Secure Student Records Management System (SRMS)

1. Overview of the System

The **Secure Student Records Management System (SRMS)** is a database-driven academic management system designed with **security-first principles**.
The system integrates a **Python Tkinter GUI** with a **SQL Server backend**, where all sensitive logic, authorization, and data protection are enforced **inside the database**.

The GUI acts only as a **presentation layer**, while SQL Server is responsible for:

- Authentication
- Authorization
- Data confidentiality
- Integrity enforcement
- Auditing

2. Security Models Used in the System

The SRMS system implements **five complementary security models**, each addressing a different aspect of system security

2.1 Authentication (Database-Level Authentication)

- User authentication is handled entirely in SQL Server using the stored procedure:
    - sp_User_Login
- Passwords are **never stored in plain text**.
- Passwords are hashed using:
    - HASHBYTES('SHA2_256', NVARCHAR)
- Hash comparison is done inside SQL Server.
- On successful login, the database returns:
    - Username
    - Role
    - Clearance Level

Result:
The application never verifies passwords locally; authentication is fully centralized and secure.

2.2 Role-Based Access Control (RBAC)

1. Each user is assigned one role.
2. Access to operations is restricted based on the user's role.
3. RBAC is enforced using:
4. Role checks
5. Role hierarchy (via RBAC_RANK table)
6. Centralized enforcement is implemented in:
7. sp_CheckAccess
8. RBAC guarantees that users can only perform actions explicitly allowed for their role

2.3 Mandatory Access Control (MAC / MLS – Bell–LaPadula)

The system implements **Multi-Level Security (MLS)** using the **Bell–LaPadula mode**

meaning	level
puplic	1
student	2
TA	3
Instructor	4
admin	5

Rules Applied:

- **No Read Up**: users cannot read data above their clearance.
- **No Write Down**: users cannot write data below their clearance.

MLS rules are enforced in:

- sp_CheckAccess

This prevents data leakage even between valid roles

2.4 Cryptographic Data Protection

Sensitive data is protected using **SQL Server encryption**.

Encryption Hierarchy:

1. **Database Master Key**
2. **Certificate**
3. **AES-256 Symmetric Key**

Encrypted Data Includes:

- Usernames (stored encrypted)
- Student phone numbers
- Grades
- Any sensitive attribute classified above "Public"

Encryption and decryption occur **only inside stored procedures**, never in the GUI

2.5 Auditing and Accountability

- All sensitive or privileged actions are logged.
- Logging is done via:
    - sp_LogAction
- Logs are stored in:
    - LOGS table
- Logs are accessed using:
    - vw_Admin_Logs (read-only view)

This ensures **non-repudiation** and full traceability of actions

3. System Roles

The SRMS system defines **five distinct roles**, each with strictly limited permissions.

3.1 Admin

- Full system control
- Manages users, roles, courses, and assignments
- Approves or denies role requests
- Views audit logs (read-only)

3.2 Instructor

- Views assigned courses
- Views students in their courses
- Inserts, updates, and deletes grades
- Views attendance
- Can view safe aggregated statistics

3.3 Teaching Assistant (TA)

- Views assigned courses
- Views students
- Manages attendance records
- Cannot access grades

3.4 Student

- Views own profile
- Updates own phone number
- Views own courses, grades, and attendance
- Can submit role upgrade requests

3.5 Guest

- Views public (unclassified) course information only
- No access to private or sensitive data

4. Stored Procedures (SPs)

All database operations are executed through **Stored Procedures only**.
Direct table access is not allowed

4.1 Authentication and Session

- sp_User_Login
- sp_CheckAccess

4.2 User Management

- sp_User_Register
- sp_User_Create
- sp_User_UpdateRole
- sp_User_Delete
- sp_User_GetAll

4.3 Course Management

- sp_Admin_CreateCourse
- sp_Admin_UpdateCourse
- sp_Admin_DeleteCourse
- sp_Admin_GetCourses
- sp_Get_PublicCourses

4.4 Assignment Management

- sp_Admin_AssignInstructorToCourse
- sp_Admin_UnassignInstructorFromCourse
- sp_Admin_AssignTAToCourse
- sp_Admin_UnassignTAFromCourse

- sp_Admin_EnrollStudent
- sp_Admin_RemoveStudent

4.5 Instructor Operations

- sp_Instructor_ViewCourses
- sp_Instructor_ViewStudentsByCourse
- sp_Instructor_SaveGrade
- sp_Instructor_DeleteGrade
- sp_Instructor_ViewGradesByCourse
- sp_Instructor_ViewAttendanceByCourse

4.6 TA Operations

- sp_TA_ViewCourses
- sp_TA_ViewStudentsByCourse
- sp_TA_ViewAttendance
- sp_TA_AddAttendance
- sp_TA_UpdateAttendance
- sp_TA_DeleteAttendance

4.7 Student Operations

- sp_Student_ViewProfile
- sp_Student_UpdateOwnPhone
- sp_Student_ViewCourses
- sp_Student_ViewGrades
- sp_Student_ViewAttendance

4.8 Role Request Workflow

- sp_RoleRequest_Submit
- sp_RoleRequest_GetPending
- sp_RoleRequest_Approve
- sp_RoleRequest_Deny

5. Database Views

Views are used for **controlled, read-only access**

`vw_Admin_Logs`	Allows Admin to view audit logs safely
(Other views)	Used to expose filtered or decrypted data safely

6. Database Schema Description

USERS	**System accounts with role and clearance**
STUDENT	**Student personal data**
INSTRUCTOR	**Instructor data**
TA	**Teaching Assistant data**
COURSE	**Course information**
GRADES	**Encrypted grades per student**
ATTENDANCE	**Attendance records**
COURSE_STUDENT	**Student enrollment**
INSTRUCTOR_COURSE	**Instructor-course assignment**
TA_COURSE	**TA-course assignment**
ROLE_REQUESTS	**Role upgrade requests**

LOGS	**Audit trail**
RBAC_RANK	**Role hierarchy**

6.2 Design Principles

- **Soft deletes instead of hard deletes**
- **Encrypted sensitive attributes**
- **Foreign key integrity**
- **Centralized access control**
- **Clear separation of duties**

The SRMS project demonstrates a secure, database-centric architecture where:

- **Security decisions are enforced at the database level**
- **The GUI is intentionally thin and untrusted**
- **Multiple security models work together**
- **The system is resistant to misuse, escalation, and data leakage**

This design follows best practices in secure database systems and multi-level access control.

7. System Screenshots and Functional Demonstration

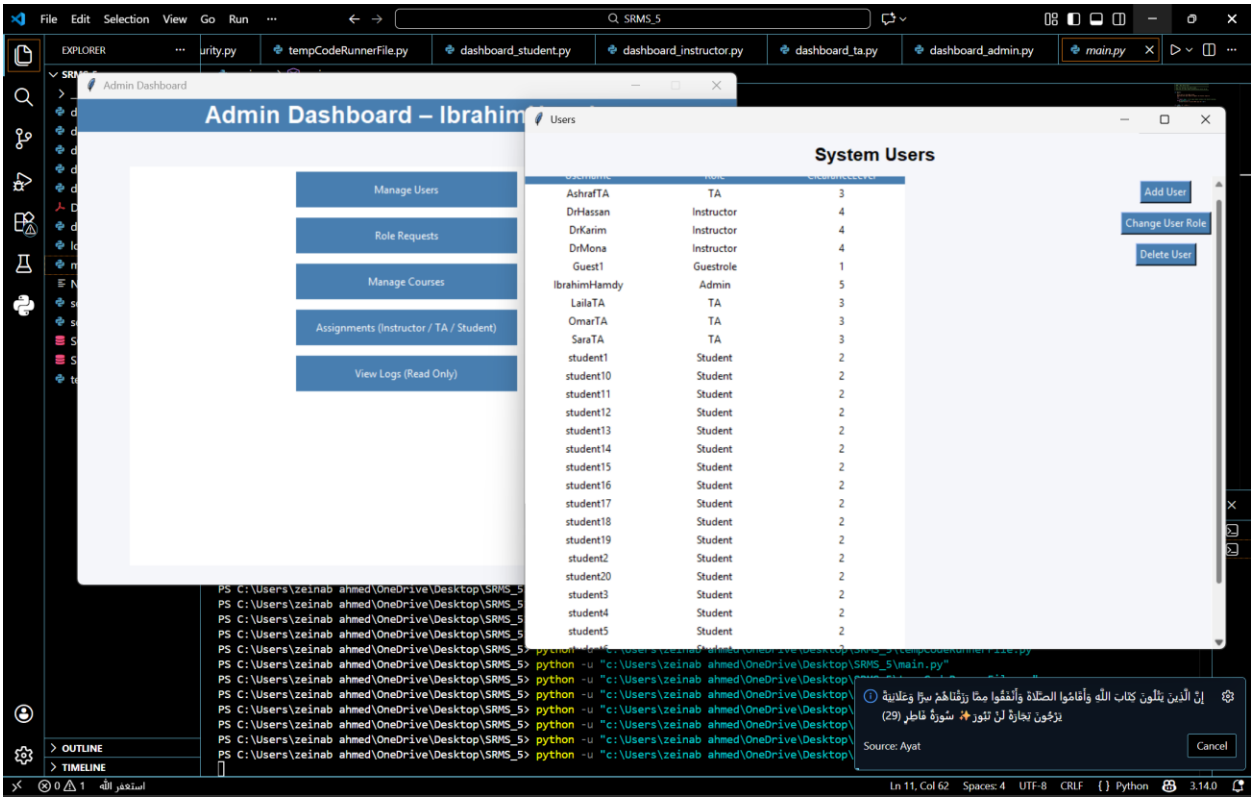
This section presents selected screenshots from the SRMS application to demonstrate the **system functionality, role-based access control, and security enforcement**.
Each screenshot corresponds to a specific module and highlights how the system design principles are applied in practice.

7.1 Admin Dashboard – Main Control Panel

Figure 7.1: Admin Dashboard – Main Control Panel

This screen represents the main dashboard available only to users with the **Admin role** and the highest clearance level.
From this dashboard, the admin can manage users, review role requests, manage courses, assign instructors, teaching assistants, and students to courses, and view system logs in read-only mode.

This dashboard serves as the **central administrative control point** of the SRMS system.



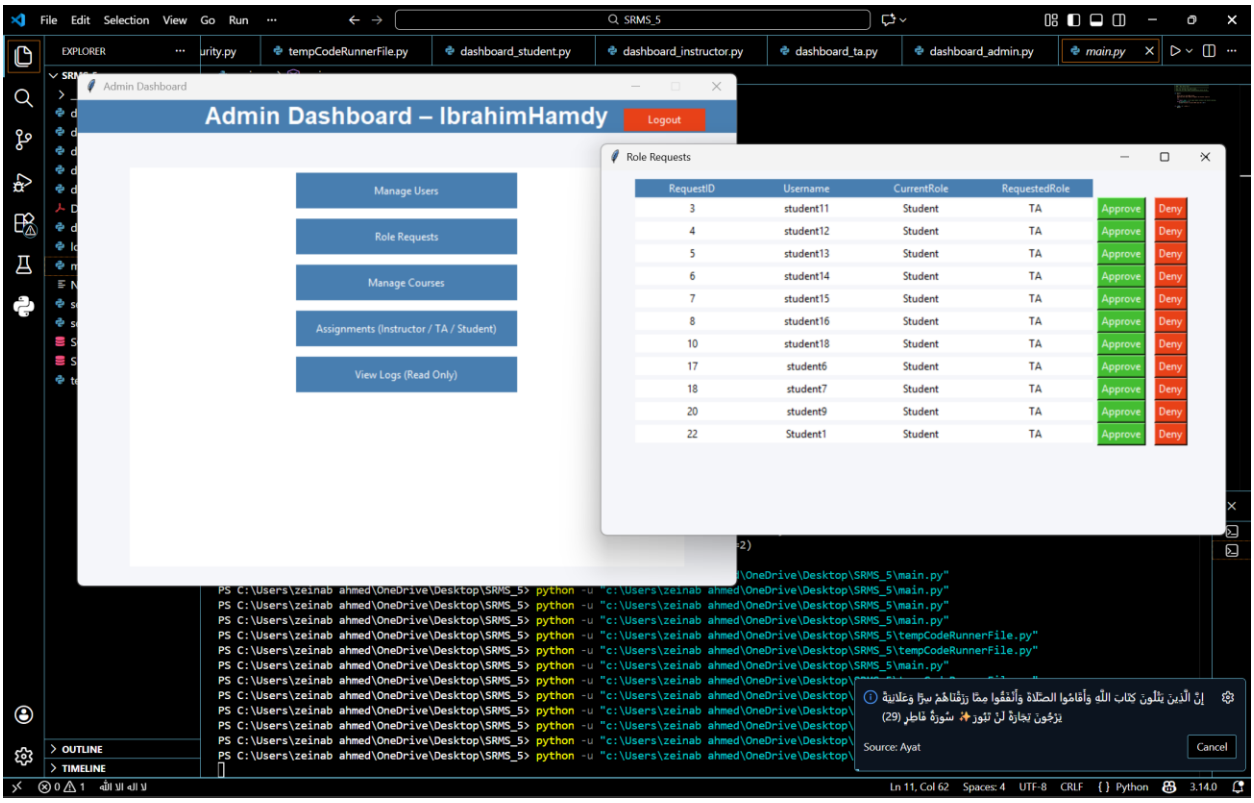
7.2 System Users Management

Figure 7.2: System Users Management Screen

This screen displays all system users along with their assigned roles and clearance levels. The admin can perform the following actions:

- Add new users
- Change user roles
- Delete users

All user management operations are executed through **stored procedures**, ensuring that **Role-Based Access Control (RBAC)** rules are strictly enforced at the database level.



7.3 Role Upgrade Requests Management

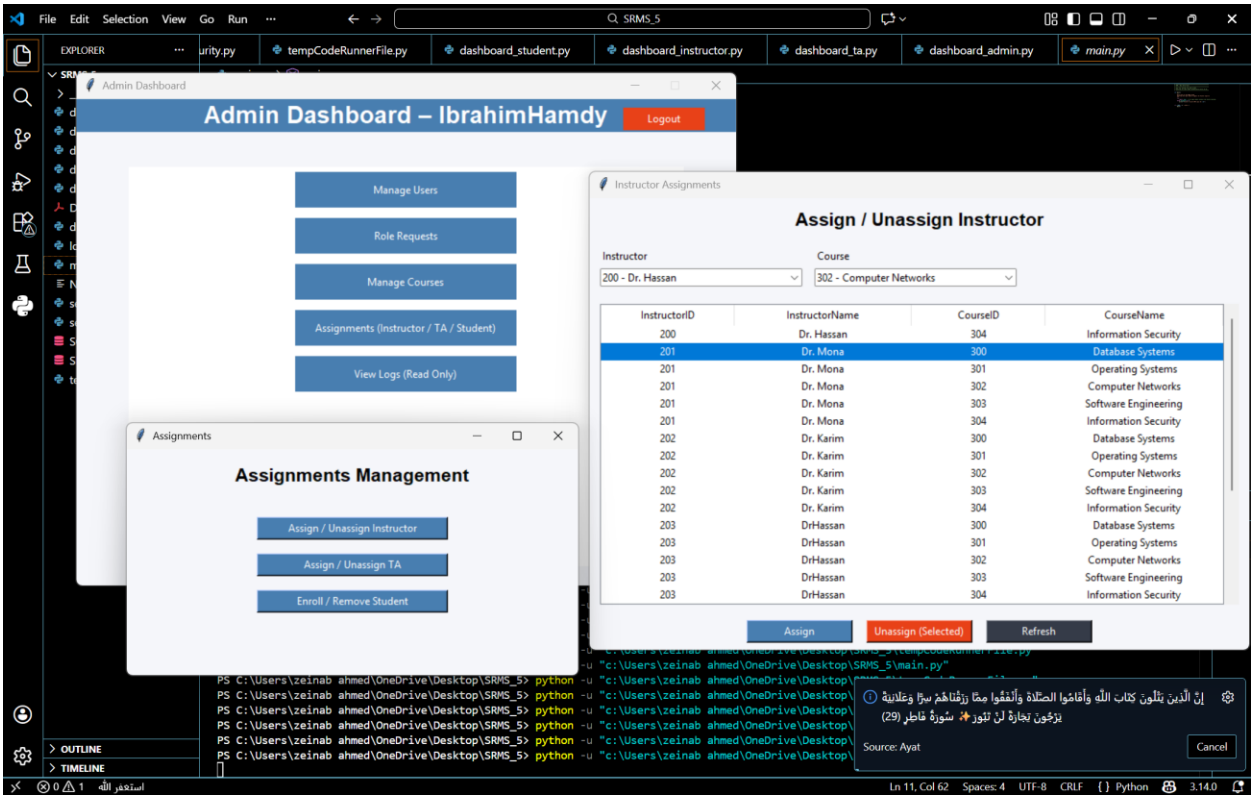
Figure 7.3: Role Upgrade Requests Management Screen

This screen shows pending role upgrade requests submitted by students.
The admin can approve or deny each request through dedicated controls.

When a request is processed:

- The user role is securely updated in the database
- The request status is changed accordingly
- The action is logged in the system audit logs

This workflow ensures **controlled role transitions** and prevents unauthorized privilege escalation.



7.4 Instructor Assignment Management

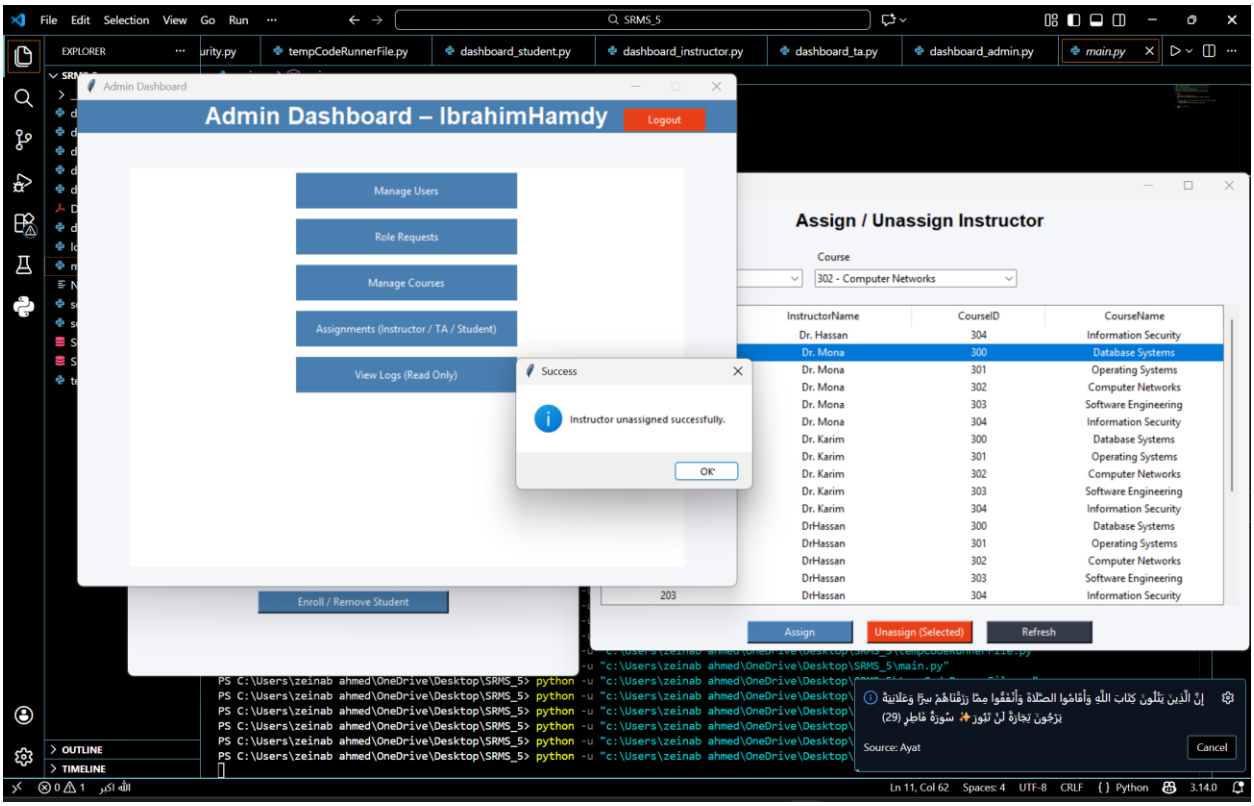
Figure 7.4: Assign / Unassign Instructor to Course Screen

This screen allows the admin to assign or unassign instructors to courses.
It displays current instructor–course assignments and provides controlled options for modification.

All assignment operations are validated by the database to ensure that:

- Only admins can perform assignment changes
- Instructors are assigned only to valid courses
- Data integrity is preserved

This screen demonstrates secure **relationship management** between academic entities.



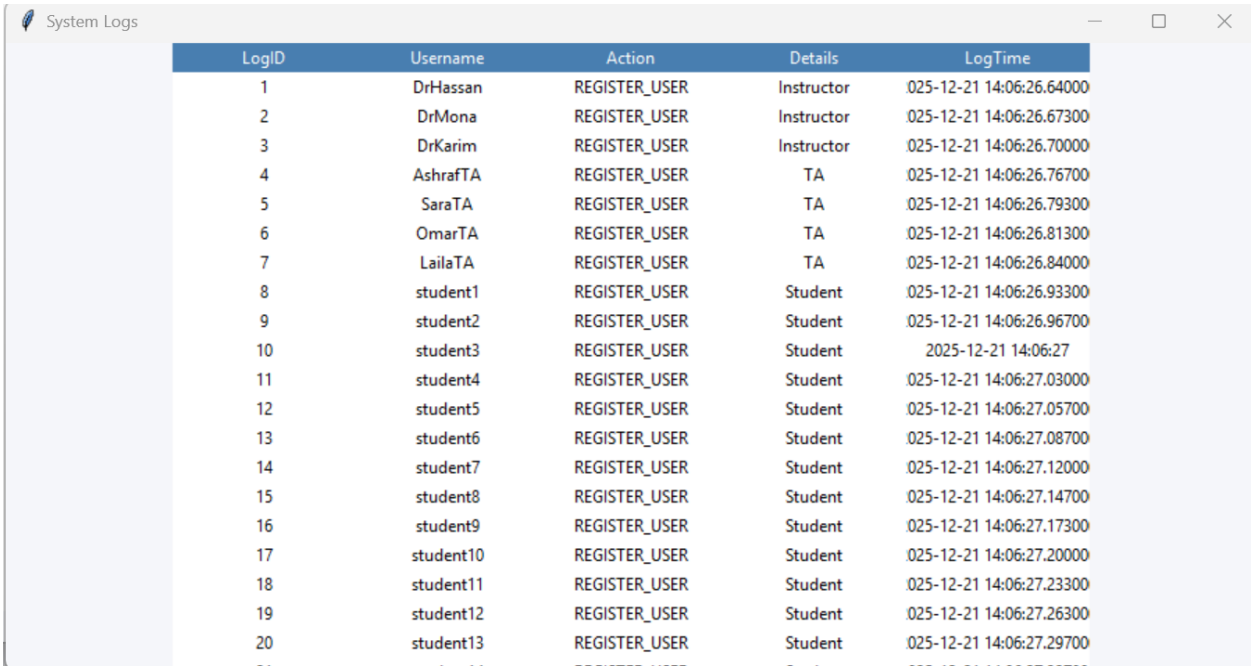
7.5 System Audit Logs (Read-Only)

Figure 7.5: System Audit Logs Screen

This screen displays the system's audit logs, including:

- Username
- Action performed
- Action details
- Timestamp

The logs are generated automatically for sensitive operations such as user registration, role changes, and assignments.
This view is **read-only** and accessible only to the admin, ensuring **accountability, traceability, and non-repudiation**.



LogID	Username	Action	Details	LogTime
1	DrHassan	REGISTER_USER	Instructor	025-12-21 14:06:26.64000
2	DrMona	REGISTER_USER	Instructor	025-12-21 14:06:26.67300
3	DrKarim	REGISTER_USER	Instructor	025-12-21 14:06:26.70000
4	AshrafTA	REGISTER_USER	TA	025-12-21 14:06:26.76700
5	SaraTA	REGISTER_USER	TA	025-12-21 14:06:26.79300
6	OmarTA	REGISTER_USER	TA	025-12-21 14:06:26.81300
7	LailaTA	REGISTER_USER	TA	025-12-21 14:06:26.84000
8	student1	REGISTER_USER	Student	025-12-21 14:06:26.93300
9	student2	REGISTER_USER	Student	025-12-21 14:06:26.96700
10	student3	REGISTER_USER	Student	2025-12-21 14:06:27
11	student4	REGISTER_USER	Student	025-12-21 14:06:27.03000
12	student5	REGISTER_USER	Student	025-12-21 14:06:27.05700
13	student6	REGISTER_USER	Student	025-12-21 14:06:27.08700
14	student7	REGISTER_USER	Student	025-12-21 14:06:27.12000
15	student8	REGISTER_USER	Student	025-12-21 14:06:27.14700
16	student9	REGISTER_USER	Student	025-12-21 14:06:27.17300
17	student10	REGISTER_USER	Student	025-12-21 14:06:27.20000
18	student11	REGISTER_USER	Student	025-12-21 14:06:27.23300
19	student12	REGISTER_USER	Student	025-12-21 14:06:27.26300
20	student13	REGISTER_USER	Student	025-12-21 14:06:27.29700

7.6 Summary of Screenshot Section

The screenshots presented in this section demonstrate that:

- Access to system functionality is strictly controlled by user roles
- Sensitive operations are protected by database-level security
- The GUI acts only as a presentation layer
- All critical actions are audited and logged

Together, these screenshots validate the correctness and security of the SRMS system design.