

CEG3136 Lab 3 Report

Group 9:

- Monique Diemert
 - Hongyi Lin
 - [CEG3136 Lab 3 Report](#)
 - [Code Snippet](#)
 - [States and Transitions Coverage](#)
-

Code Snippet

```
void system_update_state(alarm_system_t *system, user_t *logged_in_user){
    static const uint32_t kWaitingTimeSecond = 10;
    static const uint32_t kMsPerSecond = 1000;
    int i;

    switch (system->state){
        case UNARMED:
            if (logged_in_user != NULL && logged_in_user->state == LOGGED_IN) {
                // printf("OK, good if you see this.\n"); // debug
                system->prev_state = system->state;
                system->state = WAITING_TO_ARM;
            }
            break;
        case WAITING_TO_ARM:
            if (logged_in_user != NULL) {
                uint32_t logged_in_time = (logged_in_user->current_timestamp
                                           - logged_in_user->logged_in_timestamp) /
                kMsPerSecond;
                // printf("%d seconds after logged in\n", logged_in_time); // debug
                if (logged_in_user->state == LOGGED_IN) {
                    if (logged_in_time >= kWaitingTimeSecond) {
                        system->prev_state = system->state;
                        system->state = ARMED;
                    }
                } else {
                    // printf("Logged during waiting to arm...\n"); // debug
                    system->prev_state = system->state;
                    system->state = UNARMED;
                }
            }
            break;
        case ARMED:
            if (logged_in_user != NULL) {
                if (logged_in_user->state == LOGGED_OUT) {
                    system->prev_state = system->state;
                    system->state = UNARMED;
                }
            }
    }
}
```

```

    }
}
for (i = 0; i < 64; ++i) {
    if (system->sensor_list[i].state == TRIGGERED) {
        system->prev_state = system->state;
        system->state = ALARMED;
        break;
    }
}
break;
case ALARMED:
    if (logged_in_user != NULL) {
        if (logged_in_user->is_super_user) {
            system->prev_state = system->state;
            system->state = UNARMED;
        }
    }
    // reset the sensors
    for (i = 0; i < 64; ++i) {
        sensorm_reset(&system->sensor_list[i]);
    }
    break;
}
system_fsm_coverage_update (system);
}

```

States and Transitions Coverage

FSM State Coverage:

```

UNARMED 3
WAITING_TO_ARM 3
ARMED 3
ALARMED 1

```

FSM Transition Coverage:

```

UNARMED -> WAITING_TO_ARM 3
UNARMED -> ARMED 0
UNARMED -> ALARMED 0
WAITING_TO_ARM -> UNARMED 0
WAITING_TO_ARM -> ARMED 3
WAITING_TO_ARM -> ALARMED 0
ARMED -> UNARMED 1
ARMED -> WAITING_TO_ARM 0
ARMED -> ALARMED 1
ALARMED -> UNARMED 1
ALARMED -> WAITING_TO_ARM 0
ALARMED -> ARMED 0

```