Internal Review of Super eazy Application

1. Document from Yabi

I've gone through the document sent by Yabi. It covers the schema design, available fields in tables, relationships between them, and some basic integration notes. It's not complete, but it's useful and will help as a reference while exploring the application further.

2. Tables

Found around 400 database tables in the project.

- Not including dispatcher table.

The size and number of tables make the system quite large and complex.

Some grouping or modular structure might help make sense of it-could be worth mapping domains or feature areas later.

Cleanup may also be needed in places.

3. Models

A lot of Eloquent models exist, each tied to the database schema.

The structure and style aren't fully consistent across all models.

In several cases, business logic is sitting directly in the models-this could be moved to services later to make things cleaner.

4. Imports

There are import classes used to bring in external data, likely from CSV files.

It looks like Laravel Excel or something similar is being used.

The import logic is scattered and could be more centralized.

Error handling and validation during import may need improvement.

5. Exports

Export functionality exists and seems mainly for reports or syncing data out.

CSV is the primary format being used.

Some export logic is hardcoded or closely tied to specific use cases-could be made more modular or reusable.

6. Jobs

Background jobs are defined under App\Jobs.

They handle common async tasks like sending emails or generating reports.

However, some jobs are too tightly coupled with controllers-this might need to be separated out for better maintainability.

Still no clear signs of job dispatchers or queue tracking tables (like jobs, failed_jobs), which needs further checking.

7. Library

There's a Library (or sometimes called Support) folder with helpers and utility functions.

These are reused across different parts of the application to avoid duplication.

It's functional but could use some cleanup or organization-there are probably unused files or outdated logic in there.

8. Mail

Email functionality is built using Laravel's Mailable classes.

Templates are in place and look like they're being used properly.

Some mail classes are very basic, while others contain mixed-in logic that would be better moved to services.

More consistent structure and error handling would help here too.

Conclusion

This Laravel application is very large and clearly built for a complex use case. The document from Yabi is a good starting point and gives useful insight into the structure, especially the database side. Still, there are a lot of inconsistencies, missing parts (like job dispatchers), and tightly coupled logic that will need cleanup over time.

This is just the beginning of the review-there's a lot more to look into and gradually improve...