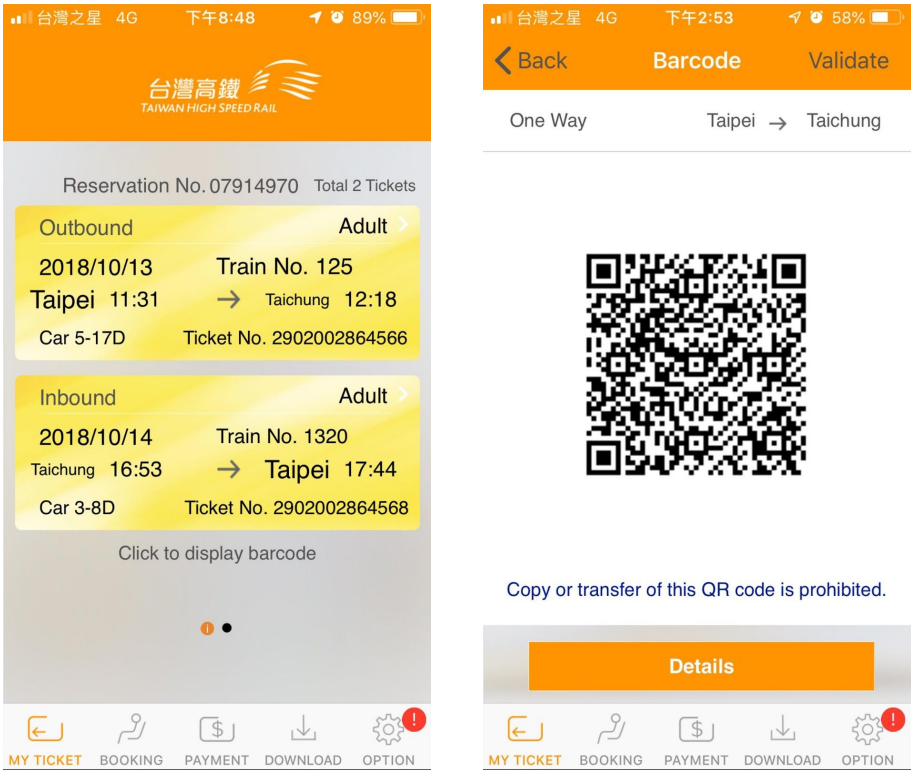
CS6750, Project P

MingTing Lu
mlu305@gatech.edu

Heuristic Evaluation

What Works Well and Why

Taiwan High Speed Rail (THSR) is the fastest rail in Taiwan. The company developed an application both on iOS and Android, called "*T-Express*" (T-EX in brief). Today, I would like to analyze and redesign T-EX.



(a) (b)

Figure 1. (a) The e-ticket list in T-EX. (b) The QR code ticket screen.

T-EX has several functions in different pages: My Ticket (view e-ticket), Booking, Payment, and Download (reserved tickets). The key feature is to use the app to

generate a QR code to take the train. In other words, you can take the train you reserved from the app or from the website through T-EX. (The turnstile in the train station is capable of using QR code to pass.) This is a very convenient design for users because they don't need to hold the ticket and to be afraid of losing the ticket while taking the train. This makes the interface of the turnstile **easy** to use.

The task of display the QR code ticket on the screen has a small **gulf of evaluation**. After tapping the ticket in Figure 1.a, the screen will directly show the QR code of that ticket (interface output). The user captures that the QR code is displayed (interpretation: there is a QR code now), and then checks whether the QR code is the target ticket (evaluation) with the destination information provided at the top of the screen (see Figure 1.b, "*one way Taipei -> Taichung*"). The button "*Details*" at the bottom of this screen leverages **discoverability** of heuristic principle. This is also a good example of a **good representation** because it eliminates the extraneous details from the screen in order to keep it neat and tidy.

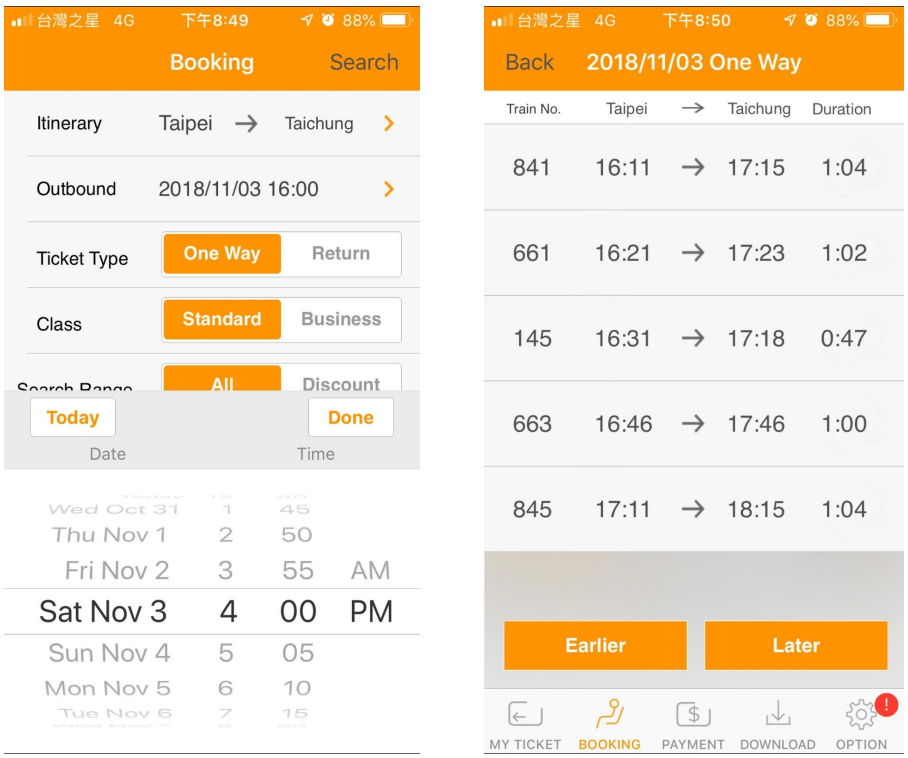
Further, the design of the ticket list is good for 2 reasons. First, the layout and the information on the e-ticket are similar to the physical train ticket, which uses the **mapping** principle to help users recognize that is the train ticket. Second, the tickets are laid in grids, clear separation between tickets leverages the **structure** principle.

What Doesn't Work Well and Why

However, in the aspect of the **gulf of execution,** displaying the QR code ticket is a little bit large for a novice user comparing to the gulf of evaluation. They may hard to identify their intention of opening the QR code ticket, for there is only a little indication: "Click to display barcode". Lack of proper **discoverability** of tapping the ticket in the ticket list to open it up makes the action and the affordance be hard to identify by novice users.

Another feature of T-EX is that is could reserve a seat directly from a user's smartphone. Nonetheless, the function has several parts doesn't work well. First, in the booking page, the outbound time setting is not **consistent** between the form and the selection scroll. In Figure 2.a, on the top, the outbound time is "*2018/11/03 16:00*", but in the selection field, it is "*Sat Nov 3, 4, 00, PM*". Second, the design of scrolling selection of outbound time is easy for users to make a **slip**. Especially when the date is very crucial to the process of making a reservation, a slip on date selection

without notice by the user would be a disaster. Further, the time selection is not that important in this part, but it takes 3 fields to select (4, 00, PM). This is not a good **representation** because of the extraneous details here. Moreover, in the seat searching stage, the "*Search*" button (at the top right in Figure 2.a) is not that clear for users to find. This could be recognized as lacking **discoverability**. Furthermore, The style of this button differs from other buttons on the screen which is not **consistent**. The "*Search*" button is only grey font color with no outline, while the "*Today*" and "*Done*" buttons are white background color with orange font color and square shape. Last but not least, after pressing the "*Search*", it turns to the search result page (see Figure 2.b). To view other available trains, a user needs to press "*Earlier*" or "*Later*", which doesn't follow the **ease** and **comfort** of heuristic principles. Also, here has the same problem of different button style, and here is the third type of button: orange background with white font color, which the style is the selected field in the previous page.



(a) (b)

Figure 2. (a) The seat search form of the booking process. (b) The search result (available trains).

Interface Redesign

Since this is a mobile app, I layout my redesign with 4 mock-ups. The redesign in each figure is outlined with a red line and explains in the figure notation.

Figure 3. Adding QR code icon on the e-ticket. Removing extraneous information "*Adult*" "*Outbound/Inbound*", and "*Ticket No.*"

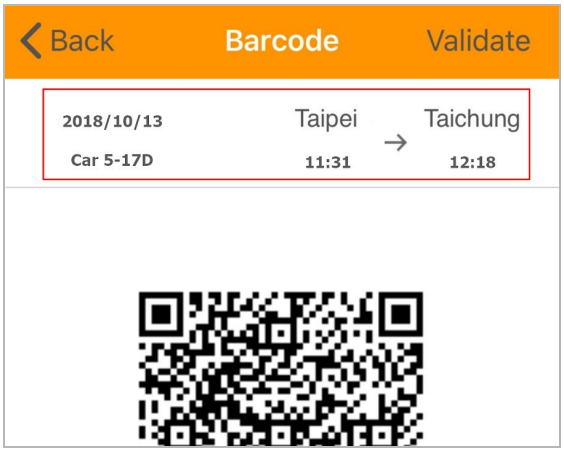


Figure 4. Adding train information, including date, bound and arrive time, and seat number.

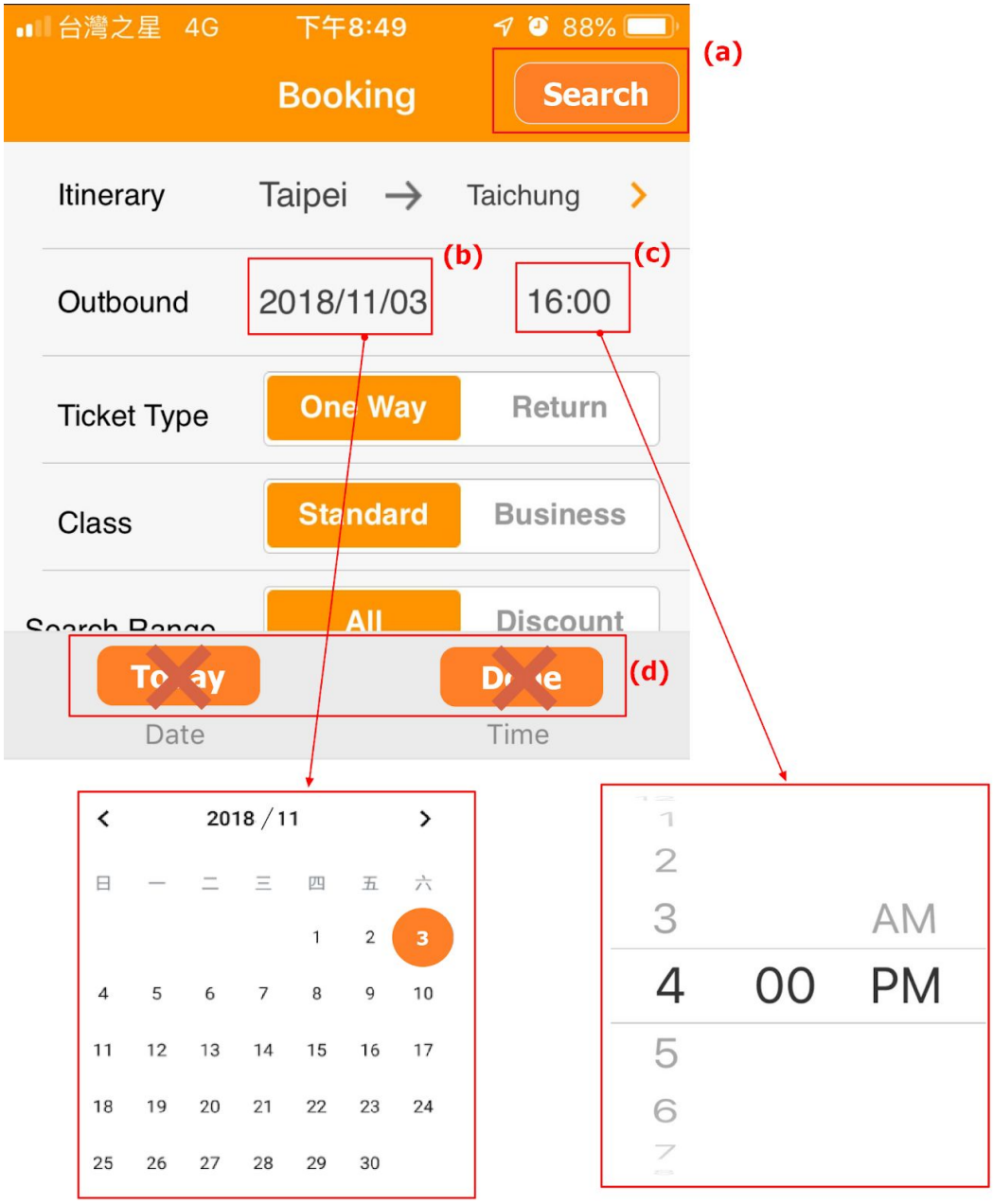


Figure 5. (a) Make the "Search" button be similar to other buttons in this app. (b) Tap the outbound date to select a date. The date selection method is the same as Google Calendar. (c) Tap the outbound time to select time. Only need to select the hour in the 24-hour clock, removing AM/PM selection. (d) Remove buttons when date selection. The selection panel will automatically close after tapping a specific date.

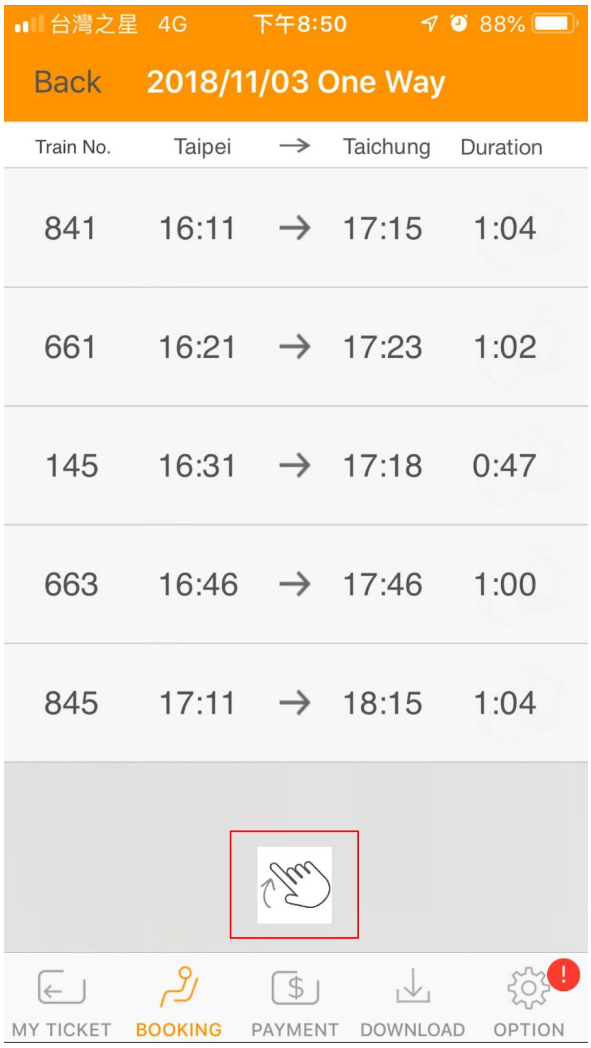


Figure 6. Make the the page could be swiped up and down, and add a scroll bar on the side when scrolling and a gesture indication in the first page (with 10 trains). The app will load the previous trains while scrolling up to the top of the list, and load the later trains while scrolling down to the bottom. Just like viewing the wall posts from the middle of the wall on facebook.

Interface Justification

Redesign Ticket List (See Figure 3)

My first redesign of the interface of T-EX is to add a small QR code icon button on the yellow e-ticket (see Figure 3). This is to add a clue for **discoverability** that a button on the ticket will display the QR code of that ticket (but actually you can tap anywhere on the yellow e-ticket to call out the QR code.) This is to narrow the **gulf of execution**, making the action could be identified more intuitively.

Further, to make the user know it is a button, I choose an orange background icon to make the button style be **consistent** with the most frequent button style in this application. I would like to change all the button in this style to make the button style be consistent so that user will know what are buttons and what are not.

Moreover, The information on the yellow e-ticket is a little bit too much. Information such as "*Inbound/Outbound*", "*Adult*", and "*Ticket No.*" are not necessary at all while a user is viewing ticket list. These extraneous details could be put in the "*Detail*" in Figure 1.b. This will make the page simpler and will become a better **representation**. After removing extraneous information, there is more space for QR code button.

Redesign QR Code Ticket (See Figure 4)

In the original version, the QR code ticket displays the type of the ticket (said, "One Way" in Figure 1.b) and the stations that the train bound from and bound for. This is a good representation because of no extraneous information on it. However, here it still could be improved on the aspect of the **gulf of evaluation**. The information here is too concise to precisely evaluate whether the QR code is exactly the train I tap on. Therefore, I remove the "*One Way*" because I think it is not helping a user to evaluate from the previous page (ticket list). Instead, I put the date, the time that the train bounds and arrives, and the car and seat information. These will not make the screen too crowded but make the information be **consistent** with the yellow e-ticket in the previous page.

Considering the **context** of passing the turnstile, a user might need to tune up the screen brightness manually to make the QR code be readable by the QR code reader

on the turnstile. To leverage the principle of **simplicity**, it could be designed that the brightness will automatically turn to the brightest in the QR code ticket, and turn to the original setting when leaving QR code ticket page. Further, when passing the turnstile, a user might want to check the car and seat information, so in my opinion, putting such information here makes the user **easier** to use, without one more click to back to the e-ticket page or go to the detail page.

Redesign Booking (See Figure 5)

Obviously, I have so many opinions on the train searching page of the booking process. First is the button design. The button style is not **consistent** through the app and here, the "*Search*" button might not be perceived as a button at all because its font color is grey and located beside the page title (see Figure 5.a). Either add a small arrow "*Search >*" (like the "*< Back*" in Figure 1.b) or replace the text with the button (like "*Details*" in Figure 1.b). Keep the button styles be consistent will largely help shorten users' **learning curve**. This also improves the **gulf of execution** because the user would be easier to identify the action of search available trains.

For outbound time setting, the date and time selection panel must be **consistent** with the field in the form ("*2018/11/03*" versus "*Sat Nov 3*"). The date selection changes into the same as date picking of adding an event in Google Calendar so that the user will see "*2018/11*" in the panel title plus a date to tap to select (see Figure 5.b). The calendar view selection leverages the **mapping** principle, mapping to the real calendar in month view. After tapping the date, the date selection is completed and the date selection panel will automatically close. Therefore, the "*Done*" button is not necessary anymore and could be removed. This change also makes the date selection not only consistent with the upper part of the page but also **easier** and **simpler** than before. Moreover, it is easy to put a **constraint** in the calendar view. For those dates should be select, just need to grey those dates and deny to tap.

Third, I notice that in the search result, the bound time is not an important role in search requirement. Therefore, in the time selection, I redesign that only the hour is required to be picked and the minutes is not necessary, making the time selection **simpler**. Moreover, to make the time display in the time selection panel be **consistent** with the search form, I would like to change the time setting into the 24-hour clock. The scrolling method will not be changed because it is a good design

that leverages the mapping principle. However, the "*Done*" button will be unavoidable if I keep using the scrolling method in time selection panel.

Redesign Search Result (See Figure 6)

The last redesign is a little bit hard to mock-up. The current search result uses 2 buttons ("*Earlier*" and "*Later*" in Figure 2.b) to enable users to view available trains. This is an okay design but could be more intuitively, said making it be able to **directly manipulate**. The structure of the search result is lots of tiles just the same as the posts on the facebook wall. Therefore, I would like to let users **map** it with the experience on the facebook. When they scroll down to the bottom, the later trains will be loaded, vice versa. In this case, the app doesn't need to load every available train at the first time, and the bound time just offers a rough starting point for a user to select proper trains.

Moreover, adding this **affordance** doesn't mean the user knows that. A indication to let users identify such action is essential; that is, leveraging **discoverability** of heuristic principle to narrow the **gulf of execution**. Therefore, in the redesign, I put an icon of swiping up and down (see Figure 6) and making it flash until the user scrolls the search result.