Mobile Platform Application Testing

1 User Testing

1.1 Introduction

This section will look into the testing of the mobile application by recurring participants within different stages of development. Prior to the development of the application, initial user testing was performed against mock designs of the app. The results of the initial session gathered within the application proposal provided a template design for the application development.

Screenshots of changes and a user testing the application is found in Appendix A.

1.2 Participants

The same volume of participants will be used to carry out the user testing of the application within the development phases. Due to unforeseen circumstances, the participant group used for testing were students because of the ease of being able to test the application one to one. As mentioned previously, initial user testing was conducted against a prototype design without any development.

Participant A: University Student, Female, Psychology
Participant B: University Student, Male, Media

1.3 User Testing Results

1.3.1 First Session

For the first session of user testing, participants were asked to test the application within the earliest stages of the application. The first pieces of functionality that were implemented was authorisation (Sign in and Sign up) and had navigation to entries, graphs and had implemented the user profile section. This was important testing since users were able to use the application with some main functionality being registering an account, signing in and viewing their profile.

Participant A

Test	Pass/Fail	Improvements	Feedback
Registering an account on Sign Up page	Pass	- Possibly include some iconography for fields to	Registration worked as expected and did not fail.

		make the view user friendly. - Make selecting a gender a drop-down rather than checkboxes.	The layout is good and responsive. Great to see field validation when fields are empty or not in the correct format.
Sign in to the application with registered email and password	Pass	N/A	Signing into the application was straight forward, just requires email and password.
Navigate through application	Pass	N/A	Navigating through each page was straight forward, nice iconography used for each pane to separate features although no content for 'home' or 'graphs'.
View account profile	Pass	- There is a noticeable lag with details being loaded into the accounts page.	Viewing the profile page shows the details of the user as expected apart from the password. There is a very short delay in retrieving user details.
Update account	Pass	- Make updating details more noticeable when opting to update user details.	Updating the user account works. However, there is no way to see if updating an email works since there is no way to log out. What would be useful is a way to make it evident that fields are editable upon clicking edit.
Delete account	Pass	N/A	Works as expected. The account is deleted and cannot be accessed.

Additional Comments

Considering that this app is still in its early stages, the layout and navigation are quite good. So far with the minimal functionality that has been implemented, everything seems to have passed with the caveat that certain features could have been tweaked to improve usability. Certain features that strikeout as good are the abilities to select the date of birth through a calendar pop out which is user friendly, plus the feature to hide and uncover entered passwords. Simple, but effective features. Error validation present in input fields is good, not over the top. It would be useful to have included the ability to log out to test the updating of accounts. Accessibility is good, like the navigation at the bottom of the screen, it is easy to navigate with thumbs. At this current stage, not much functionality in terms of the main purpose but has pretty good functionality. One thing that could also be improved is that whenever the application is closed, the user needs to sign-in again rather than remain logged in. It could be better to remain logged in.

Participant B

Test	Pass/Fail	Improvements	Feedback
Registering an account on Sign Up page	Pass	- There is an issue with registering an account with a password that is lower than a certain number. Popup says 'Account already exists'. - Think it would be good to navigate onto the main application upon successfully registering an account. - A spelling mistake in terms and conditions.	Registration works well however a small error found was when registering an account with using a password lower than a certain number of characters. Validation is good i.e. pop-ups and error highlighting on fields.
Sign in to the application with registered email and password	Pass	- Could possibly include an option to reset a user's password if forgotten.	Signing into the application works as expected. Possibly include a forgotten password link.
Navigate through application	Pass	N/A	Navigation was simplistic, plain on pages with nothing implemented.
View account profile	Pass	- A little delay in loading the user's details on the profile page.	Profile page follows the same design as the sign-in/up pages which is consistent. A noticeable delay on the profile page, not critical.
Update account	Pass	- Possibly looking at the use of a floating button for editing and saving a user's profile. - Gender field is still editable after updating a user account.	Updating the user account works, simple but could move the edit/save button to a floating button.
Delete account	Pass	N/A	Deleting the account works and navigates back onto the sign-in page. When entering deleted account email and password, an error pop-up displays that account does not exist.

Additional Comments

Having tested the application within the earliest stage of development, it is safe to say that functionality works as expected. Appreciated the consent to store data within the terms and conditions. Missing an 's' and the end of conditions. Was expecting a way to log out of the application since there is no way to get back onto the sign-in page. A bug that I found within the application on a whole was rotating the screen horizontally would cause the application to exit, not sure what the issue is but would be nice to fix. Quite positive so far, just requires the rest of the features to be implemented. Worth taking into account improvements stated i.e. password with a low amount of characters suggests that the email is already used for registering a new account. Final positives are that the text is easily readable and the colour theme works well.

Actions based on the first session of user testing

After reviewing the results from the first test session, it is evident that whilst there was a lack of functionality, the features present function correctly and are easy to understand. A common issue for both users was the lack of a 'logout' button which would have been good to demonstrate the update of an account email, therefore the functionality for logging out of the application was implemented. A couple of bugs with the application on a whole was that rotating the app horizontally would trigger the app to crash and that quitting the app would require the user to authenticate into the application again. In regards to these bugs, the first app modification was to restrict the orientation to portrait within the manifest file to remove the issue with the app exiting upon rotation. Another action done was to add a check within the authentication activity to see if the user is still logged into firebase, thus allowing the user to remain logged in. These updates were considered massive as it would affect the usability of the application.

Other actions involved including iconography within the user sign-in/up as input fields did look rather plain, this way it makes the UI more user friendly. One feature that changed significantly was the loading of user details. Upon navigating onto the profile page, an async call to Firestore would load the details within the profile fragment, hence causing a delay in setting the fields with data. To minimise this delay, the details are stored within the main activity and loaded into the profile fragment. An event listener is also placed to update the activity if the account details have changed to account for profile updates. As identified, there was an error when registering an account with a short password. Looking further into this, Firebase does not support passwords shorter than 8 characters long. Therefore a change to the validation was done. Other small modifications have also been addressed.

1.3.2 Second Session

For the second session of user testing, the application had been updated. New features such as logging out, recording an entry tracking their mood, thoughts, location and picture, and then viewing the entry were implemented. Also, actions based on the results of the previous user testing session were addressed and shown to participants. The progress from the last session is significant as it provides a key piece of functionality within the application.

Participant A

Test	Pass/Fail	Improvements	Feedback
Log out of application	Pass	N/A	Does what is expected, logs out of the application and directs onto the sign-in page.
Add emotion to entry	Pass	- Possibly make it so that when selecting a different emotion that you do not have to deselect the previous emotion. - Possibly remove the neutral emotion and replace it with stress.	Selecting an emotion is easy and evident with an emoji present, tag for emotion and checkbox. Really like the use of different colours to separate the emotions. Selecting and deselecting an emotion could be more user friendly when changing emotion.
Add thoughts to entry	Pass	N/A	Able to write a thought into the thought box and is an adequate size. Nice to know that is optional to record thoughts.
Add location to entry	Pass	- Possibly return the current location within the search bar.	Worked as expected, able to record the location when feeling a particular emotion. Nice to have the option to search location, use current location or have none at all. Not critical but would be nice to return the current location with search bar when clicking on current location.
Add a picture to an entry	Pass	- Possibly increase the size of the image view on the entry.	Able to select a photo from device storage, or take a picture from the camera. This is a neat feature if wanting to use the photo as a reminder of something that happened within the entry. Only negatives would be the size of the photo when viewing it on the entry.
View Entry	Pass	- Possibly have the option to filter entries based on emotions.	As soon as an entry is added to entries, a card with emotion is present including date/time plus thoughts. This is a cool feature because a user

			can view an entry by clicking on the card. One thing that would improve usability would be the ability to view entries based on an emotion i.e. Happy.

Additional Comments

Reflecting back on the updated features, a couple of things have changed from existing functionality. The use of iconography within sign-in/up pages are easy to understand and user friendly, nice to notice that gender options have been changed from checkboxes to a dropdown box since it does not clutter the screen. Other small improvements suggested can be seen from the previous section which is good. Adding an entry and viewing said entry is quite neat, quite user friendly and I like the different colour for each emotion. One thing that would benefit the app is that if the app is targeted towards students and workers, it would be good to record 'stress' as an emotion rather than having a neutral emotion. Saying this, usability could be improved by filtering the emotions within the view entries page and when selecting an emotion you do not have to deselect another emotion manually to select another. So far the app is starting to have some functionality but still is missing the graph features, apart from that looking very consistent and professional looking.

Participant B

Test	Pass/Fail	Improvements	Feedback
Log out of application	Pass	N/A	Logs the user out as expected.
Add emotion to entry	Pass	- Possibly make emojis clickable for selecting the emotion.	The use of colours and emojis to represent emotions is great. Whilst it is easy to identify which emotion has been selected i.e. checkbox it would be easy to also be able to click the emoji to select the emotion rather than checkbox. Like the pop up if no emotion has been set.
Add thoughts to entry	Pass	N/A	Simple text box to write thoughts.
Add location to entry	Pass	- Possibly remove shadow around search bar border as it seems a little different to other fields on page.	Recording location works well, nice to be able to give the user the option on how they want to record location. Enjoyed the animation as well.

			Only suggestion is to probably remove shading around the search box as it is not consistent with the thoughts text box.
Add a picture to an entry	Pass	N/A	Like the ability to be able to add a picture to the entry from the device gallery, or through taking a picture from the camera.
View Entry	Pass	- Possibly list the entries in chronological order. - Possibly add some graphics to say if there are no entries present add an entry.	Viewing entries does work, but not very appealing when there are no entries present - just a blank view. Another thing to point out would be viewing records chronologically since it is quite strange to have the dates in a random order.

Additional Comments

So far, the app is coming along. Really do like the consistent theme and layout at times. The new updates from the previous session have made a difference such as having the feature to log out and loading the profile page pretty much has negligible lag. In terms of the new adding entries and viewing entries, everything works well and is rather great in terms of usability. A couple of little updates would improve overall user experience however pretty good, especially now with some core functionality. A little nice update on the view entries page would be to chronologically order the entries by date and time.

Actions based on the second session of user testing

After reviewing the feedback from the second session of user testing it seems that both participants appreciated the changes made in response to the first session. Although critical updates were added, some smaller 'nice to haves' have been noted and will be addressed after all functionality has been implemented. In terms of improvements within this iteration of user testing, some stand out features were to do with the layout of the entries page, and usability within the add entry page. One action that was taken was to ensure that entries are listed in chronological order. This was at first a hard task considering that date and time are stored as strings, but a way to solve this issue was by storing the timestamp of the entry within Firestore so that entries can be filtered by timestamp which gives a long that can be compared against each entry. This was done and worked effectively. Going forward within the entries, filtering based on emotions was also implemented. An update to the entries page that allowed filtering was storing all the entries within a list and being able to manipulate that list within a recycler view. This meant that the recycler view can change based on emotions selected within a drop down box. Finally, selecting an emotion for an entry was not as user friendly since a user had to deselect the checkbox before selecting a new emotion. To improve this feature, an on click listener was added to each checkbox to mimic the behaviour of only being able to

have one emotion set. As mentioned previously, small updates will be addressed as soon as the application has the requirements needed for a minimum viable product.

1.3.3 Third Session

For the third session of user testing, the final functionality required for a minimum viable product. The final features included was viewing emotion statistics within different graphs. The graphs feature was the final feature required to complete the app meaning that participants can fully test full functionality of the app and can suggest final improvements before the final release. As mentioned in the second user session, there were updates done within viewing the entries and adding an entry.

Participant A

Test	Pass/Fail	Improvements	Feedback
Filter emotions in view entries	Pass	N/A	Filtering emotions works as expected. Nice functionality since able to filter specifically on an emotion and view the entry.
Update a recorded entry	Pass	N/A	Updating an entry works as expected. Good to be able to change location, thoughts and emotion so easily.
Delete a recorded entry	Fail	- An entry can be deleted if a photo is selected but can not be deleted if a photo is not stored.	Functionality for deleting a record works when an image is attached to an entry, however when trying to delete an entry without an image - the delete button does not work.
View emotions plotted on a map	Pass	- Possibly add a little info on each marker on the map to give a little info on why the emotion was set at a particular location.	The map feature plotting all recorded emotions on a map is such a cool feature. From the animations to the emojis marked on the map, really engaging. Only minor improvement would be to show a little info about the emotion when clicking on the markers i.e. thought for that entry or date and time.
View stats on charts	Pass	N/A	The use of a bar and pie chart is really appealing

			to the user. Especially animations used, makes viewing statistics of emotions recorded rather user friendly.

Additional Comments

Now that the app has all functionality implemented, it is good to be able to test the app with all features. Most of the new features implemented work, the only thing that did not work was deleting an entry without a picture which is something that should be fixed considering that a user can choose to include a picture to an entry or not. Apart from that the graphs functionality is a really good way to reflect on emotions, especially the map feature. Overall, the app is easy to navigate through and rather user friendly. There are still a couple more little improvements that should make the final release, but the app still looks professional. One thing to consider is the use of emotions, if the app is aimed towards education or workers then including emotions like stress would be better emotions to record.

Participant B

Test	Pass/Fail	Improvements	Feedback
Filter emotions in view entries	Pass	- There is a slight bug that duplicates the number of entries flicking in between adding an entry and viewing entries. - Possibly rename home to entries.	The filtering works well. Selecting an emotion or all filters as expected. Slight bug when flicking in between adding an entry and being directed back to the entries page causes duplication of entries.
Update a recorded entry	Pass	N/A	Can update recorded entry and updates are saved straight away.
Delete a recorded entry	Fail	- Bug trying to delete entry with image.	Deleting entries works for an entry with an image attached, however not for entries without an image.
View emotions plotted on a map	Pass	N/A	Viewing emotions scattered across the map if a location is recorded is a really unique way of viewing stats on recorded emotions.
View stats on charts	Pass	N/A	Nice use of graphics in the form of a bar and pie chart to represent the recorded emotions. Along

			with animations makes the features appealing to the user.

Additional Comments

So far, the app is coming along. Really do like the consistent theme and layout at times. Now that the app is in the final stages before release, it is good to see the majority of functionality implemented works well - minus the deletion of entries. There are a couple of things that could be improved. In terms of usability, a user would have to navigate onto 'Home' to then add an entry when it could be easier to have a navigation tab to add an entry within the bottom navigation as it removes the extra step. Another thing would be to rename home to entries since it is a page holding entries. Other than that there are smaller features that could be implemented before a 'first release'. A really big positive is the graph's features, really useful and creative ways of displaying stats on emotions recorded.

Actions based on the third session of user testing

After reviewing the feedback from the third session of user testing, it is quite positive that the final features received good feedback along with overall feedback on the app through each session. One critical piece of feedback was that the delete entry functionality was not implemented correctly. One of the first actions based on the recent user testing session was to update the functionality of deleting an entry for a user. Currently, the logic makes an async call to delete a picture from Cloud Storage, then proceeds to delete the entry from firestore which is the reason for deletion of entries failing if a picture is not present. To resolve this issue, a check to see if an image is stored was implemented and regardless if there is an image or not, delete the entry. If an image is present, delete image then entry. Another action taken was addressing the duplication of entries upon switching between viewing entries and adding entries. The issue was that after adding an entry, the user would be directed back to the entries page through popping back stack. After identifying this as the issue for the duplication, the solution was to navigate back onto the entries fragment rather than popping backstack. With these updates done, there are no more visible functional bugs.

Since the recent user session was tested against the application with features required for a minimum viable product, it was necessary to review all improvements that had been suggested that were not considered critical to the functionality of the app, more towards usability, hence further actions taken were to review improvements and implement if possible.

1.3.4 Final Session

The fourth installment of user testing takes into account the updates from the third session, along with mini updates that have been added as a result of feedback from previous installments. One of the big feature updates was the deletion of entries which failed previously and the duplication of entries. Within this testing session, the users will review the app on a whole with improvements made.

Participant A

Test	Pass/Fail	Improvements	Feedback
Delete an entry	Pass	N/A	Previously, an entry with a photo could be deleted but without a photo could not be deleted. Update to functionality allows the feature to pass.
Select emotion on emoji	Pass	N/A	Nice to be able to select an emotion based on emoji, including the fact that stress is a new emotion. Improves usability.
Filter entries based on emotion	Pass	N/A	Works as expected, now includes emotion for stress which is nice.
Forgot password	Pass	N/A	Nice new feature for users to be able to reset password if forgotten. Works as expected and sends a password reset in the form of an email. Also nice validation if email does not exist.
View info on entries on map	Pass	N/A	Viewing emotions on a map was a good feature that worked, and it is nice to see that previous improvement on viewing info on each emotion has been done. Makes it nice to understand why a person was feeling a specific emotion within an area on the map.
View entries	Pass	N/A	Viewing entries works, nice improvement to have additional info such as the emotion and the location if recorded. Also, colours added to each emoji to match the add entry page is good.
Log out	Pass	N/A	Log out works as expected but is more accessible for user in the top right corner of the screen.

Additional Comments

Overall, the app is quite easy to use and adds a unique way of viewing emotions with a diary in comparison to other methods. One of the biggest features has got to be the recording of emotions and statistics of emotions plotted against a map and charts, very insightful and appealing to the user. One thing overall that could have been improved was possibly filtering between dates, that would have been good to grab an insight on feelings within a specific date range. The functionality is really good, not over complicated making it easy for a user who is not an avid user of tech to understand. Really great to see previous improvements being acted on, especially the update to display information on the emotion on the map and replacing the 'neutral' mood with 'stress' is great for gauging stress, especially towards users who are dealing with stress. The change of the view entries page was also a nice touch, user friendly, engaging and provides the user with much more information in comparison to just having an emoji and a thought within a box. Even going onto the entry page and viewing small little updates such as the ability to click on emoji's to select emotions and having addresses saved when clicking current location improves usability. For further improvements, possibly adding more colours may make the app striking, but it is completely fine as it is right now, and possibly adding the feature to select what emotions a user wants to record.

Participant B

Test	Pass/Fail	Improvements	Feedback
Delete an entry	Pass	N/A	Deleting entry for entries without an image now works. Delete now works fully.
Filter entries	Pass	N/A	Slight bug from the previous session that duplicated entries are no longer present, and renaming of 'Home' to 'Entries' makes the page easier to identify.
Forgot password	Pass	N/A	New feature to reset password on sign-in page works as expected.
View info on entries on map	Pass	N/A	Works as expected and user friendly if the user wants to remember what they were feeling.
View entries	Pass	N/A	Massive change in design, much more user friendly and appealing. Use of colours and extra info improves the functionality a lot in comparison to the boring and non informant previous design.

Select emotion on emoji	Pass	N/A	Clicking on emoji selects emotion which improves accessibility.
Log out	Pass	N/A	Logout moved from profile page to top of the app bar which is easier than having to navigate onto profile page and clicking logout.

Additional Comments

As the app is in its final stage of development, general comments would be that navigation is easy throughout and that functionality is all working. Personal favourites are the graphics used within charts, feature displaying emotions on a map based on recorded entries alongside info and the range of things that can be recorded within an entry. Changing the navigation bar by renaming 'Home' to 'Entries' and 'Graph' to 'Stats' was easier to identify what each page is about and represents features on each page. The addition of an 'Add Entry' page makes it easier to access the feature. A couple things to improve upon would be the use of a couple more graphics to make the app a bit more appealing i.e. a logo. Also possibly renaming the top of each page with a different name rather than 'MyEmotion', it is not critical but would engage user user attention. A little improvement would also be making the edit profile a bit more noticeable. Otherwise, the app is good all round. Functionality definitely is there, and manipulation of data recorded is presented well.

Actions based on the fourth session of user testing

After the final user session, it is evident that users found the updates positive. The final actions that will be taken is adding a logo onto the sign in screen and adding a couple of toast messages to improve usability.

1.4 Evaluation of user testing

Going through multiple sessions of user testing within the development of the mobile application was crucial in terms of improving the application and ensuring that all areas of the application had been tested before the first release of the application. Having 2 participants to review the app provides a meaningful and raw insight into both functionality and design of the app. Whilst testing can be undertaken by developers, perspective may be skewed since they developed said features and knows how features work. Due to unforeseen circumstances, user testing was performed by students studying at university. A positive note is that both participants have their own useful insights on the app, one from a psychology background, and another from a media background which provided useful insights on what could be improved or removed. Whilst testing for if a feature passes or fails, the user testing also provided beneficial information which also matters such as if the design was user friendly or not, or if the feature could have been improved - more gauged into what an end user would want to see as opposed to an engineers development of just working functionality. Feedback was generally positive overall, especially with the improvements being implemented as a result of feedback from end users. Having user testing at different stages, whilst might

be time consuming at first, prevents further time wasted towards the end of an application i.e. updating normal android TextEdit views to InputTextEdit views to match material design.

2 Unit Testing

2.1 Introduction

This section will look into unit testing within the mobile application using Junit. Unit testing is an important aspect of software engineering as it determines whether the logic implemented performs as expected. As part of Test Driven Development, unit tests are developed to structure how the engineer expects the logic to work ensuring that functionality works as expected minimising error debugging after implementing said functions.

Unit testing examples provided within this section show examples of important classes being tested to ensure that results are expected.

2.2 Unit Test Registration Validation

Screenshots of unit testing for the ValidateRestigration class have been attached within Appendix B. The ValidateRegistration was developed for handling the validation of user registration. The reason for testing this is due to registration being an important feature that allows users to register an account within the application. For the validation class to be successful, both positive and negative tests need to be accounted for.

For each input field within the user registration, there is a corresponding validation method within the class. Each method returns a ValidationMessage that contains a boolean if the input is valid, and String being the validation message that will be displayed to the user if incorrect. This means that both valid and invalid input will be tested against to see if the correct validation messages are returned.

The unit test suite for ValidateRegistration performs assertions based on validating input. Developing the unit tests within the beginning took time to write out the logic in a way that would pass each test. Providing ‘fake’ input data would prove whether functionality was working. The results of unit testing proved to be successful with 20 unit tests passing out of 20, and has a coverage of 100% for the ValidateRegistration. This ensured that validation was working successfully before integrating within the user registration ensuring that registration validation is covered.

The results of these unit tests ensured that validation was implemented correctly, as well as ensured that a user was not able to register an account with erroneous data.

2.2 Unit Test Emotion Marker

Screenshot of unit testing for the EmotionMarker class has been attached within Appendix C. The EmotionMarker was developed for plotting emotions on the map for animated statistics of emotions plotted around the map. The reason for testing this is

due to entries being able to record an emotion, but some entries may not have a thought or location attached.

For each entry, an emotion has to be recorded. However, thoughts and location are optional. The EmotionMarker will return a marker with entry information if present which will include a position, title and snippet.

The unit test suite for EmotionMarker performs assertions based on whether a marker has been set with a position, title and/snippet based on variables that would be gathered from an entry. All unit tests pass and through identifying new test cases, the logic was changed to account for negative path options. Test coverage was 100% for the EmotionMarker class. This resulted in emotions being marked on a map.

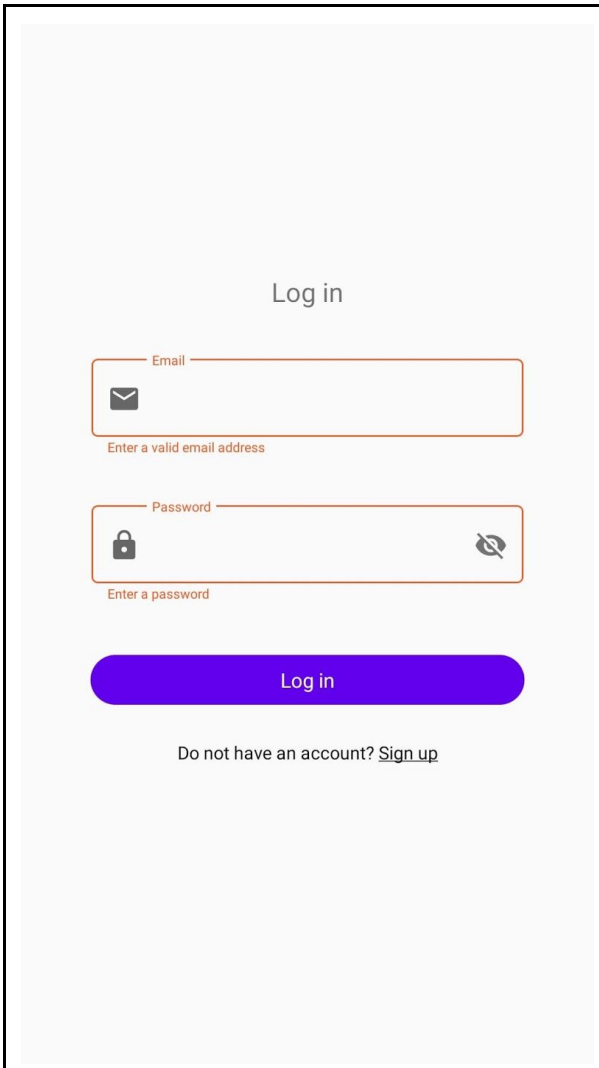
At first, the original way of implementing functionality rendered markers with empty snippets and was not taken into consideration if a position was not set. Through unit testing, markers were able to be set correctly alongside having validation included.

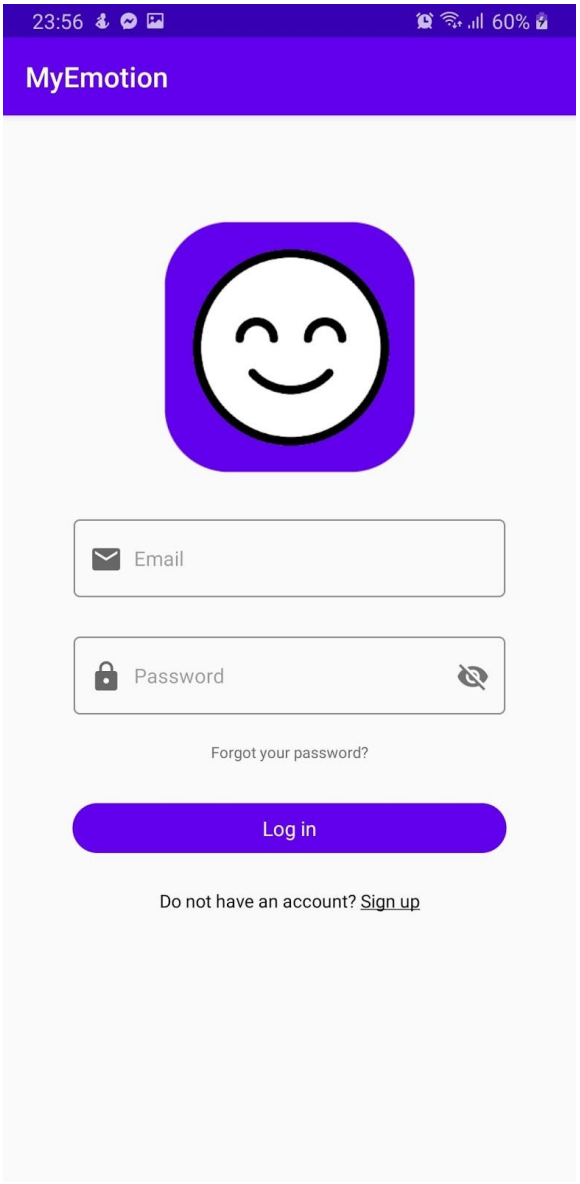
3 Appendices

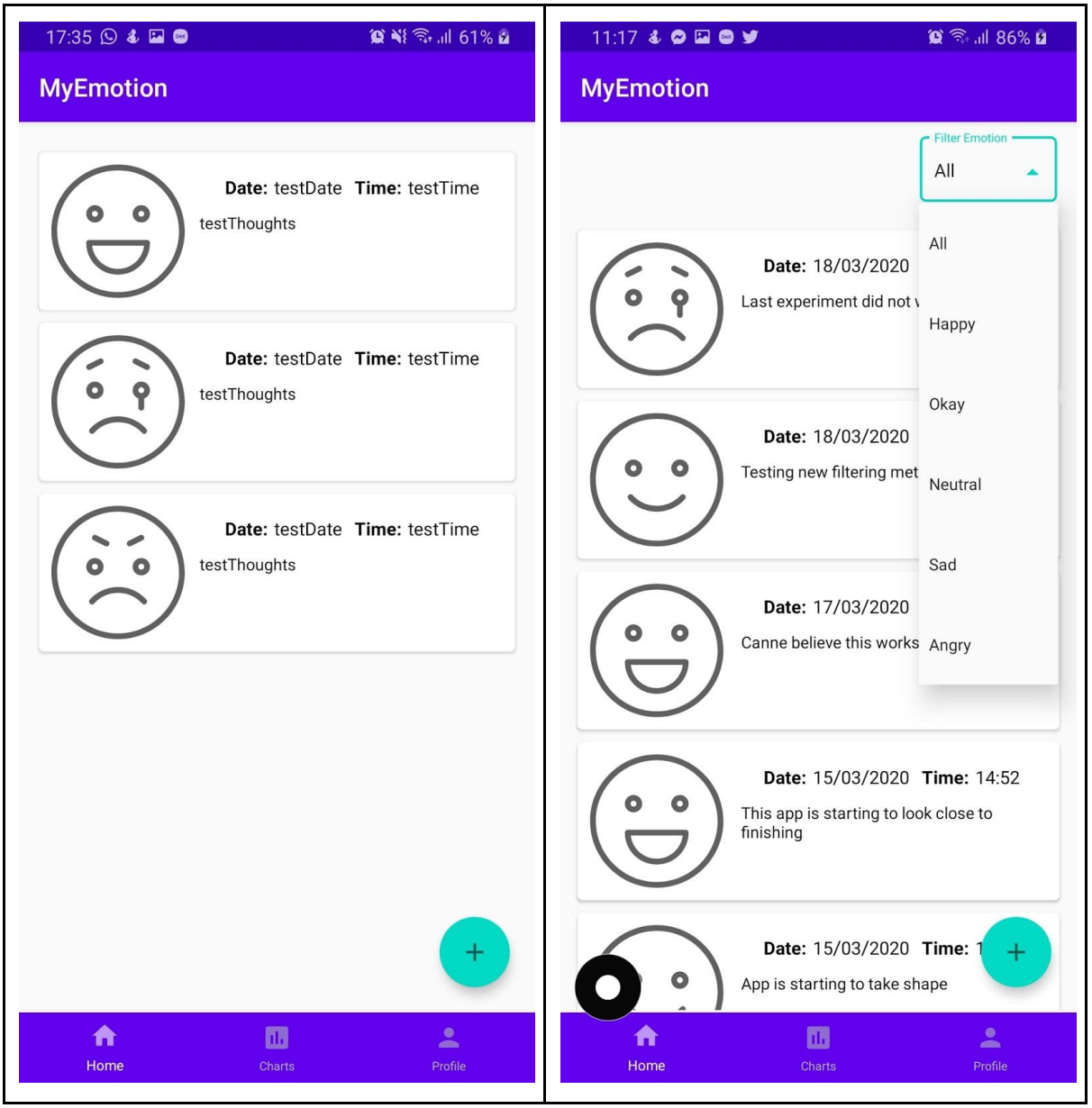
3.1 Appendix A

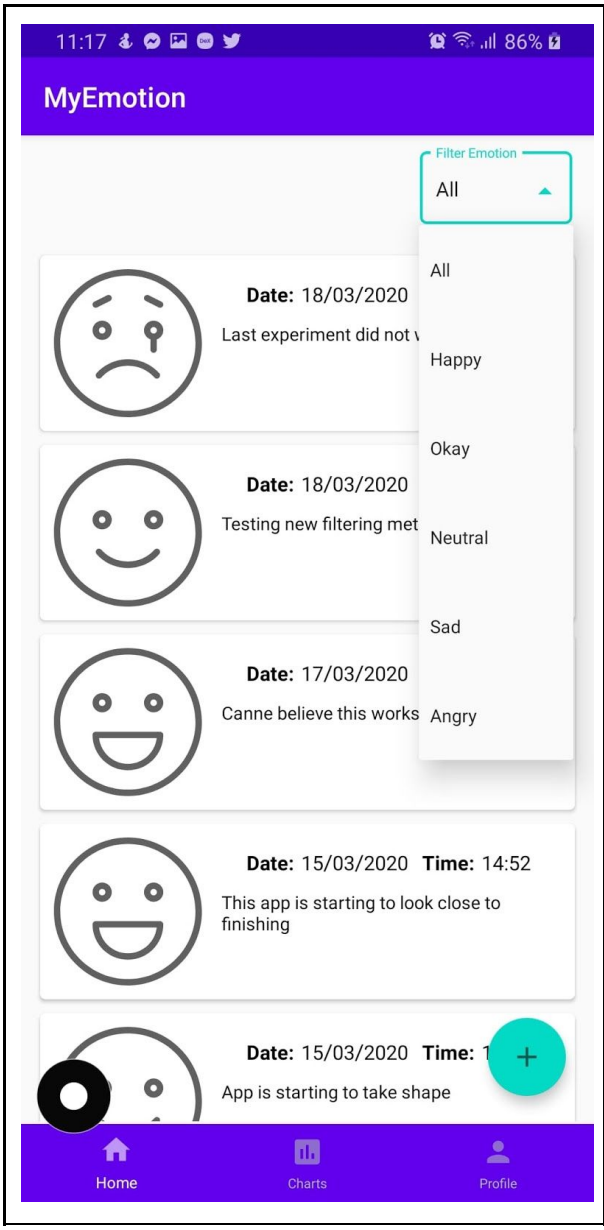
User Testing Changes

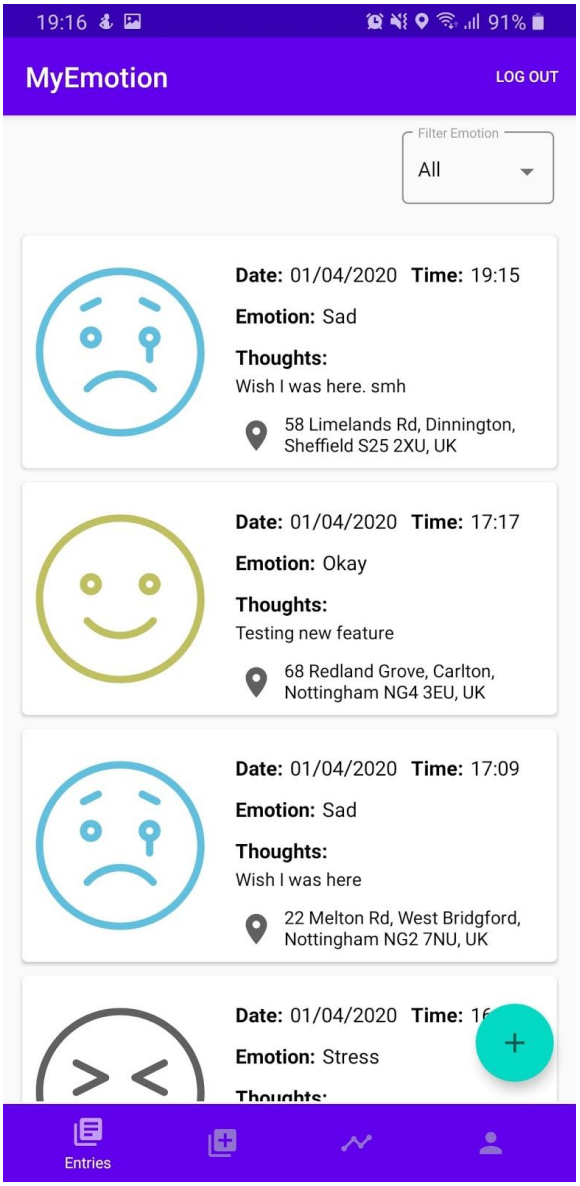
Before	After

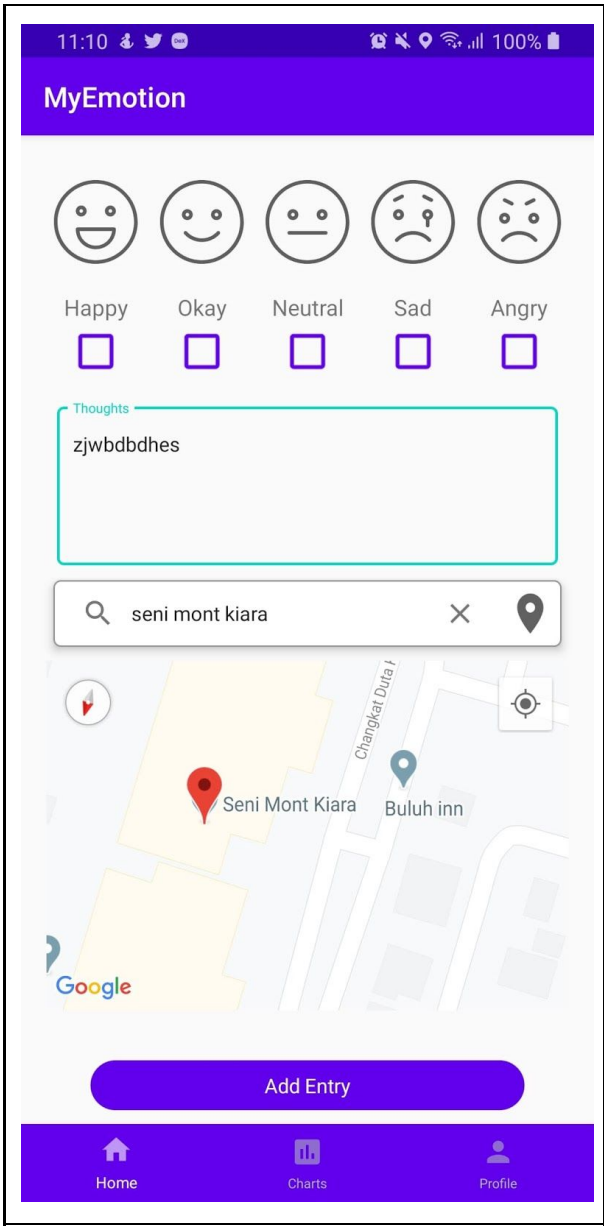

Log in
Email
Enter a valid email address
Password
Enter a password
Log in
Do not have an account? Sign up

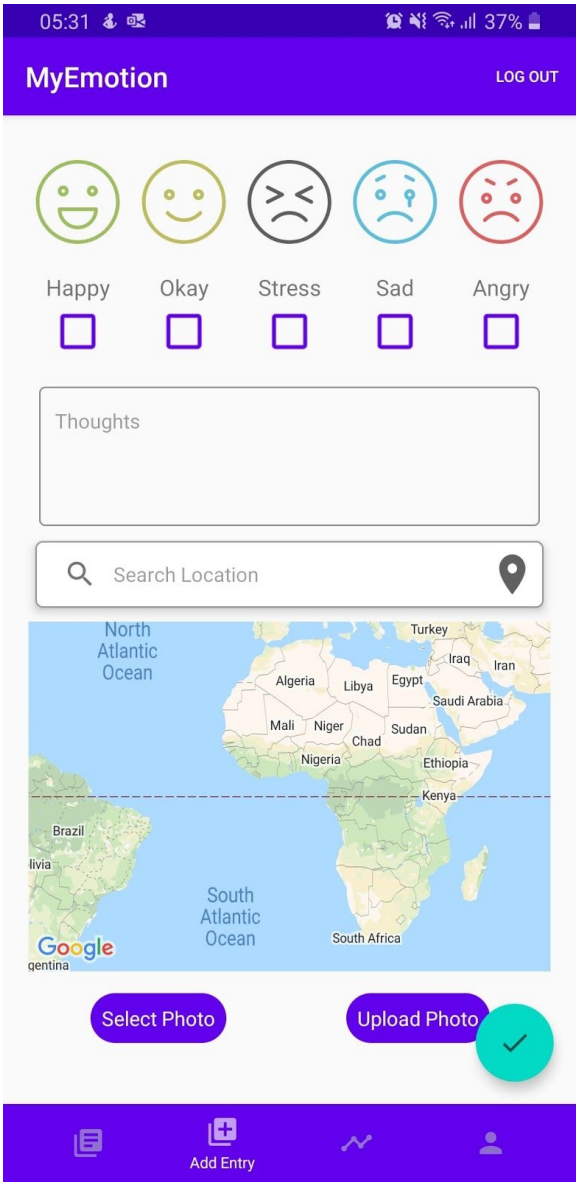

23:56
60%
MyEmotion
Email
Password
Forgot your password?
Log in
Do not have an account? Sign up

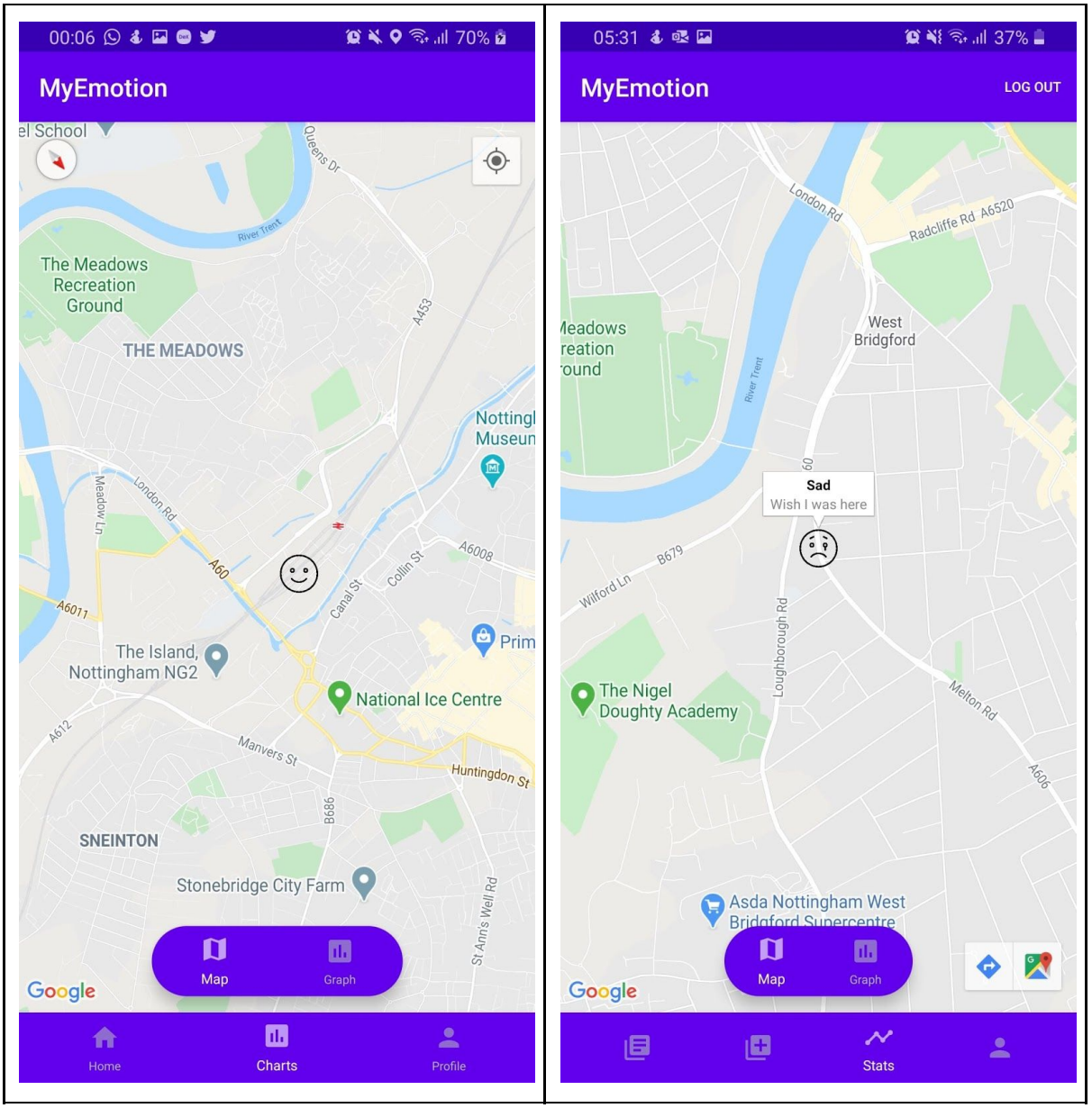

17:35
61%
MyEmotion
Date: testDate Time: testTime
testThoughts
Date: testDate Time: testTime
testThoughts
Date: testDate Time: testTime
testThoughts
Home
Charts
Profile
11:17
86%
MyEmotion
Filter Emotion
All
All
Happy
Okay
Neutral
Sad
Angry
Date: 18/03/2020
Last experiment did not
Date: 18/03/2020
Testing new filtering met
Date: 17/03/2020
Canne believe this works
Date: 15/03/2020 Time: 14:52
This app is starting to look close to finishing
Date: 15/03/2020 Time:
App is starting to take shape
Home
Charts
Profile


11:17
86%
MyEmotion
Filter Emotion
All
All
Happy
Okay
Neutral
Sad
Angry
Date: 18/03/2020
Last experiment did not
Date: 18/03/2020
Testing new filtering met
Date: 17/03/2020
Canne believe this works
Date: 15/03/2020 Time: 14:52
This app is starting to look close to finishing
Date: 15/03/2020 Time:
App is starting to take shape
Home
Charts
Profile


19:16
91%
MyEmotion
LOG OUT
Filter Emotion
All
Date: 01/04/2020 Time: 19:15
Emotion: Sad
Thoughts:
Wish I was here. smh
58 Limelands Rd, Dinnington, Sheffield S25 2XU, UK
Date: 01/04/2020 Time: 17:17
Emotion: Okay
Thoughts:
Testing new feature
68 Redland Grove, Carlton, Nottingham NG4 3EU, UK
Date: 01/04/2020 Time: 17:09
Emotion: Sad
Thoughts:
Wish I was here
22 Melton Rd, West Bridgford, Nottingham NG2 7NU, UK
Date: 01/04/2020 Time: 16
Emotion: Stress
Thoughts:
Entries

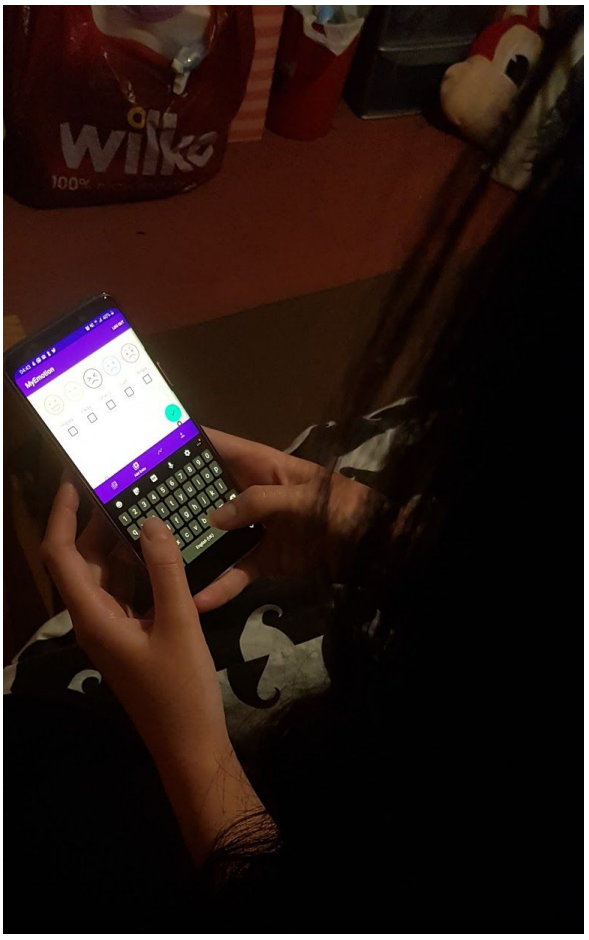

11:10
100%
MyEmotion
Happy
Okay
Neutral
Sad
Angry
Thoughts
zjwbdbdhes
seni mont kiara
Changkat Duta
Seni Mont Kiara
Buluh inn
Google
Add Entry
Home
Charts
Profile

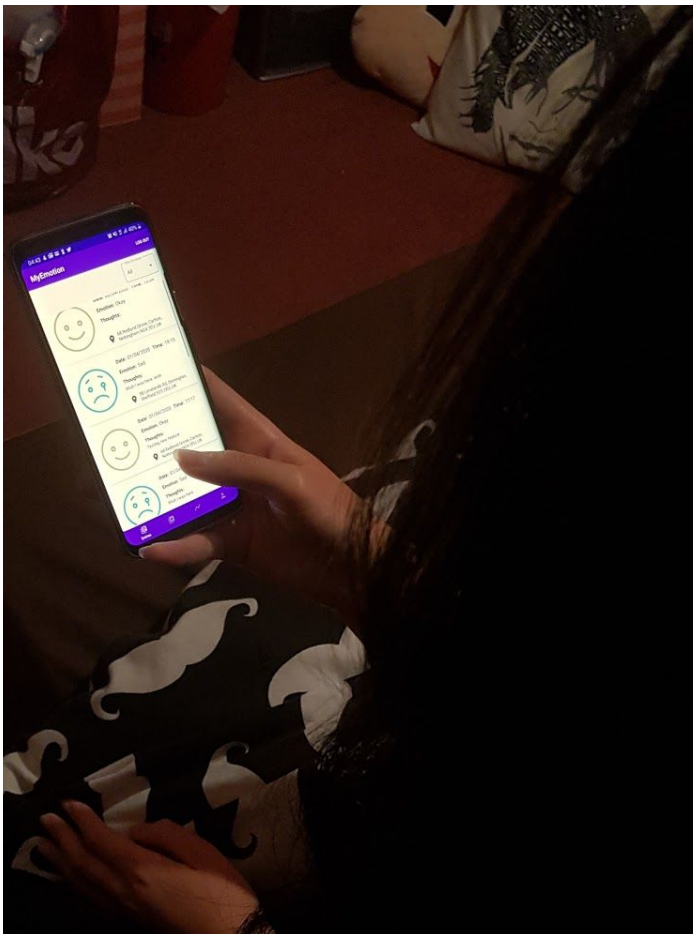

05:31
37%
MyEmotion
LOG OUT
Happy
Okay
Stress
Sad
Angry
Thoughts
Search Location
North Atlantic Ocean
Turkey
Iraq
Iran
Algeria
Libya
Egypt
Saudi Arabia
Mali
Niger
Chad
Sudan
Nigeria
Ethiopia
Kenya
Brazil
South Atlantic Ocean
South Africa
Google
Select Photo
Upload Photo
Add Entry



User Testing Evidence

Permission for use of photo within document was allowed by participant.





3.2 Appendix B

```java
public void validateEmptyFirstName() {
    String fakeFirstName = "";
    validateRegistration.setFirstNameVal(fakeFirstName);
    ValidateRegistration.ValidationMessage sut = validateRegistration.validateFirstName();
    assertFalse(sut.getCheck());
    assertEquals( expected: "Field must not be empty", sut.getMessage());
}

@Test
public void validateInvalidFirstName() {
    String fakeFirstName = "12345";
    validateRegistration.setFirstNameVal(fakeFirstName);
    ValidateRegistration.ValidationMessage sut = validateRegistration.validateFirstName();
    assertFalse(sut.getCheck());
    assertEquals( expected: "First name must contain letters", sut.getMessage());
}

@Test
public void validateValidFirstName() {
    String fakeFirstName = "Test";
    validateRegistration.setFirstNameVal(fakeFirstName);
    ValidateRegistration.ValidationMessage sut = validateRegistration.validateFirstName();
    assertTrue(sut.getCheck());
    assertEquals( expected: "", sut.getMessage());
}

@Test
public void validateEmptyLastName() {
    String fakeLastName = "";
    validateRegistration.setLastNameVal(fakeLastName);
    ValidateRegistration.ValidationMessage sut = validateRegistration.validateLastName();
    assertFalse(sut.getCheck());
    assertEquals( expected: "Field must not be empty", sut.getMessage());
}

@Test
public void validateInvalidLastName() {
    String fakeLastName = "12345";
    validateRegistration.setLastNameVal(fakeLastName);
    ValidateRegistration.ValidationMessage sut = validateRegistration.validateLastName();
    assertFalse(sut.getCheck());
    assertEquals( expected: "Last name must contain letters", sut.getMessage());
}

@Test
public void validateValidLastName() {
    String fakeLastName = "Test";
    validateRegistration.setLastNameVal(fakeLastName);
    ValidateRegistration.ValidationMessage sut = validateRegistration.validateLastName();
    assertTrue(sut.getCheck());
    assertEquals( expected: "", sut.getMessage());
}

@Test
public void validateInvalidEmail() {
    String fakeEmail = "";
    validateRegistration.setEmailVal(fakeEmail);
    ValidateRegistration.ValidationMessage sut = validateRegistration.validateEmail();
    assertFalse(sut.getCheck());
    assertEquals( expected: "Enter a valid email", sut.getMessage());
```

```java
149 public void validateEmptyPassword() {
150     String fakePassword = "";
151     validateRegistration.setPasswordVal(fakePassword);
152     ValidateRegistration.ValidationMessage sut = validateRegistration.validatePassword();
153     assertFalse(sut.getCheck());
154     assertEquals( expected: "Field must not be empty", sut.getMessage());
155 }
156
157 @Test
158 public void validateInvalidPassword() {
159     String fakePassword = "test";
160     validateRegistration.setPasswordVal(fakePassword);
161     ValidateRegistration.ValidationMessage sut = validateRegistration.validatePassword();
162     assertFalse(sut.getCheck());
163     assertEquals( expected: "Password must have a minimum length of 8 characters",
164             sut.getMessage());
165 }
166
167 @Test
168 public void validateValidPassword() {
169     String fakePassword = "Password";
170     validateRegistration.setPasswordVal(fakePassword);
171     ValidateRegistration.ValidationMessage sut = validateRegistration.validatePassword();
172     assertTrue(sut.getCheck());
173     assertEquals( expected: "", sut.getMessage());
174 }
175
176 @Test
177 public void validateEmptyDob() {
178     String fakeDob = "";
179     validateRegistration.setDobVal(fakeDob);
180     ValidateRegistration.ValidationMessage sut = validateRegistration.validateDob();
181     assertFalse(sut.getCheck());
182     assertEquals( expected: "Enter a date", sut.getMessage());
183 }
184
185 @Test
186 public void validateValidDob() {
187     String fakeDob = "01/01/01";
188     validateRegistration.setDobVal(fakeDob);
189     ValidateRegistration.ValidationMessage sut = validateRegistration.validateDob();
190     assertTrue(sut.getCheck());
191     assertEquals( expected: "", sut.getMessage());
192 }
193
194 @Test
195 public void validateEmptyGender() {
196     String fakeGender = "";
197     validateRegistration.setGenderVal(fakeGender);
198     ValidateRegistration.ValidationMessage sut = validateRegistration.validateGender();
199     assertFalse(sut.getCheck());
200     assertEquals( expected: "Select a gender", sut.getMessage());
201 }
202
203 @Test
204 public void validateValidGender() {
205     String fakeGender = "Male";
206     validateRegistration.setGenderVal(fakeGender);
207     ValidateRegistration.ValidationMessage sut = validateRegistration.validateGender();
208     assertTrue(sut.getCheck());
```

3.2 Appendix C

```java
public class EmotionMarkerTest {

    private EmotionMarker emotionMarker;

    @Before
    public void setUp() { emotionMarker = new EmotionMarker(); }

    @After
    public void tearDown() { emotionMarker = null; }

    @Test
    public void emptyMarkerWithoutOptions() {
        MarkerOptions sut = new EmotionMarker().getMarker();
        assertNull(sut.getPosition());
    }

    @Test
    public void setLatLonPosition() {
        String fakeLat = "5.024012";
        String fakeLon = "-1.429342";
        emotionMarker.setLat(fakeLat);
        emotionMarker.setLon(fakeLon);
        MarkerOptions sut = emotionMarker.getMarker();
        assertEquals(fakeLat, String.valueOf(sut.getPosition().latitude));
        assertEquals(fakeLon, String.valueOf(sut.getPosition().longitude));
    }

    @Test
    public void emptyLatLonNullPosition() {
        String fakeLat = "";
        String fakeLon = "";
        emotionMarker.setLat(fakeLat);
        emotionMarker.setLon(fakeLon);
        MarkerOptions sut = emotionMarker.getMarker();
        assertNull(sut.getPosition());
    }

    @Test
    public void setEmotionTitle() {
        String fakeLat = "5.024012";
        String fakeLon = "-1.429342";
        String fakeEmotion = "TestEmotion";
        emotionMarker.setLat(fakeLat);
        emotionMarker.setLon(fakeLon);
        emotionMarker.setEmotion(fakeEmotion);
        MarkerOptions sut = emotionMarker.getMarker();
        assertEquals(fakeEmotion, sut.getTitle());
    }

    @Test
    public void emptyEmotionNullTitle() {
        String fakeLat = "5.024012";
        String fakeLon = "-1.429342";
        String fakeEmotion = "";
        emotionMarker.setLat(fakeLat);
        emotionMarker.setLon(fakeLon);
        emotionMarker.setEmotion(fakeEmotion);
        MarkerOptions sut = emotionMarker.getMarker();
        assertNull(sut.getTitle());
    }

    @Test
    public void emptyEmotionNullTitle() {
        String fakeLat = "5.024012";
        String fakeLon = "-1.429342";
        String fakeEmotion = "";
        emotionMarker.setLat(fakeLat);
        emotionMarker.setLon(fakeLon);
        emotionMarker.setEmotion(fakeEmotion);
        MarkerOptions sut = emotionMarker.getMarker();
        assertNull(sut.getTitle());
    }

    @Test
    public void setThoughtsSnippet() {
        String fakeLat = "5.024012";
        String fakeLon = "-1.429342";
        String fakeThoughts = "This is a test";
        emotionMarker.setLat(fakeLat);
        emotionMarker.setLon(fakeLon);
        emotionMarker.setThoughts(fakeThoughts);
        MarkerOptions sut = emotionMarker.getMarker();
        assertEquals(fakeThoughts, sut.getSnippet());
    }

    @Test
    public void emptyThoughtsNullSnippet() {
        String fakeLat = "5.024012";
        String fakeLon = "-1.429342";
        String fakeThoughts = "";
        emotionMarker.setLat(fakeLat);
        emotionMarker.setLon(fakeLon);
        emotionMarker.setThoughts(fakeThoughts);
        MarkerOptions sut = emotionMarker.getMarker();
        assertNull(sut.getSnippet());
    }

    @Test
    public void emptyLocationEntryNotMarked() {
        String fakeLat = "";
        String fakeLon = "";
        String fakeEmotion = "Test Emotion";
        String fakeThoughts = "Test Thoughts";
        emotionMarker.setLat(fakeLat);
        emotionMarker.setLon(fakeLon);
        emotionMarker.setEmotion(fakeEmotion);
        emotionMarker.setThoughts(fakeThoughts);
        MarkerOptions sut = emotionMarker.getMarker();
        assertNull(sut.getPosition());
        assertNull(sut.getTitle());
        assertNull(sut.getSnippet());
    }

}
```