VOLUNTEER FINDING APPLICATION

PROJECT REPORT

21AD1513- INNOVATION PRACTICES LAB

Submitted by

THIRISHYA M	**211422243338**
SWETHA R	**211422243329**
SANDYA S	**211422243276**

in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123

ANNA UNIVERSITY: CHENNAI-600 025

November, 2024

BONAFIDE CERTIFICATE

Certified that this project report titled "**VOLUNTEER FINDING APPLICATION**" is the bonafide work of "**THIRISHYA.M (211422243338),SWETHA.R(211422243329),SANDYA.S(211422243276)** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

INTERNAL GUIDE
MR.DINESH M.E
Department of AI &DS

HEAD OF THE DEPARTMENT
Dr.S.MALATHI M.E., Ph.D
Professor and Head,
Department of AI & DS.

Certified that the candidate was examined in the Viva-Voce Examination held on ……

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

We hereby declare that the project report entitled **"Volunteer Finding Application"** which is being submitted in partial fulfilment of the requirement of the course leading to the award of the 'Bachelor Of Technology in Artificial Intelligence and Data Science' in Panimalar Engineering College, Autonomous Institution Affiliated to Anna university- Chennai is the result of the project carried out by me under the guidance of Mr.S.Dinesh M.E., Professor in the Department of Information Technology. I further declared that I or any other person has not previously submitted this project report to any other institution/university for any other degree/ diploma or any other person.

Place: Chennai (SWETHA R)

Date: (THIRISHYA M)

(SANDYA S)

It is certified that this project has been prepared and submitted under my guidance.

Date: **Dr.S.MALATHI M.E.,Ph.D**

Place: Chennai Professor/AI&DS

ACKNOWLEDGEMENT

I also take this opportunity to thank all the Faculty and Non-Teaching Staff Members of Department of Computer Science and Engineering for their constant support. Finally I thank each and every one who helped me to complete this project. At the outset we would like to express our gratitude to our beloved respected Chairman, **Dr.Jeppiaar M.A.,Ph.D,** Our beloved correspondent and Secretary **Mr.P.Chinnadurai M.A., M.Phil., Ph.D.,** and our esteemed director for their support.

We would like to express thanks to our Principal, **Dr. K. Mani M.E., Ph.D.,** for having extended his guidance and cooperation.

We would also like to thank our Head of the Department, **Dr.S.Malathi M,E.,Ph.D.,** of Artificial Intelligence and Data Science for her encouragement.

Personally we thank <<**Mr.S.Dinesh M.E.,**>>, Department of Artificial Intelligence and Data Science for the persistent motivation and support for this project, who at all times was the mentor of germination of the project from a small idea.

We express our thanks to the project coordinators **DR. A.Joshi M.E., Ph.D.,** Professor & **Dr.S.Chakaravarthi M.E.,Ph.D.,** Professor in Department of Artificial Intelligence and Data Science for their Valuable suggestions from time to time at every stage of our project.

Finally, we would like to take this opportunity to thank our family members, friends, and well-wishers who have helped us for the successful completion of our project.

We also take the opportunity to thank all faculty and non-teaching staff members in our department for their timely guidance in completing our project.

LIST OF CONTENTS

ABSTRACT

The volunteer-finding application is a digital platform designed to streamline and optimize the process of connecting volunteers with impactful opportunities within their communities. In response to the growing demand for efficient volunteer recruitment and community engagement, this application enables non-profit organizations, community groups, and event organizers to access a pool of committed individuals looking to make a difference. By employing location-based services and skill-matching algorithms, the application allows users to discover local volunteering opportunities that align closely with their skills, interests, and availability, thus fostering a more personalized and meaningful experience.The platform also supports organizations by providing tools for event management, volunteer scheduling, tracking hours, and gathering post-event feedback to assess volunteer satisfaction and improve future engagement. In addition to aiding organizations, volunteers can use the app to build a personal profile, track their contributions, and gain insights into the causes they support, which can enhance their volunteer portfolios and personal development.By facilitating seamless and effective volunteer matching, this application not only helps bridge the gap between volunteers and organizations but also strengthens community ties and empowers individuals to contribute to social, environmental, and cultural initiatives. Ultimately, this tool fosters a culture of service and cultivates resilient, well- connected communities that are better equipped to address local challenges.

KeyWords: Volunteer matching ,Community engagement ,Skill-based matching ,Volunteer recruitment ,Social impact.

LIST OF FIGURES

LIST OF ABBRIEVATIONS

SQL	Structured Query Language
JDBC	Java Database Connectivity
JFreeChart	JAVAFreeChart
ODBC	Open Database Connectivity
J2ME	JAVA 2 MICRO EDITION
JAVA IDE	Java Integrated Development Environment

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

The Volunteer Finding Application is an innovative digital platform aimed at connecting individuals who are eager to volunteer with meaningful opportunities in their local communities. By leveraging **location-based** and **skill-based matching**, the app enables users to find volunteering opportunities that align with their interests and expertise, making the process more personalized and impactful. Volunteers can easily browse, apply, and keep track of their engagements, enhancing their overall experience. Organizations, including non-profits and community groups, benefit from streamlined **recruitment** and **event management** tools within the application. These features allow them to effectively recruit volunteers, manage events, and track volunteer hours, fostering better engagement and communication. By bridging the gap between volunteers and organizations, the application promotes community involvement and social impact, ultimately helping to create stronger, more connected communities.

1.2 AIM OF THE PROJECT

The primary aim of the Volunteer Finding Application is to create an efficient platform that connects individuals who are eager to volunteer with organizations in need of their support. By facilitating easy access to a diverse range of volunteering opportunities, the application enhances community engagement and promotes a culture of service. This allows volunteers to discover roles that match their interests and skills while enabling organizations to tap into a broader pool of dedicated individuals ready to make a difference. Furthermore, the application empowers users by providing personalized matching based on their unique profiles and offers organizations essential tools for effective volunteer recruitment and management. By encouraging active participation in local initiatives, the project aims to strengthen community ties and foster social responsibility. Ultimately, the Volunteer Finding Application seeks to drive positive social impact by increasing volunteer involvement, enriching both the lives of volunteers and the communities they serve. Through this initiative, we envision a vibrant community where volunteerism is celebrated and where individuals can make a meaningful difference. Together, we aim to build a sustainable network that supports ongoing civic engagement and social change.

1.3 OBJECTIVE OF THE PROJECT

The Volunteer Finding Application aims to achieve several key objectives to maximize its impact on the community. First, it seeks to facilitate volunteer engagement by providing a user-friendly platform that allows individuals to easily discover and apply for volunteering opportunities that match their skills and interests. Second, it aims to enhance organizational efficiency by offering tools for effective recruitment, event management, and volunteer tracking, enabling organizations to engage with volunteers more effectively. Additionally, the application strives to promote community connection by encouraging active participation in local initiatives and fostering collaboration between volunteers and organizations. It also aims to support personal growth for volunteers by allowing them to build profiles, track their contributions, and gain insights into their impact. Ultimately, the project seeks to drive social impact by increasing volunteer participation in various social, environmental, and cultural initiatives, contributing to the overall betterment of the community.

1.4 SCOPE OF THE PROJECT

The scope of the Volunteer Finding Application encompasses a comprehensive suite of features designed to connect volunteers with organizations in need of their support. The application will allow users to create detailed profiles, specifying their skills, interests, and availability, which will enhance the matching process. Organizations will also be able to register and create profiles, where they can post various volunteering opportunities, detailing the requirements and expectations for each role. This initial setup is crucial for ensuring that both volunteers and organizations can effectively communicate their needs and capabilities. The application will include a robust

matching algorithm that pairs volunteers with suitable opportunities based on their profiles and preferences. Volunteers will receive personalized notifications about relevant openings, while organizations can manage their recruitment processes through a user- friendly dashboard. Features such as messaging systems will facilitate direct communication between volunteers and organizations, allowing for efficient coordination of volunteer activities. Additionally, the application will provide event management tools, including scheduling, tracking volunteer hours, and collecting feedback, to help organizations assess their programs and improve future engagements. Moreover, the project will focus on promoting community engagement by highlighting local initiatives and events within the application. Users will be encouraged to participate in various social, environmental, and cultural activities, fostering a sense of belonging and civic responsibility. The application will also incorporate analytics tools to enable organizations to evaluate the effectiveness of their volunteer programs and measure their social impact. Ultimately, the scope of the Volunteer Finding Application aims to create a sustainable and scalable platform that enhances volunteerism and strengthens community bonds, making volunteering an integral part of local life.

CHAPTER 2

LITERATURE REVIEW

2.1 FoodScan: Food monitoring App by Scanning the Groceries Receipts.

This paper presents FoodScan, an Android application designed for individuals aged 70 and older, particularly in rural areas or those with limited technical skills, to manage their food intake using grocery receipts. Unlike traditional diet tracking apps that require manual entry of each food item, FoodScan simplifies this process by generating dietary recommendations directly from scanned receipts. The app's development involved reviewing existing calorie control applications and testing various algorithms to streamline food entry. A pilot study with 109 volunteers from rural Spain and Portugal found that 93% of users found the app easy to download, while 66% considered it user-friendly. Additionally, many participants reported that the recommendation charts assisted with diet management and improving healthy eating habits, demonstrating FoodScan's potential as an effective tool for older adults to monitor their food intake.

AUTHOR: Beatrix sainz-De-Adajo,Jose Manuelgarcia-Alonso,José Javier Berrocal-Olmeda; Sergio Laso-Mangas; Isabel De La Torre-Díez.

YEAR: 21 December 2020

2.2 Building an application framework to connect NGOs and Volunteers.

The NGO sector struggles to get volunteers and yet individuals interested in volunteering some of their time to socially meaningful causes are not able to find suitable opportunities. This paper aims to set up platforms to allow NGOs to communicate these volunteering opportunities and for volunteers to enroll for those that match their preferences. The platform is developed both as an Android application and a Web-based interface. The paper also explores how gamification can be used to make the process fun and competitive to increase engagement with the volunteers.

AUTHOR:**Mridule Goel,Aryan Agarwal,Namit Chandwani,Tanmay Dixit**

YEAR: 05 April 2021

2.3 Volunteer Assisted Collaborative Offloading and Resource Allocation in Vehicular Edge Computing.

This paper investigates the use of Vehicular Edge Computing (VEC) to enhance the quality of service (QoS) for computation-intensive vehicular applications by leveraging idle resources in volunteer vehicles to support overloaded VEC servers. We introduce a model for volunteer-assisted VEC, where cost and utility functions are defined for both requesting vehicles and VEC servers, creating incentives for volunteer vehicles to help by offering rewards. Through the application of Stackelberg game theory, we analyze the interactions between requesting vehicles and VEC servers to identify optimal strategies and demonstrate the existence of a unique Stackelberg equilibrium. Furthermore, we develop a genetic algorithm to efficiently find the best pricing strategy for VEC servers and propose a volunteer task assignment algorithm to effectively allocate tasks among volunteer vehicles. Our extensive simulations show that the proposed approach significantly reduces offloading costs for vehicles and enhances the utility of VEC servers compared to other methods.

AUTHOR: Feng Zeng,Qiao Chen, Lin Meng, Jinsong Wu.

YEAR: 18 March 2020

2.4 Assessment of Blood Donor Information Using Kernel Density Maps:

This study introduces a web application aimed at streamlining the process of acquiring blood donor information, which is crucial for ensuring that critical patients receive the right blood type without delay. We utilize data from CRH Hospital in Gangtok, Sikkim, analyzing 391 blood donor records through ArcGIS to create a regional heatmap. The data includes 165 A+ Ve, 94 B+ Ve, 43 AB+ Ve, 3 B- Ve, and 86 O+ Ve donors. The application allows users to easily find nearby blood donors and contact them directly via phone calls. Additionally, users can register their personal details, including blood type, which will be displayed on the heatmap for future reference. This tool also helps local blood banks identify regions with a high demand for specific blood types, facilitating the organization of donation camps and promoting volunteer participation.

AUTHOR:Madhab Nirola; Mousumi Gupta; Arpan Sharma.

YEAR: 09 February 2023

2.5 A Design of Volunteer Computing System Based on Blockchain.

This paper presents a volunteer computing system designed using blockchain technology, which enhances the application range of blockchain by addressing issues related to scalability and single points of failure in traditional centralized client/server frameworks. By leveraging the decentralized, persistent, and auditable features of blockchain, the proposed system allows for greater scalability, accommodating more computing jobs and participants while ensuring longer-term and more stable computing resources. Additionally, the system offers traceability, enabling anyone to access the blockchain to see which volunteers contributed to specific computing tasks and the outcomes of their contributions. The study explores the integration of the client/server framework with blockchain technology, discussing the system's advantages and disadvantages in terms of feasibility, scalability, security, authenticity, and traceability.

AUTHOR: Boxuan Shan

YEAR: 29 March 2021

2.6 Identification and Analysis of Incentive System Parameters for Large Social Groups of Volunteers.

This paper investigates the challenge of identifying parameters for an implicit incentive system designed to enhance volunteer engagement in large social groups. It introduces a methodology for determining the parameters of the incentive function, based on the assumption that this function decreases monotonically in relation to the actions of agents, who are presumed to act rationally.The study leverages statistical data on the structural dynamics of volunteers in the Russian Federation from 2016 to 2020 to analyze the coefficients of the incentive function and estimate the total incentive fund. This analysis offers valuable insights into structuring effective incentives to boost volunteer participation and engagement within these social groups.

AUTHOR:M.I. Geraskin.

YEAR: 09 July 2024

2.7 A Collaborative Citizen Science Platform for Real-Time Volunteer Computing and Games.

This paper explores the integration of three tools: CERN's Virtual Atom Smasher (VAS) game, LiveQ job distribution middleware, and CitizenGrid, a platform for citizen cyberscience (CCS) projects. By combining volunteer computing (VC), which allows individuals to contribute computing resources, with volunteer thinking (VT), which encourages interactive participation in solving human computation tasks, the integration aims to enhance the scientific and educational objectives of CCS initiatives. It provides an overview of each tool and details the integration process, demonstrating how this unified platform can engage volunteers in scientific endeavors. It also discusses potential applications for the resulting system, emphasizing its ability to promote collaboration and improve the efficiency of citizen-driven scientific research.

AUTHOR:Poonam Yadav; Ioannis Charalampidis; Jeremy Cohen; John Darlington; Francois Grey.

YEAR: 08 January 2018

2.8 V - Vibe: Efficient Volunteer Management using Mobile Technology.

Volunteers often encounter challenges such as unclear instructions, limited support, and communication gaps, but their commitment to making a positive impact drives them. The proposed Volunteer Management System (VMS) aims to address these issues by serving as a digital platform that connects volunteers and organizations. Volunteers can easily register, create profiles, and highlight their skills, while organizations can promote upcoming events through a feed visible to volunteers. The VMS enhances ongoing support and communication, enabling volunteers to discover and sign up for events that align with their interests and availability. In doing so, this system transforms good intentions into meaningful actions, fostering a more connected and effective volunteer community.

AUTHOR: Prathamesh Pandharinath Wadekar, Shreyas Kumar Sachan, Sujal Kishor Patil, Harshal Mahesh Wagh, Manimala Mahato.

YEAR: 20 August 2024

CHAPTER 3

SYSTEM DESIGN

3.1 EXISTING SYSTEM

An existing volunteer finding application typically includes a variety of key features designed to facilitate connections between volunteers and organizations. Users can create accounts and set up detailed profiles.
Volunteers have the opportunity to list their skills, interests, and availability, while organizations can provide information about their missions, goals, and specific needs. This dual-profile approach helps both parties understand how they can best collaborate and engage. One of the core functionalities of these applications is event listings, where organizations can post upcoming volunteer opportunities. These listings include crucial details such as the date, location, duration, and specific tasks involved. Events are often categorized by type—such as environmental, educational, or healthcare initiatives—making it easier for volunteers to navigate and find opportunities that align with their interests and expertise. Additionally, robust search and filter functions allow volunteers to narrow down opportunities based on their preferences, enhancing the user experience. Effective communication tools are also a fundamental component of these applications. Many platforms include messaging systems or forums that enable direct interaction between volunteers and organizations. This feature helps clarify roles, address questions, and provide important updates about events. Furthermore, rating and feedback systems allow volunteers to share their experiences after completing an opportunity, promoting accountability and continuous improvement within the volunteering community. Beyond these features, many applications integrate additional tools to enhance usability. Calendar integration allows volunteers to manage their schedules and keep track of upcoming commitments, while organizations can access reporting and analytics to understand volunteer engagement trends.

Training resources may also be provided to prepare volunteers for their roles effectively. Additionally, mobile compatibility ensures that volunteers can find and register for opportunities on the go. Although existing systems offer these essential functionalities, there remains room for improvement in areas such as user experience, communication efficiency, and the personalization of opportunities based on individual volunteer profiles. Overall, enhancing these

systems could lead to greater volunteer retention and satisfaction, ultimately fostering a more engaged and effective volunteer community. By focusing on innovative features and user-centered design, these applications can better meet the evolving needs of volunteers and organizations alike.

3.1.1 DRAW BACKS

- **Technical Barriers**: Some potential volunteers may lack access to technology or the internet, limiting their ability to register and participate in opportunities. Additionally, those who are not tech-savvy may struggle to navigate the application effectively.

- **User Engagement**: Maintaining user engagement can be challenging, as volunteers might lose interest over time, especially if they do not receive timely updates or relevant opportunities that match their skills and interests.

- **Quality Control**: Ensuring the quality and reliability of organizations posting opportunities can be difficult. Without proper vetting processes, volunteers may encounter untrustworthy organizations or poorly organized events, leading to a negative experience.

- **Communication Gaps**: While the application aims to bridge communication between volunteers and organizations, misunderstandings or miscommunications may still occur. If the messaging system is not user-friendly or if response times are slow, this can lead to frustration for both parties.

- **Incentive Structure**: The lack of a robust incentive structure may hinder volunteer participation. Without proper recognition or rewards, volunteers might feel undervalued, which can decrease motivation to engage regularly.

- **Limited Customization**: If the application does not offer enough

customization options for volunteer profiles or preferences, users may find it challenging to discover opportunities that truly resonate with their interests and skills.

3.2 PROPOSED SYSTEM

The Prpoposed Volunteer Management System helps the volunteer Admin to easily monitor the volunteer request and users database.The propose system takes a system approach of how to bridege the gap between admin,volunteer, and needer.It improves the existing system by providing a common ground to ease the process of volunteers and needer.

Advantages

- The proposed volunteer management system helps the people who are in need of a volunteer by giving them all details of volunteer group availability or regarding the volunteers with same volunteer group.Our application work 24x7 users can get information of volunteer at anytime

3.2.1 SYSTEM ARCHITECTURE

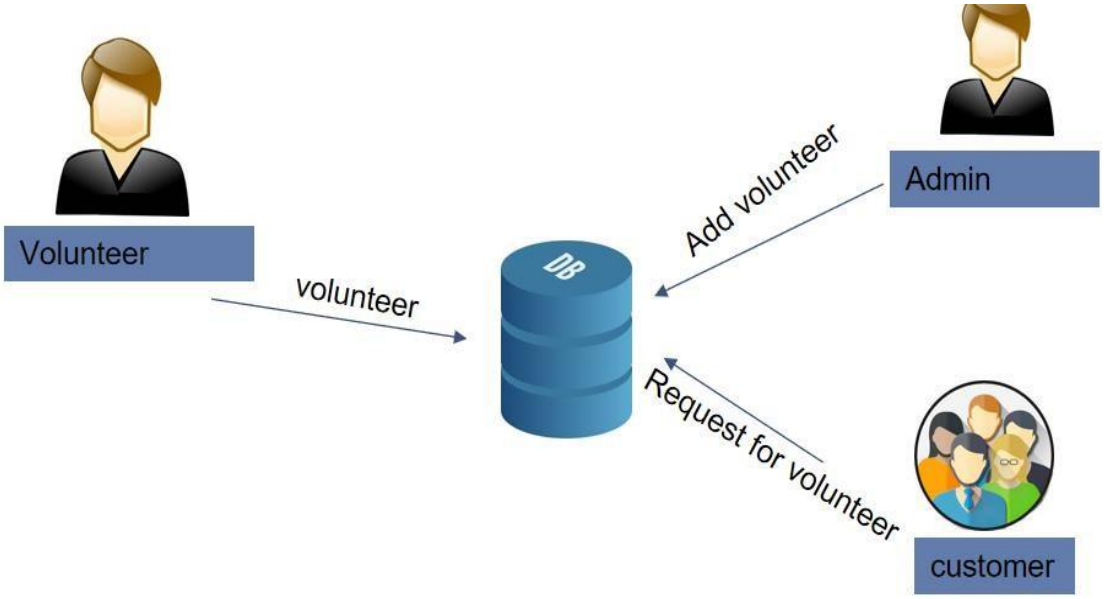


Fig 3.1

3.2.2 MERITS

- **Improved Accessibility**: The application provides a centralized platform for volunteers to discover and engage with various opportunities, making it easier for individuals to find ways to contribute to their communities.

- **Personalized Matching**: Advanced algorithms and search functionalities allow volunteers to find opportunities that align with their interests, skills, and availability, leading to more meaningful and fulfilling volunteering experiences.

- **Enhanced Communication**: The integrated messaging system facilitates real-time communication between volunteers and organizations, ensuring that questions are answered promptly and that updates are communicated effectively.

- **Community Building**: By connecting like-minded individuals and organizations, the application helps foster a sense of community and collaboration, encouraging volunteers to engage with others who share their passions.

- **Positive Social Impact**: Ultimately, the application aims to enhance the effectiveness of volunteer efforts, contributing to positive social change and addressing various community needs more efficiently.

3.3 SYSTEM DESIGN

3.3.1 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system.

The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

3.3.2 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
2. Select methods for presenting information.
3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- ❖ Convey information about past activities, current status or projections of the
- ❖ Future.
- ❖ Signal important events, opportunities, problems, or warnings.
- ❖ Trigger an action.

Confirm an action.

CHAPTER 4

MODULES

4.0 MODULES

This Project has following Modules:

1. ADMIN
2. VOLUNTEERS
3. CUSTOMERS

4.1 ADMIN

1. Login
2. Add all details of volunteer group
3. View Volunteer
4. Logout

4.2 VOLUNTEER & CUSTOMERS

1. Register
2. Login
3. Volunteer register
4. Search
5. Request Volunteer
6. Logout

4.3 ER DIAGRAM

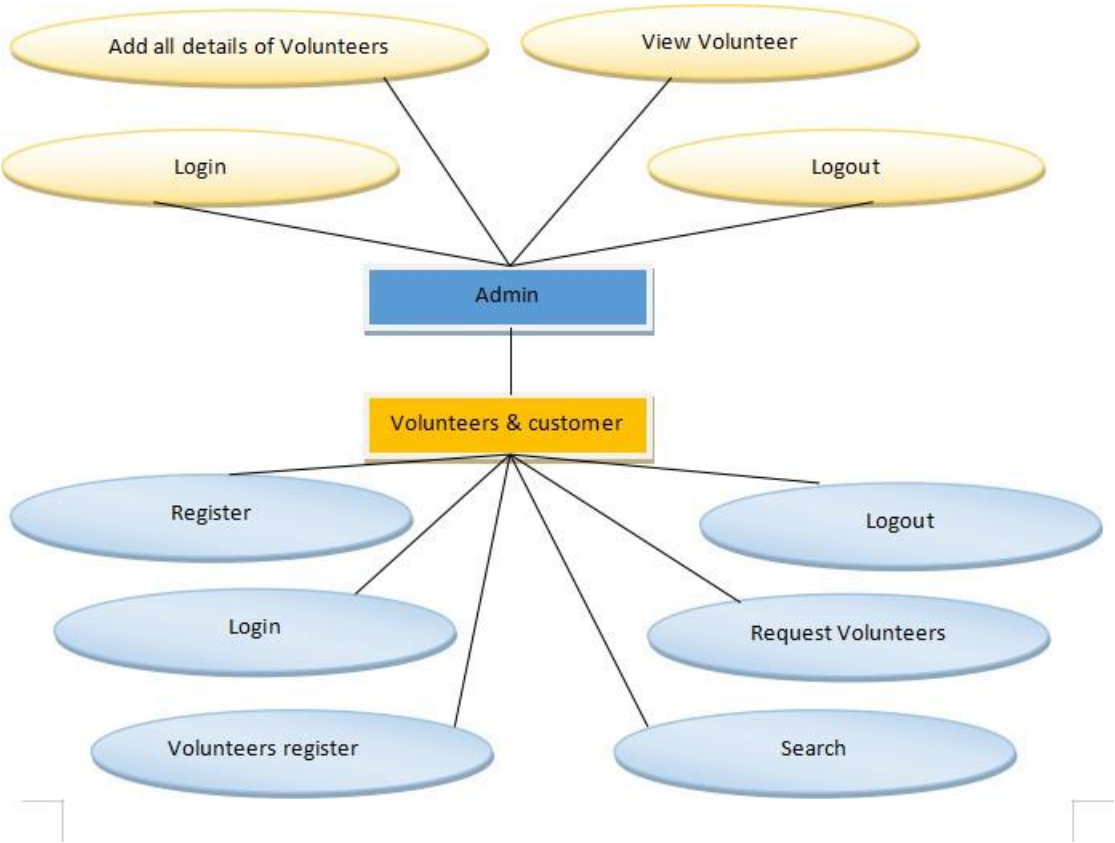


Fig:4.1

CHAPTER 5

SYSTEM REQUIREMENTS

5.1 INTRODUCTION

Java technology is both a powerful programming language and a versatile platform widely used in software development. Recognized for its simplicity and robustness, Java supports a wide array of applications—from web-based applets to complex server-side applications. This document provides a comprehensive overview of Java's architecture, its core features, and the tools available within its ecosystem, including networking capabilities, database connectivity via JDBC, and data visualization through libraries like JFreeChart.

5.2 REQUIREMENTS

5.2.1 HARDWARE REQUIREMENTS:

- System : Intel i3.
- Hard Disk : 160 GB.
- Monitor : 15 VGA Colour.
- Mouse : Logitech.
- Ram : 4 GB.

SOFTWARE REQUIREMENTS:

- Operating system : Windows XP/7/8/10
- Coding Language : JAVA
- Front End : NetBeans Software
- Backend : My SQLyog Community 32

5.3 TECHNOLOGY USED

5.3.1 Java Technology

Java technology is both a programming language and a platform.

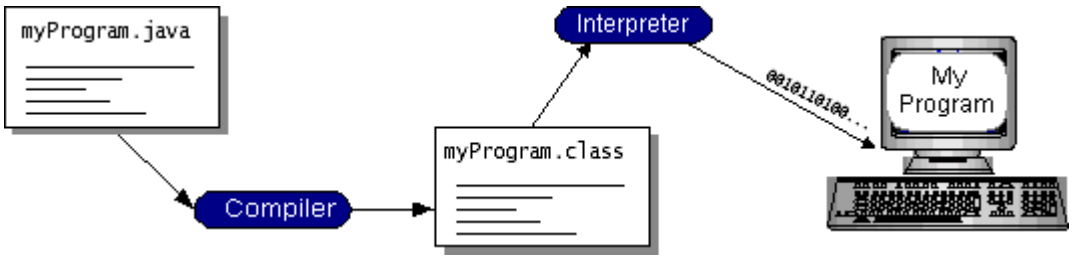
The Java Programming Language

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

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

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called *Java byte codes* —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following

figure illustrates how this works.



You can think of Java byte codes as the machine code instructions for the *Java Virtual Machine* (Java VM). Every Java interpreter, whether it's a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make "write once, run anywhere" possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.

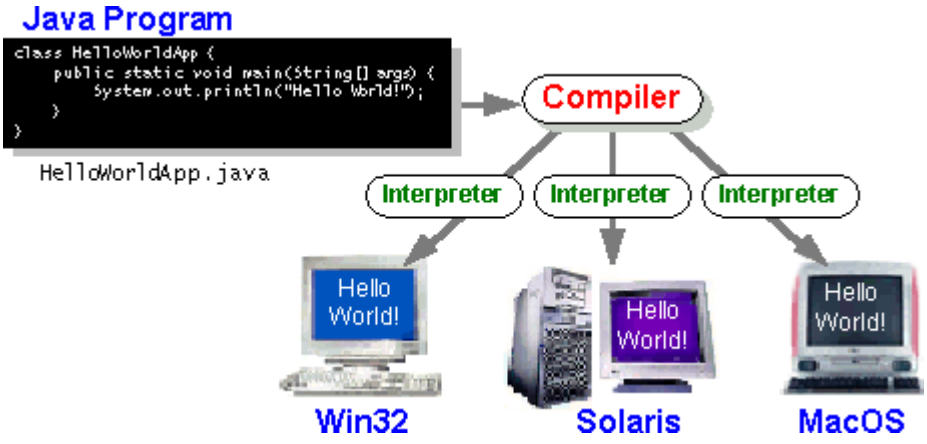


Fig:5.1

5.3.2 The Java Platform

A *platform* is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Windows 2000, Linux,

Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware.

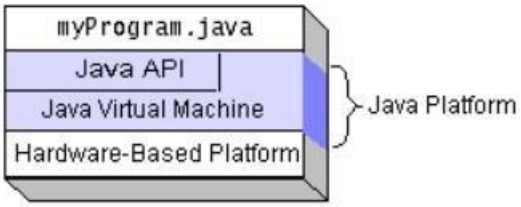
The Java platform has two components:

- The *Java Virtual Machine* (Java VM)
- The *Java Application Programming Interface* (Java API)

You've already been introduced to the Java VM. 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 of related classes and interfaces; these libraries are known as *packages*. The next section, What Can Java Technology Do? Highlights what functionality some of the packages in the Java API provide.

The following figure depicts a program that's running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.



Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time byte code compilers can bring performance close to that of native code without threatening portability.

5.3.3 What Can Java Technology Do?

The most common types of programs written in the Java programming language are *applets* and *applications*. If you've surfed the Web, you're probably already familiar

with applets. An applet is a program that adheres to certain conventions that allow it to run within a Java-enabled browser.

However, the Java programming language is not just for writing cute, entertaining applets for the Web

An application is a standalone program that runs directly on the Java platform. A special kind of application known as a *server* serves and supports clients on a network. Examples of servers are Web servers, proxy servers, mail servers, and print servers. Another specialized program is a *servlet*. A servlet can almost be thought of as an applet that runs on the server side. Java Servlets are a popular choice for building interactive web applications, replacing the use of CGI scripts. Servlets are similar to applets in that they are runtime extensions of applications. Instead of working in browsers, though, servlets run within Java Web servers, configuring or tailoring the server.

How does the API support all these kinds of programs? It does so with packages of software components that provides a wide range of functionality. Every full implementation of the Java platform gives you the following features:

- **The essentials**: Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.
- **Applets**: The set of conventions used by applets.
- **Networking**: URLs, TCP (Transmission Control Protocol), UDP (User Data gram Protocol) sockets, and IP (Internet Protocol) addresses.
- **Internationalization**: Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.
- **Security**: Both low level and high level, including electronic signatures, public and private key management, access control, and certificates.
- **Software components**: Known as JavaBeans™, can plug into existing component architectures.

- **Object serialization**: Allows lightweight persistence and communication via Remote Method Invocation (RMI).
- **Java Database Connectivity (JDBC™)**: Provides uniform access to a wide range of relational databases.

The Java platform also has APIs for 2D and 3D graphics, accessibility, servers, collaboration, telephony, speech, animation, and more. The following figure depicts what is included in the Java 2 SDK.

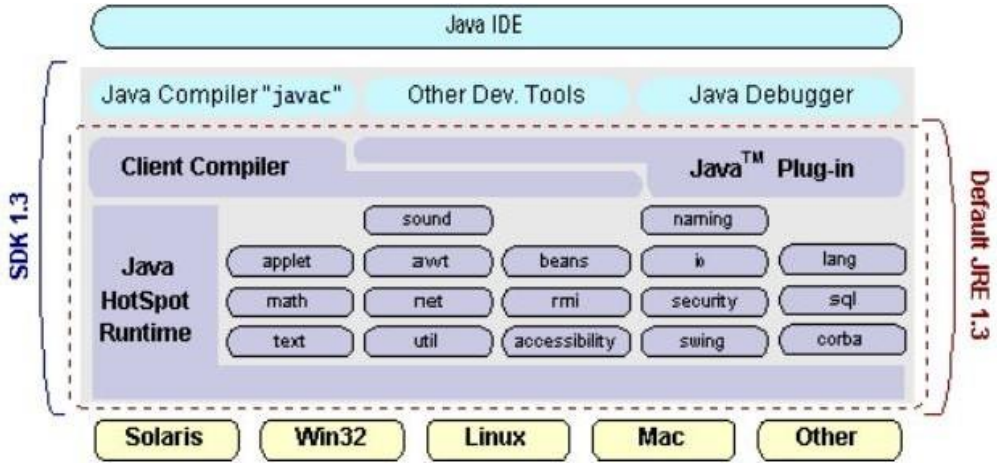


Fig:5.2

5.3.4 ODBC

Microsoft Open Database Connectivity (ODBC) is a standard programming interface for application developers and database systems providers. Before ODBC became a *de facto* standard for Windows programs to interface with database systems, programmers had to use proprietary languages for each database they wanted to connect to. Now, ODBC has made the choice of the database system almost irrelevant from a coding perspective, which is as it should be. Application developers have much more important things to worry about than the syntax that is needed to port their program from one database to another when business needs suddenly change.

Through the ODBC Administrator in Control Panel, you can specify the particular database that is associated with a data source that an ODBC application program is written to use. Think of an ODBC data source as a door with a name on it. Each door

will lead you to a particular database. For example, the data source named Sales Figures might be a SQL Server database, whereas the Accounts Payable data source could refer to an Access database. The physical database referred to by a data source can reside anywhere on the LAN.

The ODBC system files are not installed on your system by Windows 95. Rather, they are installed when you setup separate database application, such as SQL Server Client or Visual Basic 4.0. When the ODBC icon is installed in Control Panel, it uses a file called ODBCINST.DLL. It is also possible to administer your ODBC data sources through a stand-alone program called ODBCADM.EXE

From a programming perspective, the beauty of ODBC is that the application can be written to use the same set of function calls to interface with any data source, regardless of the database vendor. The source code of the application doesn't change whether it talks to Oracle or SQL Server. We only mention these two as an example. There are ODBC drivers available for several dozen popular database systems. Even Excel spreadsheets and plain text files can be turned into data sources. The operating system uses the Registry information written by ODBC Administrator to determine which low-level ODBC drivers are needed to talk to the data source (such as the interface to Oracle or SQL Server). The loading of the ODBCdrivers is transparent to the ODBC application program. In a client/server environment, the ODBC API even handles many of the network the application.

The advantages of this scheme are so numerous that you are probably thinking there must be some catch. The only disadvantage of ODBC is that it isn't as efficient as talking directly to the native database interface. ODBC has had many detractors make the charge that it is too slow. Microsoft has always claimed that the critical factor in performance is the quality of the driver software that is used. In our humble opinion, this is true. The availability of good ODBC drivers has improved a great deal recently. And anyway, the criticism about performance is somewhat analogous to those who said that compilers would never match the speed of pure assembly language. Maybe not, but the compiler (or ODBC) gives you the opportunity to write cleaner programs, which means you finish sooner. Meanwhile, computers get faster every year.

5.3.5 JDBC

In an effort to set an independent database standard API for Java; Sun Microsystems developed Java Database Connectivity, or JDBC. JDBC offers a generic SQL database access mechanism that provides a consistent interface to a variety of RDBMSs. This consistent interface is achieved through the use of "plug-in" database connectivity

modules, or *drivers*. If a database vendor wishes to have JDBC support, he or she must provide the driver for each platform that the database and Java run on.

To gain a wider acceptance of JDBC, Sun based JDBC's framework on ODBC. As you discovered earlier in this chapter, ODBC has widespread support on a variety of platforms. Basing JDBC on ODBC will allow vendors to bring JDBC drivers to market much faster than developing a completely new connectivity solution was released soon after.

The remainder of this section will cover enough information about JDBC for you to know what it is about and how to use it effectively. This is by no means a complete overview of JDBC. That would fill an entire book.

JDBC Goals

Few software packages are designed without goals in mind. JDBC is one that, because of its many goals, drove the development of the API. These goals, in conjunction with early reviewer feedback, have finalized the JDBC class library into a solid framework for building database applications in Java. The goals that were set for JDBC are important. They will give you some insight as to why certain classes and functionalities behave the way they do. The eight design goals for JDBC are as follows:

1. **SQL Level API**

The designers felt that their main goal was to define a SQL interface for Java. Although not the lowest database interface level possible, it is at a low enough level for higher-level tools and APIs to be created. Conversely, it is at a high enough level for application programmers to use it confidently. Attaining this goal allows for future tool vendors to "generate" JDBC code and to hide many of JDBC's complexities from the end user.

2. S***QL*** **Conformance**

SQL syntax varies as you move from database vendor to database vendor. In an effort to support a wide variety of vendors, JDBC will allow any query statement to be passed through it to the underlying database driver. This allows the connectivity module to

handle non-standard functionality in a manner that is suitable for its users.

1. **JDBC must be implemental on top of common database interfaces** The JDBC SQL API must "sit" on top of other common SQL level APIs. This goal allows JDBC to use existing ODBC level drivers by the use of a software interface. This interface would translate JDBC calls to ODBC and vice versa.

2. **Keep it simple**

This goal probably appears in all software design goal listings. JDBC is no exception. Sun felt that the design of JDBC should be very simple, allowing for only one method of completing a task per mechanism. Allowing duplicate functionality only serves to confuse the users of the API.

3. **Use strong, static typing wherever possible**

Strong typing allows for more error checking to be done at compile time; also, less error appear at runtime.

4. **Keep the common cases simple**

Because more often than not, the usual SQL calls used by the programmer are simple SELECT's, INSERT's, DELETE's and UPDATE's, these queries should be simple to perform with JDBC. However, more complex SQL statements should also be possible.

Finally we decided to precede the implementation using Java Networking.

And for dynamically updating the cache table we go for MS Access database.

Java ha two things: a programming language and a platform.

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

- Simple
- Architecture-neutral
- Object-oriented
- Portable

- Distributed
- Interpreted
- Robust
- Secure

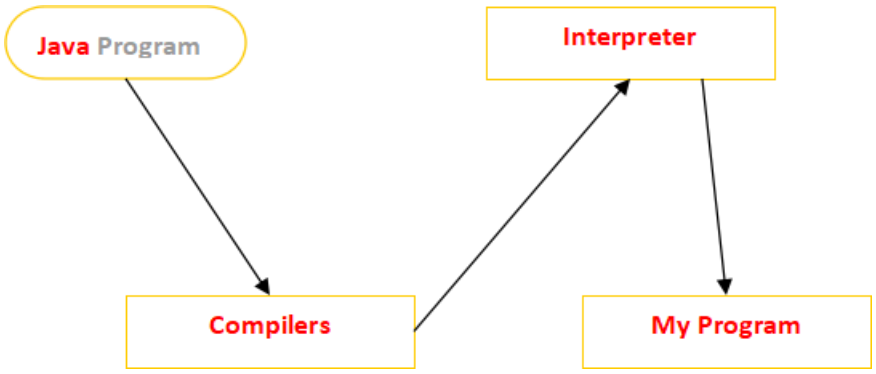
High-performance

multithreaded

Dynamic

Java is also unusual in that each Java program is both compiled and interpreted. With a compile you translate a Java program into an intermediate language called Java byte codes the platform-independent code instruction is passed and run on the computer.

Compilation happens just once; interpretation occurs each time the program is executed. The figure illustrates how this works.



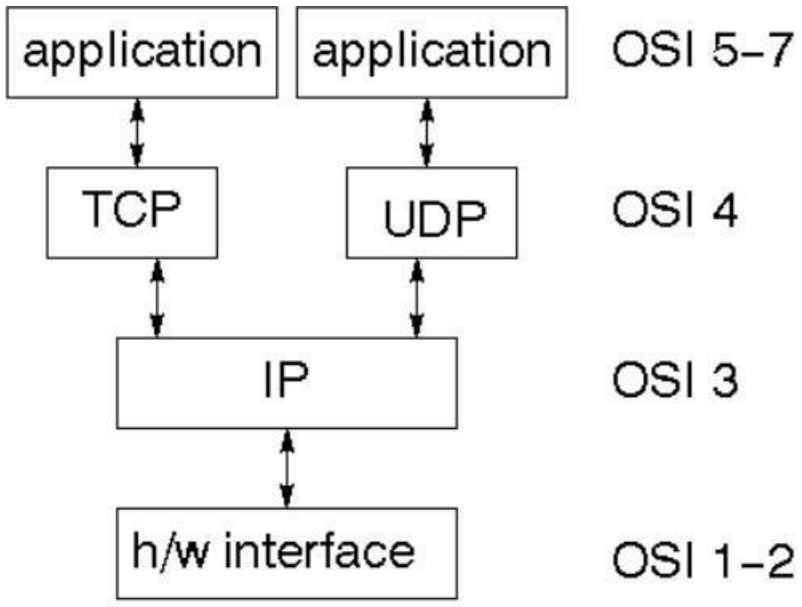
You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it's a Java development tool or a Web browser that can run Java applets, is an implementation of the Java VM. The Java VM can also be implemented in hardware.

Java byte codes help make "write once, run anywhere" possible. You can compile your Java program into byte codes on my platform that has a Java compiler. The byte codes can then be run any implementation of the Java VM. For example, the same Java program can run Windows NT, Solaris, and Macintosh.

5.3.6 NETWORKING

TCP/IP stack

The TCP/IP stack is shorter than the OSI one:



TCP is a connection-oriented protocol; UDP (User Datagram Protocol) is a connectionless protocol.

IP datagram's

The IP layer provides a connectionless and unreliable delivery system. It considers each datagram independently of the others. Any association between datagram must be supplied by the higher layers. The IP layer supplies a checksum that includes its own header. The header includes the source and destination addresses. The IP layer handles routing through an Internet. It is also responsible for breaking up large datagram into smaller ones for transmission and reassembling them at the other end.

UDP

UDP is also connectionless and unreliable. What it adds to IP is a checksum for the contents of the datagram and port numbers. These are used to give a client/server model - see later.

TCP

TCP supplies logic to give a reliable connection-oriented protocol above IP. It provides a virtual circuit that two processes can use to communicate.

Internet addresses

In order to use a service, you must be able to find it. The Internet uses an address scheme for machines so that they can be located. The address is a 32 bit integer which gives the IP address. This encodes a network ID and more addressing. The network ID falls into various classes according to the size of the network address.

Network address

Class A uses 8 bits for the network address with 24 bits left over for other addressing. Class B uses 16 bit network addressing. Class C uses 24 bit network addressing and class D uses all 32.

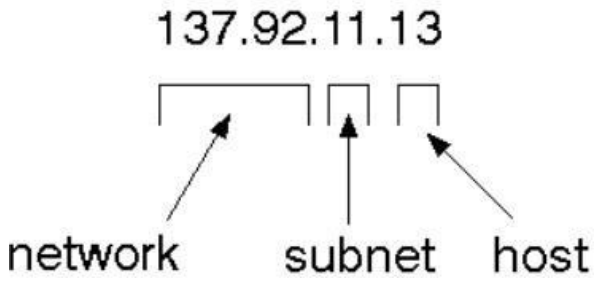
Subnet address

Internally, the UNIX network is divided into sub networks. Building 11 is currently on one sub network and uses 10-bit addressing, allowing 1024 different hosts.

Host address

8 bits are finally used for host addresses within our subnet. This places a limit of 256 machines that can be on the subnet.

Total address



The 32 bit address is usually written as 4 integers separated by dots.

Port addresses

A service exists on a host, and is identified by its port. This is a 16 bit number. To send a message to a server, you send it to the port for that service of the host that it is running on. This is not location transparency! Certain of these ports are "well known".

Sockets

A socket is a data structure maintained by the system to handle network connections. A socket is created using the call socket. It returns an integer that is like a file descriptor. In fact, under Windows, this handle can be used with Read File and Write File functions.

Here "family" will be AF_INET for IP communications, protocol will be zero, and type will depend on whether TCP or UDP is used. Two processes wishing to communicate over a network create a socket each. These are similar to two ends of a pipe - but the actual pipe does not yet exist.

JFree Chart

JFreeChart is a free 100% Java chart library that makes it easy for developers to display professional quality charts in their applications. JFreeChart's extensive feature set includes:

- A consistent and well-documented API, supporting a wide range of chart types;
- A flexible design that is easy to extend, and targets both server-side and client-side applications;
- Support for many output types, including Swing components, image files (including PNG and JPEG), and vector graphics file formats (including PDF, EPS and SVG);

JFreeChart is "open source" or, more specifically, free software. It is distributed under the terms of the GNU Lesser General Public Licence (LGPL), which permits use in proprietary applications.

1. Map Visualizations

Charts showing values that relate to geographical areas. Some examples include: (a) population density in each state of the United States, (b) income per capita for each country in Europe, (c) life expectancy in each country of the world. The tasks in this project include:

Sourcing freely redistributable vector outlines for the countries of the world, states/provinces in particular countries (USA in particular, but also other areas);

Creating an appropriate dataset interface (plus default implementation), a rendered, and integrating this with the existing XYPlot class in JFreeChart;

Testing, documenting, testing some more, documenting some more.

2. Time Series Chart Interactivity

Implement a new (to JFreeChart) feature for interactive time series charts --- to display a separate control that shows a small version of ALL the time series data, with a sliding "view" rectangle that allows you to select the subset of the time series data to display in the main chart.

3. Dashboards

There is currently a lot of interest in dashboard displays. Create a flexible dashboard mechanism that supports a subset of JFreeChart chart types (dials, pies, thermometers, bars, and lines/time series) that can be delivered easily via both Java Web Start and an applet.

4. Property Editors

The property editor mechanism in JFreeChart only handles a small subset of the properties that can be set for charts. Extend (or reimplement) this mechanism to provide greater end-user control over the appearance of the charts.

J2ME (Java 2 Micro edition):-

Sun Microsystems defines J2ME as "a highly optimized Java run-time environment

targeting a wide range of consumer products, including pagers, cellular phones, screen-phones, digital set-top boxes and car navigation systems."

Announced in June 1999 at the JavaOne Developer Conference, J2ME brings the cross-platform functionality of the Java language to smaller devices, allowing mobile wireless devices to share applications. With J2ME, Sun has adapted the Java platform for consumer products that incorporate or are based on small computing devices.

1. General J2ME architecture

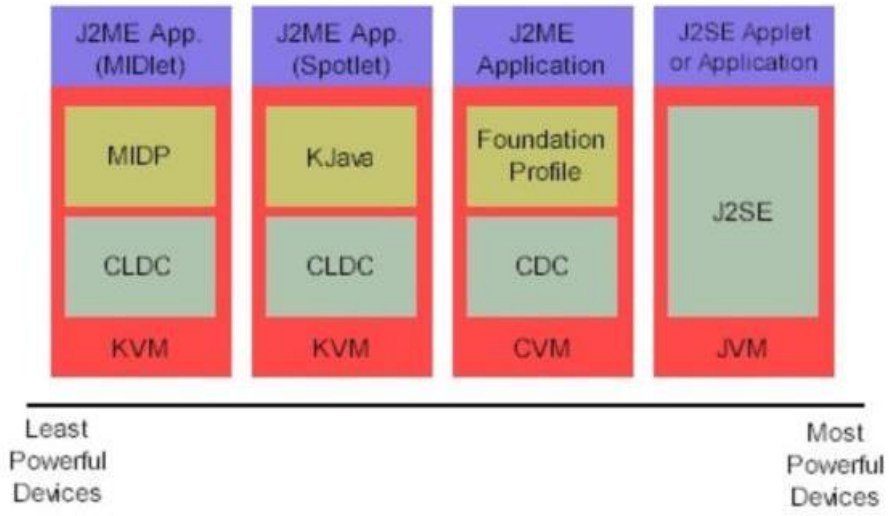


Fig:5.3

J2ME uses configurations and profiles to customize the Java Runtime Environment (JRE). As a complete JRE, J2ME is comprised of a configuration, which determines the JVM used, and a profile, which defines the application by adding domain-specific classes. The configuration defines the basic run-time environment as a set of core classes and a specific JVM that run on specific types of devices. We'll discuss configurations in detail in the The profile defines the application; specifically, it adds domain-specific classes to the J2ME configuration to define certain uses for devices. We'll cover profiles in depth in the The following graphic depicts the relationship between the different virtual machines, configurations, and profiles. It also draws a

parallel with the J2SE API and its Java virtual machine. While the J2SE virtual machine is generally referred to as a JVM, the J2ME virtual machines, KVM and CVM, are subsets of JVM.

Both KVM and CVM can be thought of as a kind of Java virtual machine -- it's just that they are shrunken versions of the J2SE JVM and are specific to J2ME.

2. Developing J2ME applications

Introduction In this section, we will go over some considerations you need to keep in mind when developing applications for smaller devices. We'll take a look at the way the compiler is invoked when using J2SE to compile J2ME applications. Finally, we'll explore packaging and deployment and the role precertification plays in this process.

3. Design considerations for small devices

Developing applications for small devices requires you to keep certain strategies in mind during the design phase. It is best to strategically design an application for a small device before you begin coding. Correcting the code because you failed to consider all of the "gotchas" before developing the application can be a painful process. Here are some design strategies to consider:

* Keep it simple. Remove unnecessary features, possibly making those features a separate, secondary application.

* Smaller is better. This consideration should be a "no brainer" for all developers. Smaller applications use less memory on the device and require shorter installation times. Consider packaging your Java applications as compressed Java Archive (jar) files.

* Minimize run-time memory use. To minimize the amount of memory used at run time, use scalar types in place of object types. Also, do not depend on the garbage collector. You should manage the memory efficiently yourself by setting object references to null when you are finished with them. Another way to reduce run-time memory is to use lazy instantiation, only allocating objects on an as-needed basis. Other ways of reducing overall and peak memory use on small devices are to release resources quickly, reuse objects, and avoid exceptions.

4. Configurations overview

The configuration defines the basic run-time environment as a set of core classes and a specific JVM that run on specific types of devices. Currently, two configurations exist for J2ME, though others may be defined in the future:

* Connected Limited Device Configuration (CLDC) is used specifically with the KVM for 16-bit or 32-bit devices with limited amounts of memory. This is the configuration (and the virtual machine) used for developing small J2ME applications. Its size limitations make CLDC more interesting and challenging (from a development point of view) than CDC. CLDC is also the configuration that we will use for developing our drawing tool application. An example of a small wireless device running small applications is a Palm hand-held computer.

* Connected Device Configuration (CDC) is used with the C virtual machine (CVM) and is used for 32-bit architectures requiring more than 2 MB of memory. An example of such a device is a Net TV box.

5. J2ME profiles

What is a J2ME profile?

As we mentioned earlier in this tutorial, a profile defines the type of device supported. The Mobile Information Device Profile (MIDP), for example, defines classes for cellular phones. It adds domain-specific classes to the J2ME configuration to define uses for similar devices. Two profiles have been defined for J2ME and are built upon CLDC: KJava and MIDP. Both KJava and MIDP are associated with CLDC and smaller devices. Profiles are built on top of configurations. Because profiles are specific to the size of the device (amount of memory) on which an application runs, certain profiles are associated with certain configurations.

A skeleton profile upon which you can create your own profile, the Foundation Profile, is available for CDC.

Profile 1: KJava

KJava is Sun's proprietary profile and contains the KJava API. The KJava profile is built on top of the CLDC configuration. The KJava virtual machine, KVM, accepts the same byte codes and class file format as the classic J2SE virtual machine. KJava

contains a Sun-specific API that runs on the Palm OS. The KJava API has a great deal in common with the J2SE Abstract Windowing Toolkit (AWT). However, because it is not a standard J2ME package, its main package is com.sun.kjava. We'll learn more about the KJava API later in this tutorial when we develop some sample applications.

Profile 2: MIDP

MIDP is geared toward mobile devices such as cellular phones and pagers. The MIDP, like KJava, is built upon CLDC and provides a standard run-time environment that allows new applications and services to be deployed dynamically on end user devices. MIDP is a common, industry-standard profile for mobile devices that is not dependent on a specific vendor.

It is a complete and supported foundation for mobile application development. MIDP contains the following packages, the first three of which are core CLDC packages, plus three MIDP-specific packages.

* java.lang

* java.io

* java.util

* javax.microedition.io

* javax.microedition.lcdui

* javax.microedition.midlet

* javax.microedition.rms

CHAPTER 6

CONCLUSION & REMARKS

6.1 CONCLUSION

In conclusion, the proposed volunteer finding application presents a comprehensive solution to enhance the engagement between volunteers and organizations, addressing existing challenges within the volunteer ecosystem. By providing a user-friendly platform equipped with advanced search functionalities, real-time communication tools, and personalized matching, the application streamlines the process of finding meaningful volunteer opportunities that align with individual interests and skills. The integration of incentive structures, training resources, and feedback mechanisms further enriches the user experience, fostering motivation and continuous improvement for both volunteers and organizations. Through data-driven insights and a focus on community building, the application not only enhances volunteer participation but also contributes to a more effective and impactful approach to addressing social needs. Ultimately, this project aspires to transform good intentions into meaningful actions, creating a thriving volunteer community that positively impacts various sectors and fosters social change. By leveraging technology to connect individuals with opportunities, the application aims to empower volunteers, support organizations, and drive collective efforts toward a more engaged and socially responsible society.

6.2 Future Enhancements

The volunteer finding application has the potential for several future enhancements that can further improve its functionality and user experience. These enhancements could include:

1. **Mobile Application Development**: Creating a dedicated mobile app for both Android and iOS platforms can improve accessibility and convenience, allowing users to find and register for volunteer opportunities on the go.

2. **Advanced Analytics and Reporting**: Implementing more sophisticated analytics tools that provide organizations with detailed reports on volunteer engagement, demographic insights, and impact measurement can help refine strategies and improve program effectiveness.

3. **Gamification Features**: Introducing gamification elements, such as challenges, leaderboards, and rewards for achieving certain milestones, can increase engagement and motivate volunteers to participate more actively.

4. **Community Forums and Networking**: Adding features for community discussions or networking opportunities can foster collaboration among volunteers, allowing them to share experiences, ideas, and resources.

5. **Integration with Social Media**: Enhancing the application with social media integration can facilitate sharing of volunteer opportunities and experiences, potentially attracting a wider audience and encouraging more people to get involved.

6. **Volunteer Skill Development Programs**: Partnering with educational organizations to offer skill development or certification programs through the platform can provide volunteers with valuable training and enhance their employability.

7. **Multilingual Support**: Implementing multilingual support can broaden accessibility, allowing non-native speakers to navigate the application and participate in volunteer opportunities.

8. **AI-Powered Recommendations**: Utilizing artificial intelligence to provide personalized recommendations for volunteering opportunities based on users' past activities, preferences, and skills can enhance the matching process and improve user satisfaction.

9. **Offline Functionality**: Developing features that allow users to access certain information or functionalities offline can ensure that volunteers can still engage with the application in areas with limited internet connectivity.
10. **Integration with Local Services**: Collaborating with local businesses and service providers to offer discounts or benefits to volunteers can create additional incentives for participation, enhancing the overall volunteer experience.

These future enhancements aim to create a more dynamic, engaging, and effective platform that meets the evolving needs of volunteers and organizations. By continually improving the application, the project can foster a more connected and impactful volunteer community, ultimately driving greater social change.

CHAPTER 7

REFERENCES

[1] Beatrix sainz-De-Adajo,Jose Manuelgarcia-Alonso,José Javier Berrocal-Olmeda; Sergio Laso-Mangas; Isabel De La Torre-Díez "FOODSCAN: FOOD MONITORING APP BY SCANNING THE GROCERIES RECEIPTS"

[2] Mridula Goel,Aryan Agarwal,Namit Chandwani,Tanmay Dixit "BUILDING AN APPLICATION FRAMEWORK TO CONNECT NGOs AND VOLUNTEERS"

[3] Feng Zeng,Qiao Chen, Lin Meng, Jinsong Wu "VOLUNTEER ASSISTED COLLABRATIVE OFFLOADING AND RESOURCE ALLOCATION IN VEHICULAR EDGE COMPUTING."

[4] Madhab Nirola; Mousumi Gupta; Arpan Sharma"ASSESSMENT OF BLOOD DONOR INFORMATION USING KERNEL DENSITY MAPS"

[5] Boxuan Shan "A DESIGN OF VOLUNTEER COMPUTING SYSTEM BASED ON BLOCKCHAIN"

[6] M.I. Geraskin "IDENTIFICATION AND ANALYSIS OF INCENTIVE SYSTEM PARAMETERS FOR LARGE SOCIAL GROUPS OF VOLUNTEERS"

[7] Poonam Yadav; Ioannis Charalampidis; Jeremy Cohen; John Darlington; Francois Grey "A COLLABRATIVE CITIZEN SCIENCE PLATFORM FOR REAL-TIME VOLUNTEER COMPUTING AND GAMES"

[8] Prathamesh Pandharinath Wadekar, Shreyas Kumar Sachan, Sujal Kishor Patil, Harshal Mahesh Wagh, Manimala Mahato "V - VIBE: EFFICIENT VOLUNTEER MANAGEMENT USING MOBILE TECHNOLOGY"

[9] R. Han, W. Wang and J. Shi, "MOTIVATION OF SICHUAN EARTHQUAKE VOLUNTEERS AND ITS IMPLICATION FOR EMERGENGENCY MANAGEMENT”-2011

[10] D. Endo and K. Sugita, "A VOLUNTEER CLASSIFICATION METHOD FOR DISASTER RECOVERY”-2010

[11] C. -N. Lee and C. -Y. Yang, "AN INFORMATION SERVICE PLATFORM FOR HOSPITAL VOLUNTEER TEAM”-2016

[12] E. Saleh and C. Shastry, "TASK MIGRATION IN VOLUNTEER COMPUTING SYSTEMS”-2022

[13] M. I. Geraskin, "STATISTICAL ANALYSIS OF TRENDS IN DYNAMICS OF LARGE SOCIAL GROUPS OF VOLUNTEERS”-2023

[14] L. Butgereit and R. A. Botha, "AN ARCHITECTURE FOR SYNCHRONOUS MICRO-VOLUNTEERING IN AFRICA USING SOCIAL MEDIA”-2013

[15] A. Arora, A. Sharma and Chitra, "AID-UP A PLATFORM TO CONNECT NGO’S AND VOLUNTEER TEACHERS”-2024

[16] L. Li and M. N. Ulaganathan, "DESIGN AND DEVELOPMENT OF A CROWDSOURCING MOBILE APP FOR DISASTER RESPONSE”-2017

[17] B. Solemon, I. Ariffin and N. N. Azmi, "MOBILE PLATFORM FOR EXPLORING THE POTENTIAL OF VOLUNTEERED GEOGRAPHIC INFORMATION FOR ASSET REGISTER”-2014

[18] M. Goel, A. Agarwal, N. Chandwani and T. Dixit,”BUILDING AN APPLICATION FRAMEWORK TO CONNECT NGO’S AND VOLUNTEERS”-2021

[19] A. Dasgupta, S. K. Ghosh and P. Mitra, "A MOBILE VOLUNTEERES GEOGRAPHIC INFORMATION MANAGEMENT PLATFORM FOR RURAL HEALTH INFORMATICS”-2015

[20] Y. Hao, Z. Wang and X. Xu, "GLOBAL AND PERSONAL APP NETWORKS: CHARACTERIZING SOCIAL RELATIONS AMONG MOBILE APP'S"-2016