Software Engineering- Risk Analysis

# Project Topic (e.g. Online Grocery Store Management System)

Group Members

|  |  |  |
| --- | --- | --- |
| Member Name | Member Surname | Student ID |
| Alican | Kaya | 49429 |
| Ege Bora | Güzel | 58351 |
| İrfan | Bag | 49279 |
| Cenker | Sule | 58566 |
| Efe Berk | Tancı | 59115 |
| Berke | Turk | 58169 |

# Introduction

This Risk Analysis document focuses specifically on the Hotel Booking and Management Platform—an online system designed to streamline hotel reservations, guest payments, and administrative operations. The goal of this analysis is to identify, evaluate, and prepare for potential threats that could jeopardize project objectives, user satisfaction, and operational stability. Covering technical, operational, financial, and regulatory risk categories, this document establishes a clear methodology for assessing probability and impact, and defines mitigation and contingency plans. Ongoing monitoring triggers are included to ensure risks are revisited throughout development and deployment.

## Risk 1 – Payment Gateway Outage

**Description:** The third-party payment gateway may experience downtime or service interruptions, preventing guests from completing transactions.

**Category:** External

**Likehood**: Dependence on an external service with occasional outages.

**Impact**: Inability to process payments will halt revenue flow and erode user trust.

**Mitigation plan**: Integrate at least two payment providers and implement automatic fallback to a secondary gateway.

**Contingency plan**: If the primary gateway fails, reroute transactions to an alternate provider (e.g., Stripe if PayPal is down).

**Monitoring approach**: Set up uptime monitoring and alerts on transaction success rates via an external monitoring service.

**Review trigger**: Trigger a risk reassessment if transaction success falls below 98% within any 24-hour window.

## Risk 2– SQL Injection Vulnerability

**Description:** An attacker could exploit insecure database queries to perform SQL injection, leading to data breaches or corruption.

**Category:** Technical

**Likehood**: Medium – Risk arises if developers use unparameterized queries or insufficient input validation.

**Impact**: High – Could compromise sensitive user data and damage system integrity.

**Mitigation plan**: Enforce parameterized queries and prepared statements for all database operations; conduct regular code reviews and developer training.

**Contingency plan**: Immediately patch any vulnerable endpoints, rotate database credentials, and invalidate compromised sessions.

**Monitoring approach**: erform weekly automated security scans and monthly penetration tests.

**Review trigger**: Reassess risk upon detection of any SQL injection issue during a security audit or in production logs.

## Risk 3– Server Infrastructure Failure

**Description:** The application server or database server may suffer hardware failure or configuration errors, causing service downtime.

**Category:** Operational

**Likehood**: Servers can fail due to hardware faults or misconfiguration.

**Impact**: High – Downtime impacts all users and revenue streams.

**Mitigation plan**: Employ redundant server instances behind a load balancer and maintain automated backups.

**Contingency plan**: Activate a hot standby server or failover to a backup environment.

**Monitoring approach**: Use infrastructure monitoring tools (e.g., CloudWatch, Nagios) with alert thresholds.

**Review trigger**: Trigger reassessment after any unplanned outage exceeding 5 minutes.

## Risk 4 – Team Skill Gap

**Description:** The development team lacks experience with certain required technologies (e.g., advanced PHP features or payment integrations).

**Category:** Technical

**Likehood**: Medium – Newer team members may be unfamiliar with the full tech stack.

**Impact**: Medium – Could lead to coding delays and increased bug rates.

**Mitigation plan**: Schedule focused training sessions, pair programming, and bring in an external consultant if needed.

**Contingency plan**: Reassign complex tasks to more experienced developers or adjust the project timeline.

**Monitoring approach**: Track progress against sprint goals and conduct weekly skill assessments.

**Review trigger**: Reevaluate risk if two consecutive sprints miss key deliverables due to technical issues.

## Risk 5 – Budget Overrun

**Description:** Project expenses may exceed the allocated budget due to scope creep or unexpected costs.

**Category:** Financial

**Likehood**: Low – Budget is well-defined but small changes can accumulate.

**Impact**: Medium – May require scope reduction or additional funding.

**Mitigation plan**: Implement monthly expense reviews, enforce change control procedures, and prioritize features.

**Contingency plan**: Reduce non-critical features or request incremental budget from stakeholders.

**Monitoring approach**: Maintain a budget tracking spreadsheet with variance reports each month.

**Review trigger**: Trigger review if expenditure exceeds 75% of allocated budget at the project midpoint.

## Risk 6– Regulatory Compliance Changes

**Description:** Changes in data protection regulations (e.g., GDPR, KVKK) may require application updates.

**Category:** External

**Likehood**: Low – Regulations change infrequently but carry strict requirements.

**Impact**: Medium – Non-compliance can result in legal penalties and service disruption.

**Mitigation plan**: Subscribe to regulatory bulletins, conduct quarterly compliance audits, and design the system with configurable privacy settings.

**Contingency plan**: Engage legal counsel to quickly implement mandated changes and temporarily disable non-compliant features.

**Monitoring approach**: Track regulatory updates and compliance status in a centralized risk log.

**Review trigger**: Reassess risk upon publication of relevant regulatory amendments.

# Evaluation Criteria

**1. Risk Identification (2 points)**

* Risk List (1 point): Identify at least six specific risks that could affect the project (e.g., technical difficulties, limited resources, or scope changes). Risks should be relevant to the project and avoid vague descriptions.
* Risk Categories (1 point): Assign each risk into a category (e.g., technical, operational, financial, external). This shows an understanding of different risk types and their origins.

**2. Risk Assessment (2 points)**

* Likelihood Assessment (1 point): Assign a likelihood level (e.g., low, medium, high) to each risk and explain why (e.g., "High likelihood because the team lacks experience with this tool").
* Impact Assessment (1 point): Define the impact level (e.g., low, medium, high) for each risk and describe its potential effect (e.g., "High impact as it could delay the project by weeks").

**3. Risk Mitigation Strategies (2 points)**

* Mitigation Plan (1.5 points): Suggest at least one practical mitigation strategy per risk (e.g., "Train the team on the new software" or "Schedule extra testing phases"). Strategies must be specific, realistic, and linked to the risks.
* Contingency Plan (0.5 points): For at least one high-impact risk, include a backup plan (e.g., "If the vendor delays delivery, source materials from another supplier"). This shows readiness for critical situations.

**4. Risk Monitoring and Review (1 point)**

* Monitoring Approach (0.5 points): Explain how risks will be tracked during the project (e.g., "Weekly reviews in team meetings" or "Maintain an updated risk log").
* Review Triggers (0.5 points): Specify at least one event that will trigger a reassessment of risks (e.g., "If a key team member leaves" or "If the budget is cut").

**5. Timely Submission (1 point)**

* Submit the document by the deadline to earn this point. Late submissions, even if high-quality, will not receive this point.

**Additional Guidelines**

* **Clarity and Specificity**: Risks and strategies must relate directly to the project. Generic statements (e.g., "Something might go wrong") will lose points.
* **Consistency**: Mitigation and monitoring plans should match the risks’ likelihood and impact. Inconsistent plans (e.g., over-preparing for a minor risk) will reduce the score.
* **Document Quality**: While not scored separately, the document should be well-organized and easy to read. Poor structure or unclear wording may affect the evaluation.