ONLINE CHILD ADOPTION

A PROJECT REPORT

Submitted by

KUMAR M	420618104020
NAVEEN KUMAR A	420618104028
NIRMAL KUMAR A	420618104029
VALLARASU K	420618104060

In partial fulfilment for the award of the degree

Of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

ANNAI TERESA COLLEGE OF ENGINEERING –THIRUNAVALUR

ANNA UNIVERSITY-CHENNAI 600 025

APRIL-2021

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report “**ONLINE CHILD ADOPTION”** is the bonafide work of “**KUMAR M (420618104020), NAVEEN KUMAR A(420618104028), NIRMAL KUMAR A(420618104029), VALLARASU K (420618104060)** ” who carried out the project work under my supervision

SIGNATURE

Mrs.A.RAMYA,M.Tech,

HEAD OF DEPARTMENT

ASSISTANT PROFFESOR

COMPUTER SCIENCE AND ENGINEERING

ANNAI TERESA COLLEGE OF ENGINEERING

THIRUNAVALUR-607 204

SIGNATURE

Mr.G.RAJKUMAR.,M.E

SUPERVISOR

ASSISTANT PROFFESOR

COMPUTER SCIENCE AND ENGINEERING

ANNAI TERESA COLLEGE OF ENGINEERING

THIRUNAVALUR-607 204

Submitted for the project viva-voice held on __________________

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We take this opportunity to express our gratitude to all, whose contribution in this project work can never be forgotten.

Our sincere thanks to our honourable chairman **Ln.Dr.PRAKASHMULL CHORDIA**, M.Com., Ph.D**.,** for this support during the entire course period.

We express our gratitude to our managing director **Dr.P.DINESH KUMAR CHORDIA,** M.Com.,Ph.D., for this support during the entri course period.

We thank our **PRINCIPAL Dr.J.KRISHNA KUMAR,** M.Tech., Ph.d., for this worthy encouragement to make this project as a successful one.

We express our thanks to our **HEAD OF THE DEPARTMENT Mrs.A.RAMYA**, M.Tech, for her valuable suggestion and guidance for the development completion of this project.

Word fail to express my gratitude to my **PROJECT GUIDE .**
Mr.G.RAJKUMAR.,M.E, who took special interest **on** my project and gave his consistent support and guidance during all stages of this project.

Finally, we thank all the **STAFF MEMBERS OF COMPUTER SCIENCE AND ENGINEERING** of our college who helped to complete this project. Above all we thank our **PARENTS AND FAMILY MEMBERS** for their constant support and encouragement for completing this project

ABSTRACT

Women can bear own children or adopt them. Extending economic theories of fertility, we Provide a first theoretical treatment of the demand for adoption. We show that the propensity to adopt a child increases in the degree of own altruism, infertility, relatedness to the child, costs of own child birth, and any adoption-specific monetary return that is received net of the costs of adopting the child. Our model makes several testable predictions which receive empirical support. These include a higher propensity to adopt among infertile adults, relatives, women with high earnings potential, and celebrities. Central to our analysis is the notion that own children and adoptees can be viewed as imperfect) substitutes whose demands are endogenous and jointly determined. Concerning individual motives for adoption, we show that the propensity to adopt a child increases in the degree of own altruism, infertility, the costs of own child birth, the relatedness (or otherwise founded sympathy) to a child that is orphaned or abandoned, adoption specify monetary return that is received net of the costs of adopting a child.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVATIONS

ASF	Apache Software Foundation
DNS	Domain Name Server
J2EE	Java 2 Enterprises Edition
JDBC	Java Database Connectivity
JNI	Java Native Interfaces
JSP	Java Server Pages
JVM	Java Virtual Machine
MSCS	Microsoft Cluster Server
MVC	Model View Controller
RMI	Remote Method Inovation
URL	Uniform Resource Locator

CHAPTER – 1

INTRODUCTION

1.INTRODUCTION

1.1 ABOUT THE PROJECT

. Every child deserves a family-whether it's birth family, or even inside a family-style orphanage. Millions of children worldwide need the care and support that only a family can give. A child's hope foundation relights the flame of hope in children's hearts by helping them find the real meaning of family in whatever their unique circumstances. For children without a home, we raise money to build and support family style orphanage to increase the number of children who are being cared for.For children already in the care of a family-style orphanage, we provide support in three major ways: First, we provide loans to prospective adoptive families to offset the steep cost of adoption .Second, we collect the resources needed to improve and expand existing orphanage. Lastly, we believe it's not enough to just send money andbuilding supplies. We send volunteers directly to orphanage to give children more of that they really need-family. Our volunteers become virtual aunts, uncle, brothers, sisters and grandparents to children in other countries. "ADOPTION may be defined as the process of providing parents for children for families when birth parents are unwilling or unable to care for their offspring". An international adoption occurs when single parent or couple from a particular country planning to adopt a child from another country. This type of adoption usually gone through an adoption agency. Sometimes, the adoptive parents do not even meet with the adopted child until the adoption processes were done. A person or couple have to be of good repute. According to the law, whoever has a record of violence, sexual assault,

emotional abuse or a crime against a child will disqualify them from adopting a child. According to the statistic report, the number of adoptions is different in all the states and territories. States with larger populations for example New South Wales, Victoria,Queensland has the highest number of adoptions, but lowest rate of adoptions per 100,000 population. Rates of adoptions were highest in Australian Capital Territory, the Northern Territory and Tasmania

1.2 OBJECTIVES

- system has a friendly approach to the user and also provides an easy access of organization
- So the parents and the charity organization is to have about their children.details in fingertips.
- The system is a friendly approach because the parents feel comfortable to communicate with the charity people about the child details.

CHAPTER – 2

SYSTEM ANALYSIS

2 SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

A recent cases in tamilnadu shows that the existing system has allowed child trafficking to take place for years under the guise of a perfectly legal adoption process.Front line investigation and documents available with it reveal that is not an isolated case. Bending rules, circumventing norms and following illegal and unethical ways to source children and sell them to foreigners under the guise of adoption is not uncommon among some agencies in Tamil nadu. While more than Indian parents have registered for children the agency has not shown interest to complete home studies of the families.

2.2 DISADVANTAGES OF EXISTING SYSTEM

- Several registers or records are not properly maintained the registration of society is not proper.
- The main disadvantage of the existing system is the parents should go for searching child all charities and adopt the child which they choose to give a new opportunity.
- Online child adoption if the existing system is not secure.

2.3 PROPOSED SYSTEM

- No two families are alike. No two families are alike. No two children are alike.That's why A child's hope foundation works to help children in a variety of situation and circumstances because every child deserves the hope and comfort of a family.
- Computer professionals - help the person who does not have knowledge of

internet to use the system. The person who does not have the knowledge of the computer can seek the computer professionals. The computer professional can teach the users in the form of tutorial Center or the computer professional can directly go to the users place and teach about the computer. Or else the user can learn from the website they have created.

- In this web application it is more secure than existing system, in this proposed system we are introducing a secured registration form for the admin reference.

2.4 ADVANTAGES OF PROPOSED SYSTEM

- The features of the proposed system are that it has the full details about the children, and their characteristics in the finger tips.
- The proposed system has a friendly approach to the user and also provides an easy access of organization .
- About the child the charity organization should be well know because the parents can approach them about the child details.
- So the parents and the charity organization is to have about their children.details in fingertips.
- The system is a friendly approach because the parents feel comfortable to communicate with the charity people about the child details.

CHAPTER – 3

SYSTEM SPECIFICATION

3. SYSTEM SPECIFICATION

3.1 HARDWARE REQUIREMENTS

- System : Pentium IV 2.4 GHz.
- Hard Disk : 40 GB.
- Floppy Drive : 1.44 Mb.
- Monitor : 15 VGA Colour.
- Mouse : Logitech.
- Ram : 512 Mb.

3.2 SOFTWARE REQUIREMENTS

- Operating system : - Windows XP/7.
- Coding Language : JAVA/J2EE
- Data Base : MYSQL

CHAPTER – 4

SYSTEM DESIGN

4.SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

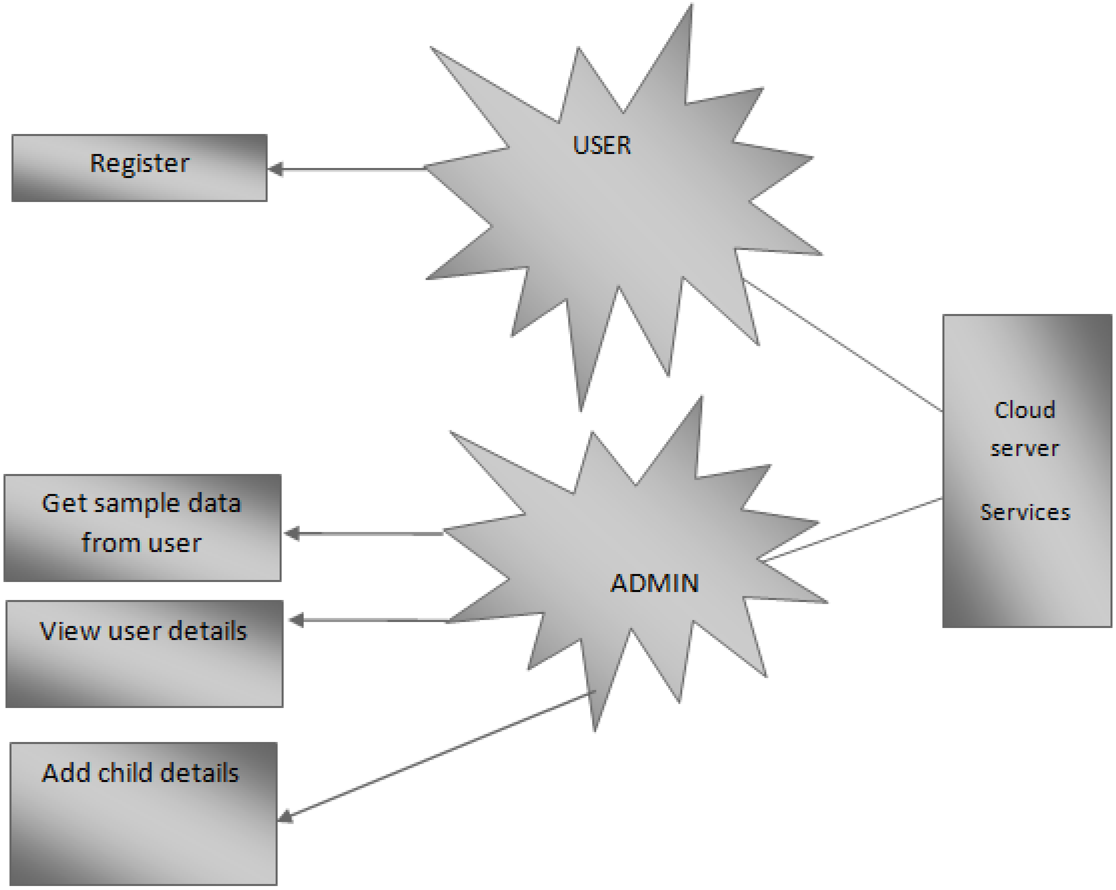


Figure:4.1 System Architecture

4.2 DATA FLOW DIAGRAM

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

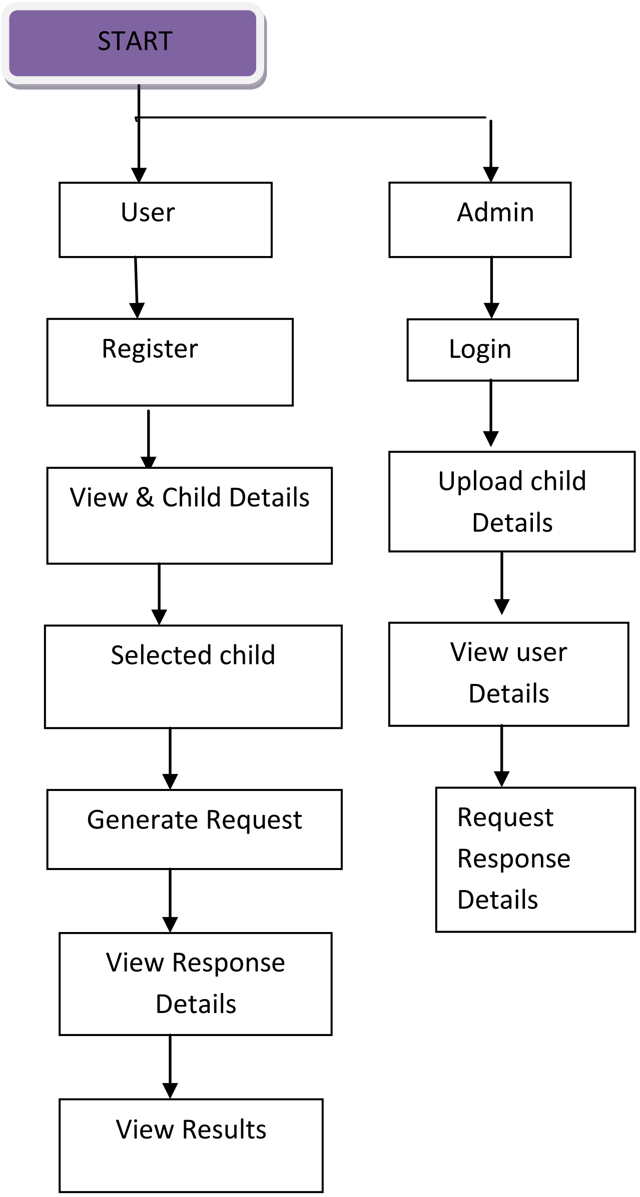


Figure:4.2 Data Flow Diagram

4.3 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

4.4 GOALS

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.

4.5 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

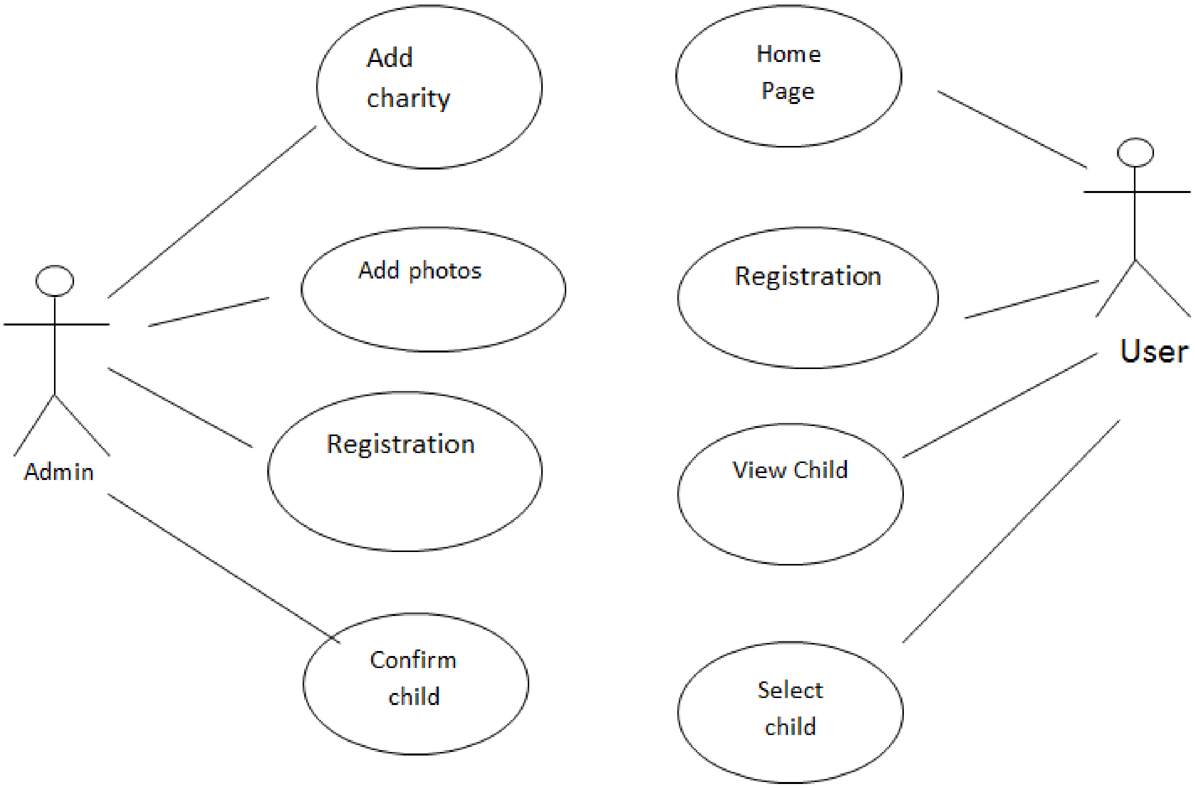


Figure:4.5 Use Case Diagram

4.6 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) 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 the classes. It explains which class contains information.

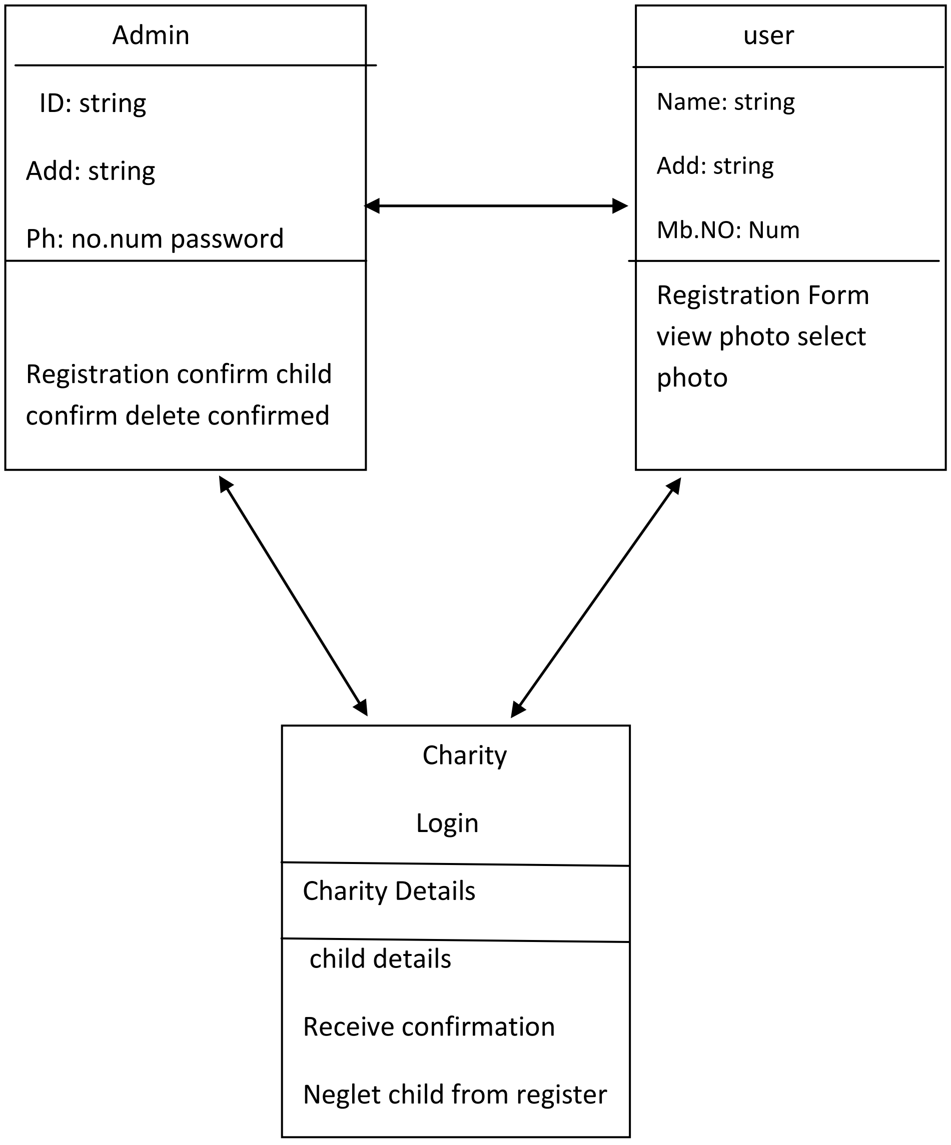


Figure:4.6 Class Diagram

4.7 SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

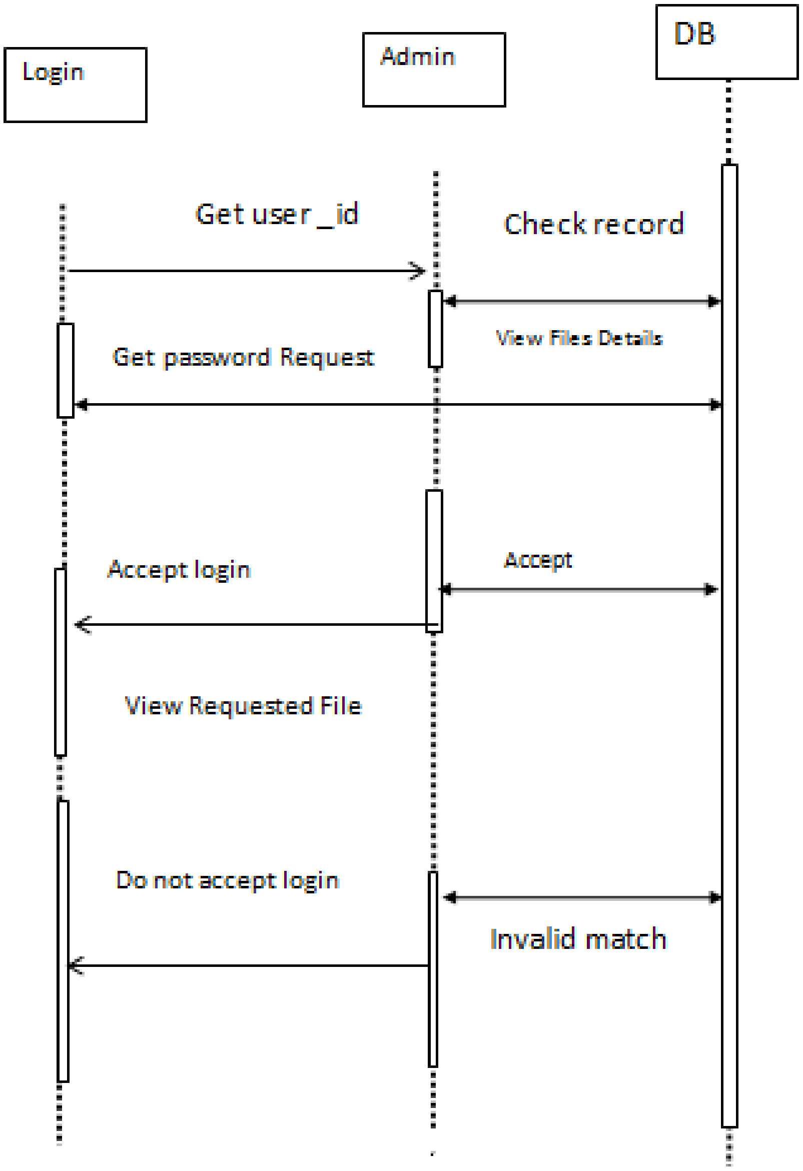


Figure:4.7 Sequence Diagram

4.8 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

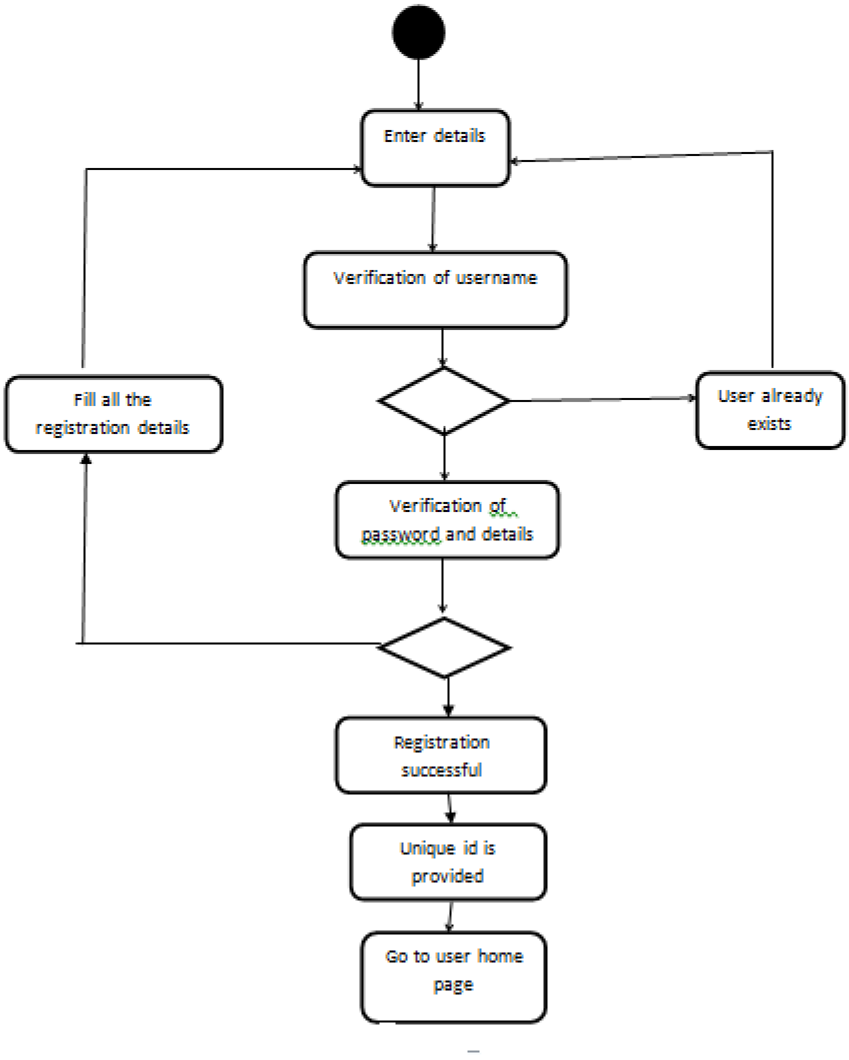


Figure:4.8 Activity Diagram

CHAPTER – 5

MODULE DESCRIPTION

5. MODULE DESCRIPTION

5.1 LIST OF MODULES

- ❖ Admin modules
- ❖ User module
- ❖ Services module

5.2 MODULE DESCRIPTIONS

5.2.1 ADMIN MODULE

In Data admin module, Initially Data user must have to register their detail. After successful registration data owner can login and upload files into cloud server . Data Owner(admin) can approve or reject the file request sent by data users. After request approval data owner will send the verification object through mail.

5.2.2 USER MODULE

- User register with their details and Login with registered details.
- In User module, Initially Data Users must have to register their detail and after login he/she has to verify their login . Data Users can search all the files upload by data admin. He/she can send request to the files and then request will send to the admin. If admin approve the request then he/she will receive verification object key in registered mail

5.2.3 SERVICES MODULE

The functionality of services module in this web application is to view all the charity trust over the state or country in this web application the information about the charity is viewed clearly

CHAPTER – 6

SYSTEM TESTING

6. SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.1 TYPES OF TESTS

6.1.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.1.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.1.3 FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

6.1.4 SYSTEM TEST

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test.

System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

6.1.5 WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

6.1.6 BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works

6.1.7 UNIT TESTING

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

TEST STRATEGY AND APPROACH

Field testing will be performed manually and functional tests will be written in detail.

TEST OBJECTIVES

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

FEATURES TO BE TESTED

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

6.2 INTEGRATION TESTING

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

6.3 ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

CHAPTER – 7

SYSTEM STUDY

7. SYSTEM STUDY

7.1 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

7.1.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

7.1.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

7.1.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER – 8

CONCLUSION & FUTURE ENCHANCEMENT

8. CONCLUSION & FUTURE ENCHANCEMENT

8.1 CONCLUSION

We know that children require nurturing environments to thrive today uation..and to have promising prospects for tomorrow.common sense,confirmed by my research, tells that children who are neglected or orphaned by certain reason will do best if removed from the bleak and placed permanently with families where they will receive the kind of nurturing likely to help them removed from their situation.thus ,it is clearly stated that child adoption has more advantages than disadvantages and child adoption should be strongly encouraged ignoring the little disadvantages

8.2 FUTURE ENHANCEMENTS

We are aiming to provide knowledge and new life for humans with a new adopted children with peaceful manner process via mobile phones or other system communication technologies we will provide all types of facilities to improve the enhancement of the adoption of children in the combined charities organization and also this will help to learn more technologies about the websites of child adoption

APPENDIX 1

SOFTWARE ENVIRONMENT

SOFTWARE ENVIRONMENT

A1.1 Java Technology

Java technology is both a programming language and a platform.

A1.2 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.

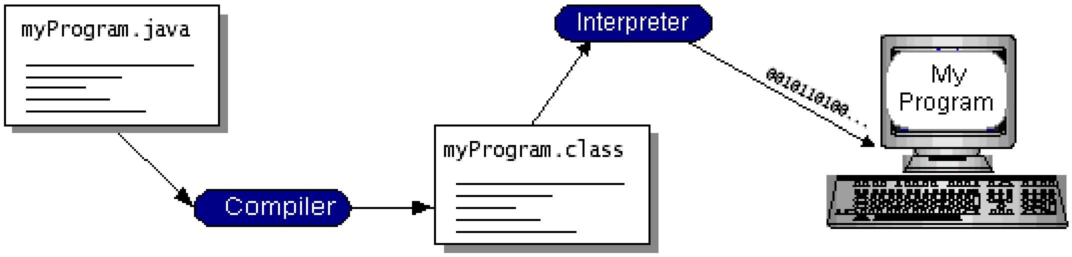


Figure: A1.2 The Java Programming Language

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.

A1.3 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 differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

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.

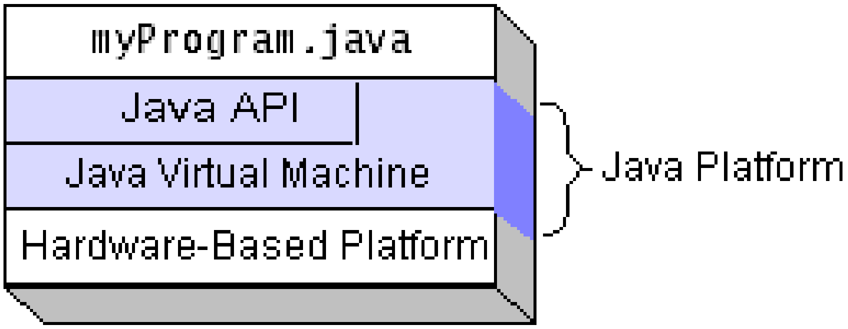


Figure: A1.2 3 The Java Platform

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.

A1.4 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. The general-purpose, high-level Java programming language is also a powerful software platform. Using the generous API, you can write many types of programs.

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.

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^{TM}, can plug into existing component architectures.
- **Object serialization**: Allows lightweight persistence and communication via Remote Method Invocation (RMI).
- **Java Database Connectivity (JDBC^{TM})**: 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.

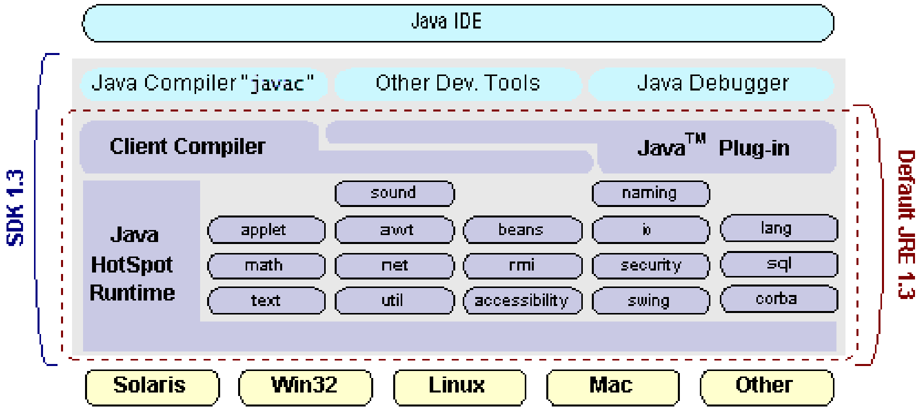


Figure: A1.2 4 The Java Platform IDE

A1.5 How Will Java Technology Change My Life

We can’t promise you fame, fortune, or even a job if you learn the Java programming language. Still, it is likely to make your programs better and requires less effort than other languages. We believe that Java technology will help you do the following:

- **Get started quickly**: Although the Java programming language is a powerful object-oriented language, it’s easy to learn, especially for programmers already familiar with C or C++.
- **Write less code**: Comparisons of program metrics (class counts, method counts, and so on) suggest that a program written in the Java programming language can be four times smaller than the same program in C++.
- **Write better code**: The Java programming language encourages good coding practices, and its garbage collection helps you avoid memory leaks. Its object orientation, its JavaBeans component architecture, and its wide-ranging, easily extendible API let you reuse other people’s tested code and introduce fewer bugs.
- **Develop programs more quickly**: Your development time may be as much as twice as fast versus writing the same program in C++. Why? You write fewer lines of code and it is a simpler programming language than C++.
- **Avoid platform dependencies with 100% Pure Java**: You can keep your program portable by avoiding the use of libraries written in other languages. The 100% Pure Java^{TM} Product Certification Program has a repository of historical process manuals, white papers, brochures, and similar materials online.

- **Write once, run anywhere**: Because 100% Pure Java programs are compiled into machine-independent byte codes, they run consistently on any Java platform.
- **Distribute software more easily**: You can upgrade applets easily from a central server. Applets take advantage of the feature of allowing new classes to be loaded "on the fly," without recompiling the entire program.

A1.6 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 a 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. There is a 16-bit and a 32-bit version of this program and each maintains separate list of ODBC data sources

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 ODBC drivers is transparent to the ODBC application program. In a client/server environment, the ODBC API even handles many of the network issues for the application programmer.

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.

A1.7 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.

JDBC was announced in March of 1996. It was released for a 90 day public review that ended June 8, 1996. Because of user input, the final JDBC v1.0 specification 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.

A1.8 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. **SQL 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.

3. **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.

4. **Provide a Java interface that is consistent with the rest of the Java system**

   Because of Java’s acceptance in the user community thus far, the designers feel that they should not stray from the current design of the core Java system.

5. **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.

6. **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.

7. **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 proceed 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
Object-oriented
Distributed
Interpreted
Robust
Secure
Architecture-neutral
Portable
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.

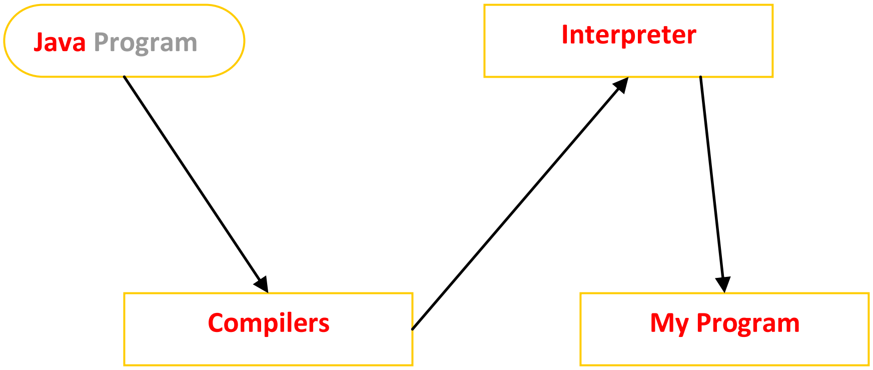


Figure:A1.8 JDBC

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.

A1.9 Networking

TCP/IP stack

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

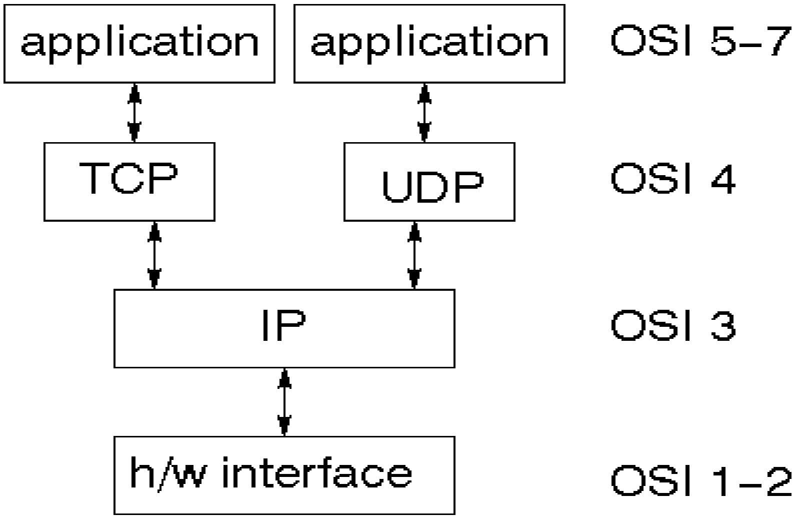


Figure: A1.9 Networking

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.

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". 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

A1.10 JFree Chart

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.

```
#include <sys/types.h>
#include <sys/socket.h>
```

```
int socket(int family, int type, int protocol);
```

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.

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.

A1.11 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

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

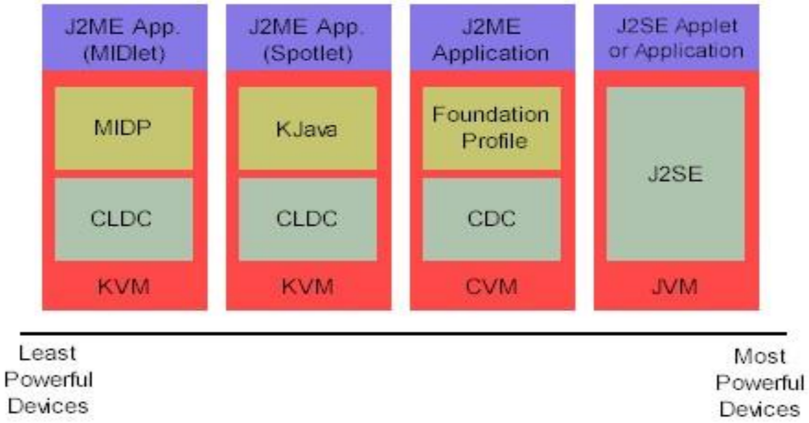


Figure: A1.11 J2ME (Java 2 Micro edition)

. 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 preverification 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.

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. 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. CLDC packages, plus three MIDP-specific packages.

* java.lang

* java.io

* java.util

* javax.microedition.io

APPENDIX 2

SOURCE CODE

A2.SOURCE CODE

```
<%@ page import="java.sql.*"%>
<%@ page import="java.io.*"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<style type="text/css">
.wrapper{
height:auto;
width:1024px;

margin: 0px;
  border: 1px solid #4CAF50;

}
#top{
height: 100px;
  width:1024px;
  background: url(childcareoverview.jpg);
  background-repeat: no-repeat;
  background-size: 1024px;
  }
  #men{
  height: 50px;
  width:1024px;
  float:left;
  padding-right:0px;

   background-color:Gray;
  }
  #cent{
   height: 500px;
  width:1024px;
  float:left;
  }
  #b1
{
float:left;
height:490px;
width:300px;
background-color:#999933;

}
  #b2
{
float:left;
height:490px;
width:350px;
}
    #b3
{
float:left;
height:490px;
width:374px;
background-color:#ff6347;

}
    #b4
{
float:left;
height:490px;
width:374px;
margin:50px 0px 0px 50px;
```

```
}
 #r{
 padding-right: 30px;

 }
  #r1{
 color: #ffffff;
 }
 a {
 text-decoration: none;
 color: #ffffff;
</style>
        <% String path = request.getContextPath();%>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
        <script type="text/javascript">
          function validateform()
          {
                if(document.registerForm.username.value.trim()=="" || document.registerForm.username.value.trim()==null)
                 {
                    alert("Enter the username");
                    document.registerForm.username.focus();
                    return false;
                 }
               else if(document.registerForm.userpass.value.trim()==""||document.registerForm.userpass.value.trim()==null)
                 {
                    alert("Enter the Password");
                    document.registerForm.userpass.focus();
                    return false;
                 }
```

```
            }
            function requestSend(childId, userId){

                        var
url='<%=path%>/loginServlet?control=newRequest&childId='+childId+'&userId='
+userId;

                        document.InboxForm.action = url;
                              document.InboxForm.submit();
                              return false;
            }
        </script>
</head>

<body  >
<center>
<div class="wrapper">
<div id="top">

</div>
<div id="men">
<table width="100%">
<tr>
<td id="r1" align="left">Welcome
<%=session.getAttribute("name")%></td><td>&nbsp;&nbsp;</td>
<td id="r" align="right"><a href="index.jsp">Log out</a></td>
</tr>
</table>
</div>
<div id="cent"  style="border: 1px solid gray;">

<div id="b2">
<%
Blob image = null;
```

```
Connection con = null;
byte[ ] imgData = null ;
Statement stmt = null;
ResultSet rs = null;
try {
Class.forName("com.mysql.jdbc.Driver");
con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/childdb","root","root"
);
stmt = con.createStatement();
rs = stmt.executeQuery("select c.child_id, first_name, last_name, photo, age,
gender, r.status,r.child_id  from contacts c left outer join request r on c.child_id =
r.child_id");

out.println("<center><table border=1>");
out.println("<tr><td nowrap
width='70%'>Information</td><td>Photo</td><td></td><td>Status</td></tr>");
String result = "",next="", app = "";
boolean hasRecord = false;
while (rs.next()) {
        image = rs.getBlob(4);
        //if(rs.getString(7)!= null){
        app = rs.getString(7)!= null && rs.getString(7).equals("1")? "Aproved" :
"Pending";
        next = rs.getString(8)!= null ? "" : "<Button
onclick=requestSend('"+rs.getString(1)+"','"+session.getAttribute("user_id")+"')>R
equest</button>";
        result = "First Name: " + rs.getString(2) + "<br>Last Name: " +
rs.getString(3) + "<br>Age: " + rs.getString(5) + "<br>Gender: " +  rs.getString(6);
        out.println("<tr><td nowrap>"+result+"</td><td><img width='150'
height='150' src='image.jsp?child_id="+rs.getInt(1)+"' /></td><td>"+next
+"</td><td>"+app +"</td></tr>");
        hasRecord = true;
}
if(!hasRecord)
        out.println("<tr><td colspan=6>No Records to Dislay</td></tr>");

out.println("</table></center>");

} catch (Exception e) {
out.println("Unable To Display image");
out.println("Image Display Error=" + e.getMessage());
} finally {
try {
rs.close();
stmt.close();
con.close();
} catch (SQLException e) {
e.printStackTrace();
}
}
%>
<form name='InboxForm' method='post'></form>
</div>
<div id="b4">

</div>
</div>
</div>
</center>
</body>
</html>
<%@ page import="java.sql.*"%>
<%@ page import="java.io.*"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
<title>JSP Page</title>
<% String path = request.getContextPath();%>
<link rel="stylesheet" href="style.css" />
<script type="text/javascript">
          function validateform()
          {
                if(document.registerForm.username.value.trim()=="" ||
document.registerForm.username.value.trim()==null)
                 {
                     alert("Enter the username");
                     document.registerForm.username.focus();
                     return false;
                 }
              else
if(document.registerForm.userpass.value.trim()==""||document.registerForm.userp
ass.value.trim()==null)
                 {
                     alert("Enter the Password");
                     document.registerForm.userpass.focus();
                     return false;
                 }
          }
          function requestSend(childId, userId){

           var
url='<%=path%>/loginServlet?control=approveRequest&childId='+childId+'&user
Id='+userId;

           document.InboxForm.action = url;
                 document.InboxForm.submit();
                 return false;
        }
       </script>
</head>
```

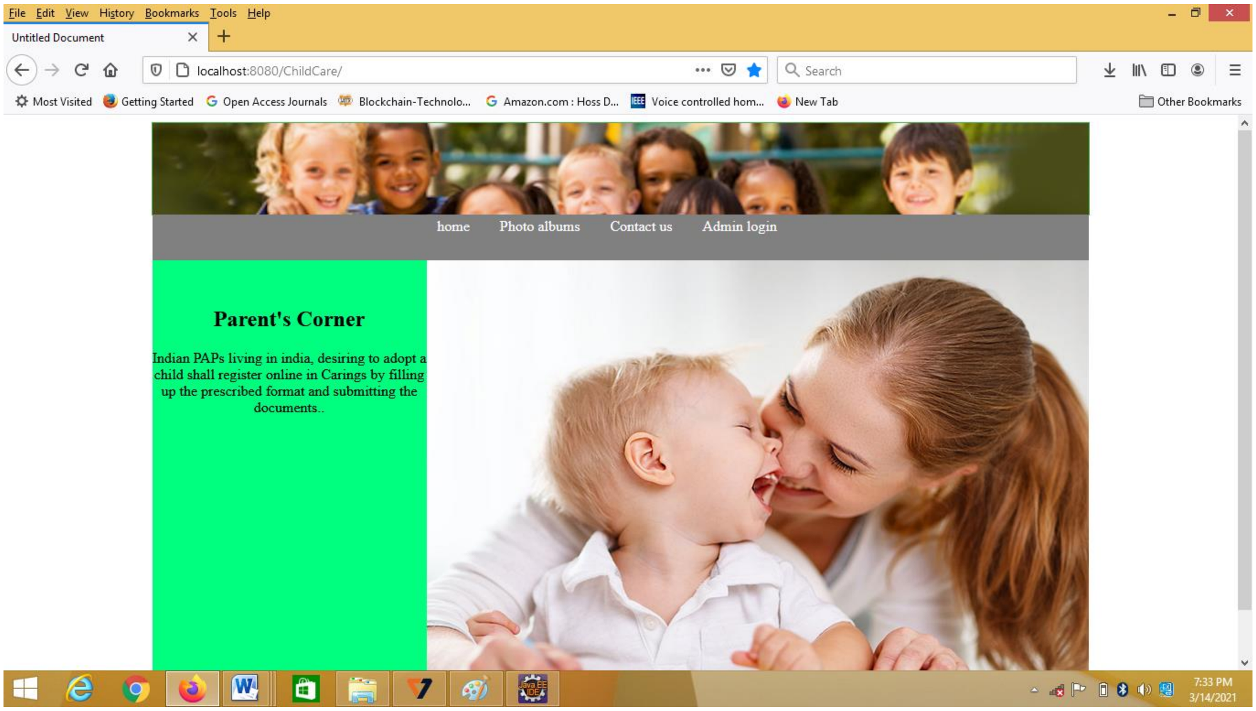
```
<body>

        <center>
                <div id="wrapper">
                        <div class="header">

                                <h2>Child Care System</h2>
                        </div>
<div id="men">
<table width="100%">
<tr>
<td id="r1" align="left">Welcome
<%=session.getAttribute("name")%></td><td>&nbsp;<a
href="registerList.jsp">Click hered to view user details</a>&nbsp;</td>
<td id="r2"><a href="AddChild.jsp">Click hered to Add child</a></td>
<td id="r" align="right"><a href="index.jsp">Log out</a></td>
</tr>
</table>
</div>
                        <div class="content" style="height: 700px;">
                                <div style="width: 300px; margin: 0 0 50px 0;">

                                        <%
Blob image = null;
Connection con = null;
byte[ ] imgData = null ;
Statement stmt = null;
ResultSet rs = null;
try {
Class.forName("com.mysql.jdbc.Driver");
con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/childdb","root","root"
);
stmt = con.createStatement();
```

```
rs = stmt.executeQuery("select c.child_id, c.first_name, c.last_name, c.photo,
c.age, c.gender, r.user_name,r.mobile,r.user_id,re.status from contacts c left outer
join request re on re.child_id = c. child_id left outer join t_register r on re.user_id =
r.user_id");

out.println("<br><br><table border=1>");
out.println("<tr><td
nowrap><b>Information</b></td><td><b>Photo</b></td><td
nowrap><b>Request from</b></td><td><b>Status</b></td></tr>");
String a = "", b="", result = ""; boolean hasRecord = false;
while (rs.next()) {
        a = rs.getString(7) != null ? "Name:" + rs.getString(7) + "<br> Phone:" +
rs.getString(8): "";
        b = rs.getString(7) != null && rs.getString(10)!= null &&
rs.getString(10).equals("1") ? "Approved" : rs.getString(7) != null &&
rs.getString(10)==null ?"<Button
onclick=requestSend('"+rs.getInt(1)+"','"+rs.getString(9)+"')>Approve</button>" :
"";
        image = rs.getBlob(4);
        result = "First Name: " + rs.getString(2) + "<br>Last Name: " +
rs.getString(3) + "<br>Age: " + rs.getString(5) + "<br>Gender: " +  rs.getString(6);
        out.println("<tr><td nowrap>"+result+"</td><td><img width='50'
height='50' src='image.jsp?child_id="+rs.getInt(1)+"' /></td><td>"+ a
+"</td><td>"+b+"</td></tr>");
        hasRecord = true;
}

if(!hasRecord)
        out.println("<tr><td colspan=6>No Records to Dislay</td></tr>");
out.println("</table>");

} catch (Exception e) {
        e.printStackTrace();
out.println("Unable To Display image");
out.println("Image Display Error=" + e.getMessage());
}finally {
        try {
                rs.close();
                stmt.close();
                con.close();
                } catch (SQLException e) {
                e.printStackTrace();
                }
                }
                %>
                </div>
                </div>

                </div>
                <form name='InboxForm' method='post'></form>
                   </center>
                   </body>
                </html>
```

APPENDIX 3

SCREEN SHOTS

A3.SCREEN SHOTS



Parent's Corner

Indian PAPs living in india, desiring to adopt a child shall register online in Carings by filling up the prescribed format and submitting the documents..

Figure:A3 Home Page

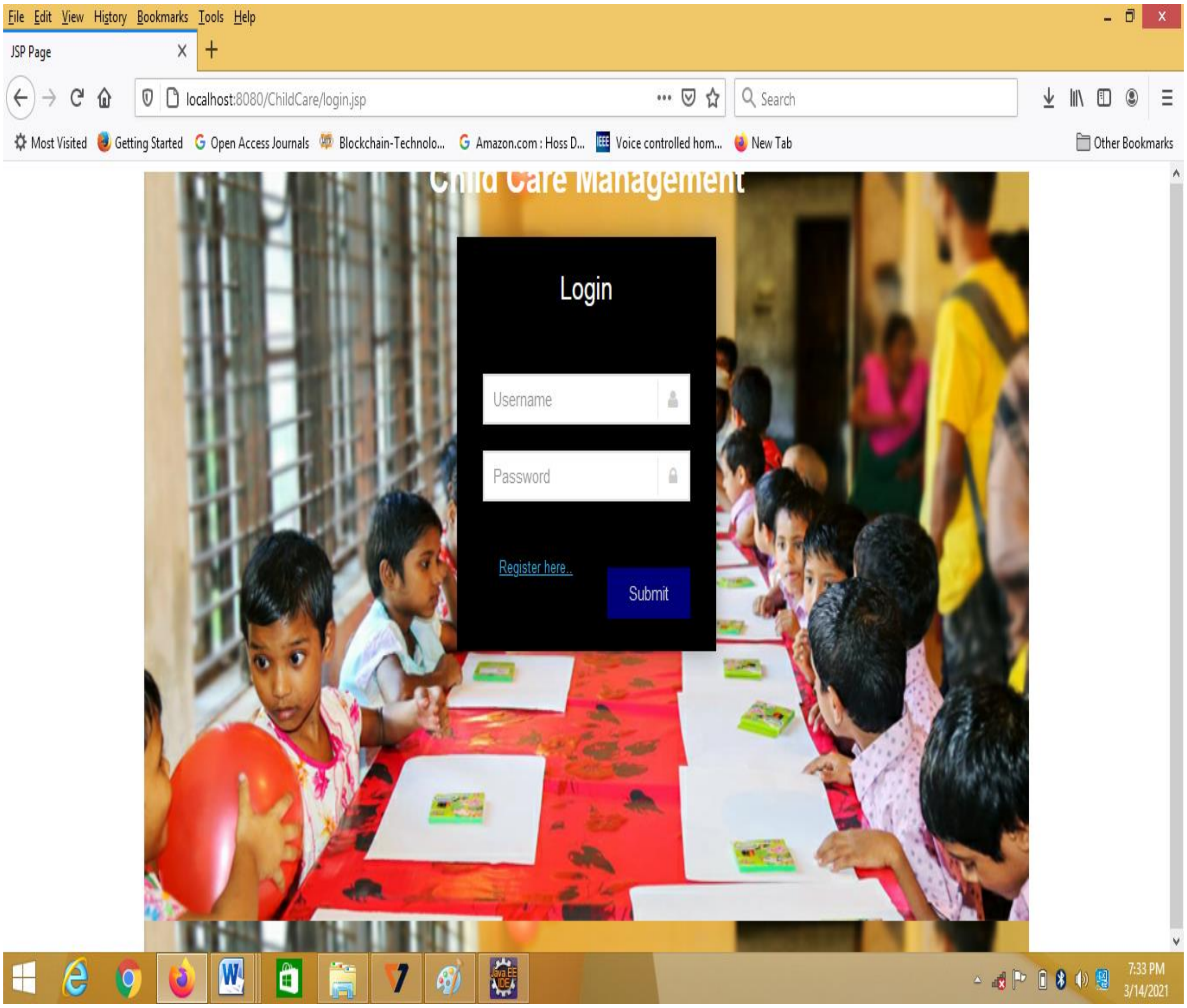
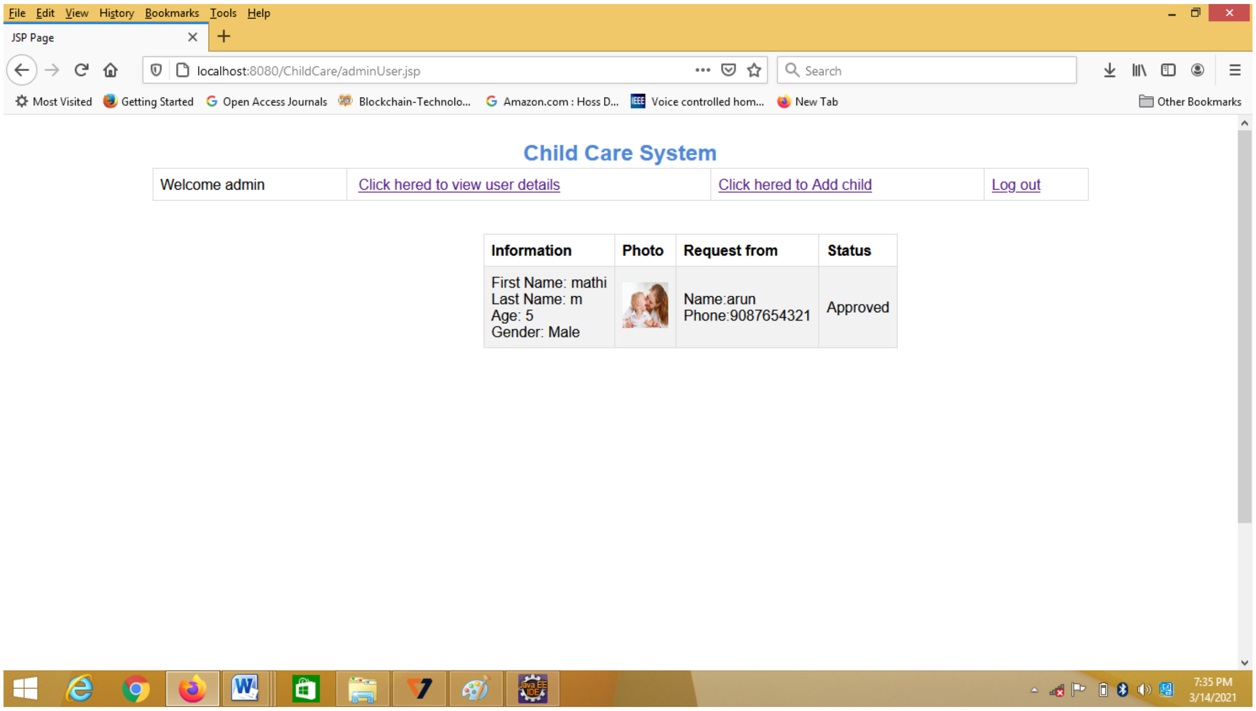


Figure:A3 User login Page

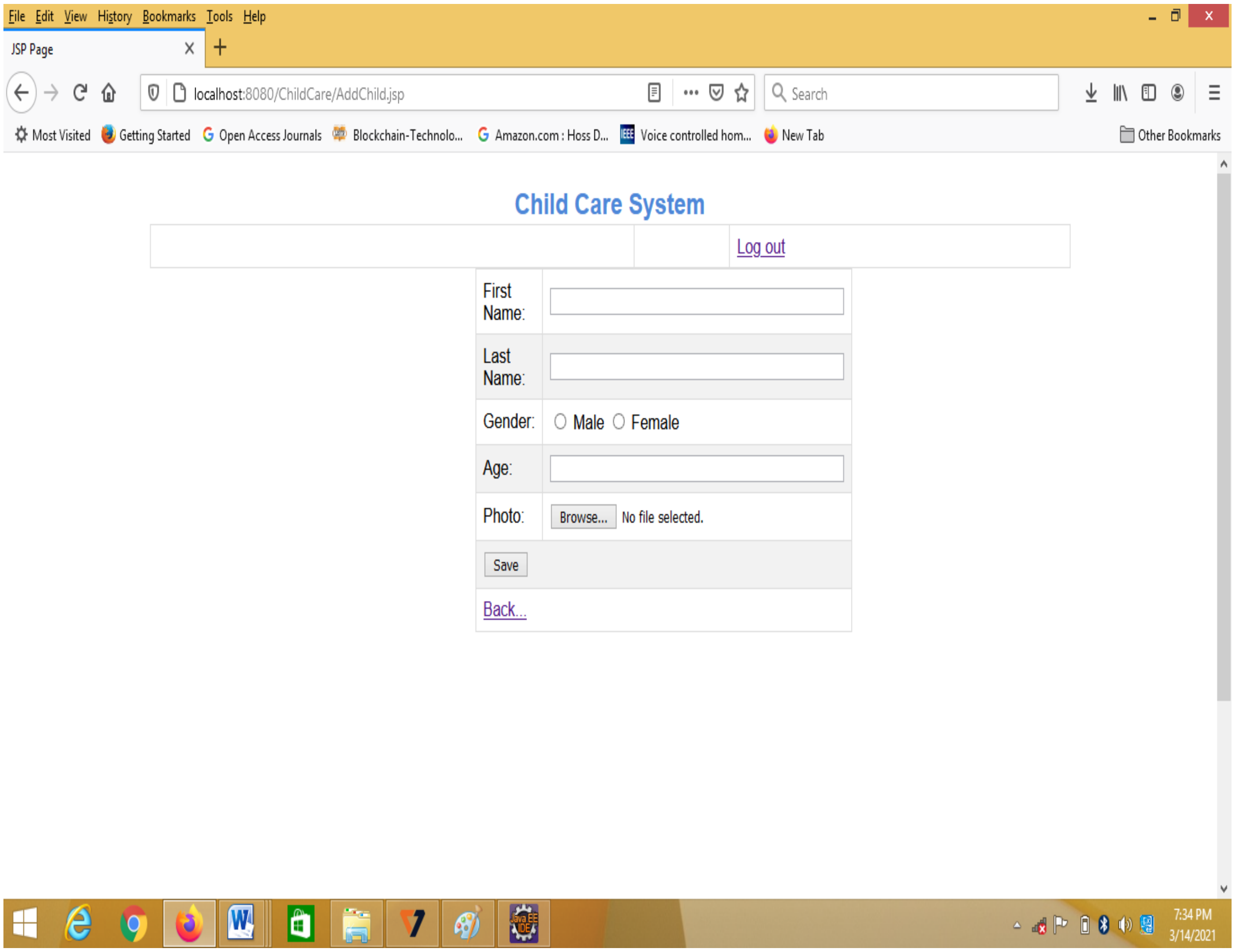


Child Care System

Welcome admin	Click hered to view user details	Click hered to Add child	Log out

Information	Photo	Request from	Status
First Name: mathi Last Name: m Age: 5 Gender: Male		Name:arun Phone:9087654321	Approved

Figure:A1 Admin login



Child Care System

Log out

First Name:	
Last Name:	
Gender:	○ Male ○ Female
Age:	
Photo:	Browse... No file selected.
Save	
Back...	

Figure:A3 Add child

A3. REFERENCES

1. www.iimidr.ac.in
2. www.globalgiving.org
3. www.adoptionpolicy.org