Code Review Report

The project source code and its classes have been analyzed by static code review tool FindBugs and below are the results and descriptions from the analysis.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SlNo** | **Problem** | **Severity** | **Defect type** | **Finder** |
| 1 | [Method may fail to close stream](http://findbugs.sourceforge.net/bugDescriptions.html#OS_OPEN_STREAM) | Minor | Coding standards - Missing | Hari |
| 2 | [Reliance on default encoding](http://findbugs.sourceforge.net/bugDescriptions.html#DM_DEFAULT_ENCODING) | Minor | Coding standards - Missing | Hari |
| 3 | Is the correct form of storage being used and is memory available for storage? | Open Question | Data Structure - Missing | Apeksha |
| 4 | Are the layouts suitable for all the screen dimensions? | Major | User Interface - Missing | Hari |
| 5 | Fields CALENDAR\_COLS and EVENT\_COLS should be package protected. | Major | Security | Apeksha |
| 6 | [Incorrect lazy initialization of static field](http://findbugs.sourceforge.net/bugDescriptions.html#LI_LAZY_INIT_STATIC). | Minor | Coding standards - Missing | Apeksha |
| 7 | Writing to fields SERV\_STATE, thisObject, lastLocation and appContext from instance method | Minor | Coding standards - Wrong | Hari |
| 8 | Logging or debugging code present? | Minor | Performance -Wrong | Hari |
| 9 | [Load of known null value](http://findbugs.sourceforge.net/bugDescriptions.html#NP_LOAD_OF_KNOWN_NULL_VALUE) for resultList | Minor | Coding standards - Wrong | Hari |
| 10 | Default case missing in switch statements | Minor | Coding standards - Wrong | Apeksha |
| 11 | Field ‘resultList’ dereferenced without null check | Major | Coding standards - Wrong | Apeksha |

**Problem Description**

[**Method may fail to close stream**](http://findbugs.sourceforge.net/bugDescriptions.html#OS_OPEN_STREAM)

Java file and line number: com.se.profilebuddy.adapter.PlacesAutoCompleteAdapter.java (line 108)

Description: The ‘autoComplete’ method creates an IO stream object at line 124, does not assign it to any fields, pass it to other methods that might close it, or return it, and does not appear to close the stream on all paths out of the method.  This may result in a file descriptor leak.  It is generally a good idea to use a finally block to ensure that streams are closed. The Reader needs to be closed. [1]

[**Reliance on default encoding**](http://findbugs.sourceforge.net/bugDescriptions.html#DM_DEFAULT_ENCODING)

Java file and line number: com.se.profilebuddy.adapter.PlacesAutoCompleteAdapter.java (line 124)

Description: Call to a method ‘InputStreamReader(InputStream)’ which will perform a byte to String (or String to byte) conversion, and will assume that the default platform encoding is suitable. This will cause the application behavior to vary between platforms. Use an alternative API and specify a charset name or Charset object explicitly. [1]

**Fields CALENDAR\_COLS and EVENT\_COLS should be package protected**

Java file and line number: com.se.profilebuddy.action.CalendarHelper.java (lines: 39 and 43)

Description: Fields ‘CALENDAR\_COLS’ and ‘EVENT\_COLS’ could be changed by malicious code or by accident. The field could be made package protected to avoid this vulnerability. [1]

[**Incorrect lazy initialization of static field**](http://findbugs.sourceforge.net/bugDescriptions.html#LI_LAZY_INIT_STATIC)

Java file and line number: com.se.profilebuddy.action.EventHelper (lines: 52 – 54)

Description: Lazy [Incorrect lazy initialization of static field](http://findbugs.sourceforge.net/bugDescriptions.html#LI_LAZY_INIT_STATIC) in method ‘getInstance(Context)’, Because the compiler or processor may reorder instructions, threads are not guaranteed to see a completely initialized object, if the method can be called by multiple threads. You can make the field volatile to correct the problem. Fields should be checked to not null to resolve this issue. [1]

**Writing to fields SERV\_STATE, thisObject, lastLocation and appContext from instance method**

Java file and line number: com.se.profilebuddy.service.LocationService.java (line 61, 73, 135), com.se.profilebuddy.activity.MainActivity.java (line: 48)

Description: The fields mentioned are written from instance methods which is generally a bad practice.

[**Load of known null value**](http://findbugs.sourceforge.net/bugDescriptions.html#NP_LOAD_OF_KNOWN_NULL_VALUE) **for variable ‘resultList’**

Java file and line number: com.se.profilebuddy.adapter.PlacesAutoCompleteAdapter.java (line: 138 and 143)

Description: The variable ‘resultList’ referenced at this point is known to be null due to an earlier check against null. Although this is valid, it might be a mistake (perhaps you intended to refer to a different variable, or perhaps the earlier check to see if the variable is null should have been a check to see if it was non-null). [1]

**Default case missing in switch statements**

Java file and line number: com.se.profilebuddy.action.CalendarAction.java (line: 184 - 193), com.se.profilebuddy.action.ProfileAction.java (line: 93 - 102)

Description: Methods ‘changeProfile(int)’ and ‘ProfileAction(int)’ contains switch statement where default case is missing. Usually you need to provide a default case. [1]

**Field ‘resultList’ dereferenced without null check**

Java file and line number: com.se.profilebuddy.adapter.PlacesAutoCompleteAdapter.java (line: 53 and 59)

Description: Field ‘resultList’ is never initialized within any constructor, and is therefore could be null after the object is constructed. Elsewhere, it is loaded and dereferenced without a null check. [1] .Could result in null pointer exception.

**Summary**

The coding standards, Data Structures, UI and Error handling are all up to standards. We have identified some issues which we feel needs to be dealt with. We have classified issues to Major (3 issues), Minor (7 issues) and Open Questions (1 issue). Please look into the issues and try to address them in the coming iterations.

**References**

1. FindBugs