1. **ESSAY (50%)**

**Note: all essay questions should be answered within 300 until 500 words.**

1. **(Weight 10%)** When considering the design of an object-oriented system, a set of criteria exists, that can be used to determine whether the design is a good one or a bad one. A good design is one that balances trade-offs to minimize the total cost of the system over its entire lifetime. **Describe the following design criteria including coupling, cohesion, and connascence!**

**Answer (500 Words):**

1. **Coupling**

* Coupling refers to how interdependent or interrelated the modules (classes, objects , and methods) are in a system.
* The Higher the interdependency, the more likely changes in part of a design can cause changes to be required in other parts of the design.
* **Types of Interaction Coupling :**

|  |  |  |
| --- | --- | --- |
| LEVEL | TYPE | DESCRIPTION |
| GOOD | No Direct Coupling | The methods do not relate to one another; that is, they do not call one another |
|  | Data | The calling method passes a variable to the called method. If the variable is composite, the entire object is used by the called method to perform its function. |
|  | Stamp | The calling method passes a composite variable to the called method, but the called method only uses a portion of the object to perform its function. |
|  | Control | The calling method passes a control variable whose value will control the execution of the called method. |
|  | Common or Global | The methods refer to a “global data area” that is outside the individual objects. |
| BAD | Content or Pathological | A method of one object refers to the inside (hidden parts) of another object. |

1. **Cohesion**

* Cohesion refers to how single-minded a module (class, object, or method) is within a system.
* A class or object should represent only one thing, and a method should solve only a single task.
* **Types of Method Cohesion :**

|  |  |  |
| --- | --- | --- |
| LEVEL | TYPE | DESCRIPTION |
| GOOD | Functional | A method performs a single problem-related task. |
|  | Sequential | Combines two functions in which **the output from the first one is used as the input to the second one**. |
|  | Communicational | Combines two functions that **use the same attributes to execute**. |
|  | Procedural | Supports multiple weakly related functions. |
|  | Temporal or Classical | Supports multiple related functions in time |
|  | Logical | The choice of the specific function is chosen based on a control variable that is passed into the method. |
| BAD | Coincidental | Cannot be defined or it performs multiple functions that are unrelated to one another. |

1. **Connascence**

* From an object-oriented design perspective, Connascence means that two modules (classes or methods) are so intertwined that **if you make a change in one, it is likely that a change in the other will be required**.
* A subclass should **never** directly access any hidden attribute or method of a superclass. A subclass shouldn’t have ability to the properties of its superclass.
* **Types of Connascence :**

|  |  |
| --- | --- |
| TYPE | DESCRIPTION |
| Name | If a method refers to an attribute, it is tied to the name of the attribute. |
| Type or Class | If a class has an attribute of type A, it is tied to the type of the attribute. |
| Convention | A class has an attribute in which a range of values has a semantic meaning. |
| Algorithm | Two different methods of a class are dependent on the same algorithm to execute correctly. Insert and find an element in the same array. |
| Position | The order of the code in a method or the order of the arguments to a method is critical for the method to execute correctly. |

1. **(Weight 10%)** Applications are of little use without the data that they support. One of the leading complaints by end users is that the final system is too slow, so to avoid such complaints project team members must allow time during design to carefully make sure that the file or database performs as fast as possible. The design of the data management layer addresses these concerns. It includes both the design of data access and manipulation classes and the actual data storage. The design of the data access and manipulation classes should ensure the independence of the problem domain classes from the data storage format. **Name and describe, at least 3 out of 5 choices about the different types of object persistence formats!**

**Answer (313 Words):**

1. **Object-Relational Database**

* Prolonged for managing the objects to preserve. It is generally performed with employment of user defined data types.
* Inheritance favors the language reliant. Often too fine assistance for normal data management manipulations which trade may await by utilizing it along with the simple utilization of SQL.

**Strength:**

* Relational Database extended to handle storage of objects
* Use of user-defined data types
* Extended SQL to handle complex object data
* Inheritance tends to be language dependent

**Weakness:**

* Limited support for object orientation
* Impedance mismatch between tables and objects
* Inheritance tends to be language dependent

1. **Object-Oriented Database**

* There are two major implementations for assisting the object persistence inside the object -oriented database section. First, including the persistence extensions for the object oriented programming language.
* Second, creating the overall database alone. Often assist where the data is critical and few assistance to inheritance. It can offer the straight assistance for orientation of an object.

**Strength:**

* Standard from object data management group
* Support complex data
* Object ID assigned
* Some support for inheritance
* Sharper learning curve

**Weakness:**

* Technology is still maturing
* Skills are hard to find
* Bad at manipulating deep analysis

1. **Sequential Access**

* An electronic record of details which are needed to format a specific transactions. Sequential (permits the sequential file manipulations since data in order as per the identifier).

**Strength:**

* Optimized for operations on entire file.
* Access to specific objects not efficient

**Weakness:**

* Inefficient for searching
* Additions & deletions of files require additional maintenance
* No control access

1. **Random Access**

* An electronic record of details which are needed to format a specific transactions. Random files (random files duration of response is quick. Files are preserved the data as per the particular formula or the order of particular identifier).

**Strength:**

* Optimized for finding specific
* Entire file operations not efficient

**Weakness:**

* Inefficient for report writing
* No control access
* Data redundancy

1. **(Weight 10%)** From a user interface design perspective, mobile application is both exciting and challenging. However, just because these mobile devices have the ability to surf the web, does not mean that a simple web interface is the answer. These devices have limited screen space and have capabilities, such as touch screens and haptic feedback (such as vibration or pulses), which regular computers do not. **Describe the unique issues related to designing user interfaces for mobile computing platforms!**

**Answer (388 Words):**

Unique issues related to designing user interfaces for mobile computing platforms:

1. **User Interface Processors**

* A user interface processor (or a “disconnected-rarely terminal”) **only performs drawing and low-level event processing**, leaving **the rest of an application’s computation to a remote server**.
* **Placing most of the computational power on the mobile device seems to be overkill if this is how the machines are actually used**. In designing a device that has stringent constraints on price, power consumption, and size, one must omit components that are not absolutely necessary.
* Running applications on a remote host and using the mobile device primarily for the user interface solves some of the hard problems faced by designers of mobile computer operating systems.
* **First, the system re-initialization problem becomes less complicated**. When the mobile device crashes, it can be rebooted and the user can continue the application with no loss of data or state. This assumes that the remote hosts are more reliable than the mobile devices.
* **The second major problem** that the user interface processor approach solves is the **data consistency problem**.

1. **Mobile Tasks**

* An analysis of mobile computing tasks may produce a vastly different set of application priorities than an analysis of the use of a typical office workstation. **It will certainly produce a different set of tasks**.
* For example, **an analysis may indicate that the primary tasks to be performed on mobile devices are**: scheduling, short note taking, email reading and composition, large database browsing, and filling out forms that must be incorporated into large home office databases.
* **Compare this set of tasks with the typical tasks performed on office workstations by researchers**: large document preparation and layout, programming, email reading and composition, and the creation of graphics presentations.
* Most of the workstation tasks involve a large amount of data entry, whereas **the mobile tasks mainly involve small amounts of data entry** and the presentation of existing information.

1. **Mobile Device Sizes**

* **The primary reasons for this are the small size of the displays and the wide variety of input devices used** on mobile computers.
* **For example, the small amount of screen real estate on the Mobile Devices**, about, **would tend to favor interfaces that have few static interface** elements occupying space, such as scroll bars, palettes, and icons.

1. **(Weight 10%)** Testing is more critical to object-oriented systems than to systems developed in the past. Based on encapsulation (and information hiding), polymorphism (and dynamic binding), inheritance, reuse, and the actual object-oriented products, thorough testing is much more difficult and critical. **Mention the 4 steps of software testing, including: Unit Testing, Integration Testing, System Testing, and Acceptance Test.**

**Answer (400 Words):**

1. **Unit Testing**

* Unit testing is a level software testing where **individual units/components of a software are tested**.
* The purpose is **to validate that each unit of the software performs as designed**.
* Unit Testing **is the first level of software testing and is performed prior to Integration testing**.
* It is normally **performed by software developers themselves or their peers**. In rare cases, it may also be performed by independent software testers.

1. **Integration Testing**

* Integration testing is a software testing methodology used **to test individual software components or units of code to verify interaction between various software components and detect interface defects**. Components are tested as a single group or organized in an iterative manner.
* Integration is a key Software Development Life Cycle (SDLC) strategy. **Small software systems are integrated and tested in a single phase**, whereas **larger systems involve several integration phases to build a complete system**, such as integrating modules into low- level subsystems for integration with larger subsystems.

1. **System Testing**

* **System testing is a phase in the software testing cycle where a total and integrated application /system is tested**. The focus of the system testing is to evaluate the compliance of the entire system with respect to the specified requirements.
* System testing **helps in approving and checking the business, functional, technical, and any non-functional requirements of the application** concerning the architecture as a whole.
* **The scope of system testing is not only limited to the design of the system but also to the behavior and believed expectations of the business**. In accordance with the software test cycle, **system testing is performed before acceptance testing and after integration testing**. Independent users or testers are given the tasks to perform in the system testing phase.

1. **Acceptance Testing**

* Acceptance Testing involves the **implementation of a software component or system component to evaluate one or extra properties of Interest**.
* **Acceptance Testing Include :**

1. Meets the requirements that guided its design and development.
2. Responds correctly to all kinds of inputs and give Correct and optimal Output.
3. Performs its functions within an acceptable time, is sufficiently usable, can be installed and run in its intended environments.
4. Include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use.
5. **(Weight 10%)** What are the obstacles for provisioning the physical architecture layer with cloud technologies? What, if any, are the issues related to security in the cloud computing environment?

**Answer (463 Words):**

**One most important obstacles are Operational Consistency** in which when operations are required to manage local resources through one methodology and cloud deployed using another resources. **These kind of inconsistency increase the cost of managing the implementation and operational risk**. Performance and availability can be compromised by inconsistent but monitoring and visibility gets damaged due to lack of functionality in cloud environments.

**The other obstacles with cloud technologies are service**. Enabling the infrastructure is necessary to automate and process respectively. It has become challenges due to lack of standardization within the infrastructure origin while many organization have standardized on XML and web services. **This makes difficult to find people with right mix of development skills to service-enable the infrastructure**. This challenges can be mitigated by making multiple API calls with each infrastructure component.

**Next obstacles are Inter-Cloud Architecture** in which designing a private-cloud with the transition to a hybrid architecture can be discourage. **This requirement include secure connection between the private and public cloud environments**.

**Another obstacles with cloud technologies are making of deployment policies** to be operational consistent and to have reusable coding, to overcome of this obstacles, **the public cloud resources are integrated with infrastructure systems**, it helps in cost reduction, and increases elasticity and flexibility.

**Other obstacles are Maintaining of Compliance**. **It is difficult to maintain Compliance across distributed workloads**, by using the traditional security policies like firewall, Key management etc. the compliance can be maintained in public cloud environments.

**Issues related to security in the cloud computing environment:**

1. **Denial of Service (DoS) Attacks**

* A cloud service often host many customers. **Any attack against a customer affect other customers sharing the same cloud provider networks**. So, this is a concern in the Security of cloud computing.

1. **Domain Name Servers (DNS) Attacks**

* DNS attacks are increasing in popularity. **If a cloud provider is prepared and successfully migrates a traditional denial of service attack, a DNS attacks simply goes after the Domain Name Servers** (DNS)

1. **Application Attacks**

* An attacks could find search feature on a website, and submit a query for a file keyword that doesn't exist, **producing a "NOTHING FOUND" Error**. Let's say this attack is sending 3000 of these searches. **It will hit the database running a script 3000 times**. This will decrease the performance.

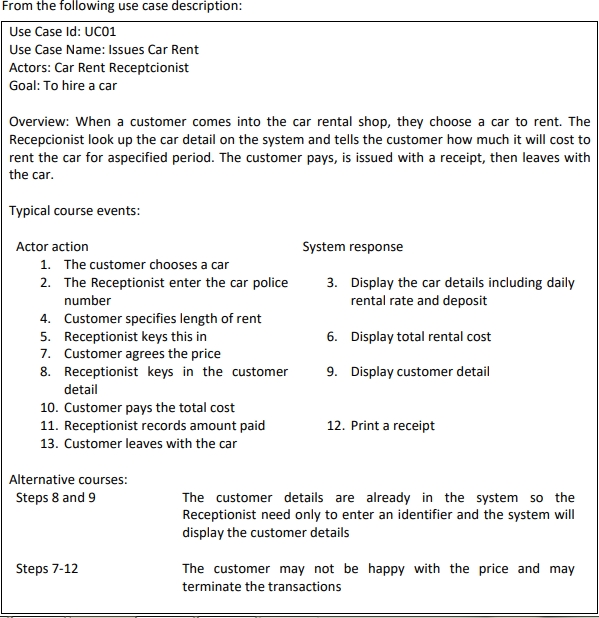
1. **Bot Network Attacks**

* Bot networks "zombies" will occur when a computer within a networks has been compromised through a virus like "Worm" or "Trojan". **It may create traffic in cloud-service**.

1. **Perimeter security of the networks (Demilitarized zones) Attacks**

* Protecting the networks infrastructure using firewalls is the most common past of securing your systems. **These firewalls may not be strong enough that will be created "Demilitarized Zones" (DMZ**) where public facing web services can be offered.

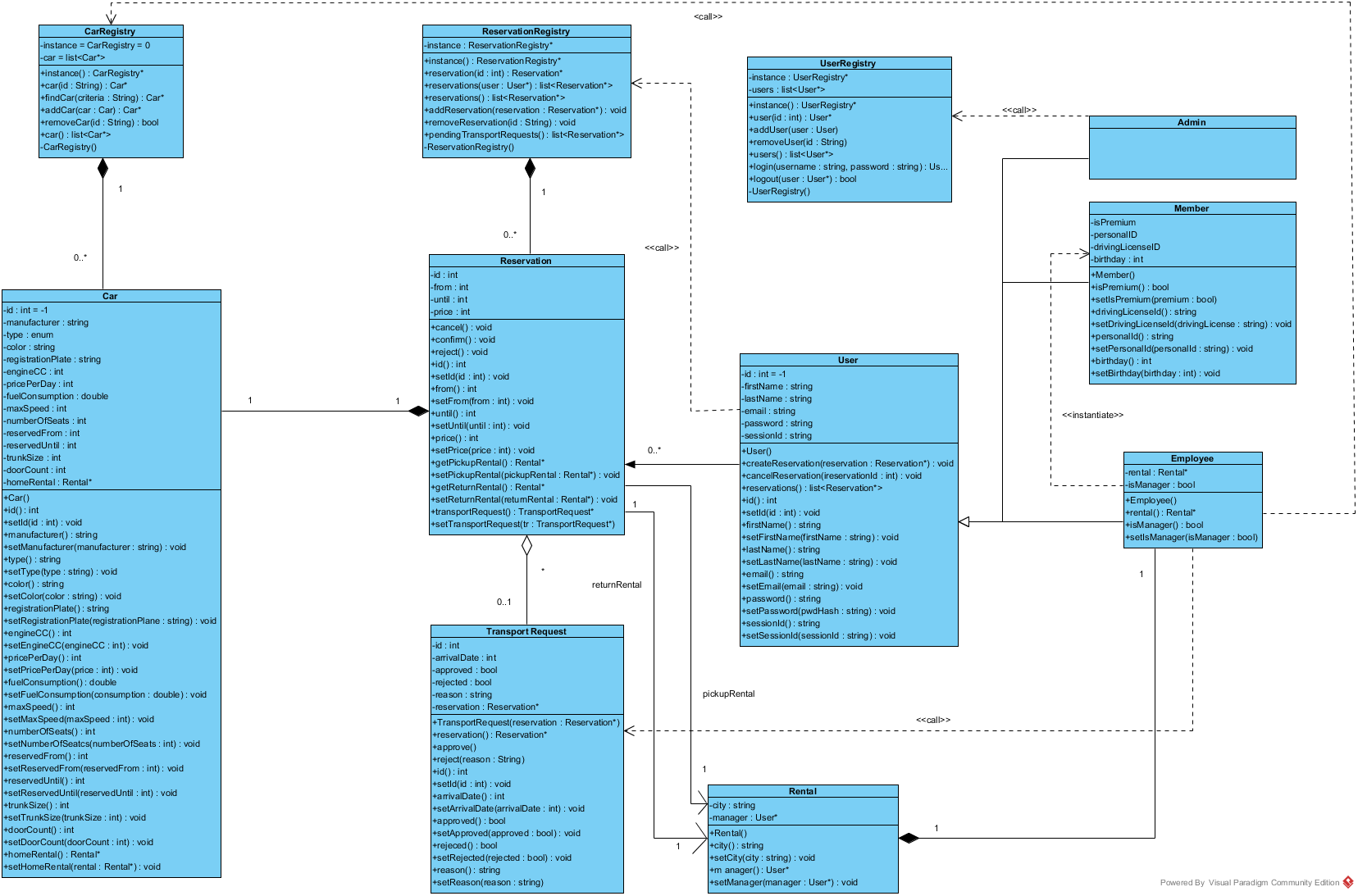
1. **CASE (50%)**



From the use case description above (simple two columns format, with alternative courses), you are asked to draw the static aspect of modeling:

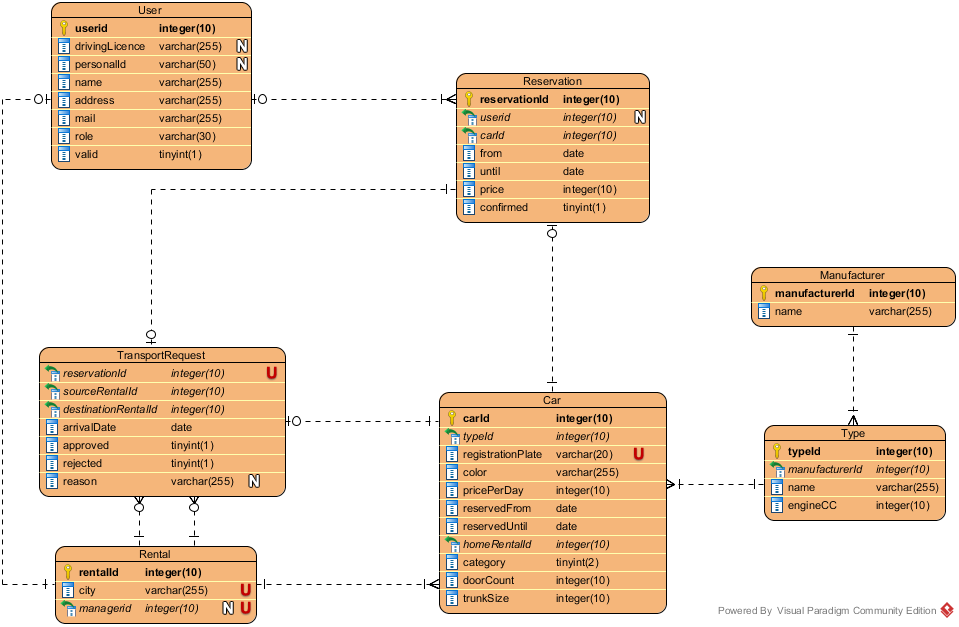
1. **(Weight 20%)** The **class diagram for this scope of this module**, draw the object classes, do not use “system” as class!!!

**Answer (Zoom to 200% to get Clear View of The Diagram):**



1. **(Weight 15%)** Map your **class diagram into relation model diagram**

**Answer (Zoom to 130% to get clear view of the diagram):**



1. **(Weight 15%)** Assume that the car rental is the integrated system and has some branch offices in the country, **draw the deployment diagram** to show the need for infrastructure of the system.

**Answer (Deployment Diagram):**

