TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

PROJECT PROPOSAL

ON

e-Narad : *File Transfer Application*

Submission Date: Ashad , 2078

Submitted by:

Anish Sapkota (076BCT008)

Kushal Subedi (076BCT031)

Nabin Khanal (076BCT036)

Submitted to

Department of Electronics and

Computer Engineering

Acknowledgement

We express our sincere gratitude to all those who have helped us imagine this idea.

Our special thanks goes to our lecturer, Daya Sagar Baral for his guidelines, suggestions and instructions which have served as a contributor towards the inception of this project.

We sincerely thank the Department of Electronics and Computer Engineering, Pulchowk Campus for giving us an opportunity to work on this project to expand our knowledge on Object Oriented Programming and work in a team.

Table of contents

Introduction

e-Narad is a file transfer application where one can send and receive large files through hotspot. It resembles the presently available popular apps like ShareIt, ShareMe to some extent. The app let us transfer a chunk of files between two users connected on the same hotspot.

Objectives

The main objectives to be met in this project can be summarized as follows:

1. To create a project on Object Oriented Programming (OOP) making its concept clear to us.
2. To explore the features of C++ language.
3. To be familiar with resource reusability by making user defined header files.
4. To learn the basics of networking for transferring files over network using SFML
5. To be familiarized with graphics programming using wxWidgets.
6. To make the program occupy minimum memory and be as fast as possible.
7. To make us able to work on major projects in the coming future.
8. To learn to work as a team.

Existing Systems

At present there are many similar applications at various online platforms with different names. The most widely used version of this application is ShareIt for Android, iOS and PC too.

Proposed System

Description

The app UI is just as simple for anyone to get familiar in a minute. It resembles the existing modern app UI so that the person using the app does not feel any difficulties in navigating. The app comprises of three sections: Menubar, Body and Status bar. Menubar is used for quicking

navigation across the app with bunch of tools like its own branding which consist of creating new windows, exits. Also, menubar has About and Help section for the users to know the app more clearly.

Talking about the body, which is the core of our app consists of following buttons for joining across the network and quickly transferring your files across the devices with ease.

- **Send:** This leads to the window where you can select files from your file-browser. The server starts listening to the client.
- **Receive:** This leads to the menu where you wait for connecting to the server and start receiving the file.
- **Exit**: This exits the program.

Finally, talking about the status bar, it shows different quick messages about the events happening in the app so that the user can also know what is happening in the background.

This is just a framework in which we are going to work. There might be more features in the actual project, and the project's actual structure can be slightly different than the one shown by the block diagram.

System Block Diagram

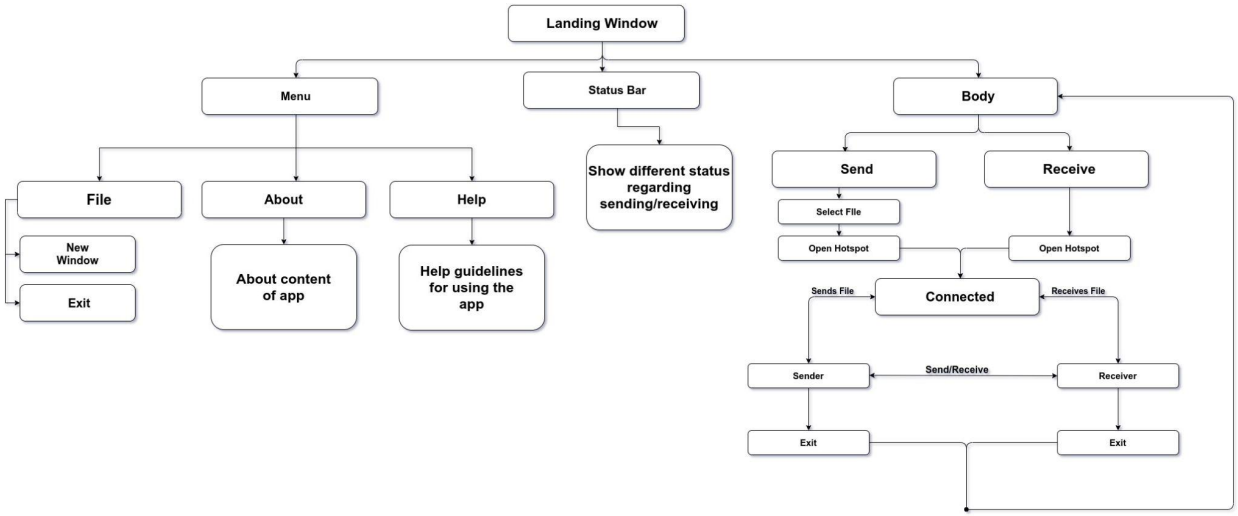

FIG: e-Narad app blockdiagram

Methodology

This project would be based on C++ programming language utilizing wxWidgets, SFML network library, and “Object Oriented Programming” concept. Hence, different classes will be created with the required number of private and public member data and member functions for smooth running of the program preserving the concept of data hiding. The concept of “code reusability”, that is, inheritance will be used. The events for each object will have been handled by the different member functions so that they form a final outlook working together simultaneously.

For the completion of the project, first we will collect necessary materials. We will go through various books and online resources of C++ and Object Oriented Approach.

As for the compiler, IDE and Operating System, we will be using CLion as IDE and GCC (GNU Compiler Collection) as compiler in 3 different Linux distributions viz. Arch, Manjaro and Ubuntu. Providing diversity to the users (players), the program will be cross platform and they can use it on any platform. For graphics we will be using the wxWidgets library.For networking,

we will be using SFML library. After collecting necessary materials, we will develop the algorithm for our project and work as per it. Since the main motto of the project is to learn OOP, the coding will be done utilizing all the features of C++ like objects, classes, data abstraction and encapsulation, inheritance, polymorphism, dynamic binding, message passing etc.

Initially, the basic code for individual components of the application will be written separately adopting the concept of modular programming. In this program, the main components are graphics and networking. Once all the components work independently on the modeling level, they will be binded together to make a proper application.

As per the condition of remote working, we will be using Github for collaboration and our individual code to be merged, and also facebook messenger as a communication platform.

Project Scope

This program has a wide range of opportunities in the real world application. It will serve as a means of utility. As for the system, it will just appear as a clone of the many programs that have been developed till date. If promoted, this application can attract many other similar kinds of applications. It can be developed as an advanced program by adding a nice and vivid user interface and many other features.

Project Schedule

The schedule that we will adopt for our project can be summarized below:

- Analyzing the core concept to develop proper program: 2 day
- Discussion on the project related topics: 1day
- Making project schedule: 1 day
- Initial coding for creating logic: 3 days
- Intense coding the program: 7-10days
- Execution and testing the program: 3 to 4 days

- Debugging: 5 days
- Program Documentation: 2 days

The above mentioned schedule has been planned approximately for providing an aid to develop our project and might get altered according to the circumstances encountered.