

AE2HCI

INDIVIDUAL HCI PROJECT REPORT

November 10, 2015

Name: Yichen PAN

Student ID: 6514897

University of Nottingham, Ningbo, China

School of Computer Science

1 INTRODUCTION

This report mainly talks about the process of the design activities conducted individually. The main goal of the project is to design and develop a smart phone app that enables users to delivery any items from any location to any destination. The users consist of two groups of people: senders who ask for deliery request and delivers who help deliver objects.

Basically, after the group gave out the questionnaire, conducted the interviews with users and gathered and collected requirements from users, individual design activities begin. There are mainly two types of design: conceptual design and concrete design. At the beginning of the conceptual design, what the APP will do is decided based on the elicited functional requirements. Then, by communicating with users, the conceptual model is captured and how to interact with the app is understood. During this period, task analysis is adopted to analyze the underlying rationale and purpose. Hierarchical Task Analysis(HTA) diagrams can be found in Appendix A. Some issues such as the relation among functions and interface metaphors are also considered so that the concept can be expanded into prototypes. In this stage, achieving empathy with users and efficiently communication is really essential. After the conceptual design is finished, low-fidelity and mid-fidelity prototypes are designed based on what have been understood. During the concrete design, users are involved through out the whole design process. They are actually co-designers and cooperative prototyping and analysis is conducted together with them. In this case, due to immediate feedback and instant modification, there are no two distict iterations, instead, a gradual evolution as illustrated below.

2 BACKGROUND

Some key concepts are adopted during the design stage: interaction design, user experience, cognitive perspective, conceptual model and design, and participatory design. First of all, Rogers, Sharp and Preece(2014) points out that interaction design focuses on designing engaging interfaces with well thought out behaviors. Understanding how users and technology communicate with each other is fundamental to this field so that user-centered is always the first considered principle of this project. The notion of the user experience is central to interaction design. There are many aspects of the user experience taken into consideration during the design: usability, efficiency, utility, learnability and memorability (Rogers, Sharp and Preece, 2014). A number of design principles are also considered: visibility, feedback, constraints, consistency and affordance. Especially, according to Norman(1998), affordance

means 'to give a clue', which is an attribute of the object that allows people to know how to use it. As a result, maintaining the app at an easy-to-learn level is an essential concern. Moreover, cognitive perspective is strongly related to these design principles because it helps users understand by examining humans' abilities and limitations. They also give many implications to the design, such as adopting metaphors(i.e. icons) to rely on recognition rather than recall due to the limitation of short memory of humans(Miller's Theory ,1956). However, in the practice, some trade-offs occur when applying different design principles. For example, remaining consistent may result in inconsistency with something else. Finally, the ideas of participatory design is adhered to strictly when prototyping.

3 THE DESIGN RATIONALE

3.1 Requirements Gathering and Analysis

When working individually, some new requirements from deliveryman are also collected and documented as shown below through conducting an unstructured interview with the deliveryman outside the campus:

1	The deliveryman should be able to get notified when there is a new delivery given to him.
2	The deliveryman should be able to access to all deliveries associated with him.

All collected requirements can be found in Appendix B.

3.2 General Design Convention

3.2.1 Button

By default, there are two kinds of buttons in the app. The first is the borderless button. When displayed in a content area, a borderless button uses context, color, and a call-to-action title to indicate interactivity. And when it makes sense, a content-area button can display a thin border or tinted background to make it distinctive. The second type is button with border. For example, all "confirm" buttons are with border so that they are particularly distinguished.

3.2.2 General Layout

By default, an app interface is divided into three parts: head, body and bottom. These three parts remain consistent in each interface. This design reflects consistency implication of Gestalt psychology and adheres to the design principles of interaction design.

The head part is mainly a navigation bar, used for displaying the title of current interface and "go-back" button. The body part is used for displaying main content. The bottom part is divided into four different metaphor icons. The first one, an abstract human figure, corresponds to 'My Account' interface. The deliveryman figure corresponds to service function, which will jump to 'Service' interface. The clock icon will make it jump to 'History' interface after being touched. The final one is the default 'More' icon. These metaphor designs can enable users to leverage metaphorical understanding to more unfamiliar functionality. They also make learning the app easier and help users understand underlying conceptual model. Based on the feedback from users and individual experience, when holding an iphone, the thumb is often directed to the center of the phone and we pay more attention to the upper part of the screen. In terms of Fitt's law(1954), a button closer to the thumb and with bigger size would take less time to manipulate. As a result, after evaluationg important content or functionality of each interface, principal items are placed in the upper half of the screen and near the left side of the screen, as shown in the figure 1.

For example, with regard to the "ID:xxxx" interface, the most important information(i.e. estimated time, picture of the delivery and consignee address) are centralized in the center of the screen.

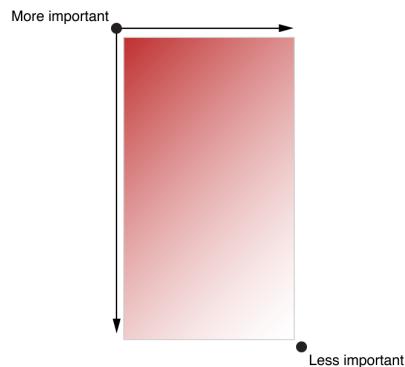


Figure 1: The layout

3.2.3 Negative space

According to Gestalt psychology, the memory of humans is strongly limited, it is necessary to avoid possible interruptions and highlight focus. In this case, negative space makes important content and functionality more noticeable and easier to understand, and can also instill a sense of calm and tranquility, which makes an app look more focused and efficient.

3.3 Low-fidelity prototype

3.3.1 Solid lines & Dotted lines

In general, there are two different types of lines, solid lines and dotted lines. According to Gestalt psychology, we perceive similar objects as things that are perceptually grouped together. As a result, solid lines are mainly used to group similar things together and separate them from other different things. On the other hand, related items are placed along dotted lines and make subtle difference among them distinct. For example, with regard to delivery confirm interface, the first part is all information related to receiver and it needs particularly assured by the user. As a result, two solid lines are used to separate them from the below part. On the other hand, each specific information related to the delivery object is related to each other, so dotted lines are used to align them.

3.3.2 Log-in/Sign-up Interface

As shown in the picture below, there is much negative space around input field which makes it efficient and clean, and this design is requested by the users.

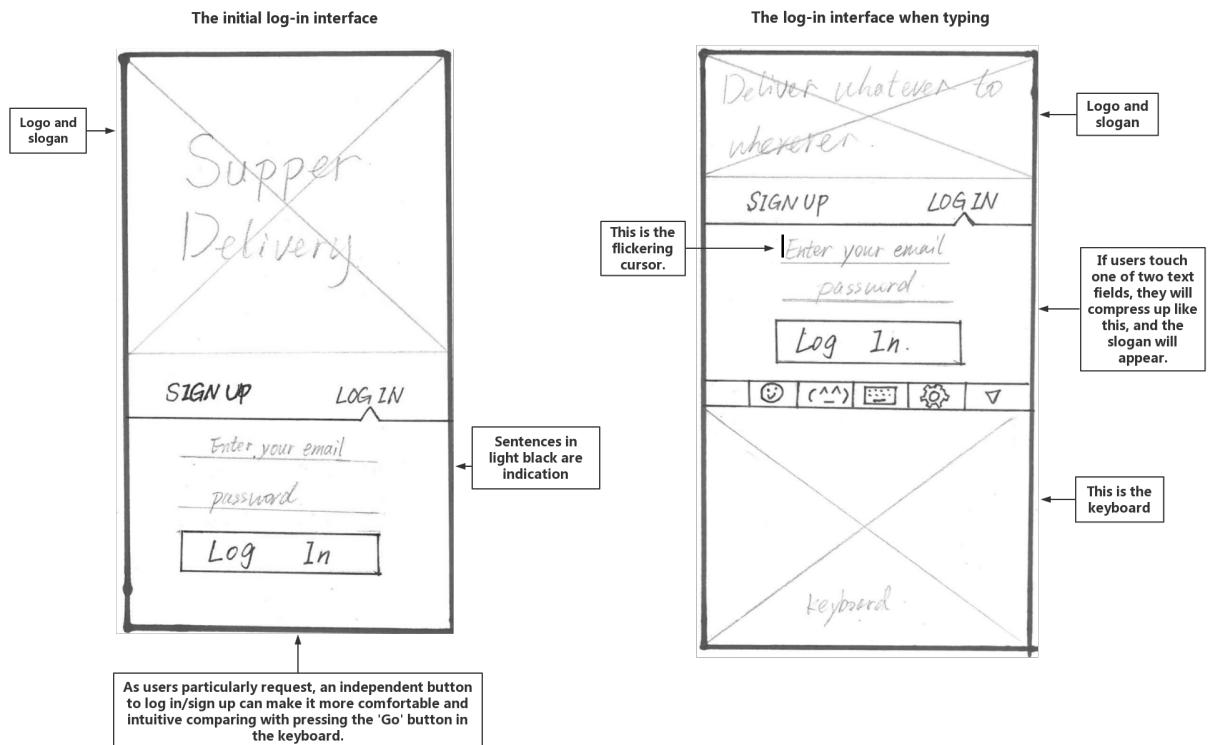


Figure 2: Log-in interface

3.3.3 Delivery History Interface

This interface is the default interface after users log in. It mainly fulfills No.5 and No.6 requirements. Because users find it impossible to remind of the delivery only based on word descriptions, especially when the number of deliveries is large and they are complex. As a result, the picture is displayed to help users memory, which reflects the design implication of Gestalt psychology: it is much easier to recognize information than to recall it.

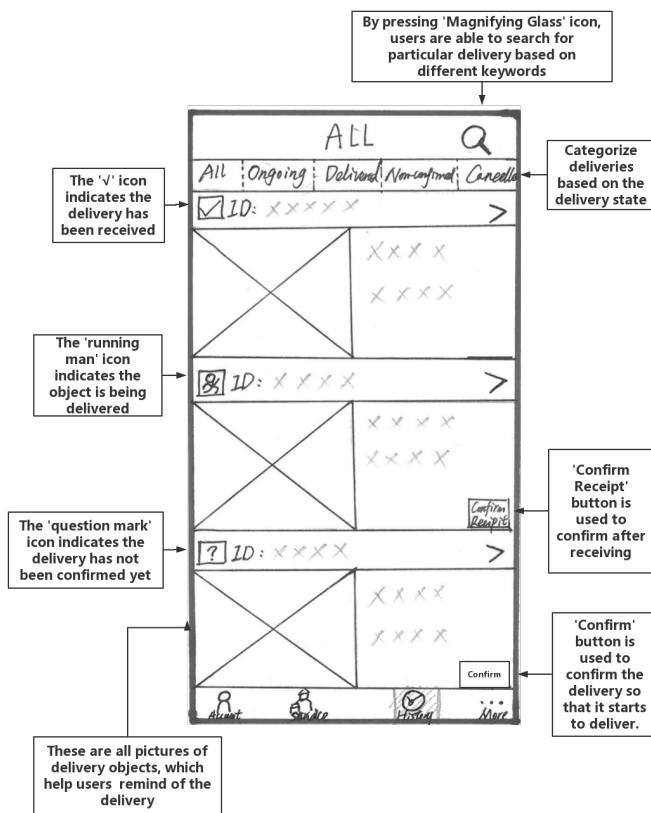


Figure 3: Delivery history interface

3.3.4 ID:xxxx Interface

Figure 4.a shows the interface designed in the first iteration. However, the feedback from users says merely through ID, they cannot recognize the delivery object. Moreover, they want to be able to adjust the arrival time and consignee address(No.8 requirement) easily in this interface and see their decided consignee address if they forgot.

As a result, after negotiating with users, the improved interface in the second iteration

includes the picture of the delivery object to help users remind. In addition, the area of the location part is diminished so that necessary information users have to know when they are going to receive deliveries, such as arrival time, consignee address and identification of the delivery, is more distinct. More information related to locations could be accessed through pressing ' > ' button which links to 'Tracking History' interface. This reflects one design implication of Gestalt psychology: helping users chunk information and displaying limited amounts that we require users to store in memory momentarily. Moreover, 'Confirm Receipt' button is added so that users can confirm conveniently because based on the conceptual model, users often do checking the arrival time, consignee address and confirming the receipt at the same time. The No.8 requirement is fulfilled here.

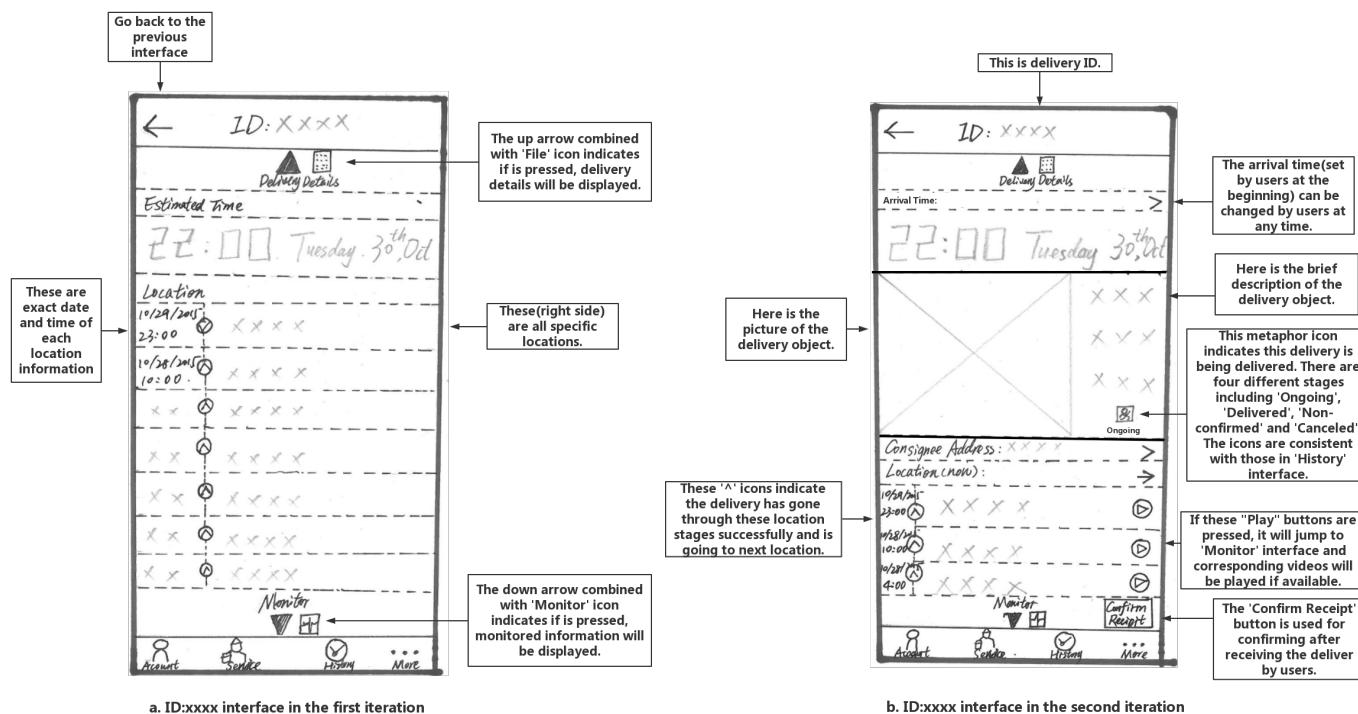


Figure 4: ID:xxxx interface

3.3.5 Tracking History Interface

This interface is displayed after users press ' > ' button in the 'ID:xxxx' interface.

As the figure 5.a shows, this is the original design. However, after showing it to the users, they find that there is no way for them to know what the current delivery is. Moreover, after changing the bottom part into some specific function, they cannot navigate among different interfaces freely.

Consequently, an improved interface recovers the bottom part and make it consistent with other interfaces and moves two icons to the right-up corner of the app. Additionally, considering the conceptual model, because it is sometimes necessary to know how the deliveryman looks like, the picture of the deliveryman along with his information is displayed. For convenience, the location part is scrollable.

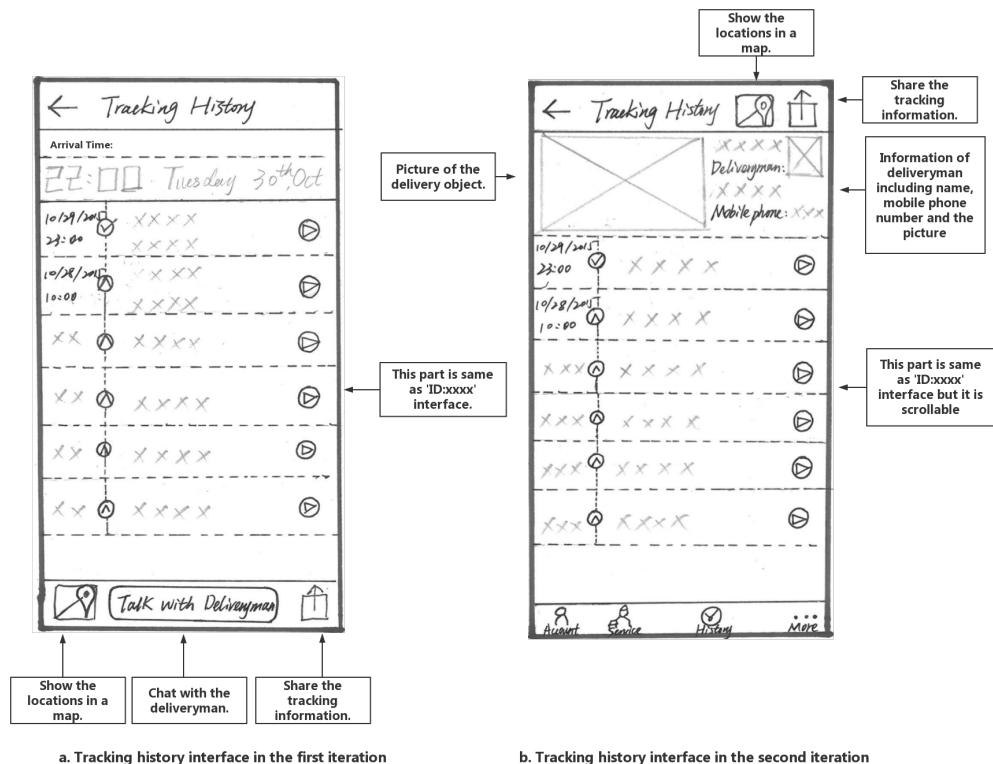


Figure 5: Tracking history interface

3.3.6 Delivery Details Interface

This interface mainly provides necessary detailed information related to the delivery object.

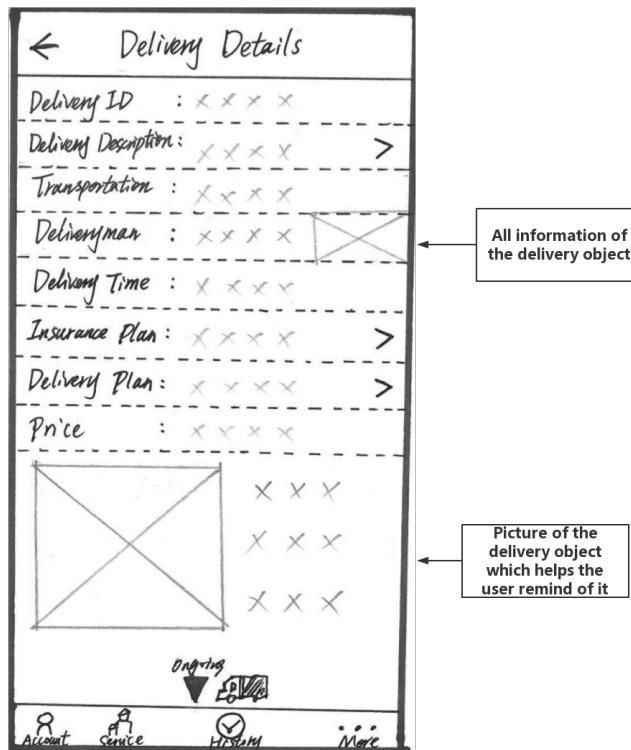
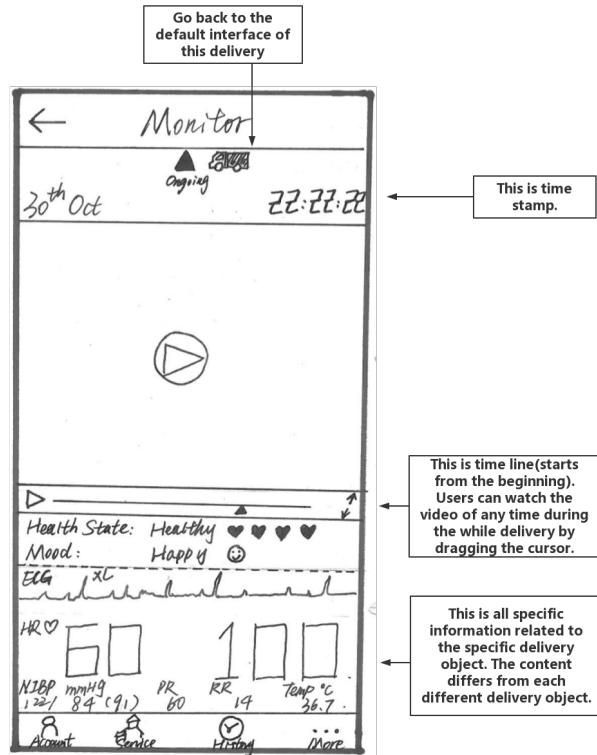


Figure 6: Delivery details interface

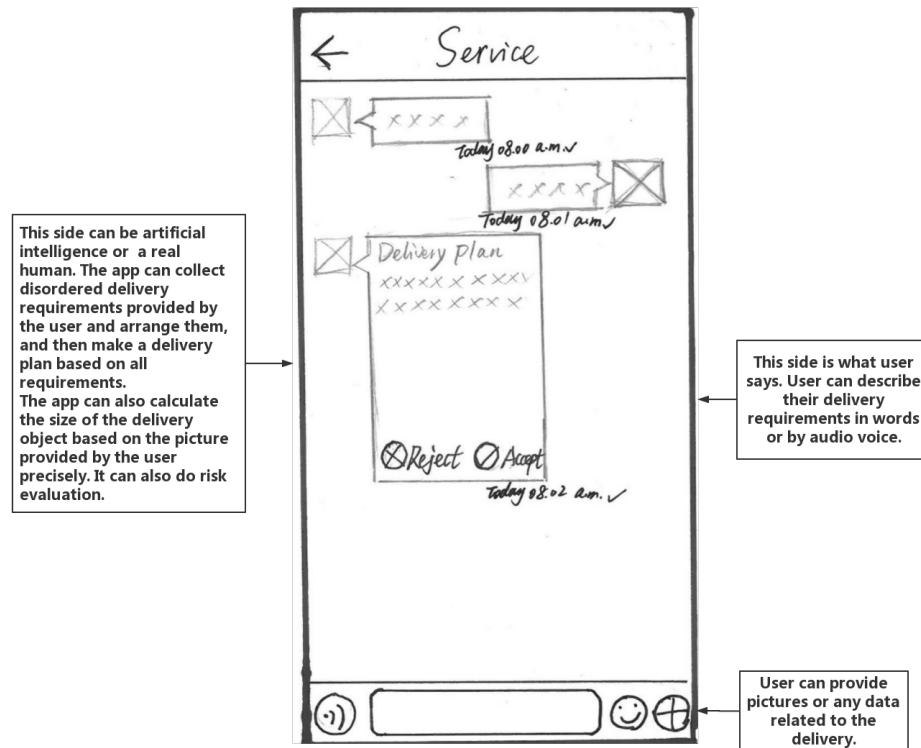
3.3.7 Monitor Interface

The video is not played automatically at the beginning which may be abrupt. This interface also provides specific data about specific delivery object, which fulfills No.8 requirement. For example, health state, mood and vital signs is provided if the delivery is a living body. However, there is actually a trade off made between the completeness of the information and the cleanliness of the interface. In this case, solid lines are used to separate video from a chunk of complicated information and this chunk of information is separated by dotted lines. This design helps neutralize the predicament by allowing users to make a decision whether they only look at the video or perhaps a mishmash of media.

**Figure 7:** Monitor interface

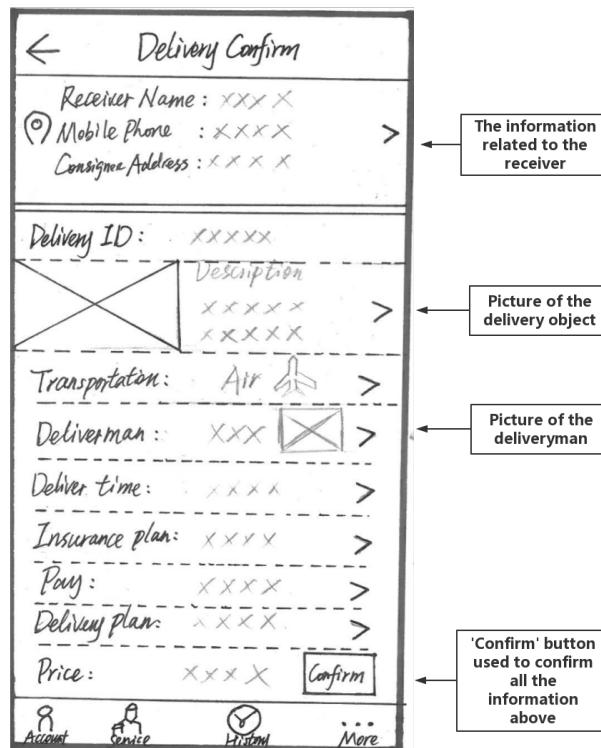
3.3.8 Service Interface

This interface is mainly used for providing a bridge between the user and the app. Whenever the user wants to deliver an object or needs help, he can post his requirements and all necessary information here in a free chatting way. Because the app is capable of collecting and arranging all the user posts and work out a detailed plan automatically, by using this interactively communicative way, the users feel it is the most efficient and convenient. No.1, No.2 and No.3 requirements are fulfilled here.

**Figure 8:** Service interface

3.3.9 Delivery Confirm Interface

This interface displays after the user accepts the delivery plan provided by the app. The double solid line is particularly designed to distinguish the most important information(head part) from some relatively trivial information. There are also some metaphor designs involved, such as the plane icon for the air transportation. No.12 requirement is fulfilled.



The diagram illustrates the 'Delivery Confirm' interface. It features a header with a back arrow and the title 'Delivery Confirm'. Below the header, there is a section for receiver information: 'Receiver Name: XXXX', 'Mobile Phone: XXXX', and 'Consignee Address: XXXX'. To the right of this section is a box labeled 'The information related to the receiver'. The next section contains 'Delivery ID: XXXXX' and a placeholder 'Description' with 'XXXXXX' and 'XXXXXX' entries. To the right is a box labeled 'Picture of the delivery object'. Following this is a section for 'Transportation: Air' with an airplane icon, a section for 'Deliverman: XXX' with a person icon, and sections for 'Deliver time:', 'Insurance plan:', 'Pay:', 'Delivery plan:', and 'Price: XXXX'. A 'Confirm' button is located at the bottom right of the price section. To the right of the price section is a box labeled 'Confirm' button used to confirm all the information above. At the bottom of the interface are navigation icons for 'Account', 'Service', 'History', and 'More'.

Figure 9: Delivery confirm interface

3.3.10 Select Deliveryman Interface

This interface allows users to select deliveryman freely, which fulfills No.4 and No.11 requirements.

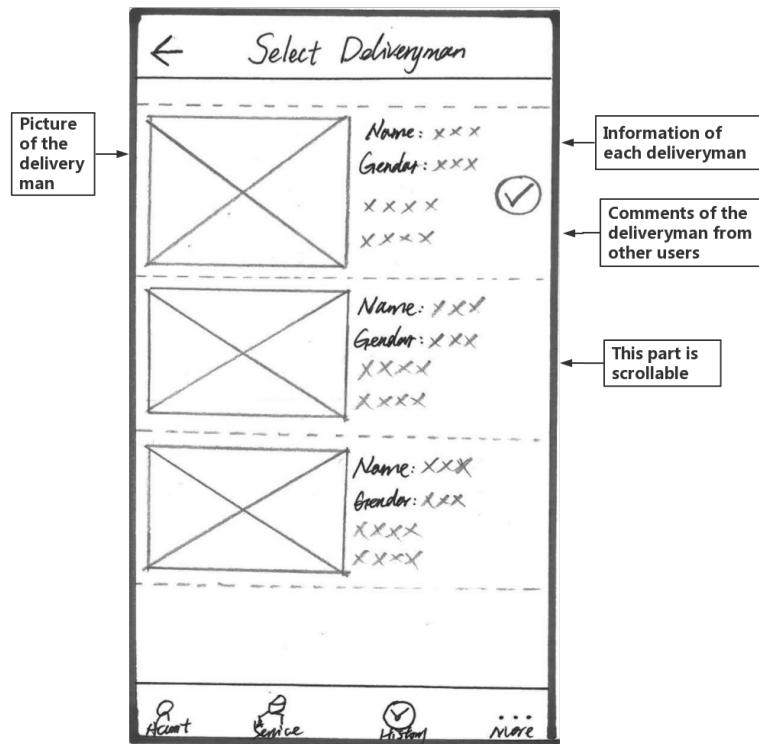


Figure 10: Delivery confirm interface

3.3.11 Confirm Receipt Interface

This interface allows the user to confirm the receipt, evaluate the deliveryman and give comments, which fulfills No.10 and No.11 requirements.

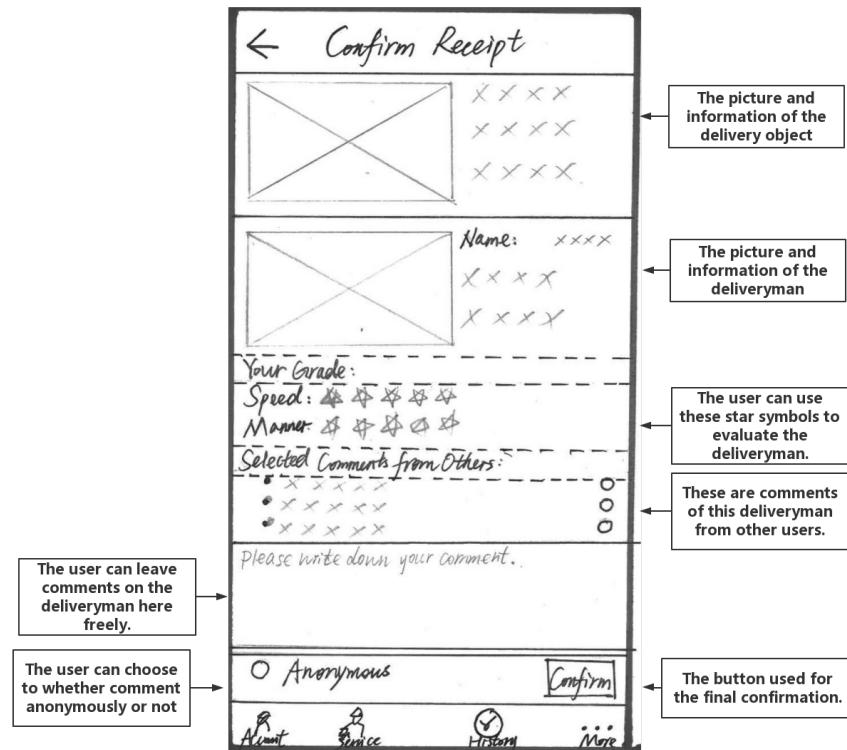


Figure 11: Confirm receipt Interface

3.4 Mid-fidelity prototype

3.4.1 Overview

The structure or general layout of interfaces in mid-fidelity prototype is almost the same as that in low-fidelity prototype apart from the changes show in the figure 12 below and some minor changes. As a result, design ideas are mainly illustrated in the low-fidelity prototype part.

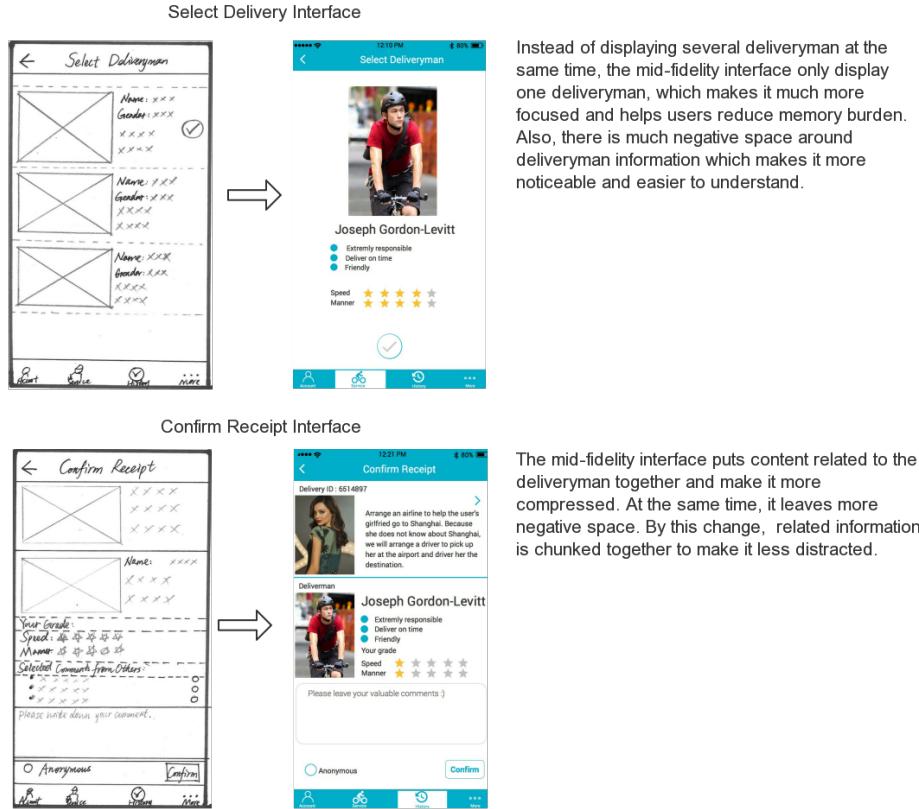


Figure 12: Changes between low-fidelity and mid-fidelity prototype

3.4.2 Navigation

Several navigation designs are implemented to ensure a user's path through the content is logical, predictable, and easy to follow.

- **Navigation bar:** A navigation bar makes navigation through an information hierarchy more easy and management of screen contents ordered. Figure 13 below shows one typical navigation bar used in the app.

**Figure 13:** Navigation bar

- Tab bar: A tab bar allow users able to switch between different subtasks, such as 'account', 'service', 'history' and 'more' in the app.

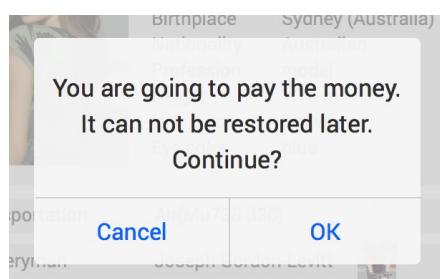
**Figure 14:** Tab bar

- Segmented control: A segmented control can give users a way to see different categories or aspects of the content on the screen; it does not enable navigation to a new screen.

**Figure 15:** Segmented control

3.4.3 User control and freedom

In the user-centered design, users should initiate and control actions. This app takes this principle into consideration seriously. For example, users are able to easily escape from places they unexpectedly find themselves in by naviagation bar. Moreover, the app gives out a reconfirmation dialog after the user taps on the 'pay' button, shown as in figure 16.

**Figure 16:** Dialog

3.4.4 Color and Typography

Basically, the app uses two different background colours: dark turquoise for the sender interfaces and deep pink for the deliverer interfaces as shown in the figure 17 below. Apart from this, colour contrasts can also be found between button background and button titles, interactive and noninteractive elements. Additionally, they are also used to distinguish text content from the context. Colours are only used to help offer visual continuity, imply interactivity and instill vitality, but carefully designed to avoid distracting users.

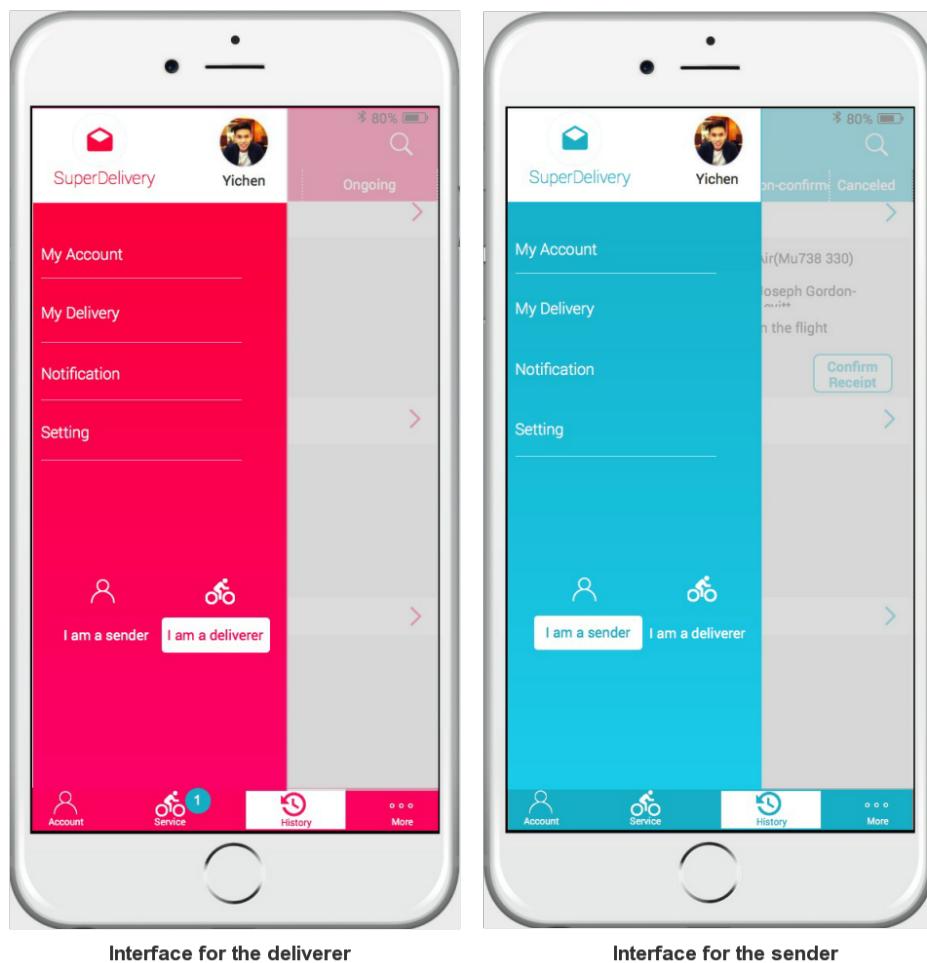


Figure 17: Two contrasted interfaces

3.4.5 Anmation

Subtle animation can make the app experience more engaging and dynamic. For example, screen slide animation used in the app helps enhance the sense of direct manipulation. Changes of the colors of buttons help users visualize the results of their actions as shown in figure 18. The animation in the app also agrees with the practice life and keeps consistent throughout the app. For example, whenever tapping on any ">" button, the screen changes into another screen in a left-slide way, which accords with the intution of right arrow.

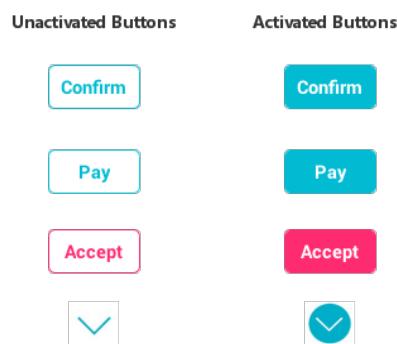


Figure 18: Comparison between activated buttons and unactivated buttons

3.4.6 Detailed Design

- Badge: A badge is a small red oval that displays the number of pending notification items. In the app, it is used to notify senders whenever there is a new request. It appears over the upper-right corner of an apps icon as convention.



Figure 19: The badge which notifies senders

- Gratuitous changes are avoided in layout.
- Because mixing several different fonts may make the app seem disjointed and sloppy, the app limits the font and the styles into a small wide range.
- Feedback: Several independent feedback screens are especially designed to display after users finished some important action. According to Gestalt principles, a delay of

more than 8-10 seconds will cause an interruption, so these interfaces automatically jump to other interfaces within 2 seconds.

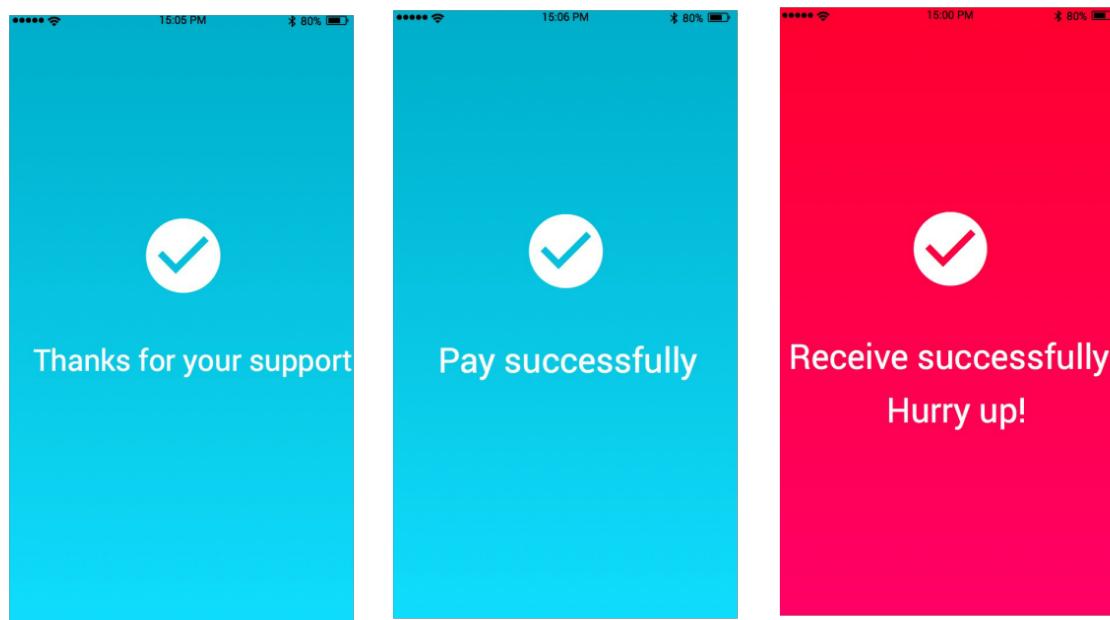


Figure 20: Some feedback screens in the app

3.4.7 Demo and interactive prototype

The video demo and source html file of the interactive mid-fidelity prototype can be accessed through the link: <http://pan.baidu.com/s/1mgH8DF6>

4 CONCLUSION

To sum up, the whole project goes through several stages including requirements gathering, requirements analysis, requirements specification, task analysis, conceptual design and prototype design. Many concepts and ideas related to HCI, cognitive perspective and participatory design are understood and practiced. Through the project, a deeper understanding of user-centered design is developed.

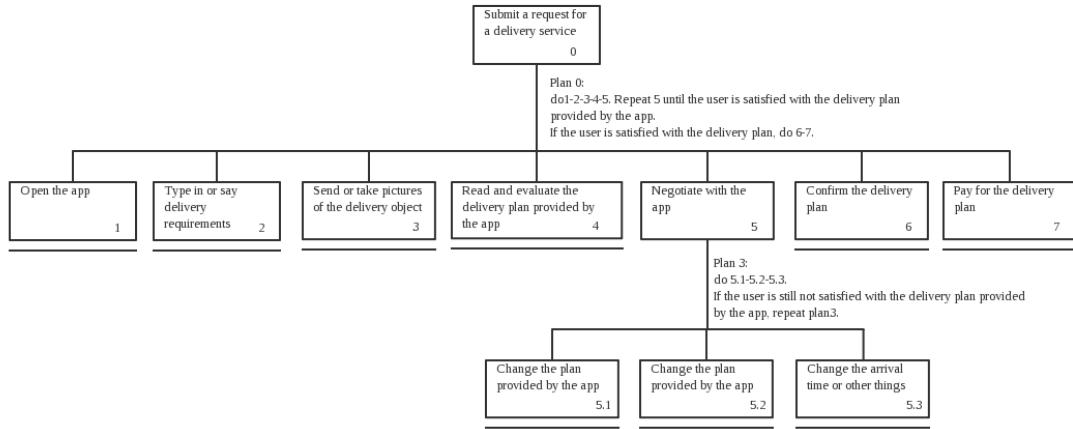
REFERENCES

- MACKENZIE,I. S. (1992) Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*,7,91-139.
- MICROSOFCTORPORATION (1992) The Windows Interface, An Application Design Guide.
- Microsoft Press. MILLER, G. (1956) The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information. *Psychological Review*, 63,81-97.
- NORMAN, D. (1999) Affordances, conventions and design. *ACM Interactions Magazine*, MaylJune 1999, 38-42.
- NORMAN, D. (1983) Some observations on mental models. In Gentner, D. and A. L. Stevens (eds.) *Mental Models*. Hillsdale, NJ: Lawrence Earlbaum Associates.
- Rogers Y., Sharp H. and Preece J. (2014) *Interaction Design: Beyond Human-Computer Interaction*, John Wiley & Sons; 4th Ed.
- PREECEJ., (2000) *Online Communities: Designing Usability, Supporting Sociability*. New York: John Wiley and Sons. This book combines usability and sociability issues to do with designing online communities.

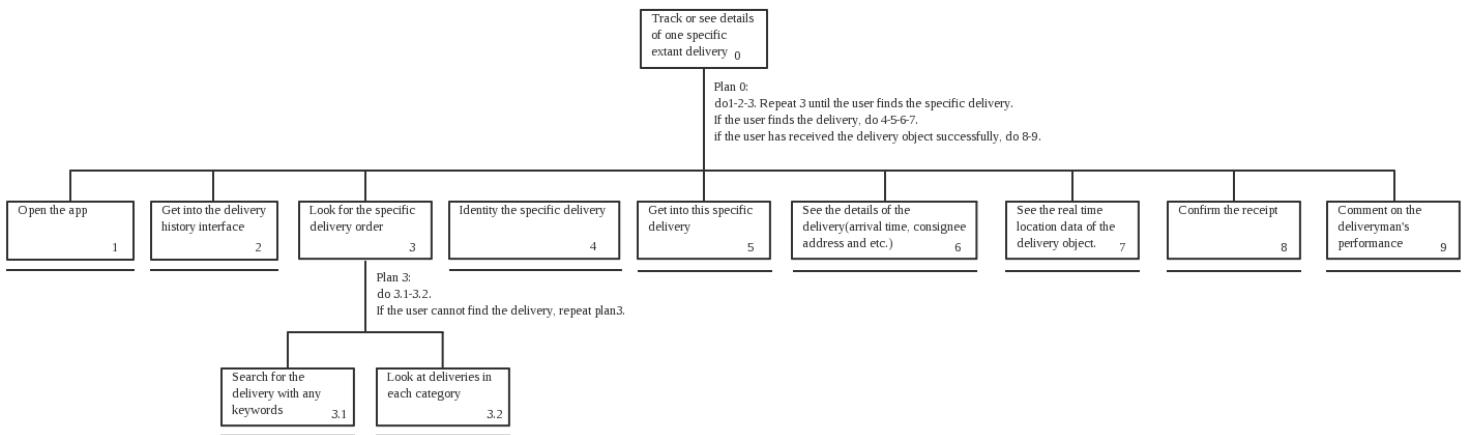
APPENDICES

1. Appendix A: Three HTA diagrams can be accessed through the attachment.
2. Appendix B: Requirement specification can be accessed through the attachment.

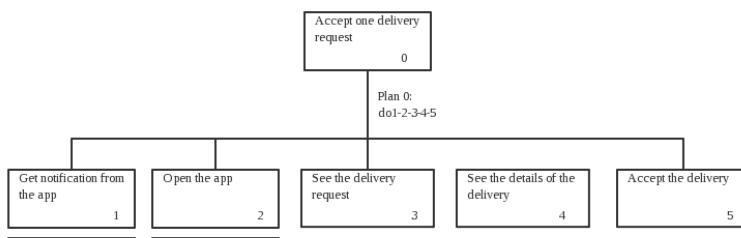
A graphical representation of the task analysis for submitting a delivery request



A graphical representation of the task analysis for tracking or seeing details of the delivery



A graphical representation of the task analysis for accepting one delivery request



Requirements Specification

User requirements:

1. The users should be able to upload pictures of the delivery object and specific data(length, width) can be captured automatically.
2. The user can input information of the delivery objects by typing words and voice.
3. When describing the delivery, AI(like siri) or real people can be provided for communication.
4. The users can choose the deliveryman based on his/her appearance, working performance and users' comments.
5. The users can search for specific delivery with multiple keywords.
6. During the whole process of the delivery, the user can get real time location data, exact estimated time and specific property data.
7. The users can decide and change the destination and receiving time at anytime.
8. The users can get the notification that the items are ready to delivery for them.
9. The user can pay in different ways.
 - AliPay
 - PayPal
 - Credit cards
 - Cash
 - Instalments (monthly, annually)
 - Asking other users to pay.
10. The users can assess the delivery using star symbols and comments.
11. The users can see remarks from others and evaluation of a specific deliveryman.
12. There should be a risk assessment for each specific delivery so that different insurance strategy and delivery plan can be made.

13. The deliveryman should be able to get notified when there is a new delivery given to him.
14. The deliveryman should be able to access to all deliveries associated with him.

System requirements:

1. The app should be able to analyse the pictures uploaded by users and capture the data of the delivery object.
2. The app should be able to parse the literal or voice information provided by the users.
3. The app should be able to provide real people or artificial intelligence to communicate with users to gather and analyse delivery requirements.
4. The app should be able to let users choose different deliveryman.
5. The app should be able to let users assess the delivery performance of deliveryman and store comments and evaluation of each deliveryman from users.
6. The app should be able to calculate and provide real time location data, exact estimated time and specific property data of one delivery.
7. The app should be able to allow users to change the destination and receiving time at anytime.
8. The app should be able to send notification to users when their delivery is ready to delivery.
9. The app should provide different ways for users to pay.
 - AliPay
 - PayPal
 - Credit cards
 - Cash
 - Instalments (monthly, annually)
 - Asking other users to pay.
10. The app should provide the assessing page that includes star symbols and comments.
11. The app should provide the information of the deliveryman and the remarks from other users.
12. The app should make and display different insurance strategy and delivery plan depend on the risk assessment.
13. The app should notify deliveryman whenever there is a new request.