Stage 2 Team Project (IN2018) module 2022/3

Assessment Criteria

Author: Dr Martin Walter
based on a document by Ilir Gashi, Peter Mellor, Peter Popov, and Vladimir Stankovic

Version: 1.0

History

Version	Note	Date	Author
1.0	Created, based on Developing a Software Product The GARage IT System (GARITS) Student's Brief (Feb 2022)	28 January 2023	MW

Contents

1 General criteria for documents

Written documents submitted for marking will be assessed using the general criteria covering presentation and style. In addition to this, each particular deliverable will be assessed according to the criteria covering its content as listed below. Fairness of the marking will be ensured by the same section of all deliverables being marked by the same marker(s).

For Requirements Specification & System Design document, and Individual Reports, 10% of the overall mark will be awarded for General Criteria.

General Criteria:

- Appropriate title, page numbering and version control (2)
- Introduction, including Purpose & Scope, of the document (2)
- Use of language appropriate to audience (consultant vs. customer), and Spelling and Grammar (2)
- Clear layout and structure (2)
- Appropriate use of graphics and diagrams (2)

Total: 10%

2 Requirements Specification and System Design (Due: Week 5, 40% of total project marks)

This deliverable will be assessed against the following particular criteria:

- Requirements Specification
  - Description of the existing system (what the company have already). This should simply be in natural language, using your own words. Students should not, however, "design" the current system; (5)
  - Full use case diagram(s) defining collections of use cases and their interactions with actors covering the whole functionality of the new system; (15)
  - Use case specifications for **10 key use-cases**, with main and alternative course of actions made clear, actors defined, pre- and post-conditions stated and all interactions specified; (15)
  - Indexed list of all use cases prioritised according to users' priorities and impact of projected risks during development (i.e., time and budget problems). This should help drive the evolution of the design and implementation tasks. (5)
- System Design
  - Fully refined and correct Design class diagram(s) showing Entity, Boundary (i.e. GUI) and Control classes, associations (including roles and navigability), cardinalities, methods (i.e. operations) and attributes. A complete set of operations should be specified including: parameter lists, return types, visibility, exceptions, set and get operations/methods, constructors and destructors. Also, a complete set of attributes including types and default values must be provided.
    The class diagram needs to include classes from the implementation domains (e.g. DB connectivity). Packages should be used to show the system architecture and the interfaces (and the respective implementation classes) between the sub-systems. (20)
  - ER diagram, and relational database schema (specified to the 3rd Normal form) represented with a complete set of DDL statements (CREATE TABLE statements). Also, a

representative set of Data Manipulation Language (DML) statements (2 SELECTs, 2 INSERTs, 2 UPDATEs and 2 DELETEs) must be provided i) adhering to the team's DB schema and ii) with meaningful values. Also, all necessary SQL DML statements which are needed to create **2** non-trivial reports pertaining to the case-study must be provided.

    - Class diagram is not the same as database schema. In any case, the teams are expected to use Relational Database Management Server (RDBMS) – in this case MySQL, not an OO DBMS. The database chosen should offer transactional support.

(20)

- GUI designs. **Design** the visual appearance of the GUI forms (screenshots); **map** these to the boundary classes shown in the class diagram; and **show** how the users will **navigate** through the GUI, i.e. through the chosen set of menus/forms etc.

(10)

Total: 90%. The remaining 10% of the marks for this deliverable will be awarded for the General Criteria.

The target audience for this deliverable is the customer and the team consultant. It should provide explanatory material in layman's terms (i.e., use cases) complemented by technical specifications represented in a structured fashion (e.g. class diagram).

3 Working Product (Due: Week 11 and 12, 50% of total project marks)

Accompanying the final software system should be an Implementation Report describing the implemented software, problems encountered in implementing the initial design, solutions/fixes, and the results of testing.

A copy of the fully commented source code, or in the case of code generated automatically the appropriate files used by the code generating software, should be submitted in the dedicated submission area on Moodle, as a .zip file. **Failure to supply the source code will result in Zero marks being awarded for the whole of the Working Product stage.**

The working product will be assessed using the following criteria:

- Working product Demo
    - Completeness and quality of the product in relation to the requirements (75)
    - Ease of use and consistency of the GUI (5)
- Implementation Report
    - Software architecture/Compilation/Run-time components (8)
    - Testing plans and reports (12)

Teams using code generation environments (e.g. NetBeans, Eclipse, IntelliJ IDEA, Visual Studio etc.) should describe the structure of their implementation by referring to the following:

- Forms and their properties
- Controls and dialog boxes, menus etc., and their properties
- Database tables
- Coded modules

During the working product "Demo", a checklist will be used by the staff to award marks for the presence of features in the product which are stated in the requirements, and also the quality of

implementation of those features. Functionality will be assessed using two **sets of scenarios**, which describe the expected functionality of the product. These scenarios, which will be executed with the completed product after its deployment on a target machine, will be made available to the teams as follows:

1. The first set of scenarios will be given to the teams **at least two weeks before the final demo**. The teams must use this set to prepare for the final demo. Before the start of their final demo the teams are expected to have cleared the database used by the application and to have executed the entire first set of scenarios. **At the start of the "Demo" assessment the database used by the application should contain the data generated by executing the scenarios included in the first set and nothing else.**
2. At the demonstration each team will be presented with another set of scenarios, similar to those included in the first set distributed in advance, and asked to execute them with their product.

The marker of the Demo will ask the demonstrator(s)/team to show features in accordance with the requirements.

The software demonstration ("Demo"), and the Showcase event, is planned to take place in week 11. The details about the Demo, and the Showcase event if one is to be held, (including the schedule, etc.) will be **confirmed about two weeks before the demonstration.**

The target audience for the product is the customer, except for the Implementation Report, which is for the consultant.

4 Individual Final Report (Due: Week 12, 10% of total project marks)

The target audience for the Individual Report is the consultant. Students ought to use the teams' Project Binder and the respective Individual Diary as inputs when writing the Individual Report. The individual report will be assessed using the following criteria:

- Table of the effort expended in each phase against estimated effort. In case of (substantial) discrepancy an explanation must be provided. Details need to be provided about *own* contribution to the team deliverables. (25)
- Description of how the project went and statement of any problems *personally encountered* and how they could be avoided in future. (15)
- Description of how the team worked together, and providing evidence-based assessment *about the contribution of the other members* to the team deliverables. (15)
- Lessons learnt. (20)
- Statement of what you would do differently in future. (15)

Total: 90%. The remaining 10% of the marks for this deliverable will be awarded for General Criteria.

5 Originality of Work

Although the teams are starting from the same statement of requirements, each team is expected to work independently and to tackle the problem in their own way. Obviously, there will be overall similarities in the deliverables from different teams, and there will be plenty of opportunities to see what other teams are doing, either in the class/Q&A sessions or in private discussions between the members of different teams. However, copying of (parts of) documents or software of one team by

another is, of course, not allowed, and will be treated as plagiarism of coursework! If copying is detected[1], all students/teams involved will be given **zero** marks for the copied deliverables.

[1] Checks will be made! For example, via automatic comparison of source code submissions, etc