LaSalle College
Montréal

Final Evaluation: 40%

Course Identification

Name of program(s) – Code(s):	INFORMATION TECHNOLOGY PROGRAMMER ANALYST (LEA.3Q)
Course title:	IOS MOBILE DEVELOPMENT
Course number:	420-DM3-AS
Group:	07484
Teacher’s name:	Daniel de Rezende Barbosa Carvalho
Duration:	3 hours
Semester:	Winter 2022

Student Identification

Name: ____ Student number : ______________

Date : ______________________ 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 competency – 00SR

Develop native applications without a database -00SR

Evaluated elements of the competencies

1. Analyze the application development project.
2. Generate or program the graphical interface.
3. Program the application logic.

Statement of the evaluated competency – 00SX

Develop applications for connected objects -00SX

Evaluated elements of the competencies

1. Analyze the application development project.
2. Generate or program the graphical interface.
3. Program the object’s application logic and the control or monitoring application logic.

Competency: Develop native applications without a database-00SR

General ministerial and institutional performance criteria:

− Sense of intellectual curiosity
− Sense of critical thinking
− Analysis and initiative
− Sense of Organization

Elements of the competency	**Performance criteria specific to each element**
1. Analyze the application development project.	1.1 Accurate analysis of design documents 1.2 Proper identification of tasks to be carried out.
2. Prepare the computer development environment.	2.1 Proper installation of software and libraries on the development platform 2.2 Proper configuration of the target platform 2.3 Proper configuration of the version control system 2.4 Proper importing of the source code
3. Generate or program the graphical interface.	3.1 Appropriate choice and use of graphic elements for display and input 3.2 Proper integration of images 3.3 Adaptation of the interface based on the display format and resolution
4. Program the application logic	4.1 Proper programming of interactions between the graphical user interface and the user 4.2 Proper programming of communications between the peripheral devices and the software functions of the target platform 4.3 Effective use of execution threads 4.4 Proper integration of sounds and videos 4.5 Proper application of internationalization techniques 4.6 Precise application of secure coding Techniques
5. Control the quality of the application.	5.1 Precise application of test plans in the emulator and on the target platform 5.2 Thorough reviews of code and security 5.3 Relevance of the corrective actions 5.4 Compliance with issue tracking and version control procedures 5.5 Compliance with design documents

6. Participate in the deployment of the application.	6.1 Appropriate preparation of the application in view of its deployment or installation 6.2 Proper deployment or installation of the application
7. Produce the documentation.	7.1 Proper identification of the information to be written up 7.2 Clear record of the work carried out

Competency: Develop applications for connected objects. -00SX

General ministerial and institutional performance criteria:

− Sense of intellectual curiosity
− Sense of critical thinking
− Analysis and initiative
− Sense of Organization

Elements of the competency	**Performance criteria specific to each element**
1. Analyze the application development project.	1.1 Accurate analysis of design documents 1.2 Proper identification of tasks to be carried out.
2. Prepare the computer development environment.	2.1 Proper installation of software and libraries on the development platform 2.2 Proper configuration of the target platform 2.3 Proper configuration of the version control system 2.4 Proper importing of the source code
3. Generate or program the user interface.	3.1 Appropriate choice and use of graphic elements for display and input 3.2 Proper integration of images 3.3 Adaptation of the interface based on the display format and resolution

4. Program the object's application logic and the control or monitoring application logic.	4.1 Proper programming of data gathering, processing and transmission instructions 4.2 Proper programming of interactions between the interface and the user 4.3 Appropriate use of data exchange services 4.4 Proper application of internationalization techniques 4.5 Precise application of secure programming techniques 4.6 Proper transfer of the application onto the connected object 4.7 Compliance with time constraints
5. Control the quality of the application.	5.1 Precise application of test plans 5.2 Thorough reviews of code and security 5.3 Relevance of the corrective actions 5.4 Compliance with issue tracking and version control procedures 5.5 Compliance with design documents
6. Participate in the deployment of the application.	6.1 Appropriate preparation of the application in view of its deployment or installation 6.2 Proper deployment or installation of the application 6.3 Proper execution of pre-operational tests 6.4 Suitability of the information transmitted to users regarding
7. Produce the documentation.	7.1 Proper identification of the information to be written up 7.2 Clear record of the work carried out

Instructions

- Your exam must be submitted by uploading your project via Omnivox. Deadlines are shared on Omnivox in the assignment box and must be respected.

- The submission will happen in two parts:

  ● The first assignment, named “**final exam app code**”, should receive all your source code files (compress before sending).
  ● The second assignment, named “**final exam app video**”, should receive the .mov file showing that your application is working as expected.

- Your project name should be defined as “**FINALEXAM_[STUDENT NUMBER]”.** (E.g.: **FINALEXAM_1234567**)

- It is the teacher’s responsibility to identify language errors. If such errors are found, teachers may deduct up to 5% of the final 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:

Question 1	100 points
TOTAL:	**100 POINTS**

Question 1

Company ABCD is developing an application to manage its employees. This application will be used exclusively by the Human Resources team. The goal for this first version is to develop the user interface including the capability to list, insert, update, and delete employees. Information should be saved as permanent data using Core-Data.

Your task is to build this IOS Application, following the documentation and layouts given below.

IMPORTANT

○ Respect the MVC (model-view-controller) design pattern.

○ Build a clean and readable code. Use well-defined variables, UI objects and function/method names.

○ Present clear messages to the user.

□ **Question 1 – Part 1**	20 pts	Core-data entity definition, subclass extension, and implementation

□ Entity name:	**Employee**	
□ Attributes:	□ **uuid**	UUID?
	□ **name**	String?
	□ **email**	String?

□ Using CoreDataProvider and CoreDataProviderProtocol, create an **extension** for the class and implement the methods:

□ static func **all** (context : NSManagedObjectContext) -> [Employee]

To list all existing records.

□ func **save**(context : NSManagedObjectContext) -> UUID?

To insert/update a record.

□ func **delete**(context : NSManagedObjectContext) -> Bool

To remove a record from the database.

□ Question 1 – Part 2 | 20 pts | User interface design and navigation

□ Design the application using **main.storyboard** and following the given layout:

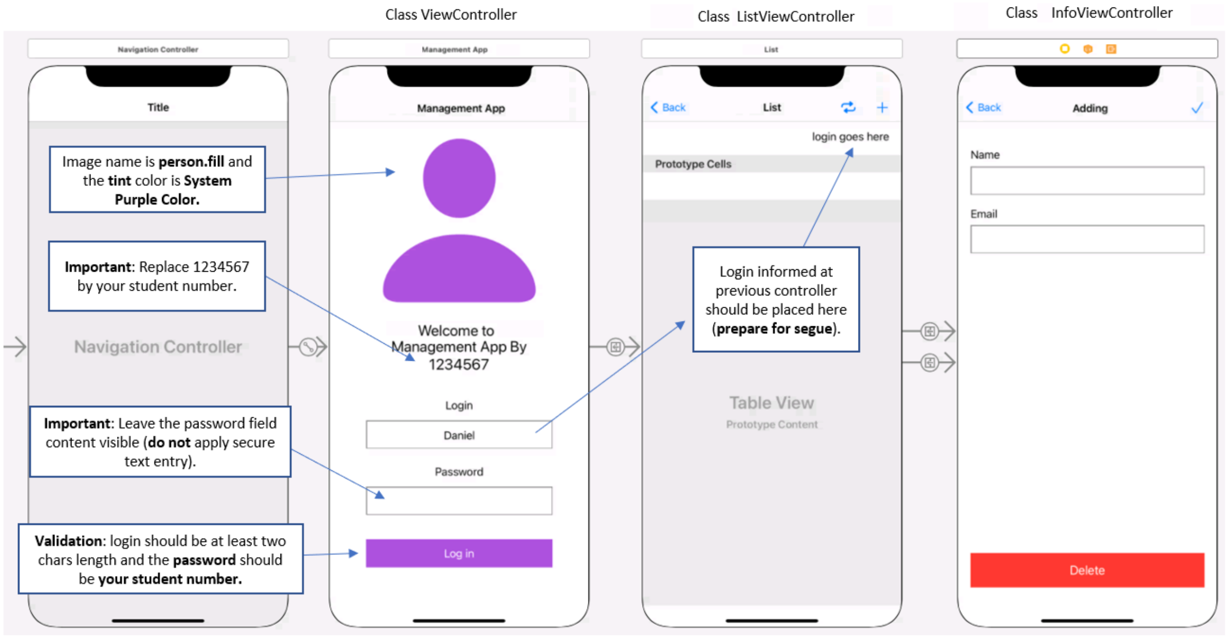

Image name is **person.fill** and the **tint** color is **System Purple Color.**

Important: Replace 1234567 by your student number.

Important: Leave the password field content visible (**do not** apply secure text entry).

Validation: login should be at least two chars length and the **password** should be **your student number.**

Login informed at previous controller should be placed here (**prepare for segue**).

□ Create the respective cocoa-touch classes and assign them to each controller (ListViewController and InfoViewController).

□ Create file **enumSegues.swift** inside a group/folder named **"Source"** and define the values for your Segues.

□ When UI is finished, remember to apply proper constraints (application must be responsive).

□ Include the **Toast** class file in your project and use the proper method to show messages to the user.

□ Question 1 – Part 3 | 10 pts | Coding ViewController

□ Implement the code to validate the user entry (login and password). Make sure you are following the validation instructions written inside the boxes (image above).

□ You should send the user name to ListViewController and replace the label text "**login goes here**" with its value.

□ **Question 1 – Part 4**	20 pts	Coding ListViewController

□ Implement the protocols **UITableViewDelegate** and **UITableViewDataSource**.

□ Each row of **tableView** should present an **employee’s name**.

□ Implement the bar button “ + ”, sending the user to **InfoViewController** to add new employee. The **button delete** should be **hidden** at this time.

□ Implement the bar button “ ⇄ ”, **refreshing** the **tableView** data.

□ If the user selects a row, you should show **InfoViewController** presenting the employee information.

□ **Question 1 – Part 5**	30 pts	Coding InfoViewController CRUD (create, read, update, delete)

□ Implement the code to handle the two operation modes available inside this activity - **insert** or **edit/delete**.

□ Show the correct title for this controller depending on the operation (**adding** or **showing**)

□ Implement the button “ ✓ ” to save/update information.

□ If the user is on **EditMode**, the **delete button** will be available. Implement this action, sending the user back to the previous view controller if the operation succeeds.

Correction Grid

Elements of the competency:
- 00SR : 1. Analyze the application development project.
- 00SX : 1. Analyze the application development project.

Performance criterion:

1.1 Accurate analysis of design documents
1.2 Proper identification of tasks to be carried out.

- 00SR : 3. Generate or program the graphical interface.
- 00SX : 3. Generate or program the graphical interface.

Performance criterion:

3.1 Appropriate choice and use of graphic elements for display and input
3.2 Proper integration of images
3.3 Adaptation of the interface based on the display format and resolution.

Criterion-elements:

Proper UI design using given asset files, following prototype images (object colors, sizes, margins, paddings, alignments).

Correct application of constraints adding proper responsiveness to the user interface.

UI objects identification following good practices and standards.

Well-defined messages and instructions when validating user input.

The right use of Segues, Navigations and Protocols to send and refresh data between view controllers.

Question 1	Highly satisfactory	Satisfactory	Unsatisfactory	Highly unsatisfactory	Total
Part #2	20 Completely correct	12 ..< 20 Almost correct	4 ..< 12 Incomplete implementation	< 4 Completely wrong	**/ 20**

Elements of the competency:
- 00SR : 1. Analyze the application development project.
- 00SX : 1. Analyze the application development project.

Performance criterion:

1.1 Accurate analysis of design documents
1.2 Proper identification of tasks to be carried out.

Elements of the competency:
- 00SR : 4. Program the application logic.
- 00SX : 4. Program the object’s application logic and the control or monitoring application logic

Performance criterion:

4.1 Proper programming of data gathering, processing and transmission instructions
4.2 Proper programming of interactions between the interface and the user
4.3 Appropriate use of data exchange services
4.4 Proper application of internationalization techniques
4.5 Precise application of secure programming techniques
4.6 Proper transfer of the application onto the connected object
4.7 Compliance with time constraints

Criterion-elements:

Use of variables, classes, protocols, enums, and function names following standards.

Field validation following given instructions.

Appropriate use of Segues and its functions to validate and send data between view controllers.

Proper use of protocols and stubs implementation.

Correct implementation of core-data entities. Methods insert, update, delete well defined.

Well-defined event handlers for user interactions (gestures, dialog actions).

No syntax or compilation errors (application running). No logic errors.

Question 1	**Highly satisfactory**	**Satisfactory**	**Unsatisfactory**	**Highly unsatisfactory**	**Total**
Part #1	20 Completely correct	12 ..< 20 Almost correct	4 ..< 12 Incomplete implementation	< 4 Completely wrong	**/ 20**
Part #3	10 Completely correct	6 ..< 10 Almost correct	2 ..< 6 Incomplete implementation	< 2 Completely wrong	**/ 10**
Part #4	20 Completely correct	12 ..< 20 Almost correct	4 ..< 12 Incomplete implementation	< 4 Completely wrong	**/ 20**
Part #5	20 Completely correct	18 ..< 30 Almost correct	6 ..< 18 Incomplete implementation	< 6 Completely wrong	**/ 30**

Correction Grid for Language

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