Software Engineering (CSC 510) Mini project 2 report

Architecture Design and Description

Gati Shah (gshah)
Priyanka Shankaran (priyan)

Architecture diagram description:

An architectural style is a particular pattern that focuses on the large-scale of a system. They are a sets of principles that shape an application. Architectural patterns are important artifacts containing specialized design knowledge to build good-quality systems. Complex systems often exhibit several architectural patterns in their design which leads to the need of architectural pattern composition.

This software architecture utilizes a layering scheme. The layering principles are similar to those used in the ISO/OSI Reference Model. The following principles and rules are among those used in defining the architecture:

a. Hierarchical approach
b. Lower layers provide services
c. Upper layers maintain states
d. Commands flow down
e. Data flow up and down
f. No direct software dependencies “up’ the architecture; down only
g. The lower layers have no dependencies on higher layers

There are six layers in the architecture. The layers are named and described, from the top layer down, as follows:

1. **User**: This is the component which sends a request to the software. The user for the system can be differentiated into collector and the moderator. The moderator reviews both individual instances and aggregate reports of activities and the collector goes on reports data. Each have their own set of tasks and services available from the system.

2. **Client Application**: This is the interface to the outside world (website). The interface can either be mobile client interface or web client interface.

    - **Load Balancer:** The job of a load balancer is to distribute the load - incoming HTTP(S) requests amongst a pool of computers running web servers. The load balancer may simply redirect the request to a different server. The load balancer also monitors the liveness of each of its computers and if one goes down, simply redirects traffic to others in the pool.

3. **Web Server:**
    - **Web application layer**: This component provides the mechanics necessary to create the interface to the outside world (web services, web application frameworks)
    - **Static content layer**: This layer serves as the software that accepts and supervises the static HTTP content.

4. **Application Server**: The application layer along with the 3rd party services such as GPS & tracking and the the computer vision component serves as the core of the application. The various controllers works in collaboration to provide the necessary service and serve the request generated by the user. The various controllers within the application server are defined as follows:
    - **Association & Calibration controller:** The calibration and association controller works together to calibrate and associate a video frame with a gps position. The association controller also coordinates with the upper layers as to where the data should travel next (Data access component, other controllers, computer vision component or 3rd party services).
    - **Computer Vision Analyzer:** The computer vision analyzer calls the computer vision component and extracts the video frames, time position, suspected type of activity, and set of geometric boundaries and overlays provided by the computer vision component. It distributes frames of videos to different processing nodes that run the computer vision analysis.
    - **Geographically distributor:** The geographically distributor provides the load balancing and the dynamic distributed computing for the application. It determines which community moderator is responsible for a given geographic area and redirects the requests sent by this user to the geographically closest server.
    - **Duplicate issues detector:** This controller detects any duplicate issues submitted by the user or detected from the frames and eliminates it. This prevents duplicate entries being entered in the databases.

5. **Data Access and Mapping layer:** The data access layer makes the data available in the database readily available to the controllers in the application layer. It makes data suitable for use for the various layers above it. It also serves the layers above by mapping the data to right database and storing the data retrieved from the upper layer to the right database.

6. **Database layer:** The database layer has the storage system and metadata required for storing the data collected and retrieved from various users. The database has the following relational tables to store all the information:

- Users : Collector or moderator
- Street
- GPS tracking history
- Pending issues

The architectural interaction between components is established by **explicit** **invocation** which allows the client to invoke operation on the server. The client should provide information on how to locate and invoke operation explicitly through operation "name" and "parameters".

The data access component and the database together acts as a **repository** model and mediates between the database layer and the application layers of the software. It queries the data source for the data, maps the data from the database to a application layer, and persists changes in the application layer to the database. A repository separates the actual logic from the interactions with the underlying database. The separation between the database and application provides the following benefits:

- It centralizes the data logic.
- It provides a flexible architecture that can be adapted as the overall design of the application evolves.

References:

1. http://en.wikipedia.org/wiki/Load_balancing_(computing)
2. Software Architecture in Practice By Len Bass, Paul Clements, Rick Kazman