**Project Vision**

**Smart Travel Application**

Project Description

Travelling is certainly an activity that more and more people do nowadays, not only does it allow people live new experiences but it also gives the chance to relax and spend time in family. However, there are factors that can completely limit or ruin this experience by bringing in frustration or disappointment. One of the most relevant is time, since it can be relatively short when there is more than one destination to visit. Despite the fact that many travelers make an effort to plan an optimal route without following a formal methodology, it may result in a tedious and worthless activity provided that the final result may be completely useless in terms of saving time. With all this in mind, an application that would help travelers plan their trips by providing them with a time-effective route will allow them to forget about all the stress that comes with planning a travel route.

Deliverables / Outcomes

A GUI application will be developed for this project. It will display a map of Ireland, where users will be able to add destinations to finally be provided with a time-effective route.

Functionality & Technology

Smart Traveler will be a desktop app that will display a map of Ireland where users will be able to mark all destinations to visit. After clicking the button to request an optimal path, the application will start processing all locations and using certain algorithms, it will come back to the user with a time-effective route to visit all places only once; this means the path will not pass by the same location more than one time. The final result will be displayed to the user on a panel that will indicate the order in which the locations must be visited.

List of key features:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **MUST HAVE** | | **SHOULD HAVE** | | **COULD HAVE** | | **WON’T HAVE** | |
| **ID** | **Feature** | **ID** | **Feature** | **ID** | **Feature** | **ID** | **Feature** |
| 001 | Implement Brute Force algorithm | 010 | Use one of the 3 algorithms to calculate an optimal route | 018 | Save results for future similar requests | 020 | Include maps of other countries |
| 002 | Implement Nearest neighbor algorithm | 011 | Clear existing selection | 019 | Suggest popular places | 021 | Database integration |
| 003 | Implement a Genetic algorithm | 012 | Not crash at any moment |  |  |  |  |
| 004 | Display map of Ireland | 013 | Display total number of kilometers to travel |  |  |  |  |
| 005 | Allow users to add destinations | 014 | Delete one selection at a time |  |  |  |  |
| 006 | Submit user selection and apply algorithm | 015 | Read JSON result from Google Maps API calls |  |  |  |  |
| 007 | Display results generated after applying an algorithm | 016 | Display error message when user submits less than 3 locations |  |  |  |  |
| 008 | Get coordinates of selected locations | 017 | Allow users to set an initial point for tour |  |  |  |  |
| 009 | Make HTTP requests through Google Maps API |  |  |  |  |  |  |

To accomplish the development of this project, the following items will be required:

* Algorithms to solve the TSP (Brute force, nearest neighbor and genetic algorithm)
* Java SE 8u112 (latest version at the moment)
* NetBeans IDE 8.2 (latest version at the moment)
* Google Maps API
* GitHub

Audience

All travelers around the world would be beneficiated by using this app, therefore, every person who travels is considered part of the audience.

Data, Information, Content

The data used in this application will be drawn from the Google databases through the Google Maps API. Once the user has selected all the desired locations and press the button to start the process, HTTP requests will be made via the API in order to retrieve all the information needed, which will be finally sent to the application in JSON format.

Partners

NA