



# Project Specifications

## Project Overview

The course project is your opportunity to develop skills and expertise in mobile application development. It is designed to take you through all stages of mobile development, from coming up with a new idea for an application to designing, implementing, and testing the application. The final deliverable will (hopefully) be a marketable and operational Android application. Please consider the concepts of viability, feasibility, and desirability when designing your application.

Your application can be in an area and target any user-group(s) of your choosing. If possible, it should use available mobile device features, such as a camera, GPS, accelerometers, and so on, in an interesting and useful manner. The application should also use Bluetooth, WiFi (i.e., IEEE 802.11) or cellular networking in some capacity. The point is not to create a “Franken-App”<sup>1</sup> but to combine features to provide functionality that is useful, meaningful, and seamless.

The application should also provide interesting and useful functionality. It should not be a technology demonstration or a “toy” application. For example, you should not create your own version of SkipTheDishes, Uber, or Amazon with a tiny pre-defined database to “demonstrate” how it would work once you have actual customers. The application will likely have multiple views (screens) and is expected to perform a useful task. That is, if no one would ever want to buy your application, even if it were free, then you may need to rethink your application and requirements.

You should be able to divide the functionality of your application into three distinct levels:

1. **Minimum Functionality:** The minimum amount of functionality that you expect to implement, assuming all your carefully laid plans are disrupted.
2. **Expected Functionality:** The functionality you expect to implement, assuming most things go according to plan.
3. **Bonus Functionality:** The functionality you would like to implement, assuming you managed to get everything done a lot earlier than expected. This is your opportunity to be really creative and showcase advanced mobile computing concepts.

Your application should also have a high level of usability. While you will not have a lot of time to test and refine your application, it should look as professional and high-quality as possible.

---

<sup>1</sup>A FrankenApp, is like Frankenstein’s monster – a collection of parts/features that is poorly assembled and for which many of the parts/features are inappropriate and ill-fitting.

Some important elements of usability are:

1. How enjoyable and easy your application is to use,
2. How well your application meets user expectations,
3. How attractive your application looks, and
4. How effective your application functions.

Usability and conformance with usability heuristics is not an optional element that you add after you have created a working application; it is required and necessary component of the entire development process. Unusable applications are equivalent to trash. Of course, there is a trade-off to be made when deciding whether to improve usability or coding new features as you will not have time to do everything. It is also important that you consider the usefulness and marketability of your application. That is, how well does your application distinguish itself from other applications in the same domain/market/category?

### The “No Credit for Backend Service” Clause

Marks are awarded solely for the portions of the application (i.e., frontend) that execute on a mobile device. If the implementation of your application includes a remote server (e.g., a backend service such as the Amazon cloud), you should be aware that **no marks are awarded for the server portion of the implementation**. That is, you should avoid implementing a backend service. You are encouraged to use API services instead of trying to implement your own. However, use of a remote database is encouraged and your data model, access protocols, and integration of this database into your architecture model will be evaluated. While Firebase will not be an ideal solution for many of your projects, you should consider using it as the supports (e.g., examples, tutorials) are very strong.

### Tool Use

For the project, you will need to use Android Studio and produce a working executable that uses the same emulator and version as you are currently using in the lab. You may use any other tools that you desire (e.g., libraries, APIs). Please be aware that I will not award points for things you do not do yourself but you will be rewarded for effective use of available tools and resources.

You will be using git (cs hosted gitlab) during the coding part of the project. I expect students to have lots of commits, use branches and commit code even if the code is to "perfect" or compiling. Your git should tell the story of the development process. All students are expected to commit to the group repository.

### Production and Output

More is not necessarily better. I have seen 400 LOC outperform 10 KLOC on the same task. A larger quantity of code will not necessarily have higher marks than submissions with less code, as the quality, complexity, performance, and effectiveness of the code is of considerable importance.

Failures, prototypes, and similar activities are not always evident or easy to measure. I recognize this and will not penalize anyone who can clearly demonstrate (in their final report) that they did research, prototyping, problem-solving, and other activities that do not directly lead to the immediate production of code. Writing the “right” 500 LOC is more

important than writing more code of lesser quality. Please document your efforts and failures as well as your achievements and successes.

*Fail early, fail often!*

### Quality

Quality is a concept that means “you did something well.” In the software community there are many definitions of quality.<sup>2</sup> Quality cannot be measured by an all-encompassing checklist or rubric. There is a subjective element that is associated with experience, skill, and ability. You will not receive an ‘A’ level grade for something that is merely average and standard. Completing all the requirements is not sufficient for an ‘A.’ That is, your mark will be based on effort and the results of that effort. Something done well is not the same as something done satisfactorily (i.e., satisfies requirements). Do not think that following instructions is the same as doing high quality work.

For example, suppose I ask you to write a sentence about feeding a cat. “My cat eats fish,” “My little black cat, Missy, loves Friskies seafood delight and will meow for an hour until I feed her each morning.” – both sentences describe feeding a cat, but the second is higher quality because of things that are hard to measure (grammar, depth of expression, insight, vocabulary, ability to elicit imagination, and so on). Code is like English in that the subjective elements of quality are hard to describe. Quality also means including appropriate citations and references. Plagiarism and academic dishonesty are clear indications of low quality!

## Project Proposal

The project proposal sets the stage for your application. The project proposal should answer many of the questions that are asked before any development and implementation takes place. The proposal should describe what it is you are planning to do and how much you expect to accomplish. It is essentially a contract between you and the instructor. The proposal should be at least 10 pages in length (longer is acceptable; please use an 11-point serif-font, justified paragraphs) and should contain the following content.

**Title Page and Abstract:** This includes the title of your project, your contact information, and the abstract -- a brief 100-word summary of what you intend to do and that briefly describes the application you are proposing.

**Introduction and Background:** You should set the stage for your proposal, describing what your project will be, why this is an interesting idea, and what you intend to accomplish. You may also want to include any background or other relevant information that the reader should know before reading the remainder of the proposal. It could be useful to describe similar applications and how you differ from them.

**Users:** Identify your target audience. That is, who you want to use your application. Explain how you are appealing to this audience. **Be specific; details matter!**

---

<sup>2</sup> See [https://en.wikipedia.org/wiki/Software\\_quality](https://en.wikipedia.org/wiki/Software_quality) for a summary and more details.

**Purpose, Benefits, and Usage:** Describe what the purpose of your application is and what benefits users will derive; i.e., Big C Context -- why would users (as identified in the previous section) use this application? Where and how will this application be used? What constraints (e.g., connectivity, location of use, memory needs) will impact the application? What is the user's state of mind (e.g., distracted, under stress) while using the application and how will it affect use? That is, provide both the Big C and small c contexts for this application.

**Functionality:** This section describes all the functions that your application will (hopefully) perform. As mentioned previously, the functionality should be divided into three parts:

1. **Minimum Functionality** – what you must complete for a minimally functioning application that does something.
2. **Expected Functionality** – what you plan to complete for a useful application.
3. **Bonus Functionality** – what you consider optional; useful functionality that your application can do without, but which really makes it better.

Note this is a very important section because impacts my expectations and your project grad. Achieving minimum functionality is typically required for a C grade, most expected functionality is required for a B, and some bonus functionality is expected for an A. However, I would prefer that you “dream big” and have too much functionality that you can't complete than you be overly conservative and finish all the bonus functionality of a simple and unimaginative application. **I expect this section to be the longest and most detailed part of the proposal.** As far as I am concerned, you can never “over-specify” an application. Failure to plan is planning to fail!<sup>3</sup> There is no required format for your functional specification as your content and detail matters more than the format.

### **Ensure that you provide all the necessary use-cases!**

The use-cases (i.e., user stories) will provide you with the foundation upon which to develop your design in the next part of the project.

**Architecture:** My first question is always, “**have you reduced or eliminated the back end?**” In this section, describe the high-level organisation of your application. While it is entirely expected that this could change throughout the design and implementation of the application, an initial high-level design demonstrates that you have done some research and considered some of the issues that you will need to contend with throughout the project.

**Tools:** What tools will you use for project management, progress tracking, fault tracking, and so on? You are not required to use any tools other than Android Studio and git, but it may be beneficial to do so.

---

<sup>3</sup> Attributed to Benjamin Franklin.

**Other:** You may also want to include a risk assessment, some preliminary test plans, or implementation planning, perhaps using timelines, Gantt charts, and other illustrations. There is no required length on the document, but it needs to clearly demonstrate that you have thought deeply about what needs to be done and have some ideas on what you will need to do. The one thing I can stress is that the more evidence I see of critical or creative thinking, the more likely you are to achieve a high mark. Simply using the structure and titles presented here (e.g., functionality, architecture, tools) is not evidence of critical or creative thought.

## Learning Outcomes

This project supports Course Objectives a. to f. from the course syllabus.

In addition, upon completion of this project, students should be able to:

1. Evaluate and research application ideas for viability, desirability and feasibility.
2. Create an ad-hoc and informal software requirements specification based on use-cases.
3. Integrate a user-centric approach (e.g., target audience focused) with application and feature design.
4. Express their development goals accurately and concisely in a written document.

## Evaluation Criteria

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

This assignment will have a mark out of 40. I expect the average score for a component to be somewhere between a 7 and an 8 (7 is the 70% pass that a graduate student needs). I will give half marks (e.g., 7.5, 8.5). 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 grammar is so poor as to significantly hinder the document's readability).

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.

When marking, I ask myself:

- Is your idea **creative and original** or is it simply another attempt at solving a well-known problem (e.g., another “To-Do” list application or something equally common).
- Did you put some thought into the application? Is it simply “obvious” or are there insights that show a deeper analysis of the problem and domain? Is there **critical thought**?
- Did you do **sufficient research** of the domain? Do you know your competitors and what has already been done?
- Does everything you say make sense? Is there **sufficient detail**, consistency, and accuracy?
- 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 and has little to no research or critical thinking?
- Does this look like a bunch of stuff found on Google that shows little to no originality?
- Is there a lot of extra “fluff”? Longer isn't better if the extra stuff doesn't support and add to goals (satisfying the outcomes).
- Do you have all the required parts? Is this a complete report?

## Document Requirements

I expect your submissions to adhere to the following requirements:

- i. Paragraphs should be justified.
- ii. Use of a font with serifs because it is known to be easier to read.
- iii. A satisfactory reference list with no secondary references and associated citations in IEEE, APA, or any other documented format. (Mendeley Cite is a good tool for references)
- iv. No use of point form. Minimal (if any) use of bulleted or numbered lists.
- v. Good, but not great, grammar. Put your effort into thinking, not into writing. I'd rather see original work with weak grammar than content with exceptional grammar that was written by someone else.

# Mobile Computing Project Proposal

## Appendix A: Marking Rubric

Component	Unacceptable: 1-2	Minimal: 3-4	Good: 5-6	Really Good: 7-8	Exceptional: 9-10
<b>Idea:</b> 1. Feasible? Can you do it? 2. Desirable? Do users want this? 3. Viable? Would it be successful? 4. Target audience clearly identified?	Only a few sentences or very confusing and doesn't make sense. Off-topic. Not really feasible, desirable, or viable. No target audience identified.	You address one or two of the 3 criteria, but there is clearly a lot missing. Could be made feasible, desirable, and viable. A "Frankenapp." Describes an audience but the audience is the wrong one or wasn't detailed sufficiently	Picked an acceptable idea that meets all 3 criteria and partly explains why it does. Something sensible, but not unique, interesting, or useful. The suggested audience is reasonable and is described somewhat but not in detail.	Proposed an acceptable app. Justified their reason for picking it and explained those reasons well with respect to the three criteria. An app that has a good chance of being used. Audience is adequately described and indicates an understanding of user perspectives and expectations.	Proposed an acceptable app and explained it well. Provided excellent resources to help justify choices. Very creative and original ideas. Answered everything well and thoroughly without adding extra "fluff." Concise. Not much or no backend needed.
<b>Research: Have you done suitable exploration of the domain, the competition, the target audience, and so on?</b>	No or very minimal research.	You provide some indication that you did more than just think something up to create an assignment.	You have satisfactory ideas that suggest you understand the domain, but it is not a deep or insightful understanding.	You know the domain and have examined it sufficiently. You know what has been done, what users expect, and what will work. You convince me that you know the topic better than average.	You present yourself as a domain expert, with a deep understanding of the problem domain, other similar apps, and other issues. I am convinced you know the topic at an exceptional level.
<b>Detail: Is this vague or precise, particularly the functionality?</b>	Little to none. Everything is vague or confusing.	You have a few details but a lot is vague and uncertain.	You describe things to the level that I can understand what you intend.	You describe things in a way that permits implementation, but a lot of questions will need to be asked.	Everything is described at a level in which it could be implemented without many questions asked.
<b>Presentation</b> 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.	No references. Not very attractive but meets expectations for the level of student.	Weak references (e.g., all websites), weak formatting of references. Looks OK, but not professional (poor fonts, low quality clip art, bad kerning/spacing) etc.	The document has perfect grammar and spelling. It looks professional. References are thorough and not just websites. Content is well organised. Not a lot of extra "fluff" – it is concise. Almost no room for improvement at all.