

Android-Based Mobile Application for Door-to-Door Product Delivery

<https://doi.org/10.3991/ijim.v13i03.9598>

Samer Atawneh (✉)

Saudi Electronic University, Riyadh, Saudi Arabia,
satawneh@seu.edu.sa

Basil Al-Kasasbeh

Arab Open University, Riyadh, Saudi Arabia,

Mohammed Ben Rshed

Saudi Aramco, Dahrhan, Saudi Arabia

Abstract—Many companies, such as Aramex, FedEx, and SMSA, offer product delivery services. However, the delivery process through these companies is costly and/or requires the customer's physical attendance at the company to get the sent shipments. There is a persistent need to improve the delivery process in Saudi Arabia to reduce the effort, cost, and time that the customer spends to get the shipped products. This paper presents a new delivery approach in Saudi Arabia by developing an Android-based MobApp that allows the customers to use their mobile devices to send and receive shipped products at their doorstep by submitting online requests through the developed MobApp. The proposed MobApp, named as Door-to-Door (D2D) product delivery MobApp, guarantees fast and costless service among its competitors. The MobApp will provide its customers a reliable delivery process. It aims to provide a domestic delivery chain with whomever to wherever within Saudi Arabia. In addition, the proposed delivery MobApp allows the customers to create, update and track the delivery orders. Moreover, the proposed D2D delivery MobApp is easy to install and use, it provides a friendly GUI and has a powerful steady performance.

Keywords—MobApp, delivery process, shipped products, quality attributes, use cases.

1 Introduction

Mobile devices became the sole gatekeeper for many valuable personal information [1]. The global dissemination of mobile devices has changed the peoples' data utilization behavior during the last decade [2]. In addition, developing mobile applications (MobApps) has emerged as one of the most focused software industry areas [3]. Today, mobile applications are considered as an integral part of human's daily life [4]. The growing popularity of technologies and MobApps are leading many companies to de-

velop strong relationships with customers through MobApps [5]. Today, many companies are using MobApps to support various activities in several sectors, such as education and industry. Android OS incorporates third-party applications that run on the top of Android-based mobiles [6]. This OS has many advantages over simplicity, extensibility and flexibility [7].

Recently, people use the latest technology and devices to perform their daily duties. One of these duties is buying online by placing online product orders. People place online orders while they are at home. However, approximately 33% of consumers are away from their houses, and most of these people are below 35 years old [8]. The aim of this paper is to propose and develop an Android-based MobApp to integrate the features of different types of delivery services in one spot. The system will serve customers whom would like to receive commercial or personal products to their doorstep, which may include purchased electronic devices, household items whether from a supermarket or a local shop, as well as pre-ordered food delivery service. Additionally, customers may request an item delivery such as personal devices, furniture and envelopes directly from their work or residence to others within Saudi Arabia. The proposed MobApp provides a convenient and affordable delivery service that serves customers within the city boarder and between different cities within Saudi Arabia. It provides an alternative solution to solve some of the problems such as waiting in long queues at outlets to receive the sent items and to minimize driving and traveling times of customers by extending coverage of the delivery services.

The rest of this paper is organized as follows. Section 2 briefly reviews current MobApps in the area of product delivery. Sections 3 introduces the proposed D2D delivery MobApp. Section 4 presents the implementation and testing of the proposed MobApp. Finally, Section 5 draws the conclusion and possible options for future works.

2 Literature Review

It is worth mentioning that consumers still more commonly use mobile devices to place their orders for takeaway/home delivered meals rather than ordering by computer. A Survey work showed that 57% of respondents had ordered by mobile and 36% had ordered by computer [9]. Overall, 83% of respondents had ordered takeaways and other home-delivered meals directly from a restaurant at some time, while 43% had placed such an order through a third-party platform provider (such as Deliveroo or Just Eat) [9]. However, online ordering has increasing rapidly in recent years, and is expected to continue to do so. Domino's (the pizza chain) which has developed an advanced online ordering system and application generates almost 80% of all its UK deliveries from online ordering. Third-party platforms such as UberEats, Deliveroo and Hungry House also helped to facilitate growth in online orders and facilitated several smaller restaurants to attract online customers [8]. There are quite several good examples of delivery applications currently in business. Some of these applications, such as UberEAT, Careem, and Shop&Ship, have inspired the idea of creating the proposed delivery MobApp. These applications have different features. This section reviews three similar delivery applications and exploring the capabilities and services they provide.

2.1 UberEats

UberEats [10] is a food delivery service was extended from Uber limousine service. UberEats has hundreds of restaurants to choose from. When the customer uses the restaurant’s application, he/she can scroll through the feed for inspiration or search for a particular restaurant. When the customer finds something he/she likes, he/she taps to add it to the cart. When the customer is ready to check out, he/she will see his/her address, the estimated delivery time, and the price of the order including tax and booking fees. When everything looks right, the customer just taps “Place order” and that is it. A nearby UberEats delivery driver—on a car, on a bike, or scooter—will go to the restaurant to pick up the customer’s order. The customer will be able to see the name and the photo of the driver and track the order on the map. UberEATS guarantees about 30 minutes as a delivery time with no delivery fees [8]. Figure 1 illustrates UberEats application selecting “delivery pick up options” and “delivery location” screens.

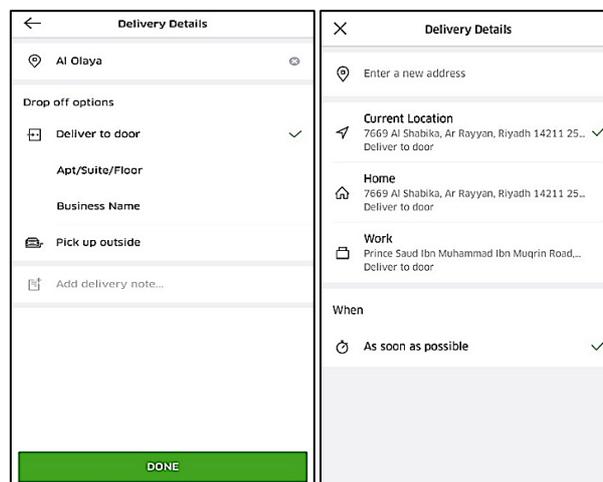


Fig. 1. UberEats application: Selecting delivery pick up options and delivery location

2.2 Careem box

Careem [11], which operates in major cities such as Riyadh and Dubai, has released a new delivery service called “Careem Box” for sending, tracking, and receiving small items, such as keys, documents, pharmacy pickups, and even a pair of shoes just booking a box. When you need an item to travel across a town, you book a box by selecting it from the car type menu. The nearest bike will be matched with you and you will see an estimated time of delivery and fare estimate. You can track your items in a real time as they travel. In addition, once the delivery is made, the delivery driver is charged to your Careem account and you can pay the way you want: cash, card or credit [11]. Figure 2 illustrates Careem application “user account menu” and “pickup location map” screens.

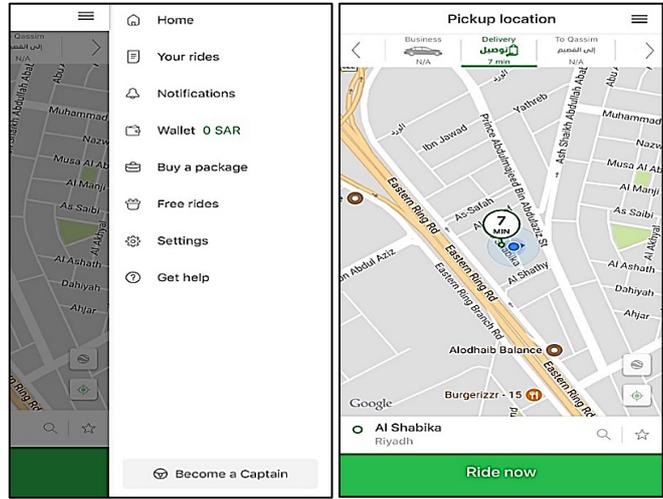


Fig. 2. Careem application: User account menu and pickup location map

2.3 Shop and ship

Shop and Ship [12] is currently available globally in over 80 destinations. Once registered with Shop and Ship, you get 24 personalized physical addresses in several global locations. You can use these addresses to shop from different countries and it will be delivered to where you live. The customer may order online from any retail shop located overseas or within the country using the provided address by Shop and Ship application for delivery. The requested items from Shop and Ship warehouse will be dispatched to the customer’s home or work address as ordered. Figure 3 illustrates Shop and Ship application “Login page” and “user account” screens.

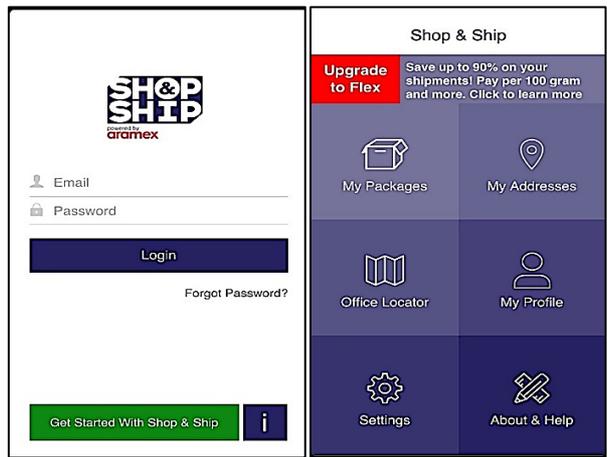


Fig. 2. Shop and Ship application: Login and user account pages

3 Proposed Product Delivery MobApp

This section provides a full description for the proposed D2D product delivery MobApp. The key functions of the proposed MobApp and the main services that provided to the customers along with the types of the users who can interact and use this MobApp will be discussed. The software requirements (Both functional and non-functional requirements) and the environment Platform and assumptions will be discussed in this part as well.

3.1 Proposed MobApp functions

The proposed D2D product delivery MobApp offers simple and effective navigational process through the application functions that include:

- New customer registration.
- User pickup and desired delivery locations.
- Flexible delivery schedule.
- Shipment tracker dialog.
- Notification emails upon shipment request approval.
- Summary of last requests.
- Different payment methods.
- Customer evaluation of the level of service.

3.2 Types of users and main services of D2D product delivery MobApp

Three classes of users will be able to use the proposed D2D product delivery MobApp; these types are customer, administrator, and employee classes. The customer is the primary focus class for the proposed MobApp where the main goal of this MobApp is to make customers able to send and receive product shipments by submitting online requests. The administrator has full controls over all activities being performed by customers. In addition, the administrator has the ability to view, edit, add and delete any service from that offered services list. Moreover, the administrator can handle the help requests, feedbacks, and complaints received from the customers. The Employee is responsible to pick up and deliver the package and the administrator assigns an employee for each shipment request. Figure 4 illustrates the key users of the proposed D2D product delivery MobApp.

The proposed D2D product delivery MobApp is where customers can send and receive shipments such as envelopes, household material and devices, and personal and business-related packages. The delivery process through D2D delivery MobApp does not require customer's physical attendance, rather, customers are able to send and receive the sent shipments at their doorsteps by submitting online requests. Upon initial installation of D2D delivery MobApp, a customer will have to register to the service. A new registration requires assigning unique customer ID, and during the registration process, customers will add their addresses and contact information, login password, and accept service terms and conditions.

Once an account is created, a confirmation email is sent to customer email address. Afterward, a customer will fill out a request form for each new shipment that include information about the package, as well as the required pickup and delivery address, and date and time of product delivery. Next, D2D MobApp admin will review the created shipment request and assign an employee for pickup and delivery this shipment. When the employee is assigned, the customer will receive a confirmation email with delivery timeline. D2D MobApp allow customers to track their shipments, and update and cancel current requests prior to be processed. The system provides two payment methods to close out the shipment request, a customer will choose to pay in cash upon delivery or at pickup time, or submit an online payment using a credit card. At the end of each delivery, the customer will be able to submit a service feedback, evaluate the service, and submit a complaint. Shipments' history record will be available under each customer account for preview and future references. Further, the customers of D2D delivery MobApp have the right to update their account information as required, and include their shipment address and contact details. The following subsections describe the main counterparts of the proposed D2D product delivery MobApp.

Customer: The customer can send and receive product packages through the D2D delivery MobApp. If the D2D MobApp serves the customer needs then the customer will use it repeatedly. Upon each request, a customer should provide details of date and time, and pickup location and the shipment destination. Cost is calculated based on the size of the package and the delivery destination.

Employee: The employee is the person who is responsible to pick up and deliver the package. To fulfil a customer request, the employee acquires accurate information to complete the task properly, such as package pickup location and delivery destination. In addition, he/she must check that the content of package meets the package information submitted online by the customer.

Administrator: The administrator sets rules and conditions of the delivery process, monitor package delivery and provide customer support. In addition, the administrator updates the MobApp features according to customer feedback and business demand, and updates the delivery cost when deemed necessary to maintain market share value.

Package: The package size and content depend on the customer selection. The package specifications must be described clearly in the customer delivery request including the dimension and weight, so the administrator can plan the delivery process and assign an employee for pickup process. Furthermore, the payment of the package delivery depends on the package specification, more specifically, on the package size and delivery distention.

Request: The request submitted by the customer contains the package size and content and the pickup and delivery location. The request specifies the time of pickup and the acknowledgment of receiving the package by the employee. When the customer submits the request, he/she can monitor the request status and evaluate the service once concluded.

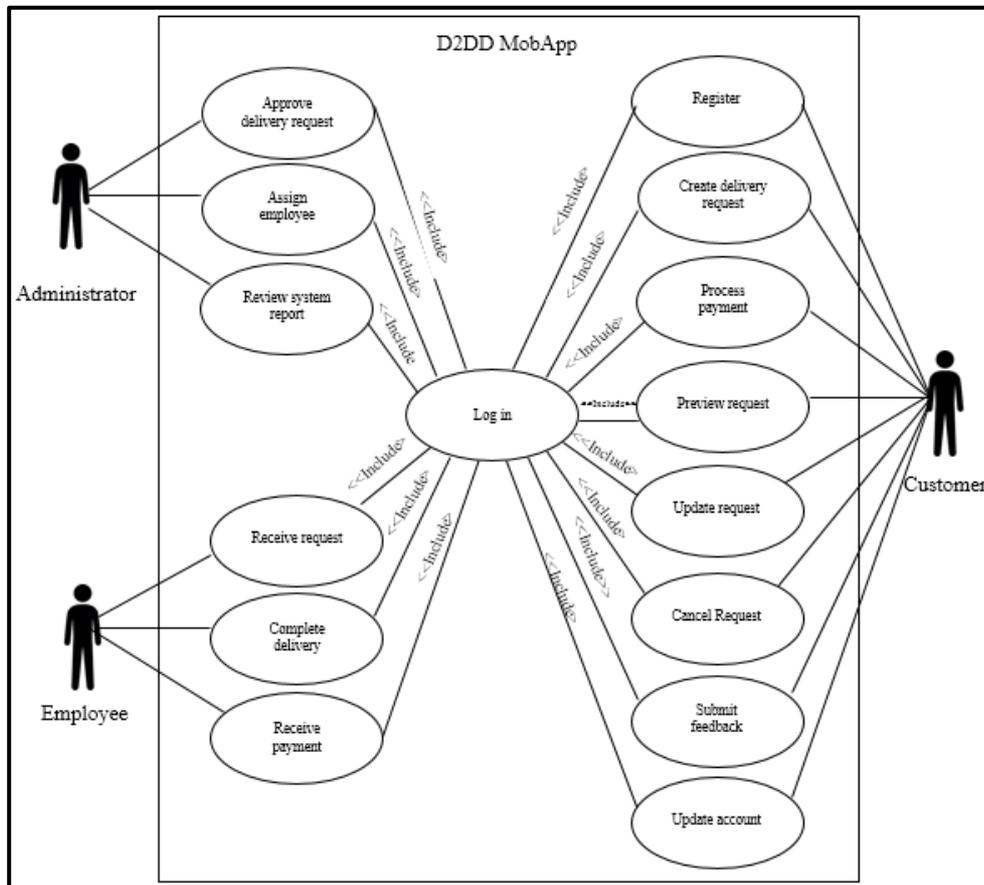


Fig. 3. Use Case Diagram of D2D product delivery MobApp

3.3 Software requirements

In this section, both functional and non-functional requirements of the proposed D2D delivery MobApp will be discussed.

Functional Requirements (FRs): As mentioned early, the proposed MobApp has three types of users: customer, administrator, and employee. Each user has different functional requirements. This part gives the key FRs for these users.

FR1: Creating a new user account (i.e., registration) including the following:

- Event/Use Case: Creating a new user account.
- Description: The MobApp allows users to create new accounts to use the application.
- Rationale: The customer needs to download the D2D delivery MobApp and enter valid data.
- Fit-criteria: The application successfully created a new account for the customer.
- Dependencies: None.

- Conflicts: None.

FR2: Creating a Delivery Request including the following:

- Event/Use Case: Creating a delivery request.
- Description: The MobApp allows customers to create a new delivery request. The customer fills up the request information (source, destination, time ...) and receives a confirmation message after the request successfully submitted.
- Rationale: The customer needs to register to the D2D delivery MobApp.
- Fit-criteria: The application successfully accepted a new delivery request.
- Dependencies: FR1: Registration.
- Conflicts: None.

FR3: Paying delivery fees including the following:

- Event/Use Case: Paying delivery fees.
- Description: The MobApp allows customers to pay fees of their delivery requests. The customer reviews the request, the application displays the payment options, and the customer chooses and pays by a suitable method. The application validates the payment method and sends a notification message to the customer.
- Rationale: The customer needs to put a delivery request first through the D2D delivery MobApp.
- Fit-criteria: The application successfully accepted the payment process.
- Dependencies: FR2: Create a delivery request.
- Conflicts: None.

FR4: Track request including the following:

- Event/Use Case: Track request status.
- Description: The MobApp allows customers to track their active requests statuses.
- Rationale: The customer needs to create a delivery request and pay the fees through the D2D delivery MobApp.
- Fit-criteria: The application successfully shows the active requests statuses.
- Dependencies: FR2 & FR3: Create a delivery request and pay the delivery fees.
- Conflicts: None.

FR5: Update request order including the following:

- Event/Use Case: Update request.
- Description: The MobApp allows customers to update their delivery request orders. The customer changes the request information (source, destination, time ...) as necessary and receives a notification message after the request successfully submitted.
- Rationale: The customer needs to put a delivery request through the D2D delivery MobApp.
- Fit-criteria: The application successfully accepts the update on the request order.
- Dependencies: FR2: Create a delivery request.
- Conflicts: None.

FR6: Cancel request order including the following:

- Event/Use Case: Cancel request order.
- Description: The MobApp allows customers to cancel their delivery request orders before the administrator confirms the requests.
- Rationale: The customer needs to put a delivery request through the D2D delivery MobApp.
- Fit-criteria: The application successfully cancels on the request order.
- Dependencies: FR2: Create a delivery request.
- Conflicts: None.

FR7: Submit feedback including the following:

- Event/Use Case: Submit feedback.
- Description: The MobApp allows customers to evaluate the service and submit feedbacks and complaints. Once a customer submitted a feedback or a complaint, the application validates feedback/complaint entry and accept the request.
- Rationale: The customer needs to create a delivery request and pay the fees through the D2D delivery MobApp.
- Fit-criteria: The application successfully accepts on the feedback/complaint request.
- Dependencies: FR2 & FR3: Create a delivery request and pay the delivery fees.
- Conflicts: None.

FR8: Update account including the following:

- Event/Use Case: Update account.
- Description: The MobApp allows customers to update their personal information. Once a customer submitted the new information, the application validates the information entry and sends the customer a notification message.
- Rationale: The customer needs to register to the D2D delivery MobApp.
- Fit-criteria: The application successfully accepts the new customer information.
- Dependencies: FR1: Registration.
- Conflicts: None.

Non-Functional Requirements (NFRs): The proposed D2D delivery MobApp has been developed to operate in different operating conditions, such as different supported screen resolutions and network bandwidth situations and network types (3G, 4G, and Wi-Fi). In addition, important technical issues were taken into consideration to achieve an acceptable performance level of the proposed D2D delivery MobApp. It is highly recommended for the user of the proposed MobApp to regularly check the Google Play Store for any new security updates. Additionally, the user should download a well-known anti-virus application to protect his/her mobile device from any Internet viruses, malware and unlicensed applications that may steal his/her identity. Moreover, a list of non-functional requirements (NFRs), known as quality attributes, were taken into consideration. This section presents these NFRs:

- Usability: The proposed D2D delivery MobApp is ease to use: users can perform the operations of the MobApp in a simple way.
- Availability: The proposed MobApp is available 24 hours a day, 7 days a week, and provides the delivery process during the working hours.
- Look and feel: The application interface is simple and does not contain many colors.
- Efficiency: The proposed MobApp performs all the required services properly.
- Compatibility: The proposed MobApp is compatible with Android-based devices with Android version 5.0 and above.
- Cultural: The language used in the proposed MobApp is a formal one (English).
- Security: The proposed MobApp allows users to access the application functions only after they login to the system. In addition, the proposed application sends the users a system message when any activity is executed, whether successfully or not. In addition, the users' private information is securely encrypted.

3.4 Environment platform, constrains, and assumptions

The proposed D2D product delivery MobApp will run on all Android devices with version 5.0 and above. The MobApp is developed using Android Studio platform and its supporting library, besides Java PL. A set of constraints were taken into consideration during the design and implementation stages. These constraints include the following:

- Customers and employees are responsible for their profile contents, usernames and passwords
- Customers should provide the accurate information of location and contents of packages.
- Customers have to approve the delivery process after the delivery.
- Employees are responsible for the package safety during the delivery process.
- Employees have to verify the payment accuracy.
- Employees have to guarantee pickup and delivery process.
- Verification of package recipient will be via a confirmation code.

To accomplish the success of delivery, the delivery process should stick on the requirements and follow some assumptions. These assumptions were selected to support the implementation and to avoid any problem might produce problems for the proposed MobApp. Following are summarizing a list of the selected assumptions:

- The mobile operation system is Android (v. 5.0 or later).
- The proposed MobApp supports English language.
- The proposed MobApp uses Map to serve the source and destination of delivery.
- Delivery scope will be initially in Riyadh city.
- Customers and employees should be familiar with Android operating system and Google Map application.

4 D2D Delivery MobApp: Implementation, Testing and Discussion

This section provides all the details about the implementation of the proposed D2D product delivery MobApp, the testing results and evaluation.

4.1 Implementing D2D product delivery MobApp

As mentioned before in the previous sections, customers can send and receive the sent shipments at their doorsteps by submitting online requests through the proposed D2D delivery MobApp. The proposed MobApp runs on all Android devices with version 5.0 and above. It is important to notice that to avoid any conflict and to make sure that the database will remain consistent all the time, the proposed MobApp was developed to be with a direct connection with the centralized database server. Moreover, using a direct connection will guarantee that any executed delivery process from the MobApp side will be directly reflected on the centralized database. This guarantees that the database will be always up to date. Figure 5 illustrates the structure of the D2D product delivery MobApp.

For successful operability and stability, the developed MobApp is linked to a sustainable online database, called “Firebase”, which is provided by Google. The “Database Handler Java” classes created via Android Studio linked via Google API file. The database schema of the developed MobApp uses Firebase JSON tree structure; here are no tables or records. That is, when you add data to the JSON tree, it becomes a node in the existing JSON structure with an associated key.

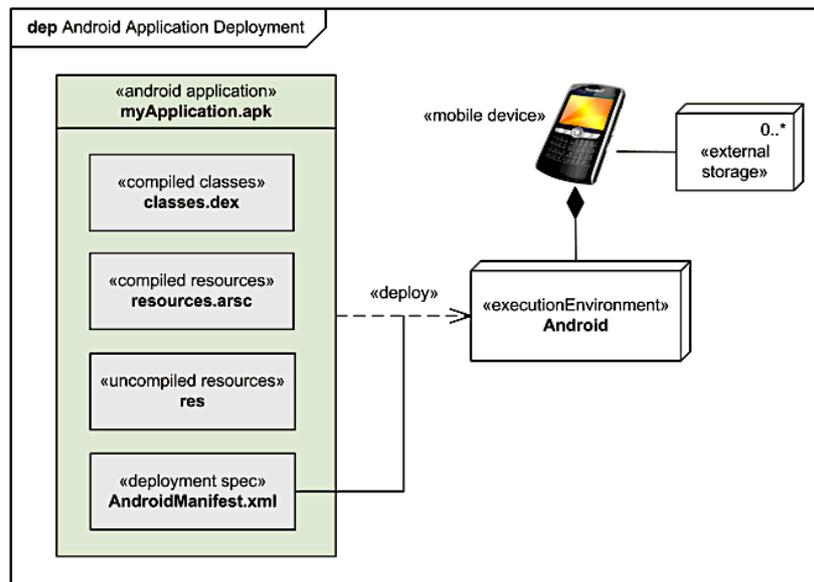


Fig. 4. The architecture of D2D product delivery MobApp

4.2 D2D delivery MobApp interface

The D2D product delivery MobApp successfully achieved all the mentioned requirements and services mentioned in Section 3. This sub-section illustrates the GUI for some of the MobApp services. Figure 6 shows the welcome and login screens of D2D MobApp. The welcome screen shows the first page the MobApp user will see when opening the D2D delivery MobApp, this screen splashes for 4 seconds then application switches the view to login home screen. The Login Screen allows the MobApp user to select whether to access as a customer, an administrator or as an employee. Upon selecting the user type, the second login screen will appear to allow user entering access cardinality.

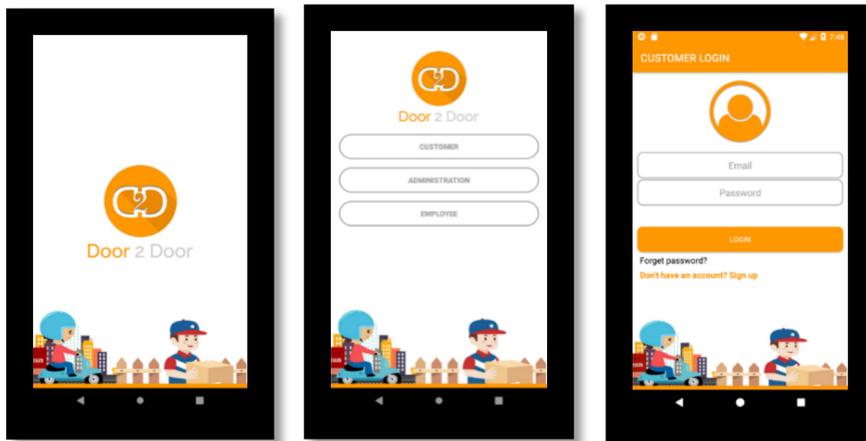


Fig. 5. Welcome and Login Screens of D2D MobApp

Figure 5 shows the customer registration and homepage screens. The customer registration screen shows the case when the customer uses the D2D Delivery MobApp for the first time, he/she will be accessing the registration form through the user login page. The customer enters his/her details and click the register button. Upon clicking register, the customer is directed to the homepage of the D2D Delivery MobApp. The customer homepage screen allows the customer to preview his profile information, check the requests that he/she have created, and create new requests.

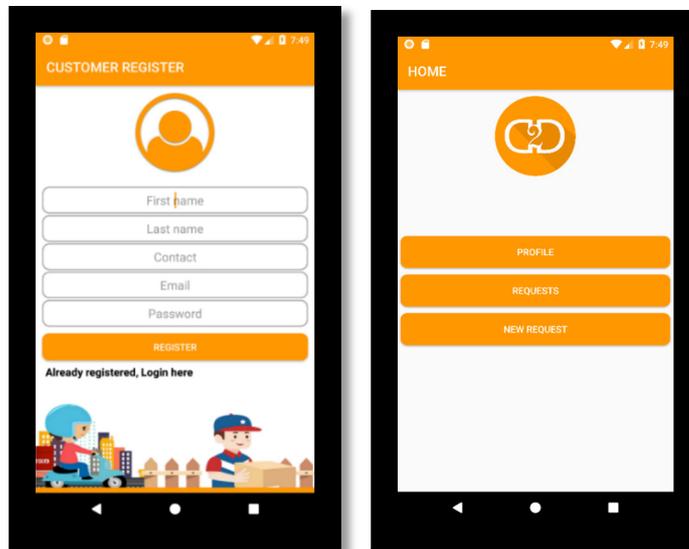


Fig. 6. The Customer Registration and Homepage screens

Figure 7 shows the screens of the new delivery request and the list of customer requests. The new delivery request allows the customer to add the details of the package, such as the package type, specific package weight and the package description. Additionally, the Google map appears to allow customer allocating his/her exact location for easy delivery process. The list of customer requests screen shows the created requests by the customer. The details of the request allows the customer to follow up on the request to the complete delivery.

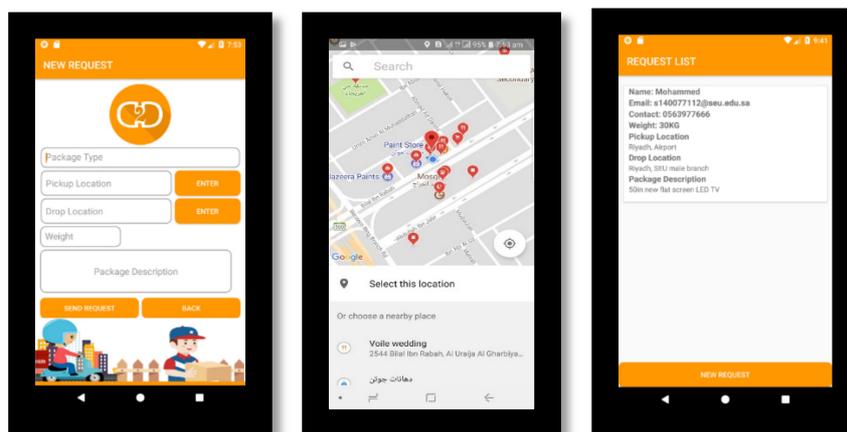


Fig. 7. The new delivery request and the customer requests List Screens

Figure 8 shows the screens related to the MobApp administrators and employees. The Administrator homepage screen provides two main functions: adding new employees and reviewing the customer requests in order to process the delivery requests. The list of new requests screen shows the list of all customers' requests that appear under the administrator account. The list contains detailed information about all requests. The employee homepage screen shows two main functions: the profile view that shows the employee information and the list of customer requests. The employee can preview the details of all customers' requests assigned to him.

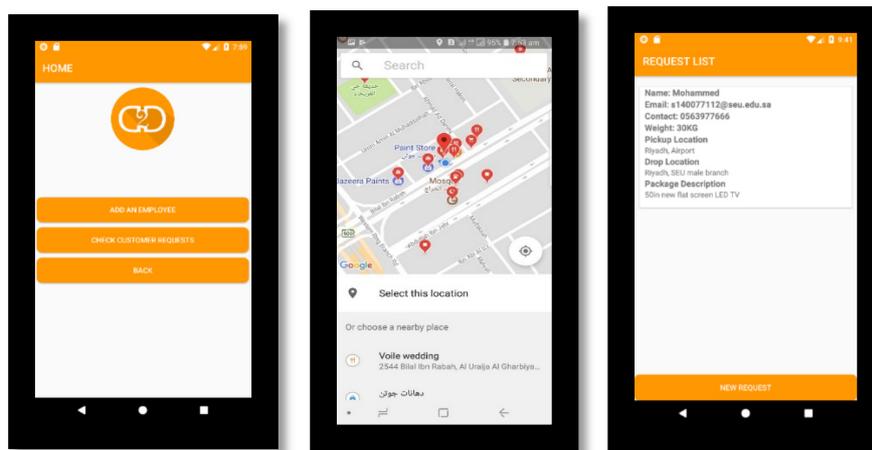


Fig. 8. The screens of the administrator Homepage, the list of New Requests, and the employee Homepage Screen

4.3 Software testing

Different tools and devices have been used to test the developed D2D delivery MobApp, such as Android Studio and mobile phones with Android operating system. Testing has been performed by using the Black Box testing, where the test cases were performed without any knowledge of the MobApp code. Testing was performed for all users, where different cases were performed. All services of the developed MobApp were tested to show their accurate functionality. In addition, regression test cases were performed to uncover any bug or defect in the existing functions of the MobApp after correcting the found errors. This section lists the conducted test cases related to the developed D2D product delivery MobApp functionality.

Table 1. Test Case 01: Customer Registration

Identifier	TC-1 (Customer Registration)
Priority	High
Related Requirement	UC-1 (Register)
Short Description	Customer Registration to D2D Product Delivery MobApp Test Case
Pre-condition	Welcome screen will splash, login homepage opens for selecting the user type, then at the user login screen, a button is pressed to register a new user.
Input data	User name, customer contact, customer email, customer new password
Detailed steps	The customer will add his name and contact number, email address and new password, then press signup button.
Expected results	The screen popups notifying a successful registration if the entered data is correct.
Post-condition	The user will click on login if registration is successful or update the entered information and click register again.

Table 2. Test Case 02: Customer Login

Identifier	TC-2 / Customer Login
Priority	High
Related Requirement	UC-1 / (Register)
Short Description	Customer Login to D2D Delivery MobApp Test Case
Pre-condition	The customer uses the registered email address to login to the service
Input data	Email address and password
Detailed steps	The customer enters the email address as registered and fill-in the password, then clicks sign-in button.
Expected results	The MobApp opens the customer homepage and a successful sign-in short message will appear to the customer screen.
Post-condition	Customer input valid information and click sing-in

Table 3. Test Case 03: Customer Create New Request

Identifier	TC-3 (Customer Create New Request)
Priority	High
Related Requirement	UC-2 (Create Delivery Request)
Short Description	Customer creates a new delivery request through the MobApp
Pre-condition	Customer logged-in to homepage.
Input data	Package type, pickup and drop-off location, package weight, and package description.
Detailed steps	Customer clicks the new request button, the new request layout view will appear, the customer fills-out the request form with package details, adds pickup and drop-off location and includes package description, and then customer clicks send request button.
Expected results	Short message appears with statement request sent successfully; the view will change to customer homepage.
Post-condition	Delivery request sent successfully.

Table 4. Test Case 04: Customer Review Request

Identifier	TC-4 (Customer Review Requests)
Priority	Medium
Related Requirement	UC-4 (Preview Request)
Short Description	Customer Review Requests List Test Case
Pre-condition	Customer has created a delivery request before viewing the request list
Input data	Click on requests button in homepage
Detailed steps	When customer clicks requests button, the MobApp opens requests list. The customer will scroll through the page to find a particular request details.
Expected results	The requests list appears and the list of all created requests are listed with the details of each request.
Post-condition	Details of all delivery request appear.

Table 5. Test Case 05: Administrator Add Employee

Identifier	TC-5 (Administrator Adds Employee)
Priority	High
Related Requirement	UC-2 (Create Delivery Request)
Short Description	Administrator Adds Employee Test Case
Pre-condition	Administrator logged-in to homepage
Input data	Employee name, employee email and contact, and temporary password.
Detailed steps	Administrator clicks add employee button, the add employee form will appear, the administrator adds the employee name, contact details, and the email address with temporary password to be change by employee upon first log-in. the administrator then clicks add employee button.
Expected results	A short message appears showing that the employee has been added successfully and the view will be changed to the administrator homepage
Post-condition	The employee has been added successfully.

Table 6. Test Case 06: Employee Log-in

Identifier	TC-6 (Employee Login)
Priority	High
Related Requirement	-
Short Description	Employee Log-in to D2D Delivery MobApp
Pre-condition	Administrator adds the employee to the employees list.
Input data	Valid registered email address and password
Detailed steps	When the employee accesses the MobApp main page, he/she clicks the employee access button, the employee login page will open and the employee will add the registered email address and password then click sign-in button.
Expected results	Employee homepage will display on screen.
Post-condition	The employee logged-in to the MobApp.

4.4 Discussion

The developed D2D product delivery MobApp successfully achieved all mentioned services that should be provided by it. The MobApp was developed with a friendly GUI and is ease to use, where the customer can easily create delivery requests and track the delivery process, and the administrator can perform and monitor the operations of the MobApp in a simple way. The performed test cases have discovered and corrected the

found defects and proved the correctness of the developed D2D delivery MobApp. Moreover, the MobApp was developed to operate in different operating environments and support different screen resolutions and networks. In addition, the quality attributes were taken into consideration to strength the developed application.

5 Conclusion and Future Work

This paper presented a MobApp that aims to enhance the delivery service experience in Saudi Arabia. The customers can use the developed D2D delivery MobApp to request delivery of endless list of permissible products. The proposed delivery MobApp allows its customers to create, update and track their delivery requests. The MobApp reduces the problems raised by using current ways, such as time and cost. The D2D delivery MobApp has been successfully designed and implemented and it offers a user-friendly layout and easy navigation. In term of the efficiency, security, and quality attributes, the developed MobApp is found useful and efficient. Some options for future works include the following: the proposed MobApp can be upgraded to support other smartphone's operating systems, such as Windows and iOS. In addition, the MobApp can support Arabic language to cover a wider range of customers in Saudi Arabia.

6 References

- [1] Amalfitano, et al., "A General Framework for comparing Automatic Testing Techniques of Android Mobile Apps," *Journal of Systems and Software*, 2016. <https://doi.org/10.1016/j.comcom.2016.04.026>
- [2] K.-W. Lim, et al., "Characterizing and predicting mobile application usage," *Computer Communications*, vol. 95, pp. 82-94, 2016. <https://doi.org/10.1016/j.jss.2016.09.049>
- [3] M. Usman, et al., "A product-line model-driven engineering approach for generating feature-based mobile applications," *Journal of Systems and Software*, vol. 123, pp. 1-32, 2017. <https://doi.org/10.1016/j.tele.2017.11.007>
- [4] W.-K. Tan, et al., "Smartphone application personality and its relationship to personalities of smartphone users and social capital accrued through use of smartphone social applications," *Telematics and Informatics*, vol. 35, pp. 255-266, 2018. <https://doi.org/10.1016/j.tele.2017.01.006>
- [5] A. Tarute, et al., "Mobile application driven consumer engagement," *Telematics and Informatics*, vol. 34, pp. 145-156, 2017. <https://doi.org/10.1016/j.cose.2016.11.007>
- [6] A. Feizollah, et al., "AndroDialysis: Analysis of Android Intent Effectiveness in Malware Detection," *Computers & Security*, vol. 65, pp. 121-134, 2017. <https://doi.org/10.1016/j.compeleceng.2018.07.052>
- [7] S. Li, et al., "Springbok: An App deployment accelerator for Android smart devices," *Computers & Electrical Engineering*, vol. 71, pp. 223-236, 2018.
- [8] J. Allen, et al., "An analysis of online shopping and home delivery in the UK," *Freight Traffic*, 2017.
- [9] R. McGrath, "Attitudes towards Home-Delivery and Takeaway Food in the UK," 2016.
- [10] UberEats. Available: <https://www.ubereats.com>. [Accessed September 2018]
- [11] Careem. Available: <https://www.careem.com>. [Accessed September 2018]
- [12] Shop and Ship. Available: <https://www.shopandship.com>. [Accessed September 2018]

7 Authors

Samer Atawneh received his PhD Degree in information security from Universiti Sains Malaysia (USM) in 2015. Currently, Dr. Atawneh is an Assistant Professor at the College of Computing and Informatics, Saudi Electronic University, Saudi Arabia. He has published several research papers in International Journals and Conferences with high reputation. His research interests lie in information security, steganography, software engineering and mobile applications.

Basil Al-Kasasbeh received his PhD Degree in Systems and Networks Engineering from Siberia State University of Telecommunication and Information Technology in 2001. Dr. Al-Kasasbeh is an Associate Professor at the Faculty of Computer Studies, Arab Open University, Saudi Arabia. He has published several research papers in International Journals and Conferences with high reputation.

Mohammed Ben Rshed received his BSc Degree in information technology from Saudi Electronic University in 2018. Currently, Mr. Ben Rshed works at Aramco, Saudi Arabia. His research interests lie in mobile applications.

Article submitted 2018-09-27. Resubmitted 2018-11-23. Final acceptance 2018-11-25. Final version published as submitted by the authors.