



UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Cos301 : Test Policy
for the SmartCard Application

October 4, 2018

NAMES:

Mufamadi Khodani
Setoaba Phuti
Dandashe Kagiso Xolo

STUDENT NUMBER:

u14197520
u13032616
u14245681

Contents

1	Mission of Testing	1
2	Test Description	1
2.1	Unit testing	1
2.1.1	Storage	1
2.1.2	Application activities and functions	1
2.2	Test Cases	1
2.2.1	Incorrect Log In functionality	2
2.2.2	Correct Log In And functionality	2
2.2.3	Register And LogIn functionality	2
2.2.4	Generate QR Code functionality	2
2.2.5	Scan QR Code functionality	2
2.2.6	View Details functionality	2
2.2.7	Update user details functionality	3
2.2.8	NFC functionality	3
2.2.9	View Cards Functionality	3
2.2.10	NFC Send functionality	3
2.2.11	Import Card Functionality	3
2.2.12	Create and View Appointment Functionality	3
2.2.13	Navigation Functionality	3
3	Test Evaluation	4
4	Quality level to be achieved	4
5	Approach to Test process improvement	4
6	Git test cases	4
7	History	4

1 Mission of Testing

To effectively deliver efficient and effective software that meets the customers requirements. To further ensure that quality software is produced.

2 Test Description

We include three categories of tests:

- Small tests which are unit tests that you can run in isolation from production systems. They typically mock every major component and should run quickly on your machine.
- Medium tests which are integration tests that sit in between small tests and large tests. They integrate several components, and they run on emulators or real devices.
- Large tests which are integration and UI tests that run by completing a UI workflow. They ensure that key end-user tasks work as expected on emulators or real devices.

Our Testing process will be divided into two sections. Unit testing and Automated testing.

- Unit testing is done using Junit
- Automated testing is done using Espresso

We choose to use Espresso because it is free and easily intergratable or comes built in with android studio which is the platform we are using to develop our app ,and it also works hand-in-hand with junit so we can simply automate our junit unit tests to automated tests with a small configuration or change to the test code.

2.1 Unit testing

2.1.1 Storage

- Check if the information is stored correctly.
- Check if the information can be retrieved from the database.

2.1.2 Application activities and functions

- Ensure that an appointment can be set and retrived
- Validate that emails are of the correct form
- Ensure that a user can be created

2.2 Test Cases

Below are our tests which are grouped by Large Automated tests which consist of the listed Unit tests within them

2.2.1 Incorrect Log In functionality

- test for when No input is entered to the email and and password field
- test for when only the email is provided
- test for when the email and password are both incorrect
- test for when correct email is provided and incorrect password

Link to test case- [Github IncorrectLogIn functionality test case](#)

2.2.2 Correct Log In And functionality

- tests if a Registered User can login to the app
 - tests if a user can logout after a successful login
 - tests if the App displays a message to warn the user that they are about to log out
- Link to test case- [CorrectLogInAndLogoutTest](#)

2.2.3 Register And LogIn functionality

- Checks if a new user can register an account
 - Checks if passwords match when confirming password when registering
 - Checks if a new user can Log in with their details
- Link to test case- [RegisterNewUserAndLoginTest](#)

2.2.4 Generate QR Code functionality

- Checks if the generate Qr Code button works
 - Check if the QR code generates correct information
- Link to test case- [GenerateQrTest](#)

2.2.5 Scan QR Code functionality

- Checks if the Scan QR code button works
 - Checks if the Scan QR code button opens the camera intent
 - Checks if the Qr code is scanned and the user information is displayed and saved
- Link to test case- [QrScannerTest](#)

2.2.6 View Details functionality

- Checks if the view details button works
- Checks if the view details activity displays the current user information

2.2.7 Update user details functionality

- Checks if the update user details button works
- tests if the user can update thier information

2.2.8 NFC functionality

- Tests if the App notifies the user if the NFC technology is not switched on.
- Tests if the App notifies users that do not have NFC

2.2.9 View Cards Functionality

- Tests whether a user can view thier saved cards
- Tests the functionality of the search tool that searches for saved card
- Tests if a user can delete a saved card
- Tests if a user can view the details of a saved card
Link to test case- ViewCardsTest

2.2.10 NFC Send functionality

- Tests the Nfc send function and if it sends correct information

2.2.11 Import Card Functionality

- Tests whether the import card button works
- Tests whether the capture card button opens the camera intent

2.2.12 Create and View Appointment Functionality

- Tests whether a user can Create an Appointment
- Tests whether a user can view saved Appointments
- Tests whether a user can delete an appointment

2.2.13 Navigation Functionality

- Tests if the navigate me button opens the google maps API
- Tests if the find me button returns the current locaton or an error message if it couldnt find the location

3 Test Evaluation

We will use JUnit and Espresso to ensure that implementations have been functions as it should. Espresso automates tests , JUnit is a testing framework for Android and Java applications.

4 Quality level to be achieved

We aim to have no outstanding faults prior to product release.

5 Approach to Test process improvement

Project review meetings are held in regards to improving the project.

6 Git test cases

link to the github test cases folder

Test Cases

7 History

snapshot of some test cases

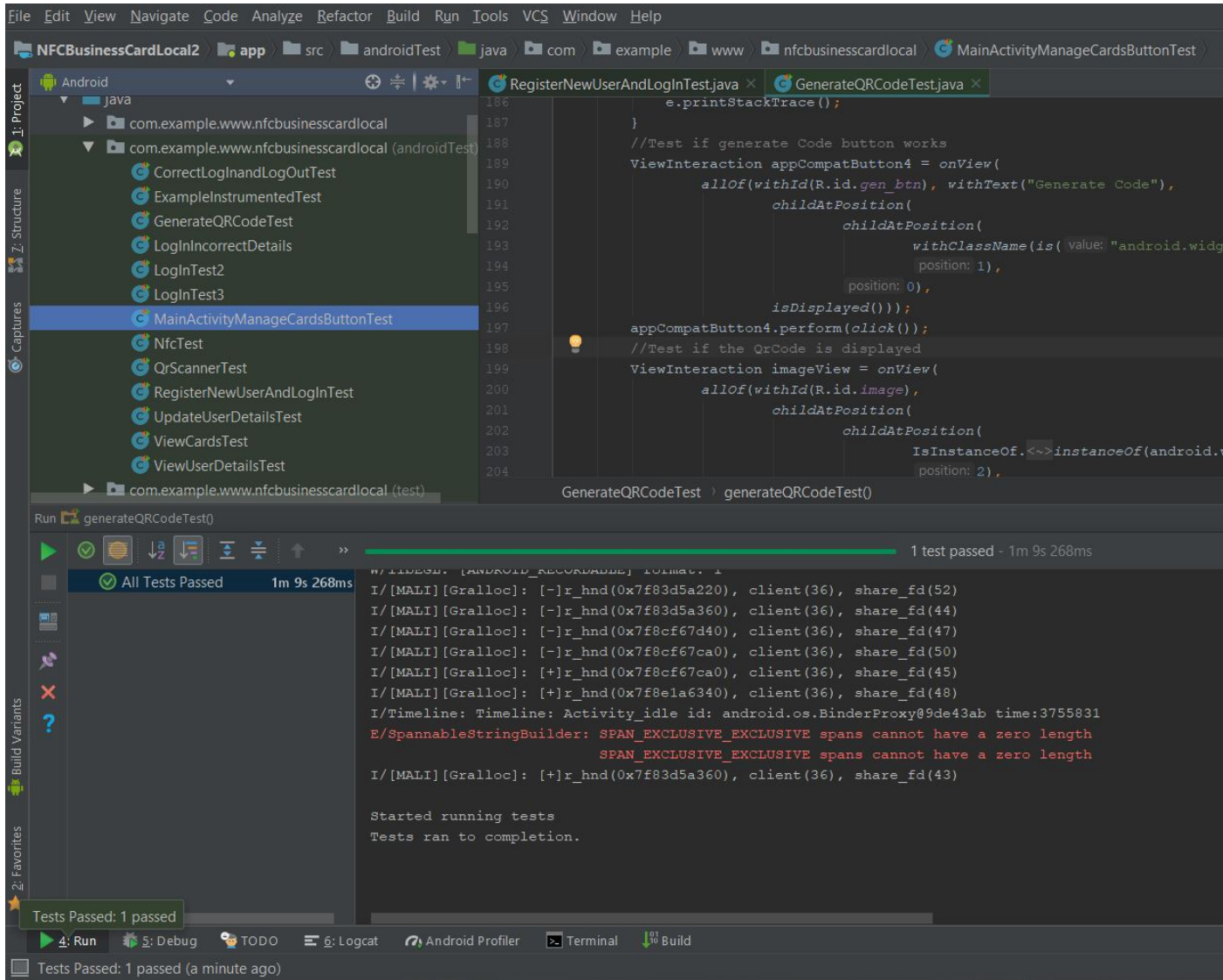


Figure 1: GenerateQr test pass

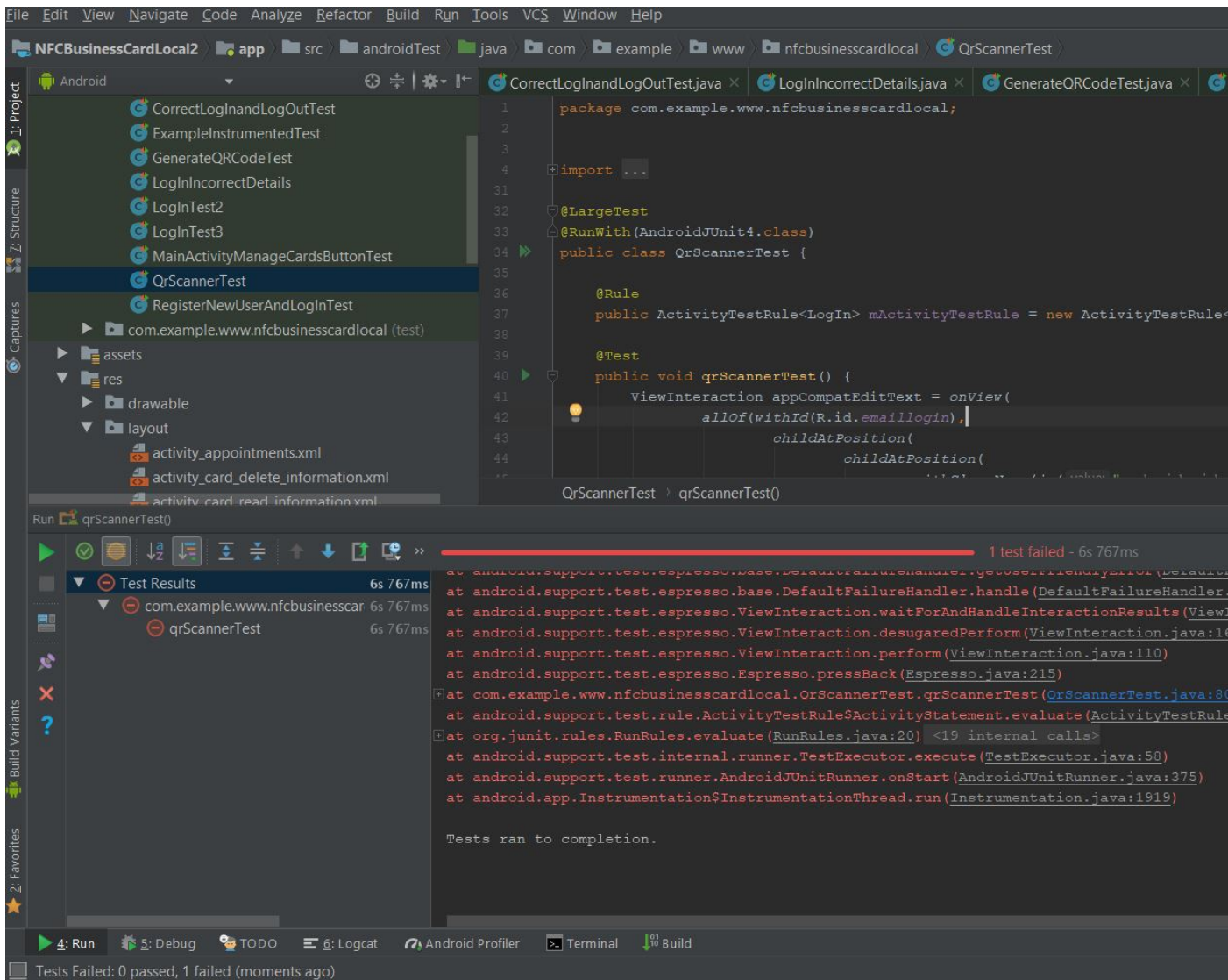


Figure 2: Qrscan test fail

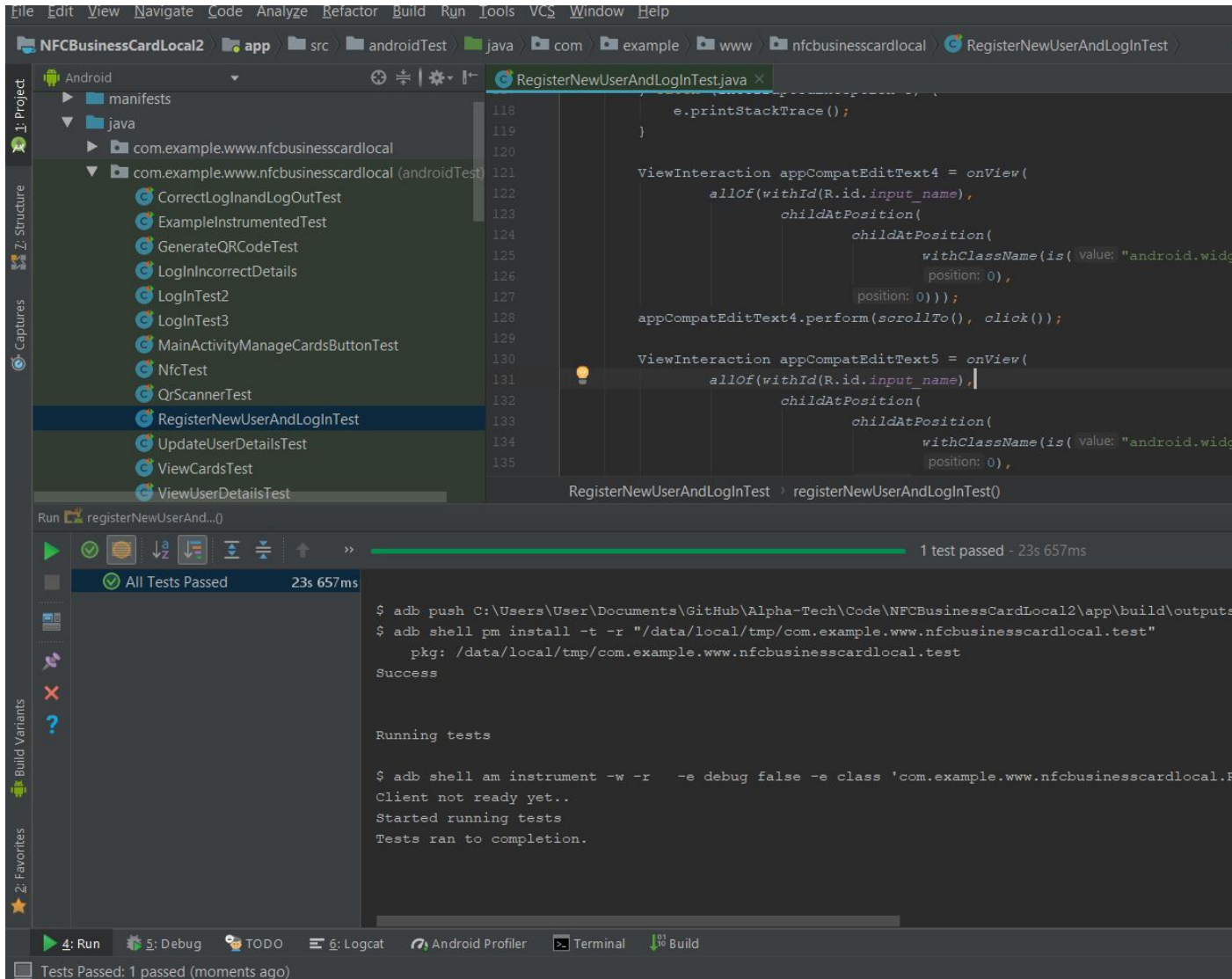


Figure 3: Register new user And login test pass

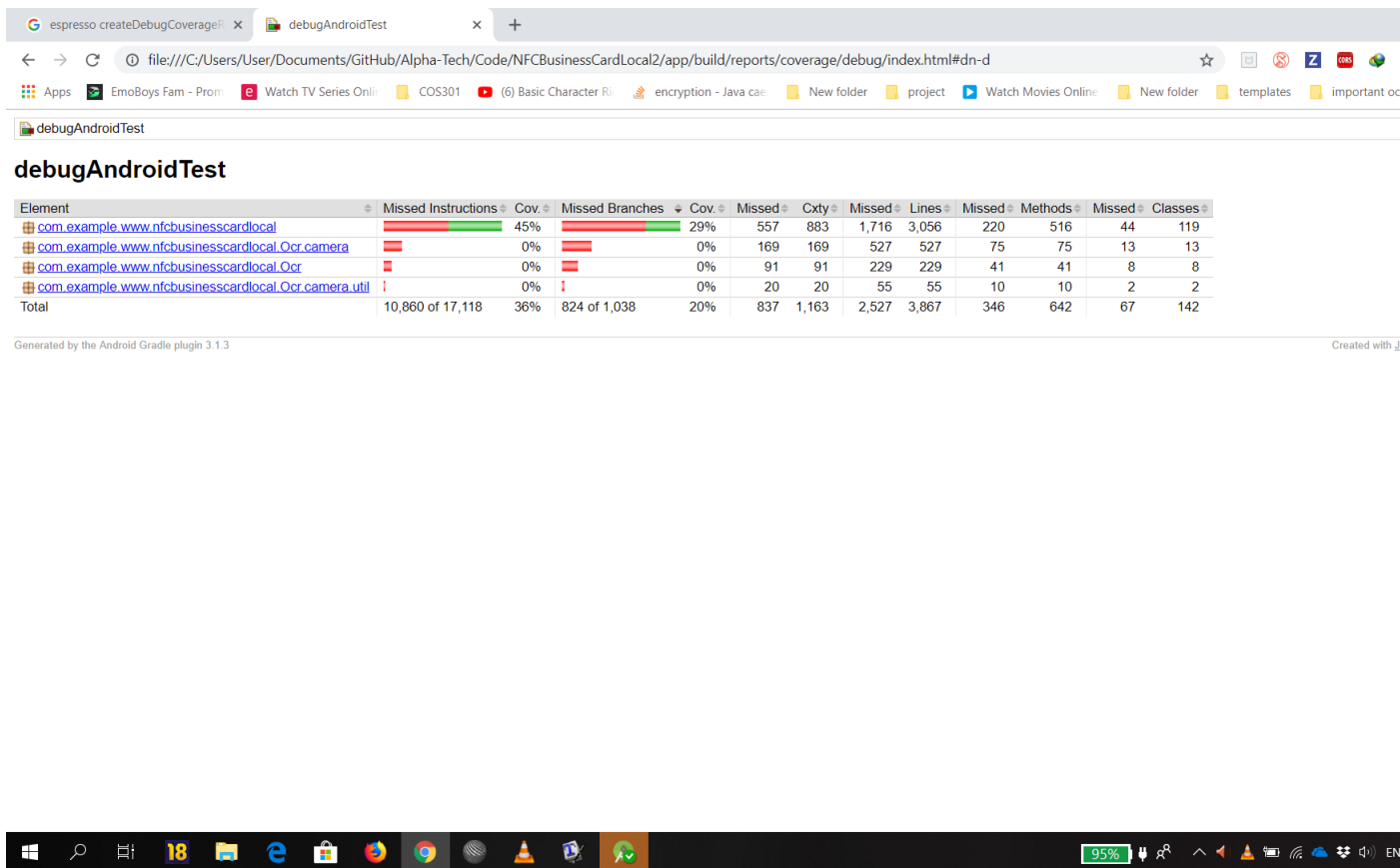


Figure 4: test coverage