

# **Human Computer Interaction**

## **Assignment III**

### **Full Design and Specification**

#### **Food Order Website**

Teammates

HoKang Yu | yu2749luca

Hongdan Zhu | hongdanzhu

Xiaoyan Qin | qxy

Yunzhu Chen | angrycyz

# **Table of Contents**

[Design Specification \[Part I\]](#)

[Semantic Level Design](#)

[Syntactic Diagram](#)

[Lexical Model](#)

[Documentation and Guidelines \[Part II\]](#)

[Task-Command Analysis \[Part III\]](#)

## Design Specification [Part II]

*We will be following the Design-by-Level theory for the design specification. From your previous assignment, you have already completed the conceptual-model. You may optionally submit any revisions you have.*

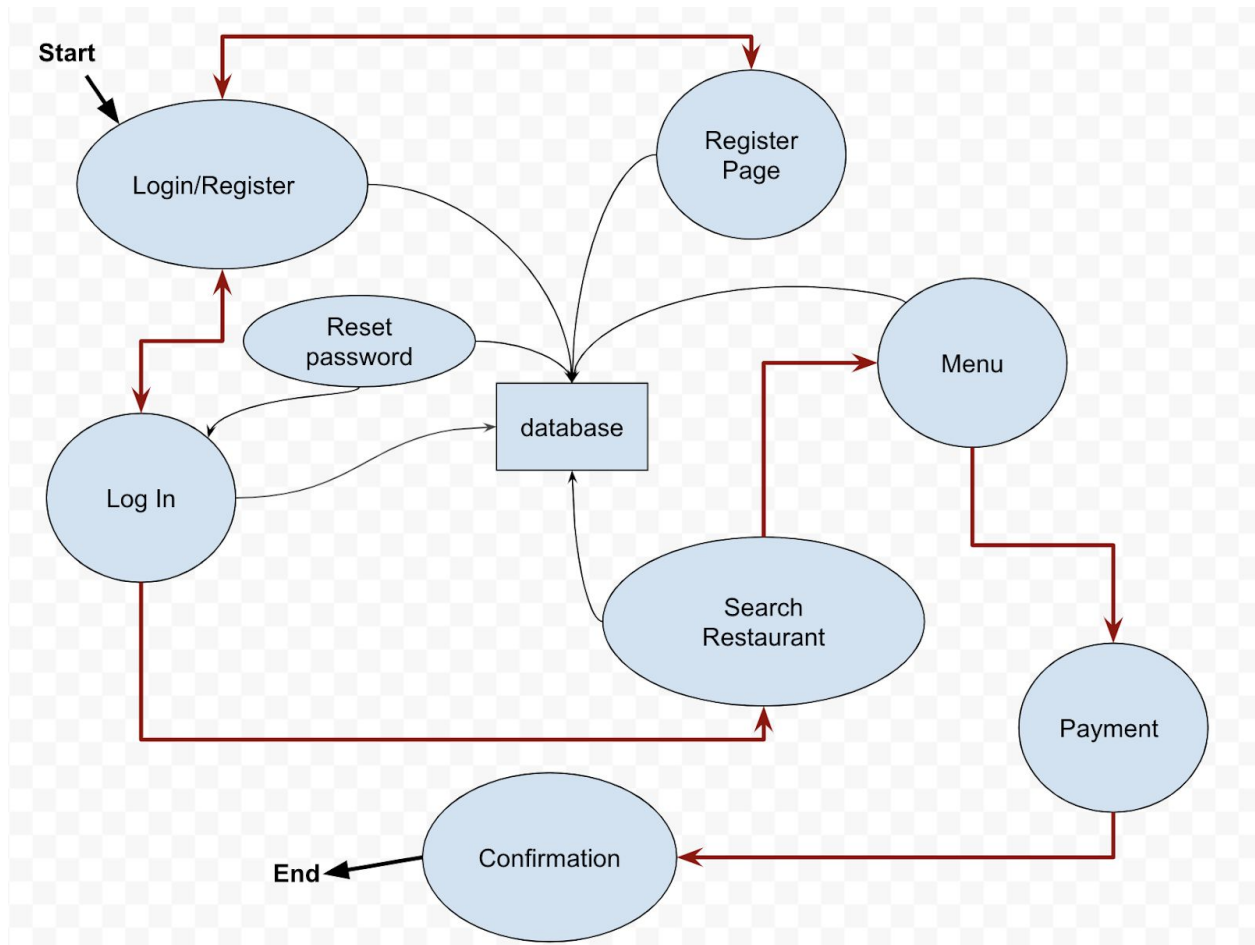
- **Semantic Level Design**

*Semantic diagram gives an organized listing of the functionality of a system. This should include at least 4 functions, with their parameters, description, feedback, and errors.*

Function Name	Parameter	Description	Feedback	Errors
Log-in	Username/ Password	Login to use our service	If success, menu page will be shown	If account info does not match, an error message will be prompted
Search Area	Integer representing error code	Enter Area code to find restaurant	If success, a list of restaurants will be shown	If area code is invalid, an error message will be prompted
Payment	Credit Card No, Expired Date, Security Code	Enter payment information to confirm pick-up order	If success, an estimated pick-up time will be shown	If payment is invalid, an error message will be prompted and no order will be sent
Register	Name, Email, Phone Number	Enter required info in order to register	If success, our database will store users' info. And we will send a verifying email to users' mailbox to confirm.	If the email has been used to create an account, an error message will be prompted.

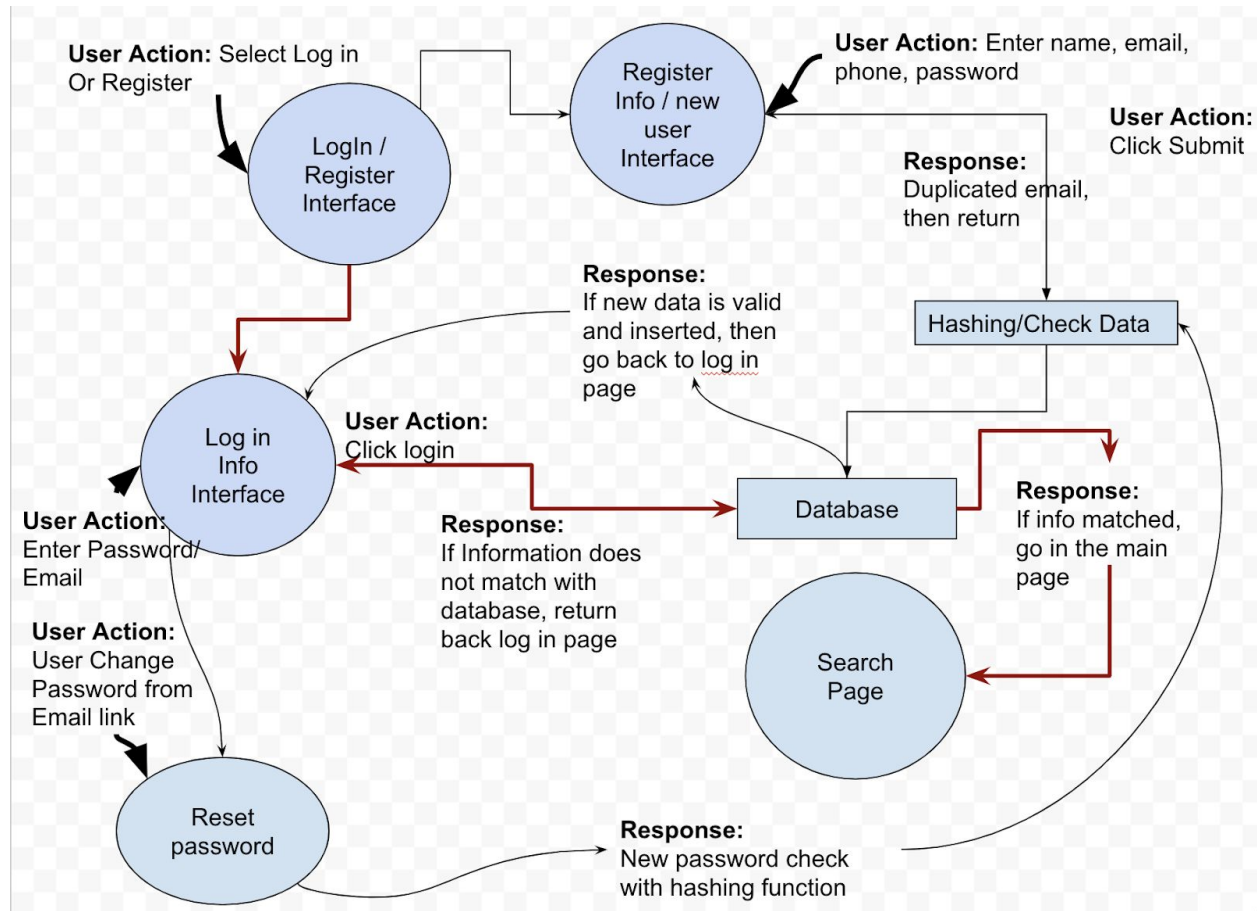
- **Syntactic Diagram**

*This is a listing of user actions and system responses. You should have at least a high level syntactic diagram, and then go into detail on 2 subsystems. These two subsystems could be the commands that are in your task analysis for instance.*



[OverView Diagram]

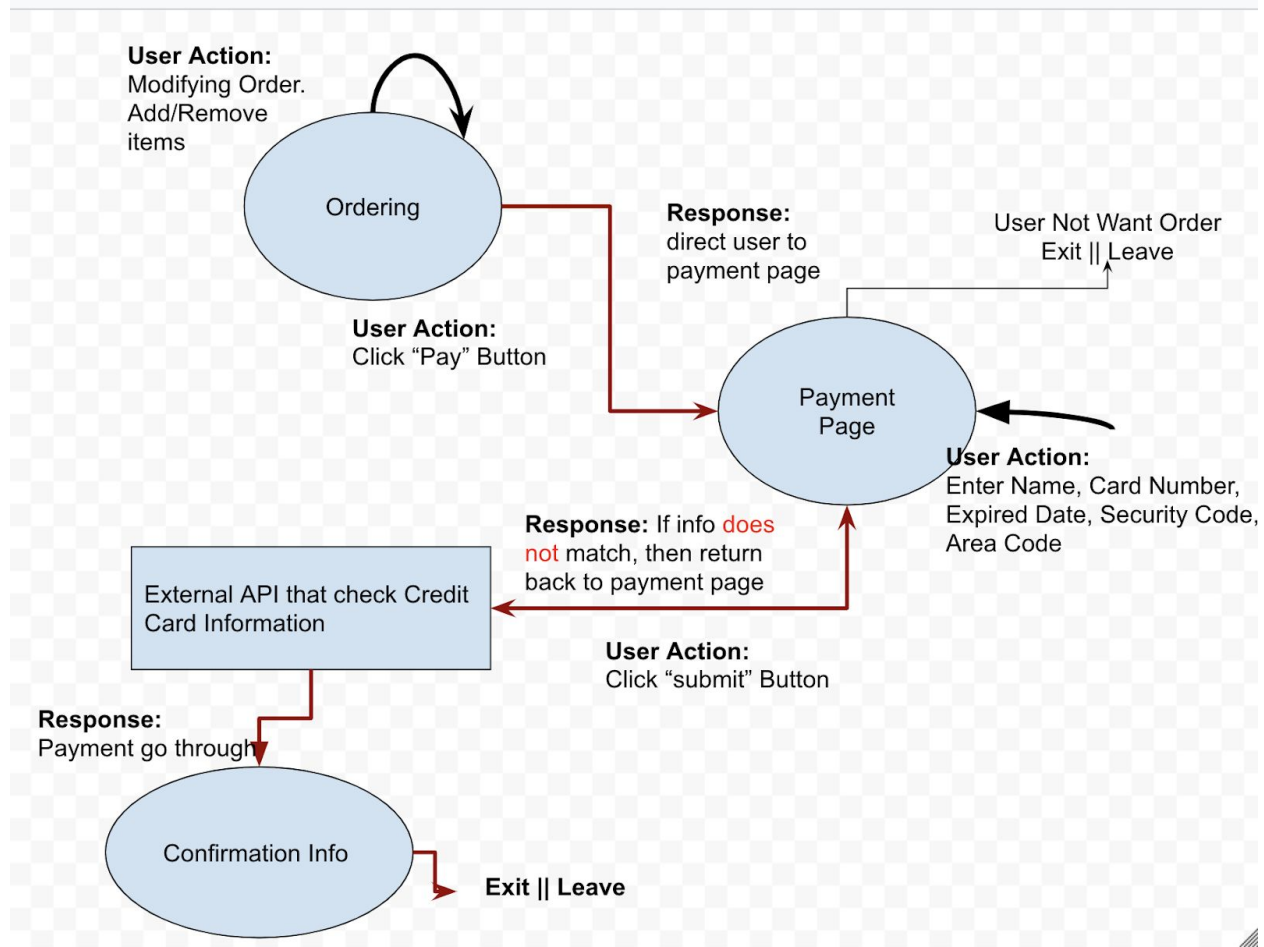
The above picture is the overview syntactic diagram for our entire system. User will need to log in, search restaurants, select orders from menu, then pay for the service. Then we give a detail information for two subsystem, users' account information page and payment page. In order for readers to understand the diagram easier, we use round shape to represent a front-end, square shape to represent back-end, and **dark-red arrow** to represent main flow of the process.



[ LogIn/Register Sub System Detail]

The above picture is a detailed diagram for the account information page. User will need to click if he/she/it wants to log in or register in the home page. When logging in, we need to check if username and password are input correctly corresponding to our database. If not, then a message will remind user, then the user will be brought back to the login page. If the user input the right information, then the user will be brought to the **search page**. And user will also have an option to reset password using email. The new password will need to be checked with our hashing function before going into the database.

When a new user is registering an account, he/she/it will need to enter email, phone, and password. If the email has being used or password is too weak when we check with our hashing function, then the user will be prompted with a message telling he/she/it to revise account info. If the new info has passed our hashing function, then the new info will be inserted into database; then the user will be brought to log in page. Then the user will need to log in with the information.



[ Payment, Sub-System Detail, Info]

The above diagram is the detail information for payment page. When user is done with selecting and modifying items in menu page, he/she/it will click “pay” button on the page. The user will then be prompted to enter name, credit card, expiration date, and security code of the credit card. Then user will click “submit”, then the information will be tested and checked with an external API. If the API-check result is negative, the user will be brought back to the payment page. If the result is positive, the user will be brought to the confirmation page.

## Lexical Model

*This is a glossary for each of the user actions and system responses that you have highlighted in your syntactic diagram.*

### Login:

After user **input information** (email and password) in the login page, user will need to **click log in**. Then the information will be checked with our hashing function and database. If the information is accurate, then the user will be directed to search page. However, if the information is incorrect, the user will be directed back to the login page. If the user choose **click reset** password, user will receive an email to reset password.

### Register:

After user **input information** (name, email, phone) in the register page, the information will be checked with our hashing function and database. If the information is unique, then the user will be directed to the login page. If the information is a duplicate, then the user will receive an alert message and be asked to **revise the information**.

### Payment Page:

In the menu page, user will need to **click pay** in order to go into the payment page. After user **input** all the required information in the payment page, user will need to click **submit** in order to send the credit card information to be checked. If the information does not pass through the external API, then the user will be brought back to the payment page.

## **Documentation and Guidelines [Part II]**

### **Documentation:**

This food ordering for pick website we is a product that allows customers to order food pick-ups for multiple restaurants. Instead of paying 5 dollars for Uber drive or Doordash dasher to deliver your food, our website allows you to pick up your food on your way home!

#### **Find restaurants and view menu:**

1. Firstly, customers can register an account on the website or just order as a guest. You can click on the register button to set up a username and password.
2. Once you have a username and password, you can use them to log in. You can sign in to our website to manage their account information like payment method, saved restaurants, liked food.
3. To start ordering, you can now put in the postal code of the area that the restaurants you may want to order located in.
4. Then, our webpage will give you a list of the restaurants that are located near that poster code. The opening time and food menu of the restaurants will also be listed on the page.
5. You can then click into the restaurant you are interested in and check their menu today and the pictures of food, to decide what you want to eat.

#### **Check chart and make a payment:**

6. Our website allows you to add food into chart and view the char. Food in chart could be deleted if the customer change their mind and no longer want it.
7. Our website allows multiple payment methods like credit card, Paypal or debit card. Our site currently only allows online payment instead of pay in the store when you pick up.

#### **Check order status:**

8. After you paid the price for the food, you will be able to see the confirmation of this transaction and you can also see the estimated time of when the food will be ready.
9. The restaurant will receive your order and start preparing your food. You can view your order status to see if your food in “preparing” or “ready for pickup”.

In this way, you can save a huge amount of time waiting in the waitlist lane to get a table, or sitting in the restaurant dine in and waiting for server get your food. You save a lot of time when you make a payment in store, give your card to server and wait for server come back and give tips.

You may have multiple restaurants' and cafes' apps that help your order a pick-up from them. You may have already registered accounts with multiple restaurants just to order food pick-up from them. With these many accounts, you may even have a hard time updating software or personal information with each app. We are launching a product that solves this issue by being the medium of you and all the restaurants for order-pick-ups.



< Locations

Log In

Korean BBQ beef, sauteed onions, pineapple, sesame vinaigrette, lettuce, red sauce and sesame seeds served on crispy fried garlic rice buns.

**Chicken KoJa**

**\$6.99**

Korean BBQ chicken, sesame vinaigrette lettuce, tomato, cilantro, spicy gochujan sauce, sesame seeds, lemon wedge served on garlic fried rice bun

**Braised Pork KoJa**

**\$7.99**

Miso coconut braised pork, arugula, wonton chips, garlic aioli, masago, lemon wedge, served on garlic fried rice buns.

**Bowls**



**Ahi Tuna Bowl**

**\$11.99**

Ahi tuna, wasabi mayo, seaweed salad, imitation crab salad, green onions, masago, wonton crisps, soy ginger dressing, red sauce served on your choice of white or brown furikake seasoned rice and sesame vinaigrette spring mix.

**Beef Bowl**

**\$9.99**

Korean BBQ beef, sauteed onions, red sauce, kimchi, masago, green onions, fried shallots, sesame seeds served on your choice of white or brown furikake seasoned rice and sesame vinaigrette spring mix.

**Short Rib Bowl**

**\$10.99**

Korean BBQ short rib, katsu aioli, kimchi, masago, green onions, fried shallots, sesame seeds served on your choice of white or brown furikake seasoned rice and sesame vinaigrette spring mix.

**Chicken Bowl**

**\$7.99**

Korean BBQ chicken, spicy gochujan sauce, kimchi, masago, green onions, fried shallots, sesame seeds served on your choice of white



**KoJa Kitchen - Alameda**

2680 5th St Ste D  
Alameda, CA 94501

**Pickup - Order for Now**

SUBTOTAL \$0.00

TAXES \$0.00

Order amount is below \$0.50 pickup minimum.

Checkout

\$0.00

**\*Sample ordering page and checkout page**

**Beef Bowl**

**\$9.99**

Korean BBQ beef, sauteed onions, red sauce, kimchi, masago, green onions, fried shallots, sesame seeds served on your choice of white or brown furikake seasoned rice and sesame vinaigrette spring mix.

**Short Rib Bowl**

**\$10.99**

Korean BBQ short rib, katsu aioli, kimchi, masago, green onions, fried shallots, sesame seeds served on your choice of white or brown furikake seasoned rice and sesame vinaigrette spring mix.

**Chicken Bowl**

**\$7.99**

Korean BBQ chicken, spicy gochujan sauce, kimchi, masago, green onions, fried shallots, sesame seeds served on your choice of white or brown furikake seasoned rice and sesame vinaigrette spring mix.

**Braised Pork Bowl**

**\$8.99**

Miso coconut braised pork, garlic aioli, kimchi, masago, green onions, fried onions, sesame seeds, lemon wedge served on your choice of white or brown furikake seasoned rice and sesame vinaigrette spring mix.

**Tacos & Salad**



**Sides**



**Drinks**



2680 5th St Ste D  
Alameda, CA 94501

**Pickup - Order for Now**

1 Beef Bowl \$9.99

1 Chicken Taco \$3.09

1 Soy Garlic Wings \$7.99

SUBTOTAL \$21.07

TAXES \$2.05

Checkout

\$23.12

## **GuideLines:**

### **Design and Style guidelines**

1. We will keep the consistent implements standards by using familiar icons, standard text styles, and uniform terminology, so that user can finish the transaction from one activity to another smoothly.
2. We will try to make the ordering process fast and easy as our main style.
3. We will try to make frame and color as clean as possible since we are making a food ordering app.

### **Team Guidelines**

1. We will build the frontend web pages by using HTML, CSS and JavaScript.
2. For the restaurant locations information, we will call the Google Map APIs to get the data back.

### **User Interaction**

1. Loading:  
when it takes long for the web page to show up, we will have a loading bar to remind the user that the webpage is not frozen.
2. Authentication/Permission:  
When our web request the users' current location, we will provide a brief explanation on the page to let the user know the reason for request the permission.
3. Data entry:  
We will provide dynamically validate field values for users to help them correct typos and mistakes.  
When we ask users for their email address, we will show the helpful data entry shortcuts for email.
4. Restaurant search:  
We will optimize the user experience by developing an autocomplete restaurant search bar providing a list of restaurants in the suggestion list.

### **Page Views**

1. We will implement the scroll of the pages when one page is not enough for the room. The users can scroll down and up to look at the information they want.
2. If the user clicked something on the page by mistake, we can let them use the backward function to return to the last page.
3. We will also try to remember users' favorite restaurants and favorite dishes. The next time they log into the account, they can see their record and favourites in the main page.

## Task-Command Analysis [Part III]

*Perform a task-command analysis using the KLM model. You should perform the task-command analysis for each of your 3 functionalities that you highlighted in your previous assignment. Make sure to have an analysis at the end, where you can comment on improvements/insights found after doing the analysis (if it is perfect, then comment why).*

### How to apply KLM

#### 1) Log-in (*Assume users already home hands to mouse*)

How users do the task	Keystroke-level actions	Standard execution time(sec)	Total estimated time
Users click 'account' box, then enter the name in the box, and then click the password box and enter the 'password' in the box. Finally users click the 'log-in' button.	<b>P</b> - Point to the 'account' box	1.1	$7.9 + t + 2 * n * K$
	<b>BB</b> - Press the 'account' box and release	0.2	
	<b>M</b> - Remember username	1.2	
	<b>H</b> - Home hands to keyboard	0.4	
	<b>T(n)</b> - Type username(K operators)	$n * K$ sec	
	<b>H</b> - Home hands to mouse	0.4	
	<b>P</b> - Point to the 'password' box	1.1	
	<b>BB</b> - Press the 'password' box and release	0.2	
	<b>M</b> - Remember password	1.2	
	<b>H</b> - Home hands to keyboard	0.4	
	<b>T(n)</b> - Type password (K characters)	$n * K$	
	<b>H</b> - Home hands to mouse	0.4	
	<b>P</b> - Point to the 'log-in' button	1.1	
	<b>BB</b> - Click and release 'log-in' button	0.2	
	<b>W(t)</b> - Waiting for the system to respond	t	

#### 2) Search Area (*Assume users already home hands to mouse*)

How users do the task	Keystroke-level actions	Standard execution time(sec)	Total estimated time
Users press the 'search' button, a 'search' box displays, users select the 'search' box, enter the zip code in the 'search' box, press 'enter' key on the keyboard.	<b>P</b> - Point to the 'search' button	1.1	$4.28 + t + n * 5$
	<b>BB</b> - Click and release 'search' button	0.2	
	<b>P</b> - Point to the 'search' box	1.1	
	<b>M</b> - Remember the zip code	1.2	
	<b>H</b> - Home hands to the keyboard	0.4	
	<b>T(n)</b> - Type zip code(zip code - 5 characters)	$n * 5$	
	<b>K</b> - Keystroke 'enter' key	0.28	
	<b>W(t)</b> - Waiting for the system to respond	t	

3) Payment (*Assume users already home hands to mouse*)

How users do the task	Keystroke-level actions	Standard execution time(sec)	Total estimated time
Users press the 'pay for order' button, and select the 'Name on card' box, enter the name of credit card owner, then select the 'Numbers on card' box, 'Expire date' box, and 'CVV' box, and enter the card number, expiration date, CVV number respectively.	<b>P</b> - Point to the 'pay for order' button	1.1	$8.44 + n * (K + 7) + t$
	<b>BB</b> - Click and release 'pay for order' button	0.2	
	<b>P</b> - Point to the 'Name on card' box	1.1	
	<b>M</b> - Remember the name	1.2	
	<b>H</b> - Home hands to the keyboard	0.4	
	<b>T(n)</b> - Type name(K characters)	$n * K$	
	<b>K</b> - Keystroke 'Tab' key	0.28	
	<b>M</b> - Remember the number	1.2	
	<b>T(n)</b> - Type card number(16 characters)	$n * 16$	

	<b>K</b> - Keystroke 'Tab' key	0.28	
	<b>M</b> - Remember the expiration date	1.2	
	<b>T(n)</b> - Type date(4 characters)	$n * 4$	
	<b>K</b> - Keystroke 'Tab' key	0.28	
	<b>M</b> - Remember the CVV	1.2	
	<b>T(n)</b> - Type CVV(3 or 4 characters)	$n * 3$	
	<b>W(t)</b> - Waiting for the system to respond	t	

Analysis:

- 'Log-in' and 'payment' functionality require the basic information provided by users, which cannot be simplified. However, 'Log-in' functionality should allow user to use tab on the keyboard other than using the mouse, which could save the time pointing mouse to the next input box. Besides, 'search by area' functionality can be improved by providing users frequently used zip code, which can reduce the keystroke-level actions such as pointing to the search box P, type T(n), and save the time of mental act. It could also be improved by adding a new feature 'current location' which directly gets the current location and zip code, so that the keystroke-level action mental act could also be omitted.