Project Spec

“Step-by-step:

A program for teachers to track their classroom development”

Silvia Alberti

College of Design, Iowa State University

HCI 5840

Professor Harding

June 4, 2025

Step-by-step: A program for teachers to track their classroom development

General description of the project (2 pts)

How would I describe this to a non-tech person, perhaps a teacher?

Step-by-Step is a Python desktop program for teachers to help them track their class's development. The program will allow the user (teacher) to log each child's progress by interacting with the interface and track children' development, make observations, and conduct analyses. The user will be able to pick a developmental category (such as "mobility") and track the progress according to the state developmental charts. The program is meant to streamline the user's workflow while also providing a clear overview of the individual, which is helpful for any future interventions and/or sharing purposes.

The program is meant for teachers in early childhood education, and possibly elementary as well. It is not targeted directly towards middle school and high school, although it has the potential to be re-adapted to the appropriate development stages.

Moving forward, I would like to further target this towards classrooms that specifically have children with disabilities, as it is the focus of my thesis. However, for now, as I am not in the education field, and therefore not an "expert on the subject," it might be simpler to build the program around general developmental milestones and maybe re-adapt it to the specifics of disabilities in the future.

As for external tools, I am thinking of using Pandas as well as Matplotlib, which I am mostly already familiar with. Pandas will make the data collection much clearer and more

organized, while Matplotlib will allow me to create data graphs and charts for the user whenever they want to visualize the data. For my first version, I believe that I won't need AIs or GUIs; however, I am open to implementing those in version 2.

Version 1 Features:

I was thinking of focusing on one child, jumping directly to the "post-login" page as suggested in the lectures. I was thinking of having predefined categories in a dropdown for each developmental aspect. However, since I will not be using a GUI yet, I was going to number the options in the CLI version. Additionally, it would be nice for the user to have the option to add a note for each log to better specify what exactly the log refers to. With that, I am assuming I will have a free text input as well.

As mentioned, Matplotlib will be used for data visualization. Lastly, I was not planning on having a file saving/export feature for the user to use yet, but that is something I can definitely implement in version 2.

So, do I have any ideas for version 2?

Yes! I already discussed them above, but I am going to summarize them below:

1. Creating a GUI, possibly using Tkinter as discussed in the class lecture

~~2. Adding the option of tracking multiple children (therefore a classroom)~~

I have decided to drop this feature for now. The reason I took this decision is due the fact that I would like to focus on the GUI and the exporting of the data for Version 2. I

would rather do that and then, in the future, expanding the project to multiple children/classrooms.

3. Having an option to store and save the data, maybe via CSV
4. ~~(Long stretch goal) Using AI to parse text and completely get rid of the dropdown. For example, the teacher will be able to simply type a prompt describing a situation for a child, and AI parses it and "puts it in the right spot." I understand this might be a bit complicated and could produce some errors as AI isn't perfect yet, so I am not sure if I should shoot for it, but I figured to mention it anyway. Any advice will be appreciated!~~

Same as the classroom expansion above, this is still on my list of improvements, but it might need to be a long-term goal rather than the following 2 weeks.

Task Vignettes (User activity "flow") (4 pts)

VIGNETTE 1: Logging a new observation

The teacher observed a child, Nico, helping a peer during playtime. The peer is clearly frustrated, and the child reassures him, saying "it's okay, let me help you," and together the two children accomplish a task. After noticing, the teacher opens the CLI, selects the developmental category "Social Emotional Development," and inputs a number that corresponds to one of the social emotional development milestones (see chart below for reference). The teacher also adds a note: "Nico helped comfort a crying friend and assisted him during playtime." The observation is then stored and associated with the category and Nico's overall progress.

Technical details:

- Category selected via numbers on CLI menu
- Number and note input inserted as free text
- Appending new entry to the Pandas data frames
- Dataframe 1 (category related) will have timestamp, number, and note
- Dataframe 2 (all logs) will have an increment of his progress in the social emotional development category

VIGNETTE 2: Viewing Nico's Social Emotional Progress

The teacher is having a parent-teacher meeting with Nico's parents and is telling them how Nico has shown a lot of social emotional development skills lately. They ask for specifics, so the teacher pulls up the platform and refers to each log, showing them the notes as well as the overall progress milestone chart in the category.

Technical details:

- Viewing the Social Emotional Development dataframe for Nico
- Aggregating entries by date in chronological order
- Using Matplotlib to plot a chart

VIGNETTE 3: View overall Progress for Nico

This time the teacher wants to see if Nico is needing some help in other categories. The teacher recalls logging a lot of social emotional progress for Nico but can't remember the last time she logged Nico's cognitive progress. So the teacher opens Nico's profile and

looks at the overall pie diagram for the kid. As expected, Nico is lacking cognitive development, so the teacher plans a lesson to help him on that aspect.

Technical details:

- Viewing Nico's overall progress with the all logs Data frame
- Grouping logs by category
- Using Matplotlib to plot a pie chart shing the proportions for each category

VIGNETTE 4: Logging a new observation through the GUI **NEW**

During free time, the teacher sees Nico working on a puzzle. He uses trial and error, talks himself through it, and later explains his strategy to a classmate. Afterward, the teacher opens the Step-by-Step app and picks "Cognitive Development" from a dropdown, types in the milestone number, adds a note, then hits submit. The log is saved right away and added to both his cognitive progress and overall tracking.

Technical details:

- Category selected via dropdown in the GUI
- Milestone number and note typed into input fields
- Submit button adds log to both dataframes
- Entry saved to CSV, so it's not lost
- Log appears in both the specific category and overall progress tracking

Technical "flow" (3 pts)

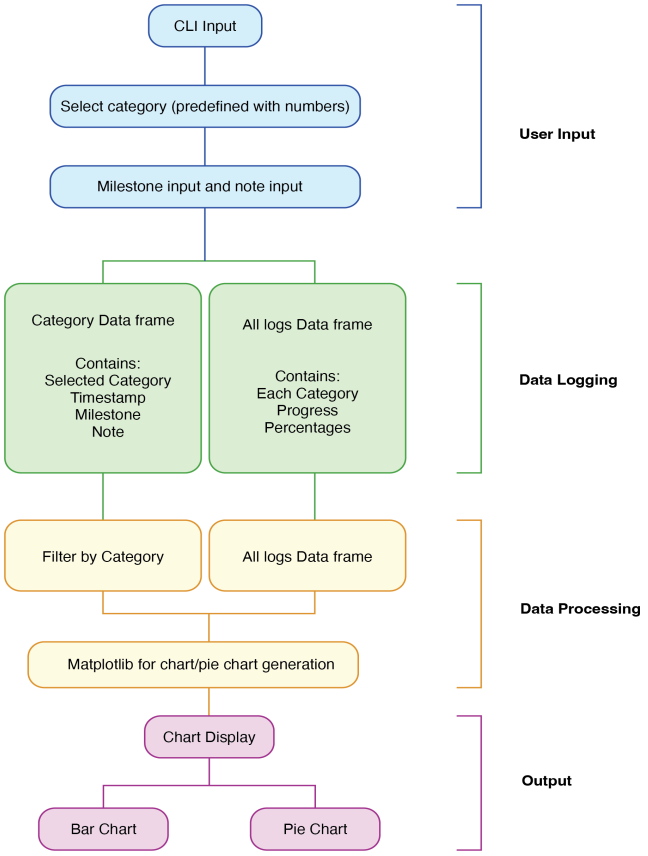
As I mentioned the program will start with the teacher interacting with a CLI. The teacher is presented with a numbered list of developmental categories (such as Social Emotional Development, Mobility, etc.). After selecting a category and entering the corresponding number, the teacher inputs a milestone number and adds a free text note describing the observation.

Once the input is sent, the program appends the data to two data frames:

- The category-specific Data frame: it will contain all the logs for that category alone (such as social emotional development), including a timestamp, milestone number, and the note, which will then be visualized in a graph showing the progress.
- The all-logs Data frame: it will contain data on all the categories of the development, which will then be visualized in a pie chart with % as proportions.

When the teacher needs to visualize the progress in a specific category, the program filters the relevant category Data frame, sorts logs by date, and uses Matplotlib to generate a chart showing progress over time. If the teacher wants to see overall progress for the child, the app processes the all-logs Data frame, grouping the entries by category, and displaying a pie chart showing the proportions.

Since all of this will happen in memory only, no data will be saved/exported in version 1. Below I tried my best to illustrate the flow I am imagining, using Adobe Illustrator.

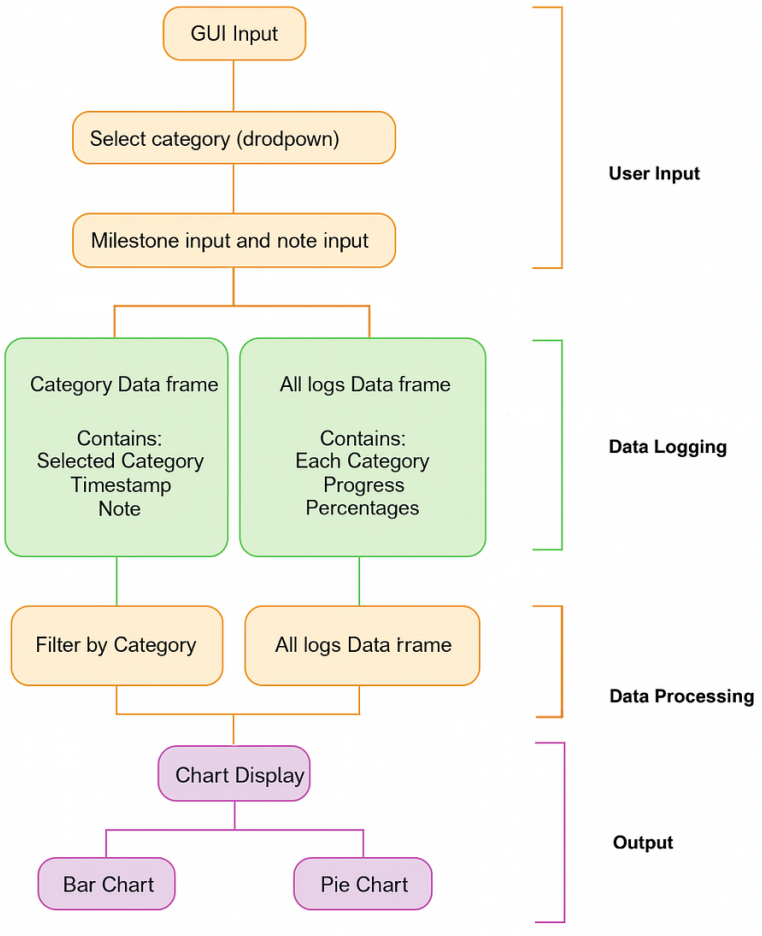


Technical "flow" For Version 2 NEW

For Version 2, I'm planning to move everything into a GUI so that it's easier and more intuitive to use, especially for teachers who aren't familiar with working in the terminal. The process will stay the same: the user selects a category, enters a milestone number, and writes a note, but instead of typing into a CLI, they'll interact with a clean layout made of dropdowns, text fields, and buttons. I also want to make sure that inputs are validated, so the program won't crash if someone types something weird (like a letter

instead of a number). Additionally, I would like to turn the lo-fi GUI into something more Hi-Fi and related to my field of expertise (graphic design and UX/UI)

Another big change is that logs will be saved. In Version 1, everything disappeared when the program closed but now would like to make sure the new logs get written directly to a CSV. The graphs will still work like but this time I want to make them pop up in the GUI itself so the whole experience feels smoother and more complete. I also would like to implement an export option so the user can save the tables and graphs.



*Also updated to reflect Version 2

Final (self) assessment (1 pts)

After working through the spec, the biggest change I had to make from my original idea was narrowing the scope down to just one child and removing all GUI elements for version 1. Originally, I was thinking more visually and had imagined something like a dashboard interface (that was the UX Designer in me thinking), but I now realize that staying within a CLI structure is more realistic at my current skill level, and it can also help me better understand the "mechanics" of my program without overcomplicating it with a GUI yet, which might end up confusing me whenever I am troubleshooting and debugging.

I feel fairly confident that I can implement the specs I have written. I understand the logic and structure of how the data should flow, and I'm already familiar with using pandas and matplotlib, which are the main tools I'll rely on. Writing this document also helped me clarify all the ideas I had, and I now feel much calmer and less "scared" of Python.

To be completely honest, I think the hardest part/biggest risk for me is and will be losing my confidence. Coming from the design world, I have always seen coding as such a complicated world that I was not confident I would ever be a part of. Writing this PDF helped a lot, but down the road I hope that the confidence will stick with me whenever I encounter an obstacle. I tend NOT to give up, so hopefully that doesn't change! Technically speaking, the CLI interaction and having it update multiple DataFrames cleanly might be the biggest challenge. I also need to make sure that the filtering and charting steps are working properly without overwriting or losing any data.

So far, is it still a learning curve? Yes (thankfully, otherwise what am I even doing here?). But it is a learning curve that no longer scares me. With that, I do not have anything in mind that I CURRENTLY (and desperately) need help with, yet. I know how I work, and I can be very persistent. However, I may need some help down the road, especially after version 1, as my next features are a bit more challenging. For version 1 though, I think once I figure out how to get past the hardest parts, I can do it. (**NEW** I have included my latest reflections for Version 1 in the other document "Alberti_HCI5840_ReviewDoc_Version1")