

Party Hub

Software Design Specification

M. Buckley (mb) – J. Davy (jd) – C. Roberts (cr)
10-11-2024 - v0.01

Table of Contents

1. SDS Revision History.....	1
2. System Overview.....	1
3. Software Architecture.....	2
4. Software Modules.....	3
4.1. Party Module.....	3
4.2. Host Module.....	5
5. References.....	6
6. Acknowledgements.....	6

1. SDS Revision History

Date	Author	Description
10-8-2024	cr	Created and edited the initial document.
11-8-2024	mb	Added dynamic and static diagrams for modules.
12-6-2024	all	Final changes to match the final project submission.

2. System Overview

PartyHub is a comprehensive party availability website developed to assist patrons in efficiently identifying and RSVP'ing for parties. The tool is crafted to automate party invitations and patron cohort traffic analysis, providing a streamlined approach to the weekend's events.

PartyHub reduces patrons' time and effort in finding and RSVPing for parties. Automating the analysis of potentially mediocre parties helps patrons enhance their poise and mitigate boredom more effectively. This tool speeds up the identification process and improves the accuracy of going to fun parties, making it an essential asset for any party-goer needing some networking.

The website encompasses the following: PartyHub provides some popular public parties that have historically been a hit. It identifies the originating addresses of the host(s) and patrons. It uses this information to evaluate them per-user basis to see if the patron will vibe with the crowd.

We opted to use APIs to gather comprehensive location data for each address. Party Hub analyzes RSVPs and approval ratings data to generate a custom featured party list for each user.

We chose MySQL to store, manage, and analyze extensive data from party and patron traffic analyses. It also provides querying and visualizing data tools for quick decision-making and reporting.

PartyHub offers a user-friendly interface that allows patrons to view real-time data, receive alerts, and configure settings based on their specific requirements. It supports interaction through a web-based platform, ensuring accessibility and ease of use for users with different technical backgrounds. It also ensures that all data handling and processing comply with relevant data protection laws and best practices in cybersecurity. It implements security measures to protect the integrity and confidentiality of the data processed by the website.

3. Software Architecture

PartyHub is designed to analyze hosts' names and addresses to detect vibes and deduce potential party vibes.

Party Analysis Component

Extracts patron accounts and hosts' names using our proprietary AI, utilizing the Party Intelligence reports to determine if the party will fit the patron well.

Party Intelligence Component

Query's our proprietary API for each extracted host's name and party address.

Party Report Generation Component

Generate a summary report of the party analysis results.

UI Component:

Provides a user-friendly interface for users to RSVP for parties, host and edit parties. Some elements of the UI component include Input fields for raw names and addresses, results display area, and administrative controls for managing API keys and settings.

Component Interaction:

The UI interacts with the server by requesting data and transferring input raw party data and uploaded party analysis reports through the UI.

The Server interacts with and accesses the database by fetching/storing/deleting data per request, doing so safely via authentication and hashing.

Design Rationale:

The rationale behind PartyHub is to provide a comprehensive party networking solution by integrating party analysis. This integration allows for a multifaceted approach to party-going, enhancing the accuracy of choosing to attend fun parties using diverse data sources such as address history and hosts' history. A user-friendly interface ensures accessibility for various user web browsing skill levels, promoting efficiency and ease of use. Additionally, the design emphasizes strict adherence to data security and legal compliance, ensuring that PartyHub is a trustworthy and effective tool for combating weekend boredom. This approach addresses real-world networking needs and adapts to the dynamic nature of networking challenges.

4. Software Modules

4.1. Party Module

Primary Function:

Provide a list of featured parties so a potential patron can view information such as the host, address, and date from the party's received headers. Also, provide information about a patron's friends and their activities.

Functionality:

The script processes raw party headers to identify different hosts and RSVP'ed patrons, which it then sends for party and patron traffic analysis.

Interface Specification:

Accepts raw party data as input, which reads from the patrons downloaded party data and outputs the parsed party information, including party analysis results.

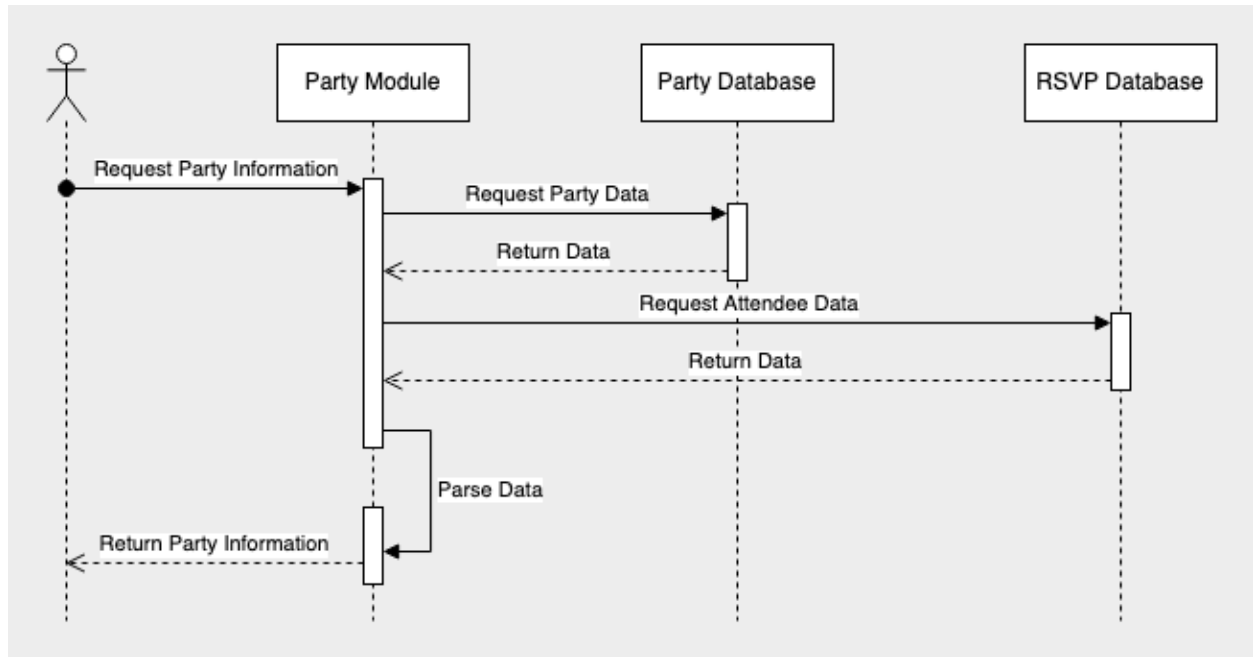


Figure 1.1: Dynamic Model of Party Module (Feed/RSVP)

Figure 1.1 displays the sequence in which the website uses raw party data as input, which reads from the patrons's downloaded party data and outputs the parsed party information, including party analysis and traffic analysis results.

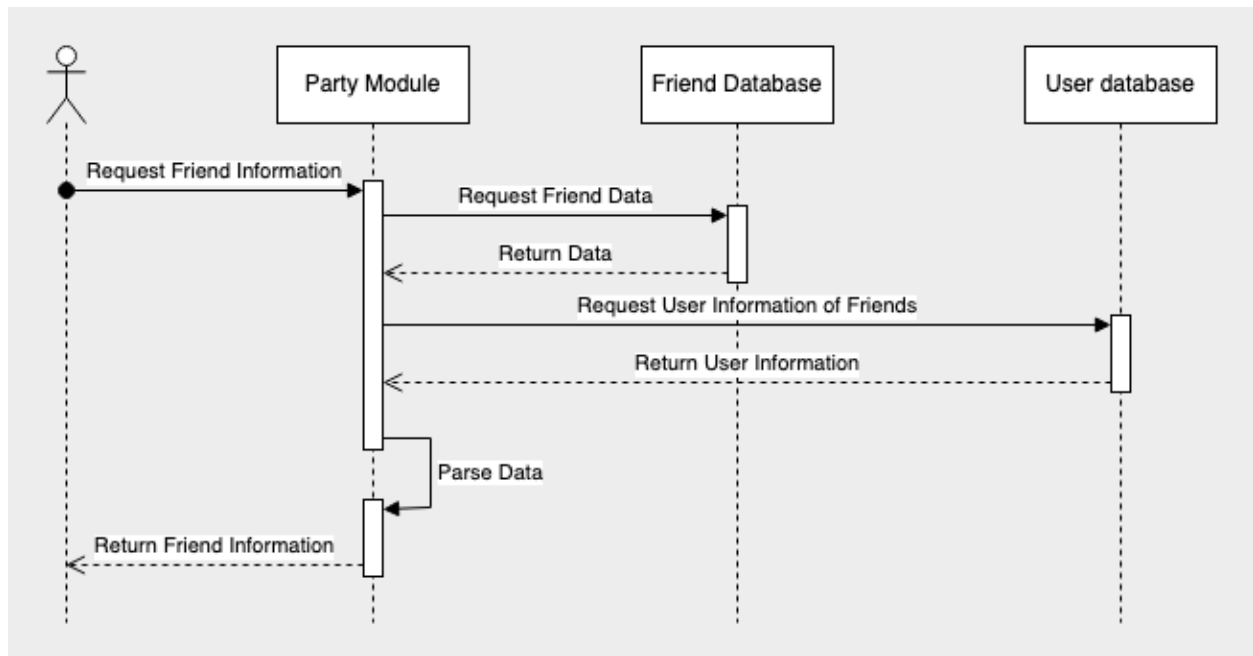


Figure 1.2: Dynamic Model of Party Module (Friends)

Figure 1.1 displays the sequence in which the module receives a request from the user; reads from the friends and user databases, and reports that information back to the user.

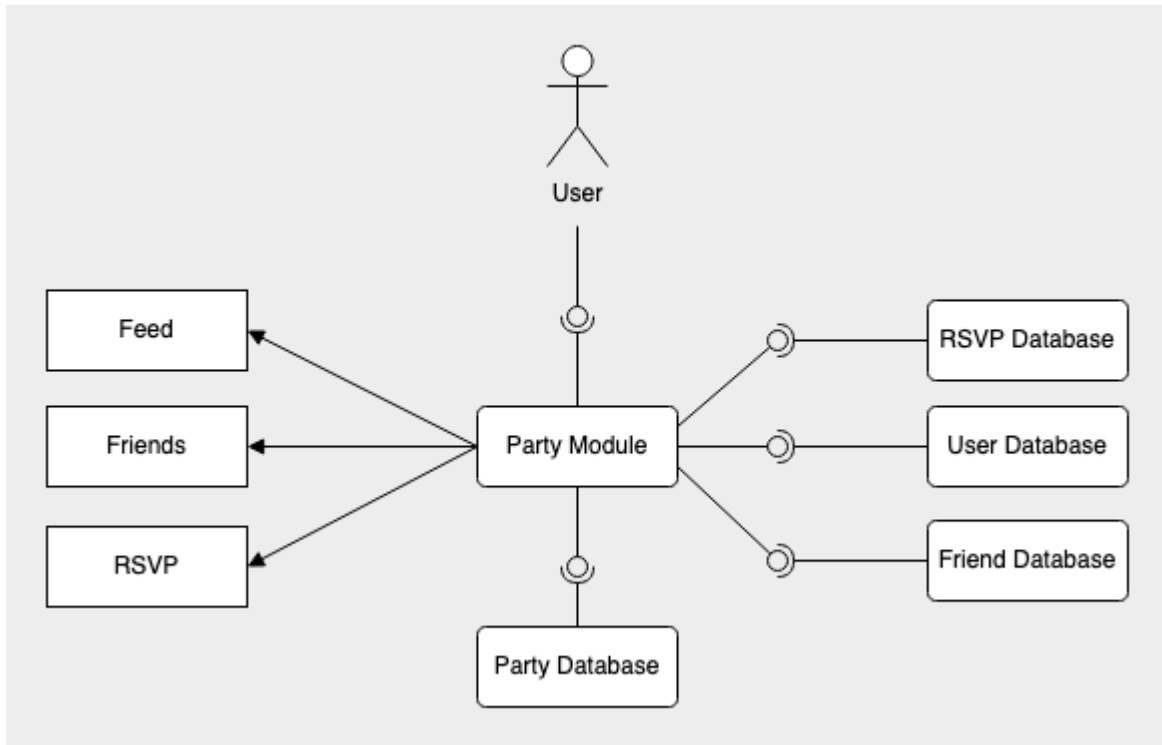


Figure 1.3: Static Model of Party Module

Figure 1.3 displays how the Party Module interacts with the Party, RSVP, Friend, and User databases.

Design Rationale:

To automate a supplied a list of featured parties for a potential patron

4.2. Host Module

Primary Function:

Interfaces with our Host API to create parties

Functionality:

Sends requests to our Host API to post party data such as the hosts, the desired address, and time to the database.

Interface Specification:

Takes an address and returns a result declaring the post successful or a possible failure/error.

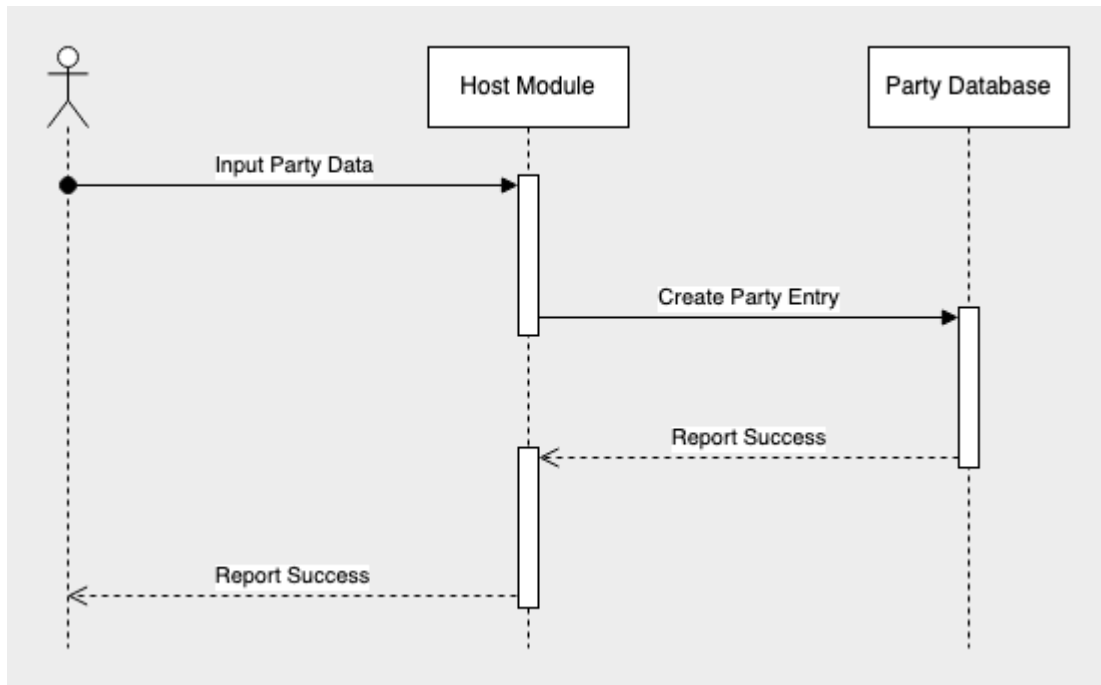


Figure 2.1: Dynamic Model of Host Module

Figure 3.1 displays the sequence in which the user inputs an address and returns a result declaring the post successful or a possible failure/error.

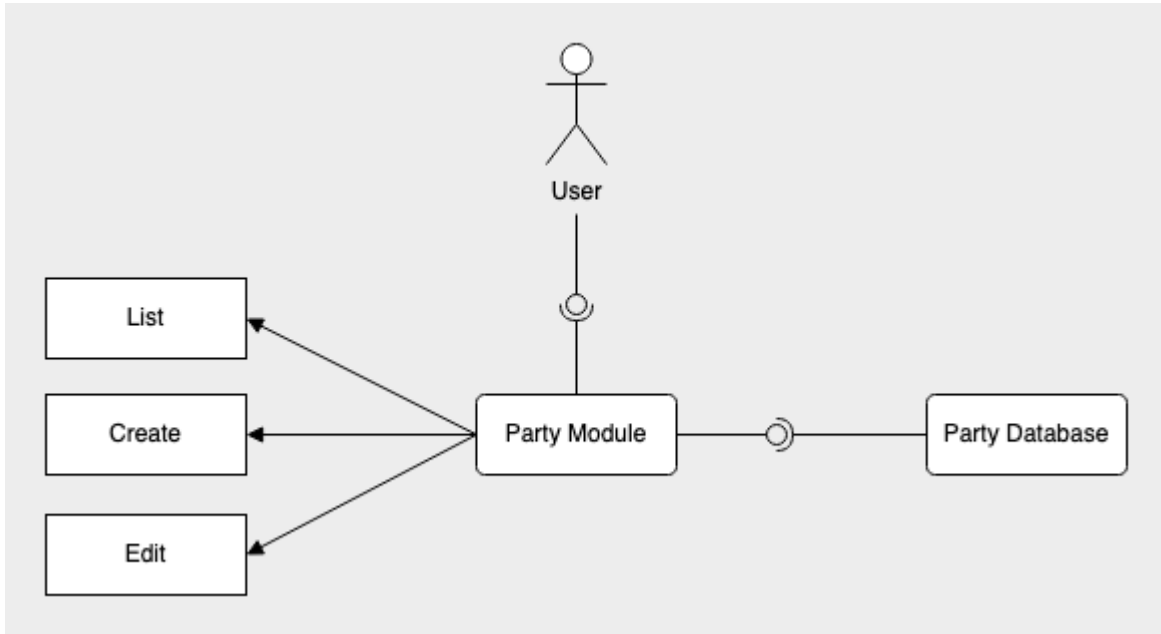


Figure 2.2: Static Model of Host Module

Figure 3.2 displays how the Host Module interacts with the Host API.

Design Rationale:

Easily allow users to create potential parties from addresses noted in the headers.

5. References

Ronny Fuentes, Kyra Novitzky, Jack Sanders, Stephanie Schofield, Callista West. (2021). Fetch: [Fetch: The Tinder for Doggy Playdates Software Design Specification](#)
[Fetch: The Tinder for Doggy Playdates](#)

6. Acknowledgements

All sources were based on the submission and article highlighted on the course canvas here:
[The Tinder for Doggy Playdates Software Design Specification](#)
[The Silver Bullet: Essence and Accidents of Software Engineering](#)
[CS 422 Project Evaluation Criteria](#)