INFO2320 Assignment 2

For this assignment, the same **INFO2320 A1 rules** will apply.

Before you begin:

In MS Teams:

 Download the **A2_VPRU_TeamYY_Template.docx** from **eConestoga** and copy to your team's **General\A2** folder on your MS Teams **Files** tab.

 Rename your document to **A2_VPRU_Team#.docx**

- Put your entire **A2 solution** in this single **MS Word** document.

 Fill out the **Assignment 2 header** with your section, team # and student names.

Individual VP Lab (20% of A2 mark)

 Do the VP tutorial / lab for **Domain Class Diagram** and **State Machine Diagram**. Save your individual work to **A2_Lab_<student name>.vpp** and make a copy to **General\A2** folder of your **MS Teams**.

 Demo your individually completed diagrams to one of your team members.

 Fill out the **Individual Lab** section of your **A2_VPRU_Team#.docx**

- *Reminder: Members must sign their own name.*

 Read the **VPRU Case Study.pdf** from **A1** again, as needed.

Team Assignment 2 (80% of A2 mark)

You are working with the case study **Vacation Packages 'R' Us** (**VPRU**) Travel Service. This assignment will include the two subsystems:

- Resort Relations Subsystem (**RRS**)
- Vacation Booking Subsystem (**VBS**)

Accounting & Finance Subsystem (AFS) and Social Networking Subsystem (SNS) are not included in this assignment.

In Visual Paradigm:

 Create a new VP 16.3: **A2_VPRU_Team#.vpp**

- When your team solution is completed, copy the UML Domain Class diagram and State Machine diagram to your document **A2_VPRU_Team#.docx**

Task 1: Class Attribute Table

To get started, use the Event Table, Use Case Diagram and VPRU background information from **A1**.

Tip: Do not manufacture any new information.

Optional: You can clarify the classes and attributes by constructing a Class-Attribute-Values (CAV) table. See sample CAV table for RMO Customer 101, 102 and 103 from your Chapter 4 slides.

Task 1A: VPRU Resort Relations Subsystem (RRS)

 Identify the domain classes and attributes for **VPRU Resort Relations Subsystem** (**RRS**) and fill out the **Class Attribute** table below.

Class	**Attribute**	**Synonym** (optional)
Resort		
	resortID	
	resortName	
	…	
Vacation Package		
	voucherNumber	
	…	
…		
	…	

Task 1B: VPRU Vacation Booking Subsystem (VBS)

 Identify the domain classes and attributes for **VPRU Vacation Booking Subsystem** (**VBS**) and fill out the **Class Attribute** table below.

Class	Attribute	Synonym (optional)
Student		
	studentID	
	firstName	
	…	
VP Booking		Reservation
	bookingNumber	
	webID	
	…	
…		

Task 2: Domain Class Diagram (DCD)

Task 2A: VPRU Resort Relations Subsystem (RRS)

 Based on the domain classes and attributes you identified in Task 1A above, draw a **Domain Class Diagram (DCD)** showing domain classes, attributes and associations with multiplicity. Start your **DCD** by adding a **Resort** class and a **Vacation Package** class.

Tip: You can make reasonable assumptions for associations and multiplicities as long as they do not contradict the background information provided. Adding an identifier or {key} is optional in domain modeling. PKs and FKs must not be in the class diagram.

Optional: You can clarify the associations and multiplicities by constructing a Semantic Net or UML object diagram(s). See sample Semantic Net for RMO Customer-Order-OrderItem ERD from your Chapter 4 slides.

Task 2B: VPRU Vacation Booking Subsystem (VBS)

 Based on the domain classes and attributes you identified in Task 1B above, draw a **Domain Class Diagram (DCD)** showing domain classes, attributes and associations with multiplicity. Start your **DCD** by adding a **Student** class and a **VP Booking** class.

 Drag and drop existing **RRS** domain class(es) that have association(s) to **VBS**.

Hint: To show that the class(es) came from another subsystem, show the *fully-qualified owner* and hide the attributes and methods.

 Your **Task 2B** solution must identify at least one aggregation and at least one class hierarchy. *Hint: A customer account can have multiple transactions. In addition, a transaction can be either a "charge transaction" or a "payment transaction".*

Tips for DCD (both **Task 2A** and **Task 2B**):

 Review for some possible overlap with the list of classes and attributes from **Task 1 Class-Attribute** tables.

- Resolve duplicate classes/ attributes, if any (e.g., VP Booking vs. Reservation)
- Add missing classes and attributes
    - *Hint: business transactions must have a transaction date and time*
- Remove redundant attributes (e.g., those that represent associations)
- Add associations and multiplicity
- Remove any redundant associations (e.g., those that represent indirect associations)
- *Review and revise your solution.*

Task 3: VP Booking Object Life Cycle

Task 3A: VPRU VP Booking State Table

 Identify the relevant states of the **VP Booking** object life cycle (see all related VPRU background information from A1) and fill out the **state table** below

State ID	State Name	Transition Causing Exit
S01	On Hold	Primary guest pays initial deposit; …
	…	

Task 3B: VPRU VP Booking State Machine Diagram

 Based on the **state table** above, draw a **State Machine Diagram (SMD)**. Your diagram must include states, transitions with events, actions and guard conditions, as appropriate.

 Include the **State ID** in your **SMD**.

 Iterate Task 1, Task 2 and Task 3 above until they are all complete and consistent.

 Copy all the **UML diagrams** to your solution document **A2_VPRU_Team#.docx**.

 Checkpoint: Use the **A2 marking sheet** to self-evaluate your solution.

Submission Requirements:

 Create a **A2_Team#.zip** file that contains **all** the docx and vpp solution files (missing solution = zero mark). Include a copy of **all individual lab vpp files** from **General\A2** folder of your **MS Teams**.

 Submit your **A2_Team#.zip** file to the Assignment Dropbox on **eConestoga** (i.e., **A2_Dropbox**).