# Systemdokumentation – Lunchify

## 1. Architekturübersicht

Die Anwendung basiert auf einer JavaFX-Client-Architektur mit REST-Kommunikation zu Supabase.

**Architekturdiagramm:**

Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

## 2. Technologien

- Java 17  
- JavaFX 21  
- PostgreSQL (über Supabase gehostet)  
- Supabase Storage (für Datei-Uploads)  
- REST-API Upload-Service mit HttpURLConnection  
- FXML zur UI-Definition  
- CSS für Styling  
- JUnit5 für Unit Tests

## 3. Aufbau & Pakete

Die Applikation ist modular aufgebaut:

- `controller`: Alle JavaFX Controllerklassen

- `model`: Datenmodelle für User, Invoice, etc.

- `service`/`util`: Upload-Service, DB-Verbindung

- `view`: FXML-Dateien für UI-Screens

- `css`: Stylesheets für UI

## 4. Supabase-Integration

Supabase dient als Backend für Daten und Dateiablage:

**Beispiele:**

- Tabelle `benutzer`: Nutzerverwaltung

- Tabelle `rechnung`: Rechnungen

- Storage Bucket `rechnung`: Dateiablage

**Gesamte Übersicht des Datenmodells:**

Ein Bild, das Screenshot, Text, Design enthält.

KI-generierte Inhalte können fehlerhaft sein.

## 5. Sicherheit

- Rollenbasiertes Login (user, admin)  
- Passwort-Hashing mit BCrypt  
- Zugriffsschutz über Supabase Row-Level-Security (RLS)  
- Upload nur authentifizierten Usern erlaubt

## 6. Testkonzept

- Unit Tests mit JUnit5 für Kernlogik (z. B. Validierungen, Reimbursement Calculation)  
- GUI-Tests manuell anhand der Testfälle (siehe Testfall-Dokumentation)  
- Datenbanktests lokal mit Test-Daten möglich