

# Indledning

Dette dokument præsenterer en detaljeret rapport om udviklingen af et baggagesorteringssystem for en lufthavn.

baggagesorteringssystemet er designet til at håndtere den komplekse proces, som sker dagligt på en lufthavn. Det hjælper med at tjekke passagerne ind og registrere, sortere og håndtere deres bagage, samt transport af bagage til de korrekte gater.

Det er implementeret i en c# console app og integrerer med en MySQL database for at håndtere om fly, passagerer og flights. Formålet er at gøre det både lettere og hurtigere at sortere og transportere bagage og at forbedre den overordnede brugeroplevelse i lufthavnen.

## Problemformulering

Trafikken på lufthavnen er som bekendt steget kraftigt, så vi har været nødt til at gøre noget ved bagagesorteringen.

## Kravspecifikation

### Funktionelle Krav:

- Systemet skal kunne registrere fly og tildele terminaler og gates
- Systemet skal kunne håndtere check-in af passagerer og deres bagage
- Systemet skal kunne sortere bagage til den korrekte terminal og de korrekte gates
- Systemet skal kunne håndtere boarding af passagerer og lastning af bagage på fly
- Systemet skal logge alle hændelser og generere statusrapporter

### Ikke- funktionelle Krav:

- Systemet skal være skalerbart for at håndtere en stor mængde data og brugere
- Systemet skal være pålideligt med minimal nedetid
- Systemet skal have en brugergrænseflade, der er let at bruge og forstå. Systemet skal være sikre og beskyttede mod uautoriseret adgang

# Hændelsestabel

Hændelse	Brugerhandling	Systemrespons
Check-in af passager	Passageren scanner deres pas ved FrontDesk	Systemet registrerer passagerens detaljer og genererer en boarding pass
Check-in af bagage	Passageren afleverer deres bagage	Systemet registrerer bagagen, genererer en QrCode og sender bagagen til sorteringssystemet
Bagagesortering – til terminal	Bagage modtages i sorteringssystemet	Systemet sorterer bagagen til den korrekte terminal
Bagagesortering – til gates	Bagage modtages i sorteringToGateSystemet	Systemet sorterer bagagen til den korrekte gate
Boarding af passager	Passageren scanner deres boarding pass ved gaten	Systemet registrerer passageren som boarded og opdaterer frontDesk/gate status
Lastning af bagage	Bagage transporteres til flyet	Systemet registrerer bagagen som lastet på flyet og opdaterer flyet/opdatererfrontDesk/gate status

## Use cases og specifikationer

### Use Case 1: Check-in Passagerer

Primære Aktør: Passager, FrontDesk medarbejder

**Interessenter og deres interesser:**

- Passger: Vil have deres bagage checket ind og modtage en boarding pass
- FrontDesk Medarbejder: Skal sikre at alle passagerer og bagage er korrekt registreret

**Forudsætninger:**

- Passageren har en gyldig billet – er i passagerer list af flight
- Flight har fået en FrontDesk og en Gate, Check-in skranken er åben

### Efterfølgende betingelser:

- Passageren har modtaget en boarding pass
- Bagagen er registeret, har fået en QRCode og er klar til sortering

### Normal Flow:

- Passageren check-in ved FrontDesk med sit pas, får en boarding pass, check bagagen ind
- System verificerer oplysninger og generer en boarding pass
- Passageren afleverer deres bagage ved skranken
- Systemet registrerer bagagen og genererer en QRCode
- Bagagen sendes til SorteringSystemet

### Undtagelser:

- Hvis passagerens oplysninger er ugyldige, afvises check-in

## Funtionslister

- **AirportManager**
  - o StartAirport()
- **FlightsManager**
  - o GetUniqueFlightDates()
  - o AssignFrontDeskAndGateToFlight(List<Flight> flightsOfTheDay)
  - o PrintFlightsInformation(List<Flight> flights)
- **Frontdesk**
  - o CheckInPassengersAndLuggage()
  - o GetPassengersOnFlight(Flight flight)
  - o CheckInPassenger(Passenger p)
  - o CheckInLuggage(Luggage l)
  - o GenerateBoardingPass(Passenger p, Luggage l)
- **SortingSystem**
  - o void SortLuggage(Luggage l)
  - o SendLuggageToSortingFacility()
  - o AddLuggage(List<Luggage> luggageToAdd)
- **SortingToGateSystem**
  - o SortLuggageToGate(ConcurrentQueue<Luggage> outgoingTerminalA, ConcurrentQueue<Luggage> outgoingTerminalB)

- **BoardingManager**
  - BoardPassengers()
  - LoadLuggage()

## Konclusion