Question 1

1. Bespoke software product.

* This software (online teaching and learning system) is developed for specific users which are TARC students and academics and its copyright belongs to TARC.
* The requirements specification is controlled by the users (students and academics) which user requirements will be gathered through an interview.

1. Prototype

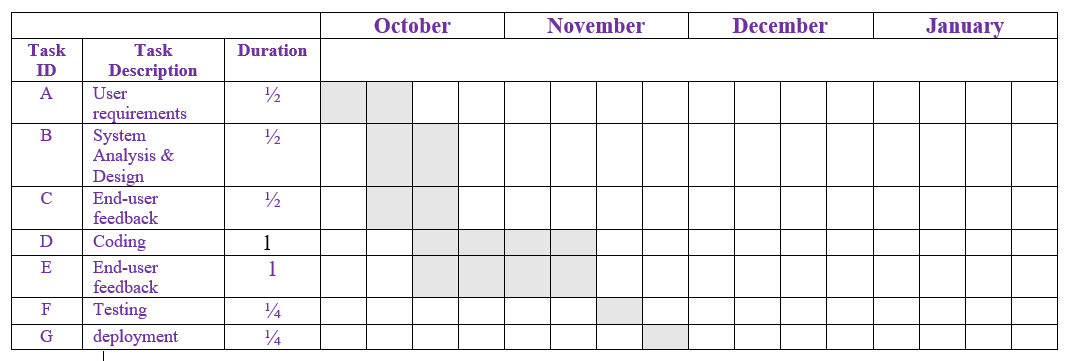
* The user requirements are not clear
* Prototyping allow changes happen in which the user requirements from students and academics could change frequently for the new online teaching and learning system
* Low budget and the new system is not complex which prototyping model only consume low cost
* Prototyping require less staff which appropriate to this scenario: only one junior software engineer and one intern involved in the software development
* **Availability** - The new online teaching and learning system should work as required and available 24 hours 7 days which is accessible for academics and students regardless of the location, technology used without downtime.
* **Reliability** - The new online teaching and learning system should be able to sustain and function as it is expected. It should produce and display the correct information such as class schedules, semester result, attendance tracking to the user. On top of that, it should be well-functioning in unexpected environments such as server overload.
* **Usable**- The new online teaching and learning system provides students and academics with a satisfying experience which is easy to use with a good and consistent user interface design, understandable work process, and effective functionality



* **Physiological** – We should provide good employment or comfortable working conditions for interns such as Wi-Fi access to ease the internet connection or tea/coffee making facilities
* **Safety** – We should provide long term employment by converting the intern into full time employees/staff.

Question 2

1. -



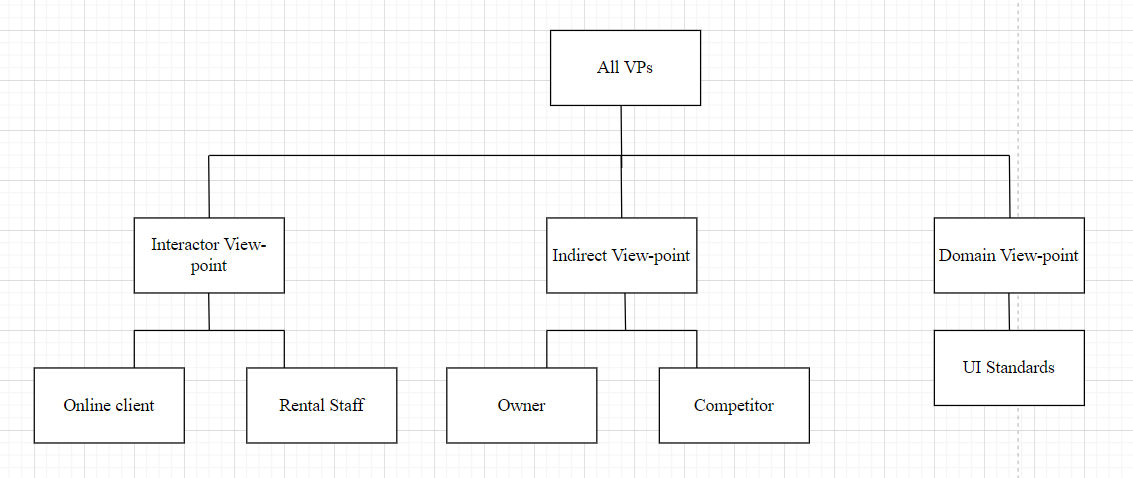
1. **5 Functional requirements**

* The system should provide an online login ID with a password for user to log into system
* User can select the types of vessel and make a booking for the available vessel
* The system should calculate the rental fee once user confirm the transaction
* The system will check on the payment details such as credit card number, CCV, expiry date and etc
* The system will generate online receipt to the user

**2 non functional requirement**

* **Efficiency** - The system shall verify the payment within 1 minute
* **Security** - The system shall secure customer’s credit card information

1. -



Question 3

1. **Black box testing**. It is a testing technique in which the system functionality will be tested without knowing the internal code implementation and mostly determined by studying the system input as well as the output.

| Program name: CBB online ERP system  Test date: - | | | | | |
| --- | --- | --- | --- | --- | --- |
| No. | Test case | Test data | Expected Result | Actual Results | Remarks |
| 1 | To test the customer support services by starting online chat with the customer service staff | “I have question regarding my transportation services” | Query message appear in the customer support representative window | - | - |

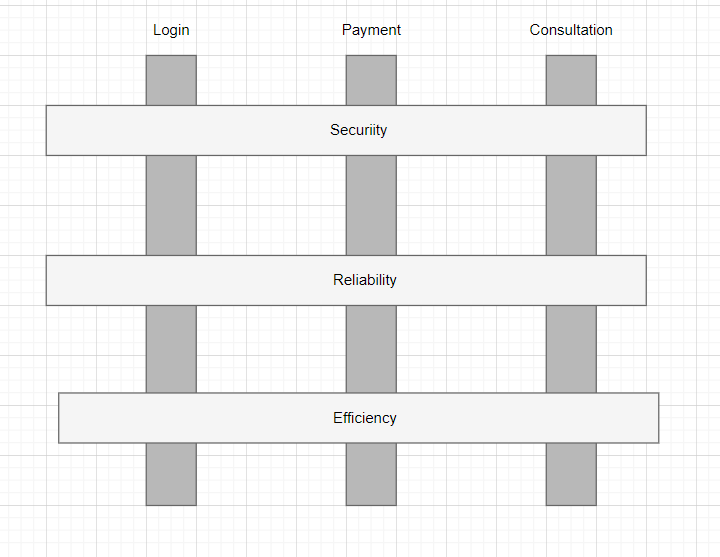
**Stress/Performance testing**. It is a testing technique that tests the reliability of the system to ensure it can support the intended load by stressing the system

| Program name: CBB online ERP system  Test date: - | | | | | |
| --- | --- | --- | --- | --- | --- |
| No. | Test case | Test data | Expected Result | Actual Results | Remarks |
| 1 | To test whether system able to support intended load when there are many delivery services execute simultaneously | Delivery details:  Customer name  Product  Delivery address  Contact number | The system works as required which process all the delivery process and services without facing any downtime | - | - |

* **Consistency** - The interface in the system should be consistent all the way to avoid any user confusion. Moreover, the similar operation should be carry out in the same way
* **User guidance** - The system should provide help features whenever user are in doubt or facing some problem in booking vessel, checking booking progress, make payment and etc

1. **Web development tools** - E.g. *Visual Studio* to write and edit the source code to develop the particular web pages like forms, graphic and etc

**Data modeling tools** - E.g *Database management system (DBMS)* to manage and store the related data in the database such as payment, product, order, order details, customer and etc



Question 4

1. (i)

**Aperiodic stimuli response**

Stimuli (detect low battery) -> Response (Move back to the docking section to recharge)

Stimuli (detect obstacles) -> Response (Change the move direction)

Stimuli (detect elderly want to open door) -> Response (move towards the door and open it)

**Periodic stimuli response**

Every 1 minutes, check the battery level

Every

(ii)

| Worst cohesion: **Coincidental cohesion** | Best cohesion: **Functional cohesion** |
| --- | --- |
| This cohesion made from parts of a module that are grouped arbitrarily  E.g. “Utilities class” | This cohesion describe the elements of the module are grouped with a well-defined purpose and single task |

Appropriate cohesion: ***Functional cohesion*** because in this cohesion, every single task is well defined and related to each other within a particular module.

1. Factors affecting the maintenance cost

| **Characteristic of the legacy system** | **Maintenance cost** | **Comments** |
| --- | --- | --- |
| High dependency between one module to another | High | When a module is changed, other modules which are dependent on it will be affected |
| Developed by low-level programming language | High | Low-level programming language are machine dependent, not portable and hard to debug, maintain |
| tested for 3 months before deployed | Low | System is tested for long time means it is thoroughly tested with appropriate testing strategies and techniques |
| All versions are stored in a configuration management system | Low | All versions are trackable easily through the configuration management system |
| developed in 1990 | High | The older age a system, the higher maintenance cost |

(ii)

* High business value but low system quality
* The legacy system should be **re-engineered** because it has high business value for their well growing business that brings significant profit to the company. However, the legacy system has obsolete programming language (low level programming language), weak program structure and modularization and it’s a old system thus need to be re engineered so that can improve the system structure and create new documentation to easier maintenance in the future and make the code easier to understand