**Group Members:** Ana Cehan, Sasha Holt, Nick Baker, Emmanuel Ankutse, Marcus Waller, Thomas Luo, Anudeep Metuku, Carlos Montemayor

To-Do Proposal

### Project Schedule

|  |  |
| --- | --- |
| **Milestone** | **Date** |
| Proposal Acceptance | February 6th |
| Schedule | February 6th |
| Storyboards | February 8th |
| Design/Decomposition | February 13th |
| Design Review | February 14th |
| Test Plan Review | February 20th |
| Code Complete | February 25th |
| Unit Test | March 1st |
| Integration Complete | March 6th |
| Demo | March 8th |
| Integration Testing Complete | March 12th |
| Delivery and Final Acceptance Test | March 13th |

### Special Problems

Please note that proposed solutions to said problems are found in the indented portions.

* Prioritizing objects
  + Can be solved via class hierarchies and extending one parent abstract class
* Dragging and dropping objects
  + Having ample testing time for this feature
  + Having a "ghost" object that tracks the user's proposed change
  + "Ghost" object also lets user know how they will be changing the onscreen data
  + "Ghost" object also helps with back-end testing
* Visually pleasing and professional display
  + Multiple design iterations are needed; can use profession layouts for inspiration
* Making it easy to understand and quick/easy to use
  + Simplicity is key
  + Class organization should reflect our overall goal
  + User cannot be overwhelmed at any time
* Having complete objects disappear (for navigation)
  + Having ample testing time for this feature
  + Having visibility methods integrated into classes and objects
* Menu Bar that remains constant despite user actions
  + Making this a separate class and UI section that is unchanged yet interactable
* Making double-clicking bug-free
  + Having ample testing time for this feature
  + Making a "click counter" integrated into objects and classes for inter-class communication and refreshes
* Glitches from rewriting files
  + Having ample testing time for this feature
  + From RFP: "write to a new file and only when that write is complete,

rename the new file to replace the old. When the user quits or dismisses a window, the program should complete any write operations before it exists."

* Easily navigable; few clicks to do what user needs to do
  + Aka has to have excellent design
  + Will most likely require several design iterations to perfect (use group's diverse ideas)
  + Some time required for group to agree on a unified theme
* Scrolling down a window (in pop-up windows as well)
  + Research and utilize built-in Java resources for this specific purpose
  + Allow ample testing time for integrating this into existing, coded segments
* Right-clicking
  + Research and utilize built-in Java resources (mouse listeners) for this specific purpose
  + Allow ample testing time for integrating this into existing, coded segments

### Issues and Suggestions

* It would be interesting for a group to create the To-Do list specifically tailored for the AOOD group project (a kind of CANVAS without the terrible design and technical issues)
  + Each group member could "log in" and see their individually assigned tasks, due dates, what others are working on, etc.
  + Each group member could also add their own smaller tasks, set personal deadlines, reorganize existing lists (if they have permission, that is), etc.
  + The lead engineer, meanwhile, would have superior access and be able to add/manage group members' tasks
  + Extra ambitious: different groups could have a space to share images of what they are working on/communicate with other groups