University of Macedonia
Department of Applied Informatics
Department of Information Systems
Course: **Systems Analysis and Design**
Teachers: Efthymios Tambouris and Kon/nos Vergidis

Group Work – Final Submission

In the final submission you are asked to submit three files:

1. A **docx or pdf** file with the complete analysis and design of the system describing the scenario given to you from the beginning of the lesson, 2. a **Visual Paradigm file ,** and 3. a **compressed file** with the source code as extracted from development environment software you will use.

The work must be complete and comprehensive. Your homework should be based on what we did each week. Your text should additionally contain an "Executive Summary" section at the beginning, an "Introduction" section, and a "Conclusions" section at the end. There should be two Annexes A and B at the end. Annex A entitled 'Group Self-Assessment' is given to you to complete at the end of this document. In Appendix B entitled 'Team Functioning and Timesheets' you will report how the team worked, if they met (so agendas and minutes are also needed) and each team member's timesheets with the sum of the total hours worked by each **member .**

The "Staff Summary" is very important. This is a summary of the paper and should not be a description of the table of contents, but should contain all the main points and conclusions of your study.

Your final text should be comprehensive and not require knowledge of other information to be understood. You should not, however, repeat the theory. Consider your text going to the client for approval, who does not know its contents beforehand. So you need to explain what you will present (eg in the introduction) and at the end summarize your conclusions.

There is no limit to the number of pages.

Indicative contents:

The work should include the following:

1) To create • a
system request (system request) • a feasibility
study for the new PES of EHDE.

* *For the system application you should limit yourself to the case study information. Whereas for the feasibility study where information is limited (e.g. no financial data), estimates will have to be made.*

2)

- To record the business processes (EB) identified in the case study.
- To model the processes, according to what the case study for the existing situation (AS-IS) states, using UML activity diagrams.
- To model, using the UML diagram of activities, the new processes, as they will be configured after the introduction of the new information system (TO-BE situation) and to explain how the problems that existed in the processes were dealt with.
- Choose one of the BPA, BPI, BPR methods and justify the choice.

** You do not need to show objects and Information Systems in the diagrams. Limit yourself to the information given in the pronunciation. If you absolutely have to make any additional admission this should be explicitly stated.*

3)

- To choose information collection methods. • To record
user stories and epics, based on the information presented in the case study. Then create the corresponding backlog taking into account the dependencies between the user stories and their priorities.

- To record the functional and non-functional requirements.

4) To create the UML diagram of use cases for the new PS.

5) Write the verbal descriptions of the PCs of the usage diagram that you presented earlier. Verbal descriptions should be accompanied by indicative (mock up) screens.

** Each team member will undertake to write verbal descriptions and mock up screens for 2 PX of the diagram. If your chart contains fewer VPs in total than members*2, then they will be assigned proportionally (as far as possible) to the team members.*

6) To record the domain model of the problem (domain model), illustrating the classes and the relationships between them. The text analysis method should be used in the verbal descriptions of all the system PXs that you presented earlier. In your reply,
in addition to the final model you should also include your working method as well as the most important intermediate results according to the principles of text analysis (ie initial list of nouns, which were excluded and why etc). (It is expected that the domain model will also contain any fields that have already been discovered from the verbal descriptions).

7) To create 2 object diagrams showing basic elements of the diagram classes.

8) Revise the verbal descriptions of the use cases in the design stage, highlighting the differentiated points.

9) To create the sequence diagrams of the use cases of the scenario. Sequence diagrams should include the basic as well as alternative flows for the respective use cases. In addition, you should detail the messages being exchanged in these diagrams so that it is clear which of them are functions (with their parameters, if any) and which are user actions/commands.

10) Identify the elements resulting from the construction of the sequence diagrams and the final form of the verbal descriptions. Enrich the class diagram with the above elements. The rich class diagram should detail anything that can be extracted from the design stage.

11) Implementation.

- Based on the results of the previous topics, proceed with the implementation of the system classes with an emphasis on the functionality of the class associations (message exchange).

- For purposes of testing/demonstrating the functionality of the program, you will develop in the main() method a suitable driver program (through commands to create objects, connections between them and calling methods on these objects).
- You can also implement a simple text menu that performs basic functions. • What we are mainly interested in is the demonstration of messaging and usage Lists!
- No need to build Database and GUI (it is optional).

Finally, provide main with appropriate print commands that will inform about the action that has just been successfully executed (eg "New appointment has been created"). Add to the object classes used in main, printData() method that will display the property values of each object. At the end of main call through appropriate for loops the printData() method on the objects
Catalogues.
Note that if you need to implement any method or attribute (property, variable) that is not described in the class or sequence diagram, you should explain (with a short and concise comment within the code) why this need arose and why it is not documented in the previous diagrams . In the text of your paper **you should mention** whether or
not the code is complete and running correctly, and anything else you find useful.

The main application class is CarOps.java, which when executed (calling the main method) performs the following functionality **(or some equivalent based on your design):**

- Constructs 3 objects of the RepairJob class and places them in an object list type Repair Work. The objects created have the following data:
  - o Repair Task1("Oil Change", 20 euros)
  - o Repair Job2("Cab filter change", 5 euros)
  - o Repair Task3("Brake Maintenance", 30 euros)
- Constructs 6 objects of the PartType class and places them in a list of objects type TypePart. The objects created have the following data:
  - o Spare Part Type1("Pack of 4lt oils", 30€)
  - o Part Type2("Oil filter", €20)
  - o Spare Part Type3("Cabin filter", €30)
  - o Spare Part Type4("Front wheel brake pad", €5)
  - o Spare Part Type5("Rear wheel brake pad", €5)
  - o Spare Part Type6("Brake fluid", 10 euros)

- Constructs 2 objects of the Repair class and associates them with appropriate objects that represent the repair operations and parts usages performed as part of the repairs. It then adds them to a list of objects. The objects created have the following data:

  o Repair1: includes performing the tasks RepairWork1 and RepairWork2 and using one (1) piece each of SpareType1, SpareType2 and SpareType3 spare parts. The duration of the repair is 1 day.

  o Repair2: includes performing RepairTask3 and using 4 pieces of SpareType4, 4 pieces of SpareType5 and 1 piece of SpareType6. The duration of the repair is 2 days.

- Prints the Repair Jobs, Parts Types, and Repairs items to the screen by looping through the above item lists using iteration structures. Specifically:

  o For each Repair Work, it displays its name and cost on the same line,

  o For each Type of Spare Part, it displays the name and cost per on the same line piece,

  o For each Repair, it shows the repair duration in days and the total repair cost on the same line. It should be noted that the total repair cost is calculated using the getTotalCost() method belonging to the Repair class. The method calculates the cost as the sum of the labor cost and the cost of parts used in the repair. The cost of each use of a spare part is obtained as a product of the cost of the specific type of spare part with the quantity used in the repair. Regarding the cost of work, it is obtained by adding up the individual costs of the work performed in the context of the repair.

Notes:
- To create the UML diagrams you should use Visual Paradigm. However, we remind you that the diagrams should also be inserted in the text as images.
- As a code development tool you should use Eclipse (www.eclipse.org). The folder containing the corresponding project with the source code should be delivered compressed.
- All assumptions that need to be made should be recorded.

ATTENTION: Any copying automatically leads to nullification of ALL members of ALL groups, regardless of who copied from whom.

Appendix A: Group Self-Assessment

Each group should complete the following tables and data:

Job Self-Assessment:

• Work grade for effort (1-10): • Work grade for end result (1-10): • Final grade for your teamwork (1-10):

Argue why you should get this grade (up to 150 words – no more).

Comparative contribution of each member to the final result:

Full name	Contribution (%)
Total	100%

The sum of all individual (individual) contributions should be 100%. It is therefore a relevant contribution. If you agree on the percentages, there is likely to be a corresponding differentiation in the grade of the team members. In case you disagree on the relevant percentages, you can not fill in the table, in which case all team members will get the same score.

Appendix B: Group Mode and Timesheets

It will be completed according to the instructions given above.