School of Computer Science

THE UNIVERSITY OF SYDNEY

COMP9120 Database Management Systems

Assignment 2: Database Application Development

Group assignment (16%)

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 3 members. 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 Sunday 20th October (Week 11).** 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 THREE** files:

- An assignment coversheet as a PDF document (.pdf file suffix) which is available for download from this link on Canvas.

- A SQL file (`CSHschema.sql`) containing the SQL statements necessary to generate the database schema and sample data. This must contain the original schema and insert SQL statements, and any stored procedures (functions) you may have added.

- A Python file (`database.py`) containing the Python code you have written to access the database.

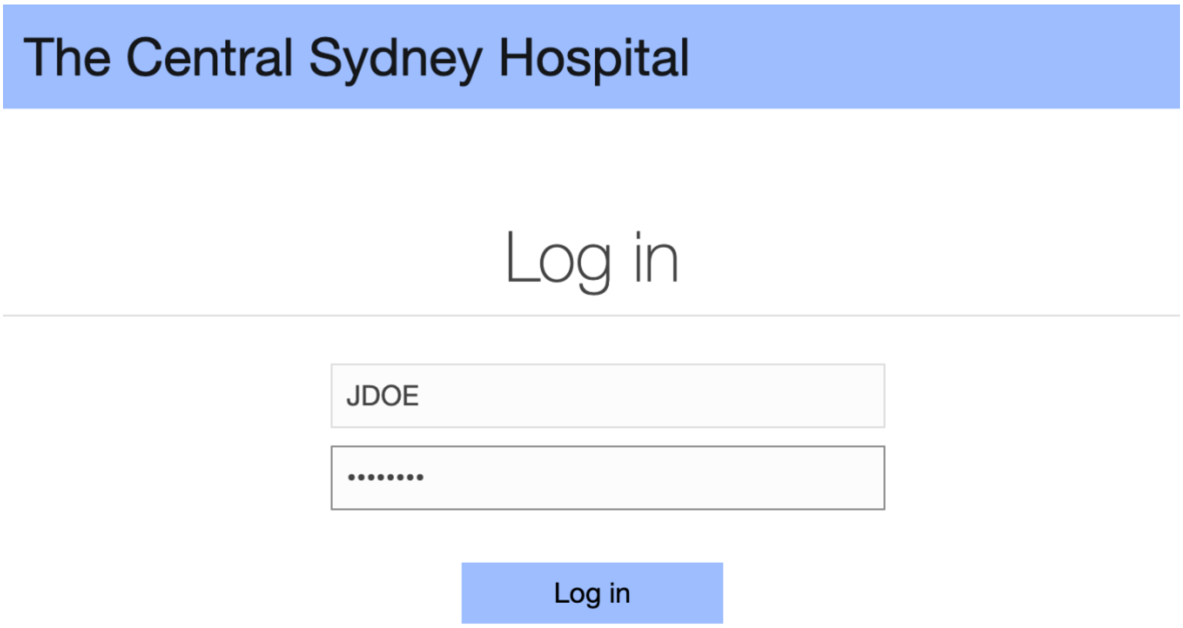
Your code should be your own, except for the scaffold code provided. You are strictly prohibited from using any generative AI tools, such as Microsoft Copilot or ChatGPT to complete any part of this assignment. Failure to comply with this policy may result in academic penalties as outlined in the academic integrity guidelines.

Task 1: The Central Sydney Hospital (CSH) System

In this assignment, you will be working with a modified version of the CSH database as described in Assignment 1. This assignment builds upon and partially use your work designing the CSH hospital database. More specifically, your task is to implement an interface, referred to as CSH system, through which a user interacts to access and manipulate the CSH database. Your main task in this assignment is to handle requests for reads and writes to the database using your User Interface (UI). We first describe the main features that the CSH system must include from a UI perspective, and then discuss where your database code needs to be implemented.

Logging In

The user is defined as any authorized administrator at the CSH hospital. The first form a user is presented with, when starting the CSH system, is the Login, as shown in Figure 1. This feature requires that the administrator enters their username and password to be validated prior to being successfully logged in to the system. Security features such as password encryption/hashing is out of scope for this assignment. Once logged in, the user is taken to the admission list page to view their associated admissions.



The Central Sydney Hospital

Log in

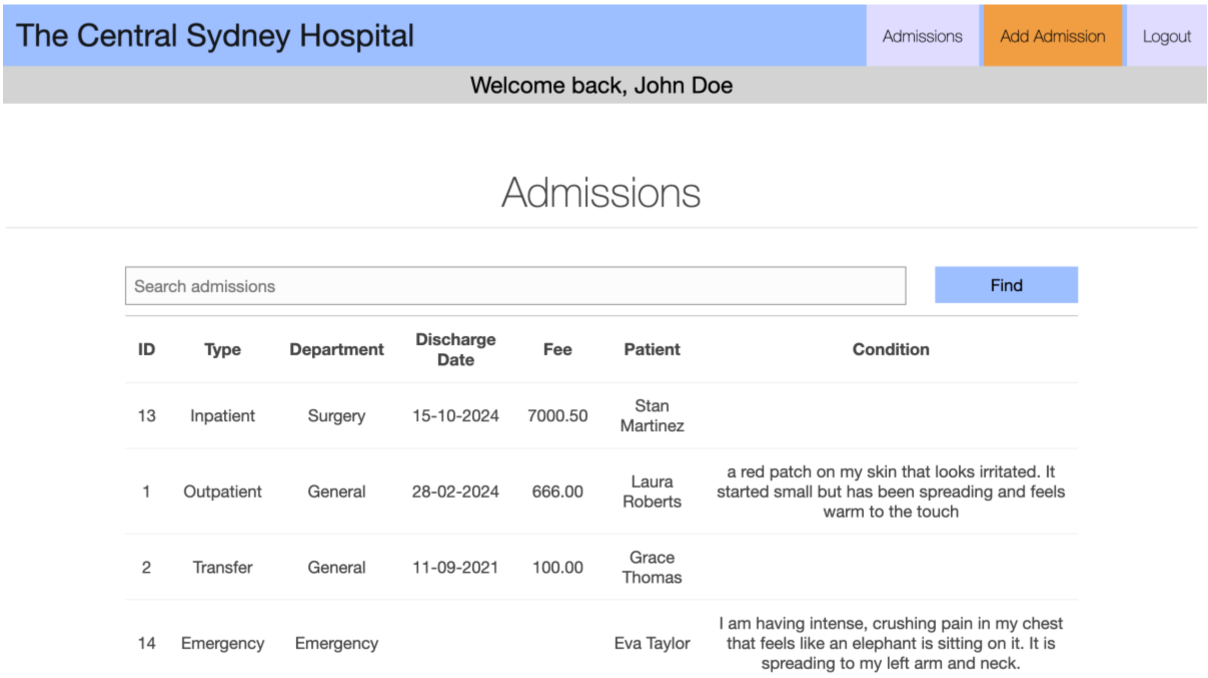
JDOE

••••••••

Figure 1 - Login

Viewing Admission List

Once a user is logged in, they are shown a list of all their associated admissions, as shown below in Figure 2. This list must be ordered such that the most recently discharged admission appear at the top. The list is also sorted by patient full name in ascending order, and admission type name in descending order. Admissions without a discharge date should be at the bottom of the list. Each admission has a type name, department, discharge date, fee, patient and condition. Each administrator is assigned a set of admissions to administer and review.



The Central Sydney Hospital

Welcome back, John Doe

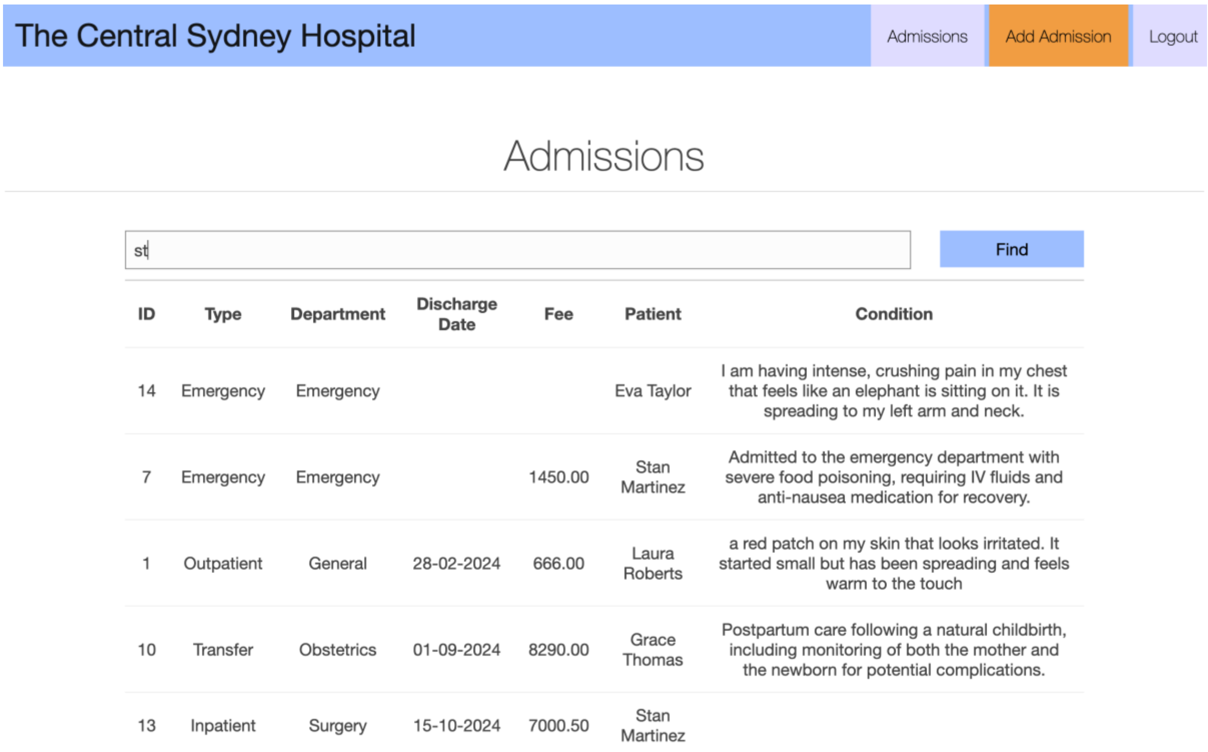
Admissions

ID	Type	Department	Discharge Date	Fee	Patient	Condition
13	Inpatient	Surgery	15-10-2024	7000.50	Stan Martinez	
1	Outpatient	General	28-02-2024	666.00	Laura Roberts	a red patch on my skin that looks irritated. It started small but has been spreading and feels warm to the touch
2	Transfer	General	11-09-2021	100.00	Grace Thomas	
14	Emergency	Emergency			Eva Taylor	I am having intense, crushing pain in my chest that feels like an elephant is sitting on it. It is spreading to my left arm and neck.

Figure 2 - Viewing Admissions List

Finding Admissions

A user can search through all admissions 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 the retrieved admissions and shown on the list are those that include this word or phrase in the admission type name, department name, patient’s full name or condition. The search is **case insensitive**. For example, given the search keyword ‘*st*’, Find will return all admission that include the keyword ‘*st*’ in the admission type name, department name, patient’s full name or condition. Searching with a blank/empty keyword field will show all of the logged in user’s associated admissions. Any search results returned must be ordered such that admissions without a discharge date at the top, then by discharge date in ascending order and patient’s full name ascending. The search results must exclude those admissions whose discharge dates are older than 2 years (from today’s date).

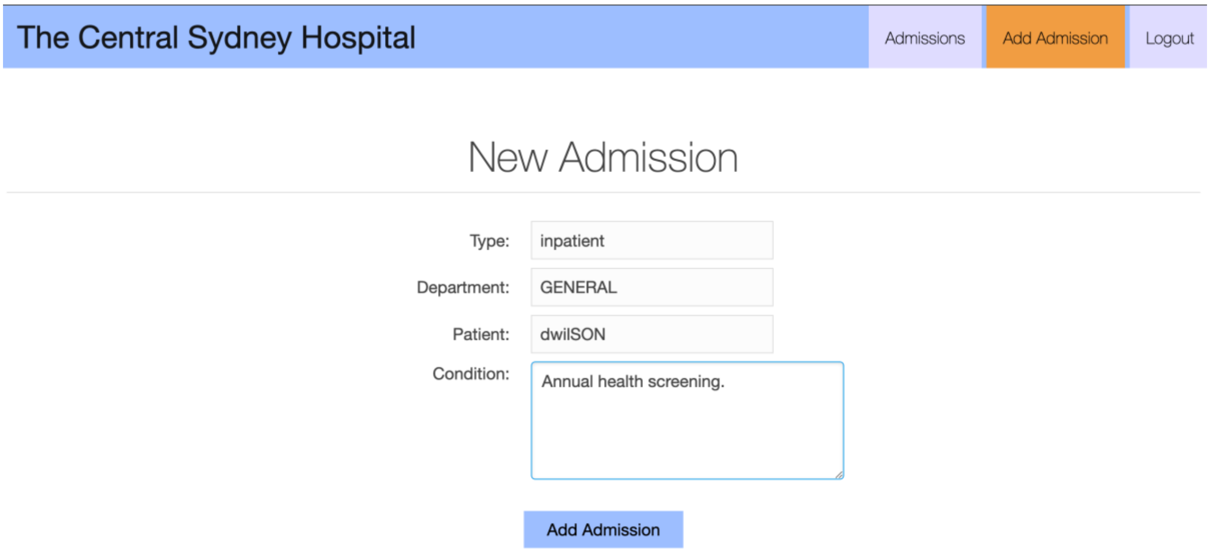


ID	Type	Department	Discharge Date	Fee	Patient	Condition
14	Emergency	Emergency			Eva Taylor	I am having intense, crushing pain in my chest that feels like an elephant is sitting on it. It is spreading to my left arm and neck.
7	Emergency	Emergency		1450.00	Stan Martinez	Admitted to the emergency department with severe food poisoning, requiring IV fluids and anti-nausea medication for recovery.
1	Outpatient	General	28-02-2024	666.00	Laura Roberts	a red patch on my skin that looks irritated. It started small but has been spreading and feels warm to the touch
10	Transfer	Obstetrics	01-09-2024	8290.00	Grace Thomas	Postpartum care following a natural childbirth, including monitoring of both the mother and the newborn for potential complications.
13	Inpatient	Surgery	15-10-2024	7000.50	Stan Martinez	

Figure 3 - Finding Admissions

Adding an Admission

Users may also add a new admission by clicking on the Add Admission tab in the title bar. They may enter admission details such as admission type name, department name, discharge date, patient id and condition, in the popup dialog. Once the popup dialog appears, click on Add Admission button, as shown in Figure 4. A new admission will not be assigned a discharged date or fee.



Type: inpatient

Department: GENERAL

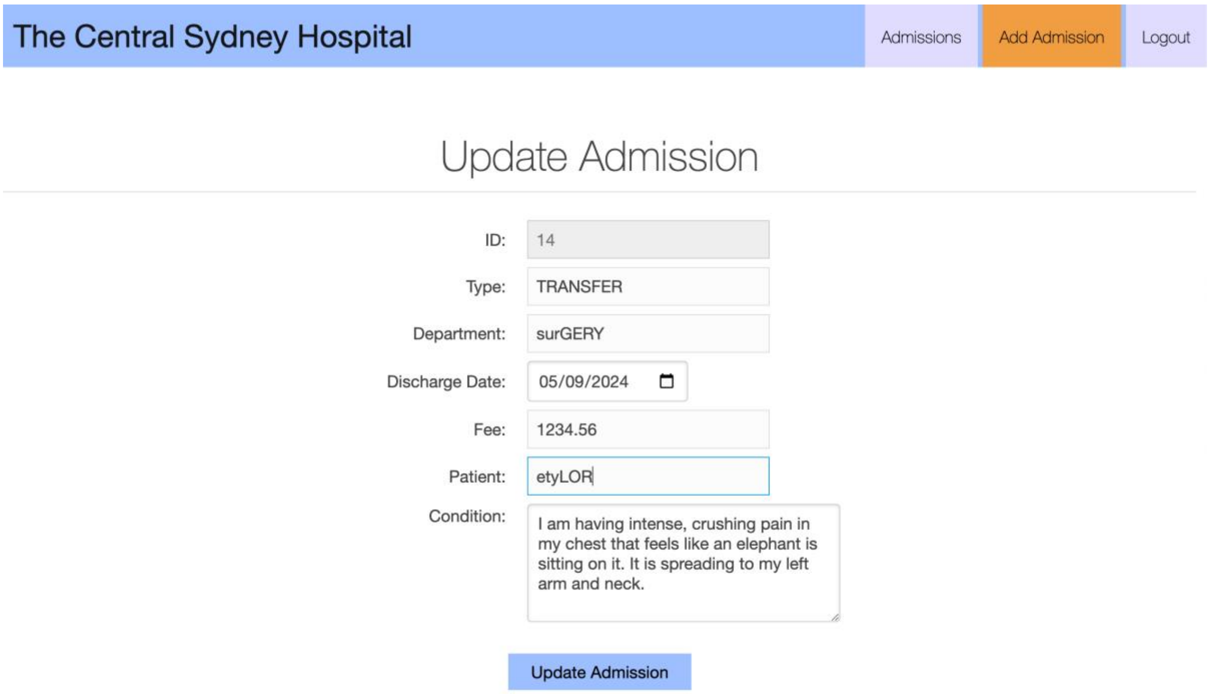
Patient: dwilSON

Condition: Annual health screening.

Figure 4 - Adding an Admission

Updating an Admission

Users can also update an admission by modifying data in the admission details screen as shown in Figure 5, by clicking on Update Admission button. You can access this update screen by clicking on an admission from the list of admissions in the Admissions tab.



Update Admission

ID: 14

Type: TRANSFER

Department: surGERY

Discharge Date: 05/09/2024

Fee: 1234.56

Patient: etyLOR

Condition: I am having intense, crushing pain in my chest that feels like an elephant is sitting on it. It is spreading to my left arm and neck.

Figure 5 - Updating an Admission

Database Interaction Code

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

1. **CSHschema.sql**: a file which contains SQL statements you need to run to create and initialise the CSH database, before starting the application

   https://canvas.sydney.edu.au/files/39291865/download?download_frd=1

2. **Assignment2_PythonSkeleton.zip**: a zip file encapsulating the Python project for the CSH system

   https://canvas.sydney.edu.au/files/39291881/download?download_frd=1

To inspect the CSH 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 CSH system. The main modules are the Flask framework for the GUI and the psycopg2 module for the PostgreSQL database access. Similar to tutorial 7, **you will need to install the Psycopg2 module and the Flask module**.

The skeleton code follows the structure described below:

- The main program starts in the **main.p**y 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 in **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://127.0.0.1:5001/ (If that doesn't work you can also try http://0.0.0.0:5001/). Please note that, to terminate the application, you will need to stop the local web server which is running in the background.

Task 2: Functions Implementation

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., **`CSHschema.sql`**). Specifically, you need to modify and complete these five functions:

1. **`checkLogin`** (for login)
2. **`findAdmissionsByAdmin`** (for viewing admission list)
3. **`findAdmissionsByCriteria`** (for finding admissions)
4. **`addAdmission`** (for adding an admission)
5. **`updateAdmission`** (for updating an admission)

Note that, for each function, the corresponding action and outcome should be **implemented by issuing SQL queries** to the database management system. If you directly output the result set, pre-process, manipulate and/or make changes to the input or output datasets using Python code or additional modules (libraries) i.e. without issuing SQL queries, you are considered as cheating, and you will get penalised heavily and most likely get zero point for the assignment.

No additional Python modules or libraries should be imported.

Marking

This assignment is worth 16% of your final grade for the unit of study. Your group's submission will be marked according to the attached rubric.

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:

Percentage of contribution	Proportion of final grade received
< 5% contribution	0%
5 - 10% contribution	20%
11 - 15% contribution	40%
16 - 20% contribution	50%
21 - 24% contribution	60%
25 - 28% contribution	80%
29 - 30% contribution	90%
> 30% contribution	100%

Note: The above table assumes that each group will have 3 members, so, on average, around 33% contribution is expected from each member of the group. In special case, if a group has less than 3 members then the contribution percentage will be adjusted accordingly. You must justify your contribution percentage by providing a detailed explanation of your individual contribution on the assignment coversheet mentioned before. You must also regularly record and maintain a diary of your group meetings and discussions on Canvas. Furthermore, we may run random face-to-face interviews to understand and justify your contribution, if needed.

Marking Rubric

Your submissions will be marked according to the following rubric, with a maximum possible score of **16** points.

	Part Marks (0 – 1.5 pts)	**Full Marks (2 – 2.5 pts)**
Login	Can correctly login the user 'JDOE' and validate his username and password.	All valid users can be logged in successfully, and unsuccessful user logins should be rejected.
View Admission List	Correctly list all admissions associated with user 'jdoe' in the correct order (see Figure 2)	Correctly list all admissions associated with a logging user, in the correct order, for all possible username input from Figure 1.
Find Admissions	Correctly list admissions for keyword "st" (see Figure 3)	Correctly list admissions for all possible keywords.
Add Admission	Can correctly add an admission to the database.	Can correctly add all valid admissions to the database. Admissions entered with invalid details should be rejected.
Update Admission	Can correctly update the status of an admission.	Can correctly update details of all admissions, ensuring the updated details for an admission are valid.

Stored Procedure (Function)	A couple of stored procedures (functions) are correctly created in the submitted SQL file.	A couple of stored procedures (functions) are correctly created in the submitted SQL file, and correctly called in two of the five specified functions.

	No Marks (0 pt)	**Full Marks (1 pt)**
Record Keeping of Group Discussions	One or more issues reported or found with group member contribution, or with maintaining records of group meetings and discussions regularly on Canvas.	No issue reported or found with group member contribution. All group members participate and regularly maintain a diary of group meetings and discussions on Canvas.