COURSEWORK SUBMISSION FORM

STUDENT USE		STAFF USE	
Module Name	Database Systems Development	First Marker's (acts as signature)	
Module Code	5BUIS009C-n	Second Marker's (acts as signature)	
Lecturer Name	Dmitriy Pochitaev	Agreed Mark	
UoW Student IDs		**For Registrar's office use only (hard copy submission)**	
WIUT Student IDs	00019257, 00015541, 00017102		
Deadline date	04/07/2025		
Assignment Type	Group		

SUBMISSION INSTRUCTIONS

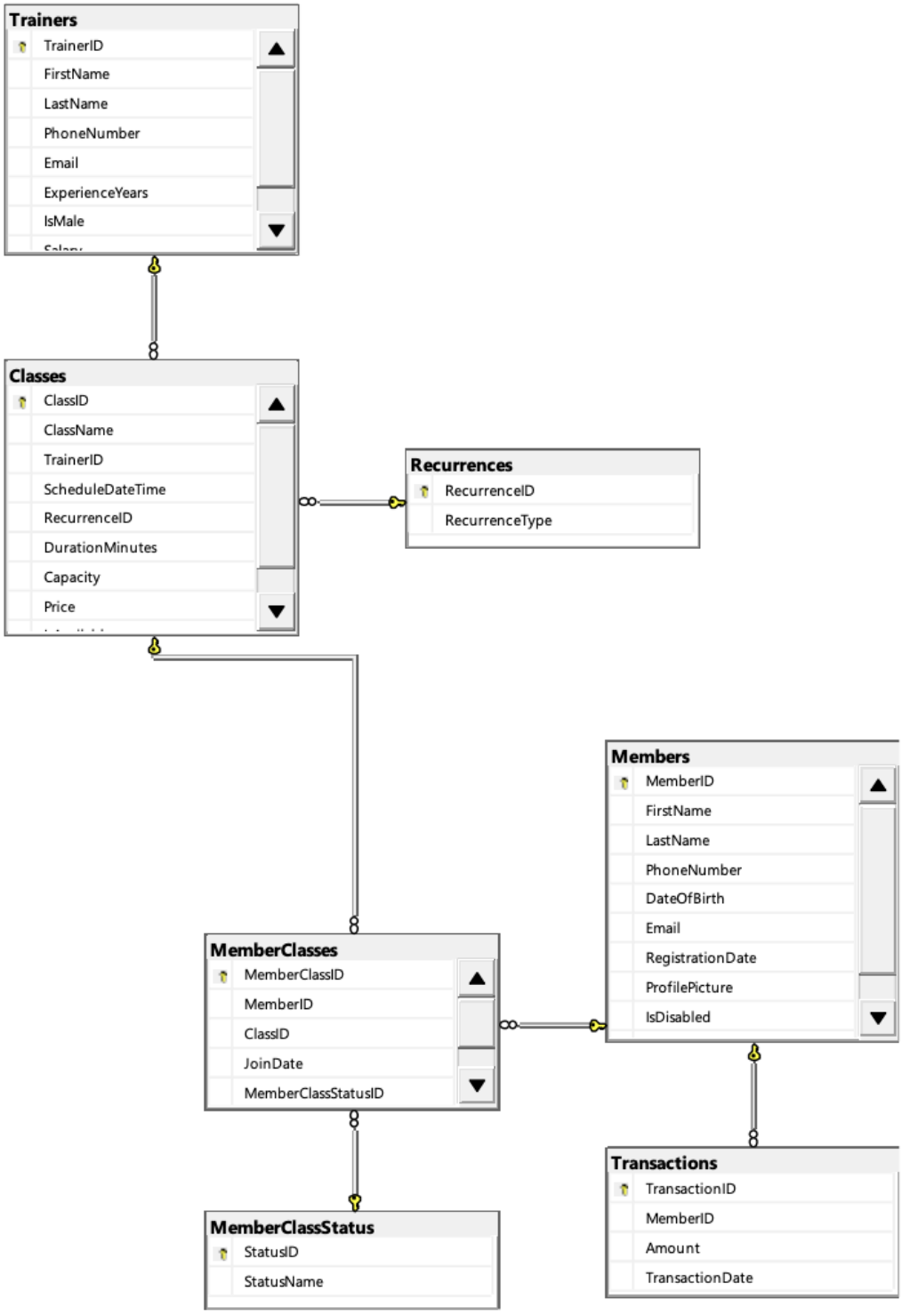
COURSEWORKS *must* be submitted in *both* HARD COPY (to the Registrar's Office) *and* ELECTRONIC unless instructed otherwise.

For hardcopy submission instructions refer to: http://intranet.wiut.uz/Shared%20Documents/Forms/AllItems.aspx - Coursework hard copy submission instructions.doc

For online submission instructions refer to: http://intranet.wiut.uz/Shared%20Documents/Forms/AllItems.aspx - Coursework online submission instructions.doc

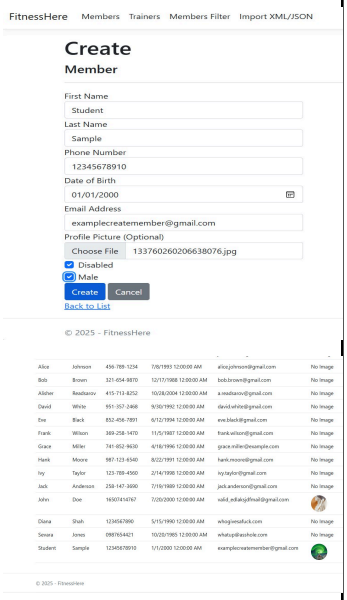
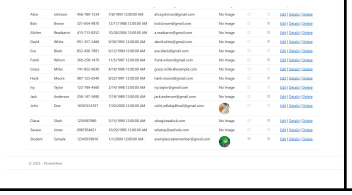
MARKERS FEEDBACK (Continued on the next page)

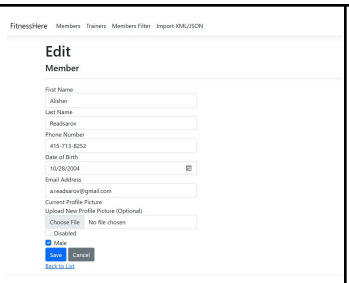
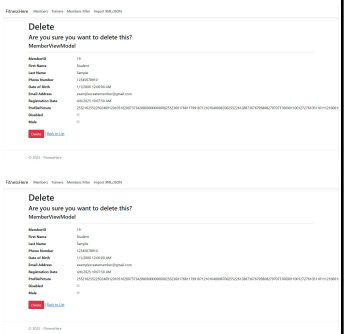
Relational DB Diagram



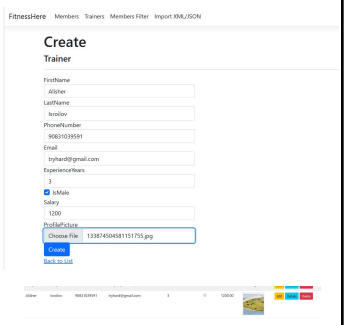
Black Box Testing

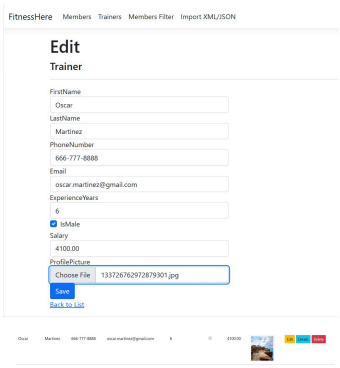
Stored Procedures CRUD

Test description	Expected results	Actual results	Comments
Create a member with valid data (a profile picture, full name, etc) using stored procedures.	New member is added to the Members table in the database, and is listed in the application.	The member is successfully saved and displayed in the application. 	Member is successfully created.
Update an existing member's data.	Member's data should be updated in the database and, in turn, in the database.	Alisher Readsarov's data is correctly changed (last name to Reads and email) in list and database. 	Correctly updated.

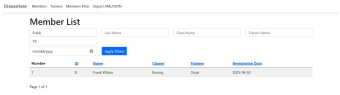
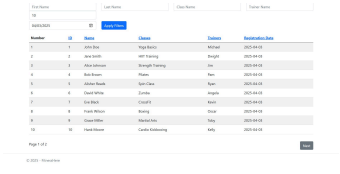
			
Delete a member.	Should delete the Student Sample that we created during the first test.	Was successfully removed from the application list and database. 	It takes a while to delete (after pressing delete, one needs to wait approximately 15-20 seconds), but it does eventually ask for deletion confirmation.

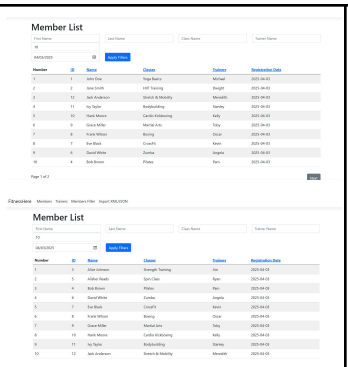
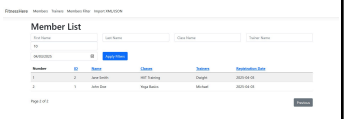
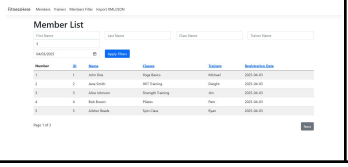
EF CRUD

Test description	**Expected results**	**Actual results**	**Comments**
Create a trainer with valid data using entity framework.	The new trainer Alisher Isroilov is added to the Trainers table and is visible in the Trainers list.	The trainer is successfully saved and displayed in the application. 	-

Update a trainer's information.	Oscar Martinez now has a profile picture that is updated in the database and visible in the application.	Oscar Martinez's new pfp is visible. 	-
Delete a trainer's information.	Deleting trainer added during create testing, Alisher Isroilov. Should be removed from the database and no longer visible in trainers list.	No longer visible on the list. 	Takes time to ask for confirmation of deletion, but the procedure itself works just fine.

Filtration/Sorting/Paging

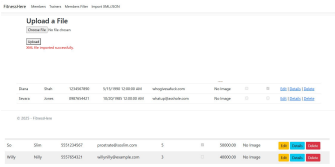
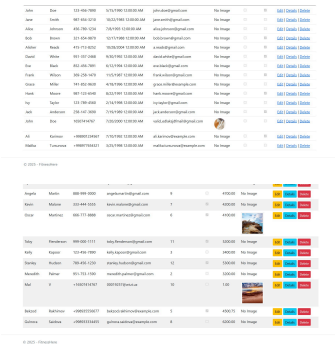
Test description	**Expected results**	**Actual results**	**Comments**
Filter data by first name. Searching "Frank" in the list.	Should have 1 result, Frank Wilson.	Procedure successfully filters out all Members other than Frank. 	-
Filter data by date. Search for members registration date 04/03/2025.	Should have 12 results, only 10 displayed on first page, as all insertions were done before search date.	10 results displayed on first page, 2 on the second page. Successful filtration by date. 	-
Sort by Name: Descending and Ascending.	Should go from descending to ascending upon clicking "Name" in Members Filter page.	Successfully toggles between ascending and descending on click of "Name" column.	-

			
Pagination, load second page.	Upon clicking “Next,” the rest of the Members should be visible.	Additional two members visible upon entering the second page. Pagination works. 	-
Pagination, choosing the number of results per page. Choose 5 for page count.	Should output 5 members per page.	Successfully reduces page count to 5. 	-

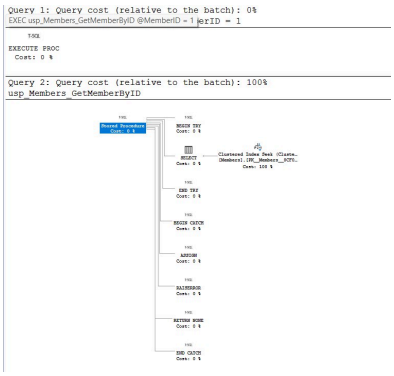
Export

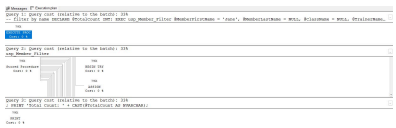
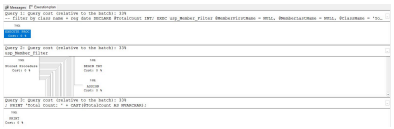
Test description	**Expected results**	**Actual results**	**Comments**
Export data to XML. Run “EXEC usp_Members_XMLExport;” in SQL Server.	Should export filtered to XML in required format (nested).	Procedure successfully exports filtered into XML.	-
Export data to JSON. Run “EXEC usp_Members_JSONExport;”	Should export filtered to JSON in required format.	Procedure successfully exports filtered into XML.	Exports into one line.

Import

Test description	Expected results	Actual results	Comments
Import data into Members and Trainers via XML file.	Should give red text “XML imported successfully” within the import page. Members page should now have Diana Shah and Sevara Jones. Trainers page should now have So Slim and Willy Nilly.	Procedure successfully imports into two tables. 	XML import data file available within the project as “import_xml.xml”.
Import data into Members and Trainers via JSON file.	Should give red text “JSON file imported successfully.” within the import page. Members page should now have Ali Karimov and Malika Tursunova. Trainers page should now have Bekzod Rakhimov and Gulnora Saidova.	Procedure successfully imports into two tables. 	JSON import data file available within the project as “import_json.json”.

Query Performance Optimization

BEFORE	INDEX	AFTER
	CREATE UNIQUE NONCLUSTERED INDEX index_Members_Email ON Members(Email);	

	CREATE INDEX index_Members_FirstName_ LastName_RegistrationDate ON Members (FirstName, LastName, RegistrationDate);	
	CREATE INDEX index_Members_MemberID ON Members (MemberID);	
	CREATE INDEX index_Classes_ClassName ON Classes (ClassName);	

Roles & Users

Implemented in database layer, in file “indexes, users and privileges.sql.

Appendix A

The fitness class studio, Fitness Here, offers a wide range of fitness classes aimed at improving health and wellness. The studio offers several types of memberships and provides various fitness classes, such as Yoga, HIIT, Strength Training, Pilates, and many others. The studio uses an integrated database management system to manage members, trainers, classes, and transactions, as well as providing detailed access controls for different user roles within the system.