**Online ticketing system software requirements Manage ticket bookings for events, cinemas or travel.**

**Version 1.0**

Prepared by:  
**Group Name:** 1

1. **Tu Hoang Phi Long**
2. **Cao Thanh Son**
3. **Huynh Nguyen Xuan Huy**
4. **Tran Minh Nhat**

**Instructor:** MSc. Nguyen Trung Thuan  
**Course:** Requirements Engineering  
**Lab Section:** ClassRoom RE2  
**Teaching Assistant:** N/A

**Table of Contents**

1. Introduction
   * 1.1 Document Purpose

This document is intended to provide a comprehensive and detailed overview of an online ticketing system. The system is designed to manage the ticket booking process for events, cinemas and tours. It includes the requirements, rules and features needed to meet the needs of customers and stakeholders, and provides guidance for development and implementation teams.

* + 1.2 Product Scope

The online ticketing system will allow users to search, book and pay for tickets for events, movies and tours. The system will include features such as:

-Search for events, movies or tours.

-Book tickets and manage orders.

-Make secure online payments.

-Send booking confirmations via email or text message.

-Manage user information and booking history.

1.3 Intended Audience and Document Overview

\*Intended audience:

-End users: Customers who book tickets for events, cinemas or tours.

-Administrators: People who manage the system, check content and handle issues that arise.

-Event organizers: People who post event information and track the number of tickets sold.

-Development team: Programmers and software engineers who refer to the documentation to develop and maintain the system.

\*Document overview:

-The document includes items from general description, technical requirements, design interface, to operation and support procedures. It will be the basis for developing and testing the online ticketing system.

1.4 Definitions, Acronyms, and Abbreviations

-Online Ticketing System: A system that supports ticket setup over the internet.

-QR Code: Quick response code, used to confirm tickets at events or fairs.

-API (Application Programming Interface): Application setup interface for analyzing other features.

-UX/UI: User Experience (User Experience) and User Interface (User Interface).

1.5 Document Conventions

-Main headings are numbered in "XY" format for easy reference.

-Terminology specifications will be explained in section 1.4.

-Any changes or updates will be clearly stated in the Appendix.

1.6 References and Acknowledgments

The online ticketing system is a digital platform that allows users to search, book and pay online for events, cinemas or travel packages. The system strives to create a convenient, fast and secure experience for customers, while providing powerful management tools for event organizers.

\* Outstanding features:

- User-friendly interface, support on both computers and mobile devices.

- Notification and reminder system

- Ticket confirmation via QR code and q

- Detailed reports on ticket sales status, revenue and user activity.

1. **Overall Description**

2.1 Product Overview

* + The online ticketing system is a digital platform that allows users to search, book and pay online for events, cinemas or travel packages. The system strives to create a convenient, fast and secure experience for customers, while providing powerful management tools for event organizers.

\* Outstanding features:

* + User-friendly interface, support on both computers and mobile devices.
  + Notification and reminder system
  + Ticket confirmation via QR code and q
  + Detailed reports on ticket sales status, revenue and user activity.

2.2 Product Functionality

\*The system will include the following main functions:

-Search and browse events: Users can search for events, movies or tours based on many criteria such as time, location, genre, and ticket price.

-Book tickets: Users can select the number of tickets, ticket type and add to cart.

-Online payment: The system supports many payment methods such as credit card, e-wallet and bank transfer.

-Order management: Users can view, edit or cancel the order.

-Send confirmation: After successful payment, the system will send an email or text message confirming the ticket booking to the user.

-User account management: Users can create and manage personal accounts, including personal information and ticket booking history.

-Event management: Administrators can add, modify or delete information about events, movies and tours.

* + 2.3 Design and Implementation Constraints

-Device constraints: The system must work well on popular devices such as smartphones, tablets, and desktop computers.

-Performance: The system must be able to handle high user traffic during major events without interruption.

-Security: It must adhere to high security standards, including encryption of payment data and personal information.

-Multilingual support: The interface and content must be translated into multiple languages ​​to serve a diverse user base.

-Integration with third-party APIs: To process payments and send notifications, the system must integrate with services such as PayPal, Stripe, and Twilio.

* + 2.4 Assumptions and Dependencies

\*Assumptions:

-Users have stable internet access when using the system.

-Event organizers provide detailed and accurate information about the event.

-Users are responsible for their own personal information and accounts.

\*Dependencies:

-Depends on payment API providers and notification delivery services.

-The system's infrastructure, including servers and cloud storage services, is provided by third parties such as AWS or Google Cloud.

-Legal regulations and privacy policies of each country apply to the system.

1. Specific Requirements
   * 3.1 External Interface Requirements

The online ticketing system will have the following front-end interfaces:

\*User Interface (UI):

-User-friendly web interface, easy to navigate.

-Mobile interface optimized for mobile devices and tablets.

-Main pages include: Home page, Event search page, Event detail page, Shopping cart page, Payment page, and Account management page.

\*API interface:

-Provide RESTful API for external applications to access event information, book tickets, and manage accounts.

-API will support methods such as GET, POST, PUT, and DELETE for different operations.

\*Administrator interface:

-A separate interface for administrators to manage events, users, and orders.

-Provide reports and statistics on revenue and number of tickets sold.

* + 3.2 Functional Requirements

The system will have the following functional requirements:

1.Search and Browse Events:

-Users can search for events by name, genre, date and location.

-The system displays a list of events that match the search criteria.

2.Book Tickets:

-Users can select the event and the number of tickets to book.

-The system allows users to select the ticket type (e.g. regular ticket, VIP ticket).

3.Payment:

-The system supports multiple payment methods (credit card, e-wallet, bank transfer).

-Users receive a confirmation notification after successful payment.

4.Order Management:

-Users can view, edit or cancel orders placed.

-The system stores the user's ticket booking history.

5.User Account Management:

-Users can create accounts, log in and manage personal information.

-The system allows users to change passwords and update personal information.

6.Event Management (for Admins):

-Admins can add, modify or delete events.

-The system allows admins to view revenue reports and number of tickets sold.

* + 3.3 Use Case Model

Use Case 1: Event Ticketing

Actor: End User.

Description: User searches for events, selects tickets, and completes payment to receive QR codes or e-tickets.

Main Flow:

1.User searches for events by name or category.

2.Selects an event and views details.

3.Selects the number of tickets and seats (if applicable).

4.Proceeds to payment via the integrated portal.

5.Receives e-tickets via email or in a personal account.

Use Case 2: Event Management

Actor: Event Organizer.

Description: Organizer creates and manages events to sell tickets and track revenue.

Main Flow:

1.Logs into the system as organizer.

2.Creates a new event or edits an existing event.

3.Updates ticket quantity, price, and time.

4.Tracks ticket sales via dashboard.

Use Case 3: Validates tickets at the event

Actor: Organizer or ticket checker.

Description: Scan QR code to validate ticket.

Main flow:

1.Use QR code reader app or device.

2.Scan e-ticket from attendee.

3.Confirm valid ticket information and allow entry.

4.Store used ticket status in system.

1. Other Non-functional Requirements
   * 4.1 Performance Requirements

-Response Time: The system should have a response time of less than 2 seconds for search and booking requests. Page load time should not exceed 3 seconds on mobile and desktop devices.

-Processing Capacity: The system should be able to handle at least 1000 concurrent users without performance degradation.

-Database Optimization: Database queries should be optimized to ensure high performance, especially in search and payment operations.

-Scalability: The system should be scalable to handle user and event growth without changing the underlying architecture.

* + 4.2 Safety and Security Requirements

-Data Security: All user personal information and payment data must be encrypted during transmission and storage.

-User Authentication: The system must require users to authenticate via login name and password. A secure password recovery feature is required.

-Access Rights: The system must have different levels of access rights for users and administrators, ensuring that only administrators have the right to edit event information and manage users.

-Attack Detection and Prevention: The system must have protections against attacks such as SQL injection, XSS (Cross-Site Scripting) and DDoS (Distributed Denial of Service).

-Data Backup: Data must be backed up periodically to ensure recovery in case of data loss.

* + 4.3 Software Quality Attributes

\*Availability:

The system is always ready to use and easy to access from any device.

\*Usability:

The interface is intuitive and easy to use even for users who are not familiar with technology.

\*Compatibility:

-Works well on popular browsers (Chrome, Firefox, Safari) and Android and iOS operating systems.

\*Maintainability:

-The source code must be clear and have full documentation for easy maintenance and updating.

\*Efficiency:

-Optimize the use of system resources to minimize operating costs and maximize performance.

\*Scalability:

-Design a flexible system to integrate new features without affecting current operations.

\*Reliability:

-Ensure stable operation, no serious errors in core functions.

1. Other Requirements

\*Multi-language support:

-The system should support at least two languages ​​(e.g. English and Vietnamese) to serve a wide range of users.

\*Integrate AI technology:

-Support AI chatbots to answer customer questions and suggest suitable events based on user preferences.

\*Reporting and analytics:

-Provide detailed reports on ticket sales, user data analysis, and event performance for organizers.

\*Technical support:

-Provide 24/7 customer support channels via email, phone, or live chat.

1. Appendices

6.1 Related Documents

Technical Documentation:

-Payment Integration API Guide.

-System Deployment and Server Configuration Documentation.

References:

-Similar ticketing systems such as Eventbrite, Ticketmaster, and Booking.com.

-PCI DSS and OWASP Top 10 Security Standards.

6.2 Abbreviations and Definitions

-API: Application Programming Interface.

-QR Code: Quick Response Code.

-UX/UI: User Experience / User Interface.

-2FA: Two-Factor Authentication.

-DDoS: Distributed Denial of Service.

6.3 Tables and Figures

-System Architecture Diagram.

-Data Flow in the Ticketing Process.

-User Interface Example (Mobile and Desktop).

6.4 Contact

Developer Information:

Email: thanhson29012005@gmail.com

Phone: 0966164897

Address: 254 Đ.Nguyễn Văn Linh,Thanh Khê, Đà Nẵng ,Việt Nam

**1. Introduction**

**1.1 Document Purpose**

The purpose of this document is to provide a detailed overview of the online ticketing system, including functional and non-functional requirements, user interfaces, use cases, as well as other requirements related to the development, implementation, and maintenance of the system. This document will serve as a guide for stakeholders, including:

-Developers: Provide the information needed to design and develop the system according to the requirements.

-Project managers: Help track progress and ensure that requirements are implemented on time and within budget.

-End users: Provide information about the features and functionality of the system, helping them understand how to use the system effectively.

-Administrators: Guide the management and maintenance of the system, ensuring that it operates stably and securely.

This document also serves as a basis for testing and evaluating the system, ensuring that all requirements are met and the system functions as expected. By providing a clear and detailed document, we hope to facilitate the development and implementation of an online ticketing system.

**1.2 Product Scope**

The online ticketing system provides a comprehensive solution for booking tickets for events, cinemas, and travel. Users can search, book, pay online, and receive e-tickets via QR codes

Event organizers use the system to manage events, track revenue, and authenticate tickets. The system ensures a convenient, secure experience for users and complies with legal requirements for security and online payments.

Objective: To meet the needs of fast, efficient ticket management, and enhance the experience for both users and organizers.

**1.3 Intended Audience and Document Overview**

**Intended Audience**

This document is intended for the following audiences:

1.Developers:

Understand the functional and non-functional requirements of the system to implement it as designed.

2.Project Managers:

Monitor progress and ensure the project is within scope and objectives.

3.Event Organizers:

Understand how the system can be used to effectively manage events and sell tickets.

4.Stakeholders:

Evaluate the ability to meet market needs and business objectives.

**Document Overview**

The document is organized into the following main sections:

-Introduction: Purpose, scope, and definitions of key terms related to the system.

-General Description: Introduces the system structure, key functionality, and associated assumptions.

-Specific Requirements: Describes the interface requirements, functionality, and use case models in detail.

-Non-functional requirements: Address performance, security, and other quality attributes of the system.

-Additional requirements: Language support, AI integration, and data analytics.

-Appendix: Provides references, abbreviations, and support contact information.

**1.4 Definitions, Acronyms, and Abbreviations**

1.4.1 Definitions

* Online ticketing system: An online platform that allows users to search, book, and manage tickets for various events.
* User: An individual who uses the system to search for and book tickets for events.
* Administrator: A person who has the authority to manage the system, including adding, modifying, and deleting events and managing users.
* Event: Activities or programs that users can attend and book tickets for, such as concerts, seminars, or sporting events.
* Ticket: A paper or electronic code that allows a user to attend a specific event.

1.4.2 Acronyms and Abbreviations

* UI: User Interface - the part of the system where the user interacts with the system.
* UX: User Experience - the overall feeling the user has when using the system.
* API: Application Programming Interface - a set of rules that allows different software to communicate with each other.
* DB: Database - where information about users, events and tickets is stored.
* SSL: Secure Sockets Layer - a security technology that helps protect data transmitted over the internet.
* GDPR: General Data Protection Regulation - a European Union regulation on the protection of personal data.

1.4.3 Other terms

* Online payment: The process of making a financial transaction over the internet to buy tickets.
* Promotion: Discounts or incentives to attract users to book tickets.
* Seat: A specific location in an event that users can choose when booking tickets..

**1.5 Document Conventions**

1.5.1 Text Format

-Title: All main headings will be bold and larger in size for easy identification.

-Subheadings: Subheadings will be italicized and smaller in size than the main heading.

-Lists: Lists will be presented in numbered or bulleted form for easy reference.

-Citations: Important quotes or information will be placed in boxes or italics for prominence.

1.5.2 Symbols and Icons

-Symbols: Symbols such as “✔” will be used to indicate completed or confirmed features.

-Icon: Icons will be used to illustrate concepts or features, making it easier for readers to visualize.

1.5.3 Language

-Primary language: The document will be written in Vietnamese, with some sections possibly translated into English to serve international audiences.

-Technical terms: Technical terms will be clearly defined in section 1.4 to ensure that the reader can understand the meaning correctly.

1.5.4 Figures and Statistics

-Measurements: All figures and statistics will be presented in metric units (m, kg, l) and currencies will be clearly stated (VND, USD).

-Charts and Images: Charts and images will be used to illustrate complex figures and information, with captions for clarification.

1.5.5 References

-References: All references will be listed at the end of the document, including books, articles, and online resources.

-Links: Links to online documents will be presented in underlined text and in a different color for easy identification.

1.5.6 Document Updates

-Versions: Each version of the document will be numbered and clearly stated with the date of issue.

-Revision History: A revision history table will be provided to track changes and updates in the document.

**1.6 References and Acknowledgments**

1.6.1 References

The following is a list of documents, books, articles, and online resources that were consulted during the preparation of this document:

1.Books:

-Sommerville, I. (2011). Software Engineering. 9th Edition. Addison-Wesley.

-Pressman, R. S. (2014). Software Engineering: A Practitioner's Approach. 9th Edition. McGraw-Hill.

2.Articles:

-Boehm, B. W. (1988). "A Spiral Model of Software Development and Enhancement". ACM SIGSOFT Software Engineering Notes, 11(4), 14-24.

-Parnas, D. L. (1972). "On the Criteria to Be Used in Decomposing Systems into Modules". Communications of the ACM, 15(12), 1053-1058.

3.Online resources:

-W3C. (2023). "Web Content Accessibility Guidelines (WCAG)". https://www.w3.org/WAI/WCAG21/quickref/

-OWASP Foundation. (2023). "OWASP Top Ten". <https://owasp.org/www-project->top-ten/

1.6.2 Acknowledgements

We would like to express our sincere thanks to the individuals and organizations who supported the development of this document:

-Development Team: Thank you to the members of the development team for their comments and continuous efforts to perfect the online ticketing system.

-Test users: Thank you to the users who participated in the testing process and provided valuable feedback, helping to improve the system's features and user experience.

-Consultants: Thank you to the experts in the fields of information technology and project management who provided knowledge and guidance throughout the development of the document.

**2. Overall Description**

**2.1 Product Overview**

The online ticketing system is a platform that allows users to easily search, book, and manage tickets for various events, including but not limited to concerts, sporting events, conferences, and other entertainment activities. The platform is designed to provide a smooth, secure, and convenient user experience, saving users time and effort in booking tickets.

**2.2 Product Functionality**

1. End User Features

1.Search for events/movies/travel:

-Search by keyword, location, date, or category.

-Filter results by price, event type, or rating.

2.Book tickets online:

-Select number of tickets and seat location (if applicable).

-Optionally add additional services (e.g. food, gifts).

3.Secure payments:

-Support multiple payment methods: credit card, e-wallet, bank transfer.

-Integrate two-factor authentication (2FA) for transactions.

4.Manage e-tickets:

-Receive tickets via email/QR code.

-Store and access ticket purchase history.

5.Notifications and reminders:

-Receive ticket booking confirmation notifications.

-Send reminders about event times and locations.

2. Event Organizers Features

1.Event Management:

-Post events/movies/tours with detailed information (description, images, dates, ticket prices).

-Customize ticket types (regular, VIP) and maximum quantity.

2.Revenue Tracking and Ticket Sales:

-Show real-time ticket sales status.

-Report revenue, number of tickets sold, and customer feedback.

3.Ticket Validation:

-QR code scanning tool at the entrance.

-Validate tickets and prevent fraud.

4.Send notifications:

-Notify schedule changes or event updates.

-Send thank you emails/SMS after the event.

3. System-wide Features

1.User Management:

-Register/login via email, social media accounts, or business accounts.

-Securely manage personal information and passwords.

2.Security system:

-Encrypt transaction data and personal information.

-Comply with international security standards such as PCI DSS.

3.Reporting and analyzing:

-Analyze user behavior data to optimize revenue.

-Report event performance and predict trends.

4.Multilingual support:

-The interface system can be customized in languages ​​(English, Vietnamese, etc.).

5.Integration and expansion:

-Support integration with CRM or marketing automation systems.

-Easily expandable to serve other markets or fields.

**2.3 Design and Implementation Constraints**

**Development Languages:**

* Frontend: JavaScript/TypeScript with React.js for web apps and React Native or Flutter for mobile apps.
* Backend: Python (using Django or Flask) or Node.js to build APIs.

**Technologies used:**

1.Frontend:

* React.js: Build web user interfaces.
* React Native/Flutter: Create cross-platform mobile apps.
* Redux/Context API: State management for complex apps.
* CSS Frameworks: Use Bootstrap or Tailwind CSS to optimize interface design.

2.Backend:

* Django REST Framework or Express.js: Develop fast and powerful APIs.
* WebSocket: Support real-time features like ticket status updates.
* OAuth2: Authenticate users via Google/Facebook accounts.

3.Database:

* PostgreSQL: Store relational information like events, users, tickets.
* MongoDB: Stores unstructured data such as transaction logs or attachments.

4.Payments and notifications:

* Stripe/PayPal/ZaloPay: Integrates online payments.
* Twilio/SendGrid: Sends SMS and email notifications.

5.Infrastructure:

* AWS/GCP: Hosting and deploying infrastructure.
* Cloudflare: Protects systems from DDoS attacks and optimizes content delivery.

**Limitations:**

1.Budget:

* Financial constraints affect the choice of paid tools and services, such as AWS premium plans or complex payment services.

2.Infrastructure and performance:

* Initially running only on the cloud, the system may have difficulty increasing the number of users and concurrent transactions.
* Limited bandwidth in some areas may degrade the user experience.

3.Security:

* Requires significant cost and effort to ensure the system complies with security standards (PCI DSS, GDPR).

4.Development Time:

* Short development timeframes (e.g. less than 6 months) can impact the quality of testing and product optimization.

5.Integration:

* Relies on the stability of third-party services such as payment gateways or notification systems, which can cause issues beyond control.

6.Cross-platform Support:

* Ensuring consistency between the web interface and mobile app can be time-consuming and labor-intensive to develop.

**2.4 Assumptions and Dependencies**

1. Assumptions:

1.End users:

* End users have stable internet access to use the system.
* Users are familiar with using mobile applications or websites.
* Users will provide accurate information when registering an account or booking tickets.

2.Event organizers:

* Organizers have enough information to post events, including time, location, and ticket price.
* Organizers can handle customer support requests related to the events they manage.

3.Technical infrastructure:

* The system will be deployed on a cloud platform (AWS, Google Cloud, or Azure) to ensure stability and scalability.
* Transaction data and personal information will be protected according to modern encryption standards.
* Third-party payment gateways (PayPal, Stripe, ZaloPay, etc.) operate stably and have full API documentation.

4.Ticket Management:

* E-tickets will be accepted at event venues without requiring physical tickets.
* The organizer will ensure that the control gate (QR code scanning) at the event is working properly.

5.Legal Regulations:

* The system complies with personal data protection regulations (e.g. GDPR if applicable to EU users).
* Financial transactions comply with PCI DSS standards to ensure payment security.

2. Dependencies:

1.Payment Gateway:

* The system depends on payment service providers (Stripe, PayPal, ZaloPay) to process online transactions.
* The processing time and transaction status of payment gateways may affect the user experience.

2.Notification sending system:

* Relies on email services (SendGrid, Amazon SES) and SMS (Twilio) to send ticket booking confirmations and event notifications.

3.Cloud Infrastructure:

* The system is deployed on cloud services (AWS, Google Cloud) to ensure availability and performance.
* The scalability and security of the system will depend on the configuration and services of the cloud provider.

4.Third-party API integration:

* Features such as user authentication via social networks (Google, Facebook) or event location maps (Google Maps) will depend on the availability of APIs from these providers.

5.User devices:

* The system depends on the user's device (computer, phone) to ensure a good experience, especially with QR code and e-ticket features.

6.Development team:

* The success of the system depends on the ability and experience of the development team in using the selected technologies.

7.Network services:

* The system depends on the internet connection to handle real-time requests, such as ticket booking and QR code scanning.