Course Project Documentation

CS101 Project

RemindIt! – Calendar and reminder utility app

TEAM ID:274

KARAN BHUWALKA: 140110086

SOUMITRA CHATTOPADHYAY: 140110025

HIMANI MEHTA: 140110005

ABHISHEK JHINJHA: 140110056

TABLE OF CONTENTS

1. Introduction:

In today’s world where one struggles to keep track of professional and personal commitments our program is designed to cater to these requirements and provide ease in organization. The purpose of this document is to give a complete overview of the Calendar and Reminder utility. Functionalities like adding, editing, deleting and repeating reminders can be easily done through the user friendly and interactive UI of this app.

2. PROBLEM STATEMENT

This is a simple calendar application where the month view and today's agenda are combined together. It gives you a good overview of your events at hand.

- intuitive navigation with transitions between day, week and month view
- flexible recurrences (e.g., every week or every day)
- first select weekly or daily, then set the interval
- Shows markers for dates with events
- Shows today's agenda

3. SOFTWARE REQUIREMENT:

1. Qt Creator – For building the project code.
2. Visual Studio – Used by the Qt creator to display windows and use the debugger.

4. INTERFACE:

User Interfaces

A calendar will be displayed on the screen with the days of the month. Any reminders present will be displayed inside the box containing dates. User can click on this particular date and a window will pop up where user can add, edit or delete reminders for that date.
There will be another version of display where all reminders will be listed. User can select these reminders to edit/delete. User can add reminders through an option in form of a button

Hardware Interfaces

Alarm will ring through speakers.

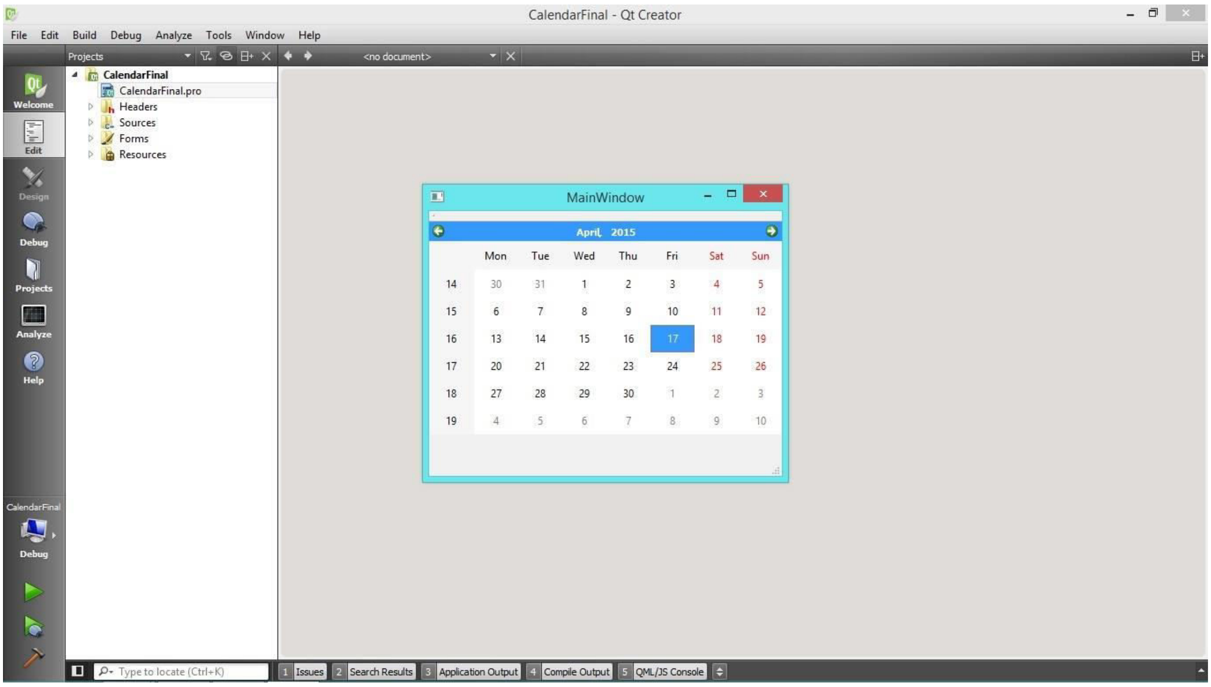
Software Interfaces

Will use the system clock and use that data to keep track of and implement reminders.

4. Implementation:

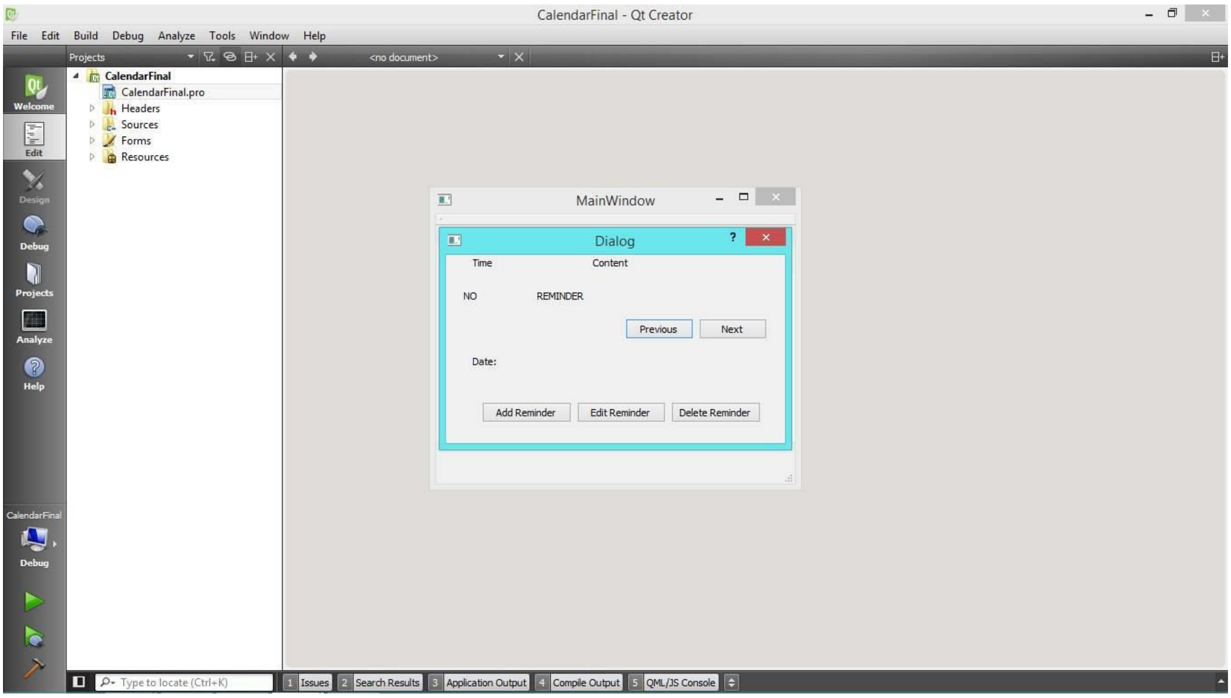
Functionality:

The month view UI, this window is displayed right at the start of the program.



	Mon	Tue	Wed	Thu	Fri	Sat	Sun
14	30	31	1	2	3	4	5
15	6	7	8	9	10	11	12
16	13	14	15	16	17	18	19
17	20	21	22	23	24	25	26
18	27	28	29	30	1	2	3
19	4	5	6	7	8	9	10

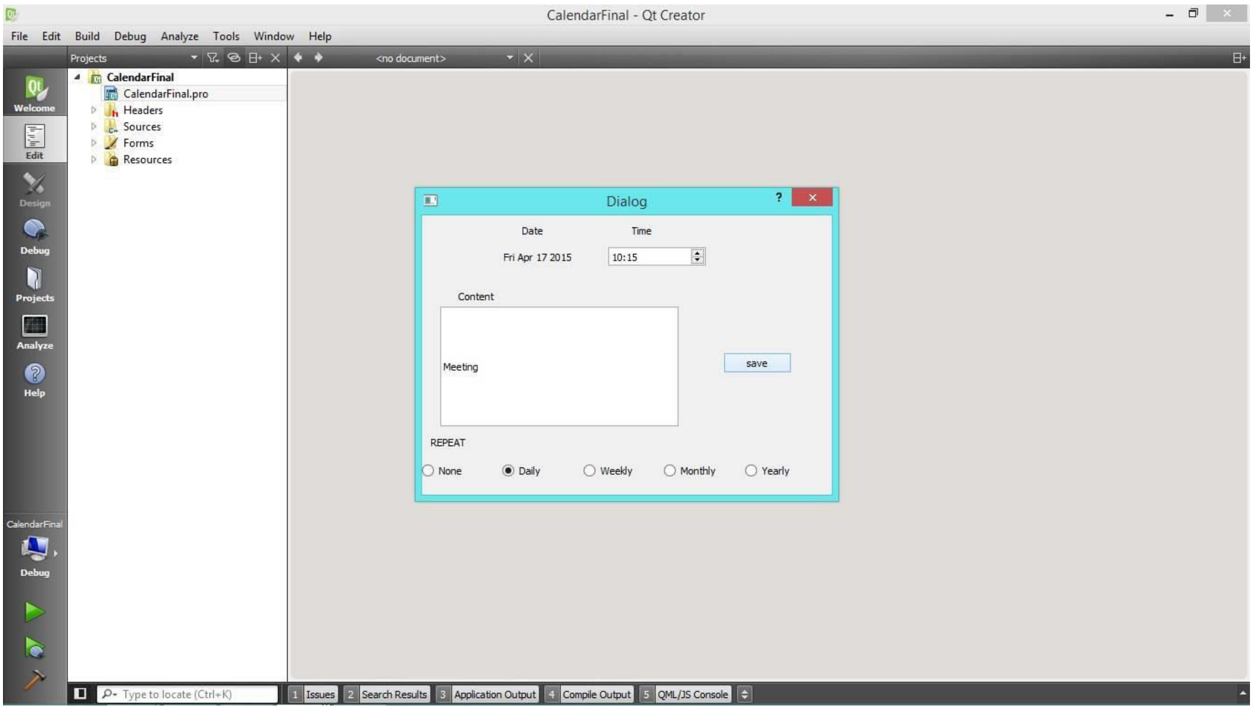
On clicking a particular date, a dialog box pops up, we can add, edit or delete reminders through this window.



Time	Content
NO	REMINDER

Date:

Here is the window for adding the reminder. We can also repeat our reminders on a daily, weekly, monthly, yearly basis.



Date Time

Fri Apr 17 2015 10:15

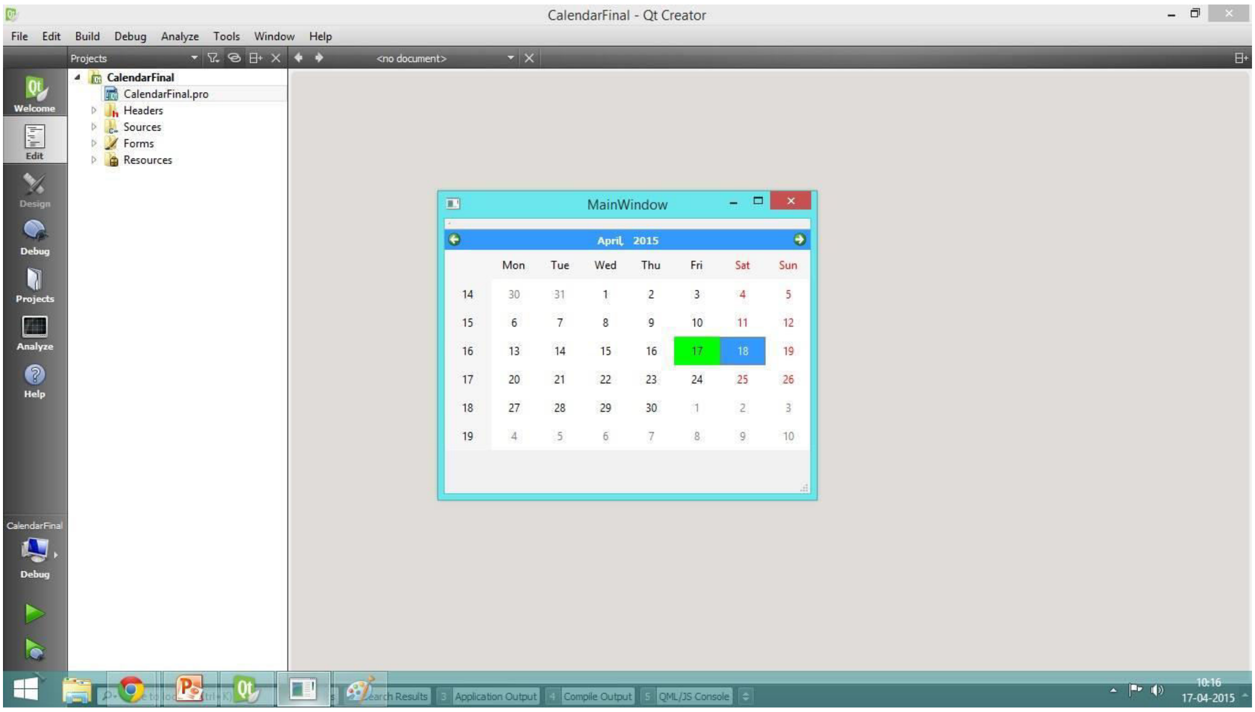
Content

Meeting

REPEAT

○ None ● Daily ○ Weekly ○ Monthly ○ Yearly

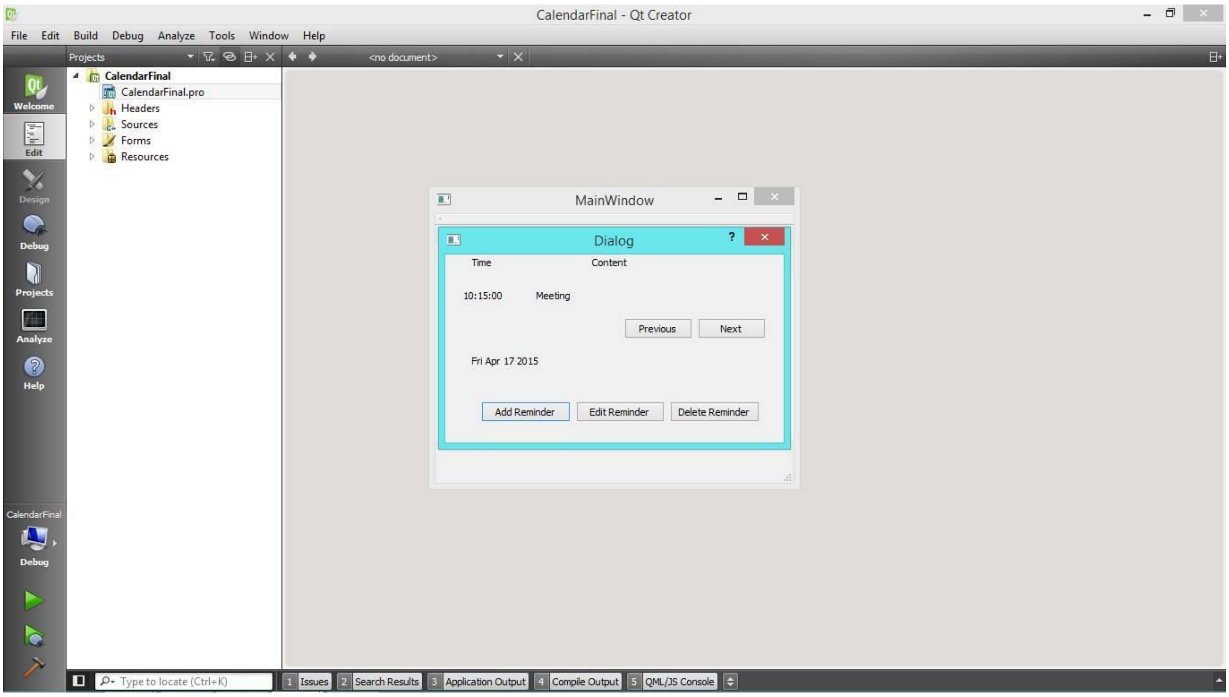
We can view all our reminders through the month view itself.



April, 2015

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
14	30	31	1	2	3	4	5
15	6	7	8	9	10	11	12
16	13	14	15	16	17	18	19
17	20	21	22	23	24	25	26
18	27	28	29	30	1	2	3
19	4	5	6	7	8	9	10

We can also view multiple reminder on one day, using the previous/ next buttons.

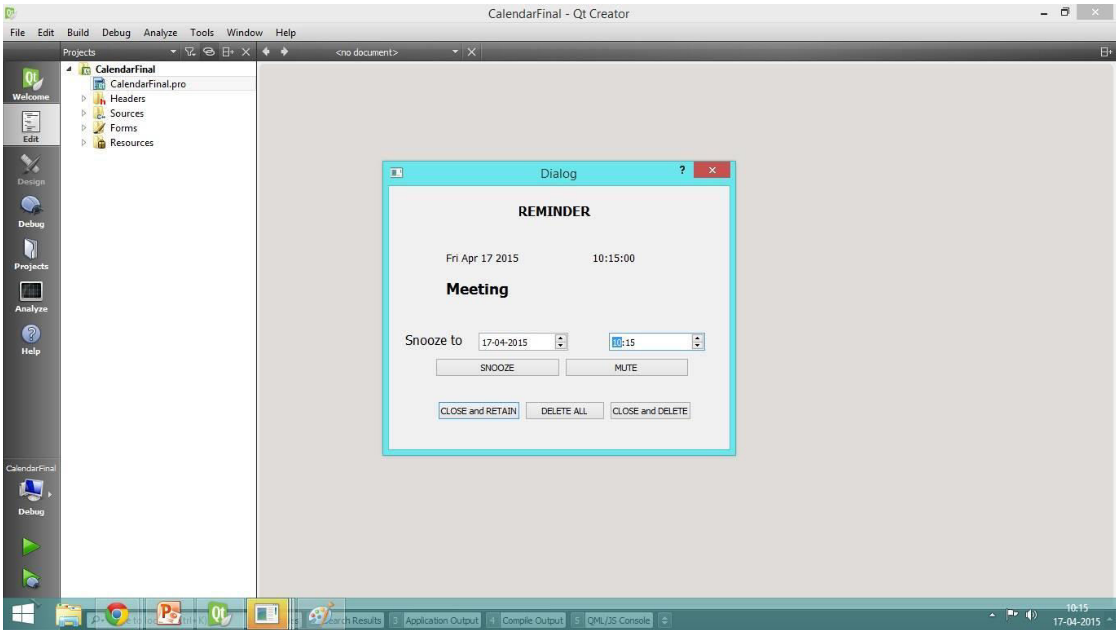


Time Content

10:15:00 Meeting

Fri Apr 17 2015

And when the time is up, a dialog pops up with the system ringer, allowing us to snooze the reminder, retain it, or delete it.



REMINDER

Fri Apr 17 2015 10:15:00

Meeting

Snooze to 17-04-2015 10:15

On clicking “Close and Retain”, It keeps this reminder stored and creates a new entry for it if it’s a repeat reminder. On “Delete All” It deletes its repeat components as well. On “Close and Delete", it deletes this reminder but keeps repeat components. On snoozing, the reminder date/time gets changed to whenever it has been snoozed to.

6. Discussion of System/Challenges:

A) What are worked as per plan?

We were able to learn and implement our program through a GUI

We were able to display a Calendar and access it's properties, so that we can get the position of a particular date and colour in behind it in case a reminder is there. Therefore, we didn't need a dull display that just lists reminders.

We were able to access the computer's ringer which plays a tone on the time of the reminder being reached instead of just a popup

(B) Changes made in plan:

We initially planned on using objects and dynamically allocating memory for reminder date/time/content. However that was not possible and we decided to use arrays to store the reminder data which means there's an upper limit to the number of reminders that can be stored. We've set it to hundred but it can be increased/decreased.

Moreover, because of this, we can't create infinite reminders. This means that if we have a repeat reminder, we can't store it again and again till the calendar ends. And we can't colour in every day of the calendar if it's a daily reminder. Therefore, we decide to create the new reminder (in case of a repeat reminder) only after the previous reminder is reached.

We were unable to access the internet through the platform and thus, could not have the functionality whereby the program sends an email before a reminder to the user.

Bugs:
If there is more than one reminder at a particular time on a particular day, only the one created first is displayed

7. Future Work:

Program can be made to include not only reminders, but also To-Do lists, Memos etc, with different priorities.
Instead of only a normal calendar display, on clicking a date, an hour-wise display can shoot

up, where we can fill in reminders on different hour slots. Reminders can have durations as well.

We can send an email/ text message etc. to the user, in case his reminder is nearing.

We can synchronise the program between different devices, and store reminders there. Moreover, we can sync the application with a networking site like Facebook and store reminders for birthdays of friends, events, etc by just logging into the account.

8. Conclusions:

The experience of making a CS Project was enjoyable and educational for the entire team. We hope we have managed to accomplish an easy-to-use, interactive utility which can be useful in organising and keeping a tab on different events in a person's life. We believe this program can serve as a base and one can go on to add more functionalities/features to make the utility comprehensive and unrestricted to one system.

Through this project, we gained great insight into the effort and challenges required to build a program from ground up. We were challenged at different stages of the project and learnt a lot through these challenges.

9. References:

An Introduction to programming through C++ by Abhiram Ranade.

Lecture Slides by Prof. Kavi J. Arya

YouTube tutorials for Qt Programming:
https://www.youtube.com/watch?v=6KtOzh0StTc&list=PL2D1942A4688E9D63

stackoverflow.com: various questions

Qt Developer Guide: https://qt-project.org/doc/qt-4.7/index.html

Qt Download Link: https://www.qt.io/download-open-source/

Microsoft Visual Studio download link: http://www.microsoft.com/en-us/download/details.aspx?id=44921