

COMP9120 Database Management Systems**Assignment 2: Database Application Development****Group Assignment (12%)****Introduction**

The objectives of this assignment are to gain practical experience in interacting with a relational database management system using an Application Programming Interface (API) (Python DB-API). This assignment additionally provides an opportunity to use more advanced features of a database such as functions.

This is a group assignment for teams of about 3 members, and it is assumed that you will continue in your Assignment 1 group. You should inform the unit coordinator as soon as possible if you wish to change groups.

Please also keep an eye on your email and any announcements that may be made on Ed.

Submission Details

The final submission of your database application is due at 11:59pm on Friday 28th May. You should submit the *items for submission* (detailed below) via Canvas.

Items for submission

Please submit your solution to Assignment 2 in the 'Assignment' section of the unit's Canvas site by the deadline, including EXACTLY TWO files:

- A SQL file (**SETSchema.sql**) containing the SQL statements necessary to generate the database schema and sample data. This should contain the original schema and insert SQL statements, and any changes or additions you may have made.
- A Python file (**database.py**) containing the Python code you have written to access the database.

Section 1: Introducing the Sydney Entertainment & Theatres (SET) System

In this assignment, you will be working with the SET System which is currently under development. The system still requires work in numerous areas, including interaction with the database. Your main task in this assignment is to handle requests for reads and writes to the database coming from the user interface (UI). We first describe the main features that the SET System should include from a UI perspective, and then discuss where the majority of your database code needs to be implemented.

Logging In

The first form a user is presented with when starting the SET system is Login, as shown in Figure 1. This feature is still under development and currently requires that a user (sales agent) enters their username and password to be validated prior to successfully log in to the system. Security features such as password encryption/hashing will be implemented at a later stage (and not as part of this assignment). Once logged in, the user is taken to the Booking List page to view their associated customer bookings for the various theatre performances.

Log in

Log in

Figure 1 – Login form

Viewing Booking List

Once a user has logged in, they are shown a list of all their associated bookings, and their details, as shown below in Figure 2. This booking list must be ordered by customer's full name. Each booking may or may not have an assisting sales agent, and can also have a customer name, a performance name, performance date, and a description of booking instruction. A booking is associated with a sales agent if the sales agent is recorded as the user assisting the customer with his/her theatre performance booking.

Sydney Entertainment & Theatres System				Bookings	Add Booking	Logout
Performance Booking List						
Search Booking				Find		
Booking No	Customer	Performance	Performance Date	Booking Agent	Booking Instruction	
1	Bob Smith	The Lion King	2021-06-05	Novak Djokovic	I'd like to book 3 additional seats	
5	Mia Clark	Disney's Frozen	2021-07-18	Novak Djokovic	Please upgrade my seats to Box Seats	
3	Ruby Miller	Death of a Salesman	2021-06-27	Novak Djokovic	I want to add meals to my booking	

Figure 2 – Viewing Booking List

Finding Bookings

A user can search through all bookings by entering a word or phrase (a 'keyword') in the field next to the Find button, as shown in Figure 3, and then clicking on Find. When such a keyword is specified, then only bookings including this word or phrase in their customer full names, booking agent full names, or performances will be retrieved and shown in the list. The search must ignore case sensitivity. For example, given the search keyword 'lion', Find will return all bookings that include the word 'lion' in their customer full names, booking agent full names, or performances. Searching with a blank/empty keyword field will show all of the logged in user's associated bookings. Any search results returned must be ordered by customer's full name.

Sydney Entertainment & Theatres System

BookingsAdd BookingLogout

Performance Booking List

lion

Find

Booking No	Customer	Performance	Performance Date	Booking Agent	Booking Instruction
1	Bob Smith	The Lion King	2021-06-05	Novak Djokovic	I'd like to book 3 additional seats
4	Peter Wood	The Lion King	2021-06-05	Jeff Alexander	Can you please waitlist 4 seats?

Figure 3 – Finding Bookings

Adding a Booking

Users may also add a new booking by clicking on the *Add Booking* tab in the title bar, entering booking details in the popup dialog that appears, and then clicking on Add Booking button. User must enter a dash ('-') as the Booking Agent if there is no agent associated with that booking.

Sydney Entertainment & Theatres System

BookingsAdd BookingLogout

New Booking

Customer Email:

Performance:

Performance Date:

dd/mm/yyyy

Booking Agent:

Booking Instruction:

Add Booking

Figure 4 – Adding a Booking

Updating a Booking

Users can also update a booking by modifying data in the booking details screen as shown in Figure 5, and clicking on Update Booking button. You can access this update screen by clicking on a booking from the list of bookings in the Bookings tab.


Sydney Entertainment & Theatres System		Bookings	Add Booking	Logout
--	--	----------	-------------	--------

Update Booking

Booking No:

Customer:

Performance:

Performance Date: 

Booked By:

Booking Instruction:

I want to add meals to my booking

Figure 5 – Updating a Booking

Database Interaction Code

The files that are needed for the Python version of assignment are as follows:

1. **SETSchema.sql**: a file which contains SQL statements you need to run to create and initialise the SET system database, before starting the application
https://canvas.sydney.edu.au/files/15805311/download?download_frd=1
2. **Assignment2_PythonSkeleton.zip**: a zip file encapsulating the Python project for the SET system
https://canvas.sydney.edu.au/files/15937292/download?download_frd=1

To inspect the SET system code, you need to unzip the ZIP archive first, which will create a folder that includes the name **Assignment2_PythonSkeleton**. If you experience any difficulties installing and exploring the project, ask your tutor or lecturer for assistance.

The skeleton code uses a number of Python modules to implement a simple browser-based GUI for the SET system. The main modules are the Flask framework for the GUI and the pycpg2 module for the PostgreSQL database access. Similar to tutorial 7, **you will need to install the Pycpg2 module and the Flask module**. The skeleton code follows the structure described below:

- The main program starts in the **main.py** file. You need to use the correct username/password details as specified in tutorial 7, and then implement the missing database access functions – including any necessary SQL code statements required – in the data layer **database.py**.
- The presentation layer is done via a simple HTML interface that can be accessed from a web browser. The corresponding page templates are located in the **templates/** subdirectory, their CSS style file is **static/css**.
- The transition between the different GUI pages and the initialisation of the Flask framework is done in the **routes.py** file. It currently just invokes the pages, but there is no further business logic implemented yet.

You can run the code by running **"python main.py"**. This starts a local web server and prints out some

debug messages in the terminal; the GUI can then be accessed with any web browser on the same computer via the local URL <http://127.0.0.1:5000/> (If that doesn't work you can also try <http://0.0.0.0:5000/>). Please note that, to terminate the application, you will need to stop the local web server which is running in the background.

Section 2: Your Task

Core Functionality

In this assignment, you are provided with a Python skeleton project that must serve as the starting point for your assignment. Your task is to provide a complete implementation for the file `database.py`, as well as make any modifications necessary to the database schema (i.e., `SETSchema.sql`). Specifically, you need to modify and complete these functions:

1. `checkUserCredentials` (for login)
2. `findBookingsBySalesAgent` (for viewing booking list)
3. `findBookingsByCustomerAgentPerformance` (for finding bookings)
4. `addBooking` (for adding booking)
5. `updateBooking` (for updating booking)

Note that, for each function, the corresponding action should be implemented by issuing SQL queries to the database management system. If you directly output the result without issuing SQL queries, you are considered as cheating, and you will get zero point for the assignment.

Marking

This assignment is worth 12% of your final grade for the unit of study. Your group's submission will be marked in line with the rubric that follows.

Group member participation

If members of your group do not contribute sufficiently, you should alert the unit coordinator as soon as possible. The course instructor has the discretion to scale the group's mark for each member as follows:

Level of contribution	Proportion of final grade received
No participation.	0%
Full understanding of the submitted work.	50%
Minor contributor to the group's submission.	75%
Major contributor to the group's submission.	100%

Marking Rubric

Your submissions will be marked according to the following rubric, with a maximum possible score of 12 points.

	Part marks (0 – 1)	Full marks (1.5 – 2)
Login	Can correctly login the user 'novak' and validate his username and password.	All valid users can be logged in successfully, and unsuccessful user logins should be rejected.
View Booking List	Correctly list all bookings associated with user 'novak' in the correct order (see Figure 2).	Correctly list all bookings associated with a user, in the correct order, for all possible username input from Figure 1.
Find Booking	Correctly list bookings for keyword "lion" (see Figure 3).	Correctly list bookings for all possible keywords.
Add Booking	Can correctly add a booking ('abrown87@outlook.com', 'The Lion King', '04/07/2021', 'novak', 'Please book 5 house seats') to the database.	Can correctly add all valid bookings to the database. Bookings associated with invalid customer or sales agent should be rejected.
Update Booking	Can correctly update the instruction of a booking as shown in Figure 5.	Can correctly update details of all bookings, ensuring the associated sales agents are valid.
Stored Procedure	A stored procedure is correctly created in the submitted SQL file.	A stored procedure is correctly created in the submitted SQL file, and is correctly called in one of the specified functions.