

APPENDIX 1: Specimen Layout for Declaration/Statements page to be included in Taught Master's Degree Dissertations

| | |
|----------------------------|--|
| CANDIDATE'S ID NUMBER | C1676095 |
| CANDIDATE'S SURNAME | Please circle appropriate value Mr / Miss / Ms / Mrs / Rev / Dr / Other please specify .Zhuo..... |
| CANDIDATE'S FULL FORENAMES | Zhuo Cui |

DECLARATION

This work has not previously been accepted in substance for any degree and is not concurrently submitted in candidature for any degree.

Signed (candidate) Date ..7-9-2018.....

STATEMENT 1

This dissertation is being submitted in partial fulfillment of the requirements for the degree of ...MSc.....(insert MA, MSc, MBA, MScD, LLM etc, as appropriate)

Signed (candidate) Date 7-9-2018.....

STATEMENT 2

This dissertation is the result of my own independent work/investigation, except where otherwise stated.
Other sources are acknowledged by footnotes giving explicit references. A Bibliography is appended.

Signed (candidate) Date ..7-9-2018.....

STATEMENT 3 – TO BE COMPLETED WHERE THE SECOND COPY OF THE DISSERTATION IS SUBMITTED IN AN APPROVED ELECTRONIC FORMAT

I confirm that the electronic copy is identical to the bound copy of the dissertation

Signed (candidate) Date ..7-9-2018.....

STATEMENT 4

I hereby give consent for my dissertation, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed (candidate) Date ..7-9-2018.....

STATEMENT 5 - BAR ON ACCESS APPROVED

I hereby give consent for my dissertation, if accepted, to be available for photocopying and for inter-library loans **after expiry of a bar on access approved by the Graduate Development Committee.**

Signed (candidate) Date ..7-9-2018.....

Love Cardiff Android Application

Author

Zhuo Cui

Supervisor

Natasha Edwards

Moderator

James Osbourne

Abstract

The Love Cardiff android app is a mobile application, which aims to make a tied bond between the local in Cardiff and also an attraction for travellers to gain more information about Cardiff. The Love Cardiff App was conceptualised to create a unique platform to supplement all-rounded Cardiff information for users to directly acquire. The functionality of this app includes register and login, scan the latest specified Cardiff news, the map navigation supported by Google Map, the weather forecast, the post of unique stories by users themselves, the publishing of comments and the update of personal details.

Table of Contents

| | |
|--|-----------|
| ABSTRACT | 1 |
| INTRODUCTION..... | 4 |
| OUTLINE | 4 |
| AIMS AND OBJECTIVES..... | 5 |
| BACKGROUND..... | 5 |
| METHODOLOGY | 8 |
| SOFTWARE DEVELOPMENT METHOD | 8 |
| SOFTWARE DEVELOPMENT TOOLS..... | 8 |
| ANALYSIS | 10 |
| FUNCTIONAL REQUIREMENTS | 10 |
| NON-FUNCTIONAL REQUIREMENTS..... | 11 |
| DESIGN..... | 12 |
| USE CASES DESIGN | 12 |
| ACTIVITY LIFECYCLE DESIGN | 23 |
| DATABASE DESIGN | 24 |
| USER INTERFACE DESIGN..... | 25 |
| IMPLEMENTATION | 30 |
| CONFIGURATION | 30 |
| LOGIN/REGISTER | 32 |
| SCAN NEWS | 33 |
| SCAN STORIES..... | 33 |
| POST STORIES | 34 |
| MAKE COMMENTS..... | 34 |
| CHECK WEATHER..... | 35 |
| MAP NAVIGATION..... | 35 |
| MANAGE ACCOUNT | 35 |
| TESTING | 36 |
| SYSTEM COMPLETENESS..... | 36 |
| ACCEPTANCE CRITERIA RESULTS - FUNCTIONAL REQUIREMENTS..... | 36 |
| ACCEPTANCE CRITERIA RESULTS – NON-FUNCTIONAL REQUIREMENTS | 38 |
| FUTURE IMPROVEMENTS..... | 40 |
| FUNCTIONALITY..... | 40 |
| USER INTERFACE..... | 40 |
| ETHICAL ISSUE..... | 40 |
| SECURITY | 41 |
| CONCLUSION..... | 42 |
| REFLECTION ON LEARNING | 43 |
| APPENDIX..... | 44 |
| APPENDIX A - LOVE CARDIFF QUESTIONNAIRE PART 1 (FOR USER STUDY) | 44 |
| APPENDIX B - LOVE CARDIFF QUESTIONNAIRE PART 2 (FOR USABILITY STUDY) | 46 |
| APPENDIX C – THE RESULTS FOR LOVE CARDIFF QUESTIONNAIRE PART 1..... | 49 |

| | |
|--|-----------|
| APPENDIX D – THE RESULTS FOR LOVE CARDIFF QUESTIONNAIRE PART 2 | 53 |
| REFERENCES | 60 |

Introduction

Outline

Cardiff, the city of incomparable natural and cultural beauty, which had an estimated population of 350,000 in 2014 and attracts more than 18 million people each year (Worldpopulationreview.com, 2018). In light of the attraction of Cardiff and the convenience of mobile application, the push to develop a mobile app served for Cardiff skyrocketed. As for citizens, this app plays the role of a tied bond linked to citizens. As for tourists, this app gives users access to the specified local information. Understandably, the Love Cardiff app will have a lasting strategic impact on the city. What's more, the integration of the Love Cardiff will supplement the latest news for users and provide the unique opportunity for users to connect with others. On a broader level, introducing a user-friendly, highly accessible mobile app further cements Cardiff as a top-tier city through embracing technology to connect its users, accommodate users' needs and offering the ability to acquire specified news and posts in Cardiff through the click of a couple of buttons.

According to the report by the International Data Corporation that Android is the most popular mobile operating system, which constituted to 85% of the OS market in 2017 (IDC, 2018), the Love Cardiff app developed in an Android version has a large number of potential users. The main functionality of Love Cardiff app includes the local latest news display, the unique posts about current events, the weather forecast and the navigation. Besides, following the article named “Future Trends of Popularizing API Use”, the APIs not only merely cover every sphere, but also have high integration and usability (Medrano, 2018). Therefore, there are three APIs (the News API, the HeWeather API, and the Google Map API) requested in the Love Cardiff app to realise the corresponding functionalities.

With the more increasing number of app occurred in the software market during the postgraduate, my curiosity for how to develop an app becomes stronger. Fortunately, working on this app provides an excellent opportunity to transfer my interest into the app development practice on the basis of the knowledge and skills acquired during the postgraduate, such as the layout design in the CMT112 (Web Application Development), the database introduction in the CMT207 (Information Modelling & Database Systems), the project modelling in the CMT301(Business and IT management). Besides, as for the topic of the app, I would like to

choose a topic about the Cardiff where I have spent all my graduate year at here, and it also brought endless joy to me. In this way, I wish this app could have a positive influence on Cardiff as bringing convenience to the local and attracting more people to visit this beautiful city. Therefore, this is my original motivation for developing the Love Cardiff App.

Aims and Objectives

The primary aim of this project is to develop and successfully implement the Love Cardiff application on various Android devices to make all-rounded information about Cardiff easily accessible and permanently available and to stimulate users engagement. This aim is achieved by accomplishing a set of objectives, which are defined as:

1. Define the requirements for the Love Cardiff app by designing and evaluating questionnaires.
2. Determine what particular tools are needed and helpful for the application developing process.
3. Implement and test:
 - Document the issues encountered
 - Ensure this application will run on 90% of Android devices.
 - Ensure all functionalities could be implemented successfully on the Android devices.
4. Conduct a usability questionnaire to analyse and evaluate the user experience feedback from volunteers.
5. Visualize and document how the Love Cardiff app could be developed and updated further.

Background

As for the mobile application development background, from a recent survey in 2018, Android users could choose between 3.8 million apps (Statista, 2018), leading to the intensive competition which skyrocketed among mobile applications. What's more, according to the questionnaire for user study, around 65% participants prefer to use a smartphone to connect to the Internet as shown in figure 4 placed in the appendix and over than 80% participants have mobile phone support by the Android operating system. As a result, the quantity of the potential users for the Love Cardiff is considerable.

Due to the travel apps have the same aims as the Love Cardiff, which give users access to the specified local information and provide the unique opportunity for users to connect with others.

Concerning about the market share distribution, a researching for the ten best travel apps for Android has done primely. The short description for each one has listed below:

1. Airbnb (Rating in Google Play: 4.4 stars)

Airbnb allows people to rent out spare rooms to random travellers. It boasts 2.5 million homes across over 190 countries. The user also enables to add events to itinerary if they are interested in other stuff.

2. Booking.com (Rating in Google Play: 4.7 stars)

Its main feature is having over one million hotels, motels, and other accommodations at its disposal. Additionally, it has info on local attractions, landmarks, and even local Wi-Fi hotspots. There are also some convenience tools like offline maps, paperless booking, adding reservations to a personal event calendar, and more.

3. Gas App UK (Rating in Google Play: 4.4 stars)

Gas App UK is a road tripper's best friend. The primary functions of the app include providing useful tools such as a Pipe Sizing Calculator, Combustion Ratio Guidelines and Purge Calculators, instructions for boilers, gas fires and warm air units, documents such as warning notices, gas safety records, and job sheets.

4. Google Translate (Rating in Google Play: 4.4 stars)

The google translate is one of the most essential travel apps. It features the ability to translate virtually any language into almost any other language. Additionally, there is a camera feature that lets users point their phone at a sign or a menu to get an instant translation. Best of all, there's a real-time speech translator that can users talk to people.

5. Google Trips (Rating in Google Play: 4.1 stars)

The Google Trips is a trip planner app that lets users put all of their reservations, plans, and stuff to do all in one place. In addition to letting users store all of that stuff, the app has a list of nearby attractions that are both interesting and family friendly. Best of all, the app works offline so the user will have access to the info even in areas with no web access.

6. Guides by Lonely Planet (Rating in Google Play: 4.6 stars)

The Guides by Lonely Planet is a developer on Google Play. It features fun stuff to do in over 100 cities all over the world. It also contains audio phrasebooks, offline maps, and a bunch of other fun stuff.

7. KAYAK (Rating in Google Play: 4.5 stars)

KAYAK's app is an all-in-one app that lets users book hotels, flights, and rental cars. It will also let users track their pre-existing travel plans even if they did not book with

them. It rounds out the experience with things like price alerts, deal alerts, and price forecasts to help users find a better deal.

8. Priceline Hotel Deals, Rental Cars & Flights (Rating in Google Play: 4.5 stars)

This one of the most well-known travel apps features over 800,000 hotels along with thousands of rental car and flight options. It also features small quirks like Google Wallet support, future reservation tracking, and more.

9. Uber (Rating in Google Play: 4.2 stars)

Uber can help users find rides around town. They are available in plenty of countries and wait times should not be too bad most of the time.

10. Yelp (Rating in Google Play: 4.3 stars)

Yelp is a local guide to finding just the place to eat, shop, drink, relax, and play. While traveling, this app can help users learn more about the local area, find out where the good spots are, and help users avoid the places that aren't great.

According to the features including in the ten best travel apps for Android, a chart could be depicted as shown in figure 1 below.

| | Rent Room | Book Hotel | Book Flight | Rent Car | City Explosion | Fuel | Translate | Planner | Map | Transportation |
|-------------------------|---|------------|-------------|----------|----------------|------|-----------|---------|-----|----------------|
| Airbnb | ✓ | | | | ✓ | | | | ✓ | |
| Booking.com | | ✓ | | | ✓ | | | | ✓ | |
| Gas App | | | | | | ✓ | | | ✓ | |
| Google Translate | | | | | | | ✓ | | | |
| Google Trips | | | | | ✓ | | | ✓ | ✓ | |
| Guides by Lonely Planet | | | | | ✓ | | | | ✓ | |
| KAYAK | ✓ | ✓ | ✓ | | | | | ✓ | | |
| Priceline | ✓ | ✓ | ✓ | | | | | | | |
| Uber | | | | | | | | | ✓ | ✓ |
| Yelp | | | | | ✓ | | | | ✓ | |
| Love Cardiff | Local News, Weather Broadcast, Scan Stories | | | | ✓ | | | | ✓ | |

Figure 1

It is clear to see that the city explosion and Map are the top two frequent features developed in five and seven apps respectively, which are also considered in the Love Cardiff app. Additionally, unlike the other top apps, the Love Cardiff provides three unique features (local news, weather broadcast and stories), containing substantial potential users without any market share distribution.

In this case, the Love Cardiff app is a relatively new format app that focuses on providing several tailor-made services for a particular city. In conclusion, the Love Cardiff could be seen as a blueprint for this type of app that contributes to attracting more the local and travellers get involved.

Methodology

In this section, the software development method and tools were used to describe the methodology for the project.

Software Development Method

The Love Cardiff android app follows the Agile software development. After comparing to the waterfall model, the Agile software development was chosen as a model to implement the project. Unlike the waterfall model leading to designers may not be aware of future difficulties when designing a new software product or feature (McConnell, 2004), the Agile development methods divide the Love Cardiff project into several small subsections that minimize the amount of up-front planning and design. In this case, to ensure each progress was made and each activity was developed successfully, a new application was built to test each feature added over time. In the end, the project was assembled with all the activities.

In addition, the survey for the user study by using a questionnaire (placed in the appendix), and the self-learning of software development tools were undertaken as a foundation for the project.

Software Development Tools

Java Development Kit

The Java Development Kit (JDK) is a software development environment used for developing Java applications and applets (Techopedia.com, 2018). According to the Android Studio has to use the Java tool chain to build, in this case, the JDK need to install before start using the Android Studio. It has three essential steps to act as a part of the prepared work, which includes download, install and configure the latest JDK.

Android Studio

Android Studio is the indispensable tool for the Android app development, which defined as “the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development” (Android Developers, 2018). Due to the Android Studio provides smoother and richer GUI, easy access to Google services within Android Tools and open source development (Ducrohet, 2018), this IDE was chosen over Eclipse.

Android Studio is a free and open source available on Mac, leading a cost-saving budget to this project. Also, it is a useful development tool as it provides me several default-layout followed by design guidelines. What's more, it helps me to increase the productivity by its automate

work. Lastly, there were considerable tutorial videos and learning platform for Android Studio available on the websites, which give me help for the self-study.

Android Virtual Device Manager

Android Virtual Device Manager enables me to create AVDs(Android Virtual Devices) which I can run my developed app as a testing trial in the virtual device on my computer and check the layout and usability without installing the app on an actual physical device.

News API

News API is a simple and easy-to-use API that returns JSON metadata for headlines and articles live all over the web right now (Newsapi.org, 2018). News API enables me to use News article metadata discovered just by creating a request with an API key, which is a real benefit for developers to implement the News functionality in an effective way by following the “Get Started” instruction. The command of the News API can be found on the official website at <https://newsapi.org>.

HeWeather API

HeWeather API servers for around 50,000 cities all over the world and ensures the 99.99% usability without unexpected interruption. Additionally, HeWeather API also provides the official weather broadcast in the numerous companies, such as the Amap.com, the Neusoft software company, the Gree Electric household appliances company. JSON is the data type retrieved from HeWeather API, which benefits me to parse it in Android and display the content in the weather broadcast activity. The command of the HeWeather API can be found on the official website at <http://www.heweather.com>.

Google Map SDK for Android

According to the survey of “Best sat-nav apps 2018”, the Google Map comes as standard with various kind of operating system, due to it provides a clear and intuitive map layout with the excellent use of colours to distinguish large roads, parks and smaller streets (Saarinen, 2018). In this case, the Google Map SDK is requested for fulfilling the navigation function. This not only enables me to add maps based on the Google Map data, but also provides me access to use the API calls to add markers, polygons, and overlays to a basic map. The command of the Google Map SDK for Android can be found on the official website at <https://developers.google.com/maps/documentation/android-sdk/intro>.

Bmob Cloud Database

Regarding the database of this app, the cloud database is considered beforehand by taking its advantages of the stored database can be accessed from anywhere via Internet connection, and

it only costs about 3 cents per gigabyte to store data internally. Based on what I have learned during my undergraduate, the Bmob cloud database is the common cloud database system. Besides, the Bmob Cloud Database official website provides particular instructions and video tutorial for developers to practice in a program. In this case, the Bmob Cloud Database is required to restore user information, post information and comments. The more details around Bmob Cloud Database could be accessed at <https://www.bmob.cn>.

Analysis

This section is used to analyse the project requirements through the functional and non-functional aspects separately. For the detailed functional requirements, in the next chapter, those should be implemented.

Functional Requirements

Functional requirements are defined as “In software engineering and systems engineering, a functional requirement defines a function of a system or its component, where a function described as a specification of behaviour between outputs and inputs”(Fulton, Vandermolen, 2017). To bring the Love Cardiff app to fruition, a set of functional requirements were developed and represented as the primary services that will be on offer through The Love Cardiff. The functional requirements could be defined as:

Must-Have

1. The application will allow the user to register or sign in.

Acceptance criteria:

- The sign-in or register displayed as an entrance to the home screen.

2. The application will allow the user to scan the Cardiff local news.

Acceptance criteria:

- The local news displayed on the home screen.
- The user could refresh the page by using a finger to pull up or down.
- The user could view a specify news by pressing on it.

3. The application will allow the user to utilize the map for navigation, which supported by the GPS technology.

Acceptance criteria:

- The map could be zoom in or out by pinching with two fingers.

4. The application will allow the user to post stories.

Acceptance criteria:

- The user could input the title and content to complete their posts.
5. The application will allow the user to view and make a comment to the posts from other users.

Acceptance criteria:

- The posts could be displayed in a page straightforwardly.
 - The user could refresh the post page by using a finger to pull up or down.
 - The user could view detailed posts by tapping it.
 - The user could comment on the others' post.
6. The application will allow the user to check the weather forecast.

Acceptance criteria:

- The weather information should include the instant weather observations and the weather forecasts for the next two days.

Should

7. The user could reset their password as if he forgot it.
8. The user could check whether the personal data is accurate to abide by the Data Protection Act (Legislation.gov.uk, 2018) and change their information details after logging into the app.
9. The user could start the navigation by input the name of the building.
10. The user should be given the options of whether he allows being retrieved his location or not.
11. The user should enable to delete his posts.
12. The app should allow the user to log out easily.
13. The app should support all Android software versions.

Will Not

14. The application contains no advertisements.

Non-functional Requirements

A collection of non-functional requirements was also developed to aid the complement of functional requirements. According to ISO 25010, a quality in use model composed of five characteristics (some of which are further subdivided into sub-characteristics) that relate to the outcome of interaction when a product is used in a particular context of use (ISO/IEC, 2011).

The non-functional requirements include:

1. Compatibility

- The Love Cardiff will be device-agnostic. The aim is to be fully compatible with the Android operating system on different devices.

2. Usability

- The app will be intuitive, interactive and aesthetically pleasing with simple navigation.
- The user interface should design a consistent theme and utilize appropriate Android features.

3. Reliability

- The Love Cardiff will be available 24/7
- The application will be programmed in such way that the code will be fault-tolerant for unexpected input and will be able to run smoothly without crashing or resulting in bad user experience.
- The user will have a personal folder of his profile uploaded to the server in case of any unauthorized data modification with the ability to recover data.

4. Security

- Authenticity should confirm by two measures (multiple password input, or email confirmed).
- A protocol will be developed to protect user privacy, which enables the user to decide if they willing to be retrieved from the geo-location or not.

5. Maintainability

- The Love Cardiff will be maintained by myself and can be outsourced on a contract basis as well.
- Defects and standards of product quality will continuously be monitored.

Design

Use Cases Design

The UML case diagram was depicted below acting as the blueprint for the software development of Love Cardiff app, shown in Figure 2.

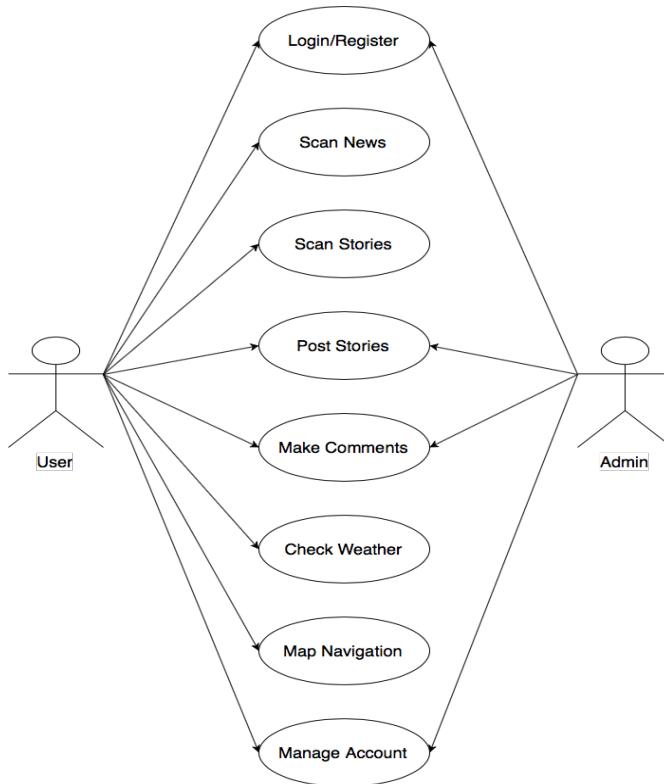


Figure 2

Use Case 1 – Login/Register

User

The user could log in or register an account by inputting credentials. The application could check the input whether matching or not in the cloud database. Then, in the case of login, giving user a hint “Login Successfully” before switching the main activity or “Wrong Password” to let user try once again, in the case of register, giving the user a hint “Submit Successfully” as the prompt for creating account successfully or “Submit Unsuccessfully” as the prompt for creating account unsuccessfully.

Basic Flow

1. Implement the Love Cardiff Android app.

The user installs and implements the app on a mobile device.

2. Input credentials or register a new account

The application shows the login frame that within two textboxes of username and password and the button for presenting the register frame. The user inputs credentials or clicks the “Register” button to implement the registration process.

3. Click “Log in” or “Submit” button

The user presses the “Log in” or “Submit” button to finish the login or registration process.

Alternate Flow

1. Cancel Login/Register

The user cancels the login/register process by pressing the back button which is available in each activity and takes the user back to the previous activity. The login or registration will be discarded and not saved.

2. Username or password not input

If the user forgets or misses out a field to input, then the app will inform the user by showing a hint “Null Input!”.

3. Username or password is incorrect

If the user inputs unmatched username and password, then the app will present a hint “Incorrect Password!” for the user to input once again.

4. Close application

The user closes the application by shutting down the application from the app management control in the Android system.

Admin

The admin can get the authority and log in the backstage of the database system to scan the username and password details.

Pre-condition

The admin logs in the backstage of the database system.

Basic Flow

The admin scans the user’s account information by clicking on the “User” table.

Alternate Flow

1. The admin’s username or password not input

If the admin forgets or misses out a field to input, then the database system will hint the admin by showing a “Null Input!” toast.

2. The admin’s username or password is incorrect

If the admin inputs unmatched username and password, then the database system will present a hint “Incorrect Password!” for the admin to input once again.

3. Cancel scan users’ account

The admin cancels scan users’ account by shutting down the database system.

Use Case 2 – Scan News

The user could scan the list of Cardiff local news presenting as the main activity on the screen and click on a news item to get detailed information by jumping to the official news page.

Pre-condition

The user logs in the app successfully.

Basic Flow

1. Scroll the news list

The user scrolls up and down the news list to find which they would like to view.

2. Click on the news title

The user clicks on the new title and the official news page containing detailed information will appear on the screen.

3. Press “Back” button

The user jumps back to the news list by pressing on the “Back” button.

Alternate Flow

1. Close application

The user closes the application by shutting down the application from the app management control in the Android system.

2. Click the “Stories” button in a sorter

The user could scan stories posted by other users by clicking on the “Stories” button in the sorter which including news and stories options.

3. Select an option from the menu bar

The user selects one of the options on the menu bar to jump to other activities. For example, the “Scan News” option refers to Use Case 2, the “Post Stories” option refers to Use Case 4, the “Make Comments” option refers to Use Case 5, the “Check Weather” option refers to Use Case 6, the “Map Navigation” option refers to Use Case 7, the “Manage Account” option refers to Use Case 8.

Use Case 3 – Scan Stories

The user could scan the list of stories’ title posted by other users and click on each item to jump to the story activity. In the story activity, the user able to acquire title, content and author information about the story posted by other.

Pre-condition

The user is in the News activity.

Basic Flow

1. Click the “Stories” button in a sorter

The user could scan stories by clicking on the “Stories” button in the sorter which including news and stories options in the main activity.

2. Scroll the stories list

The user scrolls up and down the story list to scan the title of each story.

3. Click on the stories’ title

The user clicks on the stories’ title and the story activity containing detailed information will appear on the screen.

4. Press the “Back” button

The user jumps back to the story list by pressing on the “Back” button.

Alternate Flow

1. Close application

The user closes the application by shutting down the application from the app management control in the Android system.

2. Click the “News” button in a sorter

The user scans Cardiff latest news by clicking on the “News” button in the sorter which including news and stories options.

3. Select another option from the menu bar

As mentioned above in the Use Case 2 - Alternate Flow - 3, the user selects one of the options on the menu bar to jump to other activities.

Use Case 4 – Post Stories

User

This case allows users to edit and publish stories by filling in several input textboxes. The application presents an initial hint to users for the input.

Pre-condition

The user logs in the app successfully.

Basic Flow

1. Click the menu bar icon

The user switches on the menu bar by clicking on the menu bar in the main activity.

2. Select the “Post Stories” option

The user clicks on the “Post Stories” option to jump into the referring activity.

3. Fill in the textbox

The user inputs title and content.

4. Click “Submit” button

The user clicks the submit button to post the story.

Alternate Flow

1. Title or content not input

If the user forgets or misses out a field to input, then the app will inform the user by showing a hint “Null Input!”.

2. Close application

The user closes the application by shutting down the application from the app management control in the Android system.

3. Select another option from the menu bar

As mentioned above in the Use Case 2 - Alternate Flow - 3, the user selects one of the options on the menu bar to jump to other activities.

4. Press “Back” Button

The process of posting cancelled and the user backs to the previous activity.

Admin

The admin can log in the backstage of the database system to manage the stories data, such as insert a new story data, update users' stories information and delete stories.

Pre-condition

The admin logs in the backstage of the database system.

Basic Flow

1. Click the “UserThings” table

The admin scans the user's stories information by clicking on the “UserThings” table.

2. Manage the “UserThings” table

The admin inserts a new story by inputting username, title and content, or update users' stories information by editing the corresponding textboxes, or delete stories by clicking on the “Delete” button.

1. Click the “Save” button

The admin clicks on the “Save” button to save the modifications in the “UserThings” table.

Alternate Flow

1. The admin's username or password not input

If the admin forgets or misses out a field to input, then the database system will hint the admin by showing a “Null Input!” toast.

2. The admin’s username or password is incorrect

If the admin inputs unmatched username and password, then the database system will present a hint “Incorrect Password!” for the admin to input once again.

3. Cancel manage users’ stories

The admin cancels manage users’ stories by shutting down the database system.

4. Manage data in another table

The admin selects another table to manage the corresponding data.

Use Case 5 – Make Comments

User

The user can comment on the story which posted by other users. Followed by submitting successfully, the comment including the content, username and publish time will be presented below the detailed stories.

Pre-condition

The user is at the detailed story activity.

Basic Flow

1. Tap the comment textbox

The user taps the textbox below the detailed stories and then input the comment.

2. Click the “Submit” button

The user clicks on the “Submit” button, and this brings up a pop-up text “Submit Successfully !” that as a hint for the user.

Alternate Flow

1. Submit unsuccessfully

As a result of the comment submitted unsuccessfully, the user will receive a toast “Submitted unsuccessfully, please try it again !”. Then, the user could implement two steps following the basic flow once again.

2. Close application

The user closes the application by shutting down the application from the app management control in the Android system.

3. Select another option from the menu bar

As mentioned above in the Use Case 2 - Alternate Flow - 3, the user selects one of the options on the menu bar to jump to other activities.

4. Press the “Back” Button

The user can press the back button to back to the previous activity.

Admin

The admin can log in the backstage of database system to manage the comment data, such as insert a new comment, update users' comment and delete comments.

Pre-condition

The admin logs in the backstage of database system.

Basic Flow

1. Click the “UserComment” table

The admin scans the user's comments information by clicking on the “UserComment” table.

2. Manage the “UserComment” table

The admin inserts a new comment by inputting username and comment, or update users' comments information by editing the corresponding textboxes, or delete comments by clicking on the “Delete” button.

3. Click the “Save” button

The admin clicks on the “Save” button to save the modifications in the “UserComment” table.

Alternate Flow

1. The admin's username or password not input

If the admin forgets or misses out a field to input, then the database system will hint the admin by showing a “Null Input!” toast.

2. The admin's username or password is incorrect

If the admin inputs unmatched username and password, then the database system will present a hint “Incorrect Password!” for the admin to input once again.

3. Cancel manage users' comments

The admin cancels manage users' comments by shutting down the database system.

4. Manage data in another table

The admin selects another table to manage the corresponding data.

Use Case 6 – Check Weather

The user could view the broadcast for the Cardiff weather.

Pre-condition

The user logs in the app successfully.

Basic Flow

1. Click the menu bar icon

The user switches on the menu bar by clicking on the menu bar in the main activity.

2. Select the “Check Weather” option

The user clicks on the “Check Weather” option to view the weather broadcast.

Alternate Flow

1. Close application

The user closes the application by shutting down the application from the app management control in the Android system.

2. Select another option from the menu bar

As mentioned above in the Use Case 2 - Alternate Flow - 3, the user selects one of the options on the menu bar to jump to other activities.

3. Press “Back” Button

The user can press the back button to back to the previous activity.

Use Case 7 – Map Navigation

The user could view their current location and activate the route function by searching a destination and creating a marker on the map. After creating a marker, the user could click on the google map icon placed at the bottom of the map to open the google map application spontaneously, which provides the route function based on the Google Maps data with the Maps SDK for Android which automatically handles access to Google Maps servers.

Pre-condition

The user logs in the app successfully.

Basic Flow

1. Click the menu bar icon

The user switches on the menu bar by clicking on the menu bar in the main activity.

2. Select the “Map Navigation” option

The user selects the “Map Navigation” option to view the map.

3. Zoom the map

The user checks the local location by pinching to zoom in and out on Google Maps, or double tap on the screen, then drag their finger up or down.

4. Click on the “Searching” image button

The user clicks on the “Searching ” icon and inputs the name of the destination to create a marker on the map.

5. Click on the “Route” image button

The user clicks on the “Route” image button and jumps to the Google Map app that provides the navigation.

Alternate Flow

1. Close application

The user closes the application by shutting down the application from the app management control in the Android system.

2. Disable location service on the mobile device

The user will receive a toast “Please open the GPS service first !”. Then, the user could utilize the map navigation function.

3. Without the Google Map Application on the mobile device

The user will receive a toast “Please download the Google Map !”.

4. Select an option from the menu bar

As mentioned above in the Use Case 2 - Alternate Flow - 3, the user selects one of the options on the menu bar to jump to other activities.

5. Press the “Back” Button

The user can press the back button to back to the previous activity.

Use Case 8 – Manage Account

User

In this case, the user could change their password and log out the account.

Pre-condition

The user logs in the app successfully.

Basic Flow

1. Click the menu bar icon

The user switches on the menu bar by clicking on the menu bar in the main activity.

2. Select the “Manage Account” option

The user selects the “Manage Account” option to implement the function in features.

3. Fill in the textbox

The user inputs the new password in the textbox.

4. Click the “Save” button

The user clicks on the “Save” button to change the password of their account.

5. Click the “Log Out” button

The user clicks on the “Log Out” button to log out the application and logs in the app using the new password.

Alternate Flow

1. Close application

The user closes the application by shutting down the application from the app management control in the Android system.

2. Change password unsuccessfully

The user will receive a toast “Change password unsuccessfully !” as the hint of the change password process unimplemented.

3. Select an option from the menu bar

As mentioned above in the Use Case 2 - Alternate Flow - 3, the user selects one of the options on the menu bar to jump to other activities.

4. Press “Back” Button

The user can press the back button to back to the previous activity.

Admin

The admin can get the authority and log in the backstage of the database system to manage data, such as insert a new user data, update users’ personal information and delete the account.

Pre-condition

The admin logs in the backstage of the database system.

Basic Flow

1. Click the “User” table

The admin scans the user’s account information by clicking on the “User” table.

2. Manage the “User” table

The admin inserts a new account by inputting username and password, or update users’ personal information by editing the corresponding textboxes, or delete account by clicking on the “Delete” button.

3. Click the “Save” button

The admin clicks on the “Save” button to save the modifications in the “User” table.

Alternate Flow

1. The admin's username or password not input

If the admin forgets or misses out a field to input, then the database system will hint the admin by showing a “Null Input!” toast.

2. The admin's username or password is incorrect

If the admin inputs unmatched username and password, then the database system will present a hint “Incorrect Password!” for the admin to input once again.

3. Cancel manage users' account

The admin cancels manage users' account by shutting down the database system.

4. Manage data in another table

The admin selects another table to manage the corresponding data.

Activity Lifecycle Design

The life cycle of activity is managed by the classes, which defines the following events:

- OnCreate() – Method called in the created state, activities load and displayed the user interface elements and allocate resources such as the display resource, and internal storage resource, and other resources required for this application.
- OnStart() – Method called in the started state to get ready for the user implementation.
- OnResume() – Method called in the resumed state when all actions are displayed. For example, according to the user input to redraw the user interface elements.
- OnPause() – Method called in the paused state, the activity is being paused, and the Love Cardiff application is put to the background, while another application is brought to the front and started.
- OnStop() – Method called in the stopped state when the application is put to sleep and no longer visible to the user.
- OnRestart() – Method called when the activity has been stopped and is restarting again.
- OnDestroy() – Method called in the destroyed state, which means all processes are destroyed and all device resources released.

The activity lifecycle describes the application runtime from launch to the end of the application life. The figure 3 shows the Love Cardiff application activity lifecycle.

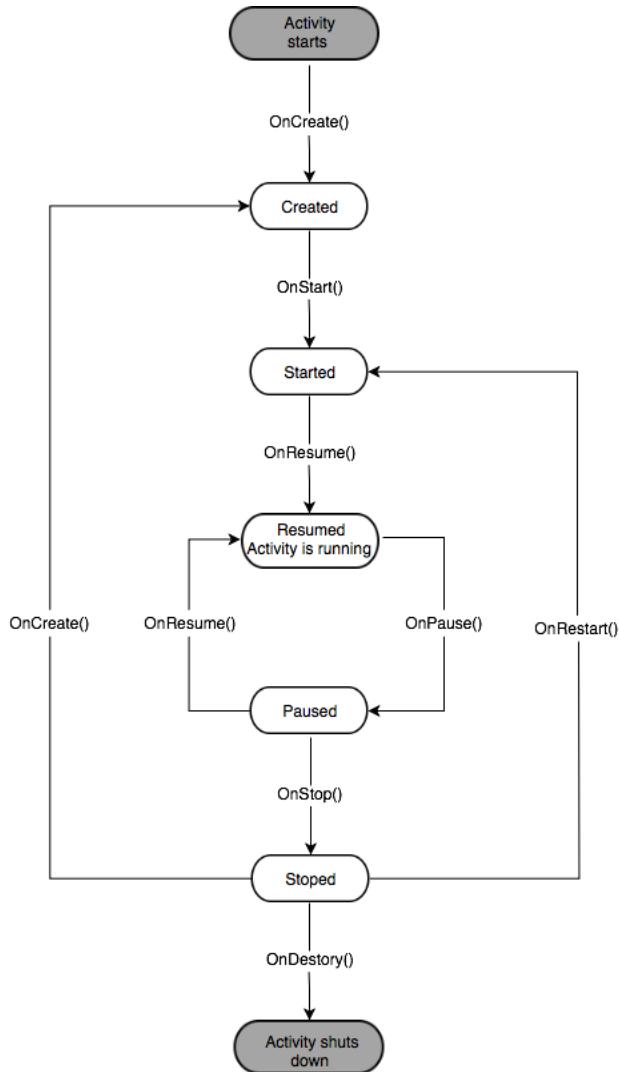


Figure 3

Database Design

The database design is considered in the Love Cardiff application to store the data corresponding with the use cases. As mentioned above in the Software Development Tools chapter, taking advantage of its high usability and convenient to recall, Bmob Cloud Database is utilized. Additionally, three database tables are created to store different related data, accessed by various functionalities within the application. These are “User”, “UserThings” and “UserComment”. The JSON data type is used in the data interaction between the Bmob Cloud Database and the Love Cardiff application.

The “User” table stores the username and password that required by the login/register feature as shown in table 1 below.

| Data Name | Data Type | Description |
|-----------|-----------|----------------------------------|
| ID | String | The unique string for each user |
| Username | String | The name of users' account. |
| Password | String | The password setting up by users |
| createdAt | Date | The date of the account register |

Table 1

The “UserThings” table stores the stories posted by users that required by the post stories feature as shown in table 2 below.

| Data Name | Data Type | Description |
|-----------|-----------|---|
| ID | String | The unique string for each story |
| Username | String | The author name of the particular story |
| Title | String | The title of the story |
| Content | String | The content of the story |
| createdAt | Date | The date of the story posted |

Table 2

The “UserComment” table stores the comments published by users that required by the make comments feature as shown in table 3 below.

| Data Name | Data Type | Description |
|-----------|-----------|---|
| ID | String | The unique string for each comment |
| Username | String | The author name of the particular comment |
| Comment | String | The content of the comment |
| createdAt | Date | The date of the comment made |

Table 3

User Interface Design

The user interface plays a vital role in software development, which aims to bring simple and intuitive layout for users to understand how to utilize the app without any confusion. As for the Love Cardiff app, by following the Android design standards and as possible as using standard Android UI components such as textbox and buttons associated with the features to bring familiarity to users. The user interface for each use case has screenshots and introduction in this section.

Use Case 1 – Login/Register

A picture is set as the background and the “CardView” layout is used to display the login activity. Additionally, a floating action button is used to provide flexible movement between

the “login” and “register” features. The floating action button has a universally standard symbol (+), which is derived from the Android design library and recommended in the material design guides. This design enables the user to easily identify the function of this button without confusing. The user interface design for the login and register activities is shown below in figure 4 and figure 5 respectively.

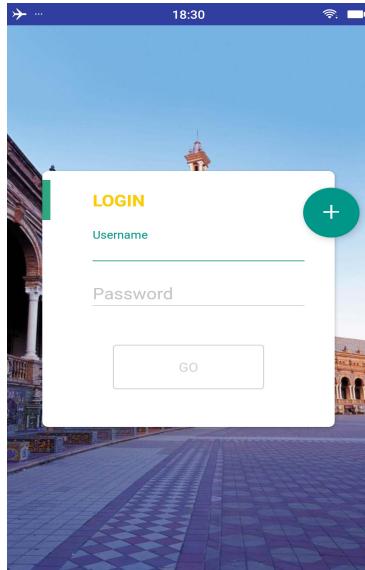


Figure 4

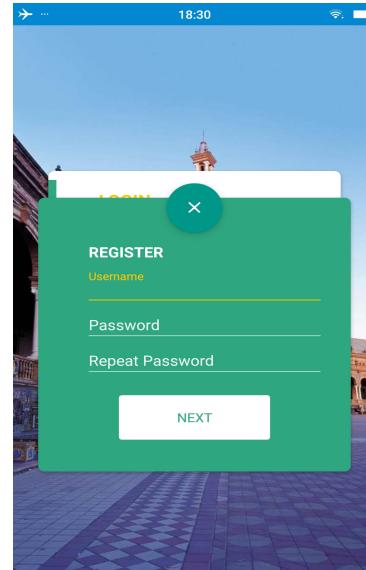


Figure 5

Use Case 2 – Scan News

The Cardiff News is displayed in the listview to the user which contains the textview and the imageview. This consistent layout for each news item highlights these are separate but equal items, which makes it easy for users to browse. In addition, to separate the “News” page and the “stories” page, the TabLayout is used which contains viewPager and fragments. The user interface design for the scan news activity is shown below in figure 6.

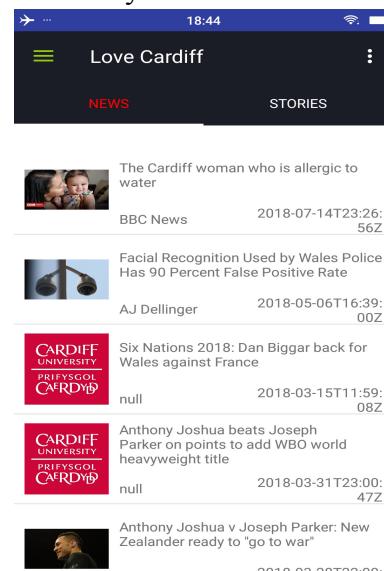


Figure 6

Use Case 3 – Scan Stories

The stories posted by users are displayed in the listview with the linear layout. To highlight the title of each story for users to scan easily, the font size is enlarged. As mentioned in the use case 2, the TabLayout is used to separate the “News” page and the “stories” page. The user interface design for the scan stories activity is shown below in figure 7.

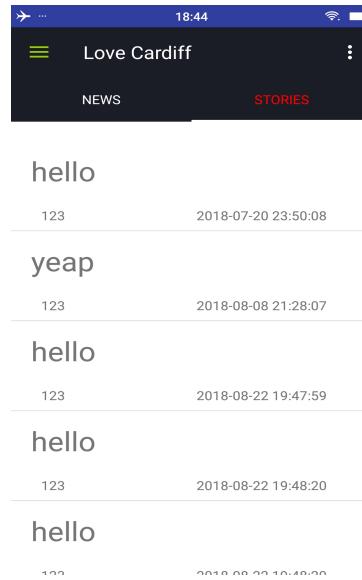


Figure 7

Use Case 4 – Post Stories

In this activity, the Material Design EditText is requested from the GitHub (link: <https://github.com/rengwuxian/MaterialEditText>) and placed with the linear layout. At the bottom of the activity, a “submit” button is set to support the post stories functionality. The user interface design for the post stories activity is shown below in figure 8.

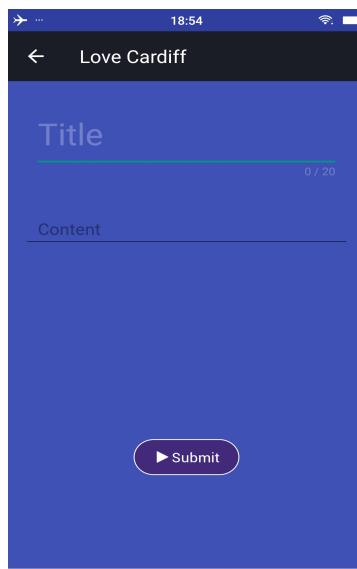


Figure 8

Use Case 5 – Make Comments

As for the make comments activity, a picture is set as the background and three textview are recalled to display the title, content and time respectively, which followed by a scrollview to list the previous comments from other users. An edit text box and a Bootstrap button (link: <https://getbootstrap.com/docs/4.0/components/buttons/>) are placed at the end to support the corresponding transaction for users. The user interface design for the make comments activity is shown below in figure 9.



Figure 9

Use Case 6 – Check Weather

In this activity, a picture is set as the background and several textview are placed with the linear layout to display the data which retrieved from the API endpoint. The user interface design for the check weather activity is shown below in figure 10.



Figure 10

Use Case 7 – Map Navigation

The Google Maps Activity by default is requested to implement the map navigation functionality and display the current location on the map. Additionally, a search button is placed at the bottom, which initially designed to allow the user to search a place. The user interface design for the map navigation activity is shown below in figure 11.



Figure 11

Use Case 8 – Manage Account

In the manage account activity, the “CardView” is recalled to display the TextInputLayout for users to update their account information. What’s more, the two buttons are designed to support the “submit” and “sign out” features. The user interface design for the manage account activity is shown below in figure 12.

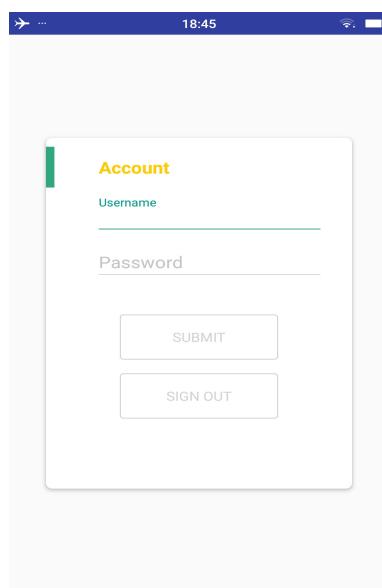


Figure 12

Implementation

The following section details the configuration of the development environment and the specified implementation of all functionalities on features. The methodology which provided in the methodology section is utilized to support the implementation.

Configuration

The Android Studio 3.1.4 was utilised as the integrated development environment (IDE) to develop this app, as it mentioned in the previous Methodology-Software Development Tools part. It is straightforward to create the project after I launched the Android Studio. To begin, click Start a new Android Studio project and name the app as Love Cardiff. Then, select the device and SDK version. According to the instruction “By targeting API 15 and later, your app will run on approximately 100% of devices”, I selected Android SDK 4.0.3 for Phone and Tablet, as shown in Figure 13.

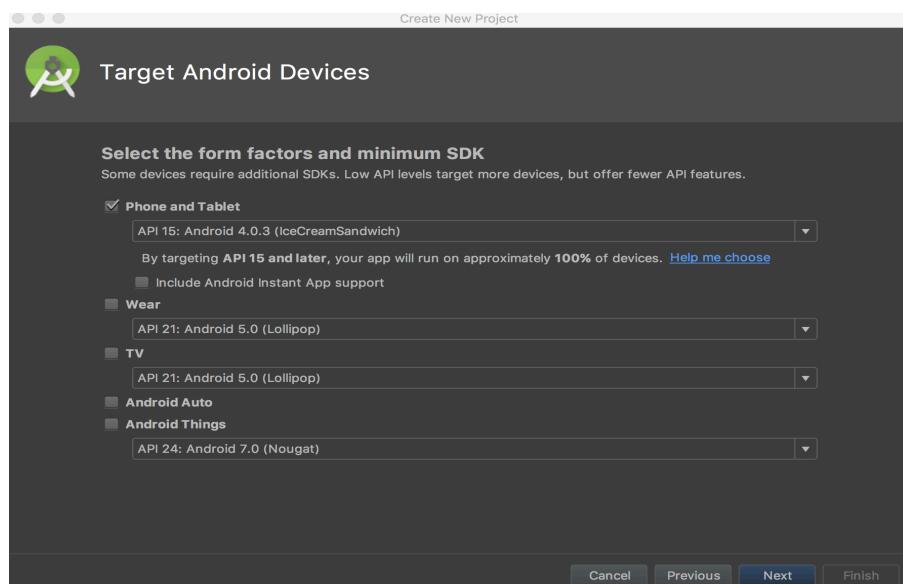


Figure 13

After selecting the SDK version, click Next to initiate the first activity by selecting the Empty Activity template. Until now, the creation of the Love Cardiff project finished followed by the Android Studio auto-generates files and folders for the initial project structure and auto-configures the application build environment. By default, as same as all the Android applications, the Love Cardiff project has a hierarchy structure, as shown in Figure 14. This project includes manifests folder, Java source code folder (contains database folder, login folder, and utils folder) and res folder (includes images and the layout XML file).

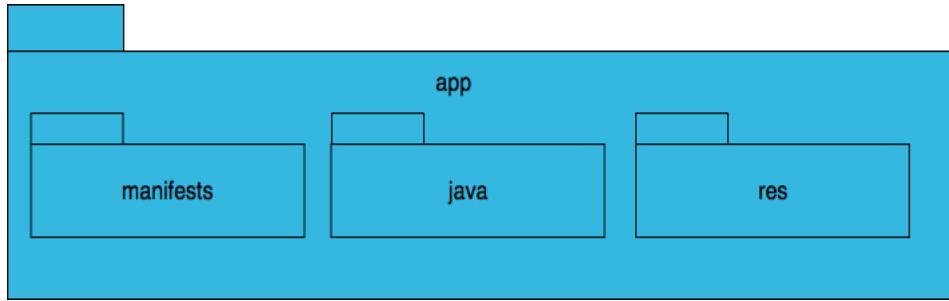


Figure 14

The list of permissions that the requests from the Love Cardiff app along with the reason for using them have been provided below:

- **Permission Code:** android.permission.INTERNET
Reason: This permission allows the app to open network sockets. It is requested to enable the app to post and get information from the server when a user logs in, scans the local news, checks the weather, uses the Google map, scans stories posted by other users, posts a story, makes comments, changes the password.
- **Permission Code:** android.permission.ACCESS_NETWORK_STATE
Reason: This permission allows the app to access information about networks. The app requests this permission to evaluate whether the mobile device is connected to the internet, which is required when the implementation needs data communicating with the database or APIs. Besides, this permission provides feedback to users if the internet connection is unavailable.
- **Permission Code:** android.permission.ACCESS_WIFI_STATE
Reason: The app requests this permission to access information about Wi-Fi networks. This permission is requested by the app to evaluate whether the Wi-Fi connection is working. What's more, this permission provides feedback to users if the Wi-Fi connection is unavailable.
- **Permission Code:** android.permission.ACCESS_FINE_LOCATION
Reason: This permission allows the app to determine as precise a location as possible from the available location providers, including the Global Positioning System (GPS) as well as Wi-Fi and mobile cell data. When the user utilizes the map function along with associated with the Google Map data, this permission will provide the current fine location.
- **Permission Code:** android.permission.WAKE_LOCK

Reason: This permission allows using PowerManager WakeLocks to keep processor from sleeping or screen from dimming.

- **Permission Code:** android.permission.WRITE_EXTERNAL_STORAGE

Reason: This permission allows the app to read from external storage. This permission is requested when the user needs to store the corresponding file locally before the user wants to upload them.

- **Permission Code:** android.permission.READ_PHONE_STATE

Reason: This permission allows read-only access to phone state, including the phone number of the device, current cellular network information, the status of any ongoing calls, and a list of any PhoneAccounts registered on the device.

The description of every permission mentioned above can be found on the official Android website at <https://developer.android.com/reference/android/Manifest.permission>.

Login/Register

In the login activity, in order to store user account information, the application needs to link with the Bmob cloud database using the API key. The main code is shown as follow:

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.login_layout);  
    isPermissionAllGranted(new String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE, Manifest.permission.READ_PHONE_STATE}, requestCode: 3);  
    Bmob.initialize(context: this, appkey: "56404633619c00d9184f451b80010918");  
    initView();  
    setListener();  
}
```

Then, according to the input from users, the username and the password need to be compared whether matching with each other. After this, a corresponding toast will show up as a hint for the user to implement the next step. The main code is shown as follow:

```
private void setListener() {  
    bt_go.setOnClickListener((view) -> {  
        String name_str = et_username.getText().toString();  
        String pw_str = et_password.getText().toString();  
        String repw_str = et_repeatpassword.getText().toString();  
        if (name_str.equals("") || pw_str.equals("") || repw_str.equals(""))) {  
            Toast.makeText(context: register.this, text: "Null input", Toast.LENGTH_SHORT).show();  
            return;  
        } else if (!pw_str.equals(repw_str)) {  
            Toast.makeText(context: register.this, text: "Incorrect password", Toast.LENGTH_SHORT).show();  
            return;  
        }  
        user userObj = new user();  
        userObj.setUsername(name_str);  
        userObj.setPassword(pw_str);  
        userObj.save((s, e) -> {  
            if (e == null) {  
                Toast.makeText(context: register.this, text: "Submit Success", Toast.LENGTH_SHORT).show();  
                animateRevealClose();  
            } else {  
                Toast.makeText(context: register.this, text: "Submit Failure", Toast.LENGTH_SHORT).show();  
            }  
        });  
    });  
}
```

In the login activity, the get() and set() method are recalled to retrieve the data from users' input and store them in the database, which enables the implementation of the register functionality. The main code is shown as follow:

```
public class user extends BmobObject {
    private String username;
    private String password;
    public String getName() { return username; }
    public void setUsername(String username) { this.username = username; }

    public String getPassword() { return password; }
    public void setPassword(String password) { this.password = password; }
}
```

Scan News

The scan news activity calls the openConnection() method and casting the result to HttpURLConnection. After this, the data need to be retrieved and parsed. Lastly, get the string type data and add them to the list. The main code is shown as follow:

```
URL url = new URL("https://newsapi.org/v2/everything?q=cardiff&apiKey=ce81253c97924a7e8f99539c64cc130b");
HttpURLConnection openConnection = (HttpURLConnection) url
    .openConnection();
openConnection.setConnectTimeout(40000);
openConnection.setReadTimeout(40000);
int responseCode = openConnection.getResponseCode();
if (responseCode == 200) {
    InputStream inputStream = openConnection
        .getInputStream();
    String parseSteam = StreamUtils.parseSteam(inputStream);
    JSONObject jsonObject = new JSONObject(parseSteam);
    JSONArray jsonArray = jsonObject.getJSONArray("articles");
    for (int i=0;i<jsonArray.length();i++){
        JSONObject jsonObject1 = jsonArray.getJSONObject(i);
        String author = jsonObject1.getString("author");
        String title = jsonObject1.getString("title");
        String description = jsonObject1.getString("description");
        String urlToImage = jsonObject1.getString("urlToImage");
        String new_url = jsonObject1.getString("url");
        String publishedAt = jsonObject1.getString("publishedAt");
        list_author.add(author);
        list_title.add(title);
        list_description.add(description);
        list_urlToImage.add(urlToImage);
        list_url.add(new_url);
        list_publishedAt.add(publishedAt);
    }
}
```

Scan Stories

In the Scan stories activity, a java class document named "UserThings" is created and linked with "UserThings" table which is mentioned above in the database design. In addition, the get() method is used to retrieve the corresponding data which should be displayed for users to browse.

The main code is shown as follow:

```
public class UserThings extends BmobObject {
    private String username;
    private String thingId;
    private String thingTitle;
    private String thingContent;
    public String getName() { return username; }
    public void setUsername(String username) { this.username = username; }
    public String getThingId() { return thingId; }
    public void setThingId(String thingId) { this.thingId = thingId; }
    public String getThingTitle() { return thingTitle; }
    public void setThingTitle(String thingTitle) { this.thingTitle = thingTitle; }
    public String getThingContent() { return thingContent; }
    public void setThingContent(String thingContent) { this.thingContent = thingContent; }
}
```

Post Stories

In the post stories, setOnClickListener() is used to implement the post functionality, which contains the get() and set() method to retrieve the data from users' input and store them in the corresponding data table. Then, a toast will show up as a hint to notify users whether they have posted the story successfully or not. The main code is shown as follow:

```
private void setListener() {
    submit.setOnClickListener((view) -> {
        final String th_title = title.getText().toString();
        final String th_content = content.getText().toString();
        SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
        Date date = new Date(System.currentTimeMillis());
        final String time = simpleDateFormat.format(date);
        Explode explode = new Explode();
        explode.setDuration(500);
        explode.setExitTransition(explode);
        explode.setEnterTransition(explode);
        if (th_title.equals("") || th_content.equals("")) {
            Toast.makeText(context: ThingActivity.this, text: "null content", Toast.LENGTH_SHORT).show();
            return;
        }
        UserThings thingObj = new UserThings();
        thingObj.setThingTitle(th_title);
        thingObj.setThingContent(th_content);
        thingObj.setUsername(user_name);
        thingObj.setThingId(time);
        thingObj.save((s, e) -> {
            if (e == null) {
                Toast.makeText(context: ThingActivity.this, text: "Submit Success", Toast.LENGTH_SHORT).show();
                Intent intent = new Intent(packageContext: ThingActivity.this, MainActivity.class);
                startActivity(intent);
            } else {
                Toast.makeText(context: ThingActivity.this, text: "Submit Failure", Toast.LENGTH_SHORT).show();
            }
        });
    });
}
```

Make Comments

In the make comments activity, a java class document named “UserComment” is created and linked with “UserComments” table which is mentioned above in the database design. The main code is shown as follow:

```
public class UserComment extends BmobObject {
    private String username;
    private String thingId;
    private String comment;
    public String getName() { return username; }
    public void setUsername(String username) { this.username = username; }
    public String getThingId() { return thingId; }
    public void setThingId(String thingId) { this.thingId = thingId; }
    public String getComment() { return comment; }
    public void setComment(String comment) { this.comment = comment; }
}
```

Then, calling the setOnClickListener() method on the “submit” button and a toast will show up to notify users whether they have commented on the story successfully or not. The main code is shown as follow:

```
UserComment commentObj = new UserComment();
String author_name = "";
author_name = (String) new SharedPreferencesHelper(context: ThingView.this).getData(key: "author", defValue: "姓名");
commentObj.setComment(user_comment);
commentObj.setUsername(author_name);
commentObj.setThingId(things_id);
commentObj.save((s, e) -> {
    if (e == null) {
        setInfo();
        comment.setText(null);
        Toast.makeText(context: ThingView.this, text: "Submit Success", Toast.LENGTH_SHORT).show();
    } else {
        Toast.makeText(context: ThingView.this, text: "Submit Failure", Toast.LENGTH_SHORT).show();
    }
});
break;
```

Check Weather

The weather information is retrieved from the server by opening the HttpURLConnection. Then, the JSON data type needs to be parsed into the string type. Lastly, get the string type data and add them to the list, which displayed on the screen and allows users to browse. The main code is shown as follow:

```
private void get_information(final String location){
    city_name = location;
    new Thread(){
        public void run() {
            try {
                URL url = new URL( spec: "https://free-api.heweather.com/s6/weather/forecast?key=6e2cadce7c004a3e88a9c986f40937e1&lang=en&location="+location);
                HttpURLConnection openConnection = (HttpURLConnection) url
                    .openConnection();
                openConnection.setConnectTimeout(5000);
                openConnection.setReadTimeout(5000);
                int responseCode = openConnection.getResponseCode();
                if (responseCode == 200) {
                    InputStream inputStream = openConnection
                        .getInputStream();
                    String parseStream = StreamUtils.parseStream(inputStream);
                    JSONObject jsonObject = new JSONObject(parseStream);
                    JSONArray jsonArray = jsonObject.getJSONArray( name: "HeWeather6");
                    JSONObject allJSONObject = jsonArray.getJSONObject( index: 0 );
                    String daily_forecast = allJSONObject.getString( name: "daily_forecast");
                    JSONArray jsonArray_d = new JSONArray(daily_forecast);
                    for (int i = 0; i < jsonArray_d.length(); i++) {
                        JSONObject jsonObject1 = jsonArray_d.getJSONObject(i);
                        String date = jsonObject1.getString( name: "date");
                        String cond_txt_d = jsonObject1.getString( name: "cond_txt_d");
                        String tmp_max = jsonObject1.getString( name: "tmp_max");
                        list_date.add(date);
                        list_cond_txt_d.add(cond_txt_d);
                        list_tmp_max.add(tmp_max);
                    }
                }
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    };
}
```

Map navigation

In the map navigation activity, the user can obtain the current location from the Google Map. The navigation functionality is supported by the Google Map Android SDK, and the GoogleApiClient.Builder() method is called to fulfill the implementation. The main code is shown as follow:

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_maps);
    if (mGoogleApiClient == null) {
        mGoogleApiClient = new GoogleApiClient.Builder( context: this)
            .addConnectionCallbacks(this)
            .addOnConnectionFailedListener(this)
            .addApi(LocationServices.API)
            .build();
    }
    // Obtain the SupportMapFragment and get notified when the map is ready to be used.
    SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
        .findFragmentById(R.id.map);
    mapFragment.getMapAsync( onMapReadyCallback: this);
    createLocationRequest();
}
FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
fab.setOnClickListener((view) -> { loadPlacePicker(); });
```

Manage Account

The manage account activity allows the user to change password. The getText() is called to retrieve the user's input. After this, according to the input from users, the username needs to be compared and then update the password. Lastly, a corresponding toast will show up as a hint for the user. The main code is shown as follow:

```

private void setListener() {
    account_go.setOnClickListener((view) -> {
        final String name_str = account_username.getText().toString();
        final String pw_str = account_password.getText().toString();
        final String[] str = {};
        Explode explode = new Explode();
        explode.setDuration(500);
        getWindow().setExitTransition(explode);
        getWindow().setEnterTransition(explode);
        BmobQuery<User> query = new BmobQuery<>();
        query.addWhereEqualTo("username", name_str);
        query.findObjects((list, e) -> {
            if (e == null) {
                for (User user : list) {
                    str[0] = user.getObjectId();
                }
                Toast.makeText(context, Account.this, text: "1"+str[0], Toast.LENGTH_SHORT).show();
            }
            final User userObj = new User();
            userObj.setPassword(pw_str);
            userObj.update(str[0], (e) -> {
                Toast.makeText(context, Account.this, text: "2"+str[0], Toast.LENGTH_SHORT).show();
                if(e==null){
                    Toast.makeText(context, Account.this, text: "Success", Toast.LENGTH_SHORT).show();
                }else{
                    Toast.makeText(context, Account.this, text: "fail", Toast.LENGTH_SHORT).show();
                }
            });
        });
    });
}

```

Testing

System Completeness

This part is an overall review of the completeness of the Love Cardiff app in regards to the functional and non-functional requirements met, and the objectives and functionalities provided in the chapter Analysis.

Acceptance Criteria Results - Functional Requirements

Must-Have

- The application allows the user to register or sign in.

Acceptance criteria met: True

Confirmed during the development testing and the usability testing through the question 4 and 5 in the questionnaire for usability study (the result shown in appendix D).

- The application allows the user to scan the Cardiff local news.

Acceptance criteria met: True

Confirmed during the development testing and the usability testing through the question 6 in the questionnaire for usability study (the result shown in appendix D).

- The application allows the user to utilize the map for navigation, which supported by the GPS technology.

Acceptance criteria met: False

Implemented unsuccessfully

- The application allows the user to post stories.

Acceptance criteria met: True

Confirmed during the development testing and the usability testing through the question 11 in the questionnaire for usability study (the result shown in appendix D).

- The application allows the user to view and make a comment to the posts from other users.

Acceptance criteria met: True

Confirmed during the development testing and the usability testing through the question 12 in the questionnaire for usability study (the result shown in appendix D).

- The application allows the user to check the weather forecast.

Acceptance criteria met: True

Confirmed during the development testing and the usability testing through the question 9 in the questionnaire for usability study (the result shown in appendix D).

Should

- The user could reset their password as if he forgot it.

Acceptance criteria met: False

Not implemented in time

- The user could check whether the personal data is accurate to abide by the Data Protection Act (Legislation.gov.uk, 2018) and change their information details after logging into the app.

Acceptance criteria met: False

Not implemented in time

- The user could find start the navigation by input the name of the building.

Acceptance criteria met: False

Implemented unsuccessfully

- The user should be given the options of whether he allows being retrieved his location or not.

Acceptance criteria met: True

Confirmed during the development testing.

- The user should enable to delete his posts.

Acceptance criteria met: False

Not implemented in time

- The app should allow the user to log out.

Acceptance criteria met: True

Confirmed during the development testing.

Will Not

- The application contains no advertisements.

Acceptance criteria met: True

Confirmed during the development testing.

Acceptance Criteria Results – Non-Functional Requirements

1. Compatibility

- The Love Cardiff will be device-agnostic. The aim is to be fully compatible with the Android operating system on different devices.

Acceptance criteria met: True

Confirmed during the development testing.

2. Usability

- The app will be intuitive, interactive and aesthetically pleasing with simple navigation.

Acceptance criteria met: True

Confirmed during the usability testing through the question 15 in the questionnaire for usability study (the result shown in appendix D).

- The user interface should design a consistent theme and utilize appropriate Android features.

Acceptance criteria met: True

Confirmed during the development testing and the usability testing through the question 17 in the questionnaire for usability study (the result shown in appendix D).

3. Reliability

- The Love Cardiff will be available 24/7

Acceptance criteria met: True

Confirmed during the development testing.

- The application will be programmed in such way that the code will be fault-tolerant for unexpected input and will be able to run smoothly without crashing or resulting in bad user experience.

Acceptance criteria met: True

Confirmed during the development testing.

- The user will have a personal folder of his profile uploaded to the server in case of any unauthorized data modification with the ability to recover data.

Acceptance criteria met: False

Not implemented in time

4. Security

- Authenticity should confirm by two measures (multiple password input, or email confirmed).

Acceptance criteria met: False

Not implemented in time

- A protocol will be developed to protect user privacy, which enables the user to decide if they willing to be retrieved from the geo-location or not.

Acceptance criteria met: True

Confirmed during the development testing.

5. Maintainability

- The Love Cardiff will be maintained by myself and can be outsourced on a contract basis as well.

Acceptance criteria met: False

- Defects and standards of product quality will continuously be monitored.

Acceptance criteria met: False

Future Improvements

The following section details the potential improvements of the Love Cardiff app, which is considered to improve the user experience in the future. These potential improvements are described from the functionality, user interface, ethical issue and security perspectives respectively.

Functionality

As for the “Post Stories” functional requirement, it enables the user to submit text as the content of stories to post based on the current system. Testers like this feature and are willing to post stories for other users to scan, while the majority of them felt this functionality could go further. The implementation of this feature would need to provide more options of content type for users to choose, such as photos and videos.

As for the “Make Comments” functional requirement, it allows the user to comment on stories which posted by other users, but the user cannot delete posts currently. In this case, “delete posts” functionality should be developed in the future to allow the user could delete what he posted previously.

A portion of the “Map Navigation” application is implemented unsuccessfully. Currently, the app can display the user’s location supported by the Google Map. However, the searching function cannot work smoothly and it does not allow the user to search a place and jump to the Google Map application which provides the navigation service. Due to the time limitation, this implementation problem has not found a good solution by far. In the future work, this bug needs to be fixed up to provide the complete functionality for users.

User Interface

Considering the user interface personalization, a customized theme should be achieved in the future, which aims to presents each user with the unique interface and meets the needs of specific users, rather than provides a one style fit all’s experience for all users. At present, the Love Cardiff app only provides an unchangeable theme for users.

The orientation problem should be solved for the Love Cardiff app in the future. In order to provide the optimized oriental layout to the user, more XML files should be developed and then the application could choose an appropriate layout for the activity based on the context.

Ethical Issue

The ethical issue occurs in the “Make Comments” functionality. Due to handling this problem, the Love Cardiff app should be added the capacity to block offensive comments with

a system that uses DeepText AI classification system to detect and eliminate any comments which it deems to violate of the app's community.

Security

The database is used to store the data from the Love Cardiff application. To secure the data and prevent data theft and leakage, the data encryption should be used to secure data within the application sandbox against malware and other forms of criminal access. In addition, the individual data elements should be encrypted and controlled to protect user privacy.

Conclusion

The primary aim of the project has been met in the Love Cardiff android. Through analysing the survey for the market share distribution and the questionnaire for user study, the clear functional and non-functional requirements is created. During the implementation, a consistent environment and user interface has been maintained by adding appropriated Android design materials and the application can answer the most of the functionalities (exclude the map navigation feature). The questionnaire for usability study was developed to evaluate the user experience. In this way, due to the fulfillment mentioned above, I suppose that this project is a success.

Reflection on Learning

Developing the Love Cardiff Android app is provided me with an excellent opportunity to enhance my skills and lead me to be more competitive in my future software development career. The reflection on learning is around three aspects: software development skills and self-learning ability, organization and management.

As for a beginner of the Android Studio, I learned the software development tools beforehand. Fortunately, there are innumerable opening sources on the websites which allow me to use. Also, the <https://www.imooc.com> and <https://www.youtube.com> are the top 2 useful websites for me to find some tutorials, which enhanced my programming skills. What's more, my self-learning ability has an enormous development as well. During the development of this project, when some bugs appeared, I documented the problems and searched them online to find the solution from other's experience. In this way, the most of the problems have resolved. However, when I encountered the unsuccessfully searching functionality in the map navigation activity, I tried to fix the problem, due to the time limitation and my development skill limitation, I couldn't resolve the problem in an appropriate way. After communicating with my supervisor, the searching functionality would be implemented as a part of the future work. In this way, during the period of developing the Love Cardiff Android app, my software development skills and self-learning ability has significantly improved, which has a positive influence on the future job.

Furthermore, although this is the first time that I manage a project on my own, I am satisfied with the positive result achieved in this project. I feel that my organization and management skills have improved drastically since starting the project. Honestly, before working on the Love Cardiff project, I prefer to work on a project without any real plan or follow the plan which made by the project leader. When I worked on this project, I managed myself to make a plan based on the agile software development approach beforehand. In this way, the initial plan I created had clear aims, objectives and work plan that enabled me to make regular and steady progress in the project. As a result, my organization and management ability has a rapid improvement.

APPENDIX

Appendix A - Love Cardiff Questionnaire Part 1 (For User Study)

1. What is your gender?

- Female
- Male
- Prefer not to say

2. What is your age?

- 18 to 24
- 25 to 35
- 36 or older
- Prefer not to say

3. What is the highest level of school you have completed of the highest degree you have received?

- Less than high school degree
- High school degree
- Bachelor degree
- Master degree
- Doctor degree or above
- Prefer not to say

4. Which of the following devices do you most prefer to use to connect to the internet?

- Computer tablet
- Desktop computer
- Laptop computer
- Smart phone

5. What type of phone do you have?

- Android

- iPhone
- Blackberry
- Others

6. How could you get the latest news in your local area ?

- Watching TV
- Browsing website
- Using news apps
- Others

7. If you explore a city through an app, what kinds of information do you hope to gain?

- The latest local news
- Map
- Weather
- Others

8. Would you like to post your story for other users to read and comment through an app?

- Yes
- No

9. What is your attitude towards a local app which obtains specified information about a certain city?

- Extremely interested
- Very interested
- Somewhat interested
- Not so interested
- Not interested at all

10. When using an application, which of the following factors most important to you ?

- Layout
- Error-free functionality
- Usability
- Trustworthiness

Appendix B - Love Cardiff Questionnaire Part 2 (For Usability Study)

1. What is your gender?

- Female
- Male
- Prefer not to say

2. What is your age?

- 18 to 24
- 25 to 35
- 36 or older
- Prefer not to say

3. What is the highest level of school you have completed of the highest degree you have received?

- Less than high school degree
- High school degree
- Bachelor degree
- Master degree
- Doctor degree or above
- Prefer not to say

4. Were you able to register an account for the app?

- Yes
- No

5. Were you able to log into the account you just registered with?

- Yes
- No

6. Were you able to add and save profile information about yourself?
 - Yes
 - No
7. Were you able to change the password?
 - Yes
 - No
8. Were you able to scan the news?
 - Yes
 - No
9. Were you able to check the weather forecast?
 - Yes
 - No
10. Were you able to utilize the navigation map?
 - Yes
 - No
11. Were you able to post your story?
 - Yes
 - No
12. Were you able to make a comment below others' stories?
 - Yes
 - No
13. I think that I would like to use the app frequently.
 - Strongly agree
 - Agree
 - Neither agree nor disagree
 - Disagree
 - Strongly disagree

14. I think the app is easy to use.

- Strongly agree
- Agree
- Neither agree nor disagree
- Disagree
- Strongly disagree

15. I think the layout of the app is intuitive.

- Strongly agree
- Agree
- Neither agree nor disagree
- Disagree
- Strongly disagree

16. I found the various functions in this app were well integrated.

- Strongly agree
- Agree
- Neither agree nor disagree
- Disagree
- Strongly disagree

17. I thought there was too much inconsistency in this app.

- Strongly agree
- Agree
- Neither agree nor disagree
- Disagree
- Strongly disagree

18. I was able to easily find my way around the app to get the information I was looking for.

- Strongly agree

- Agree
- Neither agree nor disagree
- Disagree
- Strongly disagree

19. I am satisfied with the app.

- Strongly agree
- Agree
- Neither agree nor disagree
- Disagree
- Strongly disagree

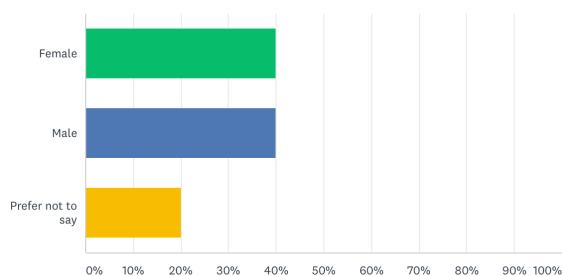
20. I would like to recommend the app to my friends, colleagues or family members who want to get some information about Cardiff.

- Strongly agree
- Agree
- Neither agree nor disagree
- Disagree
- Strongly disagree

Appendix C – The Results for Love Cardiff Questionnaire Part 1

1. What is your gender?

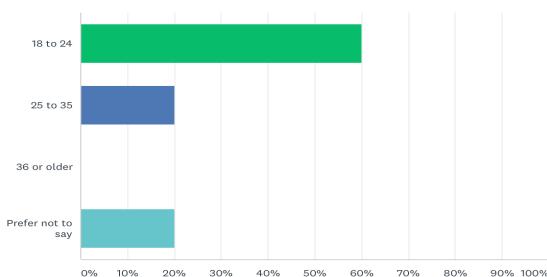
Answered: 5 Skipped: 0



| ANSWER CHOICES | RESPONSES |
|----------------------|-----------|
| ▼ Female | 40.00% |
| ▼ Male | 40.00% |
| ▼ Prefer not to say | 20.00% |
| Total Respondents: 5 | |

2. What is your age?

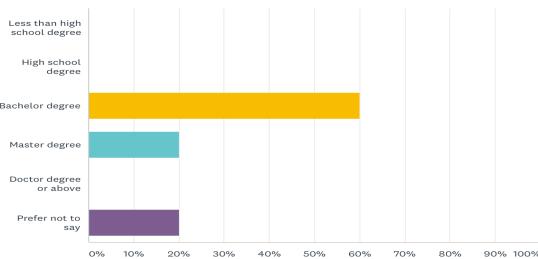
Answered: 5 Skipped: 0



| ANSWER CHOICES | RESPONSES |
|-----------------------------|-----------|
| ▼ 18 to 24 | 60.00% |
| ▼ 25 to 35 | 20.00% |
| ▼ 36 or older | 0.00% |
| ▼ Prefer not to say | 20.00% |
| Total Respondents: 5 | |

3. What is the highest level of school you have completed of the highest degree you have received?

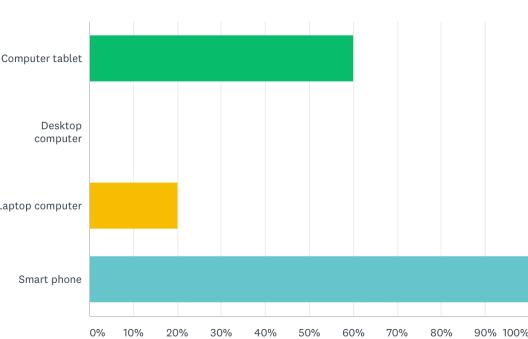
Answered: 5 Skipped: 0



| ANSWER CHOICES | RESPONSES |
|--------------------------------|-----------|
| ▼ Less than high school degree | 0.00% |
| ▼ High school degree | 0.00% |
| ▼ Bachelor degree | 60.00% |
| ▼ Master degree | 20.00% |
| ▼ Doctor degree or above | 0.00% |
| ▼ Prefer not to say | 20.00% |
| Total Respondents: 5 | |

4. Which of the following devices do you most prefer to use to connect to the internet?

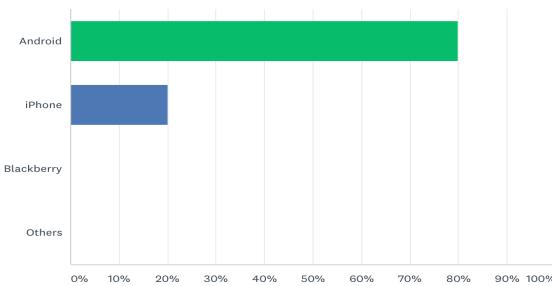
Answered: 5 Skipped: 0



| ANSWER CHOICES | RESPONSES |
|-----------------------------|-----------|
| ▼ Computer tablet | 60.00% |
| ▼ Desktop computer | 0.00% |
| ▼ Laptop computer | 20.00% |
| ▼ Smart phone | 100.00% |
| Total Respondents: 5 | |

5. What type of phone do you have?

Answered: 5 Skipped: 0

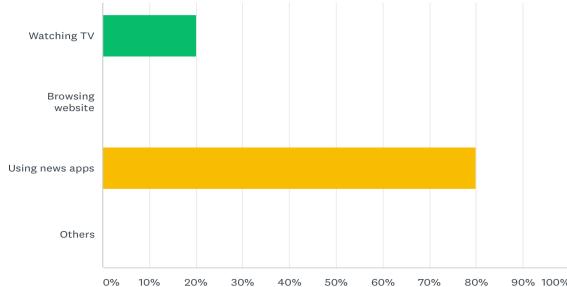


| ANSWER CHOICES | RESPONSES |
|-----------------------------|-----------|
| ▼ Android | 80.00% 4 |
| ▼ iPhone | 20.00% 1 |
| ▼ Blackberry | 0.00% 0 |
| ▼ Others | 0.00% 0 |
| Total Respondents: 5 | |

How could you get the latest news in your local area?

6.

Answered: 5 Skipped: 0

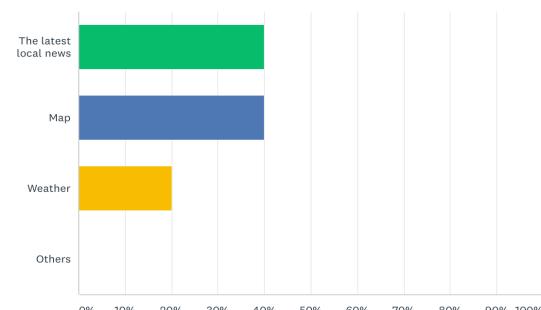


| ANSWER CHOICES | RESPONSES |
|--------------------|-----------|
| ▼ Watching TV | 20.00% 1 |
| ▼ Browsing website | 0.00% 0 |
| ▼ Using news apps | 80.00% 4 |
| ▼ Others | 0.00% 0 |
| TOTAL | 5 |

7.

If you explore a city through an app, what kinds of information do you hope to gain?

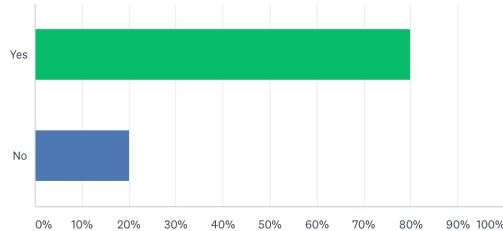
Answered: 5 Skipped: 0



| ANSWER CHOICES | RESPONSES |
|-------------------------|-----------|
| ▼ The latest local news | 40.00% 2 |
| ▼ Map | 40.00% 2 |
| ▼ Weather | 20.00% 1 |
| ▼ Others | 0.00% 0 |
| TOTAL | 5 |

8. Would you like to post your story for other users to read and comment through an app?

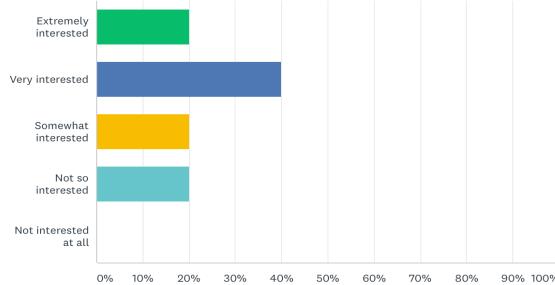
Answered: 5 Skipped: 0



| ANSWER CHOICES | RESPONSES |
|----------------------|-----------|
| ▼ Yes | 80.00% |
| ▼ No | 20.00% |
| Total Respondents: 5 | |

9. What is your attitude towards a local app which obtains specified information about a certain city?

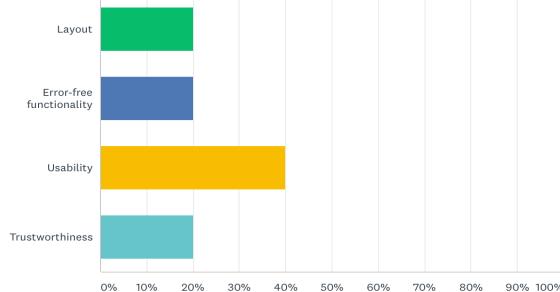
Answered: 5 Skipped: 0



| ANSWER CHOICES | RESPONSES |
|-------------------------|-----------|
| ▼ Extremely interested | 20.00% |
| ▼ Very interested | 40.00% |
| ▼ Somewhat interested | 20.00% |
| ▼ Not so interested | 20.00% |
| ▼ Not interested at all | 0.00% |
| Total Respondents: 5 | |

10. When using an application, which of the following factors most important to you?

Answered: 5 Skipped: 0

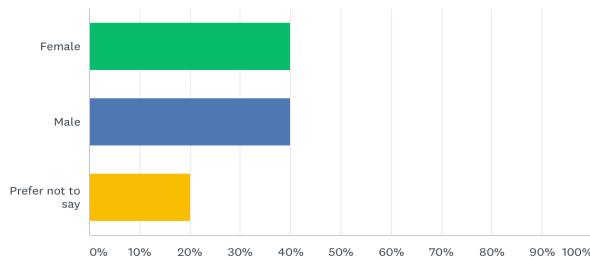


| ANSWER CHOICES | RESPONSES |
|----------------------------|-----------|
| ▼ Layout | 20.00% |
| ▼ Error-free functionality | 20.00% |
| ▼ Usability | 40.00% |
| ▼ Trustworthiness | 20.00% |
| TOTAL | 5 |

Appendix D – The Results for Love Cardiff Questionnaire Part 2

1. What is your gender?

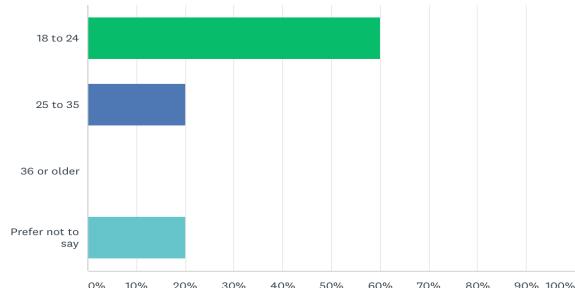
Answered: 5 Skipped: 0



| ANSWER CHOICES | RESPONSES |
|-----------------------------|-----------|
| ▼ Female | 40.00% |
| ▼ Male | 40.00% |
| ▼ Prefer not to say | 20.00% |
| Total Respondents: 5 | |

2. What is your age?

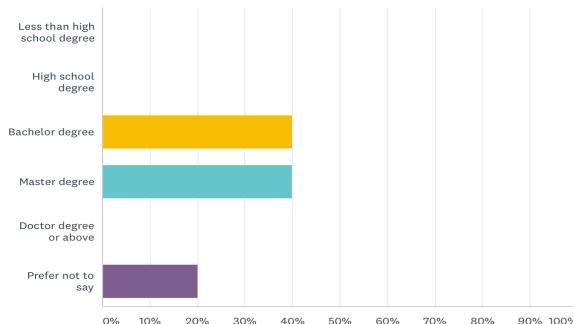
Answered: 5 Skipped: 0



| ANSWER CHOICES | RESPONSES |
|-----------------------------|-----------|
| ▼ 18 to 24 | 60.00% |
| ▼ 25 to 35 | 20.00% |
| ▼ 36 or older | 0.00% |
| ▼ Prefer not to say | 20.00% |
| Total Respondents: 5 | |

3. What is the highest level of school you have completed or the highest degree you have received?

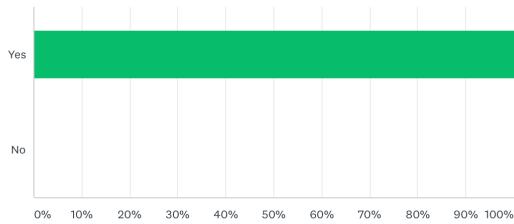
Answered: 5 Skipped: 0



| ANSWER CHOICES | RESPONSES |
|--------------------------------|-----------|
| ▼ Less than high school degree | 0.00% |
| ▼ High school degree | 0.00% |
| ▼ Bachelor degree | 40.00% |
| ▼ Master degree | 40.00% |
| ▼ Doctor degree or above | 0.00% |
| ▼ Prefer not to say | 20.00% |
| Total Respondents: 5 | |

4. Were you able to register an account for the app?

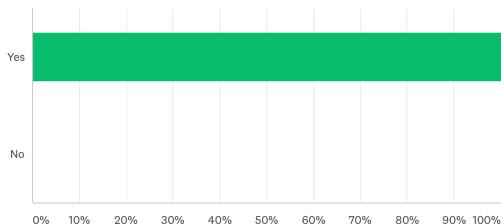
Answered: 5 Skipped: 0



| ANSWER CHOICES | RESPONSES |
|----------------------|-----------|
| ▼ Yes | 100.00% |
| ▼ No | 0.00% |
| Total Respondents: 5 | |

5. Were you able to log into the account you just registered with?

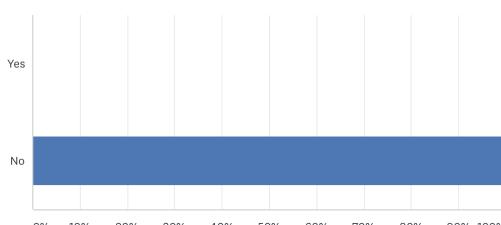
Answered: 5 Skipped: 0



| ANSWER CHOICES | RESPONSES |
|----------------------|-----------|
| ▼ Yes | 100.00% |
| ▼ No | 0.00% |
| Total Respondents: 5 | |

6. Were you able to add and save profile information about yourself?

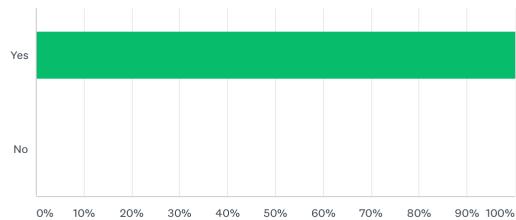
Answered: 5 Skipped: 0



| ANSWER CHOICES | RESPONSES |
|----------------------|-----------|
| ▼ Yes | 0.00% |
| ▼ No | 100.00% |
| Total Respondents: 5 | |

7. Were you able to change the password?

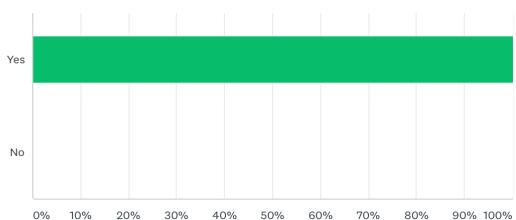
Answered: 5 Skipped: 0



| ANSWER CHOICES | RESPONSES |
|----------------------|-----------|
| ▼ Yes | 100.00% |
| ▼ No | 0.00% |
| Total Respondents: 5 | |

8. Were you able to scan the news?

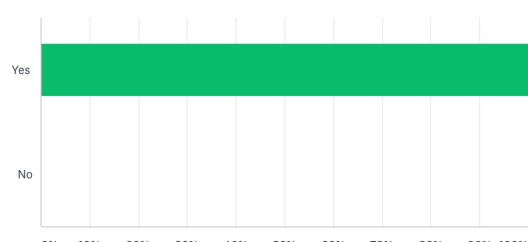
Answered: 5 Skipped: 0



| ANSWER CHOICES | RESPONSES |
|----------------------|-----------|
| ▼ Yes | 100.00% |
| ▼ No | 0.00% |
| Total Respondents: 5 | |

9. Were you able to check the weather forecast?

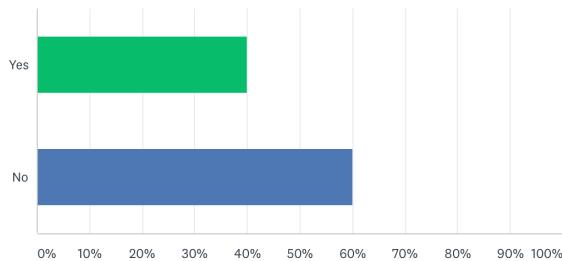
Answered: 5 Skipped: 0



| ANSWER CHOICES | RESPONSES |
|----------------------|-----------|
| ▼ Yes | 100.00% |
| ▼ No | 0.00% |
| Total Respondents: 5 | |

10. Were you able to utilize the navigation map?

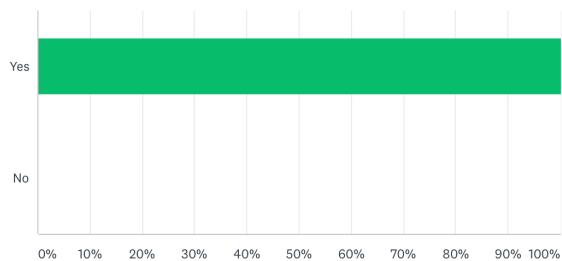
Answered: 5 Skipped: 0



| ANSWER CHOICES | RESPONSES |
|----------------------|-----------|
| ▼ Yes | 40.00% |
| ▼ No | 60.00% |
| Total Respondents: 5 | |

11. Were you able to post your story?

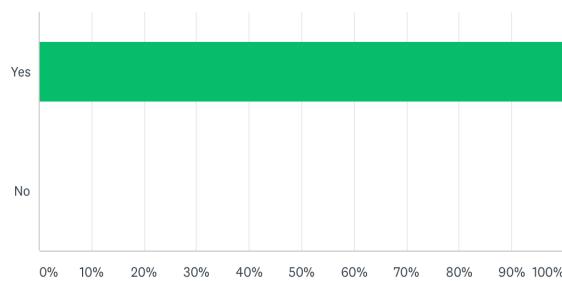
Answered: 5 Skipped: 0



| ANSWER CHOICES | RESPONSES |
|----------------------|-----------|
| ▼ Yes | 100.00% |
| ▼ No | 0.00% |
| Total Respondents: 5 | |

12. Were you able to make a comment below others' stories?

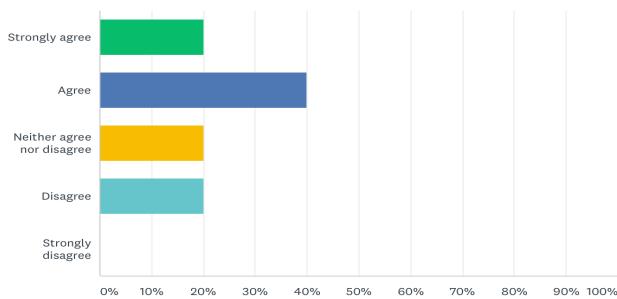
Answered: 5 Skipped: 0



| ANSWER CHOICES | RESPONSES |
|----------------------|-----------|
| ▼ Yes | 100.00% |
| ▼ No | 0.00% |
| Total Respondents: 5 | |

13. I think that I would like to use the app frequently.

Answered: 5 Skipped: 0

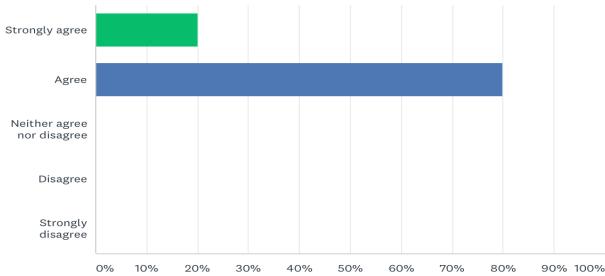


| ANSWER CHOICES | RESPONSES |
|----------------------------|-----------|
| Strongly agree | 20.00% |
| Agree | 40.00% |
| Neither agree nor disagree | 20.00% |
| Disagree | 20.00% |
| Strongly disagree | 0.00% |

Total Respondents: 5

14. I think the app is easy to use.

Answered: 5 Skipped: 0

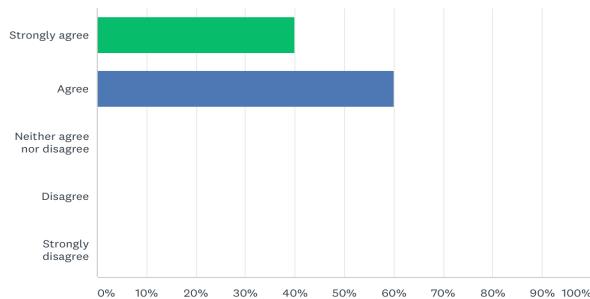


| ANSWER CHOICES | RESPONSES |
|----------------------------|-----------|
| Strongly agree | 20.00% |
| Agree | 80.00% |
| Neither agree nor disagree | 0.00% |
| Disagree | 0.00% |
| Strongly disagree | 0.00% |

Total Respondents: 5

15. I think the layout of the app is intuitive.

Answered: 5 Skipped: 0

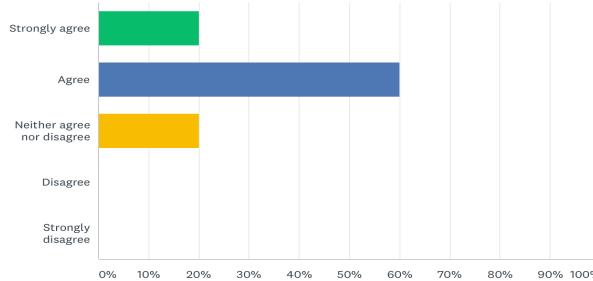


| ANSWER CHOICES | RESPONSES |
|----------------------------|-----------|
| Strongly agree | 40.00% |
| Agree | 60.00% |
| Neither agree nor disagree | 0.00% |
| Disagree | 0.00% |
| Strongly disagree | 0.00% |

Total Respondents: 5

16. I found the various functions in this app were well integrated.

Answered: 5 Skipped: 0

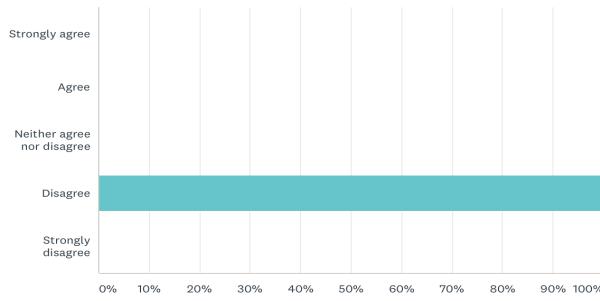


| ANSWER CHOICES | RESPONSES |
|----------------------------|-----------|
| Strongly agree | 20.00% |
| Agree | 60.00% |
| Neither agree nor disagree | 20.00% |
| Disagree | 0.00% |
| Strongly disagree | 0.00% |

Total Respondents: 5

17. I thought there was too much inconsistency in this app.

Answered: 5 Skipped: 0

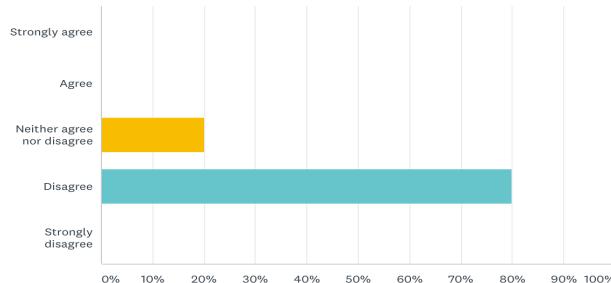


| ANSWER CHOICES | RESPONSES |
|----------------------------|-----------|
| Strongly agree | 0.00% |
| Agree | 0.00% |
| Neither agree nor disagree | 0.00% |
| Disagree | 100.00% |
| Strongly disagree | 0.00% |

Total Respondents: 5

18. I was able to easily find my way around the app to get the information I was looking for.

Answered: 5 Skipped: 0

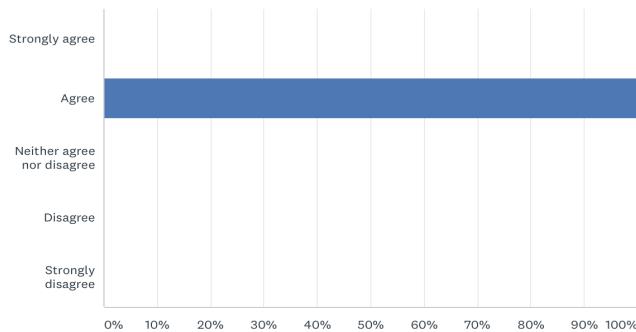


| ANSWER CHOICES | RESPONSES |
|----------------------------|-----------|
| Strongly agree | 0.00% |
| Agree | 0.00% |
| Neither agree nor disagree | 20.00% |
| Disagree | 80.00% |
| Strongly disagree | 0.00% |

Total Respondents: 5

19. I am satisfied with the app.

Answered: 5 Skipped: 0

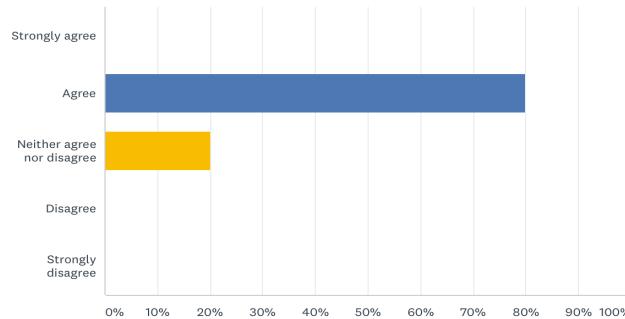


| ANSWER CHOICES | RESPONSES |
|----------------------------|-----------|
| Strongly agree | 0.00% |
| Agree | 100.00% |
| Neither agree nor disagree | 0.00% |
| Disagree | 0.00% |
| Strongly disagree | 0.00% |

Total Respondents: 5

20. I would like to recommend the app to my friends, colleagues or family members who want to get some information about Cardiff.

Answered: 5 Skipped: 0



| ANSWER CHOICES | RESPONSES |
|----------------------------|-----------|
| Strongly agree | 0.00% |
| Agree | 80.00% |
| Neither agree nor disagree | 20.00% |
| Disagree | 0.00% |
| Strongly disagree | 0.00% |

Total Respondents: 5

REFERENCES

1. Android Developers. (2018). *Download Android Studio and SDK tools* | *Android Developers*. [online] Available at: <https://developer.android.com/studio/index> [Accessed 31 Jul. 2018].
2. Ducrohet, X. (2018). *Android Studio: An IDE built for Android*. [online] Android Developers Blog. Available at: <https://android-developers.googleblog.com/2013/05/android-studio-ide-built-for-android.html> [Accessed 13 Aug. 2018].
3. Legislation.gov.uk. (2018). *Data Protection Act 2018*. [online] Available at: <http://www.legislation.gov.uk/ukpga/2018/12/contents/enacted> [Accessed 24 Jul. 2018].
4. McConnell, Steve (2004). *Code Complete, 2nd edition*. Microsoft Press. ISBN 1-55615-484-4
5. Medrano, R. (2018). *Future Trends of Popularizing API Use*. [online] API Management Blog - Akana. Available at: <https://blog.akana.com/future-trends-of-popularizing-api-use/> [Accessed 19 Aug. 2018].
6. Newsapi.org. (2018). Documentation - News API. [online] Available at: <https://newsapi.org/docs> [Accessed 19 Aug. 2018].
7. Saarinen, M. (2018). Best sat-nav apps 2018. [online] Auto Express. Available at: <https://www.autoexpress.co.uk/accessories-tyres/63573/best-sat-nav-apps-2018> [Accessed 19 Aug. 2018].
8. Statista. (2018). *App stores: number of apps in leading app stores 2018* | *Statista*. [online] Available at: <https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/> [Accessed 21 Jul. 2018].
9. Techopedia.com. (2018). *What is Java Development Kit (JDK)? - Definition from Techopedia*. [online] Available at: <https://www.techopedia.com/definition/5594/java-development-kit-jdk> [Accessed 30 Jul. 2018].
10. Fulton, R.; Vandermolen, R. (2017). "Chapter 4: Requirements - Writing Requirements". *Airborne Electronic Hardware Design Assurance: A Practitioner's Guide to RTCA/DO-254*. CRC Press. pp. 89–93. ISBN 9781351831420.
11. IDC: The premier global market intelligence company. (2018). *IDC: Smartphone OS Market Share*. [online] Available at: <https://www.idc.com/promo/smartphone-market-share/os> [Accessed 22 Jul. 2018].
12. ISO/IEC 25010:2011(2018). *ISO/IEC 25010:2011 - Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models*. [online] Iso.org. Available at: <https://www.iso.org/standard/35733.html> [Accessed 19 Jul. 2018].
13. Worldpopulationreview.com. (2018). [online] Available at: <http://worldpopulationreview.com/world-cities/cardiff-population/> [Accessed 22 Jul. 2018].
14. Zelkowitz, M. (2004). *Advances in computers*. Amsterdam: Academic Press, pp.1-66. ISBN 9780080471907.