Version 1.0 – Dec 6, 2017

System analysis and design

Husky airplane rental management system

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 |
| 0.3 | 11/16/17 | Function Diagram Refine | Add a basic description about account management |
| 0.9 | 12/5/17 | Design and Non-functional section | Initial Creation |
| 1.0 | 12/6/17 | Minor updates |  |

Contents

[Section 1 General System Requirements 5](#_Toc500320530)

[1.1 Project Sponsor 5](#_Toc500320531)

[1.2 Business need 5](#_Toc500320532)

[1.3 Business requirements 5](#_Toc500320533)

[1.4 Business value: 6](#_Toc500320534)

[1.5 Special issue or constraints: 6](#_Toc500320535)

[Section 2 Feasibility Analysis 7](#_Toc500320536)

[2.1 Technical Feasibility 7](#_Toc500320537)

[2.2 Economic Feasibility 7](#_Toc500320538)

[2.3 Organizational Feasibility 8](#_Toc500320539)

[Section 3 System Functional Analysis 10](#_Toc500320540)

[3.1 Use Case Diagram 10](#_Toc500320541)

[Section 4 System Structure Analysis 19](#_Toc500320542)

[4.1 Class Diagram 19](#_Toc500320543)

[4.2 Object Diagram 20](#_Toc500320544)

[Section 5 System Behavior Analysis 21](#_Toc500320545)

[5.1 Sequence Diagram 21](#_Toc500320546)

[5.2 Communication Diagram 22](#_Toc500320547)

[5.3 State Machine 23](#_Toc500320548)

[Section 6 Data Management Layer Design 23](#_Toc500320549)

[6.1 Object Persistence Format 23](#_Toc500320550)

[6.2 Data Access and Manipulation Classes 24](#_Toc500320551)

[Section 7 HCI Design 24](#_Toc500320552)

[7.1 Use scenarios. 24](#_Toc500320553)

[7.2 Windows Navigation Diagrams 25](#_Toc500320554)

[7.3 Storyboard 27](#_Toc500320555)

[Section 8 Physical Architecture Design 30](#_Toc500320556)

[8.1 System Deployment Diagram 30](#_Toc500320557)

[8.2 Non-functional requirements 30](#_Toc500320558)

[Section 9 Conversion Plan & Change Management 32](#_Toc500320559)

[9.1 Conversion Process 32](#_Toc500320560)

[9.2 Conversion Evaluation 33](#_Toc500320561)

[9.3 Change management plan 34](#_Toc500320562)

[9.3 Pilot Evaluation 36](#_Toc500320563)

[Appendix A – Work Breakdown Schedule 37](#_Toc500320564)

[Appendix B – JBGE Analysis 38](#_Toc500320565)

**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

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

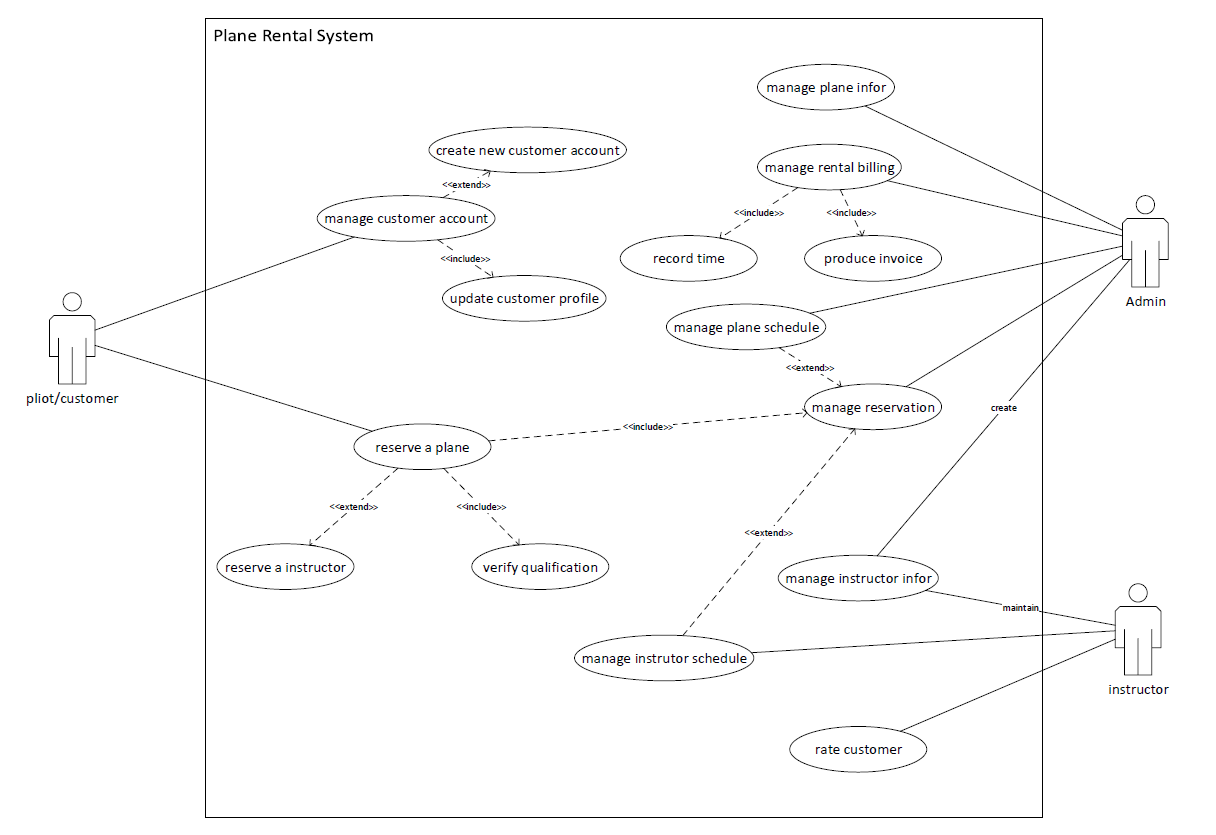
* Users/Stakeholders
  + Project champion(s): Husky Air management
  + Users: the system users include Husky Air office staff responsible for managing the reservations and rentals of the planes.

**Section 3 System Functional Analysis**

**3.1 Use Case Diagram**

Excepted system will contain four major functions: manage profile, rental plane, manage schedule and charge rental billing. There are also some sub-functions in each major function respectively, e.g. update customer profile, reserve a plane, change plane schedule and record time.

Use case diagram [figure 3-1] gives an overview of the activities involved in these functions. 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. 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 plane 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 fly 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 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 reservation** | **ID 2** | | **IM Level**: high |
| **Primary Actor**: customer, system admin | | **Use 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.   1. after a successful flight, system admin will know this rental event is completed. | | | |
| **SubFlows**:  S-1 reschedule a plane   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   1. customer or system admin find an existing reservation 2. customer or system admin cancel the existing reservation 3. 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 plane.  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.

|  |  |  |  |
| --- | --- | --- | --- |
| **User Case Name: manage rental billing** | **ID 3** | | **IM Level**: high |
| **Primary Actor**: system admin | | **Use Case Type**: detail, essential | |
| **Stakeholders and Interests**:  Admin – calculate the bill for each order and create invoice  Customer – pay bill and receive invoice | | | |
| **Brief Description**: this use case depicts how system admin calculates the bill and creates an invoice after one (or more) successful plane rentals. | | | |
| **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 hobbs time to admin 2. admin records the hobbs time and calculates 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]

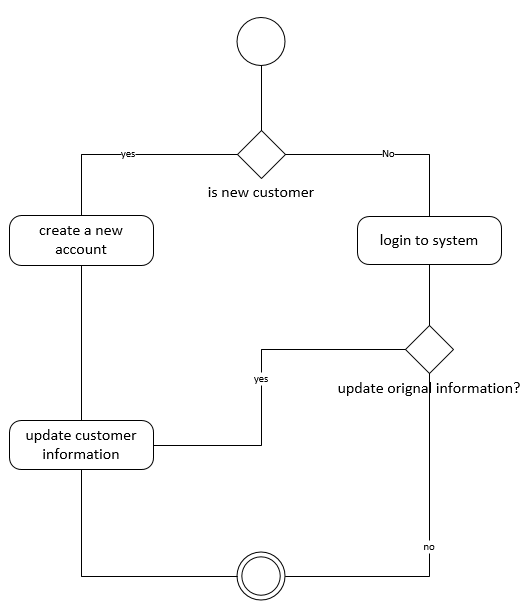
* **Manage customer account:** This use case describes how an existing customer logs in to the system in order to view or update the information contained in the customer’s account profile. This use case is extended by the create new customer account use case for instances where a new customer would like to setup an account.

|  |  |  |  |
| --- | --- | --- | --- |
| **User Case Name: manage customer account** | **ID 4** | | **IM Level**: high |
| **Primary Actor**: customer | | **Use Case Type**: detail, essential | |
| **Stakeholders and Interests**:  Customer – existing and new customers | | | |
| **Brief Description**: this use case depicts how a user can login to the system and access their customer profile in order to view or update their information. The use case can be extended by the create new customer account use case for customers that do not have an existing account. | | | |
| **Trigger**: a new or existing customer wants to log in to the Husky Air rental website | | | |
| **Precondition:**   1. Customer navigates to login page for Husky Air rental system | | | |
| **Relationships**:  Association: customer  Include: update customer profile  Extend: create new customer account  Generalization: | | | |
| **Normal Flow of Events**:   1. Customer logs in to system with username and password 2. Customer views account profile information | | | |
| **SubFlows**:  S-1 Create new customer account   1. Customer signs up for new account 2. Creates username/password 3. Logs in to system with username/password 4. Updates customer information for profile   S-2 Update customer profile   1. Customer updates profile information | | | |
| **Alternate/Exceptional Flows**:  1a. Username and/or password is invalid, system displays an error message and returns to login screen | | | |

[Table 3.2-4]

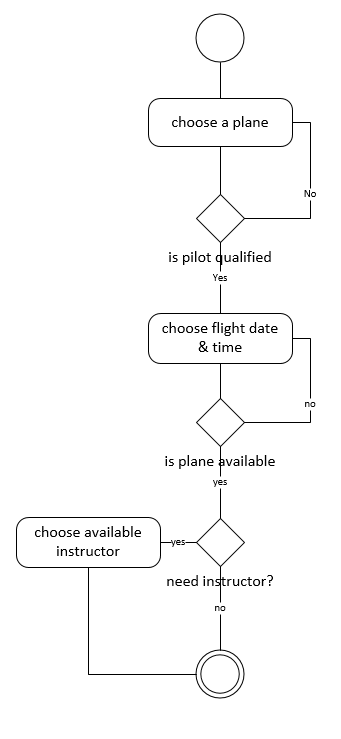
**3.3** **Activity Diagram**

* **Manage User Account**: This activity diagram shows how a customer manage his account. If a new customer tends to rental plane, he has to create a new account and upload some basic information for reference. If he already owns an account, then he can choose to update his information. But some of information needs to be approved by admin before active. More detail for this process can be found in Use Case Description ID=4.



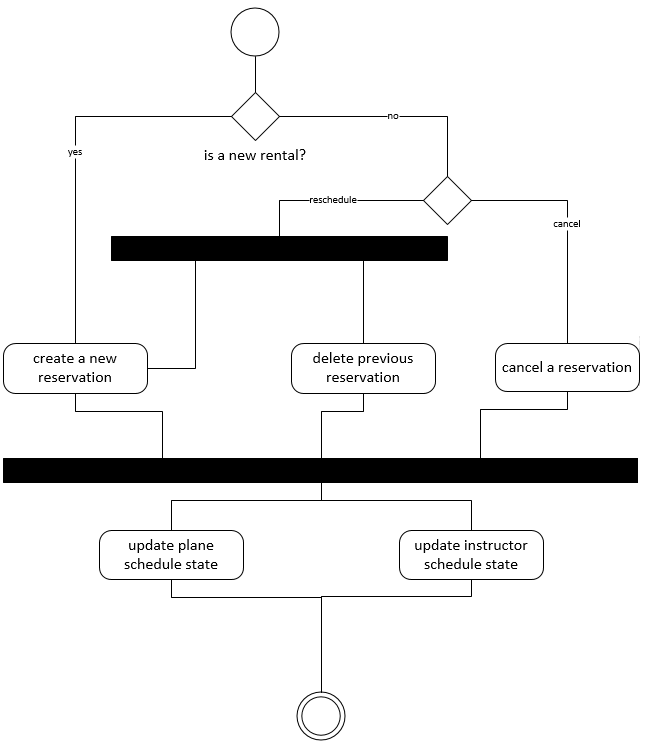
[figure 3.3-1]

* **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 this process is found in the associated Use Case Description ID=1.

****

[figure 3.3-2]

* **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.

****

[figure 3.3-3]

* **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.

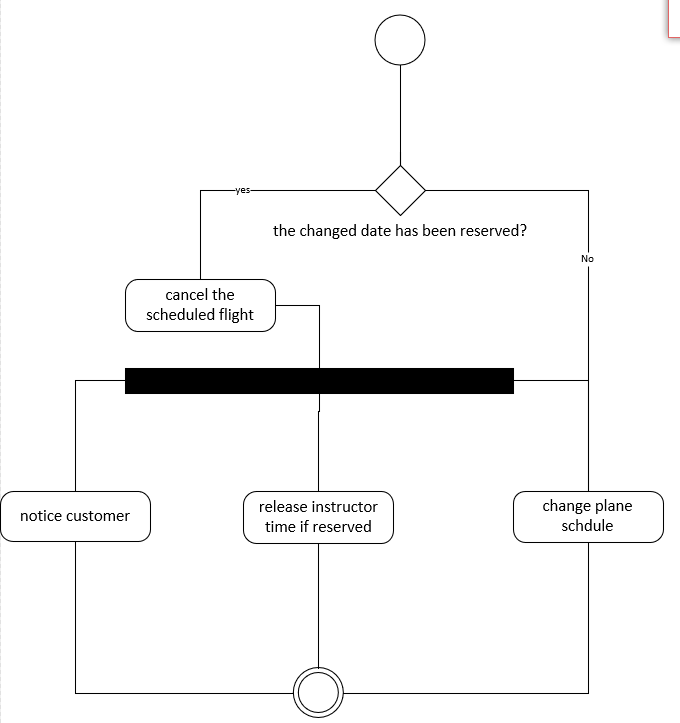
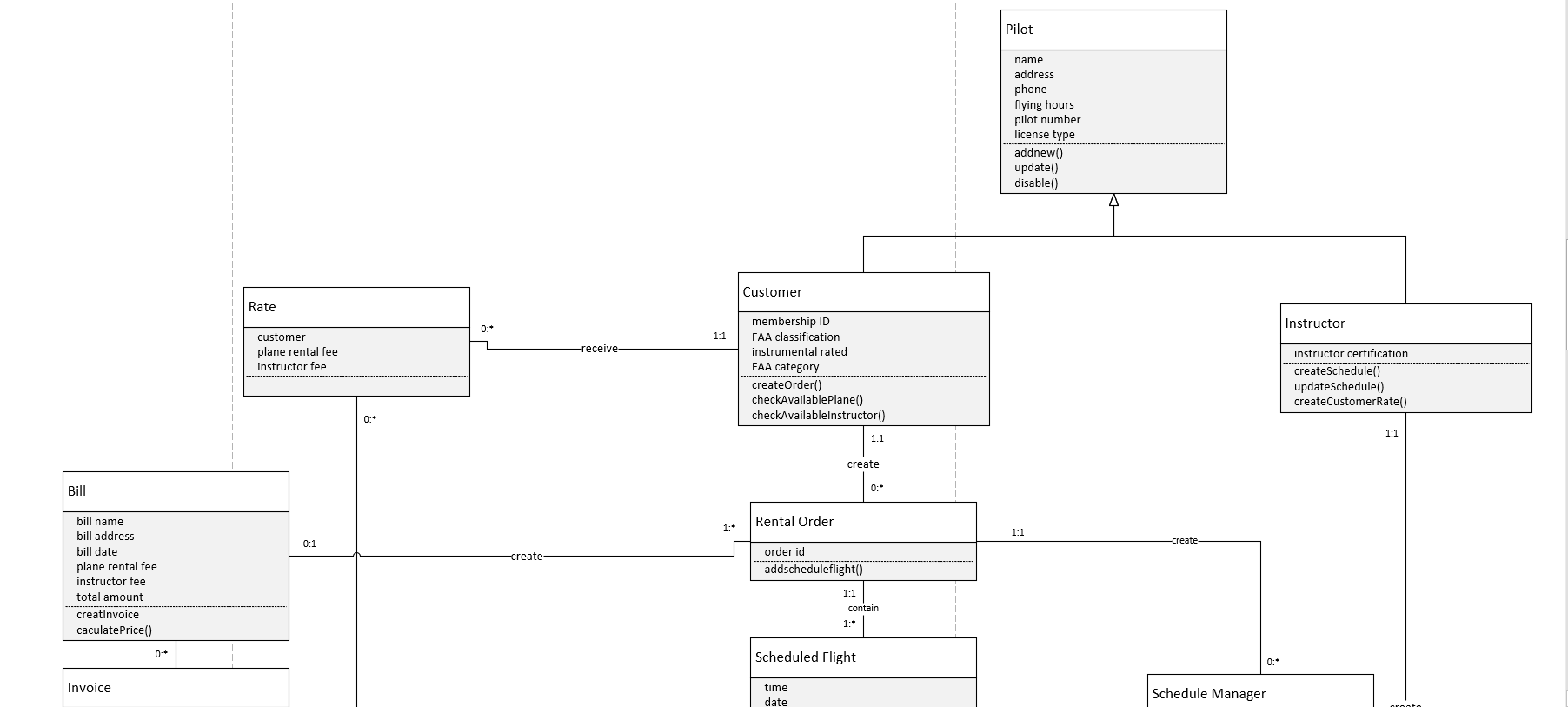


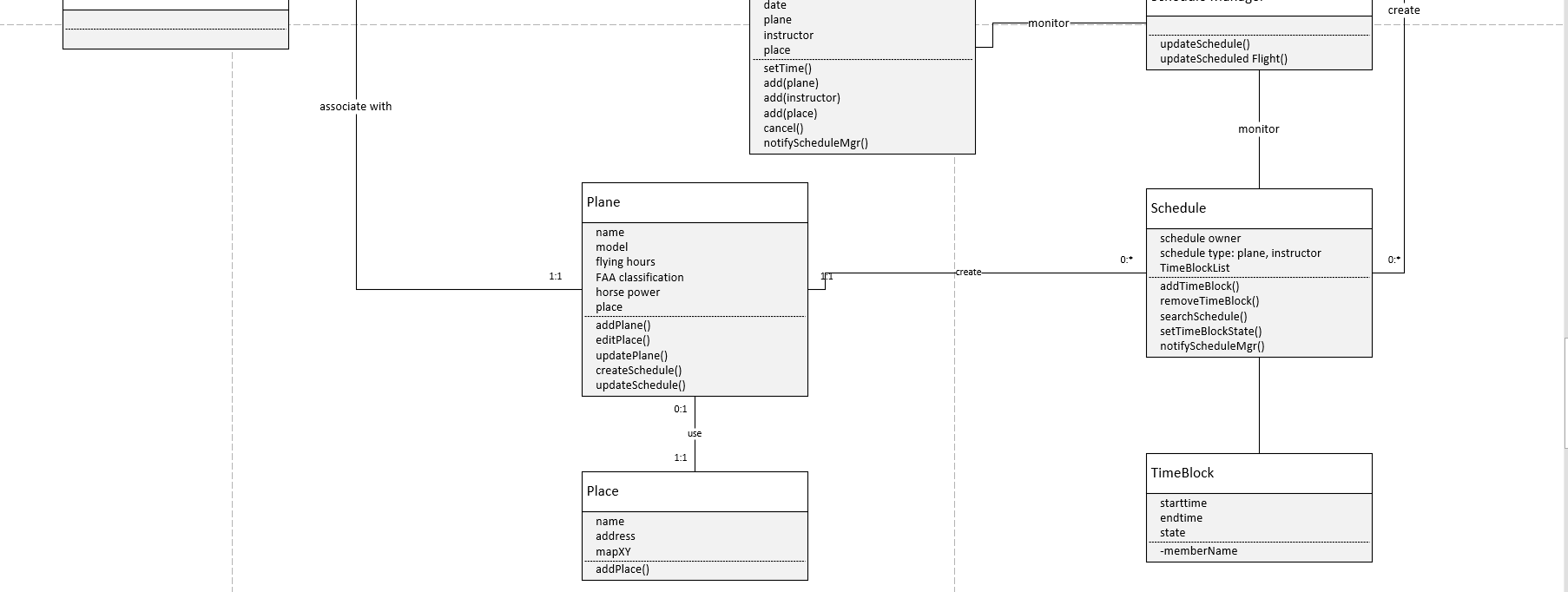
figure 3.3-4]

**Section 4 System Structure Analysis**

* 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 available 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 affect another. A “Bill” is created by “Order” after a successful rental, and caculates the total price and invoice.

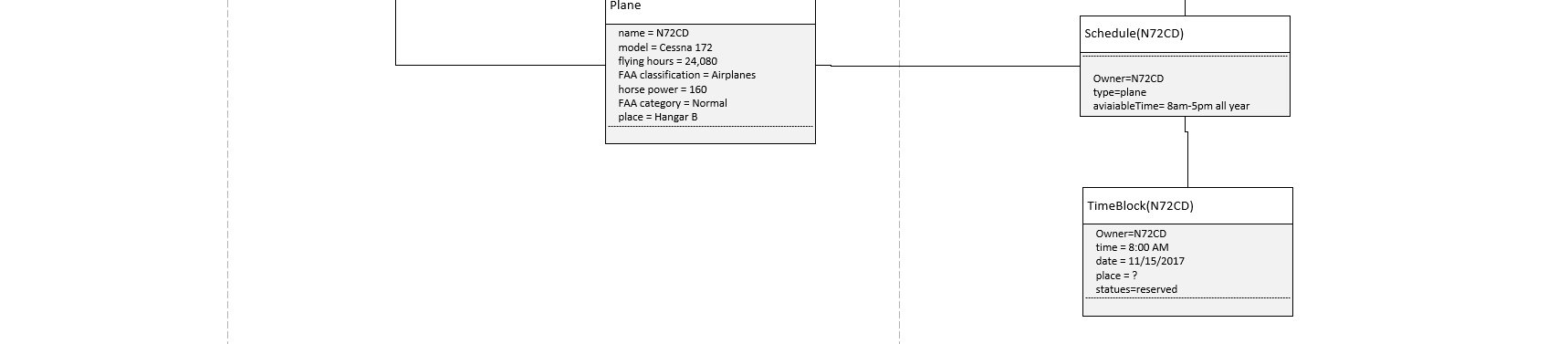
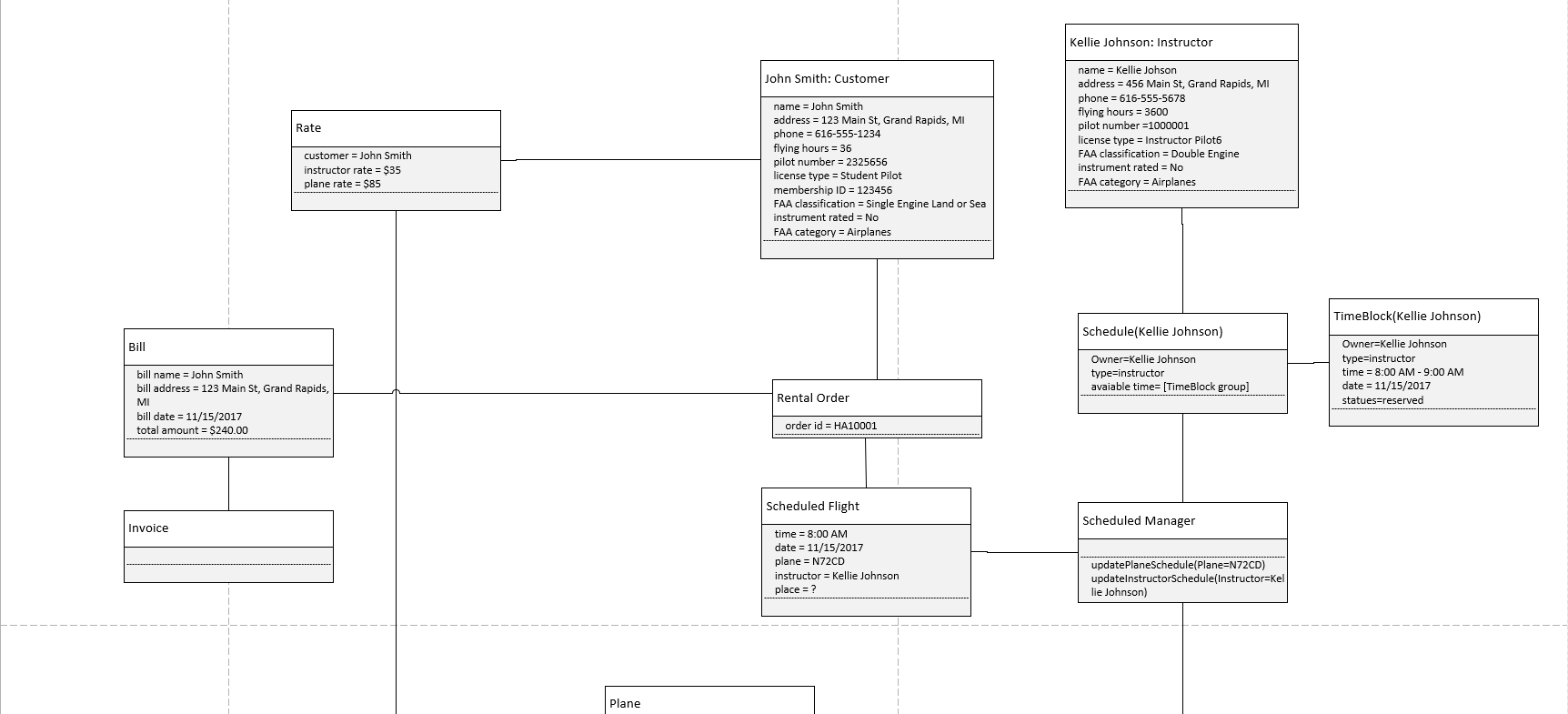
****

****

[figure 4.1-1]

* 1. **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.

****

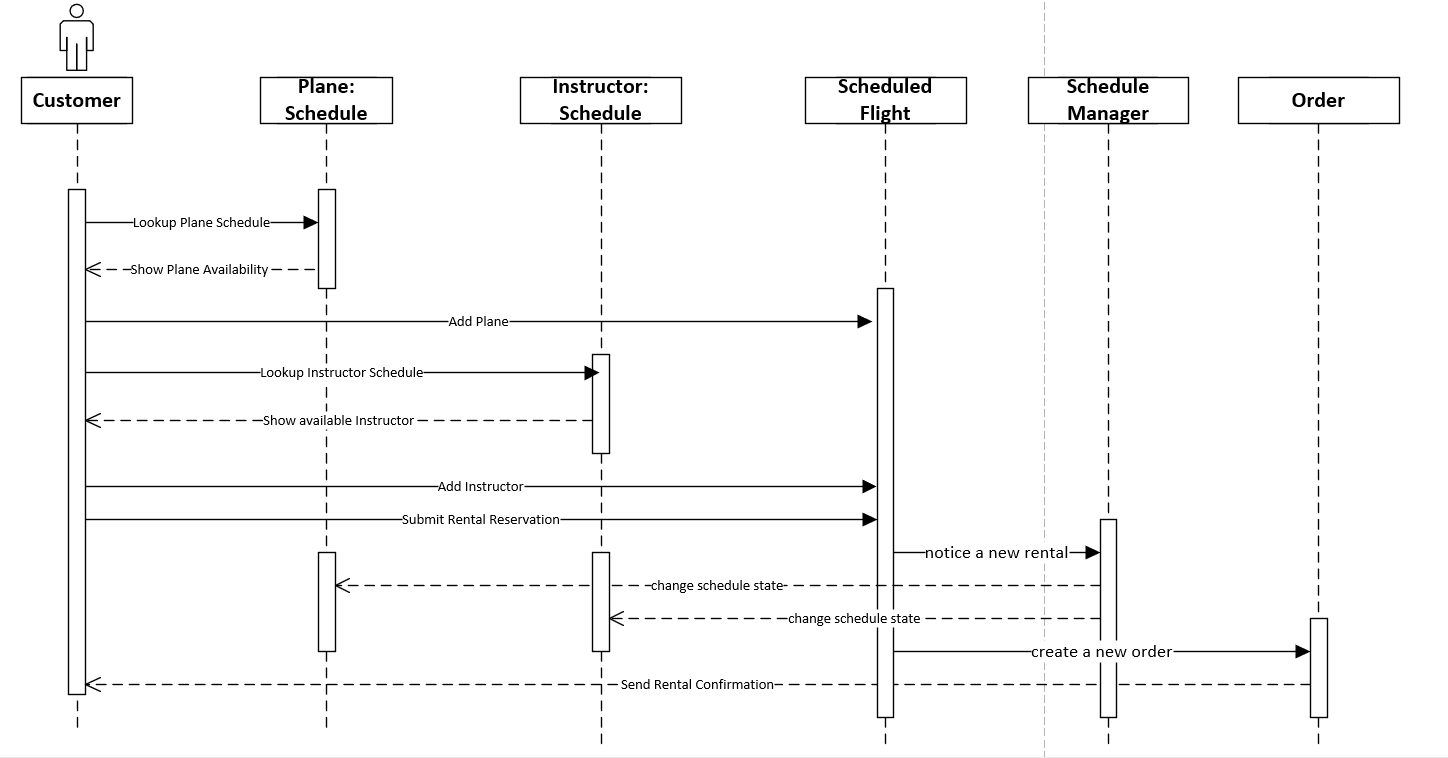
[figure 4.2-1]

**Section 5 System Behavior Analysis**

## **5.1 Sequence Diagram**

* Reserve a plane (Customer)

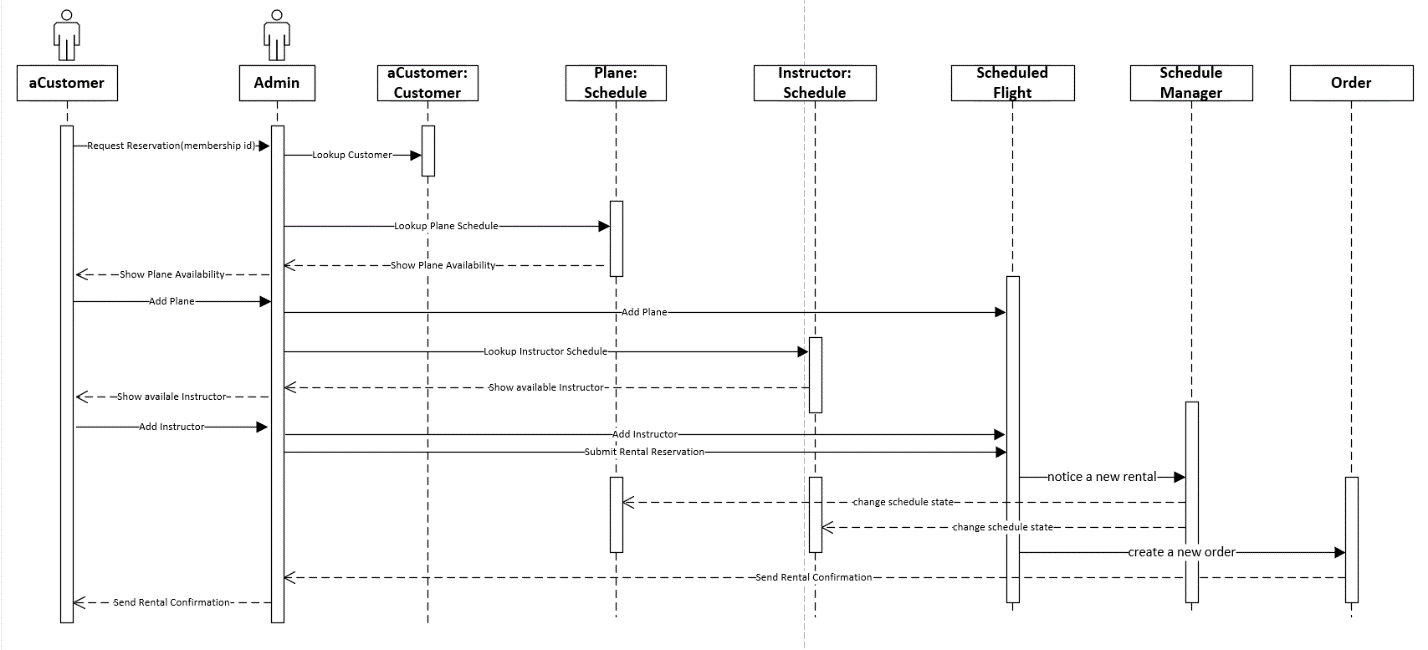
The following sequence diagram depicts the process when customer reserve a plane by himself through the website.

****

[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.

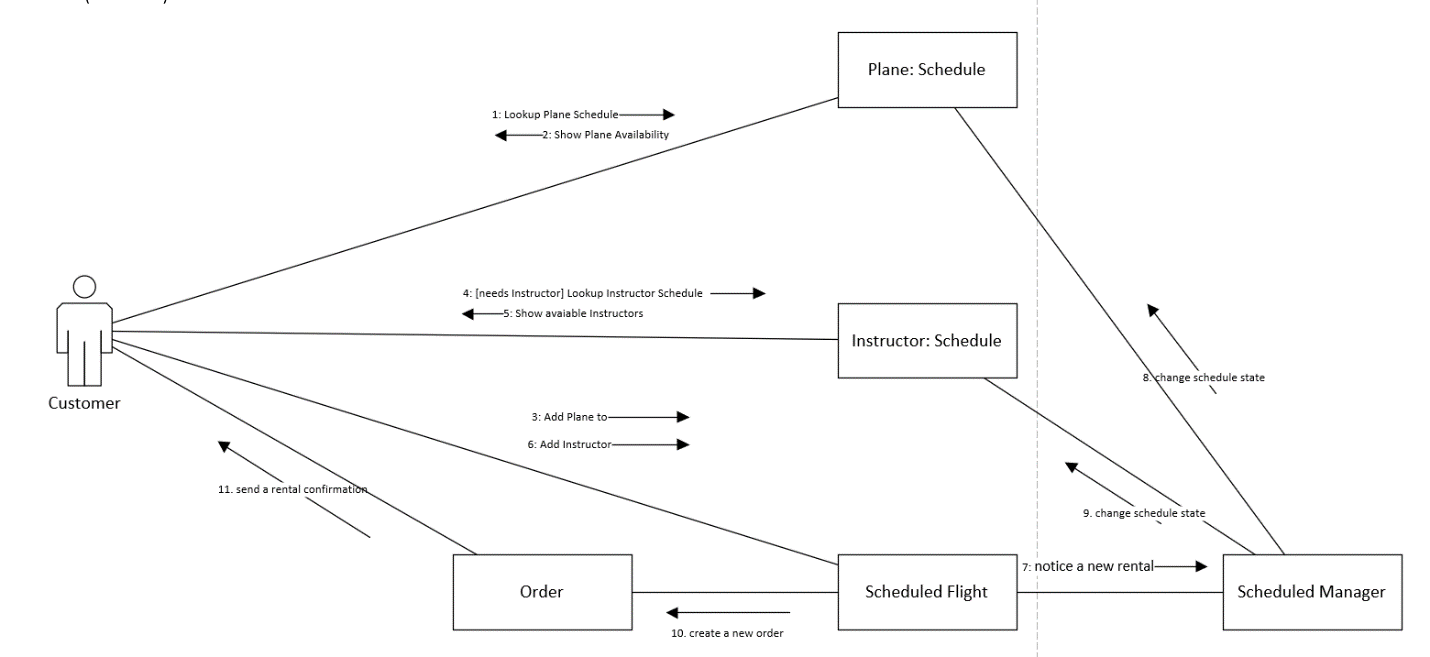
****

[figure 5.1-2]

## **Communication Diagram**

* Reserve a plane (Customer)

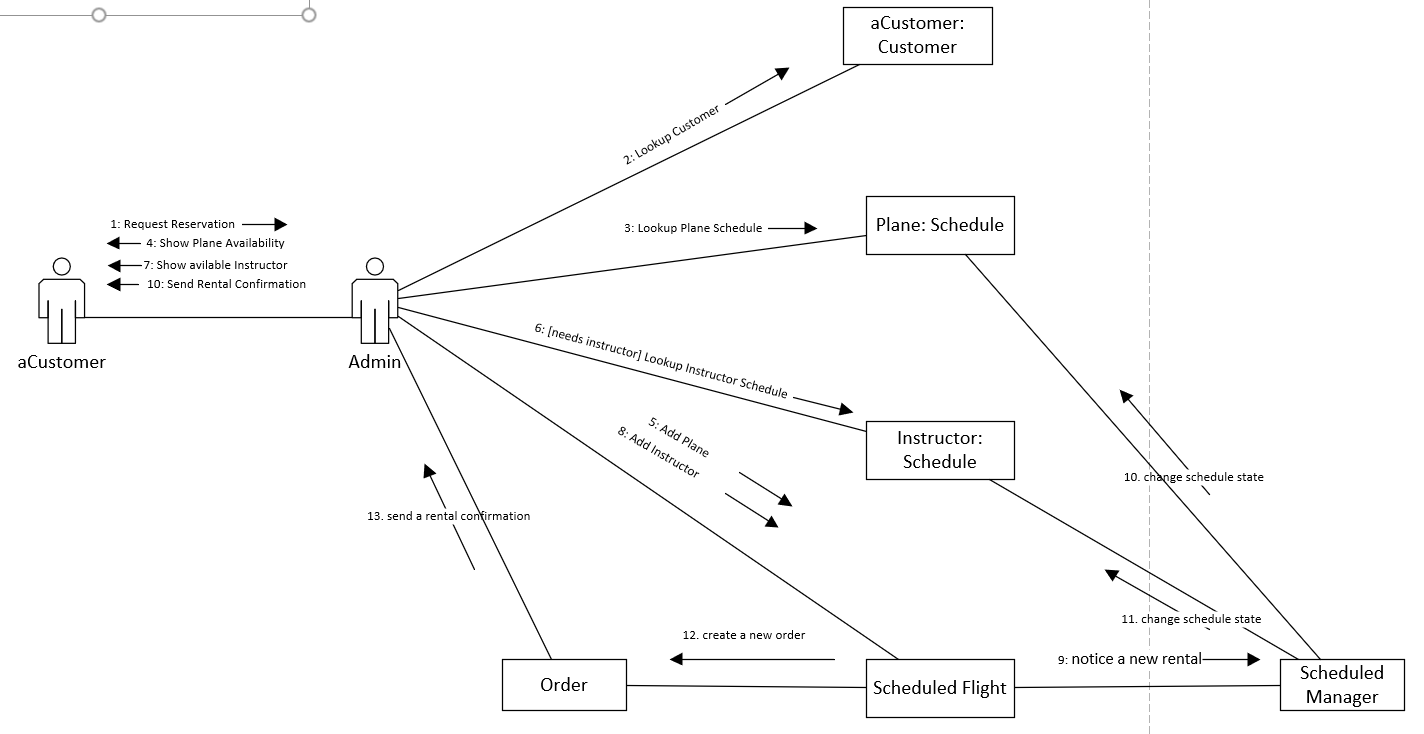
This communication diagram corresponds to the sequence diagram for the use case Reserve a plane (customer) [figure 5.1-1].



[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].

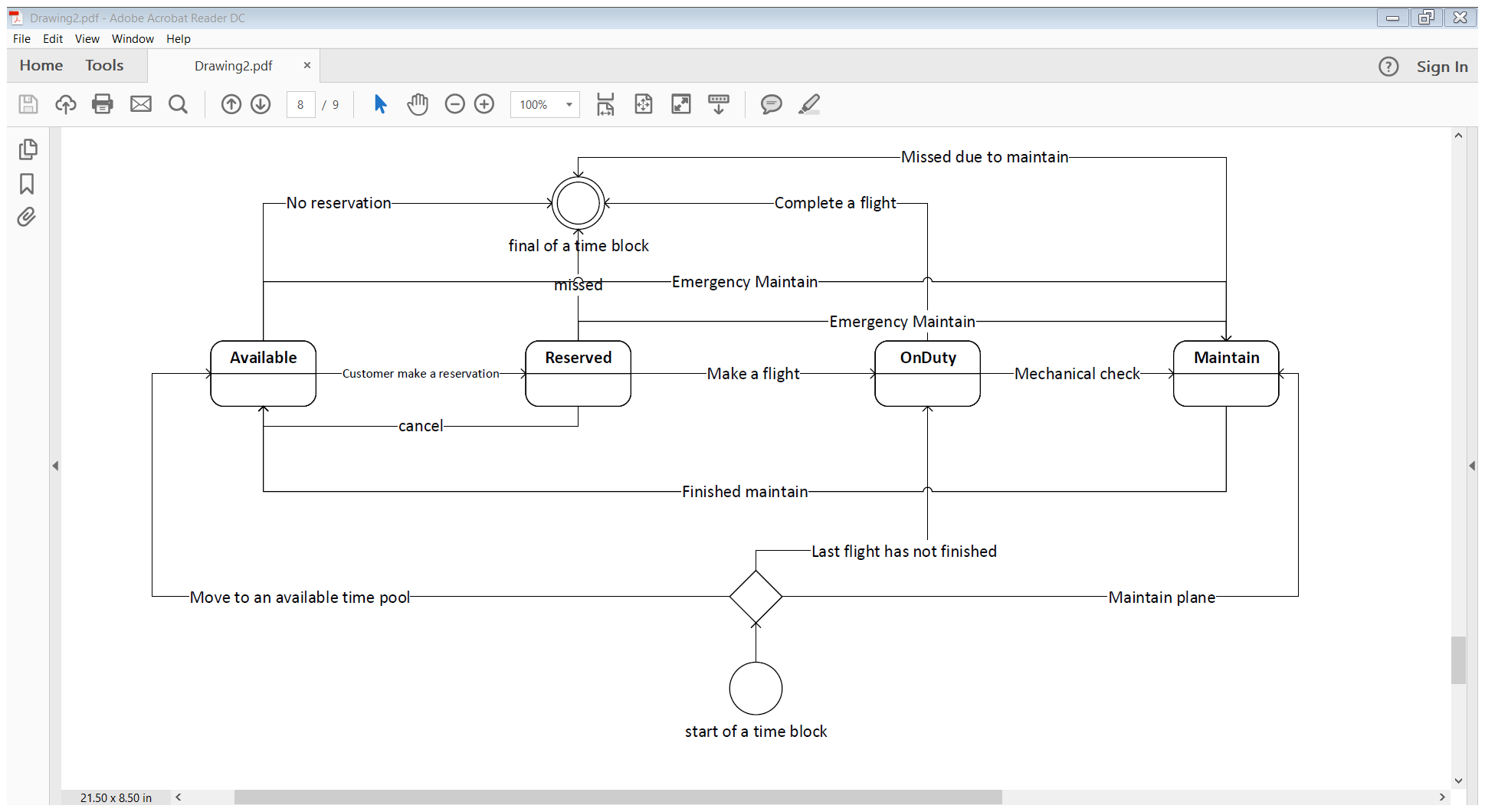
****

[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]

**Section 6 Data Management Layer Design**

## **6.1 Object Persistence Format**

The Husky Air plane rental system should use an Object-Oriented Database Management System (OODBMS). An OODBMS is able to support both simple and complex data types and also directly supports object-orientation. The use of an OODBMS system can help Husky Air avoid impedance mismatches between objects in the software system and data stored in tables. Additionally, the Husky Air system may need to store complex data types such as images, videos or audio so an OODBMS should be used to support these needs.

Since the Husky Air system will enable customers and other users to make, update, and cancel plane and instructor reservations, generate invoices, and perform other functions, the system needs data storage formats that are well suited for continuous data updates from many users and respond quickly to queries from users. An OODBMS can support transaction-processing systems such as this.

The system needs to support random access files and operations such as finding and updating specific objects in an efficient manner. The application will utilize all different types of files including master files, lookup files, transaction files, audit files and history files. Examples of master files for the system include customer and instructor profile information and information about the Husky Air planes. The system will use lookup files in certain instances, for example if a pilot is renting a plane the system can verify that pilot is qualified to rent a specific plane by using lookup files to validate a pilot’s certifications and license type. Transaction files will be used to update the master files and to update plane and instructor schedules, customer reservations, and so on. Examples of history files in the system will include old customer and instructor data, information about past plane rentals and instructor lessons, and other past transactions.

## **6.2 Data Access and Manipulation Classes**

There are several data access and manipulation classes are required for the system. The DAM classes include: **Customer-DAM, Instructor-DAM, Rental Order-DAM, Scheduled Flight-DAM, Plane-DAM, Schedule-DAM, and Bill-DAM**. These classes will need to be capable of at least reading and writing to the problem domain objects and tables. Since access to the database is frequent, we designed a database operation class to execute update, insert or delete operations.

**Section 7 HCI Design**

## **7.1 Use scenarios.**

Use Scenario: Existing Customer Reserves a Plane

1. Customer logs in to system to make a reservation (1)
2. Customer selects an available plane (2)
3. Customer selects desired date and time for plane rental (3)
4. Customer selects an available instructor if required (4)
5. System records rental reservation and sends confirmation

Use Scenario: Customer manages Existing Rental Reservation

1. Customer logs in to manage an existing rental reservation (1)
2. Customer selects option to manage reservations (2)
3. Customer selects option to reschedule or cancel an existing reservation (3)
4. Customer selects new date/time for reservation
5. System confirms new reservation and removes old reservation from system

Use Scenario: Admin Manages a Rental Billing

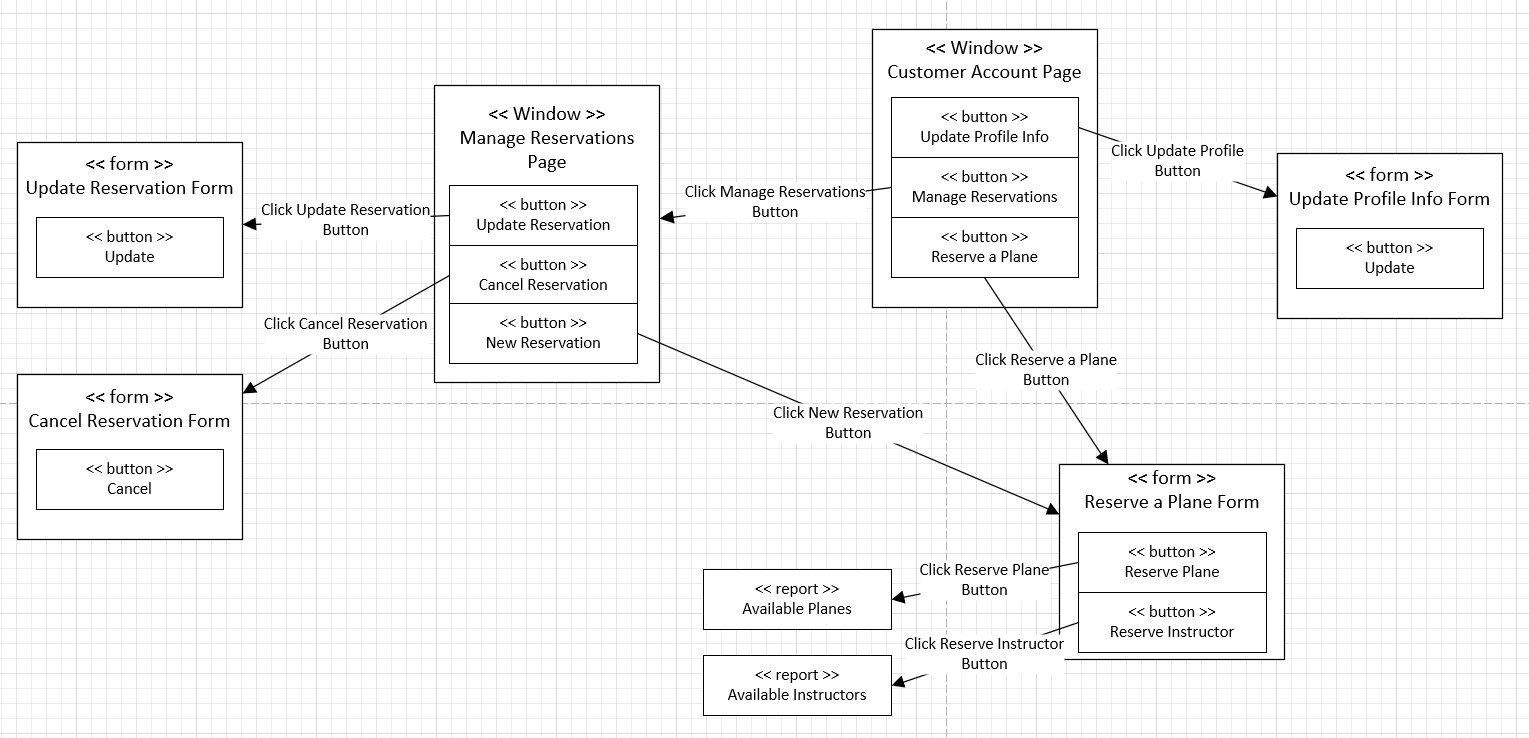
1. Customer submits beginning and ending Hobbs time to Husky Air Admin after a successful plane rental (1)
2. Admin records times to calculate total price for plane rental (2)
3. Admin adds instructor fee (if applicable)
4. Admin generates final invoice for customer (3)
5. Customer receives invoice (4)
6. Customer pays invoice (5)
7. Admin records invoice as paid

Use Scenario: Existing Customer Manages Customer Account

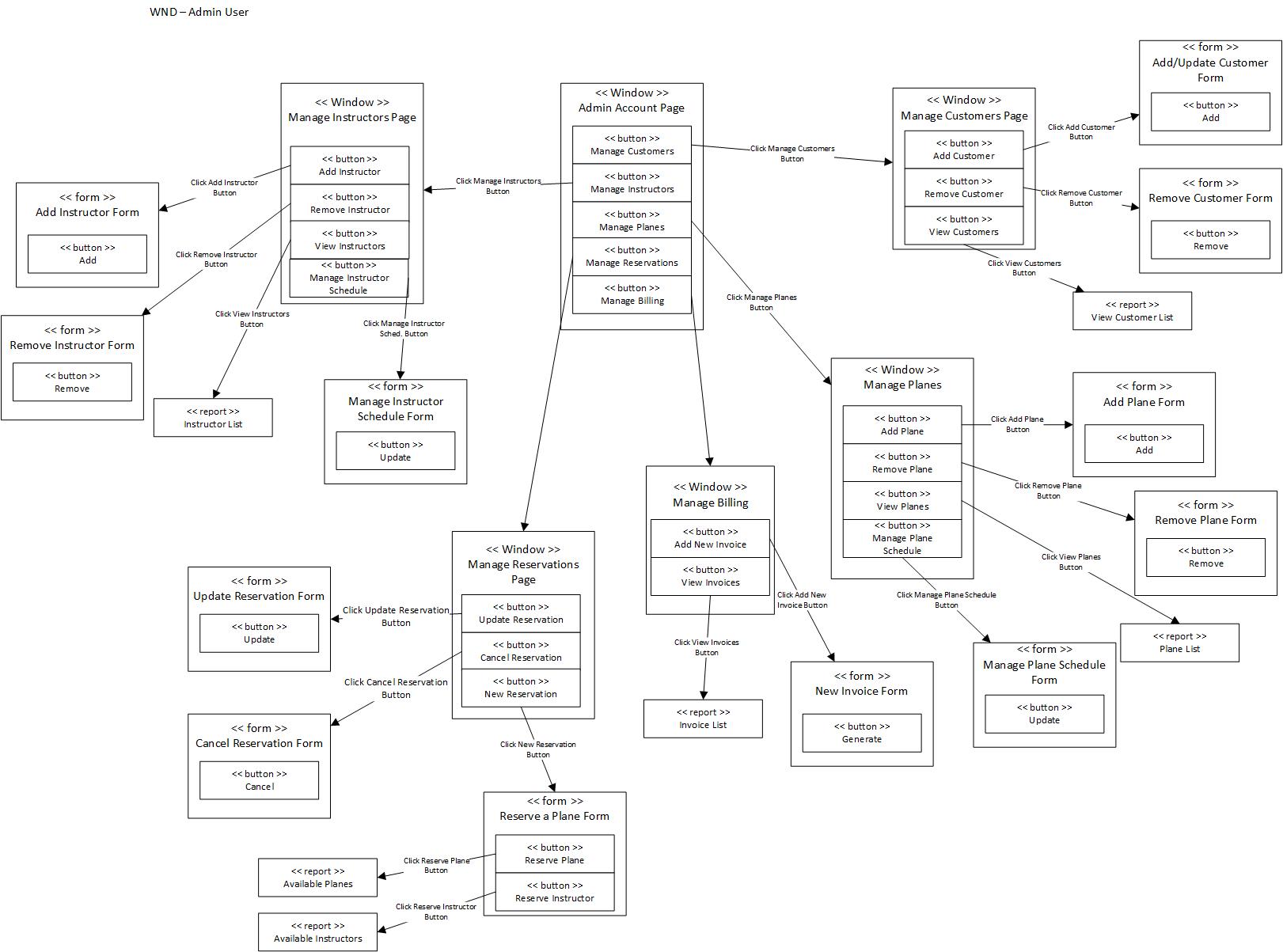
1. Customer logs in to system (1)
2. Customer is presented with account profile page (2)
3. Customer selects profile information to update (3)

## **Windows Navigation Diagrams**

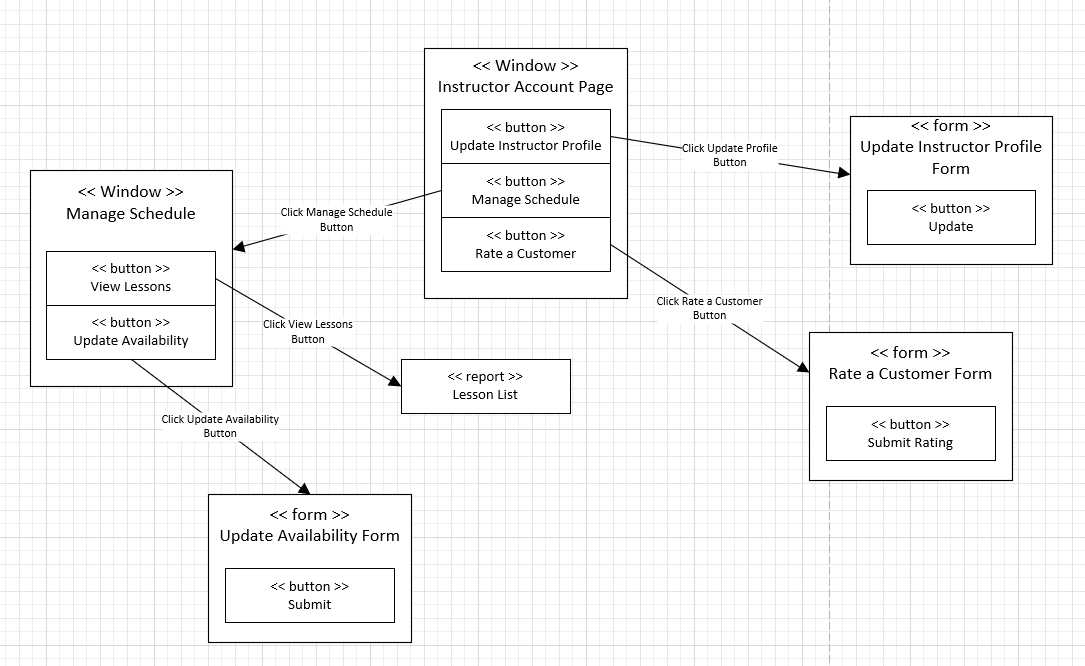
WND: Customer



WND: Admin

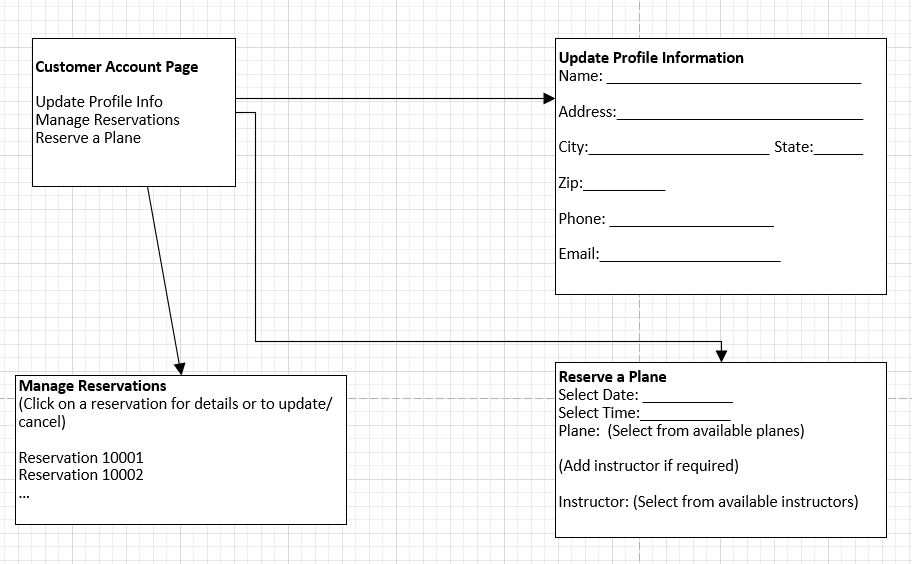


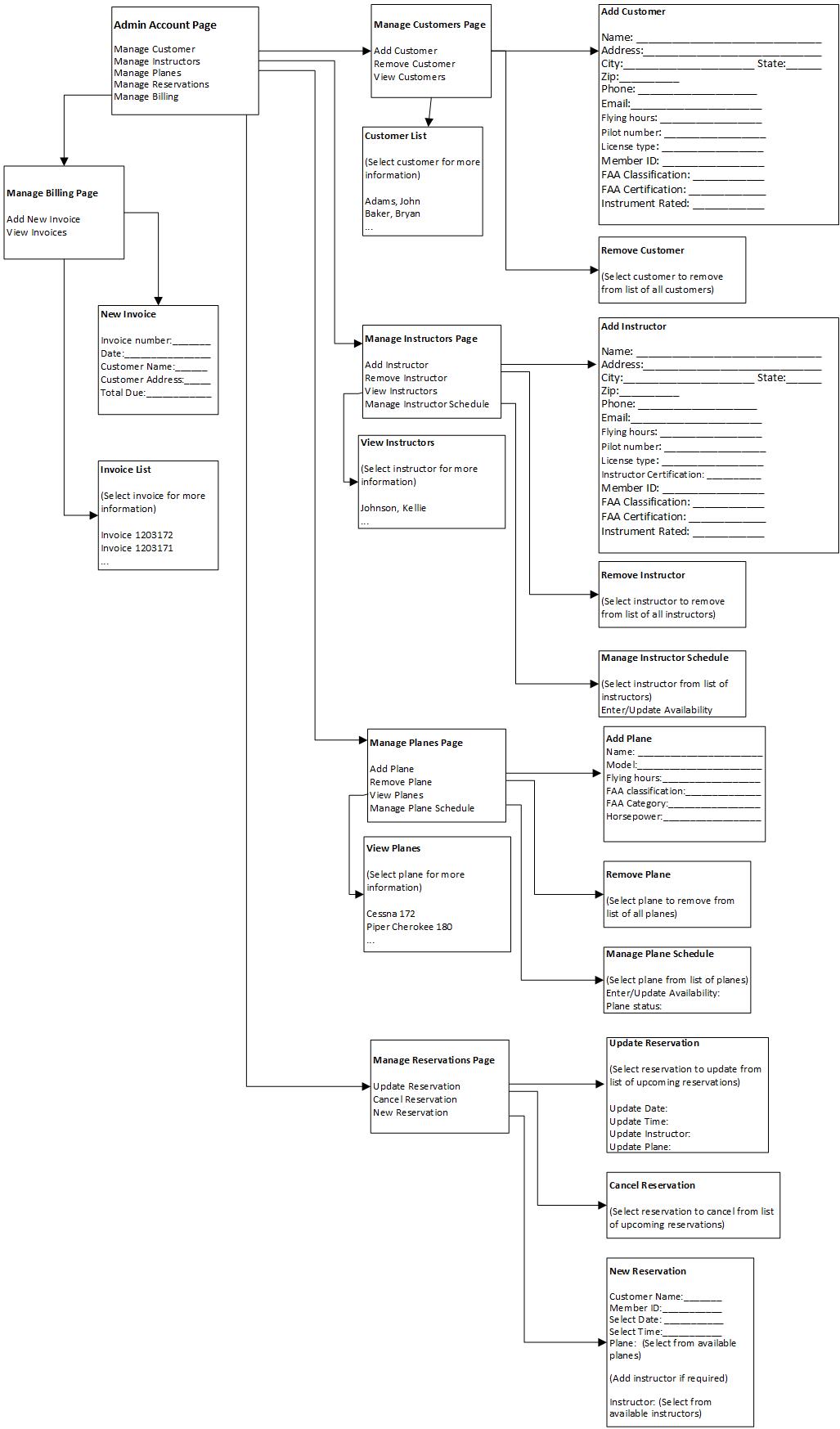
WND: Instructor



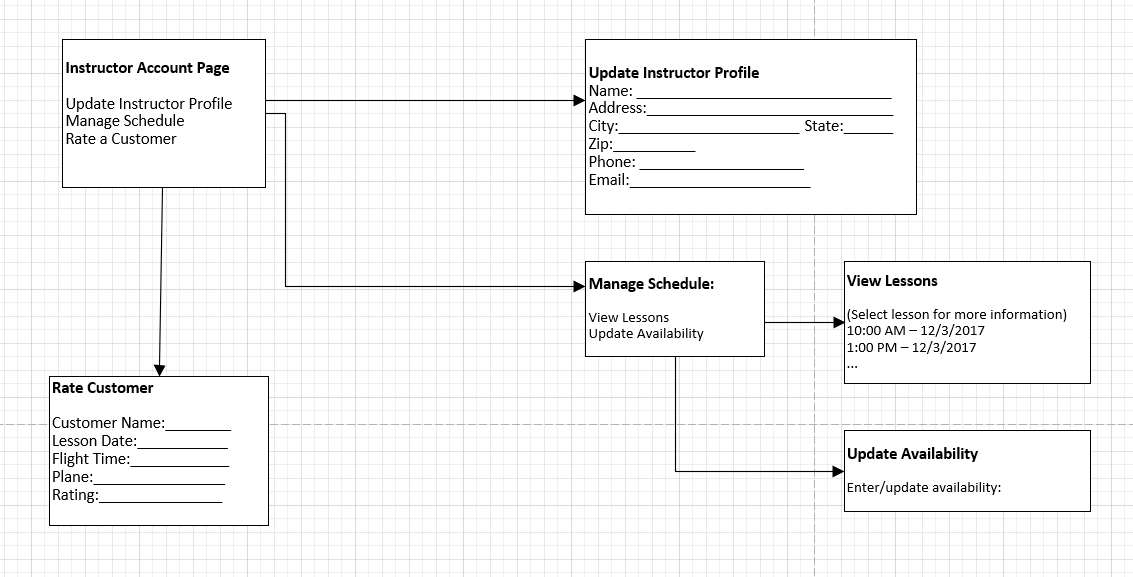
## **Storyboard**

Storyboard: Customer



Storyboard: Admin

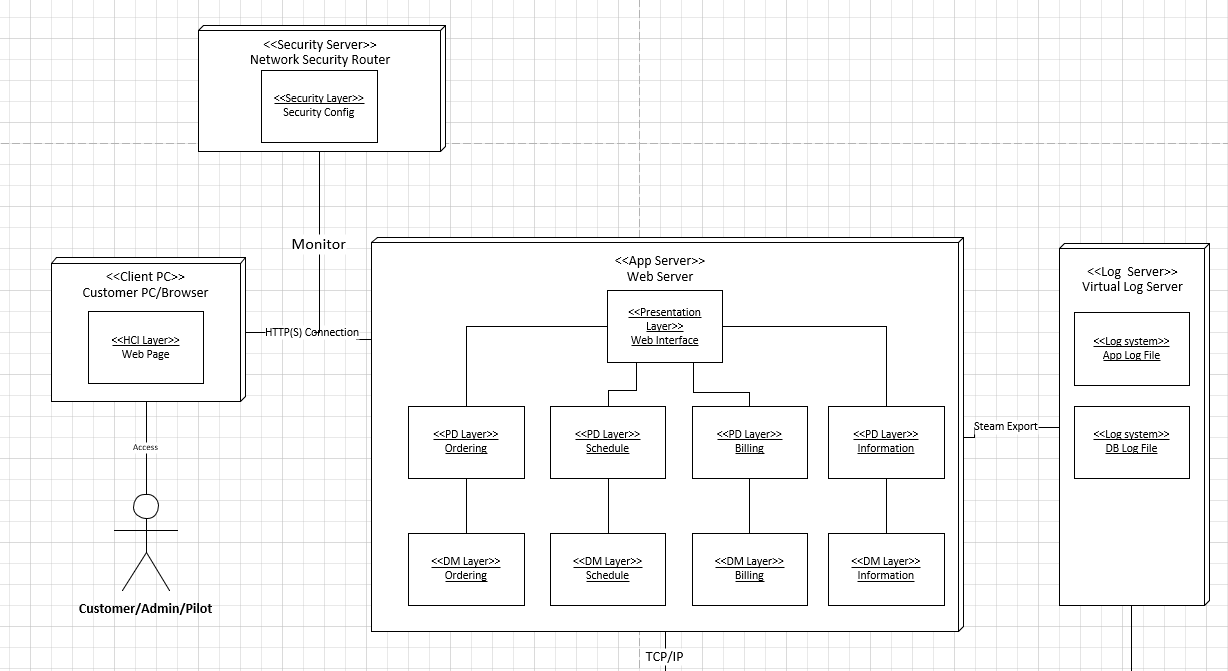
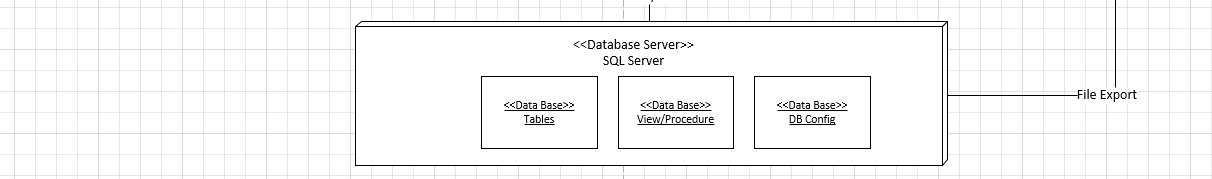
Storyboard: Instructor



**Section 8 Physical Architecture Design**

## **8.1 System Deployment Diagram**

A server-client module is adopted for the new HA rental system. Clients can access the web application server through firewall, and the web application server can access the database server through intranet. There is a virtual log server to record the events in firewall, app server and database. \* information layer includes customer, plane, and instructor profiles.



## **8.2 Non-functional requirements**

* Operational

|  |  |
| --- | --- |
| Type of Requirement | Description |
| Technical Environment Requirements | * System can be accessed by mainstream browsers, e.g. Chrome, IE, Firefox on PC or MAC * Whole office has an always-one network connection to the system or database * System provides capability to support user ordering through smart phone with only HCI layer upgrading. |
| System Integration Requirements | * System needs to import user information through excel or another flat file. * System might integrate with SMS system which can deliver some important messages to the users. * System might need to integrate with HR system to achieve authority management. |
| Portability Requirements | * System needs to easily be transferred to major smartphone OS, e.g. android, iOS through the mobile app. |
| Maintainability Requirements | * System will fix bugs and release new version every three months. * System can be upgraded to support some sensors to record actual flight information with one-month advance notice. |

* Performance

|  |  |
| --- | --- |
| Type of Requirement | Description |
| Speed Requirements | * Response time must be less than 2s through customer client; and less than 1s through intranet. (except advance analysis functions, e.g. reports) * Database must be updated in real-time, which means order is valid immediately. |
| Capacity Requirements | * Support maximum of 30-50 simultaneous users at peak user time (with plane number growth, the maximum access users will be increased accordingly.) |
| Availability and Reliability Requirements | * System requires 24/7 on-line with 99% uptime performance except scheduled maintenance. * Scheduled maintenance will be 2-3 times per year and less than 12 hours each time. |

* Security

|  |  |
| --- | --- |
| Type of Requirement | Description |
| System Value Estimates | * A system outage is estimated to cost $1000 per day, per plane * A complete loss of data is not acceptable. |
| Access Control Requirements | * Customer can only create order and update limited information of his own * System admin can update order and schedules of pilot or plane, or create basic information of pilots and planes. * System admin can create bills and invoices * Pilot can update his own information and create schedule |
| Encryption and the Authentication Requirements | * All users need username and password to login to system * Data base is required to maintain user name and password information * The web connection between the different branches and server should be secure. |
| Virus Control Requirements | * Network router is accountable to check viruses * App server and database server is installed anti-virus software |

* Cultural and political

|  |  |
| --- | --- |
| Type of Requirement | Description |
| Customization Requirements | * Two languages support: English and Spanish |
| Legal Requirements | * Personal Information cannot be exported outside of company * Personal information cannot be divulged by oral or written method. |

**Section 9 Conversion Plan & Change Management**

The conversion plan depicts the strategies and the process of how HA Plane rental company will convert from the manual system to the electronic system.

**Conversion Strategy**

Since there is not any legacy system, we adopt a strategy that implants the whole system in one of the branches performing the pilot job and replace the manual work system. The manual process is retained for backup purpose in case there are some critical problems that the pilot system cannot handle. Additionally, the pilot will be split to pilot A and pilot B. In pilot A, the requests of customers will be performed by the company employees, e.g. system admin. In pilot B, customers can login to the system by themselves and reserve a plane. We chose this strategy because it can keep risks in a controlled area as well as quickly develop a group of training leaders that can deliver knowledge to other branches. The brief chart of the conversion strategy can be seen below:

## **9.1 Conversion Process**

One week before the pilot implementation, all the hardware, software and data should be prepared. With the exception of the client PC, all the servers and databases are located in the headquarters, and the pilot branch accesses it through a security network.

* Hardware conversion

Hardware vendor should deliver and install the application server, database server and security hardware one month in advance and ensure the network is workable between server and client in the branches and internet.

The necessary client hardware upgrade might be deployed after hardware evaluation. All the retired client PC will be replaced by new hardware one month in advance. All the hardware is listed below:

|  |  |  |  |
| --- | --- | --- | --- |
| Hardware Name | Numbers | Location | Install Date |
| Application Server | 1 | HQ | 20/12/2017 |
| DB Server | 1 | HQ | 30/12/2017 |
| Security firewall | 2 | HQ+Branch | 25/12/2017 |
| Network | 2 | HQ+Branch | 5/1/2018 |
| Client PC | 15 | Branch | 10/1/2018 |

* Software conversion

After the hardware is prepared, the Operation system, database and other required software needs to be installed on server or clients a half month in advance. Also, the rental server application should be deployed on the application server.

|  |  |  |  |
| --- | --- | --- | --- |
| **Software Name** | **Number** | **Hardware** | **Install Date** |
| **Windows 10 server R2** | 2 | Application server  DB server | 1/1/2018 |
| **SQL Server 2017** | 1 | DB server | 1/1/2018 |
| **Anti-virus** | 15 | New client | 6/1/2018 |

* Data Conversion

Since there is no legacy system, only configuration data and basic profile data will be built on the system. The plane, instructor, employee and customer information would be created by importing from Excel sheets into database through ETL. The historical rental transaction data will not be included in this stage.

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Source** | **Destination** | **Transfer Date** |
| **Plane profile & price** | Plane.xlsx | Plane related tables | 13/1/2018 |
| **Customer Profile** | Customer.xlsx | Customer related tables | 13/1/2018 |
| **Instructor profile** | Instructor.xlsx | Instructor related tables | 13/1/2018 |
| **Employee** | Employee.xlsx | Employee and organization | 14/1/2018 |
| **Authority** | Role.xlsx | Role | 14/1/2018 |

* Backup & Recovery Plan

The data server performs backup job at night each day; the application server performs backup job at night every week or before launching a new version of the system. If any irreparable critical error happens, the recovery action should be executed in a half hour.

## **9.2 Conversion Evaluation**

After completing all tasks of conversion plan, there are several criteria to evaluate the result of hardware, software and data respectively.

|  |  |  |  |
| --- | --- | --- | --- |
| **Scope** | **Criteria** | **Completed** | **Issue** |
| **Hardware** | * all the servers work correctly * the network connected with the designed speed * security firewall works correctly |  |  |
| **Software** | * all required software is installed properly as designed performance * database can be accessed through network |  |  |
| **Data conversion** | * all data imported properly |  |  |
| **Backup** | * complete backup and restore tests * run backup tasks correctly as scheduled |  |  |

## **9.3 Change management plan**

Change management is an essential part of deploying HA rental system, which helps our employees and managers adapt to the new system without undue stress. There are three major steps in change management.

* **Awareness**

Before the project kick-off and in the life time of the whole project, making people aware is important which allows people to adapt to the new process or skills comfortably. The basic changes include:

|  |  |
| --- | --- |
| **Changes** | **Affected Role** |
| Manual process will be replaced by system, all the business processes will be handled in this new system | All employees  All customers |
| Operate rental order through browser | Call center employee  Customer  Front-site employee |
| Check the order price and detail on-line | Finance  Sales |
| Make schedule | Plane admin  Instructor |

The new system evolution is a persistent process. Any employee or customer can promote his change request to the change committee which will make the decision and feedback. The change committee includes: Vice president(sponsor), Project manager, IT, business department liaison, finance liaison, etc.

* **New SOP**

According to the new system, a new coordinate SOP (HA Plane Rental System SOP) might be launched which is related to the awareness but more detail. For example, a call center employee should follow this SOP to answer costumer’s call for reservation and make the current operations to create(update) an order for this customer. Then send the successful reservation feedback to the customer. The possible sections in the new SOP are depicted as below:

|  |  |
| --- | --- |
| **SOP Section** | **Responsible Person** |
| Customer Reservation | Font-site employee  Call center employee  System admin |
| Plane Schedule | Plane Admin |
| Instructor Schedule | Instructor |
| Change Request Management | All employees |
| Unite Price Change | System Admin |
| Order history export  Order detail export | Finance  Sales  System Admin |

* **Access Cost & Benefit**

For the conversion stage and post-conversion stage, the new system brings significant benefit, as well as some cost. The following table describes the costs and benefits.

|  |  |
| --- | --- |
| **Cost** | **Benefit** |
| Learning curve | Increased efficiency and skills for employees |
| Training cost | Less unpleasant areas for frontline employees |
| Employee needs time to adapt | Easier to schedule plane and instructor for admin and instructor |
| Customer needs time to adapt | increased efficiency for support functions, e.g. finance, IT, supply chain |
| System maintenance cost | Possibility to purchase more planes for management level |
|  | Competitive advantages in the field of plane rental |
|  | Increased comfortable for customer |

* **Training**

Classroom Training is adopted in the pilot period, then a computer-based Training will be extended to the whole company in the second period. The representatives in each department and maintenance team take responsibility for the questions from the internal users.

* **Bonus**

In the period of implementing the new system, the company will give a bonus for the person or team who:

* Becomes most familiar with system, and is able to be an instructor of system in his department
* promotes the most valuable suggestion
* the quickest team which completes the training process and uses system with least errors

## **9.3 Pilot Evaluation**

The whole pilot estimate takes 3 months, then it will be evaluated if the new system would be deployed into the whole company. The evaluation contents will be listed as below:

|  |  |
| --- | --- |
| **Section** | **Measure criteria** |
| hardware | No significant problem, e.g. shut down, very slow  Running functionally by 7\*24 |
| Software | No conflicts between software.  Running functionally by 7\*24 |
| Data | No data missing or lost  Performance achieve target |
| Back up & restore | Back up performs functionally each day  Restore performs functionally in any given test |
| Employee feedback | No significant stuck in the business because of system  All the high-middle bug has been fixed |
| Sponsor feedback | Business has been improved by system  No more grand business change which leads to system changes |
| Customer feedback | Mostly comfortable the new rental system  Average more positive feedback in survey |
| Cost | Cost in the scope of plan |

Appendix A – Work Breakdown Schedule

* **Analysis**
  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. 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. 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. 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. Create project analysis report document – Jie Week3
  2. Review analysis report document – John, Jie Week3
  3. Submit final version of analysis report document – John Week 3
* **Design**
  1. Design Plan [Group Task – John, Jie] Week4
     + Review the analysis
     + Break down the work
     + Clear relationship between tasks
     + Assign the tasks
  2. Data Management Layer – John Week5
  3. Human-Computer Interface Design – John Week5
  4. Physical architecture Design – Jie Week5
  5. Conversion and change management plan – Jie Week5
  6. Validation documents – John, Jie Week6

# **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. So, we describe these user case with their subflows in four user case descriptions. | 15 | 4 |
| Activity Diagram(s) | Required. Four use cases have one or more decision logic, so we drew the activity diagrams to clarify them, which includes: “manage user account, reserve a plane, manage reservation and manage schedule” | 4 | 4 |
| 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. | 1 | 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. | 1 | 0 |