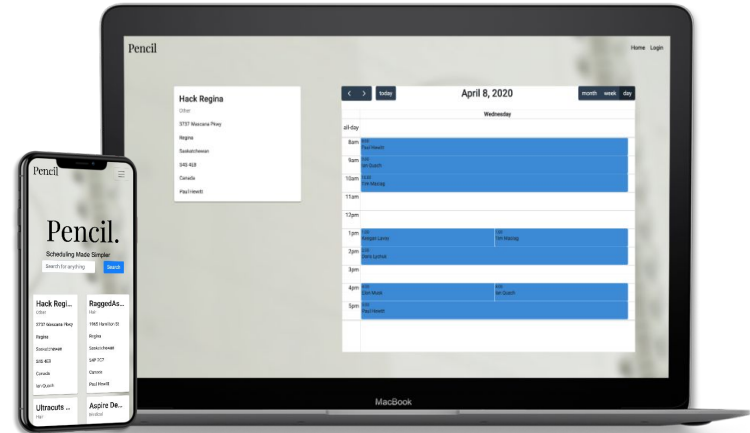


Pencil.

Scheduling Made Simple



Paul Hewitt & Ian Quach
ENSE Capstone



Paul
Full Stack Developer



Ian
Frontend Developer

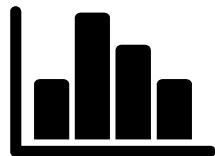
1.

The Problem

Why was Pencil made

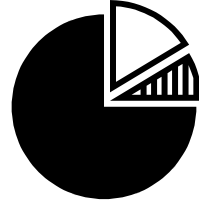
Background

- Rapid rise in small businesses and local entrepreneurs
- Many of these businesses require scheduling software to handle the booking of clients
- Solutions exist, but they are often rigid, and locked behind expensive paywalls
- These same solutions often do not promote local businesses and entrepreneurs
- It can be hard support local business when you don't know the businesses exist



350,190

Total appointments booked on BookedIn from
April-June 2018



40%

Of all online bookings happen outside of business hours

Meet Andy



@chungcuts

- Currently uses YouCanBook
- Cuts 13-15 clients a day
 - 6 days a week
- ~350 clients total

Andy's Wants

- Lightweight, easy to use application
- Low maintenance, significant uptime
- Customizable time slots
- User tracking
- SMS notifications
- Payment Integration

2.

The Plan

How Pencil was designed

Market Research

- One of the first things we did was research competitors
- Compiled a list of what they do well, and gaps that need to be filled
- Identified some of the ‘big players’
 - YouCanBook
 - SquareSpace
 - BookedIn

Pencil's Innovation

- Act as online business aggregator allowing users to discover new businesses and services
- Lightweight, users do not need a Pencil account to use the application
- Pencil is able to keep track of your most loyal customers, allowing for gamification
- Integrations with Facebook and Google's API

How we stack up

	Pencil	YouCanBook	SquareSpace
Gamification/Rewards	✓	✗	✗
Business Aggregator	✓	✗	✗
Third Party Integrations	✓	✓	✗
Customer Tracking	✓	✗	✗
Google Maps API	✓	✗	✓

Design Methodologies

- Agile
 - 'Sprints', 'Scrums', Standups
- User Stories, Customer Personas
- Lo-Fi Prototyping
- ScrumBan
 - Combination of Scrum and KanBan
 - Structure of Scrum, Flexibility of KanBan

Use Cases

Owner

Glen is a small business owner in Regina, Saskatchewan. He owns and operates a tattoo parlor. Glen relies on Google searches in order to be discovered, but due to poor search engine optimization, he often is buried under some of the other bigger tattoo parlors in town. He keeps a small log book at his front desk that he uses to keep track of his appointments, which are exclusively made over the phone.

Glen learns about Pencil, and signs his business up. He finds that he is suddenly booking more clients than ever, as customers of other services on Pencil are discovering him. His archaic log book is a thing of the past, and he is focusing on opening a second parlor soon.

Use Cases

Customer

Charlotte is a student at the University of Regina. A member of the Hill School of business, she strongly believes in supporting local entrepreneurs. Furthermore, she is a member of the AudacityYQR movement, reinforcing her love for local business. Her eyelash technician has just moved away, and she is tasked with finding a new one.

Charlotte wants to avoid any of the big chain studios, instead supporting someone from the same city as her. She learns about Pencil from a friend, and starts browsing. She quickly connects with a technician from White City, and has her first appointment this week.

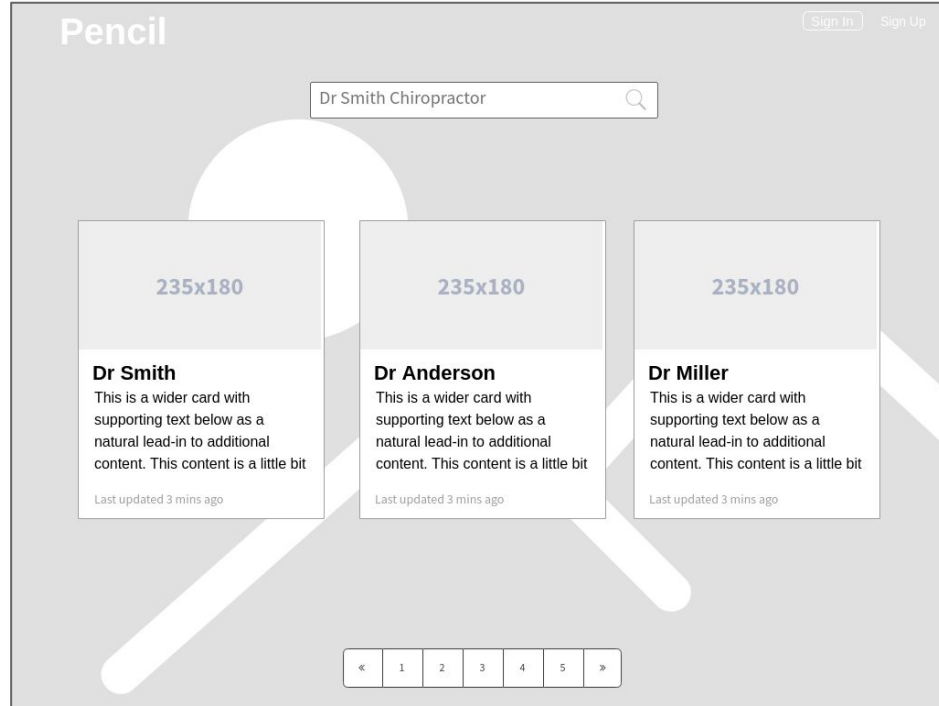
Customer Personas

Name	Andy - Business Owner	Steve - Looking for a new barber
Goals	Andy is looking for a new, online product to handle his scheduling. He is looking for something lightweight, that users can quickly visit, and book an appointment	Steve has been looking for a new barber. He would like to support a local business, one that is not too far away
Motivations	<ul style="list-style-type: none">• Would like to get more customers to book with him faster, and more efficiently• Would like an intuitive, easy to learn interface	<ul style="list-style-type: none">• Would like to see a comprehensive list of barbers in his area• Needs a service provided to him by a professional
Frustrations	<ul style="list-style-type: none">• His current system requires customers to create accounts before booking• His current system does not allow for custom time slots• His current system does not track users	<ul style="list-style-type: none">• There is no cohesive list of local service providers• Making accounts on other scheduling apps is tedious, and it is annoying to keep track of login information

User Stories

- As a *customer* I want to be able to see appointments offered by a service
- As a *customer* I want to book my desired appointment
- As a *customer* I want to receive notifications about my appointment
- As a *customer* I would like to discover new businesses
- As a customer I would like to rebook easily

Lo-Fi Prototypes



KanBan

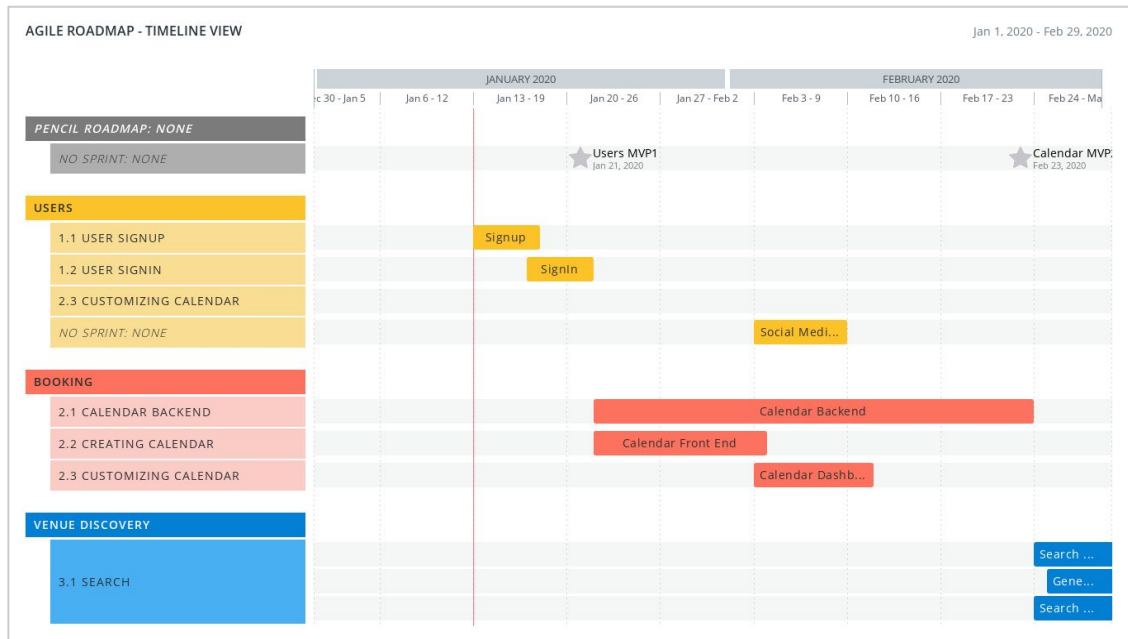
	Epic	Steps	Assigned to
Need to Start	Sign up/login	Sign up page	
		Login page	
		Business sign up	
	Search	Search design	
		Search results display	
	Scheduler	Time picker modal	
		Appointment type	
		Rebooking	
	Employee Control	Hierarchy system	
		Owner admin rights	
Employee specific pages/schedule			
Payment	Prepay appointments		
	Credit card authorization		
Reminders	SMS notification, set and forget		
	Scheduled automatic texts		
User Tracking			
	Gamification		
In Progress			
Complete			

Our first KanBan, October 2019

ScrumBan Process

- Combines the structure of Scrum, with the flexibility of KanBan
- Sprints based off of KanBan tasks
- Great for prioritization
- Associated MVPs with Sprints

RoadMap



3.

The Tech

What makes Pencil work

Technical Planning

Frontend

- Angular
 - HTML, CSS, TypeScript
- BootStrap

Backend

- AWS
 - GoLang
- ServerLess Framework
- REST API

Technical Planning

- GitHub for Source Control
 - Used Commitizen for neat and tidy commit messages
 - `type(scope)`: Short description of commit

Feat: A new feature

Fix: A bug fix

Docs: Documentation only changes

Style: Changes that do not affect the meaning of code

Refactor: A code change that neither fixes a bug or adds a feature

Technical Justification

■ Web App

- ☐ Not limited to platform, like iOS or Android
- ☐ Lots of businesses have computers at their front desk
- ☐ Allow clients to book appointments on the go
- ☐ Owners can monitor schedules away from the office

■ Angular

- ☐ Developer familiarity would allow for rapid development and prototyping
- ☐ Industry standard front end framework

■ Bootstrap

- ☐ Ensures responsive design, quick UI prototyping

Technical Justification

■ **AWS**

- ☐ Arguably the most popular cloud computing platform
- ☐ AWS experience is invaluable for heading into industry
- ☐ Would allow us to meet the uptime requirement

■ **Go**

- ☐ Google's statically typed, compiled language
- ☐ One of Stack Overflows most desired language
- ☐ Known for its speed, and integration libraries

■ **REST API**

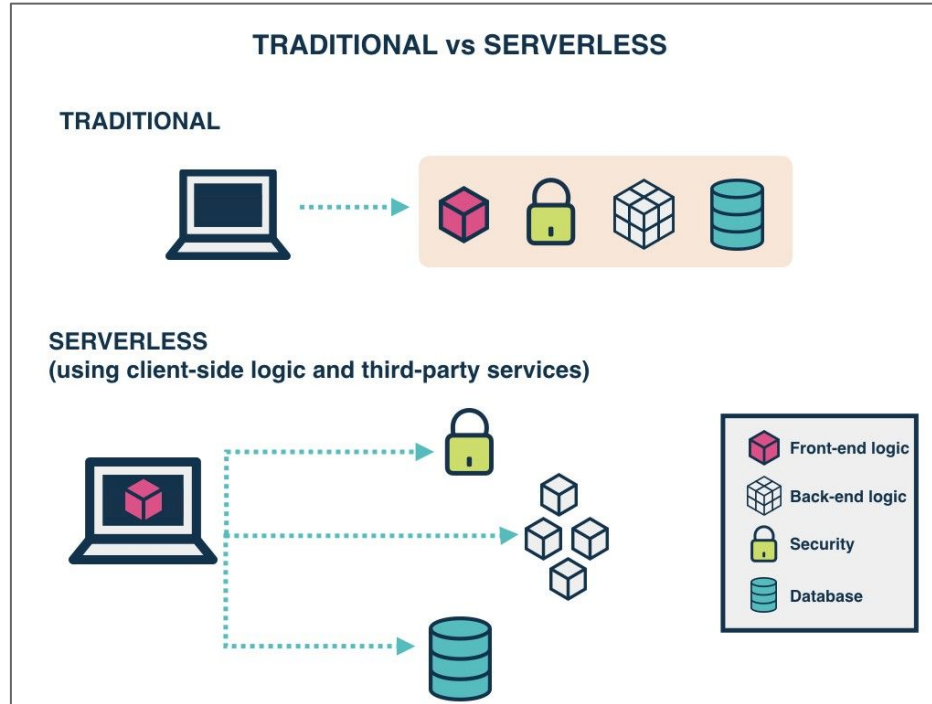
- ☐ Industry standard
- ☐ Much simpler to implement than SOAP
- ☐ Fast

Technical Justification

■ **Serverless Framework**

- ☐ Allows Pencil to use the Serverless cloud computing model
- ☐ Gives AWS more operational responsibility
- ☐ Keeps costs down, only pay for what you need
- ☐ A new, popular model that seemed fun

Technical Design Patterns



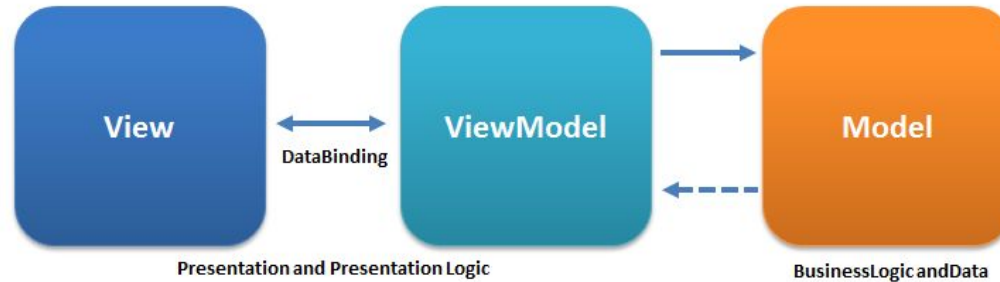
Technical Design Patterns

The Serverless 10 Commandments

1. Build only what differentiates you, outsource what doesn't.
2. Favor serverless cloud services, stateless compute, events, APIs & open-source.
3. Code is a liability. Keep it to a minimum.
4. No instances, servers or containers, unless there is no other way.
5. Establish customer satisfaction as the highest priority of engineering.
6. To deliver the best possible outcomes, maintain the freedom to choose the best possible services.
7. Focus on product goals and experience over technology.
8. Compose and configure, before writing code.
9. Own the full lifecycle of everything you build.
10. Deploy is the new commit.

Technical Design Patterns

- Model-View-ViewModel (MVVM)



Services Architecture

- Pencil uses the Services architecture for the REST API
 - 4 unique Services
 - Each serves a specialized purpose
 - Each Service has different functionalities based on HTTP Verb
 - GET, POST, etc
 - Each Service is mapped to an AWS Lambda Function
- REST API is hosted on AWS

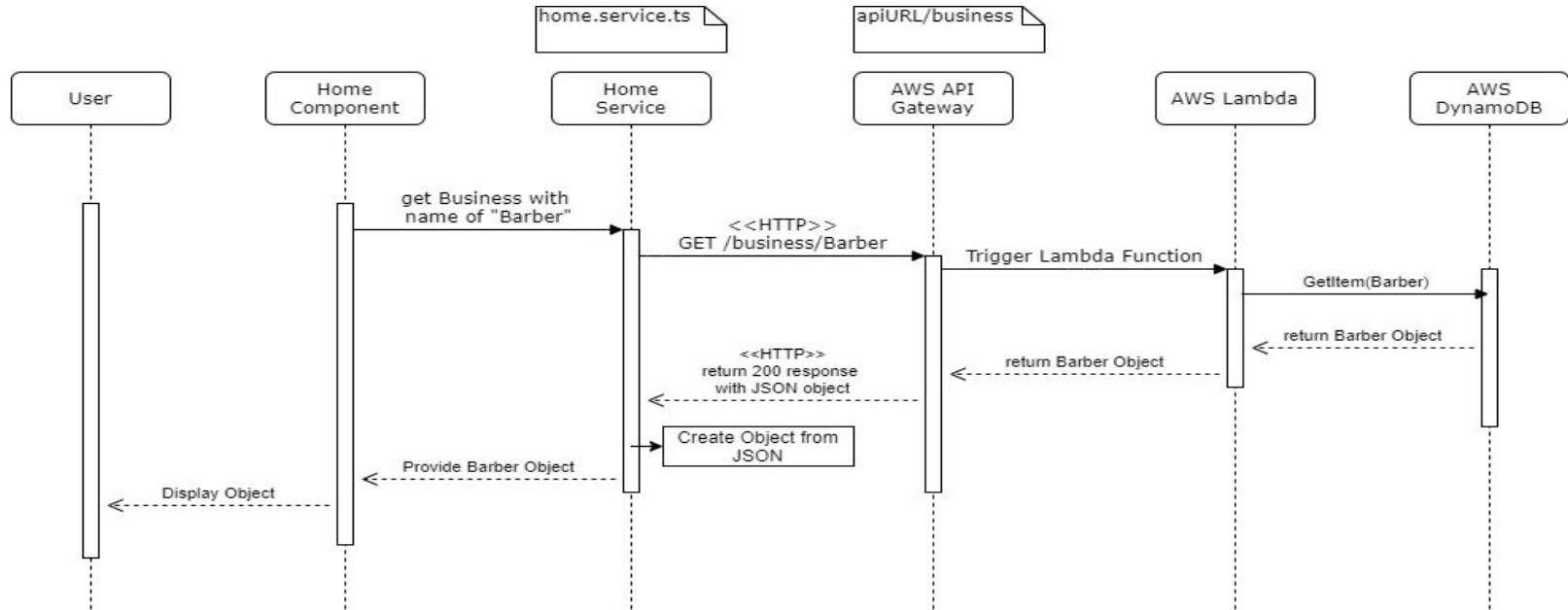
Business Service

- Purpose: To retrieve individual businesses from the database
- Route: /business
- HTTP Method: GET
- Params: Name of business to retrieve
- Headers: Content-Type, application/json
- URL: Pencil API URL
- Body: N/A
- Database: Businesses Table

Creating a new business

```
public createBusiness(businessForm): Observable<any> {  
    const url = `${environment.apiUrl}businesses`;   
    const headers = this.createHeaders();  
    return this.http.post<any>(url, businessForm, headers);  
}
```

Retrieving a business



AWS

- S3
- API Gateway
- Lambda
- DynamoDB
- CloudWatch Logs

AWS S3

- Using a makefile, Pencil's backend Go files are compiled into .bin files
- These files are uploaded to S3 on deploy via ServerLess Framework
- API Gateway uses the files in the S3 bucket

AWS API Gateway

- API Gateway unpackages the files from the S3 bucket
- Creates our REST API
 - Creates the API URL
 - Separates the .bin files into appropriate services, routes, and endpoints
 - Maps each service to an AWS Lambda function

AWS Lambda

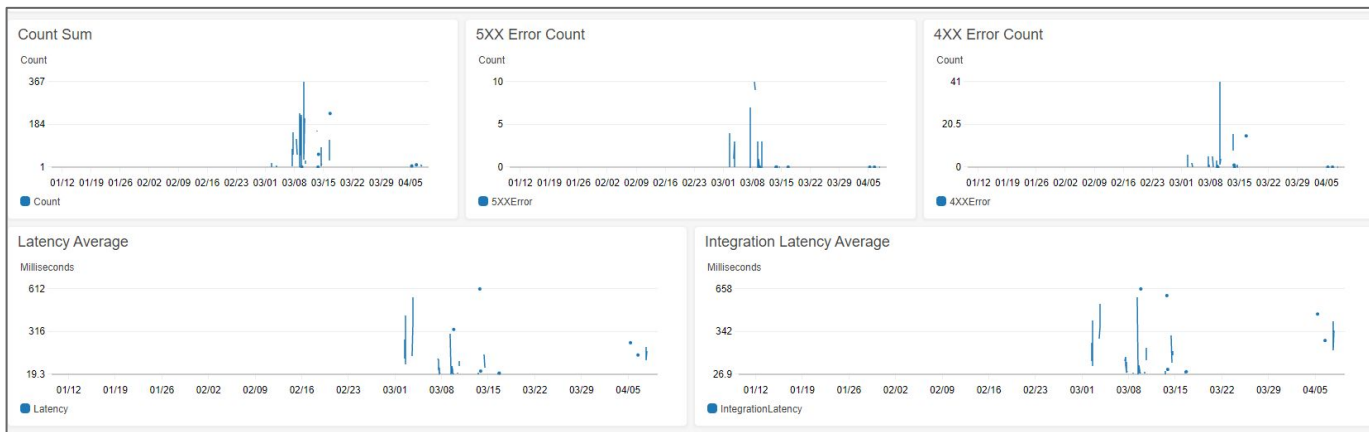
- Lambda allows you run code without setting up servers
 - Obvious choice for our serverless application
- Depending on the type of HTTP request API Gateway receives, the proper Lambda function will trigger
- This function will remain 'active' until no triggers are received for ~30 minutes
- The Lambda function executes code, interacting with the database, DynamoDB

AWS DynamoDB

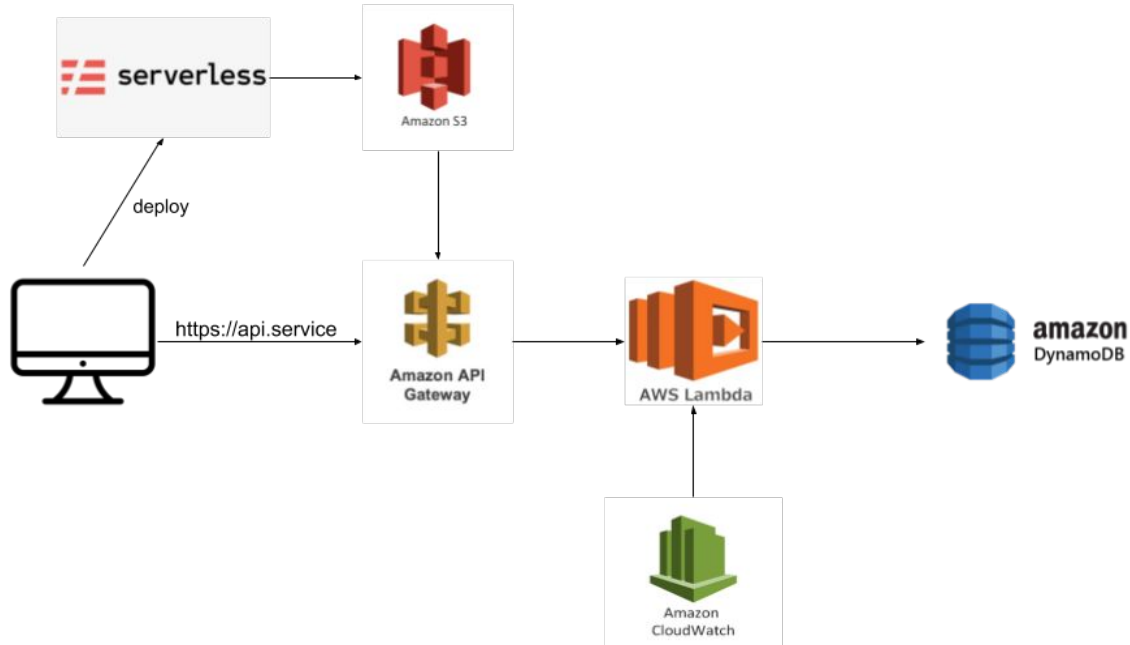
- Serverless Database
- NoSQL
- Each table represents a collection of objects
 - These objects must contain attributes
- The 'Primary Key' is called a Partition Key

AWS CloudWatch Logs

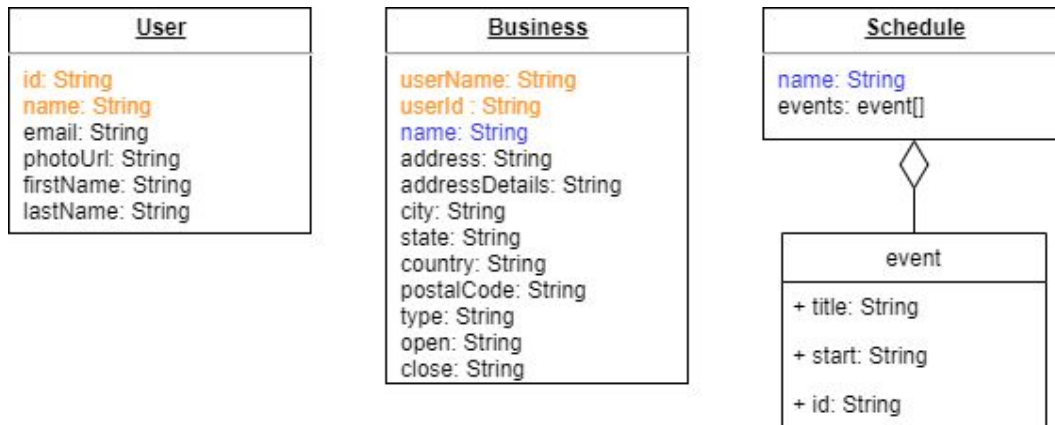
- Logging service
- Monitors and records data about all aspects of our REST API
- Provides a dashboard for at a glance monitoring



AWS Diagram



Data Models

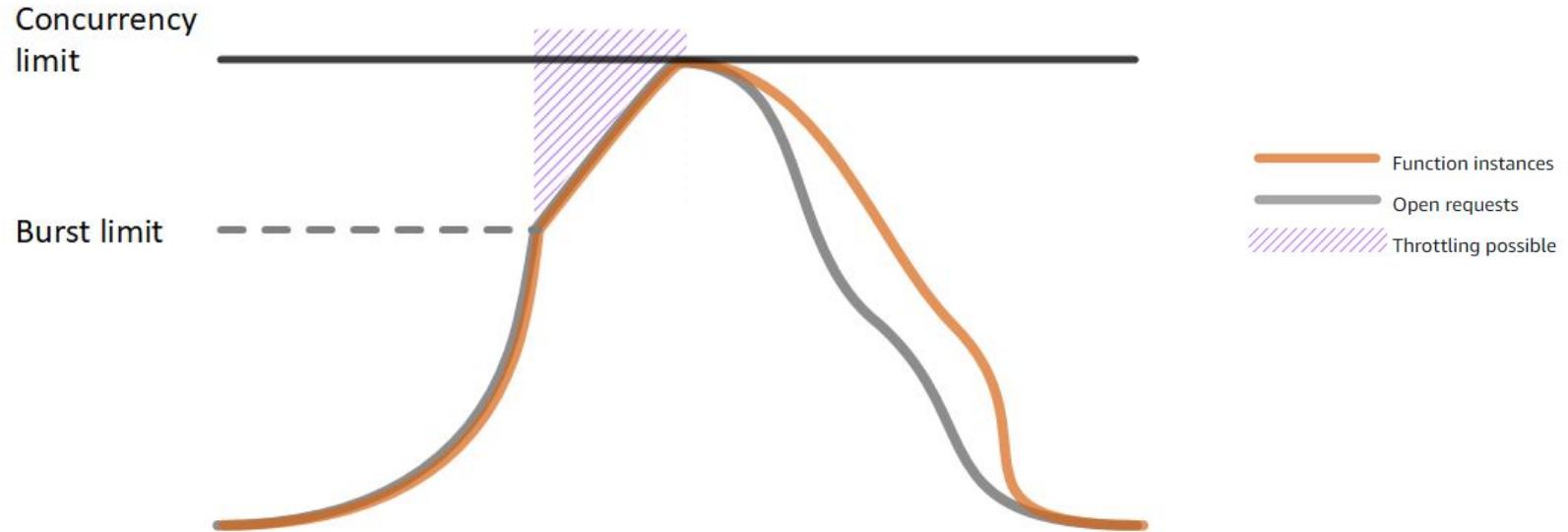


REST API Performance

- As Pencil is Serverless, it is able to scale efficiently under extreme loads
 - Each Lambda function is able to initially handle 1000 concurrent requests, with that limit increasing by 500/minute
- Potential choke point might be the database, as we only have one table per object

REST API Performance

Function Scaling with Concurrency Limit



Integrations

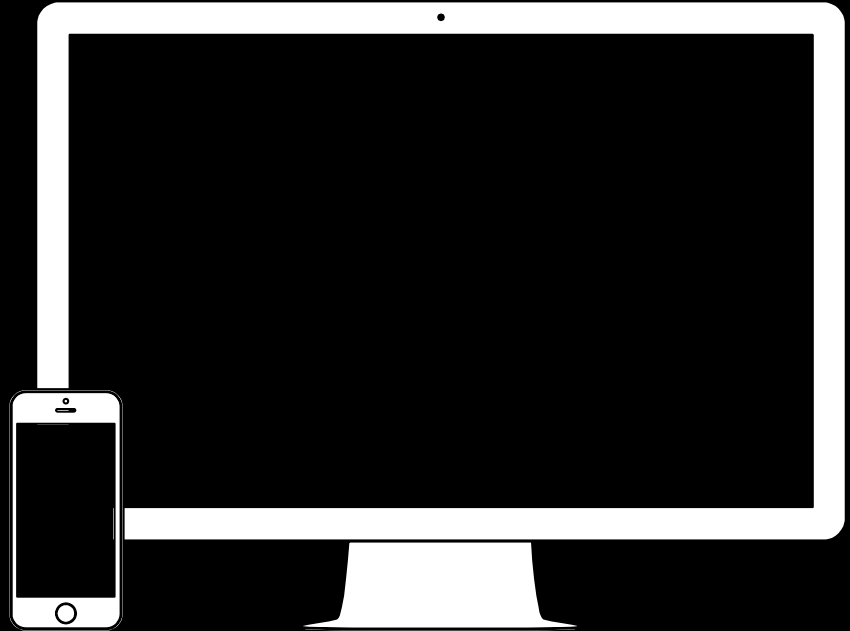
- Pencil uses the Google Maps API in order to display local businesses on a map
- Pencil is integrated with the Facebook API to handle user authentication
 - Facebook account is required to use Pencil at this time
 - Upon clicking on Login, a user is redirected to Facebook
 - Upon successful login, they are redirected to Pencil
 - Pencil is returned the users Facebook Object
 - This object is directly stored in Pencils user table

4.

The Implementation

Showing off Pencil

Demo



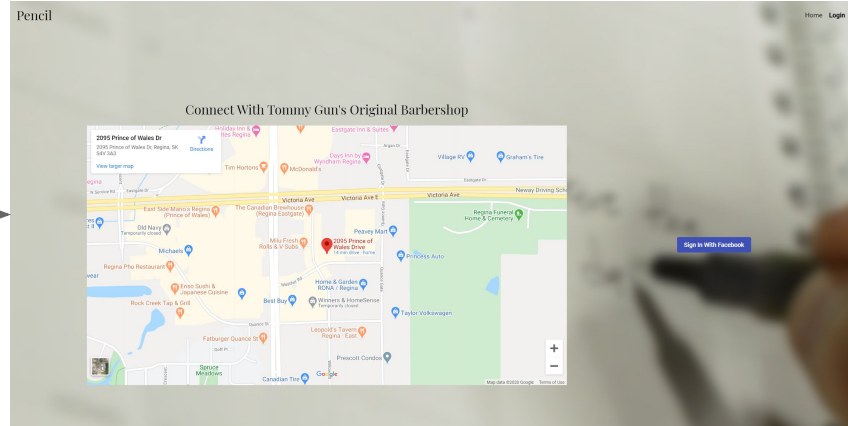
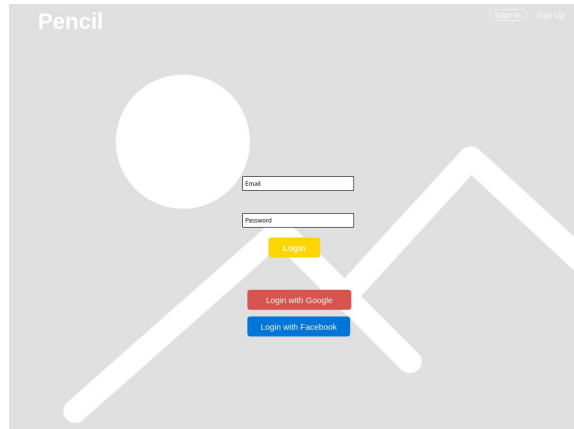
Code Testing

- Jasmine, Karma
- ng test can be used to run tests
- Test results will be output to a web browser window

```
describe('HomeService', () => {  
  const service: HomeService;  
  beforeEach(() => { service = new HomeService(); });  
  
  it('#getBusiness should return the name Ragged Ass Barbers',  
    (done: DoneFn) => {  
    service.getBusiness('Ragged Ass Barbers').subscribe(value => {  
      expect(value.name).toBe('Ragged Ass Barbers');  
      done();  
    });  
  });  
});
```


User Testing

- Single web page design
 - Snappy and responsive
- Started testing with lo-fi designs in AdobeXD
- Loved ones and parents were used as testers



User Testing Results

- Started testing with lo-fi designs in AdobeXD
 - 5 total testers
- Older folks, some less tech literate
 - Found it bland, calendar concerns
- Younger testers
 - Not very eye catching, nothing memorable
 - Some dated design choices

User Testing Results

- Coded Design Test
- Arlene Hewitt
 - ☐ Tech illiterate
 - ☐ Single login source?
 - ☐ Navigation
- Don Hewitt
 - ☐ Liked embedded map
 - ☐ Screen resizing looked off on smaller screens
 - ☐ Business cards on home screen
- SARS-CoV-2 concerns

Budget

Expense	Description	Cost
Labor	155hr * \$40/hr	\$6,200.00
Hosting Fees	AWS	\$0.79
License Fees	N/A	\$0.00
Marketing	Digital Poster	\$0.00
		\$6,200.79

Business Plan

- Pencil will utilize a “Freemium” business model
 - ☐ Certain features are free for all users
 - ☐ Premium features are reserved for paying customers

- Planned Premium Features
 - ☐ Customized owner calendars
 - ☐ Gamification
 - ☐ Promoted or Pinned Businesses
 - ☐ Manage more than one business

Next Steps

- Flesh out the data visualization/analytics
- Add another Social Media service for user authentication
- Add deeper customization to schedules and calendars
- Implementation of the 'Premium' features
- SMS notifications (AWS SNS)
- Payment Options (Stripe)

GitHub

- Comprehensive Design Document
- Code Quality Review
- Project Experience Report
- User Guide
- Documents from the last 8 months

<https://github.com/paulhewitt/Capstone>

Reflection

- Scheduling is an extremely complicated process
- Social media login is double edged sword
- Sometimes 'better' UI does not equal better UX
- AWS documentation could be much better

Thanks!

Any questions?

Send us an email!

- paul@hewitt.dev
- quach.ian07@gmail.com

Slide Template

- Presentation template by SlidesCarnival