*Every moment, opportunities to live and experience new cultures fade away. Because it is impossible to delegate all personal time to searching for travel opportunities, cheap travel options go unnoticed. In their wake, loom expensive tickets that steal from tight budgets better spent far away than online. Instead of missing great deals, delegate the searching tedium to an automaton, i.e. your own Virtual Nomad*

# Proposal and Timeline

Build a set of tools with the common purpose of finding great opportunities to travel. Intend to leverage Web Scraping, Data Analytics, Graph Theory, Concurrency, and more to find cheap flights, interesting travel routes, and delegate the act of searching to an automated system using minimal background resources and compact storage methods. Ideally, this project will outlive the Senior Seminar course.

February

* Research and Context
  1. Evaluate several travel listing data sources
  2. Choose 2-4 initial travel listing sources
  3. Dive deep into Golang
  4. Explore data flow issues, in order to…
  5. Decide decent and simple format for Listing data
* Build web parsing toolset for scraping travel listings across multiple sites
  + HTML crawling, parsing, and extraction
  + Credentials and Authentication support
  + Dynamic Page element loading (Use Embedded JavaScript?)
* Development of Actual Features
  + Select satisfactory (e.g. min cost) listing from a collection
  + View details for specific listing
  + View detail summaries for top-priority listings
  + Travel preferences specification
    - Choose locations of interest: []
    - Choose available date ranges: [(A0, A1), (B0, B0)]
    - Choose acceptable times of day to start travel: [MORNING, DAY, EVENING]
  + Logging and debugging toolset
* Initialize test suite for new, regression, and debugging oriented tests
* Ansible Management of server and tasks (or Alternative DevOps practices)

March

* Choose and Deploy a satisfactory data storage option
* Revisit and Improve Data Format
* Generalize data collection across time axis
* Turn on background scraping into DB for later use
* Connect DB to Tableau
* Choose 2-3 Dashboard visuals through Tableau analytics aimed at the following examples
  + Is an extended layover cheaper than a traditional layover?
  + View listing trends over time
  + View listing trends across vendors
  + View listing trends across locations
* Prepare initial results for Presentation (in Mid-March)

April

* Attempt primitive route Finding (see Stretch)
* Fuzzy selection – for choosing amongst similar listings given ranges and preferences
* Track listings meeting criteria for a time range and a set of destinations
* Implement good-deal alerts
* Tune threshold for tracking a listing
* Tune filter for alerts over tracked listings

# Design Principles

* Concurrency
* Polymorphism
* Don’t Repeat Yourself (DRY)
* @Modern Debugging Article
* Build own Data Struct & Algorithms where practical and non-distracting

Stretch Goals

* Expand from local alerts to deliverable notifications
* Extended Layover Generation
  + Create complex routes with extended layovers (optimal for traveling to a pit-stop city)
* Expand to multiple types of travel (e.g. Bus, Train, Bike, more?)
* Leverage AI+NLP to improve parsing
* Design for concurrency on NUMA architecture using Node Replication (NR)