MALMÖ HÖGSKOLA

Malmö University

Programming Using C#, Basic Course

Project Assignment

Mandatory (for VG or A, B)

Farid Naisan
University Lecturer
Department of Computer Science

Table of Contents

Project Assignment

A project work is the final obligatory assignment, and as a matter of fact it is the final examination moment of this course. In this assignment, you freely determine the idea and the contents of your software project by yourselves. You may have probably wanted to develop a program for your own use or for your family, a friend or your colleagues at your office. Now, you have the opportunity.

1. Objectives

The main goal of this work is to give you an opportunity to put t the knowledge you have gained and the skills you have developed throughout the training modules into practice, and produce a fully functional software that solves a real world's problem. Another important issue is to get experience in teamwork which is a modern time's real-life scenario. However, if you wish to do the project work on your own, it is ok.

2. Description of the Project

2.1 The assignment is recommended to be carried out in groups of 2 to 3 persons, but if you prefer to work by yourself, you may do so. The group members do not necessarily need to have any physical meetings and therefore may be located at different places of the world, communicating via mail, phone or the platform.

2.2 Groups are built by students themselves. Use the platform (Project Module) to build a new group or join another group.

2.3 The assignment consists of two parts, a **Project Proposal** (project idea) and the **Project Implementation** (VS project). As soon as a group has finalized a project idea, a short project proposal is to be prepared and submitted as Part One of the assignment. The details of the proposal document are given later in this document. Your teacher will look at your proposal and let you know if you can proceed to the implementation phase, i.e. the programming part.

2.4 The project does not have to be very large. Estimate the amount of work about the size of one of the recent assignments (in terms of time) for each group member or if you do it individually. You can roughly estimate about 40 hours per person.

2.5 The instructor should be able to easily trace the work of every group member in the project implementation.

2.6 Each class is to be programmed by a single person. The name of the person who has written the class must be present as comment in the code-file. Make sure that you follow a good programming style, as you have learned during the semester, and remember that the form classes should only maintain code that is related to GUI. All other logics must be placed in separate classes.

2.7 A group member must review and make a quality control of another group member's work. All members are responsible for the whole project.

2.8 In case of insufficient work or poor participation by a group member, the member will be requested by the instructor to submit complementary work in order to qualify for a passing grade. When the application as a whole is poorly done, the whole project will be returned for completion.

3. Project Proposal

In this part of the project you choose a topic, make an analysis of the problem and sketch a design of the solution (classes and their relations). This job is to be done **before** starting the programming work, absolutely not after! The proposal is submitted as a MS Word or a PDF document, consisting of about 3-4 pages. A checklist for the contents of the proposal is outlined below. Please note that the purpose of this report is not to write a user-manual, neither it is to explain the functionalities. Follow the outline below:

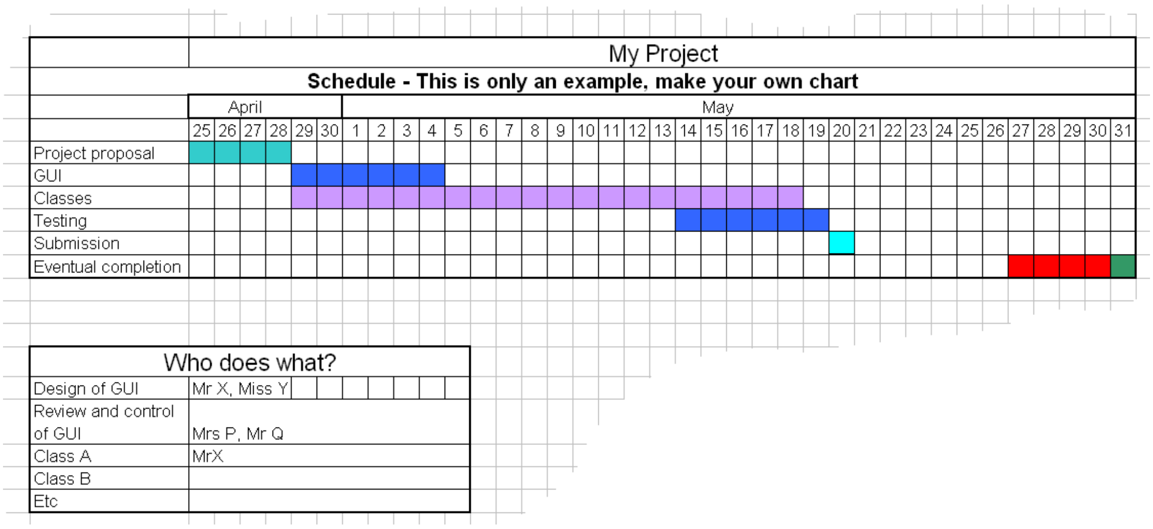
3.1 **A title page** with the name of the project, the name and personal number of every member.

3.2 **A brief description of the objectives**, why you chose this problem and who will be having the honour of using your software.

3.3 A **class-diagram must** be presented. A design for the solution of the problem showing the objects and their relations. Take your time and try to make a good design of your solution. This will minimize – and optimize – your programming job. You should particularly think about:

3.3.1 Finding objects.
3.3.2 Finding attributes and methods for every object
3.3.3 Determining the relations between them.

3.4 A **time schedule** showing your work plan for the project period, i.e. from start to the deadline. You can use an Excel sheet to make your planning. The following is only an example; make your own chart, may be more detailed.



	My Project																																				
	Schedule - This is only an example, make your own chart																																				
	April						May																														
	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Project proposal	🟦	🟦	🟦	🟦																																	
GUI					🟦	🟦	🟦	🟦	🟦	🟦																											
Classes					🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪													
Testing																				🟦	🟦	🟦	🟦	🟦	🟦												
Submission																										🟦											
Eventual completion																																	🟥	🟥	🟥	🟥	🟩

Who does what?	
Design of GUI	Mr X, Miss Y
Review and control of GUI	Mrs P, Mr Q
Class A	MrX
Class B	
Etc	

4. Project Implementation

This is the programming part of the project and based on your planning and design in the previous part. You should start this part before you have submitted the project proposal and obtained the approval of your instructor.

If you have a good solution design from the previous part, your programming work will also go smoothly and effectively. However, **the implementation doesn't have to follow the solution designed in the project proposal strictly**. Changes can be done when necessary without the approval of the instructor.

You are expected to add documentation into the code files in such an extent (not to exaggerate) that your instructor (or programmers in general) can easily follow your thoughts and anticipations in your programming code. By choosing names of classes, variables, methods and other identifiers that are informative and long enough, you can reduce the amount of documentation. Those parts of code that are self-descriptive require no extra documentation. The following criteria will be taken into consideration by the instructor when grading your project work:

4.1 **GUI**: The program must contain a graphical user interface (GUI) that is user-friendly and easy to handle. Some menus, for example for storage of data to files, an About Box and a program icon (ico, bmp, jpg, etc) should be included in the application.

4.2 **Error handling**: The program must run satisfactorily and should not crash for common errors such as division by zero, index out of range and unhandled exceptions due to invalid input given by the user.

4.3 **OOP**: It is expected that the basics of OOP that you have learned in this course, particularly encapsulation and inheritance are applied in the solution. Use of constructors and properties as well as private fields is very important.

4.4 **Data storage**: Data is to be saved into files and retrieved from files. Ordinary text files or xml files for storage of data are sufficient. Databases and data bindings are normally not allowed. In case you have special reasons for using these features, you must contact your instructor before starting your work.

4.5 You may use other technologies such WPF, XNA, LINQ or third-party components, but make sure to write enough code by yourself. Your own coding should be at least the size of Assignment 4 or Assignment 5.

5. Project Ideas – some examples

In case you don't figure out a project idea by yourselves, here are some suggestions:

- **Multimedia Store** selling/leasing music albums and films.
- **Library** giving loans of books, films, articles and newspapers.
- **Contact Registry** for your personal use or for your organization to store phone numbers, e-mail addresses and post addresses.
- **Animal Motel** where people can leave their pets (dogs, cats, snakes, rabbits, etc).
- **Booking Program** for dentists, barbers or other specialists for handling of appointments.
- **Real-Estate program** for property dealers showing different type of housings.

- **Food Recipes** for Mom (and sometimes Dad) who always wonder what to cook for super this evening.

6. Submission

Submission is done group-wise (one submission for the whole group) if you are working in a group and individually when working individually. Both, the Project Proposal and Implementation (Code Project) are submitted in the same way as other assignments. **Do not forget to include all group members when submitting either parts on the ITS.**

Congratulations!

You're now a programmer!
But, you may still face problems, and again, don't give up. Ask for help!

Farid Naisan,
Course responsible and Instructor