Assignment 3: Fully-Developed Use Case Descriptions with SSD and Activity Diagrams

Yilong Yu – Ben Powers – [Jaehyun](javascript://) Park

8170995 – 7986250 – 8105702

Systems Analysis and Design / INFO2320

[David Turton](https://conestoga.desire2learn.com/d2l/le/content/229806/viewContent/4630975/View) / [Sachin](javascript://) Parikh

October 15th, 2018

Team Name: Scoopy-doo Teacher Name: [Sachin](javascript://) Parikh Due: Oct 15, 2018 6:00 PM

Systems Analysis & Design – Assignment 3:

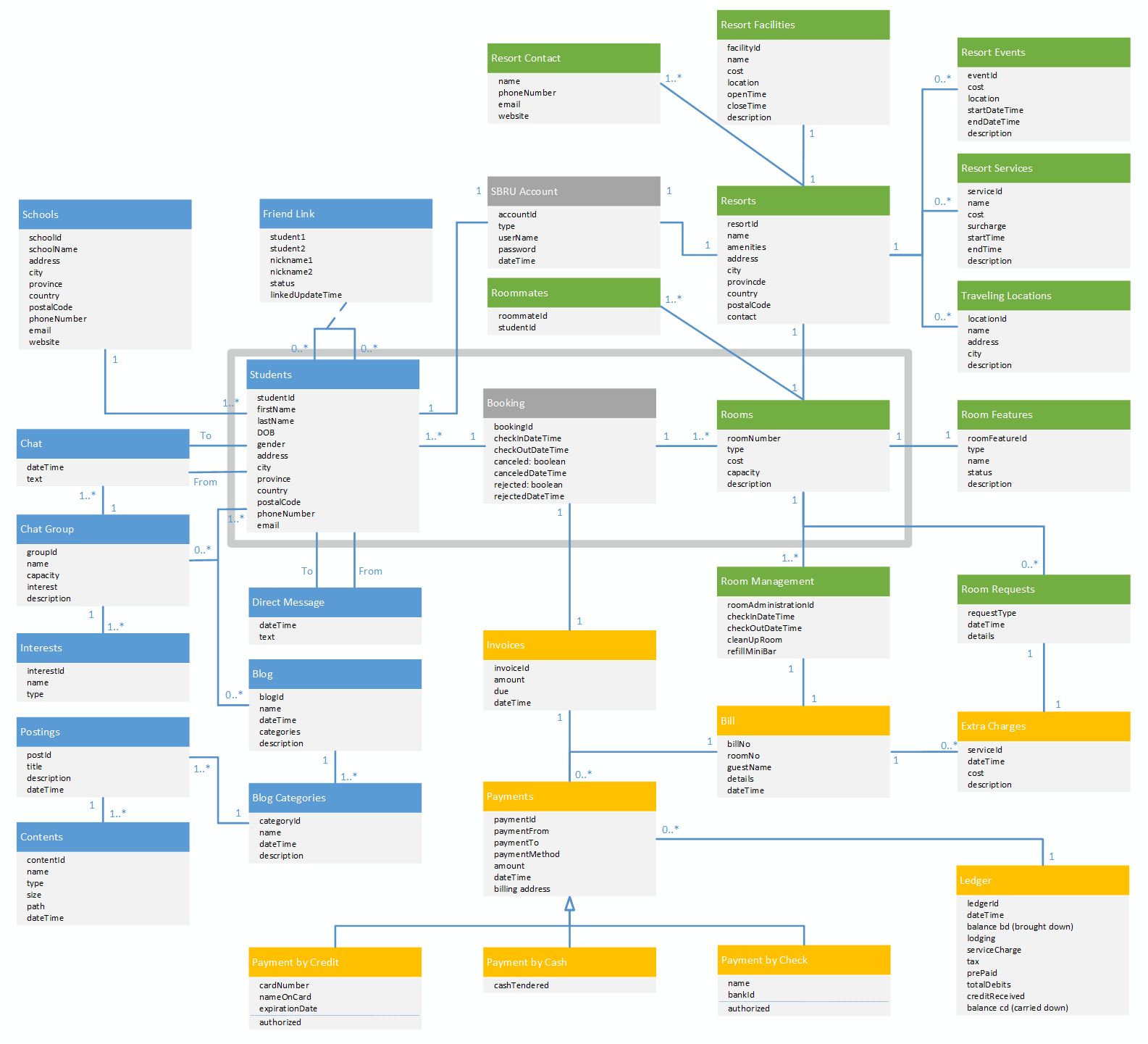
Fully-Developed Use Case Descriptions with SSD and Activity Diagrams

You are working with the case **Spring Breaks ‘R’ Us** (**SBRU**) Travel Service. The four major subsystems of the solution being proposed are:

* Resort Relations
* Student Booking
* Accounting & Finance
* Social Networking

Chapter 5 Questions (page 152)

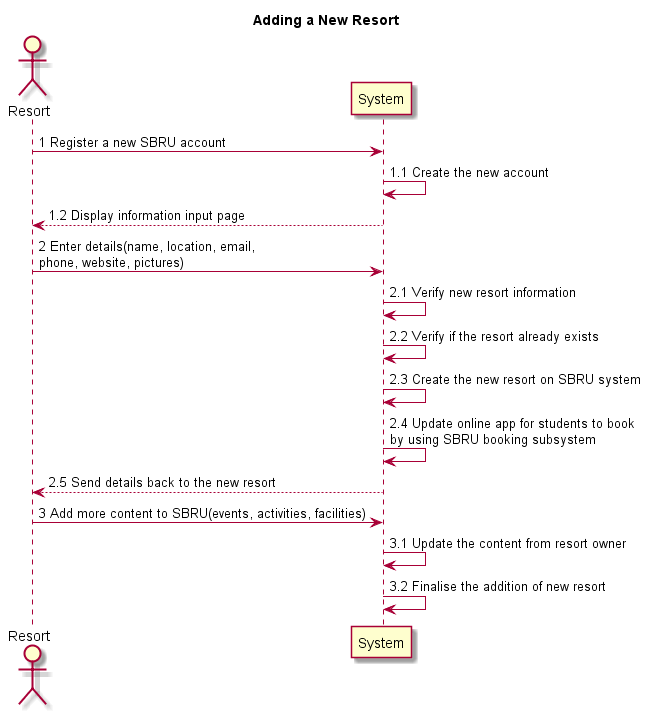
1. Create a single combined Domain Class Model starting from what you created in the first two assignments, enhanced to fully support all four subsystems of SBRU. Show all classes, associations, attributes and multiplicities.



1. For the Use Case *Add new resort*, write a fully developed Use Case Description and a System Sequence Diagram (SSD) for it. Use your domain classes to help understand the flow of activities.

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Add new resort | |
| **Scenario** | Register and add a new resort on SBRU system | |
| **Triggering Event** | New resort wants to use their resort with SBRU booking services. | |
| **Brief Description** | SBRU adds a new resort, new resort is posted on SBRU booking services. | |
| **Actors** | SBRU & Resort owner | |
| **Related Use Cases** |  | |
| **Stakeholders** | Resort Owner, SBRU Owner, SBRU location manager. | |
| **Preconditions** | New resort must exist, SBRU must provide booking services | |
| **Postconditions** | New resort must exist, SBRU must provide booking services, revenue earned will be recorded. | |
| **Flow of Events** | **Actor** | **System** |
| 1. Resort owner registers a new SBRU account 2. Resort owner enters relevant details (name, location, email, phone, website, pictures) 3. Resort owner adds more content to SBRU (events, activities, facilities) | * 1. System creates the account   2. Display information input page   2.1 Verifies new resorts information  2.2 Verify if the resort already exists  2.3 Create the new resort on SBRU system  2.4 Updates online app for students to book by using SBRU booking subsystem  2.5 Send details back to the new resort  3.1 Update the content from resort owner  3.2 Finalize the addition of new resort |
| **Exception Conditions** | 1.1 New account registration fails.  2.2 The resort already exists.  2.4 System does not update. | |

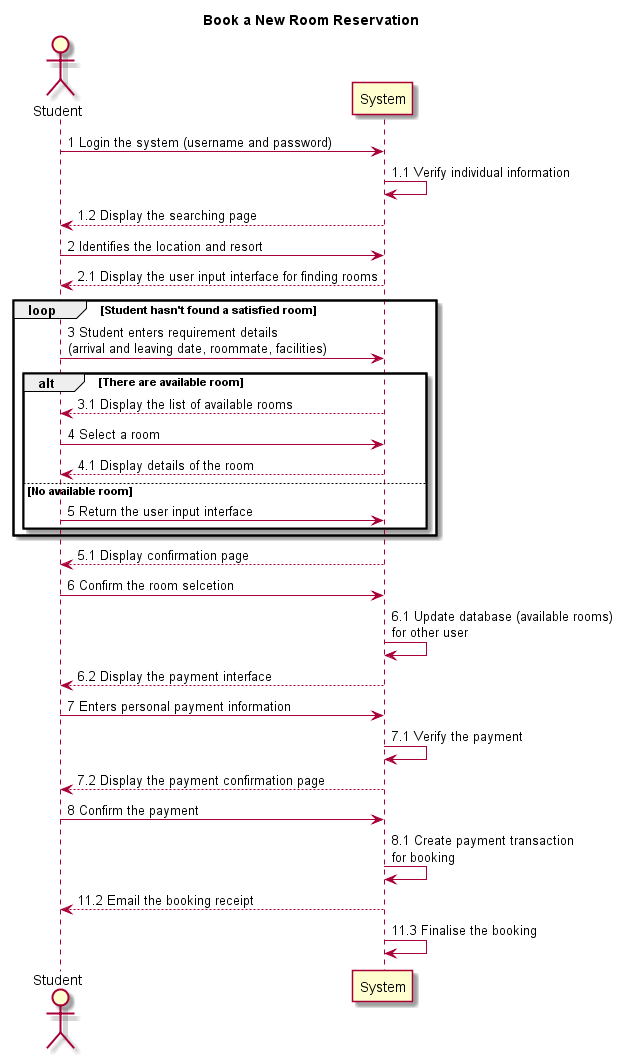
Please check the SSD next page.



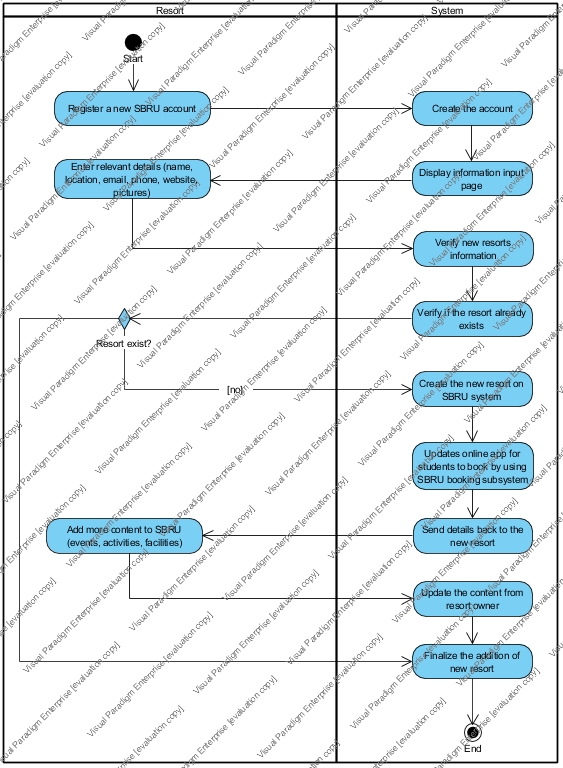
1. For the Use Case *Book a reservation*, write a fully developed Use Case Description and an SSD for it. Again, use your domain classes to help understand the flow of activities.

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Book a reservation | |
| **Scenario** | Book a room reservation online | |
| **Triggering Event** | Student uses an electronic device to book a room that has been selected from the list of available rooms online | |
| **Brief Description** | Student uses computer, cellphone, or tablet to login the SBRU system, search a location or resorts, browse resorts and available rooms, select an room, browse the room’s details, confirm the booking, make payment for the booking, finalize the booking.  SBRU system allows personal or group booking | |
| **Actors** | Student & SBRU | |
| **Related Use Cases** | Includes *login the SBRU system*  Includes *contact with friends*  Includes *create student group*  Includes *pay for the reservation* | |
| **Stakeholders** | Technical department: to maintenance the booking subsystem  Accounting department: to verify student’s payment, and transfer the payment to the resort  Marketing department: to collect the data of students’ booking for studying booking behaviour  Student: to book a room quickly  Resort: to receive student’s payment, and then offer services for students | |
| **Preconditions** | Student account must really exist  Resort must really exist  The information of resorts/rooms is real and effective  Student must have adequate money in his/her account for payment | |
| **Postconditions** | The list of available room must be updated  The booking must be related with a student/group and resort  The payment transferring line must be created for money transferring | |
| **Flow of Events** | **Actor** | **System** |
| 1. Student logins the SBRU by entering username and password 2. Student identifies the location and resort 3. Student enters requirement details (arrival and leaving date, roommate, facilities) 4. Student selects a room 5. Student chooses to return to the user input interface; Repeat the step 3, 4 until student decides a room that will be booked 6. Student confirms the room selection 7. Student input payment information 8. Student confirms the payment | 1.1 Verify student’s individual information  1.2 Display the Searching page  2.1 Display the user input interface for finding rooms  3.1 Display the list of all available rooms  4.1 Display the details of the room  5.1 Display the confirmation page  6.1 Update database (available rooms) for other users  6.2 Display the payment interface  7.1 Verify the payment  7.2 Display the payment confirmation page  8.1 Create payment transaction for booking  8.2 Email the booking receipt  8.3 Finalise the booking |
| **Exception Conditions** | 1.1 Login fails, then:  a. student registers a new SBRU account or,  b. system invokes *Maintenance user information* use case  6.1 If there is no available rooms, listing rooms fails. Student can:  a. quit the booking system or,  b. re-search other resorts and rooms  10.1 Payment transaction fails, then:  a. booking is cancelled or,  b. student chooses other payment method to complete the booking | |

Please check the SSD next page.



1. Draw an Activity Diagram to show the flow of activities associated with the Use Case *Add new resort*.



1. Do a CRUD analysis based on your current Domain Class Model and the Use Cases you have built.
   1. Use this to ensure the domain classes, use case activities and SSDs all fully support each other … otherwise, you’ll lose marks.

SBRU Account

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case vs.  Entity/Domain Class | SBRU Account | Students | Resorts |
| Create SBRU Account | C |  |  |
| Look up SBRU Account | R |  |  |
| Update SBRU Account | U |  |  |
| Delete SBRU Account | D |  |  |
| Link Student into SBRU account | U | R |  |
| Unlink Student into SBRU account | U | R |  |
| Link Resort into SBRU account | U |  | R |
| Unlink Resort into SBRU account | U |  | R |

Resort Relations

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Use Case vs.  Entity/Domain Class | SBRU Account | Resort | Resort Contact | Resort Facilities | Resort Services | Resort Events | Traveling Locations |
| Add new Resort | R | C | C |  |  |  |  |
| List up all Resort | R | R | R |  |  |  |  |
| Look up a Resort |  | R | R |  |  |  |  |
| Update Resort Info. |  | U |  |  |  |  |  |
| Update Resort Contact |  | R | U |  |  |  |  |
| Look up a Resort Facilities |  | R | R | R |  |  |  |
| Look up a Resort Services |  | R | R |  | R |  |  |
| Look up a Resort Events |  | R | R |  |  | R |  |
| Look up Tourism Attraction |  | R | R |  |  |  | R |
| Add new Facility |  | R | RC | C |  |  |  |
| Add new Service |  | R | RC |  | C |  |  |
| Add new Event |  | R | RC |  |  | C |  |
| Add new Tourism Attraction |  | R | RC |  |  |  | C |
| Update Resort Facility |  | R | RU | U |  |  |  |
| Update Resort Service |  | R | RU |  | U |  |  |
| Update Resort Event |  | R | RU |  |  | U |  |
| Update Tourism Attraction |  | R | RU |  |  |  | U |
| Delete Resort Facility |  | R | UD | D |  |  |  |
| Delete Resort Service |  | R | UD |  | D |  |  |
| Delete Resort Event |  | R | UD |  |  | D |  |
| Delete Tourism Attraction |  | R | UD |  |  |  | D |
| Delete Resort Contact |  | R | D |  |  |  |  |
| Delete Resort | R | D | D | D | D | D | D |
| Use Case vs.  Entity/Domain Class | Resort | Rooms | Room Features | Room Mgmt. | Room Requests | Room-mates |  |
| Register a Room | R | C | C | C |  |  |  |
| Look up Room info. | R | R | R | R |  |  |  |
| Update Room info | R | U | U | U |  |  |  |
| Check In | R | R | R | U | C | C |  |
| Check Out | R | R | R | U | R | R |  |
| Unregister a Room | R | D | D | D | D | D |  |
| Use Case vs.  Entity/Domain Class | Resort | Rooms | Room Requests | Resort Facilities | Resort Services | Resort Events | Extra Charges |
| Book a Resort Facility | R | R | U | R |  |  | C |
| Update Booking | R | R | U | R |  |  | U |
| Cancel Booking | R | R | U |  |  |  | D |
| Request Service | R | R | C |  | R |  | C |
| Update Service Request | R | R | U |  | R |  | U |
| Cancel Service Request | R | R | U |  |  |  | D |
| Join the Event | R | R | U |  |  | R | C |
| Update Joining | R | R | U |  |  | R | U |
| Cancel Joining | R | R | U |  |  |  | D |

Student Booking

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Use Case vs.  Entity/Domain Class | SBRU Account | Students | Booking | Resorts | Rooms | Invoices | Bill |
| Set Tour days | R | R | C |  |  |  |  |
| Search Resort Available | R | R | R | R |  |  |  |
| Search Room Available | R | R | R | R | R |  |  |
| Book a Room | R | R | U | R | R | C | C |
| Update Booking | R | R | U | R | R | U |  |
| Cancel Booking | R | R | D | R | R | D |  |
| Pay for the room deposit | R | R | U | R | R | U | U |
| Pay for the room full price | R | R | U | R | R | U | U |

Accounting & Finance

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use Case vs.  Entity/Domain Class | Invoices | Bill | Extra Charges | payment | ledger |
| Pay for the room deposit | R | R |  | C | C |
| Pay for the room full | R | R |  | C | C |
| Pay Extra Charges | R | R | R | C | C |
| Cancel Payment | R | R | R | U | U |
| Analyze Revenue & Profit | R | R | R |  | R |

Social Networking

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use Case vs.  Entity/Domain Class | Student | Schools | Friend Link | Direct Message |  |
| Look up Friend | R | R |  |  |  |
| Add new Friend | R | R | C |  |  |
| Update Friend | R | R | U |  |  |
| Delete Friend | R | R | D |  |  |
| Send Message | R |  | R | C |  |
| Read Message | R |  | R | R |  |
| Use Case vs.  Entity/Domain Class | Student | Chat Group | Interest | Chat |  |
| Create new Chat Group | R | C | C |  |  |
| Update Chat Group Info | R | U | U |  |  |
| Add Member to Chat | R | U |  |  |  |
| Remove Member to Chat | R | U |  |  |  |
| Join to Chat Group | R | U |  |  |  |
| Unjoin from Chat Group | R | U |  |  |  |
| Send Message | R | U |  | C |  |
| Read Message | R | U |  | R |  |
| Delete Chat Group | R | D | D | UD |  |
| Use Case vs.  Entity/Domain Class | Student | Blog | Blog Categories | Postings | Contents |
| Create new Blog | R | C | C |  |  |
| Update Blog | R | U | U |  |  |
| Search Blog | R | R | R |  |  |
| Subscribe Blog | R | U | RU |  |  |
| Unsubscribe Blog | R | U | RU |  |  |
| Add new Posting | R | R | U | C |  |
| Attach Files to Posting | R | R | U | U |  |
| Search Posting | R | R | R | R | R |
| Update Posting | R | R | R | U | U |
| Delete Posting | R | R | R | D | D |
| Delete Blog | R | D | D |  |  |

Hand In

* Zip your answers into a single file and upload it to the assignment folder on eConestoga … only have one team member upload per team.
* Hand the marking sheet in during the lab period for the week it is due, with your team name, lab section number and teacher name.