Cameron Dugan, Josh Gifford, Wilder Scott

Professor Salem Othman

Senior Project

3 August 2024

Design Document – NabGo

## Project 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.

## Major Software Components

The major software components of our application are listed below in their intended order of completion. Figures 1-3 shows sequence diagrams for intended interactions.

1. Training dataset of Go boards and labels
2. Convolutional neural network model for Go game board and stone detection from images
3. Integrate with KataGo AI player for fast game analysis
4. Integrate with Ollama to provide chat bot for assistance with game rules, strategy, etc.
5. Front end interface for image upload
6. Front end interface for AI game analysis/suggestions
7. Front end interface for interaction with chat bot
8. JavaScript functions that send HTTP requests to back end for image prediction and analysis
9. Back end to accept and process HTTP requests from front end and respond with results
10. Button on the UI to copy Smart Game Format (SGF) of current board state for use on external Go tools
11. Integration with Google OAuth2 to allow users to login to site
12. Database to track user logins

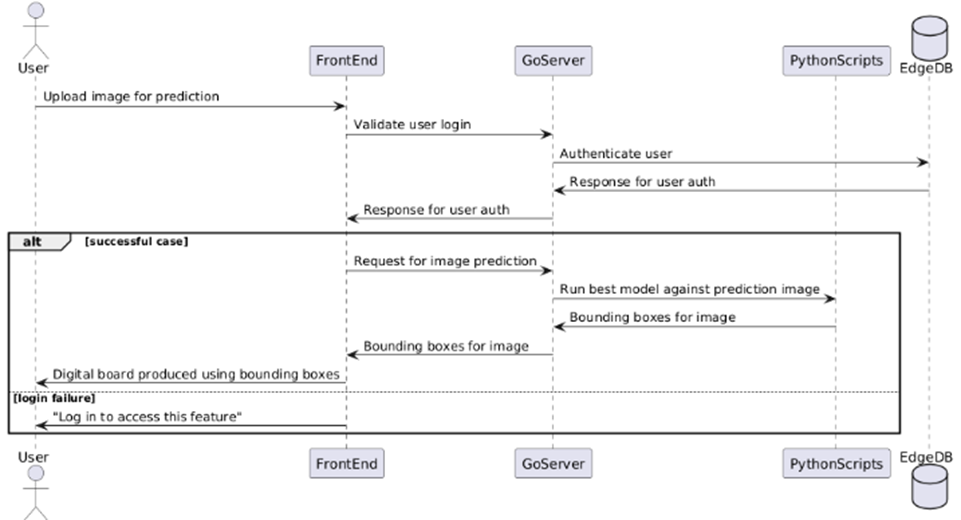


Figure 1: Sequence diagram for image upload and reproduction on digital board

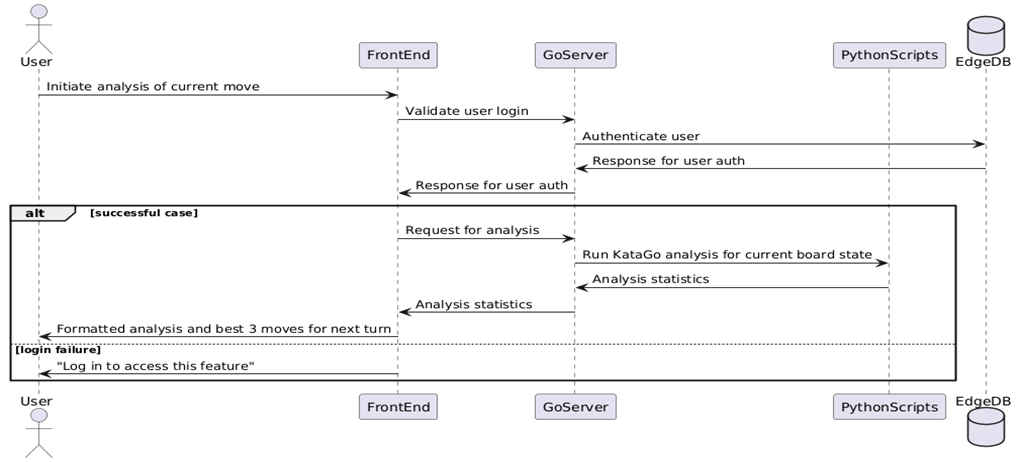


Figure 2: Sequence diagram for KataGo analysis

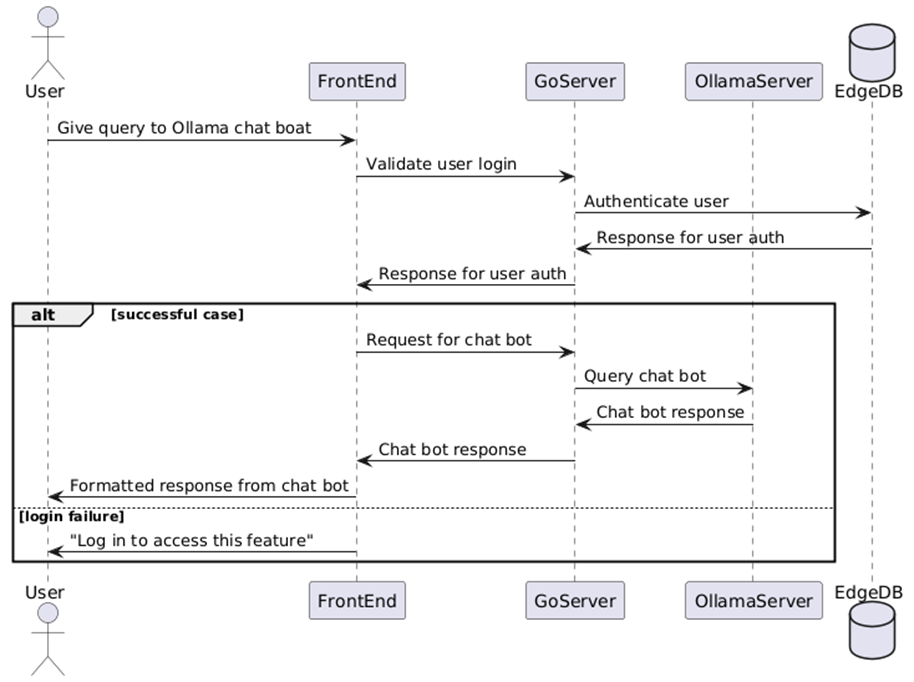


Figure 3: Sequence diagram for interaction with Gollama chat bot

## Software Tools

* **Languages**
  + Front End
    - HTML/CSS
    - JavaScript
  + Back End
    - Go
    - Python
* **Acquiring Datasets**
  + Roboflow Universe
* **Object Detection**
  + YOLOv8n from Ultralytics
* **Game Analysis**
  + KataGo AI Player
* **Chat Bot**
  + Meta Llama3 via Ollama
* **Data Storage**
  + EdgeDB
* **Development Environments**
  + NeoVim
  + Visual Studio Code
* **Version Control**
  + GitHub

## Existing Components

The following section designates the existing components that will be utilized during our implementation as well as their purpose.

1. KataGo AI Go Player

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

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

This section will briefly discuss the major design decisions that are essential to the delivery of our product. This will be separated into sections based on decisions that have been made versus decisions that are still being considered.

**Made**

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.

## Project Milestones

This section details the major milestones that will mark the progress of this implementation.

1. Project proposal and design decisions
2. Data cleaning
3. Implementation of front-end functionality
   1. Full game features such as
      1. Forward/backward history movement
      2. Removing captured pieces from board
      3. Ability to edit predicted game board for any inaccuracies
   2. Image upload UI
   3. Analysis panel to view game statistics
   4. UI to interact with chat bot
4. Training for aforementioned machine learning models
5. KataGo integration
6. Llama3 integration
7. System-wide testing
8. Final wrap-up and presentations

## Project Goals

The primary goal of this project is to create a hub for players to sharpen their skills in the game of Go. This goal will be addressed by the following aspects of our project.

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.

In addition, if this project were to be expanded into a commercial product, more areas would need to be addressed. We would need to consider a payment system which could be approached in several ways. We could implement a system that utilizes a back end for blocking access to specific features such as analysis or exporting beyond strategic usage limits, or we could utilize a pay-per-use model. Either way, this is a decision that would need to be further investigated in order to utilize our project in a commercial setting.

## Possible Concerns

This section outlines the potential concerns for the project. These are potential roadblocks in the delivery of our application.

1. Data quality
   1. If our quality is low or the data we have doesn’t hold enough predictive power, we could be looking at far fewer features
2. Model performance
   1. If every model we try doesn’t perform to our needs, we could be looking at far less implementation
3. Integration issues
   1. Components work on their own, but break when put together.
4. CDN issues
   1. If our CDN fails us, we might need time to look into other hosting options
5. Performance issues
   1. If our code isn’t fast, we could be looking at a long turnaround time on each iteration attempting to fix bugs