Volume 6, Issue 10, October 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

"Easy-Reload"- An Android Application to Reload a Phone Easily

S. Babiththira & Ivantha Guruge
Faculty of Computing, Sri Lanka Institute of Information Technology,
Sri Lanka

Abstract: ***The aim of this project is to create an android application to reload a phone easily. Nowadays reload to a phone is very important to most of the people in Sri Lanka. When mobile users reload to a phone by the recharge card sometimes mobile users can enter the number wrongly. When mobile users recharge from a shop retailer can also enter mobile users' number as wrong. So in these situations mobile users lose their money. Sometimes mobile users didn't notify that what is the balance in their phone. In this situation if mobile users want to make an immediate call the balance is not enough. This system designed to avoid these problems. Only scratch the card then through application the phone automatically read the card numbers and reload it automatically. If mobile users want to save a reload card number in case of emergency mobile users can read the card number with the help of optical character reading after reading the card number it save in the phone itself if the phone balance is low the phone is automatically reloaded with the card number. In this application character recognizing done by OCR Tesseract technology.***

Keywords: Optical Character Reading-OCR, Android, Tesseract, Mobile, Reload, Prepaid.

I. INDRODUCTION

Topping up a mobile account is always been a trouble to mobile subscribers all over Sri Lanka. Subscribers who wish to add credit to their account either has to buy a scratch card or use the reload feature from a local store.

Most of the users do make mistake while keying in the number from the scratch card. Some might not even know the process of toping it up. Adding, people who are visually impaired face this difficulty the most. When users recharge from a shop the retailer can also enter the users' number as wrong. So in these situations users lose their money. In some situation users didn't notify that the balance credit of their phone. In this situation if users want to make an immediate call the balance is not enough.

The project designed and implemented as an android application named as "Easy-Reload" to quote the above complications.

With the help of optical character reading "Easy-Reload" application will reload a phone easily. First scratch the recharge card then take a photo of the card application read the card and reload the amount according to the service provider. If users want to save a reload card number in case of emergency users can read the card number with the help of "Easy-Reload" after reading the card number it save in the phone itself if the phone balance is low the users can use the saved recharge card number.

The system useful to prepaid connection users.

- ✓ The main feature of the application is scanning prepaid card number and Process the card number and application select the carrier based on the number format and send a reload request based on the carrier.
- ✓ Save the card number for future use. Saving the reload card number facility use to the customer when the phone balance is low.
- ✓ Provide facility to Share the balance manually by selecting the contact from contact list.
- ✓ And set a Periodic reload function (e.g.:- 9.00am every day). Through this feature users can set a periodic reload function which is reload the phone in every day or twice a day or once a week.

Flowing will be the research questions

- ✓ How Optical Character Reading could be developed for read the numbers?
- ✓ How to provide output with minimal error-rate?
  How is it possible to reduce the errors from the input?
- ✓ How to develop server based application with quick response time?
- ✓ What would be the mechanism that can be used to automatic reload?

II. BACKGROUND

Nowadays reload to a phone is very important to most of the people in Sri Lanka. To reload the phone users use the reload card and manually enter the card number to reload their phone. The users go to a retail shop from there they will reload their phone.

There is no specific application which resembles "Easy-Reload" directly; some of these OCR apps which were found on Google's Play store which uses similar features as "Easy-Reload" are listed below.

OCR-Text Scanner is App to recognize text from image based on "Tesseract" OCR.It runs your mobile phone to text scanner. Internet connection is not required to run this app.

This app lets users load a Pre captured image and then enhance it, after which the image is passed to an OCR engine and then the text is extracted. The main drawback using this app is that users cannot capture an image in app, rather they will have to use an image which is already captured [1].

Invoice123 This app keeps track of your entire invoice by scanning the invoice number and then by scanning the price it also has more added features like scanning bar codes, QR codes and so on. The main drawback using this app is Poor Language support (does not support English) [2].

Text Fairy (OCR Text Scanner) by Renard Wellnitz does a similar work like what "Easy-Reload" does. It also uses the back camera of an android device and captures an image and extracts all the texts from it .Main purpose of the app is to convert font styles so that users could read the text they prefer in a font style which they prefer [3].

After considering all these application, there were no specific app to read a scratch card and perform an operation with it, rather all the above listed apps where mainly used to extract text's only. There were applications using OCR and Tesseract OCR engine technologies in various countries. In Sri Lanka few related applications are available.

"Easy-Reload"- An android application to reload a phone easily will be a targeting this gap hoping to fill the gap. The application includes importance features such as read a prepaid reload card number and reloads the phone automatically. This project will be a huge improvement and also will be an inspiration to other researchers who have similar interest on this research area.

III. METHODOLOGY

A methodology is a formalized approach of implementing SDLC (Software Developing Life Cycle). It indict to the framework that is used to structure, plan, and control the process of developing an information system. The system was developed by using Prototype methodology of the SDLC.

The prototyping methodology performs the planning, analysis, design, implementation and testing phases concurrently, and all five phases are performed repeatedly in a cycle until the system is completed.

A. Planning

The planning phase is the most critical and important step in SDLC. The starting point of the project life cycle is planning phase. In this phase project team identified why "Easy-Reload" built and determinant how the project team went about build it. First of all the project value identified by project team. In this stage the project need and the basic functionalities of the application are identified.

Feasibility analysis helped to identify the risks associated with the application and dominates whether to proceed with the project. Identify the limitations of available technology to develop the system is called as technical feasibility. Identify the cost and benefits associated with the application called as financial feasibility. To identify how well the application will be ultimately accepted by the users the organizational feasibility was done.

B. Analysis

The analysis phase answers the questions of who will use the application, what the application will do, and where and when it will be used. Problem analysis helped to identify the current application. Through this the team identified the strength and weaknesses of the application.

Gathered the information through referring the past similar researches. And the team came up with a solution for the problem. The solution was "Easy-Reload".

The information gathering done by questionnaires and referring other similar applications. And the team members identified the researches and existing research papers and books related to OCR Tesseract technology. Involved in this information gathering. In this step team members got innovative ideas to implement the application successfully.

C. Design

Figure 1 Architecture diagram

Figure 1 illustrate the architecture diagram. The hardware, software requirements needed to "Easy-Reload". This is a mobile application therefore an android mobile application development tools are needed. A security plan needs to be designed to address how to keep "Easy-Reload" and its data secure.

D. Implementation

Here what the project team did the transformation of the design output into programs that are executable. A good implementation reflects the design decisions.

The goal was to implement a system correctly, efficiently, and quickly. With the quick design a working model of the product is implemented. Android Studio used for android application implementation.

The software component of OCR Tessaract technology required to solve the research problem was built. Programming language java is used to programming. SQLite is used to database development.

E. Testing

A test plan will be developed which will define a series of tests that will be conducted."Easy-Reload" tested through unit test. It carried out to ensure that the each and every function of "Easy-reload" are working properly or not.

Integration testing carried out to assess whether the all set of functions of "Easy-reload" are working together or not. That worked together without errors. System test performed to ensure that all modules and programs work together without error. Acceptance performed in order to make sure the system is complete, meets the need that is to increase efficiency.

IV. RESULTS

"Easy-Reload" developed as an android application.
Functionalities of "Easy-Reload" are described in this section.

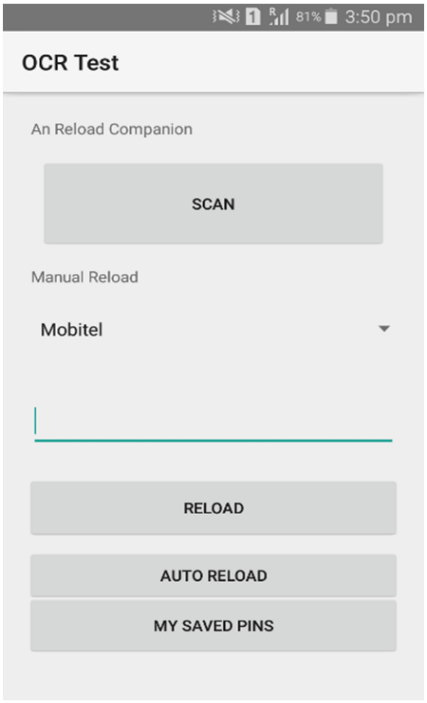

Figure 2 Main Interface

The figure 2 is the home interface of "Easy-Reload". The interface has buttons to scan the card, to do a manual reload and check the save card details.

Figure 3 Initialize Tesseract OCR engine and camera

The figure 3 shows how the camera and Tessaract OCR engine are initialize.

Figure 4 Camera Interface to capture the top up card number.

The figure 4 show the interface to take a photo of top up card.

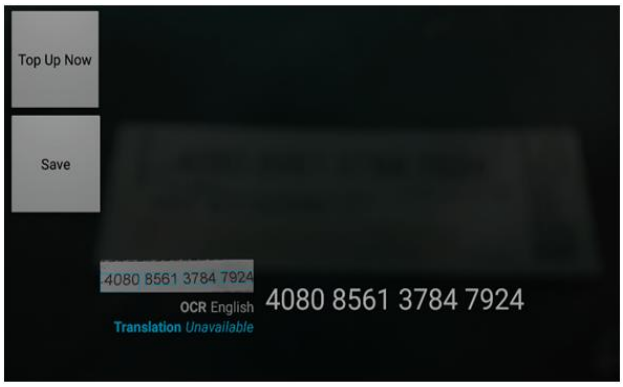

Figure 5 Screen to top up or save card number

The figure 5 shows the interface after take the photo of top up card. After image capturing two buttons visible in the interface one is to perform the reload function. Other one is to save card number.

If 'Top up now' button pressed the figure 6 displays in the app.

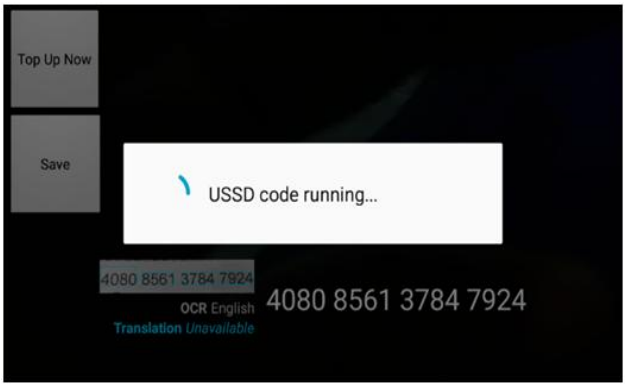

Figure 6 Perform top up function

The figure7 shows the recharging succeeded message came from service provider.

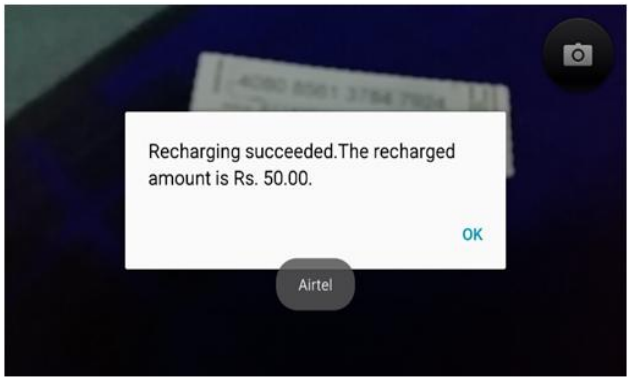

Figure 7 successfully recharged message

If 'save' button in figure 5 clicked the figure 8 appears in the app. According to the card amount the user should enter the amount.

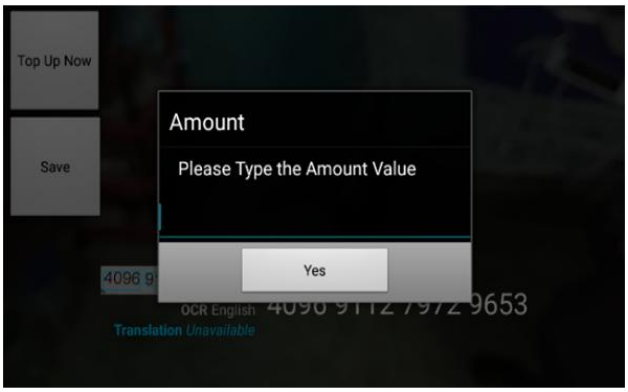

Figure 8 enter the amount value of card

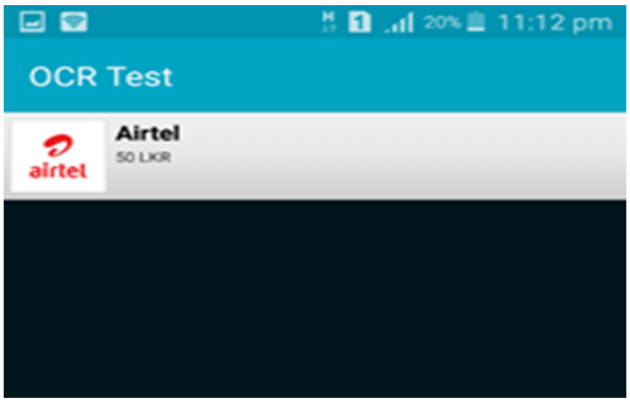

Figure 9 Saved top up card list

The figure 9 shows the list of save top up cards.

V. DISSCUSSION

The purpose of this project is to create a Standalone Mobile Application. Which address the issues in Introduction. By combining latest Tesseract OCR technology. The product would automatically track Numbers on the material and automate the process for you.

The research accomplished by modeling with the necessary functions for the store premises and mobile application for the prepaid connection users.

The app will be tested in various android devices. From those tests app didn't have any bugs or issues. To define the system accuracy app tested in various ways system accuracy will be 90 percentage.

The reasons to the accuracy level of app are

- OCR never recognized the characters in low light
  At first after take a photo in low light and passing it to OCR Tesseract just gave a message "OCR failed". Accuracy was not increased to a level so that the results were not better at low light.
- Portrait mode did not work
  Camera with OCR engine supported only landscape mode, but users' preferred to use this application in portrait mode. Steps were taken to convert the application to portrait mode, for this purpose Library files where altered, after altering the library files the application worked in portrait mode, but the performance was considerable low, So when compared to user satisfaction and usability, usability should be given a high priority, so portrait problem was ignored, thus the application was developed in landscape mode.

VI. CONCLUSION

"Easy-Reload" is an android application to reload a phone easily. With the help of optical character reading. Users can easily reload a phone. Only scratch the card then through application the phone automatically read the card and reload it automatically.

If users want to save a reload card number in case of emergency users can read the card number after reading the card number it save in the phone itself if the phone balance is low the phone is automatically reloaded with the card number.

During the development of the project the following are the limitations that figured out

- The interface of this android application and its functions should be user friendly and it should be in normal simple language format. The user should have easy access and the system should be attractive.

- All interfaces, error messages and confirmation messages of the system will only be supported for English language.
- The main constraint here would be the checking the genuineness of the users, which is not always possible. There can be security risks involved.
- Disabled people and people with no computer usage knowledge will be unable to use this system.
- Response time for requesting or waiting for a reply should take no longer than nine seconds.
- All the functionalities of the "Easy-Reload" should always available to users according to their user level.
- All the users have the basic knowledge of android phone in order to request a service from "Easy-Reload".

When use some kind of techniques user can get more benefits "Easy-Reload".

- PC operates using the operating system Windows 7 with Processor: Core i5 2.8 GHz.
- Java language is required to produce the system and android studio used to develop the system.
- Character Recognizing
  The system will use the Google Tesseract OCR technology to recognize the characters.
- SQLite database used to maintain the data.
  After implementation of the code, it should run on the following device
  Minimum requirements
  - Android device with an API level 16 or more (android 4.1 jelly bean)
  - A rear camera.
  - A single core 1000mhz or more CPU

  Recommended requirements
  - Android device with an API level 21 or more (android 5.0 lolipop)
  - A rear camera 5 megapixel sensor or more.
  - A dual core 2.0 GHz or more CPU

VII. FUTURE WORK

Many aspects of this application could be improved. Firstly this app could be developed to support multiple mobile service providers rather than just one, so that more users could benefit from this application. Furthermore the accuracy of the character recognizer could be increased by either training the already existing Tesseract engine or changing the OCR engine to a more reliable one. The interface of the application should be improved as currently it is not that user friendly.

The application does not support multiple SIM devices, only the first active SIM is taken to consideration when the application is used. But this application could be developed to support multiple SIM devices.

Periodic reload function (e.g.:- 9.00am every day). Through this feature users can set a periodic reload function which is reload the phone in every day or twice a day or once a week.

ACKNOWLEDGEMENT

The project team of Easy-Reload would like to declare our honest sense of gratitude to our institution – Sri Lanka Institute of Information Technology (SLIIT). We are also grateful to Lecture in charge for the subject Comprehensive Design/Analysis Project Ms. Gayana Fernando who helped us in many ways to a great extent in the project. Also very special thanks to Mr.Yasas Jayaweera for their endless support given at times of difficulty as well as to our seniors and the lecture panel. The completion of this undertaking could not have been possible without the participation and assistance of so many people whose names may not all be enumerated.

REFERENCES

[1] Apps, R. (2015, 07 23). *OCR - Text Scanner*. Retrieved from https://play.google.com/store/apps/details?id=com.offline.ocr.english.image.to.text

[2] Tung, A. (2015, 07 23). *invoice 123*. Retrieved from https://play.google.com/store/apps/details?id=tw.idv.arlen&hl=en

[3] Wellnitz, R. (2015, 07 23). *Text Fairy (OCR Text Scanner)*. Retrieved from https://play.google.com/store/apps/details?id=com.renard.ocr&hl=en

[4] anonymous. (2015, 05 30). *matworks*. Retrieved from matlab: http://in.mathworks.com/products/matlab/

[5] anonymous. (2015, 06 12). *OPENCV*. Retrieved from opencv for android : http://opencv.org/platforms/android.html

[6] apache. (2015, 12 07). *license*. Retrieved from http://www.apache.org/licenses/LICENSE-2.0

[7] android. (2015, 5 12). *android developer*. Retrieved from android ndk: https://developer.android.com/ndk/downloads/index.html

[8] author, m. (2105, 05 09). *Doc.opencv*. Retrieved from OpenCV thresholding : http://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html

[9] Axiata, D. (2015, 07 24). *dialog.lk*. Retrieved from factsheet: https://www.dialog.lk/fact-sheet

[10] Bell, B. P. (2015, 07 24). *Solo Scrums*. Retrieved from http://www.pbell.com/index.cfm/2007/6/17/Solo-Scrums

[11] ecllipse.org. (2015, 07 17). *eclipse*. Retrieved from eclipse juno: https://eclipse.org/juno/

[12] Gohr, A. (2015, 12 22). *Splitbarain.org*. Retrieved from Linux OCR Software Comparison: http://www.splitbrain.org/blog/2010-06/15-linux_ocr_software_comparison

[13] google. (2015, 6 12). *tesseract*. Retrieved from language data: https://code.google.com/p/tesseract-ocr/downloads/list

[14] Hildenbrand, J. (2015, 06 29). *Android Central*. Retrieved from What is android : http://www.androidcentral.com/what-android

[15] IDEA, i. (2015, 06 23). *intelij IDEA*. Retrieved from intelij IDEA: https://www.jetbrains.com/idea/

[16] ISTQB Exam Certification. (2015). *What is Prototype model- advantages, disadvantages and when to use it?* Retrieved 03 03, 2015, from ISTQB Exam Certification: http://istqbexamcertification.com/what-is-prototype-model-advantages-disadvantages-and-when-to-use-it/

