Architecture/Design (PHP:Laravel)

Database Design:

Client

-id, name, address

Vendor

-id, name, address

Service

-id, name, category

Vendor Service

-Service\_id, vendor\_id

Client Request

* Service\_request, vendor\_id, client\_id,

Application development

Laravel provides **jobs and queues** using which, we can implement, the delayed request to vendor, functionality.

Generate the job:

php artisan make:job RequestVendorJob

command will generate Apps\Jobs\RequestVendorJob

* Put all the business logic in the job **handle** function.
* Necessary params from request is passed to the constructor of the job.

Delayed Dispatch:

For the delayed processing of the job from the queue, we will use the below code in the controller

RequestVendorJob::dispatch(params)->delay(now()->addMinutes(10));

The job dispatched will be available for processing after 10 mins.

Queue Setup:

We will use database as the queue connection.

So no change in config\queue.php and we keep database by default.

Create table to store jobs.

php artisan queue:table

Migration cmd

php artisan migrate

After the above setup in the project, any request by client will then dispatch the job.

The processing of the job is done by a worker.

To start the worker,

php artisan queue:work

we can set the number of tries the worker can try to execute the job if it fails

php artisan queue:work –tries=3

In order to keep track of the failed jobs, run

php artisan queue:failed-table

All the failed jobs are pushed to this table.

To retry the jobs, cmd

php artisan queue:retry all

Production Deployment

Through Forge

Important fields to configure:

* Queue connection
* Queue jobs
* Processes
* Maximum tries