NOMADS

NOMAD TRAVEL

(NT)

SOFTWARE DESIGN DOCUMENT

Prepared by:
Camille Lewis
Mustafa Sameen
Nathaniel Curl

RELEASE DATE
MARCH 12, 2023

Table of Contents

Executive Summary

NOMAD Travel (NT) is a mobile application that simplifies users' logistical travel planning. NOMAD Travel is intended to operate in the market similarly to Expedia, Canoe, and Hopper but includes unique features. NOMAD Travel will utilize travel regulation information from respective governments to provide the user with visa-dependent travel options. NOMAD Travel will utilize personalized e-commerce models and will be distributed as a stand-alone application on mobile devices.

This document provides nontechnical information regarding the purpose and behavior of NT.

Project Description

NOMAD Travel is attempting to break into the personal travel planning industry. The app is intended to target a gap that currently exists in the travel industry: travel regulation information, such as visa travel status, is not easily accessible to consumers of personal travel planning. NOMAD Travel will provide customized travel destinations and opportunities to users depending on their visa and location of origin. Nomad travel will also be able to recommend destinations to users based on preferences they set. NOMAD Travel will use information from travel laws and local governments to connect users to destinations, depending on whether or not they can easily travel to the destination with their Visa.

NOMAD travel must be standalone and not require any network connectivity. The NOMAD Application engines must be installable directly on end user hardware. To appeal to the largest Market, NOMAD travel should be hardware and operating system agnostic.

NOMAD travel will feature an intuitive and user-friendly Graphical User Interface (GUI) that allows travelers to navigate and interact with the app's features and content seamlessly. The GUI will be designed to be visually appealing and easy to understand. It will provide users with a comfortable and engaging experience and enhance the user experience to promote user engagement with the app.

NOMAD travel features a recommendation system that will intake user preferences including climate and activities and output ideal destinations for the user. The recommendation system is designed to offer the user desirable and logical choices for their next vacation, business trip, or getaway. It will save users time and effort with planning logistics of travel, activities, and accommodations. This efficiency will encourage users to travel more frequently using NOMAD.

Document Versioning

Date	Owner	Comment
Mar 6, 2023	Camille Lewis	Feature Matrix & Description, Data Formats
Mar 6, 2023	Mustafa Sameen	Architecture Overview, Detail Design
Mar 6, 2023	Nathan Curl	Diagram, Architecture

Features

The feature matrix enumerates the features requested for the project and the discussion section provides details regarding the intent of the feature. The ids will be used for traceability. Features that all stakeholders have agreed can be removed should strike-through the feature id and have a comment added to discuss the feature being dropped.

Priority Codes:

H - High, a must have feature for the product to be viable and must be present for launch
M - Medium, a strongly desirable feature but product could launch without
L - Low, a feature that could be dropped if needed

Feature Matrix

ID	Pri	Feature Name	Comment	BRD ID
P.1	H	Language	Implementation in JAVA	
I.1	H	CLI		
C.1	H	AppLoop		e1,e2,e3,d1,d2,d3,d4
C.2	M	UserProfile		e1,e2
C.3	H	DataGenerator		e1,e2,e3,d1,d2,d3,d4
C.4	M	DataProcessor		d1,d2,d3,d4
C.5	M	Country		e3,d1,d2,d3,d4
T.1	M	Error Handling		e5

Feature Discussion

P.1 - Language

NOMAD must be multi-platform and run as a standalone application. JAVA is the language selected for this application to run all necessary functions and possible expansions that may include a future GUI or more complex user/data interactions. This language will also support our actions necessary for CLI support.

I.1 - CLI

NOMAD Travel has a fairly simple user interaction pattern that is well suited for CLI operation. NOMAD Travel provides a command line interface for executing travel searches in a command line terminal.

C.1 - AppLoop

Apploop is the main class of the software. It contains a DataReader/Writer and a Scanner to retrieve user input. All NOMAD user interactions will use the same basic event loop.

1. Display welcome message
2. Display user prompts to sign in or create a new profile
3. If new user, Get user input
4. Create User Profile, (jump to step 8)
5. If returning user, prompt user to view destinations or view favorites
6. If user selects view favorites display list of favorites from user profile favorites file, prompt user to return to view destinations or view favorites
7. If user selects view destinations (jump to step 8)
8. Use User Profile to run Destination Generator
9. Display Destination List to User
10. Prompt User to select a Country Card, favorite Countries, or Exit App
11. If user selects a country card, display country information
12. Prompt user to favorite country card or return to Destination List
13. If user favorites a country card, Save user favorited Countries to favorites file within user profile using a dataReaderWriter
14. Return to country card
15. If user selects return to Destination List, display list of destinations
16. Exit application if user selects exit

C.2 - UserProfile

A class that contains information collected from user input and a scanner that always runs within AppLoop. Contains user's Name, Nationality, and a ArrayList<Countries> of favorited destinations.

C.3 - DataGenerator

To create a list of recommended destinations for the user, the application will create an ArrayList<Countries> of Country objects. The DataGenerator utilizes a User Profile object and a DataProcessor Object to create the ArrayList of countries. The ArrayList of countries will be displayed to the user after the DataGenerator has stopped running and successfully created an ArrayList of countries.

C.4 - DataProcessor

The data processor will use user input from a scanner initialized in the AppLoop to parse known .csv databases using a hashmap with key info from the user profile to create a list of destination recommendations that is made of country objects.

C.5 - Country

A class that contains information, based on user nationality, on the countries' Visa, Name, and Capital City.

T.1 - Error Handling

If the user enters invalid information into the scanner, the user will be displayed an error message and prompted to re-enter correct information.

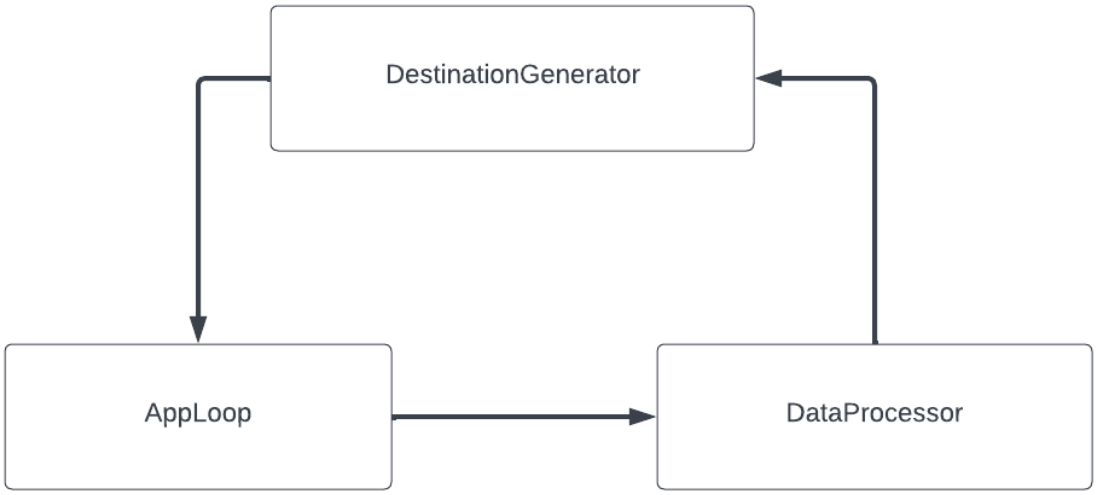
System Design

This section describes the system design in detail. An overview of the major system components and their roles will be provided, followed by more detailed discussion of components. Component discussion will reference the technical features the component helps satisfied.

Note: Design co-evolves with implementation. It is important to start with some sense of components and their interactions, but design docs should be updated during implementation to capture the actual system as-built.

Architecture Overview

High-level System Architecture



Major Components

AppLoop

I.1, C.1, T.1
User interaction is handled via the AppLoop. When NOMAD Travel is started, only the CLI will run. The AppLoop is responsible for the basic event loop in NOMAD and to manage user interactions, display information, obtain input, generate destination options based on user preferences, and prompt the user to make a selection.

This component uses the Command pattern to provide the functionality to encapsulate user actions as commands and execute them using the event loop.

DestinationGenerator

C.2, C.3, C.4
The Destination Generator is a component in the NOMAD Travel app responsible for generating a list of destination options based on the user's preferences and criteria. It implements the Strategy Pattern such that it uses the user profile data like their nationality and generates a recommended list of destinations based on visa availability and travel regulation information. The generated list is then presented to the user, who can select a destination from the list to learn more about it or favorite it for future references.

DataProcessor

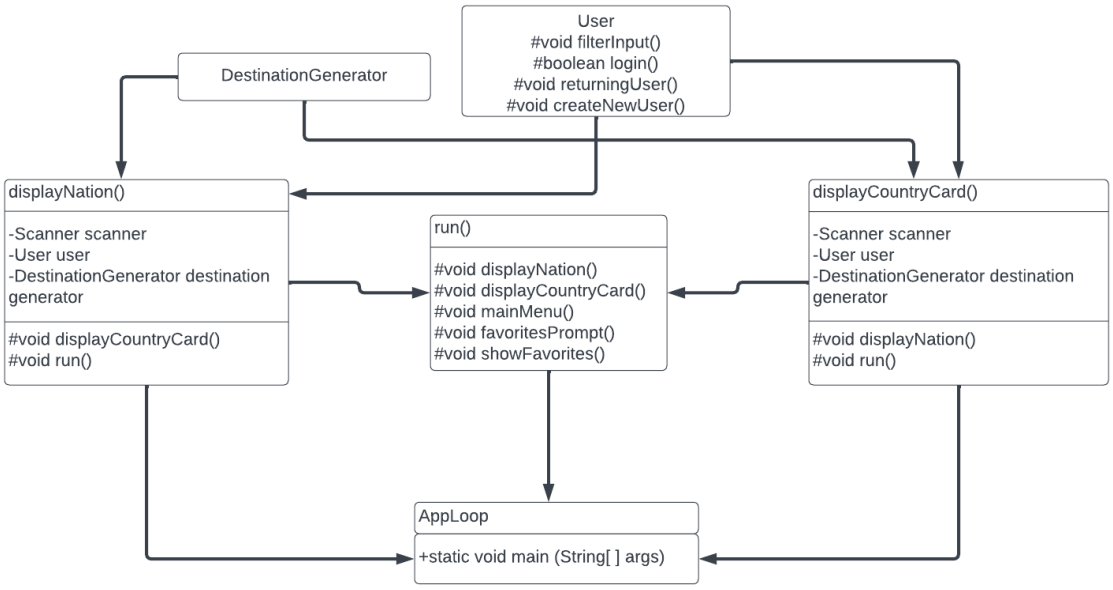
C.2, C.4
The DataProcessor component receives user input and uses a hashmap to extract relevant data from known .csv databases, generating a list of country objects that match the user's preferences. The component then ranks the country objects based on various factors and returns a final list of destination recommendations to the AppLoop as an ArrayList of Country objects.

Detail Design

AppLoop

The AppLoop is the central component of NOMAD Travel that manages the basic event loop and handles user interactions. When the app starts, the AppLoop displays a welcome message to the user and prompts the user to sign in or create a new profile. If the user signs in, the AppLoop loads the user profile data from a file. If the user creates a new profile, the AppLoop creates a new profile file and prompts the user to enter their profile information (e.g., name, age, budget, travel preferences).

AppLoop



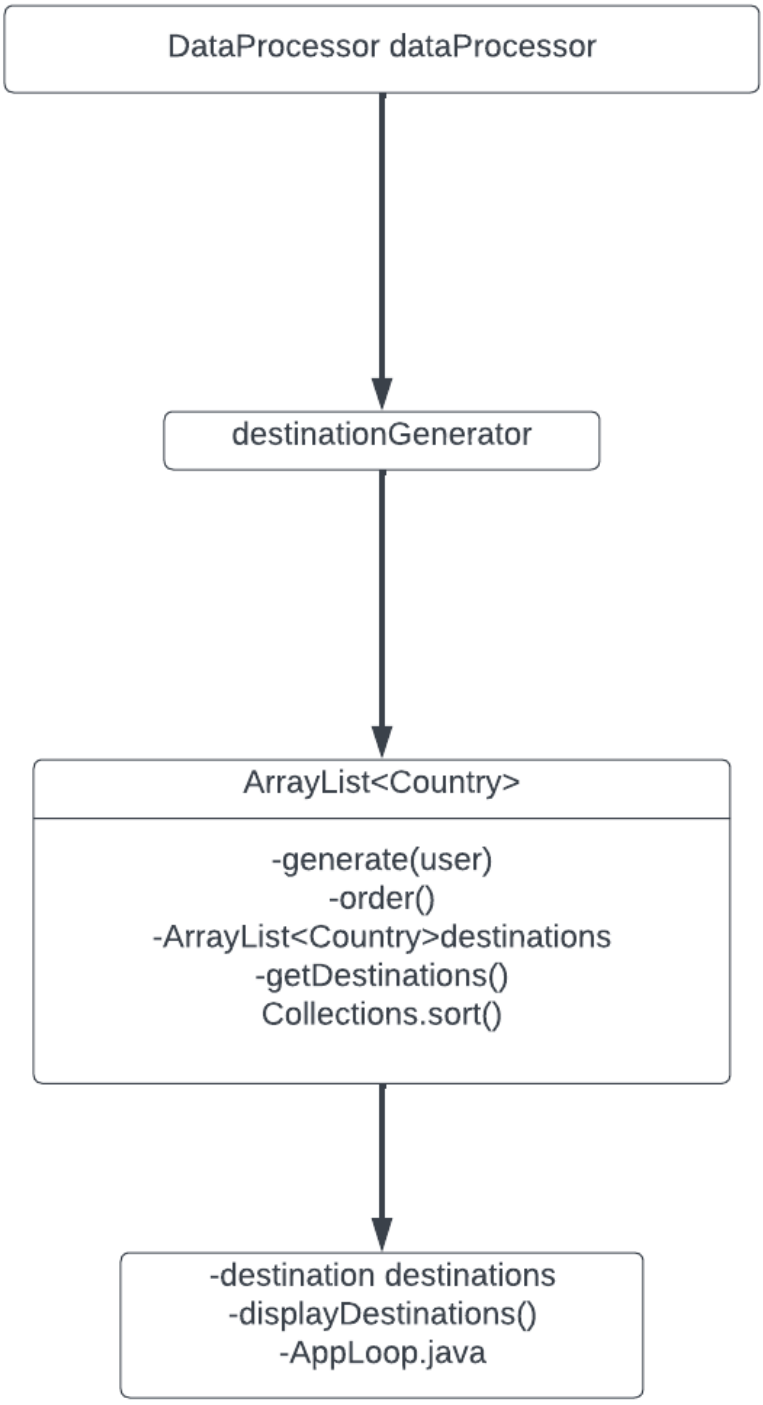
User Interface

The user interface for NOMAD Travel is a Command Line Interface (CLI) that displays messages, prompts, and menus to the user and accepts input from the user.

DestinationGenerator

The Destination Generator component in the NOMAD Travel app is responsible for generating a list of destination options that match the user's preferences and criteria. It implements the Strategy Pattern, which allows different algorithms to be used for generating the list of destinations based on user input or other factors.

DestinationGenerator



User

User is a class that holds essential user information collected from input and a scanner within the AppLoop. It includes data such as the user's Name and Nationality, which are used to provide personalized recommendations. The class also contains an ArrayList<Countries> of favored destinations, allowing users to save their preferred travel locations for future reference. It will have a favoriteDestination() method that will take in a Country object and add to the ArrayList of favorited countries.

Country

Country is a class that holds essential information about a country based on the user's nationality. It includes data such as the country's Visa requirements, Name, and Capital City, allowing users to make informed decisions about their travel destination. The class is used by AppLoop to display country information and by Destination Generator to store country information after extracting data with the DataProcessor.

Generate

Generate is a method that generates a list of recommended travel destinations based on various factors, including the user's nationality, visa availability, and travel regulation information.

Order

Order is a method that sorts the list of recommended destinations based on visa availability for the user.

DataProcessor

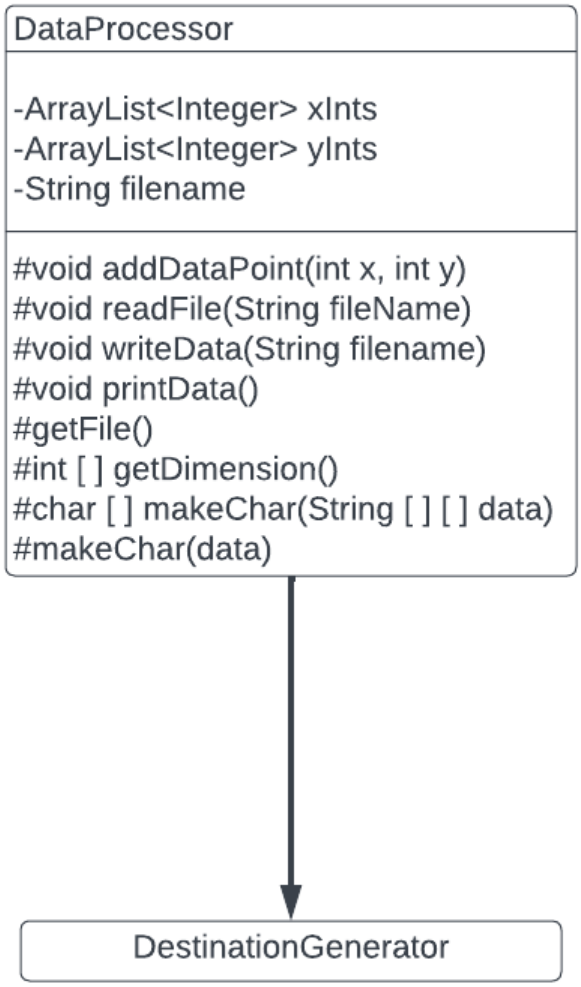
The DataProcessor component is responsible for reading and writing data. It generates a list of recommended destinations and writes user information to a .csv file.

DataReader

A subclass that reads data from .csv datasets. The getVisaInfo() method takes the user's nationality as input and reads the visa information of other countries. It then creates a hashmap with country names as keys and visa information as values.
The getCountries() method takes this hashmap and reads data from another dataset of countries to create an ArrayList of Country objects with all the relevant information.

DataWriter

DataWriter will write the user profile data and the list of favorite destinations into a csv file.


DataProcessor
-ArrayList<Integer> xInts
-ArrayList<Integer> yInts
-String filename
#void addDataPoint(int x, int y)
#void readFile(String fileName)
#void writeData(String filename)
#void printData()
#getFile()
#int [ ] getDimension()
#char [ ] makeChar(String [ ] [ ] data)
#makeChar(data)
DestinationGenerator

Data Formats

User Profile Format

The user profile is a .txt text file that includes user information such as the user name and nationality.

Example File

```
[name]
“User Name”
User Nationality
```

Country File Format

The Country File is a .txt text file that includes information stored in the ArrayList<Countries> Destination recommendations created by the Dataprocessor and DestinationGeneration classes. The format of the data Countries is found in the Database example below, and a .txt file example is also provided below.

Example Country Data

Country	Region	Population	Area (sq. mi.)	Pop. Density (per
227 unique values	SUB-SAHARAN AF... 22% LATIN AMER. & C... 20% Other (131) 58%	7026 1.31b	2 17.1m	0
Afghanistan	ASIA (EX. NEAR EAST)	31056997	647500	48,0
Albania	EASTERN EUROPE	3581655	28748	124,6
Algeria	NORTHERN AFRICA	32930091	2381740	13,8
American Samoa	OCEANIA	57794	199	290,4
Andorra	WESTERN EUROPE	71201	468	152,1
Angola	SUB-SAHARAN AFRICA	12127071	1246700	9,7
Anguilla	LATIN AMER. & CARIB	13477	102	132,1
Antigua & Barbuda	LATIN AMER. & CARIB	69108	443	156,0
Argentina	LATIN AMER. & CARIB	39921833	2766890	14,4
Armenia	C.W. OF IND. STATES	2976372	29800	99,9
Aruba	LATIN AMER. & CARIB	71891	193	372,5
Australia	OCEANIA	20264082	7686850	2,6

Example Country .txt File

```
[name]
“Country Name”, Visa, Capital
```