Luna Take Home Technical Interview

Assignment Window: 72 Hours

Thursday, November 20 at 4 PM - Sunday, November 23, 4 PM.

1. Core Rules

- Candidates should demonstrate adherence to code quality standards and best practices appropriate to SwiftUI and the programming languages of their choice.
- All templates, architecture choices, and design decisions should be documented in the README.md file, including use of coding agents.
- You are allowed to use the Luna app you have seen as inspiration, and it is expected that in order to showcase your ability that you will come up with a solution to the best of your ability based on user-centric design, while also showcasing your skills, efficiency, and creative thinking.

Track 1: iOS Frontend

Profile: SwiftUI
Goal: Develop an intuitive user interface that allows users to see places and which of their friends, mutuals, or new people also want to go, to convince users to go to these places. The user may also express interest in a place to save the place for later to be invited by someone else who sees the content and sees that the user is interested as well. All of these have the goal of generating social connection/attendance of the places and purchases.
Architecture Standard:

- Follow the **MVVM (Model-View-ViewModel)** architectural pattern for SwiftUI implementation.

Product Market Kit Considerations:

- **Intuitive Usability:** The interface should be immediately understandable and easy to navigate.
- **Social Growth:** Design features that naturally encourage users to invite friends to the platform.
- **Utility:** The interface should provide a lot of user value to encourage users to talk to their friends about it.
- **Requirements:**
    - The user interface should include views that display the current venue and recommended individuals to meet, and/or current users and their preferred locations depending on your design choices.
    - Additional views may be designed to facilitate user flows for selecting preferred

individuals and venues, and implement the invitation process.
    - The specific interaction patterns and visual design are at your discretion, and you will be tested on this.

Track 2: Backend

Profile: Strong emphasis on AI agents and recommendation algorithms.
Goal: Develop a backend recommendation engine capable of identifying optimal venues and compatible individuals.
Note: Consider how data sources such as time spent viewing posts, filter interaction frequency, real-time location data, expressed interest in places and people, and/or connected social media accounts might contribute to the recommendation engine's effectiveness.
Requirements:

- **Recommendation Engine:** Choose your preferred methodology (machine learning models of your choice, or other approaches) to power the recommendation system. The engine should include:
    - **Spatial Analysis:** Determine the best location for the user based on their preferences and constraints.
    - **Social Compatibility:** Assess high compatibility scores for people to go with the user and incentives to attend.
- **Agents:** Implement agents to generate automated reservations, bookings, or purchases once users have agreed to go to a place for a seamless user experience
- The specific implementation and architecture are at your discretion, and you will be assessed on this.

Track 3: Full Stack

Profile: SwiftUI and Backend Development.
Goal: Deliver a comprehensive end-to-end prototype by integrating the components of your choice from both Track 1 and Track 2.
Assessment Note: Since Track 3 encompasses both frontend and backend development, your submission will be evaluated holistically on the integration quality and end-to-end functionality of the individual components that you choose to implement from Track 1 and 2.. The elements from Track 1 and 2 that you decide to implement should be written on the README.md so you can be evaluated accordingly.

3. Submission Guidelines

Deadline: Sunday, November 23 at 4 PM.
Submission Email: nico@lunacommunity.ai
Required Deliverables:

- **GitHub Repository:** Should be publicly accessible.
- **README.md:** Include setup instructions, dependencies/prerequisites, templates and instances coding agents were used, cite third-party resources utilized, architecture choices, and design decisions. Diagrams are recommended.
- **Video Walkthrough:**
    - Format: Unlisted YouTube link.
    - 7 minute restriction.
    - Share your design inspiration and architectural decisions.
    - Showcase why the application is most convenient and useful for users.