# CA326 Third Year Project

## Testing Documentation for Card Connect

\*\*Abe Grogan - 17314893 and Luke Edgeworth - 17509283\*\*

# Unit Testing

For the app we decided to unit test the core components of the Database, NFC and UI.

Results for each test case are as follows.

### Testing of DataBaseHelper

\*Testing Database insert\*

![](./images/db-1.png)

\*Testing Database delete\*

![](./images/db-2.png)

\*Testing Database update\*

![](./images/db-3.png)

All tests ran with expected results

We also ran stress tests for the database itself on both an emulated Google Pixel 2 and a physical Huawei p8 lite 2017. While the performance while in the app was fine, there were various times it took to start the app, here are the examples, we worked from 0 entries to 2000 entries as for the context of the app it would be the reasonable extreme for the lowest and highest in the app.

#### Stress test

\*\*emulation - Google Pixel 2\*\*

2000 entries - 18.33s

1000 entries - 4.64s

500 entries - 2.83s

100 entries - 1.0s

0 entries - instantaneous

\*\*Real phone -Huawei p8 lite 2017\*\*

2000 entries - 10.23s

1000 entries - 3.90s

500 entries - 3.27s

100 entries - 1.06s

0 entries - 0.57s

### Testing of NFC

To test our NFC fragments, we emulated the application on our phone and ran the tests from our devices. We tested the fragments initially to see if there were any connection issues when an NFC card was tapped to it. We then tested the response of our read fragment to a blank NFC card, seeing if it would prompt the appropriate toast message or write the empty info to a new contact.

### Testing of UI

While there were no physical errors in the UI while we were developing, we still created tests using espresso to check if textview, editview and buttons existed, displayed the text they had and could be clicked. These test cases can be found in src/androidTest/java in the app code.

As part of testing the UI in our application, we used the app on multiple devices (emulated and physical): a Huawei p8 lite 2017, a Huawei honor 10, a Xiaomi Redmi note 8T and Google Pixel 2. We found that the apps scaled well to the different screen sizes for most screens. We noted however that on the smallest screen we tested (Google Pixel 2), that the View Contacts buttons (export, edit, delete) were quite cluttered. We reduced more padding around each to solve this issue. Screenshots of how the app scales on multiple devices are included below.

Huawei p8 lite 2017 <br/>

<img src=./images/p8\_1.jpg width="150"/>

<img src=./images/p8\_2.jpg width="150"/>

<img src=./images/p8\_3.jpg width="150"/>

<img src=./images/p8\_4.jpg width="150"/>

<img src=./images/p8\_5.jpg width="150"/>

<img src=./images/p8\_6.jpg width="150">

Huawei honor 10 <br/>

<img src=./images/honor10\_1.jpg width="150">

<img src=./images/honor10\_2.jpg width="150">

<img src=./images/honor10\_3.jpg width="150">

<img src=./images/honor10\_4.jpg width="150">

<img src=./images/honor10\_5.jpg width="150">

<img src=./images/honor10\_6.jpg width="150">

Google Pixel 2 <br/>

<img src=./images/pixel2\_1.jpg width="150">

<img src=./images/pixel2\_2.jpg width="150">

<img src=./images/pixel2\_3.jpg width="150">

<img src=./images/pixel2\_4.jpg width="150">

<img src=./images/pixel2\_5.jpg width="150">

<img src=./images/pixel2\_6.jpg width="150">

We also ensured to handle all possible errors. For example, when given an invalid information in the NFC Card (anything other than text or not in our specific format), the app prints an appropriate toast message.

<img src=./images/NFC\_read\_error.jpg width="150"/>

# User-Acceptance Testing

As part of our Testing for the project we surveyed a small sample of potential users to gain any feedback. For this, ethical approval was not required as the users were anonymous and the app didn't store or send personal user data, as well the users could use dummy data to transfer between phones. Here is the survey we gathered.

\*\*What do you think of the app design (user interface)?\*\*

1. I like it, it’s simple and straight to the point.

2. I like that everything is interactable and I don't have to dig deep into the app to find what I want.

3. I really like the contrasting colours of the buttons, which make it easy to see and know what to do.

4. Simple to understand the UI but a bit dull for my tastes.

5. While it was simple, I didn't know what the additional notes field was for in the Profile.

\*\*What do you think of the functionality of the app (profile, nfc and contacts)? \*\*

1. I like the concept, the nfc is easy to use and I could easily carry the card in my wallet.

2. Creating a profile with my contact info is useful for networking

3. Reading the nfc is simple, though i would want to write or read straight away.

4. I like the ability to edit, delete or export contacts, it can help separating my personal and business life.

5. It was a bit confusing at first, but I got a handle on it and was able to read and view contacts with ease.

\*\*On a scale of 1 to 10, how would you rate the app, 1 being disappointing and 10 being very good and tell us why.\*\*

1. 10, it’s a very inventive app!

2. 8, a good app but i’m not sure if i would use it alot as i’m not a businessman

3. 7, it seems a little slow to start the app and press the read button if i want to get someones information quickly

4. 8, the dialog boxes stuck and I had to press back to get back to the menu

5. 7, while it was simple and easy to use once i got into it, it was difficult to find the manual the testers mentioned and I had to ask where it was.

\*\*Are there any suggestions that you would have to improve the app? Please tell us what you would want and why.\*\*

1. No, it’s perfect as it is.

2. Bluetooth could be a good addition, as I use bluetooth more often and more people would have bluetooth on their smartphones as it’s an older technology.

3. I would like to read the nfc card from the menu as it’s quicker than pressing the button.

4. The Dialog boxes from reading and writing nfc could be removed after reading to make it less confusing for the user.

5. You could add a button on the main menu to access the manual as I think most people wouldn’t know where to access it from the toolbar. You could also change the instructions in the “additional notes” text field, as i didn’t know what its purpose was, so an example would be nice.

This was a very helpful exercise as it helped us to find things that user wanted, like a bigger button for the user manual and allowing the user to go to the app by just reading their NFC card, and fixing bugs such as the stuck dialog box. However, we couldn’t achieve other things users wanted due to infeasibility or time constraints (such as bluetooth functionality)

# Continuous Integration/Regression Testing

Due to the nature of our project, continuous integration was vital. This required coordination and consistent communication between us, which was useful as we would work together in some aspects using pair programming or discussing our own individual work. We mostly worked on the WIP branch for our project, though in hindsight we should have used merged to the master more often in case of a failure that could have lost elements of our project.

We would also perform regression testing, where with each feature we added, we would run it with the rest of the app so that we could see if there were any unwanted changes or crashes. This was a very effective method as any potential bugs became apparent as the feature was added, leaving us with minimal bugs to fix once we achieved full functionality. We would also check if our functions were integrated by examining the transfer of information between them, such as exporting contacts from the profile activity into the database.