**OOAD Home Work 03**

**PART 1**

**Mxx150430**

**Pros:**

1. The Requirements clear and specific

2. Object in domain model is well defined

3. The architecture design diagram well defined MVC architecture

4. Full dressed use cases are well designed and applied GRASP

**Cons:**

1. For the requirements, No.1 should also allow managers to register for the website based on it’s further system design. The requirements is missing that the system should handle the cars record for repairing and maintenance.

2. For the domain model, the relationship between Invoice and Reservation should be multi to one because one reservation could include not only one cars and rented by more than one aspect of customers. The relationship between Manager and Staff should be multi and multi. The object Staff conflicts with object Manger because manager could also be an employee, the object Staff should alter its name. Also the model is missing the design for the car record track which is mentioned in the system requirement.

1. For the use case diagram, the actor Manager conflicts with actor Staff which is mentioned in the previous Cons. Also, missing use case for managing the car’s records.
2. For the fully dressed use case, each success scenario is missing clearly definition of the actor’s action. Each scenario should show generalization for actors.

**Hxy153530**

**Pros:**

1. The architecture of the system is well declared in MVC architecture.

2. Full dressed use case are done well and GRASP was applied.

3. Design sequence diagram is well designed, all aspects of the requirement was fulfilled.

**Cons:**

1. For the requirement , it should not only simply mark from the original requirement paragraph, but also should make a analysis and derive more former and more detailed requirement from it.
2. The definition between requirement and constrain is not clear, for example the constrain c should be derived into an requirement but not an constrain.
3. For the domain model, across each two objects which are related should be marked the numerical relations.
4. For the domain model, most of the object is missing attributes. The design or the domain model is not clear because the missing of the relations and attributes.

5. For the domain model, the object Vehicle and the object PassengerCar should be redesigned, because based on the requirement, the company currently only has passenger cars to rent.

6. The literal architecture design is vague, it is mainly derived from the original definition of the MVC architecture, there is not enough specific detail concentrating on the real definition of the actual system.

7. For the system architecture design, although the architecture is designed literally, the architecture diagram is missing.

1. For the use case diagram, the design of some use case are not necessary or inappropriate, for example, Choose Return Location, Find A Car On the Website or Phone, Check Rental Car’s Information should be combined as View Car information.
2. For the use case diagram, some use cases should be share by all the actors and some functionality which described in the original requirement is missing.
3. For the design sequence diagram, most of the diagram is missing an input or a controller.

**Part 2 Compare the two works with each other.**

For the work from mxx, most part of the work are well designed the only flaw is all the system design is missing to keep track of the car record.

For the work from hxy, the work is not as well designed as the mxx’s work, but for the last DCD part, hxy’s work is better.

Although these two works all have their strong point and weak points, I think mxx’s work is better in general for most of the parts.