Final Report: ThunderCats Car Rental

**Name:**

Thundercats Vehicle Rental Service

**Github Link:** [ThunderCats Repo](https://github.com/CatBeam/ThunderCatsMS4)

**Group Members:**

Matt Downs

Sam Nugent

Ben Rahman

Zach Theiss

**Instructions:**

Navigate to the ThunderCats Repository on Github. Located in the repository directory are three folders and a copy of this report.

Navigate to Heuristic Evaluations:

-Heuristic Evaluations for both the Manager and Receptionists can be located by opening the “HeuristicEvals” folder in the main repository directory.

Navigate to Manager Artifacts:

-The Manager Site HTML pages can be located by opening the “ManagersSite” folder in the main repository.

The following list items are Page Name and HTML Page pairings for locating Manager Site Artifacts. They are listed in the same order as the repository folder.

* car\_damage : Car Damages
* car\_info : Mustang GT Details Page
* car\_search : Mustang GT Results Page
* check-in\_screen : Car Check-in
* check-out\_screen : Car Check-out
* home\_login : Manager Login Portal
* manager\_request\_page : Manager’s Inventory Request Form
* manager\_search : Manager’s Map-Assisted Search
* navigation\_home\_screen : Manager Home Page

Navigate to Receptionist Artifacts:

-The Receptionist Site HTML pages can be located by opening the “ReceptionistSite” folder in the main repository.

The following list items are Page Name and HTML Page pairings for locating Receptionist Site Artifacts. They are listed in the same order as the repository folder.

* confirm-pickup : Confirm Customer Rental Pickup Page
* index : Receptionist Home Page
* walk-in : Receptionist Walk-In Customer Form

**Description:**

The application will be one that the entire company can use across all of its locations to allow for a computerized record keeping of their rentals. This can be used in a manner to share cars across stores and also keep better track of where a vehicle might be, its condition, when it will be returned, its rental history, and how much it is expected to cost. This application will be used purely by the company and no customer will have access to it, thus making it an internal system.

**Problem Statement & Intended Users:**

Currently it is difficult to track the vehicles histories, locations, and rental dates, and have it in a nice format for the receptionist to use, along with managers to show how the rentals are going. A new system is required that will be easy to manage, easy to book, and can spit out results efficiently and accurately.

The users will be two main types, receptionists and managers. Diving in deeper to these two, the receptionist will be an individual with minimal computer knowledge and will not be versed in keyboard shortcuts nor command line interface (CLI), this will drive a strong graphic user interface and well developed user interfaces along with user experiences. The system needs to be as intuitive as possible for the limited experience, and the depth of accessing different aspects of the system should be limited (ie, only a few clicks to get a car and all its details).

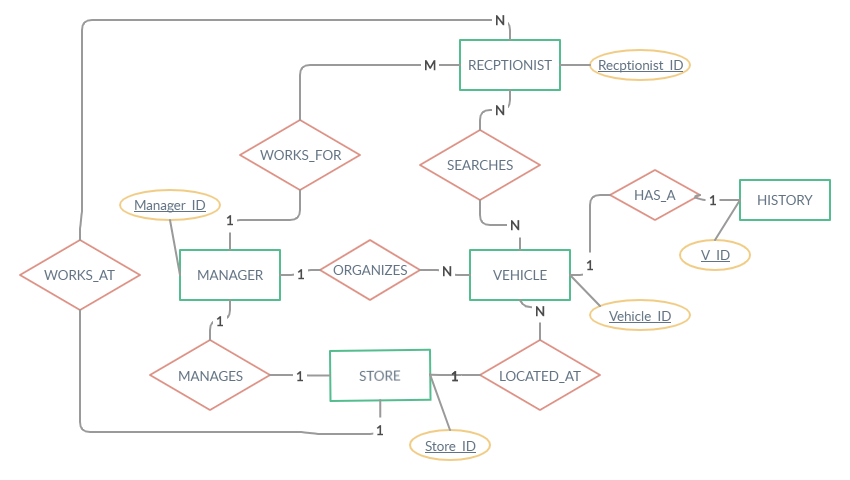
The second main user, managers, will have a bit more computer knowledge, but their controls will be different; instead of renting out cars, they handle the movement, tracking, analytics, and backloging. Managers will be knowledgeable on information pertaining to tasks similar to those in Microsoft Office products (graph creation, data manipulation, etc.) but there should still be a strong emphasis on user interface creation and still not using CLI and very limited keyboard shortcuts (Ctrl+S, Ctrl+C, etc.).

**Summary:**

With having two different types of people we are working with, this will require a second “area” of the program, one that focuses on customer inquiries and another than will revolve around managing the database. The receptionist can focus on booking customers and having that information available to them, as well as all the information on a vehicle (current booking, location, history, etc.). The manager will focus on the “back-end” of the database and will bring together different analytical components along with the “managerial” side of the business, such as, adding new vehicles, removing vehicles, changing locations, and management of renting and how specific vehicles can be rented (deposit needed, insurance required, etc.).

There are other aspects to the system that are required, such as, reminders for return, email invoices, tracking peek locations and times for vehicles, recommended requirements for returns (tank level), graphic information about the damage to vehicles, and photo storage of the vehicle. All of these can be helpful to the receptionist as it will make it easier to show multiple vehicles and the specific information about each. Managers will also be happy to have this system for them, as it will expand their knowledge of the corporation and allow for thoughtful and detailed reasoning behind actions in the company. Rather than blind-firing and attempting to come up with solutions based on what they notice, opening it to a more mathematical and calculated reasoning will help make a true solution.

**Data Model**



Entity Attributes List:

**Receptionist** - Receptionist\_ID (Primary), Name, M\_ID, S\_ID, R\_Login\_Num

**Manager** - Manager\_ID (Primary), Name, S\_ID, Employee\_Num, M\_Login\_Num

**Vehicle** - Vehicle\_ID (Primary), Make, Model, Year, Color, Condition, Rental\_Start, Rental\_End, Mileage, Fuel Level, S\_ID, isAvaliable

**History**- V\_ID (FK), Total Rentals, Rental\_Start, Rental\_End, Return Condition, Start\_Mileage, End\_Mileage, Start\_Fuel, End\_Fuel

**Store** - Store\_ID (Primary), Address, City, State/Province, Country, Postal\_Code,

**Tasks:**

1. **Customer**

T1.1 Rent Car

T1.1.1 Search for preferred car

T1.1.1.1 Can be done by specifics of car (Year, Model, etc.)

T1.1.2 Validate license

T1.1.3 Take information about customer and record

T1.1.3.1 Check if already in system

T1.1.4 Create rent information (Specifics)

T1.1.5 Take deposit

T1.2 Edit Rental

T1.2.1 Search for current/future rental

T1.2.1.1 Cancel

T1.2.1.2 Extend

T1.2.1.3 Move dates

T1.2.1.4 Change information

T1.2.1.5 Change Cars

T1.2.1.6 Prepay

1. **Receptionist**

T2.1 Check Car Out

T2.1.1 Search for existing booking

T2.1.2 Find car for customer

T2.1.3 Record customer information

T2.1.4 Match database car information to physical car (Mileage, damage, etc.)

T2.1.5 Mark car as rented

T2.1.6 Take/Verify deposit

T2.1.7 Give key

T2.2 Return Car

T2.2.1 Search for rental (customer or number)

T2.2.2 Verify vehicle

T2.2.2.1 Mileage

T2.2.2.2 Damage

T2.2.2.3 Gas

T2.2.3 Input in system

T2.2.3.1 Creates Cost

T2.2.4 Charge customer

T2.2.5 Get Key back

T2.3 Identify Car History

T2.3.1 Look by model, make, year, or history

T2.3.2 Load in the car

T2.3.3 Adjust current view to history

1. **Manager**

T3.1 Move Car

T3.1.1 Identify car to change locations

T3.1.2 Access database

T3.1.3 Search for car

T3.1.4 Set location to new location

T3.2 Create Analytics

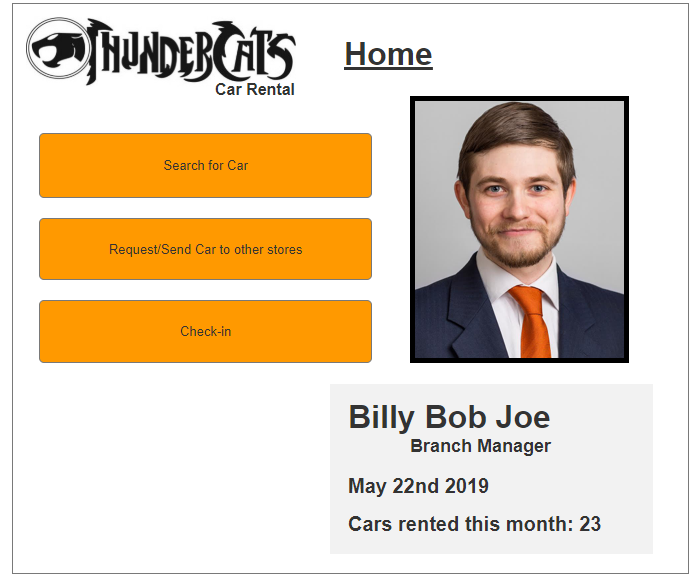
T3.2.1 Search for car or location

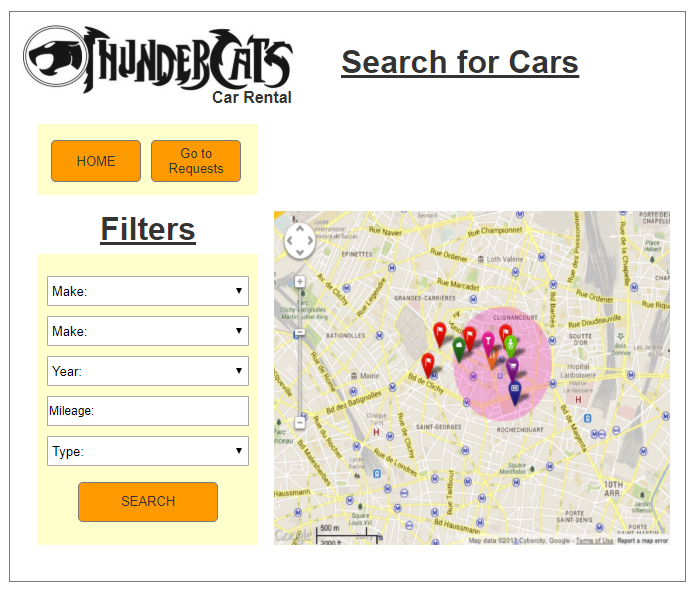
T3.2.2 Choose the type of analytics (Graph, Excel, etc.)

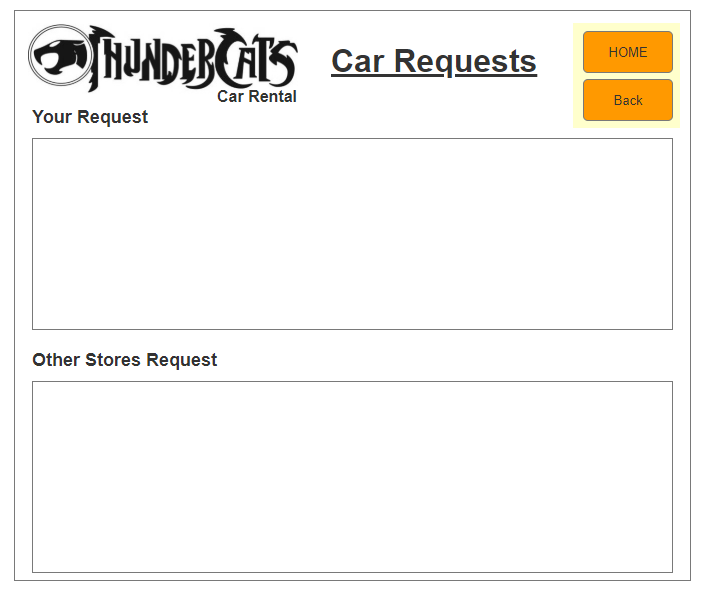
T3.2.3 Output

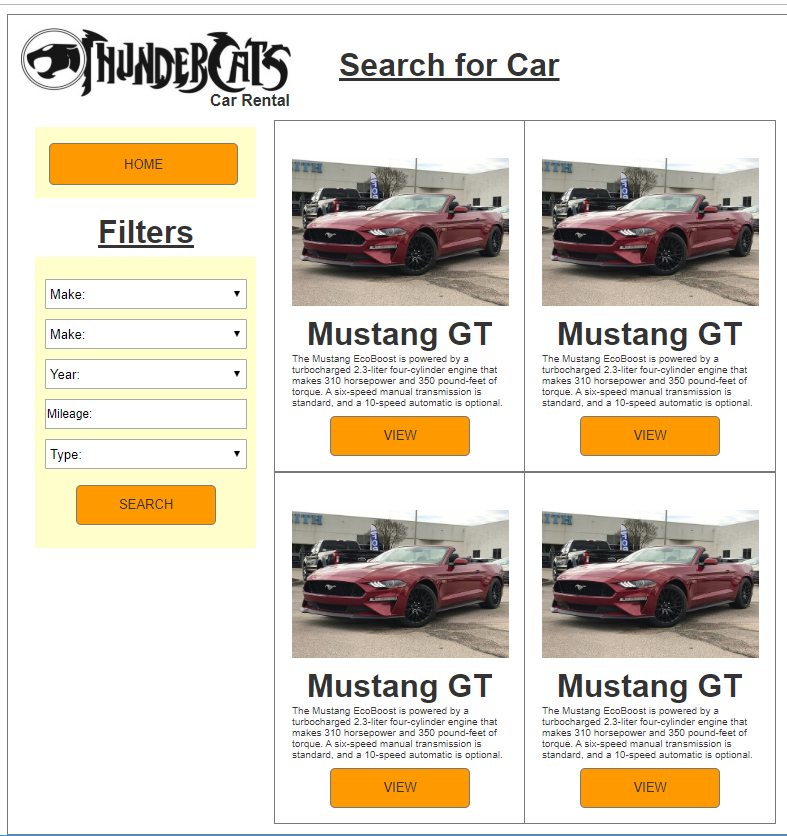
**Manager Site Screenshots**

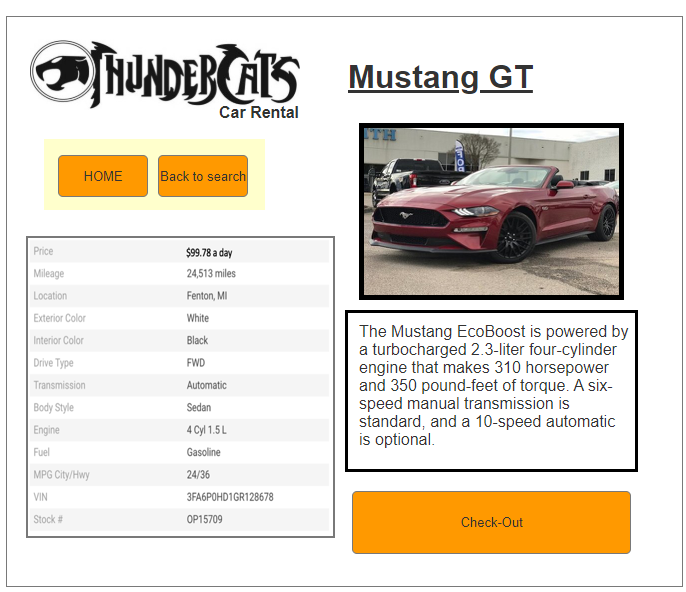
**A screenshot of a cell phone

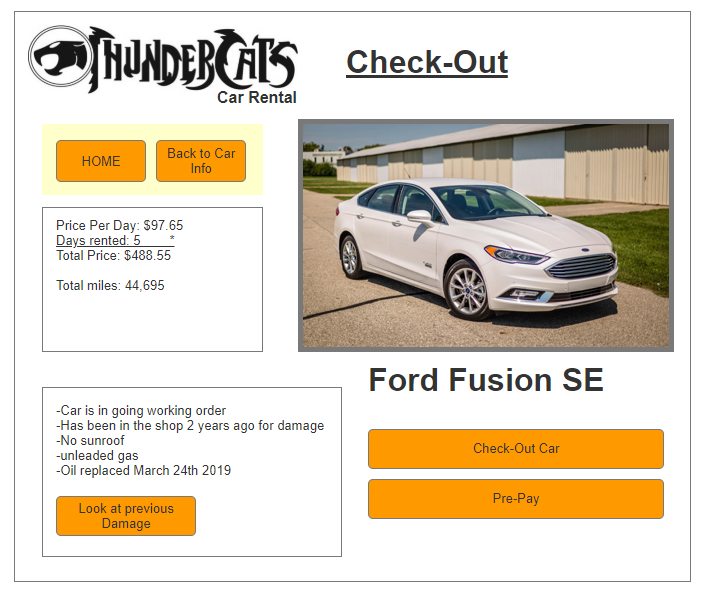
Description automatically generated**

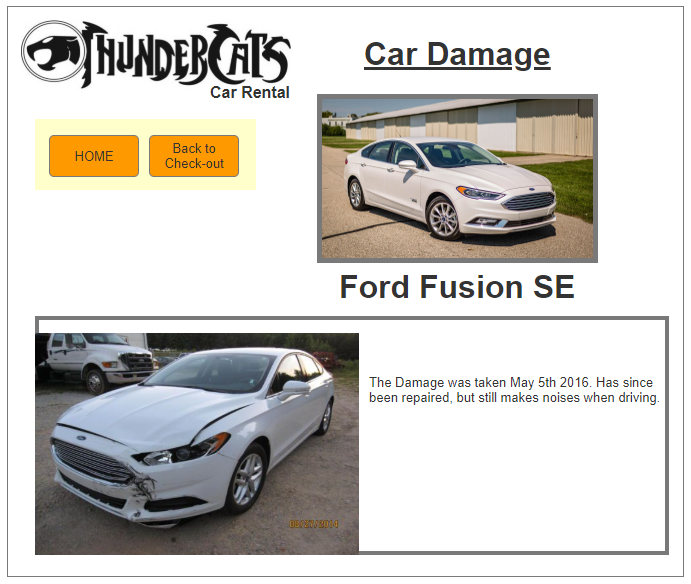




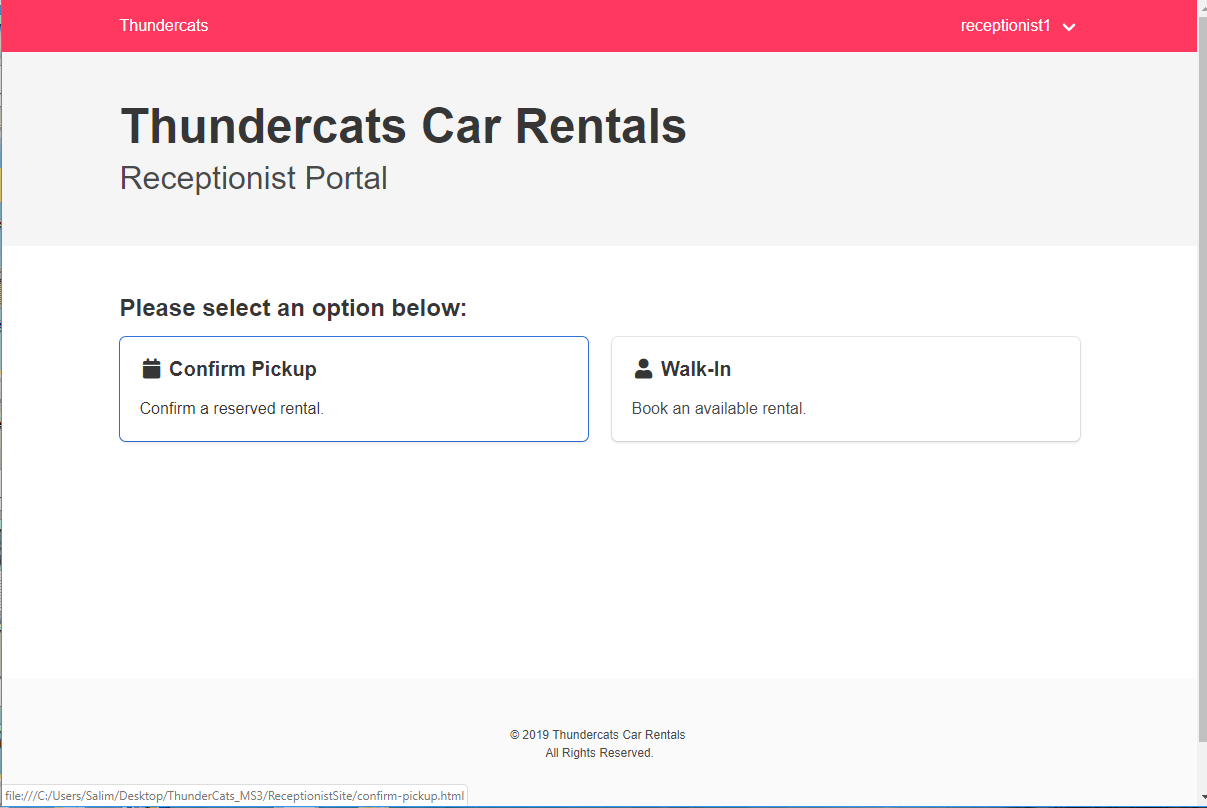


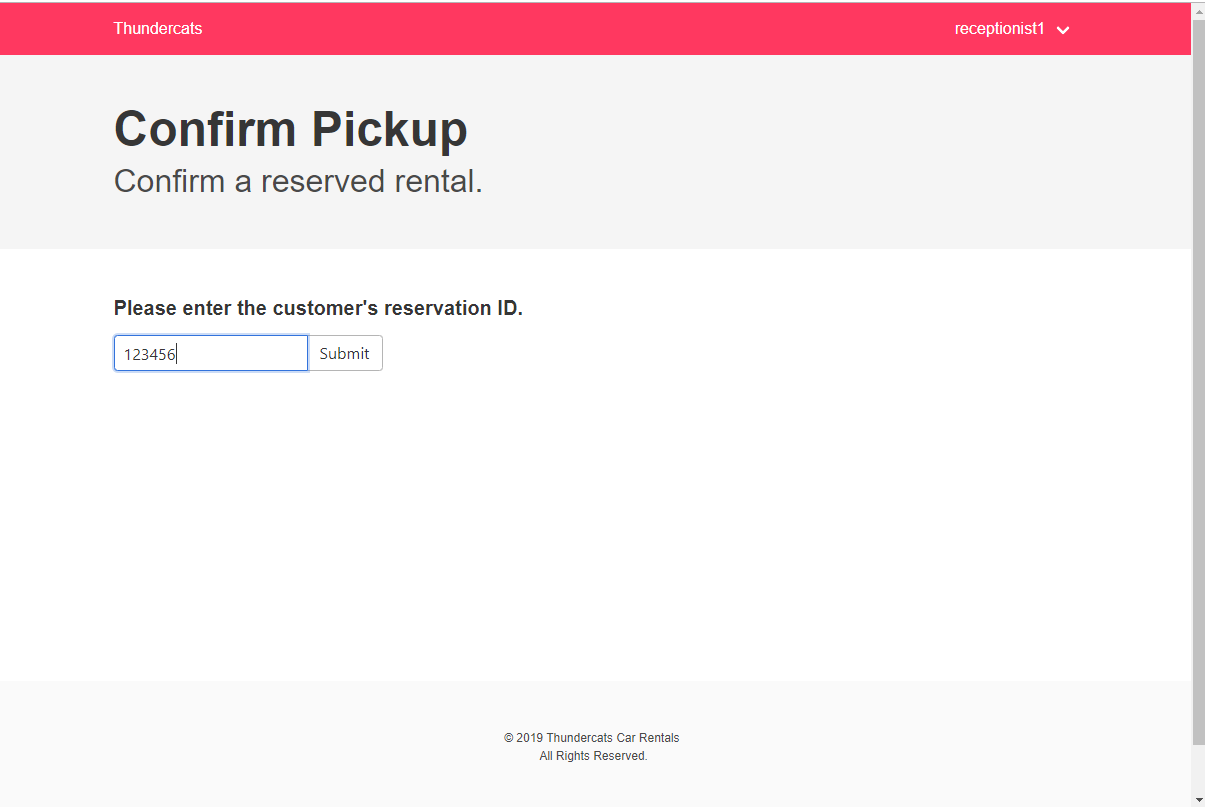


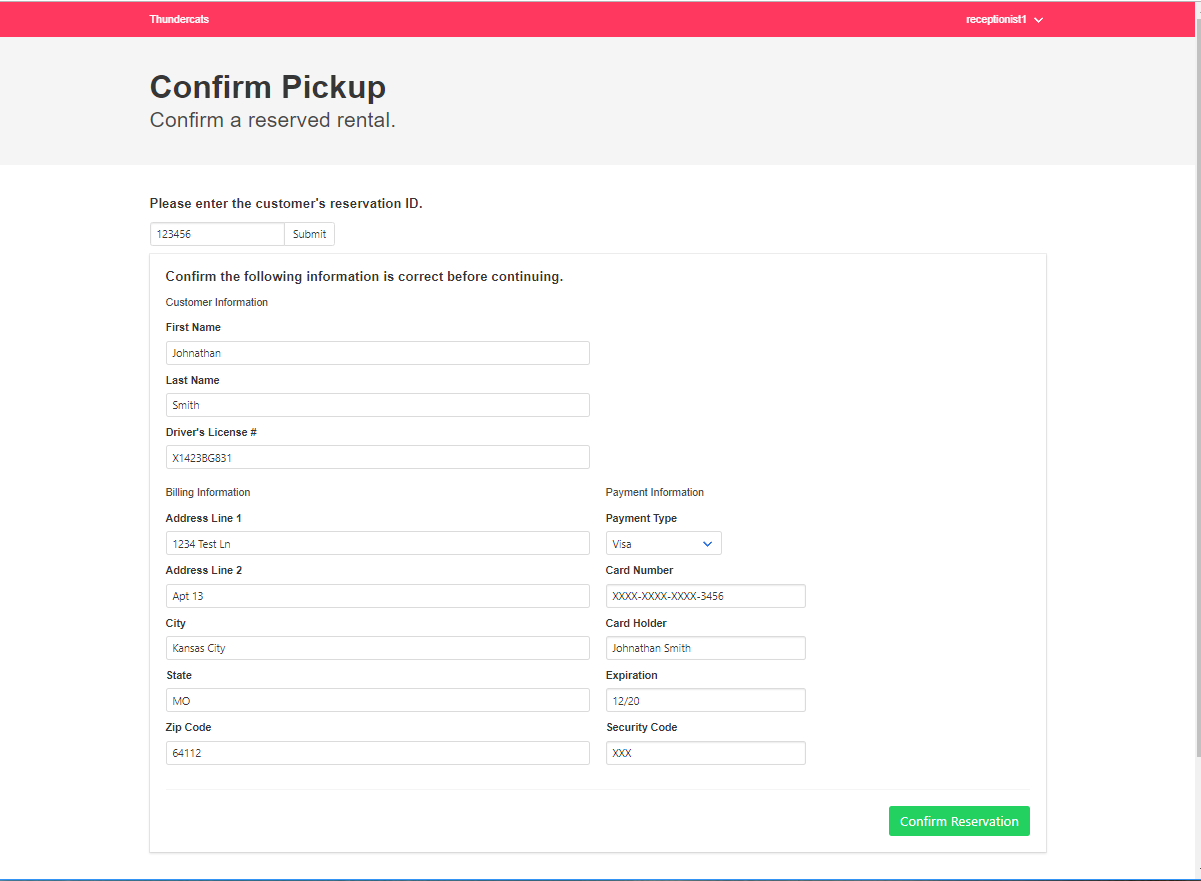


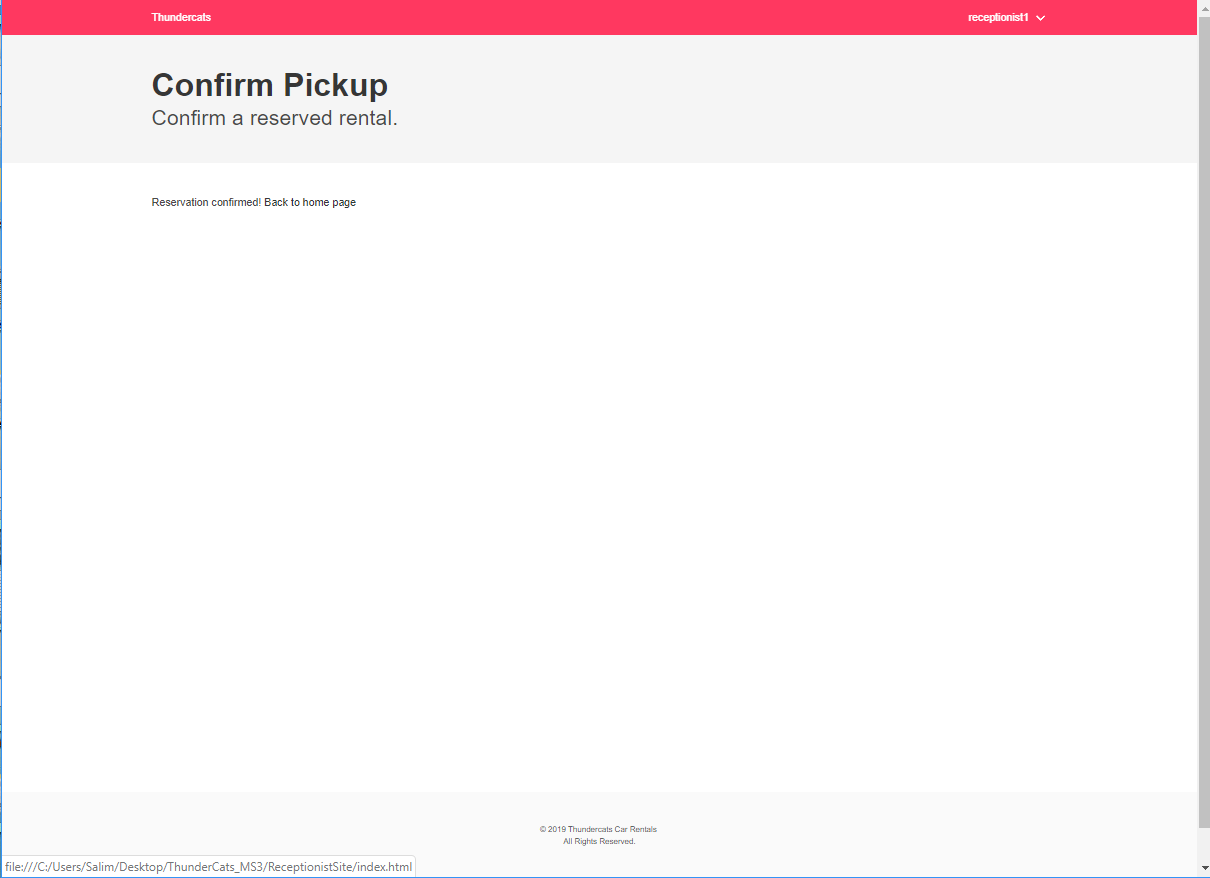


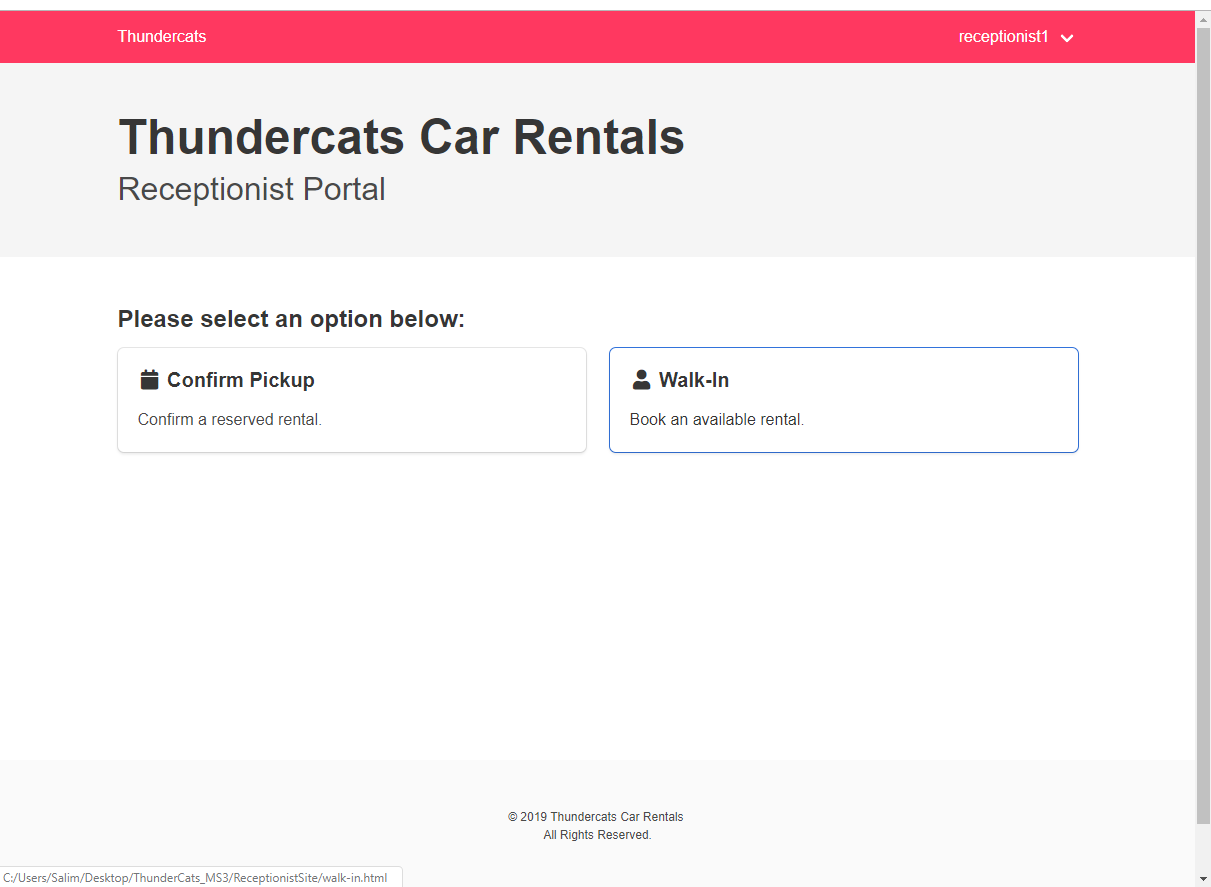
**Receptionist Site Screenshots**

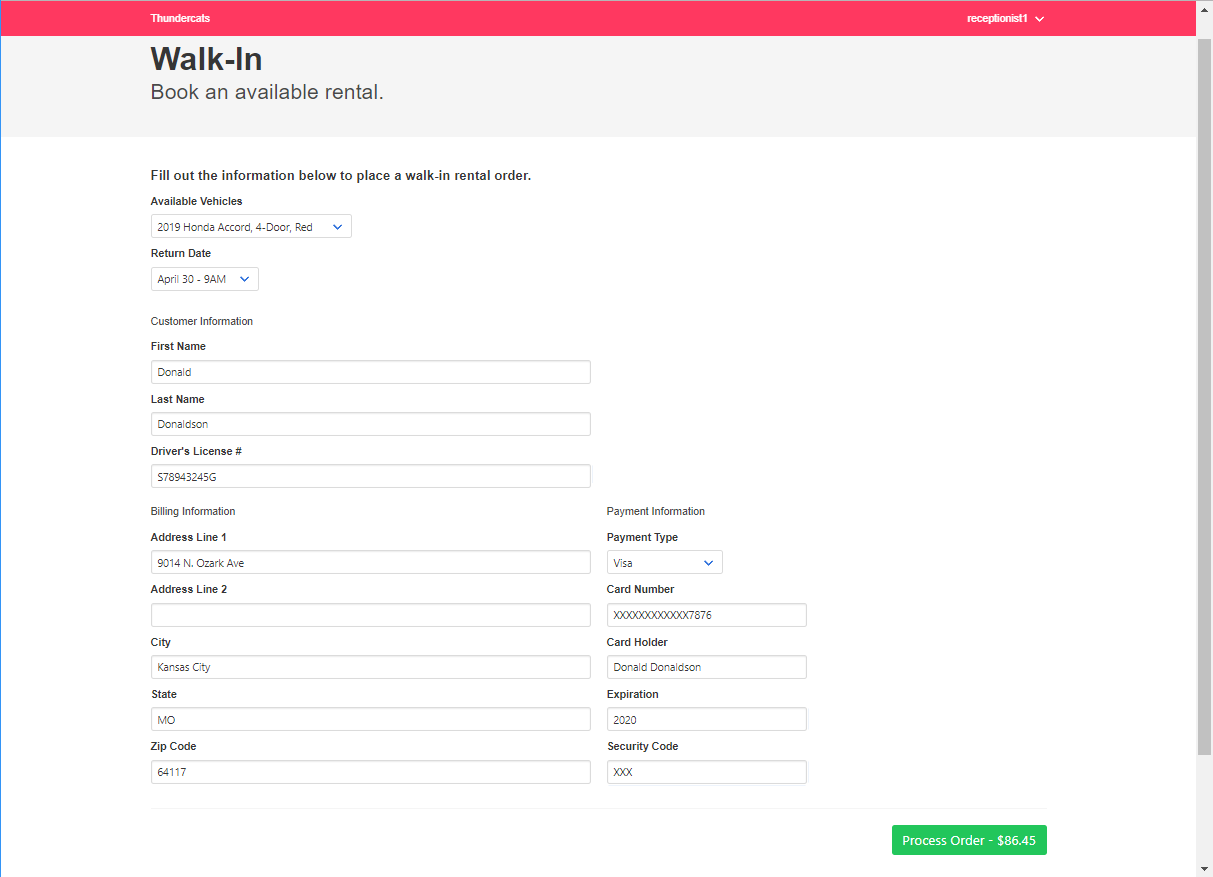












**Manager Site Heuristics**

Using the Neilson Norman Group's 10 Usability Heuristics for UI Design

Visibility of System Status

In terms of providing the user with situation-appropriate feedback, the Thundercats Car Rental Service Manager site follows all conventions outlined by this heuristic. Although there are no expected wait times over 0.5 seconds, the site’s buttons indicate their use by changing the cursor icon to the widely recognized Hand Cursor icon.

Match Between System and the Real World

While there are not instances of industry-specific jargon, the language found on the manager's site closely resembles language that would be used in real-world situations. In addition, the freedom given to the user allows for managers to quickly exit tasks and redirect to another. This flexibility closely resembles real-world situations where many customers must be serviced abruptly.

User Control and Freedom

In interacting with the system, the user is given a sufficient level of control and freedom. To elaborate, unlike the receptionist site, the manager site allows users to start a task but does not require them to finish that task before starting another. At any point, while completing a task, the user can exit from that task and direct to the Home page, or to another task.

Consistency and standards

The system has a consistent structure throughout the site. The data presentation is presented in a reasonable format, however, some pages lack summary or overview data that may be useful to the user in a variety of situations. For example, in the "Check-In Screen" window, the user is given the model, and brand but not given other relevant summary data like the year; or the option to view preexisting damage.

In terms of standard conventions, the system follows UI conventions with two exceptions. One, throughout the site the title of the page is present but underlined. Which indicates a false affordance to the user by seeming clickable. Two, the text box located in the “Car Info” window is given a heavy border. Which may indicate to the user that this field is editable since it does not match the layout of similar data displays.

Error Prevention

There is some error prevention built into the Manager system, notably the input restrictions when filtering vehicles. However, the website does not offer any other significant built-in error prevention measures such as tooltips.

Recognition rather than Recall

The Manager system leverages recognition by directing users towards their desired task on the Home page. Users should be able to quickly recognize the option they want. However, this recognition could be further improved by using icons in appropriate places,

Flexibility and Efficiency of Use

In terms of flexibility, the system is friendly to both the novice user and the advanced. Due to the restricted user controls, any new user can be easily trained to learn the system. One deficiency that should be noted is that the user sometimes must continue down the predefined path in order to view certain data that relates to an entity on another page. For example, if the user wants to search for a certain vehicle and view its damage report, they take the following path: Home Page -> Invoke(Search for Car) -> Search for Car Page -> Invoke(View) -> Car Info Page -> Invoke(Check-Out) -> Check-Out Page -> Invoke(Look at previous Damage).

In terms of efficiency, the system allows the efficient execution of tasks in most cases. It accomplishes this by having the user select their task at the Home page, taking the user step by step through their selected task, and after completion, either loops the user back to the Index page to select the next task or allows a direct redirect to related tasks. However, it should be noted that the system lacks shortcuts, and other time-saving actions that are attractive to the advanced user.

Aesthetic and Minimalist Design

The Manager site uses its design effectively, presenting the user only with critical information needed to complete their task. In addition, to keep tasks restricted to their logical flow, the user is only prompted to enter data that is critical to completing that task’s current subtask.

Help Users Recognize, Diagnose, and Recover from Errors

Error recognition and recovery are two major aspects currently lacking from the system. Alongside the error prevention problems, there are no currently available functions the user can exercise to recover from erroneously inputted data.

Help and Documentation

In addition to tooltips, the system could also benefit from further documentation. Whether it is in the form of explicit instructions or a User guide. While the system is easy enough to master for most users, there are some cases where a trainee or other brand-new user could benefit from additional instructions.

**Receptionist Site Heuristics**

Using the Neilson Norman Group's 10 Usability Heuristics for UI Design

Visibility of System Status

In terms of providing the user with situation-appropriate feedback, the Thundercats Car Rental Service Receptionist site follows all conventions outlined by this heuristic. Although there are no expected wait times over 0.5 seconds, the site buttons indicate their selection through color change and indicate selection when clicked.

Match Between System and the Real World

While there are not instances of industry-specific jargon, the language throughout the website clearly states the intent of each performable action and records data in a logical flow that mirrors similar real-world tasks.

User Control and Freedom

There is a limited amount of control given to the user. Although the user specifies the order of operations by choosing either "Walk-In" or "Confirm Pickup" in the Receptionist Portal, the rest of the site restricts the user's choices. To elaborate, there are no reversible actions in the system. Thus, when the user inputs and then confirms a data entry or invokes a function, that action cannot be undone.

Consistency and standards

The website offers consistent structure and data presentation throughout the site. In addition to its consistent structure, the website also follows standard UI conventions commonly found in other websites. For example, when filling out customer data, the user should be able to leverage their recognition of the template to quickly enter the applicable data.

One area of concern that should be noted is that while each entity’s data is encapsulated in its own region, the space between these regions is significantly smaller than it should be.

Error Prevention

There is some error prevention built into the Thundercats Rental system, notably the input restrictions when entering payment information. However, the website does not offer any other significant built-in error prevention measures such as tooltips. That being said, the presentation of data in a logical flow that is similar to the real-world tasks does help eliminate several errors.

Recognition rather than Recall

The system leverages user recognition by presenting data in a consistent and organized manner. Coupled with the data presentation, the flow of data presentation is both similar enough to allow users to quickly recognized the information that needs to be entered and differs enough to allow users to identify their location on the site.

Flexibility and Efficiency of Use

In terms of flexibility, the system is friendly to both the novice user and the advanced. Due to the restricted user controls, any new user can be easily trained to learn the system.

In terms of efficiency, the system allows the efficient execution of tasks. It accomplishes this by having the user select their task at the Index page, taking the user step by step through their selected task, and after completion, loops the user back to the Index page to select the next task. However, for the advanced user, the system lacks shortcuts, and additional time-saving features.

Aesthetic and Minimalist Design

The system uses its design effectively by presenting the user only with critical information or feedback needed to complete their task. In addition, to keep tasks restricted to their logical flow, the user is only prompted to enter data that is critical to completing that particular task’s current subtask.

Help Users Recognize, Diagnose, and Recover from Errors

Error recognition and recovery are two major aspects currently lacking from the system. Alongside the error prevention problems, there are no currently available functions the user can exercise to recover from erroneously inputted data.

Help and Documentation

Aside from lacking tooltips, the system does an adequate job in instructing the user’s current action by leaving explicit instructions in the webpage. Coupling this with the ease of use and ease of learning for users, additional documentation besides tooltips may be overkill and add unneeded confusion to the system.

**Limitations**

The primary limitation of our Car Rental service sites is that they lack a database to read and write data too. As well as sample data to pull from the database. In addition to the database limitation, when planning Managers’ site originally, we had a map feature planned that would display locations. However, because we do not have a database, integrating the real-time map into our manager site unrealistic.