CA400 Functional Specification

for

Personal OCR Accounting Application

Prepared by Daniel Maguire

Student number: 12528133

25/11/2016

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This document contains the OCR Accounting applications system requirements and its analysis. It is to be used as a reference for the system's design.

1.2 Product Scope

The goal of development is to produce an easy to use personal accounting application.

The application make use of OCR technologies to capture and store purchase information from receipts, tickets and invoices. The data will be stored in a database and made available to the user to view. The applications main function is to allow the user to monitor day to day purchase habits and help provide suggestions as to how to improve current spending.

The aim of this application is to provide the user with an easy to use tool that will allow them to take control of their purchase habits.

1.3 References

Android.com "http://developer.android.com/about/dashboards/index.html"
Tesseract Documentation "https://github.com/tesseract-ocr/tesseract/wiki/Technical-Documentation"
OpenCV Documentation "http://opencv.org/documentation.html"

2. General Description

2.1 Product Perspective

This product is a personal accounting application. The user can use this application to store receipt data in a personal database accessed through the application. It allows the user to access this information in a day to day calendar view. The app will provide insight for the user into their purchase habits and will allow the user to view their purchase history through many different representations.
If used on a daily basis the application will allow the user to manage and understand their spending habits in the hope of bringing more control to the user. The aim is to provide the end user a tool that will allow them to educate themselves in their own spending affairs as well as providing some insight in the form of smart suggestions that may also aid the user in their decision-making process.

2.2 Product Functions

The application allows the user to capture purchase data from receipts by taking advantage of the devices built in camera to capture an image of the receipt. Image correcting algorithms are then applied to the captured image to improve image quality and rotate the image accordingly to allow the applications OCR software can read the receipt clearly.

The application will scan emails for signs of purchase data from online vendors where the user would not have received a purchase receipt. It will then give the user the option to store this information in the users purchase database.

The purchase data stored will be displayed in a user-friendly calendar view to allow the user to quickly access purchase history from a specific date with ease.

The application will analyse spending patterns and provide smart notifications such as if the users spending is increasing too much above their manually set limits, or if increase or decrease in spending habits from previous week.

2.3 User Classes and Characteristics

This product will be used by a wide variety of users from young to old. The application will be aimed at students and working individuals that would like to take control of their spending habits more effectively.
The apps will have a simple and user friendly user interface. A high level of expertise not be necessary to use the app. A user who is able to use their smartphone comfortably should have no problems using the product.
From a user's point of view, the main objectives of the product will be its usability and reliability. Making use of the app requires frequent use to capture receipt data so the application needs to be able to be quickly launched and used so it does not start to feel like a burden every time they make a purchase.

2.4 Operation Scenarios.

1. The use case for a first-time user using the application is as follows:
    a. First the user will be asked register with the application.
    b. The user will enter their name, email and password and registers an account.
    c. The user will then be brought to the main menu
    d. The user will then have the option to scan a receipt.
    e. The user clicks the scan receipt button and takes a picture of the receipt.
    f. The purchase information is scanned and displayed on screen to the user to confirm all information was correctly processed.
    g. If the user clicks yes and the information is stored in the database, else if the user clicks no the user is prompted edit it correctly or retake the picture.

2. The use case for a regular user using the application is as follows:
    a. The user will be asked to log in if they have not already done so.
    b. The application will display the main menu to the user.
    c. The user is now presented with the choice of capturing a receipt image or viewing their purchase history.
    d. The user chooses to view their purchase history and is presented with a calendar view.
    e. Spending recommendations are displayed at the lower half of the screen.
    f. The user clicks on the 14th August on the calendar.
    g. The application displays all the purchases made on that day.

2.5 Operating Environment

This will be an Android based application. The minimum SDK it will run on is Android 4.4 (API 19) with the target SDK being the current Android 7.0 (API 25). It is built to run on an Android smartphone with a build in camera. For the purpose of early development, we may use an Android device emulator to run the app, developing the camera activity into the application will require a physical Android device. The application will be developed in an Android Studio enviornment.

2.6 Design and Implementation Constraints

The main concern for developing and implementing is the lack time for making myself familiar with the OCR and image processing aspects of the application and other possible technologies to be used. It will take some time to learn and understand these clearly to be able to implement them correctly in the application.
The OCR feature is a time-consuming task and it is one that I am prioritising as the application is almost non-functional without this feature.
The app will require some memory on the user's phones but this will likely not be an issue.

2.7 Assumptions and Dependencies

The application will make use of Googles Tesseract API to handle the OCR aspect of the app. After researching many possible OCR API's, Tesseract seems to stand out as the clear choice. The Android version of Tesseract is Tess-Two, which allows use of all of Tesseracts features ported nicely for Android development.
I plan to use the OpenCV library for the image processing aspect of the application as it has useful image enhancing and transforming tools included in its library, it can now be used directly in Java development.
I will be using JUnit to test the code as it is currently built into the testing suite in Android Studio. XML will be used for designing the user interface of the application. The application will also make use of many of the of the other tools in the Android SDK.

2.8 Technologies Used

Languages:

- Java - as it is the main platform for android app development
- XML - for the UI layout
- SQL - for querying and updating the database
- PHP - Bridge connection between app and external SQL database

Hardware:

- Android Smartphone

3. Functional Requirements

Description - The application must be able capture and process images into text.
Criticality - Very high
Technical issues - Text may not be read correctly if the ink on the receipt runs or the text starts to fade too much. Some characters may resemble other characters too closely and may be read as the wrong character.
Dependencies with other requirements - This is dependent on the application having access to the devices camera. Without this, the application will not be able to capture the image.

Description – The application must be able to access the devices camera.
Criticality - Very high
Technical issues - If permissions are not correctly set in the in the applications manifest file or if the user denies access to the camera this will cause issues.
Dependencies with other requirements – The user must have a built-in camera on the device.

Description – The processed receipt data must be stored in the users' database.
Criticality – Very high
Technical issues – This may fail if the users account was not set up correctly.
Dependencies with other requirements – Database must be live and connection must be established.

Description – The application must allow the user to view their purchase history.
Criticality – Very high
Technical issues – The application might encounter problems connecting to the database.
Dependencies with other requirements – Database must be live and user must have an internet connection or user must have database entries stored locally.

Description – The application must be able to connect to the database.
Criticality – High
Technical issues – The database server might be down or an internet connection may not be able to be established.
Dependencies with other requirements – Database must exist and be functional. Connection details must be correct.

Description - The application must be able to enhance the image and correct the receipt rotation.
Criticality - High
Technical issues – The receipt image may not have defined edges which will make correcting the image rotation difficult.
Dependencies with other requirements – The application must have access to the OpenCV image processing tools.

Description – The application must allow the user to log in.
Criticality - High
Technical issues - The user may have forgotten their log in credentials or a connection to the server may not be able to be established.
Dependencies with other requirements – The program must query the database for the correct matching information in order to validate the user.

Description – The application must allow the user to register.
Criticality - High
Technical issues – An internet connection may not be able to be established or the user may be entering invalid details.
Dependencies with other requirements – There must be a live connection to the database in order to store the new users' credentials.

Description -The application must be able to pull purchase information from users' emails.
Criticality - Moderate
Technical issues - The application may not be able to access the users emails if the user has not configured this setting.
Dependencies with other requirements - The user must have an email address configured on the same device as the application.

Description – The application should provide recommendations based on the users purchase history.
Criticality - Moderate
Technical issues – The user may not have a purchase history if they are a new user.
Dependencies with other requirements – There must be a live connection to the users purchase history receipt database or the receipt data must be stored locally.

Description – The application should display graphical representations of the users purchase history.
Criticality – Low
Technical issues – The user may not have a purchase history if they are a new user.
Dependencies with other requirements – There must be a live connection to the users purchase history receipt database or the receipt data must be stored locally.

4. Other Nonfunctional Requirements

4.1 Performance Requirements

A device must be running Android version 4.4 or above for this application to be compatible. It is recommended that the device has at least 512mb of RAM and at least an 800MHz processor for the application to run smoothly with other applications open in the background. There should be at approximately 40 MB of free space on the device.

4.2 Security Requirements

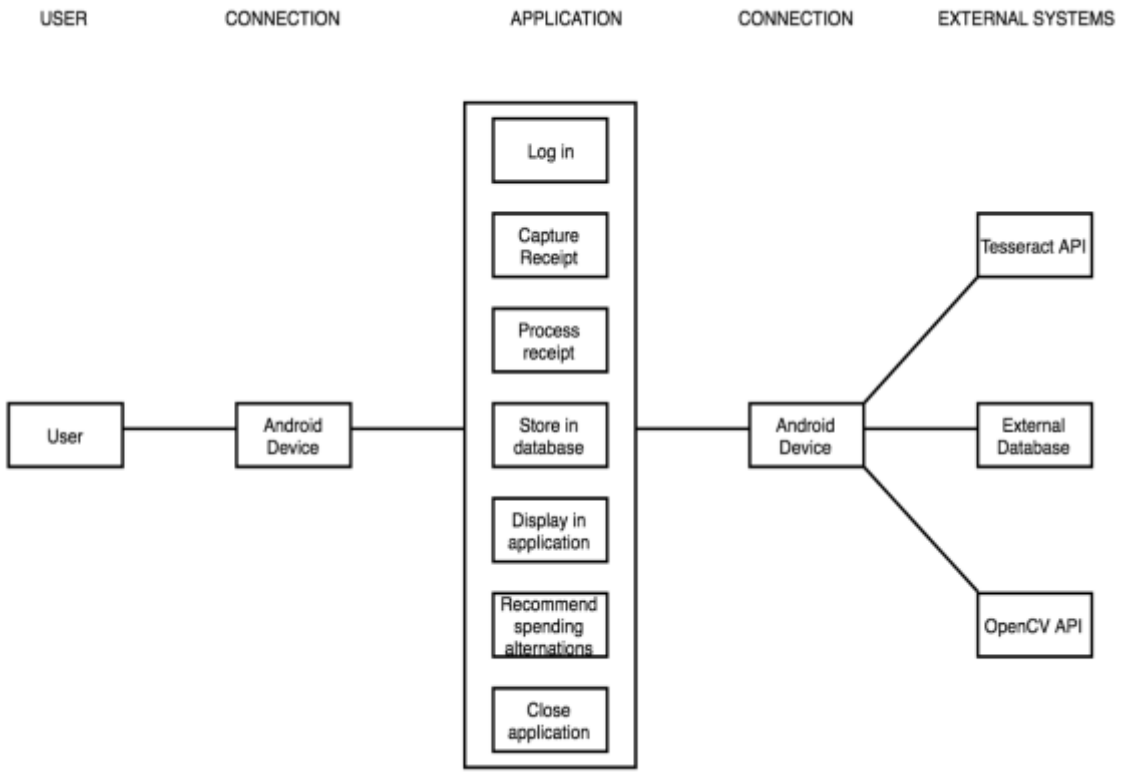
Unauthorized access to the application and its data by external applications on the system is not allowed. This ensures the integrity of the application from accidental or malicious damage.
On first use of the application the user will have to create an account. All purchase information received will be backed up and stored in the user's personal external database. This is necessary in the case of accidental loss or damage to the user's device. It also means the user can access their information on other devices if desired. The users account information encrypted. This is to secure against attackers, and mainly for the users to know that even developers involved in the application cannot access their data, hence the keys must not be accessible by the tech team.

4.3 Software Quality Attributes

For the customer, the applications main quality attributes are reliability, availability, performance and usability. A key part of the applications main function is to is capture receipt information using the in-built camera on demand. The user wants to capture these images fast and effectively without having to retake the image again and again. For this reason, performance and availability are main quality attributes of this application.
Another quality attribute of the application is reliability. This application has to be trustworthy and reliable for the user to take full advantage of it.
A secondary quality attribute of the application is usability. This application will be targeted at any individual who wants to take control of their spending habits more effectively. This is a large demographic that covers young tech savvy individuals, but it also covers an older generation that may not be as familiar with many of the technical concepts of modern applications. So, usability is essential for these people that may not adapt to new technologies easily.

5. System Architecture

Below is a diagram outlining the applications architecture. On the left of the diagram it can be seen how the user interacts with the main application. On the right the external components needed for the application to function can be seen. The center of the diagram contains the main application and its components.



6. High Level Design

6.1 User Interfaces

Image 1

Image 2

Above are mock-ups of how I plan to have design the calendar view of the application. I would like to keep the layout of image 1, with the bottom half of the screen displaying purchase information and/or possibly recommendations for the user. I would like to keep the design focus on image 2 as it gives the illusion of a friendlier, easier to use user interface so the user will feel more comfortable navigating the application.

6.2 Hardware Interfaces

This application will take advantage of the built-in camera hardware in the device to capture images of the receipts. The devices GPS hardware will also be used for geotagging purposes to allow the user to know where each receipt was captured. The application will also have to make use of the devices inbuilt storage to temporarily store receipt images on the device.

6.3 Software Interfaces

The minimum SDK this application will run on is Android 4.4 (KitKat) with the target SDK being the current Android 7.0 (Nougat). This allows the application to be compatible with approximately 80% of all Android user devices. This will allow the application to function on a majority of the android communities' devices while still keeping a majority of the modern features introduced in KitKat.

7. Preliminary Schedule

7.1 Schedule

September	▪ Decide on project idea ▪ Discuss with lecturers ▪ Conduct necessary research
October	▪ Find a project supervisor ▪ Make project proposal ▪ Present proposal ▪ Configure GitLab with project ▪ Start project blog
November	▪ Begin building application ▪ Get Tesseract to work correctly in application ▪ Start using OpenCV software to help with image processing
December	▪ Improve accuracy of Tesseract by training data. ▪ Start building and configuring database ▪ Start testing
January	▪ Add log in and registration functionality ▪ Continue testing
February	▪ Add smart notifications to application ▪ Create email scraper feature ▪ Start working on UI layout
March	▪ Conduct user tests ▪ Fine tune application based on user tests ▪ Continue to work on UI.
April	▪ Conduct more user tests.
May	▪ Create Technical manuals ▪ Video walkthrough ▪ Submit work ▪ Final year expo ▪ Demonstration

7.2 GANNT Chart