

Detaillierte Systemdokumentation inklusive der Systemarchitektur. Siehe Projektrichtlinien.

1. Einleitung

1.1. Projektziel

EazyBooks ist eine Finanzmanagement-App, die es Benutzern ermöglicht, ihre Budgets zu verwalten, Transaktionen zu verfolgen und Berichte zu generieren. Die Anwendung wurde mit Angular entwickelt und nutzt Firebase Firestore als Datenbank und Firebase Hosting für das Deployment.

1.2. Anwendungsübersicht

Die Anwendung bietet eine benutzerfreundliche Oberfläche für die Verwaltung von Finanzdaten.

Hauptfunktionen sind:

- Benutzerregistrierung und -anmeldung (mit Firebase Authentication)
- Budgetverwaltung
- Transaktionsverwaltung
- Berichtserstellung
- Verwaltung von Kategorien
- Benutzereinstellungen
- PWA-Unterstützung für Offline-Funktionalität (wird ausgebaut in PRO-4)

2. Architekturübersicht

2.1. Technologiestack

- Frontend: Angular (TypeScript, HTML, CSS)
- Backend: Firebase Firestore
- Hosting: Firebase Hosting
- PWA-Support: Angular Service Worker

2.2. Architekturstil

- Client-Server_Architektur: Die Anwendung ist eine Single Page Application mit Angular als Client und Firebase als Backend.
- Komponentenbasierte Architektur: Die Anwendung ist modular aufgebaut. Jede Funktion ist in wiederverwendbare Komponenten aufgeteilt.

3. Komponenten und Module

3.1. App Module

Das AppModule ist der Einstiegspunkt der Anwendung. Es importiert alle notwendigen Angular-Module und registriert die Dienste sowie die Anwendungsrouten.

3.2. Routing Module

Das Routing-Modul definiert die Navigationslogik der Anwendung. Routen werden den jeweiligen Komponenten zugewiesen, um eine reibungslose Navigation zwischen den Seiten zu ermöglichen.

- Beispielrouten: /dashboard, /budgets

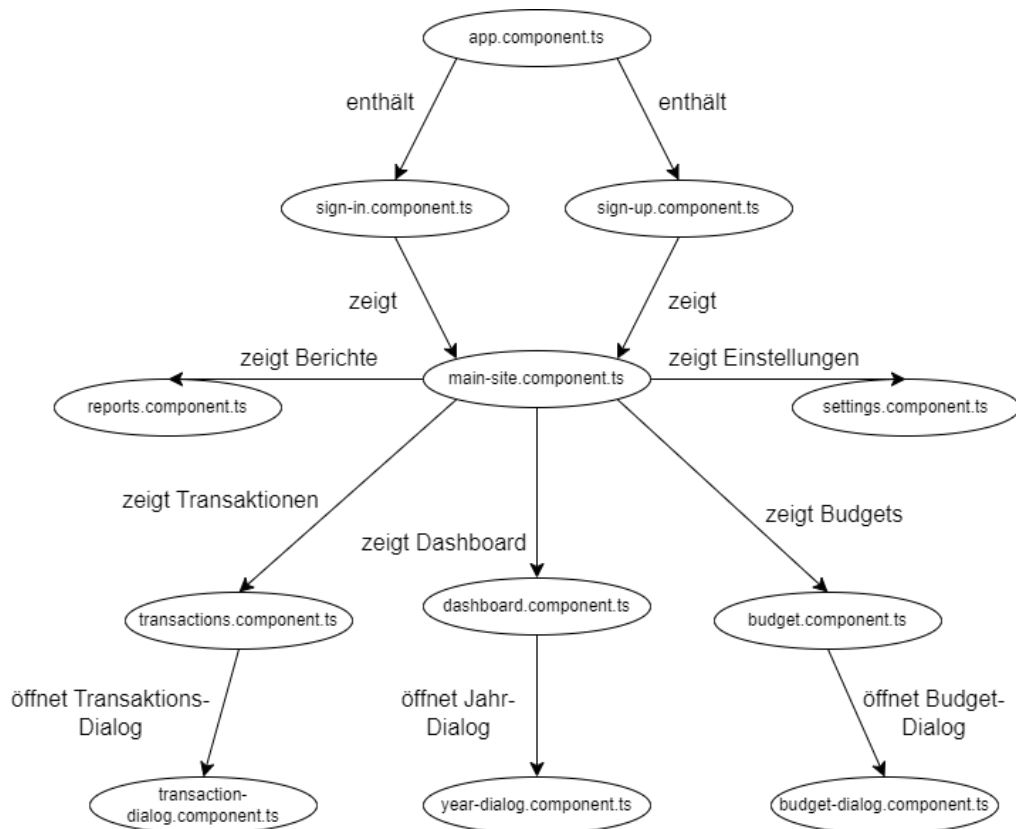
4. Klassendiagramm

4.1. Hauptkomponenten

- app.component.ts
- main-site.component.ts
- dashboard.component.ts
- transactions.component.ts
- budget.component.ts
- reports.component.ts
- settings.component.ts

4.2. Dialog-Komponenten

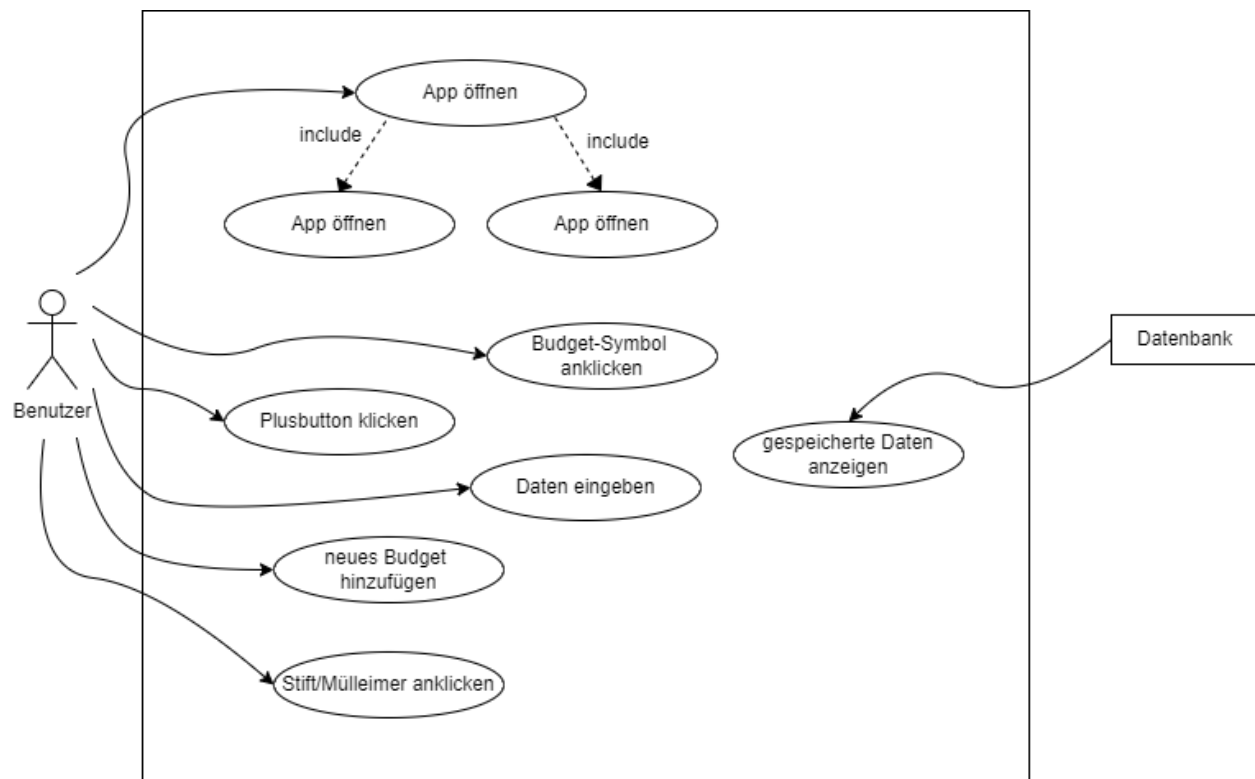
- budget-dialog.component.ts
- transaction-dialog.component.ts
- year-dialog.component.ts



5. Use-Case-Diagramme

5.1. Haupt-Use-Case - Ein neues Budget hinzufügen und bearbeiten

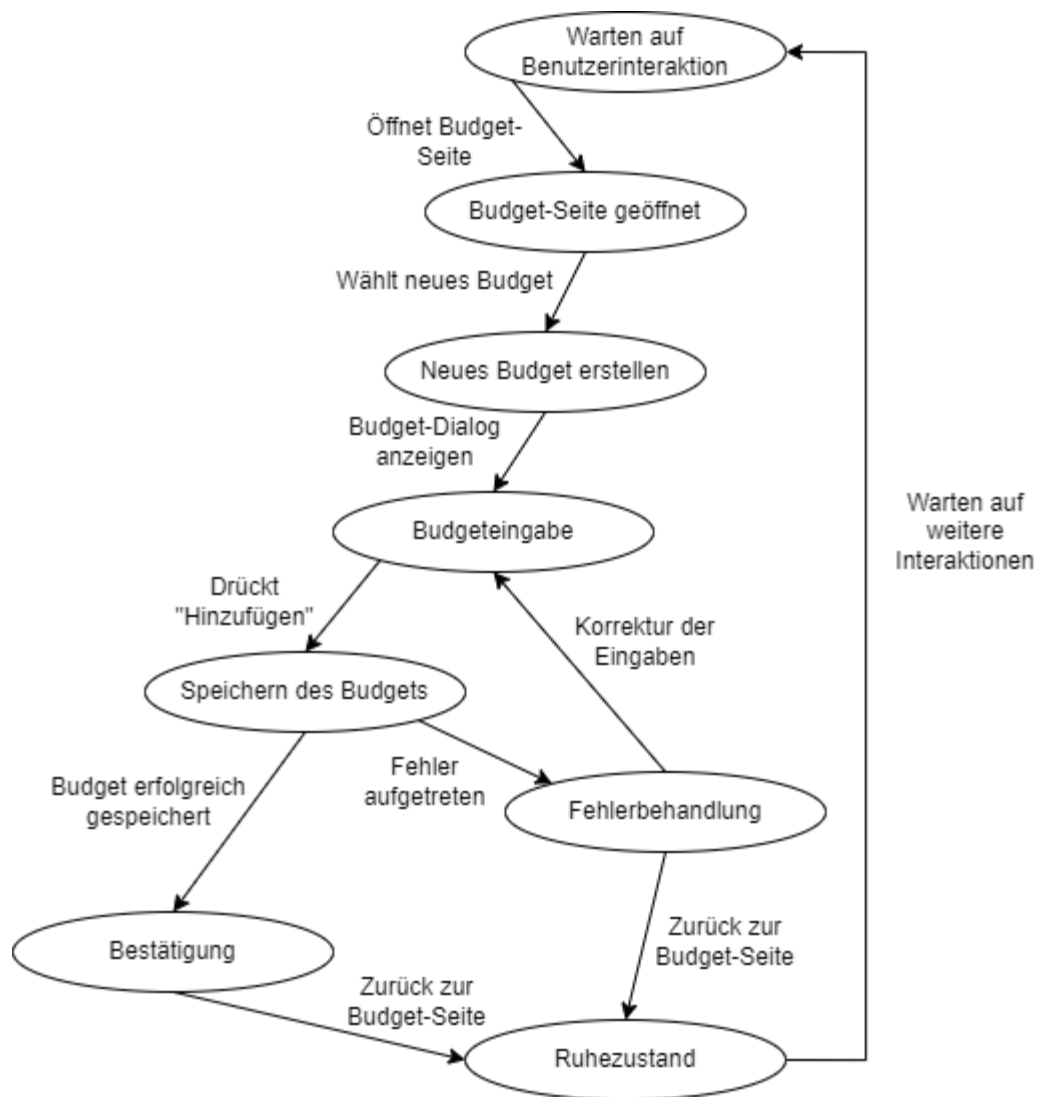
- Man öffnet die App und meldet sich an / registriert sich.
- Dann klickt man auf das Budget-Symbol, um auf das Budgetfenster zu gelangen.
- Man klickt unten rechts auf das Plus, um ein neues Budget zu erstellen.
- Ein neues Fenster öffnet sich, wo man alle erforderlichen Daten eingibt.
- Dann klickt man auf Hinzufügen, um die Daten in der Datenbank zu speichern.
- Im Budgetfenster werden dann alle Budget aus der Datenbank in der jeweiligen Tabelle geladen.
- Möchte man den Eintrag bearbeiten, dann klickt man auf den Stift.
- Möchte man den Eintrag löschen, dann klickt man auf den Mülleimer.



6. Zustandsdiagramm

Zustandsdiagramm für die Verwaltung von Budgets:

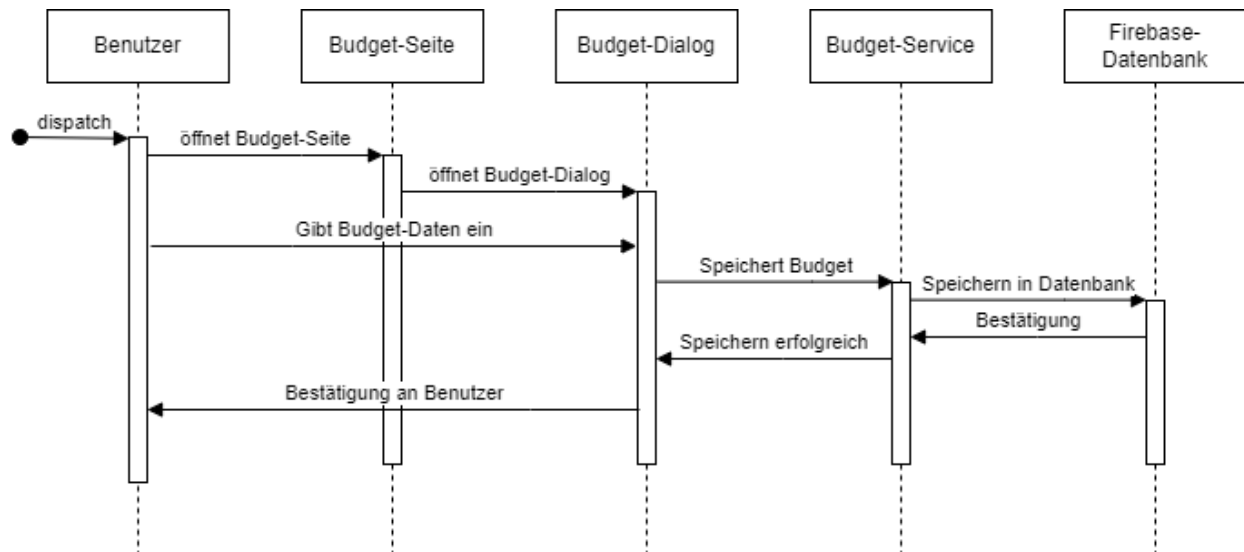
1. Die EazyBooks-App verwaltet Budgets, und der Benutzer interagiert dabei in verschiedenen Zuständen. Das Zustandsdiagramm zeigt den Ablauf bei der Erstellung und Verwaltung von Budgets:
2. Warten auf Benutzerinteraktion: Die App befindet sich im Ruhezustand und wartet darauf, dass der Benutzer eine Aktion ausführt.
3. Budget-Seite geöffnet: Wenn der Benutzer die Budget-Seite öffnet, wird die Ansicht zur Budgetverwaltung angezeigt. Der Benutzer hat die Möglichkeit, bestehende Budgets anzusehen oder ein neues Budget zu erstellen.
4. Neues Budget erstellen: Der Benutzer wählt die Option, ein neues Budget zu erstellen. Die App wechselt in den Zustand, in dem der Budget-Dialog angezeigt wird.
5. Budgeteingabe: Im Budget-Dialog gibt der Benutzer die notwendigen Informationen ein, wie Titel und Betrag des Budgets. Während dieses Zustands kann der Benutzer die Eingaben noch anpassen.
6. Speichern des Budgets: Nachdem alle Daten eingegeben wurden, drückt der Benutzer auf „Speichern“. Das Budget wird an den Budget-Service übergeben und in der Firestore-Datenbank gespeichert.
7. Bestätigung: Nach erfolgreicher Speicherung wird dem Benutzer eine Bestätigung angezeigt, und die App kehrt zur Budget-Seite zurück.
8. Fehlerbehandlung: Sollte es bei der Speicherung zu einem Fehler kommen, wird der Benutzer informiert, und die App kehrt zum Zustand „Budgeteingabe“ zurück, damit der Benutzer die Eingaben korrigieren oder es erneut versuchen kann.
9. Ruhezustand: Nach Abschluss der Aktion (Erstellung oder Bearbeitung eines Budgets) kehrt die App wieder in den Wartezustand zurück und wartet auf weitere Benutzerinteraktionen.



7. Interaktionsdiagramme

- Sequenzdiagramm Budget-Erstellung

1. Benutzer öffnet die Budget-Seite: Der Benutzer startet die Interaktion, indem er auf die Budget-Seite navigiert. Die Anwendung zeigt dem Benutzer eine Übersicht der bereits bestehenden Budgets an.
2. Ein Klick auf „Neues Budget“ öffnet den Budget-Dialog: Auf der Budget-Seite klickt der Benutzer auf die Schaltfläche „Neues Budget“, wodurch ein Dialog zur Eingabe der Budgetinformationen geöffnet wird.
3. Benutzer gibt Budgetdaten ein: Der Benutzer gibt die notwendigen Daten für das Budget, wie den Titel und den Betrag, in das Dialogformular ein.
4. Budget-Daten werden gespeichert: Nachdem der Benutzer auf „Speichern“ klickt, werden die eingegebenen Daten an den budget.service.ts gesendet.
5. Speicherung in Firestore: Der budget.service.ts übermitteln die Budgetdaten an die Firestore-Datenbank, wo sie gespeichert werden.
6. Bestätigung an den Benutzer: Sobald die Daten erfolgreich gespeichert wurden, erhält der Benutzer eine Bestätigung, dass das neue Budget erfolgreich erstellt wurde, und wird wieder zur Budget-Seite zurückgeführt.



8. Aktivitätsdiagramm

Ablauf des Aktivitätsdiagramms für die Budgetverwaltung:

1. Start des Prozesses:
Der Benutzer öffnet die EazyBooks-App und meldet sich mit seinen Anmeldedaten an.
2. Navigieren zur Budget-Seite:
Nach erfolgreicher Anmeldung navigiert der Benutzer zur Budget-Seite, auf der er bestehende Budgets verwalten oder ein neues Budget erstellen kann.
3. Entscheidungspunkt – Auswahl der Budget-Aktion:
Der Benutzer entscheidet, ob er ein neues Budget erstellen, ein bestehendes Budget bearbeiten oder ein Budget löschen möchte.
4. Neues Budget erstellen:
 - a. Der Benutzer wählt die Option „Neues Budget erstellen“.
 - b. Die App navigiert ihn zu einem Eingabeformular, in dem er die Budgetdetails (z. B. Titel, Betrag) eingeben kann.
 - c. Nach der Eingabe kann der Benutzer das Budget speichern.
5. Bestehendes Budget bearbeiten:
 - a. Falls der Benutzer ein Budget bearbeiten möchte, wird das bestehende Budget im Eingabeformular geladen.
 - b. Der Benutzer kann die Werte anpassen und anschließend die Änderungen speichern.
6. Budget löschen:
 - a. Falls der Benutzer ein Budget löschen möchte, wird eine Bestätigungsabfrage angezeigt.
 - b. Nach der Bestätigung wird das Budget entfernt.
7. Erneuter Entscheidungspunkt nach Eingaben:
 - a. Falls die Eingaben korrekt sind, wird das Budget gespeichert oder gelöscht.
 - b. Falls Fehler auftreten oder der Benutzer die Eingabe anpassen möchte, kann er Änderungen vornehmen und erneut speichern.
8. Rückkehr zur Budget-Seite:
 - a. Nach erfolgreicher Budgetverwaltung kehrt der Benutzer zur Budget-Seite zurück.
 - b. Dort kann er weitere Aktionen durchführen oder die App verlassen.
9. Ende des Prozesses:
 - a. Der Ablauf endet, sobald der Benutzer keine weiteren Aktionen mehr durchführt.

