# TABLE OF CONTENTS

**CHAPTER NO. TITLE PAGE NO.**

**ABSTRACT i**

**LIST OF FIGURES ii**

**1. INTRODUCTION** 7

1.1 Purpose 7

1.2 Scope 8

1.3 Aim 8

1.4 Objectives . 8

1.5 Methodologies 9

**2. ANALYSIS** 9

2.1 Introduction 10

2.2 Data Flow Diagram 11

2.3 Specific Requirements 14

2.3.1 Functional Requirement 14

2.3.2 Non-Functional Requirement 15

2.4 External Interfaces 16

2.4.1 User Iterfaces 16

2.4.2 Software System Attribute 17

2.4.3 Database Requirements 18

2.4.4 Performance Requirements 18

**3. DESIGN** 19

**4. SOFTWARE DEVELOPMENT METHODOLOGY** 20

**5. SYSTEM TESTING** 22

**6. OUTPUTS** 24

**7. CONCLUSION** 25

**8. FUTURE ENHANCEMENT** 25

**BIBILIOGRAPHY** 26

**APPENDIX** 26

## Abstract

The purpose of ZIPPY SOFTWARE is to make efficient communication between two peers without involvement of any external server services. The main objective of this software is to make efficient use of the intra- infrastructure network in any organisation and establish real-time text, audio and file sharing services between all the clients who are connected to the same network.

It is purely a desktop based service and involves no external use of any other service to establish communication between two peers. Different users connected through Zippy can text chat, audio call and send and receive files to and from each other. As it is a P-P protocol connection, no delay involving the transmission of data to and from server is observed. Zippy is purely intra-based and meant to be used only within an organisation’s internal network infrastructure.

It is purely platform independent and can be used across cross-OS computing systems having support for JAVA. Implementation of this software is varied ranging from institutional organisations, hospitals, corporate offices and all those fields where intra-communication plays a vital role.

## LIST OF FIGURES

Page no.

# INTRODUCTION

This document will propose all features and procedures to develop the system. This document specially containing details about objectives, scope limitation, process model, primary requirements, team development, possible project risks, project schedule, and finally monitoring and reporting mechanisms.

ZIPPY is very useful for Educational Institute and official to connect one another easily, save the delay involved in the communication process by taking the help of intra-structure network. It will help both the Institutes and offices to connect peers to peers easily. But the disadvantages for this software is it cannot be used if a particular system is not connected to a particular network to which all the other peers are connected.

By the effective use of this software, organisations involving large scale communication within peers can achieve the same within less time and with higher efficiency.

## 1.1 Purpose

The purpose of this software is to provide efficient communication between peers or clients within a same network infrastructure. External use of any other services involving transmission of bandwidth is avoided to the maximum which helps in achieving seamless communication in a more efficient and faster way.

Users do not need to connect to the internet to use this software. Just being connected to the same network infrastructure would suffice. Thus it involves cost-cutting in the form of avoidance of use of external services such as servers and internet bandwidth of ISPs.

## 1.2 Scope:

The scope of this software is to provide text as well as audio communication within peers over the local area network. Files of large sizes can be seamlessly transferred without any loss as it doesn’t involve any external use of servers. Thus the integrity is maintained without any third party listening to a particular conversation between two peers. It is light-weight and can easily run over any LAN.

**Benefits:**

This website reduces the manual work, maintaining accuracy, increasing efficiency and saving time. Also institutes or office need not go to develop a new software each time, instead they just put the id and connected both of them. For students and employee, it saves time of going to far away centers and also they can view their information then and there.

## 1.3 Objectives

**a) General Objective**

The general objective of ‘**Zippy Software'** is to create a platform for online creation, actual sitting and make sharing some files.

**b) Specific Objectives**

1. Enables the peer to connect to a particular peer using IP
2. Enables to peer to connect to many peers listening over the same network
3. Enables text chat between peers
4. Enables audio calls between peers
5. Enables file sharing between peers

# 2. ANALYSIS

## 2.1. INTRODUCTION

**FUNCTIONAL REQUIREMENTS**

Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available. Each requirement should be uniquely.

**Check for its working condition:**

1. **SELECT USER AVAILABILITY:**

Input: USER IP ADDRESS

Output: USER CONNECTION CONFIRMATION

1. **TEXT COMMUNICATION:**

Input: TEXT CHAT SELECTED

Output: BOTH PEERS INITIATE TEXT CHAT

1. **AUDIO COMMUNICATION:**

Input: AUDIO CALL SELECTED

Output: BOTH PEERS INITIATE AUDIO CALL

1. **FILE SHARING:**

Input: USER SENDS A FILE

Output: USER RECEIVES THE FILE AND GIVES CONFIRMATION

**NON FUNCTIONAL REQUIREMENTS**

**Performance Requirements**

As it is going to be used by the entire concerned customer within the organization, the system should have a good performance in terms of speed and accuracy. The proposed system should be accurate and fast enough to handle huge data. It should provide fast communication between server and clients.

**Safety Requirements**

As the system is going to handle parts records for a long run eliminating the manual system, it is supposed to ensure the retaining of data avoiding or eliminating any probable cause for data loss.

**Security Requirements**

The software should not allow unauthorized access to any module of the system. Besides, it should maintain the privileges granted to users at various user levels.

**The prioritization of the software quality attributes are assumed as under:**

1) Accurate and hence reliable.

2) Secured.

3) Fast speed.

4) Compatibility.

*5)* Portability

.

## 2.2. DATA FLOW DIAGRAM(S):

 A DFD also known as ‘bubble chart’, has the purpose of clarifying system requirements and identifying major transformations. It shows the flow of data through a system. It is a graphical tool because it presents a picture. The DFD may be partitioned into levels that represent increasing information flow and functional detail. Four simple notations are used to complete a DFD. These notations are given below:-

**DATA FLOW:-** The data flow is used to describe the movement of

information from one part of the system to another part. Flows represent data in motion. It is a pipe line through which information flows. Data flow is represented by an arrow.

**PROCESS:-**  A circle or bubble represents a process that transforms

incoming data to outgoing data. Process shows a part of the system that transform inputs to outputs.

**EXTERNAL ENTITY:-** A square defines a source or destination of system data. External entities represent any entity that supplies or receive information from the system but is not a part of the system.

## 2.3. SPECIFIC REQUIREMENTS:

This section provides software requirements to a level of detail sufficient to enable designers to design the system and testers to test the system.

### 2.3.1 Functional Requirements

1. It deals with the functionalities required from the system which are as follows: The software will help the colleges/organizations/companies to connect each other without no internet.
2. Only authorized persons can access related details.
3. Those peoples are connect with each other through this software ,they are contact through audio calling.
4. User can contact with another user with the help of text message.
5. Both users can share their files with in a second when they are connect with each other.
6. It will be separate functionalities with the separate buttons.

**The technologies used to develop this site are:-**

The GUI part of the site is developed using JAVA SWING while the middle-ware is dealt using pure JAVA language. For networking SOCKET PROGRAMMING is used.

**2.3.2 Non Functional Requirements**

They are the quality requirements that stipulate how well a software does what it has to do.

**Performance**

No. of terminals to be supported is dependent on the server that we will use at the time of deployment. The software application server used should provide good performance and ability to manage performance with techniques such as support for caching. Its performances is very simple and ver fast. Both the users can access to connect easily.

**Availability**

Zippy connect software has 24\*7 availability. It can be accessed for 24 hours a day. For this UPS support must be on the server site with a backup of at least 8 hours in case of power failure.

Users can take connect during when another user will present there, however can open software anytime to access other information.

**Reliability**

It means the extent to which program performs with required precision. The website developed should be extremely reliable and secure so that information about any information etc. is not leaked.

**Usability**

This Software should be user friendly and should require least effort to operate. The server used should provide services like session management to maintain sessions in the application.

This software is using without costs and no internet connection.

**Portability**

Zippy connect software is made using only Java which are platform independent and can be transported to other servers with minimum effort.

## 2.4. External Interface Requirements:

### User Interfaces:

* **Hardware interfaces:**
* Support for printer for printing results then and there.
* Screen resolution of at least 800X600 is required for proper and complete viewing of screens. Higher resolution will be accepted.
* **Software interfaces:**
* Any windows based operating system.
* MS Dos-for database.
* Notepad++ for developing code.
* **Communications interfaces**

Through Local area network connection.

### 2.4.2 SOFTWARE SYSTEM ATTRIBUTE

o **Reliability and Fault Tolerance**

Fault tolerance includes details about recovery from failure.

o **Security**

Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any

user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.

o **Software Quality Attributes**

Specify any additional quality characteristics for the product that will be

important to either the customers or the developers. Some to consider are:

adaptability, availability, correctness, flexibility, interoperability, maintainability,

portability, reliability, reusability, robustness, testability, and usability. Write

these to be specific, quantitative, and verifiable when possible. At the least, clarify

the relative preferences for various attributes, such as ease of use over ease of

learning.

o **Performance**

If there are performance requirements for the product under various

circumstances, state them here and explain their rationale, to help the developers

understand the intent and make suitable design choices. Specify the timing

relationships for real time systems. Make such requirements as specific as possible.

You may need to state performance requirements for individual functional

requirements or features.

### 2.4.3 DATABASE REQUIREMENT(S):

* **Databses:**
  + - Any windows based operating system.
    - MS Command Prompt-for database.
    - Notepad++ for developing code.

**2.4.4 PERFORMANCE REQUIREMENTS:**

This subsection specifies numerical requirements placed on the software or on the human interaction with the software, as a whole. Numerical requirements will include:

* 300 terminals will be supported at a time
* text ,audio and file sharing will be supported java.
* All the transactions will be processed within seconds.

### 2.4.5 Logical Database Requirements:

# 3. DESIGN

## 3.1. INTRODUCTION

Design is the abstraction of a solution; it is a general description of the solution to a problem without the details. Design is view patterns seen in the analysis phase to be a pattern in a design phase. After design phase we can reduce the time required to create the implementation.

In this chapter we are introduce context diagram, models, system architecture, principal system object, design model and object interface.

a) **Conceptual Data Modeling** analyzes the overall data requirement of the proposed

information system.

b) **Logical Data Base Design** transform the conceptual data model into a standard relation called relation based on relational database theory and a process called Normalization.

c) In **Physical Data Base Design** and Definition, one decides on the organization of the

database in computer storage (usually disk) and defines the physical structure of data base management system

## 3.2. BUSINESS RULE

A business rule is “a statement that defines or constraints some aspect of the business. It is intended to assert business structure or to control or influence the behavior of the business.”

Most organizations have a host of rules and/or policies that impact its database. But there are

also some business rules that cannot be represented in common data modeling notation. Those rules that cannot be represented in a variation of an entity-relationship diagram are stated in natural language and some can be represented in the relational data mode.

**3.3 DFD (Data Flow Diagram) :**

**3.4 ACTION CHARTS:**

**ACTION CHART FOR TRANSFER**

****

**ACTION CHART FOR USER CONNECTIVITY**

****

# 4. SOFTWARE DEVELOPMENT METHODOLOGY

The establishment and use of sound engineering principles in order to obtain economically developed software that is reliable and works efficiently on real machines is called software engineering.

Software engineering is the discipline whose aim is:

1. Production of quality software

2. software that is delivered on time

3. cost within the budget

4. satisfies all requirements.

Software process is the way in which we produce the software. Apart from hiring smart,

knowledgeable engineers and buying the latest development tools, effective software development process is also needed, so that engineers can systematically use the best technical and managerial practices to successfully complete their projects.

A software life cycle is the series of identifiable stages that a software product undergoes during

its lifetime .A software lifecycle model is a descriptive and diagrammatic representation of the software life cycle .A life cycle model represents all the activities required to make a software product transit through its lifecycle phases .It also captures the order in which these activities are to be taken .

Life Cycle Models

There are various life cycle models to improve the software processes.

WATERFALL MODEL

PROTOTYPE MODEL

ITERATIVE ENHANCEMENT MODEL

EVOLUTIONARY MODEL

SPIRAL MODEL

In the project, Waterfall model is followed.

WATERFALL MODEL



**Feasibility study**

The feasibility study activity involves the analysis of the problem and collection of the relevant information relating to the product. The main aim of the feasibility study is to determine whether it would be financially and technically feasible to develop the product.

o  **Requirement analysis and specification**

The goal of this phase is to understand the exact requirements of the customer and to document them properly.(SRS)

o  **Design**

The goal of this phase is to transform the requirement specification into a structure that is suitable for implementation in some programming language.

o  **Implementation and unit testing**

During this phase the design is implemented. Initially small modules are tested in isolation from rest of the software product.

o  **Integration and system testing**

In this all the modules are integrated and then tested altogether.

o  **Operation and maintenance** .

Release of software inaugurates the operation and life cycle phase of the operation.

The phases always occur in this order and do not overlap.

# 5. SYSTEM TESTING

## 5.1. INTRODUCTION

* Software testing is the process of executing a program with intension of finding errors in the code. It is a process of evolution of system or its parts by manual or automatic means to verify that it is satisfying specified or requirements or not.
* Generally, no system is perfect due to communication problems between user and developer, time constraints, or conceptual mistakes by developer.
* To purpose of system testing is to check and find out these errors or faults as early as possible so losses due to it can be saved.
* Testing is the fundamental process of software success.
* Testing is not a distinct phase in system development life cycle but should be applicable throughout all phases i.e. design development and maintenance phase.
* Testing is used to show incorrectness and considered to success when an error is detected.

## 5.2. System Overview

Testing included running the system over the Command prompt for connection purpose.

It was worth noting that the system does well in all the three browsers named above. The above browsers were chosen because they are the frequently used browsers all over the world. Many internet browsers prefer Mozilla Firefox, Google chrome and Internet Explorer. It is too worth noting that many people prefer Mozilla Firefox and thus for that reason, configuration of local host was done with Mozilla Firefox. Mozilla is fast and easy to use unlike the other two browsers. The other measure taken included the creation of folders on the local host, where users project/pictures would be stored after uploading. The system does not contain heavy scripts that can be rejected by some browsers and thus makes it comfortable for users to use the system in any of the above mentioned browsers or any other browser like Netscape although it is not tested using Netscape.

## 5.3 Objectives Of Software Testing :

The software testing is usually performed for the following objectives:-

**Software Quality Improvement:-**

The computer and the software are mainly used for complex and critical applications and a bug or fault in software causes severe losses. So a great consideration is required for checking for quality of software.

**Verification And Validation:-**

**Verification** means to test that we are building the product in right way .i.e. are

we using the correct procedure for the development of software so that it can meet

the user requirements.

**Validation** means to check whether we are building the right product or not.

**Software Reliabilty Estimation:-**

The objective is to discover the residual designing errors before delivery to the customer. The failure data during process are taken down in order to estimate the software reliability.

## 5.4. Principles Of Software Testing :

Software testing is an extremely creative and challenging task. Some important principles

of software testing are as given:-

* + - All tests should be traceable to customer requirements.
    - Testing time and resources should be limited i.e. avoid redundant testing. It is impossible to test everything.
    - Use effective resources to test.
    - Test should be planned long before testing begins i.e. after requirement phase. Test for invalid and unexpected input conditions as well as valid conditions. Testing should begin in “in the small” and progress towards testing “in the large”. For the most effective testing should be conducted by an independent party.
    - Keep software static (without change mean while) during test.
    - Document test cases and test results.
    - Examining what the software not doing which it expected to do and also checking what it is doing that was not expected to do.

**STRATEGY FOR SOFTWARE TESTING**

Different levels of testing are used in the test process; each level of testing aims to test different aspects of the system.

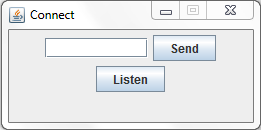
* + The first level is **unit testing**. In this testing, individual components are tested to

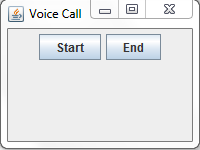
ensure that they operate correctly. It focuses on verification efforts.

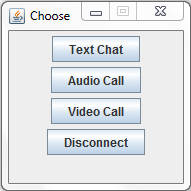
The second level is **integration testing**. It is a systematic technique for constructing the program structure. In this testing, many tested modules are combined into the subsystem which are then tested. The good here is to see if the modules can be integrated properly.

* + Third level is **integration testing**. System testing is actually a series of different tests .

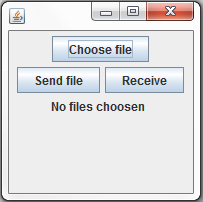
**SCREEN SHOTS**

****

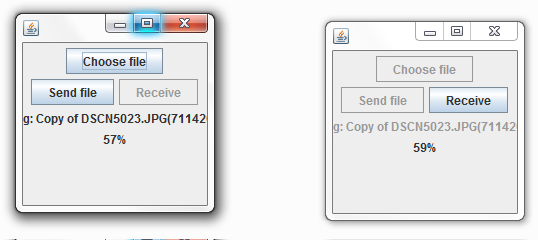
****

****

# 



# 



# 7. CONCLUSION

In the beginning, the task was to develop a Zippy connect software which would enable students or in corporate sector to take connection and easily contact between them without any cost and internet through system. In the end of the development phase, the system was able to achieve everything. The user details are well saved in a java database which is a better record keeping mechanism than the currently used physical files.

In the long run, the system was able to support the above mentioned services to the user. It enabled provision of different platforms for a normal user and an administrator.

# 8. FUTURE ENHANCEMENT(S)

Since the system is dynamic, further enhancements can be incorporated into the system. First, a mobile apps which enables students or employee to get real time notifications of this software. Also, a mail server to enable users register with the correct valid emails so that when they register, a link can be send to their emails so that they mail click it to activate their registration. This will also enable users to receive updates from the site and any other subscriptions like newsletters and news.

# BIBLIOGRAPHY:

# APPENDIX

### REFERENCES:

1. <http://www.absoluteastronomy.com/encyclopedia>
2. **Socket Programming** www.devshed.comPHP
3. **SOMMERVILLE**, IAN. 7th Edition 2004. Software Engineering. London: Pearson Education Limited.
4. Software Engineering: Effective Teaching and Learning Approaches and Practices Heidi J. C. Ellis

Trinity College, USA Steven A. Demurjian University of Connecticut, USA

1. Fundamentals of Software Engineering by Rajib Mall http://www.phindia.com