Review Document: BPMN Teaching and Execution Platform Design

Prepared for Stakeholder Review

Overview

This document outlines a phased approach to building a BPMN (Business Process Model and Notation) platform designed for educational use, with a future path toward real-world workflow execution. The platform will be hosted initially on Firebase and later expanded to support live execution environments.

🏫 Phase 1: Teaching / Scholarly Mode

Firestore Structure (Per-User)

```
users/{uid}/
├── wipDiagrams (COLLECTION)
│   └── {diagramId}: {
│       title: string,
│       xml: string,
│       lastEdited: timestamp
│     }
│
├── submittedDiagrams (COLLECTION)
│   └── {diagramId}: {
│       title: string,
│       xml: string,
│       submittedAt: timestamp,
│       locked: true
│     }
│
├── finalDiagrams (COLLECTION)
│   └── {diagramId}: {
│       title: string,
│       xml: string,
│       finalizedAt: timestamp
```

```
|   }
```

Key Features

Feature	Description
WIP Editing	Users manage their work-in-progress diagrams
"Turn In"	WIP moves to `submittedDiagrams`, becomes locked
Finalization	Optional promotion of a diagram to `finalDiagrams`
Instructor Access	Global view of all user submissions
Versioning (TBD)	Optional diagram version history

👨‍🏫 Phase 2: Instructor Interface

Structure for Submissions

classes/{classId}/submissions/{uid}_{diagramId}

Instructor Capabilities

- Browse submissions by user and class
- Provide comments, scores, and feedback
- Compare WIP vs Final versions
- Export diagrams as XML for archival or offline use

🚀 Phase 3: Execution Mode (Real Workflows)

Goals

- Support real-time BPMN execution beyond academic use
- Replace Firebase backend with a performant local stack
- Add state evaluation, event tracking, and decision logic

Proposed Execution Stack

[Frontend UI]
↓
[Local API Gateway / Service Layer]
↓
[Redis: Token/State Management]
↓
[Worker / Executor]
→ Evaluates flows and logs results

Execution Engine Features

Feature	Purpose
Redis Queue	Track flow tokens and states
Diagram Forking	Launch new instance from a final diagram
Input Evaluation	Handle condition expressions / user input
Monitoring	Dashboard for active and past executions
Lightweight Stack	Custom executor avoids full BPM engine

🔄 Transition Path

Phase	Data Location	Description
Teaching (WIP)	Firestore: `wipDiagrams`	Students draft and edit flows
Turned In	Firestore: `submittedDiagrams`	Diagrams are locked for grading
Final Copy	Firestore: `finalDiagrams`	Ready for reuse or execution
Execution	Local Redis + Executor	Runs actual business workflows

---

📄 Next Steps

- Approve Firestore schema for MVP
- Build turn-in / finalize UI workflows
- Define instructor access controls and feedback schema
- Evaluate Redis executor architecture (Q3 2025 target)

Prepared by: Jeff Heisler
Date: June 21, 2025