UNIVERSITI TUNKU ABDUL RAHMAN

JANUARY TRIMESTER FINAL ASSESSMENT

**UECS2344 SOFTWARE DESIGN**

BACHELOR OF SCIENCE (HONOURS) SOFTWARE ENGINEERING

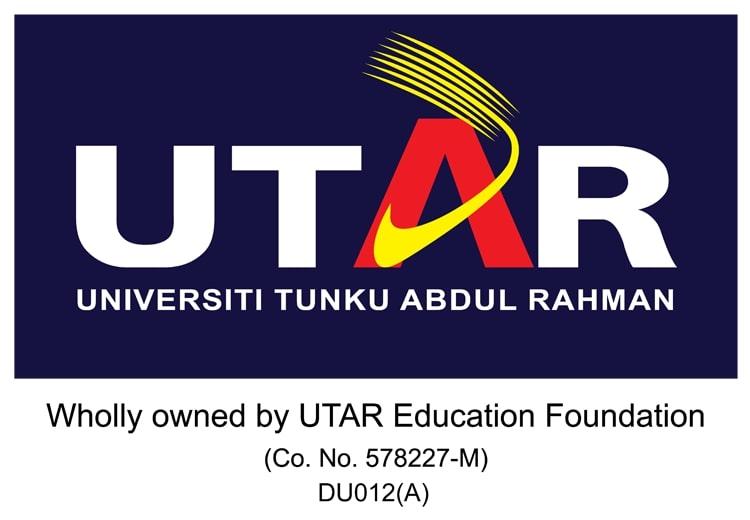
**ANSWER SHEETS**

|  |  |
| --- | --- |
| Name (as stated in Student Card) | Student ID |
| TAN YING YAO | 1703648 |
| Index No | Faculty/Centre |
| U00339EBSEF | LKC FES |
| Programme | Submission Date |
| SOFTWARE ENGINEERING | 18 MAY 2020 |

*For Examiner’s Use Only*

|  |  |
| --- | --- |
| **QUESTION NUMBER** | **MARK** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **TOTAL** |  |
| **TOTAL (100%)** |  |

**Final Assessment Declaration Statement**



**DECLARATION**

I, TAN YING YAO (Name), Student ID. 1703648

hereby solemnly and fully declare and confirm that during my programme of study at Universiti Tunku Abdul Rahman, I shall abide and comply with all the rules, regulations and lawful instructions of Universiti Tunku Abdul Rahman and endeavour at all times to uphold the good name of the University.

I hereby declare that my submission for this Final Assessment is based on my original work, except for citations and quotations which have been duly acknowledged. I am fully aware that students who are suspected of violating this pledge are liable to be referred to the Disciplinary Committee of the University.

Programme: BACHELOR OF SCIENCE (HONOURS) SOFTWARE ENGINEERING

(Digital) Signature: TAN YING YAO

Student’s I.C. No: 971210-14-5673

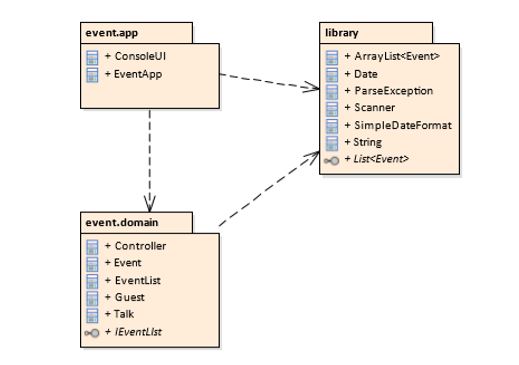
Date of Submission: 18 MAY 2020

**Question 1**

I have created the application of E-Commerce Training Management System as part of the coursework requirements. The application that has been written was design with a structured programming approach with each function acting as an independent module. Every module performs a specific function and helps manage the complexity of an application by increasing its legibility.

Our event management system was designed specifically for high cohesion and low coupling. Cohesion is the degree of relatedness of elements within a module while coupling refers to the degree of relatedness between modules. Each of our modules have a singular and specific purpose which results in a high degree of relatedness of all the elements within the module. Each line of the function is specifically written for the purpose of said function and contains zero irrelevant code which helps improve the development and testing phase while increasing the maintainability of the application.

Our module is also extremely independent and loosely coupled to other modules which causes the absence in unnecessary dependencies on other modules which reduces the degree of relatedness to other modules. Our application is highly independent and have low coupling to other functions which also results in maintainability and reusability as any changes made won’t affect other functions.



**Question 2**

There are many advantages of using Object-Oriented Systems Development with UML modelling approach such as the effective communication of the system. It is the perfect visual language to communicate detailed information about the system. UML is also highly flexible and widely recognized which results in specific customization to modelling elements and interactions in a diagram. Class Diagram is used to display the relationships between classes modelled in system. Use-Case diagram displays the business requirements and interaction of the system and its environment. Package Diagram groups other UML elements to form higher-level constructs. Sequence diagrams showcases the steps taken for the system to execute.

This is extremely helpful during our system development as it is a visual language that let us model the design of the system architecture. It also helps us in modelling the business process, showcase structure, sketching ideas and generating the code. The approach also enables us to describe the architecture, capture system behaviour, model data structure and detailed specification of system. Thus, it allows us to build an effective and efficient system development process.

**Question 3**

**Bold: Primary Key, Underline: Foreign Key**

Event (**EID**, Talk Title, Date, Venue, Theme, TID, GID)

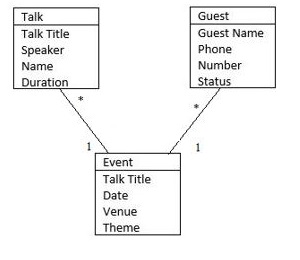
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| EID | Talk Title | Date | Venue | Theme | TID | GID |
| 1 | Physics of Marketing | 23/2/2010 | KB101 | Education | T1 | G1 |
| 2 | Consumer Irrationality | 26/5/2011 | KB201 | Inspiration | T2 | G2 |
| 3 | Brand Control | 02/7/2012 | KB301 | Education | T3 | G3 |
| 4 | Choice Selection | 17/11/2013 | KB401 | Guide | T4 | G4 |
| 5 | Science of Retail | 13/9/2014 | KB501 | Guide | T5 | G5 |

Talk (**TID**, Talk Title, Speaker Name, Duration)

|  |  |  |  |
| --- | --- | --- | --- |
| TID | Talk Title | Speaker Name | Duration |
| T1 | Physics of Marketing | Bob Tucker | 30 Minutes |
| T2 | Consumer Irrationality | Alan Turing | 90 Minutes |
| T3 | Brand Control | Pagliacci | 60 Minutes |
| T4 | Choice Selection | Isaac Newton | 20 Minutes |
| T5 | Science of Retail | Tan Ah Gao | 10 Minutes |

Guest (**GID**, Guest Name, Phone Number, Status)

|  |  |  |  |
| --- | --- | --- | --- |
| GID | Guest Name | Phone Number | Status |
| G1 | Jon Connington | 1300887676 | Accepted |
| G2 | Edric Elba | 1300889999 | Accepted |
| G3 | Bobby Tucker | 1300880069 | Accepted |
| G4 | William Hardeep | 1300889090 | Accepted |
| G5 | Sum Tim Wong | 1300882020 | Accepted |



**Question 4**

Façade pattern and Template Method pattern could be used effectively for the event management system. Façade pattern is designed to help a client class interact with a set of classes and shields the complexities of the set of classes by providing a single class with simplified methods required by the client. This can work well with event management system as the pattern provides a single subsystem interface to these classes. Our design of the system matches some of the description of this pattern as we have a subsystem of classes that contains different function. The client can access the UI to choose the desired function and execute them.

Template Method pattern creates a method which defers the steps of the implementation to the subclasses. The method defines the steps to execution while the subclasses contains the detailed implementation of the steps of the algorithm. Template method consist of certain fixed order and implementation is unique for different subclasses. The superclass calls methods from the subclasses and is widely known as the Hollywood Principle. Methods in base class without implementation are referred as Hooks and are overridden by subclasses Our design of the system contains different function with the execution but the implementation is in the classes.

**Question 5**

Layered Architectural Pattern and Model-View-Controller (MVC) Architectural Pattern can best used for the deployment of the system. Layered Architectural Pattern organises the system into multiple layers containing specific functions. The lowest layer typically contains core services while the top layers are for clients. This allows replacement of entire layer as long as the communication between them are unchanged. Our systems top layer contains the UI while the functions lays at the bottom layer which makes it a two-layered (two-tires) architectural pattern.

|  |
| --- |
| User Interface |

|  |
| --- |
| Application Functionality |

Model-View-Controller (MVC) Architectural pattern separates the presentation and interaction from the system data. The system is typically structured into 3 logical elements with the model, view and controller. This allow the structure of the data to change independently of its presentation and supports presentation of the same data in different ways with changes made in one view updating in others. Our system already has a controller and the view is combined with the model, using this pattern could increase our code complexity while keeping our view model simple. Example of the MCV pattern in our system implemented:

