Date: 10th October 2015

Time: 11.45am – 6.45pm

Venue: SIS GSR 3-4

Attendees: All

Agenda:

* Research for Iteration 3 functionalities

The following was decided after much discussion:

**Algorithm:**

**Smartphone Usage Heatmap**

* Retrieve apps based on timestamp and semantic place –getSmartphoneHeatMap (timestamp, semanticPlace)
* Within the 15-minute processing window, loop through the apps and store inside HashMap<String, String> for gathering the student’s unique mac address and the latest semantic place that he/she has been through, “considering that the user is at the most recent location for the whole duration”. Key: MacAddress, Value: SemanticPlace
* Parse ValueSet of the HashMap into ArrayList<String> for sorting into alphabetical order.
* For each MacAddress in the HashMap, run through the ArrayList and see if there is matching records with the list. If macAddress doesn't exist in List<App>, remove from HashMap of macAddresses and last reported location.
* For each semantic place, calculate the crowd density and display the crowd density per semantic place.

**Social Activeness Report**

For the first part of the report:

* We will retrieve by student’s mac address, start date, end date and calculate the app usage time based on “Social” category. This will be primarily be modelled based on Basic App calculation, using the gregorian calendar.

For the second part of the report:

* From the location retrieved based on the student’s mac address, we are interested to know who else is with the student by retrieveLocationByStartEndDateAndMacAdd(startDate, EndDate, macAddress).
* In relation to others, we will use the user’s start time and end time as reference, while sending query to retrieve other users’ timestamp present in the semantic place.
* Split the input of start time and end time into 15-minutes interval window.
* We will need to ascertain the start time of the other users before the start time of the user, to check if the timestamp of the first other user is his/her starting or ending time.
* For each period of time, retrieveLocationByStartEndDateAndSemanticPlaceOrderByMac, we will retrieve out the login user and other users present in the semantic place. Subsequently, we will calculate the location time spent per semantic place.
* Next, we will retrieveLocationByStartEndDateAndSemanticPlaceOrderByTimestamp, to figure out the alone time and social time spent together with other users, based on the time gaps found in the timestamp.
* retrieveAppDAOByStartEndDate: for each list of the apps, we will calculate the app usage time between the current App and the next App.
* Calulcate the percentage of alone time and social time, based on the total time spent in the SIS building

**Algorithm for Bootstrap with location data**

**Loading Location Data**

1. Wipe database table
2. Validate the csv files
3. Load the csv files into database

Wipe table

All databases (App, AppLookUp, Demographic, Location, LocationLookUp) has to be deleted using DAO delete function.

Validate csv files

The only possible valid combinations of files upload:

3 csvs (App, AppLookUp, Demographic)

5 csvs (App, AppLookUp, Demographic, LocationLookUp, Location)

6 csvs (App, AppLookUp, Demographic, LocationLookUp, Location, Location-delete)

Always validate LocationLookUp before Location csv file.

Validate it using InputStreamReader and CSVReader.

Check for all the appropriate headers in each csv files. Make sure that there is no missing fields for the required fields. If there is missing fields, add it into an errorList to display.

LocationLookUp

Invalid location id: if the location-id is not a valid positive non-zero integer value.

Invalid semantic place: format "SMUSISL<level number><specific location>" or "SMUSISB<level number><specific location>", where level number is a single digit positive non-zero integer, and specific location is a string of length >= 1 (case-insensitive)

Location

Validate using regex

Invalid location: if the location-id is not one of the valid location-id in the location-lookup file

Invalid mac address: if the value is not a SHA-1 hash value (a hexadecimal number, 40 digits long)

Invalid timestamp: if the date & time is not of the correct format: YYYY-MM-DD HH:MM:SS

Duplicate row: if there is already an existing record with values: <timestamp>, <mac-address>. The latest row (i.e having the largest row number) will be loaded into the database and the previous row(s) will be discarded with this error message

If there is no missing fields, it will continue to validate for the correct data inputs such as invalid location, invalid macaddress, invalid timestamp, duplicate row.

Load the csv files into database

Upload the all the data which clear the validation into the database by doing a batch commit. It will upload 30 records at a time.

**Deletion of Location Data**

Check if location-delete csv file exists. If location-delete.csv exists, validate the location by iterating through the records to validate if it matches the location.csv format.

If there is an empty location-id with the valid data of timestamp and mac-address that exists in the current database, it will be deleted successfully. This is also the same if there is a valid location-id is present.

Data will be deleted line by line.

To be completed:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | **Due Date** |
| 1  2 | Smartphone overuse and social activeness SD  Loading and deletion of location data SD | Shing Hei, Amos  Remy, Chu Qian | 11 October  11 October |
|  |  |  |  |

The meeting was adjourned at 6.45 PM.

Prepared by,

Yap Swee Lian, Jennifer

Vetted and edited by,

Koh Chu Qian