

# Inhalt

- Anforderungen
- Use Cases
- Proof of Concepts
- Tech Stack
- Externe API's
- Rapid Prototyp
- Ausblick Audit 4

# Anforderungen

#### · Manuelle Eingabe von Kontodaten

Das System muss eine benutzerfreundliche Oberfläche bereitstellen, um manuell Einkommens- und Ausgabedaten eingeben zu können

Das System sollte dem Benutzer ermöglichen, relevante Informationen präzise und einfach einzutragen.

Das System muss eine Plausibilitätsprüfung der manuell eingegebenen Daten durchführen, um sicherzustellen, dass sie den erwarteten Format- und Wertebereichen entsprechen.

## Übersichtliche Darstellung

Das System sollte verschiedene Diagrammtypen verwenden, um die freie Liquidität, Einnahmen und Ausgaben klar und leicht verständlich darzustellen.

Das System sollte dem Benutzer eine klare und visuelle Darstellung seiner finanziellen Situation anzeigen

Das System muss die Prognosen visuell ansprechend und leicht verständlich darstellen, um den Benutzer für die Altersvorsorge zu motivieren.

## Ausgaben kategorisieren

Das System muss vordefinierte Ausgabekategorien bereitstellen.

Das System sollte dem Benutzer ermöglichen, Ausgaben bestimmten Kategorien zuzuordnen, um einen detaillierten Überblick zu erhalten.

Das System muss dem Benutzer die Möglichkeit bieten, auch manuell Ausgabekategorien definieren zu können.

#### · Prognosen erstellen

Das System ermöglicht es dem Benutzer, manuell seine Sparrate einzutragen.

Das System muss dem Benutzer die Möglichkeit bieten, verschiedene Anlageoptionen auszuwählen.

Das System muss die Prognosen in einer zeitlichen Entwicklung bis zur Rente darstellen

Das System sollte dem Benutzer klare Erläuterungen zu den Annahmen und Berechnungen der Prognosen bieten

Für das dritte Audit haben wir unsere Anforderungen in die für den Prototypen relevanten Funktionen unterteilt.

# Manuelle Eingabe von Kontodaten:

Das System muss eine benutzerfreundliche Oberfläche bereitstellen, um manuell Einkommens- und Ausgabendaten einzugeben.

Das System sollte dem Benutzer ermöglichen, relevante Informationen präzise und einfach einzutragen.

Das System muss eine Plausibilitätsprüfung der manuell eingegebenen Daten durchführen, um sicherzustellen, dass sie den erwarteten Format- und Wertebereichen entsprechen.

# Übersichtliche Darstellung:

Das System sollte verschiedene Diagrammtypen verwenden, um die freie Liquidität, Einnahmen und Ausgaben klar und leicht verständlich darzustellen.

Das System sollte dem

Benutzer eine klare und visuelle Darstellung seiner finanziellen Situation anzeigen.

## Ausgaben kategorisieren:

Das System muss vordefinierte Kategorien für Ausgaben bereitstellen.

Das System sollte dem Benutzer ermöglichen, Ausgaben diesen Kategorien zuzuordnen, um einen detaillierten Überblick zu erhalten.

Das System muss dem Benutzer die Möglichkeit bieten auch manuell Ausgabekategorien definieren zu können.

# Prognosen erstellen:

Das System ermöglicht es dem Benutzer, manuell seine Sparrate einzutragen.

Das System muss dem Benutzer die Möglichkeit bieten, verschiedene Anlageoptionen auszuwählen.

Das System muss die Prognosen in einer zeitlichen Entwicklung bis zur Rente darstellen.

Das System sollte dem Benutzer klare Erläuterungen zu den Annahmen und Berechnungen der Prognosen bieten.

Das System muss die Prognosen visuell ansprechend und leicht verständlich darstellen, um den Benutzer für die Altersvorsorge zu motivieren.

## Registrierung

use case Registrierung actors Benutzer precondition

Die Webanwendung ist zudem erreichbar und funktionsfähig.

#### main flow

- 1. Der Benutzer ruft die Webseite auf, um sich zunächst zu registrieren.
- 2. Der Benutzer gibt seine Informationen ein (Vorname, Nachname, Alter, E-Mail-Adresse und ein Passwort seiner Wahl)
- 3. Das System prüft, ob die eingegebenen Daten korrekt sind, z.B. ob die E-Mail-Adresse gültig und ob das Passwort sicher genug ist.
- 4. Wenn die Daten akzeptiert werden, erstellt das System einen neuen, personalisierten Account für den Benutzer.
- 5. Der Benutzer bekommt eine Bestätigungsmail, um die Registrierung zu verifizieren.
- 6. Nach erfolgreicher Bestätigung wird der Benutzer zur Hauptseite, bereits angemeldet, weitergeleitet.
- 7. Der Benutzer definiert langfristige und mittelfristige Ziele im Bereich der Altersvorsorge.

postcondition Ein neuer Benutzer-Account wurde erstellt.

end Registrierung

Zusätzlich zu den Use-Case-Diagrammen aus dem vorhergehenden Audit 2, haben wir nun auch textuelle Use Cases für alle relevanten Vorgänge definiert, welche zudem auch die Anwendungslogik unseres Systems beschreiben.

## Anmeldung

## use case Anmeldung

actors Benutzer

precondition Benutzer besitzt einen Account, ist jedoch noch nicht angemeldet.

## main flow

- 1. Der Benutzer gibt seine E-Mail und sein Passwort ein.
- 2. Das System prüft die eingegebenen Daten auf Korrektheit.
- 3. Der Benutzer ist angemeldet und wird auf die Hauptseite

## postcondition

Erfolgreiche Anmeldung des Benutzers

# exceptional flow Fehlerhafte Daten

2a. Das System stellt eine Inkonsistenz der Daten fest.4. Der Benutzer wird auf den Fehler hingewiesen.

## postcondition Fehlerhafte Anmeldedaten

Der Benutzer wurde nicht angemeldet und auch darüber informiert.

## end Anmeldung

## Manuelle Eingabe von Kontodaten

# use case Manuelle Eingabe actors Benutzer

**precondition** Der Benutzer hat sich erfolgreich angemeldet und möchte nun seine Kontodaten einpflegen.

#### main flow

- 1. Der Benutzer gibt Einkommens- (Jobbezeichnung + Betrag) und Ausgabedaten (Ausgabebeträge in passenden Ausgabekategorien) manuell in die benutzerfreundliche Oberfläche ein.
- 2. Das System führt eine Plausibilitätsprüfung der manuell eingegebenen Daten durch.
- 3. Der Benutzer besitzt die Berechtigung, seine Daten jederzeit anzunassen
- 4. Der Benutzer bestätigt die manuelle Eingabe seiner Kontodaten.

**postcondition** Die manuell eingegebenen Kontodaten wurden erfolgreich erfasst und sind im System einsehbar.

exceptional flow Fehlerhafte oder unvollständige Daten

- 3a. Das System erkennt fehlerhafte oder unvollständige Daten.
- 3b. Der Benutzer wird auf die Fehler hingewiesen.
- 3c. Der Benutzer hat die Möglichkeit, fehlerhafte Daten zu korrigieren.

#### postcondition Fehlerhafte Daten

Die manuelle Eingabe von Kontodaten konnte nicht abgeschlossen werden, der Benutzer wurde darüber informiert.

## end Manuelle Eingabe

## Prognosen erstellen

use case Prognosen erstellen

actors Benutzer
precondition Der Benutzer hat sich erfolgreich angemeldet und seine
Kontodaten manuell eingegeben.

## main flow

- 1. Der Benutzer navigiert sich zur Startseite.
- 2. Der Benutzer trägt manuell seine Sparrate ein.
- 3. Der Benutzer wählt eine der möglichen Anlageoptionen aus.
- 4. Das System stellt die Kapitalentwicklung auf den Zeitraum

bis zum Renteneintritt visuell dar.

postcondition Prognose wurde erstellt

 $\label{lem:prognose} \mbox{Dem Benutzer wurde erfolgreich eine Prognose erstellt und angezeigt.}$ 

end Prognosen erstellen

## Übersichtliche Darstellung

# use case Übersichtliche Darstellung actors Benutzer

**precondition** Der Benutzer hat sich erfolgreich angemeldet, seine Kontodaten manuell eingegeben und möchte sich seine Kontodaten anzeigen lassen.

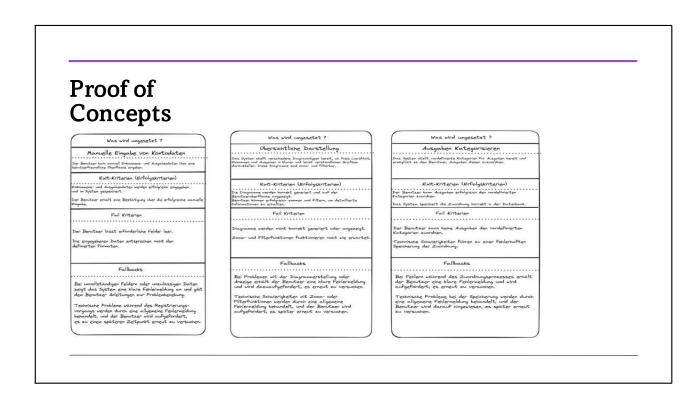
## main flow

- 1. Der Nutzer navigiert sich zur Startseite.
- 2. Das System verwendet verschiedene Diagrammtypen, um die freie Liquidität, Einnahmen und Ausgaben klar und leicht verständlich darzustellen.
- 3. Der Benutzer kann sich eine klare und visuelle Darstellung seiner finanziellen Situation anzeigen lassen.

## post condition

Der Benutzer hat einen übersichtlichen Überblick über seine freie Liquidität, Einnahmen und Ausgaben.

end Übersichtliche Darstellung



In den ersten beiden Audits wurden unsere POC`s nur grob beschrieben. Diese wurden nun detailliert ausgearbeitet und formuliert. Dabei wurden auch Exit- und Fall-Kriterien berücksichtigt, sowie mögliche Fallbacks.

# **Proof of** Concepts

## Was wird ungesetzt?

System ist Philip Progresses au ersteller. Die Progresse zeiger, ich des Kaptal über die Tabre erthickelt, besierend unf der gelegten moustlichen Sparreite und seins. Die Progressen besiehen sich auf die gewählte det der sureskrige.

Exit-Kriterien (Erfolgskriterien)

System erstellt Prognosom erfolgreich
erstellten Prognosom exjem Akhatelertuicklung über die
alte Zetspanne und berücksichtigen die gewählte Altersvors

# Fail Kriterien

# Fallbacks

## Was wird umgesetzt?

Visuelle Darstellung

Das Systex kan Proposes visuell darsteller, un der Bendzer für die
Altersionsenge zu metheren.

Exit-Kriterien (Erfolgskriterien)

e Vauelle Durstellung der Prognesen ist anspressiond und leises
estandeise,
midser bisonen de Entwikklung bres Kapitals über die Jahre klar eren.

# Fail Kriterien

ie Visuelle Durstellung der Prognosen ist unklar oder eblerhaft.

Fallbacks

Bei Problemen wit der visuellen Darstellung erhalten
Berutzerklare Anweisungen zur Fehlerberbung.

Falls die Darstellung nicht den erwarteten Anforderungen entspricht, wird dem Benutzer erklärt, wie er die Einstellungen anpassen kann.

# Tech Stack und externe API's

## **Frontend**

#### HTML

- weitverbreitete textbasierte Auszeichnungssprache
- Darstellung von Webseiten in Browsern

#### CSS

- Computersprache
- Aufbauen und Strukturieren von Webseiten

#### JavaScript

- Programmiersprache
- ermöglicht Interaktion zwischen Anwender und System

Die folgenden Folien beschreiben all unsere verwendeten Technologien zur Realisierung unseres Prototyps. Wir fokussieren uns auf Frontend, sowohl auf die Verwendung der Frameworks Vue.js und Bootstrap als auch auf die Bibliothek Chart.js.

# Tech Stack und externe API's

# Frameworks

#### Vue.js

- JavaScript-Webframework
- Erstellen von Single-Page-Webanwendungen

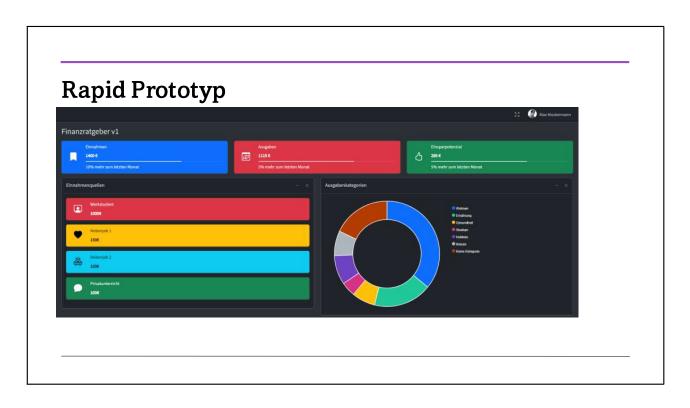
#### Rootstran

- Frontend-CSS-Framework
- Gestaltungsvorlagen basierend auf HTML und CSS
- meistverbreitete Oberflächengestaltungselemente wie z.B. Formulare, Buttons, Tabellen, usw.

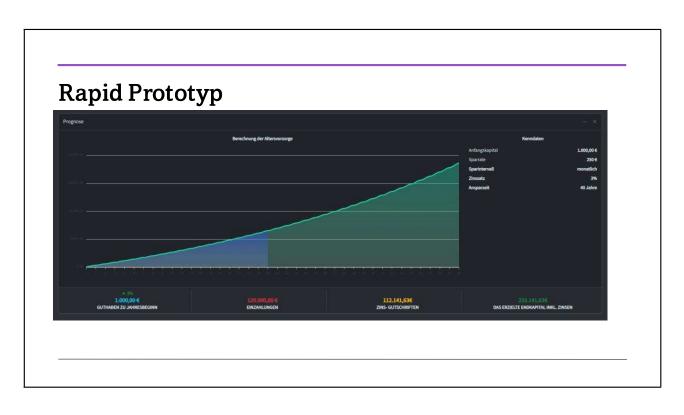
## Bibliotheken

## Chart.js

- Open-Source-JavaScript-Bibliothek
- Datenvisualisierung
- unterstützt 8 Diagrammtypen: Balken, Linien, Flächen, usw.



Auf den nächsten beiden Folien sieht man nun die erste Umsetzung unseres Rapid Prototyps unter der Verwendung der bereits genannten Tech Stacks. Es werden bereits die Einnahmen und Ausgaben, das daraus errechnete Einsparpotenzial und mögliche Einnahmequellen visuell dargestellt. Zudem sieht man auch eine erste Grafik in Form eines Kreisdiagramms zu den erstellten Ausgabekategorien. Der Prototyp ist lauffähig und wird in Github und zusätzlich auch hier abgelegt: https://github.com/alexzlmn/finanzratgeber



Auf dieser Folie sieht man eine Grafik in Form eines Liniendiagramms, welche die Berechnung der Altervorsorge durch die Verzinsung von monatlichen Sparbeiträgen visualisiert.

# **Ausblick Audit 4**

- Weitere Ausarbeitung unseres Prototyps: Erweiterung um Registrierung, Anmeldung, manuelle Dateneingabe, Berechnung der Prognosen unter Berücksichtigung verschiedenster Anlagestrategien mithilfe unseres Tech Stacks
- Codeinspektion unseres finalen Prototyps
- Fazit und kritische Einordnung unseres Projekts anhand unserer ursprünglichen Zielsetzung
- Abschließendes Poster für die finale Präsentation