**MINI PROJECT**

**TITLE : RAPPEL**

**Group Members :-**

**Sumedha Belur (304)**

**Siddhi Hinge (318)**

**Revati Jagtap (321)**

**A MINI PROJECT REPORT ON**

**RAPPEL**

SUBMITTED TO SAVITRIBAI PHULE PUNE UNIVERSITY

IN PARTIAL FULFILLMENT OF THE DEGREE

**COMPUTER ENGINEERING**

**(THIRD YEAR)**

**BY**

**SUMEDHA BELUR (304)**

**SIDDHI HINGE (318)**

**REVATI JAGTAP (321)**

**UNDER THE GUIDANCE OF**

**Prof. N.W.Dangare**

****

**DEPARTMENT OF COMPUTER ENGINEERING**

**(THIRD YEAR)**

Shalaka Foundation’s

**KEYSTONE SCHOOL OF ENGINEERING**

Near Handewadi Chowk ,Urali Devachi,

Shewalewadi,Pune,Maharashtra-412308

****

**DEPARTMENT OF COMPUTER ENGINEERING**

Shalaka Foundation’s

**KEYSTONE SCHOOL OF ENGINEERING**

Near Handewadi Chowk ,Urali Devachi,

Shewalewadi,Pune,Maharashtra-412308

**CERTIFICATE**

This is to certify that the mini project report entitled

“**RAPPEL**”

Submitted by

**Sumedha Belur (304)**

**Siddhi Hinge (318)**

**Revati Jagtap (321)**

is a bonafide work carried out by them under the supervision of **Prof.N.W.Dangare** and it is submitted towards the partial fulfillment of the requirement of Savitribai Phule Pune University for the award of Third Year Computer Engineering.

**Project Guide External Examiner**

###### Head

###### Dept. of Computer Engineering

**Place:Hadapsar**

**Date :22/10/2018**

**Certificate by Guide**

This is to certify that **Group Id**-**1** have completed the mini project work under my guidance and supervision and that, I have verified the work for its originality in documentation, problem statement, implementation and results presented in the mini project work. Any reproduction of other necessary work is with the prior permission and has given due ownership and included in the references.

**Place: Hadapsar**

**Date :** 22/10/18

Signature of Guide

(Name of guide)

**ACKNOWLEDGEMENT**

The success and final outcome of this project required a lot of guidance and assistance from many people and We are extremely privileged to have got this all along the completion of our project. All that We have done is only due to such supervision and assistance and we would not forget to thank them.

It is our privilege to express our sincerest regards to our project coordinator Prof .N.W.Dangare for their valuable inputs, able guidance, encouragement, whole-hearted cooperation and constructive criticism throughout the duration of our project.

We deeply express our sincere thanks to our Head of our Computer Department for encouraging and allowing us to present the project on the topic “RAPPEL“at our department premises for the partial fulfillment of the requirements leading to the award.

We take this opportunity to thank all our lecturers who have directly or indirectly helped our project. We pay our respects and love to our parents and all other family members and friends for their love and encouragement through out our career. Last but not the least we express our thanks to our friends for their cooperation and support.

Our thanks and appreciations also go to my colleague in developing the project and people who have willingly helped us out with their abilities.

**ABSTRACT**

Every day, we have many chores to do, doing housework, business work, meetings, birthday parties, phone calls, billings, TV programs and so on. it is not easy to memorize all and handle all well, in order not to forget or miss any important things, we can ask help from some remainder application and it does be workable and helpful

A Reminder System is important for busy people because we do not have to remember all our works. And writing them down somewhere is not a very good idea because then we have to keep track of that piece of paper.

The Reminder System lets you “set and forget‟ and then get reminded when the time comes. And so it becomes a perfect tool for business or personal follow up.

The purpose of this paper is to learn the most important attributes needed to remind upcoming Birthdays, Anniversary,Events, Important Meetings, etc.

RAPPEL is an application which helps the user about the registered event reminders . It is an application which gives the remainder of saved event before the event via email system. Due to the email system user get an idea about event before event actually occur anywhere at any time on user’s mobile.

As user come to know about events early user can plan the event according to his/her convenience.

INDEX

|  |  |  |
| --- | --- | --- |
| **Sr No**. | **Chapter Name** | **Page No.** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**CHAPTER 1 : INTRODUCTION**

Now a days, mobile phones are used everywhere and every time in our daily life ranging from communication, entertainment to health and wellness applications. Today we know very well that the use and the demand of Mobile Application Development are increasing and it makes inspiration to create user friendly and effective variety of application.

The various social networking sites provide a feature to remind the birthday / anniversary of the loved one, friends and relatives upcoming events and greet them by posting messages on the wall. But what happens when users net is not working on that day or net is too slow in speed or there are some people who are unaware if internet and latest social networking sites.?

And when he suddenly remembers that time has passed there is no option for this problem. Once the time is passed, the person has to do work or wish their dearest person after the time is pass away.

And may be the person may be suffering due to this problem. So application relieves us from all these problems and provides a very user-friendly environment.

Our proposal was to make yet another reminder application which would list all upcoming events like Birthday,Anniversary, Best Wishes, Festivals, Important Meetings, Tasks of the day etc on a Android Smartphone, which would give Greeting notiﬁcations of friends and relatives on registered date and time. The application will show lists of all contacts with birthdays, anniversary, meetings etc so user does not have to search anymore.

In this project the user only needs to add the some information about the birthday person or events and enter the date of birthday , and when the birthday occurs the remainder will be given. This reminder concept is more simple to execute but also very needful in our day to day life . Every person in this busy world needs this remainder .

This remainder will remind user about the event before the event occur via email system. So that user can plan the event according to his or her convenience.

Rappel has an inbuilt support for sending emails when event occur. This application ensures that you will never miss an important date , event, anniversary or birthday of loved once.

The best part about this application is it is very easy to handle and and user can set remainder at any time anywhere with less steps and with less complexity.

**CHAPTER 2 : PACKAGES**

For this project we have been using Python3 , it provides various packages . Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently whereas the other languages use punctuations. It has fewer syntactical constructions than other languages.

The packages used in this project are as follows :-

1. easygui

2. csv

3. datetime

4. smtplib

1. **EASYGUI** :-

EasyGui provides an easy-to-use interface for simple GUI interaction  
with a user. It does not require the programmer to know anything about  
tkinter, frames, widgets, callbacks or lambda. All GUI interactions are  
invoked by simple function calls that return results.

## Default arguments for EasyGui functions :-

Most arguments to EasyGui functions have defaults. Almost all of the boxes display a message and a title. The title defaults to the empty string, and the message usually has a simple default.

Eg msgbox("Danger, Will Robinson!")

### Different button boxes of easygui:

### msgbox

msgbox displays a message and offers an OK button.

Eg def msgbox(msg="(Your message goeshere)",title="",ok\_button="OK"):

### ynbox

ynbox offers a choice of Yes and No, and returns either True of False.

### buttonbox

To specify your own set of buttons in a buttonbox, use the buttonbox() function.The buttonbox can be used to display a set of buttons of your choice. When the user clicks on a button, buttonbox() returns the text of the choice. If the user cancels or closes the buttonbox, the default choice (the first choice) is returned.buttonbox displays a message, a title, and a set of buttons. Returns the text of the button that the user selected.

### enterbox

enterbox is a simple way of getting a string from the user.

EasyGui provides a few basic functions for allowing a user to navigate through the file system and choose a directory or a file.

### diropenbox

diropenbox returns the name of a directory.

### fileopenbox

fileopenbox returns the name of a file.

1. **CSV :-**

(Comma Separated Values) format is the most common import and export format for spreadsheets and databases.These differences can make it annoying to process CSV files from multiple sources. Still, while the delimiters and quoting characters vary, the overall format is similar enough that it is possible to write a single module which can efficiently manipulate such data, hiding the details of reading and writing the data from the programmer.The csv module implements classes to read and write tabular data in CSV format. Programmers can also describe the CSV formats understood by other applications or define their own special-purpose CSV formats.

The [csv](https://docs.python.org/3/library/csv.html#module-csv) module defines the following functions:

1. csv.**reader**(*csvfile*, *dialect='excel'*, *\*\*fmtparams*)

2. csv.**writer**(csvfile, dialect='excel', \*\*fmtparams) etc

1. **DATETIME :-**

The [**datetime**](https://docs.python.org/2/library/datetime.html#module-datetime) module supplies classes for manipulating dates and times in both simple and complex ways. While date and time arithmetic is supported, the focus of the implementation is on efficient attribute extraction for output formatting and manipulation. For related functionality, see also the [**time**](https://docs.python.org/2/library/time.html#module-time) and [**calendar**](https://docs.python.org/2/library/calendar.html#module-calendar) modules.

Available types :

class datetime.**date**

An idealized naive date, assuming the current Gregorian calendar always was, and always will be, in effect. Attributes: [**year**](https://docs.python.org/2/library/datetime.html#datetime.date.year), [**month**](https://docs.python.org/2/library/datetime.html#datetime.date.month), and [**day**](https://docs.python.org/2/library/datetime.html#datetime.date.day).

class datetime.**time**

An idealized time, independent of any particular day, assuming that every day has exactly 24\*60\*60 seconds (there is no notion of “leap seconds” here). Attributes: [**hour**](https://docs.python.org/2/library/datetime.html#datetime.time.hour), [**minute**](https://docs.python.org/2/library/datetime.html#datetime.time.minute), [**second**](https://docs.python.org/2/library/datetime.html#datetime.time.second), [**microsecond**](https://docs.python.org/2/library/datetime.html#datetime.time.microsecond), and [**tzinfo**](https://docs.python.org/2/library/datetime.html#datetime.time.tzinfo).

class datetime.**timedelta**

A duration expressing the difference between two [**date**](https://docs.python.org/2/library/datetime.html#datetime.date), [**time**](https://docs.python.org/2/library/datetime.html#datetime.time), or [**datetime**](https://docs.python.org/2/library/datetime.html#datetime.datetime) instances to microsecond resolution. etc

1. **SMTPLIB :-**

Simple Mail Transfer Protocol (SMTP) is a protocol, which handles sending e-mail and routing e-mail between mail servers.Python provides **smtplib** module, which defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon.Here is a simple syntax to create one SMTP object, which can later be used to send an e-mail –

import smtplib

smtpobj = smtplib.SMTP([host,[port,[local\_host]]])

Here is the detail of the parameters −

* **host** − This is the host running your SMTP server. You can specify IP address of the host or a domain name like tutorialspoint.com. This is optional argument.
* **port** − If you are providing *host* argument, then you need to specify a port, where SMTP server is listening. Usually this port would be 25.
* **local\_hostname** − If your SMTP server is running on your local machine, then you can specify just *localhost* as of this option.

An SMTP object has an instance method called **sendmail**, which is typically used to do the work of mailing a message. It takes three parameters −

* The *sender* − A string with the address of the sender.
* The *receivers* − A list of strings, one for each recipient.
* The *message* − A message as a string formatted as specified in the various RFCs.

**CHAPTER 3 : METHODOLOGY AND ALGORITHM**

We developed the application through several steps as shown below in Fig. 1. The steps comprises of all the design and development efforts that we investigated

The following sections describe each step in detail :-

STOP

Go to HOMEPAGE

Mail sent

Successfully added Anniversery

Successfully added Birthday

Enter

Birthday inputs

Enter

Anniversery inputs

Exit the program

If any birthday/

anniversery

then send them

mail

Exit

Check Reminder

Add Anniversery

Add Birthday

Go to HOMEPAGE

STOP