SKYVE
Online SMS and Contacts Operator-Independent Archiver

Angelo Dimech
Abstract

SKYVE : Online SMS and Contacts Operator-Independent Archiver

Angelo Dimech

The mobile phone is a relatively new means of communication. Nevertheless due to its usability it has evolved to become one of the most used ways of communication over the last ten years.

A mobile phone is a medium that has the main scope of allowing its user to call and send short text messages (SMS) to other people from practically anywhere. Due to the inconvenience of having to carry a separate medium to store contacts information, mobile phones have always had the functionality to store contacts details. Being a portable device it is very easily lost or stolen, and when this happens people are generally frustrated due to having lost their contacts and SMS messages more than about the mobile phone itself.

One of the easiest ways of overcoming this frustration is by archiving the information on the mobile onto another medium. Currently the only way to archive information from your mobile phone is by using proprietary software and connecting the mobile phone to your PC. This type of archiving is mostly useful when one loses his mobile phone but has the disadvantage that it can only be accessed through the same PC. This project will deal with a solution to overcome the mentioned problem. The solution is intended to became a platform for other solution that can be added in the future for archiving SMS and contact details.
Table of Contents

Goals and Objectives......................................................................................................................... 3
Chapter 1 - Introduction......................................................................................................................... 4
  Project Scope........................................................................................................................................ 4
  Retrieval of Contacts and SMS ........................................................................................................... 5
  Tools to Access the Retrieved SMS and Contacts ............................................................................... 5
Chapter 2 - Literature.............................................................................................................................. 6
  Background Knowledge ....................................................................................................................... 6
  What Methodology to Use ................................................................................................................. 6
    ASP.Net............................................................................................................................................. 6
    Java.................................................................................................................................................. 7
Chapter 3 – System Analysis.................................................................................................................. 9
  Structured System Analysis and Design Method .................................................................................. 9
  Analysis of Current System ................................................................................................................ 9
  Business Specifications ..................................................................................................................... 9
  Website ....................................................................................................................................... 10
  Skyver Application ............................................................................................................................. 12
Chapter 4 – Methodology In Action...................................................................................................... 14
  Project Plan ....................................................................................................................................... 14
Chapter 5 - Design................................................................................................................................. 15
  Database Designs ............................................................................................................................... 15
  RDBMS Selection ............................................................................................................................... 23
  Product Overview .............................................................................................................................. 27
    SQL Server 2005............................................................................................................................ 27
    Visual Studio .net ............................................................................................................................ 28
    Adobe Photoshop CS2 .................................................................................................................... 28
    Macromedia Dreamweaver ............................................................................................................. 29
  Solution Design .................................................................................................................................. 29
    Application Design ........................................................................................................................ 30
    Website Design ............................................................................................................................... 31
  Any Roles Involved ............................................................................................................................. 38
Chapter 6 - Implementation...................................................................................................................... 39
  Implementation ................................................................................................................................... 39
  Implementation Code........................................................................................................................ 39
  Testing Strategy .................................................................................................................................. 48
  Testing Environment ........................................................................................................................... 49
  Testing Procedures .............................................................................................................................. 49
  Release Management ......................................................................................................................... 50
  Software ............................................................................................................................................ 50
  Conclusion ......................................................................................................................................... 50
  Test Cases ......................................................................................................................................... 50
    Console Application Testing ......................................................................................................... 50
    Web Application Testing ................................................................................................................. 55
  User Guide ......................................................................................................................................... 61
Chapter 7 – Summary and Conclusions .................................................. 75
  Objectives that have been Met.......................................................... 75
  Processes Carried Out and Decisions Taken........................................ 75
  What I Have Achieved from this Project............................................ 76
  Success Factors .................................................................................. 76
  Future Enhancements ......................................................................... 78
Chapter 8 – Bibliography ...................................................................... 79
Goals and Objectives

The objective of this solution is to have a tool that will backup Contacts and SMS and retrieve them anytime online. This solution should be independent on the Mobile Phone and Operator of the Network being used.

The objective is to access any contact and SMS received and send by just accessing a site from the internet.
Chapter 1 - Introduction

Project Scope

The scope of this project is to find a global solution to backup Contacts and SMS. In order to achieve this goal a tool to retrieve the contacts from the mobile phone is required and a tool to access and manage the information retrieved in required. The following figure will outline the scope of the project and the components used to build up the project.

Skyve Overview

The purpose of this sub-system is to retrieve the SMS and contacts from the mobile phone. The application will first connect to the mobile phone, and then the user will be able to synchronize the contacts on the mobile phone to the contacts on the web. This application will be the interface for the mobile phone contacts and SMS as well as the online stored Contacts and SMS.

Communication from the mobile phone to the application can be done through various ways mainly through a USB cable, Bluetooth, IrDA (infrared) and serial cable. For this project the communication will be done through USB cable. Due to this connection, communication at the lower level is done through AT Commands which communication through the USB port on the pc to the mobile modem.

This application will be written using Microsoft Visual Studio and it will make use of AT commands for pc-to-mobile communication.
Retrieval of Contacts and SMS

The tool should have the following features:

1. Detection/Connection to Mobile Phone
2. Manage SMS from Mobile/SIM (Send, Read, Delete, Store)
3. Manage contacts from Mobile/SIM (Create, find, delete, read)
4. Manage SMS from Server
5. Manage Contacts from Server (Create, find, delete, read)
6. Synchronization function to update SMS/Contacts Online

Tools to Access the Retrieved SMS and Contacts

Special care will be taken during the design of the interface to ensure user friendliness, effortless navigation, trouble-free search and location of information and cross-browser compatibility to ensure that the site is accessible by all audiences.

This part of the project includes the creative and artistic impression which will then be followed by the development and design of optimized graphics for quick download, simple navigation, ease of use, accessibility, search engine optimization for correct indexing and more.

The website’s main features will be:

- Login/Logout
- Manages Contacts (Create, find, delete, read)
- Manage SMS (Send, Read, Delete)

The SMS and Contacts information are managed from to the Database through the Business and Data
Chapter 2 - Literature

Background Knowledge

As mentioned before, contacts and SMS of your mobile phone or PDA are very important to store. Storing in your home PC is a good thing but not the best thing. Ideally they would be stored somewhere on the web and can be accessed from anywhere in the world where there is an internet connection. Hence I envision creating an online service where people can archive their SMS messages as well as their contacts and offer them the possibility to access them online. This offers an added benefit to the normal archiving offered by the mobile manufacturers. The web portal will also have the functionality to manage the contacts and SMS and possibility to send SMS through web interface.

In order to achieve the above described solution the following development platforms where considered.

- JAVA
- .NET

What Methodology to Use

Methodologies are explicit mechanisms for helping to solve ‘problem’. However, as soon as we have more than one methodology for solving problems, we have an additional system is fortunate to have a wide range of methodologies: it is estimated that there are over 1000 brand-named methodologies in use all over the world.

Most methodologies originate from academic, industrial, business and consultancy companies. Understandably, some of the creators or owners of methodologies try to promote their wide use. However, the ‘choice’ problem is not resolved as they are reluctant to divulge the weakness, shortcomings or inappropriateness of their methodologies, and are not open about the difficulties or situations that are not covered. Therefore, potential methodology users need some independent support to help them evaluate and select a relevant methodology for the ‘problem situation’ that they face.

ASP.Net

ASP.Net was suitable for the development of this project because it is an integrated web application platform which provides the necessary services to construct and set up an enterprise-class web application. It offers an infrastructure which is definitely more secure and scalable. It is part of the .Net Framework that simplifies application development in the highly distributed environment of the internet. Whilst programming, Developers may refer to the .Net framework class library which is included in ASP.Net. This is an object oriented collection of types which programmers may use to develop applications.

ASP.Net offers a large number of benefits, the ones below are the ones we came across and found very useful whilst the development of this project:
Multiple Language Support: Unlike ASP, ASP.Net pages are created with advanced programming languages such as VB.Net and C#. On the other hand ASP classic pages are created with only scripting languages such as JavaScript and VBScript.

Compilation of ASP.Net pages: When an ASP.Net page is triggered it's immediately compiled and cached on the server thus improving performance. This therefore permits early binding, strong typing and Just in time compiling to native code.

Ease of configuration: ASP.Net uses XML based files to store the configuration settings which are readable and coded by any user. Once configured correctly than ASP.Net applies the new configuration settings to web resources whenever it locates a change in the configuration files without the need of rebooting the server.

Code Behind the Logic: Once again ASP.Net was chosen because unlike ASP the code behind file (aspx.cs) consists of event handlers, initialization code and other code for the user interface in the aspx file. On the other the drawback of ASP is that the classic pages don't provide behind the html page. In fact Both HTML and scripts are present in just one single page.

Browser-independent applications: ASP.Net caters for detecting the type of browser present and browser compatibility issues when it generates code for a server control.

Security: ASP.NET provides default authorization and authentication schemes for Web applications.

Scalability and Performance: Performance and scalability of your application will be definitely improved thanks to ASP.Net output caching.

Simplified Deployment: ASP.Net allows you to deploy a web application as easily as any other HTML webpage. All that has to been done is to copy it to the server without needed to run regsvr32 to register any components. As mentioned before the server configuration settings are stored in an XML file within the application.

Java

Some java advantages are listed below:

1. Fully cross platform APIs including support for Graphics, Windowing and Multimedia. All future APIs will be cross platform. Java has even more APIs than Windows. Look at the current

2. Plenty of libraries available. Many are free and/or open source.

3. High density of OO design patterns put to practice compared to C#/.NET. This makes developing complex applications much easier (at the cost of making 'my first text editor' type projects slightly more difficult).

4. Well designed APIs following good OO practice.
5. The Java Community Process allows people with an investment in Java have a say. New Java features can be proposed, discussed and voted for in an open democratic forum. Microsoft does NOT have anything like this which is unfortunate for anyone investing in .NET.

6. Currently supported by multiple commercial vendors (not one).

7. Although not recognised by any international standards committee, Java is an open standard governed by Sun and the JCP members (or which there are MANY). C# is an ECMA standard but without support from Microsoft, very few third parties have been willing to implement it. Sun also tends to be less secretive than Microsoft about their products. The C#/CLI ECMA submission is also just a subset of the .NET framework. It has yet to be seen whether any future APIs for .NET will ever be submitted for standardisation (it doesn't look good).

8. Smallest unit of distribution is a class (.class file). This helps in application development, debugging and deployment. Having a one to one mapping between types and the file system is simpler because file systems are well understood.
Chapter 3 – System Analysis

Structured System Analysis and Design Method

The SSADM method involves the application of a sequence of analysis, documentation and design tasks concerned with the following.

Analysis of Current System

After the research conducted it resulted that there were no current global available solutions to backup sms and contacts from the mobile phone to an online server with the possible to access them from anywhere online. The solutions that were available were to backup the mobile phone details on your local computer, and these solutions were limited to a particular brand of mobiles (ex nokia has its own SDK to backup the mobile phone details). The proposed system is intended to backup contacts and sms for all GSM phones having AT commands and upload them online and eliminate the limitation to access them only from your local pc.

Business Specifications

The front-end application will work over the web. The backend application will be saved on the client’s PC. Both the front-end website the backend client application will communicate with the same database residing on an online server. The public pages of the site will be available to the general public through the World Wide Web. The private information of the users will be available after user logs in and is authorized to view his personal information, which is retrieved from the database.

Access to the information in the database is done through a Generic Data Layer which interacts with the database directly. This generic layer will be used for both the front end website and the backend client’s application.

Figure 2: 3-Tier Data Layer
Website

Home Page: The homepage will always be the most common landing page for most website visitors to the Skyver website. With this in mind, the homepage will be designed in a way to attract users by providing the information they expect to see and the relevant quick links.

The homepage will include some of the following information:

- **Homepage Banner:** This will give an overview of the application in general
- **Easy Navigation:** The menu is the most important feature in the website. Without a proper menu, not all users will access all areas in the site as required.
- **Sign In/Login:** The home page will contain a sign in/login for registered users. This part of the homepage will also have a forgot password link which will help users retrieve their password.
- **Register:** The register link will user to the register page
- **Download Link:** This will guide the user to download the required application to proceed with archiving SMS and contacts
- **Product Information Link:** This will guide the user to read details about features and functionality of this product. This will be a useful page for new visitors to learn about the Skyver solution
- **Register Page:** The register page is the page where the user will be able to register their account with Skyver. The user is prompted to enter his details including the following:
  1. Name
  2. Surname
  3. Email
  4. Username
  5. Password
  6. Confirmed password
  7. Address Line1
  8. Address Line 2
  9. Country
  10. Contact Number
- **Product Functionality Page:** This page will be a page that will gives information about the SKYVE application including Instructions how to download, install, use the product and all the features of the Application/Website.
- **Application Downloads Page:** This page will instruct the user to download the Skyver application. Instruction how to install the application and how to make best use of it will also be included.
- **Links Page:** A This Link Page will contain links to various sources of information/other websites. The Links will be useful for users to get more information such as links to supported phone home pages.
• **Contact us/Help Page:** A useful section of the website will be to provide a reference point for those who wish to contact Skyver. Basic contact information such as postal address and telephone numbers will be provided as well as addresses to point out the office. Furthermore, an enquiry form will also be made available to allow website visitors to submit enquiries or questions directly to SKYVE and receive any replies from your staff by email.

• **User Profile/Settings:** This option is available only for registered users. Once a user signs in the webpage, user can update the profile, which consist of the details that user submitted on registration. User has also the option to change various data such as password.

• **View User Contacts:** This option is available only for registered users. A user will be able to view his/her archived contacts

• **View User SMS:** This option is available only for registered users. A user will be able to view his/her archived SMS

• **Manage SMS:** This option is available only for registered users. A user will be able to manage his/her archived SMS. This functionality will include:
  1. Delete
  2. Search
  3. Add to a Group

• **Manage Contacts:** This option is available only for registered users. A user will be able to manage his/her archived contacts. This functionality will include:
  1. Create
  2. Edit
  3. Delete
  4. Search
  5. Add to a Group
Skyver Application

The Skyver application will be the connection point between the user’s mobile and the Skyver solution. This application consists of the following features:

- **Application Login**: The application login details will be the same as the login for the Skyver website. The user will be prompted with a username and password in order to login the Skyver application. This information will be used to upload the SMS and contacts later to the server and link the contacts/SMS to the user.

- **Mobile Phone Communication through USB**: This In order to retrieve information from the mobile phone a communication layer is required. A lot of research was done on this area to identify the best method to do the communication. The first option was to use AT commands through the windows application. After some trial and error, this solution resulted to be quite time consuming so other third party components were searched. Another choice was to use a commercial communication layer which is Mobile Edit (www.mobiledit.com). This solution was good but considering this project is a proof of concept, the company sponsoring my thesis did not agree for the moment to purchase this communication module. After lots of research, going through developer’s forums and web search, a free communication module that suits the application needs was found. This module is an open source module which consists of a library where communication can be done through the mobile phone and the windows application. The library can be found in the following URL http://www.scampers.org/steve/sms.

- **Mobile Phone Identification/Mobile Status**: In order for the application to retrieve messages and contacts the application needs to establish communication with the phone. The application will indicate that connection with the mobile phone is established and the mobile phone is identified.

- **Retrieval of Contacts**: Once communication is established and phone is identified one of the major Skyver functions is to retrieve contacts from the mobile phone. Contacts retrieved can be both from the SIM card memory and from the mobile phone memory.

- **Contacts Manipulation (Create, Edit, Delete)**: Once the contacts are retrieved on the application user have the possibility to manipulate them in the application. The task of the user here is to sort out the contacts and get the contacts ready to be archived.

- **Backup of Contacts**: Once contacts are sorted and manipulated the Skyver user can back up the contacts through the following options
  
  o Backup to local File
  
  o Backup to an online Server (which later can be accessed from an online website)

  To backup to an online Server an active internet connection is required.
• **Retrieval of Messages:** Once communication is established and phone is identified one of the major Skyver functions is to retrieve messages from the mobile phone. Contacts retrieved can be both from the SIM card memory and from the mobile phone memory.

• **Messages Manipulation (Create, Edit, and Delete):** Once the messages are retrieved on the application user have the possibility to manipulate them in the application. The task of the user here is to sort out the messages and get the messages ready to be archived.

• **Back up of Messages:** Once messages are sorted and manipulated the skyver user can back up the contacts through the following options:
  
  - Backup to local File
  - Backup to an online Server (which later can be accessed from an online website)

To backup to an online Server an active internet connection is required.
Chapter 4 – Methodology In Action

Project Plan

In order to accomplish this project, the company has provided a research and development project schedule, as shown in the figure below.

The letters A-F in the figure above indicates different stages of the Report writing.
A - Thesis Proposal
B - Introduction and Background
C - Design of System
D - Implementation chapters
E - First Draft including Analysis, Testing Results and Future Enhancements.
F - Final Version
Chapter 5 - Design

Database Designs

The following section describes the Database structure for the Skyver Solution

Table Friendly Name: Users

Description: The following tables will store the registered users that will be allowed to make use of the system.

Table Name: tbl_Users

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Default</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>pk</td>
<td>UserId</td>
<td>int</td>
<td>Identity</td>
</tr>
<tr>
<td></td>
<td>Username</td>
<td>varchar(255)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Password</td>
<td>varchar(255)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Email</td>
<td>varchar(255)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Name</td>
<td>varchar(255)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Surname</td>
<td>varchar(255)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Address1</td>
<td>varchar(255)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Address2</td>
<td>varchar(255)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Country</td>
<td>varchar(255)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DateJoined</td>
<td>datetime</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IsLoggedIn</td>
<td>Boolean</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>LastDateLoggedIn</td>
<td>datetime</td>
<td>✓</td>
</tr>
<tr>
<td>Column Name</td>
<td>Column Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UserId</td>
<td>This column will contain an auto generated number as a unique record identifier.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Username</td>
<td>This column will contain the username of the registered user.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td>This column will contain the encrypted password of the registered user.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Email</td>
<td>This column will contain the email of the registered user.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>This column will contain the name of the registered user.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Surname</td>
<td>This column will contain surname of the registered user.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Address1</td>
<td>This column will contain the first part of the address of the registered user.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Address2</td>
<td>This column will contain the second part of the registered user.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Country</td>
<td>This column will contain the country of the registered user.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DateJoined</td>
<td>This column will contain the date the user registered.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IsLoggedIn</td>
<td>This column is a Boolean which will indicate if a user is logged in at a particular instance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LastDateLoggedIn</td>
<td>This column will contain the date of the last time the user logged in</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Friendly Name: SMS

Description: The following tables will store the archived SMS of the users

Table Name: tbl_SMS

<table>
<thead>
<tr>
<th>Column</th>
<th>Data Type</th>
<th>Default</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
<td>SmsId</td>
<td>int</td>
<td>Identity</td>
</tr>
<tr>
<td>FK</td>
<td>UserId_fk</td>
<td>int</td>
<td></td>
</tr>
<tr>
<td>FK</td>
<td>Status_fk</td>
<td>varchar(255)</td>
<td></td>
</tr>
<tr>
<td>FK</td>
<td>Type_fk</td>
<td>varchar(255)</td>
<td></td>
</tr>
<tr>
<td>FK</td>
<td>Group_fk</td>
<td>varchar(255)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Recipient</td>
<td>varchar(255)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Text</td>
<td>varchar(255)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ArchivedDate</td>
<td>varchar(255)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Deleted</td>
<td>varchar(255)</td>
<td></td>
</tr>
</tbody>
</table>

Column Name | Column Description
-------------|--------------------------------------------------
SmsId        | This column will contain an auto generated number as a unique record identifier.

UserId_fk    | This column will contain the foreign key of the user of the SMS

Status_fk    | This column will contain the foreign key of the status of the SMS
<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Default</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
<td>ContactId</td>
<td>int</td>
<td>Identity</td>
</tr>
<tr>
<td>FK</td>
<td>UserId_fk</td>
<td>int</td>
<td>☒</td>
</tr>
<tr>
<td>FK</td>
<td>Group_fk</td>
<td>int</td>
<td>☑</td>
</tr>
<tr>
<td>Name</td>
<td>varchar(255)</td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>Surname</td>
<td>varchar(255)</td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>ShortName</td>
<td>varchar(255)</td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>MobileNumber</td>
<td>varchar(255)</td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>OtherNumber</td>
<td>varchar(255)</td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>ArchivedDate</td>
<td>DateTime</td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>Column Name</td>
<td>Column Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ContactId</td>
<td>This column will contain an auto generated number as a unique record identifier.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UserId_fk</td>
<td>This column will contain the foreign key of the user of the sms</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group_fk</td>
<td>This column will contain foreign key to the group the contact belongs. If this</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>column is null then this contact does not belong to any group</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>This column will contain name of the contact</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Surname</td>
<td>This column will contain surname of the contact</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ShortName</td>
<td>This column will contain the short name of the contact</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MobileNumber</td>
<td>This column will contain name of the contact</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OtherNumber</td>
<td>This column will contain the any number of the contact which usually is the home</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ArchivedDate</td>
<td>This column contain the date when the contact was last modified.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ModifiedDate</td>
<td>This column contain the date when the contact was last modified.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deleted</td>
<td>This column will indicate if a contact has been deleted.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Friendly Name: System Error

Description: The following tables will store errors occurred whilst the system is running.

Table Name: tbl_SystemErrorLog

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Default</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
<td>ErrorLogID</td>
<td>bigint</td>
<td>×</td>
</tr>
<tr>
<td>UserIPAddress</td>
<td>varchar(20)</td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>Description</td>
<td>varchar(255)</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>StackTrace</td>
<td>text</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>URL</td>
<td>varchar(255)</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>ErrorDate</td>
<td>datetime</td>
<td></td>
<td>×</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Column Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorLogID</td>
<td>This column will contain an auto generated number as a unique record identifier.</td>
</tr>
<tr>
<td>UserIPAddress</td>
<td>This column will contain the IP address of the User (client) performing the action.</td>
</tr>
<tr>
<td>Description</td>
<td>This column will contain the Exception Message Information.</td>
</tr>
<tr>
<td>StackTrace</td>
<td>This column will contain the Exception stack trace information.</td>
</tr>
<tr>
<td>URL</td>
<td>This column will contain the URL in which the error occurred.</td>
</tr>
<tr>
<td>ErrorDate</td>
<td>This column will contain the date and time of when the error occurred.</td>
</tr>
</tbody>
</table>
Table Friendly Name: Group

Description: The following tables will contain list of groups which will SMS and Contacts

Table Name: tbl_Group

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Default</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
<td>GroupId</td>
<td>bigint</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>Name</td>
<td>varchar(20)</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>varchar(255)</td>
<td>✓</td>
</tr>
</tbody>
</table>

Column Name | Column Description
---|-------------------------------------
GroupId | This column will contain an auto generated number as a unique record identifier.
Name | This column will contain name of the Audit Type
Description | This column will contain Description Type

Table Friendly Name: Group Type

Description: The following tables will contain the list of group types

Table Name: tbl_GroupType

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Default</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
<td>GroupTypeId</td>
<td>bigint</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>Name</td>
<td>varchar(20)</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>varchar(255)</td>
<td>✓</td>
</tr>
<tr>
<td>Column Name</td>
<td>Column Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GroupTypeId</td>
<td>This column will contain an auto generated number as a unique record identifier.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>This column will contain name of the Group Type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>This column will contain Description of the Group Type</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table Friendly Name: SMS Status**

**Description:** The following tables will contain the list of SMS Status

**Table Name:** tbl_SMSStatus

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Default</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
<td>SMSStatusId</td>
<td>bigint</td>
<td>✗</td>
</tr>
<tr>
<td>Name</td>
<td>varchar(20)</td>
<td></td>
<td>✗</td>
</tr>
<tr>
<td>Description</td>
<td>varchar(255)</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Column Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMSStatusId</td>
<td>This column will contain an auto generated number as a unique record identifier.</td>
</tr>
<tr>
<td>Name</td>
<td>This column will contain name of the SMS Status</td>
</tr>
<tr>
<td>Description</td>
<td>This column will contain Description SMS Status</td>
</tr>
</tbody>
</table>
Table Friendly Name: SMS Type

Description: The following tables will contain the list of SMS Types

Table Name: tbl_SMSType

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Default</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
<td>SMSTypeId</td>
<td>bigint</td>
<td>×</td>
</tr>
<tr>
<td>Name</td>
<td>varchar(20)</td>
<td></td>
<td>×</td>
</tr>
<tr>
<td>Description</td>
<td>varchar(255)</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

Column Name | Column Description
---|------------------------------------------------|
SMSTypeId   | This column will contain an auto generated number as a unique record identifier.
Name        | This column will contain name of the SMS Type
Description | This column will contain Description of the SMS Type

RDBMS Selection

The relational database management system can be described as a type that stores data in the form of related tables. These type of databases are considered to be one of the best because they require few assumptions about how data is related or how it will be extracted from the database. Due to this the database can be viewed in different ways. It is important to know that a great advantage of a relational system is that a stand alone database can be spread across a number of tables. On the other hand a flat file database does not allow this since each database is self contained in a single table.

Considering between MySQL and SQL server was a very delicate topic when considering an application for data management. After intense research I must say that both are efficient at storing your data in an organised and readable way through a user interface. However they differ drastically in many areas. There during the project analysis we asked the same question as many other developers asked; “Which is better, MySQL or SQL Server”.

Skyver • Uppsala University
In order to understand better which best fitted our needs we had to tackle a few questions such as:

1. How flexible do you need your data to be?
2. How secure must your data be?
3. Do you need any type of support whilst managing your data?

Standard Language Support

One must know that a very important aspect when developing a database is how to access your data using different standard protocols. Both of the two mentioned database do a good job in this in supporting all major protocols which are used for an application to communicate to a SQL database. Transact SQL is a common protocol involving a series of statements which an application can use to access data and create new tables in a SQL database. Such statements are used to insert, update and delete records.

ANSI which stands for American National Standards Institute is referred to as the international standard for this type of programming. For example ANSI decides whether the word INSERT will be used in the programming language as opposed to the word ADD to insert a new record in the database. Both databases do a good job in supporting all major protocols but MySQL does not completely follow the ANSI standards. This is a problem due to the fact that if your data is constantly increasing in size then there might be a problem in handling vast amounts of data.

MySQL is suitable more towards selecting data so that it could be displayed, updated and saved again. It is weaker in the areas of inserting and deleting data. MySQL would be a good choice for tracking clients and creating dynamically populated pages with information from the database. For an application of moderate to large scale that is used for commerce of any kind, SQL Server is by far the better choice with more options and functionality to add, manipulate, update and delete data.

Below we can find technical differences in MySQL and SQL Server when it comes down to the ANSI standard:

- MySQL does not support Triggers, SQL Server does.
- MySQL does not support User Defined Functions, SQL Server does.
- MySQL does not have Cursor Support, SQL Server does.
- MySQL does not support Stored Procedures, SQL Server does.
- MySQL does not have FULL JOIN capabilities, SQL Server does.
- MySQL has limited Import/Export capabilities, SQL Server has full support for Import/Export.
- MySQL has limited Transaction Support while SQL Server offers full Transaction Support.

Therefore SQL Server is a better choice for the company that will need to have full control over their data to edit and change it at our own needs.
Comparison between MySQL and SQL Server:

<table>
<thead>
<tr>
<th>Feature</th>
<th>SQL Server</th>
<th>MySQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL View Support</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Triggers</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Stored Procedures</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>User Defined Functions</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SQL SELECT</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CURSOR Support</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>XML Support</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>FULL JOIN</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Referential Integrity</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Transaction Support</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Full Text Support</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Import/Export Support</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Auto Tuning</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Database Management Tools</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Query Analysis Tools</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Online backup support</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Clustering Support</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Security through stored procedures &amp; views</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>OLAP Services</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Data Mining</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Data Reporting</td>
<td>✓</td>
<td>✗</td>
</tr>
</tbody>
</table>

Security Support

Security is another major concern when coming to decide between the two databases. Nowadays securing critical data from third party people is an ever increasing and trying task. SQL Server has been certified as C2 compliant, which ensures that the database has sufficient security for government security. MYSQL supports security via the SQL Grant command which is limited to granting security at the table level. Meaning that if any data within the table needs to be protected from a particular user than the entire table has to be secured from that user. On the other hand SQL Server offers security at a column level meaning that any portion of data in the table can be secured from any particular user while allowing that same user to see other portions of the table data.

SQL Server and MySQL also make their database more secure by abstracting its data behind a layer of stored procedures. This ensures that developers never see how the actual data is represented.
Support and Service

Support is another aspect one should consider when you want to manage data. You must know what kind of support to should be able to receive from the data manufacturer. Both MySQL and SQL Server offer support plans that one can find available from their sellers. It’s good to know also that both offer some free kind of support as well as paid ones.

MySQL claims employ approx 100 people worldwide, which make up the entire company consisting of product development, support staff, sales staff, distribution, and clerical workers. This is a very small number to assist people from all around the world and may worry some database administrators. On the other hand SQL Server, from Microsoft, has a significantly large support program and large staff backing the program.

Pricing

MySQL is free, how does FREE sound? It can be used for any purpose whether it’s used to manage a large company data or to populate a small web site with data.

SQL Server on the other hand, could take a small independent company’s budget and eat it for lunch! However, the many outstanding features that are far and away above MySQL offset the price paid for SQL Server drastically.

Conclusion

After having studied in detail both databases and their features SQL Server is the wisest choice sue to its rich and many features in manipulating, securing and managing data. MySQL definitely lacks support in the basic database features meaning that the development of an application to interface with the database will be both more expensive and will take longer to finalise. All this extra time and code will cost time and money to develop and maintain the system.
Product Overview

**SQL Server 2005**

Microsoft SQL Server 2005 offers:

**Faster Results:** It provides a new Management Studio, integration with Visual Studio 2005, and the Microsoft .NET common language runtime - all of which help you build, debug, and operate applications faster and more efficiently.

**Better Decisions:** SQL Server 2005 provides a comprehensive business intelligence platform for data integration, analysis, and reporting that helps you turn insight into action and make better decisions, faster.

**Trusted Platform:** SQL Server 2005 supports the highest performance, availability and security to run your most demanding applications with native data encryption, secure default settings, and password policy enforcement.

System Minimum Requirements

<table>
<thead>
<tr>
<th>Processor</th>
<th>600-megahertz (MHz) Pentium III-compatible or faster processor; 1-gigahertz (GHz) or faster processor recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>Microsoft Windows 2000 Server with Service Pack (SP) 4 or later; Windows 2000 Professional Edition with SP 4 or later; Windows XP with SP 2 or later; Windows Server 2003 Enterprise Edition, Standard Edition, or Datacenter Edition with SP 1 or later; Windows Small Business Server 2003 with SP 1 or later</td>
</tr>
<tr>
<td>Memory</td>
<td>512 megabytes (MB) of RAM or more; 1 gigabyte (GB) or more recommended</td>
</tr>
<tr>
<td>Hard Disk</td>
<td>Approximately 350 MB of available hard-disk space for the recommended installation Approximately 425 MB of additional available hard-disk space for SQL Server Books Online, SQL Server Mobile Books Online, and sample databases</td>
</tr>
<tr>
<td>Drive</td>
<td>CD-ROM or DVD-ROM drive</td>
</tr>
<tr>
<td>Display</td>
<td>Super VGA (1,024x768) or higher-resolution video adapter and monitor</td>
</tr>
<tr>
<td>Other Devices</td>
<td>Microsoft Mouse or compatible pointing device</td>
</tr>
<tr>
<td>Other Requirements</td>
<td>Microsoft Internet Explorer 6.0 SP1 or later For Reporting Services, you need Microsoft Internet Information Services (IIS) 5.0 or</td>
</tr>
</tbody>
</table>
**Visual Studio .net**

**Microsoft Visual Studio offer:**

ASP.Net was used to create most of the entire structure of the website since it is surely the latest Microsoft programming framework which allows the rapid development of powerful web applications.

We decided to use this type of application for the simple reason that is a powerful database driven application. ASP.Net allows building applications that interface with a database. It is an object oriented application and has a considerable number of tools which will certainly allowed us to develop the project faster together with more functionality.

Explaining into a more technical lever ASP.Net is fast simply because of:

1. **Compiled Code:** The code within the application is complied into “machine language” before your visitor enters this site.

2. **Caching:** This is the storage information which will be reused in a memory location so that there would be faster access in the future thus improving the performance of web applications. Another point at hand is that ASP.Net allows the caching of data from a database simply because it will not tend to slow down by frequent visits to a database when the data does not change very often.

ASP.Net also allows multiple language support so that developers can code in more than 25 .Net languages. This is a great advantage since it will allow programmers to develop their programs in the language they know best thus having no problem to find developers to support the work on the site being developed.

One should keep in mind that ASP.Net applications can only be hosted on a Microsoft web server.

**Adobe Photoshop CS2**

The Adobe Photoshop allows us to bring out the best in our digital images and transforming them into anything you can imagine.

Photoshop allows us to create unique graphics, it has a considerable number of art tools and filter effects which could give your image a special design which would attract the user’s eyes.
There are thousands of Photoshop tutorials online where one could learn different techniques and even get great help from where to start off from if you are new to Adobe Photoshop. However once you learn how to use Photoshop to make your images looking the way you really want them to look then I can assure you that you won’t be able to stop yourself from editing every digital image that you come across.

The software is definitely user friendly and whatever you need to learn about Adobe Photoshop there will be a tutorial online for that for sure.

**Macromedia Dreamweaver**

Macromedia Dreamweaver was used for the html code of the website since it is an extremely flexible tool by allowing us to view the html page in multiple ways. There is the “design view” which shows us a sample of the page would be rendered in a browser. On the other hand the “code view” will show us the code found behind the html page. To help the developer whilst the production of the web page Macromedia Dreamweaver also offers a different view called the “split view” option where the developer can view both the code and design at the same time.

This a great advantage since the programmer may see the changes in the design in real time after editing the code. Once we were happy we the design of the page than we copied the html code in the ASP.net application to code the backend functionalities.

**Solution Design**

The main User interface screens of the proposed solution are outlined below:

**Application Design**

- Application Login
- Main Application Screen

**Website Design**

- Home Page
- Register Page
- About Us
- Links
- Product Features
- Forgot Password
- Contact Us
- Contacts Not Logged In
- Create Group
- View Contacts
- View Messages
- View Individual Message
**Application Design**

Application Login

The application Login will be used to access the Mobile application. In order to access the main application the following are required:

- Username
- Password
- Port (Communication Port)
- Timeout
- Baud Rate

The above fields will be entered in the following screen

![Skyver](image.png)

*Figure 3: Console Design*

**Main Application Screen**

Once a user is authenticated and logs in the following screen will appear
The main purpose of this screen is to choose the required contacts and SMS and upload them to the online server so later they will view and managed online through the web interface.

**Website Design**

A master page will be used to have the basic design and style. Different pages will display different information. The pages are mainly divided into 2 types. Type 1 are the public pages which can be seen by the general public and type 2 are the pages used by logged in users.

**Home Page**

The first user interface of user will be with this home page which will be accessible from the web. The following is the first artistic impression of the home page.
Register Page

After a visitor visits the skyver website, visitors have the possibility to register to be able to download the application and after upload the SMS /Contacts. The design of the registration page is the following
Figure 6: Register Page Design

About Us

The about us page will give information about the Skyver Solution and the team behind the Skyver Solution.
Links

The links page will display useful links, including partners of the skyver solution as well as forums/help pages for the skyver users.

![Image of Links Page Design]

**Figure 7: Links Page Design**

Forgot Password

If register user have forgotten their password the following page will be useful to retrieve back their password.

![Image of Forgot Password Page Design]

**Figure 8: Forgot Password Design**
Contact Us

This contact us page is available for all visitors and users. The aim of this page for any is for visitor/user to contact the administrators of the skyver solution.

![Contact Us Design](image)

**Figure 9: Contact Us Design**

Contacts Not Logged In

If a user clicks on the contacts page and is not logged the following page will appear.

![Contacts Not Logged In](image)
View Contacts

The following page is for logged in user which will display the contacts of the logged in user. The contacts can be managed and sorted into groups. Search within the contacts is also available.

![View Contact Design](image)

**Figure 10: View Contact Design**

View Messages

The messages page will display all the users messages in the context of the groups they are related. User can manage all the messages (search and delete) as well as sort the messages into different groups. Search functionality will also be available.
View Individual Message

The following page will be used to display individual message with more details. Similar page will be done for the contacts, where each individual contact can be viewed in greater detail. Contacts can also be edited as apposed to SMS. A modification date to show the timestamp when a date has been changed can also be displayed.

![Image of View Individual Message Design]

**Figure 11: View Individual Message Design**

Create Group

Groups can be created to sort out SMS and contacts.
Any Roles Involved

The application involves 2 main roles which are:

- **Standard User**

A standard user will be a normal user and will be able to use the functionality of the application once he/she has registered through the site to upload contacts and SMS. The login details can then be used to access the uploaded SMS and contacts from the website.

- **Administrator User**

An Administrator user can have the functionality of a standard user, as well as other functionalities such as Deleting users, accessing User’s Audit and accessing system Errors.
Chapter 6 - Implementation

When a system is finally implemented conflicts will surely arise and made visible if the implementation of this project being worked on is poorly managed and organised. During the implementation stage users play a significant role since an important factor to achieve successful implementation is adequate user support, especially with web-based applications since user requirements and Internet technology are continuously changing. A system is considered to fail if it is not utilised the way it was intended to or if it is not used at all so that anticipating potential problems and applying appropriate corrective strategies can increase the chances for system success.

Implementation

Nowadays, The Web technology brings challenges. We are faced with competing architecture, platforms and tools. Most of them are still evolving. We are being challenged to explore new methodologies and best practices to address Web-specific development issues such as maintenance of content-rich Web applications, security, application scalability and ever-increasing demand for fast system deployment by customers. For many developers, building Web applications is a “mad science” (callaway,1997). The traditional system development methods such as prototyping methods can still be effective. But they need to be adapted and enriched in the new development environment.

The technology enables us to deliver application easily and quickly and provides more efficient methods to do maintenance and updates of our applications. The web application development methodology is more responsive to user needs and quicker to customize applications for specific users. The web-based applications promise a number of advantages over traditional non-web-based applications as follows:

The strengths of Web Application Development Methodology can provide an iterative process of system development. Working closely with users, developers design and build a scaled-down functional model of a desired system. The developer demonstrates the working model to the user and then continues to develop the prototype based on the feedback received until developer and the user agree that the prototype is ‘good enough’. At that point, the developer either throws away the prototype or starts building the real system (the throw away prototype is used solely to understand the user’s requirements) or completes any remaining work on the prototype and releases the prototype as the final product (evolutionary prototype).

These strengths would be used to improve system development by creating a prototype that would meet the functional requirements as obtained from the users and help eliminate design flaws that might occur if requirements were not fully captured the first time.

Implementation Code

The following projects make up the solution of the Skyver Project

- Website
- Application
- Mobile Communication
Website

The website project contains the web pages that will be presented as the frontend of the solution.

Application

The application contains the interface with the mobile. The application output is an executable that uses the other projects in the solution.
In order to communicate with the mobile phone a third party dlls has been used. The dll names are:

- GSMCommServer
- GSMCommShared
- GSMCommunication
- PDUConvertor

These GSM communication libraries were download from the following site: http://www.scampers.org/steve/sms/libraries.htm.

The Application project uses the following dlls in order to communicate with the Mobile Phone.

Below is sample code to communicate with the GSMComm dlls.

```csharp
public bool CheckMobilePort()
{
    Cursor.Current = Cursors.WaitCursor;
    GsmCommMain comm = new GsmCommMain(port, baudRate, timeout);
    try
    {
        comm.Open();
        while (!comm.IsConnected())
        {
        }
    }
    catch
    {
        // Handle exception
    }
    finally
    {
        Cursor.Current = Cursors.Default;
    }
    return comm.IsConnected();
}
```
The comm object initialized in the following line of code

```csharp
GsmCommMain comm = new GsmCommMain(port, baudRate, timeout);
```

is used throughout the application as an interface between the application and the mobile phone.

The main form of the application is loaded once the the comm object communicates successfully with the mobile port and a connection between the port and the mobile phone is established:

```csharp
private void Skyver_Load(object sender, EventArgs e)
{
    try
    {
        // Prompt user for connection settings
        int port = GsmCommMain.DefaultPortNumber;
        int baudRate = GsmCommMain.DefaultBaudRate;
        int timeout = GsmCommMain.DefaultTimeout;
        sessionUser = new tblUser();
        frmConnection dlg = new frmConnection();
        dlg.StartPosition = FormStartPosition.CenterScreen;
        dlg.SetData(port, baudRate, timeout);
        if (dlg.ShowDialog(this) == DialogResult.OK)
        {
            dlg.GetData(out port, out baudRate, out timeout, out sessionUser);
            ApplicationSession.User = sessionUser;
        }
    }
    else
    {
        Close();
        return;
    }
}
```
{ 
retry = false;
try 
{
    Cursor.Current = Cursors.WaitCursor;
    comm.Open();
    LoadSkinDetails();
    SetPhoneDetails();
    Cursor.Current = Cursors.Default;

    //add audit user logged in

    AddAuditRequest auditRequest = new AddAuditRequest
    {
        User = ApplicationSession.User,
        action = "User Logged In",
        AuditTypeId = 3
    };

    auditManager.AddAudit(auditRequest);
}
catch (Exception) 
{
    Cursor.Current = Cursors.Default;
    if (MessageBox.Show(this, "Unable to open the port.", "Error",
        MessageBoxButtons.RetryCancel, MessageBoxIcon.Warning) == DialogResult.Retry)
    
        retry = true;
    else
    {
        Close();
        return;
    }
}
while (retry);
}
catch (Exception ex) 
{
    AddSystemErrorRequest errorRequest = new AddSystemErrorRequest
    {
        User = ApplicationSession.User,
        Description = "Error on Loading Skyver Application",
        StackTrace = ex.Message.ToString()
    };

    systemErrorManager.AddSystemError(errorRequest);
}
}

The credentials used in the Connection form are saved in a static class ApplicationSession, which will be used through out the session of the application, to record actions done by user, such as uploading of SMS and Contacts.

The applications contains also all the forms and usercontrols used in the Skyver Mobile application, and the code behind any events. The following code depicts the event triggered when user click the upload Contacts button.

private void btnUploadContacts_Click(object sender, EventArgs e) 
{
    Cursor.Current = Cursors.WaitCursor;

    List<tblContact> contactList = new List<tblContact>();

    for (int i = 0; i < this.dgvContacts.Rows.Count; i++)
The above code retrieves all the selected contacts from the user interface and uploads them to the server. In order to do this the Workers dll is invoked, which then uses the Core and Data dlls to communicate with the database. These mentioned dlls are explained in more details in the following sections.

Workers

The workers project contains the business logic of the application. Each entity (such as user, sms and contact) has its own manager and its own manager has its own methods. Below is part of the Contact Manager class, with implemented methods.

```csharp
public class ContactManager : Manager, IContactManager
{
    public ContactManager(ModelContainer context)
        : base(context)
    {
    }

    public AddContactResponse AddContact(AddContactRequest request)
    {
        // Implementation of AddContact method
        // ...
    }
```
// Do validation here
ValidationResult validationResult = new ValidationResult();
validationResult.Valid = true;

var result = new AddContactResponse { ValidationResult = validationResult };

tblContact contact = new tblContact();

contact.Name = request.Name;
contact.MobileNumber = request.MobileNumber;
contact.ArchivedDate = DateTime.Now;
contact.tblUser = request.User;
// save user not to retrieve all the time
contact.Deleted = false;

ContactDao contactDao = new ContactDao();
contactDao.Context = (SuperscoutModelContainer)Context;

contact = contactDao.Save(contact);
result.ItemId = contact.Id;
return result;

public bool ContactExists(int userId, string Name, string MobileNumber)
{
    ContactDao contactDao = new ContactDao();
    contactDao.Context = (SuperscoutModelContainer)Context;

    return contactDao.CheckContactExistsByUser(userId, Name, MobileNumber);
}

Core

The core project contains the classes used to represent the entities in the project. To encapsulate the entities the ADO.NET Entity Framework is used. The ADO.NET Entity Framework an object-relational mapping (ORM) framework for the .NET Framework.

Data

The layer is the actual layer that communicates with the Database. The Data layer contains various entity DAOs (Data Access Objects) that are used to do CRUD (Created, Retrieve, Update and Delete) functions. Below is a sample of the ContactDao

public class ContactDao : Dao<tblContact, int>, IContactDao
{
    public bool CheckContactExistsByUser(int userId, string Name, string MobileNumber)
    {
        bool contactExists = false;

        tblContact currentContact = Context.tblContactSet.FirstOrDefault
            (contact => (contact.Name == Name) && (contact.MobileNumber == MobileNumber) &&
            (contact.tblUser.Id == userId));

        if (currentContact != null)
            contactExists = true;

        return contactExists;
    }
}

Please note that basic functions that are used for all the Daos are in the Dao class. Each specific Dao (ex. ContactDao) inherits from the Dao class. The ContactDao class contains
only the specific functions for that particular entity. Below is part of the code for the DAO class.

```csharp
public abstract class Dao<T, IdT> : IDao<T, IdT> where T:class
{
    public virtual T GetById(IdT id)
    {
        string parentEntitySet = FindParentEntitySet();
        T entity = ContextcreateQuery<T>(string.Format("select value o from oftype({0}, {1}) as o where o.Id = {2}", parentEntitySet, typeof(T).FullName, id)).FirstOrDefault();
        return entity;
    }

    public virtual List<T> GetAll()
    {
        return GetAllAsQueryable().ToList();
    }

    public virtual IQueryable<T> GetAllAsQueryable()
    {
        string parentEntitySet = FindParentEntitySet();
        IQueryable<T> entities = Context.createQuery<T>(string.Format("select value o from oftype({0}, {1}) as o", parentEntitySet, typeof(T).FullName));
        return entities;
    }

    public virtual List<T> GetAll(int start, int limit)
    {
        string parentEntitySet = FindParentEntitySet();
        List<T> entities = Context.createQuery<T>(string.Format("select value o from oftype({0}, {1}) as o order by o.Id skip(" + start + ") limit(" + limit + ")", parentEntitySet, typeof(T).FullName)).ToList();
        return entities;
    }

    public virtual T Save(T entity)
    {
        string parentEntitySet = FindParentEntitySet();
        Context.AddObject(parentEntitySet, entity);
        Context.SaveChanges();
        return entity;
    }

    public virtual T Update(T newEntity, T originalEntity)
    {
        string parentEntitySet = FindParentEntitySet();
        EntityKey key;
        if (originalEntity == null)
        {
            Context.AttachTo(parentEntitySet, newEntity);
            key = Context.CreateEntityKey(parentEntitySet, newEntity);
            EntityStateEntry en = Context.ObjectStateManager.GetObjectStateEntry(key);
            en.SetModified();
            Context.Refresh(RefreshMode.ClientWins, newEntity as object);
        }
        else
        {
            key = Context.CreateEntityKey(parentEntitySet, originalEntity);
            if (key == null)
            {
                Context.AttachTo(parentEntitySet, newEntity as object);
            }
            else
            {
                Context.Attach(originalEntity as IEntityWithKey);
                Context.ApplyPropertyChanges(key.EntitySetName, newEntity as object);
            }
        }
    }
}
```
ObjectStateEntry en = Context.ObjectStateManager.GetObjectStateEntry(key);
en.SetModified();
Context.Refresh(RefreshMode.ClientWins, originalEntity as object);
}
this.CommitChanges(originalEntity as object);
return originalEntity;
}

public virtual void Delete(T entity)
{
    Context.DeleteObject(entity as object);
}

The Dao class is an abstract class and therefore cannot be initialized but referenced from another class. The Dao class communicates with the ADO.Net entity framework to insert the data in the database. The instance of the Entity Framework is the Context object reference used in the above code.

Util

The util project contains common utilities that are used throughout the projects of the solution and isolated in one class. The aim of this project to avoid duplication of code through the different projects in the Solution.

Workers.Test

The workers.Test project is used for unit testing. The aim of this project is to unit test each business logic function that was developed. Below is a unit test for uploading an sms to the database from the SMSManagerTest class.

[TestMethod]
public void AddSms()
{
    var userManager = WorkersHelper.GetUserManager();
    var smsTypeManager = WorkersHelper.GetSmsTypeManager();
    var smsStatusManager = WorkersHelper.GetSmsStatusManager();

    tblUser user = userManager.GetUserById(1);
    tblSmsType smsType = smsTypeManager.GetSmsTypeById(1);
    tblSmsStatus smsStatus = smsStatusManager.GetSmsStatusById(1);
    var smsManager = WorkersHelper.GetSmsManager();

    AddSmsRequest addSmsRequest = new AddSmsRequest
    {
        Recipient = "998383838",
        TextMessage = "This is an sms...",
        user = user,
        smsStatus = smsStatus,
        smsType = smsType
    };

    smsManager.AddSms(addSmsRequest);
}
Setup

The .Net Setup project is used as a packaging tool for the Skyver Mobile Solution. This encapsulates the executable and the dlls from the projects and packages them into an installer.

Testing Strategy

A test strategy can be described as a plan on how one is going to approach project testing. Its main goal is to audit the precision of the software and to determine whether it meets the user requirements stated at the beginning of the project. The process of testing will ensure that system is:

1. Accurate in the design.
2. Meet user requirements.

As a result we can say that the software developed was tested for errors, project logic errors and missing requirements before it will be finally released to the public.

Below we can find some aspects that need to be tackled in a testing strategy.

Project Name: Skyver

Overview

- **Testing Stage:** During this stage we need to identify the type of testing to be undertaken.

- **Testing Types:**
  
  - **Unit Testing:** Once each module of the system is developed it is tested using testing data to ensure that it meets its specific requirements. A technology called “White box testing” is used to test each function in a top down fashion. This is considered a valid test strategy as testing can be carried out whilst it has been developed.

  - **Integration Testing:** is the process of software testing where individual software procedures are combined together and tested as one group. This process occurs right after unit testing, and takes as input functions that have been unit tested, puts them together in larger aggregates, applies tests which were defined in the integration test plan and delivers the integrated system ready for system testing as its output. As a conclusion we can say that the purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items.

  - **System Testing:** is testing of software or carried out on a complete, integrated system to evaluate the system's compliance with its specified requirements. Unlike unit testing system testing falls under the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. As mentioned in integration testing system testing takes, as its input, all of the "integrated" software components that have successfully passed. System testing is a more limiting type of testing since it tends to detect defects both within the "inter-assemblages" and also within the system as a whole.
- **Usability Testing:** is the process carried out to evaluate a product by testing it on users. It focuses on measuring human-made product capacity to meet its intended purpose. Web sites or web applications are typical examples of such products which commonly benefit from usability testing.

- **Scheduled for:** Here a start and end date for all the testing process has to be issued.

- **Location:** Where testing is going to be carried out. (Example: Test Centre Lab X Malta)

- **Participants:** Here it is important to identify the people who will be involved in testing. If resources have been not yet nominated then the skills required have to be outlined.
  - **Testing Manager:** Angelo Dimech
  - 3 Testers will be nominated and need to have a good understanding of all the processes being carried out also being familiar with the manual processes involved.

**Testing Environment**

- **Environment:** The environment to be used during testing has to be outlined here. A test area will need to be set up one for users to carry out user acceptance testing of data input and other processing. Another area will need to be set up for test reports.

- **Equipment Environment:** Here we have to determine how many PCs would be needed and its essential that they will be of different operating systems, in order to hit all areas.

- **Data:** The data to be used for testing has to be identified here.

- **Backup:** It’s important to identify how often data backups should be undertaken and who has responsibility for backups. Also cover how long backups should be retained.

- **Restore:** We also need to identify how and when a restore should take place.

**Testing Procedures**

- **Problem Identification:** It is important to identify the procedure to be used when a tester finds a suspected defect. There should be a nominated resource to receive all defects. In some cases there may be more than one resource e.g. one person for applications problems and one for operational problems.

- **Defect Rectification:** In this scenario we need to identify how the defect will be managed once received. This procedure would normally be under the control of the person or people rectifying the defect.

- **Re-Testing:** Procedure for re-testing rectified defects has to be identified in this stage.
• **Sign-off Testing Instructions**: Here total testing will be signed off. This includes all activities in the "Test Plan" and any rectification of defects.

**Release Management**

**Release Management Considerations**: One of the major risks in testing is not having a proper Release Management Procedure. The same program is modified by two people at the same time, and only one modification gets into production. It is important to put in place a proper release management process.

**Software**

• **Test Software Management**: Identification of any software that will be used to manage the testing will be identified here. This includes the organisation of the activities to be tested, and the management of defects.

• **Testing Software**: If any automated testing software such as WinRunner is to be used, detail the software. Also identify if the software will need to be purchased.

• **Performance Testing Software**: Detail the software if any performance testing software is to be used. Identify if the software will need to be purchased.

**Conclusion**

If all the above will be carried out as follows and in professional way, the software being developed will surely be released bug free thus meeting all requirements as discussed in the analysis staged.

**Test Cases**

Test cases for both the console application and web application where carried out. These can be found below.

*Console Application Testing*

<table>
<thead>
<tr>
<th>Test Case: 001 – Check Login</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>Username: adimech</td>
</tr>
<tr>
<td>Username: adimech</td>
</tr>
</tbody>
</table>
### Test Case : 002 - Check Mobile Connectivity

**Required Information:** Mobile Connected to Port com16

<table>
<thead>
<tr>
<th>Input</th>
<th>Expected Output</th>
<th>Actual Output</th>
<th>Pass/Fail</th>
<th>Fixes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Com-Port:</strong> 16</td>
<td>Successfully connected to the phone</td>
<td>Successfully connected to the phone</td>
<td><strong>Passed</strong></td>
<td><strong>None</strong></td>
</tr>
<tr>
<td><strong>Baud rate:</strong> 19200</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Timeout (ms):</strong> 300</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Com-Port:</strong> 50</td>
<td>Connection error: Unable to open port COM16: Access to the port is ‘COM 32’ is denied.</td>
<td>Connection error: Unable to open port COM16: Access to the port is ‘COM 32’ is denied.</td>
<td><strong>Passed</strong></td>
<td><strong>None</strong></td>
</tr>
<tr>
<td><strong>Baud rate:</strong> 19200</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Timeout (ms):</strong> 300</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Com-Port:</strong> 32</td>
<td>Connection error: Unable to open port COM16: Access to the port is ‘COM 32’ is denied.</td>
<td>Connection error: Unable to open port COM16: Access to the port is ‘COM 32’ is denied.</td>
<td><strong>Passed</strong></td>
<td><strong>None</strong></td>
</tr>
<tr>
<td><strong>Baud rate:</strong> 19200</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Timeout (ms):</strong> 300</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Test Case : 003 – Load Mobile Details after successful Connection

**Required Information:** Messages: 102, Contacts 431

<table>
<thead>
<tr>
<th>Input</th>
<th>Expected Ouput</th>
<th>Actual Output</th>
<th>Pass/Fail</th>
<th>Fixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td><strong>Operator:</strong> go mobile</td>
<td><strong>Operator:</strong> go mobile</td>
<td><strong>Passed</strong></td>
<td><strong>None</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Phone Name:</strong> Sony Ericsson K750i</td>
<td><strong>Phone Name:</strong> Sony Ericsson K750i</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Battery Strength: numeric value</td>
<td>Battery Strength: numeric value</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------------</td>
<td>--------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Signal Strength: numeric value</td>
<td>Signal Strength: numeric value</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Current User: Angelo Dimech</td>
<td>Current User: Angelo Dimech</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total SMS: 102</td>
<td>Total SMS: 102</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Contacts: 431</td>
<td>Total Contacts: 431</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Test Case : 004 – Check Tree Menu Links

<table>
<thead>
<tr>
<th>Input</th>
<th>Expected Ouput</th>
<th>Actual Output</th>
<th>Pass/Fail</th>
<th>Fixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clicked: Phone Details</td>
<td>Phone Details Loaded</td>
<td>Phone Details Loaded</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td>Clicked: Contact&gt;Phone</td>
<td>Phone Contacts loaded</td>
<td>Phone Contacts loaded</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td>Clicked: Contact&gt;SIM</td>
<td>SIM Contacts loaded</td>
<td>SIM Contacts loaded</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td>Clicked: Messages&gt;Phone</td>
<td>Phone messages loaded</td>
<td>Phone messages loaded</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td>Clicked: Messages&gt;SIM</td>
<td>SIM messages loaded</td>
<td>SIM messages loaded</td>
<td>Passed</td>
<td>None</td>
</tr>
</tbody>
</table>

## Test Case : 005 – Check Main Menu Links

<table>
<thead>
<tr>
<th>Input</th>
<th>Expected Ouput</th>
<th>Actual Output</th>
<th>Pass/Fail</th>
<th>Fixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clicked: Help&gt;About</td>
<td>About Form Loaded</td>
<td>About Form Loaded</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td>Clicked: Retrieve&gt;Read Phone SMS</td>
<td>Phone SMS loaded</td>
<td>Phone SMS loaded</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td>Clicked: Retrieve&gt;Read SIM SMS</td>
<td>SIM SMS loaded</td>
<td>SIM SMS loaded</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td>Clicked: Retrieve&gt;Read SIM Contacts</td>
<td>SIM contacts loaded</td>
<td>SIM contacts loaded</td>
<td>Passed</td>
<td>None</td>
</tr>
</tbody>
</table>
**Test Case : 006 – Check Contact Search**

<table>
<thead>
<tr>
<th>Input</th>
<th>Expected Output</th>
<th>Actual Output</th>
<th>Pass/Fail</th>
<th>Fixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search query: “cu”</td>
<td>4 results</td>
<td>About Form Loaded</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td>Search query: “cust”</td>
<td>2 results</td>
<td>Phone SMS loaded</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td>Search query: “”</td>
<td>All Contacts should be retrieved</td>
<td>No Action</td>
<td>Failed</td>
<td>FIX 02</td>
</tr>
</tbody>
</table>

**Test Case : 007 – Check SIM Search**

<table>
<thead>
<tr>
<th>Input</th>
<th>Expected Output</th>
<th>Actual Output</th>
<th>Pass/Fail</th>
<th>Fixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search query: “c”</td>
<td>57 results</td>
<td>57 results</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td>Search query: “did”</td>
<td>3 results</td>
<td>3 results</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td>Search query: “”</td>
<td>All Contacts should be retrieved</td>
<td>No Action</td>
<td>Failed</td>
<td>FIX 03</td>
</tr>
</tbody>
</table>

**Test Case : 008 – Check if correct number of contacts are retrieved**

**Required Information:** SIM Contacts:84, Phone Contacts:347

<table>
<thead>
<tr>
<th>Input</th>
<th>Expected Output</th>
<th>Actual Output</th>
<th>Pass/Fail</th>
<th>Fixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click: Read SIM Contacts</td>
<td>84 results</td>
<td>84 results</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td>Click: Read Phone Contacts</td>
<td>347 results</td>
<td>347 results</td>
<td>Passed</td>
<td>None</td>
</tr>
</tbody>
</table>
### Test Case: 009 – Check if correct number of messages are retrieved

**Required Information:** SIM Messages: 1, Phone Messages: 101

<table>
<thead>
<tr>
<th>Input</th>
<th>Expected Output</th>
<th>Actual Output</th>
<th>Pass/Fail</th>
<th>Fixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click: Read SIM Messages</td>
<td>1 results</td>
<td>1 results</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td>Click: Read Phone Messages</td>
<td>101 results</td>
<td>101 results</td>
<td>Passed</td>
<td>None</td>
</tr>
</tbody>
</table>

### Test Case: 010 – Check if contact checkbox selection works

<table>
<thead>
<tr>
<th>Input</th>
<th>Expected Output</th>
<th>Actual Output</th>
<th>Pass/Fail</th>
<th>Fixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click: Phone Contacts&gt; Select All</td>
<td>347 Checkboxes checked</td>
<td>347 Checkboxes checked</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td>Click: SIM Contacts&gt; Select All</td>
<td>84 Checkboxes checked</td>
<td>84 Checkboxes checked</td>
<td>Passed</td>
<td>None</td>
</tr>
</tbody>
</table>

### Test Case: 011 – Check if messages checkbox selection works

<table>
<thead>
<tr>
<th>Input</th>
<th>Expected Output</th>
<th>Actual Output</th>
<th>Pass/Fail</th>
<th>Fixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click: Phone Messages&gt; Select All</td>
<td>101 Checkboxes checked</td>
<td>101 Checkboxes checked</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td>Click: SIM Messages&gt; Select All</td>
<td>1 Checkbox checked</td>
<td>1 Checkbox checked</td>
<td>Passed</td>
<td>None</td>
</tr>
</tbody>
</table>

### Test Case: 012 – Check if upload of contacts and messages works

<table>
<thead>
<tr>
<th>Input</th>
<th>Expected Output</th>
<th>Actual Output</th>
<th>Pass/Fail</th>
<th>Fixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click: Phone Contacts&gt; Select All &gt; Upload Contacts</td>
<td>Contacts Successfully Uploaded</td>
<td>Contacts Successfully Uploaded</td>
<td>Passed</td>
<td>None</td>
</tr>
</tbody>
</table>
### Web Application Testing

#### Test Case: 013 – Register User

<table>
<thead>
<tr>
<th>Input</th>
<th>Expected Output</th>
<th>Actual Output</th>
<th>Pass/Fail</th>
<th>Fixes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name:</strong> Angelo</td>
<td>Password field does not match the confirm password field</td>
<td>Password field does not match the confirm password field</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td><strong>Surname:</strong> Dimech</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Address 1:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Address 2:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Country:</strong> Malta</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Email:</strong> <a href="mailto:adimech@gmail.com">adimech@gmail.com</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Username:</strong> adimech</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Password:</strong> angelo</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Confirm Password:</strong> angelo2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Name: Angelo        | All the above fields should be required                                          | All the above fields are required please make sure each field is filled in!! | Passed    | None  |
| Surname:            |                                                                                  |                                                                               |           |       |
| Address 1:          |                                                                                  |                                                                               |           |       |
| Address 2:          |                                                                                  |                                                                               |           |       |
| **Country:** Malta |  |  |  |  |
| **Email:** angelo.dimech@gmail.com |  |  |  |  |
| **Username:** |  |  |  |  |
| **Password:** angelo | User Successfully uploaded | User Successfully uploaded | Passed | None |
| **Confirm Password:** test123 |  |  |  |  |
| **Name:** Angelo |  |  |  |  |
| **Surname:** Dimech |  |  |  |  |
| **Address 1:** Straight Street |  |  |  |  |
| **Address 2:** Sliema |  |  |  |  |
| **Country:** Malta |  |  |  |  |
| **Email:** angelo.dimech@gmail.com |  |  |  |  |
| **Username:** adimech |  |  |  |  |
| **Password:** angelo | Invalid Username and Password | Invalid Username and Password | Passed | None |
| **Confirm Password:** angelo2 |  |  |  |  |

### Test Case: 014 – Check Login

<table>
<thead>
<tr>
<th>Input</th>
<th>Expected Output</th>
<th>Actual Output</th>
<th>Pass/Fail</th>
<th>Fixes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Username:</strong></td>
<td>Invalid Username and Password</td>
<td>Invalid Username and Password</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td><strong>Password:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Username:</strong> adimech</td>
<td>Invalid Username and Password</td>
<td>Invalid Username and Password</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td><strong>Password:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Username:</strong> adimech</td>
<td>Users logged in successfully</td>
<td>Users logged in successfully</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td><strong>Password:</strong> test123</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Test Case: 015 – Skyver Application Download

<table>
<thead>
<tr>
<th>Input</th>
<th>Expected Output</th>
<th>Actual Output</th>
<th>Pass/Fail</th>
<th>Fixes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test Case : 016 – Messages</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Input</strong></td>
<td><strong>Expected Output</strong></td>
<td><strong>Actual Output</strong></td>
<td><strong>Pass/Fail</strong></td>
<td><strong>Fixes</strong></td>
</tr>
<tr>
<td>Username Logged In and click on messages</td>
<td>Messages should load successfully</td>
<td>Messages loaded successfully</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td><strong>New Group Created:</strong> Family</td>
<td>New Group Family should be listed successfully under other folders.</td>
<td>Family Group was created and placed under the other folders</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td>Delete one particular message</td>
<td>The particular message should be deleted and sent to the trash folder</td>
<td>Message deleted and sent to trash folder</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td>5 messages where selected and the delete button is clicked</td>
<td>5 messages should be deleted from the relevant folder and moved to the trash folder</td>
<td>5 messages where deleted and the 5 messages where moved successfully to the trash folder</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td>5 messages where selected and moved to the new folder “Family” which was just created.</td>
<td>5 messages should be moved to the new folder and removed from the current folder.</td>
<td>5 messages where successfully moved to the Family folder</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td>Test Case : 017 – Search Messages</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Input</strong></td>
<td><strong>Expected Output</strong></td>
<td><strong>Actual Output</strong></td>
<td><strong>Pass/Fail</strong></td>
<td><strong>Fixes</strong></td>
</tr>
<tr>
<td><strong>Search Query:</strong> “+35699829833”</td>
<td>10 messages should be displayed in the grid</td>
<td>10 messages where successfully displayed in the grid</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td><strong>Search Query:</strong>””</td>
<td>All results should be displayed in the grid</td>
<td>All results were displayed in the grid</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td><strong>Search Query:</strong>”+44”</td>
<td>No Results should be displayed</td>
<td>No Matches Found</td>
<td>Passed</td>
<td>None</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Case : 018– Permanently Delete Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input</strong></td>
</tr>
<tr>
<td>Select Message from Trash folder and press delete</td>
</tr>
</tbody>
</table>
## Test Case: 019 – Contacts

<table>
<thead>
<tr>
<th>Input</th>
<th>Expected Output</th>
<th>Actual Output</th>
<th>Pass/Fail</th>
<th>Fixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username Logged In and click on Contacts</td>
<td>Contacts should load successfully</td>
<td>Contacts loaded successfully</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td><strong>New Group Created:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Family</td>
<td>New Group Family should be listed successfully under other folders.</td>
<td>Family Group was created and placed under the other folders</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td>Delete one particular contact</td>
<td>The particular contact should be deleted and sent to the trash folder</td>
<td>contact deleted and sent to trash folder</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td>5 contacts where selected and the delete button is clicked</td>
<td>5 contacts should be deleted from the relevant folder and moved to the trash folder</td>
<td>5 contacts where deleted and the 5 contacts where moved successfully to the trash folder</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td>5 contacts where selected and moved to the new folder “Family” which was just created.</td>
<td>5 contacts should be moved to the new folder and removed from the current folder.</td>
<td>5 contacts where successfully moved to the Family folder</td>
<td>Passed</td>
<td>None</td>
</tr>
</tbody>
</table>
## Test Case: 020 – Search Contact

<table>
<thead>
<tr>
<th>Input</th>
<th>Expected Output</th>
<th>Actual Output</th>
<th>Pass/Fail</th>
<th>Fixes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Search Query:</strong> “Dimech”</td>
<td>2 Contacts should be displayed in the grid</td>
<td>2 Contacts where successfully displayed in the grid</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td><strong>Search Query:</strong> “”</td>
<td>All results should be displayed in the grid</td>
<td>All results were displayed in the grid</td>
<td>Passed</td>
<td>None</td>
</tr>
<tr>
<td><strong>Search Query:</strong> “zz”</td>
<td>No Results should be displayed</td>
<td>No Matches Found</td>
<td>Passed</td>
<td>None</td>
</tr>
</tbody>
</table>

## Test Case: 021 – Permanently Delete Messages

<table>
<thead>
<tr>
<th>Input</th>
<th>Expected Output</th>
<th>Actual Output</th>
<th>Pass/Fail</th>
<th>Fixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Contact from Trash folder and press delete</td>
<td>Contact should be deleted completely and the trash folder should be refreshed without that particular Contact.</td>
<td>Contact Permanently deleted and the trash grid was successfully refreshed.</td>
<td>Passed</td>
<td>None</td>
</tr>
</tbody>
</table>
User Guide

Pre-Requisites

- Windows XP/Vista or superior
- Pentium 4 1.5GHz or Athlon XP 1500+ Processor or higher
- 512MB or higher
- 1 GB free hard disk space
- .NET Framework 3.5

Setting up the Database (For administration users, where the database will be installed on the server.)

The following are the steps required to manually set up the database.

1. Click on SQL Server Management Studio
2. Right Click on databases and add new Database and name it skyver.
3. Run the script SkyverSchema.sql which can be found in the scripts folder.
4. Run the script SkyverData Reference.sql which can be found in the scripts folder.
Figure 15: Database Scripts

After the above mentioned steps the Database will be finally up and running.

User's guide for using Skyver Solution

Register on Skyver

In order to download the Skyver mobile application setup it is mandatory to register online through the Skyver web site. Registration can be done from the register link found on the homepage. The screenshots below shows a typical example.
Figure 16: Register Link

Figure 17: Register Online

All fields should be entered in order for the registration to be successful. Once registration is successful, a welcome page is shown on the website.
Once users have registered the setup can be downloaded from the download link.

Once the link is clicked, the following dialogue box will appear. Click on the save button to download the file onto your computer.
Once setup file is downloaded double click on SkyverSetup file, follow instruction in the application wizard and the application will be installed.

Once installed, click on the application icon.

The first screen to show will be the following.
In the textboxes username and password, enter the same username and passwords user to register in the website, then click on check. If the username and password match, the username and password text boxes will be disabled and the mobile connection settings will be enabled.

Now make sure your mobile is connected through USB with the data cable and the required driver is installed. Once the mobile is connected either allow your pc to locate the mobile drivers automatically from the internet, or install the drivers manually from the mobile driver folder.

To know on which port your mobile is connected, do the following procedure.

Right click on my computer from the desktop, and then click on the option manage. Once the manage option is clicked the computer management screen will appear.
Figure 23: Device Manager

After that click on Device Manager, Open the Modems tree view and right click on the connected Mobile modem and select Properties.

Once the mobile properties window is open, click on the Advanced Tab, click on the port properties below and the port number will be highlighted.

Use this port number to enter it in the Skyver Connection Dialog. (It is recommended to keep the Baud rate and Timeout rate as the default values that appear in the Login Dialogue).

After that click on the test button in the Skyver Connection dialog and if the application connects successfully to the Mobile the Mobile connection Text boxes will be disabled and the login button will be highlighted. A message box will denote if the mobile is successfully connected to the phone or if any problems have occurred during the connection with the mobile phone.

This should lead you to Skyver Mobile Application Main Screen.
The first screen will show the Mobile Phone details.

Retrieve of Phone SMS

The phone SMS can be retrieved either from the tree view on the left hand side of the menu or else from the Retrieve menu item under the main Menu Item Phone SMS.

After the SMS are retrieved, all the retrieved SMS are displayed in the Data grid. A total number of retrieved SMS is denoted at the bottom of the data grid.

If the user wants to upload any SMS user is required to select the required SMS from the Data grid. A check all function is also available to select all SMS, also user has the option to load only particular SMS. The search function can be used to retrieve particular SMS. Once user has selected his/her desired SMS, user can click on the upload button to upload the SMS in to the online Server. A confirmation Message will denote the user with the number of SMS uploaded. (Please note that duplicate SMS will not be uploaded to server).
Similar functionalities are available for the following:

- Retrieval of SIM SMS
- Retrieval of Phone Contacts
- Retrieval of SIM Contacts
Website User Guide

Login

Once user has registered, user can log in the website with the credentials previously registered.

![Web Login Image]

**Figure 26: Web Login**

Once a user can log, he/she can view the Messages by clicking on the message button found under the banner. The following screen will list all the messages uploaded by the user.

User’s messages uploaded and retrieve from the Skyver website

A user can manage the Messages from the website by doing the following functions.

**View all messages**

This function will list all the messages uploaded by the user in the data grid.
Delete messages

Users have the possibility to delete any un-required messages by clicking on the delete link in the data grid near the required SMS. These messages will then be transferred to the trash folder. Users can permanently delete SMS by clicking on the delete button once in the trash folder.

![Figure 27: Web Delete Messages](image)

Search Messages

At the top most of the data grid there is a search functionality to search for any particular SMS. In order to search for a particular SMS user is required to type any particular text in the provided text box and click the Search button. The matching SMS can be found below in the data grid.

![Figure 28: Search Messages](image)

Group messages to a particular group
User has the possibility to group any particular messages to a particular group (ex Family). In order to do this functionality user has to create a group on the left hand side of the screen under the sent folder from the create a group link. The following page will appear.

![Create Group](image)

**Figure 29: Create Group**

User is required to enter the name of the group as well as the group description. Once the group is saved the group will appear on the left hand side of the screen.

![Group Created](image)

**Figure 30: Group Created**

Once a group is created user can sort out and manage the SMS into groups. This can be done by viewing all the messages again, and select any particular SMS and bind the SMS to the required group.

View individual messages into more details
The following functionality is useful to view any SMS in more detail. The data grid will only display limited amount of information and this function will enable the user to view the SMS in greater information.

![Image of Skyver interface](Image)

**Figure 31: View Message**

A user can also manage the Contacts from the website by also doing the following functions.

- View all Contacts
- Delete Contact
- Group Contacts
- View individual contact into more details

(Please refer to SMS function for further details)

**View all messages by a particular Contact in a thread**

This particular function is usual to load all the messages of a particular contact after each other. This function will show the history of all the messages of a particular contact.

**Figure 32: View Messages per User**

**Edit particular contact**

User has the possibility to edit a particular contact and update his/her particular details.
Chapter 7 – Summary and Conclusions

Objectives that have been Met

- Successfully connected mobile with backend application
- Retrieved all mobile information, including messages and Contacts from mobile phone
- Uploaded mobile information to online server
- Retrieved mobile information from the server and placed it successfully in the web application
- Management of Contacts and Messages was carried out successfully

Processes Carried Out and Decisions Taken

Application - Mobile Phone Communication

In order to retrieve the contacts and SMS from the mobile phone a mechanism was required. This was one of the most critical decisions in the development of the system. Various solutions were available:

- Communication Directly with AT commands
- Commercial library MobileEdit (www.mobiledit.com)
- Open source library (GSMComm) (http://www.scampers.org/steve/sms)

In order to have a global free solution for all the mobile phones the library GSMComm was used.

Development Platform

A decision after the design was finalized was which development platform to use. The shortlisted platforms were JAVA and .NET. The .NET platform was used in developing this system.

RDBMS

SQL Server 2005 was used as a database from a selection of different RDBMS systems including:

- Microsoft Access
- Oracle
- MySQL
What I Have Achieved from this Project

Developing this project, helped me go through all the list cycle of an IT project development start from requirements, re-search to find a solution, designing the solution, Implementing it and finally testing the solution and having a user acceptance Testing. Other experience achieved was how to integrate data, behaviour and processes into the SSADM and Windows/Web Application methodology. This project also taught me how to manage a project, research for solutions and how design/Feasibility study are critical before initialization of the implementation of the project.

Success Factors

Moreover, in order to implement the project successfully, a number of critical success factors can be considered in the following manners:

The schedule is realistic - The most common cause of failure is an unrealistic schedule, usually imposed without the input or the concurrence of the project manager or team members. Most often, the imposed schedules have no rationale for specific dates, but are only means to "hold the project manager to a schedule." A realistic schedule will include all the required tasks to implement the project along with their durations, assigned resources and task dependencies.

Commitment to Schedule - The Working Group forum is where many issues will be resolved. To expedite a resolution to a problem or/and issue, ad-hoc meetings may be called at short notice. Unlike the workshops or checkpoint meetings, ad-hoc meetings at short notice will involve only specific users to address some very specific issues or some quick clarifications on certain ambiguities that may arise.

There will be one informal review for each interim deliverable. A deliverable is interim in the sense that it is submitted and presented to designated users for comment and review after a logical section has been developed. The reviewer will record their comments on the Comment Form. The Working Group ensures that all comments be collected within the expected time frame (based on 3 to 5 working days, a fairly sufficient amount of time according to past experience) after the deliverable has been submitted for review. Thereafter, only further comments on the amended items will be accepted; any other comments will be managed as Request for Change. The project team Project Manager will assume full responsibility in seeing that the intended corrections/modifications are rolled in properly. After making all the required corrections/modifications, the product will be submitted for final review.

Focus on Scope - It is vital that the project team constantly monitors the project progress in a way that it does not veer off course otherwise it may adversely impact the project schedule. All parties involved in the project have to stay focused on meeting the objectives and the essential requirement of the project in order to avoid any occasion of wasting time and resources in unrelated and irrelevant matters. Frequent references to the scope of the FS and SA&D should be made.
Level of Detail - The FS and SA&D will go into a level of detail that can reasonably cover all objectives relevant to the EPD. It should encompass information that a feasibility study would normally call for with the principal aim to depict how the proposed systems can feasibly contribute to an environment such as EPD.

The team has the right skill set - Without the right skills dedicated to the team, the project will fail. The emphasis is on "dedicated to the team."

The right tools have been chosen - The first decisions to be made are the categories of tools: Document Management System, PKI, Database and administration, and so on. The tools must match the requirements of the organization, the users, and the project.

Users are properly trained - In spite of what the vendors tell you, users must be trained and the training should be geared to the level of user and the way they plan to use the system. All users must learn about the business data, and power users should have additional in-depth training on the data structures.

As a result, the use of a systematic approach (i.e. SSADM) to Skyver SA&D stage can revolutionise the technical and decision making processes resulting in reduced time lags in decision making, increased levels of confidence in decisions and more accurate decision. While the project had fixed deliverables, there was a need to maintain flexibility without sacrificing quality and provide well control of project management, system design and quality review.

Therefore, the resulting methodology was a hybrid of system investigation, system study, system analysis and system design from the SSADM, and the prototyping aspect of Web Application Development Methodology and sought to utilise on the implementation stage.

It is imperative that management continues to place emphasis on acquiring new technologies to facilitate continued restructuring, improve their business processes and meet the challenges that lie ahead. Managers should recognise the need of use computer tools to assist in making quick and accurate decisions and the importance of management support for the use of information technology. The success of any technology is dependent on the ability of the users to utilise the technology and user training is a key factor, as untrained users can result in unnecessary increases in cost.
Future Enhancements

Administration Role

The current prototype has only 1 type of user which is the Standard User. A future enhancement would be to include an Administrator user which should do the following functionality:

- Manage Users
- Access System Audit
- Access System Errors

Support for more mobile Phones

The current application was developed and testing using the mobile Phone Sony Ericsson K750i. A future enhancement will be for the system to support as much phones as possible.

User Maintenance from Application

A good future enhancement would be to allow User to edit his/her personal details from the application, not only from the Website.

Sending of SMS

Another Ideal feature would have been the possibility to send SMS from both the Application and the website.
# Chapter 8 – Bibliography

## References

- [http://www.scampers.org/steve/sms/libraries.htm](http://www.scampers.org/steve/sms/libraries.htm)
- [http://www.asp.net/](http://www.asp.net/)