A Synopsis Report
on

Charging Buddy

by

1. Sagar Ambilpure
2. Aditya Chavan
3. Uzair Chhapra

under the guidance of

Prof. D.P. Kapse

MCT
MANJARA CHARITABLE TRUST
RAJIV GANDHI INSTITUTE OF TECHNOLOGY, MUMBAI

Department of Computer Engineering

University of Mumbai

October - 2019

MCT
MANJARA CHARITABLE TRUST
RAJIV GANDHI INSTITUTE OF TECHNOLOGY, MUMBAI

Juhu-Versova Link Road Versova, Andheri(W), Mumbai-53.

Certificate

Department of Computer Engineering

This is to certify that

1. Sagar Ambilpure
2. Aditya Chavan
3. Uzair Chhapra

Have satisfactory completed this synopsis entitled

Charging Buddy

Towards the partial fulfillment of the

BACHELOR OF ENGINEERING
IN
(COMPUTER ENGINEERING)
as laid by University of Mumbai.

Guide | **Head of Department**

Prof. D.P. Kapse | **Dr.Satish Y. Ket**

Principal

Dr.Sanjay Bokade

Internal Examiner | **External Examiner**

Declaration

We wish to state that the work embodied in this synopsis titled "Charging Buddy" forms our own contribution to the work carried out under the guidance of "Prof. D.P. Kapse" at the Rajiv Gandhi Institute of Technology.

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. we also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. we understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Sagar Ambilpure (705)

Aditya Chavan (714)

Uzair Chhapra (715)

Acknowledgement

We wish to express our sincere gratitude to Dr. Sanjay U. Bokade, Principal and Dr. Satish. Y. Ket , H.O.D of Department Computer Engineering of Rajiv Gandhi Institute of Technology for providing us an opportunity to do our project work on “Charging Buddy”.

This project bears on imprint of many peoples. We sincerely thank our project guide Mr/Mrs. Prof. D.P. Kapse for his guidance and encouragement in carrying out this synopsis work.

Finally, we would like to thank our colleagues and friends who helped us in completing project work successfully.

1. Sagar Ambilpure
2. Aditya Chavan
3. Uzair Chhapra

Abstract

Currently, when a user's cell phone runs out of battery and there is no charging outlet nearby there very little or nothing that the user can do about it. Even if a charging outlet is available nearby one would have to stay close to the charging outlet until the phone is charged. Our system would provide such users with a solution which would enable them to do other things while the cell phone is being charged without worrying about their phones being stolen. Our charging stations would be located at various public places which would be available for all the registered users. These charging stations would be like vending machines i.e. fully automated using IoT. These charging stations, when placed in stores, would attract more customers increase dwell time which would consequently increase sales.

Mobile phone is our means to remain connected. While the phones have progressively got more powerful processors and large touch screen interfaces, their power requirement has increased correspondingly. Unfortunately, battery technology has not been growing at a comparable pace. Hence, there is a need to frequently charge the batteries. While travelling, people face a common problem of charging electronic appliances. Most of the cell phone users have experienced having their cell phones run out of battery when they are not near a charging outlet. Our system provides such users an elegant solution to this widespread problem.

Contents

List of Figures **i**

List of Tables **ii**

List of Algorithms **iv**

List of Acronyms **v**

List of Symbols **vi**

1 Introduction **1**
- 1.1 Introduction Description 1
- 1.2 Organization of Report 1

2 Literature Review **3**
- 2.1 Survey Existing System 3
- 2.2 Limitations Of Existing System 4
- 2.3 Problem Statement and Objectives 5
  - 2.3.1 Objectives 5
- 2.4 Scope 5

3 Proposed System **6**
- 3.1 Framework/Algorithm 6
- 3.2 Details of Hardware and Software 7
  - 3.2.1 Software Requirements 7
  - 3.2.2 Hardware Requirements 8
- 3.3 Design Details 9
  - 3.3.1 Detailed Design 9
- 3.4 Methodology/Procedures 10
  - 3.4.1 Procedures 11

4 Implementation **13**
- 4.1 Implementation Plan 13
- 4.2 Project Status 13

5 Conclusions **15**

References **16**

List of Figures

3.1 Apriori Algorithm flowchart 6
3.2 The Proposed System/System Architecture 9
3.3 Use Case Diagram 10
3.4 MQTT Sequence Diagram 10
3.5 Algorithm of Apriori 11
3.6 Flow of Apriori 12

4.1 Gantt Chart:Implementation Plan 13
4.2 AWS MQTT on Website 14
4.3 AWS MQTT on Terminal 14

List of Tables

List of Algorithms

- Apriori Algorithm
- K-Means ALgorithm
- MQTT Protocol

List of Acronyms

- MQTT - Message Queuing Telemetry Transport
- AWS - Amazon Web Services
- HTML - Hypertext Markup Language
- CSS - Cascading Style Sheets

List of Symbols

Chapter 1

Introduction

1.1 Introduction Description

This project is basically about developing a mobile charging station. In the contemporary world, mobile phones have become an indispensable tool for mankind. Smart phones of today are basically a portable computer. With the rise in speed, power and capability in mobile phones, there has been an equal and opposite effect on its power consumption. Even though the processors today strive to be power efficient, yet they do consume a lot of power in the first place.

No doubt, battery technologies today have also improved over the years. But this improvement is no where near in proportion to the demands of the power hungry microprocessors used in mobile phones. What this basically means is that, phones, say around 10 years back, weren't smart enough to do many tasks that they today. But what this also means is that, those phones used to last for days if not weeks on a single charge. The problem of on the go charging for mobile phones never really existed back then. It is only in the recent past this problem has started to take center-stage.

The solution to this battery problem in smartphones of today has not been completely found. Smartphone manufacturers have tried solutions like fitting the largest possible batteries in the phones, optimising their software, and various other methods. In the recent years, faster charging methods have also been implemented in mobile phones so that they safely draw large amounts of power to charge quickly. All such solutions have undoubtedly been helpful, but the fact of the matter is that even after all these methods, there are tons of smartphone users who have to be on-the-go all the time and do not get access to charging their mobile phones. Such people generally carry mobile power banks.

The problem with power banks is that they are cumbersome to carry and that they also need to be charge again and again for storing the extra juice. While this is reasonable enough solution to the charging problem, but it is quite a cumbersome one. The basic idea of our project is to provide public charging stations with a locker mechanism for minor power top ups at major public places.

1.2 Organization of Report

- **Ch.1 Introduction:** The current scenario of smartphones and the increasing problem of keeping them charge always.

- **Ch.2 Review of Literature:**Configurations of the Servers used for the project which handle all user services, communication and access control.
- **Ch.3 Proposed System:**Explains the the entire architecture of the project.
- **Ch.4 Implementation Plan:**Talks about what and how the proposed system has been implemented.
- **Ch.5 Conclusion:**The solution to the problem of keeping the smartphones of today powered up all the time may not have yet been found, but this project certainly does take a big step in the right direction.

Chapter 2

Literature Review

Mobile charging stations are not new to the people. In fact, they have been present for quite a while now. The reason they are not heard of enough or appreciated enough is that, in its existing form, it is quite cumbersome to use.

2.1 Survey Existing System

There are various mobile charging stations that have already been developed. These stations basically use wired technology within them to handle or secure mobile phones. Most charging stations are actually not secured. The owner has to stand near their mobile phones to make sure nobody else runs away with it. Another drawback of the current form of mobile charging station is that fact that such stations work in an entirely offline manner.

The mobile battery charger starts charging a mobile connected to it when a coin is inserted at the coin insertion slot at the input stage. The type of coin and the size will be displayed at the LCD display for the user so as to ensure correct coin insertion. Any other coin, if inserted in the slot will be returned to refund box. A sensor attached to the coin insertion slot accepts the coin into the battery charging unit and start charging the mobile battery for a specific period controlled by the software of the microcontroller. The sensor is an IR sensor.

The resistance of the sensor decreases when IR (infrared) light falls on it. A good sensor will have near zero resistance in presence of light and a very large resistance in absence of light. When the coin obstruct the IR light falling on a sensor, it sends a pulse to the control unit authorizing the start of charging the mobile battery connected to the device. Two IR sensors are used for positive authentication of the charging process [1].

Coin accepted or rejected is based on the diameter of the coin. This invokes microcontroller along with LCD interface displays the selection of mobile option if particular mobile is selected for charging the corresponding routine is activated and charge the mobile for a particular duration of time .When the routine completes, it indicates charge complete message through LCD display. Similarly the same procedure is followed for charging more than four different mobiles simultaneously.

2.2 Limitations Of Existing System

There are various limitations to the current model of a mobile charging station. Some of them are as follows:

- **Security:** Most current charging stations are open in nature. There are no personalised lockers for a user's mobile phone to be put in.

- **Hardwired:** The current forms of mobile charging stations are all hardwired. Even those with lockers in them open their locks by passing messages through wires. Wireless communication has never been used in such systems.

- **No IoT Device:** For seamlessly using the proposed charging station, we believe that IoT has to be incorporated. This is missing in the current systems. The reason for equipment interface unit is all the electronic home apparatuses are associated with the raspberry pi board which is associated with the WiFi by utilizing Wi-Fi module. All the electronic machines are worked and controlled through our advanced mobile phone or PC or tablet [2].

- **No Cloud Connectivity:** The AWS IOT platform act like a cloud server for exchanging the information in between end user and system. AWS cloud server provides the facility of interfacing any gateway device such as Raspberry Pi, and Arduino board for controlling any application [3].

- **No Smart Locker Unlock:** Door Automation is emerging technology in Home Automation. From the last decades a number of standards have been defined for Door Lock Appliances. The main objective of Door automation is to provide Security locks for door, comfort, connivance security and energy efficiency for user with help of IoT and WSN [4].

- **No Web Interface:** Using Web Frameworks to develop a website has become indispensable in today's world. Marketing can only be done if the product can be found on the web. We've all made sandwiches using bread at some point of time in life. Frameworks in web development are like the bread, they are the base on which sandwiches, here web applications, web services and software are built. Just like you have umpteen options in making a sandwich regarding choice of bread, fillings used, flavours and so on. We get number of options when it comes to web application development frameworks – including the correct framework, specific language, correct libraries and so on [5].

- **No Side Benefits:** There are no perks for the user to get attracted towards using the existing version of mobile charging stations. There could be features like incentives to the user, recommendation system on the user's buying habits, etc. Also, there should be a benefit for the place that is installing the charging station. If users are able to leave their phone to charge securely, then there is a high chance that they may roam around and increase the sales of the shops nearby.

2.3 Problem Statement and Objectives

The problem statement is to develop a unique mobile charging station with web connectivity. This station would run on an IoT device (Raspberri Pi) and would communicate to our web Django server using the MQTT Protocol over Amazons AWS Server. The problem to be solved here is the fact the mobile phones today need to be charged up at least once day, if not more.

2.3.1 Objectives

- To develop a smart mobile charging station. This station should have wireless Internet connectivity.
- To implement a mobile application to show position of our Mobile Charging station on the map.
- To make a Mobile Charging Station and secure it's open and close locker mechanism using IoT.
- To develop a server for our application from where booking requests and user history can be managed.
- To provide an API to shopping mall stores for integrating user purchase details with our application.
- To provide product recommendations based on purchase history to user in our Charging Station partnered stores.

2.4 Scope

The scope of the project is not only malls and shopping complexes, but at all public spaces. This project though, requires care and maintenance. As such it is best suited to be placed at spaces where the entire charging station cannot be stolen from.

Also this mobile charging station requires a decent internet connection for all of its smart features to function properly.

Chapter 3

Proposed System

The Proposed System is different than the traditional systems used so far in many ways. Our Project ChargingBuddy gives user a secured way to charge his/her phone without worrying about any theft. ChargingBuddy will be set up in the various shops around the city. When using this system the user will be tension relieved about the device and hence will spend time looking around the shop. After receiving the feedback on what the customers usually look around for in the shop we will be sending recommendations to the chargingbuddy user. By the use of this system not only will the customer be benefited but also the shop.

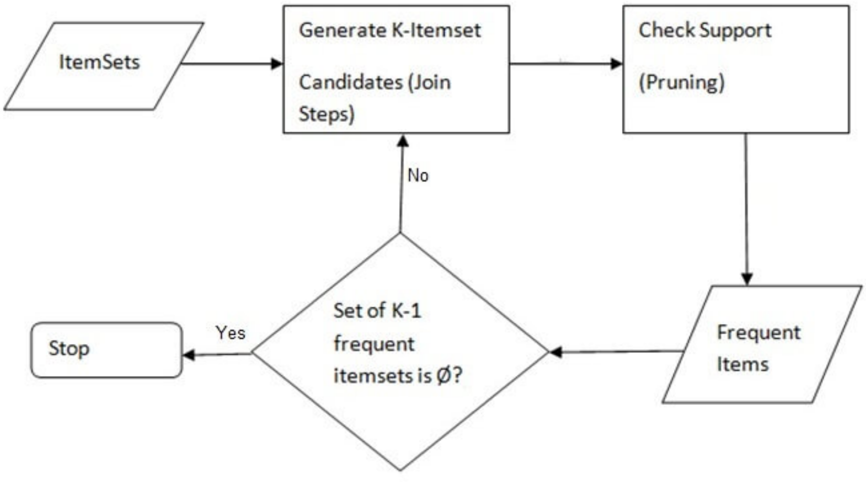


Figure 3.1: Apriori Algorithm flowchart

3.1 Framework/Algorithm

Django is a high-level Python Web framework that encourages rapid development and clean pragmatic design. Hence becomes easy for managing our project. Using python as a language helps in analysing data in an easy and efficient way. For recommendation we are planning to use Apriori Algorithm for association and K-means for clustering. If the shopping data of users gets over 20gb or 30gb we will switch our recommendation system. From a single machine to a cluster of machining working together using hadoop framework.

Apache Hadoop is a collection of open-source software utilities that facilitate using a network of many computers to solve problems involving massive amounts of data and computation.

3.2 Details of Hardware and Software

3.2.1 Software Requirements

- **AWS MQTT:** MQTT is a widely adopted, lightweight messaging protocol designed for constrained devices. For more information, see MQTT. The AWS IoT message broker supports Quality of Service (QoS) levels 0 and 1. Although the AWS IoT message broker implementation is based on MQTT version 3.1.1, it deviates from the specification as follows: In AWS IoT, subscribing to a topic with QoS 0 means a message is delivered zero or more times. A message might be delivered more than once. Messages delivered more than once might be sent with a different packet ID. In these cases, the DUP flag is not set. AWS IoT does not support publishing and subscribing with QoS 2. The AWS IoT message broker does not send a PUBACK or SUBACK when QoS 2 is requested. When responding to a connection request, the message broker sends a CONNACK message. This message contains a flag to indicate if the connection is resuming a previous session. When a client subscribes to a topic, there might be a delay between the time the message broker sends a SUBACK and the time the client starts receiving new matching messages. The MQTT specification provides a provision for the publisher to request that the broker retain the last message sent to a topic and send it to all future topic subscribers. AWS IoT does not support retained messages. If a request is made to retain messages, the connection is disconnected. The message broker uses the client ID to identify each client. The client ID is passed in from the client to the message broker as part of the MQTT payload. Two clients with the same client ID are not allowed to be connected concurrently to the message broker. When a client connects to the message broker using a client ID that another client is using, the new client connection is accepted and the previously connected client is disconnected." On rare occasions, the message broker might resend the same logical PUBLISH message with a different packet ID.

  The message broker does not guarantee the order in which messages and ACK are received.

- **Django Server:** Django is a Python-based web framework which allows you to quickly create web application without all of the installation or dependency problems that you normally will find with other frameworks. When you're building a website, you always need a similar set of components: a way to handle user authentication (signing up, signing in, signing out), a management panel for your website, forms, a way to upload files, etc. Django gives you ready-made components to use. Django can build almost any type of website. It can also work with any client-side framework and can deliver content in any format such as HTML, JSON, XML etc. Some sites which can be built using Django are wikis, social networks, new sites etc. Since Django framework is made for making web development easy, it has been engineered in such a way that it automatically do the right things to protect the website. For example, In the Django framework instead of putting a password in cookies, the hashed password is stored in it so that it can't be fetched easily by hackers.

3.2.2 Hardware Requirements

- **Raspberry Pi 3 Model B:** Raspberry Pi 3 - Raspberry Pi 3 is the third generation Raspberry Pi. It is a miniature marvel, packing considerable computing power into a footprint no larger than a credit card. The processor at the heart of the Raspberry Pi system is a Broadcom BCM2837 system-on-chip (SoC) which houses a 1.2 GHz Quad Core ARM Cortex-A53 processor. The vast majority of the system's components, including its central and graphics processing units along with the audio and communications hardware, are built onto that single component along with 1 GB LPDDR2 memory chip at the centre of the board. It is not just this SoC design that makes the BCM2837 different to the processor found in a typical desktop or laptop, however, it also uses a different instruction set architecture (ISA), known as ARM.

  The Pi comes equipped with on-board 10/100 BaseT Ethernet Socket, HDMI and Composite RCA port for video, 3.5 mm audio output jack, 15-pin MIPI Camera Serial Interface (CSI-2), Display Serial Interface, Bluetooth 4.1, 802.11 b/g/n Wireless LAN, Micro SDIO for Micro SD Card, 4 USB 2.0 Connectors, 40 pin header containing 27 GPIO pins and Micro USB socket for power supply.

  The Raspberry Pi is a single board computer and is designed to run an operating system called GNU/Linux Raspbian. Hereafter referred to simply as Linux. Unlike Windows or OS X, Linux is open source, so it is possible to download the source code for the entire operating system and make whatever changes desired. The Raspberry Pi 3 can also run Windows 10 IoT and many other embedded operating systems most of which are Linux derivatives. The operating system should be loaded in a MicroSD card and boot from it. With powerful computing resources, large number of multimedia interfaces and GPIO pins, Raspberry Pi 3 is a suitable choice to run a software oriented complex IoT or Embedded project that requires sufficient computing power as well as large scale sensor connectivity. With on-board Bluetooth and Wi-Fi, this 3rd generation Pi can be easily deployed in an IoT network. The key specifications of the Raspberry Pi 3 are summarized in the following table

- **DC 12V Electric Lock Assembly Solenoid:** This DC 12V Cabinet Door Lock Electric Lock Assembly Solenoid can be used for locking sell-machine, storage shelf, file cabinet and etc. The hidden way of unlocking can be used for an emergency. The lock works as the circuits disconnects, and it will unlock as the instant power-on. It is steady, durable and energy-saving and had a long lifespan. In the anti-theft and shockproof design, the lock is better than other kinds of locks. After connecting the wires and when the current is available, the electric lock can control the door's opening and closing.

  Features:

  1. Iron Body Material
  2. High quality ultra-compact electric lock.
  3. Rustproof, durable, safe, convenient to use.
  4. Suction which tightly sucks the iron, thus locking the door.
  5. Applicable for being installed in the escape door or fire door electronic controlled system.

6. Adopts the principle of electric magnetism, when the current through the silicon, the electromagnetic lock will achieve a strong.

3.3 Design Details

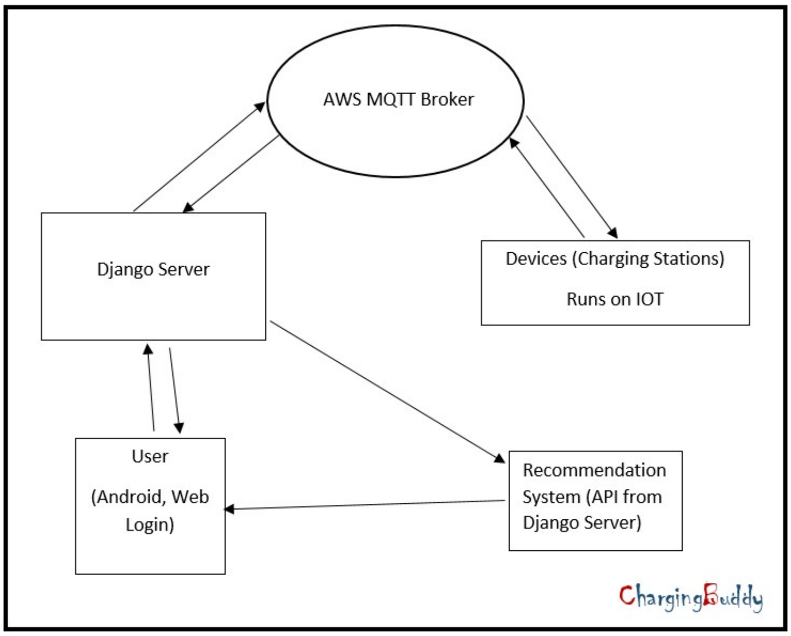


Figure 3.2: The Proposed System/System Architecture

- When the user wishes to charge his device using our system he/she will open chargingbuddy app or chargingbuddy website. They will be asked to give access of their location so the nearest charging station can be shown to them.
- Once the user reaches his nearest station he can open his locker through phone. An get request will be send from mobile phone to the Django server. Django server will publish this request using MQTT to open the locker. Locker containing Raspberry Pi will be a subscriber of the MQTT topic on the AWS server.
- AWS MQTT Broker will tell Raspberry pi, which will be waiting for the instructions, to open the corresponding locker.

3.3.1 Detailed Design

- The user will check for the available lockers in the stations. Accordingly user can do following tasks:
  1. Book a slot for using it afterwards, by giving some token amount.
  2. Visit the available unused locker and put the phone on charging and shop around.

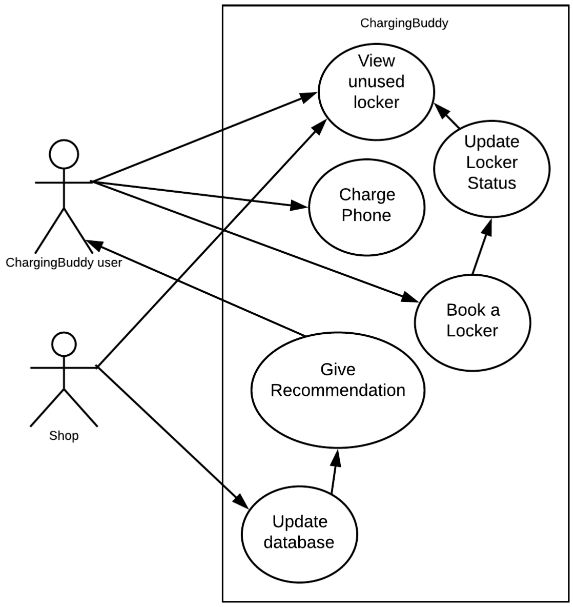


Figure 3.3: Use Case Diagram

- Upon successfully keeping the device into the locker the current state of locker will be changed to busy so other users can charge their phone in another locker.
- While the device is charging all the shopping done by the customer will be saved for giving proper and useful recommendations to the user.

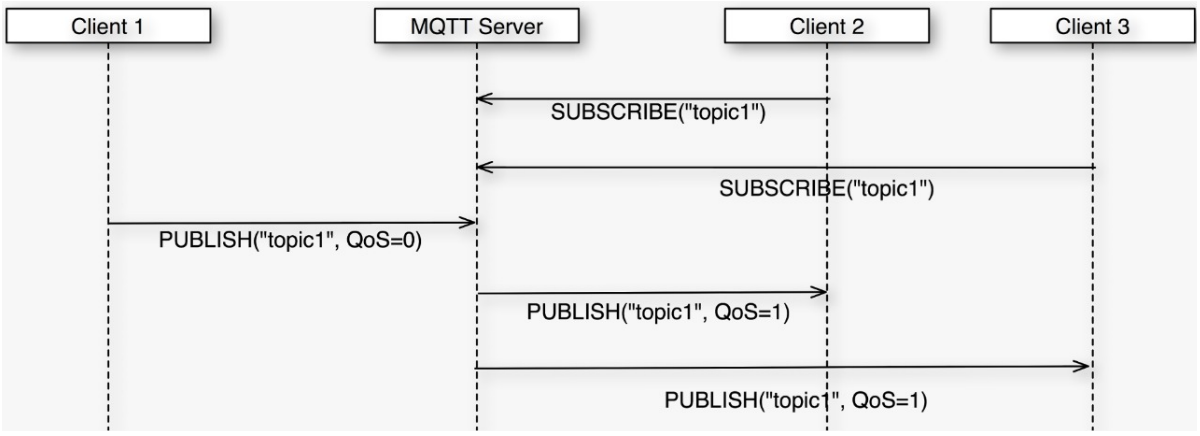


Figure 3.4: MQTT Sequence Diagram

Small description about above diagram.

3.4 Methodology/Procedures

- Currently we are using apriori algorithm. Apriori is an algorithm for frequent item set mining and association rule learning over relational databases.

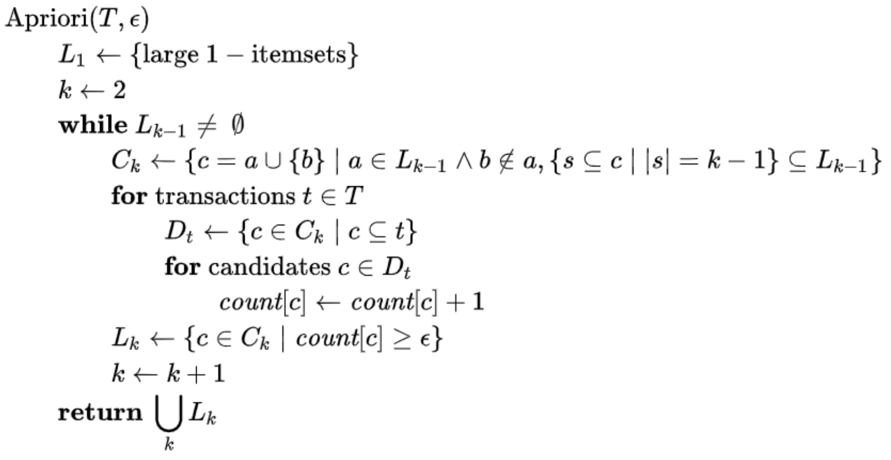
- It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database.

- The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database, this has applications in domains such as market basket analysis. We will use it for shopping recommendation

- For user server communication, we propose to implement MQTT Protocol. This a very light weight and efficient protocol for sending and receiving messages at the same time.

- For providing a good user interface, we propose to use Django Framework for a solid and robust user management system.

3.4.1 Procedures



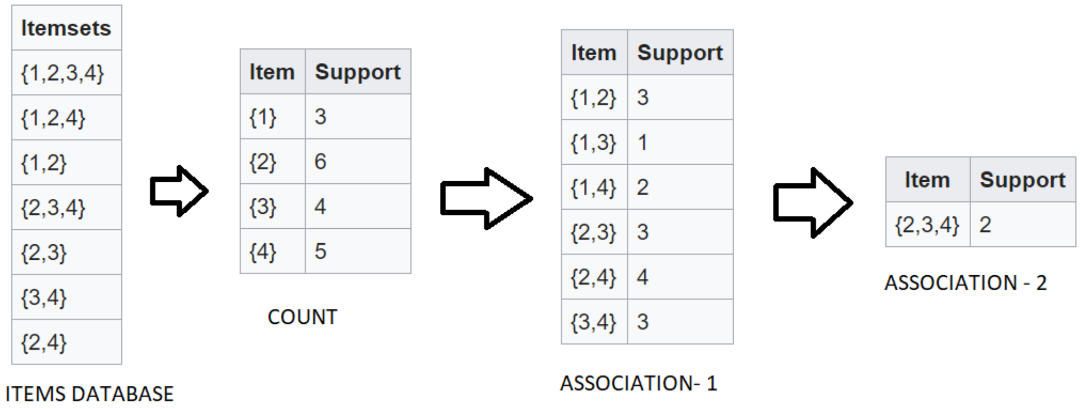
```
Apriori($T, \epsilon$)
    $L_1 \leftarrow \{\text{large } 1 - \text{itemsets}\}$
    $k \leftarrow 2$
    while $L_{k-1} \neq \emptyset$
        $C_k \leftarrow \{c = a \cup \{b\} \mid a \in L_{k-1} \land b \notin a, \{s \subseteq c \mid |s| = k - 1\} \subseteq L_{k-1}\}$
        for transactions $t \in T$
            $D_t \leftarrow \{c \in C_k \mid c \subseteq t\}$
            for candidates $c \in D_t$
                $count[c] \leftarrow count[c] + 1$
        $L_k \leftarrow \{c \in C_k \mid count[c] \geq \epsilon\}$
        $k \leftarrow k + 1$
    return $\bigcup_k L_k$
```

Figure 3.5: Algorithm of Apriori

- Apriori uses a ”bottom up” approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found.

- As we can see 1st we had a set of database of all the items bought by people. We iterate them and get the count of each item.

- Further we get a Small estimation of association between items depending upon what we set ”Support” to.



Itemsets
{1,2,3,4}
{1,2,4}
{1,2}
{2,3,4}
{2,3}
{3,4}
{2,4}

ITEMS DATABASE

Item	Support
{1}	3
{2}	6
{3}	4
{4}	5

COUNT

Item	Support
{1,2}	3
{1,3}	1
{1,4}	2
{2,3}	3
{2,4}	4
{3,4}	3

ASSOCIATION- 1

Item	Support
{2,3,4}	2

ASSOCIATION - 2

Figure 3.6: Flow of Apriori

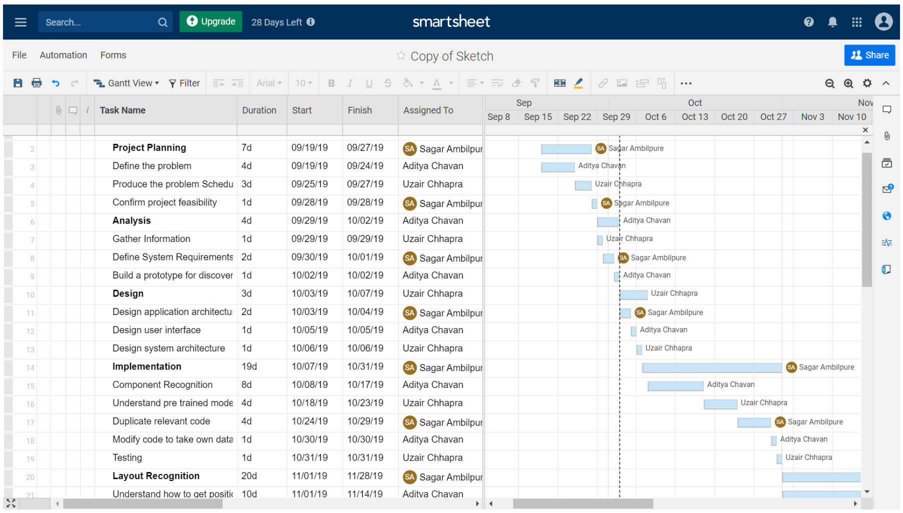
- Higher the support needed more iterations will take place and a greater group of association can be made
- In our flow diagram we see in last iteration we get association of items 2,3 and 4. Hence people who bought any of the these item will get recommendation of other items in the association.

Chapter 4

Implementation

4.1 Implementation Plan

At crowded places a person is forced to stand besides phone when he keeps it for charging because of fear of theft. So by implementing this system in places like malls and shops there is high probability of people keeping their phone on charging and then continue with their shopping. This is enabled because of security to the phone which we are planning to provide. This will also leads to people spending time on shopping rather than standing near phone and hence increasing the probability of rise in revenue of the shops.



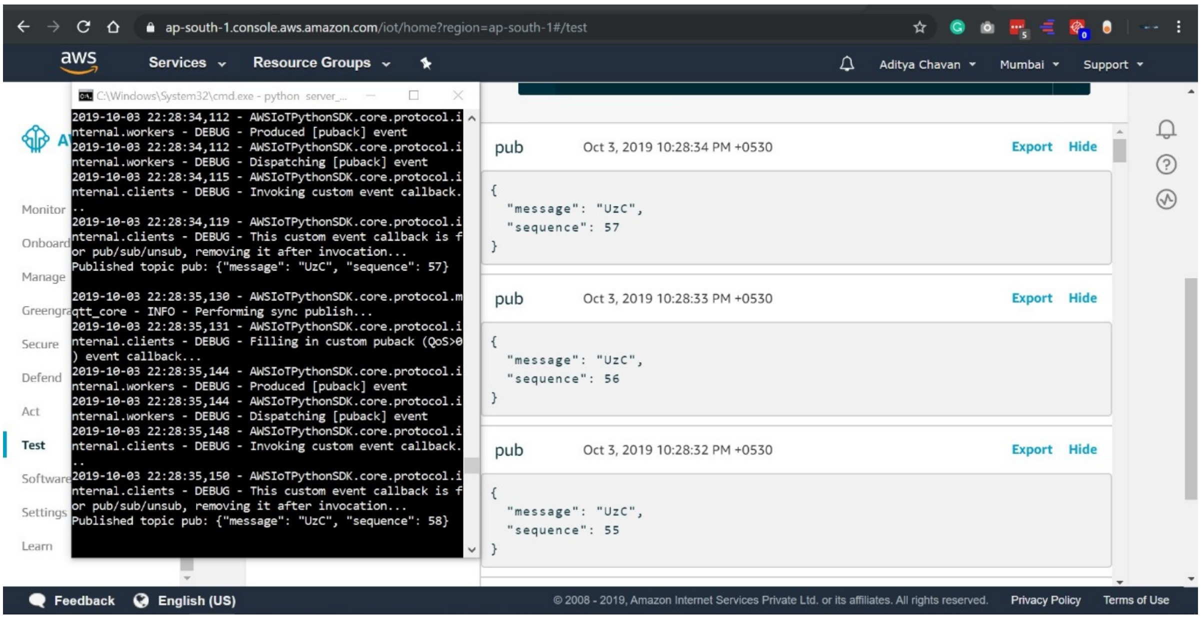
	Task Name	Duration	Start	Finish	Assigned To
2	**Project Planning**	7d	09/19/19	09/27/19	SA Sagar Ambilpur
3	Define the problem	4d	09/19/19	09/24/19	Aditya Chavan
4	Produce the problem Schedu	3d	09/25/19	09/27/19	Uzair Chhapra
5	Confirm project feasibility	1d	09/28/19	09/28/19	SA Sagar Ambilpur
6	**Analysis**	4d	09/29/19	10/02/19	Aditya Chavan
7	Gather Information	1d	09/29/19	09/29/19	Uzair Chhapra
8	Define System Requirements	2d	09/30/19	10/01/19	SA Sagar Ambilpur
9	Build a prototype for discover	1d	10/02/19	10/02/19	Aditya Chavan
10	**Design**	3d	10/03/19	10/07/19	Uzair Chhapra
11	Design application architectu	2d	10/03/19	10/04/19	SA Sagar Ambilpur
12	Design user interface	1d	10/05/19	10/05/19	Aditya Chavan
13	Design system architecture	1d	10/06/19	10/06/19	Uzair Chhapra
14	**Implementation**	19d	10/07/19	10/31/19	SA Sagar Ambilpur
15	Component Recognition	8d	10/08/19	10/17/19	Aditya Chavan
16	Understand pre trained mode	4d	10/18/19	10/23/19	Uzair Chhapra
17	Duplicate relevant code	4d	10/24/19	10/29/19	SA Sagar Ambilpur
18	Modify code to take own data	1d	10/30/19	10/30/19	Aditya Chavan
19	Testing	1d	10/31/19	10/31/19	Uzair Chhapra
20	**Layout Recognition**	20d	11/01/19	11/28/19	SA Sagar Ambilpur
21	Understand how to get positi	10d	11/01/19	11/14/19	Aditya Chavan

Figure 4.1: Gantt Chart:Implementation Plan

4.2 Project Status

- Currently we have successfully completed receiving and publication of request using AWS MQTT. As we can see in the Screenshot one side is publisher and another subscriber.
- Whatever a publisher publishes the corresponding subscriber will receive the data and can process it to do following tasks.

- In our implementation we are passing a sequence number which is increasing by 1 factor after every publication. As we see it takes no time for the subscriber to receive it.


ap-south-1.console.aws.amazon.com/iot/home?region=ap-south-1#/test
aws
Services
Resource Groups
Aditya Chavan
Mumbai
Support
C:\Windows\System32\cmd.exe - python server_...
Monitor
Onboard
Manage
Greengra
Secure
Defend
Act
Test
Software
Settings
Learn
Export
Hide
Feedback
English (US)
© 2008 - 2019, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

```
2019-10-03 22:28:34,112 - AWSIoTPythonSDK.core.protocol.i
nternal.workers - DEBUG - Produced [puback] event
2019-10-03 22:28:34,112 - AWSIoTPythonSDK.core.protocol.i
nternal.workers - DEBUG - Dispatching [puback] event
2019-10-03 22:28:34,115 - AWSIoTPythonSDK.core.protocol.i
nternal.clients - DEBUG - Invoking custom event callback.
..
2019-10-03 22:28:34,119 - AWSIoTPythonSDK.core.protocol.i
nternal.clients - DEBUG - This custom event callback is f
or pub/sub/unsub, removing it after invocation...
Published topic pub: {"message": "UzC", "sequence": 57}

2019-10-03 22:28:35,130 - AWSIoTPythonSDK.core.protocol.m
qtt_core - INFO - Performing sync publish...
2019-10-03 22:28:35,131 - AWSIoTPythonSDK.core.protocol.i
nternal.clients - DEBUG - Filling in custom puback (QoS>0
) event callback...
2019-10-03 22:28:35,144 - AWSIoTPythonSDK.core.protocol.i
nternal.workers - DEBUG - Produced [puback] event
2019-10-03 22:28:35,144 - AWSIoTPythonSDK.core.protocol.i
nternal.workers - DEBUG - Dispatching [puback] event
2019-10-03 22:28:35,148 - AWSIoTPythonSDK.core.protocol.i
nternal.clients - DEBUG - Invoking custom event callback.
..
2019-10-03 22:28:35,150 - AWSIoTPythonSDK.core.protocol.i
nternal.clients - DEBUG - This custom event callback is f
or pub/sub/unsub, removing it after invocation...
Published topic pub: {"message": "UzC", "sequence": 58}
```

pub Oct 3, 2019 10:28:34 PM +0530

```
{
  "message": "UzC",
  "sequence": 57
}
```

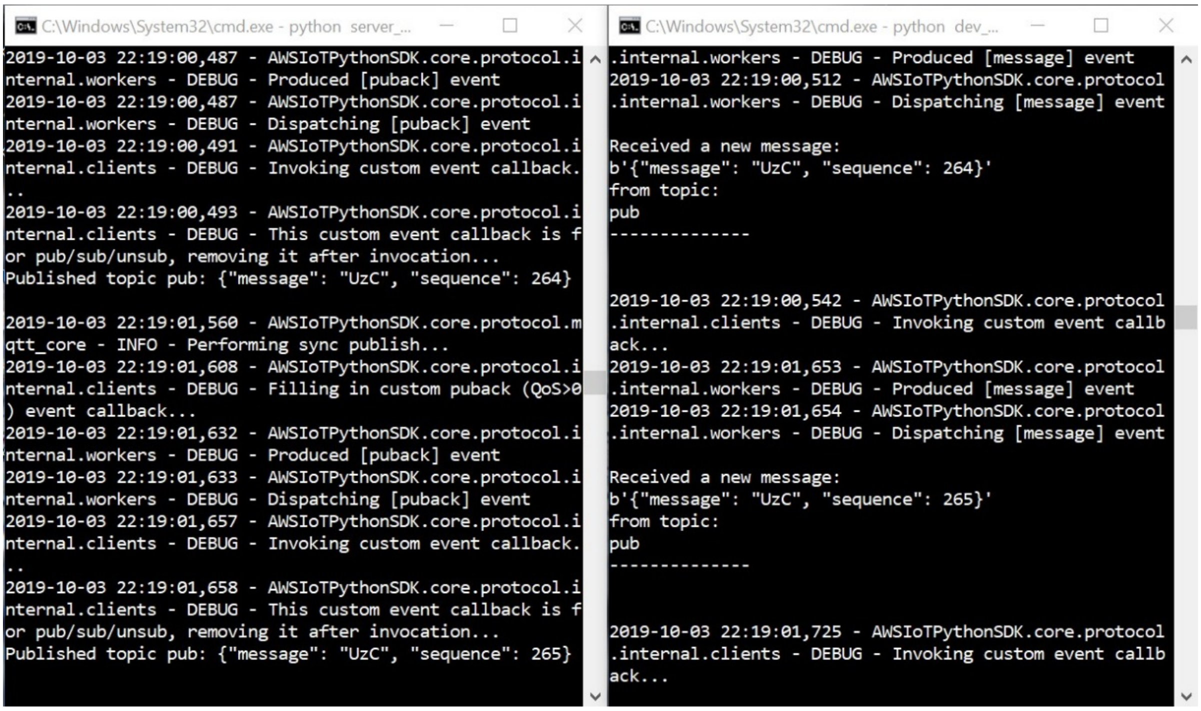
pub Oct 3, 2019 10:28:33 PM +0530

```
{
  "message": "UzC",
  "sequence": 56
}
```

pub Oct 3, 2019 10:28:32 PM +0530

```
{
  "message": "UzC",
  "sequence": 55
}
```

Figure 4.2: AWS MQTT on Website


C:\Windows\System32\cmd.exe - python server_...
C:\Windows\System32\cmd.exe - python dev_...

```
2019-10-03 22:19:00,487 - AWSIoTPythonSDK.core.protocol.i
nternal.workers - DEBUG - Produced [puback] event
2019-10-03 22:19:00,487 - AWSIoTPythonSDK.core.protocol.i
nternal.workers - DEBUG - Dispatching [puback] event
2019-10-03 22:19:00,491 - AWSIoTPythonSDK.core.protocol.i
nternal.clients - DEBUG - Invoking custom event callback.
..
2019-10-03 22:19:00,493 - AWSIoTPythonSDK.core.protocol.i
nternal.clients - DEBUG - This custom event callback is f
or pub/sub/unsub, removing it after invocation...
Published topic pub: {"message": "UzC", "sequence": 264}

2019-10-03 22:19:01,560 - AWSIoTPythonSDK.core.protocol.m
qtt_core - INFO - Performing sync publish...
2019-10-03 22:19:01,608 - AWSIoTPythonSDK.core.protocol.i
nternal.clients - DEBUG - Filling in custom puback (QoS>0
) event callback...
2019-10-03 22:19:01,632 - AWSIoTPythonSDK.core.protocol.i
nternal.workers - DEBUG - Produced [puback] event
2019-10-03 22:19:01,633 - AWSIoTPythonSDK.core.protocol.i
nternal.workers - DEBUG - Dispatching [puback] event
2019-10-03 22:19:01,657 - AWSIoTPythonSDK.core.protocol.i
nternal.clients - DEBUG - Invoking custom event callback.
..
2019-10-03 22:19:01,658 - AWSIoTPythonSDK.core.protocol.i
nternal.clients - DEBUG - This custom event callback is f
or pub/sub/unsub, removing it after invocation...
Published topic pub: {"message": "UzC", "sequence": 265}
```

```
.internal.workers - DEBUG - Produced [message] event
2019-10-03 22:19:00,512 - AWSIoTPythonSDK.core.protocol
.internal.workers - DEBUG - Dispatching [message] event

Received a new message:
b'{"message": "UzC", "sequence": 264}'
from topic:
pub
--------------


2019-10-03 22:19:00,542 - AWSIoTPythonSDK.core.protocol
.internal.clients - DEBUG - Invoking custom event callb
ack...
2019-10-03 22:19:01,653 - AWSIoTPythonSDK.core.protocol
.internal.workers - DEBUG - Produced [message] event
2019-10-03 22:19:01,654 - AWSIoTPythonSDK.core.protocol
.internal.workers - DEBUG - Dispatching [message] event

Received a new message:
b'{"message": "UzC", "sequence": 265}'
from topic:
pub
--------------


2019-10-03 22:19:01,725 - AWSIoTPythonSDK.core.protocol
.internal.clients - DEBUG - Invoking custom event callb
ack...
```

Figure 4.3: AWS MQTT on Terminal

Chapter 5

Conclusions

The proposed system was developed taking in mind the benefits of the Users and Shops. In this project we presented a system which would help users charge their phone and move around without any hesitation. Even in crowded shops and malls where there is a high chances of phone getting stole. This system will make people worry free and give them freedom to wander and buy items around them and hence increasing shop's revenue. We achieved this by using highly secured django server for User to Server connection and highly secured AWS MQTT for server to station connection. Also data given by the shops to the system will provide users a shopping recommendations and Offers notifications on the shops they visit. Recommendations is always useful for users as well as shops. In the result we can clearly observe that the system not only adapt new problem faced due to increased population but also user friendly recommendation system which keep personal data untouched by any other person.

Bibliography

[1] M.S.Varadarajan, Veltech Dr.RR, Dr.SR Technical University Chennai, India - Coin Based Universal Mobile Battery Charger *IOSR Journal of Engineering (IOSRJEN),* ISSN: 2250-3021 Volume 2, Issue 6 (June 2012), PP 1433-1438.

[2] E. Rammohana Reddy, K. Sankara, Analyzing Consumerization - Internet of Things Based Home Automation Control System Using Raspberry Pi*International Journal of Scientific Research in Computer Science, Engineering and Information Technology,* 2018 IJSRCSEIT — Volume 3 — Issue 4 — ISSN : 2456-3307.

[3] Deepak B. Andore - AWS IOT Platform based Remote Monitoring by using Raspberry Pi*International Journal of Latest Technology in Engineering, Management Applied Science (IJLTEMAS)* , Volume VI, Issue X, October 2017 — ISSN 2278-2540.

[4] 1Nareshkumar R. M., Apoorva Kamat, Dnyaneshvari Shinde - Smart Door Security Control System Using Raspberry Pi ,' *proceeding of the International Journal of Innovations Advancement in Computer Science* , 6, Issue 11 November 2017.

[5] Prof. B Nithya Ramesh, Aashay R Amballi, Vivekananda Mahanta - DJANGO THE PYTHON WEB FRAMEWORK *International Journal of Computer Science and Information Technology Research* , Vol. 6, Issue 2, pp: (59-63), Month: April - June 2018.

[6] Bajorek, Marcin, and Jedrzej Nowak - "The role of a mobile device in a home monitoring healthcare system,*Computer Science and Information Systems (FedCSIS)* , 2011 Federated Conference on. IEEE, 2011.

[7] Tupakula, Udaya, Vijay Varadharajan, and Sunil Kumar Vuppala. - Security Techniques for Beyond 3G Wireless Mobile Networks *Embedded and Ubiquitous Computing (EUC), 2011 IFIP 9th International Conference on. IEEE,* , 2011.

[8] *"Home Automation as a service"*, International Journal of Computer Networks and Wireless Communications (IJCNWC), June 2012.

[9] S. Nazeem Basha, Dr. S.A.K. Jilani - An Intelligent Door System using Raspberry Pi and Amazon Web Services IoT *International Journal of Engineering Trends and Technology (IJETT),* vol. 33, pp. 84- 89,2016.

[10] Rajendra Nayak, Neema Shetty - Secured Smart Home Monitoring System Using Raspberry-PI, *International Journal Innovative Research and Development (IJIRD),*, vol. 5, pp. 339-342,2016.