

By Cameron Cobb

Compared to the previous version of our software architecture diagram, numerous improvements and additions have been made to adjust and reflect our data management strategy. First, we added an additional database to store manager information such as manager name, username, password, unique ID, work schedule, and work location. This will be useful when we create relationships between various tables of data (like creating a relationship between users and managers) in such a way that will follow data normalization, where we will specifically isolate independent multiple relationships since many managers can help and assist many customers/users. We will discuss the various strategies for this in the tradeoff discussion below. The second improvement we made to the diagram was adding two additional pages: one page that stores the directory of cars available and another that serves as a transaction/payment page where users can edit/add payment methods and check out selected vehicles in their cart. We also added more use cases in the web pages such as an option to return a car, get directions to the nearest facility where the selected car is stored, view shopping cart, and updating user log in information. Finally, we distinguished between interactions between the webpages and the interactions with the databases: the interactions and flow of the website were signified with solid arrows while interactions with the databases and other 3rd party apps/systems were illustrated in dashed arrows. This makes it easier to visualize the relationships between the different users, use cases, and elements of our website application. We also attached additional text boxes next to each arrow relationship to show the inputs and outputs of each use case in the SWA diagram as well as how this information is used to edit and/or make additional entries in our databases. For example: searching a vehicle by location first requires a “user location”, which is seen in a text box as an output of the “Search Vehicle By Location” box in the diagram. This location is found by interacting with the 3rd party GPS app connected to the software system. This “user location” is returned and used by the vehicle directory page to filter vehicles from the car rental database based on their nearest location to the user. These vehicles are found by passing “car information” from the vehicle directory page to the car rental database. These vehicles are essentially filtered based on the user location because each car entity in the database is sorted based on their location attribute, where cars with a closer distance to the user’s location will be towards the top of the vehicle directory page and cars farther away from the user will be towards the bottom.