Each week, add a new block **above** the existing blocks and fill it out by Sunday (that is, at the end of the quarter, the document should have Week 10, then Week 9, then Week 8, …). Each Sunday, submit a PDF of your doc as-is to Gradescope. Replace XXX in the title with your group number (e.g. A16-1). See here for more instructions.

Week 7 (due Sunday, February 23th)

Outside of your meeting with your mentor, when did your group meet this week?
Time: All throughout Saturday and Sunday
Location: Discord
Attendees: Phu Dang, Eric Chen, Daniel Li, Bofu Zou

What did each group member do this week?

- Eric: I worked on understanding the prompts.yml configuration file and fixing a line which initially was allowing for unrelated queries to be answered by the agents. After that, I was able to implement the basic logic for queries that were related to smart homes to be answered by the sql agent and to divert unrelated queries away by simply telling the user that the chatbot cannot answer questions like that.

- Daniel: I mainly worked on developing the skeleton of our project website, which includes the layout as well as the actual contents that we'll put in the website. I also briefly assisted Eric and Bofu understand the prompts.yml and config.yml files.

- Bofu: I worked on implementing the RAG content in the chatbot. I implemented additional tasks (bill classification, RAG and RAG evaluation) in the prompts.yml and added conversation flow in the config.yml. I transferred the information of the bill pdf into a txt file so it could be easier extracted when RAG is triggered. However, due to the issues in the generate function, I have to put inline prompts in the chatbot.py to execute the RAG functionality.

- Phu: I worked on planning out our integration strategy in NeMo. I realized a challenge regarding how NeMo works, where NeMo's output rail implementation is exclusively meant to run on NeMo's response (or OpenAI's response through NeMo to be precise) to NeMo's own input rail, meaning both input and output rails execute in the same "rails.generate()" call, which we do not want as our actual LLM response generation process is separated from NeMo. We want to first initiate the input rail, if passes, the user prompt will be passed to the LLM for the main response, then that response will get passed to an output moderation rail to ensure all requirements are met. We do not want this entire process to happen in one NeMo execution call as that gives us limited ability to modify the process.

Where did they get stuck?

- Bofu: In the original plan, after I constructed the prompts instruction in prompts.yml and config file, I can invoke the RAG functions directly in the chatbot.py. However, the llmrails.generate does not support arguments like metadata to invoke the prompts in config files. I have to put inline prompts of the RAG in the chatbot python file instead of routing via config-based flows

What will each group member do next week?

- Eric: Next week, I will be working on making sure the chatbot flows like a normal chatbot in which you can see what the responses to previous questions were. Currently, the chatbot only shows the most recent answer, and it doesn't track when you ask an additional question.
- Daniel: I will be working on the guardrails layer for our app, and definitely continue to fill out the report and website to reflect the developments my team has produced in the coming weeks.
- Bofu: For next week, I will be further working on the RAG content to solve the current issues and try to find a better way to implement it. I will also help in integrating the current work together with Phu and Eric to make the codes connect and make specific functions like Text2SQL and RAG work in a smoother way.
- Phu: I will integrate our existing NeMo framework with Bofu's RAG evaluation guardrails to hopefully have a well-functioning NeMo guardrail system. Afterwards, our final step will be hooking the system with Streamlit for our UI using the existing NeMo x Streamlit integration we have from week 4-5.

As a group, do you still feel on track to execute your project proposal?

We were slightly behind schedule due to the long weekend, but with a properly-outlined paper, beautiful poster, and working website report, we are confident that we are on track to execute our project proposal. The integration piece involving NeMo, PostgreSQL, Text-2-SQL, RAG, Persistence Layer, and Streamlit is our biggest bottle-neck, once we have a working demo, all else should be much more manageable for us.

Week 6 (due Sunday, February 16th)

Outside of your meeting with your mentor, when did your group meet this week?

Time: Friday night - Sunday night
Location: Discord Messaging
Attendees: Phu Dang, Eric Chen, Daniel Li, Bofu Zou

What did each group member do this week?

- Eric: I worked on reducing the runtime of the agent, as well as the load time on Streamlit. This required reducing the query size in the beginning and utilizing more efficient methods for the Langchain SQL agents.
- Daniel: I worked on finalizing the config file of the output moderation guardrail for quality assurance of our output. I also worked on elaborating more on the LaTex report, and continued adding new parts to our report in accordance to our recent developments.
- Bofu: I worked with Eric on cleaning up the Text-2-SQL workflow and developed RAG prompts and RAG evaluation guardrails. Coordinated with Phu to integrate the RAG eval guardrails with our latest NeMo framework. I sent some reporting materials to Daniel to plan out the skeleton of our website report.
- Phu: I started the poster powerpoint and design, we are collaborating through SharePoint. I made a logo for us (Digital Bouncers), building off of a color palette from Coolors and I made the logo design on Canva. I started a lay out for our poster using an efficient 4-column layout with designated sections taking from the most prominent components of our paper. I collaborated with the team on design feedback and preliminary poster content for the checkpoint submission.

Where did they get stuck?
We got stuck on how to actually reduce the runtime of the agents since a lot of it is being handled by the agent itself. So it required playing around with different methods, but it still was taking a long time to run until we figured out how to reduce query sizes. Our poster design process was super smooth and we are looking forward to showcasing our product.

What will each group member do next week?

- Eric: I will be working again on chaining together the logic of the guardrails and making sure it made sense to proceed. The first objective is to still just reject the user input if it did not make sense in the context of smart home energy usage. But for another goal, it would be good to see if it makes sense to use Text2SQL at all, even if it’s a smart home related query.
- Daniel: I will assist with other components of the guardraiL layer of our app. I will also take the lead on developing the skeleton for our website in preparation for the checkpoint coming up.
- Bofu: I will focus on "bill explanation" using "in-prompt context provision" as in Zayd's demo last quarter, using the "utility bill explanation" pdf we found from earlier this quarter. This RAG workflow will be initiated when the user asks about "bill

explanation". For a clean workflow, we will try to build the RAG-eval guardrails on top of the existing NeMo input moderation framework.

- Phu: I will take on more of the integration efforts and further build out our NeMo Guardrail framework to integrate with Bofu's progress on RAG, Eric's progress on Text-2-SQL query feasibility rail, and the existing NeMo frame that Daniel and I have been working on.

As a group, do you still feel on track to execute your project proposal?
Due to the long weekend holiday, we hit a roadblock in being able to make a lot of progress together. We are a little bit behind schedule in terms of tying the guardrails with the agent input and outputs, but we should be able to get back on track during Week 7.

Week 5 (due Sunday, February 9nd)

Outside of your meeting with your mentor, when did your group meet this week?
Time: 6 PM, Sunday, February 9
Location: Zoom
Attendees: Phu Dang, Eric Chen, Daniel Li, Bofu Zou

What did each group member do this week?

- Eric: Eric worked on finalizing the Text2SQL workflow to be ready for integration with our guardrails and UI. Eric worked with Bofu to set up the Streamlit UI to begin orienting our project as a product. Eric also re-organized our GitHub repository for convenient navigation and execution when others run our app — we have a lot moving components across multiple working notebooks, so a new folder system helped us better organize our code, including two primary folders: "app" containing the Streamlit and NeMo integration UI code and "config" for the NeMo guardrails configurations.

- Daniel: Daniel worked on finalizing our input moderation guardrail through NeMo to be ready for integration with the Streamlit UI and the LangGraph persistence layer. Daniel also spearheaded the LaTeX report by editing the provided departmental template, created an outline, and updated the report with our progress. Daniel coordinated with everyone to gather progress, findings, conclusions, and visual materials for our paper report.

- Bofu: Bofu worked with Eric on the Text2SQL workflow and gathered reporting materials for our fine-tuning work from last week. Bofu helped Eric re-organize our Github repository and streamlined our branching and merging workflows for smooth collaboration. With the basic Overleaf plan limiting us to only 2 editors for the LaTeX report (Eric and Daniel), Bofu (and Phu) will add reporting materials to a word document and Daniel will spearhead the transfer to Overleaf.

- Phu: Phu integrated Daniel's finalized NeMo input moderation guardrail with Eric's working Streamlit UI, as well as Bofu and Eric's Text2SQL workflow. Created the docker files and pushed the image to GitHub Container Registry. Phu also researched and implemented Docker's runtime secrets provision workflow using environment variables, rather than using Streamlit's "secrets.toml" file for enhanced security. Phu also gather reporting materials for our paper related to "compliance" with the prevalent, contemporary AI regulations and developed a narrative on what our product means pertinent to data/AI security, privacy, and "democratization" of data ownership by furthering "data minimization" best practices to inspire a future of secure digital interactions with "digital bouncers".

Where did they get stuck? (we will divide our answer into two moving components this week)

- Integration: We anticipated that the process of combining the NeMo guardrails framework with Streamlit and the LangChain agentic workflows will be very challenging as they operate in differing ways. Between the Streamlit UI and NeMo, we have a working demo. Initially, we were afraid that NeMo would not work with Streamlit because Streamlit "resets" everytime there a new input on the UI (website), we think this is because Streamlit is ultimately a "state-based" system, not dynamic, therefore every UI component and variables reset every time there is a change in input, which theoretically would erase NeMo's config setup every time — however, that's not the case and we were able to successfully integrate the two. We "sort of" have a runtime issue, however, the wait time is still within a "tolerable" range that we would use this tool in the real world. Due to the complications behind integration and getting to understand each workflow's details, we did not have the luxury to optimize runtime.

- Containerization and secrets management: We had to choose between creating a conda environment versus using Docker. We chose Docker for its flexibility, cloud-based workflow with robust push/pull of images. We were stuck for a short moment when trying to figure out the best way to manage the numerous secrets associated with our tool, including the PostgreSQL database credentials, OpenAI API key, and GHCR token. We opted to use the optimal approach, which utilizes environment variables with secrets provision at runtime (when a user executes the Docker image and run our Streamlit app locally), which prevented us from having to use Streamlit's "secrets.toml" in-file storage system or any other form of storage in files, which significantly enhances the privacy and security that we intend to encourage as the purpose of our capstone.

What will each group member do next week?

- Eric: Eric will continue to refine the Streamlit UI with Daniel and Bofu to chain the UI with the guardrails. Further organizes our GitHub repository and reporting materials (paper, markdown website, Streamlit website, and poster).

- Daniel: Daniel will develop the output moderation guardrails for quality assurance of our outputs, ensuring all components asked are included (for ex: trends, anomalies, actionable recommendations). Further develop the reporting materials (paper, markdown website, Streamlit website, and poster), especially the LaTeX paper.

- Bofu: Bofu will develop the RAG workflow and RAG evaluation guardrails in NeMo, to be integrated with the rest of our moving components. Further organizes our GitHub repository and reporting materials (paper, markdown website, Streamlit website, and poster).

- Phu: Phu will implement a LangGraph persistence layer for our chatbot (the integration with moving parts from Eric, Daniel, and Bofu) to facilitate conversations using checkpoints stored in PostgreSQL. Develop a flow diagram of our product showing visually how our moving components work together, including the "guarding" steps along the workflow (e.g. Yes/No input moderation guardrail → if yes, either Text2SQL or RAG is initiated depending on the user's inquiry; otherwise, we will deny the user's request if it is off topic or include inappropriate requests). Set up and share a PowerPoint for our poster through SharePoint for team collaboration, as well as starting the initial design for our poster.

As a group, do you still feel on track to execute your project proposal?

Yes, we are very confident of being on track to execute our project proposal, even far exceeding our initial expectations as we got two novel LangChain agentic workflows to work! Including Text2SQL and LangGraph's persistence layer. Our guardrails are chaining together with the Streamlit UI quite nicely as well, we also made strong progress with our LaTeX report and GitHub code organization. Our Docker containerization worked perfectly and was developed much faster than we anticipated, which was a huge progress jump for us.

Week 4 (due Sunday, February 2nd)

Outside of your meeting with your mentor, when did your group meet this week?
Time: 5 PM, Sunday, February 2nd
Location: Zoom
Attendees: Phu Dang, Eric Chen, Daniel Li, Bofu Zou

What did each group member do this week?

- Eric: I, along with Bofu, worked on implementing a Text2SQL agent on a Jupyter Notebook. Initially, I started with pandas DataFrame agents since it was more familiar, but then we pivoted to building off of Phu's PostgreSQL instance to work off of. Therefore, we combined the data we queried using SQL with LangChain's SQL agent, which relies on OpenAI's API. I then decided to use the prompts I created back in Week 3 to test what the advantages and disadvantages of using the Text2SQL agent would be. In the process, we also discovered an important guardrail in limiting the user to one question each time they chat.

- Daniel: This week I mainly worked on the non-technical side of the project, which consisted of the report and also the website. Made the template and wrote the abstract, and introduction sections. Additionally, I broke down all the sections and subsections that will be needed for our project, especially the methods section of our project. I also worked on creating a skeleton of our website deliverable, which will be primarily based on the content of our final report.

- Bofu: This week I collaborated with Eric to work on the Text2SQL method of implementing the dataframe to the LLM. I tried fine-tuning the model with the dataset last week but the result didn't go well. Thus, we used the langchain agent along with the OpenAI api, the dataset was installed in Phu's PostgreSQL database. With the langchain agent, the model could access the dataset by calling sql statements. We tested some prompts and it went well on the part which requires direct numerical answers from the dataset. However, the prompts that require reasonable answers couldn't benefit from the Text2SQL strategy.

- Phu: Created a Postgres instance to store the smart home data for Eric and Bofu's implementation of the Text2SQL workflow using Heroku. Worked on incorporating a limit on the number of interactions tracked by LangGraph's persistence layer from last week to manage the context window and reduce model distractions/hallucinations due to unnecessary/irrelevant contexts. Combined the persistence layer with a NeMo input guardrail to begin the final product workflow.

Where did they get stuck?

- Eric: The main part I got stuck on was figuring out where we would even start with the Text2SQL implementation because our mentors gave us a resource which primarily relied on AWS platforms. We had nothing of the sort, so I had to dig through some articles and found one on Medium which talked about LangChain. It was also difficult not reverting back to pandas because that is what I am used to.

- Daniel: I need to synchronize our conclusions better since each of us worked on a different part for the first few weeks. I'll have to compile all of that together for next week in our checkpoint report as well as our final report.

- Bofu: The part I got stuck on is the part that some prompts couldn't get improved or even reasonable from the Text2SQL implementation. Questions like asking advice (e.g. should I turn off the ac in August) from chatbot would not get a correct response, unlike questions like asking detailed numerical answers (e.g. which appliance used the most energy in August).

- Phu: Since LangGraph's persistence layer and NeMo Guardrails are two distinct frameworks, having them work together in one flow/script is currently a challenge, with minor bugs; however, we are confident that it is possible to have the two frameworks work together. It is primarily a matter of understanding how the two frameworks work, especially LangGraph because most of the processing steps are "under-the-hood".

What will each group member do next week?

- Week 5 guardrail work distribution
  - Input moderation (Phu & Daniel)
    - Goal: check for prompt injections, non-canonical requests, etc.
  - RAG evaluation (Phu & Daniel)
    - Goal: fact-check the model responses with EDA findings
  - SQL-suitability guardrail (Eric & Bofu)
    - Goal: steer away from SQL-generation if the user prompt isn't suitable (prevent freak-outs)

- Final report coverage
  - Describe the data
  - Does the product work?
  - Focus on guardrails , what happens with improper inputs
  - What did we try, what worked, what are future outcomes?

- Eric: I will start working on the guardrails for SQL-suitability since we don't want to waste API calls when we know that SQL statements themselves are not sufficient to answer the question. Another thing is that I will need to work on how the Text2SQL component interacts with the rest of our final product like the persistence layer. But since there is a checkpoint coming up soon, I will help with pushing my code to the repo and reporting what we've done so far on the methods in our report checkpoint.

- Daniel: I believe that I'll work on completing the project report prior to the check in. I'll compile all our findings in our report and start developing the overall structure of our website and start filling out some details. I'll also work on developing input guardrails for our final product.

- Bofu: Based on the discussion, next week Eric and I are going to do some guardrail related work to make sure the user inputs work well with the Text2SQL part since each call of the chatbot will cost a few seconds to let the langchain execute the SQL query. We don't waste the api call from unsuccessful prompts. We are going to design a method (guardrail) to make the Text2SQL more efficient and collaborate with other functions of the chatbot.

- Phu: I will fix the bugs we are having to get NeMo and the persistence layer to work together. Afterwards, we should have a more streamlined vision of what our final deliverables, especially the Streamlit chatbot product and its structure, will look like. I will work with Daniel to bring our working script close to being a "final deliverable" and potentially integrate it with Bofu and Eric's Text2SQL workflow. Daniel and I will also work on the input moderation and RAG evaluation guardrails that are amongst the primary functions of Digital Bouncers.

As a group, do you still feel on track to execute your project proposal?

Yes, we are confident that we are on track to execute our project proposal. Eric and Bofu were able to get the Text2SQL flow successfully working, despite certain issues we saw, such as the framework "freaks out" when the user's prompt is not suitable to generate a SQL query off of, which we framed as an "opportunity" because those scenarios allow us to develop guardrails to address them. Daniel has also started on the final paper report, which is a large step forward since we are half-way through the quarter. Plus, we have a working LangGraph persistence layer from last week, which we are confident in achieving integration with NeMo Guardrails to structure our final chatbot. As of today, we have the following working: Text2SQL flow, LangGraph persistence layer, PostgreSQL data and conversation state storage; and the following are close to working: Persistent NeMo guardrails framework and robust Text2SQL workflow.

Week 3 (due Sunday, January 26th)

Outside of your meeting with your mentor, when did your group meet this week?
Time: 5 PM, Sunday, January 26th
Location: Discord
Attendees: Phu Dang, Eric Chen, Daniel Li, Bofu Zou

What did each group member do this week?

- Eric: Collaborated in a team with Bofu to tokenize the numerical dataset. Current attempt: turning each row into a sentence (turning quantitative data into qualitative) — this is because fine-tuning a pre-trained LLM on IoT / numerical data is quite unpopular and challenging, something like a book / traditional text data would be much easier.
- Daniel: Looked at Mistral and Claude, tested through the HuggingFace API. Collaborated in a team with Phu on in-prompt data provision techniques, will further try out different in-prompt data provision methods while controlling prompt / token length, reduce distracting information / hallucinations, and ultimately output accuracy (actual ground-truth room/appliance energy usage) and quality (tone, friendliness, readability).
- Bofu: Collaborated with Eric on fine-tuning. Worked on the best way to tokenize the dataset for our purposes, tried to use Google's flan-t5 model (OpenAI models were expensive). Worked on turning each row into a sentence, currently takes a long time to train (onward of 8 hours). Ongoing research has shown LLMs do not perform well with numerical data, which is a predominant challenge for us. Pending output from the training to evaluate fine-tuned model results.
- Phu: Collaborated with Daniel on in-prompt data provision techniques and implementing a persistence layer to facilitate conversations and fault tolerance with the LLM models (able to remember previous interactions, even when the kernel restarts or computer crashes). Got a persistence layer to successfully work through LangChain's LangGraph using Postgres as cloud storage for conversation "checkpoints". These checkpoints are configured through threads for conversation tracking and in LangChain's agentic workflow to connect with Postgres, recollect context from previous interactions, and store new interactions (prompt & response). Successfully implemented a Postgres database on Heroku to store and access conversation checkpoints.

Where did they get stuck?

- Eric: Ran into issues about tokens data structure (vector padding) during the tokenization process. Will experiment with other techniques from HuggingFace. Need to finalize the most optimal data structure (vector storage) for our time-series dataset.

- Daniel: Kernel kept crashing, will try Google Colab to see if it will be better. Not sure what our final product will look like (framework used — NeMo or raw API call or fine-tuned model, and format (product- or research-oriented webpage), which will inform the finalized prompt structure.

- Bofu: The runtime is currently too high, we will need a different approach to training the data. Perhaps aggregate the dataset different to retain the time-sensitive granularity, but also broad enough to capture seasonal patterns and the annual scope of our dataset.

- Phu: Figuring out the most optimal persistence implementation technique took some time, especially choosing between manually implementation and pre-built frameworks. However, it quickly became apparent that we should ideally incorporate a cloud storage framework for storage flexibility and accessibility, so we went with LangGraph's persistence framework. After the decision, it then took some time to understand LangGraph's implementation and integration with PostgreSQL, which we are glad to have figured out and currently have a working notebook. It was also helpful that we have an existing PostgreSQL subscription through Heroku, which made spinning up a Postgres instance very easy.

What will each group member do next week?

- Eric: This upcoming week will be the second and final week that I will be working on fine tuning. The data has been preprocessed, and I hope that my notebooks will be able to continue training the models without running into issues with Colab preemptively taking away the GPU. It will also be up to Bofu and I to evaluate how the model will respond to the prompts that Daniel and Phu will give us.

- Daniel: So for next week, I'll refocus my efforts more on fine tuning the large language model. As I mentioned early, I have different prompt data provision techniques that I like currently with gpt 4; the main issue is that I don't know whether or not I will like these provision techniques right now. As such, I will redirect my focus to support the fine tuning group, to better streamline the process so we can reach the final product faster.

- Bofu: Will look for better approaches to aggregate our data for faster training and a more efficient data structure (for ex: our demo fine-tune notebook from our summer pre-work uses DatasetDict as the train and evaluation set storage structure), we will look at ChromaDB and other similar structures meant for vector/embeddings storage for the most efficient training runtime.

- Phu: Will need to incorporate a storage limit to manage the context window from which the LLM recollects prior conversation contexts; this will be crucial to manage distractions and ultimately, hallucinations, for our models; while keeping the size of checkpoints store in check, which relates to cost and operation constraints in the real-world / production. Will work with the rest of the team to finalize the final product picture to start developing more refined components to comprise that "picture".

As a group, do you still feel on track to execute your project proposal?
Yes. We are confident that we are on track given our successful persistence layer implementation and in-prompt data provision techniques with good output quality, this was the initial vision of our tool. The fine-tuning component is experiencing tokenization and runtime issues; however, these were expected and we are content with the prospects of fine-tuning not end up working at all, given that LLMs are known to be inadequate at processing numerical / time-series data from our preliminary research and ultimately, the fine-tuning component could be a research portion of our final deliverable as we attempted something challenging and novel, so no work is going to be wasted. Regarding the RAG components (bill explanation and energy prices) and guardrails, we are confident at tackling them after our shots at fine-tuning and persistence because we have defined the exact approaches we want to take. For example, in-prompt context provision for RAG, as Zayd, co-founder of Guardrails AI, has shown us, which eliminates the hassles of tokenization, vector storage, and similarity search. Besides, we each have experience from last quarter regarding guardrail implementation for all three areas (input moderation, RAG evaluation, and output moderation) and have the code to implement them — so, we are confident that we are on track to execute our project proposal.

Week 2 (due Sunday, January 19th)

Outside of your meeting with your mentor, when did your group meet this week?
Time: 5 PM, Saturday, January 18th
Location: Zoom
Attendees: Phu Dang, Eric Chen, Daniel Li, Bofu Zou

What did each group member do this week?

- Eric: During this week, I looked into the appliances in our smart home dataset, specifically the fridge, microwave, and dishwasher. I crafted 5 prompts and expected answers based on the ground truth data, and found out that microwaves and dishwashers have very similar monthly energy consumption patterns, whereas the fridge has a more seasonal curve to it (energy consumption is highest during the summer).

- Daniel: So this week, I focused on EDA in terms of the weather with respect to energy usage, as well as came up with some prompts related to those parameters. The main thing that I found was that most of the weather metrics had barely any correlation to energy consumption, if at all. The only two metrics that had any real correlation was temperature and humidity and those barely had much correlation.

- Bofu: During this week, I did a data analysis on the location part of the dataset, focusing on the trend and ground truth statistics of each location's contribution to the energy consumption. From the analysis, I found that the home office has consumed the highest amount of energy in 11 out of 12 months in a year, following with the barn. The change doesn't really follow the pattern of time. I tried to find the correlation between each location's energy use and temperature, but the correlation is relatively little.

- Phu: I worked on further experimenting with ways to effectively get the LLM to understand our tabular IoT data without using excessive tokens per prompt, as cost may be incurred fast. Since our smart home dataset contains timestamps in 1-minute intervals, the granularity of our data means we cannot include the entire dataset in our prompt, so I experimented with different aggregation methods. Besides a monthly group-by, I tried incorporating trends and summary statistics for each location and appliance in the home, which showed little improvements from our initial tests. I also set up our NeMo guardrails framework, the configuration files, and walked my teammates over the NeMo framework architecture for those who did not use NeMo in their individual projects last quarter. To incorporate energy prices data later, I picked a location in the U.S. that could have the sort of temperature range from our dataset, which is Denver, Colorado.

Where did they get stuck?

- Eric: Our development strategy for creating the prompts involved a step in which we were supposed to create a fact check guardrail. I wasn't too familiar with this since it was framed in the context of NeMo guardrails which I'm not very familiar with. Therefore, a lot of those sections were just me talking about how I would tell the guardrail to fact check the expected answer.

- Daniel: As I mentioned above, weather didn't have any real correlation to energy consumption, so I had to really dig deep to figure out if there was any real relationship between the two. Luckily, there were seasonal energy consumption changes that were shown in the weather fluctuations, so I went off of that.

- Bofu: I tried to dig out useful information and statistics about location to help build the chatbot, but it is hard to find out some patterns. I also have some confusion about the development strategy, more specifically, to come up with testing scenarios, but eventually I worked that out after talking with Eric and Phu. I am not really familiar with fine-tuning LLM, so it would be a challenge next week.

- Phu: I got stuck for a while at envisioning the holistic picture of our project and what our final deliverable will look like beyond the course requirements (paper, website, poster). We named our tool "Digital Bouncers" because we were excited at the idea of creating secure gatekeepers of digital interactions, a metaphor that allows our audience to relate an image of strength and controllability over their data. Like a bouncer at a club, our tool ensures that only safe, appropriate, and necessary data is allowed in or out. It filters and moderates inputs (user queries) and outputs (chatbot responses), blocking anything that doesn't comply with privacy, security, or appropriateness guidelines. However, perhaps because we were pushed to narrow the scope of our ideas, which we appreciate for easy planning and execution in the past 1-2 weeks, we inadvertently diverted from the security component and "bouncers" theme, which is critical for creating use cases for guardrails. For example, our two specific use cases: energy use summary creation (Spotify-wrapped style) and utility bill explanation, are too narrow for implementing guardrails as there are seldom opportunities for leakage of private information, which is the key demand driver for guardrails today, considering the common queries a typical homeowner may ask and the responses they receive (e.g., not many people would include their address when asking about a charge on their bill). Of course, a scenario we thought about is one of the homeowner's children may mess around, in typical children fashion, and ask our tool to say something inappropriate out of curiosity to see the tool's response; which would be a use case for guardrails, however, such scenarios are not compelling as they are rare. Nonetheless, exciting opportunities remain with RAG evaluation and output moderation guardrails (fact- and appropriateness-checking). We are confident that our project is further "crystallizing" every week and we will figure out compelling use cases to have a novel and exciting project to serve our initial inspiration of *protecting and moderating digital interactions* with digital bouncers.

What will each group member do next week?

- Eric: Along with many of the other members of my team, we are moving forward with training and LLM on the smart home dataset instead of uploading raw data each time. More specifically, I'm going to figure out how to tokenize the tabular dataset so that it is ingestible for an LLM.
- Daniel: Next week, I will be experimenting with fine tuning the RAG with our specific dataset. I'll also get started on thinking about the streamlit app, and how we want to summarize the energy consumption breakdown for our app.
- Bofu: Based on the discussion this week, I think each of us would do some experiments with fine-tuning the LLM for our chatbot based on the dataset we have studied. It is

also set to be a task that we need to tokenize the dataset. It is still on the agenda that I need to start working on the RAG development for the upcoming weeks.

- Phu: We will experiment with fine-tuning a pre-trained model, which I reckon the tokenization process will be challenging for our numeric IoT time-series dataset as LLMs are best for text data, given that they operate by predicting the next token/sequence of text. However, we believe it's a worthwhile experiment due to its unpopular nature and time-series data can be reformatted into sentences. For example: the following csv string "2025-1-19:17:15,0.09,0.01,0.05" with columns "timestamp,fridge,microwave,dishwasher" can be written as "The fridge used 0.09 kW, the microwave used 0.01 kW, and the dishwasher used 0.05 kW of energy at 17:15 on January 19, 2025". As I was writing this, I thought of splitting our management into two approaches to give the best shot at building a framework that can process time-series data reasonably well, so we might have two folks focusing on "fine-tuning" a pre-trained model, which we are unsure if the customized model would be compatible with NeMo, and the other two folks will focus on "in-prompt data provision" (as we have been experimenting) with LangGraph's built-in persistence layer to facilitate conversations.

  This will help "diversify" our approaches while keeping our "narrow" purpose of "processing time-series data."

  I will bring this thought back to my team.

As a group, do you still feel on track to execute your project proposal?
Yes. Despite the challenges above, we're confident that we're heading in the right direction and will create a novel and compelling case study of how LLMs handle time-series data with guardrails in the flow.

Schedule progression assessment — please see our updated schedule below:

Adjusted schedule (Spring 2025)

- Week 1 (DONE)
- Week 2 (DONE)
  - test prompts created
  - RAG documents identified (include in prompts directly (Zayd's demo))
  - final Streamlit chatbot deliverable pictured

    - attempted in-prompt data feed → has a sense of what works and doesn’t
    - Settled on NeMo as the primary framework (covered configuration files architecture as a team)

- Week 3 (Jan 18 - 24)
    - - Experiment with fine-tuning (https://github.com/pndang/llm-comet/blob/main/fine_tuning.ipynb)
    - - Identify pre-trained model starting checkpoint (e.g., flan-t5-base)
    - - Use AutoTokenizer + Seq2SeqLM trainer for fine-tuning
    - - Use either Comet (already has working demo notebook) or TruLens for model registration/storage
    - - Steps: tokenize our tabular dataset, store in vectors, split into train and eval chunks, and train

- Week 4 (Jan 27 - Feb 1)
    - Begin integration into final framework (NeMo)
    - Begin RAG
    - Start testing input/output moderation and RAG eval test prompts

- Week 5 (Feb 3 - 7)
    - List key criteria for work distribution to begin drafting report materials
    - Split into poster- and web-team
    - Start drafting reports

- Week 6 (Feb 10 - 16)
    - Work on LaTex paper as a team

- Week 7 and onwards (contingency for progress overruns)

Week 1 (due Sunday, January 12th)

Outside of your meeting with your mentor, when did your group meet this week?
Time: 5 PM, Saturday, January 11th
Location: Zoom
Attendees: Phu Dang, Eric Chen, Daniel Li, Bofu Zou

What did each group member do this week? Where did they get stuck?

- Daniel: This week was mostly the logistic side of things. I went over the dataset very briefly to get a feel for it, and also had a better understanding of my specific role for the project. We talked about the general structure of our project and also split up the specific tasks that we'd each be working on.
- Eric: Since we mostly reviewed things we needed to work on this week, I helped with narrowing down to the scope alongside my other teammates before our first meeting this quarter. In our other meeting, we solidified a good plan of our project as well as who would be responsible for different parts of the data we are using for the project.
- Phu: I spent time exploring the dataset to get a good grasp of what's available for us to develop specific use cases and test scenarios (guardrails) for. My EDA work comprises of graphs, summary statistics, figuring out what some ambiguous columns might mean, for example, we just found out that column "furnace" is the sum energy consumption for both furnaces in the subject house (labeled "furnace_1" and "furnace_2" — I think little insight nuggets like this are crucial for us to effectively and accurately work with the data we have. I also tested out how well we could get the LLM to digest and process tabular data, which we are keeping track of good prompts + responses to potentially use as one-shot and/or few-shot context learning later. Our current observations are heading in the right direction.
- Bofu: For this week, in general we discussed the content and workflow of our project and set up a plan throughout the entire quarter. I downloaded the dataset and got to know the basic structure and figures of each column and picked my job in the team plan.

What will each group member do next week?

- Daniel: I will be doing some data analysis on the effects of weather on the electricity bill. I will also be working with Bofu on the functionality of delivering the summary on electric usage. In terms of the guardrails, I will explore different options within Nemo guardrails to see which ones we could apply to our LLM. Finally, since my ESA is on weather, I will be coming up with weather related test prompts to see the performance of our LLM on that.

- Eric: My main task is to understand appliance data in the smart home dataset, particularly trends and ground truth statistics, and ideate specific prompts and expected (correct) responses. But I also will be working with Phu on finding utility prices for our dataset because alongside energy savings, we want to think about how we can lower costs for people since this is about smart homes and how we can build tools for a better future.

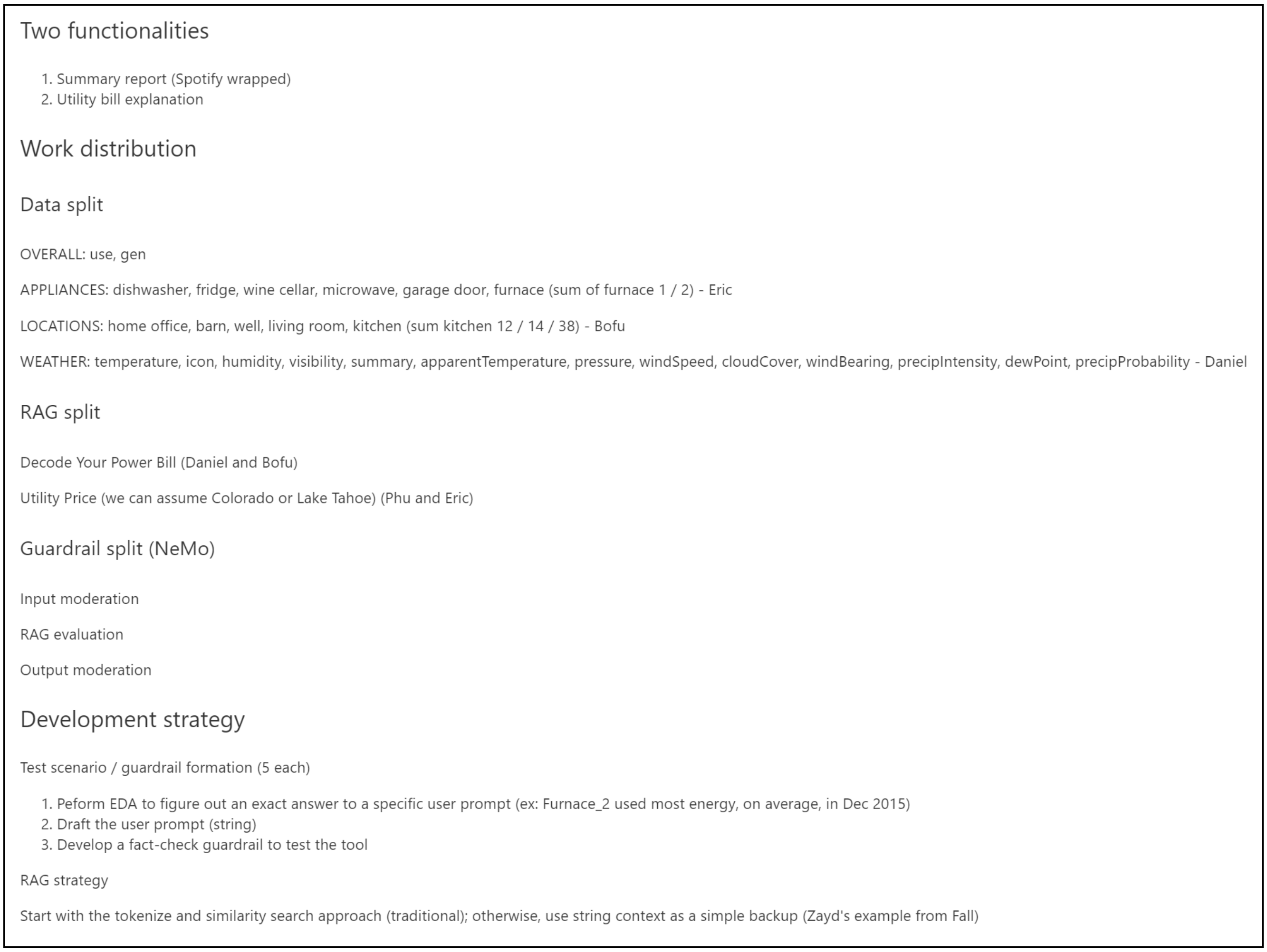
- Phu: I will conduct prompt engineering to best instruct the LLM in understanding its role as a smart home data assistant, this would be the "system" content prompt portion of an OpenAI API call, or the role prompting in NeMo guardrails' Colang configuration files. The other end would be response formatting; since this project was inspired to be an AI smart home product proof-of-concept, I will include instructions for the LLM to return user-friendly responses to keep length, wording, conversational flow in check for the best user experience.

- Bofu: My task for next week is about doing analysis on the location part of the dataset. I will analyze how each location contributes to the power bill and how each location's electricity usage will vary with other variables (like weather). I will also find out the difference between each location and how the difference makes changes on the bill. For the RAG split, I will cooperate with Daniel to work through decoding the bill.

As a group, do you still feel on track to execute your project proposal?

Yes, we are confident that we are on track to execute this project, largely thanks to the "specificity" that our mentors pushed for us to achieve in our proposal, which helped us work very conveniently by thinking of concrete tasks and goals to achieve each week, at least in Week 1 for now, we have a clear vision of the final product, what to be done, and work distribution. We think one of the most productive plans that could turn out to be very beneficial to us by the end of this quarter is "scoping" (screenshot below). For example, we have decided on two specific RAG documents, serving two specific functionalities, that is straightforward to develop guardrails / test prompts for. And to facilitate this testing, we developed a concise 3-step process to determine the ground truth to be tested against through EDA, then one specific prompt / question a user may ask, and measure accuracy as components of a fact-check guardrail framework.

Screenshot from explore.ipynb



Two functionalities

1. Summary report (Spotify wrapped)
2. Utility bill explanation

Work distribution

Data split

OVERALL: use, gen

APPLIANCES: dishwasher, fridge, wine cellar, microwave, garage door, furnace (sum of furnace 1 / 2) - Eric

LOCATIONS: home office, barn, well, living room, kitchen (sum kitchen 12 / 14 / 38) - Bofu

WEATHER: temperature, icon, humidity, visibility, summary, apparentTemperature, pressure, windSpeed, cloudCover, windBearing, precipIntensity, dewPoint, precipProbability - Daniel

RAG split

Decode Your Power Bill (Daniel and Bofu)

Utility Price (we can assume Colorado or Lake Tahoe) (Phu and Eric)

Guardrail split (NeMo)

Input moderation

RAG evaluation

Output moderation

Development strategy

Test scenario / guardrail formation (5 each)

1. Peform EDA to figure out an exact answer to a specific user prompt (ex: Furnace_2 used most energy, on average, in Dec 2015)
2. Draft the user prompt (string)
3. Develop a fact-check guardrail to test the tool

RAG strategy

Start with the tokenize and similarity search approach (traditional); otherwise, use string context as a simple backup (Zayd's example from Fall)