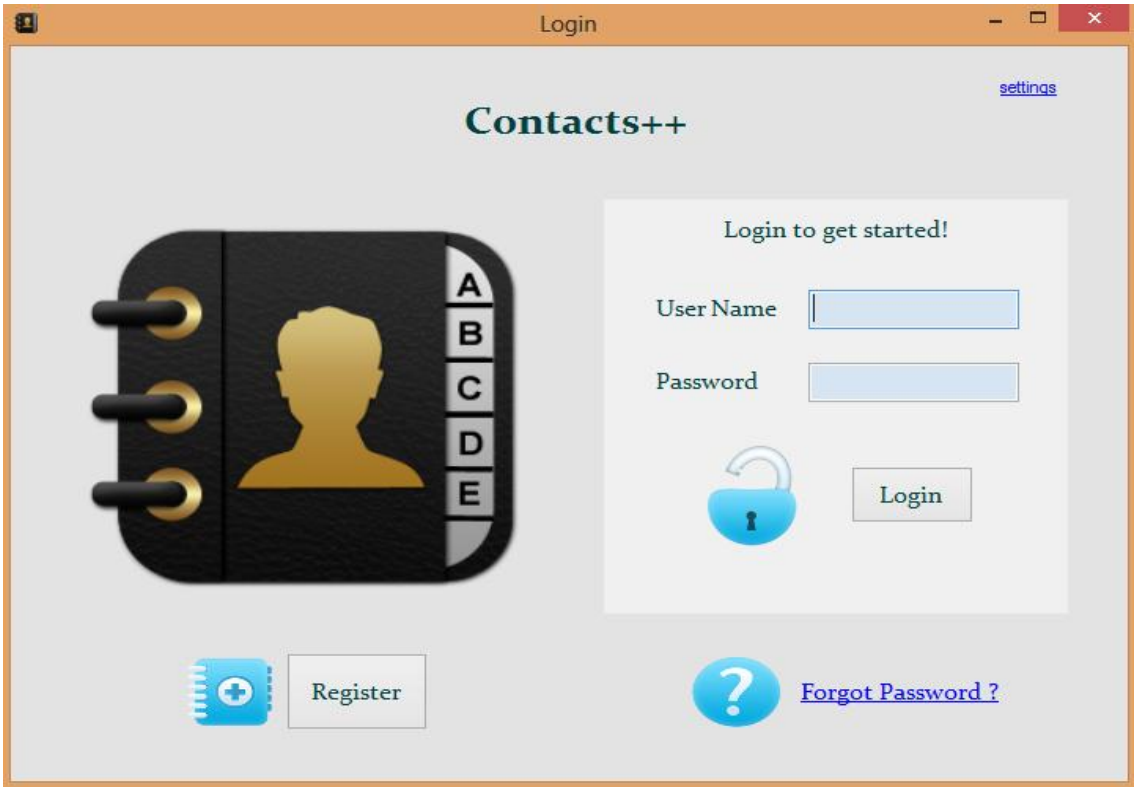
CONTACT MANAGEMENT SYSTEM

Group Number 7

Introduction:

- Aditya Kumar Jha -11010104
- Bhanuteja Gullapalli -11010112
- Prashanth Verma -11010129
- Harshith Reddy -11010149
- Rahul R Huilgol -11010156
- Rohit Kamra -11010159



Introduction:

The **Contact Management software** is an easy to use software program that enables users to easily store, edit, find and manage information of their contacts and reminders, offline on a computer.

The software allows the user to maintain an account which provides him with various options to manage his contacts.

Specifications of the Software:

- *Application Type*: .exe
- *Supported Operating System*: Microsoft Windows
- *Programming Language Used*: Visual C++
- *IDE* : Visual studio 2010
- *Database Used*: MySql
- *Format of Import/Export Files:* .csv
- *Input/ Output Folders:* The files to be imported can be chosen from any location. The exported csv files can also be saved to any folder. The log files are stored in a folder Log, which lies in the same folder as the executable.

Code and Database specifications: The application is developed in Visual Studio 2010 as a 'Windows Form Project' using Visual C++ and has forms for login, register, password recovery, user's home, adding contacts, viewing contact specific info, recovering contacts, adding reminders and viewing user's notifications.

The application features a Graphical User Interface featuring labels, images, list view, buttons, text boxes and backgrounds. This was developed and customized in the 'Design' mode of the project.

The software uses four tables of a MySQL database, one for maintaining the login credentials, one for storing the basic contact details, one for storing extra contact fields and the last one for storing reminders specific to a contact.

The connection to the database from the Forms is established by adding a reference of MySQL.Data to the project.

The functionalities like adding, editing, deleting, restoring contacts are implemented by suitably using MySQL queries and C++ functions.

For importing and exporting contacts, we use the standard .csv file format, with the first line of the file as usual indicating the fields of the contact.

For importing it is mandatory for the first two fields to indicate the name and mobile no. of the contact respectively, because these are the basic details of a contact in the contacts table of the database.

Installing the software:

The software can be installed by running the installer, which prompts the user to enter the login details of his/her MySQL server on localhost. Using this information, we generate the SQL database and tables for storing the details.

A total of 2 executables are installed in the system of which one is the main software (ContactsPlusPlus.exe) and the other is the database installer i.e. which installs the raw database structure needed to store all software specific data (dbinstall.exe). The user needs to add his preference of details for the database connection settings, while installing the database. The program is installed in the C:/Program Files/ContactsPlusPlus folder by default and in this folder there is a folder named Log for the log files. After the setup, the shortcuts for both executables are created on the desktop.

Features and Modules:

- **SQL SETTINGS:** If the user wants to the change the username, password of the SQL server, then he/she can do so using the 'MySQL Settings' form, which can be accessed from the login form.

- **LOGIN**: The user can view and modify his contacts, only after he logs into his account with his username and password. When the user clicks the login button, his corresponding details are searched for in the 'login' table and if the fields match, then he is given access to his contacts, i.e the home window opens, else an error message pops up.

- **REGISTER**: For using the software each user first needs to register with a username and password of his/her choice. These fields are stored in the login table.

- **PASSWORD RESET**: The user is provided with an option of resetting his password in case he forgets it, by answering his security question. The security question and its answer are taken from the user at the time of registration and are stored in the login table.

- **HOME WINDOW:** If the user login is successful, then his home window loads in which contains all the essential labels and buttons. The list view of the window contains all the contacts with the basic info name and mobile no. For more info specific to the contact, the user needs to double click on the contact in the list-view. The home window also contains buttons to view notifications and contact modifying, searching options along with delete and restore. The contacts are shown in an ascending alphabetical order.

- **SEARCHING CONTACTS**: The user can search his contacts using various fields like name, mobile, email, address, etc. and the results of the contact like name and mobile are showed in the list-view, while other details specific to the contact can be viewed by clicking on the contact in the list view. The search is incremental and all the values containing the query as a substring are shown as results. This has been done using a wildcard and the keyword 'like' in SQL queries.

- **ADD CONTACTS**: After the user logs in, he is presented with his home form, containing the list of his contacts and various buttons/features. The add contact, is one such feature in the home form.
  When the user clicks the add contact button, a new form opens up prompting the user to enter details of the contact.
  Among the details entered by the user, the basic details like his username, contact name and mobile are stored in the 'contacts' table whereas the other extra fields are stored in the 'contacts1' table. Even more fields are available in the edit contact form.
  Note that each contact is identified by the 'sno' specific to that particular contact.

- **DELETE CONTACT:** The user can delete a specific contact, by selecting a specific contact from the list view and then clicking on the delete contact button. During deletion the flag specific to that entry is made 0.
  Note that by default, when you add or import contacts this field has value 1

- **RESTORE CONTACTS**: The user is given this option of restoring his contacts, which allows him to get back his deleted contacts. While restoring the contacts, the flag is again reset to 1.

- **ADDING EXTRA FIELDS**: The user can add extra fields specific to a contact in a separate window (contact-info window) which pops up when he 'double-clicks' on the contact in the list-view. All the extra fields, he adds are stored in the 'contacts1' table in the format 'sno--userid—fieldname—value—flag'. The flag is again by default set to 1.

- **DELETING FIELDS**: Apart from the option of deleting the entire contact, the user is provided with options to delete fields specific to a contact. In this the flag of the specific contact is set to 0 in the 'contacts1' table.

- **RESTORING FIELDS**: The user can restore the deleted fields of a contact by double-clicking on the required field, from the list of deleted fields in the restore-field window. While restoring the fields, the corresponding flag is reset to 1.

- **EDIT CONTACT FIELDS:** The user can edit the values of the fields specific to each of the contacts in the contact-info window. After editing the field the user needs to click on the 'Done' button adjacent to the particular field, so that the fields are modified in the database using the 'update' query. The user can edit any field present in the form listing all contact details.

- **IMPORTING CONTACTS:** The user can import contacts from a standard .csv file into his existing list. The file to be imported must have its first two fields corresponding to name and mobile no. of the contact. During importing, the first two fields of name and mobile no. are imported into the first table 'contacts' while the remaining extra fields are added into the 'contacts1' table.

- **EXPORTING CONTACTS**: The user can export all his existing contacts into a .csv file. The user can select the folder where the CSV file needs to be generated.
  The first line of the exported file contains the format of the fields. This first line contains all the unique fields amongst all the contacts of the user in the contacts1 and contacts table. During exporting the fields like name, mobile no. are added from the 'contacts' table, while the remaining fields are added from the 'contacts1' table. Then the values of the contacts are added by suitably checking for their corresponding field position in the first line.

- **SET REMINDERS**: The user can set reminders specific to a contact, so that he can be reminded about the event with a pop up, if he is logged in at that time. These reminders are stored in a separate table 'reminders', each row of which contains the sno, date, time, text. The reminder is implemented by using a timer for the user's home form. The timer compares the user's System Time with the values in the database every one minute and pops out the reminder message if a match is found.

- **VIEW NOTIFICATIONS**: The user is provided with options to view his notifications, sorted by date and time. The notifications window consists of two list views – one showing the present day's events and the other showing all the notifications.

- **ADDING USER PHOTO:** The user can upload images for each contact, which can be viewed in the contact info window, specific to the contact. When the user uploads a photo of his contact, a copy of the image is made and stored in the software's directory.

- **LOGFILE:** The log-file of each user stores the information about his activity in a table Log. As soon as the user 'logs in', 'adds contact', 'logs out',etc. his corresponding log file gets updated with his current activity. The format of the log file is time-date-activity. Note that each user has a log file corresponding to his name in the format "logfile_userid".

- **LOGOUT:** After the user is done with his activity, he can logout of his session using the Logout – (link label) in his home window. After clicking on the logout label, the user's home window is hidden and the login window is displayed.

Format of Database Tables:

1. Login Table

The table has four fields

- **userid**- This is an id unique to each user. This field has auto-increment condition, which makes sure each registered user gets a unique id.
- **username** - This column contains the usernames of all the registered users.
- **password –** this column stores the passwords of the user with the corresponding username in the username column
- **security question**- this column contains the security question which can be invoked for password resetting.
- **Security answer**- the answer to the security question

	userid	username	password	question	answer
▶	1	rohit	rohit	rohit	rohit
	2	bhanu	bhanu	Hi?	Hi
*	NULL	NULL	NULL	NULL	NULL

2. Contacts Table

- **sno**- This is a field unique to each contact. This field is set to auto-increment so that each contact gets a unique id.
- **userid**- this field holds the user-id to which the contact belongs to.
- **name** - this field holds the username of the contact.
- **mobile no**.- this column holds the mobile number specific to the contact.
- **Flag1** – this lets us identify if the contact is deleted or not. 1 is default value which means it is not deleted

sno	name	mobile	userid	flag1
1	harshith	8823822555	1	BLOB
2	Rohit	983223243	1	BLOB
3	Bhanu	3242342343	1	BLOB
4	Rahul	8828211123	1	BLOB
5	Aditya	8011232442	1	BLOB
NULL	NULL	NULL	NULL	NULL

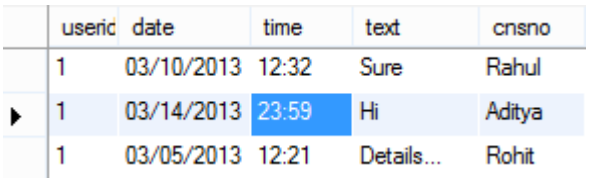
3. Contacts1 table:

- **sno**- This is a field unique to each contact common to both 'contact' and 'contact1' tables. This helps us link each row to a contact in contacts table
- **fieldname**- this column stores the fieldname of a contact detail
- **value** – the value of the corresponding field
- **Flag2** - this lets us identify if the contact is deleted or not. 1 is default value which means it is not deleted

sno	userid	fieldname	value	flag2
1	1	Email Id	harshit@gamai.com	BLOB
1	1	Telephone	020 - 1888623	BLOB
1	1	Address	Dispur	BLOB
2	1	Email Id	rohit@yahoo.com	BLOB
2	1	Telephone	030 - 2341234	BLOB
2	1	Address	Guwahati	BLOB
3	1	Email Id	bhanu@gmail.com	BLOB
3	1	Telephone	343 - 324324324	BLOB
3	1	Address	Guwahati	BLOB
4	1	Email Id	rahul@gmail.com	BLOB
4	1	Telephone	020 - 312433223	BLOB
4	1	Address	Delhi	BLOB
5	1	Email Id	jha@ya.com	BLOB
5	1	Telephone	040 - 3132133	BLOB
5	1	Address	Patna	BLOB

4. **Reminder table:**

- **Date** – The date of the reminder set
- **time** – Time set for the reminder
- **text** – Description of the event
- **contact** - this lets us identify which contact the reminder is from



	useric	date	time	text	cnsno
	1	03/10/2013	12:32	Sure	Rahul
▶	1	03/14/2013	23:59	Hi	Aditya
	1	03/05/2013	12:21	Details...	Rohit

Challenges we faced and successfully resolved during the course of the project:

- Initially we were not able to pass variables from one form to another - added a header file "Global.h" that consisted of extern variables and included the header file in various forms.
- We were not able to hide the login form after the user logged in - hid the parent form when child was shown and checked for a value thrown by the child form every second and once the desired value was thrown re-displayed the child form
- We were thinking of creating a backup table for deleted contacts but this would cause wastage of memory instead of that we kept a flag variable to check for the same.
- We were unable to incorporate images directly into the database so stored them in a specific folder
- We were unable to set the username, password and port for the database dynamically - it was sorted by prompting the user for these details at the time of installation and provided him with the option of resetting it by saving these details in a text file

Limitations:

- The password of the user is directly stored in the login table, without any md5 hashing or encoding.
- The contacts can be imported/exported only in standard .csv format.
- The software doesn't allow importing/exporting of the images of the contact.
- The reminders can be set only if the System time is in the following format MM/DD/YYYY.
- The software doesn't provide automatic syncing feature, instead the user has to manually export his contacts.
- The user can't use characters like ',' and ';' in the security question as they are used in SQL injection.
- If the user clicks on "Delete'' contact button without any contacts in the table/list view he will run into an unhandled exception.