Adel Ozdem

adel.ozdem@laverne.edu

Department of Computer Science, University of La Verne

Comps

Prof. Juan Rodriguez

January 21, 2026

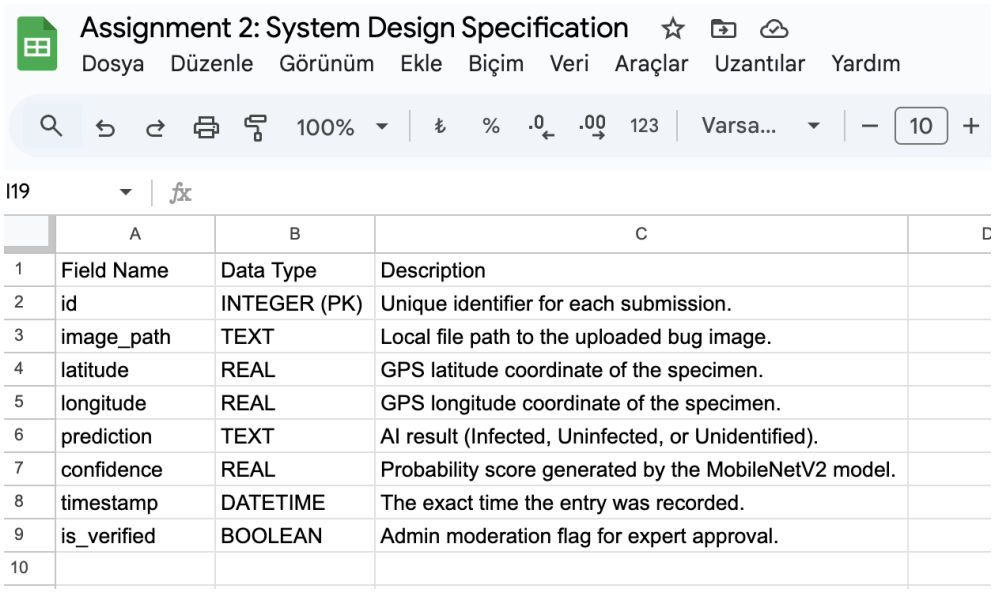
1. Project Overview & Progress

The **Chagas AI System** aims to provide an automated, mobile-ready solution for identifying *Triatoma dimidiata* and its infection status. Following the completion of the Database & API milestone on January 20, 2026, the project has transitioned from conceptual design to technical implementation. The system now focuses on bridging the gap between field photography and expert-level diagnostic accuracy.

2. Technical Architecture & Database Design

The system utilizes a **SQLite** database to manage metadata and geospatial data efficiently. The following table defines the schema for the core tracking table:

Database Schema: bug_records



	A	B	C	D
1	Field Name	Data Type	Description	
2	id	INTEGER (PK)	Unique identifier for each submission.	
3	image_path	TEXT	Local file path to the uploaded bug image.	
4	latitude	REAL	GPS latitude coordinate of the specimen.	
5	longitude	REAL	GPS longitude coordinate of the specimen.	
6	prediction	TEXT	AI result (Infected, Uninfected, or Unidentified).	
7	confidence	REAL	Probability score generated by the MobileNetV2 model.	
8	timestamp	DATETIME	The exact time the entry was recorded.	
9	is_verified	BOOLEAN	Admin moderation flag for expert approval.	
10				

3. API Endpoints (Flask Backend)

The **Backend API Gateway** acts as the central hub for the entire system. Below are the primary endpoints developed to handle communication between the UI and the AI engine:

- **POST /upload**: Receives image files and GPS metadata from the mobile-ready web UI.
- **GET /process/<id>**: Triggers the **Preprocessing Module** to normalize lighting and remove background noise.

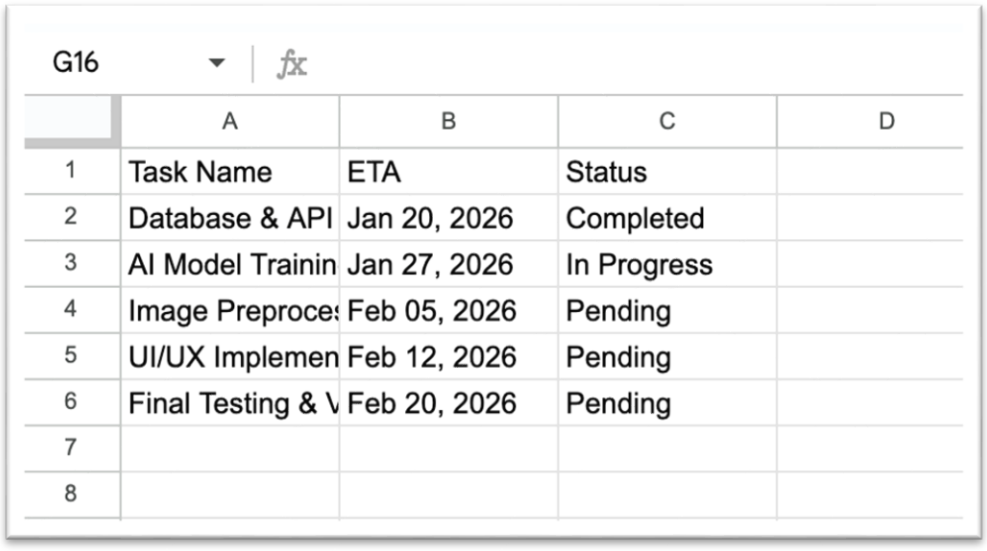
- **POST /classify**: Sends preprocessed images to the **MobileNetV2** model for infection detection.
- **GET /map-data**: Retrieves coordinates from the database to populate the real-time risk map.
- **PATCH /admin/verify**: Allows experts to update the is_verified status via the Admin Moderation Panel.

4. System Robustness & Fallback Logic

As identified in the Design Structure Matrix (DSM) analysis, the system's accuracy is highly dependent on the quality of input.

- **Data Augmentation:** To compensate for limited datasets, the **Data Augmentation Engine** generates variations in orientation and light to improve model training.
- **Fallback Logic:** In the event of an AI model failure, the system triggers a simulation mode to maintain UI stability.
- **Data Integrity:** If the database becomes unreachable, local caching is implemented to prevent data loss for health workers in remote areas.

5. Timeline & Next Steps



	A	B	C	D
1	Task Name	ETA	Status	
2	Database & API	Jan 20, 2026	Completed	
3	AI Model Trainin	Jan 27, 2026	In Progress	
4	Image Preproces	Feb 05, 2026	Pending	
5	UI/UX Implemen	Feb 12, 2026	Pending	
6	Final Testing & V	Feb 20, 2026	Pending	
7				
8				