Using the Gemini LLM API

Recitation 5 — 6.1040 Fall 2025

Prep 2 Review

1. **Redeem Google Cloud Credits (*canvas/piazza link here)**
2. **Make a Gemini API key for Gemini on Google Cloud**
3. **Check Gemini API key works with Scheduler app**

Any issues with any of these things?

Prep 2 Code

Concept Design

purpose Help you organize activities for a single day

principle You can add activities one at a time, assign them to times, and then observe the completed schedule

state

a set of Activities with

a title String

a duration Number

a set of Assignments with

An activity Activity

a startTime Number or Undefined

a TimeSystem

actions

addActivity(title: string, duration: number): Activity

removeActivity(activity: Activity)

assignActivity(activity: Activity, startTime: number)

unassignActivity(activity: Activity)

requestAssignmentsFromLLM()

https://github.com/61040-fa25/intro-gemini-schedule

Prep 2 Code

Concept Design

```
Concept DayPlanner

purpose Help you organize activities for a single day

principle You can add activities one at a time, assign them to times, and then observe
the completed schedule

state                                    actions
  a set of Activities with                 addActivity(title: string, duration: number): Activity
    a title String                         removeActivity(activity: Activity)
    a duration Number                      assignActivity(activity: Activity, startTime: number)
    a startTime Number or Undefined
                                           unassignActivity(activity: Activity)
  a set of Assignments with
    a set of Activities to Times           requestAssignmentsFromLLM()

  a TimeSystem
```

How do you see the LLM impacting the *purpose* and *principle* of this concept?

How may the LLM potentially impact the *invariants* of the concept?

How may you change the concept to allow for *other* LLM augmentations?

https://github.com/61040-fa25/intro-gemini-schedule

Prep 2 Code - LLM-Augmented Scheduler

How may you change the concept to allow for *other* LLM augmentations?

purpose Help you organize activities for a single day

principle You can add activities one at a time, assign them to times, and then observe the completed schedule

a set of Activities with
- a title String
- a duration Number

a set of Assignments with
- An activity Activity
- a startTime Number or Undefined

a TimeSystem

actions
- addActivity(title: string, duration: number): Activity
- removeActivity(activity: Activity)
- assignActivity(activity: Activity, startTime: number)
- unassignActivity(activity: Activity)
- requestAssignmentsFromLLM()

State Changes?

Ideas here…

Action Changes?

Ideas here…

https://github.com/61040-fa25/intro-gemini-schedule

What does the LLM *currently* do with our Scheduler?

```
requestAssignmentFromLLM()
// given a set of activities (which include durations and possible start times), will assign activities into an organized schedule
```

```
{
  “assignments”: [
    {
      “title”: “Gym Session”,
      “startTime”: 27, // 1:30pm, duration = 1 hour
    },
    {
      “title”: “Math Homework”,
      “startTime”: 32, // 4:00pm, duration = 2 hours
    },
    {
      “title”: “Project Work”,
      “startTime”: 35, // 5:30pm, duration = 1.5 hours
    }
  ]
}
```

Example LLM output… Do you see any problems with this?

What would you do to prevent the LLM from making mistakes like this?

What else can the LLM do with our Scheduler?

How may you change the concept to allow for *other* LLM augmentations?

purpose Help you organize activities for a single day

principle You can add activities one at a time, assign them to times, and then observe the completed schedule

a set of Activities with
- a title String
- a duration Number

a set of Assignments with
- An activity Activity
- a startTime Number or Undefined

a TimeSystem

actions
- addActivity(title: string, duration: number): Activity
- removeActivity(activity: Activity)
- assignActivity(activity: Activity, startTime: number)
- unassignActivity(activity: Activity)
- requestAssignmentsFromLLM()

State Changes?

Ideas here…

Action Changes?

Ideas here…

Also consider LLM error prevention, including prompt injection

What else can the LLM do with our Scheduler?

How may you change the concept to allow for *other* LLM augmentations?

purpose Help you organize activities for a single day

principle You can add activities one at a time, assign them to times, and then observe the completed schedule

state

a set of Activities with

a title String

a duration Number

a set of Assignments with

An activity Activity

a startTime Number or Undefined

a TimeSystem

actions

addActivity(title: string, duration: number): Activity

removeActivity(activity: Activity)

assignActivity(activity: Activity, startTime: number)

unassignActivity(activity: Activity)

requestAssignmentsFromLLM()

State Changes?

Try modifying the concept to incorporate "importance" for each activity assignment — how does that change how your LLM integration should work?

Action Changes?

Show only high-importance activity assignments?

What else can the LLM do with our Scheduler?

```
state
    a set of Activity with
        a title String
        a duration Number // in half-hour units, so 3 is 90 mins
        an optional startTime Number // in half-hour slots from midnight, so 14 is 7:00am
    a set of Assignment with
        an Activity
        an startTime Number
        an importanceWeight Number // 0 -- not at all important <> 5 -- very important
```

Concept change

```
STUDENT PREFERENCES:
- Exercise activities work well in the morning (6:00 AM - 10:00 AM)
- Classes and study time should be scheduled during focused hours (9:00 AM - 5:00 PM)
- Meals should be at regular intervals (breakfast 7-9 AM, lunch 12-1 PM, dinner 6-8 PM)
- Social activities and relaxation are good for evenings (6:00 PM - 10:00 PM)
- Avoid scheduling demanding activities too late at night (after 10:00 PM)
- Leave buffer time between different types of activities
- Prioritize high-importance activities earlier in the day when possible
- Important activities have an importance weight from 0 (not important) to 5 (very important), where higher importance activities sho
- Students typically consider activities as more important when they are directly related to their goals (e.g., studying for an exam
```

Prompt change

```
Return your response as a JSON object with this exact structure:
{
  "assignments": [
    {
      "title": "exact activity title from the list above",
      "startTime": valid_slot_number_0_to_47,
      "importanceWeight": importance_weight_number_0_to_5
    }
  ]
}

Return ONLY the JSON object, no additional text.`;
```