CROSS | OVER

Instructions

- Try to complete as much as possible within the given time frame. If you need more time, please ask for an extension. You must complete full-functionality of the application with industry-level coding style/commenting. Unfinished assignments will not be considered.
- Please note that you are expected to work on the assignment independently. Discussing assignment details with colleagues or any indication of outside help will be considered cheating.
- Please do not expect too much hand-holding as this is an evaluation assignment.
- Read the complete assignment before you start. Understand clearly what is required so that your work will be appropriate and easier.

Preconditions

1. You should work on local machine.
2. Tools and Technologies
    1. iOS environment:
        1. Swift 3 or later.
        2. Xcode 8 or later.
        3. iOS 8 and above.
        4. Cocoapods.
        5. Unit tests.
        6. UI unit tests.
    2. Android environment:
        1. Java 8.
        2. Android Studio 2.2.2 or later.
        3. Gradle.
        4. Unit tests.
        5. UI Unit tests.
    3. Server environment:
        1. Node.js.

        2. Npm.
        3. Mongodb
    3. Do not use any proprietary technology or tools that are not available for evaluation.
    4. Feel free to use any open source/free framework to aim in the application development.
    5. Do not replace any of the minimum required technologies/tools listed above.
    6. THE DELIVERY WILL NOT BE EVALUATED IF YOU FAIL TO FOLLOW ANY OF THE ITEMS ABOVE.

Objective

The Saitama prefecture wants to encourage the local people as well as tourists to use bicycles for transportation. Cycling has many benefits, e.g decreasing stress, improve mobility, reduce anxiety, increase cardiovascular fitness and so on. You are responsible for creating a mobile application to connect to Jitensha Saitama REST service v2.0, providing an amazing experience to your users.

Functional Specifications:

You must to create an application implementing all the requirements below:

1. Registration screen:
    1. User's email.
    2. User's password.
2. Login screen:
    1. User's email.
    2. User's password.
3. Payment history screen:
    1. Credit card information.
    2. Place information.
4. The app must get all available bicycle rental places near Saitama prefecture, present them within a mobile map provider, showing only their names and marks:
    1. Place's name.
5. The integrated map solution can be:
    1. Android Google map.
        1. https://developers.google.com/maps/documentation/android-api
        2. https://developer.android.com/training/maps/index.html
    2. iOS mapkit.

    1. https://developer.apple.com/reference/mapkit

6. The user can rent a bike just by touching one of the available places and select Rent operation, providing his/her credit card information as well as place's id.

Non Functional Specifications:

1. The access token never expires.
2. The application must handle possible internet connective issues.
3. If possible sensitive data must be stored safely.
4. All the communication must be use https channel.

Assumptions:

1. The backend portion is based on Node.js and mongodb technologies:
    1. Download the zip package from this link.
    2. Please follow the instructions within README.md documentation to put the service up and running.
    3. To get more information about rest api, please refer to the doc folder.
2. Do not change the JSON format.
3. There's no UI specification, but it requires at least an acceptable interface with easy navigation.
4. The expected solution must be coded and designed with these items in mind:
    1. Abstraction.
    2. Modularity.
    3. Extensibility.
    4. Information Hiding.
    5. Security.
    6. Maintainability.
    7. Extensibility.

Deliverables

Application Demo:

Record a video demonstration of your work using a screencast tool like screencast-o-matic (or any other tool you prefer) commenting on the application execution. Save the video to your local machine and include it with the delivery package.

PS: Try to cover all possible flows including the corner cases, validations and so on.

Design Document:

A document (PDF preferred) containing the following information and diagrams:

1. List of technologies and design patterns used.
2. Explanation of the architecture/design implementation, e.g architectural code organization.
3. Diagrams for component interaction, activity, and sequence of the important components.

Source code:

Please provide all necessary files to proper build and run the solution.

To be evaluated

1. Efficacy of your submission: fundamentally how well your solution is able to achieve the assignment objective and solve the stated problem.
2. Code quality
    1. Code modularity.
    2. Application organization across files and within each file - please ensure you follow the framework standards.
    3. Code documentation - balancing between self documenting code and comments.
    4. Unit and integration testing.
    5. Exception handling where available and expected in the frameworks you're using.
    6. For any technology used, the correct usage of that technology based on established best practices.
3. Design:
    1. Clarity and completeness of design documents.
    2. Fitness of solution to problem.
    3. Efficiency of communication flows between frontend and backend.
4. Functional completeness.
5. The application must be compiling with successful unit testing execution, covering the requirements.

Delivery / What to submit

Please, read and follow this section carefully. Any delivery that does not follow this section will score much less or simply won't be evaluated.

First of all, review Delivery Instructions (Sent to your personal mail), which describes general delivery process. Delivery for this assignment should consist of: Archive named <your_name - SA_<technology>>.zip containing the following.

1. Source code.

2. Video demonstration.

3. Design documentation containing architecture as well as designing information and diagrams.

4. A readme file with run/install instructions plus any assumptions you made to fill in unclear requirements or any issues you may find.

Structure of the resulting zip file should be of the following format:

<your_name - SA_<technology>>.zip
\Design.pdf <<< or any other preferred format
\readme.txt <<< or any other format
\Source <<< containing all necessary code to run the solution
\Demo <<< containing your video