Human Computation Based Platform for Citizen Services in Smart Cities

Adnan Yahya[0000-0002-8769-1961] and Yazan Yahya and Nibras Misk and Hamzah Hijja
Electrical and Computer Engineering Department, Birzeit University, Birzeit, Palestine
yahya@birzeit

Abstract. We describe a platform that utilizes the power of human computation to provide improved services to citizens in a smart city context. The platform enables users to report problems to authorities and facilitates communication between parties to work on solutions to encountered problems. An important aim is to involve citizens in the city decision-making process. System users can be registered who provide necessary information directly or through their social networking accounts. Alternatively, users can proceed as guests and use the system anonymously, without registration. For registered users the profile information and use history can be used to strengthen the validity/trust of their contributions. Users can perform several actions on the platform including: reporting a problem using username or anonymously, voting on reports submitted by others, reviewing the status of their reports and responding to requests to perform specific tasks on their travel route. We describe the system and focus on human computation aspects, including: user privacy issues, data collection methods, data reliability and data quality assessment, user engagement incentives and reward system, models of interaction between the user and the system (Push/Pull) and mention frameworks used for system implementation and testing. Although focused on citizen services, the platform is flexible and extensible to tasks in other areas.

Keywords: Smart City Services, Human Computation, Citizens Services, Data-Driven Services.

1 Introduction and Background

1.1 Motivation

The huge technological advances are affecting all aspects of life and many daily activities are increasingly employing modern technological solutions for solving problems. Several factors encouraged us to investigate the use of human computations for better citizens services in a smart city context. Human Computation (HC) is defined as the approach to solve computational tasks, particularly those tasks that are hard to automate, through the contribution of humans[14]. Technology supported city services are still weak in our country. The main aim of the HC platform is to strengthen communications between citizens and authorities to enhance collaboration on reporting and solving different kinds of problems affecting citizens life and essential for smooth efficient functioning in terms of time and cost. Another aspect is raising the trust level between citizens and responsible authorities and allowing citizens to be informed about city activities in a timely and transparent manner. A solid relationship between citizens and authorities makes users important and active link in the services–governance chain. We wanted to employ the concept of CrowdSourcing which allows the integration of human effort into the decision making processes directed at solving critical problems. Combined with HC elements, CrowdSourcing is essential when dealing with problems that even computer systems cannot solve easily.

1.2 Related Systems

Human Computations for improved citizen services are becoming an important approach to turn to. Here are some examples:

- **Youknow:** a social networking platform for problem solving and deployment solutions. It allows one to communicate with decision makers, based on the principle: "You have fair rights and you have duties”. In Youknow users specify to whom the problem is directed. Decision makers look at these problems, reply to them and make a decision. The platform was launched March 1, 2015[12].
- **City Smart Services:** a cross-platform application, designed for municipalities. Citizens use the application to submit public issues to the municipality. Municipality receives the reports and assigns cases to available technicians to solve the issues raised. The technician receives a notification about the case, accepts the job and fixes the problem on location and declares the problem solved[13].

1.3 Distinguishing Features of our Platform

After reviewing the existing systems, we strived to have our system posses features that make it appealing to potential users: individuals and institutional. Here is a summary:

- Supports the concept of human computation, things computer technologies cannot do well, and a single user cannot adequately report the problem so we let users do it as community and their aggregate input is the main system input[1,2].
- Outsources work by encouraging the contribution of members of the public (community) of tasks that are usually handled by institutions.
- Takes data trustworthiness in high consideration and has rules and acceptance criteria for user data that affects and is affected by user trustworthiness.
- Is cross-platform: the resulting app runs on both iOS and Android devices.
- It works in multiple modes: people *volunteer* to report but are also *asked* to report. The system uses two-way communications between users and authorities: users report to authorities and authorities ask users for their input on problems, which makes our approach more engaging and democratic. Users can also monitor the progress of their reports (status of the report: submitted, in progress, solved).
- The system is extensible: it easily tailored for other applications of similar nature, there is no need to rewrite the code for additional applications of similar nature. Potential applications are by no means limited to municipalities: one can easily think

of applications like monitoring wireless connectivity in a smart city setting, monitoring water availability and leakage problems in a town.
- Care for the user: user information is in safe hands. The passwords are saved in encrypted form and the user is always aware of the current privacy settings and have the option of anonymous reporting.

2 System Description and Design

In this section we describe the system we implemented to utilize HC for improved municipal services; its components, functionality and distinguishing features.

2.1 User Profiling

User Profiling is the process of collecting information about the user. This information can be used to understand users and appreciate their interests/capabilities for enhancing the user input and assessing its quality and for providing better user experience/satisfaction[4]. The main approaches of user profiling are:

- *Explicit User Profiling:* also called Static Profiling. In this approach, the user profile is constructed from user's data one gets by filling forms. There are some problems when we only depend on explicit profiling as users are reluctant to share their actual information out of concern about privacy or because the form filling process might be boring so the user may try to avoid it or provide incorrect information[4].
- *Implicit User Profiling*: also called Profile Extraction. Here one extracts the needed user information from different sources, such as web pages or social media platforms like Twitter, Google plus and Facebook[4]. Another source is user behavior-based techniques which help user profiling systems gather information of interest about users, for example by observing user web navigation patterns.

For our platform, we allow both profiling methods. We give the option to the user to register using explicit profiling, either by filling forms or we extract user information from social media (Facebook or Google) accounts. Furthermore, we allow *Guest* login to the platform, which means using functionalities without registration. However, we decided to restrict the number of daily logins for guest users to discourage spammers. For this purpose, we used Universally Unique IDentifier (UUID), a.k.a. Globally Unique IDentifier (GUID): a 128-bit number used to identify information in computer systems. When generated according to the standard methods, UUIDs are unique for practical purposes, without depending for their uniqueness on a central registration authority or coordination between the parties generating them. While the probability of duplicate UUIDs is not zero, it is close enough to zero to be negligible[3]. Version 1 UUIDs, the most common, combine a MAC address and a timestamp to produce sufficient uniqueness. In the event of multiple UUIDs being generated fast enough that the timestamp doesn't increment before the next generation, the timestamp is manually incremented by 1. If no MAC address is available, or if its presence would be undesirable for privacy reasons, 6 random bytes sourced from a cryptographically secure random number generator may be used instead. Each mobile device is assigned a UUID and we extract the UUID of the guest user's mobile device, independent of the operating system (iOS, Android) and save it in our database allowing us to track guest user, logging into the platform repeatedly to prevent spam or suspicious logins.

2.2 Data Collection Methods

Data collection refers to collecting information from participants to help solving problems. When collecting data from devices, there are two choices, either *push* or *pull*. Push is the process of sending data from devices to server without prompting, while pull means the process of receiving data from the devices to the server by polling them periodically. There is also another choice called hybrid that uses *push* and *pull* together depending on the application context and requirements[5].

Pushing is preferred when the server should send a subset of data to different devices in real time. A disadvantage is that the sent data is limited, so sending a variety of data subsets may bring the model down[5].

Pulling is preferred when the devices will send different types of data to the system. This is a great choice, but also has some disadvantages; first, it may not be in real time, and if it was in real time, it will consume a lot of resources[5]. We settled on a hybrid system, that uses pushing and pulling together. Pulling is used to send user data to the platform; this can be done in real time or delayed if the user is not online, while pushing is used by the platform to send data/requests to users in real time, though reading requests by devices can be delayed if not online.

So, in our system *pulling* is used in the following cases/scenarios:

1. The user reports his problem on the platform.
2. The user provides his route and waits for system requests of nearby problems.
3. The users can vote on the reported problems on the platform.
4. The users can respond to the system requests to check on nearby problems.

Pushing, on the other hand, is used in the following cases:

1. The platform notifies the users of a reported problem nearby her/his home location or along her/his route with a request to confirm that the problem really exists/solved.
2. The platform notifies the user when the status of his report changes either to solved or partially solved or unsolvable.

The System Data Pipeline given the Information flow is as depicted here:

User	➔	**Municipality Personnel**	➔	**Case Delegation**	➔	**Solicit Feedback**	➔	**Case Resolved**	➔	**User Notification**

2.3 Location Awareness

Most modern smart phones come with built-in GPS sensors to get earth coordinates. In our system we use GPS and Google Map API to provide two features:

1. *Location detection:* during the reporting process, the reporter may need to provide location of the problem. For that the device GPS must be enabled.

2. *On_my_route (directions and routing) tasks*: where the system asks the user to define a route by providing a starting point and destination. The system uses Google Directions API to get the route then looks for reported problems within a given distance

from the route for the user to look into them by pushing the problems to the user inbox. The user can access the reports displayed on his/her device and act on them while travelling (Figure 1). For that, we used Google Maps API and the Google Directions API:

Google Maps API: allows users to customize maps with own content and imagery for display on mobile platforms and websites. This API provides four basic map types: roadmap, satellite, hybrid, and terrain[7].

Google Directions API: a service that calculates routes and directions between locations using an HTTP request. This API can calculate directions using one of several modes of transportation, including walking, driving or cycling. It returns directions as a series of waypoints and specifies destinations, origins, and waypoints as strings[8].
The API calculates directions and returns the most efficient routes. The API considers many factors like time, distance, number of turns. Calculating directions is a time and resource intensive process[8]. Once we have the trip route, our system ranks the reported problems by their distance from the travel route and pushes the closest to the user as suggested tasks for possible feedback.

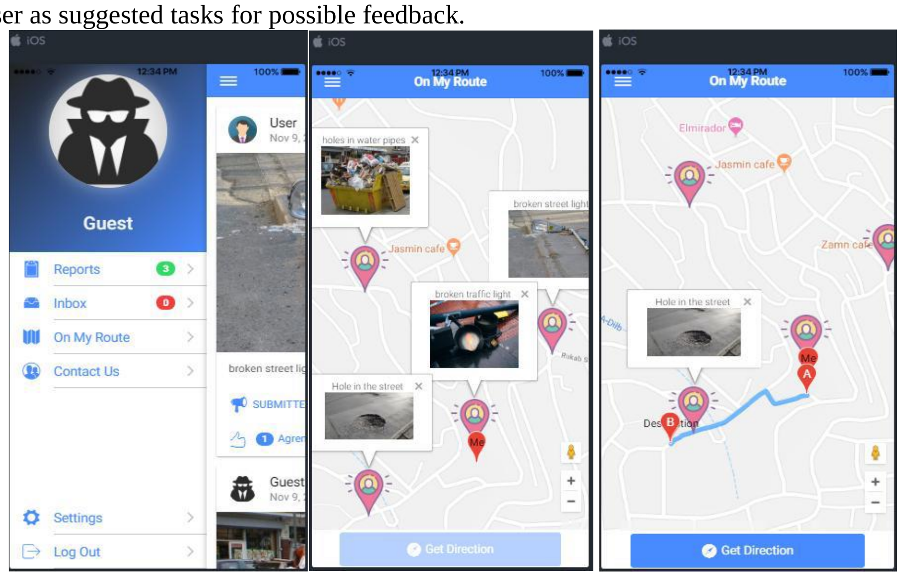

Fig. 1. General and On_My_Route Functionality Screen Shots

2.4 Data Quality Assessment

User provided data may not always be trusted, accurate or even correct. Thus, one needs to make sure that the quality of such data is properly evaluated before incorporation into the decision-making process[16].

Data Trustworthiness: in any Human Computation System, Data Trustworthiness is one of the most important issues6 to consider carefully. Since the platform is dependent on human contribution, certain rules must be set to ensure data trustworthiness and to deal with any errors inherent with information provided by human users. Looking for typical approaches to validate user data, we found the following to be of interest[6]:

1. Using majority vote to assess the correctness of data and ensure trustworthiness.
2. Gathering redundant data from users to check the agreement level of reporting users on the same piece of information. For example, collect multiple problem reports and check if some validate others submitted previously. This approach can be costly and time-consuming.

Evaluation of user credibility based on quality of information provided earlier then assigning trustworthiness levels of data based on user credibility. Acceptance of generated data will be based on the aggregate (weighted sum) of user- supplied data on the same problem. In our platform for Human Computation Services, we adopted the following approach for assessing quality of data provided by users:

1. Notify the platform users' who are in the same geographical location of the reported problem, and ask them to verify the report.
2. Measure the degree of agreement between respondents on the existence of the problem, whether they are registered or non-registered to assign a metric called "Trust Ratio". In case of disagreement between users, we use the Cappa Measure (Inter–Judge Agreement).
3. If a user reports a problem and our system declares it trustworthy after proper evaluation, the user's trustworthiness ranking increases, according to the following formula and his future contributions will be weighted accordingly.

$$\boldsymbol{User\ Ranking} = \frac{\boldsymbol{Number\ of\ Accepted\ Reports}}{\boldsymbol{Total\ Number\ of\ Reports}} \times \mathbf{100\%}$$

4. For municipality services, three major elements increase the quality (trustworthiness) of data if included in the report. These are:
   i. Digital Attachments: Image or Video demonstrating the problem.
   ii. Geographical Tagging: of Location where the problem was observed.
   iii. Time–Stamp: Exact time when the problem was observed.

   The time stamp and geotags are generated automatically and can be attached to the images, even if data transmission is delayed, say due to lack of internet connection.

It is important to stress that data trustworthiness is affected by involved users credibility and that user credibility/rank affects the trustworthiness of the submitted data.

2.5 Assessment of Agreement between Users:

It is essential for Human Computation (HC) results to be reproducible. Additionally, we need the results of any Human Computation process to be of high validity or high utility and better than out-of-chance agreement at minimum[14]. *Reproducibility* can be defined as the extent to which the results of a HC process can be duplicated by different human contributors operating under different conditions or using functionally equivalent metrics. Reproducibility is related to consistency of results, meaning that the more the results are consistent, the higher the chance of reproducibility[14].

Measuring Reproducibility: to have confidence in data obtained in HC tasks we must measure reliability, to make sure the evidence obtained is independent of measuring instrument or person. The following three types of reliability are of interest [14]:

- *Stability:* best described as the extent to which the results of a HC process are stable (i.e. unchanging) over time. Stability is focused on results being consistent, such that if someone repeats the same task over time, the results stay the same.
- *Accuracy:* is the extent to which a HC process outputs high validity results. Measuring accuracy requires comparing data obtained by human contributors against valid answers to prior known questions.
- *Reproducibility:* describes the extent to which a process can be repeated many times by human contributors, under different conditions at different locations.

Reproducibility mainly has two aspects: inter–rater reliability and inter–method reliability. We focused on inter–rater reliability, which is of most concern for HC. One form of inter–rater reliability is percent agreement but for our HC we can use the Cappa Measure (Inter–Judge Agreement)[14].

2.6 Privacy Issues

Many human computation applications are highly dependent on data provided by users, and thus we want to avoid losing users through under-considered or badly documented practices. For highly engaging volunteers in participatory systems, privacy must be supported. Privacy in participatory projects is defined as the right to manage access to personal data contributed voluntarily. Personal data is defined as data that contains identified or identifiable information about the contributor[9].

In our system, we adopt privacy considerations to protect users personal data to encourage users to continue contributing without fear for their data and to protect against possible legal litigation and potential lawsuits. Our system must maintain users privacy, and make privacy principles as clear as possible and visible to all. Some design principles were stated to ensure privacy[10]. We include them in participation terms and give the user the choice to accept or reject participation. These principles are:

- Participant primacy: users personal data will be collected with her/his knowledge and explicit agreement.
- Data legibility: the users will be informed how their data will be used in the system.
- Longitudinal engagement: users will be given the option to change the permissions of sharing their personal data over time if situation changed.

Additionally, based on Privacy in Participatory Research[9], some ethical principles were derived for Participatory Research Design. We opted to adopt four of them here:

- Ethical Engagement: according to our system guidelines and goals, we will identify the ethical principles and take them into consideration in design, implementation and development processes, and post them beside our system policies.
- Informed Participation: detailed explanation of participation as well as legal compliance related issues will be provided to users before accepting any participation.
- Evolving Consent: if participation terms change, the system policies will be changed, and users will be informed promptly by repeating the informed consent process.
- Evolving Choice: we designed the system to support user's choice in response to changing contextual situations. For example, a registered user may choose to share

personal information in order to receive benefits in one geographic location like a public space but reject sharing the same information in own private home.

These principles are offered as guidelines for our system to support stronger relationships between the users and service providers, increase the trust in our system, and encourage user participation by keeping them informed and in control of personal data.

2.7 Reward System

A Reward System is a measure directed at rewarding users to motivate them to participate and use the system. The following clarify the major characteristics of our Reward System:

- Typically, the first and most important reward users get is improved city services in that problems they face in their environment are communicated faster to the authorized parties who can work toward a solution. So, instead of a formal visit to the municipality, users can report the problem whenever they observe it through the platform, without the need for formal, boring procedures.
- Whenever a user successfully reports a problem on the platform, a number of points is added to his/her account, and users with highest numbers of points are elevated to premium users with special benefits from municipality[11].
- To encourage participation, we thought of adding elements such as: the first reporter of a problem gets more points when verified, supported reports get better rewarded, quick responses to verification requests get better rewards, inviting new people to enroll and participate results in reward points and so on.
- The more points users collect, the higher their ranking becomes. We have three user rankings on the system: Gold, Silver and Bronze, and these are accounted for when assessing trustworthiness of the user and user supplied data.

2.8 Users and Roles

We know that users on the platform are either registered or non–registered (anonymous). On our platform, a series of tasks are always available and geotagged to both registered and anonymous users. However, when it comes to voting on a problem report, registered users vote is given greater weight and higher priority and anonymous user vote value is only 10% of that of a registered user. The platform is flexible in that it accepts multiple roles for the same user: a registered user is allowed anonymous access and non-registered users can upgrade to registered users when desired.

The main and most vital role of both types of users is to report problems or collaborate on them. Other roles for users on our platform include: monitoring the status of the report they submitted, including degree of agreement with other users, periodically checking the platform to see if the problem was solved or a feedback is provided.

On the other end, designated users manage and keep track of all the reported problems in their domains and their status: these are Authorized Users like a Municipality. Since each reported problem has at least one division responsible for it, we agreed to give direct Authorized Access to a *super* authorized user in the Municipality who then delegates these rights to the responsible divisions.

The workflow is as follows: once a submitted problem is accepted (based on degree of agreement between reporting), it is the super authorized user responsibility to forward the problem to the responsible division, and from there the division administrator keeps track of the problem and work towards solving it. Problems not solved for a long time are flagged with red or yellow colors, to alert that follow up is needed.

Another point worth mentioning is the fact that our platform is extensible is the sense that any additional tasks and additional roles of users in other fields of human services without the need for major system rewriting.

3 Deployment and Testing

We have fully implemented the mobile application, for both iOS and Android devices in addition to the backend components. For the latter we developed a web portal for authorized parties to login and keep track of the problems under their responsibilities, with static login credentials provided by our side, to login and keep track of the problems under their responsibilities. As mentioned earlier, authorized parties (like municipalities) have the ability to change the report status as work progresses.

For the implementation, after considering available options we settled on: Ionic for cross platform framework and Firebase for cloud services deployment. The functionalities implemented and available to users are: report a problem, monitor the status of reported problems and check to see if any problems exist on the route a user is taking (on-my-way functionality, Figure 1). We have released it on a local server. The interface is bi-lingual (Arabic-English) and the language choice is controlled by individual users. During the development, we worked with a local municipality and a group of volunteers. We also asked some potential users to test the main functionalities and the usability of the application, most of their experiments were successful, and we got positive reactions. The users were appreciative of the potential of the platform for improved services to the city residents and provided useful feedback for future modifications.

4 Conclusions and Possible Extensions:

During the work on the system we faced some challenges. First, is how to encourage citizen users to use the application and our answer was the reward system. Second, is how to trust user data reported on the platform. We agreed to take into account the ranking of the user who had reported and users agreement on the report to decide the degree of trust in the collected data. Third, is how to encourage entities like municipalities to use the application to handle real-life problems reported from real users. Reluctance emanated from concerns regarding many problems being documented and visible to outsiders. Our platform put together the real clients of an organization and the management to help them provide their views and suggestions or talk about the problems facing them and open the way for the organization to view the suggestions and problems of users and work on their solution in a collaborative manner. Fourth, is that the problems encountered by users are now documented, and if not solved, they will be proof that the entity (municipality) failed to cater for its constituents needs, something that may discourage service providers from adopting the system. We tried to emphasize the positive aspects of the system work to allay such fears.

The mobile application is still not released on Google Play Store or Apple Store, but we have tested it for the work of main functionalities as well as usability.

Finally, we believe that our system provides the community with an application that utilizes human computation, allowing citizens from community to participate in decision-making process. From another side, the platform makes local authorities (municipalities for example) aware of any problems with public services (garbage collection, street-lights failure, state of streets, etc..) as soon as encountered by citizens which may help save effort and resources and improve response time[15].

Possible improvements include having the approach work on other systems with additional tasks and additional roles. Processing of textual data for better system performance is another improvement aspect. We are also studying adding the ability to “Go Live” when reporting a problem on the platform so that a user can report a problem and engage in real-time and the possible fallout for data quality. “Go Live” feature consumes extra upload bandwidth and thus depends on the network connection type.

References

1. Wikipedia, *Human-based computation*, https://en.wikipedia.org/wiki/Human-based_computation, Accessed August 2, 2021.
2. Chen, L. *An Overview to Human Computation*, Inst. of Information Science, Taiwan, 2009.
3. Wikipedia, Universally unique identifier. https://en.wikipedia.org/wiki/Universally_unique_identifier. Accessed August 2, 2021.
4. Kanoje, S., Girase, S. and Mukhopadhyay,D. *User Profiling Trends, Techniques and Applications*, International Journal of Advanced Foundation and Research in Computer 1-1, 2014.
5. SevOne, *To Pull or Push IoT Data? That is the Question*, https://www.sevone.com/blog/pull-or-push-data-iot-data-question, Accessed March 11th, 2018.
6. Ipeirotis, P., Paritosh, P.:*Managing Crowdsourced Human Computation*, Hyderabad University, India. 2011.
7. Google,*Maps JavaScript API*, https://developers.google.com/maps/documentation/javascript/tutorial, Accessed May 15, 2018.
8. Google, *Directions API*, https://developers.google.com/maps/documentation/directions/intro, Accessed on May 15, 2018.
9. Bowser, A. and Wiggins, A.: *Privacy in Participatory Research: Advancing Policy to support Human Computation,* Human Computation, vol.2, # 1, pp 19-44, 2015.
10. Shilton, K., Burke, J., Estrin, D., Govindan, R., & Kang, J.:*Designing the personal data stream: Enabling participatory privacy in mobile personal sensing*, Information and Internet Policy, (2009). Washington, USA, pp. 3-4.
11. Goh, D., Than, E. and Lee, C.: *An Investigation of Reward Systems in Human Computation Games*, Nanyang Technological University, Singapore, 2015.
12. Youknow,إنّو بتعرف , http://www.youknow.ps/ , Accessed on March 29th, 2018.
13. City Smart Services, http://citysmartservices.resco.net/ , Accessed on March 13, 2018.
14. Paritosh, P. *Human Computation Must Be Reproducible.* CrowdSearch, pp. 20-25. 2012.
15. OECD. Smart Cities and Inclusive Growth. 2020. https://www.oecd.org/cfe/cities/OECD_Policy_Paper_Smart_Cities_and_Inclusive_Growth.pdf
16. Neupane, C. Wibowo, S. Grandhi, S. Deng, H. A Trust-Based Model for the Adoption of Smart City Technologies in Australian Regional Cities. Sustainability 2021, 13, 9316.