A web-based distributed system for classification, storage and analysis of human electrogastrograms

Zhao, Zhiqiang (Simon)

http://hdl.handle.net/1880/42896
master thesis

University of Calgary graduate students retain copyright ownership and moral rights for their thesis. You may use this material in any way that is permitted by the Copyright Act or through licensing that has been assigned to the document. For uses that are not allowable under copyright legislation or licensing, you are required to seek permission.

Downloaded from PRISM: https://prism.ucalgary.ca
A WEB-BASED DISTRIBUTED SYSTEM FOR CLASSIFICATION, STORAGE AND ANALYSIS OF HUMAN ELECTROGASTROGRAMS

by

Zhiqiang (Simon) Zhao

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

CALGARY, ALBERTA
SEPTEMBER, 2002

© Zhiqiang (Simon) Zhao 2002
ABSTRACT

The Internet has been used as a major information resource in the area of telemedicine. Using the Internet technology, we have developed a distributed telemedical digital signal processing and database application in the area of electrogastrography (EGG). The aim of this paper is to present an Internet-based system for storage, analysis and classification of human EGG. The system was implemented using Java programming language and Client/Server technology. Advanced digital signal processing algorithms were also incorporated. The system allows the user to submit and classify their own EGG records and view on the Internet the plots and the statistical results stemming from the analysis. Classified abnormal EGG patterns that are associated with pre-diagnosed gastric disorders are stored in an open-architecture EGG database. Thus, growing worldwide evidence for the clinical utility of EGG could be collected, maintained and expanded. The study attempts to quantitatively relate abnormal EGG patterns to various clinical conditions.
ACKNOWLEDGEMENTS

I wish to express my appreciation to my supervisor Dr. Martin P. Mintchev for his guidance, support and valuable suggestions throughout the course of this research.

I would like to thank my wife, Katherine Li for her support during my graduate study. Without her support the project would not have been successfully completed.

The research presented in this thesis was supported in part by the Department of Electrical and Computer Engineering, University of Calgary, and the Alberta Software Engineering Research Consortium.
# TABLE OF CONTENTS

Approval page................................................................. ii  
Abstract................................................................. iii  
Acknowledgements....................................................... iv  
Table of Contents....................................................... v  
List of Tables............................................................ x  
List of Figures............................................................ xi  
List of Symbols, Nomenclature and Abbreviations.................... xiv

CHAPTER ONE: INTRODUCTION.................................................. 1  
1.1. Electrogastrography..................................................... 1  
1.2. Overview of web-based telemedicine .................................. 2  
1.2.1. What is telemedicine?............................................... 2  
1.2.2. Web-based telemedicine.......................................... 3  
1.2.3. Contemporary state-of-the art in telemedicine.................... 4  
1.3. Aims of the study..................................................... 6  
1.4. Structure of the thesis............................................... 6

CHAPTER TWO: EGG SIGNAL ACQUISITION AND ANALYSIS............. 8  
2.1. EGG data acquisition ................................................. 8  
2.2. EGG signal analysis.................................................. 9  
2.2.1. Fast Fourier / Fast Hartley Transform.......................... 10  
2.2.2. Time-Frequency plots.......................................... 13  
2.2.3. Statistical evaluation of the Time-Frequency points............. 15  
2.3. Criteria for normal EGG pattern, and abnormal EGG patterns .... 18  
2.3.1. Normal EGG......................................................... 18
CHAPTER THREE: WEB-BASED DISTRIBUTED SYSTEMS

3.1. Introduction to distributed object system

3.1.1. What is distributed computing?

3.1.2. Why do we need to distribute computing?

3.2. Object-oriented technology

3.2.1. Object-oriented analysis and design

3.2.2. Object-oriented programming

3.3. Distributed Technology

3.3.1. Client/Server distributed architecture

3.3.1.1. Two-tier architecture

3.3.1.2. Three-tier architecture

3.3.2. Communication in a distributed object system

3.3.2.1. Features of a distributed system

3.3.2.2. Common Object Request Broker Architecture (CORBA)

3.3.2.3. Remote Method Invocation (RMI)

3.3.3 CORBA vs. RMI

3.4 Web technology

3.4.1 Fundamentals of the web technology

3.4.2 Java Applet graphical user interface

3.4.3 Servlet technology
3.5. Principles of Relational Database Design ...................................................... 42

3.5.1. Entity-Relationship data model ................................................................. 43

3.5.2. Relational data model .............................................................................. 45

3.5.3. Relational database design using ER-Relational mapping ....................... 45

3.6. Connecting to a database ........................................................................... 46

3.6.1. Java Database Connectivity .................................................................. 46

3.6.2. Open Database Connectivity ................................................................. 48

3.7. Conclusion .................................................................................................. 49

CHAPTER FOUR: SYSTEM DESIGN AND IMPLEMENTATION .......................... 50

4.1. System requirements analysis ................................................................. 50

4.2. System architecture ................................................................................... 53

4.3. Design and implementation of user interfaces ........................................ 55

4.3.1. Java Applet ........................................................................................... 55

4.3.1.1. Getting the data from the database ..................................................... 55

4.3.1.2. Statistical evaluation ....................................................................... 55

4.3.1.3. Rendering ......................................................................................... 56

4.3.2. HTML forms ......................................................................................... 59

4.4. Middle tier ............................................................................................... 60

4.4.1 Design and implementation of the RMI remote objects ......................... 60

4.4.1.1. Using factory design pattern ............................................................. 61

4.4.1.2. Using the replicated object design pattern ....................................... 64

4.4.2. Design and implementation of servlets ............................................... 66
4.4.2.1. Extracting the data from the uploaded file........................................ 66
4.4.2.2. Calculating T-F values from the submitted spectral values................. 68
4.4.2.3. Classification of the calculated T-F values into different EGG patterns. 69
4.4.2.4. HTTP tunneling with servlets .................................................. 71
4.5. Modeling and designing the EGG database.............................................. 72
4.6. Web server deployment ..................................................................... 74
4.7 System testing ................................................................................. 75
  4.7.1. Compatibility testing ................................................................. 75
  4.7.2. Function testing .................................................................. 76
  4.7.3. Regression testing ................................................................. 76
  4.7.4. Performance testing ............................................................... 76
CHAPTER FIVE: RESULTS........................................................................... 78
  5.1. System overview .................................................................... 78
  5.2. Observers ............................................................................. 79
  5.3. Active participants .................................................................. 79
  5.4. Study of classified normal and abnormal EGG patterns ..................... 81
    5.4.1. Normal electrical activity – Normal EGG .............................. 81
    5.4.2. Abnormal electrical activity – Pattern 1 ............................ 83
    5.4.3. Abnormal electrical activity – Pattern 2 ............................ 85
    5.4.4. Abnormal electrical activity – Pattern 4 ............................ 87
    5.4.5. Abnormal electrical activity – Pattern 5 ............................ 89
5.5. Classification of EGG recordings with pre-diagnosed GI disorders ............... 91

5.6. Preliminary evaluation of sensitivity and specificity of EGG based on the data
       extracted from the Tele-EGG system.........................................................92

5.7. Additional limitations and features......................................................... 95

5.8. Conclusions.............................................................................................. 96

REFERENCES .................................................................................................... 98

APPENDIX ........................................................................................................ 105
LIST OF TABLES

Table 1. Differences between CORBA and RMI ......................................................... 39

Table 2. Correspondence between ER and relational models. ....................................... 46

Table 3. Components of Java Three-Tier architecture .................................................... 49

Table 4. Disorder-based sensitivity obtained after categorizing the 191 submitted records. Statistically significant number of cases per disorder are denoted with bold .................................................................................................................. 93
LIST OF FIGURES

Figure 1. Standard electrode positions and recording parameters for electrogastrographic recordings ................................................................. 9

Figure 2. PDF functions ............................................................................. 17

Figure 3. Two-tier architecture ................................................................. 29

Figure 4. A 3-tier architecture ................................................................. 30

Figure 5. General architecture of a distributed object systems ............... 32

Figure 6. The stub, the ORB, and the skeleton ......................................... 34

Figure 7. Message sequence chart between RMI objects ......................... 36

Figure 8. Structure of the RMI system .................................................... 37

Figure 9. Servlets technology architecture .............................................. 42

Figure 10. JDBC architecture ................................................................. 47

Figure 11. User diagram for modeling system requirements ................... 52

Figure 12. Three-tier system architecture ............................................... 53

Figure 13. EGG data flow diagram ........................................................ 54

Figure 14. Block diagram for calculating the Gastric averages of the submitted T-F points .................................................................................... 56

Figure 15. User interface graphical components ..................................... 56

Figure 16. Sequence diagram of use case of plotting T-F points into a T-F plot ................................................................. 58

Figure 17. Sequence diagram of use case of plotting PDF ....................... 58
Figure 18. Class diagram of remote objects ............................................................... 60

Figure 19. Sequence diagram of the use case of retrieving T-F data value ............. 61

Figure 20. Factory Design Pattern for RMI ............................................................... 62

Figure 21. Modified class diagram utilizing the Replicated Design Pattern .......... 65

Figure 22. HTML client and servlet object interaction diagram ............................ 67

Figure 23. Block diagram for calculating the T-F values of a single EGG channel .... 69

Figure 24. Classification algorithm flow chart ......................................................... 70

Figure 25. Entity-Relationship diagram ................................................................. 72

Figure 26. The relational database scheme, which is generated from the mapping of the ER diagram ................................................................. 73

Figure 27. Time-Frequency plot of normal EGG with frequency resolution of 0.23 cpm, time intervals of 4.27 minutes and 0% overlap ................................. 81

Figure 28. Basic averages obtained from a volunteer with normal gastric electrical activity ................................................................. 82

Figure 29. Probability Density Function plots with frequency resolution of 0.23 cpm in the frequency range 1-9 cpm. The maxima coincide at 3.05 cpm .......... 82

Figure 30. Time-Frequency plot recorded from a patient with abnormal GEA (Pattern 1). ........................................................................................................ 83

Figure 31. Basic averages obtained from a patient with abnormal GEA (pattern 1) .... 84

Figure 32. Probability Density Function plots of a patient with abnormal EGG (pattern 1). The maxima coincide at 2.78 cpm ....................................................... 84

Figure 33. Unstable Time-Frequency plots obtained from a patient with abnormal
EGG (Pattern 2)................................................................................................. 85

**Figure 34.** Basic averages calculated from a patient with abnormal EGG (Pattern 2) . . . 86

**Figure 35.** Probability Density Function plots (Pattern 2). The maxima do not
coincide. .................................................................................................................. 86

**Figure 36.** Time-Frequency plots recorded from a patient with abnormal
EGG (Pattern 4) ...................................................................................................... 87

**Figure 37.** Basic averages calculated from a patient with abnormal EGG (Pattern 4).
Five out of the 8 channels are stable. ...................................................................... 88

**Figure 38.** Probability density functions calculated from a patient with abnormal
EGG (Pattern 4) Note that the maxima coincide at 3.95 cpm , which is
outside the normal frequency range. ...................................................................... 88

**Figure 39.** Time-frequency plots acquired from a patient with abnormal
EGG (Pattern 5) ...................................................................................................... 89

**Figure 40.** Basic averages calculated from a patient with abnormal EGG (Pattern 5) . . . 90

**Figure 41.** Probability density functions calculated from a patient with abnormal
EGG (Pattern 5). Seven EGG channels are unstable. The maxima coincide,
but outside the normal range. .................................................................................. 90

**Figure 42.** Summary of classified EGG records. ..................................................... 92
**LIST OF SYMBOLS, NOMENCLATURE AND ABBREVIATIONS**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EGG-</td>
<td>Electrogastrography</td>
</tr>
<tr>
<td>GEA-</td>
<td>gastric electrical activity</td>
</tr>
<tr>
<td>HTTP-</td>
<td>HyperText Transfer Protocol</td>
</tr>
<tr>
<td>FFT-</td>
<td>Fast Fourier Transform</td>
</tr>
<tr>
<td>FHT-</td>
<td>Fast Hartley Transform</td>
</tr>
<tr>
<td>API-</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>RMI-</td>
<td>Remote Method Invocation</td>
</tr>
<tr>
<td>JDBC-</td>
<td>Java Database Connectivity</td>
</tr>
<tr>
<td>ODBC-</td>
<td>Open Database Connectivity</td>
</tr>
<tr>
<td>T-F-</td>
<td>Time-Frequency</td>
</tr>
<tr>
<td>JVM-</td>
<td>Java Virtual Machine</td>
</tr>
<tr>
<td>SQL-</td>
<td>Structure Query Language</td>
</tr>
<tr>
<td>PDF-</td>
<td>Probability Density Function</td>
</tr>
<tr>
<td>PSTN-</td>
<td>Public Switched Telephone Network</td>
</tr>
<tr>
<td>ADSL-</td>
<td>Asymmetric Digital Subscriber Line</td>
</tr>
<tr>
<td>ISDL-</td>
<td>Integrated Services Digital Network</td>
</tr>
<tr>
<td>WWW-</td>
<td>World Wide Web</td>
</tr>
<tr>
<td>cpm-</td>
<td>cycles per minute</td>
</tr>
<tr>
<td>DFT-</td>
<td>Discrete Fourier Transform</td>
</tr>
<tr>
<td>DHT-</td>
<td>Discrete Hartley Transform</td>
</tr>
</tbody>
</table>
CHAPTER ONE: INTRODUCTION

1.1. Electrogastrography

The main functions of the stomach are mixing, grinding and transporting consumed food. For the normal performance of these functions, adequate synchronized gastric motility is required. Gastric motility is largely controlled by the spontaneous electrical activity of gastric smooth muscles [1,2,3]. Electrogastrography (EGG) represents the non-invasive cutaneous recordings of gastric electrical activity (GEA) [1]. This technique could become a valuable clinical tool for recognizing gastric electrical abnormalities [2]. By placing electrodes on the abdominal wall and transforming the internal GEA signals into a weak and very close to sinusoidal signal, these recordings can potentially be used for recognizing normal and abnormal gastric motility [1, 3].

Although electrogastrographic signals have been recorded since 1921, the lack of objective methods for their interpretation impeded their clinical applicability. Recent computer-based quantitative methods for evaluation of EGG renewed the practical promise of this technique [4]. Unfortunately, its diagnostic value is still in question. At present, no standardized way for collecting and classifying EGG records has been suggested, and a systematic and sufficiently extensive database of classified signals relating EGG abnormalities to specific clinical disorders is lacking [4].
1.2. Overview of web-based telemedicine

1.2.1. What is telemedicine?

With the continuing advances in information technology, the applications of computers and telecommunication technology in medicine are growing rapidly. The modern information technology has greatly affected various aspects in medicine including delivery of health care, physician-patient relationship, tele-consulting and distance education. In the early 1990s, two technological developments brought about emerging of telemedicine [5]. The first of these developments was the increasing utilization of high-speed, high-bandwidth telecommunication systems around the world. The second was the invention of digital devices capable of capturing and transmitting image data and other digital data. Recently, the Internet, with its broad accessibility and scalability, has become a major medical information resource and a new platform for telemedicine [6].

Generally, telemedicine, a various collection of technologies and clinical applications, can be defined as the remote delivery of health care and sharing of medical knowledge using telecommunication systems. A standing committee of the Institute of Medicine at George Washington University gives formal definition of telemedicine as “the use of electronic information and communication technologies to provide and support health care when distance separates the participants” [7]. It involves the integration of advanced information technologies, human–machine interface and healthcare and medical technology [7]. Telemedicine usually has three basic common aspects: (1) information supplements used to support medical decision making; (2) signal
acquisition and processing; and (3) processes between individuals and institutions to enable physicians to practice medicine at a distance [5].

Telemedicine combines a lot of technologies including telephone, radio, facsimile, modem and video. It may involve real time process such as interactive video, or asynchronous protocols for the delivery of text or graphic data, audio information, still images, short video clips and full-motion video [7].

1.2.2. Web-based telemedicine

In the last few years, the World Wide Web (WWW) has become the most accessible and utilizable computer-based information resource in hospitals and clinical institutions in general [5]. Although WWW is typically used as an infrastructure for publishing and retrieving information on the Internet, its world-wide acceptance has been a major motivation for software developers to use it to create standard front-ends in client/server applications. With the continuing advances in WWW technology, great opportunities have been provided for developing a wide range of distributed web applications for accessing various public health systems using the Internet. Thus, WWW has become the infrastructure for providing access to complex applications from virtually any machine and operating system, allowing users to communicate with each other using a common application server [8]. The uniqueness of web-based applications is related to the fact that both the information and the application are logically stored in a centralized location that might be distributed over the network using several servers which act as resource providers. This delivers a standardized technical solution for telemedicine applications corresponding to their distributed nature [8]. There are enormous advantages in using a standardized communication platform for both patient and physician, naturally
leading to web-based telemedicine applications. Thus, in medical and clinical research, web technologies have been widely utilized to create a broad range of Internet applications, generally known as web-based telemedicine [9,10,11]. In such applications, Internet has been extensively used to convey medical data and images between patients and medical professionals for the purposes of research, scientific investigation and even tele-diagnosis regardless of the distance that separates the parties involved [10]. Today the use of WWW for telemedicine applications has became almost compulsory.

1.2.3. Contemporary state-of-the art in telemedicine

In the last few years, many application projects focused on telemedicine were launched around the world [5,6,8]. In these projects, telemedicine provides medical care to underserved populations, who reside in rural and geographically isolated areas and allows physicians to consult medical experts and specialists; it permits physicians to share patient records and diagnostic images in clinical research, despite geographic separation; it also makes an improvement of medical education for rural healthcare professionals, linking community hospitals with sponsoring medical center [5,6,8,9].

The interest for telemedicine, especially for web-based telemedicine, has been widely demonstrated in a lot of other recent publications and on the Internet. Now all around the world, a variety of telemedical and telemedicare systems have been developed based on the Internet and web technologies [10,11,12, 13, 14].

One of the first published web-based telemedicine application, Home Asthma Telemonitoring system, has been published in 1998 [12]. In this application, the system allows patients who suffer from asthma to record the data at home with a portable spyrometer connected to a laptop. The recorded data are then sent to the hospital through
Public Switched Telephone Network (PSTN). The physician can view and check the data using WWW.

Another important web-based telemedicine application in the field of cardiology is the Heart Care System [13]. The system is a computerized cardiac recovery service which is designed to supply home-care support for patients in the first three months after cardiac artery by-pass graft surgery. Web-TV is used for the patients to access to the system service from home. The web pages have also been designed to compensate the limited visualization capabilities of Web-TV.

Recently [14], a web-based telemedicine system for electrocardiographic (ECG) monitoring in the home was proposed, which was completely implemented using web-based technology. This application utilizes ActiveX controls and Microsoft certificates (a specialized software for data downloading) via WWW at the client side. This solution enables the users to work with a complete, Internet-based transparent system.

In Korea, a web-based picture-archiving and communication system was developed for transferring medical images and enabling remote collaboration between two or more client computers located anywhere on the Internet [10].

In Japan, a remote collaboration system for telemedicine using the Internet has been used to facilitate consultation with radiological experts [11].

In summary, web-based telemedicine has played an important role in medical research. Since no particular networking technologies are required, such applications can be easily accessed by all stakeholders through common web browsers rather than utilizing specialized tools. Moreover, Internet connectivity provides an affordable and popular communication channels around the world. Thus, users can visit web-based
telemedicine systems independently of time and place, as long as Internet connections are available.

1.3. Aims of the study

EGG is a non-invasive and relatively inexpensive technique to assess GEA. Although Electrogastrography show its potentially promising in recent medical research, it is often misunderstood and misinterpreted because of its clinical infancy [15]. At the present, EGG is not a reliable tool for reliably identifying gastric disorders, but it could be a valuable test for determining frequency instability and electrical uncoupling in the stomach. Further progress in EGG quantification and interpretation could improve the capabilities of this non-invasive technique. Thus, it is very important to popularize the methodology and the interpretation of EGG. Telemedicine could be a powerful, but not presently utilized resource in this direction. The objective of this work is to provide a comprehensive, easy-to-use and worldwide accessible distributed system for registered submission, classification and storage of human electrogastrograms based on Internet and web technology. Such system could offer clinicians and researchers in the area of EGG adequate and inexpensive training in modern EGG methodology and interpretation over the Internet. In addition, this system could provide preliminary evaluation of sensitivity and specificity of EGG for diagnosing various gastric motility disorders.

1.4. Structure of the thesis

The thesis is structured into five chapters.

Chapter 2 introduce the methods of EGG signal acquisition and analysis. Then the quantitative criteria for normal EGG pattern, and abnormal EGG patterns are outlined.
Distributed object technology is discussed in Chapter 3. In this chapter, Object-oriented technology, distributed technology and web technology are described in detail. In addition, database model technique is also introduced.

Chapter 4 describes the design and development of Tele-EGG system which includes requirement analysis using UML, system architecture, component modeling and system deployment.

Finally, Chapter 5 present the study results of classified normal and abnormal EGG patterns using this system, and conclusion of the work.
CHAPTER TWO: EGG SIGNAL ACQUISITION AND ANALYSIS

2.1. EGG data acquisition

Gastric electrical activity can be recorded using different techniques [1]. The recordings can be made from an intact stomach using electrode wires implanted into the serosa or the mucosa of the stomach, a technique known as gastric electrical activity (GEA) recording, or by placing electrodes on the anterior abdominal wall, known as the cutaneous recordings of GEA, or electrogastrography (EGG). Internal GEA is obtained using invasive and uncomfortable methods, and is used exclusively for research purposes [1]. EGG represents a non-invasive procedure for indirect, inexpensive assessment using abdominal electrodes [1,2,3,16]. During EGG recordings, the internal gastric electrical signals are transformed by the electrical conductivity of the body [1], changing the internal GEA signal into a low-amplitude and very close to a sine wave-shape signal with a frequency of repetition of approximately 0.05 Hz (3 cycles-per-minute, cpm). Along with gastric electrical signals noise artifacts from various external sources are also recorded [1].

Experimentally, EGGs are usually acquired for 1 hour in fasting state and 1 hour after a standard 500 Kcal meal (52% carbohydrate, 19% protein and 29% fat)[1]. The signals are sampled with 10 Hz sampling frequency and filtered with specially designed EGG bandpass amplifiers. After the acquisition, digital filtering in the frequency band of 0.02-0.15 Hz is performed. The sampling frequency is reduced to 2 Hz after the digital filtering to simplify the data processing procedures [15].

Figure 1 depicts standard electrode positions used for multichannel EGG
recordings in our research. Five electrodes are positioned on the abdominal projection of the gastric axis. Various combinations between these five electrodes form 8 standard EGG channels [17].

GAIN RANGE: 0.002 mV/cm

Channel Combinations
1. 1-2
2. 2-3
3. 3-4
4. 4-5
5. 1-3
6. 2-4
7. 1-4
8. 1-5

Filters:
1. High-Pass Filter: Time Constant = 5s (All Channels)
2. Low-Pass Filter: Cut-Off Freq.=0.08 Hz (All Channels)

Figure 1. Standard electrode positions and recording parameters for electrogastrographic recordings.

2.2. EGG signal analysis

Among the four EGG recording parameters (amplitude, frequency, time shifts between different channels and waveform), frequency is found to be the only quantity suitable for reliable evaluation of EGG [2,4]. In order to examine the EGG frequency and its dynamics, the time-frequency approach based on spectral analysis has been widely
utilized among different methods for quantification and statistical analysis of EGG [1,4,17,18,19,20,21]. Compared to other approaches, the time-frequency technique can make EGG analysis more quantitative by examining the dynamics of the dominant spectral peaks in the frequency domain, thus removing most of the noise-associated uncertainties, typical for these recordings [1]. The methods to assess EGG frequency include fast Fourier or fast Hartley transform (FFT or FHT), spectral plots and statistical evaluation of these plots.

2.2.1. Fast Fourier / Fast Hartley Transform

The acquired EGG recordings are time-domain signals. They have to be transformed to frequency-domain using mathematical transformations. Fourier transform is a commonly used transformation and it is defined by the following pair of integral equations [22]:

\[ F(f) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} V(t) e^{-j2\pi ft} dt \]  \[ 2.10a \]

\[ V(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} F(f) e^{j2\pi ft} df \]  \[ 2.10b \]

where \( V(t) \) is the signal function in the time domain, and \( F(f) \) is its image function in the frequency domain. Equation [2.2.a] transforms signals that are functions of time to functions of frequency. Equation [2.2.b] converts signals that are functions of frequency into functions of time. \( F(f) \) can also be defined in a complex function [22]:
\[ F(f) = |F(f)| e^{i\phi(f)} \]  \[2.11\]

where \(|F(f)|\) is the magnitude spectrum of \(V(t)\), and \(\phi(f)\) is the phase spectrum. The square of the magnitude spectrum is called the power spectrum.

Hartley transform is a real alternative to the complex Fourier transforms \[23\]. The pair of transforms is given by:

\[ H(f) = \frac{1}{2} \int_{-\infty}^{+\infty} V(t) \times \text{cas}(2\pi ft) \, dt \]  \[2.12a\]

\[ V(t) = \frac{1}{2} \int_{-\infty}^{+\infty} H(f) \times \text{cas}(2\pi ft) \, df \]  \[2.12b\]

where \(\text{cas}(x) = \cos(x) + \sin(x)\). Similarly to the Fourier transform, the pair of equations can be used to perform time-to-frequency and frequency-to-time transforms.

The Fourier transform is the even part of the Hartley transform minus \(j\) times the odd part:

\[ E(f) = \frac{1}{2} \left[ H(f) + H(-f) \right] = \int_{-\infty}^{+\infty} V(t) \times \cos(2\pi ft) \, dt \]  \[2.13a\]

\[ O(f) = \frac{1}{2} \left[ H(f) - H(-f) \right] = \int_{-\infty}^{+\infty} V(t) \times \sin(2\pi ft) \, dt \]  \[2.13b\]

Then, it can be easily derived that:
\[ E(f) \cdot j \times O(f) = F(f) \] 

Consequently, power spectra can be obtained from the Hartley transforms using:

\[ |F(f)|^2 = \frac{[H(f)^2 - H(-f)^2]}{2} \]  

However, modern signal analysis involves the processing of discrete signals which are regularly sampled in time. Thus, the discrete Fourier and Hartley transforms (DFT and DHT) are utilized to calculate series coefficients for a discrete periodic signal.

The DFT pair is given with [22]:

\[ F(n) = \frac{1}{N} \sum_{k=0}^{N-1} f(k) \times \exp(-j2\pi nk / N) \quad [2.16a] \]

\[ f(k) = \sum_{n=0}^{N-1} F(n) \times \exp(j2\pi nk / N) \quad [2.17b] \]

where \( k \) is the index in time domain, and \( n \) is the index in frequency domain, \( N \) is the number of the sampled points (transform length).

The DHT pair is given by [23]:

\[ H(n) = \frac{1}{N} \sum_{k=0}^{N-1} f(k) \times \text{cas}(2\pi nk / N) \quad [2.18a] \]

\[ f(k) = \sum_{n=0}^{N-1} H(n) \times \text{cas}(2\pi nk / N) \quad [2.18b] \]

The calculation of DFT (or DHT) involves heavy multiplication and takes a large amount of computer time. In order to minimize the calculation time, fast algorithms for DFT or DHT have been utilized. These algorithms are known as the fast Fourier
transform (FFT) [22,24] and the fast Hartley transform (FHT) [23].

There are two basic groups of fast algorithms which are known as decimation in frequency and decimation in time [24]. The first group of fast algorithms decomposes the original signal into sections of sequential samples, until a number of two-sample signals are produced. Thus, the calculation of the discrete transform of a two-sample signal becomes very fast because only additions and subtractions are performed. The transform of a two-sample signal can be easily given with:

\[
F(0) = f(0) + f(1)
\]
\[
F(1) = f(0) - f(1)
\]  
[2.19]

The decomposition of the DFT into N/2 two-sample DFTs requires that the length N be a power of 2.

The second main group of algorithms is based on an alternative decimation scheme known as decimation in time. In this case the original data sequence length is divided successively by two by selecting alternate samples in each successive sequence. The end result of this decomposition is again N/2 pairs of samples, which are then transformed using equation [2.19].

2.2.2. Time-Frequency plots

The frequency spectrum of a time segment taken from the analyzed signal represents the distribution of frequencies present in this particular segment. This time segment is also known as time window. The length N of the transform is related to the spectral resolution. The length of the time window determines the smoothness of the spectrum. It is not necessary to choose a length of transform that is much longer than the
length of the time window, since the missing portion of the signal has to be padded with zeros. On the contrary, if the time window is very big with respect to the period of the signal and this period changes for a small amount of time within this huge time window, this change possibly cannot be observed clearly, especially if other noise signals are also present. Thus, a desire to make N as big as possible (to increase the spectral resolution) is restricted by the actual length of the time window [25].

It has been widely accepted that EGG signals are analyzed using 512-point fast Fourier or fast Hartley transform (FFT or FHT) and signals are sampled with a sampling frequency of 2 Hz [3,18]. Thus the frequency transforms are to be performed on a time window with a length of 256 seconds (4.27 minute). In this case, the frequency resolution is 0.003906 Hz or 0.234 cycles-per-minute (cpm). As a result, spectra from successive 4.27 min intervals should be acquired so that the dynamics of the dominant frequency components in the whole recording can be followed. Once the successive time intervals are processed, the obtained EGG spectra can be arranged one behind another with a small horizontal shift to form a “three-dimensional” (3D magnitude/power-frequency-time) plots which were initially proposed by Kingma et al.[26]. This approach was later used by Van Der Schee et al. [27,28] for spectral analysis and became known as running spectrum analysis. In order to follow the frequency dynamics of the EGG signals better, it is also possible to overlap the successive time intervals, e.g. if the first spectrum is obtained for the time interval (0-255 sec), the second could be calculated in the interval (128-383 sec) instead of (256-511 sec). This introduced a 50% overlap between the successive time windows. The number of the spectra in the 3D plots is thus increased[15].
Two ways of presenting successive spectra in the 3D plot have been utilized: (1) each spectrum is separately normalized to 100% in a given frequency window; (2) the absolute maximum of all spectra is found and normalized to 100%. If only the frequency is analyzed, the first method is preferable if huge differences in the signal amplitudes are present [15].

Because of the individual normalization of each spectrum, one of the dimensions in the 3D plots, the magnitude, can be completely eliminated and the maxima of the dominant peaks from each spectrum can be arranged in a new two dimensional plot versus time and connected with lines. This new two-dimensional plot is known as the time-frequency (T-F) plot [4]. It can be considered as two-dimensional representation of the standard three-dimensional plot associated with running spectrum analysis [17]. It is quite obvious that each time-frequency plot is built by points, each of which represents the frequency of the dominant peak in the corresponding spectrum. When overlap is introduced, the interval $T_f$ between any two successive points in the time-frequency plot is given with [15]:

$$T_f = \left[ \frac{N}{Fs} \right] \times \frac{(100 - OVL)}{100}$$  \hspace{1cm} [2.21]

where $Fs$ is the sampling frequency, $OVL$ is the percentage of overlap introduced, and $N$ is the number of points of the Fourier transform.

**2.2.3. Statistical evaluation of the Time-Frequency points**

When a signal with duration $Ds$ is sampled with interval $Ts$, the signal interval $\Delta t$ transformed into frequency domain can be calculated by:
\[ \Delta t = N \times Ts \]  \hspace{1cm} [2.31]

where \( N \) is the number of points of the transform. For each signal interval \( \Delta t \), only one dominant frequency peak is obtained, to which corresponds a single time-frequency point. Thus, the number of points in the time-frequency plot \( N_{tf} \) is given with [15]:

\[
N_{tf} = (\text{int}) \left[ \frac{D_s}{(N \times Ts)} \right] / \left( 1 - \frac{\text{OVL}}{100} \right) 
\]

, Or

\[
N_{tf} = (\text{int}) \left[ \frac{D_s \times Fs}{N} \right] / \left( 1 - \frac{\text{OVL}}{100} \right) 
\]  \hspace{1cm} [2.32]

where OVL is the percentage overlap between successive time intervals and Fs is the sampling frequency. The points are later arranged versus time and connected with lines to form the actual time-frequency plot. These points can be utilized for various methods of simple statistical evaluation, which include mean value, variance and standard deviation, as well as the Root Mean Square (RMS) value.

Mean value \( M_n \) is defined as [29]:

\[
M_n = \frac{\sum F_i}{N_{tf}}, \text{ cpm} 
\]  \hspace{1cm} [2.33]

where \( F_i \) are the frequency values of the points in cycles per minute (cpm).

The variance \( V_r \) can be calculated as [29]:

\[
V_r = \frac{\sum (F_i - M_n)^2}{N_{tf}} = \frac{\sum F_i^2}{N_{tf}} - M_n^2, \text{ cpm}^2 
\]  \hspace{1cm} [2.34]
The standard deviation $SD$ is the square root of the variance and when assessing time-frequency plots is measured in cycles per minute (cpm). The SD can be obtained as:

$$SD = \sqrt{\sum (Fi - Mn)^2 / N_{if}, cpm}$$  \[2.35\]

The Root Mean Square (RMS) value can be obtained by:

$$RMS = \sqrt{\left(\sum (Fi)^2\right) / N, cpm}$$  \[2.36\]

The Probability density function (pdf) is another tool for evaluating the time-frequency points [29]. This function measures the probability (in %) of one unknown process or function to have a given value [15]. The pdf of the T-F points obtained from a sine wave with fixed frequency can be present as a single line of 100% at that frequency. However, if the frequency of the sine wave changes with the time, many different frequencies would be present, and the pdf would become a curve. Connecting the number of occurrences at different frequency gives a graphical representation to the pdf (Figure 2.).

Figure 2. PDF functions
\( f_a \) and \( f_b \) are the frequencies of two different sin wave. Formally, the probability, \( Pr \), for the frequency of the sine wave \( f_x \) to belong to a period of frequency \( f_a \)-\( f_b \) (\( f_b > f_a \), see Figure 2.) can be given with [15]:

\[
Pr(f_a < f_x < f_b) = \frac{S_{ab}}{S_{total}}
\] [2.37]

where \( S_{ab} \) is the area between the frequencies \( f_a \) and \( f_b \), and \( S_{total} \) is the total area under the curve.

When processing digital signals it is convenient to represent the pdf curve as a histogram. Each horizontal step in this histogram is equal to the frequency resolution of the time-frequency plot, while vertically the percentage of the points associated with the given frequency is shown. The frequency resolution \( Fr \) of the digital time-frequency plot is given with [15]:

\[
Fr = \frac{F_s}{N}
\] [2.38]

Where \( F_s \) is the sampling frequency and \( N \) is the number of points of the transform.

### 2.3. Criteria for normal EGG pattern, and abnormal EGG patterns

In order to establish differences between normal and abnormal EGG patterns, a study of several groups of patients was performed [15]. Based on statistical assessment of the T-F points extracted from the EGG recordings of these patients, important criteria for recognizing normal EGG, and six abnormal EGG patterns were derived [15]. The criteria can be summarized as follows:

#### 2.3.1. Normal EGG:

At least 3 out of the 8 standard EGG channels have mean values of the T-F points
in the range of 2.5-3.75 cpm; the standard deviation of the T-F points is <0.45 cpm, and the PDFs of the T-F points exhibit single-peak distribution with coinciding maxima.

2.3.2. Abnormal Patterns:

Pattern 1.

Irregular gastric electrical activity with normal electrical coupling of different parts of the stomach. The standard deviation of more than 6 out of 8 standard EGG channels should be >0.45 cpm. The maxima of the PDFs of at least 3 out of the 8 channels must coincide in the normal range of 2.5-3.75 cpm;

Pattern 2.

Irregular gastric electrical activity combined with electrical uncoupling of different parts of the stomach. The standard deviation of more than 6 out of 8 standard EGG channels should be >0.45 cpm. The maxima of the PDFs of more than 6 out of the 8 standard EGG channels do not coincide in the normal frequency range of 2.5-3.75 cpm;

Pattern 3.

Regular gastric electrical activity in the normal frequency range, but abnormal electrical coupling of different parts of the stomach. Gastric electrical frequency is in the normal range and the standard deviation of the T-F points is <0.45 cpm in at least 3 out of 8 EGG channels. The maxima of the PDFs of at least 6 out of the 8 EGG channels do not coincide in the normal frequency range of 2.5-3.75 cpm;

Pattern 4.

Regular gastric electrical activity outside the normal frequency range combined with normal electrical coupling. Gastric electrical frequency is outside the normal range, but the standard deviation of the T-F points is <0.45 cpm in at least 3 out of the 8 EGG
channels. The maxima of the PDFs of at least 3 channels coincide, but outside the normal frequency range of 2.5-3.75 cpm;

**Pattern 5.**

Irregular gastric electrical activity combined with normal electrical coupling, but outside the normal frequency range. The standard deviation of more than 6 out of 8 standard EGG channels is >0.45cpm. The maxima of the PDFs coincide, but outside the normal frequency range of 2.5-3.75 cpm;

**Pattern 6.**

Regular gastric electrical activity outside the normal frequency range and abnormal electrical coupling. Gastric electrical frequency is outside the normal range, but the standard deviation of the T-F points is <0.45 cpm in at least 3 out of the 8 EGG channels. The maxima of the PDFs of at least 6 out of the 8 EGG channels do not coincide.

2.4. **Summary**

Among the 4 EGG recording parameters including amplitude, frequency, time shift between each different channel and waveform, frequency is found to be the only reliable quantity for evaluation of EGG. In order to examine the EGG frequency and its dynamics, quantification and statistical analysis of EGG have to be performed. The methods to assess GEA frequency include frequency analysis using FFT/FHT, Time-Frequency plots and Statistical evaluation of Time-Frequency plots. Based on the quantification and statistical analysis of EGG, the useful quantitative criteria differentiating normal from abnormal GEA can be obtained and assessed.
3.1. Introduction to distributed object system

Distributed computing originated around 1970 with the emergence of two major technologies: minicomputers, then workstations, and later, and computer networks (eventually Ethernet and the Internet)[30]. With the continuing advances of the technology of computing devices, networks and software, distributed computing has become an important part of current information industry. At present, thousands of workstations and small personal computers are connected by various networks working together to accomplish large business and engineering tasks.

3.1.1. What is distributed computing?

As we know, programs are run in certain address space, while an address space is a collection of memory ranges. If a program is executed only in a single address space, we call this local computing [30]. On the contrary, distributed computing means that a program makes calls in other address spaces in addition to its own. The other address spaces might be located on different computers. [30].

3.1.2. Why do we need to distribute computing?

Generally, applications are distributed across multiple computers because of one or more of the following reasons: (1) Some applications must be executed on multiple computers because the data accessed is located on multiple computers; (2) Some applications need to take advantage of multiple processors computing in parallel to
execute some complicated tasks; (3) The users of the application communicate and interact with each other via a piece of the distributed application on his or her computer, and with shared objects [31].

Distributed applications use network protocols and computer operating systems to perform coordinated tasks across the network. The client/server model is the most common form of distributed applications in which one program (the client) communicates with another program (the server) for the purpose of exchanging information [32].

As object-oriented technology has been increasingly utilized in the software industry, distributed object systems have been coming into stage. Combined by object technology, distributed technology and web technology, distributed object systems offer a clear, elegant and scalable paradigm for constructing distributed systems [33]. A distributed object system is a collection of objects that isolate the requesters of services (clients) from the providers of services (servers) by a well-defined encapsulating interface [34]. In a distributed object system, objects can be dispersed across the network and can be accessed by users/applications across the network. An object on one machine can send messages to objects on other machines. The objects can be clients, servers, or both.

The ability to distribute objects across the network is extremely useful because it can greatly simplify the design of large complicated information systems, which allows the benefits of object-oriented design to be leveraged in a distributed environment. The
underlying problem can be closely modeled and the limitations of current networking techniques can be ignored. [34].

3.2. Object-oriented technology

The trend of present information technology is to extend the object-oriented concepts to enterprise wide distributed systems. Since all entities can be viewed and modeled as objects, object technology is the core and backbone of distributed systems [35]. Object-oriented technology involves object-oriented analysis and design, and object-oriented programming.

3.2.1. Object-oriented analysis and design

In the software process, almost anything can be modeled as an object. Object orientation is a technique of modeling a real-world system with software that uses abstraction with objects, encapsulated classes, and communication via messages, class hierarchies, and polymorphism [36]. The conceptual definition of a class can be thought of as a collection of encapsulated data items, and operations on that data using methods. A class is also a description or template of a set of objects [36]. This set of objects share common attributes and common behavior. Objects are instances of a class or of a member of a class, which is created and exists while the program is running. In an object-oriented system, classes can have hierarchical structure [37]. Specialized classes are constructed as we move down the hierarchy. Similarly, classes with more generalized features are observed as we move up the hierarchy. In other words, inheritance relations are used to describe specializations and generalizations of the concepts represented by the classes. Inheritance is a relation between classes that allows definition and implementation of a
class to be generated from another [37]. Polymorphism is another concept of object-oriented systems which means that object references can apply to more than one class, subject to limitations of inheritance hierarchy [37]. Classes are composed of data members and methods which form the attributes of the class. A method in a class is invoked by sending a message to an instance of that class, i.e. to an object. Objects interact by exchanging messages.

Object-oriented analysis includes building a model of the system, and answering the question of what the system will do without concern for how it will do it. The primary product of object-oriented analysis is a conceptual model that expresses the proposed system and the real-world entities with which the system interacts [36]. The conceptual model includes the identification of the relationships and interactions between the entities and the system. Thus, in the conceptual model, the entities have to be identified first. Then the relationships between the entities have to be recognized.

Object-oriented design deals with the design of the internal logic of a program using the results of the analysis, and produces a detailed design of the classes that will provide that internal logic [36].

Numerous methods have been described for the analysis and the design of object-oriented systems. The most significant approach is the unified approach using the Unified Modeling Language (UML), which consists of a common set of models and notations that can be used for object-oriented analysis and design. Generally, UML is used for visualizing, specifying, constructing, and documenting the skeleton of an object-oriented system [38]. Issues relating to the use of UML include defining the appropriate set of
objects, determining the appropriate relationships of objects, defining the appropriate set of operations (object behavior), and determining how to represent them best.

UML is used to model the system by mainly capturing the structural, or static, features of the system and the behavioral, or dynamic, features of the system [38]. The structural model emphasizes the static features of the modeled system, and the behavioral model emphasizes the dynamic features of the modeled system.

The Class Diagram is one of the UML structural modeling diagrams. In UML, the most basic element of a class diagram is a class. Class Diagrams are the most common diagrams found in modeling object-oriented systems. In a Class Diagram, a set of classes, interfaces, and collaborations and their relationships are described, which illustrate the static design view of an object-oriented system.

Use Case Diagrams and Sequence Diagrams belong to UML behavior modeling. A Use Case Diagram illustrates the intended system functions (use case), the system surroundings (actors) and the relationships between the use case and the actors. They provide an overview of the system structure and a starting point for the software design. Use Case Diagrams also assist substantially in classifying the requirement specifications for the object-oriented system.

A Sequence Diagram is an interaction diagram that describes the time ordering of message exchanges between the objects. It usually depicts a set of objects and messages sent and received by these objects.
3.2.2. Object-oriented programming

Object-Oriented (OO) Programming is an improvement in programming technology that works naturally with the object-oriented analysis and design process [37]. There are several objected-oriented programming languages available to choose from, including Smalltalk, C++ and Java.

C++ is a widely used OO programming language. It is compatible with C (it is actually its superset), so that existing C code can be incorporated into C++ programs. C++ programs are fast, efficient and powerful. However, C++ is unfortunately too complex, too error-prone, and difficult to teach and learn because of its wealth of features [39]. Also it is not suitable for distributed web-based applications because it was not designed for the Internet.

Java is a new programming language which provides full support for object-oriented programming. It retains some of the important features of C++ while eliminating most disadvantages [39]. At the same time, Java introduces some new techniques that are superior to C++. One of them is called Java Applet, the rich Graphical User Interface (GUI) that can be displayed by a web browser. This technique makes Java very suitable for coding Internet applications.

The features of Java that are of particular interest in distributed applications are summarized as follows:

1. Java is objected-oriented. As an instance of a class, an object can be thought of as a computing agent. Distributing a system implemented in Java can be thought of as simply distributing its objects over the network, and establishing networked
communication links between them using Java's built-in network support [35].

2. Java is an interpreted language. Code written in Java can be compiled into bytecodes rather than native machine code [40]. These bytecodes run on Java Virtual Machine (JVM), a virtual hardware architecture that is implemented using software running on the "real" machine and its operating system. Java bytecodes are platform-independent because Java programs can run on any operating system that a Java Virtual Machine has been added to. This is very important for applications distributed over the Internet or other networks because any available PC or workstation on the network can host virtually any agent in a distributed system. Once the elements of the system have been defined using Java classes and compiled into Java bytecodes, they can migrate without recompilation to any of the hosts available [40].

3. The Abstract object interface is another valuable feature that Java offers for developing distributed system [41]. If a class is declared as implementing a specified interface, then the class has to implement the methods specified in the interface. The advantage of interfaces lies in that other objects in the system can be implemented to talk to the specified interface without knowing how the interface is actually implemented in a class. If a class has to be moved to a remote host, then the local implementation of the interface can act as a stub, passing calls to the interface over the network to the remote class.

3.3. Distributed technology

Distributed technologies refer to technologies that facilitate the integration of objects in a distributed system [34]. Objects in a distributed system communicate with each other using the distributed technologies such as Remote Method Invocation (RMI)
and Common Objects Request Broker Architecture (CORBA). Distributed technology serves as glue that brings the objects of a distributed system together in a client/server architecture.

### 3.3.1. Client/Server distributed architecture

In a distributed system, some computers act as servers, and some act as clients [34]. A client sends requests to a server. The requests are transmitted to the server through a communication system for processing. The server responds to requests and transmits results back to the client through the communication system.

In a distributed object system, all entities are modeled as objects. In this case, a client can be seen as an object. The elements of database, the tables, can be modeled as a set of data objects which serve as interfaces to the data in the database. According to the way in which clients interact with data objects, the architecture of distributed object systems falls into two different categories, described in the following two subsections.

#### 3.3.1.1. Two-tier architecture

In a two-tier architecture, a client interacts directly with a data object located on the database server, with no intervening agent or object [43, 44, 45]. In this design, the client can send data requests directly to the data objects on the server and the server sends back the results to the client without any help of agents. In a two-tier application, all processing including functional logic and calculations, is performed on the client side, while the server database serves only as a repository for data. The diagram of a two-tier architecture is shown on Figure 3.
The two-tier application is simple, well understood and easy to build. However, some drawbacks of the 2-tier model can jeopardize the development of distributed applications which require better scalability and performance [43,45]. The two-tier architecture is only suitable for distributed applications which have a small number of users. As the number of users increases, the performance diminishes. This limitation is a result of the server maintaining a connection with each client, even when no work is being performed. However, the connections are valuable resources of the server. Each client requires its own connection and each connection requires CPU time and memory [43].

3.3.1.2. Three-tier architecture

The other approach in designing client-server’ communication is known as three-tier architecture, in which a middle tier (or an "agent") is introduced between the client and the server [43, 44, 45]. There are various ways to implement the middle tier, such as transaction processing monitor technology, a message server, an application server, or a Object Request Broker (ORB) architecture [43].

The three-tier architecture with ORB or Remote Objects is widely used in distributed object systems. In this type of 3-tier models, the application is broken into
three formal tiers: the presentation tier, the middle tier and the data tier. Figure 4 shows a 3-tier architecture.

![Figure 4. A 3-tier architecture](image)

**Presentation Tier**

The presentation tier is typically a GUI, which is responsible for the visual aspects of the application-displaying the client objects and applying the client view of data. The client programs do not query databases. They let the middle-tier perform these tasks for them [45].

**Middle Tier**

The middle tier plays a vital role in a three-tier application, and is the only entity that communicates with the data object of the database on behalf of the clients. In a distributed system, a remote object or ORB can be added as a middle tier. Thus, the interactions with data tier are managed centrally in the middle tier, and the client program could become small and simple [45].

**Data Tier**

The data tier is made up of objects that encapsulate database routines and interact directly with the Database Management System (DBMS) product(s) [45].
The benefits of the 3-tier model are enormous. It can improve the scalability of the distributed system because the database no longer requires a connection from every client -- it only requires connections from a smaller number of middle tier objects [43]. Since all updates go through the middle tier, the latter can ensure that only valid data is allowed to be updated in the database [43]; Changes to processing logic only need to be updated on the middle tier objects and do not have to be distributed to all clients.

The drawbacks of the 3-tier model lie in its complexity [45]. In general, it is more difficult to build a 3-tier application compared to a 2-tier application. The points of communication are doubled (client to middle tier to server, instead of simply client to server)

3.3.2. Communication in a distributed object system

Distributed application involves calls to other address spaces either on the same computer or on a remote machine. Currently, there are two major technologies widely utilized in distributed object-oriented applications: (1) Common Object Request Broker Architecture (CORBA), and (2) Remote Method Invocation (RMI). All distributed systems based on these two technologies share some common features.

3.3.2.1. Features of a distributed system

A distributed system requires locating or invoking objects on remote host computers, and interacts with them as if they were located on the same computer [35]. To meet this requirement, the distributed object systems have to possess some necessary features: interface specification, registration/naming service, object manager and object
communication protocol. The general architecture for a distributed object system is shown on Figure 5.

A distributed object system uses an object interface specification to generate the implementation of server objects, an interface between the object implementation and the object manager, called a skeleton, and a client interface called a stub. The skeleton is used to create new objects by the server and to forward the client requests for the object implementation. The stub is used by the client to route transactions to the object on the server. The object manager manages the skeletons and object references on the server. On the server side, a registration service registers the object implementation with a naming service and an object manager, so that it can be addressed by clients. Then the object can be stored in the server storage area for object skeletons. After an object is fully registered with a server, a client can request a remote object through the naming service.

Figure 5. General architecture of a distributed object systems [35]
3.3.2.2 Common Object Request Broker Architecture (CORBA)

CORBA is a distributed object standard developed by the Object Management Group (OMG) and their corporate members and sponsors. This standard describes an architecture, interface and protocols that distributed objects can use to interact with each other [46]. The purpose of CORBA is to be a generic framework for building distributed object systems. This framework consist of three major elements: Object Request Broker, Interface Define Language (IDL) and Internet Inter-ORB Protocol (IIOP).

The IDL is used to define the object interface. The interface for a remote object can be written using IDL to define data types, including the operations that an object supports, as well as data variables available on the object. The IDL definitions and types can be mapped to programming languages such as C, C++, Smalltalk, and Java by a special IDL compiler. Compiling the IDL file will yield stub and skeleton, which serve as proxies for the client and the server of the CORBA object, respectively. The stub and the skeleton don’t have to be compiled into the same programming language which is a principle feature of the CORBA standard.

The ORB is the core part of the CORBA framework. It serves as the object manager and provides the registration/naming service. In a CORBA distributed system, the client and the server of CORBA objects use an ORB to talk to each other. Thus, both client and server side of a distributed object have an object manager. The ORB on the client side is responsible for accepting client requests for a remote object, finding the implementation of an object using naming service, accepting the client-side reference to the remote object, routing client method calls through the object reference to the remote object implementation, and accepting any results for the client. On the server side, the
ORB enables object servers to register new objects. The server ORB receives the request from a client for a given object and uses the object skeleton to invoke the corresponding object method to create a new object. The server ORB generates a reference for the new object, and sends this reference to the client. The client ORB converts the reference into a stub and uses it to invoke methods on the remote object. The server ORB receives the request and calls the method on the object implementation through the skeleton. The server ORB marshal the returns values and sends them back to the client ORB, where they are unmarshaled and further delivered to the client. The server ORB and the client ORB communicate via the Internet Inter-ORB Protocol (IIOP). The figure 6 shows the relationship between the stub, the ORB, and the skeleton.

![Diagram of stub, ORB, and skeleton](image)

Figure 6. The stub, the ORB, and the skeleton [46]

### 3.3.2.3 Remote Method Invocation (RMI)

When two Java applications are running in different address spaces, a virtual barrier that surrounds each application prevents one application from being able to call
functions in the other address space. The Remote Method Invocation (RMI) technology provides a solution to bypass this actual barrier [42], which is used for creating communication channels between Java application objects, in which the methods of remote Java objects can be invoked from other Java virtual machines, possibly on different hosts. [42].

The RMI distributed system includes several elements: Remote Object Interface, Remote Object and RMI registry [35]. Unlike CORBA, in which each interface is written in a specific IDL, all remote object interfaces in RMI are written in pure Java programming language. The client stub and the server skeleton are generated from the interface. Once the interface of the remote object is defined, a server implementation of the remote object can be written. The remote interface does not introduce any methods to the object interface, it just serves to mark remote objects for the RMI system. All implementation of remote objects is done on the server side.

The RMI registry serves as both an Object Manager and a Naming Service for the distributed object system [35]. The registry runs in the Java runtime environment on the server of remote objects, binding remote objects to names, and listing the available objects on the server [35]. The registry listens to a well-known port (e.g. 1099) on the local host for connections. The interaction between RMI objects is illustrated on Figure 7.
Figure 7. Message sequence chart between RMI objects

*RMI Architecture*

From an architecture aspect, the RMI system is built in three layers [47]:

- Stub/skeleton layer which includes client-side stubs and server-side skeletons
- Remote reference layer
- Transport layer

The relationship between each layer is shown in Figure 8. Usually the client and the server belong to a separate layer which is called application layer, which sits on top of the RMI system.
The Stub/Skeleton Layer

The stub/skeleton layer is the interface between the application and the RMI system, which transmit data to the remote reference layer of the RMI system. The transfer of data can be accomplished via object serialization, which is a Java built-in mechanism that enables Java objects to be transmitted between address spaces [47]. In this way, the object is transmitted by value between Java virtual machines. Otherwise, the remote object is passed by reference.

The Remote Reference Layer

The Remote Reference Layer deals with the lower-level transport interface. This layer understands how to interpret and manage references made from clients to the remote server objects.

The Transport Layer

The transport layer is based on TCP/IP connections between the clients and the server in a network. Here, an actual connection is set up and transport of data from one
machine to another is taken care of. The transport layer sets up a stream connection, which can be accessed by the reference layer. It also monitors the connections, listens for incoming calls, and passes data to and from the remote reference layer.

A client invokes a method on the local stub for a remote object. The stub carries out the method call to the transport layer via the remote reference layer. In order to dispatch to a remote object, the transport layer routes the remote call up to the remote reference layer. Any required server-side behavior is handled by the remote reference layer before the requests can be passed to the server-side skeleton. Once the requests are acquired, the server-side skeleton of a remote object makes an up call to the remote object implementation which carries out the actual method call. The return value of a call is then sent back through the skeleton, the remote reference layer, and the transport layer on the server side, and further up through the transport layer, the remote reference layer, and the stub on the client side.

In addition to RMI allowing the objects be passed as a parameter in the distributed system, it also supports dynamic class loading on demand. If a necessary stub is not available on the client machine, it can be dynamically loaded to the receiver’s Java virtual machine [47].

### 3.3.3 CORBA vs. RMI

CORBA and RMI are two standards for distributed object systems. They share some similarities in terms of functionality. There are also some critical differences between these two technologies. The differences are summarized in Table 1. In the present application, Java-based RMI technique will be used.
### Table 1. Differences between CORBA and RMI

<table>
<thead>
<tr>
<th>Feature</th>
<th>CORBA</th>
<th>RMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic stub downloads</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Dynamic class downloads</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Objects can be passed-by-value</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Interface descriptions</td>
<td>Yes (via CORBA IDL)</td>
<td>Yes (via Java interfaces)</td>
</tr>
<tr>
<td>URL-based naming</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Communication Protocol</td>
<td>IIOP between ORBS</td>
<td>Object serialization</td>
</tr>
<tr>
<td>Language specificity</td>
<td>Open</td>
<td>Java only</td>
</tr>
</tbody>
</table>

#### 3.4 Web technology

The World Wide Web (WWW) is, essentially, a large distributed system based on client/server computing [48]. Although the web is generally a document transfer system, it offers the mechanisms for distributed application communication. The functionality of HTML, HTTP protocol, and server extension mechanisms such as servlet provide a flexible communication channel from client browser to the web server.

#### 3.4.1 Fundamentals of the web technology

The core concepts in the web technology have been HTML, HTTP and the notion of URL (Uniform Resource Locators). HTML, provides functionality such as forms for entering data, tables, and frames to create web pages. The HTTP is an application-level protocol which has been in use in WWW to transfer HTML documents across the net. URLs are strings that specify how to access network resources, such as HTML
documents. The most important use of URLs is in HTML documents to identify the targets of the hyperlinks [33].

The relationship between a client and a server can be generally summarized as a request-response model. The client requests some action be carried out and the server acts upon the request and responds to the client. A common approach of this request-response model is illustrated with WWW browsers and WWW servers. When users choose a web site to browse in their browser, a request is sent to the corresponding web server (the server application) for processing. Normally the server responds to the client by sending the HTML web pages. All communication between client browsers and web servers is through HTTP which locates data on the Internet using URLs. The two most common HTTP request types (also known as request methods) from the client are \texttt{GET} and \texttt{POST}. A \texttt{GET} request retrieves information from the web server. A \texttt{POST} request is used for HTML forms and other operations that require the client to transmit a block of data to the server.

\subsection*{3.4.2 Java Applet graphical user interface}

In most cases, web applications require complicated, dynamic graphical user interfaces with rich functionality rather than simple static documents that contain data. The web interface should be able to display in a web browser. There are typical choices: Java applets and Active X. A Java applet is a Java-language program that can be transmitted from a server to a client computer, where it is executed through an HTML browser [40]. Unlike the HTML document, an applet is dynamic, not static: an applet is a general program, and can accept input from mouse clicks, the client keyboard, or other
interactions. It is embedded in the HTML using an “applet tag”. When the browser reads the tag, the applet can be automatically invoked for the user to interact with it. A Java applet can also communicate with the server objects using different technology such as HTTP, RMI and CORBA [40]. One valuable feature of a Java applet is that it can write once and run everywhere. Because standard web browsers provide a Java Virtual Machine which acts as an interpreter, a Java applet will run on all platforms for which the browser is implemented; For security reason, most web users are hesitant to install programs on their own computers. A Java applet can be downloaded to the user computers on demand and it is not allowed to access and operate on the file and data on the user machine. A Java applet can be created and modified on the central server and distributed to any users on the Internet. This can significantly reduce both maintenance and administration costs.

3.4.3 Servlet technology

Servlets, the latest technology for enhancing the capabilities of a web server, are small platform-independent Java-based programs or components that run on a web server [49]. Servlets extend the functionality of servers in much the same way as applets extend the functionality of browsers. Servlet technology enables developers to create complete, distributed applications, in which the user interface logic is displayed in the Web browser and the business logic is managed in server programs [49].

Architecturally, servlets interact with a servlet engine through requests and responses. The servlet engine is a software package which serves as a web server plug-in for running Java applications on the server. A client program, such as a web browser,
makes a request to a web server. The web server passes the request to the servlet engine, which in turn forwards the request to the appropriate servlet. The servlet then sends the required response back via the same route to the client. The servlet technology architecture is shown on Figure 9.

![Servlets technology architecture](image)

**Figure 9. Servlets technology architecture**

### 3.5. Principles of relational database design

Database design is defined as the design the logical and physical structure of one or more databases to accommodate the information needs of the users in an organization for a defined set of applications [50]. Database design phases roughly can be divided into the following five steps:

- Requirement Collection and Analysis
- Conceptual Design
- Logical Design (Data Model Mapping)
- Physical Design
- Implementation
In the first step of the database design process, requirements have to be collected and analyzed. The result of this step is a concisely written set of requirements, which is specified in details. The next step is a conceptual design process, which creates a conceptual schema based on the request from the first step. Logical Design transforms the conceptual schema from a high-level data model into an implementation data model. The output from this step is a database schema. The physical design is the process of choosing specific storage structure and access paths for the database to achieve good performance. This process includes analyzing the database query and optimizing the query process [50].

3.5.1. Entity-Relationship data model

The Entity-Relationship (ER) Model is the most common method used to build data models for relational databases [51]. The ER Model is a conceptual data model that views the real world as consisting of entities and relationships. The model visually represents these concepts by the ER diagram. The basic constructs of the ER model are entities, relationships, and attributes [51].

**Entity**

Entities are the principal objects about which information is to be collected in the ER model. An entity is analogous to a table definition in the relational model. An entity instance is analogous to a row in the relational table.

**Attributes**
Attributes describe the entity with which they are associated. A particular instance of an attribute is a value. Each simple attribute of an entity instance is associated with a value. An entity usually has an attribute, whose values are used uniquely to identify an instance of an entity. An entity can have composite attributes. The composite attribute can be divided into a few simple attributes [51].

Relationships

A relationship represents an association between two or more entities. Relationships are classified in terms of degree and cardinality. The degree of a relationship is the number of entities associated with the relationship. The cardinality of a relationship is the actual number of related occurrences for each of the two entities. The basic types of cardinality for relations are: one-to-one, one-to-many, and many-to-many [51].

ER Notions

All notational styles represent entities as rectangular boxes and relationships as lines connecting these boxes. Each style uses a specific set of symbols to represent the cardinality of a connection. The symbols used for the basic ER constructs are:

- Entities are represented by labeled rectangles. The label is the name of the entity.
- Relationships are represented by a diamond box, the two end points of which are connected to two entities with lines. The name of the relationship is written inside the box.
- Attributes are listed inside an oval box. Key attributes are underlined.
• Cardinality relations are represented by numbers which are written on both sides of a relationship diamond box.

3.5.2. Relational data model

The relational model represents the database as a collection of tables [51]. In the relational model, the data is represented in the form of two-dimensional relational tables. A relational table is a flat file composed of a set of named columns and a number of unnamed rows. The columns of the tables contain information about the table. The rows of the table represent occurrences of the instance of a data object represented by the table. A data value is stored in the intersection cell of a row and a column. In a relational table, columns are not repeating groups or arrays, and all values in a column come from the same domain. There should not be two rows in a relational table that are identical. There is at least one column, or a set of columns, the values of which uniquely identify each row in the table. Such columns are called primary keys. A foreign key is a column or columns, which values are the same as the primary key of another table. A foreign key can be seen as a copy of a primary key from another relational table. The association between relational tables can be described as a relationship, which is expressed by the data values of the primary key and the foreign key. The relationship between two relational tables is made by matching the values of the foreign key in one table with the values of the primary key in another. Relational tables can be represented concisely by eliminating the sample data and only showing the table name and the column names. The relational model provides a standard for the design of a relational database.
3.5.3. Relational database design using ER-Relational mapping

Based on the concept of Entity-Relationship and of the relational model, the logical database design (or data model mapping) can be performed using a special mapping algorithm called ER-to-Relational Mapping [51]. This process is known as designing a relational database schema based on conceptual modeling.

The mapping algorithm is designed according to the correspondence between ER and relational models. The summary of this correspondence is depicted in table 2:

<table>
<thead>
<tr>
<th>ER Model</th>
<th>Relational Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entities</td>
<td>Relational Tables</td>
</tr>
<tr>
<td>1:1 or 1:N relationship</td>
<td>Foreign key</td>
</tr>
<tr>
<td>M:N relationship</td>
<td>“Relationship” table and two foreign keys</td>
</tr>
<tr>
<td>Simple attribute</td>
<td>Attribute</td>
</tr>
<tr>
<td>Composite attribute</td>
<td>Set of simple component attributes</td>
</tr>
<tr>
<td>Key attribute</td>
<td>Primary key</td>
</tr>
</tbody>
</table>

Table 2. Correspondence between ER and relational models.

3.6. Connecting to a database

3.6.1. Java Database Connectivity

Java Database Connectivity (JDBC) is a standard application-programming interface (API) for database-independent connectivity between Java application/Applet and a variety of Structured Query Language (SQL) databases, primarily relational databases [52]. The JDBC API includes a set of classes and interfaces written in Java programming language and makes it easy for Java applications to send SQL statements to
relational database systems. This set of classes and interfaces refer to the package `java.sql` which is the core API of JDBC [52].

To use the JDBC API with a particular database management system, a JDBC technology-based driver is necessary to act as a bridge between JDBC technology and the database. The driver acts as a translator that takes generic database access calls and translates them to database-specific calls, which allow access to the particular database directly from the Java applications [52].

The architecture of a JDBC implementation is depicted in the Figure 10, which shows all major classes and structures. The classes which implement each layer can be found in the `java.sql` package.

![Figure 10. JDBC architecture [47].](image)

In the JDBC implementation, a Java application will instantiate a JDBC driver under the control of the DriverManager class included in the `java.sql` package. The driver
is database-specific. When the Java application needs to connect to a particular database, it asks the DriverManager for a reference to the driver which can communicate with that database. Using the reference returned by the DriverManager, the application can now ask the driver to open a connection to the database. The driver separates the application from the database implementation, networking issues and the actual location of the database. Once the driver has opened the connection to the database, it returns a reference to a Connection object which is used as an agent for the application to request service from the database. The Connection object also provides the application with a Statement object which is used to build the SQL statements that are executed in the database. After a Statement object executes the query, the selected rows are returned in a ResultSet object, which allows the application access to the returned rows and provides mappings between SQL data types and the appropriate Java classes and primitives [47].

### 3.6.2 Open Database Connectivity

There is another database access API called Open Database Connectivity (ODBC) [47]. It is also widely used as a programming interface for accessing relational databases. It provides the ability to connect to almost all databases on almost all platforms. However, it is not appropriate for direct use from Java programming language because it uses a C interface. Calls from a Java application to C code have a number of disadvantages in the security, limited portability and increased the maintenance burden on each client. The ODBC procedure methodology does not match well with the Java object model. Using ODBC from Java is best done with the help of the JDBC-ODBC bridge driver [47].
3.7. Conclusion

Distributed object systems can be analyzed and designed using object-oriented analysis and object-oriented design concepts and implemented using object-oriented programming. Different system architectures and distributed technologies have been applied to build distributed object systems. Among them, Java distributed computing is a popular approach because of Java's distributed and objected-oriented nature. Typically, a Java distributed object system adopts a three-tier architecture which is achieved by HTML, Applet, and Java Application on the client, Servlet or RMI server or ORB on the middle tier, and JDBC communication to the Database layer. The components and the interactions typical for this architecture is summarized in table 3.

<table>
<thead>
<tr>
<th>Client</th>
<th>Client to Middle Tier communication</th>
<th>Middle Tier</th>
<th>Middle tier to database communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML</td>
<td>HTTP</td>
<td>Servlet</td>
<td>JDBC</td>
</tr>
<tr>
<td>HTML with applet</td>
<td>HTTP</td>
<td>Servlet</td>
<td>JDBC</td>
</tr>
<tr>
<td>Java Application</td>
<td>Java Remote Method Protocol</td>
<td>RMI Server</td>
<td>JDBC</td>
</tr>
<tr>
<td>Java Applet</td>
<td>Java Remote Method Protocol</td>
<td>RMI Server</td>
<td>JDBC</td>
</tr>
<tr>
<td>Java Application (Not a Java 3 tier)</td>
<td>IIOP</td>
<td>CORBA</td>
<td>JDBC</td>
</tr>
</tbody>
</table>

Table 3. Components of Java Three-Tier architecture

From the above table it can be concluded that the middle tier can be achieved in Java using RMI or Java IDL (an implementation of CORBA). The use of both technologies is similar, since the first step is to define an interface for the object. However, RMI provides a Java only solution to distributed objects. If the server objects are built in Java, the client objects should also be Java-based.
CHAPTER FOUR: SYSTEM DESIGN AND IMPLEMENTATION

The software development of Tele-Electrogastrography consists of three phases: (1) requirement analysis; (2) system design; and (3) system implementation.

4.1. System requirements analysis

The design objective of the Tele-Electrogastrography system is to provide a comprehensive, easy-to-use and worldwide accessible distributed system for registered submission, classification and storage of human electrogastrograms. To achieve this purpose, the system should provide clinical professionals, researchers and medical doctors interested in the EGG research area the following functionalities:

- Generating Time/Frequency (T-F) plots from calculated power spectrum values supplied by the users;
- Performing statistical evaluation of the T-F points, including mean values, standard deviations and probability density functions;
- Classifying the EGG data into one of the 7 pattern groups (normal, and abnormal Patterns 1-6) according to the statistical evaluation;
- Associating EGG recordings with pre-diagnosed gastrointestinal motility disorders.

According to the above scenarios, use cases are used to help in the modeling of the requirement specification of Tele-EGG. In this modeling, described in Figure 11, the actor is defined as the user of the system, whereas use cases are scenarios that actors carry out. Three actors are identified as: general viewers, active participants and system administrator respectively. There are total of nine use cases defined in the conceptual
model. A brief description of each use case is as follow:

1. Use Case: Submit EGG data

   Active participants and the system administrator can submit EGG power spectrum values. This case also includes another use case: (8) Classify EGG data, which classifies the submitted case into different EGG patterns and stores the calculated Time-Frequency values in the database.

2. Use Case: Plot T-F points

   Users (viewers and active participants) can view Time-Frequency plots of the EGG T-F points stored in the database. This case also includes another use case: (9) Retrieve T-F values, which performs the functions of querying the database and retrieving the required data.

3. Use Case: Plot PDF

   Users (viewers and active participants) can view probability density function plots of the T-F points.

4. Use Case: Display statistics

   Users (viewers and active participants) can view statistical evaluations of the T-F points.

5. Use Case: Examine case-based diagnosis

   Users (viewers and active participants) can examine the relationship between the obtained EGG patterns and a given set of pre-diagnosed GI disorders.

6. Use Case: Associate submitted EGG data with the set of pre-diagnosed GI disorders

   This use case is included in the use case “Examine case-based diagnosis”. It sets up a relationship between a submitted case and a given GI disorder.
7. Use Case: Maintain the database

The system administrator can input, delete and update the EGG data in the database.

![User diagram for modeling system requirements](image)

Figure 11. User diagram for modeling system requirements
4.2. System architecture

Architecture represents the overall structure of a system [43]. The Tele-Electrogastrography system discussed in the present study was built using a three-tier distributed application architecture consisting of: (1) data tier; (2) middle tier; and (3) user interface tier. Figure 12 represents the developed client-server architecture. The advantage of three-tier architecture has been discussed in the previous chapter. The User interfaces in the Tele-Electrogastrography application are created using Java applets and HTML, which can be downloaded on demand and run on a standard web browser. The clinicians and researchers can use standard web browsers to directly access the application around the world, without the installation of any specific client program on their machines. The middle tier includes the web server, RMI remote objects and servlets. It acts as the gateway or broker between the interface tier and the data tier and

![Three-tier system architecture](image-url)
plays a vital role in the Tele-Electrogastrography system. In the middle tier, RMI remote objects are responsible for handling data requests from the Java applet client and for retrieving the required T-F values from the database. Since the data processing is fully implemented using a RMI remote object, the applet program on the client side is relatively simple. This approach is known as "thin" client design, which makes the file size of the applet client program small and rapidly downloadable from the web server. The servlets are designed to extract the spectral values from a submitted HTML form, supplied by the users, calculate the T-F values and classify the obtained quantitative assessment into the database. The data tier and the middle tier are currently running on the same web server. The data flow among the components of the system is shown on the Figure 13:

Figure 13. EGG data flow diagram
4.3. Design and implementation of user interfaces

4.3.1. Java Applet

In most web-based distributed applications, Java applets have been widely utilized to generate dynamic charts, plots and other graphical features, and is supported by all popular web browsers (i.e. Netscape Navigator and Internet Explorer). The Java applet in Tele-EGG is designed to present the Time-Frequency plots, PDF plots, and to display the statistical results for EGG analysis. All graphical components are built in Java Abstract Windowing Toolkit (AWT).

4.3.1.1. Getting the data from the database

Time-Frequency values are stored in the database on the server. These data have to be transmitted to the client applet for visualization and analysis. Transmitting of these data is performed by a RMI remote object. First, the applet receives a reference to the RMI remote object from the server host’s rmiregistry as follows:

\[
\text{Naming.lookup("/" + getCodeBase().getHost() + "/RMIserver")};
\]

Once the reference is obtained, the Java applet can invoke the methods of the remote object which is designed to fetch the T-F data from the database. The transmitted data values can be stored in a local array for further calculation and rendering.

4.3.1.2. Statistical Evaluation

Once the required multi-channel spectral data has been obtained, statistical evaluation can be performed and displayed in the applet. The algorithm for calculating mean value, variance, standard deviation and RMS is described in the following block diagram. (Figure 14)
Figure 14. Block diagram for calculating the Gastric averages of the submitted T-F points

M-Mean value, N-Number of the T-F points, D-Value of T-F points

4.3.1.3. Rendering

The applet graphic interface consist of two main components; they are the Panel and Image Layers (Figure 15).

Figure 15. User interface graphical components
The T-F values of each channel are plotted on an image layer. All image layers are overlapped and arranged on the Panel. The Panel and the Image layer use two separate coordinate systems respectively: device coordinates and virtual coordinates. Device coordinates are defined in terms of pixels, and usually the upper-left-hand corner is the origin of the coordinate system, with \( x \) coordinates increasing to the right and \( y \) coordinates increasing downwards. In contrast, virtual coordinates are more flexible in the units employed, the position of the origin, and even in the direction in which the coordinates increase. Here, virtual coordinates are defined by assigning coordinates to the edges of a device, include a linear transformation to device coordinates, and are floating point quantities with the origin at the lower left. The linear transformation is described by the following equations:

\[
X_{Device} = X_{Virtual} \\
Y_{Device} = H_{panel} - Y_{Device} \tag{4.31}
\]

All graphical operations such as drawing lines, grids and PDF bars within each layer are done using the Graphics class which is included in the Java Abstract Windowing Toolkit (AWT). The Graphics class provides methods for drawing simple geometric shapes, text, and images to the graphical destination. All output to the graphical destination occurs via an invocation of one of these methods. The sequence diagrams of drawing the T-F plot, and the PDF of a single EGG channel are shown on the Figures 16, and Figure 17, respectively:
Figure 16. Sequence diagram of use case of plotting T-F points into a T-F plot

Figure 17. Sequence diagram of use case of plotting PDF
4.3.2. HTML forms

The web pages of Tele-Electrogastrography, including both text and graphics, are written in HTML to provide system information and internal links among the various web pages. HTML forms provide several input type elements including Text Box, Submit Button and File Browser, which can be used to input text and to attach files. Common uses of forms are surveys, on-line order forms, feedback, or essentially any web page in which input is required from the user in order to accomplish a given task or provide a given service. Tele-Electrogastrography utilizes HTML forms to allow active participants to upload the files containing spectral values to the web server with a POST request. The HTML form for submission can be implemented with the following tag definition:

```html
<FORM METHOD=POST
ACTION="http://mint8.enel.ucalgary.ca/gasServlet/
Servlet/gasServlet.InputGasDataServlet">

"METHOD" specifies the POST request that the web server (mint8.enel.ucalgary.ca) will use to pass the form data to the program which processes it, and "ACTION" tells the server exactly which program (gasServlet.InputGasDataServlet) that is. The web server forwards the POST request and the submitted HTML form to the corresponding servlets for further processing.
4.4. Middle tier

The middle tier of the Tele-Electrogastrography application includes the web server, the RMI remote objects and the servlets.

4.4.1 Design and Implementation of the RMI remote objects.

RMI remote objects have been designed using two different patterns: Factory Object and Replicated Object [53]. The original design adopted in Tele-EGG is the Factory Object pattern. The class diagram and the sequence diagram of this design is shown on Figures 18, and 19.

Figure 18. Class diagram of remote objects
4.4.1.1. Using factory design pattern

The Factory Method is a creational pattern [54]. This pattern helps to model an interface for creating an object which at creation time can let its subclasses decide which class to instantiate. It is called a Factory Pattern because it is responsible for
"manufacturing" an object. It helps instantiate the appropriate subclass by creating the right object from a group of related classes [54].

Using the object Factory Design Pattern can reduce the number of objects that need to register with the RMI registry when designing an RMI distributed object system. In order to invoke a method on a remote object, the Java applet has to obtain a reference to the remote object. The RMI registry is a simple server-side name service that allows remote clients to get a reference to a remote object. However, the client applications can dynamically request the creation of new remote objects, and registering all of these remote objects with the RMI registry is unnecessary. When using the Factory Design Pattern, only one remote object is registered in the RMI registry and acts as the Factory Object. Upon a client request, the Factory Object would in turn provide application-specific support for creating or finding other remote objects. The utilization of the Factory Design Pattern in the Tele-EGG system is depicted on Figure 20:

![Diagram of Factory Design Pattern for RMI](image-url)

Figure 20. Factory Design Pattern for RMI
The DBManagerImpl (see Figure 20) acts as a factory object. It is used to request additional remote references and returns these remote references to the Java applet client.

1. The DBManagerImpl object is registered using the rebind() method in the rmiregistry;

2. The Java applet requests a reference to the DBManagerImpl object;

3. The rmiregistry returns a remote reference to the DBManagerImpl object;

4. The Java applet invokes a remote method on the DBManagerImpl object to get a remote reference to data objects, which include PatternImpl, CaseImpl and TFDataSetImpl;

5. The DBManagerImpl object returns a remote reference to an existing data object implementation;

6. The Java applet invokes a remote method on the data object implementation.

When using the Factory Design Pattern, the reference to a data object is passed to an applet client, which is called pass-by-reference. With the pass-by-reference, data objects have to implement a remote interface that extends a java.rmi.Remote class and all methods in the data object are remote calls over the network. An instance of a data object represents a row in a database table. In order to get the T-F value of each channel to create the Time-Frequency plots, the Java applet has to obtain a set of references to the instances of the TFDataSetImpl object (see Figure 20) which are created by the factory object and actually represent the rows of the table tblTFDataSet. The DBManagerImpl object returns the collection of references to the instances of the TFDataSetImpl object to
the client. The client makes remote calls over the network back and forth on the methods such as `getChannel1()`, `getChannel2()` and so on. This could lead to an excessive number of network calls, which would increase the data transfer time dramatically. Thus the performance of the system could be downsized. To solve this problem, another design pattern was introduced in designing the RMI remote objects. This approach is known as the Replicate Object Pattern [53].

4.4.1.2. Using the replicated object design pattern

Besides RMI giving clients access to remote objects by reference, it also supports accessing remote objects by value. In this case, a remote call results in a return value of an object that implements `java.io.Serializable` interface instead of that extend `java.rmi.Remote` class. The java.io package includes classes that can convert an object into a stream of bytes and reassemble the bytes back into an identical copy of the original object. An object that is passed by value is “serialized” (or marshaled) on the server, sent to the client’s Java virtual machine (JVM), and “reconstituted” (or unmarshaled), instantiating a copy of the remote object in the client’s JVM [53]. This object is then called a replica. Methods invoked on this copy of the remote object are just like any other Java methods. They are executed locally, without any further communication with the server. When using this pass-by-value technique, the problem of network communication such as latency can be avoided. Using the pass-by-value approach for most objects in the system and replicating them at the client end is called Replicate Object Design Pattern [53]. Object replication is particularly useful when there is a lot of remote method calls between various clients and the server.
In our system, the DBManagerImpl object is still used as the factory object which is responsible for creating instances of data objects. On the server side, all of the data object classes have to be serialized by implementing `java.io.Serializable` interface. When the Java applet requests a reference to the DBManagerImpl object, the rmiregistry returns a remote reference to the DBManagerImpl object. The Java applet makes remote calls on the methods of the DBManagerImpl object. Instead of returning the references to the data objects, the DBManagerImpl object marshals the data objects and passes them by value to the Java applet. The methods of the replicated objects can be invoked locally. Thus the number of the remote method calls can be reduced dramatically. As a result, the performance of the system would be greatly improved. The modified class diagram is shown in Figure 21.

![Modified class diagram utilizing the Replicated Design Pattern](image)

Figure 21. Modified class diagram utilizing the Replicated Design Pattern
4.4.2. Design and implementation of servlets

Servlets are executable Java programs which run on the server side and can be automatically launched by the web server [49]. Located on the web server, servlets can communicate with the HTML interface by responding to HTTP requests and processes, and store data submitted by an HTML form into the database. When the servlet receives a POST request, the “doPost” method of the servlet is called to handle the request. This method is combined with the necessary algorithms to extract the data contained in the submitted HTML form and to store them into the database. In Tele-Electrogastrography, the servlet is designed to implement three algorithms. The first is related to retrieval of the parameters for the FFT/FHT transforms and the relevant spectral values from the submitted HTML forms of active participants. The second algorithm is used to calculate the T-F values using the submitted spectral values of 8 standard EGG channels in a specified frequency window (default is 1 to 9 cpm). The last algorithm handles the classification and the storage of the T-F values in the EGG database. After the T-F values are successfully stored in the database, the “doPost” method generates and sends a confirmation HTML document to the users. The interaction between HTML client and servlet is shown in the Figure 22.

4.4.2.1. Extracting the data from the uploaded file

An HTML form provides a way for the user to attach and submit data. In an HTML form, there are series of fields to be supplied by the user. Each field has a name. The names of the fields are unique within a form. These fields include text field, option
field and field for file attachment. For example, in Tele-EGG the user fills an HTML form which contains a text field (field1) and field for file attachment. Once this

![HTML client and servlet object interaction diagram](image)

Figure 22. HTML client and servlet object interaction diagram

form is submitted to the web server, the whole form content becomes the body of an HTML message in such a way that the message can be parsed. The format of the message is listed below:

```
Content-type: multipart/form-data, boundary=7d01ecf406a6
-----------------------------7d01ecf406a6

Content-disposition: form-data; name="field1"

Value of field1
-----------------------------7d01ecf406a6
Content-disposition: form-data; name="pics"; filename="file1.txt"
Content-Type: text/plain
Content-Transfer-Encoding: binary
```
The HTML message is parsed in several sections. Each section contains a content-disposition header. The attached file name may be described using the “filename” parameter in the “Content-disposition” header, which is followed by the contents of the attached file. In the implementation procedure, the whole HTML message is read into a Java input stream object first. Next, the section with “filename” parameter in the “Content-disposition” header has to be found. In this section, the contents of the attached file can be located by determining its start and end point. Then the body of contents can be written into a Java output stream object.

4.4.2.2 Calculating T-F values from the submitted spectral values

Once the spectral values have been extracted from the HTML form, these data can be used to calculate T-F points which represent the dominant peaks of frequency spectrum. Since the spectral values are obtained from time-domain signals using FFT/FHT transforms, the parameters of these transforms have to be submitted in order to compute the dominant peaks of each spectra. The parameters include the sampling frequency (Fs), the numbers of points of the FFT / FHT transform (N), the % overlap (OVL) and the duration of the recording (Ds). The number of the spectra of each channel (N_{tf}) can be calculated using Equation [2.32]. The algorithm for calculating the T-F values of a given EGG channel is visualized in the block diagram of Figure 23. All spectral values have to be normalized per individual spectrum before the calculation.
Get frequency window (default is 1 to 9 cpm)

Get Frequency Resolution $\Delta F (\Delta F = \frac{F_s}{N})$

Convert lower and upper frequency of given window to sampling steps
(Upper=$\frac{1}{60} \times \Delta F$, Lower=$\frac{9}{60} \times \Delta F$)

$I = 0$;

End

No

Is $I < N_{tf}$?

Yes

Dominant_value = 0; Domain_frequency = Lower

$I = I + 1$

K = Lower

Is $K < \text{Upper}$?

No

K = K + 1

Yes

Is spectral value DIK > Dominant_value?

No

Dominant_value = DIK; Dominant_frequency = K;

Yes

Figure 23. Block diagram for calculating the T-F values of a single EGG channel

4.4.2.3. Classification of the calculated T-F values into different EGG patterns

Once the T-F values have been obtained for each channel, statistical evaluation of T-F points has to be performed [15], calculating their mean values, standard deviation,
RMS and PDF (see also Chapter 2). The PDF maxima of each channel can be further compared with each other so that the number of channels with coinciding PDF maxima in the normal frequency range can be determined. Then an algorithm for the classification of the EGG signals into different patterns can be developed according to the criteria for identifying normal and abnormal EGG [15] (Figure 24). After the classification is completed, the T-F value can be stored into the database in a pre-defined relationship structure. The storage process is accomplished using SQL queries which are executed by the JDBC.

![Classification algorithm flow chart](image-url)

Figure 24. Classification algorithm flow chart
4.4.2.4. HTTP tunneling with servlets

Some corporate establishments and scientific institutions may set up firewalls to protect their private networks being accessed from outside the firewall over the Internet. The firewall is typically implemented on a gateway device and is usually configured to allow only HTTP protocol requests pass through (by enabling only incoming requests on the default HTTP port 80), while blocking both incoming and outgoing requests on other ports [35]. This allows the users outside the firewall to only browse web pages and access nothing else. RMI clients typically communicate with remote objects using direct socket connection on the default port 1099 or 2099 which leads to fast data transferring. If these ports are configured to be disabled, the remote method calls would not pass through the firewall and the Tele-EGG would not be able to run properly. In order to solve this problem, HTTP tunneling with servlets is introduced as a way of bypassing this restriction and enabling Java applets to communicate with the servers. This is done by creating servlets on the server that listen for requests on port 80 from the Java applet. The applet simply has to open an HTTP socket connection to the specified servlet URL. Once this connection is created, then the applet can get an object output stream or an object input stream on the servlet. To send request messages to the servlets, the applet writes an serialized object containing requests into the output stream. Once the corresponding servlet is invoked, it gets the sent object from the object input stream. Upon the request, the servlet gets the T-F values from the database and packs them into a serialized object which is then written into the object output stream. Thus, an applet can read this object from its object input stream and get the required T-F values. Thus the object stream acts as a pipe between the applet and the servlet.
4.5. Modeling and designing the EGG database

The database is the heart of the Tele-EGG system, providing a powerful storage mechanism for the submitted EGG data. Similarly to the software development, the first step of the database design is requirement collection and analysis [50]. During this step, data requirements have to be fully understood and documented to form the specifications. Once all the requirements have been collected and analyzed, the database conceptual model should be designed so that data entity and relationships between them could be clearly outlined. The Entity-Relationship (ER) model [51] was applied in the design of the EGG database. The ER diagram is shown in Figure 25.

![Entity-Relationship Diagram]

Figure 25. Entity-Relationship Diagram
There are three entities derived from the specification requirements which are illustrated in rectangular boxes, the Case entity, the User Profile entity and the T-F Data entity. Each entity has a set of attributes denoted by oval boxes. The Case entity represents each data file that users submit. Along with the file, the parameters which are specified as attributes of the Case entity have to be provided. The User Profile contains personal contact information. The T-F data entity represents the multi-channel T-F. Each entity has a key attribute which makes the entity uniquely identified. The user can submit one or multiple data files. Each submitted file contains multi-channel spectral values, which can be further used to calculate the T-F data set. The described associations are modeled as relationships between the three entities and are depicted with diamond boxes in the diagram of Figure 25.

Based on this conceptual model, the relational database scheme can be created by mapping the ER diagram. The mapping should follow the specified algorithm which is known as ER-to-Relational Mapping Algorithm introduced in the Chapter 3. The result of the mapping is a set of relational tables shown in Figure 26.
Figure 26. The relational database scheme, which is generated from the mapping of the ER diagram.

The database design phase was followed by database implementation using the Microsoft database management system Access (Microsoft Corp, Redmond, WA).

4.6. Web server deployment

The web server provides the primary gateway between the data sources and the client browser. In addition to communicating with the client interface through the HTTP protocol, the web server can be used to host other application programs, including RMI objects and servlets needed for the distributed web-application.

Tele-Electrogastrography was deployed using the popular Apache web server, created by the Apache Software Foundation (ASF, Delaware, USA), which also provides free support and software for open-source projects [55]. The fast and stable performance of the Apache server has been tested in many web-based applications [55]. Although servlets are located on the web server, the web server still needs to interact with the servlets through a servlet engine. ASF also provides a free servlet engine called TOMCAT, which is the official reference implementation for Java Servlet technologies [55]. In the Tele-Electrogastrography system, the TOMCAT container was integrated into the Apache Web Server. The Apache Web Server was configured to forward the POST request to TOMCAT. Upon an HTTP POST request, the TOMCAT invokes the servlet automatically. The Apache Server and the TOMCAT were both deployed on MS Windows platform.
4.7. System testing

Compared to conventional applications, web-based applications have many more users accessing the system simultaneously. Moreover, different web browsers, operating platforms and deployment environments exist among the potentially numerous web clients. All of these factors make web-based distributed applications much more complicated and difficult to test [56]. Thus, it is critical to have the web applications extensively tested to determine whether they function and perform adequately, and whether they are compatible across different browser and operating system configurations. A few aspects should be considered when testing web applications, which include interactions between HTML pages, Internet connections, Java applets and server-side programs. In the Tele-Electrogastrography system, several types of testing were performed which are described below.

4.7.1. Compatibility testing

Compatibility testing was performed to ensure the web application could perform as expected across multiple operating systems and browser configurations [56].

In the Tele-Electrogastrography system, the HTML and the Java applet interface was extensively tested using two major browsers: Netscape 4.7 and Internet Explorer 5.0 which were configured on different operating platforms including MS Windows 95/98, MS Windows NT and MS Windows 2000.

4.7.2 Function testing

The purpose of function testing is to ensure that the web application performs and functions properly according to design specifications [56]. In the Tele-
Electrogastrography system, the Java applet and the database access were fully tested to ensure that they function as expected.

The Java applet has been tested to display correct T-F plots, PDF plots and statistical results. Initially, the Java applet was tested in a standalone mode using a JVM interface such as "APPLETVIEWER". After that, it was examined within the completed web page from a user perspective.

The EGG database access was tested to ensure that the EGG data were correctly formatted, displayed and inserted. In addition, database security was also tested to ensure that only those with the required privilege levels could view and manipulate the EGG data.

4.7.3. Regression testing

Regression testing checks whether changes introduce errors in the web application [56]. In the Tele-Electrogastrography system, regression testing was performed every time new functions or methods were created and/or modifications or updates to the existing system were made, thus preventing the occurrence of new errors and bugs.

4.7.4. Performance testing

In order to check whether multiclient access was supported correctly, Tele-Electrogastrography was run on 4 client machines at the same time. These clients accessed the system over the Internet and performed various system operations. Moreover, the clients connected to the Internet through different bandwidth connections including Asymmetric Digital Subscriber Line (ADSL), cable and modem, which were used to test three important factors for evaluating EGG system performance: (1) the
downloading time of the Java applet; (2) the elapsed time for updating and/or building T-F plots between cases; and (3) the uploading time for data files. Test results indicated that system functions executed correctly and smoothly in a multi-access environment, and the speed for downloading the Java applet, plot updating, and file uploading were satisfactory, particularly when utilizing ADSL and high speed cable connections.
CHAPTER FIVE: RESULTS

5.1. System overview

The Tele-Electrogastrography system was developed using the programming environment and the graphical interface builder of IBM Visual Age For Java (Version 3.02, IBM Corporation, USA). The system provides EGG clinicians and researchers with several important Internet-based functionalities, which are listed below:

- Time/Frequency plots;
- Statistical evaluation of the T-F points (the points building the T-F plots) [4], including mean values, standard deviations and probability density functions;
- Categorization according to the statistical evaluation, and the introduced criteria for distinguishing normal from abnormal EGG;
- Association of EGG recordings with pre-diagnosed gastrointestinal motility disorders.

Tele-Electrogastrography system can be executed in different web browsers configured on different platforms ranging from MS Windows 95/98, and MS Windows NT, to MS Windows 2000. It can support multi-user access over the Internet regardless of the place and time. It takes less than 8 seconds for users to download Java applet and less 14 seconds for users to upload files on high-speed Internet connections. For low bandwidth connections such as a modem link, these parameters were obviously lower because of the narrowed bandwidth over the Internet.
5.2. Observers

A Java applet interface was provided for observers to examine the T-F values and their statistical analysis including their means, standard deviations and probability density functions. Using this interface, the viewers can choose to observe different patterns of EGG abnormalities and cases. After a pattern and a case are chosen, the relevant parameters used to obtain the T-F values are displayed, e.g. the sampling frequency, the numbers of points used for the FHT/FFT transforms, and the percent overlap. Then the observers can examine also the T-F plots, check the statistical values and the probability density functions by clicking corresponding buttons on the interface. The interface also provides an option to view single channel or multi-channel T-F and PDF plots by checking or unchecking a box shown on the top of the applet interface.

5.3. Active participants

The Tele-Electrogastrography system provides an option for active participants to submit, analyze, examine and classify their own EGG records. This feature will automatically classify the records into different patterns and store the records in our database. The EGG records are provided in the form of ASCII files with an extension ".dat", containing power spectral values in a predefined format, which is described to the respective participant. The data file can be generated using standard data acquisition software. Along with the file that is to be submitted, several recording parameters related to the technique of calculating the spectral values need to be provided by the participant. The parameters include the sampling frequency, the numbers of points of the FFT / FHT transforms, the % overlap and the duration of the recording. The primary and the secondary diagnosis for the patient associated with a specific EGG record should also be
provided. The data file can be selected, attached and submitted by pressing the corresponding buttons on the interface. It is crucial for the user to provide the correct recording parameters, because the system will reject files if the number of spectral values does not correspond to the submitted parameters. For example, a record containing 28672 spectral values corresponds to the following parameters: sampling frequency 2 Hz, 512 FFT/FHT points, 0% overlap and 60 minutes duration of the recording. Thus, active participants should follow the specified format to arrange the data in the files. The format can be summarized as follows:

Spectra of each standard EGG channel should be arranged in ascending order starting with Channel 1 and ending with Channel 8. Spectral values in each spectrum should be separated by space and arranged in ascending order of the sampling frequency step. The total number of data in the submitted files can be obtained using the formula:

\[
\text{Total Number} = \left\lceil \frac{D \times 60.0}{(M \times Ts) / (1 - OV / 100)} \right\rceil \times \frac{M}{2} \times 8 \quad [5.31]
\]

where \( D \) is the duration of the recording in minutes, \( M \) is the number of points for the FFT/FHT, \( OV \) is the percent overlap, and \( Ts = 1/Fs \) represents the sampling interval.

After a file is successfully submitted, the system classifies the record and a confirmation page is displayed with the determined EGG pattern and case number. The case number reflects a registration in the database. The active participant can then follow the provided instructions to enter the Tele-Electrogastrography system as an observer and examine the T-F plots, the statistical values and the PDF plots of the newly submitted and classified EGG record.
5.4. Study of classified normal and abnormal EGG patterns

The signals of all cases studied were acquired with a 10 Hz sampling frequency. For the offline analysis, the sampling frequency was reduced to 2 Hz after digital filtering with sampling filters in the frequency band 0.02-0.1 Hz.

5.4.1. Normal electrical activity – Normal EGG

Typical time-frequency plots, basic averages and probability density functions obtained from a volunteer with normal gastric electrical activity (Normal EGG) are displayed in Figures 27-29 respectively. The time-frequency plots were stable. The standard deviation of 7 channels (Channel 1, 2, 3, 4, 6, 7, 8) are smaller than 0.45 cpm and the maxima of the probability density function coincide at frequency 3.05 cpm in the normal range of 2.5-3.75 cpm.

Figure 27. Time-Frequency plot of normal EGG with frequency resolution of 0.23 cpm, time intervals of 4.27 minutes and 0% overlap
Figure 28. Basic averages obtained from a volunteer with normal gastric electrical activity

Figure 29. Probability Density Function plots with frequency resolution of 0.23 cpm in the frequency range 1-9 cpm. The maxima coincide at 3.05 cpm
5.4.2. Abnormal electrical activity – Pattern 1

Typical time-frequency plots, basic averages and probability density functions plots obtained from a patient with abnormal gastric electrical activity (Pattern 1) is shown in Figures 30-32. 7 EGG channels are unstable. The maxima coincide in the normal range.

Figure 30. Time-Frequency plot recorded from a patient with abnormal GEA (Pattern 1).
<table>
<thead>
<tr>
<th>Channel1</th>
<th>Mean</th>
<th>Variance</th>
<th>Std Deviation</th>
<th>RMS</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel2</td>
<td>2.90</td>
<td>0.17</td>
<td>0.41</td>
<td>2.93</td>
<td>2.93</td>
</tr>
<tr>
<td>Channel3</td>
<td>2.66</td>
<td>0.22</td>
<td>0.47</td>
<td>2.70</td>
<td>2.81</td>
</tr>
<tr>
<td>Channel4</td>
<td>2.70</td>
<td>0.32</td>
<td>0.57</td>
<td>2.75</td>
<td>2.81</td>
</tr>
<tr>
<td>Channel5</td>
<td>2.91</td>
<td>0.74</td>
<td>0.86</td>
<td>2.65</td>
<td>2.81</td>
</tr>
<tr>
<td>Channel6</td>
<td>3.16</td>
<td>0.53</td>
<td>0.73</td>
<td>3.25</td>
<td>2.93</td>
</tr>
<tr>
<td>Channel7</td>
<td>3.62</td>
<td>1.76</td>
<td>1.33</td>
<td>3.85</td>
<td>2.81</td>
</tr>
<tr>
<td>Channel8</td>
<td>3.10</td>
<td>0.58</td>
<td>0.76</td>
<td>3.19</td>
<td>2.81</td>
</tr>
</tbody>
</table>

Figure 31. Basic averages obtained from a patient with abnormal GEA (pattern 1).

![Probability Density Function Plot](image)

Figure 32. Probability Density Function plots of a patient with abnormal EGG (pattern 1).

The maxima coincide at 2.78 cpm
5.4.3. Abnormal electrical activity – Pattern 2

Typical time-frequency plots, basic averages and probability density functions obtained from a patient with abnormal gastric electrical activity (Pattern 2) are shown in Figures 33-35 respectively. Time frequency plots are unstable. Standard deviations calculated from these plots are greater than 0.45 cpm in 7 out of 8 EGG channels. The maxima of the probability density functions do not coincide.

![Time Frequency Plot](image)

Figure 33. Unstable Time-Frequency plots obtained from a patient with abnormal EGG (Pattern 2).
Figure 34. Basic averages calculated from a patient with abnormal EGG (Pattern 2).

Figure 35. Probability Density Function plots (Pattern 2). The maxima do not coincide.
5.4.4. Abnormal electrical activity – Pattern 4

Figure 36, Figure 37 and Figure 38 show the time-frequency plots, basic averages and probability density functions obtained from a patient. This case was classified as abnormal EGG (Pattern 4). It is very obvious that the time-frequency plots are stable. However, the PDFs of more than three EGG channels coincide at 3.95 cpm, which is outside the normal frequency range of 2.5-3.75 cpm.

![Figure 36](image-url)

Figure 36. Time-Frequency plots recorded from a patient with abnormal EGG (Pattern 4).
Figure 37. Basic averages calculated from a patient with abnormal EGG (Pattern 4). Five out of the 8 channels are stable.

Figure 38. Probability density functions calculated from a patient with abnormal EGG (Pattern 4) Note that the maxima coincide at 3.95 cpm, which is outside the normal frequency range.
5.4.5. **Abnormal electrical activity – Pattern 5**

Figure 39, Figure 40 and Figure 41 show the time-frequency plots, basic averages and probability density functions obtained from a patient. This case was classified as abnormal EGG (Pattern 5). It is very obvious that the time-frequency plots are unstable. The PDFs of more than three EGG channels coincide at 2.11 cpm, which is outside the normal frequency range of 2.5-3.75 cpm.

![Time Frequency Plot](image)

Figure 39. Time-frequency plots acquired from a patient with abnormal EGG (Pattern 5)
Figure 40. Basic averages calculated from a patient with abnormal EGG (Pattern 5)

<table>
<thead>
<tr>
<th>Channel</th>
<th>Mean</th>
<th>Variance</th>
<th>Std Deviation</th>
<th>RMS</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel 1</td>
<td>2.28</td>
<td>0.51</td>
<td>0.72</td>
<td>2.39</td>
<td>1.88</td>
</tr>
<tr>
<td>Channel 2</td>
<td>2.26</td>
<td>0.55</td>
<td>0.74</td>
<td>2.38</td>
<td>2.11</td>
</tr>
<tr>
<td>Channel 3</td>
<td>2.01</td>
<td>0.13</td>
<td>0.36</td>
<td>2.04</td>
<td>1.99</td>
</tr>
<tr>
<td>Channel 4</td>
<td>1.98</td>
<td>0.49</td>
<td>0.70</td>
<td>2.10</td>
<td>1.88</td>
</tr>
<tr>
<td>Channel 5</td>
<td>2.09</td>
<td>0.29</td>
<td>0.53</td>
<td>2.16</td>
<td>1.99</td>
</tr>
<tr>
<td>Channel 6</td>
<td>2.14</td>
<td>0.26</td>
<td>0.51</td>
<td>2.20</td>
<td>2.11</td>
</tr>
<tr>
<td>Channel 7</td>
<td>2.38</td>
<td>0.56</td>
<td>0.75</td>
<td>2.49</td>
<td>2.23</td>
</tr>
<tr>
<td>Channel 8</td>
<td>2.33</td>
<td>0.40</td>
<td>0.64</td>
<td>2.41</td>
<td>2.23</td>
</tr>
</tbody>
</table>

Figure 41. Probability density functions calculated from a patient with abnormal EGG (Pattern 5) Seven EGG channels are unstable. The maxima coincide, but outside the normal range.
5.5. Classification of EGG recordings with pre-diagnosed GI disorders

Currently, 191 cases of EGG recordings from volunteers and patients have been classified and stored in the EGG database. In the future, growing worldwide evidence for the clinical utility of EGG could be collected and expanded. As the accumulation of EGG recordings in the database grows, more reliable relationship between EGG patterns and the available diagnostic information could be found using the described classification methodology. Based on the current EGG data available in the database, the relationships found between EGG patterns and pre-diagnosed disorders are depicted in Figure 42. The data collected so far from the Province of Alberta (Canada) indicate that abnormal EGG patterns are observed over a wide spectrum of gastrointestinal motility abnormalities. Of particular interest might be the notable prevalence of EGG abnormalities (Patterns 1, 2 and 5) in Dyspepsia, Gastroesophageal Reflux (Patterns 1, 2, 4 and 5), Post-Fundoplication (Patterns 1 and 5), Unexplained Nausea and Vomiting (Pattern 1), and Intestinal Pseudo-Obstruction (Patterns 1 and 5). It should be noted, however, that as a result of the pattern classification of abnormal EGG recordings and their association with various pre-diagnosed gastrointestinal motility disorders, in some disease groups only a few cases are present. This underlines once again the necessity of a world-wide depository for pre-diagnosed EGG recordings, so that the case numbers in a given disease group reach statistical significance.
5.6. Preliminary evaluation of sensitivity and specificity of EGG based on the data extracted from the Tele-EGG system

Based on the data submitted to the Tele-EGG system (total of 191 records), the overall of the Tele-EGG test sensitivity was calculated according to the expression:

\[
\text{Sensitivity} = \frac{\text{NR}_{\text{EGG}}}{\text{NR}} \times 100, \%
\]  \[5.61\]

where \(\text{NR}_{\text{EGG}}\) is the number of records with an abnormal EGG pattern and \(\text{NR}\) is the total number of abnormal records as determined by an objective independent diagnosis. Specificity was calculated as

\[
\text{Specificity} = \frac{\text{HR}_{\text{EGG}}}{\text{HR}} \times 100, \%
\]  \[5.62\]

where \(\text{HR}_{\text{EGG}}\) is the number of records with a normal EGG pattern as identified by the telemedical system, and \(\text{HR}\) is the total number of obtained from symptom-free normal
volunteers[57].

Although in many of the studied disorders the case numbers did not reach statistical significance (>30, [58]), the following table presents some preliminary data on the specificity of the standard 8-channel EGG test.

<table>
<thead>
<tr>
<th>Disorders</th>
<th>Sensitivity [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Achalasia</td>
<td>50</td>
</tr>
<tr>
<td>Anorexia Nervosa</td>
<td>0</td>
</tr>
<tr>
<td>Constipation</td>
<td>25</td>
</tr>
<tr>
<td>Dumping Syndrome</td>
<td>50</td>
</tr>
<tr>
<td>Dyspepsia</td>
<td>65</td>
</tr>
<tr>
<td>Esophageal Spasm</td>
<td>33</td>
</tr>
<tr>
<td>Gas Bloating Syndrome</td>
<td>0</td>
</tr>
<tr>
<td><strong>Gastroesophageal Reflux</strong></td>
<td>60</td>
</tr>
<tr>
<td>Gastroparesis</td>
<td>22</td>
</tr>
<tr>
<td>Intestinal Pseudo-Obstruction</td>
<td>100</td>
</tr>
<tr>
<td>Post Fundoplication</td>
<td>75</td>
</tr>
<tr>
<td>Post Gastric Surgery</td>
<td>100</td>
</tr>
<tr>
<td>Scleroderma</td>
<td>50</td>
</tr>
<tr>
<td>Unexplained Burping</td>
<td>50</td>
</tr>
<tr>
<td>Unexplained Nausea</td>
<td>67</td>
</tr>
</tbody>
</table>

Table 4. Disorder-based sensitivity obtained after categorizing the 191 submitted records. Statistically significant number of cases per disorder are denoted with bold.
As seen in Table 3, from these very preliminary data it appears that the EGG test is with low sensitivity (<50%) for disorders such as Anorexia Nervosa, Constipation, Esophageal Spasm, Gas Bloating Syndrome and Gastroparesis. It is worth mentioning that in more than one instance, EGG has been suggested as a possible diagnostic tool for these disorders. Among the various other disorders, only Gastroesophageal Reflux cases reached statistically significant number (over 30). However, the sensitivity of the EGG test for this disease is also low, around 60%. On the other hand, the EGG test exhibits good specificity of about 87%, which is further enhanced by the very low sensitivity of the test to disorders of the lower gastrointestinal tract such as Constipation and Gas Bloating Syndrome. This indicates, that when an EGG record is classified as normal by the developed system, it most probably is. The sensitivity is high for post-operatively detected EGG abnormalities (e.g. in Post Fundoplication and Post Gastric Surgery cases), which is another indirect evidence that the recordings legitimately reflect EGG abnormalities, since the latter have been well documented post-operatively. Nevertheless, it should be noted that these post-surgical abnormalities are not relevant from diagnostic point of view because these records are obtained from patients in post-operation recovery, whose stomachs did not perform their function well anyway.

In some disease groups, including Dyspepsia, Intestinal Pseudo-Obstruction and Unexplained Nausea, the EGG test showed high sensitivity (>65%). However, only a few cases have been classified until the completion of this thesis, which underlines once again the fact that statistically significant number of cases are needed, in order to objectively show the reliability of multichannel EGG as a possible diagnostic test for these disorders.
With the completion of the Tele-EGG system, conditions now exist for the collection of EGG-tested disorder cases world-wide in a far more comprehensive fashion.

5.7. Additional limitations and features

The speed of data transmission through the Internet can largely affect the performance of any web-based system. Although there are many factors which can lead to slow connections (e.g. traffic, time of the day and server capability), the bandwidth limitation is the main obstacle for a large amount of data to be transferred fast over the Internet [31]. Currently, most users connect to the Internet via public telephone networks and narrowband Integrated Services Digital Network (ISDN) which cause slow data transmission and file downloading because of their limited bandwidth [31]. The ever-increasing number of Internet connections can also result in low speed even for a fast cable connection. For Tele-Electrogastrography, low speed links can result in slow downloading of the Java applet which can lead to longer waiting time after users change to the option of viewing the T-F graphical presentation of different cases. Thus, for better viewing the presentation of stored EGG data, a fast and wide bandwidth link to access the system is recommended.

Another issue that needed to be addressed is the missing RMI feature of the MS Internet Explorer. Without it, Internet Explorer users cannot adequately view the T-F plots, their statistics and the PDF plots. To overcome this problem, the users need to download Java plug-in kit from the web site of Sun Microsystems, Inc (www.sun.com) and install it on their machine. Because downloading the Java Plug-in kit will take about 20 minutes for average Internet connections, this is not very convenient for clinicians.
and researchers. A small, handy and downloadable program, which can be executed to add the necessary RMI class files into the Internet Explorer Java Virtue Machine (JVM) on user machines was developed using Visual Basic (Microsoft Corp, Redmond, WA) and Windows API in our system. The program was embedded in the system website (mint8.enel.ucalgary.ca). Instead of downloading the Java Plug-in kit, users can download this program from the EGG system web site and install it only in a few easy clicks of the mouse. The program does not affect the performance of the user system, and is invisible to applications which do not utilize RMI.

5.8. Conclusions

In this study, we report the successful implementation of a complex web-based distributed system for classification, storage and analysis of human electrogastrograms. This system provides a powerful and easy to use tool for training clinicians and researchers how to reliably record and adequately interpret human electrogastrograms using the Internet. Predefined EGG abnormalities have been described and stored in this system for worldwide reference. Biomedical engineers and medical doctors can greatly benefit from this system for training in EGG equipment, recording procedures, signal processing and data interpretation. Classified abnormal EGG patterns that are associated with pre-diagnosed gastric disorders can be stored in the EGG database. Thus, growing worldwide evidence for the clinical utility of EGG could be collected, maintained and expanded. The system can be regarded as an experimental tool for more reliably relating abnormal EGG patterns to certain clinical conditions, which facilitates the transformation of EGG into a routine clinical procedure in the near future. At the moment, the system has been used by GI Motility Laboratory in Edmonton (Alberta), Cleveland Clinic(OH)
and the Institute of Physiology in St. Petersburg (Russia).
REFERENCES


Appendix

The Deployment of Tele-Electrogastrography Server

The deployment of Tele-Electrogastrography includes the installation and execution phases.

1 Installation:

1.1. Apache web server installation and configuration

A. Insert the backup CD disk into the CD–ROM. Run the installation file apache_1.3.19-win32-no_src-r2.msi which is in the directory \installationfiles on the CD disk and install it to the directory: C:\Program Files\Apache Group\Apache. After Installation, copy ApacheModuleJserv.dll from directory \installationfiles on CD disk to the directory C:\Program Files\Apache Group\Apache\Modules.

B. Create a new directory named C:\apache on the C drive and copy the “policy” file from the directory \installationfiles on the CD disk to C:\apache.

C. Copy all files and folders which are under the directory \htdocs on the CD disk to directory C:\Apache Group\Apache\htdocs.

D. Open the file httpd.conf under the directory C: \Program Files\Apache Group\Apache\conf using “notepad”. Add “Include "C:\tomcat\conf\tomcat-apache.conf"” to the end of the file, save and close the file.
1.2. Java 2 SDK installation

Run the installation file `j2sdk-1_3_0_02-win.exe` which is under the directory `\installationfiles` on the CD disk and install it to the directory `C:\ jdk1.3.0_02`. Click the **Start** button at the windows toolbar, point to the **Settings**, click **Control Panel** and then double click the **System** icon. When a pop-up dialog box appears (See Figure 1), click the **Advanced** tab and click button **Environment Variables**. A dialog box appears (See Figure 2). Click **Edit** button at the bottom of dialog and another dialog pops up (See Figure 3). Add `C:\jdk1.3.0_02\bin` to the **Variable Value** box and Click **Ok** to dismiss the dialog. Click the **New** button at the bottom to create a new system variable named **JAVA_HOME** and set the value as `C:\jdk1.3.0_02`. (See Figure 4). Copy the file "**java1.bat**" from the CD disk to the C:\ root directory.

(Figure 1)
Environment Variables

User variables for mint18

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEMP</td>
<td>C:\Documents and Settings\Administrat...</td>
</tr>
<tr>
<td>TMP</td>
<td>C:\Documents and Settings\Administrat...</td>
</tr>
</tbody>
</table>

System variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS</td>
<td>Windows_NT</td>
</tr>
<tr>
<td>OS2LibPath</td>
<td>C:\WINNT\system32\os2\dlls</td>
</tr>
<tr>
<td>Path</td>
<td>C:\WINNT\system32;C:\WINNT;C:\W...</td>
</tr>
<tr>
<td>PATHEXT</td>
<td>.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;...</td>
</tr>
<tr>
<td>PROCESSOR_AR...</td>
<td>x86</td>
</tr>
</tbody>
</table>

(Figure 2.)

Edit System Variable

Variable Name: Path
Variable Value: ROUP\APACHE;C:JDK1.3.0_02;BIN C:\tor

(Figure 3.)

Edit System Variable

Variable Name: JAVA_HOME
Variable Value: C:JDK1.3.0_02

(Figure 4.)
1.2.1. Tomcat installation and Configuration

A. The next step is to run the Tomcat installation program. Run the installation file *jakarta-tomcat-3.2.1.zip* which is under the directory \installationfiles on the CD disk and unzip it to C:\. Rename C:\jakarta-tomcat to C:\tomcat. Click the Start button at the windows toolbar, point to the Settings, click Control Panel and then double click the System icon. When a pop-up dialog box (See Figure 1), click the Advanced tab and click button Environment Variables. A dialog box appears (See Figure 2). Click New button at the bottom. When the dialog box pop up, create a new system variable named TOMCAT_HOME and set the value as C:\tomcat. (See Figure 5)

![Edit System Variable](image)

(Figure 5.)

B. Create a sequence of directory under C:\tomcat which is depicted below:

C:\tomcat\webapps\gasServlet\WEB-INF\classes\gasServlet

Copy all files and folders from the directory \gasServlet on the CD disk to the directory C:\tomcat\webapps\gasServlet\WEB-
Open the file `server.xml` under the directory `C:\tomcat\conf` using "notepad" and add the following text in the file:

```xml
Context path="/gasServlet"
docBase="webapps/gasServlet"
crossContext="false"
debug="0"
reloadable="true">
</Context>
```

### 1.4 Database Configuration

**A** Open the Windows Control Panel. Double-click on the **ODBC**

**Data Sources** icon

**B** The **ODBC Data Source** Administrator appears. Click on the **User**

**DSN** tab. (See Figure 6.)
C Click Add to add a new data source

D The **Create New Data Source** dialog appears. Select the **Microsoft Access Driver** and click **Finish**. (See Figure 7.)
The ODBC Microsoft Access Setup dialog appears. Type "gasDatabase" in the Data Source Name text box. (See Figure 8). Click the Select button. A Select Database dialog appears (see Figure 9). Choose the gasDatabase.mdb from the directory C:\Apache Group\Apache\htdocs and Click Ok to dismiss the dialog.
F Click the **Advance** button (See Figure 8). A **Set Advance Option** dialog appears. (See Figure 10). Type "zhao" in Login text box and "as05" in **Password** text box and Click **Ok** to dismiss the dialog.

![Set Advanced Options](image)

*(Figure 10.)*

G Click **OK** to dismiss the **ODBC Microsoft Access Setup** dialog. (See Figure 8).

H Click **OK** to dismiss the **ODBC Data Source Administrator** and close the **control panel**. (See Figure 6).
2 Execution:

2.1. Start Tomcat

Open an MS-DOS console window. Go to the directory C:\tomcat\bin and run command `tomcat.bat -run`

2.2. Shut down Tomcat

Open an MS-DOS console window. Go to the directory C:\tomcat\bin and run the file `shutdown`

2.3. Start Apache web server

Select the "Start Apache as console app" option from the Start menu. This will open a console window and start Apache web server. The window will remain active until you stop Apache.

2.4. Shut down Apache web server

Open an MS-DOS console window. Under DOS prompt, type "`apache -k shutdown`". Then press "Enter"

2.5. Start RMI registry

Open an MS-DOS console window. Under DOS prompt, type "`start rmiregistry`". Then press "Enter".

2.6. Start RMI application server

Open an MS-DOS console window. Under DOS prompt, type "`java1`". Then press "Enter". A message, "DBManager bound in registry" will show on the computer screen.