VegSecAI - Conceptual Design Document

Project Overview

VegSecAI is an AI-powered system designed to classify vegetables based on images captured by the user.

The system aims to make it easy for anyone to identify vegetables using a simple and secure interface. With integrated cybersecurity features, VegSecAI ensures that user data and interactions are safe.

The system supports a variety of functionalities, including real-time vegetable identification, historical image tracking, Face Identification and more. This document outlines the conceptual design and vision for the VegSecAI project, including architecture, module breakdown, use case examples, and GUI concepts. It serves as a guide for the creation and deployment of the VegSecAI system.

Table of contents

[**Project Architecture 2**](#_x4wxk4n9o40t)

[**Module table 3**](#_47cm25u63p80)

[**Use case examples 5**](#_4piqg6evdg3)

[Use case no. 1 {Frontend} 5](#_71pnlm11hl93)

[Use case no. 2 {Backend} 6](#_6v4hzvhqiy2v)

[Use case no. 3 {AI Training by admin} 7](#_th0yjfwlqez4)

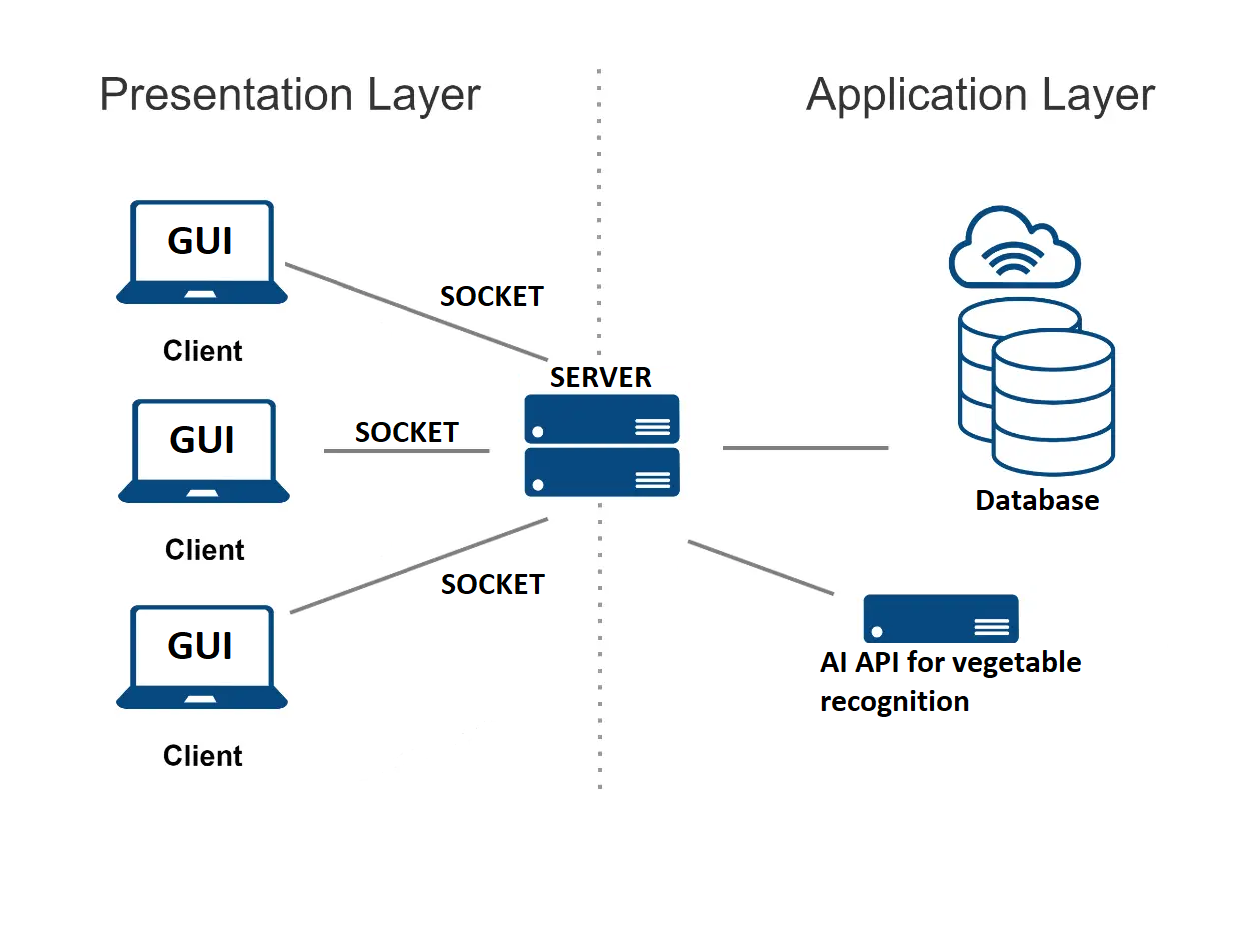
[**GUI Vision - Concept 8**](#_ya69iixxod5g)

[Screen 1 {Main Menu} 8](#_kuacv4pgatsn)

[Screen 2 {Response Received} 9](#_hggrqojmm5m0)

[Screen 3 {View history} 10](#_xodiw4xtgkv1)

# Project Architecture



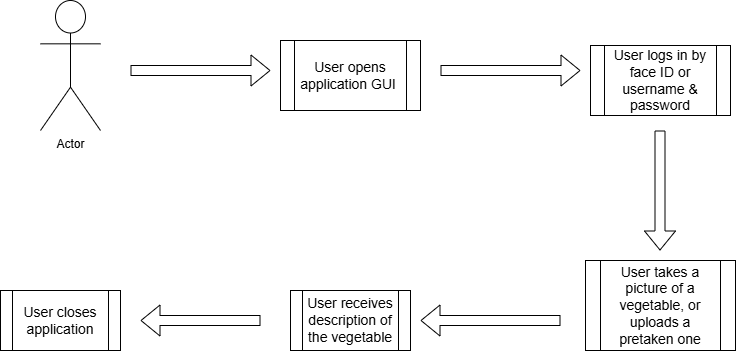
# Module table

|  |  |
| --- | --- |
| Module name | Module functionality and further description |
| ממשק המשתמש. מוצג לכל המשתמשים בפתיחת האפליקציה. (GUI) | הממשק יהיה רשום בשפת פייתון, ככל הנראה או TKINTER או FLASK. תחילה יוצגו למשתמש אפשרויות לכניסה למערכת: כניסה על ידי שם משתמש וסיסמא (אשר ערך hash שלהם יעבור בסוקט מאובטח, למאגר נתונים מאובטח), או כניסה על ידי זיהוי פנים (לשם כך נדרשת מצלמה, כמובן).  לאחר מכן, תציג המערכת למשתמש את האפליקציה עצמה: שם ישנו מקום עבור המשתמש להעלות תמונה, הסבר על מה האפליקציה עושה וכיצד להשתמש בה.  לאחר שימוש המשתמש באפליקציה, תוצג תשובת הבינה המלאכותית באופן אסתטי על המסך, ותהיינה אפשרות עבור המשתמש להעלות תמונה נוספת לבינה. |
| השרת (Server) | השרת הוא יהיה זה שיריץ את מודל הבינה המלאכותית (נכון לעכשיו). כאשר המשתמש יפתח את האפליקציה יוקם סוקט מאובטח אל מול השרת. השרת יהיה מסוגל לקבל חיבורים מכמה משתמשים בו זמנית.  בעת צילום תמונה, התמונה תגיע אל השרת, שם תקבל הבינה את התמונה ותעבד אותה. לאחר מכן תשלח בחזרה אל המשתמש את התשובה לגבי איזה ירק זוהה בתמונה. |
| מאגר הנתונים (Database) | מאגר הנתונים יהיה בעבור שמירת הפרצופים בעבור זיהוי הפנים, ובעבור שמירת ההאשים של הסיסמאות עם שמות המשתמשים המתאימים באופן מאובטח.  בנוסף, ייתכן ובמאגר הנתונים יהיו שמורים גם התמונות שהועלו בעבר והמשתמש שהעלה אותם, וכך יוצגו כל פעם מחדש למשתמש כאשר הוא מתחבר לאפליקציה (היסטוריית העלאות). |
| מודל הבינה המלאכותית (AI) | הבינה המלאכותית נכון לעכשיו רצה על השרת, אך דבר זה נתון לשינוי, כיוון והמחשבים הבית ספריים אינם חזקים. (GPU חלש אם בכלל יש להם).  דפ"א ב הוא שהבינה תרוץ על מכונה אחרת כלשהי, והשרת יתחבר אליו.  מנוע הבינה המלאכותית יהיה Image to Text, אשר מקבל תמונה ומחזיר תשובה המכילה את סוג הירק אשר נמצא בתמונה.  התכנון הוא לקחת API מוכן של בינה אשר מאומנת בנושא זה, ולאמן אותה עוד. |
| אבטחת הנתונים (Data security) | * כל הסיסמאות יהיו שמורות כערך Hash ולא Hardcoded לDatabase. * כלל הנתונים אשר עוברים במרחב הקיברנטי (בסוקטים) יהיו מוצפנים כדי להגן ממתקפות שונות. |

# 

# Use case examples

## Use case no. 1 {Frontend}

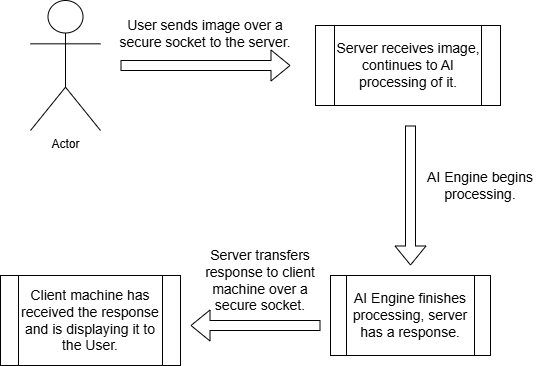


Above is the illustration of the usage of the application.

The user begins by opening the GUI. Then, admin is prompted to pick one of the two available methods to log in: Either FaceID or by Username & Password. Then, the user is prompted to upload/take (if a camera is connected) a picture.

Once a picture is uploaded, it is sent to the server over a secure socket. Afterwards, the client’s machine receives the response - a description of the vegetable in the picture. It is then displayed nicely for the User.

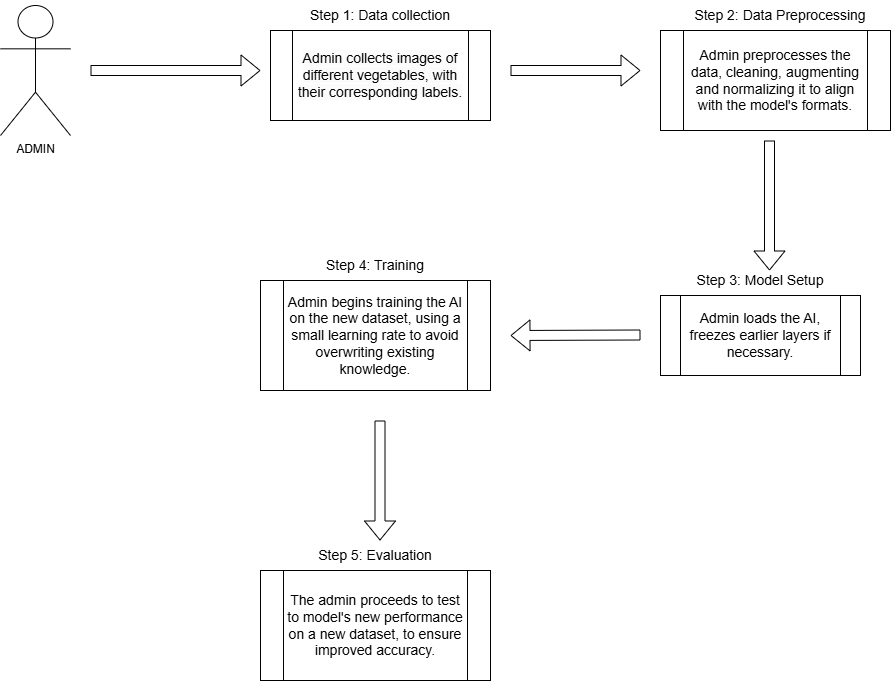
## Use case no. 2 {Backend}



Above is the illustration of the backend side of the application.

The user sends an image to the server over a secure socket, server receives it and begins processing it on the AI engine. Once done, the server receives a response from the AI which it then sends over the secure socket back to the client machine, where it is then displayed nicely for the user.

## Use case no. 3 {AI Training by admin}

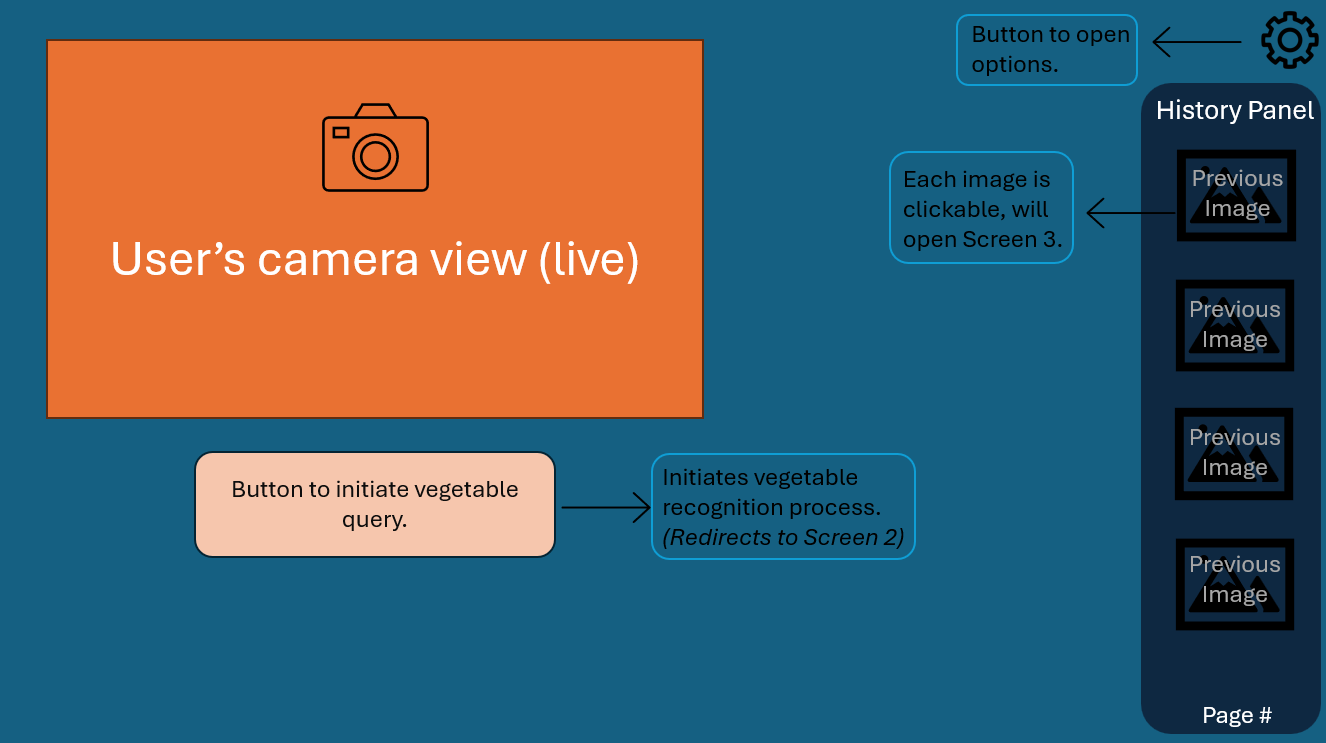
Above is the illustration of the training process of the AI engine. The Admin begins by collecting a set of images with their labels. Then, the admin proceeds to normalize them. 

Afterwards, the admin loads the AI model, freezing earlier layers if necessary to fine-tune higher layers.

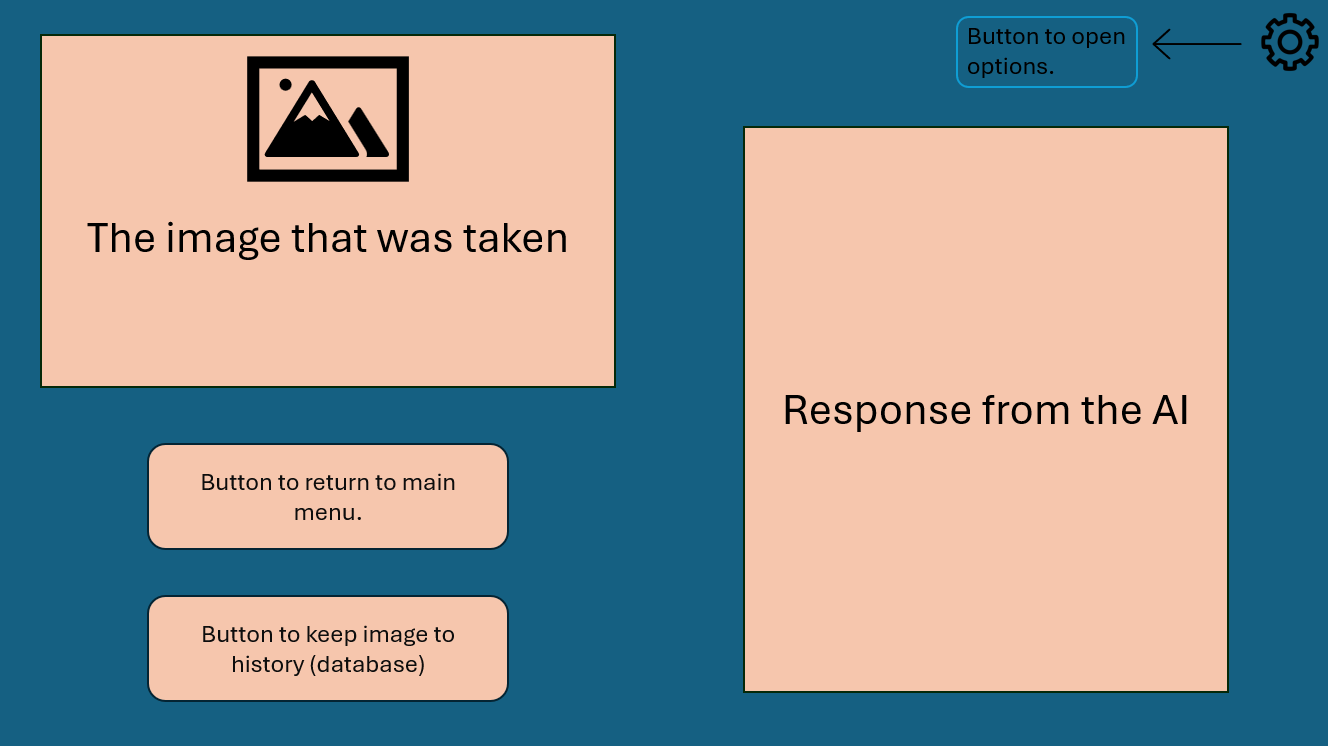
Then, the admin can begin training the AI. Finally, the admin conducts an evaluation to ensure improved accuracy.

# GUI Vision - Concept

## Screen 1 {Main Menu}



## Screen 2 {Response Received}



## Screen 3 {View history}

