REVATURE

Health Pair

Project 3:
Connecting patients with providers

Developed by: Paul Edwards, Shawn Halcomb, Kyran Bryant, and Tyler Claypool

Overview

Created through the cooperation of a Development team and a Business Analyst team!

FIND A DOCTOR NEAR YOU!

- Connects patients with the *perfect* provider for their situation:
  - Filters by Insurance Accepted
  - Filters by Provider Specialty
- Sorts providers nearest to the patients current location
- Patients are able to save their profile and insurance information
- Performs as a lightweight and easily accessible web application

Project objective:
Connect patients with providers

Understanding the problem

Insurance

Providers typically only accept certain insurances.

Patients should be able to filter providers by insurance accepted, or have a “cash-pay” option.

Specialty

Not all providers are trained in the same specialities.

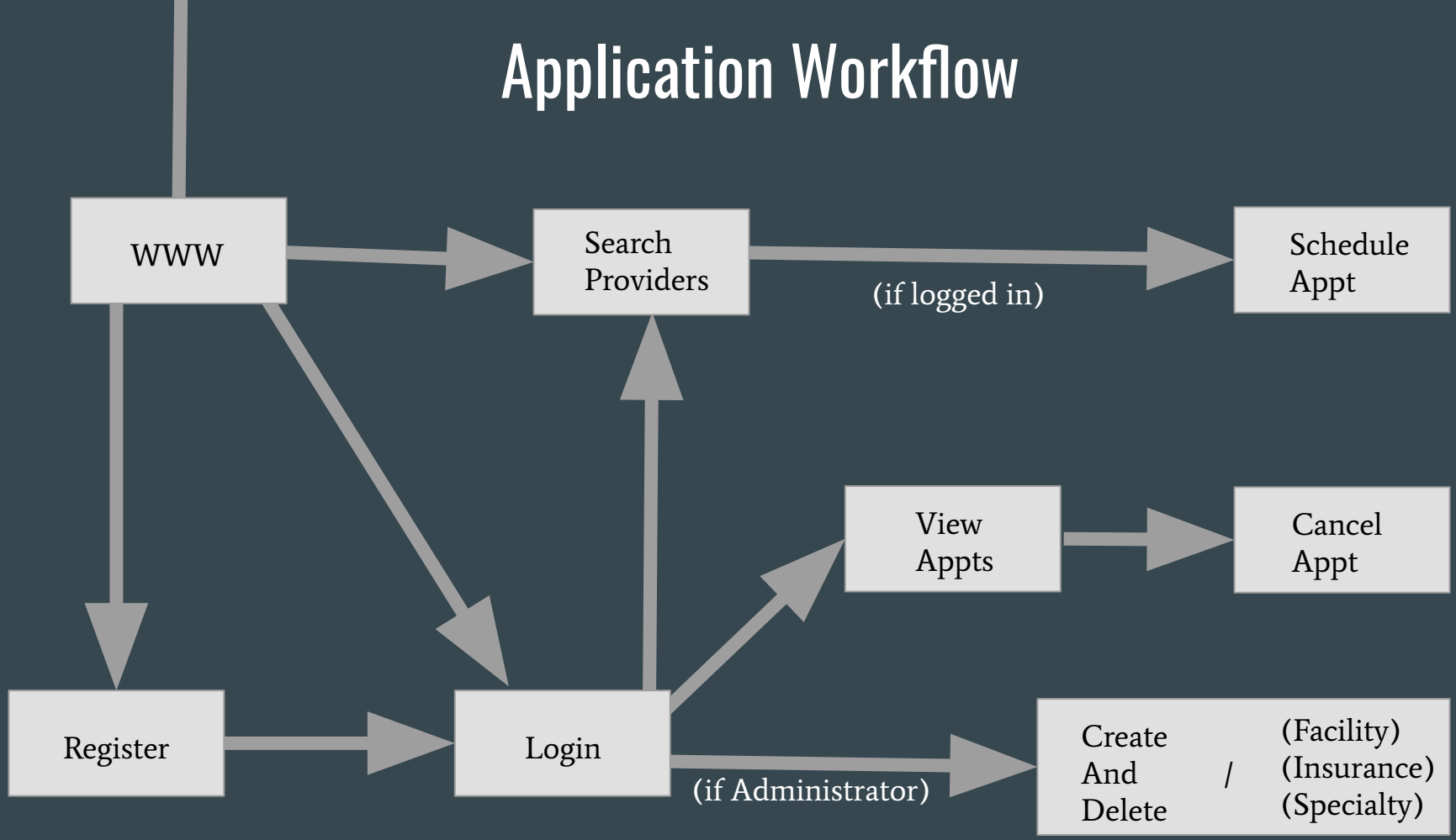
Patients need to be able to filter out for the specialist best fit for their situation.

Distance

Patients appreciate the security of having their specialist nearby.

Patients should be able to sort providers by distance.

Application Workflow

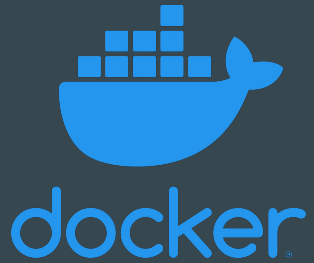


Frontend Technology

- Angular
- TypeScript
- jQuery
- Bootstrap
- Forms
- Validation
- Modals

Demo

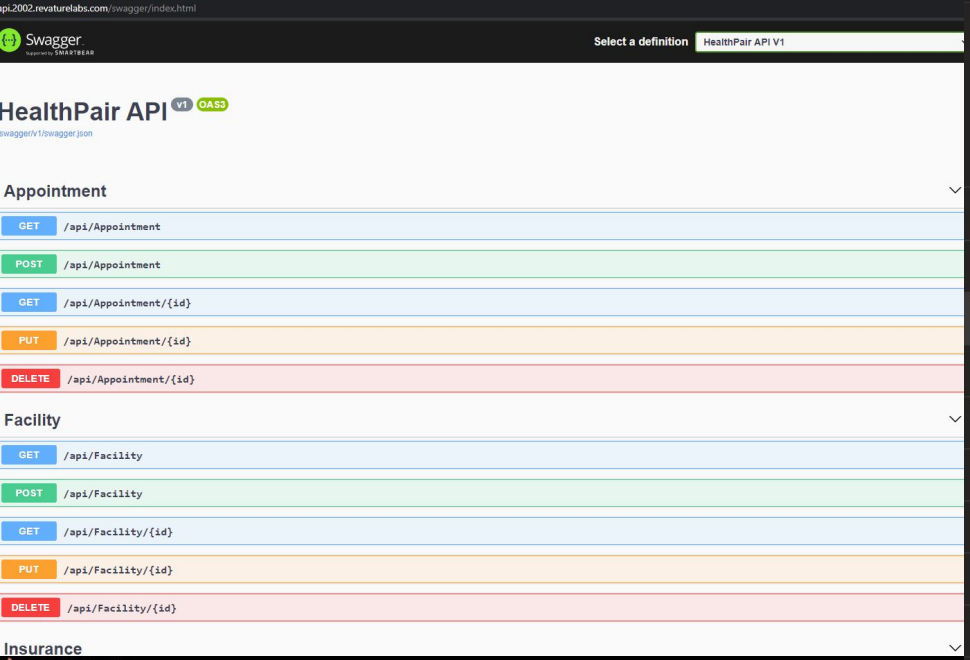
Technology Overview

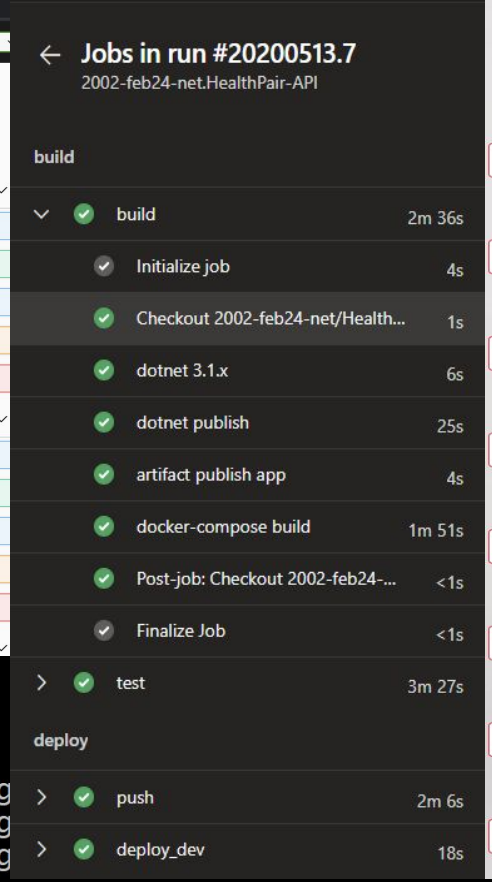


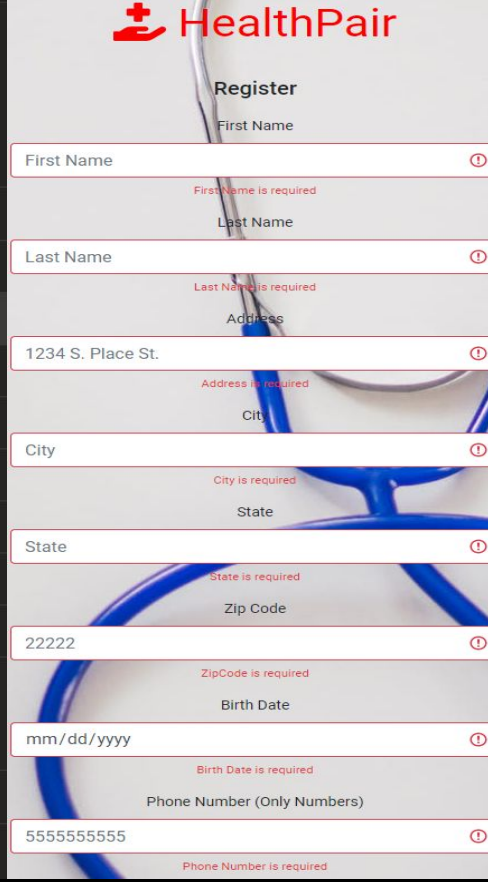
GitHub

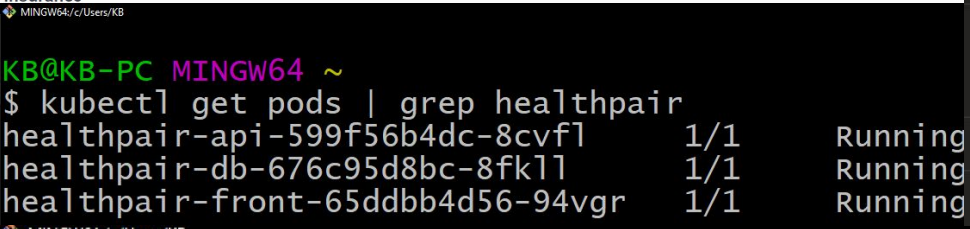
Backend Technology

- Azure PipeLines
  - SonarCloud Code Analysis
  - Docker Containerized
  - Kubernetes Management
- PostgreSQL
  - Code-First Approach
  - Allows Access through API
- Repository Pattern
  - Mapper
- Swagger Initialized
- Exception Handling/Validation
- Authentication


api.2002.revaturelabs.com/swagger/index.html
Swagger
Select a definition
HealthPair API V1
HealthPair API
V1
OAS3
swagger/v1/swagger.json
Appointment
GET /api/Appointment
POST /api/Appointment
GET /api/Appointment/{id}
PUT /api/Appointment/{id}
DELETE /api/Appointment/{id}
Facility
GET /api/Facility
POST /api/Facility
GET /api/Facility/{id}
PUT /api/Facility/{id}
DELETE /api/Facility/{id}
Insurance


Jobs in run #20200513.7
2002-feb24-net.HealthPair-API
build
build 2m 36s
Initialize job 4s
Checkout 2002-feb24-net/Health... 1s
dotnet 3.1.x 6s
dotnet publish 25s
artifact publish app 4s
docker-compose build 1m 51s
Post-job: Checkout 2002-feb24-... <1s
Finalize Job <1s
test 3m 27s
deploy
push 2m 6s
deploy_dev 18s


HealthPair
Register
First Name
First Name
First Name is required
Last Name
Last Name
Last Name is required
Address
1234 S. Place St.
Address is required
City
City
City is required
State
State
State is required
Zip Code
22222
ZipCode is required
Birth Date
mm/dd/yyyy
Birth Date is required
Phone Number (Only Numbers)
5555555555
Phone Number is required


MINGW64:/c/Users/KB
KB@KB-PC MINGW64 ~
$ kubectl get pods | grep healthpair
healthpair-api-599f56b4dc-8cvfl 1/1 Running
healthpair-db-676c95d8bc-8fkll 1/1 Running
healthpair-front-65ddbb4d56-94vgr 1/1 Running
MINGW64:/c/Users/KB
KB@KB-PC MINGW64 ~
$ kubectl get services | grep healthpair
healthpair-api LoadBalancer 10.100.2.66 a6ce75131bda94b61ab4b7f5f28aea65-578836505.us-east-2.elb.amazonaws.com
healthpair-db ClusterIP 10.100.66.102 <none>
healthpair-front LoadBalancer 10.100.148.154 a846dd4e8e4e14701bf08588e007d22b-636555553.us-east-2.elb.amazonaws.com

Challenges

Challenge 1	• Distance Calculations
Challenge 2	• Different results/outcomes on different Computers
Challenge 3	• Time zone discrepancies.
Challenge 4	• Cors issues.

DevOps

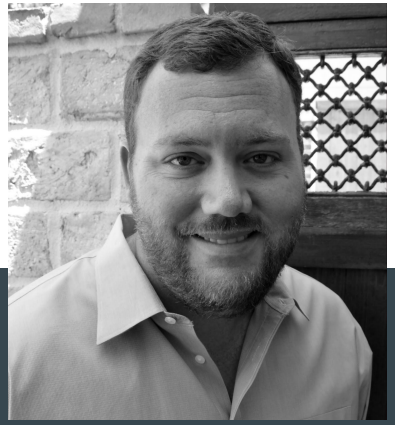
Findings

All challenges were overcome as a team.

Client Implications:

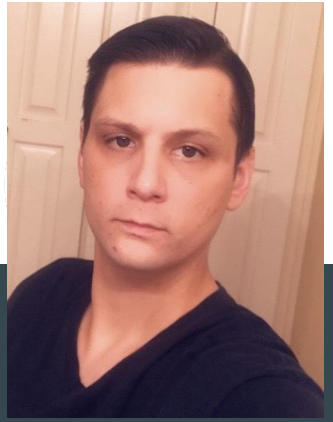
- Business Analyst support
- Scrum sprint
- Trello board
- Daily Standups

Business Analyst Team



Josh Robbins

Team Lead



Devin Kraus

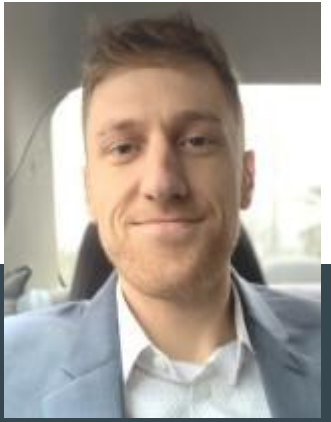
Business Analyst

Donald Mullen

Business Analyst

Zach Holt

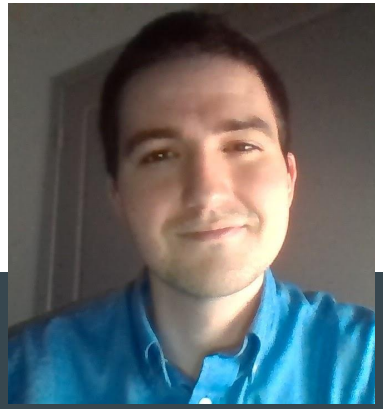
Business Analyst



Josh Zellner

Business Analyst

Development Team

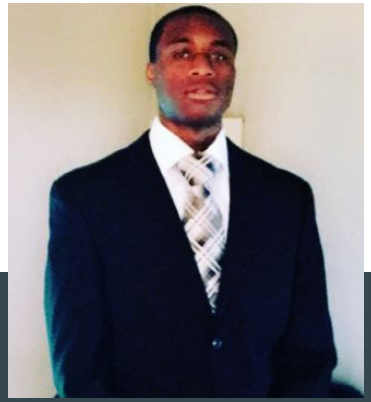


Paul Edwards

Team Lead

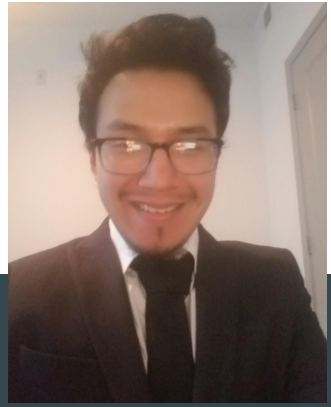
Shawn Halcomb

Developer



Kyran Bryant

Developer



Tyler Claypool

Developer

Questions?