

# **BMIT3173 Integrative Programming**

## **ASSIGNMENT 202505**

Student Name :

Student ID :

Programme :

Tutorial Group :

System Title :

Modules :

### **Declaration**

- I confirm that I have read and complied with all the terms and conditions of Tunku Abdul Rahman University of Management and technology's plagiarism policy.
- I declare that this assignment is free from all forms of plagiarism and for all intents and purposes is my own properly derived work.

## Plagiarism Statement Form

I, Name (Block Capitals) \_\_\_\_\_ Student ID \_\_\_\_\_ Programme  
\_\_\_\_\_ Tutorial Group \_\_\_\_\_ confirm that the submitted work are all my own  
work and is in my own words.

I (Student Name) acknowledge the use of AI generative technology.

Signature: \_\_\_\_\_

Date: \_\_\_\_\_

## Table of Contents

1. Introduction to the System	4
2. Module Description	4
3. Entity Classes	4
4. Design Pattern	4
4.1    Description of Design Pattern	4
4.2    Implementation of Design Pattern	4
5. Software Security	4
5.1    Potential Threat/Attack	4
5.2    Secure Coding Practice	4
6. Web Services	5
7. Index	6

## 1. Introduction to the System

Instruction: Provide a clear overview of your system and the specific / typical organization(s) which will use it.

## 2. Module Description

Instruction:

- Provide an easy-to-understand description of the complete modules that you oversaw.
- Separate the modules by the functions you handle.
- Use **figures from your system (print screen)** to describe each function.

## 3. Entity Classes

Instructions:

- a) Include the entity class diagram of your system.
  - Remember that entity class diagram  $\neq$  entity-relationship diagram (ERD)!
- b) For the entity class(es) that you oversaw:
  - Remember to **use object references** instead of foreign keys to represent relationships between classes. Ensure that your entity classes' implementations are consistent with the entity class diagram.

## 4. Design Pattern

Instruction: Each member in a team needs to implement a unique design pattern

- Singleton and MVC design patterns are not counted as one of the chosen design patterns.
- Each design pattern must be unique and cannot be duplicated by other team members.

### 4.1 Description of Design Pattern

- Describe the design pattern that you implemented.

### 4.2 Implementation of Design Pattern

Instruction:

- Draw a class diagram to show how your design pattern was implemented.
- Provide justification for your choice of design pattern.

Note: You should organize your code according to the Model-View-Controller (MVC) architecture, but this should not be the design pattern that you write about here.

## 5. Software Security

Instruction: Identify potential threats or attacks within your module and outline secure coding practices that can be implemented to mitigate these risks.

### 5.1 Potential Threat/Attack

- Each student is required to identify **TWO (2) potential threats or attacks** that could affect the module they are handling. Provide a detailed description of the **types of attacks**.
- 

### 5.2 Secure Coding Practice

- For each identified threat or attack, describe the **secure coding practices** you have

implemented to prevent or mitigate the attack.

- Each secure coding practice must be unique and cannot be duplicated by other team members.
- **Input validation** is mandatory for all modules; however, it should not be counted as one of the strategies for the attacks you mention.

Include relevant **figures** (such as screenshots), **code snippets**, and **descriptions** to support your explanation.

## 6. Web Services

Instruction:

You are required to implement both the exposure and consumption of web services, adhering to the **Interface Agreement (IFA)** standards. Your module must interact with other modules using web services.

### 1. Service Exposure

Your module must expose a web service for consumption by other modules. This web service should be either:

- **REST API (JSON-based)** – This is the recommended choice for modern, lightweight, and flexible services.
- **SOAP Web Service (XML based)** – A more structured and legacy approach, should be used if specifically required or requested.

### 2. Service Consumption

Your module must also consume a web service provided by another team member, ensuring seamless integration with other system components and enabling efficient data exchange between modules

### 3. The format for the Interface Agreement (IFA) is as below: -

#### Webservice Mechanism

	Description
Protocol	SOAP / RESTFUL
Function Description	<i>Example: Retrieves user information by user ID</i>
Source Module	[your module] <i>Example: User Management</i>
Target Module	[who can consume your service] <i>Example: Customer Service, Analytics Module</i>
URL	[URL that the web service can be called] <i>Example: http://example.com/api/getUserInfo</i>
Function Name	[Function name of your service] <i>Example: getUserInfo</i>

**Web Services Request Parameter (provide)**

Example:

Field Name	Field Type	Mandatory/Optional	Description	Format
userId	String	Mandatory	Unique ID of the user	Can only contain alphabet and number.
queryFlag	Integer	Mandatory	Flag on information needed	1: get Contact No. 2: get Address 3: get both contact number & address

**Web Services Response Parameter (consume)**

Example:

Field Name	Field Type	Mandatory/Optional	Description	Format
status	String	Mandatory	Status of the request	A: Active I: Inactive
userName	String	Mandatory	Name of the user	Can only contain alphabet and number.
userEmail	String	Optional	Email of the user	Only acceptable email address
userDetails	Object	Mandatory	Detailed information about the user	Contains the details that are needed from the query  HpNo: Handphone No. OffNo: Office No. HouseAdd: House Address OffAdd: Office Address

Instruction:

- Give an overview of how web service technology was used in your module.
- IFA for the request and response for the web service that you have prepared.
- Include clear code snippets and descriptions.

**7. Index**

- Create an index file that includes all your figures/printscreens along with the source paths for each screen.

**Note: While documentation constitutes 15% of the overall assignment grade, the clarity and depth of your explanations will play a crucial role in your performance across other tasks. It is your responsibility to ensure that the report includes all required deliverables before submission.**