CAREBOT

Caregiver Connect — Crowdsourcing Team

Sprint 1 Presentation | CS 495 Senior Capstone | University of Alabama

Cole Segura

Carlos Marquez

Sheala Miller

Jaylon Sanders

Jarl Evanson

GitHub: github.com/Distrubuted-Autonomy-Lab/carebot-deployable-crowdsourcing

Website: https://distrubuted-autonomy-lab.github.io/crodwsourcing-senior-info-website/

February 24, 2026

PROJECT FOCUS

What is Carebot and what are we building?

The Application

Django web app backed by PostgreSQL, PostGIS, and pgvector that helps Alabama residents find healthcare resources through a Google Gemini AI chatbot with semantic similarity search.

Stack: Django 5.0, PostgreSQL 17, PostGIS, pgvector, Sentence Transformers, Gemini, Docker Compose

Where We Started

- Docker, Django, PostGIS, and environment research
- Resource Submission: Rebuild/Adjustments
- Moderation System: Rebuild/Adjustments
- Docker Portability
- Security Improvements

Our Focus

- Enable community resource submissions
- Build moderation workflow
- Implement merge/deduplication system
- Improve development environment & security

SPRINT 1 GOALS

Prioritized objectives for Jan 20 – Feb 24, 2026

Fix Resource Submission Flow

Implement Merge Functionality

Improve Docker Environment

Initial Security Improvements

Research, Documentation, & Onboarding

MAJOR ACHIEVEMENTS

Development Environment

- Getting Docker running across all team machines was a major sprint focus
- Database initialization handles existing data smoothly
- Initial security improvements

Submission Flow

- Structured public form for resource recommendations
- Submissions flow into the moderation queue

Moderation

Approve as New: Creates a new resource in the system

Merge into Existing: Updates individual fields on a matched resource

Reject: Marks submission as rejected

Research

- Researched existing tools used in project
- Developed understanding of topics relevant to project scope

PROJECT & SPRINT BACKLOG

Sprint Backlog

Comprehensive end-to-end testing

Unit tests for submission & merge

HTTPS enforcement

Fuzzy matching improvements

Bulk moderation (multi-select)

Email/SMS notifications

Project Backlog

- Resource submission form + moderation queue
- Approve, merge, and reject actions
- Smart merge with deduplication
- Docker compatibility (Mac + Windows)
- Security configuration improvements
- Setup guides and documentation

NOT COMPLETED & LESSONS LEARNED

Carrying to Sprint 2/3

- Comprehensive end-to-end testing
- Security headers for production
- Unit tests for submission and merge flows
- Bulk moderation (approve/reject multiple)

Lessons Learned

- Docker requires cross-platform testing early — Mac and Windows behave differently
- Inherited codebases need architecture review before making changes
- LLM-enabled IDEs helped us navigate and contribute to a large codebase more easily
- Early documentation saves significant time for team onboarding
- Regular scrum meetings kept the team aligned through blockers

TEAM CONTRIBUTIONS

Cole Segura

Coordinated team direction, development environment, documentation

Carlos Marquez

Development environment setup, testing, bug reporting

Sheala Miller

Scrum coordination, security research, project website

Jaylon Sanders

Application research, feature testing, documentation

Jarl Evanson

Environment and build setup, feature research, testing configurations

DEMO

Submission → Moderation → Merge