Design of Mobile Application for Travelers to Transport Baggage and Handle Check-in Process

Sara Y. Ahmed
Department of Information Technology,
Faculty of Computing and Information Technology,
King Abdulaziz University, Jeddah, Saudi Arabia
Department of Scientific Computing,
Faculty of Computer and Information Sciences,
Ain Shams University, Cairo, Egypt
PO Box 42808, 21551, Jeddah, Saudi Arabia

***Abstract*—In this paper, an Android based application called 'Baggage Check-in Handling System' is developed for helping travelers/passengers transport their baggage to the airport and handle the check-in process. It is merging the idea of online baggage check-in, and tracking technology together. The application is stimulated from the rapid growth of on-demand ride services, such as UberX and Lyft and the wide spread adoption of smart-phones. The proposed system enables travelers to make an appointment before the flight's take-off by requesting a driver to pick up the traveler's baggage to transport to the airport. Then, travelers can track the driver's location using Geographical Position System (GPS). Eventually after the check-in process, the driver will send a unique barcode provided for the baggage to travelers through the application. As a result, the traveler will have the choice of directly proceeding to the flight gate. The application is created for Android platform operating system, and developed in Java programming language using the Android software development kit (SDK). Additionally, data between database and server have been exchanged using phpMyAdmin. The application uses an authentication technique called Secure Hash Algorithm (SHA). This technique is designed to improve the scalability of authentication and reduce the overhead of access control.**

Keywords*—Baggage handling system; tracking technology; baggage barcode; android platform; Android software development kit (SDK); phpMyAdmin, Secure Hash Algorithm (SHA)

I. Introduction

In recent years, airport departure processes have drastically changed, as traveling became a necessity to many individuals. Most travelers/passengers face a lot of difficulties and complications during traveling, like being late or missing their flights, or wasting their time waiting for their baggage to be weighted. Airport check-in [1] is the process in which travelers are accepted by an airline at the airport prior to travel. The airlines typically use service counters found at airports. The check-in is normally handled by an airline itself or by a handling agent working on behalf of an airline. Travelers usually hand over any baggage that they do not wish or are not allowed to carry in to the aircraft's cabin and receive a boarding pass before they can proceed to board their aircraft.

Check-in is usually the first procedure for a traveler when arriving at an airport, as airline regulations [2] require travelers to check in by certain times prior to the departure of a flight. This duration spans from fifteen minutes to four hours depending on the destination and airline.

The main problem is standing for a long time in the baggage's queue check-in counter at the airport [2], and then waiting until boarding time which consumes a lot of time and stress. Time-consumption and stress are two of the many consequences of waiting. Moreover, traveling is harder for people who are older, pregnant, or accompanied by their children. From the airport operators' perspective, there are limited terminal space and capacity, along with optimizing income problems [3].

The aim of this paper is to introduce the implementation and design of on-demand service application called 'Baggage Check-in Handling System' that uses technology in order to make the check-in process swift and less time consuming for the traveler. 'Baggage Check-in Handling System' application is stimulated from the developments in information and communication technology [4], which have enabled new advantages that offer a wide range of real-time and demand-receptive trips.

Companies such as Lyft and Uber [5], have developed smart-phone applications to connect riders with drivers. The transport request indicates at least the pickup or drop-off location through Geographical Position System (GPS) [6]. A confirmation panel is created to present information equivalent to either the pickup or drop-off location. Through these applications, confirmation and cost of transportation, and a rating system are provided. Users' credit card information can be saved within the system to facilitate future trips. Complete description of ride sourcing is challenging though, as the services are quickly growing [7].

The contribution of our mobile phone application, 'Baggage Check-in Handling System', is that it provides transport services for the travelers' baggage after authenticating its information. The proposed system enables travelers to make an appointment before the flight's take-off, by requesting a driver to pick up the traveler's baggage from the selected location to the airport. Meanwhile, travelers can track their baggage through Geographical Position System (GPS) [8]. Eventually after the check-in process, the driver will send a unique barcodes for the baggage to the traveler through the application. As a result, the traveler will have the choice to directly proceed to the flight's gate. Many technologies are used in our application such as the flight's booking information technology which is used to retrieve the showing booking number, or by an e-ticket. Moreover, the application is created for Android platform operating system [9], and developed in Java programming language using the Android software development kit (SDK) [10]. Finally, data between database and server have been exchanged using phpMyAdmin [11].

In fact, identifying and authenticating users can not only avoid illegal accesses, it can also reduce unnecessary redundancies of access control. 'Baggage Check-in Handling System' uses an authentication technique introduced in [12] called Secure Hash Algorithm (SHA). This algorithm was published in 2003 as the secure hash standard. SHA is a cryptographic hash function and employed in several widely used applications and protocols. This algorithm is designed to improve the scalability of authentication and reduce the overhead of access control. It is used for handling a compressed representation of a message. Given an input message, SHA produces an output called the message digest. It is claimed to be secure because it is infeasible to compute the message corresponding to a given message digest. Also, it is particularly improbable to find two messages hashing to the same value.

The remainder of the paper is organized as follows: Section II describes the proposed application scenario. The technical details to implement the suggested system and the data collection in the process of analyzing the project to gather user information is introduced in Section III. The implementation of the system, and the tools used to implement the features are presented in Sections IV and V. The usability study is introduced in Section VI. Finally, conclusions and future works are drawn in Section VII.

II. Application Scenario

The central goal for the proposed application 'Baggage Check-in Handling System' is to create a marketable mobile application based on people's needs in handling the baggage check-in process. The application consists of two main users, the driver, and the traveler. The application's scenarios for both users are introduced in Fig. 1.

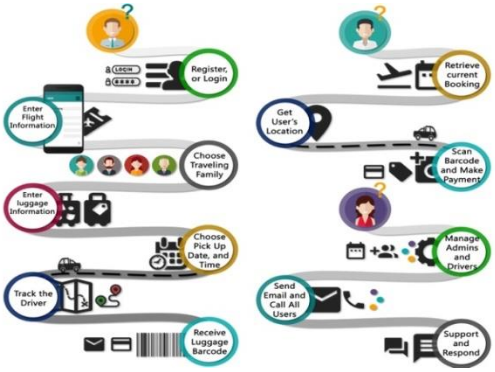

Fig. 1. Application scenarion for traveler and driver.

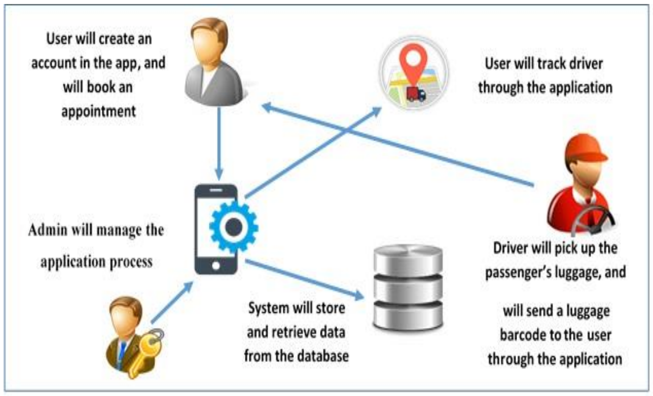
User will create an account in the app, and will book an appointment
User will track driver through the application
Admin will manage the application process
System will store and retrieve data from the database
Driver will pick up the passenger's luggage, and will send a luggage barcode to the user through the application

Fig. 2. High-level system architecture.

A. High-Level System Architecture

Fig. 2 introduces the high-level system architecture [13] for the proposed application, in which the traveler creates an account with all his/her and families' information. To book the service, the flight will authenticate then it will have at least three hours prior the flight. During booking, the driver will pick up the baggage while the passenger is able to track him until the check-in point. After a successful check-in process, the service's cost will be withdrawn from the passenger's pre-defined bank account, and then receive the baggage's barcodes that will be saved in his/her account in case of any problem. Finally, the passenger will proceed directly to the flight gate. All relevant information related to the traveler, and flight information will be verified through the application.

B. The Behavior and Functionality of the Application

A use case [14] shows the behavior or functionality of a system (see Fig. 3). It consists of a set of sequences of interactions between a system and a user in an environment. The first use case is the traveler login. Registration - as a new traveler - is an extended case from it. The second use case is the traveler's reservation of the pick-up baggage service. Including that, he/she will enter the booking information, select a time from the available time slot, and approve the policy. In the driver's interface, a schedule with the traveler's information, time, and location is displayed. The third use case is the traveler's ability to track the driver. Following that, the fourth use case, the driver scans the barcode then saves it and sends it to the traveler. The final use case allows the admin to log in into his/her interface, and manage the system with adding, editing, removing drivers, users, and admins; and updating the booking and scheduling to the driver.

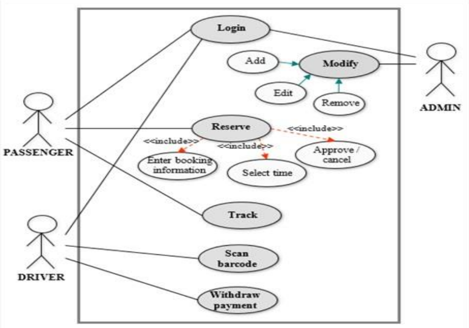

Fig. 3. Application use case.

III. Analysis and Design

This section introduces more technical details to implement the suggested system and the data collection in the process of analyzing the project to gathering user information.

A. Data Collection Techniques

The user profile questionnaire [15] includes the user's general information: age, gender, and their evaluation in using similar systems to the application. It also includes their experiences in traveling, the time they spend to check-in their baggage before their take off. Fig. 4 depicts the analysis of the questionnaire created using "Typeform" website, with a thousand and twenty-three responses. Most people worry about the security side of this application and how much trust they can put into it. Some people request for the application to have the government's permission, and that their confirmation should be displayed in the users' signing up process. Other people suggest the having a concise explanation of the procedure of the baggage's insurance in case it gets lost or harmed would make the application guaranteed, trustworthy, and reliable. The remaining statements recommend more services and techniques such as providing a video camera inside the car to watch the driver's movements, taking care of breakable/fragile baggage, adding a baggage packing service, and displaying the baggage's weight to estimate if extra payment is required.

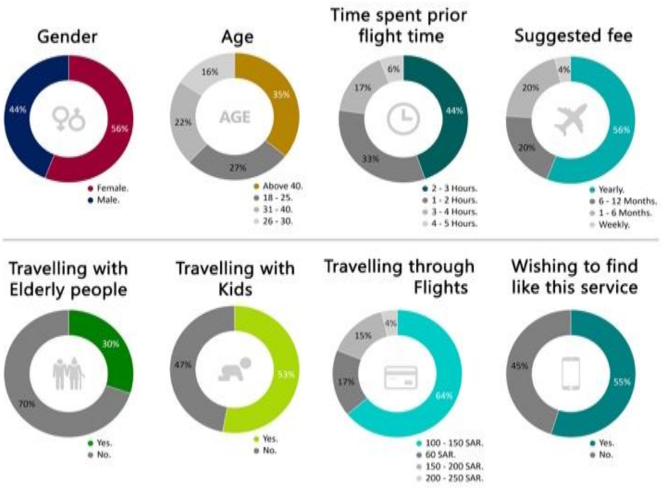

Fig. 4. Questionnaire analysis.

B. Functional Requirements

Functional requirement [16] defines the function of a system and its component. Table I represents the function as a set of inputs, outputs, and behavior for each actor.

C. Non-Functional Requirements

In requirement engineering, a non-functional requirement [16] is a necessity that specifies criteria that can be used to judge the operation of a system. The non-functional requirements of the application are introduced in Table II.

TABLE I. Functional Requirements

	No.	Requirement
Create main admin account	**1**	When using the system for the first time, an admin account should be created.
Admin	**1**	The admin shall have the ability to login to his/her account.
	2	The main admin shall have the ability to add, delete, and update drivers, and sub-admins.
	3	The admin shall have the ability to add, delete, and update users (travelers).
	4	The admin shall be able to adjust the booking schedule, and assign appointments to drivers.
	5	The admin should be able to send confirmations to the travelers via email, or messages.
	6	The admin should have the ability to view the user's feedback.
User (Traveler)	**1**	The user shall have the ability to create a new account and login to his/her account.
	2	The user shall have the ability to reserve a driver after filling his information.
	3	The user shall upload his/her boarding pass.
	4	The user should book within the time range.
	5	The user should be able to add extra services to his baggage such as wrapping, and/or breakable.
	6	The user shall view and approve the price decided by the system, and approve the policies.
	7	The user shall be able to cancel his/her reservation if the arrival time had not come yet.
	8	The user should leave his/her review of the application at the end of the process.
System	**1**	The system shall be able to generate the users' flight information.
	2	The system shall be able to assign appointments to different drivers.
	3	The system shall calculate the price.
	4	The system shall send a confirmation message to the travelers after their successful booking.
	5	The system shall send the driver's information.
	6	The system shall upload and save the boarding pass of the traveler to the driver.
	7	The system shall provide the user with the tracking of the driver.
	8	The system should notify the user about the driver's status via sending the user notifications.
	9	The system shall send and save the barcode tags to the users' profile.
	10	In case of user cancelation, the system should cancel and reassign the reservation to the driver.
	11	The system shall withdraw the predefined cost at the end of the check-in.
	12	The system should display the user's feedback.
Driver	**1**	The driver shall have the ability to login to his account.
	2	The driver should confirm that he picked up the baggage.
	3	The driver should specify his status, either if he is on the way, picked-up the traveler's baggage, arrived at the airport, or done with the check-in.
	4	The driver shall scan the barcode tags after checking in the baggage, and then send them to the traveler.
	5	The driver should shift to the appointment.

TABLE II. NON-FUNCTIONAL REQUIREMENTS

No.	Requirement	Description
1	**Usability**	- The application must be easy and simple for all types of people that vary from ages and backgrounds to use.
2	**Response Time**	- Notifications should be sent in an appropriate time. - Booking and scheduling must be implemented in the right timing, with no delays.
3	**Reliability**	- Many users can use the application at the same time, and deliver services to all users, as it was intended to.
4	**Security**	- The application contains the users' confidential data that cannot be seen or accessed into by anyone, except the users themselves, and the admin. - The application should be supported by a trusted organization. - The drivers' car will be tracked for security and safety issues.
5	**Maintainability**	- New features could be added to the application in the future.
6	**Warranty**	- The service should compensate the users' money in case any of his/her baggage got damaged or lost.

D. Software and Hardware Requirements

‘Baggage Check-in Handling System’ will be implemented using Android OS application. The requirements (shown in Fig. 5) are an Android Operating System, a Software developed by a Java programming language, Global Positioning System, and SQL database to store data whereas the hardware requirements are Android smart-phones with the Android operating system and Server.

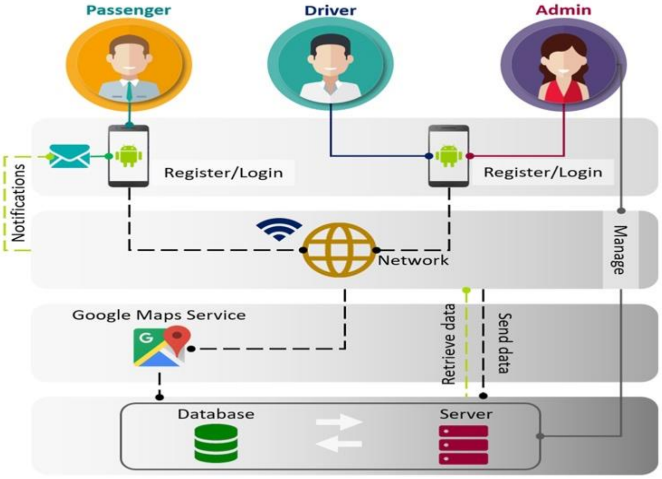

Fig. 5. Software requirements.

IV. APPLICATION METHDOLOGY

The application uses the Agile Development Methodology [17], since Agile is an Incremental Software Development Method. Furthermore, it advocates adaptive planning, evolutionary development, early delivery, and continuous improvement. It also encourages rapid and flexible response to change [17].

A. High Level Diagram

High-Level Design (HLD) explains the architecture that would be used for developing a software product [18]. Fig. 6 provides an overview of an entire system identifying the main components that would be developed for the product and their interfaces. We overviewed possible scenarios that will be followed between the main components of our system. The first scenario is for creating a new account for a traveler and validating inserted information with application database. The second scenario includes booking appointments for picking up baggage and saves the time and location into the application’s database, then receives the saved barcode from it. The third scenario is retrieving the travelers’ information such as their family name and phone number then saving it in the database. Finally, after the driver receives a booked appointment’s information from the database, he uploads the scanned barcodes to the application’s database.

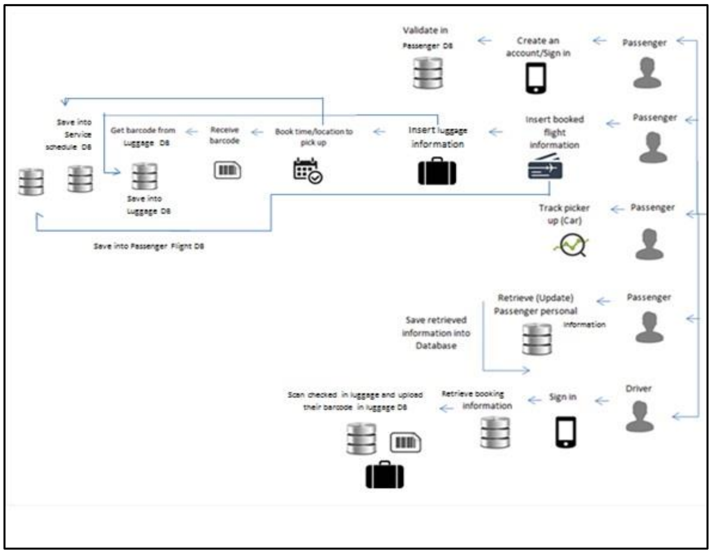
Validate in Passenger DB
Create an account/Sign in
Passenger
Save into Service schedule DB
Get barcode from Luggage DB
Receive barcode
Book time/location to pick up
Insert luggage information
Insert booked flight information
Passenger
Save into Luggage DB
Save into Passenger Flight DB
Track picker up (Car)
Passenger
Retrieve (Update) Passenger personal information
Passenger
Save retrieved information into Database
Scan checked in luggage and upload their barcode in luggage DB
Retrieve booking information
Sign in
Driver

Fig. 6. High level design.

B. Sequence Diagram

The sequence diagram [19] in Fig. 7 illustrates ‘Baggage Check-in Handling System’ and shows the process followed between the system object. The traveler shall have an account by signing up into the application. Once the account has been created, the traveler will log into the application. After that, he\she will enter the booking information, and choose the appropriate time from the available time slot for booking. The driver will come at the specified date and time and pick up the baggage then the traveler will track his/her baggage through the map in the application until the driver checks them in. Drivers will have an account given by the admin. After that, the driver can sign in and a find booking schedule in his account and the traveler’s information for each booking. The admin is given the ability to modify the traveler’s, driver’s and new admin’s information, and edit all the appointments and assign them to the drivers.

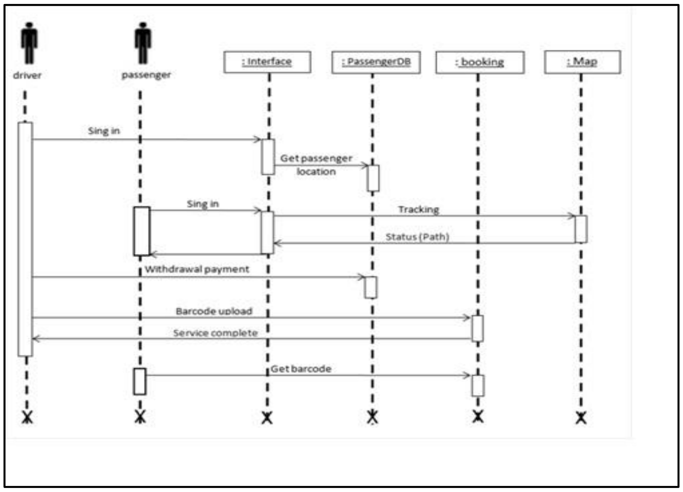

Fig. 7. Application sequence diagram.

C. Class Diagram

Mainly, Fig. 8 shows six classes of the system which are: Admin, Traveler, Driver, Appointment, Baggage, and Tracking.

login (): A function that verifies the username and password of travelers, and drivers.

addFamily (): Allows the registered traveler to add his/her family members.

setBooking (): Allows the travelers to reserve the service by setting time, date, and location.

setTrackRoute(tRoute): Allows the system to set the actual route into the system.

setScanBarcode (): Allows the driver to scan the baggage number and barcode, then send it to the traveler.

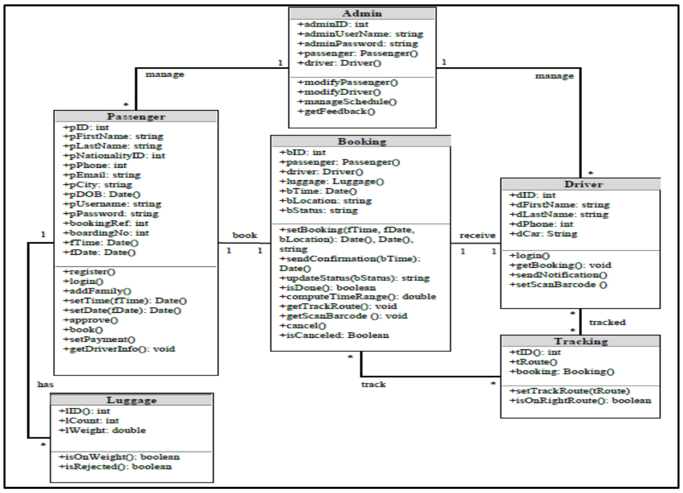

Fig. 8. Class diagram.

V. Application Implementation

This section is about the implementation of the system, and the tools used to implement tasks, such as; GPS (Global Positioning System), the database, and other functions. In some parts, android libraries are used to support and improve the functionality.

A. Objectives

The success criteria [20] of the proposed application is to minimize the possibility of missing a flight because of the time taken to check-in, also to relieve travelers from the check-in stress and decrease their effort in trying to fit in time to check-in.

B. Implementation Tools

The implementation tools that are used in the proposed application are Android Studio tool, the official integrated development environment (IDE) for Android platform development, Geographical Position System (GPS), phpMyAdmin to handle the administration of MySQL using a web browser (It can perform various tasks such as; creating, modifying or deleting databases, tables, fields or rows; executing SQL statements; and managing users and permissions) , and Graphical User Interface Tools were used which are Java programming language made for defining and activating the XML layout (The XML language used to define the main structure of the interface). The authentication technique called Secure Hash Algorithm (SHA) is used to authenticate users and to reduce the overhead of access control, it is a cryptographic hash function used for handling a compressed representation of a message. Also, it is particularly improbable to find two messages hashing to the same value.

C. Implementation Process

- Creating database tables with all relations and constrains.
- Setting the phpMyAdmin files to manipulate the database from the android studio code.
- Designing the layout of the application interfaces.
- Prepare the code for each layout.
- Filling the database with information.
- Authentication of users' information.
- Testing the application.

D. Results and Discussions

As shown in Fig. 9, the screens of the 'Baggage Check-in Handling System' are vertically positioned, with large icons and clear text. The screens reveal information about the application scenario in which during the booking's date and time the passenger will be notified by the status in the form of a notification, beginning with the driver's arrival until the successfully checked in baggage. When a passenger books an appointment, the specified driver will be notified then decides whether to approve or decline the appointment. Once the driver picks up the baggage, the passenger will track him until he reaches the airport. Passengers can contact the driver by clicking on "Contact driver". When the baggage is checked in, the driver will scan the barcodes through the application, then it will be saved to the passenger's profile.

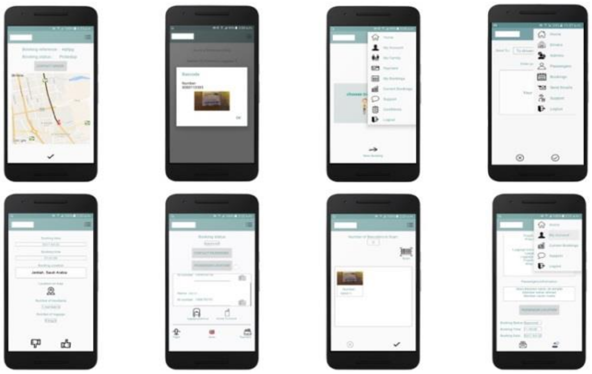

Fig. 9. Some screens of the ‘Baggage Check-in Handling System’.
(a) Tracking the Driver. (b) Baggage Barcode Number. (c) Traveler Menu.
(d) Driver Menu. (e) Grid View of all Bookings. (f) Booking Status.
(g) Scanning Barcode. (h) Traveler Information.

VI. USABILITY STUDY

Usability study is a way of seeing how easy it is to use the application by testing it with real users [21]. Users are asked to complete tasks while being observed to see where they encounter problems and experience confusion, and to check if it meets the user's expectations. If more users encounter similar problems, recommendations will be made to overcome these usability issues. In order to evaluate the usability of the proposed system, a usability study is conducted in which five participants were asked to perform a set of tasks. The focus was on the main features of the proposed system, which are registering as a new user, logging in, booking an appointment, adding a new family member, viewing booking information, tracking the driver, and adding a complaint. As results, it was observed that the executions of the different tasks revealed the ease of using the application's main functionalities.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we are seeking to provide an airport check-in service for baggage through a mobile application. This service allows a traveler to make an appointment before the flight’s take off by about three to five hours, by requesting a trusted driver to pick up the passenger's baggage from the selected location to the airport, so that the passenger could have the choice of directly proceeding to the flight’s gate. The application is developed for all passengers but especially for people that get tired easily like elderly people, pregnant women, and families with heavy baggage. The application is created for Android platform operating system, and developed in Java programming language using the Android software development kit (SDK). Furthermore, the authentication of users is done using Secure Hash Algorithm (SHA). The results reveal the ease of using the application's main functionalities. We know that security is the main concern when using the application. The future works are supporting multiple operating systems rather than only Android, providing multiple languages, and being sponsored and adopted by a wide company to solve the lack of security.

ACKNOWLEDGMENTS

I would like to thank the students Shahad Sait, Sarah Al-Ghamdi, Amna Moussa, Shorouq Basnawi, and Mona Abdulbaqi of King Abdulaziz University, KSA, for their contribution in the development of the ‘Baggage Check-in Handling System’.

REFERENCES

[1] S. Jain, R. R. Creasey, J. Himmelspach and K. P. White, "Check-in Processing: Simulation of Passengers with Advanced," in Winter Simulation Conference, AUSTRALIA, 2011.

[2] Gillen, David and W. G. Morrison, "Aviation security: Costing, pricing, finance and performance.," Journal of Air Transport Management , vol. 48, pp. 1-12, 2015.

[3] Tuan. and T. D. Cao, "Improving travel information access with semantic search application on mobile environment," in In Proceedings of the 9th International Conference on Advances in Mobile Computing and Multimedia, New York, NY, USA, 2011.

[4] A. Smirnov, A. Kashevnik, N. Shilov, N. Teslya and A. Shabaev, "Mobile application for guiding tourist activities: tourist assistant-tais," in Open Innovations Association (FRUCT16), 16th Conference , 2014.

[5] MacMillan and Douglas, "Uber touts its employment opportunities," The Wall Street Journal, 2015.

[6] CIRSTEA and ALICE, "The Implications of Mobile Commerce Applications," International Journal of Scientific Knowledge, vol. 6, no. 2, May 2015.

[7] Adewumi and Adewole, "Developing a mobile application for taxi service company in Nigeria," in 2015 International Conference on Computing, Communication and Security (ICCCS), 2015.

[8] P. Verma and J. Bhatia, "Desing and Development of GPS-GSM Based Tracking System with Google Map Based Monitoring," nternational Journal of Computer Science, Engineering and Applications (IJCSEA) , vol. 3, no. 3, pp. 33-40, 2013.

[9] V. S and Kirthika.B, "Android Operating System: A Review," International Journal of Trend in Research and Development, vol. 2, no. 5, pp. 260-264, 2015.

[10] G. Pandey and D. Dani, "Android Mobile Application Build on Eclipse," International Journal of Scientific and Research Publications, vol. 4, no. 2, 2014.

[11] P. D. Team, "PHP: Security and Safe Mode - Manual," http://php.net/manual/en/features.safe-mode.php.

[12] Wade Trappe, Lawrence C. Washington.. *Introduction to Cryptography with Coding Theory*. New Jersey: Pearson Prentice Hall, 2006.

[13] A. Finkelstein, J. Kramer, B. Nuseibeh and L. Finkels, "Viewpoints: a framework for integrating multiple perspectives in system development.," International Journal of Software Engineering and Knowledge Engineering, pp. 31-57, 1992.

[14] Davis and A. M., Software Requirements: Objects, Functions, & States, Prentice-Hall, 1993.

[15] Hausman and Angela, "A multi - method investigation of consumer motivations in impulse buying behavior," Journal of Consumer Marketing, vol. 17, no. 5, pp. 403-426, 2000.

[16] Cleland-Huang and Jane, "Toward improved traceability of non-functional requirements," in TEFSE '05 Proceedings of the 3rd international workshop on Traceability in emerging forms of software engineering, California, 2005.

[17] Qusef, A. De Lucia and Abdallah, "Requirements Engineering in Agile Software Development," Journal of Emerging Technologies in Web Inteliigence, vol. 2, no. 3, August 2010.

[18] Pressman and R. S., Software Engineering: A Practitioner's Approach, 2005.

[19] JingLiu, JoshDehlinger and RobynLutz, "Safety analysis of software product lines using state-based modeling," Journal of Systems and Software, vol. 80, no. 11, pp. 1879-1892, 2007.

[20] NitinAgarwal and UrvashiRathod, "Defining ‘success’ for software projects: An exploratory revelation," International Journal of Project Management, vol. 24, no. 4, pp. 358-370, 2006.

[21] M.MorenoaMaria and IsabelSanchez, " Analysing the impact of usability on software design,” Journal of Systems and Software,vol. 80, no. 9, pp. 1506-1516, 2007.