Abstract:

Our project is a take on the game Geoguesser adapted to the OSU campus. Our game will be a web-based application where users are shown a 360 image from around the Oregon State University Campus and asked to guess the location of the image taken. Our major developmental goals are, a user authentication system, user profile, leaderboard system, and integrated locational data.

1. Team info

Provide a concise summary of the project team and project artifacts. Specifically:
List each team member and their role in the project.

- Blake - Database
- Joy - API/ data collection
- Kevin Tran - Backend
- Kevin Nguyen - Backend
- Sam - UI Designer/Dev
- Gavin - Frontend
- Lukas - API/data collection

Role Description:

- Database:
  Creation of database and integration with application
- API/data collection:
  Collection of photos and integration of Google Maps API's
- Backend Developer:
  Creation of server and backend logic
- Frontend Developer:
  Creation of website and integration of website with backend processes and user interface
- UI designer/Dev:
  Creation of user interface and frontend structure

Communication/Tools:

Link to each project-relevant artifact such as your git repo (this can be empty for now).
https://github.com/blakethomas12/Team-18

List communication channels/tools and establish the rules for communication.
Discord server
Rules:

- Communicate any changes
- Members report when they can't be present
- Give everyone a chance to talk
- Ask for help when there's an issue with their task

2. Product description

Goal:

The goal of this project is to help students engage more with the OSU campus by playing a game while familiarizing themselves with the different classrooms and buildings.

Current practice: How is it done today, and what are the limits of current practice?

The inspiration for this project, Geoguesser, is designed around trying to guess the general area around the world that the player is shown, which limits how accurate guessing can be within smaller areas and distances.

Novelty: What is new in your approach and why do you think it will be more successful than other approaches? Do not reinvent the wheel or reimplement something that already exists, unless your approach is different.

This would be a project that works on a more local scale, one that is more applicable to the current living environment that the group and other students at OSU are currently familiar with, thus encouraging a more observant approach to everyday student life.

Effects: Who cares? If you are successful, what difference will it make?

Students at Oregon State University will seek this game for simple, but engaging entertainment. If the product is successful, students will have more knowledge about the OSU campus and its landmarks.

Technical approach: Briefly describe your proposed technical approach. This may include what system architecture, technologies, and tools you may use.

Create a website connected to a custom database of locations and the Google Street View API. We serve a random picture or location from the database to the user. The is linked to a point on the map. The user inputs a point on the map. This is analyzed, and the user is given a certain amount of points for how close they are. These points are stored in the database and added to the leaderboard, which is displayed at the end of the game.

Risks: What is the single most serious challenge or risk you foresee with developing your project on time? How will you minimize or mitigate the risk? Don't state generic risks that would be equally applicable to any project, like "we might run out of time".

There may be limits to the Google Maps API, but as long as we stay under the limit of requests per month, we should be able to use it for free.

Use Cases:

1. Blake
    a. Actors: User
    b. Triggers: Change of Username
    c. Preconditions: User exists, new name is different than old name, new name is not taken
    d. Postconditions (success scenario): Username has been changed
    e. List of steps (success scenario)
        1. User requests change of username
        2. User asked for new username
        3. Check if old username = new username, reprompt if true
        4. Check if new username is taken, reprompt if true
        5. Username passes checks and is updated in database
    f. Extensions/variations of the success scenario
        1. User fails check if old name = new name, then corrects name
        2. User fails check if name is taken, then enters a free name
    g. Exceptions: failure conditions and scenarios
        1. User does not exist
        2. Database is unreachable(on update or checks)
2. Lukas
    a. Actors: User
    b. Triggers: Start new round
    c. Preconditions:
    d. Postconditions (success scenario): User is served location
    e. List of steps (success scenario)
        i. User starts new round
        ii. Random location is pulled from database
        iii. Corresponding streetview is requested from API
        iv. Correct streetview is shown to user
    f. Extensions/variations of the success scenario
        i.
    g. Exceptions: failure conditions and scenarios
        i. Database is unreachable
        ii. API is unreachable
        iii. Incorrect location
        iv. Incorrect streetview
3. Kevin Tran
    a. Actors: User
    b. Triggers: Submits Location
    c. Preconditions: The round has started and user is presented with the location
    d. Postconditions (success scenario): User guess is recorded and tallied
    e. List of steps (success scenario)
        i. User clicks a location on the map and submits a guess
        ii. Backend received the guess coordinates and the correct coordinates

iii. Backend calculates the distance between two locations
iv. Points are rewarded and the user's score is stored in a database.

f. Extentions/Variations of the success scenario
   i. User are shown two points on the map that correspond to their guess and the correct answer.

g. Exceptions: Failure Conditions and Scenarios
   i. Invalid Guess
   ii. API unreachable
   iii. Database Unreachable

4. Joy Lim
   a. Actors: User
   b. Triggers: View Leaderboard
   c. Preconditions:
      i. User must be logged in
      ii. System has recorded scores for at least one player
      iii. Leaderboard data is up-to-date
   d. Postconditions (success scenario):
      i. Leaderboard option is displayed in main menu
      ii. Leaderboard will display the top 5 best players' username and scores
   e. List of steps (success scenario)
      i. User selects "Leaderboard" option from main menu
      ii. System will retrieve user scores from leaderboard data
      iii. Scores will be sorted in descending order
      iv. First 5 scores will be displayed to the user
   f. Extensions/variations of the success scenario
      i. User leaves before scores are recorded
   g. Exceptions: failure conditions and scenarios
      i. Leaderboard is empty meaning the database contains no scores
5. Gavin Fifer
   a. Actors: User
   b. Triggers: Login
   c. Preconditions: no account is currently logged in
   d. Postconditions (success scenario): The user is logged into their account
   e. List of steps (success scenario)
      i. User clicks the Login button
      ii. System will display the Username and Password fields
      iii. User fills in Username and Password fields and clicks Submit button
      iv. System validates the Username and Password entered
      v. System returns User to the main page now logged in
   f. Extensions/variations of the success scenario
      i. The Login button is now replaced with the Logout button
      ii. The User can see their Username next to the Logout button
   g. Exceptions:
      i. Invalid Username/Password

    ii. The account is already logged into somewhere else
    iii. Database with Usernames/Passwords cannot be reached

6.
    a. Actors: User
    b. Triggers: New user registration
    c. Preconditions:
        i. User isn't currently logged in
        ii. User provides valid registration details
    d. Postconditions (success scenario):
        i. User account is created and stored in the database
    e. List of steps (success scenario):
        i. User selects the "new account" button from the menu
        ii. System prompts user for username, email, and password
        iii. User enters details and submits
        iv. System checks if submitted details already exist in database
        v. The check passes and the new account is created and stored
        vi. The user is automatically logged into their account
    f. Extensions/variations of the success scenario:
        i. User inputs an option profile picture
        ii. A "forgot password" routine is initiated
    g. Exceptions:
        i. Username/email already exists in database
        ii. Database is currently unavailable
7. Kevin Nguyen
    a. Actors: User
    b. Triggers: Reset Password
    c. Preconditions:
        i. User account already exists
        ii. User provides the correct email for the account
    d. Postconditions (success scenario): User is able to reset their password through the link sent in their email
    e. List of steps (success scenario)
        i. User clicks on "Forgot Password" in the login page
        ii. System asks for the email connected to the account
        iii. User enters in the email and clicks submit
        iv. System verifies that email exists in the database
        v. A reset link is sent to the email address
        vi. User clicks on the links, enters a new password, and submits it
        vii. System updates the new password in the database
        viii. User is back in the login page and enters their new password and logs in
    f. Extentions/Variations of the success scenario
        i. User enters an invalid email address and a message pops up saying to try again.
    g. Exceptions: Failure Conditions and Scenarios

i. Database is down
ii. Email address is not found

Non-Functional Requirements:

1. Databases must be able to grow and serve a growing user base.
2. Photos must not include people's identities
3. Front end must be easy to use and simple to learn

External requirements:

- Since our product is web based the url must be public and accessible to others
- Our project will also include a set of instructions to set up a server and start to program

3. Additionally, add the following:

Major Features:

- Location Data
- Leaderboard
- User Authentication
- User Profiles

Stretch Goals

- Multi-User competition
- User Submissions
- Reward System

4. Timeline

Week 1: project idea, role creations, research into roles, tools and common practices
Week 2: basic foundation of website created(basic server uses, interactable front end)
Week 3: database and api integration, data gathering, game logic
Week 4: major features implementation, data gathering
Week 5: finishing major features, data gathering
Week 6: Final touches to all parts, adding in the smaller details(ui fixes, visual components, cleaning up code)
Week 7: Major testing, test all components for bugs
Week 8: Major testing, test all components for bugs
Week 9: Final testing, preparation for final version