PlanIt! - Final Report

Dechen Han, Ginny Ji, Jason Pu, Kexin Sha

Word count: 1992/2000

Penalty: 0%

(including text in the table, excluding cover page, permissions, screenshots)

1. Introduction

There is a growing demand for AI-generated trip itineraries. Imagine planning a three-day trip to Rome with a budget and preferences but feeling unsure about how to spend your time. While GPT-4 can generate plans based on inputs, they often lack realism. Our project addresses this by creating realistic, user-aligned trip plans using a multi-agent LLM system.

LLM-powered systems are ideal for this task, excelling at processing complex inputs, generating natural language outputs, and iteratively refining solutions to align better with user needs. Our system uses three agents: one selects a plan from a database based on user inputs, another refines it for better realism and alignment through prompt engineering, and the third verifies if the final plan meets user requirements. The plans focus on food, accommodations, and exploration within the budget and preferences, excluding transportation.

To measure effectiveness, we compare our results to a baseline model using GPT-4, evaluating how refinements improve realism and alignment with user expectations.

2. Background & Related Work

Some existing tools and research are closely related to our project. For example, TripIt is an itinerary management app that consolidates travel plans from email bookings and organizes them into structured schedules [1]. Unlike TripIt, which focuses on organizing pre-existing travel details, our project aims to create entirely new travel plans. However, generating complex plans directly with LLMs presents significant challenges. Hao et al. demonstrate that while LLMs excel at generating initial travel plans from general queries, these plans often lack rigorous verification and miss critical details, resulting in unreliable outcomes. Additionally, verifying the feasibility of these plans poses another challenge, leading to inconsistent solutions [2]. To address these issues, our approach refines and customizes existing travel plans from our database rather than generating them entirely from scratch, ensuring accuracy and reliability.

3. Data

We collected a dataset of 256 unique travel plans to be used by the Selection Agent with the RAG model. Specifically, We gathered 80 examples from popular travel websites like Expedia and TripAdvisor, generated 160 additional examples using GPT-4, and added 16 manually created examples. This ensured the data was diverse and sufficient for our system to process effectively.

Since the itineraries were curated from diverse sources, we converted each itinerary into a structured format and ensured consistent text formatting. Further, we manually deduplicated the dataset and tagged each itinerary with the following labels: country, destination, duration, budget and points of interest.

After cleaning, we organized the results into a structured CSV format. Each data entry includes labeled key factors with a detailed itinerary. This structured format allows for efficient filtering and analysis by our different agents. One data example is shown in **Table 1**.

Country	Destination	Duration	Points of Interest	Budget	Detail
Japan	Kyoto	4	8	400	Day 1: Arrive in Kyoto and city tour; Day 2: …

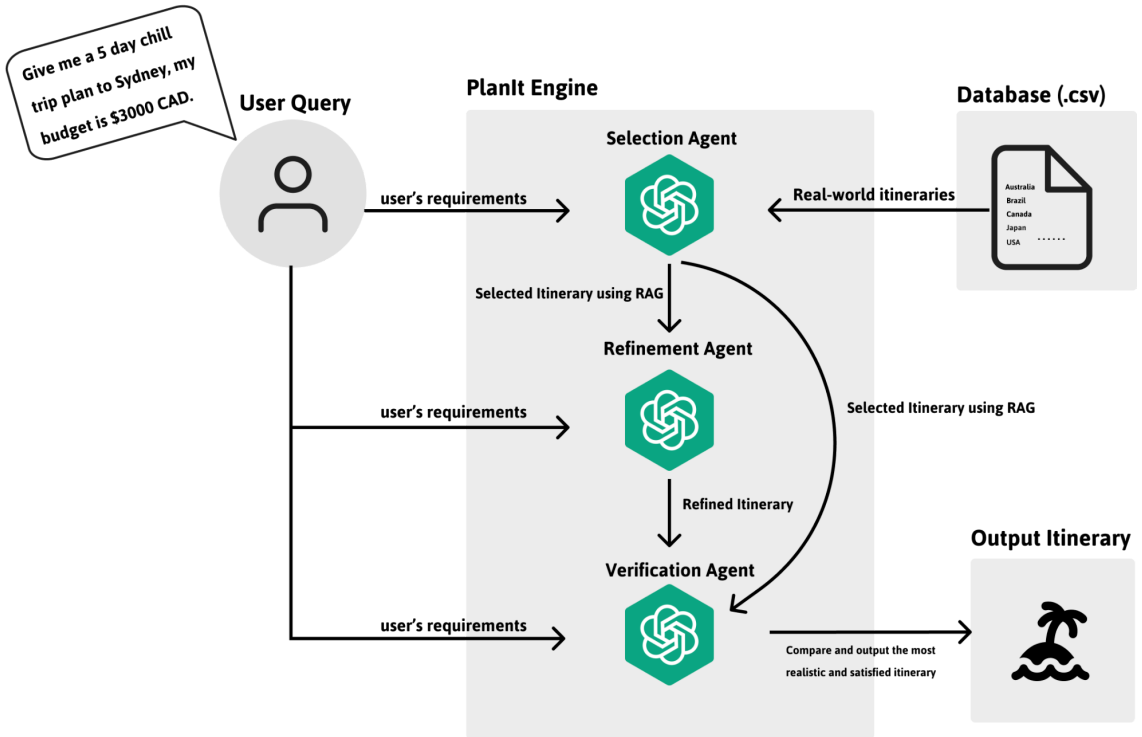
Table 3.1. Example of a CSV Data Entry

Similarly, our result data is structured in a JSON file for easy access within the system and presented to the user. For clarity, an example result has been converted into **Table 2**.

user_query	"I want a 7-day trip to China with a budget of 500”
selected_trip	“Day 1: Visit Tiananmen Square and …”
refined_trip	“Day 1: Explore Beijing's traditional …”
selected_verification	"User Satisfaction Rate: 50% …”
refined_verification	"User Satisfaction Rate: 90% …”

Table 3.2. Example of a JSON Result

4. Illustration



5. Architecture and Software

Our system is structured as a pipeline incorporating three GPT-4o-based agents: the Selection Agent, Refinement Agent, and Verification Agent.

The process begins with the **Selection Agent** using RAG to enhance accuracy by retrieving the most relevant real travel plans from a pre-existing database instead of generating content from scratch. Key details such as destination, duration, and budget, are extracted from the user query, using GPT-4o and converted into vector embeddings. Similarly, travel plans in the database are also embedded. The system identifies the top five matching plans by comparing embeddings. GPT-4o then evaluates these five plans against the full user query to determine their alignment with the user's needs and selects the most suitable plan for further refinement.

Next, the **Refinement Agent** tailors the selected travel plan to better meet the user's specific requirements. This agent adjusts details such as budget constraints, preferred activities, and travel logistics to ensure the itinerary is customized and user-centered. Once refined, the itinerary proceeds to the Verification Agent for evaluation.

The **Verification Agent** evaluates both the initially selected itinerary and the refined version for quality and feasibility. It scores these itineraries using two metrics: the User Satisfaction Rate, which measures how well the itinerary matches the user's preferences, and the Trip Realism Score, which assesses the plan's feasibility considering budget, time, and logistical constraints. A weighted calculation of these scores determines the most suitable itinerary to present to the user.

6. Baseline Model and Single-Prompt Comparison

To evaluate the effectiveness of our system, we established two key metrics: User Satisfaction Rate and Trip Realism Score (defined in Section 5). User Satisfaction Rate is calculated using an well-engineered gpt prompt to evaluate the percentage of user requirements (e.g., budget, preferences, activity types) satisfied by the itinerary. Trip Realism Score is assigned by another well-engineered gpt prompt using structured rubric, where itineraries are evaluated against factors such as practicality, logical scheduling, and adherence to real-world constraints. These scores are assigned by our Verification Agent, a specialized component designed to analyze itineraries and provide consistent evaluations based on these metrics. Please refer to Appendix A for system prompts of these evaluator agents.

As a baseline, we use GPT-4 to generate itineraries directly from user inputs - one single Prompt (Appendix B). To validate the Verification Agent's reliability in scoring realism, we cross-verified its outputs with a human evaluator using the same rubric. This ensures the agent consistently reflects human judgment of itinerary feasibility.

7. Quantitative Results

To measure the system's performance quantitatively, we proposed a weighted score defined as: Weighted Score = 0.8 * User Satisfaction Rate + 0.2 * Realism Score. We believe that placing greater emphasis on the user satisfaction rate is appropriate because ensuring that the refined plan aligns with users' requirements is more critical than verifying its feasibility. Another reason for assigning the weights in this manner is that the user satisfaction rate can be measured precisely by checking how many requirements are actually met, whereas the realism score involves some degree of subjective judgment. To minimize the influence of such subjectivity, we assigned the realism score a lower weight. We illustrate randomly selected plans' user satisfaction rate and realism score separately in the table in Appendix C.

The figure below compares the average scores of three plan types (refined plans, selected plans, and baseline plans) across 20 randomly generated user queries. On average, the refined plan achieved the highest score, surpassing the baseline model's plan by 10.8 points and outperforming the selected plan by 22 points.

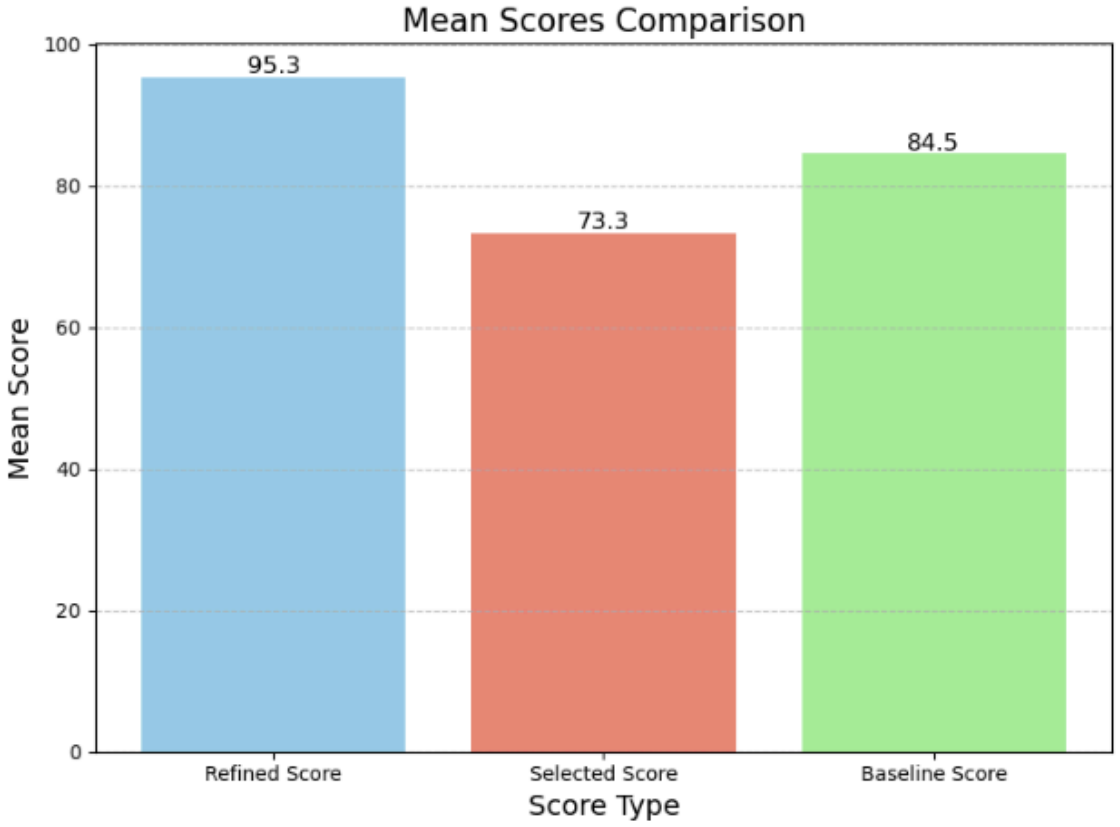


Figure 7.1. Mean Score Comparison

Further, we manually calculated the primary component of the weighted score — User Satisfaction Rate — and then compared it to the score (a percentage) determined by the verification agent. This step is crucial because the User Satisfaction Rate accounts for 80% of the total weight, making its accuracy a strong indicator of the system's overall performance. Further, GPT-4o is known to perform poorly in arithmetic, and we wanted to ensure it correctly calculated the percentages. So, we manually verified the calculated scores. In a

random check, we confirmed that 4 out of 4 manually calculated scores precisely matched those produced by the verification agent.

8. Qualitative Results

After understanding the quantitative methodology for measuring the system's success, it is equally important to assess the results qualitatively. In this section, we illustrate instances in which the system performs well and cases in which it fails. The image below shows a successful outcome based on a specific user query. In this example, the query specifies a 5-day trip to Thailand, a budget of $1,000, and an interest in trying local foods and spending time at the beach. The system provides a summary of the recommended itinerary, proposing a 5-day trip to Bangkok and Pattaya within the given budget. These key details fully align with the user's requirements. In the following section, the system generates detailed daily plans. We consider these plans reasonable, as they closely reflect the user's needs. For example, on Day 1, the system identifies a location where the user can enjoy street food—one of their stated interests. On Day 3, it arranges a trip to Pattaya and suggests a beach for relaxation, further demonstrating its responsiveness to the user's preferences.

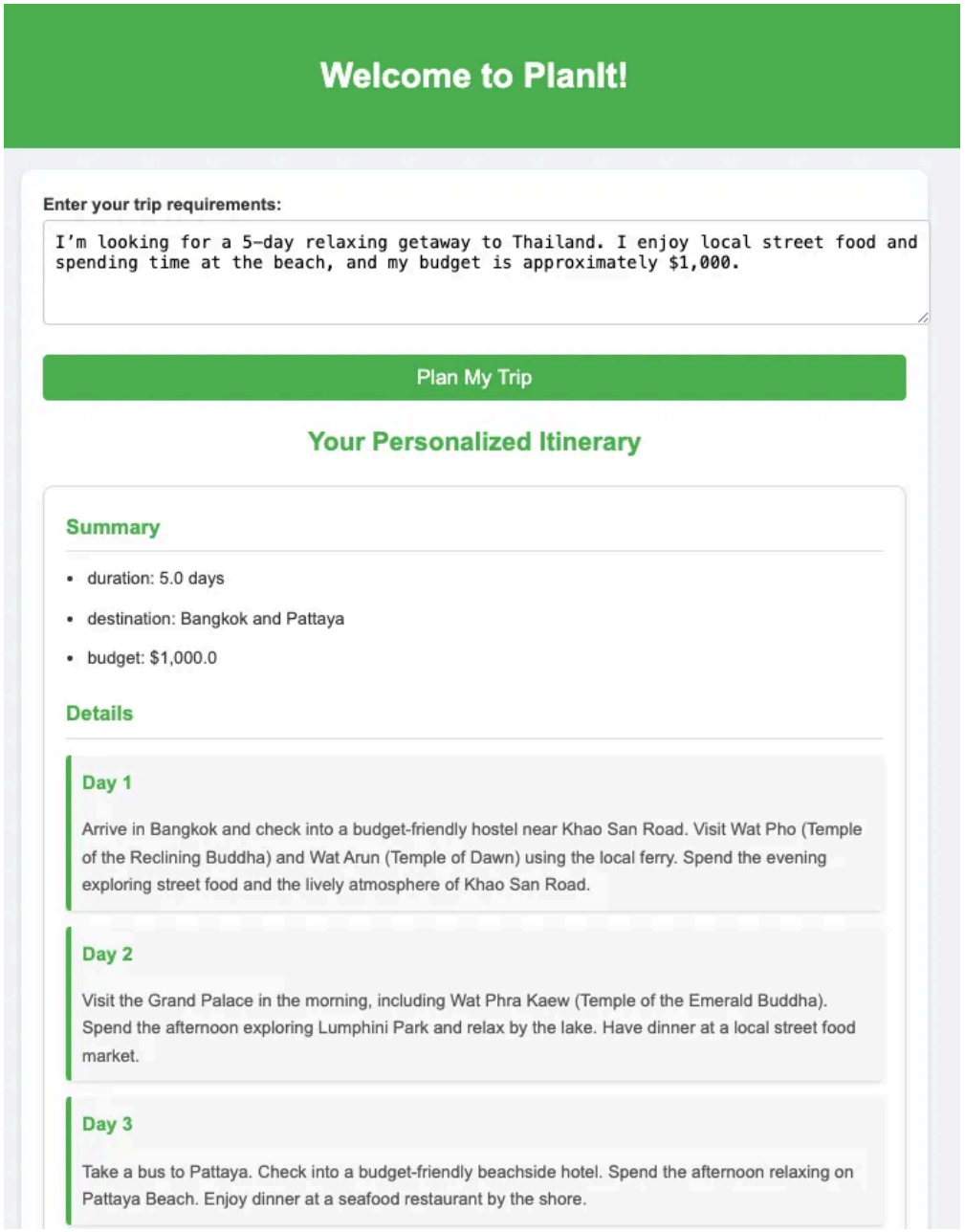


Figure 8.1. A successful example

Note: This is a partial screenshot of the trip plan as the output is too long to be captured completely, there are in total 5 days with detailed plans.

On the other hand, the system can also produce recommendations that fail to meet user expectations. For instance, given the query, "I want a 17-day trip to Norway with a budget of $4,586," the system generated a 15-day itinerary with a score of 58, which is lower than the baseline model's score, as shown below. While we observed a significant improvement in feasibility, the system did not satisfy the user's requirement for a 17-day trip. As a result, the low user satisfaction rate led to a lower weighted score overall. Even though the refined plan has a lower score, the refined plan still proposes a better trip plan with a significant amount of details than the baseline model. For example, the baseline model recommends that "*Day 1:*

Arrive in Oslo. Days 2-3: Drive to Lillehammer and explore the town…" but the refined plan mentions that "*Day 1: Oslo attractions, including Munch Museum. Day 2: Train to Bergen and visit Fish Market. Day 3: Explore Bergen, including Bryggen district and Mount Fløyen via funicular…*". The refined plan lays out the plan for each day with attractions to visit. In contrast, the baseline model only mentions fewer activities and sometimes combines plans for more than 1 day, making it less helpful and attractive for users. Ultimately, the refined plan can still be valuable even though the score may indicate otherwise.

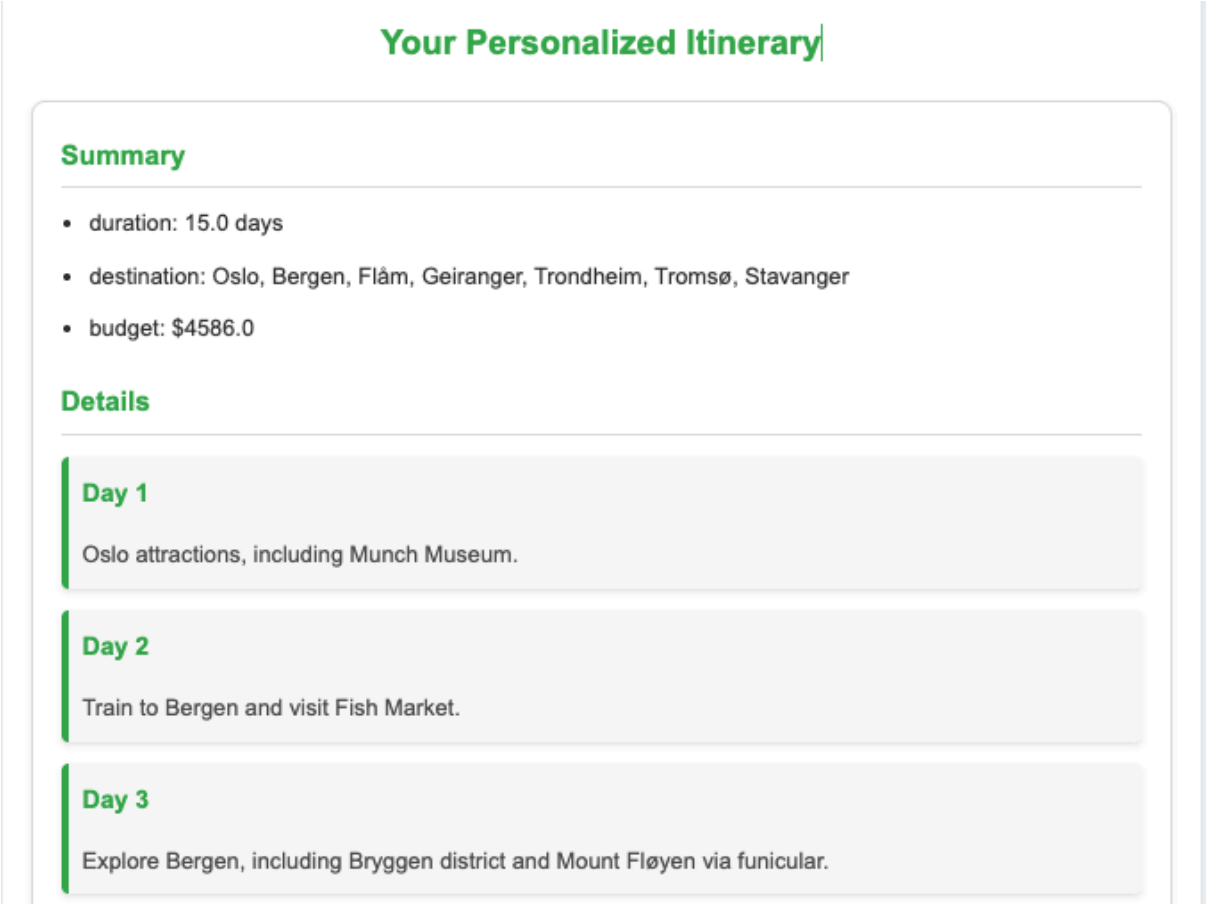


Figure 8.1. A failed example

Note: This is a partial screenshot of the trip plan as the output is too long to be captured completely, there are in total 15 days with detailed plans.

9. Discussion and Learning

Our results align intuitively, showing that the refinement agent improves plan alignment with user requests in most cases. However, we observed surprising outcomes and failure cases.

user query	refined score	selected score	baseline score
1. I want a 7-day trip to China with the budget of $5000.	98.0	65.0	26.0

2. I want a 17-day trip to Norway with the budget of $4586.	58.0	94.0	98.0

Table 9.1. Comparison of Example Queries

In the first query, the refinement agent excelled. The baseline plan scored only 26, failing to meet key criteria like duration. The refined plan scored 98 by adjusting transportation and balancing activities, delivering a near-perfect match.

In the second case, the refined plan scored 58 compared to the baseline's 98. The agent shortened the trip to 15 days to fit the budget, improving feasibility but deviating from the user's duration preference. This highlights a trade-off: optimizing realism while respecting core user requirements.

10. Team Work and Progress

Our team is collaborating effectively, consistently meeting our earlier deadlines through regular working sessions and meetings.

Contributor	Specific Contribution	Progress Report Sections	Final Report Sections
Jason	● Data collection and cleaning ● Selection Agent	3	2, 5
Kexin	● Data collection and cleaning ● Refinement Agent ● Build baseline one prompt GPT4 ● Generate & visualize 20 user queries and relevant results ● Evaluate system performance ● Manually verified system generated scores	1, 2, 8	7, 8
Dechen	● Data collection and cleaning ● Contribute to building Refinement Agent ● Build the main pipeline to connect all agents ● Add front-end demo interface ● Code clean & refactor	4, 5	3, 4

Ginny	• Data collection and cleaning • Verification agent • Implemented APIs for communication among agents • Visualization of system performance • Manually verified system generated scores	4, 6, 7	1, 6, 9

11. Reference

[1] Tripit - highest-rated trip planner and flight tracker, https://www.tripit.com/web (accessed Dec. 10, 2024).

[2] Y. Hao, Y. Chen, Y. Zhang, and C. Fan, “Large Language Models Can Solve Real-World Planning Rigorously with Formal Verification Tools,” 2024, doi: 10.48550/arxiv.2404.11891.

12. Appendix

Appendix A: System Prompts

A.1: System Prompt for Selection Agent

```
"You are a travel assistant. Extract the user's preferences from
the following query: '{user_input}'. Return a Python dictionary
with keys: 'country', 'destination', 'budget', 'duration', and
'points_of_interest'. country should be which country the user
wants to go, destination should be what area/city/attraction the
user wants to go, budget should be how much money the user wants
to spend, durations should be how long the user want to travel,
points of interest should be how many tourist attraction the user
want to go.

Output must have the same format like this and wrapped in braces
without anything else: 'country': 'Mexico', 'destination': None,
'budget': 2500, 'duration': 8, 'points_of_interest': 'natural
sites' Use None for missing information and ensure the output is
in one line."
```

A.2: System Prompt for Refinement Agent

```
"""
    You are the Refinement Agent in a multi-agent trip planning
system. Your primary goal is to adjust and enhance a trip plan
provided to you to align it precisely with the user's specific
requirements. These requirements include budget, duration, density
of points of interest, and destination preferences.

    Your goals and guidelines:

    1. Interpret User Requirements: Review the user's
requirements, paying special attention to:
        - Budget: Ensure the trip plan fits within the specified
budget, making adjustments to hotel, transport, and activity costs
as necessary.
        - Duration: Adjust the trip plan to match the specified
travel duration, adding or removing activities as needed.
        - Density of Points of Interest: Modify the schedule based
on the desired pace--either adding more points of interest for a
high-density plan or allowing more downtime for a relaxed
itinerary.
        - Destination Preferences: Ensure the trip aligns with any
specific destinations or regions the user wishes to visit.
    2. Adapt with Subtle Adjustments: Begin with the plan provided
by the Selection Agent and make careful modifications rather than
overhauling the structure entirely.
    3. Handle Conflicting Requirements Thoughtfully: If you
identify any conflicting requirements (e.g., low budget with
high-density activities), prioritize the user's explicit
preferences.
    4. Communicate Adjustments Clearly: Document each modification
you make to the trip plan, explaining how it better satisfies the
user's needs. If constraints limit your adjustments, provide a
concise explanation.
    5. Success for You Means: Delivering a trip plan that meets
100% of the user's stated requirements. Your answer should include
the trip plan analysis.
    """
```

A.3: System Prompt for Validation Agent

```
"""
Given the user requirement and a trip, tell me the percentage of
user requirements are satisfied by the trip. Also tell me if the
trip is realistic or not based on the ticket price, hotel price,
agenda, etc, and give me a reason why you think so. Structure your
answer like this (do not include '[' or ']' in your answer):

User Satisfaction Rate: [% of requirements satisfied by the trip]
[REASON]

Trip Realism Score: [% of how realistic this trip is]
[REASON]

Your inputs will be USER QUERY and TRIP.

USER QUERY:
[USER QUERY]

TRIP:
[TRIP]
"""
```

Appendix B: Example of One Single Prompt

```
"I want a 3-day trip to South Korea with the budget of 10000."
```

Appendix C: Example of User Satisfaction Rate and Realism Score for User Queries

Example	User Satisfaction Rate - Refined Plan	Realism Score - Refined Plan	Final Score - Refined Plan	User Satisfaction Rate - Baseline	Realism Score - Baseline	Final Score - Baseline Plan
User Query 1	100	80	96	100	40	88
User Query 2	50	90	58	100	95	99
User Query 3	60	60	60	30	80	40

13. Permissions

Below is each team member's permission to post the following three items on a course website (like https://www.eecg.utoronto.ca/~jayar/ece1786.2022/)

	Dechen Han	Ginny Ji	Jason Pu	Selina Sha
permission to post video: yes/no OR wait till see video	wait till see video	wait till see video	wait till see video	yes
permission to post final report: yes/no	yes	yes	yes	yes
permission to post source code: yes/no	yes	yes	yes	yes