

# Design Journey Part 2

Group name: VAVO

Members' names: Jarrod Ashley, Keshav Varma, Robert Oxer, Zachary Vinegar

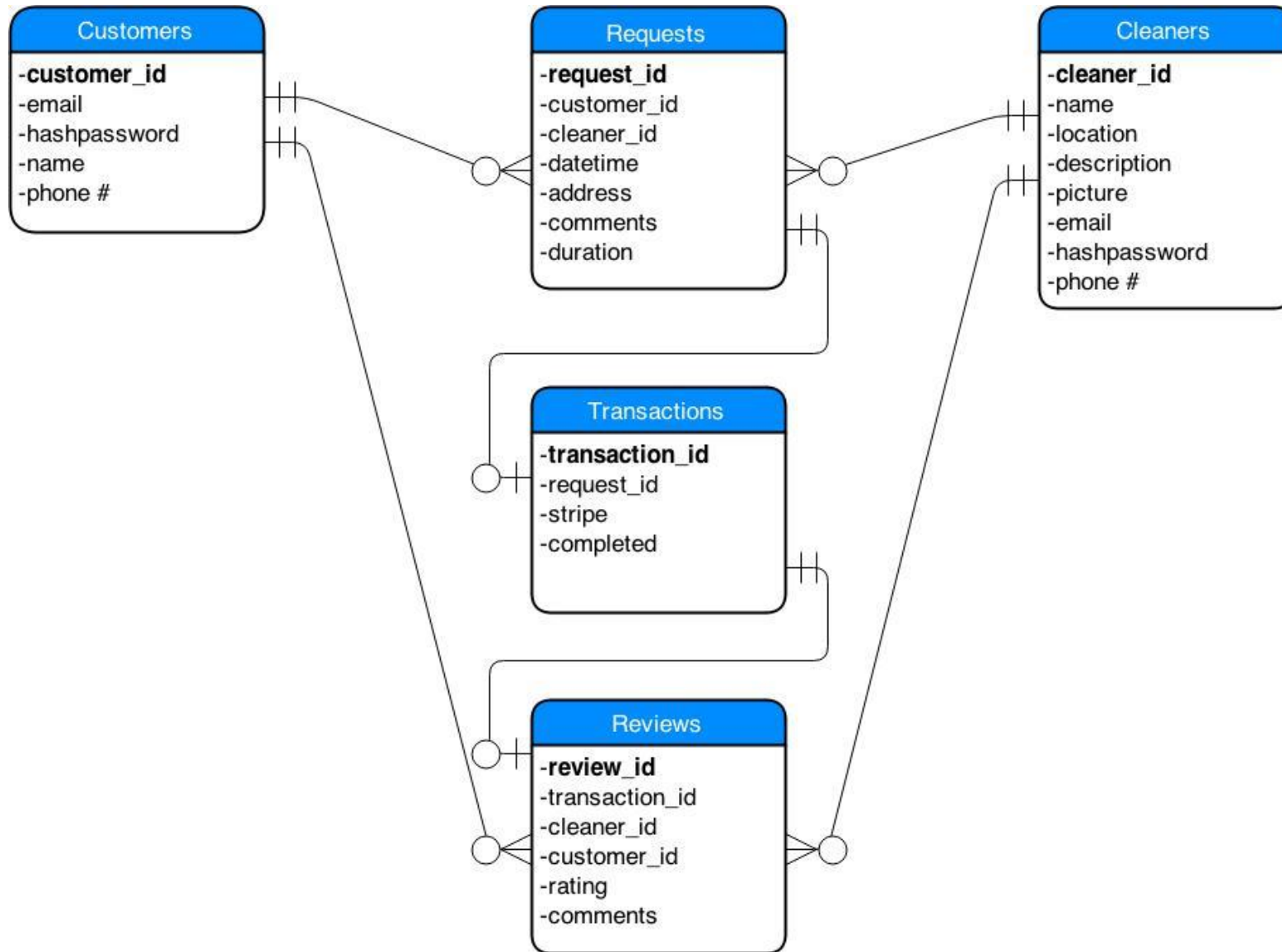
Members' NetIDs: jua4, kdv8, rdo26, zzv2

Section: 201

## **Part 1: Database Design**

### **Conceptual ER Diagram (entities and relationships, see Lectures 17 and 18)**

In this part, please copy and paste your ER diagram for your database below (you can make your ER diagram using any tool of your choice). Make sure the relationships between each entity are clear and well thought-out. Don't forget to indicate what kind of relationship each arrow represents. Your database description should go on the next page.



## Database Description

Tell us what the database does. Make sure that you include enough detail so that we are able to understand what is going on in your ER diagram.

We plan to implement five different tables corresponding to the unique features of our website. We have decided to add a Reviews table as it connects with 3 other tables (Customers, Cleaners, and Transactions). There is a one-to-zero or many relationship between Customers and Requests and between Cleaners and Requests, meaning one customer can make zero or many requests and one cleaners can accept zero or many requests. Additionally, there is a one-to-zero or many relationship between Customers and Reviews and between Cleaners and Reviews, meaning one customer can make zero or many reviews and one cleaner can receive zero or many reviews. The relationship between Requests and Transactions is one-to-zero or one, meaning one request will either result in one transaction or no transaction because a cleaner can deny a request. The relationship between Transactions and Reviews is one-to-zero or one, meaning one transaction will either result in one review or no review at all since customers will have the option of not completing a review.

## Part 2: Website Layout

### Content Organization

This should be an improvement upon the table you used in **Design Journey Part 1**

<b>Main navigation</b> (List your site's navigation here)	<b>Sub category</b> (sub categories)	<b>Content</b> (List all the content corresponding to main navigation and sub categories)
<b>Request Cleaner</b>  Allows the user to find a nearby cleaner and schedule a time for the cleaning appointment.	<b>Location &amp; Time</b> <b>Cleaner Results</b> <b>Contact Info</b> <b>Payment</b>	<b>Location &amp; Time:</b> form to enter address of house to clean, time to start, and duration; <b>Cleaner Results:</b> list all cleaners available at that day and time, sort by distance; also include history of previously used cleaners; <b>Contact Info:</b> form to enter name, email, password, and phone number; or if they have done this before a login form with just email and password; <b>Payment:</b> form for credit card information;
<b>Rate &amp; Review</b>  After the job has been completed, the customer is allowed to rate the cleaner and provide feedback on the experience.		Form to rate (0-5 stars with .5 increment) a cleaner, and an optional text field to leave a comment about the most recent job
<b>Job Requests</b>  A place for the cleaner to view their profile and find information about their jobs.	<b>Recent Activity</b> <b>Schedule</b>	<b>Recent Activity:</b> recent job requests for logged in cleaner - sorted by date of creation; <b>Schedule:</b> calendar view of upcoming jobs;
<b>My Profile</b>		User can view/change name, email, phone #, profile picture, etc.
<b>How It Works</b>		FAQ's and general information about the reservation process

Help page for more details on the request process as well as payment options and other useful information.		
Register/Login		Where a user can sign up for / login to an account as a customer or as a cleaner

## Navigational Structure

Explain how users will move between pages. What kind of navigational aids will you have? Will there be a menu bar? A drop-down menu? Tabs? Will you have this available across all your pages?

Tell us why you chose a particular navigation scheme over other possible choices, how the overall navigation of your site will work, how the various pages will be linked, and how the the navigation categories make sense from a user's perspective. You may find it helpful to include a diagram of your site map here.

Overall, the high level navigation will be a fixed menu bar with the 3 main links: Request Cleaner, Rate & Review, and How It Works; in addition there will also be a login button on the far right of the menu bar. If the user is logged in as a cleaner, then there will be 3 different links: Job Requests, Ratings & Reviews, and My Profile.

The Request Cleaner section will guide the customer through the appointment process step by step. The 4 steps are: Location & Time, Cleaner Results, Contact Info, and Payment. There will be a standard wizard style display that will allow the user to see where they are in the appointment process (e.g. Step 3/4: Contact Info) and will also let the user return to any previous step and make changes.

We chose this design over other possible choices because we felt it was the simplest way to convey the functionality our website offers. We want to make interaction as simple as possible (in as few clicks as possible) as that will encourage better business.

From a user's perspective, there are 3 main tasks a customer may want to accomplish on our site: finding out how the cleaning appointment works, requesting a cleaner, and rating/reviewing previous cleaners. That is why these 3 tasks correspond to the 3 main links on our menu bar.

Our navigational structure was based off of our client's needs and the requirements of this project. We utilized real-world example websites like homejoy.com, taskrabbit.com, and handy.com to visualize how all the separate functionalities tie together.

### **Part 3: Interactive Functionality**

What interactive features will your site have? What PHP and Javascript elements will you include? Describe how the interactivity meets the needs of the clients/target audience.

Our website fits the client's needs through its interactive elements described in the following two sections. Through our scheduling functions, we meet the client's request that the site allow cleaners and possible clients to schedule a mutually agreeable time for services. Our matching algorithm meets the users' desire to find possible cleaners in the fastest way possible. The notification system allows users to not need to be logged in at all times to make use of the site's services. Our PHP form validation helps cleaners and their clients have the most accurate and usable information on each other. Our interactivity added to the scheduling system through AJAX meets the client's need that all types of people be able to use the site for their cleaning services, because of its simple, self-explanatory usage.

## PHP Interactivity

For each piece of PHP interactivity that you plan to implement, describe what the interaction is, how you will implement it, and which pieces of PHP code are required to complete it. You can describe these in terms of functions if you like, but only if you want to. If there is overlap between PHP and JavaScript interactivity, describe the interaction both here and in the JavaScript Interactivity section on the next page.

- Scheduling functionality:
  - The site will allow cleaners to specify their availability using an interactive calendar that will use Javascript and AJAX to update the database without having them submit a form each time. The PHP code will keep track of these events in the database. This consists mainly of inserting, updating, and removing entries from a table in the database, which we have done in the past.
- Matching algorithm
  - We will be implementing an algorithm to find the best cleaner for a given request based on a few metrics. Per our client's instructions, we will start by verifying cleaner availability, considering the cleaner's location, and finally based on the cleaner's reviews/ratings. We can potentially also factor in the customer's request history in case they prefer a cleaner that they've had an experience with previously. This code requires basic math for the first version, but could be tuned later if we have more time and become more complicated.
- Notifications
  - After a user requests a cleaner, we will send an email notification to the cleaner to allow them to accept and review a request (the email will contain any information provided during the request process).. The email will contain links that allow them to accept or reject a request. If the cleaner accepts, the customer will receive an email notification notifying them that their request is confirmed. Otherwise, they will receive an email that says the cleaner is not available and also provide a link to request a different cleaner. This will require us to use the PHP mail interface and also provide one-time links using hashes/tokens to allow users to click on links within the email. We also want to implement mobile notifications which requires us to send emails to email addresses specified by mobile carriers that will then get sent as an SMS to the customer.
- Request Creation



- Since our user interface is mostly based on AJAX, the PHP code will need to provide an API that will allow us to validate addresses, find available cleaners, and create requests via Javascript. This will require us to handle POST and GET requests along with form data like we have in previous projects.

## JavaScript Interactivity

For each piece of JavaScript interactivity that you plan to implement, describe what the interaction is, how you will implement it, and which pieces of PHP code are required to complete it. You can describe these in terms of functions if you like, but only if you want to. If there is overlap between PHP and JavaScript interactivity, describe the interaction both here and in the PHP Interactivity section on the previous page.

- Scheduling
  - A drag and drop interface similar to Google Calendar will be implemented using FullCalendar and custom event hooks to add and edit the availability schedule for cleaners.
- Interface
  - The biggest use of Javascript will be to make the interface as simple to use for the customers. The cleaner request form will be presented as a wizard with multiple steps. Each step will be processed via AJAX and submitted to a PHP page in the background so that the user doesn't have to navigate between different pages. Javascript will be used to show the different steps of the form and also validate and submit each step to the server.
- Validation
  - We will use Javascript to validate forms (as well as on the server side in case Javascript is turned off). Since we have a lot of interactive forms, there will be a lot of validation required. The implementation will be similar to our past projects.

Compared to the first milestone, did you make any changes to your plan to use the existing libraries (e.g. editor.js, jQuery Cookie, Image Sliders, jQuery) for the site? If so, write down the libraries, what you have to do to incorporate those libraries, and how much of your own code will satisfy the project requirements. If there is no change, write down N/A.

We will probably use FullCalendar to allow us to display a calendar view to the cleaners to specify their availability. The library itself only provides a mechanism for displaying a calendar and provides a few callbacks that are triggered by specific events. We will still need to write a lot of code to be able to create and edit events since the library does not support this functionality natively.

#### **Part 4: Additional Comments**

If you feel like you haven't fully explained your design choices, or you want to explain specific functions in detail, do so here. You can use this space to justify your design choices or ask other questions about the project and process.