Capstone Weekly Status Report 7

Date and Time: October 19, 2025

Student Name: Satwik Alla

Capstone Project: SBOM Manager - Software Bill of Materials Management System

Capstone Sponsor: Software Engineering Department **Ethical Software Lab** – In-house project

Capstone Advisor: Prof. Mohammad Samarah & Prof. Melissa Sahl

Status Details:

Work accomplished this Week

Backend Foundation & Core SBOM Features:

- Project Planning & Infrastructure: Created comprehensive 40+ page project plan with architecture, database schema, and 8-week timeline. Configured Supabase with PostgreSQL database (3 core tables), Row Level Security policies, authentication system, and storage bucket.
- Complete Authentication System: Built 5 FastAPI endpoints (register, login, refresh token, get current user, password reset) using JWT tokens, bcrypt password hashing, and Supabase Auth with email verification.
- File Upload & Storage: Implemented multi-format upload system (ZIP, APK, IPA, EXE, DEB, RPM) with 50MB size validation, SHA256 hashing for deduplication, and organized Supabase storage by user ID.
- SBOM Generation Pipeline: Integrated Syft CLI tool for automated SBOM generation with CycloneDX/SPDX support, background processing via FastAPI BackgroundTasks, 5-minute timeout handling, and automatic platform detection (Android, iOS, Windows, macOS, Linux).
- Component Storage & Management: Built database integration extracting/storing components from SBOM data, managing application-component relationships, detecting duplicates, and tracking processing status (processing/completed/failed).
- Application Management API: Implemented 12 functional endpoints including authentication (5 endpoints), file upload/status (2 endpoints), and application management (5 endpoints: list with pagination/filtering, get details, get components, export SBOM, delete).

- Testing & Documentation: Verified end-to-end workflow from registration → login → upload → SBOM generation → component storage using Swagger UI. All core features operational with proper error handling and API documentation.

Work Planned for Next Week

Search Functionality:

- Implement PostgreSQL full-text search with fuzzy matching for partial queries
- Build search API endpoint with multi-filter support (platform, OS, category, binary type)
- Add pagination, sorting options (relevance/date/name), and result ranking
- Create search service with query preprocessing and result highlighting
- Performance testing (target: <1 second response time)
- **Stretch Goal**: Begin frontend React + Vite setup if time permits

Problems, obstacles, needs, or questions that I need help with from client, advisor, or instructor

- Should we prioritize search accuracy or speed in next week implementation?
- Is PostgreSQL full-text search sufficient for MVP, or plan for dedicated search engine later?
- For frontend: use shadcn/ui component library or build custom components?

Overall Status

- Project Status Color: Green
- Project Status Color from Previous Week: Green