YorkU Marketplace – Sprint 3 Schedule & Critical Path

EECS 3311 – Software Design

December 1, 2025

Introduction

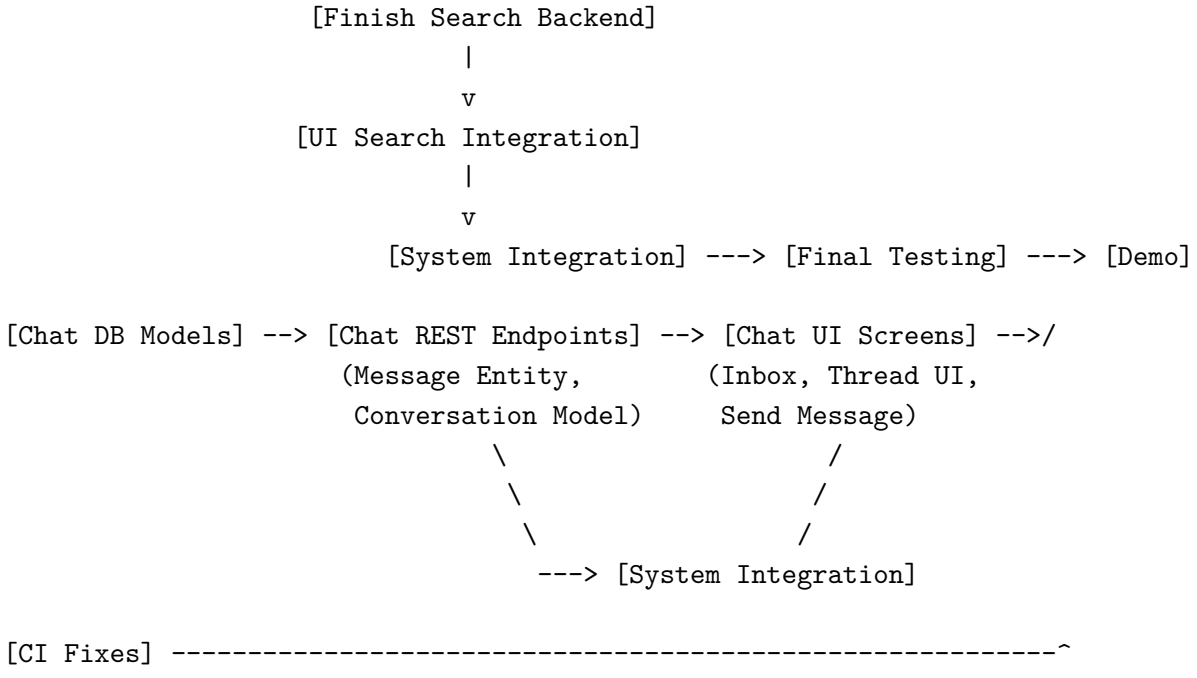
Sprint 3 was the final sprint and focused on:

- Completing the **chat (direct messaging) feature**,
- Finalizing the **search backend and UI integration**,
- **System integration**, testing, and demo preparation.

Network Diagram

The following ASCII diagram summarizes the key Sprint 3 tasks and their dependencies.

High-level view:



Task Descriptions

Below are the main Sprint 3 tasks, grounded in the actual structure of the YorkU Marketplace codebase.

1. Chat DB Models

- Creation of `Message` and `Conversation` entities.
- Mapped to DB tables using JPA (e.g., under `src/main/java/.../message/`).
- Responsible for storing messages between buyer and seller.

2. Chat REST Endpoints

- Service and controller methods to:
  - Create new conversations,
  - Send messages,
  - Fetch conversations and message history (e.g., by user pair).
- Uses Spring repositories for persistence.
- Exposed to the frontend via JSON endpoints.

3. Chat UI Screens

- Frontend pages/components (e.g., `inbox.html`, `chat.html`).
- Shows list of conversations (inbox) and individual threads.
- Uses JavaScript (or template logic) to:
  - Load messages,
  - Send new messages,
  - Poll periodically for new messages ("live-ish" updates).

4. Search Backend Finalization

- Completes the backend search logic that builds on Sprint 2 work.
- Includes:
  - Text-based search on titles/descriptions,
  - Category filter,
  - Min/max price filters,
  - Sorting by newest listing.

5. Search UI Integration

- Wires the search bar and filter fields to the backend endpoint.
- Updates the main listing page to:
  - Display filtered results,
  - Show thumbnails,
  - React to changes in filters.

6. System Integration

- Ensures that:
  - Auth, listings, search and chat all work end-to-end.
  - Ownership checks are respected (only owner can edit/delete listing).
  - DTOs and controllers use consistent models.

7. CI Fixes

- Updates GitHub Actions (e.g., `maven-publish.yml`).
- Ensures:
  - Project builds successfully,
  - Tests run in CI,
  - Branch protection rules are not blocked.

8. Final Testing & Demo Preparation

- Manual end-to-end tests:
  - Login → Create Listing → Search → Contact Seller → Chat → Delete Listing.
- Bug fixes and final polish for the recorded demo.

Critical Path

The **critical path** is defined as the longest dependent sequence of tasks whose delays would directly delay the completion of the sprint.

Identified Critical Path:

$$ChatDBModels \rightarrow ChatRESTEndpoints \rightarrow ChatUIScreens \rightarrow SystemIntegration \rightarrow FinalTesting \rightarrow Demo$$

Why This Is the Critical Path

- The chat feature introduced a new vertical slice:
  - Database layer,
  - Service/REST layer,
  - UI layer.
- Each of these layers depended on the previous one:
  - Chat UI could not be implemented before the REST endpoints existed.
  - REST endpoints could not be written before models/entities were finalized.
- System integration and final testing required a working chat feature, so they were blocked until the full chat stack was ready.

Parallel Tasks and Non-Critical Work

Some tasks could run in parallel or had slack time relative to the critical path:

- Search backend and search UI integration,
- CI pipeline fixes,
- Minor UI cleanups and bug fixes.

Delays in these tasks had more room to be absorbed without impacting the final demo date, as long as they completed before system integration and final testing.

Schedule Status

- All critical path tasks were completed before the end of Sprint 3.
- No user stories were rolled over to a Sprint 4 (project ended with Sprint 3).
- The sprint completed *on schedule*.

Lessons Learned

- Starting backend work early (models + repositories) made it easier to unblock UI.
- Parallelizing non-critical tasks (search, CI, minor UI) increased effective team velocity.
- System integration tends to reveal hidden coupling; scheduling it earlier would reduce end-of-sprint pressure.