Team Members:
Jason Boulware
Austin Hollis
Jeffrey Williams
Rowen Loucks
November 28, 2018
Software Engineering Team Term Project:
Deliverable #5

Introductions:

For this week's deliverable, our team had to do 5 fault injections in the code and test how those fault injections affect our test cases.

Process:

For our fault injections, we used the following methods: fraction_to_milliseconds(str_milli), centiseconds_to_milliseconds(centi), isAllowedFilename(filename) and can_view_activity(user, viewing_user).

Inside of these methods were if statements and we would insert a 'not' to get the opposite if it was a single if statement without an else statement. We would change an 'and' to an 'or' and vice versa.

Original	**def fraction_to_milliseconds(str_milli):
	if not str_milli:
	return 0
	...
	...**

Fault Injection	**def fraction_to_milliseconds(str_milli): if not not str_milli: return 0 ... ...**

Original	**def centiseconds_to_milliseconds(centi):
	return int(centi) * 10 if centi else 0**

Fault Injection	**def centiseconds_to_milliseconds(centi):
	return int(centi) * 10 if 0 else centi**

Original	**def isAllowedFilename(filename):
	return isVideoFilename(filename) or isAudioFilename(filename) or isTorrentFilename(filename)**

Fault Injection	**def isAllowedFilename(filename):
	return isVideoFilename(filename) and isAudioFilename(filename) or isTorrentFilename(filename)**

Original	**def isAllowedFilename(filename):
	return isVideoFilename(filename) or isAudioFilename(filename) or isTorrentFilename(filename)**

Fault Injection	**def isAllowedFilename(filename):
	return isVideoFilename(filename) and isAudioFilename(filename) and isTorrentFilename(filename)**

Original	**def can_view_activity(user, viewing_user): return (user == viewing_user or viewing_user.is_superuser or user.created_by == viewing_user)**
Fault Injection	**def can_view_activity(user, viewing_user): return (user == viewing_user and viewing_user.is_superuser or user.created_by == viewing_user)**

Report:

Upon testing the fault injections it turned out that whichever method we changed the code in would fail while the others would still pass. Injecting those 5 fault injections caused an obvious turn of events in the amount of test cases that passed and the ones that did not pass.