2(b):

The aim of this question is to find venues the user visited in last several days, the main strategy to solve this problem is shown below: Firstly, I used the search/tweets API to get the user’s tweets with Swarmapp check-in messages; then, find use split() to find the check-in id from user’s tweets; after that, use foursquare checkins resolve API(https://api.foursquare.com/v2/checkins/resolve) to get venue name through the check-in id we got.

**Issues:**

There are so many Issues I met when I was doing this problem; firstly, I need to look for a correctly API from both twitter and foursquare API documentations to get desire date. Secondly，I had to analysis data I got to retrieve the desire information(ie. Check-in Id),,after that, the largest issue is that some users cannot find through Twitter and foursquare API, so that it makes a large challenge during testing the system.

Design choice:

For overcoming the problem of searching user’s tweets with check-in information, I used search/tweets API swarmapp/com/c/ as query and set parameter “from ” be a given user, and I used setDate() to get the days we need.

For finding out all the user’s tweets with check-in message, I split ‘expanded\_url’ properties of tweets to get check-in ids because it is easy to find that all the check in messages are shown by the format swarmapp/com/c/check-in ID,After that,I used foursquare API 'https://api.foursquare.com/v2/checkins/resolve' to get venues name.

For storing the information, I built two tables in the database which are named “users Information”(which used to store “user name”, ”user id”, ”user location” and “user’s description”) and ”locationVisited”(which used to store user id, venue id and venue name) ,I design them with the concept of relational database, I used “user id” in the table “usersInformation” as the foreign key in the table “locationVisited”.

Advantage:

the advantage of this system is that it can basically implement all the functionalities of assignment demand.

Disadvantage:

Firstly the system is not so robust and always suffers from all kinds of issues, such as some users cannot find through this system.

Secondly, the system is not so efficiency and always cannot retrieve some user when processing.

Finally, the database is not so efficiency when accessing data.