**GenSec Industries: Occupancy Management Design Document**

  
  
  
  
  
  
  
  
  
  
  
Johnson Domacasse 4471709  
Nedyalko Tenev 4808231  
Tuan Kiet Ho 4832649  
Farros Ramzy 3767353

[Document history 3](#_Toc149725811)

[Terms, Abbreviations 3](#_Toc149725812)

[1. Introduction 4](#_Toc149725813)

[2. System Description 4](#_Toc149725814)

[3. System Design 5](#_Toc149725815)

[3.1. Use Cases 5](#_Toc149725816)

[3.2. User Requirements 7](#_Toc149725817)

[3.3. Functional Requirements 8](#_Toc149725818)

[3.4. Non-Functional Requirements 8](#_Toc149725819)

[4. System Structure 9](#_Toc149725820)

[5. System Behavior 9](#_Toc149725821)

[5.1. Sequence Diagrams 10](#_Toc149725822)

[5.2. state machines 10](#_Toc149725823)

[6. Recommendations & Conclusions 10](#_Toc149725824)

[7. Reflection 10](#_Toc149725825)

[8. References 11](#_Toc149725826)

List of Figures

List of Tables

# Document history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Status | Author | Description |
| 0.1 | 2023-06-09 | Draft | Johnson | Document creation |

# Terms, Abbreviations

|  |  |
| --- | --- |
| Abbreviation | Description |
| UC | Use Case |
| UR | User Requirement |
| FR | Functional Requirement |
| NFR | Non-Functional Requirement |
| UI | User Interface |
| OOS | Out Of Scope |
|  |  |

# Introduction

**(Short description of project)**

The goal of the project is to handle large numbers of people through check-ins, security areas, duty-free areas, and gates. This will ensure the optional amount of security lines are open, and people are routed to the emptiest lines to avoid queues. Our project is intended to organize the current airport system so that it is more structured.

# System Description

The occupancy system in this project prioritizes the gate functionality of most airport currently have, which is to provide an entrance for any aviation passengers to do a check-in to the Security-Check area of the airport before they are waiting their flight inside the Duty-Free area.

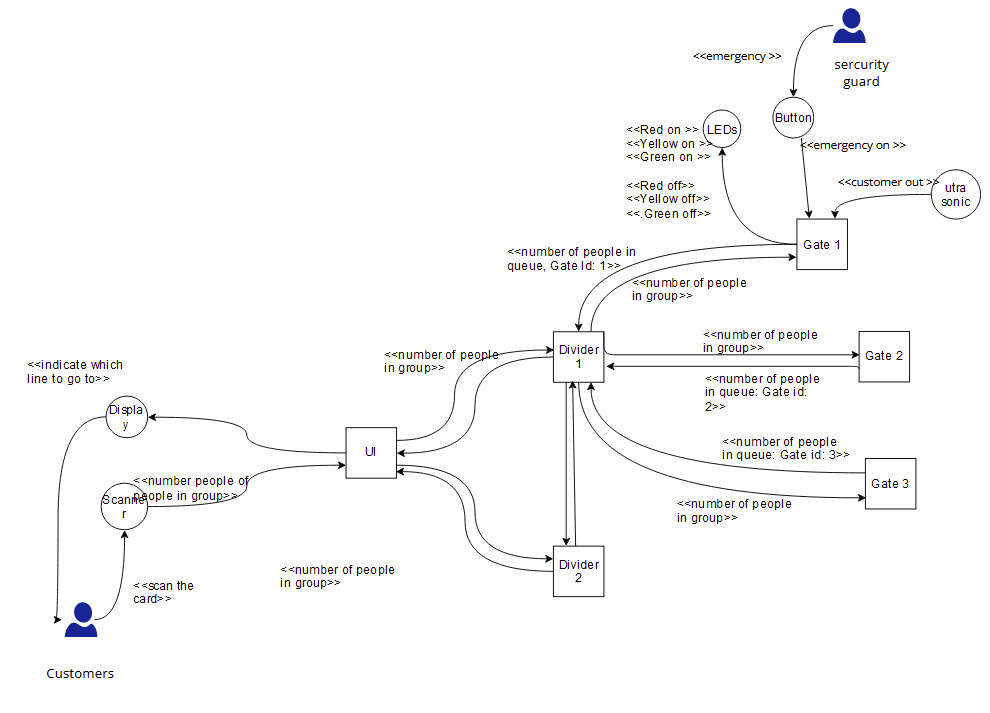
Aside from the gate functionality, the system will also use the decentralized system to keep up the efficiency of each entrance program, and some behavior planning in this design to keep the system secure for both the user and the maintainers.

Mainly, the system will behave as a normal entry and exit gate as from the entrance queue, Security-Check, until the Duty-Free area. But in an emergency, the gate will only act as an exit gate.

On emergency, the system will open the gate from the Security-Check area to let people from that area to go out and find a safe point of the airport (depends on the guide of the airport security guards). And the system will also keep the entrance to the Duty-Free area locked to make sure that there will be no one to trespass from the Security-Check area into the Duty-Free area during emergency.

To offer more insight on the occupancy system, a dashboard-like system will be installed. This allows the user to monitor the behaviour of the system as well.

**(description of our system)**



**(describe the different modes our system has as well (our states))**

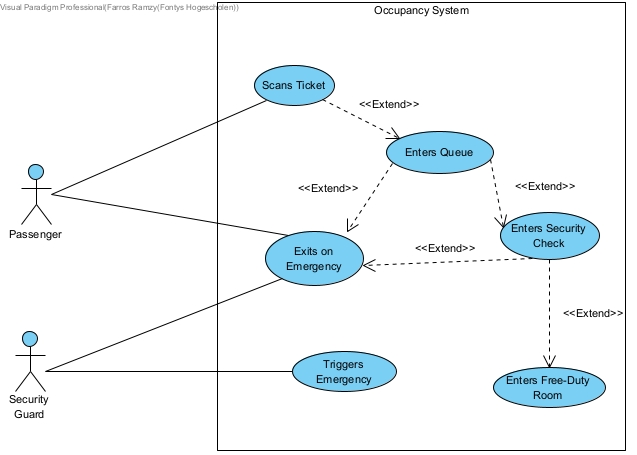
**(table with description of all the states)**

**(safety precautions)**

# System Design

## Use Cases

Below you will find tables that depict the use cases of the occupancy program. It shows the system boundary, specifies how actors interact with the system.



*Figure 1: depiction of the occupancy system in a happy flow.*

|  |  |
| --- | --- |
| **Use Case ID:** UC\_001 | **Use Case:** Choosing entry line |
| **Description:** | An incoming passenger follows the guide on the “Divider” UI display to choose an entry queue which has low or decent traffic flow. |
| **Pre-Condition:** | * Passenger scans the ticket to the scanner. |
| **Post-Condition:** | * The “Divider” UI displays the queue guide. * The passenger follows the guide to go to the correct queue line. |
| **Actor:** | passenger |

|  |  |
| --- | --- |
| **Use Case ID:** UC\_002 | **Use Case:** Entering the Security-Check area |
| **Description:** | A passenger in an entry queue enters the Security-Check area to do a luggage checking procedure before entering the Duty-Free area. |
| **Pre-Condition:** | * The entry gate is closed. * Number of occupancies are not at maximum. * The passenger waits in the queue line before passing the entry gate. |
| **Post-Condition:** | * The entry gate is open. * The passenger enters the Security-Check area. |
| **Actor:** | passenger |

|  |  |
| --- | --- |
| **Use Case ID:** UC\_003 | **Use Case:** Entering the Duty-Free area |
| **Description:** | A passenger in the Security-Check Area enters the Duty-Free area after finishing the luggage checking procedure. |
| **Pre-Condition:** | * The Duty-Free entrance is closed. * The passenger finished the luggage checking procedure. |
| **Post-Condition:** | * The Duty-Free entrance is open. * The passenger exits the Security-Check area and enters the Duty-Free room. |
| **Actor:** | passenger |

|  |  |
| --- | --- |
| **Use Case ID:** UC\_004 | **Use Case:** Exiting the Security-Check area on Emergency |
| **Description:** | An airport security guard triggers the emergency protocol somewhere to inform people in the airport to leave the building. |
| **Pre-Condition:** | * Alarm trigger is off. * There are people inside the Security-Check & Duty-Free areas. * Active Security-Check entries are closed. * Emergency Exits from the Duty-Free area is closed. |
| **Post-Condition:** | * Alarm trigger is on. * Active Security-Check entries are open. * Emergency Exits from the Duty-Free area is open. * People inside the Security-Check area and its queue are moving back to go to the building safe point through the open Security-Check entries. * iPeople inside the Duty-Free room are moving out from the building using the emergency exit inside the duty-free area. |
| **Actor:** | passengers & security guards |

|  |  |
| --- | --- |
| **Use Case ID:** UC\_005 | **Use Case:** Distributing Entry Lines’ Activity |
| **Description:** | If a queue line for an entrance to the Security-Check area is busy, another gate should be active. Therefore, new passengers can go, make a new entrance queue to make the Security-Check procedure efficient. |
| **Pre-Condition:** | * An active queue entry detects that the max queue line is already reached. * A busy signal is being sent to the “Divider”. |
| **Post-Condition:** | * “Divider” activates an inactive gate system if available. * “Divider” tells passengers to go to the line which is not busy to control the queue traffic. * Another security guard do the luggage check duty to the passenger in the new activated gate system. |
| **Actor:** | Passengers & security guards |

## User Requirements

*A user requirement describes what a user expects the system to do. Below you will find the user requirements for the occupancy management system.*

|  |  |  |
| --- | --- | --- |
| **User Requirements ID** | **Description** | **Use Cases ID** |
| **UR\_01** | Passengers should be able to see the UI display on the “Divider” system. | UC\_001  UC\_005 |
| **UR\_02** | Passengers should use the UI to be guided to their security lane for the security check. | UC\_001  UC\_005 |
| **UR\_03** | Passengers and their luggage need to move together throughout the process. | UC\_003 |
| **UR\_04** | Passengers should be able to follow the safety rules of the airport during emergencies. | UC\_004 |
| **UR\_05** | Security guards should have access to activate or deactivate the emergency alarm manually. | UC\_004 |
| **UR\_06** | Security guards should be able to recognize the new open entry gate and let an employee do the security-check procedure for the passengers there. | UC\_005 |

## Functional Requirements

*Functional requirements define what a product must do, what its features and functions are. Below you will find the functional requirements for the occupancy management system.*

|  |  |  |
| --- | --- | --- |
| **Functional Requirements ID** | **Description** | **Use Cases ID** |
| **FR\_01** | The “Divider” system must be able to scan the passenger’s ticket. | UC\_001 |
| **FR\_02** | “Divider” system must be able to display the entry queue for the user. | UC\_001  UC\_005 |
| **FR\_03** | “Divider” system must be able to decide the entry traffic, based on communication between all active gates. | UC\_001  UC\_005 |
| **FR\_04** | Gate system must be able to communicate with each other and to the “Divider” system. | UC\_005 |
| **FR\_05** | Gate system must be able to detect its maximum queue to control the traffic flow of passengers. | UC\_001  UC\_005 |
| **FR\_06** | “Divider” system must be able to communicate with another “Divider” system. | UC\_005 |
| **FR\_07** | The occupancy management system’s data must be transferred to the database so a maintainer can download it to check the system if needed. It can also serve to display on a UI to be seen by an employee high in command. (OOS) | UC\_002  UC\_003 |
| **FR\_08** | The system should have access to read the alarm trigger of the airport. (OOS) | UC\_004 |
| **FR\_09** | The alarm trigger should be accessible so that a security guard can trigger an emergency case. | UC\_004 |

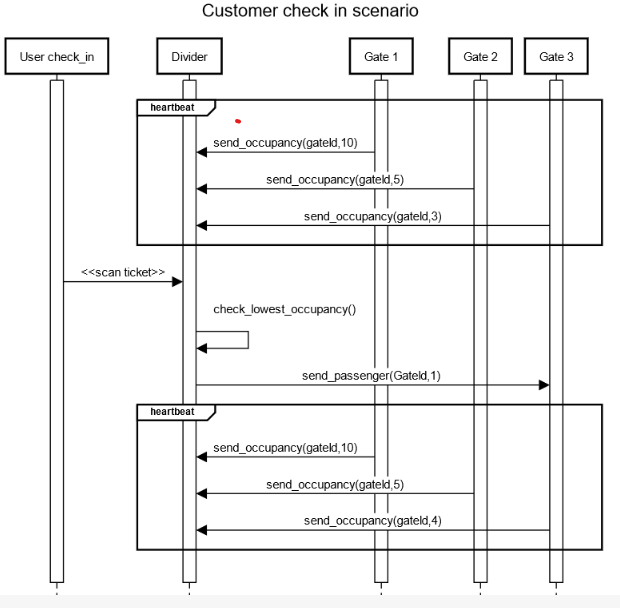
## Non-Functional Requirements

*Non-functional requirements describe the general properties of a system. They are also known as quality attributes. Below you will find the non-functional requirements for the occupancy management system.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Non-Functional Requirements ID** | **Name** | **Description** | **Use Cases ID** |
| **NFR\_01** | Performance | Response time of communication between the Gate Systems to the “Divider” system and each other must be quick. | UC\_001  UC\_002  UC\_003 |
| **NFR\_02** | Security | The security-check gate must be closed until a ticket is scanned. The gate will then open to let passengers through and close after. The gate should be a one-way gate (unless emergency case is triggered) | UC\_002  UC\_003  UC\_004 |
| **NFR\_03** | Reliability | If one gate disconnected, the system should communicate to decide a solution (either opens a new gate, or divide active lines equally) | UC\_001  UC\_005 |
| **NFR\_04** | Maintainability | The system must notify the user if a malfunction happens, and the system must be extendable and upgradable for future possibilities. | UC\_001  UC\_002  UC\_005 |
| **NFR\_05** | Error Handling | The system must display a clear and informative message of error at a malfunction in case of access denial. | UC\_001  UC\_002  UC\_005 |

## Concrete scenario

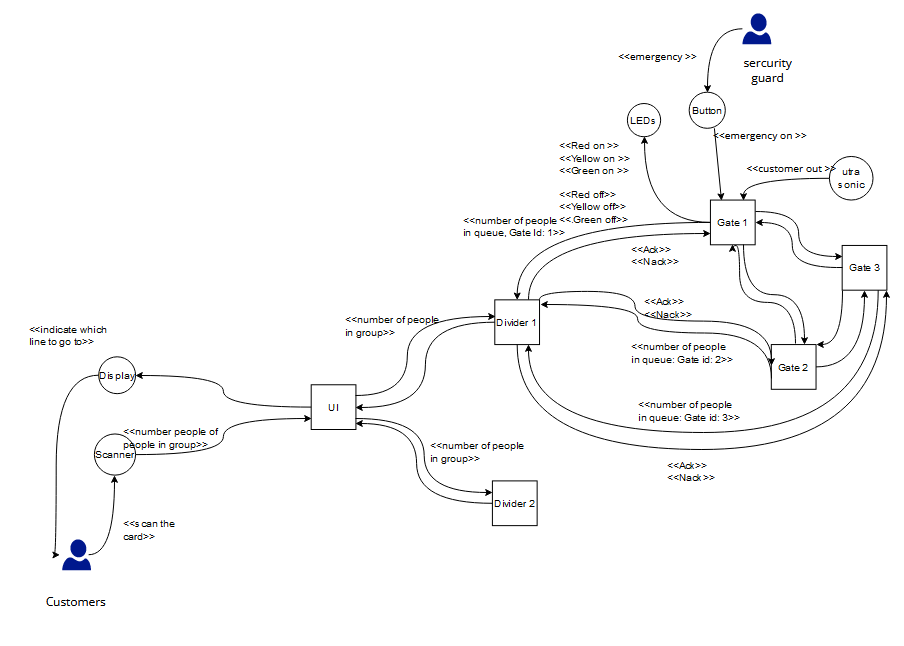
The following sequence diagram is shown to give a better understanding of how the system works. This scenario makes use of UC\_002 to increase the occupancy in a specific line. On top of this, we show how we plan to use the heartbeat method to keep track of the “alive” nodes.



# System Structure

The following components have been identified as parts of the system. The components interact with each other and together they provide the functionality of the system according to the requirements.

**(table with component name and responsibility)**



# System Behavior

The system is expected to behave in a way that is easy to use both for customers and by security personnel. First, a customer is expected to scan his boarding card on the scanner connected to the UI system. This then decides where he should be appointed to – every divider is responsible for 3 gates. After the customer goes to their appointed gate, they pass through security. On their way to the exit, they pass an ultrasonic sensor which makes sure that their exit through the system is successfully counted and they are cleared to go.

In case of an emergency, the security personnel have access to a button on the gate, which will enable an emergency mode. This makes sure that all the gates are open, and people are routed to the nearest exit.

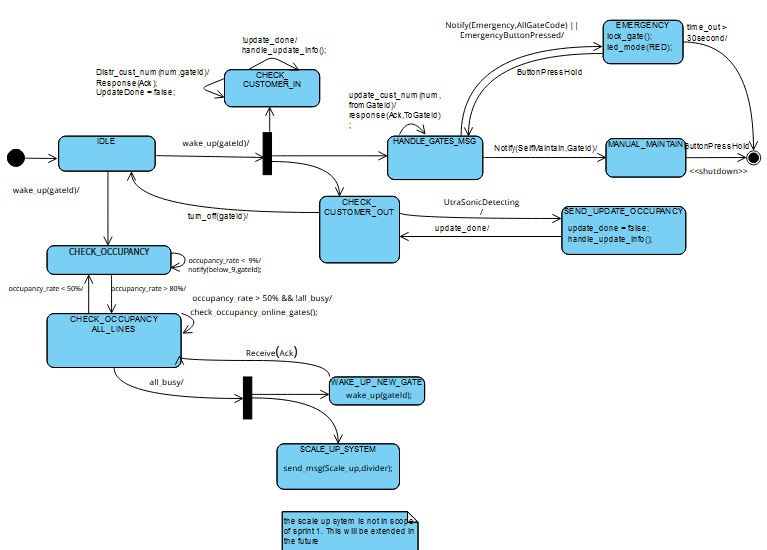
## Sequence Diagrams

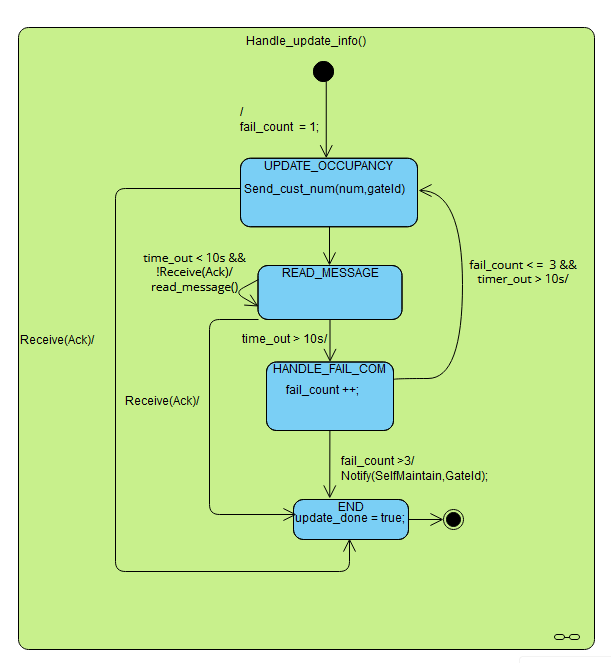
*In this section different scenarios are considered and the corresponding interactions between the involved components are described by sequence diagrams. Some sequence diagrams show how an actor (for instance the user) interacts with the system (by sending or receiving messages).*

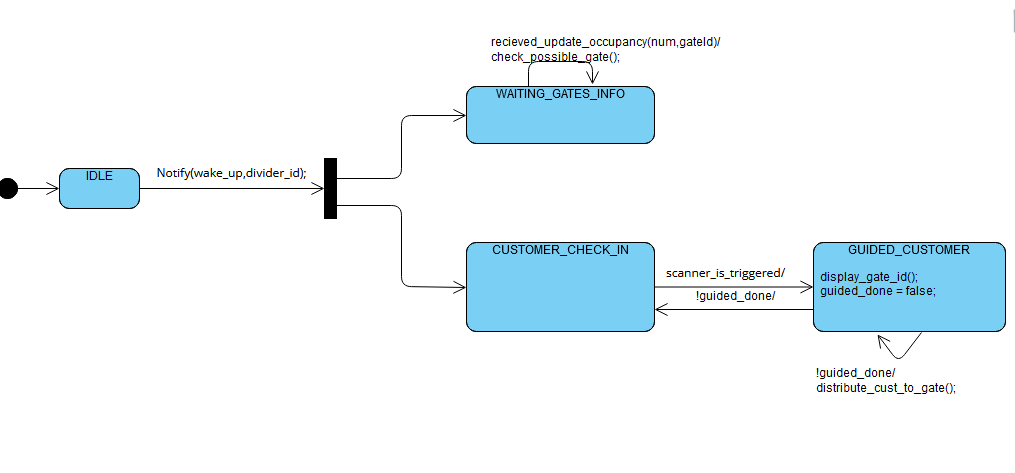
**(in this section, we must decide on some scenarios depending on which use case we choose, and build a sequence diagram around it)**

## state machines

In this section the behavior of the previously identified system components is described by means of state machines. The components exchange messages with each other, one component sends the message (which we call “out message”) and the target component receives the message (which we call “in message”). Messages can contain parameters that carry the data that are exchanged between different components.







**(In this section, we must include the message tables we have of our different components along with the corresponding state machine)**

# Recommendations & Conclusions

(**In this section, we must give the conclusion we were able to come to in this project. We must also give recommendations for our system if it is to be further developed (mention the extra dividers)**

# Reflection

**(In this section, each groupmate will give their reflection on how they think the project started, progressed, and concluded)**

# References

**(In this section, all the references we have used for research must be placed. If applicable, annotations can be made referencing a piece of text to this reference.)**