

## M3 Condos Locatifs Appointment Scheduler Service Milestone 2 Summary

### Team members

Name and Student id	GitHub id	Number of story points that member was an <b>author</b> on.
Marc-Andre Leclair - 27754876 marc.leclair0113@gmail.com	<a href="#">16614353</a>	0
Wei He - 27279442 hewei196126@gmail.com	<a href="#">17327871</a>	1
Matthew Teolis - 40005332 matthewteolis@gmail.com	<a href="#">7413768</a>	0
Gibran Khan - 27647875 jbk5988@gmail.com	<a href="#">14865281</a>	0
Melissa Duong - 26993249 meliduong13@gmail.com	<a href="#">8919733</a>	0
Dylan Fernandes - 40002559 dylangfernandes@gmail.com	<a href="#">11564855</a>	0
Ema Dijmarescu - 29113401 ema.dijmarescu@gmail.com	<a href="#">32820346</a>	1
Vartan Benohanian - 27492049 vartanbeno@gmail.com	<a href="#">23425661</a>	0

### Project summary

This appointment scheduler is built with the goal of reducing misorganization, missed appointments, and the difficulty of two parties finding a mutual available time. A business shall post available times each week, and clients will be free to book a desired time-slot, all while seeing which time-slots have already been taken by other clients. This would avoid potential needless back-and-forth communication. As it is now, booking an appointment with [M3 Condos Locatifs](#) is always done through phone calls and/or emails, which isn't very convenient.

### Risk

The most significant risk to the project is that the expected traffic and hence the expected server load is not easy to quantify as the system that is being created is being built to address a client need that has not been quantified or forecasted before. That is to say that we do not know of the expected number of clients that the system may have to support. To address this we are going to use a web hosting service that fits the clients budget and that is designed to be reasonably handle the

requirements of a small business. Our current options are [Microsoft Azure](#) and [Amazon Web Services](#) (AWS) whose pricing and suitability we will discuss with the client.

### Legal and Ethical issues

- Issues with privacy (calendar sharing)
- Abuse of the rating system (users may be overly harsh)

### Velocity

There are no user stories that are signed off so the velocity is 0. We set up our GitHub, and created a list of main user stories for the project.

<https://github.com/vartanbeno/AppointmentScheduler/milestone/2>

Our backlog can be found [here](#).

Iteration 1 (0 stories, 0 points)

Iteration 2

Describe what you did and link in the stories

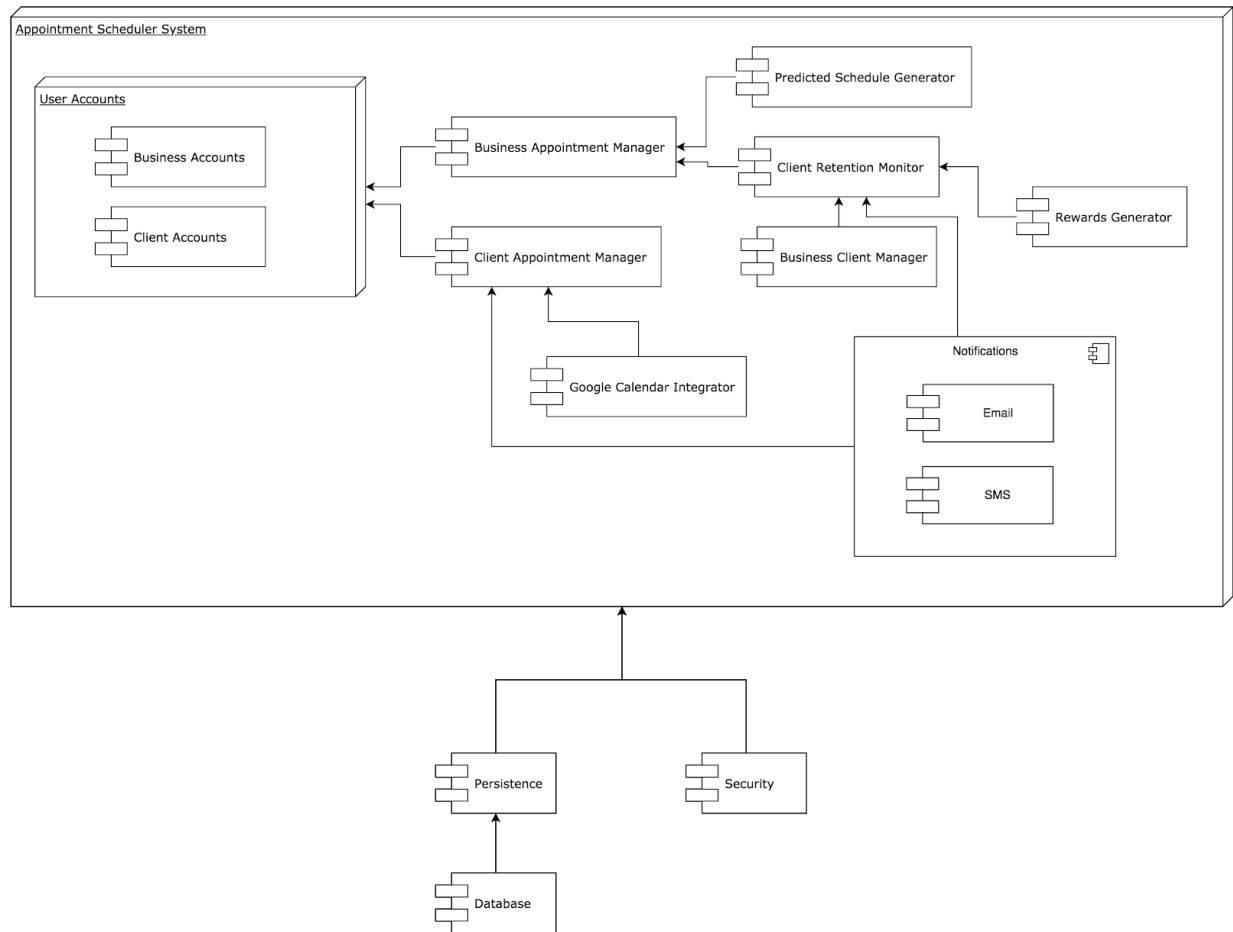
Iteration 3

Describe where you are going. I don't see a Iteration3 label!

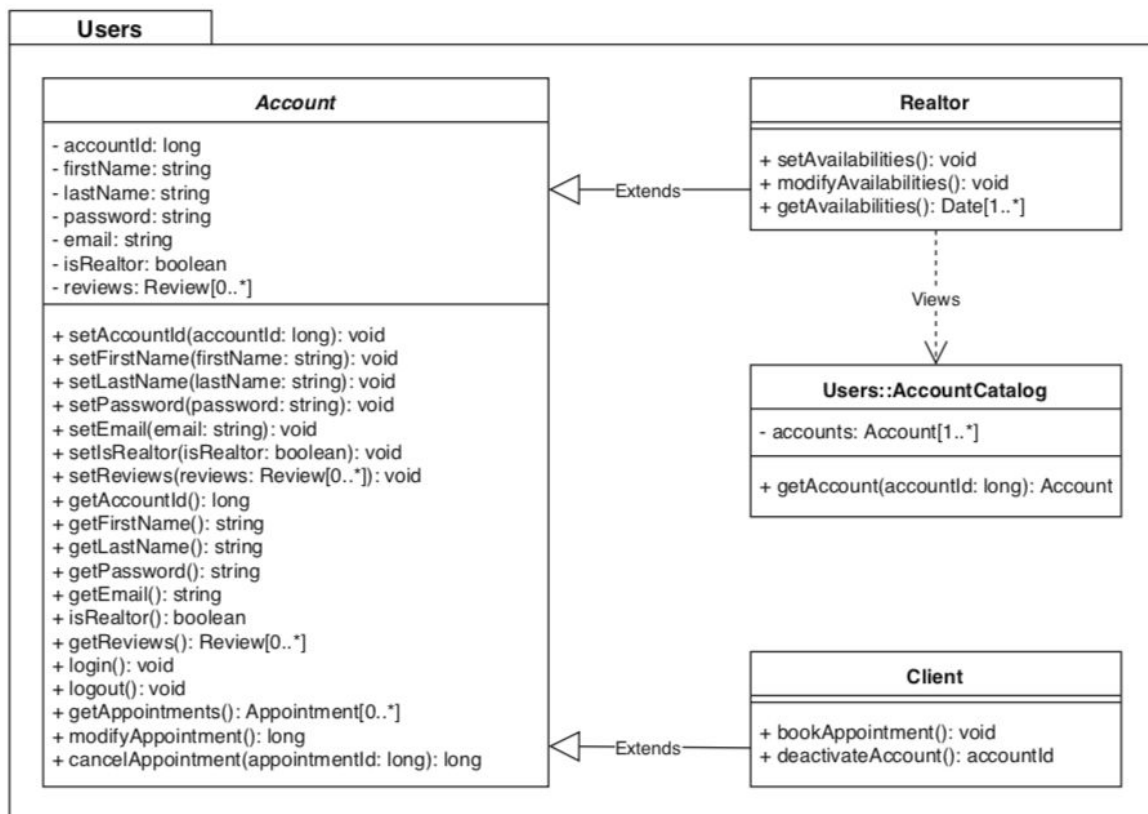
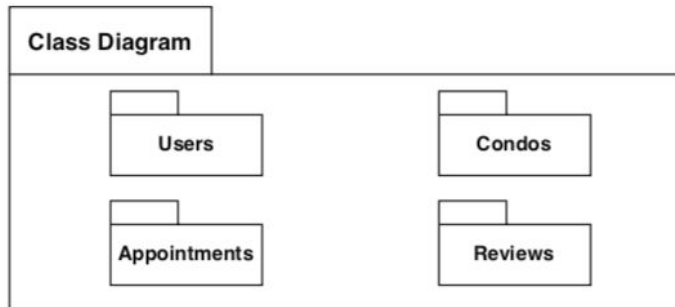
Iteration 4

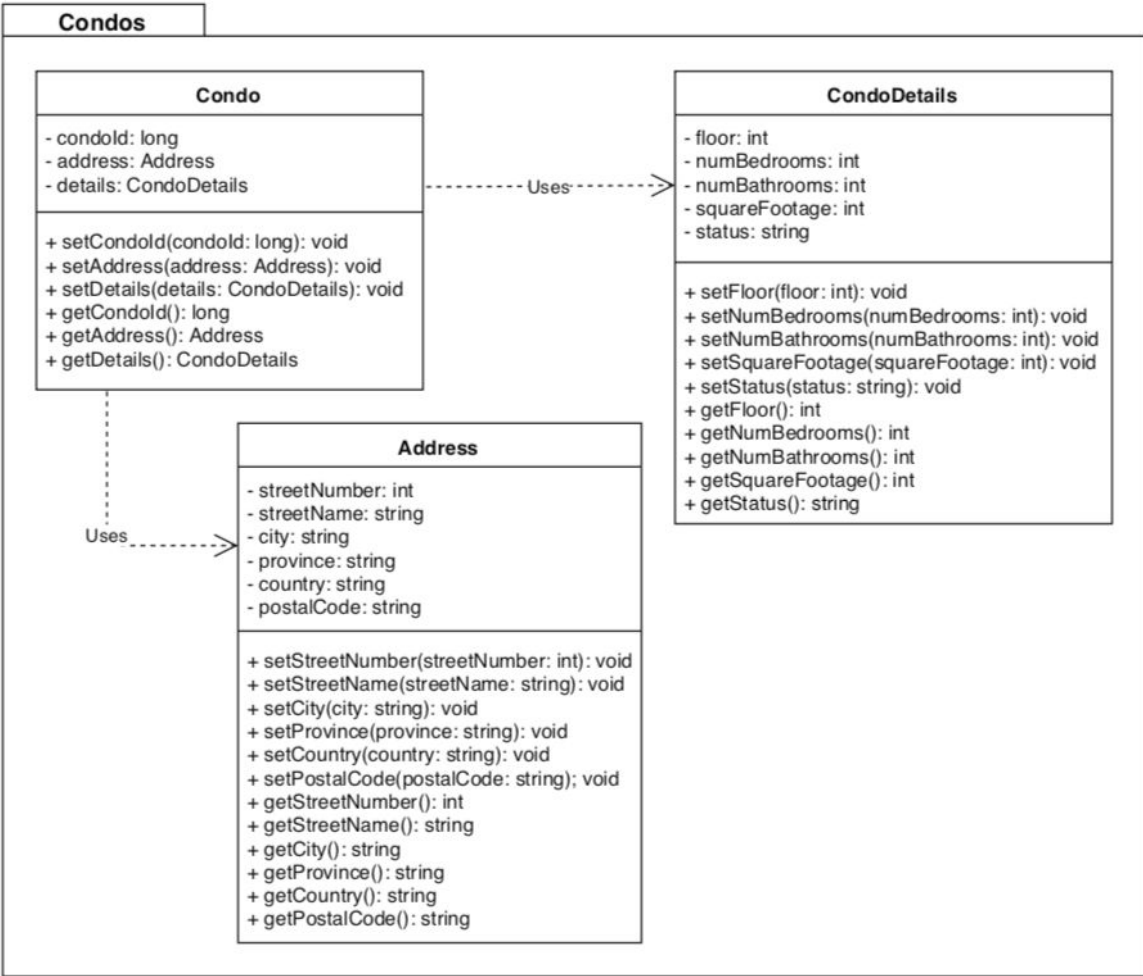
Ideally you plan atleast two iterations in advance

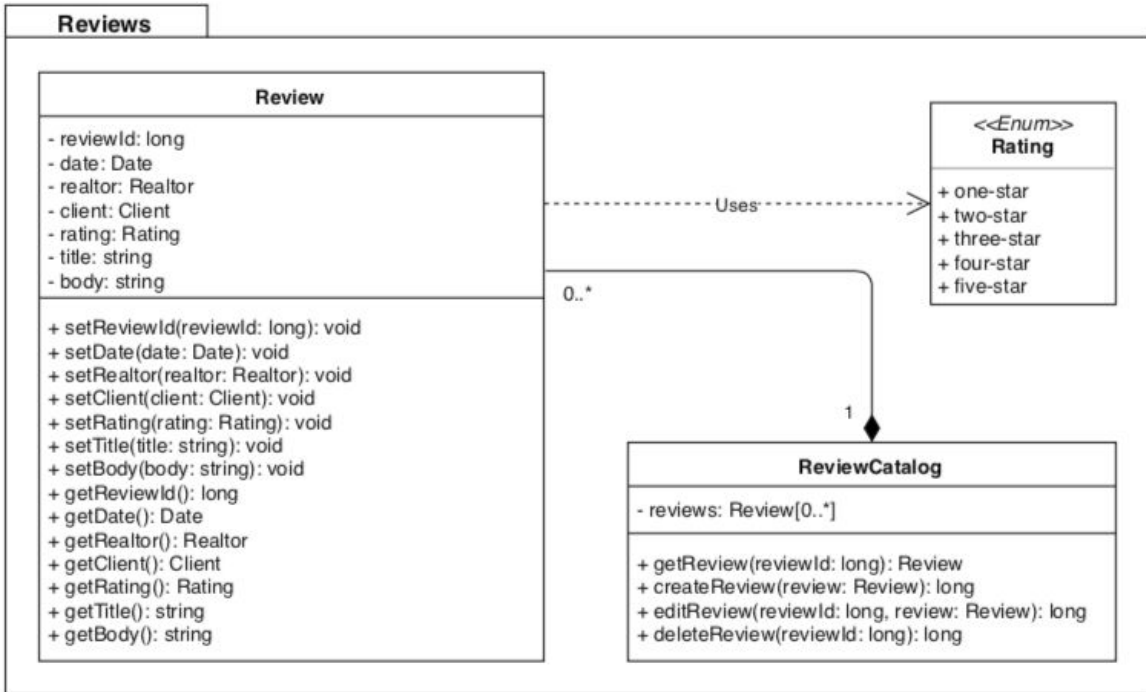
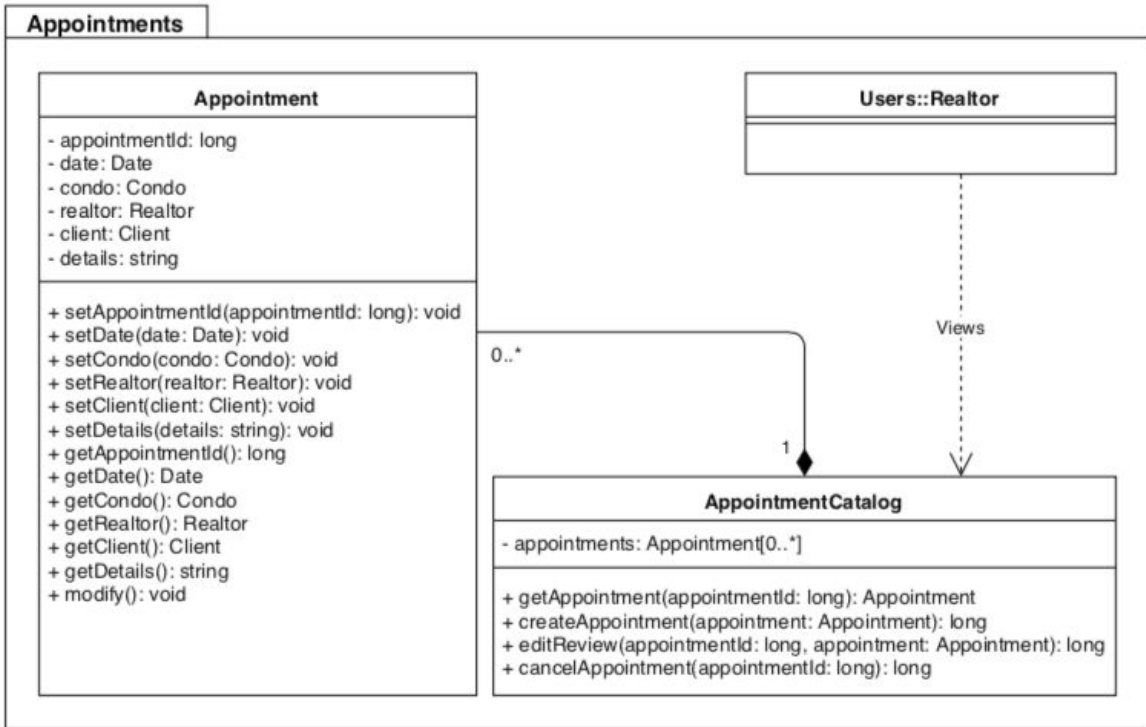
### Overall Architecture Diagram



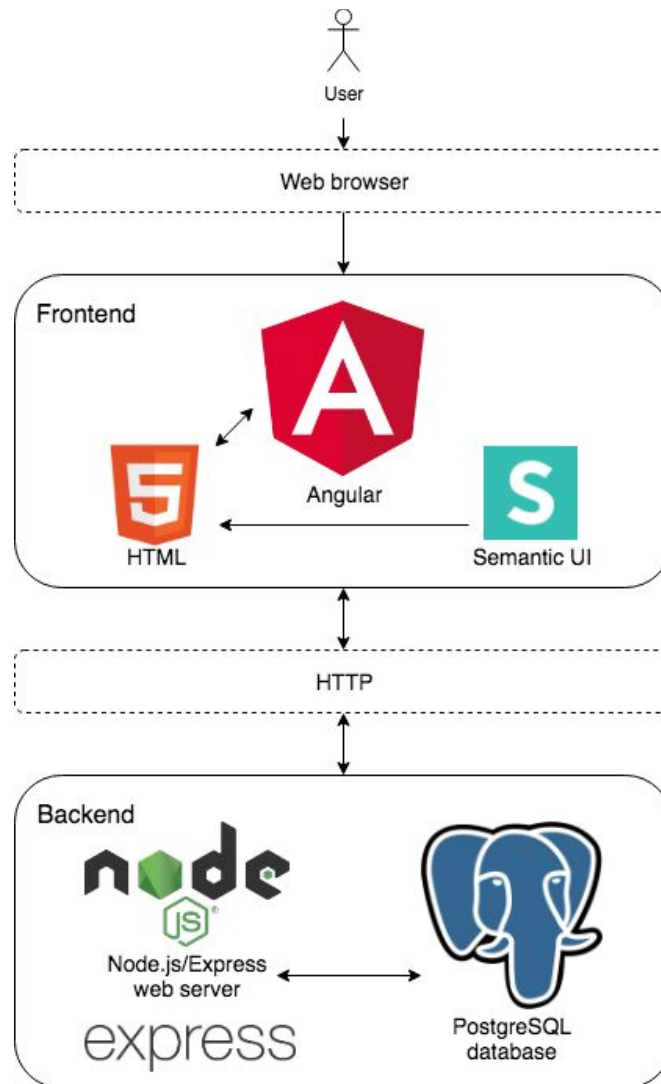
## Overall Class Diagram







## Infrastructure



### Frontend: [Angular/Semantic UI](#)

Angular is an open source frontend framework created by Google for building client applications using HTML and TypeScript. It is a component-based framework. The components represent the views of the application. Each component has an HTML template which makes up a view that's display to the client. Components can use services as dependencies which provide all sorts of functionality, and by using data-binding techniques, can change the data in the view in real-time.

Angular has an easy learning curve, so team members who aren't familiar with it won't have a hard time getting the grasp of it.

Semantic UI is simply a CSS framework used for styling. This is essentially chalked up to preference, and can easily be changed in the future.

### Backend: [Node.js/Express](#)

Node.js is an open source JavaScript run-time environment that, instead of executing code on a browser (client-side), it runs scripts on the server-side to produce web pages.

Express is an open source web application framework to be used with Node.js. It is used to design and implement API endpoints in the backend. We can combine these various endpoints with their corresponding HTTP requests to execute the user's desired operation.

### Storage: [PostgreSQL](#)

PostgreSQL is an open source relational database system. For the most part, it conforms with the SQL standard, so anyone familiar with can get used to it quickly.

To integrate PostgreSQL with Node.js, we will be using the [pg](#) (short for node-postgres) npm package, which provides modules for connecting to a PostgreSQL database, conducting queries, and getting results.

### Feature Toggling: [fflip](#)

“fflip” is a feature toggling module integrated with Express.js. It allows for definitions that determine the criteria for feature availability. Examples include limiting features to paid users or to a certain percentage of the user base. It also allows the creation of custom criteria for toggling. It is also one of the most used modules of its type present on GitHub and has garnered a positive reputation.

### Notes

We are considering using Google's open source [Go](#) programming language for our backend. It is currently a topic of discussion in the team.

### Name Conventions

Since the codebase will be heavily consisted of JavaScript, we will be using the [JavaScript Style Guide and Coding Conventions](#).

For the PostgreSQL database, we will be using the [SQL Style Guide by Simon Holywell](#).

### Code

None for now

### Testing and Continuous Integration

We picked Travis as our Continuous Integration tool to make builds of good quality. Travis will start a new build whenever a commit is made on the master branch and will check the code for errors. By configuring Travis as our project progresses, we



can automatically enforce a certain quality in our code and automatically run all the tests as soon as we merge to the master branch. Right now, we added the configuration file for Travis in our repository, and the file will get updated with more detailed scripts once the basic setup of the project is done.

The initial commit for the travis.yml file can be viewed [here](#). Travis' successful build can be viewed [here](#).