CSC675-775, Section 3

Flight Monitor

Team 7

Piero Calenzani (Lead) - pcalenzani@gmail.com
Jimmy Kwan (Back-end) - jkwan3@mail.sfsf.edu
Chinmay Purav (Front-end) - cpurav@mail.sfsu.edu
Jason Wei (Github Master) - zwei@mail.sfsu.edu
Navneet Singh (Back-end) - nsingh4@mail.sfsu.edu
Pratik Singh (Front-end) - kobez2417@gmail.com

Table of Contents

1. Executive Summary

The Airport Flight Monitor is an application that will serve airport organizations with keeping track of flights and passengers. The application helps management with airport employees, terminal gates, and baggage claim as well as keeping track of what any luggage the passengers check-in. This project will be designed with the primary user being airport managers and other employees. Additional implementation can include the concurrent status of given flights for the airport to display to passengers. The project's database will include only the relevant information to an airport with the primary focus being flight scheduling.

This application is designed with smaller airports in mind and will observe all gates from the given airport and provide the necessary information to keep track of incoming and departing flights. Since the airport itself is being kept in mind as the user, terminal gates and baggage claim are important information to display with each flight in the project's scope. Ticketing and pilots will be monitored through this application for each participating flight company in the airport.

In addition, summarized flight info can be readily displayed for passengers to view through any airport screens the airport wishes to hook up. This usability extends to individually checked luggage and its corresponding baggage claim carousel and allows potential passenger users as well as airport personnel to view baggage info for corresponding flights without having access to unnecessary information.

2. Use Cases

Use Case #1: Bob - Visitor, Chris - Front desk employee

Bob is picking up a friend from his flight but needs to know exactly where to go. Bob asked Chris for the flight information of his friend. So, Chris uses the application to check the schedule and corresponding gate of the flight.

Use Case #2: Paul - Airport Luggage Employee, Tom - Passenger

When a flight has arrived, Paul needs to know which belt the bags must be distributed to. Paul opens our application and knows which baggage claim has been assigned for this particular flight. Tom gets off the flight and he is wondering where his luggage is, so he asks the employee for help. Since Paul already checked the baggage claim, he points out the baggage claim to Tom.

Use Case #3: Michael - Passenger, Sam - employee

Michael's flight gets delayed and assigned to a different gate, so Sam updates flight departure times, assigns gate and flight duration by using the application. Then, Sam broadcasts the updated flight information.

Use Case #4: James - Flight Traffic Controller, Brian - Pilot

Brian needs to know which gate is assigned to his current flight and contacts James to know the designated gate. James uses this application to know the available gates and coordinate with Brian.

3. Entities Glossary

Airplane –

Flights need planes which are owned by **companies** and flown by **pilots**.

Airport –

Location where **passengers** board **flights**. **Airplanes** can land or take off from an airport's **gates**.

Baggage claim –

Baggage claim is where **passengers** pick up their **luggage**, after their **flight**.

Company –

Each **flight** is managed through a company. Companies employ and assign the **airplanes** and **pilots** to the **airport**.

Employee –

An employee's duties in the **airport** include check-in **luggage**, helping **passengers** and transfer **luggage** from **flights** to the correct **baggage claim.**

Flights –

Flights are the main entity, which is the main transportation at the **airport** that takes **passengers** to and from locations. They also carry any checked-in **luggage**.

Gate –

Gates are where **airplanes** dock and allow **passengers** to board or deboard the **airplane**.

Luggage –

Luggage is checked in by **passengers** and carried on **airplanes**.

Passenger –

A person who has purchased a **ticket** and is traveling on a **flight**.

Pilot –

Pilots fly the airplanes and need know what terminal gate they need to dock at.

Ticket –

Admission sold by companies for passengers to board flights.

4. Business Rules

1. Flight must be flown by **at least one** pilot, on a **single** airplane.
2. Flights must carry **at least one** passenger.
3. Airports must contain **multiple** gates.
4. Airports must employ **at least one** employee.
5. Passengers can check-in **at most one** luggage per flight.
6. Passengers must order **at least one** ticket.
7. Tickets are sold by **one** company.
8. Pilots are hired by **at least one** company.
9. Airplanes are owned by **one** company.
10. Luggage can be held at **one** baggage claim, within the airport.
11. Gates can dock **at most one** airplane

5. Initial List of Entities and Relationships

Entity: Airport (strong)
Relations: employs, contain, has
Attributes:

- airport_id (key)
- location
- name

Entity: Airplane (strong)
Relations: owned_by, used, docked_at
Attributes:

- aid (key)
- name
- cid (FK)
- tid (FK)

Entity: Flight (strong)
Relations: uses, flown_by, carries, transports
Attributes:

- fid (key)
- aid (FK)
- schedule
- tid (FK)

Entity: Passenger (strong)
Relations: order, checks_in, carried_by
Attributes:

- passenger_id (key)
- name
- fid (FK)

Entity: Ticket (weak)
Relations: bought_by, ordered_by
Attributes:

- tid (key)
- schedule (FK)
- fid (FK)
- passenger_id (FK)

Entity: Employed (weak)
Relations: airport_hires (employee), employed_by (the airport)
Attributes:

- employment_id (key)
- pilot_id (FK)
- cid (FK)

Entity: Gate (weak)
Relations: contained_by, docks
Attributes:

- gid (key)
- name
- airport_id (FK)

Entity: Luggage (weak)
Relations: checked_in_by, held_at, transported_by
Attributes:

- passenger_id (FK)
- bcid (FK)
- fid (FK)

Entity: BaggageClaim (weak)
Relations: (airport) has, holds
Attributes:

- bcid (key)
- name
- airport_id (FK)

Entity: Pilot (strong)
Relations: hired_by, flies
Attributes:

- pilot_id (key)
- fid (FK)
- name

Entity: Company (strong)
Relations: owns, sells, hires
Attributes:

- cid (key)
- name

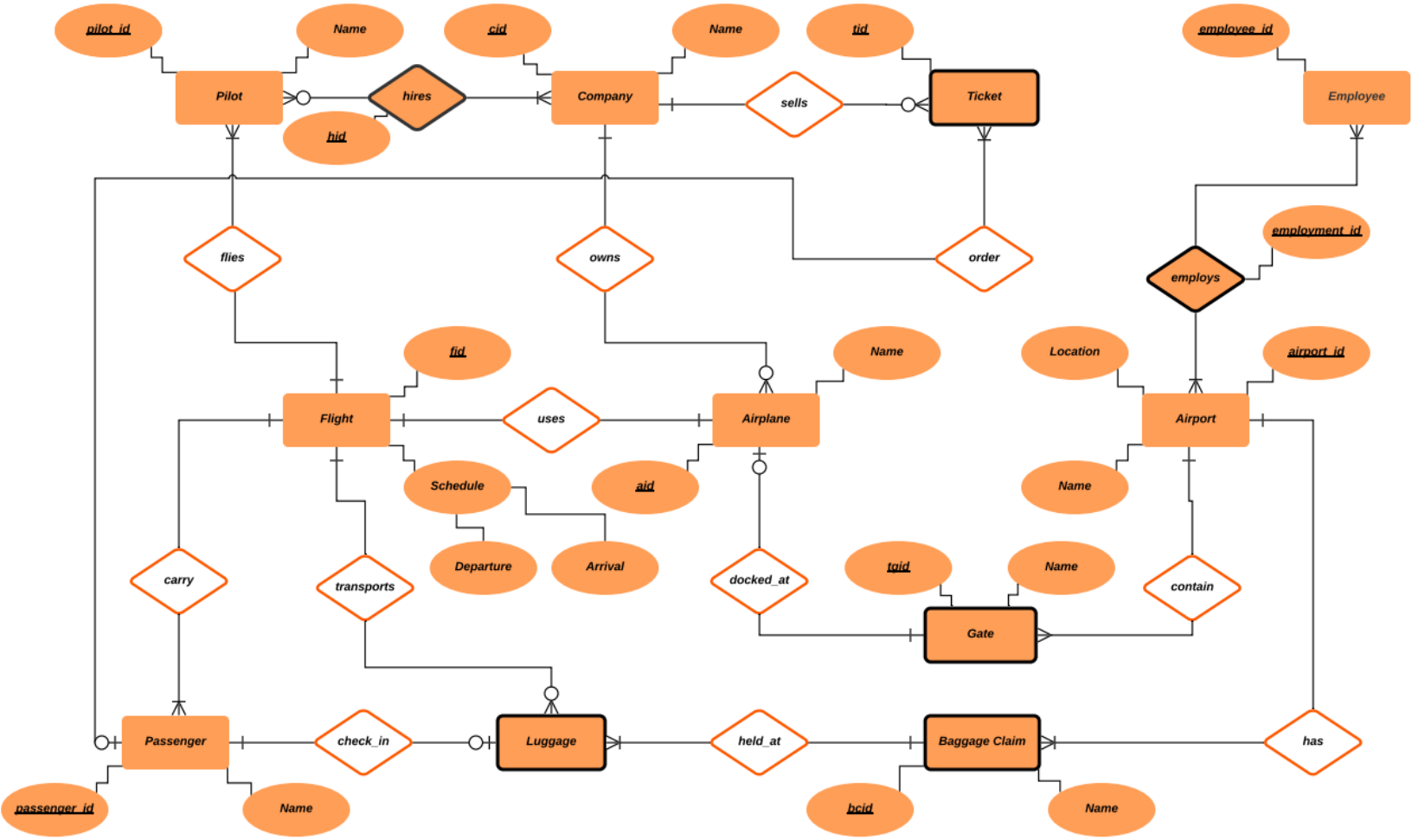
Entity: Company_hired (weak)
Relations: company_employs (pilot), employed_by (the company)
Attributes:

- hire_id (key)
- pilot_id (FK)
- cid (FK)

Entity: Employee (strong)
Relations: employed_by
Attributes:

- employee_id (key)

6. Entity Relationship Diagram



7. ERDs Test

Business rule	Entity1	Relationship	Type Rel	Entity2	Participation Constraint	Pass/Fail	Modify
1 (a)	Flight	Flown By	1:M	Pilot	At Least one pilot must be required	Pass	
1 (b)	Flight	Has	1:1	Airplane		Pass	
2	Flight	carry	1:M	Passenger	At Least one passenger must be required	Pass	
3	Airport	Have	1:M	Gates		Pass	
4	Airport	Must Have	1:M	Employees	At least one Employee must be required	Pass	
5	Passenger	Check in	1:1	Luggage Per Flight	At most one Luggage required	Pass	
6	Passenger	Must Order	1:M	Ticket	At Least one ticket must be ordered	Pass	
7	Tickets	Are Sold	M:1	Company		Pass	
8	Pilots	Are Employed	M:1	Company	At least owned by one company	Pass	
9	Airplane	Owned	M:1	Company		Pass	
10	Luggage	Picked Up	M:1	Baggage Claim		Pass	
11	Gate	Have	1:1	Airplane	At Most one airplane required	Pass	

8. Initial List of Non-functional Requirements

STORAGE

1. Capacity must be able to handle the amount of passenger and flight traffic on any day.
2. Data shall be stored on MySQL database on deployment server.

SECURITY

1. Non-relevant information should be safely secured from public view.

PERFORMANCE

1. Application needs to have quick performance time.
2. Application should be able to handle 50+ users at any time.
3. Application shall be reliable after release.

Capability

1. Application shall be optimized for desktop/phone displays.
2. Application must have good operability for average adult worker.
3. Application must have good maintainability post-release.

Scalability

1. Scalability must be able to account for any given airport size.

9. Team Member Contribution

Work done by each team member. Enumerated on a scale from 1-10.

1. Piero (Team Lead) - 9
    - Wrote executive summary
    - Wrote initial nonfunctional requirements
    - Helped write each use case
    - Helped write and clarify business rules
2. Jason - 10
    - Wrote initial business rules and helped edit them
    - Wrote and fixed entities glossary
    - Helped write and edit use cases
    - Helped edit ERD after version 1 feedback
3. Jimmy - 8
    - Wrote and revised list of entities and attributes
    - Helped edit ERD after version 1 feedback
    - Helped write use cases
4. Chinmay - 8
    - Created and designed initial ERD
    - Helped edit ERD after version 1 feedback
    - Helped write use cases
5. Navneet - 7
    - Late on initial contribution
    - Wrote entities glossary
    - Helped edit use cases and nonfunctional requirements
    - Absent in slack channel (Resolved)
6. Pratik - 8
    - Provided team with sample project features
    - Helped revise executive summary
    - Completed ERD tests