Attendance Planning & Decision Support System

Version: 1.0

Date: January 2026

Team: TE Data Science, Batch 2023-27

Project Type: 6th Semester Mini-Project

EXECUTIVE SUMMARY 1.1 Problem Statement Students face attendance-related academic detention due to lack of forward-looking planning tools. Current systems only show historical percentages, leaving students unable to answer critical questions:

"Can I skip today's class safely?" "When does recovery become mathematically impossible?" "Which subject needs urgent attention?" This leads to:

Surprise detentions (students thought they were safe) Impossible recovery attempts (trying when it's too late) Poor resource allocation (focusing on wrong subjects) 1.2 Solution Overview A decision support system (not a tracking system) that transforms attendance from a backward-looking report into a forward-looking planning resource. The system calculates attendance buffers, recovery paths, and critical deadlines using:

Academic calendar (holidays, teaching weeks) Subject timetables (lectures, labs per week) Current attendance state (from monthly faculty reports or manual entry) 1.3 Success Criteria Students can make skip decisions in ≤30 seconds Point of No Return calculated accurately for all subjects System works with monthly updates (not daily tracking) 90% of calculations explainable on whiteboard (no black boxes) 1.4 Out of Scope (Critical Constraints) ❌ Not a biometric/RFID replacement ❌ No automated attendance marking ❌ No integration with institute ERP (for MVP) ❌ No ML/AI predictions ❌ No social/collaborative features 2. USER PERSONAS Persona 1: Crisis Responder (Primary Target - 60% of users) Profile: Rajesh, TE DS student, received defaulter notice in February

Current attendance: 68% in ML, 72% in DAV Motivation: High (detention threat) Tech savvy: Medium Update frequency: Once when in crisis, then weekly Use Case: "I got a defaulter notice. Is it too late to recover? If not, what exactly do I need to do?"

Key Features Needed:

Upload faculty defaulter list → auto-populate data Recovery feasibility check Week-by-week recovery roadmap Persona 2: Preventive Planner (Secondary - 30% of users) Profile: Priya, TE DS student, currently at 80-85% across subjects

Wants to skip strategically for placements/internships Motivation: Medium (proactive planning) Update frequency: Weekly or before important skip decisions Use Case: "I have a placement drive next week. Which 3 classes can I safely skip?"

Key Features Needed:

"What if I skip?" simulation 7-day lookahead with risk labels Subject-wise buffer visibility Persona 3: Last-Minute Checker (Tertiary - 10% of users) Profile: Ankit, TE DS student, checks app morning of each class

Currently safe but wants daily confirmation Motivation: Low (convenience) Update frequency: Daily (but inconsistent) Use Case: "Can I skip today's 9:30 AM ML class?"

Key Features Needed:

Today's schedule with safe/risky labels Quick impact preview Fast decision (no multi-step process) 3. FUNCTIONAL REQUIREMENTS 3.1 Core Features (Must-Have for MVP) Feature 1: Traffic Light Dashboard User Story: As a student, I want to see which subjects need urgent attention at a glance.

Requirements:

Display all subjects with color-coded status: 🔴 Red: Deficit (currently <75%) OR buffer ≤1 🟡 Yellow: Caution (buffer 2-4) OR approaching PNR 🟢 Green: Safe (buffer ≥5) Show current percentage per subject Show buffer/deficit count Update in real-time as data changes Acceptance Criteria:

All subjects visible without scrolling on mobile (compact cards) Tap subject card → detailed view Color scheme follows standard traffic light convention Works with stale data (shows confidence level) Feature 2: Skip Impact Simulator User Story: As a student, I want to see consequences before skipping a class.

Requirements:

For any upcoming class, show: Current percentage Percentage after skipping Buffer change Risk level change Allow "what if" testing without committing Batch simulation (select multiple classes) Acceptance Criteria:

Calculation completes in <200ms Shows both current and projected state side-by-side Warns if action causes status change (Green→Yellow) Blocks simulation if would drop below 75% Calculation Logic:

New % = Current_Attended / (Current_Conducted + 1) × 100 New Buffer = Total_Classes × 0.25 - (Total_Classes - New_Attended - New_Skipped) Feature 3: Point of No Return (PNR) Calculator User Story: As a student, I want to know the exact date when recovery becomes impossible.

Requirements:

For each subject, calculate date when: (Required_Attendances - Current_Attended) > Remaining_Classes Display as: Exact date if within semester "Already passed" if recovery impossible "Safe - no PNR this semester" if buffer sufficient Show countdown (e.g., "12 days until PNR") Acceptance Criteria:

Accounts for holidays and non-teaching days Updates daily automatically Shows warning at 2 weeks before PNR Critical alert at 1 week before PNR Calculation Logic:

python min_required = total_classes × 0.75 deficit = max(0, min_required - current_attended) if deficit > remaining_classes: PNR = "ALREADY PASSED" else: weeks_until_pnr = (remaining_classes - deficit) / classes_per_week PNR_date = current_date + timedelta(weeks=weeks_until_pnr)

Feature 4: Recovery Path Generator
User Story: As a student in deficit, I want a concrete plan to reach 75%.

Requirements:
- For subjects <75%, generate:
  - Total additional attendances needed
  - Week-by-week breakdown
  - Maximum achievable % (if recovery impossible)
- Show as timeline/roadmap
- Mark "achievable" or "impossible" clearly

Acceptance Criteria:
- Shows minimum required attendance streak
- Allows 1-2 buffer misses in plan (realistic)
- Displays "Plan difficulty": Easy/Medium/Hard/Impossible
- Updates as student attends classes

Output Format:

ML Recovery Plan (Currently 68%) Goal: 75% by 15 May 2026

Required: Attend 22 of next 28 classes Strategy:

- Week 3-6: Attend ALL classes (4/week) ← Critical
- Week 7-10: Can miss max 1 per week
- Week 11+: Can miss max 1 per week

Result: 75.8% at semester end ✓ ACHIEVABLE

3.2 Data Input Features

Feature 5: Multi-Modal Data Entry
User Story: As a student, I want flexible ways to input attendance based on what data I have.

Requirements:

Option A: Upload Faculty Document (For Defaulters)
- Accept PDF/Excel of defaulter list
- Auto-parse subject names, conducted, attended
- Show preview before confirming
- Handle common formats from different faculties

Option B: Manual Entry (For Non-Defaulters)
- Guided subject-by-subject wizard
- Real-time validation:
  - Attended ≤ Conducted
  - Conducted ≈ Expected (based on weeks elapsed)
- Save partial progress

Option C: Quick Batch Update
- "This week, across all subjects, I missed X classes"
- Distribute proportionally or per-subject
- Fast entry for weekly check-ins

Option D: Fresh Start Mode
- Start tracking from today
- Assume 100% attendance so far (or user estimate)
- Explicitly mark as "estimated baseline"

Acceptance Criteria:
- At least 2 input methods work in MVP
- Data validation catches impossible values
- Clear error messages for validation failures
- All paths lead to same data structure

3.3 Secondary Features (Nice-to-Have)

Feature 6: 7-Day Planner
- Calendar view of upcoming week
- Each class labeled Safe/Risky/Critical
- Multi-select for batch skip planning

Feature 7: Monthly Re-sync Prompt
- Notification when ~30 days since last update
- "Update your data for accurate planning"
- Quick re-entry flow

Feature 8: Attendance History Log
- View past updates (audit trail)
- Manual correction of errors
- Export to CSV

---

4. NON-FUNCTIONAL REQUIREMENTS

4.1 Performance
- Dashboard load: <500ms
- Calculations (PNR, buffer): <200ms
- Works offline (all calculations client-side)
- Mobile-first (80% of usage expected on phones)

4.2 Data Requirements
- Data stored locally (localStorage/IndexedDB)
- No backend database required for MVP
- Export capability (JSON/CSV)
- Import previous semester data

4.3 Usability
- Zero learning curve (intuitive within 2 minutes)
- Max 3 taps to reach any decision
- No jargon (avoid "conducted/sanctioned")
- Works in portrait mobile mode

4.4 Accuracy
- All calculations deterministic and verifiable
- Show formula on demand ("How is this calculated?")
- Confidence levels when data is stale:
  - HIGH: Updated within 7 days
  - MEDIUM: 8-14 days old
  - LOW: 15-30 days old
  - UNRELIABLE: >30 days (force update)

4.5 Maintainability
- Code must be explainable in viva (no complex libraries)
- Single developer can debug entire codebase
- Clear separation: UI / Logic / Data

---

5. USER FLOWS (Key Journeys)

Flow 1: First-Time Setup

1. Welcome screen
2. Choose data input method
3. Enter semester dates (auto-detected from uploaded calendar)
4. Enter/upload attendance data
5. Validation and preview

6. Dashboard ready Time: 2-3 minutes

Flow 2: Daily Skip Decision

1. Open app → See today's classes
2. Tap "What if I skip?" on target class
3. See impact preview
4. Decide: Skip or Attend
5. (Later) Mark actual attendance Time: 30 seconds

Flow 3: Crisis Recovery Planning

1. Upload defaulter notice
2. System shows all subjects in deficit
3. Select subject → View recovery plan
4. See if recovery is possible
5. Get week-by-week roadmap Time: 1 minute

Flow 4: Weekly Maintenance

1. Open app Sunday evening
2. Prompt: "Quick check-in for this week"
3. "I missed ___ classes this week"
4. Distribute across subjects (or per-subject)
5. See updated dashboard Time: 20 seconds
6. DATA MODELS 6.1 Semester Configuration json { "semester_id": "even_2025_26", "start_date": "2026-01-19", "end_date": "2026-05-15", "teaching_weeks": 16, "holidays": [{"date": "2026-03-25", "name": "Holi"}, {"date": "2026-04-10", "name": "Good Friday"}], "exam_weeks": [{"start": "2026-05-01", "end": "2026-05-15"}] } 6.2 Subject Model json { "subject_code": "ML", "subject_name": "Machine Learning", "lectures_per_week": 3, "labs_per_week": 1, "tutorials_per_week": 0, "lab_weight": 2, "total_expected_sessions": 80, "attendance": { "conducted": 41, "attended": 28, "last_updated": "2026-01-23", "confidence": "HIGH", "source": "manual_entry" } } 6.3 Calculated State (Derived) json { "subject_code": "ML", "current_percentage": 68.2, "buffer": 0, "deficit": 20, "status": "CRITICAL", "remaining_sessions": 39, "pnr_date": "2026-02-05", "recovery_possible": true, "recovery_plan": { "required_attendances": 22, "weeks_needed": 7, "difficulty": "HARD" } }

---

7. BUSINESS LOGIC RULES

7.1 Attendance Calculation

Current % = (Attended / Conducted) × 100

Buffer = Floor(Total × 0.25) - (Total - Attended) where Total = Conducted + Remaining

Deficit = Ceiling(Total × 0.75) - Attended

7.2 Status Classification

IF current_pct < 75 OR buffer <= 0: status = "CRITICAL" (Red) ELIF buffer <= 1 OR
(pnr_date - today) <= 14 days: status = "RISKY" (Yellow)
ELIF buffer >= 5: status = "SAFE" (Green) ELSE: status = "CAUTION" (Yellow)

7.3 PNR Calculation

min_required = Total × 0.75 current_shortfall = min_required - Attended

IF current_shortfall > Remaining: PNR = "ALREADY PASSED" ELIF current_shortfall <= 0: PNR = "NONE" (Safe) ELSE: sessions_until_pnr = Remaining - current_shortfall weeks = sessions_until_pnr / sessions_per_week PNR = today + (weeks × 7 days)

---

8. CONSTRAINTS & ASSUMPTIONS

8.1 Assumptions
1. Semester has 16 teaching weeks (standard)
2. Lab sessions count as 2× lecture hours
3. Minimum attendance requirement is 75% (institute policy)
4. Students can access timetable and calendar PDFs
5. Monthly defaulter lists are available from faculty
6. Students have smartphones with modern browsers

8.2 Known Limitations
1. Cannot auto-sync with biometric systems (manual entry required)
2. Accuracy depends on user updating data
3. No notification system (students must open app)
4. Single-user only (no class-wide analytics)
5. Cannot account for medical leave adjustments automatically

8.3 Risk Mitigation
| Risk | Mitigation |

Students don't update data	Event-driven prompts, batch updates, work with stale data
Incorrect manual entry	Validation rules, sanity checks, preview before save
Timetable changes mid-semester	Manual adjustment capability, re-import option
Different attendance policies per subject	Configurable minimum % per subject
Data loss	Export/import functionality, localStorage backup

---

9. ACCEPTANCE CRITERIA (Definition of Done)

9.1 MVP Completion Checklist
- [ ] Dashboard shows all subjects with traffic light status
- [ ] Skip simulator works for any class
- [ ] PNR calculated accurately (validated against manual calculation)
- [ ] Recovery plans generated for deficit subjects
- [ ] Data entry via manual + upload works
- [ ] Data validation catches errors
- [ ] Works on mobile (responsive design)
- [ ] Calculations match Excel reference sheet (100% accuracy)
- [ ] Explainable in viva (all formulas documented)

9.2 Testing Requirements
- [ ] Unit tests for all calculation functions
- [ ] Test with sample data from actual defaulter lists
- [ ] Edge cases handled:
  - Subject with 100% attendance
  - Subject with 0% attendance
  - Subject where recovery is impossible
  - Semester with 0 holidays
  - Semester with mid-term break
- [ ] Cross-browser testing (Chrome, Safari minimum)
- [ ] Mobile responsive (375px to 1920px width)

---

10. MILESTONES & TIMELINE

Week 1-2: Foundation
- [x] PRD finalization
- [ ] Design mockups
- [ ] Tech stack setup
- [ ] Data models implementation

Week 3-4: Core Features
- [ ] Attendance calculation engine
- [ ] Dashboard UI
- [ ] Skip simulator

- [ ] PNR calculator

Week 5-6: Data Input
- [ ] Manual entry flow
- [ ] Document upload + parsing
- [ ] Validation layer
- [ ] Fresh start mode

Week 7: Polish
- [ ] Recovery planner
- [ ] 7-day view
- [ ] Error handling
- [ ] Mobile optimization

Week 8: Testing & Documentation
- [ ] End-to-end testing
- [ ] Viva preparation
- [ ] Demo video
- [ ] User manual

---

11. APPENDIX

11.1 Sample Calculations (Reference)

Scenario: ML subject, Week 2 of 16
- Lectures: 3/week × 16 weeks = 48
- Labs: 1/week × 16 weeks × 2 hours = 32
- Total sessions: 80
- Conducted so far: 8 (2 weeks × 4 sessions/week)
- Attended: 6
- Missed: 2

Calculations:

Current % = 6/8 × 100 = 75% Remaining = 80 - 8 = 72 Min required = 80 × 0.75 = 60 Buffer = 60 - (80 - 6 - 72) = 60 - 2 = 58?

CORRECT: Current attended = 6 Total sessions = 80 Min required at end = 80 × 0.75 = 60 Already attended = 6 Still need to attend = 60 - 6 = 54 Remaining sessions = 72 Buffer = 72 - 54 = 18 classes can be missed

Status: SAFE (Green) 11.2 Glossary Buffer: Number of classes that can be missed while staying ≥75% Deficit: Number of additional classes needed to reach 75% Conducted: Total classes held so far (present + absent) PNR: Point of No Return - date when 75% becomes impossible Recovery Path: Step-by-step plan to reach 75% from deficit Document Approved

By:

[Student Name, TE DS, Roll No.]

[Guide Name, Department]

Date: [DD/MM/YYYY]