System Design Document (SDD)

Template

State University of Zanzibar (SUZA)

BSc Computer Science

Document Information

Project Name	[Enter Project Name]
Version	1.0
Date	[Enter Date]
Author(s)	[Team Members]

Contents

1 Introduction

1.1 Purpose

[Describe the purpose of this design document.]

1.2 Scope

[Define what aspects of the system this document covers.]

1.3 Design Goals

- Scalability
- Maintainability
- Security
- Performance
- Usability

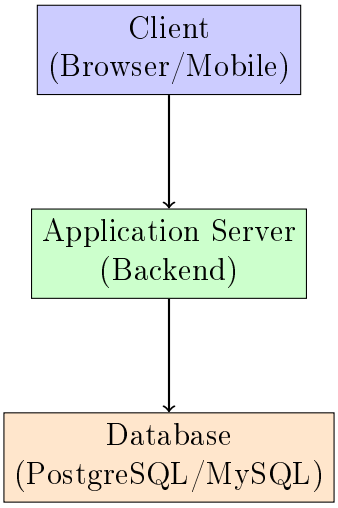
2 System Architecture

2.1 Architecture Overview

[Provide a high-level description of the system architecture.]

2.2 Architecture Diagram

[Insert or describe your system architecture diagram]



2.3 Architecture Pattern

[Describe the architectural pattern used]

- **Pattern:** [e.g., MVC, Microservices, Layered]
- **Justification:** [Why this pattern was chosen]

3 Technology Stack

3.1 Frontend Technologies

Technology	Version	Purpose
React / Vue / Angular	[version]	UI Framework
HTML5	5	Markup
CSS3 / Tailwind	[version]	Styling
JavaScript / TypeScript	[version]	Client Logic

3.2 Backend Technologies

Technology	Version	Purpose
Node.js / Python / Java	[version]	Server Runtime
Express / Django / Spring	[version]	Web Framework

3.3 Database

Technology	Type	Purpose
PostgreSQL / MySQL	Relational	Primary Database
MongoDB	NoSQL	[if applicable]
Redis	In-Memory	Caching

3.4 DevOps and Deployment

Tool	Purpose
Git / GitHub	Version Control
Docker	Containerization
AWS / Heroku / Vercel	Hosting

4 Database Design

4.1 Entity Relationship Diagram

[Include or describe your ERD]

4.2 Database Schema

Table: users

Column	Type	Constraints	Descr
id	INT	PRIMARY KEY, AUTO_INCREMENT	User I
username	VARCHAR(50)	UNIQUE, NOT NULL	Userna
email	VARCHAR(100)	UNIQUE, NOT NULL	Email
password_hash	VARCHAR(255)	NOT NULL	Hashe
created_at	TIMESTAMP	DEFAULT NOW()	Creati

Table: [table_name] [Repeat format for each table]

4.3 Database Relationships

- users → orders (One-to-Many)
- orders → products (Many-to-Many)

more relationships

5 API Design

5.1 API Overview

[Describe the API architecture - REST, GraphQL, etc.]

5.2 API Endpoints

Authentication Endpoints

Method	Endpoint	Description	Auth
POST	/api/auth/register	Register new user	No
POST	/api/auth/login	User login	No
POST	/api/auth/logout	User logout	Yes

Resource Endpoints

Method	Endpoint	Description	Auth
GET	/api/users	Get all users	Yes
GET	/api/users/:id	Get user by ID	Yes
PUT	/api/users/:id	Update user	Yes
DELETE	/api/users/:id	Delete user	Yes

5.3 Request/Response Examples

POST /api/auth/register

Request:

```
{
    "username": "john_doe",
    "email": "john@example.com",
    "password": "securepassword123"
}
```

Response (Success - 201):

```
{
    "success": true,
    "message": "User registered successfully",
    "data": {
        "id": 1,
        "username": "john_doe",
        "email": "john@example.com"
    }
}
```

6 User Interface Design

6.1 UI/UX Principles

- Consistency across all pages
- Mobile-first responsive design
- Clear navigation and feedback
- Accessibility compliance

6.2 Screen Layouts

[Include wireframes or mockups for key screens]

1. **Login Page:** [Description]
2. **Dashboard:** [Description]
3. **Main Feature Page:** [Description]
4. **Settings:** [Description]

6.3 Navigation Flow

[Describe how users navigate through the application]

7 Security Design

7.1 Authentication

- Method: JWT / Session-based
- Token expiration: [time]
- Refresh token mechanism: [yes/no]

7.2 Authorization

- Role-based access control (RBAC)
- User roles: Admin, User, Guest

7.3 Data Protection

- Passwords hashed using bcrypt
- HTTPS for all communications
- Input validation and sanitization
- SQL injection prevention
- XSS protection

8 Component Design

8.1 Component 1: [Component Name]

- **Purpose:** [What it does]
- **Responsibilities:** [List of responsibilities]
- **Dependencies:** [What it depends on]
- **Interface:** [How other components interact with it]

9 Deployment Architecture

9.1 Deployment Diagram

[Describe or include deployment diagram]

9.2 Environment Configuration

- **Development:** localhost
- **Staging:** [staging URL]
- **Production:** [production URL]

10 Appendices

10.1 Glossary

[Technical terms and definitions]

10.2 Revision History

Version	Date	Author	Changes
1.0	[date]	[name]	Initial version