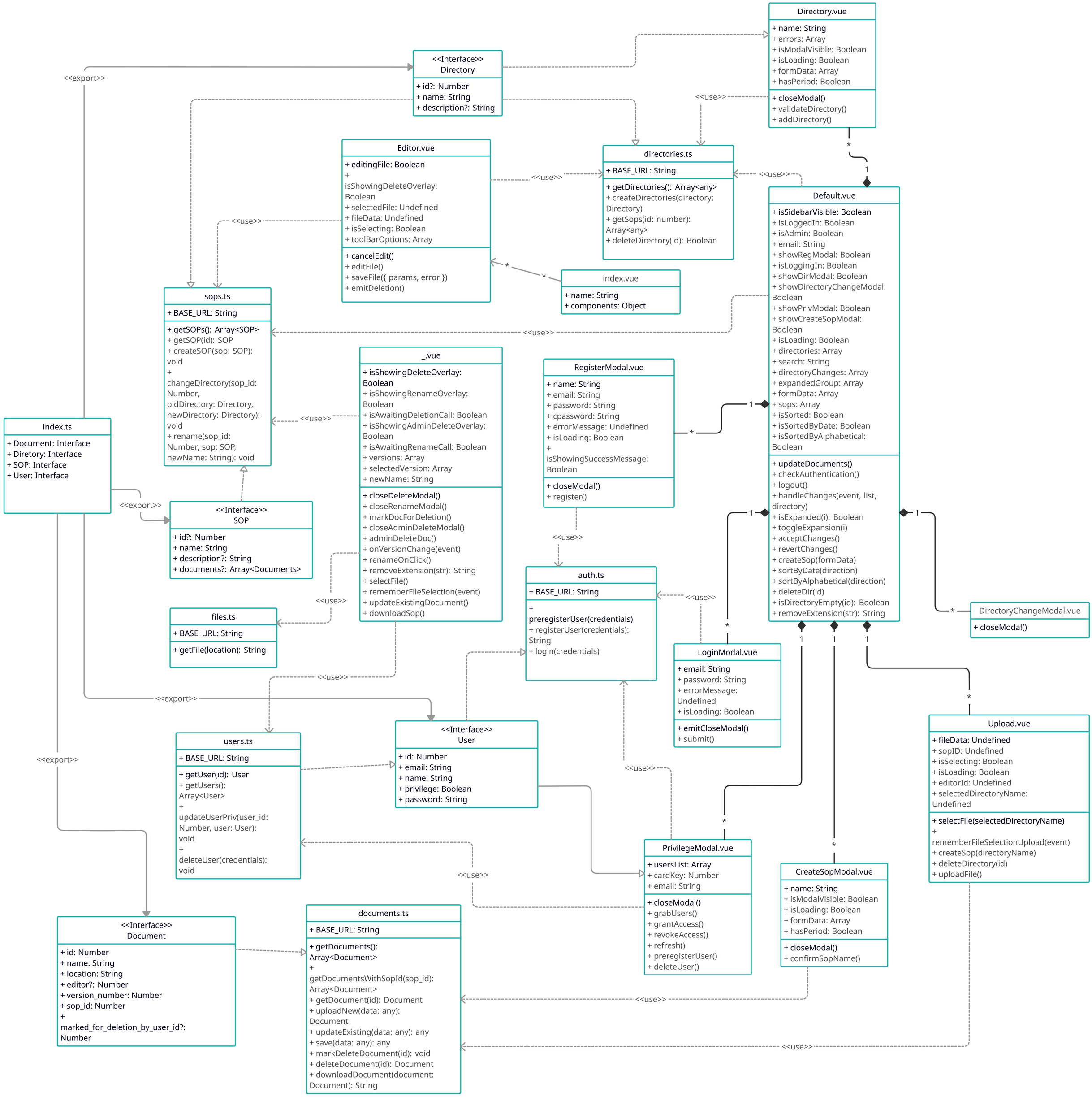
Directory.vue
+ name: String
+ errors: Array
+ isModalVisible: Boolean
+ isLoading: Boolean
+ formData: Array
+ hasPeriod: Boolean
+ closeModal()
+ validateDirectory()
+ addDirectory()
<<Interface>>
Directory
+ id?: Number
+ name: String
+ description?: String
<<export>>
<<use>>
Editor.vue
+ editingFile: Boolean
+ isShowingDeleteOverlay: Boolean
+ selectedFile: Undefined
+ fileData: Undefined
+ isSelecting: Boolean
+ toolBarOptions: Array
+ cancelEdit()
+ editFile()
+ saveFile({ params, error })
+ emitDeletion()
directories.ts
+ BASE_URL: String
+ getDirectories(): Array<any>
+ createDirectories(directory: Directory)
+ getSops(id: number): Array<any>
+ deleteDirectory(id): Boolean
Default.vue
+ isSidebarVisible: Boolean
+ isLoggedIn: Boolean
+ isAdmin: Boolean
+ email: String
+ showRegModal: Boolean
+ isLoggingIn: Boolean
+ showDirModal: Boolean
+ showDirectoryChangeModal: Boolean
+ showPrivModal: Boolean
+ showCreateSopModal: Boolean
+ isLoading: Boolean
+ directories: Array
+ search: String
+ directoryChanges: Array
+ expandedGroup: Array
+ formData: Array
+ sops: Array
+ isSorted: Boolean
+ isSortedByDate: Boolean
+ isSortedByAlphabetical: Boolean
+ updateDocuments()
+ checkAuthentication()
+ logout()
+ handleChanges(event, list, directory)
+ isExpanded(i): Boolean
+ toggleExpansion(i)
+ acceptChanges()
+ revertChanges()
+ createSop(formData)
+ sortByDate(direction)
+ sortByAlphabetical(direction)
+ deleteDir(id)
+ isDirectoryEmpty(id): Boolean
+ removeExtension(str): String
index.vue
+ name: String
+ components: Object
sops.ts
+ BASE_URL: String
+ getSOPs(): Array<SOP>
+ getSOP(id): SOP
+ createSOP(sop: SOP): void
+ changeDirectory(sop_id: Number, oldDirectory: Directory, newDirectory: Directory): void
+ rename(sop_id: Number, sop: SOP, newName: String): void
_.vue
+ isShowingDeleteOverlay: Boolean
+ isShowingRenameOverlay: Boolean
+ isAwaitingDeletionCall: Boolean
+ isShowingAdminDeleteOverlay: Boolean
+ isAwaitingRenameCall: Boolean
+ versions: Array
+ selectedVersion: Array
+ newName: String
+ closeDeleteModal()
+ closeRenameModal()
+ markDocForDeletion()
+ closeAdminDeleteModal()
+ adminDeleteDoc()
+ onVersionChange(event)
+ renameOnClick()
+ removeExtension(str): String
+ selectFile()
+ rememberFileSelection(event)
+ updateExistingDocument()
+ downloadSop()
RegisterModal.vue
+ name: String
+ email: String
+ password: String
+ cpassword: String
+ errorMessage: Undefined
+ isLoading: Boolean
+ isShowingSuccessMessage: Boolean
+ closeModal()
+ register()
index.ts
+ Document: Interface
+ Diretory: Interface
+ SOP: Interface
+ User: Interface
<<Interface>>
SOP
+ id?: Number
+ name: String
+ description?: String
+ documents?: Array<Documents>
auth.ts
+ BASE_URL: String
+ preregisterUser(credentials)
+ registerUser(credentials): String
+ login(credentials)
files.ts
+ BASE_URL: String
+ getFile(location): String
LoginModal.vue
+ email: String
+ password: String
+ errorMessage: Undefined
+ isLoading: Boolean
+ emitCloseModal()
+ submit()
DirectoryChangeModal.vue
+ closeModal()
Upload.vue
+ fileData: Undefined
+ sopID: Undefined
+ isSelecting: Boolean
+ isLoading: Boolean
+ editorId: Undefined
+ selectedDirectoryName: Undefined
+ selectFile(selectedDirectoryName)
+ rememberFileSelectionUpload(event)
+ createSop(directoryName)
+ deleteDirectory(id)
+ uploadFile()
users.ts
+ BASE_URL: String
+ getUser(id): User
+ getUsers(): Array<User>
+ updateUserPriv(user_id: Number, user: User): void
+ deleteUser(credentials): void
<<Interface>>
User
+ id: Number
+ email: String
+ name: String
+ privilege: Boolean
+ password: String
PrivilegeModal.vue
+ usersList: Array
+ cardKey: Number
+ email: String
+ closeModal()
+ grabUsers()
+ grantAccess()
+ revokeAccess()
+ refresh()
+ preregisterUser()
+ deleteUser()
CreateSopModal.vue
+ name: String
+ isModalVisible: Boolean
+ isLoading: Boolean
+ formData: Array
+ hasPeriod: Boolean
+ closeModal()
+ confirmSopName()
<<Interface>>
Document
+ id: Number
+ name: String
+ location: String
+ editor?: Number
+ version_number: Number
+ sop_id: Number
+ marked_for_deletion_by_user_id?: Number
documents.ts
+ BASE_URL: String
+ getDocuments(): Array<Document>
+ getDocumentsWithSopId(sop_id): Array<Document>
+ getDocument(id): Document
+ uploadNew(data: any): Document
+ updateExisting(data: any): any
+ save(data: any): any
+ markDeleteDocument(id): void
+ deleteDocument(id): Document
+ downloadDocument(document: Document): String