CS31520/CS31620 Assessed Assignment 2019-20
Assignment: Conference App

Chris Price / Chris Loftus

Release date: Tues, 15th October 2019

Submission date: Tues 3rd December 2019

Type of hand in: Blackboard Assignment

This is an individual assignment, worth 60% of the module assessment.

Problem

Conference attendees would like an online version of the programme for the conference they are attending. It should give a time-ordered list of conference sessions, so that they can see what is happening, and provide details of talks and speakers. The attendee should be able to make a list of the sessions that they really don't want to miss.

Details of each session

Details of all the sessions at the conference are stored in the provided SQLite database. The database has the following tables:

sessions:

id: TEXT	\\ Primary key for the sessions record
title: TEXT	\\ Title of the talk or workshop
content: TEXT	\\ Description of the talk or workshop
locationId: TEXT	\\ Reference to the relevant location in locations table
sessionDate: TEXT	\\ Date on which the talk takes place
sessionOrder: INTEGER	\\ Order of sessions within each day (0 is first, 1 is second...)
timeStart: TEXT	\\ Start time for session in format hh:mm
timeEnd: TEXT	\\ End time for session in format hh:mm
sessionType: TEXT	\\ Type of session (talk / workshop / coffee / lunch / dinner)
speakerId: TEXT	\\ Reference to the talk speaker in the speakers table

speakers:

id: TEXT	\\ Primary key for the speakers record
name: TEXT	\\ Name of the speaker
biography: TEXT	\\ Details about the speaker
twitter: TEXT	\\ Twitter handle for the speaker

locations:

id: TEXT	\\ Primary key for the locations record
name: TEXT	\\ Building name for the location, e.g. "Physics Main"
latitude: REAL	\\ The latitude component of the location
longitude: REAL	\\ The longitude component of the location
description: TEXT	\\ Description of the location

There are also a set of photos of the speakers (see pics.zip). The name of the photo for a speaker is made up of the speaker's id plus ".jpg".

List sessions functionality

The app should provide a list of all of the sessions at the conference (including breaks and meals), in the order in which they happen. The list should give the time of the sessions, the title of the session, the speaker's name (if any). If the session has further details (workshop and talk sessions have further details, others do not, see next section), then the user should be able to select a session and get further details of that session.

Session details functionality

Where the user chooses to get further details of a talk or workshop, the app should show details of that talk or workshop, including title, time, location, content, speaker details and photo. The user should be able to add / delete the selected talk or workshop to / from their list of favourite talks. Only talks and workshops can be favourited.

List favourites functionality

The user should be able to get a list of all of the conference sessions that they have favourited and look at the session details by picking an item from that list.

User Interface design of the app

App UI design is very important. You need to think through what might be the best design for your app, and prototype it before you build it. You must hand in evidence that you have done this kind of prototyping. This kind of prototyping was discussed in lectures.

Construction of your app

Make sure you have a software design using appropriate UML diagrams.

Build the app based on the initial UI and software designs. Clearly, you will iterate between implementation and design as you proceed. Module workshops and Android workbooks and Apple's App Development with Swift book covered all of the aspects of app construction that you will need for the implementation - overall app formatters such as tabbed screens, or hierarchical screens, lists of content, interaction with buttons, labels and text boxes, storing and accessing data within an app. You should aim to build a working app to match your design.

Testing of the app

You should aim for automated unit testing and user interface testing of your app. How this can be set up will be covered in lectures, workshops and associated workbooks, but you will need to decide what is to be tested, as well as how it is tested. Although not as good, if you cannot achieve automated testing then make sure that you have a manual test table.

Submission of the assignment

By the deadline, you should submit the following via Blackboard:

- A copy of your app (hopefully completely functional, but you should hand it in even if it is not functional as not all the marks are for functionality).
- A non-functional prototype showing your UI design for your app. This might be paper-based (electronically scanned), a set of linked screens in a presentation tool such as Powerpoint, or a set of screens in Xcode, FluidUI or Adobe XD.
- A report on the project containing:

- A description of how you decided on your design for the app: The UI and UML software design.
- A plan for testing the app, and details on how testing went.
- A reflection on how the assignment went - what went well, what might have gone better and why, what you have learned, and a summary of how well you think you did the project and what mark you think it deserves.

Submission details

In submitting your work, you are confirming that you comply with the Department and University statement and policy on Unacceptable Academic Practice. When you submit your report, you will be confirming that you understand the policies and confirm that your work meets those policies.

If you cannot access Blackboard, you need to email your report to either Chris Price for CS31520 (cjp@aber.ac.uk) or Chris Loftus for CS31620 (cwl@aber.ac.uk) by the time of the submission on Blackboard. Your email should also explain the problem that has prevented you from uploading to Blackboard. If you send the email after the deadline for submission, it will be treated as a late entry. Please see the Student Handbook [1] for details on how this would be processed.

Breakdown of marks for the project

Marks for the project will be assessed on the following basis:

Criterion	**Value**
Quality of the design (UI and software)	20%
Technical achievement (whether it meets the requirements)	20%
Flair (how much your app is a delight to use and going beyond what we ask for)	10%
Quality of product (usability, robustness)	20%
Quality of automated testing (plan and execution)	10%
Quality and coverage of the report	20%

References

[1] Computer Science Department (2015) “Student Handbook” (Online) https://impacs-inter.dcs.aber.ac.uk/en/cs-undergraduate/official-information/student-handbooks (Accessed 9th October 2019)