## CM3110 – Mobile App Development

Coursework: Will the Sun Shine Again? App Testing Document

Code/Method Tested	Test Data	Expected Output	Actual Output
WelcomeActivity	n/a	The super class onCreate() method is	The super class onCreate() method is
onCreate() Method		called with the savedInstanceState	called with the savedInstanceState
		Bundle, and the layout of the Activity	Bundle, and the layout of the Activity
The purpose of this test is to check that		is set to activity_welcome.xml. A	is set to activity_welcome.xml. A
the onCreate method runs correctly		debug log message is made saying	debug log message is made saying
when performing tasks that only need to		"In the onCreate event handler".	"In the onCreate event handler".
happen once during the activity's life.			
This includes assigning values to		The forecastDB field is assigned a	The forecastDB field is assigned a
variables, calling any helper methods,		value – the return value from the	value – the return value from the
and logging any messages once		getDatabase() method in the	getDatabase() method in the
commands have been successfully		ForecastDatabase class. The	ForecastDatabase class. The
executed.		forecastDAO field is assigned the	forecastDAO field is assigned the
		return of the forecastDao() method	return of the forecastDao() method
From a broader perspective, this is a test		from the forecastDB field. A log	from the forecastDB field. A log
to make sure the app launches correctly.		debug message is made saying	debug message is made saying
		"Initialised database okay".	"Initialised database okay".
		The sharedPrefs field is assigned the	The sharedPrefs field is assigned the
		return value from	return value from
		getSharedPreferences, using the	getSharedPreferences, using the
		sharedPrefFile String as a parameter.	sharedPrefFile String as a parameter.
		mEditor is assigned the return from	mEditor is assigned the return from
		the edit() method of the sharedPrefs	the edit() method of the sharedPrefs
		field. The updateLocations() helper	field. The fetchLocations() helper
		method is called. A debug log	method is called. A debug log

		message is made saying "Initialised the shared preferences okay".	message is made saying "Initialised the shared preferences okay".
		The btn variables are assigned Button components from the interface. These all have an onClickListener set. A debug log message is made saying "initialised the buttons and their click listeners okay".	The btn variables are assigned Button components from the interface. These all have an onClickListener set. A debug log message is made saying "initialised the buttons and their click listeners okay".
		The spinLocations variable is assigned the spinnerLocations View component from the interface. The adapter variable is assigned a new ArrayAdapter object, using the ArrayList of chosen locations as one of the parameters. The spinLocations variable has its adapter set to adapter. The spinner has an onItemSelectedListener attached. A debug log message is made saying "Set up spinners alright".	The spinLocations variable is assigned the spinnerLocations View component from the interface. The adapter variable is assigned a new ArrayAdapter object, using the ArrayList of chosen locations as one of the parameters. The spinLocations variable has its adapter set to adapter. The spinner has an onItemSelectedListener attached. A debug log message is made saying "Set up spinners alright".
WelcomeActivity onStart() Method  The purpose of this test is to check that the onStart method runs correctly when making the activity visible to the user and preparing it for interactivity.	n/a	The super class onStart() method is called and a debug log message is made saying "In the onStart event handler".	The super class onStart() method is called and a debug log message is made saying "In the onStart event handler".

WelcomeActivity onPause() Method	n/a	The super class onPause() method is called and a debug log message is made saying "In the onPause event	The super class onPause() method is called and a debug log message is made saying "In the onPause event
The purpose of this test is to check that the onPause method runs correctly when indicating that this activity is not in the foreground but should not be destroyed as it is expected to be		handler".	handler".
resumed soon.			
WelcomeActivity onStop() Method	n/a	The super class onStop() method is called and a debug log message is made saying "In the onStop event	The super class onStop() method is called and a debug log message is made saying "In the onStop event
The purpose of this test is to check that the onStop method runs correctly when a new activity has covered the screen, or when this activity has been terminated.		handler".	handler".
WelcomeActivity onDestroy() Method	n/a	The super class onDestroy() method is called and a debug log message is made saying "In the onDestroy event	The super class onDestroy() method is called and a debug log message is made saying "In the onDestroy event
The purpose of this test is to check that the onDestroy method runs correctly when the user has completely finished using the activity.		handler".	handler".
WelcomeActivity onResume() Method	n/a	The updateLocations helper method is called, the super class onDestroy() method is called and a debug log	The updateLocations helper method is called, the super class onDestroy() method is called and a debug log
The purpose of this test is to check that the onResume method runs correctly when the user returns to this activity from another or from outside of the app.		message is made saying "In the onResume event handler".	message is made saying "In the onResume event handler".

WelcomeActivity	No button on the	Nothing happens as no methods are	Nothing happens as no methods are
onClick() Method	interface is	called after the interface is	called after the interface is
	pressed.	displayed.	displayed.
The purpose of this test is to check that			
the onClick method does not run unless			
a button is pressed.			
WelcomeActivity	The button to	The onClick method/event handler is	The onClick method/event handler is
onClick() Method	move to the	called, sending the	called, sending the
	Preferences	buttonPreferences view as a	buttonPreferences view as a
The purpose of this test is to check that	screen is pressed.	parameter. A new context object is	parameter. A new context object is
the onClick method processes the		created and assigned the value of	created and assigned the value of
clicking of each button correctly.		the WelcomeActivity context. A	the WelcomeActivity context. A
		debug log message is made saying	debug log message is made saying
		"In the onClick method".	"In the onClick method".
		The if statement is entered. The	The if statement is entered. The
		condition should match the initial	condition should match the initial
		condition (i.e. the id of the view is	condition (i.e. the id of the view is
		buttonPreferences) so the body of	buttonPreferences) so the body of
		the if condition is executed.	the if condition is executed.
		A Class object is created to store the	A Class object is created to store the
		PreferencesActivity.class object. An	PreferencesActivity.class object. An
		Intent object is created using the	Intent object is created using the
		context object and the destination	context object and the destination
		class object. The startActivity()	class object. The startActivity()
		method is called using the newly	method is called using the newly
		created intent as the parameter,	created intent as the parameter,
		launching the preferences screen.	launching the preferences screen.

WelcomeActivity	The button to	The onClick method/event handler is	The onClick method/event handler is
onClick() Method	move to the	called, sending the buttonSearch	called, sending the buttonSearch
	Search screen is	view as a parameter. A new context	view as a parameter. A new context
The purpose of this test is to check that	pressed.	object is created and assigned the	object is created and assigned the
the onClick method processes the		value of the WelcomeActivity	value of the WelcomeActivity
clicking of each button correctly.		context. A debug log message is	context. A debug log message is
		made saying "In the onClick	made saying "In the onClick
		method".	method".
		The if statement is entered. The	The if statement is entered. The
		condition doesn't match the initial	condition doesn't match the initial
		condition, so the body of the if	condition, so the body of the if
		condition is skipped. The condition	condition is skipped. The condition
		should match the else if condition	should match the else if condition
		(i.e. the id of the view is	(i.e. the id of the view is
		buttonSearch) so the body of the	buttonSearch) so the body of the
		else if condition is executed.	else if condition is executed.
		A Class object is created to store the	A Class object is created to store the
		SearchActivity.class object. An Intent	SearchActivity.class object. An Intent
		object is created using the context	object is created using the context
		object and the destination class	object and the destination class
		object. The startActivity() method is	object. The startActivity() method is
		called using the newly created intent	called using the newly created intent
		as the parameter, launching the	as the parameter, launching the
		search screen.	search screen.
WelcomeActivity	The button to	The onClick method/event handler is	The onClick method/event handler is
onClick() Method	download new	called, sending the buttonDownload	called, sending the buttonDownload
	forecasts is	view as a parameter. A new context	view as a parameter. A new context
	pressed.	object is created and assigned the	object is created and assigned the

The purpose of this test is to check that the onClick method processes the clicking of each button correctly.

value of the WelcomeActivity context. A debug log message is made saying "In the onClick method".

The if statement is entered. The condition doesn't match the initial condition, so the body of the if condition is skipped. The condition of the else if condition isn't met, so the body of the else if condition is skipped. The condition should match the else condition (i.e. the id of the view is buttonDownload) so the body of the else condition is executed.

The deleteForecastsDB method is called and a debug log message is made saying "Old forecasts deleted".

A new DownloaderTask object is created and then a new String is created.

The String is created by concatenating the JSON\_WEB\_ADR\_PART1, currentLocation, and JSON\_WEB\_ADR\_PART2 Strings. If there are any spaces in the

value of the WelcomeActivity context. A debug log message is made saying "In the onClick method".

The if statement is entered. The condition doesn't match the initial condition, so the body of the if condition is skipped. The condition of the else if condition isn't met, so the body of the else if condition is skipped. The condition should match the else condition (i.e. the id of the view is buttonDownload) so the body of the else condition is executed.

The deleteForecastsDB method is called and a debug log message is made saying "Old forecasts deleted".

A new DownloaderTask object is created and then a new String is created.

The String is created by concatenating the JSON\_WEB\_ADR\_PART1, currentLocation, and JSON\_WEB\_ADR\_PART2 Strings. If there are any spaces in the

		currentLocation String, these are	currentLocation String, these are
		replaced with +s before the	replaced with +s before the
		concatenation. The complete URL is	concatenation. The complete URL is
		then used in a debug log message.	then used in a debug log message.
		then used in a debug log message.	then used in a debug log message.
		The try/catch statement is entered.	The try/catch statement is entered.
		A URL object is created used the	A URL object is created used the
		concatenated String. A debug	concatenated String. A debug
		message is made after the URL is	message is made after the URL is
		made. The DownloaderTask object is	made. The DownloaderTask object is
		then sent the URL object for its	then sent the URL object for its
		execute method.	execute method.
		If the URL is incorrect in any way and	If the URL is incorrect in any way and
		cannot be used, the catch statement	cannot be used, the catch statement
		is executed and a debug log message	is executed and a debug log message
		printed to say so.	printed to say so.
WelcomeActivity	An item is	The item at the selected position in	The item at the selected position in
onItemSelected() Method	selected from the	the spinner is converted to a String	the spinner is converted to a String
	spinner.	and stored in the currentLocation	and stored in the currentLocation
The purpose of this test is to check that		variable. The	variable. The
the onItemSelected method correctly		getForecastsForLocation() method is	getForecastsForLocation() method is
processes the selection of any item in		then called. A Toast telling the user	then called. A Toast telling the user
the spinner.		which location they have just	which location they have just
		selected appears.	selected appears.
WelcomeActivity	No item has been	Nothing happens as this method has	Nothing happens as this method has
onNothingSelected() Method	newly selected (it	no body.	no body.
	is not possible to		

	1	T	
The purpose of this test is to check that	have no item		
the onNothingSelected method correctly	selected at all).		
processes the lack of selection of any			
item in the spinner.			
WelcomeActivity	n/a	The sharedPrefs variable is initialised	The sharedPrefs variable is initialised
updateLocations() Method		using the getSharedPreferences() method and the sharedPrefFile	using the getSharedPreferences() method and the sharedPrefFile
The purpose of this test is to check that		parameter. A debug log message is	parameter. A debug log message is
the user's chosen locations are retrieved		made to confirm this has happened.	made to confirm this has happened.
from the shared preferences.			
		A counter variable is initialised. A while loop is entered. Which goes through all the LOCATION_X Strings in the shared preferences until one returns a null.	A counter variable is initialised. A while loop is entered. Which goes through all the LOCATION_X Strings in the shared preferences until one returns a null.
		If a String does not return a null, it is logged with a debug message before it is added to the chosenLocations ArrayList. This ArrayList's size is then printed upon completion of the while loop.	If a String does not return a null, it is logged with a debug message before it is added to the chosenLocations ArrayList. This ArrayList's size is then printed upon completion of the while loop.
		A new ArrayAdapter is created using the updated ArrayList. The same	A new ArrayAdapter is created using the updated ArrayList. The same
	,	adapter is then set for the spinner.	adapter is then set for the spinner.
WelcomeActivity	n/a	The sharedPrefs variable is initialised	The sharedPrefs variable is initialised
saveLocations() Method		with the getSharedPreferences()	with the getSharedPreferences()
		method and sharedPrefFile variable	method and sharedPrefFile variable
		as the parameter. The mEditor is also	as the parameter. The mEditor is also

The purpose of this test is to check that the user's chosen locations are saved in the shared preferences.		initialised as the return from the edit() method of the sharedPrefs object.	initialised as the return from the edit() method of the sharedPrefs object.
		A for loop is entered for the length of the chosenLocations ArrayList. For each item in the ArrayList, it is added to the shared preferences in a String called LOCATION_X (where X is the counter/ArrayList position + 1).	A for loop is entered for the length of the chosenLocations ArrayList. For each item in the ArrayList, it is added to the shared preferences in a String called LOCATION_X (where X is the counter/ArrayList position + 1).
		The mEditor then applies these changes.	The mEditor then applies these changes.
WelcomeActivity updateMainSummary() Method  The purpose of this test is to check that the main summary section of the	A Forecast object.	All the relevant View components from the interface are stored in appropriately named variables using the findViewById method.	All the relevant View components from the interface are stored in appropriately named variables using the findViewByld method.
interface is correctly updated to show the current forecast for the current location.		The values for the TextViews are set using the setText() method. Each value is accessed through the Forecast object's getter methods. Any addition Strings are concatenated and any necessary conversion are done through the Converter helper class.	The values for the TextViews are set using the setText() method. Each value is accessed through the Forecast object's getter methods. Any addition Strings are concatenated and any necessary conversion are done through the Converter helper class.
		The setWeatherIcon helper method is called to change the current weather icon/image.	The setWeatherIcon helper method is called to change the current weather icon/image.

WelcomeActivity	A List of Forecast	All the needed Forecast objects from	All the Forecast objects from the List
updateHourlyForecasts () Method	objects.	the List are stored in their own	are stored in their own individual
		individual Forecast objects/variable	Forecast objects/variable for ease of
The purpose of this test is to check that the hourly forecasts section of the		for ease of access later.	access later.
interface is correctly updated to show		All the relevant View components	All the relevant View components
the current and next 5 (3 hour)		from the interface are stored in	from the interface are stored in
increments for the current location.		appropriately named variables using the findViewById method.	appropriately named variables using the findViewById method.
		The values for the TextViews are set using the setText() method. Each	The values for the TextViews are set using the setText() method. Each
		value is accessed through the	value is accessed through the
		Forecast object's getter methods.  Any addition Strings are	Forecast object's getter methods.  Any addition Strings are
		concatenated.	concatenated.
		The setWeatherIcon helper method	The setWeatherIcon helper method
		is called to change the weather icons/images.	is called to change the weather icons/images.
WelcomeActivity	A List of Forecast	All the needed Forecast objects from	All the needed Forecast objects from
updateDailyForecasts() Method	objects.	the List are stored in their own	the List are stored in their own
		individual Forecast objects/variable	individual Forecast objects/variable
The purpose of this test is to check that		for ease of access later.	for ease of access later.
the hourly forecasts section of the		All the color and Manager and a	All the released Management
interface is correctly updated to show		All the relevant View components from the interface are stored in	All the relevant View components from the interface are stored in
the current and next 5 (3 hour)			
increments for the current location.		appropriately named variables using the findViewById method.	appropriately named variables using the findViewById method.
		the interious	the inicial way in the inicial and inicial

	The values for the TextViews are set
	using the setText() method. Each using the setText() method. Each
	value is accessed through the value is accessed through the
	Forecast object's getter methods. Forecast object's getter methods.
	Any addition Strings are Any addition Strings are
	concatenated. concatenated.
WelcomeActivity A weather	The weather id is converted to a   The weather id is converted to a
setWeatherIcon() Method from a Fore	ast smaller code by dividing it by 100. smaller code by dividing it by 100.
object and	n
The purpose of this test is to check that   ImageView	An if statement is entered. If the An if statement is entered. If the
the specified ImageView has its source	original id value is equal to 800, the original id value is equal to 800, the
changed to accurately reflect the	ImageView has its source changed to   ImageView has its source changed to
weather given for the location/forecast	display a sun icon. display a sun icon.
time.	
	If not, the else statement's body is
	executed. Within this is a switch executed. Within this is a switch
	statement based on the converted statement based on the converted
	id. Depending on the value of the id. Depending on the value of the
	converted id, an image resource is converted id, an image resource is
	set that accurately corresponds to set that accurately corresponds to
	the reference material from the API the reference material from the API
	codes. codes.
WelcomeActivity An ArrayLis	of A new InsertForecastsTask object is A new InsertForecastsTask object is
insertForecastsDB() Method Forecast ob	ects. created. created.
The purpose of this test is to check that	The ArrayList of Forecasts objects is The ArrayList of Forecasts objects is
the method correctly calls the right task	converted into an array for use with converted into an array for use with
to insert the newly downloaded	the DAO. The task is told to execute the DAO. The task is told to execute
forecasts into the database.	using the newly created array. using the newly created array.

WelcomeActivity deleteForecastsDB() Method	n/a	A new DeleteForecastsTask object is created and then told to execute.	A new DeleteForecastsTask object is created and then told to execute.
The purpose of this test is to check that the method correctly calls the right task to delete all existing forecasts from the database.			
WelcomeActivity getForecastsForCurrentLocation() Method	n/a	A new UpdateGUIForLocationTask object is created and then told to execute.	A new UpdateGUIForLocationTask object is created and then told to execute.
The purpose of this test is to check that the method correctly calls the right task to retrieve the required forecasts from the database.			
WelcomeActivity - InsertForecastsTask doInBackground() Method  The purpose of this test is to check that the method interacts with the DAO to add forecasts to the database.	An array of Forecast objects.	The forecastDAO object is used to call its insertForecasts() method using the array of Forecast objects supplied in the parameters.	The forecastDAO object is used to call its insertForecasts() method using the array of Forecast objects supplied in the parameters.
WelcomeActivity - InsertForecastsTask onPostExecute() Method  The purpose of this test is to check that the method correctly finishes the job of the task.	A Void object.	The super class's onPostExecute method is called using the Void object as the parameter.	The super class's onPostExecute method is called using the Void object as the parameter.
WelcomeActivity - DeleteForecastsTask doInBackground() Method	n/a	The forecastDAO object is used to call its deleteAllForecastsFor()	The forecastDAO object is used to call its deleteAllForecastsFor()

		method using the global	method using the global
The purpose of this test is to check that		currentLocation variable as its	currentLocation variable as its
the method interacts with the DAO to		parameter.	parameter.
delete all forecasts from the database			
for a specific location			
WelcomeActivity - DeleteForecastsTask	A Void object.	The super class's onPostExecute	The super class's onPostExecute
onPostExecute() Method		method is called using the Void	method is called using the Void
		object as the parameter.	object as the parameter.
The purpose of this test is to check that			
the method correctly finishes the job of			
the task.			
WelcomeActivity -	n/a	The forecastDAO object is used to	The forecastDAO object is used to
UpdateGUIForLocationTask		call its getAllForecastsForLocation()	call its getAllForecastsForLocation()
doInBackground() Method		method using the global	method using the global
		currentLocation variable as its	currentLocation variable as its
The purpose of this test is to check that		parameter.	parameter.
the method interacts with the DAO to			
get all forecasts from the database for a		This List of Forecast objects is	This List of Forecast objects is
specific location		returned.	returned.
WelcomeActivity -	A List of Forecast	The super class's onPostExecute	The super class's onPostExecute
UpdateGUIForLocationTask	objects.	method is called using the List as the	method is called using the List as the
onPostExecute() Method		parameter.	parameter.
The purpose of this test is to check that		An if else statement is entered. If the	An if else statement is entered. If the
the method correctly finishes the job of		if condition is met, i.e. the size of List	if condition is met, i.e. the size of List
the task.		passed in is greater than 0, the body	passed in is greater than 0, the body
		of the if statement is executed.	of the if statement is executed.
		In this body, the	In this body, the
		updateMainSummary() helper	updateMainSummary() helper

method is called and the first object method is called and the first object in the List is sent through as the in the List is sent through as the parameter. Then the parameter. Then the updateHourlyForecasts() helper updateHourlyForecasts() helper method is called, followed by the method is called, followed by the updateDailyForecasts() helper updateDailyForecasts() helper method. Both use the whole List as method. Both use the whole List as their parameter. their parameter. If the if condition was not met, the If the if condition was not met, the body of the else statement is body of the else statement is executed instead. A message is executed instead. A message is stored in a String object, and a Toast stored in a String object, and a Toast object created using the app's base object created using the app's base context and the message String. This context and the message String. This Toast is then shown to the user to Toast is then shown to the user to tell them they haven't got any tell them they haven't got any forecasts stored/downloaded for the forecasts stored/downloaded for the selected location. selected location. WelcomeActivity - DownloaderTask An ArravList of The super class's onPostExecute() The super class's onPostExecute() method is called using the ArrayList onPostExecute() Method ForecastLocation method is called using the ArrayList objects as the parameter. A message is also as the parameter. A message is also The purpose of this test is to check that made in the debug log. made in the debug log. the method correctly precedes the job of An ArrayList of Forecast objects is An ArrayList of Forecast objects is the task. created. created. A for loop the length of the amount A for loop the length of the amount of ForecastLocation objects is of ForecastLocation objects is

			<del>-</del>
		entered. For each object, a new	entered. For each object, a new
		Forecast object is made and its	Forecast object is made and its
		setters are used, in combination with	setters are used, in combination with
		the getters of the ForecastLocation	the getters of the ForecastLocation
		objects, to insert data into the	objects, to insert data into the
		object. Some of these are converted	object. Some of these are converted
		using the Converter helper class. The	using the Converter helper class. The
		Forecast object then has its toString	Forecast object then has its toString
		printed in the debug log. The	printed in the debug log. The
		Forecast object is then added to the	Forecast object is then added to the
		new ArrayList.	new ArrayList.
		The insertForecastsDB() helper	The insertForecastsDB() helper
		method is called using the Forecast	method is called using the Forecast
		object ArrayList as its parameter,	object ArrayList as its parameter,
		then the	then the
		getForecastsForCurrentLocation()	getForecastsForCurrentLocation()
		helper method is called.	helper method is called.
WelcomeActivity - DownloaderTask	A URL object.	The URL is stored in a singular object	The URL is stored in a singular object
doInBackground() Method		and a String to store the JSON is	and a String to store the JSON is
		initialised as null. An ArrayList of	initialised as null. An ArrayList of
The purpose of this test is to check that the method correctly downloads the		ForecastLocation objects is created.	ForecastLocation objects is created.
JSON i.e. the main point of the task.		A try/catch statement is entered.	A try/catch statement is entered.
		The JSON downloaded using the	The JSON downloaded using the
		getResponseFromHttpUrl method	getResponseFromHttpUrl method
		from the HttpDownloader class is	from the HttpDownloader class is
		stored in the downloaded String.	stored in the downloaded String.

		If an IOException occurs, this is caught in the caught statement.	If an IOException occurs, this is caught in the caught statement.
		If there was JSON successfully downloaded, it is passed to the parser helper method. If there wasn't, it isn't.	If there was JSON successfully downloaded, it is passed to the parser helper method. If there wasn't, it isn't.
		The ArrayList of ForecastLocations is returned.	The ArrayList of ForecastLocations is returned.
WelcomeActivity - DownloaderTask parseJSON() Method	A String of JSON.	An ArrayList of ForecastLocation objects is created and a try/catch statement is entered.	An ArrayList of ForecastLocation objects is created and a try/catch statement is entered.
The purpose of this test is to check that			
the method correctly parses the JSON downloaded from the doInBackground method.		The whole JSON String is stored in as JSONObject. The JSONObject called city is extracted and stored in its own object. The JSONArray called list is extract and stored in its own object.	The whole JSON String is stored in as JSONObject. The JSONObject called city is extracted and stored in its own object. The JSONArray called list is extract and stored in its own object.
		An ArrayList of JSONObjects is created. A for loop is entered for then length of the JSONArray. Each object within the JSONArray is stored as an individual JSONObject in the ArrayList.	An ArrayList of JSONObjects is created. A for loop is entered for then length of the JSONArray. Each object within the JSONArray is stored as an individual JSONObject in the ArrayList.
		The city's name and its country are extracted from the city object and stored in variables.	The city's name and its country are extracted from the city object and stored in variables.

Another for loop is entered, this time for the length of the JSONObject ArrayList. For each object, the remaining forecast details are extracted and stored in variables. A new ForecastLocation object is created using all of these variables as parameters in the constructor. This new object is added to the ArrayList.

If a JSONException occurs, the catch statement is entered.

The ArrayList of ForecastLocation objects is returned.

Another for loop is entered, this time for the length of the JSONObject ArrayList. For each object, the remaining forecast details are extracted and stored in variables. A new ForecastLocation object is created using all of these variables as parameters in the constructor. This new object is added to the ArrayList.

If a JSONException occurs, the catch statement is entered.

The ArrayList of ForecastLocation objects is returned.

PreferencesActivity onCreate() Method

The purpose of this test is to check that the onCreate method runs correctly when performing tasks that only need to happen once during the activity's life. This includes assigning values to variables, calling any helper methods, and logging any messages once commands have been successfully executed.

n/a

The super class onCreate() method is called with the savedInstanceState Bundle, and the layout of the Activity is set to activity\_preferences.xml. A debug log message is made saying "In the onCreate event handler".

The sharedPrefs field is assigned the return value from getSharedPreferences, using the sharedPrefFile String as a parameter. mEditor is assigned the return from

The super class onCreate() method is called with the savedInstanceState Bundle, and the layout of the Activity is set to activity\_preferences.xml. A debug log message is made saying "In the onCreate event handler".

The sharedPrefs field is assigned the return value from getSharedPreferences, using the sharedPrefFile String as a parameter. mEditor is assigned the return from

From a broader perspective, this is a test to make sure the activity loads correctly.

the edit() method of the sharedPrefs field.

The spinLocations variable is assigned the spinnerLocations View component from the interface. The adapter variable is assigned a new ArrayAdapter object, using the String array resource of available locations as one of the parameters. The spinLocations variable has its adapter set to adapter. The spinner has an onltemSelectedListener attached.

All the TextViews are stored in their appropriate variables. All the buttons are stored in their appropriate variables. These all have an onClickListener set.

The prepTextViews() and preButtons() helper methods are called, followed by the fetchLocations() helper method. Finally, the other helper methods, locationsButtonsVisible() locationsButtonsInvisible() are called.

the edit() method of the sharedPrefs field.

The spinLocations variable is assigned the spinnerLocations View component from the interface. The adapter variable is assigned a new ArrayAdapter object, using the String array resource of available locations as one of the parameters. The spinLocations variable has its adapter set to adapter. The spinner has an onItemSelectedListener attached.

All the TextViews are stored in their appropriate variables. All the buttons are stored in their appropriate variables. These all have an onClickListener set.

The prepTextViews() and preButtons() helper methods are called, followed by the fetchLocations() helper method. Finally, the other helper methods, locationsButtonsVisible() locationsButtonsInvisible() are called.

PreferencesActivity onStart() Method	n/a	The super class onStart() method is called and a debug log message is made saying "In the onStart event	The super class onStart() method is called and a debug log message is made saying "In the onStart event
The purpose of this test is to check that the onStart method runs correctly when		handler".	handler".
making the activity visible to the user and preparing it for interactivity.			
PreferencesActivity onPause() Method	n/a	The saveLocations() helper method is called.	The saveLocations() helper method is called.
The purpose of this test is to check that the onPause method runs correctly when indicating that this activity is not in the foreground but should not be destroyed as it is expected to be resumed soon.		The super class onPause() method is called and a debug log message is made saying "In the onPause event handler".	The super class onPause() method is called and a debug log message is made saying "In the onPause event handler".
PreferencesActivity onStop() Method	n/a	The saveLocations() helper method is called.	The saveLocations() helper method is called.
The purpose of this test is to check that the onStop method runs correctly when a new activity has covered the screen, or when this activity has been terminated.		The super class onStop() method is called and a debug log message is made saying "In the onStop event handler".	The super class onStop() method is called and a debug log message is made saying "In the onStop event handler".
PreferencesActivity onDestroy() Method	n/a	The super class onDestroy() method is called and a debug log message is made saying "In the onDestroy event	The super class onDestroy() method is called and a debug log message is made saying "In the onDestroy event
The purpose of this test is to check that the onDestroy method runs correctly		handler".	handler".

when the user has completely finished using the activity.			
PreferencesActivity onResume() Method	n/a	The super class onResume() method is called and a debug log message is made saying "In the onResume event	The super class onResume() method is called and a debug log message is made saying "In the onResume event
The purpose of this test is to check that		handler".	handler".
the onResume method runs correctly when the user returns to this activity			
from another or from outside of the app.			
PreferencesActivity	No button on the	Nothing happens as no methods are	Nothing happens as no methods are
onClick() Method	interface is pressed.	called after the interface is displayed.	called after the interface is displayed.
The purpose of this test is to check that			
the onClick method does not run unless a button is pressed.			
PreferencesActivity	The button to add	The onClick method/event handler is	The onClick method/event handler is
onClick() Method	the currently	called, sending the	called, sending the
	selected location	buttonPreferences view as a	buttonPreferences view as a
The purpose of this test is to check that	to the user's list.	parameter. A Toast and String	parameter. A Toast and String
the onClick method processes the clicking of each button correctly.		message are declared, ready for use.	message are declared, ready for use.
		The if statement is entered. The	The if statement is entered. The
		condition should match the initial	condition should match the initial
		condition (i.e. the id of the view is buttonAdd) so the body of the if	condition (i.e. the id of the view is buttonAdd) so the body of the if
		condition is executed.	condition is executed.
		If the user does not have less than 10	If the user does not have less than 10
		locations, the message is set and the	locations, the message is set and the
		Toast is displayed to tell them they	Toast is displayed to tell them they

		can't have any more locations until	can't have any more locations until
		they delete one.	they delete one.
		If they do have less than 10	If they do have less than 10
		locations, the list is checked to see if	locations, the list is checked to see if
		the selected location is already	the selected location is already
		included. If it is, the user is told they	included. If it is, the user is told they
		cannot add a duplicate. If it isn't, the	cannot add a duplicate. If it isn't, the
		location is added to the next	location is added to the next
		available spot in the ArrayList	available spot in the ArrayList
		(replacing a null value).	(replacing a null value).
			, ,
		The number of locations counter is	The number of locations counter is
		increased. The	increased. The
		locationsButtonsVisible() and	locationsButtonsVisible() and
		locationsButtonsInvisible() helper	locationsButtonsInvisible() helper
		methods are called. The message is	methods are called. The message is
		set and the Toast displayed to tell	set and the Toast displayed to tell
		the user they've added the location	the user they've added the location
		to their list.	to their list.
PreferencesActivity	The button to	The number of locations counter is	The number of locations counter is
onClick() Method	remove the first	checked. If it is equal to 1, the user is	checked. If it is equal to 1, the user is
	item in the user's	told via a Toast they cannot delete	told via a Toast they cannot delete
The purpose of this test is to check that	list.	this location until they have added	this location until they have added
the onClick method processes the		another.	another.
clicking of each button correctly.			
		If they have more than 1, the	If they have more than 1, the
		location is removed from the	location is removed from the
		ArrayList and the	ArrayList and the
		locationsButtonsVisible() and	locationsButtonsVisible() and

		locationsButtonsInvisible() helper methods are called.	locationsButtonsInvisible() helper methods are called.
PreferencesActivity	The button to any	The location is removed from the	The location is removed from the
onClick() Method	other item in the user's list.	ArrayList, the number of locations counter is decremented, and the	ArrayList, the number of locations counter is decremented, and the
The purpose of this test is to check that		visible and invisible helper methods	visible and invisible helper methods
the onClick method processes the		are called.	are called.
clicking of each button correctly.			
PreferencesActivity	An item is	The item at the selected position in	The item at the selected position in
onItemSelected() Method	selected from the spinner.	the spinner is converted to a String and stored in the currentLocation	the spinner is converted to a String and stored in the currentLocation
The purpose of this test is to check that		variable.	variable.
the onItemSelected method correctly			
processes the selection of any item in			
the spinner.			
PreferencesActivity	No item has been	Nothing happens as this method has	Nothing happens as this method has
onNothingSelected() Method	newly selected (it	no body.	no body.
	is not possible to		
The purpose of this test is to check that	have no item		
the onNothingSelected method correctly	selected at all).		
processes the lack of selection of any			
item in the spinner.			
PreferencesActivity	n/a	The sharedPrefs variable is initialised	The sharedPrefs variable is initialised
updateLocations() Method		using the getSharedPreferences()	using the getSharedPreferences()
		method and the sharedPrefFile	method and the sharedPrefFile
The purpose of this test is to check that		parameter. A counter variable is	parameter. A counter variable is
the user's chosen locations are retrieved		initialised at 0.	initialised at 0.
from the shared preferences.			
		The LOCATION_X Strings in the	The LOCATION_X Strings in the
		shared preferences are iterated	shared preferences are iterated

		The same to the state of the st	the state of the Contract of the
		through with a for loop, storing even the nulls in an ArrayList.	through with a for loop, storing even the nulls in an ArrayList.
		This ArrayList is iterated through and	This ArrayList is iterated through and
		for every value that isn't a null, the	for every value that isn't a null, the
		counter is incremented.	counter is incremented.
PreferencesActivity	n/a	All the 10 location text views are	All the 10 location text views are
prepTextViews() Method		stored in an ArrayList.	stored in an ArrayList.
The purpose of this test is to check that			
the text views are all stored properly.			
PreferencesActivity	n/a	All the 10 location removal buttons	All the 10 location removal buttons
prepButtons() Method		are stored in an ArrayList.	are stored in an ArrayList.
The purpose of this test is to check that the buttons are all stored properly.			
• • • • • • • • • • • • • • • • • • • •	n /n	The Associate of hosterno and took	The Agreed iste of buttons and tout
PreferencesActivity locationsButtonsVisible() Method	n/a	The ArrayLists of buttons and text views are iterated through, both for	The ArrayLists of buttons and text
locations Buttons visible() Method		the length of number of locations	views are iterated through, both for the length of number of locations
The purpose of this test is to check that		counter. The visibility to all of these	counter. The visibility to all of these
the buttons and text views that should		is set to visible. The text for each text	is set to visible. The text for each text
be visible to the user are made visible.		view is set to the corresponding	view is set to the corresponding
		value in the locationsList ArrayList.	value in the locationsList ArrayList.
PreferencesActivity	n/a	The ArrayLists of buttons and text	The ArrayLists of buttons and text
locationsButtonsInvisible() Method		views are iterated through, both for	views are iterated through, both for
		the length of the difference between	the length of the difference between
The purpose of this test is to check that		the length of the ArrayLists and the	the length of the ArrayLists and the
the buttons and text views that should		number of locations counter. The	number of locations counter. The
be visible to the user are made invisible.		visibility to all of these is set to	visibility to all of these is set to
		invisible.	invisible.

PreferencesActivity saveLocations() Method  The purpose of this test is to check that the user's chosen locations are saved in	n/a	The sharedPrefs variable is initialised with the getSharedPreferences() method and sharedPrefFile variable as the parameter. The mEditor is also initialised as the return from the	The sharedPrefs variable is initialised with the getSharedPreferences() method and sharedPrefFile variable as the parameter. The mEditor is also initialised as the return from the
the shared preferences.		edit() method of the sharedPrefs object. The editor is used to clear any existing sharedPreferences.	edit() method of the sharedPrefs object. The editor is used to clear any existing sharedPreferences.
		A for loop is entered for the length of the number of locations counter. The locationsList ArrayList is iterated through and for each item in the ArrayList, it is added to the shared preferences in a String called LOCATION_X (where X is the counter/ArrayList position + 1).	A for loop is entered for the length of the number of locations counter. The locationsList ArrayList is iterated through and for each item in the ArrayList, it is added to the shared preferences in a String called LOCATION_X (where X is the counter/ArrayList position + 1).
		The mEditor then applies these changes.	The mEditor then applies these changes.
SearchActivity onCreate() Method	n/a	The super class onCreate() method is called with the savedInstanceState Bundle, and the layout of the Activity	
The purpose of this test is to check that the onCreate method runs correctly when performing tasks that only need to happen once during the activity's life. This includes assigning values to		is set to activity_search.xml. A debug log message is made saying "In the onCreate event handler".	

variables, calling any helper methods, and logging any messages once commands have been successfully executed.

From a broader perspective, this is a test to make sure the activity loads correctly.

The forecastDB field is assigned a value – the return value from the getDatabase() method in the ForecastDatabase class. The forecastDAO field is assigned the return of the forecastDao() method from the forecastDB field. A log debug message is made saying "Initialised database okay".

The sharedPrefs field is assigned the return value from getSharedPreferences, using the sharedPrefFile String as a parameter. mEditor is assigned the return from the edit() method of the sharedPrefs field.

The spinner variables are all assigned their corresponding spinner View components. Each also has a unique adapter made for it, using either a String array resource, or an ArrayList. The spinners all have an onItemSelectedListener attached.

All the Buttons are stored in their appropriate variables These all have an on Click Listener set.

SearchActivity onStart() Method  The purpose of this test is to check that the onStart method runs correctly when making the activity visible to the user and preparing it for interactivity.	n/a	The super class onStart() method is called and a debug log message is made saying "In the onStart event handler".	
SearchActivity onPause() Method  The purpose of this test is to check that the onPause method runs correctly when indicating that this activity is not in the foreground but should not be destroyed as it is expected to be resumed soon.	n/a	The super class onPause() method is called and a debug log message is made saying "In the onPause event handler".	
SearchActivity onStop() Method  The purpose of this test is to check that the onStop method runs correctly when a new activity has covered the screen, or when this activity has been terminated.	n/a	The super class onStop() method is called and a debug log message is made saying "In the onStop event handler".	
SearchActivity onDestroy() Method  The purpose of this test is to check that the onDestroy method runs correctly when the user has completely finished using the activity.	n/a	The super class onDestroy() method is called and a debug log message is made saying "In the onDestroy event handler".	

SearchActivity onResume() Method	n/a	The super class onResume() method is called and a debug log message is	
The purpose of this test is to check that the onResume method runs correctly when the user returns to this activity		made saying "In the onResume event handler".	
from another or from outside of the app.			
SearchActivity onClick() Method	No button on the interface is pressed.	Nothing happens as no methods are called after the interface is displayed.	Nothing happens as no methods are called after the interface is displayed.
The purpose of this test is to check that the onClick method does not run unless a button is pressed.		. ,	. ,
SearchActivity	The first button is	The onClick method/event handler is	The onClick method/event handler is
onClick() Method	clicked.	called, sending the buttonSearch1	called, sending the buttonSearch1
		view as a parameter.	view as a parameter.
The purpose of this test is to check that			
the onClick method processes the		A new context object is created and	A new context object is created and
clicking of each button correctly.		assigned the value of the	assigned the value of the
		SearchActivity context. A debug log	SearchActivity context. A debug log
		message is made saying "In the	message is made saying "In the
		onClick method".	onClick method".
		The if statement is entered. A Class	The if statement is entered. A Class
		object is created to store the	object is created to store the
		SearchResultsByHourActivity.class	SearchResultsByHourActivity.class
		object. An Intent object is created	object. An Intent object is created
		using the context object and the	using the context object and the
		destination class object. The	destination class object. The
		startActivity() method is called using	startActivity() method is called using

		the newly created intent as the	the newly created intent as the
		parameter, launching the search	parameter, launching the search
		results screen.	results screen.
SearchActivity	The second	The onClick method/event handler is	The onClick method/event handler is
onClick() Method	button is clicked.	called, sending the buttonSearch2	called, sending the buttonSearch2
		view as a parameter.	view as a parameter.
The purpose of this test is to check that			
the onClick method processes the		A new context object is created and	A new context object is created and
clicking of each button correctly.		assigned the value of the	assigned the value of the
		SearchActivity context. A debug log	SearchActivity context. A debug log
		message is made saying "In the	message is made saying "In the
		onClick method".	onClick method".
		The if statement is entered. A Class	The if statement is entered. A Class
		object is created to store the	object is created to store the
		SearchResultsValueActivity.class	SearchResultsValueActivity.class
		object. An Intent object is created	object. An Intent object is created
		using the context object and the	using the context object and the
		destination class object. The	destination class object. The
		startActivity() method is called using	startActivity() method is called using
		the newly created intent as the	the newly created intent as the
		parameter, launching the search	parameter, launching the search
		results screen.	results screen.
SearchActivity	The third button	The onClick method/event handler is	The onClick method/event handler is
onClick() Method	is clicked.	called, sending the buttonDeleteAll	called, sending the buttonDeleteAll
		view as a parameter.	view as a parameter.
The purpose of this test is to check that			
the onClick method processes the		The deleteForecastsDB() helper	The deleteForecastsDB() helper
clicking of each button correctly.		method is called.	method is called.

		A String message is declared and initialised, shown in a Toast object to tell the user their Forecasts have been deleted.	A String message is declared and initialised, shown in a Toast object to tell the user their Forecasts have been deleted.
SearchActivity onItemSelected() Method  The purpose of this test is to check that the onItemSelected method correctly processes the selection of any item in the spinner.	An item is selected from the spinner.	The item from the spinner which has just been selected is stored in the corresponding global variable. Some of these are converted before doing so.	The item from the spinner which has just been selected is stored in the corresponding global variable. Some of these are converted before doing so.
SearchActivity onNothingSelected() Method  The purpose of this test is to check that the onNothingSelected method correctly processes the lack of selection of any item in the spinner.	No item has been newly selected (it is not possible to have no item selected at all).	Nothing happens as this method has no body.	Nothing happens as this method has no body.
SearchActivity fetchLocations() Method  The purpose of this test is to check that the user's chosen locations are retrieved from the shared preferences.	n/a	The sharedPrefs variable is initialised using the getSharedPreferences() method and the sharedPrefFile parameter. A debug log message is made to confirm this has happened.  A counter variable is initialised. A while loop is entered. Which goes through all the LOCATION_X Strings in the shared preferences until one returns a null.	The sharedPrefs variable is initialised using the getSharedPreferences() method and the sharedPrefFile parameter. A debug log message is made to confirm this has happened.  A counter variable is initialised. A while loop is entered. Which goes through all the LOCATION_X Strings in the shared preferences until one returns a null.

			1
SearchActivity saveSearchItems() Method  The purpose of this test is to check that the user's chosen search terms are saved in the shared preferences.	An id to tell the method which button has been pressed.	If a String does not return a null, it is logged with a debug message before it is added to the chosenLocations ArrayList.  The sharedPrefs variable is initialised using the getSharedPreferences() method and the sharedPrefFile parameter. The editor is also set up.  An if statement is used to determine which sets of values are being put into the shared preferences. Each of	If a String does not return a null, it is logged with a debug message before it is added to the chosenLocations ArrayList.  The sharedPrefs variable is initialised using the getSharedPreferences() method and the sharedPrefFile parameter. The editor is also set up.  An if statement is used to determine which sets of values are being put into the shared preferences. Each of
		into the shared preferences. Each of these are put into Strings with appropriate labels and the editor applies these changes.	into the shared preferences. Each of these are put into Strings with appropriate labels and the editor applies these changes.
SearchActivity deleteForecastsDB() Method	n/a	A new DeleteForecastsTask object is created and told to execute.	A new DeleteForecastsTask object is created and told to execute.
The purpose of this test is to check that the task is executed.			
SearchActivity - DeleteForecastsTask doInBackground() Method	n/a	The forecastDAO object is used to call its deleteAllForecasts() method.	The forecastDAO object is used to call its deleteAllForecasts() method.
The purpose of this test is to check that the method interacts with the DAO to delete all forecasts from the database for a specific location			

SearchActivity - DeleteForecastsTask onPostExecute() Method	A Void object.	The super class's onPostExecute method is called using the Void object as the parameter.	The super class's onPostExecute method is called using the Void object as the parameter.
The purpose of this test is to check that the method correctly finishes the job of the task.			
ForecastDatabase	n/a		
As this code is the same as that used in the labs, minus the modifications of the names for classes etc, this class was not individually tested, but observed during its interactions with other classes/methods. The class performed as expected and operated perfectly.			
Forecast	n/a		
As this code is simply getters and setters generated by Android Studio, this class was not tested individually, but observed during its interactions with other classes/methods. The class performed as expected and operated perfectly.			

ForecastLocation	n/a		
As this code is simply getters and setters			
generated by Android Studio, this class			
was not tested individually, but observed			
during its interactions with other			
classes/methods. The class performed as			
expected and operated perfectly.			
	T .		
HttpDownloader	n/a		
As this code is the same as that used in			
the labs, this class was not individually tested, but observed during its			
interactions with other classes/methods.			
The class performed as expected and			
operated perfectly.			
operated perfectly.			
Converter	A String	The String is checked for the	The String is checked for the
weatherDesc() Method	containing the	character at the start being	character at the start being
	original	converted to uppercase. Then, any	converted to uppercase. Then, any
The purpose of this test is to check that	description, all in	additional characters which follow a	additional characters which follow a
the weather description is correctly	lowercase.	space are also converted to	space are also converted to
formatted.		uppercase. The converted String is	uppercase. The converted String is
		returned.	returned.
Converter	A double	The double is put through an if	The double is put through an if
windDirection() Method	containing the	statement to determine which	statement to determine which
	original value for	description should be returned.	description should be returned.

			1
The purpose of this test is to check that the wind direction is accurately	the wind direction.	The values used were calculated by	
converted to a description.	direction.	hand.	
Converter windSpeed() Method	A double containing the original value for	The double is put through an if statement to determine which description should be returned.	The double is put through an if statement to determine which description should be returned.
The purpose of this test is to check that	the wind	The estate of the section of the sec	
the wind speed is accurately converted to a description.	direction.	The categories used were from the Beaufort scale as described on Wikipedia.	
Converter time() Method	A String containing the date and time in	The characters 11-15 (HH:MM) are extracts from the String as a substring and returned.	The characters 11-15 (HH:MM) are extracts from the String as a substring and returned.
The purpose of this test is to check that the time of the forecast is correctly extracted from the input.	the format YYYY- MM-DD HH:MM:SS		
Converter date() Method	An ArrayList containing the date in parts.	The parts are concatenated together as a single String and returned in the format DDMMYYYY.	The parts are concatenated together as a single String and returned in the format DDMMYYYY.
The purpose of this test is to check that the date of the forecast is correctly extracted from the input			
Converter date() Method	A String containing the date in the	The date parts are extracted into different substrings.	The date parts are extracted into different substrings.
The purpose of this test is to check that the date of the forecast is correctly converted to a different format.	format DDMMYYYY.	The day of the week is found by calling the findDayOfWeek() method.	The day of the week is found by calling the findDayOfWeek() method.
		2 case statements and the day of the week are concatenated together to	2 case statements and the day of the week are concatenated together to

		form the returned String in the format e.g. Monday 1st December.	form the returned String in the format e.g. Monday 1 <sup>st</sup> December.
Converter	A String	An ArrayList is returned with the day,	An ArrayList is returned with the day,
dateParts() Method	containing the	month, and year all as separate	month, and year all as separate
	date in the	Strings within the ArrayList.	Strings within the ArrayList.
The purpose of this test is to check that	format YYYY-MM-		
the date of the forecast is correctly	DD.		
separated into different parts within an			
ArrayList.			
Converter	3 Strings: the day,	The Key/Value method is utilised	The Key/Value method is utilised
findDayOfWeek() Method	the month and	through a series of switch and if	through a series of switch and if
	the year.	statements to return a day of the	statements to return a day of the
The purpose of this test is to check that		week.	week.
the correct day of the week is found for	OR		
date of the forecast using the			
Key/Vaulue method from mathforum.org	A String with the		
	date in the		
	format		
	DDMMYYYY.		
ForecastDao	An array of	The Room @Insert tag will deal with	The Room @Insert tag will deal with
insertForecasts() Method	Forecast objects.	the implementation of the method, but all Forecast objects within the	the implementation of the method, but all Forecast objects within the
The purpose of this test is to check that		given array are added to the	given array are added to the
the array of Forecast objects is inserted		database.	database.
correctly into the database.		database.	database.
ForecastDao	A String with the	The Room @Query tag will deal with	The Room @Query tag will deal with
getAllForecastsForLocation() Method	city name.	the implementation of the method,	the implementation of the method,
Sec. in orceases or Location() Method	city name.	but the query itself is correct	but the query itself is correct
	1	Sat the query resem is connect	but the query resem is connect

			1
The purpose of this test is to check that		meaning that all Forecast objects	meaning that all Forecast objects
the return list of Forecast objects is		returned are those relating to the	returned are those relating to the
correct		given location.	given location.
ForecastDao	n/a	The Room @Query tag will deal with	The Room @Query tag will deal with
deleteAllForecasts() Method		the implementation of the method,	the implementation of the method,
		but the query itself is correct	but the query itself is correct
The purpose of this test is to check that		meaning that all Forecast objects are	meaning that all Forecast objects are
all Forecasts are deleted.		deleted.	deleted.
ForecastDao	A String with the	The Room @Query tag will deal with	The Room @Query tag will deal with
deleteAllForecastsFor() Method	city name.	the implementation of the method,	the implementation of the method,
·		but the query itself is correct	but the query itself is correct
The purpose of this test is to check that		meaning that all Forecast objects	meaning that all Forecast objects
the Forecasts relating to the given		related to the given location are	related to the given location are
location are all deleted.		deleted.	deleted.
ForecastDao	A String with the	The Room @Query tag will deal with	The Room @Query tag will deal with
getAllForecastsHourly() Method	city name and an	the implementation of the method,	the implementation of the method,
	int specifying	but the query itself is correct	but the query itself is correct
The purpose of this test is to check that	how many rows	meaning that all Forecast objects	meaning that all Forecast objects
the Forecasts relating to the given	to return.	related to the given location in the	related to the given location in the
location within the specified time range		time range are returned.	time range are returned.
are returned.		_	
ForecastDao	A String with the	The Room @Query tag will deal with	The Room @Query tag will deal with
getAllForecastsGreaterValue() Method	city name and an	the implementation of the method,	the implementation of the method,
	int specifying	but all Forecast objects related to	but the query itself is incorrect
The purpose of this test is to check that	how many rows	the given location with the search	somewhere as ALL Forecast objects
the Forecasts relating to the given	to return.	term having a value greater than or	for the specified location are
location with the search term having a		equal to the specified value are	returned.
value greater than or equal to the		returned.	
specified value are returned.			

ForecastDao getAllForecastsLessValue() Method  The purpose of this test is to check that the Forecasts relating to the given location with the search term having a value less than or equal to the specified value are returned.	A String with the city name and an int specifying how many rows to return.	The Room @Query tag will deal with the implementation of the method, but all Forecast objects related to the given location with the search term having a value less than or equal to the specified value are returned.	The Room @Query tag will deal with the implementation of the method, but the query itself is incorrect somewhere as ALL Forecast objects for the specified location are returned.
SearchResultsByHourActivity onCreate() Method	n/a	The super class onCreate() method is called with the savedInstanceState Bundle, and the layout of the Activity	The super class onCreate() method is called with the savedInstanceState Bundle, and the layout of the Activity
The purpose of this test is to check that		is set to	is set to
the onCreate method runs correctly		activity_search_results_by_hour.xml.	activity_search_results_by_hour.xml.
when performing tasks that only need to		A debug log message is made saying	A debug log message is made saying
happen once during the activity's life. This includes assigning values to		"In the onCreate event handler".	"In the onCreate event handler".
variables, calling any helper methods,		The forecastDB field is assigned a	The forecastDB field is assigned a
and logging any messages once		value – the return value from the	value – the return value from the
commands have been successfully		getDatabase() method in the	getDatabase() method in the
executed.		ForecastDatabase class. The	ForecastDatabase class. The
		forecastDAO field is assigned the	forecastDAO field is assigned the
From a broader perspective, this is a test		return of the forecastDao() method	return of the forecastDao() method
to make sure the activity loads correctly.		from the forecastDB field. A log	from the forecastDB field. A log
		debug message is made saying	debug message is made saying
		"Initialised database okay".	"Initialised database okay".
		The sharedPrefs field is assigned the return value from	The sharedPrefs field is assigned the return value from

SearchResultsByHourActivity	n/a	getSharedPreferences, using the sharedPrefFile String as a parameter. mEditor is assigned the return from the edit() method of the sharedPrefs field.  The fetchParams() helper method is called. The search results text view is stored in a variable and then the searchForecastsDB() helper method is called. The super class onStart() method is	getSharedPreferences, using the sharedPrefFile String as a parameter. mEditor is assigned the return from the edit() method of the sharedPrefs field.  The fetchParams() helper method is called.  The search results text view is stored in a variable and then the searchForecastsDB() helper method is called.  The super class onStart() method is
onStart() Method  The purpose of this test is to check that the onStart method runs correctly when making the activity visible to the user and preparing it for interactivity.	.,, 3	called and a debug log message is made saying "In the onStart event handler".	called and a debug log message is made saying "In the onStart event handler".
SearchResultsByHourActivity onPause() Method  The purpose of this test is to check that the onPause method runs correctly when indicating that this activity is not in the foreground but should not be destroyed as it is expected to be resumed soon.	n/a	The super class onPause() method is called and a debug log message is made saying "In the onPause event handler".	The super class onPause() method is called and a debug log message is made saying "In the onPause event handler".
SearchResultsByHourActivity onStop() Method	n/a	The super class onStop() method is called and a debug log message is	The super class onStop() method is called and a debug log message is

The purpose of this test is to check that the onStop method runs correctly when a new activity has covered the screen, or when this activity has been terminated.		made saying "In the onStop event handler".	made saying "In the onStop event handler".
SearchResultsByHourActivity onDestroy() Method  The purpose of this test is to check that the onDestroy method runs correctly when the user has completely finished using the activity.	n/a	The super class onDestroy() method is called and a debug log message is made saying "In the onDestroy event handler".	The super class onDestroy() method is called and a debug log message is made saying "In the onDestroy event handler".
SearchResultsByHourActivity onResume() Method  The purpose of this test is to check that the onResume method runs correctly when the user returns to this activity from another or from outside of the app.	n/a	The super class onResume() method is called and a debug log message is made saying "In the onResume event handler".	The super class onResume() method is called and a debug log message is made saying "In the onResume event handler".
SearchResultsByHourActivity fetchParams() Method  The purpose of this test is to check that the search terms are retrieved correctly from the shared preferences.	n/a	The sharedPrefs variable is initialised using the getSharedPreferences() method and the sharedPrefFile parameter. A debug log message is made to confirm this has happened.  The current location and the amount of hours to search for are retrieved from the shared preferences.	The sharedPrefs variable is initialised using the getSharedPreferences() method and the sharedPrefFile parameter. A debug log message is made to confirm this has happened.  The current location and the amount of hours to search for are retrieved from the shared preferences.
SearchResultsByHourActivity searchForecastsDB() Method	n/a	A new SearchForecastsTask object is created and told to execute.	A new SearchForecastsTask object is created and told to execute.

The purpose of this test is to check that the search task is made and told to execute.			
SearchResultsByHourActivity - SearchForecastsTask doInBackground() Method	n/a	The forecastDAO object is used to call its getAllForecastsHourly() method.	The forecastDAO object is used to call its getAllForecastsHourly () method.
The purpose of this test is to check that the method interacts with the DAO to find all forecasts for a specification to show the temperature within a certain time frame.			
SearchResultsByHourActivity - SearchForecastsTask onPostExecute() Method	A List of Forecast objects.	The super class's onPostExecute method is called using the Void object as the parameter.	The super class's onPostExecute method is called using the Void object as the parameter.
The purpose of this test is to check that the method correctly finishes the job of the task.		If the array size is 0, the user is told no results could be found.	If the array size is 0, the user is told no results could be found.
		If the array size is greater than 0, the results are displayed in the textbox.	If the array size is greater than 0, the results are displayed in the textbox.
SearchResultsValueActivity onCreate() Method	n/a	The super class onCreate() method is called with the savedInstanceState Bundle, and the layout of the Activity	The super class onCreate() method is called with the savedInstanceState Bundle, and the layout of the Activity
The purpose of this test is to check that the onCreate method runs correctly when performing tasks that only need to		is set to activity_search_results_value.xml. A	is set to activity_search_result_value.xml. A

happen once during the activity's life.		debug log message is made saying	debug log message is made saying
This includes assigning values to		"In the onCreate event handler".	"In the onCreate event handler".
		in the officieate event flandler.	in the officieate event flandler.
variables, calling any helper methods,		The forecast DD field is assigned a	The forecast DD field is assigned a
and logging any messages once		The forecastDB field is assigned a	The forecastDB field is assigned a
commands have been successfully		value – the return value from the	value – the return value from the
executed.		getDatabase() method in the	getDatabase() method in the
		ForecastDatabase class. The	ForecastDatabase class. The
From a broader perspective, this is a test		forecastDAO field is assigned the	forecastDAO field is assigned the
to make sure the activity loads correctly.		return of the forecastDao() method	return of the forecastDao() method
		from the forecastDB field. A log	from the forecastDB field. A log
		debug message is made saying	debug message is made saying
		"Initialised database okay".	"Initialised database okay".
		The sharedPrefs field is assigned the	The sharedPrefs field is assigned the
		return value from	return value from
		getSharedPreferences, using the	getSharedPreferences, using the
		sharedPrefFile String as a parameter.	sharedPrefFile String as a parameter.
		mEditor is assigned the return from	mEditor is assigned the return from
		the edit() method of the sharedPrefs	the edit() method of the sharedPrefs
		field.	field.
		The fetchParams() helper method is	The fetchParams() helper method is
		called.	called.
		The search results text view is stored	The search results text view is stored
		in a variable and then the	in a variable and then the
		searchForecastsDB() helper method	searchForecastsDB() helper method
		is called.	is called.
SearchResultsByHourActivity	n/a	The super class onStart() method is	The super class onStart() method is
onStart() Method		called and a debug log message is	called and a debug log message is

The purpose of this test is to check that the onStart method runs correctly when making the activity visible to the user and preparing it for interactivity.		made saying "In the onStart event handler".	made saying "In the onStart event handler".
SearchResultsByHourActivity onPause() Method	n/a	The super class onPause() method is called and a debug log message is made saying "In the onPause event	The super class onPause() method is called and a debug log message is made saying "In the onPause event
The purpose of this test is to check that the onPause method runs correctly when indicating that this activity is not in		handler".	handler".
the foreground but should not be destroyed as it is expected to be resumed soon.			
SearchResultsByHourActivity onStop() Method	n/a	The super class onStop() method is called and a debug log message is made saying "In the onStop event	The super class onStop() method is called and a debug log message is made saying "In the onStop event
The purpose of this test is to check that the onStop method runs correctly when a new activity has covered the screen, or when this activity has been terminated.		handler".	handler".
SearchResultsByHourActivity onDestroy() Method	n/a	The super class onDestroy() method is called and a debug log message is made saying "In the onDestroy event	The super class onDestroy() method is called and a debug log message is made saying "In the onDestroy even
The purpose of this test is to check that the onDestroy method runs correctly when the user has completely finished using the activity.		handler".	handler".
SearchResultsByHourActivity onResume() Method	n/a	The super class onResume() method is called and a debug log message is	The super class onResume() method is called and a debug log message is
	1		1

			1
The purpose of this test is to check that the onResume method runs correctly when the user returns to this activity from another or from outside of the app.		made saying "In the onResume event handler".	made saying "In the onResume event handler".
• • • • • • • • • • • • • • • • • • • •	,		
SearchResultsByHourActivity fetchParams() Method	n/a	The sharedPrefs variable is initialised using the getSharedPreferences() method and the sharedPrefFile	The sharedPrefs variable is initialised using the getSharedPreferences() method and the sharedPrefFile
The purpose of this test is to check that		parameter. A debug log message is	parameter. A debug log message is
the search terms are retrieved correctly from the shared preferences.		made to confirm this has happened.	made to confirm this has happened.
, and the second		The current location, the search term, the search value, and whether the user wants the results greater than/equal to are retrieved from shared preferences.	The current location, the search term, the search value, and whether the user wants the results greater than/equal to are retrieved from shared preferences.
SearchResultsByHourActivity searchForecastsDB() Method  The purpose of this test is to check that the search task is made and told to execute.	n/a	Depending on the searchTerm value, an if/else if/else statement will be used to guide which task object is created. Nested if statements help further this guidance based on whether the user wants greater/less than values.	Depending on the searchTerm value, an if/else if/else statement will be used to guide which task object is created. Nested if statements help further this guidance based on whether the user wants greater/less than values.
		The chosen task object is then created and told to execute.	The chosen task object is then created and told to execute.
SearchResultsByHourActivity - SearchGreaterTemperatureForecastsTask SearchGreaterHumidityForecastsTask SearchGreaterWindSpeedForecastsTask	n/a	The forecastDAO object is used to call its getAllForecastsGreaterValue() method.	The forecastDAO object is used to call its getAllForecastsGreaterValue() method.

doInBackground() Method  The purpose of this test is to check that the method interacts with the DAO to find all forecasts matching the user's specification.			
SearchResultsByHourActivity - SearchGreaterTemperatureForecastsTask SearchGreaterHumidityForecastsTask SearchGreaterWindSpeedForecastsTask onPostExecute() Method The purpose of this test is to check that the method correctly finishes the job of the task.	A List of Forecast objects.	The super class's onPostExecute method is called using the Void object as the parameter.  If the array size is 0, the user is told no results could be found.  If the array size is greater than 0, the results are displayed in the textbox.	The super class's onPostExecute method is called using the Void object as the parameter.  If the array size is 0, the user is told no results could be found.  If the array size is greater than 0, the results are displayed in the textbox, but these results are not accurate as they are just for EVERY forecast for the chosen location.
SearchResultsByHourActivity - SearchLessTemperatureForecastsTask SearchLessHumidityForecastsTask SearchLessWindSpeedForecastsTask doInBackground() Method  The purpose of this test is to check that the method interacts with the DAO to find all forecasts matching the user's specification.	n/a	The forecastDAO object is used to call its getAllForecastsLessValue() method.	The forecastDAO object is used to call its getAllForecastsLessValue () method.

SearchResultsByHourActivity	A List of Forecast	The super class's onPostExecute	The super class's onPostExecute
- SearchLessTemperatureForecastsTask	objects.	method is called using the Void	method is called using the Void
SearchLessHumidityForecastsTask		object as the parameter.	object as the parameter.
SearchLessWindSpeedForecastsTask			
onPostExecute() Method		If the array size is 0, the user is told no results could be found.	If the array size is 0, the user is told no results could be found.
The purpose of this test is to check that the method correctly finishes the job of the task.		If the array size is greater than 0, the results are displayed in the textbox.	If the array size is greater than 0, the results are displayed in the textbox, but these results are not accurate as they are just for EVERY forecast for the chosen location.