Seaton Valley Council Test Objective and Plan

# Objectives

The objective of testing the product is to discover any faults with its functionality or graphical interface. Removing all faults from the product is essential for delivering a product of high standard. Testing will also be used to make sure that the product adhered to the specification and requirements set during the planning phase.

During the development of the product, we will be continuously testing to eradicate the majority of faults. This is to reduce the workload required once a stable prototype has been produced. Version control will be used throughout as to aid the testing phase. Code coverage tools will also be utilized to measure the extent of our testing techniques.

The testing team will manage all testing documentation and software and will also be responsible for reporting found faults to the programming and design team.

The equipment required to test our product will be:

1. PC capable of running Android Studio
2. Android emulators of phones with multiple screen sizes
3. Physical Android phone with developer mode enabled
4. Micro USB cable

# Testing Techniques

Multiple approaches to testing the application will be used, these being:

* Black Box Testing - General testing of the application without understanding of implementation.
* White Box Testing - General testing of the application with understanding of implementation.
* Static Testing - Proofreading source code for typing errors.
* Dynamic Testing - Examination of inputs and outputs and other responses of the running application.
* Unit Testing - Testing individual functional elements.
* Agile Testing - Testing the application alongside development.
* Software Verification and Validation Testing - Verifying that the application fits the specified requirements.

# Test Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Test** | **Expected Result** | **Actual Result** | **Cause and Fix** | **Comments** |
| **1** | **General App Functionality** | | | | |
| 1.1 | Download the app/Upload to an Android phone. | App will be downloaded/uploaded to phone. | Both Android Studio and a standalone APK file install successfully. |  |  |
| 1.2 | Launching the app. | The app should be able to be launched across all target Android versions and display the apps home screen. | Worked in most instances, however if the app was launched when the cache was freshly cleared, and the app was removed from the android task manager the app would crash. |  |  |
| 1.3.1 | Component resizing to screen size. | All images/icons/components will be resized and positioned to fit the device screen size. | All components adapted to the screen appropriately, however on smaller device screen sizes the home pages icons were scaled to be too small and could not be clicked for navigation.  ***(xlarge screen sizes only)***  When viewing webviews on xlarge screen size devices the layout was not formatted suitably. | The icons labels were overlaying the icons, preventing the user from being able to click icons. Added a transparent grid over all icons that will be used for navigation instead.  Website layout for xlarge screen sizes was configured for desktop devices.  Removed header and footer of HTML code to improve layout. |  |
| 1.3.2 | Images are rescaled to desired size. | Image resolution adapts to screen size. | All images rescaled to the devices screen size. Larger screens had higher resolution images and smaller screens have more compressed images. |  | To reduce application lag. Using assets generated from nsimage.brosteins.com. |
| **2** | **Navigation** | | | |  |
| 2.1 | Page icons linking to other pages. | When an icon is clicked the user is taken to the selected page immediately. | On icon click the application opened the desired page after a brief animation. |  |  |
| 2.2 | Navigation bar. | When the screen is swiped from the left, no matter what instance the app is in, the navigation bar will be opened. | Across all app states/instances the navigation bar/draw opened. When done on the home page the draw opened out much farther than other pages. | Small configuration error regarding the other pages navigation draw. Changed other pages value to match home page. |  |
| 2.3 | Navigation bar icons linking to other pages. | When an icon in the navigation bar pane is clicked the user is taken to the corresponding page. | All navigation bar icons linked to the desired page. |  |  |
| 2.4 | Share option. | When the share option is selected, the user will be prompted to select a platform to share a specified advertisement message on. | The user was presented with various platforms to share the message on. Once a platform was selected the message was in the platforms message field to be sent (the procedure varies from platform to platform). |  | Uses devices installed ACTION\_SEND platforms (email, text, installed messaging apps etc.).  Tested using Facebook Messenger, Snapchat, Gmail, and Text. |
| **3** | **Weather** | | | | |
| 3.1 | Data/Internet connection check. | Checks if the device has a data/internet connection, if not disable the feature. | When data/internet was turned off no weather data was displayed. |  |  |
| 3.2 | Connection to OpenWeather API using API key. | A successful connection to the OpenWeather API using the stored API key. | No compilation/authentication errors generated with stored API key and API methods worked accordingly. |  |  |
| 3.3 | Download weather data for Seaton Valley. | Using stored longitude and latitude, fetch weather JSON for Seaton Valley. | JSON data downloads with no errors using the stored longitude and latitude. |  |  |
| 3.4 | Extract, convert and display temperature. | Seaton Valley temperature (in kelvin) is converted to Celsius, rounded, and then displayed on screen. | Error toast message did not display, and the temperature was extracted, rounded, and displayed (using Volley API). |  | Compared with the value from OpenWeather website. |
| 3.5 | Fetch and display corresponding weather icon. | Using the fetched temperature, fetch the corresponding weather icon from OpenWeather and display next to temperature. | Using stored weather data, the correct icon was fetched, resized to 100x100, and displayed next to the temperature using the Picasso API. |  | Compared with the icon from OpenWeather website. |
| 3.6 | Display full weather report upon click. | When the weather icon is clicked open the full weather report in app webview. | When the weather icon was clicked the full weather report of Seaton Valley was displayed in a webview. |  | Full weather report courtesy of OpenWeather.  Only if there is a data/internet connection. |
| **4** | **Carousel** | | | | |
| 4.1.1 | Latest news articles loaded. | Latest news articles are loaded into the frame space and include a brief introduction to the article in question. | The three latest news articles were loaded into the frame space. However, the “read more” option was overflowing into the frame covering the article text. | The brief article description was sometimes too long causing it display behind the read more option. Added a limit to the length of the description. |  |
| 4.1.2 | Linking to news article. | Clicking on an article in the carousel displayed the article in another instance. | Correct article was loaded in a separate instance. |  |  |
| 4.2 | Navigation between articles. | Swiping left and right navigates through the loaded news articles. | Sliding left and right did navigate through the articles but did not lock onto each article. | Missing implementation. Added code to lock onto articles. |  |
| **5** | **News** | | | | |
| 5.1 | Data/Internet connection check. | Checks if the device has a data/internet connection, if not notify user. | Opening the news without data/internet connection displays a prompt to the user. |  |  |
| 5.2.1 | Load cached news. | If the device contains cached news from previous use, load said news. | Cached news is loaded, even without data/internet connection. |  |  |
| 5.3 | Loading icon. | Display loading icon while processing information. | Icon loaded. |  |  |
| 5.4.1 | Fetch data from Seaton Valley website. | Fetch data from Seaton Valley website JSON file. | If internet/data connection available; data is fetched from the Seaton Valley website in JSON format. |  | Data retrieved from: <http://seatonvalleycommunitycouncil.gov.uk/> |
| 5.4.2 | Extract news fields from data. | Filter through JSON file and extract wanted data. | Relevant data is extracted from the JSON files successfully. |  |  |
| 5.4.3 | Generate news cards from extracted data. | For each news element found in the data, generate a news card. | All cards with most recent news are created up to the user-defined limit. |  | Limits to number of posts to display can be found in settings. |
| 5.5 | Display all generated news cards. | All generated news cards are displayed on screen in a list format and are also scrollable. | All cards are displayed in a reverse chronological order list and can be scrolled through. |  |  |
| 5.6 | Link to news article. | Pressing on a news card should display the corresponding article in a WebView. | The correct article is displayed when a card is selected. |  | If there is no internet connection a blank screen will be displayed with a prompt notifying the user that a connection is required. |
| 5.7 | Caching data. | After a successful download of news cards, store cards in cache for future use. | All loaded cards were stored in cache successfully. |  |  |
| 5.8 | Sharing article. | Once an article is displayed, clicking the share option in the top right shares to the users selected platform (Sharing introduction of article and a URL link to the full article). | The selected article’s introduction and URL link was sent using the selected platform. |  | Uses devices installed ACTION\_SEND platforms (email, text, installed messaging apps etc.).  Tested using Facebook Messenger, Snapchat, Gmail, and Text. |
| **6** | **Twitter** | | | | |
| 6.1 | Data/Internet connection check. | Checks if the device has a data/internet connection, if not notify user. | Opening the tweet activity, without data/internet connection displays a prompt to the user. |  |  |
| 6.2 | Connection to Twitter Kit API using API key. | A successful connection to the Twitter Kit API using the stored API key. | The application successfully connects to the Twitter API. |  |  |
| 6.3 | Target and display Seaton Valley twitter timeline. | Using Twitter Kit API target account @SeatonValleyCC and display its timeline. | Tweets from @SeatonValleyCC are loaded correctly. |  |  |
| 6.4.1 | Log in to a Twitter account. | When a favourite option of a tweet is selected display a new screen allowing the user to log in to a valid Twitter account. | When the user was not already logged in, attempting to favourite a tweet allowed them to log into a Twitter account. |  |  |
| 6.4.2 | Favourite a tweet. | If the user is logged in, clicking the favourite option favourites the tweet in the users Twitter account. | When the favourite option was selected the tweet was favourited using the users Twitter account. |  | Confirmed by comparing to the Twitter website. |
| 6.4.3 | Share a tweet. | When the share option of a tweet is selected, the user can share said tweet to a platform of their choice. | The selected tweet (URL) was shared to the specified platform. |  | Uses devices installed ACTION\_SEND platforms (email, text, installed messaging apps etc.).  Tested using Facebook Messenger, Snapchat, Gmail, and Text. |
| 6.5 | Refresh timeline. | When the user drags down from the top of the screen the timeline is refreshed and reloaded. | When dragged down from the top the timeline was refreshed. |  |  |
| 6.6 | Displaying tweet in third party application. | When a tweet from the timeline is selected open said tweet in devices default app. | When a tweet is selected the tweet either loads in the default browser or the user’s twitter app. |  | Default app being either the official twitter app or an internet browser app. |
| **7** | **Events** | | | | |
| 7.1 | Data/Internet connection check. | Checks if the device has a data/internet connection, if not notify user. | Opening events without data/internet connection displays a prompt to the user. |  |  |
| 7.2 | Load cached events. | If the device contains cached events from previous use, load said events. | Cached news is loaded, even without data/internet connection. |  |  |
| 7.3 | Loading icon. | Display loading icon while processing information. | Icon loaded. |  |  |
| 7.4.1 | Fetch data from Seaton Valley website. | Fetch data from Seaton Valley website JSON file. | If internet/data connection available; data is fetched from the Seaton Valley website in JSON format. |  |  |
| 7.4.2 | Extract event fields from data. | Filter through JSON file and extract wanted data. | Relevant data is extracted from the JSON files successfully. |  |  |
| 7.4.3 | Generate event cards from extracted data. | For each event element found in the data, generate an event card. | All cards with most recent news are created up to the user-defined limit. |  |  |
| 7.5 | Display all generated event cards. | All generated event cards are displayed on screen in a list format and are also scrollable. | All cards are displayed in a reverse chronological order list and can be scrolled through. |  |  |
| 7.6 | Link to event information. | Clicking on an event should display the events information in a WebView. | The correct article is displayed when a card is selected. |  | If there is no internet connection a blank screen will be displayed with a prompt notifying the user that a connection is required. |
| 7.7 | Caching data. | After a successful download of event cards, store cards in cache for future use. | All loaded cards were stored in cache successfully. |  |  |
| 7.8 | Sharing event. | Once an event is displayed, clicking the share option in the top right shares to the users selected platform (Sharing introduction of event and a URL link to the full event page). | The selected event’s introduction and URL link was sent using the selected platform. |  | Uses devices installed ACTION\_SEND platforms (email, text, installed messaging apps etc.).  Tested using Facebook Messenger, Snapchat, Gmail, and Text. |
| 7.9 | Add event to calendar. | When an event is selected, clicking the “+” icon in the bottom right allows a user to add the event to their devices calendar. | The event was added to the devices calendar app and the title and description fields were automatically filled out. However, the “+” icon was not shaded correctly on Android KitKat. | KitKat does not have the “background-tint” option. Implemented legacy code. |  |
| **8** | **Places** | | | |  |
| 8.1 | Data/Internet connection check. | Checks if the device has a data/internet connection, if not notify user. | Opening places without data/internet connection displays a prompt to the user. |  |  |
| 8.2.1 | Place categories. | Place categories are displayed in list form. | Categories are displayed correctly. |  |  |
| 8.2.2 | Clicking a category. | Display places from selected category. | Places for corresponding category displayed correctly. |  |  |
| 8.3.1 | Places List. | Places are displayed in list form. | Each place and its information are displayed in a list format. |  |  |
| 8.3.2 | User clicks a location. | Pinpoint the selected location on Google Maps fragment. | Pin is displayed on the location of the place selected. |  |  |
| 8.4 | User clicks navigate. | Google Maps app should launch and begin navigation from current location to selected location. | When the pin is pressed an option to open google maps or open navigation from your current location is displayed on the map fragment. |  |  |
| **9** | **Contact** | | | | |
| 9.1.1 | Validate first name field. | If first name validation passes allow message send, else stop message send and notify user. | If the first name field is empty a user prompt is displayed, otherwise first name is valid, and message was sent. |  |  |
| 9.1.2 | Validate last name field. | If last name validation passes allow message send, else stop message send and notify user. | If the last name field is empty a user prompt is displayed, otherwise first name is valid, and message was sent. |  |  |
| 9.1.3 | Validate email. | If email validation passes allow message send, else stop message send and notify user. | When email is invalid or empty a user prompt is displayed, otherwise message was sent. |  |  |
| 9.1.4 | Validate message field. | If message validation passes (if not true) allow message send, else stop message send and notify user. | If the message is longer than 20 characters then the message is valid and sent, however if it is null or below 20 characters the user is notified. |  |  |
| 9.2.1 | Send message button click. | Clicking the button should send an email to the specified email account. If any validation checks fail, don’t send the email, and notify the user of failed fields. | The user is notified when fields are found to be invalid by checks. If all fields are valid, email is created and sent. |  |  |
| 9.2.2 | Contact enquiry stopped from being sent if there’s no data/internet connection. | Email blocked from being sent and notification displayed informing user to turn on data/Wi-Fi. | Email was not sent, and notification did display. |  |  |
| 9.2.3 | Contact enquiry received in Seaton Valley email inbox. | Email populated with form inputs sent to the Seaton Valley email. | Email with correct content was received. |  |  |
| 9.2.4 | Confirmation email received in submitter's inbox. | Email, confirming the contact enquiry was sent, received in submitter's email inbox. | Confirmation email was received. |  |  |
| **10** | **Report** | | | | |
| 10.1.1 | Validate first name field. | If first name validation passes allow message send, else stop message send and notify user. | If the first name field is empty a user prompt is displayed, otherwise first name is valid, and message was sent. |  |  |
| 10.1.2 | Validate last name field. | If last name validation passes allow message send, else stop message send and notify user. | If the last name field is empty a user prompt is displayed, otherwise first name is valid, and message was sent. |  |  |
| 10.1.3 | Validate email field. | If email validation passes allow message send, else stop message send and notify user. | When email is invalid or empty a user prompt is displayed, otherwise message was sent. |  |  |
| 10.1.4 | Validate report message field. | If message validation passes (if not null) allow message send, else stop message send and notify user. | If the message is longer than 20 characters then the message is valid and sent, however if it is null or below 20 characters the user is notified. |  |  |
| 10.2.1 | Send report button click. | Clicking the button should send a report email to the specified email account. If any validation checks fail, don’t send the email, and notify the user of failed fields. | The user is notified when fields are found to be invalid by checks. If all fields are valid, email is created and sent. |  |  |
| 10.2.2 | Report stopped from being sent if there’s no data/internet connection. | Email blocked from being sent and notification displayed informing user to turn on data/Wi-Fi. | Email was not sent, and notification did display. |  |  |
| 10.2.3 | Report received in Seaton Valley email inbox. | Email populated with form inputs sent to the Seaton Valley email. | Email with correct content was received. |  |  |
| 10.2.4 | Confirmation email received in submitter's inbox. | Email, confirming the report was sent, received in submitter's email inbox. | Confirmation email was received. |  |  |
| **11** | **Council** | | | |  |
| 11.1 | Correct information displayed. | Correct information is displayed from the strings xml file. | All specified information was displayed correctly. |  |  |
| **12** | **Info** | | | |  |
| 12.1 | Correct information displayed. | Correct information is displayed from the strings xml file. | All specified information was displayed correctly.  However, redundant information was included in the text. | Irrelevant text was accidently sourced and was stored in the strings file. Removed irrelevant information. |  |
| **13** | **Settings** | | | |  |
| 13.1 | Clear Cache Button. | Device cache is cleared when button is clicked alongside notification message being displayed. | Application cache was cleared, and a notification message was displayed. |  | Tested by comparing application cache size. |
| 13.2.1 | Limit News Posts | Only the specified number of news cards should be displayed in the News activity. | News posts are limited to the pre-set value selected by the user. |  |  |
| 13.2.2 | Limit Event Posts | Only the specified number of event cards should be displayed in the Events activity. | Event posts are limited to the pre-set value selected by the user. |  |  |
| 13.3 | Option to disable notifications. | When the slider is turned to off the device will not receive notifications from the Seaton Valley Firebase service. | Notifications were received inconsistently when compared to the sliders state. | The slider only applied to foreground notifications and not background notifications. Implemented a subscribe messenger channel in the Firebase console and made it so that turning the slider off unsubscribed from the channel. |  |