COP 5725

Database Management Systems

Project Deliverable 2

Group 21

Chen, Zhaoyang <zhaoyang.chen@ufl.edu>

Feng, Zhongxi <zhongxi.feng@ufl.edu>

Niu, Yicong <yicong.niu@ufl.edu>

Qi, Lin <linqi@ufl.edu>

Interactive City Weather Report

Project Phase II

User Interface Design & Database Design

Contents

Overview and Description of the Application 3

User Interface Design 3

Conceptual Database Design 6

Overview and Description of the Application

For everyone, weather information is an indispensable part of daily life. At the same time, the analysis of past weather data can provide a good understanding of the local climate change. Research on the weather also helps people travel, arrange flights, and agricultural production. In this project, we will build a website that illustrates complex trend queries using comprehensible and pretty graphs to show our viewers a detailed analyzing result. By viewing the graphs, our viewers may get straightforward information about how the weather changes over time in a specific area. The website also accepts customized query input by users.

User Interface Design

The entry page in our project is the user login page. The corresponding UI interface is designed as a box saying “User login”, which requires user input of the user ID in the first row, and a password in the second row. Like most designs, passwords should be encrypted as a line of black dots as one types in. Also, if the user has input either the mismatched ID or password, the corresponding error message will be shown next to the box and it will require another try from the user. This feature would be the first page of our application.

To access the data and our service, a user must log in with her userID and password. Since we are in the picturing stage, many functionalities could variate with project growth. User authentication could be first set up in our database, for example, stored in a table named UserInfo. Only by passing the correct combination of userID and password stored in UserInfo could one log in successfully. Later with everything becoming full-fledged, we could achieve user logging in through dynamic registration, which will direct the new user to a registration page before logging in. This would also be achieved by clicking on a link on the entry page.

Besides, there should be a note reminding users of the format of their ID and password during registration. For example, the password should contain at least 6 characters along with symbols and numbers. Users also need to re-type their password for confirmation. On the other hand, the ID should be unique and there will be popping up caveats if a user attempts to register an ID conflicting with existing ones.

Moving forward, after successful authentication, the next page should be an introduction page/background showing what we do. A concise and beautiful background always captures people’s eyes. Considering that our dataset is the weather report of 30 cities in North America(the

U.S. and Canada), we could put a map on the entry page. The user could move a mouse over a specific area and thus trigger a detailed display of the weather report with of that area. This enhancement of interaction is what we will be heading for during this semester-long project.

Complex trend queries are integrated into the map. Each complex query will be illustrated in graphs after clicking into a specific city on the map. Queries include, first, monthly (average / highest / lowest) temperature changes along with certain weather, say foggy, moist, or clear, in each city. Second, monthly changes in the duration of a certain wind speed in each city. Third, monthly changes in the duration of a certain wind direction in each city. Furthermore, how the monthly temperature of cities in different latitudes varies.

In our dataset, the weather is described from four major features, namely humidity, temperature, pressure, and wind. In the UI design, we plan to include the four features in a tab on the top of the entry page. Thus, by clicking into one feature tab, a user could get a deeper understanding of that specific feature. Also, due to a large number of possible queries that could be triggered by one general feature, it is necessary to specify the specific queries under each tab. For example, we need to design 5 complicated queries, so we could display the result here. One could get the temperature changing trend in Spring in cities in the western hemisphere under the “Temperature” tab.

For each query result, we plan to trigger the display by clicking a button or by submitting a web form.

In the first case, like mentioned before, if we plan to show a pre-designed query that displays the trend in temperature, we will put a button there and wait for the triggering event. By clicking it, our service will return the result from the backend and display it on the web page. Regarding query analysis, we plan to return the results in flow charts so one can tell the difference in an instant.

In the latter case, if someone wants to get customized data and try to find something interesting, she could obtain new data by inputting a customized query. We will provide the users with a dropdown list regarding each field. Such as a list of city names, a list of seasons, and a list of years. After selecting each value, one could click the “Submit” button and the backend will handle the query. So, the user could find the humidity value in Vancouver in Spring in 2016 under the “Humidity” feature tab. Of course, this dropdown list feature should also be included under the corresponding feature tab for usability and clarity. For this kind of user query, we plan to return the result in a table since it is merely static data. All the query results could be displayed in the raw data mode, where the table can be downloaded as a file.

Switching between tabs gives a more effective way of exploring our dataset. During the switch/data display, the background page would remain unchanged for usability.

The clear description about the flows of action and web pages is vividly depicted in the Figure 1.

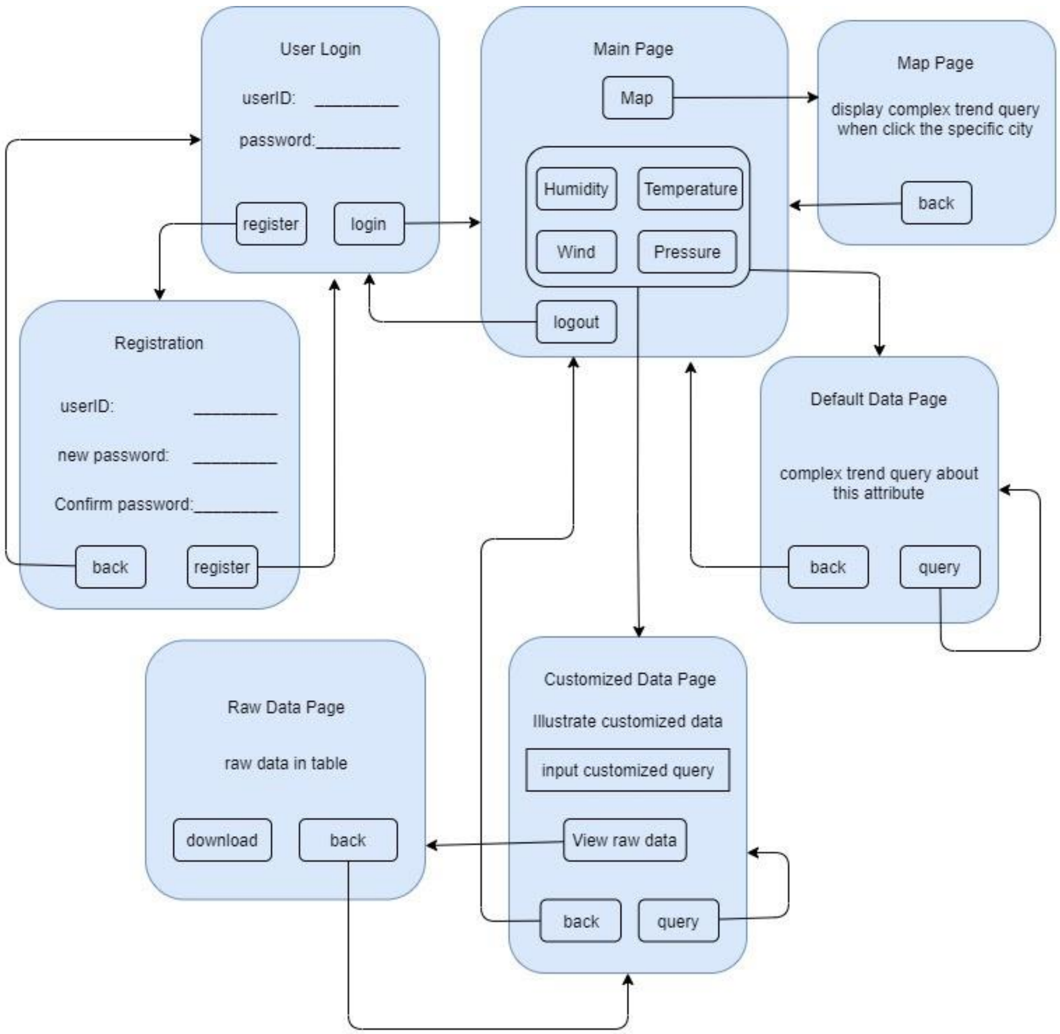

Figure 1: Flow charts of User Interface

Conceptual Database Design

Here are the first three records of our source data. Based on the dataset, we will illustrate how we define the entity sets, relationship sets and attributes.

Extracted from the original dataset, these tables demonstrate some examples of each attributes. In these tables, each of attributes has a row of datetime since the weather is hourly changing in different cities. In the table of another entity(city), there is identity by name, country, altitude and longitude. Also, we make another identity for weather type. Since there is an enormous number of examples, we would like to take few from the dataset to discover the entity-relationships among these tables.

Humidity			
Datetime	Vancouver	Portland	San Francisco
2012-10-01 13:00:00	76.0	81.0	88.0
2012-10-01 14:00:00	76.0	80.0	87.0
2012-10-01 15:00:00	76.0	80.0	86.0

Pressure			
Datetime	Vancouver	Portland	San Francisco
2012-10-01 13:00:00	null	1024.0	1009.0
2012-10-01 14:00:00	null	1024.0	1009.0
2012-10-01 15:00:00	null	1024.0	1009.0

Temperature			
Datetime	Vancouver	Portland	San Francisco
2012-10-01 13:00:00	284.63	282.08	289.48
2012-10-01 14:00:00	284.62904131	282.083251974	289.474992813
2012-10-01 15:00:00	284.626997923	282.091866475	289.460618112

Wind_direction			
Datetime	Vancouver	Portland	San Francisco
2012-10-01 13:00:00	0.0	0.0	150.0
2012-10-01 14:00:00	6.0	4.0	147.0
2012-10-01 15:00:00	20.0	18.0	141.0

Wind_speed			
Datetime	Vancouver	Portland	San Francisco
2012-10-01 13:00:00	0.0	0.0	2.0
2012-10-01 14:00:00	0.0	0.0	2.0
2012-10-01 15:00:00	0.0	0.0	2.0

City_attributes				
CID	City_name	Country	Latitude	Longitude
CID1	Vancouver	Canada	49.24966	-123.119339
CID2	Porland	United States	45.523449	-122.676208
CID3	San Francisco	United States	37.774929	-122.419418

Weather_description	
WID	Weather Description
WID1	mist
WID2	broken clouds
WID	light rain

An Entity-Relationship model (This is shown in Figure 2) illustrates the specific relationship among entities of our database. According to the original dataset, we formalized two main entities of primary keys: users and climate records. For the entity of the user, all of the users can look up climate records and there are four attributes: UID, Password, E-mail, and Profile-photo. Besides, there is one special type of user, defined as an Administrator, which is entitled to update the climate records. Another primary key(climate records) has several attributes: Climate-ID (as for different identifications of climate), CID (ID of the city), and date-time, humidity, pressure, temperature, and wind (attributes: direction and speed). In addition, climate records have a relationship with a foreign key: weather description, which has two attributes: weather types and WID). Since cities are significant elements in the original data set, we decided to define that the climate records are located in different cities. Apparently, the entity(city), has four attributes: CID, name, country, latitude, and longitude.

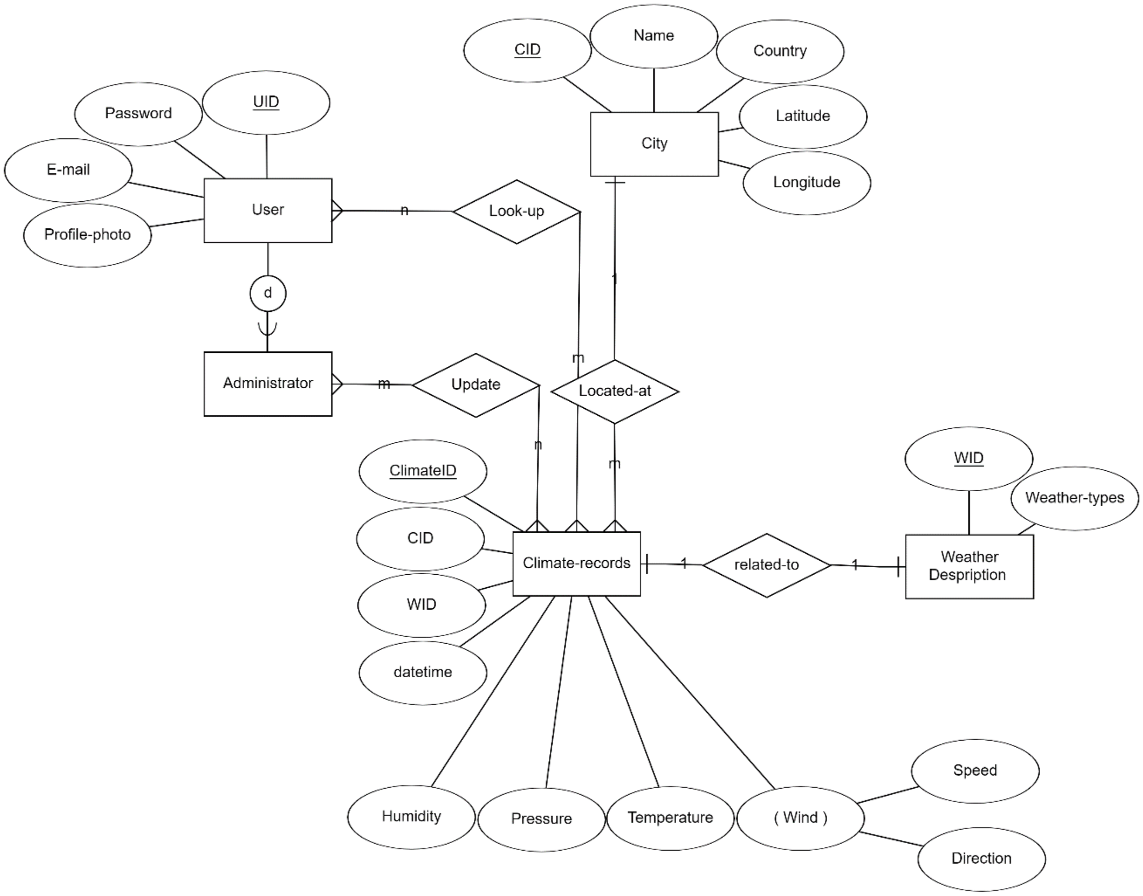

Figure 2: Entity-Relationship Diagram