**EECS3311 Group Project**

**Summer 2025**

**Deliverable 1**

**Purpose of the First Deliverable**

The purpose of this deliverable is to produce

1. A collection of sequence diagrams for each use of the 8 use cases presented in the Project Description Document
2. A list of functionalities that need be implemented (e.g. data loading, data tabulation, data visualization, comparisons etc.)
3. An initial decomposition of your system in the form of modules and their interactions. You may use a blocks and arrows type of diagram, or if you would like to try, use a component diagram
4. An implementation plan in the form of a GANTT Diagram (<https://en.wikipedia.org/wiki/Gantt_chart>)
5. Test cases which will guide your development (i.e. the test cases which your code has to pass for implementing the use cases – see Test-Driven Development approach). We will expect to see initially at least 10 test cases. For example, for the meal logging module some test cases can be

* Input: Ingredients and quantities provided
* Output: Data about calories, food groups and nutrients is correctly fetched from the database.
* Input: Ingredient or meal not found in the database provided
* Output: The data is not fetched, message appears
* Input: Partial data is not provided (e.g., quantity or name of ingredient)
* Output: The data is not fetched, message appears

1. The implementation of an initial set of system functionality. Some suggestions for source code implementation for deliverable 1 include:
2. The profile creation
3. The meal logging functionality
4. The food item swap functionality
5. Correct use of the JFreeChart module and at least one visualization working.
6. At least one calculation (e.g., after one swap) happening in the background and confirmed with the right test case.
7. A report as discussed below in Section “What to hand In”

In this Assignment, you will learn to:

* Analyze user requirements and use cases
* Produce sequence diagrams as a requirements specification aid
* Decompose a complex problem into sub-problems
* Start understanding and addressing software architecture design
* Produce an initial Software Design Document (SDD) from use cases
* Initiate implementation using design patterns. Keep in mind that the final implementation has to have six design patterns.

**Assigned**

Tuesday, May 20, 2025 (please check e-class regularly for any updates or revisions).

**Due**

This assignment is due Friday, June 20, 2023 by 11:59 pm (midnight-ish) by e-class submission. Please check e-class regularly for any updates regarding the submission process). One submission per group.

**Late Penalty**

Late project will be accepted for up to two days after the due date, with weekends counting as a single day. Submissions late up to 24 hours will incur a penalty of 10%, and submissions up to 48 hours will incur a penalty of 20%. No submissions after 48 hours of the deadline are accepted. The same time windows hold if there is an extension given by the instructor.

**Group Effort**

This stage of the project is expected to be a group effort, with each member of the group contributing equally in a reasonable fashion.  If it is determined that you are guilty of cheating on the assignment, you could receive a grade of zero with a notice of this offence submitted to the Dean of your home faculty for inclusion in your academic record.

**What to Hand in**

Your submission, as noted above, will be electronically via e-class in the form of a document (e.g. DOC or PDF) file. Please check e-class regularly for any changes related to the submission process. You are to complete the deliverable by:

1. providing the required documentation using the template provided to you in the Project Resources section and
2. the implementation of:
   1. The profile creation
   2. The meal logging functionality
   3. The exercise logging functionality
   4. Correct use of the JFreeChart module and at least one visualization working.
   5. At least one calculation (like the BMR) happening in the background and confirmed with the right test case.
3. The implementation is to be provided as a link to a GitHub repository clearly added in the Document.
4. The project contribution attestation signed by all members OR a letter by each member outlining their own and their team’s contribution, if not all members agree on equal contribution.

Make sure each of your modules has an associated main class to execute each of such module separately, even though modules may not yet work together. For example, the visualizations can be invoked and appear in their own respective windows, or the data can be fetched and printed in the console in a tabular format.

You are encouraged to use a collaboration platform and source code repository like GitHub. If you do so, you can add the URL for the repository at the top of your submitted document (and make sure that the repository can be accessed by the TAs and the instructor). In this case, you do not have to submit your code base as an archive, but only your document as a pdf.

**Assignment Task**

In this assignment, you will craft an initial Software Design Document (SDD) that captures key ideas of the software design of your system that meets the requirements of the use cases presented in the Project Description Document. You can use the SDD Template document provided to you.

Your SDD documentation based on the given template must consist of the following sections and content:

**1.  Main Page**

The main page will contain:

* A title for the project, and a sub-title indicating that this is the requirements documentation for that project.
* The Document Change Control revision history of your document in tabular form.  It should be updated as people add to and edit your requirements documentation.  It should be formatted as follows:

| **Version** | **Date** | **Author(s)** | **Summary of Changes** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

* The table of Contents.

**2.  Introduction**

The Introduction section, should have the following content.

**Purpose**: This sub-section outlines the general objectives of the project.

**Overview**:This sub-section provides an overview of the problem being solved by this software and the requirements of the system.  This should be an executive summary.

**References**: A list of references to other documents that might provide context or otherwise assist in the understanding of this document.

The Introduction section will be similar to the one in your SRS submitted in Assignment 2.

**3. Major Design Decisions**

Text describing significant design choices, and modularization criteria. Modularization criteria include high cohesion and low coupling. Explain how your design aims to achieve these criteria.

**4.  Sequence Diagrams**

One sequence diagram for each of the 8 use cases described in the Project Description document. Make sure that in each diagram you use correct and complete notation capturing the corresponding use case. Each sequence diagram has to be accompanied by a short description explaining it (one paragraph).

**5.  Architecture**

Provide an initial decomposition of you system in terms of modules and interactions. At this stage you can use a block and arrow diagram or a UML component diagram.

Provide a description of each module and its interaction with other modules in the form of the following tables.

|  |  |  |  |
| --- | --- | --- | --- |
| **Modules** | | | |
| **Module Name** | **Description** | **Exposed Interface Names** | **Interface Description** |
| M1 | “a description” | M1:I1  M1:I2 | M1:I1 “a description”  M1:I2 “another description” |
| M2 | “another description” | M2:I3 | M2:I3 “a third description” |

|  |  |  |
| --- | --- | --- |
| **Interfaces** | | |
| **Interface Name** | **Operations** | **Operation Descriptions** |
| M1:I1 | <return type> I1:Op1() used by M2, M4  <return type> I1:Op2(int x) used by M3 and M9 | M1:I1:Op1(): “a description”  M1:I1:Op2(int x): “xxxxx” |
| M1:I2 | <return type> M1:I2:Op3() used by M9 | M1:I2:Op3() used by M1 |

Here we have two modules M1, and M2. M1 exposes two interfaces I1, and I2. Interface I1 has two operations Op1(), and Op2(int x), while I2 has one operation Op3(). Please note that the return type is included in the Operations column. This section will be revised in deliverables 2 & 3.

**6.  Class Diagrams and Initial Implementation**

For this section, you are to create a UML class diagram illustrating the initial design for the modules you implement for deliverable 1. It is expected that these diagrams in your report will be revised, refactored, updated, and enhanced in deliverables 2 & 3. You can use ObjectAid, PlantUML or another class extraction tool to present your class diagrams at the end. Your class diagrams should have:

* Classes
* Attributes, along with their types and visibility
* Methods, along with their parameters, parameter types, return types, and visibility
* Associations, hierarchy and implementation links

You are also to provide a textual description of each of you classes and their methods in the form of a table such as:

|  |  |  |
| --- | --- | --- |
| **Class name** | **Attribute/Method name** | **Description** |
| C1 | attribute-name: attribute-type | “a description” |
| C1 | <return type> method1(param-type, param-name)) | “a description” |

This section will be revised in deliverables 2 & 3.

Your implementation for deliverable 1 will have:

1. The profile creation
2. The meal logging functionality
3. The exercise logging functionality
4. Correct use of the JFreeChart module and at least one visualization working.
5. At least one calculation happening in the background and confirmed with the right test case.

Most likely your code will be refactored in deliverables 2 & 3 to include proper design patterns and conform to better design decisions. Make sure you document your code (JavaDoc would be highly appreciated with a potential bonus!). Make sure your code is well thought of (i.e. uses interfaces, abstract classes etc. as needed and as per the Project Description Document).

**7.  Design Patterns**

For this section, discuss the design patterns you used / planning to use. You are expected to correctly implement by Deliverable 3 at least six design patterns in order to obtain full marks. The intention to implement the six design patterns (including which and how) will have to be documented in the SDD for Deliverable 1. This section of your report will be revised in deliverables 2 & 3. This section of your report would be substantially enhanced in deliverables 2 & 3.

**8.  Activities Plan, Product Backlog, and Sprint Backlog**

In this section provide the following:

1. a complete product backlog list (i.e. what needs to be implemented as a whole). Identify the list if items to be implemented for each deliverable out of the backlog list. You are expected every four weeks to implement, refactor, integrate, and run your code.
2. A GANTT diagram with a schedule of your planned activities

This section of your report may be slightly revised in deliverable 2.

**Group Meeting Logs**

In this Section you write minutes of each meeting, list the meeting attendance, what the topics of discussion in the meeting were, any decisions that were made, and which team members were assigned which tasks. These minutes must be submitted with the project report in each deliverable and will provide input to be used for the overall assessment of the project.

**9. Test Driven Development (TDD)**

At this point you do not have fully implemented your code yet, but hopefully you have some use cases implemented.

In this section, you will provide an initial set of test cases you consider fundamental for verifying your requirements if you had assumed you have the code implemented. Following then the concept of TDD, you write your code so that these tests will pass. Of course, testing assumes that more tests should be written once the code is implemented, but this is a starting point we can use for our course here. You should provide the final and complete list of test cases with your Deliverable 3 (complete implementation).

For this deliverable provide a list of 10-12 test cases in the form of a table as follows:

|  |  |
| --- | --- |
| Test ID | The unique Id of the test case |
| Category | Which part of the system is tested (*e.g. evaluation of parameters to fetch data from the database*) |
| Requirements Coverage | The unique ID of the requirement tested (*e.g. UC1-Successful-Data-Load*) |
| Initial Condition | Initial conditions required for the test case to run (*e.g. the system has been initiated and runs*) |
| Procedure | The list of steps required for this test case (*e.g.*  *1. The user provides a meal*  *2. The user provides a date*  *3. The user provides a list of ingredients and quantities*  *4. The user clicks on the log meal button* |
| **Expected Outcome** | The expected outcome of the test case (*e.g. The nutrient data is fetched from the database and appears in a table*) |
| **Notes** | Any other notes you may want to add for this test case, which are also reflected in the requirements specification (*e.g. the user should provide valid input*) |

This part corresponds to section 8 of the template provided.

From the point you complete and submit this deliverable you will keep on working on your implementation and complete as required the list of any additional test cases you want to include. That is this section of your report will be revised in deliverables 2 & 3.