

Cloud Based Mobile Application for Tracking User Activities

^[1] Piyush Choudhary, ^[2] Karan Gaikwad, ^[3] Vinay Jadhav ^[4] Dr. Sharmila Gaikwad
^{[1][2][3][4]} Department of Computer Engineer,

Rajiv Gandhi Institute of Technology
Mumbai, India

^[1] piyush91094@gmail.com, ^[2] karanconfusedsoul007@gmail.com, ^[3] vinayjadhav204@gmail.com
^[4] sharmila.gaikwad@mctrgit.ac.in

Abstract— The world has transformed into an extremely fast-paced life, where while following our schedules, interacting with modern technology and anticipating about the future, time flies without a hint of realization. This paper presents a way which attempts to keep a track of your life by allowing the users to visit a day in their past, recent or distant, and reminisce it. Since a lot of information about a person's activities is available through their mobile phone, mobile phone related technologies are used to realize the goal. The implementation of this idea provides not only an opportunity to the people to imbibe the nostalgia and bring a smile to their face but also comes handy in practical situations such as for an alibi, or to know the activities at a particular time. While the attempt to implement this concept will involve processing of data which maybe be privy to the users, utmost care will be taken to ensure that under no circumstances will the information will be accessible to anyone else apart from them.

Keywords: Mobile Computing, Android, Cloud Computing, Record User Activities, Client-Server

I. INTRODUCTION

While thinking of a project like this, two main aspects are there to be emphasized on. One, the availability of information on demand, irrespective of place, time, etc. and second is the placing on control on the hands of user. The second aspect places the entire data and the processes which can be performed on the data under the user's control, which encourages people to trust the application and be comfortable with it. We are interested in one thing: turning the phone into the only tool needed to keep a record of the user's life. It isn't interested in usernames and passwords, or targeted advertising, but rather providing a way to increase accountability among its users. After activating its services, App automatically records a user's location, audio, media, call logs and texts depending on what the user actually wants to record. To save the material, users visit the app and indicate the data they'd like to save. Most of the previous attempts made towards this aim, have a narrow approach where only selected features are targeted and are kept track of. This provides an incomplete embodiment of the person's past. In the application proposed in this paper, a larger number of features are targeted to provide a more robust and complete experience when the user should choose to reminisce the life they have lived.

The paper follows up with a detailed account of what the system is, the fundamentals it is based on and the technology involved. In the latter part before concluding, a

glimpse is given of what the system might look like and the achievements made till now, to demonstrate the feasibility of the idea.

II. PROPOSAL

The idea about this application is that what if someday you want to revisit the past? This application stores your whereabouts, your calls, texts, images clicked, etc. and stores it in a secure database for future viewing. At any point, events regarding to a certain day can be seen and reminisced. The application offers this exhilarating experience and allows you to share it with others.

An account linked to the device is optional, which allows you to back up your history in case of device malfunction or any other unfortunate activity. Syncing with this account allows you to view your history across devices and hence keep your memories perpetual. The storage would be cloud based, which would enable the user to access their data across devices/platforms. The application would be synchronized with the cloud at frequent intervals as chosen by the user.

We found the sweet spot whereby you collect the most amounts of data but you don't have too much hit on the battery. So that means a location is captured once every 15 minutes, and images are updated once an image is clicked. Video, the biggest drain on battery life, automatically gets saved when the device is connected to Wi-Fi and charging is on. Basically user will have the option to choose when the data should be updated. The app

will choose the best available option to increase the backup performance. The way it is intended to approach the development of this system is essentially by dividing it into 2 aspects. First is setting up the cloud functionality on the rudimentary level. After running some tests to ensure the core functionality working we can move on to the next phase.

The objective is to have at least a basic form of cloud server beforehand, so that when the application is developed initial testing can begin immediately. The cloud sever can be given enhanced capabilities later on. The second aspect is the development of the mobile application. For this process the project is object oriented and has entailed java programming language. The mobile application will be initially developed only for the android platform. Currently we have established a server on one of our laptop just for testing purpose and have successfully managed to setup the cloud. In the upcoming time we plan to shift to a dedicated server which can work more fluently and can function on a large scale. We have used ubuntu-14.04.3-server edition along with LAMP editions to setup our cloud server. The apache server has been directed to take incoming connections on port 8080. The router has been configured accordingly to redirect all incoming connections from the public ip on a specified port to the server machine. For the mobile application we will be using java on android studio. Currently the application will be available only on the android platform. This might be extended to other platforms later. The implementation plan of the entire system is divided into 3 phases, which are to be done sequentially. The three phases are Cloud Server Setup, Android Application Development and establishing the connection between cloud and application.

A. Cloud Server Setup

Setting up the cloud server initially helps us in testing the core functionalities that would be required for the project to work. It is necessary that the initial functions required from the cloud is ready before the application to avoid any wastage of time. Following are the steps that are to be taken to successfully establish a rudimentary cloud server.

- ❖ Step 1: Installing Ubuntu
- ❖ Step 2: Configure Apache2
- ❖ Step 3: Configure the Router
- ❖ Step 4: Secure the Server
- ❖ Step 5: Map Public IP Address to a Domain Name (optional)
- ❖ Step 6: Transfer Media (Linux)

The machine has been configured to host 2 separate servers. First is the web server, which allows the user to access their information from any place. Thus making the system available across multiple devices. Some

of the communication between the mobile application and the server (eg. The download feature) uses the web server as its portal. The web server is apache based and has been equipped with access restrictions by using encryption and passwords, which would only allow genuine users to access their content (Fig. 1).

The second server is used for file transfer and managing new user connections (eg. Upload feature). User accounts are created by this server, when first time connections are detected. Each time a new user connects to the server, a password is accepted from them which along with their email id, which is extracted automatically, becomes their login credentials. Using their emails, a separate dynamic space is allocated which is used to store the user information securely (Fig. 2).

The following figures show the implementation of a web server along the aforementioned lines.



Fig 1. Each user has to enter their password to access content.

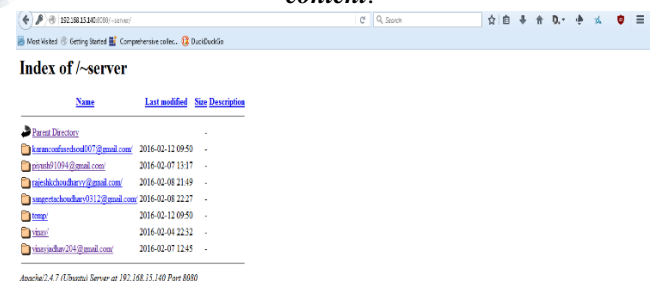


Fig 2. The list of users registered with the server.

B. Android Application Development

Android Application is at the core of the project. This is the part through which all the data enters into the system and provides an interface to the users using the system. The code for the application will be written in java using the Android Studio IDE. The application initially would be developed to function as a stand-alone, i.e. without the involvement of the cloud. After successful implementation of that, we can incorporate the cloud aspect as phase 3. The initial focus of the application would be provide efficient and robust functionality, hence compromising on the user interface slightly which would be developed later.

C. Establishing Cloud Connectivity & Testing

Once both the cloud and the android application are ready, we can proceed on to linking them. This would be final phase towards the completion of the project. In this phase the android application would be connected to the server and data transfer between them would be enabled. This would make the synchronization feature complete.

Once the connection establishment is done, we can begin with the testing and correction procedure.

III. RESULT

To perform the various actions available regarding the users location, an interface is developed to provide convenience to the user. Depending on what the user wants to do, the interface proves as an easy access to it. The following information is regarding the results achieved in this context.

A basic layout of the location feature is shown in Fig 3, below.

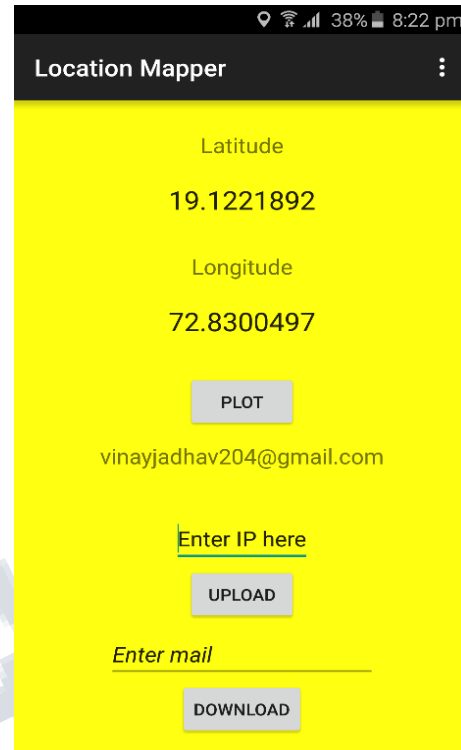


Fig 3. Interface for location information

The current latitude and longitude coordinates are always shown. This information is extracted from both the GPS and the NETWORK feature (whichever is available, prioritizing GPS). Since we yet do not have a dedicated server and a static ip, a field is provided to manually enter the ip address of the server which allows the user to communicate with the server. On installing the application for the first time, a file is created which will store all the location information about the user in it. The user's email id is extracted in order to uniquely identify each user and hence allocate a separate space for them on the server. The upload button allows the user to manually send the data to the server, which maintains the synchronization between the data present in the device and the data at the server. The download button allows the user to download a friend's location information history (if the respective permissions are given). This allows people to share information among them. The files are shown in Fig 4.

To download the information of a different user, the email field has to be used. If the permissions are available the information of the user corresponding to the email id is downloaded. The plot functionality allows the user, to plot their or someone else's last known location on the map. Someone else's location can only be plotted of their file is already available (download feature). Fig 5 shows how on plotting the coordinates of a different user, both your and the other user's location is shown.

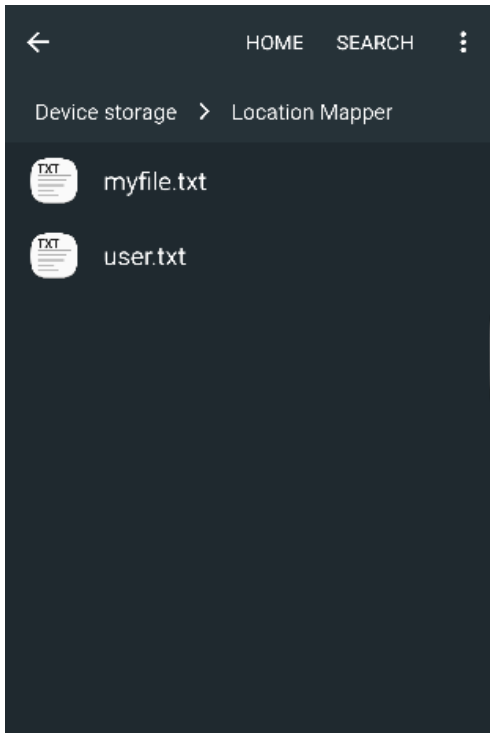


Fig 4. Files generated on installing the application and downloading a user's information

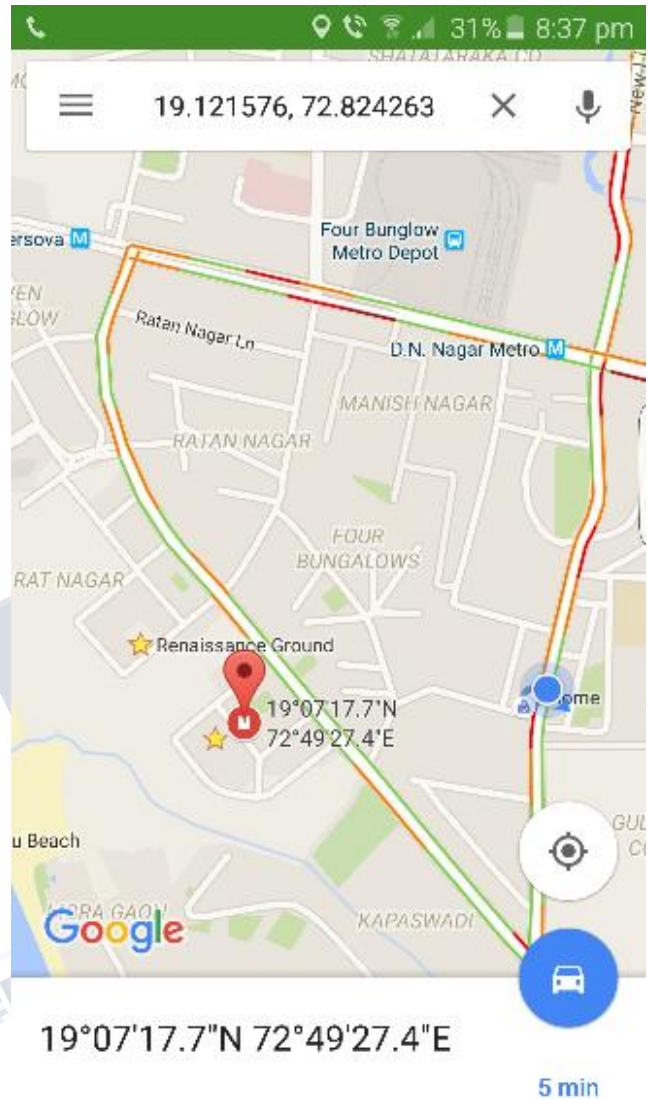


Fig 5. The red marker indicates the other user's location and the blue dot indicates yours.

IV. CONCLUSION

A technique is presented in which it is attempted to present a way to develop a system which can neatly store the information happening in the daily lives of users, which at the moment may not seem very important but would have a great sentiment value at a late point in their lives. Not just sentiment value, but has many practical important purpose too. The system is attempted, to the best of our knowledge, to be made as secure as possible by encrypting the data and restricting access to it so that the information remains private to the users. Currently due to a very low budget and dearth of time, the system has not

been developed to its full potential, but can be expanded and built on very flexibly.

REFERENCES

- [1] Bill Sunderland and Stephen Anderson, Intel Corporation, "Implementing Virtualization In A Global Business-Computing Environment," *Intel Leap Ahead*, 2008.
- [2] Patrick Wendell and Michael J. Freedman, "Going Viral: Flash Crowds in an Open CDN"
- [3] Peter Mell; Timothy Grance; Lee Badger; Tim Grance; Robert Patt-Corner; Jeff Voas, "The NIST Cloud Computing synopsis and Recommendations," *National Institute of Standards and Technology Special Publication 800-145*, 2011
- [4] Daniel Díaz-Sánchez, Florina Almenarez, Andrés Marín, Davide Proserpio, and Patricia Arias Cabarcos, "Media Cloud: An Open Cloud Computing Middleware for Content Management," *IEEE Transactions on Consumer Electronics*, 2011
- [5] Yonggang Wen, Senior Member, IEEE, Xiaoqing Zhu, Member, IEEE, Joel J.P.C. Rodrigues, Senior Member, IEEE, and Chang Wen Chen, "Cloud Mobile Media: Reflections and Outlook," *IEEE Transactions on Multimedia*, 2014
- [6] Huanjing Yue, Xiaoyan Sun, Jingyu Yang, and Feng Wu, "Cloud-Based Image Coding for Mobile Devices—Toward Thousands to One Compression," *IEEE Transactions on Multimedia*, 2013
- [7] Karthik Challa, "Cloud Services for an Android Based Home Security System", *SJSU ScholarWorks*, 2013
- [8] "Cloud Backup and Disaster Recovery Meets Next-Generation Database Demands", *A Forrester Consulting Thought Leadership Paper Commissioned By Microsoft*, 2014
- [9] www.instructables.com/id/how-to-host-your-own-cloud.com