

College of
Computing

FILE SYNCHRONIZATION TOOL

ABERCHA ZINEB
AIT MAGOURT ADNANE
KEMMOUNE ANASS

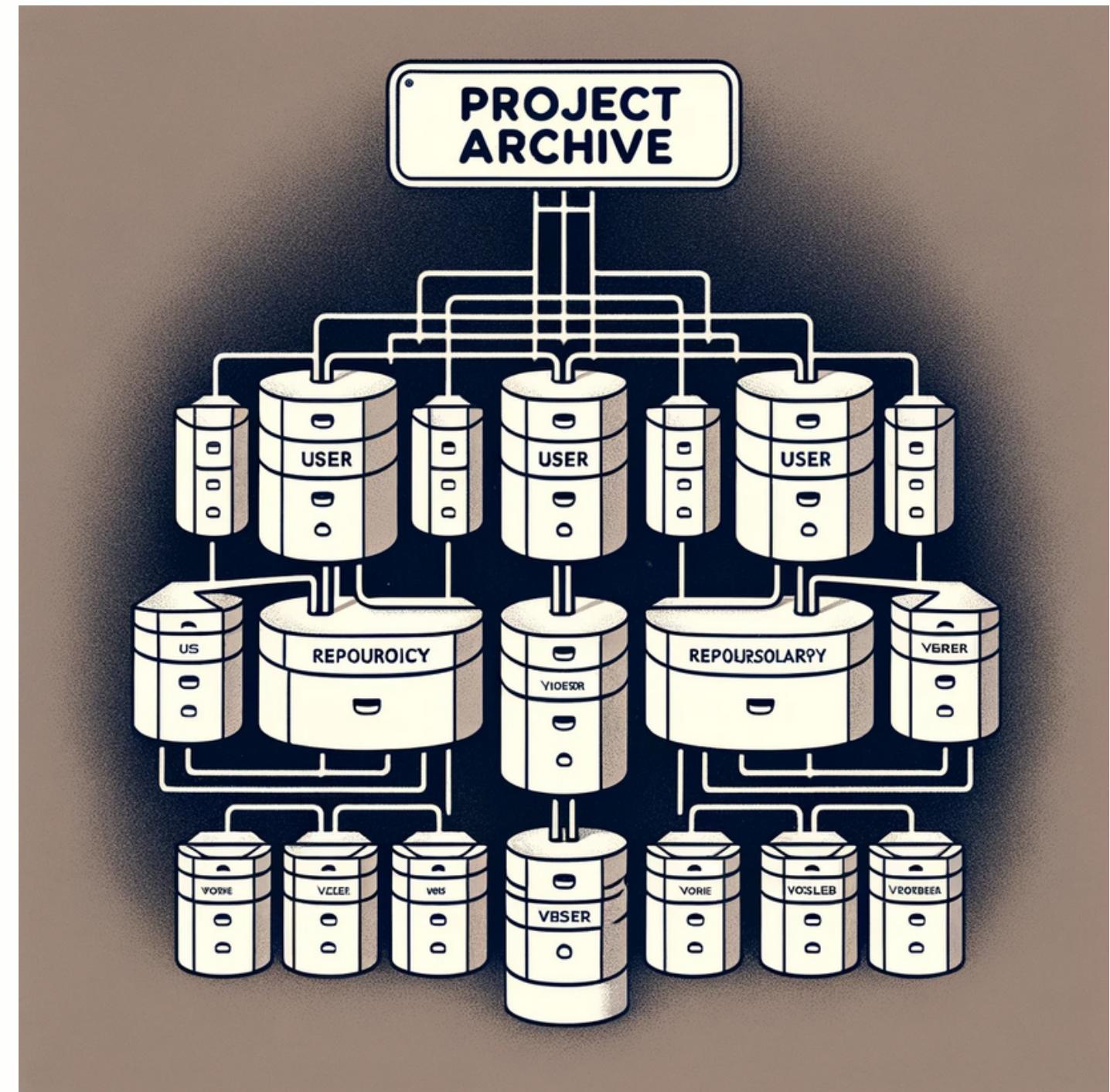


Table of contents:

- 1 APPLICATION STRUCTURE
- 2 MAIN FUNCTIONS
- 3 SNIPPETS OF CODE
- 4 GUI (TEST)

1. Application structure:

The architecture of our project consists of an archive. This archive contains the user folders such that each user folder has repository folders and each one of those contains a folder for each past version.



2. Main functions

There are 2 main files controller.cpp for the backend andmainwindow.cpp for the front end.

1. **login(const QString &username, const QString &password)**
2. **registerUser(const QString &username, const QString &password)**
3. **listUserRepos()**
4. **useRepo(const QString &repoPath, bool isNewRepo, bool cloneRepo)**
5. **initRepository()**
6. **addFile(const QString &fileName)**
7. **commitChanges(const QString &message)**
8. **pull()**
9. **clone()**
10. **getLogLines()**
11. **emptyInfo()**
12. **logout()**

3. Snippets of code

```
100  bool Controller::useRepo(const QString& repoPath, bool isNewRepo, bool cloneRepo) {
101    isNewRepo = isNewRepo;
102    cout << isNewRepo << endl;
103    workingDirectory = repoPath;
104    qDebug() << "Using repo: " << workingDirectory;
105    repoName = QString::fromStdString(lastFolderName(workingDirectory.toStdString()+"\\"));
106    cout << repoName.toStdString() << endl;
107
108    if(cloneRepo){
109      clone();
110    }
111    else if (isNewRepo) {
112      initRepository();
113      // Copy whole repo to archive
114      int newVersion = 0;
115      QString newVersionPath = archiveDirectory + "\\\" + username + "\\\" + repoName + "\\\" + QString::number(newVersion);
116      QDir().mkpath(newVersionPath);
117      copy_directory(workingDirectory.toStdString(), newVersionPath.toStdString());
118
119      // Update lastVer and lastVerPath
120      lastVer = newVersion;
121      lastVerPath = newVersionPath;
122    } else {
123      lastVer = getGreatestFolderVersion((archiveDirectory + "\\\" + username + "\\\" + repoName).toStdString());
124      if (lastVer == -1) {
125        throw std::runtime_error("No version found");
126      }
127      lastVerPath = getLastRepoVersionPath();
128    }
129
130    // Process each line (replace this with your logic)
131    for (const QString& line : lines) {
132      if(!controller->addFile(line)){
133        ui->textEdit->setText(ui->textEdit->toPlainText() + "\nFile: " + line + "not changed!");
134      }
135    }
136    qDebug() << "Line:" << line;
137  }
138 }
```

Not Committed Yet 2023-12-28

4. Test (GUI)

Thank you for your attention

