LaSalle College
Montréal

Final Evaluation: 40%

Course Identification

Name of programs– Codes:	COMPUTER SCIENCE - PROGRAMMING (420.BP) INFORMATION TECHNOLOGY
Course title:	**MULTI-TIER APPLICATIONS DEVELOPMENT**
Course number:	420-DA3-AS
Group:	03200
Teacher's name:	Houria Houmel
Duration:	Extended
Semester:	Fall 2025

Student Identification

Name: _____ Student number: _____

Date: December 15, 2025_____ Result: _____

☒ I declare that this is an original work, and that I credited all content sources of which I am not the author (online and printed, images, graphics, films, etc.), in the required quotation and citation style for this work.

Standard of the Evaluated Competencies

Statement of the evaluated competencies – Codes

Develop native applications with a database – 00SS

Evaluated elements of the competency

1. Analyze the application development project.
2. Prepare the computer development environment
3. Prepare the database(s)
4. Create the graphical user interface
5. Program the application logic
6. Control the quality of the application

Instructions

- Please upload your completed final exam entitled “Medical-Clinic”
- Your “Medical-Clinic” Application must be opened in Visual Studio 2022 beforehand
- Your Medical-Clinic” database must be open in Microsoft SQL Server 2022.
- Presentation of your session project after submitting the final exam part.

◦ No breaks are allowed during the presentation.

- It is the teacher’s responsibility to identify language errors. If such errors are found, teachers may apply a penalty of up to 10% of the grade (IPEL – Article 5.7).
- Plagiarism, attempts at plagiarism or complicity in plagiarism during a summative evaluation results in a mark of zero (0). In the case of recidivism, in the same course or in another course, the student will be given a grade of '0' for the course in question. (IPEL – Article 5.16).
-

Mark Breakdown

This evaluation is on 100 points, distributed as follows:

Q1. Prepare the development environment.	**5 points**
Q2. Program the application logic. **Class Library Project.** **Implement the database classes.**	**20 points**
Q3. Program the application logic. **Class Library Project.** **Implement the business classes.**	**30 points**
Q4. Create the graphical user interface **Windows Forms Application** **Implement the user interface classes.**	**30 points**
Q5. Control the quality of the application	**5 points**
Q6, Produce the documentation	**10 points**

Total : 100 points

Case study

Medical Clinic Management Application

Health-Care Clinic.

The **Health-Care Clinic** is a growing private healthcare institution. Currently, it faces several challenges in managing its daily operations. The administration is overwhelmed by appointment scheduling, coordinating doctors' shifts, and communicating with patients. Information is scattered across paper files and spreadsheets, slowing down receptionists' work and leading to frequent mistakes, such as double bookings or poorly coordinated doctor schedules.

To address these issues, **Health-Care Clinic** has decided to develop a centralized digital application to streamline all its operations. The goal is to implement an intuitive and secure solution to manage patients, doctors, and appointments while automating repetitive tasks as much as possible. The application will require an authentication system to ensure that only authorized users can access sensitive information. Each user will have specific roles: administrators will manage all operations, while doctors and receptionists will have limited access to their respective functions.

The system must also manage doctors' schedules.
Since each doctor has different availability and specialties, the system must prevent scheduling conflicts by blocking appointments when a doctor is already fully booked.

The clinic also expects a smooth patient registration process, collecting necessary details like name, date of birth, and contact information. The system should offer a comprehensive view of each patient, including visit history and appointment records.

In addition, the clinic aims to plan and view appointments according to available time slots, in line with its opening hours. Once an appointment is booked, it must be visible immediately to both doctors and receptionists, with real-time notifications for any changes or cancellations.

Administrators will also need the ability to generate monthly reports to monitor activities, such as the number of appointments scheduled or canceled, as well as each doctor's occupancy.

To ensure the project's longevity, the application must be designed in a modular way, facilitating the addition of new features in the future.
The clinic also expects to receive complete documentation describing the application's functionality, along with access to design diagrams and source code to allow for future adjustments or enhancements.

SUBDIVISION OF THIS SESSION PROJECT

This session project (project-exam) is divided as following :

1- Project part (30%) is completed,

2- Final exam part (40%) is to do in this evaluation

1- PROJECT PART (30%) - Done

Targeted competency elements:

1- Analyze the application development project. (00SS-1)

2- Prepare the database (00SS-3)

2- FINAL EXAM PART (40%) - TO DO

Targeted competency elements:

1- Prepare the computer development environment. (00SS-2)

2- Generate or program the graphical user interface. (00SS-2)

3- Program the application logic. (00SS-5)

4- Control the quality of the application. (00SS-6)

5- Produce the documentation (00SS-8)

Implementation of a 3-Tiers Architecture using

Windows Forms App. (.Net Framework),

C# Language,

Entity Framework

Requirements :

1- Software to use:

- Visual Studio 2022 (C# Language)
- Microsoft SQL Server 2022 (Server database)

2- Project Template to be used:

Windows Forms Application (.Net Framework)

3- Data Source to be used :

ADO .Net Entity Data Model – EF- Database First Approach

4- Project Code Source to be used:

LINQ-EF (LINQ To Entity)

5- Multi-Tiers Architecture

The Application must be structured as follows:

5.1- DAL (Data Access Layer)

- DbContext:

It contains the DbContext, which is used by the BLL to provide database connection.

Data access classes must be generated by the scaffolding.

- Database Schema:

It contains the database schema

5.2- BLL (Business Logic Layer)

The back-end of the application, It must contain all the data exchange services

- DTO (Data Transfer Object)

Defines a common object shape that is used across the application-back-end to front-end

A common data exchange language

- Services:

The Services define operations that are available to the application front-end and user

It also contains validation and data transformation logic

- Models

Model classes implementation

- Documentation:

It contains the technical documentation

5.3- UI (Graphic User Interface Layer)

- It consumes DTO
- It creates instances of Services to perform data operations

UI Layer **must be built as multi-forms graphic application**

Questions:

Q1 : Prepare the development environment. - (5 points)

1- Install required NuGet Packages.

Use Package Manager Console to install the Entity Framework package.

2- Configure Connection String (App.config) to connect the entity data model to the SQL Server database

```
<connectionStrings>
    - - - -
    - - - - -
</connectionStrings>
```

Q2: Data Access Class Development - (20 points)

Technical Requirements :

- All data access classes must be created in a folder named « DAL ».
- Data access classes must be generated by the **scaffolding**

Q3: Business Classes Development - (30 points)

Technical Requirements :

- Business service classes use data access classes.
- Database connection information must already be added to the application configuration file.
- **Comment out your source code**

Q4: Presentation classes development - (30 points)

Technical Requirements :

- All presentation classes must be placed in a folder named « UI ».
- Use of services classes and DTO classes in presentation classes.
- Use of appropriate Windows Forms controls to create a user-friendly interface
- **Comment out your source code**
-

Q5. Control the quality of the application - (5 points)

Technical Requirements :

- Te*st this 3-tier architecture application and report the test results in a file named:* « Health-Clinic_Test_Plan».
- Test all update operations at the interface level, then verify that these updates are effective in the database under SQL Server.
-

Q6. Produce the documentatiom - (10 points)

Produce the technical documentation.

This application must demonstrate a complete implementation of a 3-tiers architecture using Windows Forms, C#, Microsoft SQL Server, and ADO .Net Entity Data Model

The separation of concerns between the UI, BLL, and DAL layers ensures that the system is maintainable, scalable, and easy to understand.

The project includes full CRUD functionality.

By following the layered architecture, the user interface remains clean and independent, while all business rules and database operations are handled in their respective layers.

Evaluation grid

Question 1 — Prepare the development environment (5 points)

Competency: Prepare the computer development environment (00SS-2)

Performance Criteria	Highly Satisfactory	Satisfactory	Unsatisfactory	Highly Unsatisfactory	Total
3.3 Compliance with the data model	**2.5**— EF package installed via Package Manager Console; versioning correct; dependencies resolved; configuration verified and documented.	**1.5–2** – EF installation successful but missing minor verification steps.	**1–1.5** – EF package installed but incorrectly configured or with missing dependencies.	**0** – No EF installation or installation prevents project execution.	/5
3.2 Proper insertion of initial or test data	**2.5** – Connection string correctly structured, secure, valid, and fully functional with SQL Server.	**1.5–2** – Functional connection string with minor errors or missing parameters.	**1–1.5** – Connection string incomplete or unstable.	**0** – No valid connection string configured.	

Question 2 — Data Access Class Development - (20 points)

Competency: Program the application logic — Data layer portion (00SS-5)

Performance Criteria	Highly Satisfactory	Satisfactory	Unsatisfactory	Highly Unsatisfactory	Total
5.2 Programmation correcte des interactions entre l'interface utilisateur graphique et l'utilisateur	**10** – DAL folder implemented with required structure (Context + Models folders); naming conventions respected; clean separation of concerns.	**5–10** – DAL mostly structured correctly with minor inconsistencies.	**1–5** – DAL created but incorrectly organized or missing folders.	**0** – No DAL created.	/20
5.3 Choix approprié de clauses, d'opérateurs, de commandes ou de paramètres dans les requêtes de base de données	**10** – Context and model classes correctly generated from database; EF mappings validated; no redundant modifications.	**5–10** – Classes generated and functional with minor issues.	**1–5** – Scaffolding incomplete or incorrect.	**0** – No scaffolding performed or classes unusable.	

Question 3 — Business Logic Layer Development - (30 points)

Competency: Program the application logic (00SS-5)

Performance Criteria	Highly Satisfactory	Satisfactory	Unsatisfactory	Highly Unsatisfactory	Total
5.1 Programmation ou intégration correcte des mécanismes d'authentification et d'autorisation	**10** – All business classes properly structured in BLL\Models; classes are cohesive, clean, and follow naming conventions.	**5–10** – Structure mostly correct with minor inconsistencies.	**1–5** – BLL created but poorly organized.	**0** – No BLL created.	
5.4 Traitement correct des données de la base de données	**10** – Business services correctly consume DAL classes, applying best practices and clear separation from UI.	**5–10** – BLL uses DAL correctly with minor inefficiencies.	**1–5** – Minimal connection to DAL; poor separation of concerns.	**0** – No interaction with DAL.	/30
3.1 Création ou adaptation correcte de la base de données locale ou distante	**10** – Complete, robust CRUD (Create, Read, Update, Delete) with optimized LINQ queries, validation, and exception handling.	**5–10** – CRUD operations functional with minor issues. **3–5** – Correct configuration with minor issues.	**1–5** – CRUD incomplete or inefficient. **1–3** – Partially correct.	**0** – No CRUD logic implemented. **0** – Incorrect or missing configuration.	

Question 4 — Presentation Layer (UI) Development (30 points)

Competency: Program the graphical user interface (00SS-4)

Performance Criteria	Highly Satisfactory	Satisfactory	Unsatisfactory	Highly Unsatisfactory	Total
4.1 Choix et utilisation appropriés des éléments graphiques pour l'affichage et la saisie	**10** – Clean, intuitive, multi-form UI; consistent styling; excellent workflow and user experience.	**5–10** – Functional UI with appropriate controls and acceptable design.	**1–5** – UI incomplete, poorly organized, or difficult to use.	**0** – No usable UI created.	
4.3 Adaptation de l'interface en fonction du format d'affichage et de la résolution	**10** – All UI actions correctly interact with BLL classes; no direct DAL usage; data flow respects the multi-tier model. **10 -**Controls chosen appropriately, validation included UI responsive and professional.	**5–10** – Mostly correct integration with some minor issues. Controls used correctly with minor ergonomic issues.	**1–5** – BLL used partially; some logic improperly placed in UI. Controls misused or insufficient.	**0** – No BLL usage; logic directly in UI or missing.	/30

Question 5 — Control the Quality of the Application (5 points)

Competency: Control the quality of the application (00SS-6)

Performance Criteria	Highly Satisfactory	Satisfactory	Unsatisfact-ory	Highly Un-satisfactory	Tot al
4.1 Choix et utilisation appropriés des éléments graphiques pour l'affichage et la saisie	**2.5** – Complete test plan with detailed test cases, results, screenshots, and conclusions.	**1.5-2** – Test plan present with essential tests covered.	**1–1.5** – Partial or superficial test plan.	**0** – No test plan produced.	/5
5.4 Traitement correct des données de la base de données	**2.5** – All UI updates validated with SQL Server; BLL/DAL flow verified; bugs documented.	**1.5–2** – Most updates tested successfully.	**1–1.5** – Incomplete or partial testing.	**0** – No testing performed.	

Question 6 — Documentation (10 points)

Competency: Produce documentation (00SS-8)

Performance Criteria	Highly Satisfactory	Satisfactory	Unsatisfactory	Highly Un-satisfactory	Tot al
5.3 Choix approprié de clauses, d'opérateurs, de commandes ou de paramètres dans les requêtes de base de données	**5**– Documentation fully describes architecture, components, requirements, workflow, and installation.	**2.5 -4-** – Covers essential elements with minor omissions.	**1 -2.5** – Documentation limited or unclear.	**0** – Documentation missing.	/10
4.3 Adaptation de l'interface en fonction du format d'affichage et de la résolution	**5**– Clear, organized, professional, well-formatted.	**2.5 -4** – Understandable with some structure issues.	**1 -2.5** – Hard to read or poorly structured.	**0** – Not usable.	

Clear Communication	Clear Communication, **most of the time**	Vague Communication	Unclear Communication
- 0	- 0,5	- 1,5	- 2
(Word Choice) Use of precise and rich vocabulary	(Word Choice) Use of precise vocabulary	(Word Choice) Use of imprecise vocabulary	(Word Choice) Use of inappropriate vocabulary
- 0	- 0,5	- 1,5	- 2
(Format/Type of work) Respect of norms	(Format/Type of work) Respect of **most of the** norms	(Format/Type of work) Non-respect of the norms	(Format/Type of work) Inappropriate in relation to the required norms
- 0	- 0,5	- 1,5	- 2
(Linguistic Code) (≤2 mistakes / page)	(Linguistic Code) (3-7 mistakes/page)	(Linguistic Code) (8-10 mistakes/ page)	(Linguistic Code) (>10 mistakes/ page)
- 0	- 0,5 - 2.5	- 2.5 - 3.5	- 4