Form: Layr - High-Level Design Document (Application Creation/Modernization)

I. Project Overview

1. Document Metadata

- (Text Field) Document Title: \[Data Entry Field]

- (Text Field) Document Version: \[Data Entry Field]

- (Date Field) Document Creation Date: \[Data Entry Field]

- (Date Field) Document Last Modified Date: \[Data Entry Field]

- (Text Field) Author(s): \[Data Entry Field]

- (Text Field) Reviewer(s): \[Data Entry Field]

- (Text Area) Project Keywords: \[Data Entry Field] \\*For tagging/search\\*

2. Executive Summary

- (Text Area) Overall Project Summary: \[Data Entry Field] \\*Concise overview\\*

- (Text Area) Key Objectives: \[Data Entry Field] \\*High-level goals\\*

- (Text Area) Target Audience: \[Data Entry Field]

- (Text Area) Key Architectural Decisions: \[Data Entry Field] \\*Summary of choices\\*

- (Text Area) Summary of Expected Benefits: \[Data Entry Field]

3. Scope and Objectives

- (Text Area) Project Description: \[Data Entry Field] \\*Detailed description\\*

- (Text Area) Goals of Project: \[Data Entry Field] \\*Specific, measurable goals\\*

- (Text Area) Scope of Project: \[Data Entry Field] \\*Inclusions and exclusions\\*

- (List) In-Scope Items: \[Repeating Field]

- (List) Out-of-Scope Items: \[Repeating Field]

4. Current State (For Modernization - Skip if New App)

- (Text Area) Description of Existing System: \[Data Entry Field]

- (Text Area) Current Architecture Overview: \[Data Entry Field]

- (List) Technologies Used: \[Repeating Field] \\*With versions if possible\\*

- (List) Pain Points / Limitations: \[Repeating Field] \\*Specific issues\\*

- (Text Area) Current Data Model Overview: \[Data Entry Field]

- (Text Area) Current Integration Landscape: \[Data Entry Field]

5. Target State

- (Text Area) Vision for New/Modernized Application: \[Data Entry Field] \\*Desired future\\*

- (Text Area) Key Success Metrics: \[Data Entry Field] \\*How success is measured\\*

- (Text Area) Target User Personas: \[Data Entry Field]

II. Architecture Design

6. Architectural Style

- (Dropdown) Chosen Style: \[Microservices, Monolithic, Serverless, Event-Driven, etc.]

- (Text Area) Rationale for Choice: \[Data Entry Field] \\*Why this style?\\*

- (Checkbox Group) Architectural Drivers: \[Scalability, Performance, Maintainability, Cost, Time-to-Market, Security, Reliability] \\*Prioritized list\\*

- (Text Area) Diagram (Reference): \[File Upload or Link to Diagram] \\*UML, etc.\\*

7. Domain-Driven Design (DDD)

- (Text Area) Core Domains and Subdomains: \[Data Entry Field]

- (Text Area) Bounded Contexts: \[Data Entry Field] \\*With responsibilities\\*

- (Text Area) Domain Model Highlights: \[Data Entry Field] \\*Key entities, relationships\\*

- (Diagram (Reference): \[File Upload or Link to Diagram] \\*Context Map, etc.\\*

8. Microservices Architecture (If Applicable)

- (List - Section Group - Repeatable)

- Microservice Name: \[Text Field]

- Microservice Responsibility: \[Text Area]

- APIs Exposed: \[Text Area] \\*With specifications if available\\*

- Data Owned: \[Text Area] \\*Data sovereignty\\*

- Communication Pattern: \[Dropdown - REST, gRPC, Message Queue (Kafka, etc.), Sync/Async]

- Technology Choice Rationale: \[Text Area] \\*Language, framework, etc.\\*

- Dependencies (Other Services): \[List]

- Diagram (Reference): \[File Upload or Link] \\*Service Map, etc.\\*

9. API Design

- (Dropdown) API Style: \[RESTful, GraphQL, gRPC]

- (Text Area) API Versioning Strategy: \[Data Entry Field]

- (Text Area) Authentication / Authorization: \[Data Entry Field] \\*Protocols, etc.\\*

- (Text Area) Rate Limiting / Throttling: \[Data Entry Field]

- (Text Area) API Documentation Strategy: \[Data Entry Field] \\*Swagger, etc.\\*

- (Text Area) Error Handling Strategy: \[Data Entry Field]

10. Data Architecture

- (Dropdown - Multi-Select) Data Storage Technologies: \[Relational (PostgreSQL, etc.), NoSQL (MongoDB, Cassandra), Cache (Redis), Object Storage (S3), Graph (Neo4j), etc.]

- (Text Area) Data Model Overview: \[Data Entry Field] \\*Logical model summary\\*

- (Text Area) Data Migration Strategy (Modernization): \[Text Area] \\*Detailed plan\\*

- (Text Area) Data Consistency Requirements: \[Data Entry Field] \\*ACID, eventual consistency\\*

- (Text Area) Data Security and Privacy: \[Data Entry Field] \\*Encryption, masking, etc.\\*

- (Diagram (Reference): \[File Upload or Link] \\*ERD, etc.\\*

11. Data Flow and Integration

- (Text Area) System Boundary Diagram (Reference): \[File Upload or Link]

- (List - Section Group - Repeatable) Integration Point:

- System Name: \[Text Field]

- Purpose of Integration: \[Text Area]

- Data Exchanged: \[Text Area]

- Integration Pattern: \[Dropdown - API (REST, etc.), Messaging (Kafka), File Transfer, Shared Database]

- Technology Used: \[Text Field]

- Security Considerations: \[Text Area]

- Diagram (Reference): \[File Upload or Link] \\*Sequence Diagram, etc.\\*

III. Functional and Non-Functional Requirements

12. Use Cases and Requirements

- (Section Group - Repeatable)

- Requirement ID: \[Text Field] \\*Unique ID\\*

- Requirement Name: \[Text Field] \\*Short, descriptive name\\*

- Description: \[Text Area] \\*Detailed description\\*

- Priority: \[Dropdown - High, Medium, Low]

- User Stories (Links): \[Repeating Field] \\*Link to user story tickets\\*

- Use Case Diagram (Reference): \[File Upload or Link]

- (Checkbox) Implemented:

13. Non-Functional Requirements

- (Section Group - Repeatable)

- Requirement ID: \[Text Field] \\*Unique ID\\*

- Requirement Name: \[Text Field] \\*Short, descriptive name\\*

- Description: \[Text Area] \\*Detailed description\\*

- Category: \[Dropdown - Performance, Scalability, Reliability, Security, Usability, Accessibility, Maintainability]

- Target Value: \[Text Field] \\*Specific, measurable target\\*

- Measurement Method: \[Text Area] \\*How to verify\\*

- (Checkbox) Met:

IV. Technology and Implementation

14. Technology Stack

- (Table - Repeatable Rows)

- Technology Area: \[Dropdown - Language, Framework, Database, Cloud Service, etc.]

- Technology: \[Text Field] \\*e.g., Java, Spring Boot, AWS S3\\*

- Version: \[Text Field] \\*Specific version\\*

- Rationale: \[Text Area] \\*Why this choice?\\*

15. Component Design

- (Section Group - Repeatable)

- Component Name: \[Text Field]

- Component Type: \[Dropdown - Service, Library, UI Component, etc.]

- Responsibility: \[Text Area] \\*What does it do?\\*

- Inputs/Outputs: \[Text Area]

- Dependencies: \[List]

- Diagram (Reference): \[File Upload or Link] \\*Class Diagram, etc.\\*

16. Integration Strategy

- (Text Area) Internal System Integrations: \[Data Entry Field]

- (Text Area) External System Integrations: \[Data Entry Field]

- (Dropdown - Multi-Select) Integration Patterns: \[API (REST, etc.), Messaging (Kafka), File Transfer, Shared Database, Events]

17. Deployment Strategy

- (Dropdown) Deployment Model: \[Cloud (AWS, Azure, GCP), On-Premise, Hybrid]

- (Text Area) Infrastructure as Code (IaC) Tools: \[Data Entry Field] \\*Terraform, etc.\\*

- (Text Area) CI/CD Pipeline: \[Data Entry Field] \\*Tools, process\\*

- (Text Area) Containerization and Orchestration: \[Data Entry Field] \\*Docker, Kubernetes\\*

- (Text Area) Monitoring and Logging: \[Data Entry Field] \\*Tools, strategy\\*

- (Text Area) Release Management Strategy: \[Data Entry Field] \\*Frequency, process\\*

V. Cross-Cutting Concerns

18. AI/ML Considerations (If Applicable)

- (Text Area) AI/ML Use Cases: \[Data Entry Field] \\*Where is AI/ML used?\\*

- (Text Area) Model Training and Deployment: \[Data Entry Field] \\*Process, tools\\*

- (Text Area) Data Requirements for AI/ML: \[Data Entry Field] \\*Volume, type, quality\\*

- (Text Area) AI/ML Model Monitoring: \[Data Entry Field] \\*Drift detection, etc.\\*

- (Text Area) Ethical Considerations: \[Data Entry Field] \\*Bias, fairness, etc.\\*

19. Security and Access Controls

- (Text Area) Authentication and Authorization: \[Data Entry Field] \\*Protocols, identity providers\\*

- (Text Area) Data Encryption: \[Data Entry Field] \\*At rest, in transit\\*

- (Text Area) Vulnerability Management: \[Data Entry Field] \\*Scanning, patching\\*

- (Text Area) Compliance (GDPR, etc.): \[Data Entry Field] \\*Specific regulations\\*

- (Text Area) Security Testing: \[Data Entry Field] \\*SAST, DAST, etc.\\*

20. Scalability and Performance

- (Text Area) Scalability Strategy: \[Data Entry Field] \\*Horizontal, vertical scaling\\*

- (Text Area) Performance Optimization: \[Data Entry Field] \\*Techniques, tools\\*

- (Text Area) Caching Strategy: \[Data Entry Field] \\*Levels, technologies\\*

- (Text Area) Load Testing Strategy: \[Data Entry Field]

21. Reliability and Availability

- (Text Area) High Availability Design: \[Data Entry Field] \\*Redundancy, failover\\*

- (Text Area) Disaster Recovery Plan: \[Data Entry Field] \\*RPO, RTO\\*

- (Text Area) Monitoring and Alerting: \[Data Entry Field] \\*Tools, thresholds\\*

- (Text Area) Backup and Restore Strategy: \[Data Entry Field]

22. Resiliency and Disaster Recovery

- (Text Area) Resiliency Patterns: \[Circuit Breaker, Retry, etc.]

- (Text Area) Failover Strategy: \[Active-Active, Active-Passive]

- (Text Area) Disaster Recovery Sites: \[Primary, Secondary]

- (Text Area) Recovery Time Objectives (RTO): \[Data Entry Field]

- (Text Area) Recovery Point Objectives (RPO): \[Data Entry Field]

- (Text Area) Disaster Recovery Testing: \[Frequency, Types of Tests]

23. Compliance and Regulatory

- (Text Area) Data Privacy Regulations: \[GDPR, CCPA, etc.]

- (Text Area) Industry-Specific Regulations: \[HIPAA, PCI DSS, etc.]

- (Text Area) Accessibility Standards: \[WCAG]

- (Text Area) Audit and Logging Requirements:

24. Risks and Mitigations

- (Table - Repeatable Rows)

- Risk Description: \[Text Area]

- Likelihood: \[Dropdown - High, Medium, Low]

- Impact: \[Dropdown - High, Medium, Low]

- Mitigation Strategy: \[Text Area]

- Contingency Plan: \[Text Area]

- Owner: \[Text Field]

VI. Conclusion and Next Steps

25. Summary

- (Text Area) Project Summary: \[Data Entry Field] \\*Concise recap\\*

26. Next Steps

- (List) Action Items: \[Repeating Field] \\*Specific tasks\\*

- (List) Open Issues: \[Repeating Field] \\*Unresolved questions\\*

27. References and Appendices

- (List) References: \[Repeating Field] \\*Links to other docs\\*

- (List) Appendices: \[Repeating Field] \\*Supporting material - diagrams, etc.\\*

Application Description

An app – a modern web-based platform designed to streamline the onboarding experience for enterprise users. The application provides centralized access to role-specific tasks, documentation, compliance workflows, and collaboration tools, enhancing visibility and reducing time-to-productivity for new hires across distributed teams.

Use Case

The application serves large enterprises that onboard hundreds to thousands of employees annually, particularly across multiple geographies and departments. HR teams and IT administrators use the platform to configure onboarding workflows, while new hires interact with personalized dashboards to complete required tasks. The app ensures consistent onboarding quality, integrates with existing HR and identity systems, and provides analytical insights into onboarding bottlenecks.  
  
Example Use Cases:  
- Automating document verification and provisioning access to internal tools.  
- Guiding new employees through compliance training.  
- Coordinating team introductions and manager-led check-ins.  
- Providing status visibility for HR and hiring managers.

Baseline Systems

As this is a new application being developed, there are no direct legacy systems it is replacing. However, the current onboarding processes rely on the following fragmented tools:  
- Manual email communication chains.  
- SharePoint-hosted checklists and documents.  
- Third-party task management tools like Asana or Trello.  
- Spreadsheets for tracking onboarding status.  
- HR systems such as Workday and SuccessFactors (for employee record management).  
  
These systems lack integration, visibility, and automation, leading to delays, inconsistent experiences, and increased HR effort.

Requirements

Functional Requirements:

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Requirement Name | Description | Priority |
| FR-001 | Onboarding Task Engine | Ability to configure dynamic onboarding task flows per department or role. | High |
| FR-002 | Document Upload and Signing | Support for uploading, previewing, and e-signing onboarding documents. | High |
| FR-003 | Role-Based Access Control | Users only see data and actions relevant to their role (HR, IT, new hire). | High |
| FR-004 | Notification Engine | Email and in-app reminders for upcoming or overdue onboarding tasks. | Medium |
| FR-005 | Analytics Dashboard | HR users can view onboarding completion trends and task bottlenecks. | Medium |

Non-Functional Requirements:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Requirement ID | Requirement Name | Description | Category | Target Value | Measurement Method |
| NFR-001 | System Availability | The application must be available 99.9% of the time. | Reliability | 99.9% uptime | Monitoring logs |
| NFR-002 | Response Time | UI actions should respond within 300ms under normal load. | Performance | ≤ 300ms | Synthetic testing |
| NFR-003 | Data Privacy Compliance | Must comply with GDPR and CCPA regulations. | Compliance | Full adherence | Compliance audit |
| NFR-004 | Multi-browser Support | App should work seamlessly on latest versions of Chrome, Edge, Firefox. | Usability | 100% test pass | Browser compatibility tests |