

BEACON BASED SMART SHOPPING SYSTEM USING IOT

Submitted in partial fulfillment of the requirements for
the award of
Bachelor of Engineering degree in Computer Science and Engineering
by

YADARAJU DIVYA VEERA VENKATA S

JIYYAPARAJU(Reg.No.37110856)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

**SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI-600 119**

MARCH-2021



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with "A" grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai-600119
www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **YADARAJU DIVYA VEERA VENKATA S JIYYAPARAJU (Reg.No.37110856)** who carried out the project entitled "**BEACON BASED SMART SHOPPING SYSTEM USING IOT**" under my supervision from **November 2020** to **March 2021**.

Internal Guide

Dr.NAGARAJAN.G, M.E., Ph.D.

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

YADARAJU DIVYA VEERA VENKATA S JIYYAPARAJU hereby declare that the Project Report entitled “**BEACON BASED SMART SHOPPING SYSTEM USING IOT**” is done by me under the guidance of **Dr.NAGARAJAN.G, M.E., Ph.D.,** Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE:CHENNAI

SIGNATUREOFTHECANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D., and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.NAGARAJAN.G, M.E., Ph.D., Assistant Professor**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Localization is one main and important part of Internet of Things (IoT) where the Location of Everything (LoE) it plays a major role to develop most services in IoT area. On the other hand, data mining techniques are essential and analyses when we have big data through IoT platforms. Indeed, integration of location-based methods and data mining analysis process can make a smart system service for IoT structure and applications. For this use, we design a smart shopping platform including four modules, location of every component, data collection of components, data analyzing component and data mining component. Then accurate localization scheme named “location orbital” is evolved that estimates the current location of mobile objects (users) based on both current and the former places or areas.

TABLE OF CONTENTS

ABSTRACT	v
LIST OF FIGURES	viii

CHAPTER NO	TITLE	PAGE NO
1	INTRODUCTION	1
	1.1 OVERVIEW	1
	1.2 SCOPE OF THE PROJECT	1
	1.3 DOMAIN OVERVIEW	2
	1.3.1 INTERNET OF THINGS	2
	1.3.2 IOT-KEY FEATURES	2
	1.3.3 IOT SOFTWARE	3
2	LITERATURE SURVEY	5
	2.1 LITERATURE SURVEY 1	5
	2.2 LITERATURE SURVEY 2	5
	2.3 LITERATURE SURVEY 3	6
	2.4 LITERATURE SURVEY 4	7
3	AIM AND SCOPE OF THE PRESENT INVESTIGATION	8
	3.1 AIM	8
	3.2 SCOPE	8
4	SYSTEM DESIGN AND METHODOLOGY	9
	4.1 EXISTING SYSTEM	9
	4.2 PROPOSED SYSTEM	9
	4.2.1 ADVANTAGES OF PROPOSED SYSTEM	9
	4.2.2 DISADVANTAGES OF PROPOSED SYSTEM	9
	4.3 REQUIRMENTS SPECIFICATIONS	10
	4.3.1 HARDWARE REQUIRMENTS	10

4.3.2 SOFTWARE REQUIRMENTS	10
4.4 SOFTWARE DESCRIPTION	11
4.4.1 JAVA	11
4.4.2 INTRODUCTION TO JAVA	11
4.4.3 WORKING OF JAVA	12
4.4.4 THE JAVA PROGRAMMING LANGUAGE	12
4.4.4.1 THE JAVA PLATFORM	14
4.4.5 APACHE TOMCAT SERVER	15
4.4.6 ANDROID INTRODUCTION	17
4.4.6.1 LINUX KERNEL	19
4.4.6.2 LIBRARIES	19
4.4.6.3 ANDROID RUNTIME	19
4.4.6.4 APPLICATION FRAME WORK	19
4.4.6.5 APPLICATIONS	20
4.5 SYSTEM DESIGN	20
4.5.1 ARCHITECTURE DIAGRAM	21
4.5.2 SEQUENCE DIAGRAM	21
4.5.3 USE CASE DIAGRAM	22
4.5.4 ACTIVITY DIAGRAM	23
4.5.5 COLLABORATION DIAGRAM	24
4.5.6 DATA FLOW DIAGRAM	24
4.5.7 CLASS DIAGRAM	25
4.6 SYSTEM IMPLEMENTATION	26
4.6.1 LIST OF MODULES	26
4.6.2 LOCATION OF EVERYTHING COMPONENT	26
4.6.3 DATA COLLECTION COMPONENT	26
4.6.4 DATA FILTERATION / ANALYZING COMPONENT	26
4.6.5 DATA MINING COMPONENT & LOCATION-BASED SUGGESTIONS	27

5	RESULTS AND DISCUSSION	28
	5.1 RESULT	28
6	CONCLUSION AND FUTURE WORK	32
	REFERENCES	33
	APPENDIX	35
	A) SOURCE CODE	35

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
4.4.4	WORKING OF JAVA COMPILER	14
4.4.4.1	JAVA PLATFORM	19
4.5.1	ARCHITECTURE DIAGRAM	21
4.5.2	SEQUENCE DIAGRAM	22
4.5.3	USE CASE DIAGRAM	23
4.5.4	ACTIVITY DIAGRAM	23
4.5.5	COLLABORATION DIAGRAM	24
4.5.6	DATA FLOW DIAGRAM	25
4.5.7	CLASS DIAGRAM	25
5.1	USER REGISTRATION	28
5.2	USER LOGIN PAGE	28
5.3	SELECTING AND ADDING ITEMS TO CART	29
5.4	SELECTING LOCATION FOR SHOPPING AND OFFER CHECK	29
5.5	SCANNING PRODUCT QR CODE FOR BILL	30
5.6	BILL FOR THE ITEMS SHOPPED	30
5.7	SELECTION PAYMENT OPTION	31
5.8	SUCCESSFUL COMPLETION OF PAYMENT	31

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Outsourcing computation to the cloud is now common practice, and it is not surprising that the Security-as-a-Service (SECaaS) paradigm has arisen to counter the growing level of cyber threats. These services may take various forms, such as secure data storage, identity and access management (IAM), and intrusion detection services to screen incoming traffic. These services can be provisioned in a similar manner to other cloud services, either through advance subscription or dynamically through on-demand options.

Localization is one important part of Internet of Things (IoT) where the Location of Everything (LoE) system plays a important role to improve most services in IoT area. On the other hand, data mining techniques are essential analyses when we have big data from IoT platforms. Indeed, integration of location-based methods and data mining analysis process can make a smart system service for IoT scenarios and applications. For this purpose, we design a smart shopping platform including four components, location of everything component, data collection component, data filtering/analyzing component and data mining component. Then a novel accurate localization scheme named “location orbital” is developed that estimates the current location of mobile objects (users or everything) based on both current and the previous locations.

1.2 SCOPE OF THE PROJECT

Location of Everything (LoE) is promising method for improving IoT scenarios and applications. This research designed a smart shopping mall platform based on four components include location of everything component, data collection component, data filtering/analyzing component and data mining component.

In addition, we developed a novel accurate localization scheme named “location orbital” that estimates the current location of mobile objects (users or everything). Finally, an implementation of the experiment in a store was conducted to examine performance evaluation of the location-based scheme.

Benefits

- Improve the shopping experience for all the patrons of the store
- Increase efficiency of the checkout process
- Eliminate long waiting queues at the checkout counter

Features

- Automated shopping item detection
- Automated communication system to make payments at counter
- Automated data formatting and organization for use by the shopper

1.3 DOMAIN OVERVIEW

1.3.1 Internet Of Things

IoT (Internet of Things) is an advanced automation and analytics system which exploits networking, sensing, big data, and artificial intelligence technology to deliver complete systems for a product or service. These systems allow greater transparency, control, and performance when applied to any industry or system.

IoT systems have applications across industries through their unique flexibility and ability to be suitable in any environment. They enhance data collection, automation, operations, and much more through smart devices and powerful enabling technology.

IoT systems allow users to achieve deeper automation, analysis, and integration within a system. They improve the reach of these areas and their accuracy. IoT utilizes existing and emerging technology for sensing, networking, and robotics.

IoT exploits recent advances in software, falling hardware prices, and modern attitudes towards technology. Its new and advanced elements bring major changes in the delivery of products, goods, and services; and the social, economic, and political impact of those changes.

1.3.2 IoT – Key Features

The most important features of IoT include artificial intelligence, connectivity, sensors, active engagement, and small device use. A brief review of these features is given below –

- **AI** – IoT essentially makes virtually anything “smart”, meaning it enhances every aspect of life with the power of data collection, artificial intelligence

algorithms, and networks. This can mean something as simple as enhancing your refrigerator and cabinets to detect when milk and your favorite cereal run low, and to then place an order with your preferred grocer.

- **Connectivity** – New enabling technologies for networking, and specifically IoT networking, mean networks are no longer exclusively tied to major providers. Networks can exist on a much smaller and cheaper scale while still being practical. IoT creates these small networks between its system devices.
- **Sensors** – IoT loses its distinction without sensors. They act as defining instruments which transform IoT from a standard passive network of devices into an active system capable of real-world integration.
- **Active Engagement** – Much of today's interaction with connected technology happens through passive engagement. IoT introduces a new paradigm for active content, product, or service engagement.
- **Small Devices** – Devices, as predicted, have become smaller, cheaper, and more powerful over time. IoT exploits purpose-built small devices to deliver its precision, scalability, and versatility.

1.3.3 IoT Software

IoT software addresses its key areas of networking and action through platforms, embedded systems, partner systems, and middleware. These individual and master applications are responsible for data collection, device integration, real-time analytics, and application and process extension within the IoT network. They exploit integration with critical business systems (e.g., ordering systems, robotics, scheduling, and more) in the execution of related tasks.

- **Data Collection**

This software manages sensing, measurements, light data filtering, light data security, and aggregation of data. It uses certain protocols to aid sensors in connecting with real-time, machine-to-machine networks. Then it collects data from multiple devices and distributes it in accordance with settings. It also

works in reverse by distributing data over devices. The system eventually transmits all collected data to a central server.

- **Device Integration**

Software supporting integration binds (dependent relationships) all system devices to create the body of the IoT system. It ensures the necessary cooperation and stable networking between devices. These applications are the defining software technology of the IoT network because without them, it is not an IoT system. They manage the various applications, protocols, and limitations of each device to allow communication.

- **Real-Time Analytics**

These applications take data or input from various devices and convert it into viable actions or clear patterns for human analysis. They analyze information based on various settings and designs in order to perform automation-related tasks or provide the data required by industry.

- **Application and Process Extension**

These applications extend the reach of existing systems and software to allow a wider, more effective system. They integrate predefined devices for specific purposes such as allowing certain mobile devices or engineering instruments access. It supports improved productivity and more accurate data collection.

CHAPTER 2

2.1 LITERATURE SURVEY 1

Title	IoT APPLICATIONS ON SECURE SMART SHOPPING SYSTEM
Authors	Mrs. Devi Ramakrishnan ¹ , Pavithira, Monisha, Likitha
Published Year	2018
Description	In today's life going to supermarkets for shopping is increasing rapidly. People take the item and put it into trolley. After done with shopping they go for billing at the Billing counter but as there are many people standing in Queue for billing purpose, So lots of time is required for the individuals for billing because of existing barcode technology. To reduce this time we are proposed a system based on RFID technology. The system contains the items attached with RFID tag, RFID reader which reads the tag information when put into the trolley. Then this information is send to main billing server which calculates the total amount of purchased items and sends the calculated bill to the device attached to trolley for displaying it on LCD. Along with this system we are implementing an Android application for controlling the trolley movements. The application is based on the Trolley number and total amount of purchased items.

2.2 LITERATURE SURVEY 2

Title	Where Buy: A Location-based Shopping App with Products-wise Searching
Authors	Kin Chi Chan, Tak Leung Cheung, Siu Hong Lai,
Published Year	2017
Description	It is usual for a consumer to search a product based on its category and go to related kind of shop to buy a product, e.g. food in

	supermarket, a pencil from a stationary shop and etc. While it is not uncommon nowadays for a shop to sell various categories of goods at the same time, like a newspaper stand does sell toys, an accessory shop has stationary. However, consumer may not easily notice and purchase these goods, especially if they are in hurry or not familiar with the shops nearby. With the emergence and popularity of many shopping search engines (shop bots), we can actually provide a better matching between the consumer and seller. In this paper, we developed a shopbot app system (Where2Buy) on smart phone that can search and filter the nearby shops which sell the desired products. To simplify the input process, our system allows users to search by text or voice and fuzzy matching is supported to widen the scope of searching. Detailed information of related product and shops are displayed in the result, together with a navigation map showing the best route to the target shops. If the desired product is not available nearby, substitutes in the same category will be recommended for the users. Our user study evidences that our system is simply and easy to use. More than half of the participants prefer Where2Buy than the other available shopbots in Hong Kong.
--	---

2.3 LITERATURE SURVEY 3

Title	Smart Phone Based Robotic Control for Surveillance Applications
Authors	M.Selvam
Published year	2014
Description	The robotics and automation industry which is ruled the sectors from manufacturing to household entertainments. It is widely used because of its simplicity and ability to modify to meet changes of needs. The project is designed to develop a robotic vehicle using android application for remote operation attached with wireless camera for monitoring purpose. The robot along with camera can wirelessly transmit real time video with night vision capabilities. This

	is kind of robot can be helpful for spying purpose in war fields.
--	---

2.4 LITERATURE SURVEY 4

Title	Smart Mart: IoT-based In-store Mapping for Mobile Devices
Authors	Dylan Hicks, Kevin Mannix, Hannah M. Bowles, Byron J.
Published Year	2013
Description	Quite often, when shopping in a supermarket (e.g. Walmart), shoppers are frustrated at locating the items on the shopping list and no assistance is available. On the other hand, retailers also lose a large volume of sales as a result. In this paper, we present a feasibility study that leverages the Internet of Things (IoT) technology to make store items “smart” so that they can automatically register and update their location information in an information retrieval system, allowing shoppers to search, locate, and map them on the store floor plan using mobile devices. A free-accessible Android-based mobile app Smart Mart has been developed to demonstrate the promise of this preliminary work. Continuous development of this research could lead to a complete change in our day-to-day shopping experience.

CHAPTER 3

AIM AND SCOPE OF THE PRESENT INVESTIGATION

3.1 AIM

The aim of the project is that to design a location-based real time shopping system to provide an efficient IoT based advanced recommendation service for customers and managers by using four components, including location of everything component, data collection component, data filtering/analysing component and data mining component, and develop a novel accurate localization scheme called location orbital that estimates the current location of mobile objects based on both current and the previous locations, and test and implementation of the experiment in a store to practically examine performance evaluation of the location-based scheme.

3.2 SCOPE

The Scope of the project is that the Location of Everything (LoE) is promising method for improving IoT scenarios and applications. This research designed a smart shopping mall platform based on four components include location of everything component, data collection component, data filtering/analyzing component and data mining component.

In addition, we developed a novel accurate localization scheme named "location orbital" that estimates the current location of mobile objects (users or everything). Finally, an implementation of the experiment in a store was conducted to examine performance evaluation of the location-based scheme.

Benefits

- Improve the shopping experience for all the patrons of the store
- Increase efficiency of the checkout process
- Eliminate long waiting queues at the checkout counter

Features

- Automated shopping item detection
- Automated communication system to make payments at counter
- Automated data formatting and organization for use by the shopper

CHAPTER 4

SYSTEM DESIGN & METHODOLOGY

4.1 EXISTING SYSTEM

In the existing system users don't have any suggestions about shopping area, they used to search manually and they will go to that place. Here the major drawback is what if the user need product is not there, and what if the product in that particular shop is very expensive than the same product which is available in another shopping markets. And also user doesn't aware of the offers nearer to him. And if user goes to new place he doesn't know where are the shopping malls which he needs to buy in that shop.

4.2 PROPOSED SYSTEM

Location of Everything (LoE) is promising method for improving IoT scenarios and applications. This research designed a smart shopping mall platform based on four components include location of everything component, data collection component, data filtering/analyzing component and data mining component. In addition, we developed a novel accurate localization scheme named "location orbital" that estimates the current location of mobile objects (users or everything). Finally, an implementation of the experiment in a store was conducted to examine performance evaluation of the location-based scheme.

4.2.1 ADVANTAGES OF PROPOSED SYSTEM

- To simplify the input process, our system allows users to search by text or voice and fuzzy matching is supported to widen the scope of searching.
- If the desired product is not available nearby, substitutes in the same category will be recommended for the users.
- This process saves the time of customer and also reduced the man power in the malls.

4.2.2 DISADVANTAGES OF PROPOSED SYSTEM

- It becomes difficult for the store to do inventory management as all items can be automatically read and easily logged on to.

- Another limitation is that we do not consider if the product is out of stock. If the user wants to buy a product urgently, the stock can be an important criteria for consumers.
- Manual registration of dynamic locations for hundreds of thousands of items can be very labor-intensive and error-prone.
- The popular global navigation satellite systems (GPS) are generally not suitable to establish indoor locations, since microwaves will be attenuated and scattered by roofs, walls and other objects.

4.3 REQUIREMENT SPECIFICATION

This proposed system runs effectively on a computing system that has the minimum requirements

4.3.1 HARDWARE REQUIREMENT

The basic hardware required to run the program are:

- 1 Hard Disk : 80GB and Above
- 2 RAM : 4GB and Above
- 3 Processor : P IV and Above

4.3.2 SOFTWARE REQUIREMENT

The basic software requirements to run the program are:

- 1 Windows 7 and above
- 2 JDK 1.7
- 3 J2EE
- 4 Tomcat 7.0
- 5 MySQL

4.4 SOFTWARE DESCRIPTION

4.4.1 JAVA

Java is an object-oriented programming language developed initially by James Gosling and colleagues at Sun Microsystems. The language, initially called Oak (named after the oak trees outside Gosling's office), was intended to replace C++, although the feature set better resembles that of Objective C.

4.4.2 INTRODUCTION TO JAVA

Java has been around since 1991, developed by a small team of Sun Microsystems developers in a project originally called the Green project. The intent of the project was to develop a platform-independent software technology that would be used in the consumer electronics industry. The language that the team created was originally called Oak.

Around the time the First Person project was floundering in consumer electronics, a new craze was gaining momentum in America; the craze was called "Web surfing." The World Wide Web, a name applied to the Internet's millions of linked HTML documents was suddenly becoming popular for use by the masses. The reason for this was the introduction of a graphical Web browser called Mosaic, developed by ncSA. The browser simplified Web browsing by combining text and graphics into a single interface to eliminate the need for users to learn many confusing UNIX and DOS commands. Navigating around the Web was much easier using Mosaic.

Web surfing has become an enormously popular practice among millions of computer users. Until Java, however, the content of information on the Internet has

been a bland series of HTML documents. Web users are hungry for applications that are interactive, that users can execute no matter what hardware or software platform they are using, and that travel across heterogeneous networks and do not spread viruses to their computers. Java can create such applications.

4.4.3 WORKING OF JAVA

Those who are new to object-oriented programming, the concept of a class will be new to you. Simplistically, a class is the definition for a segment of code that can contain both data (called attributes) and functions (called methods).

When the interpreter executes a class, it looks for a particular method by the name of **main**, which will sound familiar to C programmers. The main method is passed as a parameter an array of strings (similar to the `argv []` of C), and is declared as a static method.

To output text from the program, we execute the **println** method of **System.out**, which is java's output stream. UNIX users will appreciate the theory behind such a stream, as it is actually standard output. For those who are instead used to the Wintel platform, it will write the string passed to it to the user's program.

Java consists of two things :

- Programming language
- Platform

4.4.4 THE JAVA PROGRAMMING LANGUAGE

Java is a high-level programming language that is all of the following:

- Simple
- Object-oriented
- Distributed
- Interpreted
- Robust
- Secure
- Architecture-neutral
- Portable
- High-performance
- Multithreaded
- Dynamic

The code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

Java is unusual in that each Java program is both compiled and interpreted. With a compiler, you translate a Java program into an intermediate language called **Java byte codes** – the platform independent codes interpreted by the Java interpreter. With an interpreter, each Java byte code instruction is parsed and run on the computer. Compilation happens just once; interpretation occurs each time the program is executed. This figure illustrates how it works:

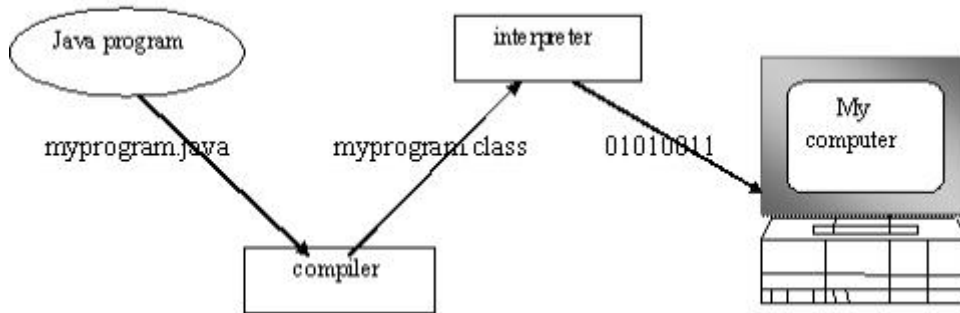


Fig.4.4.4 Working of Java Compiler

You can think of Java byte codes as the machine code instructions for the **Java Virtual Machine (JVM)**. Every Java interpreter, whether it's a Java development tool or a Web browser that can run Java applets, is an implementation of JVM. That JVM can also be implemented in hardware. Java byte codes help make “write once, run anywhere” possible.

You can compile your Java program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the JVM. For example, that same Java program can be run on Windows NT, Solaris and Macintosh

4.4.4.1 THE JAVA PLATFORM

A platform is the hardware or software environment in which a program runs. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other, hardware-based platforms. Most other platforms are described as a combination of hardware and operating system.

The Java platform has two components :

- The Java Virtual Machine (JVM)
- The Java Application Programming Interface (Java API)

You've already been introduced to the JVM. It's the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries **(packages)** of related components. The following figure depicts a Java program, such as an application or applet, that's running on the Java platform. As the figure shows, the Java API and Virtual Machine insulates the Java program from hardware dependencies.

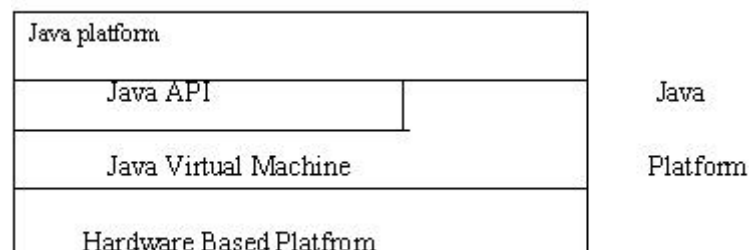


Fig.4.4.4.1 Java Platform

As a platform-independent environment, Java can be a bit slower than native code. However, smart compilers, weel-tuned interpreters, and just-in-time byte compilers can bring Java's performance close to that of native code without threatening portability.

4.4.5 APACHE TOMCAT SERVER

Apache Tomcat (formerly under the Apache Jakarta Project; Tomcat is now a top level project) is a web container developed at the Apache Software Foundation. Tomcat implements the servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, providing an environment for Java code to

run in cooperation with a web server. It adds tools for configuration and management but can also be configured by editing configuration files that are normally XML-formatted. Because Tomcat includes its own HTTP server internally, it is also considered a standalone web server.

Environment

Tomcat is a web server that supports servlets and JSPs. Tomcat comes with the Jasper compiler that compiles JSPs into servlets.

The Tomcat servlet engine is often used in combination with an Apache web server or other web servers. Tomcat can also function as an independent web server. Earlier in its development, the perception existed that standalone Tomcat was only suitable for development environments and other environments with minimal requirements for speed and transaction handling. However, that perception no longer exists; Tomcat is increasingly used as a standalone web server in high-traffic, high-availability environments.

Since its developers wrote Tomcat in Java, it runs on any operating system that has a JVM.

Product features

Tomcat 3.x (initial release)

- implements the Servlet 2.2 and JSP 1.1 specifications
- servlet reloading
- basic HTTP functionality Tomcat 4.x
- implements the Servlet 2.3 and JSP 1.2 specifications
- servlet container redesigned as Catalina
- JSP engine redesigned as Jasper
- Coyote connector

- Java Management Extensions (JMX), JSP and Struts-based administration
- Tomcat 5.x
- implements the Servlet 2.4 and JSP 2.0 specifications
- reduced garbage collection, improved performance and scalability
- native Windows and Unix wrappers for platform integration
- faster JSP parsing

History

Tomcat started off as a servlet specification implementation by James Duncan Davidson, a software architect at Sun. He later helped make the project open source and played a key role in its donation by Sun to the Apache Software Foundation.

Davidson had initially hoped that the project would become open-sourced and, since most open-source projects had O'Reilly books associated with them featuring an animal on the cover, he wanted to name the project after an animal. He came up with Tomcat since he reasoned the animal represented something that could take care of and fend for itself. His wish to see an animal cover eventually came true when O'Reilly published their Tomcat book with a tomcat on the cover.

4.4.6 Android Introduction:

Android is a Linux based operating system it is designed primarily for touch screen mobile devices such as smart phones and tablet computers. The operating system has developed a lot in last 15 years starting from black and white phones to recent smart phones or mini computers. One of the most widely used mobile OS these days is android. The android is software that was founded in Palo Alto of

California in 2003.

The android is a powerful operating system and it supports large number of applications in Smartphones. These applications are more comfortable and advanced for the users. The hardware that supports android software is based on ARM architecture platform. The android is an open source operating system means that it's free and any one can use it. The android has got millions of apps available that can help you managing your life one or other way and it is available low cost in market at that reasons android is very popular.

The android is a operating system and is a stack of software components which is divided into five sections and four main layers that is

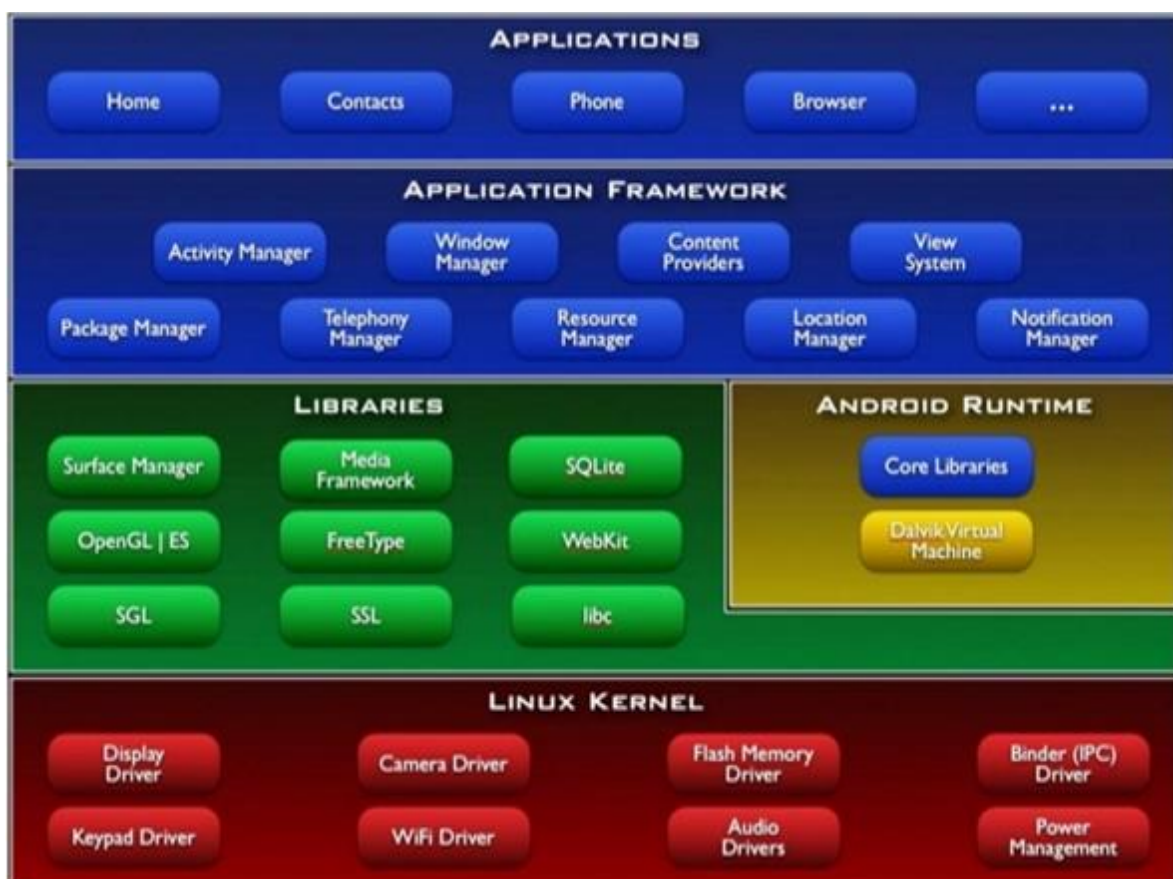


Fig.4.4.6 Android Architecture

4.4.6.1 Linux kernel:

The android uses the powerful Linux kernel and it supports wide range of hardware drivers. The kernel is the heart of the operating system that manages input and output requests from software. This provides basic system functionalities like process management, memory management, device management like camera, keypad, display etc the kernel handles all the things. The Linux is really good at networking and it is not necessary to interface it to the peripheral hardware. The kernel itself does not interact directly with the user but rather interacts with the shell and other programs as well as with the hard ware devices on the system.

4.4.6.2 Libraries:

The on top of a Linux kennel there is a set of libraries including open source web browser such as webkit, library libc. These libraries are used to play and record audio and video. The SQLite is a data base which is useful for storage and sharing of application data. The SSL libraries are responsible for internet security etc.

4.4.6.3 Android Runtime:

The android runtime provides a key component called Dalvik Virtual Machine which is a kind of java virtual machine. It is specially designed and optimized for android. The Dalvik VM is the process virtual machine in the android operating system. It is software that runs apps on android devices.

The Dalvik VM makes use of Linux core features like memory management and multithreading which is in a java language. The Dalvik VM enables every android application to run it own process. The Dalvik VM executes the files in the .dex format.

4.4.6.4 Application frame work:

The application frame work layer provides many higher level services to

applications such as windows manager, view system, package manager, resource manager etc. The application developers are allowed to make use of these services in their application.

4.4.6.5 Applications:

You will find all the android applications at the top layer and you will write your application and install on this layer. Examples of such applications are contacts, books, browsers, services etc. Each application performs a different role in the overall applications.

Advantages:

- Android is Linux based open source operating system , it can be developed by any one
- Easy access to the android apps
- You can replace the battery and mass storage, disk drive and UDB option
- Its supports all Google services
- The operating system is able to inform you of a new SMS and Emails or latest updates.
- It supports Multitasking
- Android phone can also function as a router to share internet
- Its free to customize
- Can install a modified ROM
- Its supports 2D and 3D graphics

4.5 SYSTEM DESIGN

In this section we present implementation, experimental evaluation of the system. We conduct our experiments in a store and used indoor positioning technologies iBeacons. The iBeacon standard is developed by Apple and its app run on Android and iOS. The iBeacons use BLE to transmit data over short

distances.

In this experiment, four iBeacons are used in the store. The iBeacon signal interval used in the experiment was 1285ms and broadcasting power set as 12dBm. This is the default settings for iBeacon. In this experiment, time is divided into slots and each time slot is equal to 1 second close to iBeacon signal interval. The useful communication range (r) of iBeacon nodes is 10 m ($RSS < -90$ dBm). The measured received signal strength from iBeacons collected by the user's smart phone as mobile object that move through the area.

4.5.1 ARCHITECTURE DIAGRAM

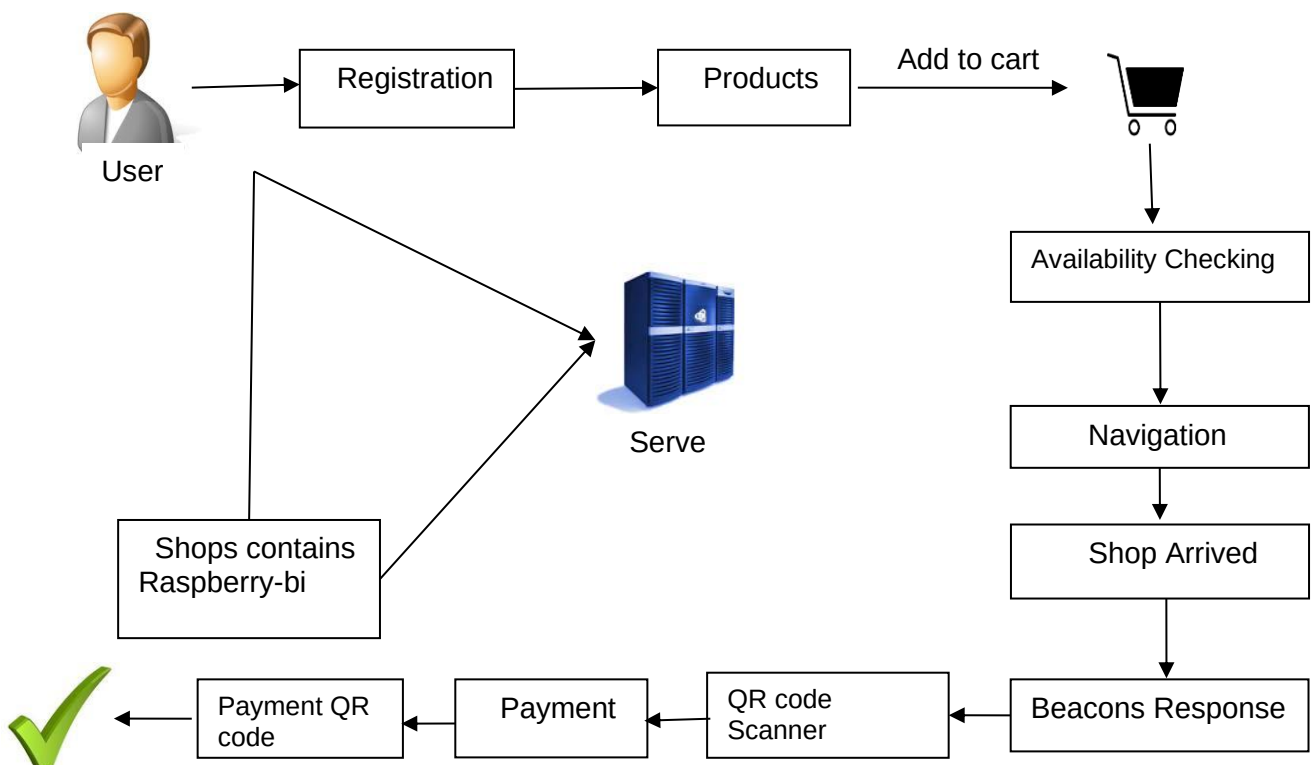


Fig 4.5.1 Architecture Diagram

4.5.2 Sequence Diagram:

A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message

Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.

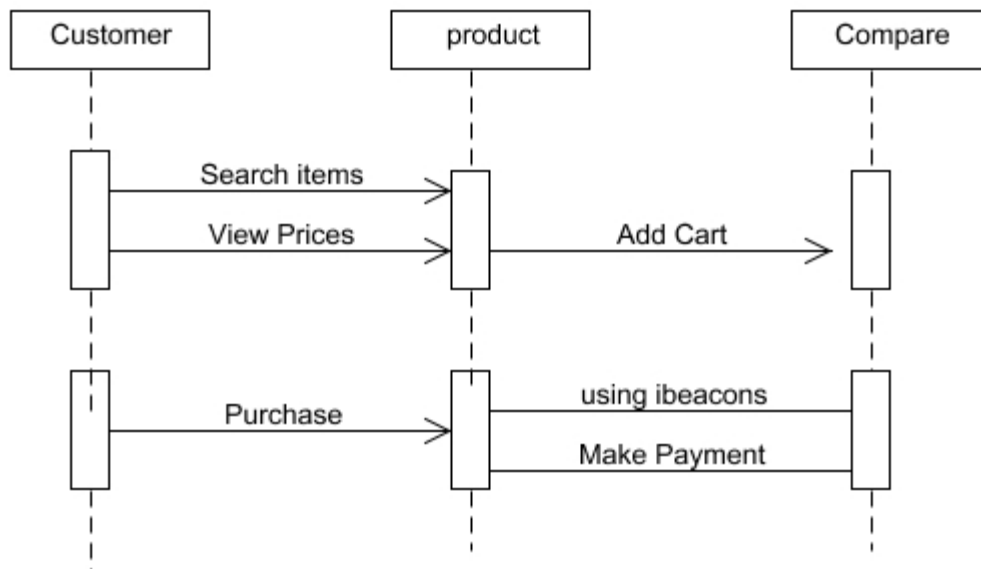


Fig 4.5.2 Sequence Diagram

4.5.3 Use Case Diagram:

Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group. UML includes a set of graphic notation techniques to create visual models of software intensive systems. This language is used to specify, visualize, modify, construct and document the artifacts of an object oriented software intensive system under development.

A Use case Diagram is used to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases.

Use case diagram consists of two parts:

Use case: A use case describes a sequence of actions that provided something of measurable value to an actor and is drawn as a horizontal ellipse.

Actor: An actor is a person, organization or external system that plays a role in one or more interaction with the system.

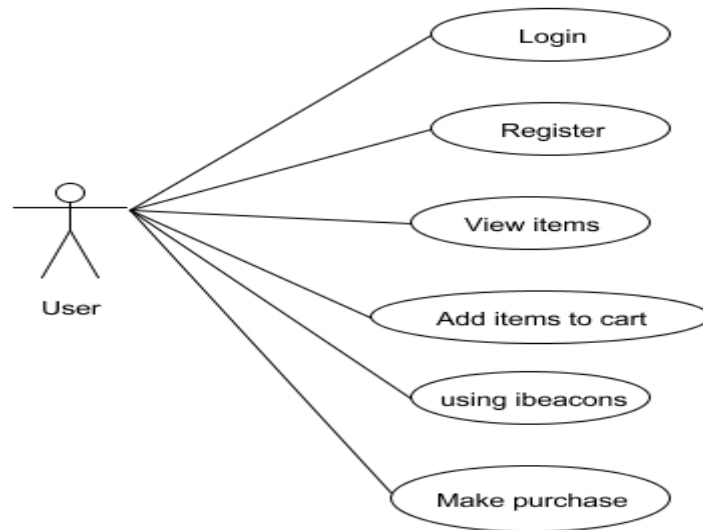


Fig 4.5.3 Use Case Diagram

4.5.4 Activity Diagram:

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

The most important shape types:

- Rounded rectangles represent activities.
- Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.
- A black circle represents the start of the workflow.
- An encircled circle represents the end of the workflow.



Fig 4.5.4 Activity Diagram

4.5.5 Collaboration Diagram:

UML Collaboration Diagrams illustrate the relationship and interaction between software objects. They require use cases, system operation contracts and domain model to already exist. The collaboration diagram illustrates messages being sent between classes and objects.

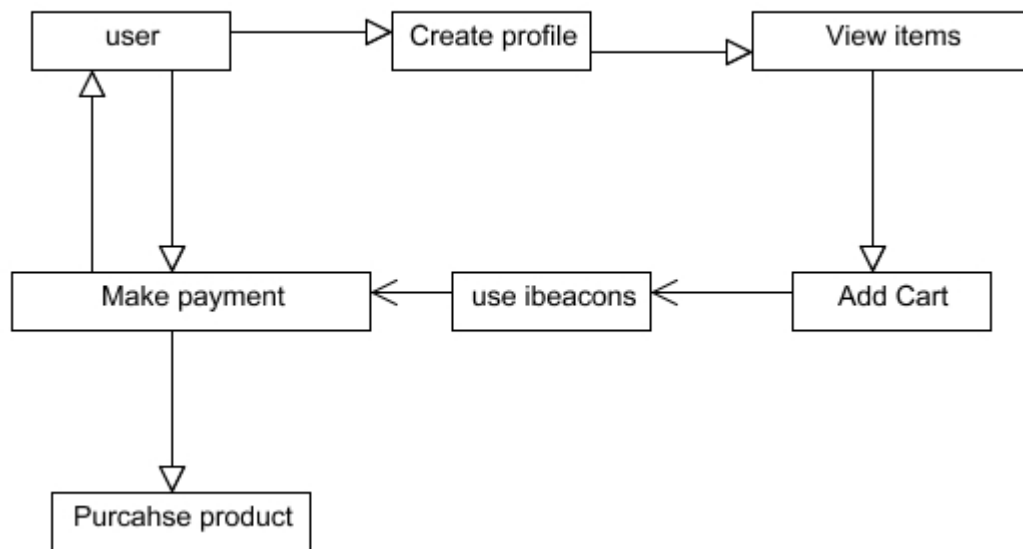
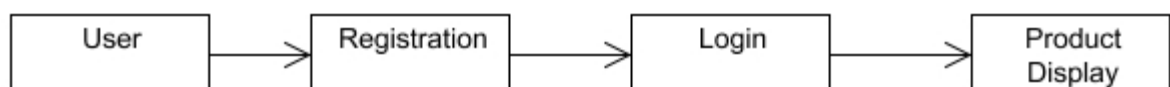


Fig 4.5.5 Collaboration Diagram

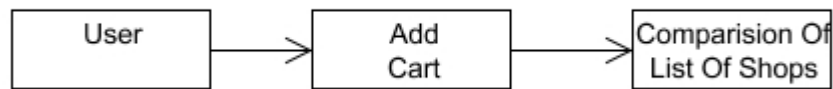
4.5.6 DATA FLOW DIAGRAM:

A Data Flow Diagram (DFD) is a graphical representation of the “flow” of data through an information system, modeling its aspects. It is a preliminary step used to create an overview of the system which can later be elaborated DFDs can also be used for visualization of data processing.

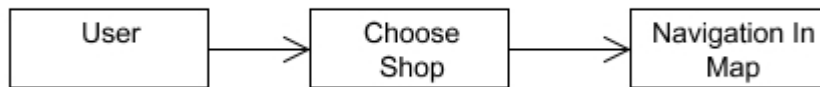
Level 0:



Level 1:



Level 2:



Level 3:

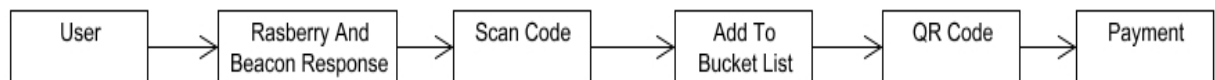


Fig 4.5.6 Data Flow Diagram

4.5.7 Class Diagram

A Class diagram in the Unified Modelling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

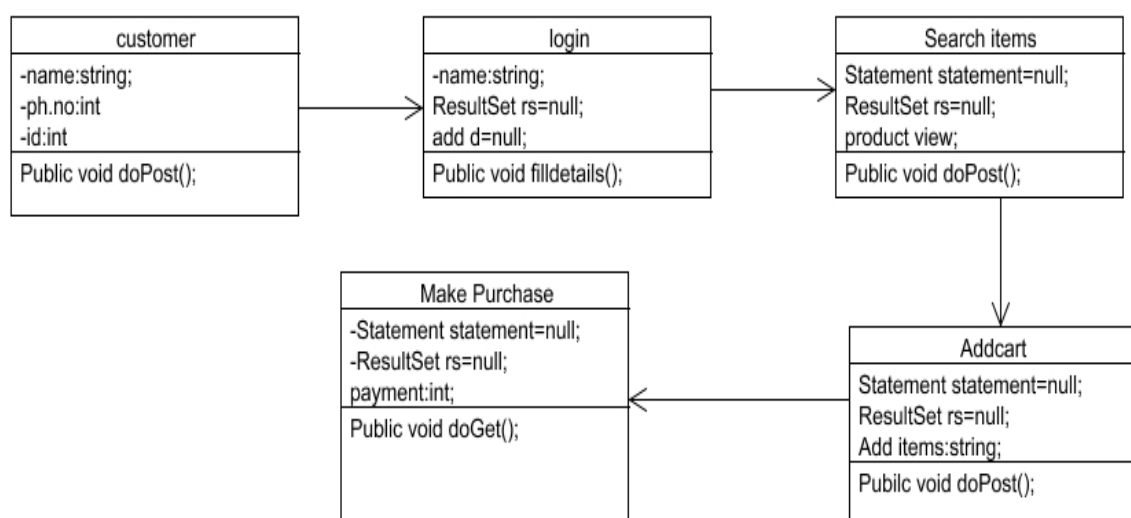


Fig 4.5.7 Class Diagram

4.6 SYSTEM IMPLEMENTATION

4.6.1 LIST OF MODULES

- Location of Everything Component
- Data collection Component
- Data Filtering/Analyzing Component
- Data Mining Component & Location-based Suggestions

4.6.2 Location of Everything Component

This component provides location of devices, goods, staff, customers and any objects that have a role in the system. Locations of everything are recorded at the central server and the system can track and navigate all objects based on an accurate localization mechanism.

4.6.3 Data collection Component

In this component two groups of data will be collected. The first group is user's data (location, interest, social connections etc). The second group is shopping mall relevant data (stores, products, offers, etc.).

Online central server collects data from customers smart mobile phone (crowd sourcing) and sensors with IoTs devices. Some complementary sources may be available in particular services (like wifi access point) which will be considered as well

4.6.4 Data Filtering/Analyzing Component

This component provides pre-processing on the collected data to unify the format as well as filtering unreliable data (e.g., faulty signals from sensors). Analyzing phase is performed on a trustable, useful content to enhance the output of the system. Thus, different sampling methods such as random sampling and stratified sampling are tested.

A distributed parallel analyzing system is established while the collected data is

various in natures. NoSQL database is utilized to address the analyzing challenges of the large volume of the data. Different sources of data are analyzed separately. One of the main tasks in this module is to build profiles of customers and keep their historical purchases for future recommendations. In addition, this component will ensure privacy of users is guaranteed.

4.6.5 Data Mining Component & Location-based Suggestions

The designed service is specified based on the real time shopping application. In this component, an efficient data mining and machine learning technique is performed on all data filtering component. Thus, the output is recommendation/ prediction services for end users (customers) and advanced services for service providers/businesses as well as the planning and decision making.

The localization scheme in this research is developed for mobile sensor/object like customers. In a mobile network, every object needs to get its location periodically. Finally the estimated location in current time-slot can be calculated by averaging the coordinates of location orbital. This phase includes two steps; firstly the mobile object should select the optimum numbers of signal samples.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 RESULT

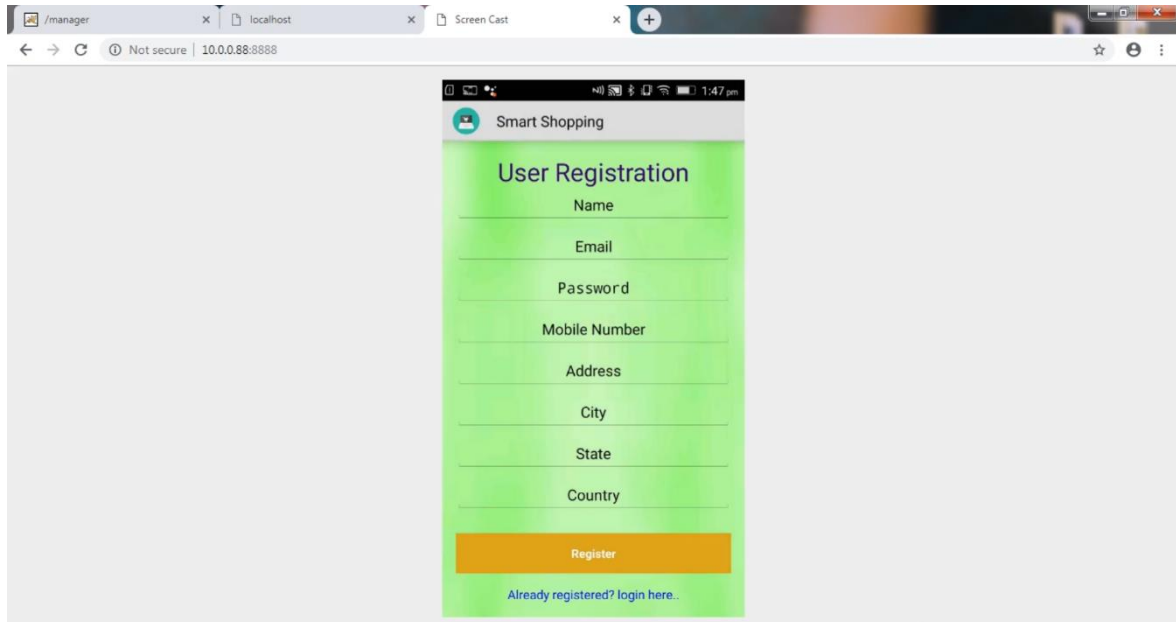


Fig 5.1 User Registration

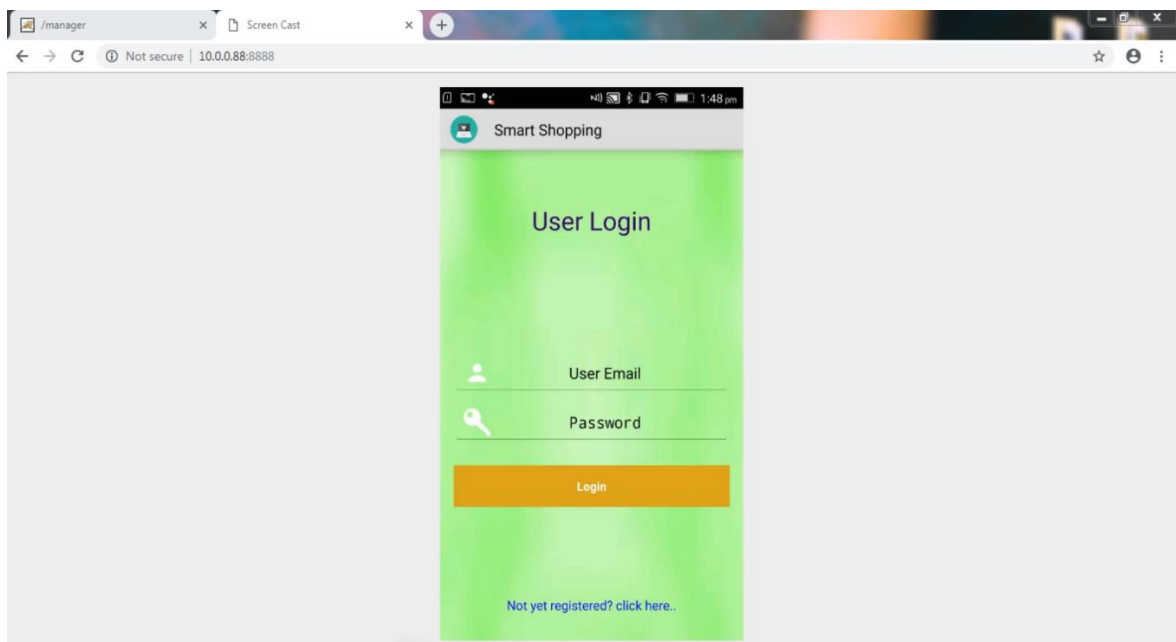


Fig 5.2 User Login Page

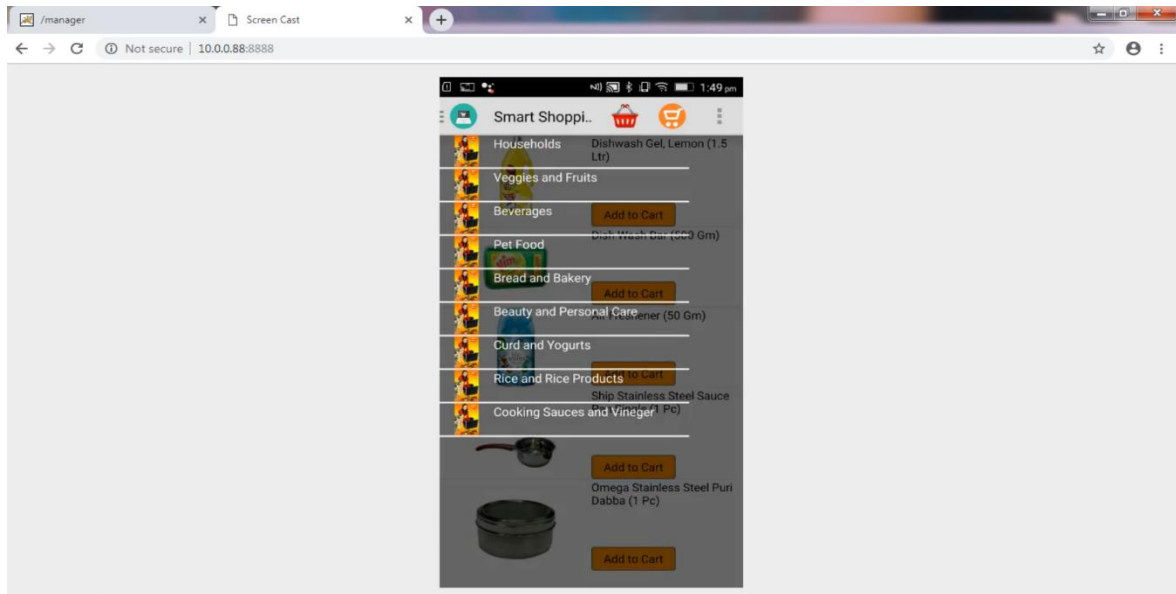


Fig 5.3 Selecting and adding items to Cart

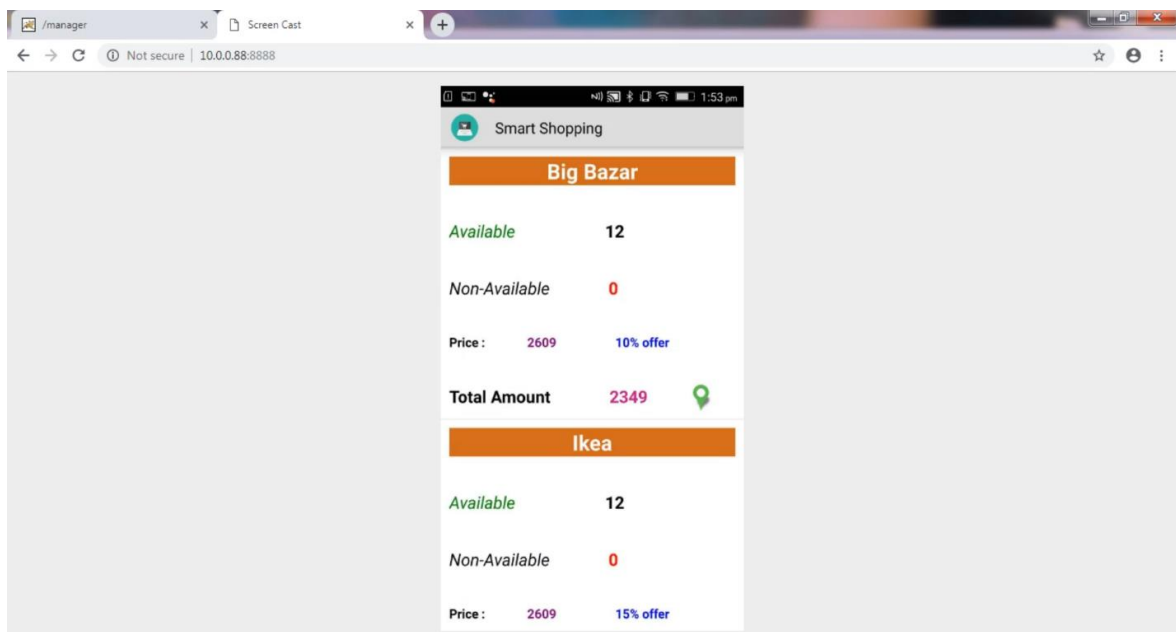


Fig 5.4 Selecting Location for Shopping and Offer Check

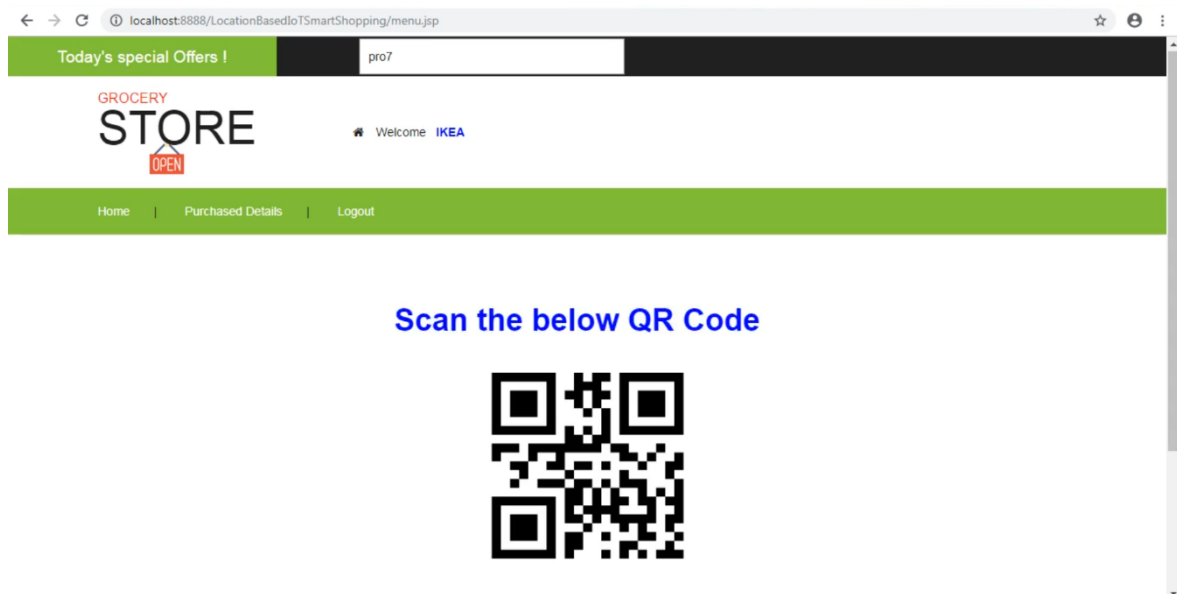


Fig 5.5 Scanning Product QR code for Bill

S.No	Product Id	Product Name	Count	Price
1.	pro4	Fresh Bananas (1 Kg)	3	217
2.	pro5	Fresh Apple Red (1 Kg)	2	306
3.	pro6	Fresh Strawberry (1 Kg)	2	340
4.	pro7	Fresh Brinjal Bharta (1 Kg)	1	47
5.	pro1	Dishwash Gel, Lemon (1.5 Ltr)	1	213
6.	pro3	Air Freshener (50 Gm)	1	107
7.	pro38	Kohinoor Charminar Everyday Rice (Broken)	1	285
8.	pro41	Daawat Rozana Gold Basmati Rice (Medium Grain)	1	374
9.	pro43	Ching's Secret Dark Soy Sauce	1	40
10.	pro49	American GardenBbq Original Sauce	1	213
Total :				2142 /-

© 2018 All rights reserved

Fig 5.6 Bill for the items Shopped

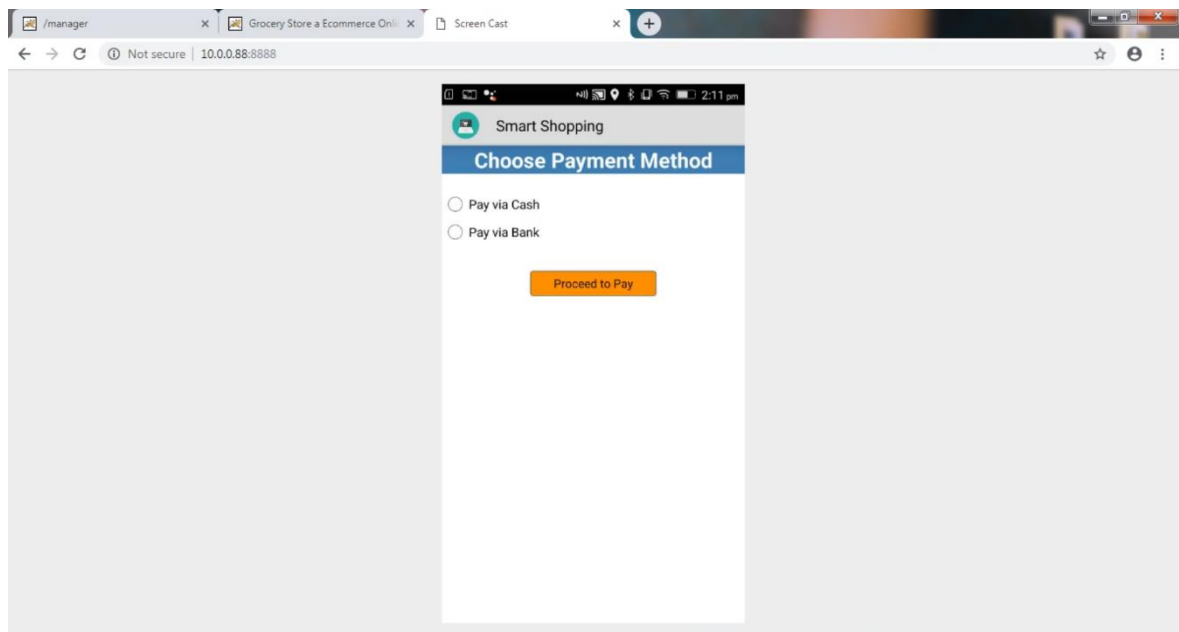


Fig 5.7 Selecting Payment Option

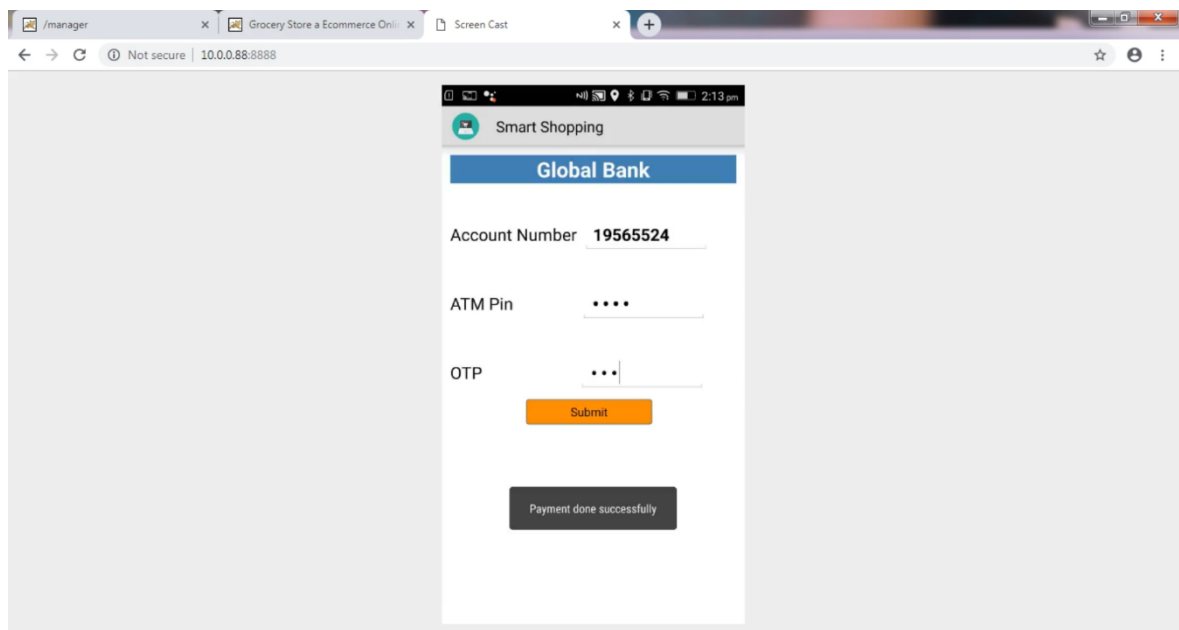


Fig 5.8 Successful Completion of Payment

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

The results obtained from the beginning our model is looking realistic as compared to the existing works. Thus, the model constructed using convolutional neural networks is performing satisfactorily as well as it increases the efficiency of sharpening the blur image and reduced the training time of the model as compared to the developed models like lstm .The proposed model takes sequence of blurry images in different resolutions and generates latent sharp image at each layer and provides the sharp image at full resolution by converting back into the original format. Thus, it reduces the time complexity and gives stable results. Compared with the existing two-frame architectures, the multi frame architecture of can avoid repeated computations caused by multiple inferences when aligning multiple images. We applied logic to image denoising, image superresolution and superresolution tasks.

6.2 FUTUREWORK

In future, developing image deblurring with Bi-Directional LSTM Approach. In bi-directional lstm model the data can transfer from past-lstms to future lstms and future lstms to past-ltms, So that it can develop a images blur image to high resolution image and also convert image in to sentence .In this approach texts in the images also can occur in good resolution without any blur. same way this model can used for old videos also it can deblur the texts in videos and it can help the people to see the old videos in high resolution without blur.

REFERENCES

1. A. Farahzadi, P. Shams, J. Rezazadeh, and R. Farahbakhsh, "Middleware technologies for cloud of things-a survey," *Digital Communications and Networks*, Elsevier, 2017.
2. D. Singh, G. Tripathi, and A. J. Jara, "A survey of internet-of-things: Future vision, architecture, challenges and services," in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, March 2014, pp. 287–292.
3. A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, Fourthquarter 2015.
4. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, Feb 2014.
5. J. Rezazadeh, M. Moradi, A. S. Ismail, and E. Dutkiewicz, "Superior path planning mechanism for mobile beaconassisted localization in wireless sensor networks," *Sensors Journal*, IEEE, vol. 14, pp. 3052–3064, 2014.
6. M. Hubert, M. Blut, C. Brock, C. Backhaus, and T. Eberhardt, "Acceptance of smartphone-based mobile shopping: Mobile benefits, customer characteristics, perceived risks, and the impact of application context," *Psychology and Marketing*, vol. 34, no. 2, pp. 175–194, 2017.
7. J. Rezazadeh, M. Moradi, and A. S. Ismail, "Efficient localization via middle-node cooperation in wireless sensor networks," in *International Conference on Electrical, Control and Computer Engineering*, June 2011, pp. 410–415.
8. M. Moradi, J. Rezazadeh, and A. S. Ismail, "A reverse localization scheme for underwater acoustic sensor networks," *Sensors*, vol. 12, pp. 4352–4380, 2012.

9. R. Nallanthighal and V. Chinta, "Improved grid-scan localization algorithm for wireless sensor networks," *Journal of Engineering*, Hindawi, vol. 5, no. 10, pp. 21–27, 2014.
10. P. Martin, B.-J. Ho, N. Grupen, S. Muñoz, and M. Srivastava, "An ibeacon primer for indoor localization: Demo abstract," in *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, 2014, pp. 190–191.
11. J. Rezazadeh, M. Moradi, A. S. Ismail, and E. Dutkiewicz, "Impact of static trajectories on localization in wireless sensor networks," *Wirel. Netw.*, vol. 21, no. 3, pp. 809–827, 2015.
12. J. Wang, P. Urriza, Y. Han, and D. Cabric, "Weighted centroid localization algorithm: Theoretical analysis and distributed implementation," *IEEE Transactions on Wireless Communications*, vol. 10, no. 10, pp. 3403–3413, 2014.

APPENDIX

A) SOURCE CODE

BankWebServiceDelegate.java

```
package com.wsdl;

import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebResult;
import javax.jws.WebService;
import javax.xml.ws.RequestWrapper;
import javax.xml.ws.ResponseWrapper;

@WebService(name = "BankWebServiceDelegate", targetNamespace =
"http://bankwebser.com/")
public interface BankWebServiceDelegate {

    @WebMethod
    @WebResult(targetNamespace = "")
    @RequestWrapper(localName = "setPayment", targetNamespace =
"http://bankwebser.com/", className = "com.wsdl.SetPayment")
    @ResponseWrapper(localName = "setPaymentResponse",
targetNamespace = "http://bankwebser.com/", className =
"com.wsdl.SetPaymentResponse")
    public String setPayment(
        @WebParam(name = "arg0", targetNamespace = "") String
arg0,
        @WebParam(name = "arg1", targetNamespace = "") String
arg1);

    @WebMethod
    @WebResult(targetNamespace = "")
    @RequestWrapper(localName = "commonMet", targetNamespace =
"http://bankwebser.com/", className = "com.wsdl.CommonMet")
    @ResponseWrapper(localName = "commonMetResponse",
targetNamespace = "http://bankwebser.com/", className =
"com.wsdl.CommonMetResponse")
```

```

        public String commonMet();
    }

```

BankWebServiceService.java

```

package com.wsdl;

import java.net.MalformedURLException;
import java.net.URL;
import java.util.logging.Logger;
import javax.xml.namespace.QName;
import javax.xml.ws.Service;
import javax.xml.ws.WebEndpoint;
import javax.xml.ws.WebServiceClient;

@WebServiceClient(name = "BankWebServiceService", targetNamespace =
"http://bankwebser.com/", wsdlLocation =
"http://localhost:8888/BankWebServer/BankWebServicePort?wsdl")
public class BankWebServiceService extends Service {
    private final static URL BANKWEBSERVICESERVICE_WSDL_LOCATION;
    private final static Logger logger = Logger
.getLogger(com.wsdl.BankWebServiceService.class.getName());
    static {
        URL url = null;
        try {
            URL baseUrl;
            baseUrl = com.wsdl.BankWebServiceService.class.getResource(".");
            url = new URL(baseUrl,
"http://localhost:8888/BankWebServer/BankWebServicePort?wsdl");
        } catch (MalformedURLException e) {
            logger
.warning("Failed to create URL for the wsdl Location:
'http://localhost:8888/BankWebServer/BankWebServicePort?wsdl', retrying as a
local file");

            logger.warning(e.getMessage());
        }
        BANKWEBSERVICESERVICE_WSDL_LOCATION = url;
    }

```

```

    }

    public BankWebServiceService(URL wsdlLocation, QName serviceName) {
        super(wsdlLocation, serviceName);
    }

    public BankWebServiceService() {
        super(BANKWEBSERVICESERVICE_WSDL_LOCATION, new
QName(
            "http://bankwebser.com/", "BankWebServiceService"));
    }
    @WebEndpoint(name = "BankWebServicePort")
    public BankWebServiceDelegate getBankWebServicePort() {
        return super.getPort(new QName("http://bankwebser.com/",
            "BankWebServicePort"),
BankWebServiceDelegate.class);
    }
}

```

CommonMet.java

```

package com.wsdl;

import javax.xml.bind.annotation.XmlAccessType;
import javax.xml.bind.annotation.XmlAccessorType;
import javax.xml.bind.annotation.XmlType;

/**
 * <p>
 * Java class for commonMet complex type.
 *
 * <p>
 * The following schema fragment specifies the expected content contained within
 * this class.
 *
 * <pre>

```

```

* <complexType name="commonMet">
*   <complexContent>
*     <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
*       <sequence>
*         </sequence>
*     </restriction>
*   </complexContent>
* </complexType>
* </pre>
*
*
*/
@XmlAccessorType(XmlAccessType.FIELD)
@XmlType(name = "commonMet")
public class CommonMet {

}

```

CommonMetResponse.java

```

package com.wsdl;

import javax.xml.bind.annotation.XmlAccessType;
import javax.xml.bind.annotation.XmlAccessorType;
import javax.xml.bind.annotation.XmlElement;
import javax.xml.bind.annotation.XmlType;

/**
* <p>
* Java class for commonMetResponse complex type.
*
* <p>
* The following schema fragment specifies the expected content contained within
* this class.
*
* <pre>

```

```

* <complexType name="commonMetResponse">
*   <complexContent>
*     <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
*       <sequence>
*
*                               <element      name="return"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
*       </sequence>
*     </restriction>
*   </complexContent>
* </complexType>
* </pre>
*
*
*/
@XmlAccessorType(XmlAccessType.FIELD)
@XmlType(name = "commonMetResponse", propOrder = { "_return" })
public class CommonMetResponse {

    @XmlElement(name = "return")
    protected String _return;
    public String getReturn() {
        return _return;
    }
    public void setReturn(String value) {
        this._return = value;
    }
}

```

