

# Logging

## Objectives:

- Log all user activity including:
  - Log ins, successful and unsuccessful attempts
  - Logouts
  - All user posts
  - All user actions
  - All errors

## Background:

Logging user activity and errors will provide proper feedback and knowledge for improving the web app as well as finding and correcting common errors in code. Log ins will be logged to prevent an infinite loop of login attempts as well as potentially blocking a user from having unlimited login attempts. Logging errors is essential for fixing issues as soon as possible.

## Requirements:

- Log successful login attempts
- Log unsuccessful login attempts
- Log logouts
- Log all user posts
- Log all user actions on the web app
- Log all errors
- Privacy needs to reflect that user actions are being logged.

## Non-Functional Requirements:

- 

**Success:** All user actions and errors will be successfully logged and available for viewing and assessment.

**Failure:** Not all user actions or errors will be logged, leaving a black-area of things that might not be able to be assessed for fixing or addressing.

### 4.25.1 Functional

Req#	Priority	Description	Rationale
<b>General / Base Functionality</b>			
FR-LA-G-001	1	Log all errors	So that developers are able to address errors and fix them in a timely manner.
FR-LA-G-002	1	Log all user actions	To track user activity and see common trends.

<b>Security Requirements</b>			
FR-LA-S-001	1	Log all login attempts	This will help distinguish potential login attackers and potentially ban them by IP.

## User Access Controls Diagram

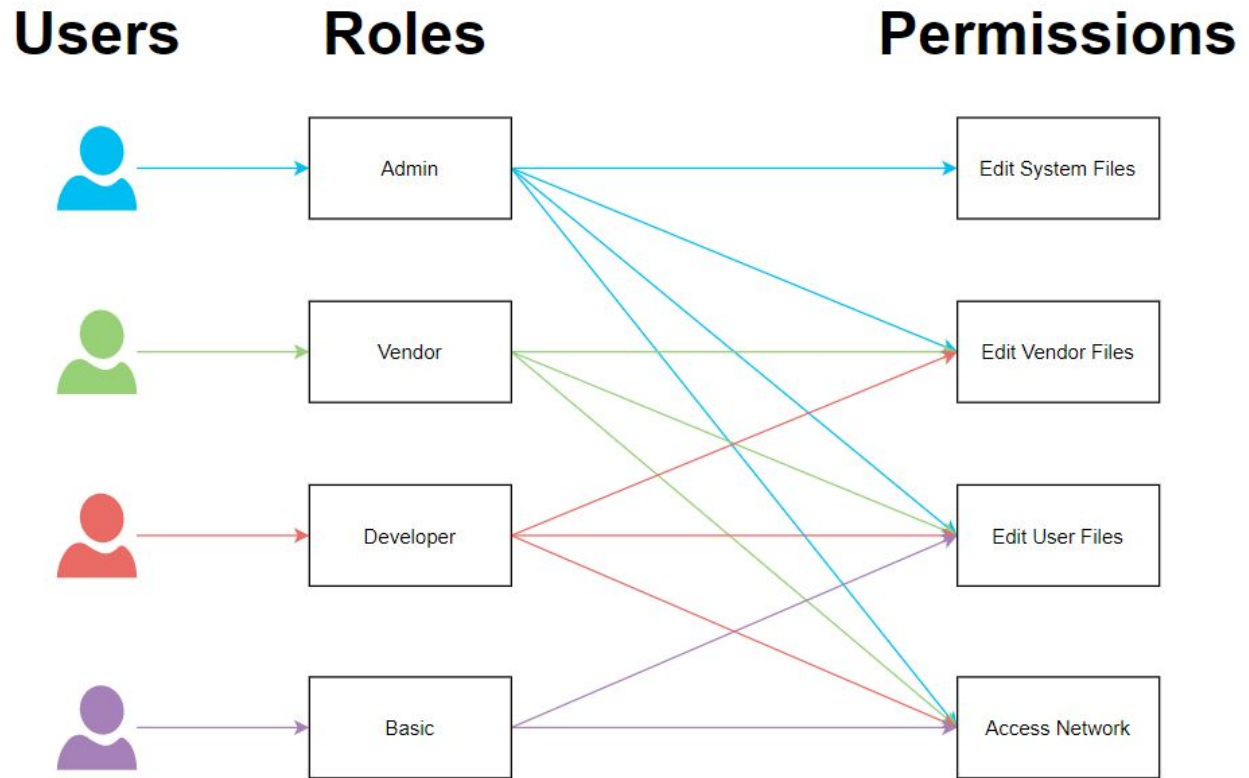


Fig.1 Role-Based Access Control