Project Zoom
Construction Management

Requirements Specification V1.0

Project Zoom Construction management

A **multi-language**, **internationalized SaaS platform** based on tasks (jobs) and tickets (defects), integrating multiple functions such as task allocation, progress tracking, defect management, document management, safety training, and financial management.

Requirements Specification Revised Version (V1.0 by Alex Zhao)

Table of contents

一、Overview

Note: This English version is translated through Google Translate. Please refer to the Chinese version.

Project Zoom is a comprehensive management system designed specifically for construction sites. It aims to improve site management efficiency, transparency and collaboration through digital means . The system is centered on jobs and tickets , and integrates multiple functions such as work task allocation, progress tracking, defect management, document management, safety training and financial management . As an international multilingual system, Project Zoom aims to break down language and cultural barriers and provide a full range of management solutions for global construction project management .

1.1、System Goal

The main objectives of the system include:

1. Simplify work processes, improve management efficiency , and enhance project management capabilities ;
2. Improve construction site management efficiency and reduce delays caused by information asymmetry ;
3. , and promote communication among all parties through systematic task and document management ;
4. Optimize resource allocation and reduce waste ;

5. Improve project quality and strengthen safety management ;
6. Simplify financial processes and reduce management costs ;
7. Provide data support to assist decision making ;
8. Achieve multi-language support to facilitate international team collaboration;
9. Supports multi-time zone management to facilitate global team collaboration ;
10. Provide cross-platform support to achieve a seamless working experience on different devices.
11. Provide intelligent report generation, automatically adapting to different countries' reporting formats and requirements ;

Through these goals, Project Zoom will not only improve the management efficiency of individual construction projects, but also provide a unified, intelligent management platform for multinational construction companies through future expansion , achieving project collaboration and resource optimization on a global scale.

1.2、User roles

- ✓ Site Manager
- ✓ Subcontractor
- ✓ Worker
- ✓ Accountant

- ✓ Engineer
- ✓ Another (other characters who may be involved)

1.3、Brief description of the glossary

Note: The flowcharts in this article can be double-clicked to enlarge and browse

- **Job:** A specific work task assigned to site personnel, which must be completed within a specified time, and the progress and results must be tracked through the system.
- **Ticket:** Records problems found on the construction site or change requests, which need to be processed and tracked through the system.
- **Time Extension:** A worker requests an extension when he or she is unable to complete a task on time, stating the reason and redefining the completion time.
- **Variation:** During the project, changes in work content or scope due to changes in requirements or conditions require signature confirmation from both parties.

1.4、Disclaimer

This system design document (hereinafter referred to as the

"Document") is for informational purposes only and is intended to provide guidance for the development and implementation of the "Project Zoom Construction Management App". The information and design solutions in the Document are based on current best practices and technical standards, but their applicability and accuracy in other environments are not guaranteed.

1. Accuracy of Information

The information in this document is based on our current understanding and assumptions and may be updated as technology advances and business needs change. We make no express or implied warranties as to the completeness, accuracy or reliability of the information in this document.

2. Limitation of Liability

In no event will we be liable for any direct, indirect, incidental, special or consequential damages, including, but not limited to, business interruption, loss of data or lost profits, arising out of the use of or reliance on the information in this document.

3. Third-party tools and services

The third-party tools, services or products that may be mentioned in this document are for reference only and do not constitute any form of recommendation or guarantee. When using these tools or services, users should evaluate their applicability and risks.

4. Intellectual Property

All contents in this document, including but not limited to text, graphics, design and code, are protected by copyright and other intellectual property laws. They may not be copied, distributed or used for commercial purposes without authorization.

5. Modifications and Updates

We reserve the right to modify or update this disclaimer and the contents of this document at any time without prior notice. It is the user's responsibility to regularly review the document for the latest information.

二、Business functional requirements

2.1 Job Management

2.1.1 Task Management Description

- Create and assign work tasks
- Set mission goals and time frames

- Designated person in charge (subcontractor or worker)
- Record task progress (percentage)
- Processing of time extension applications
- Recording timeout completion

2.1.2 Task Management Flowchart

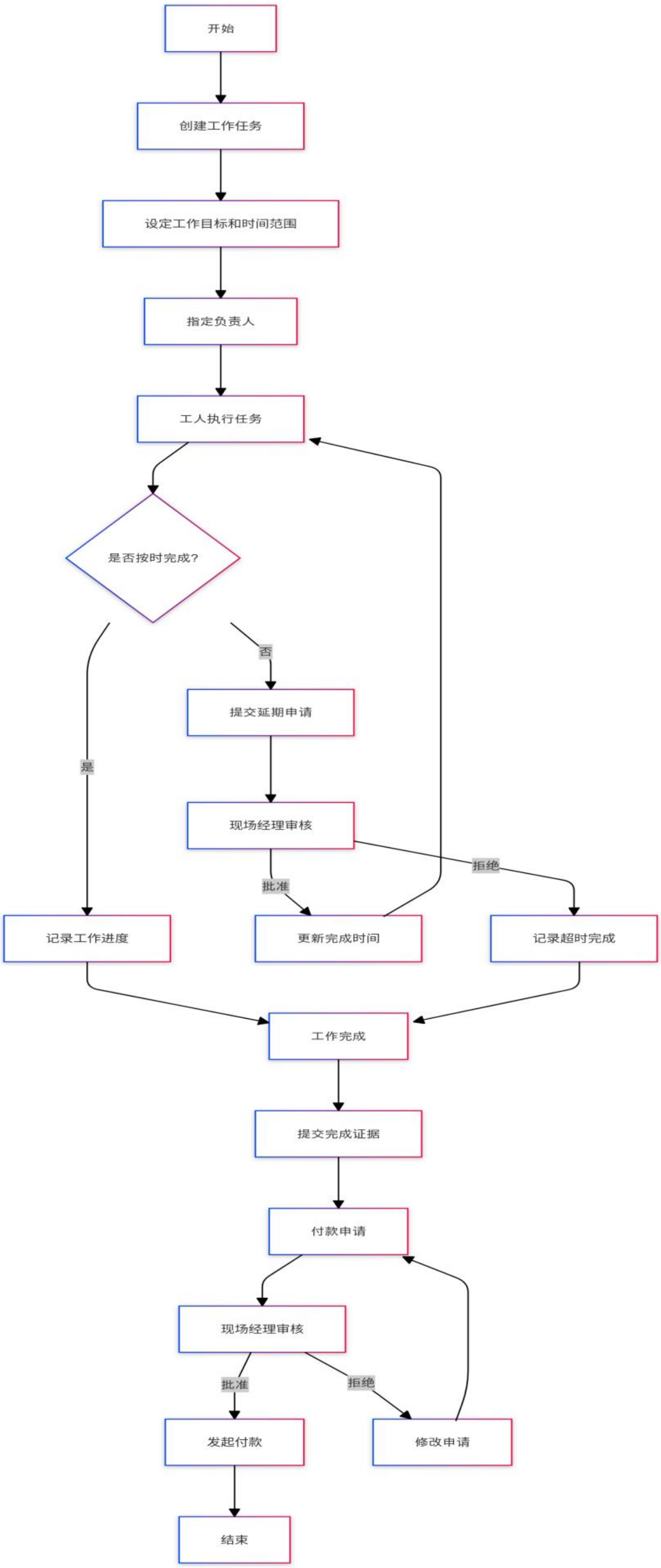


Figure 2.1.2.1

Process Description:

step	describe
1	Start the task management process
2	Create a work task
3	Set mission goals and time frames
4	Designated person in charge (subcontractor or worker)
5	Workers perform tasks
6	- Determine whether it is completed on time task progress if completed on time - Submit an extension request if the deadline is not met
7	- Site Manager reviews extension request - If approved, update the completion time and return to the task - If rejected, the record will time out.
8	Submit proof of completion
9	Submit payment request
10	- Site manager reviews payment request - If approved, initiate payment - If rejected, revise your application and resubmit
11	End the task management process

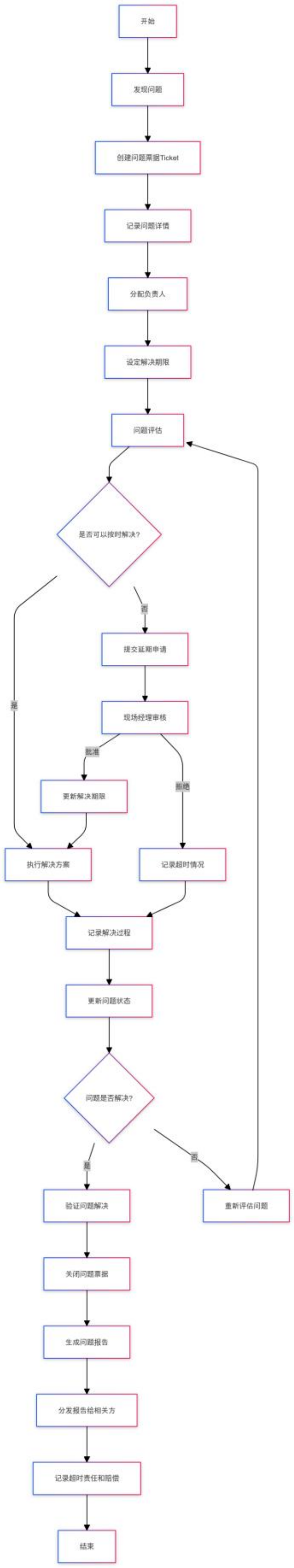
2.2 Ticket Management

2.2.1. Defect Management Description

- Create and assign defect tickets ;
- Record defect details (including pictures, videos, etc.) ;
- Track defect resolution progress ;

- Generate and distribute defect reports ;
- Record and track defects ;
- Generate defect reports ;
- Manage maintenance records ;

2.2.2 Defect Management Flowchart



2.2.2.1 Figure

The flow chart is described as follows:

No	step
1	Defect discovery: Project managers or workers discover quality problems or defects on the construction site.
2	Create Defect Ticket: Create a new defect ticket in the system to track the entire defect management process.
3	Record defect details: Describe in detail the nature, location and severity of the defect.
4	Take photos/videos of defects: Use your mobile device to take photos or videos of the defects to better illustrate the problem.
5	Mark defects on the picture: Use the tools provided by the system to mark the specific location and scope of the defects on the photo.
6	Add Text Description: Add any additional information or instructions to help maintenance personnel better understand the problem.
7	System automatically generates reports: Based on the collected information, the system automatically generates a detailed defect report.
8	Send report to relevant work type supervisor: Send the report to the subcontractor or work type supervisor responsible for the relevant work.
9	Arrange maintenance: The relevant person in charge arranges maintenance work and allocates personnel and resources.
10	Perform repair work: Maintenance personnel perform repair work based on the information in the report.
11	Maintenance personnel upload completion evidence: After the maintenance is completed, the worker uploads photos, videos or other supporting materials of the repair.
12	Check repair results: Project managers or quality inspectors verify repair results.
13	Determine whether it is repaired: - If the repair is successful, proceed to the step of closing the Defect Ticket. - If the repair is not successful, reallocate the repair task and return to the step of arranging repairs.

No	step
14	Close Defect Ticket: Confirm that the defect has been fixed and close the corresponding defect ticket.
15	Update defect management database: Update the information of the entire defect management process into the system database.
16	Generate final report: The system generates a final report covering the entire defect management process, including the initial report, repair process and final results.
17	End: Complete the entire defect management process.

2.3 Document Management

2.3.1 Document Management Description

- Upload and manage drawings (PDF format)
- Drawing marking and modification records
- Keep meeting minutes and site diaries
- Managing variation files

2.3.2 Document Management Flowchart

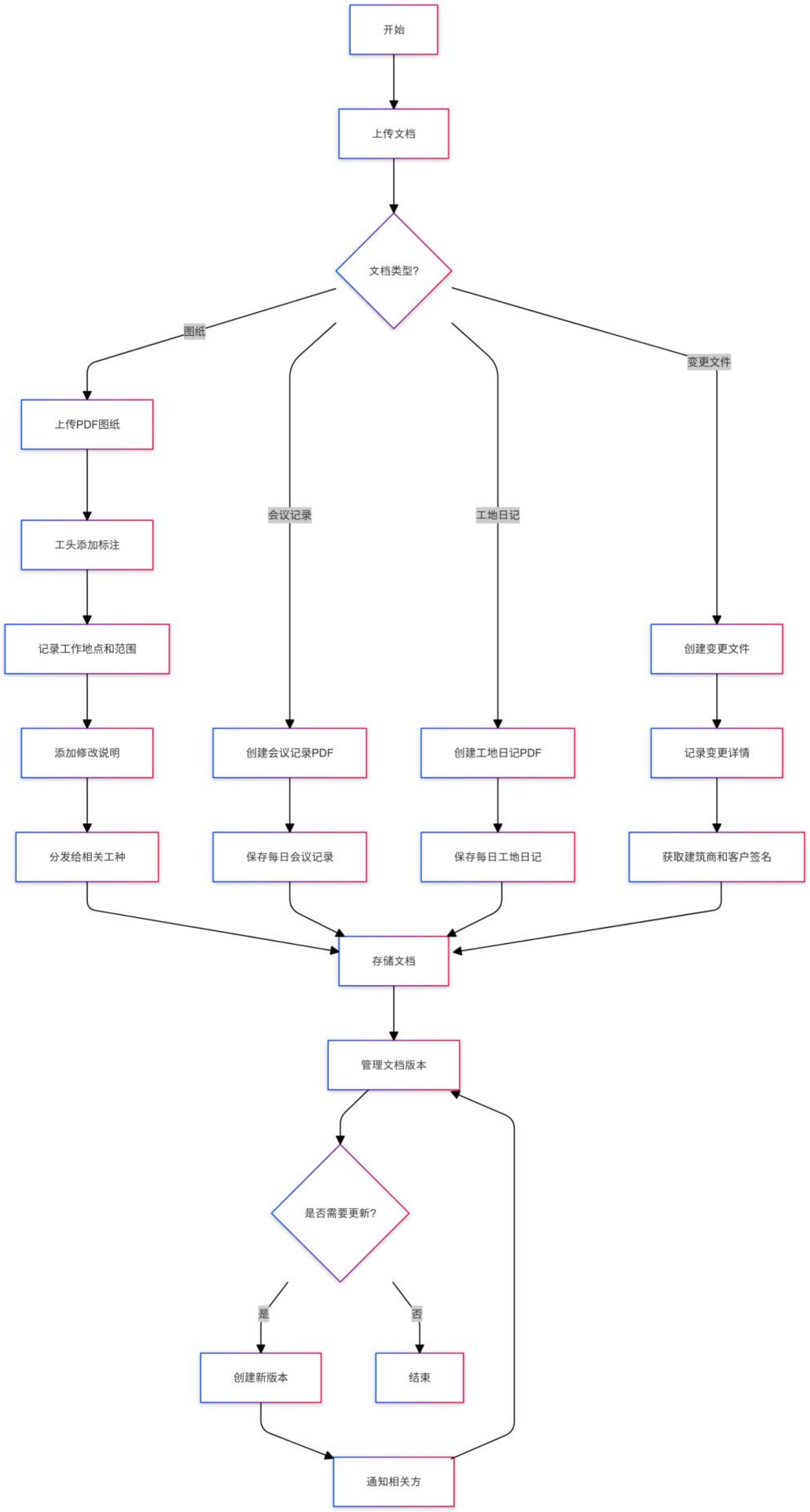


Figure 2.3.2.1

Document management flow chart text description:

No	step
1	Start the document management process
2	Upload Documents
3	Different processing is performed according to the document type: – Drawings: a. Upload PDF drawings b. Foreman adds annotations c. Record work location and scope d. Add modification instructions e. Distribute to relevant trades – Meeting minutes: Create and save daily meeting minutes PDF – Site diary: Create and save daily site diary PDF – Change documents: a. Create change documents b. Record change details c. Obtain signatures from the builder and the client
4	Storing Documents
5	Managing Document Versions
6	Check if an update is needed – If an update is needed, create a new version and notify the relevant parties – If an update is not needed, end the process

Table 2.3.2.1

2.4 Safety training management

2.4.1. Safety training description

- Release safety training videos and documents ;

- Record worker training ;
- Collect statistics on workers' training and assessment;

  Management worker qualification certificate

2.4. Safety training flow chart

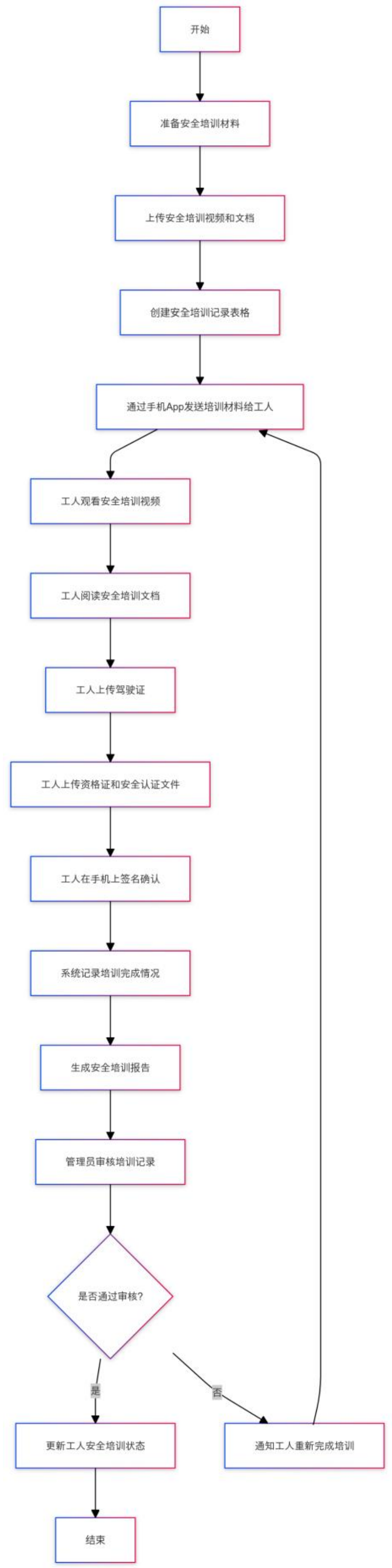


Figure 2.4.2.1

No	step
1	Begin the safety training process
2	Preparing safety training materials
3	Upload safety training videos and documents
4	Create a safety training record form
5	Send training materials to workers via mobile app
6	Workers watching safety training video
7	Workers reading safety training documents
8	Workers upload driver's license
9	Workers upload qualification certificates and safety certification documents
10	Workers sign and confirm on their mobile phones
11	System records training completion
12	Generate safety training reports
13	Administrator review training records
14	Determine whether the audit is passed
	- Update worker safety training status if passed
	- If not, notify the worker to complete the training again

15	End the safety training process

Table 2.4.2.1

2.5 Payment Management

2.5.1. Payment Management Description

- ✧ Record evidence of work completed (pictures, videos, etc.)
- ✧ Manage invoices and payment requests
- ✧ Processing payment confirmation process

2.5.2 Payment Management Flowchart

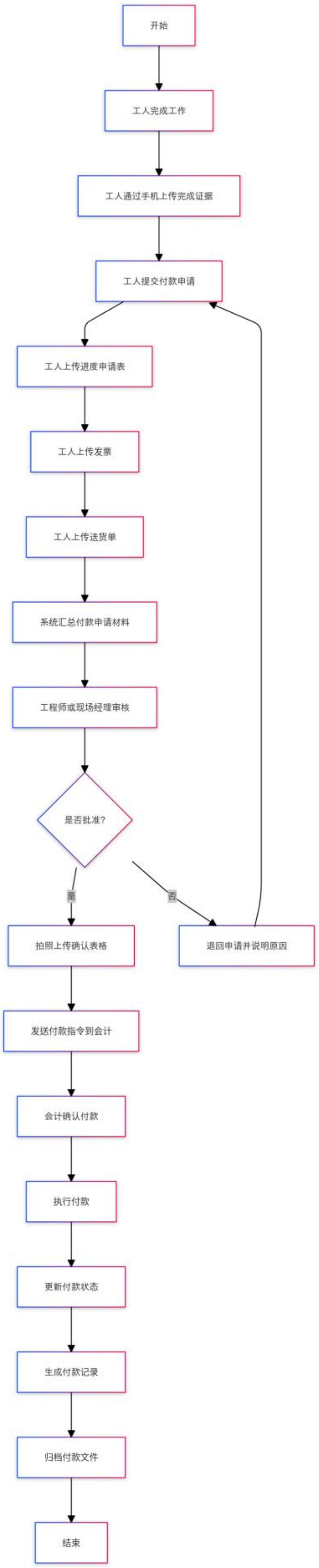


Figure 2.5.2.1

Payment Management Process Description

No	step
1	Start the payment management process
2	Workers complete the work
3	Workers upload completion evidence (pictures, videos, etc.) via mobile phone
4	Workers submit payment requests
5	Workers upload progress application form
6	Workers upload invoices
7	Worker uploads delivery note
8	System summarizes payment application materials
9	Review by engineer or site manager
10	Determine whether to approve
	- If approved, take a photo and upload the confirmation form
	- If not approved, return the application and explain the reason
11	Send payment instructions to accounting
12	Accounting confirms payment
13	Execute Payment
14	Update payment status

No	step
15	Generate payment records
16	Archive payment files
17	End the payment management process

Table 2.5.2.1

2.6 Report Management

2.6.1. Report Management Description

- Custom report templates
- Automatically populate report data
- Generate report in PDF format

2.6.2 Report Management Flowchart

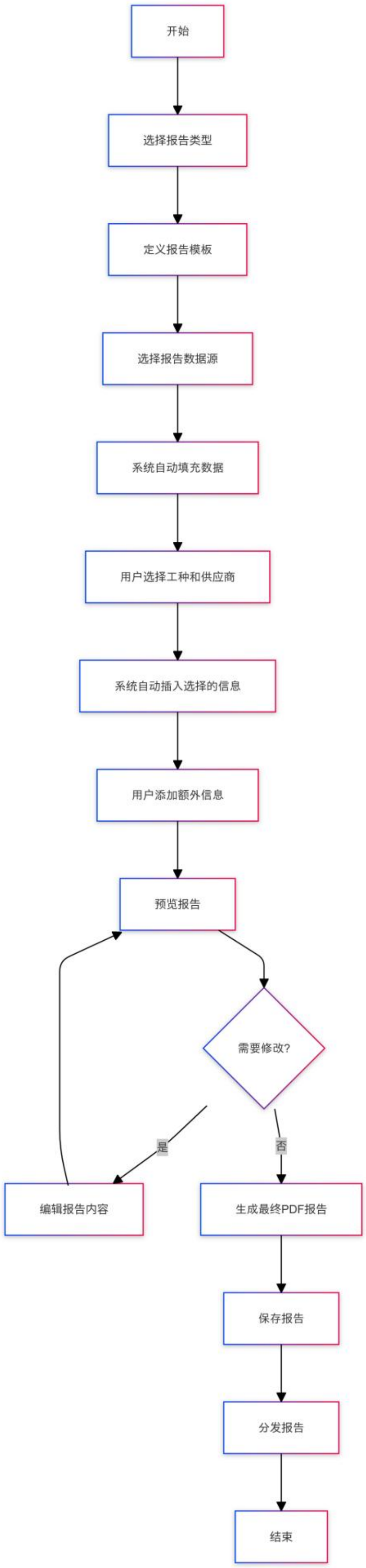


Figure 2.6.2.1

Process description:

No	step
1	Start the report generation process
2	Select the report type (e.g. weekly work schedule, progress report, etc.)
3	Define report template (if new report type)
4	Select the report data source (select relevant data from the system)
5	The system automatically fills data into the report template
6	The user selects the relevant type of work and supplier
7	The system automatically inserts the selected type of work and supplier information
8	Any additional information or instructions added by the user
9	Preview the generated report
10	Determine whether modification is needed
	- If you need to modify, return to edit the report content
	- If no changes are needed, proceed to the next step
11	Generate final report in PDF format
12	Save the report to the system
13	Distribute reports to relevant parties

No	step
14	End the report generation process

Table 2.6.2.1

2.7 User Registration and Subscription

2.7.1 User registration and subscription process

From the customer's perspective, the following steps are involved:

1. **Visit website:**

1. Customers visit our company's official website to learn about our products and services.

2. **Choose a plan:**

1. Browse different subscription plans and compare features and prices.
2. Choose a plan that fits your needs.

3. **Register an account:**

1. Provide basic information (such as name, email, password) to register.

2. You may need to verify your email address (or need mobile phone SMS verification, this is pending due to issues with international mobile phones receiving SMS) .

4. Fill in the details:

1. Provide additional personal or business information (such as company name, address, phone number).

5. Select payment method:

1. Select a payment method (e.g. credit card, PayPal).
2. Enter your payment information.

6. Confirm subscription:

1. Review and confirm your subscription plan and payment information.
2. Agree to the Terms of Service and Privacy Policy.

7. Complete your registration:

1. Submit your information to complete your registration and subscription.
2. Receive a confirmation email or notification.

8. Get started:

1. Log in to the SaaS application and start using the service.
2. Choose to join an enterprise or create a new one for trial.
3. There will be a guided tutorial or setup wizard to help users get started.

9. To manage your subscription:

1. View and manage subscriptions in your user account.
2. Can upgrade, downgrade or cancel subscription.

2.8 SaaS Basic Platform

The following only lists the necessary functions without going into details:

1. User Management:

1. User registration, login, logout
2. User Information Management
3. Password reset and modification
4. Multi-factor authentication

2. Organization Management:

1. Create and manage organizations

2. Organizational Hierarchy

3. Organization member management

3. Role and permission management:

1. Creating and managing roles

2. Assign permissions

3. Role-Based Access Control (RBAC)

4. Subscription and Billing:

1. Subscription plan management

2. Billing and Invoicing

3. Payment Gateway Integration

5. Multi-tenancy support:

1. Tenant Isolation

2. Tenant configuration management

6. Logging and Auditing:

1. User activity log

2. System Operation Audit

3. Security incident monitoring

7. Notifications and Messages:

1. Email and SMS notifications
2. In-app messaging
3. Notification template management

8. Data Analysis and Reporting:

1. Data statistics and analysis
2. Custom report generation
3. Data Visualization

9. API Management:

1. API Key Management
2. API Access Control
3. API usage monitoring

10. File and Document Management:

1. File upload and storage
2. Document version control
3. File sharing and permission management

11. Configuration and Setup:

1. System Configuration Management

2. App Settings
3. Localization and internationalization support

12. Security and Compliance:

1. Data encryption
2. Compliance management (such as GDPR)
3. Security strategy and protection

13. Customer Support and Feedback:

1. Work Order System
2. User feedback collection
3. Support Documents and FAQ

三、User interface requirements

3.1. Multi-terminal integrated user interface

In order to achieve a seamless working experience, the system must realize the following multi-terminal integration:

1. Web:

- ✓ Scope of application: Suitable for office staff, project managers and customers ;

- ✓ Users who visit the website for the first time need to log in on the web, and the system administrator/user administrator's organization management needs to be done on the web;
- ✓ Necessity: Provides comprehensive functional access, suitable for complex data entry and analysis work ;

2. Desktop (supports both Windows and MacOS):

- ✓ Scope of application: Mainly for project managers and construction site managers.
- ✓ Necessity: Supports offline work and is suitable for processing large amounts of data and complex drawing management.

3. Mobile (iOS/Android phones and tablets):

- ✓ Scope of application: Suitable for on-site workers, supervisors and mobile office managers.
- ✓ Necessity: Provide real-time information updates and quick problem reporting, and support on-site photography and document uploading.

3.1. Necessity of a multi-terminal integrated user interface

The necessity of providing multi-platform support:

1. Comprehensive coverage: Users with different roles can access the system in different scenarios.

2. Information synchronization: Ensure data on all platforms is consistent and up-to-date.

3. Improved efficiency: Users can choose the device that best suits their current work environment.

4. Enhanced collaboration: Supports seamless collaboration and information sharing across different devices.

5. Adaptability: Meet the diverse working environment needs of the construction industry.

By providing web, desktop and mobile support, the Project Zoom management system can fully meet the various needs of construction site management and improve work efficiency and collaboration quality.

四、Performance and safety requirements

4.1 Performance requirements

1. Response time:

○ Page loading time: less than 2 seconds on average

○ API response time: 90% of requests should be completed within 500ms

○ Complex data processing operations: no more than 5 seconds

2. Concurrent Users:

- Supports at least 10,000 concurrent users
- Capable of handling 50,000 concurrent connections at peak times

3. Scalability:

- The system should be able to scale easily to support growth in the number of users
- Supports horizontal expansion, which can increase system capacity by adding more servers

4. Data processing capabilities:

- Capable of processing at least 1,000 database transactions per second
- Support fast upload and download of large files (such as drawings and videos)

5. Availability:

- System uptime should reach 99.9%
- Planned maintenance time should not exceed 4 hours per month

6. Data backup and recovery:

- Support real-time data backup
- Data recovery time objective (RTO) is no more than 1 hour

4.2 Performance assurance measures

1. Microservices Architecture:

✧ Split the system into independent microservices, such as user management, document management, task management, etc .;

✧ Use containerization technologies (such as Docker) and orchestration tools (such as Kubernetes) to manage and scale microservices ;

2. Load Balancing:

✧ Use a load balancer (such as Nginx or a cloud service provider's load balancing service) to distribute traffic ;

✧ Implement intelligent routing to distribute requests to the most suitable server ;

3. Cache strategy:

✧ Use a distributed caching system (such as Redis) to cache frequently accessed data

- ✧ Implement multi-level caching, including browser caching, CDN caching and server caching

4. Database optimization:

- ✧ Use the read-write separation strategy to divert read operations to the slave database
- ✧ Implement database sharding to distribute data to multiple database instances based on user or project ID
- ✧ Optimize database indexes and query statements

5. Asynchronous processing /asynchronous architecture :

- ✧ Use message queues (such as RabbitMQ) to handle non-real-time tasks
- ✧ Implement a background job processing system to handle time-consuming operations (such as report generation)

6. CDN acceleration:

- ✧ Use a content delivery network (CDN) to speed up the loading of static resources
- ✧ Store large files (such as drawings and videos) in object storage services and distribute them through CDN

7. Front-end optimization:

- ✧ Implement lazy loading and on-demand loading to reduce initial loading time
- ✧ Use server-side rendering (SSR) to improve the first screen loading speed

8. Monitoring and Auto-scaling:

- ✧ Implement a comprehensive monitoring system to detect performance bottlenecks in a timely manner
- ✧ Configure automatic expansion policies to automatically increase or decrease server resources based on load

9. API current limiting and circuit breaking:

- ✧ Implement API current limiting mechanism to prevent a single user from over-consuming resources
- ✧ Use the circuit breaker pattern to prevent cascading failures in the system

10. Multi-tenant architecture:

- ✧ Realize data isolation to ensure data security for different users
- ✧ Optimize multi-tenant database queries to improve query efficiency

4.3 Security measures

Since this system is in an open Internet environment, the following measures are taken to ensure the security of the SaaS platform in the Internet environment:

1. User authentication and authorization:

- ✧ Implement a strong password policy that requires passwords to contain uppercase and lowercase letters, numbers, and special characters ;
- ✧ Support multi-factor authentication (MFA), such as SMS verification code or authenticator app ;
- ✧ Implement role-based access control (RBAC) to ensure that users can only access resources within their permission scope ;
- ✧ Authentication and authorization using OAuth 2.0 and OpenID Connect ;

2. Data Encryption:

- ✧ Encrypt all data in transit using TLS 1.3 ;
- ✧ Encrypt stored sensitive data using AES-256 ;
- ✧ Implement end-to-end encryption to protect communications between users ;

3. API Security:

- ✧ API authentication using API keys and JWT (JSON Web Tokens) ;
- ✧ Implement API request limiting and throttling to prevent abuse ;

✧ Use HTTPS to ensure secure API communication ;

4. Cybersecurity:

✧ Deploy a Web Application Firewall (WAF) to defend against common Web attacks ;

✧ Implementing intrusion detection systems (IDS) and intrusion prevention systems (IPS) ;

✧ Conduct vulnerability scans and penetration tests regularly ;

5. Data Isolation:

✧ Implement strict multi-tenant data isolation to ensure that data of different customers cannot be accessed by each other ;

✧ Use Virtual Private Cloud (VPC) to isolate the network environments of different customers ;

6. Audit log:

✧ Record detailed logs of all critical operations, including user logins, data access, and system configuration changes ;

✧ Implement a security information and event management (SIEM) system to monitor and analyze security events in real time ;

7. Compliance:

✧ Ensure that the system complies with relevant industry standards and regulations, such as GDPR, CCPA, etc .;

✧ Conduct regular security audits and compliance checks ;

8. Security Development Lifecycle:

✧ Implement secure coding practices during the development process ;

✧ Use automated tools for code security analysis ;

✧ Conduct security training regularly to improve the security awareness of the development team ;

9. Backup and Restore:

✧ Implement regular data backup strategies, including offsite backups ;

✧ Develop and test disaster recovery plans ;

10. Third-party integrated security:

✧ Perform security assessments on all third-party integrations ;

✧ Limit the access permissions of third-party applications and grant only the minimum necessary permissions ;

11. Client Security:

✧ Implement client certificate verification and access through https to ensure that only authorized devices can access the system ;

✧ Implement application-level encryption and secure storage in mobile applications ;

12. Subscription Management Security:

✧ Process subscription payments using a secure payment gateway

✧ Implement real-time verification of subscription status to ensure that only users with valid subscriptions can access the service

✧ Provides a secure subscription management interface that allows users to view and manage their subscription status

13. Security Updates and Patch Management:

- ✧ Establish a rapid response mechanism to promptly address newly discovered security vulnerabilities
- ✧ Implement an automated security patch deployment system

By implementing these security measures, Project Zoom management system can provide users with highly secure SaaS services in an Internet environment. These measures not only protect the security of the system and data, but also ensure that user privacy and compliance requirements are met.

五、Multilingualism and internationalization

When designing and developing the "Project Zoom Construction Management App", supporting multiple languages and internationalization (i18n) was an important consideration.

5.1. Supported languages

According to the difficulty of developing an internationalized multilingual system, the system mainly supports languages written from left to right (note: languages written from right to left are not considered in this system, such as Tibetan, Arabic, etc.), and mainly supports the following languages:

- English
- French
- German
- Spanish
- Italian
- Portuguese
- Russian
- Chinese
- Chinese (Traditional)
- Japanese
- Korean

5.2. Support for multiple languages and internationalization

1. **Multi-language support:**

✧ Provide multiple left-to-right language interfaces, allowing users to select the interface language according to their language preferences.

✧ Supports dynamic loading of language packs, making it easy to add and update languages.

2. **Localized formats** :

✧ Supports date, time and number formats for different regions.

✧ Allow users to select and set their own locale.

3. **Currency and unit conversion** :

✧ Supports multiple currencies display and conversion to meet the financial needs of different countries.

✧ Provides switching between metric and imperial units to adapt to measurement standards in different regions.

4. **Cultural and Legal Compatibility:**

✧ Consider cultural differences and legal requirements in different countries and regions.

✧ Provides functionality and compliance support to meet local regulations.

5. **Multiple time zone support:**

✧ Allows users to set and manage different time zones.

✧ Ensure smooth project management and collaboration across time zones.

5.3. The necessity of multilingualism and internationalization

1. **Expanding market coverage** :

- Supporting multiple languages and internationalization can help applications enter more international markets and attract users from different countries.

2. **Improve user experience** :

- Providing localized user interfaces and functions can improve user satisfaction and usage experience.

3. **Enhance competitiveness** :

- In the globalized market, applications that support multiple languages and internationalization are more competitive and can meet the needs of more customers.

4. **Compliance requirements** :

- Different countries and regions have different legal and regulatory requirements. Supporting internationalization helps ensure application compliance.

5. **Fostering global collaboration** :

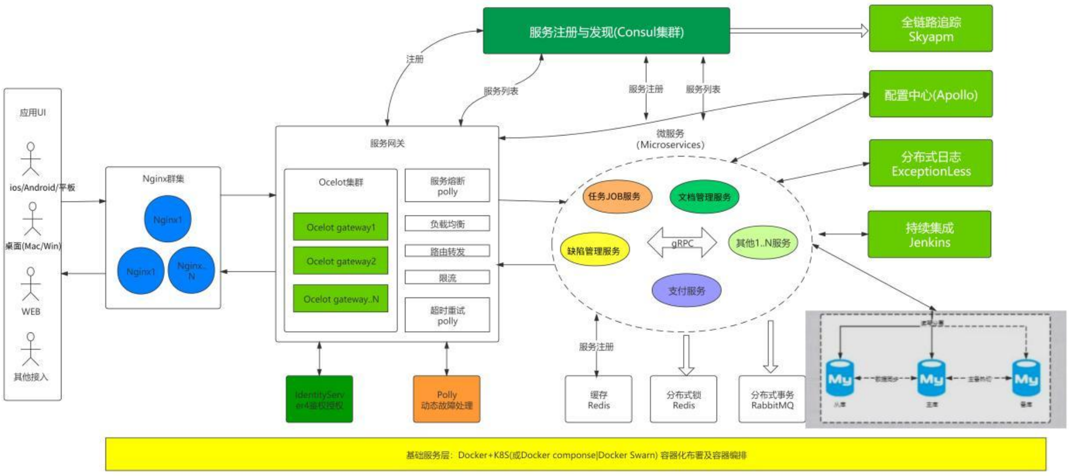
- In international projects, supporting multiple languages and time zones can facilitate collaboration and communication among team members.

By supporting multiple languages and internationalization, Project Zoom can not only meet the needs of local users, but also provide consistent high-quality services around the world, enhancing its competitiveness in the international market.

六、System Architecture

The technical options of this system mainly adopt open source technical architecture, which can ensure low cost and support from the open source community.

6.1 Architecture Diagram



Architecture diagram description:

1. Considering the business growth, the system should have high availability, elastic scalability (automatically expand and shrink with business capabilities), high concurrency, massive data processing, CI/CD

continuous integration, and continuous release capabilities;

2. The system supports Serverless deployment such as Tencent Cloud, Alibaba Cloud , AWS or Azure cloud platforms, as well as self-built server cluster deployment;

3. The system architecture supports any client type, including but not limited to: WEB, APP, etc.;

4. Use Nginx cluster to form a high-availability, high-performance HTTP and reverse proxy WEB service cluster;

5. The system has only one unique entrance: service gateway;

6. The system uses Ocelot cluster as service gateway to provide system load balancing, routing forwarding, service fuse and other functions, and cooperates with Polly to provide dynamic fault handling capabilities;

7. All access needs to be authenticated and authorized by Identity Server 4

- Centralized authentication service for all applications;

- Single sign-on and sign-out for third-party applications;

- Identity Server4 is an authentication and authorization middleware that implements OpenId Connect and OAuth2.0 protocols;

8. Various microservice applications form a service cluster, such as task management service , defect management service, document service , security training service, etc.; each microservice is designed with DDD (Domain Driver Design), and each service communicates through gRPC;

9. In order to support rapid development, deployment and load balancing, use Consul cluster to provide service registration and discovery;

10. In order to quickly analyze and locate anomalies, Skyapm is used to provide full-link tracking function;

11. In order to support rapid configuration, deployment and release, the system uses Apollo as the configuration center; Jenkins is used to provide automated deployment and continuous integration functions;

12. Use distributed logs to quickly locate anomalies and the data center to provide data sources;

13. The system database adopts read-write separation technology to provide high-performance reading and writing of the database;

14. Use K8s+Docker to improve container cluster and container orchestration capabilities. Kubernetes is a complete distributed system support platform with complete cluster management capabilities, multi-level security protection and access mechanisms, multi-tenant application support capabilities, transparent service registration and discovery mechanisms, built-in intelligent load balancers, powerful fault discovery and self-repair capabilities, service rolling upgrades and online capacity expansion capabilities, scalable resource automatic scheduling mechanisms, and multi-granular resource quota management capabilities. Based on Docker technology, it provides containerized

applications with a series of complete functions such as deployment and operation, resource scheduling, service discovery, and dynamic scaling, which improves the convenience of large-scale container cluster management;

6.2 Why do we choose cloud-native development instead of pure Serverless cloud development?

Consideration 1 : Later flexibility and cost! If Serverless development is adopted, the development speed is fast, but the later flexibility is insufficient, and as the business grows, a large amount of fees will be paid to the infrastructure provider;

Consideration 2 : We can build our own infrastructure, so why pay someone else? By choosing cloud-native development, we can gain greater flexibility, control, and portability, which is suitable for complex and diverse application requirements. Although Serverless provides the advantages of rapid development and automatic expansion, in some cases, cloud-native development may be more in line with business needs and technical requirements.

6.3 Database Selection

This system uses two databases:

1、Business database: Choose the relational PostgreSQL database; in big data tables or multi-tenant development, use technologies such as sharding to improve performance;

2、Non-business databases such as logs: Elasticsearch is a non-relational database designed for full-text search and analysis and is suitable for processing large amounts of log data.

6.4 Frontend

WEB front end: using vue+element plus open source framework

Desktop (MacOs+Windows): Developed with Flutter;

iOS/Android/Tablet: Developed with Flutter and adapted to each platform;

Website: Using Nuxt.js+bootstrap+html

6.5 Backend

Adopt the advanced .Net9 web api front-end and back-end separation technology;

6.6 Middleware

Cache: Redis

Message service: RabbitMQ forms a heterogeneous architecture

For other middleware, please refer to the architecture diagram

6.7 Cloud Services

- Deployed on AWS or Azure cloud platforms ;
- Use containerization technologies such as Docker for easy scalability and maintenance ;

七、Development plan and cost

7.1 Development Plan

No	Job Description	time	Remark
1	Requirements Analysis and Design	4 weeks	
2	Front-end development	12 weeks	WEB, desktop (MacOS, Windows), iOS\Anddroid mobile terminal
3	Backend Development	8 weeks	SaaS Service Platform
4	Testing and debugging	4 weeks	
5	Deployment and launch	2 weeks	
	total	30 weeks	

7.2. Development Cost Estimation

No	Job Description	Working days	Man-day price	Price
1	Requirements Analysis and Design	30	1500	45000
2	SaaS service platform development	60	1200	72000
3	Desktop(mac/window)	30	1200	36000
4	Web business development	30	1200	36000
5	Android/IOS/Tablet	60	1200	72000
6	Website development (registration, subscription, etc.)	30	1200	36000
7	Testing, debugging, and launching	30	800	24000
	total			￥321,000.00

Note: System analysts are charged 1,500 yuan per work day, and programmers are charged 1,200 yuan per work day. This is the lower common quotation standard for domestic companies.

7.3 Source code delivery method

After completing the development of "Project Zoom Construction Management App", source code delivery is an important step. The following is the detailed method of source code delivery:

1. Code repository:

✓ Provides access to Git repositories, usually delivered through platforms such as GitHub, Gitee , etc.

✓ Make sure your repository contains all necessary branches, tags, and commit history.

2. Packaging and downloading:

✓ Provide a compressed package of source code (such as ZIP or TAR.GZ format) for easy downloading.

✓ Make sure the zip file contains all code, configuration files, and necessary documentation.

3. document:

✓ Contains a detailed README file explaining how to build, run, and deploy the application.

4. Version Control Systems:

✓ Use Git as the version control system.

✓ Make sure all code has been committed to the master branch and marked as a stable version.

✓ Provide API documentation, architecture diagrams, and other technical documentation.

5. Dependency Management:

✓ Provide a manifest file of dependencies (e.g. package.json, requirements.txt) to ensure that the build environment can be reproduced.

✓ Contains instructions on how to install and manage dependencies.

6. Build and deploy scripts:

✓ Provide automated build and deployment scripts (such as Makefile, Dockerfile, CI/CD configuration files).

✓ Make sure the script is tested and works properly in the target environment.

7. Security and Compliance:

✓ Make sure that the source code does not contain sensitive information (such as API keys, passwords).

- ✓ Provide security audit reports and compliance documentation (such as GDPR compliance).

8. **Support and Maintenance:**

- ✓ Provide contact information for technical support and problem solving after delivery.
- ✓ Describe the subsequent maintenance and update plan.

Through the above methods, the integrity, availability and security of source code delivery are ensured, so that the recipient can smoothly use and maintain the project source code.

八、Risk assessment and response measures

8.1 Data Security Risks

In a SaaS platform like Project Zoom, data security is of paramount importance . Here are the main data security risks we identified and their mitigation strategies:

8.1.1 Unauthorized Access

- ● Implement strong authentication mechanisms, including multi-factor authentication
- ● Use role-based access control (RBAC)

- Regularly review user permissions
- Implement the principle of least privilege

8.1.2 Data Leakage

Risk: Sensitive data may be stolen or accidentally leaked. Countermeasures:

- Encrypt all data in storage and in transit
- Implement data desensitization technology
- Conduct regular security audits and penetration tests
- Develop and rehearse a data breach response plan

8.1.3 Cross-tenant data pollution

Risk: In a multi-tenant environment, the data of one tenant may be leaked to another tenant. Countermeasures:

- Implement strict data isolation mechanism
- Implement tenant isolation at both the application and database layers

8.1.4 Data integrity loss

Risk: Data may be accidentally or maliciously modified, resulting in loss of integrity. Countermeasures:

- Implement data validation and audit logging
- Use digital signatures to ensure data integrity
- Back up data regularly and verify the integrity of the backups
- Implement version control and change tracking

8.1.5 Compliance Risk

Risk: Failure to comply with data protection regulations (such as GDPR, CCPA, etc.) may result in legal and financial risks.

- Conduct regular compliance audits
- Implement data privacy protection measures such as data anonymization
- Provide data subject rights management functions
- Maintain a transparent data processing policy

8.1.6 Insider Threats

Risk: Insiders may abuse their access rights. Countermeasures:

- Implement detailed access logging and monitoring
- Implement a multi-person approval mechanism for sensitive operations
- Conduct regular security awareness training

- Implement employee resignation process and revoke permissions in a timely manner

8.1.7 Third-party integration risks

Risk: Integration with third-party services may introduce new security vulnerabilities. Countermeasures:

- Perform security assessments on all third-party integrations
- Limit third-party access rights and follow the principle of least privilege
- Monitor third-party access activities
- Regularly review and update third-party integrations

8.1.8 Data Loss

Risk: Data loss due to hardware failure, human error, or catastrophic events. Response strategies:

- Implement a comprehensive backup strategy, including offsite backup
- Regularly test your data recovery process
- Use high availability architecture and data replication
- Develop and test a disaster recovery plan

By identifying these risks and implementing appropriate mitigation strategies, we can greatly improve Project Zoom’ s data security, protect our users' sensitive information, and ensure system reliability and compliance.

8.2 User Acceptance Risk

When launching a new system like Project Zoom, user acceptance is a critical factor for success. Here are the main user acceptance risks we identified and their mitigation strategies:

8.2.1 Steep learning curve

Risk: The system is too complex and it is difficult for users to quickly master its usage. Countermeasures:

- Designed with an intuitive, user-friendly interface
- Provide interactive tutorials and guided processes
- Implement progressive feature disclosure to avoid presenting all complex features at once
- Provide detailed user manual and video tutorial

8.2.2 Functions do not meet user needs

Risk: System functions do not match actual user needs. Countermeasures:

- Conduct in-depth user needs research
- Establish a user feedback mechanism to collect and respond to user

opinions in a timely manner

- Provides the ability to customize and extend functionality

8.2.4 Lack of localization support

Risk: The system is not adapted to the language and cultural requirements of different regions.

- Provide multi-language interface and documentation
- Support for different date, time and currency formats
- Consider legal and compliance requirements in different regions
- Provide localized customer support

8.2.5 Resistance to change

Risk: Users are used to the old system and are unwilling to change. Countermeasures:

- Emphasize the clear benefits of the new system
- Providing incentives to encourage early adoption
- Phased rollout to allow users to adapt gradually
- Provide adequate training and support

8.2.5 Insufficient support for mobile devices

Risk: The system experience is not good on mobile devices. Countermeasures:

- Adopt responsive design to ensure a good experience on various devices
- Develop a dedicated mobile app
- Optimize key functions on mobile devices
- Consider offline working mode and adapt to the network conditions of the construction site

By identifying these risks and implementing appropriate mitigation strategies, we can improve user acceptance of Project Zoom, ensure smooth promotion and use of the system, and ultimately achieve the desired business goals.

---- End ----