
CTRL-Freaks presents:
GeoVision AI
Jessica Babos, Andrew Hua,
Kenneth Riles, Shadi Zgheib
& Zachary Mclaughlin


REQUIREMENTS

What is the purpose of this software?

- This software will function as an artificial intelligence that will possess the ability to identify specific locations from scanning a photograph of a distinct area or object.
- It will do so by using a built-in GIS system that pinpoints the precise location and coordinates within a cartographical grid (latitude and longitude).
- Furthermore, it can assess unique artifacts, sculptures or monuments to retrieve their locations.

Who is it intended for?

- Travel enthusiasts (Tourists)
- Photographers
- Cultural Heritage Researchers
- Historians
- Archaeologists
- Content Creators and Social Media Influencers (Streamers)
- Curators and Museum Professionals
- Geography and Earth Science Students or Educators

Goals for this software

- Providing curious end-users with the ability to experience a new, cutting-edge artificial intelligence that can provide them with instantaneous solutions to researching the location of an image on the internet or taken by another user.
- It can especially help tourists search for enticing sites to visit from gazing at an image on a billboard or website.
- In addition, this software will be accessible on multiple platforms (PC (Windows) and Mobile (Android)).


Specifications

Key features

Required Features

- User logs into or creates account
- Display invalid user credentials
- User account is authenticated via email verification
- Passwords are encrypted
- Internal database stores user credentials
- User has the ability to upload or drag and drop image file from file system
- Image file formats (.jpg, .png, .gif, .bmp, etc..) supported
- External database stores image files from the internet
- Location results are displayed if found or not found
- Error message is displayed if search and photo-matching algorithm fails
- The AI can detect objects (landmarks, artifacts, plantlife, etc..) within images
- database

Desired Features

- Software supports multiple platforms (desktop and mobile)
- Multiple users can use this AI simultaneously
- Users can choose to save credentials in cloud for quicker access
- Users can update account credentials
- Image in results screen has increased resolution, quality and lighting contrast
- GUI is specific and readable
- GUI fits on different platforms
- Results screen displays full location (city, state, region, country, continent), distance coordinates (latitude and longitude) and facts about the location
- Location search algorithm is fast and optimized for multi-user use

Additional features

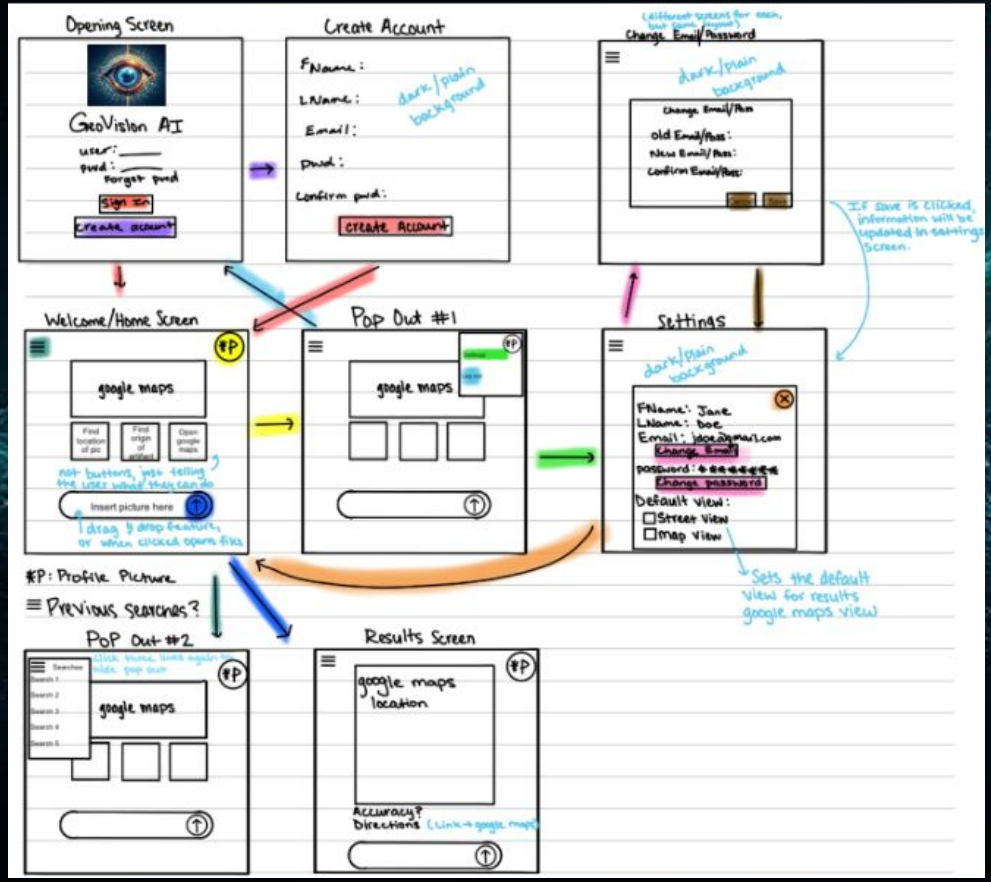
- If no local match is detected within search, then Azure AI API will search within cloud
- User can change from map-view to street-view
- User can interact with the map or street-view widget to explore location
- GUI will be unique and defining the software as its own property

Group assigned tasks

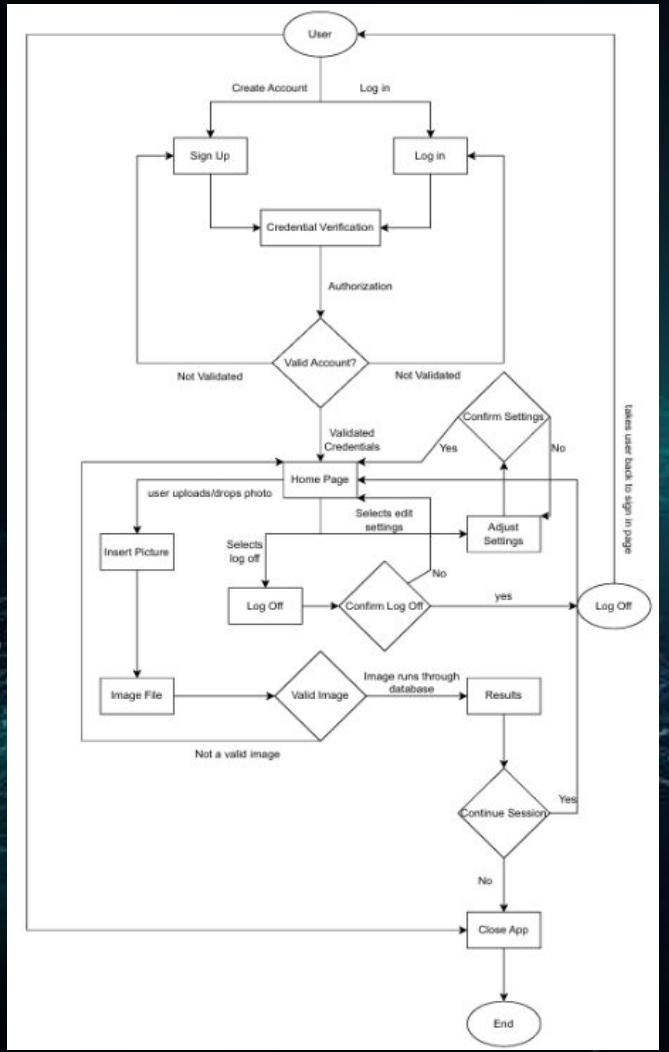
1. Jessica Babos: Project Manager, Editor, Tester, Analyst, Application Developer. ShowResults, LogOut
2. Andrew Hua: Application Developer, Tester, Analyst. CreateAccount
3. Kenneth Riles: Backend developer, Researcher, Tester. UpdateAccountInfo
4. Shadi Zgheib: Backend, ML/AI developer, Tester. Search, InsertImage
5. Zachary Mclaughlin: Full Stack Developer, Framework Tester. ForgotPassword, ResumeSearch


Design

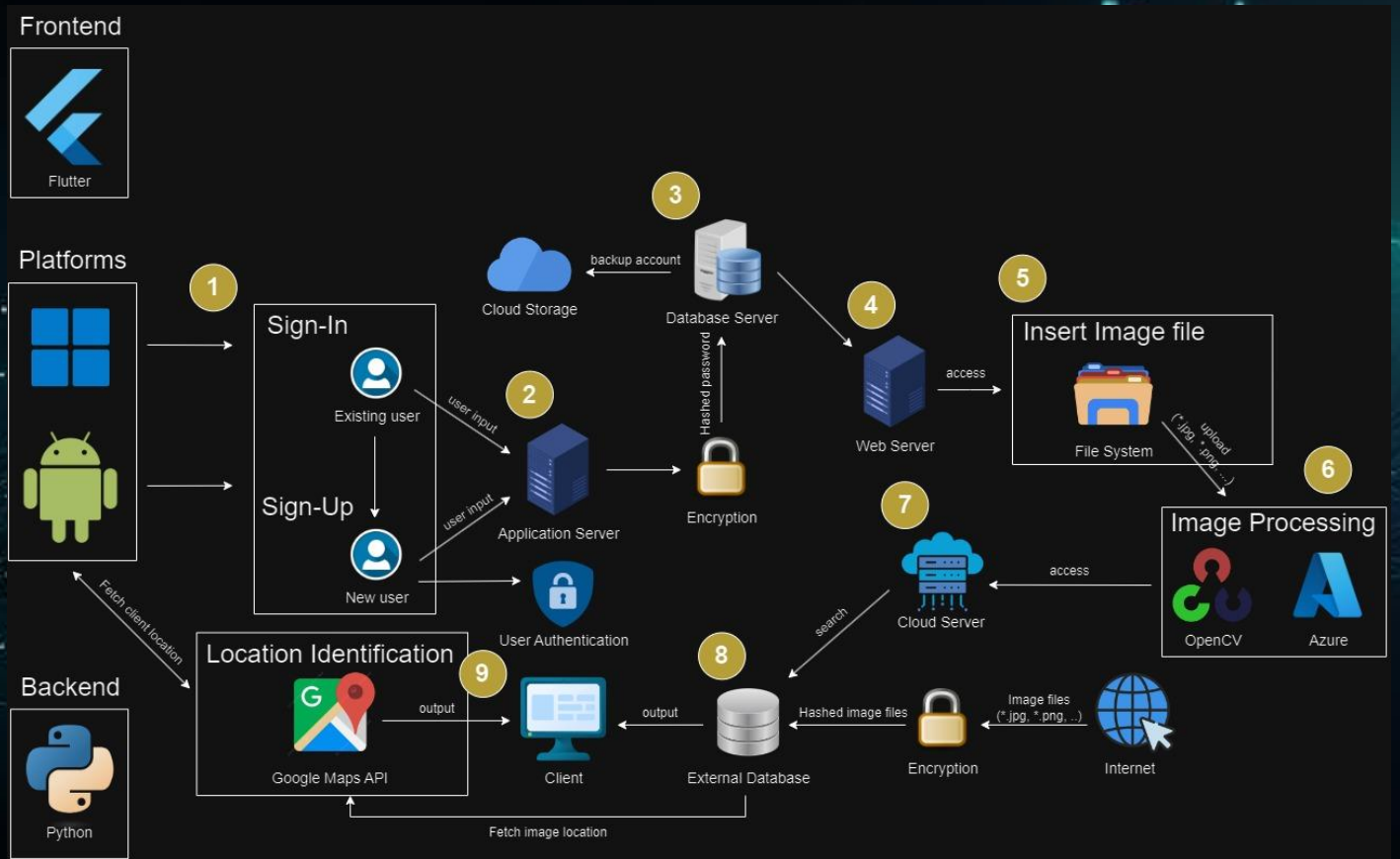
Application design



Process-Flow Diagram



Application Architecture




SOFTWARE
IMPLEMENTATION

Technologies/Tools

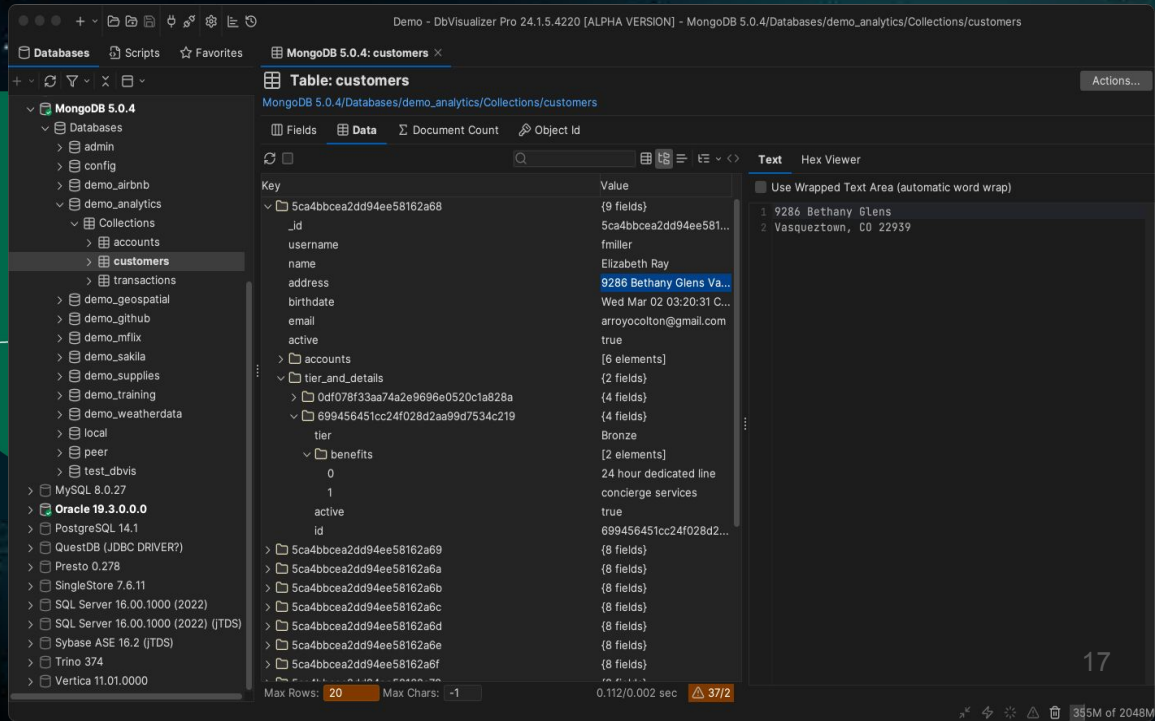
- **Frontend:** PYQT5
- **Backend:** Python (OpenCV, Azure AI Vision integration)
- **Database:** MongoDB
- **AI Processing:** OpenCV, Azure AI Vision API
- **Mapping Service:** Google Maps API
- **Package Manager:** pip (for Python dependencies)

Algorithms

- User privacy
  - A one-way hashing algorithm will encrypt user information (usernames, passwords), user requests, image files and server communications.
  - A cloud computing algorithm will help establish a secondary layer of user data protection.
- AI Image Processing
  - A combination of image scanning and object detection will be used to teach the AI to recognize certain geographic layouts in images and compare them within the internal database.
  - External database of images imported from the internet will be compared with via cloud computing.
- Geographic locating
  - The Google Maps API uses Dijkstra's algorithm to find the shortest path from the starting point to the location found.
  - Uses Curve algorithm (Ramer-Douglas-Peucker) to estimate and record the area coordinates of the image from the map. This data will help in pinpointing the precise location.

Database

- Internal and external database:
  - MongoDB



Security risks and concerns

- The user could lose their account information from a cyber attack if they decide to not save it into the cloud.
- An unauthenticated user could get blocked from some client-specified services (such as changing their email and password).
- If a user decides to save their username and password in cookies (so they won't have to login next time), then their data is at risk of getting stolen.
- Server maintenance could delay or disable user progress in an application.
- Users with malicious intent could upload tampered images to manipulate the program or get unintended information.
- The app may inadvertently store or use copyrighted images.
- Storing user uploaded images in a publicly accessible location could lead to a data breach.
- Google Maps API could expose the client's location to hackers attempting to spy on the user.
- Users may upload sensitive or personal images that contain identifiable information or sensitive locations.

How to resolve those issues

- Implement a secure password policy, and require MFA (Multi-Factor Authentication).
  - Make users verify that it is them.
- Set up two-factor authentication (private PIN number or security questions) for logging into their account.
- Schedule maintenance during periods of low user activity, such as late at night when the fewest users are likely to be affected.
- Implement image integrity checks like a hash to detect tampering.
- Securing the image by encryption and or restricting access to the location for only authorized users.
- Disable location tracking on a user when they are not using the app.
- Ensure end-to-end encryption.

Legal and Ethical Concerns

Ethical

- AI models might incorrectly identify locations, leading to potential misinformation. This could have serious consequences, such as misinforming travelers, researchers, or individuals who rely on the accuracy of the information.
- If the internal database or object detection algorithm has an inherent bias, certain types of buildings, artifacts, or landscapes might be overrepresented or underrepresented, leading to skewed results.
- The program could inadvertently expose private or sensitive locations (e.g., homes, private properties, or restricted government areas) when identifying buildings or landscapes. Identifying such locations without the owner's consent could lead to privacy violations.

Legal

- Users could upload/leak pornographic/scandalous images online.
- If the program collects or processes images of individuals, buildings, or other personal data, it could fall under regulations like the General Data Protection Regulation (GDPR) or California Consumer Privacy Act (CCPA).
- If the program misidentifies a location or provides incorrect information(e.g., directing users to dangerous locations), the developers could be held legally liable for any harm caused.
- Users or property owners may want to request that their images or locations be removed from the internal database, invoking their right to be forgotten under laws such as the GDPR.

How are they solved?

- Continuously update and improve the model's accuracy, and clearly communicate to users that the results may not always be 100% accurate.
- Ensure that the internal database is diverse and represents a wide range of locations and cultures. Audit the algorithm for biases and inaccuracies regularly.
- Implement safeguards that exclude sensitive or private locations from being identified or shown. Provide users with options to flag certain images or request removal from the database.
- Use real-time image scanning to analyze uploaded content before it's accepted. If inappropriate content is detected, the upload is rejected immediately, and the user receives an error message.
- Comply with data protection laws by implementing data anonymization techniques (ex. blurring faces, license plates, addresses), obtaining explicit consent from users before processing personal data, and allowing users to request the deletion of their data. Provide a clear privacy policy that outlines data usage, retention, and user rights.
- Provide a disclaimer that makes it clear that the program may not always provide accurate results and that users should verify information independently. Limit liability through well drafted terms of service and disclaimers.
- Create a process for users to request the removal of specific images from the internal database. Ensure that the program respects users' rights to control their data.


~Demo~

That concludes our presentation!

Please feel free to ask us any questions!

~Thank you!~