

WORKFLOW PROCESS THROUGH WAP

By

Bushra Saleem

A report submitted to the
Department of Computer Science, Quaid-i-Azam University
as a partial fulfillment of the requirement
for the award of the degree of M.Sc. in Computer Science

January 2002



DISS
COM
1252

QUAID-I-AZAM UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

Dated: March, 2002

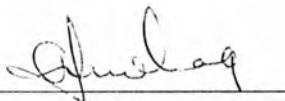
FINAL APPROVAL

This is to certify that we have read the project report submitted by Ms. Bushra Saleem and it is our judgement that this report is of sufficient standard to warrant its acceptance by the Quaid-i-Azam University, Islamabad for the degree of Master of Science in Computer Science.

COMMITTEE:

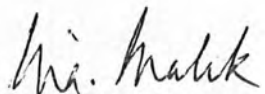
1. EXTERNAL EXAMINER

Dr. A. F. M. Ishaq
COMSATS Institute of Information Tech.
Sector H-8/1,
Islamabad.



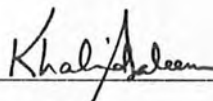
2. SUPERVISOR

Dr. Masud Ahmad Malik
Professor
Dept. of Computer Science
Quaid-I-Azam University
Islamabad.



3. INCHARGE

Mr. Khalid Saleem
Incharge
Deptt. Of Computer Science
Quaid-I-Azam University
Islamabad.



ACKNOWLEDGEMENTS

All praise is to the Almighty, The Most Gracious and The Most Merciful, Who has been the real source of all my accomplishments. I am insatiably thankful to Him for the ability, the chance and the courage He gave me through my parents, sisters and brother to fulfill my obligations.

This work was not possible without the help and encouragement from number of persons. Among them Mr Waqar Ahmed, software developer at Ultimous, not only offered me the project but also was the motivating factor during the highs and the lows. His constant encouragement and advice have gone a long way in materializing the project.

My project supervisor, Dr. Masud Malik, deserves more than a customary acknowledgement for his critical appreciation, invaluable help and suggestions throughout my project.

I am greatly in debt to my teachers, who offered guidance and warm support not only during the project but also throughout the two years of my studies at QAU. They were always very encouraging and often spared their precious time to counsel me.

I must mention my friends at the university for always being there for me whenever I needed them and bearing with me in all sorts of moods. I will always cherish the moments we spent together.

Bushra Saleem



PROJECT BRIEF

Project Title: WorkFlow Process Through WAP
Organization: Ultimus Pakistan
Undertaken By: Bushra Saleem
Supervised By: Dr. Masud Malik
Department of Computer Science
Quaid-i-Azam University, Islamabad.
Starting Month: August 2001
Completion Month: January 2002
Software Used: SQL Server, Wireless Markup Language, VB Script and Active
Server Pages 3.0, Ultimus Product
Operating System: MS Windows 2000 (Advanced Server)
Machine Used: Pentium III

ABSTRACT

This project relates to WAP (wireless Application Protocol), which is quite a new technology and is still in its evolutionary stage. My work is aimed to provide the Ultimus users a workflow facility using mobile phones through WAP.

As the number of WAP users are growing much more rapidly than the number of Internet users, so now is the time to determine what works for your company and what will meet the needs of your customers.

CONTENTS

Chapter1 Introduction to the project.....	1
1.1 Introduction.....	2
1.2 Project definition.....	3
1.3 Scope of the project.....	4
1.4 Objectives.....	4
1.5 Feasibility study.....	4
Chapter 2 The WAP technology.....	6
2.1 Introduction.....	7
2.2 Limitation of WAP.....	7
2.3 Basic terminologies.....	8
2.4 Difference between WEB and WAP.....	11
2.5 Importance of WAP.....	11
Chapter 3 Introduction toUP.Phone.....	12
3.1 Introduction.....	13
3.2 UP.Phone features.....	13
3.3 UP.Simulator modes.....	14
3.4 Entering text.....	15
Chapter 4 Requirement analysis.....	16
4.1 Introduction.....	17
4.2 Functional requirements.....	17
4.3 Non functional requirements.....	18
4.4 Requirement specification.....	18
4.5 Analysis modeling.....	21
Chapter 5 System design.....	28
5.1 Introduction.....	29
5.2 Data design.....	29
5.3 Architectural design.....	30

5.4	Interface design.....	35
5.5	Procedural design.....	36
Chapter 6 Implementation.....		38
6.1	Introduction.....	39
6.2	Features of the used languages.....	39
6.3	Modules.....	43
Chapter 7 Testing and Evaluation.....		45
7.1	Introduction.....	46
7.2	Test cases.....	47
7.3	Evaluation.....	48
References.....		49
Appendices.....		50
A	Site hierarchy diagram.....	51
B	Overview of WAP transaction.....	54
C	Syntax for defining WML deck.....	58
D	Configuring your web server.....	61
E	How to use WAP facility.....	63
F	Using UP.Simulator.....	65

Chapter 1

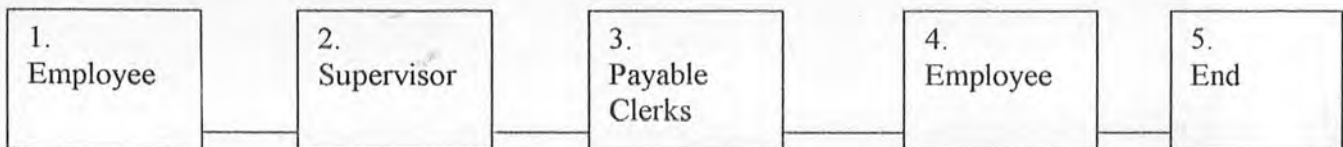
INTRODUCTION TO THE PROJECT

1.1 INTRODUCTION

This project has been taken for Ultimus, one of the leading software house in Pakistan. Ultimus is involved in the automation of various business transactions. For this, purpose it has released various versions of its product called ULTIMUS, which is used to automate different business transactions. This product has gained acceptance because of its good quality.

Ultimus is a Web-based Client/Server application that lets users design, simulate, implement, monitor and measure workflow for different administrative business processes such as Purchase Orders, Expense Reports, Change Orders and many more. Ultimus is easy to set up, it does not require any programming, scripting, or macros. Ultimus is easy to integrate with existing applications. It is easy to use, anyone with Web access and a Web browser can participate in workflow.

To demonstrate the capabilities of Ultimus, let us consider the example of expense approval, a very simple example is as follows,



(to notify)

The expense report process has the following rules:

1. Any employee of the company can write an expense report.
2. The employee's supervisor must approve every expense report.
3. All expense reports must go to the Payable Clerk for assigning accounting information, which also includes the check print date.
4. The employee must be notified of the check print date.

A "process map" is a graphical representation of how a process progresses (or "flows") from one step to another and what happens at each step. Ultimus implements a business workflow by breaking it down into simple steps and defining rules about when each step is invoked. Steps are tied together by "links". By linking the steps a user can define a process in the form of a flowchart, which is called a "process map." Ultimus lets us graphically create process maps.

For the expense report example, the process map consists of four steps as shown in the above figure. The first step is the "Employee" step. We can define a step for one employee by naming her the "recipient." In this case the "recipient" for the first step is the group "All Employees" which means that any employee of the company can initiate an expense report. Each step also has a form associated with it that the user completes to accomplish the tasks for that step. Forms are tailored for the unique needs of the tasks at each step.

The second step of the process is the "supervisor." It is tied to the first step by an unconditional link. This means that the expense report will always go to the supervisor of the employee who filled out an expense report. For the "Supervisor," the form consists of the Expense Report with an additional "Supervisor's Approval" signature.

The third step of the process is the "Payable Clerk" in the accounting department meaning that expense reports will always go from the "Supervisor" to the "Payable Clerk." The "recipient" for the "Payable Clerk" step consists of the payable clerk in the accounting department. The form for the step consists of the approved expense report with the check print date line for the payable clerk to enter.

Step 4 is the Employee step. This is to notify the employee who initiated the expense report that it has been approved and the date the check will be printed. This simple example provides with an introduction to the basic operations of Ultimus. Ultimus allows users to easily define and implement such processes.

1.2 PROJECT DEFINITION

My project concerns with providing this workflow facility through the WAP. The project is to initiate the Ultimus process through WAP (wireless application protocol) using the mobile phones. It has two major phases:

- I. Study of Ultimus workflow product and WAP.

- II. Initiating Ultimus workflow process of business automation, and also to access already initiated tasks and to make changes if required.

1.3 SCOPE OF THE PROJECT

The scope of the project is as follows,

- I. Connecting the mobile phone to the Ultimus server.
- II. Initiating a business transaction through the mobile phone by displaying the initiators input form on the micro browser and then submitting it after getting the input from the user. This will cause the transaction to begin.
- III. Accessing the Inbox tasks of the logged in person and submitting after making changes.
- IV. Checking the status of the task that was initiated.
- V. Displaying the output on the micro browser.

1.4 OBJECTIVES

Since Ultimus product is used world wide, so client satisfaction is a top priority. As the technology has advanced over the time, so have the facilities that they provide to their clients.

This software will to provide the facility to be able to initiate a business transaction without the restriction of having to sit in front of their computes screens. As WAP frees them to do so from anywhere, whether they are at home or in their cars, on their way to holiday. (Provided they are within the geographical constraints of their network providers). Thus meeting the challenges of today's technology.

1.5 FEASIBILITY STUDY

The feasibility study consists of following three main parts namely Economic feasibility, Technical feasibility, Operational feasibility.

ECONOMIC FEASIBILITY

The Ultimus product is provided by the organization and other required software can be downloaded from the Internet.

TECHNICAL FEASIBILITY

- I. All the services required for developing the system are available.
- II. The software development kit (phone.com) is easily available from the Internet for downloading. It's emulator helps to provide the real time interface for the application.
- III. The wireless access to the Ultimus process will free the clients from the constraints of the wires and desktop computers.
- IV. The phone.com has build in micro browser (UP browser).

OPERATIONAL FEASIBILITY

Since the number of WAP users are growing very rapidly, it will soon become inevitable for various organization to provide their clients with the new facility that WAP offers. Because of this very reason, many foresighted companies are working on it. Ultimus is one of them, who is going to use my software, in order to provide their clients with WAP facility.

Chapter 2

THE WAP TECHNOLOGY

2.1 INTRODUCTION

In a broad, general view, there's not much difference between "the web" and WAP(Wireless Application Protocol). Actually, WAP is just as much part of "the web" as your typical PC based browser such as Netscape or Internet Explorer. In this chapter, we use the word for normal web servers and web browsers "the web". If we think of "the web" as a concept, with its specifications and protocols explaining how things must operate in order to do what we want them to do, we can think of WAP as a concept too.

WAP is a set of specifications and protocols which explain how things should and must operate in order to be WAP. The difference between "the web" and WAP is of course that WAP is meant for wireless devices, including but not limited to mobile phones. In the wireless environment everything is simple and smaller. We have very narrow bandwidths, very little power from batteries to drive very small processors. Everything must be smaller and simpler. If we have followed the hype on WAP we will probably have heard that WAP will die when faster connections become available, such as GPRS, but this just isn't the case. These higher bandwidth technologies will only improve the services that will become available through WAP.

2.2 LIMITATIONS OF WAP

Unfortunately the WAP concept is plagued by much hype. Many people will no doubt think of WAP as the same as a "mobile phone from which you can surf the web", and this in a way is true, but not fully. WAP is very much in it's infancy, and expecting things, that we will probably going to end in disappointment. WAP is more or less at the evolutionary stage that "the web" was about five years ago.

In short, the limits of WAP today is plain text with some simple styling like **bold** and *italics*, but only on some devices. Monochrome images on displays that are typically 100 by 40 pixels. No audio, and no fancy animations. But this is today, and many companies are working hard on exciting new technologies to make WAP more interesting as anything one can imagine.

2.3 BASIC TERMINOLOGIES

WAP DEVICE

This term indicates the physical device that we use to access WAP applications and content. It doesn't necessarily have to be a mobile phone – it might be a PDA or a Handheld computer. More generally it is every WAP compliant device.

WAP CLIENT

In a network environment, a client is typically the logical entity that is operated by the user and communicates with the server entity. In the WAP world, a client is the entity that receives content from the Internet via a WAP gateway. This is usually (but not necessarily) the WAP browser. Commonly, WAP browser and WAP client are often used interchangeably.

WAP BROWSER

This is a software running on a WAP device that interprets the WAP content arriving from the Internet and decides how to display it on the screen of the WAP device. WAP browser are available for all WAP devices and are frequently referred to as **Micro browsers**.

WAP GATEWAY

A WAP gateway is a particular piece of software which is usually located on server of a mobile operator, and handles incoming requests from your WAP phone. It is responsible for compiling the WML pages into bytecode (WMLC) which can be understood by a WAP device. The WAP gateway also takes care of the conversion required during WTLS/SSL sessions. A WAP gateway is simply the piece of the chain that makes sure everything your WAP device receives is in a format it can understand. *(See a pictorial representation of mobile, gateways and application server in Appendix B)*

NETWORK OPERATOR

This is the company or organization that provides carrier services to its subscribers. As an example, the company you are paying your telephone bill to is your network

operator. A network operator enables you to mask calls to other phones from your telephone and, in addition provides you with different services.

BEARER SERVICES

These are the different ways that a mobile phone can communicate with the wireless network. To send and receive data from an application server, mobile phones have to establish some sort of connection with the WAP gateway. A bearer service is the method they use to do this. In GSM networks, for example we either use SMS (Short Message Service) or CSD (Circuit Switched Data). With the former bearer, the gateway has to divide the information that is to be sent to the phone into a lot of little messages (just like when you send text message to a friend using a mobile). With CSD, we communicate with the gateway using a data connection, which is not dissimilar to the way the modem in your computer communicated with the ISP.

GPRS NETWORK

GPRS is an upgraded GSM network that gives you the possibility to be always connected to the mobile Internet. You pay for the data you actually download, and not per minute you are connected to the net. The always on technology opens up a lot of brand new services, like better gaming opportunities and continuous email delivery. And most importantly it is cheaper.

I-MODE MOBILE INTERNET SERVICE

I-mode is a mobile Internet service from the Japanese tele provider NTT DoCoMo, started in February 1999. From being a pure Japanese service, i-mode is now expanding to other countries. The i-mode phone network is packet switched, so the users are charged for the amount of data they transfer, not the time they consume. The phones are always online (if they are within reach of the signals), and email can be pushed out to the recipients immediately. This makes email the most popular service. Other services are weather forecasts, ticket booking, online banking, stock trading, chatting etc.

CONTENT/ORIGIN/APPLICATION SERVER

These three names are used interchangeably. They denote the element that hosts the Internet content that is sent to clients when they make a request for it. A web server,

providing HTML content (but also WAP content if properly configured). (*To configure your application server so that it can accept WAP contents, see Appendix D*)

WBMP

WBMP stands for Wireless BitMaP. It is the default picture format for WAP. The current version of WBMP is called type 0. WBMP's are uncompressed, monochrome black/white bitmaps intended for use in devices with small screens and narrow bandwidth connection.

DECK & CARDS

Because mobile phones are not as powerful as computers, and certainly do not have as large screens, the code has to take on a special reduced form - a 'deck'. A deck is comprised of a series of 'cards'. Because mobile phones have very little memory, you have to display all of your information as little chunks. These are called cards. A group of these chunks (cards) is called a deck. A WAP enabled browser will deduce that the WML document is a deck, so all you have to do is tell it where the cards start and finish. Usually you would have between one and three cards. To implement a card, you need to use the card tag.

WML (Wireless Markup Language)

WML is a wireless markup language based on Extensible Markup Language (XML). XML, like HTML, is a Standardized Generalized Markup Language (SGML) variant. The WML language defines *elements* and *attributes* that let us specify the user interface components, called *cards* that users see on their browsers. Just as a Web browser can navigate from one page to another, the mobile phone browser can navigate from one card to another. The smallest unit of WML you can send to a mobile is a *deck*--one or more cards that a user is likely to access together. When a browser receives a WML deck, it displays the content defined in the first card and allows the user to respond. Depending on the card definition, the user can respond by entering text or choosing an option. (*The high-level syntax for defining a WML deck is given in Appendix C*)

2.4 DIFFERENCE BETWEEN WAP AND WEB

The WAP stack has many similarities to the Internet set of technologies. For instance, the Wireless Markup Language used to create WAP pages is very similar to HTML used to create WWW pages. Similarly, the WMLScript is based on JavaScript. It should be noted that both WML and WMLScript are adapted and optimized for a wireless environment (example compression to save bandwidth).

2.5 IMPORTANCE OF WAP

Consider that we are delivering content or, for instance, selling our products, via the Internet channel. We are also aware that the number of people using mobile access to the Internet (e.g. via GSM and their laptop or PDA) is increasing rapidly. We know that the number of mobile phone users is growing much more rapidly than the number of Internet users (there are more than 120 million GSM users today). Shouldn't these people have access to our services and products using WAP.

Here are some facts concerning WAP [1].

- It's designed to make the most out of current wireless technologies.
- It's rapidly being put to work by hundreds of companies that want to be on the leading edge of the transformation to ubiquitous Internet access.
- Now is the time to determine what works for your company and what will meet the needs of your customers.
- It isn't perfect, but it can get the job done.

Chapter 3

INTRODUCTION TO UP.PHONE

3.1 INTRODUCTION

While Nokia has gained some early market in Europe due to the timely release of the 7110 [2], but internationally, it is generally Phone.com who are supplying most of the browsers to phone manufacturers. That is for now many phones produced by third party manufacturers have and will be using the Phone.com browser. Example of some of the manufacturers currently using Phone.com browser are Alcatel, Motorola, Samsung, Mitsubishi.

Like Nokia, Phone.com offers a development kit for free download from its website, named the UP.SDK. The most important program included with it is the U.P simulator, which consists of a browser and an information window. Unlike Nokia, there is no WML editor supplied with the toolkit.

3.2 UP.PHONE FEATURES

Although UP.Phone user interfaces vary, all UP.Phones have the following features:

- I. A display with a minimum resolution of three lines of twelve fixed-width characters.
- II. Support for text entry. UP.Phone models typically allow users to enter text by pressing the number keys.
- III. A minimum of two programmable user interface mechanisms called *ACCEPT* and *OPTIONS*. Some UP.Phone models implement *ACCEPT* and *OPTIONS* with function keys; others implement them with a jog shuttle or context-dependent user gestures. Users activate the *ACCEPT* mechanism to acknowledge a message or confirm a choice; they activate the *OPTIONS* mechanism to display context dependent information or options.
- IV. A mechanism for backing up--called *PREV*. Many UP.Phones have a dedicated key for this function, typically labelled with a left arrow or the word *BACK*.

3.3 UP.SIMULATOR MODES

The UP.Simulator has two modes:

HTTP DIRECT

In this mode (the default configuration), the UP.Simulator loads WML directly from a Web server, bypassing the UP.Link Server.

UP.LINK SERVER

In this mode, the UP.Simulator interacts with an UP.Link Server just like a real UP.Phone. This mode requires access to an UP.Link Server, and you must register the UP.Simulator on the UP.Link Server before you can use it. You must use this mode to test UP.Link features that run on the server, for example, bookmarks, fax services, and notifications.

3.4 ENTERING TEXT

On a real UP.Phone, users enter text by pressing keys on the keypad. To enter text in the UP.Simulator, one can either use your computer keyboard or use the mouse to click on the keypad area of the screen. The cards that allow text entry display the current *text entry mode* above the OPTION function key. In *ALPHA* and *SMART* modes, each key enters different characters depending on the number of times you click it (or press it on a real UP.Phone). In *ALPHA* mode, the device cycles through the characters in the same order every time. In *SMART* mode, the device cycles through the characters based on your previous entries--in other words, it tries to guess the character you want using a word-recognition algorithm. (*The UP Simulator is shown in Appendix F*)

The following table identifies the characters for each key in the generic UP.Simulator:

Key	Characters	Key	Characters
1	1	8	T U V ü 8
2	A B C Å Ä Æ Ç 2	9	W X Y Z 9
3	D E F É 3	*	* . @ : / - _ ~
4	G H I 4	0	0 + \$ & % \$ £ ¥
5	J K L 5	#	# = () < > ! i
6	M N O Ñ Ö Ø 6	▶	space
7	P Q R S B 7	CLR	erases char to left

3.5 NAVIGATION KEYS

The default UP.Simulator keypad provides the following navigation keys.

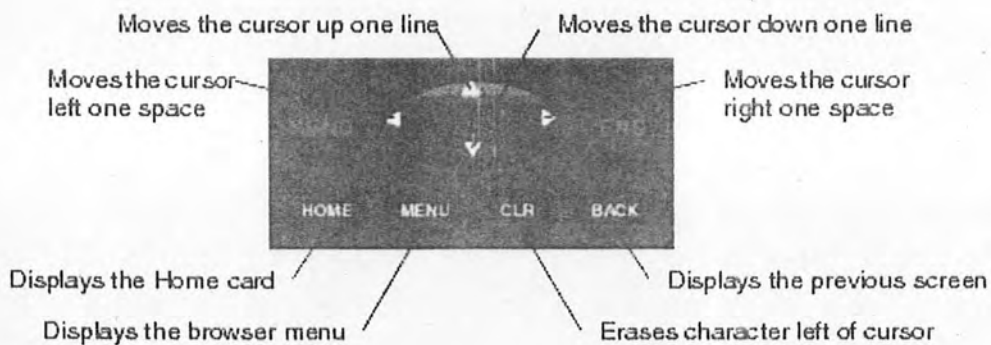


Figure . Generic UP.Simulator navigation keys reference[3]

Chapter 4

REQUIREMENT ANALYSIS

4.1 INTRODUCTION

Requirement analysis enables the system engineer to specify software function and performance, indicate software's interface with other system elements, and establish constraints that software must meet. Requirement analysis provides the software designer with models that can be translated in to data, architectural, interface and procedural design. Finally the requirement specification provides the developer and the customer with the means to assess quality once software is built.

The functional and non functional requirements of the system are given below.

4.2 FUNCTIONAL REQUIREMENTS

After analyzing the system, I have identified the following functional requirements

CONNECTING TO THE ULTIMUS SERVER

The mobile phone will be connected to the Ultimus server through the web server. The Web server will be configured so that it can accept the WML files.

Rationale: If the web server is not configured, it will not accept WML files.

ACCESSING THE SERVICES

Once the mobile owner is connected, a list of services that can be provided to him will be shown. The selection of the service will be the users responsibility.

Rational: As there can be more than one service in the list so the selection will be best done by the user, according to his requirements.

An appropriate form according to the selected service will be generated. The user will have to select or fill the forms as they appear, through mobile keypad and submit.

Rationale: To initiate the business transaction the input form has to be filled and then submitted. Also the desired information can be obtained by selecting the relevant option from the displayed list.

4.3 NON FUNCTIONAL REQUIREMENTS

After analyzing the system I, have identified the following functional requirements

Displaying of input and output forms using WML.

Rationale: Since WML uses low bandwidth than the conventional HTML Which is used for the displaying of web pages, therefore it is more suitable for the WAP (Wireless Application Protocol) purposes.

Phone.com mobile with its UP browser is used.

Rationale: It is generally Phone.com who are supplying most of the browsers to phone manufacturers all over the world [3]. That is for now many phones produced by the third party manufacturers are using Phone.com browser.

Minimum or no graphics will be included.

Rationale: Since graphics use lot of memory and mobile phones have a very limited capacity. Also some browser do not support graphics and some who support do not show a clear picture.

Short and meaningful text.

Rationale: Since the browser screens are very small as compared to the Computer screens therefore short and meaningful as one deck of WML Cards can be only up to 1400bits.

4.4 REQUIREMENT SPECIFICATION

The requirement specifications of the system are described using form-based approach. This approach relies on defining one or more standard forms or templates to express the requirements. The specification is structured around the functions performed by the systems, the system manipulations and the events it handles.

Function: connecting to Ultimus server

Description: First of all the user must be an owner of an internet enabled mobile phone, which he will configure according to his mobile network. Then to access the Ultimus services he must enter the Ultimus home page URL. After his authentication he is able to connect himself to the Ultimus server. But in case of an emulator just type in the URL.

Input: URL of Ultimus home page.

Source: User will provide input using mobile phone.

Output: Authentication form.

Destination: The output will go to the database from there it will be checked.

Requires: Mobile phone with Internet facility.

Precondition: The mobile phone should be configured to its mobile gateway but in case of emulator mobile phone should be configured to web server (IIS), so that it can accept the wml files.

Post condition: The authenticated user can access the various services.

Side effects: Takes more time than normal web applications.

Function: Initiating the process

Description: Once the client is authenticated he can select the one of task from the list, corresponding to the submitted login.

Input: User id

Source: The source will be the login provided by the user.

Output: Task list

Destination: Database to generate the form of selected task.

Requires: Selecting of a task from the task list.

Precondition: The task must be selected from the list and submitted.

Post condition: The task will be submitted for further processing's.

Side effects: Might take two screens or enable scrolling if the task list is large.

Function: Displaying of form

Description: The variables of the corresponding submitted task will be retrieved from the database, using ASP and then converting them into WML for displaying on the mobile phone in order to get the input.

Input: The values of the displayed variables.

Source: User will provide the values.

Output: successfully submitted task if the values are correctly submitted to the database else transaction failed message.

Destination: To the main spread sheet and local spreadsheet of the process or the database.

Requires: input of values from the user.

Precondition: Some variables are defined for the selected task.

Post condition: successfully or failed transaction message.

Side effects: Lot of keying from the mobile keypad.

Function: Accessing the Inbox tasks.

Description: The various tasks that are send to the logged in person will be accessible, so that he can view those tasks and make changes if he wants.

Input: The selected task.

Source: The user will select the task from the list of tasks stored in his Inbox.

Output: form of that task with its variable and values.

Destination: The values will be submitted to the main and local spread sheet of the task.

Requires: Selection of the Inbox service.

Precondition: Inbox has some tasks.

Post condition: The new values of the variables will be submitted to the main and local spread sheets.

Side effects: Some keying from the mobile keypad.

Function: Getting the Incident Status information

Description: The status of various tasks that are send to the logged in person can be checked, that is to see which stages are complete and which are still active.

Input: The selected task.

Source: The user will select the task from the list of tasks stored in his Inbox.

Output: Status of that task showing which stages are completed and which are not.

Requires: Selection of the Status service.

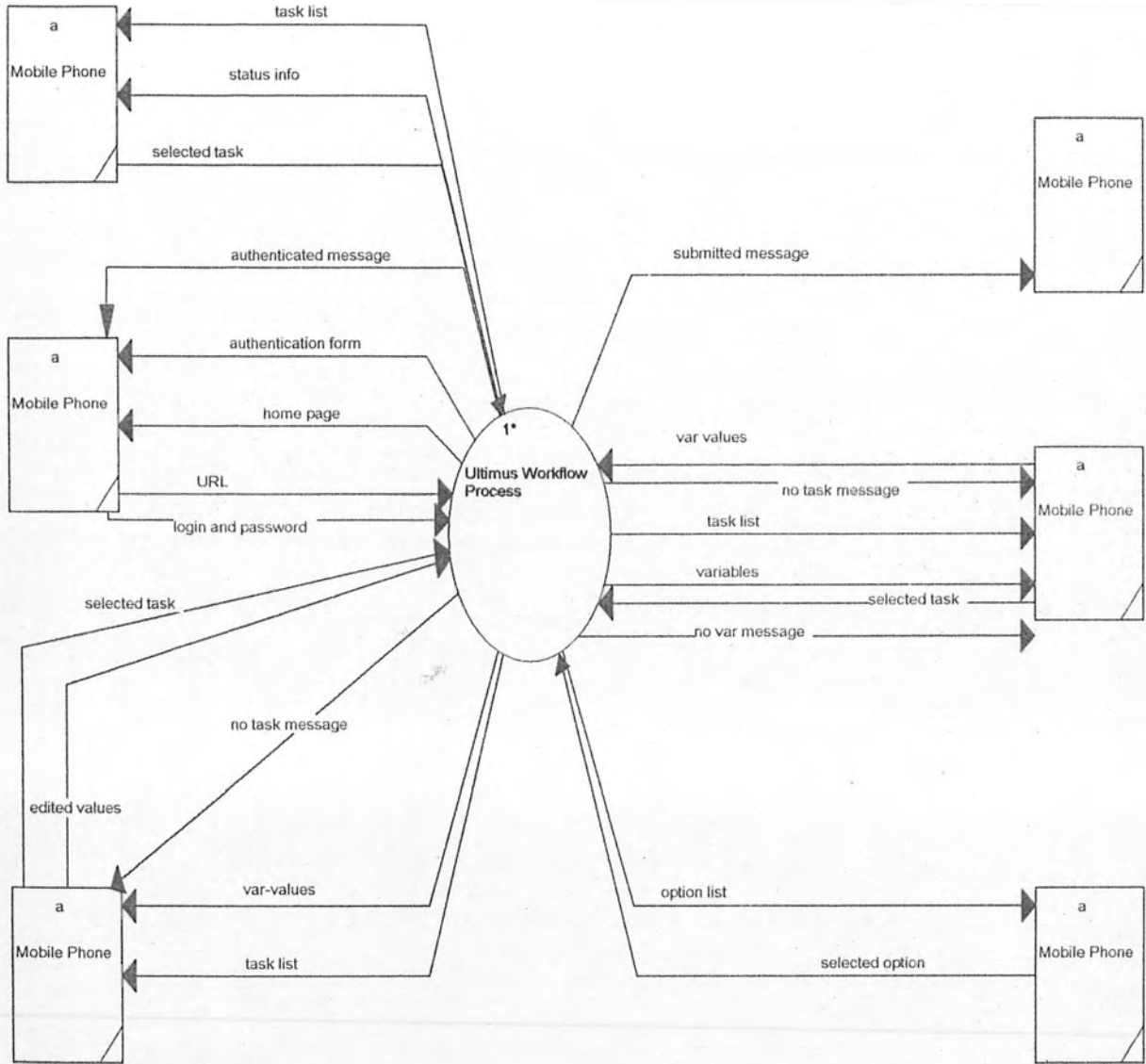
Precondition: Inbox has some tasks.

Post condition: The state of the task will be known

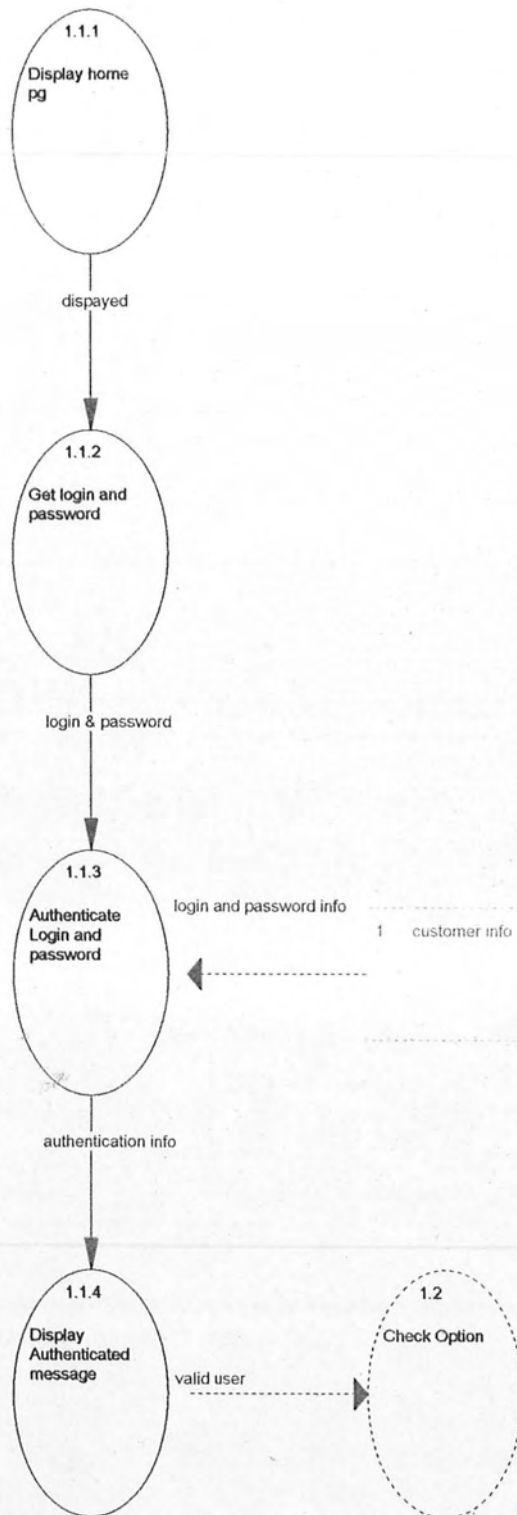
Side effects: None

4.5 ANALYSIS MODELING

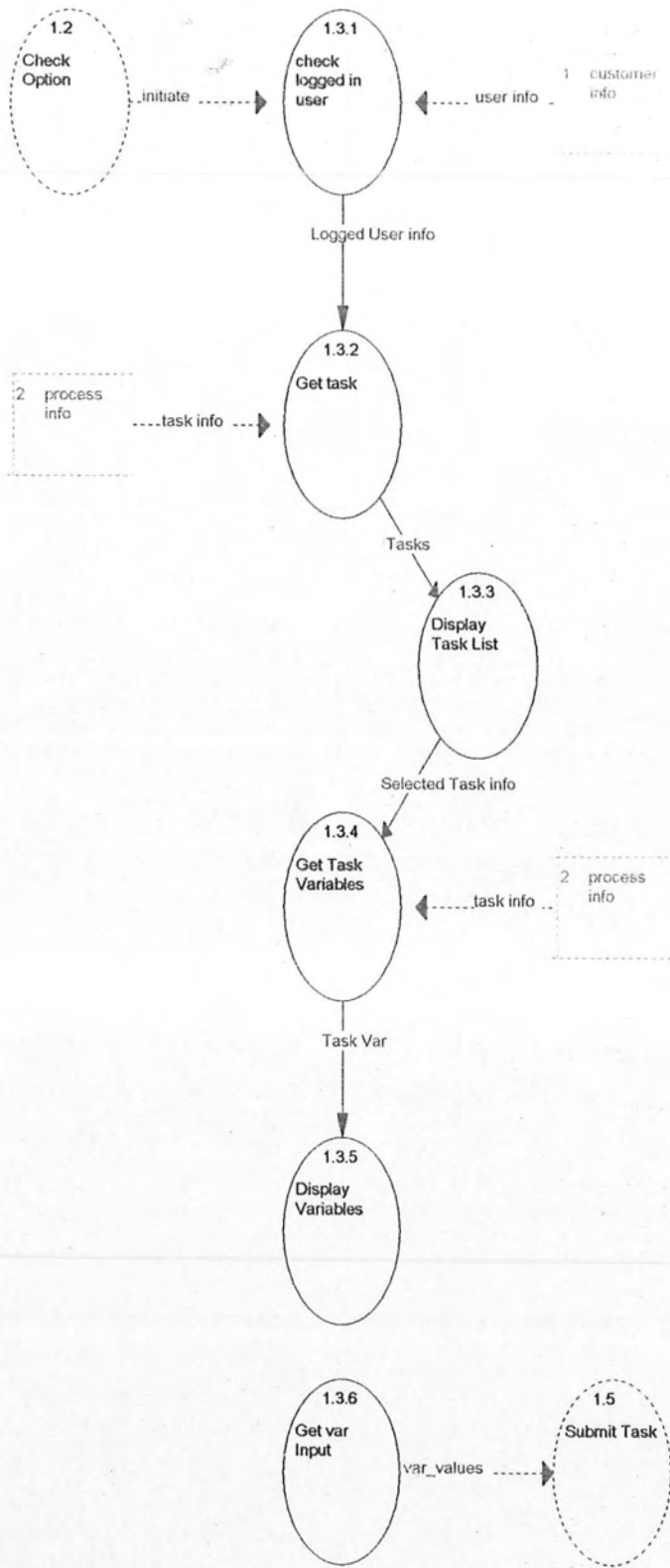
Analysis modeling is actually a set of models. It is the first technical representation of a system. The Data Flow Diagram of the system is given as follows.



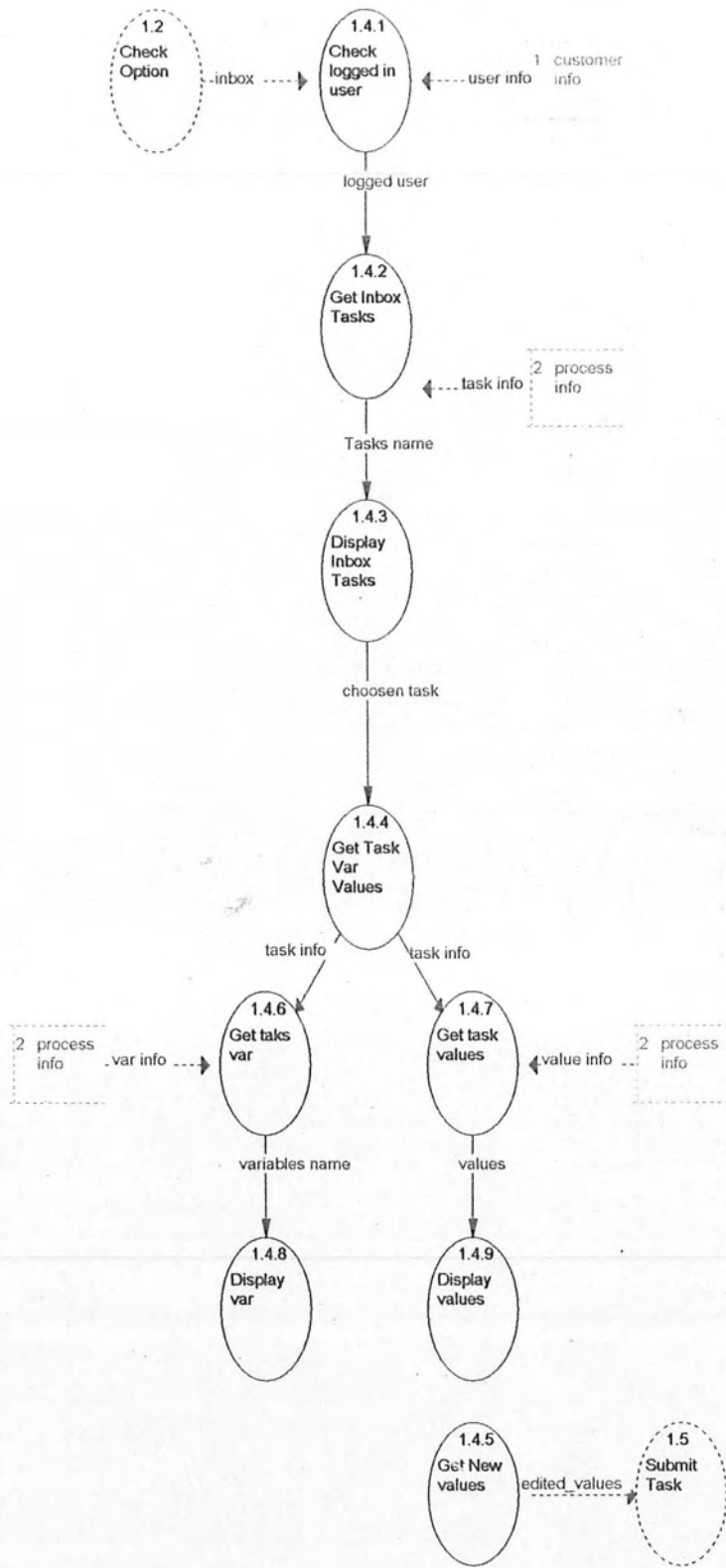
1.1 Connecting to Ultimus



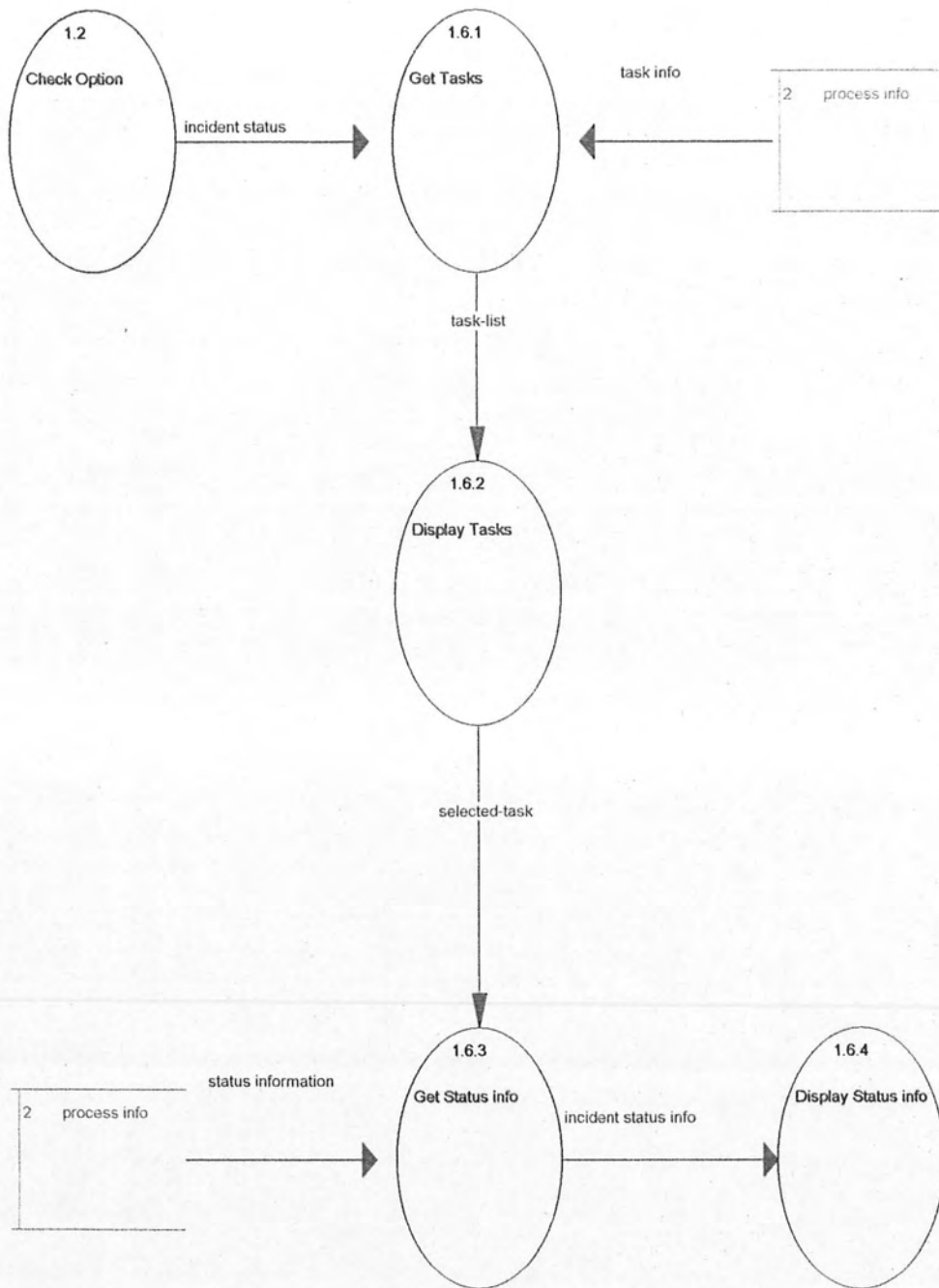
1.3 Initiate Process



1.4 Check Inbox



1.6 Check Incident Status



Chapter 5

SYSTEM DESIGN

5.1 INTRODUCTION

Software design sits at the technical kernel of the software engineering process and is applied regardless of the software process model that is used. Design is the place where quality is fostered in software development. Design is the only way that we can accurately translate a customer's requirements into a finished software product or system[4].

Software design is an iterative process through which requirements are translated into a blueprint for constructing the software. Software design involves Data Design, Architectural Design, Interface Design and Procedural Design

5.2 DATA DESIGN

Data design of my system consists of the following tables.

Assignment Table

Field Name	Field Description	Data Type
Client	Client name to which the process step is assigned	Char
Name	Process name	Char
Assigned To	Client name to which the step is assigned	Char
Identifier	Unique step identifier	Char
Label	Step name	Char

Client Info Table

Field Name	Field Description	Data Type
User id	Unique id for every user	Char
Name	Name of the user	Char
Password	The password through which the user is authenticated	Char
Department name	The name of the department in which the user works	Char

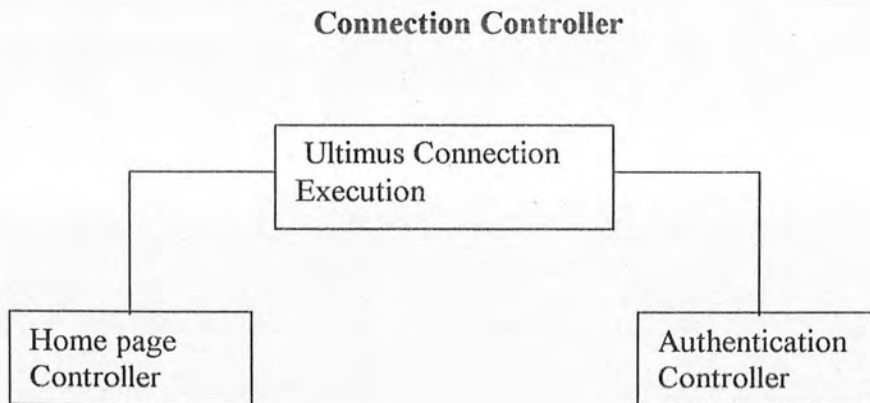
Local SS Table

Field Name	Field Description	Data Type
Task id	Task id of the task that is submitted	Char
Incident	Incident number	Int
Local SS	Binary data containing the Local SS for the incident	Image

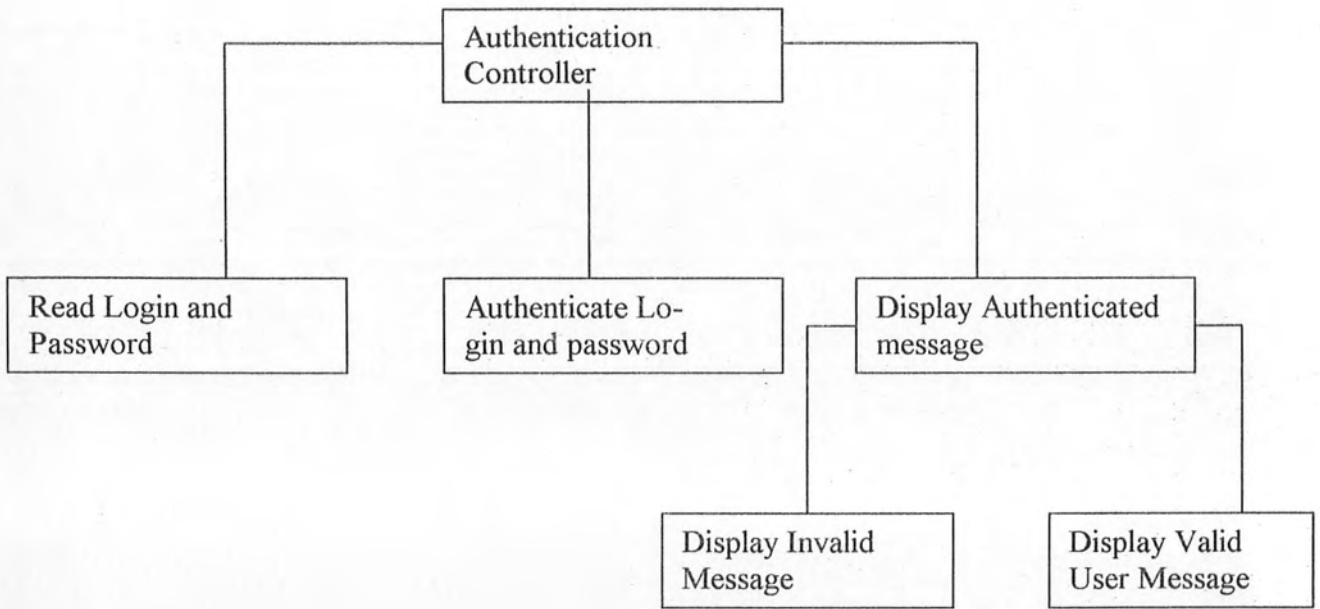
5.3 ARCHITECTURAL DESIGN

The primary objective of architectural design is to develop a modular program structure and represent the control relationships between modules, thus defining interfaces that enable data to flow throughout the program.

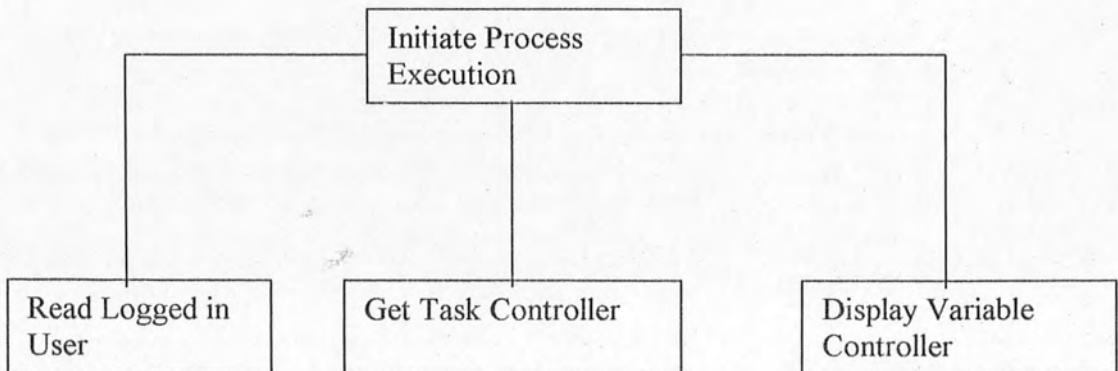
Different modules of the system are as follows



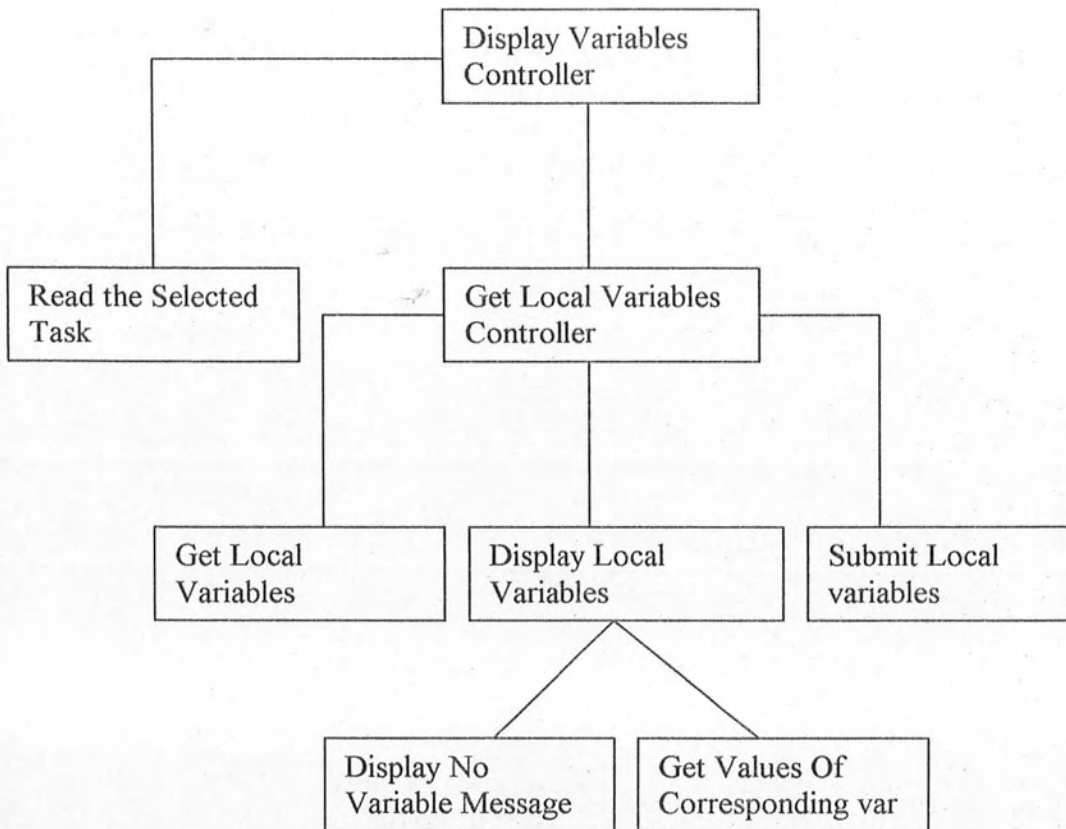
Authentication Controller



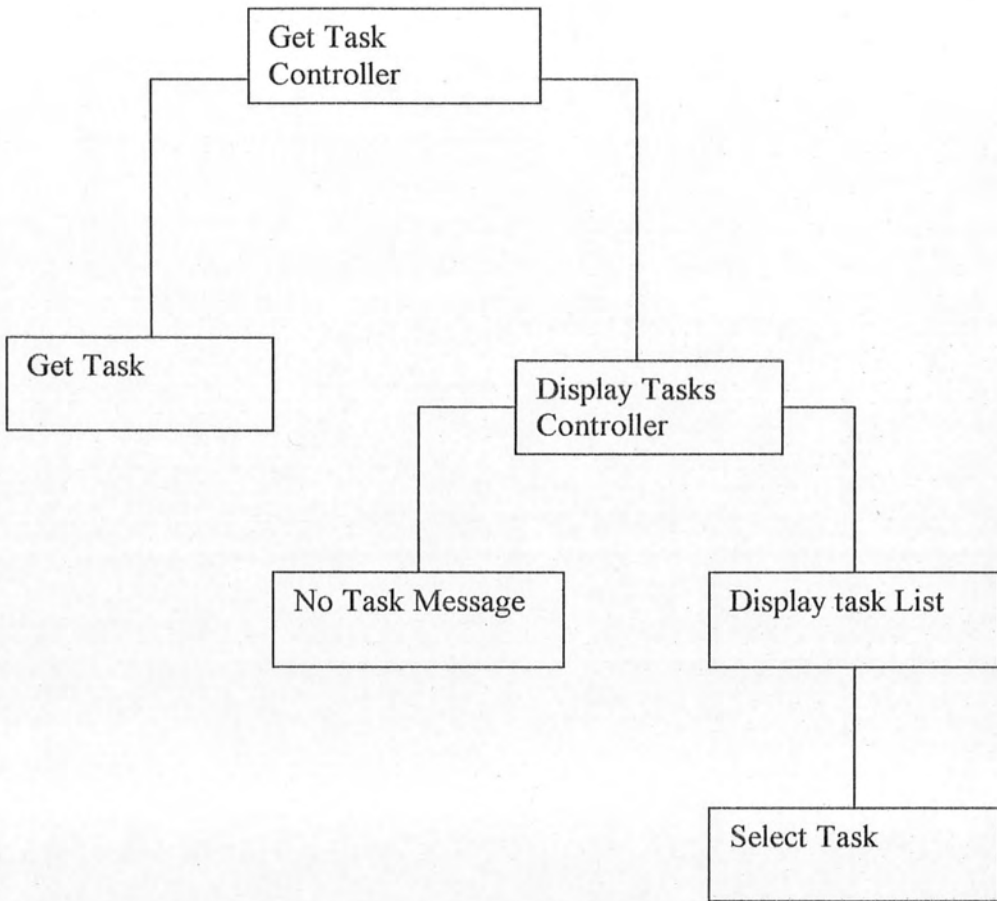
Initiate Process Controller



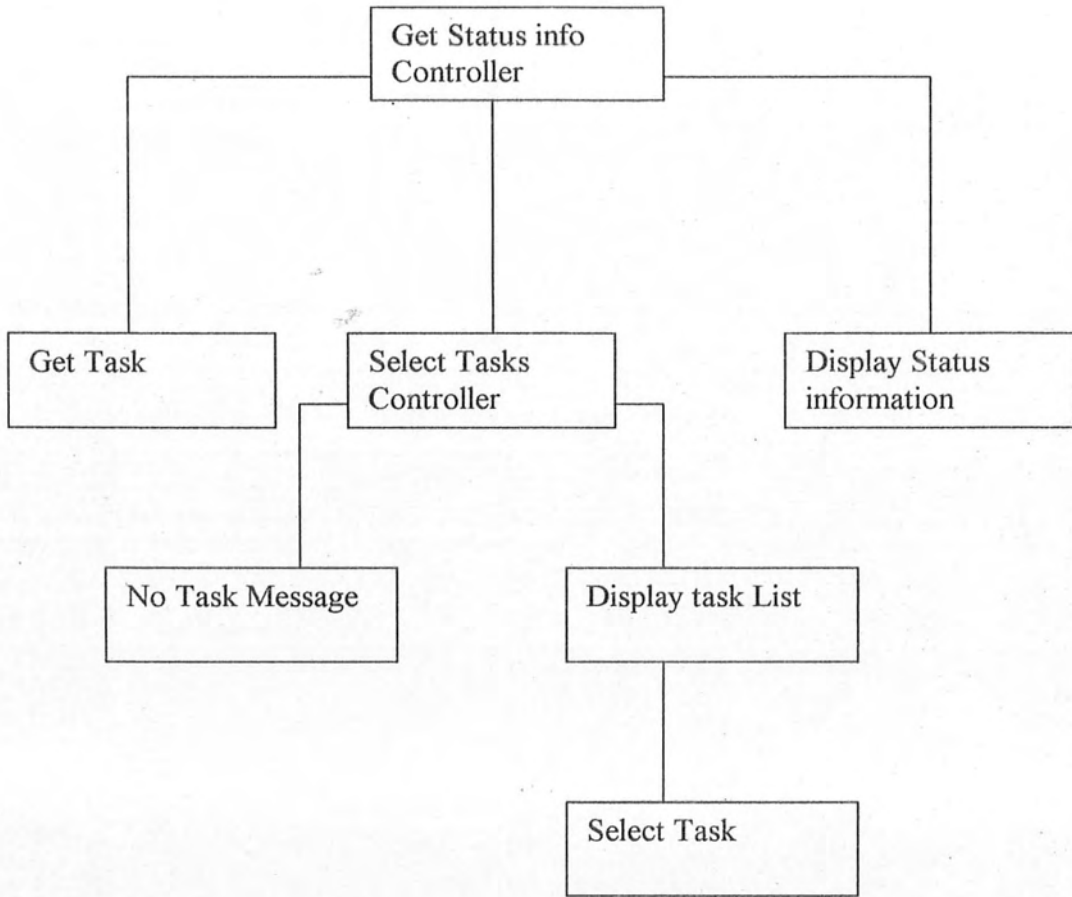
Display Variable Controller



Get Task Controller



Get Status Info



5.4 INTERFACE DESIGN

The interface design focuses on three areas of concern:

- (a) The design of interfaces between software modules.
- (b) The design of interfaces between the software and other nonhuman producers and consumers of information.
- (c) The design of the interface between a human and the system.

The interface between human and the system is very important especially if the machines so used are compact, have small display area, very limited power of processing and little memory capacity. Such is an example of a mobile phones. There are two types of interface that we should consider Input Design and Output Design.

INPUT DESIGN

Mobile phones mostly have text based interface, because of the limited processing and poor display of graphics. The pages so designed have minimum graphics and little but meaningful text, and are limited to 4 x 20 character display. Also that the UP.Browser always reserves one line for function key labels and status.

Since the entry of text is not always easy, therefore are designed pages to reduces or minimize the keying from the user. In this regard we have tried to get the input by letting user pick up one of the displayed choices on the micro browser.

Input forms are designed with following considerations.

- I. Uniform layout of the input screens
- II. Screens suit the purpose for which they are displayed
- III. Meaning full label are used
- IV. Get the input by selecting one of the choices, where possible as to minimize the chances of invalid input and to reduce keying

OUTPUT DESIGN

One of the important purpose of output design is to present the user with data in a meaningful way. Output forms are designed with following considerations.

- I. Meaningful messages are displayed in order to tell the user about the status of the workflow.

- II. If the required step does not have any data, meaningful message is displayed.
- III. The interface is not like windows graphical one, but it has very little graphics.
(Site view diagram is given in Appendix A)

5.5 PROCEDURAL DESIGN

The procedural design is undertaken after data, architectural, and interface designs have been established. Functionality of the system is accomplished through the following modules, Login, Initiate, Inbox and Status. Following is the Login module in Program Design Language (PDL) form.

PROCEDURE Authenticate User;

INTERFACE RETURNS Authenticated Message

TYPE mssg

TYPE login : string

TYPE password :string

TYPE result : boolean

Result=Call verify user()

IF (result=True) THEN

 mssg="successfully verified user

ELSE

 Mssg="user not verified "

END IF

END PROCEDURE Authenticate User

PROCEDURE Verify User

INPUT PARAMETER login

INPUT PARAMETER password

OUTPUT True/False

Check the login

IF (login invalid) THEN

 RETURN False

ELSE

 Check password

 IF(password valid) THEN

 RETRUN True

 ELSE

 RETURN False

 END IF

END IF

END PROCEDURE Verify User

Chapter 6

IMPLEMENTATION

6.1 INTRODUCTION

The hardware and software used to implement the system are as described below.

DEVELOPERS END RESOURCES

Hardware Resources

Machine requirement Pentium II or above

RAM (128 Mb or higher)

Hard disk

Software Resources

Operating System Windows NT/2000

SQL Server 7.0

Ultimus Product

Phone.Com Simulator

IIS (Internet Information Server)

MS Word

Ascent

Microsoft InterDev

CLIENT END RESOURCES

The client will be requiring a mobile phone with an Internet connection. (*see Appendix E, as how to use WAP*)

6.2 FEATURES OF THE USED LANGUAGES

To achieve the objective of the system, features of ASP (Active Server Pages) 3.0, VbScript and WML (Wireless Markup Language) were used. Since Vb Script is a well-known scripting language for quite some time, therefore we will only emphasize on ASP and WML.

6.2.1 ACTIVE SERVER PAGES

Active Server Pages (ASP) is a great tool for creating dynamic pages. ASP is a Microsoft technology, and it works by allowing the functionality of a programming language. ASP code is processed by a special piece of software- the web server.

Seven objects make up the core of ASP. These are known as the built in objects. These objects are Server Object, Application Object, Session Object, Request Object, Response Object, Objectcontext Object and ASPError Object.

SERVER OBJECT

The server object is an object that provides a home to miscellaneous properties and methods provided by the we server itself. This object allows to do things such as create an instance of an ActiveX component, change the course of execution by jumping to another page using the Transfer and Execute properties etc [5].

APPLICATION OBJECT

Since the pages of the application that we create function together, therefore the developer would want some control over the application as a whole. This is the responsibility of the Application object, to notify when an application is first started, so that one perform some startup processing. Similarly it notify when an application end's etc [5].

SESSION OBJECT

ASP allows the developer to track a user from page to page in an application through the use of a session. Each client has a reference to a unique session object. The session object allows one to be notified when a user session begins and ends and store some information that can be accessed by the client throughout the session [5].

REQUEST OBJECT

When a web browser or other client application asks for a page from a web server, this is called making a request. Along with the actual page that the client wants, it can send a great deal of information to the server as well. The Request object is responsible for packaging up that information to make it easily accessible to the ASP application [5].

RESPONSE OBJECT

The primary feature of ASP is the ability to dynamically create web pages. The basic task needed to execute this feature is the ability to tell the client what information to display. There are a number of different ways to shape what the client will display. The Response object exists to provide an efficient interface to control the output to the client[5].

OBJECTCONTEXT OBJECT

The objectcontext object helps to develop applications out of components. It does this by allowing one to handle transactions from within an ASP page. A transaction is single unit of work, that must either succeed in its entirety or if it fails, must be undone completely returning the system to the state it was before the transaction was started. The objectcontext object allows access to MTS in order to start or terminate a transaction [5].

ASPERROR OBJECT

The ASPError object contains details of any errors generated by an ASP script or by the asp.dll itself. It helps in handling of error messages by directing the user to a standard error page [5].

6.2.2 WIRELESS MARKUP LANGUAGE

WML is a markup language based on Extensible Markup Language (XML). XML, like HTML, is a Standardized Generalized Markup Language (SGML) variant. The WAP Forum provides a formal Document Type Definition (DTD) for WML. The WML language defines *elements* and *attributes* that let us specify the user interface components, called *cards*, that users see on their Phones. Just as a Web browser can navigate from one page to another, the micro-Browser can navigate from one card to another.

A card can specify multiple user actions by including one or more of the following:

- Formatted text**--including text, images, and links
- input elements**--which let the user enter a string of text
- select elements**--which let the user choose from a list of options

fieldset elements--which act as organizational containers for other elements

The smallest unit of WML you can send to a Phone is a *deck*--one or more cards that a user is likely to access together. When a Phone receives a WML deck, it displays the content defined in the first card and allows the user to respond. Depending on the card definition, the user can respond by entering text or choosing an option. Phones with larger displays typically present each card as a single screen. Some smaller devices present each card as a collection of screens. Here are some of WML features, Contexts which help you organize and define your service's interface, defining access control, letting users bookmark cards or decks in your service, interacting with the UP.Browser cache, using timers.

USING CONTEXT STRUCTURE

WML provide the concept of nested *contexts* to help us structure our WML service. Specifically, contexts help you do the following:

- I. Organize our service's user interface so that it is intuitive to the user
- II. Provide scoping for variables

Contexts correspond to tasks the user wants to carry out. Each context consists of one or more cards, however many services need to provide more complex contexts, which contain sub-contexts or child contexts. For example, suppose we are implementing the inbox portion of an email service. The overall context that a user accomplishes with an email inbox is viewing mail. A nested context is reading an individual piece of mail. In some cases, contexts are nested several levels deep [6].

CONTEXTS AND VARIABLE SCOPING

The scope of a WML variable is a context. A context can include many cards; any card in a context can reference variables in any other card in the context. Until the user invokes a destination that uses the `<spawn>` element, or issues an `<exit/>` or `<throw/>` task to pop the current context, the variables remain in scope [6].

DEFINING ACCESS CONTROL

All WML decks are public by default. Whenever we access a URL, we gain access to any variables it uses, creating a potential security risk. To control this risk, we can

define *access control* rules on a per-deck basis. To specify which decks (in other words, URLs) can access a particular deck, you must specify an `<access>` statement in the deck header.

USER BOOK MARK SERVICE

Micro-browser bookmarks are similar to conventional Web browser bookmarks. When a user bookmarks a card, the micro-browser creates a bookmark that consists of two items:

- I. A label that identifies the bookmark--this label comes from the `title` attribute within the `<card>` definition. If you do not specify a card `title`, the Browser uses the first line of display text as the bookmark label.
- II. The URL to open when the user selects the bookmark.

INTERACTING WITH THE CACHE

Like conventional Web browsers, the micro-browsers have a memory cache. It caches decks that the user has visited most recently, so it can quickly redisplay a deck without requesting it again from the Web server. When the cache exceeds the Phone's memory capacity, the device drops the least-recently visited decks. If a deck contains time-sensitive information, we can specify a shorter Time To Live (TTL) so that the device will reload the deck from the server more frequently. You can set the TTL for a deck by including a `<meta>` statement within the deck `<head>` element.

USING TIMERS

The `<timer>` element lets us invoke a task automatically after some period of user inactivity. Timers have a number of useful applications, including the following:

- I. Displaying a splash screen for 5 seconds before advancing to the next card.
- II. Automatically updating a stock quote every 15 minutes.
- III. Limiting the time that sensitive data is displayed on the screen.

6.3 MODULES

The system consists of four main modules Login, Initiate, Inbox and Incident Status module.

LOGIN

This module takes the login id and password and verify whether the user is a registered Ultimus user, if yes then he can access various services provided by the Ultimus product through his mobile phone. PDL of this module is given in chapter 5.

INITIATE

This module provides the facility to the logged in user to initiate any task, which are provided to the user according to his access rights.

INBOX

This module provides the facility to the logged in user to check, whether he has received any task from other clients, and if he has then he is able to see the previously submitted values and can add or change them if required. These values are then again submitted for further processing.

INCIDENT STATUS

In order to facilitate the user, so that he can check which stages of his initiated task are completed and which are still active, he can use this module.

Chapter 7

TESTING AND EVALUATION

7.1 TESTING

Like any other phase testing is also an important phase, as it is this phase where the developer can verify that the software behaves as intended by its designer. Thus testing is a process of running a software with the intent of finding an error. Testing is an iterative process with information being fed back from later stages to earlier parts of the process. The testing stages that I have followed are as follows.

UNIT TESTING

I tested the individual components to ensure that they operated correctly. Each component was tested independently, without other system components.

MODULE TESTING

A module is a collection of dependent components. Again the components as a module were tested.

SUB-SYSTEM TESTING

Then I combined the related modules to test them as a sub-system.

SYSTEM TESTING

I integrated the sub-system to test them as a whole system, to check whether they were working properly or not.

STRESS TESTING

The stress testing was used by designing stress tests to check the abnormal quantity, frequency and volume.

7.2 TEST CASES

The test cases that were used during the above mentioned stages are as follows

Test-Id	Test Cases	Expected Result	Observed Result
1	When invalid login and password is entered	Message invalid user should be displayed	Message is displayed
2	When valid login and password is entered	Message authenticated user should be displayed	Message is displayed
3	When the user is found invalid	User is not allowed to access the Ultimus services	User is taken back to the authentication form
4	When the task has no variables	Message no variables should be displayed	Message is displayed
5	When task is not submitted	Message failed transaction should be displayed	Message is displayed
6	When no task is found in the Inbox	Message "no tasks or empty" should be displayed	Message is displayed
7	When the task list increases the maximum capacity, that a deck could handle	The tasks should be shifted to the next deck	The remaining tasks are displayed on the new deck
8	The user interface of the system	There should be minimum keying through mobile keypad	The keying is avoided where possible, by letting the user do most of their task through selection of choices displayed

7.3 EVALUATION

The objectives and the requirements mentioned by the organization have been met. The software is fulfilling all the functional and non-functional requirements of the system.

As this technology is relatively new so not much work has been done on it. There are many work areas, which are still underdevelopment. One of the main problems, which I have faced and which might be resolved with time and new advancements, is that different micro-browser supported by different mobile phones display their content differently. Others do not support certain tags that work on one browser quite efficiently. So WML page written for one browser will be displayed poorly in any other browser. But with new technology we might be able to modify the existing pages a little bit, so that they can be displayed equally well in any kind of browser. This will free the users from being restricted to specific kind of browsers for better display.

Secondly we can introduce a facility to initiate the whole process by document attachments, without filling out any forms. Also new WAP compliant flobots can be introduced.

Ultimus


PAKISTAN

To,
Chairman,
Computer Science Dept.
Quaid-i-Azan University
Islamabad.

Dear Sir,

This is to certify that Ms. Bushra Saleem has successfully completed her project with us, namely, "Workflow through WAP". Her work meets the functional requirements of the project we specified for her.

We are very pleased with her work and wish her all the luck in her future endeavors.



Imran A. Gondal
Manager Development.
March 26, 2002

ULTIMUS PAKISTAN

27-A/1, LANE 2, GULISTAN COLONY, NATIONAL PARK ROAD, RAWALPINDI 46000, PAKISTAN.

Tel. +92 51-525093, 563674 Fax. +92-51-527169 E-Mail ultpk@workflowzone.com

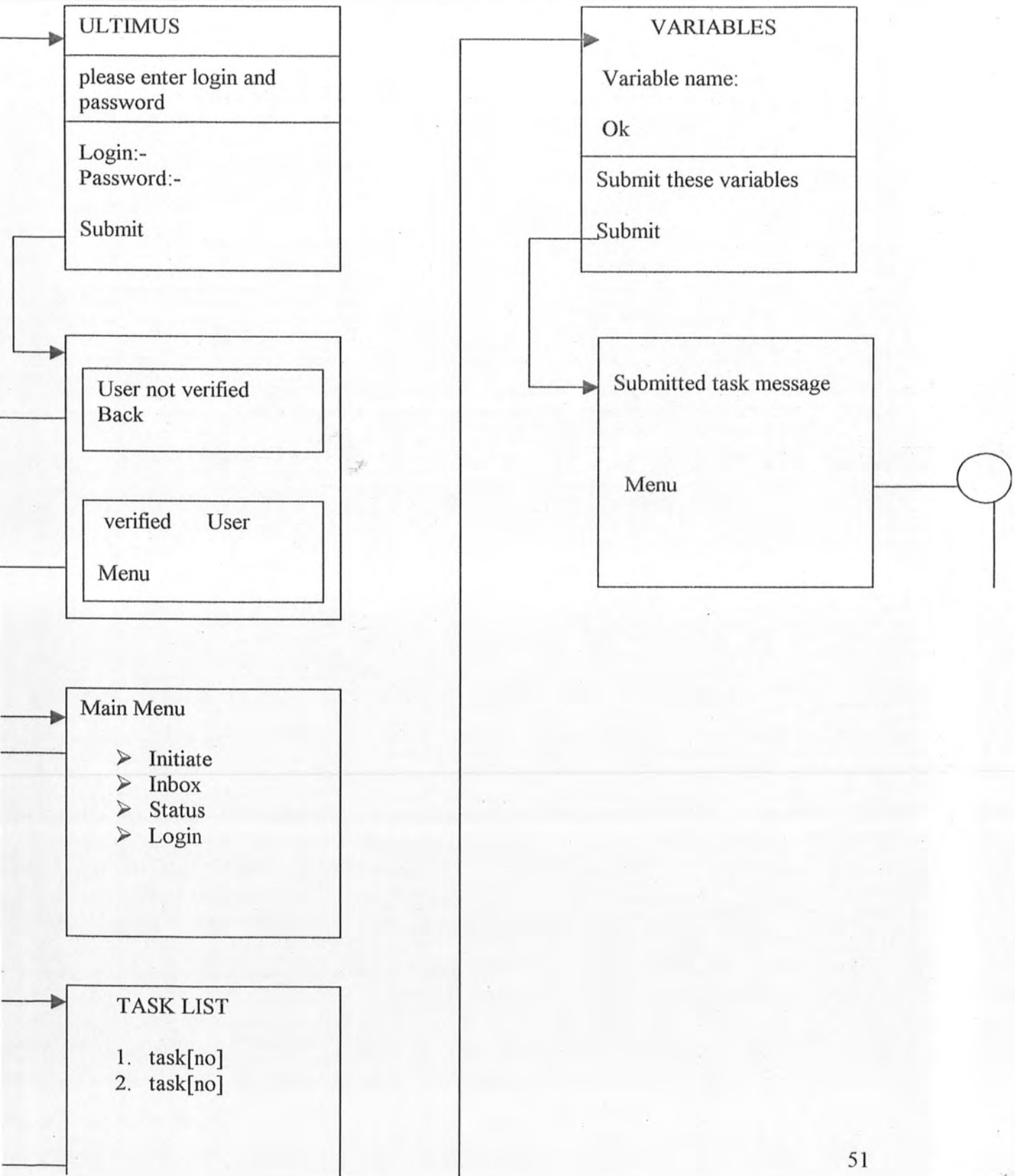
www.workflowzone.com

APPENDICES

(A)

SITE HIERARCHY DIAGRAM

SITE VIEW DIAGRAM



A



- Initiate
- Inbox
- Status
- Login

TASK LIST

1. task[no]
2. task[no]

Submit

More

TASK LIST

1. task[no]
2. task[no]

Submit

VARIABLES

- Var_name[value]
- Var_name[value]

Edit

Submit

Var_name:

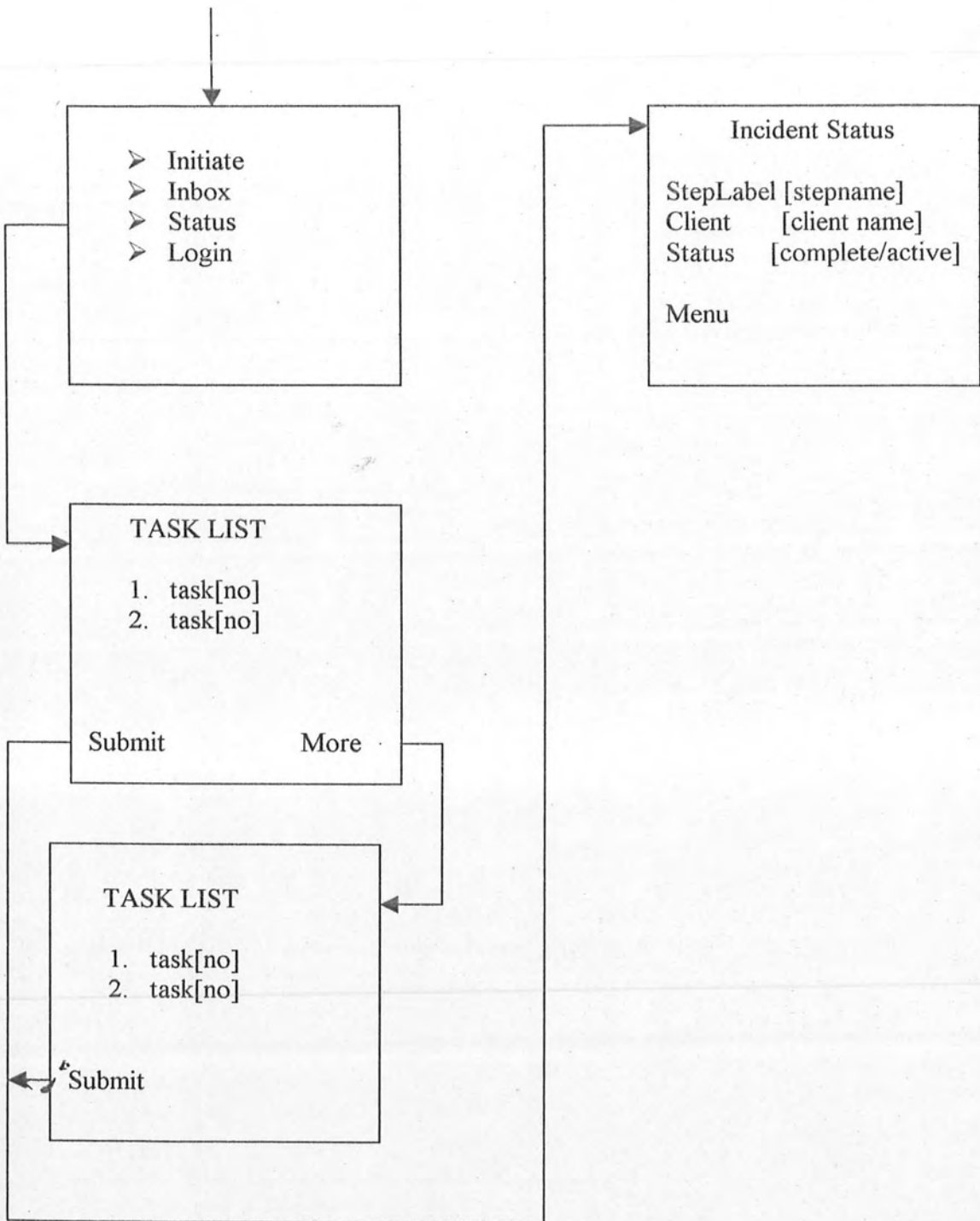
Submit

Submitted task message

Menu



B



(B)

OVERVIEW OF WAP TRANSACTION

OVERVIEW OF MOBILE TRANSACTION

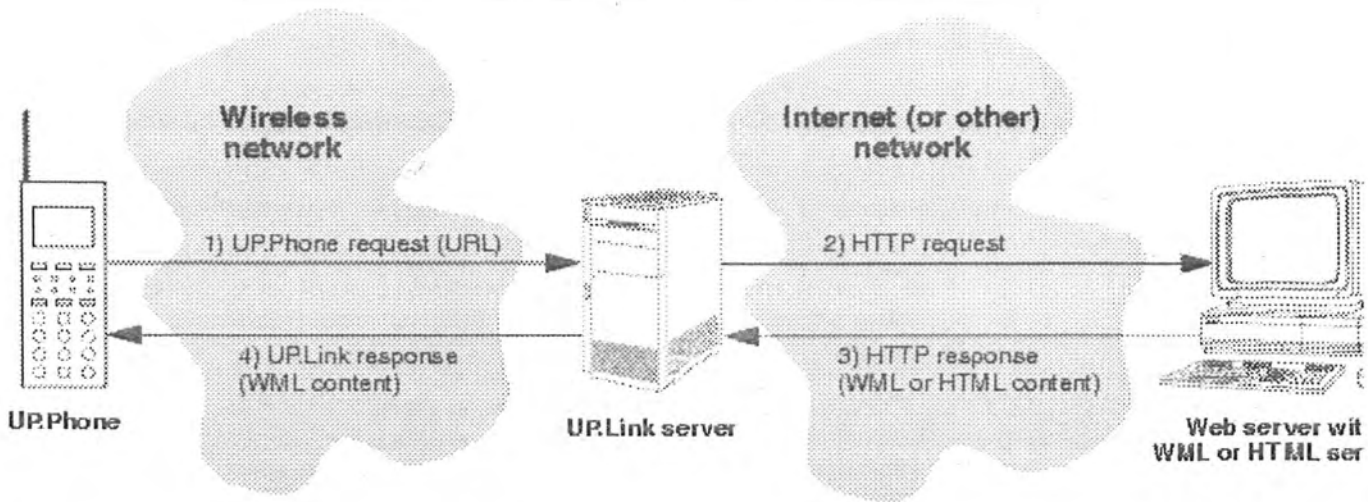
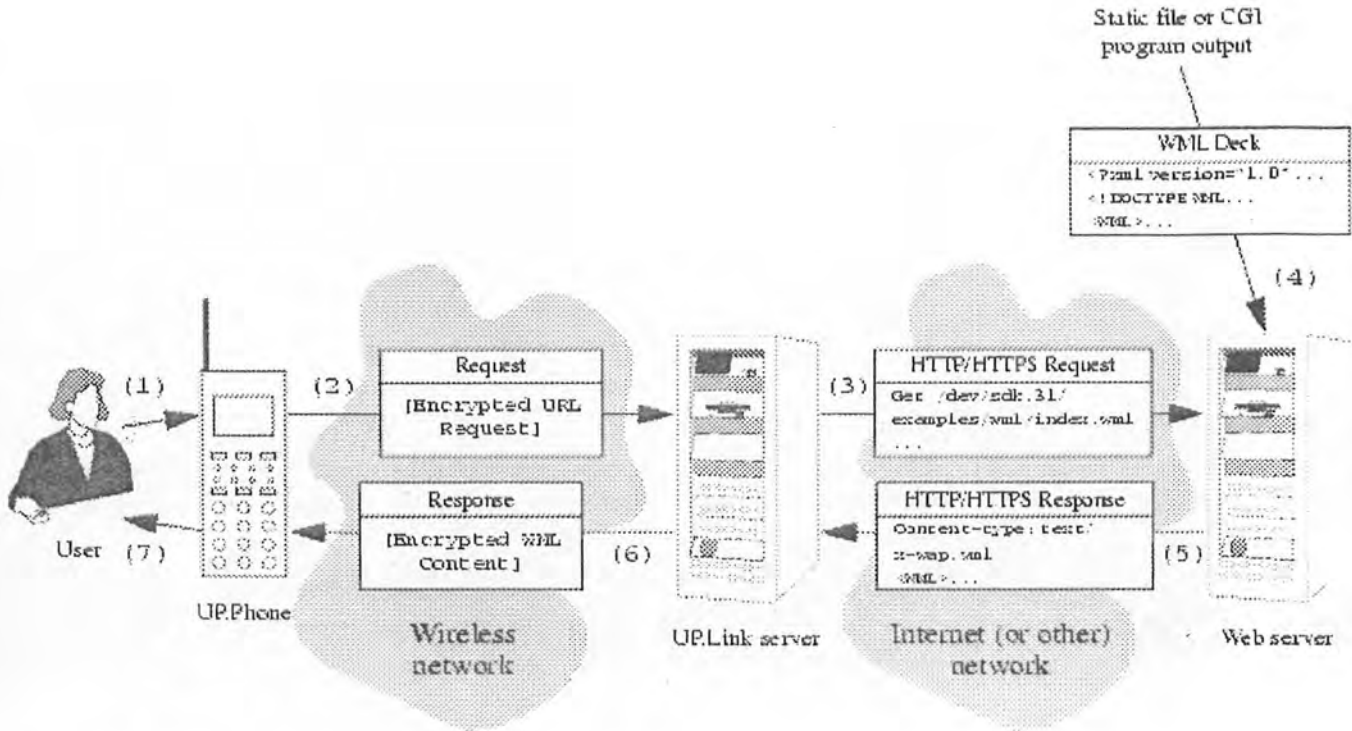


Figure: (Openwave Getting Started Guide)

A TYPICAL WML TRANSACTION



reference figure (Openwave Getting Started Guide)

1. The user uses an mobile phone to request a URL.
2. The micro browser creates a request containing the URL and information that identifies the subscriber and sends it to the UP.Link Server (Gateway).
3. The gateway parses the request, generates a conventional HTTP or HTTP Secure (HTTPS) request, and sends it to the Web server.
4. The Web server parses the request and determines what to retrieve. If the URL specifies a static file (as in this example), the Web server retrieves it. If the URL specifies a CGI program, the Web server launches the program.
5. The Web server prepends an HTTP header to the static file or CGI program output and sends it back to the gateway.

6. The gateway parses the response, validates the WML, generates a response (removing the HTTP or HTTPS header), and sends it to the phone.
7. The micro browser parses the response and displays the content to the user.

(C)

SYNTAX FOR DEFINING WML DECK

WML SYNTAX

The high-level syntax for defining a WML deck is shown below:

```
<?xml version="1.0"?>
  <!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML1.1//EN"
    "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
  <head>
    access control and meta information
  </head>
  <template>
    deck-level event bindings
  </template>
  <card>
    card-level event bindings
  </card>
</wml>
```

Element	Description
<wml>	Required. The <wml> element specifies a deck.
<head>	The <head> element specifies information about the deck as a whole, including metadata and access control information. All WML decks are private by default--you must explicitly define a deck as public for cards or decks outside your domain to access it.
<template>	The <template> element defines deck-level event bindings, i.e. characteristics that apply to all cards in the deck. You can override these characteristics for a particular card by specifying the same event bindings within the <card> element.
<card>	Required. One or more <card> elements define the actions and user interface for the deck. A card can contain the following components: <ul style="list-style-type: none">• Formatted text--which can include text, images, and links• <input> elements--which allow the user to enter a string of text• <select> elements--which allow the user to choose from a list of options• <fieldset> elements--which act as organizational containers for other elements

Most WML elements have one or more **attributes**, some of which are required (indicated in bold) while others are optional. Attributes let you specify additional information about

how the device should handle the element. Although the exact syntax of a WML statement depends on whether or not the element has content, attributes always appear within the element start tag, as shown below:

```
<element a1="value1" a2="value2" ...> content </element>  
<element a1="value1" a2="value2" .../>
```

You must enclose attribute values within single (') or double (") quotation marks and separate each attribute-value pair with *white space* (space, tab, newline, or carriage return characters). White space is *not* allowed, however, between the attribute name, equal sign, and attribute value.

Like HTML, WML uses the printable 8-bit character set and converts one or more contiguous newlines, carriage returns, tabs, or spaces to a single space. Unlike HTML, WML is case-sensitive--all WML elements and attributes must be lowercase.

(D)

CONFIGURING YOUR WEB SERVER

WAP APPLICATION SERVER

To make your server WAP compliant, you need to add the following MIME types to those the web server knows about.

Content	MIME	Extension
WML	text/vnd.wap.wml	wml
Compiled WML	application/vnd.wap.wmlc	wmlc
WML Script	text/vnd.wap.wmlscript	wmls
Compiled WML Script	application/vnd.wap.wmlscriptc	wmlsc
Wireless bitmap/WBMP	Image/vnd.wap.wbmp	wbmp

MIME (Multipurpose Internet Mail Exchange) is a specification for the format of data that can be sent over the Internet. When the server sends data in response to a request it receives, it sends a MIME type with it. This MIME type can also be explicitly set by the application. Normally the file extension of the requested file is associated with a MIME type and so the server automatically issues the correct MIME type. Then when a browser receives information from the server, it checks its MIME type to see what to do with it. If for example, it sees that the data has a MIME type of "image/vnd.wap.wbmp" then it knows to display it as a picture.

(E)

HOW TO USE WAP FACILITY

USING WAP ENABLED MOBILE PHONE

First, you need a WAP enabled phone. When you buy a WAP phone, it will probably be configured by an operator for immediate use. Therefore the first thing you should do is to find "Services" or "Internet" on the menu. When you enter this category, one of the next options is Home. If your phone is set up, entering Home should take you to an operator's home page. If this does not work, you need to set up your phone by yourself. Many operators send you an automated set up via SMS. This is called an Over The Air (OTA) configuration. If your operator offers this service, you should find it at their Web pages.

(F)

USING UP.SIMULATOR

UP.SIMULATOR

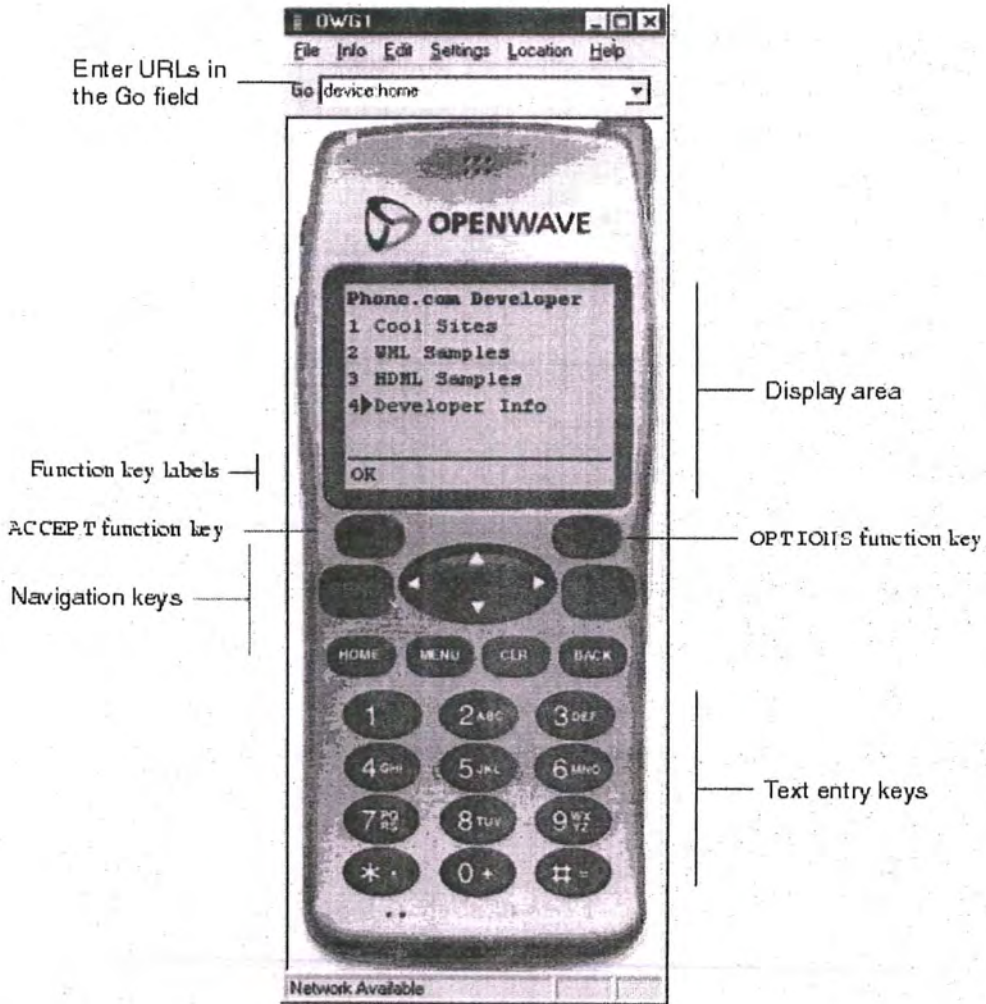


figure (reference UP.SDK Getting Started Guide)