LET: Local Enquiry Terminal

Da Zhang, Bowen Zhi
Department of Electronic Engineering
Shanghai Jiao Tong University Shanghai, China
Email: johnson92104@gmail.com,zhibowen@126.com

***Abstract*—Crowd sourcing which is characterized by the philosophy "borrowing brain throughout the world" has given birth to a variety of business models. Also, the popularization of GPS-based mobile devices provide us significant opportunity to design applications based on Location Based Service(LBS). In daily life, people may want to get resources from somewhere which is very far away from his current location. There is the possibility that he may get it or not after taking a long distance journey, namely, the uncertainty causes inefficiency. In order to deal with such kind of problems, we design and implement LET-Local Enquiry Terminal combined with Social Network Service(SNS) and Location Based Service(LBS).**

I. INTRODUCTION

With the rapid development of WEB technology 2.0, interactive and collaborative ways of experiences have occupied more and more important role in todays telecommunication society. Millions of people are using social network tools and interactive chatting applications to connect, meet and share. Different form the previous unidirectional communicative patterns, the interaction and cooperation provides people with more selectivity, autonomy, efficiency and less cost. Since the ubiquitous concept of crowd sourcing first published at 2006, network has brought significant changes to our world and lives tremendously and quietly. Meanwhile, booming market of mobile devices offer us numerous possibilities and opportunities to further our design and innovation. Currently widely used mobile terminals is not only capable of macro-scale parallel computation as traditional PC laptops, but also component for offering unprecedented Internet mobility. In conclusion, the connection between real world and virtual network has become tighter, deeper and closer.

People would depend more and more on mobile network terminal to solve complicated problems in our daily life. With the basic structure of cooperation system, people have solved significant difficult problems. Researchers designed a smart taxi net to help match taxis and passengers more effectively[1], a research group in SJTU also develops a collaborative download system to increase data transmit rate[2]. However, lots of complicated problems could be solved through such ways, here comes an example:

Mike gets up early in the morning and wants to find somewhere available in the gym to play badminton with his classmates. However, the exercise playground were always full of participants at campus because there are numerous sports lover there. Therefore, Mike does not know whether there is some playgrounds available before he goes there. So, if he gets well prepared and find there is no place to play after a long way journey to several gyms. It would be disappointing for him because he wastes a lot of time and gets a bad mood at the end. But result could be extremely different if he could get some information in advance. Here are some practical solutions:

- *Solution 1:* A phone call to his friend might be a good choice if some of his friends are already in the gym or somewhere nearby at that moment. But this solution is not such feasible because no one could be lucky enough to expect a friend at any destination at any moment.
- *Solution 2:* A published message on some SNS websites such as Facebook or RenRen could be soluble because some of your friends could give you drawbacks after seeing that message. That could be excellent at first glance, but people might reply to you several hours later which can not guarantee real time services.
- *Solution 3:* Asking the administrator could do some help before you go there. This solution seems suitable for this situation but not so general for other scenarios. Because it is probable that no one is responsible for administrating.

Thinking about similar scenarios in our daily lives, we may want to take a campus bus but do not know whether the bus stop is crowded or not, we may also want to ask your teacher for help but do not know whether he is at the office building. The common features of such scenarios is that we do not know whether the resource is available at a relatively far destination, the uncertainty causes inefficiency.

In this report, we aim to tackle with such complicated problems with our innovative system design. We design, implement, evaluate and improve an Android smart phone collaborative system to help get real-time information and make efficient decisions. We combine location based services and social network services together to give users high quality of service. To our best knowledge, our system is first presented and implemented with mobile phones. Nevertheless, in order to make a well-functional practical system, we create and design several mechanisms and algorithms in our system implementation.

The main contributions of this report are as follows:

- proposing a new real-time serving problems popular in daily lives and providing a new system architecture to tackle this problems. Opening a new door with real-time social network service.
- demonstrating a novel practical Android phone application to provide people real-time services and evaluating properties of our system in campus.

Fig. 1. System Architecture. Basic operation mode is shown above, we use location information to fix every user and a centralized server to forward messages

The rest of this report is organized as follows. Section II will introduce the critical problem of local enquiry system, the motivation and challenge associated with it. In section III, we will provide the details of our system design for both server and mobile terminals and its operating mechanisms in this system. Further implementation details and 3 specific technical problems would be discussed in section IV. Related works would be discussed in section V.

II. MOTIVATION AND SYSTEM REQUIREMENTS

The motivation of our system design and basic requirements will be discussed in this section.

A. Motivation

Do you often feel about the inefficiency caused by information barrier when you get confused standing in a long waiting queue full of disappointment? In fact, people all have such problems in today's society due to high development pace and eagerness for time saving. People with high correlation would possibly care about same information at one time and would give help to others with high priority. Such situations exist commonly in today's campus, office building, pleasure ground or even supermarkets. The common knowledge is that people there would have similar purposes and willing to help others in order to get help in the future. Also, another basic assumption is that every single problem could be solved with the intelligence of such correlative people. In our report, we define the area discussed above as local area because people would eager to cooperate at such places.

Providing each participants with necessary information could aid their decisions, which presents to be of huge benefits to both sides. In order to successfully tackle this problem, we intend to design a public intelligent platform, which provides real-time location of each user in our system, and a fair schedule for both. The system design and implementation presents basic asking and answering procedure shown in Fig.1, users get necessary information through cooperation of offering and asking for help. The system design and implementation has several research problems which will be discussed in next section.

B. System requirements

The research challenges includes: 1)design a mechanism to provide users with incentives to participate in our system in order to guarantee the basic cooperation structure. 2)optimizing GPS location services to self-adapting in our system to be a power-efficient mobile system. 3)detecting and ruling out cheating procedures in our system to provide best QoS services.

In our work, we intend to overcome these underlying challenges to design and implement a stable and intelligent system, which could provide far more beneficial service with instant location information. The design goals and system requirements include,

- Giving users virtual credits to provide sufficient motivation in participating in our system, providing them with necessary location information to help them find the best question answerer.
- Providing instant real-time information matching schedule for question asker and question answerer to cut the askers' waiting time and save answer's precious time.
- Analyzing different patterns of moving users in our

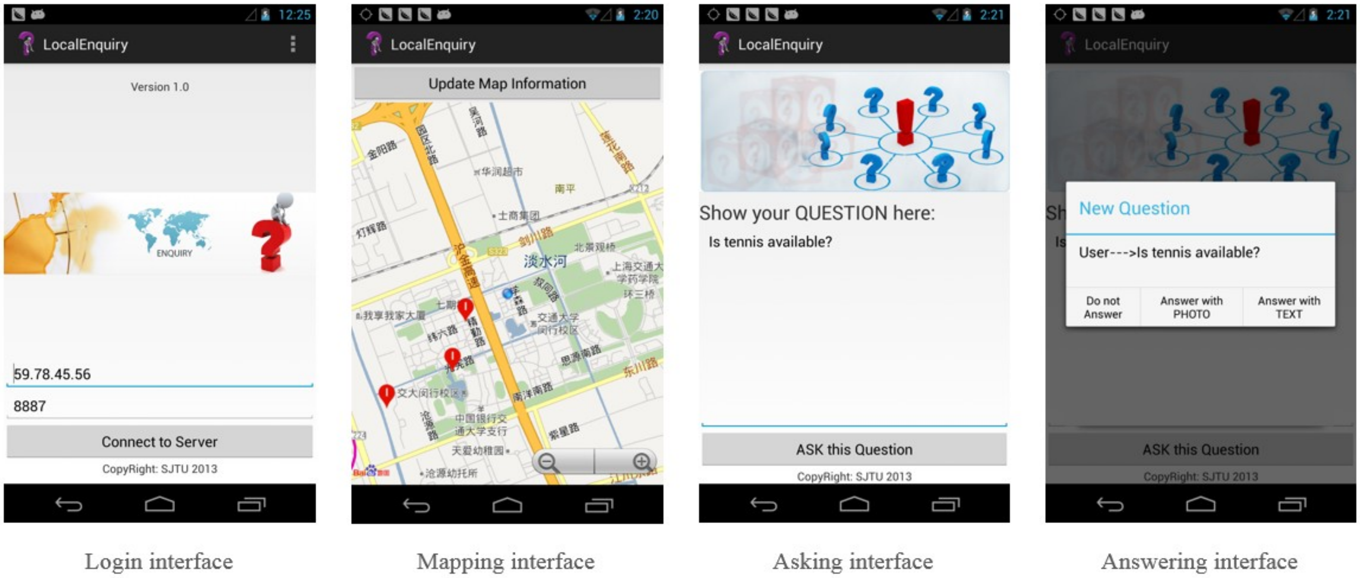

Fig. 2. User Interface. Basic interface are shown in the figure above

system to divide them into several different groups which will be adapted with different location updating frequencies in order to implement an energy-saving system.

- Providing the Minority is subordinate to Majority algorithm in our system and implement that algorithm in practical situations.
- simulation our system in daily use and improving insufficient details.

III. Basic Design

In this section, we will discuss about the basic architecture and software design method including Android application design strategies. The diagram of system design is shown in Fig.3 as follows.

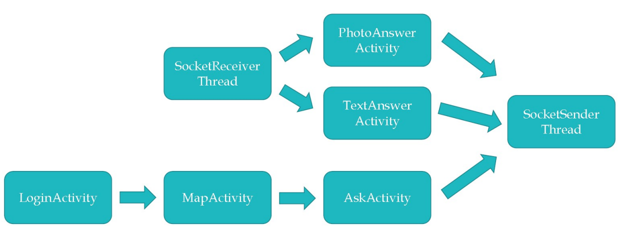

Fig. 3. Basic system design structure

The following section will be divided into four subsections: 1)some specific terms will be introduced in the first subsection to give a brief introduction of LET system. 2)Location service will provide specific details about interface we use in our GPS services and how it can help to find proper answerer. 3)Asking and answering service will demonstrate how to implement real-time functions in our system and how it performs. 4)Server side application will enumerate how server functions in our LET system. Each functional interface is shown in Fig.2.

A. Specific definition

In the following reports, we will discuss some details about our implementation strategies and design. In order to make our reports brief and convictive, we will give some abbreviations and definitions at this subsection.

Our system will be abbreviated as LET for local enquiry system in the following report. The users in our system can be divided into two groups , namely, question asking users and question answering users. The users who ask questions is the primary user(PU) and the users who answer questions is the assistant users(AU).

B. Map and location service

Location based service is necessary in implementing our system, when users log in our application a map containing his current location and other users' locations will be displayed on the screen. This basic procedure is needed because the answerer he choose is dependent on where he is. For example, if we want to play badminton in the gym, we should surly find someone near the gym proper to send a question.

In implementing location based services, we simply utilize map API from Baidu Inc. and follow the user guide provided online. The procedure could be simply shown as follow steps.

Map and location service operating procedures:

Step 1: The PUs and AUs update their current location messages to centralized server as soon as they log in.

Step 2: With location information collected above, the centralized server compact them and broadcast it to all alive users.

Step 3: The map on the PU or AU side will be updated. The updated map not only includes user's current location(marked with a blue point) but also includes other users' current location(marked with a red circle).

Step 4: If an user has quitted or could not be available for a long time, the location information will be erased. And when others click *Update location information* Button, new map will be updated.

C. Asking and answering service

In the above section, we discuss about the implementation of LBS and the necessity of LBS in our LET system. In this section, we will further one step to designing basic social network service structure and methods to providing real-time services.

Our LET system which is shorted for local enquiry terminal applies basic procedures of asking and answering questions to provide sufficient information for PUs. The basic operating procedure could be simply shown as follows.

Asking and answering service operating procedures:

Step 1: PU logs in the map service interface and he has a prepared question in his mind. For example, in our previous example Mike wants to play badminton and he wants to know whether gym is available or not.

Step 2: After updating location information, AUs will be displayed and marked with special icons on map which is discussed in map and location service. PU will choose someone near the destination to send this question. In previous example, the destination would be the gym.

Step 3: PU will type and send this question to selected AU, then socket will be builded and running the asking and answering procedure.

Step 4: When AU gives the answer back after few minutes and after several authentication methods, credit will be allocated and process is closed.

There are two points we should make specifically in this service:

- *conversion between AU and PU:* In our LET system, the differences between AUs and PUs are not so obvious because of the basic cooperation properties. In one procedure, someone is identified as a PU and others acting as AU, however, in other scenarios they will play reversal roles. All users are AUs in this system.
- *real-time service:* The most significant differences among our system and typical SNS applications is that we provide real-time service. This is implemented in our software structure. If someone is running our application and a new question arrived, a dialog interface will pop immediately waiting for users to give a response-to ask or not. This means that we give high priority to our application in mobile phones.

The credit mechanism and some cheat-proof protocol will be discussed in following sections.

D. Server side service

Server is the centralized controller to offer infrastructure network, server is the brain acting as a key role in our LET system. Without such a server, whole functions will be disabled. The functionalities should include:

- *information storage:* server should store every users' location information simply as a XML file and when users need to get updated map information, he could simply download this XML file.
- *transmission medium:* Our network structure is running in infrastructure mode and need a central AP to transmit information and server acts as AP in our system.
- *central controller:* There are many factors we should limit and control, such as calculate and provide virtual credits for users, adapting different updating interval and help to detect and avoid cheating in our system.

IV. FUNCTIONAL MECHANISM

A. Credit bonus mechanism

As we can see, no one would like to service for the whole system for free. So, it's necessary to add the credit bonus mechanism in our system. We would like to imitate the mechanism of Bitcoin. Bitcoin is a cryptocurrency first described in a 2008 paper by pseudonymous developer Satoshi Nakamoto, who called it a peer-to-peer, electronic cash system. And In [3], the role of Bitcoin is defined as followed:

The processing of Bitcoin transactions is secured by servers called bitcoin miners. These servers communicate over an internet-based network and confirm transactions by adding them to a ledger which is updated and archived periodically using peer-to-peer filesharing technology.In addition to archiving transactions, each new ledger update creates some newly minted bitcoins. The number of new bitcoins created in each update is halved every 4 years until the year 2140 when this number will round down to zero. At that time no more bitcoins will be added into circulation and the total number of bitcoins will have reached a maximum of 21 million bitcoins.

So, our system is similar to the peer-to-peer system. The action of answering question and even keeping the system online has the same function of mining in the system of BitCoin. And the use of this new bonus in our system is for transaction and asking questions. That's the reason and motivation of people participating in our system.

Meanwhile, Bitcoin is an open source P2P digital currency, and the protocol of Bitcoin system is easy to get from the Internet. Therefore, we use the similar algorithm, in which the speed of creating new bonus is improved, so that the credit bonus system might work as well as the Bitcoin system.

In our system, the creating process of bonus is like this:

step 1. New transactions are broadcast to all nodes.

step 2. Each user node collects new transactions into a block.

step 3. Each user node works on finding a difficult proof-of-work for its block.

step 4. When a node finds a proof-of-work, it broadcasts the block to all nodes.

step 5. New coins are successfully collected by the receiving node which found the proof-of-work.

step 6. Nodes accept the block only if all transactions in it are valid and not already spent.

step 7. Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

step 8. Repeat.

The hash table used in our system is also similar to the one used in the Bitcoin, however considering the capacity of mobile device is limited, the size of hash is smaller than 256 and will be calculated in a virtual way in cloud.

In server side, two consecutive H-256 (provisionally) hashes are used for transaction verification. R-160 is used after a hash for bonus digital signatures or ”addresses”. A bonus address is the hash of an ECDSA public-key, computed as follows:

Address/Public-key = Version+R-160(H-256(public key))
Checksum = 1st 4 bytes of H-256(H-256(Key hash))
Address = Base58Encode(Key hash+Checksum)

It’s the method of calculating in Bitcoin, however it’s suitable for our bonus system. And the only differences are that the size of hash is smaller to make it easy to create bonus and the computing work is based on the virtual computer in sever not the mobile device. In this way, we encourage users to keep themselves online to answering question, rather than wasting the energy of their mobile phone to calculate the hash table.

B. Self-adaptive GPS algorithm

As a common knowledge, location based services especially GPS service waste lots of energy. In our system, we should think about how to avoid energy wasting and be more power-efficient. In our LET system, we design and implement a self-adaptive location algorithm which is aimed at adapting different location interval to different users. In order to implement that, we should find a basic standard to define different users. The procedure is discussed as follows.

At first, each user will be initialized with location updating interval T_0 which means that every smart phone will turn on their GPS service with period T_0 and immediately update this location information $L_0(x_0, y_0)$ to centralized server.

Fig. 4. Different user patterns

At second step, the centralized server will trace every single node i to calculate factor moving rate M_{ri}, which can be simply calculated as

$$M_{ri} = \frac{\sqrt{(x_{ij} - x_{i0})^2 + (y_{ij} - y_{i0})^2}}{T_i}$$

where T_i is the current updating interval which is initialized as T_0 and $L_{ij}(x_{ij}, y_{ij})$ is the $j-th$ location information(current) for node i.

Then, the centralized server should estimate an moving rate threshold . The moving rate threshold is influenced by many factors including feedback QoS and users’ remaining power and so on. Thus server could calculate the general moving rate threshold Mt_r on the basis of general variables. However, for each node the threshold could not be the same value because of several differences among different phones. Thus the threshold for every node is Mt_{ri} which is modified through Mt_r.

Finally, the server should compare the current moving rate M_{ri} and moving rate threshold Mt_{ri} for each user. If $Mt_{ri} > M_{ri}$ which means that the user moves too slowly, thus the M_{ri} should be adapted which indicates changing T_i in order to let M_{ri} approaches Mt_{ri}. This is also suitable when $Mt_{ri} < M_{ri}$. Thus every node will have a balanced interval T_b for adapting T_i several times.

Different moving patterns are shown in Fig.4. After self-adapting procedures, each mobile terminal need not to turn on their GPS every second. In contrast, they could find a proper interval to best save their power and enjoy high quality of service.

C. Cheat-proof mechanism

Another influential interference in cooperation system is cheating, even if we assume that every user is willing to participate in our LET, cheaters also act as a fatal problem because people probably do not know the truth and give back some false information back. The situations will be worse in our daily lives because other bad users will be included:

- Users who is no responsible and will answer with irresponsible answers.
- Users who intend to destroy our system thus give bad answers.
- Users who wants to get virtual credits but do not want to participate.

Fig. 5. Different user patterns

We design and implement *Minority is subordinate to Majority* mechanism to help cheat-proof. In our basic structure, we implement point-to-point communication which only need one PU and one AU to participate, however, in order to avoid cheating behaviors we need to further our cooperation mechanism. In cheat-proof structure, one PU and several AUs will participate. When a PU sends a question to his selected AU, the centralized server will broadcast the message to several nearby AUs and get all possible answers back.

We should obey following rules to select help users:
Step 1: Decide the number of help users needed in cheat-proof N_h, N_h should not be too large which will cause wasteful network flows nor too low which possibly could not avoid cheaters.
Step 2: Choose users nearby which satisfies to minimun

$$e = \frac{\sum_{i=1}^{N_h} D_i}{N_h}$$

Step 3: The same question will be sent to all of those AUs and get answers back to server.
Step 4: The server will check the answers and identify the answer chosen by most users the best answer, this answer will be sent back to PU.
Step 5: Credits and be automatically allocated among them.

However, it is not so simple for server to check many complicated answers because they are all created with different formats by different AUs. Here are two ways to tackle this problem, first one is to limit the format of question which may possibly require PU's question could be answered with YES/NO. Second one is to offer semantics algorithm, this is not implemented right now but included in our future works.

V. Related Work

There has been a growing concern on the research on system humanization and intellectualization . We would like to focus on the sematic mechanism to solve the problem that the format of question is inflexible and the standard of judgement is unreliable.

The addition of intelligent sematic system will bring the new life for the present one, ensuring that the energy will be saved by identifying and combination of those similar question. Users have the opportunity to raise some flexible question beyond the normal format of pressing the button for choosing the fixed question. Meanwhile, the timeliness and classification management will be more powerful and efficient, using the technical of extracting key words in sematic system.

In addition, the analysis of user searching history provide our a possible way to realize the deep learning in our system. Based on the user's information of key words and location counting in the search history, the system will get aware of the deep need of user and choose the right answer as the willing and judgement of users.

There are also some challenges and problems in the processing of realizing the sematic part and deep learning part. First, the different users have the different customs of using language, and the normal sematic system may not be suitable for the Q&A pattern-based system, of which the performance will be unstable and unreliable. Our solution is that using the method of choosing the button and the method of semantics in the same time. When users apply their questions to the system in the semantic way, they have to press the button to make sure the system understand them, by choosing the right one among some choice provided by the system, which are some possible meaning in its sematic. Second, the history of users' searching and location is the privacy of users. So, we have to establish the privacy protecting mechanisms to identifying the privacy of information,avoiding giving away some personal information of users from the data history.

VI. Conclusion

In this report, we propose a new technical problem in application layer and we design, implement and evaluate a novel smart phone application structure to tackle this problem. The basic design procedures and implementation method have been discussed in detail above. We also design and apply three different algorithms to improve our system functionality and we evaluate the operation of our system in real world. Related works and future development are included in our report.

References

[1] Y. Yuan, K. Miao, D. Zhang, L. Sun, C. Chen*"An OSGi-Based Smart Taxi Service Platform"*, Services Computing Conference (APSCC), IEEE, Dec 2012.

[2] Dandapat, S.K. ; Niranjan, R. ; Ganguly, N.*"Framework for Collaborative Download in Wireless Mobile Environment"*, Mobile Data Management (MDM), IEEE. 2012.

[3] Wikipedia,the free encyclopedia, Bitcoin Open source P2P digital currency,2013