TOPIC: EMPLOYEE MANAGEMENT SYSTEM NAME:PRATEEK

1. Introduction

The purpose of this document is to provide a detailed Low-Level Design (LLD) for the Advanced Employee Feedback Management System. The system facilitates comprehensive feedback collection, performance evaluation, and organizational development through 360-degree feedback, peer reviews, self-assessments, and AI-powered analytics. It employs a desktop application architecture using Python with tkinter GUI framework and MySQL database for local deployment.

This design uses core Python libraries with local AI processing capabilities, ensuring complete data privacy and independence from external services or internet connectivity.

2. Module Overview

The project consists of the following modules:

2.1 User Management

- Manages employee profiles, authentication, role-based access control, and organizational hierarchy.

2.2 Feedback Management

- Handles creation, submission, tracking, and lifecycle management of various feedback types including self-assessment, peer feedback, and manager reviews.

2.3 Local AI Analytics

- Processes feedback content using local Python libraries for sentiment analysis, automated categorization, and priority detection without external dependencies.

2.4 Review and Workflow Management

- Manages feedback review processes, approval workflows, response tracking, and escalation mechanisms.

2.5 Performance Analytics and Reporting

- Provides comprehensive analytics, trend analysis, performance insights, and exportable reports with data visualization capabilities.

2.6 Notification and Communication

- Manages internal notifications, alerts, reminders, and system communications through desktop notifications and in-application messaging.

3. Architecture Overview

3.1 Architectural Style

- Application Type: Desktop Application
- Frontend: Python tkinter with custom widgets and forms
- Backend: Python business logic with direct database connectivity
- Database: MySQL with mysql-connector-python
- AI Processing: Local Python libraries (TextBlob, pandas, numpy, matplotlib)

3.2 Component Interaction

1. Desktop GUI communicates with Python controller classes through event handling.
2. Controller layer processes business logic and coordinates data operations.
3. Model layer interacts with MySQL database for data persistence and retrieval.
4. AI engine processes feedback content locally using offline Python libraries.
5. All components operate within single Python application process without external API calls.

4. Module-Wise Design

4.1 User Management Module

4.1.1 Features

- Add, update, and delete employee profiles
- Role-based authentication and authorization
- Manage organizational hierarchy and reporting relationships

- Department and team structure configuration
- User session management and access control

4.1.2 Data Flow

- User enters authentication credentials via tkinter login form
- System validates credentials against MySQL users table using password hashing
- Upon successful login, user session established with role-based permissions
- Profile updates processed through controller classes and saved to database
- Organizational relationships maintained through foreign key references

4.1.3 Entities

User

- UserID (Primary Key)
- Username
- Email
- PasswordHash
- FirstName
- LastName
- Department
- Role (Employee, Manager, HR, Admin)
- ManagerID (Foreign Key)
- IsActive
- CreatedAt

4.2 Feedback Management Module

4.2.1 Features

- Create, update, and submit feedback entries
- Support multiple feedback types (self, peer, 360-degree, manager)
- Anonymous feedback submission capability
- Draft saving and scheduled submission
- Feedback categorization and rating systems

4.2.2 Data Flow

- Employee initiates feedback creation through desktop interface forms
- Form data validated using Python validation classes before submission

- Feedback content processed by local AI engine for analysis and categorization
- Processed feedback with AI metadata stored in MySQL database
- Status tracking and notifications triggered based on feedback lifecycle
- Real-time updates reflected in user interface components

4.2.3 Entities

Feedback

- FeedbackID (Primary Key)
- EmployeeID (Foreign Key)
- ManagerID (Foreign Key)
- CategoryID (Foreign Key)
- Subject
- Content
- FeedbackType
- Rating
- IsAnonymous
- Status
- SentimentScore
- SentimentLabel
- PriorityLevel
- CreatedAt
- UpdatedAt

4.3 Local AI Analytics Module

4.3.1 Features

- Sentiment analysis using TextBlob and custom algorithms
- Automatic content categorization based on keyword matching
- Priority detection for urgent feedback items
- Trend identification and pattern recognition
- Statistical analysis and insights generation

4.3.2 Data Flow

- Feedback text received from Feedback Management module for processing
- Text preprocessing and normalization performed using Python string operations
- Sentiment analysis executed combining rule-based and TextBlob approaches

- Category classification performed through keyword analysis and scoring
- Priority assessment based on sentiment scores and content indicators
- Analysis results returned with confidence scores and recommendations

4.3.3 Entities

AnalyticsInsight

- InsightID (Primary Key)
- InsightType
- Department
- TimePeriod
- Data (JSON format)
- CreatedAt

4.4 Review and Workflow Management Module

4.4.1 Features

- Manage review assignments and tracking
- Process feedback responses and comments
- Handle escalation workflows for urgent items
- Create and track action items
- Maintain approval history and audit trails

4.4.2 Data Flow

- Feedback submission triggers automated workflow initiation
- System determines appropriate reviewers based on organizational hierarchy
- Review assignments created and internal notifications dispatched
- Reviewer responses processed and linked to original feedback entries
- Action items generated automatically for feedback requiring intervention
- Workflow status updates propagated throughout system components

4.4.3 Entities

FeedbackResponse

- ResponseID (Primary Key)
- FeedbackID (Foreign Key)
- ResponderID (Foreign Key)
- ResponseText

- CreatedAt

ActionItem

- ActionID (Primary Key)
- FeedbackID (Foreign Key)
- AssignedToID (Foreign Key)
- Title
- Description
- Status
- DueDate
- CreatedAt

4.5 Performance Analytics and Reporting Module

4.5.1 Features

- Generate comprehensive performance dashboards
- Create trend analysis and comparative reports
- Provide department-wise and individual analytics
- Export reports in multiple formats (CSV, PDF)
- Visualize data using charts and graphs

4.5.2 Data Flow

- Analytics requests initiated from desktop dashboard interface
- Data aggregation performed using pandas operations on MySQL datasets
- Statistical calculations executed using numpy functions and algorithms
- Charts and visualizations generated using matplotlib library
- Reports formatted and prepared for display or file export
- Dashboard components updated with real-time analytics data

4.5.3 Entities

PerformanceMetric

- MetricID (Primary Key)
- EmployeeID (Foreign Key)

- Period
- OverallRating
- SentimentScore
- FeedbackCount
- CategoryScores
- CreatedAt

4.6 Notification and Communication Module

4.6.1 Features

- Manage in-application notifications and alerts
- Send reminders for pending actions and deadlines
- Provide system announcements and updates
- Handle escalation notifications for urgent items
- Track notification delivery and read status

4.6.2 Data Flow

- System events trigger notification creation through event handlers
- Notification content generated using template-based messaging
- Recipients determined based on roles, relationships, and preferences
- Notifications displayed in desktop interface and stored in database
- Read status tracked and updated when users acknowledge notifications
- Cleanup processes remove old notifications based on retention policies

4.6.3 Entities

Notification

- NotificationID (Primary Key)
- UserID (Foreign Key)
- Type
- Title
- Content
- IsRead
- CreatedAt

5. Database Design

5.1 Entity Relationships

5.1.1 User

- Primary Key: UserID
- Foreign Key: ManagerID (references UserID)

5.1.2 Feedback

- Primary Key: FeedbackID
- Foreign Keys: EmployeeID, ManagerID, CategoryID

5.1.3 FeedbackResponse

- Primary Key: ResponseID
- Foreign Keys: FeedbackID, ResponderID

5.1.4 ActionItem

- Primary Key: ActionID
- Foreign Keys: FeedbackID, AssignedToID

5.1.5 PerformanceMetric

- Primary Key: MetricID
- Foreign Key: EmployeeID

5.1.6 Notification

- Primary Key: NotificationID
- Foreign Key: UserID

6. User Interface Design

6.1 Application Screens

1. Login and Authentication Interface
2. Main Dashboard with Role-based Navigation
3. Feedback Creation and Submission Forms

4. Feedback Review and Management Interface
5. Performance Analytics Dashboard
6. 360-Degree Feedback Collection Wizard
7. User Profile and Settings Management
8. Administrative Console for System Management
9. Reports Generation and Export Interface
10. Notifications and Alerts Center

6.2 Key Interface Components

- Multi-tabbed main application window with role-specific visibility
- Data entry forms with real-time validation and AI preview capabilities
- Sortable and filterable data grids for feedback listings
- Interactive charts and graphs for analytics visualization
- Progress indicators for long-running operations and AI processing
- Context-sensitive help and user guidance tooltips

7. Non-Functional Requirements

7.1 Performance

The system must process feedback submissions and AI analysis within 3 seconds for optimal user experience.

7.2 Scalability

Designed to efficiently handle up to 1,000 employees with responsive performance and optimized database operations.

7.3 Security

Data must be secured using password hashing, role-based access control, and parameterized database queries to prevent unauthorized access.

7.4 Usability

The user interface must be intuitive with minimal learning curve, supporting keyboard navigation and accessibility features.

7.5 Reliability

System must provide comprehensive error handling, data validation, and recovery mechanisms for interrupted operations.

8. Technology Stack

8.1 Core Technologies

- Programming Language: Python 3.9+
- GUI Framework: tkinter with ttk themed widgets
- Database: MySQL 8.0+ with mysql-connector-python
- AI Libraries: TextBlob, pandas, numpy
- Visualization: matplotlib for charts and graphs
- Authentication: hashlib for secure password hashing

8.2 Development Environment

- IDE: PyCharm, Visual Studio Code, or Python IDLE
- Database Tools: MySQL Workbench or command-line interface
- Version Control: Git for source code management
- Testing: Python unittest framework
- Package Management: pip for dependency installation

9. Assumptions and Constraints

9.1 Assumptions

- Target systems have Python 3.9+ installed or can accommodate installation
- MySQL server is available locally or on accessible network infrastructure
- Users possess basic desktop application navigation skills
- System administrators can perform initial database setup and configuration
- Adequate system resources available for AI processing and data visualization

9.2 Constraints

- The system will operate exclusively in local environment without internet connectivity requirements

- No cloud deployment or external service integration planned at this stage
- All AI processing must occur locally using offline Python libraries
- Database operations limited to MySQL platform without multi-database support
- Desktop application only with no web browser or mobile device compatibility
- Concurrent user support limited to shared MySQL database capacity