CS-223
CODE REVIEW DOCUMENT

for

Project 7
Classroom Visualisation App-1

Prepared by:
Group-12
Mitansh Jain - 160101042
Sujoy Ghosh - 160101073
Akul Agrawal - 160101085

April 23, 2018

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 and understandability
- 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 practice. In this phase, team members are selected and asked to perform the inspection, in which we check the coding standards. Then, we look for 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

All these 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 Coding Guidelines/Task Description

- Declaration of Global Constant Variables should be done in All Caps and separated by an underscore.
- For variable naming CamelCase is used.
- Local variables are named all small.
- Class names start with capital letters and separated by underscore.
- Function and function parameter naming is done in lowerCamelCase.
- **OTBF** indentation style is used for curly braces.
- Single tab indentation is used.
- Variables are to be named meaningfully.
- Headers are absolutely necessary in each module.
- Comments are required for each function.
- On an average four lines of code should have a single line of comment.
- At most 6 public Global variables can be used in a module.
- Do not use a coding style that is too clever or too difficult to understand.
- Avoid obscure side effects: The side effects of a function call include modification of parameters passed by reference, modification of global variables, and I/O operations. An obscure side effect is one that is not obvious from a casual examination of the code.
- Length of a function should be within 20 lines

- Header Format is as follows :-

  <header>

  Module Name

  Date of creation

  Author

  Modification history

  Synopsis

  Global variables

  Functions

  < /header>

3 Code Testing Team

3.1 Team Profile

The code testing team comprises of the following members, all of whom are Undergraduates currently pursuing Bachelor of Technology at Indian Institute of Technology Guwahati, India in the Department of Computer Science and Engineering. All of the members are currently in the sophomore year.

- Harshit Agrawal
- Harshit Gupta
- Harshit Sharma

All members of the team are proficient in Java and have past experience in developing android applications.

4 Code Inspection Reports

4.1 Code Report by Harshit Agrawal:

- The headers of each module had all the details that are required from a good header, like - Name of the module, Date on which the module was created, Author's name, Modification history, Synopsis of the module, Different functions supported, along with their input/output parameters.
- There is scope of improvement in commenting of the code in FdActivity.java as the functionality of some functions are not explained properly
- Proper indentation is followed while writing code.
- No JUMP (go to) statements were used in the modules.
- Error handling is minimal in insert method in DBHelper.java module.
- Parameter naming could be improved for insert methods in DBHelper.java and DBImgHelper.java
- Variable name is not CamelCase in EditStudentActivity.java line number 74 in code document.
- No Non-Terminating Loops were found in the module.
- OTBF indentation style has been properly followed.
- Proper validation and error handling is done in all Activity modules(There are 9 Activity modules and 3 Database module).
- Capitalizing the Global Variables was very useful to read the code
- Function names are according to proper conventions.
- Overall flow of code is understandable easily.

4.2 Code Report by Harshit Gupta:

- Nil uninitialized variables found in the module.
- Parameter naming could be improved in DBHelper.java and DBImgHelper.java
- The headers of each module had all the details that are required from a good header, like - Name of the module, Date on which the module was created, Author's name, Modification history, Synopsis of the module, Different functions supported, along with their input/output parameters.
- Error handling is minimal in insert method in DBImgHelper.java module.
- OTBF indentation has been followed in java modules.
- Proper indentation is followed while writing code.
- All the loops terminated according to their condition. No Non-Terminating Loops were found.
- There were none uninitialized variables found in any module.
- There is scope of improvement in commenting of the code in FdActivity.java as the functionality of some functions are not explained properly
- Proper validation and error handling is done in LoginActivity, SignupActivity, MainActivity and other 5 Activity modules.
- Overall flow of code is understandable easily.
- Function names are done according to CamelCase convention.
- Capitalizing the Global Variables was very useful to read the code specially in databases.

4.3 Code Report by Harshit Sharma:

- There were none uninitialized variables found in any module.
- There is scope of improvement in commenting of the code in FdActivity.java as the functionality of some functions are not explained properly.
- Error handling is minimal in insert method in DBHelper.java and DBImgHelper.java module.
- OTBF indentation has been followed in java modules.
- No JUMP (go to) statements were used in the modules.
- All the loops terminated according to their condition. No Non-Terminating Loops were found.
- Variable name is not CamelCase in EditStudentActivity.java and DeleteStudentActivity.java in line number 74 and 61 respectively in code document.
- When Camera Session starts, time is taken to train the images. During that time, it may predict incorrect answers
- The headers of each module had all the details that are required from a good header, like - Name of the module, Date on which the module was created, Author's name, Modification history, Synopsis of the module, Different functions supported, along with their input/output parameters.
- Functions are named according to proper conventions.
- Functions are exactly doing what the function name denotes.
- Some functions are long, i.e. have length greater than 30 lines. They could be decomposed (e.g. onCreate() of ViewStudentActivity.java)
- Variable name reflects its meaning and usage.

5 Conclusion

The following violations of conventions were found to be common among the observations made by the individual team members and are listed below for future correction

- Error handling is minimal in insert method in DBHelper.java and DBImgHelper.java module.
- Parameter naming could be improved for insert methods in DBHelper.java and DBImgHelper.java
- There is scope of improvement in commenting of the code in FdActivity.java as the functionality of some functions are not explained properly