

SFWR ENG 4HC3

Assignment 2 – Parking Meter

Group Members:

Kevin Carmona-Murphy	1059136
Kemal Ahmed	1150377
Anmol Sharma	1325523
Khalil Rahin	1048577

URL: <https://goatandsheep.github.io/ParkingMeter/>

We chose to do a design that simulates a real parking machine, while keeping in mind that it would be a touch screen system. As a group we agreed that having multiple pages would make it optimal for usability. In the home page that a user first sees, we have the option of selecting either a new ticket or returning a ticket that has time left on it for refund.

It is important to note that in the refund page we are simulating the action of scanning a ticket and the machine provides a refund based on how much time is left. For our design we chose a minimum of a half hour selection, and increments of half hour to make it easy for the user.

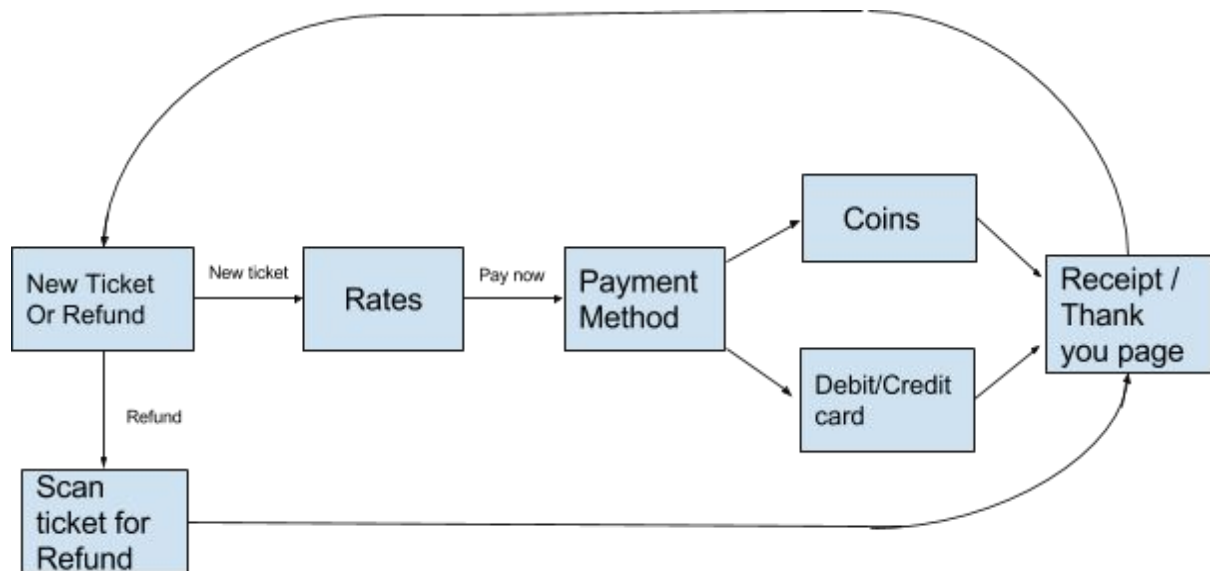
Another important design decision to note is that we chose to have a cancel button on all pages (highlighted in red giving meaning to the button). Pressing the cancel button goes back directly to the main page. This is was important in achieving usability in case a user wants to go start over or simply exit the transaction.

Aside from the cancel button, we also have a back button on each page after the 1st page, which when clicked it goes to the last page. This is because if a user simply wants to go back a page instead of pressing cancel and starting all over again. It allows quick and easy transition from pages and lets users make decisions to save time.

[Back button goes to the last page, Cancel button cancels the transaction]

Upon confirming your refund in the refund page, the system gives a couple seconds to simulate the collection of returned change, and then goes to a “Thank you” page for a couple seconds, then back to the main page.

Below is a flow diagram of our system design showing the different pages we have.



In the Rates page we chose to have max 5 hours to select if user wants hourly rate since our flat rate is \$10 for the entire day and after 5 hours (which would be more than \$10 per hourly) wouldn't make sense. So if they want more than 5 hours, they can just select the flat fee. The cost automatically gets updated based on what you select.

For the payment method of coins we simulated the process of actually inserting a coin into the machine and you can select the kind of coin to insert. Inserting more money than owed results in the machine giving you the appropriate change back and displaying it in the receipt how much you paid and how much you're owed. Also displayed in the receipt is the date and ticket number.

For the payment method of debit/credit card we simulated the process of inserting a debit/credit card, entering your pin and pressing ok to pay. A receipt is also "printed" after. This is represented in the smaller window to the right of the main screen. This displays all context specific hardware interfaces such as inserting coins, inserting a credit/debit card, or scanning/printing a receipt.

The decision to abstract the aforementioned external interfaces to a secondary screen was made to ensure the simulation is more realistic, since coins and cards are never inserted into the main interface of a parking machine. We chose to leave it blank even when no context specific hardware was necessary because this shows the user running the simulation that information will eventually appear in that space.

In general when creating our buttons and overall layout/style, we took visibility, ease of use, and the norman principles into consideration. We made sure it is consistent page to page, button to button, layout and styles are simple and easy to read or see. Overall functionality with great usability was the main focus on our design.