2017

System analysis and design

HUSKY AIRPLANE RENTAL MANAGEMENT SYSTEM

Version 0.2 – Nov 15, 2017

Revision History

Version	Date	Name	Description
0.1	10/9/17	System Requirements draft	Initial Creation
0.11	10/30/17	System Requirement update	Detailing some user requirements
0.2	11/13/17	System design draft	Initial Creation

Contents

Section 1 General System Requirements

1.1 Project Sponsor

Project Sponsor should be one of Husky Air management, who can take charge of all the departments involved.

1.2 Business need

Husky Air is a plane rental company who provides plane rental and instructional service to qualified public renters. Along with rapid business growth, the original manual system is becoming less suitable to support its rental process and future new services. The company will extend its business scope by adding more instructors and a variety of models of planes which is too complicated for the current manual system. Meanwhile, the efficacy of the usage of planes is not ideal in terms of reservation change by customers. Therefore, Husky Air needs a rental management system to support the increasing complexity of its rental processes, instructor, plane and customer management as well as lower costs.

1.3 Business requirements

- **Information management for pilots (customers), planes and instructors.**
  The basic information of customer, planes and instructors is the foundation for a successful rental business. The automatic information management will improve the efficiency as well as reduce the manual mistakes.
  - Customers must create a new account with his information (see blew) if he is a new customer. If customers already have an account, he need to login for the identification purpose.
  - Customers can create/update his customers' basic information including name, address, phone, flying hours in each spec planes. Additionally, some information related to rental process is also need to be stored, such as pilot's number, license type, classification, instrumental rated, FAA certification (category, class).
  - System admin can create/update planes' basic information including name, model, flying hours, and capability. In addition, some information related to the rental process will also need to be stored, such as horse power, FAA certification (category, class) and location.
  - System admin can create/update instructors' information, including name, experience, and certification. Instructor is allowed to update his information.
  - System or admin can record trained history of pilot with his instructor information.
- **Rental management**
  Rental management is the process that customers request their rental service while Husky Air responses to deliver the proper plane and instructional service as promised. The systematic rental processes would give opportunities to handle more complex requests from customers. There are three sections in this requirement.
  - Customers make a rental request.

        - Customer choose the plane and instructor (optional) while the plane should be respect to his flight qualification and trained history from the instructor.
        - System matches a rental request of customer with instructor's and plane's schedule. Customer can make a successful rental request.
        - Customer/admin can update/cancel the existing rental requests. Once the request has been canceled, the booked plane or instructor is reset to available for others.
        - A success rental is result of cohesion status among customer, plane and instructor in the given day.
    - Admin can initial or update the plane schedule, system can update the plane schedule in term of planes' renting status. Also, system should record the beginning and ending hobbs for each fight.
    - Instructors manage their schedule, indicate when they are available. System should update instructor's status when he is reserved by customer.
    - After a successful rental, the customer receives a rating for the specific plane flown, this record will be used by the instructor to "checkout" the pilot/customer in another type of plane. The instructor may suggest the pilot take a lesson with a different type of plane in order to see how comfortable the pilot is while flying various planes.
- **Cost charging management**
  Cost charging is the final step of a successful rental. Customer should clearly know how much he has been charged and the detail of items.
    - Admin needs to maintain the unit price for each plane and the unit instructor price. These numbers will be shown in the system for customer checking price.
    - System calculates the total time for each flight. According to beginning and ending hobbs, price of plane, system will automatically give the total price. If customer has paid, admin will enter the receipt number.

1.4 Business value:

- Increase the usage of planes and instructors
- Extend the business to new type of planes and add the number of planes and instructors to meet the increasing requirements
- Raise the efficiency and comfort of making appointment
- System will be able to automate several tasks to save time and administrative costs
- Eventually, increase the profit and attract more customers
- System will provide easier, more thorough record keeping and ability to automatically generate reports

1.5 Special issue or constraints:

- Budget of the project: initially considering it is 2-3 person, and scope only within "business requirements" section.

- Time constraints. Since there is no existing system, more time may be needed for proper planning and analysis for the new system.
- How many departments will be involved? Typically, all the related departments should be in the project team, i.e. sales, marketing, finance, supply chain, support etc.

Section 2 Feasibility Analysis

2.1 Technical Feasibility

- Familiarity with Functional area:
  The Husky Air staff is very familiar with the functional areas and processes involved in their current system. However, the current system is a manual system, so the users do not have familiarity using any type of application or existing system to perform the processes and functions involved. The analysis process will involve working closely with the Husky Air employees that will be using the system in order to understand the business processes and other aspects the system will need to support. The functions may be changed or appended through the new system implementation, therefore, the process of understanding functions is iterative.

- Familiarity with Technology:
  This will be a new system replacing the current manual system, so it will require basic training for the Husky Air users in order to familiarize them with the system itself and how to perform the necessary functions and processes. The technology itself is not new so risks regarding familiarity with the technology are relatively low.

- Project Size:
  The project does not seem to be very large in size – the goals can be accomplished with a development team of 2-3 people. The total amount of time needed to complete the project is estimated to be 320 total hours.

- Compatibility:
  No information provided about any other existing Husky Air systems with which the new system will need to integrate with.

2.2 Economic Feasibility

Costs and Benefits

- Development Costs:
  Software development
  Hardware
  Installation
  Associated software license fee
- Operational costs:
  User training
  Hardware upgrades

Software Maintenance/License
Server/maintenance costs
Other (i.e. help-desk or supporter)

Benefits:

- Ability to manage increasing number of instructors and planes available for lessons and rentals in a much more efficient way.
- Application will be able to automate several tasks to save time and administrative costs.
- Increase the efficiency of scheduling reservations and record keeping.
- Convenient way for customers to schedule or modify/cancel an existing reservation.
    - Help avoid/reduce revenue loss from plane idle time
- Increase customer base
- Increase number of lessons booked and plane rentals.
- Extend the business to new type of planes and add the number of planes and instructors to meet the increasing requirements

Cost Benefit Analysis

	Amount
Benefits:	
Increased sales	unknown
Total Benefits	**unknown**
Development Costs:	
Hardware	$4,000
Software development*	$32,000
License fee	$2,000(TBD)
Total Development Costs	**$38,000**
Operational Costs:	
User training	$2,000
Maintenance	$2,400
Server	$3,600
Others	$2,000
Total Operational Costs	**$10,000**
Total of All Costs	**$48,000**

*320 total hours at $100/hour, includes time for installation

2.3 Organizational Feasibility

- Is the project strategically aligned with the business?

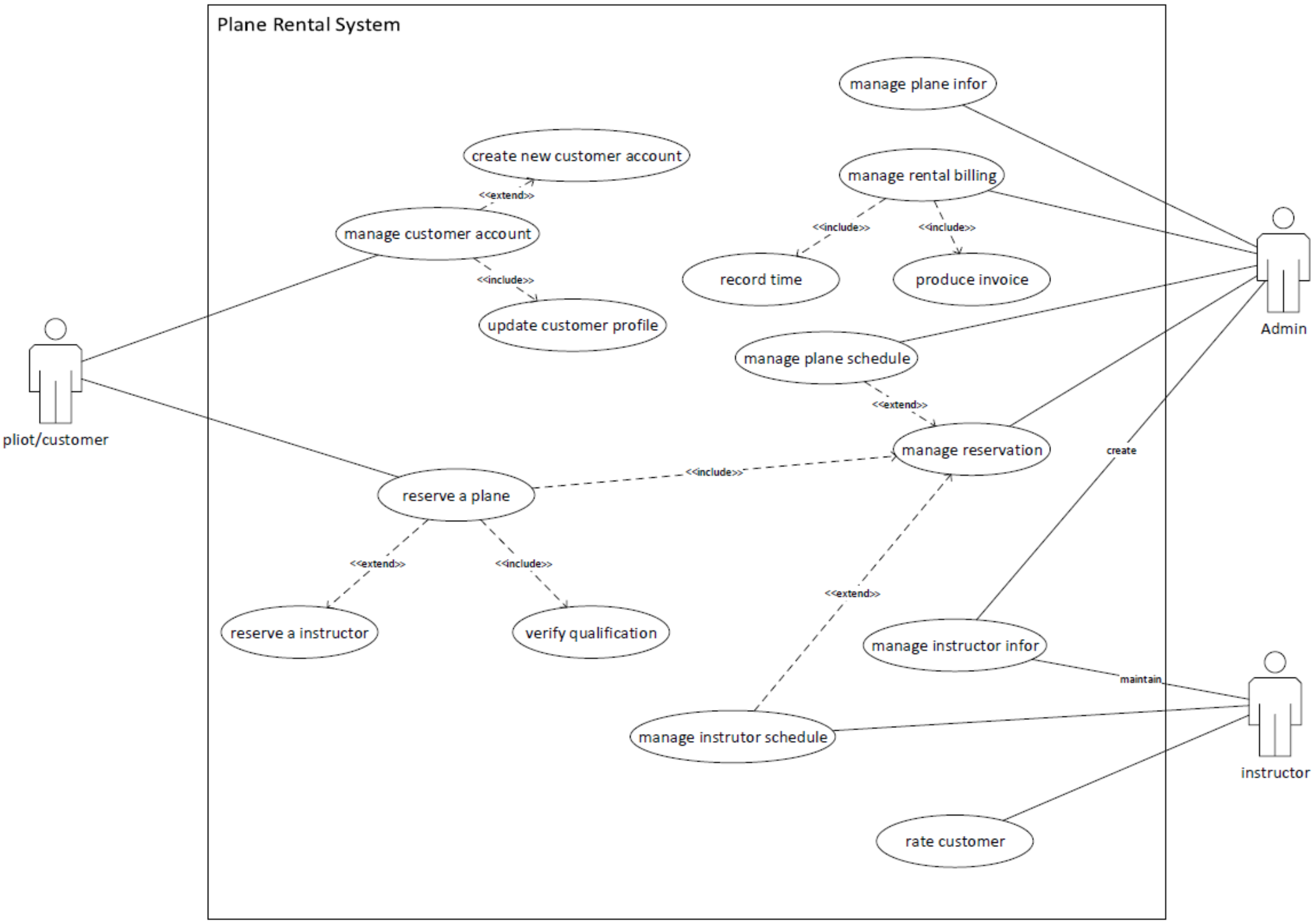
This project is strategically aligned with the business objectives for Husky Air. The new system can greatly improve efficiency of the business processes currently being performed through the manual system, can reduce the time needed to perform tasks, and will increase data and record keeping capabilities. Husky Air wants to improve scheduling and make more planes and instructors available to customers and the new system will enable them do that. The system will also help Husky Air increase its customer base and sales of rental planes and instructor lessons.

- Users/Stakeholders
    - Project champion(s): Husky Air management
    - Users: the system users include Husky Air office staff responsible for managing the reservations and rentals of the planes.

Section 3 System Functional Analysis

3.1 Use Case Diagram

The use case diagram gives an overview of the activities involved in the plane rental system. There are three main actors: customer, system admin and instructor. Customer creates an account to order a rental service; system admin and instructor delivery the rental service to the customer on a scheduled time. The purpose of this diagram is to illustrate all the ways each of the actors can interact with a system. Following chart you can find the actions of three major use cases: reserve a plane, manage reservation and manage rental billing. The first two use cases show how the actors cooperate with each other to make a successful rental. The last one achieves charge purpose. A successful rental service should be an agreed time within customer, plane and instructor, in case one of them changes schedule, another one should correspond change.



[figure 3-1]

3.2 Use Case Description

- **Reserve a plane:** This use case depicts how a customer can reserve a plane as well as update or cancel a plane rental service. It contains the use cases “verify qualification” and “manage reservation”, and relates to use case “reserve an instructor”.

User Case Name: reserve a plane	**ID 1**	**IM Level**: high
Primary Actor: customer	**Use Case Type**: detail, essential	
Stakeholders and Interests: Customer - create, update, cancel a rental reservation Instructor – ensure customer get consult on schedule Admin – ensure customer get rental plane on schedule		
Brief Description: this use case depicts how customers reserve a plane as well as update or cancel a plan rental service		
Trigger: customer logs in to the system to request rental service or update/ cancel a rental service **Type**: aggregation		
Precondition: 1. customer already has an account with essential information		

2. plane and instructor essential information exists
3. plane and instructor schedule is current

Relationships:
Association: customer
Include: manage reservation, verify qualification
Extend: reserve instructor
Generalization:

Normal Flow of Events:

1. customer logs in to the system in order to make a reservation (or calls system admin, then system admin would play a role as customer)
2. customer chooses the plane model appropriate for his qualification: S-1 verify qualification
3. customer chooses rental day according to plane schedule: S-2 manage reservation
4. if customer asks for an instructor, he needs to choose instructor from available instructors list: S-3 reserve instructor
5. customer can update or cancel the rental reservation
   if updating a rental reservation: S-2-1 needs match with plane and instructor schedules, and update their schedule as well.
   if canceling a rental reservation: S-2-2 needs to notify admin and instructor and update their schedule

SubFlows:
S-1: verify qualification (need to be discussed with stakeholders)

1. Based on customer’s FAA certification, customer can only fly a group of planes which matches his category.
2. Based on customer’s training history of high performance plane, customer would play a plane over 200 horsepower.

S-2: manage reservation (Use Case Description ID=2)

1. Update rental reservation
2. Cancel rental reservation

S-3: reserve instructor

1. According to the plane reservation date, customer can find all the instructors available that day.
2. Customer chooses the instructor

Alternate/Exceptional Flows:
S-1a: if customer prefers a high-performance plane, he needs to join a ground and flight training
S-3a: if there is no instructor available on chosen date, customer needs to change plane date or cancel reservation.

[Table 3.2-1]

- **Manage rental reservation**: This use case depicts how a customer or system admin manages a rental reservation; and how the changes within the plane or instructor schedules lead to updating a customer reservation

User Case Name: manage rental reservation	**ID 2**	**IM Level**: high
Primary Actor: customer, system admin	**User Case Type**: detail, essential	

Stakeholders and Interests:
Customer – schedule or cancel a reservation
Instructor – change instructor schedule may affect current reservation
Admin – change plane schedule may affect current reservation or admin cancels reservation directly

Brief Description: this use case depicts how a customer user or system admin manages a rental reservation; and how the changes within the plane or instructor schedules lead to updating customer reservation.
Trigger: customer/admin changes the rental date; customer/admin cancel a rental; plane or instructor schedule changed by admin or instructor; **Type**: included or extended
Precondition: 1. a reservation already exists 2. plane or instructor changes the date which already had been reserved
Relationships: Association: reserve a plane, manage plane schedule, manage instructor schedule, system admin Include: Extend: Generalization:
Normal Flow of Events: 1. customer reserves a plane (Use Case Description ID=1) 2. plane and instructor schedule marked reserved on the flight date 3. customer reschedules a plane: S-1 reschedule a plan 4. customer/admin cancels a plane: S-2 cancel a reservation 5. system admin changes plane schedule or instructor changes his schedule if changed date is reserved by customer, the original reservation will be canceled, and customer will receive a notification. 6. after a successful flight, system admin will know this rental event is completed.
SubFlows: S-1 reschedule a plan 1. customer chooses plane and date with instructor 2. customer makes a new rental reservation (Use Case Description ID=1) 3. plane and instructor are available on the previous reserved date S-2 cancel a reservation 4. customer or system admin find an existing reservation 5. customer or system admin cancel the existing reservation 6. plane and instructor are available on the previous reserved date
Alternate/Exceptional Flows: S-1a: if customer fails to reschedule due to no plane or instructor available, customer can only keep or cancel original plan. S-1b: if system admin or instructor cancels the date which has been reserved by customer, this reservation will be canceled or rescheduled.

[Table 3.2-2]

- **Manage a rental billing**: This use case depicts how the system admin calculates the bill and creates an invoice after one (or more) successful plane rentals.

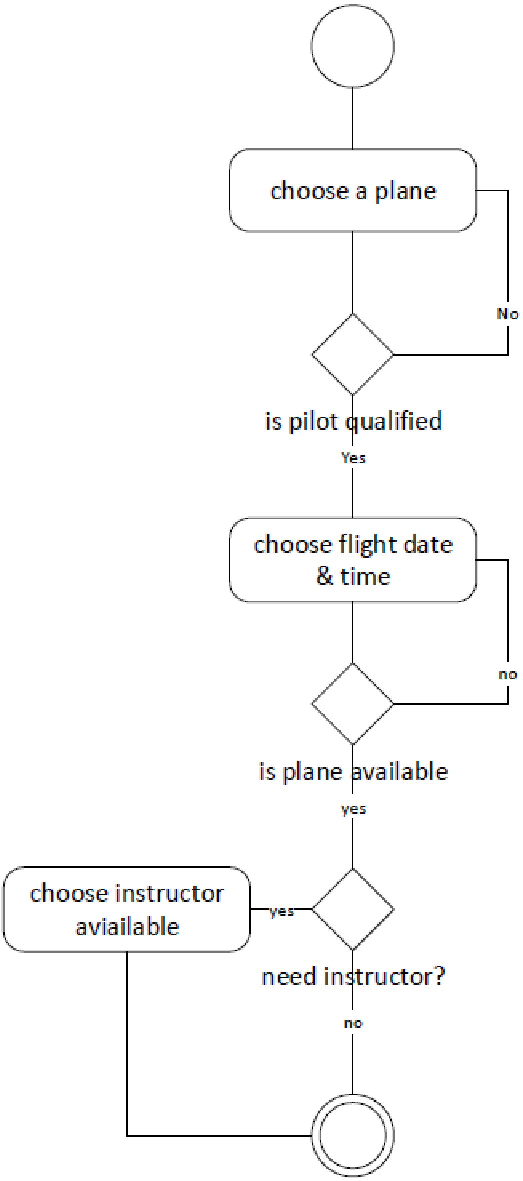
<table>
<tr><td colspan="2"><b>User Case Name: manage a rental billing</b></td><td colspan="2"><b>ID 3</b></td><td><b>IM Level</b>: high</td></tr>
<tr><td colspan="3"><b>Primary Actor</b>: system admin</td><td colspan="2"><b>User Case Type</b>: detail, essential</td></tr>
<tr><td colspan="5"><b>Stakeholders and Interests</b>:
Admin – calculate the bill for each order and create invoice
Customer – pay bill and receive invoice</td></tr>
<tr><td colspan="5"><b>Brief Description</b>: this use case depicts how system admin calculates the bill and creates an invoice after one (or more) successful plane rentals.</td></tr>
</table>

Trigger: customer completes order activities which includes one or more flights, and records all the flights hobbs times into system by system admin. **Type**: aggregation
Precondition: 1. all the flights in an order have been completed 2. customer or system admin cancels order
Relationships: Association: system admin Include: record time, produce invoice Extend: Generalization:
Normal Flow of Events: 1. after each successful flight, customer submits his flight hobb to admin 2. admin records the hobb and calculate the total price for each rental. S-1 3. after all rental completed in an order, create the bill and invoice. S-2 4. send the bill and invoice to customer 5. customer pay the bill, and the bill marked paid
SubFlows: S-1 record hobbs time 1. before/after each flight, customer writes down the start time and end time 2. customer submits the time record to admin, admin records it into system 3. each total price equals hobbs time * unit price 4. after all events in an order are completed, all costs are totaled to obtain the total bill price for a customer S-2 create invoice 1. admin or system gets the detail items for each flight 2. admin or system gets customer business information 3. create invoice(s) for a completed order 4. print receipt
Alternate/Exceptional Flows: S-2a: customer does not require receipt, then S-2-4 is skipped

[Table 3.2-3]

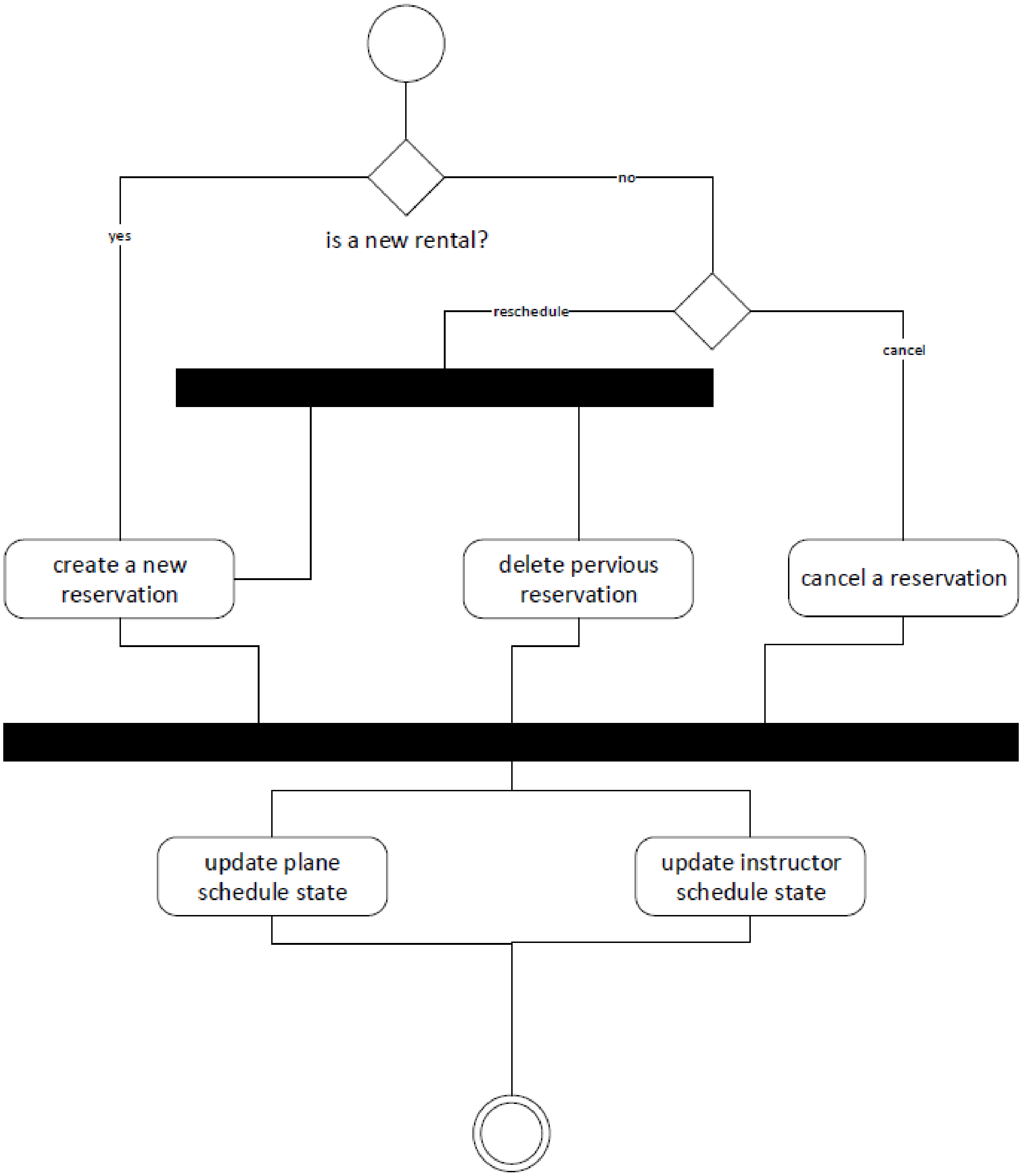
3.3 Activity Diagram

- **Reserve a Plane**: This activity diagram depicts the process of how a customer reserves a plane and an instructor (optional). The purpose of an activity diagram is to show the activities performed by the system as well as the decisions being made during a process. The activity diagram depicts the activity flow and sequence from one activity to another. More detail regarding the process is found in the associated Use Case Description ID=1.



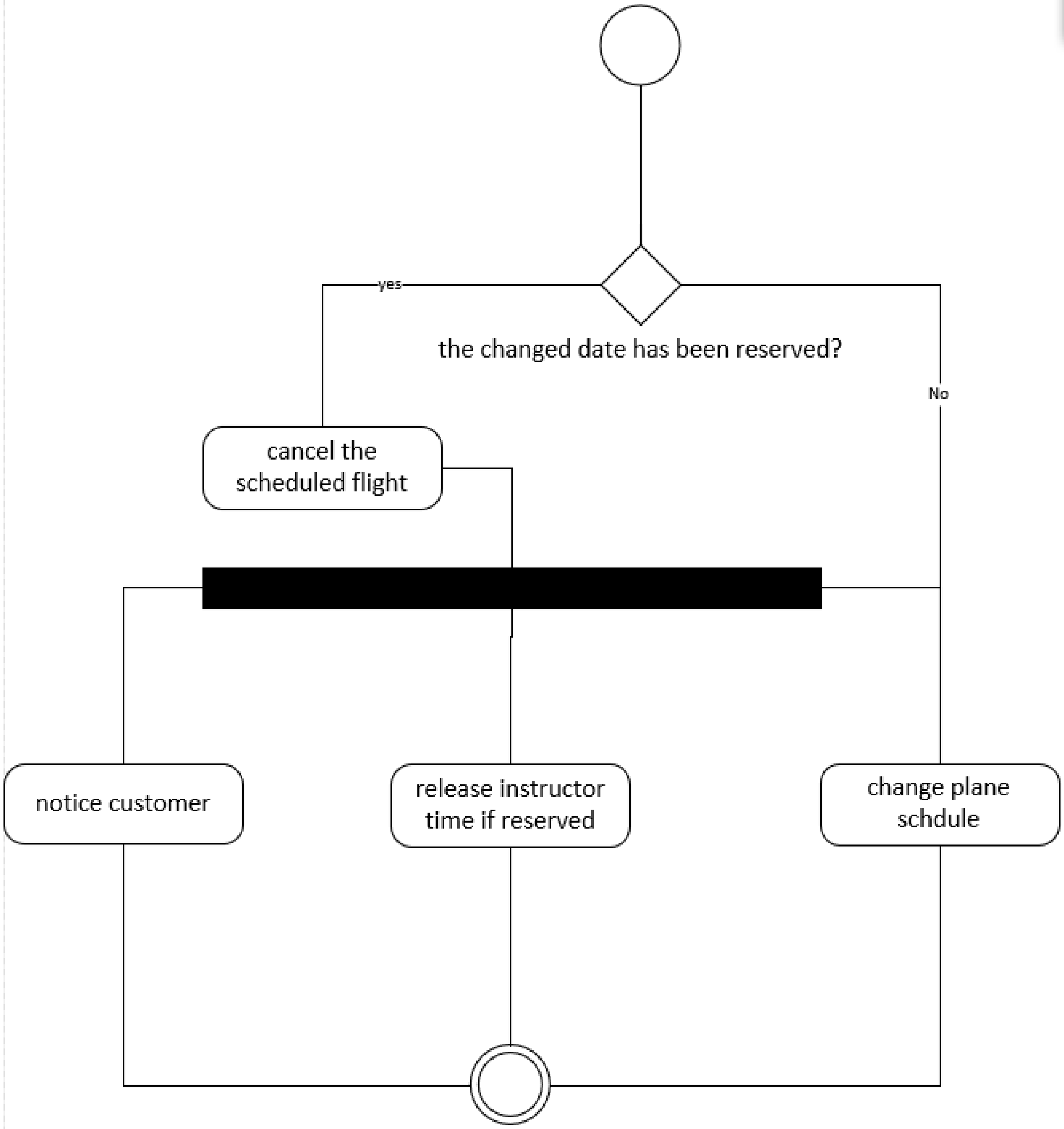
[figure 3.3-1]

- **Manage a reservation**: This activity diagram depicts the process how customer or system admin manage a rental reservation. More detail regarding this process can be found in Use Case Description ID=2.


is a new rental?
yes
no
reschedule
cancel
create a new reservation
delete pervious reservation
cancel a reservation
update plane schedule state
update instructor schedule state

[figure 3.3-2]

- **Change Plane schedule**: this activity diagram depicts how interplay between plane schedule and scheduled flight. If the plane schedule change relates to reserved time, the original flight will be canceled as well as notice customer and release time of instructor.

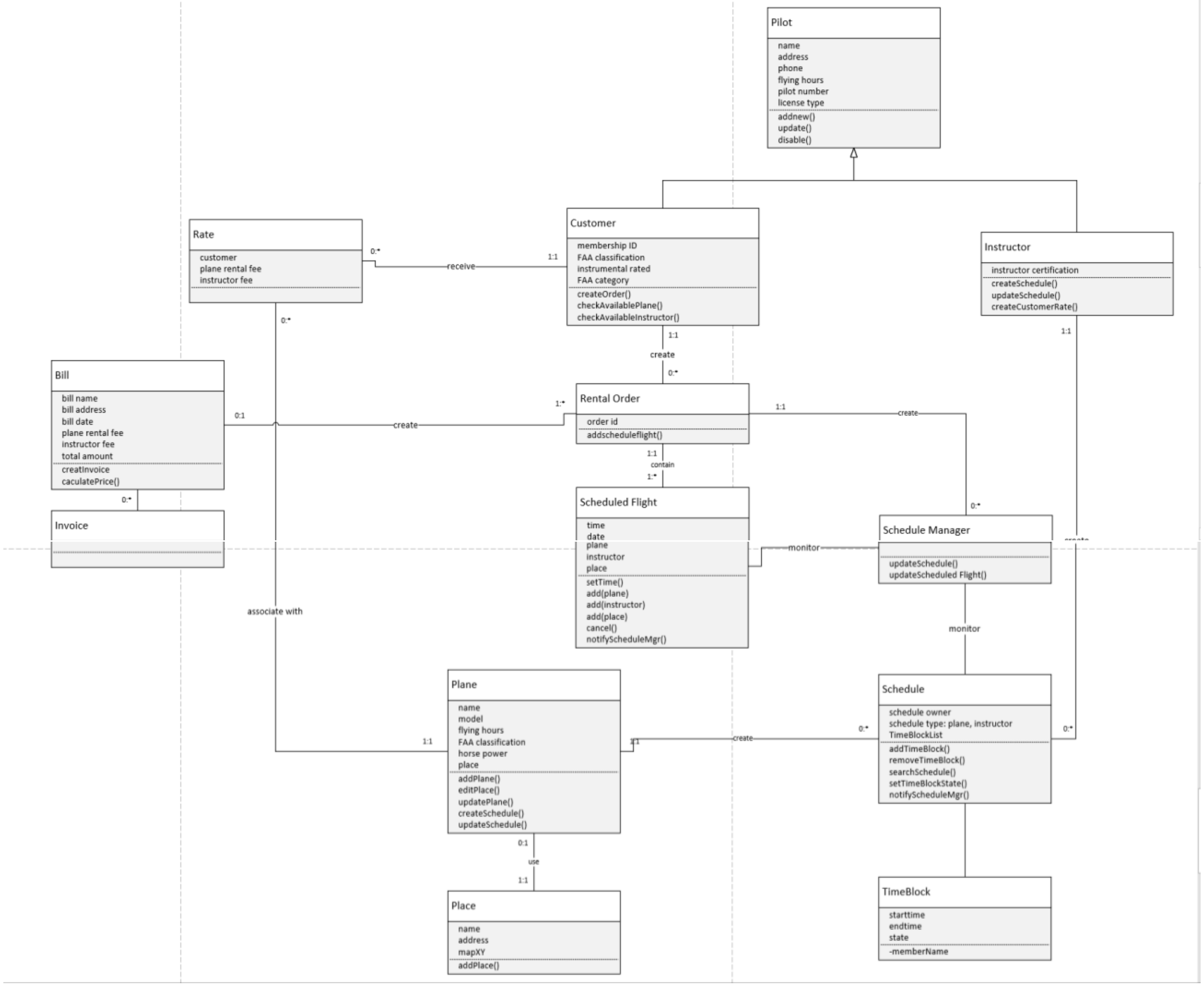

yes
the changed date has been reserved?
No
cancel the scheduled flight
notice customer
release instructor time if reserved
change plane schdule

[figure 3.3-3]

Section 4 System Structure Analysis

4.1 Class Diagram

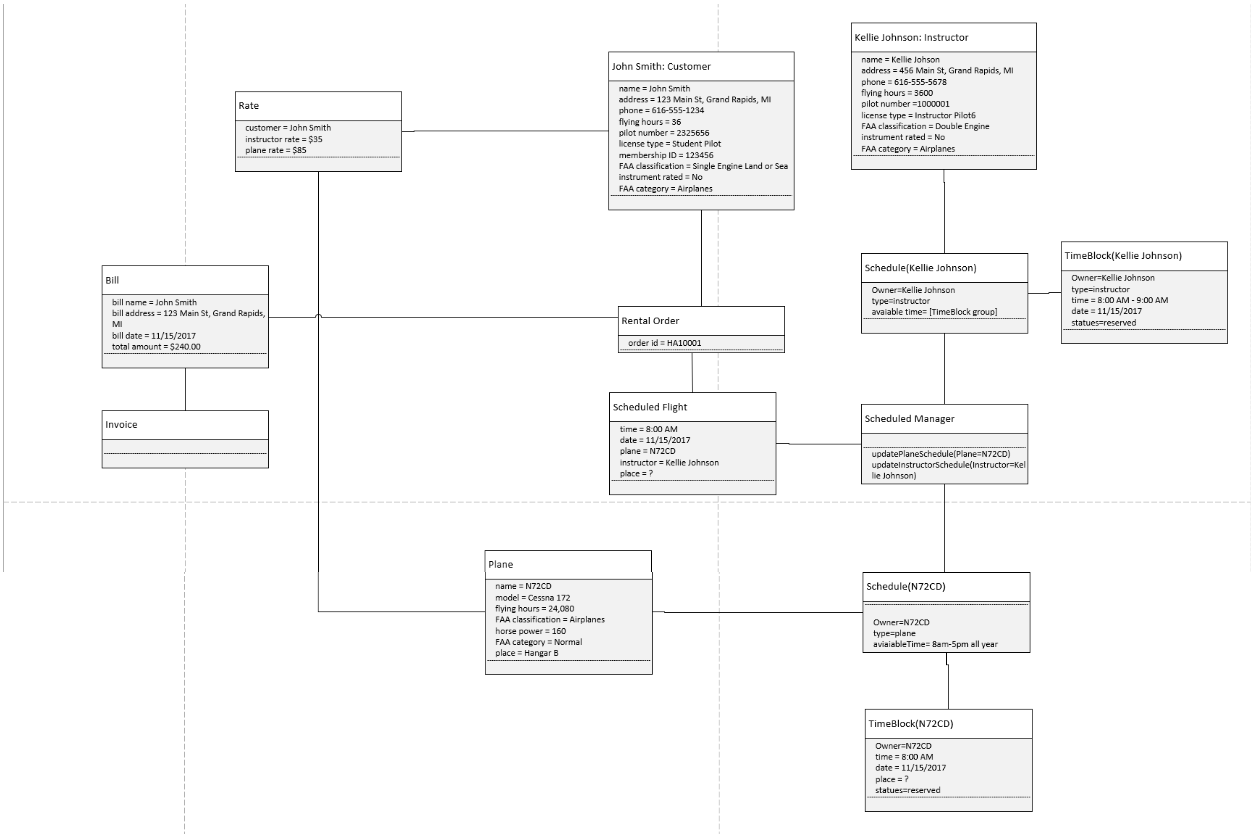
In the following class diagram, class "customer" and "instructor" inherit from a superclass "pilot" which contains the basic information of pilot. A "customer" can create an "order" with one or many " scheduled flight". A "schedule" is created seperately for each "instructor" and "plane"(they publish their own aviable time). Class "schedule manager" is responsible for managing the state changes within "scheduled flight" or "schedule" for both "planes" and "instructors" in case a change in one of them will interaffect. A "Bill" is created by "Order" after a successful rental, and caculates the total price and invoice.



[figure 4.1-1]

4.2 Object Diagram

This object diagram shows an instance where a customer “John Smith” orders a flight with the plane “N72CD” and an instructor “Kellie Johnson” on Nov, 15th, 2017. The object diagram is used to show a snapshot of the detailed state of a system at a given moment in time. The object diagram includes all of the data and data values that a system will contain and can help show the data structure.

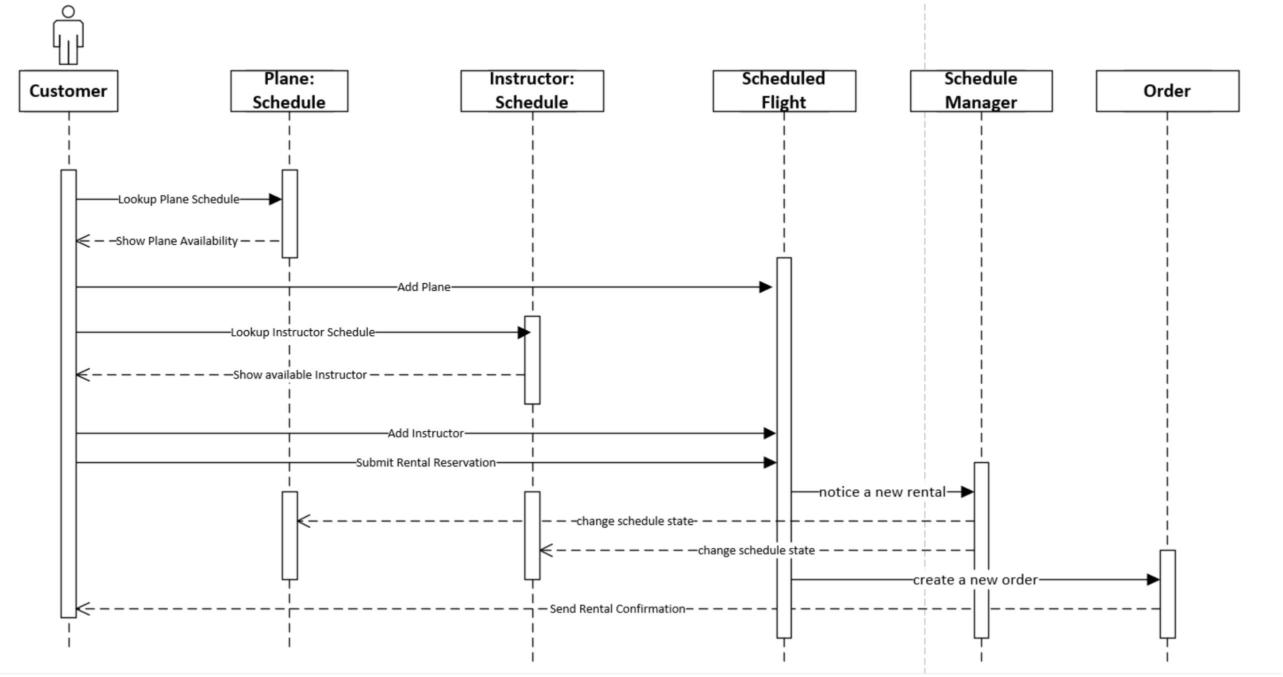


[figure 4.2-1]

Section 5 System Behavior Analysis

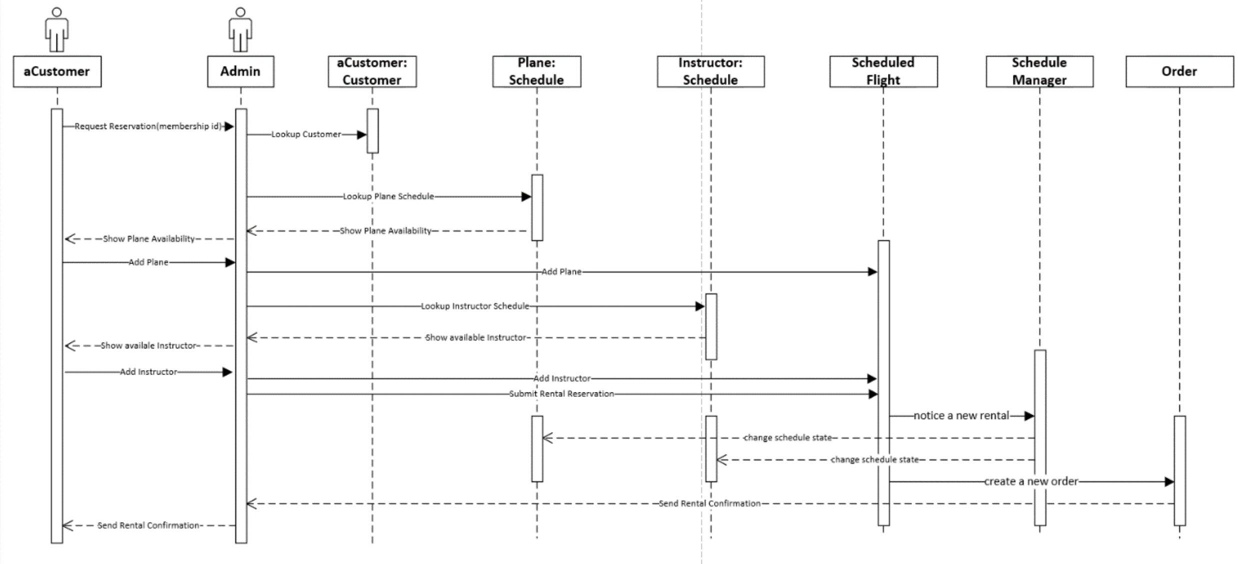
5.1 Sequence Diagram

- Reserve a plane (Customer)
  This sequence diagram depicts the process when customer reserve a plan by himself through website.



[figure 5.1-1]

- Reserve a plane (Admin)
  This sequence diagram depicts the process when a customer reserves a plane by admin through phone or email.

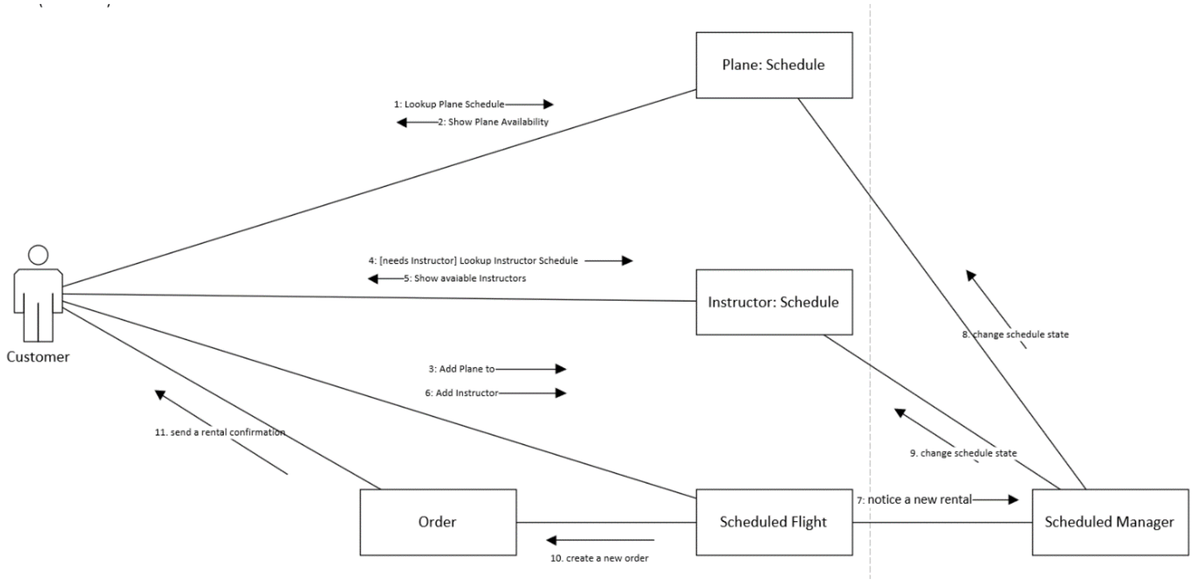


[figure 5.1-2]

5.2 Communication Diagram

- Reserve a plane (Customer)

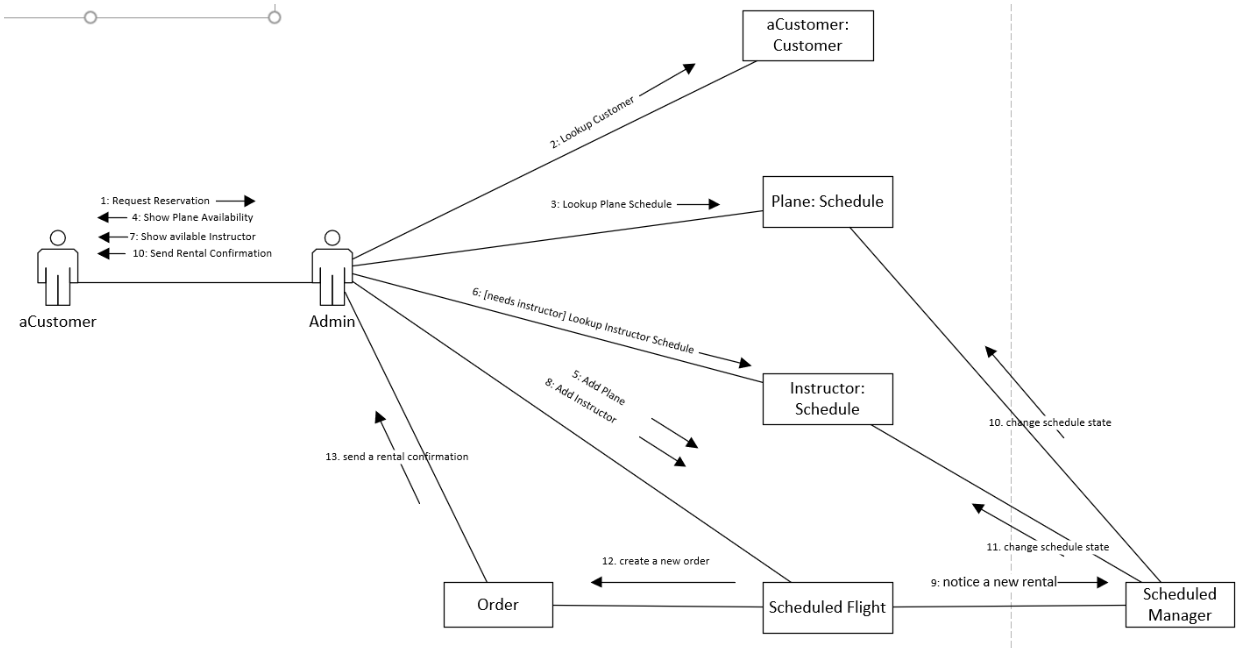
This communication diagram corresponds to the sequence diagram for the use case Reserve a plane (customer) [figure 5.1-1].



[figure 5.2-1]

- Reserve a plane (Admin)

This communication diagram corresponds to the sequence diagram for the use case Reserve a plane (admin) [figure 5.1-2].

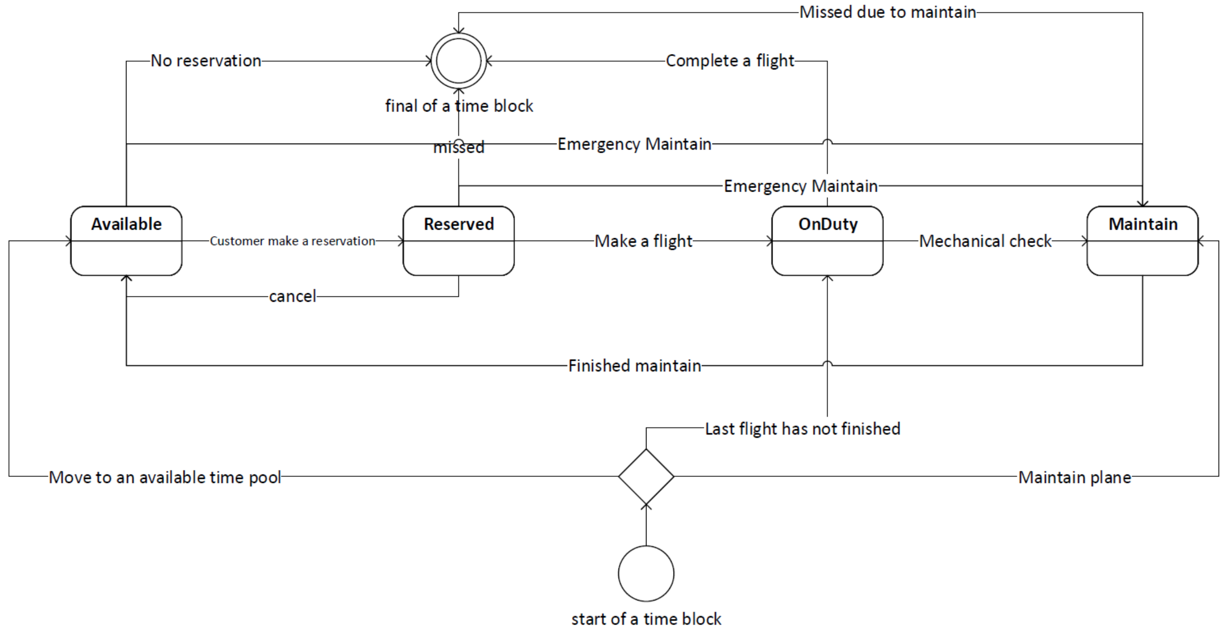


[figure 5.2-2]

5.3 State Machine

Plane TimeBlock: a schedule consists of many time blocks, in each time block, plane has four states: available = waiting for reservation; reserved = already reserved by customer; onDuty = plane is on air; maintain = plane is under maintenance.

The initial states will be one of “available”,”onduty” or “maintain”. Since time passed over a specific time block, this state machine went to end.



[figure 5.3-1]

Appendix A – Work Breakdown Schedule

WBS for project "Airplane Rental Management System"

- **Analysis**

1.1. Analysis Plan [Initial Group Task - John, Jie] Week1

- Review the business requirements
- Clarify any assumptions and unknowns
- Prepare initial JBGE analysis
- Assign tasks in term of WBS

1.2. Functional Modeling [control by Jie] Week 1

- Identify user-cases – John, Jie
- Use-case diagram - Jie
- Use-case description(s) - Jie
- Activity diagrams – Jie
- Balance functional diagram - Jie
- Review/update JBGE analysis – John, Jie

1.3. Structural Modeling [control by John] Week2

- Identify classes [partial parallel with "identify user-case"] - John, Jie
- Create class diagram - Jie
- Create object diagram(s) - John
- Balance structural & functional diagram - John
- Review/update JBGE analysis - John, Jie

1.4. Behavioral Modeling [control by John] Week2

- Create Sequence diagrams - John
- Create comms diagrams - John
- Create state/transition diagrams [parallel with "class diagram"]- Jie
- Verify/validate STDs – John
- Balance all diagrams – John, Jie

1.5. Create project analysis report document – Jie Week3

1.6. Review analysis report document – John, Jie Week3

1.7. Submit final version of analysis report document – John Week3

Appendix B – JBGE Analysis

In this project, our purpose of creating documents is to keep the ones which are reviewed by customers or team members, and identify those that need to be updated in the iterations (we don't want to restart these logics again and again). Therefore, we created a use case diagram and class diagram for overview and explained the complex points inside by another diagram (see below). Two models we didn't create: CRC Cards and CRUDE analysis, because the information had been sufficiently covered through other models.

Deliverable	JBGE Rationale	Plan	Actual
FUNCTIONAL			
Use Case Diagram	Required. Basic tool to communicate with customers and team members in order to eliminate misunderstanding. And it is easy to update but important to the following jobs.	1	1
Use Case Description(s)	Required. Make sure our understanding for complex use cases is the same with customers in detail. The complex use cases are: "reserve a plane, manage reservation, manage rental billing and manage plane/instructor". They all contain subflows or are related to another complex user case. Others are simple or common functions in a typical system.	4	3
Activity Diagram(s)	Required. Three complex use cases have one or more decision logic, so we drew the activity diagrams to clarify them, which includes: "reserve a plane, manage reservation and manage schedule"	3	3
STRUCTURAL			
Class Diagram	Required. It converts the use cases to structural level for team members (or IT dep). Because we only focus on high level at this moment, we will miss some attributes or methods in detail. And we didn't consider design level work such as interface and UI class.	1	1
CRC Card(s)	Not required. Walking through the use case diagram is a good way to find out if all use cases and relationships are covered by class list on paper.	0	0
Object Diagram(s)	Required. Object diagram is related to real world, we drew one object diagram under a typical situation in case we missed some classes.	1	1
BEHAVIORAL			
Sequence Diagram(s)	Required. But we only need to create complex ones related to the activity diagrams.	2	2
Comms Diagram(s)	Required. Only need to create ones related to sequence diagram to find the information communication between actor and classes.	2	2
State/Transition Diagram(s)	Required for those classes that must maintain state. There are two complex state machines for plane and instructor in each time block.	2	1
CRUDE Analysis	Not Required. Both customer and team member wouldn't review it. So we did it on the paper.	0	0