Diagrams for C-lift project

Herman Berget, Markus Ho-Yen

18. mars 2018

1 Class Diagram

Controller

+current_floor: unsigned int
+direction: elev_motor_direction_t
+state: state_t
+queues: bool**

+startup(Controller_t* ctrl): void
+run(Controller_t* ctrl): void
-wait_at_floor(Controller_t* ctrl): void
-update_floor_indicator(Controller_t *ctrl, int floor): void
-stop_if_reached_a_floor(Controller_t *ctrl): void
-initialize_controlstruct(Controller_t *ctrl, unsigned int current_floor, State_t state): void
+check_stop(Controller_t* ctrl): bool
+activate_stop(Controller_t* ctrl): void
+toggle_direction(Controller_t* ctrl): void
+open_door_if_at_floor(): void
+get_direction_from_current_and_destination_floor(Controller_t* ctrl, int destination_floor): elev_motor_direction_t

Queue

+rotate_queues(Controller_t* ctrl): void
+add_floors_in_queue_from_lights(Controller_t *ctrl): void
+is_queue_empty(const bool queue*, const int size): bool
+is_all_queues_empty(Controller_t *ctrl): bool
+clear_orders(Controller_t* ctrl): void
+add_button_to_queue(Controller_t *ctrl, elev_button_type_t button, unsigned int floor): void
+find_extreme_in_primary(const Controller_t* ctrl): int
+decide_next_elev_movement_from_queue(Controller_t* ctrl): void
+remove_floor_from_queue_if_in_primary_queue(Controller_t *ctrl, int floor): bool

ButtonOperations

+read_buttons_and_light_up_buttons(): void
+reset_button_lights_at_floor(int floor): void
+reset_button_lights_except_stop_light(): void

2 Sequence Diagram

1. Heisen står tom og stille i 2. etasje med døra åpen

2. En person signaliserer fra 1. etasje at hun ønsker transport oppover

3. Når heisen kommer, bestiller hun transport til 3. etasje

4. Scenariet avsluttes når heisen befinner seg i 3. etasje og heisdøra åpnes
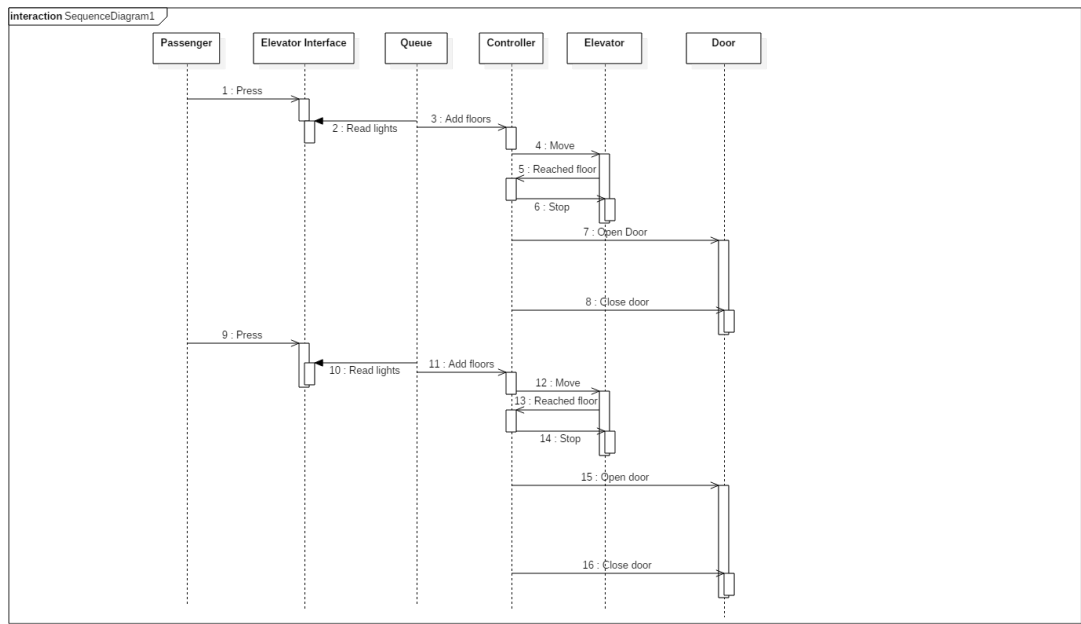

3 Use Case Diagram

Main success scenario with extensions

1. Customer presses up/down-button.

2. System orders elevator to go to customer floor and opens door.

   (a) Elevator is going the opposite direction
       i. Elevator doesn't stop unless someone wants to go down, or someone in the elevator has selected the floor.
       ii. When Elevator is going the same direction as desired floor movement -> 2.

3. Customer goes in the elevator

4. Customer presses a floor button to select a floor.

   (a) Stop-button is pressed
       i. Reset all buttons
       ii. If on floor -> open door
       iii. -> 4.c.ii
   (b) Presses multiple buttons
       i. Add selected floors in queue.
   (c) Presses the same floor button as current floor.
       i. "Close doortimer resets
   (d) Doesn't press button.
       i. Close door after three seconds.
       ii. Wait until button press. If
           A. Up/down -> 1.
           B. Floor button -> 3.

5. System closes door after three seconds

6. Moves in desired direction.

   (a) Presses one or more buttons
       i. Stop-button is pressed -> 4.a.i
       ii. Multiple buttons -> Add selected floors in queue.

7. Elevator stops in next selected floor on the way

   (a) Presses one or more buttons
       i. Stop-button is pressed -> 4.a.i
       ii. Multiple buttons -> Add selected floors in queue.

8. System opens door.

   (a) Presses one or more buttons
       i. Stop-button is pressed -> 4.a.i
       ii. Same floor as current -> 4.c.i
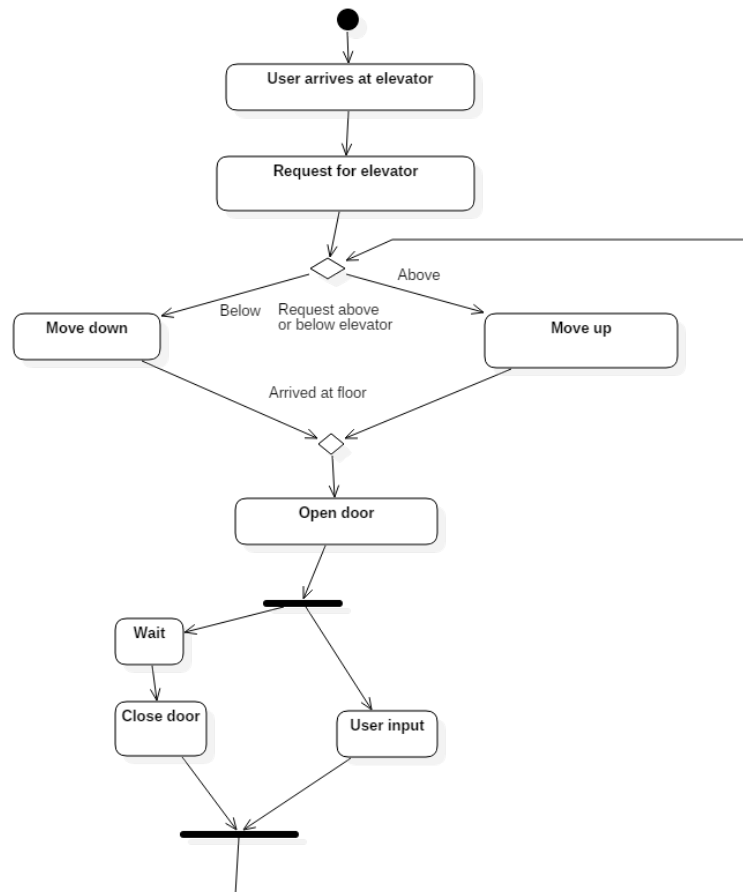   iii. Multiple buttons-> 4.b.i
   (b) If

i. queue empty -> 4.d.i
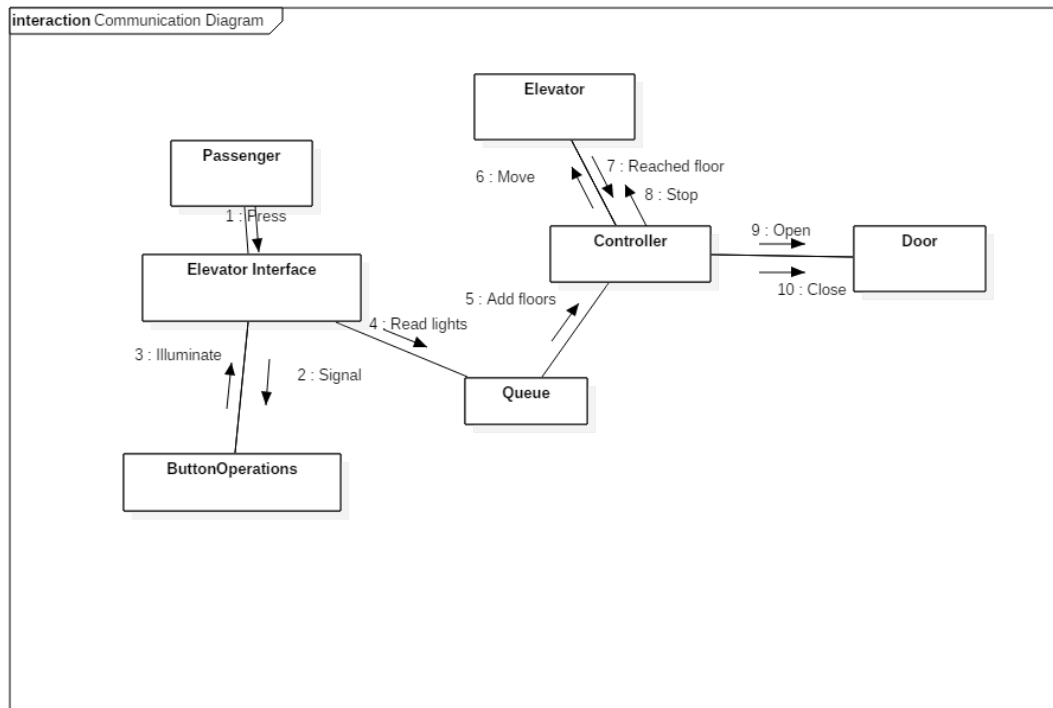ii. Else -> 5.
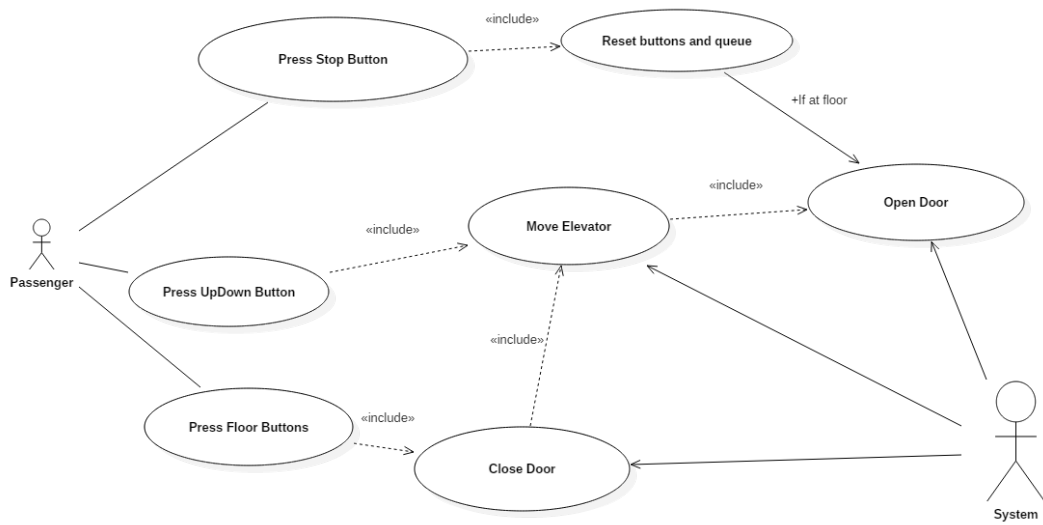
4 State Diagram
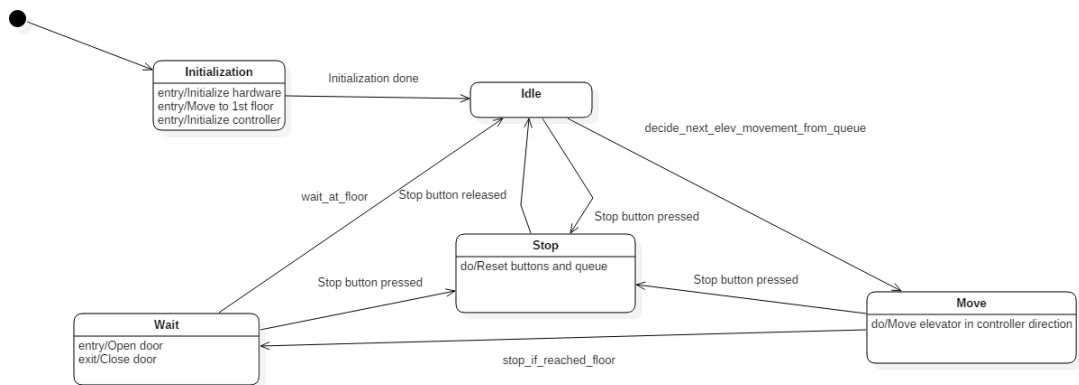
5 Activity Diagram

6 Communication Diagram

7 Timing Diagram