

MyMenu

CMPUT 401

Product Overview

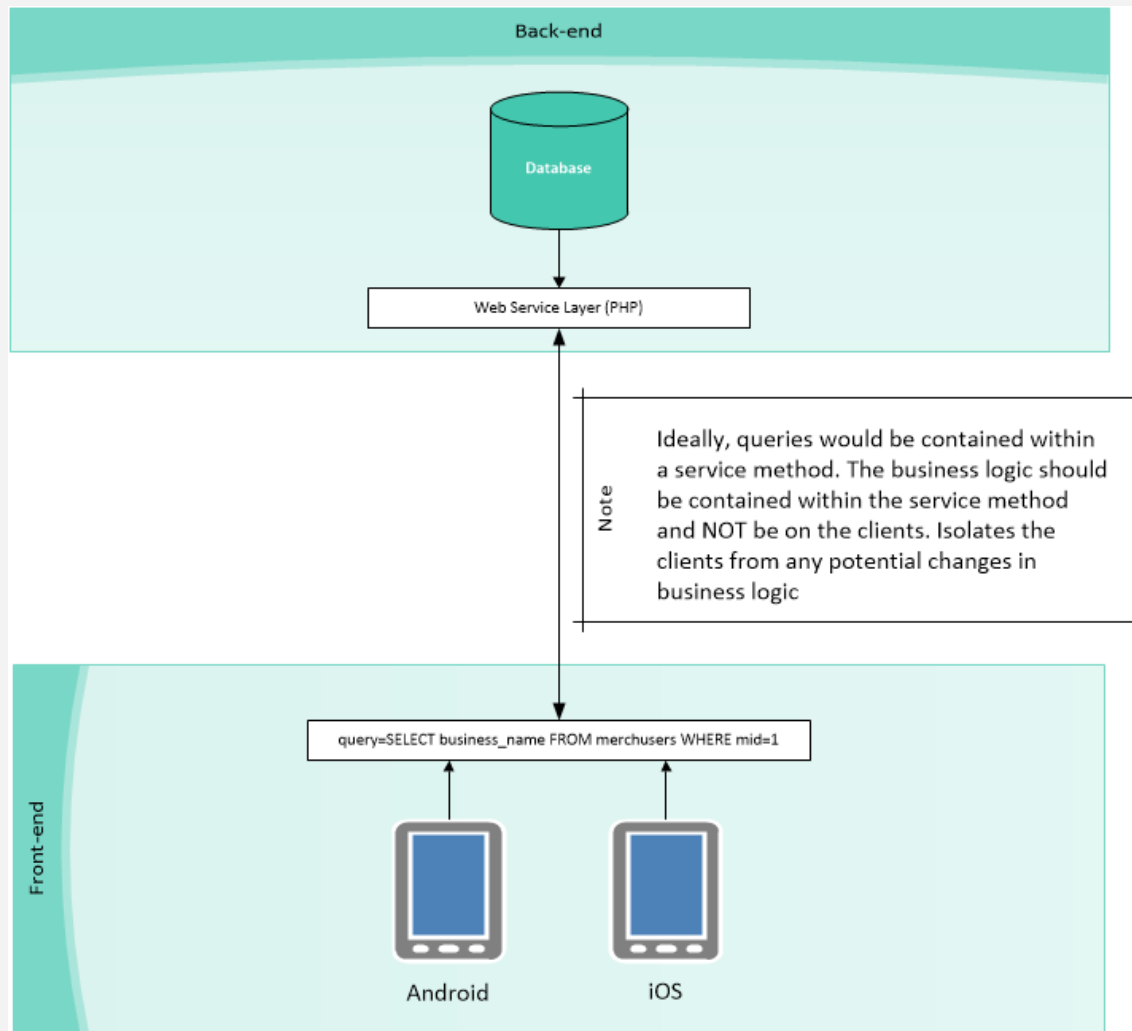
MyMenu is an App that is meant for users with food limitations that want to have an easy way to find out foods they can eat. The App has the Restaurants sign up and upload their menus to the online database. Users will be able to log in to their account, which holds their food preferences and restrictions. They will be able to see Restaurants near them, select them and then look at the menu for whatever restaurant they choose. The menu will show only the items that the user can eat based on the profile. Users will also be able to see specials that restaurants will advertise. The App will be available for iOS and Android.

Example User Scenarios

- A user that has a peanut allergy will be able to see food a restaurant offers that will cause an allergic reaction.
- If the user still wants an item, they can check to see if the item can be altered to fit their dietary needs.

Chosen Architecture

- We used Objective-C and Java to make two separate native tablet applications.
- We used MVC in both projects
- We used the clients existing server and database schema
- The clients database uses MYSQL



Implementation Challenges

- Creating views in a set orientation then trying to change over. i.e) landscape to portrait
- Trying to create consistency between our design implementation on a tablet and the clients existing phone design.

Management Challenges

We did not have any true management problems.

- We took turns addressing, creating and assigning issues.
- We met 3-4 times a week so communication between members was not a problem
- Client responded to questions in a quick and efficient manner

What we thought worked

- Pair Programming
- Meeting often and communicating on a regular basis
- Having a single client contact to funnel information

What we learned

- All of the members learned Objective-C and iOS.
- MySQL Trigger Syntax
- Members learnt proper software engineering techniques
- Proper Github source control

Open Source

	iOS	Android
Networking	AFNetworking	OkHttp, Retrofit
Serialization	RaptureXML	SimpleXML
Calendar	TimesSquare	TimesSquare
Concurrency	ReactiveCocoa, ReactiveViewModel	RxJava
Image Loading	SDWebImageView	Picasso
Testing	OMock	Retrofit-Mock, Spoon, Esspresso

What we would do differently

- Develop an iOS project in both orientations as you go.
- Develop both applications simultaneously.
- Better separation of concerns with respect to business logic.

Back-end



query=SELECT business_name FROM merchusers WHERE mid=1

Web Service Layer (PHP)

Note

Ideally, queries would be contained within a service method. The business logic should be contained within the service method and NOT be on the clients. Isolates the clients from any potential changes in business logic

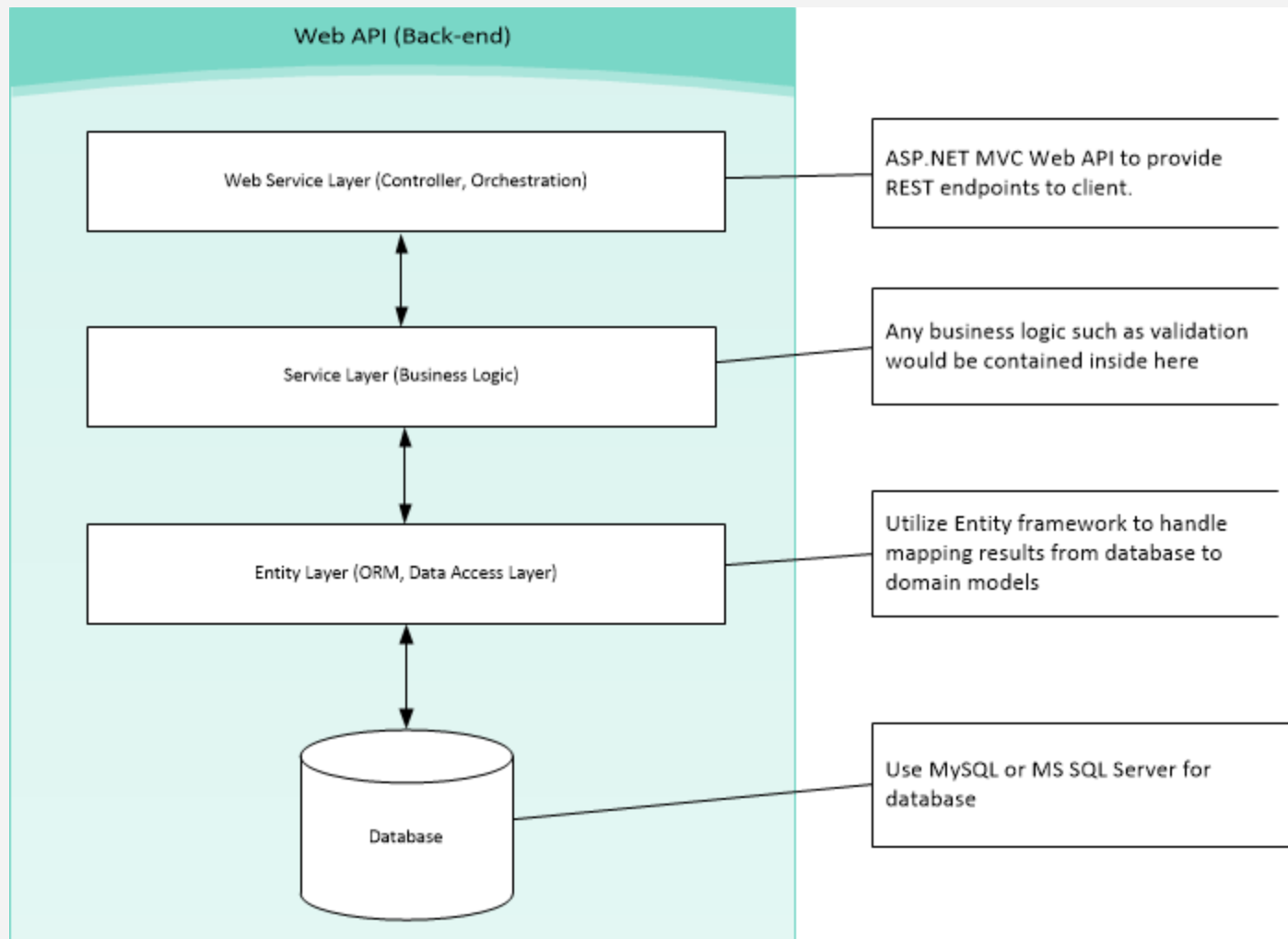
Front-end



Android



iOS



Questions?