

## 0.1 Task assignment

During the second team meeting we went over the requirements for deliverable 1 and made a list of all of the things that needed to get done. We quickly saw that this was a significant amount of work and that there were some dependencies between the tasks (ex. do interviews before use cases). We divided the tasks into 3 categories: 1) Information gathering, 2) Consolidation/Report, 3) Coding and prototyping. A cursory evaluation of the workload vs. the deadline showed that we could not possibly deliver all of these tasks on time. Rather than cut out the early-stage prototyping entirely, we decided on making a priority matrix of our tasks with the simple rule that higher priority tasks should be completed before an individual started work on the lower priority ones.

We reasoned that the top priority items were those on the critical path of deliverable 1. We all had a good idea of what a calculator should be but we also knew better than to make a product for the developers. Consequently, we absolutely needed the information from the interviews as soon as possible so that we could align our efforts with what the calculator's potential users wanted. This step was so critical and urgent that we put 4 people on it with highest priority.

We then consolidated and paraphrased the interview data. We were surprised at just how much information we got from 5 interviewees. Firstly, we extracted the key features each user wanted. Secondly, we distilled down each interview into the key themes that were important for that interviewee. Thirdly, we were able to classify our interviewees based on their expected usage of the calculator (basic use, mathematical use). Lastly, we looked for commonality across the different wants of the interviewees and obtained what we think is a much better approximation of what the market wants from a calculator.

One team member was tasked with coming up with an outline of our testing strategy. We decided that since the users all seemed to value accuracy of the mathematical function on the calculator then a test-driven approach to development would be appropriated.

Another person was tasked with researching the algorithms needed for the implementation of the non-trivial calculator functions. Some functions had multiple algorithms that varied in complexity. For this first iteration, it was decided that simplicity should be the key criteria in the choice of algorithm since the deadline was so tight. We all agreed that we could reevaluate this strategy for the next iteration.

The very last priority was the implementation of the mathematical functions. Some team members were eager to start coding but we decided that it would be a much better idea to focus on the requirements gathering at this stage of development. Consequently, only a very rough implementation of some of these functions appear as code.

Table 1: Task Priority Matrix

<b>Name</b>	<b>Priority 1</b>	<b>Priority 2</b>	<b>Priority 3</b>
Ashkhan	Interviews/personas	Testing strategy (TDD)	$a^x, 10^x$
Daniel F.	Interviews/personas	Algorithm research	$x^y, \sqrt[y]{x}$
Daniel T.	Coordinate activities	Use case diagram/desc	$e^x$
Jaime	Interviews/personas	Interview summary	$\cos(x), \sin(x)$
Justin	Interviews	Strategy	$\ln(x)$