UNIVERSITY NAME
Faculty Name
Department Name

Course Code and Name
Year - Semester

Course website address

Assignment/Project
Reserve Hotel

Due: Date & Time

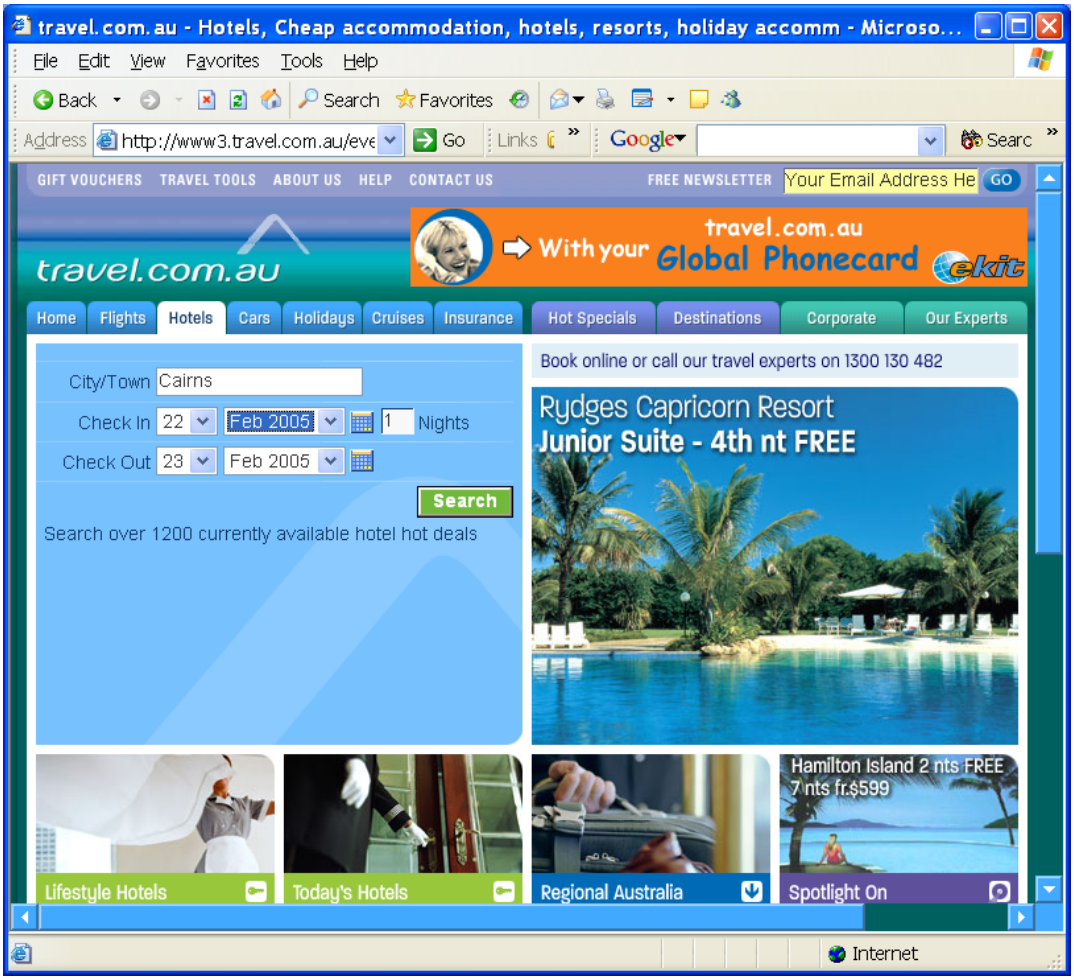
Number of marks

Application domain for the assignment

This is a system modeling assignment, with particular emphasis placed on the database and on the application logic. The problem domain for the assignment is a web application for hotel reservations. The application is called "Reserve Hotel". The scope of "Reserve Hotel" application is determined from the functionality available in an existing system by travel.com.au – Australia's leading on-line travel company. Students should look at http://www3.travel.com.au/hotels.html for details. The next section shows some screenshots taken from the travel.com.au portal. The screenshots are annotated to define functional requirements and constraints for "Reserve Hotel".

Screenshot 1 demonstrates that "Reserve Hotel" is part of a larger web-based solution for all sorts of travel reservations and arrangements. From the viewpoint of "Reserve Hotel" these should ~~be treated as external functions with no dependencies~~ to "Reserve Hotel". Screenshot 1 demonstrates also that the online traveler/customer is asked to type in the destination city, select the check in date and *either* specify the number of nights or select the check out date. The Search button allows to submit the request to the system to find available hotels on the dates specified.

If the City/Town location does not exist in the database, the application offers the traveler to find the location by browsing through the selections, as shown in *Screenshot 2*.

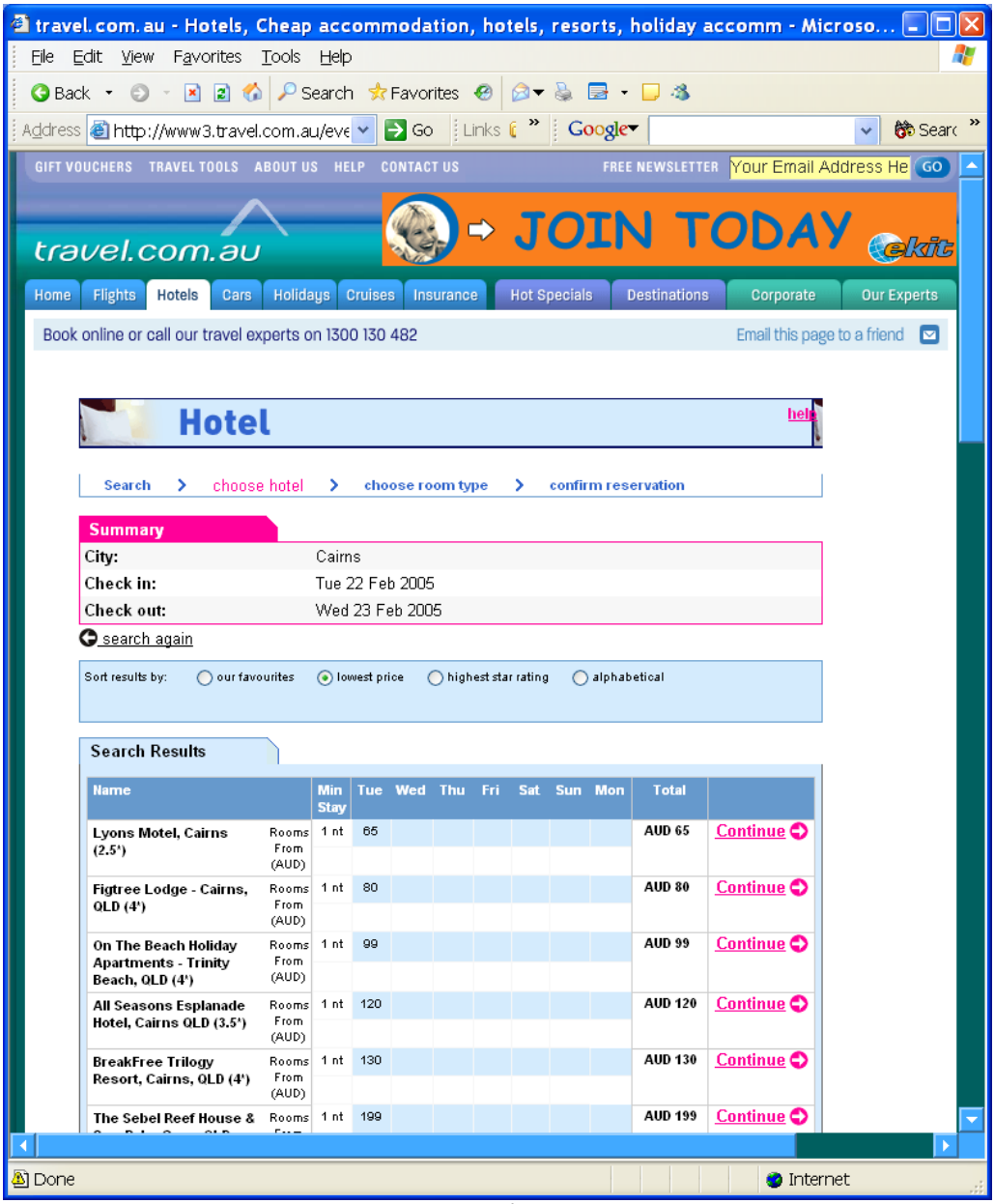


Screenshot 1

Browse from our selections

Capital Cities	Regional Australia	Our Favourites	Rest of World
Sydney	NSW Regional	New Zealand	Bali
Melbourne	VIC Regional	Hotels Under $100	Thailand
Brisbane	QLD Regional	2 hrs from Sydney	Hong Kong
Adelaide	SA Regional	2 hrs from Melbourne	Singapore
Perth	WA Regional	2 hrs from Brisbane	Malaysia
Canberra	NT Regional	Designer Hotels	Asia
Darwin	TAS Regional	B&B's	USA
Hobart		Romantic Escapes	United Kingdom
		Islands	Europe
		Tropical North QLD	Worldwide
		Gold Coast	UK B&Bs
		Beach Houses	Ireland
		Breakfree Resorts	
		Pacific Islands	

Screenshot 2



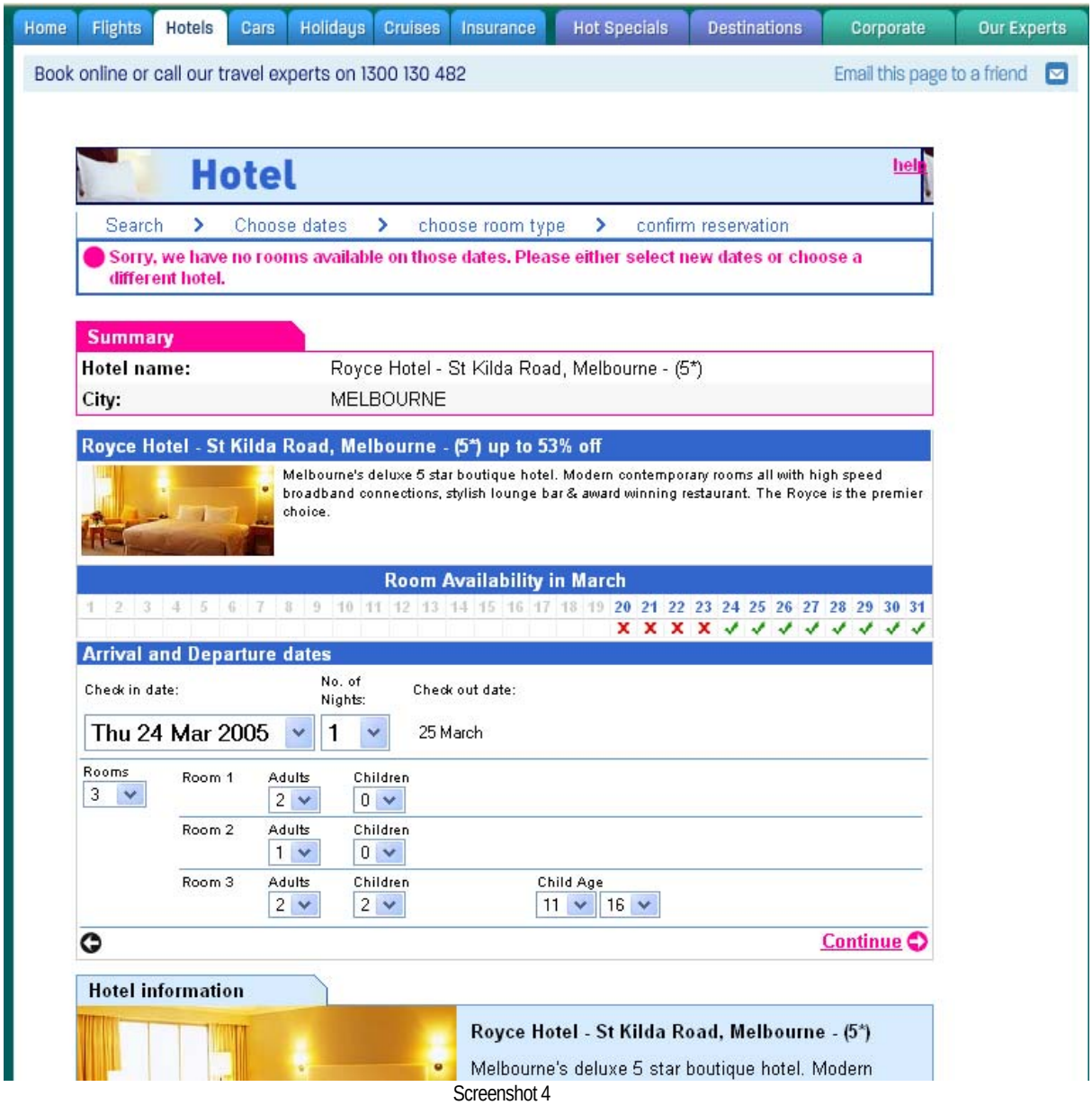
Screenshot 3

Screenshot 3 shows a web page resulting from a successful search. The page shows:

- *workflow* that informs that making a reservation consists of four steps: (1) search, (2) choose hotel, (3) choose room type, and (4) confirm reservation
- *summary* that displays the original search criteria
- *search again* option, which returns the customer to Screenshot 1
- option to *sort results by*: (1) our favourites, (2) lowest price, (3) highest star rating, and (2) alphabetical
- *search results* listing the hotels and basic information about them, as well as the *continue* option to proceed with the reservation

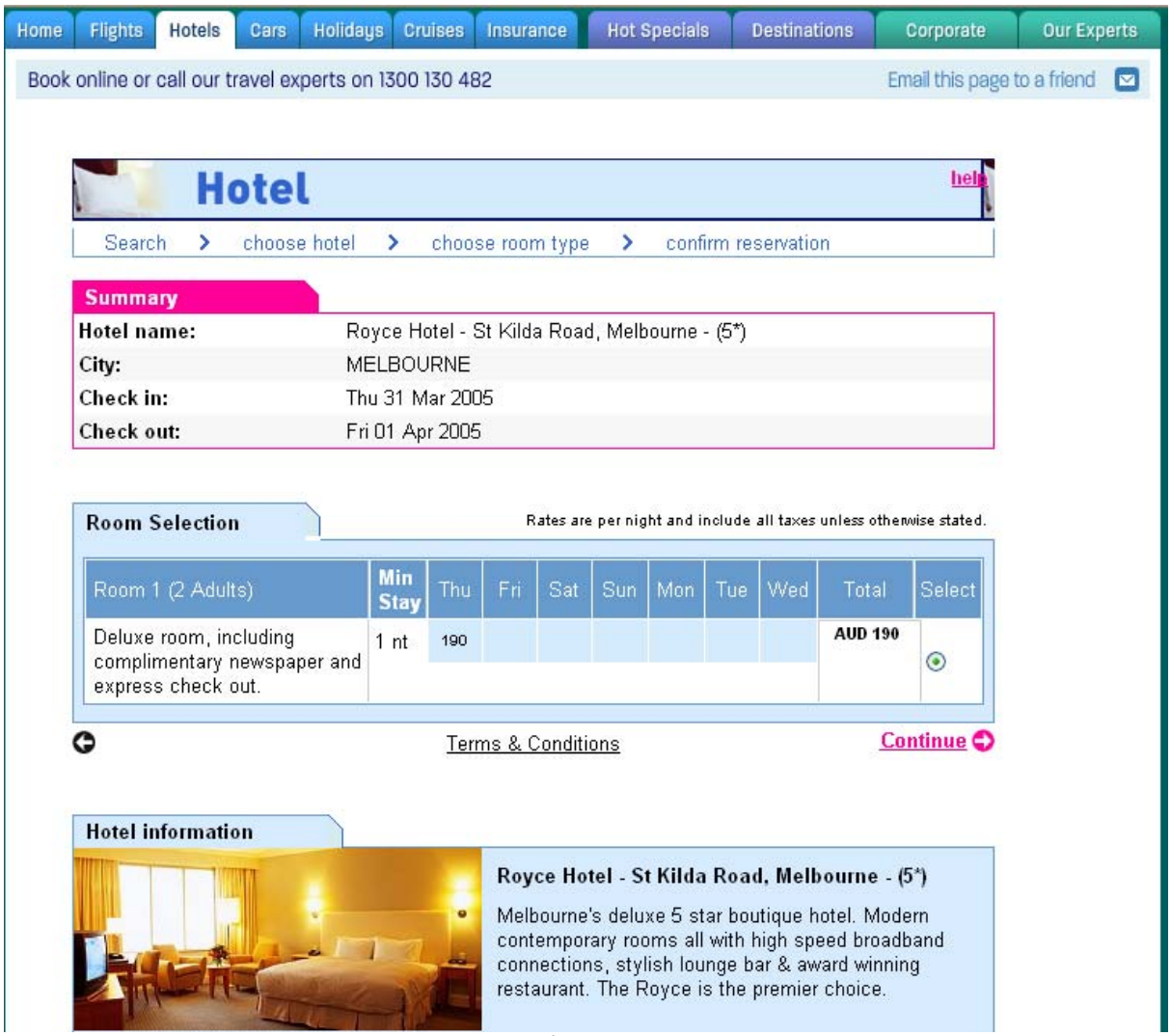
Screenshot 4 shows a web page resulting from pressing a Continue button in Screenshot 3. The page informs the customer about the room availability (if *some* rooms are available) and allows the customer to modify the check in date and the number of nights. It also invites the customer to specify required rooms and the number of adults and children in each room.

After pressing Continue in Screenshot 4, the system verifies if the exact reservation request can be met by the hotel and, if not, the system displays an informational message as shown in the upper part of Screenshot 4 and the system displays the same web page.



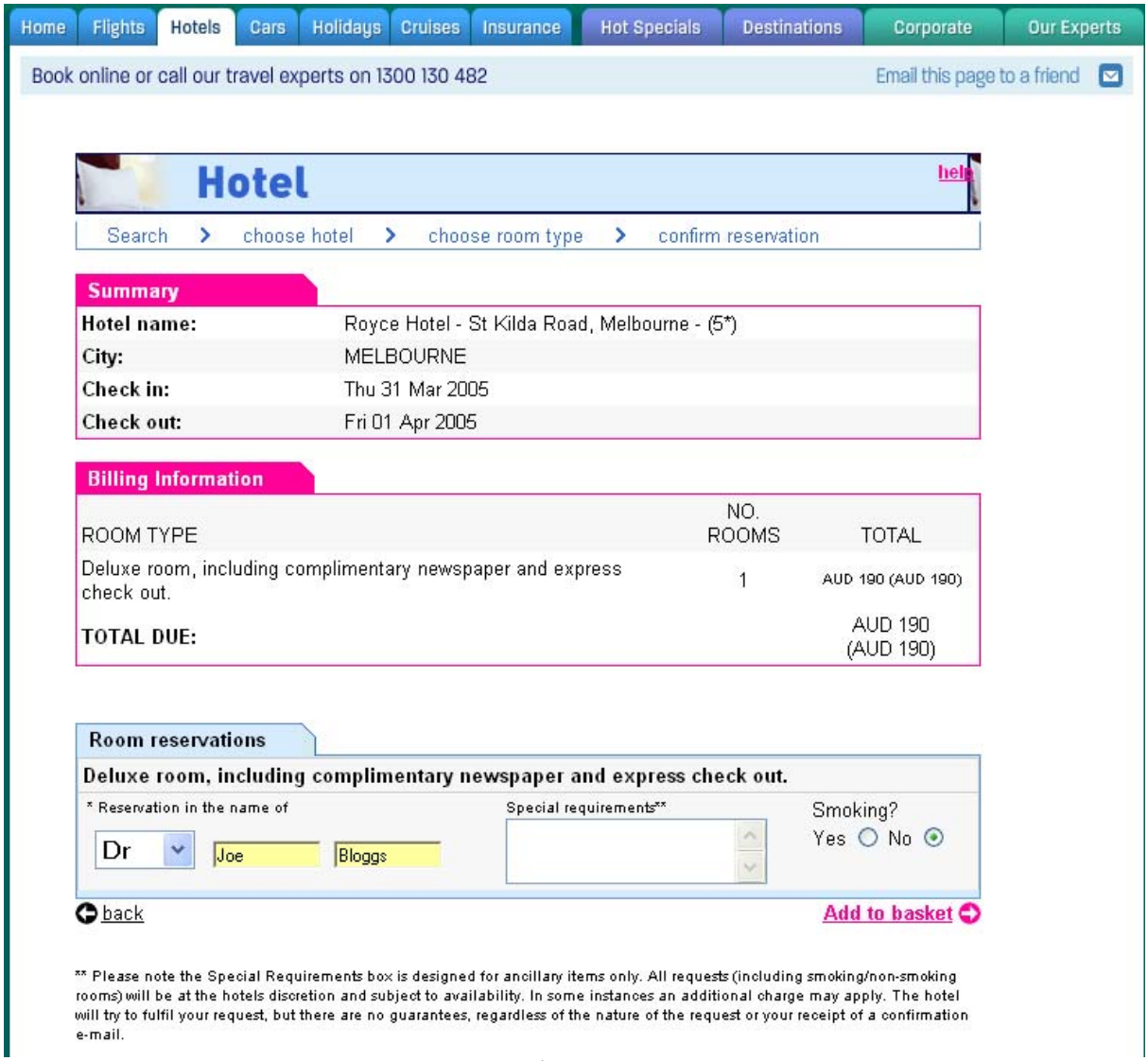
Screenshot 4

If the reservation is possible to be made, the system displays room selection information that contains the price, as in *Screenshot 5*.

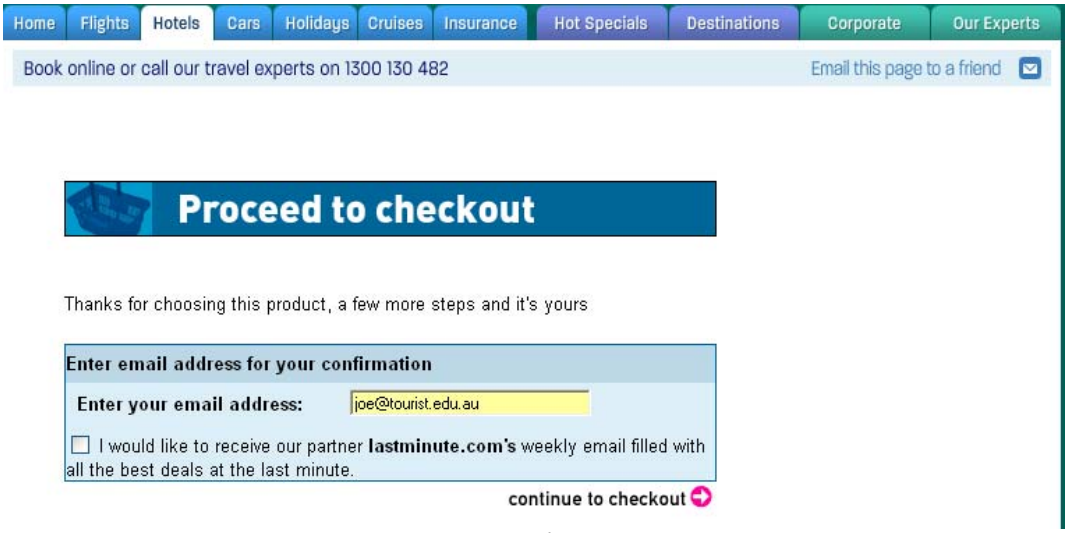


Screenshot 5

Screenshot 5 shows a web page with all the reservation information just prior to placing the reservation in the customer's shopping basket. To add the reservation to the basket, the customer must provide his/her full name and email address, shown in *Screnshots 6 and 7*.

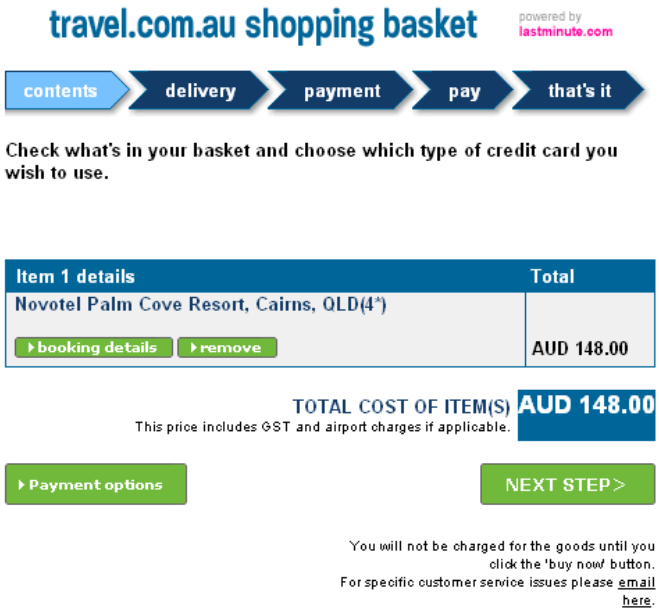


Screenshot 6



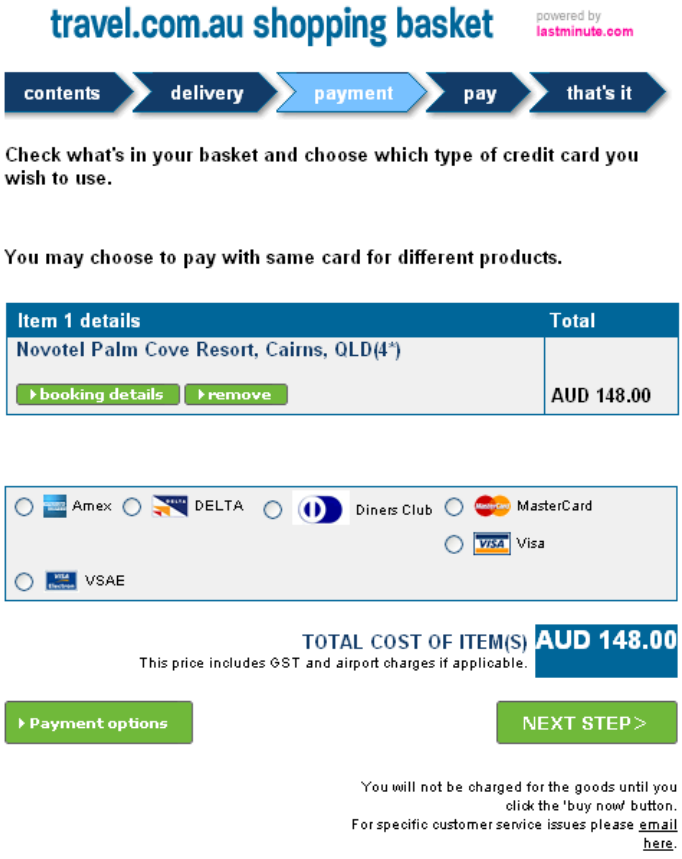
Screenshot 7

Screenshot 8 shows the contents of the shopping basket. Prior to proceeding to the *Next Step*, the customer can ask for *booking details* and for *payment options* or can *remove* the item from the basket.



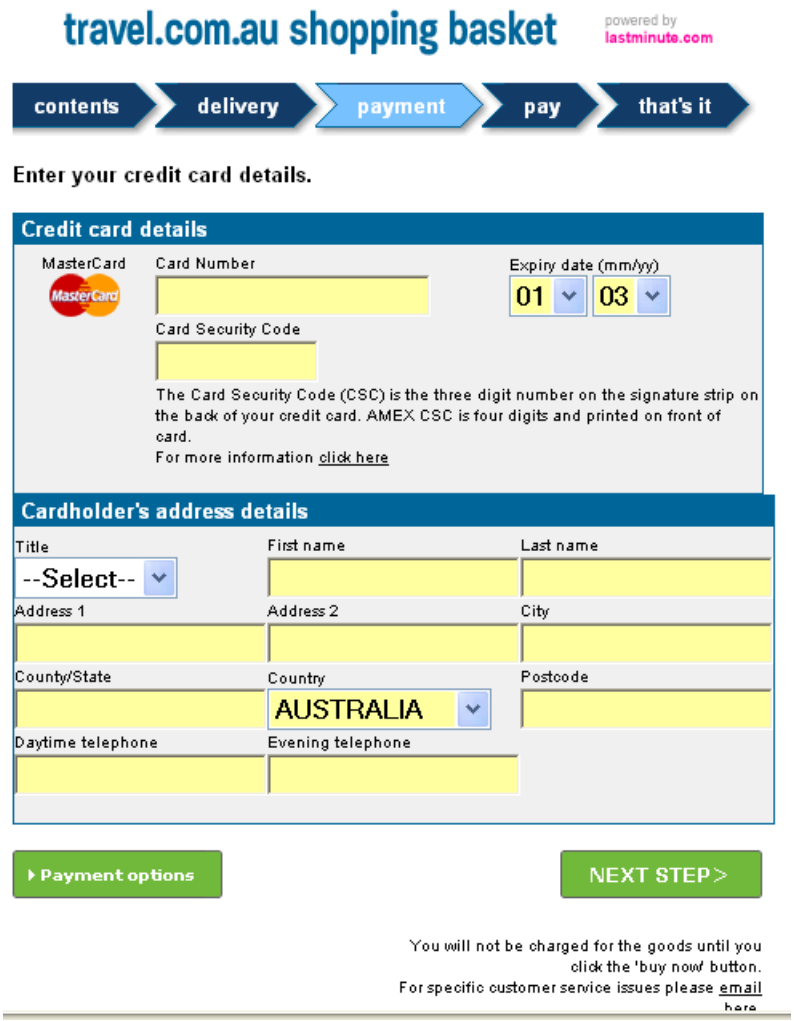
Screenshot 8

Screenshot 9 presents the customer with the selection of credit cards acceptable for making a payment.



Screenshot 9

Screenshot 10 shows the form that the customer needs to fill in and submit as the payment in order to secure the reservation. Once the payment is made, "Reserve Hotel" will email the customer the confirmation of the payment and the reservation.



Screenshot 10

Assignment tasks

Task 1 (? marks)

Develop a detailed use case model for "Reserve Hotel". The model should take a user's (rather than system's) perspective. Hence, for example, "Acknowledge Basket Contents" rather than "Display Basket Contents". For the same reason, define use cases for handling user's input and actions, but do not be specific about system's computations and web page creation activities.

Show dependencies between use cases to visualize the workflow-like behavior of the application. Specify «include» and «extend» relationships and any important constraints.

Answer to Task 1

…

Task 2 (? marks)

Develop a class model for business objects and business logic in "Reserve Hotel". Note that according to the PCMEF meta-architecture, the business objects belong to the entity subsystem and the business logic to the mediator subsystem. As per the PCMEF class naming principle, name your entity classes starting with letter E, and the mediator classes starting with letter M. A related way of distinguishing between entity and mediator classes is to assume that all entity classes will persist in the database, whereas most, or even all, mediator classes are transient. Hence, for example, `MShoppingBasket` but `ECreditCard` (because there is no need to store shopping basket information in the database, but the provided credit card details will be kept in the database whether or not the successive payment is successful).

The class model should show attributes in classes and relationships between classes. Associations should be used to "legitimize" in the static structures all important dynamic communication paths between classes. This includes associations between mediator and entity classes. All associations should have multiplicities carefully defined. There is no need to specify attribute types and operations (methods).

Answer to Task 2

…

Task 3 (? marks)

Use your class diagram from Task 2 and make the CASE tool to automatically transform it to a relational database schema (targeting Oracle). Transform only the entity classes and relationships between them. Discuss the obtained relational diagram. Identify the transformation strengths and weaknesses.

Answer to Task 3

…

Task 4 (? marks)

Use your class diagram from Task 2 and manually transform it to a relational database (targeting Oracle). For class hierarchies use the approach of creating a table for each class.

Transform only the entity classes and relationships between them. You can provide a relational design diagram in any notation you like, but ultimately you need to write CREATE TABLE statements for the schema.

Identify any assumptions made in the process of mapping. Specify the constraints for all referential integrity relationships. You do not need to write CREATE ASSERTION statements. Just write short descriptions of what assertions would be needed on insert, update, and delete actions on the tables For update and delete decide on one of typical actions: restrict, cascade, set null. Determine also for all relationships if "change parent allowed" and if "parent mandatory". For example, in the case of Department – Employee relationship, we may say "change parent allowed" if an employee can "switch" from one department to another and we may say "parent not mandatory" if some employees may not be assigned to any department. Note that "parent mandatory" has a consequence on the insert assertions.

Use triggers to enforce constraints. Write a few CREATE TRIGGER statements for these assertions that enforce substantially different constraints (i.e. no need to write "duplicate" triggers that implement similar assertions).

Answer to Task 4

…

Task 5 (? marks)

Refer to Screenshots 1 and 3 and to your answers to previous questions. Consider that the customer has filled in the form in Screenshot 1 and pressed the Search button. Assume that the data has been entered correctly, including a City/Town name that exists in the database.

Design a sequence diagram necessary to accomplish the Search function. Follow the PCMEF meta-architecture. Refer also to Chapter 15 of the PSE ("Practical Software Engineering") book (if you do not have the book, you can download Chapter 15 from http://www.comp.mq.edu.au/books/pse/about_book/PSE_sampleChapters.htm).

Assume that:

- the processing starts by a HTTP PUT request from the form on the web page in Screenshot 1
- there is only one control class to perform user actions, which is implemented as a servlet
- the page in Screenshot 3 is implemented as a JSP
- there is no need to show any JavaBeans as the vehicle for transporting search results data from the servlet to the JSP page

- there is only one data mapper class and one identity map class (rather than separate mappers and identity maps to represent individual entity classes)
- the data searched for is not in the cache and the program will need to search for data in the database
- you can use return messages to represent return types of messages
- there is no need to model the behavior of "lazy load" and "unit of work"

Answer to Task 5

…