ArchReview Studio System Design Document

Executive Summary

Project Type: Web Application

A centralized web application for reviewing and validating system designs before production rollout.

Core Features

- Design review workflows

- diagram rendering engine

- role-based access

- audit logs

- design comparison

- artifact generation

Constraints

- Performance

- Cost

- Security

- Compliance

- Time-to-market

User Scale: 100-1000 MAU, 50 concurrent users

System Overview

Functional Goals

- Enable teams to convert technical specifications into visual diagrams

- Generate review-ready artifacts for stakeholders

- Provide role-based access for different user personas

Non-Functional Requirements

- Ensure high performance for concurrent users

- Meet SOC 2 compliance standards

- Provide a scalable and reliable architecture

Primary User Personas

- Project Managers

- System Architects

- Quality Assurance Engineers

Architecture Design

The ArchReview Studio application will be built using a microservices architecture.

System Components

Component	Responsibility	Technologies	Interfaces
API Gateway	Handle incoming requests and route them to the appropriate microservice	NGINX, Kubernetes	RESTful API
Design Service	Handle design review workflows and artifact generation	Node.js, Express.js	RESTful API
Database	Store user data and design artifacts	MySQL	SQL

Database Design

Database Type: Relational

High availability and performance

Table: users

Column	Type	Nullable	Description
id	int	False	Unique user ID
username	varchar	False	User username
email	varchar	False	User email
password	varchar	False	User password

Table: designs

Column	Type	Nullable	Description
id	int	False	Unique design ID
name	varchar	False	Design name
description	text	False	Design description
artifact	blob	False	Design artifact

Cost Estimation

Cost Item	Monthly Cost	Rationale
Infrastructure	$1000.0	Estimated monthly cost for infrastructure
Development	$5000.0	Estimated monthly cost for development

Testing & QA Strategy

Unit testing – All components

Integration testing – All integrations

End-to-end testing – All user flows

Load and stress testing – All components

Security testing – All components

Appendices

Glossary

API Gateway: A component that handles incoming requests and routes them to the appropriate microservice

Design Service: A component that handles design review workflows and artifact generation

References

https://www.nginx.com/resources/wiki/start/topics/tutorials/configuring-nginx/

https://kubernetes.io/docs/home/

System Architecture & Diagrams

System Architecture Diagram (Mermaid Code):

```
flowchart LR
User -->| Requests | API_Gateway
API_Gateway -->| Routes | Design_Service
Design_Service -->| Reads/Writes | Database
Database --> Design_Service
Design_Service --> API_Gateway
API_Gateway --> User
```

User Flow Diagram (Mermaid Code):

```
flowchart TD
Start --> Login
Login --> Dashboard
Dashboard --> ViewDesigns
ViewDesigns --> End
```

Database ER Diagram (Mermaid Code):

```
erDiagram
users {
int id
varchar username
varchar email
varchar password
}
designs {
int id
varchar name
text description
blob artifact
}
users ||--o{ designs : creates
```