

Garage Automation

Group 10:

Subteam 1: Guy Rubinstein, Arthur Rafal

Subteam 2: Gao Pan, Kendric Postrero, Vatsal Pandya

Subteam 3: Peter Luo, Yunqi Shen

User Application

We created the user interface for our garage automation system.

Here, the user is able to register for an account with a payment method or login to an existing account. Once logged in the user is able to view their history of parking sessions, check which vehicles are registered to their account, and add or remove payment methods.

We extended the built in Django user model to create a custom account model that fit our needs.

We tested the interface we created by exhausting every possible use case that was addressed. This meant that every single functionality of the website was checked under all possible conditions.

The use cases that were addressed by the user interface include:

- UC-1 Registering for an account
- UC-5 Adding/Removing payment methods to/from the account
- UC- 8 Viewing past parking sessions
- UC-3 Remove vehicle from account

Parking Lot Monitoring

We created the system for monitoring the cars that are currently inside the parking lot. We set up the use case for cars to enter and exit the parking lot.

Use Case: Entering

- Registered User: A registered User must have signed up for an account to have the ease of entering getting through without having to input a form of payment at the entrance turnstile. Camera takes a photo and the license plate analyzer software outputs your license plate and inputs it into our system.
- Guest User (w/ Credit Card): A guest user can also have the benefit of ease of exiting, but they are at a disadvantage from registered user because they have to swipe their credit card at the entrance turnstile. Our system still takes a photo of their license plate and adds them into the system as a guest user to later use when they exit.
- Guest User (w/ Cash Only): A guest user who only has cash has the biggest disadvantage in our system because they have to use the classic method of taking a ticket and paying the ticket at a ticket machine.

Use Case: Exiting

- Registered User: A registered user has all of their information saved in our system beforehand so at the exit turnstile they just drive and the turnstile will let them out easily.
- Guest User (w/ Credit Card): A guest user also has their license plate and payment information saved so at the exit turnstile they will just drive and the turnstile will also let them out easily.
- Guest User (w/ Cash Only): This user with a ticket has the biggest inconvenience because they have to hand in their paid ticket for the turnstile to let them out.

Management Dashboard

The Manager of the garage logs in and view the dashboard starting on the second page.

Use Cases:

- View profits of the current day and graphs of **flow vs. time** -- Matplotlib
- Manually **change price** settings and **disable** certain parking lots in case of malfunctions of hardwares. -- Django Admin System
- View **Transactions** and **Payment History** (Search with given name/license plate number/transactionID/range of date and time) along with a calendar, which are on the third page -- SQLite

Test:

```
$ python manage.py test manager
```

Future Additions

1. We plan on integrating the systems so that the license plate is able to recognize and process the correct User Type automatically.
2. For the maps, we plan to provide real time updates to the spot occupancy by using channels (websockets) in Django.
3. We also plan on creating a mobile responsive version of the occupancy map to display to the end-user interface.
4. We will provide a better UI for the manager-side and work to add the features we described in our proposal.
5. Using the Django framework, we will continue to add unit and integration tests, using the testing framework.