

**Software Requirement Specifications**

Educational Process Mining for Workflow Optimization in eLearning Systems

**Submitted by**

Affan Abdulwahid (F22BINFT1M01222)

**Submitted to**

# Dr. Mustafa Hameed

**Department of Information Technology**

**Faculty of Computing**

**The Islamia University of Bahawalpur**

## MEETING DETAILS

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr No** | **Details** | **Date** | **Supervisor Signature** |
| **1** | Topic Selection and Approval | 01/11/2025 |  |
| **2** | Dataset Selection (EPM Dataset) | 04/12/2025 |  |
| **3** | SRS Finalization & Review | 05/12/2025 |  |
| **4** |  |  |  |
| **5** |  |  |  |

## SUMMARY

This Software Requirements Specification document defines the requirements for an **Educational Process Mining (EPM)** system designed to optimize student learning workflows in digital electronics courses. The document establishes functional and non-functional requirements for a desktop application that processes historical interaction logs from the Deeds Simulator, used by 115 first-year engineering students across 6 laboratory sessions at the University of Genoa.

The system automates the discovery of learning patterns through process mining algorithms, generating visual workflow maps (Directly-Follows Graphs, Heuristics Nets) and statistical reports correlating student behaviors with academic performance. Key requirements include:

* Batch ingestion of 600+ CSV event logs with attendance tracking.
* Real-time detection of disengagement patterns (idle time analysis, off-task behavior quantification).
* Grade correlation analysis with configurable performance thresholds.
* Interactive dashboard visualization supporting instructor-driven workflow optimization.

The document specifies **14 functional requirements** covering data preprocessing, process discovery, anomaly detection, and comparative analysis, alongside **7 non-functional requirements** ensuring performance (120-second ingestion time), privacy (local-only data storage), and reproducibility (parameter logging). Three detailed use cases with alternative flows map system behavior for dataset ingestion, workflow visualization, and cohort comparison scenarios.

## 1. INTRODUCTION

The project focuses on **Learning Analytics and Educational Data Mining**, specifically addressing the lack of visibility into student study patterns in digital electronics courses. While current Learning Management Systems (LMS) record grades and completion status, they do not explain the *process* students use to arrive at those outcomes—whether through systematic study, trial-and-error, or disengagement.

### 1.1 Purpose

The purpose of this software is to automate the discovery of student learning models from raw event logs. It aims to:

* Visualize actual "learning paths" taken by students using process discovery algorithms.
* Identify inefficiencies and disengagement markers through bottleneck analysis and behavioral pattern detection.
* Correlate specific process patterns with exam results using the dataset's detailed grade mapping to session concepts.
* Provide actionable recommendations for course redesign and personalized student interventions.

### 1.2 Scope

The system is limited to analyzing historical log data in CSV format.

**IN SCOPE:**

* Process data from the "Processes" directory containing 6 session folders.
* Clean and aggregate data from 115 students, handling variable file availability and attendance gaps.
* Provide an interactive dashboard for instructors to explore process maps, compare student cohorts, and generate reports.
* Map exam questions to specific exercises using provided metadata files.
* Generate disengagement reports flagging at-risk students.

**OUT OF SCOPE:**

* Real-time monitoring of students during active lab sessions.
* Modifying the original Deeds Simulator source code or LMS integration.
* Predictive modeling for future student performance (limited to descriptive analytics).
* Automated grading or assessment generation.

### 1.3 Product Perspective

This system operates as a standalone analytical tool that sits outside the live educational infrastructure. It takes two primary inputs:

1. **CSV Event Logs** exported from the Deeds Simulator.
2. **Excel Grade Sheets** containing exam and assignment scores.

The system utilizes the XES (eXtensible Event Stream) standard internally for log representation, making it compatible with other process mining frameworks if needed for future research.

**System Context:**

[Deeds Simulator] → CSV Logs → [This System] → Process Maps/Reports → [Instructor]

[Excel Grades] ───────┘

### 1.4 User Characteristics

**Primary User - Instructor (Administrator Role):**

* Faculty member teaching Digital Electronics or similar laboratory-based courses.
* Familiar with educational concepts but not necessarily technical process mining terminology.
* Requires actionable insights (e.g., "Student X is stuck on Boolean Algebra") rather than raw statistical tables.
* Expected to use the system 2-3 times per semester for course improvement analysis.

**Secondary User - Data Analyst/Researcher:**

* Educational researcher studying learning patterns.
* Comfortable with technical terminology and statistical methods.
* May want to export processed event logs for advanced clustering or machine learning.
* Expected to use API or export features for custom analysis.

### 1.5 Similar Apps and Systems / Literature Review

**Existing Tools:**

1. **ProM (Process Mining Framework):**
   * *Strengths:* Powerful academic tool with 300+ plugins, supports advanced conformance checking.
   * *Limitations:* Steep learning curve; lacks session-based longitudinal analysis; no built-in grade correlation features.
2. **Disco (Fluxicon):**
   * *Strengths:* Commercial tool with intuitive UI, excellent for business process mining.
   * *Limitations:* Expensive licensing; does not support custom "Grade Correlation" features needed for educational contexts.
3. **Celonis Academic Alliance:**
   * *Strengths:* Enterprise-grade process mining.
   * *Limitations:* Cloud infrastructure requires data uploads (privacy concerns); no educational-specific features.

Gap Identified:

No existing tool combines session-based longitudinal tracking, automated grade correlation with exercise-level granularity, and disengagement detection specifically for eLearning environments. This system fills that gap.

### 1.6 Proposed Technologies

|  |  |  |
| --- | --- | --- |
| **Component** | **Technology** | **Justification** |
| **Programming Language** | Python 3.10+ | Rich ecosystem for data science; PM4Py library native support. |
| **Process Mining** | PM4Py 2.7+ | Open-source, actively maintained, supports Heuristics/Inductive miners. |
| **Data Processing** | Pandas 2.0+ | Industry standard for CSV manipulation; handles 100MB+ datasets efficiently. |
| **Visualization** | Graphviz, Matplotlib | Graphviz for Petri nets/DFGs; Matplotlib for statistical correlations. |
| **Frontend Framework** | Streamlit 1.28+ | Rapid dashboard development; Python-native; supports file uploads. |
| **Database** | SQLite 3 | Lightweight, serverless; sufficient for storing aggregated logs/configs. |
| **Version Control** | Git/GitHub | Required by university; facilitates SDD tracking. |

### 1.7 Assumptions

1. **Dataset Integrity:** The dataset is static and provided offline.
2. **ID Consistency:** Student IDs in CSV filenames correspond exactly to the student\_Id column in Excel grade sheets.
3. **File Structure:** The directory structure follows the University of Genoa EPM standard (Processes/SessionX/).
4. **Hardware:** The system will run on standard academic computing resources (min 8GB RAM).

## 2. REQUIREMENTS

### 2.1 Functional Requirements

#### Module 1: Data Ingestion & Preprocessing

**1. Batch Directory Traversal and File Discovery (FR-01)**

* **Purpose:** To automatically locate and inventory all student interaction log files across multiple session folders.
* **Input:** Root Directory Path (Processes/ folder).
* **Output:** File Inventory List and Structure Validation Report.

**2. Attendance Tracking and Missing Data Logging (FR-02)**

* **Purpose:** To identify students absent from specific sessions by cross-referencing discovered files against the complete roster.
* **Output:** attendance\_report.csv (Binary attendance matrix) and Missing Files Log.

**3. Timestamp Parsing and Temporal Validation (FR-03)**

* **Purpose:** To convert European-format timestamps into standardized ISO 8601 format and validate chronological integrity.
* **Processing:**
  + Parse dd.MM.yyyy HH:mm:ss.
  + Convert to ISO 8601.
  + Validate end\_time >= start\_time.
* **Output:** Standardized start\_time\_iso, end\_time\_iso, and duration\_ms.

**4. Exercise Field Normalization (FR-04)**

* **Purpose:** To standardize inconsistent exercise labels arising from incomplete automatic detection.
* **Rules:** Replace generic "Es" or blank fields with "Unassigned\_Exercise\_Generic". Identify cross-session anomalies (e.g., Session 3 exercise appearing in Session 1 folder).

**5. Unique Case Identifier Generation (FR-05)**

* **Purpose:** To create globally unique identifiers for each trace (case) required by PM4Py.
* **Format:** S{session}\_U{student\_Id:03d}\_Ex{exercise} (e.g., S1\_U001\_ExEs\_1\_1).

**6. Sparse Feature Detection and Exclusion (FR-06)**

* **Purpose:** To exclude interaction metrics with negligible variance (e.g., always 0) to reduce noise.
* **Threshold:** Variance < 0.01.

#### Module 2: Process Discovery & Analysis

**7. Heuristics Miner Algorithm Implementation (FR-07)**

* **Purpose:** To discover process models from event logs using the Heuristics Miner algorithm.
* **Parameters:**
  + dependency\_threshold (default: 0.5)
  + and\_threshold (default: 0.1)
* Calculation:  
    
  $$dependency = \frac{freq(A \to B) - freq(B \to A)}{freq(A \to B) + freq(B \to A) + 1}$$
* **Output:** PM4Py HeuristicsNet object and Visual Directly-Follows Graph (DFG).

**8. Exercise-to-Exam Question Mapping (FR-08)**

* **Purpose:** To enable correlation between specific exam performance and interaction patterns.
* **Processing:** Map Es\_1\_1 (log activity) to ES 1.1 (2 points) (Excel column).

**9. Exercise Difficulty Filtering (FR-09)**

* **Purpose:** To isolate analysis to specific exercises (Single, Range, or All in Session).
* **Output:** Filtered Event Log and Aggregate Statistics Table.

**10. Disengagement Report Generation (FR-10)**

* **Purpose:** To identify at-risk students exhibiting disengagement patterns.
* **Logic:** Flag student if:
  + Off-task time >= 20%
  + Idle sequences >= 1 (where idle > 5 mins)
  + Attempt rate < 50%
* **Output:** disengagement\_report.csv with recommended actions.

**11. Cross-Session Anomaly Detection (FR-11)**

* **Purpose:** To identify students accessing exercises outside their designated session (Working Ahead or Reviewing).

**12. Process Map Visualization Rendering (FR-12)**

* **Purpose:** To provide rich visual feedback on workflow nodes and edges.
* **Visual Logic:**
  + **Node Size:**  
    $$20 + (10 \times \log(frequency))$$
  + **Node Color:** **Red** if avg\_idle\_time > 5 mins; **Green** if documentation phase.
  + **Edge Thickness:** Proportional to transition count.

**13. Data Export and Report Generation (FR-13)**

* **Purpose:** Export analysis results for external use.
* **Formats:** PNG/SVG (Process Maps), CSV (Reports).

**14. Performance Cohort Splitting and Comparison (FR-14)**

* **Purpose:** To reveal behavioral differences between high and low-performing students.
* **Method:**
  1. Split students based on grade threshold (Median, Custom, or Pass/Fail).
  2. Generate two Heuristics Nets.
  3. Compute differential metrics (e.g., Difference in avg idle time).
* **Output:** Split-screen dashboard and Comparative Statistics Table.

### 2.2 Non-Functional Requirements

* **NFR-01: Performance:** Ingest full dataset (~600 files) in < 120 seconds. Generate Heuristics Net in < 10 seconds.
* **NFR-02: Reliability:** Gracefully handle missing values and malformed CSVs; log all errors to errors.txt.
* **NFR-03: Usability:** Core workflow (Ingest → Analyze) achievable in 3 clicks. Use plain English labels (e.g., "Workflow Map" instead of "Petri Net").
* **NFR-04: Privacy & Security:** Anonymize outputs (Student IDs only); strictly local storage (no cloud uploads).
* **NFR-05: Reproducibility:** Log all analysis parameters (thresholds, filters) to a JSON file for every run.
* **NFR-06: Configurability:** All thresholds (idle time, off-task %) must be user-configurable via UI.
* **NFR-07: Maintainability:** Codebase must follow PEP 8 standards; modular architecture.

## 3. USE CASES AND FLOW OF PROCESSES

### Use Case 1: Batch Ingest Dataset (UC-01)

* **Actor:** Instructor (Administrator)
* **Description:** Import raw logs, validate integrity, normalize fields, and generate unified event log.
* **Basic Flow:**
  1. User selects "Dataset Root Directory" (Processes/ folder).
  2. System validates directory structure (6 session folders).
  3. User clicks "Start Processing".
  4. System iterates through all files, parsing timestamps and normalizing exercises.
  5. System generates "Attendance Report" and saves data to SQLite DB.
  6. System displays summary: "✓ 124,567 records loaded."

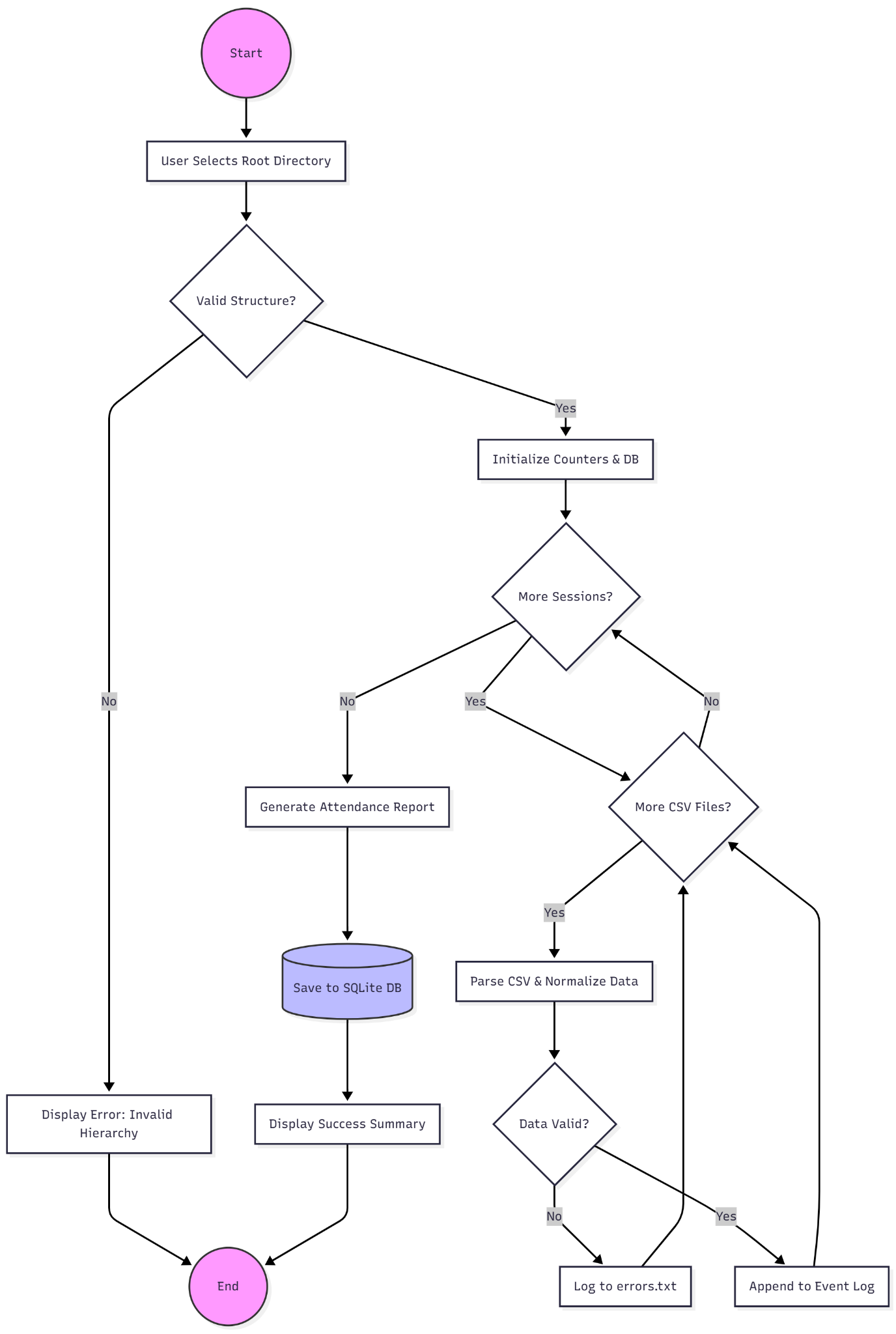
### Use Case 2: Analyze Learning Workflow (UC-02)

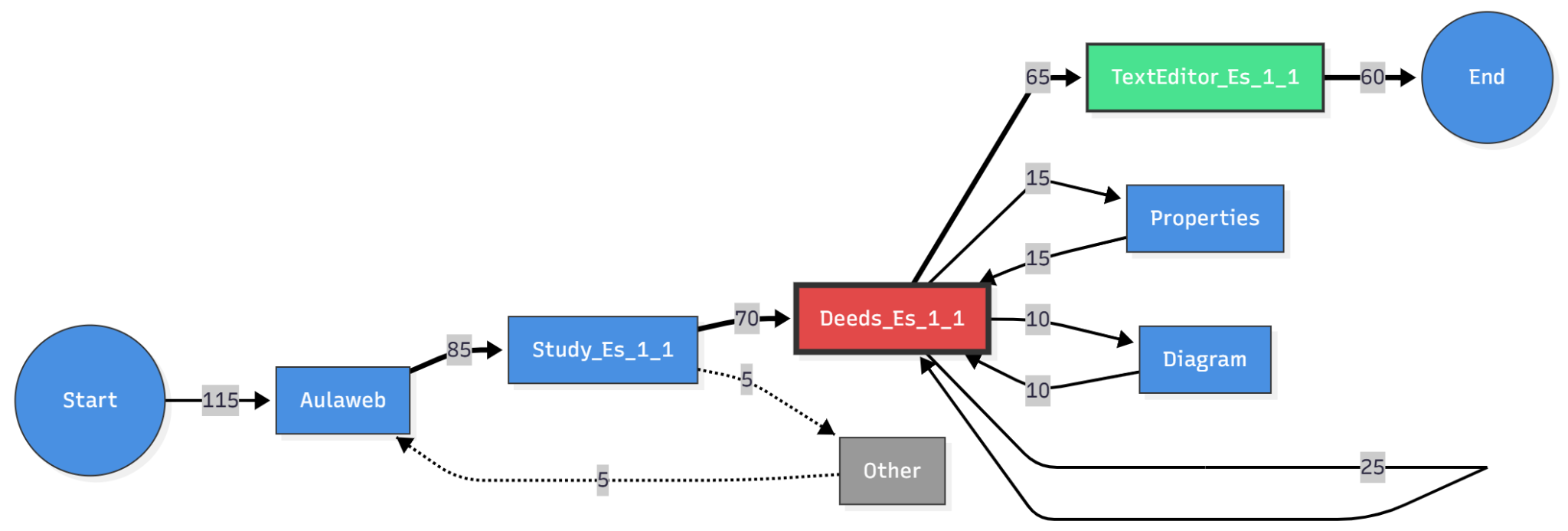
* **Actor:** Instructor
* **Description:** Generate a visual process map for a specific exercise to identify bottlenecks.
* **Basic Flow:**
  1. User selects Session (e.g., Session 1) and Exercise (e.g., Es\_1\_1).
  2. User adjusts Algorithm Parameters (Dependency Threshold).
  3. User clicks "Generate Process Map".
  4. System renders Directly-Follows Graph. Nodes with high idle time (>5 min) are highlighted in **Red**.
  5. User interacts with graph (zoom, click-to-highlight).

### Use Case 3: Compare High vs. Low Performers (UC-03)

* **Actor:** Instructor
* **Description:** Split students into cohorts based on grades and compare their workflows side-by-side.
* **Basic Flow:**
  1. User uploads final\_grades.xlsx.
  2. User selects threshold (e.g., Median Split).
  3. System splits students into High and Low cohorts.
  4. System generates two process maps side-by-side.
  5. System calculates statistics table (e.g., "Low performers idle 133% more").
  6. System provides automated recommendations.

## 4. DIAGRAMS





## APPENDIX A: DATASET COLUMN SPECIFICATIONS

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Unit** | **Description** |
| session | Integer | - | Lab session number (1-6) |
| student\_Id | Integer | - | Anonymous student identifier (1-115) |
| exercise | String | - | Exercise code or blank |
| activity | String | - | Active window/tool name |
| start\_time | DateTime | - | Activity start timestamp |
| end\_time | DateTime | - | Activity end timestamp |
| idle\_time | Integer | ms | Time user inactive during activity |
| mouse\_wheel | Integer | units | Amount of mouse wheel scrolling |
| keystroke | Integer | count | Number of keyboard keys pressed |

## APPENDIX B: ACTIVITY TYPE TAXONOMY

|  |  |  |
| --- | --- | --- |
| **Activity Label** | **Context** | **Interpretation** |
| **Aulaweb** | LMS | Reading course homepage, downloading materials |
| **Study\_Es\_X\_Y** | Exercise Content | Reading instructions for Exercise X.Y |
| **Deeds\_Es\_X\_Y** | Simulator | Active circuit design/simulation |
| **TextEditor** | Documentation | Writing solution report |
| **Diagram** | Timing | Viewing simulation waveforms |
| **Other** | Off-task | Irrelevant website or app usage |