Software Development Aspects of a Mobile Food Ordering System

HRISHIKESH KULKARNI SERGIU DASCALU FREDERICK HARRIS

University of Nevada Reno, Nevada, USA {kulkarni, dascalus, fredh}@cse.unr.edu

Abstract— Mobile devices and wireless technologies are making a large impact on our lives. Companies that develop mobile applications are continuously evolving their products, aiming at satisfying the customers' needs even more comprehensively and at higher levels of quality. The Mobile Food Ordering System (MFOS) proposed in this paper is one of the tools that intends to provide a food ordering application on mobile devices for ordering food from different stores with the option of both delivery and pickup. MFOS offers many useful features to customers, including the map location of the food store, detailed menus, and previous order lookup. By using MFOS, the users do not have to wait in a queue at the restaurants. The paper presents the main functions, key usability aspects, software model, and prototype of MFOS.

Keywords— Pocket PC application, food ordering system, mobile ordering system, human-computer interaction, software model, prototype.

I. INTRODUCTION

We may not realize it but technology is making a large impact on the human life. It feels so strange when we hear that it is only 25 years since the first commercial cellular call was made in the United States [1], and today there are more than 262 million wireless subscribers in this country, 83% of the total Unites States population [1]. With the increasing number of cellular subscribers, cellular technology is also becoming more advanced. With increasing popularity and capability of mobile devices, new applications are being launched everyday. Today, many things that we can do using computers we can also do with mobile devices. But simply copying computer applications to mobile devices does not work, as user characteristics of computers and mobiles are quite different. Even though mobility and compactness are major advantages for mobile applications, the small screen, and the small and few buttons of mobile devices are among limitations that make it necessary for mobile application developers to study humancomputer interaction for mobile devices and develop their applications accordingly.

Mobile shopping or ordering is one of the most popular areas now-a-days. One can buy songs or any applications right from one's iPhone [2, 3]. One can also bid or buy items from eBay straight away using one's mobile [4]. Similarly, food ordering is also one of the major areas of mobile application development. Some companies have already launched their food ordering products in the market [5, 6, 7, 8, 9]. But these applications haven't fully exploited the current capabilities of

PDAs and missed on some of the required features which would benefit the end users. Thus, we have decided to design and develop a Pocket PC application, Mobile Food Ordering System (MFOS), using human-computer interaction principles for mobile devices.

As they say, time is money! The MFOS software will greatly benefit customers as they don't have to waste time going to the store and waiting in the queue. They will able to place food orders using their mobile device from anywhere even when they are traveling. They will get options to compare prices from different food stores and check out their weekly specials. Stores such as Pizza Hut [10] also have ordering systems like ordering from the Internet, or by phone call. However, not everyone is always in the vicinity of computer and then it also consumes more time to call an operator and decide on the menu. By using the proposed MFOS application, users can have updated menus on their mobile devices and they do not need to go for the inconvenient ways of listening to the options on phone, or finding a newspaper flyer.

The paper, in its remaining part, is organized as follows. Section 2 presents an introductory background about mobile devices in terms of new technologies in handheld devices, current research in human-computer interaction for mobile applications. Section 3 provides details of MFOS's requirements specification and use case modeling. Section 4 presents several of the more important aspects of MFOS's design. Section 5 describes MFOS's prototype. A brief look at related work is provided in Section 6. Finally, Section 7 outlines directions of future work and concludes the paper.

II. BACKGROUND

Before developing a mobile application, a thorough research needs to be done on which mobile device(s) one wants to target. As there are so many different mobile devices currently available in the market, each device may differ in screen size, operating system, and/or supported programming language. So studying the currently available handheld devices and their technologies is a must for a developer for figuring out his or her target audience and selecting the appropriate development tool(s).

According to the research in [11], it has been observed that Personal Digital Assistant (PDA) or Smartphone devices are currently owned by only a relatively small percentage of mobile phone users. However, with growing sales, the potential market for PDAs is increasing. The modern mobile phone market offers different devices for a wide variety of customer tastes and lifestyles. Some phones are small and sleek, and are popular for their ease of carrying, while some are chosen for their appearance so they can become a form of fashion symbols. Mobile manufactures are still developing new devices to attract different age groups and cultural groups. These devices will not only look different but also possess capabilities to attract specific groups of people, for example, phone manufactures are trying to portray their phone as a game console to attract the youngsters.

The Smartphones or PDAs attract both business and general users. They attract business users by providing productivity tools such as Word, Excel, E-mail, Organizer and they also include a virtual pop-up QWERTY keyboard for easy typing. They also attract general mobile users with their still camera, video camera, music player, radio, voice recording, games and internet browsing features. iPhone is a very good example of a PDA which contains all the above features and attracts basically all groups of people. Hence, considering the promising future of the PDA and Smartphone market, it is beneficial to write applications for PDAs that target large groups of people.

Research has shown that simply transferring a regular computer application into a mobile application almost always leads to a suboptimal mobile experience [12]. Hence before starting the development of applications for handheld devices, a developer should consider the mobile device user's characteristics as well as the limitations of mobile interfaces.

Based on an overview of the literature further detailed in [15], some of the mobile user characteristics and limitations can be outlined as follows: (1) Mobile users are mobile, as they are not sitting comfortably at a desk, like typical desktop or laptop users do; (2) Mobile users are interruptible and easily distracted. Mobile users use mobile devices while they are on a move, hence they easily can get distracted by the surrounding physical world. Sometimes mobile applications can be preempted by the mobile device itself (e.g. phone), for instance when receiving a call or message; (3) One cannot take something designed and implemented for a large PC monitor and simply re-write it for a Pocket PC device as mobile devices have very small screen resolution (e.g. 352 x 416 for Nokia E60), while usual PC monitors generally have much larger (e.g., 1280 x 1024) screen resolution; (4) Mobile users do not have a mouse to use for navigating through the screens. They have to navigate by using arrow keys, which can be tedious; (5) Basic mobile devices have 12-key numeric keypads, while Pocket PCs or smartphones have QWERTY keyboards, making them much more comfortable for the user. In mobile devices, the size of the keys is small, the distances between keys are also small, and the device itself is not normally placed on a firm surface; (6) As mobile phones are not connected by wires and can be carried to various locations, the quality and the strength of the communication signal cannot be guaranteed; (7) Unlike desktop or laptop computers, mobile phones are not always connected to a power supply, thus the more the mobile phone is in operation, the higher is the battery consumption.

HCI research is continuously aiming at providing design guidelines for mobile application development considering

mobile characteristics and limitations. Some of these design guidelines are as follows: (1) Store user preferences: as typing on mobile phones is much harder and time consuming than on desktop computers, if user preferences can be stored (e.g., a home address) the text entry can be reduced; (2) Use input sources from within the mobile instead of asking for manual feeding: specifically, use a GPS for location name (city and state names), camera for bar codes, and the mobile's clock for date and time. Also, use auto completion, image recognition or speech recognition to reduce text entry; (3) Provide quick response: mobile phones users are impatient and on the move, therefore the application should generate quick responses for them; (4) Ensure that the data is available when the user asks for it, regardless whether the network is available or not; (5) Promptly close network connections and file handlers and store the Record Management System (RMS) record shortly after use.

III. SPECIFICATION ASPECTS AND USE CASES OF MFOS

A. Functional and Non-Functional Requirements

This section describes functional and non-functional requirements of MFOS. Functional requirements define functions of a software system, i.e. how a software system should take a set of inputs, software system's behavior, and a set of outputs that a software system should produce.

Non-functional requirements represent various types of constraints placed on the software. Such constraints include both technological constraints and developer constraints such as process lifecycle used, implementation language, etc. [13].

Table I describes several functional requirements of MFOS. Due to space limitations, not all functional and non-functional requirements are shown in this paper.

B. Use Case Diagram and Use Cases

This section describes the use case diagram of MFOS. The MFOS system is represented by a bounded box labeled with the name of the system. The MFOS user is represented by an external actor, who interacts with the use cases inside the system boundary. Figure 1 shows the use case diagram of MFOS with one actor, i.e. the MFOS user, and 10 use cases. These use cases represent the core functionality of MFOS.

IV. MFOS DESIGN SPECIFICATION

A. Layered Architecture

Figure 2 shows the layered architecture of MFOS. It depicts the MFOS user interacting with the C# .NET Compact Framework forms on the mobile device. As displayed in Figure 2 this architecture has three layers: the Presentation Layer, the Business Layer, and the Data Access Layer. The Presentation Layer consists of C# .NET Compact Framework forms. The Business Layer consists of a set of objects that contain the business and application logic [14]. The bottom layer consists of three parts: the Phone API, the Service Gateways, and the Data Access Layer. Phone API consists of functions with which native phone functions can be called, such as SMS,

R 1	The MFOS shall provide multiple food stores for a user's selection.
R2	The MFOS shall provide option of pickup or delivery for a user's order.
R3	The MFOS shall provide a list of cities where MFOS service is available.
R4	The MFOS shall display multiple available store locations of selected food store for user's selection, when the selected order method is pick up.
R5	The MFOS shall show the map of the store location.
R6	Using MFOS the user shall be able to call the food store location.
R7	The MFOS shall display the prices of different food items.
R8	Using MFOS the user shall be able to add multiple food items to the cart in the same order.
R9	Using MFOS the user shall be able to zoom in and zoom out the map.
R10	Using MFOS the user shall be able to add the same food item more than once.

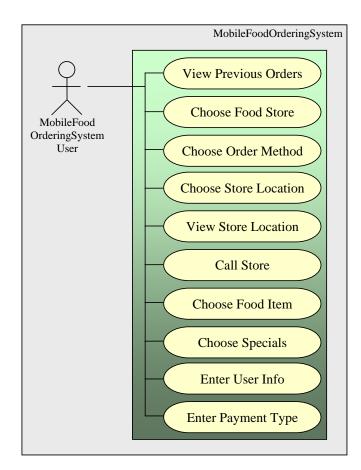


Figure 1. Use Case Diagram of MFOS

calling a number, etc. Using the Service Gateways, web services such as Google Static Maps API can be used. The Data Access Layer consists of the data access objects and the database itself [14]. MFOS uses a local database, managed by the SQL Server CE to temporarily store data on a mobile device. In the future, it will communicate with a larger SQL Server database on a web server through the Internet or on a PC through active synchronization.

B. State Chart Diagram

State chart diagrams describe the states and responses of a class. Figure 3 depicts the state transitions during MFOS execution. It describes the basic behavior of the system in response to external stimuli.

V. PROTOTYPE DETAILS

This section describes the MFOS prototype functionality with the help of screenshots and brief explanations. Due to space limitations, only two screenshots of MFOS "in action" are shown. More details about MFOS's prototype can be found in [15]. The screenshots presented in Figures 4-7 show MFOS running on the Microsoft Pocket PC 6 Emulator provided with the Microsoft Visual Studio .NET 2008 Integrated Development Environment (IDE) [16].

If the Start button is tapped on the Welcome screen, the order placing process begins, where first the user is directed to choose the food store. Figure 4 shows the Choose Food Store screen where the user will see all the food stores available in the system to choose from. When he or she selects any food store, the store logo is displayed to improve the look and feel of the prototype as well as for the ease of the user to visually confirm what he or she selected. For example, as shown in Figure 4, when the user tapped on the Indian Delicious food store, this store's logo is displayed.

Figure 5 shows the Choose Order Method screen where the user will be asked to choose the order method, either pickup or delivery. According to the user's selection, the next options

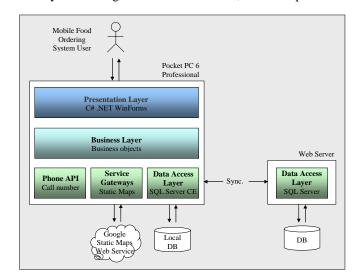


Figure 2. The Layered Architecture of MFOS

will be shown, e.g. if the user chooses pickup he or she will be asked to choose the store location from where he or she wants to pick up his or her order, whereas the user will not receive this screen if he or she chooses delivery. The user will also be directed to choose the city and state on the same screen. The city and state information narrows the store locations according to the user's choice.

The user can also see the map of the store by using the Show Map button on the Choose Store Location screen. As the application is already connected to the internet, it will get the map using the Google map web service, which indicates the store location on the map with a red pin. The user can also zoom-in and zoom-out to get a better idea of the store location. Figure 6 shows one of the store locations on the map.

After selecting the store location for pickup from the Choose Store Location screen or directly after choosing the Delivery option, the food menu will be shown to let the user select his or her favorite food for order. In the Food Menu screen the user will see all the food items that the food store offers. Next to the name of the item the user will also see the price of the food item. To add a food item to the cart, the user will just tap on the food item and then taps on the Add to Cart button. A confirmation message that "the item is added to the cart" will also be shown. From the same screen the user can go to the cart for checkout or he or she can check food store specials. Figure 7 shows some of the food items in the food menu screen of the Indian food store.

VI. RELATED WORK

As mobile phones and mobile technology are becoming more advanced and popular, many mobile food ordering products are being launched to attract customers. This section summarizes the differences between some of these food ordering products in the market and MFOS. The MFOS project developed is essentially a prototype created in the academic environment and the commercial products are much more powerful. Thus, the brief comparison on the basis of features offered by different products presented in this chapter is for information purposes only, and should not be taken as an indicator of the products' strengths. For more precise details, the reader is asked to visit the references provided for the products mentioned in this chapter. CityMint [5] is a food ordering product for iPhone devices designed to order food from local food stores in New York City. CityMint lets one browse the menus of local places and add different food items into a single order. The user can choose either delivery or pickup depending upon store offerings; the application also has an option of paying right from one's phone. After placing the order it will send a user confirmation via e-mail with one's estimated delivery time, and details of the order [17]. MFOS aims to offer these features as well. Additionally, MFOS allows users to select food items from store specials, shows the store location on the map, facilitates calling the store from within the application, and as MFOS aims to work with fast food and restaurant chains this application could in the future incorporate more cities.

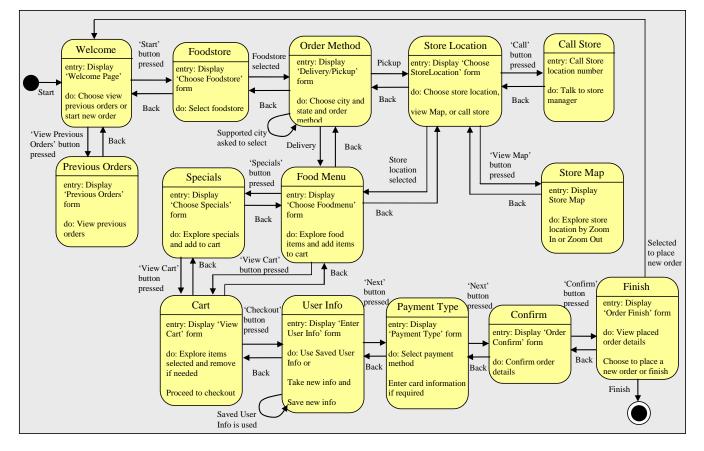


Figure 3. State Chart Diagram of MFOS



Figure 4. Choose Food Store screen in MFOS



Figure 5. Choose Order Method as well as City and State in MFOS

Food To Go [6] is a PDA application where the user can enter the quantity of the food items from the food store menu. The user also has the option of placing special cooking instructions. Then the users just have to give their name, address, phone number, and estimated pickup time. Although MFOS works on the same technology (Pocket PC), it has some differences in functionality. Specifically, the current version of MFOS does not take special cooking instructions and does not allow the user to select the estimated pickup time. However, MFOS is designed to offer some other useful features such as delivery option, store map, and calling the store.

GetQuik [7] is a web as well as a mobile-based application. Using the GetQuik mobile application one can select the food store, store location, choose food items or choose from favorites, and confirm the order. GetQuik also works on iPhones. MFOS aims to offer similar features except the option of selecting food items from favorites and iPhone and WAP-based phone compatibility. However, MFOS is designed to offers other features, such as maps and credit card payment.

GoCelly [8] is an SMS-based food ordering application for pickup. The user first needs to configure different orders on their website and save them with a nickname. For example, the user chooses the food store, then chooses the food items, customizes the order and saves it with a nickname such as MyOrder1. Next, whenever the user wants to place this order he or she can SMS the nickname to GoCelly service number. But this is restricted in terms of changing orders on the fly and also requires the user to remember all the nicknames corresponding to their orders. MFOS displays the food menu every time one wants to order so one does not have to remember all the nicknames for the orders. MFOS also provides a rich user interface and other options for users.

The tools mentioned above are powerful commercial applications. They provide some features that MFOS does not provide and are currently operating on the market. However, none of the above applications currently includes a store map, or the option of calling the store from within the application (which MFOS includes in its design).

VII. FUTURE WORK AND CONCLUSIONS

This section discusses some of these potential areas where further work can be done to improve the MFOS application.

As we have shown in the Section VII, the MFOS prototype offers a set of valuable functions but several useful functions which are offered by other products are missing in the MFOS prototype.

Following are some of the features required to be added in the MFOS prototype to make it a more complete and better application. (1) MFOS should include the ability of saving favorite items so that the user can skip some steps of placing an order and save time. (2) The user should be able to give special instructions for his or her order. (3) The user should be able to select a custom pickup time according to his or her preference. (4) The user should be able to repeat the order from his or her last orders. (5) The Menu should be more detailed in terms of images and descriptions, such that the user will know better about the food items he or she can choose from. (6) The user should be able to add multiple items into the cart at a time. (7) Provide a map with driving directions.

For increasing the customer base MFOS Application should work on many mobile devices. Currently MFOS only works on Pocket PC with Windows Mobile 6 Professional. To reach a wider audience it is necessary to make sure that MFOS can work on other platforms as well. As a development language we used C# so with relatively few modifications this application could work on other Windows Mobile OS. For



Figure 6. Map of the Store Location in MFOS



Figure 7. Food Menu screen of one of the food stores in MFOS

other handheld devices such as iPhone and Blackberry this application has to be re-written in programming languages compatible with their OS. Another idea for reaching more customers would be for users who do not have internet connections on their devices, they should be able to send SMS messages for ordering. MFOS has been developed considering the usability of mobile application as well as considering human-computer interaction challenges of mobile devices. We are planning to test this product with real users and according to their experiences we would like to further improve MFOS' usability.

The goal of this paper was to explore the human-computer interaction challenges in designing commercial applications for mobile devices through the development of MFOS. MFOS was designed and developed to provide Pocket PC users with a single tool with which they could order their favorite food from different cuisines without having to wait in the queue and, importantly, be able to order food from anywhere. While developing the interface of MFOS, researching humancomputer interaction challenges as well as designing a system which can deal with them and provide more useful features to the user were significant objectives of this paper.

MFOS aims to offer some "cool" and handy features to the users, which other products in the market do not provide in the same way as MFOS. The location of the food store on the map is one of the very useful features of MFOS, which uses the latest functions of the Pocket PC. Similarly, the ability to call the food store, view the menu, pay the order with a credit card, save user information, and view previously placed orders are also several other valuable features of the system.

REFERENCES

- CTIA-The Wireless Association Celebrates 25 Years of Mobile Communication http://www.ctia.org/media/press/body.cfm/prid/1781 Accessed: 12/28/08
- [2] Apple iPhone Features, http://www.apple.com iphone/features/ itunes.html, Accessed: 12/26/08
- [3] Apple iPhone Features, http://www.apple.com/iphone/features/ appstore.html, Accessed: 12/26/08
- [4] eBay Mobile, http://pages.ebay.com/mobile/ Accessed: 12/26/08
- [5] CityMint Food on the Go, http://www.citymint.com/, Accessed: 12/27/08
- [6] Mobile Food To Go, http://www.mobilefoodtogo.com/, Accessed: 12/27/08
- [7] GetQuik Restaurant Orders Made Easy, http://www.getquik.com /Pages/Common/Demo.aspx, Accessed: 12/27/08
- [8] gocelly Food On the Go VIP Style, http://www.gocelly.com/, Accessed: 12/27/08
- [9] GoMobo: Order Food Ahead, Skip the Line!, http://gomobo.com/, Accessed: 12/27/08
- [10] Pizza Hut!, http://www.pizzahut.com/index.html, Accessed: 12/26/08
- [11] Attewell, J., Mobile technologies and learning: A technology update and mlearning. Project summary, London: Learning and Skills Development Agency, 2005.
- [12] Ballard B., Designing the Mobile User Experience, Wiley, 2007.
- [13] Sommerville, I., *Software Engineering*, 8th edition, Addison-Wesley, 2006.
- [14] Lee, V., Schneider, H., and Schell, R., *Mobile Applications Architecture, Design, and Development.* Prentice Hall, 2004.
- [15] Kulkarni H., "Mobile Food Ordering System (MFOS)", Master thesis, University of Nevada, Reno, USA, 2009.
- [16] Windows Mobile 6 Professional and Standard Software Development Kits Refresh, http://www.microsoft.com/downloads/details.aspx? familyid=06111A3A-A651-4745-88EF-D48091A390B&displaylang= en, Accessed: 12/26/08
- [17] CNET: Order food on your phone with CityMint, http://news. cnet.com/8301-17939_109-9950302-2.html, Accessed: 12/27/08