Section I: Cover Page

- **Project Title**: **School Management System**
- **Student**: Omar Ahmed
- **Student ID**: 922505867
- **GitHub Username**: omarazaz1

Checkpoint #	Date Submitted
Checkpoint III	April 02, 2024.

Section II: Table of Contents

Section III: Project Description

A school management system database is intended to address various urgent difficulties encountered by educational institutions today, such as fragmented data management, poor communication routes, and the time-consuming manual processing of academic and administrative work. This system's single platform intends to simplify operations, improve data accuracy and increase stakeholder participation across the board.

The fundamental impetus for developing the school management system database comes from the requirements to:

Consolidate data Management: Schools frequently use distinct systems for grades attendance, scheduling, and financing, resulting in data silos and inefficiencies.

Improve Communication: The lack of integrated communication tools makes it difficult for instructors, students, and parents to communicate successfully.

Automate Administrative Tasks**:** Manual administration operations are time-consuming and prone to mistakes, thus an automated solution is required to free up important resources.

System Overview

As its core, the School Management System database serves as the backbone for a comprehensive platform that integrates various functions such as student information management, academic records, scheduling, and financial transactions. By centralizing data, the system ensures that information is consistently updated and easily accessible, facilitating more informed decision-making and streamlined operations.

Unique Features

The School Management System will introduce several unique features designed to enhance usability and functionality:

Customizable Dashboards: Users will have personalized dashboards that provide fast access to the most relevant information.
Advanced Analytics: The system will offer advanced analytics capabilities, enabling the generation of insights into academic performance, resource utilization, and financial management.

Integrated Communication Tools: A built-in messaging and notification system will support direct communication between teachers, students, and parents, fostering a collaborative community.

Enhanced Security and Privacy: With robust data encryption and access controls, the system will ensure the security and privacy of sensitive information.
Market Relevance

<u>Two existing software tools that would significantly benefit from integrating with the School Management System database are:</u>

Canvas: learning management system, Canvas could leverage the School Management System database to enhance its capabilities in scheduling, real-time analytics, and integrated communication tools. This integration would allow for a more seamless user experience, where academic and administrative data are synchronized across platforms, enabling educators and students to access a wider array of services within a single interface.

QuickBooks for Education: Known for its financial management solutions, QuickBooks could benefit from integration with the School Management System database by streamlining the financial operations of educational institutions. This would include automated billing, tuition management, and scholarship disbursements, linked directly to student and course data. The integration would provide schools with a holistic view of their financial health, directly tied to academic metrics and enrollment data, facilitating better financial planning and analysis.

Section IV: Functional Database Requirements

1. User

1.1. User shall create only one account.

1.2. User shall be able to log into the system from multiple devices.
1.3. User shall have at least one role (eg , Student, Teacher, Administrator).
1.4. User can be assigned to multiple classes.
1.5. User shall update their personal information, including password, email, and contact number.

2. Account

2.1. An account shall be created by only one user.
2.2. account shall have a unique username and password.
2.3. account can be deactivated or reactivated by administrators only.
2.4. account shall log the last login time and date.

3. Role
3.1. A role shall be linked to many users .
3.2. A role defines access and permissions within the system.
3.3. Roles include Administrator, Teacher, Student, and Parent.

4. Student
4.1. A student shall be enrolled in multiple courses.
4.2. A student shall have a unique student ID.
4.3. A student can have multiple guardians .
4.4. A student shall have grades recorded for each course enrollment.
4.5. A student's enrollment history shall be trackable over time.

5. Teacher

5.1. Teacher can teach multiple courses.
5.2. Teacher shall create, update, and grade assignments.
5.3. Teacher can advise multiple students
5.4. Teacher shall have a schedule including all classes they teach.
5.5. Teacher can access and update student performance records.

6. Course

6.1. A course can have multiple enrolled students .
6.2. course shall be assigned to at least one teacher .
6.3. course may have prerequisites .
6.4.   course includes multiple assignments and exams .
6.5. course shall have a defined maximum capacity for students.

7. Grade

7.1. A grade is associated with a student and a course .
7.2. Grades include assignments, exams, and final marks.
7.3. A grade shall be entered by the teacher of the course.
7.4. A student can view grades for their courses.
7.5. Grades can be revised by a teacher within a defined period.

8. Assignment

8.1. An assignment belongs to a course .
8.2. An assignment shall have a submission deadline.
8.3. Students can submit assignments multiple times until the deadline.
8.4. Assignments can be graded with feedback provided to students.

9. Schedule

9.1. A schedule is associated with each student and teacher .
9.2. A schedule includes class times, locations, and exam dates.
9.3. Schedules can be updated to reflect changes in course timings or locations.

10. Classroom

10.1 A classroom shall be assigned to multiple courses throughout the academic year .
10.2. A classroom has a defined capacity limit.
10.3 Classroom assignments consider course capacity and equipment needs.

11. Parent/Guardian

11.1. A parent/guardian can be linked to multiple students.
11.2 Parents/guardians can view their children’s academic progress and attendance.
11.3. Parents/guardians receive notifications for academic events, grades, and absences.

12. Attendance

12.1. Attendance records are kept for each student for each class session.
12.2. Teachers mark attendance for each class session.
12.3. Attendance records can be reviewed by students, parents, and administrators.

13. **Exam**

13.1. Exam shall be scheduled for the classes .
13.2.Exams grades are input to the system by the teachers.

14. **School Event**

14.1. School events can be added to the calendar.
14.2. Events can have reservation capabilities for participants.

15. **Library Resource**

15.1. Users can check the library resources.
15.2. Setup due dates for the return

16. **Textbook**

16.1. Textbooks should be assigned for the courses.
16.2. Textbooks can be checked out from the library.

Section V: Non-functional Database Requirements

1. Performance

1.1. The database system shall support concurrent access by up to 1000 users without significant degradation in response times.
1.2. The system shall process transactions within 2 seconds under normal load conditions.

1.3. Query response times shall not exceed 5 seconds for complex queries under normal operational conditions.
1.4. The system shall provide real-time updates to user interfaces within 1 second of data changes.
1.5. Batch processing of daily attendance and grading updates shall be completed within a maximum of 1 hour.

2. Security

2.1. All user passwords shall be stored using secure hash algorithms.
2.2. The system shall implement role-based access control (RBAC) to limit access to sensitive data based on user roles.
2.3. The database shall enforce data encryption in transit and at rest.
2.4. The system shall support secure login mechanisms, including two-factor authentication for administrative roles.
2.5. Regular security audits shall be performed, with vulnerabilities addressed within a defined timeframe.

3. Scalability

3.1. The database shall be designed to easily scale horizontally to accommodate growth in user numbers and data volume.
3.2. The system shall support the addition of new schools, each with up to 2000 students, without significant changes to the database schema.
3.3. The system's architecture shall support the distribution of database load across multiple servers.
3.4. The database shall maintain performance levels with a 10% annual growth in data volume.
3.5. The system shall provide mechanisms for easy backup and restoration to ensure data integrity during scaling operations.

4. Reliability

4.1. The database system shall achieve 99.9% uptime, excluding scheduled maintenance.
4.2. The system shall automatically recover from common database failures without data loss.
4.3. Data backup shall be performed daily, with the capability to restore data to any point within the last 30 days.
4.4. The system shall provide real-time data replication to a secondary site to ensure availability in case of a primary site failure.
4.5. All critical components of the system shall have redundancy to minimize single points of failure.

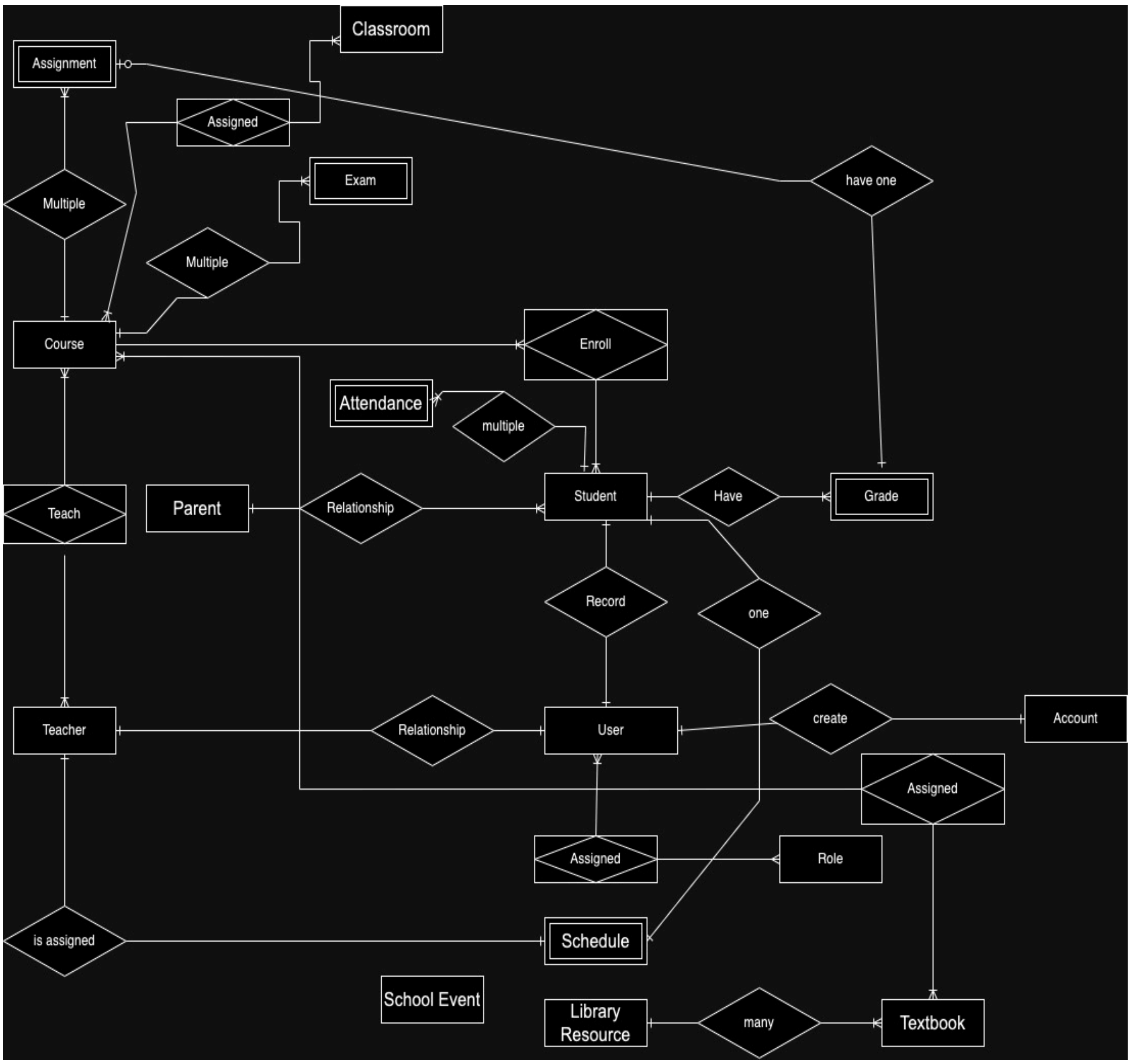
5. Usability

5.1. The system shall provide an intuitive web-based interface accessible on both desktop and mobile platforms.
5.2. User guides and online help shall be available for all features within the system.
5.3. The system shall allow users to customize their dashboard views according to their preferences and roles.
5.4. The interface shall support multiple languages, including English, Spanish, and French.
5.5. The system shall provide accessibility features for users with disabilities, complying with ADA (Americans with Disabilities Act) standards.

6. Maintainability

6.1. The system shall be designed using modular architecture to facilitate easy updates and maintenance.
6.2. Code and design documentation shall be provided, adhering to industry standards for readability and maintainability.
6.3. The system shall support automated deployment processes to streamline updates and patches.
6.4. Diagnostic tools shall be included to monitor system health and facilitate troubleshooting.
6.5. The database shall support version control mechanisms for schema and data migrations.

Section VI: Entity Relationship Diagram (ERD)



Section VII: Entity Description

1. User (Strong)
   a. user_id: PK, numeric.
   b. Name: multivalue alphanumeric
   c. Last_name: multivalue alphanumeric

d. username: Alphanumeric.
e. email: key, Alphanumeric.

2. Account (Strong)
    a. account_id:PK, numeric.
    b. user_id: week key, numeric
    c. username: key, Alphanumeric.
    d. password:key, Alphanumeric.

3. Role (Strong)
    a. role_id: PK, numeric.
    b. name: Alphanumeric.
    c. Description: text

4. Student (Strong)
    a. student_id: PK, numeric.
    b. user_id: Foreign key, numeric.
    c. parent_id :key,multivalue, numeric.

5.Teacher (Strong)
    a. teacher_id: PK, numeric.
    b. user_id: Foreign key, numeric.
    c. department: Alphanumeric.

6. Course (Strong)
    a. course_id: PK, numeric.
    b. Title: key, Alphanumeric.
    c. Teacher_id: FK, numeric.
    d. Max_capacity: numric

7. Grade (Weak)
    a. grade_id: PK,numeric.
    b. student_id: FK, numeric.
    c. course_id: FK, numeric.

8. Assignment (Weak)
    a. assignment_id:PK, numeric.
    b. course_id: FK, numeric.
    c. Title: alphanumric
    d. deadline: key, numeric ”Timestamp”.

9. Schedule (Weak)
    a. schedule_id: PK, numeric.
    b. user_id: Foreign key, numeric.

    c. Class_time: key, Alphanumeric.

10. Classroom (Strong)
    a. classroom_id: PK, numeric.
    b. capacity: key, Numeric. Maximum number of occupants.
    c. Equipment: Alphanumeric.

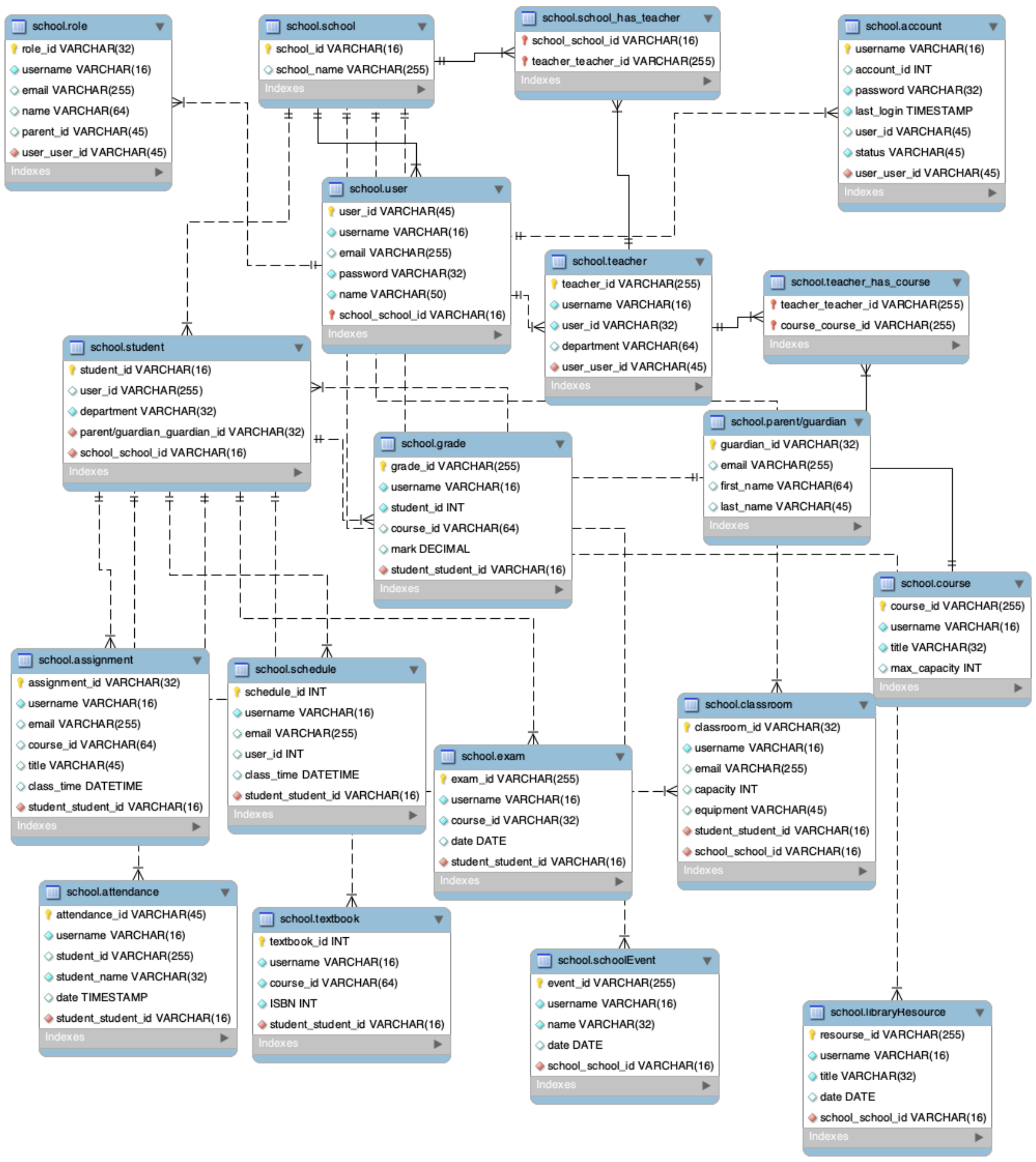
11. Parent/Guardian (Strong)
    a. parent_id: PK, numeric.
    b. name: Alphanumeric.
    c. last_name: Alphanumeric.
    d. email: key, Alphanumeric

12. Attendance (Weak)
    a. attendance_id: PK, numeric.
    b. student_id: FK, numeric.
    c. date: key, numeric "Timestamp".

13. Exam (Weak)
    *exam_id: PK, numeric.
    *course_id: FK, numeric.
    *date: FK, numeric.

14. School Event (Strong)
    a. event_id: PK, numeric.
    b. name: Alphanumeric.
    c. date: FK, date.
15. Library Resource (Strong)
    a. resource_id: PK, numeric.
    b. title: key, Alphanumeric.
    c. type: Alphanumeric.
16. Textbook (Strong)
    a. textbook_id: PK, numeric.
    b. course_id: FoK, numeric.
    c. ISBN: Alphanumeric.

Section VIII: Entity Establishment Relationship Diagram



Section IX: Constraints Description

Table	FK	ON DELETE	ON UPDATE	Comment

Account	user_id	CASCADE	CASCADE	If a user deletes their account, the account will still be there for the record.
Grade	student_id	SET NULL	CASCADE	If a student is deleted, their grades are kept for the record.but not visible for the student .
Grade	course _id	SET NULL	CASCADE	If the course is deleted, grades associated with the course are kept for records.
Assignment	Course_id	CASCADE	CASCADE	Assignment should be deleted if the course is deleted.
Schedule	user_id	CASCADE	CASCADE	
Enrollment	student_id	CASCADE	CASCADE	If student is deleted, their enrollment records should also be removed.
Enrollment	course_id	CASCADE	CASCADE	If a course is deleted all enrollments for that course should be removed.
Student	parent_id	SET NULL	CASCADE	If a parent is deleted, the student record remains but the parent reference is NULL ,while update parent information.
Attendance	student_id	SET NULL	NO ACTION	
Exam	course_id	CASCADE	CASCADE	Exams should be deleted if their corresponding course is deleted to maintain the validity of data .Updates to the course should

				update the exam details.