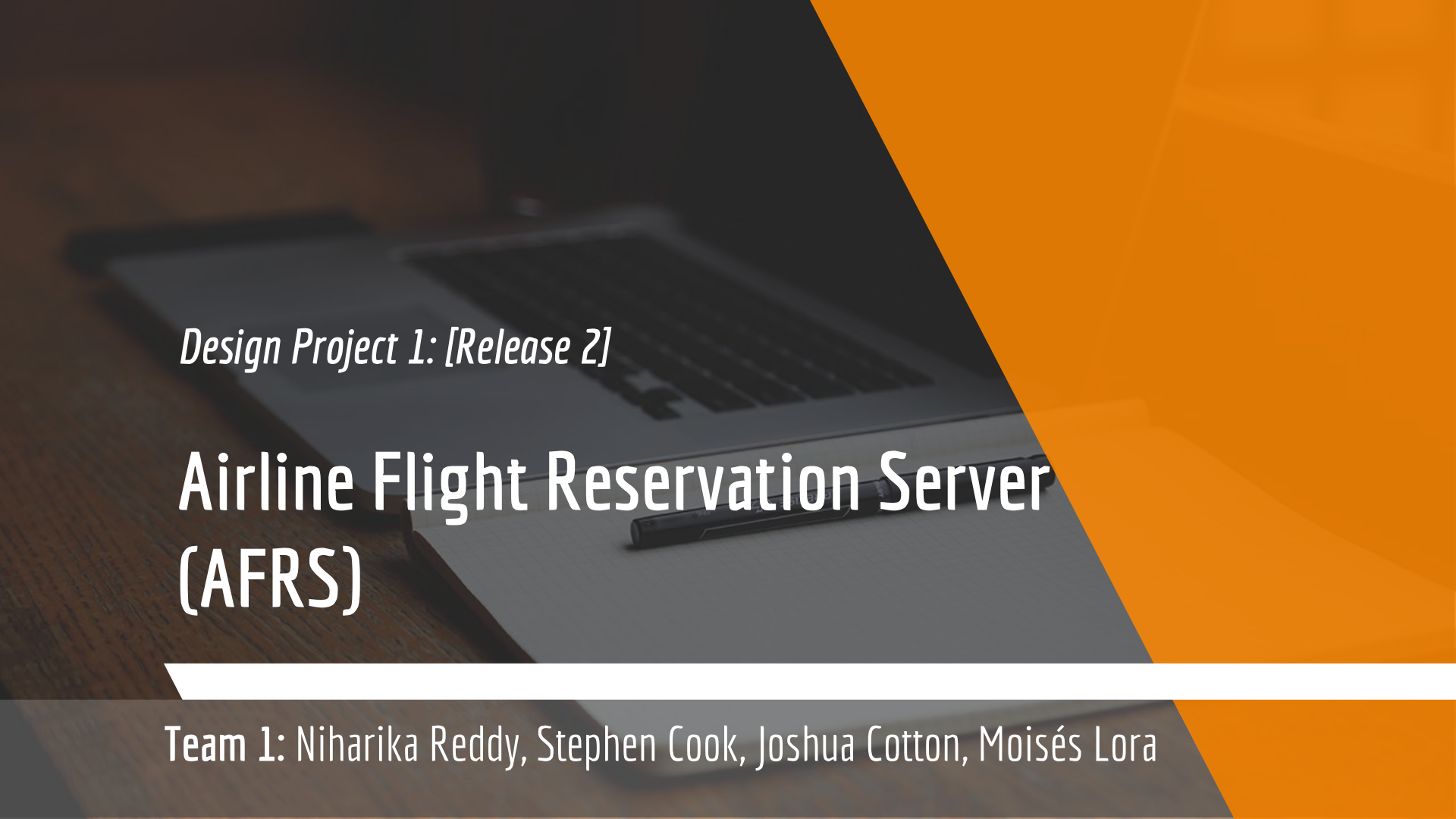


Design Project 1: [Release 2]

Airline Flight Reservation Server (AFRS)

Team 1: Niharika Reddy, Stephen Cook, Joshua Cotton, Moisés Lora

1.
Impacts of R2 Requirements

Impacts of Release 2

High

- Allowing for multiple concurrent client connections.

Medium

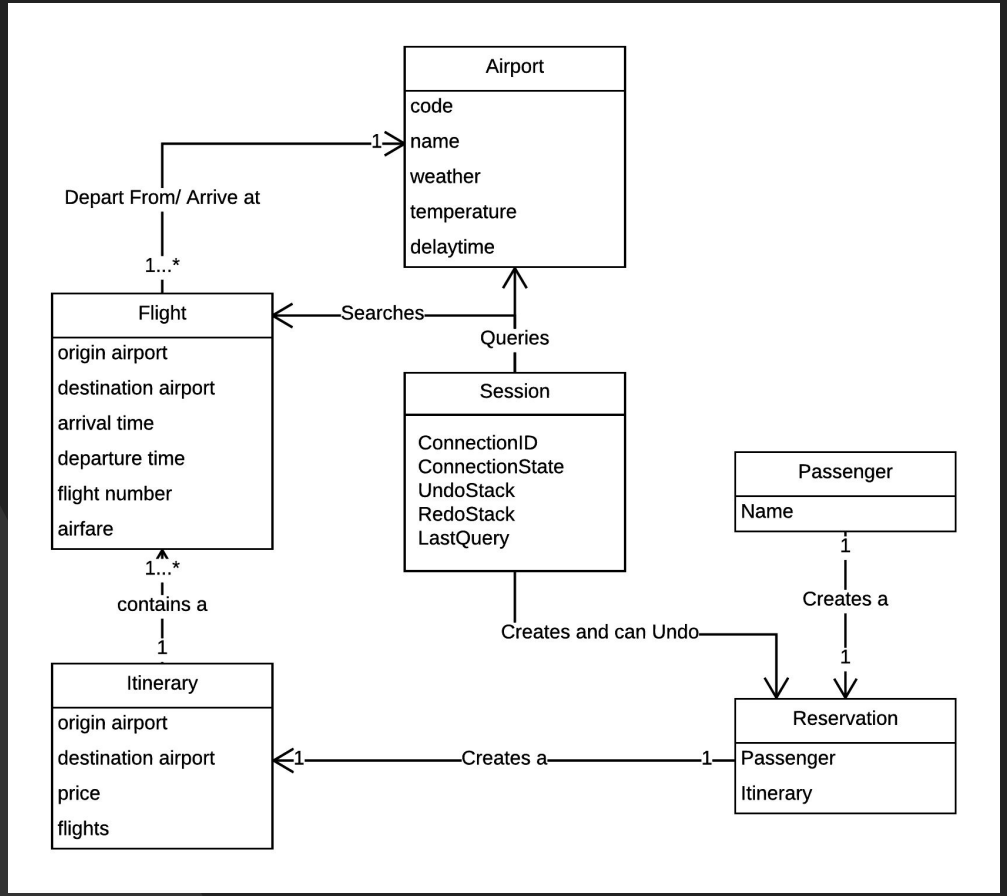
- Undo and Redo operations for reservations
- GUI

Low

- Setting up the FAA server proxies

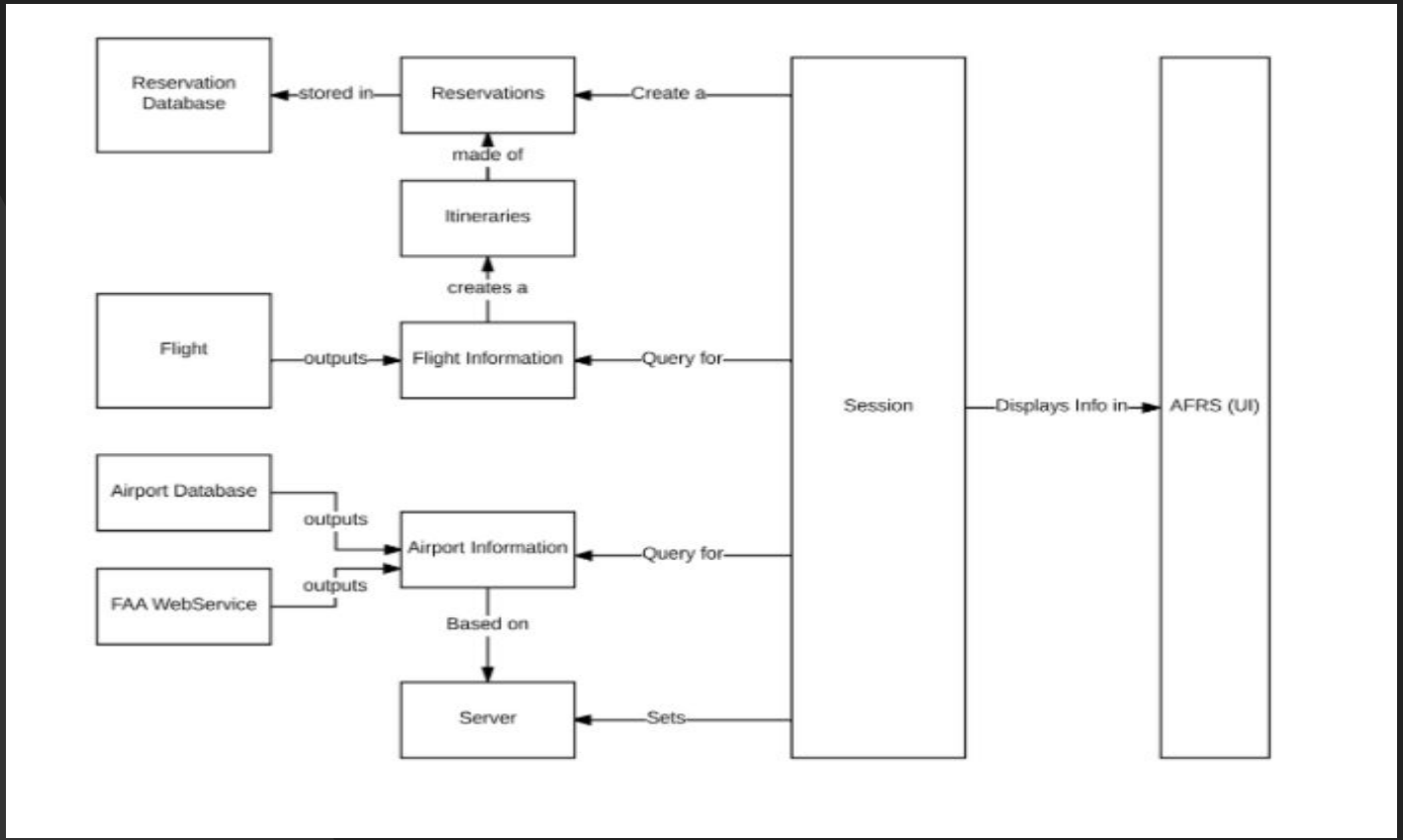
2.
Domain Model

Domain Model



3.
System Architecture

System Architecture

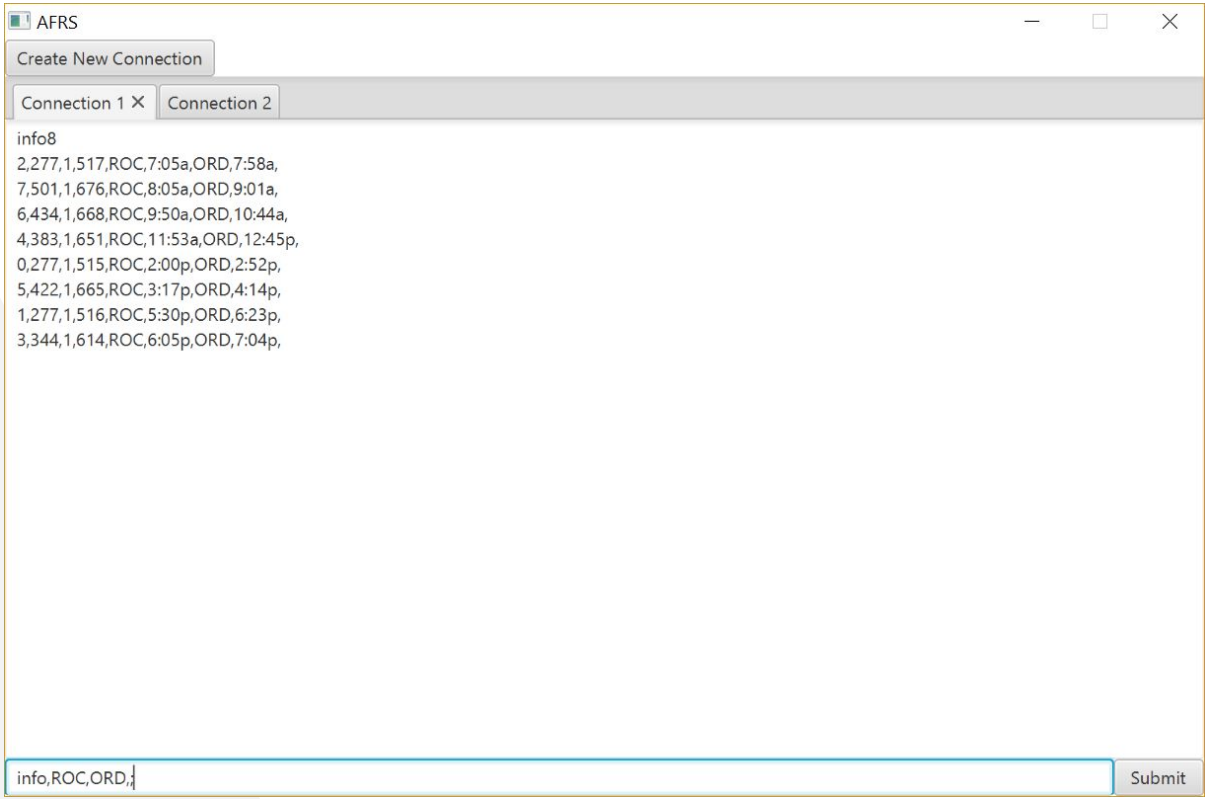


4.
Subsystems

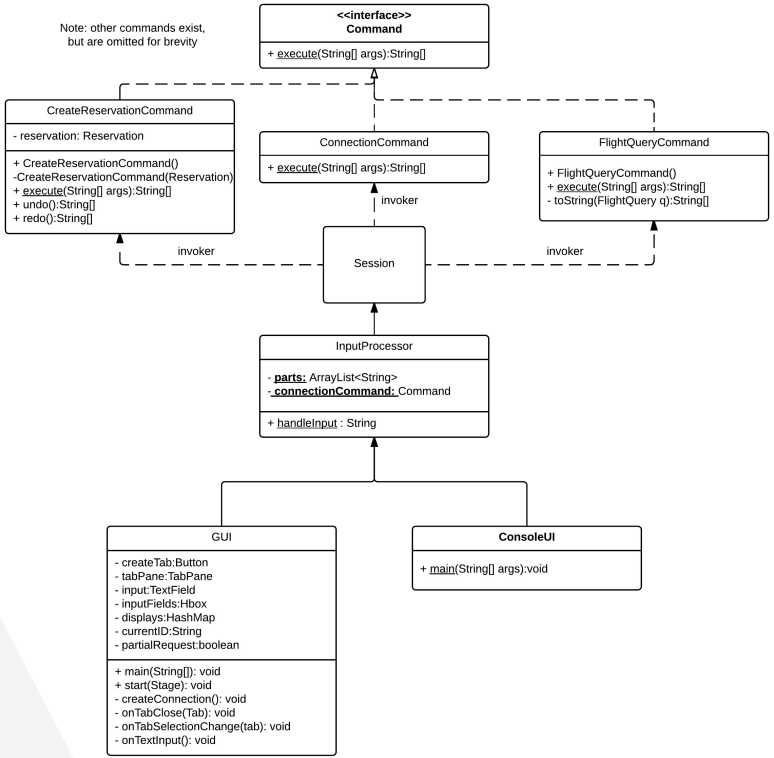
User Interface Subsystem

- Commands
  - Undo/redo
  - Connect/disconnect
  - The Session now represents the invoker
- GUI
- InputParser class

User Interface Subsystem



User Interface Subsystem

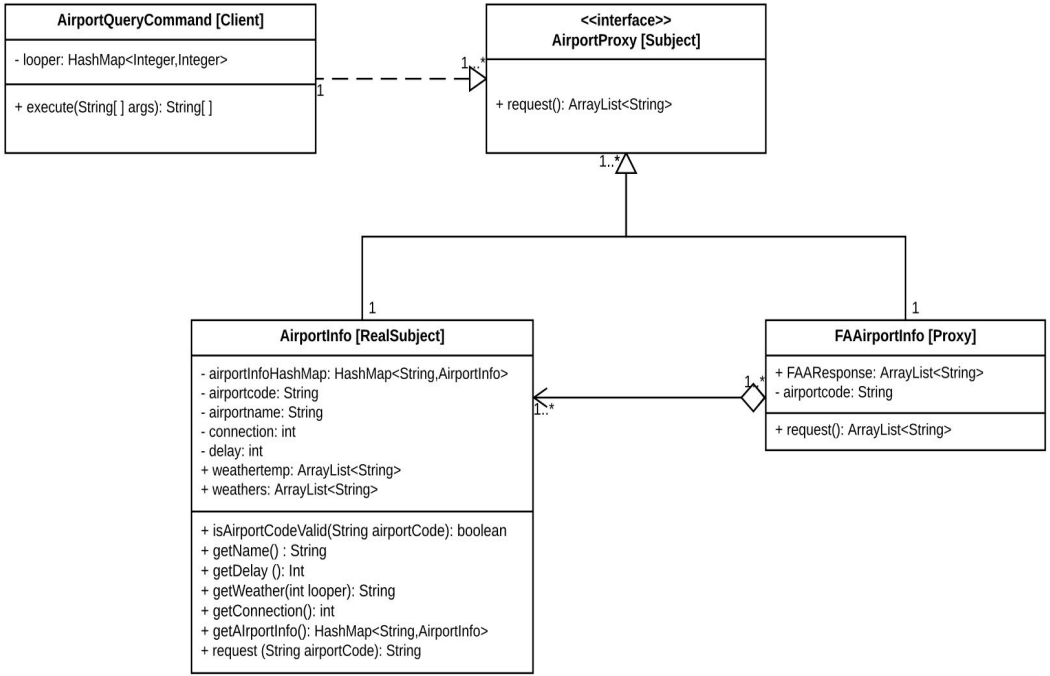


Airport Information Subsystem

Proxy Pattern:

- Remote Proxy pattern was chosen to handle FAA web service requests
- Proxy class (FAAairportInfo) provides a placeholder to provide access to an object
- FAAairportInfo class provides a level of indirection for when the client requests airport information
- Allows for higher security measures, in case the connection is unsuccessful

Airport Information Subsystem

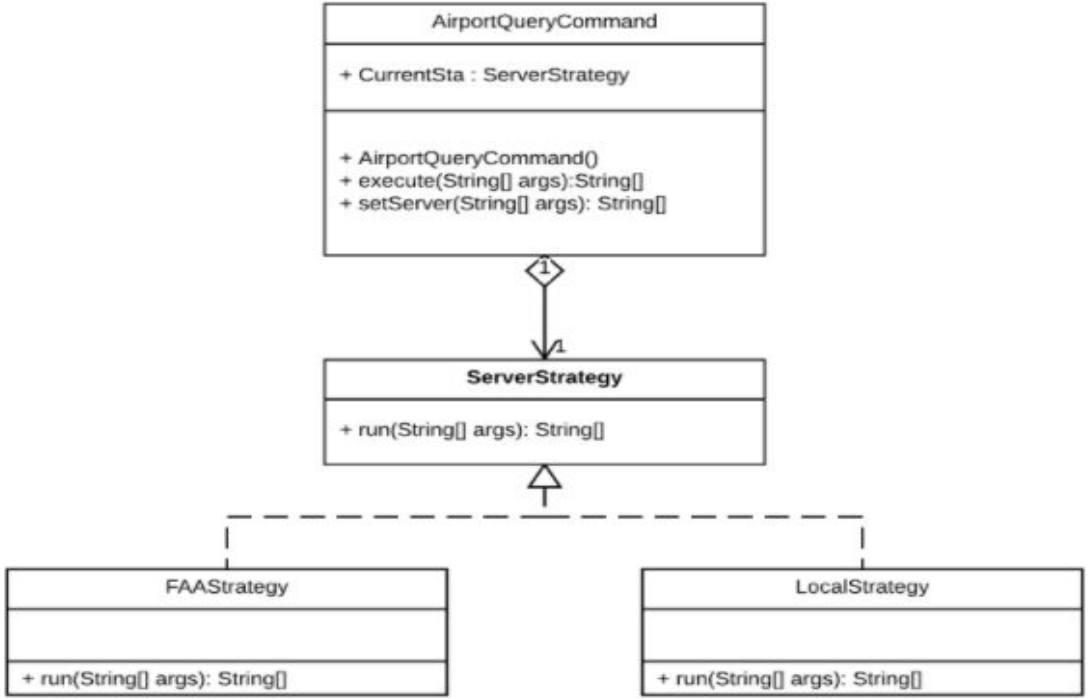


Airport Information Subsystem

Strategy Pattern:

- The “implementation” that the system had to choose between was whether to query from:
  - Local Test Files
  - FAA Server
- Added the ability to expand the system while still following Open Close Principle

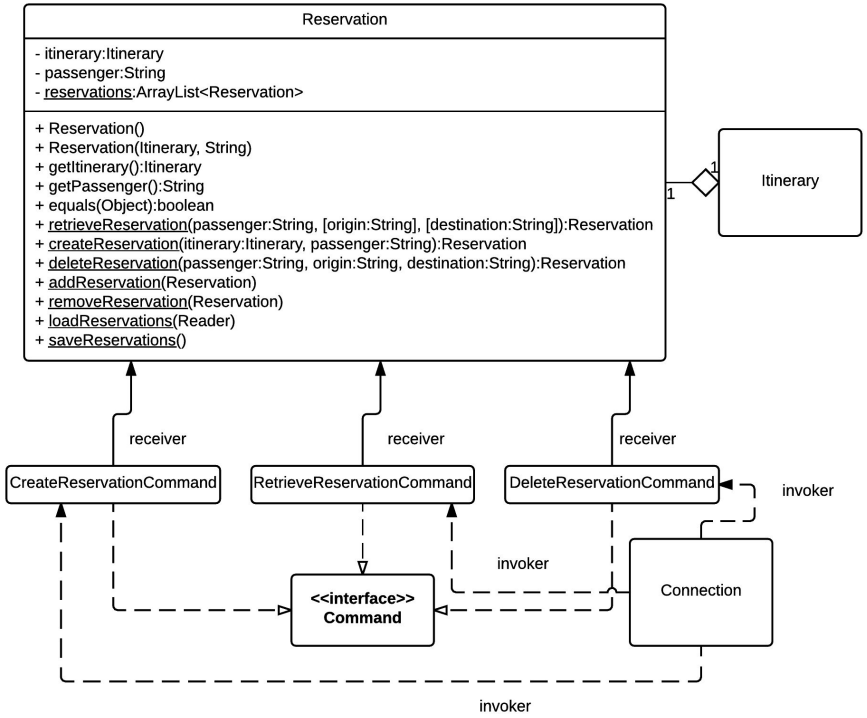
Airport Information Subsystem



Reservation Subsystem

- Persistence
- Undo/redo
- No changes to Reservation class for multiple connections

Reservation Subsystem

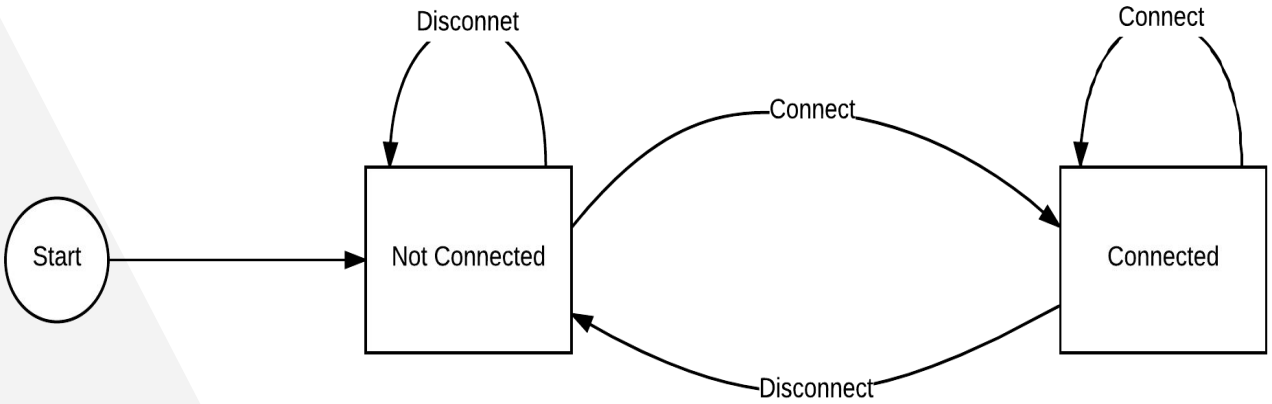


Connection Subsystem

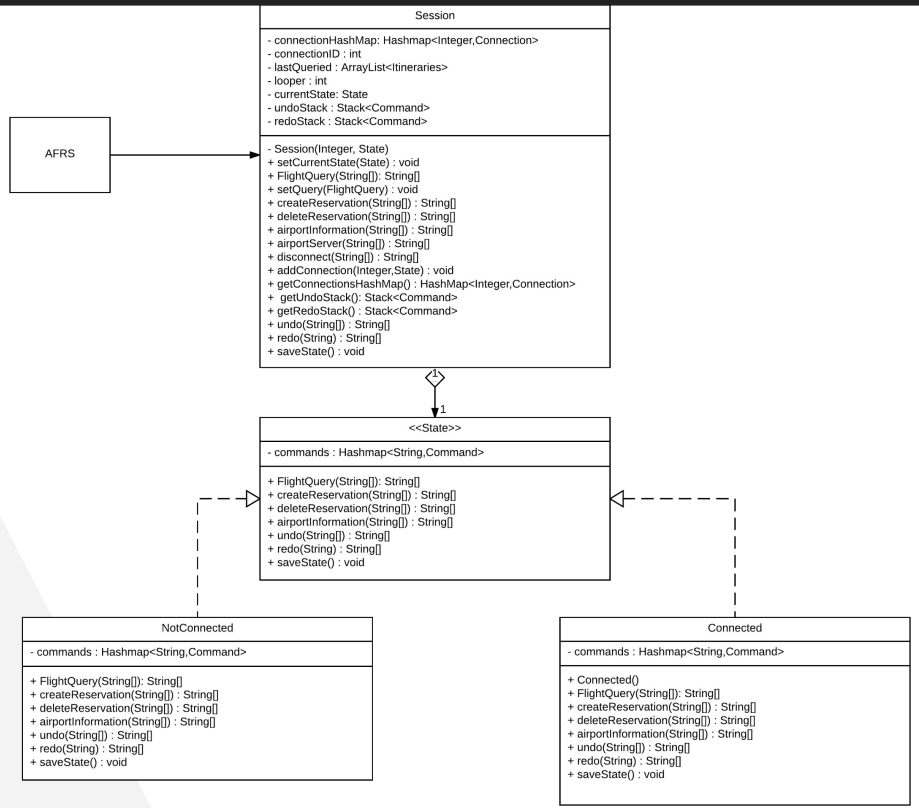
State Pattern:

- State Pattern was used to handle the two states of a Session:
  - Connected and Not Connected
- When connected, user has full functionality
- When not connected, user has no functionality.

Connection Subsystem



Connection Subsystem



5.
Strengths and Weaknesses of our Design

Strengths and Weaknesses

Strengths

- Easy to extend (Open-Closed Principle was followed]
- Focused on finalizing design documentation before implementation
- All of the requirements were implemented due to effective design and planning

Weaknesses

- Session subsystem class could turn into a GOD class if more requirements were to be added
- Copy-Paste-Programming (duplicate code) in the UI subsystem

6.
Our Team's Process

Team Process

First Two Weeks... Agile Methodologies

- Semi-Weekly SCRUM meetings
- Communicated with the customer (Professor) to clarify requirements
- Individuals and interactions over processes and tools
  - Trello
  - Slack
  - Google Docs

Last Week... Iterative Process

- Fix Design ⟶ Implement ⟶ Test ⟶ Repeat

THANK YOU!

Any questions?