

Term project: "Tiempo"

Description: A scheduling and time-management app. Allows you to create, modify, and automatically "interleave" your calendars via a sleek UI. Has full integration with Google Calendar/iCloud Calendar/Outlook Calendar meaning you can import files from these services, modify them, and export them back as you wish. Also includes a unique task-management system that makes tasks equally as present on your calendar as events (unlike other apps) and which allows you to start immediately working with a built-in pomodoro timer.

Competitive analysis: There are a few categories of "competitors" that I decided to look at. To start, for 15-112 projects related to calendaring, there were only two that I could find: Jiayi Wang's Weekly Scheduling and Rachel Luo's Intentional Time. Both had a lot of the same functionalities regarding event manipulation and Rachel's did something similar to my idea of interleaving (albeit slightly less complicated since her program moves around 0 second time intervals which are supposed to be prayer periods, whereas mine will be dealing with larger time intervals). Nonetheless, neither attempted to deal with ics files and instead created their own system independent of ICS. Also, neither one incorporated tasks directly into the calendar view (or did a pomodoro timer) which is something I'm trying to do to differentiate my calendar app.

Structural plan: File organization will be as follows:

- main.py -- stores main() and imports other files below...
- ics_parser.py -- contains function that takes in paths to an ics file, parses the file (harder than it sounds), and returns a "week" or "month" data object for use in...
- calendar_graphics.py -- contains MVC functions that display data in "week" or "month" objects and manipulates that data in the model according to events in the controller (e.g., dragging an event across days)
 - interleaver.py -- called by controller function in calendar_graphics.py. Takes in immutable event choices, wake up and bed times, and finally "week" or "month" objects and converts them to real number intervals that represent free time throughout the week. It then takes the mutable events, cuts them up into 1 hour intervals, and recursively finds a way to fit them into your calendar such that different mutable events alternate (note this only works with >1 mutable event).
- ics_converter.py -- takes our new calendar with all of its changes and edits the original ics file to reflect the modifications (then somehow allows you to export it... perhaps just manually through file system)
- task_graphics.py -- contains MVC functions for tasks
- pomodoro_graphics.py -- (different mode) contains MVC functions for pomodoro timer