COMP 636: Python and Database Assessment

Due: **5pm, Monday 31 October 2022**

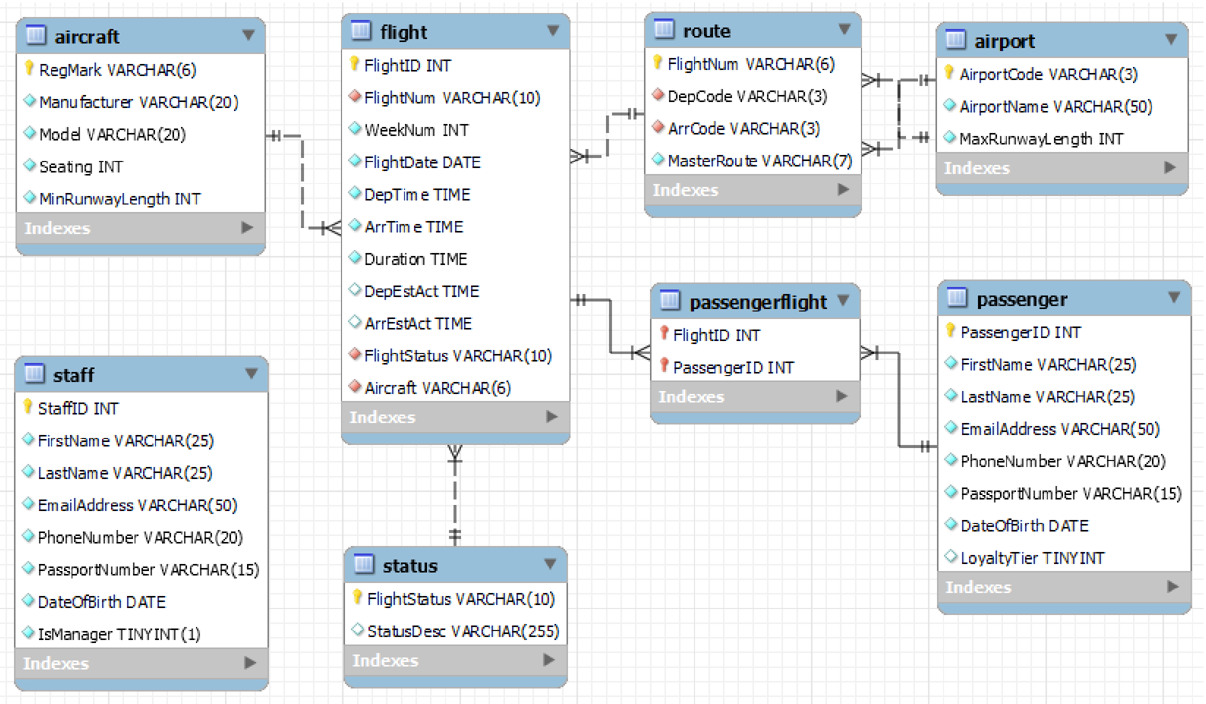
Worth **45%** of COMP636 grade

Submit via Akoraka | Learn

Instructions

This assessment develops an airline flight management web application for Air Whakatū based out of Nelson, New Zealand. It manages customer bookings, flights, routes, airports and aircraft.

Data model



Foreign keys:

Parent table and field(s)		Child table and field(s)
airport.AirportCode	—<	route.DepCode
airport.AirportCode	—<	route.ArrCode
aircraft.RegMark	—<	flight.Aircraft
route.FlightNum	—<	flight.FlightNum
status.FlightStatus	—<	flight.FlightStatus
flight.FlightID	—<	passengerFlight.FlightID
passenger.PassengerID	—<	passengerFlight.PassengerID

Requirements

The website consists of two main parts: a public area for customers to look up flight details and book flights, and an administration area for staff to add and edit flight details, bookings and passenger information.

(Note that the marks shown for each question are indicative only, to give an idea of what each part is worth, but the actual values may change slightly in marking.)

General requirements (10 marks):

1. **General functional requirements (6 marks):** General quality of Python code, SQL, HTML, format and output formatting. (See Marking schedule below for details).
2. **Page header and footer template (4 marks):** The Air Whakatū logo must be displayed at the top of every page and the contact details in the footer of every page
3. **System 'current time':** To make submissions easier to compare on the same basis, the default 'current time' for the system is assumed to be **Friday 28 October 2022, 5:00 pm** and should be hard-coded into a global variable called timeNow (although, try changing it other times as well to test the system). This affects flight listings which are relative to current time. It ensures that everyone's systems will be the same when marked. In a real system, this would be the current system time.

Public system for customers (25 marks):

4. **Home page (1 mark):** This is the landing page for Air Whakatū – when a user arrives at the website, this is the first page they see. It includes:
    a. Air Whakatū logo and contact details
    b. Links to Arrivals & Departures screen and Booking page
5. **Airport arrivals and departures (3 marks):** Shows appropriate arrivals and departures information for a selected airport from 2 days before the current day to 5 days after the current day.
6. **Customer log-in (4 marks):** Customers must log in to book a flight and change details:
    a. The passenger must login, using email address as user ID (no password is required).
    b. New passengers can register from this page.
    c. If the email address is not found, the system returns to the booking login page with an appropriate message and a suggestion to register if a new passenger.
7. **Register new passenger (3 marks):** New passenger can add their details.
8. **Passenger bookings list (3 marks):** Once logged in, the user can see their existing bookings (sorted by date-time), add a new booking, change their passenger details or log out.
9. **Edit passenger details (4 marks):** Once logged in, the user can edit their own passenger details (but not passenger ID).
10. **Cancel bookings (1 mark):** Existing bookings can be cancelled, by removing the passenger from the flight
11. **Available flights (5 marks):** User selects departure airport. All flights from that airport are displayed for the selected date and 7 days after that date, including the number of seats available on that flight. A cancel ('back') option is available to take the customer back to the previous page.
    a. Only future flights are shown.

    b. Flights that are not available to be booked are still displayed, with a description of why they are not available. Flights are not available when the flight is fully booked (i.e., the aircraft seating capacity has been filled) or the flight has been cancelled.

12. **New booking created (1 mark):** Once a flight is selected, the passenger can be added to that flight, with a confirmation screen and option to cancel.

Administrative system for staff (35 marks):

13. **Admin route (1 marks):** The admin home page is reached by the user typing /admin on the end of the public home page address. All admin page addresses should begin with /admin (e.g., http://127.0.0.1:5000/admin/someURL).
14. **Staff login (4 marks):** The staff member logs in by selecting their name from a list. No password is required for this system. The appropriate manager/staff permissions are passed to other pages. Staff details are stored in the `staff` table. These details do not need to be edited for this assessment. Most important is the `IsManager` field, which indicates whether the staff member is a manager or not.
15. **Passenger list (5 marks):** All staff can see an alphabetical list of all passengers (by last name then first name), with a search/ filter capability on passenger last name. They can click on a passenger in the list to edit their details or bookings.
16. **Add/edit passenger details (3 marks):** All staff can add or edit passenger details for any passenger. Deletion of passengers is not required.
17. **Add/edit bookings (3 marks):** Staff can make, change or cancel a booking for any passenger.
18. **Flight list (5 marks):** All staff can see a list of all flights, sorted by date, time and departure airport. This includes the aircraft reg. mark, the number of seats booked for each flight and the available seats remaining. The list can be filtered by date range, departure or arrival airport (or a combination of those). By default, the flight list shows flights to and from all airports up to 7 days from current time. Clicking on flight shows the flight manifest.
19. **Flight manifest (5 marks):** For each flight, all staff can see flight details, seating capacity and a list of all of the passengers on a flight. They can click the passenger details to edit passenger details or bookings (in the same way as the passenger list described above). Passengers are listed in alphabetical order by last name then first name. The list is numbered in order – showing 1 for the first in the list, then 2, 3, and so on.
20. **Add flights (4 marks):** Only managers can add flights. All flights from the latest week can be copied into the next week, using the query provided (`Duplicate latest week flights.sql`). Managers can also add new individual flights. For a new flight, duration is calculated automatically from the difference between scheduled departure time and scheduled arrival time. Estimated/actual times will match the scheduled times. Default status is always 'On time'. Aircraft are selected from a dropdown box, which lists them in alphabetical order by RegMark. (There is no requirement to check that an aircraft is available and not double booked.)
21. **Edit flight details (5 marks):** From the flight list, managers can edit all details of a flight, while other staff can only edit the status and the estimated/actual times.
    a. The status is selected from a dropdown box, which lists the options in alphabetical order, where the current status is displayed in the dropdown box. If status is set to Cancelled, then the estimated/actual times are set to null.

b. Aircraft for a flight (only updatable by managers) is selected from a dropdown box, which is in alphabetical order by RegMark, in the same way as for a new flight (above), but the currently assigned aircraft is shown in the dropdown box.
c. There is no requirement to delete flights once they are added. They can be either cancelled or edited.

Not required

The following features are not required in this system:

- **Database schema:** You may change data within tables, but not the database structure.
- **No payment:** Payment for flights does not need to be considered at all.
- **No need to allow for return flights:** Flights are booked one at a time, independently.
- **Aircraft availability:** There is no need to check that an aircraft is not double booked or in the correct location for each flight.
- **No user passwords:** There is no need for passwords for users

Project Requirements

You must:

- Use the provided SQL file to create the database within your MySQL database & PythonAnywhere. This also creates initial data in the database.
- Create a Flask web application that:
  - meets the functional requirements
  - is appropriately commented
  - connects to your database
  - provides appropriate routes for the different functions
  - provides templates & incorporates HTML forms to take input data
  - uses Bootstrap CSS to provide styling and formatting
- Include a brief project summary document outlining the structure of your solution (routes & functions) and detailing any assumptions and design decisions that you have made. (We recommend noting these while you are building the app.)
  - This document must be created using GitHub Markdown and saved in the README.md file of your GitHub repository.
- Create a private GitHub repository that contains:
  - All Python, HTML, images and any other required files for the web app
  - A requirements.txt file showing the required pip packages.
  - An extract of your database from MySQL – schema and data
  - Your project summary as the README.md document
  - Your repository must have a .gitignore file and therefore not have a copy of your virtual environment.
  - Add lincolnmac (computing@lincoln.ac.nz) as collaborator to your GitHub repository.
- Host your system (including database) using PythonAnywhere
  - Add lincolnmac as your “teacher” via the site configuration.

Project Hints

Create your GitHub repository first and create all your required code/files in your local folder. Regularly commit and push changes to your GitHub repository.

PythonAnywhere is case sensitive so test your app early – we will mark the PythonAnywhere version of your app.

Spend some time sketching the structure of your application before you start developing. Think about which features could share the same (or nearly the same) functions and templates. Remember that you can nest templates.

For your summary document, write down your design decisions, compromises, workarounds, etc. *as you work on* your web app to remind you, as you are likely to forget later when you write it. For example, whenever you solve an issue, or work something out, take a quick note of what you did.

Focus on functionality over looks. Do not spend too much time making a page look 'perfect' or fancy. A tidy, functional page is enough. It does not need to look like Air New Zealand or Emirates – they have whole teams working on their sites full time. There are no extra marks for fancy design.

Marking:

General project aspects	30 marks
Functional project aspects	70 marks
TOTAL	**100 marks**

General Project Aspects: (30 marks)

Project Element	Marks Available
Project summary document: - outlining the structure of your solution (routes & functions) - detailing any assumptions and design decisions that you have made. - document created using GitHub Markdown and saved in the README.md file of your GitHub repository	**20 marks** - Note: If assumptions and design decisions are not included, then maximum mark available is 8. - Up to 3 marks of these 20 are awarded for spelling, punctuation, grammar and presentation
GitHub repository setup, and shared. **PythonAnywhere** hosting, and web app correctly configured, including database setup	**5 marks**
Consistent and professional **'look and feel'** (interface, Bootstrap styling & templates)	**5 marks**
TOTAL	**30 marks**

Functional Project Aspects: (70 marks)

General functional requirements (6 marks):

- Well commented and formatted HTML, SQL, and Python code throughout.
- Data validation on forms
- Functionally correct Python
- Well-structured SQL queries
- Appropriate naming, both of variables and labels
- Appropriate and well formatted content, internal system codes not visible to customers

Indicative marking allocation:

Functionality	Expectations	Approx. marks
General		**10**
General functional requirements	As listed above	6
Header and footer	Consistent header & footer on each page using Flask templates	4
Customer		**25**
Home page	Links to Arrivals & Departures and Booking pages	1
Arrivals/ Departures page	Information relevant to a customer displayed clearly and correctly.	3
Passenger login	Existing passenger can log in using email address. If unsuccessful, prompt to register as new passenger. Link to customer registration.	4
New passenger	New passenger registered and details added.	3
Edit passenger	Passenger can edit details once logged in (not primary keys).	4
Passenger bookings list	Show current bookings sorted by date-time. Link to edit or cancel.	3
Cancel booking	Passenger is successfully removed from a flight.	1
Available flights listing	Departure airport selection. Correctly displays future flights for 7 days from current date. Shows seats available on each flight. Link to booking page (if flight or seats available).	5
New booking	Passenger successfully added to flight.	1
Admin/ staff		**35**
Route	/admin route exists and operates correctly.	1
Staff login	Staff select their name from a list on /admin page. Manager/staff permissions passed to other pages. Users who are not staff cannot access staff admin pages from URL.	4
Passenger list	Displays list of passengers. Can click to edit details or bookings. Can search or filter list by passenger last name.	5
Add/ edit passenger details	Staff can change any passenger details.	3
Add/ edit bookings	Staff can change any booking details. Appropriate interface to find booking.	3
Flight list	Staff can see a list of all flights 7 days from the current date. List can be (optionally) filtered by departure or arrival airport and/or a specific date range (from-to). Clicking displays flight manifest.	5
Flight manifest	Flight details & capacity for staff. List passengers booked on each flight in last name, first name order, with sequential numbering. Can click passenger to edit details or bookings.	5
Add flights	Only managers can add flights, as specified – copied from latest week or individually.	4
Edit flights	Managers can edit all details. Other staff only status, est/act times. Dropdown boxes display current value as described.	5
TOTAL		**70**