

Optical Character Recognition Functionality Introduction in Mobile Application for Car Diary

Ioannis Patias

Faculty of Mathematics and Informatics, University of Sofia "St. Kliment Ohridski", BG-1164 Sofia, Bulgaria

Abstract: The purpose of the paper is to develop a mobile Android application—"Car Log" that gives to users the ability to track all the costs for a vehicle and the ability to add fuel cost data by taking a photo of the cash receipt from the respective gas station where the charging was performed. OCR (optical character recognition) is the conversion of images of typed, handwritten or printed text into machine-encoded text. Once we have the text machine-encoded we can further use it in machine processes, like translation, or extracted, meaning text-to-speech transformed, helping people in simple everyday tasks. Users of the application will be able to enter other completely different costs grouped into categories and other charges. Car Log application quickly and easily can visualize, edit and add different costs for a car. It also supports the ability to add multiple profiles, by entering data for all cars in a single family, for example, or a small business. The test results are positive thus we intend to further develop a cloud ready application.

Key words: Optical character recognition, mobile application, car diary.

1. Introduction

The main focus of the application is in recognizing text in a photo of a cash receipt. OCR (optical character recognition) is a technology for converting handwritten or printed text into machine-encoded text. It is widely used to file data from printed-paper, including personal documents, invoices, bank statements, computer prints, etc. This method allows us printed texts to be edited with a text editor, stored more compactly, displayed online, and used in computer programs such as automated translation or text-to-speech conversion.

OCR is applied in many applications. The main reason for the development of such applications is the weaknesses of the paper carrier as a means of storing information. The more paper is growing, the more productivity falls. It is becoming increasingly difficult for information to be found, and its use becomes slow and inefficient. Paper information cannot be edited at the moment, as it would be possible to work with a

text file on a computer. All these factors lead to the introduction of OCR, which allows the retrieval of obsolete, expensive and hard-to-handle paper and the fitting of text able formats that can be easily found, used and edited.

The process from the end-user's point of view is very simplified, as the user only scans the document, by taking a photo. Car Log app loads the scanned file into the OCR platform, and saves the recognized file in the selected format. We will describe in details the process latter.

Looking in Google Play Store [1] we may find some applications with similar functionalities. We compared some of them (Fuel Manager, Fuel Buddy - Car mileage Log, and Car Diary) to our application (Car Log). The apps are compared to some of the more important features of similar type of applications, such as: support to more than one vehicle, addition of charges and other type of expenditures, calculation of average cost and related information, number of charges and other similar features affecting fuel consumption, simple and accessible interface, as well as the innovative functionality in the Car Log, namely, OCR for the conversion of the information from a

Corresponding author: Ioannis Patias, Ph.D., Chief Assist. Prof., research fields: artificial intelligence, embedded systems, database systems.

photo of the charge receipt. The comparison results are grouped in Table 1.

The main contribution of the paper is to show an easy way to build Android [2] mobile applications with OCR functionality.

2. Car Log Application Description

For the purpose of the demonstration, the Car Log application will be described in details, following the structure.

2.1 Description of the System Model

The application is based on triple-layer architecture. It is divided into three linked, independent layers (Fig. 1):

- Presentation layer—represents the user interface for interaction with the user. Performs the actions of displaying and collecting information;
- Business layer—represents the business logic of the application. Performs actions, related to data

processing, reading and recording in the database;

- Data storage layer—represents the database in which the information in the application is stored.

2.2 Description of the Database and the Application

For the needs of the “Car Log” application, SQLite [3] is appropriate because of its simple interface, speed and flexibility. The complete SQL database containing tables, indexes, triggers and views is represented by a single file on the disk. The designed database consists of 4 tables. every table is stored the data for the basic functional units of the application. The relations and dependencies between the objects, required for the proper operation of the application, are defined in advance in the following ER diagram (Fig. 2).

(1) The Car table contains the car profiles. It contains the following fields:

- id—a unique field for a profile
- brand—brand of the car

Table 1 Comparison of similar apps functionalities.

	Fuel manager	Fuel buddy—Car mileage Log	Car diary	Car Log
Support to more than one vehicle	-	+	-	+
Addition of charges and other type of expenditures	+	+	+	+
Calculation of average cost and related information	+	+	-	+
Simple and accessible interface	-	-	+	+
OCR for the conversion of the information from A photo of the charge receipt	-	-	-	+

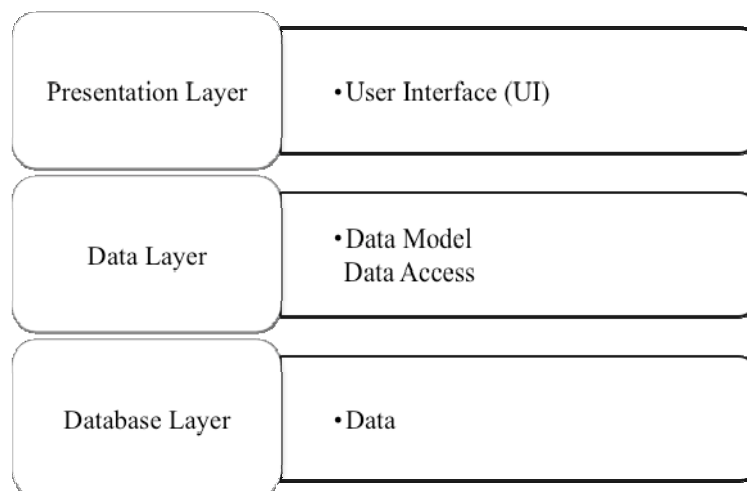


Fig. 1 Triple-layer architecture application.

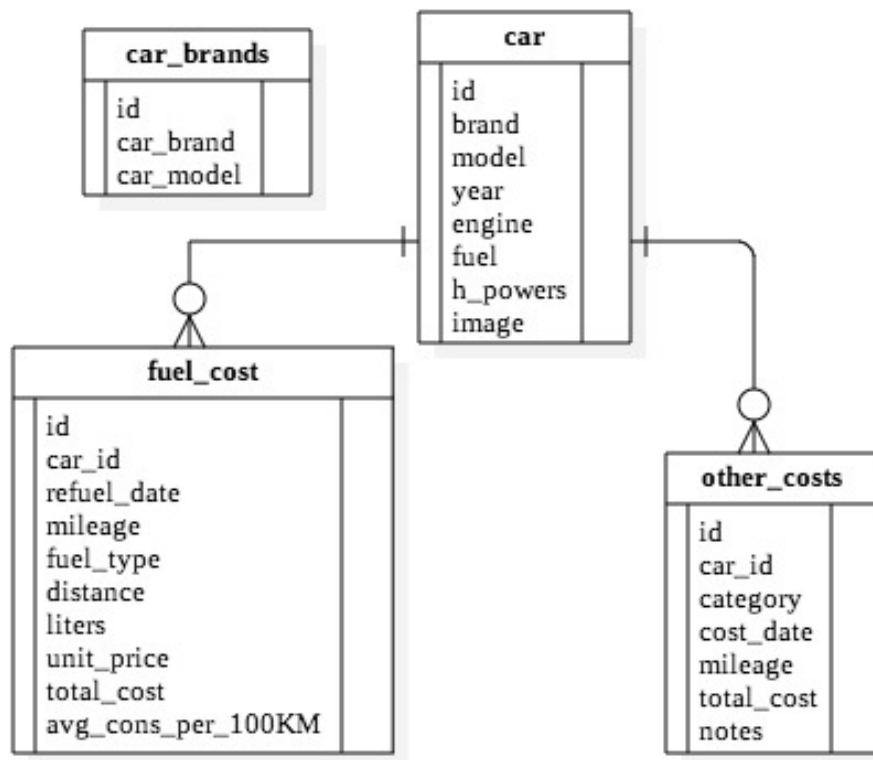


Fig. 2 ER diagram for the database used.

- model—vehicle model
- year—year of vehicle production
- engine—information about the vehicle engine
- fuel—information about the vehicle fuel type
- h_powers—vehicle's power
- image—a path that points to a file located in the file system representing a car image

(2) The fuel_cost table contains information about the entered charging for a vehicle, which is written in the following fields:

- id—unique field for the given load
- car_id—an external key pointing to a record in a car table
- refuel_date—represents the date of loading
- mileage—mileage of the car
- fuel_type—type of fuel loaded
- distance—distance traveled since the last load
- liters—amount of fuel in liters
- unit_price—unit price per liter of fuel
- total_cost—total charge for the entire load
- average_cons_per_100_km—average charge for

the specific load

(3) The other_costs table contains information about other costs incurred for a given car, which is listed in the following fields:

- id—a unique field for that extra cost
- car_id—an external key pointing to a record in a car table
- category—consumption category
- cost_date—represents the expense date
- mileage—mileage of the car
- total_cost—cost of the expense
- notes—spending notes

(4) Table car_brands is a table with only three fields. Vehicle and car model records have been entered in advance. The car_brand and car_model fields are TEXT type, car_brand cannot be NULL, but car_model can because the table is used in the business layer and there is an option to add a brand-only entry.

2.3 Implementation of the OCR Functionality

OCR in the Car Log application is implemented

through a server that accepts as input parameter the image in which the text is searched and returns a text file containing the recognized text to the image. The server used is the Abbyy Cloud OCR SDK [4]. This is a service, which is platform-independent, due to the

fact that it is accessible via the Web API and does not work on a device. The following activities diagram (Fig. 3) describes the process.

Of course there are various templates for receipts layout from the market, and the application needs to

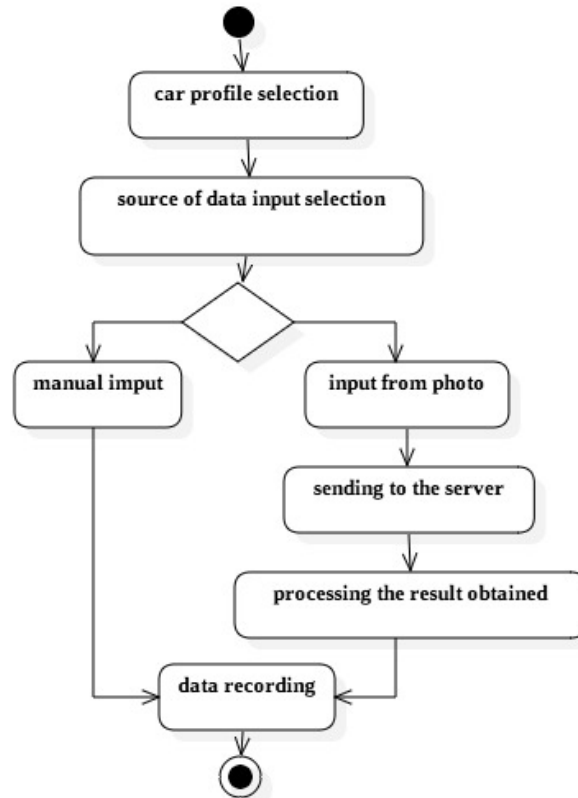


Fig. 3 Activities diagram of the OCR functionality.

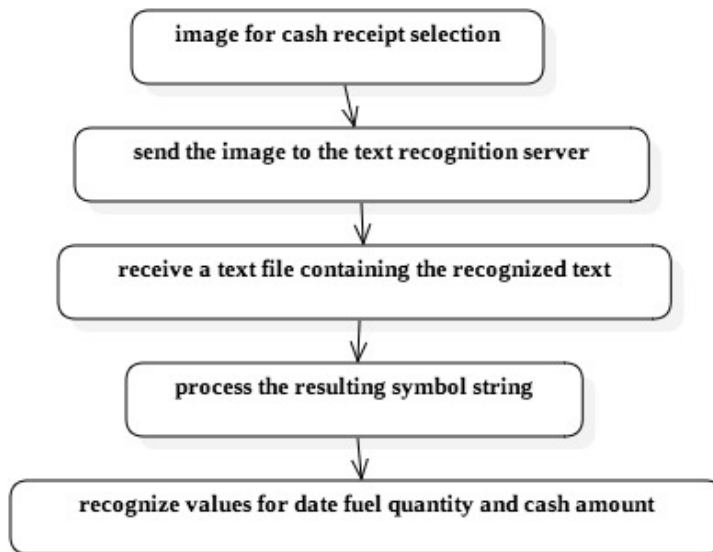


Fig. 4 Steps for various templates for receipts layout recognition.

take the layout differentiations into consideration. For the purpose the steps described in following activities diagram (Fig. 4) are used.

3. Test Results

The conducted experiments and tests confirm that the Car Log application fulfills the defined goals and objectives. The application successfully maintains a database that stores fuel consumption information entered by the user, as well as other types of costs. The car's diary successfully copied and recognized text into cash receipts, from photos from the petrol stations.

The application was also subjected to a serious test related to the location of the camera in relation to the cash register. The following tests were performed:

(1) Photo of the cash receipt at a corner—in this case with all the receipts provided the result was the same, namely, the application did not cope with the recognition of the symbols and words in the photos and thus returned a failure recognition message;

(2) Long distance photo—with these tests bit of contradictory information was obtained—some images did not pass successfully, while in some others the application successfully read the information. Probably the cause of this difference is not in the distance, but rather in the image quality and more particularly the existence or the lack of focus;

(3) Photo in dark with the phone's flash—here the application has done also controversially. Some images were read perfectly, while in some others the only recognized information was the date; and

(4) Photo with lower light without the phone's flash—in this case there is no definite result—some images were successfully processed, some others not.

Test results show that the app performs fairly well, even when the photo of the receipt is not perfect. Only

in the case of corner-snap tests the application categorically failed to process the photo, but in all other cases there was at least 50% success rate. When trying with normal photos taken in daylight and from a relatively close distance, the app did more successfully.

4. Conclusions

By using the Car Log app, we can quickly and easily visualize, edit, add different costs for a car. The application supports the ability to add multiple profiles, meaning we can enter data for all cars of a family, or a small business. The interface is extremely accessible and intuitive for work. The application can be further developed in the future, and first of all, the database could not just be local, but could be uploaded to a cloud server. In this case, the application can be logged in, on any vehicle data, and thus, by multiple devices. Also, for the case we examined we use the technique of calling the server to do the OCR, but as further development we could incorporate the OCR in the application. The application is useful, thus further developments should be considered, especially related to development of a cloud ready application.

Acknowledgements

This paper is supported by the project ДН 02/9-17.12.2016.

References

- [1] <https://play.google.com/store/search?q=car%20diary&c=apps> (April 2017).
- [2] Phillips, B., Stewart, C., and Marsicano, K. 2013. *Android Programming: The Big Nerd Ranch Guide*. (3rd Edition) ISBN-13: 978-0134706054, ISBN-10: 0134706056.
- [3] <https://developer.android.com/reference/android/package-summary.html> (2017).
- [4] <http://ocrsdk.com/documentation/apireference/processImage/> (2017).