

Travel Itinerary Application

Design Document

| | |
|------------------------------|---|
| Angel Bajracharya 4/10/24 | CSC450 Computer Networks Professor Chays |
|------------------------------|---|

Introduction

This is the design document for a network application that will be a collaborative itinerary planning platform for travelers. Multiple travelers can plan their trip using the same itinerary.

System Objectives

The objective is to design and implement a system that can manage meaningful data that is being concurrently run on multiple different processors. The meaningful data stored in objects would be the itineraries which would store events and the dates/times that they would occur and the location that they would occur in. This application will utilize a client-server architecture where there are multiple clients and one server that manages them. Clients can gain access to an itinerary object and share access to other clients. They can also perform CRUD operations on the object. Another optional feature would be that clients could vote on optimal dates/times/locations for events.

Anticipated Issues and Possible Solutions

Real-Time Synchronization/Collaboration

There will definitely be issues with synchronization of the itinerary objects as multiple clients make changes to it. Some solutions could be to lock the itinerary object when one client is accessing it so that multiple clients are not trying to make changes at the same time. At the same time, every change made by a client should immediately update the itinerary object stored by the server so that all of the other clients have the latest version of the object. The system can also implement version control where each time a client makes a change, the time and change is recorded as a unique version of the itinerary object so that changes can be reversed.

Data Inconsistency

Another issue could be inconsistency in data being input from the client. To maintain data integrity and prevent inconsistent data transmission, the system could implement data consistency checks to validate client input. Also, the system could implement error handling checkpoints to handle errors without hurting the system.

Transportation-Layer Protocols

I made the choice to use TCP for this system because it requires reliable transfer of data.

Application-Layer Protocols

I think I will be using HTTP for authentication of the application.

Algorithms

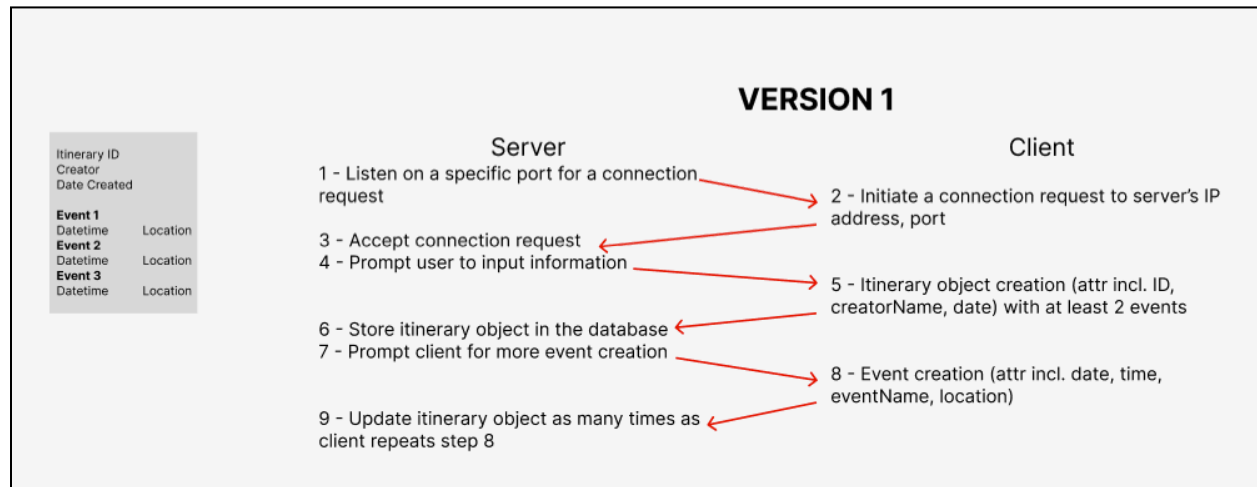


Figure 1 Algorithm for Version 1 (rapid prototype)

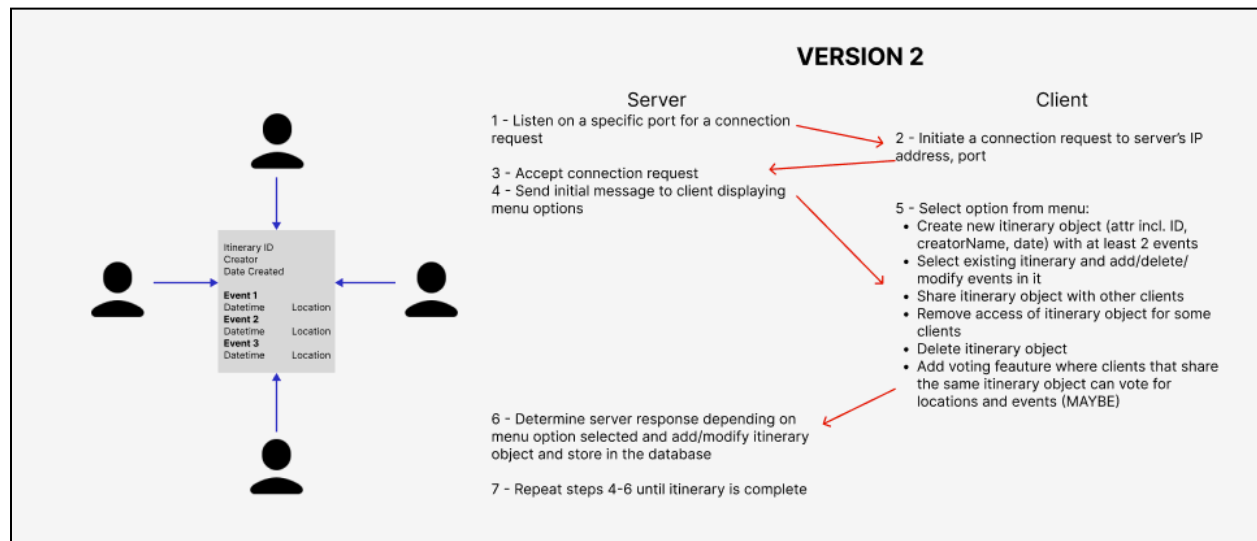


Figure 2 Algorithm for Version 2 (enhanced version)