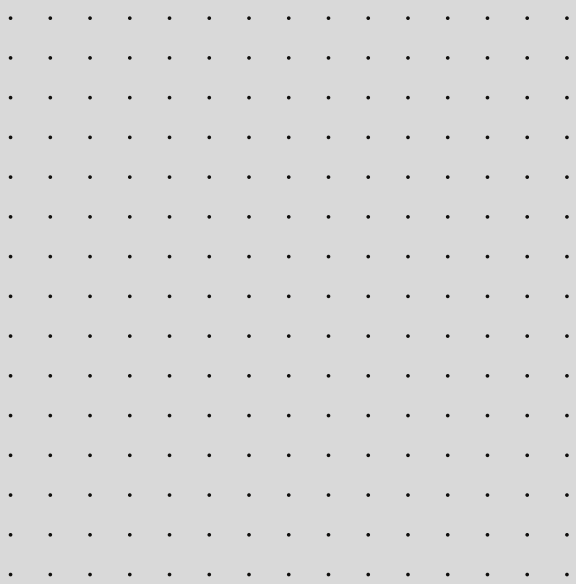


Business Requirements Document

EMPACT

PROJECT OVERVIEW

In many organizations, the Learning & Development (L&D) department faces the challenge of managing and tracking employee training programs, certifications, and development progress. With multiple ongoing programs, it becomes difficult to measure the effectiveness of training, track employee engagement, or align learning initiatives with the company's strategic goals. A comprehensive system is needed to manage training schedules, track progress, automate certification processes, and provide insightful reports on employee development and skill enhancement.

OBJECTIVES

1. **Streamline Training Requirements:** Provide an intuitive interface for managers to submit training requests and requirements.

2. **Automate Baseline Assessments:** Use generative AI to automatically generate relevant assessment questions.

3. **Dynamic Batching:** Divide employees into training batches based on their performance to tailor training duration and content.

4. **Cost Optimization:** Reduce overall training costs by minimizing unnecessary training durations for advanced employees and personalizing training paths.

SCOPE

- Development of a UI for gathering training requirements.
- Integration of OpenAI for automatic baseline assessment generation.
- Evaluation of employee performance using both automated code evaluation and manual review for subjective tasks.
- Automated dynamic batch creation based on performance scores.

KEY FEATURES AND REQUIREMENTS

1. Training Requirement Form:

- User-friendly form for managers to submit their training needs.
- Mandatory fields for training type, duration, audience, and outcomes.

2. Automated Baseline Assessment:

- Use of generative AI (OpenAI API) to create assessment questions.

3. Employee Batching

- Employees will be dynamically divided into batches based on their baseline assessment scores.
- The system will optimize training duration for each batch based on skill level.

4. Assessment Evaluation:

- Automatic evaluation of coding or written responses.
- AI-driven evaluation for open-ended questions or manual grading where necessary.

FUNCTIONAL REQUIREMENTS

Frontend:

- Developed using React and TypeScript.
- Form validation for training requirements input.
- Dashboard displaying training requests, progress, and batch formation in real-time.

Backend:

- Developed using Node.js and microservices architecture.
- Requirement Management Microservice: Handles form submissions for training requests.
- Assessment Generation Microservice: Generates assessment questions using OpenAI.
- Evaluation Microservice: Handles evaluation of employee submissions (code and non-code tasks).

- Notification Microservice: Sends email notifications to employees and managers on training updates.
- Batching Microservice: Automates employee batch formation based on performance scores.

NON-FUNCTIONAL REQUIREMENTS

1. Performance:
   - The platform should be able to handle up to 1000 simultaneous users with minimal latency.
   - Assessments should be generated within 5 seconds of submission.
2. Scalability:
   - The platform should scale to support increasing numbers of training requests and employees.
3. Security:
   - User data should be encrypted in transit (HTTPS) and at rest.
   - Role-based access control (RBAC) will be implemented to ensure only authorized users can access certain functionalities.
4. Reliability:
   - The system should ensure high availability with 99.9% uptime.
5. Usability:
   - The UI should be intuitive and easy to navigate for both technical and non-technical users.

USER STORIES

1. As a Manager, I want to submit training requests through an online form, so I can streamline the process of requesting employee training.

2. As a Manager, I want to specify training duration, type, and target audience, so the platform can cater to specific training needs.

3. As an Employee, I want to receive a baseline assessment, so I can showcase my skill level before the training.

4. As an Employee, I want to be dynamically assigned to a training batch based on my performance, so I can receive training suitable for my skill level.

5. As a Training Administrator, I want to monitor the progress of all training batches through a dashboard, so I can track the efficiency of ongoing training.

STAKEHOLDERS

- Primary Stakeholder: The Head of L&D – the main point of contact for understanding training requirements and validating the platform's output.
- Training Managers: Departmental representatives who will submit training requests.
- Employees: Participants involved in baseline assessments and training programs.
- IT Team: Responsible for maintaining the platform after its launch.

TIMELINE

Phase 1

- Requirements Gathering & Planning
- UI Design and Database Schema Finalization

Phase 2

- Develop Frontend for Training Requirement Form
- Implement Backend Microservices: Requirement Management, Assessment Generation
- Integrate Evaluation Microservice for Baseline Assessments

Phase 3

- Finalize Dashboard and Notification System
- Implement Automated Certificate Generation

Phase 4

- Testing
- Bug Fixes and Optimizations
- Deployment and Final Review

SUCCESS METRICS

1. Reduced Training Costs: Measure the reduction in training costs due to shorter training cycles for advanced employees.

2. Employee Satisfaction: Collect feedback from employees to assess the effectiveness of personalized training.

3. Efficiency of Assessment and Batching: Evaluate how quickly baseline assessments are conducted and how accurately batches are formed based on performance.

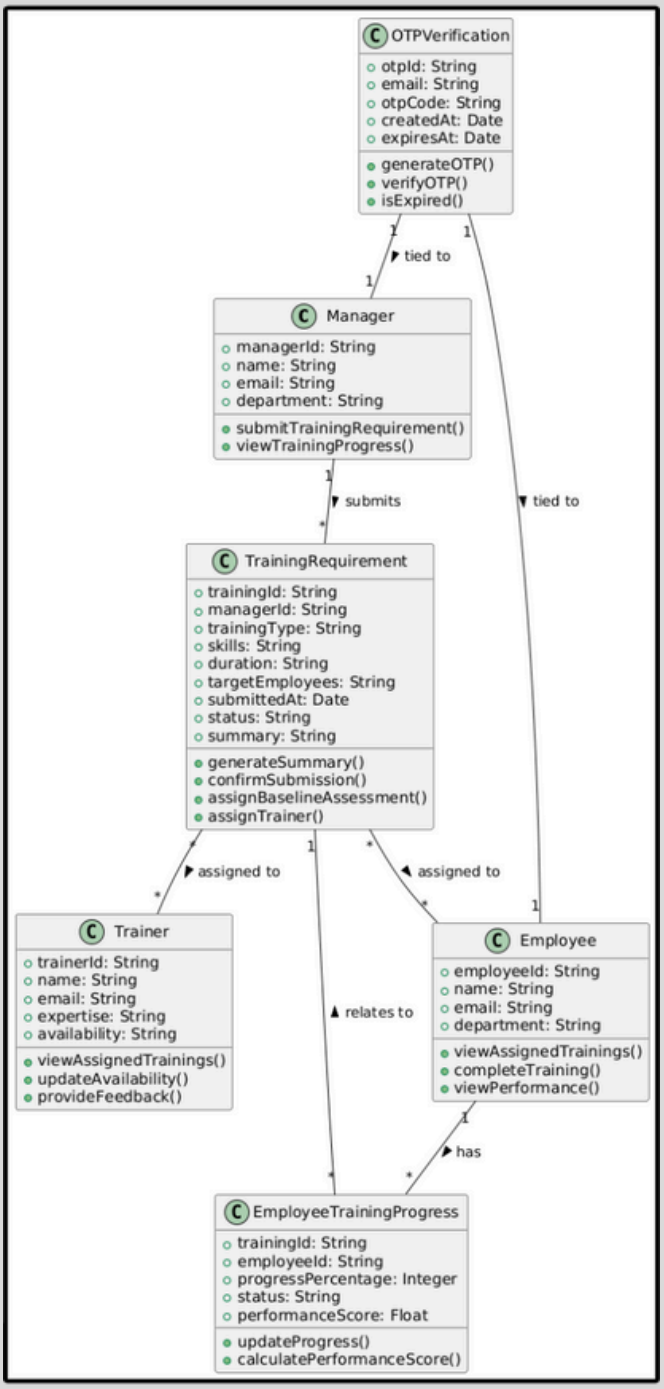
4. Completion Rate: Track the percentage of employees who successfully complete training after batch formation.

5. Automation Efficiency: Measure the time saved by automating the assessment, evaluation processes.

ER DIAGRAM


Learning and Development (L&D) Training Platform
managers
name string
email string
department string
managerId string pk
trainingRequirements
managerId string fk
trainingType string
skills string
duration string
targetEmployees string
submittedAt timestamp
status string
summary string
trainingId string pk
trainers
name string
email string
trainerId string pk
expertise string
availability string
employeeTrainingProgress
trainingId string fk
employeeId string fk
progressPercentage int
status string
performanceScore int
employees
employeeId string pk
name string
email string
department string
otpVerifications
otpId string pk
email string
otpCode string
createdAt timestamp
expiresAt timestamp

UML DIAGRAM


OTPVerification
otpId: String
email: String
otpCode: String
createdAt: Date
expiresAt: Date
generateOTP()
verifyOTP()
isExpired()
1
tied to
1
1
tied to
Manager
managerId: String
name: String
email: String
department: String
submitTrainingRequirement()
viewTrainingProgress()
1
submits
*
TrainingRequirement
trainingId: String
managerId: String
trainingType: String
skills: String
duration: String
targetEmployees: String
submittedAt: Date
status: String
summary: String
generateSummary()
confirmSubmission()
assignBaselineAssessment()
assignTrainer()
*
assigned to
*
1
relates to
*
assigned to
*
1
Trainer
trainerId: String
name: String
email: String
expertise: String
availability: String
viewAssignedTrainings()
updateAvailability()
provideFeedback()
Employee
employeeId: String
name: String
email: String
department: String
viewAssignedTrainings()
completeTraining()
viewPerformance()
1
has
*
EmployeeTrainingProgress
trainingId: String
employeeId: String
progressPercentage: Integer
status: String
performanceScore: Float
updateProgress()
calculatePerformanceScore()