CMPUT 301 Fa18 - INTRO TO SOFTWARE ENGINEERING Combined LAB LEC Fa18

Assignment 1

Assignment 1

Learning Objectives

- Solve a problem by constructing a simple, interactive application using Android and Java.
- Document an object-oriented design in Unified Modeling Language.

Problem Description

I need to empirically understand my feelings. So I need to record my current feelings. Make a simple, attractive, intuitive, mobile app to record feeings quickly. Let us call this app: FeelsBook.
The feelings that should be supported are:

love, joy, surprise, anger, sadness, fear

derived from here: http://changingminds.org/explanations/emotions/basic%20emotions.htm from

Shaver, P., Schwartz, J., Kirson, D., & O'Connor, C. (2001). Emotional Knowledge: Further Exploration of a Prototype Approach. In G. Parrott (Eds.), Emotions in Social Psychology: Essential Readings (pp. 26-56). Philadelphia, PA: Psychology Press.

Each record of a feeling will include:

- Feeling or Emotion (categorical)
- Timestamp (presented to the users in iso8601 Date&Time format e.g. 2018-09-01T18:30:00)
- Comment, Optional text (up to 100 characters)

Only the coment field might be left blank for a subscription.

The app should allow the user to:

- Quickly record an emotion with only 1 click on the app and add an optional comment.

- See a count of each emotion recorded
- Browse and view the history of emotions.
- view and edit the details past emotions.
- delete a past emotion.

The history list need not show all the information for a subscription if space is limited. Minimally, each record in the list should show at least the date and the emotion.

The history list must be ordered by time.

The app must assist the user in proper data entry. For example, use appropriate user interface controls to enforce particular data types and avoid illegal values.

The app must be persistent. That is, exiting and fully stopping the app should not lose data.

Use your campus computing ID in the app name. Specifically, the app name must show up as *YOURCCID*`-FeelsBook` (e.g., `kennyw-FeelsBook`).

Deliverables

1. **Code Base:** (5 marks)
   Your complete source code and compiled binary, implementing the working app and its user interface, will be inspected and run by the TA. The Android Studio project and APK (Android Package Kit) binary must reside in a git repo. Each class must contain comments describing its purpose, design rationale, and any outstanding issues.

2. **Video:** (1 mark)
   The video is a demonstration of the app. The video file must reside in a service like YouTube or archive.org. Do not put the video file in your git repo. The video is meant to show that the following actions actually work. No audio is needed. Maximum duration is 3 minutes. You may use the screen recording software in the labs (e.g., `ffmpeg` or `simplescreenrecorder`). Focus on just the screen of the app. For visual clarity, do not use a handheld video camera.

   1. Open the app from the launcher.
   2. Show the list of possible emotions.
   3. Add 1 joy emotion (no comment)
   4. Add 2 love emotions each with the comment "pizza"
   5. Add 1 fear (no comment)
   6. Add 1 fear with the comment "no pizza"
   7. Edit the last emotion, fear no pizza and change the date to 1 hour earlier.
   8. Go to the history and browse the emotions. Show that the "no pizza" emotion has changed.
   9. Exit and stop the app, showing in the running apps list that the app is no longer running.
   10. Open the app again.
   11. Show the counts of each emotion
   12. Show the history list
   13. Delete the fear emotions.
   14. Show the emotion counts again

3. Describe the structure of your app's object-oriented design using UML class diagram(s), saved as non-lossy image file(s). Focus on the most important classes that you designed and implemented. Add notes to describe the main responsibilities of these classes.

Hints

This is a description of the core functionality. Often, problem statements from users lack details. As you are prototyping a design, you may uncover other behaviors that have not been specified, but make sense in the context and intent of the application. For example, think about how someone might effectively use your application. It is up to you to decide what functions your design will need, based on the given problem description and valid assumptions, in discussion with your users (the TAs and instructor). You should consider asking the customer (the instructor) what they want to see.

While you may discuss your design with other students, the code and documentation must be your own work. Code from publicly available sources may be used within reason and if their licenses permit so. In particular, at least 50% of the application code should be yours. Always fully cite to give proper credit to the original developers in the source code and in the system documentation. For example, in citing a work, at least state: from whom, the date of publication, license, and URL. Include what is required by its license.

The TAs will be inspecting your code, so "major" commented-out experiments should be cleaned up so that the code is readable.

Do not skimp on the UML class diagrams in the system documentation. For neatness and readability, diagrams should be created or drawn using a vector graphics editing tool and exported in a common, non-lossy graphics format.

Besides addressing the problem correctly, your software design will be evaluated on its proper use of object-oriented design concepts, such as separation of concerns and information hiding.

You should practice using GitHub frequently to contain changes throughout development, not only as a final submission mechanism.

Consult the assignment rubric.

Losing Marks

You may lose marks for any of the following:

- missing APK file for the app
- cannot run the app after install
- cannot distinguish CCID from the app name
- video file in the git repo
- image files for UML not in a `doc/` subdirectory
- cannot view files for UML without a specialized tool
- lossy compression used in image file for UML (e.g., JPEG)
- use of SQLite for storing data
- Inappropriate URL in eClass submission

These are brown M&M rules.

Submission Procedure

The app name must show up as *YOURCCID*-FeelsBook (e.g.,kennyw-FeelsBook), so that it can be easily distinguished from other submissions.

On eClass, on two lines, post:

- a URL to your GitHub git repository for your assignment, and
- a URL to your video.

The GitHub URL should be formatted like this:

```
http://github.com/YOURGITHUBUSERNAME/YOURCCID-FeelsBook.git
```

This should be the git URL such that the TA can run:
`git clone http://github.com/YOURGITHUBUSERNAME/YOURCCID-FeelsBook.git`

and get a copy of your assignment code base including a compiled APK file, as well as the system documentation. This should be structured like your Android Studio project.

In particular, the APK file(s) should be within an `app/build/outputs/apk/` subdirectory, and the image file(s) for the UML class diagram(s) must be within a `doc/` subdirectory. Make sure these are explicitly present in your git repository.

Please make sure all the required files are included to build the app. The TA will test your app from the submitted code and APK file.

Submission status

Submission status	No attempt
Grading status	Not graded
Due date	Friday, 5 October 2018, 4:00 PM
Time remaining	11 days 6 hours
Last modified	-