

Dispatching Dashboard Assignment – Notes

Goal: Build a dashboard for dispatching (proof-of-concept). Show what I can deliver in a short amount of time.

Due: Monday

What is Dispatching?

Current process of an MSP (Managed Service Provider).

Tickets go into a virtual queue when dispatch is needed.

Tickets are prioritized based on importance.

Tickets are also assigned to skill groups (line groups):

- Line 1 → easiest → 100 points
- Line 2 → medium → 200 points
- Line 3 → hardest → 300 points

Currently Weighted_Importance and Priority fields exist but have no impact on points.

Only SkillGroup matters right now.

Application Behavior

Queue Management: Tickets go into a virtual queue. When completed → removed from queue.

Completed ticket saved in ticket archive table (SQL DB).

Points System: Points assigned based on which skill group handled the ticket.

Example: Line 1 ticket = 100 points, Line 2 = 200 points, Line 3 = 300 points.

Track who closed the ticket.

Testing: Add a test button to simulate ticket closure. Marks ticket closed, assigns points, updates archive.

Technical Setup

Backend: Can use mock repository (no need for real SQL DB). Store tickets temporarily (mock data or SQL). Handle archive logic + points calculation.

Frontend: Not priority → can be basic. Requirements: show queue of tickets, login page, graph for points/ticket history.

Authentication: Microsoft OAuth (research docs). Microsoft Entra-ID (research docs). Not mandatory but button for Microsoft SSO is nice to have. Validate login token with Ekkie.ai account.

Cloud: Use Azure (license provided, no cost).

Languages: Python or Cobalt.

Deliverables

Working backend (mock repo, tickets, queue, archive).

Points system functional.

Test button for simulating ticket closures.

Optional: Simple frontend with queue view, login page, graph for points.

Extra Notes

Doesn't require full documentation, but good to write some.

Dashboard is for looks only → backend logic is the main focus.