Message System with J2EE and AJAX technology

BY
SU SEONG LIN

Open University Malaysia

(May 2007)

ABSTRACT

Ajax, which stands for **Asynchronous Javascript And XML,** is a combined technologies of combined of HTML + CSS for presenting information on to the web, JavaScript is for dynamically interacting change with the information presented at the client site and XML and XSLT object to manipulate data asynchronously with the web server.

A Messaging system allows staff to communicate with any department in the company by submitting messages via company internal system like Content Management System (CMS). This system implementing with AJAX technology so that it look and feel like desktop application needed no page refreshes, no flickering in the browser and no waiting. With Ajax, user is able to communicate with the server behind the scenes, grab the data and display it instantly in a web page.

TABLE OF CONTENTS

ABSTRACT 2

ACKNOWLEDGMENTS 3

DECLARATION 4

TABLE OF CONTENTS 5

TABLE OF FIGURES 7

CHAPTER 1: INTRODUCTION 9

1.1 INTRODUCTION 9

1.2 ABOUT THIS PROJECT 11

1.3 RESEARCH PROBLEMS(S) 12

1.4 OBJECTIVES 13

1.5 ABOUT THIS PROJECT 14

1.6 FUNCTIONAL AND TECHNICAL REQUIREMENT OF THE SYSTEM 15

CHAPTER 2: LITERATURE REVIEW 16

2.1 INTRODUCTION 16

2.1.1 AJAX BASED WEB APPLICATION 16

2.2 OLD TECHNOLOGY, NEW TRICKS 17

2.3 AJAX TECHNOLOGY 18

CHAPTER 3: RESEARCH METHODOLOGY 20

3.1 INTRODUCTION 20

3.2 METHOD 20

3.3 DEVELOPMENT METHODS 20

3.3.1 ITERATIVE AND INCREMENTAL DEVELOPMENT LIFE CYCLE 23

3.3.2 PHASES OF ITERATIVE DEVELOPMENT 23

3.4 MODELING TECHNIQUES 27

3.5 DEVELOPMENT TOOLS 27

3.6 AJAX AND THE REBIRTH OF CLIENT/SERVER 28

3.7 TRADITIONAL WEB DEVELOPMENT VS. AJAX 29

3.8 ANATOMY OF AN AJAX INTERACTION 29

3.9 SYSTEM WORKFLOW DIAGRAM 31

3.9.1 READ MESSAGES 31

3.9.2 CREATE NEW MESSAGE 32

3.9.3 SEND DRAFT MESSAGE 33

3.9.4 REPLYING MESSAGES 35

3.9.5 READ SENT MESSAGES 36

3.9.6 FORWARDING MESSAGES 37

3.9.7 SETTING AUTO-FORWARDING 38

3.9.8 E-BROADCAST 39

3.9.9 MAINTAIN RECIPIENTS 40

CHAPTER 4: SYSTEM ANALYSIS 42

4.1 FUNCTIONAL REQUIREMENT 42
4.2 NON FUNCTIONAL REQUIREMENT 43
CHAPTER 5: SYSTEM DESIGN 46
5.1 INTRODUCTION 46
5.2 SYSTEM AND SYSTEM SOFTWARE 46
5.3 SYSTEM ARCHITECTURE 47
5.4 APPLICATION SYSTEM DESIGN 48
5.5 USER INTERFACE DESIGN 49
5.5.1 READ MESSAGES 49
5.5.2 CREATE NEW MESSAGE 52
5.5.3 SEND DRAFT MESSAGE 56
5.5.4 READ SENT MESSAGES 64
5.5.5 E-BROADCAST 73
5.5.6 MAINTAIN RECIPIENTS 77
5.6 DATABASE DESIGN 80
5.6.1 INTRODUCTION 80
5.6.2 NORMALISATION 80
5.6.3 MODELS AND LANGUAGES 81
5.6.4 E-MESSAGING DATABASE PROPERTIES 82
5.7 SECURITY DESIGN 88
CHAPTER 6: IMPLEMENTATION PLAN 91
6.1 INTRODUCTION 91
6.2 SYSTEM TESTING 91
6.3 LIST OF PROCESSES 92
6.3.1 PROJECT PLANNING PHASE 96
6.3.2 PROJECT DESIGN PHASE 98
6.3.3 BUILD & UNIT TEST PHASE 99
6.3.4 QUALITY ASSURANCE AND INTEGRATED TEST PHASE 100
6.3.5 TRAINING AND PRODUCT PRODUCTION PHASE 102
6.3.6 CLOSE OUT PHASE 103
CHAPTER 7: DISCUSSION & CONCLUSION 104
7.1 AJAX INTERACTION 104
7.2 DISCUSSION ON PROBLEM STATEMENT 105
REFERENCES 106

TABLE OF FIGURES

Figure 1 - Software Requirement 15
Figure 2 - Iterative and Incremental Development life cycle 23
Figure 3 - Iterative development life cycle 24
Figure 4 - An AJAX Interaction Provides validation Logic 30
Figure 5 - Read Message Workflow 31
Figure 6 - Read Message Process & Description 31
Figure 7 - Create New Message Workflow 32
Figure 8 - Create New Message Process & Description 33
Figure 9 - Send Draft Message Workflow 33
Figure 10 - Send Draft Message Process & Description 34
Figure 11 - Replying Messages Workflow 35
Figure 12 - Replying Messages Process & Description 36
Figure 13 - Read Sent Message Workflow 36
Figure 14 - Read Sent Message Process & Description 36
Figure 15 - Forwarding Messages Workflow 37
Figure 16 - Forwarding Messages Process & Description 37
Figure 17 - Set Auto-Forward Workflow 38
Figure 18 - Set Auto-Forward Process & Description 38
Figure 19 - E-Broadcast Workflow 39
Figure 20 - E-Broadcast Process & Description 39
Figure 21 - Maintain Recipients Workflow 40
Figure 22 - Maintain Recipients Process & Description 41
Figure 23 - Servers flow 47
Figure 24 - Web applications layers 47
Figure 25 - System Architectural Design 48
Figure 26 - Package Diagram 48
Figure 27 - Message Listing Screen 50
Figure 28 - Message Display Screen 52
Figure 29 - New Message Detail 53
Figure 30 - Message Sent Confirmation 54
Figure 31 - Message saved Confirmation 55
Figure 32 - Draft Message Listing Screen 56
Figure 33 - Draft Message Detail 58
Figure 34 - Message Sent Confirmation 59
Figure 35 - Message Saved Confirmation 60
Figure 36 - Message Listing Screen 62
Figure 37 - Reply Message Details 64
Figure 38 - Sent Message Listing Screen 65
Figure 39 - Message Display Screen 66
Figure 40 - Recipient Allowed screen 67
Figure 41 - Message To field 67
Figure 42 - Inbox Listing Screen 68
Figure 43 - Message Display Screen 70
Figure 44 - Auto Forward Details Screen 72
Figure 45 - Auto-Forwarding Setting Confirmation Screen 73
Figure 46 - E-Broadcast details screen 74
Figure 47 - Select Recipients screen 75
Figure 48 - Broadcast message Sent Confirmation screen 76
Figure 49 - Select recipient DEPARTMENT screen 77
Figure 50 - Enter recipient Information screen 78
Figure 51 - Data Saved Confirmation screen 79
Figure 52 - P_AMS_USER database table 83
Figure 53 - P_AMS_MESSAGE database table 85
Figure 54 - P_AMS_RECIPIENTS database table 86

Figure 55 - S_AMS_AUDIT database table 86
Figure 56 - S_AMS_DEPARTMENT database table 87
Figure 57 - S_REF_VALUES database table 87
Figure 58 - Security control for multi-server 88
Figure 59 - Implementation lifecycle 94
Figure 60 - Implementation Process 95
Figure 61 - Project Plan Phase 97
Figure 62 - Project design Phase 98
Figure 63 - Build & Unit Test phase 100
Figure 64 - Quality Assurance and Integrated test phase 101
Figure 65 - Close out phase 103

CHAPTER 1: INTRODUCTION

1.1 Introduction

Making Web applications look and feel like desktop applications is what this project's all about – that's what Ajax does. Although Web development is getting more and more popular, users still experience the nasty part of having to click a button, wait until a new page loads, click another button, and wait until a new page loads, and so on.

With Ajax, you communicate with the server behind the scenes, grab the data or information users requested and display it instantly in a Web page – no page refreshes needed, no flickering in the browser, no waiting. That's what the research about; at last it lets Web applications start to look like desktop applications. Web messaging application software can have the same look and feel of software on the user's desktop.

A traditional desktop application. Messaging system with J2EE technology with embedded AJAX technology, users will realize that a new breed of dynamic web applications is emerging. These applications look and act very similar to traditional desktop applications without relying on plug-ins or browser-specific features. Web messaging applications have traditionally been a set of HTML pages that must be reloaded to change any portion of the content. Technologies such as JavaScript programming language and cascading style sheets (CSS) have matured to the point where they can be used effectively to create very dynamic web applications that will work on all of the major browsers. Any web application with enable AJAX techniques to be more rich and interactive like desktop applications.

By using JavaScript technology, an HTML page can asynchronously make calls to the server from which it was loaded and fetch XML documents. The XML