

Centralized Android Devices Management

Anirudh Ruia Gali
Agali3@asu.edu

Abstract—Mobile devices are becoming increasingly popular for daily activities because of their functionality and also serve as suitable replacement for some of the household items. Current generation is using more than one device for different purposes e.g. mobile phone for calling, taking notes, reminders, health, Internet of Things(IoT) etc. while tablet for watching videos, education, form filling/surveying/polling, reading, health etc. And many more devices like watch, fitness bands are being used in conjunction with these mobile devices. Management and control of these devices can be cumbersome and needs an effective system. Keeping all constraints in management of mobile devices, we have developed an android application and backend system to manage all the android devices of an owner in a systematic manner. Such an application can help owners to manage their devices and perform administrative tasks without any hassles.

Keywords — REST; Android; Cloud

I. INTRODUCTION

Smartphone has become de facto tool that is being increasingly leveraged by people in carrying out different functionality. Computing power and storage capacity of these smartphone are increasing year on year making it very dependable for the tasks. These smartphones are equipped with variety of sensors for usage in applications than possible with any individual embedded devices. Using various capabilities of the smartphone, it can be managed and controlled. In Android environment, an app was developed using all information that can be obtained from each android smartphone or a tablet i.e. sensor, device status data and we use them to keep the owner updated about his device. Since the owner needs to be informed and aware of all of his devices, there needs to a centralized system that can authenticate, authorize, store and retrieve all of the device information of each user and works with the app. Hence, a robust system is developed and showcased in following sections in detail to know its importance in various consumer and enterprise scenarios.

II. METHODOLOGY

A. System Overview

Fig 1 shows the overview of the system. Backend logic is implemented in server instance present in Amazon Cloud while frontend is designed in an android app that can be installed in android smartphone and also in tablet.

Fig 2 shows the architecture used for the development of the application.

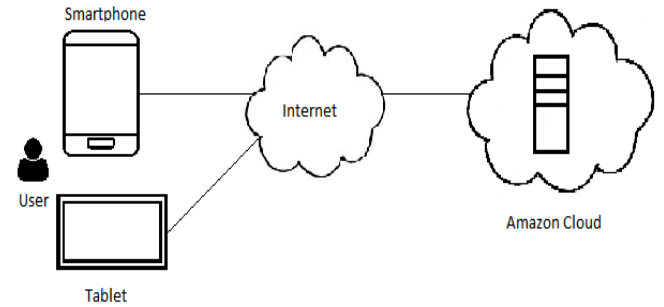


Figure 1 System Overview

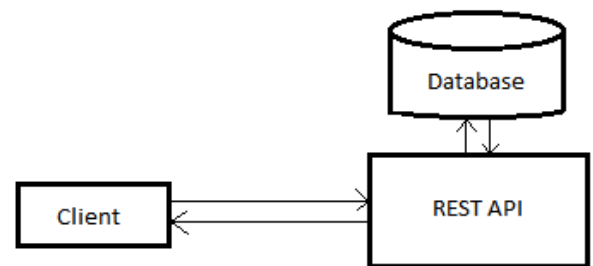


Figure 2 System Architecture

B. Backend Infrastructure

Before starting of development of the app, there needs to be a central server that can coordinate with all the incoming requests coming from the app. For this infrastructure requirement, micro instance of the windows server in Amazon EC2 [1] is chosen from Amazon Web Service, cloud computing services. Our technology stack was installed using wamp-server [2] package. A proper API is to be modelled so that an app can appropriately communicate with the server. A custom RESTful API [3] is developed due its benefits in our scenario of being responsive as well as light weight http protocol based communication, especially required in mobile devices to save from high CPU usage and mobile data bandwidth. PHP language was used to develop web services in the RESTful API. The data between the mobile device and server is transmitted in https packets in JSON format to take advantage of its easy representation and low memory required to parse. Database was created in MySQL to store the user information, devices information and other essential security details. Backend was architected in such way that it enables to add future types of devices with less code change by keeping it highly generalized and normalized.

C. Frontend User Interface

Android App is developed to facilitate the user to lookup essential information for each device such as available memory, location, batter status, apps installed, and android id for each of the registered devices of the owner. Activities, services and GPS sensors from Android SDK [4] are mainly used to capture the essential system information and push it into the central server. A strong security system is deployed to allow only authenticated and authorized users to able to update and lookup his devices from the server and will be briefly discussed in the following section. As an administrative control, owner can lock apps in any of his devices using a simple passcode. This allows him to have better control of important apps used across different devices. App background service is optimized to make only requests when mobile data or Wi-Fi is on, after certain period of time to consume less memory and bandwidth.

D. Security Model

Security is an important aspect while developing the application since important credentials and user information are being shared to the central server through internet. There is a risk of sniffing the packets while in transit between client app and the server. HTTP/SSL [5] type connection ensures that client is connecting to authentic server and the data in the packets is encrypted between client and server communication. In addition once a user logs into his app, a unique token is generated by the server and stored in the database. This unique token allows only authenticated owner to make further requests to the server and retrieve the information. Tokens generated are invalidated after certain amount of time to make sure that user logs in with his most recent credentials again and there is no limitation as to number of devices that a particular user can add to his account.

III. RESULTS AND SCREENSHOTS

A. Results

Project was developed in agile methodology and obtained a release product with all the requirements. We have used variety of tools for this project: Android Studio, Amazon EC2, REST Console Chrome and Wamp Server. The source code of the entire system can be downloaded from [6]. The app was tested on more than two devices in a user account, fixed bugs and final release found to work extraordinarily with very low latency.

B. Screenshots

Below are the some of the screenshots of the app. In figure 3 - a, b, c, d, e, f, g, h show different functionality developed in the app.

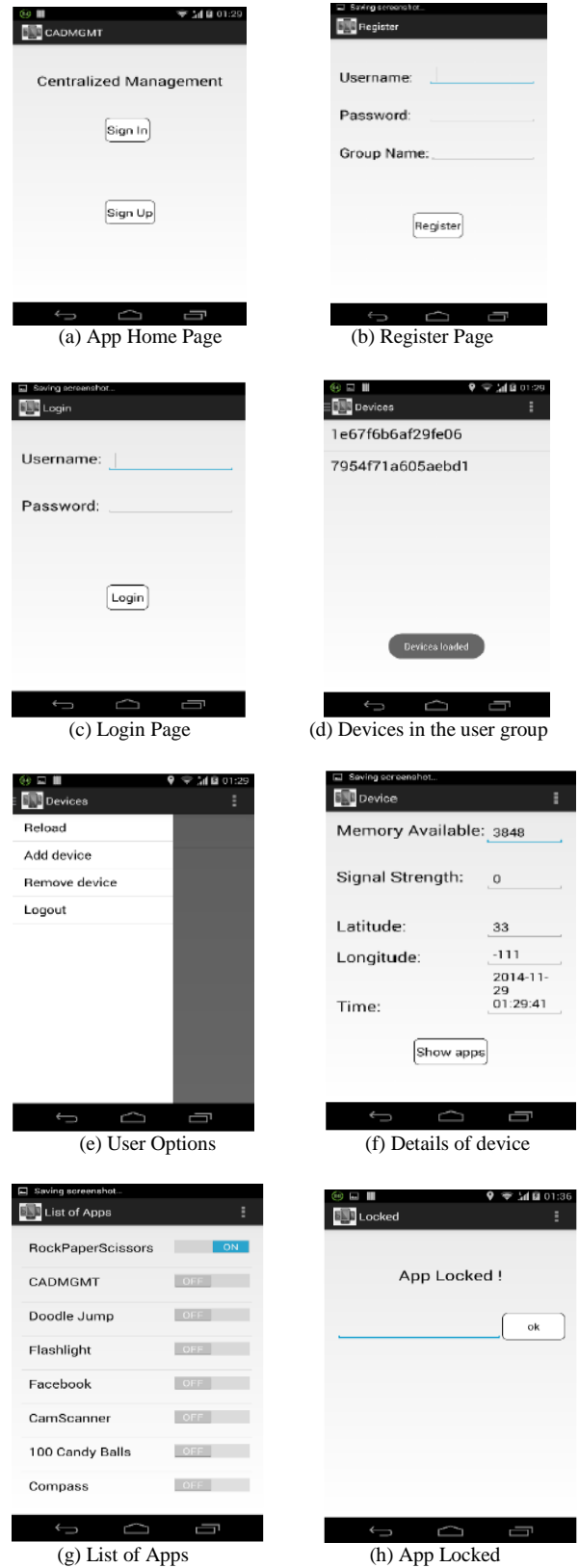


Figure 3 Screenshots of the App

IV. CONTRIBUTIONS

The project was developed in three member team and was completed in the span of two months. Contribution of each member is listed below:-

Anirudh Ruia Gali: Device list view, App lock, Device Statistics update service, Service for adding apps list to the server and User options UI.

Harish Chowdary Thiruveedi: Login, Register services and UI, Main activity UI, App list view UI, Device statistics view, Device statistics update service

Tarun Prabhukar Kasala: Login, Register services, App lock User interface, Service for adding app list to server, testing.

V. LEARNING FROM THE PROJECT

This project helped us gain experience in mobile application development and design of an REST API. Through this project, we learned the working of sensors, system functionality and other information of Android devices. Various popular cloud computing services were studied and Amazon EC2 was opted to be used in this project. Overall, we had an amazing learning curve in designing a real time application in android and solve one of the problems through this application.

REFERENCES

- [1] AWS | Amazon Elastic Compute Cloud (EC2) - Scalable Cloud Hosting. (n.d.). Retrieved May 24, 2015, from <http://aws.amazon.com/ec2/>
- [2] WampServer. (n.d.). Retrieved May 24, 2015, from <http://www.wampserver.com/en/>
- [3] Representational state transfer. (n.d.). Retrieved May 24, 2015, from http://en.wikipedia.org/wiki/Representational_state_transfer
- [4] Android SDK. (n.d.). Retrieved May 24, 2015, from <https://developer.android.com/sdk/index.html>
- [5] RFC 2818 - HTTP Over TLS. (n.d.). Retrieved May 24, 2015, from <https://tools.ietf.org/html/rfc2818>
- [6] CADMGMT source code. (n.d.). Retrieved May 24, 2015, from <https://github.com/anirudhruia/cadmgmt>