MOBILE PHONE APPLICATION FOR ONLINE CONTACT SYNCHRONIZATION

Phone Contact Back Up Application
The purpose of the thesis was to design and implement Online Contact Sync which is a backup of phone or SIM contacts to the server in case of phone lost or SIM damaged. Contact phone numbers corresponding to their names are stored in the remote server.

The application was designed and implemented and it met all the requirement set for it. The application side was developed using Eclipse IDE which is used to write all the J2ME. The server side was written in PHP scripts and a robust database was developed with MySQL to store all the contacts online. The Logical and Physical design was carried out to bring up the effectiveness of the application. The application can run on all Android operating devices.

The application worked successfully when tested on all the Android devices though sometimes due to the large number of contacts and the limitation in the use of the school database the response time was very slow.
CONTENTS

ABSTRACT

1 INTRODUCTION ............................................................................................................. 3
   1.1 Statement of Problem ............................................................................................. 3
   1.2 Scope and Objectives of Study .............................................................................. 4
   1.3 Project Methodology ............................................................................................. 4
   1.4 Justification of the Study ...................................................................................... 5

2 RELEVANT TECHNOLOGIES .................................................................................... 6
   2.1 Android Application ............................................................................................. 6
   2.2 Installation of Eclipse and Android SDK ............................................................. 6
   2.3 Description, Requirements, and Constraints of the Project .......................... 6

3 APPLICATION DESCRIPTION .................................................................................. 8
   3.1 Project Design ...................................................................................................... 8

4 DATABASE AND GUI DESIGN .................................................................................. 11
   4.1 Design of Database ............................................................................................. 11
   4.2 Design of Database GUI ..................................................................................... 12

5 IMPLEMENTATION .................................................................................................... 14
   5.1 The Login GUI ..................................................................................................... 15
   5.2 The Register GUI ................................................................................................ 16
   5.3 The menu GUI ..................................................................................................... 16
   5.4 Creating the Android Project ............................................................................. 17

6 TESTING ...................................................................................................................... 21
   6.1 Software Testing .................................................................................................. 21
   6.2 Maintenance ......................................................................................................... 24

7 CONCLUSIONS AND SUMMARY ......................................................................... 26

REFERENCES ............................................................................................................... 27

APPENDICES
LIST OF FIGURES AND TABLES

Figure 1 Use Case Diagram for an Online Contact Syncing System p8
Figure 2 Entity Relational Diagram p11
Figure 3 User Database Screenshot p12
Figure 4 Contact Database Screenshot p13
Figure 5 Show the Splash Screen of the Application p15
Figure 6 Show the Login Screen of the Application p15
Figure 7 Show the Registration Screen of the Application p16
Figure 8 Show the Menu Screen of the Application p17
Figure 9 Shows the Creation of the Project Workspace p17
Figure 10 Starting the Android Project p18
Figure 11 Shows the Generated Folders p19
Figure 12 Shows the Java Classes in the Source Folder p19
Figure 13 Show the Starting of the Emulator p20
Figure 14 Show the Path of the Project p21
Figure 15 schronize.apk Location p22
Figure 16 Coping the schronize.apk file to the Google Drive p22
Figure 17 Backing up Diagram p23
Figure 18 Copying Contacts Back to the Phone from Server p24
1 INTRODUCTION

Online Contact Sync is the backup of the phone or SIM contacts to the server in case of phone is lost or the SIM damaged. Contact phone numbers corresponding to their names are stored in the remote server. It can be updated when there is a new contact.

Applications are developed to interface between phones or SIM and the remote database where all contacts information are stored. Remote databases are designed to store individual contacts such as name, number, email, picture, address, fax etc. Accessing and retrieving individual contacts is based on authentication by identifying the user username and password for security reason.

A Client application can be developed using J2ME, Android, Flex, HTML5, Sencha etc. depending on the phone operating system, memory capacity and vendor. A Server application can be developed using PHP, ASP.NET, Python, JSP etc. to communicate with the remote database parsing queries using either JSON or XML technologies.

The online contact backup can be only supported by either cellular network or by WIFI on an internet ready mobile phone.

1.1 Statement of Problem

In the events of a change of a mobile handset or SIM, Transferring contacts can be difficult. The problem becomes more significant when you have too many contacts, more than that you can store in your SIM memory. Otherwise you can transfer your contacts to your SIM and change the handset.

In the event of a loss, it becomes completely impossible to retrieve such contacts without manually sourcing for the contacts information elsewhere. In both situations an application that allows online contacts synchronizing comes very handy.
1.2 **Scope and Objectives of Study**

The objective of the thesis was to design and implement an Online Contact Sync system that uses Java mobile application connected to a remote server. It handles the backing up and retrieving phone contacts of an individual account.

The specific objectives were:

i) To examine and analyze the various steps required for an online contact syncing.

ii) Design a database for the online contact syncing system.

iii) Design a mobile based application for the online contact syncing system.

iv) Implement the server side applet for the online contact syncing system.

v) Implement a mobile based application for the online contact syncing system.

1.3 **Project Methodology**

The project methodologies are listed as follows:

1) A thorough analysis of an existing online contact syncing system was carried out by finding out different methods or approaches that are used for their contact synchronizing system and how their data stored were verified.

2) A comprehensive relational database approach was adopted in the designing of the system.
3) A well structured class diagram which maps out the analysis and designed of mobile application was carried out.

4) Implementation of the server side applet was carried out using PHP as a server side script. MySQL was adopted for the database.

5) The mobile application was implemented using J2ME platform

1.4 Justification of the Study

This project will make a contribution to the society by eradicating the problem of lost phone contacts as a result of a phone loss or SIM damage.

As a result of the project it is now possible synchronize all the contacts on the mobile phones and back it up on the remote server. This makes it readily available for the user retrieve the contacts anywhere in the world since all the contacts are saved on the server.
2 RELEVANT TECHNOLOGIES

2.1 Android Application

Android has the potential of removing the barriers to success in the development and sale of a new generation of mobile phone application software /2/. Android gives developers a way to develop unique, creative applications and get those applications in the hands of customers. For us as mobile application developers, that means we are free to develop whatever creative mobile applications we can think of, free to market (or give them out as our option) to Android mobile phone owners, and free to profit from that effort any way we can /2/.

2.2 Installation of Eclipse and Android SDK

To start the project, we had to install Eclipse following the steps highlighted as follows;


2. Install Android SDK from http://developer.android.com/sdk/installing/index.html to extract the required Google API library to fix some bugs /1/.

3. Install the Eclipse ADT (Android Development Tools) plug-in by following the steps in the following link http://developer.android.com/sdk/installing/installing-adt.html /1/

Note: Make sure the latest version of ADT is installed.

Once the Eclipse is ready we could start to create an Android project and this will be discussed in full details in the implementation stage of the project.

2.3 Description, Requirements, and Constraints of the Project

The purpose of the task was to develop an Android application that will synchronise contact information on a phone, copy the contact and back it up to the remote
server. In order to be able to have access to the application, a user needs to first register their information on the register page and login to the access application. All the registered user information is stored in a database for easy access. The software requirement for the project includes: Android Development Environment for Android platform, Android devices with contacts and a SQL server. The constraints of this project came from the limitation from the school remote server.
3 APPLICATION DESCRIPTION

3.1 Project Design

The analysis phase of the online contact syncing system began with the sniffing the internet. Searching online added an insight on what an online contact syncing system looks like. Use cases were used during the analysis phase of a project to identify and partition the system functionality /4/. It also describes everything that can go wrong during the courses of the specified behavior and what remedial action the system will take. The use case is the part of the information gathered. In the analysis, the system was analyzed to have one user. This leads to a UML Use case diagram of the user. See Figure 1 for details.

![UML Use Case Diagram for an Online Contact Syncing System](image)

**Figure 1.** Use Case Diagram for an Online Contact Syncing System

The models in the diagram can be broken down into log in, register new account, login for authentication, backing up contact to the server and retrieving contact back to the phone at ease.
The user of the application is the actor in the above model and the function of each model is briefly explained below:

**Log in:** This is the pathway through which authorized users have access to the online contact synchronizing system. How the use case called log in works is explained below.

- The user enters a unique name and password via the keypad.
- The system validates the identifier.
- The system verifies what functionality is available to this user.

The actor in this use case is the user of the application. Conditions below show what happens when an invalid user name and password are entered:

- If the username and password are invalid, an error message is displayed.
- If the application engine fails, an error message is displayed.
- If the database server fails, an error message is displayed and no additional functionality is provided.

**Register new account:** In this use case an account is registered and the user of the application is the actor in this use case. It is explained below how the use case called register new account works:

- The user uses the keypad to input data on account information.
- After entering the necessary information, it moves to the menu page.

Other alternatives that may occur:

- If communication with the application engine is lost, an error message is shown on the display.
**Back up Contacts:** The actor in this use case is the user of the application.

This is how the use case works:

- The user clicks on back up to the server.
- There is a display of contact names that is processing.

**Retrieving Contacts:** The actor in this use case is the user of the application and it works as follows:

- The user clicks on retrieving contacts from the server.
- There is a display of contact names that is processing.
4 DATABASE AND GUI DESIGN

4.1 Design of Database

A database is a structured collection of records or data that is stored in a computer system. The structure is achieved by organizing the data according to a database model. The model in most common use today is the relational model. Other models such as the hierarchical model and the network model use a more explicit representation of relationships /4/.

One of the goals of this design phase was the ability to use special techniques to design a powerful database that will hold large number of data and can keep track of customer’s contacts. Data are saved inside the specific tables created to hold specific data on the remote school server (www.mysql.cc.puv.fi). The server side was written using PHP. The PHP will be used to parse the data to JSON format in order to use it for the server operations /3/. The database was created using MySQL.

The entity-relationship model (ERM) is an abstract and conceptual representation of data and it is used to describe the logical design of the application /4/. It also depicts the relationship of the data in the database.

The entity relational diagram for this project is shown in Figure 2.
Figure 2 explains the contents of both the User table and Contact table inside the database. There is one-to-many relationship existing between the tables. E.g. a user can have many contacts (1: M).

The table user contains entities such as id, name, username, phone number and password which form the columns inside the user table in the database. These entities form part of the register page so that when the new user registers to the application, the username and password will be saved. The id which will be generated automatically when the new user registered is unique for each user. This is because one table is being used to save all the different contacts and users have to be differentiated by their IDs as shown in Figure 3.

The contact table contains the full name, phone numbers and the unique ID assigned to each user as shown in Figure 4.

### 4.2 Design of Database GUI

For the application database, two tables were created which are the user table and contact table. The User table holds the username, password, full name and the phone number which the user supplies when registering to have access to the application and all these parameters are saved in the user table in the database shown in Figure 3.

![Figure 3. User Database Screenshot](image)
All these parameters are necessary for logging in into the application.

The contact table inside the database holds the names and phone number of all the contacts on the device so once the user registers for the application and logs in, the user can then have access to the menu of the application. On the menu list there is the back up of contacts and. Figure 4 shows the list of contacts backed up on the server with a unique user id 5 which is only for this user. Each user of the application has a unique user id and this is very important because only one table is used for all the users of this application.

![Contact Database Screenshot](image.png)

**Figure 4.** Contact Database Screenshot
5 IMPLEMENTATION

This chapter describes the system design and implementation of the Online Contact Synchronizing System that uses the Java mobile application. It will also show how the system and sub modules are implemented and the Graphical User Interface (GUI).

A system is a combination of related parts organized into a complex whole, while a system design is the grouping of the system to influence the system’s practical usefulness and a system design is a blueprint that shows how a process is to be carried out or how combination of parts should work together. The system design comprises the following:

1. Logical design
2. Physical design
3. Implementation

This section points out on the physical characteristics of the system such as the graphical user interface (GUI) which was designed using J2ME. Each interface is user-friendly and designed to accept data as inputs. Several interfaces are contained in the Online Contact Syncing System, some of which are the splash-screen GUI, login GUI, register GUI, and menu GUI.

The splash-screen GUI is the first display the user sees when he/she launches the online contact syncing system. It displays an icon of an external drive and CD-ROM. It lasts for few second and moves to the next display which is the login GUI. It is shown in Figure 5.
Figure 5. Show the Splash Screen of the Application

5.1 The Login GUI

Figure 6 shows the login GUI which is the second display the user sees after the splash screen. It contains text fields for the user to supply his/her username and password for authentication.

Figure 6. Show the Login Screen of the Application
5.2 The Register GUI

This register interface shown in Figure 7 allows the user to set up a new account if there is no existing account before. It contains text fields such as names, phone number, username, password etc. The fields are not null. The user must provide all the necessary information before he/she can be registered. The registered information is saved inside the database.

![Figure 7. Show the Registration Screen of the Application](image)

5.3 The menu GUI

After the login in or registering a new account the menu page was displayed to the user. The powerful display is the menu GUI which has two options: back up to the server and retrieving from the server.
Figure 8 shows the menu interface of the application which allows the user to select the task to be carried out either backing up to the server or copying contacts to the phone. This is the important aspect of the project.

Figure 8. Show the Menu Screen of the Application

5.4 Creating the Android Project

Figure 9 shows the idea to start a new Android project for the backing up and synchronizing an online contact and creating a workspace folder inside your computer local drive. The first step is to download the Eclipse Android Development Tools (ADT) folder inside the folder for the project in the workspace.

Figure 9. Shows the Creation of the Project Workspace
A new platform named back-up was created when the Eclipse was started. When the project was created inside the Eclipse workspace, other necessary folders needed for the execution of the project, such as the Android Manifest, Drawables, Layout, Values, Bin and some other folders were created automatically.

Figure 10. Starting the Android Project

Figure 10 shows how to name an Android project and setting the minimum required and target SDK as required for the application. This determined the version of the Android device on which the application can work and run successfully.

Figure 11 contains resources folder which holds the drawable, layout and values folders. The drawable folder holds the icons used for the application which can be in jpg or png image. The layout folder holds the project XML files which are the GUI of the application. The value folders contain the XML files of the colors, strings, dimensions and styles for setting some parameter. The project required an Android support library which was saved in the library folder. The source folder will hold project Java classes inside a package for the project.
Figure 11. Shows the Generated Folders

The source folder has the package which holds all the Java classes created for the application. This is as shown in Figure 12.

Figure 12. Shows the Java Classes in the Source Folder

There is an Emulator icon on the Eclipse tool bar. Clicking on the icon will start it. After that, set the Android Virtual Device, the Android target and the target virtual device are set, as shown in Figure 13. The application is then run on the emulator. This is just for testing the application.
Figure 13. Show the Starting of the Emulator
6 TESTING

6.1 Software Testing

After running the program successfully, it generated an apk (Android application package) file which can be found in the path of the project and this was achieved by right clicking on the project and checking the properties in order to see the location of the project as shown in Figure 14.

![Properties for schronize](image)

**Figure 14.** Show the Path of the Project

After the path had been located, the schronize folder was clicked. The Apk file which is used for the installation was copied from the already existing bin folder inside the schronize folder to the Google drive. The Figure 15 shows the path of the apk file which is our main concern in this project design.
In order to launch the apk file on the phone, the file was downloaded to the Google drive on the laptop and also on the phone to make it easy to install. This can be accessed on the phone.

On the Google drive the user has to create a folder (name it anything depending on the user).

Figure 16 diagram shows how to create a folder inside the Google drive since it is not possible to just drop the apk file inside the Google drive. Once the apk has been successively saved on the drive, it can then be accessed on the already downloaded Google drive on an Android device.

Figure 15. schronize.apk Location

Figure 16. Coping the schronize.apk file to the Google Drive
Once it is downloaded on the phone and installed, the user needs to fill all the fields on the Register page as shown in Figure 7.

Once the user has registered they can then login in order to be able to use the application and this is achieved by going to the log in page as shown in Figure 6.

After the login the user can now have access to the application menu as shown in Figure 9.

The user can then back up their contact to the server and each time a new contact is added on the phone it is required that the user logs in with their username and password to back up the new contact and this is repeated each time a new contact is added on the phone.

The main purpose of all these is that in case of a damaged phone or total loss of the phone the user can log in with their user account and copy the contact to the new phone.

When the user backs up to the server he/she will see how the contacts are loading to the remote server and once all the contacts have been successfully copied to the database, it displays a message to the user ‘thank you for backing up all your contacts in our server’ and the user can click OK to complete the action as shown in Figure 17.

Figure 17. Backing up Diagram
Figure 18 shows the main idea about the project work and this process can be carried out when the user of the application loses his mobile phone or in case of a damaged phone or SIM card. All that is just required for the user is to re-install the application on the new phone and once that is achieved successfully, the user can then log in with their username and password. If that is successful, that will take the user to the main menu of the application and all the user needs do is to click on ‘copy contact to phone’ and once all the contacts are copied to the phone, a message is displayed to the user, as shown in Figure 18 and once the user clicks OK, the action is complete.

![Figure 18: Copying Contacts Back to the Phone from Server](image)

6.2 Maintenance

It is necessary to review the new system after it has become functional in order to know if it has achieved the requirements. Four types of maintenance practice will
be performed on the proposed system and they are corrective, adaptive, perfective and preventive.

The corrective maintenance will be done immediately after the implementation phase. This will be done so that a mistake can be corrected immediately and the modification of computer will take place so that the software runs properly.

The adaptive maintenance includes adding new features that were not necessary in the development. This will be done when an additional feature is needed in the system.

The perfective maintenance takes place in order to modify the software to become more efficient. It can also include hardware upgrades to increase the speed of the system.

The preventive maintenance is acting with the intent of preventing future problems. Data backup and disaster recovery are considered preventive. All these maintenance practices will be looked into one after another.
7 CONCLUSIONS AND SUMMARY

The application was developed and implemented on the Android platform which will run on all Android operating system devices and it met all the required specification. This application is very important just in case something happens to the phone or SIM card. If the phone is lost, this is how it is possible to get all the contacts back.

For future project, one or two things can be added to the application, such as email, contacts, pictures, house address and postal code. This will help to get the full detailed information about the contacts.
REFERENCES


