PRCO-204 GROUP REPORT

Adam li, Kan Lo, Travis Lai & Walter Pang

GitHub link: https://github.com/cpass19999/PRCO204HK-Project---Airline-Team-7-

PRCO204 – Group 7

内容

Introduction

In this project, we have developed an airline booking system due to the client's needs. We designed the system as a single web API which is connected to MySQL database.

This system mainly aims at providing an online platform for the user to book flights. To achieve this requirement, users are able to search the details of the flights by selecting the origin and destination.

In addition, users are able to purchase the airline ticket through the system and check out the flight information they've booked.

Due to the concern of convenience, users can also view their own information along with being able to edit their information to prevent any outdated information.

Moreover, we have defined different levels of the account in order to meet the needs of different roles for example: Administrator, senior staff and junior staff.

Staffs can help customers to search the flight information and also purchase the airline ticket for both junior & senior level. Furthermore, staffs are available to help customer for modifying their flight booking information.

However, only senior staff can be able to edit the customer's information.

In terms of Administrators, they are able to edit all of the account's information and activate the account if it's being locked. Moreover, the administrators can edit the all of the flight information.

Assumption

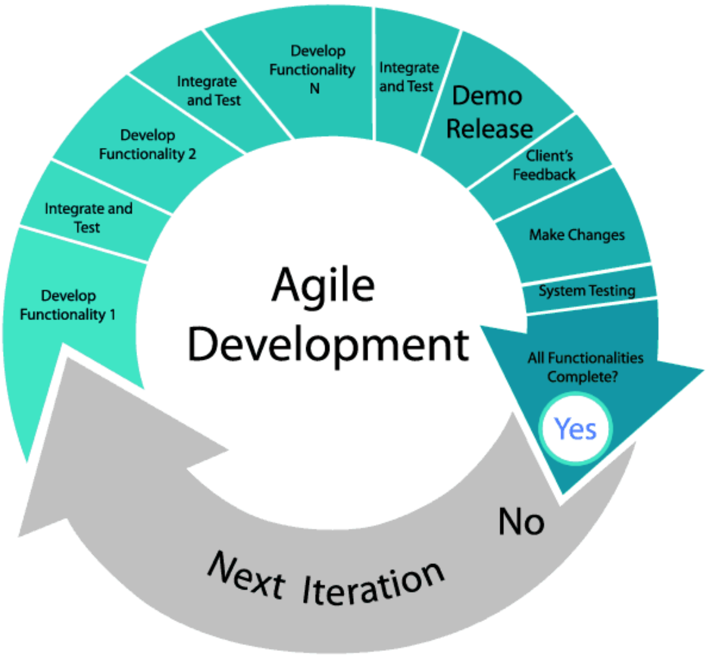
Below are the assumptions we have made during the project:

- Customers can register an account
- Customers can search the flights through the system
- Customers can book a flight through the system
- Customers can send enquiry to the flight ternary
- Customers can do the payment through the system.
- All the staff can help customers to book a flight
- Senior staff can control the flight booking request/ changes
- Senior staff can handle customer enquiry
- Administrators can edit all of the account’s information
- Administrators can edit all of the flight’s information

Project Management

Agile Development

Agile is a term used to describe approaches to software development emphasizing incremental delivery, team collaboration, continual planning and continual learning, instead of trying to deliver it all at once near the end.



Agile development methodology has been adapted during this product development this time, as we have included the below development practices for ensuring the project is under the development cycle.

- User Story Board
- Product Vision Board
- Road Map
- Risk Analysis
- Scrum Artefacts

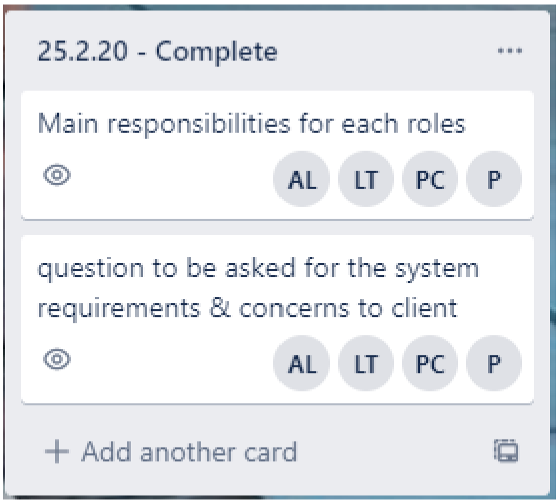
During the development of the project, we understood it was difficult to manage the progress of the project without a suitable method. To deal with problem, we have used a scrum board that helped Teams to make sprint backlog item visible. The scrum board is as important as the focal point of the project as it helps for leading the direction of the team discussion and keeps the team focused on the tasks that remain and their priorities.

By keep updating the scrum board, it is effective to review the total percentage of project completion so we can easily to review the different parts of the project such as the completed tasks, in-progress tasks and remaining tasks weekly and to adapt it to the agile development cycle – design, development, testing and deployment.

Sprint Plan and reviews

Sprint 1

Date: 25/2



25.2.20 - Complete

Main responsibilities for each roles

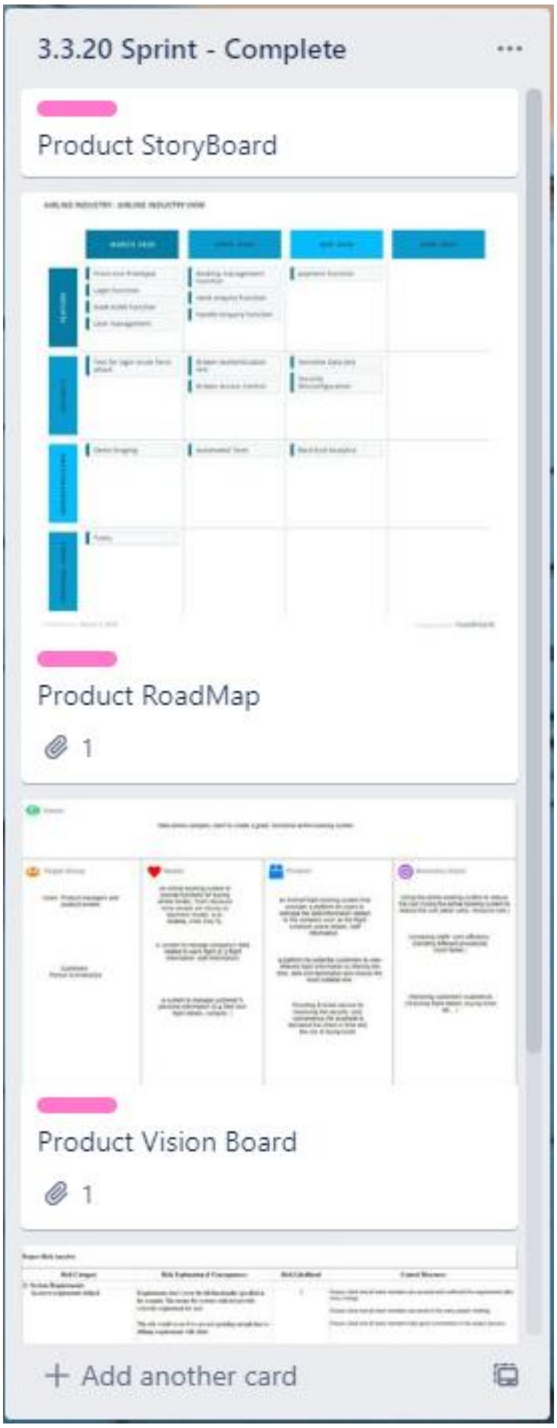
question to be asked for the system requirements & concerns to client

Plan	Start date	Expected completion date	To be completed by
Question to be asked for the system requirements & concerns to client	25/2	3/3	Everyone
Main responsibilities for each roles/person	25/2	3/3	Everyone

Review	Start date	Expected completion date	Status	Notes
N/A				

Sprint 2

Date: 3/3



3.3.20 Sprint - Complete

Product StoryBoard

Product RoadMap

1

Product Vision Board

1

+ Add another card

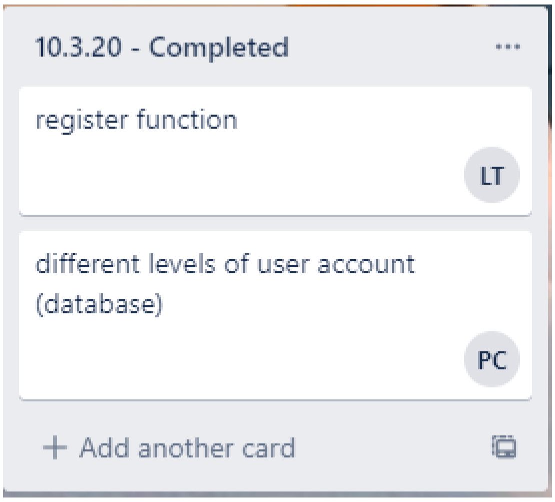
Plan	Start date	Expected completion date	To be completed by

Register function	3/3	10/3	Travis Lai
Login function (initial)	3/3	10/3	Travis Lai
Different levels of user account	3/3	10/3	Walter Pang

Review	Start date	Expected completion date	Status	Notes
Question to be asked for the system requirement & concerns to the client	25/2	3/3	Completed	
Main responsibilities for each roles/person	25/2	3/3	Completed	
Product Storyboard	25/2	3/3	Completed	Completed by Kan Lo
Product Road Map	25/2	3/3	Completed	Completed by Walter Pang
Product Vision Board	25/2	3/3	Completed	Completed by Adam Li
Risk analysis Report	25/2	3/3	Completed	Completed by Travis Lai

<u>Sprint 3</u>

Date: 10/3



10.3.20 - Completed

register function

LT

different levels of user account (database)

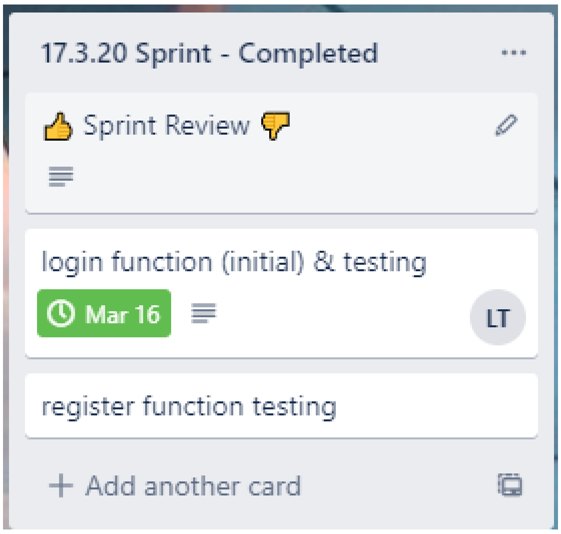
PC

Plan	Start date	Expected completion date	To be completed by
Register function Testing	10/3	17/3	Travis Lai
Login function Testing	10/3	17/3	Travis Lai

Review	Start date	Expected completion date	Status	Notes
Register function	3/3	10/3	Completed	
Login Function	3/3	10/3	In-progress	Travis was dealing with urgent personal issue so could not finish the task in time.
Different levels of user account	3/3	10/3	Completed	has been developed through the database

Sprint 4

Date: 17/3



17.3.20 Sprint - Completed

Sprint Review

login function (initial) & testing

Mar 16

LT

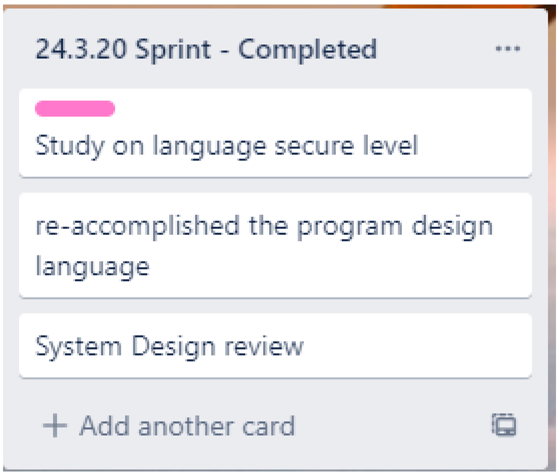
register function testing

Plan	Start date	Expected completion date	To be completed by
Review System Design	17/3	24/3	Everyone
study on system security	17/3	24/3	Everyone

Review	Start date	Expected completion date	Status	Notes
Register function testing	10/3	17/3	Completed	
Login Function	3/3	10/3	Completed	Login function has been done although it was one week later then the expected
Login function testing	10/3	17/3	Completed	

Sprint 5

Date: 24/3



24.3.20 Sprint - Completed

Study on language secure level

re-accomplished the program design language

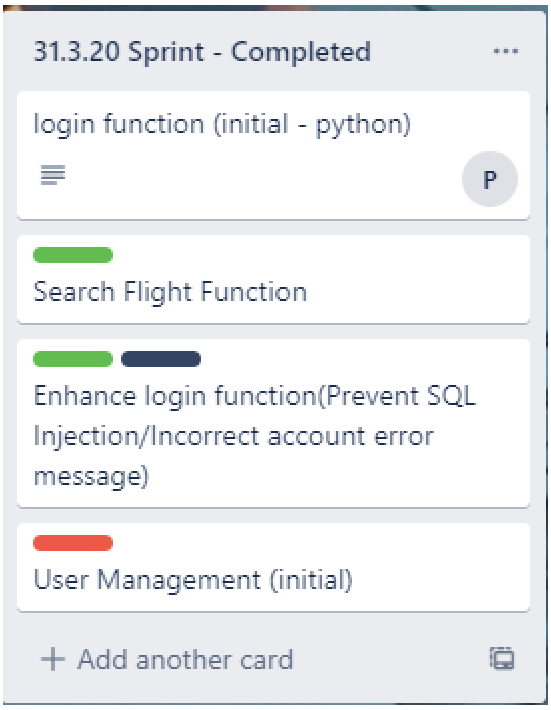
System Design review

Plan	Start date	Expected completion date	To be completed by
re-accomplish the current function of the system by using Python	24/3	31/3	Travis Lai
User account Management (initial)	24/3	31/3	Walter Pang
Search flight function	24/3	31/3	Kan Lo
Enhance login function (secure level)	24/3	31/3	Walter Pang

Review	Start date	Expected completion date	Status	Notes
Review System Design	17/3	24/3	Completed	
Study on system security	17/3	24/3	In-progress	the task was in-progress at the time there are so many reference to study however it didn’t affect the progress of the project development

Sprint 6

Date: 31/3



31.3.20 Sprint - Completed

login function (initial - python)

Search Flight Function

Enhance login function(Prevent SQL Injection/Incorrect account error message)

User Management (initial)

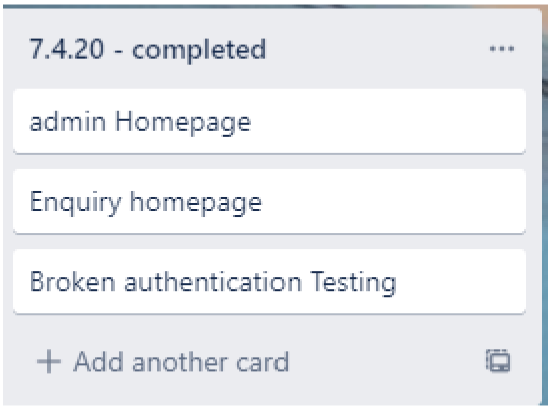
Plan	Start date	Expected completion date	To be completed by
Admin Homepage	31/3	7/4	Adam Li
Airline ticket booking function	31/3	14/4	Walter Pang
Enquiry Page	31/3	7/4	Adam Li
Broken authentication Testing	31/3	7/4	Travis Lai

Review	Start date	Expected completion date	Status	Notes
re-accomplish the current function of the system by using Python	24/3	31/3	Completed	

User account Management Initial	24/3	31/3	Completed	Tried to login the system with different level of accounts
Search flight function	24/3	31/3	In-progress	Difficulties & bugs have occurred due to develop the function with new programming language, more time was needed to come across the obstacles.
Enhance login function	24/3	31/3	Completed	Prevented SQL injection/ incorrect account information error message

Sprint 7

Date: 7/4



7.4.20 - completed

admin Homepage

Enquiry homepage

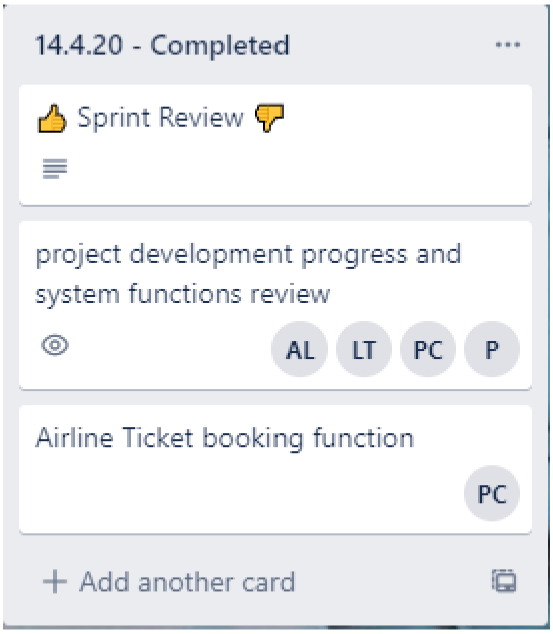
Broken authentication Testing

Plan	Start date	Expected completion date	To be completed by
Review the functions and project development progress	7/4	14/4	Everyone

Review	Start date	Expected completion date	Status	Notes
Admin Page	31/3	7/4	Completed	
Enquiry Page	31/3	7/4	Completed	
Airline Ticket booking function	31/3	14/4	In-progress	The function is in-progress and the expected date to finish is next week
Broken Authentication testing	31/3	7/4	Completed	Prevented SQL injection/ incorrect account information error message

Sprint 8

Date: 14/4



14.4.20 - Completed

👍 Sprint Review 👎

project development progress and system functions review

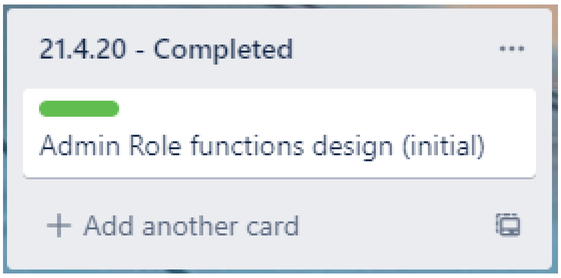
Airline Ticket booking function

Plan	Start date	Expected completion date	To be completed by
Admin role – functions design (initial)	14/4	21/4	Adam Li

Review	Start date	Expected completion date	Status	Notes
Review the functions and project development progress	7/4	14/4	Completed	
Airline ticket booking function	31/3	7/4	Completed	

Sprint 9

Date: 21/4



21.4.20 - Completed

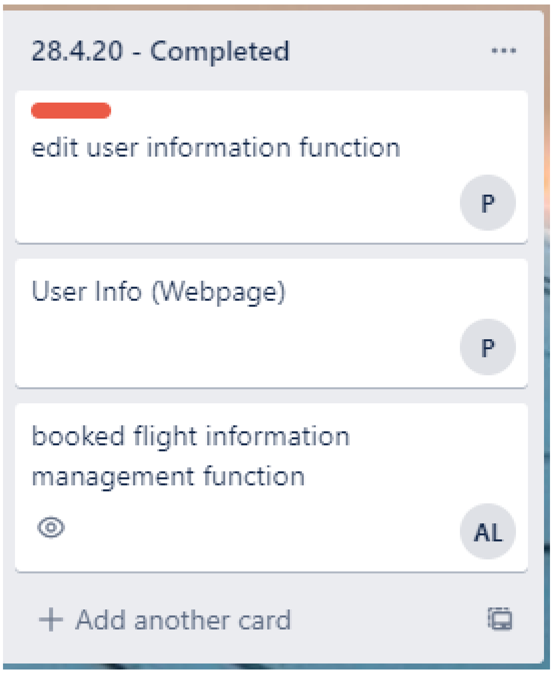
Admin Role functions design (initial)

Plan	Start date	Expected completion date	To be completed by
User information Page	21/4	28/4	Kan Lo
Edit user information function	21/4	28/4	Kan Lo
System Interface Design	21/4	5/5	Travis Lai
Booked flight information management function	21/4	28/4	Adam Li

Review	Start date	Expected completion date	Status	Notes
Admin role – functions design (initial)	14/4	21/4	Completed	

Sprint 10

Date: 28/4



28.4.20 - Completed

- edit user information function — P
- User Info (Webpage) — P
- booked flight information management function — AL

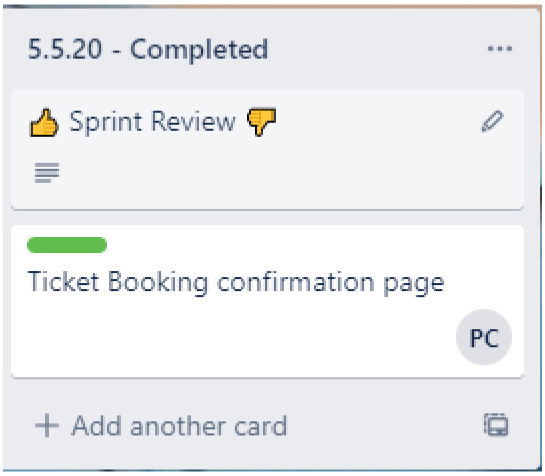
Plan	Start date	Expected completion date	To be completed by
Ticket booking confirmation Page	28/4	5/5	Walter Pang
Secure system coding	28/4	19/5	Walter Pang
Project Management Review	28/4	5/5	Everyone

Review	Start date	Expected completion date	Status	Notes
User information page	21/4	28/4	Completed	
Edit users information function	21/4	28/4	Completed	

System Interface design	21/4	5/5	In-Progress	Task is in-progress and the expected date to finish is next week

Sprint 11

Date: 5/5



5.5.20 - Completed

👍 Sprint Review 👎

Ticket Booking confirmation page

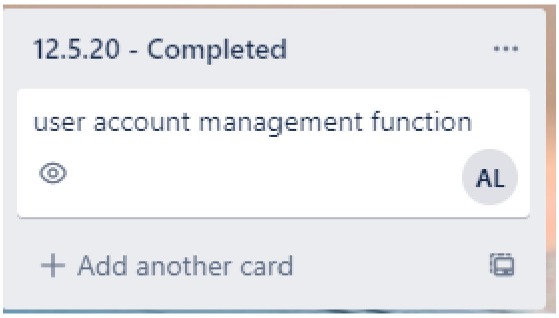
PC

Plan	Start date	Expected completion date	To be completed by
Payment function	5/5	19/5	Walter Pang
User account management function	5/5	12/5	Adam Li

Review	Start date	Expected completion date	Status	Notes
Ticket booking confirmation page	28/4	5/5	Completed	
Secure system Coding	28/4	19/5	In-Progress	Task is in-progress and the expected date to finish is two weeks later

Sprint 12

Date: 12/5



12.5.20 - Completed

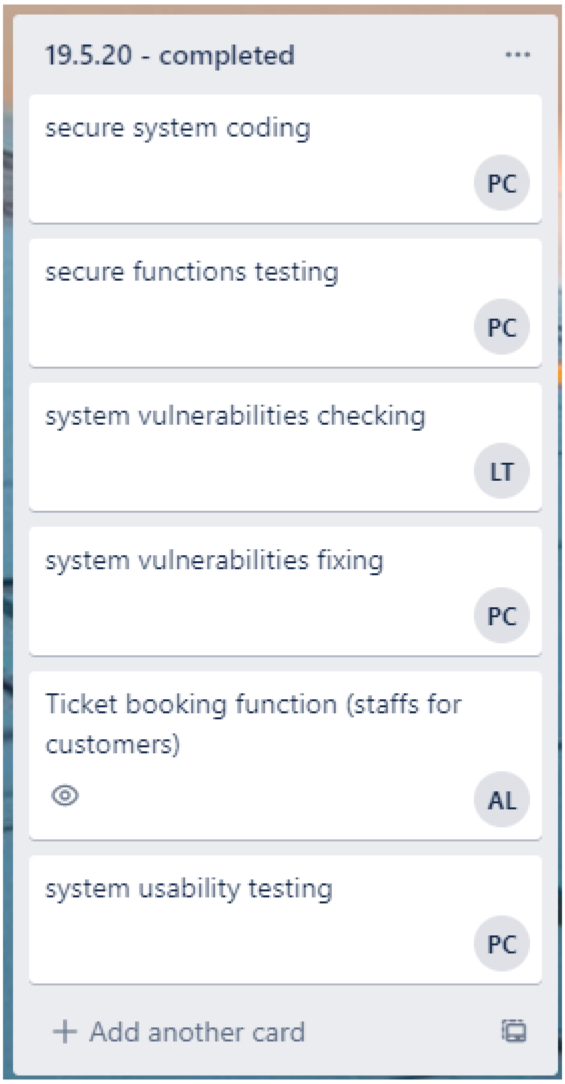
user account management function

Plan	Start date	Expected completion date	To be completed by
Ticket booking function (staffs for customers)	12/5	19/5	Adam Li
Secure functions testing	12/5	19/5	Walter Pang
Database Design Review	12/5	19/5	Walter Pang
System usability Testing	12/5	19/5	Walter Pang
System Vulnerabilities checking	12/5	19/5	Travis Lai
System vulnerabilities fixing	12/5	19/5	Walter Pang

Review	Start date	Expected completion date	Status	Notes
Secure system Coding	28/4	19/5	In-Progress	Task is in-progress and the expected date to finish is next week
User account management function	5/5	12/5	Completed	

Sprint 13

Date: 19/5



19.5.20 - completed

- secure system coding — PC
- secure functions testing — PC
- system vulnerabilities checking — LT
- system vulnerabilities fixing — PC
- Ticket booking function (staffs for customers) — AL
- system usability testing — PC

Plan	Start date	Expected completion date	To be completed by
System review (all functions & interface)	19/5	25/5	Everyone

Review	Start date	Expected completion date	Status	Notes

Secure system coding	28/4	19/5	Completed	
Secure system Coding	12/5	19/5	Completed	Task is in-progress and the expected date to finish is next week
Secure functions testing	12/5	19/5	Completed	
System vulnerabilities fixing	12/5	19/5	Completed	
Ticket booking function (staffs for customers)	12/5	19/5	Completed	
System usability testing	12/5	19/5	Completed	

Overall approach

An overall approach - used to describe the planning, methodology and tools we have adapted to bring out the deliverables.

It is important to have a good project management methodology to handle the project development in order to make sure the progress of the project can be visible so it will be easier to review and manage.

We decided to adapt APM (agile project methodology) to break down the whole project into a small pieces, so we can review each part of the project weekly to discuss which part of the project need to be improved or modified. And also it was more efficient to use our resources, greater flexibility and adaptability to change needs. As we were able to go through the design, development, testing and deployment of the project every single week.

We believe that communication is the key of a team, the better communication the team have, the more effectiveness and efficiency it brings for the project development and management.

As it is a team project so it's undeniable that without a good communication, it is impossible to deliver the ideas to each other effectively which leads to failure of getting consensus.

So we decided to have a simple conversation before the first meeting to discuss about the main responsibilities for each role and the ideas or suggestions for the project design. It was a great start for the project management as we could be able to know the strength and weakness for each one and it was helpful for tasks distribution and role allocation.

It was inevitable to have difficulties during the project development, and we came across the obstacle by having a good communication. We understood it is a team project so it is impossible to define the responsibilities so clear. We helped each other to deal with the problem we met during the project development, not only to ensure the project will be completed in time but also to improve our problem solving skills.

However, as we didn’t have any face to face communication during the time of the project development so it was important for everyone to be responsible and reliable. It took self-discipline to ensure the works is done in time with quality.

Difficulties Encountered

As this is the first time for four of us to develop a project with mainly concerned with the system security. It is inevitable that we encountered some difficulties during the project.

The first difficulty was we had no idea how to evaluate a system with security level concerns. So we did take a lot of time to study on how to make a secure system from different aspects. One example was we did change the programming language of the system once as we thought that one of the advantages to use the Python is it can be more flexible and easier to secure the system.

However as the more time we spent on the project development, the more we learnt how to secure the system from different researches and it helped us to develop basic secure functions on the system at least.

The second difficulty was to re-study on the new programming language so the development speed of the project has been decreased for a while since we changed the programming language.

The problem has been resolved by having communication more frequently, so we could gathered all of the knowledge together in order to learn more about the language and solve the problem together.

The third difficulty was lack of experience with project management such as we did change the goals and the plans of the project at the very beginning as we don't have an effective way to let them be visible to manage.

However, by using the project management tool like Trello scrum Board tool.

It was helpful to let the team be able to manage the project more effectively as we were able to manage the tasks by having a schedule, things-to-do board and weekly completed tasks boards.

Review of risk analysis

Individually rank each risk identified according to its frequency/likelihood and its severity.

Step 1 – Rank the frequency/likelihood of a given risk from 1 to 4 using these criteria:

4 = Very Likely - Almost certain to occur over the life of the project (or a 10 year period - whichever is shorter)

3 = Likely - Probably will occur during a 10-year period

2 = Unlikely - Probably will **NOT** occur during a 10-year period

1 = Very Unlikely - Almost certain **NOT** to occur during a 10-year period

Step 2 – Rank the severity of a given risk from 1 to 4 using these criteria:

4 = Very High – Would prevent goals and objectives from being achieved

3 = High – Would cause significant problems or delays in objectives being achieved

2 = Medium – Would cause relatively minor problems or delays in objectives being achieved

1 = Low – Would probably not affect project implementation

Step 3 – Compute the average frequency/likelihood and average severity ranking for each risk category.

The risk assessment tool will calculate and display these averages automatically.

Step 4 – Add the average frequency/likelihood and severity ranking for each risk category.

The risk assessment tool will calculate and display these sums automatically.

6-8 High Risk (red) – You should have a detailed mitigation action and perhaps consider modifying your goals and objectives

4-5 Medium Risk (yellow) – You should have a clearly defined mitigation action

1-3 Low Risk (green) – No mitigation action required (or a very basic action if you think it is necessary)

The risk assessment tool will determine and display whether each risk is high, medium or low automatically.

High risks categories will automatically turn cells red, medium risks will automatically turn yellow, and low risks will automatically turn green.

Risk analysis – System Requirements					
Risk Category	**Risk Explanation & Consequences**	**Risk likelihood**	**Risk severity**	**Risk Score**	**Control Measures**
Incorrect requirements defined	Requirements don't cover the full functionality specified in the scenario.This means the system could not provide correctly requirement for user. This risk would occur if we are not spending enough time to defining requirements with client. This risk can affect	3	2	5	Ensure client and all team members are received and confirmed the requirements after every change. Ensure client and all team members are joined in the every project meeting. Ensure client and all team members have good

	the timing of the design process and the functionalities included in the final system.			🟨	communtion in the project process.
Requirements continually changing	Requirements may change frequently according to the client or unpredicted things. This risk would occur if we are not spending enough time on requirements analysis and not communicating enough with the client. This can affect the development of the system as constant changes can affect the timing of the project and may cause failures in the end product.	4	3	7	Requirement changes should be accepted as a fact of the software development process. Ensure client and all team members should be reviewed and confirmed each changed requirements. No spending time to establish functions which are not user needs. Ensure the project process is on scheduled.

Developing the wrong user interface	The user interface is not match the user needs or is difficult to use. The risk will occur if we are not spending enough time to communicate with client. This can affect how the user feel the final project if the design of the interface is not suitable for the client.	3	4	7	Using the paper and graphical prototypes to show the interface before the system created. Ensure the client as much as possible during the design process.
Risk analysis – Team					

Team members do not have any pervious project experience	We need spending more time to learn how can work as a team, this could affect us could not provide enough time with the project. The risk could affect the completion of the final report.	2	3	5	Ensure all members have understands the roles and responsibilities it entails.
Unfamiliar development environment	The development team is unfamiliar with the technology being used. Including new software, methodology, programming languages, development tools etc…, This may be due to lack of experience with this type of work before. Each steps/tasks may need to spend more time to finish, and the completion time	3	3	6	Ensure the each task should be scheduled with more time considering the unfamiliarity Ensure team will need to spend more time for studying the new technologies which they haven’t used before. Communication and learn from each other is needed in order to improve the speed to adapt to the

	become unpredictable.			🟥	new development environment

Analysis & Design

Customer user stories

- Customers can register an account
- Customers can search the flights through the system
- Customers can book a flight through the system
- Customers can send enquiry to the flight ternary
- Customers can do the payment through the system.

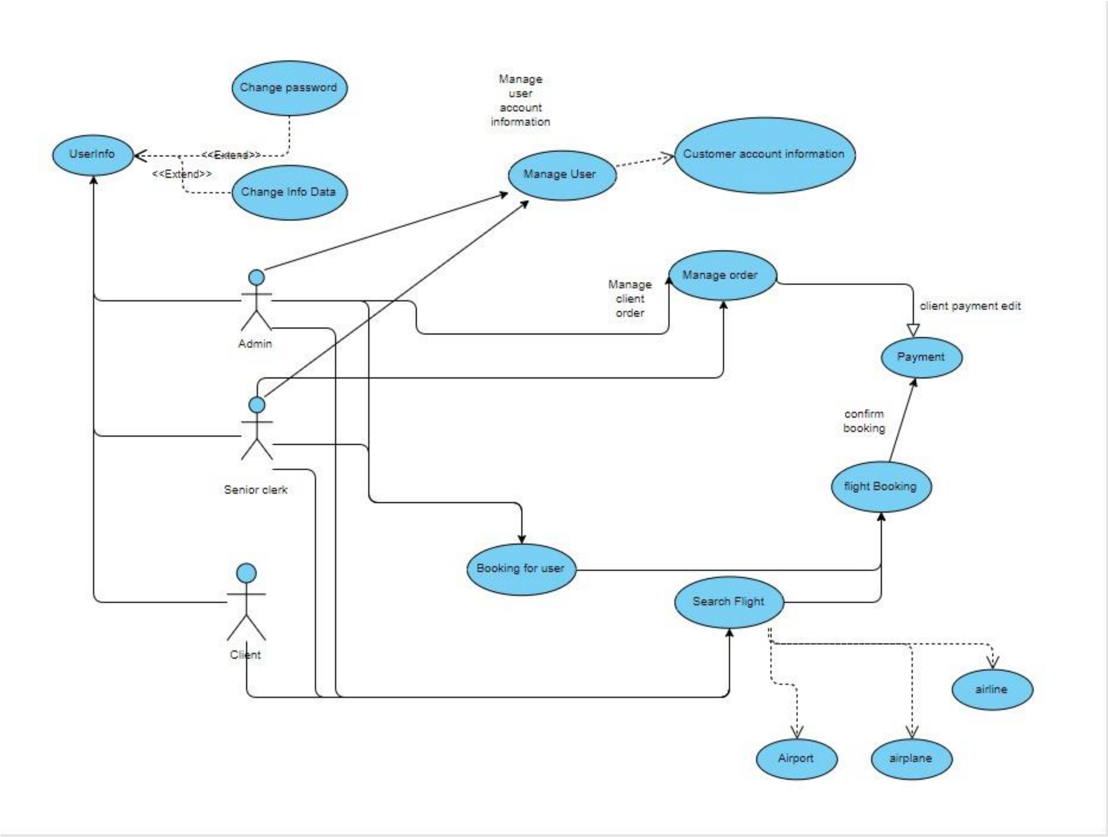
Staff user stories

- All the staff can help customers to book a flight
- Senior staff can control the flight booking request/ changes
- Senior staff can handle customer enquiry

Admin user stories

- Administrators can view &edit all of the account's information
- Administrators can view &edit all of the flight's information
- Administrators can view &edit the booking information.

Use-case Diagram



Database Design

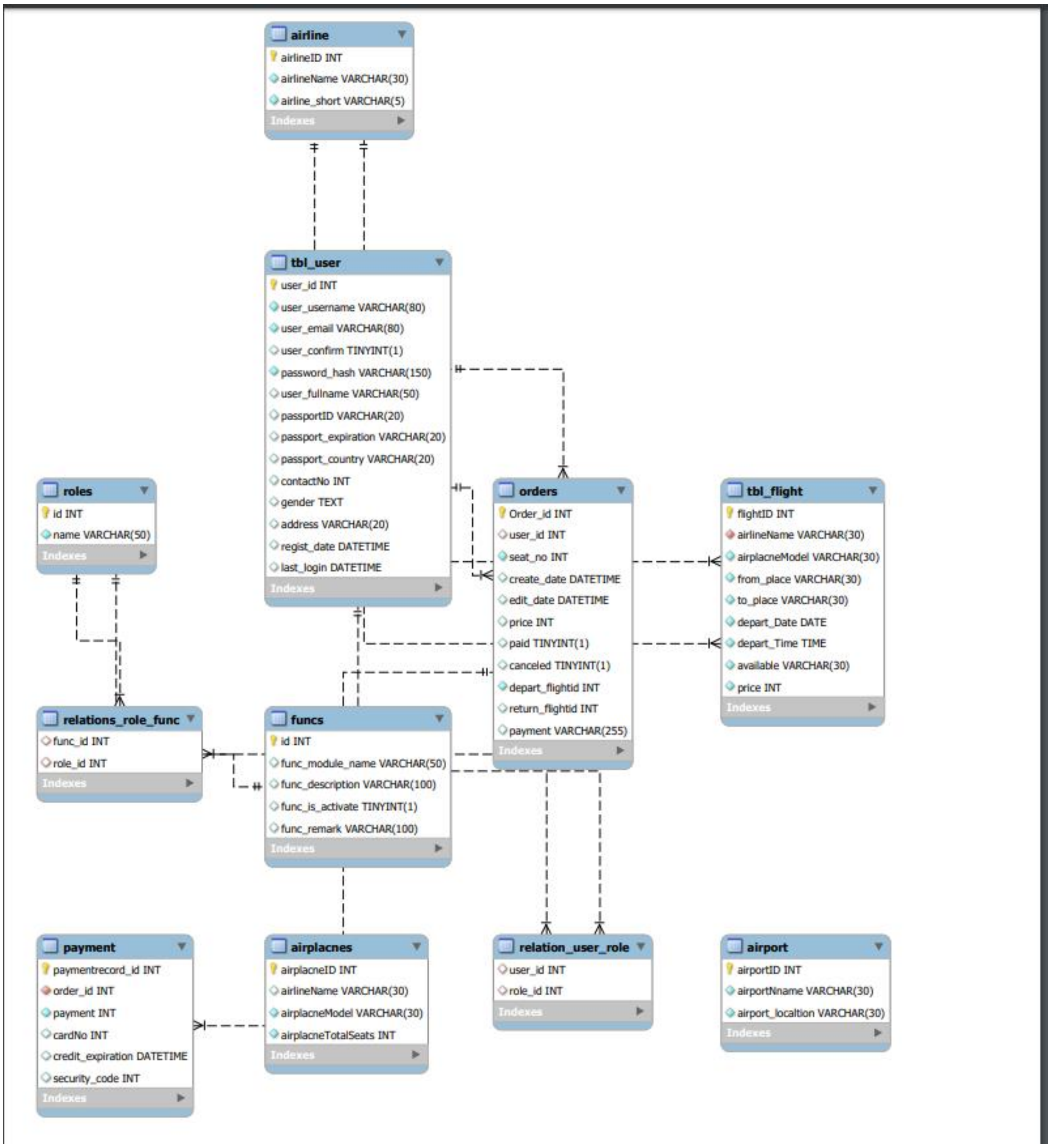
Normalization

UNF	1NF	2NF	3NF	Optimised	Entities
user_id	**user_id**	**user_id**	**user_id**	**user_id**	**User**
User_name	User_name	User_name	User_name	User_name	
user_email	user_email	user_email	user_email	user_email	
passportID	passportID	passportID	passportID *	passportID	
passport_country	passport_country	passport_country	passport_country	passport_country	
user_fullname	user_fullname	user_fullname	user_fullname	user_fullname	
contactNo	contactNo	contactNo	gender	gender	
address	address	address			
regist_date	regist_date	regist_date	last_login	last_login	
AirlineName	AirlineName	AirlineName	**Airline_short**	**Airline_short**	**Airline**
Airline_id	**Airline_id**	**Airline_id**			
airlineName	airlineName	airlineName	flightID	flightID	**flight**
airplacneModel	airplacneModel	airplacneModel	b_port_name	boat_name	
boat_id	boat_id	boat_id	from_place	from_place	
depart_Date	depart_Date	depart_Date	to_place	to_place	
depart_Time	depart_Time	depart_Time			
Order_id	Order_id	Order_id	Order_id	Order_id	**Orders**

user_id e	user_id	user_id			
seat_no	seat_no	seat_no	canceled	payment	
create_date	create_date	create_date	create_date	create_date	
price			price	price	
paid	paid	depart_flightid	depart_flightid	depart_flightid	
		return_flightid	return_flightid	return_flightid	
airportID	airportID	airportID	airportID	airportID	airport
airportNname					
airport_localtion					
		user_id			relation_user_role
		role_id			
airplacneID	airplacneID	airplacneID	airplacneID	airplacneID	**airplacnes**
airplacneModel	airlineName				
airplacneTotalSeats					
id	id	id	id	id	**funcs**
		func_description	func_module_name	func_remark	
			func_is_activate		

id	id	id	id	id	roles
		name			
			func_id		relations_role_func
			role_id		
paymentrecord_id	paymentrecord_id	paymentrecord_id	paymentrecord_id	paymentrecord_id	payment
order_id	order_id	order_id	order_id		
cardNo	cardNo	payment	credit_expiration	security_code	

Entity relationship diagram



SQL Statements

Airline

```
CREATE TABLE `airline` (
   `airlineID` int NOT NULL AUTO_INCREMENT,
   `airlineName` varchar(30) NOT NULL,
   `airline_short` varchar(5) NOT NULL,
   PRIMARY KEY (`airlineID`),
   UNIQUE KEY `airlineName` (`airlineName`),
   UNIQUE KEY `airline_short` (`airline_short`)
)
```

Airplacnes

```
CREATE TABLE `airplacnes` (
   `airplacneID` int NOT NULL AUTO_INCREMENT,
   `airlineName` varchar(30) DEFAULT NULL,
   `airplacneModel` varchar(30) NOT NULL,
   `airplacneTotalSeats` int NOT NULL,
   PRIMARY KEY (`airplacneID`),
   UNIQUE KEY `airlineName` (`airlineName`)
)
```

Airport

```
CREATE TABLE `airport` (
   `airportID` int NOT NULL AUTO_INCREMENT,
   `airportNname` varchar(30) NOT NULL,
   `airport_localtion` varchar(30) NOT NULL,
   PRIMARY KEY (`airportID`),
   UNIQUE KEY `airportNname` (`airportNname`)
)
```

funcs

```
CREATE TABLE `funcs` (
    `id` int NOT NULL AUTO_INCREMENT,
    `func_module_name` varchar(50) DEFAULT NULL,
    `func_description` varchar(100) DEFAULT NULL,
    `func_is_activate` tinyint(1) DEFAULT NULL,
    `func_remark` varchar(100) DEFAULT NULL,
    PRIMARY KEY (`id`),
    CONSTRAINT `funcs_chk_1` CHECK ((`func_is_activate` in (0,1)))
)
```

Orders

```
CREATE TABLE `orders` (
    `Order_id` int NOT NULL AUTO_INCREMENT,
    `user_id` int DEFAULT NULL,
    `seat_no` int NOT NULL,
    `create_date` datetime DEFAULT NULL,
    `edit_date` datetime DEFAULT NULL,
    `price` int DEFAULT NULL,
    `paid` tinyint(1) DEFAULT NULL,
    `canceled` tinyint(1) DEFAULT NULL,
    `depart_flightid` int NOT NULL,
    `return_flightid` int DEFAULT NULL,
    `payment` int DEFAULT NULL,
    PRIMARY KEY (`Order_id`),
    KEY `user_id` (`user_id`),
    CONSTRAINT `FK_userid` FOREIGN KEY (`user_id`) REFERENCES `tbl_user` (`user_id`),
    CONSTRAINT `orders_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `tbl_user` (`user_id`),
    CONSTRAINT `orders_chk_1` CHECK ((`paid` in (0,1))),
    CONSTRAINT `orders_chk_2` CHECK ((`canceled` in (0,1)))
)
```

Payment

```
CREATE TABLE `payment` (
    `paymentrecord_id` int NOT NULL AUTO_INCREMENT,
    `order_id` int NOT NULL,
    `payment` int NOT NULL,
    `cardNo` int DEFAULT NULL,
    `credit_expiration` datetime DEFAULT NULL,
    `security_code` int DEFAULT NULL,
    PRIMARY KEY (`paymentrecord_id`),
    KEY `FK_orderpay` (`order_id`),
    CONSTRAINT `FK_orderpay` FOREIGN KEY (`order_id`) REFERENCES `orders` (`Order_id`)
)
```

relation_user_role

```
CREATE TABLE `relation_user_role` (
    `user_id` int DEFAULT NULL,
    `role_id` int DEFAULT NULL,
    KEY `user_id` (`user_id`),
    KEY `role_id` (`role_id`),
    CONSTRAINT `relation_user_role_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `tbl_user` (`user_id`),
    CONSTRAINT `relation_user_role_ibfk_2` FOREIGN KEY (`role_id`) REFERENCES `roles` (`id`)
)
```

relations_role_func

```
CREATE TABLE `relations_role_func` (
    `func_id` int DEFAULT NULL,
    `role_id` int DEFAULT NULL,
    KEY `func_id` (`func_id`),
    KEY `role_id` (`role_id`),
    CONSTRAINT `relations_role_func_ibfk_1` FOREIGN KEY (`func_id`) REFERENCES `funcs` (`id`),
    CONSTRAINT `relations_role_func_ibfk_2` FOREIGN KEY (`role_id`) REFERENCES `roles` (`id`)
)
```

roles

```
CREATE TABLE `roles` (
    `id` int NOT NULL AUTO_INCREMENT,
    `name` varchar(50) NOT NULL,
    PRIMARY KEY (`id`),
    UNIQUE KEY `name` (`name`)
)
```

tbl_flight

```
CREATE TABLE `tbl_flight` (
    `flightID` int NOT NULL AUTO_INCREMENT,
    `airlineName` varchar(30) NOT NULL,
    `airplacneModel` varchar(30) NOT NULL,
    `from_place` varchar(30) NOT NULL,
    `to_place` varchar(30) NOT NULL,
    `depart_Date` date NOT NULL,
    `depart_Time` time NOT NULL,
    `available` varchar(30) NOT NULL,
    `price` int NOT NULL,
    PRIMARY KEY (`flightID`),
    KEY `airlineName` (`airlineName`),
    CONSTRAINT `FK_airlineName` FOREIGN KEY (`airlineName`) REFERENCES `airline` (`airlineName`),
    CONSTRAINT `tbl_flight_ibfk_1` FOREIGN KEY (`airlineName`) REFERENCES `airline` (`airlineName`)
)
```

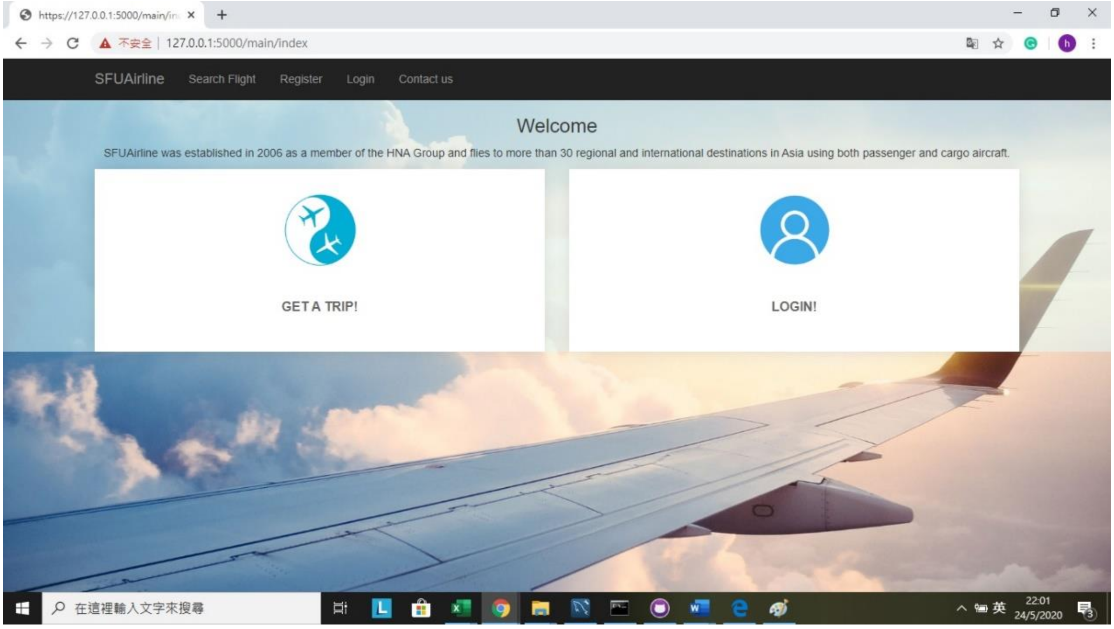
tbl_user

```
CREATE TABLE `tbl_user` (
    `user_id` int NOT NULL AUTO_INCREMENT,
    `user_username` varchar(80) NOT NULL,
    `user_email` varchar(80) NOT NULL,
    `user_confirm` tinyint(1) DEFAULT NULL,
    `password_hash` varchar(150) NOT NULL,
    `user_fullname` varchar(50) DEFAULT NULL,
    `passportID` varchar(20) DEFAULT NULL,
    `passport_expiration` datetime DEFAULT NULL,
    `passport_country` varchar(20) DEFAULT NULL,
    `contactNo` int DEFAULT NULL,
    `gender` text,
    `address` varchar(20) DEFAULT NULL,
    `regist_date` datetime DEFAULT NULL,
    `last_login` datetime DEFAULT NULL,
    PRIMARY KEY (`user_id`),
    UNIQUE KEY `user_username` (`user_username`),
    UNIQUE KEY `user_email` (`user_email`),
    CONSTRAINT `tbl_user_chk_1` CHECK ((`user_confirm` in (0,1)))
)
```

Usability

HCI

HCI is the study, planning and design of what happens when you and a computer work together. As it names implies, HCI consists of three parts: the user, the computer itself and the ways they work together.



Welcome

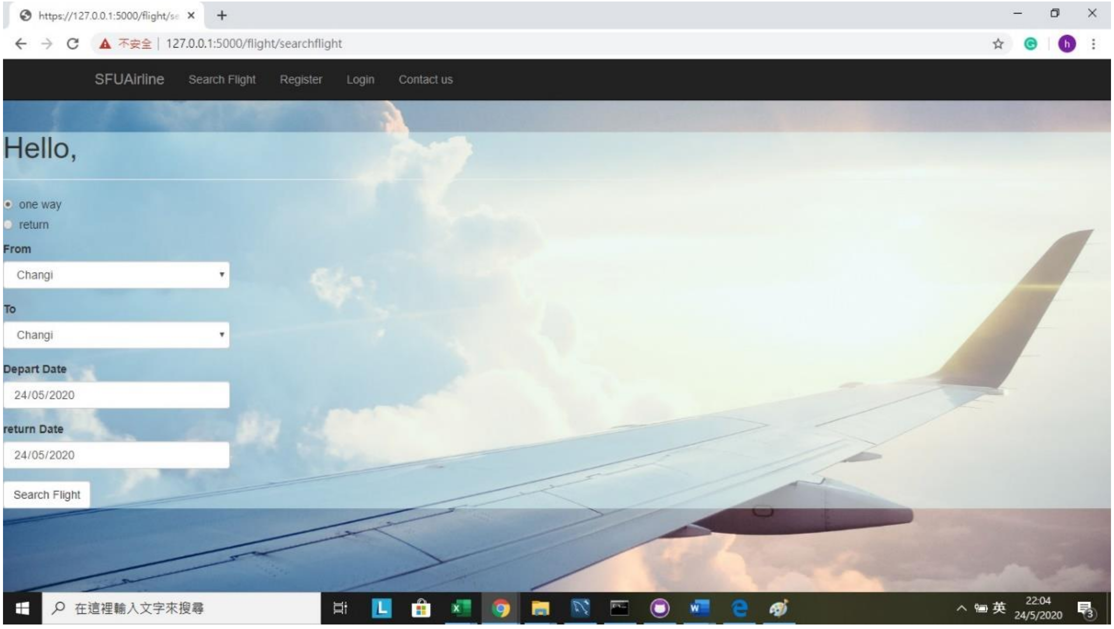
SFUAirline was established in 2006 as a member of the HNA Group and flies to more than 30 regional and international destinations in Asia using both passenger and cargo aircraft.

Clarity

One of the core of usability is clarity. We maintained the system's simplicity and readability by using a neutral color scheme and simple layout.

As this is an airline ticket booking system so we want the user to understand what is the purpose of the website once they get into the website. So the background of the system is also an airline photo to increase the recognition of the system

For the color scheme, the layouts has been designed with blue and white as the main colors, as these two colors can be associated with the cloud, airplane and the sky. These colors are not only attention-seeking, but also being representative.



Hello,

● one way
○ return

From
Changi

To
Changi

Depart Date
24/05/2020

return Date
24/05/2020

Learnability

No matter how good is the system design, it is meaningless If the users feel difficult to operate the system. So we understand it’s important to maintain the system’s learnability.

There’s no such useless buttons/process in our system. We want it to be as simple as possible to use in order to improve users experience.

For example user can search the flight easily by entering the required information in text format only.

Also the navigation bar has been design with the learnability principle, all the functions have been grouped into the navigation bar, and it provides a clear view for users to find out any of the functions they need to use in the system.

Cognitive Walkthrough

Cognitive walkthroughs are used to examine the usability of a product. They are designed to see whether or not a new user can easily carry out tasks within a given system. It is a task-specific approach to usability.

The idea is that if given a choice – most users prefer to do things to learn a product rather than to read the manual or follow a set of instructions.

To complete the cognitive walkthrough for the system, four interviewee has been invited to perform different task by using different level of the user account. During the test, same problems for the functions or system usability can be found and changes can be adapted after receiving the feedback from the testers.

Each user has been given a task sheet to understand what they need to do and evaluate, for example: user to search & book flight, staff to help customer to book flight and edit user information, admin to edit flight information and account information.

After the demonstration for the system, three testers have been given a feedback form to fill in their observation. And future improvement for the system from user level can be developed by reviewing the feedback from the testers.

In summary, the testing users gave positive feedback for the web clarity and learnability. For example to finish the tasks that have been given to them or to figure out how to use functions without any instructions. It was relied on by the simple interface and layout of the system and great design of the navigation bar. The functions are kind of “straight forward” so it was useful to guide them to finish the tasks.

However users gave feedback to suggest password requirements should be more restrictive such as there should be at least one Capital letter required for the password.

Also there were some of the typo in the system such as “register” to “regist” and “flight” to “fight”. Those mistakes would be happened a lot however they were preventable.

The cognitive walkthrough for the system was useful as it found out the components of the system which are needed to improve and prevent us to make mistakes in the system.

Usability evaluations

For the usability tests users were asked to complete a list of tasks and gave feedback to each of the behavior/requirement:

Behaviour/requirement
As a customer, I wish to book a flight ticket through the system
Good / Intuitive features
-no any distraction -easy to use
Suggested Improvements
The display of the searched flight results can be more comprehensive , for example the starting time and the departure time should be added The payment should be confirmed by the system automatically. It would be great if seat plan is provided.
Problems encountered
N/a

Behaviour/requirement
As a customer, I wish to send enquiry to the flight ternary

Good / Intuitive features
Contact details are clearly displayed on the page
Suggested Improvements
A enquiry form can be provide in the system
Problems encountered
N/a
Behaviour/requirement
As a senior staff , I wish to control the flight booking request/ changes
Good / Intuitive features
All the booked flight information clearly display on the system
Suggested Improvements
Delete function should be added for the booking management
Problems encountered
n/a

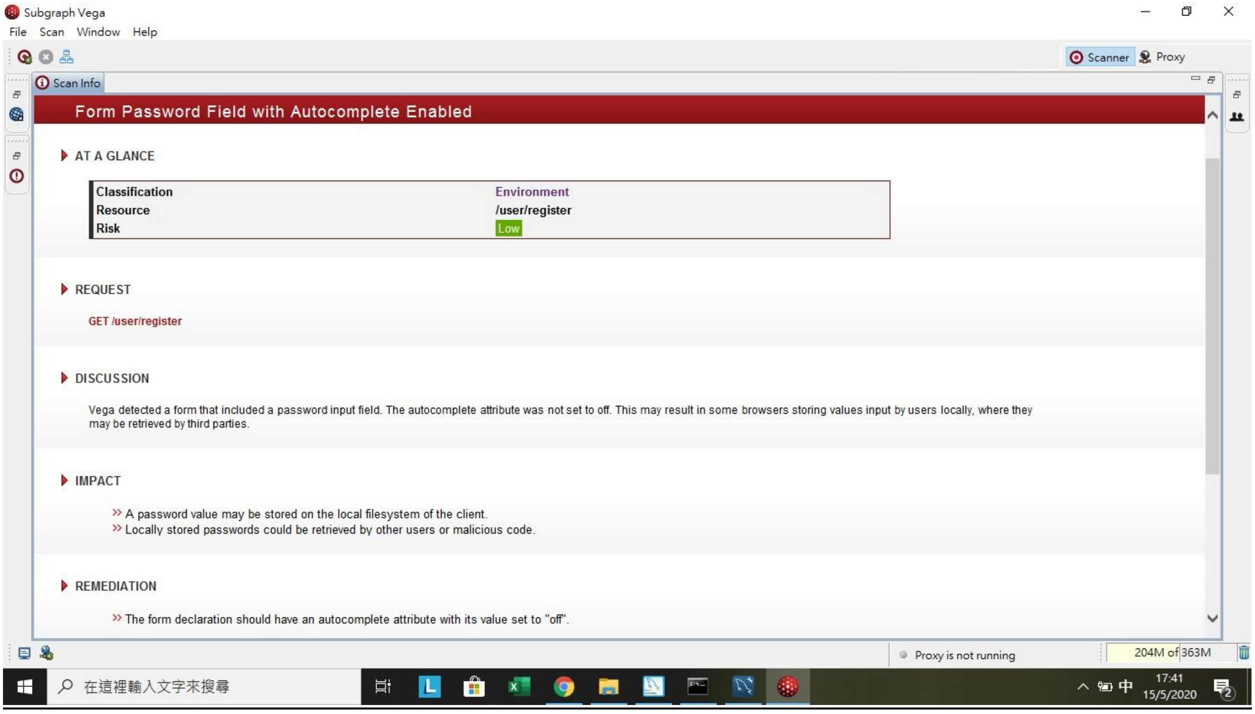
Behaviour/requirement
As a senior staff, I wish to handle customer enquiry
Good / Intuitive features
N/A
Suggested Improvements
A enquiry form can be provide in the system
Problems encountered

As there is no enquiry form to be provided in the system, staff is unable to handle the enquires through the system.

Behaviour/requirement
As a senior staff, I wish the payment can be done through the system
Good / Intuitive features
Single page has been developed to confirm the payment
Suggested Improvements
Payment details should be displayed in the confirmation page more clearly
Problems encountered
No double confirmation button caused no roll-back option

Security

Form Password Field with Autocomplete enabled



Form Password Field with Autocomplete Enabled

AT A GLANCE

Classification	Environment
Resource	/user/register
Risk	Low

REQUEST

GET /user/register

DISCUSSION

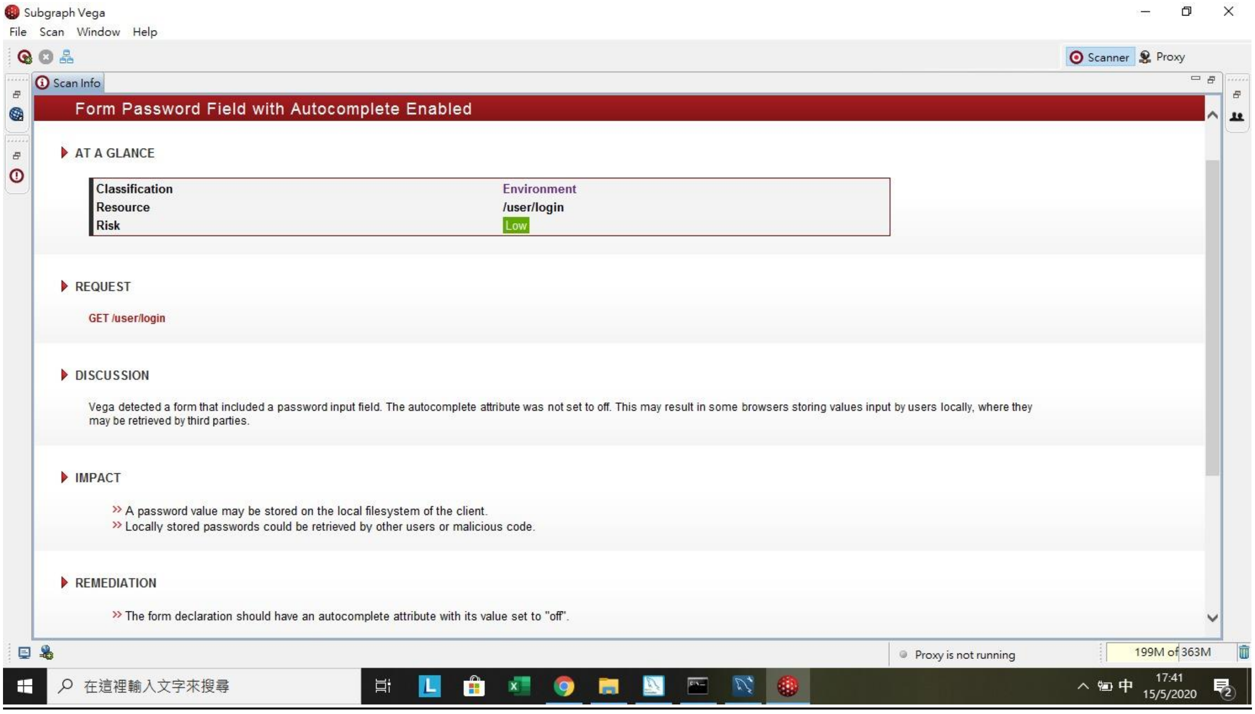
Vega detected a form that included a password input field. The autocomplete attribute was not set to off. This may result in some browsers storing values input by users locally, where they may be retrieved by third parties.

IMPACT

- A password value may be stored on the local filesystem of the client.
- Locally stored passwords could be retrieved by other users or malicious code.

REMEDIATION

- The form declaration should have an autocomplete attribute with its value set to "off".



Form Password Field with Autocomplete Enabled

AT A GLANCE

Classification	Environment
Resource	/user/login
Risk	Low

REQUEST

GET /user/login

DISCUSSION

Vega detected a form that included a password input field. The autocomplete attribute was not set to off. This may result in some browsers storing values input by users locally, where they may be retrieved by third parties.

IMPACT

- A password value may be stored on the local filesystem of the client.
- Locally stored passwords could be retrieved by other users or malicious code.

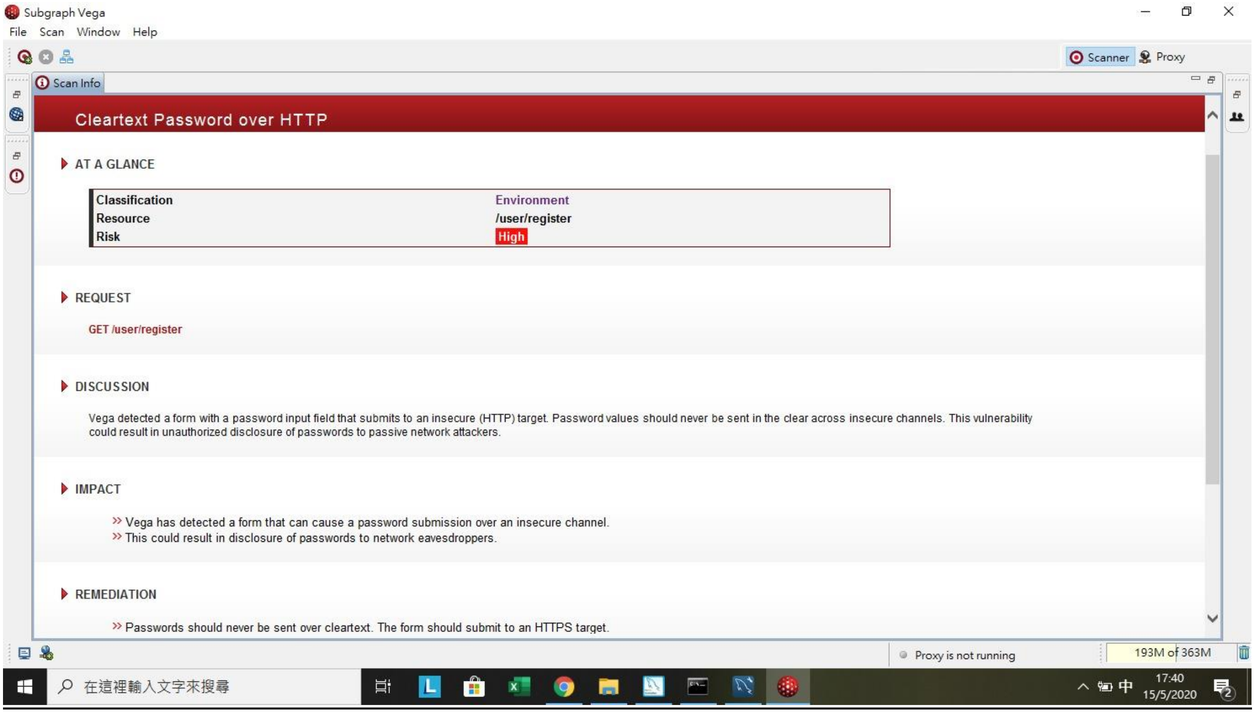
REMEDIATION

- The form declaration should have an autocomplete attribute with its value set to "off".

Solution

Add the attribute `autocomplete="off"` to the form tag or to individual "input" fields. However, since early 2014, major browsers don't respect this instruction, due to their integrated password management mechanism, and offer to users to store password internally.

Cleartext Password over HTTP



Cleartext Password over HTTP

AT A GLANCE

Classification	Environment
Resource	/user/register
Risk	High

REQUEST

GET /user/register

DISCUSSION

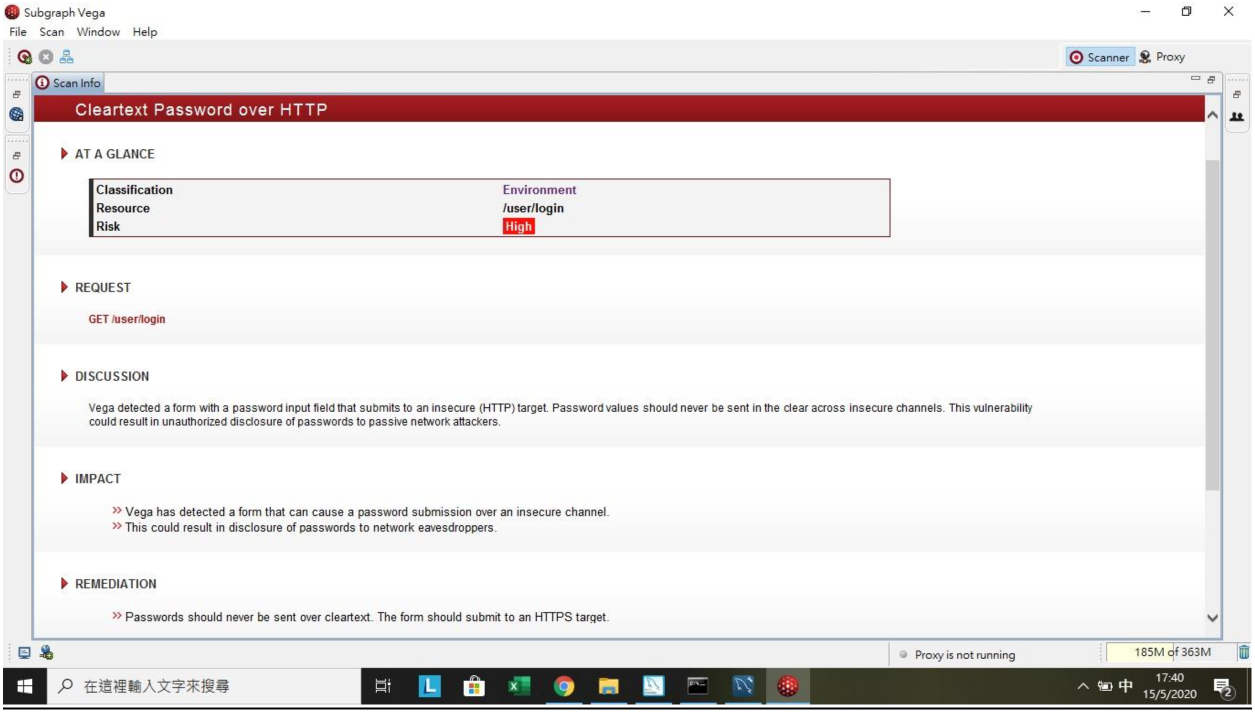
Vega detected a form with a password input field that submits to an insecure (HTTP) target. Password values should never be sent in the clear across insecure channels. This vulnerability could result in unauthorized disclosure of passwords to passive network attackers.

IMPACT

» Vega has detected a form that can cause a password submission over an insecure channel.
» This could result in disclosure of passwords to network eavesdroppers.

REMEDIATION

» Passwords should never be sent over cleartext. The form should submit to an HTTPS target.



Cleartext Password over HTTP

AT A GLANCE

Classification	Environment
Resource	/user/login
Risk	High

REQUEST

GET /user/login

DISCUSSION

Vega detected a form with a password input field that submits to an insecure (HTTP) target. Password values should never be sent in the clear across insecure channels. This vulnerability could result in unauthorized disclosure of passwords to passive network attackers.

IMPACT

» Vega has detected a form that can cause a password submission over an insecure channel.
» This could result in disclosure of passwords to network eavesdroppers.

REMEDIATION

» Passwords should never be sent over cleartext. The form should submit to an HTTPS target.

Solution

compile() method of Regex module makes a Regex object, making it possible to execute regex functions onto the *pat* variable. Then we check if the pattern defined by *pat* is followed by the input string *passwd*. If so, the search method returns *true*, which would allow the password to be valid.

SQL Injection

To prevent SQL injection, input validation is one of the most effective method to solve the security problem, by preventing symbols like “= + &” to be summited in the form, it can prevent the attackers to insert SQL query into input fields.

Moreover, ORM(Object-Relation Mapping) has been adapted to map our DB tables to our objects, allowing us to read, write and query entire objects. It reduced our use of explicit SQL so It was useful to avoid SQL injection.

Broken authentication

Broken authentication attacks aim to take over one or more accounts giving the attacker the same privileges as the attacked user. **Authentication** is “**broken**” when attackers are able to compromise passwords, keys or session tokens, user account information, and other details to assume user identities.

We have adapted certain actions to prevent broken authentication by restricting password length and password complexity, and also to prevent URL rewriting

Software engineering

The system is aimed to provide an online system for customer to purchase an airline ticket through the system. Also the staffs can be able to help their customer to purchase the airline ticket through the system as well as to manage the customer and flight information through the system.

Code reviews

The code reviews aims to ensure the system coding is easy to understand. It is important to check whether if there are comments in code and if the code is well-sorted and well-organized

One of the suggestions was to ensure the attributes or function has been defined with meaningful name, so it will be easier for understanding and further management, for example the attribute for customer id can be titled as customer_id instead of “cusid” etc….

Moreover, it was also suggested to add more comments into complicated/large functions to explain each of the steps the functions are handling so it can be more effective for further management once the modification is needed. On the contrary, small/easy functions should be attached with less comments.

After revision of the coding, the program now is more well-organized and understandable, attributes with meaningless name have been modified as many as possible.

Legal, Social, Ethical and professional issue

Legal

The project development software testing software could potentially obtain malicious software or activity which may steal the sensitive data illegally.

Social and ethical

Sensitive data or information that is retrieved could be completely legitimate and deemed as confidentially or restricted which would be offending their privacy rights. This project follow the social and ethical policy and we will only use the data for project development use without any improper actions.

Professional

The project would need an administrator to handle professional issue for example to keep the high quality & efficiency during the project development and keep analyzing the whole system to find out the possible risks. The action should be continuous to prevent loss for the stakeholders and bad affect for the society.

Shortcoming and further improvement

Upon reflection of the system design, we are satisfied how we manage the project and the speed we dealt with the difficulties, however there are also some areas of the project which will need to be improved.

One of the improvement suggestion is to research more before designing the system, as we did change our programming language during the project development and it affected our estimated time to complete the project, the speed of project development has been affected for a while caused of that as we will need to spend more time to re-develop the functions we' made before.

It can be prevented if we did more research before we designed how to develop the system. Getting more understanding with each of the programming language means we could be able to develop the project in a most suitable environment and the project can also be more comprehensive

Another improvement we could have made is to use more patterns to develop the project as it allows to create a sequence of commands by providing a queue system and extensions to add new command is easy and can be done without changing the existing code.

Furthermore, we think the interface design is not comprehensive and it can be improved in order to make it more user-friendly.

Overall, we think this project is as a good experience for us to learn more about the python language and project management skills.

References

1. Pekka Abrahamsson, Outi Salo, Jussi Ronkainen and Juhani Warsta(2002).Agile Software Development Methods: Review and Analysis [online]

https://arxiv.org/ftp/arxiv/papers/1709/1709.08439.pdf

2. Andrew Littlefield(2016). The Beginner's Guide to scrum and agile project management

   https://blog.trello.com/beginners-guide-scrum-and-agile-project-management

3. Guido Van Rossum and the Python development team (2018). Python Tutorial Release 3.7.0

   https://bugs.python.org/file47781/Tutorial_EDIT.pdf

4. Justin Clarke(2009). SQL Injection Attacks and Defense

   http://index-of.es/EBooks/SQL%20Injection%20Attacks%20and%20Defense.pdf

5. Ibad Ur Rehman(2018) What is A brute force attack

   https://www.cloudways.com/blog/what-is-brute-force-attack/

6. Acunetix. Cross-site Scripting (XSS)

   https://www.acunetix.com/websitesecurity/cross-site-scripting/