**Assignment** **1**

**Software Specification**

Dajt Mullaj

Tom van Thoor

**Task 1: Requirements**

We first identify the actors with their goals from the informal description:

**Caller:** Calling an elevator from a specific floor.

**Requester:** Request the elevator to reach a specific floor utilizing the cabin panel.

**Site manager:** Handles the emergency situations.

**Engine:** Moves the elevator cabin up or down.

**Floor door:** Protects users from falling into the elevator shaft from the floor.

**Cabin door:** Protects users from falling into the elevator shaft from the cabin.

**Floor sensor:** Detects the elevator cabin reaching a floor.

**Floor request button:** Requests an elevator to reach the floor from a specific floor.

**Panel request button:** Requests the elevator to reach a specific flor from the cabin.

**Emergency button:** Warning signal to be sent to the site manager.

**Floor door sensor:** Indicates whether the floor door is closed or open.

**Cabin door sensor:** Indicates whether the cabin door is closed or open.

Next we identify the inputs and outputs of the system.

* Inputs:

|  |  |  |
| --- | --- | --- |
| ID | Name | Rationale |
| IN000 | inpReachFloor | Receives a signal representing the disjunction of all floor sensors. |
| IN001 | inpCabinDoorOpen | Receives a signal from the cabin door sensor indicating the cabin door is open. |
| IN002 | inpCabinDoorClose | Receives a signal from the cabin door sensor indicating the cabin door is closed. |
| IN003 | inpFloorDoorOpen | Receives a signal from the floor door sensor indicating the floor door is open. |
| IN004 | inpFloorDoorClose | Receives a signal from the floor door sensor indicating the floor door is closed. |
| IN005 | inpRequestUpward | Receives an integer signal from the floor request button upward indicating which floor the elevator must reach. |
| IN006 | inpRequestDownward | Receives an integer signal from the floor request button downward indicating which floor the elevator must reach. |
| IN007 | inpPanel | Receives an integer signal from the panel request button indicating which floor the elevator must reach. |
| IN008 | inpEmergency | Receives a signal from the emergency panel button indicating the elevator must get in its out of order state. |

* Output:

|  |  |  |
| --- | --- | --- |
| ID | Name | Rationale |
| OUT000 | outOpenFloorDoor | Sends signal to open the floor door indicating the floor door to open. |
| OUT001 | outCloseFloorDoor | Sends signal to open the floor door indicating the floor door to close. |
| OUT002 | outOpenCabinDoor | Sends signal to open the cabin door indicating the cabin door to open. |
| OUT003 | outCloseCabinDoor | Sends signal to open the cabin door indicating the cabin door to close. |
| OUT004 | outMoveUp | Sends signal to the engine to move the elevator up. |
| OUT005 | outMoveDown | Sends signal to the engine to move the elevator down. |
| OUT006 | outStop | Sends signal to the engine to stop moving the elevator. |
| OUT007 | outEmergency | Sends signal to the site manager notifying the elevator is in out of order state. |

**Robustness**

We want to model exceptional behavior. For that reason we make the following assumptions:

* The elevator cabin takes 2 seconds to move from one floor to the other.
* The engine is built so that if it receives a outMoveUp signal when the elevator is at the last floor it will send a signal inEndCable to the control unit.
* The engine is built so that if it receives a outMoveDown signal when the elevator is at the ground floor it will send a signal inEndCable to the control unit.

Therefore we add the following input:

|  |  |  |
| --- | --- | --- |
| ID | Name | Rationale |
| IN009 | inEndCable | Receives signal from the engine indicating that there is no more cable to move the elevator cabin up or down. |

**Requirements**

Now that actors, input and outputs are identified we can specify the requirements. The requirements are identified by the capital letter R followed by a positive integer.

**R1**) When the panel request button is pressed the system shall illuminate the button until the desired floor is not reached.

*Rationale*: Shows the actor requester that the control unit registered its call and will send the elevator to the requested floor.

*Priority*: Should have.

**R2**) When the floor request button is pressed the system shall illuminate the button until an elevator traveling in the requested direction does not reach the floor.

*Rationale*: Shows the actor caller that the control unit registered its call and will send an elevator to pick him up.

*Priority*: Should have.

**R3**) When an elevator has no requests to service, the system should send signal outMoveDown to the engine within 500 ms.

*Dependencies:* This requirement depends on the correct implementation or R4.

*Rationale*: A default operation of the control unit will be to ensure that not active elevators are at the ground floor.

*Priority:* Could have.

**R4**) When signal inEndCable is received, the system shall send signal outStop to the engine within 10 ms.

*Rationale*: It ensures that the elevator does not try to move up or down when it reaches the top or ground floors.

*Priority*: Must have.

**R5**) When input inEmergency is received from the emergency button, the system shall send the signal outEmergency to the site manager.

*Rationale*: It ensures that in case of a requested emergency from one of the passengers, the site manager can help.

*Priority*: Must have.

**R6**) When inpPanel is received from the panel request button and the elevator is on a floor below the requested one, the system shall send a outMoveUp signal to the engine within 10 ms.

*Rationale:* The system will send the elevator towards the requested floor.

*Priority:* Must have.

**R7**) When inpPanel is received from the panel request button and the elevator is on a floor above the requested one, the system shall send a outMoveDown signal to the engine within 10 ms.

*Rationale:* The system will send the elevator towards the requested floor.

*Priority:* Must have.

**R8**) When inpPanel is received from the panel request button and the elevator is on the same floor as the requested one, the system shall send signals outOpenCabinDoor and outOpenFloorDoor to the cabin door and the floor door.

*Rationale:* The system must directly open the doors if the requested floor is the same as the one where the elevator is.

*Priority*: Should have.

**R9**) When inpRequestUpward is received from the floor request button and an elevator is idle on that floor, the system shall send signals outOpenCabinDoor and outOpenFloorDoor to the cabin door and the floor door.

*Rationale:* The system must directly open the doors if the requested floor is the same as the one where the elevator is.

*Priority:* Must have.

**R10**) When inpRequestDownward is received from the floor request button and an elevator is idle on that floor, the system shall send signals outOpenCabinDoor and outOpenFloorDoor to the cabin door and the floor door.

*Rationale:* The system must directly open the doors if the requested floor is the same as the one where the elevator is.

*Priority:* Must have.

**R11**) When inpRequestUpward is received from the floor request button and no elevator is idle on that floor, the system shall send signals outMoveUp to the engine of an elevator on a floor below.

*Rationale:* The system will send the elevator towards the requested floor.

*Priority:* Must have.

**R12**) When inpRequestUpward is received from the floor request button and no elevator is idle on that floor and no elevator is on a floor below, the system shall send signals outMoveDown to the engine of an elevator on a floor above.

*Rationale:* The system will send the elevator towards the requested floor.

*Priority:* Must have.

**R13**) When inpRequestDownward is received from the floor request button and no elevator is idle on that floor, the system shall send signals outMoveDown to the engine of an elevator on a floor above.

*Rationale:* The system will send the elevator towards the requested floor.

*Priority:* Must have.

**R14**) When inpRequestDownward is received from the floor request button and no elevator is idle on that floor and no elevator is on a floor above, the system shall send signals outMoveUp to the engine of an elevator on a floor below.

*Rationale:* The system will send the elevator towards the requested floor.

*Priority:* Must have.

**R15**) When a inpReachFloor is received from the floor sensor and the current floor was requested by the control panel, the system shall send signal outStop to the engine within 100 ms.

*Rationale:* The elevator must stop at the requested floor.

*Priority:* Must have.

**R16**) When a inpReachFloor is received from a floor sensor and the current floor was requested by the control panel and the elevator is idle, the system shall send signals outOpenFloorDoor and outOpenCabinDoor to the cabin door and the floor door after 100 ms.

*Dependency:* Depends on the correct execution of R15.

*Rationale:* The elevator must open the doors at the requested floor.

*Priority:* Must have.

**R17**) When a inpReachFloor is received from the floor sensor and the current floor was requested by the floor request button, the system shall send signal outStop to the engine within 100 ms.

*Rationale:* The elevator must stop at the requested floor.

*Priority:* Must have.

**R18**) When a inpReachFloor is received from a floor sensor and the current floor was requested by the floor request button and the elevator is idle, the system shall send signals outOpenFloorDoor and outOpenCabinDoor to the cabin door and the floor door after 100 ms.

*Dependency:* Depends on the correct execution of R17.

*Rationale:* The elevator must open the doors at the requested floor.

*Priority:* Must have.

**R19**) When the inpCabinDoorOpen is received from cabin door sensor, the system shall send a signal outCloseCabinDoor to the cabin door after 5 seconds.

*Rationale:* The elevator must have the cabin doors close to move again.

*Priority:* Must have.

**R20**) When the inpFloorDoorOpen is received from floor door sensor, the system shall send a signal outCloseFloorDoor to the floor door after 5 seconds.

*Rationale:* The elevator must have the floor doors close to move again.

*Priority:* Must have.

**R21**) If outMoveUp or outMoveDown are sent an inpReachFloor is not received within 3 seconds the system shall send signal outEmergency to the site manager.

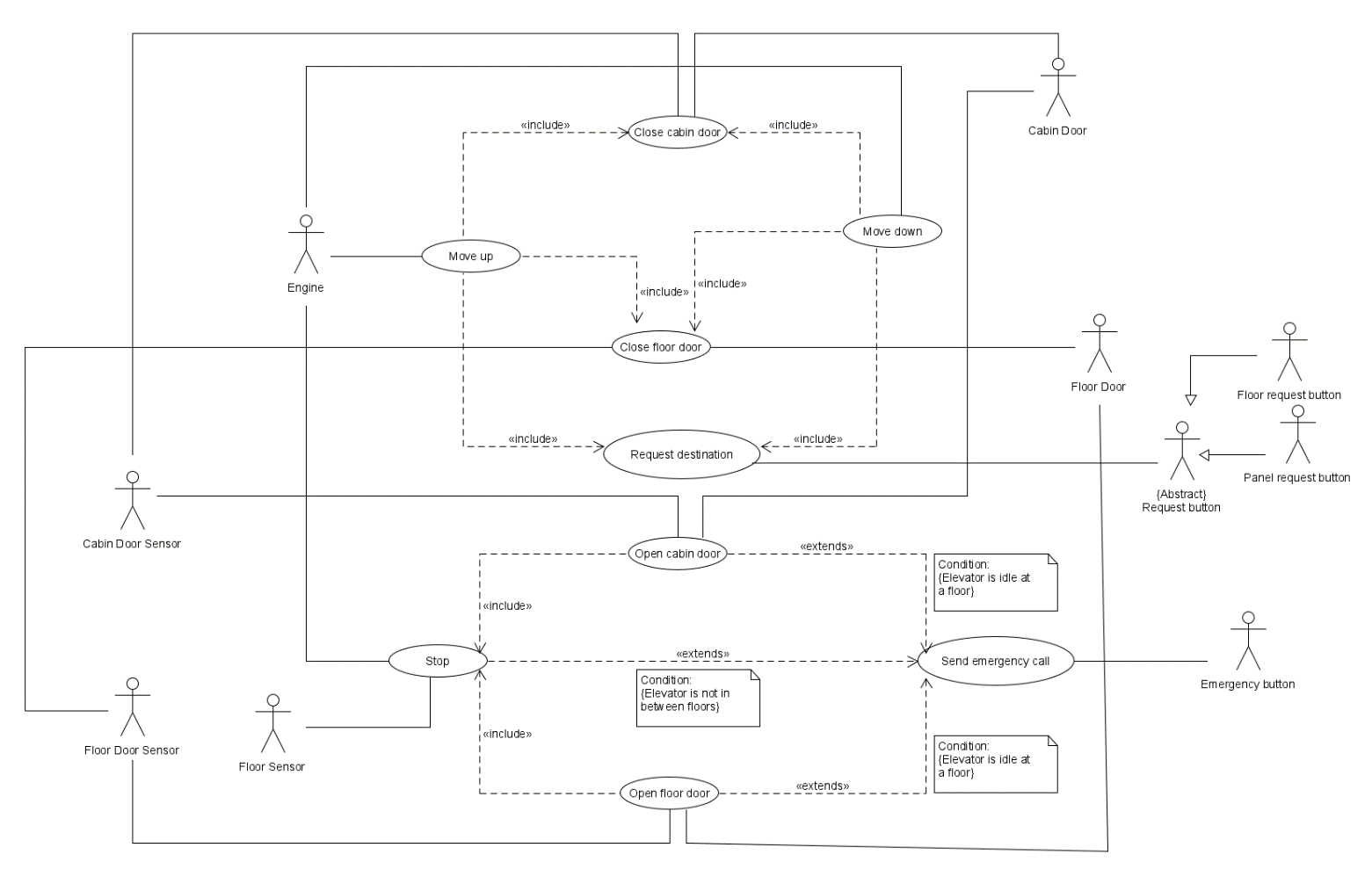
*Rationale:* This requirement ensures that the site manager intervenes if the engine is not starting.

*Priority:* Could have.

**R22**) If inEndCable is received and the elevator is not on the top floor or ground floor, the system shall send signal outEmergency to the site manager.

*Rationale:* If the elevator cannot recognize it is on the top or ground floor, these means that a floor sensor is not working or that for any other reason the control unit cannot recognize when the elevator passes one or more floors. The site manager therefore has to intervene.

*Priority:* Could have

**Task 2: Use Case Diagram**

**Use case description**

We then report the use case description for each use case.

|  |  |
| --- | --- |
| **Name:** | **Move up** |
| Short description: | An elevator cabin moves upwards after the engine receives a signal from the control unit to move up the cabin. |
| Precondition: | The elevator cabin is not moving, the doors are closed and a destination is requested. |
| Postcondition: | The elevator cabin is moving upwards. |
| Error situations: | The engine is not working. |
| System state in the event  of an error: | The elevator cabin does not move. |
| Actors: | Engine. |
| Trigger: | The elevator cabin is requested at a floor different from the one it is currently in. |
| Standard process: | (1) The control unit reads which floor is requested from the panel request button.  (2) The control unit sends a outMoveUp signal to the engine.  (3) The cabin moves upwards. |
| Alternative processes: | (1’) The control unit reads which floor is requested from the floor request button.  (2-3) As above. |

|  |  |
| --- | --- |
| **Name:** | **Move down** |
| Short description: | An elevator cabin moves downwards after the engine receives a signal from the control unit to move down the cabin. |
| Precondition: | The elevator cabin is not moving, the doors are closed and a destination is requested. |
| Postcondition: | The elevator cabin is moving downwards. |
| Error situations: | The engine is not working. |
| System state in the event  of an error: | The elevator cabin does not move. |
| Actors: | Engine. |
| Trigger: | The elevator cabin is requested at a floor different from the one it is currently in. |
| Standard process: | (1) The control unit reads which floor is requested from the panel request button.  (2) The control unit sends a outMoveDown signal to the engine.  (3) The cabin moves upwards. |
| Alternative processes: | (1’) The control unit reads which floor is requested from the floor request button.  (2-3) As above. |

|  |  |
| --- | --- |
| **Name:** | **Stop** |
| Short description: | An elevator cabin stops moving. |
| Precondition: | The elevator cabin is moving. |
| Postcondition: | The elevator cabin is not moving and is still at a specific floor. |
| Error situations: | The elevator cabin does not stop moving. |
| System state in the event  of an error: | The elevator cabin will keep moving until the top or ground floor. |
| Actors: | Engine and Floor sensors. |
| Trigger: | The elevator cabin reaches a desired floor or an emergency signal is sent to the control unit. |
| Standard process: | (1) The control unit receives a signal from the Floor sensor.  (2) The control unit sends a outStop signal to the engine.  (3) The cabin stops. |
| Alternative processes: | (1’) The elevator is in an emergency state.  (2) The control unit sends a outStop signal to the engine.  (3) The cabin stops. |

|  |  |
| --- | --- |
| **Name:** | **Close cabin door** |
| Short description: | The cabin doors are closed. |
| Precondition: | The cabin doors are open. |
| Postcondition: | The cabin doors are closed. |
| Error situations: | The cabin doors do not close. |
| System state in the event  of an error: | The cabin doors are open. |
| Actors: | Cabin door and Cabin door sensor. |
| Trigger: | The cabin doors have been open for 5 seconds or a floor has been requested and the elevator needs to start moving. |
| Standard process: | (1) The control unit sends a outCloseCabinDoor signal to the cabin doors.  (2) The control unit receives a inCabinDoorClose signal from the cabin door sensor. |
| Alternative processes: | (2’) The control unit does not receive a inCabinDoorClose signal from the cabin door sensors.  (3’) After 5 seconds the control unit sends a new outCloseCabinDoor signal to the cabin doors. |

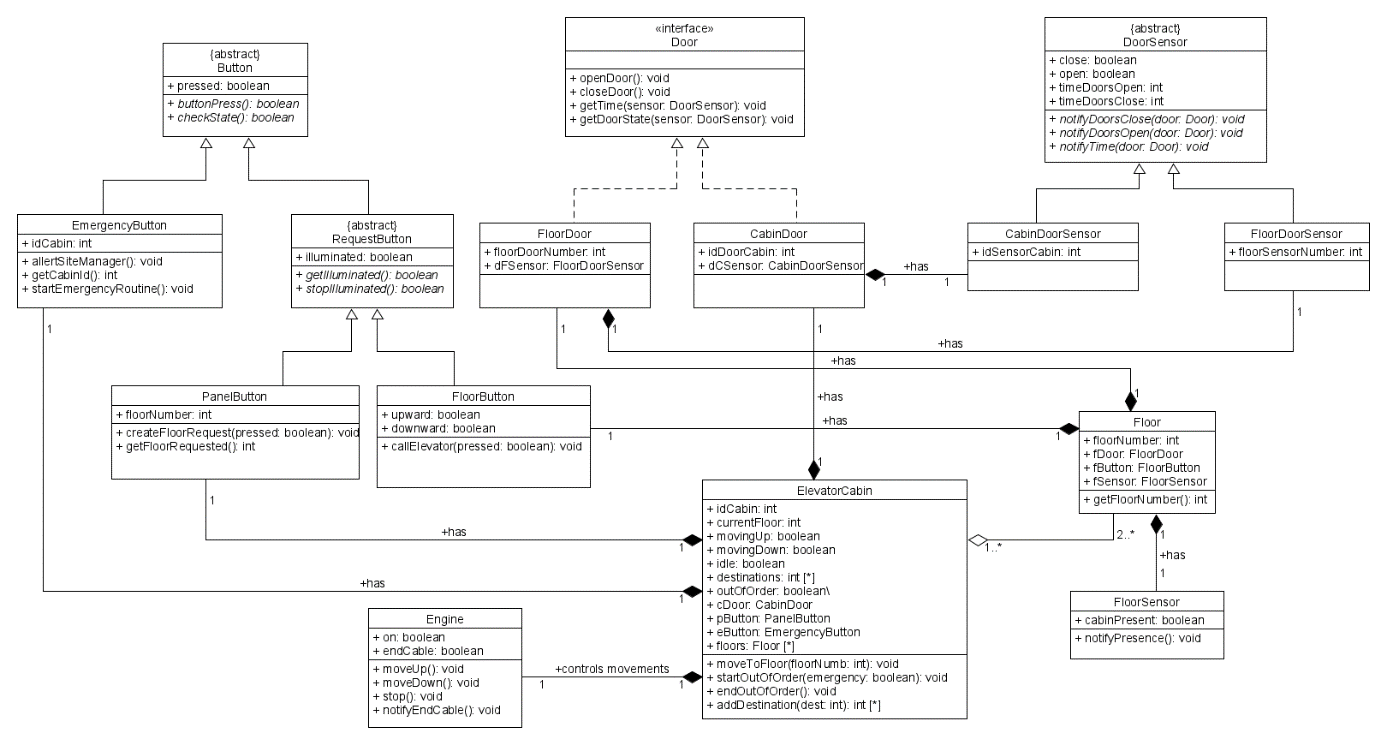
|  |  |
| --- | --- |
| **Name:** | **Close floor door** |
| Short description: | The floor doors are closed. |
| Precondition: | The floor doors are open. |
| Postcondition: | The floor doors are closed. |
| Error situations: | The floor doors do not close. |
| System state in the event  of an error: | The floor doors are open. |
| Actors: | Floor door and Floor door sensor. |
| Trigger: | The floor doors have been open for 5 seconds or a floor has been requested and the elevator needs to start moving. |
| Standard process: | (1) The control unit sends a outCloseFloorDoor signal to the floor doors.  (2) The control unit receives a inFloorDoorClose signal from the floor door sensor. |
| Alternative processes: | (2’) The control unit does not receive a inFloorDoorClose signal from the floor door sensors.  (3’) After 5 seconds the control unit sends a new outFloorCabinDoor signal to the floor doors. |

|  |  |
| --- | --- |
| **Name:** | **Open cabin door** |
| Short description: | The cabin doors are opened.. |
| Precondition: | The cabin doors are close and the elevator is not moving. |
| Postcondition: | The cabin doors are open. |
| Error situations: | The cabin doors do not open. |
| System state in the event  of an error: | The cabin doors are closed. |
| Actors: | Cabin door and Cabin door sensor. |
| Trigger: | The elevator reached a requested floor and stopped moving or an emergency signal is received. |
| Standard process: | (1) The elevator stops moving.  (2) The control unit sends a outOpenCabinDoor signal to the cabin doors.  (3) The control unit receives a inCabinDoorClose signal from the cabin door sensor. |
| Alternative processes: | (1’) An inEmergency signal is received and the elevator stopped moving where there is a floor.  (2-3) As above. |

|  |  |
| --- | --- |
| **Name:** | **Open floor door** |
| Short description: | The floor doors are opened.. |
| Precondition: | The floor doors are close and the elevator is not moving. |
| Postcondition: | The floor doors are open. |
| Error situations: | The floor doors do not open. |
| System state in the event  of an error: | The floor doors are closed. |
| Actors: | Floor door and Floor door sensor. |
| Trigger: | The elevator reached a requested floor and stopped moving or an emergency signal is received. |
| Standard process: | (1) The elevator stops moving.  (2) The control unit sends a outOpenFloorDoor signal to the floor doors.  (3) The control unit receives a inFloorDoorClose signal from the floor door sensor. |
| Alternative processes: | (1’) An inEmergency signal is received and the elevator stopped moving where there is a floor.  (2-3) As above. |

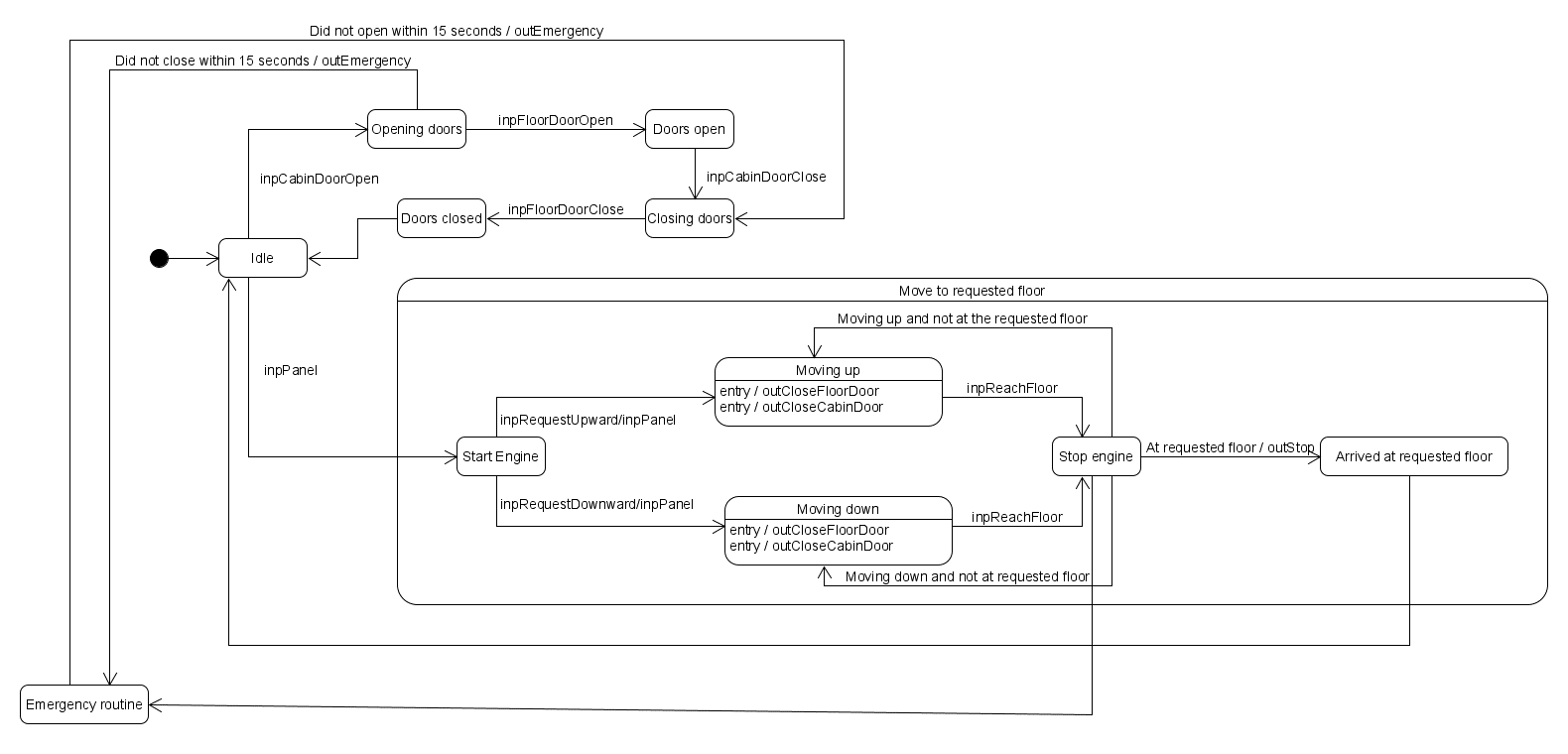
|  |  |
| --- | --- |
| **Name:** | **Request destination** |
| Short description: | A destination is requested for the elevator to reach. |
| Precondition: | The elevator is not out of order. |
| Postcondition: | The elevator is going to arrive at the desired floor. |
| Error situations: | The elevator is not moving. |
| System state in the event  of an error: | The elevator does not arrive at the requested floor. |
| Actors: | The panel request button or the floor request button. |
| Trigger: | A request button is pressed. |
| Standard process: | (1) A caller presses a floor request button.  (2) The control unit receives a signal indicating the floor to reach and which direction the user will travel.  (3) The control unit sends an elevator to the specific floor. |
| Alternative processes: | (1’) A requester presses a panel request button.  (2’) The control unit receives a signal indicating the floor to reach.  (3’) The control unit sends the elevator to the specific floor. |

|  |  |
| --- | --- |
| **Name:** | **Send emergency call** |
| Short description: | The system enters the out of order routine and an emergency signal is sent to the site manager. |
| Precondition: | The elevator is not out of order. |
| Postcondition: | The elevator is out of order. |
| Error situations: | The emergency signal is not sent to the site manager. |
| System state in the event  of an error: | The elevator remains indefinitely out of order. |
| Actors: | Emergency button. |
| Trigger: | The emergency button is pressed. |
| Standard process: | (1) The emergency button is pressed from the control panel.  (2) A out emergency signal is sent to the site manager. |
| Alternative processes: | (1-2) As above.  (3) The control unit send a outStop signal to the engine and the elevator stops moving.  (4) The control unit sends outOpenCabinDoor and outOpenFloorDoor signals to the cabin and floor doors.  (5) The cabin and floor doors open. |

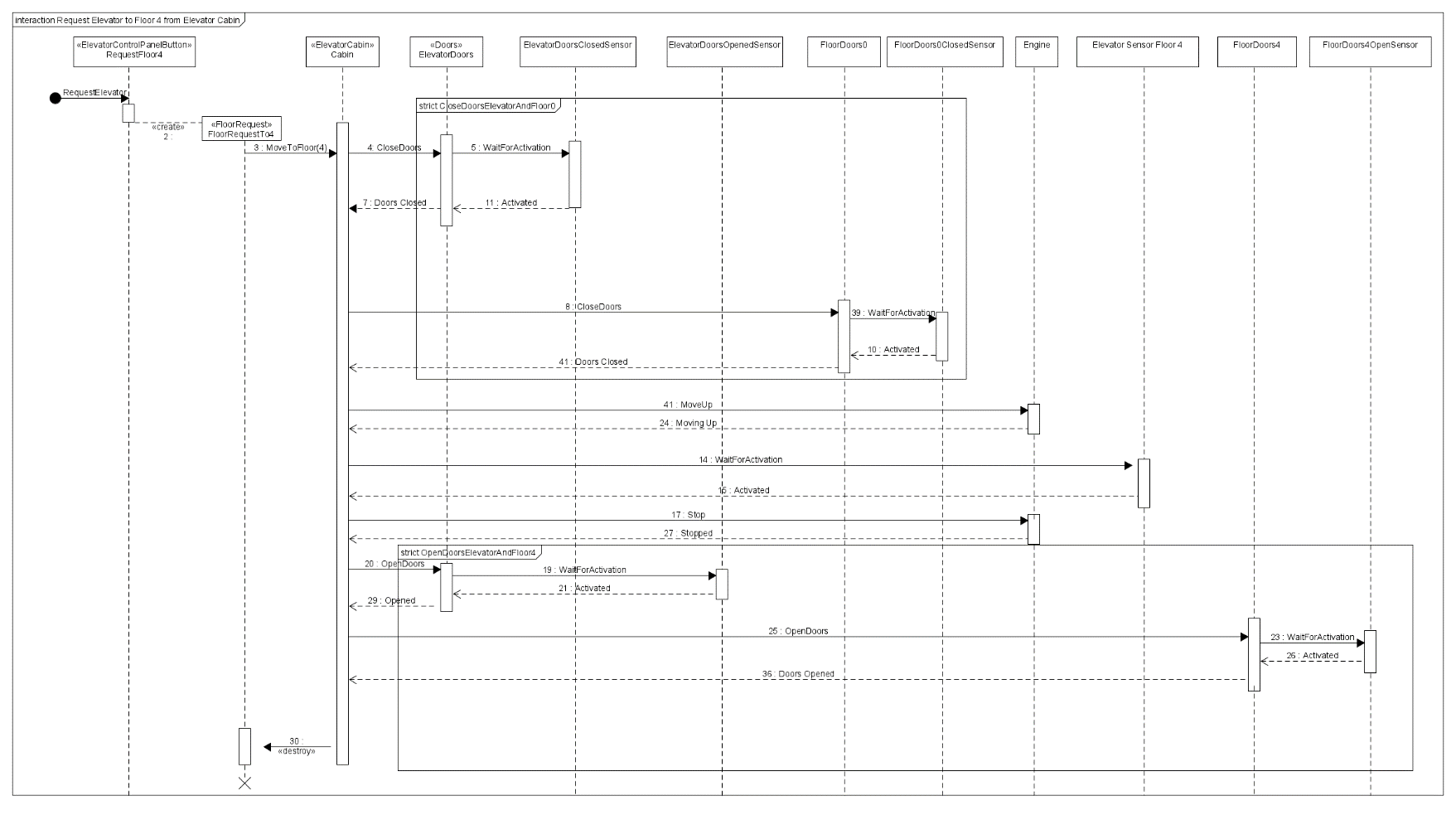
**Task 3: Class Diagram**

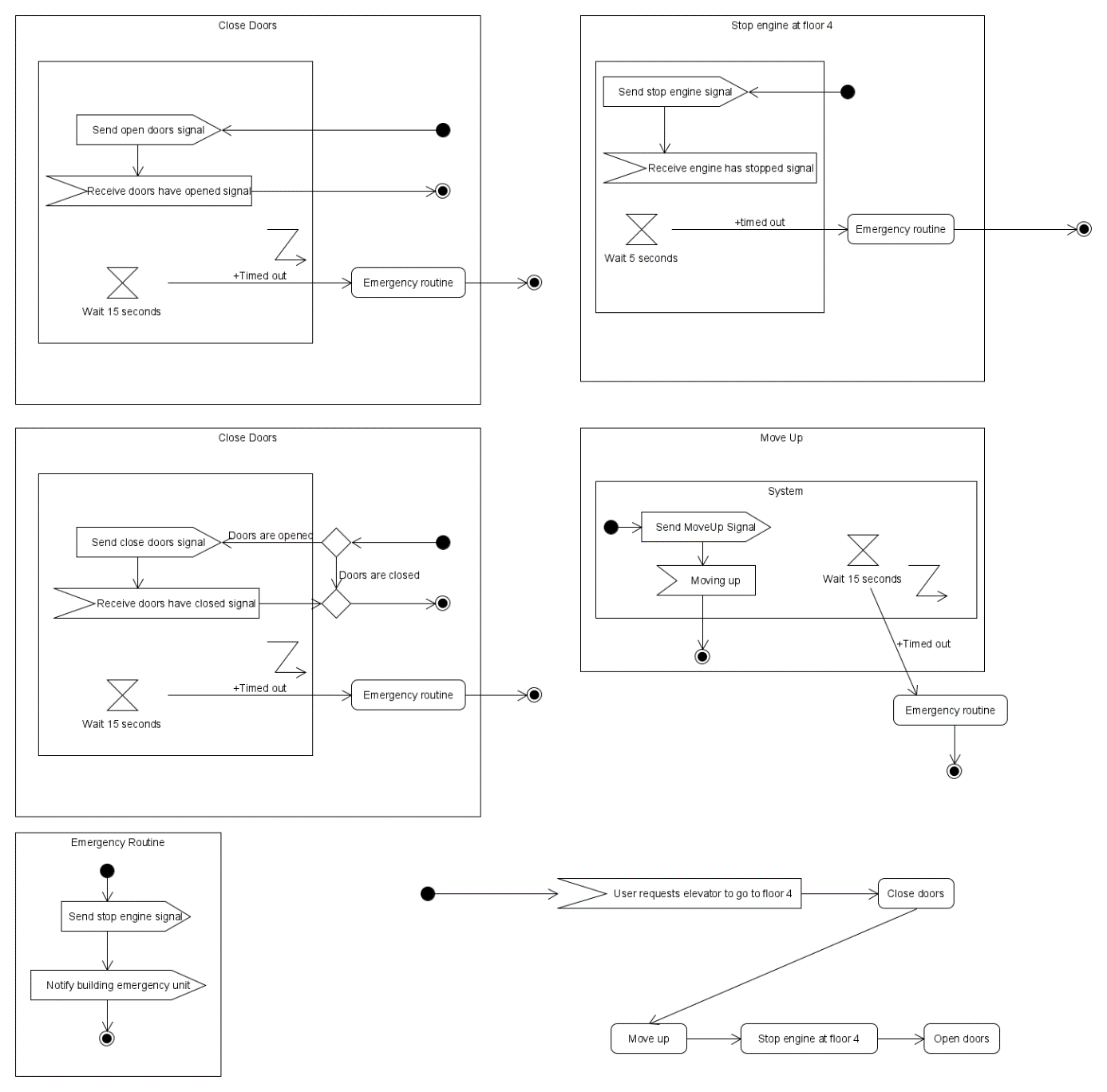
**Brief description:**

We model the system creating different classes of attributes and methods. The classes communicate with each other using methods to notify other classes and push data in other classes when needed. The methods are therefore partially inspired by the Observer design pattern, as some classes represent observers that indeed wait to get notified by sensors or buttons.

**Task 4: State Machine Diagram**

The diagram contains 2 state machines: a top level machine, with one composite state (move to requested floor).

**Task 5: Sequence Diagram**

**Task 6: Activity Diagram**

**Reflection**

We encountered difficulties at the beginning of the project because while writing the requirements we had to pay attention to not include implementation details that should have been included in different parts of the software development project.

Modelling the different diagrams also gave us some troubles as in every diagram a possible new solution to model a certain aspect would have been found and that would have required us to go back and change other models to make the assignment cohesive.

Dajt Mullaj made task 1, 2 and 3.

Tom van Thoor made task 4, 5 and 6.