

## Sprint 2 – Changes Document

**Project Name:** Polish: AI-Integrated Document Editor

**Team Name:** UNT's Best

**Team Members:** Walid Esmael, Matthew Norman, Arnav Verma, Mohamed Babiker

**Date:** November 8, 2025

---

### 1. Introduction and Overview

Polish, an AI-powered document editor that unifies writing, editing, and AI-assisted workflows into a single cloud-based platform, was improved by our team during Sprint 2. The main goals of this sprint were to implement dashboard features, authentication, and backend connection with Azure and Auth0.

We kept improving the project's design to make it more dependable, quick, and accessible. The sprint's objective was to normalize user flows from login to document management and link actual cloud services (Auth0, Azure OpenAI, Cosmos DB) in order to make the platform production-ready.

### Requirements Changes

- Added **R11–R14** to extend core functionality:
  - **R11:** Real-time collaboration using SignalR.
  - **R12:** Streaming AI responses (token-by-token) for a natural writing experience.
  - **R13:** Enhanced LaTeX export with BibTeX support.
  - **R14:** AI-generated templates for resumes, cover letters, and academic papers.
- Added **NF7–NF9** for accessibility, latency, and uptime metrics.
- Updated **authentication provider** from Azure AD B2C to **Auth0**, improving integration with React SDK and enabling social logins.
- Expanded the **system requirements** to include Auth0 credentials, live OpenAI API keys, and production database containers.

### Design Changes

- Updated **Architecture Diagram**: reflects Auth0 authentication and Cosmos DB with Azure Functions for real-time updates.

- Updated **ER Diagram**: added new entities like AllInteractions, Templates, AuditLogs and enhanced version tracking.
- Modified **Frontend Design**:  
Built dedicated Sign-In and Sign-Up pages.
  - Created the Dashboard page with real-time data placeholders.
  - Implemented the app directory structure
- Enhanced **Data Flow**: implemented WebSocket-based streaming (SignalR).  
Refined **Design Rationale**: documented trade-offs between cloud providers and design choices (Auth0 vs Azure AD B2C, SQL vs Cosmos DB).

## Testing (Sprint 2)

### Functional Coverage

- Authentication flows validated via test cases TC-AUTH-001 to 005.
- Editorial page CRUD validated via TC-EDIT-001 to 005.
- Inline AI chat scenarios covered in TC-CHAT-001 to 005.
- Azure API integration tested in TC-AZ-001 to 005.
- All critical cases passed in Staging by Nov 4.

### Automation

- **Unit tests**: Jest + React Testing Library for auth guards and editor reducers.
- **API tests**: Postman/Newman collections for CRUD and token validation.
- **E2E**: Playwright runs covering sign-up/in/out and editorial autosave flow.
- **Accessibility**: axe-core checks in CI pipeline.

### AI Log:

The team regularly utilized AI tools to speed up development, documentation, and debugging during Sprint 2 while upholding academic integrity and critically evaluating AI outcomes.

- **Documentation Support**: Early drafts of the Requirements, Design, and Changes papers were created using ChatGPT; all information was then personally edited and validated before submission.
- **Coding Help**: ChatGPT and GitHub Copilot were used to provide advice on Azure authentication setup, React component syntax, and error debugging during npm installation problems.
- **Testing Advice**: Using our Test Plan as a guide, AI produced sample test cases and assertions for Jest and Playwright.
- **Project Setup Troubleshooting**: Used AI to resolve Next.js environment issues, missing scripts, and npm errors efficiently.
- **Design Decisions**: Asked AI for recommendations on the folder structure and UI consistency within the app directory.

## Other

- **Trello board updated** to assign tasks per member:
  - *Walid Esmael*: Login/Sign-Up Page & Dashboard Page, Changes Doc
  - *Mathew Norman*: Login/Sign-Up & Test Plan Updates
  - *Mohamed Babiker*: Editor Page Frontend & Autosave / Version Control
  - *Arnav Verma*: Authentication, Design and Requirements Document Revisions
- To guarantee equality between functional and non-functional needs, we conducted a mid-sprint review.