**GRIFFITH COLLEGE DUBLIN**

**COMPUTING ASSIGNMENT TITLE SHEET**

| | |
|---|---|
| **Course:** | **BSC** |
| **Stage/Year:** | **1** |
| **Module:** | Mobile Development |
| **Semester:** | Semester 2 |
| **Assignment Number:** | 1 |
| **Date of Title Issue:** | 24th. February 2020 |
| **Assignment Deadline:** | 15th. March 2020 |
| **Assignment Submission:** | Upload the completed assignment to Moodle. Do not email. |
| **Assignment Weighting:** | 12.5/50 |

**Assignment Title**

| | |
|---|---|
| **Assignment title / brief** | |
| | Create the ContactManager application as outlined on pages 2, 3, 4 of this document. |

**Learning Outcomes**

| **Programme and related module learning outcomes that this assignment is assessing.** |
|---|
| 1, 2, 6 |

**Assessment Criteria**

| **Assessment criteria applied to this assignment:** |
|---|
| Adherence to good academic standards with regard to use of the work of others. Transparency of working, i.e. show all working. Demonstrate knowledge and understanding of the topics addressed in the questions.  Presentation (including use of the required software tools and adherence to the required file type) and professionalism of the work. |

**BSC-MD Assignment 1: ContactManager**

**1 Assignment Information**
Course: BSC (EPITA)
Stage:
Module: DM Semester: 2
Title: ContactManager
Description: Create an application to manage contact details including name, telephone and email address.
Assignment 1 Date of Issue: 24. February 2020
Assignment Deadline: Sun 15. March 2020
Assignment Submission: Upload to Moodle
Assignment Weighting: This is the first of three assignments and is weighted at 12.5% out of a total of 50% for all three assignments.

**2 Introduction**
In this assignment you are required to create a contact manager application that supports the storing of personal contact details including name, telephone number and email address. Users of the application should be able to add a new contact as well as edit and delete existing contacts. Your application will be required to use multiple activities. All activities should be implemented with standard UI elements that are available with the standard SDK. Do not use any additional libraries.

Activities should be linked through use of the intents system. You will be given basic information about the structure but beyond that you will have to design and implement the application and UI. You will be required to present and explain UI design decisions in the accompanying documentation. You will also be tasked with implementing one additional feature of your choosing.

Give your project the name ContactManager. Use the "Basic Activity" template to create your project.

**3 Assignment Submission**

Create a .zip archive of the finished project. Include the required documentation as a .pdf file entitled 'documents.pdf' in the root directory of the project. Ensure that all layout and code components are contained in the archive. Missing components will result in reduced marks. Upload the archive containing code and the documentation file using the link provided on moodle. Note that archive formats other than .zip will not be marked.

**Please note the following:**

• Use only the .pdf format for your documenation. If you do not provide any documentation, your code will not be marked. Although for illustration purposes, code snippets of 2 to 3 lines are useful, copying and pasting large sections of code does not constitute documentation.

• Code that fails to compile will incur a 30% penalty before grading. It should be possible to open your project and compile and run it without having to correct syntax errors.

• The use of libraries outside the SDK will incur a 50% penalty before grading. The standard SDK provides everything that is needed for the assignment. No additional libraries will be downloaded for the purpose of correction or marking.

• The standard late penalties apply

**Very Important:** Take note of the milestones listed in Section 4. These are meant to be completed in order. If you skip a milestone or trigger one of the failing conditions, the following milestones will not be considered for marking. For example if there are six milestones and you fail the third one, then the fourth, fifth, and sixth milestones will not be marked.

Reduced marks will be awarded for the presence of errors anywhere in the code. Validate input to ensure that it is sensible. Also note that the percentage listed after the milestone is the maximum mark you can obtain if you complete that many milestones without error.

**4 Coding Milestones (70%)**

1. Generate layouts and activities for displaying the list of contacts, adding a contact, editing a contact and deleting a contact entry. (10%)

2. Your application should contain multiple activities. Provide for navigation from one activity to the next. For your list of notes you should also add events to edit and delete notes. (20%)

3. Create a handler to an SQLite database for storing contacts. (30%)

4. Complete the add contact functionality. Users should be able to save the new contact or cancel it. (40%) You will fail this milestone if you do not use SQLite to store the notes.

5. Complete the edit functionality. Users should be able to save the edited contact or cancel the edit, leaving the contact details unchangedt. Marks will not be awarded for  this milestone if you do not use SQLite to store the notes. (60%)

6. Implement one additional feature. For the purpose of marking, the relevance and usefulness of the feature as well as the challenge involved in implementing it will be taken into account. Use your accompanying documentation to motivate and provide details of the feature. Include diagrams and screenshots where appropriate. (70%)


**5 Documentation Milestones (30%)**

1. Document why you designed the UI the way you did. Include adequate graphics such as wireframe diagrams and screenshots. Detail the key choices taken in application navigation, widget layout and position and how they support user interaction. Don't include instructions on how to operate the App or the UI. (15%)

2. Give a high level description of all methods in your Java code including the data- structures used. (30%)