

Software Requirements Specification

for Wahlinator 3000

Version 2.0

Inhalt

1.	Introduction.....	1
1.1.	Purpose.....	1
1.2.	Document Conventions	1
1.3.	Intended Audience and Reading Suggestions	1
1.4.	Product Scope	1
1.5.	References	1
2.	Overall Description	2
2.1.	Product Perspective	2
2.2.	Product Functions	2
2.3.	User Classes and Characteristics.....	2
2.4.	Operating Environment	3
2.5.	Design and Implementation Constraints	3
2.6.	User Documentation.....	4
2.7.	Assumptions and Dependencies	4
3.	External Interface Requirements.....	5
3.1.	User Interfaces.....	5
3.2.	Hardwareinterfaces	5
3.3.	Softwareinterfaces.....	5
3.4.	Communications Interfaces	5
4.	System Features	6
4.1.	Login (Identification).....	6
4.1.1.	Description and Priority	6
4.1.2.	Stimulus/Response Sequences.....	6
4.1.3.	Functional Requirements	6
4.2.	Wählen (Speichern der Wahl).....	7
4.2.1.	Description and Priority	7
4.2.2.	Stimulus/Response Sequences.....	7
4.3.	Verlinken auf Wahl-O-Mat	7
4.3.1.	Description and Priority	7
4.3.2.	Stimulus/Response Sequences.....	7
4.4.	FAQ.....	8
4.4.1.	Description and Priority	8
4.4.2.	Stimulus/Response Sequences.....	8

5.	Other Nonfunctional Requirements.....	9
5.1.	Performance Requirements.....	9
5.2.	Safety Requirements.....	9
5.3.	Security Requirements.....	9
5.4.	Software Quality Attributes	9
5.5.	Business Rules.....	9
6.	Other Requirements.....	10
6.1.	SMART.....	10
6.2.	Persona Definition	11
6.3.	Der Goldene Pfad.....	12
6.4.	Wie können wir...?	12
6.5.	Design-Concept.....	13
6.6.	Architecture-Design	16
7.	UML-Diagrams.....	17
7.1.	State Diagram	17
7.2.	Activity diagram	18
7.3.	Class diagram	19
7.4.	Context diagram	19
7.5.	Use-Case diagram	20
7.6.	Sequence diagram	20

Revision History

Name	Date	Reason For Changes	Version
Leon Rensch	02.11.2022	First Version	1.0.0
Leon Rensch	08.05.2023	Update before pitch	2.0.0

1. Introduction

1.1. Purpose

Das Ziel ist es, eine Webapplikation zu entwickeln, um das Wählen zu erleichtern und die Wahlbeteiligung zu steigern.

1.2. Document Conventions

Überschriften haben verschiedene Prioritäten

1.3. Intended Audience and Reading Suggestions

Dieses Dokument ist für die Benotung gedacht. Hauptsächlich wird es von Herrn Grün gelesen. Eigentlich ist das Dokument aber als Dokumentation für alle Stakeholder und als gemeinsames Protokoll gedacht für alle Entwickler, um bestimmte Dinge nachlesen zu können. Im Folgenden wird das entstehende Produkt näher erläutert. Welche Perspektiven werden eingenommen, welche Funktionen hat das Produkt und tiefer liegende Informationen, wie User Classes, Design Infos und Dokumentation. Die restliche Struktur und die empfohlene Reihenfolge kann man dem Inhaltsverzeichnis entnehmen.

1.4. Product Scope

Der Wahlinator 3000 soll Wähler*innen von zuhause dazu animieren auch zu wählen. Zusätzlich zu den Wähler*innen, die in Wahllokale gehen und Briefwahl machen, schafft der Wahlinator 3000 eine weitere Möglichkeit zu wählen. Zusätzlich verweist die Website auf den Wahl-O-Mat und schafft somit die Möglichkeit, sich vor der Wahl direkt nochmal zu erkundigen. Es soll möglich sein, gleichgültig wie bei den anderen Wahlverfahren an einer Bundes- oder Landtagswahl teilzunehmen. Dabei sollen alle Wahlgrundsätze und Sicherheitsrichtlinien sowie Datenschutz eingehalten werden.

1.5. References

Wahl-O-Mat, Bundeszentrale für politische Bildung, 19.10.2022:

<https://www.bpb.de/themen/wahl-o-mat/>

2. Overall Description

2.1. Product Perspective

Das Produkt wird entwickelt, um das Wählen zu vereinfachen; dadurch können mehr Menschen zum Wählen angeregt werden. Das bereits bestehende Wahlsystem wird dabei nicht verändert.

2.2. Product Functions

Das Ziel des Projekts ist es, eine Wahl online zu ermöglichen. Dadurch zeichnet sich die Wahlfunktion als Hauptfunktion unseres Projekts aus. Dies kann sich in die 5 Wahlgrundsätze untergliedern:

Allgemein:

Bevor der User wählen kann, muss authentifiziert werden, ob der Wähler wahlberechtigt ist. Als Bedingung gilt: Der Wähler darf noch nicht gewählt haben, er muss min. 18 Jahre alt sein und muss Wahlrecht haben (Deutscher).

Unmittelbar:

In der Wahl sollen die zur Wahl stehenden Personen direkt gewählt werden

Frei:

Wähler darf ohne Druck wählen können

Geheim:

Es muss umgesetzt sein, dass man nicht erkennen kann, wer welchen Kandidaten gewählt hat (möglicher Einsatz einer Zero-Knowledge-Proof-Function)

Gleich:

Es darf keine Gewichtung der Stimmen geben. Jede Stimme muss gleich viel wert sein.

2.3. User Classes and Characteristics

Wähler*innen Technik-Affin:

- Nutzung bei jeder Wahl
- Wahlfunktion & Wahl-O-Mat werden genutzt
- technische begabt
- normaler Sicherheitsstandard wie bei allen Wähler*innen
- Berechtigung zur einmaligen Stimmabgabe
- Von keinem Abschluss bis zu keiner Bildungs-Grenze nach oben
- Erfahrung abhängig vom Alter und bisheriger Wahlbeteiligung
- Benötigt werden technische, internetfähige Endgeräte mit Internetzugang und eine Wahlberechtigung
- Wichtigste und einzige Nutzer der Anwendung

Wähler*innen nicht Technik-Affin:

- Nutzung für jede staatliche Wahl soll ermöglicht werden
- Wahlfunktion & Weiterleitung zum Wahl-O-Mat
- keine oder geringe technische Erfahrung
- gleicher Sicherheitsstandard und gleiche Berechtigung für jeden Wähler
- geringe bis hohe Bildung
- Erfahrung abhängig vom Alter und bisheriger Wahlbeteiligung
- Benötigt werden technische, internetfähige Endgeräte mit Internetzugang und eine Wahlberechtigung
- Der Wählergruppe soll die Alternative angeboten werden, auch online zu wählen. wichtig dabei ist ein einfaches und übersichtliches Layout, sodass Wähler intuitiv mit der Webseite umgehen können

2.4. Operating Environment

Das Endergebnis des Projekts soll eine funktionierende Webseite sein. Authentifikation über analog beantragte ID und Passwort, Implementation einer Datenbank zur Speicherung der Wahlergebnisse und eine zusätzliche Datenbank über Personen, die gewählt haben.

2.5. Design and Implementation Constraints

Wahlgrundsätze sollen durch unser Projekt erfüllt werden. Dabei steht besonders der Datenschutz im Vordergrund.

Zusätzlich muss ein starker Fokus auf die Gewährleistung der Integrität der Wahl gelegt werden. Hierzu muss die Verfügbarkeit des Systems immer sichergestellt werden, auch bei vielen Nutzern gleichzeitig, denn eine Wahl kann nur schwer kurzfristig verschoben werden. Dies ist mit Redundanz und einer sehr hohen maximalen Auslegung der maximalen gleichzeitigen Anfragen umzusetzen. Eine Downtime während der Wahlphase würde Wähler benachteiligen und so die Wahlgrundsätze verletzen und zudem das Vertrauen in den Wahlprozess zerstören. Mit der gleichen Begründung muss ebenfalls jegliche Manipulation oder sonstige Einflussnahme von außen ausgeschlossen werden. Zur Verifizierung des Wählers kann hierbei ein Zero Knowledge Proof eingesetzt werden. Um Datenmanipulation im Nachhinein zu verhindern, kann beispielsweise eine Blockchain aufgebaut werden. Zudem muss verhindert werden, dass später aufgrund von Zeitvergleichen nachvollziehbar ist, wer wen gewählt hat.

2.6. User Documentation

- SMART
- Personas/Stakeholder
- Wie können wir...?
- Goldener Pfad
- Design-Konzept
- UML-Diagramme

2.7. Assumptions and Dependencies

Die Regierung muss mitspielen (Zustimmung im Bundestag etc.)

Die Bevölkerung muss davon erfahren

Die Entwicklung der Anwendung ist abhängig von unseren Entwicklungsfähigkeiten. Je besser wir die Website coden können, desto besser wird die Seite.

Einschränkungen würden wir wahrscheinlich vom Bund bekommen und müssten im Allgemeinen sehr eng mit den Gesetzgeber*innen zusammenarbeiten

3. External Interface Requirements

3.1. User Interfaces

- **Adminseite:**
Hier kann ein/e Regierungsmitarbeiter/in neue Nutzer anlegen und ändern.
- **Startseite/Anmeldeseite:**
Hier muss der User seine ID und sein Passwort eingeben und wird über den Button "Anmelden" angemeldet.
Außerdem gibt es einen Button zum FAQ und einen Button zum Wahl-O-Mat.
- **FAQ:**
Hier sind alle Fragen aufgelistet und mit einem Klick auf diese Frage wird darunter die Antwort sichtbar.
- **Wahl-O-Mat:**
Hier kann sich der User zu allen Parteien etc. informieren.
- **Wahlauswahlseite:**
Seite auf der alle aktuell verfügbaren Wahlarten angezeigt werden.
(Maximal Bundestags- & Landtagswahl)
- **Wahlseite:**
Hier kann der User seine verfügbaren Stimmen über Auswahlmenüs auswählen und über einen Senden-Button senden. Bevor gesendet wird, wird der User gefragt, ob er wirklich absenden will. Ein weiterer Button führt weiterhin zum FAQ.

(Siehe Design-Konzept)

3.2. Hardwareinterfaces

- Datenbank extra.
- Abfragen von ID und Passwort zum Vergleich beim Anmelden.
- Abfragen, ob mit der ID schon gewählt wurde.

3.3. Softwareinterfaces

- Frontend: Angular
- Backend: Java, Springboot
- Datenbank: MySQL

Daten werden abgefragt und gespeichert über eine SQL-Datenbank. Beim Anmelden wird das Passwort verschlüsselt und mit dem verschlüsselten Passwort in der Datenbank verglichen.

Nach dem Absenden wird die Auswahl in der Datenbank gespeichert.

3.4. Communications Interfaces

(Siehe 3.3)

4. System Features

4.1. Login (Identification)

4.1.1. Description and Priority

Der Login hat eine hohe Priorität und soll den User ermöglichen, sich eindeutig zu identifizieren. Dieser Schritt ist notwendig, um die Berechtigung zum Wählen zu bekommen. Hierbei ist die Sicherheit des Logins und die User bezogenen Daten ein Risikofaktor, der entsprechend beachtet werden muss.

4.1.2. Stimulus/Response Sequences

- 2 Eingabefelder für "Username" und "Passwort"
 - Hier gibt der User seine anmelde Daten ein
- Anmelde Button
 - Wenn der User den Button drückt, wird überprüft, ob die beiden Eingabefelder auch befüllt wurden, und gibt eine dementsprechende Fehlermeldung, falls eines der Felder nicht ausgefüllt wurde
 - Sind alle Felder ausgefüllt, werden die Eingabedaten mit der Datenbank überprüft und der User wird weitergeleitet, oder es wird ihm eine Fehlermeldung angezeigt

4.1.3. Functional Requirements

REQ-1: Eingabe Fehler

Wenn nicht beide Eingabefelder gefüllt sind, ist der Anmeldebutton nicht nutzbar und optisch ausgegraut.

REQ-2: Datenabgleich Fehler

Wenn die eingegebenen Daten nicht mit der Datenbank übereinstimmen, gibt es eine dementsprechende Fehlermeldung für den User

REQ-3: Schon gewählt

Wenn über die im Login eingegebenen Daten abgestimmt wurde, ist der User zwar angemeldet, hat aber im Wahlauswahlfenster keine Optionen mehr.

4.2. Wählen (Speichern der Wahl)

4.2.1. Description and Priority

Diese Funktion soll es ermöglichen, dass der User seine Wahl abgeben kann und diese gespeichert wird. Dabei ist es wichtig, dass das Wahlergebnis des Users nicht zu ihm zurückverfolgt werden kann. Diese Funktion hat die höchste Priorität im Projekt.

4.2.2. Stimulus/Response Sequences

- Auswahl der beiden Wählergruppen (Direktmandat und Partei)
- Abgabe der Wahl über Button "Absenden"
 - Der User wird gefragt, ob er sich sicher ist → der User hat die Möglichkeit zurück zur Wahl zu gehen oder seine Wahl zu bestätigen
 - Wird die Wahl bestätigt, wird ein neuer Datensatz in die Datenbank aufgenommen, um die Wahlergebnisse zu speichern
 - Ist der Wähler mit seiner Wahl nicht zufrieden wird, er zurück auf die Wahlseite geleitet und er kann erneut wählen
 - Wenn der Wähler seine Stimme abgegeben hat, wird der Zugang zur Wahl für ihn gesperrt, sodass er nicht nochmal wählen kann.
- REQ-4: unvollständige Wahl

Wenn der Wähler nicht alle erforderlichen Stimmen abgegeben hat, kann er nicht abgeben

4.3. Verlinken auf Wahl-O-Mat

4.3.1. Description and Priority

Die Funktion soll den Nutzer ermöglichen, sich über die Wahl und die einzelnen Wahlparteien zu informieren. Dafür soll ein Link zur Seite "wahl-o-mat.de" erstellt werden, der den User auf diese weiterleitet. Dieses Feature hat eine geringere Priorität

4.3.2. Stimulus/Response Sequences

- Es soll als Button einen Link geben, der den User auf Wahl-O-Mat weiterleitet.
- REQ-5: Weiterleitung
- Nach dem Drücken des Buttons, soll der User auf die Seite vom Wahl-O-Mat weitergeleitet werden.

4.4. FAQ

4.4.1. Description and Priority

Das FAQ hat eine mittlere Priorität und soll den User über alle aufkommenden Fragen eine Antwort geben können. Die Seite besteht aus einer Auflistung von Fragen.

4.4.2. Stimulus/Response Sequences

- Klickt der User auf eine Frage, so wird die Antwort darunter sichtbar und verschwindet beim erneuten Klicken.
- REQ-6: FAQ-Erreichbarkeit
- Das FAQ soll von jeder Seite durch einen Button erreichbar sein.
- REQ-7: Antwort anzeigen
- Wenn auf die Frage geklickt wird, wird die Antwort darunter sichtbar und verschwindet beim erneuten Klicken.

5. Other Nonfunctional Requirements

5.1. Performance Requirements

Die Webseite muss unabhängig vom Webbrowser und Betriebssystem aufgerufen werden können. Auch von einem Handy muss das Anmelden und Wählen funktionieren.

5.2. Safety Requirements

Wenn der User wählt und nicht die richtige Anzahl an Stimmen gesetzt wurden, sollen die Daten nicht gespeichert werden und der User informiert werden.

5.3. Security Requirements

Datenbank soll vor Angriffen geschützt sein. Daten auf der Datenbank sollen verschlüsselt sein. Daten vom User und das, was er gewählt hat, dürfen in keinem Zusammenhang stehen.

5.4. Software Quality Attributes

Das Design der Webseite soll auf allen Bildschirmgrößen und auch auf dem Handy ansprechend sein.

Das Wählen soll über jeden Browser und Betriebssystem ohne Probleme möglich sein.

5.5. Business Rules

Wähler können nur einmal wählen. User bezogene Daten sollen abgekapselt von der Wahl sein, damit die Wahl auch wirklich anonym ist. Alle Passwörter sollen verschlüsselt auf der Datenbank liegen und auch nie entschlüsselt werden.

6. Other Requirements

6.1. SMART

Langfristiges Ziel

Jeder soll von zuhause wählen können.

Spezifisch

Jeder soll auch von zuhause wählen können, um das Wählen zu erleichtern und die Wahlbeteiligung zu steigern.

Messbar

Eine Wahlbeteiligung von 75 % oder höher bei den nächsten Wahlen Briefwahlanteil soll dadurch verringert werden

Akzeptiert/attraktiv

- gutes Design
- gute Usability
- sicher (Einhaltung der 5 Wahlgrundsätze)
 - Sallgemein, unmittelbar, frei, gleich und geheim

Realistisch

- Umsetzung in Form einer Website
- Eindeutige Identifizierung der Person

Terminiert

- Bis zum Abgabedatum
- Ende 4. Semester. (26.05.2023)

Zusammengefasst

Bis zum 26.05.2023 soll jeder leicht und sicher von zuhause wählen können, um die Wahlbeteiligung zu steigern.

6.2. Persona Definition

Persona Staat:

- männlich, weiß, 50 Jahre
- NRW Dortmund
- unglücklich verheiratet
- Beruf: Anwalt für Familienrecht
- Ziele:
 - Frührente
 - gute Reputation gegenüber seiner Partei und der Wählerschaft
 - Höhere Wahlbeteiligung
- Frustration:
 - integer
 - nicht auswertbare Wahlzettel
 - Lobbyismus ist geil

Persona Wähler*innen mittel:

- weiblich, 30 Jahre, weiß
- München
- ledig
- Beruf: Bachelor of Arts BWL Schwerpunkt HR, aktuell Gap-Year
- Ziele:
 - „YOLO“ („You only live once“)
 - Frühe Altersvorsorge
- Frustration:
 - Probleme mit Technik
 - Wählen dauert zu lange

Persona Wähler*innen jung:

- diverse, 20 Jahre, Dreadlocks
- Berlin
- #singlelikeapingle
- Beruf: arbeitslos, Aktivistin, sitzt immer wieder für festkleben auf Straßen ein
- Ziele:
 - CO2 Ausstoß minimieren und Klimawandel stoppen
 - immer auf dem neuesten Stand rein
- Frustration:
 - Politik
 - Industrie
 - Autofahrer*innen

Persona Wähler*innen alt:

- männlich, 50 Jahre, weiß
- Nusplingen
- verheiratet
- Beruf: Ingenieur bei Zeiss
- Ziele:
 - Altersvorsorge
 - E-Bikes
 - Wohnmobile
- Frustration:

- Inflation
- Erhöhung des Renteneintritts
- Baustellen

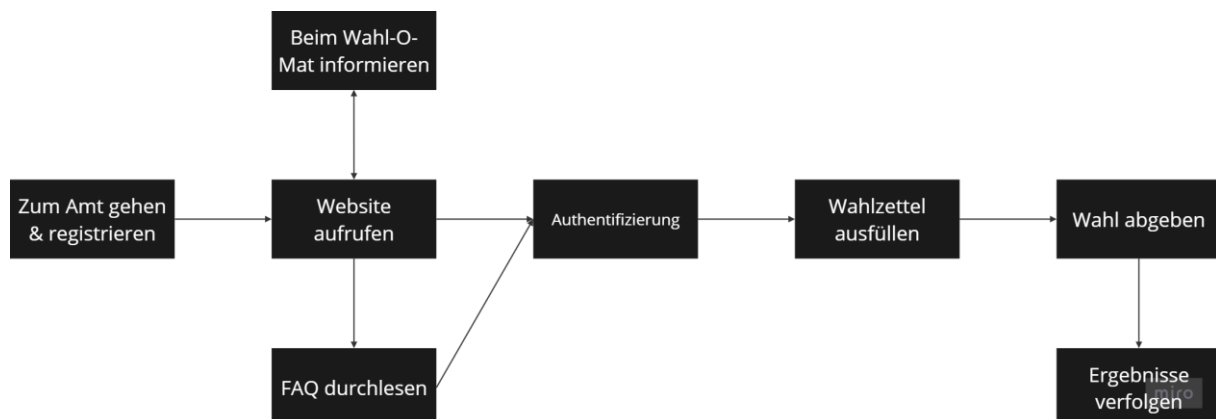
Persona Entwickler:

- männlich, 35 Jahre, Bildschirmbraun
- Stuttgart
- ledig
- Beruf: Entwickler
- Ziele
 - Einfache Umsetzung
 - Leicht wartbar
- Frustration:
 - fehlendes Semikolon im Code
 - Single
 - nervige Mutter (Weil noch zuhause wohnt)

Stakeholder

- Bürger
- Staat
- Entwickler

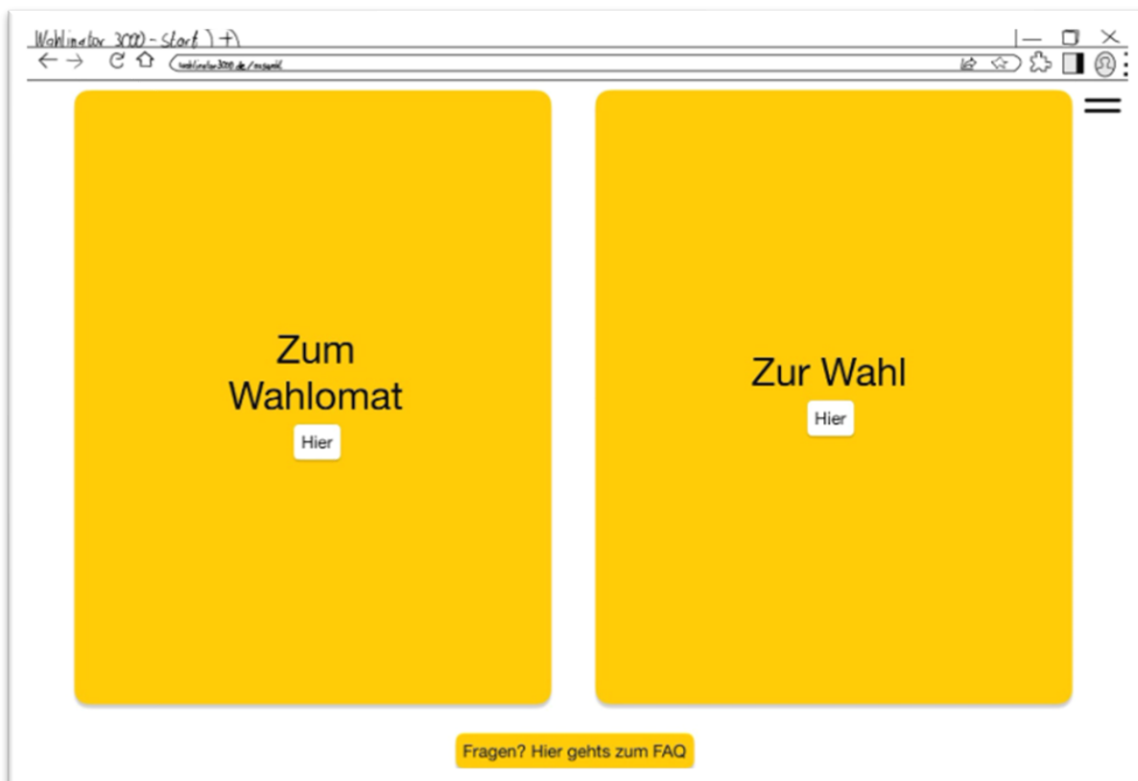
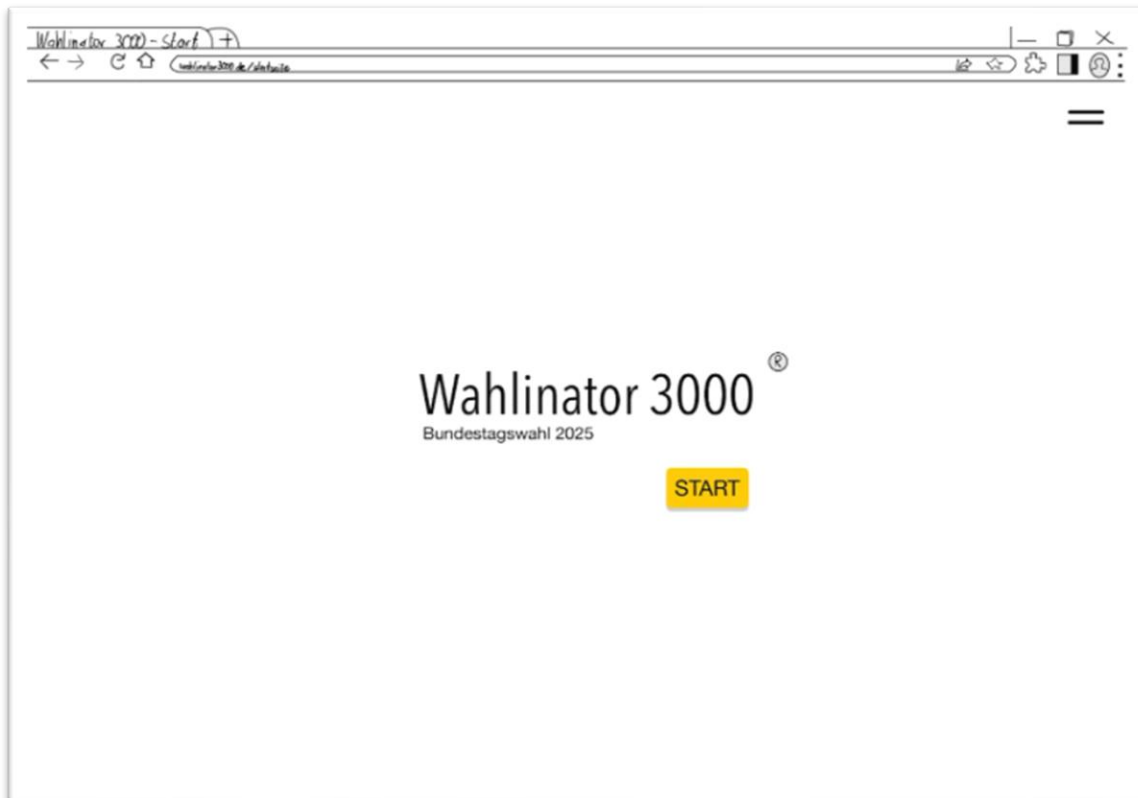
6.3. Der Goldene Pfad



6.4. Wie können wir...?

- ... das Wählen vereinfachen? (z.B. Die App möglichst ansprechend für alle Anwender gestalten)
- ... die Person feststellen/authentifizieren?
- ... Datenschutz/Sicherheit gewährleisten?
- ... Leute anwerben/ auffordern?
- ... Marketing machen?

6.5. Design-Concept



Wahlleiter 3000 - Start (+)

← → ↺ ⚙️ 🔍 🧑

Häufig gestellte Fragen

Frage 1: Hier könnte ihre Frage 1 stehen? Es ist die am häufigsten gestellte Frage. Antwort

Frage 2: Hier könnte ihre Frage 2 stehen? Es ist die am zweit häufigst gestellte Frage. Antwort

Frage 3: Hier könnte ihre Frage 2 stehen? Es ist die am dritt häufigst gestellte Frage. Antwort

Frage 1: Hier könnte ihre Frage 1 stehen? Es ist die am häufigsten gestellte Frage. Antwort

Frage 2: Hier könnte ihre Frage 2 stehen? Es ist die am zweit häufigst gestellte Frage. Antwort

Frage 3: Hier könnte ihre Frage 2 stehen? Es ist die am dritt häufigst gestellte Frage. Antwort

Frage 1: Hier könnte ihre Frage 1 stehen? Es ist die am häufigsten gestellte Frage. Antwort

Frage 2: Hier könnte ihre Frage 2 stehen? Es ist die am zweit häufigst gestellte Frage. Antwort

Frage 3: Hier könnte ihre Frage 2 stehen? Es ist die am dritt häufigst gestellte Frage. Antwort

Wahlleiter 3000 - Start (+)

← → ↺ ⚙️ 🔍 🧑

Anmeldung - Digitaler Ausweis

Wie geht das?

Ausweis-ID

Passwort

Anmelden

Wahlkreis 300 - Start

← → ↻ ↺ Wahlkreis 300.de/wahl

🔍 ⚙️ 🏠 👤

Erststimme

Partei 1: Direktmandat 1

Partei 2: Direktmandat 2

Partei 3: Direktmandat 3

Partei 4: Direktmandat 4

Partei 5: Direktmandat 5

Partei 1: Parteislogan

Partei 2: Parteislogan

Partei 3: Parteislogan

Partei 4: Parteislogan

Partei 5: Parteislogan

Fragen? Hier gehts zum FAQ

Stimme abgeben

6.6. Architecture-Design

Fragen

Gibt es eine allgemeine Anwendungsarchitektur, die als Schablone für das zu entwerfende System dienen kann?

- Repository-Architektur

Wie wird das System auf Threads oder Prozessoren verteilt?

- Das macht der Browser/Server automatisch

Welche Architekturmuster sind für das System geeignet?

- Repository-Architektur und Client-Server-Architektur

Mit welcher Strategie wird der Betrieb der Systemkomponenten gesteuert?

- (-)

Wie soll die Systemarchitektur dokumentiert werden?

- Mit UML-Diagrammen vor allem Klassendiagrammen

Welche architektonische Strukturierung ist die beste für die nicht-funktionalen Anforderungen?

- Client-Server-Architektur

Wie werden die Komponenten im System in Teilkomponenten zerlegt?

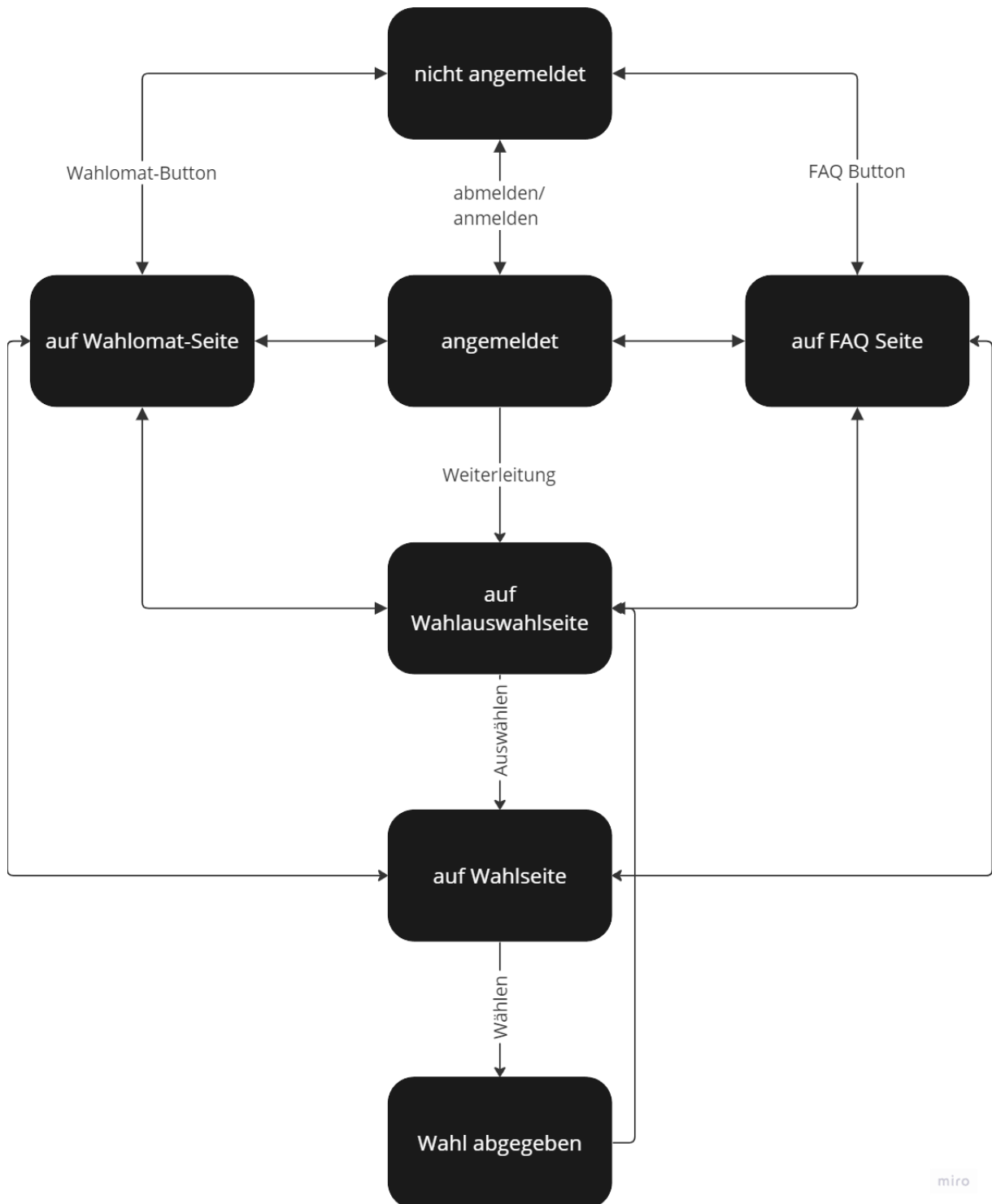
- Die verschiedenen Seiten der Webseite werden aufgeteilt und haben unterschiedliche HTML- und Typescript-Dateien. Außerdem ist das Design abgekapselt und auch der Server ist extern und wird einzeln behandelt.

Wie lautet der grundlegende Ansatz zur Strukturierung des Systems?

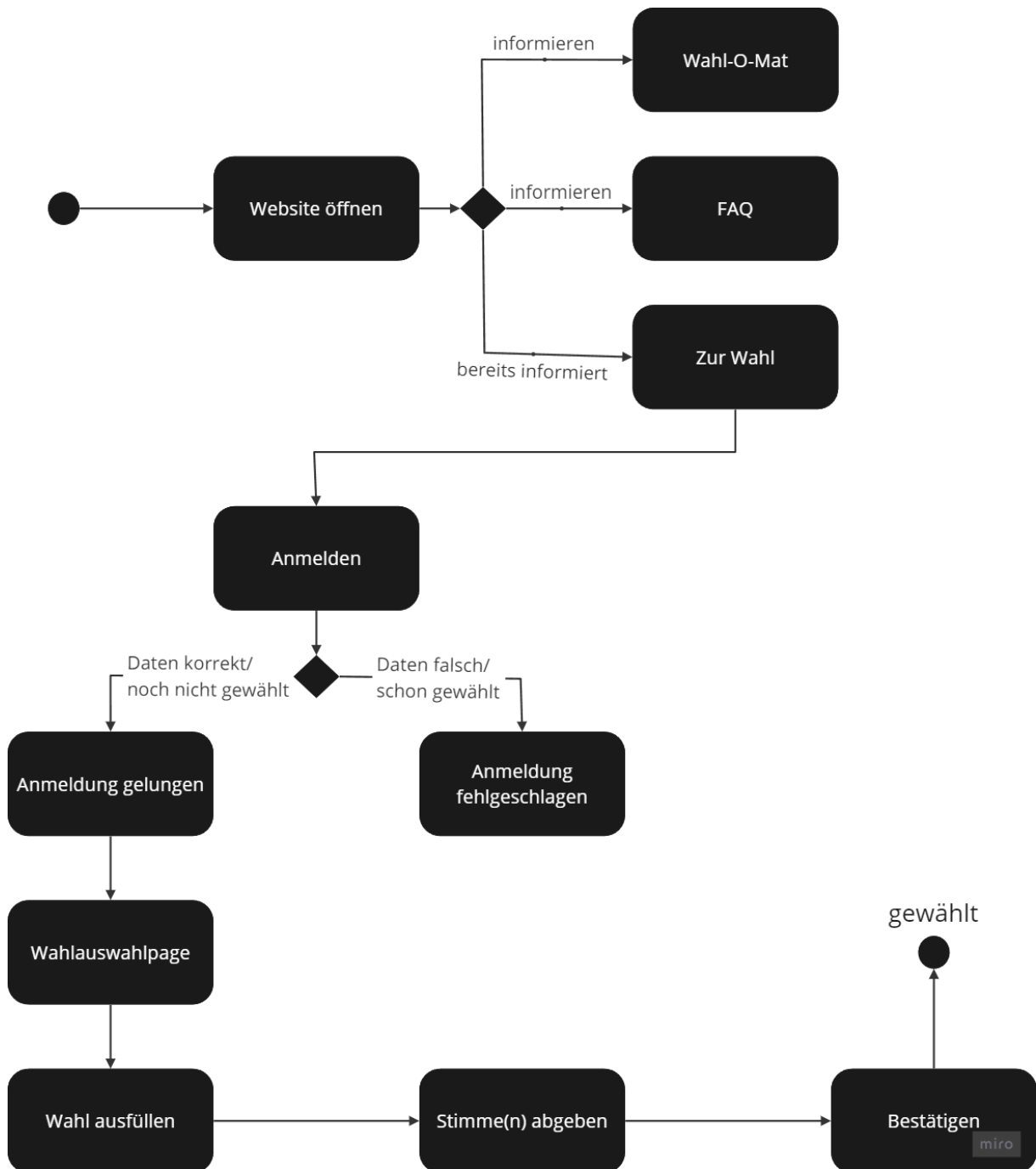
- Die Website in möglichst kleine Komponenten aufteilen

7. UML-Diagrams

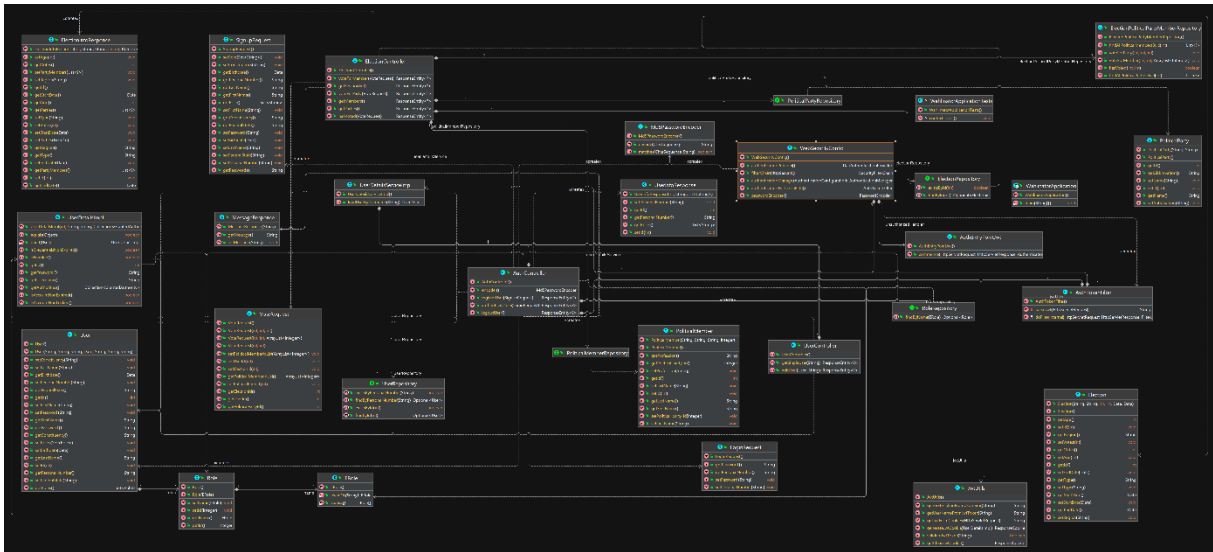
7.1. State Diagram



7.2. Activity diagram



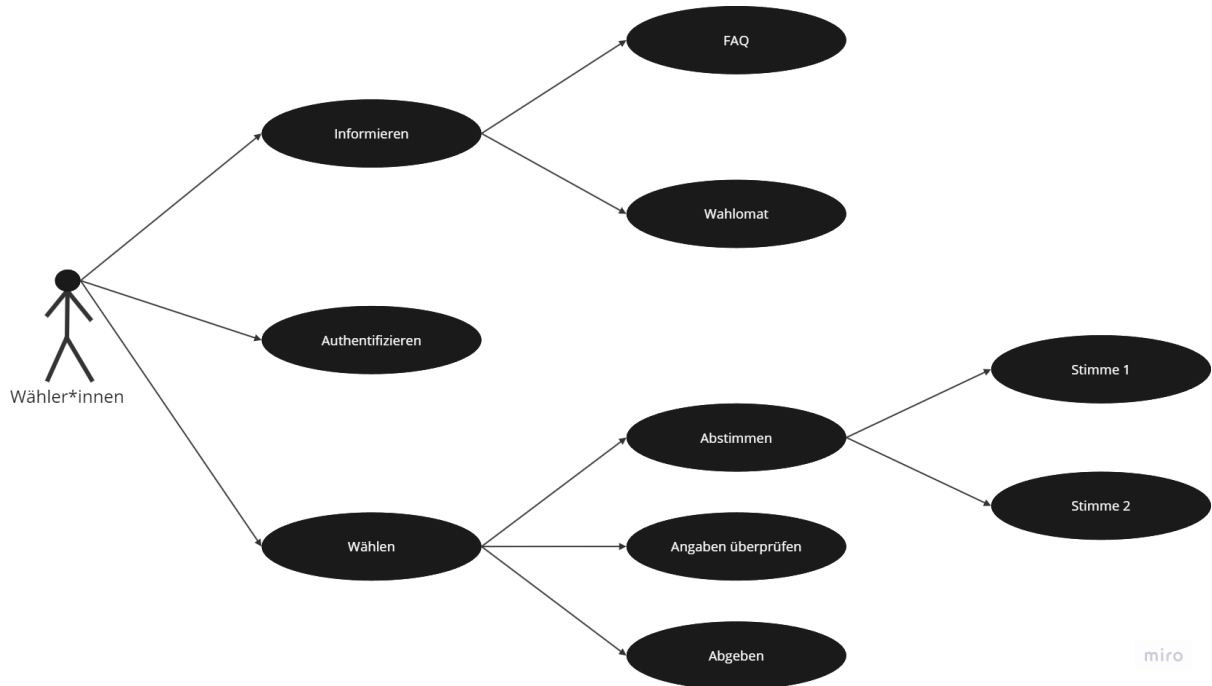
7.3. Class diagram



7.4. Context diagram



7.5. Use-Case diagram



7.6. Sequence diagram

