

Test 2 – Version 1 & 2

Last, First Name		Student ID	
------------------	--	------------	--

Do not detach the sheets since, in the past, there have been times when they became lost.

Question 1: UML Class Diagram

In many video games there is a transport vehicle that can carry people, supplies and smaller vehicles. The player's user interface contains controls to drive the transport vehicle through the map. It has a configurable display providing optional information about the transport vehicle. In this question the optional information is: transport vehicle speed, transport vehicle hit points, number of people it is carrying, number of supplies it is carrying, number of smaller vehicles it is carrying.

This question has, at least, the following elements:

- World map
- Transport vehicle
- Smaller objects: Person, Supply, SmallerVehicle
- The controls for driving the transport vehicle
- 5 transport vehicle information displays configurable to show or not show their information (speed, HP, number of people, number of supplies, number of smaller vehicles)

Answer the following question:

- a) Name three useful design patterns that would be needed in this problem and state where you would use them. [6 points]

DESIGN PATTERNS

Composite (since items can be put into the transport vehicle and treated the same)

Decorator (since hit points are reduced for items within the transport vehicle)

Singleton (for the world map since there should be only one)

Model View Controller (for the interface between the user, transport, and map)

Optional Observer but the problem does not require it since all views are updated if available.

OBJECT ORIENTED DESIGN

Modularization (world, transport, person, supply, smaller vehicle, etc.)

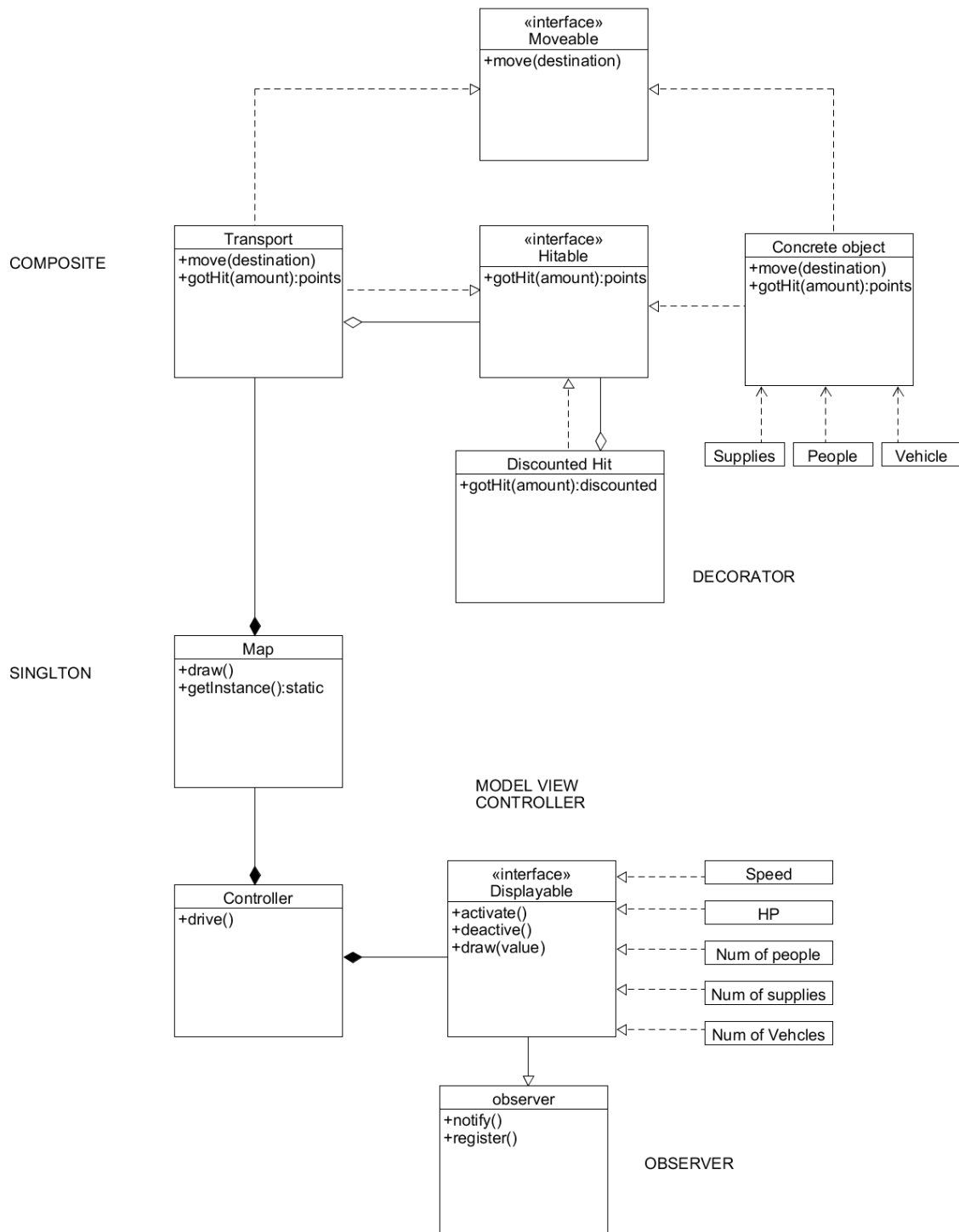
Design by contract (common interface between person, supply, and smaller vehicle)

Inheritance (the vehicles, common interface display elements)

Aggregation (smaller vehicles into transport, transport into world, displays)

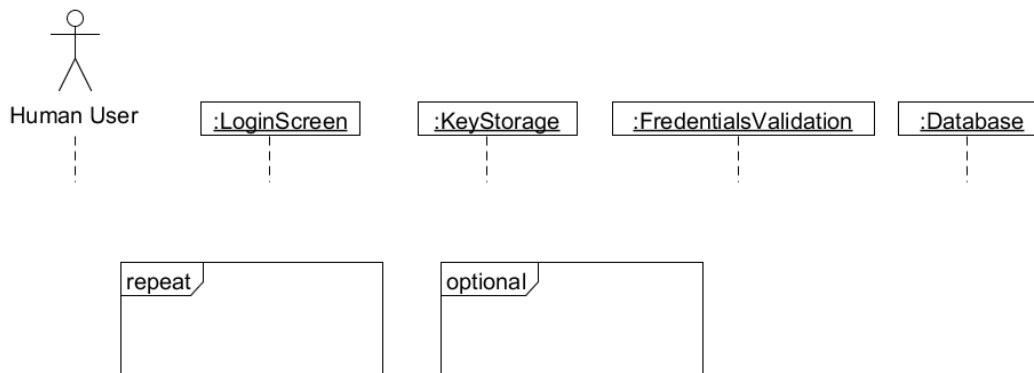
- b) Draw the class diagram for this problem using optimal design patterns and good object-oriented practices. [10 points]

It should look something like this:



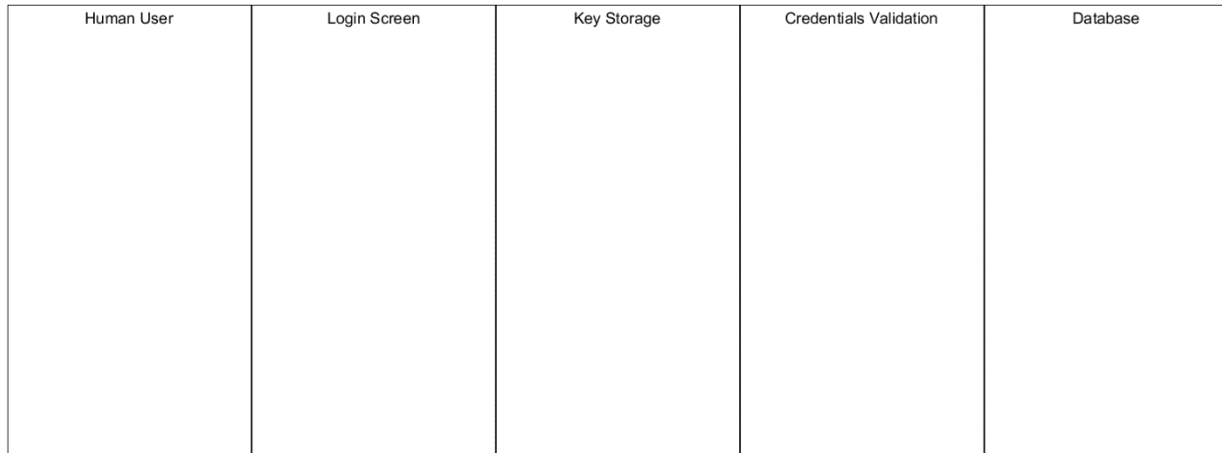
Question 2: UML Sequence Diagram [10 points] UML Activity Diagram Solution

The login procedure is a common identity validation technique. It consists of a screen asking the user to enter their username and password. Once the information is entered the user presses the submit button. Draw a sequence diagram that depicts the following login algorithm that begins after the user presses the submit button (pressing the submit button is the beginning of the validation algorithm). The algorithm begins with the LoginScreen object that implements the submit button method. The username and password are passed as parameters to the submit method. The submit method queries (calls) the browser's KeyStorage object for the user's secret key by passing the username and receiving the key or NULL if not found. The method is called getKey. If a NULL is returned the login procedure ends prematurely returning false. Submit then encrypts the password with that private key. Then the username and encrypted password is passed to the CredentialsValidation object by calling the validateUser method. The method validateUser then queries the Database object with the username to find the secret key associated with that username according to the database. The method is called findKey. The validateUser method then uses that secret key to decrypt the password and then checks the database to see if the decrypted password agrees with the password recorded in the database. The method is called findUser and has username and the decrypted password as parameters. If it matches then we are done and the algorithm returns true. If it does not match then the user is permitted to repeat this process 3 times. If the user fails 3 times the algorithm terminates returning false.

SEQUENCE DIAGRAM**Points:**

- +5 for syntax:
 - +2 for all the classes as depicted
 - +2 for the repeat box and the option (or if) box
 - +1 for NOT using activity diagram elements
- +5 for following the function call requirements as outlined in the question

ACTIVITY DIAGRAM



Points:

- +5 for Syntax:
 - +3 for using the class boxes
 - +2 for using flowchart elements
- +5 for the solution:
 - +2 for properly expressing the difference between calls to each class
 - +3 for following the requirements in the question

EXTRA PAGE