Computer Science and Creative Technologies

Coursework Specification

Module Details

Module Code	UFCF8S-30-2
Module Title	Advanced software development
Module Leaders	Rakib Abdur, Mehmet Aydin
Module Tutors	James Lear, Barkha Javed, Rakib Abdur
Year	2021-22
Component/Element Number	A
Total number of assessments for this module	Two assessments (1) Part-I: The first covers the analysis and design of a software system. (2) Part-II: The second covers the coding and testing of the design created in the first assignment.
Weighting	This coursework is worth 100 marks representing 100% of your total course grade.
Element Description	This assignment is to be completed in groups of three students.

Dates

Date issued to students	October 28, 2021
Making results and feedback to students	(1) Part-I: December 10, 2021 (2) Part-II: February 11, 2022
Submission Date	(1) Part-I: November 24, 2021 Demo session in the following week starting 29th November (during the scheduled lab session) (2) Part-II: January 12, 2022 Demo session in the following week starting 17th January (during the scheduled lab session)
Submission Place	Blackboard
Submission Time	14:00
Submission Notes (please read carefully)	(1) Part-I: Please submit a PDF document containing Use Case diagram, Class diagram, and Sequence diagram. (2) Part-II: Please submit a portfolio as a ZIP file with your code, a PDF report on Agile practices, Test cases used in your program in the form of a table, suitable program running screenshots capturing success/failure scenarios, and Individual report. Please note that all the members of a group must be present during the demonstration session, and at least one group member must upload the required portfolio/documents to blackboard. However, all the members of a group must upload their own individual report.

Feedback

Feedback provision will be	On the spot verbal feedback during the demo session + as appropriate written feedback uploaded to Blackboard.

Contents

Section 1: Overview of Assessment

This assignment assesses the following module learning outcomes:

- Analyse problems in order to identify software-solution approaches and requirements for computer-based software-intensive systems
- Apply object-oriented analysis and design techniques to a problem domain scoped at level 2 complexity
- Understand and use a Unified Modelling Language (UML) modelling tool and Python-based Interactive Development Environment (IDE) to develop object-oriented software implementations appropriate to level 2 complexity
- Design and implement Graphical User Interfaces
- Apply good practice in code design/testing, and develop the necessary transferable skills in a group work environment

The assignment is worth ***100%*** of the overall mark for the module.

Broadly speaking, the assignment requires you to design and implement a moderately realistic object-oriented system. You will produce detailed object models and designs from system requirements; use the modelling concepts provided by UML. You will then map the designs into code and perform unit testing using an automated testing tool.

The assignment is described in more detail in section 2.

This is a **GROUP** assignment.

Working on this assignment will help you to understand more clearly the concepts, problems, and techniques of object-oriented systems, and how these can be used to design and implement a reasonably large size software systems while working in a team (to a professional standard).

If you have questions about this assignment, please discuss with your lab tutors.

Section 2: Task Specification

All the tasks are based on the following scenario. Please read carefully all information contained within the following passage of text.

UWE Bristol Accommodation System

The main responsibility of the UWE accommodation office is to provide the necessary help for all the registered students who are entitled and require their accommodation on-campus. The on-campus accommodation comprises of many halls of residence, and each hall has a number of rooms. The accommodation office has a hall manager who supervises the operation of the halls. Each hall has a warden who oversees the regular cleaning and maintenance of all the rooms in that hall. Each hall of residence has a name, number, address, telephone number. The halls provide only single rooms which have a room number and monthly rent rate. The total number of rooms provided by the accommodation office should also be available. The hall number uniquely identifies each room in all of the halls controlled by the accommodation office, and is used when renting a room to a student.

The UWE Bristol Accommodation Services allow students renting rooms for the entire 12-month academic year from September to August. Each individual rental agreement between a student and the accommodation office is uniquely identified using a lease number. The data stored on each lease includes the lease number, duration of the lease (in months), address details of the hall, room number, student's name and ID number.

The room scheduling is coordinated by the hall manager. That is, the hall manager generates and maintains the room schedule, keeps a record of all the students staying in the halls and reviews applications for future bookings. The hall manager can edit and view room details. The view should show all the detailed information about a single room in a hall, including its room number, room status (namely, occupied or unoccupied), monthly rent rate and other details describing the room. The hall manager should be able to determine from this description whether a particular room is available and its suitability for a new student to occupy. On the other hand, a warden can view the room details but he has limited editing privileges and can only change a room's cleaning status which can be "clean", "dirty", or "off-line". An off-line room is one that cannot be occupied because it requires maintenance beyond a normal cleaning.

You are the leader of a team of three developers who have been asked to design and implement a system which enables the hall manager and wardens to schedule hall activities and keep track of the hall rooms in a simpler and easier way, occupied and unoccupied, enforcing the rules described above.

Section 3: Deliverables

You are asked to produce the following deliverables for the scenario given above.

Part-I

1. Produce a use case diagram to capture the functionality for the system to be built.
2. Produce a class diagram to meet all the requirements captured in the use case diagram.
3. Produce at least three sequence diagrams for some use cases that adds information.

Part-II

4. Write a joint report outlining the typical steps that you would follow to develop the system in relation to an Agile development episode. You may consider the following steps:

Strategy planning: This is the initial phase for our system development process where we as developers need to identify system users to avoid building wrong solution………

Continuous team iterations: The development phases of our project are more flexible as we continuously iterate between planning and implementation compare to conventional development methodology which is too rigid and strict……

Team communication: The essence of our project is effective communication among members either through face-to-face or online conversation…….

Simplicity: Our main objective is to practice simplicity which is important for the system to avoid the structure being too complicated……

Work Plan (Not mandatory but if they can produce would be good): Project Backlog, SPRINT Cycle

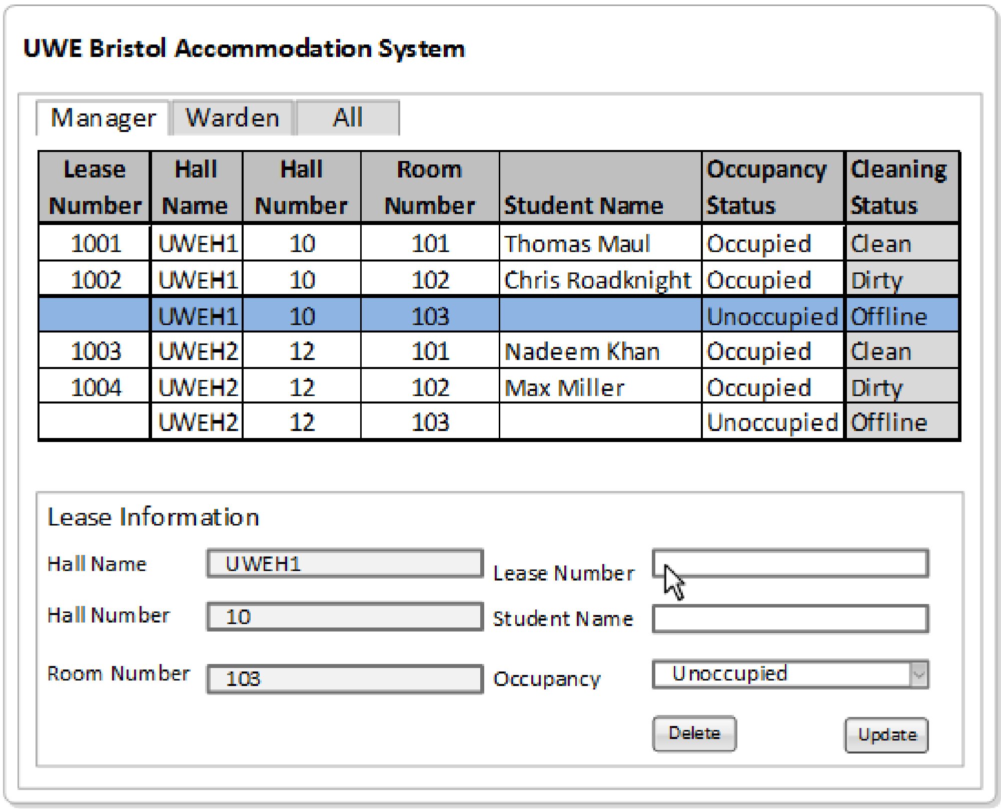
5. Develop the system using Python considering all the functionalities described above.

   You should be creative and come up with your own User Interface Design (the design below shows just an example).

   A user interface that lets

i) the manager to view and edit room details. For example, assume that there are 2 halls, each having 3 rooms, and rooms are either occupied or off-line at the moment.

When the manager selects and edits a particular room, GUI will look like the following (you can add more fields if you like, e.g., student ID, lease duration etc.):



UWE Bristol Accommodation System

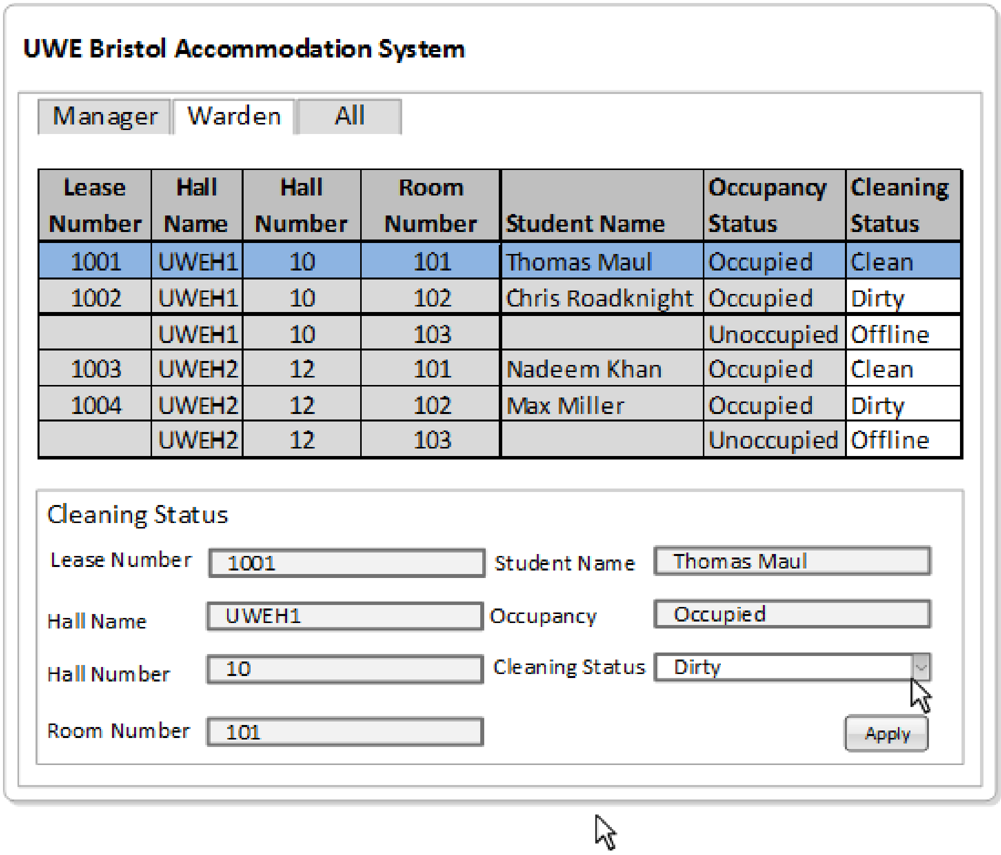
Lease Number	Hall Name	Hall Number	Room Number	Student Name	Occupancy Status	Cleaning Status
1001	UWEH1	10	101	Thomas Maul	Occupied	Clean
1002	UWEH1	10	102	Chris Roadknight	Occupied	Dirty
	UWEH1	10	103		Unoccupied	Offline
1003	UWEH2	12	101	Nadeem Khan	Occupied	Clean
1004	UWEH2	12	102	Max Miller	Occupied	Dirty
	UWEH2	12	103		Unoccupied	Offline

Lease Information

Hall Name	UWEH1	Lease Number	
Hall Number	10	Student Name	
Room Number	103	Occupancy	Unoccupied

He can then enter the required information and update accordingly. Please note that in the above selection, the manager cannot create a lease because the room is currently "off-line". A lease can only be created if the occupancy status is "unoccupied" and cleaning status is eithr "clean" or "dirty". Similarly, the manager can select and delete a lease. This will make the lease number and student name fields null/empty and the occupancy status as "unoccupied". However, the other fileds will remain the same.

i) a warden to view and edit room details. The view and edit should be similar to the above; however, a warder can only change the cleaning status of a room and cannot delete a lease. For example, he can select and change the cleaning status from "clean" to "dirty" of the lease 1001.



UWE Bristol Accommodation System

Lease Number	Hall Name	Hall Number	Room Number	Student Name	Occupancy Status	Cleaning Status
1001	UWEH1	10	101	Thomas Maul	Occupied	Clean
1002	UWEH1	10	102	Chris Roadknight	Occupied	Dirty
	UWEH1	10	103		Unoccupied	Offline
1003	UWEH2	12	101	Nadeem Khan	Occupied	Clean
1004	UWEH2	12	102	Max Miller	Occupied	Dirty
	UWEH2	12	103		Unoccupied	Offline

Cleaning Status

Lease Number: 1001
Student Name: Thomas Maul
Hall Name: UWEH1
Occupancy: Occupied
Hall Number: 10
Cleaning Status: Dirty
Room Number: 101

ii) (**Optional , but highly encouraged**) for the case of "All" as is labelled it on the GUI design, you may wish to consider this view contains a superset of the other views and allows a user to perform all actions.

6. Testing is typically a part of the program development – you should use a test strategy to test your system thoroughly. You should identify all the suitable test cases for all the classes implemented. When you test your code, you should make sure that your program does not allow bad data to be stored into your objects. Deliberately feed in your program out of range or wrong data and try to make it fail, see if you can swipe bad values into the member variables. One example test case shown below.

Test Case	**Purpose**	**Expected result**
occupancyStatus is occupied	To create a lease, occupencyStatus must be unoccupied	Appropriate error message

7. Individual Contribution and Reflective Report (500-700 word). Your report should include (a) the evidence of your work produced for the above activity with an outline and (b) reflection on your contribution to the work in relation to skills and knowledge acquired.

Section 4: Marking Criteria

Uploaded separately.

Section 5: Feedback mechanisms

On the spot verbal feedback during the demo session + as appropriate written feedback uploaded to Blackboard. Formative feedback provided in Part-I will be useful to complete Part-II of the assignment.

This Page is Intentionally Left Blank