**Security risks**

| **Risk ID** | **Description** | **Likelihood** | **Impact** | **Mitigation** |
| --- | --- | --- | --- | --- |
| S-1 | SQL Injection from unsanitized JDBC queries exposing sensitive student and financial data | High | High | Use prepared statements, input validation, and encryption |
| S-2 | Weak MySQL configurations (default credentials, poor access control, brute force vulnerability) | High | High | Strengthen database security, enforce strong credentials, enable intrusion monitoring |
| S-3 | Java Swing risks (weak encryption, hardcoded credentials, memory dump leaks) | Medium | High | Use secure encryption libraries, avoid hardcoded credentials |
| S-4 | E-learning system threats (phishing, DDoS, breaches, misuse of PII and academic/financial data) | High | High | Continuous monitoring, security training, layered defenses |
| S-5 | XAMPP risks (default passwords, open ports, exposed testing environments) | Medium | High | Secure configuration, change defaults, restrict access |

**Methodological risks (Waterfall)**

| **Risk ID** | **Description** | **Likelihood** | **Impact** | **Mitigation** |
| --- | --- | --- | --- | --- |
| M-1 | Requirements changes after initial phase causing rework and cost overrun | High | Medium | Use Agile increments with integrated feedback |
| M-2 | Late defect discovery in JAR files, leaving deprecated features unchecked | Medium | High | Continuous testing and code review |
| M-3 | Minimal end-user feedback, leading to features not aligned with user needs | Medium | Medium | Gather feedback early and iteratively |
| M-4 | Rigid timelines causing cascading delays if one team is late | High | Medium | Build buffer times, adopt Agile sprints |

**All technical risks**

| **Risk ID** | **Description** | **Likelihood** | **Impact** | **Mitigation** |
| --- | --- | --- | --- | --- |
| T-1 | Poorly handled JDBC connection strings exposing databases | High | High | Secure JDBC handling with strict encryption |
| T-2 | Sensitive data in JAR files exposed due to poor coding/configuration | Medium | High | Secure coding practices, code reviews |
| T-3 | Complexity from integrating Swing, MySQL, and XAMPP increases failure points | Medium | Medium | Modular design, testing integration points |

**Data Protection Risks**

| **Risk ID** | **Description** | **Likelihood** | **Impact** | **Mitigation** |
| --- | --- | --- | --- | --- |
| D-1 | Financial data exposure (payment details) causing compliance and liability issues | Medium | High | Strong encryption, compliance frameworks |
| D-2 | Student privacy risks (PII, academic info leaks) | High | High | Enforce privacy policies, access control |
| D-3 | Backup/recovery gaps leading to permanent data loss | Medium | High | Regular automated backups, tested recovery plans |

**Operational Risks**

| **Risk ID** | **Description** | **Likelihood** | **Impact** | **Mitigation** |
| --- | --- | --- | --- | --- |
| O-1 | Local deployment with single point of failure, poor scalability | Medium | High | Plan scalable XAMPP/cloud deployment |
| O-2 | Maintenance complexity from 8 developers working in silos | Medium | Medium | Enforce coding standards, regular sync-ups |
| O-3 | Dependence on specific software versions causing failures | Medium | Medium | Version control, environment standardization |