EAST WEST UNIVERSITY

Department of Computer Science & Engineering

Risk Analysis Assignment

Course Title: Information System Analysis and Design

Course Code: CSE 347

Section No: 02

Submitted To:

Dr. Shamim H Ripon

Professor

Department of Computer Science and Engineering

Submitted By:

Group-10

Student Name	Student ID
Md Saiful Islam	2022-3-60-045
Umme Mukaddisa	2022-3-60-317
Shanghita Naha Sristy	2022-3-60-311
Ayon Adhikary	2022-3-60-137

Abstract

This report presents a comprehensive risk analysis of a cross-platform Online Auction System designed for real-time bidding, auction management, and secure user interaction. Given the system's dependence on time-sensitive transactions, secure communications, and multiple user roles (bidders, admins), ensuring system reliability and security is essential. The analysis is conducted in two phases: a Risk Assessment Matrix identifies and prioritizes five major risks. In contrast, a Failure Modes and Effects Analysis (FMEA) provides deeper insights into the two most critical ones. Suggested mitigation strategies aim to ensure integrity, user trust, and smooth auction operations.

Introduction

An online auction platform must function seamlessly to ensure fair competition, accurate bidding, and data security. This system includes features such as live auctions, real-time bidding, private rooms, push notifications, and admin approval mechanisms. However, issues such as inconsistent auction timing, simultaneous bid conflicts, and security vulnerabilities pose significant challenges. Identifying and mitigating these risks is critical before deployment to maintain credibility and performance during live bidding events. This analysis uses two methodologies:

1. Risk Assessment Matrix – To prioritize risks based on likelihood and impact.
2. Failure Modes and Effects Analysis (FMEA) – To evaluate the top two risks in depth.

Part A: Risk Assessment Matrix

Five key risks were identified and scored using Likelihood (L) and Impact (I) on a scale of 1–5.
Risk Score = L × I.

#	Risk	Description	Likelihood	Impact	Score	Justification
1	Auction Timer Failure	Inconsistent end times across devices	4	5	**20**	Affects auction fairness and trust
2	Bid Fraud / Race Condition	Incorrect winner due to simultaneous bids	3	5	**15**	Revenue loss and potential legal issues
3	API/Security Vulnerabilities	Exposure of API keys or insecure backend	2	5	10	High potential for breaches if unprotected
4	Lack of User Verification	No two-factor or identity verification	3	3	**9**	Risk of fake accounts, bid manipulation
5	Push Notification Failure	Delay due to dependence on admin activity	3	2	6	Users may miss real-time updates

Top Two Prioritized Risks

1. Auction Timer Failure (Score: 20)
2. Bid Fraud / Race Conditions (Score: 15)

Part B: Failure Modes and Effects Analysis (FMEA)

Detailed analysis of the two highest risks using Severity (S), Occurrence (O), and Detection (D) on a scale of 1–10.

Risk Priority Number (RPN) = S × O × D.

Failure Mode	Effect	Cause	Severity (S)	Occurrence (O)	Detection (D)	RPN (S×O×D)	Action Plan
Auction Timer Inconsistency	Users may see different auction end times, leading to bid disputes	Local timer used on each device	9	4	3	**108**	Use a cloud-based global timer endpoint to sync across clients.
Simultaneous Bidding Error	An incorrect auction winner may be declared	Multiple identical timestamp bids	8	4	3	**96**	Queue server-side bid submissions with strict timestamp validation and conflict resolution logic.

Explanation of Top Mitigation Strategies

1. Auction Timer Failure

Problem:
Users viewing different countdown times due to reliance on local device clocks can lead to unfair bidding, user disputes, and reputational damage.

Solution:

- **Centralized Timer via Cloud Function:**
  Host the countdown logic on a **centralized cloud server** (e.g., Firebase Cloud Functions or AWS Lambda). This ensures that all users are referencing a **single, authoritative time source** rather than device-local clocks, eliminating desynchronization.

- **WebSocket Synchronization:**
  Integrate **WebSocket-based real-time updates** to maintain a consistently synchronized countdown across all connected users. This ensures that every second is visible in real time, minimizing delay or lag during the critical final moments of bidding.

Impact:

- Guarantees fairness and consistency.

- Prevents disputes over end times.

- Reinforces trust in the auction process.

2. Bid Fraud / Race Conditions

Problem:
When multiple users submit bids at the same moment, without proper handling, the system may register the wrong winner due to concurrency issues or a lack of order enforcement.

Solution:

- **Atomic Bid Handling with Server-Side Queuing:**
  Bids are processed using **atomic transactions** that ensure only one bid is handled at a time. This prevents concurrency conflicts and ensures that bids are **queued and processed sequentially** based on exact submission time.

- **Millisecond-Precision Timestamping:**
  Each bid is assigned a timestamp accurate to the **millisecond level**. If multiple bids are submitted within the same second, the system uses this fine-grained timing to determine priority.

- **Transactional Locking:**
  Implement **locking mechanisms** at the database or application level to enforce **exclusive**

access to the bid update operation during processing. This avoids data races and ensures bid consistency.

Impact:

- Prevents incorrect winner declarations.
- Maintains integrity during high-traffic bidding moments.
- Reduces the risk of fraud or disputes due to simultaneous bidding.

Conclusion

The risk analysis conducted for this Online Auction System highlights key vulnerabilities that could compromise both functionality and user trust. In particular, the decentralized auction timer and simultaneous bidding conflicts are critical issues that must be addressed prior to launch. Through centralized cloud timing, robust bid conflict resolution, and enhanced authentication and notification mechanisms, the system can achieve reliability, fairness, and security. This structured approach to risk management ensures a more robust deployment and sets the foundation for continuous improvement and user satisfaction in future iterations.