(i) identify at least four existing code smells in your code. For each code smell, list all relevant git commits, files and line numbers (up to 0.5 marks)

1. Large Class
File: DataGenerator.java is a large class containing too much code. Line69-87
Git commit: May 7, 2022 at 8:47:26 PM GMT+10
Commit msg: DataStream module
2. Switch Statements
File: DataGenerator.java switch statements were used in dataStream() method. Line57-65
Git commit: May 7, 2022 at 8:47:26 PM GMT+10
Commit msg: DataStream module
3. Long Method
File: app/src/main/java/com/example/sportsdating/post/service/PostService.java Line37-83
Date: May 8, 2022 at 4:40:56 PM GMT+10
Commit msg: fix endIndex bug in PostService.java
Method “getPostByPage()” is too long, making it hard to understand.
4. Data Class
There are several data classes which have only setter and getter for fields insides.
File: app/src/main/java/com/example/sportsdating/user/model/User.java
Date: May 5, 2022 at 10:24:42 PM GMT+10
rename userProfilePic to userImg, rename getUserMgr to getInstance
File: app/src/main/java/com/example/sportsdating/post/model/Post.java
Date: April 21, 2022 at 9:21:19 AM GMT+10
"Post Pojo" commit, removed extra activity classes

(ii) correct at least two existing implementation issues identified in (i). For each correction, list all relevant git commits, files and line numbers (up to 0.5 marks)

Correction for Large Class in DataGenerator.java   Line69-87
Commit Date: May 10, 2022 at 5:40:01 PM GMT+10
Refactoring DataGenerator module
Commit Date: May 19, 2022 at 10:31:45 PM GMT+10
DataGenerator, fix issues
app/src/main/java/com/example/sportsdating/datagenerator/action/Action.java
app/src/main/java/com/example/sportsdating/datagenerator/action/PostAction.java
app/src/main/java/com/example/sportsdating/datagenerator/action/UserAction.java

Correction for Switch statements in DataGenerator.java Line57-65
Correction for Large Class in DataGenerator.java   Line69-87
Commit Date: May 10, 2022 at 5:40:01 PM GMT+10
Refactoring DataGenerator module
Commit Date: May 19, 2022 at 10:31:45 PM GMT+10
DataGenerator, fix issues
app/src/main/java/com/example/sportsdating/datagenerator/action/ActionFactory.java

(iii) provide a detailed description of the existing code smells identified in your code and explain how did you solve the implementation issues and why previous solutions were not suitable and why the new ones are. In case you are unable to correct the smells found in your code, explain why it was not possible. (up to 1 mark)

Solution for large class problem in DataGenerator module: Take out the codes doing different kinds of operations(create users, posts, etc.), and created an Interface Action.java, which defines “performAction()” method. Subclasses of this interface can implements this method based on their own behavior, such as UserAction.java, PostAction.java. This makes it easy to expand in the future when there is any other actions need to be added. For example, Activity creation can be added by adding an extra subclass ActivityAction.java. This does not require any change in client code making it decoupled and flexible to change.

Solution for switch statement was resolved by Action interface and applying Factory Design pattern. ActionFactory.java is created as the factory method generating the appropriate actions according to the parameters passed from client(DataGenerator.java). This enables client(DataStream) to decide which action to perform by passing the ActionType to createAction() method. Clients need very little change whenever there is any new actions are added.