SIT305 Assessment 2 Mobile App Project

TR1 2020

DEAKIN UNIVERSITY

SIT305 - Assessment 2: Mobile App Project

Worth: 50% of your grade

Introduction

This assessment requires you to work individually or collaboratively in a team involving maximum 2 students to produce a single, unique, real-world mobile app. Each team will select a single platform for their app: Android or iOS. You will be required to develop a complete, ready-to-publish, creative, advanced mobile app, solving problems as you encounter them. The project must involve a minimum of 3 advanced programming concepts that you haven't yet learned and will research. This assessment starts in Week 6 and ends in Week 12.

Team Size

Maximum two students

* If you are working in a team, please make sure that your contribution to the deliverable needs to be **clear and stand out**, and overall the teamwork should be equivalent to two projects developed by individual students.
* If you are working in a team, you must schedule weekly online meetings between yourselves, assign tasks, and contribute an equivalent amount of program code to the app solution. You are required to use a version control system (GitHub or BitBucket) to contribute code independently on a minimum weekly basis. Especially important will be to pull changes before working on files, and commit + push them after each work period. Otherwise you will end up with version conflicts.

Topic

You can choose any type of mobile app you want, so long as it meets all the marking criteria for the mobile app project.
A brief list of ideas:

- data storage apps (e.g. finance / tax apps, travel planner etc)
- games
- a blockchain-based mobile application
- an app for IoT application
- educational apps (e.g. learning math tutorials + randomised & marked exercises, spelling tutorials, etc)

Tasks

This assessment includes two main tasks: **project proposal** and **mobile app project**.

Task 1. Project Proposal

This Project Proposal is worth 10% of your grade, so the requirements are quite substantial.

- **Overview:** A brief, well-explained overview of the app you will develop.
- **Product Purpose:** It should include:
  a. ***Target audience**:* identifying the intended users of your product and why it would appeal to them.
  b. ***Reasoning of how your project demonstrates creativity*** in its design and will provide something new or entertaining for the viewer / user.
- **Features:** A list of features that your app include. For an app of this level, assume each customer will pay $10 - $15 for your app.
- **Design:** Wireframe / simple mockup of the visual screens your app will have, UI colour choices, and any UX decision (user experience, e.g. the flow of common actions). You can sketch this on paper and take a scan / clear photo, or draw using web-based tools and take screenshots, or any graphical program at your disposal. You want to be able to visualise:
  a. How many screens you will have,
  b. What fields are on each screen,
  c. How you will navigate between screens / scenes (so the user does not get stuck on one screen, or is unable to find one screen),
  d. The overall colour scheme for your app (generally select two colours, a background and a foreground text, using an online "Colour Theory Calculator"). You can have more colours as needed, if you are going to have header/footer bars that are not in native colour schemes. You should look back at competitors in Google play or Apple Store to see what kinds of styling may be appropriate for your kind of app market.
- **Data:** A top-level overview of the data tables/structures you will store in your app. This will include both:
  a. Runtime Variables (in-memory variables in RAM), and
  b. Permanent Storage Data (fixed data from disk storage / local database). This will usually be loaded into memory when your app loads, though you may access it on an on-need basis if the data is large.
- **API/Class Structure:** A top-level list of classes & significant functions/method signatures, that your app will need to be created. This is an overview of your own class structure that you will create. How many classes will you have? Where will you be writing all your code? How many methods/functions do you need, and what do each of them do? For example, if you had a Contacts app, you might have a UserStorage class, as one of your classes, with static functions/methods to load a user. e.g. "loadUser(String username)". You would also need a way to store each User, consisting of multiple values, so it would be a separate class to describe each of your user objects, and the fields contained within. You want to know ahead of time what program code you need to write.
- **Resources Required (optional):** A list of any external images / sounds you plan to source. Ensure you have license to use them for commercial purposes (CC-BY / CC0 / Public Domain licenses). Never use anything you did not create yourself unless you can see permission to use it clearly expressed.

Task 2. Mobile App Project

This mobile app is worth 40% of your grade. Using your Project Proposal as a guide, you should apply coding using mobile-app development framework to your components in order to demonstrate a range of coding concepts in a creative manner.

You are required to use a version control system (GitHub or Bitbucket) to contribute code independently on a minimum weekly basis. It is also important to pull changes before working on files, and commit + push them after each work period. Otherwise you will end up with version conflicts. You will be assessed on both the quality of your weekly contributions, as well as the overall app performance. You will also be required to keep updated a single changelog file, listing tasks completed each week per student. Weekly progress is required to pass (any file can be changed, as long as there is some change, making progress, at least weekly).

You will also be required to create and present a 2-minute video demonstration of their project, demonstrating the project and detailing the features of the app.

Platform

The platform of your project is your choice. It can be iOS or Android.

Project Directory

Your Project Folder/Directory must contain at least the following structure:

- **readme.txt** (details of yourself Name + SID (and any team member), and an overview of your project, Bitbucket / GitHub link, etc.).
- **changelog.txt** (a list of days you worked and what you achieved). This file MUST be updated weekly.
- **Source code**. Your code should be readable and documented enough for assessor to mark it. We should be able to compile & run the app.
  - data/ (any fixed data you may need in your app, e.g. json/csv/text files)
  - images/ (all your images are in here)
  - sounds/ (all your sound files go in here)

  If you create your own graphics, put them in a folder: "raw sources/"
- **licenses.txt/csv** (one line per image / sound file you use)
- **Demonstration.mp4** An approximately 2-minute long demonstration video of your app like a teaser, and the features you wish to be graded for. You will create this in the final few days before project submission.

Submission Details

The due date of the Assessment 2 is 5pm AEST, Friday, 5 June 2020 (Week 12) - on Cloud Deakin. You must submit the following files to the Assessment 2 link prior to the due date and time. Your submission must include:

1. **Project Proposal Document (Docx)**
2. **Project Directory as described above.**

Marking

Your submissions will be marked according to the following rubric (project proposal and app project), with a maximum possible score of 50 points. To achieve a particular grade, you must meet all criteria for that grade, as per the table below. Each higher grade requires all features of the previous grade as well.

CRITERIA	NOVICE	COMPETENT	PROFICIENT
PROJECT PROPOSAL (10 POINTS)			
Overview	May be absent or missing significant required content. **0 POINTS**	Summary provides general idea of the intended product. May lack some clarity. **0.5 POINTS**	Summary provides clear and concise idea of the intended product. **1 POINTS**
Product purpose	May be absent or missing information on target audience or creativity. **0 POINTS**	Provides basic identification of target audience and creativity. May be lacking some detail or a mismatch between product design and audience. **1.5 POINTS**	Identifies clear and detailed target audience and thorough justification of creativity. Purpose is highly intertwined with project design. **2 POINTS**
Features	A list of at least 2 features. **0.5 POINTS**	Each feature also has a sentence explaining the feature. **1 POINTS**	Each feature also includes details of how that feature will operate. You may also include screenshots /sketches if appropriate. **1.5 POINTS**
Design	At least 1 wireframe of the main interface. **0.5 POINTS**	Wireframes of all screens planned for your app. **1 POINTS**	You include a subsection identifying a bullet list of most common user actions, and the number of taps to achieve the result. (e.g. Button 1 -> Action 4 -> Scroll) **1.5 POINTS**
Data	May be absent or missing critical information. **0 POINTS**	You briefly explain the type of storage system(s) you'll use, and a bullet list of the major files/tables. **0.5 POINTS**	You include short example snippets of each data object / table (all text-based file formats). **1 POINTS**
API/Class Structure	You include a list of all classes that should be developed for your primary features, and what their purpose will be. **0.5 POINTS**	For each class, you outline data fields, as well as any substantial functions/methods as method signatures. **1 POINTS**	For each substantial function/method, you briefly explain where it will be used / called from, how it will work, followed by brief pseudo-code. You will also include examples of how to call the function/method. **1.5 POINTS**
Presentation	Some minor spelling, grammatical or formatting errors throughout. Some additional proof-reading is required. **0.5 POINTS**	Very few minor spellings, grammatical or formatting errors throughout. Presentation is professional and polished. **1 POINTS**	No notable spelling, grammatical or formatting errors. Presentation is thoroughly professional and polished. **1.5 POINTS**

DEAKIN UNIVERSITY

CODE (40 points)			
Weekly progress & code commits (bitbucket/GitHub)	At least 3 unique day commits (per person). You also need an equal number of fetch/pulls. Changelog updated at least once a week (per person). **1 POINTS**	At least 5 unique day commits (per person). Changelog lists all new major features added per week. **2 POINTS**	At least 10 unique day commits (per person). You include, at the end of each week, a summary of how you are progressing related to your milestones. **5 POINTS**
Documentation	Commenting is present in project blueprints although may be very simplistic or used inconsistently. May still have some issues for someone unfamiliar with the project to understand. **1 POINTS**	Commenting is present throughout project blueprints and provides acceptable readability of the code for someone unfamiliar with the project. **2 POINTS**	Commenting is present, thorough and sophisticated throughout without overwhelming the project. The code is very easily readable for a user unfamiliar with the project. Consistency is maintained in labels and comments. **3 POINTS**
Mobile App	Satisfies coding requirements in a basic manner that is functional within the project. May have minor issues or lack thorough integration with the rest of the system. May not go beyond what has already been demonstrated in class. **15 POINTS**	Satisfies coding requirements effectively with creative or interesting functionality. Used effectively as a part of the larger system with logical implementation. Evidence of implementation beyond what has been demonstrated in class. **20 POINTS**	Exceeds advanced coding requirements effectively and/or demonstrates highly creative or interesting functionality. Sophisticated integration demonstrates thorough understanding of the concept. Clear evidence of functionality beyond what has been demonstrated in class. **30 POINTS**
Demonstration Video	May be missing **0 POINTS**	You demonstrate part of features working in the app. **1 POINTS**	You demonstrate all main features working in the app that you wish to be graded for. **2 POINTS**