The George Washington University

CSCI - 4237/6907 Software Design for Handheld Devices

Fall 2011

Final Project Report

ALLEGRO

Mobile Scanner

V1.0

Submitted on 12/12/11

Rahul Betal

Table of Contents

1) Introduction

Allegro is an Android application intended to capture the text in form of images and converts them into respective Portable Document Format (PDF). The text captured can be converted into PDF in two ways. The first method, which is a quick and dirty approach, simply uses an image and creates a PDF with one image per page. The second method employs the use of an Optical Character Recognition (OCR) web-service, to which individual image is sent and the result is stored in the PDF.

The OCR functionality can be implemented in the application by using open-source libraries like Tesserect. But, considering the constraints on a typical Android based Smartphone, inbuilt functionality will affect the overall performance also it will drain the battery resource in significant amount. Not to mention, the huge CPU consumption, that may affect other Android applications and services and/or even the application itself.

2) Allegro Features

Following are the features of Allegro Mobile Application:

- Capture and publish images as PDF album.
- Convert text into PDF.
- Share the PDF using email.

3) Allegro Limitations

Allegro has some limitations when it comes to convert the Text using OCR feature. A free limited-use, limited functionality OCR web-service http://ocrapiservice.com is used to perform OCR operations. Limitations are mainly because of the OCR web-service policies, which accept only images of JPEG and PNG format. Furthermore, there are limitations on the type of text that should be captured. The image that needs to be rendered in the text should have white background and the text should have high contrast values. The text should have large font and the DPI should be 150 or higher. The web-service fails to process the text images

which are screenshots due to low DPI. Currently, the web-service does not support OCR processing on images with hand-written text.

4) Implementation

4.1) Development Requirements

The application development required following tools:

1) Android SDK
2) Eclipse 3.7 or higher
3) Android Smartphone

An Android Smartphone is required since images are to be captured using the camera. Although an emulator can be used, but a real Android device gives the insight of application behavior on real-time basis.

4.2) External Libraries

Allegro application requires two external libraries as described below.

1) iText PDF library: iText is an open-source Java library to create PDF files. Allegro application depends heavily on this library to publish sequence of images into one integrated PDF. It also used this feature to generate PDFs from the text based response received from the OCR web-service.

2) HTTPMime library: This is a standard library from Apache project used to initiate connections with the web-service.

4.3 Application Modules

The application is composed of four modules as described below and shown in Fig.1. All the modules implemented are defined in **in.andruid.allegro** package.

1) Image Capture Module: This module captures the images using a camera. A new project is created and all the captured images are stored in the ProjectMetadata data structure. An inbuilt Android activity from MediaStore is used to invoke the camera access.

2) PDF Module: This module generates the PDF using either from the images captured so far or from the text response received from OCR web-service.

3) OCR Module: The OCR module uses images as input and communicates with OCR web-service using REST based calls. The output from OCR module is a text for each individual image which is later compiled to PDF using PDF module.

4) Communication Module: This module is optional. If user wishes to forward the PDF generated, he/she can do so by sending an email. An inbuilt android Activity for sending an email is invoked.

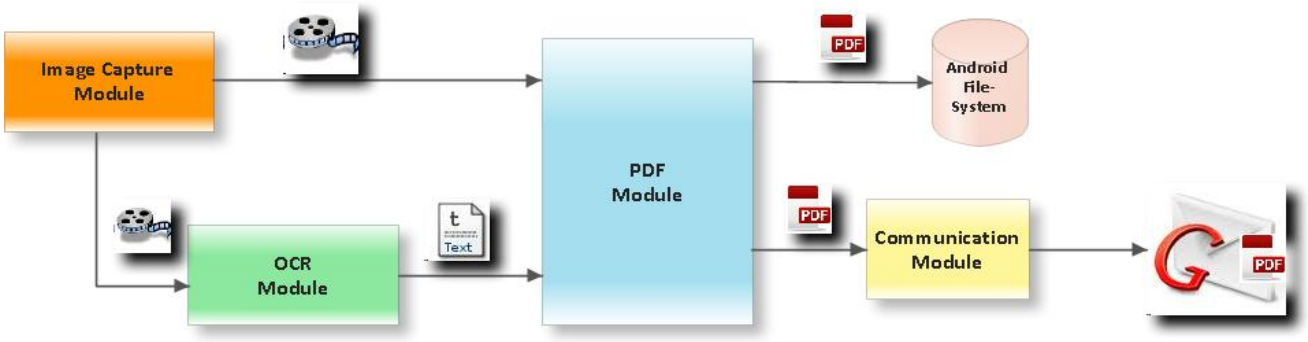

Figure 1 Allegro Modules

4.4 Description of Classes

The Allegro application is comprised of following classes as stated below. The class diagram is shown in Fig. 2.

1) Image Capture Module:
   a. AllegroActivity.java - Activity to display images and project.
   b. CustomizeDialog.java - Creates a project name dialog box.

2) PDF Module:
   a. CreatePDFandOCR.java – Router class
   b. PDFGenerator.java -Generates PDF

3) OCR Module:
   a. GeneratePDFFromOCR.java - Prepares to call web-service
   b. OCRServiceAPI.java - Calls API service.

4) Generic classes:

a. AboutMenu.java - Activity to display 'About' screen
b. MyNetworkInfo.java - Static methods for network connection
c. ProjectMenuHelp.java - Activity to display 'Help' screen.
d. SplashScreen.java - Activity to display a splash screen.
e. ProjectMetadata.java - Static class for saving project data.

All the above classes in 4, except ProjectMetadata class is reusable and I've reused them in other Android based Lab assignments with changes only made in presentation data.

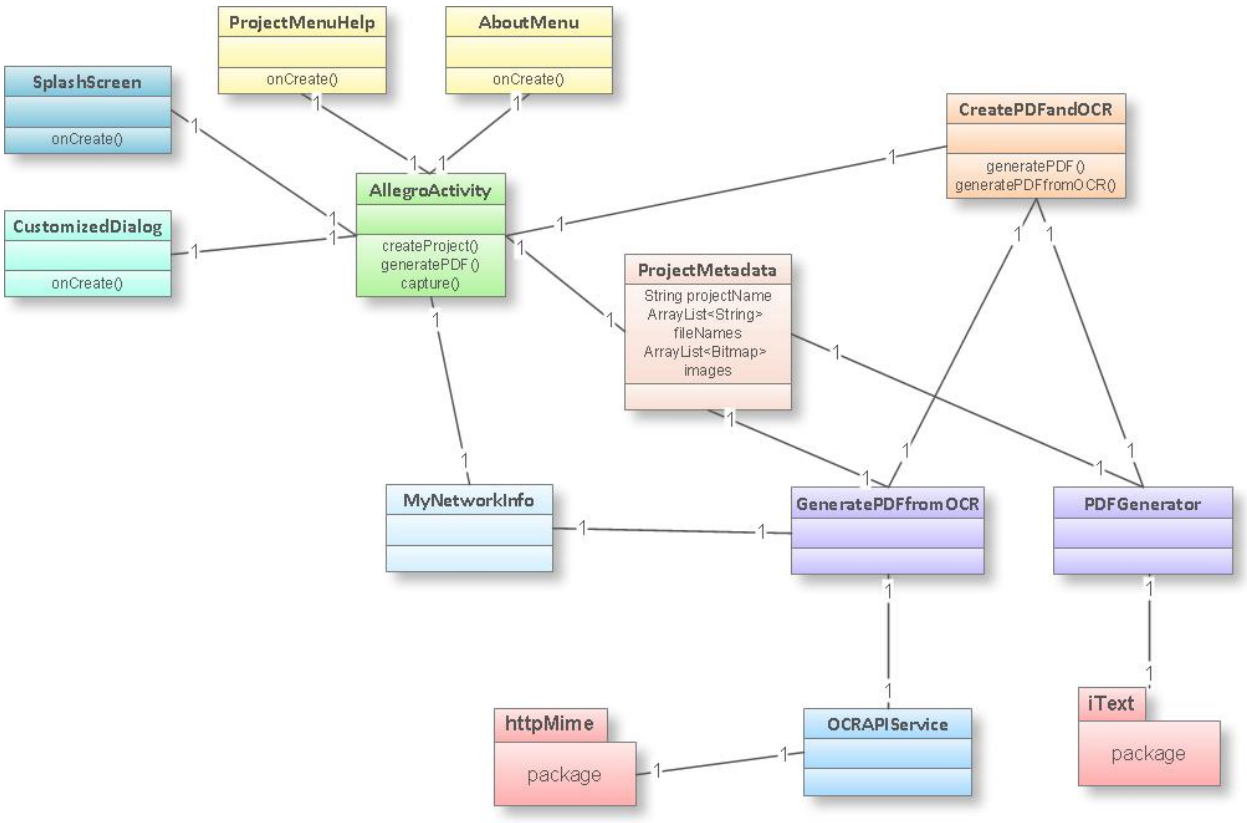

Figure 2 Class Diagram for Allegro

4.5) Allegro policies

Allegro uses two policies for its execution by considering the availability of resources. The Allegro application at the time of starting up gives preference to following policies.

1) If the battery level of Android device is lower than 10%, the application will present a message.
2) If the Internet access is not available then the application won't start.

4.6) Allegro application life-cycle

Fig.3 depicts the application life-cycle of Allegro.

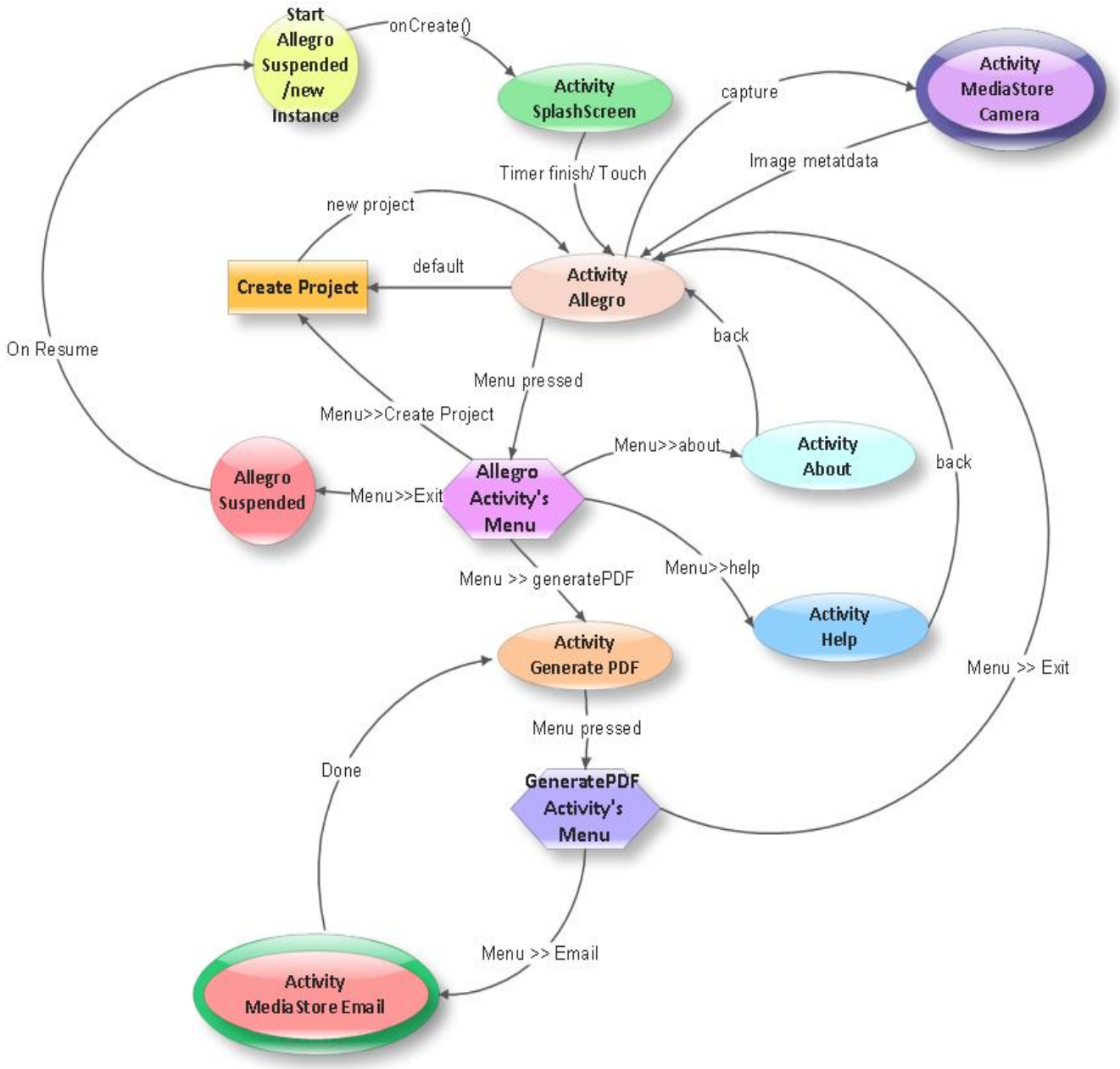

Figure 3 Allegro Lifecycle

4.7) Project Tree

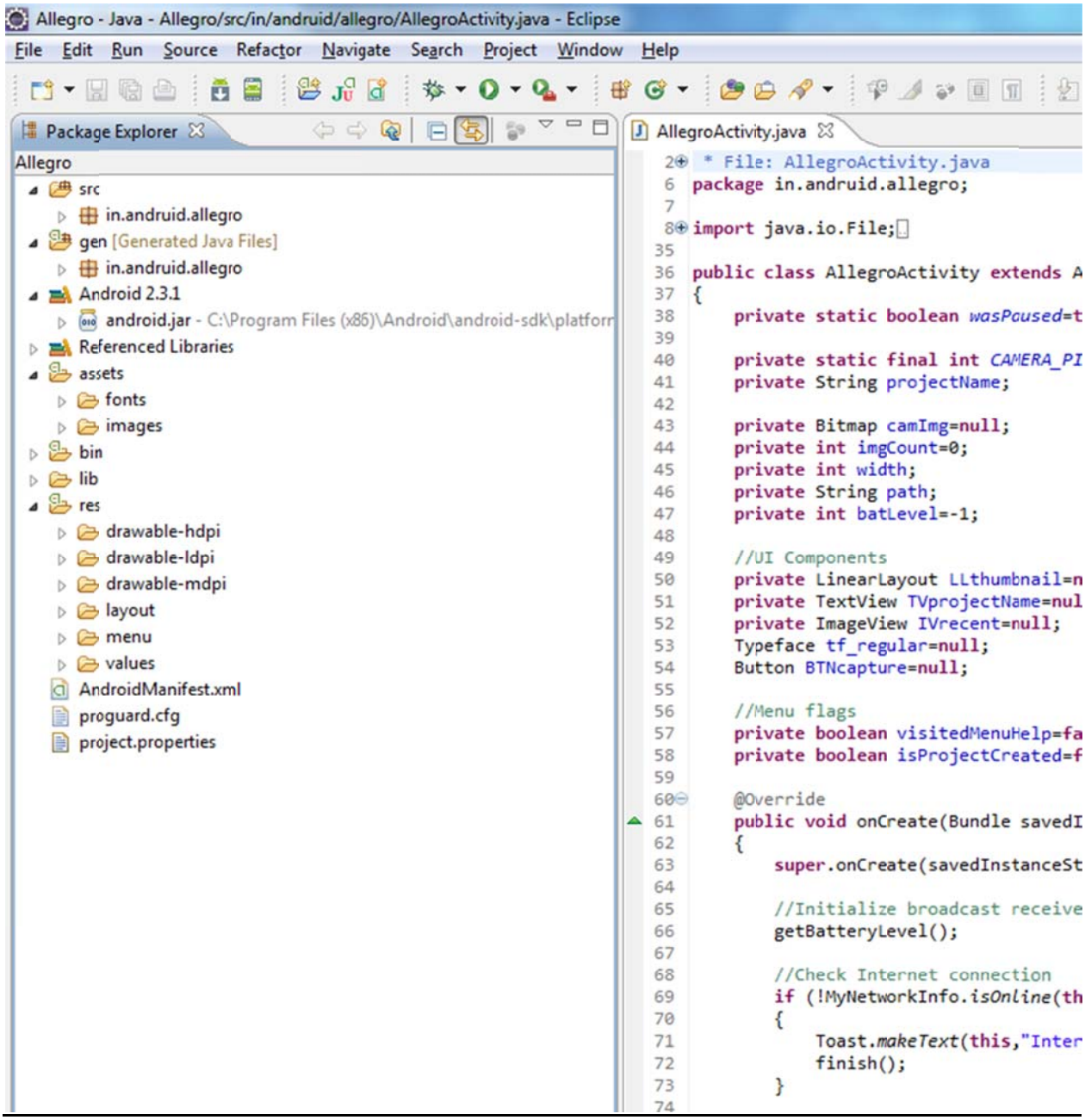

Figure 4 Allegro Project Tree in Eclipse IDE

5) Issues encountered

The major issues encountered during the Allegro development can be divided among it's respective modules.

1) **Image Capture Module:** Accessing the hardware resource like camera was bit difficult. Especially the settings like auto-focus and flash support. If Camera hardware was to be used from scratch, then creating a generic class to fine-tune its parameters like auto-focus, flash, brightness and contrast would be needed, which would further take more time. The work around was to use the inbuilt MediaStore Camera activity which comes preloaded from the vendor. This activity is already optimized to use the Camera hardware efficiently. Plus fine-tuning of parameters is not required.

2) **PDF Module:** The main challenge in this module was to select an appropriate PDF generation library which is optimized for mobile devices. iText satisfies some of the requirements though some compromises were done to implement this library. The major issue in this library is with the text and image formatting. It is prone to mash up the image if proper alignment and resizing is not performed. In quick and dirty mode, all the images captures have been pasted on the PDF document after they are normalized. Text generated from OCR operation is simply right aligned as there are no instructions about the text alignment. Headers and footers to display project name and page numbers are being implemented.

3) **OCR Module:** A free OCR web-service is really hard to find. There are numerous OCR web-services available, but no one offers a free access. The web-service Allegro is using is a bare-bones OCR utility which only accepts images in JPEG and PNG format. This means that if in a project there are 'n' images, thus it would need 'n' times to call this service. This is an overhead in terms of utilizing the bandwidth and generating extra requests. It would be nice, if this service would have accepted PDF files where all the

images that needs to be compiled are present in one single file. This approach would generate only one request to access the network. Furthermore, the size of file can be compressed further to reduce the data traffic. There was no choice, than to use this service for sending individual images and receiving the response.

6) Future Prospects

The Allegro application in its later version has the potential to improve and add more functionalities in each module, as described below.

1) **Image Capture Module:**
   - To implement the Camera access module with ability to controls features like flash, auto focus etc.
   - To implement grid-based preview screen, so that the user can adjust the distance and angle of the lens.

2) PDF Module:
   - To implement encryption functionality.
   - Ability to identify text from the image.
   - User customized header and footer.
   - User customized title.

3) **OCR Module:**
   - To implement multi-lingual support.
   - Generating formatting scripts for the generated text.
   - Accepts images with hand-written text.

7) Screenshots

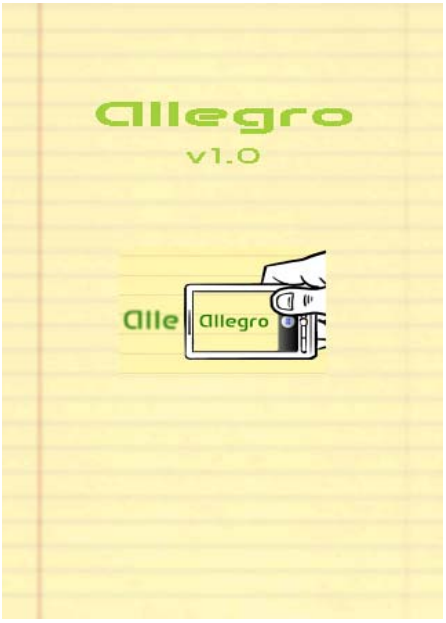

Figure 5 Splashscreen

Figure 6 New Project

Figure 7 New Project 'demo'

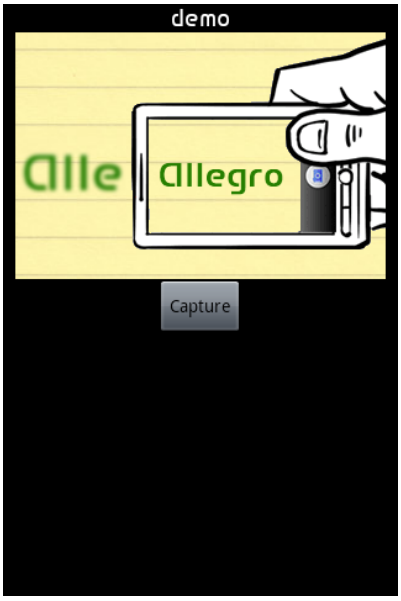

Figure 8 Initial screen

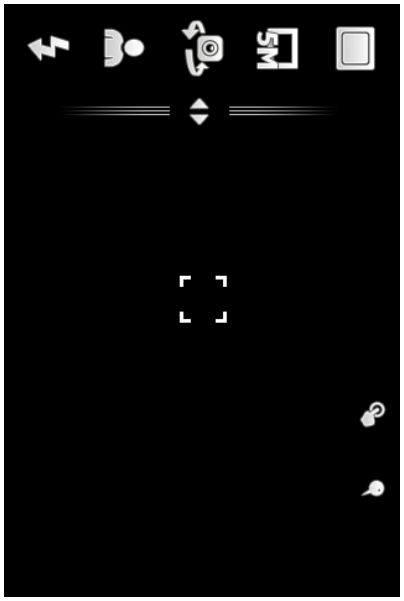

Figure 9 MediaStore Camera Activity

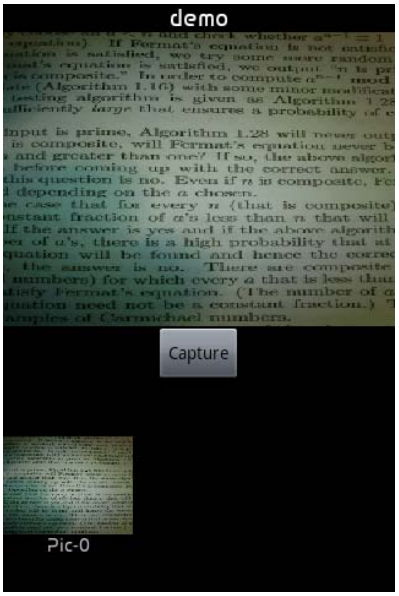

Figure 10 Image Capture

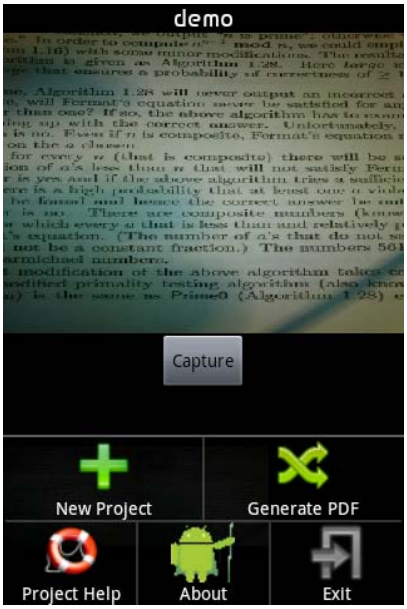

Figure 11 Allegro Menu

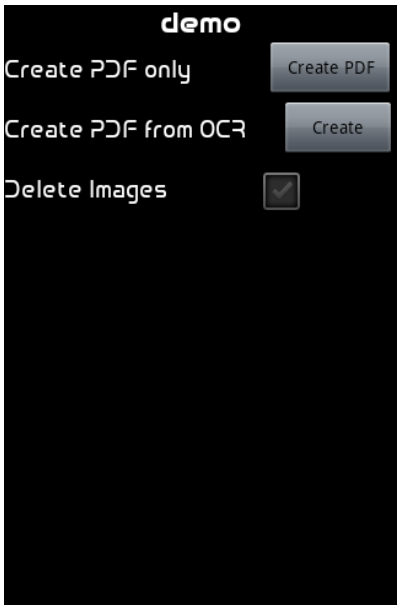

Figure 12 Generate PDF activity

Figure 13 Generating Quick PDF

Figure 14 Result Quick PDF created

Figure 15 Generating PDF from OCR

Figure 16 Result PDF Generated

Figure 17 PDF Generator Menu

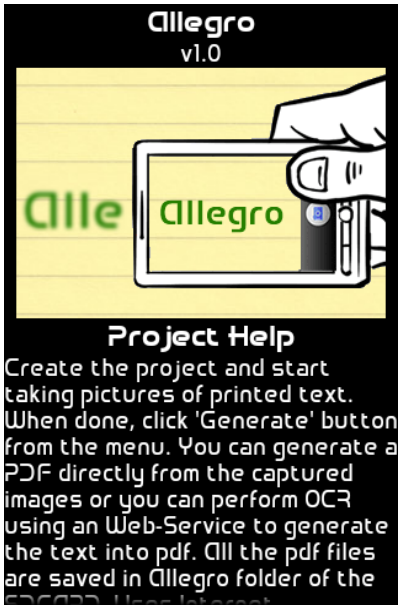

Figure 18 Allegro Help

Figure 119 Allegro About

8) Technical Summary

- Target Android Device SDK: 9
- Package: in.andruid.allegro
- External Libraries:
    1) iText PDF
    2) HTTPMime
- Permissions Used:
    1) Network State
    2) Internet Access
    3) Camera Access
    4) Data Write privilege on External Storage
- Resources Used:
    1) Fonts
    2) Images
    3) Layouts
    4) Menus
    5) Strings
- Application Size: 3.08 MB
- Screen Orientation: Fixed to Portrait.

9) Summary

The exercise of developing the above mentioned application empowered me as student of Software Design for Handheld Devices to learn develop an application with reusable components with best pragmatic approach. It proved to be good experience to understand various methodologies to be considered while developing mobile applications, particularly a 'well-balanced' one. This endeavor offered me a fertile ground to understand the Android framework in depth and develop a skill set in mobile application domain.

The future of Allegro application, if developed to its full potential is promising. I would like to improve this application and would like to port it to all the available range of Android devices.