CS-242/243

Project - #9

Smart Warning System

Code Review Document

Course Instructor -

Prof. Samit Bhattacharya

Team - #23

Jatin Goyal - 160101036
Namit Kumar - 160101046
Nitin Kedia - 160101048

Table of Contents

1. Introduction

The intent of a code review is to catch bugs/issues/defects before the offending code is deployed to a production environment and to transfer knowledge of implementation details to the rest of the team. Code review involves a slow code inspection phase to check errors.

1.1 Goals

The main reasons for performing code review is to :
• Finding bugs, since bug finding in code review are easier to find and fix than later in testing.
• Adherence to coding conventions
• Improving code quality
• Increasing efficiency, by finding trivial programming errors like data resource wastage or use of uninitialized variables.

1.2 Code Inspection

The aim of code inspection is to discover some common types of errors caused due to oversight and improper programming. In this phase, members outside of coding phase are selected and asked to perform the inspection, which is firstly checking of the coding standards. Then, the presence of certain kinds of errors like modifying a formal parameter while the routine calls a constant parameter. Considering the statistics some common programming errors which are to be checked are :
• Use of uninitialized variables
• Jumps into loops
• Improper storage allocation and deallocation
• Incompatible assignments
• Non terminating loops
• Array indices out of bounds
• Mismatches between actual and formal parameter in procedure calls
• Use of incorrect logical operators or incorrect precedence among Operators
• Improper modification of loop variables
• Comparison of equality of floating point variables are checked and reported. It is important to note that code inspection is a slow phase and no more than 400 lines of code are to be checked at a time. The report is then given to the author for appropriate changes to be done.

2. Code

Please find the following files in a zip file sent with this document:

1. LoginActivity.java
2. SignupActivity.java
3. ProfessorActivity.java
4. StudentActivity.java
5. ClassStatusActivity.java
6. ClassReviewActivity.java
7. NotificationActivity.java
8. NotificationAdapter.java
9. NotificationTouchListener.java
10. RecyclerViewAdapter.java
11. AdditionalUserData.java
12. Student.java
13. Notification.java
14. Index.js (this file must be named such else it cannot be deployed in Google Cloud Functions)

3. Code Inspection Team

The code inspection team comprises of the following members, all of whom are Sophomore undergraduates currently pursuing Bachelor of Technology at Indian Institute of Technology Guwahati, India in the Department of Computer Science & Engineering.

Team Members:

1. Vivek Raj (160101076)
2. Samyak Jain (160101059)
3. Abhinav Mishra (160101005)

All members of the team have past experience in developing android applications using Android Studio, Google Firebase and are proficient in Java.

4. Code Inspection Reports

4.1 Report By Vivek Raj

1. In SignupActivity, there is no distinction done between cases of wrong email format, small passwords, already existing users. All yield same “Authentication Failed” toast message.
2. All setOnClickListener( ) functions have formal parameters named as ‘v’ which is ambiguous.
3. At the beginning of onCreate( ) function in LoginActivity, for already logged in user StudentActivity is started without checking type of the user.
4. Many code lines are too long mainly becuase of many nested if statements, as in mJoinSessionButton,onSetClickListener.
5. The field requirements in the Signup UI should be mentioned.
6. In ProfessorActivity, readability is very bad and statements and comments are condensed together.
7. Functionality of onCreate( ) function is not mentioned in the comments.
8. The ProfessorActivity needs lot more comments than provide in the code.
9. It is difficult to read or find errors in recyclerViewAdater, NotificationTouchListener because very generic code has been used.
10. Session password should be stored in hashed format in the database for security reasons in createSessionButton.setOnClickListener( ).
11. “status” variable name is not explanatory and no comments have been provided to describe it in ProfessorActivity.
12. “Rand” variable name is not proper and can use a better name for better understandability in CountDownTimer “timer” in NotificationActivity. Also “millisUntilFinished” actually means “milliSecondsUntilFinished” is not commented.
13. Dialog in setOnClickListener() of createSessionButton and joinSessionButton have variable names “builder”, “viewInflater”, “which” which on their own do not give any relevant information about the purpose of the variable.
14. “timer” in Notification Activity onCreate() has a limit of 50 minutes after which random state will stop. So only the first 50 minutes of any session will have notifications which is not always the case.
15. In function generateAlert( ) an “alert” dictionary is created but another dictionary “payload” is created which is sent as system notification. This is redundant and can be avoided.
16. Indentation was done appropriately.
17. JUMP statements were not found anywhere.

4.2 Report By Samyak Jain

1. In LoginActivity, proper explanatory name should be used instead of ‘v’ in View parameter.
2. “Else” statements are not specified for every “if” condition. If there are no need for “else” statement, comment explaining the reason is also absent. Example - while redirecting to ProfessorActivity in LoginActivity only if user is professor check is used. Same is done in filtering blacklisted students in prepareReviewData( ) in ClassReviewActivity.
3. The code has been indented properly.
4. No JUMP (goto) statements were found in the code.
5. In SignupActivity and LoginActivity, variable name “task” doesn’t specify which task.
6. Logical error in onCreate() in LoginActivity, StudentActivity is launched without user type checking.
7. No comment what does “TAG” constant represent in each Activity.
8. In ProfessorActivity, use proper name for create session dialog, instead of generic names like “builder”. Same goes for its view which is named “viewInflater”.
9. ID’s of elements in layout XML files doesn’t indicate the type of element (Button/EditText).
10. There is inadequate comment to explain “cancel” function in create session dialog. Also in this and other dialogs have unused parameter “which” in builder.setNegativeButton( ) method.
11. In saveSessionDetails( ) in ProfessorActivity, many attributes of session are set individually and repeatedly. Instead, if a session class is made, then this can be done in fewer lines.
12. In ClassReviewActivity, “mDatabaseReference” actually refers to the joinedUsers in a session but the name indicates that it is a reference to whole database.
13. Some lines are too long in terms of characters which affects code readability.
14. No comments are given for when onCreate function triggers in each activity.
15. NotificationTouchListener has very less comments.
16. In NotificationActivity, countDownTimer “timer” starts from 3000 seconds i.e. 50 mins. If the Professor doesn’t finish his session in this time, this timer will stop along with Random State generation and notification generation.
17. In sendNotification( ) function the “alert” created is not send, instead it is used to create another dictionary to send the actual notification.
18. For classes AdditionalUserData, Notification and Student the ‘m’ prefix for variable naming previously followed has not been used.

4.3 Report by Abhinav Mishra

1. No JUMP statements were found in the codes of any file.
2. In LoginActivity, in function mLoginButton.setOnClickListener( ), parameter name is “v” . This doesn’t specify what View has been passed.
   This is also present in similar setOnClickListener( ) method of other buttons in of other activity like SignupActivity.
3. In LoginActivity, in onCreate( ) function, if user is already logged in then the code redirects to StudentActivity without checking. But the can be a professor, which will result in conflicts.
4. In SignupActivity, onComplete method createUserWithEmailAndPassword( ) has a parameter “task” which doesn’t specify what task. Also occurs in SignupActivity.
5. In StudentActivity, private class variable mSessionName is assigned but never used, making it redundant.
6. Variable names like mIsEngaged don’t make clear engaged in what upfront, similarly for mUid, mAdapter, mVauleEventListener etc.
7. Generic code is directly used for making Dialogs in ProfessorActivity, StudentActivity. They do not contain name of dialog, lack of comments etc.
8. In method builder.setNegativeButton( ) of all dialogs there is a unused int variable “which”. No comment was present explaining why.
9. In ProfessorActivity, overall the code is very condensed which affects readabililty. Also, logic for handling buttons are written in onCreate( ) method itself instead of separate functions.
10. In ClassReviewActivity, variable name “mDatabaseReference” is used to specifically refer to the joinedUsers in a session instead of the root of database.
11. Very sparse commenting is done in RecyclerViewAdapter, NotificationAdapter which makes it very difficult to understand.
12. In Notification Activity, the CountDownTimer for assigning random states starts from a finite session duration and goes to zero, never starting again. If the Session doesn’t end in that duration, alert generation will stop since states will not change.
13. Time to respond to notification, session duration limit are hardcoded instead of global constants in NotificationActivity.
14. In CountDownTimer type variable “timer”, “rand” object name is not appropriate.
15. In sendNotification( ) function, promises are not explained properly. Also the parameters passed “change” and “context” are ambiguous. Also two objects serving simlar pupose are created.
16. Indentation was done properly.

5. Conclusion

The members of the code review team submitted the reports during their final meeting with the development team. From these submitted reports, we get to know about a few logical errors that were encountered during the execution of the code inspection and were listed down.

Based on the three code inspection reports obtained, the errors identified are listed below:

1. In onCreate( ) function of LoginActivity, if the user is already logged in, then StudentActivity is launched immediately irrespective of user type (Professor/Student). This error occured since such a case occurs only when the a student clicks on a system notification.
2. In Notification Activity, the CountDownTimer “timer” starts from a hard-coded 50 minutes and for each 10 seconds passed assigns a random state to the student and saves it in database, if the session is active. But, it may be the case that the Professor doesn’t end session in 50 minutes then alert generation will stop for that student.
3. In LoginActivity, SignupActivity, ProfessorActivity, StudentActivity, setOnClickListener( ) is used the View type parameter “v” passed, it must be named appropriately. Similarly in AlertDialog definitions the “which” parameter builder.setNegativeButton( ) is not named appropriately.
4. ProfessorActivity is overall very congested and and not very reader-friendly.
5. RecyclerViewAdapter, NotificationAdapter and NotificationTouchListener need more comments.

These errors should be fixed to improve the reliability and readability of the code.