**IT306 Project Description**

As you know, the Software Development Lifecycle is comprised of several stages. Often, in a development environment, different groups are responsible for only a part of the process. The goal of this project is to give you experience at different stages, where each stage will have you working with a different application. The purpose is to allow you to experience working on phases of several projects, rather than following the same project through its lifecycle.

|  |  |
| --- | --- |
| **Deliverable** | **Points Possible** |
| Phase I: Design and Analysis | 5 |
| Phase II: Business Requirements and Preliminary Project Requirements Document | 20 |
| Phase III: Evaluation | 5 |
| Phase IV: Final Requirements Document | 15 |
| Phase V: Preliminary System Implementation | 30 |
| Phase VI: Final System Implementation | 20 |
| Phase VII: Evaluation of the Final System | 5 |
| Project Presentation | 5 |

The project is assigned to a group of 2 students. You have the opportunity to choose your partner if you do so by the end of week 1, or I will assign teams. You should plan on contacting your group member as soon as posted as the first deliverable is due shortly thereafter. Groups will be posted on the Blackboard site. Due dates for the deliverables are absolutely firm. Failure to submit requirements by the published due date will result in a non-negotiable zero.

As a group member, you will be designated as either ‘Person A’ or ‘Person B’.

Your grade will be based on individual **and** group effort.

To share the content of each project, you are required to use one of the following cloud-based source code management services:

1. Bitbucket: <https://bitbucket.org/>

Tutorial: <https://confluence.atlassian.com/bitbucket/tutorial-learn-git-with-bitbucket-cloud-759857287.html>

1. Github: <https://github.com/>

Tutorial: <https://guides.github.com/activities/hello-world/>

**Project Milestones and Deliverables**

**Participation Report**

Each team should select a leader to keep track of the member activities and their level of involvement in the project. The instructor will request a 1-page report from the team leader at the end of the semester on the percentage of the work accomplished by each member..

**Phase I – Design and Analysis**

This is description of the application you define. Your team will create a comprehensive project description document for a system of your choice. However, another group will use this document to develop the system. Please make sure the description is clear and addresses the fundamental pieces of the system.

Remember as a software developer, your job is made easier by having a list of things you need to engineer be clearly defined. (Consider the frustration you might have felt in starting this advanced class. In prerequisite courses, your instructor may have listed the assignment requirements in detail and gave you images of the intended user interface. In IT306, we do not provide as much guidance. The goal is to have you learn more about program design.) So, your document should be clear, accurate, complete and non-contradictory. Ill-defined requirements can lead to the need to rework the solution, a poor quality product, inability to complete the product on time and frustrated classmates.

**Deliverable**

**Format: Word or PDF document**

**Submission: One for group**

**Phase II**

**Business Requirements and Preliminary Project Requirements Document**

You and your partner should choose a product to design. At this stage, you will not be writing any code or specifically describing how a solution is to be approached. You will be creating a description of your envisioned product for the system proposed by another group.

You are to create a statement of the business rationale for the project. First determine a vision for the product. This statement should document:

* The reason the system is being developed,
* The intended users of the system,
* What it will accomplish for its users, and
* The scope to clarify capabilities that are and are not to be provided by the product.

At this point in the project you do not need to determine *how* the system will be designed. However, all products for the purposes of this class project will need to include some basic program design requirements. It might be helpful at this time to keep these in consideration as you choose a product and produce a statement:

* Inheritance
* Polymorphism
* Exception Handling
* Data files
* Dynamic Data Structures (plan on using data structure you learned so far, including arrays, linkedlist, queue, stack, etc.)

This document should be descriptive. You may include images if it helps clarify the intent of the product. If your rationale is a simple sentence, you may receive NO credit. Use this as a vehicle to **sell** your product before it is developed!

The project requirements document serves several purposes including User Requirements and Software Requirements. The User requirements define the project from the user’s point of view. This includes what the user should be prepared for in terms of input requirements and what the user can expect as output. “*Users”* is a loose term and can mean not only people that interact with the system, but devices – in the case of this project, files. The Software Requirements provide detailed descriptions of all of the functional and non-functional requirements the software must fulfill to meet business and user needs. The requirements documentation will provide an agreement between business managers and technical specialists on the scope of the product. Do a little research on format for Requirements Documentation and decide what suits your team the best.

For the purposes of this phase of the IT306 project, your group of two will be working in parallel on the specifications for the system you outlined in the Business Requirements Documented submitted for Phase I. One person (noted as Person A) will focus on the User Requirements, and the other (noted as Person B) will focus on the Software Requirements. *(Note the roles will be reversed during* *the next phase.)* Prior to submission, the document must be merged and submittedas one. It is assumed that there will be communication between the group members to facilitate the completion of these individual parts. It is recommended to complete this phase iteratively and provide opportunity for group members to make changes.

You do not need to write any code to complete this part of the assignment. Your design documents must be descriptive and complete. The document should flow nicely and be readable. You may use bullet points when necessary, but it should primarily be narrative. Clearly indicate names and email addresses of group members on the document.

See the Requirements on the next page.

For a group of 2:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Person** |  |  | **Requirement** |  |  | **Points** |  |  |
|  | A and B |  |  | Descriptive narrative of project’s purpose. The Business |  |  | 1 |  |  |
|  |  |  |  |  |  |  |
|  |  |  | Requirements may be used as a basis. |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  | A and B |  |  | Definition of project scope (including intended user population |  |  | 1 |  |  |
|  |  |  |  |  |  |  |
|  |  |  | and product functionality) |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  | A and B |  |  | Discussion of project limitations or requirements defects |  |  | 1 |  |  |
|  |  |  |  | User: Narrative of what users will be able to do with the |  |  |  |  |  |
|  | A |  |  | product and a description of how the application will execute |  |  | 3 |  |  |
|  |  |  |  | (ignore details of the data, focus is on *how* the product works) |  |  |  |  |  |
|  | A |  |  | User: Images of intended User Interface (expected views of |  |  | 8 |  |  |
|  |  |  |  |  |  |  |
|  |  |  | input requests and supplied output from keyboard and display) |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  | User: Outline of test plan – descriptions of UI items that |  |  |  |  |  |
|  | A |  |  | should be included when test plan is defined. Specifically what |  |  | 3 |  |  |
|  |  |  |  |  |  |  |
|  |  |  | input from a user or a data file will be required, what types of |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  | exceptions might occur based upon user or file input. |  |  |  |  |  |
|  | B |  |  | Software: Narrative of interaction between the classes that |  |  | 3 |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  | define the data (including the intended inheritance hierarchy) |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  | Software: Class APIs – for each of the class diagrams provided |  |  |  |  |  |
|  | B |  |  | include a description of all variables and methods included in |  |  | 8 |  |  |
|  |  |  |  | the class |  |  |  |  |  |
|  |  |  |  | Software: Outline of test plan – descriptions of data and |  |  |  |  |  |
|  | B |  |  | activities that should be included when a test plan is developed. |  |  | 3 |  |  |
|  |  |  |  |  |  |  |
|  |  |  | Specifically what the data requirements are and what types of |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  | exceptions may occur due to the restrictions on the data |  |  |  |  |  |
|  | A and B |  |  | Projected Usability of the Document (Comments provided) |  |  | 3 |  |  |

For a group of 3:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Person** |  |  | **Requirement** |  |  | **Points** |  |  |
|  | A,B, and C |  |  | Descriptive narrative of project’s purpose. The Business |  |  | 1 |  |  |
|  |  |  |  |  |  |  |
|  |  |  | Requirements may be used as a basis. |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  | A,B, and C |  |  | Definition of project scope (including intended user population |  |  | 1 |  |  |
|  |  |  |  |  |  |  |
|  |  |  | and product functionality) |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  | A, B, and C |  |  | Discussion of project limitations or requirements defects |  |  | 1 |  |  |
|  |  |  |  | User: Narrative of what users will be able to do with the |  |  |  |  |  |
|  | A |  |  | product and a description of how the application will execute |  |  | 3 |  |  |
|  |  |  |  | (ignore details of the data, focus is on *how* the product works) |  |  |  |  |  |
|  | A |  |  | User: Images of intended User Interface (expected views of |  |  | 8 |  |  |
|  |  |  |  |  |  |  |
|  |  |  | input requests and supplied output from keyboard and display) |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  | User: Outline of test plan – descriptions of UI items that |  |  |  |  |  |
|  | A and C |  |  | should be included when test plan is defined. Specifically what |  |  | 3 |  |  |
|  |  |  |  |  |  |  |
|  |  |  | input from a user or a data file will be required, what types of |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  | exceptions might occur based upon user or file input. |  |  |  |  |  |
|  | B |  |  | Software: Narrative of interaction between the classes that |  |  | 3 |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  | define the data (including the intended inheritance hierarchy) |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  | Software: Class APIs – for each of the class diagrams provided |  |  |  |  |  |
|  | B and C |  |  | include a description of all variables and methods included in |  |  | 8 |  |  |
|  |  |  |  | the class |  |  |  |  |  |
|  |  |  |  | Software: Outline of test plan – descriptions of data and |  |  |  |  |  |
|  | B and C |  |  | activities that should be included when a test plan is developed. |  |  | 3 |  |  |
|  |  |  |  |  |  |  |
|  |  |  | Specifically what the data requirements are and what types of |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  | exceptions may occur due to the restrictions on the data |  |  |  |  |  |
|  | A and B |  |  | Projected Usability of the Document (Comments provided) |  |  | 3 |  |  |

**Deliverable**

**Format: Word or PDF document**

**Submission: One for group**

**Phase III Evaluation**

**Document Evaluation Scoring Guide**

**Submitted by all students**

**Your Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Project Evaluated: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Use the provided scale to rate the system you have requested to evaluate. You must provide comments for each entry. This evaluation is to be submitted by EVERY student and not done as a team. Your evaluation must have constructive comments. The implementing team will use the constructive comments towards designing their application.**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Requirement** |  |  | **Points** |  |  | **Points** |  |  | **Comments** |  |  |
|  |  |  |  |  |  |
|  |  |  | **Possible** |  |  | **Awarded** |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  | Project Narrative | | 5 | |  |  |  |  |  |  |  |  |
|  | Defined Project | | 2 | |  |  |  |  |  |  |  |  |
|  | Scope | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  | Defined Project | | 2 | |  |  |  |  |  |  |  |  |
|  | Limitations | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  | Narrative of user | | 5 | |  |  |  |  |  |  |  |  |
|  | experience | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  | UML Diagrams | | 20 | |  |  |  |  |  |  |  |  |
|  | (other than classes) | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  | UI images | | 5 | |  |  |  |  |  |  |  |  |
|  | Test Plan outline – | | 5 | |  |  |  |  |  |  |  |  |
|  | complete and feasible | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  | Narrative of data | |  |  |  |  |  |  |  |  |  |  |
|  | relationships and | | 5 | |  |  |  |  |  |  |  |  |
|  | purposes | |  |  |  |  |  |  |  |  |  |  |
|  | UML Class diagrams | |  |  |  |  |  |  |  |  |  |  |
|  | – include all variables | | 10 | |  |  |  |  |  |  |  |  |
|  | and methods indicate | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  | +/- and parameters | |  |  |  |  |  |  |  |  |  |  |
|  | Class APIs – | |  |  |  |  |  |  |  |  |  |  |
|  | complete description | | 15 | |  |  |  |  |  |  |  |  |
|  | of methods | |  |  |  |  |  |  |  |  |  |  |
|  | Test Plan outline for | | 5 | |  |  |  |  |  |  |  |  |
|  | validity of data | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  | Overall Quality of | |  |  |  |  |  |  |  |  |  |  |
|  | Requirements | | 6 | |  |  |  |  |  |  |  |  |
|  | Document | |  |  |  |  |  |  |  |  |  |  |

**Phase IV: Final Requirements Document**

For Group of 2 students:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Person** |  |  | **Requirement** |  |  | **Points** |  |  |
|  | A and B |  |  | Required changes made to preliminary document |  |  | 3 |  |  |
|  |  |  |  | User: UML Diagrams – |  |  |  |  |  |
|  |  |  |  | o Actor Table (user descriptions – including people and files) |  |  |  |  |  |
|  |  |  |  | o Use Cases (description in abstract terms how actors use the |  |  |  |  |  |
|  | A |  |  | system to accomplish goals summarizing a set of related |  |  | 12 |  |  |
|  |  |  |  |  |  |  |
|  |  |  | scenarios) |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  | o Data Flow Diagram (show related inputs, processes and |  |  |  |  |  |
|  |  |  |  | outputs for the processes) |  |  |  |  |  |
|  |  |  |  | o Activity Diagram (model illustrating the flow of use cases) |  |  |  |  |  |
|  |  |  |  | Software: UML Diagrams: |  |  |  |  |  |
|  | B |  |  | o Class Diagrams for each data definition class used by the |  |  | 12 |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  | system |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  | o Activity Diagrams for system implementation |  |  |  |  |  |

For Group of 3 students:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Person** |  |  | **Requirement** |  |  | **Points** |  |  |
|  | A, B,C |  |  | Required changes made to preliminary document |  |  | 3 |  |  |
|  |  |  |  | User: UML Diagrams – |  |  |  |  |  |
|  |  |  |  | o Actor Table (user descriptions – including people and files) |  |  |  |  |  |
|  |  |  |  | o Use Cases (description in abstract terms how actors use the |  |  |  |  |  |
|  | A |  |  | system to accomplish goals summarizing a set of related |  |  | 12 |  |  |
|  |  |  |  |  |  |  |
|  |  |  | scenarios) |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |
|  | C |  |  | o Data Flow Diagram (show related inputs, processes and outputs for the processes)  o Activity Diagram (model illustrating the flow of use cases) |  |  | 12 |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | Software: UML Diagrams: |  |  |  |  |  |
|  | B |  |  | o Class Diagrams for each data definition class used by the |  |  | 12 |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  | system |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  | o Activity Diagrams for system implementation |  |  |  |  |  |

**Phase V: Preliminary System Implementation**

**System Development**

System development involves creating the project described in the requirements documentation. This will requires you to create the solution based upon the defined requirements of the project. The preliminary solution should include most of the class definition functionality and a working implementation. It is not expected that the system will be completely functional, but should be approximately 80%. Group members should be collaborating throughout this entire phase. If **any** changes are made to the requirements provided, a log file must be created and discussion with the requirements group noted.

One group member (noted as Person A) is responsible for creating the data class definitions based upon the Software Requirements provided. The other group member (noted as Person B) is responsible for creating the implementation that uses those classes. Person A is required to implement the Exception Handling in the system. It is expected that group members will work together as necessary and help with the overall completion of the project. A written test plan is to be submitted, along with the code that provides information on how to best validate the product’s robustness. Using the Test Plan Outlines provided in the requirements documentation as a base, provide a plan for the Quality Assurance engineers to use to verify that your system works as intended.

Your goal should be a completely functional product by the time you submit your Preliminary System. There should be minimal changes needed for the final system.

**Deliverable: Phase IV - Preliminary System Implementation**

**Format: JAVA files**

**Submission: One for group**

**Phase V: Preliminary System Implementation Grading**

**For Groups of two people:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Person** |  |  |  | **Requirement** |  |  | **Points** |  |  |
|  | A |  |  | Classes: Adherence to the structure provided by Class | |  |  | 5 |  |  |
|  |  |  |  |  |  |  |
|  |  |  | Diagrams in the requirements document | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  | Classes: Adherence to OO Design standards | |  |  |  |  |  |
|  | A |  |  | o | Information Hiding |  |  | 8 |  |  |
|  |  |  |  |  |  |  |
|  |  |  | o | Validation of data |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  | o | Appropriate Inheritance Structures |  |  |  |  |  |
|  |  |  |  | Exception Handling: Application should be protected | |  |  |  |  |  |
|  | A |  |  | from Exceptions raised in situations presented by file i/o | |  |  | 8 |  |  |
|  |  |  |  |  |  |  |
|  |  |  | and user input (and any other exceptions inherent to the | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  | product) | |  |  |  |  |  |
|  | A |  |  | Classes and Exceptions: Code Efficiency | |  |  | 4 |  |  |
|  |  |  |  | Implementation: Adherence to the structure provided | |  |  |  |  |  |
|  | B |  |  | by the Use Cases and Activity Diagrams in the | |  |  | 5 |  |  |
|  |  |  |  | requirements document | |  |  |  |  |  |
|  | B |  |  | Implementation: Appropriate design of methods | |  |  | 8 |  |  |
|  | B |  |  | Implementation: Appropriate use of data files | |  |  | 8 |  |  |
|  | B |  |  | Implementation: Code Efficiency and Design | |  |  | 4 |  |  |
|  | A and B |  |  | Quality of Product, based upon Requirements Document | |  |  | 5 |  |  |
|  |  |  |  |  |  |  |
|  |  |  | (Comments provided) | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

**For Groups of three people:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Person** |  |  |  | **Requirement** |  |  | **Points** |  |  |
|  | A |  |  | Classes: Adherence to the structure provided by Class | |  |  | 5 |  |  |
|  |  |  |  |  |  |  |
|  |  |  | Diagrams in the requirements document | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  | Classes: Adherence to OO Design standards | |  |  |  |  |  |
|  | A,C |  |  | o | Information Hiding |  |  | 8 |  |  |
|  |  |  |  |  |  |  |
|  |  |  | o | Validation of data |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  | o | Appropriate Inheritance Structures |  |  |  |  |  |
|  |  |  |  | Exception Handling: Application should be protected | |  |  |  |  |  |
|  | A |  |  | from Exceptions raised in situations presented by file i/o | |  |  | 8 |  |  |
|  |  |  |  |  |  |  |
|  |  |  | and user input (and any other exceptions inherent to the | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  | product) | |  |  |  |  |  |
|  | A |  |  | Classes and Exceptions: Code Efficiency | |  |  | 4 |  |  |
|  |  |  |  | Implementation: Adherence to the structure provided | |  |  |  |  |  |
|  | B,C |  |  | by the Use Cases and Activity Diagrams in the | |  |  | 5 |  |  |
|  |  |  |  | requirements document | |  |  |  |  |  |
|  | B,C |  |  | Implementation: Appropriate design of methods | |  |  | 8 |  |  |
|  | B |  |  | Implementation: Appropriate use of data files | |  |  | 8 |  |  |
|  | B,C |  |  | Implementation: Code Efficiency and Design | |  |  | 4 |  |  |
|  | A ,B, and C |  |  | Quality of Product, according to the Requirements Document | |  |  | 5 |  |  |
|  |  |  |  |  |  |  |
|  |  |  | (Comments provided) | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

NOTE: If during development, changes need to be made to the requirements document:

o They must be done with the approval of the group responsible for the requirements

1. A change log for the requirements document must be submitted for this phase (include a copy of an email with the discussion)

**Phase VI: Final System Implementation, System Evaluation and Documentation**

For Group of Two:

|  |  |  |
| --- | --- | --- |
| **Person** | **Requirement** | **Points** |
|  | Suggested changes made to the preliminary system on the implementation side |  |
| A | (Classes – including Inheritance structure and Exception | 8 |
|  | Handling) based on comments provided. |  |
|  | Suggested changes made to the preliminary system on the implementation side |  |
| B | (Implementation – including data files) based on  comments provided. | 8 |
|  |
| A, B | System Testing and Documentation | 7 |

For Group of Three:

|  |  |  |
| --- | --- | --- |
| **Person** | **Requirement** | **Points** |
|  | Suggested changes made to the preliminary system on the implementation side |  |
| A,C | (Classes – including Inheritance structure and Exception | 8 |
|  | Handling) based on comments provided. |  |
|  | Suggested changes made to the preliminary system on the implementation side |  |
| B,C | (Implementation – including data files) based on  comments provided. | 8 |
|  |
| A, B,C | System Testing and Documentation | 7 |

|  |
| --- |
| **Deliverable: Phase V - Final System Implementation Format: JAVA Files**  **System Evaluation and Documentation Format: Word or PDF** |
| **Submitted by: One for group** |

**System Testing and Documentation**

Quality Assurance is a vital part of the software development process. This part requires both individuals to accept a set of materials: Requirements Documents and a Completed System. As a reviewer, your responsibility is to determine if the system works as the requirements document intended, if the system is free of potential runtime errors and to provide external documentation of the system. By completing the following table, you will describe the effectiveness of the system when compared to the requirements document. The instructor will use this table during the project presentation to evaluate the effectiveness of your work. All the evaluated cases should have been implemented before the presentation. In the comments section, you have to provide concrete comments addressing the corresponding requirement.

All the members of the group that defined the project in the first place must evaluate your implementation. **It is your responsibility to contact the group and provide the material to them. If you fail in doing so, you will lose the credit for this part.**

**Scoring Guide: Evaluation of Final System**

(**5 points)**

**Use the provided scale to rate the system you have tested. You must provide comments for each entry. This phase is completed by the group that originally defined your project.**

**(see the next page)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Requirement** |  |  | **Points** |  |  | **Points** |  |  | **Comments** |  |  |
|  |  |  |  |  |  |
|  |  |  | **Possible** |  |  | **Awarded** |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  | Result of testing procedure | | 10 | |  |  |  |  |  |  |  |  |
|  | Class code | |  |  |  |  |  |  |  |  |  |  |
|  | ‘matches’ class | | 15 | |  |  |  |  |  |  |  |  |
|  | diagrams | |  |  |  |  |  |  |  |  |  |  |
|  | Acceptable OO | | 15 | |  |  |  |  |  |  |  |  |
|  | Design | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  | All Appropriate | |  |  |  |  |  |  |  |  |  |  |
|  | Exception | | 15 | |  |  |  |  |  |  |  |  |
|  | Handling | |  |  |  |  |  |  |  |  |  |  |
|  | Efficiency of | |  |  |  |  |  |  |  |  |  |  |
|  | class code and | | 10 | |  |  |  |  |  |  |  |  |
|  | exception | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  | handling | |  |  |  |  |  |  |  |  |  |  |
|  | Implementation | |  |  |  |  |  |  |  |  |  |  |
|  | follows design | |  |  |  |  |  |  |  |  |  |  |
|  | provided by Use | | 10 | |  |  |  |  |  |  |  |  |
|  | Cases and | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  | Activity | |  |  |  |  |  |  |  |  |  |  |
|  | Diagrams | |  |  |  |  |  |  |  |  |  |  |
|  | Use of files | | 10 | |  |  |  |  |  |  |  |  |
|  | Correct use of | |  |  |  |  |  |  |  |  |  |  |
|  | dynamic data | | 20 | |  |  |  |  |  |  |  |  |
|  | structures | |  |  |  |  |  |  |  |  |  |  |
|  | Code Efficiency | |  |  |  |  |  |  |  |  |  |  |
|  | and design | | 10 | |  |  |  |  |  |  |  |  |
|  | (implementation) | |  |  |  |  |  |  |  |  |  |  |

**Project Presentation**

**Demo of your project**

This part should be equally divided by the members. All members should participate in the project presentation. You need to feature the functionality of your system. The instructor will ask you about the functionalities, users, validations, testing, project plan, and the challenges related to the project. Be honest about the feasibility of your implementation. Create a test plan in advance to show all interesting features to your instructor. Make sure your code is running well without being trapped by unexpected bugs and errors. Check your code for different types of data validations.