SOFTWARE CODE REVIEW DOCUMENT

GROUP 22

ABHINAV MISHRA	:	160101005
SAHIB KHAN	:	160101057
SHIVAM KUMAR	:	160101066

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.

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

This document contains the complete software review of our app Classroom Monitoring. Code review for a model is carried out after the module is successfully compiled and the all the syntax errors have been eliminated. Code reviews are extremely cost-effective strategies for reduction in coding errors and to produce high quality code. Normally, two types of reviews are carried out on the code of a module. These two types of code review techniques are code inspection and code walk through. The code testing team in isolation tests different units and modules of the system. Different members of the code testing team have submitted their reports. Although, we are only performing the Code Inspection part wherein each member goes through code to discover some common types of errors caused due to oversight and improper programming. The main objectives of the inspection was to discover the algorithmic and logical errors in the code. The members noted down their findings, which were discussed in a meeting where the coder of the module were also present.

2. CODE TESTING TEAM

2.1 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 & Engineering. All of the members are currently in the sophomore year

1. Nitin Kedia (160101048)
2. Kapil Goyal (160101039)
3. Saurabh Bazari (160101061)

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

3. CODE INSPECTION REPORTS

1. Report by Nitin Kedia

After going through the code following observation were made by the reviewer

- The headers of each module had all the details that are required for a good header, - Name of the module, Date on which the module was created, Author's name, Modification history, Synopsis of the module, Different functions supported ,global variables is mentioned .
- Common convention throughout the code is to name class variables and function in camel case and local variables in snake case ,also class variables related to database starts with prefix 'm'.Class name is in camel case. Constants are written in all capital and separated by underscore.
- In MainActivity in userLogin() function and onCreate( ) function redirection logic is written twice. It can be refactored into a common redirect function.
- Some objects have similar name as class like View view ,Context context this may affect readability.
- The isPermission attribute of user is not self explanatory as it does not specify what is the permission for by name although the comments are helpful
- In SignupActivity "Radiogroup radiogroup" is not self explanatory , what it is used for.

- In SignupActivity local variable datasnapshot_user must refer only to user permission .User in general specifies a board term not just permission.
- Proper indentation is followed while writing code.
- Password for professor is hardcoded better way can be to make it a class constant and then use it.
- In ProfessorMainActivity the start session changes to end session when a session is active however the name of the button in code is buttonStartSession this could have been done better.
- All the code is not squeezed in onCreate function rather different functions are defined which results in making the code reusable.
- Commenting is substantial in most section of code ,however in some section there is scope of improvement.
- Session name and Session password has been used interchangeably for the input given by professor /student in the process of starting /joining session.
- Few lines of code containing too many character reducing readability
- In studentMainActivity in function applyTextsSessionName( ) roll number and student name are stored together in database separated by underscore ,another class could have been formed as this could have been made an object.
- In ScanActivity FPS and resolution in cameraSource.Builder is hardcoded instead use of global constant would have been ideal.
- In GridActivity variable names like noOfRowsInClass class, noOfColumnsInClass can be improved.
- In various activities else block after some if blocks are missing which may cause error.
- In GridActivity onCreate function there is redundant check on studentPositionList size >0 .If it is zero inner loop will automatically not execute.
- In ViewPastRecordActivity function selectCourse() and selectSession( ) are just wrapper function for calling respective dialogue hence redundant.
- In ViewPastRecordActivity set Adapter() function mAdapter changes after each iteration of loop however it will be sufficient to change it only once after loop termination

2. Report by Kapil Goyal

After going through the code following observation were made by the reviewer

- Common convention throughout the code is to name class variables and function in camel case and local variables in snake case ,also class variables related to database starts with prefix ‘m’.Also Class name are in CamelCase and Constants are written in ALL CAPITAL separated with underscore.
- Headers for each module have been made properly and enhance readability of the code.
- No index of array is out of bound.
- In some lines the character limit exceed 80 character limit, it decrease readability.Example is in StudentMainActivity function applyTextsSessionName().
- No JUMP (go to) statements were used in the modules.
- There are some activities where else block is not provided after some if statements, this may result in logical error.For example in MainActivity function userLogin() .
- There were none uninitialized variables found in any module.
- Commenting for if block is irregular.At some places details for if statement are provided in a comment above the if and in some places they are provided below the if statement.As in SignupActivity function onCreate() line 96 ,97 ,99, 100
- Commenting of code should be more elaborative.
- Variable naming is done nicely .Most variable names are intuitive and self explanatory .
- Function length exceed 20 lines resulting in less readability .
- The code is divided in functions which perform individual task and enhances re-usability.
- In Student Main Activity comments not provided for Bundle .
- In a class named “dialogue” Camel Case convention is not followed
- In ScanActivity FPS and resolution in cameraSource.Builder is hardcoded
- In StudentMainActivity function onActivityResult() there are 4 nested if’s which is not recommended
- Exception Handling is done in code ,though could be increased throughout the code
- In StudentMainActivity function showDialogForJoinSession() a variable is named as items .Although comments are provided but the name could have been self-explanatory.
- The password for professor in SignupActivity in function show Password Prompt()” is hardcoded.
- All the loops terminate according to respective conditions .
- The code follows Ratliff indentation throughout the project which enhances the readability
- In dialog classes the name listener of a variable is very generic.

3. Report by Saurabh Bazari

After going through the code following observation were made by the reviewer

- Common convention as found in the code is to name class name,class variables and function in camel case and local variables in snake case ,also class variables related to database starts with prefix ‘m’. Constants are written in all capital and separated by underscore.
- Header as observed in each file contained information like name of the module, date on which the module was created, author’s name, modification history, synopsis of the module, different functions supported and global variables used.
- Throughout the code multiple times object is defined with the similar name as their class/data type for example View view,Context context etc. which may lead to confusion and reduce understandability.
- In the MainActivity and the SignupActivity checking for already logged in user is done so if a user shuts down the app without logging out the next time the app is opened the user is automatically logged in with the most recent user id.
- In MainActivity in userLogin() function and onCreate( ) uses same set of code lines creating a function and reusing it would have been ideal.
- In SignupActivity in function “showPasswordPrompt()” the password for registering a new professor profile user is hardcoded ideally a constant class variable should have defined.
- In StudentMainActivity the variable “QRvalue” is a class variable but is not in camel case.
- In StudentMainActivity in function “showDialogForJoinSession()” a String array “items” is declared saves the course’s name this is evident from the comments but is not self explanatory by the name.
- Comments are abundant in most of the sections of code with proper explanation but there is scope of improvement in some sections of code like ScanActivity.
- The code follows Ratliff indentation throughout the project making the code easier to understand.
- No JUMP (go to) statements were used in the modules.
- Few code lines are very long which reduces readability.
- No uninitialized variables were found and loops were found to be terminating.
- In ProfessorMainActivity the button associated with start session performs both task of starting and ending a session however this is not intuitive from the variable’s name although it is explained in comments.
- In ScanActivity there is hardcoding for camera details which ideally should have been done by constant variables.
- There exists some if blocks in code whose else blocks are not present.

- Emphasis is given on making code functional rather than having all the logic inside one function multiple functions are created for different logic.
- In GridActivity onCreate function there is redundant check on studentPositionList size >0 in case of being zero inner loop will automatically not execute.
- In ViewPastRecordActivity function selectCourse() and selectSession( ) are just wrapper function for calling dialogue hence could have been eliminated.

4. 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.

These errors will affect the execution of application . The key errors found from the code review are as follows :-

1. Certain sections of code were identified where more commenting could be done to increase the understandability of code.
2. Some values are hardcoded in the code which should be avoided .
3. selectCourse() and selectSession( ) are just wrapper function for calling dialogue hence could have been eliminated.
4. In ViewPastRecordActivity set Adapter() function, mAdapter changes in each iteration but it could be written after the for loop.
5. Some object names were same as class name this affects readability.
6. Function were not in 20 lines limit and some line exceeded maximum 80 character limit which decrease readability of code.
7. In studentMainActivity in function applyTextsSessionName( ) roll number and student name are stored together in database separated by underscore instead of seperate values.
8. In GridActivity onCreate function there is redundant check on studentPositionList size >0 in case of being zero inner loop will automatically not execute hence it can be removed.

Fixing these errors can improve the application, Classroom Monitoring App.