DSW02B1 Group Project
Practical Work

Lecturer: Mr. KR Mabokela & O. Giwa
School of Consumer Intelligence and Information Systems,
Department of Applied Information System, University of Johannesburg.
Semester: July – December 2021
Total Marks: 100
Due Date: 07.11.2022

App Development Scope

Questions

The team project (Must strictly be **5-6** students per team) for this course involves conceptualizing, Analysis, designing, and Implementation of a mobile application running on either Windows, Android, or iOS SDK platform. Your need choice one platform that you are comfortable with and use it to develop you are Mobile Application. You may use languages such as React, React Native, Ionic, Java, flutter, Kotlin, Xamarin, etc. The application should meet a set of functional and non-functional requirements as per the description below. The teams should follow a well-managed development process that results in user-friendly, well-designed and well-documented Mobile software.

High-Level Requirements

In this day of our lives, we all can agree that the most people prefer to use mobile device more than desktop computers. Mobile devices serve as three things which are; to serves as your gateway to everything else; your daily entertainer and lastly, it represents you and your social behavior. Thus, the Mobile App that you need conceptualize and develop in this module must be represent your team values, must represent you, and most importantly be unique to you.

In addition, the Mobile App must meet the following requirements:
- Must uniquely meet a clear need and solution;
- Must be well-designed and user-friendly,
- Must have a creative user interface
- Must be innovative addressing a specific problem,
- Must have at least three domain objects,
- Must have data that persists throughout multiple user sessions,
- Must have be able to store and retrieve information to the database;
- Must have the ability to work offline (**Extremely important**);

– Must be compatible to the recent mobile devices,

– Must use at least one Internet-based service (e.g., YouTube, social media (such as Twitter, Facebook), Firebase, or

– Geolocations services - Google Maps, Twitter data analysis

– Dashboards

– Must use at least one device sensor (e.g., scanners, Camera, microphone, GPS, accelerometer, or light sensor)

– Must have API integration or other external services.

NB: All groups or teams must design an original distinct Mobile Apps. There should NOT be multiple groups or team Mobile Apps that meet the same needs or similar problem. The expectation that each group designs and develops an original Mobile App that is distinct from Mobile Apps that are already available on the market as well as other groups. Since the project entails designing and implementing an Innovative Mobile App that represents you, use this opportunity to showcase your innovative ideas, creativity in your mobile app design. In addition, your **mobile app may be a game** that meets the above requirements and originality considerations.

STRICLY PROHIPITED: Do not buy a project from anyone for marks. You will be sad to get zero mark.

Evaluation Criterion

Criterion (1): Environment Setup and Running Mobile App

Evaluation Criteria: The mobile app executes via the platform emulator (i.e., Android, iOS, or smart mobile devices).

Criterion (2): Mobile App Requirements and Design Document

This Criteria entails envisioning and designing a mobile app that meets the requirements specified in the High-Level Requirements section.

Submit: A SINGLE design document using Latex writing system that shows:

- Write details of your app conception, specifically
- Narratives (one or two paragraphs describing how someone uses your app and why your mobile app is necessary to achieve certain goals);
- A UML class diagram of your mobile app's domain model designed following the object-oriented approach discussed in the lecture session;
- A categorized list of use-cases describing users' interactions with the app (for example categories can include "Account MANAGEMENT," "Mapping," and so on);
- Your mobile app's relational model that describes tables and relations in your relational database schema (you need to include the relational model in your document, not just an ER diagram);
- Sketches (drawings of people using the mobile app as you envision, how the app connects to external services, and so on);
- Screen layout mockups; or
- Screen flows showing how clicking different User Interface elements navigates the user among mocked-up screen layouts.

Note: Your screen layout mockups OR screenshots, screen flows, and sketches may be drawn by hand, but they ***MUST be legible.***

Showcase: Your mobile app design document with the information above. You should explain your app to the lecturer during your group's meeting/presentation.

Evaluation Criteria: ***Clear evidence of the object-oriented design process***.

Criterion (3): Installing, Debugging, Profiling, App Lifecycle Management

This checkpoint involves demonstrating your ability to do the following:

- Installing your app on a mobile device;
- Debugging (e.g., stepping through source code and setting breakpoints on your app);
- Profiling your app using one or more profilers (e.g., Visual, Android Studio's, or memory profilers or Instruments for iOS);

– The mobile app lifecycle and logging. Here you must implement a simple part of your mobile app and demonstrate using the log file that you can trigger the app's lifecycle methods in Activities.

Show:

– A screenshot of at least one type of profiler (e.g., CPU or memory footprint in Android Studio, Visual Studio or XCode);
– Screenshots of the mobile app's lifecycle methods being triggered.
– Screenshots of debugging your app in Android Studio, Visual Studio code or Xcode (specifically, setting a breakpoint, viewing variables' states, and stepping through code).

Evaluation Criteria:

– Students demonstrate understanding of IDE use;
– Students demonstrate comprehension of the Android/iOS/Windows app/Cross-Platform of your choice's lifecycle.

Criterion (4): Database Integration Requirements

You need to show evidence of apps ability to perform CRUD components.

Submit:

– Screenshots of your mobile app's data storage schema
– Screenshots of your mobile app's class organization;
– Screenshots of a version control repository for your mobile app's source code.

Showcase:

– Your app's data storage schema;
– Your app's ability to perform creation, retrieval, updates, and deletions (CRUD) with your data store (an on-device relational database or an external service such as Firebase);
– The version control repository for your app and layout of your mobile app's classes.

Evaluation Criteria:

– Students demonstrate implementation of their mobile app's data storage schema (using a relational or any SQL databases);
– Students demonstrate that their mobile app can perform CRUD operations on stored data;
– Students demonstrate use of a version control system.
– You learnt CRUD during first semester, you need apply all the operations.

Criterion (5): Functional Demonstration of Your Mobile App

For this checkpoint, you must demonstrate a functioning mobile app. The app should work from end to end. ***Do not worry*** about the performance of the following; network connectivity, GPS signal reception, and so on.

Submit:

– A list of use cases and use case diagram
– Screenshots of your app running.

Showcase: Your mobile app working on a mobile device. (If it works only on an emulator, you will only receive partial marks.)

Evaluation Criteria: Check working use-cases

Criterion (6): Demonstration of Your Mobile App's Non-Functional Requirements

This checkpoint entails showing your app's non-functional capabilities such as

- usability,
- performance,
- availability,
- maintainability,
- modifiability, and

- scalability.

The app should work from beginning to end, must be robust enough to work under failures (such as loss of network connectivity and GPS signal reception, screen rotation, and termination by the OS).

Submit:
– List of use cases and NON-functional requirements met
– Screenshots of the App
Showcase: The working app on a device demonstrated to work under failures (network connectivity, GPS, screen rotation) or app can work-offline.
Evaluation Criteria: Number and quality of working use cases and non-functional requirements met.

Submission

Due: Friday, 7 November 2021

App Session Demo

Time: Each group will be given a time slot to present their work virtually.

Report Expectations

– **References**: You are responsible for citing any third-party intellectual property you use in the project.

– **Report Expectations** – References: You are responsible for citing any third-party intellectual property you use in the project. If you build an Android app, you should include an Android reference such as: [1] Android Open Source Project, Android, http://developers.android.com with an inline citation such as [1]. If your group uses Firebase for cloud storage, you should include a Firebase reference: [2] Firebase, https://firebase.google.com/.

– Repeat this process for each third-party work you use, including external (code) libraries.

– Reports without references will receive lower grades than those with references.
– **Clear Writing**: Your writing should be simple, concise, and easy to understand; if not, revise it! Your writing should demonstrate a logical sequence of thought; spelling and grammar errors should be absent. Most word processing tools (including online ones such as Google Docs) have built-in spelling and grammar checkers; please use them and note that Prospective employers and graduate schools expect clear writing. In addition, Latex writing is preferred.

– **Figures**: You are welcome to include figures in your report. Please cite them as "Figure 1. Hand-drawn figures are not allowed in the document, please draw them with software such as MS Visio, eDraw, OmniGraffle, etc.

Submission Requirements

– All teams need to create a GitHub repo to safeguard their work.
– Submit project presentation and documentation about project.
– The project presentation should have the following: Introduction, problem statement, Project Objective, Solution Design, Implementation details, system architectures.