Team GeoUnity GeoAttendance

Bence Danko, Da Thao Trinh (Lead)

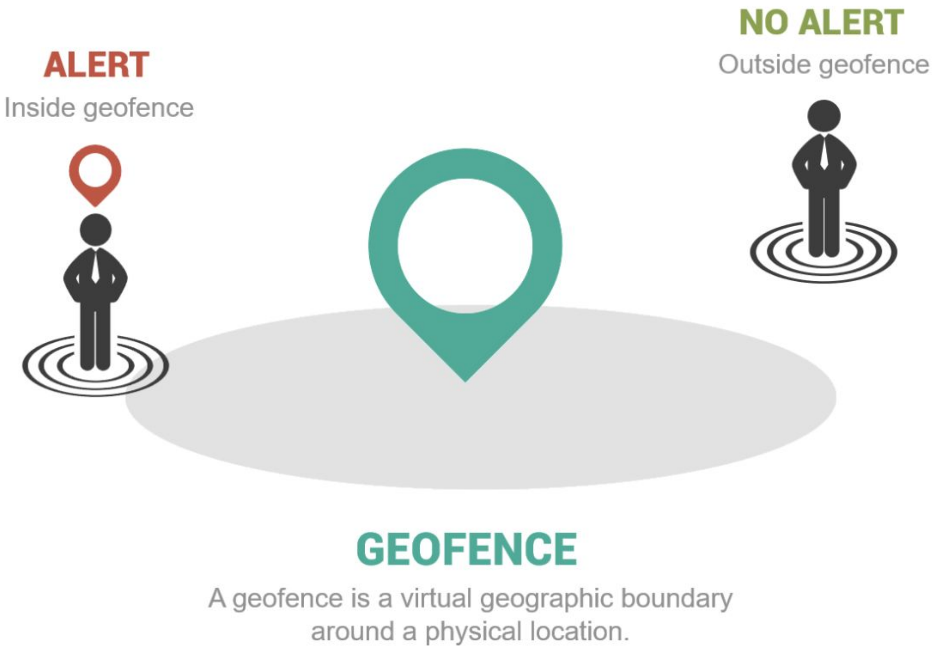
Github URL: https://github.com/bencejdanko/GeoAttendance

Project Description

GEO: short for geography or location.

FENCE: is usually a boundary line such as between you and your neighbor.

GeoAttendance leverages geofencing technology to enhance the accuracy and convenience of attendance tracking.



A geofence is a virtual geographic boundary around a physical location.

Project Description

- **Users within boundaries will be able to check-in**

- **Upon checking-in, the users will receive an email which confirms their attendance status**

- **Real-time updates on attendance and historical attendance**

Job Assignments

Task	Member
APIs	Bence
Backend	Bence
UI/UX	Da Thao
Frontend	Da Thao
Setting up development environments & Testing	Bence & Da Thao

Milestones

Milestone 1 (2/5 - 2/16): Initialize technology frontend, backend development environments.

Milestone 2 (2/12 - 2/23): Build User API, UI for Login and Registration Forms.

Milestone 3 (2/26 - 3/8): Implement Geolib library, UI for Attendance Check-ins & Testing.

Milestone 4 (3/11 - 3/22): Implement Attendance API & Testing.

Milestone 5 (3/25 - 4/5): Complete prototype & Testing.

Milestone 6 (4/8 - 4/16): Deployment.

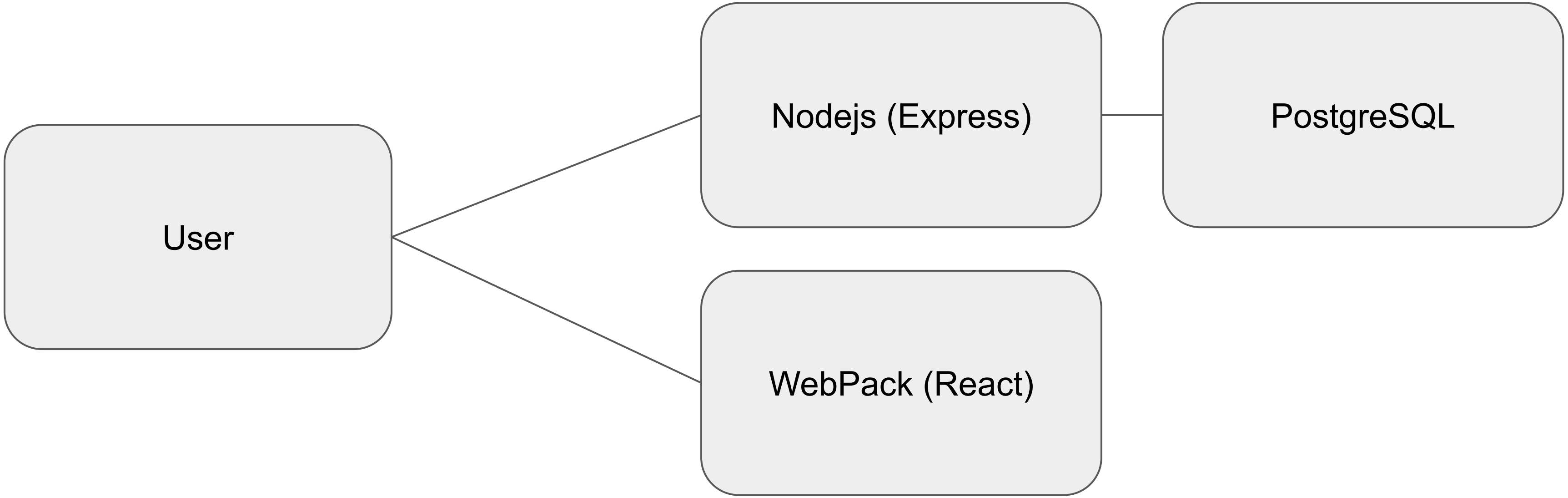
(Stretch) Milestone 7 (4/17 - End of semester): Integrate Facial Recognition Technology.

Frontend/Backend

Technologies Used

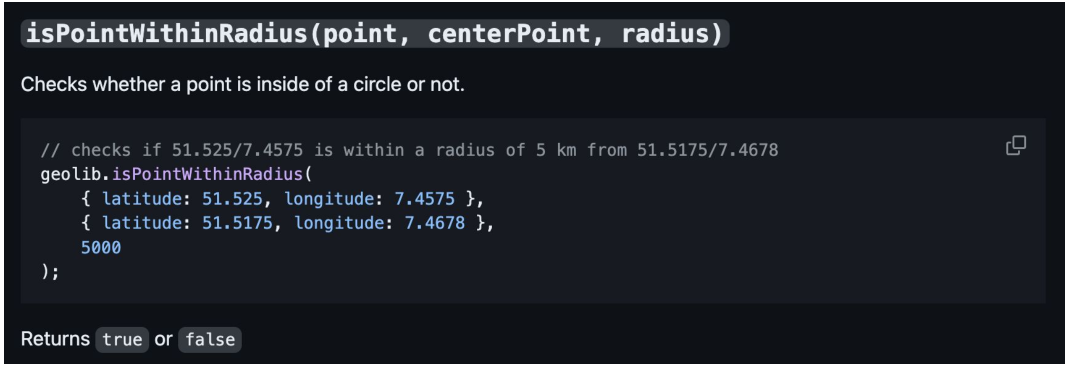
[Frontend] HTML, CSS, React, JavaScript

[Backend] Nodejs, PostgreSQL



Algorithms/AI Schemes

GeoLib Library (https://github.com/manuelbieh/geolib)



`isPointWithinRadius(point, centerPoint, radius)`

Checks whether a point is inside of a circle or not.

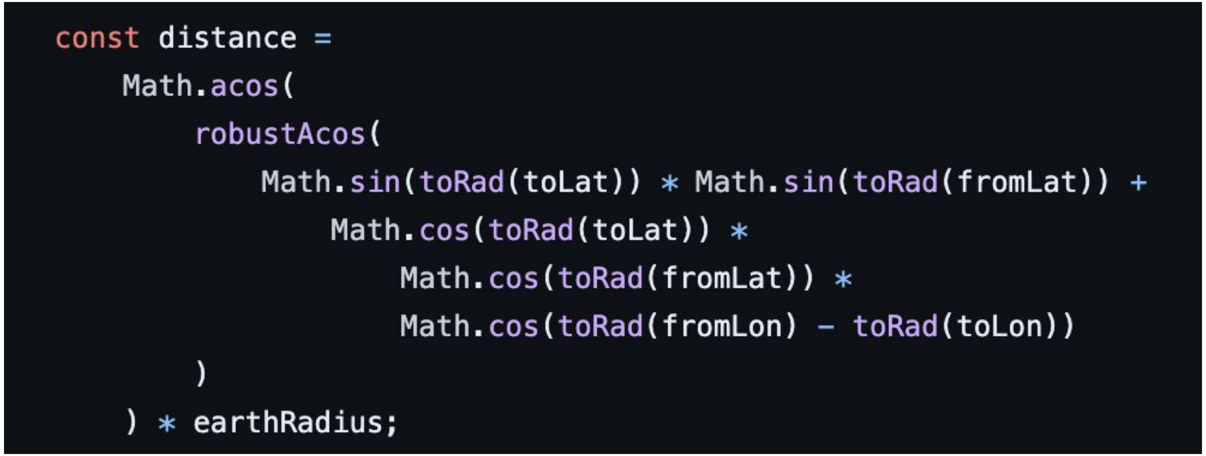
```
// checks if 51.525/7.4575 is within a radius of 5 km from 51.5175/7.4678
geolib.isPointWithinRadius(
    { latitude: 51.525, longitude: 7.4575 },
    { latitude: 51.5175, longitude: 7.4678 },
    5000
);
```

Returns `true` or `false`

- Correctness: tested with various inputs, including points inside and outside the circle, as well as edge cases such as at the center of the circle or when the inputs are undefined.
- Also handle negative coordinates and non-integer values.

Algorithms/AI Schemes

GeoLib Library (https://github.com/manuelbieh/geolib)



```
const distance =
    Math.acos(
        robustAcos(
            Math.sin(toRad(toLat)) * Math.sin(toRad(fromLat)) +
                Math.cos(toRad(toLat)) *
                    Math.cos(toRad(fromLat)) *
                    Math.cos(toRad(fromLon) - toRad(toLon))
        )
    ) * earthRadius;
```

- Complexity: O(1) - Haversine formula - Basic arithmetic operations.

Algorithms/AI Schemes

Possible Additional Algorithms (Stretch):

- Use facial recognition technology to improve the accuracy of the attendance tracking process

- Historical attendance data
    - Calculate simple trends in attendance
    - Identify outlier students

Market Space

- Streamlines attendance tracking for professors and other hosts

- Allows attendees to easily find meeting spots through their phone, without the hassle of going through manual instructions

- Host can update events, or modify sign-ups – All attendees simultaneously track any changes or updates immediately

Feedback: JumpAndLaunch

- It is very useful for teachers and instructors
- How are you going to specify a location if it's in a building with more than 1 floor?
  - We can add a location description with misc. information if needed.
- How can you track to see if the student is on the right floor?
- Are you planning on doing something like Bluetooth range?
- Can facial recognition recognize photos or masks from real faces?