**CNG 495**

**Fall – 2022**

**Term Project Proposal**

by

Ibrahim Ozkan – 2456275

Adil Bozkurt Kebapcioglu – 2455954

**Milestones achieved**

**October 31 – November 6:**

* Determined the back-end requirements of the project:
* Database structure: the initial database structure can be seen in the entity relations diagram below.

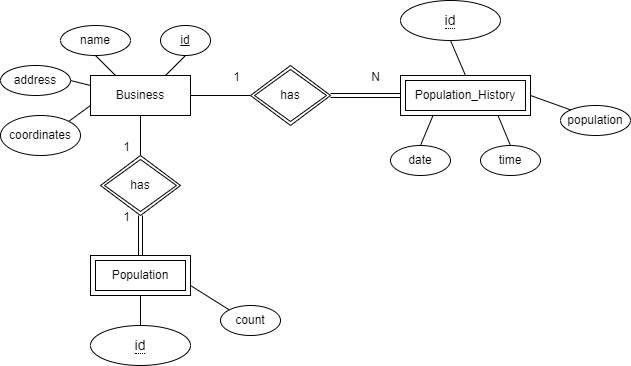


Figure 1. Entity Relation Diagram

* API requirements: the API should provide basic CRUD functionality to an admin to manage the businesses, allow the Arduino devices to increment and decrement the count of the population for a certain business, and provide the business data with the current population and population history data to the mobile client.
* Researched the integration of a Laravel project into a Linux based EC2 instance using RDS for database

**November 7 – November 13:**

* Laravel installation and project creation locally (<https://laravel.com/docs/7.x/installation>)
* Creation of Laravel Models (<https://laravel.com/docs/7.x/eloquent>)
* Setting up the routes (<https://laravel.com/docs/7.x/routing>)
* Setting up the controllers (<https://laravel.com/docs/7.x/controllers>)
* Creating the migrations (<https://laravel.com/docs/7.x/migrations>)
* Setting up the local testing environment:
  + Running XAMPP Apache and MySQL
  + Starting local development server for the Laravel project
  + Setting up Postman program
* Testing the API services locally via Postman
* Setting up GitHub locally and pushing the Laravel project folder into the “CNG495-F22-CloudComputing” repository.

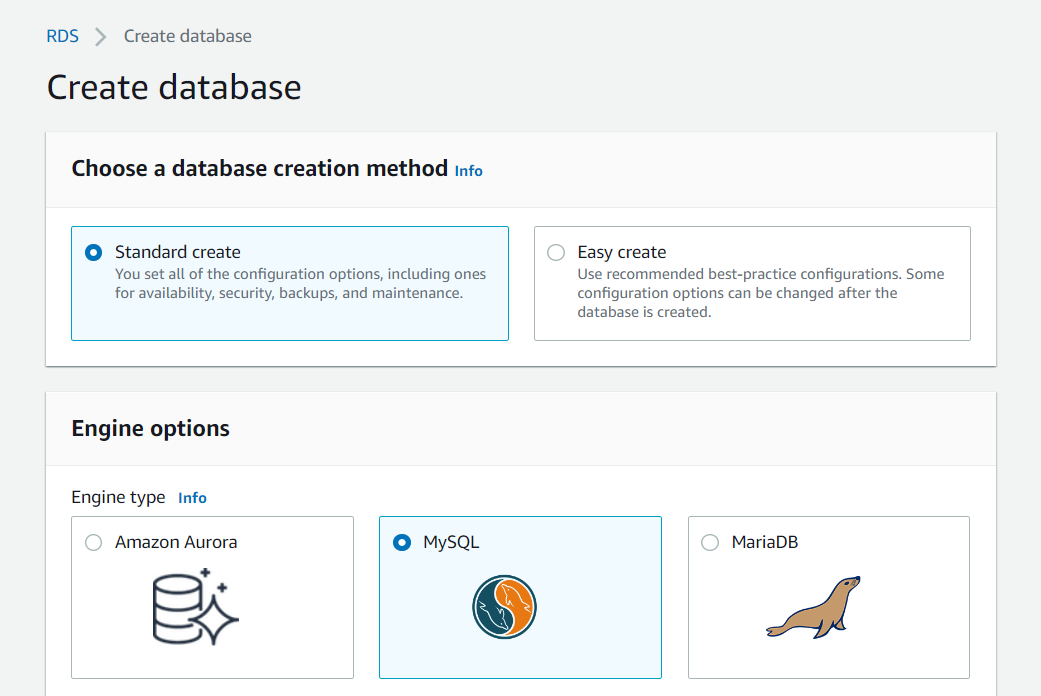
The aforementioned Laravel project can be found in the following GitHub cloning URL: <https://github.com/ibrahimozkn/CNG495-F22-CloudComputing.git>

**November 14 – November 20:**

* **AWS RDS creation:**

Go to AWS console -> RDS

Click create Database.



Choose Standard create and MySQL as seen above.

Graphical user interface, application

Description automatically generated

Choose MySQL version then Free tier.

Graphical user interface, text, application, email

Description automatically generated

Give a Database instance identifier, master username, and password.

Click Create Database on the bottom of the page.

* **AWS RDS configuration:**

Go to RDS -> Databases.

Click the created database.

Graphical user interface, text, application, Word

Description automatically generated

Under Connectivity & security, click the security group.

Graphical user interface, text, application

Description automatically generated

Navigate to Inbound rules and click Edit inbound rules.

Click Add rule then select MySQL/Aurora, select the EC2 instance, save rules.

Open the Laravel project file and navigate to .env file.

Text

Description automatically generated

Edit the .env file and add database name, username, and password that was used in the RDS setup earlier as seen in the figure above. The DB\_HOST can be obtained as follows: go to RDS -> databases, choose your database. Under Connectivity & security, there is the endpoint. Copy and paste the endpoint to the DB\_HOST variable. Save the .env file. Push the updated project folder to GitHub.

* **Deploy the Laravel file on EC2 instance:**
  + Set up Laravel on the Linux instance using the following command: **composer global require "laravel/installer=~1.1"**
  + Create the project directory
  + GitHub pull the Laravel project into the directory
  + Run the project using the following command while within the project directory:

**php artisan serve --host 0.0.0.0**

The server is now functional, and the API can be called from the following address:

“http://ec2-3-86-218-135.compute-1.amazonaws.com:8000/api/”

**Milestones remaining**

|  |  |  |  |
| --- | --- | --- | --- |
| Milestone No | Week | Description | Responsible Student(s) |
| 1 | November 28 - December 4 | Arduino UNO Configuration & Installation | Ibrahim Ozkan, Adil B. Kebapcioglu |
| 2 | December 5 - December 11 | Fully implementation of API Database | Adil B. Kebapcioglu |
| 3 | December 12 - December 18 | Fully implementation of API | Adil B. Kebapcioglu |
| 4 | December 19 - December 25 | Design UI of the mobile application for all functionalities | Ibrahim Ozkan |
| 5 | December 26 - January 1 | Setup Google Maps API for map view in mobile application | Ibrahim Ozkan |
| 6 | January 2 - January 8 | Make mobile application fully functioning | Ibrahim Ozkan |
| 7 | January 9 - January 15 | Connect all components (API, Arduino, Mobile Applications) together and make system ready to use | Ibrahim Ozkan, Adil B. Kebapcioglu |

**Delivery List**

Materials that will be delivered when project finishes are listed below:

* Code for API (Laravel)
* Code for Mobile Application (Flutter)
* Instructions for setting up EC2 server, Google API and Laravel Deployment
* APK file for Mobile Application
* Database file of demo
* Arduino code
* Arduino UNO setup instructions