



During the Summer of Capstone 1, the main core of the program had been completed, although not refined. This included the OpenCV camera system, the ruleset and main equations for determining posture through MediaPipe, a prototype GUI using PyQt5, the notification system using PyGame, and speech recognition. What remained for Capstone 2: machine learning datasets, analytics, recommendation system.

SpineWise exists as a lightweight, low/no-cost solution that tailors itself around the user's neutral good posture to help them maintain it, and correcting it through our recommendation system.

By a streak system and an analytics tab, SpineWise's users get real-time feedback and goals to strive towards in maintaining their best possible seated position. This encourages routine self-correction and ensures better spinal health.

Today's modern world is increasingly sedentary, and both students, office workers, and remote employees, among others, spend a large part of their day seated with imperfect posture that puts strain on their neck, back, and shoulders. Current solutions are restricted to expensive hardware that may be uncomfortable to most users, or software that does not curate itself to a user's neutral posture for adequate and tailored improvement.

[illegible]

- The Application detects user posture data in real time from a standard webcam using MediaPipe through OpenCV.
- The Application categorizes posture data into Good, Moderate, and Bad categories from smaller parameters.
- The Application alerts the user with an auditory alert and a visual in-app indicator for when posture is poor.
- The Application will listen to the user's voice in order to determine basic commands when the user is incapable of issuing keyboard + mouse commands.
- The Application will place data in a visual format for the user in a relevant tab, alongside assessing that data for exercise and product recommendations.

Front End (PyQt5): Displays the live camera feed, posture status, and recommendations.

- **Computer Vision (MediaPipe + OpenCV):** Tracks body and face landmarks in real time.
- **Posture Analysis (Custom Rules + ML Engine):** Uses calibrated metrics and an ML classifier to label posture as good, moderate, or bad.
- **Feedback System (Pygame + SpeechRecognition):** Provides audio alerts on posture changes and voice control.
- **Recommendations (SerpAPI):** Suggests exercises and ergonomic products based on posture issues recorded.

```

graph LR
    VCF[Live Camera Feed  
cr2.VideoCapture()] --> MT[Video Tracking  
MediaPipe + OpenCV]
    VC[Voice Commands  
pygame - SpeechRecognition] --> AS[Audio Feedback System  
Pygame + SpeechRecognition]
    MT -- "Video Feed & Landmarks" --> FEI[Front End Interface  
PyQt5 Display Dashboard  
spinevisa_gui.py - main PyQt5 dashboard (tabs, graphs, live view)]
    MT -- "Alerts & Status" --> FEI
    FEI -- "Posture Status" --> PAE[Posture Analysis Engine  
ML Classifier]
    FEI -- "Ergonomic Suggestions" --> PAE
    PAE -- "record_frame(), session, etc." --> ERE[External Recommendations  
Serp API]
    ERE -- "record_frame(), session, etc." --> PAE
    AS -- "Alerts & Status" --> MT
    PAE -- "ml.engine.py - ml predict from features(), record_frame(), session, etc." --> PAE
  
```


The material presented in this poster is based upon the work done by Javier Brasil, Oleh Krainyk, Juan Mieses, Jason Pena, and John Pena. Extended credits go to former members Emdya Permuy-Llovio, who conceptualized the project initially, and Jake Rodriguez. Special thanks go to our instructor, Masoud Sadjadi.

Test Case ID	SW_RB_ENV_UT_01(good lighting)
Purpose	Verify baseline performance of posture detection under ideal lighting
Preconditions	-Spinewise running -Camera calibrated in good posture -user seated, shoulders, head visible
Input	1. User Calibrates in neutral good posture 2. User intentionally slouches(bad posture) 3. User returns to upright posture
Expected Output	System transition from Good -> Bad -> Good appropriately(moderate = acceptable during slouching)

Test Case ID	SW_ML_ENV_UT_01(Camera further Away)
Purpose	Evaluate ML model robustness when the user is slightly far away from camera
Preconditions	-Spinewise running in ML mode -Camera moved back so the full upper body is smaller in frame -Calibration performed at this distance
Input	1. User in neutral good posture 2. User intentionally slouches(bad posture) 3. User returns to upright posture
Expected Output	-ML model continues to distinguish good vs moderate/bad despite smaller landmark scale -Some reduction in sensitivity is acceptable, but errors should be rare

SpineWise serves as a lightweight and free solution to self-monitoring one's posture health in order to maintain a routine of posture self-management.

This project, while in a deliverable state now, can be managed and updated for years to come, for improvements on its data assessment and analysis, a refinement on its recommendation and live coach system, as well as a fully fleshed out notification system and expanded datasets for more posture enhancement scenarios.