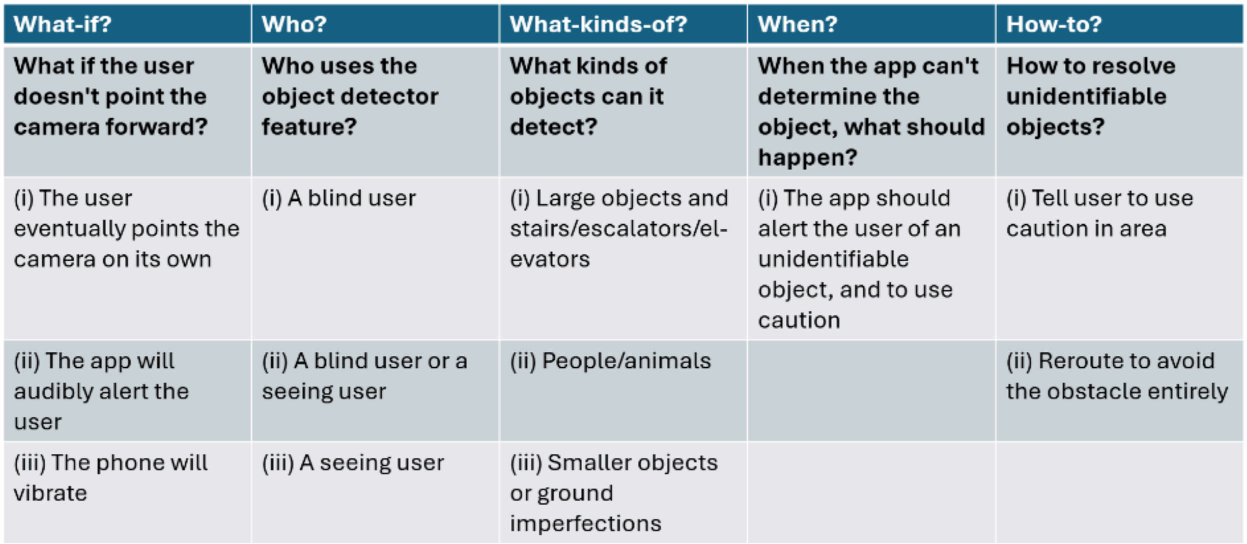
Process Specification

Our Process:

First, we needed to decide as a team what 3 core features we wanted to implement for the project. We ended up picking simulated navigation (with re-routing), companion settings, and simulated fall detection. Next was the UML and IDEF0 designing for the features, which was done by Riley and Shawn. We also met to decide what resources we wanted to use to develop, and we decided on Python and Flask, with the help of Javascript. Finally, we started developing the project itself. Jace and Shawn laid out the template for the whole project, followed by Shawn, Jace, Anthony, and Eli fully developing all of our chosen features. Some subfeatures, such as our speech to text feature for using commands, went through a few iterations that Anthony and Eli perfected.

RE Analysis:

The following is our analysis for the original plan we had come up with:

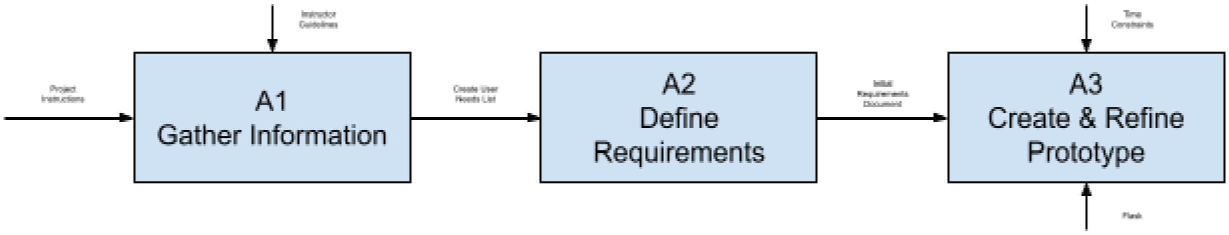


What-if?	Who?	What-kinds-of?	When?	How-to?
What if the user doesn't point the camera forward?	**Who uses the object detector feature?**	**What kinds of objects can it detect?**	**When the app can't determine the object, what should happen?**	**How to resolve unidentifiable objects?**
(i) The user eventually points the camera on its own	(i) A blind user	(i) Large objects and stairs/escalators/el-evators	(i) The app should alert the user of an unidentifiable object, and to use caution	(i) Tell user to use caution in area
(ii) The app will audibly alert the user	(ii) A blind user or a seeing user	(ii) People/animals		(ii) Reroute to avoid the obstacle entirely
(iii) The phone will vibrate	(iii) A seeing user	(iii) Smaller objects or ground imperfections		

Although we scrapped a lot of the ideas these questions came from, this analysis helped to keep us grounded when coming up with our new ideas. The following is an updated analysis for our new implementation:

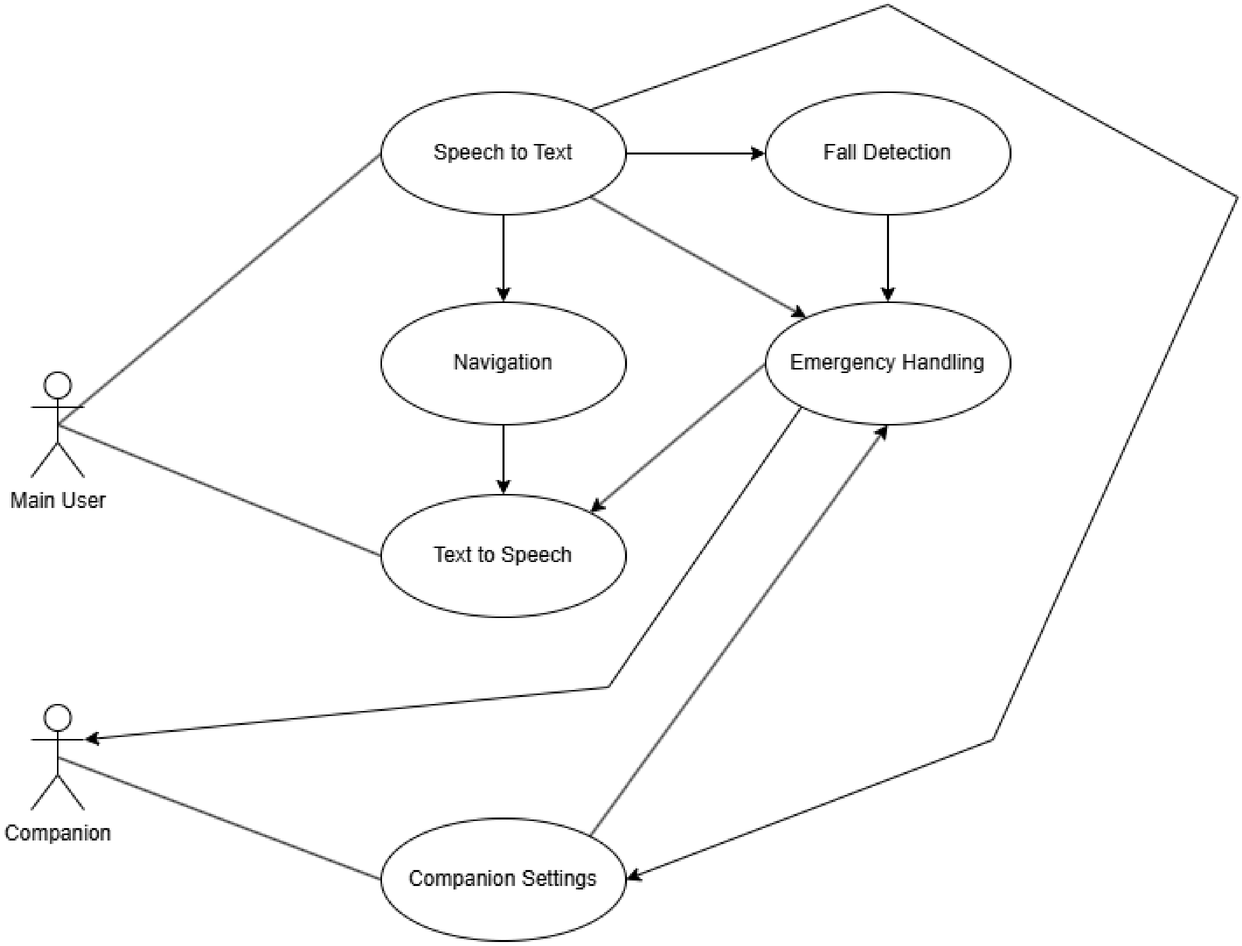
What-if?	Who?	What-kinds-of?	When?	How-to?
What if the user attempts to go the wrong direction?	**Who uses the fall detection feature?**	**What kinds of commands can the app support?**	**When there are multiple routes, how does the app pick?**	**How to determine who to call for help?**
(i) The direction leads to an alternate route, so navigation continues	(i) The primary user to get help	(i) Navigation commands	(i) The route that takes the least distance to travel	(i) Use settings to find companion info and alert them
(ii) The direction is invalid, so the system should attempt to figure out where the user is	(ii) The companion user to help the primary user	(ii) Emergency commands	(ii) The route that has the least amount of instructions	(ii) Play emergency message out loud for surrounding people
	(iii) Anyone nearby who hears the emergency message	(iii) Companion settings commands	(iii) The first route it finds	

IDEF0 Diagram:



UML Diagram(s):

Use Case:


Speech to Text
Fall Detection
Navigation
Emergency Handling
Main User
Text to Speech
Companion
Companion Settings