DeadDrop

Faran Ahmad
2013CS10220

Kartikeya Gupta
2013CS10231

Prateek Kumar Verma
2013CS10246

COP290: Design Practices

Contents

1 Objectives

We have to build an on-line file management system "Dead Drop" . A server machine maintains the files of multiple users. The user should use a simple desktop application to login into the system. The content of user's account should remain synced with the server.

2 Overall Design

1. We will begin with creating different sub components like a File Transferring System, Credential Verifier, GUI part.
2. Once the components are ready, we will Link this to the network and get basic functionality working on the local-host.
3. Once the local interface is ready, we will take this to the web portal. We will use a server to store data and users will have to send queries to it
4. Once the backend and front end is complete, we will link the two together.

3 Sub Components

1. **User Verification**

Listing 1: Class Parameters for User

```
1 class User
2 {
3     private:
4         std::string UserName;
5         std::string PassWord;
6 };
```

Listing 2: Class Parameters for User

```
1 class UserBase
2 {
3 private:
4     std::unordered_map<std::string, std::string> UsersList;
5 };
```

The User Base is a hash table in which the keys are user-names and the stored values are passwords. When the credentials of the user are to be verified, the key is looked up in the table. Inserting users is also achieved easily using this model. The features which we will be provided to the user will be to verify credentials, add new users and change password.

2. **Files of User**

   We will use boost library to detect changes in files. For each file, the path of the file and last modified time of file is stored in a database.

   Listing 3: Class Parameters for File History

```
class FileHistory
{
    private:
        std::string FolderLocation;
        int TimeOfData;
        std::vector< std::pair<std::string, int> > FileTimeBase;
};
```

   Folder Location is the path of the synced folder. The parameter "TimeOfData" contains the system time at which the data detection was done. This will be used to determine if the server or client side file is newer and then do changes accordingly. "FileTimeBase" is a vector of a string and an integer. The string is the path of the file and the time is the time at which the file was last modified.

3. **Network Managing Part**
   TODO: SOCCER

4. **GUI interface**
   TODO: FARAN

   USER LOGIN

   NEW USER

   USER FILES

   SERVER UI

4 Interaction amongst Sub Components

1. **User Authentication**

   - Client TODO soccer
   - Server TODO soccer.
     The network part mentioned above is linked with the user base file. The instruction to be performed is decoded to be a new user or credential verification. The data base of user names and passwords are accessed for this to take place and changes if needed are made accordingly to it.

2. **File Transfer**

   - Files to Transfer TODO KG

- Transferring TODO soccer

3. **File Sharing and Syncing**
   - File changes TODO KG
   - File Sharing and Permissions TODO KG
   - File Syncing TODO KG
   - File Syncing over network TODO KG TODO Soccer
4. **Front End and Back End**
   - User verification TODO Faran TODO KG
   - Add User Part TODO Faran TODO KG
   - File Managing Part TODO Faran TODO KG

5 Testing Of Components

1. **General Unit Tests**

Listing 4: Class Parameters for Test

```
class Test
{
    private:
        bool verbose;                   //Variable if test is to be conducted
        std::string description;        //String description of the test
        bool isPass;                    //Boolean if the test has passed
        void PrintPassFail(bool);       //Prints the status of the test
};
```

We will use the aformentioned class "Test" to perform unit tests on the different files created. This will ensure that all the functions work correctly against some tests.

2. **File Discovery**
   To test file discovery, a folder with different files will be used. The program will be run on this to obtain the list of files with their modified time and verified to check if it is in accordance with expectations. This will involve new files being created, files being modified and removed.
3. **File Transferring**
   TODO SOCCER
4. **UI Testing**
   TODO FARAN write about individual components
5. **Overall Testing**
   TODO KG once above 2 are done

6 Extra Features

1. De-duplication
2. Server side UI