CSCI 4176 and 5708

Mobile Computing

DALHOUSIE UNIVERSITY
FACULTY OF COMPUTER SCIENCE

Project Design

Overview

Now that you have an idea for an application, it is time to add detail to the idea and begin designing what you are going to implement. Your design doesn't need to be "heavy-weight" and can be light and adaptable, but successful development always has a design. Agile development with minimal planning is most effective when developers have experience in the domain area and can mentally re-use past designs, or components of them. When developers are less experienced, design and planning leads to a stronger product.

For the design, we will assume that you are going to use an MVC architectural pattern which you can later refine to suit MVVM or any other pattern. MVC breaks your application down into 3 key components:

1. A data **model** which you will describe using techniques such as ER diagrams. You could also use class diagrams of data classes, JSON, or any other technique to describe **all** the data for your system.
2. A UI/UX **view** of the application which you will describe with sitemaps, storyboards, wireframes, or low-fidelity prototypes.
3. A **controller** to link the model and the view, which you could describe using UML structure diagrams (e.g., class, component, object, package). Other techniques are permissible, provided you demonstrate that you've put some thought into how you will implement the Kotlin component of the system.

As you create your design, you can refine, improve, and update any functionality that you desire. You are not bound to anything you proposed and have free reign to change the details and improve the overall product. You may not "start over" with a new idea as it would take too much time and put you too far behind in the course, but you can improve and modify your initial idea.

The focus at this stage is to design a useful, usable, and desirable application. You are not required to integrate specific mobile device features if you cannot do so in an effective way that isn't just adding "bloat" to the application. I want you to create high-quality applications, not unusable Franken-apps filled with unnecessary features. However, please be aware that overly simplistic applications with no networking, use of device features, or persistent storage are subject to low grades as a consequence of being trivial and low effort to implement. That is, I'm not going to award very high marks for a calculator or stopwatch, for example.

Deliverable

The project design document describes what you are going to implement. It builds upon the proposal and adds the detail needed for a developer to move forward. A good design is one that you could hand to another student in the course, and they could implement without a lot of questions. The submitted document should be more than just a couple of pages and should not simply be a collection of unexplained diagrams. Documents of any length are permissible if they have the desired content and level of detail. It is suggested that your document cover the following topics:

Title Page and Abstract: This includes the title of your project, your contact information, and the abstract -- a brief 100-word summary of the document. Please indicate your program of study beside your name (MACS, BCS, BACS, etc.).

Introduction: You should set the stage for your design by explaining your idea. As mentioned, you can modify the idea based on my feedback, anything you've learned, and anything else that you feel improves the end product. Do not assume that I remember what is in your proposal.

Users: Identify your target audience. I know this is constantly changing and being refined as the term progresses, so this is a chance to add detail, improve, or modify the original ideas from your proposal. **Be specific; details matter!**

Data Model: Describe the data that your system is going to use. You could use an ER diagram with any format that is familiar or comfortable. Please use software to create this diagram (e.g., MS Visio, Visual-Paradigm, Lucidchart, draw.io). Google "ER diagram tool free" to find alternatives. As mentioned, you might find JSON or UML class diagrams of more use for your specific data set. The level of detail is up to you, but more is usually better. However, some applications have a simple data model that doesn't take much to describe, so you may not need much detail. Other applications are data-driven and have complex and detailed data models. Provide as much detail as is needed for the complexity of your application. I also want to see some evidence that you have considered how and where your data will be stored. What database will you be using and why are you using it?

View: Explain the user-interface and user-experience (UI and UX) using some combination of the following:

1. Sitemaps,
2. Storyboards,
3. Wireframes,
4. Low-fidelity prototypes.

I would suggest that you use an augmented sitemap to create a storyboard thus integrating items 1 and 2. Whether you use item 3 or 4 is dependent upon your preference and design skills. User-stories/use-cases are excellent tools to ensure that your UX is complete and not missing any critical functionality. As well, using CRUD to ensure that your views are not missing any needed elements is recommended. You may hand-draw your wireframes or prototypes, but I wish you to use a drawing tool to create the sitemaps and clickstreams (Adobe XD and Figma are free)

Controller: The view is linked to the model using the controller by way of Kotlin code. Thus, you can describe your controller using a UML structure diagram (e.g., a class, package, deployment, or object diagram). The combination of diagrams that you use is up to you. Again, I ask that you use some sort of drawing tool to create your diagrams (google "free UML drawing tool" for support). Ensure that your diagrams identify the APIs that you want to use. You should be doing research on available APIs and identifying those may be helpful or beneficial. For example, you may wish to start with a source such as:

`https://rapidapi.com/blog/most-popular-api/`

Other: You may also want to include a risk assessment, focusing on development risks as opposed to user-facing risks. It is also appropriate to start your implementation planning using timelines or Gantt charts, establishing your milestones, and developing some test cases (particularly if you are using test-driven development). You may wish to include some initial views created using Android Studio that you can use as high-fidelity prototypes. You might want to identify other resources, as useful libraries that you can use to reduce tedious programming tasks, particularly since Kotlin can be linked to Java code, thus making Java libraries available. Please do not feel limited by this list of ideas and include anything that you think is relevant to creating a strong design.

Learning Outcomes

This assignment supports Course Objectives a. to f. from the course syllabus. In addition, upon completion of this assignment, students should be able to:

1. Integrate a user-centric approach (e.g., target audience focused) with application and feature design.
2. Express an MVC-based design accurately and concisely in a written document.
3. Apply data modeling skills in a mobile application context.

To obtain the highest possible grade, <u>you need to clearly demonstrate that you have thought deeply, critically, and creatively about your design</u>. The more evidence I see of critical or creative thinking, the more likely you are to achieve a high mark. For example, using the structure and titles presented here (e.g., Introduction, Users, …) is not evidence of critical or creative thought. Treating the "Deliverable" as a template to be filled in, while satisfactory, doesn't indicate that one has developed any problem-solving skills.

Document Requirements

I expect your submissions to adhere to the following requirements:

i. Use of full justification.
ii. Use of a font with serifs because I find them easier to read. Examples are Century Schoolbook, Times Roman, Garamond, or Georgia Pro.
iii. Single spacing so that I can read without having to constantly scroll.
iv. The preferred font size is 11pt. However, 10 or 12 point is also acceptable.
v. A satisfactory reference list with no secondary references and associated citations in IEEE or some other well-known format.
vi. No use of point form. Minimal (if any) use of bulleted or numbered lists.
vii. Use of Canadian/International/British spelling, not US.
viii. Sufficiently correct grammar that I can understand what you are trying to say.
ix. Please use page numbering as it makes it easier for me to identify a specific page when providing feedback.
x. Submit a PDF document, not an MS Word .docx version. Do not use .zip or other compression/archival tools and formats.

Evaluation Criteria

This design will be evaluated with the rubric given in Appendix A.

This assignment will have a mark out of 40. A zero will be given for any missing part. Marks may be deducted from the final score if there is some obvious mistake or issue that isn't covered by this rubric (e.g., the content lacks originality, but is properly cited and referenced).

A small bonus may be given for work that demonstrates ideas or presentation that is unique, creative, different, or interesting. No score can go above 40 (perfect) even with any extra bonus marks. You will not be penalized in any way for trying something new, novel, different, or risky – so long as it is described well. I value effort, imagination, and courage.

Expectations are different for graduate students. They have completed their undergraduate degrees and thus are expected to perform at a level above that of the undergraduate students. Graduate students are in the upper percentiles of students with an undergraduate degree and my expectations are based on this fact. However, I recognise that the groups contain both graduate and undergraduate students and I adjust my expectations based on the numbers of each in the group.

When marking, I ask myself:

- Did you put some thought into the design? Is there evidence of critical or creative thinking?
- Is there sufficient detail and accuracy to implement the design?
- Are your choices and decisions researched and justified? Is research indicated using satisfactory references?
- Do you use an excessive amount of material from the web? Is there evidence of originality?
- How much irrelevant material is present to make it seem longer?
- Do you have all the required parts? Is this a complete report?
- Does it look like you gave this assignment much effort? There is a difference between concise (short but strong content) and low effort (short and vague). Does it feel like it was written at the last minute or rushed?
- Did you structure the document appropriately? Have you organised the material in a manner that is suitable for your content?

Component	Unacceptable: 1-2	Minimal: 3-4	Good: 5-6	Really Good: 7-8	Exceptional: 9-10
Model	If present, the model lacks detail and depth. Diagrams are absent.	You diagram the model in a rudimentary way that lacks much insight or quality. Or, you describe the model in vague, unclear, terms, perhaps in text. I couldn't implement this model. Excessively short and lacking content.	You diagram the model in a satisfactory way, but lack detail, depth, accuracy, or completeness in some places. There is a bit of supporting text. I could implement this model if given considerable additional information. May be excessively long and filled with irrelevant content (i.e., not concise).	You diagram the model clearly and accurately, but detail or completeness isn't quite perfect. Diagrams are explained and motivated. There is text to link the component to other components. There is evidence that you did research to support your work. I could implement this model but would need to ask some important questions. Not lacking much content, but also, not bloated with irrelevant content.	You diagram the model accurately, completely, in high detail, and in accordance with best practice (e.g., using BCNF, 4NF for ER models). You provide supporting text to explain and justify your choices and to integrate the model with the other components. Supporting research is strong and well done. I could implement this model without needing to ask a lot of questions. You consider various stakeholder viewpoints.
View	Scored using identical criteria for model, but focusing instead on UI and UX (e.g., storyboards, wireframes, prototypes, use-cases).				
Controller	Scored identical to criteria for model, but focusing on APIs, classes, fragments, coroutines/threads, modularisation, and code organisation.				
Presentation, including **1. Appearance** **2. Organisation** **3. References** **4. Conciseness** **5. Overall quality**	Unreadable and unclear.	Point-form, bullet-lists, sentence fragments, etc. No references. Looks childish with inconsistent spacing, big fonts, lots of wasted space, etc.	Minimal or weak references. Not very attractive but meets expectations. Violates some of the requested formatting guidelines.	Sufficient references but weak formatting of references. Looks OK, but not professional (poor fonts, low quality clip art, bad kerning/spacing) etc. May violate the formatting guidelines.	The document has strong grammar and spelling. It looks professional and attractive. References are thorough and not just websites. Content is well organised and includes additional, relevant, material. Not a lot of extra "filler" or irrelevant content; it is concise. Almost no room for improvement at all.