Cameron Dugan, Josh Gifford, Wilder Scott

Professor Salem Othman

Senior Project

3 August 2024

NabGo: Final Documentation

Overview

Our goal is to create a Go website for beginners studying Go books, videos, or websites. It is also intended for players studying by playing physical games of Go. Our project allows those players to convert their games and studies into digital representations that can be analyzed both by our application or by any existing Go software. The ultimate goal of the project is to achieve what [chessvision.ai](http://chessvision.ai) has done for the accessibility of chess, but for the game of Go.

Purpose

Go, one of the oldest board games in the world, traces its history back to ancient China. Since then, it has spread throughout the world and is still very popular today, particularly in Asia. It is a game with a simple concept, but it requires an immense amount of strategy and skill to become a high level player. Because of this, it can be overwhelming for many beginners. This project aims to provide resources to help beginners enhance their knowledge and enjoyment of Go.

Summary

Using the NabGo system, the user will have the ability to do the following:

* Log in using Google account
* Upload an image of Go board and have the image reproduced digitally
* Be able to edit the digital board before continuing to account for any inaccuracies
* Continue their game of Go on the NabGo site
* Copy the Smart Game Format (SGF) of board for use on external Go resources
* Access analysis tools including current win probability, score, and suggested moves
* Interact with chat bot for assistance with game rules, strategy, etc.

Software Tools

The following lists the software tools utilized for the completion of this project, along with a brief description of their use.

* **Languages**
  + Front End
    - HTML/CSS
      * Create and style web page
    - JavaScript
      * Enable front end functionality and send HTTP requests to back end for image prediction, analysis, etc.
  + Back End
    - Go
      * Receive and execute HTTP requests from front end and return results
    - Python
      * Execute image prediction and analysis
* **Acquiring Datasets**
  + Roboflow Universe
* **Object Detection**
  + YOLOv8n from Ultralytics
    - Train using acquired datasets and save the best model for use within application
* **Game Analysis**
  + KataGo AI Player
    - Utilize analysis engine to provide user feedback on gameplay
* **Chat Bot**
  + Meta Llama3 via Ollama
    - Allow user to access this large language model (LLM) for assistance with game
* **Data Storage**
  + EdgeDB
    - Keep track of site access
* **Development Environments**
  + NeoVim
  + Visual Studio Code
* **Version Control**
  + GitHub

Pre-Existing Components

Much of our project consisted of integrating existing libraries and tools into our application to ensure maximum performance. The following are the major tools used for the implementation of this project.

1. KataGo AI Go Player (Source: https://github.com/lightvector/KataGo)

The analysis engine provided will be used for faster game analysis. KataGo has a public-facing GitHub repository with released zips which we can use for integration.

1. Go Game Datasets: acquired via universe.roboflow.com

Our datasets will be pulled from Roboflow Universe. These datasets will be cleaned and then utilized to train our aforementioned neural networks.

1. YOLOv8n CNN provided by Ultralytics (Source: https://github.com/ultralytics/ultralytics)

The utilization of existing AI vision models, such as YOLOv8n, can be fine-tuned to improve the quality of our product.

1. Meta Llama3 via Ollama

This large language model will be the basis for our Gollama chat bot.

1. SGF library (Source: https://github.com/rooklift/sgf)

This library is utilized for converting the digital game board to SGF format.

Design Decisions

The following are the major design decisions that were made throughout the course of the implementation.

1. Market targeting
   1. We have decided to build our product with a target market of those who are beginners with the game of Go. Not focusing on expert-level analysis allows us some flexibility in the development process.
2. User-friendliness
   1. We have prioritized user-friendliness in our application. Using our product, it should be easy for players to continue their studying of the game. The application should also run efficiently and require little user interaction to acquire the desired information.
3. Web-based application
   1. Our application will be web-based. This will eliminate the need to consider having multiple ports for different devices. We will ensure that we test our application using multiple browsers, ensuring that all users will have a similar experience.
4. Use of EdgeDB
   1. For data storage, we have decided to use EdgeDB. This will be easier to work with than a traditional SQL database and allows for the implementation of sign-in options for our application if we choose to move in that direction.
5. Use of Go for back end
   1. Go is a relatively new programming language from Google. It aims to be a comfy C alternative offering better tooling for concurrent programming and web server programming. For these reasons, it felt like the best choice for our purposes.
6. Moving back end hosting to Wentworth server
   1. While integrating KataGo’s analysis engine into NabGo, we encountered a significant issue with our self-hosted server. We were only able to analyze 13 moves per minute on this server, which made efficient delivery of data to the user nearly impossible. Following this, we made the decision to move our back end onto a Wentworth-hosted server. This change proved crucial to our implementation. Not only were we able to analyze over 600 moves per minute using this server, but it gave us the overhead needed to include the Gollama chat bot in our project.
7. Addition of Ollama chat bot
   1. We decided to include this feature as it provides the user with another resource for sharpening their skills with Go. One of our group members had worked on a similar feature recently, and we decided that our project could benefit from this tool.

Diagrams

This section shows several diagrams which are helpful in explaining the NabGo application.

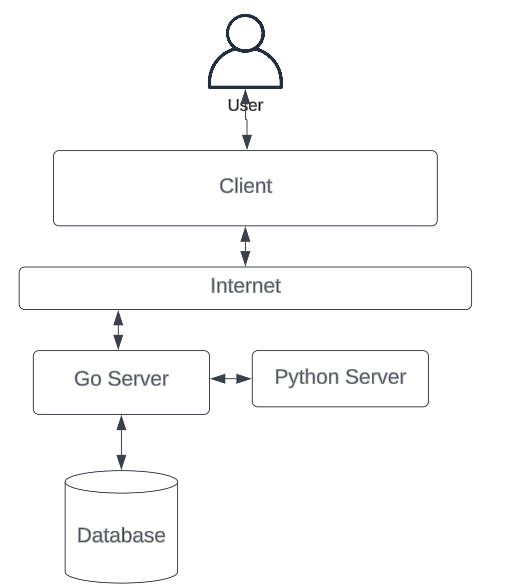


Figure 1: NabGo architecture diagram

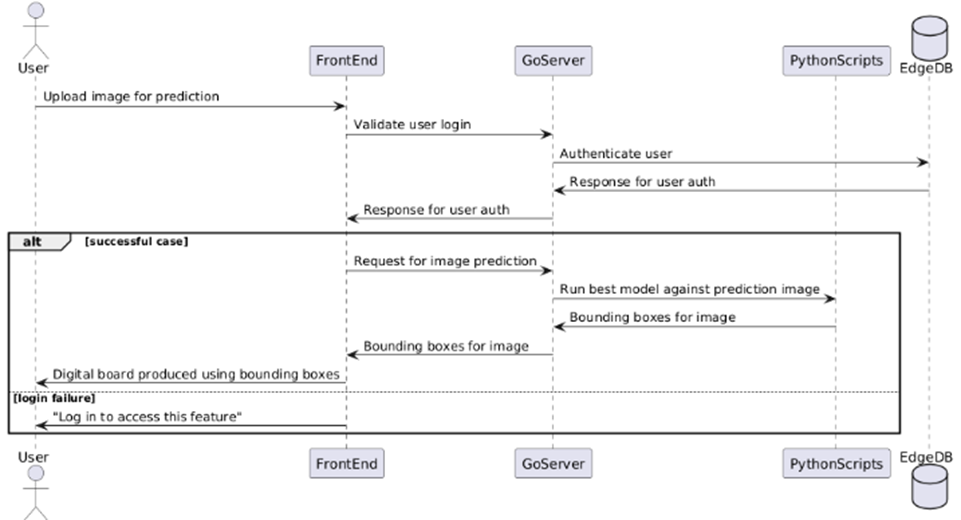


Figure 2: Sequence diagram for image upload and reproduction on digital board

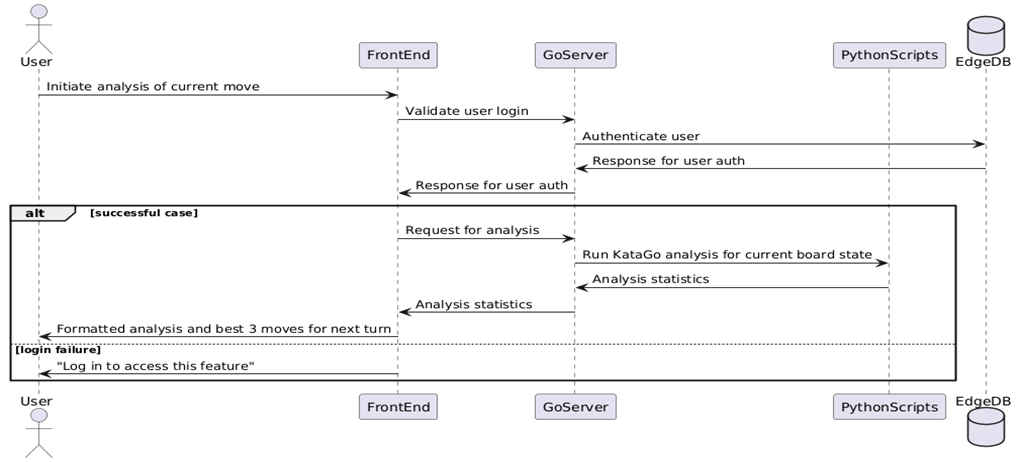


Figure 3: Sequence diagram for KataGo analysis

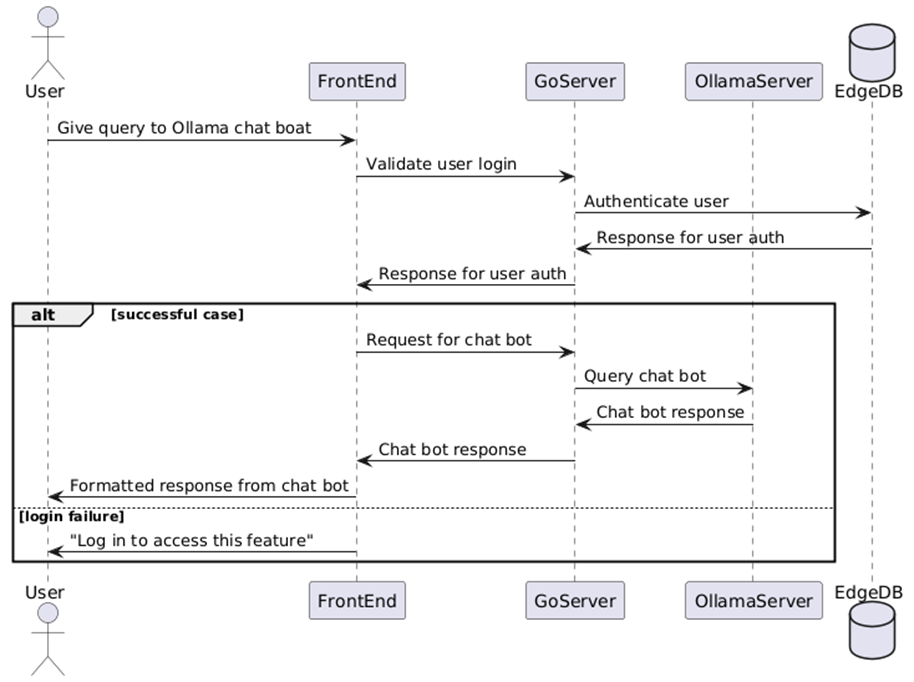


Figure 4: Sequence diagram for interaction with Gollama chat bot

Milestones

The following were milestones during the implementation in order of completion.

1. Set up of digital Go board
2. Set up of Go back end and EdgeDB database
3. Set up for integration of Google OAuth2
4. Data collection for image prediction training
5. Training of YOLOv8n model
6. Producing digital representation of game board using bounding boxes provided from prediction image
7. Implementation of SGF conversion feature
8. Setting KataGo benchmark
9. Integrating KataGo analysis and displaying it on the front end
10. Moving back end hosting to Wentworth server for faster analysis
11. Integration of Gollama chat bot

Project Goals

The following goals were laid out at the beginning of the project and were met following the implementation:

1. User-friendly front end including clear display of information and straightforward navigation and image upload
2. Machine learning models that are effectively trained in game recognition and game history to provide the user with information necessary to analyze their technique
3. UI for AI game analysis/suggestions
4. Integrate KataGo on back end for fast game analysis with zero user-required configuration

The following were stretch goals made at the beginning of the project that were not met:

1. Storing uploads users tell us they had issues with
2. Write our own Go game analyzer

Issues that Arose During Implementation

As aforementioned, the primary issue we faced during implementation was the limitations of our initial self-hosted server. Not only did we see fairly consistent crashing of this server, but once we began working with KataGo, it became clear that we could not move forward with this approach, as we could only analyze 13 moves per minute. This would cause the analysis to be slow to the point of unusability for the user. At this point, our group decided to move the back end to a Wentworth-hosted server. This proved to be vital to the success of our project, as our analysis rate jumped up to 600 moves per minute, meaning that we were able to integrate KataGo as we planned. Furthermore, access to a better server meant that we were able to explore further additions to our project. This led us to implementing the Gollama chat bot, which allowed us to provide another resource to the user for enhancing their Go knowledge.

Lessons Learned During Implementation

The most important lesson we learned during this project was the importance of collaborative team sessions when implementing major features. Our initial approach to this project was to assign one person to oversee each feature. This created a bit of a disconnect within the group during the first few weeks of implementation, as with each team member focused on a different aspect of the project, it was more difficult to move through issues, as the other team members would not be as familiar with features on which they were not actively working. Realizing this, we adopted a new approach. We began to consistently have full-team sessions when implementing major features. This proved beneficial for several reasons. First, having multiple eyes on the implementation of a feature can speed up the process and minimize mistakes. Second, it led to a more collaborative team, as all members were up-to-date on all project activities. These sessions were the primary reason we were able to deliver on all of our primary goals, and, as such, this is the most important lesson we learned.

Conclusion

We successfully achieved our primary goals of implementing accurate object detection and providing useful game analysis. In addition, we were able to implement some further features, such as the chat bot and SGF conversion. The system provides Go players with resources to enhance their skills and enjoyment of the game.

Resources

The following is a link to a document containing the resources we utilized during the implementation of this project.

[NabGo Bibliography](https://docs.google.com/document/d/1ozwZ_yOXFPLypXROpRdRfhI3P_XInCsieswEH6V-7T8)