CODING DOCUMENT

Prepared for Software Engineering Project - "Attento" version 1.3

Prepared by:- Group 11

Gourika Bang	150101025
Saurabh	150101062
Soumik Roy	150101074

6 May, 2017

Supervised by:-

Prof. Samit Bhattacharya

CONTENTS

REVISION HISTORY

S.No.	Date	Reason for modification	Version
1	25/04/2017	Original Document	1.0
2	03/05/2017	Update in original document	1.1
3	05/05/2017	Remarks from Prof. Bhattacharya	1.2
4	06/05/2017	Final Edits	1.3

2. CODE REVIEW - CODE INSPECTION

2.1 INTRODUCTION

An inspection team comprising of 4 members is chosen to bring the subtle defects to light caused due to oversight and improper coding standard. Typical defects are errors of documentation. The logical and algorithmic errors are not taken into account.

The inspection team is provided with background information for the inspection.

2.2 TEAM DETAILS

Our Inspection team comprises of 4 members who performed the code inspection to detect the errors in coding standard and tasks assigned to them are given in the section below.

All the team members have a good knowledge of Android Studio.

2.3 TASK DESCRIPTION

- Team member 1 was responsible for finding defects in the work during preparation and during the inspection meeting.
- Team member 2 lead the inspection team and was responsible for ensuring that a good inspection is achieved
- Team member 3 was responsible for guiding the inspection team through the product during the inspection meeting by reading or paraphrasing the product
- Team member 4 was responsible for accurately recording information during the meeting about each defect found on the inspection defect list.

2.4 REPORTS OF INDIVIDUAL TEAM MEMBERS

2.4.1 Report of Team Member 1

S. No.	Checklist	Review
1	Are array indices out of bounds?	No, all the array indices are within the range
2	Are the names of macros all uppercase?	Only few were lowercase in the login module
3	Are variables declared as pointers used as pointers(not integers)?	Yes, the variables declared as pointers were used as pointers
4	Do all but the most obvious declarations have comments?	The headers of the module were proper but there were few sections missing comments.
5	Are pointers immediately set to NULL (or 0) following the deallocation of memory?	Yes
6	Are unary operators adjacent to their operands?	Yes

S. No.	Checklist	Review
7	Are "if...else" trees and "switch" used clearly?	Yes
8	Are "ifs" indented?	The indentation is improper
9	Does each unit have a single purpose?	Yes each unit serves a unique purpose
10	Does each unit have one exit point?	Some functions had multiple return statements.
11	Are the error conventions consistent?	The error convention of false and true is consistent
12	Are "goto" statements used?	No

2.4.2 Report of Team Member 2

S. No.	Checklist	Review
1	Does code that passes a pointer to another function firstcheck for a valid (non-zero) pointer?	The non-null condition is checked
2	Is there code that should be in separate functions?	The onSensor function is very lengthy should be splitted.
3	Is each name used for only a single purpose?	Yes the variable names are indicative.
4	Are pointers initialized?	Yes
5	Are "IMPORT" files used according to project standards?	Some extra libraries are also imported.
6	Are all data local in scope?	No
7	Do assignment and conditional operators always have space around them?	No such convention is followed throughout the document.
8	Are parentheses used properly for precedence?	Yes
9	Is the unit header informative and complete?	The unit header is specified

2.4.3 Report of Team Member 3

S.No.	Checklist	Review
1	Are there any preprocessors?	No
2	Are there any uninitialised variables?	The variables "email" and "pass" are uninitialised
3	Are there sufficient comments to understand the code?	Yes the header file is informative and the standard convention is followed. Sufficient comments in other sections.
4	Are the functions of arrays and variables described?	Yes for all the used array functions library is imported
5	Are loops indented and visually separated from the surrounding code?	Indentation can be improved

2.4.4 Report of Team Member 4

S.No.	Checklist	Review
1	Are all constant names uppercase?	No, all were named Camel case
2	Are parentheses used properly for precedence?	Yes
3	Is the block of code that follows an "if" statementsurrounded by { } brackets?	Yes
4	Are comment lines used to group logically-related statements?	There are common comments for many sections of the code
5	Are changes made to a unit after its releases noted inthe development history section of the header?	Yes, date of modification history is mentioned in all the modules
6	Are the loops terminating?	Yes, there is no infinite/unending loop
7	Mismatches between actual and formal parameter in procedure calls?	No mismatch
8	Improper modification of loop variables.	The loop variables are incremented/decremented according to the loop syntax
9	Is there any function with more than 10 source lines?	Yes, "accelerometer" and "OnSensor" function are too lengthy
10	Comparison of equally of floating point variables.	Yes

2.5 CONCLUSION BASED ON COMPILATION OF REPORTS

2.5.1 Negative Aspects

- There were two functions that were lengthy and exceeded 10 source lines.
- The indentation was not proper.
- There were some uninitialised variables.

2.5.2 Positive Aspects

- The unit headers were properly specified with the following details
  1. Date on which the module was created.
  2. Author's name.
  3. Modification history.
  4. Synopsis of the module.
  5. Different functions supported, along with their input/output parameters.
  6. Global variables accessed/modified by the module
- Pointer errors are handled carefully.
- The loop conditions and syntax are proper with terminating correctly.
- Error handling mechanism is consistent.
- No "goto" or "jump" statements are used.