City University of Hong Kong
2024-2025 Semester A
CS3343 Software Engineering Practice

Bug Report
Group 38
Hong Kong Journey Planner

Conducted by:

Name	Position
Fan Tianrui	Project Manager
Wang Fan	Assistant Project Manager
GAO Nanjie	Developing Analyst
CHEUNG Lok Yi	Testing Engineer
LIU Hengche	Program Developer

Department of Computer Science
香港城市大學
City University of Hong Kong

Dec 4, 2024

TABLE OF CONTENT

1. Introduction

In the cycle of software development, bug report is the first step to fix the issue. It is a document that contains the information about the issue, the steps to reproduce the issue, the expected result and the actual result. It also contains the information about the environment in which the issue is found.

In our project, we use Bugzilla as the bug tracking system. Our test team will report the bugs they found in the system. The developers will then fix the bugs and update the status of the bug in the bug tracking system.

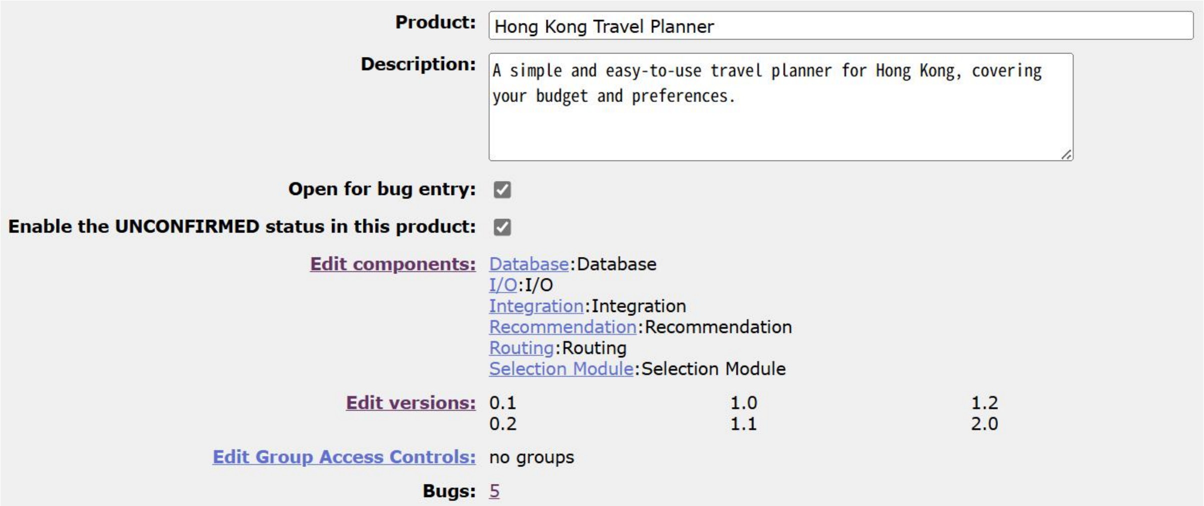
2. Repository

Bugzilla is a web-based bug tracking system that enables software developers to keep track of outstanding bugs in their product. We choose Bugzilla because it is open source and widely used in the industry. With various features like email notification, time tracking, and reporting, Bugzilla is easy to integrate with other tools we use in our project.

The project manager (PM) has the permission to create and modify information of products, components, and versions. Other users can report bugs, search bugs, and view the bug details.

Figure 1 shows the product page of Bugzilla. The product page contains the information about the product, the components of the product, and the versions of the product. The product page also contains the list of bugs reported for the product.

Figure 2 shows the bug details page. The bug details page contains the information about the bug, the steps to reproduce the bug, the expected result, the actual result, and the environment in which the bug is found. The bug details page also contains the comments from the developers and the test team.



Product: Hong Kong Travel Planner

Description: A simple and easy-to-use travel planner for Hong Kong, covering your budget and preferences.

Open for bug entry: ☑

Enable the UNCONFIRMED status in this product: ☑

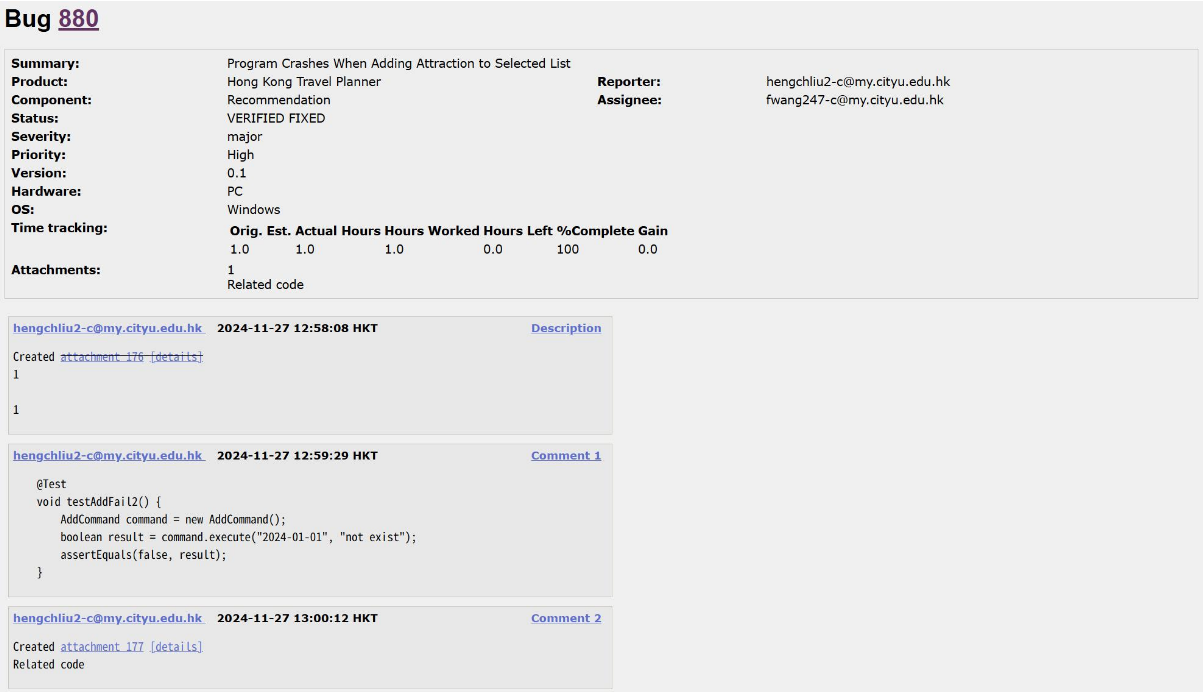
Edit components: Database:Database
I/O:I/O
Integration:Integration
Recommendation:Recommendation
Routing:Routing
Selection Module:Selection Module

Edit versions: 0.1 1.0 1.2
0.2 1.1 2.0

Edit Group Access Controls: no groups

Bugs: 5

Figure 1. Product Page



Bug 880

Summary:	Program Crashes When Adding Attraction to Selected List		
Product:	Hong Kong Travel Planner	**Reporter:**	hengchliu2-c@my.cityu.edu.hk
Component:	Recommendation	**Assignee:**	fwang247-c@my.cityu.edu.hk
Status:	VERIFIED FIXED		
Severity:	major		
Priority:	High		
Version:	0.1		
Hardware:	PC		
OS:	Windows		

Time tracking:

Orig. Est.	Actual Hours	Hours Worked	Hours Left	%Complete	Gain
1.0	1.0	1.0	0.0	100	0.0

Attachments: 1
Related code

hengchliu2-c@my.cityu.edu.hk **2024-11-27 12:58:08 HKT** **Description**

```
Created ~~attachment 176 [details]~~
1

1
```

hengchliu2-c@my.cityu.edu.hk **2024-11-27 12:59:29 HKT** **Comment 1**

```
    @Test
    void testAddFail2() {
        AddCommand command = new AddCommand();
        boolean result = command.execute("2024-01-01", "not exist");
        assertEquals(false, result);
    }
```

hengchliu2-c@my.cityu.edu.hk **2024-11-27 13:00:12 HKT** **Comment 2**

```
Created attachment 177 [details]
Related code
```

Figure 2. Bug Details Page

3. Guidelines

To make the bug report document more effective, the following guidelines is recommended:

3.1. Summary

The bug summary is a short sentence which succinctly describes what the bug is about. It should contain two parts: the problem and the location. The problem is a brief description of the issue. The location is where and when the issue occurs.

3.2. Status & Resolution

The status of the bug indicates the current state of the bug. The initial status of a bug is "Unconfirmed".

"Unconfirmed", "Confirmed", "In Progress" bugs are open bugs.

"Resolved" and "Verified" bugs are closed bugs. Users can reopen a closed bug if they find the issue still exists.

Status	Description
Unconfirmed	Nobody has confirmed that this bug exists.
Confirmed	A developer has confirmed that this bug exists.
In Progress	An assignee has been assigned to work on this bug.
Resolved	The bug has been fixed, awaiting verification.
Verified	The bug has been verified and released in the product.

The resolution of the bug indicates the final state of the bug. The resolution is set when the bug is closed.

Resolution	Description
Fixed	A fix for this bug has been checked into the tree and tested.
Invalid	The problem described is not a bug.
Won't Fix	The problem described is a bug which will never be fixed.
Duplicate	The problem is a duplicate of an existing bug.
Works for Me	Nobody can reproduce the bug as described.

3.3. Severity

The severity of the bug indicates the impact of the bug on the system. The severity is set when the bug is confirmed.

Severity	Description
Major	The bug causes the system to crash, or data loss.
Normal	The bug causes the system to behave incorrectly, but does not crash.
Minor	The bug affects the user experience slightly.
Enhancement	The bug is an enhancement request.

3.4. Priority

The priority of the bug indicates the importance of the bug to be fixed as soon as possible. The priority is set when the bug is confirmed.

Priority	Description
Highest	The bug blocks the development of the system.
High	The bug has the potential to block the development of the system.
Medium	The bug affects the core functionality of the system, but other parts of the system can still be developed.
Low	The bug affects the non-core functionality of the system.
Lowest	The bug is a minor issue that can be fixed later.

3.5. Hardware & OS

Hardware and OS are the environment in which the bug is found. Since compiling and running the project on different hardware and OS may cause different results, it is important to include the hardware and OS information in the bug report to help the developers reproduce the bug.

3.6. Assignee

An assignee is a developer who is assigned to work on the bug. The assignee is responsible for fixing the bug and updating the status of the bug in the bug tracking system.

Usually, the assignee is the developer who is most familiar with the component where the bug is found.

3.7. Description

Description is the content of the bug report. In addition to a detailed description of the bug, the description should also contain the following information:

Expected result. The result that the user expects when the system is working correctly.

Actual result. The result that the user gets when the bug occurs.

Steps to reproduce. The steps that the user takes to reproduce the bug. The steps should be clear and concise so that the developers can easily reproduce the bug.

Additional information. Any other information that is relevant to the bug. This may include screenshots, log files, or any other information that can help the developers fix the bug.

3.8. Comments

Comments is where the developers and the test team can communicate about the bug. The developers can ask for more information about the bug, or provide updates on the status of the bug. The test team can provide more information about the bug, or verify that the bug has been fixed.

4. Selected Bug Reports

4.1. Bug#1: Restaurant Recommender Crashes with IndexOutOfBoundsException

Product:	**Hong Kong Journey Planner**
Component:	Recommendation
Status:	VERIFIED FIXED
Severity:	Major
Priority:	High
Version:	1.0
Hardware:	Machintosh
OS:	Mac OS
Reporter:	Cheung Lok Yi <lycheun24-c@my.cityu.edu.hk>
Assignee:	Liu Hengche <hengchliu2-c@my.cityu.edu.hk>

Cheung Lok Yi 2024-10-17 23:24:12 HKT

Description:

When testing the program, after entering all the user preferences and the recommender starts to work, I encountered an IndexOutOfBoundsException.

I am not sure what caused the error, but it seems to be related to the regions selected by the user. The error occurs when the program tries to get the file paths for a specific day.

Expected Result:

The program should not crash and should provide recommendations based on the user preferences.

Actual Result:

Failure Trace:

```
Exception in thread "main" java.lang.IndexOutOfBoundsException: Index 1 out of
bounds for length 1
        at
java.base/jdk.internal.util.Preconditions.outOfBounds(Preconditions.java:64)
        at
java.base/jdk.internal.util.Preconditions.outOfBoundsCheckIndex(Preconditions.
java:70)
        at
java.base/jdk.internal.util.Preconditions.checkIndex(Preconditions.java:248)
        at java.base/java.util.Objects.checkIndex(Objects.java:374)
        at java.base/java.util.ArrayList.get(ArrayList.java:459)
        at UserPreferences_R.getFilePathsForDay(UserPreferences_R.java:68)
        at Recommender_R.recommendRestaurants(Recommender_R.java:16)
        at Main.main(Main.java:131)
```

Steps to Reproduce:

1. Run Main.java
2. Enter user preferences for the start date, end date, and budget. In my case, the start date is 2024-01-01, the end date is 2024-01-03, and the budget is 100-200.
3. Observe the program crash.

I observed that the program crashes immediately after entering the budget, which makes sense since Restaurant Recommender is called after the user preferences are entered.

Please investigate and fix this issue as soon as possible.

Related Code:

```java
// Main.java
    List<String> regions = new ArrayList<>();
    for (int i = 0; i < days; i++) {
        LocalDate currentDate = startDate.plusDays(i);
        String dateKey = currentDate.format(formatter);
        String selectedRegion = "";
        while (selectedRegion.isEmpty()) {
            System.out.println("Please select a region:");
            System.out.println("1. Hong Kong");
            System.out.println("2. Kowloon");
            System.out.println("3. New Territories");
            System.out.println("4. Outlying Islands");
            int regionChoice = scanner.nextInt();
            scanner.nextLine();
            switch (regionChoice) {
                // ...
            }
        }
        regions.add(selectedRegion);
    }
    dailyRegions.add(regions);
    List<String> validBudgets = Arrays.asList("100-", "100-200", "200-400", 
"400+");
    String budget;
    while (true) {
        // ...
    }
    UserPreferences_R preferences = new UserPreferences_R(startDate, endDate, 
dailyRegions, budget);
    List<Map<String, List<Restaurant>>> Restaurants = 
Recommender_R.recommendRestaurants(preferences);

// UserPreferences_R.java
public class UserPreferences_R {
    private List<List<String>> dailyRegions;
    // ...
    public List<String> getFilePathsForDay(int day, long seed) {
        Map<String, List<String>> regionToFilePaths = new HashMap<>();
        regionToFilePaths.put("Hong Kong", Arrays.asList("Admiralty.json", 
"Causeway Bay.json", "Central.json", // ...
        // ... other regions
        List<String> regions = dailyRegions.get(day);
        List<String> filePaths = new ArrayList<>();
        Random random = new Random(seed);
        for (String region : regions) {
            List<String> paths = regionToFilePaths.get(region);
            if (paths != null && !paths.isEmpty()) {
                List<String> selectedPaths = getRandomSubset(paths, random, 2);
                for (String path : selectedPaths) {
                    filePaths.add("data/" + path);
                }
            }
        }
        return filePaths;
    }
```

Liu Hengche 2024-10-18 10:05:43 HKT

Received your bug report.

Liu Hengche 2024-10-18 10:17:21 HKT

Have you tried different number of days during testing?

Cheung Lok Yi 2024-10-18 11:20:12 HKT

Yes, I have tried with different number of days, including 1, 2, 3, and 4 days. The error does not occur when there is only 1 day.

Liu Hengche 2024-10-18 12:05:43 HKT

Well that probably makes sense. `dailyRegions` is of type `List<List<String>>`, so `dailyRegions.get(day)` will return a `List<String>` for the day.

In `Main.java`, regions in each day is recorded in regions, which is then put into `dailyRegions`. What we exactly want should be `regions.get(day)`, that is, a `String` for the day; instead of `dailyRegions.get(day)`, which is a `List<String>`. `regionToFilePaths.get(region)` accepts a `String` as the key, not a `List<String>`.

I will fix this issue and update you once it is done.

Liu Hengche 2024-10-19 12:28:41 HKT

I have fixed the issue. The program should now work as expected.

I removed `dailyRegions` and used `regions` instead.

```java
// Main.java
    List<String> regions = new ArrayList<>();
    for (int i = 0; i < days; i++) {
        // ...
        regions.add(selectedRegion);
    }
    UserPreferences_R preferences = new UserPreferences_R(startDate, endDate,
regions, budget);

// UserPreferences_R.java
public class UserPreferences_R {
    private List<String> dailyRegions;
    // ...

    public List<String> getFilePathsForDay(int day, long seed) {
        // ...
        String region = regions.get(day);
        List<String> paths = regionToFilePaths.get(region);
        // ...
    }
```

4.2. Bug#2: Fail to Load Data from File due to Incorrect File Path

Product:	**Hong Kong Journey Planner**
Component:	I/O
Status:	VERIFIED FIXED
Severity:	Major
Priority:	High
Version:	1.0
Hardware:	PC
OS:	Windows
Reporter:	Wang Fan <fwang247-c@my.cityu.edu.hk>
Assignee:	Fan Tianrui <tianrufan2-c@my.cityu.edu.hk>

Wang Fan 2024-10-22 16:25:03 HKT

Description:

The program fails to load data from file due to incorrect file path. The file path is hard-coded in the `DataLoader` class. Since it is not the correct path in my environment, the program throws `FileNotFoundException` when trying to load data from the file.

Expected Result:

The program should load data from the file successfully.

Actual Result:

Failure trace is recorded below.

```
java.io.FileNotFoundException: E:\github\CS3343_project\data\scenicspots.json
(系统找不到指定的路径。)
        at java.base/java.io.FileInputStream.open0(Native Method)
        at java.base/java.io.FileInputStream.open(FileInputStream.java:219)
        at java.base/java.io.FileInputStream.<init>(FileInputStream.java:157)
        at
com.fasterxml.jackson.core.TokenStreamFactory._fileInputStream(TokenStreamFact
ory.java:318)
        at
com.fasterxml.jackson.core.JsonFactory.createParser(JsonFactory.java:1219)
        at
com.fasterxml.jackson.databind.ObjectMapper.readValue(ObjectMapper.java:3765)
        at data.DataLoader_S.loadData(DataLoader_S.java:16)
        at data.DataLoader.loadScenicSpots(DataLoader.java:21)
        at planner.TripPlanner.planTrip(TripPlanner.java:42)
        at Main.main(Main.java:15)
```

Steps to Reproduce:

1. Run the program.

2. After all user inputs, the program starts to plan the trip.

3. At the start of this process, the program tries to load data from the file and fails.

Suggested Fix:

Change to relative path in the `DataLoader` class.

As of the current case, it should be changed to `data/scenicspots.json` and `data/shopping.json`.

Related Code:

```
public class DataLoader {
    private static DataLoader instance;

    private DataLoader() {
    }

    public static DataLoader getInstance() {
        if (instance == null) {
            instance = new DataLoader();
        }
        return instance;
    }

    public List<ScenicSpot> loadScenicSpots() {
        IDataLoader<ScenicSpot> loader = new DataLoader_S();
        return
loader.loadData("E:\\github\\CS3343_project\\data\\scenicspots.json");
    }

    public List<Plaza> loadPlazas() {
        IDataLoader<Plaza> loader = new DataLoader_P();
        return
loader.loadData("E:\\github\\CS3343_project\\data\\shopping.json");
    }
}
```

Fan Tianrui 2024-10-22 23:46:06 HKT

Thanks for reporting this issue. Fixed in commit 1d2e92ad.

Wang Fan 2024-10-23 09:24:03 HKT

Thanks for the fix. I have verified that the program can load data from the file successfully now.

4.3. Bug#3: Restaurant Recommender Gives Fixed Recommendations

Product:	**Hong Kong Journey Planner**
Component:	Recommendation
Status:	VERIFIED FIXED
Severity:	Major
Priority:	High
Version:	1.3
Hardware:	Machintosh
OS:	Mac OS
Reporter:	Cheung Lok Yi <lycheun24-c@my.cityu.edu.hk>
Assignee:	Liu Hengche <hengchliu2-c@my.cityu.edu.hk>

Cheung Lok Yi 2024-10-28 16:03:55 HKT

Description:

The recommender system is giving fixed recommendations for the same day and meal time, no matter how many times I tried running the program.

This doesn't make sense since the recommender system should be random and give different recommendations each time the program is run.

Expected Result:

The restaruants should be randomly selected each time the program is run.

Actual Result:

I always get exactly the same list of restaruants for the same day and meal time, like the following:

```
=== 2024-01-01 餐厅选择 ===

日期: 2024-01-01 午餐

1. 幸屋 (幸屋)

2. BRICK LANE (中信大廈)

3. Triple O's by White Spot (太古廣場二座)

4. Simplylife Bakery Cafe (Simplylife Bakery Cafe)

5. SHAKE SHACK (太古廣場)

请输入2024-01-01 午餐的餐厅编号: 


日期: 2024-01-01 晚餐

1. Burgeroom (Fashion Walk)

2. Milu Thai มิลู่ไทย (百樂中心)
```

```
3. HeSheEat (銅鑼灣地帶)

4. Little Vegas (Little Vegas)

5. 吉列牛忌廉烏冬專門店 (百樂中心)

请输入 2024-01-01 晚餐的餐厅编号:
```

Steps to Reproduce:

1. Run the program.

2. Select the same day and meal time.

3. Repeat step 1 and 2.

4. Observe if the list of restaruants is the same each time.

Liu Hengche 2024-10-28 20:24:23 HKT

Related Code:

```java
// UserPreferences_R.java
public List<String> getFilePathsForDay(int day, long seed) {
    Map<String, List<String>> regionToFilePaths = new HashMap<>();

    String region = regions.get(day);
    List<String> filePaths = new ArrayList<>();

    Random random = new Random(seed);

    List<String> paths = regionToFilePaths.get(region);
    if (paths != null && !paths.isEmpty()) {
        List<String> selectedPaths = getRandomSubset(paths, random, 3);
        for (String path : selectedPaths) {
            filePaths.add("data\\Restaurant_Data\\" + path);
        }
    }

    return filePaths;
}

// Recommender_R.java
public List<Map<String, List<Restaurant>>>
recommendByPreferences(UserPreferences_R userPreferences) {
    for (int day = 0; day < userPreferences.getDays(); day++) {
        List<String> filePaths = userPreferences.getFilePathsForDay(day, 1);
    }
```

The problem is that the seed is always set to 1 in the `recommendByPreferences` method. This means that the random number generator will always generate the same random numbers, which will result in the same list of restaurants being selected each time the program is run.

This is previously done to make the program deterministic, but it is not what we want in this case. We should generate a new random seed each time the method is called.

Liu Hengche 2024-10-28 21:55:32 HKT

```
            long randomSeed = System.currentTimeMillis();
            List<String> filePaths = userPreferences.getFilePathsForDay(day,
randomSeed);
```

This should make the recommender system give different recommendations each time the program is run.

In addition, I wrote a test stub class for future JUnit testing:

```
class TestUserPreferences_R extends UserPreferences_R {
    public static void main(String[] args) {
        // ...
        List<String> filePaths = userPreferences.getFilePathsForDay(0, 1);
        System.out.println(filePaths);
    }
}
```

4.4. Bug#4: Tag Input is Not Validated

Product:	**Hong Kong Journey Planner**
Component:	I/O
Status:	VERIFIED FIXED
Severity:	Medium
Priority:	Normal
Version:	1.3
Hardware:	PC
OS:	Windows
Reporter:	Gao Nanjie <nanjiegao2-c@my.cityu.edu.hk>
Assignee:	Fan Tianrui <tianrufan2-c@my.cityu.edu.hk>

Gao Nanjie 2024-11-02 13:08:55 HKT

Description:

When the user inputs an incorrect tag, the program will not handle it correctly. The tags will be directly used to filter attractions, without any validation. This results in the program to filter out all attractions, leaving none for the user to choose from.

Expected Result:

The program should validate the tags input by the user. If the user inputs an incorrect tag, the program should prompt the user to re-enter the tags.

Actual Result:

If you input an incorrect tag, the program will filter out all attractions, leaving none for the user to choose from.

Like this:

```
请输入 2024-01-01 的标签（natural, cultural, activity），或输入回车跳过：

b


=== 2024-01-01 景点选择 ===

日期: 2024-01-01

请输入您选择的景点编号，以空格分隔，或输入回车跳过：
```

Where there should be a list of attractions, but there is none.

Steps to Reproduce:

1. Run the program.

2. When prompted to input tags, input an incorrect tag.

3. Observe the above result after all input is completed.

Related Code:

```
    public String getTagPreference(String dateKey) {
        System.out.println("请输入 " + dateKey + " 的标签（natural, cultural, activity），或输入回车跳过：");
        return scanner.nextLine().toLowerCase();
    }
```

Gao Nanjie 2024-11-02 13:33:45 HKT

Also I found that, the letter case of the tag is not consistent.

The easiest way to fix this is to convert everything to lowercase.

Fan Tianrui 2024-11-02 19:10:23 HKT

Thank you for your report.

Now I have fixed the bug. I have added a check to ensure that only valid tags are used to filter attractions.

```
    public String getTagPreference(String dateKey) {
        List<String> validTags = Arrays.asList("natural", "cultural", "activity");
        System.out.println("请输入 " + dateKey + " 的标签（natural, cultural, activity），或输入回车跳过：");
        String tag = scanner.nextLine().toLowerCase();
        if (!validTags.contains(tag)) {
            System.out.println("输入的标签不正确，请重新输入");
            return getTagPreference(dateKey);
        }
        return tag;
    }
```

4.5. Bug#5: Program Crashes When Adding Attraction to Selected List

Product:	**Hong Kong Journey Planner**
Component:	Selection Module
Status:	VERIFIED FIXED
Severity:	Major
Priority:	Highest
Version:	2.0
Hardware:	Machintosh
OS:	Mac OS
Reporter:	Liu Hengche <hengchliu2-c@my.cityu.edu.hk>
Assignee:	Wang Fan <fwang247-c@my.cityu.edu.hk>

Liu Hengche 2024-11-09 15:55:08 HKT

Description:

When adding an attraction to the selected list, the program crashes with a NullPointerException.

I have looked into the issue and found that the FindAttraction.find(name) method returns null when the attraction is not found. Since null objects have no properties, the program crashes when trying to access the properties of the null object in the Selected.add method.

Expected Result:

The Add command should return false when the attraction is not found, and the program should continue running without crashing.

Actual Result:

```
java.lang.NullPointerException
    at model.Attraction.equals(Attraction.java:29)
    at selection.Selected.add(Selected.java:43)
    at selection.AddCommand.execute(AddCommand.java:13)
    at java.base/java.lang.reflect.Method.invoke(Method.java:566)
    at java.base/java.util.ArrayList.forEach(ArrayList.java:1541)
     at java.base/java.util.ArrayList.forEach(ArrayList.java:1541)
```

Steps to Reproduce:

1. Run the program.
2. Use AddCommand to add an attraction that does not exist.
3. Observe the program crash.

Related Code:

```
// AddCommand.java
public boolean execute(String date, String name) {
        this.date = date;
        this.attraction = FindAttraction.find(name);
        if (Selected.add(date, attraction)) {
            addUndo(this);
            clearRedo();
            return true;
        } else {
            return false;
        }
    }

// Selected.java
    public static boolean add(String key, Attraction attraction) {
        if (!Selected.selected.containsKey(key)) {
            return false;
        }
        for (List<Attraction> list : Selected.selected.values()) {
            for (Attraction a : list) {
                                if (a.getName().equals(attraction.getName())
&& a.getNameZh().equals(attraction.getNameZh())
&& a.getLocation().equals(attraction.getLocation())
&& a.getMetroStation().equals(attraction.getMetroStation())
&& a.getReviewCount() == attraction.getReviewCount()) {
                    return false;
                }
            }
        }
        Selected.selected.get(key).add(attraction);
        return true;
    }
```

Wang Fan 2024-11-09 16:33:12 HKT

Noted with thanks.

However I cannot reproduce the issue. Could you provide the code snippet that causes the crash?

Liu Hengche 2024-11-09 19:12:34 HKT

```
    void testAddFail2() {
        AddCommand command = new AddCommand();
        boolean result = command.execute("2024-01-01", "not exist");
        assertEquals(false, result);
}
```

Wang Fan 2024-11-09 20:01:20 HKT

Oh, I see the problem now. I will fix it.
I incorrectly assumed that you used null as the key in AddCommand.execute, which had been fixed in the previous version. I will add a null check for the attraction object in Selected.add.

Wang Fan 2024-11-09 20:15:45 HKT

```
    public static boolean add(String key, Attraction attraction) {
        if (!Selected.selected.containsKey(key) || attraction == null) {
            return false;
        }
        // ...
```

That should fix the issue.

4.6. Bug#6: Attraction.equals Method Giving Wrong Results

Product:	**Hong Kong Journey Planner**
Component:	Data Model
Status:	VERIFIED FIXED
Severity:	Normal
Priority:	Medium
Version:	2.0
Hardware:	PC
OS:	Linux
Reporter:	Fan Tianrui <tianrufan2-c@my.cityu.edu.hk>
Assignee:	Wang Fan <fwang247-c@my.cityu.edu.hk>

Fan Tianrui 2024-11-10 20:29:00 HKT

Description:

When comparing Attraction objects in the Selected.add method, the Attraction.equals method is called. However, the method is not working as expected.

It always returns true if both objects are of the Attraction class, regardless of the values of their fields. This blocks the Selected.add method from adding new Attraction objects to the selected list, since new attractions are always considered equal to existing ones.

Expected Result:

The Attraction.equals method should correctly compare the fields of two Attraction objects. In the following example, the equals method should return false.

```
        Attraction a = new Attraction("Ngong Ping 360", "昂坪360", "11 Tat
Tung Road, Tung Chung, Lantau Island, Hong Kong, HK", "Tung Chung", 7599);
        Attraction b = new Attraction("Victoria Harbour", "维多利亚港",
"Victoria Harbour, Tsim Sha Tsui, Kowloon, Hong Kong, HK", "Tsim Sha Tsui",
6289);
        System.out.println(a.equals(b)); // false
```

Actual Result:

The equals method returns true.

Steps to Reproduce:

Given in Expected Result.

Related Code:

```
// Selected.java

    public static boolean add(String key, Attraction attraction) {
        if (!Selected.selected.containsKey(key) || attraction == null) {
            return false;
        }
        for (List<Attraction> list : Selected.selected.values()) {
            for (Attraction a : list) {
                if (a.equals(attraction)) {
                    return false;
                }
            }
        }
        Selected.selected.get(key).add(attraction);
        return true;
    }

// Attraction.java

        public boolean equals(Object obj) {
                if (this == obj) {
                        return true;
                }
                if (obj instanceof Attraction) {
                        Attraction other = (Attraction) obj;
        return Objects.equals(name, other.name) &&
                                Objects.equals(nameZh, other.nameZh) &&
                                Objects.equals(location, other.location) &&
                                Objects.equals(metroStation,
other.metroStation) &&
                                reviewCount == other.reviewCount;
                }
                return false;
        }
```

Wang Fan 2024-11-11 10:15:45 HKT

Noted. Have you tried to debug the Attraction.equals method to see if the fields are correctly retrieved?

Fan Tianrui 2024-11-11 11:38:12 HKT

Yes, I have printed the fields in the Attraction.equals method. They all appear as null or 0.

Wang Fan 2024-11-11 11:42:55 HKT

It would be helpful if you could provide the console output in both the Selected.add method and the Attraction.equals method.

Fan Tianrui 2024-11-11 11:45:23 HKT

In Selected.add:

```
Ngong Ping 360 Victoria Harbour

昂坪 360 维多利亚港

11 Tat Tung Road, Tung Chung, Lantau Island, Hong Kong, HK Victoria Harbour, Tsim Sha Tsui, Kowloon, Hong Kong, HK

Tung Chung Tsim Sha Tsui

7599 6289
```

In Attraction.equals:

```
null null

null null

null null

null null

0 0
```

Wang Fan 2024-11-11 12:19:34 HKT

OK I see the issue now. While fields are correctly retrieved in the Selected.add method, they all appear as null or 0 in the Attraction.equals method. This causes all comparisons to be true, and therefore any two Attraction objects are considered equal.

Wang Fan 2024-11-11 19:22:45 HKT

I have tested it out. Simply changing these fields to their getters in the Attraction.equals method should solve the issue.

Why this would happen:

```java
// ScenicSpot.java
public class ScenicSpot extends Attraction {
    // abstract
    private String name;
    private String nameZh;
    private int reviewCount;
    private String location;
    private String metroStation;
    // ...
```

As you can see ScenicSpot overrides the fields of Attraction. However, Attraction.name and ScenicSpot.name are actually two different fields.

Therefore, when direct access is made in Attraction.equals, it retrieves Attractions fields instead of ScenicSpot's, which is not initialized (null or 0). When getters are used, it retrieves the correct fields.

Code after fix:

```
@Override
        public boolean equals(Object obj) {
                if (this == obj) return true;
        if (obj == null || getClass() != obj.getClass()) return false;
        Attraction attraction = (Attraction) obj;
        return Objects.equals(getName(), attraction.getName()) &&
                Objects.equals(getNameZh(), attraction.getNameZh()) &&
                Objects.equals(getLocation(), attraction.getLocation()) &&
                Objects.equals(getMetroStation(),
attraction.getMetroStation());
        }
```

4.7. Bug#7: Feature Request - Redirect Input Stream to Text File

Product:	**Hong Kong Journey Planner**
Component:	I/O
Status:	VERIFIED FIXED
Severity:	Enhancement
Priority:	Low
Version:	2.2
Hardware:	PC
OS:	Windows
Reporter:	Gao Nanjie <nanjiegao2-c@my.cityu.edu.hk>
Assignee:	Fan Tianrui <tianrufan2-c@my.cityu.edu.hk>

Gao Nanjie 2024-11-13 12:20:01 HKT

Description:

For integration testing, it would be useful to redirect the input stream to a text file. This would allow for testing the program with different inputs without manually typing them in.

Expected Result / Suggestions:

Let Main.java accept a command line argument that specifies the path to a text file. Then, the program should read from the text file instead of standard input.

Fan Tianrui 2024-11-13 20:05:43 HKT

I have implemented the feature in commit 74bc57ac. You can now pass the path to a text file as a command line argument to Main.java, e.g. `java Main test/input.txt`.

```
    public static UserInputHandler getInstance(String[] args) { // args from
Main.java
        if (instance == null) {
            synchronized (UserInputHandler.class) {
                if (instance == null) {
                    instance = new UserInputHandler();

                    if (args.length > 0) {
                        try {
                            instance.scanner = new Scanner(new File(args[0]));
                        } catch (FileNotFoundException e) {
                            System.out.println("File not found: " + args[0] +
". Using standard input instead.");
                    }
                }
            }
        } else {
                throw new IllegalStateException("UserInputHandler has already been initialized");
        }
        return instance;
    }
```

Gao Nanjie 2024-11-13 20:15:12 HKT

Thanks for the quick implementation! However, my program is not reading from the file. I have placed the file in the `data` directory under `src`, and I am running the program with `java Main data/test.txt`. What could be the issue?

Fan Tianrui 2024-11-13 21:16:54 HKT

The file path is relative to the working directory. Althogh Main.java is in `/src`, the working directory is the root of the project.

So you need to put the file in the root of the project, or specify the absolute path to the file.

4.8. Bug#8: IndexOutOfBoundsException in Recommender_P.getRandom

Product:	**Hong Kong Journey Planner**
Component:	Recommendation
Status:	VERIFIED FIXED
Severity:	Major
Priority:	High
Version:	2.3
Hardware:	PC
OS:	Windows
Reporter:	Gao Nanjie <nanjiegao2-c@my.cityu.edu.hk>
Assignee:	Wang Fan <fwang247-c@my.cityu.edu.hk>

Gao Nanjie 2024-11-15 17:59:48 HKT

Description:

When testing the program, I encountered an `IndexOutOfBoundsException` in the `Recommender_P.getRandom` method. I have checked the code and found that the problem is caused by the `listSize` parameter. The `listSize` parameter is set to `5`, but `List.subList(int fromIndex, int toIndex)` does not handle the case where toIndex is greater than the size of the list.

Expected Result:

The method returns all plaza objects in the list if there are not enough in the list.

Actual Result:

Failure Trace:

```
Java.lang.IndexOutOfBoundsException: toIndex = 5
        at
java.base/java.util.AbstractList.subListRangeCheck(AbstractList.java:507)
        at java.base/java.util.ArrayList.subList(ArrayList.java:1138)
        at recommendation.Recommender_P.getRandom(Recommender_P.java:78)
        at
recommendation.Recommender_P.recommendByPreferences(Recommender_P.java:43)
        at
recommendation.RecommendationGenerator.generateRecommendations(RecommendationG
enerator.java:44)
        at planner.TripPlanner.planTrip(TripPlanner.java:54)
        at Main.main(Main.java:16)
```

Steps to Reproduce:

1. Run the program.

2. Select "Outlying Islands" as the region. This region has the least number of attractions.

3. Answer "no" to the question "Do you want to visit popular plaza?".

4. After all preferences are set, the program should fail with the IndexOutOfBoundsException.

Related Code:

```
public class Recommender_P implements Recommender<Plaza, UserPreferences_P>{
    private static final double MIN_RATING = 4.0;
    private static final int LIST_SIZE = 5;

    @Override
    public Map<String, List<Plaza>> recommendByPreferences(
        List<Plaza> allPlazas,
        UserPreferences_P userPreference) {

        List<List<Plaza>> recommendations = new ArrayList<>();

        for (int i = 0; i < userPreference.getDuration(); i++) {
            List<Plaza> plazas = new ArrayList<>(allPlazas);
            String region = userPreference.getDayRegion(i);
            List<String> tags = userPreference.getDayTags(i);
            boolean popular = userPreference.getDayPopular(i);
            boolean ratingFilter = userPreference.getDayRatingFilter(i);
            if (region != null) {
                plazas = filterByRegion(plazas, region);
            }
            if (tags != null) {
                plazas = filterByTags(plazas, tags);
            }
            if (ratingFilter) {
                plazas = filterByRating(plazas, MIN_RATING);
            }
            if (popular) {
                plazas = getPopular(plazas, LIST_SIZE);
            } else {
                plazas = getRandom(plazas, LIST_SIZE);
            }
            recommendations.add(plazas);
        }
        return resultMap(recommendations, userPreference);
    }

    private List<Plaza> getRandom(List<Plaza> plazas, int listSize) {
        List<Plaza> copy = new ArrayList<>(plazas);
        Collections.shuffle(copy);
        return copy.subList(0, listSize);
    }
```

Wang Fan 2024-11-15 23:11:59 HKT

Thanks for the detailed report. Now fixed the bug by using the limit method of the Stream API to handle the case where the list size is less than the required size.

```
    private List<Plaza> getRandom(List<Plaza> plazas, int listSize) {
        List<Plaza> copy = new ArrayList<>(plazas);
        Collections.shuffle(copy);
        return copy.stream()
            .limit(listSize)
            .collect(Collectors.toList());
    }
```

4.9. Bug#9: Feature Request - Plaza Recommender Should Return Nonempty

Product:	**Hong Kong Journey Planner**
Component:	Data Model
Status:	VERIFIED FIXED
Severity:	Enhancement
Priority:	Low
Version:	3.0
Hardware:	PC
OS:	Linux
Reporter:	Fan Tianrui <tianrufan2-c@my.cityu.edu.hk>
Assignee:	Liu Hengche <hengchliu2-c@my.cityu.edu.hk>

Fan Tianrui 2024-11-18 15:24:03 HKT

Description:

Although this is pretty much not a bug, I still want to report it. The Plaza Recommender may recommend zero entries. This is because the Plaza Recommender may filter out all the plazas based on the user's preferences.

Because this would affect the route planning, I recommend that the Plaza Recommender should always return at least one plaza.

Expected Result:

The algorithm should always return at least one plaza, and notify the user that there are no plazas that meet the user's preferences.

Actual Result:

The algorithm may return zero plazas, which is not expected.

Here's the console output:

```
=== 2024-01-01 商场选择 ===

日期: 2024-01-01

请输入您选择的商场编号，以空格分隔，或输入回车跳过：
```

While it is expected to print a list of plazas and wait for user to choose one, it prints nothing.

Steps to Reproduce:

1. Run the program.

2. For the user preferences, choose mutually exclusive tags (e.g. "complex" and "specialty"). Therefore, there are no plazas that meet the user's preferences.

3. After all preferences are set, the program will generate a plan but with zero plazas.

Related Code:

```
    @Override
    public Map<String, List<Plaza>> recommendByPreferences(
        List<Plaza> allPlazas,
        UserPreferences_P userPreference) {

        List<List<Plaza>> recommendations = new ArrayList<>();

        for (int i = 0; i < userPreference.getDuration(); i++) {
            List<Plaza> plazas = new ArrayList<>(allPlazas);
            String region = userPreference.getDayRegion(i);
            List<String> tags = userPreference.getDayTags(i);
            boolean popular = userPreference.getDayPopular(i);
            boolean ratingFilter = userPreference.getDayRatingFilter(i);
            if (region != null) {
                plazas = filterByRegion(plazas, region);
            }
            if (tags != null) {
                plazas = filterByTags(plazas, tags);
            }
            if (ratingFilter) {
                plazas = filterByRating(plazas, MIN_RATING);
            }
            if (popular) {
                plazas = getPopular(plazas, LIST_SIZE);
            } else {
                plazas = getRandom(plazas, LIST_SIZE);
            }
            recommendations.add(plazas);
        }
        return resultMap(recommendations, userPreference);
    }
```

Liu Hengche 2024-11-18 16:12:20 HKT

Received. I think the default logic should be changed to always return at least one plaza. I will fix this issue.

Liu Hengche 2024-11-18 16:35:39 HKT

Fixed.

```
            if (plazas.size() == 0) {
                plazas = filterByRegion(allPlazas, region);
                if (plazas.size() == 0) {
                    plazas = allPlazas;
                }
                plazas = getRandom(plazas, LIST_SIZE);
            }
```

Fan Tianrui 2024-11-18 19:24:03 HKT

I don’t really understand why is the guaranteed plaza chosen randomly.

Liu Hengche 2024-11-18 20:13:54 HKT

Yeah, your suggestion makes sense. Please check the following version:

```
if (plazas.size() == 0) {
    plazas = filterByRegion(allPlazas, region);
    if (plazas.size() == 0) {
        plazas = allPlazas;
    }
    if (popular) {
        plazas = getPopular(plazas, LIST_SIZE);
    } else {
        plazas = getRandom(plazas, LIST_SIZE);
    }
}
```

Fan Tianrui 2024-11-18 21:24:03 HKT

Yes, that's better. Thanks for the fix.

4.10. Bug#10: Singleton Class Does Not Work in JUnit Test

Product:	**Hong Kong Journey Planner**
Component:	Recommendation
Status:	VERIFIED FIXED
Severity:	Major
Priority:	Highest
Version:	3.3
Hardware:	Machintosh
OS:	Mac OS
Reporter:	Cheung Lok Yi <lycheun24-c@my.cityu.edu.hk>
Assignee:	Wang Fan <fwang247-c@my.cityu.edu.hk>

Cheung Lok Yi 2024-11-20 17:08:35 HKT

Description:

When running a single JUnit test class, the singleton class works as expected. However, when running all the test classes in the project, the program fails with an exception.

Despite the singleton class is thread-safe, the program still fails when running multiple test classes. Based on the failure trace, we can see that the program trys to initialize the singleton class multiple times, which is not expected.

Expected Result:

All test classes should be able to run without any exceptions, and does not affect each other.

Actual Result:

```
java.lang.IllegalStateException: Selected has already been initialized.
        at selection.Selected.getInstance(Selected.java:21)
        at test.AddCommandTest.init(AddCommandTest.java:37)
        at java.base/java.lang.reflect.Method.invoke(Method.java:566)
        at java.base/java.util.ArrayList.forEach(ArrayList.java:1541)
```

Steps to Reproduce:

1. In Eclipse, right click on the test folder and select "Run As" -> "JUnit Test".

2. Observe the test result. Most test cases should pass, but some of them related to the singleton class will result in an error (rather than a failure).

Related Code:

```
    public static synchronized Selected getInstance(Map<String,
List<Attraction>> selected) {
        if (instance == null) {
            instance = new Selected(selected);
        } else {
```

```java
            throw new IllegalStateException("Selected has already been 
initialized.");
        }
        return instance;
    }
```

Wang Fan 2024-11-20 20:08:23 HKT

I can only come up with a temporary workaround for this issue. I can add another method that should only be called by test classes exclusively. This method will reset the singleton instance to null, so that the singleton class can be re-initialized.

```java
    public static synchronized void resetInstance() {
        instance = null;
    }
```

Cheung Lok Yi 2024-11-20 21:24:51 HKT

I still think that we should preserve the invariant of the singleton class. I find a blog related to this issue, please take a look at it.

https://blog.davidehringer.com/testing/test-driven-development/unit-testing-singletons/

Wang Fan 2024-11-20 21:58:34 HKT

Thanks for the reference.

Wang Fan 2024-11-20 23:08:40 HKT

The solution is to use reflection to reset the instance without exposing the method to the production code. I have packaged the code into a utility class.

```java
public class TestReflectUtils {
    public static void resetSelected() throws SecurityException, 
NoSuchFieldException, IllegalArgumentException, IllegalAccessException {
        Field selectedInstance = Selected.class.getDeclaredField("instance");
        selectedInstance.setAccessible(true);
        selectedInstance.set(null, null);
    }
}
```

Now we can call this method in the `@BeforeAll` method of the test class.

5. Appendix

Appendix A: Bug Report Template

Summary:

Product:	**Hong Kong Journey Planner**
Component:	
Status:	<Unconfirmed/Confirmed/In Progress>
Severity:	<Major/Medium/Minor/Enhancement>
Priority:	<Highest/High/Normal/Low/Lowest>
Version:	
Hardware:	
OS:	
Reporter:	
Assignee:	

<Name> <Time>

Description:

A brief description of how bug happens.

Expected Result:

Expected outcome.

Actual Result:

Actual outcome. Provide logs and/or console outputs if possible.

Steps to Reproduce:

1. Run ……
2. Input ……
3. Observe ……

Related Code:

If you can locate where the problem is, provide as attachments.