Zachary Taylor CSSE371

Reflection Questions:

1. The client is attempting to create a way that automatically records the time a task took to be complete. The problem they are trying to solve is the inherent inefficiency in recording the start and end task and then calculating the time elapsed.
   1. Needs: “TODO list is terrible in every way”, “Need it swap out tasks easily” Task management in general (adding, recording, swapping).
   2. Features: “A single drop down that lets you select the task you are working on with a single checkbox to start and stop the timing”, “TODO list is compact”, “TODO list should have the option to import and export tasks in Ovo studio format”, “automatically checks off tasks when it detects that tasks were completed by the user”, “have the option to mute the audio”, “an audio mixer on the list”, “can access the camera driver”, “Desktop and mobile versions”
   3. Software Requirements: “Represent the list of tasks internally as a linked list”, “likely needs to be written in C so it can access the camera driver”, “When we load a TODO list that was in progress, it should keep the timing data form the previous tasks”
   4. There are no unmet needs.
   5. There is the extra feature of automatic list checking, muting audio, an audio mixer, access the camera driver.
   6. “likely needs to be written in C so it can access the camera driver”, the client does not currently need access to the camera driver to fit their needs.
2. They seem to add on tons of bloat features to a relatively simple request of managing a list and UI. There does not seem to be a grasp from them how complex some of the feature’s they are asking for are, especially the automatic recognition part.
3. This simple list is rife with things that are almost impossible for a 3rd year CS or really anyone to implement and therefore it is likely that this is part of the conversation with a client wherein pushback and prioritization must exist given that the time for this project is kind of short for anything of grand complication (see automatic detection). Likely this is to persuade the CS to contact the client and ask if all of these requests are really needed for their software and to assert that they cannot do some things in the amount of time, like audio mixing. It also is to force them to ask for clarification on things that they don’t understand (likely Ovo studio format).

**Issues found:**

|  |  |  |
| --- | --- | --- |
| Issue | Problem | Resolution |
| Automation of task completion detection | This is something that is far beyond what I am even aware of and I am unsure if this could even be performed in any manner. It would require some preset parameter I suppose. | Contacted the client and discussed with them if automation was really needed. We came to the conclusion that it was not. |
| The need for a Camera driver causing a language restriction | I don’t think I could write general software for a camera driver in a few days and I wasn’t sure why they would need a camera driver for what sounds like a simple list of times and task names. | Contacted the client to ask what their needs were for a camera driver. We determined that it was tied to the automatic task completion detection which is now defunct so we agreed this feature was no longer needed. Which also removes the client’s language restriction request. |
| The need for an audio mixer/ability to mute audio | I have no idea how to implement this or why a TODO list would have this feature. | Contacted the client to ask what their need was for both the audio muting and audio mixer. We came to the conclusion that they could mute the system audio removing the need to mute audio. We decided that I would not have enough time to make an audio mixer and have removed the request. |
| Adding a task being a simple drop down | While this was not really a problem the comment came up in discussion. | Contacted client and they had changed their mind on how adding tasks was to be implemented, instead now with a text box below the list with a button to add. |