Bahrain Polytechnic بوليتكنك البحرين

BICT IT8118: Advanced Programming

Semester 2 2023-23: Group Project Instructions

Tutors:	Husain Naser
Semester:	Semester 2, 2022/2023
Learning Outcomes Covered	The following learning outcomes will be assessed: 1. Design and implement desktop applications that access and manipulate a database 2. Design and implement web applications that access and manipulate a database 3. Use object-oriented techniques to design and implement custom database entity classes 4. Manage a multi-tier application which uses custom entity classes to access and manipulate a database 5. Propose an appropriate technology for a particular problem
Deadline:	4th June 2023 @ 11:59 PM

Delivery:	Project files and documents must be uploaded to Moodle's course page before the deadline. Files to be submitted are as follows:
	• Compressed Project files (one submission per group) including:
	  ○ Project Document with Screenshots *(.docx file)*
	  ○ Database SQL Script with schema and data *(.sql file)*
	  ○ Class Library Project Files *(Visual Studio Solution)*
	  ○ Windows Forms Project Files *(Visual Studio Solution)*
	  ○ ASP.NET Core MVC Project Files *(Visual Studio Solution)*

Weighting:	35%
Instructions	• You will work in groups of 3-4 students. • A Word document template will be provided for the project document. The document containing the database design and screenshots of the working solution, as well as the complete project files for all components of the project should be uploaded to Moodle (one per group) by the deadline. • Requests for extensions should be made before the deadline by 48 hours at least to Course Coordinator. Extensions will only be approved with valid reasons, up to a maximum of 2 calendar days. You are only permitted a maximum of one extension per course per semester. • If the assessment is submitted late the maximum result a student can achieve is 60% • The cut off time for submitting an assessment will be 3 calendar days after the assessment is due. A student submitting after 3 calendar days will get 0%.

Contents

Project Description

Project Overview:

Project Objective:

In this project, you will develop a Task Management System that can be used by individuals and companies to manage their tasks more effectively. The system will include features such as authentication and authorization, task creation, assignment, tracking, notifications, dashboard, and more.
The system will have a Windows Forms application and an ASP.NET Core MVC application, both of which will use a shared database and Entity Framework Core models.

IDE: Visual Studio 2019/22 **Language**: Visual C# (.NET Core) **.NET:** 5 or 6

Business Scenario:

A new company has developed a strategy for the managing tasks across different teams and departments to provide better visibility on tasks allocation, keep track of their progress, and manage their documentation in an organized manner.
The company follows a projectized organization structure. The project managers want to create and manage projects, manage project tasks, allocate staff to work them, and track their progress. Project team members need to view the tasks assigned to them, update their status, and manage task-related comments and documents.
(**Note:** manage usually includes functionality to Create, Read, Update, and Delete).

Project Groups:

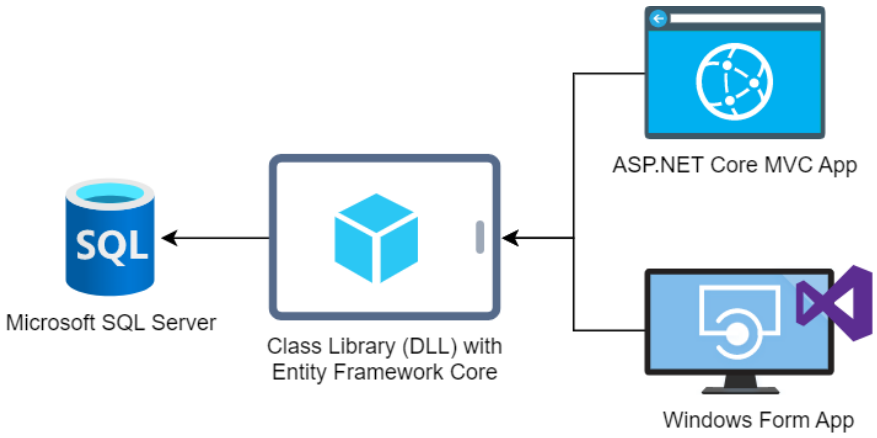
Group members assignment will be coordinated in class.

- The required number of students in each project group is 3-4 as far as possible. Each member of a project group should take a leadership role on one of the project components. However, all members of the group should work together on all the components.
- The project is assessed as a group project.

Project Components:

The project will have the following main components:

- Database (using Microsoft SQL Server)
- Class Library Project (using Entity Framework Core)
- ASP.NET Core MVC Project
- Windows Forms Application Project



Project Tasks:

The project is split into tasks to allow easier allocation between the group members, where each member can take the lead in a particular task. However, the success or failure at each task will affect the whole project. Thus, it is recommended to have group discussions, design, implementation, quality checks, and change management done collaboratively at each stage.

Task 1: Design and Implement the Database and Data Access Layer using Class Library & EF Core

Learning outcomes assessed:

- *Use object-oriented techniques to design and implement custom database entity classes*
- *Manage a multi-tier application which uses custom entity classes to access and manipulate a database*
- *Propose an appropriate technology for a particular problem*

To build a comprehensive Task Management System, you need to design your entities in a way that satisfies the requirements of the project.

Suggested Database entities *:

Entity	Entity Description
User	A user entity will store information about the users of the system, including their name, email, hashed password, and user role.
Project	This entity will store information about the projects, including the project name, description, project manager, etc. Each project can have multiple members.
Task	A task entity will store information about the tasks created by the project managers within a project, including the task name, assigned to user, description, status, deadline, etc.. A task can be assigned to a user from the project members.
Comment	A comment associated with a task, containing information or notes about the task that users or manager can post, and including comment date, time, user, comment text, etc.

Document	A document entity will store information about the documents uploaded by users related to tasks, including the document name, user, upload date, document type, binary data (or path), etc.
Notification	A notification entity will store information about the notifications sent to users, including the notification message, type, and status (read / unread)
Log	A log entity will store logging data about users' actions and any exceptions thrown with source (i.e. form or web & form/page name), type (i.e. exception or action), user, date & time, message, original values, and current values. *(Note: you can split it into separate exceptions log and audit trail entities if desired.)*
Other	You can add any other entity you require to have a fully functional system

**This list is provided as a baseline example*

You can follow either Database First or Code First approaches. In either case, make sure your database is well-designed, normalized, and includes auto-incremented primary keys, foreign keys, and correct relationships between tables.

For the user and role entities, you can design your own tables ensuring no passwords are stored in clear text. The recommended approach, however, is to use .NET Membership tables and use it to authenticate users on both your desktop and web applications.

Class Library project:

To develop the data access layer, create a Class Library project. The class library should model the database and allow access to it through Entity Framework Core Object-Relational Mapping Framework (OR/M).

Your Class Library should include:

1. Domain Classes with database tables mapping, annotated properties, and navigational properties.
2. Database Context with database connection that acts as a gateway to the Database.
3. DBSet collections which act as Database Tables

Task 2: Implement the ASP.NET Core MVC Web Application

Learning outcomes assessed:

- *Design and implement web applications that access and manipulate a database*
- *Manage a multi-tier application which uses custom entity classes to access and manipulate a database*
- *Propose an appropriate technology for a particular problem*

The ASP.NET Core Web Application should be the main interface for anyone who want to use this system. Each role will have their own permissions inside the Web Application, implemented through Authentication and Authorization (Preferably, using .NET Membership).

List of features & requirements *:

Feature	Requirements
Home	**Access Control:** • Available for guest access **Functionality:** 1. None. Includes general description of the system, its purpose, features, modules, etc. (informational only)
Authentication and Authorization	**Access Control:** • Anyone can access the registration and login pages. **Functionality:** 1. **Database Seeding:** An admin user should be seeded (inserted) in the database for elevated privileges 2. **User Registration**: Users should be able to register themselves (as normal users only) by providing their name, email, and password. 3. **User Login:** Users should also be able to login to the system using their email and password. 4. **User Profile:** Users should have a profile page to view and update their information or change their password.
Administration	**Access Control:** • Only Administrators should have access to the Administration pages **Functionality:** 1. **View Audit Trails / Logs:** The admin should be able to see audit trails and logs of the system
Project Management	**Access Control:** 1. All authenticated users should have access to create projects, where they will be assigned as project manager for projects they create. 2. Listing, searching, and reading project information are allowed for project managers or users who are members of that project. 3. Update, and Delete project are only allowed for the users assigned as project manager for each project **Functionality:** 4. **List, Search, and Filter:** A project manager should be able to search and list all projects he created, and users for projects in which they are members. 5. **Manage Projects:** The project manager should be able to create, manage, and add members to projects he created 6. **Manage Tasks:** The project manager should be able to create and manage tasks inside each project by providing the task

	name, description, status, deadline, etc. and assign them to project members
Task Management	**Access Control:** • The project manager should have full CRUD access to all information related to tasks in their own projects • Users (project members) should have Read & Update access to all tasks related to projects in which they are members, or tasks assigned to them **Functionality:** 1. **List, Search, and Filter:** Users should be able to search and view all tasks assigned to them, or task in project in which they are members, organized by project 2. **Update Task:** Users should be able to update information related to their tasks (tasks assigned to them) as well as tasks within projects in which they are members such as: ○ **Update Info:** The task information and status ○ **Manage Comments:** Users should be able to view, add, update, and delete comments about each task ○ **Manage Files:** Users should be able to view, add, and delete files related to each task
Notifications	**Access Control:** • All authenticated users: All users in any roles will have access to the notifications sent specifically to them **Functionality:** 1. **Receive Notifications:** ○ The user should be able to receive a notification when a task is assigned to them ○ The project manager should be able to receive a notification when any task status in his project(s) is updated 2. **View Notifications:** Notifications list ordered descending by date, showing status (read/unread) 3. **Notification Status:** When the user or manager opens the notification, its status should change from unread to read
Monitoring and Reporting	**Access control:** • The project dashboard should only be accessible to the project manager and members of that project **Functionality:** 1. **Project Dashboard:** Project manager and members should have access to a dashboard related to each project (which could be shown after selecting a project), showing statistical summary and aggregated values about the project. For example: ○ Number of tasks pending vs completed

	○ Number of overdue tasks ○ Number of tasks per project member ○ Etc.
Other Features	1. **Logging and Audit Trails:** The system should log any exception, or important user action in the database for audit purposes. 2. **Other:** You can add any other requirements you think is necessary to make the system more comprehensive in solving the business problem.

**This list is provided as a baseline example*

Task 3: Implement the Windows Forms Project

Learning outcomes assessed:

- *Design and implement desktop applications that access and manipulate a database*
- *Manage a multi-tier application which uses custom entity classes to access and manipulate a database*
- Propose an appropriate technology for a particular problem

The Windows Form Application should be the quick-access interface for Project Managers to manage projects and tasks, and assign them to users within their teams, and for Users to quickly update task status, and add comments. The Form Application will also implement Authentication and Authorization (Preferably, using .NET Membership), using the same user tables & passwords used for the Web Application.

List of features & requirements *:

Feature	Requirements
Authentication and Authorization	**Login:** Users should also be able to login to the system using the same email and password they use for the web application.
Project Management	**Access Control:** • All authenticated users should have access to create projects, where they will be assigned as project manager for projects they create. • Listing, searching, and reading project information are allowed for project managers or users who are members of that project. • Update, and Delete project are only allowed for the users assigned as project manager for each project **Functionality:** 1. **List, Search, and Filter:** A project manager should be able to search and list all projects he created, and users for projects in which they are members. 2. **Manage Projects:** The project manager should be able to create, manage, and add members to projects he created

	3. **Manage Tasks:** The project manager should be able to create and manage tasks inside each project by providing the task name, description, status, deadline, etc. and assign them to project members
Task Management	**Access Control:** • Users (project members) should have Read & Update access to all tasks related to projects in which they are members, or tasks assigned to them. **Functionality:** 1. **List, Search, and Filter:** Users should be able to search and view all tasks assigned to them, or task in project in which they are members, organized by project. 2. **Update Task:** Users should be able to update information related to their tasks (tasks assigned to them) as well as tasks within projects in which they are members such as: ○ **Update Info:** The task information and status ○ **Manage Comments:** Users should be able to view, add, update, and delete comments about each task ○ **Manage Files (Optional for the Form App):** Users can view, add, and delete files related to each task
Monitoring and Reporting	**Access control:** • The project dashboard should only be accessible to the project manager and members of that project **Functionality:** 1. **Project Dashboard:** Project manager and members should have access to a dashboard related to each project (which could be shown after selecting a project), showing statistical summary and aggregated values about the project. For example: ○ Number of tasks pending vs completed ○ Number of overdue tasks ○ Number of tasks per project member ○ Etc.
Other Features	1. **Logging and Audit Trails:** The system should log any exception, or important user action in the database for audit purposes. 2. **Other:** You can add any other requirements you think is necessary to make the system more comprehensive in solving the business problem.

**This list is provided as a baseline example*

Quality Standards:

Marks will be awarded for working code, but in addition to that marks will also be awarded for Code and GUI quality:

Code Quality

- Code uses the Class Library to maximise maintainability.
- The code of the applications is well-structured, organized, and easy to read and understand.
- Methods are used to eliminate repeated code.
- Class level variables are used to reduce repeated code.
- The web application views are strongly typed.
- Code is as efficient as possible to achieve the task.
- The number of database operation calls is minimized, and data is persisted where needed.
- Variables and controls have meaningful names.
- Code is commented meaningfully.
- Exception Handling/Validation is implemented.
- The system should be secure, and data should be protected from unauthorized access.

GUI Quality

- The system should be user-friendly, intuitive, and easy to use interface.
- The user interface provides all necessary functionalities.
- The web application should be responsive and work efficiently on various devices and browsers.
- Finding/Selecting data is enabled effectively.
- Filtered results are effectively displayed.
- Errors are minimised.
- Feedback is shown to the user upon successful or failed actions.
- Selected data populates related controls effectively.
- User choices are 'remembered' between form/page navigation events.
- Selected data populates related forms/pages effectively.

Note: 10% of the marks in each Application are awarded for Creativity/Innovation/Additional Functionality

Deliverables:

You need to deliver the following items as part of your project:

1. **Data Access Layer:**
    - **Database:**

      You need to create a normalized database to store data related to the entities. The database should be created using Microsoft SQL Server (either from code first or database first using SQL Server Management Studio SSMS). You should export the full database including the schema and seeding data as a (.sql) script file which includes:
        - The database schema.
        - The SQL script to create the database and all tables.
        - The SQL script to populate the database with test data.

Note: Your database must include tables used for authentication and authorization. If those are created in a separate database, then it must be submitted as well as a separate (.sql) script file.

- **Class Library:**

  You need to create a Class Library project that models your Datbase using Entity Framework Core. Your class library includes:
  - EFCore Packages.
  - Domain Classes with property annotations.
  - Database context with database connection string and DBSet collections.

2. ASP.NET Core MVC Application:

You need to create an ASP.NET Core MVC application that provides a user interface to perform the tasks mentioned in the functional requirements above under ASP.NET Core MVC heading. The application should be developed using C# and .NET 5 or 6. Deliverables include:
- The source code of the application.
- The compiled executable of the application (which is already part of the Visual Studio solution folder).

3. Windows Forms Application:

You need to create a Windows Forms application that provides a user interface to perform the tasks mentioned in the functional requirements above under Windows Form heading. The application should be developed using C# and .NET 5 or 6. Deliverables include:
- The source code of the application.
- The compiled executable of the application (which is already part of the Visual Studio solution folder).

4. Project Document (with Screenshots)

You need to create a Project Document following the **template** provided, that includes:

- The database design diagram, with short explanation or justification of the entities and relationships in case any assumptions were made.
- A sample username and password for each user role in your application for testing and demonstration purposes.
- Screenshots of the working ASP.NET Core Web Application showing different scenarios as described in the requirements and using different roles, and highlighting the approach, any additional components added, and any third-party libraries used.
- Screenshots of the working Windows Forms Application showing different scenarios as described in the requirements and using different roles, and highlighting the approach, any additional components added, and any third-party libraries used.
- Optional: You can submit a link to your group's Git-based repository (while providing necessary access), or submit a report or screenshot from the platform (i.e. Github) showing commits and contribution.

The documentation is required to evaluate your database design, test your application, and clarify your approach. It highlights the different components of your project, your quality standard, and your creative approach to solve the given problem.

Marking Criteria:

The marking criteria for the project will be as follows:

Database:

- Does the database tables design satisfy the requirements?
- Is the database design normalized and efficient?
- Are all entities and relationships correctly modelled/implemented?
- Are naming conventions for tables and columns clear? Does the database utilize auto incrementing columns, NOT NULL for required columns, and efficient column types?
- Is the database script generated and submitted with enough data to support the application testing?

[20 marks]

Class Library:

- Are all required EF Core packages installed?
- Are all entities modelled correctly into domain classes with correct types and annotations?
- Are all entities relationships correctly modelled with Navigational Properties?
- Is the DBContext class present with DBSet collections and correct Connection String syntax and placement within the project?
- Are there references included from the other projects to the Class Library project or its DLL file?

[10 marks]

ASP.NET Core MVC Application:

- Functionality (70%):
  - Does the system meet all the functional requirements mentioned?
  - Does the system work correctly and efficiently?
- User Interface (10%):
  - Is the user interface intuitive and easy to use?
  - Does the user interface provide all necessary functionalities with proper feedback to users?
  - Is the user interface clean and responsive?
  - Is the look and feel consistent across the application (fonts, button sizes, etc)?
  - Does the interface follow the GUI quality standards?
- Code Quality (10%):

  - Is the code of the applications well-structured, organized, and easy to read and understand?
  - Are coding best practices followed, such as naming conventions, commenting, and error handling?
  - Does the code follow the code quality standards?
- Creativity, Innovation, and Additional Functionality (10%)
  - Are there creative and innovative enhancements within the application (for example: in terms of user interface, functionality, decoupling, APIs, real-time components, design, architecture, security, testing, or deployment) to make it more user friendly, robust, efficient, reliable, reusable, secure, scalable, maintainable, or useful in a bigger context?
  - Is there additional functionality added based on the analysis of the baseline features and requirements given that is useful, logical, and help better solve the given business problem?

Note: In addition to evaluating your work by means of code review and live functionality testing, the system screenshots you will submit within your project document will be referred to when evaluating any of the above which you can use to clarify your approach.

[35 marks]

Windows Form Application:

- Functionality (70%):
  - Does the system meet all the functional requirements mentioned?
  - Does the system work correctly and efficiently?
- User Interface (10%):
  - Is the user interface intuitive and easy to use?
  - Does the user interface provide all necessary functionalities with proper feedback to users?
  - Is the look and feel consistent across the application (fonts, button sizes, etc)?
  - Does the interface follow the GUI quality standards?
- Code Quality (10%):
  - Is the code of the applications well-structured, organized, and easy to read and understand?
  - Are coding best practices followed, such as naming conventions, commenting, and error handling?
  - Does the code follow the code quality standards?
- Creativity, Innovation, and Additional Functionality (10%)
  - Are there creative and innovative enhancements within the application (for example: in terms of user interface, functionality, design, architecture, security, testing, or deployment) to make it more user friendly, robust, efficient, reliable, reusable, secure, scalable, maintainable, or useful in a bigger context?
  - Is there additional functionality added based on the analysis of the baseline features and requirements given that is useful, logical, and help better solve the given business problem?

Note: In addition to evaluating your work by means of code review and live functionality testing, the system screenshots you will submit within your project document will be referred to when evaluating any of the above which you can use to clarify your approach.

[35 marks]

Group Work Guidelines:

Create a healthy group environment:

- Start on time.
- Agree on preferred communication platform.
- Meet at least once a week.
- Practice respect for yourself and others.
- Come prepared to do your part.
- Be a good listener.
- No put-downs.
- Make sure everyone gets a chance to contribute or speak.
- Accept constructive criticism gracefully.
- Critique ideas, not people.
- Stay on task.
- Discuss changes and their impact on each component.
- No interruptions; let people finish talking.
- Ask for help when you're confused about what to do.
- Help others when you can.
- Do your fair share of the work.
- Review others' work

Note: The standard procedure to dealing with group members not actively responding or participating in the group work will apply. In case of group conflicts, discuss amongst your group first, and escalate to your tutor if required.

Conclusion:

The Task Management System project is an opportunity for you to showcase your skills in database design and modelling, software development, and project management. It is an ambitious project that requires attention to detail, creativity, and teamwork.

The list of entities, features, and requirements given in the project instructions is provided as a baseline with the minimum requirements to start from. However, it is encouraged that you think outside the box, research more about this problem with its different approaches and theories, and create something that solves it in an innovative way with a production-ready quality.

By the end of the project, you will have a fully functional system that enables individuals and companies to manage their tasks and projects more effectively. Good luck!