# Introduction

## Description:

GonqBox is a bilingual online file sharing application. Some of the features which GonqBox supports is the ability for users to upload, download, comment on, and delete files from within a single virtual directory. GonqBox allows users to grant permissions to both individual files, and the directory, for example, a user may allow another user of the system to view, and additionally download files from within his/her directory. In addition, Users may leave comments associated to directories and files which are to be read and dismissed by the appropriate user.

## Members:

* Gabriel Hounsome
* Joshua Basque
* Mathew Boland
* Michael O’Neil

# Architecture

## Types of Data:

* Files
* Comments
* Users
* User roles
* Folders
* Permissions
* Translations

## Functions and Features:

* Add files
* Delete files
* Update files
* Share Files
* Setting File Permissions
* Add/Register users
* Delete users
* Update users
* Lock file
* Translated UI
* Folder/File Sharing

## Network Diagram:

## 

## People:

User roles can be broken down into: site administrators, content owners, content collaborators, content viewers, and site guests.

Without registration a client is defaulted to site guest, site guests are only able to visit directories which have a file that the content owner author has made visible, and they are only able to see visible files.

Once a guest registers they obtain the content viewer permissions, this allows the user to leave comments on public directories as well as setting up their own directory, they retain all permissions established in the guest role.

Once a user has become a content viewer they are then able to upload files to their own directory, once they have done so they would be granted content owner permission over that upload, with this permission they can decide on adding file collaborators, file visibility, file write, file deletion, as well as all previous permissions.

If a user has been requested to collaborate on a file they would be granted permission to write over the existing file, as well as changing the files visibility, all other permissions would remain with the content owner.

Site administrators are able to add, edit, and delete both users and files.

## Timing:

Critical processes:

* Sitemap sketch
* Database diagram
* Create initial classes

When user uploads a file, it is there immediately. There is no verification of the file.

## Motivation:

* Secure file hosting through encryption of both user login information and directory content.
* Files are to be encrypted on storage, as a result the users encrypted files are useless if retrieved by third parties.
* Private directories/files by default, unless specified by content creator or collaborator.
* Users can share to specific friends
* User can make a specific file completely public
* User can make their folder completely public

# Project Plan

Version 0.1:

|  |  |  |
| --- | --- | --- |
| Task | Assignee(s) | Time Allotment |
| Sitemap sketch | Group | 4 Hours |
| Initial JSP views with little dynamic content. | Gabriel, Mathew | 6 Hours |
| Initial Database schema design | Joshua Basque | 2 Hours |
| Fill tables with test data | Joshua Basque | 1 Hours |
| Create initial controller classes. | Group | 6 Hours |
| Create initial model classes. | Group | 3 Hours |
| Finish JSP views. | Group | 3 Hours |

Version 0.2:

|  |  |  |
| --- | --- | --- |
| Task | Assignee | Time Allotment |
| Add file uploading |  |  |
| Add commenting Support |  |  |
| Implement max folder size |  |  |
| Add translation for UI | Gabriel Hounsome | 2 Hours |

Version 0.3:

|  |  |  |
| --- | --- | --- |
| Task | Assignee | Time Allotment |
| Add file collaboration |  |  |
| File Encryption |  |  |
| Add File Sharing |  |  |
| Add Folder Sharing |  |  |