Code Inspection Report

'Bom Dia Academia' Software Development Project

BSc in [LEI]
Academic Year 2018/2019 - 1º Semester
Software Engineering I

Group Id 83
77773, Sérgio Vaz Ribeiro, LEI-PL
78549, José Filipe Santos, LEI
78641, Elsa Teixeira, LEI

ISCTE-IUL, Instituto Universitário de Lisboa
1649-026 Lisbon
Portugal

December 2018

Table of Contents

Introduction

The software consists of an aggregator of academic information from 3 channels: Facebook, Twitter and ISCTE e-mail, and is intended to be used by ISCTE students.

The following class and package diagrams, summarize the code structure of the application*.

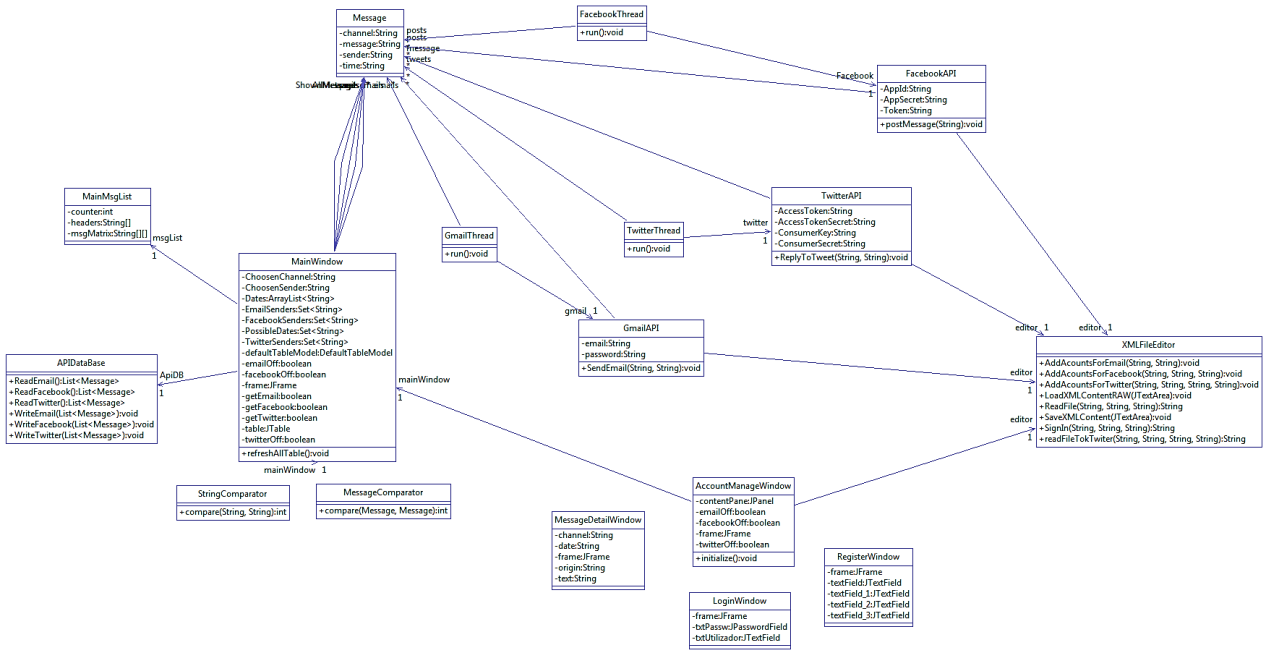

Figure 1: Class Diagram of the project.

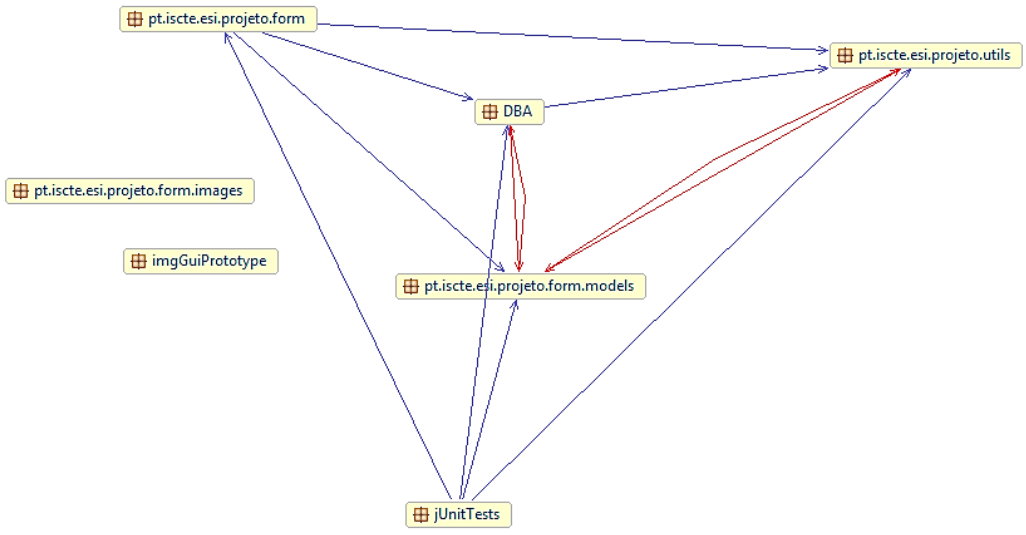

Figure 2: Package Diagram of the project.

*For a detailed view, please see these diagrams in the Diagrams folder at the root of the project.

Code inspection

Meeting date:	*07/12/2018*
Inspector:	*Elsa Teixeira*
Component name	
pt.iscte.esi.projeto.form/…	
/ AccountManageWindow	
Component was compiled:	√
Component was executed:	√
Component was tested without errors:	√
Testing coverage achieved:	*70,6%*
/LoginWindow	
Component was compiled:	√
Component was executed:	√
Component was tested without errors:	√
Testing coverage achieved:	*69,2%*
/Main:	
Component was compiled:	√
Component was executed:	√
Component was tested without errors:	√
Testing coverage achieved:	*61,1%*
/MainWindow:	
Component was compiled:	√
Component was executed:	√
Component was tested without errors:	X (reason: *API Facebook*)
Testing coverage achieved:	*43,5%*
/MessageDetailWindow:	
Component was compiled:	√
Component was executed:	√
Component was tested without errors:	√
Testing coverage achieved:	*73,9%*
/RegisterWindow:	
Component was compiled:	√
Component was executed:	√
Component was tested without errors:	√
Testing coverage achieved:	*78,6%*
pt.iscte.esi.projeto.form.models/…	
/FacebookAPI:	
Component was compiled:	√
Component was executed:	√
Component was tested without errors:	X
Testing coverage achieved:	*18,2%*

/FacebookThread:	
Component was compiled:	*√*
Component was executed:	*√*
Component was tested without errors:	*√*
Testing coverage achieved:	*78,6%*
/GmailAPI:	
Component was compiled:	*√*
Component was executed:	*√*
Component was tested without errors:	*√*
Testing coverage achieved:	*71,9%*
/GmailThread:	
Component was compiled:	*√*
Component was executed:	*√*
Component was tested without errors:	*√*
Testing coverage achieved:	*89,3%*
/Message:	
Component was compiled:	*√*
Component was executed:	*√*
Component was tested without errors:	*√*
Testing coverage achieved:	*100%*
/MessageComparator:	
Component was compiled:	*√*
Component was executed:	*√*
Component was tested without errors:	*√*
Testing coverage achieved:	*0%*
/StringComparator:	
Component was compiled:	*√*
Component was executed:	*√*
Component was tested without errors:	*√*
Testing coverage achieved:	*66,1%*
/TwitterAPI:	
Component was compiled:	*√*
Component was executed:	*√*
Component was tested without errors:	*√*
Testing coverage achieved:	*22,5%*
/TwitterThread:	
Component was compiled:	*√*
Component was executed:	*√*
Component was tested without errors:	*√*
Testing coverage achieved:	*78,6%*
DBA/…	
/APIDataBase:	
Component was compiled:	*√*
Component was executed:	*√*

Component was tested without errors:	*√*
Testing coverage achieved:	*100%*
/DBAWindow:	
Component was compiled:	*√*
Component was executed:	*√*
Component was tested without errors:	*√*
Testing coverage achieved:	*79,4%*
pt.iscte.esi.projeto.form.utils/…	
/MainMsgList:	
Component was compiled:	*√*
Component was executed:	*√*
Component was tested without errors:	*√*
Testing coverage achieved:	*99,2%*
/XMLFileEditor:	
Component was compiled:	*√*
Component was executed:	*√*
Component was tested without errors:	*√*
Testing coverage achieved:	*94,5%*
/XMLTester:	
Component was compiled:	*√*
Component was executed:	*√*
Component was tested without errors:	*√*
Testing coverage achieved:	*0%*

Code inspection checklist

1. Variable, Attribute, and Constant Declaration Defects (VC)

|X | Are descriptive variable and constant names used in accord with naming conventions? *No.*
|√ | Are there variables or attributes with confusingly similar names? No.
|√ | Is every variable and attribute correctly typed? Yes.
|√ | Is every variable and attribute properly initialized? Yes.
|√ | Could any non-local variables be made local? No.
|√ | Are all for-loop control variables declared in the loop header? Yes.
|√ | Are there literal constants that should be named constants? No.
|√ | Are there variables or attributes that should be constants? No.
|√ | Are there attributes that should be local variables? No.
|X | Do all attributes have appropriate access modifiers (private, protect,…)? No.

2. Method Definition Defects (FD)

|√ | Are descriptive method names used in accord with naming conventions? Yes.
|√ | Is every method parameter value checked before being used? Yes.
|√ | For every method: Does it return the correct value at every method return point? Yes.
|√ | Do all methods have appropriate access modifiers (private, protected, public)? Yes.

|X| Are there static methods that should be non-static or vice-versa? *Yes.*

3. Layout and Packaging Defects (LP)

|√| Is a standard indentation and layout format used consistently? Yes.
|X| For each method: Is it no more than about 60 lines long? No.
|√| For each compile module: Is no more than about 600 lines long? No.

4. Modularity Defects (MO)

|√| Is there a low level of coupling between modules (methods and classes)? No
|X| Is there a high level of cohesion within each module (methods or class)? *More or less. Some classes have no code connections with other classes, as we can see on UML diagram. However, is not relevant to the functionality of the application.*
|√| Is there repetitive code that could be replaced by a call to a method that provides the behavior of the repetitive code? No.
|√| Are the Java class libraries used where and when appropriate? Yes.

Found defects

Found defect Id	Package, Class, Method, Line	Defect category	Description
1	line 22 – TwitterAPI class	Variable, Attribute, Constant Declaration Defects (VC)	Constant name without case letter.
2	line 22 – TwitterAPI class		Constants should be public static final.
3	WriteFacebook() and other methods in APIDataBase class (DBA package).	Method Definition Defects	Some method could be static.
4	getMatrixElements() method on MainWindow class	Layout and Packaging Defects	More than 60 lines per method. In this method there are 63 lines, for example.
5	Package: pt.iscte.esi.projeto.form/	Modularity Defects	*Some classes have no code connections with other classes, as we can see on UML diagram. However, is not relevant to the functionality of the application.*

Corrective measures

We can suggest to the client to deliver an improved version later, with the corrections made. However, the fixes are not essential to the functionality of the application.

Conclusions of the inspection processs

The reception of information through facebook is compromised. However, the reason for this failure is not our responsibility.
Besides that, only minor errors were detected that did not compromise the functionality of the code.