EAST WEST UNIVERSITY
Department of Computer Science & Engineering

Project Requirement and Planning

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

Table of Contents

1. Introduction

The Online Auction System project aims to create a secure and efficient platform for real-time online auctions. It enables users to list products, participate in live bidding, and complete transactions seamlessly. Key features include real-time bid tracking, automatic bidding, secure payment integration, and administrative controls, ensuring a fair and user-friendly experience. This document outlines the Project Requirements and Planning phase, detailing the project scope, objectives, stakeholders, and requirements. It also covers resource management, risk mitigation, quality assurance, and communication strategies. By establishing a clear roadmap, this plan ensures the project progresses efficiently.

2. Project Overview

2.1. Project Title and Description

Title: Online Auction System.

The Online Auction System is a cross-platform application that allows users to participate in real-time product auctions. It provides a secure, efficient, and user-friendly platform where users can list items for auction, and others can place bids.

Users can register and authenticate securely, submit product listings with descriptions and images, and choose auction durations. Admins review and approve listings before they go live. The system supports both English auctions, where bids increase incrementally, and sealed-bid auctions, where bids are submitted privately.

Key features include real-time bidding, automatic bidding with a maximum bid limit, and live countdowns to track auction progress. Once an auction ends, the system notifies the winning bidder and seller. Secure transactions are ensured through Bkash payment integration.

An admin panel allows authorized personnel to manage auctions, monitor user activity, and resolve disputes. The system also includes reporting and analytics tools for tracking auction performance and user activity.

Overall, the Online Auction System delivers a fair, secure, and seamless auction experience. With advanced features and strong security measures, it fosters user trust and provides an efficient platform for competitive bidding.

2.2. Scopes of the Project

User Management

- Secure user registration and authentication.
- Role-based access control for buyers, sellers, and administrators.

Auction Management

- Support for English auctions (incremental bidding) and sealed-bid auctions (private bidding).
- Real-time bidding with automatic bid tracking and maximum bid automation.
- Product listings with images, descriptions, starting bids, and auction durations.
- Admin review and approval process for auction listings.

Payment and Transaction Handling

- Integration with Bkash for secure and seamless payment processing.

Notifications and Alerts

- Real-time notifications for bid updates, auction results, and payment confirmations.

Administrative Controls

- A comprehensive admin panel for managing users, auctions, and dispute resolution.

- Monitoring tools to oversee platform activity and enforce policy compliance.

Reporting and Analytics

- Auction performance tracking and user activity reports for administrators.
- Bidding history and spending reports for users to track their transactions.

2.3. Boundaries of the project

Auction Delivery and Logistics

- The system **does not** handle physical delivery or item shipping logistics.

Third-Party Integrations

- No integration with external marketplaces or e-commerce platforms.

Advanced Machine Learning

- No AI-driven predictive analytics or automated auction optimization features.

User-Generated Reviews

- The platform does not include a user review or rating system for sellers or products.

System Security

- Data encryption and secure communication protocols.
- Fraud detection and prevention mechanisms to ensure fair participation.

2.4. Purpose and Objectives of the Project

The purpose of the Online Auction System is to provide a secure, efficient, and user-friendly platform for conducting live and sealed-bid auctions. It allows sellers to list products and buyers to place competitive bids in real-time while ensuring transparency and fairness through automated bid tracking and accurate winner selection. The system integrates secure payment processing via Bkash and employs robust security measures to protect user data and transactions. Additionally, it offers an admin panel for managing auctions, monitoring user activity, and resolving disputes. With comprehensive reporting and analytics tools, the system enables users to track their bidding history and allows administrators to analyze auction performance, ensuring a reliable and seamless auction experience. The objectives of the project are:

- **Enable Secure User Participation** – Implement robust authentication and role-based access control to protect user data and ensure a safe auction environment.
- **Facilitate Real-Time Auctions** – Support English and sealed-bid auctions with automatic bid tracking, live countdowns, and instant notifications.
- **Ensure Fair and Transparent Bidding** – Provide clear auction rules, prevent fraudulent activities, and maintain a reliable bidding process.
- **Integrate Secure Payment Processing** – Utilize Bkash payment integration with an escrow system to safeguard transactions and minimize financial risks.
- **Enhance User Engagement** – Deliver a responsive, intuitive interface with live dashboards, bid history tracking, and auction insights.
- **Empower Administrative Control** – Equip administrators with tools to monitor user activities, review auction listings, and resolve disputes effectively.
- **Offer Data-Driven Insights** – Provide reporting and analytics for both users and admins to track bidding performance, spending, and auction trends.
- **Ensure Scalability and Reliability** – Build a scalable system that can handle high user traffic while maintaining system stability and performance.

3. Stakeholders and Actors

3.1. Key Stakeholders

Faculty (Course Instructor)

The **Faculty** is a key stakeholder responsible for evaluating the project based on the assignment criteria. They provide essential guidance, feedback, and assessment throughout the project lifecycle. Their role is to ensure that the project meets academic and technical standards while supporting the team in understanding and implementing the project requirements effectively.

Teaching Assistant (TA)

The **Teaching Assistant (TA)** acts as an intermediary between the project team and the faculty, offering regular feedback and technical support. They provide administrative oversight by monitoring project progress, clarifying requirements, and assisting with complex technical challenges. The TA ensures the team stays on track and that the project aligns with the faculty's expectations and academic guidelines.

Project Team Members

The **Project Team Members** are responsible for the complete design, development, testing, and delivery of the Online Auction System. Their primary responsibility is to ensure the project meets specified functional and technical requirements while adhering to deadlines. They collaborate to implement system features, resolve issues, and deliver a functional and well-documented project.

End Users (Hypothetical Users and Admins)

The **End Users** represent the intended audience of the Online Auction System, including buyers, sellers, and administrators. Although hypothetical for this course project, their needs shape the system's functionality, including user registration, auction participation, and payment processes. Designing the system with these users in mind ensures a practical and user-centered solution.

Peers (Other Students)

Peers (Other Students) play a supporting role in the project by providing informal feedback during peer reviews and class presentations. Their insights help the team identify potential improvements and ensure the project meets both technical and presentation standards. Peer feedback also promotes collaboration and knowledge sharing, contributing to the project's overall quality.

3.2. Roles and Responsibilities

Faculty (Course Instructor)

Role: Evaluator and mentor for the project.

Responsibilities:

- Provide guidance on project scope, requirements, and technical implementation.
- Review project milestones and offer feedback for improvement.
- Assess the final project based on specified evaluation criteria.
- Ensure the project meets academic and technical standards.

Teaching Assistant (TA)

Role: The **Teaching Assistant (TA)** acts as a bridge between the project team and the faculty, providing regular feedback, technical support, and administrative oversight throughout the project lifecycle.

Responsibilities:

- Offer technical and procedural assistance to ensure the project meets the required standards.
- Evaluate project progress during milestones and provide constructive feedback for improvement.

- Track team performance and report key issues or delays to the faculty.
- Ensure the project aligns with academic guidelines and technical specifications.
- Communicate faculty feedback to the team and suggest actionable improvements.

Project Team Members

Role: Designers and developers of the Online Auction System.

Responsibilities:

- Gather and document project requirements.
- Design, develop, and implement system features (e.g., user authentication, auctions, payments).
- Conduct testing and debugging to ensure functionality and performance.
- Collaborate effectively to meet deadlines and deliverables.
- Prepare project documentation and reports for submission.

End Users (Hypothetical Users, and Admins)

Role: Intended users of the system (for demonstration purposes).

Responsibilities:

- Represent the target audience for whom the system is designed.
- Define user needs (e.g., bidding functionality, payment security).
- Influence system usability, accessibility, and user experience.

Peers (Other Students)

Role: External reviewers and support network.

Responsibilities:

- Provide feedback during peer reviews or presentations.
- Share knowledge and offer technical or conceptual support.
- Compare project progress with similar groups for benchmarking.

4. Project Requirements

4.1. Functional Requirements

User Management

- Secure user registration and authentication (email, phone verification, or social login).
- Role-based access for buyers, sellers, and administrators.
- Profile management (update user details, view transaction history, change passwords).

Auction Management

- Sellers can submit product details (title, description, images, starting bid, auction duration).
- Admins review and approve auctions before they go live.
- Supports English auctions (incremental bidding) and sealed-bid auctions (private bidding).
- Automatic bid tracking, maximum bid automation, and live countdown timers.
- The system automatically determines the winner and notifies relevant users.

Payment and Transaction Handling

- Bkash integration for secure transactions.
- Transaction history for users to track payments and winnings.

Notifications and Alerts

- Real-time bid updates, auction results, and payment confirmations via email, SMS, or push notifications.

Administrative Controls

- Admin dashboard for managing auctions, users, and system performance.
- Dispute resolution system for handling complaints and policy violations.

- Monitoring tools to detect fraudulent activities and enforce compliance.

Reporting and Analytics

- Auction performance reports for administrators.
- User activity tracking (bidding history, spending reports).
- System usage insights for platform optimization.

Security Features

- Secure login with encrypted passwords and optional multi-factor authentication (MFA).
- Role-Based Access Control to restrict actions based on user roles (user, admin).
- Access restrictions ensure users can only perform actions relevant to their role.
- Secure payment through Bkash.
- Admin oversight to monitor and manage auctions, users, and system activities.

4.2. Non-Functional Requirements

Performance Requirements

- **System uptime:** 99.9% availability to handle real-time auctions without disruptions.
- **Scalability:** Ability to support high user traffic during peak bidding hours.

Security Requirements

- SSL encryption for data protection.
- Two-factor authentication (2FA) for enhanced account security.
- Access control measures to prevent unauthorized system access.

Usability Requirements

- Intuitive user interface with a seamless experience across web and mobile.
- Responsive design for compatibility with different screen sizes.

4.3. Technical Requirements

Frontend Technologies

- **Flutter and Dart** for building a cross-platform, dynamic, and responsive user interface.

Backend Technologies

- **Firebase** for handling business logic, real-time updates, and database operations.

Database and Storage

- **Firebase Firestore** for managing user and auction data.
- **Supabase** for storing product images and other media files.

Hosting and Deployment

- **Firebase Hosting** for deploying and managing the auction system.

Third-Party Integrations

- **Bkash API** for secure payment processing.
- **Firebase Authentication** for user registration and login.
- **Firebase Cloud Messaging (FCM)** for sending real-time notifications.

4.4. Business Requirements

Monetization Model

- **Commission-based** revenue (percentage from each completed auction).
- **Private listings** for sellers to highlight their auctions.
- Paid private room auction system.

User Policies

- **Strict auction rules** to prevent manipulation or unfair practices.
- **User verification system** to ensure genuine participation.

Support and Maintenance

- **Customer support module** for handling inquiries and disputes.
- **Regular software updates** to fix bugs and introduce new features.

5. Resource Management

5.1. Team Members and Their Roles

Md Saiful Islam

Role: Project Manager, Full-Stack Developer
Responsibilities:

- Oversee project planning, execution, and delivery.
- Manage team coordination and ensure timely completion of milestones.
- Develop and maintain both frontend (Flutter) and backend (Firebase) systems.
- Ensure seamless integration of real-time features and third-party services (e.g., Bkash API).

Umme Mukaddisa

Role: Frontend Developer, UI/UX Designer, Quality Assurance
Responsibilities:

- Design and implement user-friendly interfaces using Flutter.
- Create wireframes and prototypes to enhance user experience.
- Conduct unit, integration, and system testing to ensure software quality.
- Identify and resolve UI/UX issues to improve user engagement.

Shanghita Naha Sristy

Role: Frontend Developer, UI/UX Designer, Quality Assurance
Responsibilities:

- Collaborate on designing and developing intuitive user interfaces.
- Ensure consistent and responsive design across devices.
- Perform thorough testing and debugging to maintain code quality.
- Assist in gathering user feedback for continuous UI/UX improvement.

Ayon Adhikary

Role: Project Scheduler, Security Specialist, Documentation Specialist
Responsibilities:

- Develop and manage the project schedule, tracking key milestones.
- Implement security measures, including authentication and data protection.
- Conduct security audits and ensure compliance with standards.
- Create and maintain comprehensive project documentation and technical guides.

5.2. Tools, Technologies, and Resources

Development Tools

- **Flutter & Dart:** For building a cross-platform user interface (Android, Desktop, and web).
- **Firebase:** For backend services, including real-time database, authentication, and cloud functions.
- **Supabase:** For file and media storage (e.g., product images).
- **Bkash API:** For secure payment integration and processing.

Integrated Services

- **Firebase Authentication:** For secure user registration and login.
- **Firebase Cloud Messaging (FCM):** For real-time notifications and auction updates.
- **Firestore Database:** For managing and storing user, auction, and bid data.

Development and Collaboration Tools

- **Android Studio:** Primary IDE for Flutter and Firebase development.
- **Git & GitHub:** For version control, collaboration, and managing code repositories.

Testing and Debugging Tools

- **Flutter DevTools:** For profiling, debugging, and analyzing app performance.

Hosting and Deployment

- **Firebase Hosting:** For deploying the online auction system and managing production environments.

Project Management and Communication

- **Github:** For task management, tracking progress, and organizing project milestones.
- **Discord:** For team communication and quick collaboration.
- **FB Messenger**: Group chat for communication and discussion.

6. Risk Management

6.1. Potential Risks

Security Vulnerabilities

- **Incomplete Risk Identification:** Overlooking threats like auction fraud, bot bidding, or unauthorized access.
- **Weak Security Measures:** Ineffective encryption, authentication loopholes, or lack of fraud detection mechanisms.
- **Delayed Response to Threats:** Slow detection of cyber threats like DDoS attacks or hacking attempts.
- **Compliance Risks:** Failure to meet data privacy laws or Bkash payment security requirements.
- **Reverse Engineering:** Reverse engineering the application and extract data.

Delays in Development

- **Overcomplicated Risk Processes:** Excessive security reviews or manual approvals slowing feature releases.
- **Underestimated Development Risks:** Poor planning causing delays in real-time bidding and automation features.
- **Scope Creep:** Constant risk assessments leading to frequent design changes and missed deadlines.
- **Slow Dispute Resolution Mechanism:** Inefficient handling of auction disputes affecting user trust.

Resource Unavailability

- **Shortage of Skilled Experts:** Lack of cybersecurity specialists or developers to handle risk mitigation.
- **Infrastructure Issues:** Server downtimes or lack of backup systems affecting auction operations.
- **Dependence on Third-Party Services:** Reliance on external security tools in Firebase, which may fail or be unavailable.

Payment Gateway Integration Issues

- **Security Compliance Challenges:** Difficulty meeting PCI-DSS and Bkash security standards.
- **Transaction Failures:** Poor integration leading to unsuccessful payments or delayed fund transfers.
- **Fraudulent Transactions:** Users exploiting loopholes to manipulate payments or refunds.
- **Over-Reliance on a Single Provider:** If Bkash faces downtime, all transactions halt without alternatives.

6.2. Risk Mitigation Strategies

- **Security Vulnerabilities** – Implement robust encryption to protect user data. Conduct regular security audits and ensure compliance with industry standards like PCI-DSS. Enforce multi-factor authentication (MFA) and role-based access control to prevent unauthorized access.
- **Delays in Development** – Use Agile methodology for iterative progress and faster risk resolution. Prioritize critical features, streamline approval processes, and automate testing and deployment to minimize delays.

- **Resource Unavailability** – Identify backup personnel for key roles and cross-train team members to avoid skill shortages. Maintain alternative service providers for hosting, fraud detection, and security audits. Optimize cloud infrastructure to ensure scalability.
- **Payment Gateway Issues** – Test Bkash integration in a sandbox environment before deployment to prevent transaction failures. Implement fallback payment options and monitor transactions in real-time. Use tokenization for secure payment processing and fraud prevention.

6.3. Risk Assessment Matrix

Risk	Likelihood	Impact	Mitigation Strategy
Security vulnerabilities	High	High	Implement encryption and regular audits
Development delays	Medium	Medium	Agile methodology, regular check-ins
Resource unavailability	Low	Medium	Cross-training, maintaining backup resources
Payment integration issues	Low	High	Thorough testing in a sandbox environment

7. Quality Management

7.1. Quality Assurance Processes

Testing Framework

- Use Flutter test libraries for unit, widget, and integration tests.
- Implement Firebase Emulator Suite for local backend testing.

Code Reviews and Peer Testing

- Conduct regular code reviews to ensure high-quality, maintainable code for all new features.
- Perform peer testing on critical modules (e.g., payment integration, bidding logic) to catch issues early and improve collaboration among team members.

Automated and Manual Testing

- Automated Testing: Develop unit tests for core functionalities and integration tests to ensure seamless interaction between modules. Use performance testing tools to simulate high user loads and validate system stability.
- Manual Testing: Conduct user acceptance testing to ensure the platform meets user expectations. Perform security testing to identify vulnerabilities and test edge cases to ensure robustness.

Process Definition

- Conduct code reviews for all major features before merging into the main branch.
- Utilize GitHub Actions for Continuous Integration to automate testing and deployment.
- Perform regular audits on security, performance, and data integrity.

7.2. Evaluation Criteria

Functional Accuracy

- Ensure core features work as intended.
- Validate auction workflows, including English and sealed-bid auctions.

System Stability and Uptime

- Measure the system's ability to operate without crashes or failures during peak usage.

- Track uptime percentage using monitoring tools.
- Evaluate response times and latency during high-traffic auctions.
- Monitor and resolve performance bottlenecks to ensure smooth operation.

User Feedback and Satisfaction

- Collect user feedback through surveys, ratings, and direct reviews.
- Measure user satisfaction scores.
- Analyze feedback to identify pain points and areas for improvement.
- Conduct usability testing to ensure the platform is intuitive and user-friendly.

Compliance with Functional Requirements

- Verify that all system features work as specified.
- Ensure the system meets all documented functional requirements and use cases.
- Conduct acceptance testing to confirm alignment with user and business needs.

7.3. Monitoring and Controlling Quality

Continuous Monitoring

- Use Firebase Crashlytics to track and analyze real-time errors and performance issues.
- Monitor system health and log data for error detection and analysis.

Quality Control Checks

- Implement automated tests to verify new features and bug fixes.
- Perform periodic manual testing on critical user journeys.

Review Process

- Conduct weekly internal reviews to assess progress and code quality.
- Schedule monthly stakeholder meetings to gather feedback and evaluate performance.

Issue Resolution

- Use GitHub Issues to track, prioritize, and resolve defects.
- Document resolutions and perform regression testing to prevent reoccurrence.

8. Communication Plan

8.1. Communication Channels

Weekly Team Meetings

- Weekly meetings are conducted to discuss progress, issues, and next steps.
- Members can discuss the challenges and problems they are facing.
- Group discussion to get the best solution.

Real-Time Collaboration on Discord

- Connected through audio voice channel for team discussions, real-time updates, and collaboration.
- Problem or bug fixing through screen sharing.

Code Collaboration on GitHub

- Code collaboration, and issue tracking**.**
- Review open-source implementations, and integrate relevant code into the project.
- Review other team members code and track push and pull requests.

Group chat in Facebook Messenger

- Regular group discussions and progress updates.
- Discussion for scheduling meetings or calling for instant meetings.

8.2. Reporting Frequency

Status Reports–Weekly

- Shared via Discord and GitHub Issues.
- Covers progress updates, challenges faced, and solutions discussed during weekly meetings.

- Includes a summary of code changes, bug fixes, and ongoing tasks.

Progress Review-Monthly

- Conducted through team meetings and GitHub project tracking.
- Evaluate completed features, pending tasks, and any blocks.
- Assesses system performance and identifies areas for improvement.
- Reviewing the presentation slide by the Teaching Assistant (TA).

Stakeholder Meetings–(Every month)

- Involves project's university mentors(faculty and TA).
- Covers overall project progress, performance metrics, upcoming goals, and risks.
- Includes a demonstration of new features and discussion on feedback through lab presentation and personal meetings.

8.3. Documentation

Project Presentation Slide

The document is a slide form for the monthly presentation and overall overview of the project updates.

Project Requirements and Planning Documents

This document contains an overall project overview, project requirements, resources, risk, and quality management and the last one is a communication plan.

Project Report

This document highlights the key aspects of the project, emphasizing its purpose and the efficiency of its output. It also provides a detailed analysis of how well the project's requirements have been met, ensuring that all objectives have been thoroughly fulfilled.

9. Conclusion

Thorough planning and detailed requirement gathering are vital for the successful delivery of the Online Auction System. This document outlines a clear roadmap, aligning stakeholders on objectives, scope, and deliverables. It includes strategies for resource management, risk mitigation, quality assurance, and communication, ensuring efficient goal achievement. The system aims to provide a secure, transparent, and user-friendly platform for real-time bidding, focusing on scalability, performance, and security. With continuous monitoring, regular feedback, and adaptive planning, the team is committed to addressing challenges and delivering a high-quality solution that meets stakeholder expectations and enables future enhancements.