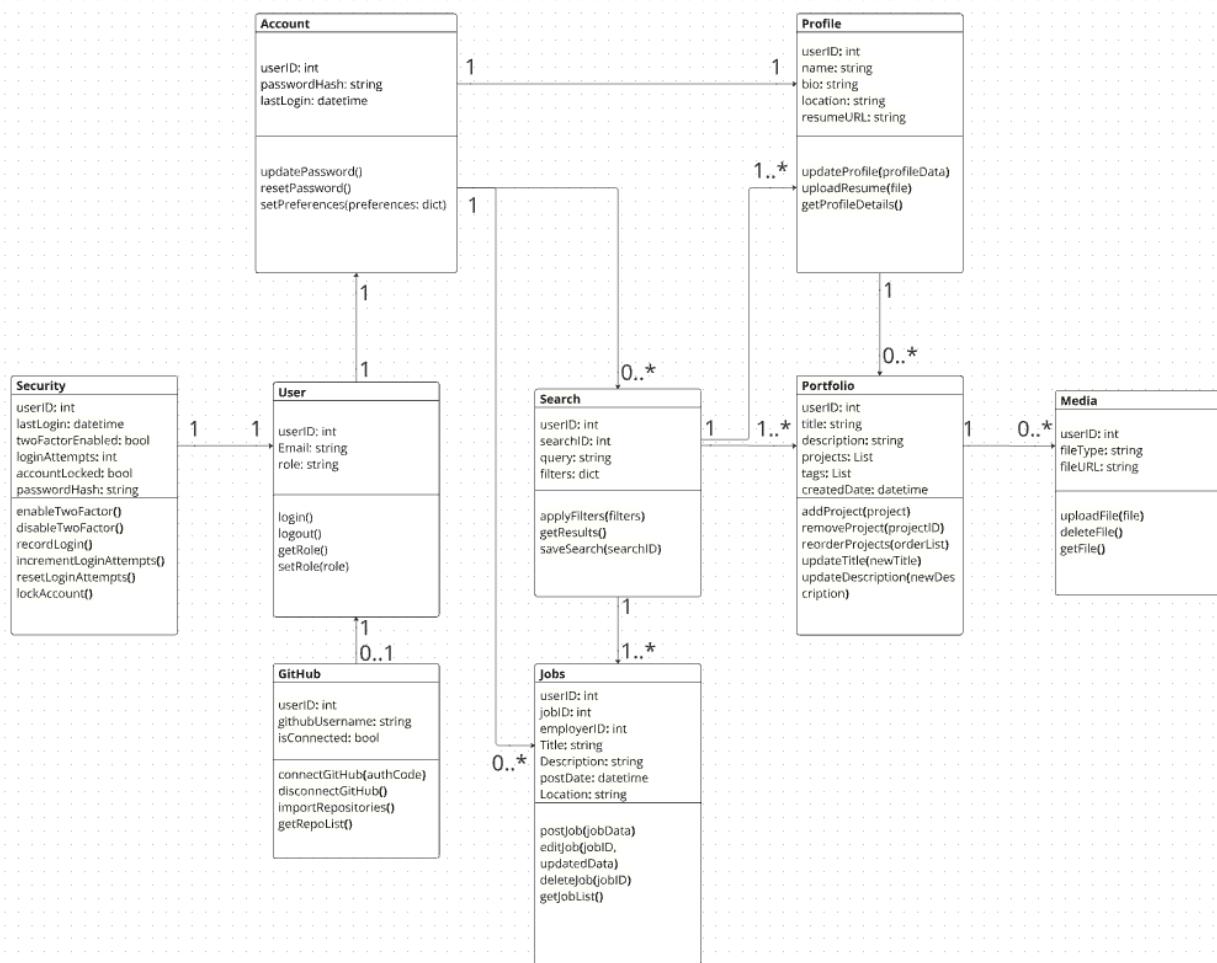


COS 420  
 Team D (Dingo)  
 Deliverable 3  
 10/30/2025

# Domain Model - SkillShow

The User component is the core of the backend. It serves to connect Security and Github functionality to the Account. The Account component connects to the core functionalities of the application, Profile, Jobs, and Search. An Account can use Profile to change their information and upload data. Profiles can contain Portfolios which showcase information through text or media. An Account can post or apply to Jobs and view their information. An Account can find Profiles, Portfolios, and Jobs through Search.



Components:

- **User**
  - userID: string
  - Email: string
  - role: string
  - login()
  - logout()
  - getRole()
  - setRole(role)
- **Account**
  - userID: int
  - passwordHash: string
  - lastLogin: datetime
  - updatePassword(newPassword)
  - resetPassword(token)
  - setPreferences(preferences: dict)
- **Profile**
  - userID: int
  - name: string
  - bio: string
  - location: string
  - resumeURL: string
  - updateProfile(profileData)
  - uploadResume(file)
  - getProfileDetails()
- **Portfolio**
  - userID: int
  - title: string
  - description: string
  - projects: List<Project>
  - tags: List<string>
  - createdDate: datetime
  - addProject(project)
  - removeProject(projectID)
  - reorderProjects(orderList)
  - updateTitle(newTitle)
  - updateDescription(newDescription)
  - addTag(tag)
  - removeTag(tag)
  - publish()
  - unpublish()
  - getPortfolioSummary()
- **GitHub**

- userID: int
- githubUsername: string
- isConnected: bool
- connectGitHub(authCode)
- disconnectGitHub()
- importRepositories()
- getRepoList
- **Media**
  - userID: int
  - fileType: string
  - fileURL: string
  - uploadFile(file)
  - deleteFile
  - getFile()
- **Jobs**
  - userID: int
  - jobID: int
  - employerID: int
  - Title: string
  - Description: string
  - postDate: datetime
  - Location: string
  - postJob(jobData)
  - editJob(jobID, updatedData)
  - deleteJob(jobID)
  - getJobList()
- **Search**
  - userID: int
  - searchID: int
  - query: string
  - filters: dict
  - applyFilters(filters)
  - getResults()
  - saveSearch(searchID)
- **Security**
  - userID: int
  - lastLogin: datetime
  - twoFactorEnabled: bool
  - loginAttempts: int
  - accountLocked: bool
  - passwordHash: string
  - enableTwoFactor()
  - disableTwoFactor()
  - recordLogin()

- `incrementLoginAttempts()`
- `resetLoginAttempts()`
- `lockAccount()`
- `unlockAccount()`
- `updatePassword(newPassword)`
- `validateSession(token)`