BSc (Hons) Computing
19.1

IOS Application Development-301CEM
Course Work

Faculty of Engineering, Environment and Computing
School of Computing, Electronics and Mathematics
Coventry University

School of Computing and Engineering
National Institute of Business Management
Colombo-7

ASSESSMENTS 1: FINAL INDIVIDUAL PROJECT (100%)

Module Title	Final Project	**Module Code**	NIB301CEM
		Final Project (100%)	
		100% Complete the final project and Upload the project into (student) GitHub profile (Make Public)	

Details of the task

Scenario:

The current situation of the world has caused problems within most of the global companies. The NIBM management has suggested a solution to this covid-19 situation. Main intention of the solution is to enable the staff to work and continue the lectures without any conflict. The application will notify if there is any student or staff member with a high body temperature or infected nearby.

How will this solution work?

1. Students will be able to continue their lectures as before
2. Management will get an idea about infected students and staff
3. University can help quarantine students and staff
4. University can easily identify the risk of the current situation

How many parties will use this application?

1. Student
2. Academic staff
3. Non – Academic staff

How will the app work?

Once user open the application, there will be bottom navigation tab bar with 3 buttons. Home, plus button and Settings tab. Default tab is Home Tab.

1. Home Tab displays
    a. A summary of the infected user(chart)
    b. Infected user's location pinned in a map
    c. Any news from university
    d. Request for help
2. + Button
    a. User can update his/her Body Temperature
    b. User can update his/her situation (5 question survey)
    c. User can create news (only staff members)
3. Settings Tab displays
    a. Create new user account (not displayed for logged in Users)
    b. Contact us
    c. Share with friend
    d. Survey Results (only for staff role can see this)
    e. User Profile
        i. Name
        ii. Address
        iii. Student index or Employee code
        iv. User Image

User will get an authentication screen once user clicked on plus button. And can login to the app by settings page directly.
Once user has logged into the system, user should complete the registration process. When you register to the application there will be two type of user roles (student and staff).

App should get users current location and send a danger alert if there are any infected people nearby. The app should make use of the sensors on the iOS device to make the user's interaction with the app more engaging. The types of sensor-related activities could be to allow the use to take and upload pictures related to user details, use the GPS sensor to determine the current location of the users and inform if there any infected people nearby.

Special role can see all the survey results in table view in settings page. You can filter those results using date and danger.

Tip: **Create score mechanism with survey. Every time when user complete the survey you can calculate the weight of the given answers.**

An iPhone only application (universal app) needs to be developed which consists of below pages.

01. Login Page.
02. Home Page (Summary View)
03. Settings Page.
04. Plus Button Screen

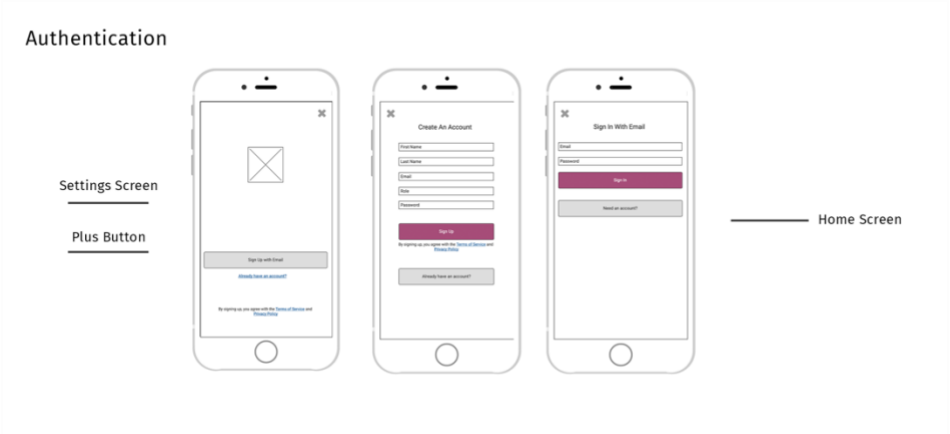
Instructions

Project Name: - YourName-IndexNumber.
App Name: - NIBM COVID19.
iOS Version: - iOS 11 to upwards. (Don't use swiftUI)
Use Firebase to integrate login and storage
(Firebase: - https://firebase.google.com/)

The app should include the following. You have the freedom to include additional functionality to improve the app.

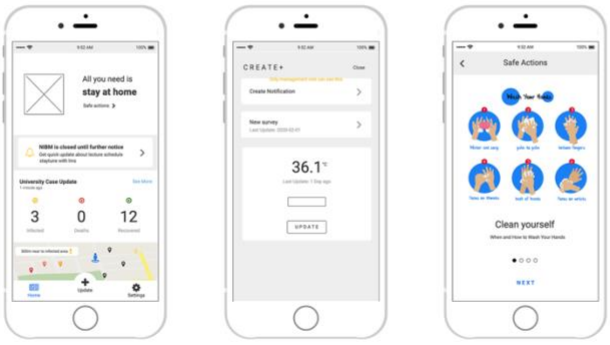
Instruction

You are free to add any additional screen to the application according to the scenario. Use the below application UI Sketch for organize your UI component.



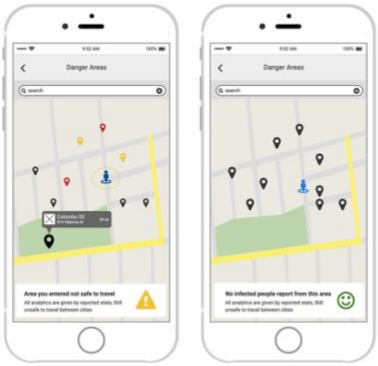
Home

Splash Screen



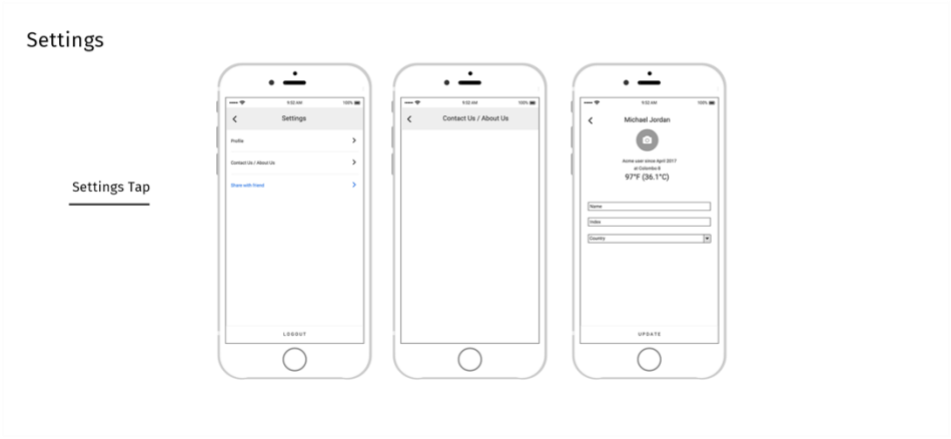
Map Full View

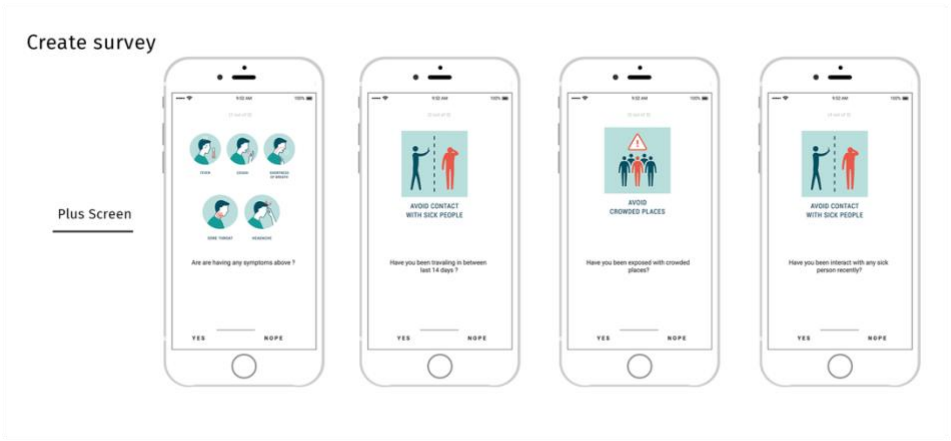
Map View From Home Screen



All Notification

Notification Cell From Home Screen





App UI

- Need to design an **unique UI** for the app screens (All screens and Pages)
- You have the **freedom** to use any color, fonts, and sizes in order to make the app more colourful and beautiful.
- Should be compatible for **iPhone 6 or later** devices.
- Should follow apple Usability Guidelines, Designing Interfaces.
- **Don't use** any third-party **UI libraries**.

UI and Unit Testing

- Write UI and unit testing for the created methods, make sure to consider the authentication flow. User login should contain password and email validation unit testing. There should be UI Testing for the Whole login screen. Such unit testing should be automated.

Submission

Students need to create their own GitHub account and create a public repository. Students should commit daily in order to prove their work. Final project repository URL should be emailed before the deadline date.

Application Profiling will evaluate from the VIVA, Student should able to explain and simulate the memory profiling

Marking scheme

Grade	Analyze the platform components and their uses across multiple devices and formats (30/100)	Design software suitable for mobile architectures (20/100)	Develop apps that can interact with external APIs and devices (20/100)	Develop apps that can communicate with sensors built into the phone hardware (30/100)
Late submission	0	0	0	0
marks <40	The gathered requirements are irrelevant or mainly incomplete. Wrong analyze work with interface builder and UI Component. Poor usage of the platform components. Poor understanding of the scenario, which is reflect in the documentation.	Incomplete flow of the application. Did not use proper design patterns. Lack of importance for creating flow for the application. Not creating correct Storyboard flow of the application design. Not following the any of the mobile architectures	Incomplete API client service. Unable to complete external API using given platform (firebase).	Does not make use of the sensors build into the iPhone to aid the experience of the user. Not following the guidelines. Application does not interact with the hardware features in the application.
Marks 40-49	Scenario is understood to an extent which is acceptable. Relevant platform component usage is seen. Analysis of the work with interface builder and UI component. The application works across multiple devices and formats.	App Flow is created but not in much detail. The visibility of the process is unclear in the flow, but overall flow of the app design is acceptable. Not following the best practices.	You include some the appropriate API, which are used in a meaningful way. Unnecessary API calls found.	Work with one sensor in the device in a basic manner. The sensor that is used is suitable for the requirements, but it does not really improving the engagement of the user.
Marks 50-59	Correct requirements analysis, usage of platform component, interface builder and UI component. You could have considered some interesting functionality.	App flow is at a good standard and shows clear design concepts. You follow some of the best practice and some of the design approaches given below • MVC • MVP • MVVM • VIPER	You cover a good selection of the API concepts and show a clear idea of appropriate https methods. There is the correction of errors and success states handling. There is Model creating for each data object form.	A single sensor is used in the application. You follow the correct guidelines when using the hardware features of the application. The interaction with the sensor is smooth and it clearly improves the engagement of the user.
Marks 60 – 69	Proper understanding of the UI component and interface builder. Consideration of platform components and their use across devices and formats. You follow IOS design guidelines for interfaces. Clear scaffolding for UI component.	The apps makes use of appropriate methods and code organization. The design approach is reflected in the organization of the app.	A very good understanding of how external API can be used to aid the development of the app. There is correct connection with the API, manipulation of data using API and careful consideration of when to call the API.	Usage of two sensors that align with the application requirement. There is correct integration with the app to engage the user.

	Common usage for reuse component. Representation of the user's interaction with data fields and inclusion of user interaction validations.			
Marks > = 70	Component reuse, creation of common methods for render UI component that are used in multiple areas. Following of IOS design guidelines for interfaces when working with multiple device and formats. Manageable code structure when designing Interfaces. Contain App theme in separate file. # Organized scaffolding for UI component. Clear usage of interface builder (easy to understand and maintain) meaning full validation for failed user actions.	The design reflects the overall architecture of the app. Code produces an app that is easy and to update. Each method is testable using an appropriate testing strategy.	App reflect common API connection and error handling for all the API responses. Before app connects to the API you make mock data objects. This should be reflected in Github. Checks internet connectivity. Write tests for API integration. Reflect API call failed in proper meaning full message. Stop unnecessary API calls when user navigate through the screens.	The app incorporate a good selection of sensor based activities. They feel part of the app rather than add-ons. The data collected from the app is clearly integrated to improve the user experience.

Component grades

Breakdown of marks –

Analyse work with interface builder and UI Component	**10**
Follow IOS design guidelines for interfaces	15
Analyse the platform components and its uses across multiple devices and formats.	15
Design software should be suitable for mobile architectures and follow best practices.	10
Develop apps that can interact with external APIs and devices.	10
Analyse the idea of internal storage and notification system	15
Develop apps that can communicate with sensors built into the phone hardware.	15
Develop software that can be easy to maintain with UI/Unit testing	5
Make the app suitable for app store (Performance testing)	5