

INFO2320 Assignment 4

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

Before you begin:

In MS Teams:

- ☐ Download the **A4_VPRU_TeamYY_Template.docx** from **eConestoga** and copy to your team's **General\A4** folder on your MS Teams **Files** tab.
 - ☐ Rename your document to **A4_VPRU_Team#.docx**
 - Put your entire **A4 solution** in this single **MS Word** document.
 - ☐ Fill out the **Assignment 4 header** with your section, team # and student names.
-

Individual VP Lab (20% of A4 mark)

- ☐ Do the VP tutorial / lab for **First-cut Design Class Diagram** and **Communication Diagram**. Save your individual work to **A4_Lab_<student name>.vpp** and make a copy to **General\A4** folder of your **MS Teams**.
 - ☐ Demo your individually completed diagrams to one of your team members.
 - ☐ Fill out the **Individual Lab** section of your **A4_VPRU_Team#.docx**
 - *Reminder: Members must sign their own name.*
 - ☐ Read the **VPRU Case Study.pdf** from **A1** again, as needed.
-

Team Assignment 4 (80% of A4 mark)

You are working with the case study **Vacation Packages 'R' Us (VPRU)** Travel Service.

In Visual Paradigm:

- ☐ Make a copy of your existing **A2_VPRU_Team#.vpp** (Domain Class Diagram) to **A4_VPRU_Team#.vpp**.
 - When your team solution is completed, copy the **Design Class Diagram (DCD)** and **Communication Diagram** to your document **A4_VPRU_Team#.docx**

Checklist for drawing the First-cut DCD and Communication Diagram:

☐ Add the controller class (as needed)

- Add method names and visibility
- Add a stereotype designation (in VP, use <<control>>)

Note: Do not include default constructors and getter/setter methods as they are implied. You can also skip the method parameters in A4.

☐ Add the first-cut navigation visibility, attribute visibility and data types, associations with multiplicity.

☐ For navigation visibility, display the arrowheads, as appropriate. **Do not** add multiplicity to associations between controller and domain classes. *Tip: Navigable property can be set to Unspecified, True or False. To display, use Presentation Options | Show Navigations Arrows.*

☐ Follow the standard notation for classes:

- class names must be in PascalNotation
- attributes and method names must be in camelNotation

☐ For identifiers in your design classes, set the **attribute** specification's Type modifier to {Key}; do not use <<PK>> stereotype; do not include the foreign keys (FKs) in your classes since FKs do not belong to a UML Class Diagram.

☐ For **controller classes** in both DCD and Communication Diagrams, show the controller class as a **robustness analysis icon**. *Tip: Set the "Display as Robustness Analysis icon" under Presentation Options as checked.*

☐ Update the **First-cut DCD** (e.g., navigation visibility), to match all the modeling elements in the **Communication Diagrams**, as appropriate.

Task 1: Design Use Case: *Add a Resort*

Scenarios:

- **S1:** For *odd-numbered* teams: Add a resort that has **multiple facilities** and **single room type** (e.g., standard room with two double beds). For each facility, add at least one activity.
- **S2:** For *even-numbered* teams: Add a resort that has a **single facility** (e.g., multi-purpose clubhouse) and **multiple room types**. Add multiple activities to the facility.

Note: Outside the scenarios mentioned above, you can make any other reasonable assumptions, as needed.

Preparation Steps:

- ☐ In the **A4_VPRU_Team#.vpp** file, create a **new** class diagram:
 - **Resort Relations Subsystem (RRS) DCD**
 - ☐ Steps to reuse the existing domain classes from **A2**:
 - Open the **Class Repository** pane in the VP **View** tab.
 - Drag and drop these domain classes to the new **RRS DCD**.
 - **Resort**, **ResortFacility**, **ResortActivity**, **RoomType**, **VacationPackage**
- Result: The associations and multiplicities will appear automatically.

Tip: Rename your domain classes, as necessary. Also, update the attributes (visibility, data type), and any other modeling elements, as needed.

Task 1A: **Resort Relations Subsystem (RRS)** First-cut DCD

- ☐ Add a new **ResortHandler** class.
 - Bases on the scenarios (see **S1** and **S2** above), add the following method names (i.e., operations) to the above controller class:
 - **addResort()**
 - **enterResortFacilitySingle()** or **enterResortFacilityMultiple()**
 - **enterRoomTypeSingle()** or **enterRoomTypeMultiple()**
 - Add the appropriate stereotype and display as a robustness analysis icon
- ☐ Add the first-cut navigation visibility, attribute data types, associations with multiplicity associated with the above use case.

*Note: You will expand the methods in the **ResortHandler** class when you draw the Communication Diagram.*

Task 1B: **Add a Resort** Communication Diagram

- ☐ Create a new **Communication Diagram** for the above use case.
- ☐ For each method name (operation) listed in the **ResortHandler** class, extend the message to all relevant objects

Tip: Remember to create a new operation, as needed.

- ☐ Update the Class Diagram (e.g., navigation visibility), to match all the modeling elements in the Communication Diagram, as appropriate.

Tip: Follow the OOD with Communication Diagram technique described in Chapter 13 of your textbook.

Task 2: Design Use Case: *Add a Draft VP Voucher*

Synonym for **VP Voucher**: **Vacation Package**

Scenarios:

- **S1**: For *odd-numbered* teams: Lookup resort that caters to wedding packages (e.g., *resort facility that has “wedding chapel” as part of its description*). You must use *lookupResortForWeddings()* as part of your solution.
- **S2**: For *even-numbered* teams: Lookup resort that has facilities for outdoor activities (e.g., *the resort has an outdoor beach ball court*). You must use *lookupResortForOutdoors()* as part of your solution.

Note: Outside the scenarios mentioned above, you can make any other reasonable assumptions, as needed.

Task 2A: **Resort Relations Subsystem (RRS)** First-cut DCD

- ☐ Add a new **VPVoucherHandler** class.
 - Add the following method names (i.e., operations) to the controller class:
 - **addVPVoucher()**
 - **lookupResortForWeddings()** or **lookupResortForOutdoors()**
 - **lookupRoomType()**
 - ☐ Add the appropriate stereotypes, navigation visibility, data types, associations with multiplicity associated with the above use case.
-

Task 2B: **Add a Draft VP Voucher** Communication Diagram

- ☐ Create a new **Communication Diagram** for the above use case.
 - ☐ For each method name (operation) listed in the **VPVoucherHandler** class, extend the message to all relevant objects
 - ☐ Update the Class Diagram (e.g., navigation visibility), to match all the modeling elements in the Communication Diagram, as appropriate.
-

Task 3: Resort Relations Subsystem (RRS) DCD

- ☐ Ensure that each method name listed in the **ResortHandler** and **VPVoucherHandler** classes are traceable (i.e., you can follow the chain of method calls).
 - ☐ Review all the other classes and remove any methods that are not used in any of the Communication Diagrams.
 - ☐ **Checkpoint:** Use the **A4 marking sheet** to self-evaluate your solution.
-

- ☐ Copy the complete and consistent **DCD** and **Communication Diagrams** to your solution document **A4_VPRU_Team#.docx**.
-

Submission Requirements:

- ☐ Create a **A4_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\A4** folder of your **MS Teams**.
- ☐ Submit your **A4_Team#.zip** file to the Assignment Dropbox on **eConestoga** (i.e., **A4_Dropbox**).