BIT502

ASSESSMENT

3

Ver: A1.0

BIT502 *Fundamentals of Programming*

Assessment 3

Weighting

40%

Learning outcomes

2 Apply the principles of interface design to create usable and accessible user interfaces.

4 Design and develop a simple database application.

Instructions

Complete and submit your assessment according to the Open Polytechnic's Assessments web page. This includes information on academic integrity, word limits and referencing.

- Include your name, student number and the assessment number.
- Number your pages.

Submission

- Submit your assessment in two files. The code file should be submitted as a zip file.
- Submit your work through your iQualify course.
- Emailed assessments will not be accepted.
- You will receive an automated notice following submission.

By submitting your assessment, you confirm that it is your own, original work.

Introduction

You need to submit this assessment in two files: the report as an MS Word file and the Python code file as a zip file.

In this assessment, you will develop a gym membership application that will be used only by the admin. Please note, this is not a client application. The application will have the following features:

- registration of new members and adding them to the database
- searching existing members in the database
- deleting records of existing members from the database
- booking existing members for fitness classes
- having a help section for the admin on how to use this application.

Task 1: Create forms and wireframes

Part 1

You are required to design and create five forms/screens for the gym membership application as below:

1. **Main screen**: This is a static window that should contain labels and navigation components/buttons to the rest of the forms, including the help window.
2. **Membership form**: This form is the same one created for Assessment 2. You can reuse the form without any changes.
3. **Search form**: This form is used to search for members based on either the member's ID, last name, membership type or a combination of last name and membership type. Use appropriate UI components and layout to accommodate these search criteria. The form should also have a component to display the retrieved information.
4. **Fitness form**: The purpose of this form is to enrol registered members in one of the following fitness classes:
    - Cardio, Thursday, 3 pm–5 pm
    - Pilates, Friday, 9 am–11 am
    - Spin, Monday, 2 pm–4 pm.
5. **Help screen**: This is a static window that should contain information on how to use this application.

(Part 1: 10 marks)

Part 2

a. You need to create wireframes for all UI screens/forms. These wireframes should display all UI elements you plan to build in the application and clearly display how navigation between screens will occur. You can submit this as an MS Word document or in jpg/png format.

(5 marks)

b. Explain how you have made your wireframe usable and accessible.

(5 marks)

(Part 2: 10 marks)

[Task 1 total: 20 marks]

Task 2: Create the database

Your next step is to design the following required tables in the database:

1. **Membership table**: This is a fixed table that would contain the type of membership and the base cost associated with the membership. Remember to have a membership ID as the primary key for this table.
2. **Fitness table**: This is a fixed table that would contain the fitness class type. Remember to have a fitness class ID as the primary key for this table.
3. **Members table**: This is one of the main tables that is used to store the details of a member. The following information needs to be stored about a member:
    - First name, last name, address, mobile number, payment frequency, extras, if any (not the cost but the type of extras) and regular payment amount. Remember to have a member ID as the primary key for this table.
    - The members table should be populated with sample data to demonstrate/test the search mechanism of this application.
4. **Booking table**: This is a link table that would store/link a member with the type of fitness class. Remember to have a booking ID as the primary key for this table.

[Task 2 total: 10 marks]

Note

You need to figure out the relation between the tables using foreign keys where necessary. There could be one or more than one foreign key in a particular table.

A single member can only be registered once as a basic, regular or premium member. Hence, a member cannot hold more than one registration type. However, a member can enrol in more than one fitness class.

Appropriately choose the data type for each field/column within a table. Furthermore, decide wisely which of the fields can accommodate null data and which cannot have null data.

Task 3: Code the application

For each of the five screens/forms you created in Task 1, complete the following tasks:

1. **Main menu screen**: Add code for navigation to all five forms/screens.

(5 marks)

2. **Membership form**: Save the data into the database. Insert the data into the members table when registering a new member. Note that the previous code that you created to save into a text file would be replaced by this code. A pop-up message should be displayed upon a successful insertion of data, or an error message should appear otherwise.

(15 marks)

3. **Search form**: The search code should retrieve the entire information from all the tables based on your search mechanism. The retrieved information could be a single set of data or more than one set of data. If there is no match for the search criteria, a no-record message should be displayed. The format to display the retrieved information is your choice. Refer to the example given at the end of the assessment, which will help you understand the search functionality.

(20 marks)

4. **Fitness form**: This form is only for registered members. The form has two sections. Refer to the example given at the end of the assessment.

(20 marks)

- Section one is a search mechanism based on the ID that would perform the following:
  - validate an existing user
  - display all the fitness classes this member has been enrolled in

- if the user is invalid, display a message saying either the ID is invalid or register before booking a fitness class.

- Section two: Only when the user has been validated as a registered user, the second section of this form is displayed with the information below.
  - A fitness class selection (use a radio button).
  - If the member selects a fitness class that he/she has already enrolled in, an error should be displayed to re-select.
  - Error-checking should be implemented if no selection has been made.
  - If the selection is valid, the booking information should be inserted into the booking table. A message should be displayed for successful insertion, and the form should be reset to the initial display; that is, hiding the selection section.
  - Create a reset button to reset the screen at any stage.

[Task 3 total: 60 marks]

Task 4: User testing and documentation

Part 1: Acceptance testing

Once you have completed your application, explain how you have conducted user acceptance testing. The test plan should be a table with multiple columns and build on the user acceptance testing.

- In the first column, list all test items (features or steps).
- In the second column, provide details of all expected results.
- In the third column, provide details of the actual results.

Note: Your test plan should be based on the application you created.

(Part A: 10 marks)

[Task 4 total: 10 marks]

Guideline for creating tables in Task 3

Data in the members table (just a rough table; don't consider this as a guideline when creating tables as there could be other fields as well). Only for explanation purposes, records highlighted in green (green) are basic, yellow (yellow) are regular, and red (red) are premium.

Table 1 Snapshot of how data is supposed to be stored in City Gym database

ID		F. name	L. name	Address	Mobile	P. freq.	Extras	R. Amount
1		John	Smith	xyz	45	weekly	24, vids	56
2		Marry	Sam	abc	22	weekly	none	13
3		Henry	Sam	def	15	monthly	diet	243
4		Ibrahim	Hussain	rty	78	weekly	24, vids, trainer	88
5		Ken	Lee	arm	59	monthly	none	113
6		Aria	Ahorangi	uop	11	weekly	vid	34
7		Sarah	Brown	zmi	18	monthly	24, trainer	206

Data is stored in the booking table as follows:

- John Smith is booked for pilates and spin.
- Marry Sam is not booked for any fitness class.
- Henry Sam is booked for only cardio.
- Ibrahim Hussain is not booked for any class.
- Ken Lee is booked for all three fitness classes.
- Aria Ahorangi is booked for spin and cardio.
- Sarah Brown is not booked for any class.

- Assuming the search criteria is **last name**, and the search input is **Lee**, the retrieved information should be:

5	Ken	Lee	arm	59	monthly	none	113

and fitness classes are cardio, Thursday, 3 pm–5 pm; pilates, Friday, 9 am–11 am; spin, Monday, 2 pm–4 pm.

- Assuming the search criteria is **membership type**, and the search input is **basic**, the retrieved information should be:

1	John	Smith	xyz	45	weekly	24, vids	56

and fitness classes are pilates, Friday, 9 am–11 am.

4	Ibrahim	Hussain	rty	78	weekly	24, vids, trainer	88

and no fitness classes are booked

- Assuming the search criteria are **membership type** and **last name**, and the search input is **regular** and **Brown**, the retrieved information should be:

7	Sarah	Brown	zmi	18	monthly	24, trainer	206

and no fitness classes are booked.

- Finally, assuming the search criteria are **membership type** and **last name**, and the search input is **premium** and **Smith**, the retrieved information should be:
NO MATCH FOUND.

Marking schedule

Task 1, Part 1	9–10	7–8	5–6	4–4	1–3
Create forms and wireframes Design and create five forms/screens for the application. • Main screen • Membership form • Search form • Fitness form • Help screen.	All five forms have been designed and created. The design of each page shows evidence of adherence to good design principles. The navigation buttons on the home screen are functional. The required content and functionality required for each page are present and accurate.	All five forms have been designed and created. The navigation buttons on the home screen are functional. The required content and functionality required for each page are present, with few minor content or functionality errors that do not detract from the purpose of the pages.	All five forms have been designed and created. The navigation buttons on the home screen are functional, with minor errors in linked destination. The required content and functionality required for each page are present, with minor content or functionality errors that do not detract from the purpose of the pages.	All five forms have been designed and created. The navigation buttons link to the wrong destinations. The required content and functionality required for each page are present, but significant inaccuracies exist.	Less than five forms have been designed and created. The navigation buttons link to the wrong destinations. Significant gaps in the required content and functionality required for each page exist.
Task 1, Part 2a	**5–5**	**4–4**	**3–3**	**2–2**	**1–1**
Display all UI elements and display navigation between screens.	All UI elements are displayed accurately. How navigation between screens will occur is shown and is accurate.	All UI elements are displayed. How navigation between screens will occur is shown. Minor inaccuracies in the UI elements are present.	All UI elements are displayed. How navigation between screens will occur is shown. Minor inaccuracies in the UI elements and how navigation will occur are present.	All UI elements are displayed. How navigation between screens will occur is shown. Significant inaccuracies in the UI elements and how navigation will occur are present.	UI elements are missing. Significant inaccuracies in how navigation will occur are present.

Task 1, Part 2b	**5–5**	**4–4**	**3–3**	**2–2**	**1–1**
Explain how you have made your wireframe usable and accessible.	A logical explanation of how the wireframes are usable and accessible is provided. The explanation is justified with reference to other accessible and usable wireframes. The created wireframes are usable and accessible.	A logical explanation of how the wireframes are usable and accessible is provided. The created wireframes are usable and accessible.	A logical explanation of how the wireframes are usable and accessible is provided.	An explanation of how the wireframes are usable and accessible is provided; however, the explanation is insufficient to prove the usability or accessibility of the wireframes.	An inaccurate explanation of how the wireframes are usable and accessible is provided.
Task 2	**9–10**	**7–8**	**5–6**	**3–4**	**1–2**
Create the database The following tables are in the database: • Membership table • Fitness table • Members table • Booking table • The members table to demonstrate/test the search mechanism.	All four tables have been accurately designed and created. The relationship between the tables (using foreign keys where necessary) is accurately determined. Member registration numbers and data choices are accurate.	All four tables have been designed and created. Few insignificant errors are present. The relationship between the tables (using foreign keys where necessary) is determined with minor inaccuracies. Member registration numbers and data choices are accurate.	All four tables have been designed and created. Minor errors are present. The relationship between the tables (using foreign keys where necessary) is determined with minor inaccuracies. Minor errors exist in member registration numbers and data choices.	All four tables have been designed and created. Significant errors are present. The relationship between the tables (using foreign keys where necessary) is determined with significant inaccuracies. Significant errors exist in member registration numbers and data choices.	Less than four tables have been designed and created. Significant errors are present. The relationship between the tables (using foreign keys where necessary) is determined, with significant inaccuracies. Significant errors exist in member registration numbers and data choices.

Task 3.1	**5–5**	**4–4**	**3–3**	**2–2**	**1–1**
Code the application **Main menu screen**	Navigation code has been accurately added for each form.	Navigation code has been added for each form. Few errors in code lead to inaccurate navigation in a form.	Navigation code has been added for each form. Minor errors in code lead to inaccurate navigation in several forms.	Navigation code has been added for each form. Significant errors in code lead to inaccurate navigation in all forms.	Navigation code has not been added for all forms.
Task 3.2	**13–15**	**11–12**	**8–10**	**6–7**	**1–6**
Code the application **Membership form**	All data inserted accurately. Old code replaced with new code.	All data inserted; few inaccuracies are present. Old code replaced with new code.	All data inserted; minor inaccuracies are present. Old code replaced with new code.	All data inserted; minor inaccuracies are present. Old code still exists.	All data inserted; significant inaccuracies are present. Old code still exists.
Task 3.3	**17–20**	**14–16**	**10–13**	**8–9**	**1–7**
Code the application **Search form code**	All data is retrievable using the search mechanism. Data displayed in search results are accurate. A message is displayed when a record is not found.	All data is retrievable using the search mechanism. There are minor errors in data displayed in search results. A message is displayed when a record is not found.	All data is retrievable using the search mechanism. There are minor errors in data displayed in search results. No message is displayed when a record is not found.	All data is retrievable using the search mechanism. There are significant errors in data displayed in search results.	Not all data is retrievable using the search mechanism. There are significant errors in data displayed in search results.

Task 3.4	**17–20**	**14–16**	**10–13**	**8–9**	**1–7**
Code the application **Fitness form code**	The search mechanism is based on ID. Validation and display work as intended. An invalid user message is displayed when a user is invalid.	The search mechanism is based on ID. Few minor inaccuracies exist in user validation and display. An invalid user message is displayed when a user is invalid.	The search mechanism is based on ID. Minor inaccuracies exist in user validation and display. No message is displayed when a user is invalid.	The search mechanism is based on ID. Significant inaccuracies exist in user validation and display.	The search mechanism is not based on ID. Significant inaccuracies exist in user validation and display.
Task 4	**10–9**	**7–8**	**5–6**	**3–4**	**1–2**
User testing and documentation	Evidence of user testing submitted. All site components were tested. No errors are missed. Expected and actual results are clearly communicated.	Evidence of user testing submitted. All site components tested. Minor errors missed. Expected and actual results clearly communicated.	Evidence of user testing submitted. All site components tested. Minor errors missed. Expected and actual results not clearly communicated.	Evidence of user testing submitted. Not all site components tested. Minor errors missed. Expected and actual results not clearly communicated.	Evidence of user testing may be submitted. Not all site components tested. Significant errors missed. Expected and actual results not clearly communicated.