Team:

- **BSCS15002 ----> Syed Taha Bilal Gilani**
- **BSCS15009 ----> Muhammad Sannan Nasir**
- **BSCS15037 ----> Muhammad Manan Qasim**
- **BSCS15072 ----> Muhammad Hamza Naveed**

Project Name: School Management/Administration System

Platform: Windows

Framework: WPF .NET

DBMS : SQLITE

Requirements:

Student Oriented Controls

- Store, Edit , Delete Student records.
- Filter student records.
- Search student records.
- Profile view for students displaying additional information(fee information, academic records, attendance)
- Attendance management System
- Generate and print student transcript

Staff Oriented Controls

- Store, Edit , Delete Staff information
- Create, Edit ,Delete Time Tables
- Manage Staff Attendance
- Staff salary management
- Staff allocation
- Statistics based on class assigned to teacher

Administrative controls

- Access control and user authentication
- Display total revenue generated (monthly, yearly)
- Track fee information for students
- Keep tracks of arrears in student fee
- Keep track fee collected per class
- Keep track of expenses
- Generate annual report (Growth rate, revenue, profit)
- Generate and send SMS alerts
- Generate and print fee vouchers

Question 1:

Do you think you mathematical verification of correctness of your system or a part of your system or a part of your system? Why?

We don't need any mathematical verification of our system because it doesn't contain any algorithm ,timed automata, state automata and any other mathematical model that needs a mathematical proof of its efficient performance. We can prove operational semantics, denotational semantics and logical prove for the security and correctness of computer language but we don't need them because we are using already verified language.

Question 2:

Can you separate various concerns of your project from functional and quality perspectives? Highlight the concerns and describe how can you handle concerns separately?

- Our project can be smartly divided into separate concerns for ease.
- Our program can be divided into three different sections
  - **Student Oriented Controls**
  - **Staff Oriented Controls**
  - **Administration Oriented Controls**
- Each section address a separate control and further divided into modules, Student Oriented controls the student activities like student attendance, student record in school e.t.c and don't have any link up with teacher and administration , similarly staff have his own record don't have any concern with administration and student and then administration that gets the student and staff record for their own perspective.

Some quality concerns are:

- Atomicity of transactions in case of multiple clients.
- User-friendly and intuitive user interface.
- Robust and responsiveness.
- Application resource optimization so that the application can run with least amount of memory.
- Database compression to save space in case of increased number of clients

Question 3:

Identify some functional modules in your system. Discuss coupling and cohesion aspects.

Modules present in our system include:

1. New Admissions of Students.
2. Fee Voucher generation for students
3. School Reports of Students (results + monthly progression)
4. Student Records (related to paid fees and arrears)
5. Teachers Record(employment + salary management)
6. Time Tables division for classes
7. Attendance system for students and teachers
8. Notification system for students and teachers

Modules for the projects will be modeled following the Factory Data Pattern so it will be made sure that we have loose coupling and better cohesion so it has the potential of retaining the concepts of Modularity, Abstraction, Anticipation of change, Generality and Incrementality

Question 4:

Identify the potential future changes in your system. Pick one potential change and discuss how would you address it in your system?

Potential future change could be usage of this system of multiple systems, let's say multiple branches of same school. We have to change the data management layer for it to support multiple machines on a same database.

Another possible change could be integration of an online learning management system for students.

Question 5:

Which increments would your suggest if you are asked to build your system incrementally?

Building system incrementally, building and testing module/modules separately and then connecting the blocks to make the system after testing working of one modules or multiple models together separately. Each iteration passes through the requirements, design, coding and testing phases and each subsequent release of the system adds function to the previous release until all designed functionality has been implemented.

If we follow an incremental approach for our project, module wise priority would be

- Basic CRUD operations for Students
- Basic CRUD Operations for Staff
- Basic CRUD controls for Administration
- Attendance Management and integrating it with Staff and Students
- Salary and Fee Management for Staff and Students Respectively
- Grading System for the Students
- Quality Assurance system for the Staff
- Notification System for (staff + students)