
CSC 2002S: MOBILE DESIGN & DEVELOPMENT

You have two assignments for this module. For A1 (Phase 1) you will demonstrate that you can create a basic Android application. For A2 (Phase 2) you will use what you are learning about design to prototype an improved application. To help you plan and prepare, both assignments are being released and described in this document.

A1 (PHASE 1) KILOJOULE COUNTER

Marks - 30 points

For this assignment you will build a basic calculator and kilojoule counter to help track and calculate your nett kilojoule intake (i.e. NKI()).

Overview

The application should open with an overview screen which contains the following elements:

- A calculator launcher (to add a diary entry)
- A list of diary entries, each showing the date and nett kilojoule intake (NKI) on that day
- The average daily NKI(), averaged over all daily entries

Calculator

Tapping on the calculator launcher (in the overview screen) will open a calculator, in which the user can enter kilojoule estimates in at least 4 categories of your own choice – but you must have at least 2 Food categories and at least 2 Exercise categories (where Food values are added to that day's NKI total, and Exercise values are deducted from the day's NKI total). For example, the Food categories might be Breakfast, Lunch, Dinner, Snacks; the exercise categories might be Gym, Sport, Jogging. The calculator should show 3 running totals and include a button for saving the total as the diary entry for a particular date. The 3 running totals would be Food total, Exercise total, Nett total (where Nett total = NKI() = Food total minus Exercise total). Saving an entry will take you to the entry's page in the diary. There should also be a button that will allow the user to exit the calculator without saving, which will return to the previous activity.

Diary

Tapping on an item in the list of diary entries will open that entry's page. Each page should display the following:

- Entered totals for the categories
- NKI for the day i.e. (sum of all Food values that day) – (sum of all Exercise values that day)
- Button for navigating to the next entry (disabled if last entry)
- Button for navigating to the previous entry (disabled if first entry)
- Button for return to Overview
- Button for calculator

Provide support for strings in at least two languages of your choice, using the phone's locale to determine what language you will use. (I'm okay if you pretend French is actually pirate language.)

When you open the application, it should always open to the last screen viewed, including any previously entered form data. Screen rotations should not delete partially entered data or take you to a different screen. If you choose to use icons for the buttons or the overview page, you may place those in the resources.

Any native code should be written in Java, not Kotlin. Please submit your source code, and a compiled APK that will run on a Google Nexus 5X emulator running Nougat (API Level 24). All files should be put into a single archive, which should

be named according to your student number and the assignment number, e.g. ABCXYZ001_CSC2002S_A1. Upload the file and then ensure that you have uploaded the correct file.

You may develop your app in whatever environment you prefer (Android Studio is strongly recommended). However, you should test that it runs with the emulator that comes with Android Studio. **You must also sign up on Vula to do a five-minute demo and code walkthrough with one of the tutors during one of the designated slots.** *Failure to show up for your demo and to successfully demonstrate your app by this date will result in zero marks for demo portion of the assignment.* If you will not be doing the demo on your own android device or on an emulator on your laptop, please be sure to arrive for your demo early so that you can set it up and ensure that it is running on an emulator on one of the available machines in time for your appointment.

I highly recommend that you start working on this right away. The biggest barrier in getting this assignment done is often getting your development environment configured and running, rather than the code itself. Please reference the Android Developer tutorials and other resources on <https://developer.android.com/training/index.html>.

GRADING

CODE: 30 MARKS

This phase will be marked on how well you follow Android conventions and make use of the facilities provided in the Android SDK. You will not gain extra marks for advanced functionality. This phase will be assessed on the submitted apk and code alone.

Category	Description	Points
Demo 10 points	Does the APK run and launch the Overview?	1
	Overview: All required elements present? All button/list items respond correctly?	2
	Diary Entry: Are all required elements present? Can we navigate Prev/Next ?	2
	Calculator: Are all required elements present? Can we create a new entry?	2
	Back: Does the app handle the back button appropriately?	1
	Lifecycle Management: Screen rotations or application exits should restore state	1
	Internationalization: Is there support for more than one language?	1
Code 5 points <i>based on code review</i>	Overview Page: Is there code/xml for the overview?	1
	Overview Button: Is there code/xml for a button?	1
	Diary Previous/Next: Is there code/xml for the buttons?	1
	Calculator Page: Is there code/xml for the calculator?	1
	Calculator Buttons: Is there code/xml for the buttons?	1
Following Conventions 15 points	Layout: Is the layout specified in an xml file located in the res/layout directory?	2
	Icons: Icons contained in resources.	1
	Locale Support: Strings should be specified in res/values/strings.xml and not hardcoded in the layout or the java files	1
	Activity/Action Bar: Does the layout use the Activity Bar for the main actions?	1
	Lists: Does the layout follow Material Design conventions for diary entry display?	1
	MVP: Are there classes that clearly correspond to the model and the presenter? Model = Representation of and access to the images View = How the model is shown to the user, the interface Presenter = Manages user input and model updates	3
	Activity: appropriate methods should be implemented to retain state (i.e. current entry) when app is destroyed, paused, or stopped	2
	Storage: Reasonable implementation for storage of diary entries	2
	Multi-threading: Diary entries should be loaded and saved in the background and not block the main thread	2

A2 (PHASE 2) IMPROVED KILOJOULE COUNTER

Marks – 50 points

You need to design and prototype a more advanced NKI diary. You should start by listing three or more new features or design changes for the system, explaining why you think those features are useful on a mobile device. Do not just give personal opinion - think about the social issues discussed in lectures, and design patterns you may have seen in similar apps. Possible features might include variations on the calculator, inclusion of weight loss tips and/or exercise tips, indication of your progress towards some goal, etc.

Next, you need to design how the interface for this functionality will appear. To that end you need to create a prototype and annotate it with a rationale for your design; be prepared to explain why your design looks the way it does with reference to the design ideas in the lectures. Simply stating that you think your design looks nice does not constitute an acceptable rationale. The more (relevant) information you can give about why the interface looks the way it does, the more marks you will receive. Where relevant, reference specific design principles and sources using academic conventions for referencing¹.

You do not need to implement this as an android application – you may use any prototyping tool, such as Invision, POP, Fluid UI, or PowerPoint. Hand-drawn images may also be acceptable, but you should justify and discuss the fidelity of the prototype.

For this assignment, you will submit a written report of up to (*at most*) 3500 words, containing:

- Overview of the app, describing your target audience and any assumptions you have made. You will want to show how the three features you choose fit together in the overall app.
- A list of features with an explanation for their inclusion
- An initial design for each feature. This will consist of a number of screenshots, with clear markings of how one transitions from one screen to the next. It may be easiest to have a diagram with numbered elements and refer to those numbers in the text. Be sure to show how errors or edge cases would be handled in your design. *Diagrams may be hand-drawn if desired but must be clear to the reader.*
- For each feature, justify the design; explain why it looks and works the way it does. Do not just state the name of relevant design principles, but make an argument for why others will appreciate your design.

GRADING

FEATURE RATIONALE AND DESIGN: 45 MARKS

Each feature is worth 15 marks. Up to 3 features will be marked. If you add more than three features, you should clearly indicate which three you want to be marked, or they will be selected at random.

- **Feature Rationale (5 marks):** Fully justify the three features that you decided to include. Include details of any research or experience that lead you to choosing them with referencing.
- **Design (5 marks):** Document your screenshots, showing how to trigger transitions, and clearly showing the interface for your feature.
- **Design Rationale (5 marks):** Explain your design fully, linking it to the concepts and ideas that you were taught in class. Include in this the ways in which it conforms to the different standards and models, if appropriate. Every design decision should be motivated.

¹ Do not reference lecture slides.

	Not Done (0)	Fair (2)	Good (3-4)	Excellent (5)
Feature Rationale	No feature specified.	Feature inadequately described and/or not sufficiently justified.	Feature justified.	Clearly articulated and cited rationale for feature.
Design Prototype	No design specified, or design unclear.	Prototype provided (of any fidelity) but transitions and where it fits in the existing app are unclear	Clearly depicted prototype including transitions, error handling and synthesis with current app	Innovative prototype, clearly articulated, leverages affordances and mappings in an intuitive way.
Design Rationale	Design not justified.	Justification is given but doesn't reference design principles or other concepts taught in class	Justification includes either an explanation of how it adheres to standards and models or it is linked to HCI and design concepts	Justification correctly references both appropriate standards and HCI design concepts.

OVERALL DESIGN AND LAYOUT: 5 MARKS

Provide an overall design for your finished app, showing how all of your features will interact. Make sure this corresponds to your explanations with the rest of the assignment. You should discuss your target users and state any assumptions. Specifying the device(s) that the app would run on may be helpful too.

Score	Description
Missing (0)	No layout section or screenshots provided
Minimal (2)	Screenshots incomplete, or screenshots provided but it is not clear how the different features and screens will interact, and screenshots do not adhere to Android design principles.
Fair (3)	Layout is specified but the features or designs do not mesh well together, or are confusing to the projected user, and do not adhere to Android design principles.
Good (4)	Interactions between features are clearly specified, with markings for how various features are invoked, and how different user interactions lead to changes in screen layout. Screenshots adhere to Android design principles. Clear synthesis of the app overall (i.e. consistent fonts, buttons, etc).
Exemplary (5)	Good (as above) and also strong discussion and targeting of the designs to a target audience, complete application specification, accounting also for application and user errors.