

Indian Institute of Technology Guwahati



CS243 : Software Engineering Lab

Under the guidance of : Prof. Samit
Bhattacharya

Project 1 : An app to detect student activity and alert generation for the instructor

Code Review Report

Group 7 :

- Mukul Verma , 150101038
- Piyush Jain , 150101046
- Shubhanshu Verma , 150101073

Code Inspection done by -

Name : Hrushikesh Turkhade

Department : Mathematics and Computing

Background : Experienced in Coding and has done few good projects.
Good knowledge of Programming practices.

Date : May 6, 2017

Contents

1	Introduction	4
1.1	Purpose	4
1.2	References	4
1.3	Coding Conventions	4
1.4	Defects Checked	4
2	Code Inspection Process	5
2.1	Description	5
2.2	Impression of the process	5
3	Defects	7
4	Conclusion	7

1. Introduction

1.1 Purpose

This code review document is a report of the team inspection of code. This review is intended to find the inconsistencies in the coding style, or any kind of logical error in code and also gives an idea about how much it is deviated from the ideal coding practices and its justification. This will help in guiding in rectification of code design related discrepancies.

1.2 References

For references, Software Requirement Specification and Design Document of the project have been referred.

1.3 Coding Conventions

Functional approach of coding style has been used. The number of source lines per function should be around 10.

1.4 Defects Checked

- Coding Conventions
- Logic Errors
- Commenting

2. Code Inspection Process

2.1 Description

Following are the steps followed for code inspection :

- Read the whole code while focusing on each module independently.
- Length of modules were checked.
- Infinite loops and other logical errors were traced.
- Commenting at all appropriate places was checked.
- Actionable feedback.
- Follow up with discussion.

2.2 Impression of the process

This code review helped to eliminate some of the basic and subtle errors which could have been eliminated just by going through the code. The list of modular, functional and logical errors have been designed in the form of detailed checklist as follows :

S.No.	Check	Remarks
Reviewing Comments and Coding Conventions		
1	Does the code respect the project coding conventions?	Yes
2	Are the variable declarations properly commented?	Yes
3	Are the functions really doing what the documentation says?	Yes
4	Are the functions parameters used for input and output identified and documented in the function header?	No
5	Are comments used to show missing or yet to be implemented functionality?	Yes
6	Is the modification log upto date?	No

Reviewing Error Handling		
1	Are errors properly handled each time the function returns?	No
2	Are error messages conveying exactly what the error has occurred?	Yes
Reviewing Control Structures		
1	Are loop ending conditions accurate?	Yes
2	Does the code contain any loops which may become unintended infinite loops?	Yes
3	Are 'if' statements used where 'if-else' can be used?	Yes
4	Is assignment('=') operator used within an 'if' condition instead of equality('==') operator?	Yes
Reviewing Functions		
1	Are variables initialized properly before use?	Yes
2	Are the functions parameters explicitly validated in code?	Yes
3	Are arrays getting checked for out of bound indices?	No
4	Did the code re-write a function which is already available in an existing library?	No
Review Code Performance Aspects		
1	Are large amounts of data being passed to functions by value?	No
2	Are there lines of code which are same but occur in many places?	Yes

3. Defects

Defects		
S.No.	Name of the module	Description of defect
1	updateOrientationAngles	Error in passing parameter
2	mild_flaw (global variable)	Variable not initialized
3	counter_reset	Assignment operator in place of equality operator

4. Conclusion

Apart from a few errors mentioned above, the whole coding style is good and is well-commented.