Now that you have worked through an entire development cycle for a mobile app, it is time to examine your personal approach and assess how it has been expanded or honed through this process. Compare your current design and development approach to the perspective you had prior to engaging with this course. Then, in your initial post, answer each of the following questions:

* How do you interpret user needs and implement them into applications?
* How do you approach designing applications?
* How do you approach developing applications?

In response to your peers, think about how their approach overlaps with your own and answer the following:

* What has your peer done effectively?
* What ideas from their approach would you like to apply to your own? Why?

Successful app development starts with understanding what the end user needs. By gathering critical system & functional requirement. This is direct feedback from the intended audience from the app’s inception to the app’s rollout. This feedback allows the developer to shape the app in real time to address concerns.

When designing an app, I consider outlining the business end of it & the user needs to create a plan that can prioritize both while maximizing features & minimizing costs. Choosing the right design model helps unify the app’s components into a cohesive system. Add in some visualization, sketching wireframes for the UI & presenting balance with aesthetics.

Development should be progress increments, while features are slowly implemented in order of importance. Each iteration involves thorough testing to validate functionality before moving forward. Only after delivering a working version should developers refine performance, polish details, and address edge cases.

Lastly, maintaining the life of the app after deployment. Having that post launch feedback, interactions with real users, & responding to that feedback is a great way of meeting the ever-changing needs of the user. The process of planning, building, testing, & refining keeps the application fresh & relevant.

Evening Dani, last week. Been a rush trying to get to the finish line.  
Before this course, I approached app development as a challenge—focused on making the app work rather than trying to understand the end user. Now, I start every project by stepping into the user’s shoes. When building my weight tracking app, I didn’t just code a login system; I designed it for a fitness enthusiast who might need fast, nutrient tracking plans to min – max at a moment’s notice. This shift from "does it function?" to "does it solve real problems?" has transformed my process.

Moving forward, I’ll carry two key lessons: First, sketches and wireframes are non-negotiable. Spending 20 minutes mapping user flows in Android Studio’s design tools saves hours of rework. Second, an app is never "done." Just as I now use post-launch feedback to improve my projects, I’ll build time for continuous refinement into every timeline. The best apps don’t just work, they adapt and so must how we build them.

Evening Steven, nice write up. Short & straight to it.

I appreciate how clearly you’ve articulated your shift from functionality-first to user-centered design—it mirrors my own journey. Your example about restructuring the notifications screen resonates; like you, I used to prioritize technical workflows over intuitive flows, but now I start by asking, *"When would a user actually need this feature?"* Your approach of mapping the user journey before adding complexity is something I’ll adopt, especially your emphasis on simplicity as a foundation. It’s a smarter way to avoid over-engineering.

Personally, it’s an impressive achievement to have that the discipline to focus on modular coding with some agile testing. Though we been learning the skills and getting to touch on them throughout our classes, I’m slower to implement them. It dooes help save oneself from last-minute integration disasters. Sometimes I can only press undo so many times to start over and see why that error occurred. Your progress inspires me to keep pushing for better technical prowess.