MADS4005 PROJECT DESCRIPTION
(35% of overall course grade)

SUBMISSION REQUIREMENTS

1. Compress your project files and any related assets into a zip file
2. Name your zip file: `advancedios-project-petersmith-10123456.zip`

   Substitute `petersmith` and `10123456` with your student name and id.
3. Upload the zip file to Blackboard by the date specified in the dropbox.

Please note that a submission will not be graded if:

1. The project does not compile
2. The project is missing required files or assets
3. The submission is late
4. The submission contains academic dishonesty.

ACADEMIC INTEGRITY

This assessment must be completed individually.

Providing **any help** to other students is **not permitted**.

Help includes, but is not limited to:

- Exchange of ideas, advice about how to approach the problem
- Sharing website links or reference materials
- Sharing of devices, such as laptops, physical IOS devices, or MacInCloud accounts
- Sharing of credentials to college provided or personal email accounts, Google Drives, Dropbox, Blackboard, or similar

Communication with individuals either in the class, or outside the class, is strictly prohibited.

Any learner suspected of breaking the above rules will not have their submission marked; and, a report will be filed with the College's Academic Integrity department.

To prevent issues of academic integrity, it is imperative that you are vigilant that your code is not duplicated, in part or whole, by another learner. Duplication can mean: identical code; or code that is semantically identical in naming and overall construction. Should the instructors discover academic dishonesty as defined above and per the College Academic Integrity Policies, **a grade of 0** will be assigned to all parties involved.

PROBLEM DESCRIPTION

Live Fit Food (https://livefitfood.ca/) is a popular Canadian meal delivery company that prepares, cooks and delivers healthy meals straight to your door. You have been contracted as a software developer to create an IOS app for the company.

The company provides 7 days worth of meals in each meal kit.

DATABASE SCHEMA

You are responsible for developing a reasonable database schema for this app. You must demonstrate that you know how to create relationships between tables. Thus, a minimum of two (2) entities must be joined using a relationship.

All database features must be implemented using CoreData.

FEATURES

1. USER SIGNUP AND LOGIN

You must provide the necessary user interfaces to signup and/or login a user. User details should be stored in CoreData.

Users login with their email address and password.

Each user must have:

- Email Address
- Password
- Profile Photo
- Phone number

A profile photo can be selected from the device's photo gallery or by taking a photo with the camera.

For this project, you only need to demonstrate that you were able to save the photo to CoreData. You can do this in a couple ways:

- Saving the actual image file to Core Data
- Saving the path to the image in Core Data

You do not need to use the photo anywhere else in the app.

2. ORDERING A MEAL KIT

Data for meal packages are ***Initially*** provided by a static JSON file. This file is included with the app at the time of install.

Upon first install, the app should read the initial data from the JSON file and store it in CoreData. From this point forward, the meal kit data should be read from CoreData.

You must provide a minimum of 4 meal kit options for the user to choose from.

Each meal kit should have:

- Name
- Description
- Price
- Photo
- Calorie count

Your app must include all necessary images to properly display the meal kit in the user interface.

For simplicity, you may assume that the user only orders 1 meal kit at a time. You may also assume that each meal kit contains 7 days worth of meals. *This statement is given so that you can appropriately price your meal kits. For example, it you set your pricing to be $5 / kit, then this means that the user is getting 7 days worth of meals for $5. That's very generous (and unrealistic!)*

- STRETCH GOAL: If you have the time and inclination, you may update your app to allow for an order to have multiple meal kits of varying quantities. However, this is not part of the official project requirements, and no additional marks will be awarded for deducted for implementing it.

You must also provide a way for the user to enter any valid coupon codes. See further in the document for more details.

3. PICKING UP THE ORDER

Because of logistical issues, you must pick up your meal kit the same day you order it. The company only starts preparing the meal kit when you are located within 100 meters of the store. It takes 15 minutes for the store to prepare your order.

The app must use the device's GPS or Wifi to automatically detect the user's current location and validate whether the company has started the process of preparing the kit. When the user enters within 100 m of the store, an Alert Dialog is displayed, indicating that the preparation of their order has started. After 15 minutes have passed, the user receives another update indicating that the order is ready to be picked up. You should display this notification in an Alert Dialog.

During your project demo, you may manually set the timer to be a smaller value (such as 1 minute). This will allow the instructor to evaluate your feature without having to wait the full 15 minutes.

4. RECEIPT

The app must display a confirmation receipt of the order.
The receipt should show:

- The name of the meal kit and its SKU
- The subtotal
- The tax (13%)
- Option to add a tip. The user can select from the following values: 10%, 15%, 20% , Other.
    - The person chooses **Other** if they want to manually select a tip amount.

5. ORDER HISTORY:

This screen shows a list of all the past orders the user has made.

6. SCRATCH CARD:

Once a day, the company has an incentive where users have the opportunity to win a coupon that gives them up to 50% off their next order. You may assume that the user only plays once per day.

The user must shake the phone 3 times, at which point the app randomly generates a random discount amount of either 0%, 10% or 50%.

The probability of winning a coupon is as follows:

- 5% chance of getting a 50% off coupon
- 30% chance of 10% off coupon

If the user shakes the phone and wins a coupon, they are shown a coupon code that can be used on their next order.

The coupon code must be randomly generated and cannot be a duplicate of any previously issued coupon codes.

Once the code is used, it is no longer valid on other purchases.

For project demonstration purposes, you may manually set the time to be around the same time as your project demo. This will simply be the process of showing to the instructor that the feature works.