Flight Surety Report

Setup and Run

Install

This repository contains Smart Contract code in Solidity (using Truffle), tests (also using Truffle), dApp scaffolding (using HTML, CSS and JS) and server app scaffolding.

To install, download or clone the repo, then:

```
npm install truffle compile
```

Develop Client

To run truffle tests:

```
truffle test ./test/flightSurety.js truffle test ./test/oracles.js
```

To use the dapp:

```
truffle migrate npm run dapp
```

To view dapp:

```
http://localhost:8000
```

Develop Server

```
npm run server truffle test ./test/oracles.js
```
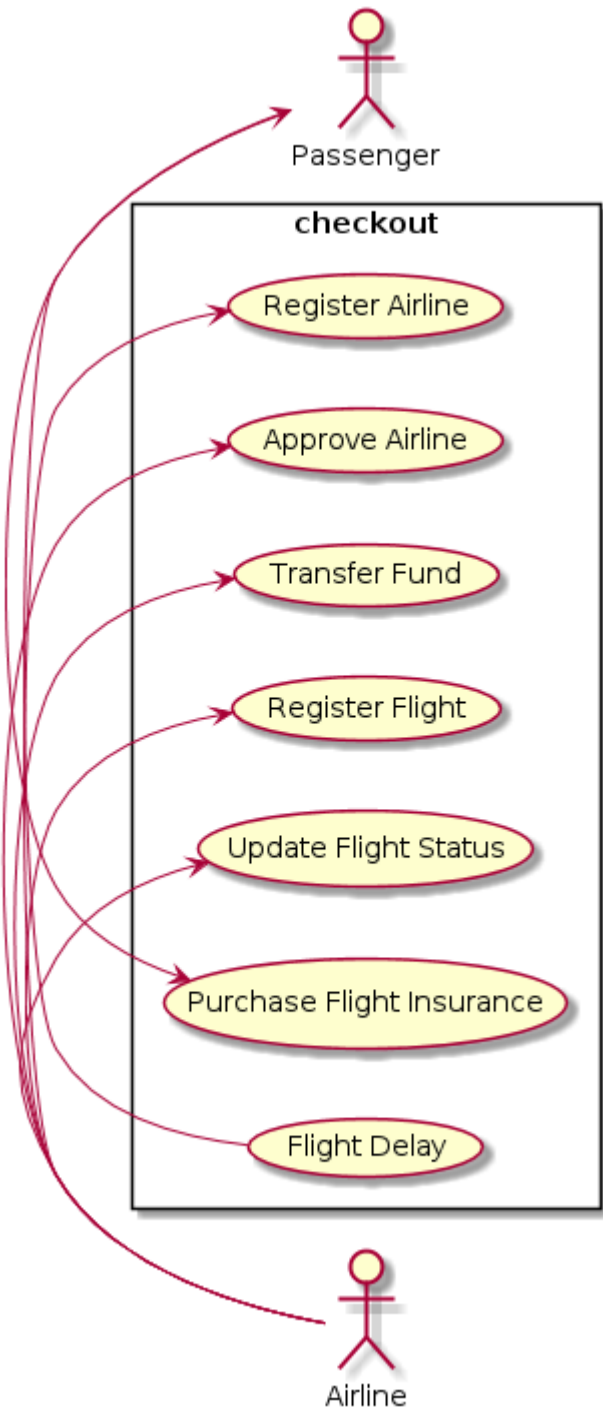
Deploy

To build dapp for prod: `npm run dapp:prod`
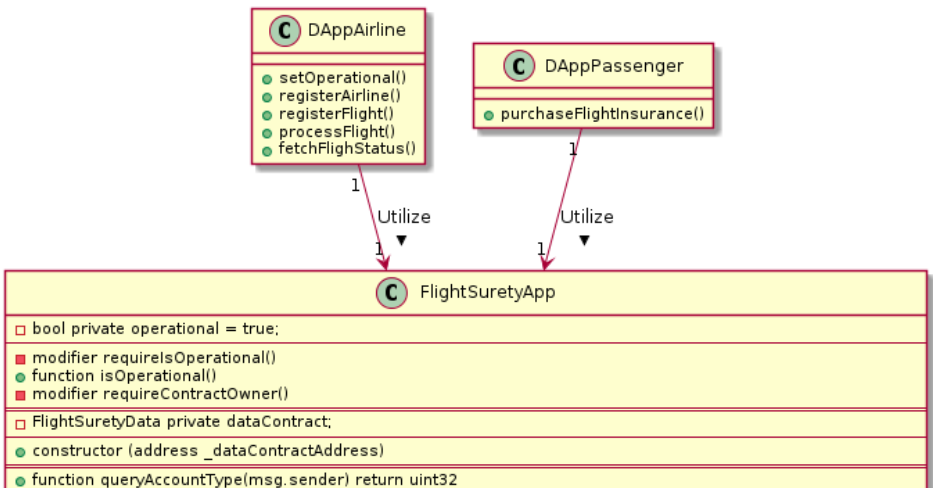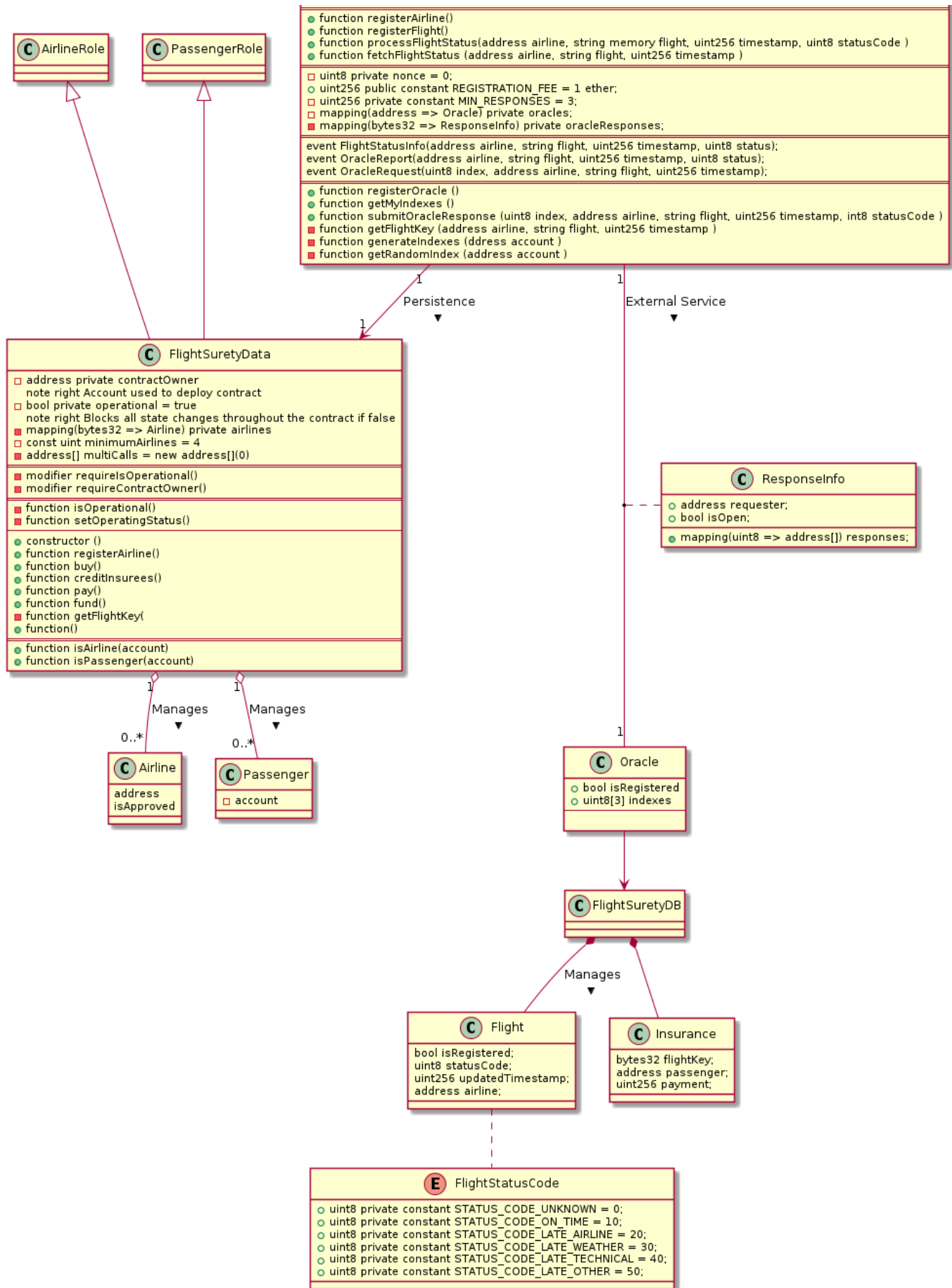
Deploy the contents of the ./dapp folder

Resources

- [How does Ethereum work anyway?](#)
- [BIP39 Mnemonic Generator](#)
- [Truffle Framework](#)
- [Ganache Local Blockchain](#)
- [Remix Solidity IDE](#)
- [Solidity Language Reference](#)
- [Ethereum Blockchain Explorer](#)
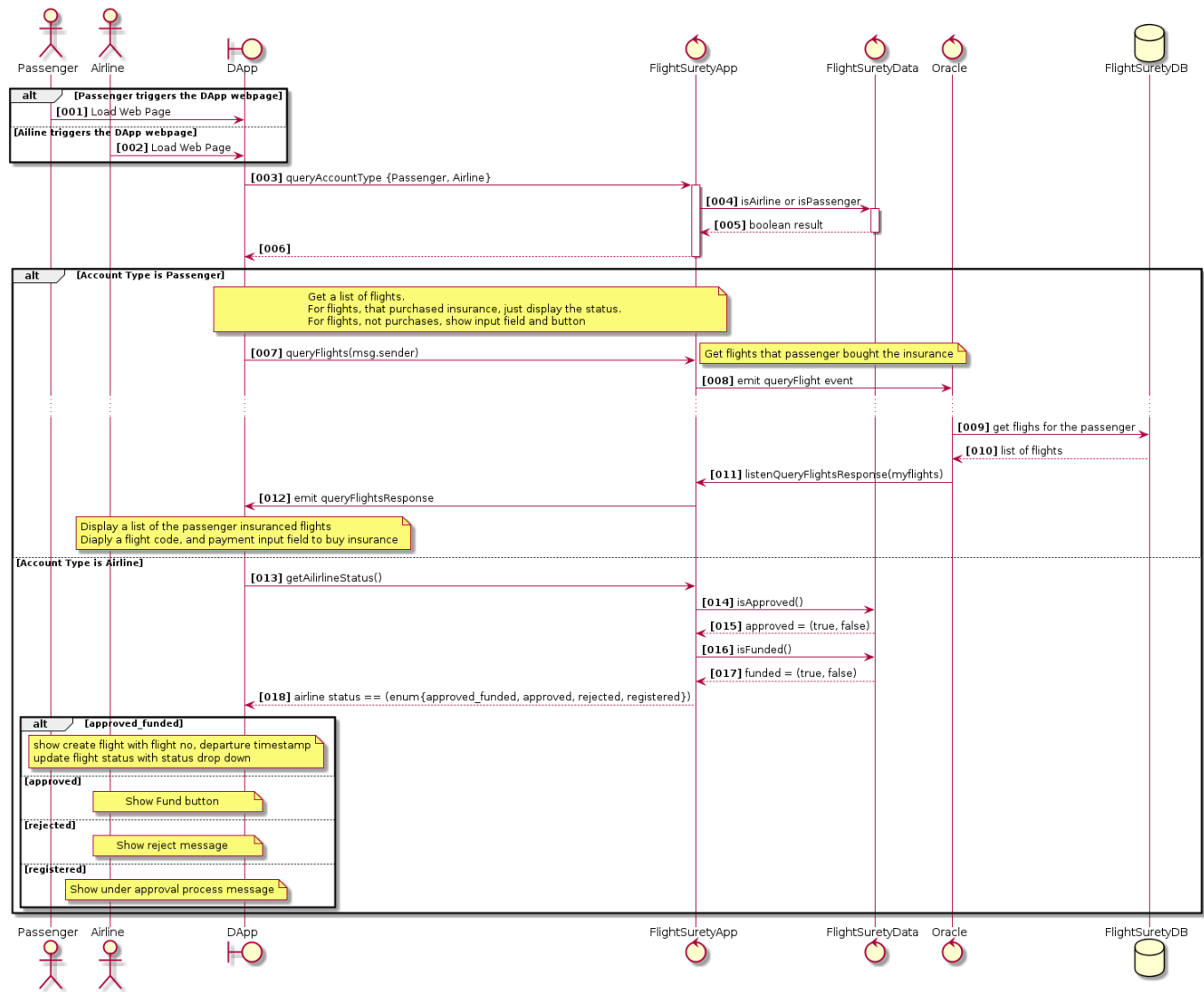- [Web3Js Reference](#)
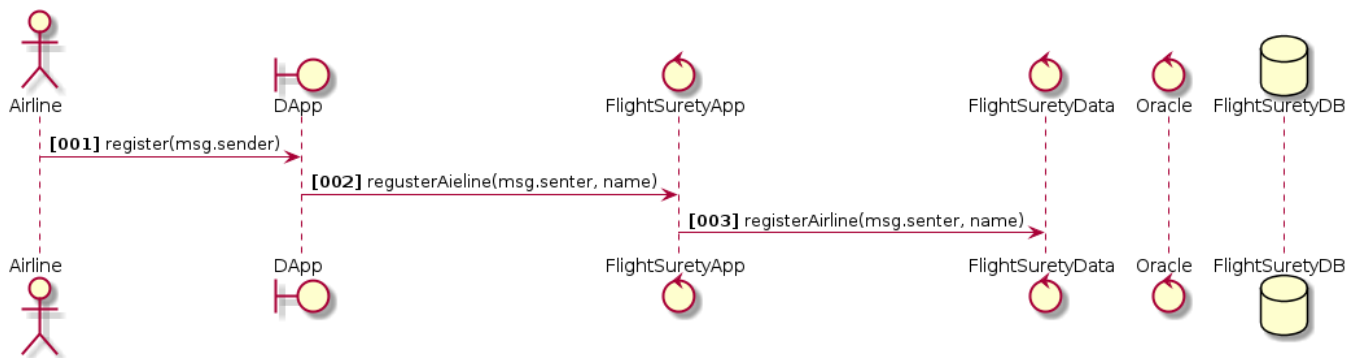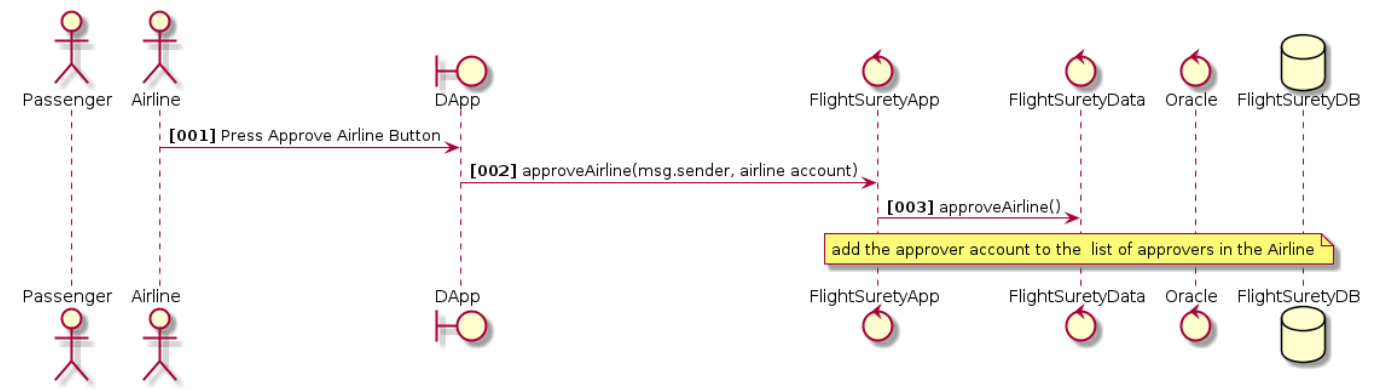
Use Cases

Class Diagram

The diagram contains the following classes and content:

AirlineRole (C)

PassengerRole (C)

Top class:
- function registerAirline()
- function registerFlight()
- function processFlightStatus(address airline, string memory flight, uint256 timestamp, uint8 statusCode)
- function fetchFlightStatus (address airline, string flight, uint256 timestamp)

- uint8 private nonce = 0;
- uint256 public constant REGISTRATION_FEE = 1 ether;
- uint256 private constant MIN_RESPONSES = 3;
- mapping(address => Oracle) private oracles;
- mapping(bytes32 => ResponseInfo) private oracleResponses;

- event FlightStatusInfo(address airline, string flight, uint256 timestamp, uint8 status);
- event OracleReport(address airline, string flight, uint256 timestamp, uint8 status);
- event OracleRequest(uint8 index, address airline, string flight, uint256 timestamp);

- function registerOracle ()
- function getMyIndexes ()
- function submitOracleResponse (uint8 index, address airline, string flight, uint256 timestamp, int8 statusCode)
- function getFlightKey (address airline, string flight, uint256 timestamp)
- function generateIndexes (ddress account)
- function getRandomIndex (address account)

Persistence ▼   External Service ▼

FlightSuretyData (C)
- address private contractOwner
  note right Account used to deploy contract
- bool private operational = true
  note right Blocks all state changes throughout the contract if false
- mapping(bytes32 => Airline) private airlines
- const uint minimumAirlines = 4
- address[] multiCalls = new address[](0)

- modifier requireIsOperational()
- modifier requireContractOwner()

- function isOperational()
- function setOperatingStatus()

- constructor ()
- function registerAirline()
- function buy()
- function creditInsurees()
- function pay()
- function fund()
- function getFlightKey(
- function()

- function isAirline(account)
- function isPassenger(account)

ResponseInfo (C)
- address requester;
- bool isOpen;

- mapping(uint8 => address[]) responses;

Manages ▼ 1 ... 0..*

Airline (C)
address
isApproved

Passenger (C)
- account

Oracle (C)
- bool isRegistered
- uint8[3] indexes

FlightSuretyDB (C)

Manages ▼

Flight (C)
bool isRegistered;
uint8 statusCode;
uint256 updatedTimestamp;
address airline;

Insurance (C)
bytes32 flightKey;
address passenger;
uint256 payment;

FlightStatusCode (E)
- uint8 private constant STATUS_CODE_UNKNOWN = 0;
- uint8 private constant STATUS_CODE_ON_TIME = 10;
- uint8 private constant STATUS_CODE_LATE_AIRLINE = 20;
- uint8 private constant STATUS_CODE_LATE_WEATHER = 30;
- uint8 private constant STATUS_CODE_LATE_TECHNICAL = 40;
- uint8 private constant STATUS_CODE_LATE_OTHER = 50;

Operation Design

Initial DApp Web Page

Register Airline

Approve Airline

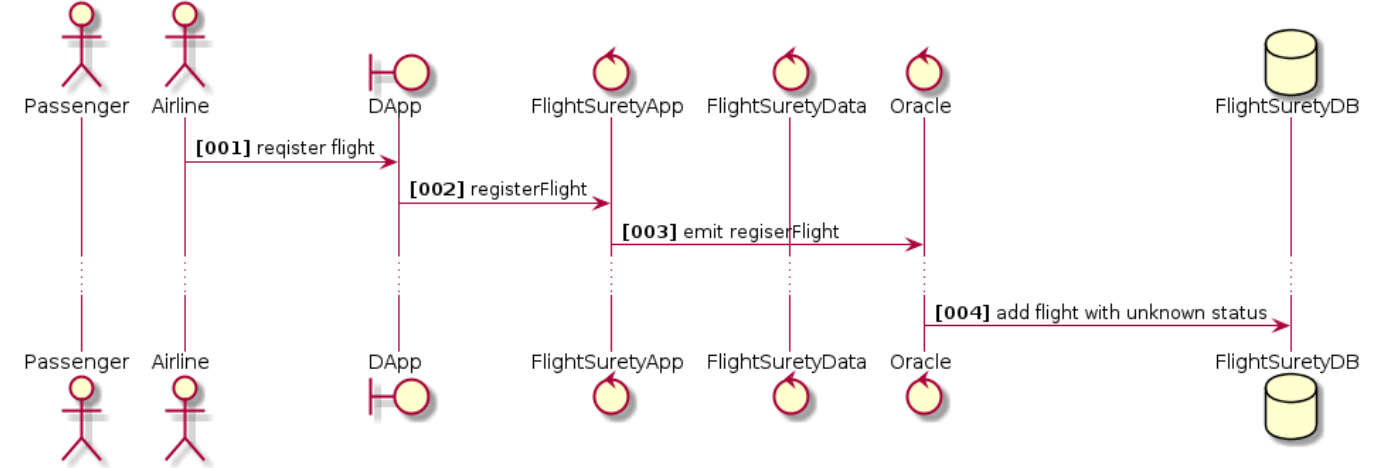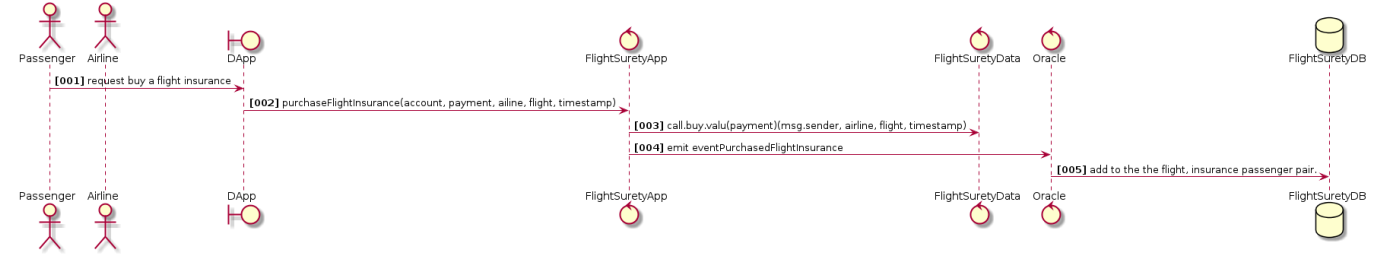## register Flight

Buy Insurance

Update Flight Status