Software Engineering

COMP433

G1: Flight Booking System

Final Project Report

Customers for Group #2

Developers for Group #9

Instructor: ***Prof. Adel Taweel***

Date: December 23, 2019

Table of Contents

Chapter 1: Project Planning and Management

1.1: Group Name

OutOfIndex: G1

1.2: Name of Members

Simon Asmar: 1162643
Sabry Alawy: 1162074
Laith Marzouka: 1160827
Layth Abufarhah: 1162636

1.3: Role of Each Member

Project Manager: Simon Asmar
Secretary: Sabry Alawy
Technical Architect: Laith Marzouka
Programmer: Layth Abufarhah

1.4: Project Management Strategy

Most of the meetings were done face-to-face immediately after the lecture once a week. Also, video conference meetings were conducted sometimes when needed. It is worth mentioning that the group was in touch for the whole semester via a messenger group and a lot of decisions and discussions were made through it. The team made decisions through voting and in case of a tie in votes the manager had the final decision most of the times. The software process model used was agile, since we had meetings with the customers every week in class, and we kept in touch to receive feedback and improve the work.

1.5: Project Manager Report

Work done by each member

Some work was done collectively while others were done individually.

1. **Simon Asmar**: management strategy, user & system requirements, effort + time estimation (draft), *scenario, activity diagram, and detailed use case for flight booking form*, overall use case diagram (draft), deployment diagram, state diagram, and sequence diagram for the *flight booking form*.

2. **Sabry Alawy**: user & system requirements, *scenario, activity diagram, and detailed use case for transactions (payments)*, overall use case diagram (draft), component diagram, design goals (draft), and sequence diagram for *transactions (payments)*.

3. **Laith Marzouka**: business outline for group 2, user & system requirements, effort + time estimation (final), *scenario, activity diagram, and detailed use case for searching for flights*, overall activity diagram, overall use case diagram (draft and final), architectural design, detailed class diagram, deployment diagram, object diagram, design goals, and sequence diagram for *searching for flights*.
4. **Layth Abufarhah**: user & system requirements, *scenario, activity diagram, and detailed use case for user registration*, overall use case diagram (draft), analysis class diagram, description of classes, component diagram, and sequence diagram for *user registration*.

Barriers and challenges faced in managing the project:-

- Sometimes it was hard to reach out to our group members and the members of the other groups that we were working with.
- Holidays and strikes prevented us from meeting with each other sometimes to discuss and work on our tasks.
- Sometimes we had a lot of things to do in subjects other than software engineering and at the same time, we had a lot of tasks due to this project. So it was a bit difficult to manage our time and complete everything on time.

Opinion & Evaluation

I am happy with how the project ended up to be. We had some challenges and barriers but the team was great. We worked collectively and helped each other understand certain things we weren't able to comprehend at first. The project itself is a great success from my point of view and I'm proud of our final product. We worked hard and we tried to make it look as perfect as possible. By working on the project we learned a lot about software engineering and the way things are done.

1.6: Project Members Report

1-Sabry Alawy

We have learnt a lot from this project and knew new people and had gained the ability to work with a team under stress. Besides the barriers we faced, we did a good job to keep on track and make this project to the end.

My tasks:

- Contributed to User and System requirements
- Contributed to the overall use-case diagram
- Contributed to the component diagram
- Did all individual tasks

2- Laith Marzouka

Fortunately, we were the first team to select a project, but we chose one of the toughest projects. This means that we wanted a challenge and to learn from it rather than just something to get done with as fast as possible. We worked hard a lot to finish this project, everyone did his part most of the times. However, we struggled a lot during this project especially in the end where we needed help most but we couldn't get it due to the recent strike. Customer's group was a very hard to get in touch with and we had to wait for the lecture time many times. Nevertheless, I am fairly satisfied with the work we have done although I believe with better situations we could have done a better job.
One thing I am sure of is that I have learnt a lot, acquired social skills along with academic knowledge. Most importantly, I dealt with serious stress and lack of time. Therefore, I call this project a successful project that made us give out power and time we barely had that time.

My tasks:

- Contributed to creating the business outline for the customers
- Contributed to User and System requirements by writing parts of it & discussing them
- Contributed to Effort/Time estimation by reviewing it
- Contributed to writing the actors and overall use-case and activity diagram
- Contributed to the filling of the assessment
- Contributed to graphing the detailed class diagram
- Contributed to graphing the object diagram
- Contributed to graphing the deployment diagram
- Contributed to reviewing the state diagram
- Did all required individual tasks

3- Layth Abufarhah

I got a great experience doing this project, even if we didn't do everything perfectly but I benefited greatly from it. It gives me a better vision of how software is constructed and what is the role of computer engineering. I'm so happy about working with this team we have cooperated well to achieve this project and we kept on touch in the university and online.

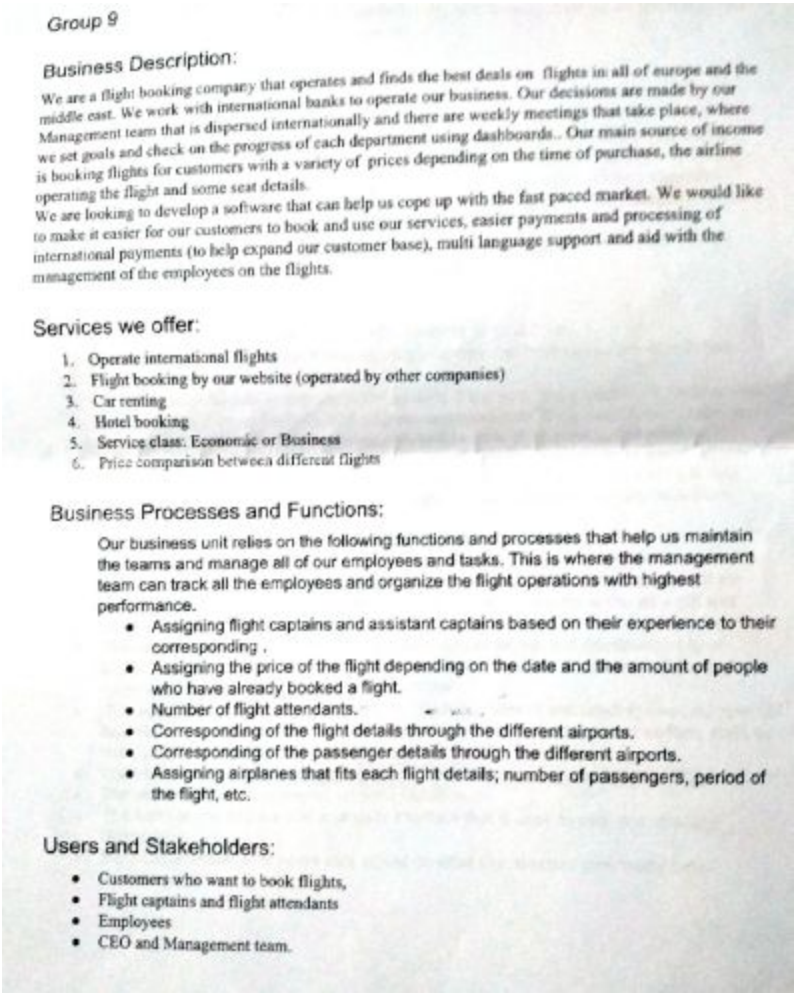
My tasks:

- Contributed to Requirement Analysis.
- Contributed to USER and SYSTEM requirements.
- Write SCENARIO for my use case, which was user system registration.
- Detailed use case for user registration.
- ACTIVITY Diagram for the use case.
- Sequence Diagram for the use case.
- Draft the overall use cases diagram.
- Draft and final description of Classes.
- Draft and final analysis class model diagram.
- Contributed to the detailed class model diagram
- Contributed to the object diagram.

- Draft and final component diagram.

Chapter 2:Requirement Elicitation and Analysis/Modelling

2.1: Business Description[1]



Group 9

Business Description:

We are a flight booking company that operates and finds the best deals on flights in all of europe and the middle east. We work with international banks to operate our business. Our decisions are made by our Management team that is dispersed internationally and there are weekly meetings that take place, where we set goals and check on the progress of each department using dashboards.. Our main source of income is booking flights for customers with a variety of prices depending on the time of purchase, the airline operating the flight and some seat details.
We are looking to develop a software that can help us cope up with the fast paced market. We would like to make it easier for our customers to book and use our services, easier payments and processing of international payments (to help expand our customer base), multi language support and aid with the management of the employees on the flights.

Services we offer:

1. Operate international flights
2. Flight booking by our website (operated by other companies)
3. Car renting
4. Hotel booking
5. Service class: Economic or Business
6. Price comparison between different flights

Business Processes and Functions:

Our business unit relies on the following functions and processes that help us maintain the teams and manage all of our employees and tasks. This is where the management team can track all the employees and organize the flight operations with highest performance.

- Assigning flight captains and assistant captains based on their experience to their corresponding .
- Assigning the price of the flight depending on the date and the amount of people who have already booked a flight.
- Number of flight attendants.
- Corresponding of the flight details through the different airports.
- Corresponding of the passenger details through the different airports.
- Assigning airplanes that fits each flight details; number of passengers, period of the flight, etc.

Users and Stakeholders:

- Customers who want to book flights,
- Flight captains and flight attendants
- Employees
- CEO and Management team.

[1] This description is not the final one, because we couldn't reach for the final version because the customer group can't seem to find it anywhere. Nevertheless, there is no real final description because we edited it many times during the discussions and it was never properly written by them

- Investors
- International banks (International payments)
- People who will track the flights of the customers (To pick them up from the airport, and let them know about the exact time of arrival /departure)

Environment and Circumstances:

Our flight booking system will maintain a good travelling atmosphere. We are tending to offer different types of flight tickets in addition of other accommodation requirements that includes car and hotel. We intend to have a software system to make the process of searching and buying flight tickets becomes easier and flexible, so we attend to provide the following services:

We need an online booking system that offers the best deals on flights; the best price, less travelling time and the most comfortable flying experience.

- It is a website application
- The system must be suitable for any person wants to travel i.e: businessmen,tourists,..etc from one country to another or from city to another in one country.
- The user shall be able to register in the system. If the user has an account, he/she shall be able to log in using his/her email address and password. If the user doesn't have an account he/she shall be able to create one, and he /she shall enter his/her valid information, which are full name, phone number, address, email address(detailed with city, street, department) and password to complete registration. When creating a new account, the system should send a confirmation email to check its validity, and then activate it.
- The system should allow unregistered users to search for trips
- the system should determine the special users automatically. The system shall mark a user to be a special user if he/she has reserved at least 8 trips in two years and If the user mark as special user the system should ask him/her to order a trip as a gift and make discounts for his/her future trips .
- The user can search for trips based on source city or airport and destination city or airport,airline,departing and returning date, number of travellers includes adults and children,and choices for adding car and/or hotel.
- The system shall display trip information, which is a takeoff and landing time, number of stops and the period of each stop -user can choose either the trip could contain stops or not- ,and the total price. Moreover,the user can see the flight rate..
- The user can buy flight ticket and reserve a hotel or car using visa card
- The user can choose trip based on price,duration.
- The website should have a simple user interface that is easy to use, not so many distractions.
- the website should offer some trips based on what the user has previously been searching for.

2.2: User Requirements and System Requirements

- **UR1: System Registration**

  SR1.1: Employee System Registration

  SR1.1.1: There is one Manager account in which he shall:

  a. Create accounts for employees.
  b. Search for employees and manage their information (check UR3).
  c. Authorize/Deauthorize employees from System Accessing Privileges (SAP) (check UR3, UR4-SR4.1):
     1. HR employees (SAP1).
     2. Flights managing employees (SAP2).
  d. Delete employees' accounts from the system.
  e. add /remove job titles.
  f. Manage Flights (check UR2).

  SR1.1.2: Employees' accounts are registered based on this information:

  a. Employee ID.
  b. Full name.
  c. Gender.
  d. Date of birth.
  e. Phone number.
  f. Job title.
  g. Email address.
  h. Password.

  SR1.1.3: All of the above information must be filled.

  SR1.1.4: Employee ID is unique and is set by the manager and is not necessarily incremental.

  SR1.1.5: Job title is not arbitrary and must be selected from a list of job titles. However, there is an option 'other' and should be filled in a suitable manner.

  SR1.1.6: If a job title is removed but there exist employees with such job title, then their job title remains the same as if it was an 'other' option.

  SR1.1.7: Employees shall access their information on the system, and shall only change their email address or password. If any other information is in need to be changed, employees have to contact the manager or HR employees 'SAP1'.

  SR1.1.8: HR employees 'SAP1' can also create employee accounts but not delete.

  SR1.2: User System Registration

  SR1.2.1: Users shall register by entering the following information:

  a. Full name.
  b. Gender.
  c. Date of birth.
  d. Email address.
  e. Password.

SR1.2.2: All of the above information must be filled.
SR1.2.2: Signed-in users may set a fixed payment method for easier future bookings.
SR1.2.3: Users shall change their information/payment method at any time.
SR1.2.4: Signed-in users shall see the history of their previous bookings.
SR1.2.5: Signed-in users shall have a 20% sale on the next ticket after every 5 tickets successfully purchased.

- **UR2: Manage Flights**

SR2.1: Add Flights

SR2.1.1: Authorized employees shall add a new flight to the system with the following information:

a. Unique flight number.
b. The number of seats for every class (first, business, economy)
c. Departure location & arrival destination.
d. Date & time of departure/arrival.
e. Duration of the flight.
f. Staff (pilots, pilot assistants, flight attendants, etc...).
g. Baggage prices for every weight (10kg and 23kg).
h. Flight meals.
i. Price of tickets for every class (first, business, economy).
j. Percentage of sale if any.
k. The number of booked seats.

SR2.1.2: All of the above information must be filled.
SR2.1.3: Flight number is unique and is assigned by authorized employees and is not necessarily incremental.
SR2.1.4: Staff members are added by choosing a job title, then a suitable list of available staff members will appear. This list has employees' names and IDs.
SR2.1.5: Available staff members are the ones that have no flights at the time of the added flight.
SR2.1.6: The flight is saved to the staff members record

SR2.2: Alter Flights

SR2.2.1: Authorized employees shall change the following information about a flight:

a. Date/Time.
b. The number of seats for every class.
c. Baggage price.
d. Staff.
e. Meal.
f. Price of tickets for every class(first, business, economy).
g. Sale percentage if any.

SR2.2.2: If a booked flight date/time is changed, then an email (if provided) must be sent to the email addresses of all customers who booked this flight.
SR2.2.3: Email body is filled by the authorised employee who changed the flight date/time.
SR2.2.4: Changes in price do not affect already purchased tickets.

SR2.3: Remove Flights

SR2.3.1: Flights booked by at least one customer should only be removed using the manager's account.
SR2.3.2: If a booked flight is removed, then an email (if provided) must be sent to the email addresses of all customers who booked this flight.
SR2.3.3: Email body is filled by the manager at the time when the flight had been cancelled.
SR2.3.4: The flight is removed from the staff members record.

- **UR3: Search for Employees & Their Flight Records**

SR3.1: Authorized employees (Manager & HR employees 'SAP1') should search for employees based on:

1. Employee ID.
2. Name.
3. Job title.

SR3.2: The list should display all of these employees' information, as well as a table of their flight records which is saved in the database for the captain, co-pilot, flight attendants etc.
SR3.3: Any employee information shall also be changed if wanted.

- **UR4: Search for Flights & Tickets**

SR4.1: Employees' Search for Flights

SR4.1.1: Authorized employees (Manager & Flight managing employees 'SAP2') should search for flights based on:

1. Flight number.
2. Date/Time of flight.
3. Departure/arrival location.
4. A staff member ID (Flights that this member is involved with).

SR4.1.2: The list should display all of these flights' information.

SR4.2: Users' Search for Flights

SR4.2.1: Any user should be able to search for any available flight registered in the system.
SR4.2.2: There are two main types of search for the user: 1) one-way 2) round-trip.
SR4.2.3: User search is basically based on:

a. Departure location.
b. Destination.

c. Date of departure.
d. Date of the return ***(*for a round-trip**)**.
e. The number of passengers.
f. Cabin class.

SR4.2.4: A list of flights should be displayed in accordance with the above parameters.

SR4.2.5: The single flight will display:

a. Flight number.
b. Date & time of departure.
c. Date & time of arrival.
d. Duration of flight.
e. price of the ticket.
f. The number of seats available.
g. Meals.

- **Order of Flights**

SR4.2.6: Tickets on sale should be displayed first as recommended.

SR4.2.7: The list should be ordered based on the best time/price correspondence.

SR4.2.8: The additional filters that must also be applicable to affect the order:

1. Price.
2. Time of the departure/arrival.
3. Duration of the total trip.
4. The number of stops.

- **Transit Trips**

SR4.2.9: If there does not exist a direct flight, transit flights shall be displayed (trip displayed as a package).

SR4.2.10: These transit flights connect the departure point to the arrival destination indirectly.

SR4.2.11: This trip package should be expanded to check the locations of stops, duration of waiting between flights, information about every flight engaged in this transit trip as described.

- **For Round-Way Trips**

SR4.2.12: Additional flights should be displayed for returning to the departure location from the arrival destination depending on the date of the return (trip displayed as a package).

SR4.2.13: This trip package should be expanded to check the information about every flight individually as described.

SR4.2.14: Any customer should be able to search for a flight's information using the Ticket ID.
SR4.2.15: If the Ticket ID is valid, the flight's information will display as indicated in (UR4-SR4.2.5).

- **UR5: Book Flights**

SR5.1: The first step is choosing the check-in baggage for the flight.
SR5.2: There are two sizes of baggage 10kg and 23kg.
SR5.3: A maximum of 2 pieces of any size of baggage is accepted to be added.
SR5.4: Next step is filling a form that contains: Full name, gender, date of birth. If the user is logged in, these parameters shall be added automatically.
SR5.5: Email address may be provided as well, but it is not necessary.
SR5.6: Payment is the last step (check UR6).
SR5.7: After a successful booking, the electronic ticket (e-ticket) shall be displayed and is able to be printed.
SR5.8: If the email address is provided, then a copy of the e-ticket is sent to the email address as a PDF form.
SR5.9: The form of the ticket is provided as an HTML page from the company.

- **UR6: Transactions (Payments)**

SR6.1: The last step of booking is the payment. Which should be done using: PayPal, VisaCard, or MasterCard.
SR6.2: The user logs into his/her PayPal account for PayPal.
SR6.3: The user enters the card's number, expiry date, and CVV for VisaCard/MasterCard.
SR6.4: The system will be connected to the bank/company associated with the chosen payment method. The bank/company will check the validity of the entered card/account and whether the balance is enough for the transaction.
SR6.5: If the balance is not enough or card/account is not valid then the system shall indicate this error to the user and the transaction will be cancelled out.
SR6.6: The System shall guarantee the security of the user's information and payment method.
SR6.7: If a flight gets cancelled the money gets refunded to the same account that the payment was made by.

2.3 Scenarios

1- Scenario By Simon Asmar - 1162643 (Booking a Flight - UR:5)

Initial assumption: The user has searched for a flight and is ready to book it.
Normal: The user clicks the check-in luggage button and then chooses one of two options “10kg” or “23kg.” The user repeats this step again making the amount of luggage check-in 2, which is the maximum.
The user then fills in a form with the correct information about his/her full name, gender, date of birth, and an optional text field for their email address.
After that, the user goes through the payment process, if that is successful an e-ticket is displayed for the user to be saved or printed. The user also gets a pdf copy of the e-ticket sent to their email address.

What can go wrong?

Alternative: The user does not provide their email address. Everything should go on normally since that part is optional.

Error: The payment process is unsuccessful. The user shall go through the payment process again, entering the correct information in all the fields in order for it to be successful.
Error: The user does not fill in all required fields in the booking form. The system should show them an error and shall not let them go to the next step before filling the fields.

Other Activities: When a user enters the flight booking page, the system must temporarily reserve a seat on the flight for him/her so that they don’t lose it by the time they fill in the form.

System state on completion: All the information provided in the flight booking form and payment form are saved in the database and the flight is booked.

2- Scenario By Sabry Alawy - 1162074 (Payments - UR:6)

Initial assumption: The user has searched for a flight and is ready to book it and he\she is in the Payment windows.
Normal: the user chooses from different types of payments: PayPal, VisaCard, or MasterCard, that has different data requirements from each other, which the user must fill to complete the booking process.

After that, when the user clicks on specific button(atomically, not concurrently with other users - we can solve it by adding control switches devices that work as queues to prevent losing data and good preforms -), The entered information will be checked from the bank/Company database, by checking the validity of the entered account and whether the balance is enough for the transaction, and the money transfer from the entered account to the company bank account, and this request of user data and the payment will be saved.

What can go wrong?

Alternative: The user must fill all the requirements that are in the specific payment type, next.

Error: If the account is not found in the bank database then the system shall indicate this error to the user and the transaction will be cancelled out.

Error: If the balance is not enough then the system shall indicate this error to the user and the transaction will be cancelled out.

Error: If a flight gets cancelled the money gets refunded to the same account that the payment was made by.

Error: The user does not fill in all required fields in the booking form. The system should show them an error and shall not let them go to the next step before filling the fields.

Other Activities: When a user clicks the confirmation button, If the queue - that is collect the waiting data to be checked or saved - is full the system shall indicate this error to the user and the transaction will be cancelled out.

System state on completion: All the information provided in the payment form are saved in the database and the money transferred to the correct account.

3- Scenario By Laith Marzouka - 1160827 (Search for a Flight - UR:4; SR:4.2)

Initial assumption: The customer enters the website and wants to consult a suitable flight for his next trip.

Normal: The customer first chooses from a checkbox if he/she wants this trip to be one way where he/she just travels to a place or a round-trip in which he/she chooses another flight to return as well (round-trip option is checked as default).

Then, the customer will search from a search field input -which contains all cities- the city he/she wants to travel from, then from another search field input chooses the city where he/she wants to travel to. After that, the customer chooses from a date field input the date of the wanted flight. In case that the customer chose a round trip, he/she fills another date field input with the return date. Finally, the customer fills another two fields, first is a number-field input in which he/she chooses the number of passengers travelling. Second, chooses the cabin class from a combo field (first class, business class, economy class).
After successfully filling all of the above fields and choosing the find flight button, the system shows all flights from-to the designated cities in the indicated days and have available seats from the selected cabin class. The flights with tickets on sale will be listed first, the order of the remaining flights will be based on time/price correspondence.
When A flight is chosen to be consulted, All details of the flight will be displayed (Flight number, date & time of departure/arrival, Duration of flight, price of the ticket, number of seats available, meals). If the trip is a round-trip, then two flights will be displayed in full details as mentioned.

What can go wrong?

Alternative: No direct flights exist from-to the designated locations. Transit trips should be displayed if possible. The transit flights connect the city of departure to the city of arrival. More than one transit flight may be applied, a single transit flight is indicated as a stop.
When the flight is chosen, the information about the transit flights will also be displayed in full details.

Alternative: The customer chooses to search by a filter (filters are located at the side of the screen):
By price: a checkbox input is selected whether to order flights by price from highest to lowest or from lowest to highest.
By the time of departure: A time interval input that can be selected and only the flights that departure at this time interval will appear.
By the time of arrival: A time interval input that can be selected and only the flights that arrive at this time interval will appear.
By the duration of the trip: A number-field input to select the maximum hours of the trip. Only the trips with a duration less than the selected hours will be displayed.
By the number of stops: A number-field input to select the maximum number of stops in the trip. Only the trips with stops less than the selected will be displayed.
Combination of filters: more than one filter of the filters mentioned above is used. Only the flights that meet all filters' conditions will be displayed.

Error: The customer missed out to fill a field or more in the searching for a flight procedure. The system should mark out these empty fields to the customer and ask him/her to fill it. Otherwise, the search will not be carried on.

Error: The customer chooses the return date -for a round-trip- before the date of the departure. The system should mark out this to the customer and not accept the search before correction.

Error: No flights exist from/to the designated location for the given dates. The system should indicate to the customer that no such flight is available at the moment.

Other Activities: The system must ask the customer to refresh the search after every five minutes to keep the integration in the system since prices/dates of flights change constantly.

System state on completion: A search record will be added to the database including the departure country, arrival country, date of departure, date of return if applicable, and the date of the search.

4- Scenario By Layth Abufarhah - 1162636 (User System Registration - UR:1; SR:1.2)

Initial assumption: the customer who concerned in booking flights is entered to the website and intending to register as a user.
Normal: user enters the website, and he doesn't have an account in the system, then from the login page, he clicks on the register button and the system should redirect the user to the registration page.
In the registration page, the user should follow a series of prompts from the system
to enter information about him/herself, full name, gender, date of birth, email address and password.

What can go wrong?

Alternative: a user enters an invalid data, the system should not get the inputs and
notify the user about this to re-fill the fields by showing an alert.
Alternative: a user enters information which is already on the database, the system
should not register the user and notify him/her about this to re-fill the fields by showing an alert.

Other Activities: The system can verify the email address entered by sending a

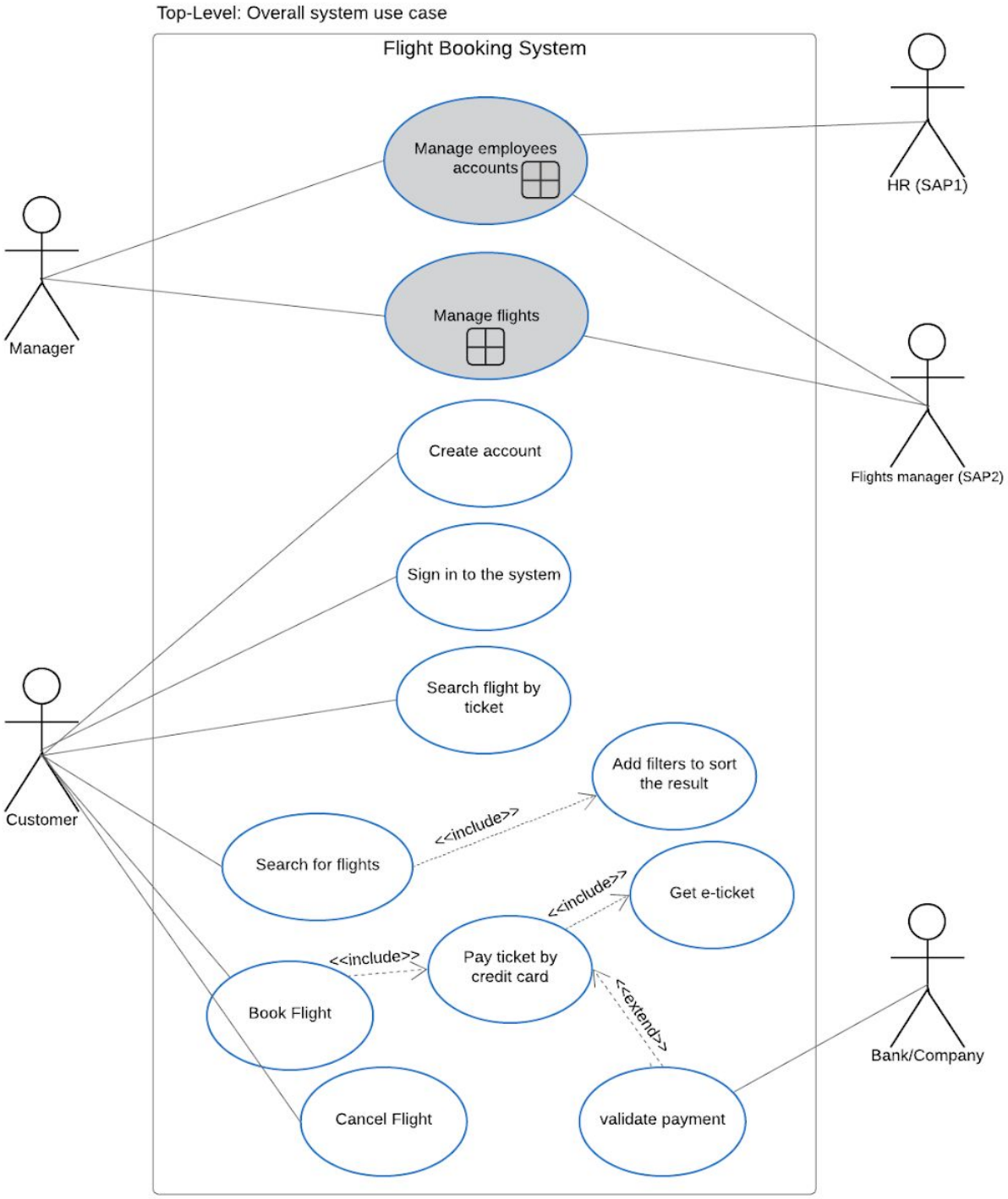
verification link to the email.

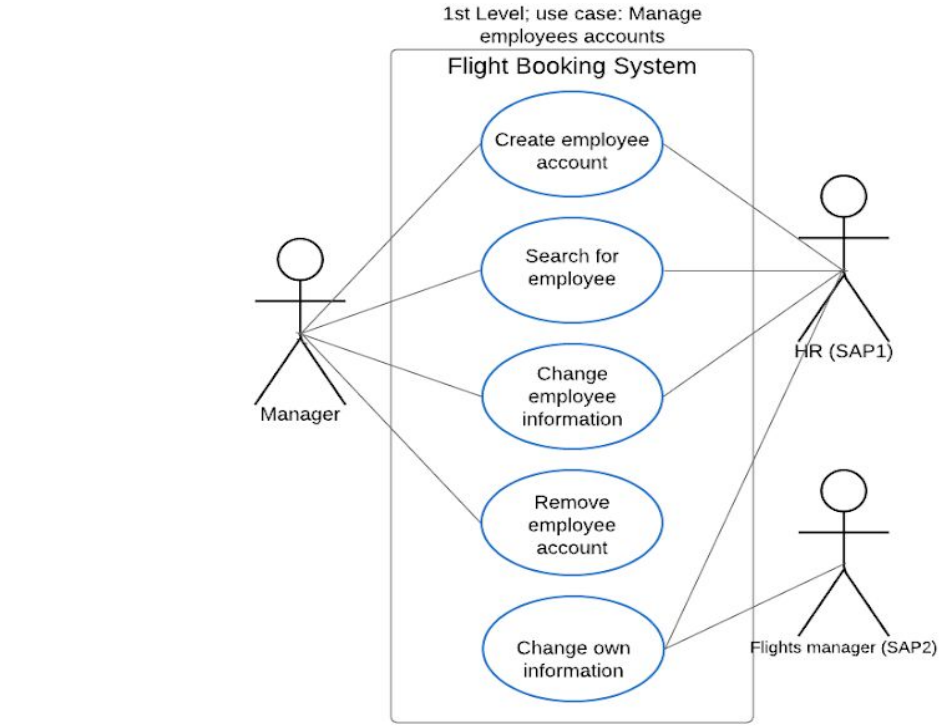
System state on completion: User logged in, all information entered in the database and user should be redirected to the main page.

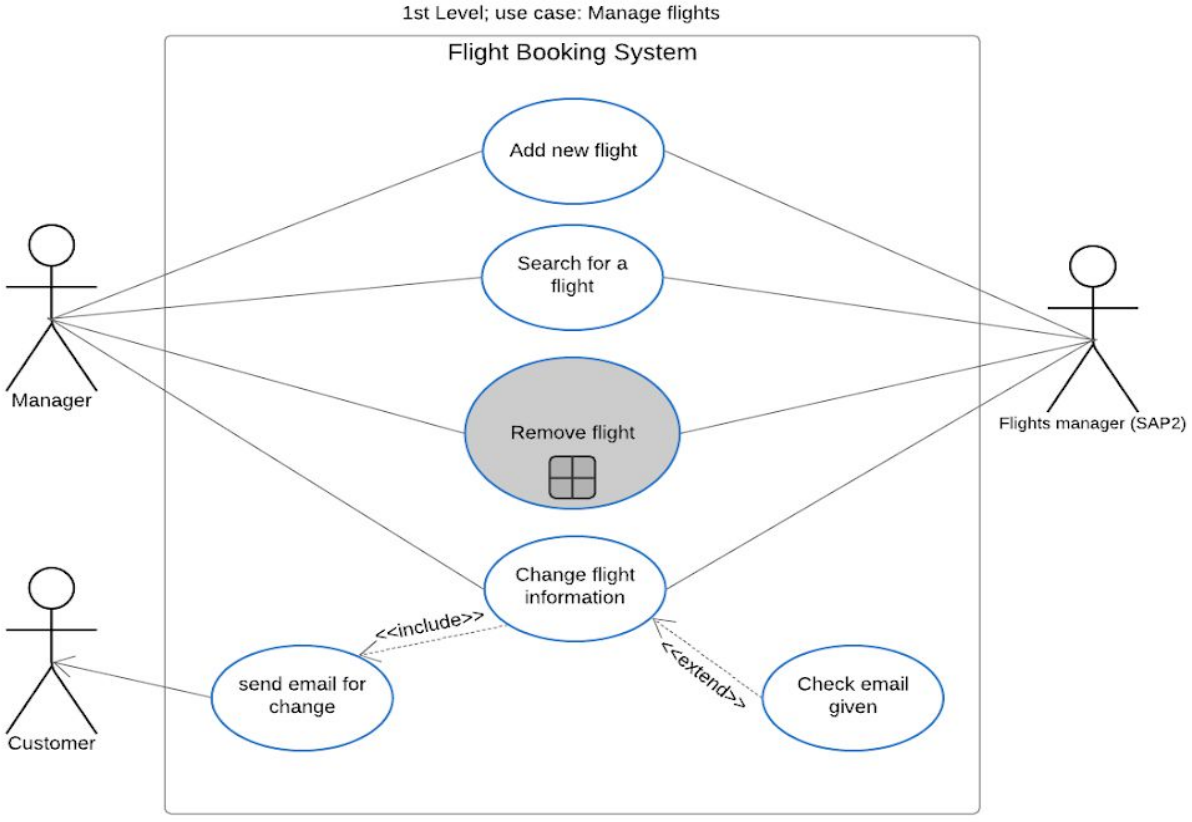
2.4: ACTORS analysis and their description

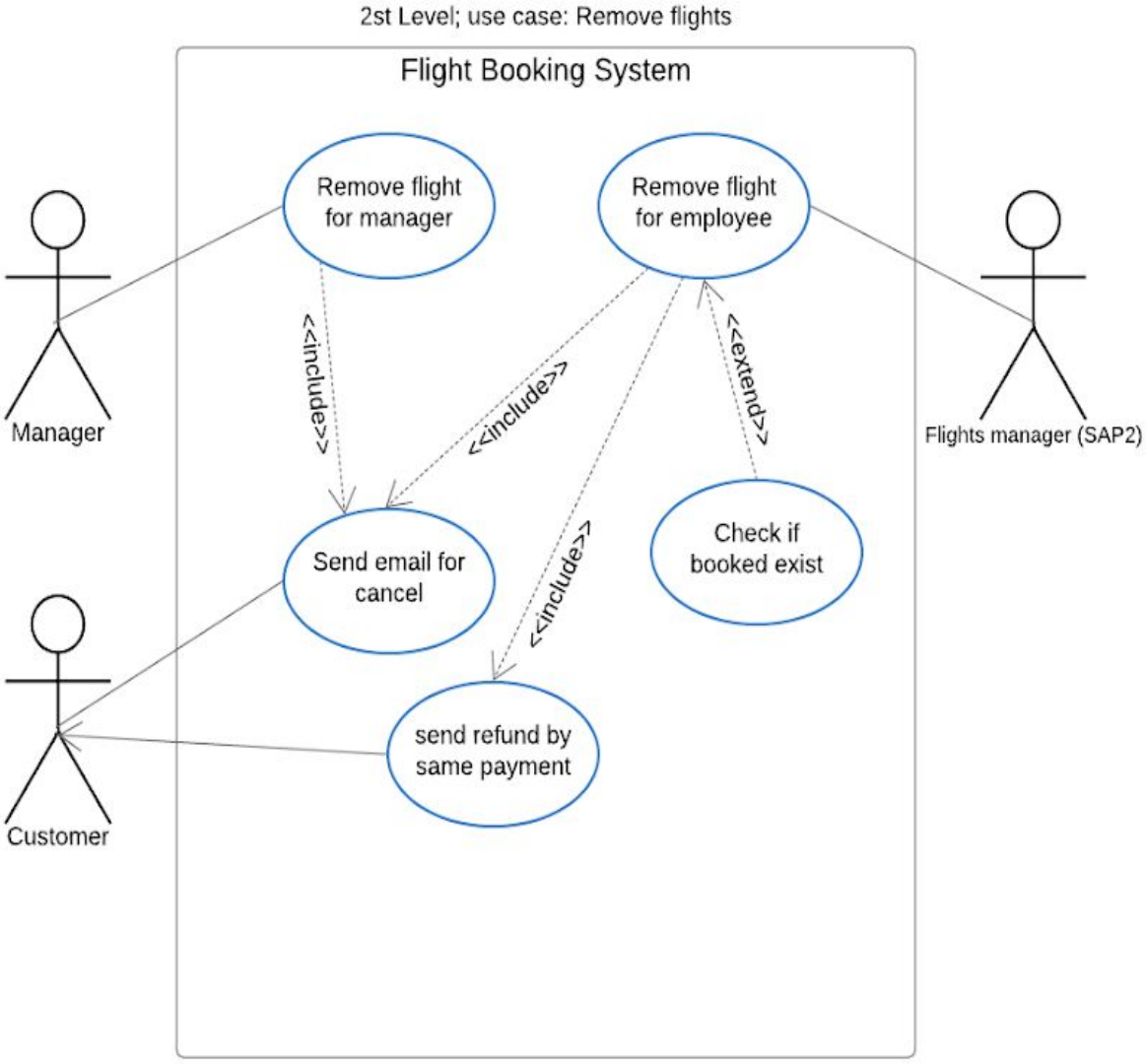
ACTOR	Semantics/Description
Manager	This actor represents someone who is an employer of the flight company, that can manage employees/flights information, and add/edit/view job titles
HR employee 'SAP1'	This actor represents someone who is an employee of the flight company, that can view, add, and edit employees' information
Flights manager 'SAP2'	This actor represents someone who is an employee of the flight company, that can view, add, edit flights' information
Customer	This actor represents someone no matter registered or not, that makes use of the website features by searching and booking flights
Bank	This actor represents an external system which verifies customers' accounts and complete payment transactions.

2.5: USE-CASE Diagram




1st Level; use case: Manage employees accounts
Flight Booking System
Create employee account
Search for employee
Change employee information
Remove employee account
Change own information
Manager
HR (SAP1)
Flights manager (SAP2)


1st Level; use case: Manage flights
Flight Booking System
Add new flight
Search for a flight
Remove flight
Change flight information
send email for change
Check email given
<<include>>
<<extend>>
Manager
Customer
Flights manager (SAP2)


2st Level; use case: Remove flights
Flight Booking System
Remove flight for manager
Remove flight for employee
Send email for cancel
Check if booked exist
send refund by same payment
<<include>>
<<include>>
<<include>>
<<extend>>
Manager
Customer
Flights manager (SAP2)

2.6: Detailed Description of key USE-CASES:

1- Detailed USE-CASE By Simon Asmar - 1162643 (Booking a Flight - UR:5)

Actors	Customer
Description	Customer may book a flight that is not fully booked.
Pre-conditions	1. The customer searched for the flight based on the date and location. 2. The flight is available (not fully booked).
Sequence/Flow of Events	1. The customer chooses the number of luggage (max 2). 2. The customer chooses one of two weight options for each luggage (10kg & 25kg). 3. If not logged in the customer fills in a form with personal information manually. If the customer is logged in this form is filled automatically. 4. The customer goes through the payment process. 5. If the customer goes through the payment process successfully an e-ticket is displayed for the customer and a pdf copy of it is sent to their email if it was given, if not successful the customer must go through the payment process again.
Data	Luggage information (quantity and weight), customer information (full name, gender, date of birth, and [optional] email address).
Stimulus/Trigger	User command issued by the customer.
Post-conditions/ Response	1. The system has updated the number of available seats on the flight, if successful. 2. Customer information and e-ticket added to the database, if successful.
Comments	No comment

2- Detailed USE-CASE By Sabry Alawy - 1162074 (Payments - UR:6)

Actors	Customer, Bank/Company.
Description	The customer wants to pay for the ticket
Pre-conditions	1. The customer searched for the flight based on the date and location. 2. The flight is available (not fully booked). 3. Finishing the first requirement for booking.
Sequence/Flow of Events	1. the user chooses from different types of payments: PayPal, VisaCard, or MasterCard. 2. the user must fill the required fields to complete the booking process 3. The entered information will be checked from the bank/Company database. 4. Checking the validity of the entered account and whether the balance is enough for the transaction. 5. If the balance is not enough then the system shall indicate this error to the user and the transaction will be cancelled out. 6. If the account is not found in the bank database then the system shall indicate this error to the user and the transaction will be cancelled out. 7. If a flight gets cancelled the money gets refunded to the same account that the payment was made by. 8. The user does not fill in all required fields in the booking form. The system should show them an error and shall not let them go to the next step before filling the fields.
Data	1. For Visa card and master card customer information (full name, card number, date of exp., CVC). 2. For PayPal check if valid account.
Post-conditions/ Response	All the information provided in the payment form is saved in the database and the money transferred to the correct account.
Comments	No comment

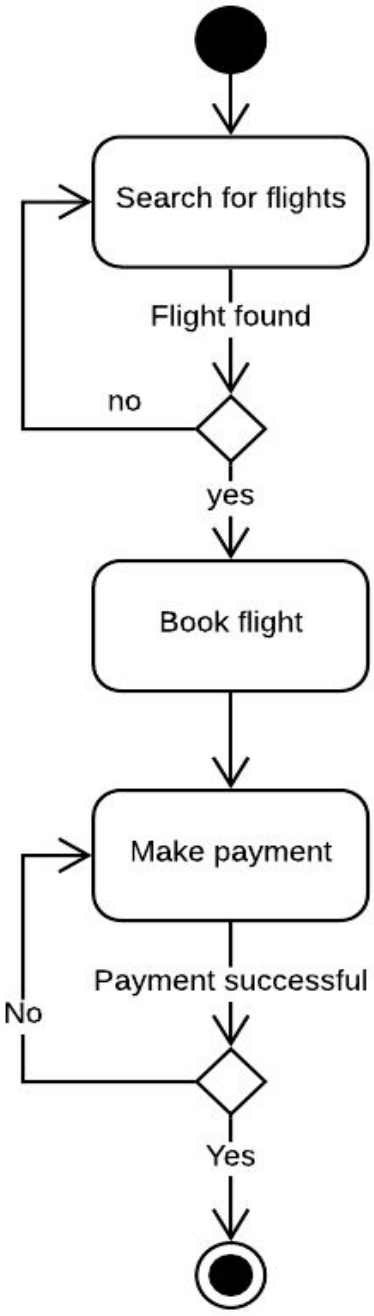
3- Detailed USE-CASE By Laith Marzouka - 1160827 (Search for a Flight - UR:4; SR. 4.2)

Actors	Customer
Description	A customer may search for a flight based on the destinations and dates
Pre-conditions	Only flights with available tickets shall appear
Sequence/Flow of Events	1. The customer chooses whether the trip is going to be one-way or round trip. 2. The customer chooses the cities of both departure and arrival. 3. The customer chooses the date of departure. 4. If the trip is a round trip type then the customer chooses the return date as well. 5. The customer chooses the number of travellers and the cabin class. 6. The customer presses the search button. 7. The customer may choose filters to sort the search result. 8. The results can be sorted by price, time, duration, the number of stops, and a combination of any of the latter filters.
Data	Flights' information (number, price, time, destination, duration, meal).
Stimulus/Trigger	User command issued by the customer.
Post-conditions/ Response	The system saves the search parameters (locations, dates) to the database.
Comments	Any user whether registered or not may search for any flight.

4- Detailed USE-CASE By Layth Abufarhah - 1162636 (User System Registration - UR:1; SR:1.2)

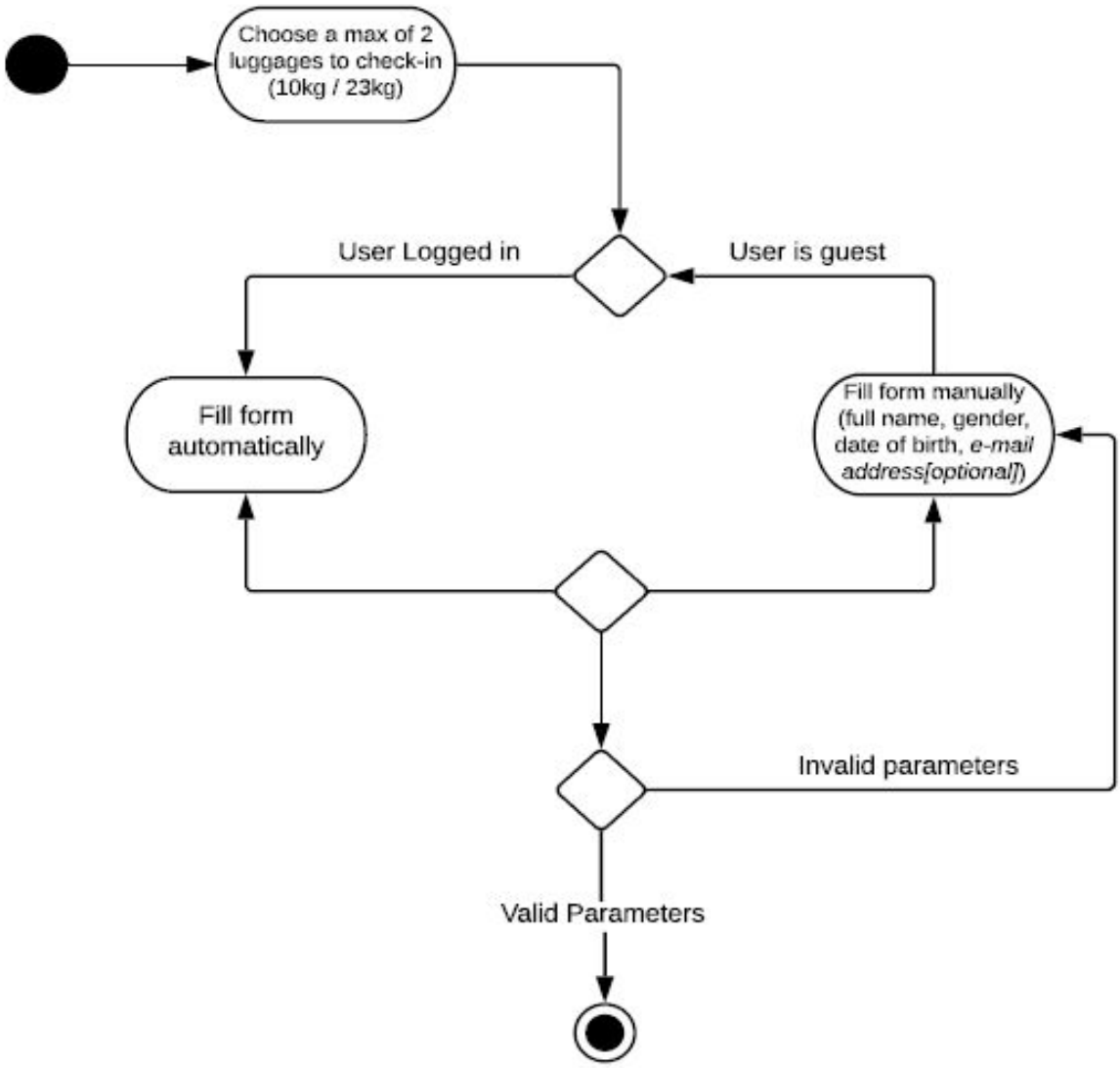
Actors	Customer
Description	A customer may register as a user in the system
Pre-conditions	The customer doesn't have an account.
Sequence/Flow of Events	1. The customer enters the website. 2. The customer clicks on the registration button. 3. The customer enters all information required. 4. The customer confirms the entered information. 5. The customer verifies the email address.
Data	User's information (full name, gender, date of birth, email address and password).
Stimulus/Trigger	User command issued by the customer.
Post-conditions/Response	The system saves the user's information to the database.
Comments	The email address shall be verified.

2.7: Activity Diagram

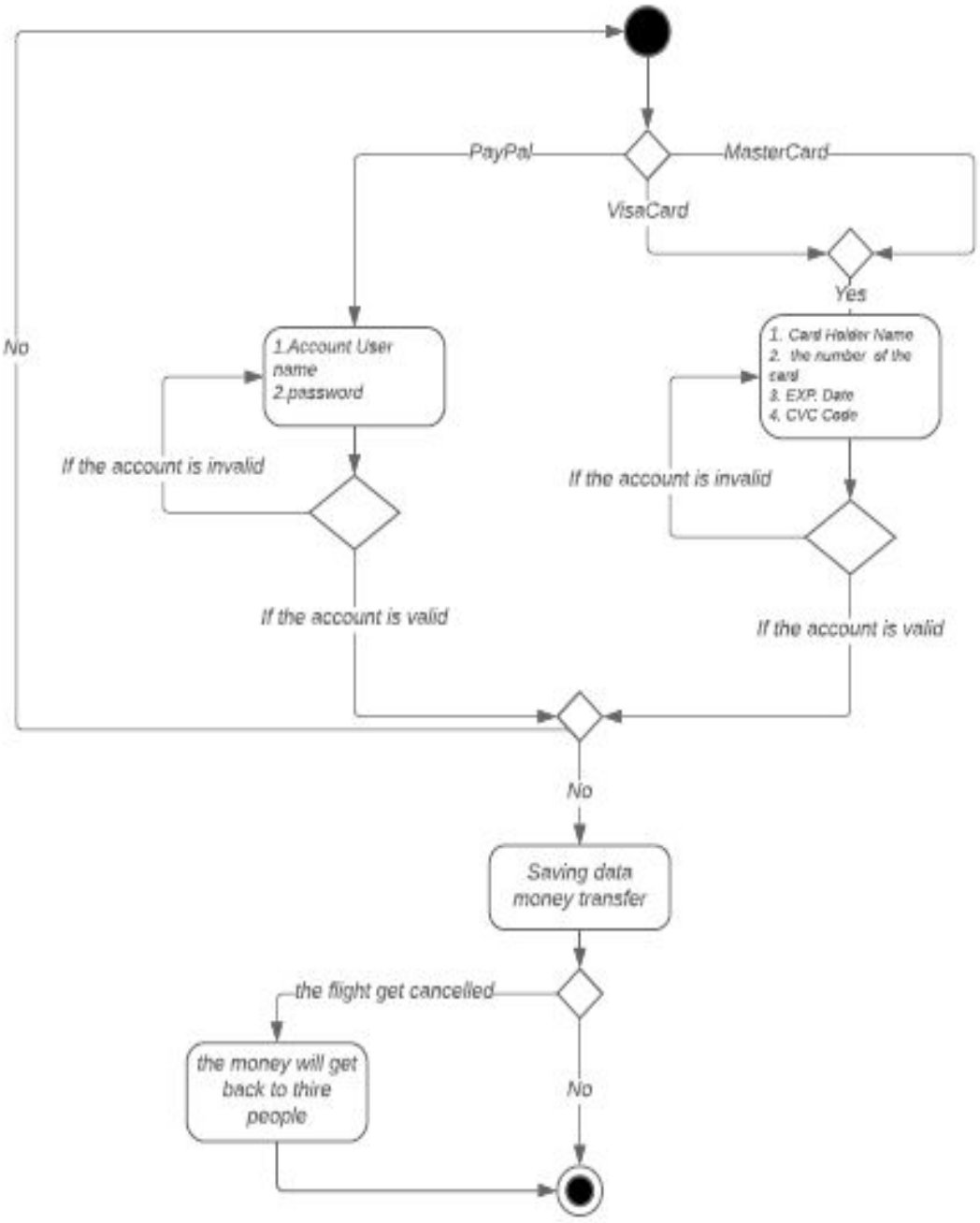


2.8: Instance Activity Diagrams

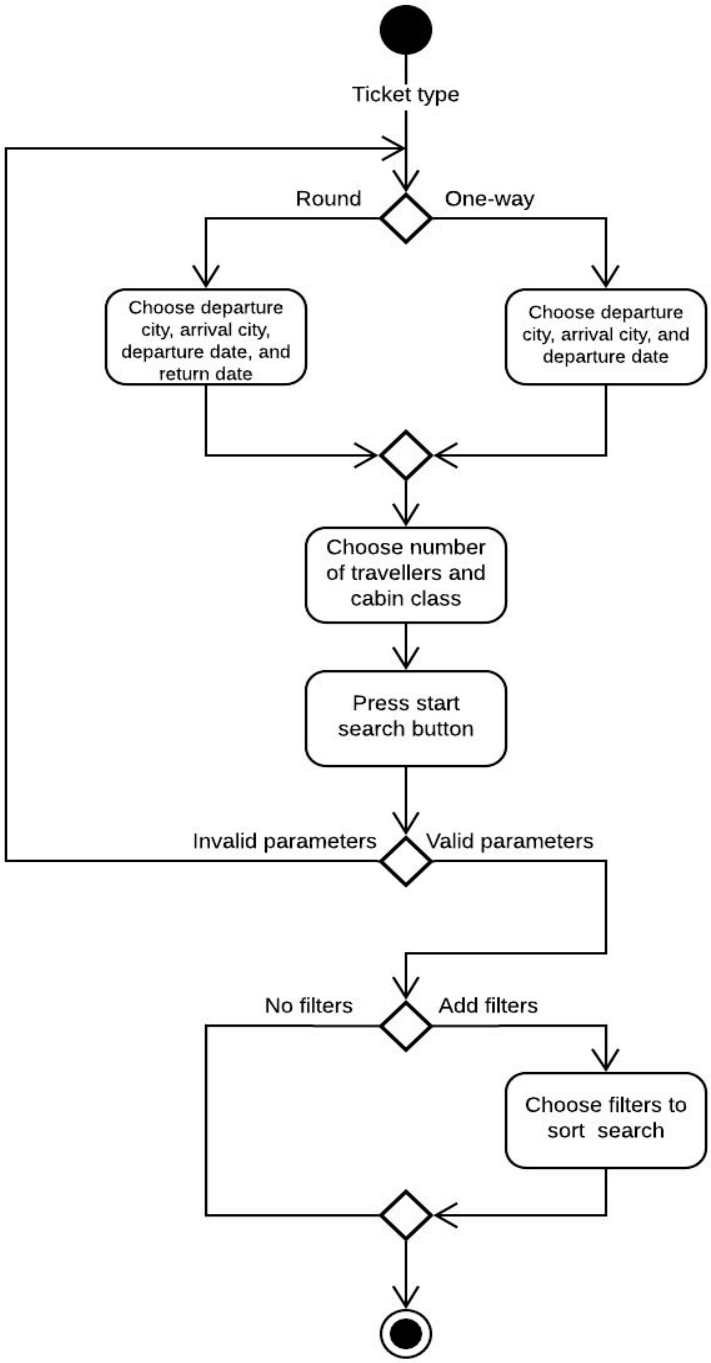
1- Activity Diagram By Simon Asmar - 1162643 (Booking a Flight - UR:5)



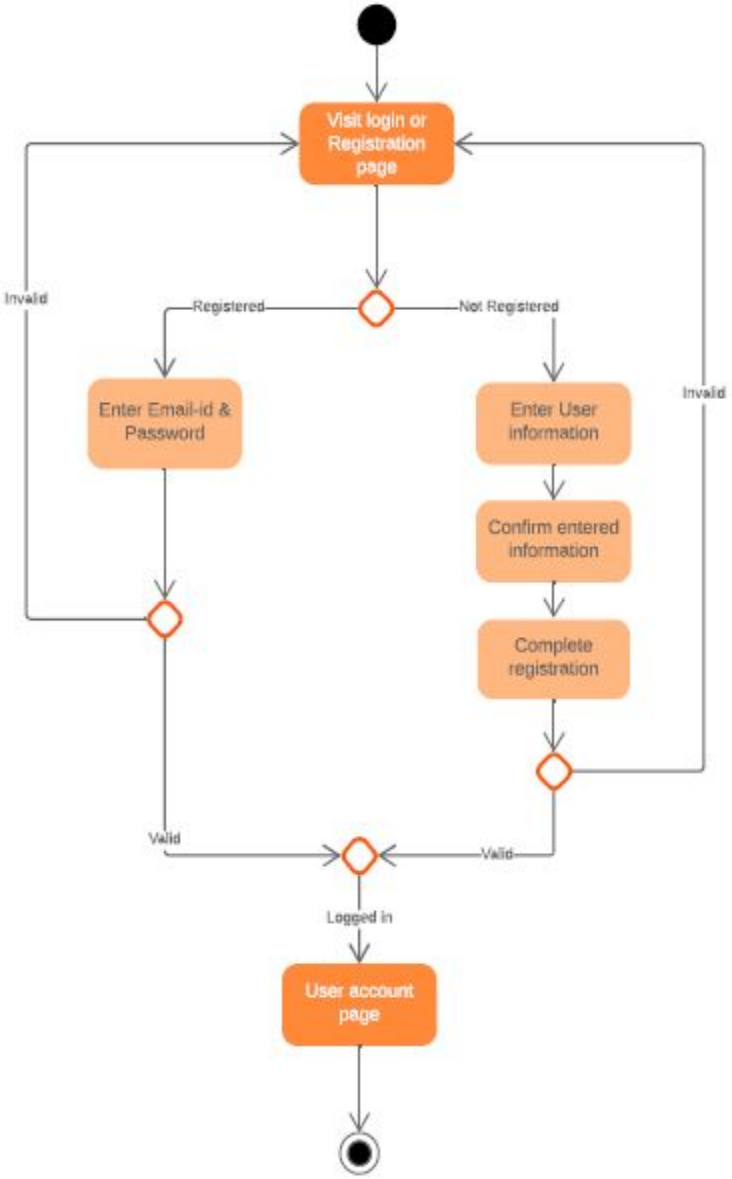
2- Activity Diagram By Sabry Alawy - 1162074 (Payments - UR:6)



3- Activity Diagram By Laith Marzouka - 1160827 (Search for a Flight - UR:4; SR. 4.2)

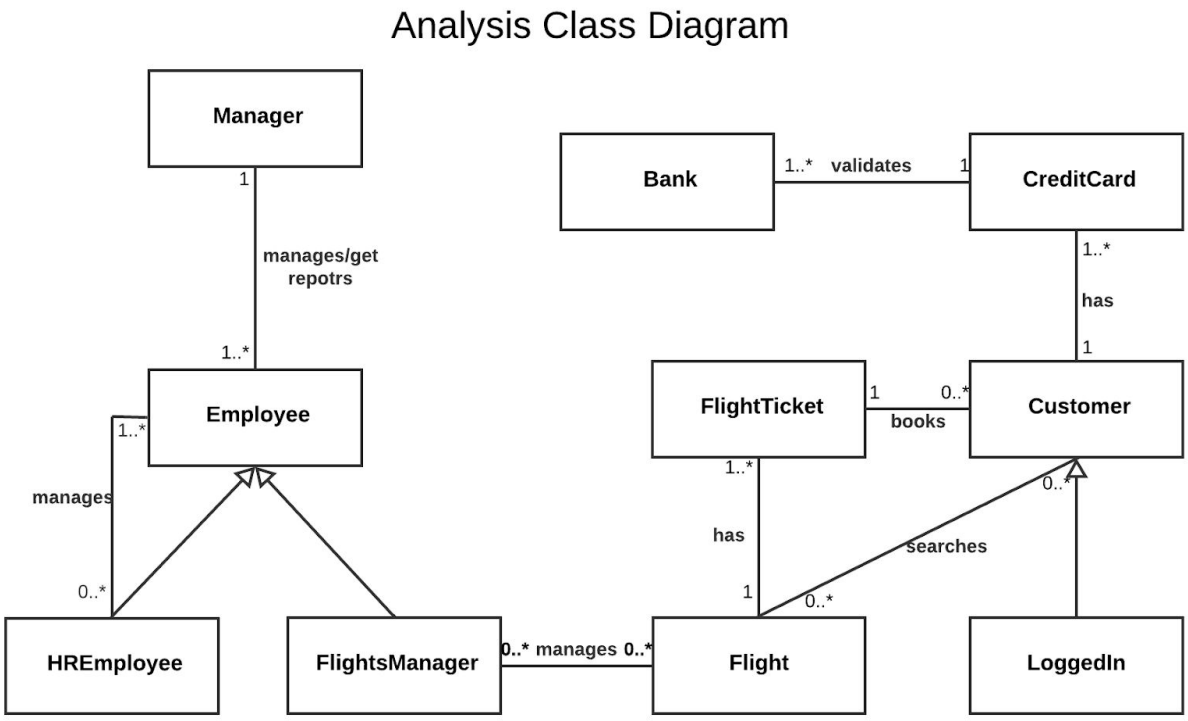


4- Activity Diagram By Layth Abufarhah - 1162636 (User System Registration - UR:1; SR:1.2)

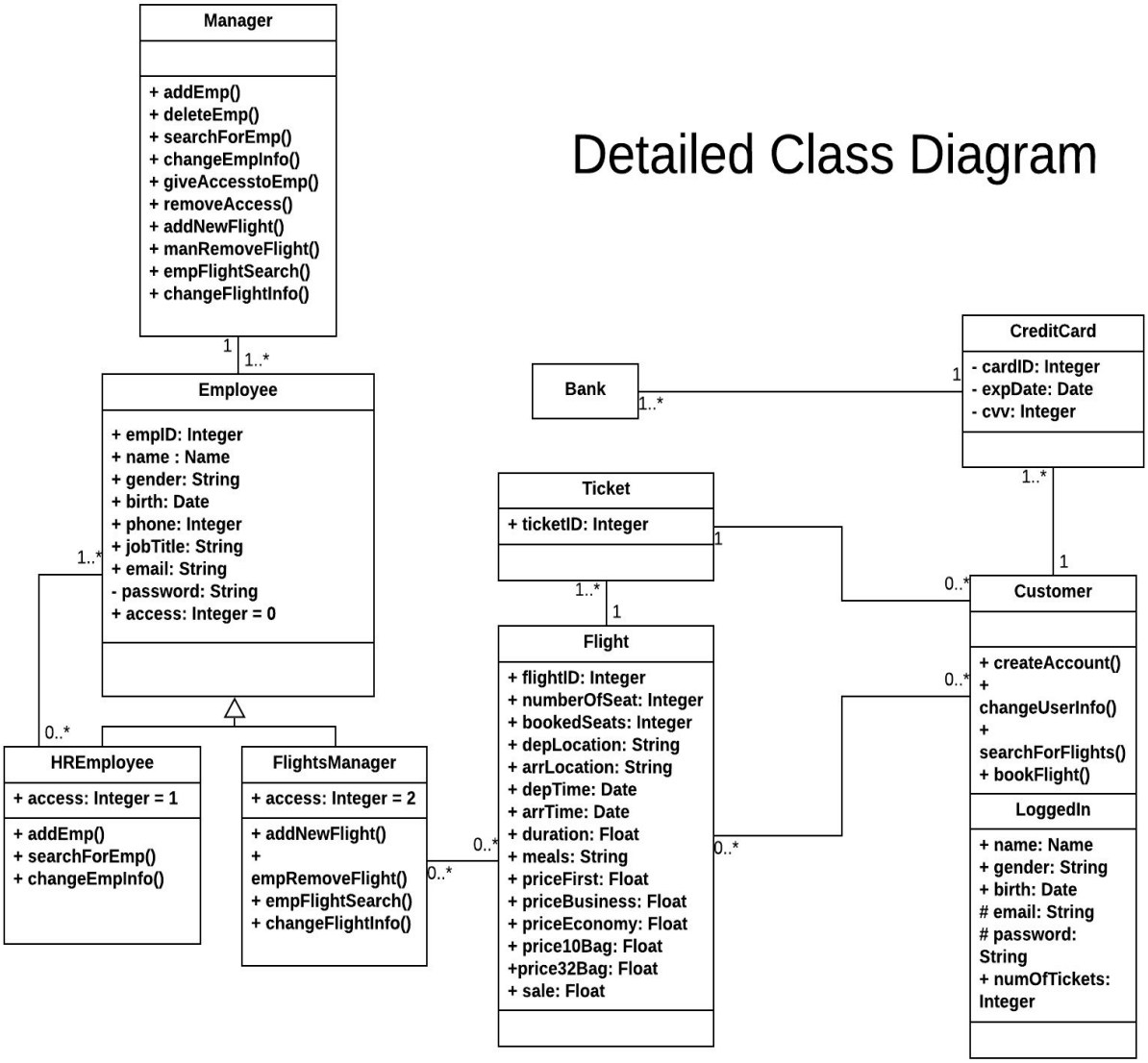


Chapter 3: System Analysis and Modelling

3.1: System CLASS Diagrams



Detailed Class Diagram



3.2: Description of Classes

Flight: this class represents the flight which was offered by the system, it must have an id, price, duration date, and the maximum number of tickets.

Ticket: this class represents the ticket of a flight which the customer books and it must have an id.

Customer: this class represents the customer or the user of the system who searches for flights and books a ticket, the customer must have a credit card.

LoggedIn: This class represents the customer that has an account and is logged in to the system.

Manager: this class represents the manager or the admin of the system who can check for all financial records and manage employees/flights information, and add/edit/view job titles.

Employee: this class represents the employee who is employed and managed by the manager there are two “sub” classes of this class.

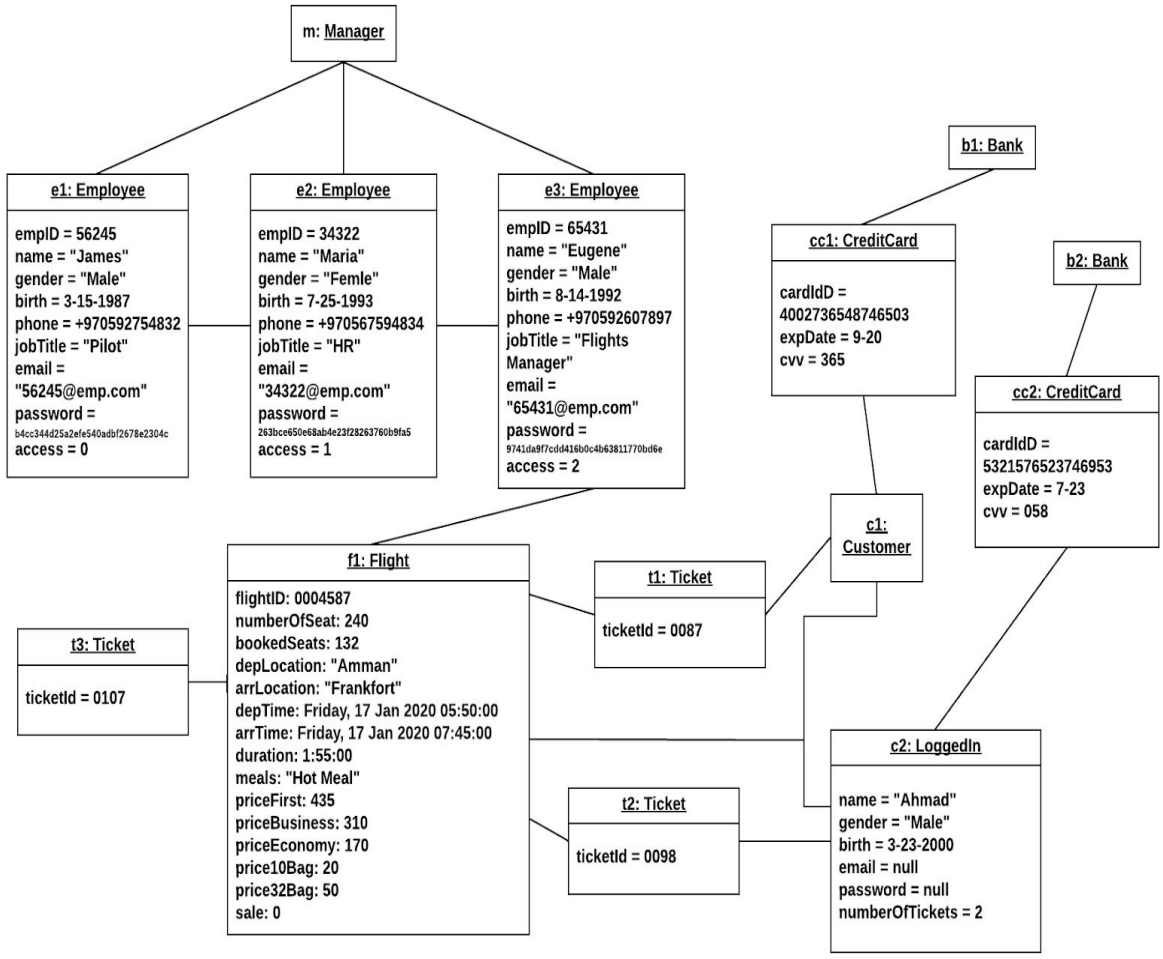
HREmployee: this class represents the employee who can view, add and edit employee’s information.

FlightsManager: this class represents the employee who can view, add and edit flight’s information.

Bank: this class represents the bank “an external system” which validates customer’s credit card and complete payment transactions

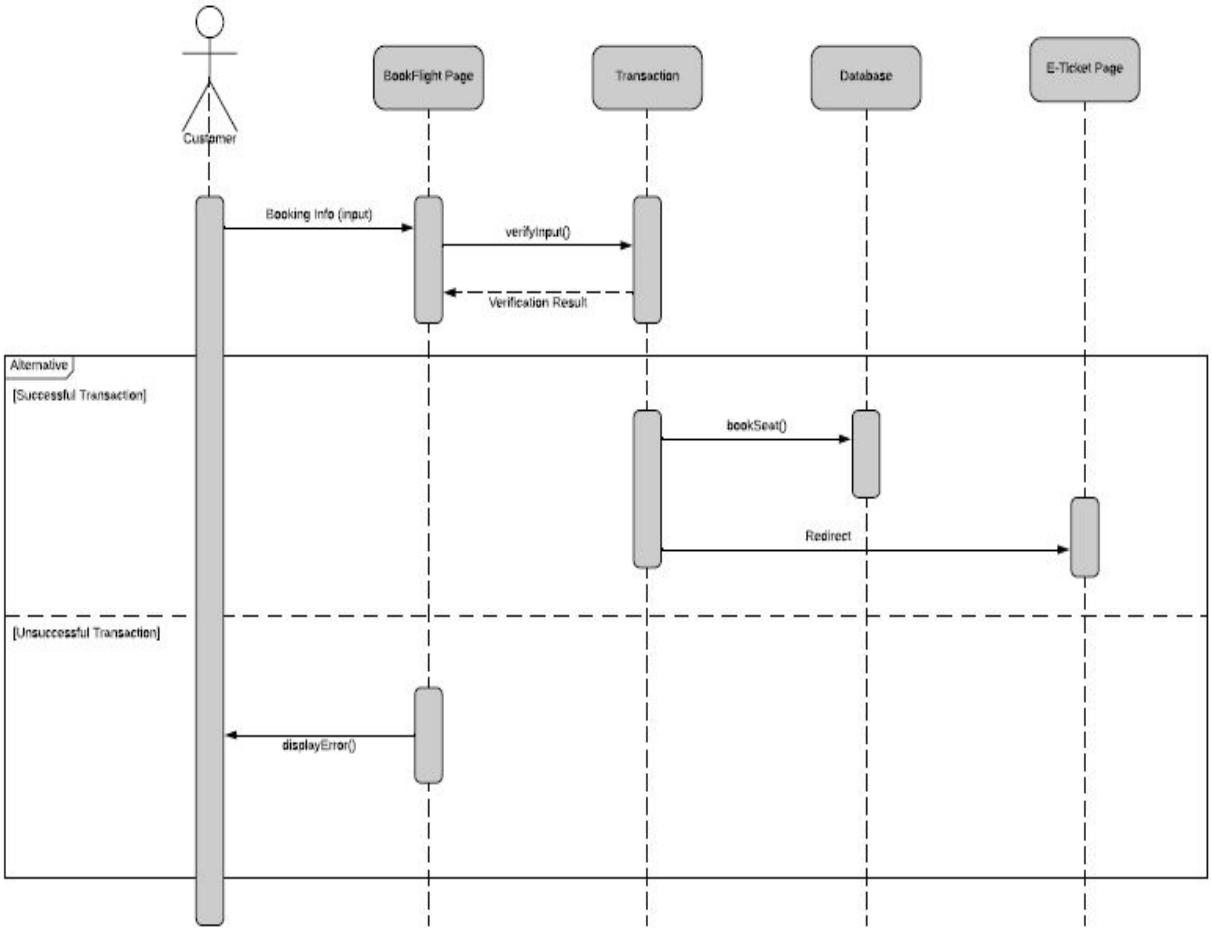
CreditCard: this class represents the credit card which must be held by the customer and validated by the bank.

3.3: Object Diagram

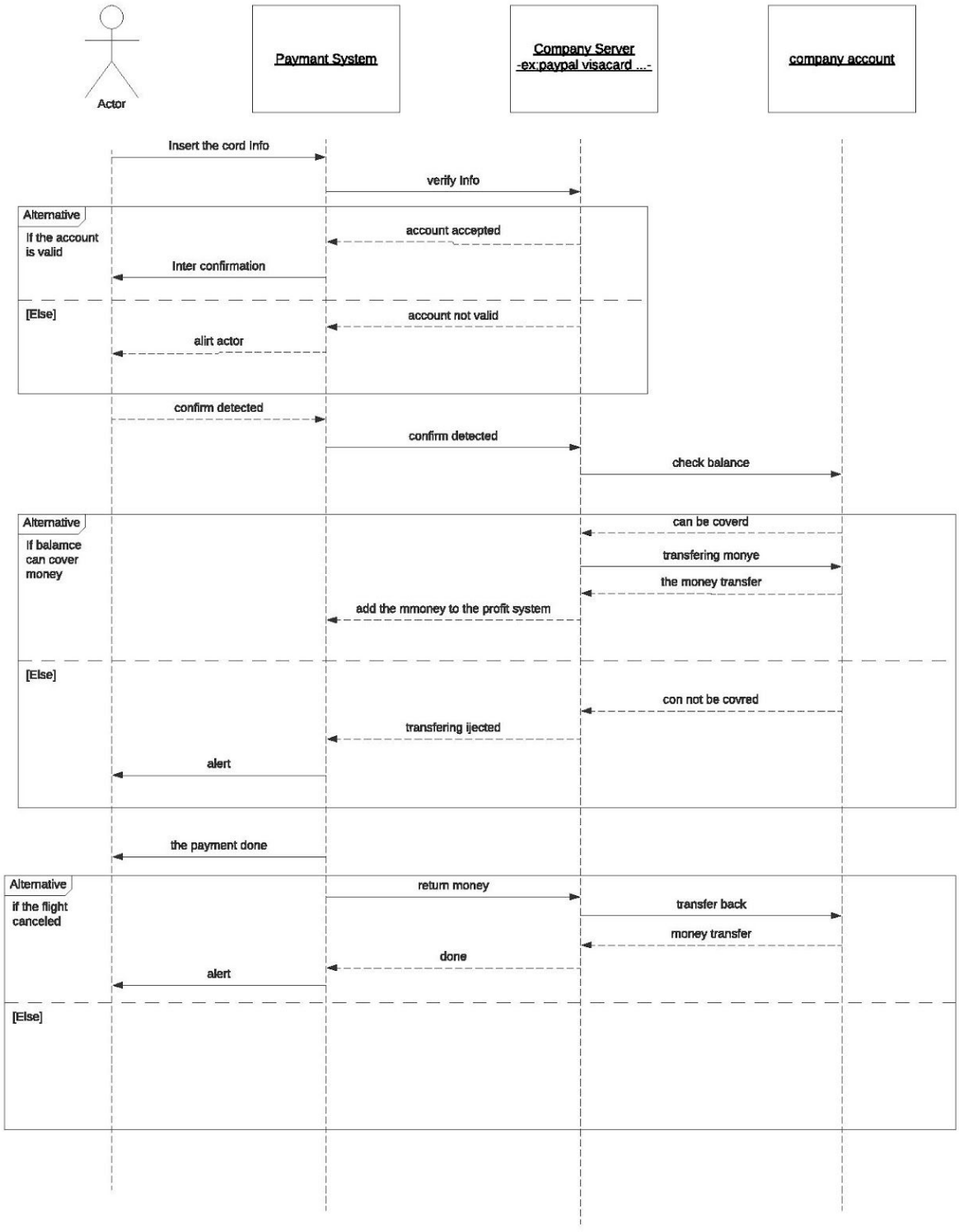


3.4: Sequence Diagrams

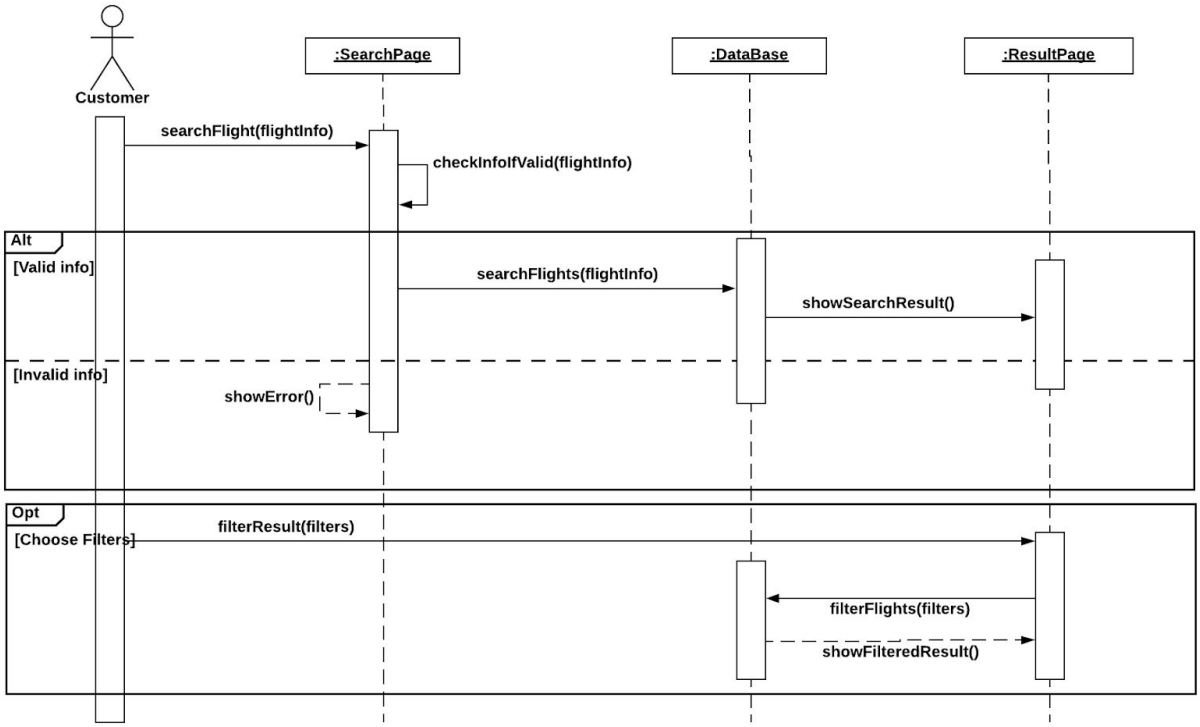
1- Sequence Diagram By Simon Asmar - 1162643 (Booking a Flight - UR:5)



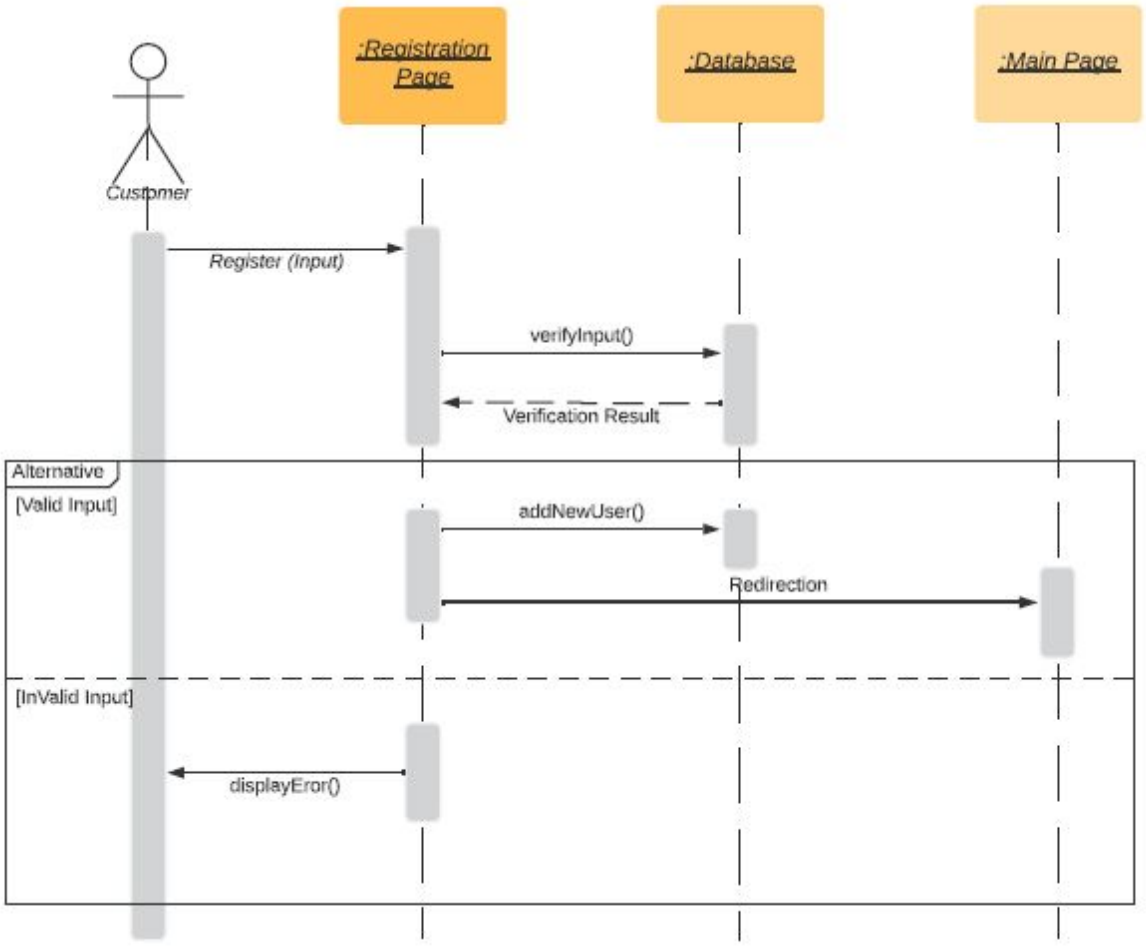
2- Sequence Diagram By Sabry Alawy - 1162074 (Payments - UR:6)



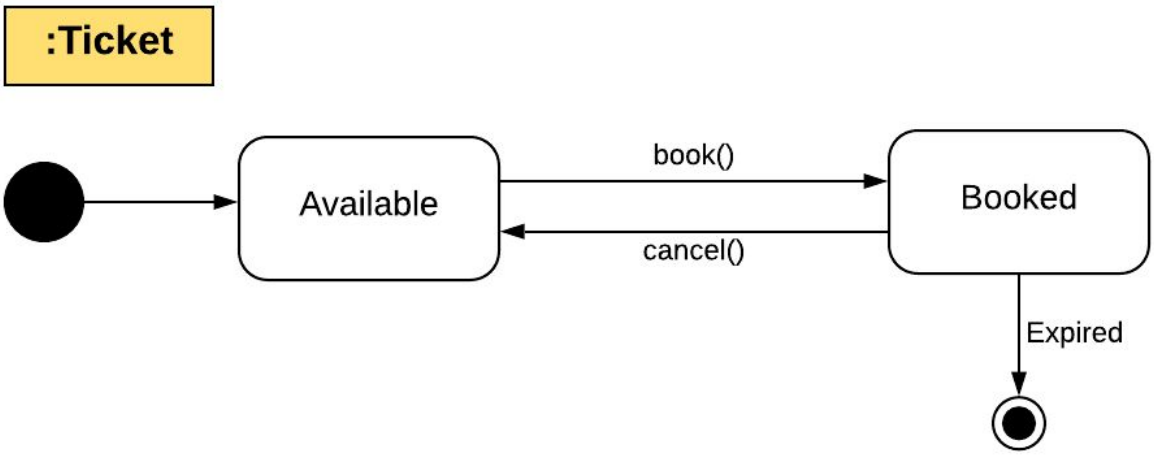
3- Sequence Diagram By Laith Marzouka - 1160827 (Search for a Flight - UR:4; SR. 4.2)



4- Sequence Diagram By Layth Abufarhah - 1162636 (User System Registration - UR:1; SR:1.2)



3.5: State Diagram



Chapter 4: System Design

4.1 Description of Design Goals

Goal 1: High Cohesion

We shall try keeping the classes and methods that are related to each other in a single place so that they form a logically single, atomic component. As an example, we combined the search engine along with the booking and payment functionalities in one component since they are highly dependent.

Goal 2: Low Coupling

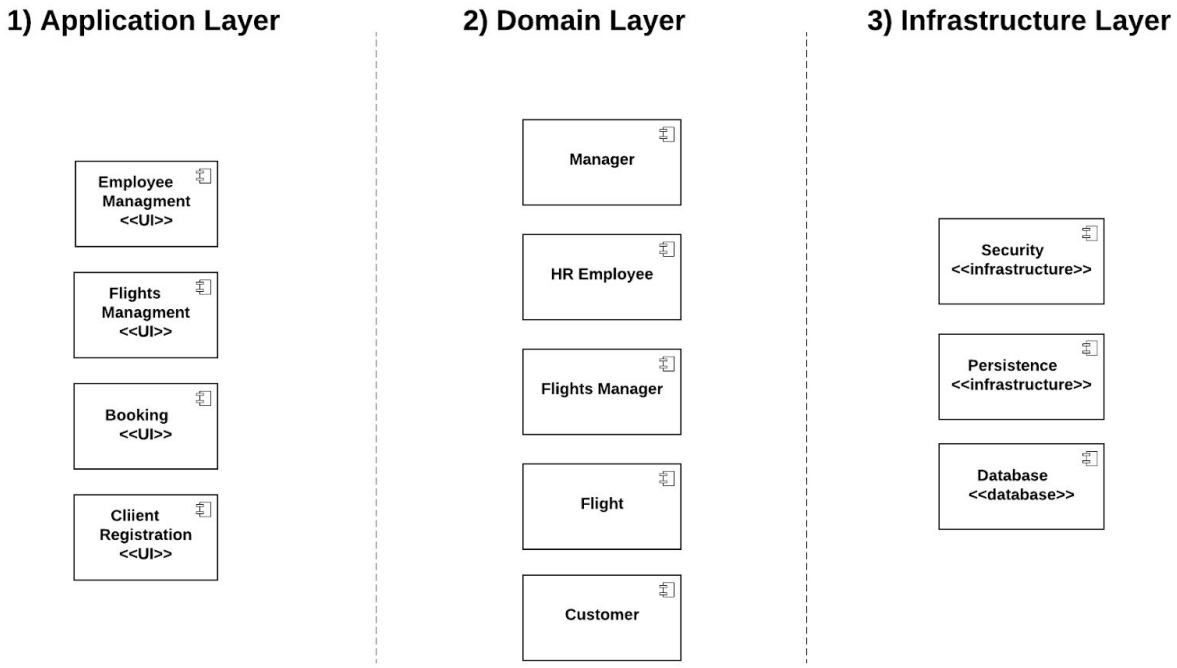
We shall try separating unrelated classes and methods as much as possible to make a single unit independent from others by minimizing the number of connections between two or more component. As an example, we separated the employee component into two individual components (HR, flight manager) since they work independently and irrelative to each other.

Goal 3: Robustness

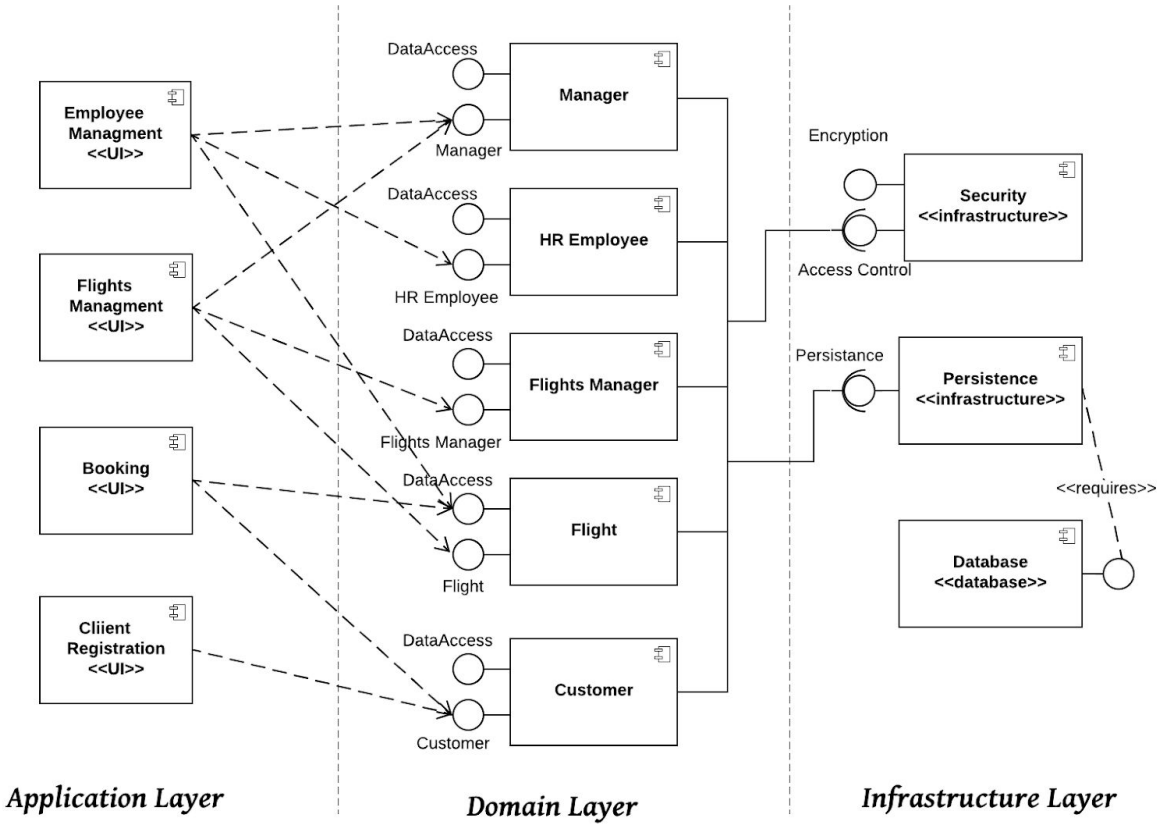
Our project is critical to the customer's profit and any failure shall cost them money and question their reliability (data lost/manipulated ...etc). So, our system must be as failure-tolerant as possible to all situations of failure. The servers used will be of good quality, and a back-up web and database servers which are located in a different place shall be provided as well.

4.2: Overall Architectural Design

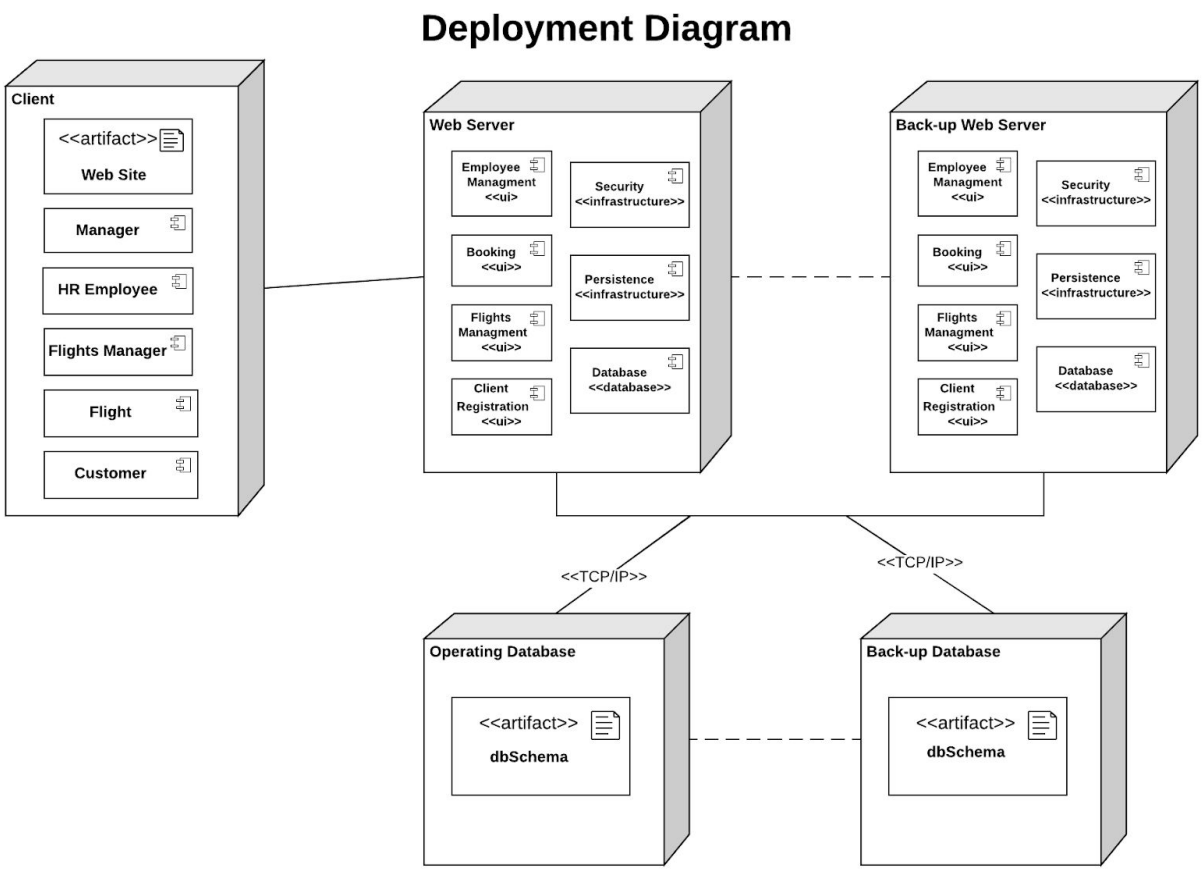
We preferred to use layered architecture in our project. Since our project is a website, the layered architecture allows us to reuse the components in the future, especially the infrastructural components.
The layered architecture enables us to make the website loosely coupled. As well as, it helps us keep related components together which increases the cohesion. As a result, the website will be highly maintainable.
Thus, this architectural design suits our design goals the best.



4.3: Component Diagram



4.4: Deployment Diagram



Chapter 5: Assessment and Effort Estimation

5.1: Assessment as developers (G9: Flight Booking)

5.1.1: User Requirement Approval

Our customers had a different view of our project as they wanted the system to be only for a specific company. However, even that we didn't like the idea initially, it turned out that it is even a better sloid idea. They gave us a detailed description of what they were looking for in such a system which helped us develop the requirements easily.
It wasn't hard getting in touch with the customer's group to ask questions and enquire about ambiguous areas since they were happy to help. For this, we especially thank customer Noura Abulaban who was very active in helping us and keeping in touch all the time during that phase.

5.1.2: Requirement Analysis

This phase slid smoothly with the customer's group, there wasn't much to ask from them really since they have already done a good job describing the business. They told us the general scenario and it was easy to induce all of the scenarios and use-cases. They did a good job checking the diagrams as they have corrected us a couple of times especially in the use-case diagrams.

5.1.3: System Modelling and Design

In this phase, there was roughly any collaboration with the customer's group, but that is not us or them to blame. We had only one quick meeting with one of them online since no meetings were applicable in the class during this phase. On the other side, We believe that they have already done their part and their help in this phase can be negligible.

5.2: Assessment as Customers (G2: Sandwich Shop)

5.2.1: User Requirement Approval

In general, The requirements given were impressive. Nevertheless, there was plenty of mistakes in the first two drafts of the requirements. We efficiently negotiated in the class meetings and the final version was really good but unfortunately, we had to do some changes since we had a hard time coming to an agreement with the price. There was a noticeable lack of communications with this group since it was nearly impossible to meet them out of the class due to a variety of restrictions. We even tried to reach out to them via Ritaj, but it took them forever to respond and it was a waste of time. Most importantly, we were unable to get a copy of the requirements (not e-copy nor hardcopy) despite that we've asked them a couple of times.

5.2.2: Requirement Analysis

We can't deny the passion they had during this phase, they worked really hard and were able to accomplish every task on time. Their draft diagrams surely had some mistakes but they were insistent on correcting everything at the moment. Despite the fact that we couldn't have many meetings but we used our times efficiently and we think that both of our groups did a very good job to put everything in order.

5.2.3: System Modelling and Design

During this phase, we couldn't have any meetings in the class due to the recent problems at the university. Also, we have already described how impossible it was to contact the customer's group online. So, we had no contact during this phase.

5.3: Effort/Time estimation calculation

	Estimated Effort	Estimated # of Developers	Total Effort
UR1	2pw	2	2 * 2 = 4pw
UR2	3pw	3	3 * 3 = 9pw
UR3	1pw	1	1 * 1 = 1pw
UR4	3pw	3	3 * 3 = 9pw
UR5	2pw	2	2 * 2 = 4pw
UR6	1pw	1	1 * 1 = 1pw
Total Effort/Avg	12pw	12/6 = 2 dev on avg needed	28pw
Schedule Time 30%	12 * 1.30 = 16w (min time)		28 * 1.3 = 37w (max time)
Cost		Avg Salary: $300	$300 * 37w = $11,100
Profit Margin (min: 20%) (max: 50%)		Min Cost → Max Cost →	$11,100 * 1.20 = $13,320 $11,100 * 1.50 = $16,650

Agreed: $17,000
34 weeks

Appendix

Meeting Minutes

A.1: Meetings of our Group

Date	Type	Attendants	Outcome
Tue, Sept. 24	Face-to-Face	Simon, Sabry Laith, Layth	Team & Roles Formed
Mon, Sept. 30	Online (messenger)	Simon, Sabry Laith, Layth	Business Selection
Thu, Oct. 10	Face-to-Face	Simon Laith, Layth	Business outline discussed
Mon, Oct. 14	Video Conference	Simon, Sabry Laith, Layth	Business outline completed
Tue, Oct. 22	Face-to-Face	Simon, Sabry Laith	User Requirements discussed + Customer's Bussiness Discussed
Wed, Oct. 23	Online (messenger)	Simon, Sabry Laith, Layth	User Requirements discussed
Mon, Oct. 28	Online (messenger)	Simon, Sabry Layth	User + System Requirements discussed
Wed, Oct. 30	Face-to-Face	Simon, Laith	Effort/Time Estimation + Customer's

			Business Discusesed
Thu, Oct. 31	**Face-to-Face**	**Simon Laith, Layth**	**User + System Requirements completed**
Sat, Nov. 9	**Video Conference**	**Simon, Sabry Laith, Layth**	**Phase Assessment**
Wed, Nov. 13	**Online (messenger)**	**Simon, Sabry Laith, Layth**	**Actors & Overall Use-Cases (draft)**
Wed, Nov. 20	**Online (messenger)**	**Laith, Layth**	**Actors & Overall Use-Cases (final)**
Thu, Nov. 21	**Face-to-Face**	**Simon Laith, Layth**	**Activity Diagram (Draft)**
Wed, Nov. 27	**Online (messenger)**	**Simon, Sabry Laith, Layth**	**Analysis Class Diagram**
Thu, Nov. 28	**Face-to-Face**	**Laith, Layth**	**Detailed & Object Class Diagrams**
Thu, Dec. 5	**Face-to-Face**	**Simon, Sabry Laith, Layth**	**State Diagram**
Tue, Dec. 10	**Face-to-Face**	**Sabry Laith, Layth**	**Component & Architectural Diagrams (Draft) + Design Goals**
Thu, Dec. 12	**Online (messenger)**	**Simon Laith, Layth**	**Component & Architectural Diagrams (Final) + Deployment Diagram (Draft)**

Sat, Dec. 21	Online (messenger)	Simon, Sabry Laith, Layth	Checking and getting everything together

A.2: Meetings with Customer's Group[2]

Date	Type	Attendants	Outcome
Tue, Oct. 10	Face-to-Face	Laith, Simon	Business outline discussed
Tue, Oct. 15	Face-to-Face	Laith, Simon	Business outline discussed
Tue, Oct. 22	Face-to-Face	Laith, Simon	User Requirements discussed
Thu, Oct. 24	Face-to-Face	Laith, Simon	User Requirements discussed
Tue, Oct. 29	Face-to-Face	Laith, Simon	System Requirements discussed
Thu, Oct. 31	Face-to-Face	Laith, Simon	User + System Requirements discussed
Tue, Nov. 12	Face-to-Face	Laith, Simon	Scenarios Discussed
Thu, Nov. 14	Face-to-Face	Laith, Simon	Overall Use-Cases diagram &

[2] Our team had to split up during the class since the number of groups was odd

			detailed description
Thu, Nov. 21	**Face-to-Face**	**Laith, Simon**	**Activity Diagrams discussed**
Tue, Nov. 26	**Face-to-Face**	**Laith, Simon**	**Activity Diagrams Checked**
Tue, Dec. 3	**Face-to-Face**	**Laith, Simon**	**Analysis $ Detailed Class Diagram + Object Diagram**