ClusterGit

Requirements Model

Version 1.0

Team Members:
Lauren Robison
Conner Reiter
Sean Sikora
Brandon Springer
Sasha Tapinsh

Title: Preliminary Requirements Model - Team#12	
Version: 1.0	Date: 10/10/25

Revision History

Date	Version	Description	Name(s)
9/26/25	.1	Rough Draft of requirements , use case diagrams, and user stories	Sasha Tapinsh
10/10/25	1.0	Finalization	Sasha Tapinsh

Table of Contents

Requirements 3

Functional Requirements 3

Non-Functional Requirements 4

Product Backlog 4

User stories 5

Faculty 5

Student 5

Admin 5

Use Cases: 6

Faculty Use Case Diagram 6

Admin Use Case Diagram 7

Student Use Case Diagram 8

Requirements

Functional Requirements

The team will create a self-hosted git server with custom requirements:

- There are two views: A student view and a faculty view.
- Students and faculty can register and login to their respective views.
- Students will be able to create repositories, commit large files (greater than two gigabytes), as well as encrypted files.
- Faculty will be able to remove repositories as the storage of the server gets full.
- Students will be able to access a web portal and view repositories, manage collaborators, approve and reject pull requests, and leave comments on pull requests.
- Students will be able to push to git repositories via git annex commands
    - Ex:
      ```
      git annex add .
      add big_file (checksum … ) ok
      git commit -a -m added
      ```
- A database server will be developed and maintained which contains user sign-ins and their repositories.

The team will also interview various professors and students after the repository is running in order to upgrade the quality of life. Based on feedback from students and teachers we can improve and refine our system to become better for both user groups.

Non-Functional Requirements

- **Security:** Since the repository system will be hosted on campus, security is a high priority, as to not introduce security vulnerabilities.
  - The database should only allow users to access information that is relevant to the user, and aptly handle permissions per repository.
- **Efficiency & Performance:**
  - The server needs to receive large files fast while not overloading the cluster computer.
- **Storage:** The system requires a lot of storage which is hopefully going to be achieved by the acquisition and linking of a NAS.
- **Availability:** Must be accepting commits at all times, especially since students enjoy waiting until the last second to work on assignments.
- **Usability:** The server must be easy to use and documentation provided must give a walkthrough on specific commands to enable large file commits as well as encrypted file commits.

Product Backlog

1. **Git support**
   a. Users can clone, push, pull, commit to repositories stored on server
   b. Git-Annex used to enable large-file version control and encrypted file support
2. **User Registration and Login**
   a. User register and login using **@pfw.edu** email address.
   b. Admin can manually move different accounts between faculty group and student group
   c. Support for collaboration invites.
3. **Repository Creation**
   a. User can create repos
   b. Teacher can create repos for students
   c. Users can manage who is a collaborator and who can commit to repositories
4. **Download and Upload support from Website**
   a. Users can upload files and folders directly to a repo from the website
   b. Users can download files from repo as a zip to keep on local machine without using git

Title: Preliminary Requirements Model - Team#12	
Version: 1.0	Date: 10/10/25

User stories

Faculty

- As a faculty user, I want to clone student repositories so I can grade larger file sized assignments
- As a faculty user, I want to view collaborators and contributions so I can ensure even split of workload on group projects
- As a faculty user, I want to view pull requests so I can suggest changes to projects
- As an instructor, I want files to be consistent even if a node fails so grading isn’t blocked.

Student

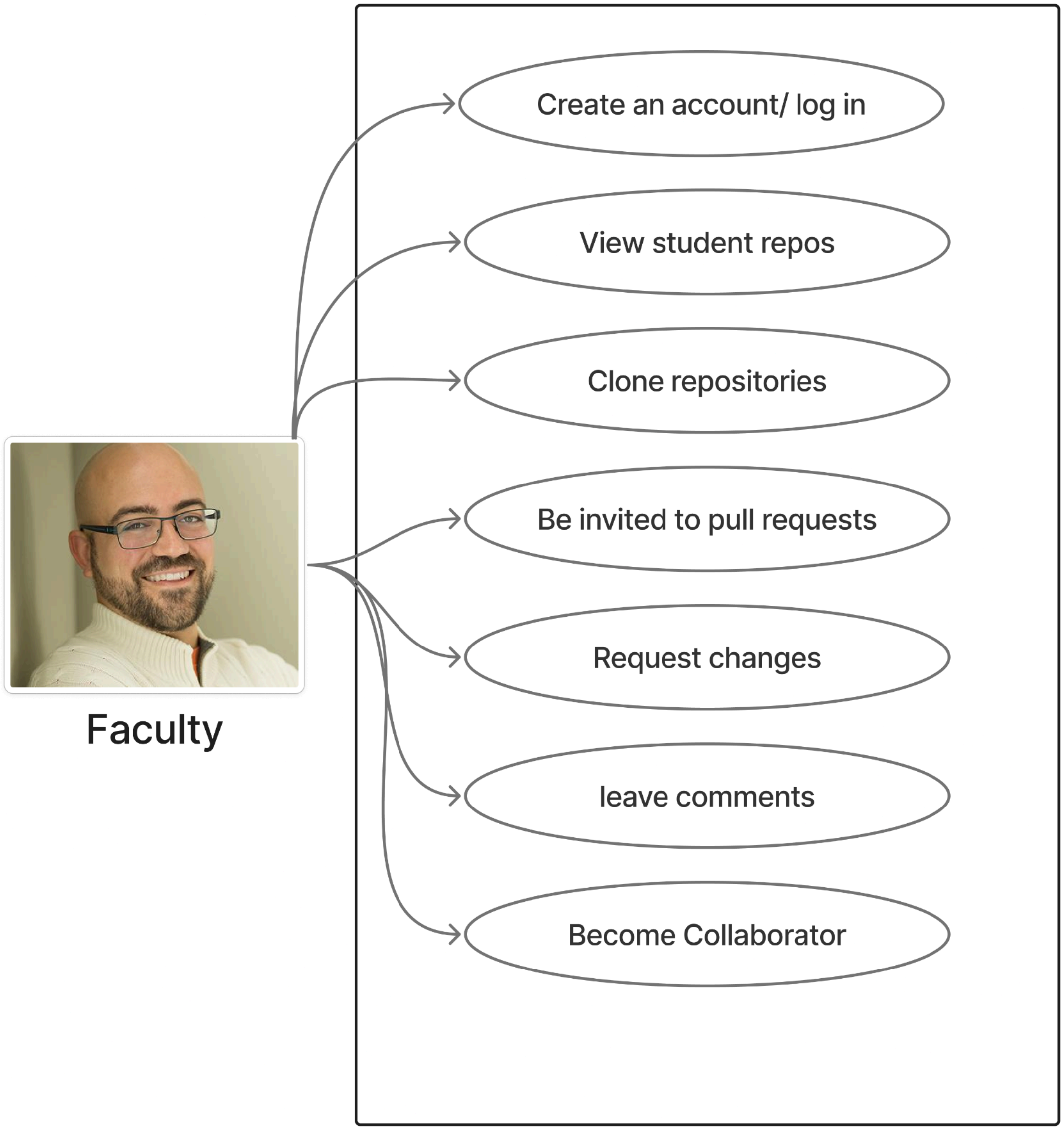
- As a student user, I want to be able to create repositories so I can submit my assignments
- As a student user, I want to be able to invite users to review pull requests so I can have my work peer reviewed
- As a student user, I want to be able to commit my assignments into a repository so I can have version control and have my previous assignments backed up.
- As a student, I want to upload a multi-GB project via a CLI so that my submission completes reliably.
- As a student, I want a web UI to submit my project so I don’t need the CLI.

Admin

- As an admin, I want to see all Pi nodes healthy in K3s so I can schedule workloads.
- As an admin, I want to be able to delete old files so I can preserve storage for future submissions
- As an admin, I want to be able to change user types so I can make users seen as teachers (graders) or students
- As an admin, I want to be able to add more storage so I can not be limited by storage capacity when dealing with larger files.
- As an admin, I want a standard Git-annex remote and content policy so large files land on the cluster, not laptops.

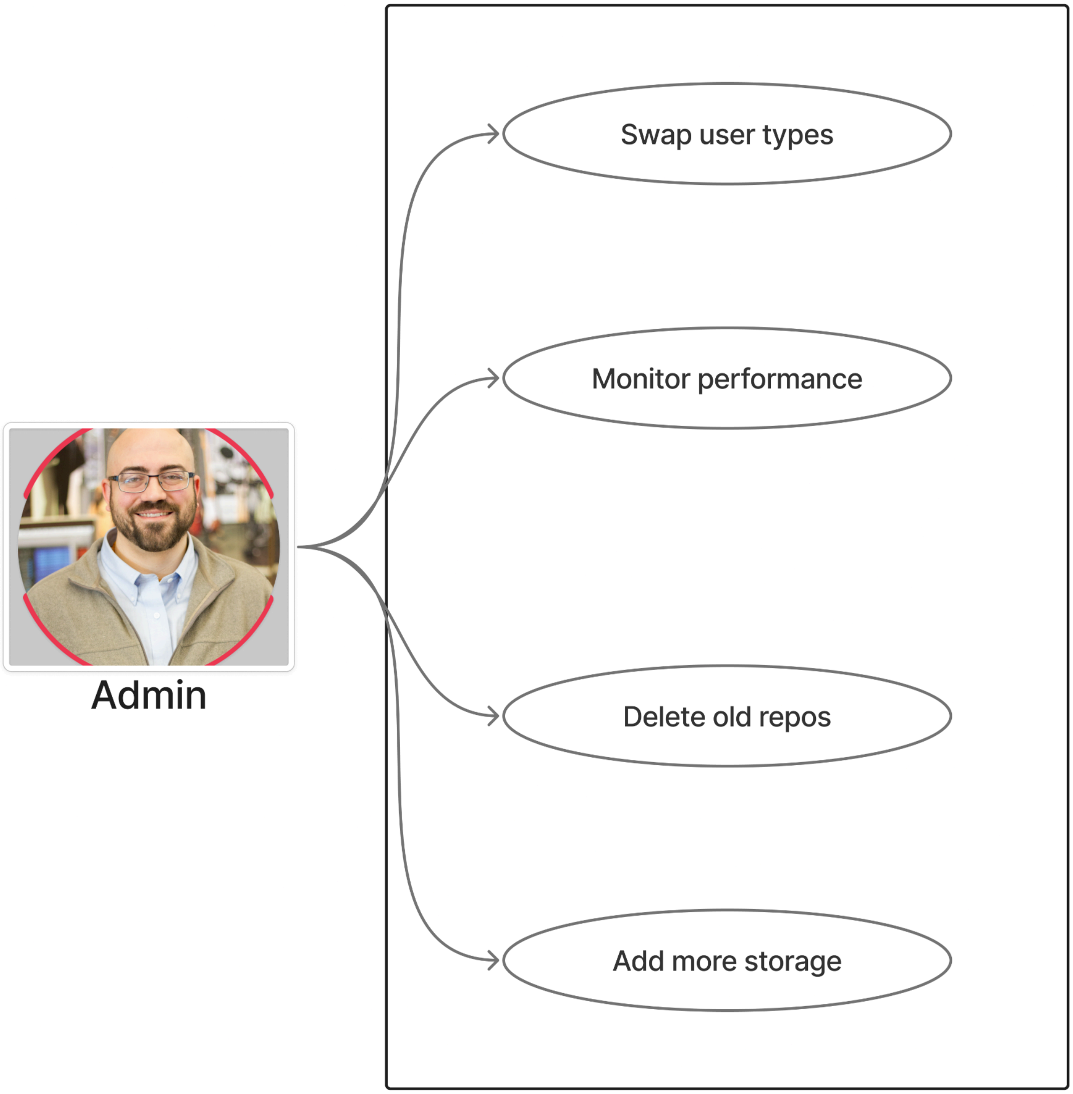
Use Cases:

Faculty Use Case Diagram



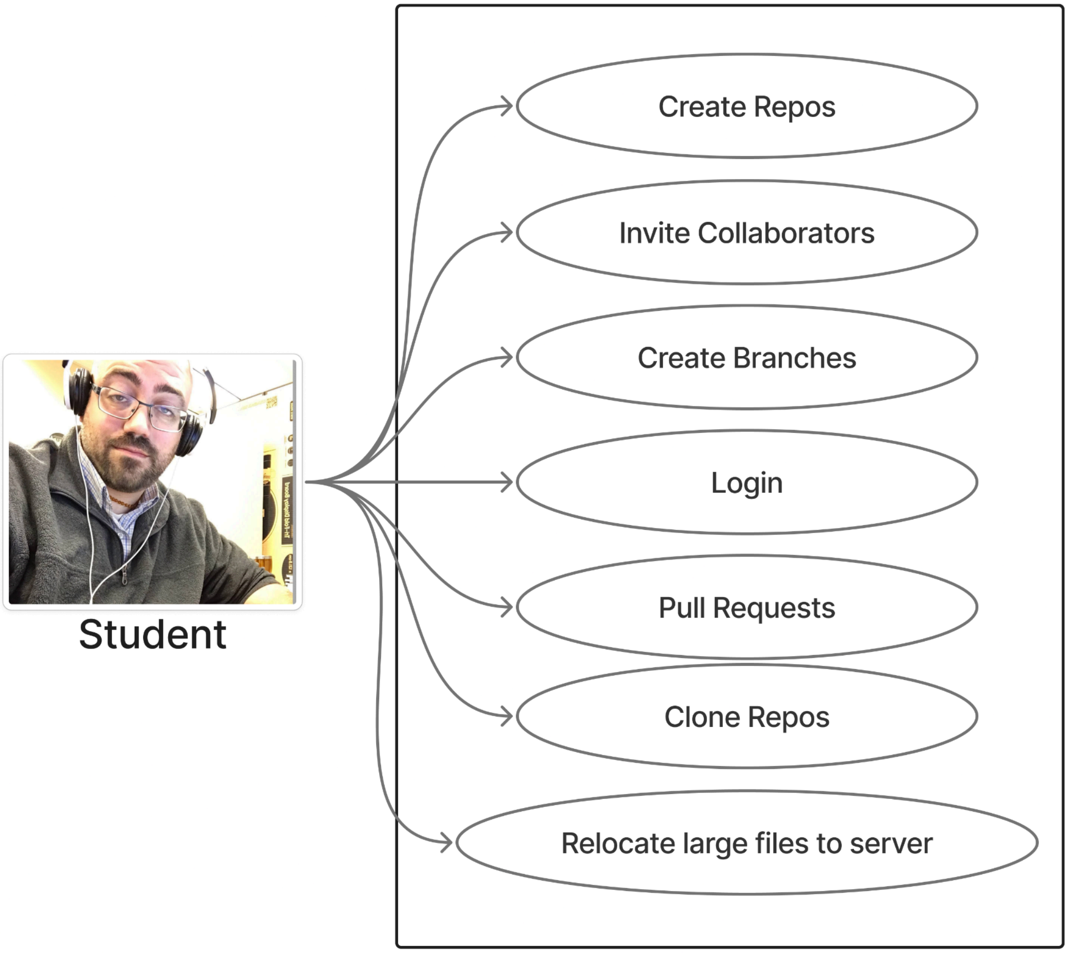
Title: Preliminary Requirements Model - Team#12	
Version: 1.0	Date: 10/10/25

Admin Use Case Diagram



Title: Preliminary Requirements Model - Team#12	
Version: 1.0	Date: 10/10/25

Student Use Case Diagram


Create Repos
Invite Collaborators
Create Branches
Student
Login
Pull Requests
Clone Repos
Relocate large files to server