RMIT UNIVERSITY

EEET2578 - Engineering Quality Assurance and Testing

EnviroSmart Application Quality Testing

Report Submission - Stage 2

Lecturer: Dr. Minh Dinh

Group 4

Tran Dam Quan (s3678708)
Ho Minh Duc (s3694653)
Nguyen Nguyen Ha Nhan (s3687637)
Le Nguyen (s3777242)
Luu Huynh Tri (s3462315)

Submission Date: 21/05/2021

Table of Contents

List of Figures

List of Tables

1. Introduction

As a smartwatch-integrated software application, EnviroSmart makes use of various built-in sensed environmental and weather devices which helps to record several information about weather condition, temperature, air quality as well as location coordinates. Under any abnormal environmental conditions, the application proceeds to send appropriate warnings to the user based on the context information it received from the sensors and suggest suitable venues of interest with reference to user's specified personal choices. These specifications is built up to the demand of users by which they can be helped to avoid the potential health risks regarding to the air pollution overexposure as well as intense temperature and weather outdoor.

Having the stage 1 report and lecturer's feedback reviewed, we have come up with a better version for this stage 2 report, specifically with the methodologies for every testing type mentioned in the test execution plan section. Apart from that, we are going to execute the list of test cases from integration, system and user acceptance testing which we manually generated in stage 1. Furthermore, we also perform the automatic unit test execution in this second stage in reference to what we are provided from the source code folder eee2578 using JUnit framework for unit testing in Java programming language. The comprehensive test strategies, assumptions and constraints are explained in the section of test execution plan. Corresponding to every test case executed in all 4 testing types, we provides the details of test data, expected and actual outputs as well as finalize the test results whether it was PASSED or FAILED. In addition, the report delineates the organization of group workloads between team personnels and discuss relevant issues that may negatively affect the results of the test plan, as well as suitable solutions to prevent undesirable scenarios. A Gantt chart for viewing project timeline, a resource allocation, including human resources and other external resources are also re-introduced in this stage 2 report.

2. Test Cases Execution Plan

2.1. Execution Objectives

This section defines the scope of the project and the means to achieve them. The goal of the test scripts is to help verifying and validating the existing features and capabilities of the EnviroSmart application to confirm whether it meets the preliminary software requirements as a fully integrated and operative product upon public release. In addition, testing effort and cost proficiency, as well as risks and plausible issues are also thoroughly identified and managed to contribute to the success of the test plan for this project.

2.2. System Software Overview

There are two source roots named ***"generated"*** and ***"src"*** placed inside the source code folder eeet2578. In the first source root ***"generated"***, specifically inside the only package helper, it consists of multiple classes as well as interfaces which help to retrieve data from the ice server. In addition, these take action as a communicator between ice server as the message subscriber and our EnviroSmart system as the message publisher. Outside, the root folder contains several predefined text files needed for the class or module to read for their respective functions and variables. These store all the information about the system users, their preferences and locations.

In order to successfully run the application system, we must activate the ice server through the command prompt if using Windows or the terminal if using MacOS with the following command line inside the bin directory of the source code project:

icebox --Ice.Config=config.icebox

Afterwards, we need to run the components of the application in the right order which is specified in the assessment details:

WeatherAlarms.java → LocationServer.java → PreferenceRepository.java → ContextManager.java → EnviroAPPUI.java → AllSensor.java

In the other source root ***"src"***, the EnviroSmart system contains two major packages which are main and support. Inside of these, it divides into different numbers of classes and modules taking responsibility to the entire system.

2.2.1. Main Package Overview

In particular, the main package consists of 5 major modules as follows:

- ***LocationServer.java*****:** reading the location configuration of system users as well as returning the input, output status to *ContextManager module*.

- ***PreferenceRepository.java:*** getting, storing the users' information along with their preferences from predefined files, getting the suggestion in term of different cases. In the end, this module returns all these information to *ContextManager module* when called.
- ***WeatherAlarms.java:*** reading the weather conditions data from predefined file and returning them to *ContextManager module*.
- ***ContextManager.java:*** accountable for communicating with other modules in order to:
  - Gather the information of system users and manage users access the system by adding and deleting the usernames.
  - Read the details of location from predefined files and manage the location searching options.
  - Evaluate sensors and alarm thresholds along with send warning messages and location suggestion to *EnviroAPPUI module* as an output for displaying on screen.
  - Setup a bridge to some modules of the IceStorm (Ice Distributed Computing Platform).
  - Set and reset the timer.
- ***EnviroAPPUI.java:*** receive users' inputs for searching options, access several classes and modules of the IceStorm and issue warning messages along with giving location suggestion and responsible for displaying these on the main menu screen.
- ***AllSensors.java:*** manage sensor data and return them to *ContextManager module,* setup a path with the IceStorm.

2.2.2. Support Package Overview

Inside the support package, it consists of 4 major classes as listed below. These are imported in specific modules when needed in order to deal with the inputs from users, read and store data from predefined text files and further purposes.

- ***LocationDetails.java:*** getting and setting location details (location name, location coordinate, information and list of location services), printing out the location details in a general defined format.
- ***Preference.java:*** getting and setting the information of system users (name, medical condition type, preferences of location suggestion).
- ***Sensor.java:*** reading, getting and setting the data (current data value, type of data, username).
- ***HandleUserInput.java:*** handling the input of user in different options (search item's information, search list of items, exit) when called in *EnviroAPPUI.java module* and querying the appropriate information from *ContextManager.java module*.

2.3. Execution Scopes and Strategies

To focus on delivering quality test plan based on the pre-defined business requirements, the EnviroSmart application will be solely focused on checking the backend functionalities on different testing levels, where the aim is to identify defects at the early stage and provide insights on how to fix them, hence preventing the manifestation of expensive errors at a later stage. Furthermore, performance evaluations are also being conducted as a following procedure to ensure that the application is capable of functioning as a proper system and is considered to be able to provide good and friendly user experiences. Any hardware or middleware components will not be tested in this project due to lack of physical devices and subscriptions from external or third-party software applications.

2.3.1. Unit Testing Strategy

JUnit is a framework for unit testing in the Java programming language. It is part of the xUnit family of unit testing frameworks, which play an important role in test-driven development. JUnit promotes the concept of "first testing, then coding," emphasizing the importance of creating test data before inserting in the piece of code. This is similar to "test a little, code a little." When creating JUnit test cases, the developer is forces to read the source code more than just writing the test code. In this stage, we will have out unit test running in Java programming language with support of external library ***JUnit 4.12***.

PURPOSE: Unit test cases involve in testing individual components or single units of the EnviroSmart application where the main objective of this test phase is to ensure that each discrete part of a class or module, is performing its corresponding task properly as expected. By deliberately implementing and utilizing this software testing in the early stage of the project, it can be less time-consuming and make the entire process to become much easier to manage and facilitates changes more straightforward.

SCOPE: In the unit testing, we will focus on the implementation of methods belong to every module across the main package. Apart from that, since we also receive another support package in the source code folder, it is necessary to have an in-depth view on numbers of classes inside. Below is the list of classes and modules where we are going to have the corresponding test methods generated:

Support package

- Sensor class
- Preference class
- LocationDetails class
- ***HandleUserInput class***

Main package

- WeatherAlarms module
- PreferenceRepository module
- LocationServer module
- AllSensors module
- ContextManager module
- ***EnviroAPPUI module***

Following out chosen strategy, the unhighlighted classed/methods shown above are testable in JUnit framework when they do not consist of any integrations related to the IceStorm. However, with the highlighted ones, there are some constraints as they contain some variables which are the communication between the classes/modules and the IceStorm.

METHODOLOGY: Since we have full access to the source code, it is easy for us to determine how to our unit tests should be conducted. Therefore, we choose to implement the ***white-box testing strategy*** along with ***automated approach*** which can help us automatically conduct all possible test cases for a specific method, field or function from the source code.

PROCESS: In order to derive unit test cases, we initially examining the project structure to locate testable files. Upon the inspection, we determined that most of the methods and variables were conducted with private modifier; hence, we must find a way to gain the access to the methods/fields for executing our unit tests later. From the laboratory 6b in tutorial class, we identify 3 methodologies that can help to overcome this obstacle:

- Give the accessibility to the methods/fields.
- Implement nested test classes inside the tested production classes.
- Conduct reflection.

At the first time, we used the first approach in order to make our process faster and the unit test was conducted without much difficulty. However, on the last moment before write the report, we start to think carefully about the change of the source code, which was from other developer(s). In that case, we decided to change to implement the reflection approach by making use of the ***Method*** and ***Field*** classes provided by Java in order to have the accessibility to every private method/field which do not have the communication with IceStorm. On the other hand, with the remain modules/classes that have some variables communicating with IceStorm, we choose to bring the methods being tested into our unit tests and modify them to fit with our test code. Yet, we still make sure that those modified functions are completely based on the algorithm of the initial methods in the source code. This not only helps us to perform unit tests on those but also keep the process less complicated than trying to access the needed variables inside the classes of IceStorm.
In addition, to prepare for unit testing, an additional package named **test.suites** is added to the project structure as a container holding all test classes (test suite) and a test runner to run the suite by using Junit testing framework. This helps managing the test execution procedure in a faster pace and does not require to run each test cases individually and manually.

For each component, at least one unit test class is created correspondingly to the target class/module, for example, PreferenceTest.java for Preference.java. The test class will contain the following elements using Junit testing framework. It is noted that not all of the elements are required to have in each test class.

1. **setUp() method**: invoked before each test method, which normally used for initializing variables, using ***@Before*** annotation.

2. **Default constructor and a collection of parameterized variables as constructor variables (**optional): helps creating multiple examples for test cases, reducing the needs of manually creating instances for testing purposes. With this element, in order to for the test run to get the parameterized variables, we use the annotation ***@Parameterized.Parameters*** to implement a public void collection of array object along with another annotation called ***@ RunWith(Parameterized.class)***

3. **tearDown() method:** invoked after each test method, which is not commonly utilized in the unit testing phase, but rather during the integration testing phase, using the annotation ***@After*** .

4. **Test cases:** testing single functions of the corresponding class, using the annotation ***@Test*** in order to make the test case runnable. Inside each test case, a number of assertions are used to compare the expected value and the actual value taken from the functions to verify whether the current test case has passed or failed. For this phase, ***assertEquals(), assertNull()*** and ***assertNotNull()*** are mostly used.

In the end, we have to confirm that no test cases return system error critically and can execute the result whether it is passed or failed. This means that in the terminal console, the test case can either return a green tick icon, confirming that the test case passes or a yellow warning, indicating that the test case fails logically. The red icon means that the test case fails to run due to wrong assertions between 2 variables or objects (not found or mismatched type) and does not return anything valuable to the testing phase. If this situation occurs, testers are required to fix the test case as soon as possible.

2.3.2. Integration Testing Strategy

PURPOSE: Integration test cases examine the data flow and dependencies between individual software modules and low-level components such as text files. Specifically, this testing phase ensures that defects in the data interconnection between modules themselves or components are detected and managed in the initial stage to reduce the probability of future lamentable occasions.

SCOPE: All test cases are administrated and deployed in both positive and negative perspective to cover every possible scenario, however, will not go into excessive detail but more in general terms. An exemplar of general test cases is the weather conditions can be divided into two categories: normal and abnormal (heavy rain, hail storm, strong wind) and used as target test cases, assuming the backend system that manage the three abnormal weather condition types are working perfectly in Unit Testing.
Specifically, this testing phase ensures the correct format conversion and continuous data transmission in both positive and negative views between all components and modules, using the incremental testing approach (bottom-up type):

- **Integration between text files and modules:** The target test cases involve verifying the data format and continuous data transferral between the pre-defined text files and corresponding modules.
- **Integration between modules**: The target test cases involve verifying the data transferral between modules are accurate and follow the software specifications.

METHODOLOGY: In the stage 1 report, we determined that we would use ***the hybrid approach***, which is a combination of Top Down and Bottom up approaches, for integration testing execution between modules. In particular, this strategy mainly focuses on testing low-level modules or individual components (text files) first and further facilitate the testing process to higher level modules. The process continues until all modules at the top are tested. The advantages of using this rather than other approaches such as top down or sandwich integration testing approach are that:

- The test conditions can be created easily.
- Modules/Classes and their respective functions are often invoked by other modules, which is more useful to test them first so that the other higher-level modules can be meaningfully and effectively integrated.
- Critical modules can be built and tested first and therefore, any errors or defects in these forms of modules are identified early in the process which helps in reducing the plausible high cost.
- Has higher success rates, with tangible and long-lasting results than other approaches. The execution time is reviewed to be faster compared to the traditional top-down approach.

PROCESS: The integration test cases are derived through ***the incremental testing approach*** (bottom-up). Modules are paired with the lowest level components and integrated initially and following that, these tested modules are further integrated for high-level module testing. This process repeats until every module at the top has been thoroughly examined. The following steps of conducting integration test cases are described as below:

- Make sure that all unit test cases are performed correctly, and no test cases result in failure during execution.
- Plot the components and models suitable for integration testing in an ascending order (low-level components at the bottom and high-level components at the top).
- Examine and conduct testing between modules based on the drawn graph. If the integration between specific modules cannot be tested due to limitations of the Ice Distributed Computing Platform, those modules are considered to be negligible, and testers proceed to the modules on the same level or move up to a higher level.
- Confirm that no test cases fail critically similar to in the unit testing phase.

2.3.3. System Testing Strategy

PURPOSE: System test cases evaluate the application system's compliance with the specified requirements on a fully integrated level in a black box manner. Through this test phase, the system is guaranteed to properly function and meets business requirements through following aspects:

- ✓ Mainly focuses on the user's ease to use the application (user-friendly), flexibility in handling controls.
- ✓ Ensure that the software application will perform well under certain real-life loads.
- ✓ Ensure that the software application has enough robustness and error handling capabilities under extreme conditions and manage to recover from crashes.

SCOPE: The system testing phase will mostly cover all modules and classes in a systematic manner where they will be tested as a fully integrated system.

METHODOLOGY: On account of the meaning of system testing, which is taking the entire system tested, we are not able to implement the test code on JUnit framework for automatic test cases execution. As a result, we will manually test and observe the output results of the system according to every test cases that we had specified from stage 1.

PROCESS: System test cases are derived in a subsequent manner as follows:

- Identify and conduct functional test cases first to ensure that all provided functionalities are working as expected. This helps ensuring that the unit and integration testing phase is delivered without flaws, respectively.

- Identify and conduct non-functional test cases to ensure that the EnviroSmart application is at a ready state for public release. These test cases are divided into 3 sub-categories of testing, which include usability testing, performance testing and recovery testing:

  1. **Usability Testing:** evaluate and guarantee that the application is able to provide friendly user experience and no improper or misleading UI interactions are conducted.

  2. **Performance Testing:** evaluate and estimate the execution performance and breakpoints of the application. This ensures that the application will always provide an immediate and delay-free connection for the user which also improves the user experience, hence creating opportunities for the program to be enhanced later on if needed in terms of scalability.

  3. **Recovery Testing:** determine how quickly the EnviroSmart system can recover after a system crash or hardware failure and confirm whether the software operations can be continued after disaster or integrity loss. This ensures that the data

transaction between modules and components continue to perform their task normally after the point at which the application crashes and assure that user's data are remain undamaged.

2.3.4. User Acceptance Testing (UAT) Strategy

PURPOSE: UAT test cases performed in the context of end user, where user stories and scenarios are specifically created to ensure the end-to-end business flow is fully validated. This helps verifying the software system to meet the stated requirements before moving to the production environment.

SCOPE: In the context of EnviroSmart application, the development team has identified a total of 6 user stories based on the software requirements and test assumptions on the UI:

- User logs in to the application
- User wants to edit the preferences
- User searches for a specific item of interest
- User searches for a list of items of interest in current location
- User receives warnings and suggestions
- User logs out of the application

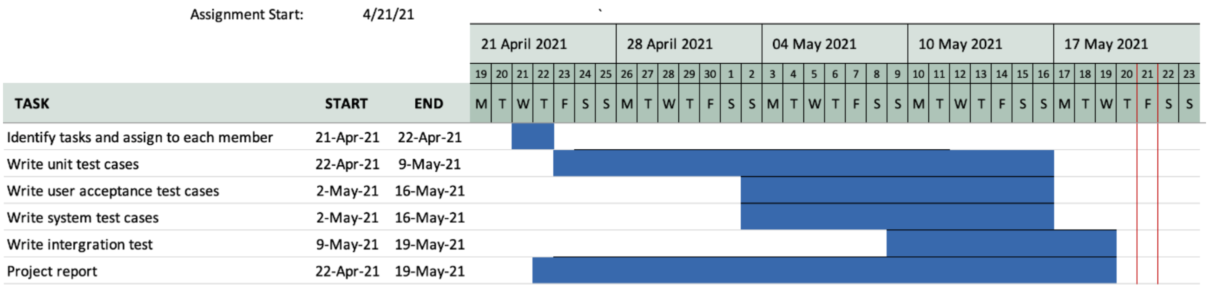
METHODOLOGY: For this testing type, instead executed the test with JUnit framework which is not appropriate since the we are acting as a user who does not contribute to the development process of this application system. Therefore, we decided to manually run the application and observe the actual results through IntelliJ IDE console where the inputs are modified respect to the test data specified in our test cases.

PROCESS: UAT test cases are written with specific test measures in mind. The test cases cover the majority of the intended scenarios and provide concrete inputs and expected outcomes. The following steps were used to perform testing in an end user perspective:

- Identify all possible features that users can perform interactions with the application (User Story).
- For each User Story, identify possible scenarios that could potentially happen and conduct test cases accordingly (User Scenarios).

2.4. Schedule & Estimation

The estimated test plan for Stage 2 is described in the following figure, where the start date and end date for each individual tasks are thoroughly identified. With a team of 5 professional personnel, the estimated duration of Stage 1 is exactly 1 month.



Assignment Start: 4/21/21

TASK	START	END
Identify tasks and assign to each member	21-Apr-21	22-Apr-21
Write unit test cases	22-Apr-21	9-May-21
Write user acceptance test cases	2-May-21	16-May-21
Write system test cases	2-May-21	16-May-21
Write intergration test	9-May-21	19-May-21
Project report	22-Apr-21	19-May-21

Figure 1: Test Schedule

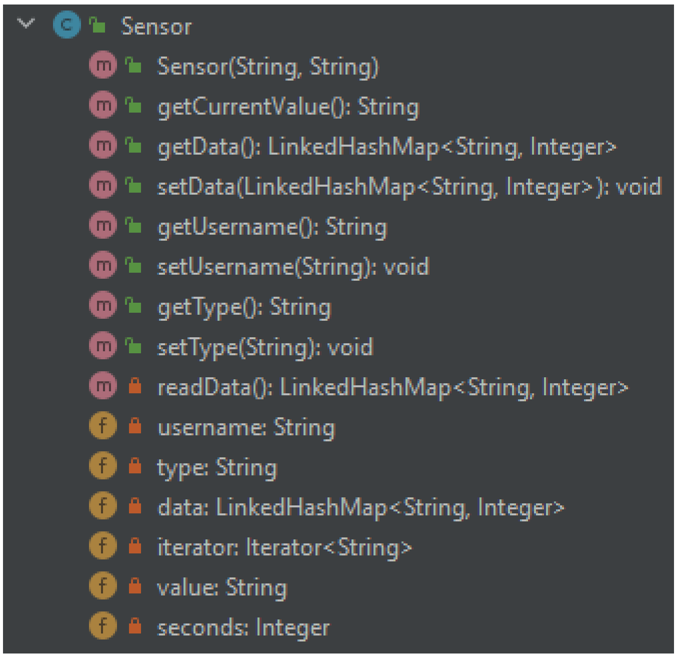
Meeting minutes (refer to the Appendix A) are also conducted for at least once a week to capture the divided workload for each individual personnel, hence creating soft deadlines for easier management.

3. Unit Test

3.1. Support Package

3.1.1. Test the getters and setters methods in the Sensor.java file

The following figure shows 8 methods that are going to be tested along with 6 objects as data outputs. Each method has it own job in order to contribute to the system.

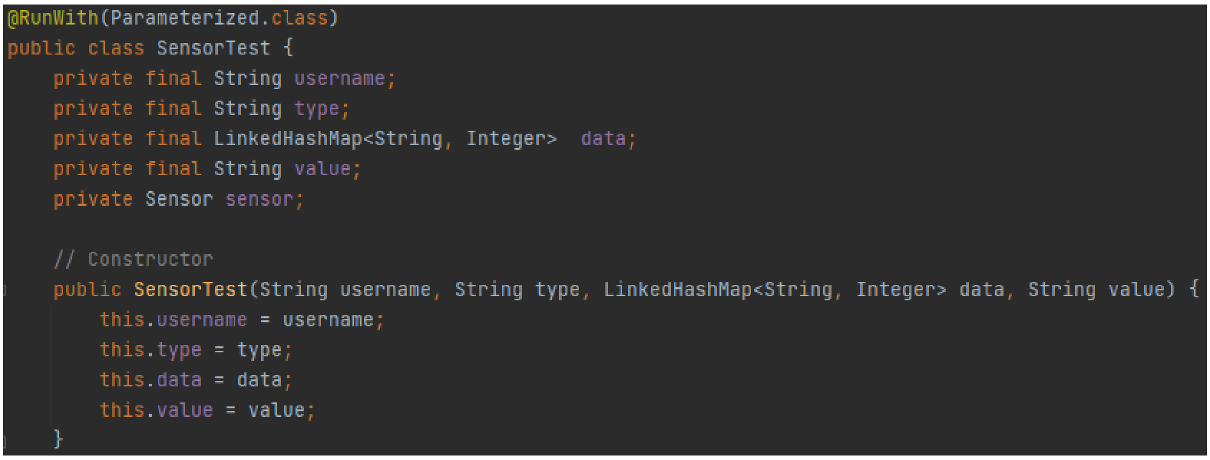


```
Sensor
    Sensor(String, String)
    getCurrentValue(): String
    getData(): LinkedHashMap<String, Integer>
    setData(LinkedHashMap<String, Integer>): void
    getUsername(): String
    setUsername(String): void
    getType(): String
    setType(String): void
    readData(): LinkedHashMap<String, Integer>
    username: String
    type: String
    data: LinkedHashMap<String, Integer>
    iterator: Iterator<String>
    value: String
    seconds: Integer
```

Figure 2: Structure of the Sensor class

- *getCurrentValue():* get all the current data from a specific text file.
- *getData():* get all data from a specific text file.
- *setData():* change the data context in a specific text file.
- *getUsername():* get the logged in username of the application.
- *setUsername():* set the logged in username of the application.
- *getType():* get the sensor type.
- *setType():* set the sensor type.
- *readData():* read the data from a specific text file.

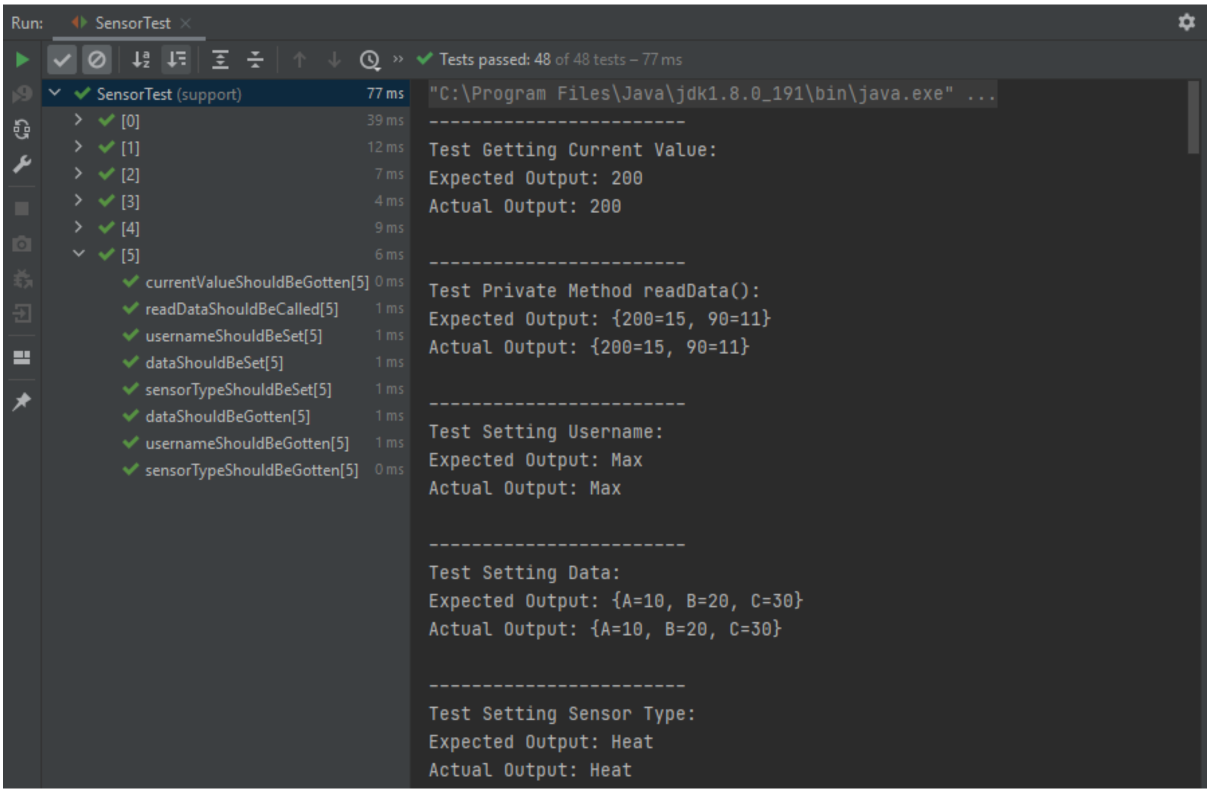
Making use of Parameterized.Parameters annotation, we declare some test data which follow the format of parameters initialized in the constructor. Those are considered to be our expected results. Hence, when we perform unit testing, these predefined methods are invoked from the Sensor class by the corresponding test methods. At that time, we are able to compare with actual data from predefined files given in the source code folder.



```
@RunWith(Parameterized.class)
public class SensorTest {
    private final String username;
    private final String type;
    private final LinkedHashMap<String, Integer>  data;
    private final String value;
    private Sensor sensor;

    // Constructor
    public SensorTest(String username, String type, LinkedHashMap<String, Integer> data, String value) {
        this.username = username;
        this.type = type;
        this.data = data;
        this.value = value;
    }
```

Figure 3: Declaration of test variables for the JUnit Test of Sensor class

After executing 8 test methods, we come up with the test results shown in the following figure where 48 out of 48 tests are passed, 6 tests per methods. Please refer to the test code files in the source code eeet2578 for further details about the test data of this Sensor tests execution.



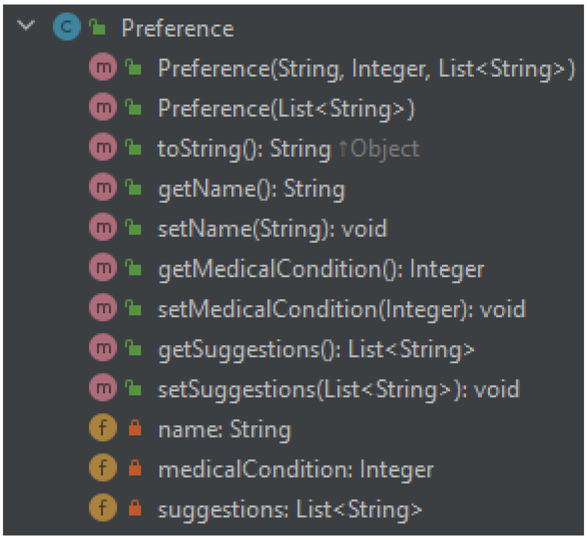
```
✓ Tests passed: 48 of 48 tests – 77 ms

✓ SensorTest (support)                 77 ms
   ✓ [0]                               39 ms
   ✓ [1]                               12 ms
   ✓ [2]                                7 ms
   ✓ [3]                                4 ms
   ✓ [4]                                9 ms
   ✓ [5]                                6 ms
      ✓ currentValueShouldBeGotten[5]   0 ms
      ✓ readDataShouldBeCalled[5]       1 ms
      ✓ usernameShouldBeSet[5]          1 ms
      ✓ dataShouldBeSet[5]              1 ms
      ✓ sensorTypeShouldBeSet[5]        1 ms
      ✓ dataShouldBeGotten[5]           1 ms
      ✓ usernameShouldBeGotten[5]       1 ms
      ✓ sensorTypeShouldBeGotten[5]     0 ms
```

```
"C:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...
------------------------
Test Getting Current Value:
Expected Output: 200
Actual Output: 200

------------------------
Test Private Method readData():
Expected Output: {200=15, 90=11}
Actual Output: {200=15, 90=11}

------------------------
Test Setting Username:
Expected Output: Max
Actual Output: Max

------------------------
Test Setting Data:
Expected Output: {A=10, B=20, C=30}
Actual Output: {A=10, B=20, C=30}

------------------------
Test Setting Sensor Type:
Expected Output: Heat
Actual Output: Heat
```

Figure 4: The number of test cases executed and theirs test results

3.1.2. Test the getters and setters methods in the Preference.java file

The following figure shows 7 methods that are going to be tested along with 3 objects as data outputs. In addition, for this Preference class, we also have a small test for the constructor Preference(List<String>) in order to check if it is called correctly. Hence, we are going to have 8 test methods in total, each has it own job in order to contribute to the system.

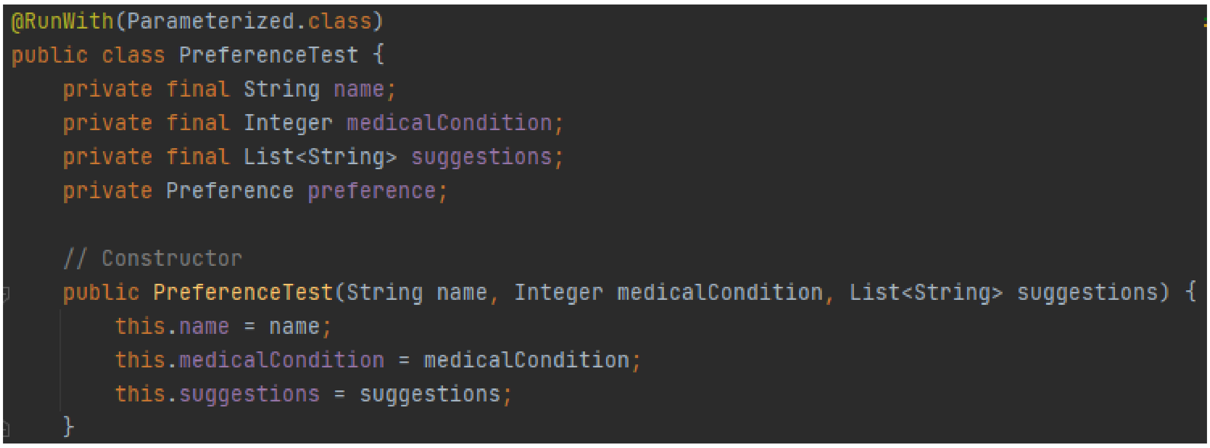


```
Preference
    Preference(String, Integer, List<String>)
    Preference(List<String>)
    toString(): String ↑Object
    getName(): String
    setName(String): void
    getMedicalCondition(): Integer
    setMedicalCondition(Integer): void
    getSuggestions(): List<String>
    setSuggestions(List<String>): void
    name: String
    medicalCondition: Integer
    suggestions: List<String>
```

Figure 5: Structure of the Preference class

- *getName():* get the name in the preference file.
- *setName():* change the name in the preference file.
- *getMedicalCondition():* get the medical condition type in the preference file.
- *setMedicalCondition():* change the medical condition type in the preference file.
- *getSuggestions():* get the suggestions in the preference file.
- *setSuggestions():* change the suggestions in the preference file.
- *toString():* print out all content in the preference file.

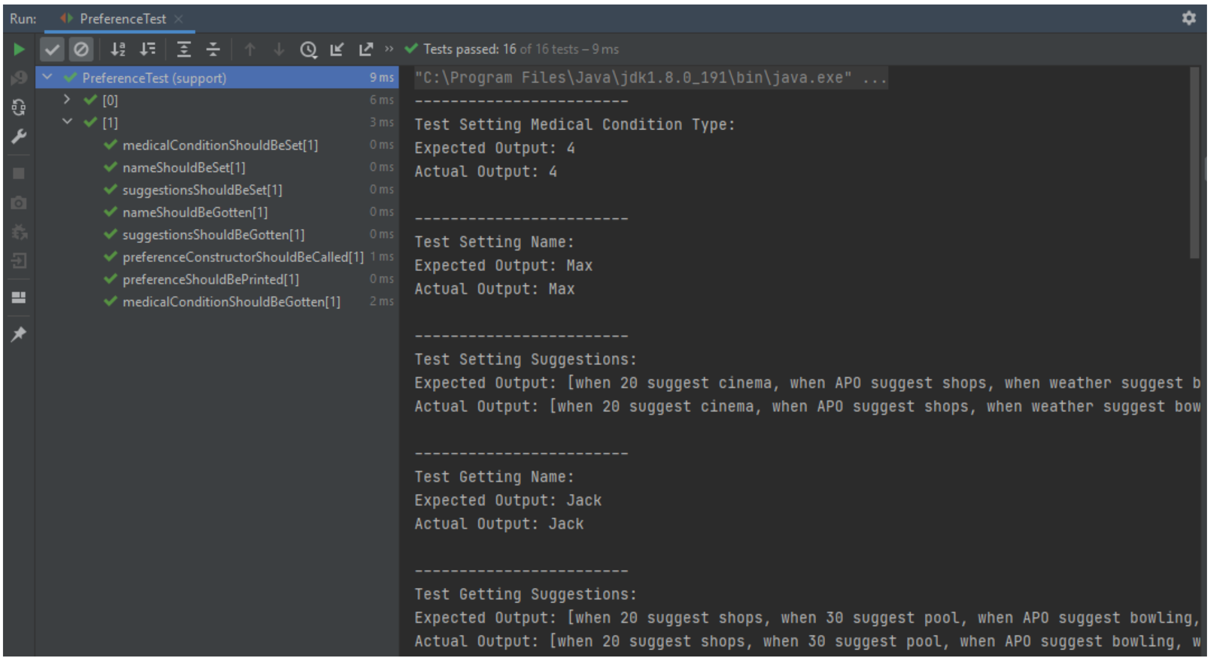
Making use of Parameterized.Parameters annotation, we declare some test data which follow the format of parameters initialized in the constructor. Those are considered to be our expected results. Hence, when we perform unit testing, these predefined methods are invoked from the Preference class by the corresponding test methods. At that time, we are able to compare with actual data from predefined files given in the source code folder.



```java
@RunWith(Parameterized.class)
public class PreferenceTest {
    private final String name;
    private final Integer medicalCondition;
    private final List<String> suggestions;
    private Preference preference;

    // Constructor
    public PreferenceTest(String name, Integer medicalCondition, List<String> suggestions) {
        this.name = name;
        this.medicalCondition = medicalCondition;
        this.suggestions = suggestions;
    }
```

Figure 6: Declaration of test variables for the JUnit Test of Preference class

After executing 8 test methods, we come up with the test results shown in the following figure where 16 out of 16 tests are passed, 2 tests per methods corresponding to 2 predefined usernames "Jack" and "David". Please refer to the test code files in the source code eeet2578 for further details about the test data of this Sensor tests execution.



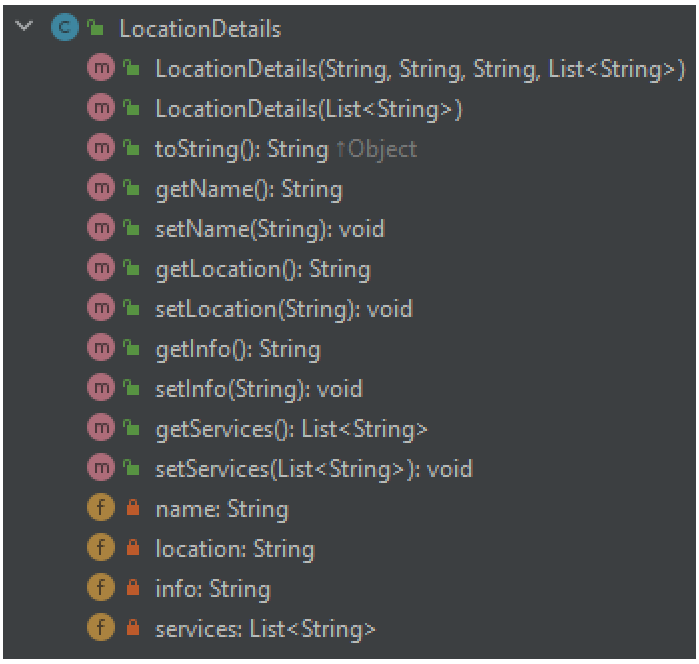
```
✓ Tests passed: 16 of 16 tests – 9 ms

✓ PreferenceTest (support)                      9 ms
  ✓ [0]                                         6 ms
  ✓ [1]                                         3 ms
    ✓ medicalConditionShouldBeSet[1]            0 ms
    ✓ nameShouldBeSet[1]                        0 ms
    ✓ suggestionsShouldBeSet[1]                 0 ms
    ✓ nameShouldBeGotten[1]                     0 ms
    ✓ suggestionsShouldBeGotten[1]              0 ms
    ✓ preferenceConstructorShouldBeCalled[1]    1 ms
    ✓ preferenceShouldBePrinted[1]              0 ms
    ✓ medicalConditionShouldBeGotten[1]         2 ms
```

```
"C:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...
------------------------
Test Setting Medical Condition Type:
Expected Output: 4
Actual Output: 4

------------------------
Test Setting Name:
Expected Output: Max
Actual Output: Max

------------------------
Test Setting Suggestions:
Expected Output: [when 20 suggest cinema, when APO suggest shops, when weather suggest b
Actual Output: [when 20 suggest cinema, when APO suggest shops, when weather suggest bow

------------------------
Test Getting Name:
Expected Output: Jack
Actual Output: Jack

------------------------
Test Getting Suggestions:
Expected Output: [when 20 suggest shops, when 30 suggest pool, when APO suggest bowling,
Actual Output: [when 20 suggest shops, when 30 suggest pool, when APO suggest bowling, w
```

Figure 7: The number of test cases executed and theirs test results

3.1.3. Test the getters and setters methods in the LocationDetails.java file

The following figure shows 9 methods that are going to be tested along with 4 objects as data outputs. In addition, for this Preference class, we also have a small test for the constructor LocationDetails(List<String>) in order to check if it is called correctly. Hence, we are going to have 10 test methods in total, each has it own job in order to contribute to the system.

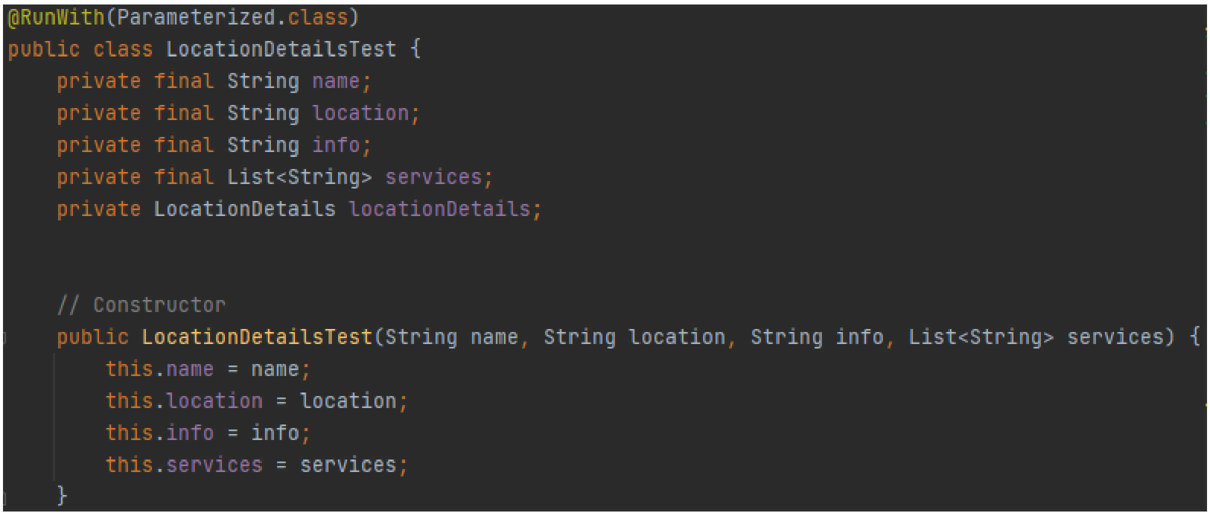


```
LocationDetails
    LocationDetails(String, String, String, List<String>)
    LocationDetails(List<String>)
    toString(): String ↑Object
    getName(): String
    setName(String): void
    getLocation(): String
    setLocation(String): void
    getInfo(): String
    setInfo(String): void
    getServices(): List<String>
    setServices(List<String>): void
    name: String
    location: String
    info: String
    services: List<String>
```

Figure 8: Structure of the LocationDetails class

- *getName():* get the name in the preference file.
- *setName():* change the name in the preference file.
- *getLocation():* get the medical condition type in the preference file.
- *setLocation():* change the medical condition type in the preference file.
- *getInfo():* get the suggestions in the preference file.
- *setInfo():* change the suggestions in the preference file.
- *getServices():* get the services of a specific location.
- *setServices():* change the services of a specific location.
- *toString():* print out all content in the preference file.
- *LocationDetails(List<String>):* the constructor of the LocationDetails.java file that takes location details as parameters.

Making use of Parameterized.Parameters annotation, we declare some test data which follow the format of parameters initialized in the constructor. Those are considered to be our expected results. Hence, when we perform unit testing, these predefined methods are invoked from the LocationDetails class by the corresponding test methods. At that time, we are able to compare with actual data from predefined files given in the source code folder.



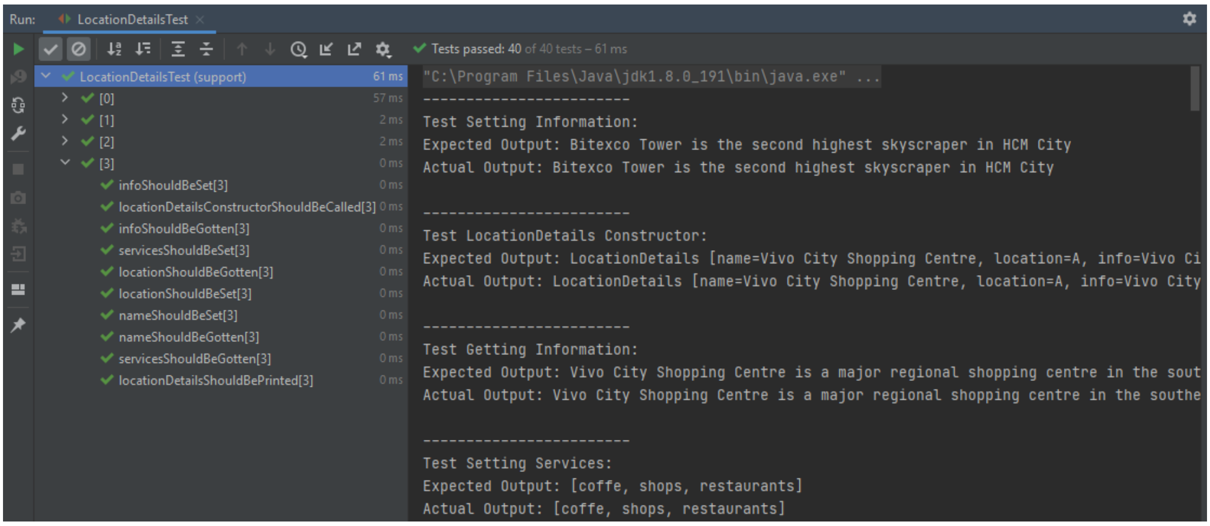
```java
@RunWith(Parameterized.class)
public class LocationDetailsTest {
    private final String name;
    private final String location;
    private final String info;
    private final List<String> services;
    private LocationDetails locationDetails;


    // Constructor
    public LocationDetailsTest(String name, String location, String info, List<String> services) {
        this.name = name;
        this.location = location;
        this.info = info;
        this.services = services;
    }
```

Figure 9: Declaration of test variables for the JUnit Test of LocationDetails class

After executing 10 test methods, we come up with the test results shown in the following figure where 40 out of 40 tests are passed, 4 tests per methods corresponding to 4 predefined location names:

- "Vivo City Shopping Centre"
- "Crescent Mall"
- "Dam Sen Parklands"
- "Ho Chi Minh City, Downtown"

Please refer to the test code files in the source code eeet2578 for further details about the test data of this Sensor tests execution.



```
Tests passed: 40 of 40 tests – 61 ms

✓ LocationDetailsTest (support)                         61 ms
  ✓ [0]                                                 57 ms
  ✓ [1]                                                  2 ms
  ✓ [2]                                                  2 ms
  ✓ [3]                                                  0 ms
    ✓ infoShouldBeSet[3]                                 0 ms
    ✓ locationDetailsConstructorShouldBeCalled[3]        0 ms
    ✓ infoShouldBeGotten[3]                              0 ms
    ✓ servicesShouldBeSet[3]                             0 ms
    ✓ locationShouldBeGotten[3]                          0 ms
    ✓ locationShouldBeSet[3]                             0 ms
    ✓ nameShouldBeSet[3]                                 0 ms
    ✓ nameShouldBeGotten[3]                              0 ms
    ✓ servicesShouldBeGotten[3]                          0 ms
    ✓ locationDetailsShouldBePrinted[3]                  0 ms
```

```
"C:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...
------------------------
Test Setting Information:
Expected Output: Bitexco Tower is the second highest skyscraper in HCM City
Actual Output: Bitexco Tower is the second highest skyscraper in HCM City

------------------------
Test LocationDetails Constructor:
Expected Output: LocationDetails [name=Vivo City Shopping Centre, location=A, info=Vivo Ci
Actual Output: LocationDetails [name=Vivo City Shopping Centre, location=A, info=Vivo City

------------------------
Test Getting Information:
Expected Output: Vivo City Shopping Centre is a major regional shopping centre in the sout
Actual Output: Vivo City Shopping Centre is a major regional shopping centre in the southe

------------------------
Test Setting Services:
Expected Output: [coffe, shops, restaurants]
Actual Output: [coffe, shops, restaurants]
```

Figure 10: The number of test cases executed and theirs test results

3.2. Main Package

3.2.1. WeatherAlarms.java

The table below shows the available functions that are required to be tested in the WeatherAlarms java file:

Table 1: List of tested function(s)

Name	Return Type	Description
readWeatherConditions()	List<Integer>	Get all weather conditions from the weather_alarms.txt

Table 2: Test Function readWeatherConditions()

Title	Test Function readWeatherConditions()
Description	Function readWeatherConditions() should be successfully called and get all the weather condition type from the weather_alarms.txt
Preconditions	LocationDetails class starts up properly.
Test Data	0 1 2 3
Steps	+ Use locationDetails.getServices() to retrieve the services of the location. + Compare with the expected value using assertEquals() function.
Expected results	[0, 1, 2, 3]
Actual results	[0, 1, 2, 3]
Test results	**PASSED**

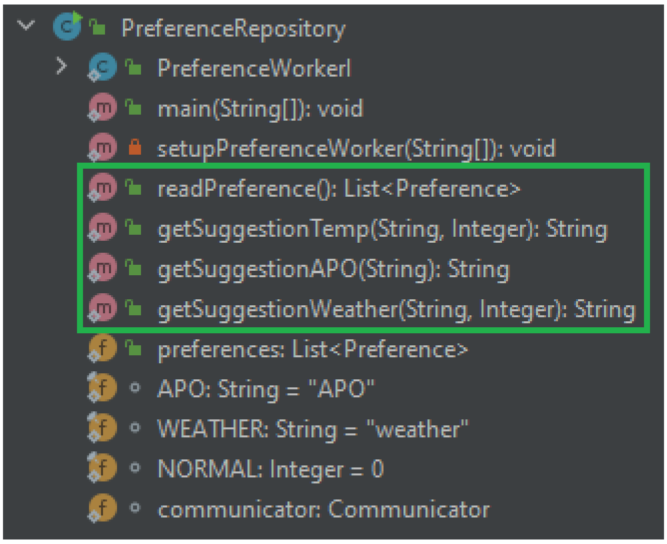
Table 3: Test Function readWeatherConditions()

Title	Test Function readWeatherConditions()
Description	Function readWeatherConditions() should be successfully called and get all the weather condition type from the weather_alarms.txt
Preconditions	LocationDetails class starts up properly.
Test Data	0 1 2 3 4

	5 6 7 8 9
Steps	+ Use locationDetails.getServices() to retrieve the services of the location. + Compare with the expected value using assertEquals() function.
Expected results	The system returns error showing that the number of the weather condition does not match the predefined one.
Actual results	The system returns error showing that the number of the weather condition does not match the predefined one.
Test results	**PASSED**

3.2.2. PreferenceRepository.java

The following figure shows 4 methods that are going to be tested along with 4 objects as data outputs. Each method has it own job in order to contribute to the system.



```
PreferenceRepository
    PreferenceWorkerI
    main(String[]): void
    setupPreferenceWorker(String[]): void
    readPreference(): List<Preference>
    getSuggestionTemp(String, Integer): String
    getSuggestionAPO(String): String
    getSuggestionWeather(String, Integer): String
    preferences: List<Preference>
    APO: String = "APO"
    WEATHER: String = "weather"
    NORMAL: Integer = 0
    communicator: Communicator
```

Figure 11: Structure of PreferenceRepository class

- *readPreference():* read all preferences from the preference text file
- *getSuggestionTemp(String, Integer):* get suggested services from the temperature preference that takes username and temperature's threshold as parameters
- *getSuggestionAPO(String):* get suggested services from the APO preference that takes username as parameter
- *getSuggestionWeather(String, Integer):* get suggested services from the weather preference that takes username and weather condition type as parameters

Table 4: Test Function readPreference()

Title	Test Function readPreference()
Description	Function readPreference() should be successfully called and get all the available preferences in the preference text file.
Preconditions	+ PreferenceRepository class starts up properly. + Preference text file exists and has corresponding context.
Test Data	name: Jack Medical Condition Type: 2 pref: when 20 suggest shops pref: when 30 suggest pool pref: when APO suggest bowling pref: when weather suggest cinema *** name: David Medical Condition Type: 3 pref: when 16 suggest pool pref: when APO suggest cinema pref: when weather suggest shops ***
Steps	+ Use preferenceRepository.readPreference() to retrieve all the available preferences in the preference text file. + Compare with the expected value using assertEquals() function.
Expected results	[Preference [name=Jack, medical condition=2, suggestions=[when 20 suggest shops, when 30 suggest pool, when APO suggest bowling, when weather suggest cinema]], [Preference [name=David, medical condition=3, suggestions=[when 16 suggest pool, when APO suggest cinema, when weather suggest shops]]
Actual results	[Preference [name=Jack, medical condition=2, suggestions=[when 20 suggest shops, when 30 suggest pool, when APO suggest bowling, when weather suggest cinema]], [Preference [name=David, medical condition=3, suggestions=[when 16 suggest pool, when APO suggest cinema, when weather suggest shops]]
Test results	**PASSED**

Table 5: Test Function getSuggestionTemp(String name, Integer tempThreshold) #1

Title	Test Function getSuggestionTemp(String name, Integer tempThreshold) #1
Description	Function getSuggestionTemp() should be successfully called and get all the suggested services from the temperature preference. The defined temperature is inside the range of the temperature preference.

Preconditions	+ PreferenceRepository class starts up properly. + Preference text file exists and has corresponding context.
Test Data	name: Jack Medical Condition Type: 2 pref: when 20 suggest shops pref: when 30 suggest pool pref: when APO suggest bowling pref: when weather suggest cinema + Defined temperature: 30
Steps	+ Use preferenceRepository.getSuggestionTemp() to get the suggested services from the temperature preference. + Compare with the expected value using assertEquals() function.
Expected results	pool
Actual results	pool
Test results	**PASSED**

Table 6: Test Function getSuggestionTemp(String name, Integer tempThreshold) #2

Title	Test Function getSuggestionTemp(String name, Integer tempThreshold) #2
Description	Function getSuggestionTemp() should be successfully called and get all the suggested services from the temperature preference. The defined temperature is outside the range of the temperature preference.
Preconditions	+ PreferenceRepository class starts up properly. + Preference text file exists and has corresponding context.
Test Data	name: Jack Medical Condition Type: 2 pref: when 20 suggest shops pref: when 30 suggest pool pref: when APO suggest bowling pref: when weather suggest cinema + Defined temperature: 10
Steps	+ Use preferenceRepository.getSuggestionTemp() to get the suggested services from the temperature preference. + Compare with the expected value using assertEquals() function.
Expected results	null
Actual results	null

Test results	**PASSED**

Table 7: Test Function getSuggestionTemp(String name, Integer tempThreshold) #3

Title	Test Function getSuggestionTemp(String name, Integer tempThreshold) #3
Description	Function getSuggestionTemp() should be successfully called and get all the suggested services from the temperature preference. The defined temperature is invalid as it is a negative number.
Preconditions	+ PreferenceRepository class starts up properly. + Preference text file exists and has corresponding context.
Test Data	name: Jack Medical Condition Type: 2 pref: when 20 suggest shops pref: when 30 suggest pool pref: when APO suggest bowling pref: when weather suggest cinema + Defined temperature: -1
Steps	+ Use preferenceRepository.getSuggestionTemp() to get the suggested services from the temperature preference. + Compare with the expected value using assertEquals() function.
Expected results	null
Actual results	null
Test results	**PASSED**

Table 8: Test Function getSuggestionAPO(String name)

Title	Test Function getSuggestionAPO(String name)
Description	Function getSuggestionTemp() should be successfully called and get all the suggested services from the APO preference.
Preconditions	+ PreferenceRepository class starts up properly. + Preference text file exists and has corresponding context.
Test Data	name: Jack Medical Condition Type: 2 pref: when 20 suggest shops pref: when 30 suggest pool pref: when APO suggest bowling pref: when weather suggest cinema

Steps	+ Use preferenceRepository.getSuggestionTemp() to get the suggested services from the temperature preference. + Compare with the expected value using assertEquals() function.
Expected results	bowling
Actual results	bowling
Test results	**PASSED**

Table 9: Test Function getSuggestionWeather(String name, Integer weather) #1

Title	Test Function getSuggestionWeather(String name, Integer weather) #1
Description	Function getSuggestionWeather() should be successfully called and get all the suggested services from the weather preference. The defined weather is inside the range of the weather condition types.
Preconditions	+ PreferenceRepository class starts up properly. + Preference text file exists and has corresponding context.
Test Data	name: Jack Medical Condition Type: 2 pref: when 20 suggest shops pref: when 30 suggest pool pref: when APO suggest bowling pref: when weather suggest cinema + Defined weather condition type: 1
Steps	+ Use preferenceRepository.getSuggestionTemp() to get the suggested services from the temperature preference. + Compare with the expected value using assertEquals() function.
Expected results	cinema
Actual results	cinema
Test results	**PASSED**

Table 10: Test Function getSuggestionWeather(String name, Integer weather) #2

Title	Test Function getSuggestionWeather(String name, Integer weather) #2
Description	Function getSuggestionWeather() should be successfully called and get all the suggested services from the weather preference. The defined weather is outside the range of the weather condition types.
Preconditions	+ PreferenceRepository class starts up properly. + Preference text file exists and has corresponding context.

Test Data	name: Jack Medical Condition Type: 2 pref: when 20 suggest shops pref: when 30 suggest pool pref: when APO suggest bowling pref: when weather suggest cinema + Defined weather condition type: 5
Steps	+ Use preferenceRepository.getSuggestionTemp() to get the suggested services from the temperature preference. + Compare with the expected value using assertEquals() function.
Expected results	null
Actual results	cinema
Test results	**FAILED**

Table 11: Test Function getSuggestionWeather(String name, Integer weather) #3

Title	Test Function getSuggestionWeather(String name, Integer weather) #3
Description	Function getSuggestionWeather() should be successfully called and get all the suggested services from the weather preference. The defined weather is invalid as the defined weather condition type is a negative number.
Preconditions	+ PreferenceRepository class starts up properly. + Preference text file exists and has corresponding context.
Test Data	name: Jack Medical Condition Type: 2 pref: when 20 suggest shops pref: when 30 suggest pool pref: when APO suggest bowling pref: when weather suggest cinema + Defined weather condition type: -1
Steps	+ Use preferenceRepository.getSuggestionTemp() to get the suggested services from the temperature preference. + Compare with the expected value using assertEquals() function.
Expected results	null
Actual results	cinema
Test results	**FAILED**

3.2.3. LocationServer.java

The table below shows the available functions that are required to be tested in the LocationServer java file:

Table 12: List of tested funcion(s)

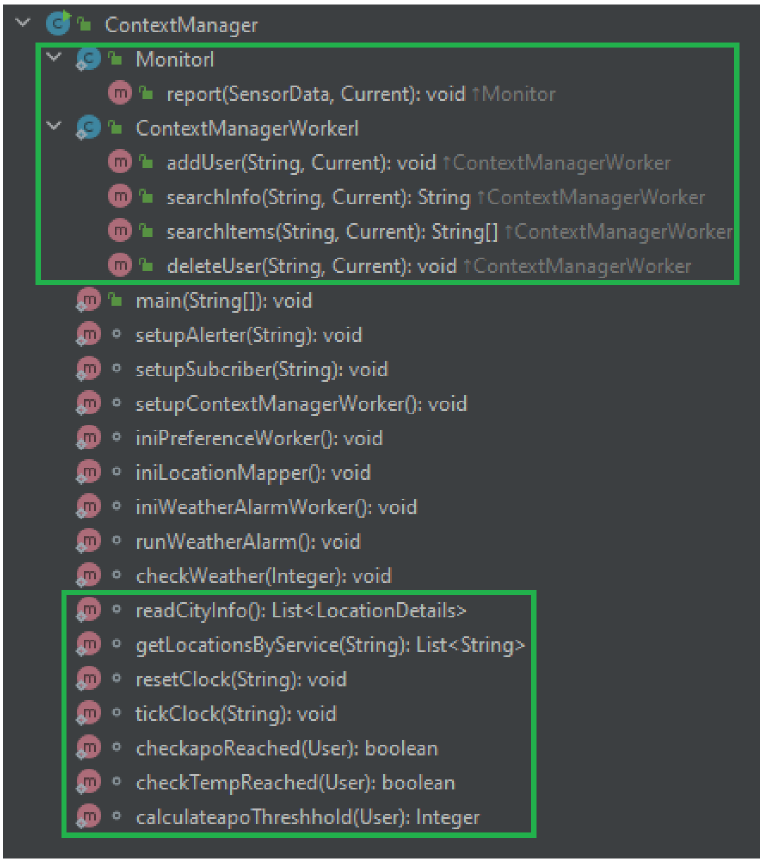
Name	Return Type	Description
readConfig()	LinkedHashMap<String, String>	Read all the configurations from the LocationServerConfig text file

Table 13: Test Function readConfig()

Title	Test Function readConfig()
Description	Function readConfig() should be successfully called and get all the location configurations from the LocationServerConfig.txt
Preconditions	+ LocationServer class starts up properly. + LocationServerConfig text file exists and has corresponding context.
Test Data	Indoor: A,B Outdoor: C,D
Steps	+ Use locationServer.readConfig() to get all the location configurations. + Compare with the expected value using assertEquals() function.
Expected results	{A=Indoor, B=Indoor, C=Outdoor, D=Outdoor}
Actual results	{A=Indoor, B=Indoor, C=Outdoor, D=Outdoor}
Test results	**PASSED**

3.2.4. ContextManager.java

The following figure shows 12 methods that are going to be tested. Each method has it own job in order to contribute to the system.



```
ContextManager
  MonitorI
    report(SensorData, Current): void ↑Monitor
  ContextManagerWorkerI
    addUser(String, Current): void ↑ContextManagerWorker
    searchInfo(String, Current): String ↑ContextManagerWorker
    searchItems(String, Current): String[] ↑ContextManagerWorker
    deleteUser(String, Current): void ↑ContextManagerWorker
  main(String[]): void
  setupAlerter(String): void
  setupSubcriber(String): void
  setupContextManagerWorker(): void
  iniPreferenceWorker(): void
  iniLocationMapper(): void
  iniWeatherAlarmWorker(): void
  runWeatherAlarm(): void
  checkWeather(Integer): void
  readCityInfo(): List<LocationDetails>
  getLocationsByService(String): List<String>
  resetClock(String): void
  tickClock(String): void
  checkapoReached(User): boolean
  checkTempReached(User): boolean
  calculateapoThreshhold(User): Integer
```

Figure 12: Structure of ContextManager class

- *report(SensorData, Current): report the activities of apo sensor and temperature sensor to*
- *addUser(String, Current):* add user to the system by using defined username.
- *deleteUser(String, Current):* delete user by username out of the system.
- *searchInfo(String, Current):* search the information of a specific item.
- *searchItems(String, Current):* search list of items of interest around the current location.
- *readCityInfo():* read the full context of the CityInfo.txt
- *getLocationsByService(String):* get location names based on whether that location has specific services or not.
- *resetClock(String):* reset the clock to 0.
- *tickClock(String):* run the clock to add 1.
- *checkapoReached(User):* check if the APO threshold has been reached.
- *checkTempReached(User):* check if the temperature threshold has been reached.
- *calculateapoThreshold(User):* calculate the APO threshold.

3.2.4.1. Test Function addUser(String username, Current current)

Table 14: Test data for unit test of method addUser(String, Current) in the ContextManager module

Test Case	Test Data	Expected Results	Actual Results	Test Results

Check if new user with username “Jack” can be initially added to the system when addUser() method is called.	username = “Jack”
	expectedResult = true
		Expect: true	Actual: true	**PASSED**

Check if new user with username “David” can be initially added to the system when addUser() method is called.	username = “David”
	expectedResult = true
		Expect: true	Actual: true	**PASSED**

Check if username “Jack” can be added again to the system when addUser() method is called.	username = “Jack”
	expectedResult = false
		Expect: false	Actual: true	**FAILED**

Check if username “David” can be added again to the system when addUser() method is called.	username = “David”
	expectedResult = false
		Expect: false	Actual: true	**FAILED**

Check if username “123” can be again to the system when addUser() method is called.	username = “123”
	expectedResult = false
		Expect: false	Actual: true	**FAILED**

Check if username “//-(” can be again to the system when addUser() method is called.	username = “//-(”
	expectedResult = false
		Expect: false	Actual: true	**FAILED**

Check if username <null> can be again to the system when addUser() method is called.	username = null
	expectedResult = false
		Expect: false	Actual: true	**FAILED**

3.2.4.2. Test Function deleteUser(String username, Current current)

Table 15: Test data for unit test of method deleteUser(String, Current) in the ContextManager module

Test Case	Test Data	Expected Results	Actual Results	Test Results

Check if new user with username “Jack” can be deleted from the system	username = “Jack”
	expectedResult = true
		Expect: true	Actual: true	**PASSED**

when deleteUser() method is called.				

Check if username “David” can be deleted from the system when deleteUser() method is called.	username = “David”
	expectedResult = true
		Expect: true	Actual: false	**FAILED**

3.2.4.3. Test Function searchInfo(String username, Current current)

Table 16: Test data for unit test of method searchInfo(String, Current) in the ContextManager module

Test Case	Test Data	Expected Results	Actual Results	Test Results

Check if user with username “Jack” searches for information of an item name “Vivo City Shopping Centre”.	username = “Jack”
	name = “Vivo City Shopping Centre”
	info = “Vivo City Shopping Centre is a major regional shopping centre in the southern suburb of Ho Chi Minh City, Vietnam. It is the second largest shopping centre in the southern suburbs of Ho Chi Minh City, by gross area, and contains the only H&M store in that region.”
		Expect: “Vivo City Shopping Centre is a major regional shopping centre in the southern suburb of Ho Chi Minh City, Vietnam. It is the second largest shopping centre in the southern suburbs of Ho Chi Minh City, by gross area, and contains the only H&M store in that region.”	Actual: Same as the expected result	**PASSED**

Check if user with username “Jack” searches for information of an item name “Crescent Mall”.	username = “Jack”
	name = “Crescent Mall”
	info = “Crescent Mall Shopping Centre is located 10km South of the Ho Chi Minh City central business district(CBD) and includes Banana Republic, Baskin Robins, CGV Cinema, Bobapop and over 130 specialty stores.”
		Expect: “Crescent Mall Shopping Centre is located 10km South of the Ho Chi Minh City central business district(CBD) and includes Banana Republic, Baskin Robins, CGV Cinema, Bobapop and over 130 specialty stores.”	Actual: Same as the expected result	**PASSED**

Check if user with username “David” searches for information of an item name “Dam Sen Parklands”.	username = “David”
	name = “Dam Sen Parklands”
	info = “The Dam Sen Parklands area was created as part of the rejuvenation of the industrial upgrade undertaken for World Expo 1988. The Parklands area is spacious with plenty of green and spaces for all ages. A big lake
		Expect: “The Dam Sen Parklands area was created as part of the rejuvenation of the industrial upgrade undertaken for World Expo 1988. The Parklands area is spacious with plenty of green and spaces for all ages. A big lake	Actual: Same as the expected result	**PASSED**

	promenade stretches the area of Dam Sen Parklands.”	promenade stretches the area of Dam Sen Parklands.”		

Check if user with username “David” searches for information of an item name “Ho Chi Minh City, Downtown”.	username = “David”
	name = “Ho Chi Minh City, Downtown”
	info = “The Ho Chi Minh City central business district (CBD), or 'the City' is located on a central point in district One. The point, known at its tip as Central Point, slopes upward to the north-west where 'the city' is bounded by parkland and the inner city suburb of District 3, District 4 and District 5.”
		Expect: “The Ho Chi Minh City central business district (CBD), or 'the City' is located on a central point in district One. The point, known at its tip as Central Point, slopes upward to the north-west where 'the city' is bounded by parkland and the inner city suburb of District 3, District 4 and District 5.”	Actual: Same as the expected result	**PASSED**

Check if user with username “Jack” searches for information of an item name “Southbank”.	username = “Jack”
	name = “Southbank”
	info = null
		Expected: null	Actual: null	**PASSED**

Check if user with username “Jack” searches for information of an item name <null>.	username = “Jack”
	name = null
	info = null
		Expected: null	Actual: null	**PASSED**

3.2.4.4. Test Function searchItems(String username, Current current)

Table 17: Test data for unit test of method searchItems(String, Current) in the ContextManager module

Test Case	Test Data	Expected Results	Actual Results	Test Results

Check if user with username “Jack” searches for list of items located around location “A”.	username = “Jack”
	currentLocation = “A”
	name = “Vivo City Shopping Centre”
		Expect: [Vivo City Shopping Centre]	Actual: [Vivo City Shopping Centre]	**PASSED**

Check if user with username “Jack” searches for list of items located around location “B”.	username = “Jack”
	currentLocation = “B”
	name = “Crescent Mall”
		Expect: [Crescent Mall]	Actual: [Crescent Mall]	**PASSED**

Check if user with username "David" searches for list of items located around location "C".	username = "David"
	currentLocation = "C"
	name = "Dam Sen Parklands"
		Expect: [Dam Sen Parklands]	Actual: [Dam Sen Parklands]	**PASSED**

Check if user with username "David" searches for list of items located around location "D".	username = "David"
	currentLocation = "D"
	name = "Ho Chi Minh City, Downtown"
		Expect: [Ho Chi Minh City, Downtown]	Actual: [Ho Chi Minh City, Downtown]	**PASSED**

Check if user with username "Jack" searches for list of items located around location "E"	username = "Jack"
	currentLocation = "E"
	name = []
		Expected: []	Actual: []	**PASSED**

Check if user with username "Jack" searches for list of items located around location "114"	username = "Jack"
	currentLocation = "114"
	name = []
		Expected: []	Actual: []	**PASSED**

Check if user with username "Jack" searches for list of items located around location <null>	username = "Jack"
	currentLocation = null
	name = []
		Expected: []	Actual: []	**PASSED**

3.2.4.5. Test Function readCityInfo()

Table 18: Test data for unit test of method readCityInfo() in the ContextManager module

Test Case	Test Data	Expected Results	Actual Results	Test Results

Read all information of Vivo City Shopping Centre from CityInfo text file.	index = 0
	name = "Vivo City Shopping Centre"
	location = "A"
	information = "Vivo City Shopping Centre is a major regional shopping centre in the southern suburb of Ho Chi Minh City, Vietnam. It is the second largest shopping centre in the southern suburbs of Ho Chi Minh City, by gross area, and
		Expect 1: Item Name is not null.	**Actual 1-4:** Same as the expected results. (this means that the CityInfo text file is read successfully)	**PASSED**

		Expect 2: Item Location is not null.		
		Expect 3: Item Info is not null.		
		Expect 4: Item Services are not null.		
		Expect 5: LocationDetails [name=Vivo City Shopping Centre, location=A, info=Vivo City Shopping Centre is a major regional shopping centre in the southern suburb of Ho Chi Minh City, Vietnam. It is the second largest	**Actual 5:** Same as the expected results.	

	contains the only H&M store in that region. services: cinema, restaurants, pool, shops, bowling”	shopping centre in the southern suburbs of Ho Chi Minh City, by gross area, and contains the only H&M store in that region., services=[cinema, restaurants, pool, shops, bowling]]		

Read all information of Crescent Mall from CityInfo text file.	index = 1
	name = “Crescent Mall”
	location = “B”
	information = “Crescent Mall Shopping Centre is located 10km South of the Ho Chi Minh City central business district(CBD) and includes Banana Republic, Baskin Robins, CGV Cinema, Bobapop and over 130 specialty stores.
	services: cinema, restaurants, shops”
		Expect 1: Item Name is not null.
		Expect 2: Item Location is not null.
		Expect 3: Item Info is not null.
		Expect 4: Item Services are not null.
			Actual 1-4: Same as the expected results. (this means that the CityInfo text file is read successfully)	**PASSED**

		Expect: LocationDetails [name=Crescent Mall, location=B, info=Crescent Mall Shopping Centre is located 10km South of the Ho Chi Minh City central business district(CBD) and includes Banana Republic, Baskin Robins, CGV Cinema, Bobapop and over 130 specialty stores., services=[cinema, restaurants, shops]]	**Actual 5:** Same as the expected results.	

Read all information of Dam Sen Parklands from CityInfo text file.	index = 2
	name = “Dam Sen Parklands”
	location = “C”
	information = “The Dam Sen Parklands area was created as part of the rejuvenation of the industrial upgrade undertaken for World Expo 1988. The Parklands area is spacious with plenty of green and spaces for all ages. A big lake promenade stretches the area of Dam Sen Parklands.
	services: restaurants, pool, shops, Ferris wheel”
		Expect 1: Item Name is not null.
		Expect 2: Item Location is not null.
		Expect 3: Item Info is not null.
		Expect 4: Item Services are not null.
			Actual 1-4: Same as the expected results. (this means that the CityInfo text file is read successfully)	**PASSED**

		Expect 5: LocationDetails [name=Dam Sen Parklands, location=C, info=The Dam Sen Parklands area was created as part of the rejuvenation of the industrial upgrade undertaken for World Expo 1988. The Parklands area is spacious with plenty of green and spaces for all ages. A big lake promenade stretches the area of Dam Sen Parklands., services=[restaurants, pool, shops, Ferris wheel]]	**Actual 5:** Same as the expected results.	

Read all information of Ho Chi Minh	index = 3
	name = “Ho Chi Minh City, Downtown”
		Expect 1: Item Name is not null.
		Expect 2: Item Location is not null.
			Actual 1-4: Same as the expected results.	**PASSED**

City, Downtown from CityInfo text file.	location = “D” information = ‘The Ho Chi Minh City central business district (CBD), or 'the City' is located on a central point in district One. The point, known at its tip as Central Point, slopes upward to the north-west where 'the city' is bounded by parkland and the inner city suburb of District 3, District 4 and District 5. services: restaurants, shops, market, bowling”	**Expect 3:** Item Info is not null.	(this means that the CityInfo text file is read successfully)	
		Expect 4: Item Services are not null.		
		Expect 5: LocationDetails [name=Ho Chi Minh City, Downtown, location=D, info=The Ho Chi Minh City central business district (CBD), or 'the City' is located on a central point in district One. The point, known at its tip as Central Point, slopes upward to the north-west where 'the city' is bounded by parkland and the inner city suburb of District 3, District 4 and District 5., services=[restaurants, shops, market, bowling]]	**Actual 5:** Same as the expected results.	

3.2.4.6. Test Function getLocationsByService(String service)

Table 19: Test data for unit test of method getLocationsByService(String service) in the ContextManager module

Test Case	Test Data	Expected Results	Actual Results	Test Results

Check if user can get the correct location(s) searched by services “cinema”.	services = “cinema”
	location = ["Vivo City Shopping Centre", "Crescent Mall"]
		Expect: [Vivo City Shopping Centre, Crescent Mall]	Actual: Same as the expected results.	**PASSED**

Check if user can get the correct location(s) searched by services “shops”.	services = “shops”
	location = ["Vivo City Shopping Centre", "Crescent Mall"]
		Expect: [Vivo City Shopping Centre, Crescent Mall]	Actual: Same as the expected results.	**PASSED**

Check if user can get the correct location(s) searched by services “bowling”.	services = “shops”
	location = ["Vivo City Shopping Centre"]
		Expect: [Vivo City Shopping Centre]	Actual: Same as the expected results.	**PASSED**

Check if user can get the correct location(s) searched by services “pool”.	services = “pool”
	location = ["Vivo City Shopping Centre"]
		Expect: [Vivo City Shopping Centre]	Actual: Same as the expected results.	**PASSED**

Check if user can get the correct location(s)	services = “restaurants”	Expect: [Vivo City Shopping Centre, Crescent Mall]	Actual: Same as the	**PASSED**

searched by services “restaurants”.	location = ["Vivo City Shopping Centre", "Crescent Mall"]		expected results.	

Check if user can get the correct location(s) searched by services “Ferris wheel”.	services = “Ferris wheel”
	location = []
		Expected: []	Actual: []	**PASSED**

Check if user can get the correct location(s) searched by services “CINEMA”.	services = “CINEMA”
	location = []
		Expected: []	Actual: []	**PASSED**

Check if user can get the correct location(s) searched by services “b=`sdf23//”.	services = “b=`sdf23//”
	location = []
		Expected: []	Actual: []	**PASSED**

Check if user can get the correct location(s) searched by services “<>”	services = “<>”
	location = []
		Expected: []	Actual: []	**PASS**

Check if user can get the correct location(s) searched by services <null>	services = null
	location = []
		Expected: []	Actual: []	**PASS**

3.2.4.7. Test Function resetClock(String username) and tickClock(String username)

Table 20: Test data for unit tests of resetClock(String username) and tickClock(String username) in the ContextManager module

Test Case	Test Data	Expected Results	Actual Results	Test Results
With username “Jack”, check if clock resets back to 0 after resetClock() function has been called.	username = “Jack”	Expect: 0	Actual: 0	**PASSED**
With username “Jack”, check if clock resets back to 0 after resetClock() function has been called.	username = “David”	Expect: 0	Actual: 0	**PASSED**
With username “Jack”, check if clock counts by 1 after tickClock() function has been called.	username = “Jack”	Expect: 1	Actual: 1	**PASSED**
With username “Jack”, check if clock counts by 1 after tickClock() function has been called.	username = “David”	Expect: 1	Actual: 1	**PASSED**

3.2.4.8. Test Function checkapoReached(User user)

Table 21: Test data for unit test of checkapoReached(User user) in the ContextManager module

Test Case	Test Data	Expected Results	Actual Results	Test Results

With username "Jack", check the APO threshold reached when clock value is smaller than the APO threshold value.	username = "Jack"
	apo_threshold = 30
	clock = 20
		Expect: false	Actual: false (this means that APO threshold is not reached)	**PASSED**

With username "Jack", check the APO threshold reached when clock value is equal to the APO threshold value.	username = "Jack"
	apo_threshold = 20
	clock = 20
		Expect: true	Actual: true (this means that APO threshold is reached)	**PASSED**

With username "Jack", check the APO threshold reached when clock value is greater than the APO threshold value.	username = "Jack"
	apo_threshold = 10
	clock = 20
		Expect: true	Actual: false (this means that APO threshold is not reached)	**FAILED**

With username "David", check the APO threshold reached when clock value is smaller than the APO threshold value.	username = "David"
	apo_threshold = 90
	clock = 30
		Expect: false	Actual: false (this means that APO threshold is not reached)	**PASSED**

With username "David", check the APO threshold reached when clock value is equal to the APO threshold value.	username = "David"
	apo_threshold = 30
	clock = 30
		Expect: true	Actual: true (this means that APO threshold is reached)	**PASSED**

With username "David", check the APO threshold reached when clock value is greater than the APO threshold value.	username = "David"
	apo_threshold = 15
	clock = 30
		Expect: true	Actual: false (this means that APO threshold is not reached)	**FAILED**

3.2.4.9. Test Function checkTempReached (User user)

Table 22: Test data for unit test of checkTempReached(User user) in the ContextManager module

Test Case	Test Data	Expected Results	Actual Results	Test Results

With username "Jack", check if the temperature	username = "Jack"
	temp_threshold = 30, 20
		Expect: true	Actual: false	**FAILED**

threshold reached when current temperature is smaller than the minimum threshold.	current_temp = 19
	temp_reached = true
			(this means that temperature threshold is not reached)	

With username “Jack”, check if the temperature threshold reached when current temperature is greater than the maximum threshold.	username = “Jack”
	temp_threshold = 30, 20
	current_temp = 31
	temp_reached = true
		Expect: true	Actual: true (this means that temperature threshold is reached)	**PASSED**

With username “Jack”, check if the temperature threshold reached when current temperature is in between the range of threshold.	username = “Jack”
	temp_threshold = 29, 15
	current_temp = 20
	temp_reached = false
		Expect: false	Actual: true (this means that APO threshold is not reached)	**FAILED**

With username “Jack”, check if the temperature threshold reached when current temperature is equal to the minimum threshold.	username = “Jack”
	temp_threshold = 29, 15
	current_temp = 15
	temp_reached = true
		Expect: true	Actual: true (this means that APO threshold is not reached)	**PASSED**

With username “Jack”, check if the temperature threshold reached when current temperature is equal to the maximum threshold.	username = “Jack”
	temp_threshold = 29, 15
	current_temp = 29
	temp_reached = true
		Expect: true	Actual: true (this means that APO threshold is reached)	**PASSED**

3.2.4.10. Test Function calculateapoThreshhold(User user)

Table 23: Test data for unit test of calculateapoThreshhold (User user) in the ContextManager module

Test Case	Test Data	Expected Results	Actual Results	Test Results

Check APO threshold calculation when current AQI is 0 and medical condition type is 1.	medical_type = 1
	current_aqi = 0
	expectedResult = 30
		Expect: 30	Actual: 5	**FAILED**

Check APO threshold calculation when current AQI is 50 and medical condition type is 1.	medical_type = 1
	current_aqi = 50
	expectedResult = 30
		Expect: 30	Actual: 30	**PASSED**

Check APO threshold calculation when current AQI is 0 and medical condition type is 2.	medical_type = 2
	current_aqi = 0
	expectedResult = 60
		Expect: 60	Actual: 10	**FAILED**

Check APO threshold calculation when current AQI is 50 and medical condition type is 2.	medical_type = 2
	current_aqi = 50
	expectedResult = 60
		Expect: 60	Actual: 60	**PASSED**

Check APO threshold calculation when current AQI is 0 and medical condition type is 3.	medical_type = 3
	current_aqi = 0
	expectedResult = 90
		Expect: 90	Actual: 15	**FAILED**

Check APO threshold calculation when current AQI is 50 and medical condition type is 3.	medical_type = 3
	current_aqi = 50
	expectedResult = 90
		Expect: 90	Actual: 90	**PASSED**

Check APO threshold calculation when current AQI is 51 and medical condition type is 1.	medical_type = 1
	current_aqi = 51
	expectedResult = 15
		Expect: 15	Actual: 15	**PASSED**

Check APO threshold calculation when current AQI is 100 and medical condition type is 1.	medical_type = 1
	current_aqi = 100
	expectedResult = 15
		Expect: 15	Actual: 15	**PASSED**

Check APO threshold calculation when current AQI is 51 and medical condition type is 2.	medical_type = 2
	current_aqi = 51
	expectedResult = 30
		Expect: 30	Actual: 30	**PASSED**

Check APO threshold calculation when current AQI is 100 and medical condition type is 2.	medical_type = 2
	current_aqi = 100
	expectedResult = 30
		Expect: 30	Actual: 30	**PASSED**

Check APO threshold calculation when current AQI is 51 and medical condition type is 3.	medical_type = 3
	current_aqi = 51
	expectedResult = 45
		Expect: 45	Actual: 45	**PASSED**

Check APO threshold calculation when current AQI is 100 and medical condition type is 3.	medical_type = 3
	current_aqi = 100
	expectedResult = 45
		Expect: 45	Actual: 45	**PASSED**

Check APO threshold calculation when current AQI is 101 and medical condition type is 1.	medical_type = 1
	current_aqi = 101
	expectedResult = 10
		Expect: 10	Actual: 10	**PASSED**

Check APO threshold calculation when current AQI is 150 and medical condition type is 1.	medical_type = 1
	current_aqi = 150
	expectedResult = 10
		Expect: 10	Actual: 10	**PASSED**

Check APO threshold calculation when current AQI is 101 and medical condition type is 2.	medical_type = 2
	current_aqi = 101
	expectedResult = 20
		Expect: 20	Actual: 20	**PASSED**

Check APO threshold calculation when current AQI is 150 and medical condition type is 2.	medical_type = 2
	current_aqi = 150
	expectedResult = 20
		Expect: 20	Actual: 20	**PASSED**

Check APO threshold calculation when current AQI is 101 and medical condition type is 3.	medical_type = 3
	current_aqi = 101
	expectedResult = 30
		Expect: 30	Actual: 30	**PASSED**

Check APO threshold calculation when current AQI is 150 and medical condition type is 3.	medical_type = 3
	current_aqi = 150
	expectedResult = 30
		Expect: 30	Actual: 30	**PASSED**

Check APO threshold calculation when current AQI is 151 and medical condition type is 1.	medical_type = 1
	current_aqi = 151
	expectedResult = 5
		Expect: 5	Actual: 5	**PASSED**

Check APO threshold calculation when current AQI is 200 and medical condition type is 1.	medical_type = 1
	current_aqi = 200
	expectedResult = 5
		Expect: 5	Actual: 5	**PASSED**

Check APO threshold calculation when current AQI is 151 and medical condition type is 2.	medical_type = 2
	current_aqi = 151
	expectedResult = 10
		Expect: 10	Actual: 10	**PASSED**

Check APO threshold calculation when current AQI is 200 and medical condition type is 2.	medical_type = 2
	current_aqi = 200
	expectedResult = 10
		Expect: 10	Actual: 10	**PASSED**

Check APO threshold calculation when current AQI is 151 and medical condition type is 3.	medical_type = 3
	current_aqi = 151
	expectedResult = 15
		Expect: 15	Actual: 15	**PASSED**

Check APO threshold calculation when current AQI is 200 and medical condition type is 3.	medical_type = 3
	current_aqi = 200
	expectedResult = 15
		Expect: 15	Actual: 15	**PASSED**

Check APO threshold calculation when current AQI is -1 and medical condition type is 1.	medical_type = 1
	current_aqi = -1
	expectedResult = null
		Expect: null	Actual: 5	**FAILED**

Check APO threshold calculation when current AQI is 201 and medical condition type is 1.	medical_type = 1
	current_aqi = 201
	expectedResult = null
		Expect: null	Actual: 5	**FAILED**

Check APO threshold calculation when current AQI is 50 and medical condition type is 4.	medical_type = 4
	current_aqi = 50
	expectedResult = null
		Expect: null	Actual: 120	**FAILED**

Check APO threshold calculation when current AQI is 50 and medical condition type is 0.	medical_type = 0
	current_aqi = 50
	expectedResult = null
		Expect: null	Actual: 0	**FAILED**

3.2.5. AllSensors.java

The table below shows the available functions that are required to be tested in the AllSensors java file:

Table 24: List of tested function(s)

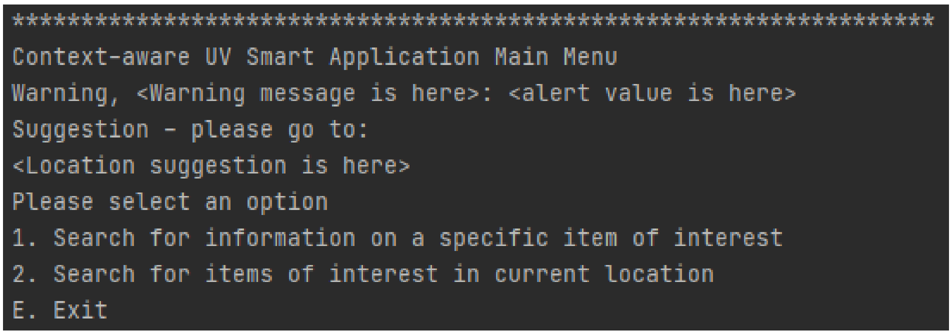
Name	Return Type	Description
getCurrentValue()	Void	Get the current value taken from the sensor

Table 25: Test Function getCurrentValue()

Title	Test Function getCurrentValue()
Description	Function getCurrentValue() should be successfully called and return the current value from the sensor.
Preconditions	+ AllSensors class starts up properly. + Text files exist and have corresponding content.
Test Data	JackLocation.txt: A, 1 C, 15 D, 14
Steps	+ Use sensor.getCurrentValue() to get the current value. + Compare with the expected value using assertEquals() function.
Expected results	A
Actual results	A
Test results	PASSED

3.2.6. EnviroAPPUI.java

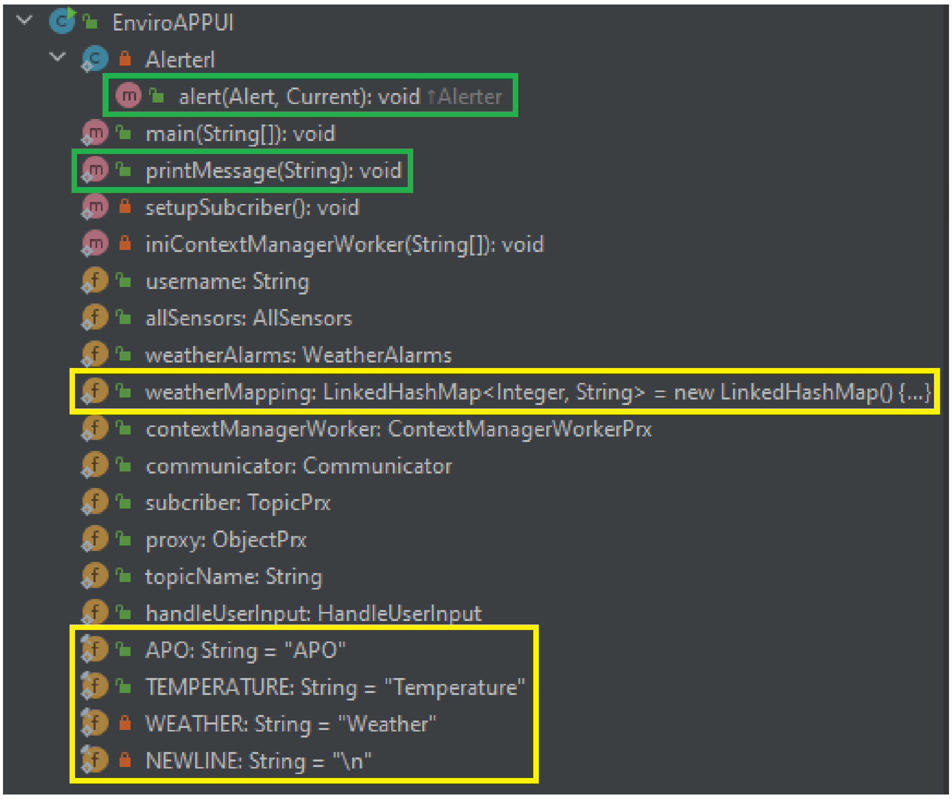
The main objective in this test is to successfully execute a numbers of expected warning messages which strictly follows the output format shown in Figure below.



```
**********************************************************************
Context-aware UV Smart Application Main Menu
Warning, <Warning message is here>: <alert value is here>
Suggestion - please go to:
<Location suggestion is here>
Please select an option
1. Search for information on a specific item of interest
2. Search for items of interest in current location
E. Exit
```

Figure 13: Format of the screen when receiving warning message and location suggestion

Furthermore, the priority order is one of the main requirements that we need to consider when printing out the warning message as well as give suggestion to users. Hence, we are going to make use of 2 methods in the EnviroAPPUI class: ***printMessage(String message)*** and ***alert(Alert alert, Current current)***.



```
EnviroAPPUI
    AlerterI
        alert(Alert, Current): void ↑Alerter
    main(String[]): void
    printMessage(String): void
    setupSubcriber(): void
    iniContextManagerWorker(String[]): void
    username: String
    allSensors: AllSensors
    weatherAlarms: WeatherAlarms
    weatherMapping: LinkedHashMap<Integer, String> = new LinkedHashMap() {...}
    contextManagerWorker: ContextManagerWorkerPrx
    communicator: Communicator
    subcriber: TopicPrx
    proxy: ObjectPrx
    topicName: String
    handleUserInput: HandleUserInput
    APO: String = "APO"
    TEMPERATURE: String = "Temperature"
    WEATHER: String = "Weather"
    NEWLINE: String = "\n"
```

Figure 14: Structure of EnviroAPPUI class

- *alert(Alert, Current):* identify the alert type whether it is APO, temperature or weather alert and give the corresponding warning message with the alert value and suggestion.
- *printMessage(String, Integer):* print out the text-based system with the warning message and suggestion, along with the 2 options of searching for specific item's info or list of items in the current location.

However, since the alert() method is associated with multiple files in the helper package which belongs to the IceStorm (a mediator between message publishers and message subscribers as mentioned in the assessment details), it is much complicated for us to generate the access and get imitation of all necessary variables existed in those files for completely executing unit test with real data. Therefore, we have brought them into our JUnit test file and made some small changes in term of the input variables based on the logic of the initial methods from the EnviroAPPUI class. Moreover, we also generate several collections of test data according to the preference of 2 users with usernames "Jack" and "David", respectively. In order to have an insight to these test data, please refer to following table.

Table 26: Test data for unit tests of printMessage(String) and alert(Alert, Current) in the ContextManager module

Test Case	Test Data	Expected Results	Actual Results	Test Results

Check if the app UI prints the correct message corresponding with the test data and the priority order among sensors and alarm values.	alertType = “Weather”
	aqiValue = 163
	tempValue = 10
	weatherValue = “strong wind”
	suggestedLocation = “Crescent Mall", "Dam Sen Parklands”
		Expect: Please refer to the Appendix B	Actual: Please refer to the Appendix B	**PASSED**

Check if the app UI prints the correct message corresponding with the test data and the priority order among sensors and alarm values.	alertType = “Weather”
	aqiValue = 163
	tempValue = 10
	weatherValue = “hail storm”
	suggestedLocation = “Crescent Mall”, “Ho Chi Minh, Downtown”
		Expect: Please refer to the Appendix B	Actual: Please refer to the Appendix B	**PASSED**

Check if the app UI prints the correct message corresponding with the test data and the priority order among sensors and alarm values.	alertType = “Weather”
	aqiValue = 163
	tempValue = 10
	weatherValue = “heavy rain”
	suggestedLocation = “Dam Sen Parklands”, “Ho Chi Minh, Downtown”
		Expect: Please refer to the Appendix B	Actual: Please refer to the Appendix B	**PASSED**

Check if the app UI prints the correct message corresponding with the test data and the priority order among sensors and alarm values.	alertType = “Temperature”
	aqiValue = 27
	tempValue = 10
	weatherValue = “normal”
	suggestedLocation = “Crescent Mall", "Dam Sen Parklands”
		Expect: Please refer to the Appendix B	Actual: Please refer to the Appendix B	**PASSED**

Check if the app UI prints the correct message corresponding with the test data and the priority order among sensors and alarm values.	alertType = “Temperature”
	aqiValue = 27
	tempValue = 15
	weatherValue = “normal”
	suggestedLocation = “Crescent Mall”, “Ho Chi Minh, Downtown”
		Expect: Please refer to the Appendix B	Actual: Please refer to the Appendix B	**PASSED**

Check if the app UI prints the correct message corresponding with the test data and the priority order among sensors and alarm values.	alertType = “Temperature”
	aqiValue = 27
	tempValue = 20
	weatherValue = “normal”
	suggestedLocation = “Ho Chi Minh, Downtown”
		Expect: Please refer to the Appendix B	Actual: Please refer to the Appendix B	**PASSED**

Check if the app UI prints the correct message corresponding with the test data and the priority order among sensors and alarm values.	alertType = “Temperature”
	aqiValue = 27
	tempValue = 25
	weatherValue = “normal”
	suggestedLocation = “Dam Sen Parklands”, “Ho Chi Minh, Downtown”
		Expect: Please refer to the Appendix B	Actual: Please refer to the Appendix B	**PASSED**

Check if the app UI prints the correct message corresponding with the test data and the priority order among sensors and alarm values.	alertType = “APO”
	aqiValue = 163
	tempValue = 25
	weatherValue = “normal”
	suggestedLocation = “Crescent Mall", "Dam Sen Parklands”
		Expect: Please refer to the Appendix B	Actual: Please refer to the Appendix B	**PASSED**

Check if the app UI prints the correct message corresponding with the test data and the priority order among sensors and alarm values.	alertType = “APO”
	aqiValue = 105
	tempValue = 25
	weatherValue = “normal”
	suggestedLocation = “Crescent Mall”, “Ho Chi Minh, Downtown”
		Expect: Please refer to the Appendix B	Actual: Please refer to the Appendix B	**PASSED**

Check if the app UI prints the correct message corresponding with the test data and the priority order among sensors and alarm values.	alertType = “APO”
	aqiValue = 58
	tempValue = 25
	weatherValue = “normal”
	suggestedLocation = "Dam Sen Parklands”, “Ho Chi Minh, Downtown”
		Expect: Please refer to the Appendix B	Actual: Please refer to the Appendix B	**PASSED**

4. Integration Test

Combined with Figure 4, the following table illustrates the testing combinations of the modules and low-level components:

Modules	Integrated components
AllSensors.java	+ Temperature text file + AQI text file + Location text file + Context Manager module
LocationServer.java	+ Location server configuration text file + Context Manager module
PreferenceRepository.java	+ Context Manager module
EnviroAPPUI.java	+ Preference text file + Context Manager module
ContextManager.java	+ City Information text file + Weather Alarm module + All Sensors module + Location Server module + Preference Repository module + EnviroApp module
WeatherAlarm.java	+ Weather alarms text file + Context Manager module

It is noteworthy to mention that most of the integration test cases below cannot be conducted as test case using JUnit due to Ice Platform restrictions. For those that can be tested through coding, an additional row is added to the test case, indicating the test result as “PASSED” or “FAILED”.

4.1. Integration between text files and modules

4.1.1. <username>Temperature.txt and AllSensors.java Module

Table 27: AllSensors.java module reads values with correct format from the temperature text file.

Title	AllSensors.java module reads values with correct format from the temperature text file.
Description	Check if the AllSensors.java module could successfully read values from the temperature text file <username>Temperature.txt with correct defined format <value, number of seconds>
Preconditions	+ User has logged into the application with their account.

	+ The temperature sensor has started in Eclipse in AllSensors.java module.
Test Data	+ Username: Jack + JackTemperature.txt: 10, 5 15, 3 20, 4
Steps	Invoke the private method getSensorData() in SensorData.class
Expected results	AllSensors.java module receives the following data: 10, 5 15, 3 20, 4
Actual results	AllSensors.java module receives the following data: 10, 5 15, 3 20, 4
Test Results	**PASSED**

Table 28: AllSensors.java module reads empty value from the temperature text file

Title	AllSensors.java module reads empty value from the temperature text file
Description	Check if the AllSensors.java module is able to read values from the temperature text file <username>Temperature.txt using the private function getSensorData()
Preconditions	+ User has logged into the application with their account. + The temperature sensor has started in Eclipse in AllSensors.java module.
Test Data	N/A
Steps	Invoke the private method getSensorData() in SensorData.class
Expected results	AllSensors.java module receives the following data: <blank>
Actual results	AllSensors.java module receives the following data: <blank>
Test Results	**PASSED**

Table 29: AllSensors.java module loops back to the beginning of the temperature text file.

Title	AllSensors.java module loops back to the beginning of the temperature text file after reaching the end of the file.
Description	Check if the AllSensors.java module is able to return to the beginning of the temperature text file <username>Temperature.txt after reaching the end of the file.
Preconditions	+ User has logged into the application with their account.

	+ The temperature sensor has started in Eclipse in AllSensors.java module.
Test Data	+ Username: Jack + JackTemperature.txt: 10, 5 15, 3 20, 4
Steps	Invoke the private method getSensorData() in SensorData.class
Expected results	AllSensors.java module receives the following data: 10, 5 15, 3 20, 4 After the end of file is reached, AllSensors.java module will receive the exact same data from the beginning, i.e., 26 degree Celsius for 40 seconds and continues as stated above.
Actual results	AllSensors.java module receives the following data: 10, 5 15, 3 20, 4 After the end of file is reached, AllSensors.java module will receive the exact same data from the beginning, i.e., 10 degree Celsius for 5 seconds and continues as stated above.
Test Results	**PASSED**

4.1.2. <username>AQI.txt and AllSensors.java Module

Table 30: AllSensors.java module reads values with correct format from the AQI text file.

Title	AllSensors.java module reads values with correct format from the AQI text file.
Description	Check if the AllSensors.java module could successfully read values from the AQI text file <username>AQI.txt with correct defined format <value, number of seconds>
Preconditions	+ User has logged into the application with their account. + The AP sensor has started in Eclipse in AllSensors.java module.
Test Data	+ Username: Jack + JackAQI.txt: 200, 15 90, 11
Steps	Invoke the private method getSensorData() in SensorData.class
Expected results	AllSensors.java module receives the following data: 200, 15 90, 11

Actual results	AllSensors.java module receives the following data: 200, 15 90, 11
Test Results	**PASSED**

Table 31: AllSensors.java module reads empty value from the AQI text file

Title	AllSensors.java module reads empty value from the AQI text file
Description	Check if the AllSensors.java module is able to read values from the AQI text file <username>AQI.txt
Preconditions	+ User has logged into the application with their account. + The AP sensor has started in Eclipse in AllSensors.java module.
Test Data	N/A
Steps	AllSensors.java module starts reading the context in JackAQI.txt by individual lines.
Expected results	AllSensors.java module does not receive any data
Test Results	**PASSED**

Table 32: AllSensors.java module loops back to the beginning of the AQI text file.

Title	AllSensors.java module loops back to the beginning of the AQI text file after reaching the end of the file.
Description	Check if the AllSensors.java module is able to return to the beginning of the AQI text file <username>AQI.txt after reaching the end of the file.
Preconditions	+ User has logged into the application with their account. + The AP sensor has started in Eclipse in AllSensors.java module.
Test Data	+ Username: Jack + JackAQI.txt: 200, 15 90, 11
Steps	Invoke the private method getSensorData() in SensorData.class
Expected results	AllSensors.java module receives the following data: 200, 15 90, 11 After the end of file is reached, AllSensors.java module will receive the exact same data from the beginning, i.e., 200 AQI for 15 seconds and continues as stated above.
Actual results	AllSensors.java module receives the following data: 200, 15 90, 11

	After the end of file is reached, AllSensors.java module will receive the exact same data from the beginning, i.e., 200 AQI for 15 seconds and continues as stated above.
Test Results	**PASSED**

4.1.3. <username>Location.txt and AllSensors.java Module

Table 33: AllSensors.java module reads values with correct format from the location text file.

Title	AllSensors.java module reads values with correct format from the location text file.
Description	Check if the AllSensors.java module could successfully read values from the location text file <username>Location.txt with correct defined format <value, number of seconds>
Preconditions	+ User has logged into the application with their account. + The location sensor has started in Eclipse in AllSensors.java module.
Test Data	+ Username: Jack + JackLocation.txt: A, 1 C, 15 D, 14
Steps	Invoke the private method getSensorData() in SensorData.class
Expected results	AllSensors.java module receives the following data: A, 1 C, 15 D, 14
Actual results	AllSensors.java module receives the following data: A, 1 C, 15 D, 14
Test Results	**PASSED**

Table 34: AllSensors.java module reads empty value from the location text file

Title	AllSensors.java module reads empty value from the location text file
Description	Check if the AllSensors.java module is able to read values from the location text file <username>Location.txt
Preconditions	+ User has logged into the application with their account. + The location sensor has started in Eclipse in AllSensors.java module.
Test Data	N/A
Steps	Invoke the private method getSensorData() in SensorData.class
Expected results	AllSensors.java module receives the following data: <blank>

Actual results	AllSensors.java module receives the following data: <blank>
Test Results	**PASSED**

Table 35: AllSensors.java module loops back to the beginning of the location text file

Title	AllSensors.java module loops back to the beginning of the location text file after reaching the end of the file.
Description	Check if the AllSensors.java module is able to return to the beginning of the location text file <username>Location.txt after reaching the end of the file.
Preconditions	+ User has logged into the application with their account. + The location sensor has started in Eclipse in AllSensors.java module.
Test Data	+ Username: Jack + JackLocation.txt: A, 1 C, 15 D, 14
Steps	Invoke the private method getSensorData() in SensorData.class
Expected results	AllSensors.java module receives the following data: A, 1 C, 15 D, 14 After the end of file is reached, AllSensors.java module will receive the exact same data from the beginning, i.e., Location A for 1 seconds and continues as stated above.
Actual results	AllSensors.java module receives the following data: A, 1 C, 15 D, 14 After the end of file is reached, AllSensors.java module will receive the exact same data from the beginning, i.e., Location A for 1 seconds and continues as stated above.
Test Results	**PASSED**

4.1.4. [configuration-file].txt and LocationServer.java Module

Table 36: LocationServer.java module reads values with correct structure from the configuration text file

Title	LocationServer.java module reads values with correct structure from the configuration text file

Description	Check if the LocationServer.java module could successfully read values from the configuration text file [configuration-file].txt with correct defined structure: Status: Location Coordinates Indoor: A, B, Outdoor: C, D,
Preconditions	User has logged into the application with their account.
Test Data	N/A
Steps	Invoke the method readConfig() in LocationServer.class
Expected results	LocationServer.java module receives the following data: Indoor: A, B Outdoor: C, D
Actual results	LocationServer.java module receives the following data: Indoor: A, B Outdoor: C, D
Test Results	**PASSED**

Table 37: LocationServer.java module reads empty values from the configuration text file

Title	LocationServer.java module reads empty values from the configuration text file
Description	Check if the LocationServer.java module is able to read values from the configuration text file [configuration-file].txt
Preconditions	+ User has logged into the application with their account. + The configuration text file is not yet written.
Test Data	N/A
Steps	Invoke the method readConfig() in LocationServer.class
Expected results	The LocationServer.java module receives the following data: <blank>
Actual results	The LocationServer.java module receives the following data: <blank>
Test Results	**PASSED**

4.1.5. [preference-file].txt and PreferenceRepository.java Module

Table 38: PreferenceRepository.java module reads complete preference values with correct structure from the preference text file

Title	PreferenceRepository.java module reads complete preference values with correct structure from the preference text file
Description	Check if the PreferenceRepository.java module could successfully read values from the preference text file [preference-file].txt that contains the following attributes:

	1/ Name 2/ Medical condition type 3/ Temperature preference 4/ APO preference 5/ Weather preference
Preconditions	User has logged into the application with their account
Test Data	name: Jack Medical Condition Type: 2 pref: when 20 suggest shops pref: when 30 suggest pool pref: when APO suggest bowling pref: when weather suggest cinema
Steps	Invoke the method readPreference () in PreferenceRepository.class
Expected results	PreferenceRepository.java module receives the following data: name: Jack Medical Condition Type: 2 pref: when 20 suggest shops pref: when 30 suggest pool pref: when APO suggest bowling pref: when weather suggest cinema
Actual results	PreferenceRepository.java module receives the following data: name: Jack Medical Condition Type: 2 pref: when 20 suggest shops pref: when 30 suggest pool pref: when APO suggest bowling pref: when weather suggest cinema
Test Results	**PASSED**

Table 39: PreferenceRepository.java module reads empty preference values from the preference text file

Title	PreferenceRepository.java module reads empty preference values from the preference text file
Description	Check if the PreferenceRepository.java module is able to read values from the preference text file [preference-file].txt if user has not specified any preferences
Preconditions	+ User has logged into the application with their account. + The preference text file is not yet written.
Test Data	Username: Jack
Steps	Invoke the method readPreference () in PreferenceRepository.class

Expected results	PreferenceRepository.java module receives the following data: <blank>
Actual results	PreferenceRepository.java module receives the following data: <blank>
Test Results	**PASSED**

Table 40: PreferenceRepository.java module reads incomplete preference values from the preference text file

Title	PreferenceRepository.java module reads incomplete preference values from the preference text file
Description	Check if the PreferenceRepository.java module is able to read values from the preference text file [preference-file].txt when user has specified at least one preference
Preconditions	User has logged into the application with their account.
Test Data	name: Jack Medical Condition Type: 2 pref: when 20 suggest shops pref: when 30 suggest pref: when APO suggest pref: when weather suggest
Steps	Invoke the method readPreference () in PreferenceRepository.class
Expected results	The PreferenceRepository.java only acknowledges the temperature preference (20 Celcius degree) and a temperature warning is sent to the user if and only if the temperature threshold is breached. The others will be considered as null.
Actual results	The PreferenceRepository.java only acknowledges the temperature preference (20 Celcius degree) and a temperature warning is sent to the user if and only if the temperature threshold is breached. The others will be considered as null.
Test Results	**PASSED**

4.1.6. weather_alarm.txt and WeatherAlarm.java Module

Table 41: WeatherAlarm.java module sends alarm events to the weather alarm text file

Title	WeatherAlarm.java module reads alarm events from the weather alarm text file
Description	Check if the WeatherAlarm.java module receives weather alarm events from the weather alarm text file weather_alarm.txt (weather condition is abnormal).
Preconditions	+ User has logged into the application with their account.

	+ Weather condition is defined as not normal.
Test Data	+ Username: Jack + Weather: 1 (heavy rain)
Steps	Invoke the method readWeatherConditions() in WeatherAlarms.class
Expected results	WeatherAlarms.java module receives the weather condition of 1
Actual results	WeatherAlarms.java module receives the weather condition of 1
Test Results	**PASSED**

Table 42: WeatherAlarm.java module does not send alarm events to the weather alarm text file

Title	WeatherAlarm.java module does not receive alarm events from the weather alarm text file
Description	Check if the WeatherAlarm.java module does not receive weather alarm events from the weather alarm text file weather_alarm.txt when the condition is normal
Preconditions	+ User has logged into the application with their account. + Weather condition is defined as normal.
Test Data	+ Username: Jack + Weather: 0 (normal)
Steps	Invoke the method readWeatherConditions() in WeatherAlarms.class
Expected results	WeatherAlarms.java module receives the following data: <blank>
Actual results	WeatherAlarms.java module receives the following data: <blank>
Test Results	**PASSED**

4.1.7. [city-information-file].txt and ContextManager.java Module

Table 43: ContextManager.java module receives city information values from the city information text file

Title	ContextManager.java module receives city information values from the city information text file
Description	Check if the ContextManager.java module successfully receives city information values from the city information text file [city-information-file].txt
Preconditions	User has logged into the application with their account.
Test Data	+ Name: Vivo City Shopping Centre + Location: A + Information: Vivo City Shopping Centre is a major regional shopping centre in the southern suburb of Ho Chi Minh City, Vietnam. It is the

	second largest shopping centre in the southern suburbs of Ho Chi Minh City, by gross area, and contains the only H&M store in that region. + Services: cinema, restaurants, pool, shops, bowling
Steps	Invoke the method readCityInfo() in ContextManager.class
Expected results	The ContextManager.java module will display the following data when requested: name: Vivo City Shopping Centre location: A information: Vivo City Shopping Centre is a major regional shopping centre in the southern suburb of Ho Chi Minh City, Vietnam. It is the second largest shopping centre in the southern suburbs of Ho Chi Minh City, by gross area, and contains the only H&M store in that region. services: cinema, restaurants, pool, shops, bowling
Actual results	The ContextManager.java module will display the following data when requested: name: Vivo City Shopping Centre location: A information: Vivo City Shopping Centre is a major regional shopping centre in the southern suburb of Ho Chi Minh City, Vietnam. It is the second largest shopping centre in the southern suburbs of Ho Chi Minh City, by gross area, and contains the only H&M store in that region. services: cinema, restaurants, pool, shops, bowling
Test Results	**PASSED**

Table 44: ContextManager.java module receives empty city information values from the city information text file

Title	ContextManager.java module receives empty city information values from the city information text file
Description	Check if the ContextManager.java module receives empty city information values from the city information text file [city-information-file].txt
Preconditions	+ User has logged into the application with their account. + The city information text file is not yet written.
Test Data	N/A
Steps	Invoke the method readCityInfo() in ContextManager.class
Expected results	The ContextManager.java receives the following data: <blank>
Actual results	The ContextManager.java receives the following data: <blank>
Test Results	**PASSED**

4.2. Integration between modules

4.2.1. WeatherAlarm.java Module and ContextManager.java Module

Table 45: Evaluate the behavior of ContextManager.java module when the WeatherAlarm.java module detects the weather is normal

Title	Evaluate the behaviour of ContextManager.java module when the WeatherAlarm.java module detects the weather is normal
Description	Check if the WeatherAlarm.java module does not send any weather alarm notifications to the ContextManager.java when the weather condition is normal, regardless of the indoor/outdoor status of the user.
Preconditions	+ User has logged into the application with their account. + Context Manager has received the indoor/outdoor status of the user from Location Server. + User has defined the preferences.
Test Data	Weather Condition: normal
Steps	+ Place the device either indoor or outdoor. + Wait for WeatherAlarm.java module to start monitoring the weather condition.
Expected results	The WeatherAlarm.java will not send any notifications to the ContextManager.java
Actual results	The WeatherAlarm.java will not send any notifications to the ContextManager.java
Test Results	**PASSED**

Table 46: Evaluate the behavior of ContextManager.java module when the WeatherAlarm.java module detects the weather is abnormal

Title	Evaluate the behaviour of ContextManager.java module when the WeatherAlarm.java module detects the weather is abnormal
Description	Check if the WeatherAlarm.java module sends a weather alarm notification to the ContextManager.java when the weather condition is abnormal
Preconditions	+ User has logged into the application with their account. + Context Manager has received the indoor/outdoor status of the user from Location Server. + User has defined the preferences.
Test Data	+ Weather Condition: heavy rain + Preference Service: mall
Steps	+ Place the device either indoor or outdoor. + Wait for WeatherAlarm.java module to start monitoring the weather condition.

Expected results	The WeatherAlarm.java will send an extreme weather notification to the ContextManager.java module
Actual results	The WeatherAlarm.java will send an extreme weather notification to the ContextManager.java module
Test Results	**PASSED**

Table 47: ContextManager.java receives readings from WeatherAlarm.java module every 60 seconds.

Title	ContextManager.java receives readings from WeatherAlarm.java module every 60 seconds.
Description	Check if the ContextManager.java module could successfully receive weather readings from the WeatherAlarm.java module for every 60 seconds.
Preconditions	+ User has logged into the application with their account.
Test Data	+ Weather Condition: hail storm + Sent time: 12:00 A.M
Steps	ContextManager.java reads the context from WeatherAlarm.java module for every 60 seconds
Expected results	The WeatherAlarm.java sends a hail storm notification to the ContextManager.java module at 12:01 A.M
Actual results	The WeatherAlarm.java sends a hail storm notification to the ContextManager.java module at 12:01 A.M
Test Results	**PASSED**

Table 48: ContextManager.java does not receives reading from WeatherAlarm.java module every 60 seconds.

Title	ContextManager.java does not receives reading from WeatherAlarm.java module every 60 seconds.
Description	The ContextManager.java module does not receive weather readings from the WeatherAlarm.java module for every 60 seconds.
Preconditions	+ User has logged into the application with their account. + Weather Alarm is malfunctioned, or the data transferal is blocked.
Test Data	+ Weather Condition: hail storm + Sent time: 12:00 A.M
Steps	ContextManager.java reads the context from WeatherAlarm.java module for every 60 seconds
Expected results	The WeatherAlarm.java does not send any notifications to the ContextManager.java
Actual results	The WeatherAlarm.java does not send any notifications to the ContextManager.java
Test Results	**PASSED**

4.2.2. AllSensors.java Module and ContextManager.java Module

Table 49: ContextManager.java receives values from AllSensors.java module

Title	ContextManager.java receives values from AllSensors.java module
Description	Check if the ContextManager.java module could successfully receive values from the AllSensors.java module
Preconditions	+ User has logged into the application with their account. + Temperature sensor, location sensor and AP sensor has started. + AllSensors.java has successfully retrieved sensor data.
Test Data	+ Temperature value: 25 + AQI Index: 150 + Location: A
Steps	ContextManager.java starts reading the context in AllSensors.java module
Expected results	The ContextManager.java receives the following data for threshold evaluation: + Temperature: 25 + AQI Index: 150 + Location: A
Actual results	The ContextManager.java receives the following data for threshold evaluation: + Temperature: 25 + AQI Index: 150 + Location: A
Test Results	**PASSED**

Table 50: ContextManager.java receives empty values from AllSensors.java module

Title	ContextManager.java receives empty values from AllSensors.java module
Description	Check if the ContextManager.java module is able to read values from AllSensors.java module
Preconditions	+ User has logged into the application with their account. + Temperature sensor, location sensor and AP sensor has started.
Test Data	+ Temperature value: N/A + AQI Index: N/A + Location: N/A
Steps	ContextManager.java starts reading the context in AllSensors.java module
Expected results	The ContextManager.java module does not receive any data
Actual results	The ContextManager.java module does not receive any data
Test Results	**PASSED**

4.2.3. LocationServer.java Module and ContextManager.java Module

Table 51: The ContextManager.java module receives values from the LocationServer.java module

Title	The ContextManager.java module receives values from the LocationServer.java module
Description	Check if the ContextManager.java module receives values from the LocationServer.java module upon request
Preconditions	+ User has logged into the application with their account. + Location Server successfully retrieve data from the [configuration-file].txt
Test Data	A list of indoor locations: A, B
Steps	ContextManager.java starts reading the context in LocationServer.java module.
Expected results	The ContextManager.java module receives a list of indoor locations: A, B
Actual results	The ContextManager.java module receives a list of indoor locations: A, B
Test Results	**PASSED**

Table 52: The ContextManager.java module receives empty values from the LocationServer.java module

Title	The ContextManager.java module receives empty values from the LocationServer.java module
Description	Check if the ContextManager.java module receives empty values from the LocationServer.java module upon request.
Preconditions	+ User has logged into the application with their account. + LocationServer.java fails to read values from [configuration-file].txt
Test Data	N/A
Steps	ContextManager.java starts reading the context in LocationServer.java module.
Expected results	The ContextManager.java module does not receive any values
Actual results	The ContextManager.java module does not receive any values
Test Results	**PASSED**

4.2.4. PreferenceRepository.java Module and ContextManager.java Module

Title	The PreferenceRepository.java receives an alert temperature from ContextManager.java module

Description	Check if the PreferenceRepository.java module is able to receive an alert temperature from the ContextManager.java module
Preconditions	+ User has logged into the application with their account. + The Context Manager does not receive any alert notifications from other modules.
Test Data	+ Username: Jack + Weather: 0 + Temperature value: 20 + Temperature Preference 1: when 20 suggest shops + Temperature Preference 2: when 30 suggest pool
Steps	+ Invoke PreferenceRepository.readPreference() to read preferences + Invoke the PreferenceRequest.class + Store username, weather and temperature value as parameters + Get the suggestions
Expected results	shops
Actual results	shops
Test Results	**PASSED**

Title	The PreferenceRepository.java does not receive an alert temperature from ContextManager.java module #1
Description	Check if the PreferenceRepository.java module should not receive an alert temperature from the ContextManager.java module when the temperature threshold hasn’t been reached.
Preconditions	+ User has logged into the application with their account. + The Context Manager receives alert notifications from other modules and send request to the Preference Repository. + User has defined the preferences. + The Preference Repository has successfully retrieved data from the [preference-file].txt
Test Data	+ Username: Jack + Weather: 0 + Temperature value: 15 + Temperature Preference 1: when 20 suggest shops + Temperature Preference 2: when 30 suggest pool
Steps	+ Invoke PreferenceRepository.readPreference() to read preferences + Invoke the PreferenceRequest.class + Store username, weather and temperature value as parameters + Get the suggestions
Expected results	empty

Actual results	N/A
Test results	N/A

Title	The PreferenceRepository.java does not receive an alert temperature from ContextManager.java module #2
Description	Check if the PreferenceRepository.java module should not receive an alert temperature from the ContextManager.java module when the input temperature is a negative number
Preconditions	+ User has logged into the application with their account. + The Context Manager receives alert notifications from other modules and send request to the Preference Repository. + User has defined the preferences. + The Preference Repository has successfully retrieved data from the [preference-file].txt
Test Data	+ Username: Jack + Weather: 0 + Temperature value: -1 + Temperature Preference 1: when 20 suggest shops + Temperature Preference 2: when 30 suggest pool
Steps	+ Invoke PreferenceRepository.readPreference() to read preferences + Invoke the PreferenceRequest.class + Store username, weather and temperature value as parameters + Get the suggestions
Expected results	empty
Actual results	N/A
Test results	N/A

Title	The PreferenceRepository.java receives an alert temperature from ContextManager.java module
Description	Check if the PreferenceRepository.java module is able to receive an alert temperature from the ContextManager.java module
Preconditions	+ User has logged into the application with their account. + The Context Manager does not receive any alert notifications from other modules.
Test Data	+ Username: Jack + Weather: 1 + Temperature value: “ ” + Weather Preference: cinema

Steps	+ Invoke PreferenceRepository.readPreference() to read preferences + Invoke the PreferenceRequest.class + Store username, weather and temperature value as parameters + Get the suggestions
Expected results	cinema
Actual results	cinema
Test Result	**PASSED**

Title	The PreferenceRepository.java does not receive a weather alert from ContextManager.java module #1
Description	Check if the PreferenceRepository.java module should not receive a weather alert from the ContextManager.java module when the weather condition is normal (0)
Preconditions	+ User has logged into the application with their account. + The Context Manager receives alert notifications from other modules and send request to the Preference Repository. + User has defined the preferences. + The Preference Repository has successfully retrieved data from the [preference-file].txt
Test Data	+ Username: Jack + Weather: 0 + Temperature value: “ ” + Weather Preference: cinema
Steps	+ Invoke PreferenceRepository.readPreference() to read preferences + Invoke the PreferenceRequest.class + Store username, weather and temperature value as parameters + Get the suggestions
Expected results	empty
Actual results	cinema
Test Result	**FAILED**

Title	The PreferenceRepository.java does not receive a weather alert from ContextManager.java module #2
Description	Check if the PreferenceRepository.java module should not receive a weather alert from the ContextManager.java module when the weather condition is a negative number (-1)
Preconditions	+ User has logged into the application with their account.

	+ The Context Manager receives alert notifications from other modules and send request to the Preference Repository. + User has defined the preferences. + The Preference Repository has successfully retrieved data from the [preference-file].txt
Test Data	+ Username: Jack + Weather: -1 + Temperature value: " " + Weather Preference: cinema
Steps	+ Invoke PreferenceRepository.readPreference() to read preferences + Invoke the PreferenceRequest.class + Store username, weather and temperature value as parameters + Get the suggestions
Expected results	empty
Actual results	cinema
Test result	**FAILED**

Title	The PreferenceRepository.java receives an APO alert from ContextManager.java module
Description	Check if the PreferenceRepository.java module is able to receive an APO alert from the ContextManager.java module
Preconditions	+ User has logged into the application with their account. + The Context Manager does not receive any alert notifications from other modules.
Test Data	+ Username: Jack + Weather: 0 + Value = "APO" + APO preference: bowling
Steps	+ Invoke PreferenceRepository.readPreference() to read preferences + Invoke the PreferenceRequest.class + Store username, weather and temperature value as parameters + Get the suggestions
Expected results	bowling
Actual results	bowling
Test Results	**PASSED**

4.2.5. ContextManager.java Module and EnviroAppUI.java Module

Table 53: The EnviroAppUI.java module sends a request for information to the ContextManager.java module and there is at least one match found

Title	The EnviroAppUI.java module sends a request for information to the ContextManager.java module and there is at least one match found
Description	Check if the EnviroAppUI.java module could successfully send a request for information (either option 1 or 2 in the menu) to the ContextManager.java module and at least one match is found.
Preconditions	+ User has logged into the application with their account. + The Context Manager has successfully retrieved data from the [city-information-file].txt. + User has chosen an option.
Test Data	+ Option: 1 + Item of interest: Vivo City Shopping Centre
Steps	Invoke Context Manager's worker searchInfo() function
Expected results	The ContextManager.java module returns the following data and display on the UI for the user: Information about Vivo City Shopping Centre: Vivo City Shopping Centre is a major regional shopping centre in the southern suburb of Ho Chi Minh City, Vietnam. It is the second largest shopping centre in the southern suburbs of Ho Chi Minh City, by gross area, and contains the only H&M store in that region.
Actual results	The ContextManager.java module returns the following data and display on the UI for the user: Information about Vivo City Shopping Centre: Vivo City Shopping Centre is a major regional shopping centre in the southern suburb of Ho Chi Minh City, Vietnam. It is the second largest shopping centre in the southern suburbs of Ho Chi Minh City, by gross area, and contains the only H&M store in that region.
Test Results	**PASSED**

Table 54: The EnviroAppUI.java module sends a request for information to the ContextManager.java module but no match is found

Title	The EnviroAppUI.java module sends a request for information to the ContextManager.java module but no match is found
Description	Check if the EnviroAppUI.java module could successfully send a request for information (either option 1 or 2 in the menu) to the ContextManager.java module but no match is found
Preconditions	+ User has logged into the application with their account. + The Context Manager has successfully retrieved data from the [city-information-file].txt. + User has chosen an option.
Test Data	+ Option: 1 + Item of interest: RMIT
Steps	Invoke Context Manager's worker searchInfo() function
Expected results	The ContextManager.java module returns the following data and display on the UI for the user: No match found for item of interest
Actual results	The ContextManager.java module returns the following data and display on the UI for the user: No match found for item of interest
Test Results	**PASSED**

Title	The EnviroAppUI.java module sends a request for searching items of interest in the current location
Description	Check if the ContextManager.java module returns a list of items of interest in user's current location requested by the EnviroAPPUI.java module
Preconditions	+ User has logged into the application with their account. + The Context Manager has successfully retrieved data from the [city-information-file].txt. + User does not choose any options.
Test Data	
Steps	Invoke Context Manager's worker searchItems() function
Expected results	N/A
Actual results	N/A
Test results	N/A

Title	The EnviroAppUI.java module sends a request for searching items of interest in the current location, but no items are found

Description	Check if the ContextManager.java module returns an empty list of interests in user's current location requested by the EnviroAPPUI.java module as there is no items of interest in the current location.
Preconditions	+ User has logged into the application with their account. + The Context Manager has successfully retrieved data from the [city-information-file].txt. + User does not choose any options.
Test Data	
Steps	Invoke Context Manager's worker searchItems() function
Expected results	N/A
Actual results	N/A
Test results	N/A

5. System Test

5.1. Functional Testing

5.1.1. Login

Table 55: Validate login - Username is not registered

Title	Validate login - Username is not registered
Description	Evaluate the behavior of the application when user enters an unregistered username
Preconditions	+ The device is connected to the Internet. + The device's on-screen keyboard is functional. + User has not yet created an account.
Test Data	Username: 'Jack1'
Steps	+ Type 'Jack1' using the on-screen keyboard. + Press "Enter" using the on-screen keyboard.
Expected results	The application returns a FileNotFoundException error.
Actual results	The application returns a FileNotFoundException error.
Test results	**PASSED**

Table 56: Validate login - Username is not entered (blank)

Title	Validate login - Username is not entered (blank)
Description	Evaluate the behavior of the application when the username is leaved blank

Preconditions	+ The device is connected to the Internet. + The device’s on-screen keyboard is functional.
Test Data	N/A
Steps	Press “Enter” using the on-screen keyboard without specifying a username
Expected results	The application returns a FileNotFoundException error.
Actual results	The application returns a FileNotFoundException error.
Test results	**PASSED**

Table 57: Validate login - Username is entered with less than 3 characters

Title	Validate login - Username is entered with less than 3 characters
Description	Evaluate the behavior of the application when the username is entered with less than 3 characters
Preconditions	+ The device is connected to the Internet. + The device’s on-screen keyboard is functional.
Test Data	Username: ‘Al’
Steps	+ Type ‘Al’ using the on-screen keyboard. + Press “Enter” using the on-screen keyboard.
Expected results	The application returns a FileNotFoundException error.
Actual results	The application returns a FileNotFoundException error.
Test results	**PASSED**

Table 58: Validate login - Username is entered with more than 30 characters

Title	Validate login - Username is entered with more than 30 characters
Description	Evaluate the behavior of the application when the username is entered with more than 30 characters
Preconditions	+ The device is connected to the Internet. + The device’s on-screen keyboard is functional.
Test Data	Username: “Jackinthewonderlandwithjohnnyyy”
Steps	+ Type ‘Jackinthewonderlandwithjohnnyyy’ using the on-screen keyboard. + Press “Enter” using the on-screen keyboard.
Expected results	The application returns a FileNotFoundException error.
Actual results	The application returns a FileNotFoundException error.

Test results	**PASSED**

Table 59: Validate login - Username is entered with space(s) in any location

Title	Validate login - Username is entered with space(s) in any location
Description	Evaluate the behavior of the application when the username is entered with space(s) in any location (beginning, middle, end, etc.)
Preconditions	+ The device is connected to the Internet. + The device’s on-screen keyboard is functional.
Test Data	Username: ‘ Jack’
Steps	+ Type ‘ Jack’ using the on-screen keyboard. + Press “Enter” using the on-screen keyboard.
Expected results	The application returns a FileNotFoundException error.
Actual results	The application returns a FileNotFoundException error.
Test results	**PASSED**

Table 60: Validate login - Username is entered with at least one special character

Title	Validate login - Username is entered with at least one special character
Description	Evaluate the behavior of the application when the username is entered with at least one special character
Preconditions	+ The device is connected to the Internet. + The device’s on-screen keyboard is functional.
Test Data	Username: ‘Jack&123**#’
Steps	+ Type ‘Jack&123**#’ using the on-screen keyboard. + Press “Enter” using the on-screen keyboard.
Expected results	The application returns a FileNotFoundException error.
Actual results	The application returns a FileNotFoundException error.
Test results	**PASSED**

Table 61: Validate login - Username is entered with number(s) placed at the beginning

Title	Validate login - Username is entered with number(s) placed at the beginning

Description	Evaluate the behavior of the application when the username is entered with at least one number at the beginning of the field
Preconditions	+ The device is connected to the Internet. + The device's on-screen keyboard is functional.
Test Data	Username: '123Jack'
Steps	+ Type '123Jack' using the on-screen keyboard. + Press "Enter" using the on-screen keyboard.
Expected results	The application returns a FileNotFoundException error.
Actual results	The application returns a FileNotFoundException error.
Test results	**PASSED**

Table 62: Verify user login with no internet connection

Title	Verify user login with no internet connection
Description	Evaluate the behavior of the application when user tries to log in to the application when there is no internet connection.
Preconditions	+ The device's on-screen keyboard is functional. + User has created an account.
Test Data	Username: 'Jack'
Steps	+ Type '123Jack' using the on-screen keyboard. + Press "Enter" using the on-screen keyboard.
Expected results	The application returns an error message: "Canot login due to no internet connectivity. Please try again later"
Actual results	The application returns an error message: "Canot login due to no internet connectivity. Please try again later"
Test results	**PASSED**

Table 63: Verify user login with two devices at the same time

Title	Verify user login with two devices at the same time
Description	Evaluate the behavior of the application when a user tries to log in to the application when his/her account is currently logged in on another device
Preconditions	+ The device is connected to the Internet. + User has created an account. + The device's on-screen keyboard is functional.
Test Data	Username: 'Jack'
Steps	+ Type 'Jack' using the on-screen keyboard. + Press "Enter" using the on-screen keyboard.

Expected results	The application returns an error message: “Your account has already logged in on another device. Please try again”
Actual results	The application returns an error message: “Your account has already logged in on another device. Please try again”
Test results	**PASSED**

5.1.2. Preferences

Table 64: Validate temperature preference – value between 0 and 60

Title	Validate temperature preference – value between 0 and 60
Description	Evaluate the behavior of the application when user defines the temperature threshold between 0 – 60 and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Temperature Threshold: 30 + Temperature Service Type: pool
Steps	+ Open the preference file. + Type “when 30 suggest pool” into one of the three preference slots + Save the preference file.
Expected results	In the preference file, for user Jack, the following preference can be found: pref: when 30 suggest pool
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when 30 suggest pool
Test results	**PASSED**

Table 65: Validate temperature preference – value below 0

Title	Validate user’s input for temperature preferences #2
Description	Evaluate the behavior of the application when user defines the temperature threshold below 0 and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Temperature Threshold: -5 + Temperature Service Type: pool

Steps	+ Open the preference file. + Type “when -5 suggest pool” into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the temperature threshold must be from 0 – 60 and ask user to define again
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when -5 suggest pool (no validations for user’s input of the preferences are implemented at this point)
Test Results	**FAILED**

Table 66: Validate temperature preference – value above 60

Title	Validate user’s input for temperature preferences #3
Description	Evaluate the behavior of the application when user defines the temperature threshold above 60 and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Temperature Threshold: 102 + Temperature Service Type: pool
Steps	+ Open the preference file. + Type “when 102 suggest pool” into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the temperature threshold must be from 0 – 60 and ask user to define again
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when 102 suggest pool (no validations for user’s input of the preferences are implemented at this point)
Test Results	**FAILED**

Table 67: Validate temperature preference – value contains alphabetic character(s)

Title	Validate user’s input for temperature preferences #4
Description	Evaluate the behavior of the application when user defines the temperature threshold contains alphabetic character(s) and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).

Test Data	+ Username: Jack + Temperature Threshold: hello + Temperature Service Type: pool
Steps	+ Open the preference file. + Type “when hello suggest pool” into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the temperature threshold must be an integer number and ask user to define again.
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when hello suggest pool (no validations for user’s input of the preferences are implemented at this point)
Test Results	**FAILED**

Table 68: Validate temperature preference – value contains special character(s)

Title	Validate user’s input for temperature preferences #5
Description	Evaluate the behavior of the application when user defines the temperature threshold contains special character(s) and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Temperature Threshold: 23#$ + Temperature Service Type: pool
Steps	+ Open the preference file. + Type “when 23#$ suggest pool” into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the temperature threshold must be an integer number and ask user to define again
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when 23#$ suggest pool (no validations for user’s input of the preferences are implemented at this point)
Test Results	**FAILED**

Table 69: Validate temperature preference – value contains space(s)

Title	Validate user’s input for temperature preferences #6
Description	Evaluate the behavior of the application when user defines the temperature threshold contains space(s) and valid corresponding service type

Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Temperature Threshold: '23 84' + Temperature Service Type: pool
Steps	+ Open the preference file. + Type "when 23 84 suggest pool" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the temperature threshold must not include a space and ask the user to define again.
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when 23 84 suggest pool (no validations for user's input of the preferences are implemented at this point)
Test Results	**FAILED**

Table 70: Validate temperature preference – value is not entered

Title	Validate user's input for temperature preferences #7
Description	Evaluate the behavior of the application when user leaves the temperature threshold blank and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Temperature Service Type: pool
Steps	+ Open the preference file. + Type "when suggest pool" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the temperature threshold must be defined and ask user to try again.
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when suggest pool (no validations for user's input of the preferences are implemented at this point)
Test Results	**FAILED**

Table 71: Validate temperature preference – value is a non-integer number

Title	Validate user's input for temperature preferences #8

Description	Evaluate the behavior of the application when user defines the temperature threshold a non-integer number and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Temperature Threshold: 34.25 + Temperature Service Type: pool
Steps	+ Open the preference file. + Type “when 34.25 suggest pool” into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the temperature threshold must be defined as an integer value and ask user to try again.
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when 34.25 suggest pool (no validations for user’s input of the preferences are implemented at this point)
Test Results	**FAILED**

Table 72: Validate temperature preference – unknown temperature service type

Title	Validate user’s input for temperature preferences #9
Description	Evaluate the behavior of the application when user defines a valid temperature threshold and an unknown corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Temperature Threshold: 34 + Temperature Service Type: poolemo
Steps	+ Open the preference file. + Type “when 34 suggest poolemo” into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the suggested service type for temperature preference is unknown and display a list of available, pre-defined service types. The user is then asked to try again.
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when 34 suggest poolemo (no validations for user’s input of the preferences are implemented at this point)
Test Results	**FAILED**

Table 73: Validate temperature preference – numeric temperature service type

Title	Validate user's input for temperature preferences #10
Description	Evaluate the behavior of the application when user defines a valid temperature threshold and a numeric value for the corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Temperature Threshold: 34 + Temperature Service Type: 23
Steps	+ Open the preference file. + Type "when 34 suggest 23" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that user cannot define the temperature service type as a numeric value. The user is then asked to try again.
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when 34 suggest 23 (no validations for user's input of the preferences are implemented at this point)
Test Results	**FAILED**

Table 74: Validate temperature preference – temperature service type is not entered

Title	Validate user's input for temperature preferences #11
Description	Evaluate the behavior of the application when user defines a valid temperature threshold and leaves the corresponding service type blank
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Temperature Threshold: 34
Steps	+ Open the preference file. + Type "when 34 suggest " into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that user cannot leave the service type blank and display a list of available, pre-defined service types. The user is then asked to try again.
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when 34 suggest

	(no validations for user's input of the preferences are implemented at this point)
Test Results	**FAILED**

Table 75: Validate temperature preference – temperature service type contains space(s)

Title	Validate user's input for temperature preferences #12
Description	Evaluate the behavior of the application when user defines a valid temperature threshold and contain space(s) in the corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Temperature Threshold: '34' + Temperature Service Type: pool and playground
Steps	+ Open the preference file. + Type "when 34 suggest pool and playground" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the service type must not include a space and ask the user to define again
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when 34 suggest pool and playground (no validations for user's input of the preferences are implemented at this point)
Test Results	**FAILED**

Table 76: Validate weather preference – valid weather service type

Title	Validate weather preference – valid weather service type
Description	Evaluate the behavior of the application when user defines the valid corresponding service type for weather alarm
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Weather Service Type: mall
Steps	+ Open the preference file. + Type "when weather suggest mall" into one of the three preference slots. + Save the preference file.

Expected results	In the preference file, for user Jack, the following preference can be found: pref: when weather suggest mall
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when weather suggest mall
Test results	**PASSED**

Table 77: Validate weather preference – unknown weather service type

Title	Validate user's input for weather preferences #2
Description	Evaluate the behavior of the application when user defines a valid weather value and an unknown corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Weather Service Type: school
Steps	+ Open the preference file. + Type "when weather suggest school" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the suggested service type for weather preference is unknown and display a list of available, pre-defined service types. The user is then asked to try again.
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when weather suggest school (no validations for user's input of the preferences are implemented at this point)
Test Results	**FAILED**

Table 78: Validate weather preference – numeric weather service type

Title	Validate user's input for weather preferences #3
Description	Evaluate the behavior of the application when user defines a numeric value for the corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Weather Service Type: 23
Steps	+ Open the preference file. + Type "when weather suggest 23 " into one of the three preference slots.

	+ Save the preference file.
Expected results	The application displays an error message, indicating that user cannot define the temperature service type as a numeric value. The user is then asked to try again.
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when weather suggest 23 (no validations for user's input of the preferences are implemented at this point)
Test Results	**FAILED**

Table 79: Validate weather preference – weather service type is not entered

Title	Validate user's input for weather preferences #4
Description	Evaluate the behavior of the application when user leaves the corresponding service type blank
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	Username: Jack
Steps	+ Open the preference file. + Type "when weather suggest" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that user cannot leave the service type blank and display a list of available, pre-defined service types. The user is then asked to try again.
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when weather suggest (no validations for user's input of the preferences are implemented at this point)
Test Results	**FAILED**

Table 80: Validate weather preference – outdoor weather service type

Title	Validate user's input for weather preferences #5
Description	Evaluate the behavior of the application when user defines a valid but outdoor service type for weather alarm
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Weather Service Type: pool

Steps	+ Open the preference file. + Type "when weather suggest pool " into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that user can only define indoor service types and display a list of available, pre-defined indoor service types. The user is then asked to try again.
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when weather suggest pool (no validations for user's input of the preferences are implemented at this point)
Test Results	**FAILED**

Table 81: Validate weather preference – weather service type contains space(s)

Title	Validate user's input for weather preferences #6
Description	Evaluate the behavior of the application when user defines a corresponding service type that contain space(s)
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Weather Service Type: restaurant mall
Steps	+ Open the preference file. + Type "when weather suggest restaurant mall" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the service type must not include a space and ask the user to define again
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when weather suggest restaurant mall (no validations for user's input of the preferences are implemented at this point)
Test Results	**FAILED**

Table 82: Validate APO preference – medical condition type is between 1 and 3

Title	Validate APO preference – medical condition type is between 1 and 3
Description	Evaluate the behavior of the application when user defines the medical condition type between 1 – 3 and valid corresponding service type
Preconditions	+ User has already created an account.

	+ The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Medical Condition Type: 2 + APO Service Type: restaurant
Steps	+ Open the preference file. + Type "2" into the medical condition type line. + Type "when APO suggest restaurant" into one of the three preference slots + Save the preference file.
Expected results	In the preference file, for user Jack, the following content can be found: name: Jack Medical Condition Type: 2 pref: when APO suggest restaurant
Actual results	In the preference file, for user Jack, the following content can be found: name: Jack Medical Condition Type: 2 pref: when APO suggest restaurant
Test results	**PASSED**

Table 83: Validate APO preference – medical condition type is below 1

Title	Validate APO preference – medical condition type is below 1
Description	Evaluate the behavior of the application when user defines the medical condition type below 1 and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Medical Condition Type: -1 + APO Service Type: restaurant
Steps	+ Open the preference file. + Type "-1" into the medical condition type line. + Type "when APO suggest restaurant" into one of the three preference slots. + Save the preference file.

Expected results	The application displays an error message, indicating that the medical condition type must be positive integer value and ask the user to define again
Actual results	In the preference file, for user Jack, the following content can be found: name: Jack Medical Condition Type: -1 pref: when APO suggest restaurant (no validations for user's input of the preferences are implemented at this point)
Test results	**FAILED**

Table 84: Validate APO preference – medical condition type is above 3

Title	Validate APO preference – medical condition type is above 3
Description	Evaluate the behavior of the application when user defines the medical condition type above 3 and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Medical Condition Type: 6 + APO Service Type: restaurant
Steps	+ Open the preference file. + Type "when APO suggest pool" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the medical condition type must be in range [1, 3] value and ask the user to define again
Actual results	In the preference file, for user Jack, the following content can be found: name: Jack Medical Condition Type: 6 pref: when APO suggest restaurant (no validations for user's input of the preferences are implemented at this point)
Test results	**FAILED**

Table 85: Validate APO preference – medical condition type contains alphabetic character(s)

Title	Validate APO preference – medical condition type contains alphabetic character(s)

Description	Evaluate the behavior of the application when user defines the medical condition type contains alphabetic character(s) and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Medical Condition Type: hello + APO Service Type: restaurant
Steps	+ Open the preference file. + Type "hello" into the medical condition type line. + Type "when APO suggest restaurant" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the medical condition type must be positive integer value and ask the user to define again
Actual results	In the preference file, for user Jack, the following content can be found: name: Jack Medical Condition Type: hello pref: when APO suggest restaurant (no validations for user's input of the preferences are implemented at this point)
Test results	**FAILED**

Table 86: Validate APO preference – medical condition type contains special character(s)

Title	Validate APO preference – medical condition type contains special character(s)
Description	Evaluate the behavior of the application when user defines the medical condition type contains special character(s) and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Medical Condition Type: 2# + APO Service Type: restaurant
Steps	+ Open the preference file. + Type "2#" into the medical condition type line.

	+ Type "when APO suggest restaurant" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the medical condition type must be positive integer value and ask the user to define again
Actual results	In the preference file, for user Jack, the following content can be found: name: Jack Medical Condition Type: 2# pref: when APO suggest restaurant (no validations for user's input of the preferences are implemented at this point)
Test results	**FAILED**

Table 87: Validate APO preference – medical condition type contains space(s)

Title	Validate APO preference – medical condition type contains space(s)
Description	Evaluate the behavior of the application when user defines the medical condition type contains space(s) and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Medical Condition Type: '3 2' + APO Service Type: restaurant
Steps	+ Open the preference file. + Type "3 2" into the medical condition type line. + Type "when APO suggest restaurant" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the medical condition type only contain 1 value and ask the user to define again
Actual results	In the preference file, for user Jack, the following content can be found: name: Jack Medical Condition Type: 3 2 pref: when APO suggest restaurant (no validations for user's input of the preferences are implemented at this point)
Test results	**FAILED**

Table 88: Validate APO preference – medical condition type is not entered

Title	Validate APO preference – medical condition type is not entered
Description	Evaluate the behavior of the application when user leaves the medical condition type blank and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + APO Service Type: restaurant
Steps	+ Open the preference file. + Type "when APO suggest restaurant" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the medical condition type must not empty and ask the user to define again
Actual results	In the preference file, for user Jack, the following content can be found: name: Jack Medical Condition Type: pref: when APO suggest restaurant (no validations for user's input of the preferences are implemented at this point)
Test results	**FAILED**

Table 89: Validate APO preference – medical condition type contains a numeric but non-integer number

Title	Validate APO preference – medical condition type contains a numeric but non-integer number
Description	Evaluate the behavior of the application when user defines the medical condition type a numeric but non-integer number and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Medical Condition Type: 2.2 + APO Service Type: restaurant
Steps	+ Open the preference file. + Type "2.2" into the medical condition type line.

	+ Type "when APO suggest restaurant" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the medical condition type must be positive integer value and ask the user to define again
Actual results	In the preference file, for user Jack, the following content can be found: name: Jack Medical Condition Type: 2.2 pref: when APO suggest restaurant (no validations for user's input of the preferences are implemented at this point)
Test results	**FAILED**

Table 90: Validate APO preference – unknown APO service type

Title	Validate APO preference – unknown APO service type
Description	Evaluate the behavior of the application when user defines a valid medical condition type and an unknown corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Medical Condition Type: 2 + APO Service Type: restaurantoooo
Steps	+ Open the preference file. + Type "2" into the medical condition type line. + Type "when APO suggest restaurantoooo" into one of the three preference slots. + Save the preference file.
Expected results	
Actual results	In the preference file, for user Jack, the following content can be found: name: Jack Medical Condition Type: 2 pref: when APO suggest restaurantoooo (no validations for user's input of the preferences are implemented at this point)
Test results	**FAILED**

Table 91: Validate APO preference – numeric APO service type

Title	Validate APO preference – numeric APO service type
Description	Evaluate the behavior of the application when user defines a valid medical condition type and a numeric value for the corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Medical Condition Type: 2 + APO Service Type: 23
Steps	+ Open the preference file. + Type "2" into the medical condition type line. + Type "when APO suggest 23 " into one of the three preference slots. + Save the preference file.
Expected results	In the preference file, for user Jack, the following content can be found: name: Jack Medical Condition Type: 2 pref: when APO suggest 23 (no validations for user's input of the preferences are implemented at this point)
Actual results	In the preference file, for user Jack, the following content can be found: name: Jack Medical Condition Type: 2 pref: when APO suggest 23 (no validations for user's input of the preferences are implemented at this point)
Test results	**FAILED**

Table 92: Validate APO preference – APO service type is not entered

Title	Validate user's input for APO preferences #11
Description	Evaluate the behavior of the application when user defines a valid medical condition type and leaves the corresponding service type blank
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).

Test Data	+ Username: Jack + Medical Condition Type: 2
Steps	+ Open the preference file. + Type "2" into the medical condition type line. + Type "when APO suggest <blank>" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that user cannot leave the service type blank and display a list of available, pre-defined service types. The user is then asked to try again.
Actual results	In the preference file, for user Jack, the following content can be found: name: Jack Medical Condition Type: 2 pref: when APO suggest (no validations for user's input of the preferences are implemented at this point)
Test Result	**FAILED**

Table 93: Validate APO preference – APO service type contains space(s)

Title	Validate user's input for APO preferences #12
Description	Evaluate the behavior of the application when user defines a valid medical condition type and a corresponding service type that contain space(s)
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Medical Condition Type: 2 + APO Service Type: restaurant mall
Steps	+ Open the preference file. + Type "2" into the medical condition type line. + Type "when APO suggest restaurant mall " into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the service type must not include a space and ask the user to define again.
Actual results	In the preference file, for user Jack, the following content can be found: name: Jack Medical Condition Type: 2 pref: when APO suggest restaurant mall (no validations for user's input of the preferences are implemented at this point)

Test Result	**FAILED**

Table 94: Validate APO preference – outdoor APO service type

Title	Validate user's input for APO preferences #13
Description	Evaluate the behavior of the application when user defines a valid medical condition type and a valid but outdoor service type for weather alarm
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Medical Condition Type: 2 + APO Service Type: pool
Steps	+ Open the preference file. + Type "2" into the medical condition type line. + Type "when APO suggest pool " into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that user can only define indoor service types and display a list of available, pre-defined indoor service types. The user is then asked to try again.
Actual results	In the preference file, for user Jack, the following content can be found: name: Jack Medical Condition Type: 2 pref: when APO suggest pool (no validations for user's input of the preferences are implemented at this point)
Test Result	**FAILED**

Table 95: Verify the valid number of different preferences in the preference file

Title	Verify the valid number of different preferences in the preference file
Description	Evaluate the behavior of the application when the number of different preferences from users are in the valid range (1 – 3) in the preference file.
Preconditions	+ User has already created an account. + The preference text file is created.
Test Data	+ Preference 1: Name: Jack Medical Condition Type: 1 pref-1: when 30 suggest pool pref-2: when APO suggest cinema pref-3: when weather suggest cinema

	+ Preference 2: Name: David Medical Condition Type: 2 pref-1: when 30 suggest pool pref-2: when APO suggest restaurant pref-3: when weather suggest mall
Steps	+ Open the preference file. + Type 2 preferences into the preference file. + Save the preference file.
Expected results	The application acknowledges the preferences from 2 different users and send warnings accordingly if occurs
Actual results	The application acknowledges the preferences from 2 different users and send warnings accordingly if occurs
Test results	PASS

Table 96: Verify the invalid number of different preferences in the preference file

Title	Verify the invalid number of different preferences in the preference file
Description	Evaluate the behavior of the application when the number of different preferences from users are outside the valid range (more than 3) in the preference file
Preconditions	+ User has already created an account. + The preference text file is created.
Test Data	+ Preference 1: Name: Jack Medical Condition Type: 1 pref-1: when 30 suggest pool pref-2: when APO suggest cinema pref-3: when weather suggest cinema + Preference 2: Name: David Medical Condition Type: 2 pref-1: when 30 suggest pool pref-2: when APO suggest restaurant pref-3: when weather suggest mall + Preference 3: Name: Alice Medical Condition Type: 2 pref-1: when 30 suggest pool

	pref-2: when APO suggest restaurant pref-3: when weather suggest mall + Preference 4: Name: Minh Dinh Medical Condition Type: 3 pref-1: when 30 suggest pool pref-2: when APO suggest restaurant pref-3: when weather suggest mall
Steps	+ Open the preference file. + Type 4 preferences into the file. + Save the preference file.
Expected results	The application returns an error indicating that there must not be more than 3 user preferences and only acknowledge the first three preferences.
Actual results	The application returns an error indicating that there must not be more than 3 user preferences and only acknowledge the first three preferences.
Test results	PASS

Table 97: Verify the empty preference file

Title	Verify the empty preference file
Description	Evaluate the behavior of the application when there are no preferences in the preference file
Preconditions	+ User has already created an account. + The preference text file is created.
Test Data	N/A
Steps	+ Open the preference file. + Delete any preferences (if any). + Save the preference file.
Expected results	The application acknowledges the empty preference, however, will not send any warnings to users in the future.
Actual results	The application acknowledges the empty preference, however, will not send any warnings to users in the future.
Test results	PASS

Table 98: Verify the valid format of the preference file and its content

Title	Verify the valid format of the preference file and its content
Description	Evaluate the behavior of the application when the input taken from users follows the correct pre-defined format of the preference file as follows: . Each entry has the name of the preference owner . Each field in the entry is terminated with a new line character

	. Each entry is separated from other entries by an empty line
Preconditions	+ User has already created an account. + The preference text file is created.
Test Data	Name: Jack Medical Condition Type: 1 pref-1: when 30 suggest pool pref-2: when APO suggest cinema pref-3: when weather suggest cinema Name: David Medical Condition Type: 2 pref-1: when 30 suggest pool pref-2: when APO suggest restaurant pref-3: when weather suggest mall
Steps	+ Open the preference file. + Type 2 preferences into the file. + Save the preference file.
Expected results	The application acknowledges the two preferences and send warnings in the future accordingly
Actual results	The application acknowledges the two preferences and send warnings in the future accordingly
Test results	PASS

Table 99: Verify the invalid format of the preference file and its content

Title	Verify the invalid format of the preference file and its content
Description	Evaluate the behavior of the application when the input taken from users does not follow the correct pre-defined format of the preference file as follows: . Each entry has the name of the preference owner . Each field in the entry is terminated with a new line character . Each entry is separated from other entries by an empty line
Preconditions	+ User has already created an account. + The preference text file is created.
Test Data	Name: Medical Condition Type: 1 pref-1: when 30 suggest pool pref-2: when APO suggest cinema pref-3: when weather suggest cinema Name: David Medical Condition Type: 2 pref-1: when 30 suggest pool pref-2: when APO suggest restaurant

	pref-3: when weather suggest mall
Steps	+ Open the preference file. + Type 2 preferences into the file. + Save the preference file.
Expected results	The application returns an error message, indicating that the format in the preference file is invalid, hence not being able to send warnings to user in the future
Actual results	The application returns an error message, indicating that the format in the preference file is invalid, hence not being able to send warnings to user in the future
Test results	PASS

5.1.3. Warning and suggestions

Table 100: Verify APO Overexposure warning – Air Quality: good, Medical Condition Type: 1

Title	Verify APO Overexposure warning – Air Quality: good, Medical Condition Type: 1
Description	Check if the application does not send an APO Overexposure warning along with appropriate suggestions of indoor locations as the threshold has not yet been breached yet regardless of user's location
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Medical Condition Type: 1 + AQI: 10 + APO Service Type: cinema
Steps	+ Place the device outdoor + Set received AQI to 10
Expected results	The application does not send an APO Overexposure warning and behaves as usual
Actual results	The application does not send an APO Overexposure warning and behaves as usual
Test Result	**PASSED**

Table 101: Verify APO Overexposure warning – Air Quality: good, Medical Condition Type: 2

Title	Verify APO Overexposure warning – Air Quality: good, Medical Condition Type: 2

Description	Check if the application does not send an APO Overexposure warning along with appropriate suggestions of indoor locations as the threshold has not yet been breached yet regardless of user's location
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Medical Condition Type: 2 + AQI: 10 + APO Service Type: cinema
Steps	+ Place the device outdoor + Set received AQI to 10
Expected results	The application does not send an APO Overexposure warning and behaves as usual
Actual results	The application does not send an APO Overexposure warning and behaves as usual
Test Result	**PASSED**

Table 102: Verify APO Overexposure warning – Air Quality: good, Medical Condition Type: 3

Title	Verify APO Overexposure warning – Air Quality: good, Medical Condition Type: 3
Description	Check if the application does not send an APO Overexposure warning along with appropriate suggestions of indoor locations as the threshold has not yet been breached yet regardless of user's location
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Medical Condition Type: 3 + AQI: 10 + APO Service Type: cinema
Steps	+ Place the device outdoor + Set received AQI to 10
Expected results	The application does not send an APO Overexposure warning and behaves as usual
Actual results	The application does not send an APO Overexposure warning and behaves as usual
Test Result	**PASSED**

Table 103: Verify APO Overexposure warning – Air Quality: moderate, Medical Condition Type: 1.

Title	Verify APO Overexposure warning – Air Quality: moderate, Medical Condition Type: 1.
Description	Check if the application sends an APO Overexposure warning along with appropriate suggestions of indoor locations when the air quality is moderate. The user is also located outdoor at the moment the threshold is breached and has a medical condition type 1 in the preferences.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Medical Condition Type: 1 + AQI: 100 => Base Time: 15 seconds. + APO Service Type: cinema
Steps	+ Place the device outdoor + Set received AQI to 100 + Wait for 15 seconds
Expected results	The application displays an APO Overexposure warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 100 Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Actual results	The application displays an APO Overexposure warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 100 Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 3. Search for information on a specific item of interest 4. Search for items of interest in current location E. Exit
Test Result	**PASSED**

Table 104: Verify APO Overexposure warning – Air Quality: unhealthy for sensitive groups, Medical Condition Type: 1

Title	Verify APO Overexposure warning – Air Quality: unhealthy for sensitive groups, Medical Condition Type: 1
Description	Check if the application sends an APO Overexposure warning along with appropriate suggestions of indoor locations when the air quality is unhealthy for sensitive groups. The user is also located outdoor at the moment the threshold is breached and has a medical condition type 1 in the preferences.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Medical Condition Type: 1 + AQI: 150 => Base Time: 10 seconds. + APO Service Type: cinema
Steps	+ Place the device outdoor + Set received AQI to 150 + Wait for 10 seconds
Expected results	The application displays an APO Overexposure warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 150 Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Actual results	The application displays an APO Overexposure warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 150 Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 3. Search for information on a specific item of interest 4. Search for items of interest in current location E. Exit
Test Result	**PASSED**

Table 105: Verify APO Overexposure warning – Air Quality: unhealthy for everyone, Medical Condition Type: 1

Title	Verify APO Overexposure warning – Air Quality: unhealthy for everyone, Medical Condition Type: 1
Description	Check if the application sends an APO Overexposure warning along with appropriate suggestions of indoor locations when the air quality is unhealthy for everyone. The user is also located outdoor at the moment the threshold is breached and has a medical condition type 1 in the preferences.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Medical Condition Type: 1 + AQI: 200 => Base Time: 5 seconds. + APO Service Type: cinema
Steps	+ Place the device outdoor + Set received AQI to 200 + Wait for 5 seconds
Expected results	The application displays an APO Overexposure warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 200 Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Actual results	The application displays an APO Overexposure warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 200 Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 3. Search for information on a specific item of interest 4. Search for items of interest in current location E. Exit
Test Result	**PASSED**

Table 106: Verify APO Overexposure warning – Air Quality: moderate, Medical Condition Type: 2

Title	Verify APO Overexposure warning – Air Quality: moderate, Medical Condition Type: 2
Description	Check if the application sends an APO Overexposure warning along with appropriate suggestions of indoor locations when the air quality is moderate. The user is also located outdoor at the moment the threshold is breached and has a medical condition type 2 in the preferences.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Medical Condition Type: 2 + AQI: 100 => Base Time: 30 seconds. + APO Service Type: cinema
Steps	+ Place the device outdoor + Set received AQI to 100 + Wait for 30 seconds
Expected results	The application displays an APO Overexposure warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 100 Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Actual results	The application displays an APO Overexposure warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 100 Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 3. Search for information on a specific item of interest 4. Search for items of interest in current location E. Exit
Test Result	**PASSED**

Table 107: Verify APO Overexposure warning – Air Quality: unhealthy for sensitive groups, Medical Condition Type: 2

Title	Verify APO Overexposure warning – Air Quality: unhealthy for sensitive groups, Medical Condition Type: 2
Description	Check if the application sends an APO Overexposure warning along with appropriate suggestions of indoor locations when the air quality is unhealthy for sensitive groups. The user is also located outdoor at the moment the threshold is breached and has a medical condition type 2 in the preferences.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Medical Condition Type: 2 + AQI: 150 => Base Time: 20 seconds. + APO Service Type: cinema
Steps	+ Place the device outdoor + Set received AQI to 150 + Wait for 20 seconds
Expected results	The application displays an APO Overexposure warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 150 Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Actual results	The application displays an APO Overexposure warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 150 Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 3. Search for information on a specific item of interest 4. Search for items of interest in current location E. Exit
Test Result	**PASSED**

Table 108: Verify APO Overexposure warning – Air Quality: unhealthy for everyone, Medical Condition Type: 2

Title	Verify APO Overexposure warning – Air Quality: unhealthy for everyone, Medical Condition Type: 2
Description	Check if the application sends an APO Overexposure warning along with appropriate suggestions of indoor locations when the air quality is unhealthy for everyone. The user is also located outdoor at the moment the threshold is breached and has a medical condition type 2 in the preferences.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Medical Condition Type: 2 + AQI: 200 => Base Time: 10 seconds. + APO Service Type: cinema
Steps	+ Place the device outdoor + Set received AQI to 200 + Wait for 5 seconds
Expected results	The application displays an APO Overexposure warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 200 Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Actual results	The application displays an APO Overexposure warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 200 Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 3. Search for information on a specific item of interest 4. Search for items of interest in current location E. Exit
Test Result	**PASSED**

Table 109: Verify APO Overexposure warning – Air Quality: moderate, Medical Condition Type: 3

Title	Verify APO Overexposure warning – Air Quality: moderate, Medical Condition Type: 3
Description	Check if the application sends an APO Overexposure warning along with appropriate suggestions of indoor locations when the air quality is moderate. The user is also located outdoor at the moment the threshold is breached and has a medical condition type 3 in the preferences.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Medical Condition Type: 3 + AQI: 100 => Base Time: 45 seconds. + APO Service Type: cinema
Steps	+ Place the device outdoor + Set received AQI to 100 + Wait for 45 seconds
Expected results	The application displays an APO Overexposure warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 200 Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Actual results	The application displays an APO Overexposure warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 200 Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 3. Search for information on a specific item of interest 4. Search for items of interest in current location E. Exit
Test Result	**PASSED**

Table 110: Verify APO Overexposure warning – Air Quality: unhealthy for sensitive groups, Medical Condition Type: 3

Title	Verify APO Overexposure warning – Air Quality: unhealthy for sensitive groups, Medical Condition Type: 3
Description	Check if the application sends an APO Overexposure warning along with appropriate suggestions of indoor locations when the air quality is unhealthy for sensitive groups. The user is also located outdoor at the moment the threshold is breached and has a medical condition type 3 in the preferences.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Medical Condition Type: 3 + AQI: 150 => Base Time: 30 seconds. + APO Service Type: cinema
Steps	+ Place the device outdoor + Set received AQI to 150 + Wait for 30 seconds
Expected results	The application displays an APO Overexposure warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 150 Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Actual results	The application displays an APO Overexposure warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 150 Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 3. Search for information on a specific item of interest 4. Search for items of interest in current location E. Exit
Test Result	**PASSED**

Table 111: Verify APO Overexposure warning – Air Quality: unhealthy for everyone, Medical Condition Type: 3

Title	Verify APO Overexposure warning – Air Quality: unhealthy for everyone, Medical Condition Type: 3
Description	Check if the application sends an APO Overexposure warning along with appropriate suggestions of indoor locations when the air quality is unhealthy for everyone. The user is also located outdoor at the moment the threshold is breached and has a medical condition type 3 in the preferences.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Medical Condition Type: 3 + AQI: 200 => Base Time: 15 seconds. + APO Service Type: cinema
Steps	+ Place the device outdoor + Set received AQI to 200 + Wait for 15 seconds
Expected results	The application displays an APO Overexposure warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 200 Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Actual results	The application displays an APO Overexposure warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 200 Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 3. Search for information on a specific item of interest 4. Search for items of interest in current location E. Exit
Test Result	**PASSED**

Table 112: Verify the timer count functionality for APO Overexposure warning #1

Title	Verify the timer count functionality for APO Overexposure warning #1
Description	Check if the timer starts to count when user moves from an indoor location to an outdoor location.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally.
Test Data	AQI: 100
Steps	+ Place the device indoor + Set the received AQI to 100 + Move the device to an outdoor location
Expected results	The time starts counting from 0 upon the application detects that the current location is an outdoor location and the AQI is not normal
Actual results	The time starts counting from 0 upon the application detects that the current location is an outdoor location and the AQI is not normal
Test Result	**PASSED**

Table 113: Verify the timer count functionality for APO Overexposure warning #2

Title	Verify the timer count functionality for APO Overexposure warning #2
Description	Check if the timer starts to count when the current AQI and the APO threshold is re-evaluated again
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally.
Test Data	+ Initial AQI: 30 + Final AQI: 60
Steps	+ Set the received AQI to 30 + Place the device outdoor + Set the received AQI to 60
Expected results	The timer counter is not increased and remain the same until it receives an AQI of 60 afterwards
Actual results	The timer counter is not increased and remain the same until it receives an AQI of 60 afterwards
Test Result	**PASSED**

Table 114: Verify temperature warning – warning is sent

Title	Verify temperature warning – warning is sent

Description	Check if the application sends a temperature warning along with appropriate suggestions of indoor/outdoor locations when the temperature threshold is breached.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Outside Temperature: 35 + Preference Temperature: 30 + Temperature Service Type: pool
Steps	+ Place the device outdoor + Set the received temperature value to 35
Expected results	The application displays a temperature above the main menu as well as suggestions for pool places as follows: Context-aware EnviroSmart Application Main Menu Warning, temperature is now 35 Suggestion – please go to Vivo City Shopping Centre Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Actual results	The application displays a temperature above the main menu as well as suggestions for pool places as follows: Context-aware EnviroSmart Application Main Menu Warning, temperature is now 35 Suggestion – please go to Vivo City Shopping Centre Please selection an option: 3. Search for information on a specific item of interest 4. Search for items of interest in current location E. Exit
Test Result	**PASSED**

Table 115: Verify temperature warning - warning is sent, but stopped afterwards

Title	Verify temperature warning - warning is sent, but stopped afterwards
Description	Check if the application stops sending temperature warning along with appropriate suggestions of indoor/outdoor locations when there is a change in temperature after user got the temperature warning.
Preconditions	+ The device is connected to the Internet.

	+ User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Initial Outside Temperature: 35 + Final Outside Temperature: 28 + Preference Temperature: 30 + Temperature Service Type: pool
Steps	+ Place the device outdoor + Set the received temperature value to 35 + Wait until the temperature warning is sent + Set the received temperature value to 28
Expected results	After the temperature warning is displayed on the main menu, the application notices the changes in temperature and remove the warning accordingly.
Actual results	After the temperature warning is displayed on the main menu, the application notices the changes in temperature and remove the warning accordingly.
Test Result	**PASSED**

Table 116: Verify temperature warning – warning is not sent

Title	Verify temperature warning – warning is not sent
Description	Check if the application does not send a temperature warning along with appropriate suggestions of indoor/outdoor locations as the threshold has not yet been breached yet regardless of user's location
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Outside Temperature: 20 + Preference Temperature: 30 + Temperature Service Type: pool
Steps	+ Place the device outdoor + Set the recorded temperature value to 20
Expected results	The application does not send a temperature warning and behaves as usual
Actual results	The application does not send a temperature warning and behaves as usual
Test Result	**PASSED**

Table 117: Verify extreme weather warning – heavy rain

Title	Verify extreme weather warning – heavy rain
Description	Check if the application sends an extreme weather warning along with appropriate suggestions of indoor locations when the weather is considered heavy rain.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Weather Condition Type: 1 (heavy rain) + Weather Service Type: cinema
Steps	Set the weather condition type to 1
Expected results	The application displays an extreme weather warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, extreme weather is detected, the current weather event is heavy rain Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Actual results	The application displays an extreme weather warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, extreme weather is detected, the current weather event is heavy rain Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 3. Search for information on a specific item of interest 4. Search for items of interest in current location E. Exit
Test Result	**PASSED**

Table 118: Verify normal weather

Title	Verify normal weather

Description	Check if the application does not send an extreme weather warning along with appropriate suggestions of indoor locations as the weather condition is normal
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Weather Condition Type: 0 (normal) + Weather Service Type: cinema
Steps	+ Place the device outdoor + Set the weather condition type to 0
Expected results	The application does not send an extreme weather warning and behaves as usual
Actual results	The application does not send an extreme weather warning and behaves as usual
Test Result	**PASSED**

Table 119: Verify extreme weather warning – hail storm

Title	Verify extreme weather warning – hail storm
Description	Check if the application sends an extreme weather warning along with appropriate suggestions of indoor locations when the weather is considered hail storm.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Weather Condition Type: 2 (hail storm) + Weather Service Type: cinema
Steps	Set the weather condition type to 2
Expected results	The application displays an extreme weather warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, extreme weather is detected, the current weather event is hail storm Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location

	E. Exit
Actual results	The application displays an extreme weather warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, extreme weather is detected, the current weather event is hail storm Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 3. Search for information on a specific item of interest 4. Search for items of interest in current location E. Exit
Test Result	**PASSED**

Table 120: Verify extreme weather warning – strong wind

Title	Verify extreme weather warning – strong wind
Description	Check if the application sends an extreme weather warning along with appropriate suggestions of indoor locations when the weather is considered strong wind.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Weather Condition Type: 3 (strong wind) + Weather Service Type: cinema
Steps	Set the weather condition type to 3
Expected results	The application displays an extreme weather warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, extreme weather is detected, the current weather event is strong wind Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Actual results	The application displays an extreme weather warning above the main menu as well as suggestions for cinema places as follows:

	Context-aware EnviroSmart Application Main Menu Warning, extreme weather is detected, the current weather event is strong wind Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 3. Search for information on a specific item of interest 4. Search for items of interest in current location E. Exit
Test Result	**PASSED**

Table 121: Verify extreme weather warning and APO Overexposure warning occur simultaneously

Title	Verify extreme weather warning and APO Overexposure warning occur simultaneously
Description	Check if the application sends an extreme weather warning along with appropriate suggestions of indoor locations when both APO threshold and weather alarm are reached/triggered. The user is located outdoor at the moment the situation occurs
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Weather Condition Type: 1 (heavy rain) + Weather Service Type: cinema + Medical Condition Type: 1 + AQI Index: 100 => Base Time: 15 seconds + Medical Service Type: restaurant
Steps	+ Place the device outdoor + Set the weather value to 1 + Set received AQI to 100 + Wait for 15 seconds
Expected results	The application displays an extreme weather warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, extreme weather is detected, the current weather event is heavy rain Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 1. Search for information on a specific item of interest

	2. Search for items of interest in current location E. Exit
Actual results	The application displays an extreme weather warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, extreme weather is detected, the current weather event is heavy rain Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 3. Search for information on a specific item of interest 4. Search for items of interest in current location E. Exit
Test Result	**PASSED**

Table 122: Verify extreme weather warning and temperature warning occur simultaneously

Title	Verify extreme weather warning and temperature warning occur simultaneously
Description	Check if the application sends an extreme weather warning along with appropriate suggestions of indoor locations when both temperature threshold and weather alarm are reached/triggered. The user is located outdoor at the moment the situation occurs
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Weather Condition Type: 1 (heavy rain) + Weather Service Type: cinema + Temperature Threshold: 30 + Outside Temperature: 35 + Temperature Service Type: pool
Steps	+ Place the device outdoor + Set the weather value to 1 + Set received temperature to 35
Expected results	The application displays an extreme weather warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu

	Warning, extreme weather is detected, the current weather event is heavy rain Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Actual results	The application displays an extreme weather warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, extreme weather is detected, the current weather event is heavy rain Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 3. Search for information on a specific item of interest 4. Search for items of interest in current location E. Exit
Test Result	**PASSED**

Table 123: Verify temperature warning and APO Overexposure warning occur simultaneously

Title	Verify temperature warning and APO Overexposure warning occuring simultaneously
Description	Check if the application sends an APO Overexposure warning along with appropriate suggestions of indoor locations when both thresholds are breached. The user is located outdoor at the moment the situation occurs
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Medical Condition Type: 1 + AQI Index: 100 => Base Time: 15 seconds + Medical Service Type: restaurant + Temperature Threshold: 30 + Outside Temperature: 35 + Temperature Service Type: pool
Steps	+ Place the device outdoor + Set received AQI to 100 + Set received temperature to 35

Expected results	The application displays an APO Overexposure warning above the main menu as well as suggestions for restaurant places as follows: Context-aware EnviroSmart Application Main Menu Warning, extreme weather is detected, the current weather event is heavy rain Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Actual results	The application displays an APO Overexposure warning above the main menu as well as suggestions for restaurant places as follows: Context-aware EnviroSmart Application Main Menu Warning, extreme weather is detected, the current weather event is heavy rain Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 3. Search for information on a specific item of interest 4. Search for items of interest in current location E. Exit
Test Result	**PASSED**

Table 124: Verify extreme weather warning, temperature warning, and APO Overexposure warning occur simultaneously

Title	Verify extreme weather warning, temperature warning, and APO Overexposure warning occuring simultaneously
Description	Check if the application sends an extreme weather warning along with appropriate suggestions of indoor/outdoor locations when all thresholds and alarms are breached/triggered. The user is located outdoor at the moment the situation occurs
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Weather Condition Type: 1 (heavy rain) + Weather Service Type: cinema + Medical Condition Type: 1 + AQI Index: 100 => Base Time: 15 seconds

	+ Medical Service Type: restaurant + Temperature Threshold: 30 + Outside Temperature: 35 + Temperature Service Type: pool
Steps	+ Place the device outdoor + Set weather condition type to 1 + Set received AQI to 100 + Set received temperature to 35
Expected results	The application displays an extreme weather warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, extreme weather is detected, the current weather event is heavy rain Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Actual results	The application displays an extreme weather warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, extreme weather is detected, the current weather event is heavy rain Suggestion – please go to Vivo City Shopping Centre, Crescent Mall Please selection an option: 3. Search for information on a specific item of interest 4. Search for items of interest in current location E. Exit
Test Result	**PASSED**

Table 125: Verify extreme weather warning – warning is sent, but stopped afterwards

Title	Verify extreme weather warning – warning is sent, but stopped afterwards
Description	Check if the application stops sending extreme weather warning along with appropriate suggestions of indoor= locations when there is a change in weather condition
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally.

	+ The application is displaying the main menu.
Test Data	+ Initial Weather Condition Type: 1 (heavy rain) + Final Weather Condition Type: 0 (normal) + Temperature Service Type: cinema
Steps	+ Set the weather condition type to 1 + Wait until the extreme weather warning is sent + Set the weather condition type to 0
Expected results	After the extreme weather warning is displayed on the main menu, the application notices the changes in weather condition and remove the warning accordingly.
Actual results	After the extreme weather warning is displayed on the main menu, the application notices the changes in weather condition and remove the warning accordingly.
Test Result	**PASSED**

5.1.4. Option 1: Search for an item of interest

Table 126: Verify the functionality of searching for an item of interest – Results found

Title	Verify the functionality of searching for an item of interest – Results found
Description	Evaluate the behavior of the application when user searches for a specific item of interest (choosing Option 1) in the main menu, and the application returns the result information to the user
Preconditions	+ The device is connected to the Internet. + The device's on-screen keyboard is functional. + User has logged into their account. + User has chosen option 1 in the main menu and the application displays the following text: Please enter name of item of interest:
Test Data	Search query: 'Vivo City Shopping Centre'
Steps	+ Type 'Vivo City Shopping Centre' using the on-screen keyboard. + Press "Enter" using the on-screen keyboard.
Expected results	The application displays the result on the screen as follows: Information about Vivo City Shopping Centre: Vivo City Shopping Centre is a major regional shopping centre in the southern suburb of Ho Chi Minh City, Vietnam. It is the second largest shopping centre in the southern suburbs of Ho Chi Minh City, by gross area, and contains the only H&M store in that region.
Actual results	The application displays the result on the screen as follows:

	Information about Vivo City Shopping Centre: Vivo City Shopping Centre is a major regional shopping centre in the southern suburb of Ho Chi Minh City, Vietnam. It is the second largest shopping centre in the southern suburbs of Ho Chi Minh City, by gross area, and contains the only H&M store in that region.
Test Result	**PASSED**

Table 127: Verify the functionality of searching for an item of interest – No results found

Title	Verify the functionality of searching for an item of interest – No results found
Description	Evaluate the behavior of the application when user searches for a specific item of interest (choosing Option 1) in the main menu, but there are no results found by the application
Preconditions	+ The device is connected to the Internet. + The device's on-screen keyboard is functional. + User has logged into their account. + User has chosen option 1.
Test Data	Search query: 'Batman Park'
Steps	+ Type 'Batman Park' using the on-screen keyboard. + Press "Enter" using the on-screen keyboard.
Expected results	The application displays the result as follows: No match found for item of interest
Actual results	The application displays the result as follows: No match found for item of interest
Test Result	**PASSED**

Table 128: Verify the functionality of searching for an item of interest – No keyword entered

Title	Verify the functionality of searching for an item of interest – No keyword entered
Description	Evaluate the behavior of the application when user searches for a specific item of interest (choosing Option 1) in the main menu and does not specify the keyword for the search query
Preconditions	+ The device is connected to the Internet. + The device's on-screen keyboard is functional. + User has logged into their account. + User has chosen option 1.
Test Data	N/A

Steps	Press "Enter" using the on-screen keyboard without specifying the keyword
Expected results	The application displays the result as follows: No match found for item of interest
Actual results	The application displays the result as follows: No match found for item of interest
Test Result	**PASSED**

5.1.5. Option 2: Search for list of items of interest in current location

Table 129: Verify the functionality of searching for list of items of interest in current location – Results found

Title	Verify the functionality of searching for list of items of interest in current location – Results found
Description	Evaluate the behavior of the application when user searches for a list of items of interest in current location (choosing Option 2) in the main menu, and the application returns the result information to the user
Preconditions	+ The device is connected to the Internet. + The device's on-screen keyboard is functional. + User has logged into their account. + The application is displaying the menu.
Test Data	User is currently at Location A
Steps	Press option 2 in the main menu: 'Search for a list of items of interest in current location'
Expected results	The application returns a list of items of interest in user's current location (location A) as follows: The following items of interest are in your location: Vivo City Shopping Centre
Actual results	The application returns a list of items of interest in user's current location (location A) as follows: The following items of interest are in your location: Vivo City Shopping Centre
Test Result	**PASSED**

Table 130: Verify the functionality of searching for list of items of interest in current location – Change locations in the middle of the process

Title	Verify the functionality of searching for list of items of interest in current location – Change locations in the middle of the process
Description	Evaluate the behavior of the application when user searches for a list of items of interest in the current location (choose Option 2) in the main menu, and immediately goes to another location in the time being. This can only happen if and only if the user is at the exact crossing line between location A and B.
Preconditions	+ The device is connected to the Internet. + The device's on-screen keyboard is functional. + User has logged into their account. + The application is displaying the menu. + For testing purpose, the coordinate boundaries of both locations are set to minimal.
Test Data	+ Initial Location : A + Final Location : B
Steps	+ Go to the crossing line of 2 locations. + Stay on Location A boundary + Press option 2 in the main menu: 'Search for a list of items of interest in current location' + Immediately changes to Location B
Expected results	The application returns a list of items of interest in user's current location (location A) as follows: The following items of interest are in your location: Crescent Mall
Actual results	The application returns a list of items of interest in user's current location (location A) as follows: The following items of interest are in your location: Crescent Mall
Test Result	**PASSED**

Table 131: Verify the functionality of searching for list of items of interest in current location – No results found

Title	Verify the functionality of searching for list of items of interest in current location – No results found

Description	Evaluate the behavior of the application when user searches for a list of items of interest in current location (choosing Option 2) in the main menu, and the application cannot find available information to the user
Preconditions	+ The device is connected to the Internet. + The device’s on-screen keyboard is functional. + User has logged into their account. + The application is displaying the menu
Test Data	N/A
Steps	Press option 2 in the main menu: ‘Search for a list of items of interest in current location’
Expected results	The application displays ‘There are no items of interest in your current location’ as the result
Actual results	The application displays ‘There are no items of interest in your current location’ as the result
Test Result	**PASSED**

5.1.6. Exit

Table 132: Verify the functionality of logout function

Title	Verify the functionality of logout function
Description	Check if the user is able to logout of the application when choosing the third option in the main menu
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User is currently on the main page. + The application displays the following text: Context-aware EnviroSmart Application Main Menu Please select an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Test Data	N/A
Steps	Press “E. Exit” (Option E) in the main menu
Expected results	The user is logged out and the application will ask for the username as follows: Context-aware EnviroSmart Application Please enter your username:
Actual results	The user is logged out and the application will ask for the username as follows:

	Context-aware EnviroSmart Application Please enter your username:
Test Result	**PASSED**

5.1.7. Start-ups

Table 133: Verify the startup sequence of the modules in the application

Title	Verify the startup sequence of the modules in the application
Description	Check if the application is running the correct sequence of modules at start-up
Preconditions	+ The device is connected to the Internet. + User has not yet opened the application.
Test Data	N/A
Steps	Open the application
Expected results	The application is started using the following sequence: Weather Alarm -> Location Server -> Preference Repository -> Context Manager -> EnviroAPPUI.
Actual results	The application is started using the following sequence: Weather Alarm -> Location Server -> Preference Repository -> Context Manager -> EnviroAPPUI.
Test Result	**PASSED**

Table 134: Verify the startup process of the three sensors

Title	Verify the startup process of the three sensors
Description	Check if the sensors are starting up normally at start up
Preconditions	The device is connected to the Internet.
Test Data	N/A
Steps	Run AllSensors.java module
Expected results	The three sensors – temperature, AP and location sensor starts sending readings to the AllSensors.java module.
Actual results	The three sensors – temperature, AP and location sensor starts sending readings to the AllSensors.java module.
Test Result	**PASSED**

Table 135: Verify the startup process of the WeatherAlarm.java module

Title	Verify the startup process of the WeatherAlarm.java module
Description	Check if the WeatheAlarm.java module is started up properly
Preconditions	The device is connected to the Internet.
Test Data	N/A

Steps	Open the application
Expected results	The WeatherAlarm.java module starts collecting and evaluating the weather condition information
Actual results	The WeatherAlarm.java module starts collecting and evaluating the weather condition information
Test Result	**PASSED**

Table 136: Verify the startup process of the LocationServer.java module

Title	Verify the startup process of the LocationServer.java module
Description	Check if the LocationServer.java module is started up properly
Preconditions	+ The device is connected to the Internet. + Sensors are working normally. + WeatherAlarm.java module is started up properly.
Test Data	N/A
Steps	Open the application
Expected results	The LocationServer.java module starts collecting data from the [configuration-file].txt and location coordinates from the location sensor
Actual results	The LocationServer.java module starts collecting data from the [configuration-file].txt and location coordinates from the location sensor
Test Result	**PASSED**

Table 137: Verify the startup process of the PreferenceRepository.java module

Title	Verify the startup process of the PreferenceRepository.java module
Description	Check if the PreferenceRepository.java module is started up properly
Preconditions	+ The device is connected to the Internet. + Sensors are working normally. + WeatherAlarm.java module and LocationServer.java are started up properly.
Test Data	N/A
Steps	Open the application
Expected results	The PreferenceRepository.java module starts collecting user preferences from the [preference-file].txt
Actual results	
Test Result	**PASSED**

Table 138: Verify the startup process of the ContextManager.java module

Title	Verify the startup process of the ContextManager.java module

Description	Check if the ContextManager.java module is started up properly
Preconditions	+ The device is connected to the Internet. + Sensors are working normally. + The following modules are started up properly in order: . WeatherAlarm.java module . LocationServer.java module . PreferenceRepository.java module
Test Data	N/A
Steps	Open the application
Expected results	The ContextManager.java can start interacting with other modules and text files (receiving data and send requests)
Actual results	The ContextManager.java can start interacting with other modules and text files (receiving data and send requests)
Test Result	**PASSED**

Table 139: Verify the startup process of the EnviroAPPUI.java module

Title	Verify the startup process of the EnviroAPPUI.java module
Description	Check if the EnviroAPPUI.java module is started up properly
Preconditions	+ The device is connected to the Internet. + Sensors are working normally. + The following modules are started up properly in order: . WeatherAlarm.java module . LocationServer.java module . PreferenceRepository.java module . ContextManager.java module
Test Data	N/A
Steps	Open the application
Expected results	The User Interface is fully loaded on the smartwatch's screen and can interact with the ContextManager.java module by sending search requests
Actual results	The User Interface is fully loaded on the smartwatch's screen and can interact with the ContextManager.java module by sending search requests
Test Result	**PASSED**

5.1.8. Shutdowns

Table 140: Verify the shutdown process of the sensors and the weather alarm

Title	Verify the shutdown process of the sensors and the weather alarm
Description	Check if the three sensors and the weather alarm is shutdown properly when user exits the UI

Preconditions	+ The device is connected to the Internet. + Sensors are working normally. + All modules are started up properly. + User is using the application.
Test Data	N/A
Steps	Close the application
Expected results	The weather alarm stops receiving and evaluating weather condition data and the sensors stop sending readings.
Actual results	The weather alarm stops receiving and evaluating weather condition data and the sensors stop sending readings.
Test Result	**PASSED**

Table 141: Verify the shutdown process of all modules

Title	Verify the shutdown process of all modules
Description	Check if all modules in the application are shut down properly.
Preconditions	+ The device is connected to the Internet. + All modules are started up properly. + User is using the application.
Test Data	N/A
Steps	Close the application
Expected results	The ContextManager.java module is shut down first, hence triggering the shutdown of other components and deregistering its subscriptions with Ice Storm
Actual results	The ContextManager.java module is shut down first, hence triggering the shutdown of other components and deregistering its subscriptions with Ice Storm
Test Result	**PASSED**

5.2. Usability Testing

Table 142: Verify the UX/UI design – font family

Title	Verify the UX/UI design – font family
Description	Check if the font used for text are aesthetical and readable by the user
Preconditions	User is using the application
Test Data	N/A
Steps	Look at the text in the application
Expected results	User can understand the text and its instruction clearly and be able to interact with the application accordingly.

Actual results	User can understand the text and its instruction clearly and be able to interact with the application accordingly.
Test Result	**PASSED**

Table 143: Verify the UX/UI design – font size

Title	Verify the UX/UI design – font size
Description	Check if the font size can be easily read by user.
Preconditions	User is using the application
Test Data	N/A
Steps	+ Place the device at the wrist of the hand and put far away at a moderate distance. + Look at the device.
Expected results	The user is still able to read the text in the application
Actual results	The user is still able to read the text in the application
Test Result	**PASSED**

Table 144: Verify the UX/UI design – line spacing

Title	Verify the UX/UI design – line spacing
Description	Check if the vertical space between individual lines is logical
Preconditions	User is using the application
Test Data	N/A
Steps	Look at the vertical spaces between each line
Expected results	The user can differentiate each line and see the text clearly as they don't overlap each other vertically
Actual results	The user can differentiate each line and see the text clearly as they don't overlap each other vertically
Test results	PASS

Table 145: Verify the UX/UI design – main menu text color

Title	Verify the UX/UI design – main menu text color
Description	Check if the font color is reading and pleasing to user's eyes
Preconditions	User is using the application
Test Data	N/A
Steps	Look at the color of the text
Expected results	The color of the text in the main menu is pleasing and has a relatively contrast color compared to the background

Actual results	The color of the text in the main menu is pleasing and has a relatively contrast color compared to the background
Test results	PASS

Table 146: Verify the UX/UI design – warning text color

Title	Verify the UX/UI design – warning text color
Description	Check if the warning message color is stand out from normal text and meets normal standard.
Preconditions	+ User is using the application. + A warning is sent.
Test Data	APO Overexposure warning
Steps	Look at the warning on top of the main menu
Expected results	The color of the warning text stands out from the normal text color (yellow as the most standard color for warning type)
Actual results	The color of the warning text stands out from the normal text color (yellow as the most standard color for warning type)
Test results	PASS

Table 147: Verify the UX/UI design – error text color

Title	Verify the UX/UI design – error text color
Description	Check if the error message color is stand out from normal text and meets normal standard.
Preconditions	+ User is using the application. + User enters an invalid input.
Test Data	N/A
Steps	Look at the error message right under the input just entered
Expected results	The color of the error text stands out from the normal text color (red as the most standard color for error type)
Actual results	The color of the error text stands out from the normal text color (red as the most standard color for error type)
Test results	PASS

Table 148: Verify the UX/UI design – layout

Title	Verify the UX/UI design – layout
Description	Check if the layout of the main menu is logical
Preconditions	User is using the application
Test Data	N/A

Steps	Look at the main menu
Expected results	The main menu should be centered horizontally and vertically so that user can focus on the center of the screen.
Actual results	The main menu should be centered horizontally and vertically so that user can focus on the center of the screen.
Test Result	**PASSED**

5.3. Performance Testing

5.3.1. Load Testing

Table 149: Evaluate the application's performance under a small proportion of user load (100 concurrent users)

Title	Evaluate the application's performance under a small proportion of user load (100 concurrent users)
Description	Check the behavior and response time of the application when there are 100 concurrent users logging in
Preconditions	N/A
Test Data	100 users
Steps	+ Load 100 users into the application's system. + Monitor the behavior of the application.
Expected results	The application directs user to homepage in under 0.25 seconds and the delay time between interactions with UI is minimal
Actual results	The application directs user to homepage in under 0.25 seconds and the delay time between interactions with UI is minimal
Test Result	**PASSED**

Table 150: Evaluate the application's performance under a moderate proportion of user load (500 concurrent users)

Title	Evaluate the application's performance under a moderate proportion of user load (500 concurrent users)
Description	Check the behavior and response time of the application when there are 500 concurrent users logging in
Preconditions	N/A
Test Data	500 users
Steps	+ Load 500 users into the application's system. + Monitor the behavior of the application.
Expected results	The application directs user to homepage in under 0.5 seconds and the delay time between interactions with UI is normal

Actual results	The application directs user to homepage in under 0.5 seconds and the delay time between interactions with UI is normal
Test Result	**PASSED**

Table 151: Evaluate the application's performance under a large proportion of user load (1000 concurrent users)

Title	Evaluate the application's performance under a large proportion of user load (1000 concurrent users)
Description	Check the behavior and response time of the application when there are 1000 concurrent users logging in
Preconditions	N/A
Test Data	1000 users
Steps	+ Load 1000 users into the application's system. + Monitor the behavior of the application.
Expected results	The application directs user to homepage in under 1-2 seconds and the delay time between interactions with UI is slightly noticeable, where it takes around 0.5-1 seconds for a change in the UI to happen
Actual results	The application directs user to homepage in under 1-2 seconds and the delay time between interactions with UI is slightly noticeable, where it takes around 0.5-1 seconds for a change in the UI to happen
Test Result	**PASSED**

Table 152: Evaluate the application's performance under an intense proportion of user load (2000 concurrent users)

Title	Evaluate the application's performance under an intense proportion of user load (2000 concurrent users)
Description	Check the behavior and response time of the application when there are 2000 concurrent users logging in
Preconditions	N/A
Test Data	2000 users
Steps	+ Load 2000 users into the application's system. + Monitor the behavior of the application.
Expected results	The application directs user to homepage in under 5-10 seconds and the delay time between interactions with UI is extremely noticeable, where it takes around 1-3 seconds for a change in the UI to happen.
Actual results	The application directs user to homepage in under 5-10 seconds and the delay time between interactions with UI is extremely noticeable, where it takes around 1-3 seconds for a change in the UI to happen.
Test Result	**PASSED**

Table 153: Evaluate the application's ability to perform query searching function under a small proportion of query load (100 concurrent queries)

Title	Evaluate the application's ability to perform query searching function under a small proportion of query load (100 concurrent queries)
Description	Check the behavior and response time of the application when there are 100 concurrent search queries
Preconditions	Users have logged in and currently at the homepage
Test Data	+ 100 queries + Search query: "Indooroopilly"
Steps	+ Select option 1 in the main menu. + Enter the search query on the search field. + Monitor the behavior of the application.
Expected results	The application returns information about the Indooroopilly Shopping Centre in under 0.25 seconds and the delay time between interactions with UI is minimal
Actual results	The application returns information about the Indooroopilly Shopping Centre in under 0.25 seconds and the delay time between interactions with UI is minimal
Test Result	**PASSED**

Table 154: Evaluate the application's ability to perform query searching function under a moderate proportion of query load (500 concurrent queries)

Title	Evaluate the application's ability to perform query searching function under a moderate proportion of query load (500 concurrent queries)
Description	Check the behavior and response time of the application when there are 500 concurrent search queries
Preconditions	Users have logged in and currently at the homepage
Test Data	+ 500 queries + Search query: "Indooroopilly"
Steps	+ Select option 1 in the main menu. + Enter the search query on the search field. + Monitor the behavior of the application.
Expected results	The application returns information about the Indooroopilly Shopping Centre in under 0.5 seconds and the delay time between interactions with UI is normal
Actual results	The application returns information about the Indooroopilly Shopping Centre in under 0.5 seconds and the delay time between interactions with UI is normal
Test Result	**PASSED**

Table 155: Evaluate the application's ability to perform query searching function under a large proportion of query load (1000 concurrent queries)

Title	Evaluate the application's ability to perform query searching function under a large proportion of query load (1000 concurrent queries)
Description	Check the behavior and response time of the application when there are 1000 concurrent search queries
Preconditions	Users have logged in and currently at the homepage
Test Data	+ 1000 queries + Search query: "Indooroopilly"
Steps	+ Select option 1 in the main menu. + Enter the search query on the search field. + Monitor the behavior of the application.
Expected results	The application returns information about the Indooroopilly Shopping Centre in around 1-2 seconds and the delay time between interactions with UI is slightly noticeable, where it takes around 0.5-1 seconds for a change in the UI to happen
Actual results	The application returns information about the Indooroopilly Shopping Centre in around 1-2 seconds and the delay time between interactions with UI is slightly noticeable, where it takes around 0.5-1 seconds for a change in the UI to happen
Test Result	**PASSED**

Table 156: Evaluate the application's ability to perform query searching function under an intense proportion of query load (2000 concurrent queries)

Title	Evaluate the application's ability to perform query searching function under an intense proportion of query load (2000 concurrent queries)
Description	Check the behavior and response time of the application when there are 2000 concurrent searching queries
Preconditions	Users have logged in and currently at the homepage
Test Data	+ 2000 queries + Search query: "Indooroopilly"
Steps	+ Select option 1 in the main menu. + Enter the search query on the search field. + Monitor the behavior of the application.
Expected results	The application returns information about the Indooroopilly Shopping Centre in around 5-10 seconds and the delay time between interactions with UI is extremely noticeable, where it takes around 1-3 seconds for a change in the UI to happen

Actual results	The application returns information about the Indooroopilly Shopping Centre in around 5-10 seconds and the delay time between interactions with UI is extremely noticeable, where it takes around 1-3 seconds for a change in the UI to happen
Test Result	**PASSED**

5.3.2. Stress Testing

Table 157: Evaluate the application's performance under an extreme proportion of user load (5000 concurrent users)

Title	Evaluate the application's performance under an extreme proportion of user load (5000 concurrent users)
Description	Check the behavior and response time of the application when there are 5000 concurrent users logging in
Preconditions	N/A
Test Data	5000 users
Steps	+ Load 5000 users into the application's system. + Monitor the behavior of the application.
Expected results	The application blocks users from entering the hompage of the application and remain on hold until there is a significant decrease in user load.
Actual results	The application blocks users from entering the hompage of the application and remain on hold until there is a significant decrease in user load.
Test Result	**PASSED**

Table 158: Evaluate the application's performance under a significantly extreme proportion of user load (10000 concurrent users)

Title	Evaluate the application's performance under a significantly extreme proportion of user load (10000 concurrent users)
Description	Check the behavior of the application when there are 10000 concurrent users logging in (breakpoint).
Preconditions	N/A
Test Data	10000 users
Steps	+ Load 10000 users into the application's system. + Monitor the behavior of the application.
Expected results	Upon opening EnviroSmart, the application crashes and redirect users back to the smartwatch's default screen and display an error, implying that the application could not respond
Actual results	Upon opening EnviroSmart, the application crashes and redirect users back to the smartwatch's default screen and display an error, implying that the application could not respond
Test Result	**PASSED**

Table 159: Evaluate the application's ability to perform query searching function under an extreme proportion of query load (5000 concurrent queries)

Title	Evaluate the application's ability to perform query searching function under an extreme proportion of query load (5000 concurrent queries)
Description	Check the behavior of the application when there are 5000 concurrent searching queries
Preconditions	Users have logged in and currently at the homepage
Test Data	+ 5000 queries + Search query: "Indooroopilly"
Steps	+ Select option 1 in the main menu. + Enter the search query on the search field.
Expected results	The application returns a message notifying that the user should try search again in a few minutes and does not return any information about Indooroopilly Shopping Centre
Actual results	The application returns a message notifying that the user should try search again in a few minutes and does not return any information about Indooroopilly Shopping Centre
Test Result	**PASSED**

Table 160: Evaluate the application's ability to perform login function under a significantly extreme proportion of query load (10000 concurrent queries)

Title	Evaluate the application's ability to perform login function under a significantly extreme proportion of query load (10000 concurrent queries)
Description	Check the behavior of the application when there are 10000 concurrent searching queries (breakpoint).
Preconditions	Users have logged in and currently at the homepage
Test Data	10000 users
Steps	+ Select option 1 in the main menu. + Enter the search query on the search field.
Expected results	The application crashes and redirect users to the smartwatch's default screen and display an error, implying that the application could not respond
Actual results	The application crashes and redirect users to the smartwatch's default screen and display an error, implying that the application could not respond
Test Result	**PASSED**

5.3.3. Endurance Testing

Table 161: Verify the application's performance to have an extreme proportion of users (3000 users) using the application continuously for 6 hours

Title	Verify the application's performance to have an extreme proportion of users (3000 users) using the application continuously for 6 hours
Description	Check the behavior of the application when there is a user load of 3000 users for 6 hours straight
Preconditions	N/A
Test Data	3000 users
Steps	+ Load 3000 users into the application's system. + Monitor the behavior of the application for 6 hours.
Expected results	The application behaves slowly and there is a certain delay of 1-3 seconds for UI interaction and more than 5 seconds for any database interactions.
Actual results	The application behaves slowly and there is a certain delay of 1-3 seconds for UI interaction and more than 5 seconds for any database interactions.
Test Result	**PASSED**

Table 162: Verify the application's performance to have an extreme proportion of users (3000 users) using the application continuously for 12 hours

Title	Verify the application's performance to have an extreme proportion of users (3000 users) using the application continuously for 12 hours
Description	Check the behavior of the application when there is a user load of 3000 users for 12 hours straight
Preconditions	N/A
Test Data	3000 users
Steps	+ Load 3000 users into the application's system. + Monitor the behavior of the application for 12 hours.
Expected results	The application behaves slowly and there is a certain delay of more than 1-3 seconds for UI interaction and more than 5 seconds for any database interactions. Furthermore, the application's UI sometimes freeze for a short duration.
Actual results	The application behaves slowly and there is a certain delay of more than 1-3 seconds for UI interaction and more than 5 seconds for any database interactions. Furthermore, the application's UI sometimes freeze for a short duration.
Test Result	**PASSED**

Table 163: Verify the application's performance to have an extreme proportion of users (3000 users) using the application continuously for 24 hours or more

Title	Verify the application's performance to have an extreme proportion of users (3000 users) using the application continuously for 24 hours or more
Description	Check the behavior of the application when there is a user load of 3000 users for 24 hours straight or more
Preconditions	N/A

Test Data	3000 users
Steps	+ Load 3000 users into the application's system. + Monitor the behavior of the application for 24 hours or more.
Expected results	At a certain point, the application completely freezes and crashes, redirecting users to the smartwatch's default screen
Actual results	At a certain point, the application completely freezes and crashes, redirecting users to the smartwatch's default screen
Test Result	**PASSED**

5.3.4. Spike Testing

Table 164: Verify the application's performance to handle a sudden increase in user load

Title	Verify the application's performance to handle a sudden increase in user load
Description	Check the behavior of the application when there is a sudden increase in user load
Preconditions	The user load is 3000 users
Test Data	7000 users
Steps	+ Load an additional of 7000 users into the application's system. + Monitor the behavior of the application.
Expected results	The application suddenly crashes and redirect users to the smartwatch's default screen and display a message indicating that the application could not respond
Actual results	The application suddenly crashes and redirect users to the smartwatch's default screen and display a message indicating that the application could not respond
Test Result	**PASSED**

Table 165: Verify the application's performance when there is a sudden decrease in user load

Title	Verify the application's performance when there is a sudden decrease in user load
Description	Check the behavior of the application when there is a sudden decrease in user load
Preconditions	The user load is 3000 users
Test Data	N/A
Steps	+ Remove 2500 users from the application's system. + Monitor the behavior of the application.
Expected results	The application, from having a certain delay for UI and backend operations, behaves well and has a significant faster response rate.
Actual results	The application, from having a certain delay for UI and backend operations, behaves well and has a significant faster response rate.

Test Result	**PASSED**

5.3.5. Volume Testing

Table 166: Verify the application's performance when there is a small volume of user data in the database

Title	Verify the application's performance when there is a small volume of user data in the database
Description	Check the behavior of the application when there is a volume of 500 users in the database
Preconditions	N/A
Test Data	500 users
Steps	+ Add 500 users into the database. + Monitor the behavior of the application.
Expected results	The application behaves normally and have negligible response time
Actual results	The application behaves normally and have negligible response time
Test Result	**PASSED**

Table 167: Verify the application's performance when there is a moderate volume of user data in the database

Title	Verify the application's performance when there is a moderate volume of user data in the database
Description	Check the behavior of the application when there is a volume of 2000 users in the database
Preconditions	N/A
Test Data	2000 users
Steps	+ Add 2000 users into the database. + Monitor the behavior of the application.
Expected results	The application behaves normally, however, functions and queries in the application become slower
Actual results	The application behaves normally, however, functions and queries in the application become slower.
Test Result	**PASSED**

Table 168: Verify the application's performance when there is a high volume of user data in the database

Title	Verify the application's performance when there is a high volume of user data in the database
Description	Check the behavior of the application when there is a volume of 5000 users in the database
Preconditions	N/A

Test Data	5000 users
Steps	+ Add 5000 users into the database. + Monitor the behavior of the application.
Expected results	The application behaves slowly and there is a noticeable delay when users interacting with the UI and querying
Actual results	The application behaves slowly and there is a noticeable delay when users interacting with the UI and querying.
Test Result	**PASSED**

Table 169: Verify if user data is overwritten if a high volume of user data is added to the database

Title	Verify if user data is overwritten if a high volume of user data is added to the database
Description	Check the user data of the application's database whether they are overwritten or not when there is a high volume of user data added to the database that exceeds the maximum number of users stored in the database (2000 users)
Preconditions	There are a fair number of users stored in the database
Test Data	+ Number of current users in the database: 500 users + Number of added users: 5000 users
Steps	+ Add 2000 users into the database. + Monitor the behavior of the user data in the database.
Expected results	The application crashes and redirects users to the smartwatch's default screen. For user data, the original data of 500 users remain untouched and the additional 2000 users are not stored in the database
Actual results	The application crashes and redirects users to the smartwatch's default screen. For user data, the original data of 500 users remain untouched and the additional 2000 users are not stored in the database.
Test Result	**PASSED**

5.4. Recovery Testing

Table 170: Verify that the data stays the same when the application crashes

Title	Verify that the data stays the same when the application crashes
Description	Check whether if there is any data loss (user data, preferences, etc.) after the application has a crash
Preconditions	+ User has logged into their account. + User has defined preferences.
Test Data	+ Temperature Threshold: 30 Celsius Degree + Temperature Service Type: pool + Username: Jack
Steps	+ Return back to the application after it has crashed.

	+ Go to the preference settings.
Expected results	The temperature preference remains the same with the following data: + Temperature Threshold: 30 Celsius Degree + Temperature Service Type: pool + Username: Jack
Actual results	The temperature preference remains the same with the following data: + Temperature Threshold: 30 Celsius Degree + Temperature Service Type: pool + Username: Jack
Test Result	**PASSED**

Table 171: Verify that the data stays the same when the application is forced to close

Title	Verify that the data stays the same when the application is forced to close
Description	Check whether if there is any data loss (user data, preferences, etc.) after the application is forced to close by the user
Preconditions	+ User has logged into their account. + User has defined preferences. + User has closed the application forcefully.
Test Data	+ Temperature Threshold: 30 Celsius Degree + Temperature Service Type: pool + Username: Jack
Steps	+ Return back to the application after closing. + Go to the preference settings.
Expected results	The temperature preference remains the same with the following data: + Temperature Threshold: 30 Celsius Degree + Temperature Service Type: pool + Username: Jack
Actual results	The temperature preference remains the same with the following data: + Temperature Threshold: 30 Celsius Degree + Temperature Service Type: pool + Username: Jack
Test Result	**PASSED**

Table 172: Verify that the data stays the same when smartwatch has a power failure

Title	Verify that the data stays the same when smartwatch has a power failure
Description	Check whether if there is any data loss (user data, preferences, etc.) when the smartwatch has a sudden power failure
Preconditions	+ User has logged into their account. + User has defined preferences.

Test Data	+ Temperature Threshold: 30 Celsius Degree + Temperature Service Type: pool + Username: Jack
Steps	+ Turn off the power forcefully. + Turn on the power again. + Open the application.
Expected results	The temperature preference remains the same with the following data: + Temperature Threshold: 30 Celsius Degree + Temperature Service Type: pool + Username: Jack
Actual results	The temperature preference remains the same with the following data: + Temperature Threshold: 30 Celsius Degree + Temperature Service Type: pool + Username: Jack
Test Result	**PASSED**

Table 173: Evaluate the recovery behavior of the application when the wireless network signal is lost for a short duration

Title	Evaluate the recovery behavior of the application when the wireless network signal is lost for a short duration
Description	Check how the application behaves when there is a sudden wireless connection lost for a short duration of time (5-10 seconds).
Preconditions	+ User has logged into their account. + Wireless connection is established.
Test Data	N/A
Steps	+ Turn off the wireless connection. + Wait for 5-10 seconds. + Turn on the wireless connection again.
Expected results	A message indicating the connection has been lost in that 5-10 seconds and the application will try to reconnect to the wireless connection. After the cut-off duration, the message disappears, and the application can be used again.
Actual results	A message indicating the connection has been lost in that 5-10 seconds and the application will try to reconnect to the wireless connection. After the cut-off duration, the message disappears, and the application can be used again.
Test Result	**PASSED**

Table 174: Evaluate the recovery behavior of the application when the wireless network signal is lost for a long duration

Title	Evaluate the recovery behavior of the application when the wireless network signal is lost for a long duration
Description	Check how the application behaves when there is a sudden wireless connection lost for a long duration of time (30 – 60 seconds).
Preconditions	+ User has logged into their account. + Wireless connection is established.
Test Data	N/A
Steps	+ Turn off the wireless connection. + Wait for 30 – 60 seconds
Expected results	A message indicating the connection has been lost for around 30 - 60 seconds and the application will try to reconnect to the wireless connection in the meantime. After 60 seconds, the application will need user to login again.
Actual results	The application will try to reconnect to the wireless connection in the meantime. After 60 seconds, the application will require user to login again.
Test Result	**PASSED**

6. User Acceptance Test

6.1. Story: User logs in to the application

6.1.1. Scenario 1: User enters a valid username

Table 175: User enters a valid username.

Title	User enters a valid username
Description	Evaluate the behavior of the application when user successfully login the application with valid username
Preconditions	+ The device is connected to the Internet. + The device's on-screen keyboard is functional. + User has successfully created an account. + The application displays following text: Context-aware EnviroSmart Application Please enter your username:
Test Data	Username: 'Jack'
Steps	+ Type 'Jack' using the on-screen keyboard. + Press "Enter" using the on-screen keyboard.

Expected results	User successfully logins into the application and a main menu will appear as follows: Context-aware EnviroSmart Application Main Menu Please select an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Actual results	A main menu appears, displayed as follows: Context-aware EnviroSmart Application Main Menu Please select an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Test Result	**PASSED**

6.1.2. Scenario 2: User enters an invalid username

Table 176: Username is not registered.

Title	Username is not registered
Description	Evaluate the behavior of the application when user enters an unregistered username
Preconditions	+ The device is connected to the Internet. + The device's on-screen keyboard is functional. + User has not yet created an account. + The application displays the following text: Context-aware EnviroSmart Application Please enter your username:
Test Data	Username: 'Jack1'
Steps	+ Type 'Jack1' using the on-screen keyboard. + Press "Enter" using the on-screen keyboard.
Expected results	The application should return a critical error message, indicating that the username is not valid.
Actual results	The terminal console outputs a java.io.FileNotFoundException, indicating that Jack1 Location file does not exist (No such file or directory)
Test Result	**PASSED**

Table 177: Username is not entered (blank)

Title	Username is not entered (blank)
Description	Evaluate the behavior of the application when the username is leaved blank

Preconditions	+ The device is connected to the Internet. + The device's on-screen keyboard is functional. + The application displays the following text: Context-aware EnviroSmart Application Please enter your username:
Test Data	N/A
Steps	Press "Enter" using the on-screen keyboard without specifying a username
Expected results	The application should return a critical error message, indicating that the username cannot left blank
Actual results	The terminal console outputs a java.io.FileNotFoundException, indicating that Location file does not exist (No such file or directory)
Test Results	**PASSED**

Table 178: Username is entered with less than 3 characters

Title	Username is entered with less than 3 characters
Description	Evaluate the behavior of the application when the username is entered with less than 3 characters
Preconditions	+ The device is connected to the Internet. + The device's on-screen keyboard is functional. + The application displays the following text: Context-aware EnviroSmart Application Please enter your username:
Test Data	Username: 'Al'
Steps	+ Type 'Al' using the on-screen keyboard. + Press "Enter" using the on-screen keyboard.
Expected results	The application displays an error message, indicating that the username text field has to be from 3-30 characters.
Actual results	N/A (since the length validation of user's input is an assumption)
Test Results	**PASSED**

Table 179: Username is entered with more than 30 characters

Title	Username is entered with more than 30 characters
Description	Evaluate the behavior of the application when the username is entered with more than 30 characters
Preconditions	+ The device is connected to the Internet. + The device's on-screen keyboard is functional. + The application displays the following text:

	Context-aware EnviroSmart Application Please enter your username:
Test Data	Username: “Jackinthewonderlandwithjohnnyyy”
Steps	+ Type ‘Jackinthewonderlandwithjohnnyyy’ using the on-screen keyboard. + Press “Enter” using the on-screen keyboard.
Expected results	The application displays an error message, indicating that the username text field has to be from 3-30 characters.
Actual results	N/A (since the length validation of user’s input is an assumption)
Test Result	**PASSED**

Table 180: Username is entered with space(s) in a random location

Title	Username is entered with space(s) in a random location
Description	Evaluate the behavior of the application when the username is entered with space(s) in any location (beginning, middle, end, etc.)
Preconditions	+ The device is connected to the Internet. + The device’s on-screen keyboard is functional. + The application displays the following text: Context-aware EnviroSmart Application Please enter your username:
Test Data	Username: ‘ Jack’
Steps	+ Type ‘ Jack’ using the on-screen keyboard. + Press “Enter” using the on-screen keyboard.
Expected results	The application displays an error message, indicating that the username text field contains at least one space.
Actual results	The terminal console outputs a java.io.FileNotFoundException, indicating that ‘ Location’ file does not exist (No such file or directory)
Test Result	**PASSED**

Table 181: Username is entered with at least one special character

Title	Username is entered with at least one special character
Description	Evaluate the behavior of the application when the username is entered with at least one special character
Preconditions	+ The device is connected to the Internet. + The device’s on-screen keyboard is functional. + The application displays the following text: Context-aware EnviroSmart Application Please enter your username

Test Data	Username: ‘Jack&123**#’
Steps	+ Type ‘Jack&123**#’ using the on-screen keyboard. + Press “Enter” using the on-screen keyboard.
Expected results	The application displays an error message, indicating that the username text field contains at least one special character.
Actual results	The terminal console outputs a java.io.FileNotFoundException, indicating that Jack&123**#Location file does not exist (No such file or directory)
Test Result	**PASSED**

Table 182: Username is entered with number(s) placed at the beginning

Title	Username is entered with number(s) placed at the beginning
Description	Evaluate the behavior of the application when the username is entered with at least one number at the beginning of the field
Preconditions	+ The device is connected to the Internet. + The device’s on-screen keyboard is functional. + The application displays the following text: Context-aware EnviroSmart Application Please enter your username
Test Data	Username: ‘123Jack’
Steps	+ Type ‘123Jack’ using the on-screen keyboard. + Press “Enter” using the on-screen keyboard.
Expected results	The application displays an error message, indicating that the username text field cannot start with a number.
Execution results	The terminal console outputs a java.io.FileNotFoundException, indicating that 123JackLocation file does not exist (No such file or directory)
Test Result	**PASSED**

6.2. Story: User searches for a specific item of interest

6.2.1. Scenario 1: User does not receive information when search for a specific item

Table 183: User searches for a specific item of interest and no results are found

Title	User searches for a specific item of interest and no results are found
Description	Evaluate the behavior of the application when user searches for a specific item of interest (choosing Option 1) in the main menu, but there are no results found by the application
Preconditions	+ The device is connected to the Internet. + The device’s on-screen keyboard is functional.

	+ User has logged into their account. + User has chosen option 1 in the main menu and the application displays the following text: Please enter name of item of interest:
Test Data	Search query: 'Batman Park'
Steps	+ Type 'Batman Park' using the on-screen keyboard. + Press "Enter" using the on-screen keyboard.
Expected results	The application displays the result as follows: No match found for item of interest
Actual results	The application returns a message as the result: No match found for item of interest
Test Result	**PASSED**

Table 184: User does not enter any keywords for query searching

Title	User does not enter any keywords for query searching
Description	Evaluate the behavior of the application when user searches for a specific item of interest (choosing Option 1) in the main menu and does not specify the keyword for the search query
Preconditions	+ The device is connected to the Internet. + The device's on-screen keyboard is functional. + User has logged into their account. + User has chosen option 1 in the main menu and the application displays the following text: Please enter name of item of interest:
Test Data	N/A
Steps	Press "Enter" using the on-screen keyboard without specifying the keyword
Expected results	The application displays the result as follows: No match found for item of interest
Actual results	The application returns a message as the result: No match found for item of interest
Test Result	**PASSED**

6.2.2. Scenario 2: User receives information when search for a specific item

Table 185: User searches for a specific item of interest and the application returns the result

Title	User searches for a specific item of interest and the application returns the result

Description	Evaluate the behavior of the application when user searches for a specific item of interest (choosing Option 1) in the main menu, and the application returns the result information to the user
Preconditions	+ The device is connected to the Internet. + The device's on-screen keyboard is functional. + User has logged into their account. + User has chosen option 1 in the main menu and the application displays the following text: Please enter name of item of interest:
Test Data	Search query: "Crescent Mall"
Steps	+ Type 'Crescent Mall' using the on-screen keyboard. + Press "Enter" using the on-screen keyboard.
Expected results	The application displays the result on the screen as follows: Crescent Mall Shopping Centre is located 10km South of the Ho Chi Minh City central business district (CBD) and includes Banana Republic, Baskin Robins, CGV Cinema, Bobapop and over 130 specialty stores.
Actual results	The application returns a message as the result: Crescent Mall Shopping Centre is located 10km South of the Ho Chi Minh City central business district (CBD) and includes Banana Republic, Baskin Robins, CGV Cinema, Bobapop and over 130 specialty stores.
Test Result	**PASSED**

6.3. Story: User searches for list of items of interest in current location

6.3.1. Scenario 1: User receives a list of items of interest in current location

Table 186: User searches for a list of items of interest in the current location and the application returns the result

Title	User searches for a list of items of interest in the current location and the application returns the result
Description	Evaluate the behavior of the application when user searches for a list of items of interest in current location (choosing Option 2) in the main menu, and the application returns the result information to the user
Preconditions	+ The device is connected to the Internet. + The device's on-screen keyboard is functional. + User has logged into their account. + The application is displaying the menu.

Test Data	User is currently at Location A
Steps	Press option 2 in the main menu: 'Search for a list of items of interest in current location'
Expected results	The application returns a list of items of interest in user's current location (location A) as follows: The following items of interest are in your location: Dam Sen Parklands
Actual results	The application returns a message as the result: The following items of interest are in your location: Dam Sen Parklands
Test Result	**PASSED**

Table 187: User searches for a list of items of interest in the current location, but changes location during the process

Title	User searches for a list of items of interest in the current location, but changes location during the process
Description	Evaluate the behavior of the application when user searches for a list of items of interest in the current location (choose Option 2) in the main menu, and immediately goes to another location in the time being. This can only happen if and only if the user is at the exact crossing line between location A and B.
Preconditions	+ The device is connected to the Internet. + The device's on-screen keyboard is functional. + User has logged into their account. + The application is displaying the menu. + For testing purpose, the coordinate boundaries of both locations are set to minimal.
Test Data	+ Initial Location : A + Final Location : B
Steps	+ Go to the crossing line of 2 locations. + Stay on Location A boundary + Press option 2 in the main menu: 'Search for a list of items of interest in current location' + Immediately changes to Location B
Expected results	The application returns a list of items of interest in user's current location (location A) as follows: The following items of interest are in your location: Dam Sen Parklands Ho Chi Minh City, Downtown

Actual results	The application returns a message as the result: The following items of interest are in your location: Dam Sen Parklands Ho Chi Minh City, Downtown
Test Result	**PASSED**

6.3.2. Scenario 2: User does not receive a list of items of interest in current location

Table 188: User searches for a list of items of interest in the current location and the application does not find any available information

Title	User searches for a list of items of interest in the current location and the application does not find any available information
Description	Evaluate the behavior of the application when user searches for a list of items of interest in current location (choosing Option 2) in the main menu, and the application cannot find available information to the user
Preconditions	+ The device is connected to the Internet. + The device's on-screen keyboard is functional. + User has logged into their account. + The application is displaying the menu
Test Data	N/A
Steps	Press option 2 in the main menu: 'Search for a list of items of interest in current location'
Expected results	The application displays: There are no items of interest in your current location' as the result.
Actual results	The application returns a message as the result: There are no items of interest in your current location.
Test Result	**PASSED**

6.4. Story: User defines the preferences

6.4.1. Scenario 1: User defines the temperature threshold and its service

Table 189: Validate user's input for temperature preferences #1

Title	Validate user's input for temperature preferences #1
Description	Evaluate the behavior of the application when user defines the temperature threshold between 0 – 60 and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).

Test Data	+ Username: Jack + Temperature Threshold: 30 + Temperature Service Type: pool
Steps	+ Open the preference file. + Type "when 30 suggest pool" into one of the three preference slots + Save the preference file.
Expected results	The application accepts the temperature threshold and its corresponding service type.
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when 30 suggest pool
Test Result	**PASSED**

Table 190: Validate user's input for temperature preferences #2

Title	Validate user's input for temperature preferences #2
Description	Evaluate the behavior of the application when user defines the temperature threshold below 0 and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Temperature Threshold: -5 + Temperature Service Type: pool
Steps	+ Open the preference file. + Type "when -5 suggest pool" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the temperature threshold must be from 0 – 60 and ask user to define again
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when -5 suggest pool (no validations for user's input of the preferences are implemented at this point)
Test Result	**FAILED**

Table 191: Validate user's input for temperature preferences #3

Title	Validate user's input for temperature preferences #3
Description	Evaluate the behavior of the application when user defines the temperature threshold above 60 and valid corresponding service type
Preconditions	+ User has already created an account.

	+ The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Temperature Threshold: 102 + Temperature Service Type: pool
Steps	+ Open the preference file. + Type "when 102 suggest pool" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the temperature threshold must be from 0 – 60 and ask user to define again
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when 102 suggest pool (no validations for user's input of the preferences are implemented at this point)
Test Result	**FAILED**

Table 192: Validate user's input for temperature preferences #4

Title	Validate user's input for temperature preferences #4
Description	Evaluate the behavior of the application when user defines the temperature threshold contains alphabetic character(s) and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Temperature Threshold: hello + Temperature Service Type: pool
Steps	+ Open the preference file. + Type "when hello suggest pool" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the temperature threshold must be an integer number and ask user to define again.
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when hello suggest pool (no validations for user's input of the preferences are implemented at this point)
Test Result	**FAILED**

Table 193: Validate user's input for temperature preferences #5

Title	Validate user's input for temperature preferences #5

Description	Evaluate the behavior of the application when user defines the temperature threshold contains special character(s) and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Temperature Threshold: 23#$ + Temperature Service Type: pool
Steps	+ Open the preference file. + Type "when 23#$ suggest pool" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the temperature threshold must be an integer number and ask user to define again
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when 23#$ suggest pool (no validations for user's input of the preferences are implemented at this point)
Test Resul	**FAILED**

Table 194: Validate user's input for temperature preferences #6

Title	Validate user's input for temperature preferences #6
Description	Evaluate the behavior of the application when user defines the temperature threshold contains space(s) and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Temperature Threshold: '23 84' + Temperature Service Type: pool
Steps	+ Open the preference file. + Type "when 23 84 suggest pool" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the temperature threshold must not include a space and ask the user to define again.
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when 23 84 suggest pool (no validations for user's input of the preferences are implemented at this point)
Test Result	**FAILED**

Table 195: Validate user's input for temperature preferences #7

Title	Validate user's input for temperature preferences #7
Description	Evaluate the behavior of the application when user leaves the temperature threshold blank and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Temperature Service Type: pool
Steps	+ Open the preference file. + Type "when suggest pool" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the temperature threshold must be defined and ask user to try again.
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when suggest pool (no validations for user's input of the preferences are implemented at this point)
Test Result	**FAILED**

Table 196: Validate user's input for temperature preferences #8

Title	Validate user's input for temperature preferences #8
Description	Evaluate the behavior of the application when user defines the temperature threshold a non-integer number and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Temperature Threshold: 34.25 + Temperature Service Type: pool
Steps	+ Open the preference file. + Type "when 34.25 suggest pool" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the temperature threshold must be defined as an integer value and ask user to try again.
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when 34.25 suggest pool (no validations for user's input of the preferences are implemented at this point)
Test Result	**FAILED**

Table 197: Validate user's input for temperature preferences #9

Title	Validate user's input for temperature preferences #9
Description	Evaluate the behavior of the application when user defines a valid temperature threshold and an unknown corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Temperature Threshold: 34 + Temperature Service Type: poolemo
Steps	+ Open the preference file. + Type "when 34 suggest poolemo" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the suggested service type for temperature preference is unknown and display a list of available, pre-defined service types. The user is then asked to try again.
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when 34 suggest poolemo (no validations for user's input of the preferences are implemented at this point)
Test Result	**FAILED**

Table 198: Validate user's input for temperature preferences #10

Title	Validate user's input for temperature preferences #10
Description	Evaluate the behavior of the application when user defines a valid temperature threshold and a numeric value for the corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Temperature Threshold: 34 + Temperature Service Type: 23
Steps	+ Open the preference file. + Type "when 34 suggest 23" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that user cannot define the temperature service type as a numeric value. The user is then asked to try again.

Actual results	In the preference file, for user Jack, the following preference can be found: pref: when 34 suggest 23 (no validations for user's input of the preferences are implemented at this point)
Test Result	**FAILED**

Table 199: Validate user's input for temperature preferences #11

Title	Validate user's input for temperature preferences #11
Description	Evaluate the behavior of the application when user defines a valid temperature threshold and leaves the corresponding service type blank
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Temperature Threshold: 34
Steps	+ Open the preference file. + Type "when 34 suggest " into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that user cannot leave the service type blank and display a list of available, pre-defined service types. The user is then asked to try again.
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when 34 suggest (no validations for user's input of the preferences are implemented at this point)
Test Result	**FAILED**

Table 200: Validate user's input for temperature preferences #12

Title	Validate user's input for temperature preferences #12
Description	Evaluate the behavior of the application when user defines a valid temperature threshold and contain space(s) in the corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Temperature Threshold: '34' + Temperature Service Type: pool and playground
Steps	+ Open the preference file.

	+ Type "when 34 suggest pool and playground" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the service type must not include a space and ask the user to define again
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when 34 suggest pool and playground (no validations for user's input of the preferences are implemented at this point)
Test Result	**FAILED**

6.4.2. Scenario 2: User defines the weather alarm and its service

Table 201: Validate user's input for weather preferences #1

Title	Validate user's input for weather preferences #1
Description	Evaluate the behavior of the application when user defines the valid corresponding service type for weather alarm
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Weather Service Type: mall
Steps	+ Open the preference file. + Type "when weather suggest mall" into one of the three preference slots. + Save the preference file.
Expected results	The application accepts the weather threshold and its corresponding service type
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when weather suggest mall
Test Result	**PASSED**

Table 202: Validate user's input for weather preferences #2

Title	Validate user's input for weather preferences #2
Description	Evaluate the behavior of the application when user defines a valid weather value and an unknown corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Weather Service Type: school

Steps	+ Open the preference file. + Type "when weather suggest school" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the suggested service type for weather preference is unknown and display a list of available, pre-defined service types. The user is then asked to try again.
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when weather suggest school (no validations for user's input of the preferences are implemented at this point)
Test Result	**FAILED**

Table 203: Validate user's input for weather preferences #3

Title	Validate user's input for weather preferences #3
Description	Evaluate the behavior of the application when user defines a numeric value for the corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Weather Service Type: 23
Steps	+ Open the preference file. + Type "when weather suggest 23 " into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that user cannot define the temperature service type as a numeric value. The user is then asked to try again.
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when weather suggest 23 (no validations for user's input of the preferences are implemented at this point)
Test Result	**FAILED**

Table 204: Validate user's input for weather preferences #4

Title	Validate user's input for weather preferences #4
Description	Evaluate the behavior of the application when user leaves the corresponding service type blank
Preconditions	+ User has already created an account. + The preference text file is created.

	+ Other preferences are pre-defined (for testing purpose).
Test Data	Username: Jack
Steps	+ Open the preference file. + Type "when weather suggest" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that user cannot leave the service type blank and display a list of available, pre-defined service types. The user is then asked to try again.
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when weather suggest (no validations for user's input of the preferences are implemented at this point)
Test Result	**FAILED**

Table 205: Validate user's input for weather preferences #5

Title	Validate user's input for weather preferences #5
Description	Evaluate the behavior of the application when user defines a valid but outdoor service type for weather alarm
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Weather Service Type: pool
Steps	+ Open the preference file. + Type "when weather suggest pool " into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that user can only define indoor service types and display a list of available, pre-defined indoor service types. The user is then asked to try again.
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when weather suggest pool (no validations for user's input of the preferences are implemented at this point)
Test Result	**FAILED**

Table 206: Validate user's input for weather preferences #6

Title	Validate user's input for weather preferences #6
Description	Evaluate the behavior of the application when user defines a corresponding service type that contain space(s)

Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Weather Service Type: restaurant mall
Steps	+ Open the preference file. + Type "when weather suggest restaurant mall" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the service type must not include a space and ask the user to define again
Actual results	In the preference file, for user Jack, the following preference can be found: pref: when weather suggest restaurant mall (no validations for user's input of the preferences are implemented at this point)
Test Result	**FAILED**

6.4.3. Scenario 3: User defines the APO threshold and its service

Table 207: Validate user's input for APO preferences #1

Title	Validate user's input for APO preferences #1
Description	Evaluate the behavior of the application when user defines the medical condition type between 1 – 3 and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Medical Condition Type: 2 + APO Service Type: restaurant
Steps	+ Open the preference file. + Type "2" into the medical condition type line. + Type "when APO suggest restaurant" into one of the three preference slots + Save the preference file.
Expected results	The application accepts the medical condition type to calculate APO threshold and its corresponding service type.
Actual results	In the preference file, for user Jack, the following content can be found: name: Jack Medical Condition Type: 2 pref: when APO suggest restaurant

Test Result	**PASSED**

Table 208: Validate user's input for APO preferences #2

Title	Validate user's input for APO preferences #2
Description	Evaluate the behavior of the application when user defines the medical condition type below 1 and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Medical Condition Type: -1 + APO Service Type: restaurant
Steps	+ Open the preference file. + Type "-1" into the medical condition type line. + Type "when APO suggest restaurant" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the medical condition type must be from 1 – 3 and ask user to define again
Actual results	In the preference file, for user Jack, the following content can be found: name: Jack Medical Condition Type: -1 pref: when APO suggest restaurant (no validations for user's input of the preferences are implemented at this point)
Test Result	**FAILED**

Table 209: Validate user's input for APO preferences #3

Title	Validate user's input for APO preferences #3
Description	Evaluate the behavior of the application when user defines the medical condition type above 3 and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Medical Condition Type: 6 + APO Service Type: restaurant
Steps	+ Open the preference file. + Type "when APO suggest restaurant" into one of the three preference slots.

	+ Save the preference file.
Expected results	The application displays an error message, indicating that the medical condition type must be from 1 – 3 and ask user to define again
Actual results	In the preference file, for user Jack, the following content can be found: name: Jack Medical Condition Type: 6 pref: when APO suggest restaurant (no validations for user's input of the preferences are implemented at this point)
Test Results	**FAILED**

Table 210: Validate user's input for APO preferences #4

Title	Validate user's input for APO preferences #4
Description	Evaluate the behavior of the application when user defines the medical condition type contains alphabetic character(s) and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Medical Condition Type: hello + APO Service Type: restaurant
Steps	+ Open the preference file. + Type "hello" into the medical condition type line. + Type "when APO suggest restaurant" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the medical condition type must be an integer number and ask user to define again.
Actual results	In the preference file, for user Jack, the following content can be found: name: Jack Medical Condition Type: hello pref: when APO suggest restaurant (no validations for user's input of the preferences are implemented at this point)
Test Result	**FAILED**

Table 211: Validate user's input for APO preferences #5

Title	Validate user's input for APO preferences #5

Description	Evaluate the behavior of the application when user defines the medical condition type contains special character(s) and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Medical Condition Type: 2# + APO Service Type: restaurant
Steps	+ Open the preference file. + Type “2#” into the medical condition type line. + Type “when APO suggest restaurant” into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the medical condition type must be an integer number and ask user to define again
Actual results	In the preference file, for user Jack, the following content can be found: name: Jack Medical Condition Type: 2# pref: when APO suggest restaurant (no validations for user’s input of the preferences are implemented at this point)
Test Result	**FAILED**

Table 212: Validate user’s input for APO preferences #6

Title	Validate user’s input for APO preferences #6
Description	Evaluate the behavior of the application when user defines the medical condition type contains space(s) and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Medical Condition Type: ‘3 2’ + APO Service Type: restaurant
Steps	+ Open the preference file. + Type “3 2” into the medical condition type line. + Type “when APO suggest restaurant” into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the medical condition type must not include a space and ask the user to define again.

Actual results	In the preference file, for user Jack, the following content can be found: name: Jack Medical Condition Type: 3 2 pref: when APO suggest restaurant (no validations for user's input of the preferences are implemented at this point)
Test Result	**FAILED**

Table 213: Validate user's input for APO preferences #7

Title	Validate user's input for APO preferences #7
Description	Evaluate the behavior of the application when user leaves the medical condition type blank and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Medical Condition Type: <blank> + APO Service Type: restaurant
Steps	+ Open the preference file. + Type "when APO suggest restaurant" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the medical condition type must be defined and ask user to try again.
Actual results	In the preference file, for user Jack, the following content can be found: name: Jack Medical Condition Type: pref: when APO suggest restaurant (no validations for user's input of the preferences are implemented at this point)
Test Result	**FAILED**

Table 214: Validate user's input for APO preferences #8

Title	Validate user's input for APO preferences #8
Description	Evaluate the behavior of the application when user defines the medical condition type a numeric but non-integer number and valid corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).

Test Data	+ Username: Jack + Medical Condition Type: 2.2 + APO Service Type: restaurant
Steps	+ Open the preference file. + Type "2.2" into the medical condition type line. + Type "when APO suggest restaurant" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the medical condition type must be defined as an integer value and ask user to try again.
Actual results	In the preference file, for user Jack, the following content can be found: name: Jack Medical Condition Type: 2.2 pref: when APO suggest restaurant (no validations for user's input of the preferences are implemented at this point)
Test Result	**FAILED**

Table 215: Validate user's input for APO preferences #9

Title	Validate user's input for APO preferences #9
Description	Evaluate the behavior of the application when user defines a valid medical condition type and an unknown corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Medical Condition Type: 2 + APO Service Type: restaurantoooo
Steps	+ Open the preference file. + Type "2" into the medical condition type line. + Type "when APO suggest restaurantoooo" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the suggested service type for APO preference is unknown and display a list of available, pre-defined service types. The user is then asked to try again.
Actual results	In the preference file, for user Jack, the following content can be found: name: Jack Medical Condition Type: 2 pref: when APO suggest restaurantoooo

	(no validations for user's input of the preferences are implemented at this point)
Test Result	**FAILED**

Table 216: Validate user's input for APO preferences #10

Title	Validate user's input for APO preferences #10
Description	Evaluate the behavior of the application when user defines a valid medical condition type and a numeric value for the corresponding service type
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Medical Condition Type: 2 + APO Service Type: 23
Steps	+ Open the preference file. + Type "2" into the medical condition type line. + Type "when APO suggest 23" into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that user cannot define the medical service type as a numeric value. The user is then asked to try again.
Actual results	In the preference file, for user Jack, the following content can be found: name: Jack Medical Condition Type: 2 pref: when APO suggest 23 (no validations for user's input of the preferences are implemented at this point)
Test Result	**FAILED**

Table 217: Validate user's input for APO preferences #11

Title	Validate user's input for APO preferences #11
Description	Evaluate the behavior of the application when user defines a valid medical condition type and leaves the corresponding service type blank
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Medical Condition Type: 2
Steps	+ Open the preference file. + Type "2" into the medical condition type line.

	+ Type “when APO suggest <blank>” into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that user cannot leave the service type blank and display a list of available, pre-defined service types. The user is then asked to try again.
Actual results	In the preference file, for user Jack, the following content can be found: name: Jack Medical Condition Type: 2 pref: when APO suggest (no validations for user’s input of the preferences are implemented at this point)
Test Result	**FAILED**

Table 218: Validate user’s input for APO preferences #12

Title	Validate user’s input for APO preferences #12
Description	Evaluate the behavior of the application when user defines a valid medical condition type and a corresponding service type that contain space(s)
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Medical Condition Type: 2 + APO Service Type: restaurant mall
Steps	+ Open the preference file. + Type “2” into the medical condition type line. + Type “when APO suggest restaurant mall ” into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that the service type must not include a space and ask the user to define again.
Actual results	In the preference file, for user Jack, the following content can be found: name: Jack Medical Condition Type: 2 pref: when APO suggest restaurant mall (no validations for user’s input of the preferences are implemented at this point)
Test Result	**FAILED**

Table 219: Validate user’s input for APO preferences #13

Title	Validate user’s input for APO preferences #13

Description	Evaluate the behavior of the application when user defines a valid medical condition type and a valid but outdoor service type for weather alarm
Preconditions	+ User has already created an account. + The preference text file is created. + Other preferences are pre-defined (for testing purpose).
Test Data	+ Username: Jack + Medical Condition Type: 2 + APO Service Type: pool
Steps	+ Open the preference file. + Type "2" into the medical condition type line. + Type "when APO suggest pool " into one of the three preference slots. + Save the preference file.
Expected results	The application displays an error message, indicating that user can only define indoor service types and display a list of available, pre-defined indoor service types. The user is then asked to try again.
Actual results	In the preference file, for user Jack, the following content can be found: name: Jack Medical Condition Type: 2 pref: when APO suggest pool (no validations for user's input of the preferences are implemented at this point)
Test Result	**FAILED**

6.5. Story: User receives warnings and suggestions

6.5.1. Scenario 1: User receives warning and suggestions when only the APO threshold is reached

Table 220: User receives APO Overexposure warning when the air quality is moderate, user is located outdoor and has a medical condition type 1

Title	User receives APO Overexposure warning when the air quality is moderate, user is located outdoor and has a medical condition type 1
Description	Check if the application sends an APO Overexposure warning along with appropriate suggestions of indoor locations when the air quality is moderate. The user is also located outdoor at the moment the threshold is breached and has a medical condition type 1 in the preferences.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.

Test Data	+ Username: Jack + Medical Condition Type: 1 + AQI: 100 => Base Time: 15 seconds. + APO Service Type: bowling
Steps	+ Place the device outdoor + Set received AQI to 100 + Wait for 15 seconds
Expected results	The application displays an APO Overexposure warning above the main menu as well as suggestions for bowling places as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 100 Suggestion – please go to <Bowling 1>, <Bowling 2> Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Actual results	The application returns the content as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 100 Suggestion – please go to <Bowling 1>, <Bowling 2> Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Test Result	**PASSED**

Table 221: User receives APO Overexposure warning when the air quality is unhealthy for sensitive groups, user is located outdoor and has a medical condition type 1

Title	User receives APO Overexposure warning when the air quality is unhealthy for sensitive groups, user is located outdoor and has a medical condition type 1
Description	Check if the application sends an APO Overexposure warning along with appropriate suggestions of indoor locations when the air quality is unhealthy for sensitive groups. The user is also located outdoor at the moment the threshold is breached and has a medical condition type 1 in the preferences.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally.

	+ The application is displaying the main menu.
Test Data	+ Username: Jack + Medical Condition Type: 1 + AQI: 150 => Base Time: 10 seconds. + APO Service Type: bowling
Steps	+ Place the device outdoor + Set received AQI to 150 + Wait for 10 seconds
Expected results	The application displays an APO Overexposure warning above the main menu as well as suggestions for bowling places as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 150 Suggestion – please go to <Bowling 1>, <Bowling 2> Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Actual results	The application returns the content as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 150 Suggestion – please go to <Bowling 1>, <Bowling 2> Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Test Result	**PASSED**

Table 222: User receives APO Overexposure warning when the air quality is unhealthy for everyone, user is located outdoor and has a medical condition type 1

Title	User receives APO Overexposure warning when the air quality is unhealthy for everyone, user is located outdoor and has a medical condition type 1
Description	Check if the application sends an APO Overexposure warning along with appropriate suggestions of indoor locations when the air quality is unhealthy for everyone. The user is also located outdoor at the moment the threshold is breached and has a medical condition type 1 in the preferences.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.

Test Data	+ Username: Jack + Medical Condition Type: 1 + AQI: 200 => Base Time: 5 seconds. + APO Service Type: bowling
Steps	+ Place the device outdoor + Set received AQI to 200 + Wait for 5 seconds
Expected results	The application displays an APO Overexposure warning above the main menu as well as suggestions for bowling places as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 200 Suggestion – please go to <Bowling 1>, <Bowling 2> Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Actual results	The application returns the content as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 200 Suggestion – please go to <Bowling 1>, <Bowling 2> Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Test Result	**PASSED**

Table 223: User receives APO Overexposure warning when the air quality is moderate, user is located outdoor and has a medical condition type 2

Title	User receives APO Overexposure warning when the air quality is moderate, user is located outdoor and has a medical condition type 2
Description	Check if the application sends an APO Overexposure warning along with appropriate suggestions of indoor locations when the air quality is moderate. The user is also located outdoor at the moment the threshold is breached and has a medical condition type 2 in the preferences.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Username: Jack

	+ Medical Condition Type: 2 + AQI: 100 => Base Time: 30 seconds. + APO Service Type: bowling
Steps	+ Place the device outdoor + Set received AQI to 100 + Wait for 30 seconds
Expected results	The application displays an APO Overexposure warning above the main menu as well as suggestions for bowling places as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 100 Suggestion – please go to <Bowling 1>, <Bowling 2> Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit.
Actual results	The application returns the content as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 100 Suggestion – please go to <Bowling 1>, <Bowling 2> Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit.
Test Result	**PASSED**

Table 224: User receives APO Overexposure warning when the air quality is unhealthy for sensitive groups, user is located outdoor and has a medical condition type 2

Title	User receives APO Overexposure warning when the air quality is unhealthy for sensitive groups, user is located outdoor and has a medical condition type 2
Description	Check if the application sends an APO Overexposure warning along with appropriate suggestions of indoor locations when the air quality is unhealthy for sensitive groups. The user is also located outdoor at the moment the threshold is breached and has a medical condition type 2 in the preferences.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.

Test Data	+ Username: Jack + Medical Condition Type: 2 + AQI: 150 => Base Time: 20 seconds. + APO Service Type: cinema
Steps	+ Place the device outdoor + Set received AQI to 150 + Wait for 20 seconds
Expected results	The application displays an APO Overexposure warning above the main menu as well as suggestions for bowling places as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 150 Suggestion – please go to <Bowling 1>, <Bowling 2> Please selection an option: 5. Search for information on a specific item of interest 6. Search for items of interest in current location E. Exit
Actual results	The application returns the content as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 150 Suggestion – please go to <Bowling 1>, <Bowling 2> Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit.
Test Result	**PASSED**

Table 225: User receives APO Overexposure warning when the air quality is unhealthy for everyone, user is located outdoor and has a medical condition type 2

Title	User receives APO Overexposure warning when the air quality is unhealthy for everyone, user is located outdoor and has a medical condition type 2
Description	Check if the application sends an APO Overexposure warning along with appropriate suggestions of indoor locations when the air quality is unhealthy for everyone. The user is also located outdoor at the moment the threshold is breached and has a medical condition type 2 in the preferences.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.

Test Data	+ Username: Jack + Medical Condition Type: 2 + AQI: 200 => Base Time: 10 seconds. + APO Service Type: bowling
Steps	+ Place the device outdoor + Set received AQI to 200 + Wait for 10 seconds
Expected results	The application displays an APO Overexposure warning above the main menu as well as suggestions for bowling places as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 200 Suggestion – please go to <Bowling 1>, Bowling 2> Please selection an option: 5. Search for information on a specific item of interest 6. Search for items of interest in current location E. Exit
Actual results	The application returns the content as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 200 Suggestion – please go to <Bowling 1>, <Bowling 2> Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit.
Test Result	**PASSED**

Table 226: User receives APO Overexposure warning when the air quality is moderate, user is located outdoor and has a medical condition type 3

Title	User receives APO Overexposure warning when the air quality is moderate, user is located outdoor and has a medical condition type 3
Description	Check if the application sends an APO Overexposure warning along with appropriate suggestions of indoor locations when the air quality is moderate. The user is also located outdoor at the moment the threshold is breached and has a medical condition type 3 in the preferences.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Username: Jack + Medical Condition Type: 3

	+ AQI: 100 => Base Time: 45 seconds. + APO Service Type: bowling
Steps	+ Place the device outdoor + Set received AQI to 100 + Wait for 45 seconds
Expected results	The application displays an APO Overexposure warning above the main menu as well as suggestions for bowling places as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 100 Suggestion – please go to <Bowling 1>, <Bowling 2> Please selection an option: 5. Search for information on a specific item of interest 6. Search for items of interest in current location E. Exit
Actual results	The application returns the content as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 100 Suggestion – please go to <Bowling 1>, <Bowling 2> Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit.
Test Result	**PASSED**

Table 227: User receives APO Overexposure warning when the air quality is unhealthy for sensitive groups, user is located outdoor and has a medical condition type 3

Title	User receives APO Overexposure warning when the air quality is unhealthy for sensitive groups, user is located outdoor and has a medical condition type 3
Description	Check if the application sends an APO Overexposure warning along with appropriate suggestions of indoor locations when the air quality is unhealthy for sensitive groups. The user is also located outdoor at the moment the threshold is breached and has a medical condition type 3 in the preferences.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Username: Jack

	+ Medical Condition Type: 3 + AQI: 150 => Base Time: 30 seconds. + APO Service Type: bowling
Steps	+ Place the device outdoor + Set received AQI to 150 + Wait for 30 seconds
Expected results	The application displays an APO Overexposure warning above the main menu as well as suggestions for bowling places as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 150 Suggestion – please go to <Bowling 1>, <Bowling 2> Please selection an option: 5. Search for information on a specific item of interest 6. Search for items of interest in current location E. Exit
Actual results	The application returns the content as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 150 Suggestion – please go to <Bowling 1>, <Bowling 2> Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit.
Test Result	**PASSED**

Table 228: User receives APO Overexposure warning when the air quality is unhealthy for everyone, user is located outdoor and has a medical condition type 3

Title	User receives APO Overexposure warning when the air quality is unhealthy for everyone, user is located outdoor and has a medical condition type 3
Description	Check if the application sends an APO Overexposure warning along with appropriate suggestions of indoor locations when the air quality is unhealthy for everyone. The user is also located outdoor at the moment the threshold is breached and has a medical condition type 3 in the preferences.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Username: Jack + Medical Condition Type: 3

	+ AQI: 200 => Base Time: 15 seconds. + APO Service Type: bowling
Steps	+ Place the device outdoor + Set received AQI to 200 + Wait for 15 seconds
Expected results	The application displays an APO Overexposure warning above the main menu as well as suggestions for bowling places as follows: The application returns the content as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 200 Suggestion – please go to <Bowling 1>, <Bowling 2> Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit.
Actual results	The application returns the content as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 200 Suggestion – please go to <Bowling 1>, <Bowling 2> Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit.
Test Result	**PASSED**

6.5.2. Scenario 2: User receives warning and suggestion when only weather alarm is triggered

Table 229: User receives an extreme weather warning (heavy rain)

Title	User receives an extreme weather warning (heavy rain)
Description	Check if the application sends an extreme weather warning along with appropriate suggestions of indoor locations when the weather is considered heavy rain.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Username: Jack + Weather Condition Type: 1 (heavy rain)

	+ Weather Service Type: cinema
Steps	Set the weather condition type to 1
Expected results	The application displays an extreme weather warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, extreme weather is detected, the current weather event is heavy rain Suggestion – please go to <Cinema 1>, <Cinema 2> Please selection an option: 5. Search for information on a specific item of interest 6. Search for items of interest in current location E. Exit
Actual results	The application returns the content as follows: Context-aware EnviroSmart Application Main Menu Warning, extreme weather is detected, the current weather event is heavy rain Suggestion – please go to <Cinema 1>, <Cinema 2> Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Test Result	**PASSED**

Table 230: User receives an extreme weather warning (hail storm)

Title	User receives an extreme weather warning (hail storm)
Description	Check if the application sends an extreme weather warning along with appropriate suggestions of indoor locations when the weather is considered hail storm.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Username: Jack + Weather Condition Type: 2 (hail storm) + Weather Service Type: cinema
Steps	Set the weather condition type to 2
Expected results	The application displays an extreme weather warning above the main menu as well as suggestions for cinema places as follows:

	Context-aware EnviroSmart Application Main Menu Warning, extreme weather is detected, the current weather event is hail storm Suggestion – please go to <Cinema 1>, <Cinema 2> Please selection an option: 5. Search for information on a specific item of interest 6. Search for items of interest in current location E. Exit
Actual results	The application returns the content as follows: Context-aware EnviroSmart Application Main Menu Warning, extreme weather is detected, the current weather event is hail storm Suggestion – please go to <Cinema 1>, <Cinema 2> Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Test Result	**PASSED**

Table 231: User receives an extreme weather warning (strong wind)

Title	User receives an extreme weather warning (strong wind)
Description	Check if the application sends an extreme weather warning along with appropriate suggestions of indoor locations when the weather is considered strong wind.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Username: Jack + Weather Condition Type: 3 (strong wind) + Weather Service Type: cinema
Steps	Set the weather condition type to 3
Expected results	The application displays an extreme weather warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, extreme weather is detected, the current weather event is strong wind Suggestion – please go to <Cinema 1>, <Cinema 2> Please selection an option:

	5. Search for information on a specific item of interest 6. Search for items of interest in current location E. Exit
Actual results	The application returns the content as follows: Context-aware EnviroSmart Application Main Menu Warning, extreme weather is detected, the current weather event is strong wind Suggestion – please go to <Cinema 1>, <Cinema 2> Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Test Result	**PASSED**

6.5.3. Scenario 3: User receives warning and suggestions when only temperature threshold is reached

Table 232: User receives a temperature warning when the outside temperature is equal or larger than the preference temperature

Title	User receives a temperature warning when the outside temperature is equal or larger than the preference temperature
Description	Check if the application sends a temperature warning along with appropriate suggestions of indoor/outdoor locations when the temperature threshold is breached.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Username: Jack + Outside Temperature: 35 + Preference Temperature: 30 + Temperature Service Type: pool
Steps	Set the received temperature value to 35
Expected results	The application displays a temperature above the main menu as well as suggestions for pool places as follows: Context-aware EnviroSmart Application Main Menu Warning, temperature is now 35 Suggestion – please go to <Pool 1>, <Pool 2> Please selection an option:

	5. Search for information on a specific item of interest 6. Search for items of interest in current location E. Exit
Actual results	The application returns the content as follows: Context-aware EnviroSmart Application Main Menu Warning, temperature is now 35 Suggestion – please go to <Pool 1>, <Pool 2> Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Test Result	**PASSED**

6.5.4. Scenario 4: User receives warning and suggestions when there are at least 2 thresholds reached at the same time

Here, plausible cases are identified in terms of priority among sensors and alarms thresholds in order to display warning as well as suggested items of locations. Table 6 represents a decision table which briefly identify all cases when there are 2 or more thresholds or alarms are reached/triggered at the same time.

Table 233: Priority Decision Table

Triggered threshold	**Two thresholds reached at a time**			**All three thresholds reached at a time**
Weather alarm	Yes	Yes	No	Yes
APO threshold	Yes	No	Yes	Yes
Temperature threshold	No	Yes	Yes	Yes
Priority	Weather	Weather	APO	Weather
Items of locations for suggestion	Indoor	Indoor	Indoor	Indoor

Table 234: User receives an extreme weather warning when both APO threshold and weather alarm are reached/triggered at the same time, and the user is located outdoor

Title	User receives an extreme weather warning when both APO threshold and weather alarm are reached/triggered at the same time, and the user is located outdoor
Description	Check if the application sends an extreme weather warning along with appropriate suggestions of indoor locations when both APO threshold and weather alarm are reached/triggered. The user is located outdoor at the moment the situation occurs

Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Username: Jack + Weather Condition Type: 1 (heavy rain) + Weather Service Type: cinema + Medical Condition Type: 1 + AQI Index: 100 => Base Time: 15 seconds + Medical Service Type: bowling
Steps	+ Place the device outdoor + Set the weather value to 1 + Set received AQI to 100 + Wait for 15 seconds
Expected results	The application displays an extreme weather warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, extreme weather is detected, the current weather event is heavy rain Suggestion – please go to <Cinema 1>, <Cinema 2> Please selection an option: 5. Search for information on a specific item of interest 6. Search for items of interest in current location E. Exit
Actual results	The application returns the content as follows: Context-aware EnviroSmart Application Main Menu Warning, extreme weather is detected, the current weather event is heavy rain Suggestion – please go to <Cinema 1>, <Cinema 2> Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Test Result	**PASSED**

Table 235: User receives an extreme weather warning when both temperature threshold and weather alarm are reached/triggered at the same time

Title	User receives an extreme weather warning when both temperature threshold and weather alarm are reached/triggered at the same time

Description	Check if the application sends an extreme weather warning along with appropriate suggestions of indoor locations when both temperature threshold and weather alarm are reached/triggered.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Username: Jack + Weather Condition Type: 1 (heavy rain) + Weather Service Type: cinema + Temperature Threshold: 30 + Outside Temperature: 35 + Temperature Service Type: pool
Steps	+ Place the device outdoor + Set the weather value to 1 + Set received temperature to 35
Expected results	The application displays an extreme weather warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, extreme weather is detected, the current weather event is heavy rain Suggestion – please go to <Cinema 1>, <Cinema 2> Please selection an option: 5. Search for information on a specific item of interest 6. Search for items of interest in current location E. Exit
Actual results	The application returns the content as follows: Context-aware EnviroSmart Application Main Menu Warning, extreme weather is detected, the current weather event is heavy rain Suggestion – please go to <Cinema 1>, <Cinema 2> Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Test Result	**PASSED**

Table 236: User receives an APO Overexposure warning when both temperature threshold and APO threshold are reached at the same time, and the user is located outdoor

Title	User receives an APO Overexposure warning when both temperature threshold and APO threshold are reached at the same time, and the user is located outdoor
Description	Check if the application sends an APO Overexposure warning along with appropriate suggestions of indoor locations when both thresholds are breached. The user is located outdoor at the moment the situation occurs
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Username: Jack + Medical Condition Type: 1 + AQI Index: 100 => Base Time: 15 seconds + Medical Service Type: bowling + Temperature Threshold: 30 + Outside Temperature: 35 + Temperature Service Type: pool
Steps	+ Place the device outdoor + Set received AQI to 100 + Set received temperature to 35
Expected results	The application displays an APO Overexposure warning above the main menu as well as suggestions for bowling places as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 100 Suggestion – please go to <Bowling 1>, < Bowling 2> Please selection an option: 5. Search for information on a specific item of interest 6. Search for items of interest in current location E. Exit
Actual results	The application returns the content as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 100 Suggestion – please go to <Bowling 1>, < Bowling 2> Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Test Result	**PASSED**

Table 237: User receives an extreme weather warning when all thresholds and alarms are reached/triggered at the same time, and the user is located outdoor

Title	User receives an extreme weather warning when all thresholds and alarms are reached/triggered at the same time, and the user is located outdoor
Description	Check if the application sends an extreme weather warning along with appropriate suggestions of indoor/outdoor locations when all thresholds and alarms are breached/triggered. The user is located outdoor at the moment the situation occurs
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Username: Jack + Weather Condition Type: 1 (heavy rain) + Weather Service Type: cinema + Medical Condition Type: 1 + AQI Index: 100 => Base Time: 15 seconds + Medical Service Type: bowling + Temperature Threshold: 30 + Outside Temperature: 35 + Temperature Service Type: pool
Steps	+ Place the device outdoor + Set weather condition type to 1 + Set received AQI to 100 + Set received temperature to 35
Expected results	The application displays an extreme weather warning above the main menu as well as suggestions for cinema places as follows: Context-aware EnviroSmart Application Main Menu Warning, extreme weather is detected, the current weather event is heavy rain Suggestion – please go to <Cinema 1>, <Cinema 2> Please selection an option: 5. Search for information on a specific item of interest 6. Search for items of interest in current location E. Exit
Actual results	The application returns the content as follows: Context-aware EnviroSmart Application Main Menu Warning, extreme weather is detected, the current weather event is heavy rain

	Suggestion – please go to <Cinema 1>, <Cinema 2> Please selection an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Test Result	**PASSED**

6.5.5. Scenario 5: User does not receive or stop receiving warnings and suggestions

Table 238: The AQI is good, and the user is located outdoor/indoor

Title	The AQI is good, and the user is located outdoor/indoor
Description	Check if the application does not send an APO Overexposure warning along with appropriate suggestions of indoor locations as the threshold has not yet been breached yet regardless of user's location
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Username: Jack + Medical Condition Type: 3 + AQI: 0 + APO Service Type: bowling
Steps	+ Place the device outdoor + Set received AQI to 0
Expected results	The application does not send an APO Overexposure warning and behaves as usual
Actual results	The application returns the content as follows: Context-aware EnviroSmart Application Main Menu Warning, significant air pollution level detected, the current AQI is 0 Suggestion – please go to <Bowling 1>, <Bowling 2> Please selection an option: 3. Search for information on a specific item of interest 4. Search for items of interest in current location E. Exit.
Test Result	**FAILED**

Table 239: The weather condition is normal, and the user is located outdoor/indoor

Title	The weather condition is normal, and the user is located outdoor/indoor

Description	Check if the application does not send an extreme weather warning along with appropriate suggestions of indoor locations as the weather condition is normal
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Username: Jack + Weather Condition Type: 0 (normal) + Weather Service Type: cinema
Steps	+ Place the device outdoor + Set the weather condition type to 0
Expected results	The application does not send an extreme weather warning and behaves as usual
Actual results	The application displays the main menu and no warning is displayed on top of the menu.
Test Result	**PASSED**

Table 240: The recorded temperature has not yet exceeded the threshold, and the user is located outdoor/indoor

Title	The recorded temperature has not yet exceeded the threshold, and the user is located outdoor/indoor
Description	Check if the application does not send a temperature warning along with appropriate suggestions of indoor/outdoor locations as the threshold has not yet been breached yet regardless of user's location
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Username: Jack + Outside Temperature: 15 + Preference Temperature: 30 + Temperature Service Type: pool
Steps	+ Place the device outdoor + Set the recorded temperature value to 20
Expected results	The application does not send a temperature warning and behaves as usual

Actual results	The application displays the main menu and no warning is displayed on top of the menu.
Test Result	**PASSED**

Table 241: User receives temperature warning at first, but the temperature drops below the preference temperature.

Title	User receives temperature warning at first, but the temperature drops below the preference temperature.
Description	Check if the application stops sending temperature warning along with appropriate suggestions of indoor/outdoor locations when there is a change in temperature after user got the temperature warning.
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Username: Jack + Initial Outside Temperature: 25 + Final Outside Temperature: 18 + Preference Temperature: 20 + Temperature Service Type: shops
Steps	+ Place the device outdoor + Set the received temperature value to 35 + Wait until the temperature warning is sent + Set the received temperature value to 28
Expected results	After the temperature warning is displayed on the main menu, the application notices the changes in temperature and remove the warning accordingly.
Actual results	The application displays the main menu and a temperature warning at first, but after a short duration of time, the warning disappears and the application shows the normal main menu
Test Result	**PASSED**

Table 242: User receives extreme weather warning at first, but the weather returns to normal again

Title	User receives extreme weather warning at first, but the weather returns to normal again
Description	Check if the application stops sending extreme weather warning along with appropriate suggestions of indoor locations when there is a change in weather condition
Preconditions	+ The device is connected to the Internet.

	+ User has logged into their account. + User has defined the preferences. + The backend system is working normally. + The application is displaying the main menu.
Test Data	+ Username: Jack + Initial Weather Condition Type: 1 (heavy rain) + Final Weather Condition Type: 0 (normal) + Temperature Service Type: cinema
Steps	+ Set the weather condition type to 1 + Wait until the extreme weather warning is sent + Set the weather condition type to 0
Expected results	After the extreme weather warning is displayed on the main menu, the application notices the changes in weather condition and remove the warning accordingly.
Actual results	The application displays the main menu and a weather warning at first, but after a short duration of time, the warning disappears and the application shows the normal main menu
Test Result	**PASSED**

6.6. Story: User logs out of the application

6.6.1. Scenario 1: User chooses the third option (E) in the main menu

Table 243: User logout of the application

Title	User logout of the application
Description	Check if the user is able to logout of the application when choosing the third option in the main menu
Preconditions	+ The device is connected to the Internet. + User has logged into their account. + User is currently on the main page. + The application displays the following text: Context-aware EnviroSmart Application Main Menu Please select an option: 1. Search for information on a specific item of interest 2. Search for items of interest in current location E. Exit
Test Data	N/A
Steps	Press “E. Exit” (Option E) in the main menu
Expected results	The user is logged out and the application will ask for the username as follows:

	Context-aware EnviroSmart Application Please enter your username:
Actual results	The application displays the following message: Context-aware EnviroSmart Application Please enter your username:
Test Result	**PASSED**

7. Project Planning Report

7.1. Resource Planning

This section provides a systematic list of all resources needed to accomplish a project task. Two major types of resources are recognized – human resource and system/equipment resource.

Table 244: Human Resource

No.	Member Type	Tasks
1.	Test Manager	+ Manage the whole project. + Acquire appropriate resources. + Build up and lead the team to the success of the project. + Apply the appropriate test measurements and metrics in the product and testing team.
2.	Tester	+ Identify and define suitable testing methods, software and automation architecture. + Verify, implement and execute test cases, test program and test suite, etc. + Execute log results, report and fix any available defects.
3.	Test Administrator	+ Establishes and maintains the Test Environment as well as its properties. + Support testers to use the test environment with available technical knowledge.
4.	SQA Members	+ In charge of conducting quality assurance testing. + Confirm whether the testing process meets preliminary expectations and fulfil the test criteria.

** NOTE: One member can be in charge of multiple member types.*

Table 245: System/Equipment Resource

No.	Resources	Description
1.	Test tool/software	+ QATouch is a comprehensive Test Case Management tool for QA teams by simplifying their test management activities and providing an immersive, integrated testing workspace.
2.	Project planning and tracking tools	+ Jira is a proprietary issue tracking product which allows teams to create user stories and issues, plan sprints and distribute tasks. + Cucumber is a collaboration tool that help boosting engineering team's performance by employing Behaviour-Driven Development (BDD).

3.	Network	LAN and Wi-Fi networks are provided by RMIT Vietnam to simulate real business and user environment.
4.	Computer devices	Personal laptops and stationary computers provided by RMIT Vietnam for testing purposes

7.2. Roles and Responsibilities

Table 246: Roles and responsibilities

Role	Name	Tasks
Leader	Tran Dam Quan	+ Assign tasks for members and manage group activities. + Setup template for the report and manage test plan of the project. + Write unit test cases, integration test cases and system test cases. + Finalize works of members in user acceptance test cases.
Member	Nguyen Nguyen Ha Nhan	+ Write unit test cases. + Write user acceptance test cases. + Write test cases on QA Touch. + Estimate the timeline and project planning + Manage group meeting information.
Member	Le Nguyen	+ Write user acceptance test cases. + Write system test cases. + Setup and manage modules on QA Touch.
Member	Luu Huynh Tri	+ Write integration test cases. + Write user acceptance test cases. + Write project planning for the report. + Write and manage test cases on QA Touch.
Member	Ho Minh Duc	+ Write integration test cases. + Write system test cases. + Write and manage test cases on QA Touch.

7.3. Challenges and Solutions

Table 247: Challenges and solutions

Challenges	Solutions
Time management is not thoroughly conducted - Initially, the integration testing phase took a longer period of time than intended to correctly identify the dependencies between individual modules and low-level components. This reduces the amount of time working on the other testings and increase more workloads and effort for team personnel to conduct.	Create multiple soft deadlines to cross check and validate each other's work more often. Moreover, the development team has conducted more offline meetings to ensure that the progress of each individual person is on track.
More unexpected test cases are identified in each testing phase – Throughout each phase, there are some new scenarios and test cases that are developed by each individual member. However, due to lack of coverage in some aspects of the software specifications, it is unsure for team personnel that if those new test cases are applicable to that particular testing phase	For any new and unsure test cases, each personnel had the responsibility to ask other members in the team and a decision would be made upon the total votes. Moreover, the team leader also checked member's works on a weekly basis to ensure that the results meet the preliminary expectations and no abundant test cases were created.
Asynchronous content in test cases – test cases are not consistent where the format, title and the content are different and does not follow any specifications. This leads to huge incoherence between each section which easily makes the report hard to understand and follow by readers	Upon finishing a section or sub-section, one person (usually the leader) is responsible for checking the content and format of each individual test case and make modifications if necessary. This helps reducing worktime and stress than making changes at the end of Stage 1

8. Conclusion

Overall, the qualiy testing report has successfully summarized the overall testing development of the EnviroSmart application in Stage 1. Preliminary software specifications are specifically stated through the conduct of the test plan where strategies (scope, testing type), risk documentation, objectives and test criteria are identified to increase the effiency and reduce management time and workloads of each individual personnel in the development team. Moreover, resource allocation is represented to make sure the development team have enough software and external resources to perform testing, and a Gantt Chart is also included to monitor and create soft deadlines for the team. In specific to the testing type, there are 3 types of testing used in Stage 1, where there are approximately 45 integration test cases, 120 system test cases and 68 user acceptance test cases, summing up to 233 test cases in total. In Integration testing phase, dependencies between 5 modules and multiple low-level components such as text files and sensors are tested to ensure the data transfer process is continuous and in the correct format. The system testing phase tests both functional and non-functional aspect of the application where non-functional includes evaluating the user experience and the performance of the application under certain extreme circustances. Finally, the UAT test cases illustrate how a end-user interacts with the application through different stories and scenarios. The test cases have proved to be conducted successfully as the test exit criteria has been met where the run rate and the pass rate is 100%. Throughout Stage 1, the team has reported to encouter certain issues related to management of time and inconsistent report writing which has resulted in some workloads have not met the intended deadlines, however, the team has come up with the solution to monitor each other’s work more frequently and attend more offline meetings to make sure that everything is on the right track and on time, hence increasing the efficiency and the end result.

9. Appendices

9.1. Appendix A

Meeting Minute No: 1

Meeting Details:

Date:	29/04/2021
Time:	11:30
Duration:	20 minutes
Attendees:	Tran Dam Quan (s3678708)
	Nguyen Nguyen Ha Nhan (s3687637)
	Luu Huynh Tri (s3462315)
	Le Nguyen (s3777242)
Absentee:	Ho Minh Duc (s3694653)
Copy To:	Minh Dinh (Lecturer)

Information /Decision:

Item No.	**Discussion Summary**
1	Briefly read feedback stage 1, identify not good sections which caused loss of points.
2	Discuss technique to start prepare for the unit testing part
3	Assign tasks to each member
4	Next meeting: 05/05/2021

Action Items:

No	**Item**	**Who**	**By**
1	• Identify not good sections which caused loss of points. • The integration part needs to be re-written and added more information for method explanation.	• Tran Dam Quan • Nguyen Nguyen Ha Nhan • Le Nguyen • Luu Huynh Tri	29/04/2021
2	• Download and setup IDE for running the source code. • Discuss technique to start prepare for the unit testing part.	• Tran Dam Quan • Nguyen Nguyen Ha Nhan • Le Nguyen • Luu Huynh Tri	29/04/2021
3	Assign tasks for the current week		05/05/2021
	• Read the source code. • Re-read the lab from week 6 to get the method for doing the unit test.	• Tran Dam Quan • Nguyen Nguyen Ha Nhan • Le Nguyen • Luu Huynh Tri • Ho Minh Duc	

Meeting Minute No: 2

Meeting Details:

Date:	05/05/2021
Time:	16:15
Duration:	15 minutes
Attendees:	Tran Dam Quan (s3678708)
	Nguyen Nguyen Ha Nhan (s3687637)
	Luu Huynh Tri (s3462315)
	Le Nguyen (s3777242)
	Ho Minh Duc (s3694653)
Copy To:	Minh Dinh (Lecturer)

Information /Decision:

Item No.	**Discussion Summary**
1	Assign tasks to each member
2	Next meeting: 11/05/2021

Action Items:

No	**Item**	**Who**	**By**
1	Assign tasks to each member (each part per member)		11/05/2021
	• Create and execute unit tests for support package, preference, weather alarm and location server files in main package (code and report). • Write additional phase of the test plan part in the report.	• Tran Dam Quan	
	• Research about methodologies which can help to execute system test cases.	• Ho Minh Duc	
	• Create and execute unit tests for context manager, all sensors, EnviroAPPUI files in main package (code and report).	• Nguyen Nguyen Ha Nhan	
	• Execute UAT test cases and update actual results in the report. • Re-read integration part and correct mistakes to prepare for execute integration test cases.	• Le Nguyen	
	• Execute UAT test cases and update actual results in the report.	• Luu Huynh Tri	

	• Re-read integration part and correct mistakes to prepare for execute integration test cases.		

Meeting Minute No: 3

Meeting Details:

Date:	11/05/2021
Time:	10:45
Duration:	45 minutes
Attendees:	Tran Dam Quan (s3678708)
	Nguyen Nguyen Ha Nhan (s3687637)
	Luu Huynh Tri (s3462315)
	Le Nguyen (s3777242)
	Ho Minh Duc (s3694653)
Copy To:	Minh Dinh (Lecturer)

Information /Decision:

Item No.	**Discussion Summary**
1	Update the progress of unit test, UAT. Start update the test cases in the report. Start code for integration test.
2	Assign tasks to each member
3	Next meeting: 14/05/2021

Action Items:

No	**Item**	**Who**	**By**
1	• Update the progress of unit test. • Transfer the works in file All Sensors from Nhan to Quan since Quan temporarily finished his unit test part.	• Tran Dam Quan • Nguyen Nguyen Ha Nhan	11/05/2021
	• Update the progress of UAT.	• Le Nguyen • Luu Huynh Tri	
2	Assign tasks to each member (each part per member)		
	• Continue to finish unit tests • Synchronize test code files. • Write report.	• Nguyen Nguyen Ha Nhan • Tran Dam Quan	14/05/2021
	• Execute integration tests. • Write report.	• Le Nguyen • Luu Huynh Tri	14/05/2021

	• Execute system tests. • Write report.	• Tran Dam Quan • Ho Minh Duc	14/05/2021

Meeting Minute No: 4

Meeting Details:

Date:	14/05/2021
Time:	14:15
Duration:	30 minutes
Attendees:	Tran Dam Quan (s3678708)
	Nguyen Nguyen Ha Nhan (s3687637)
	Luu Huynh Tri (s3462315)
	Le Nguyen (s3777242)
	Ho Minh Duc (s3694653)
Copy To:	Minh Dinh (Lecturer)

Information /Decision:

Item No.	Discussion Summary
1	Update unit, integration, system tests
2	Assign tasks to each member

Action Items:

No	Item	Who	By
1	• Update the progress of unit tests. • Update the progress of integration tests. • Update the progress of system tests.	• Tran Dam Quan • Nguyen Nguyen Ha Nhan • Le Nguyen • Luu Huynh Tri • Ho Minh Duc	14/05/2021
2	Assign tasks to each member (continue to work on the current tasks)		
	• Finish unit tests finalize test sections. • Write these in the report, work on the structure of the report (tables, figures, grammar). • Sketch Gantt chart and finalize meeting minutes.	• Nguyen Nguyen Ha Nhan	21/05/2021
	• Finish system, integration tests. • Write these in the report.	• Tran Dam Quan • Ho Minh Duc • Le Nguyen • Luu Huynh Tri	21/05/2021

9.2. Appendix B

9.3. Appendix C

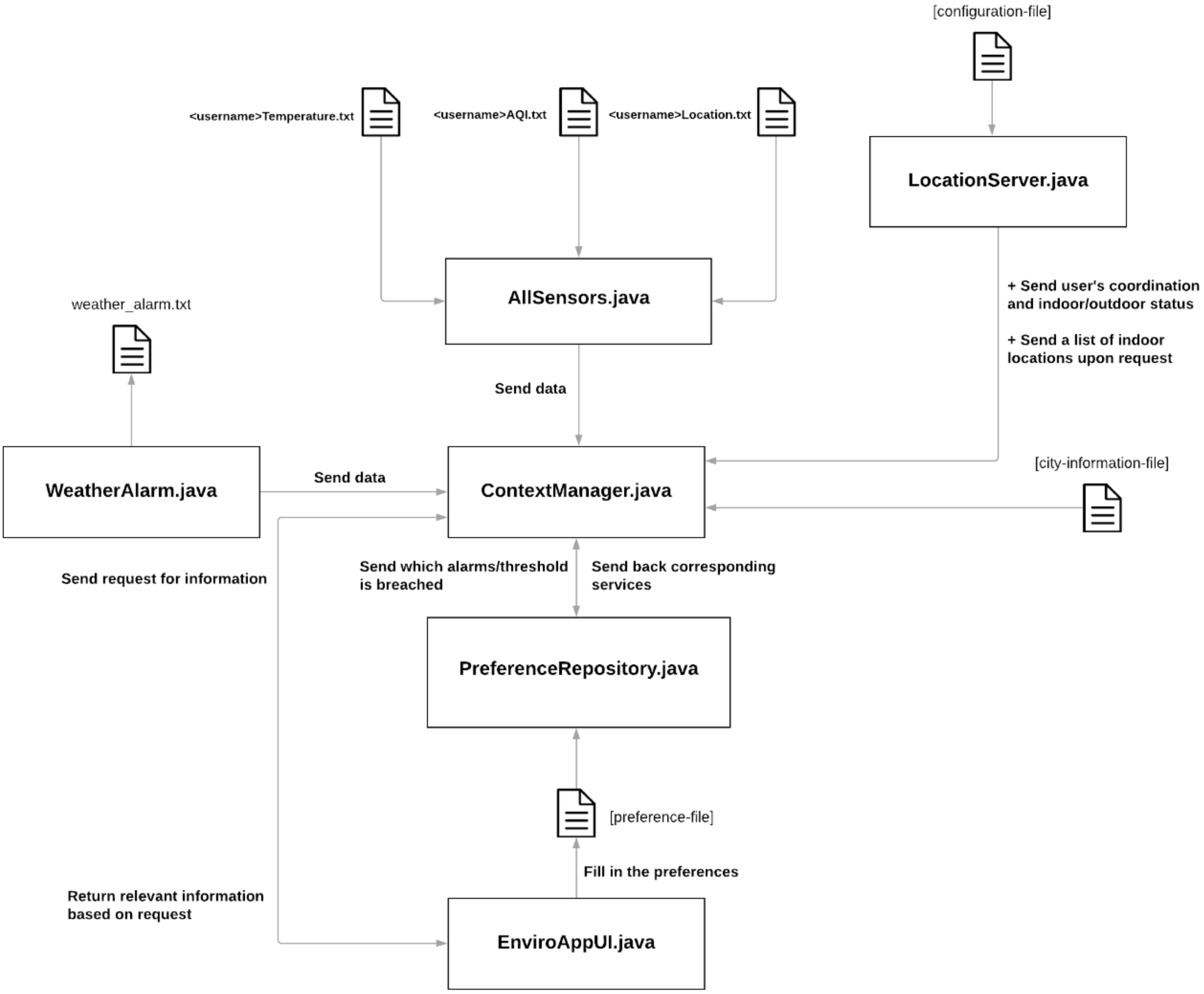

[configuration-file]
<username>Temperature.txt
<username>AQI.txt
<username>Location.txt
LocationServer.java
AllSensors.java
weather_alarm.txt
+ Send user's coordination and indoor/outdoor status
+ Send a list of indoor locations upon request
Send data
WeatherAlarm.java
Send data
ContextManager.java
[city-information-file]
Send request for information
Send which alarms/threshold is breached
Send back corresponding services
PreferenceRepository.java
[preference-file]
Fill in the preferences
Return relevant information based on request
EnviroAppUI.java

Figure 15: EnviroSmart Functional Diagram

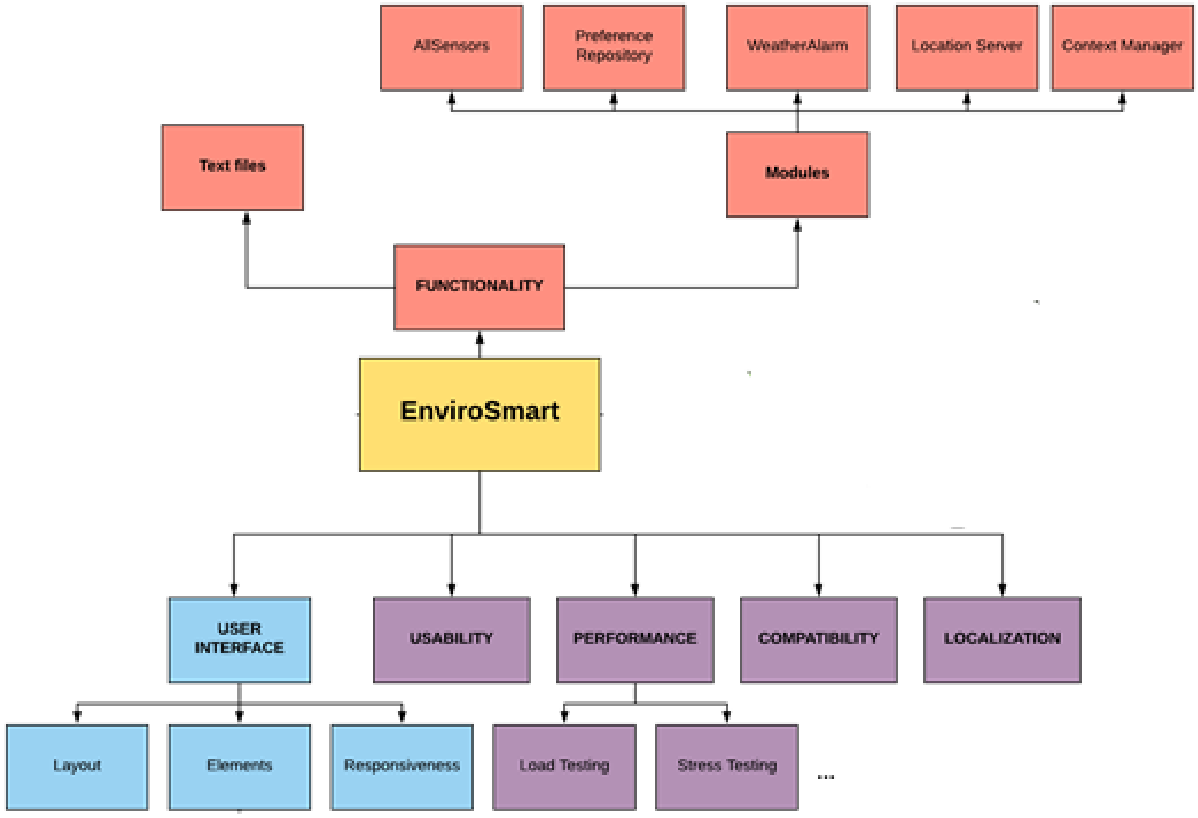
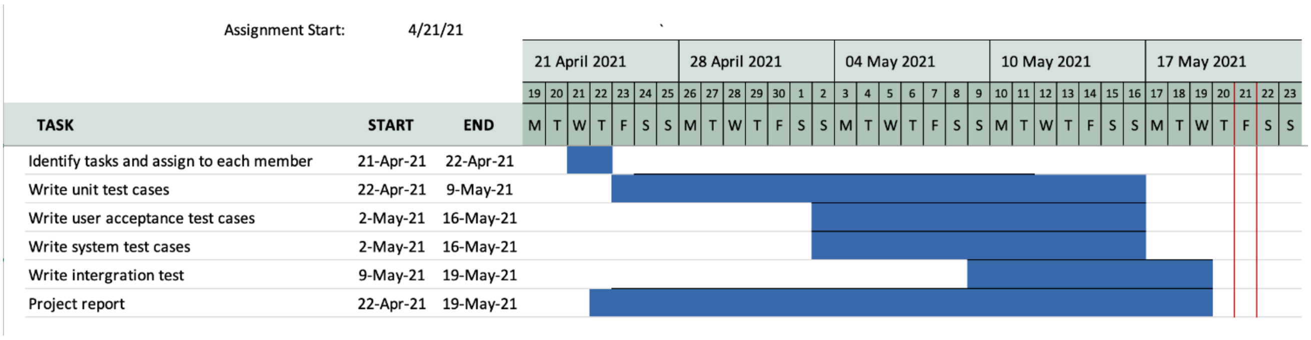


Figure 16: Test Objectives Diagram



Assignment Start: 4/21/21

TASK	START	END
Identify tasks and assign to each member	21-Apr-21	22-Apr-21
Write unit test cases	22-Apr-21	9-May-21
Write user acceptance test cases	2-May-21	16-May-21
Write system test cases	2-May-21	16-May-21
Write intergration test	9-May-21	19-May-21
Project report	22-Apr-21	19-May-21

Figure 17: Timeline of project stage 2