Globe Hopper APP Requirements

# Project

Acme World Travel Inc. has commissioned a project to enable world travelers to research countries of the world prior to embarking on a travel. Acme has decided to develop this capability through the presence of a dedicated web site supported by a persistent data store. The site will be known as **Globe Hopper.**

Travelers will be expected to register on the site with their email address and setup their secure credentials that will consist of a user identifier and a password – they could use their email as their user identifier**. (Not needed for 3 days)**

Travelers can search for all countries in the world or for specific countries based on a criteria. A Country information will consist of the following:

* Country Name
* List of Cities (no more than 3 cities including the capital city)
* Population
* Continent

For any given Country, a Traveler should see no more than 3 cities including the Capital City.

A City information will consist of the following:

* City Name
* Capital Flag
* Country ID
* Landmark 1 details
* Landmark 2 details
* Landmark 3 details

## Assumptions

* A single user with full authorization can access the application
* Up to three city landmarks will be displayed at this time for any given city
* Deployment is out of scope.

## Functional Requirements

* Ability to search for all countries of the world as a Traveler
* Ability to search for all countries in a given continent as a Traveler
* Ability to get details about the capital city of the country as a Traveler
* Ability to add/update/delete Country and City information as a Travel Agent

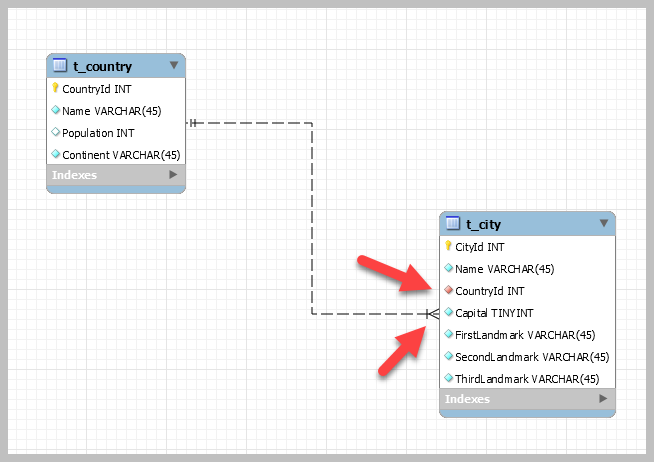
## Non Functional Requirements

* 99.99 % availability, 24 \* 7

## Development Infrastructure (DEVOPS)

* Waterfall Methodology will be used since the requirements are clear and limited.
* The application API/Webservices will be built using Python Flask as the language of implementation.
* Database will be MySQL
* For CI/CD:
  + GitHub will be used as the SCM to allow you to check in your code and
  + Build Pipeline will be built in GitHub Actions or Jenkins to build and deploy code to SonarQube for code quality
* Automation API Testing will be performed on TOSCA
* **Deployment - TBD**

## Database Schema



Country → City has a 1:N cardinality.

City → Country has a 1:1 cardinality.