**CSE 445/598 Project 3 (Part I)**

Team Megaminds-598

Prerna Satija; Nitesh Kedia; Nishant Bansal

1. **Description:**

We will build a Tourism application which is a 4-tier service oriented computing system consisting of a Presentation Layer, Application Layer, Service Layer and Data Management Layer.

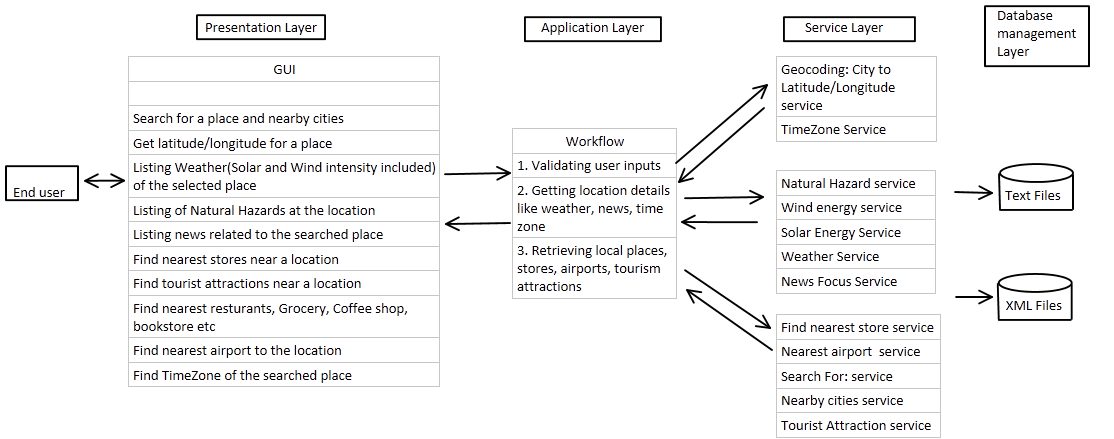
The GUI will consist of a form showing buttons for various functionalities like if a user enters a zip code or location, he can find the city that zip code belongs to as well as the nearby cities. Also users can get information about any natural hazards in that particular area, wind energy index and solar intensity, list of tourist attractions in that area and any stores/hotels/restaurants/hospitals nearby. The tourist might also want to know about any live events occurring during his visit to the place for which a date box will be provided on the GUI.

Application Layer will consist of a workflow to invoke the services of Service Layer.

The service layer will consist of around 15 services that a tourist/user would require from our Tourism application. These services are listed in sections 1.2 and 1.3.

Data Management Layer will consist of different sources that we might need during the course of this project to store the data. These could be a text file, and XML file or a dictionary.

1. **Diagram of the overall system**



1. **Service Directory**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| This page is deployed at: | | | | |
| This project is developed by: Team Megamind-598 | | | | |
| **Provider name** | **Service name, with input and output types** | **TryIt link** | **Service description** | **Planned resources need to implement the service** |
| Prerna Satija | **SolarEnergyService**  **Input:** latitude and longitude  **Output:** a number reflecting the annual average solar intensity at the location. |  | Create a service that returns the annual average sunshine index of a given position (latitude, longitude). This service can be used for deciding if installing solar energy device is effective at the location. | <https://eosweb.larc.nasa.gov/cgi-bin/sse/global.cgi?email=skip@larc.nasa.gov> |
| Prerna Satija | **WindEnergyService**  **Input**: latitude and longitude  **Output**: a number reflecting the annual average wind intensity at the location. |  | Create a service that returns the annual average wind index of a given position (latitude, longitude). This service can be used for deciding if installing windmill device is effective at the location. | <https://eosweb.larc.nasa.gov/cgi-bin/sse/global.cgi?email=skip@larc.nasa.gov> |
| Prerna Satija | **TimeZone**  **Input:** latitude and longitude  **Output:** String that gives the timezone and current time |  | Service returns the timezone and current time given latitude and longitude of a location. | <https://developers.google.com/maps/documentation/timezone/> |
| Prerna Satija | **NearbyCities**  **Input:** latitude and longitude  **Output:** current location’s city, state and nearby cities. |  | This service takes latitude and longitude as input and returns the city, state and nearby cities. | <https://developers.google.com/places/> |
| Nitesh Kedia | **WeatherService**  **Input**: US Zip code  **Output**: An array (or list) of strings, storing 5-day weather forecast for the given zipcode location. |  | Create a 5-day weather forecast service of zipcode location based on the national weather service given in planned resources. | <http://graphical.weather.gov/xml/SOAP_server/ndfdXMLserver.php?wsdl> |
| Nitesh Kedia | **News Focus**  **Input:** topic  **Output:** News related to the input provided. |  | The service provides news related to the topic/place provided as the input. | <https://news.google.com/news/feeds?q=india&output=rss> |
| Nitesh Kedia | **NearestAirport**  **Input:** latitude and longitude  **Output:** name of the nearest airport |  | The service will return the nearest airport for the given latitude and longitude. | http://airports.pidgets.com/v1/airports?near=33.4294,-111.9431&n=1 |
| Nitesh Kedia | **TouristAttractions**  **Input:** latitude and longitude  **Output:** list of places for sightseeing |  | For the given longitude and latitude location, this service will return the list of tourist attraction places. | https://maps.googleapis.com/maps/api/place/nearbysearch/json?location=33.4294,-111.9431&radius=500&key=AIzaSyAtGofwEQndI-3ZtvDWg8rOS0aam1XXpUo |
| Nishant Bansal | **NaturalHazardsService**  **Input:** latitude and longitude  **Output:** a number reflecting the natural hazards at the location. |  | Create a service that returns the natural hazards (Tsunamis, earthquake, volcanoes) index of a given position (latitude, longitude). This service can be used for building decision and insurance premium. | <http://www.nesdis.noaa.gov/>  <http://maps.ngdc.noaa.gov/>  <http://gosic.org/> |
| Nishant Bansal | **FindNearestStore**  **Input:** two strings zipcode and storeName  **Output:** string message |  | Use an existing online service or API to find the provided storeName closest to the zipcode and return the address.  If no store is found, return an error message.  (Optional: if the store is further than 20 miles, from the zipcode, return a “no stores within 20 miles” message). | [Yelp.com](http://www.yelp.com/)  <http://www.programmableweb.com>  [foursquare.com](https://foursquare.com/) |
| Nishant Bansal | **Geocoding**  **Input:** city name  **Output:** latitude and longitude |  | This service returns the longitude and latitude for the input city name | <https://developers.google.com/maps/documentation/geocoding/> |
| Nishant Bansal | **SearchFor**  **Input:** latitude and longitude, storetype  (Restaurant, Grocery, Coffee shop, bookstore, ATM/Banks, Parking lot, Hotels, Casinos, Transport, Salons, Cinema)  **Output:** array of nearby storetypes |  | This service takes the latitude and longitude, and store type, as inputs and returns the list of store names nearby. | <https://developers.google.com/maps/documentation/javascript/places>  <http://api.yelp.com/v2/search> |