Group Members: Harsh Dixit and Aarnav Subbaramu

Group Name: BhaiTech Systems

Comp312

14th Jan 2025

Final Project Part- I

Plan for Students Record Management Software

1. Software Requirement Specification (SRS)

1.1 End User Version:

Purpose and Goals (Systems Thinking - Holistic Views & Purpose):

- To streamline and automate student record management processes, reducing manual effort, minimizing errors, and improving data accessibility.

Key Features:

1. **Student Information Management** (Systems Thinking - Elements):
    - Centralized system for storing and updating personal and academic data.
2. **Attendance Tracking** (Enterprise SDLC - Functional Requirements):
    - Faculty can record and review attendance in real-time.
3. **Academic Performance Monitoring**:
    - GPA calculation, subject-wise grades, and semester reports.
4. **Extracurricular Management**:
    - Track and record activities, achievements, and participation.
5. **Secure Login and Role-Based Access** (Enterprise SDLC - Security & Compliance):
    - Different access levels for students, faculty, and administrators.
6. **Reports and Analytics** (Systems Thinking - Feedback Loops):
    - Generate reports for decision-making and feedback.

1.2 Developer Version:

System Overview (SDLC Planning and Systems Thinking):

- Modular design with separation between presentation, logic, and data layers.
- **Technology Stack**:
    - Backend: Python with Flask/FastAPI.
    - Frontend: React for UI.
    - Database: MySQL for relational data management.

Functional Requirements:

1. CRUD operations for all modules (students, attendance, grades).
2. Real-time validation for data entries (Enterprise SDLC - Testing & Quality).
3. Integration with third-party tools for import/export (e.g., MS Excel).

Non-Functional Requirements:

1. Scalability: Handle a growing student base (Enterprise SDLC - Scalability).
2. Performance: Queries and operations should complete in under 2 seconds.
3. Robust security using encryption and secure authentication protocols (DevSecOps practices).

1.3 Acceptance Criteria Document:

- The system must meet user role expectations and operate reliably.
- System performance, including login, search, and reporting, must be efficient.
- Reports must be accurate, and data must be consistent with input records.
- Security protocols must prevent unauthorized access and data breaches.

2. Project Planning

2.1 Roles and Responsibilities (Enterprise SDLC - Agile Teams and Systems Thinking):

Project Manager

- **Role:** Oversee the development of the student record management software, monitor progress, manage deadlines, and address risks to ensure timely delivery.
- **System Element:** Coordinate tasks between modules such as student records, course registration, and fee management, ensuring smooth integration.

Business Analyst

- **Role:** Gather requirements from stakeholders, including administrators, teachers, and students, and map system goals to meet user needs.
- **System Thinking:** Translate the overarching objective of efficient and secure student data management into modular system design.

UI/UX Designer

- **Role:** Design an intuitive and accessible interface for users, such as teachers managing records and students accessing their profiles.
- **System Thinking:** Focus on how usability and accessibility enhance the interaction between system components like dashboards and reports.

Backend Developer

- **Role:** Develop and maintain APIs and business logic for handling student data, such as enrollment, attendance, and grades.
- **SDLC Element:** Ensure readiness for seamless integration with the frontend and database systems.

Frontend Developer

- **Role:** Implement user-facing components like the student portal, administrator dashboard, and real-time notifications.
- **SDLC Element:** Collaborate with the backend team to ensure smooth API integration for features like data retrieval and updates.

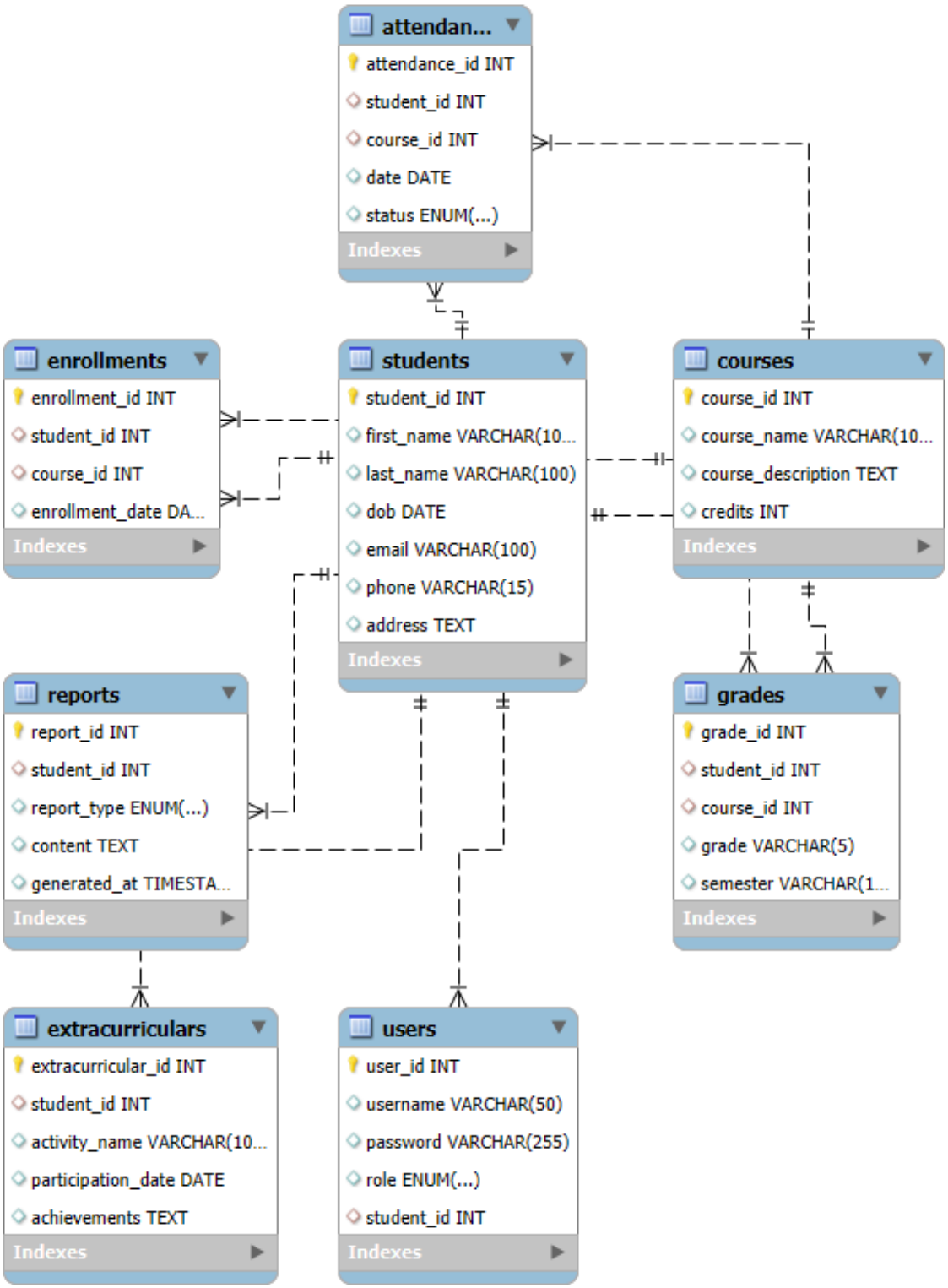
QA Tester

- **Role:** Perform unit tests, integration tests, and user acceptance tests to validate the functionality of modules, such as login systems and report generation.
- **System Thinking:** Analyze testing outcomes to identify feedback loops, such as the impact of data entry errors on reports or interdependencies between modules.

2.2 Timeline (Gantt Chart):

Task	Start	End	Duration	Responsible
Requirement Gathering	Day 1	Day 5	5 days	Business Analyst
System Design	Day 6	Day 12	7 days	UI/UX, Architect
Backend Development	Day 13	Day 27	15 days	Backend Developer
Frontend Development	Day 13	Day 27	15 days	Frontend Developer
Integration	Day 28	Day 30	3 days	Developers
Testing	Day 31	Day 38	8 days	QA Tester
Deployment	Day 39	Day 40	2 days	All Teams
Feedback and Training	Day 41	Day 45	5 days	Admins, Trainers

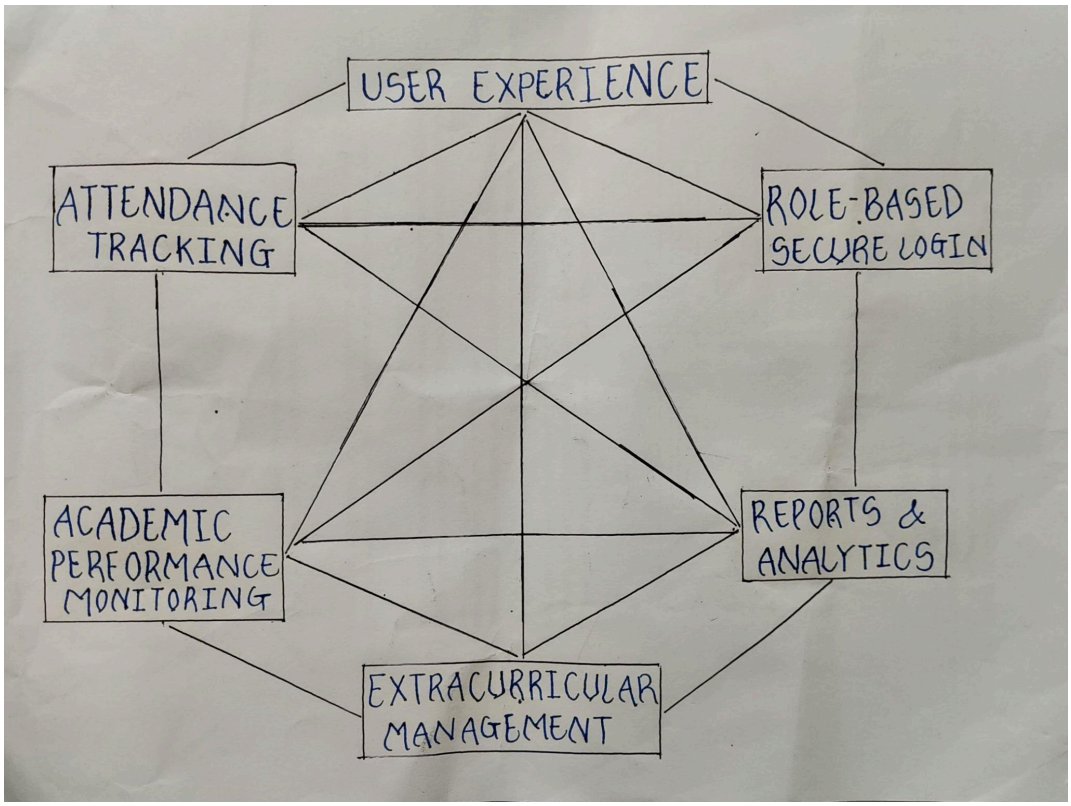
2.3 Sample ER Diagram



3. Delivery and Application of Knowledge

3.1 Systems Thinking Application:

- **Holistic Approach:** Consider interconnections between users, data, and system reliability.
- **Feedback Loops:** Create mechanisms for error correction (e.g., validation alerts during data entry).
- **Emergent Behavior:** Observe how combining features like analytics and reporting enhances decision-making.



The diagram showcases the interconnections between key components of the Students Record Management Software, using a Systems Thinking approach. **User Experience (UI/UX Design)** is central, ensuring usability across features like **Attendance Tracking** and **Academic Performance Monitoring**. **Extracurricular Management** links with both attendance and performance, while **Reports and Analytics** draw data from all elements to inform decision-making. **Secure Login and Role-Based Access** ensures appropriate user access across all components, maintaining security and smooth system operation. This interconnected design promotes efficiency and a cohesive user experience.

3.2 Enterprise SDLC Application:

- **Iterative Delivery:** Use Agile (Scrum) with sprints for incremental feature releases.
- **Automation:** Implement CI/CD pipelines for reliable deployments (Enterprise SDLC - CI/CD).
- **Testing and Quality:** Focus on functional and performance testing using tools like Pytest and Selenium.

3.3 Expected Delivery:

1. **Core System:**
    - Fully functional application accessible via web browsers.
    - Secure login with role-based dashboards.
2. **Reports and Analytics:**
    - Comprehensive grade, attendance, and activity reports.
3. **Training and Documentation:**
    - User guides for faculty and admin roles.
4. **Future Scalability:**
    - Modular architecture supporting future integrations (e.g., mobile app or additional data points).

This plan integrates the **SDLC framework** and **Systems Thinking methodology**, ensuring a comprehensive approach to the development and delivery of the **Students Record Management Software**.