



CS352 – Software Engineering II

Phase 1 Template

2017

Project Team

ID	Name	Email	Mobile
20140076	Amira Sherif Alsayed	amiiraelsayed@gmail.com	01271309802
20140333	Dina Essam Eldin	dododandoqa@gmail.com	01114180062
20140315	Youmna Essam Fawzy	yomna_essam80@yahoo.com	01022758669
20140294	Nouran Mohssen el-Halawanny	nouran_elhalawany@hotmail.com	01141515043

Staff: Dr Amr Kamel
Dr Khadiga Mohamed

a.kamel@fci-cu.edu.eg
kelbedweihiy@fci-cu.edu.eg

TAs: Eng Mohamed Samir
Eng Ragia Mohamed

m.samir@fci-cu.edu.eg
r.mohamed@fci-cu.edu.eg



Phase 1 document

Contents

Review Check List	3
Design and Code Checklist	3
Testing	5
Git repository link	11



Phase 1 document

Review Check List

Design and Code Checklist

Design Principles

- 1- Does the design follow OOP rules? ✓ 70%

Related Issues:

In class "Registration", Attributes are assigned with values. [Lines 8, 9, 10, 11, 12, 13]

In class "Admin", Attributes are assigned with values. [Lines 2, 3]

In class "Data Manager", Attributes are assigned with values. [Lines 229, 230, 231]

In class "Account", Attributes are assigned with values. [Lines 5, 6, 7, 8, 9, 10]

- 2- Is the design simple and easy to modify? ✓ 75%

Related Issues:

In class "Registration", Function SIGNUP takes another function CHECK_INFO which also takes another function LOAD_DATA.

Coding Standards

- 3- Is the code understandable and readable? ✓ 80%

Related Issues:

In class "Registration", Non understandable casting. [Line 104]

In class "Data Manager", Non understandable casting. [Line 178]

- 4- Do variable have good names? ✓ 90%

Related Issues:

In class "MCQ", meaningless variables. [Lines 28, 29]

- 5- Are spaces fairly distributed? ✓ 85%

Related Issues:

No spaces vertically and horizontally.

No spaces in class "Data Manager" [Lines 229, 230, 231], class "Add Game" [Line 25], class "MCQ" [Line 33], class "Registration" [Lines 78, 79, ..., 123, 124], class "Main" [Lines 26, 27, ..., 59, 60].



Phase 1 document

6- Does the code contain any unused methods or variables? ✓ 35%

Related Issues:

In class “Data Manager”, Useless variable’s assignment. [Lines 14, 15, 16]

In class “Data Manager”, Useless variables. [Line 64, Line 135]

Many unused imports.

Comments

7- Is the code commented enough? ✗ 10%

8- Is every class and method commented? ✗ 10%

9- Do comments follow Javadoc style? ✗ 0%

Related Issues:

The only written comments are in Franco Arab format.

10- Is Javadoc generated for all the code? ✗ 0%

11- Are there useless / wrong comments? ✓ 80%

Related Issues:

In class “Categories”, Useless comments. [Line 10, 11... 16, 17]

In class “Data Manager”, Useless comments. [Line 206, 207]

In class “TF”, Useless comments. [Line 15]

In class “Main”, Useless comments. [Line 2]

Code Structure

12- Does the code follow the design precisely? ✓ 50%

Related Issues:

The code isn’t completely and correctly implemented. There are functions that does not do its functionality.

In class “Data Manager”, function UPDATE_DATA.

In class “Add Game”, Unreachable code. [Lines 25, 26... 35, 36]

In class “Registration”, Magic numbers exists. [Line 94]

In class “Main”, Magic numbers exists. [Line 30, 53]



Phase 1 document

13- Are there very long classes or methods? ✓ 20%

Related Issues:

In class "Registration", function SIGNUP.

In class "Data Manager", function UPDATE_DATA.

Error Handling

14- Does the code handle errors and exceptions? ✓ 70%

Related Issues:

In class "Data Manager", it's not handled if the user enters anything other than the pre-defined types of games.

In class "Registration", it's not handled if the user enters anything other than the pre-defined gender.

In class "Main", the program just terminate and doesn't indicate whether the user has signed up successfully or not.

Logic

15- Do loops have correct conditions and bounds? ✓ 90%

Related Issues:

In class "Add Game", there is a case block. [Line 25, 26]

In class "Main", the MCQ game is empty.

In class "MCQ", the PLAYMCQ function's type is not logical.

In class "Data Manager", the UPDATE_DATA function's type is not logical.

In class "Registration", the SAVE_DATA function's type is not logical.

Overall

16- Are the design and code of good quality? ✓ 85% Good

Testing



Phase 1 document

1.1TFTest class

Number	Testing function	Description	Result
1.1	PlayTF(Game g)	Testing function for PlayTF function. This test case test the normal scenario. Assumption: return true with correct answers.	Passed
1.2	PlayTF(Game g)	Testing function for PlayTF function. This test case test the exception scenario. Assumption: return false with wrong answers.	Failed
1.3	PlayTF(Game g)	Testing function for PlayTF function. This test case test the exception scenario. Assumption: Write wrong type of game 'g' .	Passed



Phase 1 document

2.1MCQTest class

2.1	PlayMCQ(Game g)	<p>Testing function for PlayMCQ function. This test case test the normal scenario.</p> <p>Assumption: return true with correct answers.</p>	Passed
2.2	PlayMCQ(Game g)	<p>Testing function for PlayMCQ function. This test case test the exception scenario.</p> <p>Assumption: return false with wrong answers.</p>	Failed
2.3	PlayMCQ(Game g)	<p>Testing function for PlayMCQ function. This test case test the exception scenario.</p> <p>Assumption: Write wrong or empty file path.</p>	Passed



Phase 1 document

3.1 RegistrationTest class

3.1	Int Signin()	Testing function for Signin() function. This test case test normal scenario. Assumption: expected to return 0 “refer to Student type”	Passed
3.2	Int Signin()	Testing function for Signin() function. This test case test normal scenario. Assumption: expected to return 1 “refer to Teacher type”	Passed



Phase 1 document

3.3	Int Signin()	<p>Testing function for Signin() function. This test case test exception scenario.</p> <p>Assumption: there is the same info of type teacher and student, so expected to return "0"</p>	Failed
4.1	Account Check_Info()	<p>Testing function for Check_info() function. This test case test the normal scenario.</p> <p>Assumption: Expected to return the same account of Student and Teacher.</p>	Passed



Phase 1 document

4.2	Account Check_Info()	<p>Testing function for Check_info() function. This test case test the exception scenario.</p> <p>Assumption: Expected to failed test function with wrong formulas of data as: -ve age, wrong gender.</p>	Passed
4.3	Account Check_Info()	<p>Testing function for Check_info() function. This test case test the exception scenario.</p> <p>Assumption: Expected to Enter again Email with Wrong Email</p>	Failed



Phase 1 document

5.1Data_managerTest class

5.1	load_data()	<p>Testing function for load_data() function. This test case test the normal scenario.</p> <p>Assumption: Expected to return true with correct data in file.</p>	Passed
5.2	load_data()	<p>Testing function for load_data() function. This test case test the exception scenario.</p> <p>Assumption: Expected to return false with incorrect type of game in file.</p>	Failed

Git repository link

<https://github.com/AmiraAlsayed/Software-Engineering-I>