Hotel Reservation and Payment System (HRPS).

HRPS is an application to computerize the processes of making hotel reservation, recording of orders and displaying of records. It is solely used by the hotel staff.

The following are information about the application:

About Guest

a) Guests can be added with details like name, credit card details, address, country, gender, identity, nationality, contact when check in.
b) Identity could be driving license or passport.

About Room

a) Rooms can be categorized according to its type: Single, Double, Deluxe, VIP Suite, etc. (you may refer to hotel website for the types). Each type is at different rate.
b) Rooms have details like room number, bed type (single/double/master, WIFI enabled, facing (with view), smoking/non-smoking, status, etc.
c) Room availability/details can be checked by entering the room number (id)/guest name.
d) Room availability status can be Vacant, Occupied, Reserved, Under Maintenance.

About Reservation

a) When a room is reserved, it will have a corresponding reservation details.

Functional Requirements:

a. Create/Update/Search guests detail (*Search by name using keyword/s*)
b. Create/Update/Remove/Print reservation
c. Create/Update rooms details (include setting status like 'Under Maintenance",
d. Entering room service orders - list menu items for selection
e. Create/Update/Remove room service menu items.
f. Check room availability
g. Room Check-in (for walk-in or reservation)
h. Room Check-out and print bill invoice (with breakdowns on days of stay, room service order items and its total, tax and total amount)
i. Print Room Status statistic report by
    a. room type occupancy rate (room status = vacant) - *(room type, number and list the room number)*.
    Eg, [for illustration only]

Single : Number : 10 out of 20
Rooms : 02-03, 03-04, 03-05,…….
Double : Number : 5 out of 10
Rooms : 02-04, 05-04, 05-05,…….
…….

    b. room status *(status type and list the room number)*
    Eg, [for illustration only]

Vacant :
Rooms : 02-03, 03-04, 03-05,…….
Occupied :

Lab 3: Seating reservation application program

The application is to be developed as a **Command**
User Interface). Data should be stored in flat fi
eLearning Topics in NTULearn on FileI/O. Sam
or binary (Serializable) file. [Learn from the fun
No database application (eg MySQL, MS Access, e

**You will create your own test cases and data t
should also create test cases to test for cases* *of
room/s upon payment and removing of reservation/
with rooms under maintenance, generating of b
Appendix A for reference.*

Expected outputs:

```
(1) Show number of empty seats
(2) Show the list of empty seats
(3) Show the list of seat assignments by seat ID
(4) Show the list of seat assignments by customer ID
(5) Assign a customer to a seat
(6) Remove a seat assignment
(7) Exit

  Enter the number of your choice: 5
Assigning Seat ..
  Please enter SeatID: 10
  Please enter Customer ID: 10001
Seat Assigned!

  Enter the number of your choice: 5
Assigning Seat ..
  Please enter SeatID: 12
  Please enter Customer ID: 10002
Seat Assigned!

  Enter the number of your choice: 5
Assigning Seat ..
  Please enter SeatID: 8
  Please enter Customer ID: 10003
Seat Assigned!
```

APPENDIX A:

Suggested Test cases

The list of test cases are guide for your testing and demo video. Depending on your design and user-friendliness of your data entries process, there may be multiple steps taken.
[Note : You should also demonstrated at least 5 cases of input error checking done in your application]

a. To create (using the functions listed in Function requirement)
   - 2 guests occupied each room type (single, standard, VIP, suite, deluxe,) with different details, eg bed type, etc.
   - 2 rooms of each room availability status Under Maintenance.
   - 3 reservations with corresponding different guest details [** set check-in date as 1 week later*]
   - At least 5 room service menu items

b. Print Room Status statistic report by :
   - room type occupancy rate [*expected : details created above*]
   - room status [*expected : details created above*]

c. Print all reservations [*expected : details created above*]

d. Search for a guest and list its details [*found and not found*]

e. Update a guest's credit card detail
f. Search for this guest and list its details
g. Update a rooms details - change room status to 'Under Maintenance'
h. Create a reservation *[* set check-in date as 1 day later]*
   - Check availability of a room
   - After confirmation, [*acknowledgement receipts is printed with details*, a room status changed reserved]

i. Print all reservations [*expected : details created above*]
j. Print Room Status statistic report by :
   - room status - reserved and [*expected : inclusive of details created/changed above*]

Submission

- **Your group submission should include the following:**
  - The report (separate diagram file if diagram is unclear in report)
  - Video and audio recording of the demonstration.
  - All implementation codes and java documentation (javadoc).
  - Other relevant files (eg data files, setup instruction, etc)

This is a **group assignment**, and one submission from each group.

Submission cont’

Soft copy of your deliverables to be **uploaded** to your individual CE/CZ2002 **LAB site** (eg FEP1, FSP1, etc) in **NTULearn**. The link is provided on the left panel "Assignment Submission". Please approach your lab TA if you encounter any problem about submission.

File name convention : <lab_grp>-grp<assignment_grp#>.<ext> Eg, FEP2-grp3.pdf [<ext> can be pdf, doc, zip, mpeg, wmv or mp4]

[In the event your video file is too large to upload, you may upload to **Youtube** and **provide the link in your report**. **No online storage** (like DropBox, Google Drive, etc) is allowed – no updates after submission].

1. **ASSESSMENT WEIGHTAGE**

UML Class Diagram [**30 Marks**]

- Show mainly the Entity classes, the essential Control and Boundary classes, and enumeration type (if there is).
- Clarity, Correctness and Completeness of details and relationship.

Design Consideration [**15 Marks**]

- Usage of OO concepts and principle - correctness and appropriateness

Implementation Code [**35 Marks**]

- Diagram to Code correctness, readability, Javanaming convention, exception handling, completeness of Java Doc and overall quality.
- A Java API HTML documentation of **ALL** your defined classes using Javadoc must be submitted. The use of javadoc feature is documented in Appendix D.

Demonstration and report [**20 Marks**]

- Coverage of application essentials and functionalities, user friendliness, demo flow, innovation.
- Report structure and reflection

1. **THE REPORT**

Your report will include the following :

a) A detailed UML **Class** Diagram for the application (exported as an image)
- show clearly the class relationship, notation
- notes to explain, if necessary

UML Class Diagram [30 Marks]

b) A **write-up** on your **design considerations** and use of OO concepts in your current design. In addition,
- It is proposed that in future that guest who has made reservation will be sent a reminder message via SMS or email (OR other messaging channel**s**) a few days before the actual booked date.
- Explain how your current design and further enhancement/s (if any) can cater to this new feature- using the design principles (SOLID, etc) to ensure reusability, extensibility and maintainability of your design.

Design Consideration [15 Marks]

c) Reflection: The difficulties encountered and the way to conquer, the knowledge learnt from this course, further improvement suggestion. Strong demonstration of learning points and insights of good design and implementation practices, based on experience gained from doing the assignment.

Demonstration and report [20 Marks]

d) A duly signed **Declaration of Original Work** form (Appendix B).

e) [**Optional**] Member's work contribution and distribution breakdown.
If your group feels that marks should be given based on contribution, your group can fill up the WBS.xls(in the same folder as assignment doc) and include it in this report. All members MUST consent to the WBS contents. ***You must also email the WBS.xls to the course-coordinator with ALL members in the loop.***

1. **DEMONSTRATION**

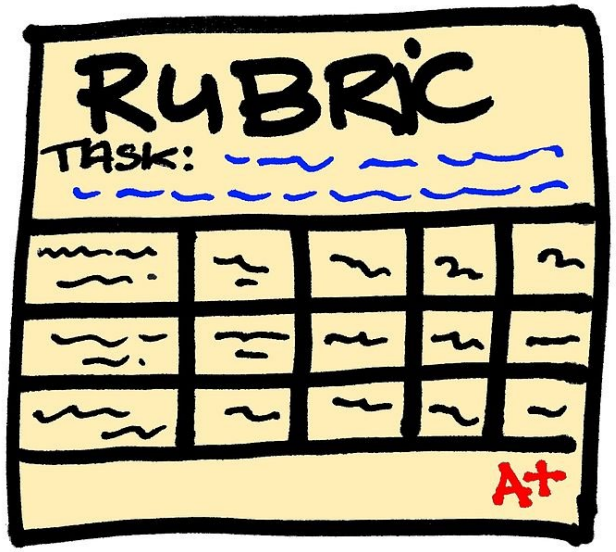
Your group is to produce a **video and audio recording** to demonstrate the working of the application – presenting ALL the required functionalities of the application and the suggested test cases in Appendix A. It is advised that you planned your demonstration in a story-boarding flow to facilitate understanding of your application. *Please introduce your members and group number at the start of video, the presenter should show his/her face while presenting.*

In the production, you may include :

a) Explaining essential and relevant information about the application
b) Run-through and elaborate on essential part/s of your implementation/coding

- ***The video duration must not exceed 15 minutes in total.***
- ***The font size used must be large enough to be readable and viewable.***
- ***The video quality must be clear.***
- ***The demo of the application is to done in real-time and NOT pre-run display.***

Grading rubrics-refer to GradingGuide file



Your group assignment and grading rubrics

Availability: Item is hidden from students. It will be available after Feb 28, 2022 12:00 AM.

Attached Files: Assignment 2022.pdf (1.221 MB)

GradingGuide_HRPS 2022.pdf (362.779 KB)

Class Diagram (30 marks)

Notes

- See the attached suggested class diagram (either versions or combination)
- These are Entity Classes for storing data – *usually nouns like Room, Guest, Car, etc.*
- Teams should have about the same classes.
- They may include Control and Boundary classes, eg HotelMgr, PaymentController, xxxUI, xxxForm)
- If you know about ER Diagram (Entity Relationship) in DB, you should able to relate to the Class relationships.

Assessment Guideline :

if the class diagram is too messy and hard to read, deduct 2 marks

- **25-30marks**
  - List of classes SIMILAR suggested (+ 3 more is fine), with detail listing of attributes and methods, appropriate relations and clear
- **19-24marks**
  - List of classes LESS THAN suggested, with detail listing of attributes and methods, appropriate relations and clear
- **12-18marks**
  - List of classes LESS THAN suggested, with MINIMUM of attributes and methods, appropriate relations and clear
- **7-11 marks**
  - List of classes LESS THAN suggested, with MINIMUM of attributes and methods, SPIDER WEB –like (a class relate to a lot of classes) class relationships – using Dependency is fine (the dotted slim arrow ---------->)
  - OR overused of <<Interface>> { more than 4)
- **3-6 marks**
  - Only Boundary or Control classes
- **0-2 marks**
  - No submission or irrelevant classes

1. **DEADLINE : 17th April 2022 (Week 13 Sunday), 11.59pm.**

Important:

Note that THREE (3) marks will be deducted for the delay submission of each calendar day. Lateness is based on the date the captured in NTULearn or subsequent resubmissions (whichever is later). So check your work before submitting.

-

Week 13