Texada GPS Read Me

Entity Relationships

From the input.txt.docx document sent with the assignment instructions, we could understand that the data consisted of products and locations. Therefore in implementing the back end application, we sorted out two main entities:

1) Product
2) Location

There is one-to-many relationship between Product and Location, since a product can visit many locations, and for each location to make sense, we need to indicate which product visited it at a particular time.

Technologies used

The below technologies were used to implement the back end:

1. Framework and Tools: The Flask framework and Python 3 were a requirement for the assignment. So the back end has been built with these technologies. The project was developed using the PyDev plugin on the Eclipse Oxygen IDE.
2. API: The back end consists RESTful APIs that consume and produce JSON objects.
3. MySQL Server: This is an open source, matured and robust database management system.
4. Flask-httpAuth: We secured our endpoints. Operations that require data modification such as CREATE, UPDATE, and DELETE, require the caller of the API to be authenticated. For this application, we created a user with username **admin** and password **admin**. This user is stored in the user table of the application database. The Basic HTTP Authentication mechanisms has been implemented for this project using Flask-httpAuth. Other authentication mechanisms such as LDAP, and OAuth 2.0 could be considered in future.
5. SQLAlchemy: The application is object oriented. To make it robust and easier to maintain, we used the SQLAlchemy for its object relational model. With this approach, we have we don’t need to write SQL statements directly into our application. Thus our application becomes easily maintainable and flexible. We could change the database engine from MySQL to Orable with little or no code change.
6. Test Driven Development(TDD): Our application was developed with approach. There are 59 automated tests. With these tests in place, we can be confident that any future changes to the software that breaks it can easily be detected. The endpoints and data access layer components each have their own tests.
7. Exception handling: The application is has been designed such that exceptions raised in the application are first logged, then a more friendly error is sent to the client in a JSON format consisting of an error message and an error code. The callers of the API are other applications. So it is important to return JSON objects for exceptions as well.
8. Data Validation: There are two layers of data validation. The first layer is in the data access components. The second layer is at the level of the constraints set on the database tables.

Date formats

All dates are converted to UTC before being saved.

Deployment Instructions

Run the below MYSQL script commands in order, on the selected database:

- source /scripts/schema.sql
- source /scripts/insert_users.sql
- source /scripts/insert_products.sql
- source /scripts/insert_locations.sql

Endpoint formats

To create locations and products

Use HTTP POST on http://host_name_or_ip:port/texada_gps/api/locations/ or http://host_name_or_ip:port/texada_gps/api/products/

To update locations and products

Use HTTP PUT on http://host_name_or_ip:port/texada_gps/api/locations/{id} or http://host_name_or_ip:port/texada_gps/api/products/{id}

To delete locations and products

Use HTTP DELETE on http://host_name_or_ip:port/texada_gps/api/locations/{id} or http://host_name_or_ip:port/texada_gps/api/products/{id}

To read locations and products by id

Use HTTP GET on http://host_name_or_ip:port/texada_gps/api/locations/{id} or http://host_name_or_ip:port/texada_gps/api/products/{id}

To read locations and products

Use HTTP GET on http://host_name_or_ip:port/texada_gps/api/locations/?page_number={page_number}&page_size={page_size} or http://host_name_or_ip:port/texada_gps/api/products/?page_number={page_number}&page_size={page_size}

The pagination parameters are optional.

Database connection settings

In the application directory, the file **/intance/config.ini** has the database url settings. Consider changing these for your environment.

Thank you!