**Risk Analysis for Student Management System**

Risk analysis involves identifying potential threats that could affect the system’s performance, security, and usability. Below are key risks associated with the Student Management System:

1. **Security Breach** – Unauthorized access to student, teacher, or admin data can lead to privacy violations. This can occur due to weak authentication, improper role-based access control (RBAC), or vulnerabilities in data transmission and storage.
2. **Data Loss** – Accidental deletion, database corruption, or hardware failures can result in the loss of critical student records, attendance logs, and grades. Without proper backups and recovery mechanisms, data retrieval could be difficult.
3. **Performance Issues** – As the number of users increases, the system may experience slow response times, particularly when fetching large student records or generating reports. Inefficient database queries, lack of caching, and inadequate server resources can contribute to this problem.
4. **Version Control Issues** – Mistakes in Git usage, such as improper commits, merge conflicts, or accidental deletions, may lead to code loss or unstable versions of the system. If not managed properly, this could delay development and introduce bugs.
5. **UI/UX Problems** – A poorly designed user interface or complex navigation can make it difficult for students, teachers, and admins to use the system effectively. Lack of user feedback and usability testing may result in frustration and reduced adoption.
6. **Server Downtime** – Hosting issues, server crashes, or unexpected traffic spikes can lead to system downtime, making it inaccessible to users. This could disrupt important operations like attendance marking, grade submissions, and announcements.

By identifying these risks early, proper mitigation strategies can be implemented to ensure the system remains secure, efficient, and user-friendly.

**Risk Table**

Student Management System

| **Risk** | **Probability (P)** | **Impact (C)** | **Risk Exposure (RE = P × C)** | | | | **RMMM (Mitigation, Monitoring, Management)** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Security Breach** (Unauthorized access to student data) | | | | 0.7 | 5 | 3.5 | **Mitigation:** Implement strong authentication, RBAC, encryption.  **Monitoring:** Log access attempts, detect anomalies. **Management:** Have a rollback mechanism and security patch plans. |
| **Data Loss** (Accidental deletion of student records) | | | | 0.5 | 5 | 2.5 | **Mitigation:** Regular database backups, version control. **Monitoring:** Monitor database change logs. **Management:** Implement data recovery procedures. |
| **Performance Issues** (Slow response time for student records) | | | | 0.6 | 3 | 1.8 | **Mitigation:** Optimize database queries, caching, indexing. **Monitoring:** Track system load and response times. **Management:** Implement auto-scaling or caching layers. |
| **Version Control Issues** (Mistakes in Git leading to loss of code) | | | | 0.4 | 3 | 1.2 | **Mitigation:** Use branches, commit frequently, code reviews. **Monitoring:** Monitor commit history and backups. **Management:** Restore from last working commit, document recovery steps. |
| **UI/UX Issues** (Poor usability for students/teachers) | | | | 0.5 | 2 | 1.0 | **Mitigation:** Conduct usability testing, feedback sessions. **Monitoring:** Track user complaints, conduct surveys. **Management:** Deploy UI/UX improvements incrementally. |
| **Server Downtime** (System becomes unavailable due to hosting issues) | | | | 0.4 | 4 | 1.6 | **Mitigation:** Use reliable cloud hosting, load balancing, and failover servers. **Monitoring:** Set up uptime monitoring and alerts. **Management:** Have a backup server or switch providers if necessary. |