**Hong Kong Institute of Vocational Education (Tuen Mun)**

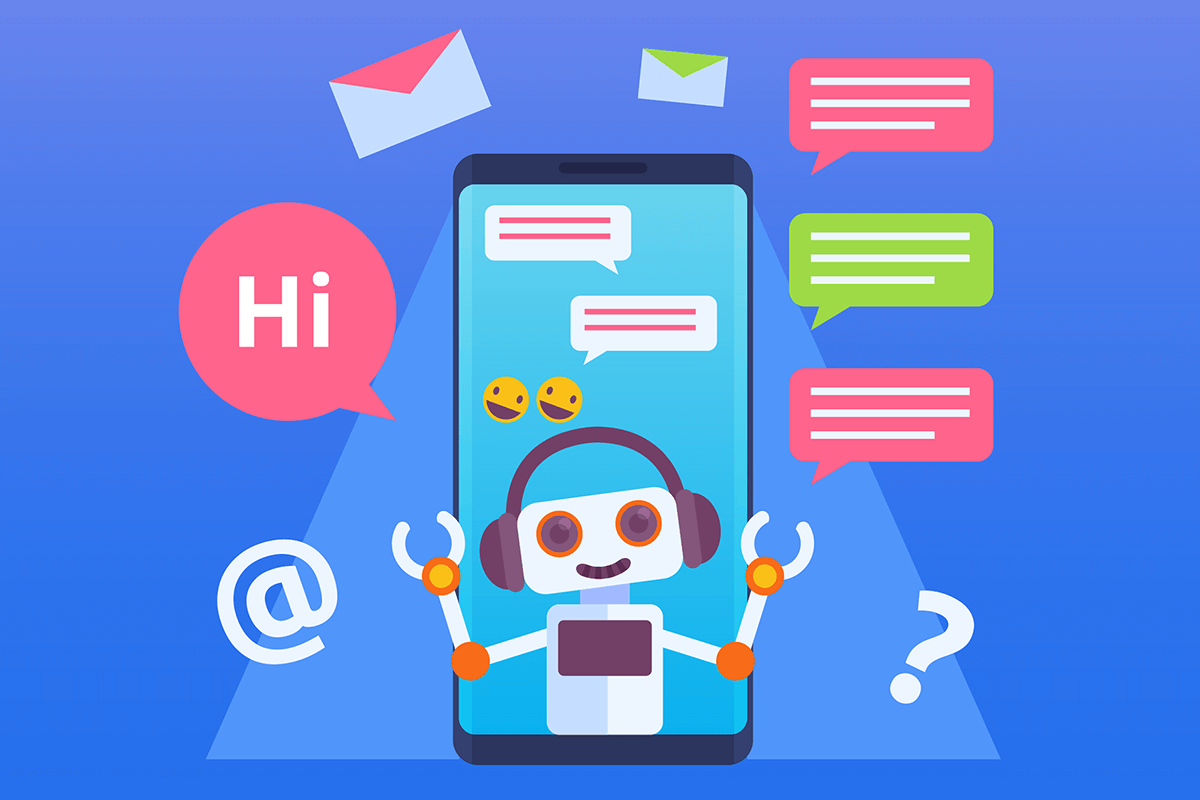
**Department of Information Technology**

**Higher Diploma in Software Engineering**

**Final Year Project (ITP4913M)**

**Final Report**

**(2020/2021)**



Real-time Chatting System for Advising Computer product

|  |  |
| --- | --- |
| Supervisor: | Mr. Yau Chak Yuen |
| Co-supervisor: | Mr. Lam Chun Fu |

We declare that this is a group project and that no part of this submission has been copied from any other student’s work or from any other source except where due acknowledgement is made explicitly in the text, nor has any part been written for us by another person.

|  |  |  |
| --- | --- | --- |
| **Student** | **Contribution to the project (%)**  **(Total 100%)** | **Signature** |
| Cheung King Hung | 33% |  |
| Lo Yee Kit | 33% |  |
| Ng Ka Ho | 33% |  |

**Abstract**

Our project is a set of the system include e-commerce, chatbot, and chat app. Users can use the chatbot or chat app to get professional advice during they pick out and buy on the e-commerce website. This system could reduce the burden on staff, and providing better service for customers, customers can get advice on computer products at any time. Advice on computer products could get from the chatbot AI that had high accuracy and precision. Or customers could get advice from experienced staff.

This report will describe the details of the e-commerce system. Firstly, the project's background will be introduced. Besides, problem analysis and suggested solutions will be described. In requirement identification, this part will design the architecture of the proposed system and scoping the proposed system. Also, this part will describe the functions which the system will provide, and the data processed by the system. Besides, the entity-relationship diagram and data dictionary will be proposed in part of the data model.

Then, this report will describe the detailed design which include the data dictionary, architectural of the products and use interface design.

Following the above, the implementation part will list the testing on the system, including the test case and results. Results and conclusions will be identified for conclude procedure of the whole project, including a summary of the results, conclusions, problems encountered, limitations of the system, and future extensions on the system.

Table of Content

[Introduction 1](#_Toc69693709)

[Problem 3](#_Toc69693710)

[1. Customers confuse with a ton of similar product. 4](#_Toc69693711)

[2. Customers cannot evaluate what level of products they need 6](#_Toc69693712)

[3. Customers would assembly with incompatible component 7](#_Toc69693713)

[4. Computer e-store with incomprehensive function for customers 8](#_Toc69693714)

[5. Service staff of computer e-store is always busy and not helpful 9](#_Toc69693715)

[Solution 11](#_Toc69693716)

[1 A helpful system specially designed for customers who purchase computer components 11](#_Toc69693717)

[Advantages 11](#_Toc69693718)

[2. A chatbot system for aids 13](#_Toc69693719)

[Advantages 13](#_Toc69693720)

[Problems of this solution and response 15](#_Toc69693721)

[1. Comprehensive service response function even on mobile for staff 16](#_Toc69693722)

[Advantages 16](#_Toc69693723)

[System Requirements 17](#_Toc69693724)

[System Architecture: 17](#_Toc69693725)

[Backend side: 18](#_Toc69693726)

[**Web Server:** 18](#_Toc69693727)

[**Database** 20](#_Toc69693728)

[**Chatbot** 22](#_Toc69693729)

[**Live chat system**: 23](#_Toc69693730)

[Frontend side: 24](#_Toc69693731)

[**Store webpage:** 24](#_Toc69693732)

[**Chat apps:** 25](#_Toc69693733)

[Suggest Specification 26](#_Toc69693734)

[Non-Functional Requirement 27](#_Toc69693735)

[Product requirement 27](#_Toc69693736)

[Security 27](#_Toc69693737)

[Organizational requirement 28](#_Toc69693738)

[Staff requirement 28](#_Toc69693739)

[Manual of the system 28](#_Toc69693740)

[Quick support 29](#_Toc69693741)

[System maintenance 29](#_Toc69693742)

[5.3 External requirement 29](#_Toc69693743)

[Database connection 29](#_Toc69693744)

[Scope of the proposed system 30](#_Toc69693745)

[Software develop process model 31](#_Toc69693746)

[**A simple structure of Throwaway Prototype Model** 31](#_Toc69693747)

[**Planning:** 32](#_Toc69693748)

[**Design:** 32](#_Toc69693749)

[**Prototyping:** 33](#_Toc69693750)

[**Implementation:** 33](#_Toc69693751)

[**Verification:** 34](#_Toc69693752)

[Function 35](#_Toc69693753)

[E-shop Website 35](#_Toc69693754)

[Chat Apps 41](#_Toc69693755)

[Login 41](#_Toc69693756)

[Send Message 42](#_Toc69693757)

[Problem Analysis 43](#_Toc69693758)

[Use case 43](#_Toc69693759)

[Use Case Diagrams 49](#_Toc69693760)

[Class Diagram 51](#_Toc69693761)

[State Machine Diagram 52](#_Toc69693762)

[Sequence Diagrams 54](#_Toc69693763)

[Detailed Design 62](#_Toc69693764)

[Json Schema 62](#_Toc69693765)

[Data Dictionary 63](#_Toc69693766)

[User Interface Design and User Guide 66](#_Toc69693767)

[Chat Apps 66](#_Toc69693768)

[Project Plan 79](#_Toc69693769)

[Implementation 80](#_Toc69693770)

[Tase Case 80](#_Toc69693771)

[Login Test 82](#_Toc69693772)

[Send Message Test 85](#_Toc69693773)

[Phone Call Test 89](#_Toc69693774)

[Send Photo Test 91](#_Toc69693775)

[Results and Conclusions 117](#_Toc69693776)

[Summary of the Results 117](#_Toc69693777)

[Conclusion 118](#_Toc69693778)

[Problems Encounter 119](#_Toc69693779)

[Technical Difficulties 119](#_Toc69693780)

[Limitations of the system 121](#_Toc69693781)

[Future Extension 121](#_Toc69693782)

[Reference 123](#_Toc69693783)

[Appendices 124](#_Toc69693784)

[Logbook 124](#_Toc69693785)

# Introduction

**Background**

With refer to Wikipedia1, Chatbots are increasingly present in businesses. With customer service taking place via messaging apps as well as phone calls, there are growing numbers of use-cases where chatbot deployment gives organizations a clear return on investment.

Chat-bots are estimated to save businesses 8 billion dollars by 2022. With 24-hour automated functionality and the ability to give you hours back in your life, we have chat-bots on the way up. We still have real chat (real-time chatting) on a one-to-one basis, an outdated way to engage with customers.

**Chat Bot / Real time chatting**

However, although chatbots not only can work for 24-hours but also accuracy is greater than humans. Real chatting or phone chatting remains the most used (48%) and preferred (80%) method for reaching support agents.

Besides, customers can get a response from real-time chat with a fast speed. An advisor who has a lot of experience able to give customers advice on the product faster and accurately. Also, it would be simplicity than a chatbot for a customer, an advisor can provide a suggestion that easier to understand. However, the most crucial reason is empathy, the consultant can give suggestions from the view of the customer.

Thus, this project needs to design and implement an e-commerce website that supports real-time chatting between the customer and computer adviser, purpose solutions the problems from different customers.

The document will point out the importance of real-time chatting and the reason it is irreplaceable. Then the project will propose solutions and conduct a requirement analysis.

After the initial analysis of the problem of e-shopping. Customers will be facing the below problems in the e-shop:

1. Cannot get help when encountering problems since service staff was busy.

2. Cannot choose the right product.

**E-commerce System Trend**

However, an e-common shop with chat-bot and real-time chatting could help the customers to solve the above problems. This project will be committed to solving the above problems. After the analysis of the problem, the objective of this project is to create a chat-bot that could be applied in an e-common shop and a mobile app that customers could chat with the staff. Those applications could advise products for customers through AI or staff.

Currently, the project needs to implement the system and build the bed within the solutions which already been analyzed. This report will describe the detail of the objective system include the use case, project plan, system architecture, and UML diagram etc.

**Purpose of this project**

This project is to develop a system that to let customers buy computer component easier. This project include phone app and website with chat UI.

# Problem

E-shop is a new style nowadays considering that it is most time cheaper than real shop because the process and price of running an e-shop are lower.

According to ecomdash, “Any business that isn’t moving toward an omnichannel retailing strategy will likely be left behind by its online savvy competitors.”[[1]](#footnote-2) The types of e-shop are lots, but one of the most special types of an e-shop is a computer store which provides different types of components for customers.

As a customer who does not have enough knowledge to build a computer by self-assembling[[2]](#footnote-3), they would not be at ease with e-shop because e-shop cannot provide staff or advisor which is very important for them. Most computer e-shop cannot continue to run because of low sales rate. Most remaining e-shops are the sub-store of many brick-and-mortar stores.

Intern of why customers do not take computer e-shop as an option, the problems are as follows:

## Customers confuse with a ton of similar product.

Even if the customers have a lot of IT and self-assembled computer knowledge, assembling a computer is a completely new area for those who do not keep updating those kinds of knowledge. The products in the computer market are incredibly big for most people. If they are unable to make a decision on their own, they may need more information from the website rather than searching by themself.

Take display cards (One of the most important parts of a computer especially for gamers) as an example, the models of new generation display cards are a lot. Display cards include different levels, brands, and architecture.

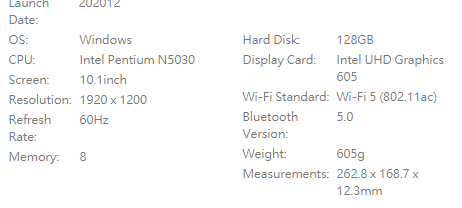
A list of display cards which is using the same model of RTX3070[[3]](#footnote-4):

Customers should look for the component for the computer but there are necessary components and unnecessary components. If the necessary component miss, the computer will not work or even be damaged. Also, there is no use to buy something they do not need.

The list of computer component[[4]](#footnote-5):



Even for the customer who looking for a brand PC which means it is assembled. Customers face the same problems as buying a laptop. The specification details are incomprehensible:

Complex specification for customers to compare with each other:

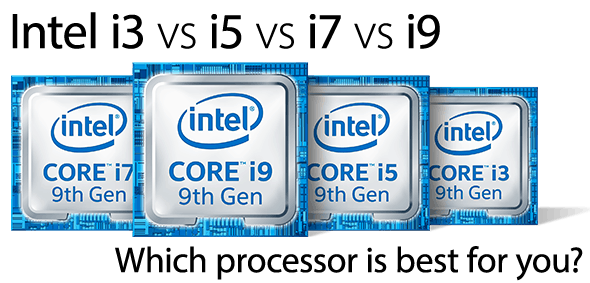
## Customers cannot evaluate what level of products they need

As mentioned, the products are a lot, and it is difficult to evaluate a budge of the assembling computer. Some situations do not make sense for most people but true: A computer with high computing performance sometimes cheaper than which not.

Customers always need advice from staff, but it is hard to achieve in an e-shop.

A computer list base on many requirements:

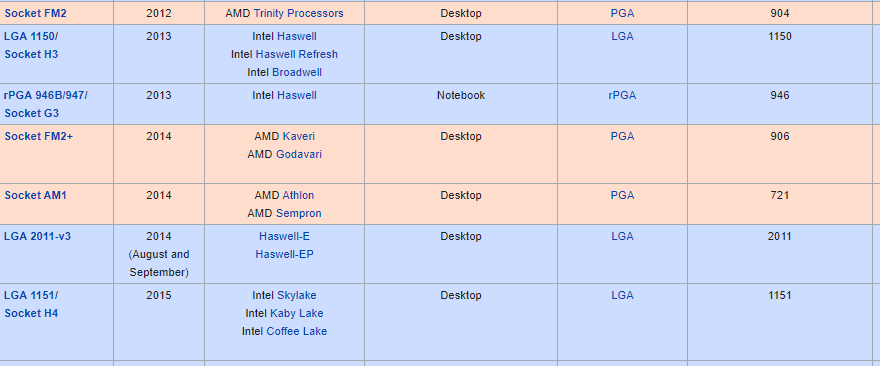
1. optionally have more RAM and better processors?
2. Higher-end audio?
3. better, higher-end graphics cards or CPU?
4. How much can you spend?
5. Do you need overclocking?



## Customers would assembly with incompatible component

It is very dangerous if a computer runs with the consist of the incompatible component. All component needs a compatible port for connection. Two incompatible ports cannot be inserted. Users may lose a lot on the condition that the expensive component does not support each other. Even worst, A user can put a CPU into an incompatible interface of the motherboard. Although two interfaces seem similar but different. The CPU and motherboard would be damage because of it. It is very hard for some customers to look for someone who can help them for checking with e-shop.

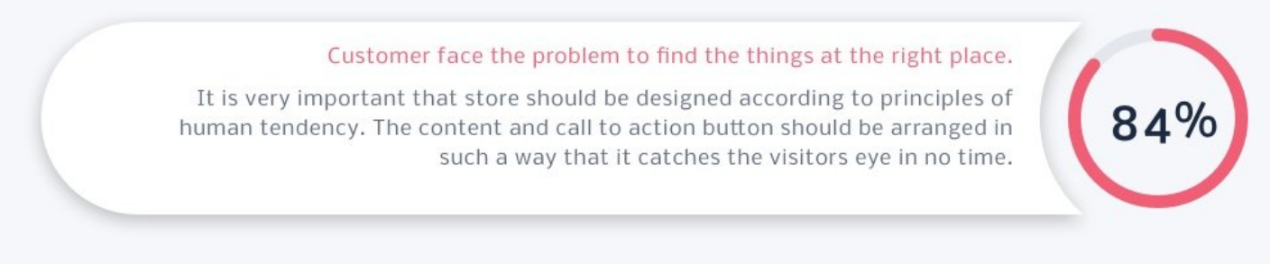
The difference of technology (e.g. DDR of Ram) and interface (e.g. socket of CPU) is very complex. A normal user should not buy any component without enough study and checking.

Socket version of CPU[[5]](#footnote-6):

## Computer e-store with incomprehensive function for customers

As mentioned before, the different types of computer components are a lot. As a result, the maximum inventory should be one of the advantages of a computer e-shop, but it also is a problem.

The traditional way to buy a PC can be very simple with aids from staff. However, if the shop follows the normal franchise e-store, it makes it confusing for the customers. At a physical store other than a computer store, We may need help for introducing of the product, but it is not a problem for e-store customers because the information and reviews of different products can be easy to find, except computer e-store. When it comes to self-assembled computers, knowledge of the computer component is highly requested.

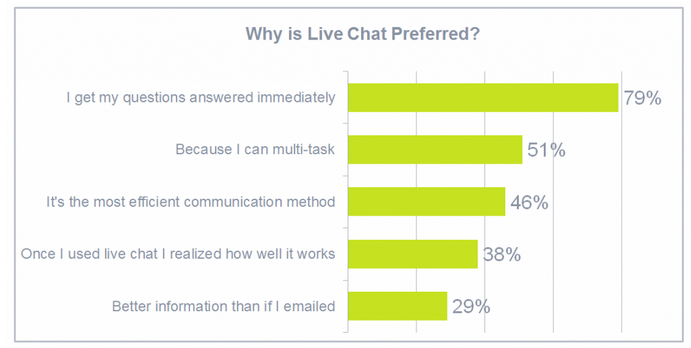


According to Thinkbigonline[[6]](#footnote-7), 84% of customers expressed their desires that it proves convenient for them when they find things on expected places on ecommerce websites. It means they are looking for standard web design conventions to follow by ecommerce web designers. If a system lacks a design for the users’ requirement of self-assembled computers and customers get no help during the purchase process, they would not be interested of the website.

Overall, the design model of online stores should not just consist of traditional online stores.

## Service staff of computer e-store is always busy and not helpful

Most e-shop includes customer service contact, but the staff of computer e-store should be a computer expert. However, even the staff has enough knowledge, the staffs are very busy[[7]](#footnote-8) because the explanation of computer components takes a long time and should be careful. Employing many experts as service staff is quite impossible considering that experts should take a high salary. As a result, without many experts, customer service would be very busy.



Shortage of manpower in a computer e-stop is an extremely difficult problem to solve. According to Reve chat[[8]](#footnote-9), 79% of customers who preferred live chat because they can get the questions answered immediately. As a result, a live chat with any jam world greatly undermine their desire to shop.

Apart from the fact that experts know more compared with most users. However, new technologies and products were created every day. The service staff should not be considered as who can answer everything. Service staff is playing their role badly because they always cannot provide service for customers.

# Solution

## 1 A helpful system specially designed for customers who purchase computer components

**For the problems no. of 4**

### Advantages

According to special requirements of computer e-shop, most of the design of the purchasing page should be changed and targeted for the users who are not able to distinguish the difference between similar components.

The followings are some special design for computer e-shop:

1. Clearly classifying

The list of computer component should classify and filter for what customers need. Including usage, budge, expected performance, and upgradability, more options setting for filter can make the purchase process more efficient and helpful.

The filter is expected to be able to skip the components when if the displaying items are incompatible.

1. Checking function for purchased items

An examination system should be implemented to help customers. The systems should be able to compare the specification and show alert and message if the purchasing items include potential incompatibility issues. For example, F-series chips of Intel CPU are normally lack in a GPU feature. Users who buy F-series independently should be warned to make sure that they have an independent display card before they submit the purchase.

Also, potential effectiveness issues are another item to check. For example, it would be helpful if the system can show a warning whenever the system estimated that the fans is not powerful enough for cooling down the overclocking CPU .

1. Included special purchasing page.

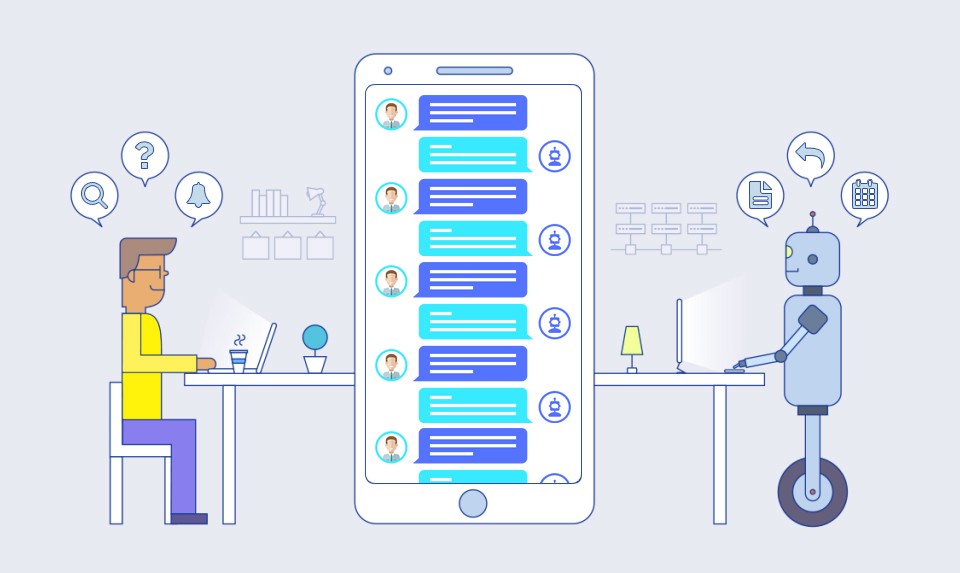
Because the inventory of computer e-shop would only include the computer components. Unlike the traditional e-shop, the purchasing page should be able to show more information for customers’ convenience.

For example, a clear classification of the types of products purchased is expected in their shopping cart (e.g. CPU, GPU, Cooler etc.).

## 2. A chatbot system for aids

**For the problems no. of 1,2,3,5**

### Advantages

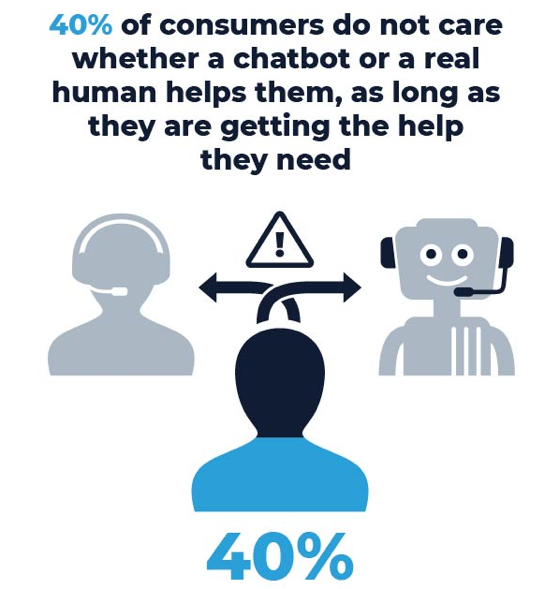
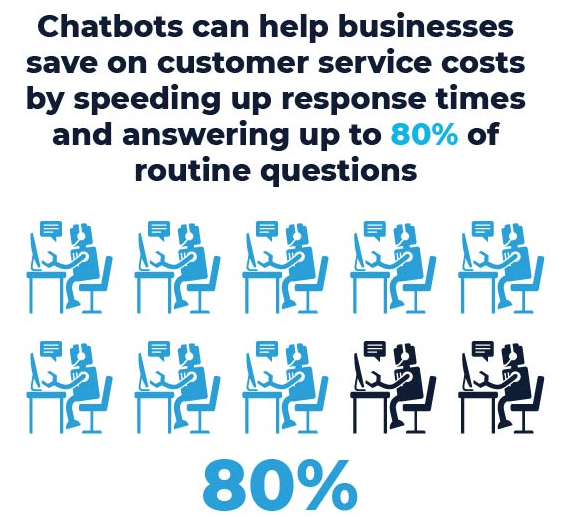


The role of a chat robot is to provide guests with the help they need without using real people to respond. This approach is suitable for large-scale information websites such as computer product websites. Customers can ask what they want and get unique responses. This approach greatly reduces manpower consumption.

Due to the various types of computer products, it is difficult for a website to create a webpage to answer general questions and to reduce the cost of manual answering. Even if all common problems can be listed and display in a website for customers, the number of problems may be as many as thousands.

Advantages to chatbots as the first line of customer support response:

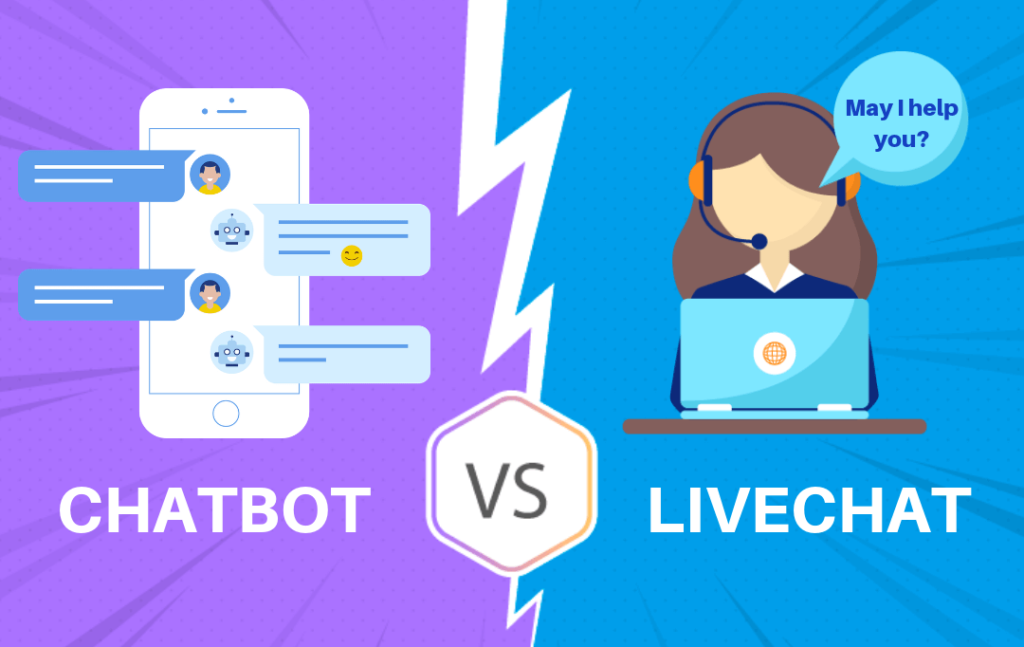
1. Instant 24/7 response
2. Answers standard initial questions (FAQs)
3. Guide customers to information they are looking for
4. Transfers conversations to live agents at the right time

According to a survey from Invesp[[9]](#footnote-10), 40% of consumers do not care whether chatbots or real people are helping them, as long as they can get the help they need. Chatbots can save up to 30% of customer support costs and can help companies save customer service costs by shortening response time and answering up to 80% of regular questions.

Because of the help of API, the chatbot making process is simpler. Most customers also have experience using chat robots.

### Problems of this solution and response



Customers will have their own difficulties when using various products. It is absolutely impossible to provide solutions only by chatbot for all possible problems form people in the world. Even if you have a corresponding solution, it is difficult to ensure that the effectiveness of the solution is 100%. Manual intervention is necessary sometimes.

As a result, the webpage must have an interface and function that guides the user from the right chat robot to the manual service. When the chat robot thinks that it is unable to provide suitable help, it should be able to forward customers to manual service quickly.

By the way, a balance must be struck between chat robots and real-time help, or the problem of high manpower cost will still exist. Therefore, the chat robot should play its role as much as possible, including having response for the question not in the database, real-time price comparison is one of the examples.

## Comprehensive service response function even on mobile for staff

**For the problems no. of 1,2,3,5**

### Advantages

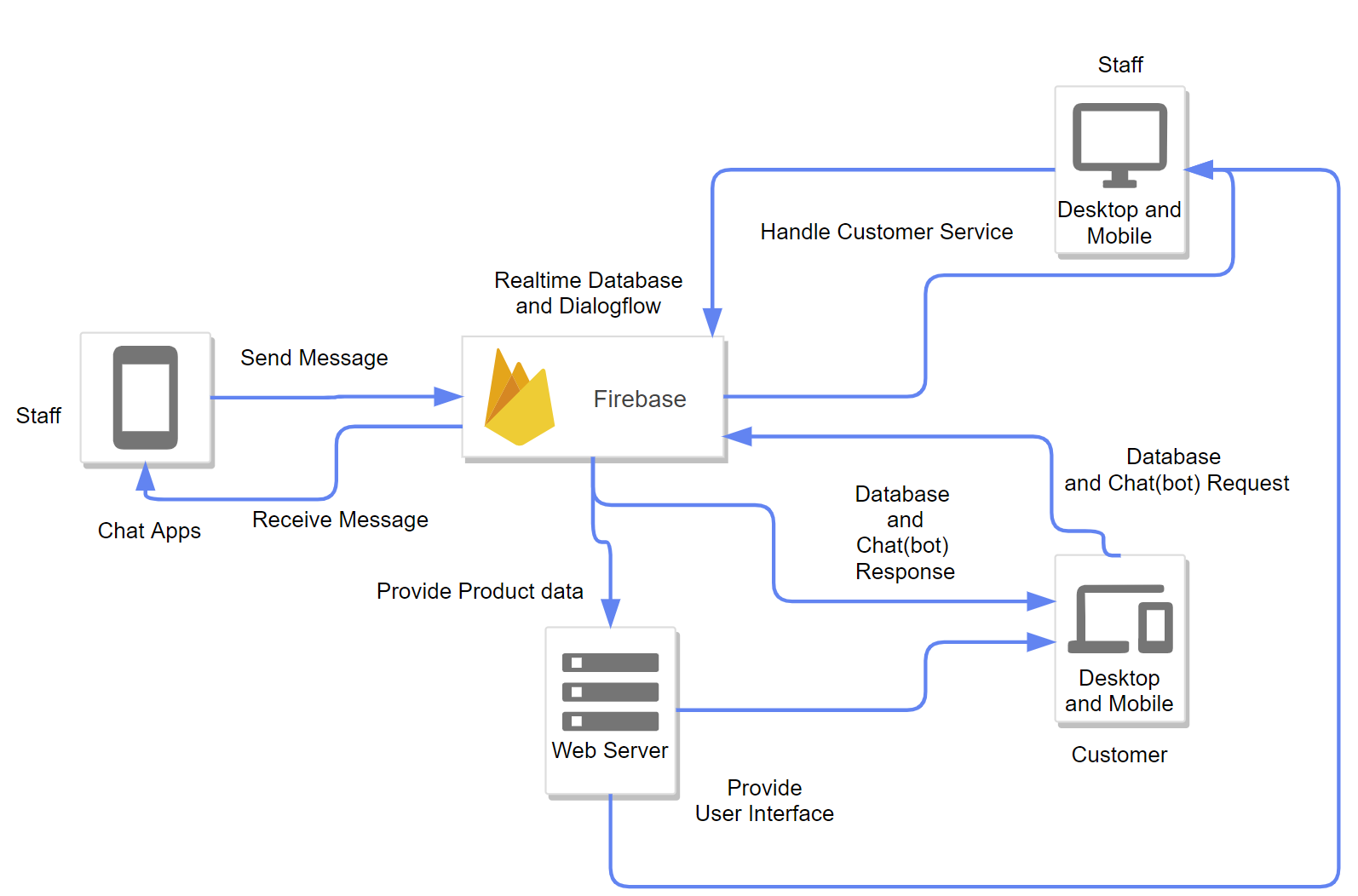
In the question part, it has been said that computer stores cannot hire a large number of experts to help customers. Therefore, in order to make every customer receive help, the system should help staffs as much as possible to quickly solve customer problems.

If the customer is forwarded by the chat robot, the customer’s problem should have been raised in the chat robot. If the system can immediately forward the problem to staff, it will improve efficiency.

The mobile phone is now the most popular tool. If staffs can use the software to find information and provide solutions to the problems raised by customers on the mobile phone, it will improve efficiency. It makes the answering process more convenient and quicker, and also convenient for staffs.

# System Requirements

## System Architecture:



## Backend side:

### **Web Server:**

**Functional Requirement**:

Since our project is a Computer e-store website with chatbot and live chat system, our requirement is of course that a web server should provide web hosting services for customers and company employees to access. Through the web server, our customers can browse and purchase computer products. On the other hand, our staff can log in to website to manage the store.

The web server will write by PHP, the web browser for the customer and staff to request the user interface.

Thanks to Firebase SDK API, call some main function of the whole system will be bypass the web server, it means the user side can communicate the Database and chatbot directly.

**Reliability Requirement**:

1. 24x7 Operation:

Because some customers may require products to be shipped overseas, and these customers may be overseas. Since the time zone of Hong Kong and overseas metropolitan areas are too different. If the website can only operate during office hours, it may affect the service of overseas customers.

1. Effect from the environment:

Since shops often provide special time slots(Black Friday), website traffic will increase greatly during special time periods and the load on the server will be too heavy. Therefore, we are equipped with powerful hardware to cope with the special time period, but this will lead to a rapid increase in power consumption, so the server should equipped with a power saving mode to avoid wasting power during normal hours and reduce electricity bills.

**Performance Requirement:**

1. Bulk request and response:

As mentioned in the Reliability Requirement, website traffic will increase greatly during special time periods and the load on the server will be too heavy. Tew web server must have sufficient network bandwidth and hardware performance to process order, so that the server runs smoothly at all times.

**Hardware Requirement:**

1. Fiber Broadband Network Interface:

Using optical fiber to transmit network packets is much faster that of copper wire, so that network will not become a performance bottleneck. It will connect to commercial broadband.

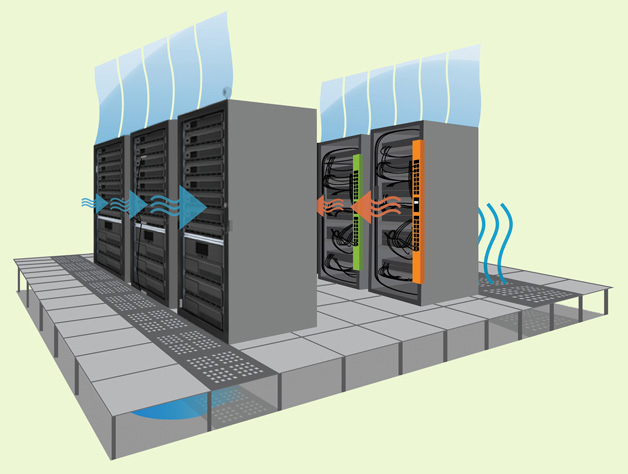
1. At least 18 core CPU, 128GB RAM and 4TB NVMe SSD:

Ensure adequate performance.

1. 80 PLUS TITANIUM Certified Power Supplies and UPS:

Equivalent to the energy label of the power supply, it is the highest grade, on the other hand, it also guarantees the reliability.

1. Reliable cooling system



**Database:**

**Functional Requirement:**

Stores need media to store product information, orders, chat record and user data. Compared with the file-bases system, Relational Database(or NoSQL) are the most reliable and best-performing storage system today. It easy to maintain.

For NoSQL, it has the advantage of easy to learn how to operate the database.

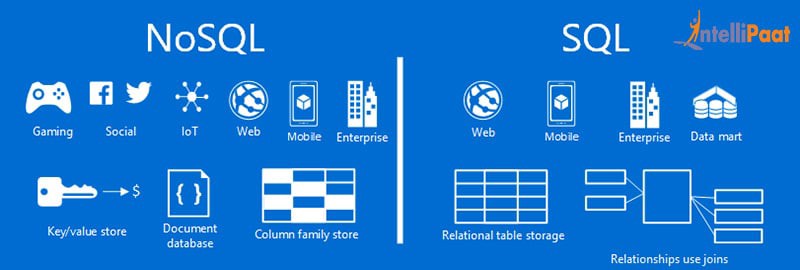
The database will use the Realtime Database.

The chatbot and real time chat system must rely on the database to function properly.

Real-time chat system will use NoSQL database.

The database must provide basic functions such as query, update, insert and delete.

The database must provide API to connect chatbot and live chat system between end-user.

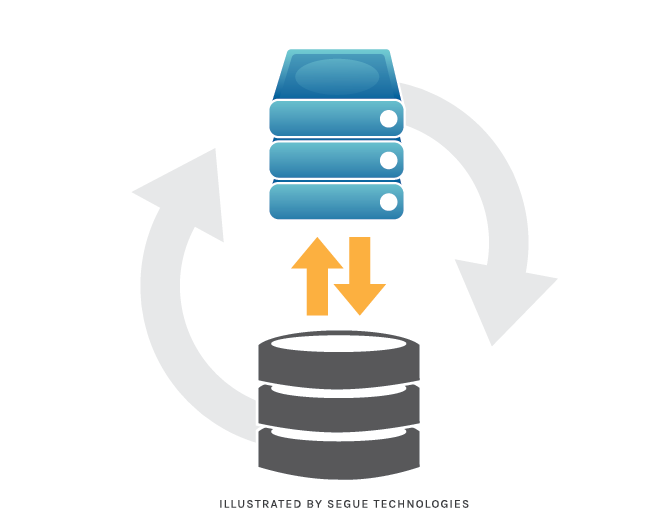


**Reliability Requirement:**

In addition to what is mentioned in the web server section, it also includes:

1. Automatic backup:

During the operation of the store, it is inevitable that there will be natural disasters and man-made losses that will cause data damage. The automatic backup function can restore the entire database data in the last backup to minimize the loss.



**Performance Requirement:**

Same as mentioned in the web server section.

**Hardware Requirement:**

Same as mentioned in the web server section

.

**Chatbot:**

**Functional Requirement:**

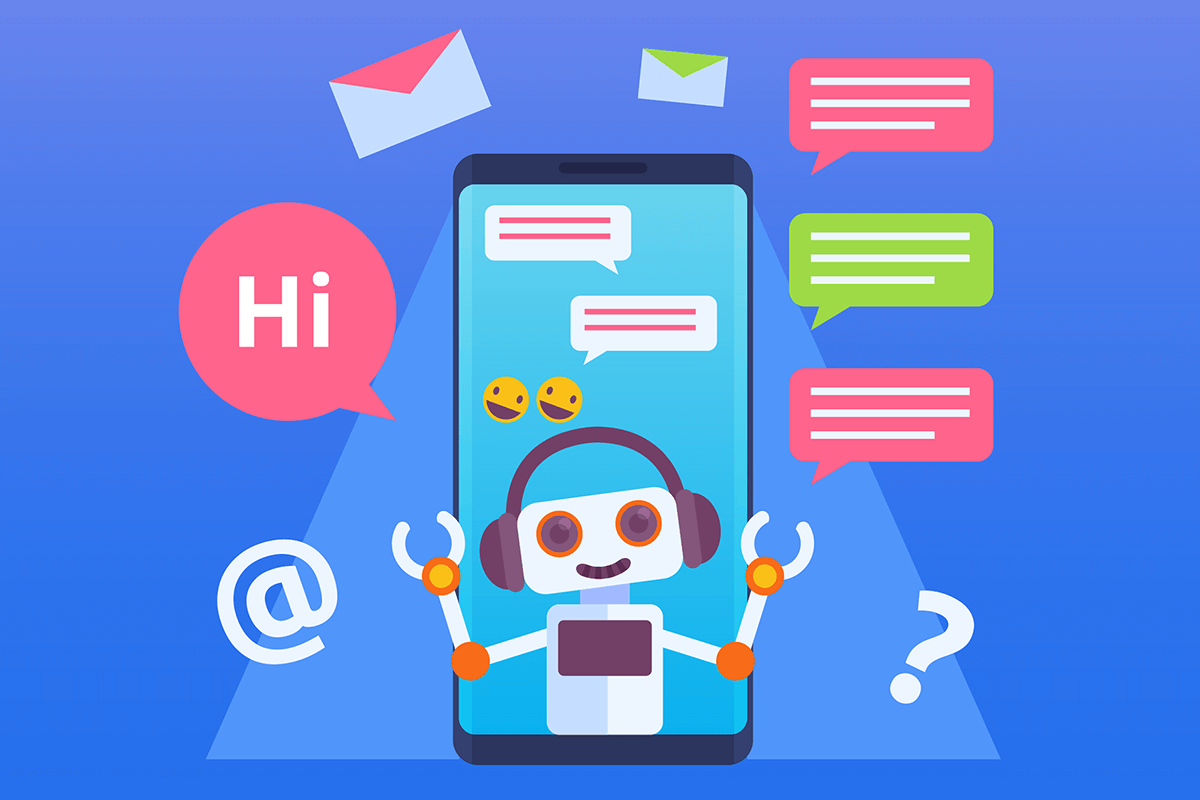
The purpose of setting up a chatbot in the computer e-store is to answer simple questions from customers.

For example: Is this processor can be use for that motherboard?

However, how does the chatbot know how many pins the processor has, so the chatbot must communicate with Database, answer the user’s request after getting the product specifications from database.

The chatbot must have the ability to read and process natural language.

Developers can pre-set some questions to guide Ai-chatbot to learn answer questions.



**Reliability Requirement:**

In addition to what is mentioned in the web server section, it also includes:

1. Hazardous command detection:

If the user requests any request to modify the database, the chatbot will redirect to the live chat system, let employees handle this process.

In addition, if user sends a very long string, the chatbot should ignore.

**Performance Requirement:**

In order to make user not feel irritated, the chatbot respond to user requests within 5 seconds.

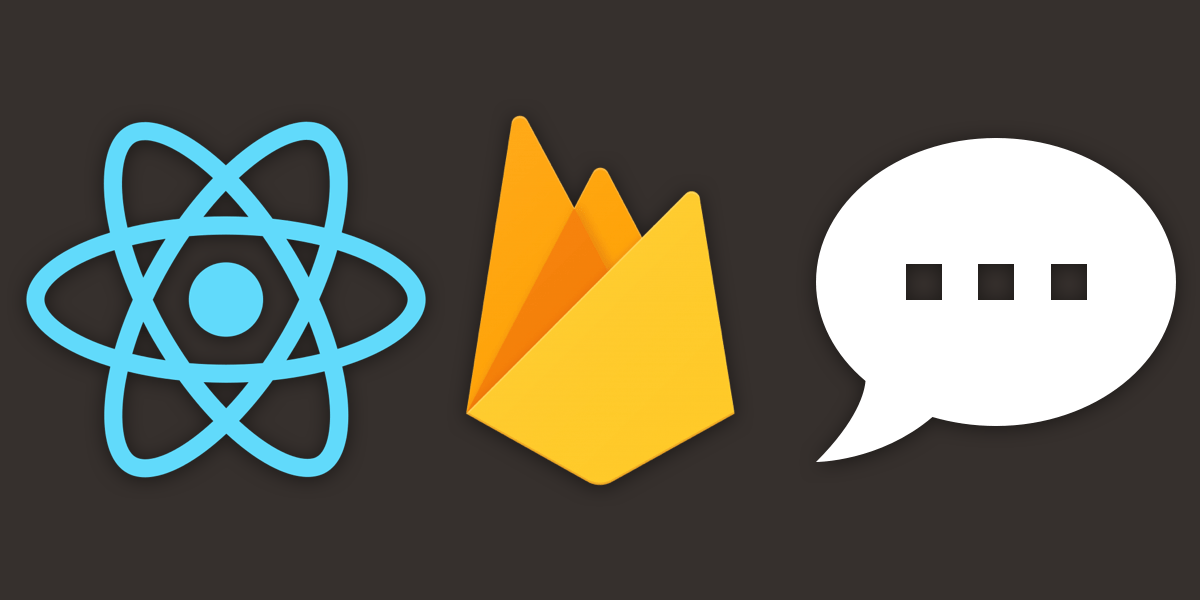
**Hardware Requirement:**

Since the chatbot we plan to use is a cloud service, there is no need to purchase hardware

### **Live chat system**:

**Functional Requirement:**

If the chatbot cannot solve the user’s problem, it will redirect the real-time chat system to allow employees to solve the user’s need. The system must support web and mobile apps to communicate with each other. Voice chat is optional.

****

**Reliability Requirement:**

Same as mentioned in the web server section.

**Performance Requirement:**

In order to make user not feel irritated, the message must be sent in a short time.

**Hardware Requirement:**

Since the Live chat system we plan to use is a cloud service, there is no need to purchase hardware.

## Frontend side:

The user side is the application and web page which the user will user to receive and view the information.

### **Store webpage:**

**Functional Requirement:**

The first time a customer contacts the store is on the homepage of the website. The homepage should contain search, registration, and login.

Function details:

1. Search

User can search or product

1. Registration

If a user needs to purchase a product, he must register as a member of the site

1. Login

After registering as a member, customer must log in to the website to place an order.

After the employees log in to the website, they can manage the store.

1. Order
2. Ask

If the customer has any questions about the product or the order, it can be solved through the chatbot or real-time chat system

1. Manage

Employees can change and cancel user orders after receiving the user’s request in the chat system.

**Performance Requirement:**

Able to browse website smoothly.

**Hardware Requirement:**

The device used by the user must be able to connect to internet.

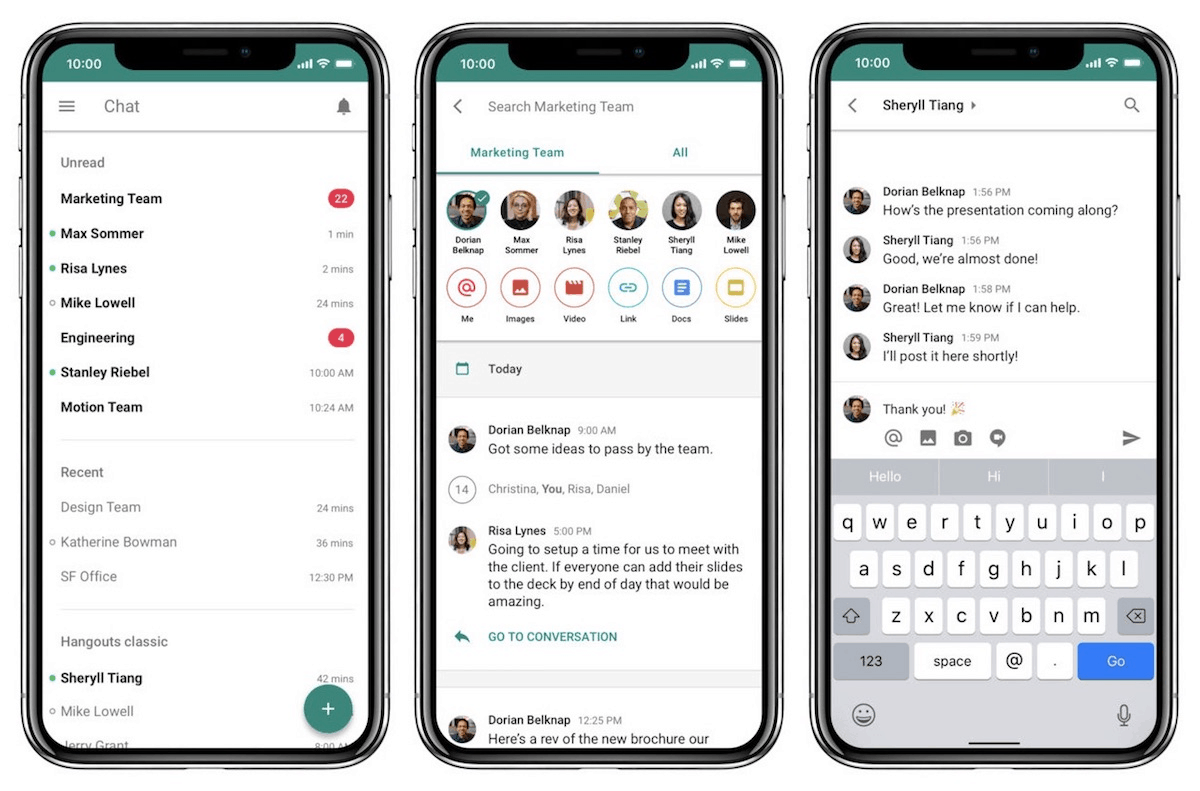
The browser used must support HTML5, CSS3, JavaScript ES6, Cookie.

### **Chat apps:**

**Functional Requirement:**

In order to make it easier for employees to work from home, employees can manage the shop through the website. Some employees are not familiar with using Chinese input methods, and also their computers are not have microphone, and to facilitate chatting with different customers in a short time, so the chat apps for employees is for smartphone only. To use two different devices to improve the efficiency of multitasking.

Function details:



Can be grouped according to customers or employees.

**Performance Requirement:**

The smartphone should able to use the apps smoothly.

**Hardware Requirement:**

The smartphone must be able to connect to internet.

### Suggest Specification

#### Web server:

|  |  |
| --- | --- |
| Operating System | Windows Server 2019 |
| Memory | >32GB |
| CPU | >8 Core CPU |
| Storage | >1TB SSD |
| Backend Language | >PHP 7 |

#### Staff Mobile Phone(For chat apps):

|  |  |
| --- | --- |
| Operating System | >Android 10 |
| Memory | >8GB |
| CPU | >2+4 Core CPU |
| Storage | >64GB |

#### Web user:

|  |  |
| --- | --- |
| Operating System | Android 10+ /ios 14 /windows 10 |
| Memory | >4GB |
| CPU | >2+4 Core CPU |
| Storage | >64GB |
| Browser | Latest version Google Chrome |

# Non-Functional Requirement

|  |
| --- |
| Product requirement |
| Security **Security (Authentication & Data Protection):**  Firebase account and token only authorized person can access  All users must login to the system to use the function.  Customer information will be secured by encryption.    **Security (Data Access Right Control and Availability):**  To manage each of the user right and data access right control, the system will set up some account with different right for each of staff and customers.  **Security (Data Integrity):**  Shop will use entity integrity, referential integrity and domain integrity for the database.  **Security (Accessibility):**  Do not allow staffs to log in to the account via the private computer directly, while only allow staffs to login into the company computer or via remote desktop.  **Security (Backup):**  Auto-Backup will be done in every day’s midnight. |

|  |
| --- |
| Organizational requirement |
| Staff requirement **Staff requirement—Manager**  Must not share the password to any people, otherwise punishment will be given to them.  If the company hire or fire the staffs, manager should ask IT department to create and delete the related accounts of those staffs.    **Staff requirement—Staffs**  Must not share the password to any people, otherwise punishment will be given to them.    **Customer requirement**  Must not share the password to any people.    Recommended customers update the latest OS to ensure the security between customers and our computers. |
| Manual of the system There will be written 2 versions of the manual.  The manual with the repair and setup details will be written for the IT department.  The manual for the operation of the staff website and chat apps will be written for our staff.  The manual for the operation of the shop website will be written for customers. |
| Quick support When the error(s) occur, the related error code will appear on the screen. Then, the user can provide the error code and the description to the IT department by email/ phone call.    Then, the IT department will reply to the users. |
| System maintenance Auto-Backup will be done in every day’s midnight. |

|  |
| --- |
| 5.3 External requirement |
| Database connection The database server will cut some kinds of disturbing connections, such as cut the connection that use for too long when too many clients attempt to access the database. |

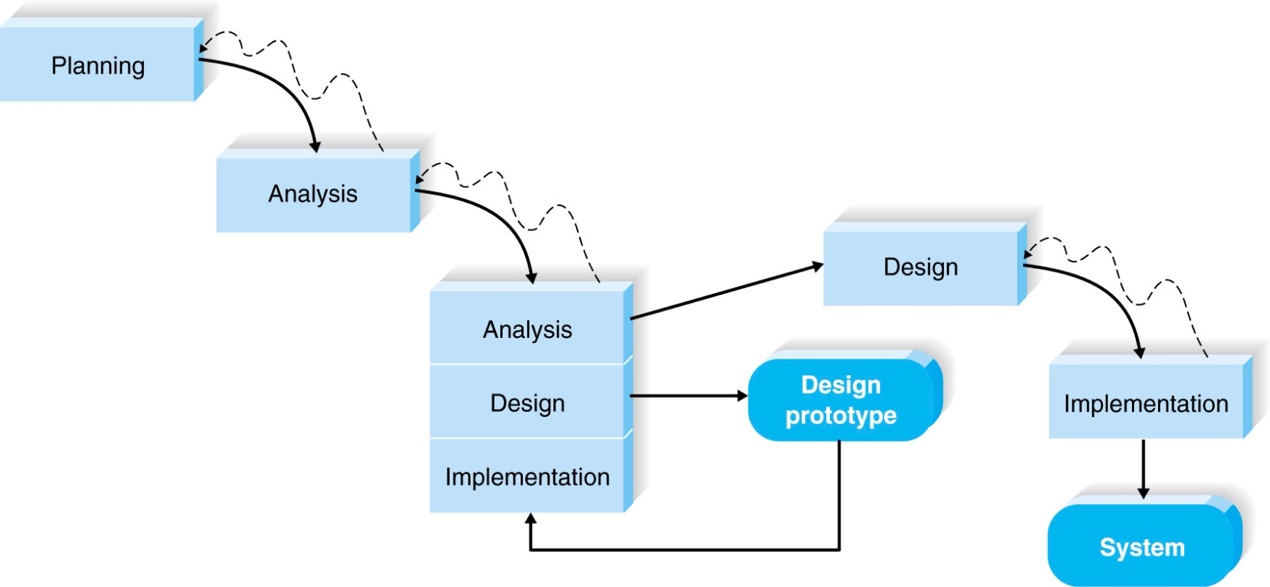
## Scope of the proposed system

The proposed system will include three part. The first part is the firebase system, all major functions run on the system. Secondly, the web server, it provides user interface and library to user web browser. Finally, The web browser will display product information from the firebase system and be able to send order request to firebase. Staff can receive message through mobile apps from other staff and customers.

# Software develop process model

Throwaway Prototype Model was preferred to be our system’s process model. The main reason the project choose Throwaway prototyping is that it let us to use prototypes to refine key issues before a system is built. Before the system is established, the problem can be determined and a more reliable system can be produced. Yet, It will take more time than the prototype method. Because of the overall analysis and design, and the need to rewrite the program, it takes more time.

### **A simple structure of Throwaway Prototype Model**

****

According to the Throwaway prototype model, the work plan for each phase should be planned before the actual start. The prototype design, analysis, design, implementation, and the testing cycle continues until the project meets planning phase requirements. This is the expected workload of each stage of the development project.

### **Planning:**

At this stage, the project needs to collect all the requirements of the system to be developed. This stage can be divided into three main parts.

Problem analysis:

Check the historical data to show that the current problem has occurred. The solution part will analyze and describe the collected data.

Solution:

Analyze the received data and identify some possible solutions to solve the problem described in Problem Analysis.

Requirement(s):

This part describes the actual functioning of the system and lists the reliability requirements, performance requirements, and hardware requirements of the system.

### **Design:**

The design phase determines how your system works in terms of hardware, software, and network infrastructure. User interfaces, charts, and reports used. Specific software, databases, and files required.

System Design:

The Use Case Diagram, Class Diagram, Sequence Diagram and State Machine Diagram of the system will be designed on this part.

Database Design:

Entity-relationship model and data dictionary of the database will be designed on this part.

Hardware Design:

All hardware must be able to withstand the explosive traffic during peak periods.

Software Design:

APIs between different software must function properly.

User interface Design:

Since the website will be viewed by different devices, the website must automatically adapt to the device’s resolution adjustment interface size.

### **Prototyping:**

The information gathered in the quick design is modified to form a prototype. This is the rough layout of the required system. Assess the user as to whether they meet the following system requirements: If the user is not satisfied with the first prototype, the next prototype will conduct.

### **Implementation:**

During the implementation phase, the entire system is divided into many modules. The number of modules depends on the number of functions implemented.

### **Verification:**

In this phase, the test plan is created for each test case. All test cases must pass the integrity check of the implemented system. This phase continues until all modules work together as expected. A user acceptance test is then carried out so that the user can determine whether the system is complete.

Module Test:

For module testing, each implemented component can operate independently.

Integration Test:

Combine modules to conduct integration tests. The purpose of this part is to ensure the modules work together.

System Test:

Test the whole system working properly or not. The chatbot, website and live chat system would test together to check the whole system is completed.

User Acceptance Test:

Users test the software to ensure that the whole system can perform the required tasks in real-world scenarios according to specifications.

**Maintenance:**

The maintenance phase occurs when the system is operational. This includes implementing changes that the software may undergo over a period or implementing new requirements after the software has been deployed to clients. The maintenance phase also includes the handling of residual errors that may still be present in the software after the test phase. In this phase, we monitor system performance, fix errors and are in the requested change are made.

# Function

## E-shop Website

**Chat bot**

This function is the chat bot system on the website, the chatbot would be included in the Chat Box environment. The chat box on the website would be hold by the chat bot first when the page is loaded. The chat bot is functioned by Dialogflow form Google Cloud. The chatbot was already trained for the sake of customers and the chat bot has a connection with the database system of our Firebase server.

Data Description:

|  |  |  |
| --- | --- | --- |
| Data Name | Data Type | Data Description |
| Message | String | The message sent or received by users. |
| Id | String | The identifier for chatbot to recognize specific user. |

**Real time chatting**

This function is the communication function for users and the staffs of the company. When the chatbot recognized that user request human handing over, the chatbot would change the receiving side from bot system to the communication system.

Data Description:

|  |  |  |
| --- | --- | --- |
| Data Name | Data Type | Data Description |
| Message | String | The message sent or received by users. |
| Id | String | The identifier for chatbot to recognize specific user. |
| Receiving side | String | The identifier of the receiving side. |

**Login and Register**

This function will allow the user to register his owner's email and password. Then, they can log in to the e-commerce web. On the web, they can save his payment way, such as master card, we chat and save their location for online shopping.

Data Description

|  |  |  |
| --- | --- | --- |
| **Data Name** | **Data Type** | **Description** |
| User ID | String | Saving their ID for auto-login and create an order and could check whether the user has a login. |

**Computer Product Guide**

This function allows the users to choose computer components by themselves with an automated guide. This system allows the user to choose the components including CPU, GPU, motherboard, CPU cooler, Case, RAM, etc.

Data Description

|  |  |  |
| --- | --- | --- |
| **Data Name** | **Data Type** | **Description** |
| select Option | String | Get user-selected type in computer components |

**Add product to cart**

When the user selected a computer product, this function allows the user to add it to his own shopping cart and list all product which his selected and showing the total price for all products.

Data Description

|  |  |  |
| --- | --- | --- |
| **Data Name** | **Data Type** | **Description** |
| Product ID | String | Saving the product information which user selected |
| Product Quantity | int | Saving user selected quantity for each product |
| Total price | float | Calculate the total price with all product |

**Payment Processing**

This function allows the users to select a payment way such as cash to make an order with a product.

Data Description

|  |  |  |
| --- | --- | --- |
| **Data Name** | **Data Type** | **Description** |
| Payment Information | String | Saving the payment information such as card number |
| Product ID | String | Store the id of ordered product |
| Delivery address | String | For product delivery |
| Payment method | String | Payment method for the order, such as PayPal, Visa |

**Product Searching**

This function allows the users to search for any product in the product guide page, all products could search by any key words, for example I7 – 10700K, input any words can be search it, ‘I7’, “10700K” etc.

Data Description

|  |  |  |
| --- | --- | --- |
| **Data Name** | **Data Type** | **Description** |
| Key Word | String | User input for search products |

**Forget Password**

This function allows the users who forgot passwords to reset the password with a page, in that page, user can input his own email address, so that system can send an email about reset password for user.

Data Description

|  |  |  |
| --- | --- | --- |
| **Data Name** | **Data Type** | **Description** |
| Email | String | Email for reset the password |

**Setting Account Information**

This function allows the users to setting any information in his own account, including first name, last name, password, email address, delivery address, and mobile.

Data Description

|  |  |  |
| --- | --- | --- |
| **Data Name** | **Data Type** | **Description** |
| Customer ID | String | Identify which account will be modifying. |
| First Name | String | Modifying account’s first name |
| Last Name | String | Modifying account’s last name |
| Email Address | String | Modifying account’s email address |
| Password | String | Modifying account’s password |
| Address | String | Setting the delivery address |

**List Orders**

This function allows the users to list all orders which he ordered. It will be displaying all information in each order, including order ID, delivery address, all products bought in this order, each product price, amount of bought.

Data Description

|  |  |  |
| --- | --- | --- |
| **Data Name** | **Data Type** | **Description** |
| Customer ID | String | Get all order from the customer. |
| Order ID | String | Display the order ID, also it can get all information from the order ID. |
| Delivery Address | String | The order’s Delivery address |
| Total Price | String | Display orders total price. |
| Product ID | String array | Save all product ID bought in the order, it can use to get product information. |
| Amount | Integer | Number of products bought in this order. |
| Product Price | Double | Display each product price, get from product\_ID |
| Date | Datetime | The order date time, hour, day, mouth |

**Comment for product**

This function allows the users to comment for a product, user also can check other user comments.

Data Description

|  |  |  |
| --- | --- | --- |
| **Data Name** | **Data Type** | **Description** |
| Customer ID | String | Detect the comment send by which user. |
| Comment Content | String | User’s comment content. |

## Chat Apps

### Login

Function Description:

The function requires the user to log in before sending any message, and it used to confirm the user’s identity. If the user has logged in, it also support the automatic login.

Data Description:

|  |  |  |
| --- | --- | --- |
| Data Name | Data Type | Data Description |
| firebaseUser | FirebaseUser | If the user has logged I before, it will save user token |
| firebaseAuth | FirebaseAuth | It is Firebase Authentication API, use to confirm user identity |
| String | email\_id | MaterialEditText View for input user email |
| String | password | MaterialEditText View for input user password |

### Send Message

Function Description:

The function handle flow of sending message.

Data Description:

|  |  |  |
| --- | --- | --- |
| Data Name | Data Type | Data Description |
| firebaseAuth | FirebaseAuth | It is Firebase Authentication API, use to confirm user identity.In this function, It can get current user UID. |
| String | sender | Store Current user UID |
| String | receiver | Store Receiver User UID |
| String | message | Message content |
| Boolean | IsSeen | Check the message is or dot seen yet |
| databaseReference | DatabaseReference | Database API |
| HashMap<String, String> | map | data to be written to the database(JSON Tree structure) |

# Problem Analysis

## Use case

|  |  |
| --- | --- |
| Use case name: | Login |
| Use case ID: | App-01 |
| Super use case: | - |
| Application: | Mobile Application |
| Actor(s): | Staff |
| Preconditions: | - |
| Description: | To Authenticate the user id. |
| Basic Flow: | 1. Input email and password 2. Press LOGIN Button |
| Alternate Flow: | If user not fount, the message box(Toast) will show” Authentication failed!” |

|  |  |
| --- | --- |
| Use case name: | Send Message |
| Use case ID: | App-02 |
| Super use case: | Login |
| Application: | Mobile Application |
| Actor(s): | Staff |
| Preconditions: | Login |
| Description: | To communicate with other user |
| Basic Flow: | 1. Type something in textbox 2. Press  Button |
| Alternate Flow: | If textbox is empty, the send button will be disable. |

|  |  |
| --- | --- |
| Use case name: | Make a phone call |
| Use case ID: | App-03 |
| Super use case: | Login |
| Application: | Mobile Application |
| Actor(s): | Staff |
| Preconditions: | Login |
| Description: | Dial to receiver, the receiver phone number depends on the user database data . |
| Basic Flow: | Click user in main page  Click  Button  5. Click  Button |
| Alternate Flow: |  |

|  |  |
| --- | --- |
| Use case name: | Send photo |
| Use case ID: | App-04 |
| Super use case: | Login |
| Application: | Mobile Application |
| Actor(s): | Staff |
| Preconditions: | Login |
| Description: | Send photo to receiver |
| Basic Flow: | Click user in main page  Click Button  Select one photo  Click send button |
| Alternate Flow: |  |

|  |  |
| --- | --- |
| Use case name: | Chat bot |
| Use case ID: | Web-01 |
| Super use case: | - |
| Application: | Application |
| Actor(s): | customers |
| Preconditions: | - |
| Description: | Connection with chat bot |
| Basic Flow: | 1. When users open the chat box, the system would be executed.  2. User can communicate with the chat bot on the chat UI when the connection built. |

|  |  |
| --- | --- |
| Use case name: | Human chatting |
| Use case ID: | Web-02 |
| Super use case: | - |
| Application: | Application |
| Actor(s): | customers |
| Preconditions: | - |
| Description: | Staffs would hand over the chatting with user |
| Basic Flow: | 1. The chat with chat bot would be end. 2. A connection between client side and communication database. 3. User can communicate with the staff on the chat UI when the connection built. |

|  |  |
| --- | --- |
| Use Case ID | UC-22 |
| Use Case Name | Create Account |
| Brief Description | Create a customer account |
| Actor | Customer |
| Main   1. Open the system website. 2. Click the Register button 3. Input customer information including first name, last name, email and password | |

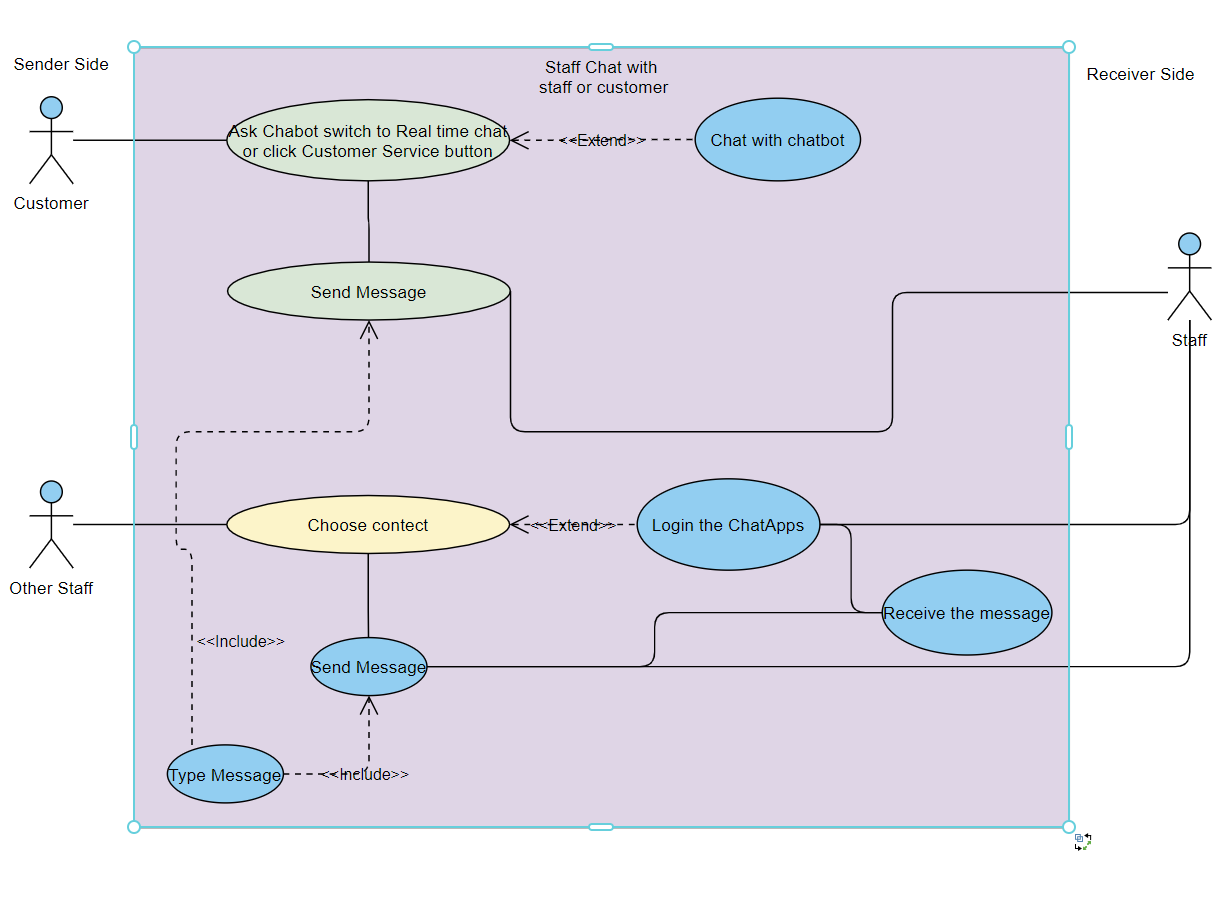
|  |  |
| --- | --- |
| Use Case ID | UC-23 |
| Use Case Name | Login (Website) |
| Brief Description | Login in with customer |
| Actor | Customer |
| Main   * Open the system website. * Click the Login button * Input customer email and password | |

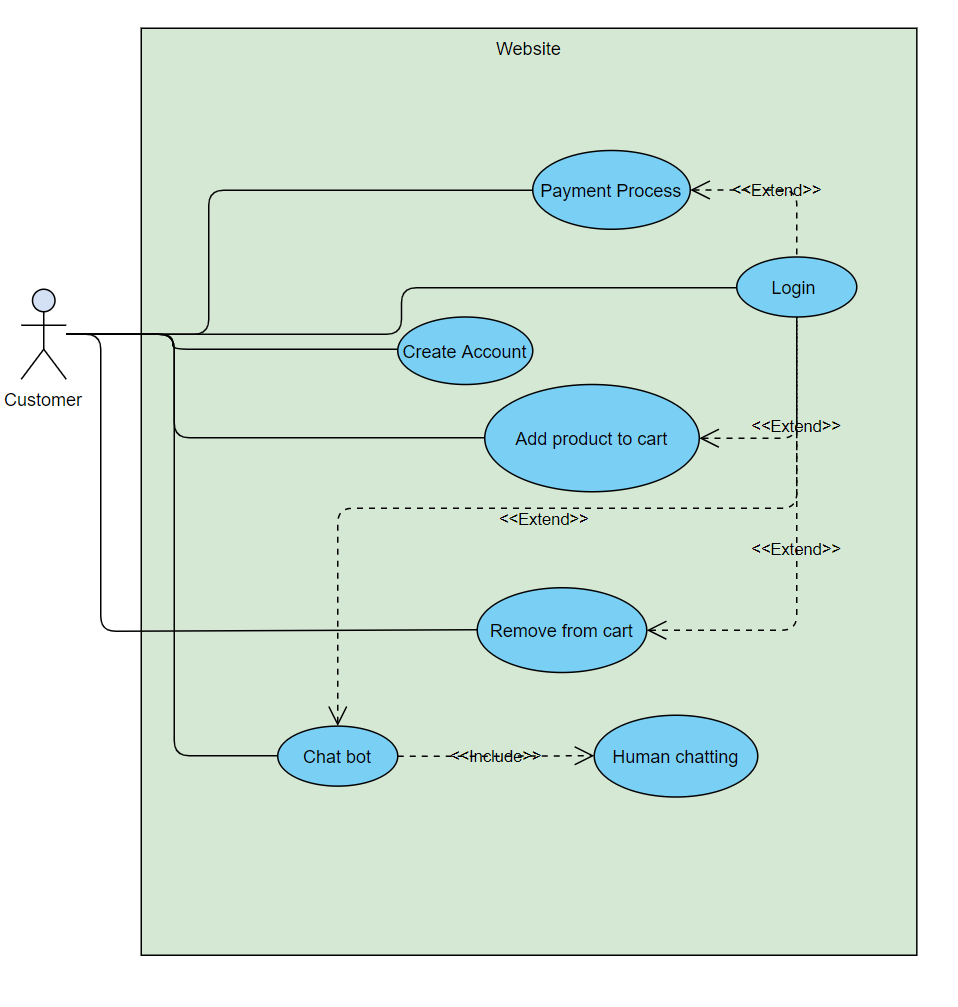
|  |  |
| --- | --- |
| Use Case ID | UC-24 |
| Use Case Name | Add product to cart |
| Brief Description | Select a product and add to cart |
| Actor | Customer |
| Main  Include (UC-23)   1. Go to product guide page. 2. Select a product and click add to cart button. | |

|  |  |
| --- | --- |
| Use Case ID | UC-25 |
| Use Case Name | Remove from cart |
| Brief Description | Remove a product from shopping cart |
| Actor | Customer |
| Main  Include (UC-23)  Include (UC-24)   1. Change to shopping cart page. 2. Select a product and click cancel button. | |

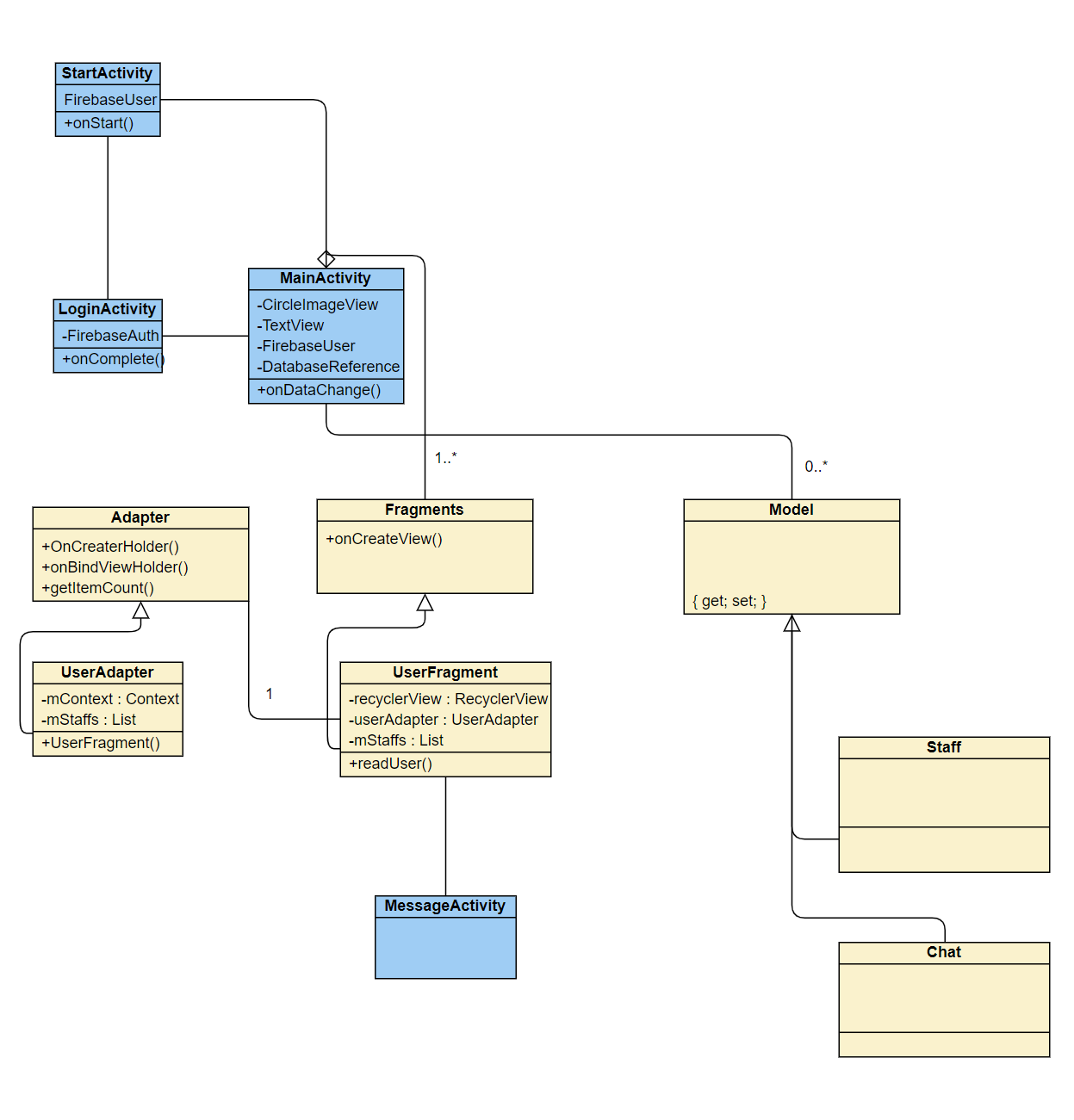
|  |  |
| --- | --- |
| Use Case ID | UC-26 |
| Use Case Name | Payment Process |
| Brief Description | Complete a payment process |
| Actor | Customer |
| Main  Include (UC-23)  Include (UC-24)   1. Change to shopping cart page and click payment. 2. Select a payment way and input the information | |

# Use Case Diagrams



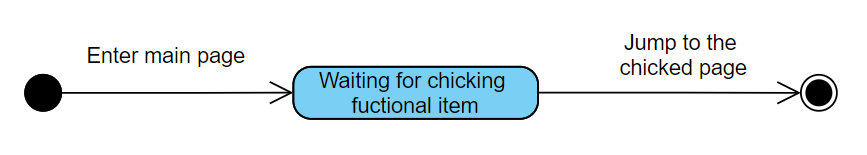


# Class Diagram

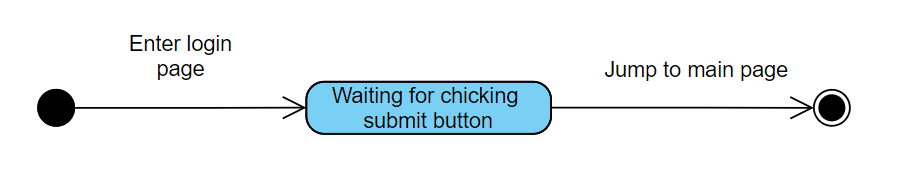


# State Machine Diagram

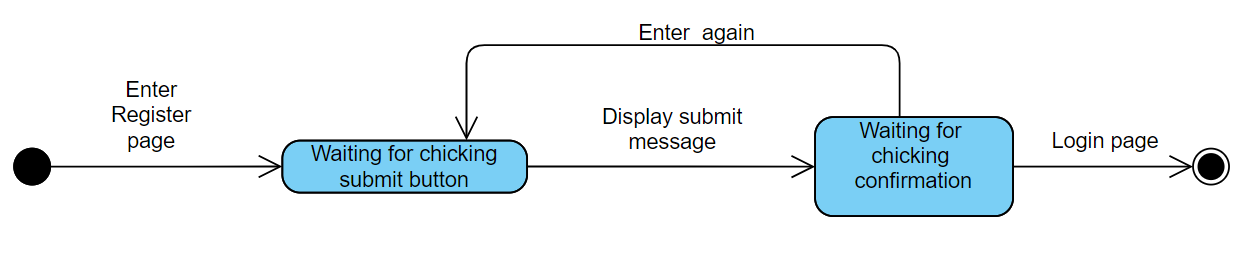
Website main and informational page



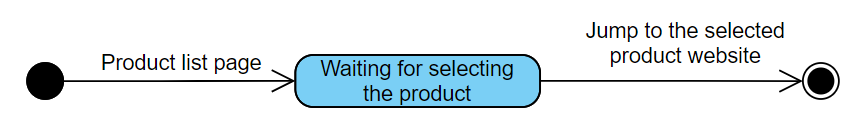
Website login page



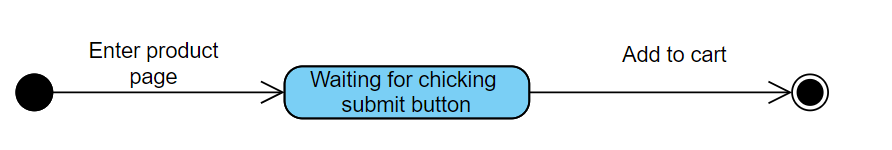
Register page



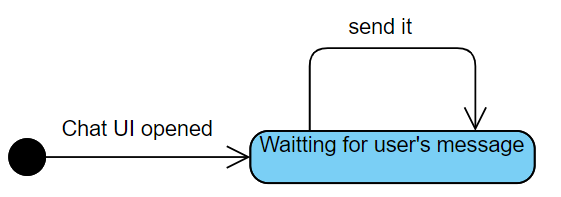
Product list page



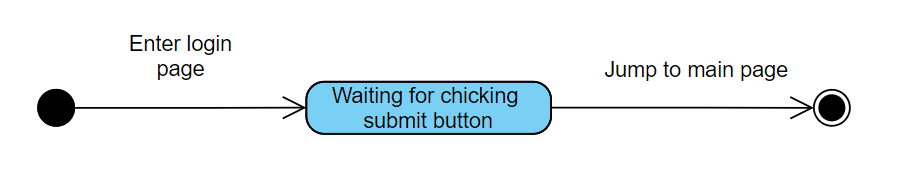
Product page



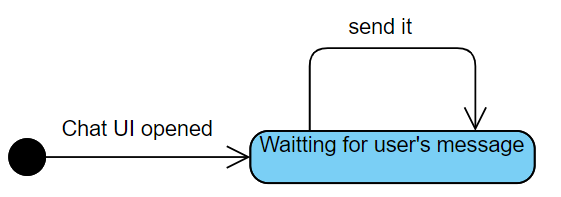
Website Chatting



Mobile login page

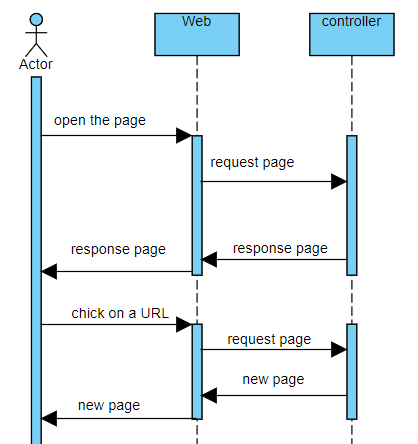


Mobile Chatting

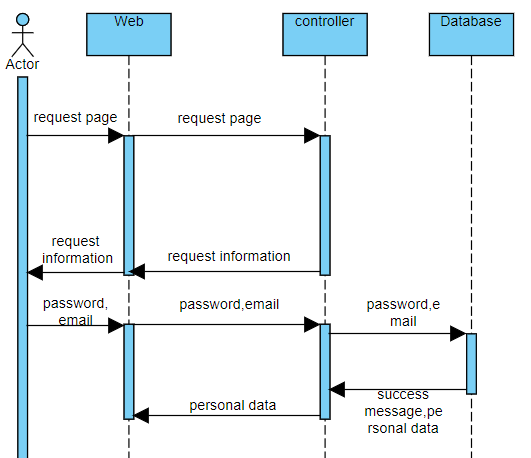


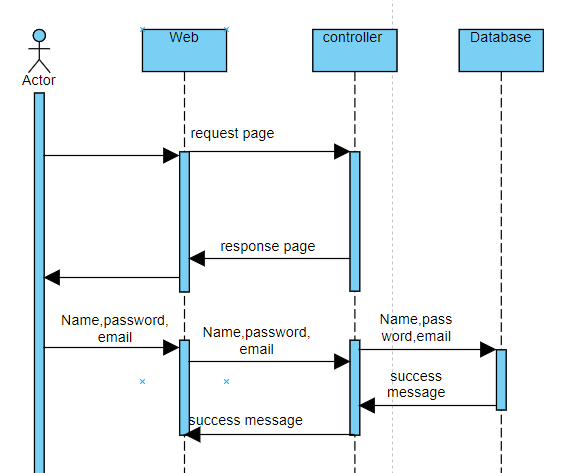
# Sequence Diagrams

Main page and all informational page

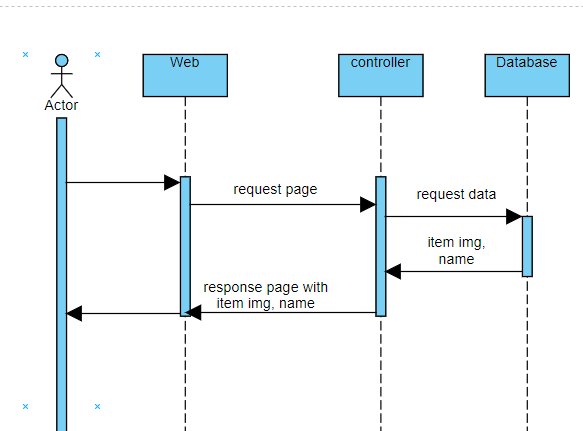


Website login page

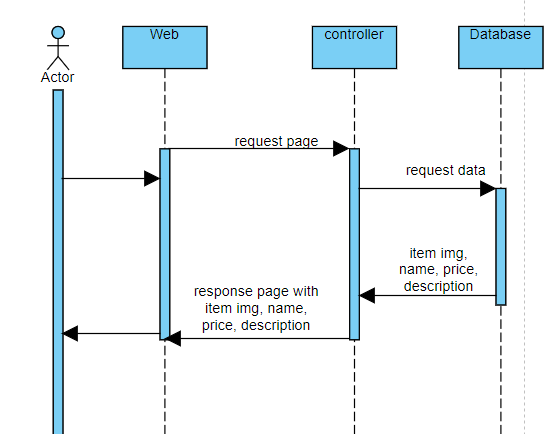


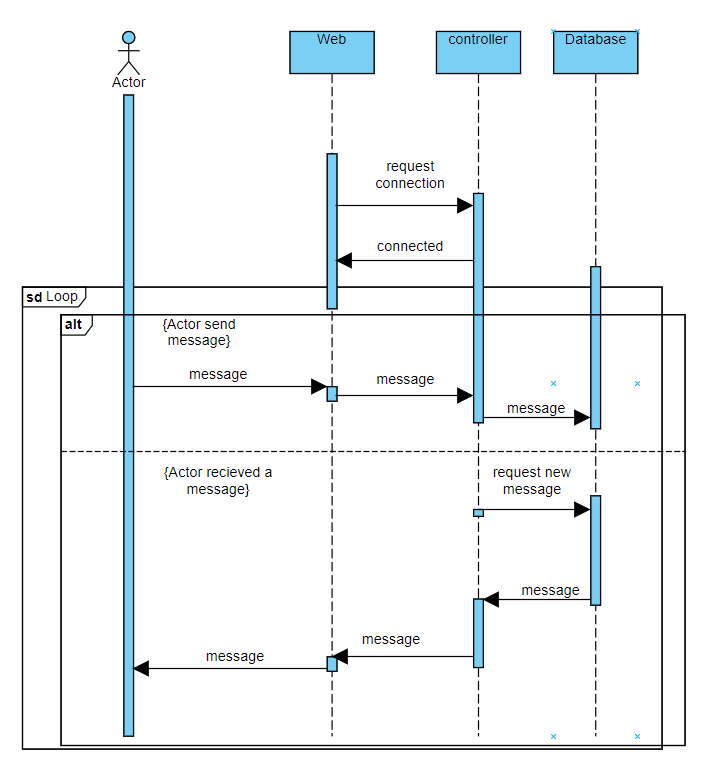
Register page 

Product list page

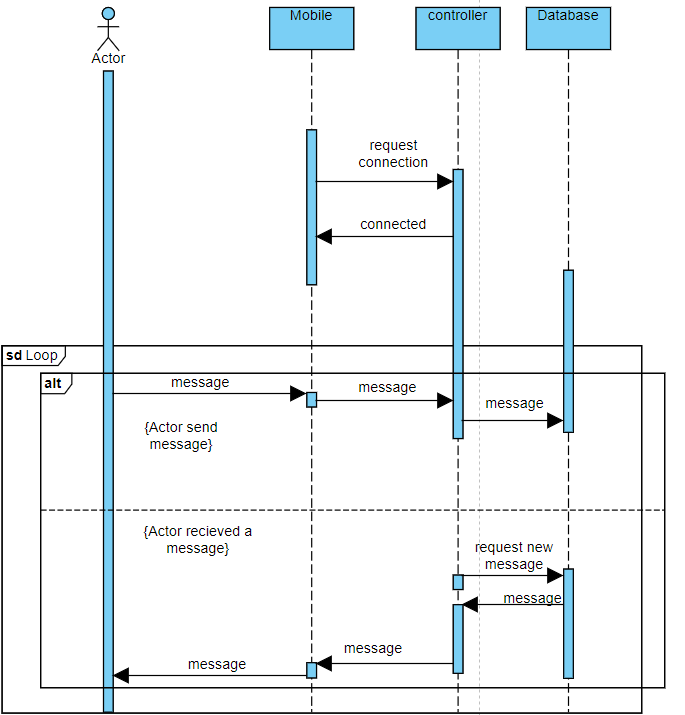


Product page

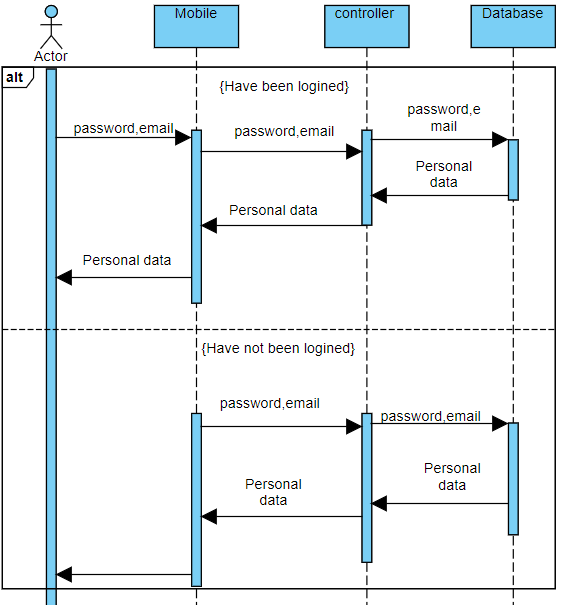


Website Chatting

Mobile Chatting

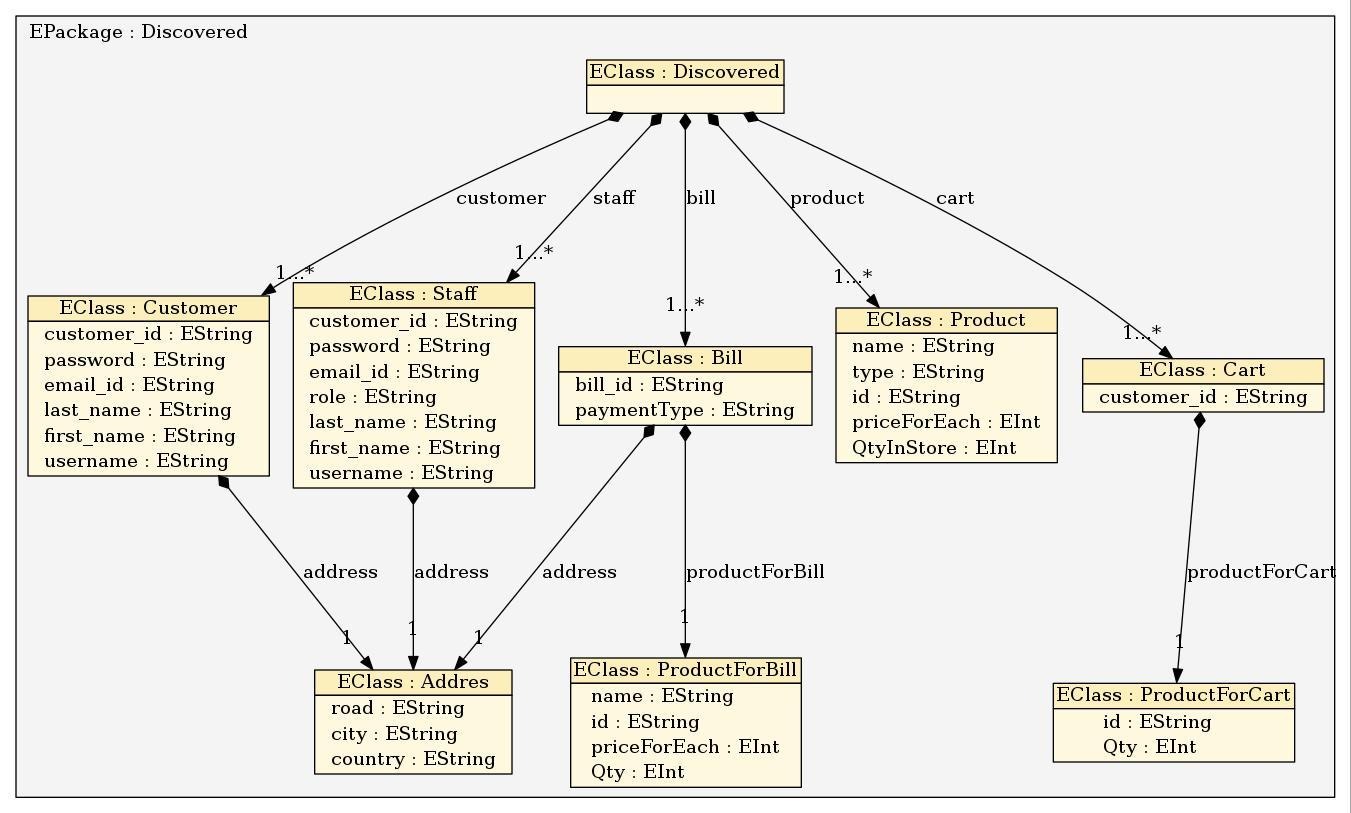


Mobile login



# Detailed Design

Json Schema:



# Data Dictionary

Customer object:

|  |  |  |
| --- | --- | --- |
| **Name** | **Data Type** | **Description** |
| customer\_id | String | The id of a customer |
| password | String | The password of a customer |
| email\_id | String | The email of a customer |
| last\_name | String | The last name of a customer |
| first\_name | String | The first name of a customer |
| username | String | The username of a customer |

Staff object:

|  |  |  |
| --- | --- | --- |
| **Name** | **Data Type** | **Description** |
| customer\_id | String | The customer of a staff |
| password | String | The password of a staff |
| email\_id | String | The email of a staff |
| last\_name | String | The last name of a staff |
| first\_name | String | The first name of a staff |
| username | String | The username of a staff |
| role | String | The role of a staff |

Bill object:

|  |  |  |
| --- | --- | --- |
| **Name** | **Data Type** | **Description** |
| Bill\_id | String | The id of a Bill |
| paymentType | String | The type of payment |

Product object:

|  |  |  |
| --- | --- | --- |
| **Name** | **Data Type** | **Description** |
| name | String | The name of a product |
| type | String | The type of a product |
| id | String | The id of a product |
| priceForEach | String | The price of a product |
| QtyInStore | String | The quantity of a product in store |

Cart object:

|  |  |  |
| --- | --- | --- |
| **Name** | **Data Type** | **Description** |
| customer\_id | String | The id of a Customer |

Addres object:

|  |  |  |
| --- | --- | --- |
| **Name** | **Data Type** | **Description** |
| road | String | The road of address |
| city | String | The city of address |
| country | String | The country of address |

ProductFoBill object:

|  |  |  |
| --- | --- | --- |
| **Name** | **Data Type** | **Description** |
| name | String | The road of address |
| id | String | The city of address |
| priceForEach | Int | The country of address |
| Qty | Int | The quantity of the product |

ProductForCart object:

|  |  |  |
| --- | --- | --- |
| **Name** | **Data Type** | **Description** |
| id | String | The id of the product |
| Qty | Int | The quantity of the product |

# User Interface Design and User Guide

## Chat Apps

|  |  |
| --- | --- |
| Start Page   1. Button 2. Login 3. Register(Testing only) |  |
| Login Page   1. Text view 2. Input Email 3. Input Password 4. Button 5. Login   Click login button to login |  |
| Main Page  Chat record  Here you can find all staff chat record  You can preview latest message and user is or not online. |  |
| Main Page  All Staff User List  Here you can find all staff user    You can type specify user name to short list user list. User name is case sensitive. |  |

|  |  |
| --- | --- |
| Main Page  Chat record  Here you can find all customer chat record |  |
| Main Page  All Customer User List  Here you can find all staff user    You can type specify user name to short list user list. User name is case sensitive. |  |
| Chat room   1. Receiver user name 2. Receiver user profile image   Left:Receiver sended message  Right:Current user sended message  You can see the your lastes sebding messaage is or not seen.   1. Message box 2. Send Button 3. Dial Button 4. Send Photo Button   To send message:  1. Type something in EditText  2. PressButton Press  Make a phone call:  Click  Button  To send Photo:  1. Click Button  2. Select one photo  3. Click send button |  |

Website

f

3

2

4

b

c

|  |  |
| --- | --- |
| Chat UI of website   1. Clickable for opening and closing the chat. 2. Name of the sender. 3. icon of the sender. 4. Message text box. 5. Message Area. 6. Input box. | 1      5  6 |
| Main Page: | |
| Header and Navbar: | |
| Login and Register  Search product  Change to different page | |

|  |  |
| --- | --- |
| Login Form:  Customer can login with his email and password |  |

|  |  |
| --- | --- |
| Register Form:  Customer can register a new account here |  |

|  |
| --- |
|  |
|  |
| Computer building page:  Customer can customize to build a computer |

|  |
| --- |
|  |
| Product Guide page:  Show all product that the company provides.  There has a selector in left side for customer select product type. |

|  |
| --- |
|  |
| Product Page:  Showing the Product information and customer and add it to shopping cart |

|  |
| --- |
|  |
| Contact Page:  Showing the company information |

|  |
| --- |
|  |
| List all products user added to shopping cart, click Proceed to CheckOut button will lead user to payment process page. |

|  |
| --- |
|  |
| User center, user can set account information, check all orders.  Click My Order Button will lead user to order page.  Click Account Setting page will lead user to account setting page. |

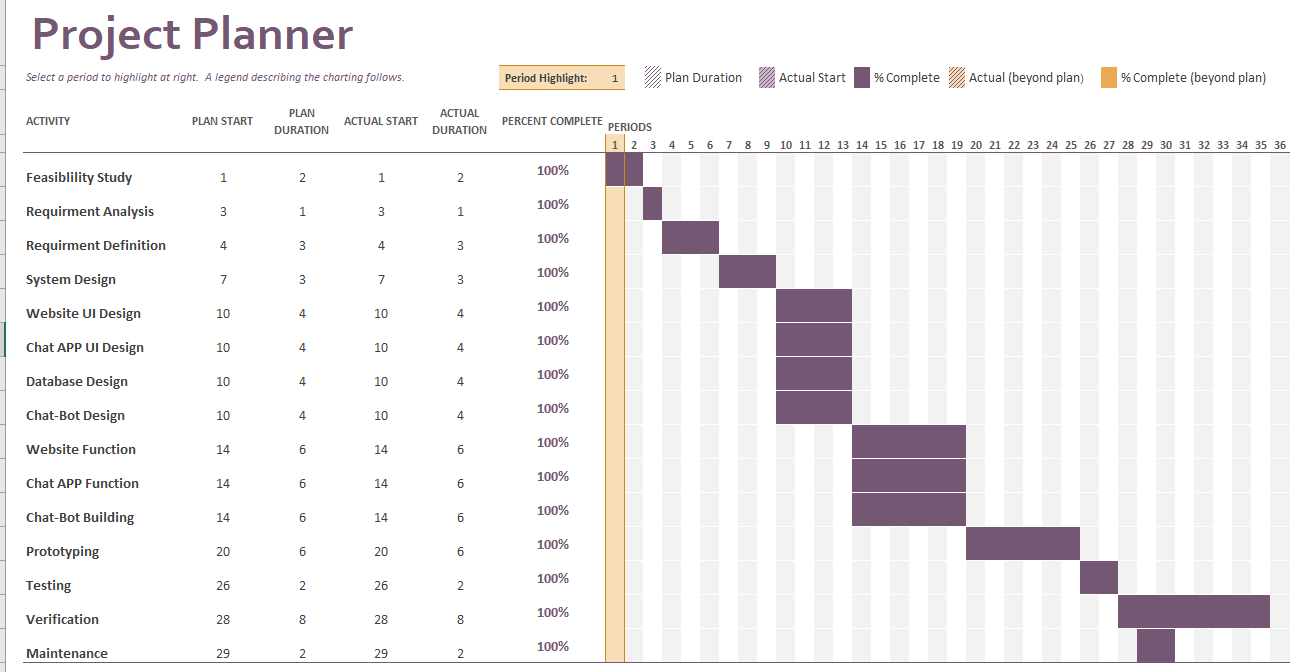
|  |
| --- |
|  |
|  |
| List all order user ordered, and all information in this order.  Click Buy it again button, will change to that product page, user can buy it again.  Click Order Details, all order details will be shown, including delivery address, payment method. |

|  |
| --- |
|  |
| Comment for the product, user can check all comment for this product with username, also, user can leave a comment here buy click Leave a Comment Button after input some think. |

|  |
| --- |
|  |
|  |
| Payment process page,  Select Deliver Address:  User can create a new address for this order deliver, or select the delivery address that user account set. If user select new address, it should be fill it all data, overrise, have a warning to user.  Select Payment Method:  User can select a payment method for this order.  Click CheckOUT button it will have a confirm for user to check the deliver address and payment method. |

|  |
| --- |
|  |
|  |
| Forget password page, for user reset his account’s password, input an email address and click send request could reset password through an email sent by system. |

# Project Plan



**Detail Plan:**

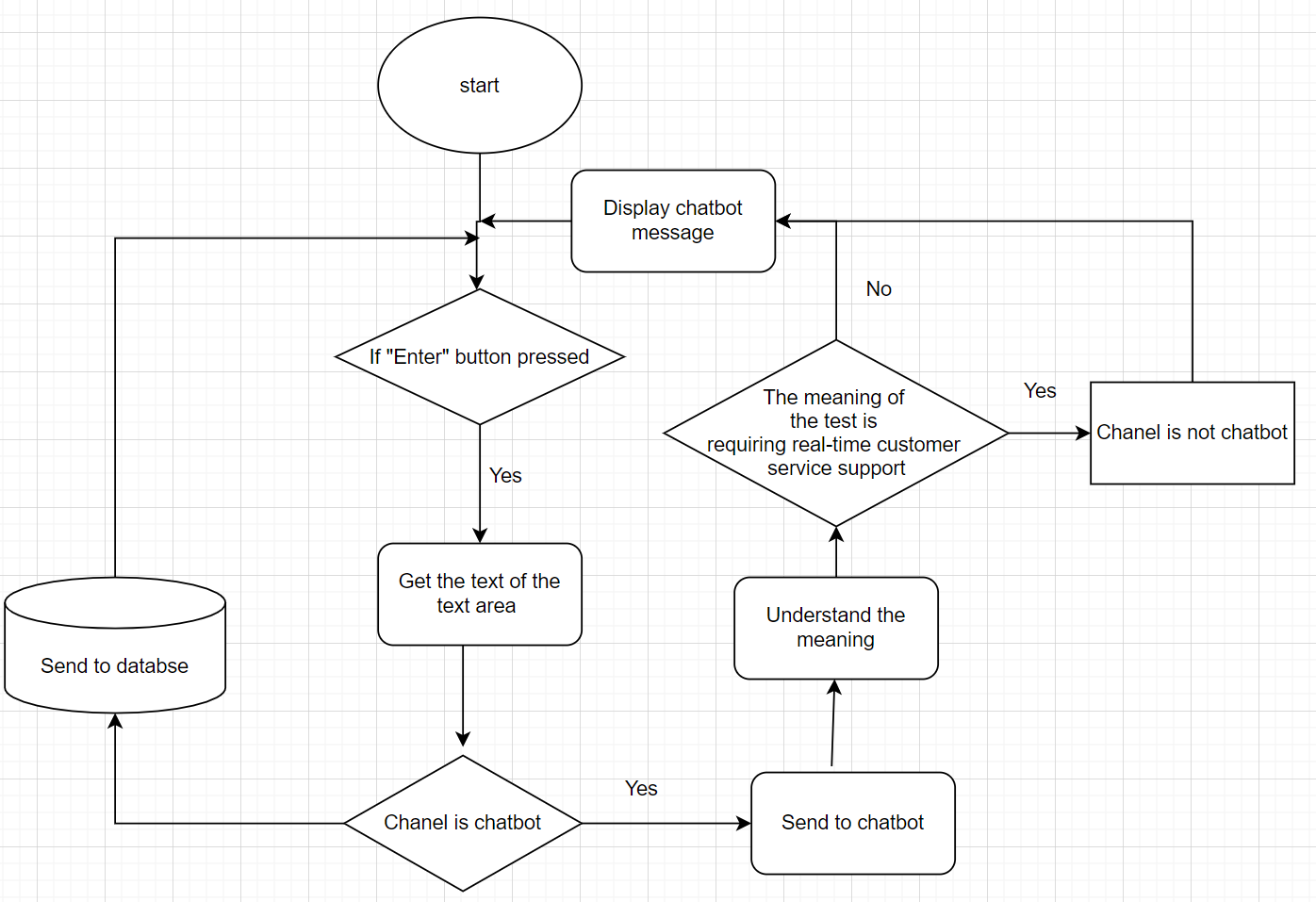
**Report:**

In the actual progress of the project, we have no deviation of the progress from the schedules. We have great project planning and management; we can divide the work clearly and complete it on time so that we could effectively finish our works and no deviation of the progress from the schedules.

# Implementation

## Tase Case

Chat UI

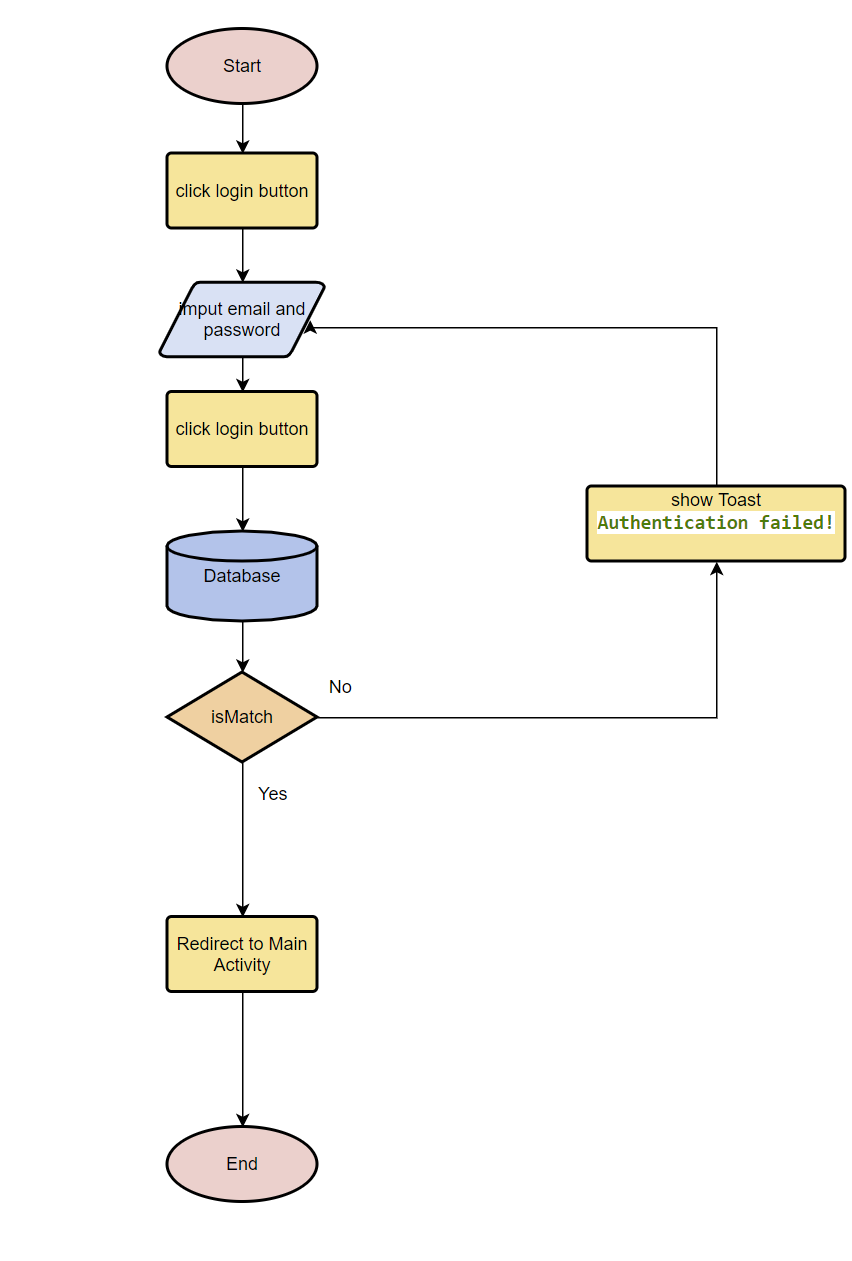


|  |  |
| --- | --- |
| **Test Case ID** | WConnectStaff-01 |
| **Test Case Description** | Change the conversation channel to the real-time customer service |
| **Steps** | 1. On the text area of the chat UI, ask for the connection with real-time customer service 2. Press “Enter” 3. After the confirmation check received, send confirm message |
| **Test Data** | Message: I need real staff  After confirmation check received  Message: Yes |
| **Priority** | High |
| **Pre-Conditions** | / |
| **Expected Result** | Connect to real-time customers |
| **Final Result** | success |
| **Test Date** | 16/4/2021 |
| **Tester** | Lo Yee Kit |

|  |  |
| --- | --- |
| **Test Case ID** | WMessaging-01 |
| **Test Case Description** | Send a message to chatbot |
| **Steps** | 1. On the text area of the chat UI, input message 2. Press “Enter” |
| **Test Data** | Message: hi |
| **Priority** | High |
| **Pre-Conditions** | / |
| **Expected Result** | Receive the proper message from the chatbot |
| **Final Result** | success |
| **Test Date** | 16/4/2021 |
| **Tester** | Lo Yee Kit |

|  |  |
| --- | --- |
| **Test Case ID** | WMessaging-02 |
| **Test Case Description** | Send a message to the customer service |
| **Steps** | 1. On the text area of the chat UI, Input message 2. Press “Enter” |
| **Test Data** | Message: Who are you? |
| **Priority** | High |
| **Pre-Conditions** | WConnectStaff-01 |
| **Expected Result** | Receive proper message from the real-time customer service |
| **Final Result** | success |
| **Test Date** | 16/4/2021 |
| **Tester** | Lo Yee Kit |

### Login Test



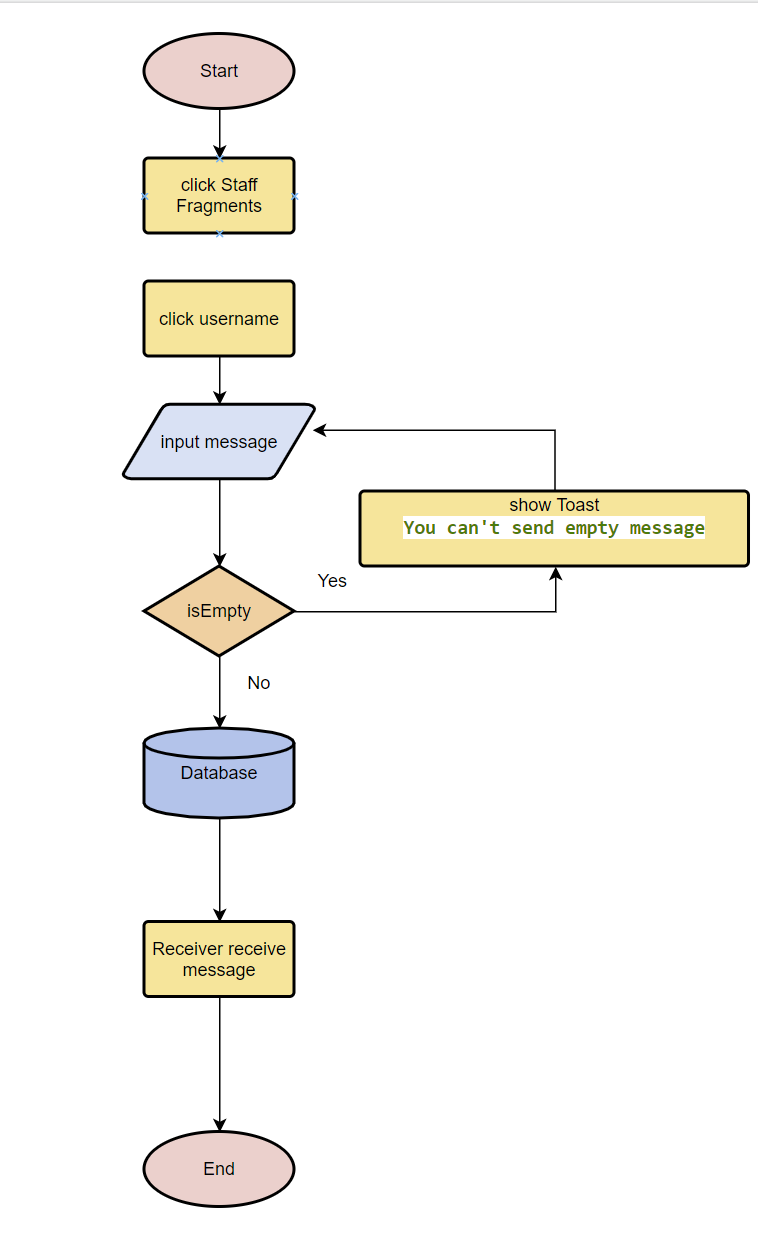
|  |  |
| --- | --- |
| **Test Case ID** | Tappl-01 |
| **Test Case Description** | Login Test |
| **Steps** | 1. Input email and password  2. Press LOGIN Button |
| **Test Data** | Email:[ggggggg@gmail.com](mailto:ggggggg@gmail.com)  Password:99999999999999 |
| **Priority** | High |
| **Pre-Conditions** | / |
| **Expected Result** | Expected Result:Redirect to Main Activity  Actual Result: As expected |
| **Final Result** | success |
| **Test Date** | 1/2/2021 |
| **Tester** | Cheung King Hung |

|  |  |
| --- | --- |
| **Test Case ID** | Tappl-02 |
| **Test Case Description** | Login Test |
| **Steps** | 1. Input email and password  2. Press LOGIN Button |
| **Test Data** | Email: yk@gmail.com  Password: 111222333 |
| **Priority** | High |
| **Pre-Conditions** | / |
| **Expected Result** | Expected Result:Redirect to Main Activity  Actual Result: As expected |
| **Final Result** | success |
| **Test Date** | 1/2/2021 |
| **Tester** | Cheung King Hung |

|  |  |
| --- | --- |
| **Test Case ID** | Tappl-03 |
| **Test Case Description** | Login Test |
| **Steps** | 1. No input anything  2. Press LOGIN Button |
| **Test Data** | / |
| **Priority** | High |
| **Pre-Conditions** | / |
| **Expected Result** | Expected Result: show Toast  All field are required!  Actual Result: As expected |
| **Final Result** | success |
| **Test Date** | 1/2/2021 |
| **Tester** | Cheung King Hung |

|  |  |
| --- | --- |
| **Test Case ID** | Tappl-04 |
| **Test Case Description** | Login Test |
| **Steps** | 1. Input not correct email and password  2. Press LOGIN Button |
| **Test Data** | Email: yk@gmail.com  Password: 111222333\* |
| **Priority** | High |
| **Pre-Conditions** | / |
| **Expected Result** | Expected Result: show Toast  Authentication failed!  Actual Result: As expected |
| **Final Result** | success |
| **Test Date** | 1/2/2021 |
| **Tester** | Cheung King Hung |

### Send Message Test

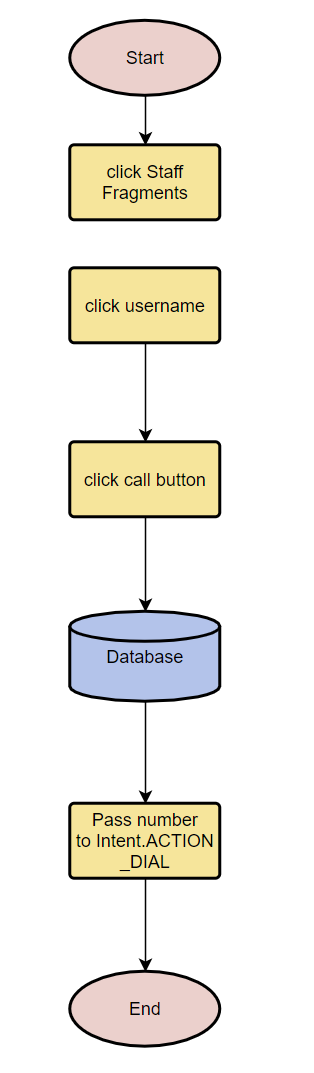


|  |  |
| --- | --- |
| **Test Case ID** | Tappm-01 |
| **Test Case Description** | Send Message Test |
| **Steps** | 1. On Phone A, Click Staff Fragments  2. Click “Yuki” (Phone B user)  3. Click bottom text area  3. input “Hi”  4: click  button |
| **Test Data** | (Phone A)  Email:[ggggggg@gmail.com](mailto:ggggggg@gmail.com)  Password:99999999999999  Text: Hi  (Phone B)  Email: yk@gmail.com  Password: 111222333 |
| **Priority** | High |
| **Pre-Conditions** | Test Case ID : Tappl-01 (Phone A)  Test Case ID : Tappl-02 (Phone B) |
| **Expected Result** | Expected Result: Phone B receive the Notifications”  Actual Result: As expected  Expected Result: Phone B receive the meaasge”Hi”  Actual Result: As expected |
| **Final Result** | success |
| **Test Date** | 2/2/2021 |
| **Tester** | Cheung King Hung |

|  |  |
| --- | --- |
| **Test Case ID** | Tappm-02 |
| **Test Case Description** | Send Message Test |
| **Steps** | 1. On Phone B, Click Staff Fragments  2. Click “Yoyo” (Phone A user)  3. Click bottom text area  3. input “What up”  4: click  button |
| **Test Data** | (Phone A)  Email:[ggggggg@gmail.com](mailto:ggggggg@gmail.com)  Password:99999999999999  (Phone B)  Email: yk@gmail.com  Password: 111222333  Text: What up |
| **Priority** | High |
| **Pre-Conditions** | Test Case ID : Tappl-01 (Phone A)  Test Case ID : Tappl-02 (Phone B) |
| **Expected Result** | Expected Result: Phone A receive the Notifications”  Actual Result: As expected  Expected Result: Phone A receive the meaasge” What up”  Actual Result: As expected |
| **Final Result** | success |
| **Test Date** | 2/2/2021 |
| **Tester** | Cheung King Hung |

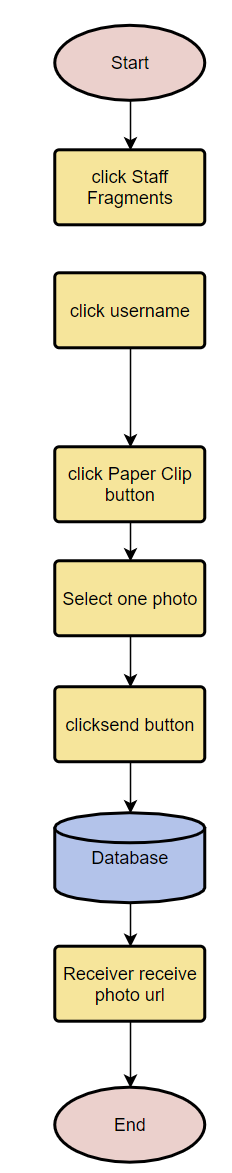
|  |  |
| --- | --- |
| **Test Case ID** | Tappm-03 |
| **Test Case Description** | Send Message Test |
| **Steps** | 1. On Phone B, Click Staff Fragments  2. Click “Yoyo” (Phone A user)  3. Click bottom text area  4: click  button |
| **Test Data** | (Phone A)  Email:[ggggggg@gmail.com](mailto:ggggggg@gmail.com)  Password:99999999999999  (Phone B)  Email: yk@gmail.com  Password: 111222333  Text: (empty) |
| **Priority** | High |
| **Pre-Conditions** | Test Case ID : Tappl-01 (Phone A)  Test Case ID : Tappl-02 (Phone B) |
| **Expected Result** | Expected Result: show Toast:  You can't send empty message!  Authentication failed!Actual Result: As expected |
| **Final Result** | success |
| **Test Date** | 2/2/2021 |
| **Tester** | Cheung King Hung |

### Phone Call Test



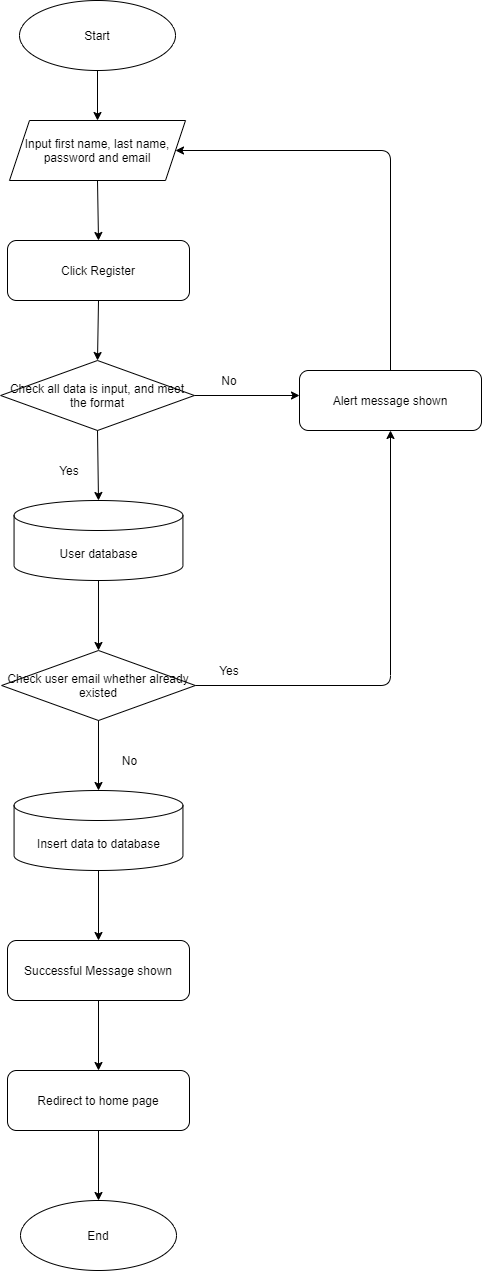
|  |  |
| --- | --- |
| **Test Case ID** | Tappc-01 |
| **Test Case Description** | Phone Call Test |
| **Steps** | 1. On Phone A, Click Staff Fragments  2. Click “Simon1sKing” (Phone B user)  4: click  button  5. Click  Button |
| **Test Data** | (Phone A)  Email:[ggggggg@gmail.com](mailto:ggggggg@gmail.com)  Password:99999999999999  Insert a sim card number 69332871  (Phone B)  Insert a sim card number 54342133 |
| **Priority** | High |
| **Pre-Conditions** | Phone B:  Test Case ID : Tappl-01 (Phone A)  Phone B:  Insert a sim card number 54342133 |
| **Expected Result** | Expected Result: Phone B receive call from Phone A  Phone A will display a phone call from 69332871”  Actual Result: As expected |
| **Final Result** | success |
| **Test Date** | 10/4/2021 |
| **Tester** | Cheung King Hung |

### Send Photo Test



|  |  |
| --- | --- |
| **Test Case ID** | Tappp-01 |
| **Test Case Description** | Send Photo Test |
| **Steps** | 1. On Phone A, Click Staff Fragments  2. Click “Yuki” (Phone B user)  3. Click Button  4. Select one photo  5. Click send button |
| **Test Data** | (Phone A)  Email:[ggggggg@gmail.com](mailto:ggggggg@gmail.com)  Password:99999999999999  (Phone B)  Email: yk@gmail.com  Password: 111222333 |
| **Priority** | High |
| **Pre-Conditions** | Test Case ID : Tappl-01 (Phone A)  Test Case ID : Tappl-02 (Phone B) |
| **Expected Result** | Expected Result: Phone B receive the Notifications”  Actual Result: As expected  Expected Result: Phone B receive the photo url  Actual Result: As expected |
| **Final Result** | success |
| **Test Date** | 16/4/2021 |
| **Tester** | Cheung King Hung |

Registration Testing



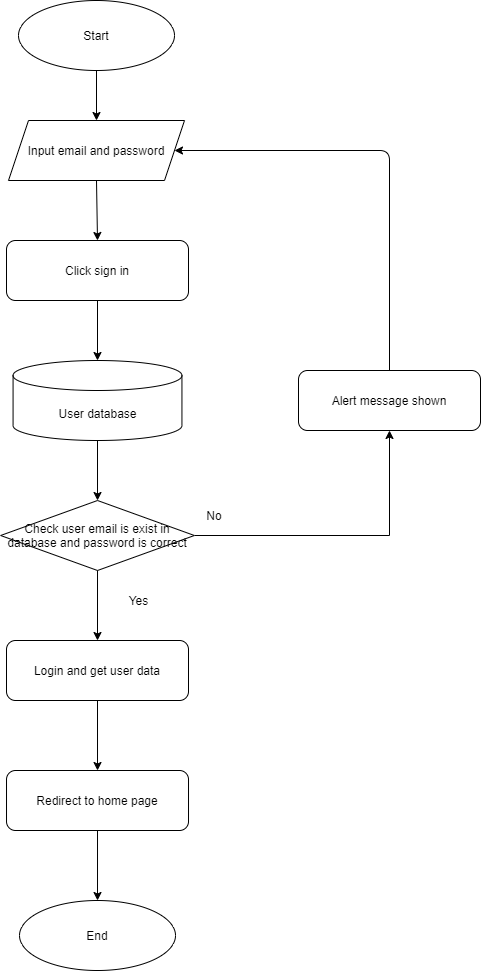
|  |  |
| --- | --- |
| **Test ID:** | RT-001 |
| **Description:** | Customer's registration |
| **Tester:** | Ng Ka Ho |
| **Tested date:** | 1/04/2021 |
| **Test data:** | First Name: Peter  Last Name: Wong  Password: 123456  Email: test1111@gmail.com |
| **Title:** | Verify customer account can be created as expect. |
| **Test Steps:** | 1. Click Register in the home page. 2. Enter test data as “First Name”, “Last Name”, “Password”, “Email”. 3. Click Register |
| **Results:** | Expected Result: Register form can be display.  Actual Result: As expected  Expected Result: Data can input successful.  Actual Result: As expected  Expected Result: Successful message is shown.  Actual Result: As expected  Expected Result: Redirect to home page.  Actual Result: As expected |

|  |  |
| --- | --- |
| **Test ID:** | RT-002 |
| **Description:** | Customer's registration |
| **Tester:** | Ng Ka Ho |
| **Tested date:** | 1/04/2021 |
| **Test data:** | First Name: Peter  Last Name: Wong  Password: 12345  Email: test |
| **Title:** | Verify customer account cannot be created as expect. |
| **Test Steps:** | 1. Click Register in the home page. 2. Enter test data as “First Name”, “Last Name”, “Password”, “Email”. 3. Click Register |
| **Results:** | Expected Result: Register form can be display.  Actual Result: As expected  Expected Result: Data can input successful.  Actual Result: As expected  Expected Result: Wrong format message is shown.  Actual Result: As expected |

|  |  |
| --- | --- |
| **Test ID:** | RT-003 |
| **Description:** | Customer's registration |
| **Tester:** | Ng Ka Ho |
| **Tested date:** | 1/04/2021 |
| **Test data:** | First Name: Peter  Last Name: Wong  Password: 123456  Email: peterwong@gmail.com |
| **Title:** | Verify customer account cannot be created as expect. |
| **Test Steps:** | 1. Click Register in the home page. 2. Enter test data as “First Name”, “Last Name”, “Password”, “Email”. 3. Click Register |
| **Results:** | Expected Result: Register form can be display.  Actual Result: As expected  Expected Result: Data can input successful.  Actual Result: As expected  Expected Result: Email already exist message is shown.  Actual Result: As expected |

|  |  |
| --- | --- |
| **Test ID:** | RT-004 |
| **Description:** | Customer's registration |
| **Tester:** | Ng Ka Ho |
| **Tested date:** | 2/04/2021 |
| **Test data:** | First Name: " "  Last Name: " "  Password: " "  Email: peterwong@gmail.com |
| **Title:** | Verify customer account cannot be created as expect. |
| **Test Steps:** | 1. Click Register in the home page. 2. Enter test data as “First Name”, “Last Name”, “Password”, “Email”. 3. Click Register |
| **Results:** | Expected Result: Register form can be display.  Actual Result: As expected  Expected Result: Data can input successful.  Actual Result: As expected  Expected Result: Please fill in all information message is shown.  Actual Result: As expected |

Login Testing



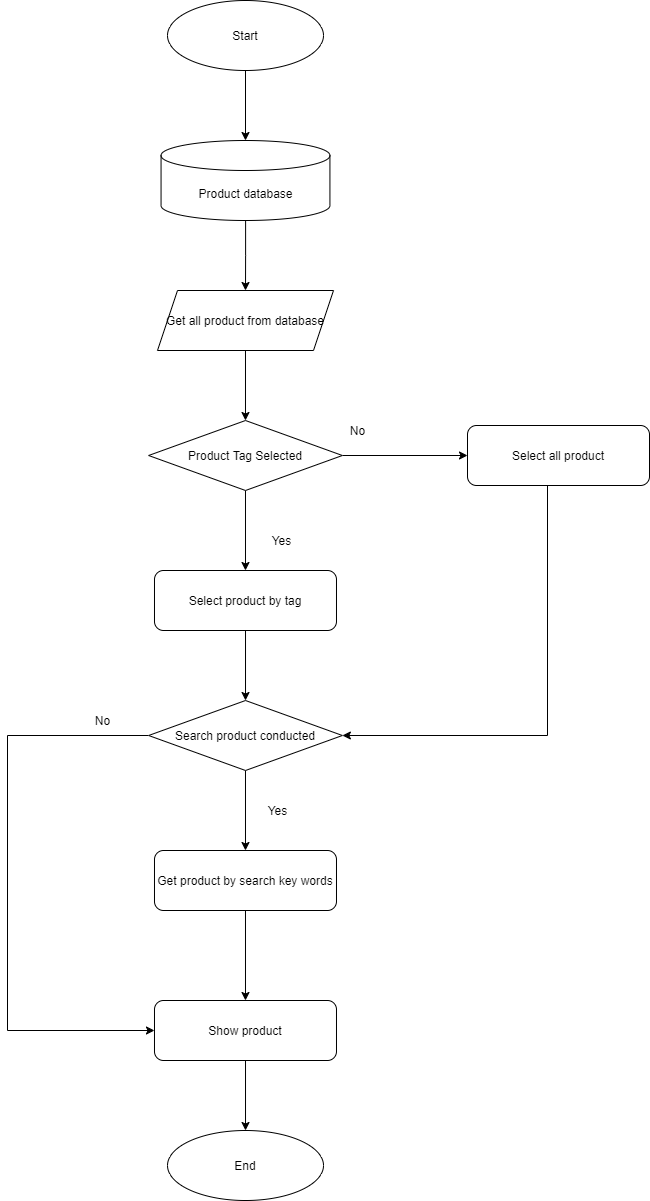
|  |  |
| --- | --- |
| **Test ID:** | LT-001 |
| **Description:** | Customer's login |
| **Tester:** | Ng Ka Ho |
| **Tested date:** | 1/04/2021 |
| **Test data:** | Email: peko5914@gmail.com  Password: 123456 |
| **Title:** | Verify customer can be login as expect. |
| **Test Steps:** | 1. Click Login in the home page. 2. Input email and password field with test data. 3. Click sign in button. |
| **Results:** | Expected Result: Login form can be display.  Actual Result: As expected  Expected Result: Data can input successful.  Actual Result: As expected  Expected Result: Successful to login.  Actual Result: As expected  Expected Result: Redirect to home page.  Actual Result: As expected |

|  |  |
| --- | --- |
| **Test ID:** | LT-002 |
| **Description:** | Customer's login |
| **Tester:** | Ng Ka Ho |
| **Tested date:** | 1/04/2021 |
| **Test data:** | Email: test@gmail.com  Password: 123456 |
| **Title:** | Verify customer cannot be login because of email did not exist as expect. |
| **Test Steps:** | 1. Click Login in the home page. 2. Input email and password field with test data. 3. Click sign in button. |
| **Results:** | Expected Result: Login form can be display.  Actual Result: As expected  Expected Result: Data can input successful.  Actual Result: As expected  Expected Result: Email didn’t exist message is shown.  Actual Result: As expected  Expected Result: Redirect to Login form.  Actual Result: As expected |

|  |  |
| --- | --- |
| **Test ID:** | LT-003 |
| **Description:** | Customer's login |
| **Tester:** | Ng Ka Ho |
| **Tested date:** | 1/04/2021 |
| **Test data:** | Email: peko5914@gmail.com  Password: \Abcdefg!$ |
| **Title:** | Verify customer cannot be login because password is wrong as expect. |
| **Test Steps:** | 1. Click Login in the home page. 2. Input email and password field with test data. 3. Click sign in button. |
| **Results:** | Expected Result: Login form can be display.  Actual Result: As expected  Expected Result: Data can input successful.  Actual Result: As expected  Expected Result: Wrong password or email message is shown.  Actual Result: As expected  Expected Result: Redirect to Login form.  Actual Result: As expected |

|  |  |
| --- | --- |
| **Test ID:** | LT-004 |
| **Description:** | Customer's login |
| **Tester:** | Ng Ka Ho |
| **Tested date:** | 1/04/2021 |
| **Test data:** | Email:  Password: |
| **Title:** | Verify customer cannot be login because password is wrong as expect. |
| **Test Steps:** | 1. Click Login in the home page. 2. Input email and password field with test data. 3. Click sign in button. |
| **Results:** | Expected Result: Login form can be display.  Actual Result: As expected  Expected Result: Data can input successful.  Actual Result: As expected  Expected Result: Please fill in all information message is shown.  Actual Result: As expected  Expected Result: Redirect to Login form.  Actual Result: As expected |

Product Guide Testing



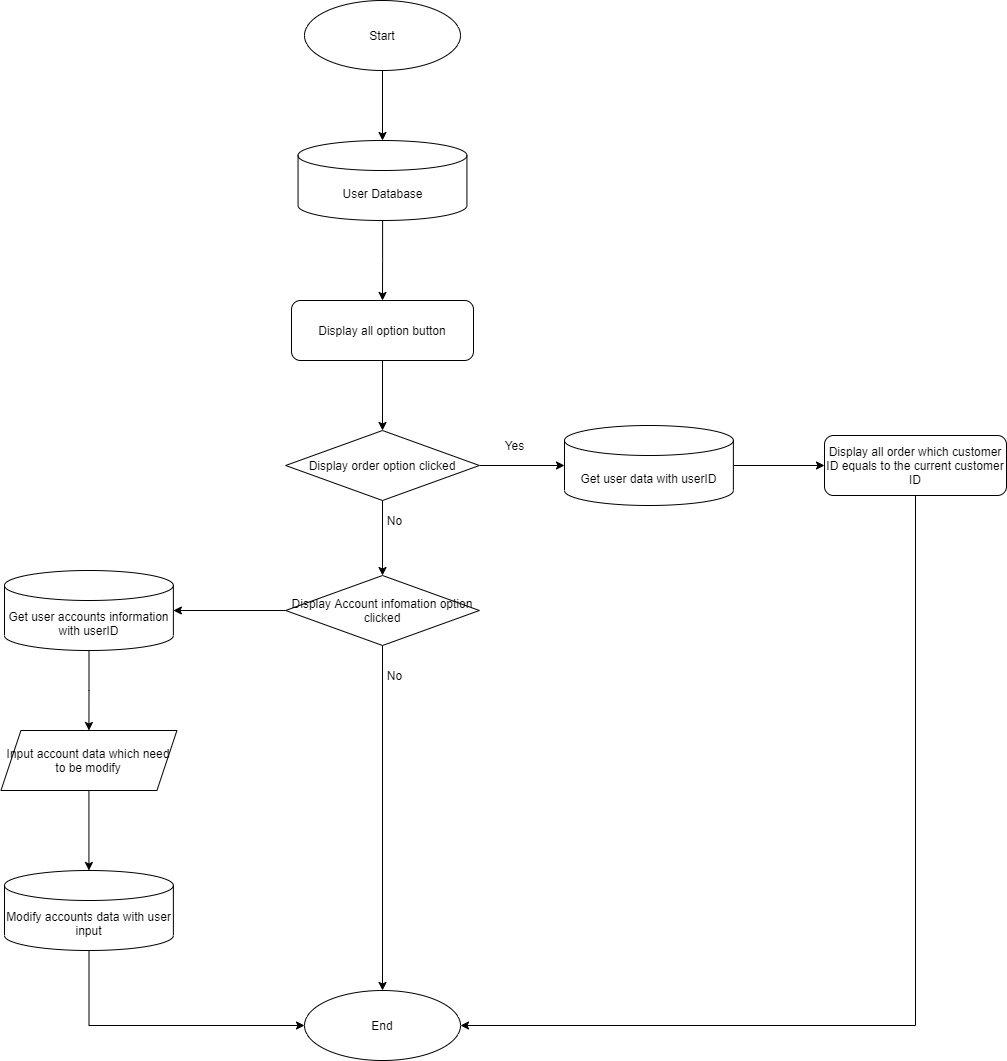
|  |  |
| --- | --- |
| **Test ID:** | PG-001 |
| **Description:** | Display products |
| **Tester:** | Ng Ka Ho |
| **Tested date:** | 1/04/2021 |
| **Test data:** | No selected Tag  No search words. |
| **Title:** | Verify products can be display as test data. |
| **Test Steps:** | 1. Go to Product guide page. 2. All Product is shown. |
| **Results:** | Expected Result: Product guide page can be access.  Actual Result: As expected  Expected Result: All products will be display.  Actual Result: As expected |

|  |  |
| --- | --- |
| **Test ID:** | PG-002 |
| **Description:** | Display GPU products |
| **Tester:** | Ng Ka Ho |
| **Tested date:** | 1/04/2021 |
| **Test data:** | GPU Tag selected  No search words. |
| **Title:** | Verify all GPU products can be display as test data. |
| **Test Steps:** | 1. Go to Product guide page. 2. Select GPU tag. 3. All GPU products displayed. |
| **Results:** | Expected Result: Product guide page can be access.  Actual Result: As expected  Expected Result: All GPU products will be display.  Actual Result: As expected |

|  |  |
| --- | --- |
| **Test ID:** | PG-003 |
| **Description:** | Display GPU products with search words |
| **Tester:** | Ng Ka Ho |
| **Tested date:** | 1/04/2021 |
| **Test data:** | GPU Tag selected  '3060’ key words inputted. |
| **Title:** | Verify all GPU products which name have ‘3060’ can be display. |
| **Test Steps:** | 1. Go to Product guide page. 2. Select GPU tag. 3. Search product with key words ‘3060’ 4. All GPU products which name have ‘3060’ displayed. |
| **Results:** | Expected Result: Product guide page can be access.  Actual Result: As expected  Expected Result: All GPU products with name ‘3060’ will be display.  Actual Result: As expected |

|  |  |
| --- | --- |
| **Test ID:** | PG-004 |
| **Description:** | Display all products with search words. |
| **Tester:** | Ng Ka Ho |
| **Tested date:** | 1/04/2021 |
| **Test data:** | No Tag selected  'ASUS’ key words inputted. |
| **Title:** | Verify all products which name have ‘ASUS’ can be display. |
| **Test Steps:** | 1. Go to Product guide page. 2. Search product with key words ‘3060’ 3. All products which name have ‘ASUS’ displayed. |
| **Results:** | Expected Result: Product guide page can be access.  Actual Result: As expected  Expected Result: All products with name ‘ASUS’ will be display.  Actual Result: As expected |

User Center Testing

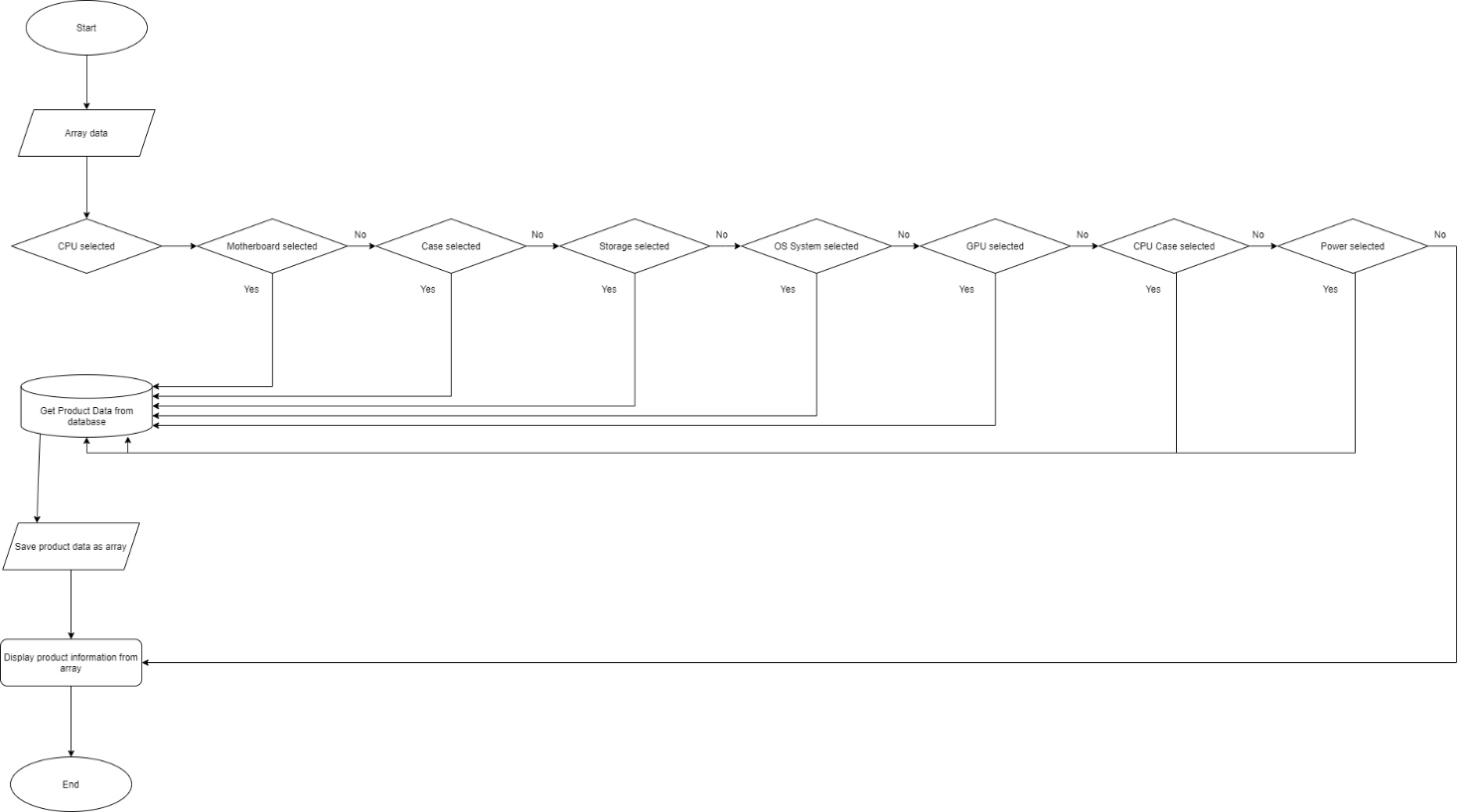


|  |  |
| --- | --- |
| **Test ID:** | US-001 |
| **Description:** | Display orders |
| **Tester:** | Ng Ka Ho |
| **Tested date:** | 1/04/2021 |
| **Test data:** | CustomerID: 8Vkr8r0Dj0bcDTeg4jHAxrNyo0l2 |
| **Title:** | Verify order can be get by customer ID and display it. |
| **Test Steps:** | 1. Go to User center page. 2. Select Order option. 3. All order displayed. |
| **Results:** | Expected Result: Order option could be operation.  Actual Result: As expected  Expected Result: All orders displayed.  Actual Result: As expected |

|  |  |
| --- | --- |
| **Test ID:** | US-002 |
| **Description:** | Display user account information |
| **Tester:** | Ng Ka Ho |
| **Tested date:** | 1/04/2021 |
| **Test data:** | CustomerID: 8Vkr8r0Dj0bcDTeg4jHAxrNyo0l2 |
| **Title:** | Verify Accounts information can be display. |
| **Test Steps:** | 1. Go to User center page. 2. Select Account information option. 3. All account information displayed. |
| **Results:** | Expected Result: Account information option could be operation.  Actual Result: As expected  Expected Result: All Account information displayed.  Actual Result: As expected |

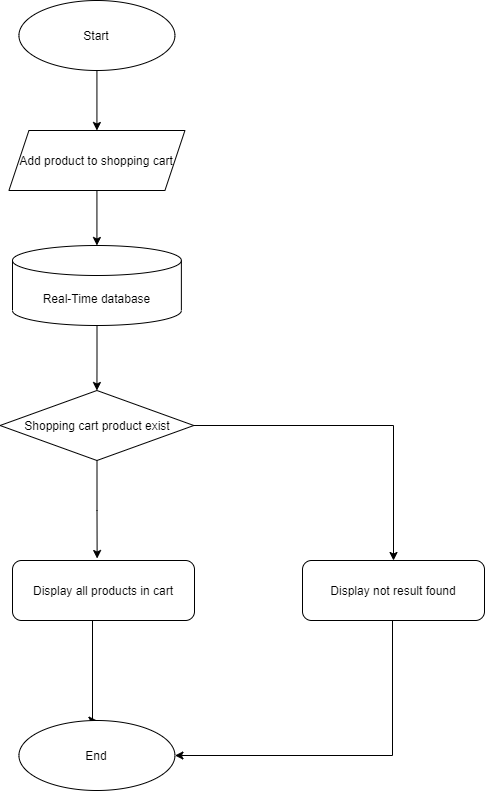
|  |  |
| --- | --- |
| **Test ID:** | US-003 |
| **Description:** | Modify user account information. |
| **Tester:** | Ng Ka Ho |
| **Tested date:** | 1/04/2021 |
| **Test data:** | CustomerID: 8Vkr8r0Dj0bcDTeg4jHAxrNyo0l2  Email: testMod@gmail.com  Password: pa$$w0rd  First Name: Test  Last Name: 123 |
| **Title:** | Verify Accounts information can be modifying. |
| **Test Steps:** | 1. Go to User center page. 2. Select Account information option. 3. Input new account information as test data and click Save change. 4. Redirect to Account information |
| **Results:** | Expected Result: Account information option could be operation.  Actual Result: As expected  Expected Result: All Account information displayed.  Actual Result: As expected |

Custom computer Testing



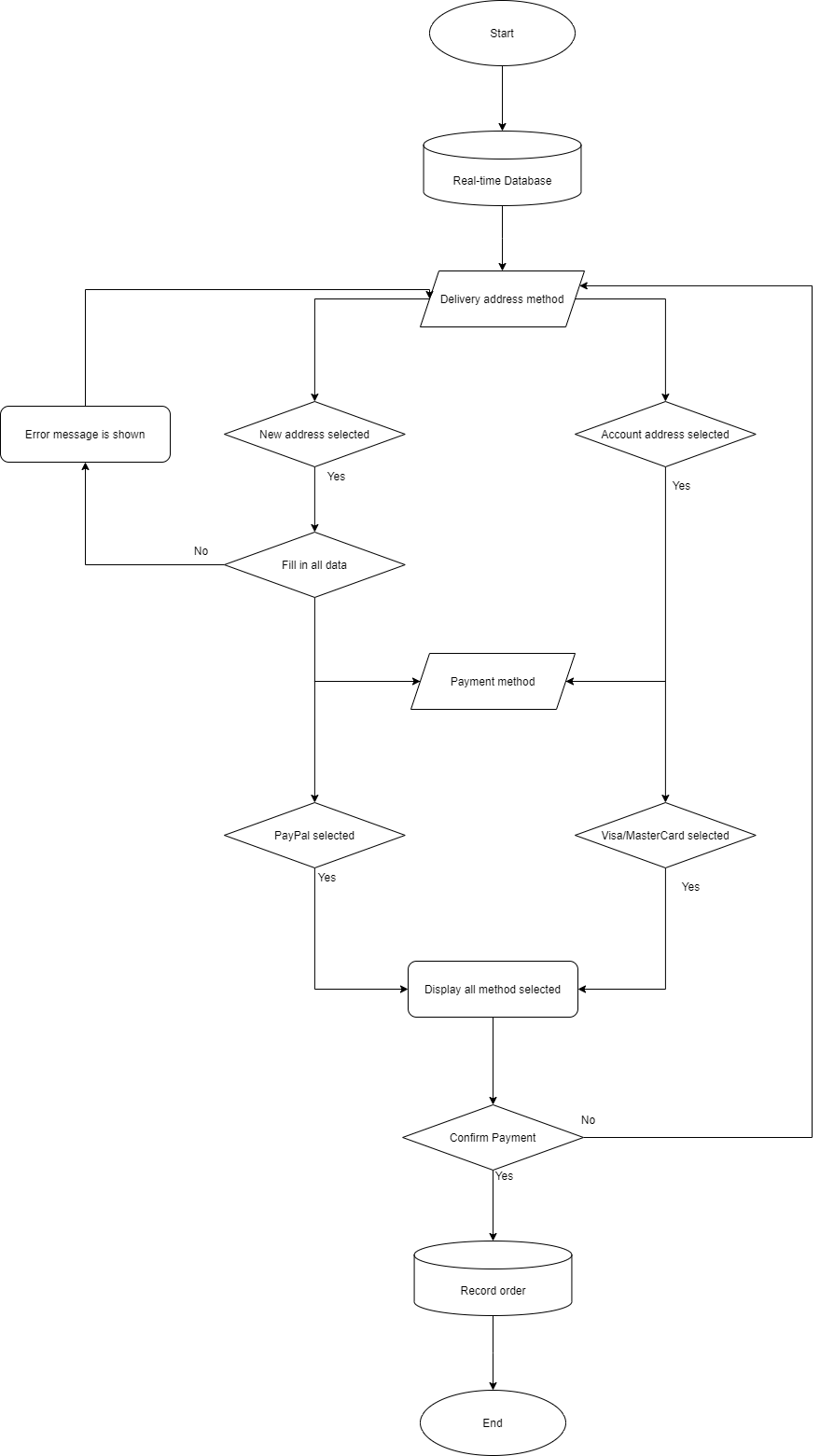
|  |  |
| --- | --- |
| **Test ID:** | CC-001 |
| **Description:** | Complete a custom computer |
| **Tester:** | Ng Ka Ho |
| **Tested date:** | 1/04/2021 |
| **Test data:** | CPU: I7-10700K  Motherboard: ASUS  GPU: GeForce RTX 3090  RAM: 16GB:  Case: Cooler Master MasterFrame 700  OS System: Windows 10 |
| **Title:** | Verify all computer component selected will be display |
| **Test Steps:** | 1. Go to custom computer page 2. Click CPU and select I7-10700K as Test data 3. Click GPU and select GeForce RTX 3090 4. Click Motherboard and select ASUS 5. Click RAM and select 16GB 6. Click Case and select Cooler Master MasterFrame 700 7. Click OS System and select Windows 10 |
| **Results:** | Expected Result: All product can be select  Actual Result: As expected  Expected Result: All product selected will be display.  Actual Result: As expected |

Shopping Cart Testing



|  |  |
| --- | --- |
| **Test ID:** | SC-001 |
| **Description:** | Add a product to shopping cart |
| **Tester:** | Ng Ka Ho |
| **Tested date:** | 1/04/2021 |
| **Test data:** | CPU: I7-10700K  Customer ID: W5amHBBFILMLQKEvBKRfToEq8On1 |
| **Title:** | Verify product can be add to cart |
| **Test Steps:** | 1. Go to product Page 2. Select product as test data and click add to cart 3. Go to Shopping cart page |
| **Results:** | Expected Result: Product can be adding to cart  Actual Result: As expected  Expected Result: All Product added to cart can be display  Actual Result: As expected |

Payment Processing Testing



|  |  |
| --- | --- |
| **Test ID:** | PP-001 |
| **Description:** | Finish a payment process |
| **Tester:** | Ng Ka Ho |
| **Tested date:** | 1/04/2021 |
| **Test data:** | Delivery address method: new address  New address setting:  Country: Hong Kong  City: Hong Kong  Road: Tsing Tin Road  Payment method: PayPal |
| **Title:** | Verify Payment process can be finish as test data |
| **Test Steps:** | 1. Go to shopping cart page and click check out 2. Select New address 3. Input address as test data 4. Select PayPal method 5. Click Payment 6. Click confirm |
| **Results:** | Expected Result: New address can be select and setting  Actual Result: As expected  Expected Result: PayPal payment can be select  Actual Result: As expected  Expected Result: Confirm message is shown  Actual Result: As expected  Expected Result: Order recorded with test data selected  Actual Result: As expected |

|  |  |
| --- | --- |
| **Test ID:** | PP-002 |
| **Description:** | Payment process debug |
| **Tester:** | Ng Ka Ho |
| **Tested date:** | 1/04/2021 |
| **Test data:** | Delivery address method: new address  New address setting:  Country: ‘’  City: ‘’  Road: ‘’  Payment method: PayPal |
| **Title:** | Verify Payment process can detect error in address setting |
| **Test Steps:** | 1. Go to shopping cart page and click check out 2. Select New address 3. No input any think in address as test data 4. Select PayPal method 5. Click Payment 6. Click confirm |
| **Results:** | Expected Result: New address can be select and setting  Actual Result: As expected  Expected Result: PayPal payment can be select  Actual Result: As expected  Expected Result: Please fill in all information message is shown  Actual Result: As expected |

# Results and Conclusions

## Summary of the Results

The help from chatbots for customers or computer shops is significant. The combination of real-time customer service and a chatbot system can be used to considerably decrease the time taken by customers and computer shop.

Stores can redistribute the expenditure to other services rather than the customer service part. Meanwhile, the real-time chatting part of the system ensures that all problems from customers can be solved as possible even though the chatbot system is not helpful for profound problems. Customers who are lacking in knowledge of a computer can easily select the products by themself with the aid of the chatbot. As a result, customers can buy the proper products according to their actual needs.

However, there are some difficulties with the FAQ database. The company may need to spend expenditure to create the FAQ database. The Chatbot responds to users’ questions by the FAQ database. If the number of FAQs is not high enough, the chatbot cannot function for most circumstances.

## Conclusion

After the whole phase, the system was built up. The chat system is well functioning on the website. It is capable of all websites because the chat UI which can provide the chatbot function and the real-time chat function to the website is developed as an API. This API can install into all kinds of the system as long as the proper UI and certification were well configured. For demonstration, a demo website is developed.

Phone application developed for the conversation of “Customers to staff” and “staff to staff” is also completed. The phone application is connected to the database server and it is capable to send messages via the database.

## Problems Encounter

### Technical Difficulties

#### Sending Notification

Sending Notification function depend on Firebase Cloud Messaging(FCM), using FCM able staff to be notified when the user closes the chat, application. However, at present, only messages can be sent between apps and apps are mutually able to receive notifications. In addition, during the prediction test, although the mobile apps can receive the message correctly, sometimes the notification dose not pop, sometimes it takes a long time to show up. It is currently expected FCM is unreliable due to far server location.

#### Database connection of the Chat UI

The connection between Chat UI and the database is not stable. As a result, the data that the chatbot required for redirection may not be able to execute or return no response.

#### Sending Photo

Since sending photo is a new feature suggested during the implementation phase, in the early develop parse we did not think of any kind of message than sending text, so we needed to modify a lot of code to support the display of images on websites and mobile applications. Currently, only send the photo to the cloud and callback to send photo URL is supported.

#### isOnline function

The chat apps can update current user status to offline when the application is closing. And also update current user status to online when the application is resume.

But, As the web browser cannot sent the data to firebase during closing, so the chat apps user only check other staff users are online.

## Limitations of the system

The current system is slow in terms of the connection. The mean is the system communication between the system and database. Since we are using the free version of firebase, server can only be selected in North American, so when user side need to communicate with firebase backend side, it takes a while to wait.

Also, we are using free plan of Azure webapps service to host our website, current the server hardware and network speed is affordable and slow. Thus, the server’s tasks processing is slow.

Moreover, since we are only three groupmates in our group, we can only focus on developing chatbot and mobile chat application, so we cannot allocate enough manpower to improve the functions of the online store.

Currently we have no relevant experience to open an online store, and we are not familiar with relevant laws. If we just setup a payment gateway and pay to test, it may violate relevant tax laws.

## Future Extension

**Development of the system**

Generalize our e-commerce system on social media, such as Facebook, Twitter, increase brand awareness. Launch new services/functions for customers, increase the attractiveness of our system.

Promote:

Create posts on social media to promote the e-commerce system, description the benefits of using our system, it can be about the Chat bot application.

Future services / functions:

Cooperation with computer product business, provide new and high-quality products for customers, and increase the products diversity. The customer could also be a seller, to selling his own computer products.

Chat apps:

Users can create a group. Message forward function. The apps can display multimedia. Enhance multimedia transmission function, such as video transmission. Increase security features, such as auto destroy message to protect our customer privacy.

Chat UI:

The company should inset more cases for different circumstances. After more FAQs are inserted, the logic of the chatbot may be chaotic. Therefore, the Intend of Dialogflow should be recategorized. The UI should also be upgraded for the degree of user-friendly.

# Reference

Benefits Of Omnichannel Selling

<https://www.ecomdash.com/advantages-omnichannel-selling/>

Want a Better PC? Try Building Your Own

<https://www.wired.com/story/how-to-build-a-pc/>

Jumbo Computer Supplies

<https://shoponjc.com/quote.php>

Price.com

<https://www.price.com.hk/>

CPU socket from WIKIPEDIA

<https://en.wikipedia.org/wiki/CPU_socket>

Online Customer Service

<https://www.paldesk.com/what-is-online-customer-service/>

Chatbot vs Live Chat: The Pros & Cons for Your Business

<https://www.revechat.com/blog/chatbot-vs-live-chat/>

8. Chatbots In Customer Service – Statistics and Trends [Infographic]

<https://www.invespcro.com/blog/chatbots-customer-service/>

# Appendices

## Logbook

Log sheets are inside the attached zip file:

20-21\_HDSE\_FYP\_Group3\_logbook.zip

1. Benefits Of Omnichannel Selling

   <https://www.ecomdash.com/advantages-omnichannel-selling/> [↑](#footnote-ref-2)
2. Want a Better PC? Try Building Your Own

   <https://www.wired.com/story/how-to-build-a-pc/> [↑](#footnote-ref-3)
3. Jumbo Computer Supplies

   <https://shoponjc.com/quote.php> [↑](#footnote-ref-4)
4. Price.com

   <https://www.price.com.hk/> [↑](#footnote-ref-5)
5. CPU socket from WIKIPEDIA

   <https://en.wikipedia.org/wiki/CPU_socket> [↑](#footnote-ref-6)
6. eCommerce Design Facts & Statistics of 2018

   <https://www.thinkbigonline.com/ecommerce-design-facts-statistics-of-2018/> [↑](#footnote-ref-7)
7. Online Customer Service

   <https://www.paldesk.com/what-is-online-customer-service/> [↑](#footnote-ref-8)
8. Chatbot vs Live Chat: The Pros & Cons for Your Business

   <https://www.revechat.com/blog/chatbot-vs-live-chat/> [↑](#footnote-ref-9)
9. Chatbots In Customer Service – Statistics and Trends [Infographic]

   <https://www.invespcro.com/blog/chatbots-customer-service/> [↑](#footnote-ref-10)