freeing users of geographic restriction, to obtain information of the service queues state. Finally, provide customized corporate TV content for each user, avoiding generalized content.

5.1 Future Work

In the future it is intend to implement the Back Office system, where it can be performed all the management of the system. And the Reports component, presenting statistical information about the use of this same system.

Also, some more features will be added, such as the employees differentiation, tickets reservation and geographical location within the organization using bluetooth.

Finally, the presented solution will be adapted to multiple branches of the same organization.

References