FLIGHT MANAGEMENT SYSTEM DESIGN

Patryk Zboch

Contents

D1: Assumptions and Expectations

Assumptions

1. The FMS is functioning correctly.
2. The communication link between the FMS and the Flight Archive System (FAS) is functioning correctly.
3. The communication link between the FMS and the Route Finder System (RFS) is functioning correctly.
4. The communication link between the FMS and the External System that sets flight strips to inactive is functioning correctly.
5. The pilot has access to an FMS interface.
6. The Air Traffic Service Assistant (ATSA) has access to an FMS interface.
7. Access to the FMS is only granted If a valid 6-digit PIN is provided first.
8. The FMS is not responsible for the physical management of flights.
9. A pilot that does not intend to make a flight will not use the FMS to create a flight plan.
10. The FMS is not responsible for checking if a pilot is certified to land at the destination airport or take off from the departure airport.
11. The FMS will only create flight strips for flight plans that have already been created.
12. The FAS and RFS will be available while the FMS is operational.
13. The FMS will reject any invalid or incomplete information provided by its users and provide an error message.
14. A flight Plan or flight strip will only be created if all necessary information is filled out by the user.

Expectations

1. The FMS system should interact with external systems like RFS and FAS and any other external software smoothly and without error. This will make the interaction fast, safe, and efficient.
2. The FMS system will record and store critical data like an existing flight plan or a flight strip in the event of a system failure.
3. The FMS system can smoothly access, verify and exchange data between its users and any external systems.
4. The FMS system should have strong security measures like access protocols and data encryption to protect data from security breaches and any unauthorized users.
5. The FMS system should be designed to have a user-friendly interface increasing the user's satisfaction with the system and making it easy to use.
6. The FMS system should be regularly maintained, and any issues should be fixed accordingly.

D2: Functional and Non-Functional Requirements

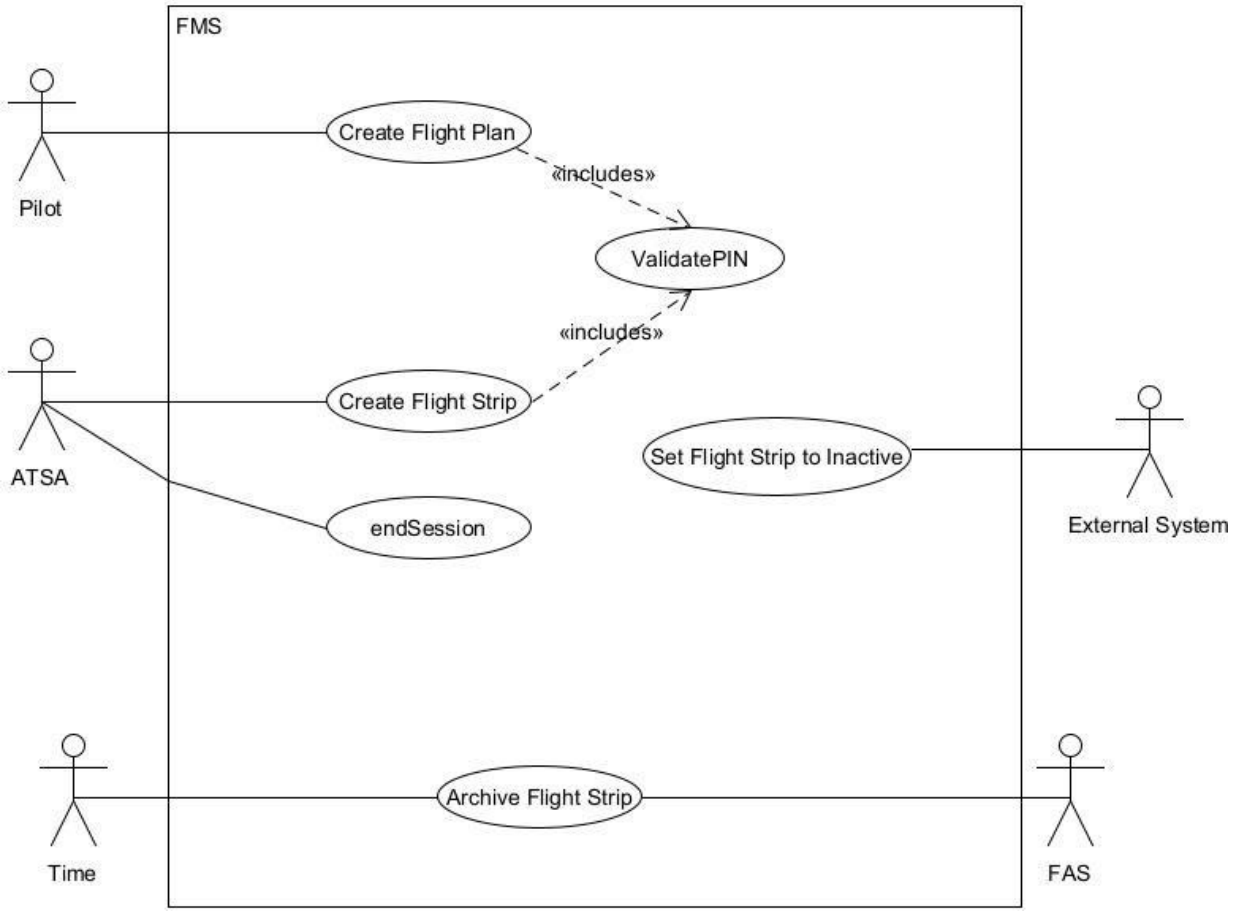
Functional Requirements

ID	Description	Priority
FR1	FMS shall assist with the management of flight plans	M
FR2	FMS shall assist with the management of flight strips	M
FR3	FMS shall interact with the pilot to create flight plans	M
FR4	FMS shall interact with the ATSA to create flight strips	M
FR5	FMS shall require a 6-digit PIN from its users before granting access	M
FR6	FMS shall manage a collection of pilot records which record: • pilot's unique identifier (PID) • pilot's PIN • Pilot name and contact phone number. • List of airports that the pilot is certified to take off from, and land at using the International Air Transport Association's (IATA) location identifier format.	M
FR7	FMS shall record the following information for each airport: • IATA code • Full Name • Name of the nearest city to the airport • Location (latitude/longitude coordinates)	M
FR8	FMS shall allow an ATSA to process multiple flight plans in each session	M
FR9	FMS shall manage a collection of ATSA records which record: AID and PIN	M
FR10	FMS shall use the Route Finder System (RFS) system to find a list of available routes between the departure and destination airport	
FR11	FMS shall send a list of inactive flight strips (FSID along with departure and destination airports (IATA codes) to the FAS at 23:59 (GMT)	M
FR12	FMS shall delete the inactive flight strips and their associated flight plan records after they have been archived	M
FR13	FMS shall end a session with the ATSA after they log out	M
FR14	FMS shall use an External System to set flight strips to inactive after a flight	M

Non-Functional Requirements

NFR1	The FMS shall have high security measures enforced to protect the system from data breaches and unauthorised access	M
NFR2	The FMS shall have a high performance to ensure that its operations are completed quickly minimizing delays	M

D3: Use Case Diagram



D4: Use Case Specifications

Create Flight Plan

Use Case: Create Flight Plan
ID: ***1***
Goal: ***The pilot creates a flight plan for a flight*** **using the FMS**
Primary actor: ***Pilot***
Secondary actor(s): **None**
Preconditions: 1. ***FMS is operational.*** 2. **Pilot has the following information to create a flight plan: a unique identifier for the flight plan (FPID), the pilot's (PID), the codes (IATA format) for the planned departure and destination airports, and an expected departure time (EDT).**
Postconditions: **1. Flight Plan is created**
Main flow: 1. ***Include(ValidatePIN)*** 2. ***FMS accesses and validates pilot's records*** 3. ***FMS requests information for the flight plan*** 4. ***Pilot enters*** **a unique identifier for the flight plan (FPID), pilot's PID, the codes (IATA format) for the planned departure and destination airports, and the expected departure time (EDT)** 5. ***FMS validates that airports entered are valid*** 6. ***FMS records the IATA code, full name, nearest city name, and location (latitude/longitude coordinates) of each airport*** 7. ***FMS creates the flight plan with the following information: FPID; pilot's PID;*** **the departure airport (IATA code); the destination airport (IATA code); EDT**
Alternative flows: ***2a. Pilot's records are invalid*** 1. ***FMS returns records are invalid message*** 2. ***Use case terminates***

5a. Invalid airports

3. ***FMS returns airports are invalid message***
4. ***Return to step 3 in the main flow***

7a. Error creating flight plan

5. ***FMS returns an error creating flight plan message***
6. ***Use case terminates.***

Create Flight Strip

Use Case: ***Create Flight Strip***
ID: 2
Goal: ***The ATSA creates a flight strip for a flight plan***
Primary actor: ***ATSA***
Secondary actor(s): ***Route Finder System (RFS)***
Preconditions: 1. ***FMS is functional*** 2. ***RFS system is functional*** 3. **A flight plan has already been created** 4. **Communication link between FMS and RFS is functioning correctly**
Postconditions: 1. ***A flight strip is successfully***
Main flow: 1. ***Include (ValidatePIN)*** 2. ***FMS accesses and validates ATSA's records*** 3. ***FMS starts a session with ATSA*** 4. ***ATSA requests a flight plan to process*** 5. ***FMS retrieves necessary information from the flight plan*** 6. ***FMS provides IATA codes for departure and destination airports and the EDT to RFS*** 7. ***Route Finder System (RFS) gets a list of available routes and sends the list back to FMS*** 8. **FMS displays available routes along with flight plan details to ATSA.** 9. **ATSA selects a route** 10. **Flight strip is created with the following information: FSID; the departure airport (IATA code); the destination airport (IATA code); the allocated route; the EDT; status flag** **Steps 4...10 are repeated until all flight plans are processed**

Alternative flows:

2a. ATSA's records are invalid

1. ***FMS returns records are invalid message***
2. ***Use case terminates***

3a. Session cannot be started

3. ***FMS returns error message.***
4. ***Use case terminates***

7a. RFS could not find any available routes

5. ***RFS returns 0 available routes to FMS***
6. ***FMS prompts ATSA that no routes are available for the flight plan***
7. ***Return to step 4 in the main flow***

10a. Error creating flight strip

8. ***FMS returns an error creating flight strip message***
9. ***Use case terminates.***

Archive Flight Strip

Use Case: ***Archive Flight Strip***
ID: ***3***
Goal: ***An inactive flight strip is archived and deleted from FMS***
Primary actor: **Time**
Secondary actor(s): **FAS**
Preconditions: 1. ***FMS is functional.*** 2. ***FAS is functional.*** 3. ***Communication link between FMS and FAS is functional*** 4. ***FMS has inactive flight strips.*** 5. ***The time is 23:59 (GMT)***
Postconditions: 1. **Inactive flight strips are archived by FAS.** 2. **FMS deletes the inactive flight strips and their associated flight plan records from its database.**

Main flow:

1. **FMS sends the FAS system the list of inactive flight strips (FSID along with departure and destination airports (IATA codes) at 23:59 (GMT).**
2. **FAS receives the list and archives inactive flight strips.**
3. **FAS prompts FMS that inactive flight strips have been archived.**
4. **FMS deletes the inactive flight strips and their associated flight plan records.**

Alternative flows:

***1a. FMS fails to send the FAS* the list of inactive flight strips**

1. ***FMS logs error***
2. ***Use case terminates***

3a. archive was unsuccessful

1. ***FAS notifies FMS that archive was unsuccessful.***
2. ***FMS logs error***
3. ***Use case terminates***

ValidatePIN

Use Case: ValidatePIN
ID: 4
Goal: FMS validates the PIN of the user
Primary actor: *User*
Secondary actor(s): None
Preconditions: 1. **FMS is functioning correctly.**
Postconditions: 1. **PIN is validated**
Main flow: 1. **user enters PIN** 2. **FMS validates the user's PIN**
Alternative flows: ***4a. invalid PIN*** 1. **FMS notifies user of invalid PIN** 2. ***Return to step 1 in the main flow***

endSession

Use Case: endSession
ID: 5
Goal: ATSA logs out and FMS ends the session
Primary actor: *ATSA*
Secondary actor(s): None
Preconditions: 1. ATSA is currently logged into FMS 2. ATSA is done using FMS and wants to end the session 3. FMS is functioning correctly
Postconditions: 1. The session between ATSA and FMS has successfully ended
Main flow: 1. ATSA wishes to log out of the FMS 2. FMS requests a confirmation that they want to log out and end the current session 3. ATSA confirms that they wish to log out and end the session 4. FMS logs ATSA out of the system and ends the session 5. FMS returns ATSA to the enter PIN screen
Alternative flows: ***2a. ATSA no longer wants to log out*** *1. ATSA selects the cancel option* *2. Log out box disappear* *3. Session between ATSA and FMS is continued and ATSA is not logged out*

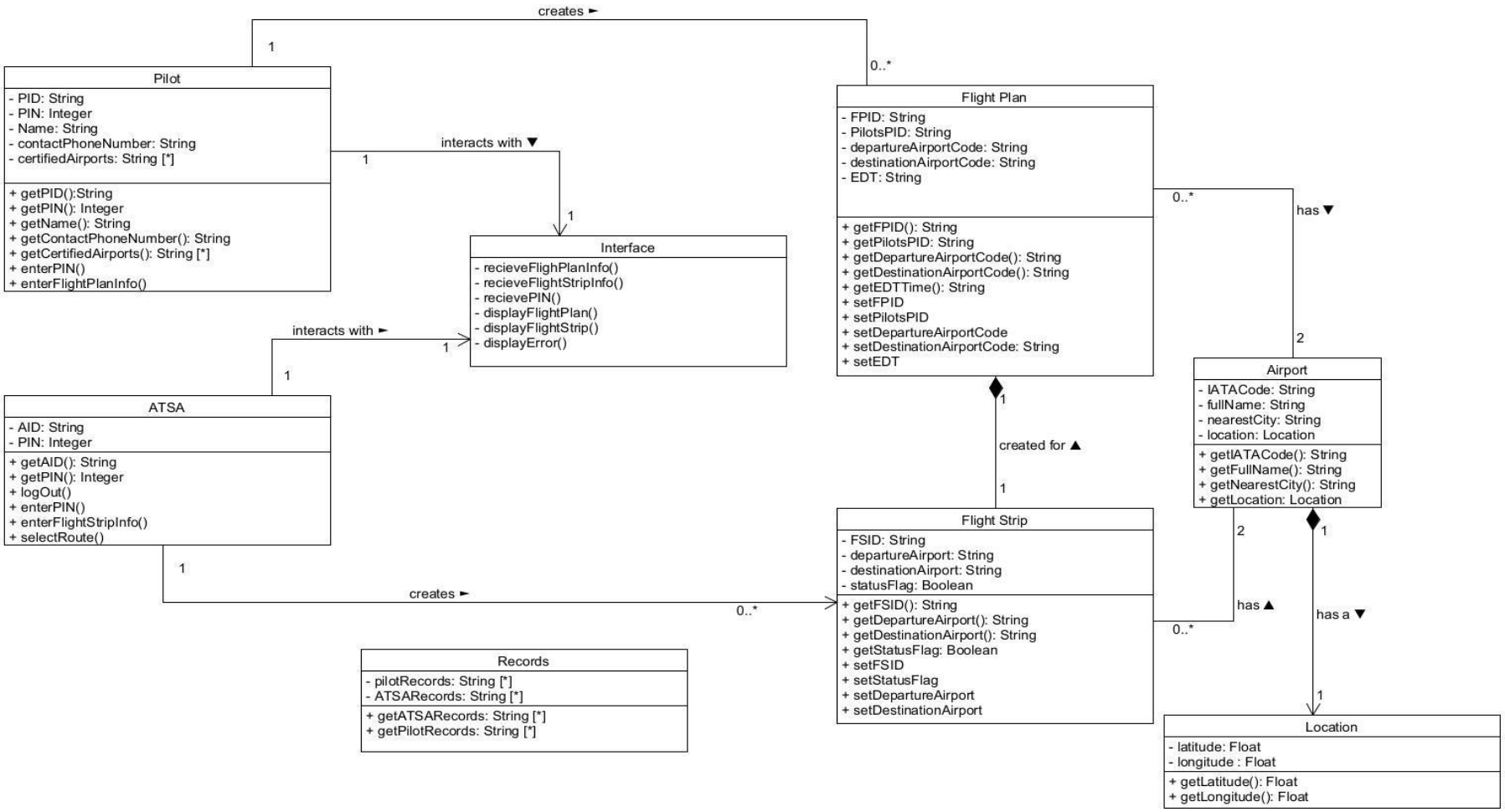
Set Flight Strip to Inactive

Use Case: Set Flight Strip to Inactive
ID: 6
Goal: External system sets the flight strip to inactive after the flight ends
Primary actor: *External System*
Secondary actor(s): None
Preconditions: 1. FMS is functioning correctly 2. External System is functioning correctly 3. Communication link between FMS and the External System is functioning correctly 4. A flight strip has been already created is its status is active 5. The flight for the corresponding flight strip has ended
Postconditions: 1. Flight strip is set to inactive by the External System
Main flow: 1. External system is notified that the flight has ended 2. External system searches the FMS for the corresponding unique identifier for the flight strip (FSID) 3. External system sets the flight strip to inactive in the FMS
Alternative flows: ***2a. Flight strip FSID could not be found in FMS>*** 1. *External System notifies FMS that flight strip could not be found* 2. *Use case terminates* ***3a. Flight strip could not be set to inactive>*** 1. *External System notifies FMS that flight strip could not be set to inactive* 2. *Use case terminates*

D5: Traceability Matrix

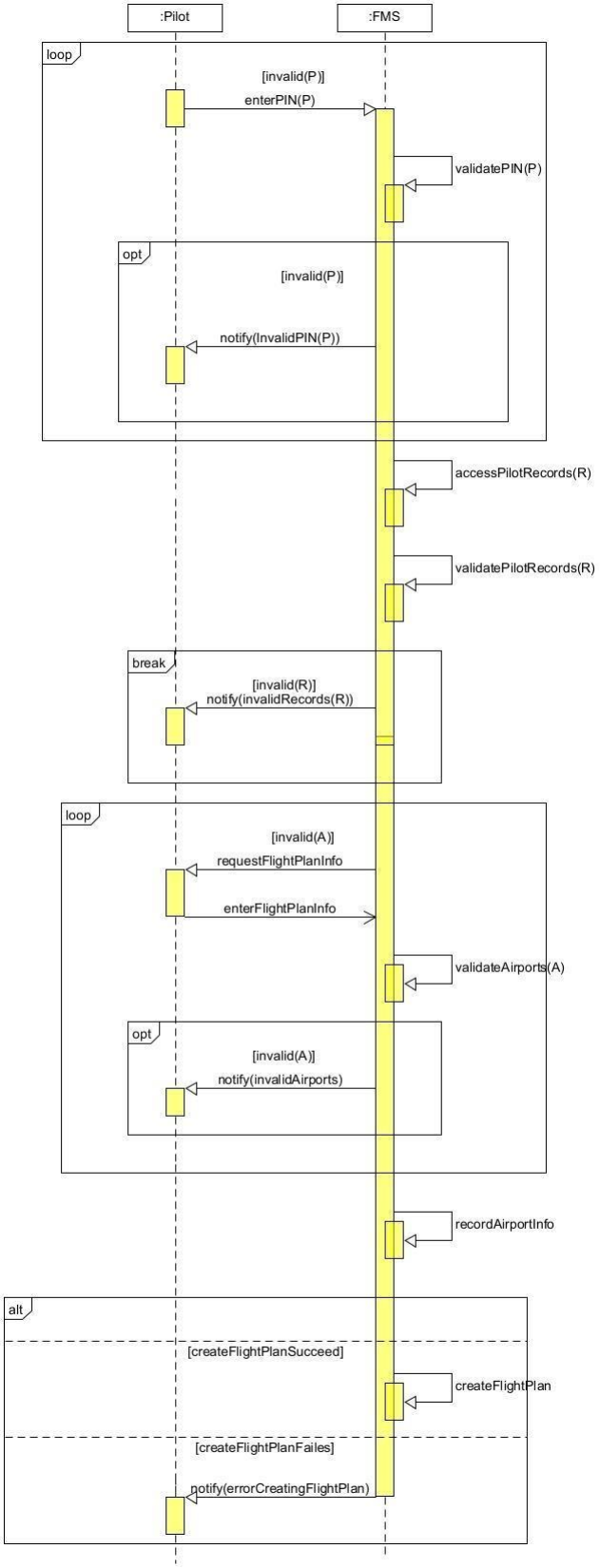
	UC1	UC2	UC3	UC4	UC5	UC6
FR1	X	X	X			
FR2		X	X			X
FR3	X					
FR4		X				
FR5	X	X		X		
FR6	X					
FR7	X					
FR8		X				
FR9		X				
FR10		X				
FR11			X			
FR12			X			
FR13					X	
FR14						X

D6: Class Diagram

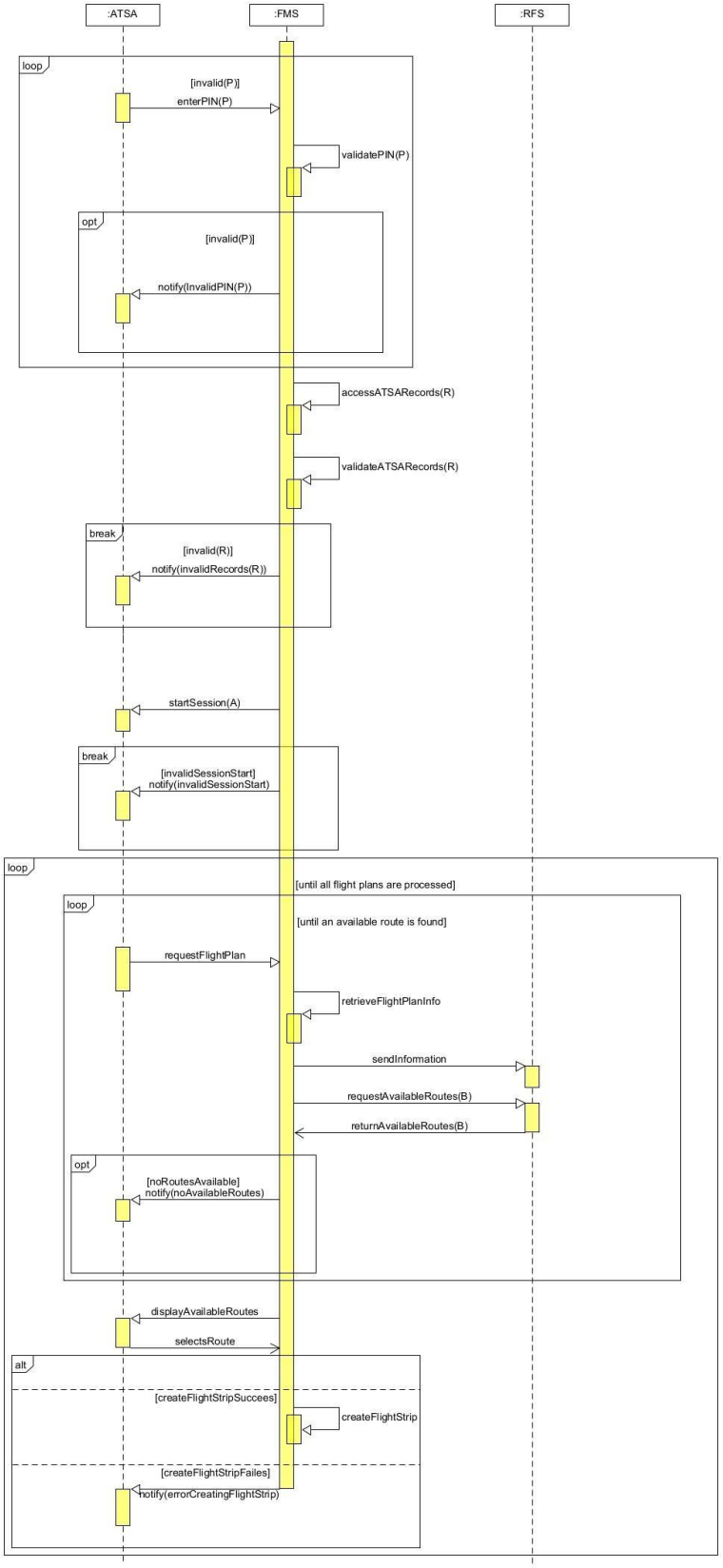


D7: Sequence Diagram

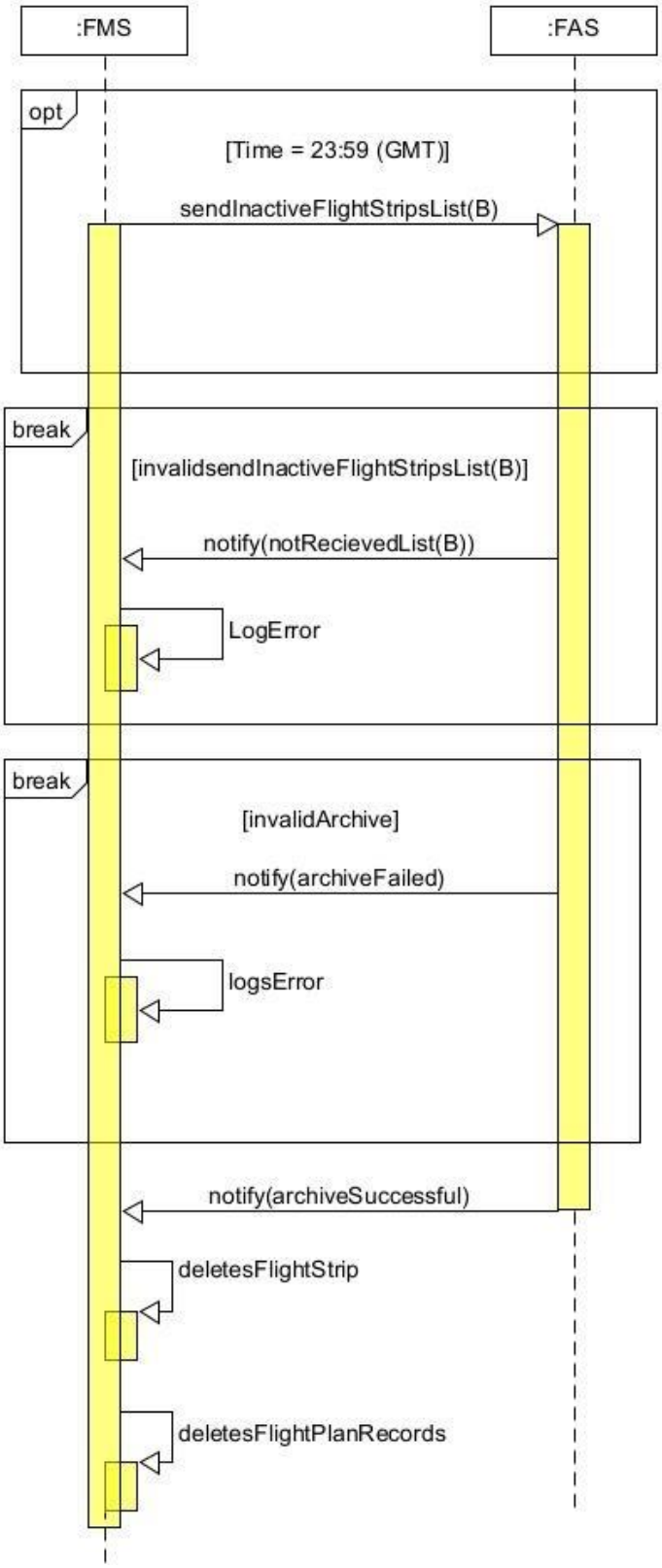
Create Flight Plan Sequence Diagram



Create Flight Strip Sequence Diagram

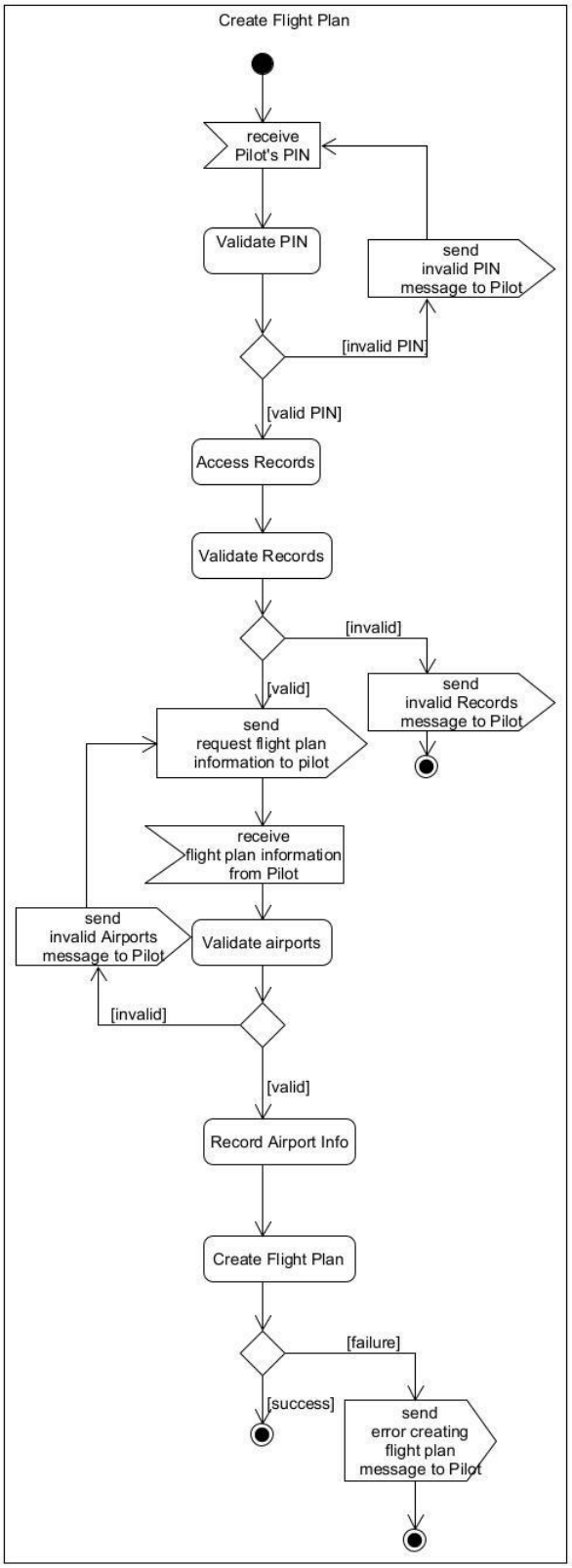


Sequence Diagram Archive Flight Strip

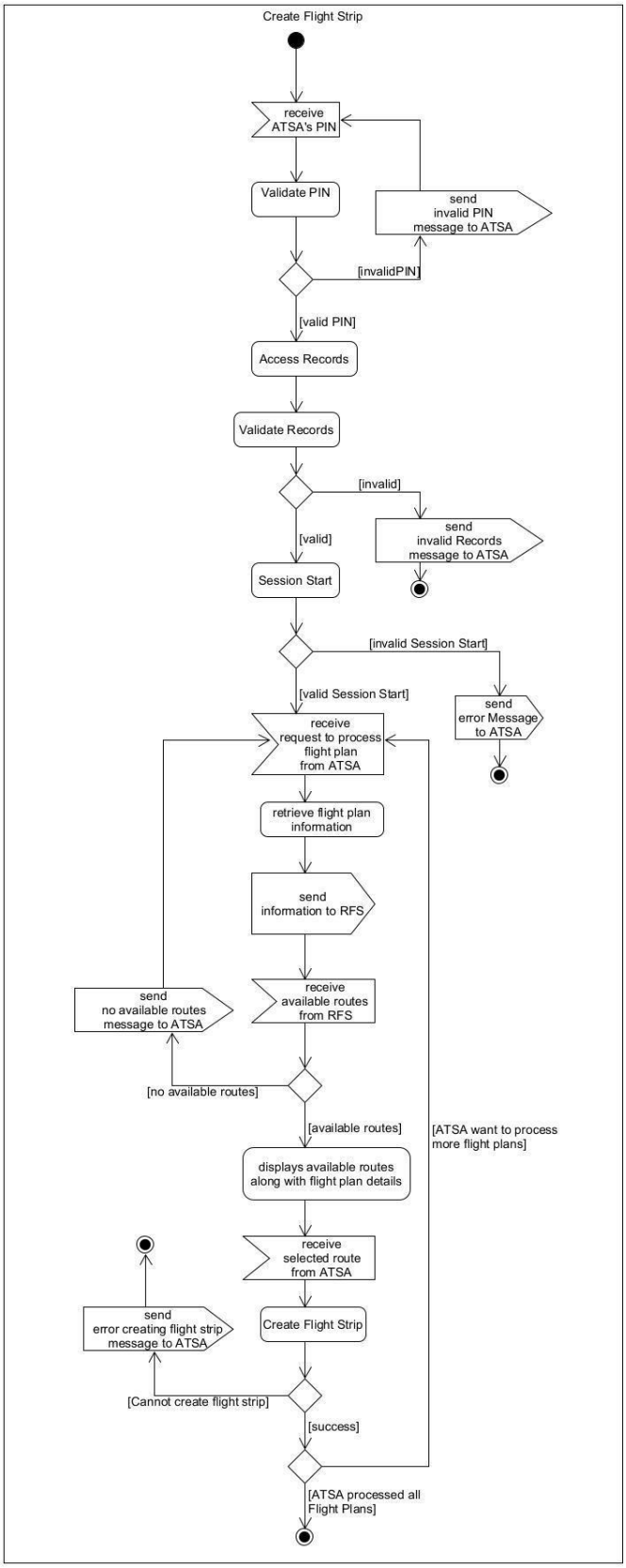


D8: Activity Diagrams

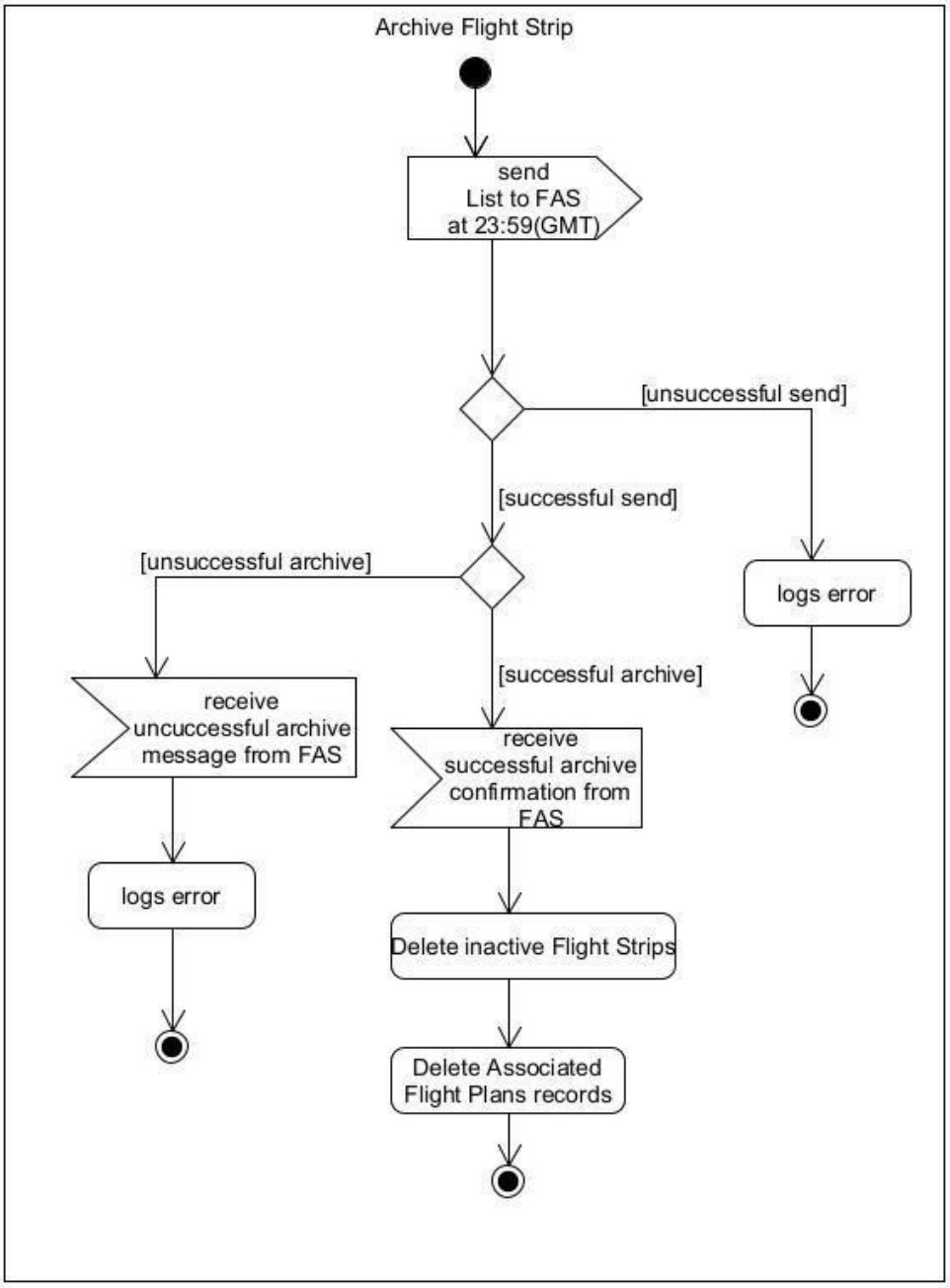
Create Flight Plan Activity Diagram



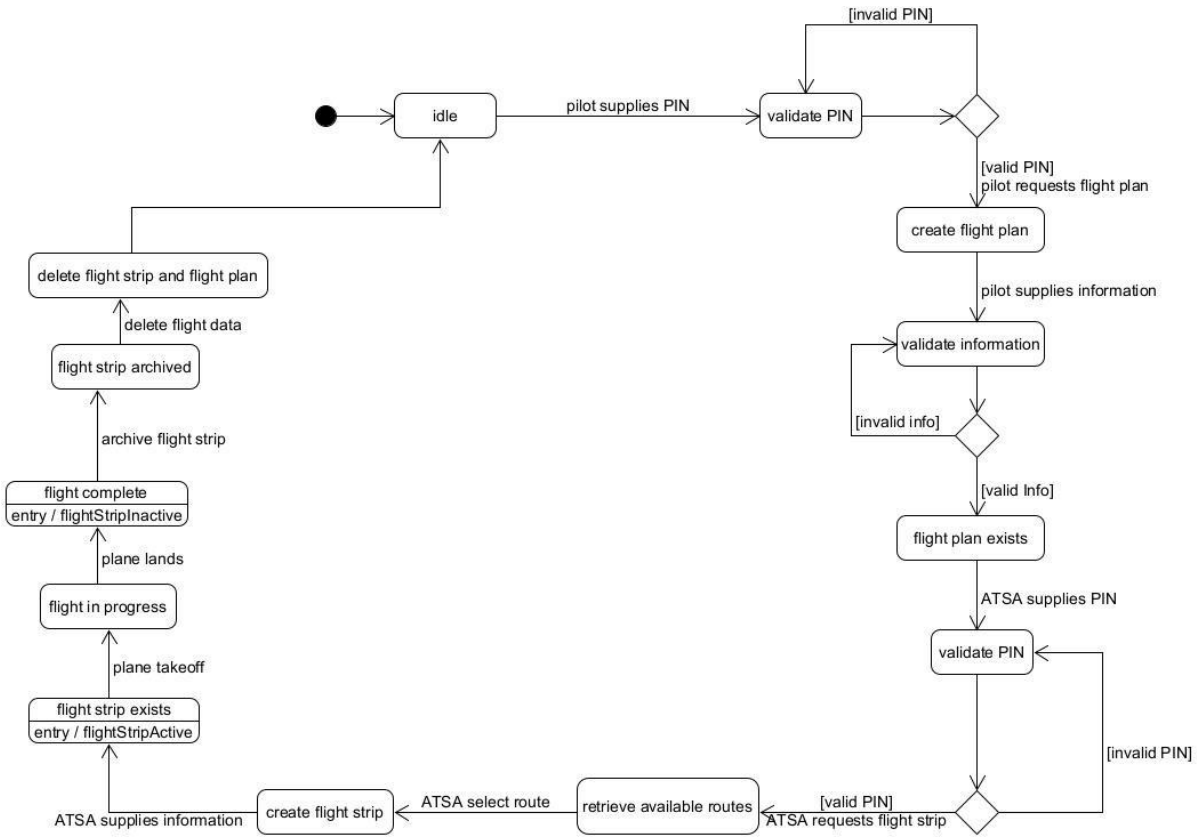
Create Flight Strip Activity Diagram



Archive Flight Strip Activity Diagram



D9: State Machine Diagram



D10: Test case scenarios

Create Flight Plan Activity Diagram

Path	Comment	Path Condition
1	The pilot enters a valid PIN, the pilots records are valid, airports entered by pilot are valid, create flight plan is successful	• ValidatePIN(valid) • ValidateRecords(valid) • ValidateAirports(valid) • createFlightPlan(success)
2	The pilot enters a valid PIN but the records are invalid	• ValidatePIN(valid) • ValidateRecords(invalid)
3	The pilot enters a valid PIN, the records are valid, the airports are valid but the creation of the flight plan was a failure	• ValidatePIN(valid) • ValidateRecords(valid) • ValidateAirports(valid) • createFlightPlan(failure)
4	The pilot enters an invalid PIN	• ValidatePIN(invalid)
5	The pilot enters a valid PIN, the records are valid but the airports are invalid	• ValidatePIN(valid) • ValidateRecords(valid) • ValidateAirports(invalid)

Create Flight Strip Activity Diagram

Path	Comment	Path Condition
1	ATSA enters an invalid PIN	• validatePIN(invalid)
2	ATSA enters a valid PIN but the records are invalid	• validatePIN(valid) • validateRecords(invalid)
3	ATSA enters a valid PIN, the records are valid, session start was valid, routes are available , creating the flight strip was successful and ASTA processed all flight plans	• validatePIN(valid) • validateRecords(valid) • sessionStart(valid) • routes(available) • createFlightStrip(success) • ATSAProcessedALLFlightPlans is TRUE
4	ATSA enters valid PIN, the records are valid but the session start was invalid	• validatePIN(valid) • validateRecords(valid) • sessionStart(invalid)
5	ATSA enters a valid PIN, the records are valid, the session start is valid but there are no available routes	• validatePIN(valid) • validateRecords(valid) • sessionStart(valid) • routes(notAvailable)
6	ATSA enters a valid PIN, the records are valid, session	• validatePIN(valid) • validateRecords(valid)

	start was valid, routes are available, creating the flight strip was a failure	• sessionStart(valid) • routes(available) • createFlightStrip(failure)
7	ATSA enters a valid PIN, the records are valid, session start was valid, routes are available, creating the flight strip was successful and ASTA has not processed all flight plans	• validatePIN(valid) • validateRecords(valid) • sessionStart(valid) • routes(available) • createFlightStrip(success) ATSAProcessedALLFlightPlans is FALSE

Archive Flight Strip Activity Diagram

Path	Comment	Path Condition
1	Send list was successful and the archive was successful	• sendList(success) • archive(success)
2	Send list was unsuccessful	• sendList(unsuccessful)
3	Send list was successful but the archive was unsuccessful	• sendList(success) • archive(unsuccessful)