

# Benutzerdokumentation Lunchify

Version 1.3 • Stand: Mai 2025

## Inhalt

1.	Einleitung .....	2
1.1.	Zweck.....	2
2.	Systemübersicht .....	2
3.	Installation & Inbetriebnahme .....	2
4.	Erstkonfiguration & Benutzerverwaltung .....	2
5.	Funktionale Bereiche (User) .....	3
5.1.	Login & Passwort .....	3
5.2.	Dashboard & Navigation.....	3
5.3.	Header: ADD NEW + (Upload), Username .....	3
5.4.	Upload & Reimbursement .....	3
5.5.	Request History .....	3
5.6.	Starred View .....	3
5.7.	Settings .....	4
6.	Funktionale Bereiche (Admin) .....	4
6.1.	Admin Dashboard .....	4
6.2.	User Management .....	4
6.3.	Settings (Admin) .....	4
7.	Typische Workflows .....	5

# 1. Einleitung

## 1.1. Zweck

Lunchify ist eine Desktop-App zur Erfassung, Verwaltung und Auswertung von Verpflegungsbelegen. Mitarbeitende können Rechnungen hochladen, ihren Status verfolgen und Favoriten markieren. Administrator\*innen verwalten Benutzer, genehmigen oder lehnen Anträge ab und gewinnen Statistiken.

### Zielgruppe

- **User:** Mitarbeitende ohne Admin-Rechte
- **Admin:** Nutzer mit erweiterten Rechten (User- und Antragsmanagement)

## 2. Systemübersicht

- **Client:** JavaFX-App
- **Backend:** Supabase (Postgres DB & Storage)
- **Kommunikation:** JDBC für DB, HTTPS für Storage
- **Sicherheit:** BCrypt-Hashing, RLS-Policies (Row-Level Security)

## 3. Installation & Inbetriebnahme

### 1. Voraussetzungen

- JRE 17+
- JavaFX-SDK im Modulpfad

### 2. Download & Start

- Executable JAR: Doppelklick

### 3. DB- & Storage-Keys

- Per ENV-Variablen SUPABASE\_URL/SUPABASE\_KEY oder in application.properties

## 4. Erstkonfiguration & Benutzerverwaltung

- **Admin-Anlage:** Admin legt neue User an (Name, E-Mail, Rolle), Default-Passwort default123, Flag must\_change\_password = TRUE.
- **Erstlogin:** Change-Password-Dialog zwingend, bis Flag gelöscht ist.

## 5. Funktionale Bereiche (User)

### 5.1. Login & Passwort

- **LoginView:** E-Mail-/Passwort-Felder, Forgot-Password-Link
- **Forgot Password:** Eingabe E-Mail → Hinweisdialog
- **ChangePasswordView:** Zwei Felder (Neu, Bestätigen), Validierung & BCrypt-Hashing

### 5.2. Dashboard & Navigation

- **Sidebar:** Logo, Home, Starred, Requests, Settings, Log out

### 5.3. Header: ADD NEW + (Upload), Username

### 5.4. Upload & Reimbursement

- **AddInvoiceView**
  - Drag-Drop/Datei-Wahl
  - Kategorie (Restaurant/Supermarkt)
  - Betrag (€)
  - Echtzeit-Berechnung Erstattung (max. 3 €/2.5 €)
  - Submit → Storage + DB

### 5.5. Request History

- **RequestHistoryView:** Tabelle (Datum, Invoice, Erstattung, Kategorie, Status)
- **Kontextmenü** (Rechtsklick): Star/Unstar Rechnung
- **Doppelklick:** Bildvorschau mit Zoom/Scroll
- **Buttons:** Edit/Delete nur für editierbare Rechnungen

### 5.6. Starred View

- **StarredView:** Auflistung aller favorisierten Rechnungen

## 5.7. Settings

- **SettingsView**
  - Change Password → ChangePasswordView
  - Update Profile → UpdateProfileView (Name, E-Mail ändern)
  - Notification Preferences → NotificationView (E-Mail-Benachrichtigung bei Statusänderung)

## 6. Funktionale Bereiche (Admin)

### 6.1. Admin Dashboard

- **Übersicht:** Statistiken & Filter
- **Tabelle:** Alle Rechnungen aller Nutzer (inkl. Nutzer-E-Mail)
- **Aktionen:**
  - Delete Invoice
  - Reject Invoice (Statuswechsel)
  - Edit Invoice (Betrag anpassen)
  - Export: CSV, JSON, XML

### 6.2. User Management

- **UserManagementView**
  - Tabelle (Name, E-Mail, Rolle)
  - Buttons: Add User, Delete User
- **AddUserView**
  - Felder: Name, E-Mail, Rolle (ChoiceBox admin/user)
  - Submit → Default-Passwort + Flag in DB

### 6.3. Settings (Admin)

- **Reimbursement Settings:** Grenzen (Restaurant: 3 €, Supermarkt: 2.5 €) anpassen
- Persistierung in Tabelle settings oder als ENV-Variablen

## 7. Typische Workflows

Use Case	Schritte
<b>Rechnung einreichen</b>	Home → ADD NEW → Datei wählen → Kategorie + Betrag → Submit
<b>Rechnung favorisieren</b>	Requests → Rechtsklick → Star Invoice → Starred → Home
<b>Passwort zurücksetzen</b>	Login → Forgot Password → E-Mail eingeben → Hinweisdialog
<b>User anlegen (Admin)</b>	Manage Users → Add User → Felder füllen → Add → Neuer User erscheint
<b>Antrag genehmigen (Admin)</b>	Manage Requests → Select → Edit/Reject → Status aktualisiert