Bar Hopper

Jacqueline Ramos, Will DeRoberts, Mark Paris, Diaeddin Motan

September 15, 2021

**INTRODUCTION**

Bar Hopper is the waze for nightlife! Find the grooviest club in town by collaborating with your friends to tag nightlife spots with real-time attributes such as ‘long line’ or ‘empty’.

* As a bar hopper I want to see bar activity so that I can make decisions for where I go to maximize my bar time.
* As a bar hopper I want to understand the atmosphere of bars so that I can find a bar to find my crowd.
* As a bar hopper I want to see COVID vaccination requirements so that I can find a bar that I feel comfortable in.

Our application will be successful if we can generate customer buyin to populate our real-time data and effectively drive decision making for our target user groups.

# ROLES AND RESPONSIBILITIES

Project Lead (Jacqueline Ramos)

Project Manager (Jacqueline Ramos)

Development Lead (Diaeddin Motan)

Developers (Diaeddin Motan, Will DeRoberts, Mark Paris, Jacqueline Ramos)

Infrastructure lead (Will DeRoberts)

Test Lead (Will DeRoberts)

Testers (Will DeRoberts, Diaeddin Motan)

Documentation (Jacqueline Ramos)

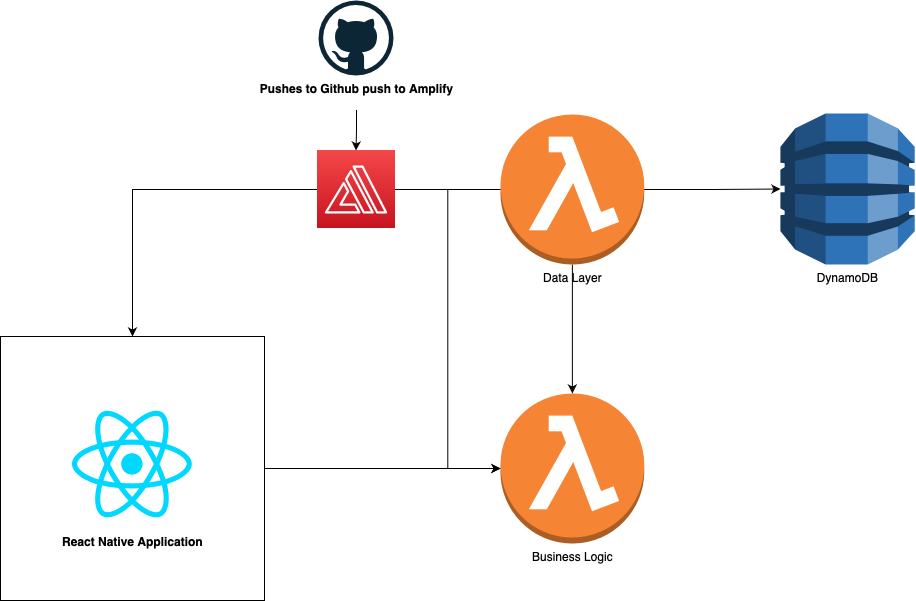
UX Designer (Mark Paris)

System Administrator (Will DeRoberts)

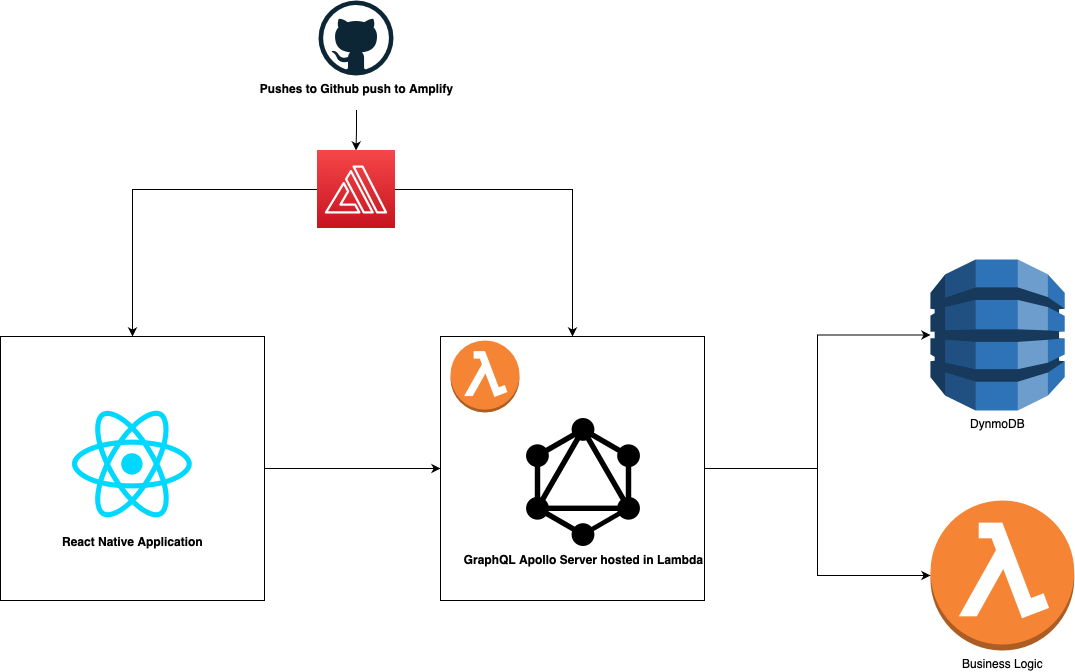
Requirements Resource (Jacqueline Ramos, Mark Paris)

# METHOD

* Software:
  + We will use either React Native or Swift for our mobile application frontend.
  + We will use Amplify (https://aws.amazon.com/amplify/) to link up FE and BE and to deploy our APIs to the cloud.
  + We will use GCP Auth for AuthN and profile management; this can integrate with any framework/cloud provider.
  + We plan to use Node.js and Javascript for our backend but may use Python following requirements analysis.
* Infrastructure:
  + We will be developing locally and pushing to github as described in the development process below
  + We’re going to host our APIs on AWS using lambdas.
  + We will be using a NoSQL database to facilitate geosearches. We are planning to use a hosted instance of MongoDB.
  + We may elect to use GraphQL following requirements analysis.



with GraphQL



* Development Process:
  + We will use Scrum as our Agile development method, using two week sprints. We will conduct zoom meetings to do sprint planning, demonstrations, and retrospectives. For standups we will use a daily text update for the last 24/next 24/blockers. We will meet in zoom or person at least once a week.
  + We will begin our project with requirements analysis and developing UX mocks and system architecture including database schema. Following completion of these milestones we will begin our scrums.
  + We’ll be conducting Code Reviews and push changes only after CR approval.
  + We will implement automated unit and integration tests which will run prior to deployment
* Build Plan:
  + We will use Pull Requests to conduct Code Reviews
  + We will define our backlog in GitHub issues
  + We will use Travis CI and Amplify to run automated test

# COMMUNICATION PLAN

## Working team meetings

Daily standups conducted over text and Zoom Scrum meetings at least once a week (most likely during class hours). Zoom sprint review meetings at the end of the two week sprint cycle.

## Status meetings

Status meetings are meetings where the project manager and/or project leader reports on status to the instructor. These will occur weekly over the duration of the project during scheduled class time with the entire team present so they can all speak to any questions of status to the instructor.

## Issues meetings

If a problem does arise, never surprise your manager. Schedule a meeting at his or her earliest convenience. This section describes how alerts will arise and the governance of when to trigger an alert – usually after a discussion at a working team meeting.

# TIMELINE AND MILESTONES

Look forward to the next 10 to 13 weeks and plan out a set of target milestones that the team expects to meet over the duration of the project, culminating in the final presentation at the end of the semester.

* 9/16/21 Project Kickoff
* 9/23/21 Technical Architecture Review
* Functional requirements defined
* Backlog Created
* Sprint 1
* MVP completed
* Sprint 2
* Testing complete
* Sprint 3
* 12/2/21 Final Demonstration

# RISKS

Describe any risks that you see for this project, and how they will be track and monitored.

* This project requires a data source to pull business information, esp lat/long and business name. Without this data source we will need to manually compile and enter this data.

# ASSUMPTIONS

* All establishments will be provided our data source
* Customers value real-time data about nightlife conditions