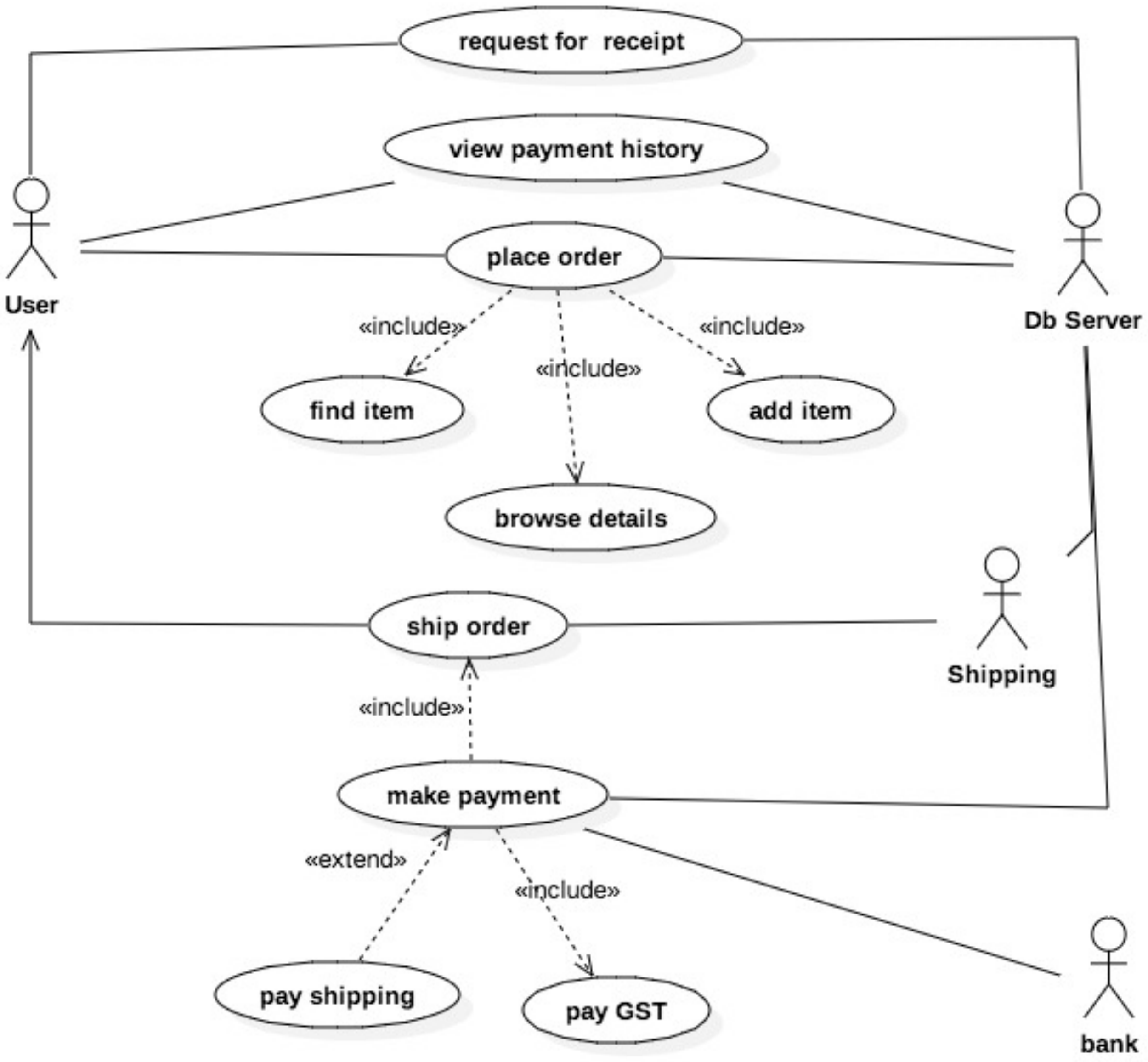
ENSF 613 - Software Requirements Analysis and Process Management

Fall 2019 – Test 2 Solutions

Question I (10 marks): In the following space draw a use case diagram that illustrates the following requirements for an Online Shopping System (OSS).

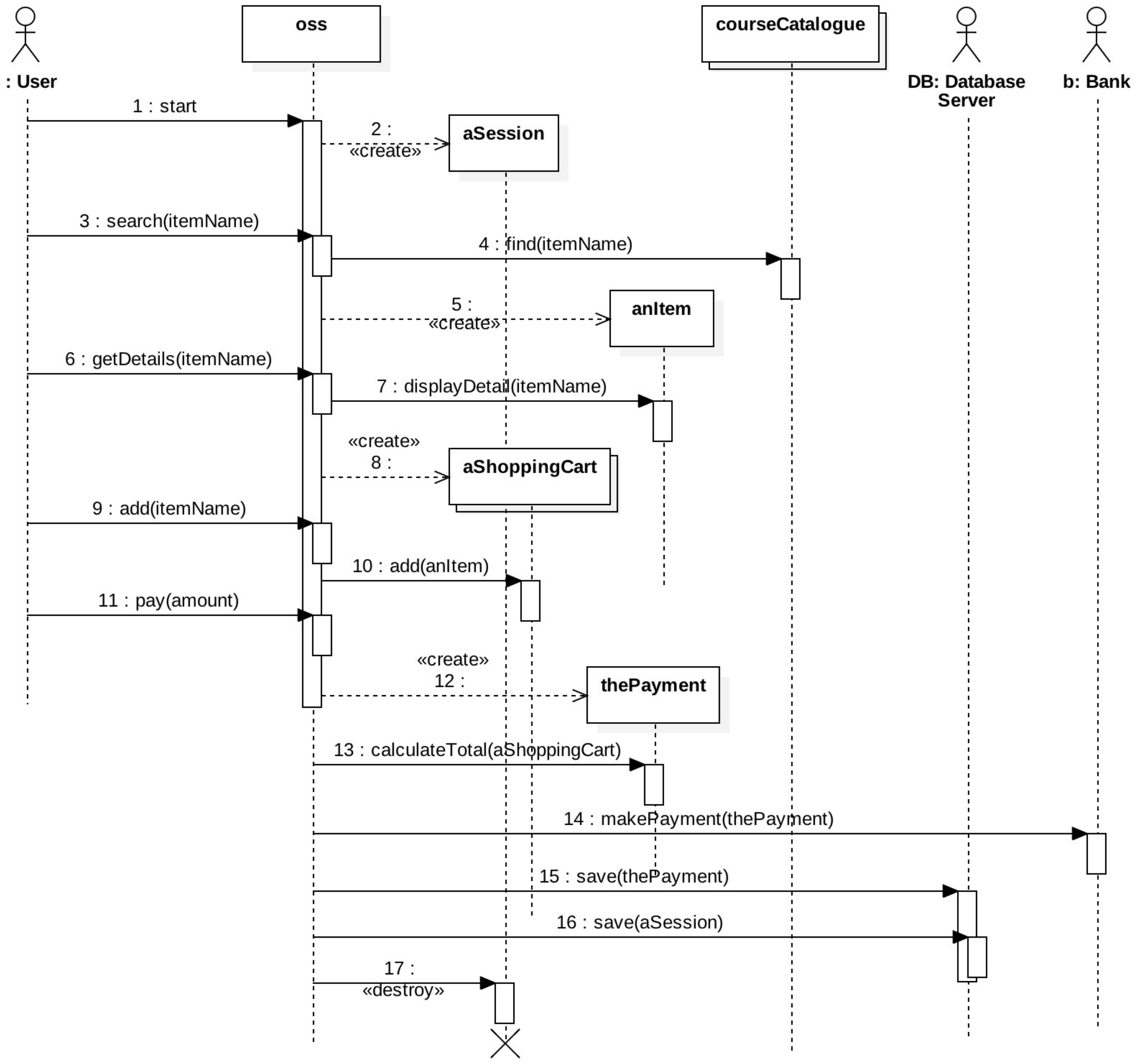
- Users should be able to sign in and place an order, that include operations such as:
    - Searching for items in the system catalogue
    - Browsing item details (description, technical specs, price, etc.)
    - Adding item(s) to the shopping-cart.
- Users should be able to make payment. Payments always must include GST-payment. However, shipping payments depend on several criteria such as delivery distance, promotion code, etc. In other words not every purchase has a shipping cost.
- Users should be able to view their payment history and print their past receipts
- Orders will be shipped to the user, once the user made relevant payments and confirmed by the bank or a financial institute.

This is only a POSSIBLE Solution based on given Scenario.



<u>Question II (10 marks)</u>: In the following space draw a sequence diagram for the given scenario that shows interaction between customer and the Online Shopping System, OSS.

- *Joe who is already logged into the system, wants to purchase some items.*
- *The system creates a session (Note: a session has date, time and other information that allows the system to retrieve this session if user leaves the webpage and returns later to continue shopping).*
- *Joe searches for an item on the catalogue.*
- *Then he browses the detail information of the item (description, specifications, price, etc.), and selects one or more items that he needs.*
- *Selected items will be added to his shopping-cart.*
- *Then he makes payment to the bank, receives a receipts via his email, and session ends.*
  *Please ignore type: **<u>courseCatalouge</u>** should read **<u>theCatalouge</u>***



<u>Question III (10 marks)</u>: Assume we want to develop a class called **Vector** (a class somewhat similar an **ArrayList** in Java) that its size and capacity for holding numeric data type (double type) will dynamically change. Here is a description of memory management policy that we would like to use:

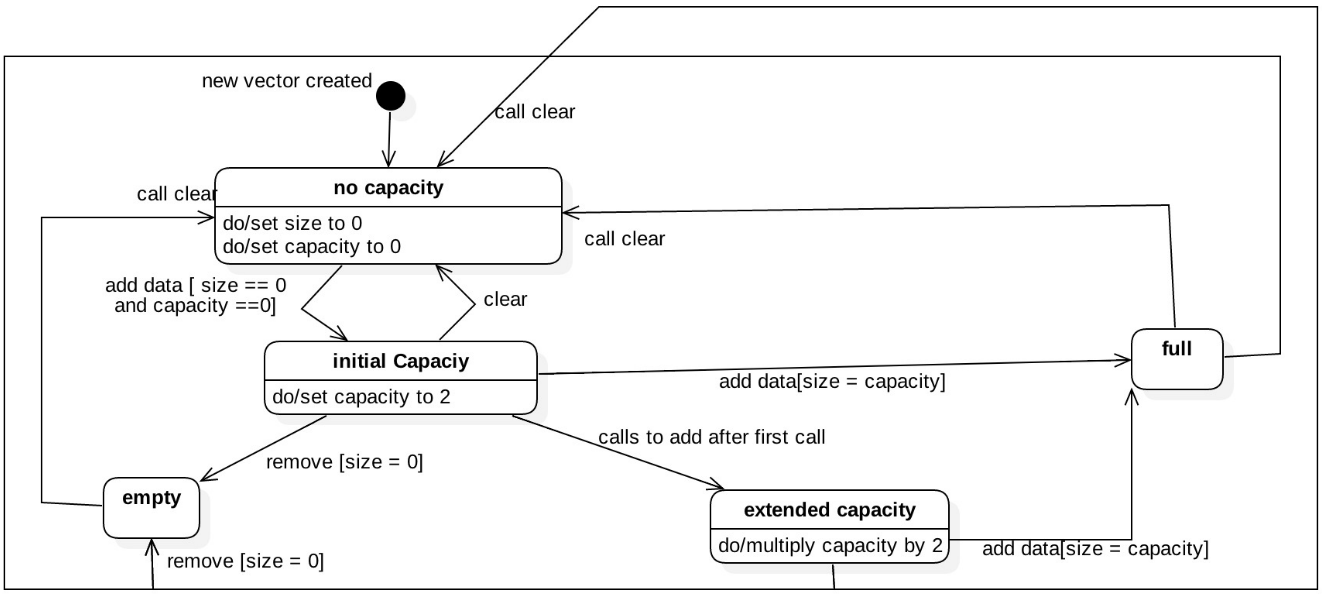
- When constructor is called, it will create the object with no data and no capacity. It means size and capacity of the vector are both zero.
- When adding new data for the **first time**, **capacity will be set to 2**, and then the data will be added. After adding the first data, vector will have one more space to add a new data.
- After adding first data, if list is full its capacity will be doubled (**will be multiplied by 2**), and then the new data will be added.
- If a data is removed from list, the remaining data to the right of the removed element will be shifted to the left to fill the gap. Then, size will be reduced by1. However the capacity remains unchanged.
- Users may also use other functions such as, **clear**, which means removes all the elements of the list, and reduces the values of size and capacity to zero.

To give you a better idea a partial java code with some pseudo-code for this class is here:

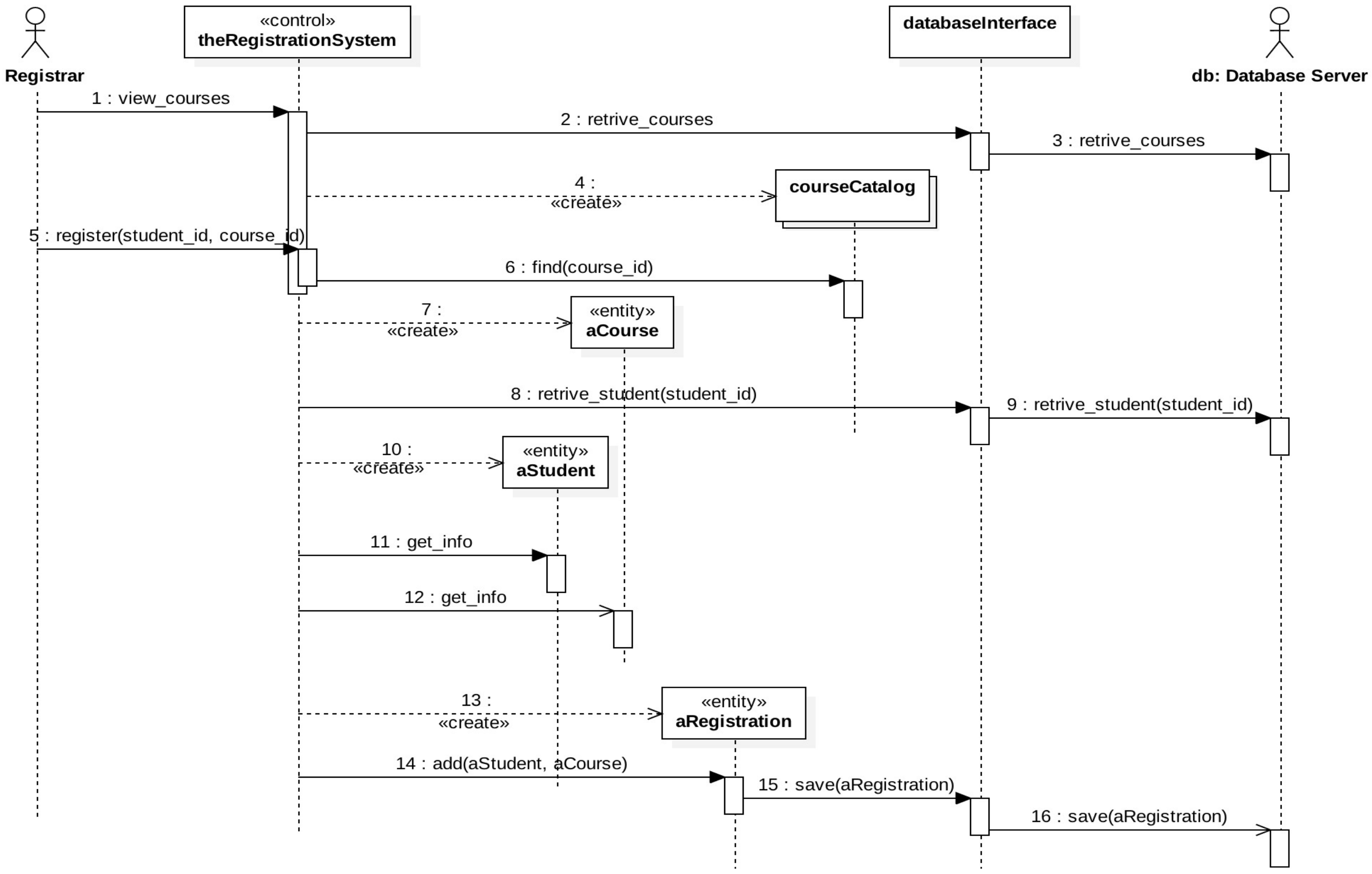
```
class Vector {
  private double [] storage;
  private int size;
  private int capacity;
  public Vector()
  { // creates a vector object,  and sets size and capacity to zero
  }
  public boolean isFull() {if (size == capacity) return true; return false; }
  public boolean isEmpty() {if (size == 0) return true; return false; }
  public boolean hasCapacity (){if (size < capacity) return true; return false; }

  public void add(int el_val) {
  // if size == 0 and capacity == 0 sets capacity to 2. Then allocates memory.
  // else if size == capacity multiplies capacity by 2. Then allocates memory
  // else if size < capacity does nothing
  // Then, adds el_val into the allocated space for storage
  }
  public void remove(int val){
  // If one or more elements match val all of those elements are removed, remaining
  // data will be shifted to the left, and the value of size will be reduced. But the
  // value of capacity remains unchanged.
  }
  public void clear(){
    // PROMISES: size and capacity will be set to zero, and memory will be
    // de-allocated
  }
}
```

Based on above description and the given code, draw a State Transition Diagram that shows the possible **states** of a **Vector** object.

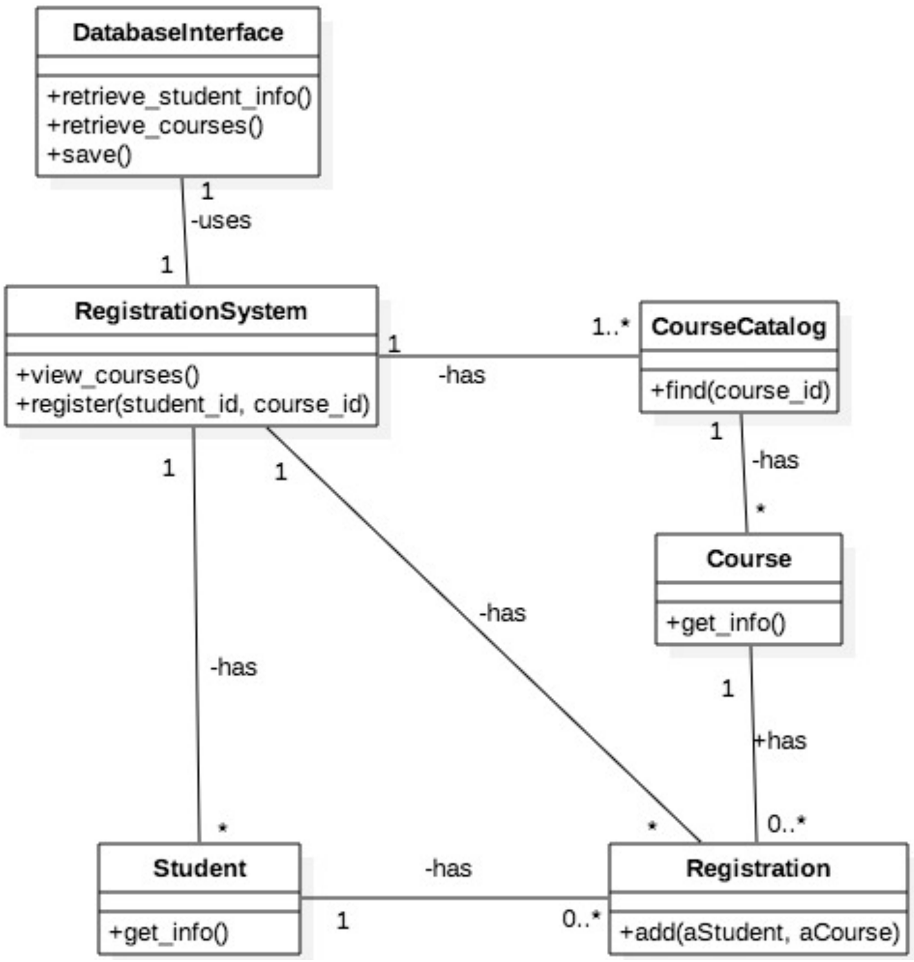
<u>Solutions:</u>



<u>Question IV (10 marks)</u>: Consider the following sequence diagram and draw an analysis-class-diagram (know as system's concept or conceptual model) that shows class names, and their behaviours (methods). Don't show attributes.



<u>Solution:</u>



DatabaseInterface
+retrieve_student_info()
+retrieve_courses()
+save()
1
-uses
1
RegistrationSystem
+view_courses()
+register(student_id, course_id)
1
-has
1..*
CourseCatalog
+find(course_id)
1
-has
*
Course
+get_info()
1
+has
0..*
1
1
-has
-has
*
*
Student
+get_info()
-has
1
0..*
Registration
+add(aStudent, aCourse)