AquaBalance

### Version 1.0.0, Datum: 5. Juli 2024

Ein Bild, das Handy, Screenshot, Text, mobiles Gerät enthält.

Automatisch generierte Beschreibung

Inhalt

[AquaBalance 1](#_Toc170999152)

[Version 1.0.0, Datum: 5. Juli 2024 1](#_Toc170999153)

[Abstract (Kurzbeschreibung) 3](#_Toc170999154)

[user stories 4](#_Toc170999155)

[Mockups 5](#_Toc170999156)

[Technische Realisierung 6](#_Toc170999157)

[Komponenten der App 6](#_Toc170999158)

[Datenfluss und Speicherung 7](#_Toc170999159)

[Testing 8](#_Toc170999160)

[Testprotokoll 10](#_Toc170999161)

[Fazit 11](#_Toc170999162)

Abstract (Kurzbeschreibung)

Im Alltag stehen Personen häufig vor dem Problem, zu wenig Wasser am Tag zu trinken. Unsere App soll diesem Problem ein wenig entgegenwirken und die Nutzer bei der auszureichenden Flüssigkeitseinnahme zu unterstützen.

Jeder Nutzer kann individuelle Informationen wie beispielsweise Gewicht oder Körpergrösse angeben. Aufgrund dessen berechnet die App automatisch den individuellen Tagesbedarf. Ausserdem wird dem Nutzer die verbleibende Menge für einen gesättigten Bedarf unterhalb des momentanen Standes angezeigt. Schlussendlich kann ein Verlauf der letzten sieben Tage auf einer separaten Seite angezeigt werden.

user stories

An dieser Stelle alle Anforderungen als User Stories auflisten:

* Als Nutzer soll ich meine Körpergrösse angeben können
* Als Nutzer soll ich mein Alter angeben können
* Als Nutzer soll ich mein Geschlecht angeben können
* Als Nutzer soll mir das Tagesziel permanent angezeigt werden
* Als Nutzer soll es möglich sein, die getrunkene Menge angeben zu können.
* Sollte ich als Nutzer mit der Eingabe der Informationen in meinem Profil fertig, sollte ich die Informationen bearbeiten können
* Der gesamte Wasserbedarf sollte stets in meinem Profil abrufbar sein
* Sollte ich als Nutzer eine Fehleingabe tätigen, so soll mir die App eine Fehlerrückmeldung zustellen.
* Als Nutzer soll es mir möglich sein, den Wochenverlauf abzurufen

Mockups

An dieser Stelle den Prototypen deiner App beschreiben. Füge einen Printscreen deiner Screens ein und beschreibe jeden Screen einzeln.

Ein Bild, das Text, Screenshot, Handy, mobiles Gerät enthält.

Automatisch generierte Beschreibung

**Profil-Seite: Home**

Auf dieser Seite ist es dem Nutzer möglich seine persönlichen Daten in die App einzutragen. Es ist auch direkt die erste Seite, die dem Nutzer angezeigt wird, damit es direkt losgehen kann. Nach Klick auf den «Calculate»- Button wird aufgrund der zur Verfügung gestellten Daten der individualisierte Wasserbedarf ausgegeben.

Ein Bild, das Text, Handy, Screenshot, mobiles Gerät enthält.

Automatisch generierte Beschreibung

**Profil-Seite: Detailansicht**

Sollte der persönliche Bedarf ausgerechnet worden, werden dem Nutzer alle relevanten Daten aufgelistet inklusive des persönlichen, täglichen Wasserbedarfs. Dazu kann der Nutzer hier nach Belieben seine Angaben bearbeiten

Ein Bild, das Handy, Screenshot, Text, mobiles Gerät enthält.

Automatisch generierte Beschreibung

**Home-Seite: Home**

Hier kann der Nutzer seinen derzeitigen Stand und Fortschritt überprüfen. Bei der Fortschrittsleiste sind die noch verbleibende Menge als auch der Fortschritt in Prozent ersichtlich. Der Fortschritt wird ebenfalls in der Flasche dargestellt.

Ein Bild, das Text, Handy, Screenshot, mobiles Gerät enthält.

Automatisch generierte Beschreibung

**Home-Seite: Regelmässiger Eintrag**

Auf dieser Seite kann der Benutzer nun fortlaufend durch den Tag seine eingenommene Menge an getrunkenem Wasser eintragen. Er kann zwischen diversen Arten von Behältern auswählen und die Menge je nachdem variieren. Nach Eintrag wird ausserdem ein Timestamp gesetzt, an dem ein Eintrag erstellt wurde

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

Automatisch generierte BeschreibungHistory-Seite**

Auf der History-seite sieht der Benutzer die eingetragenen Daten von der Startseite. Dort wird eine Liste mit den eingetragenen Litern, dem Datum und der Uhrzeit angezeigt. Ausserdem hat der Benutzer die Möglichkeit, einzelne Einträge aus der Liste zu löschen. Ein neuer Abschnitt wird nur dann hinzugefügt, wenn der Benutzer auf der Startseite auf "Reset" drückt oder wenn 24 Stunden seit dem letzten Eintrag vergangen sind.

**Mockup mit Prototyp Pfeilen:**

Ein Bild, das Text, Screenshot, Multimedia, Software enthält.

Automatisch generierte BeschreibungUnterhalb ist das Mockup ersichtlich, jedoch mit Prototyp Pfeilen. Damit ist ersichtlich, was aufgerufen wird, falls bestimmte Eingaben getätigt werden.

Technische Realisierung

### Komponenten der App

**React Native:**

Die App wurde mit React Native entwickelt, um eine plattformübergreifende Unterstützung zu gewährleisten.

**Navigation:**

Die Navigation innerhalb der App erfolgt über das @react-navigation Paket. Ein BottomTabNavigator wird verwendet, um zwischen den Hauptbildschirmen (Home, Profile und History) zu wechseln.

**UI-Bibliotheken:**

Komponenten wie *react-native-paper* *und react-native-vector-icons* werden für die Benutzeroberfläche verwendetm, um eine möglichst breitflächige Auswahl an Komponenten zu erhalten und so das Design nach Bedarf zu gestalten

**Home-Bildschirm:**

Ermöglicht den Benutzern, ihre Wasseraufnahme zu protokollieren. Eine Fortschrittsanzeige zeigt den aktuellen Stand im Verhältnis zur täglichen Wasserbedarfsempfehlung. Des Weiteren wird stets der noch verbleibende Wert angezeigt.

**Profil-Seite:**

Benutzer können Körpergewicht, Aktivitätszeit, und Klimaverhältnisse angeben. Basierend auf diesen Werten, rechnet das Programm per Kalkulationsfunktion den täglichen Wasserbedarf aus. Nach der Berechnung erscheint ein Modal mit dem entsprechenden Wasserbedarf. Dieser wird ebenfalls im AsyncStorage gespeichert und auf die Home Seite als Wert übergeben.

**History:**

**Einträge aus der Home-Seite werden aus dem AsyncStorage ausgelesen und hier als Listenansicht dargestellt. Sobald der Fortschritt zurückgesetzt wird, erstellt die History automatisch einen neuen Abschnitt.**

**AsyncStorage:**

Die App nutzt @react-native-async-storage/async-storage, um Benutzerdaten lokal auf dem Gerät zu speichern. Dies umfasst die tägliche Wasseraufnahme, individuelle Wasserbedarfswerte und andere Benutzereinstellungen.

States werden mithilfe von React Hooks (useState, useEffect) verwaltet.

**Datenfluss und Speicherung**

Die Daten der App werden hauptsächlich lokal gespeichert und verarbeitet. Beim Starten der App werden gespeicherte Daten aus AsyncStorage geladen und in den entsprechenden States abgelegt. Änderungen an den Benutzerdaten (z.B. Protokollierung der Wasseraufnahme) werden ebenfalls in AsyncStorage gespeichert, um die Konsistenz der Daten über Sitzungen hinweg zu gewährleisten.

**Laden der Daten:**

* + Beim Initialisieren der Komponenten werden Daten aus AsyncStorage geladen (z.B. loadLoggedAmount in der Home-Komponente).
  + Die geladenen Daten werden in den entsprechenden States abgelegt (z.B. setLoggedAmount).

**Speichern der Daten:**

* + Bei Änderungen (z.B. neue Wasseraufnahme) werden die aktualisierten Werte in den States gespeichert.
  + Gleichzeitig werden diese Werte in AsyncStorage geschrieben, um die Änderungen persistent zu speichern (z.B. saveLoggedAmount).

Testing

|  |  |
| --- | --- |
| **Abschnitt** | **Inhalt** |
| **ID** | **ST01** |
| Anforderungen | Als Benutzer möchte ich meine tägliche Wasseraufnahme registrieren, um meinen Fortschritt zu verfolgen. |
| Vorbedingungen | * Die App ist installiert und gestartet. * Der Benutzer ist eingeloggt. * Die Home-Seite ist geöffnet. |
| Ablauf | 1. Benutzer klickt auf den "Plus" Button. 2. Benutzer gibt eine Wassermenge in das Eingabefeld ein. 3. Benutzer klickt auf den "Hinzufügen" Button. |
| Erwartetes Resultat | * Die eingegebene Wassermenge wird zur heutigen Gesamtaufnahme hinzugefügt. * Der Fortschrittsbalken wird entsprechend aktualisiert. * Die neue Gesamtaufnahme wird in AsyncStorage gespeichert. |

|  |  |
| --- | --- |
| ID | ST02 |
| Anforderungen | Als Benutzer möchte ich meine Wasseraufnahme der letzten Woche in einer List einsehen |
| Vorbedingungen | * Die App ist installiert und gestartet. * Der Benutzer ist eingeloggt. * Es wurden in den letzten paar Tagen Wasseraufnahmen eingegeben. |
| Ablauf | 1. Benutzer navigiert zur "History" Seite. |
| Erwartetes Resultat | Eine Listenansicht zeigt die eingetragenen Wassermengen an |

|  |  |
| --- | --- |
| ID | ST03 |
| Anforderungen | Als Benutzer möchte ich meinen täglichen Wasserbedarf anpassen können. |
| Vorbedingungen | * Die App ist installiert und gestartet. * Der Benutzer ist eingeloggt. * Die Profil-Seite ist zugänglich. |
| Ablauf | 1. Benutzer navigiert zur Profil-Seite. 2. Benutzer aktualisiert seine Daten 3. Benutzer speichert die Änderungen. |
| Erwartetes Resultat | Der neue tägliche Wasserbedarf wird gespeichert und auf der Home-Seite angezeigt. |

|  |  |
| --- | --- |
| ID | ST04 |
| Anforderungen | Als Benutzer möchte ich, dass meine registrierte Wasseraufnahme beim Starten der App geladen wird. |
| Vorbedingungen | * Die App ist installiert. * Der Benutzer hat die App zuvor verwendet und Wasseraufnahmen registriert. |
| Ablauf | 1. Benutzer startet die App. |
| Erwartetes Resultat | * Die Home-Seite zeigt die korrekte Gesamtaufnahme und den entsprechenden Fortschrittsbalken an. * Die gespeicherte Wasseraufnahme wird aus AsyncStorage geladen. |

|  |  |
| --- | --- |
| ID | ST05 |
| Anforderungen | Als Benutzer möchte ich einen Fehler erhalten wenn ich bei meinem Gewicht einen unrealistischen Wert eingebe. |
| Vorbedingungen | * Die App ist installiert. * Der Benutzer hat die App zuvor verwendet und Wasseraufnahmen registriert. |
| Ablauf | 1. Benutzer navigiert zur Profil-Seite. 2. Benutzer gibt bei Gewicht eine Zahl an die kleiner als 40kg oder grösser als 400kg ist. |
| Erwartetes Resultat | * Auf der Profil-Seite erscheint einen Fehlercode, der darauf hinweist, dass, das Gewicht zwischen 40 und 400kg sein muss. |

Testprotokoll

In diesem Kapitel wird die Testausführung protokolliert. Welche Tests wurden wann von wem mit welchem Resultat durchgeführt.

Hier ein Beispiel eines Eintrages:

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Person / Datum | Erfolgreich | Bemerkungen |
| ST03 | Jonas Graf 04.07.2024 | Ja | Keine Bemerkungen |
| ST01 | Vincent Witzmann  04.07.2024 | Ja | Funktionierte problemlos |
| ST04 | Luka Ljubisavljevic  04.04.2024 | Ja | Funktioniert |
| ST02 | Sergio Soares  05.07.2024 | ja | Löschen von individuellen eintrage möglich aber ganze abschnitte nicht. |
| ST05 | Liam Roth | Ja | Hat geklappt, sogar Kommazahlen möglich. SUPER! Weiter so! |

Fazit

Hier kommt eure Reflexion zum Projekt.

* **Was lief gut/schlecht?**

**Positiv:**

In den 1,5 -2 Tagen Projektarbeit haben wir als Gruppe viele Hürden überwunden. Positiv, in der Reflexion aufgefallen, ist unsere Handhabung mit Fehlern und Blockaden. Sollte ein Teammitglied länger an einem Problem gesessen haben und nicht weitergekommen sein, so hat ein anderes Teammitglied die Aufgaben mit ihm getauscht und sich diesem Problem gewidmet. Daher konnten wir einen effizienten Team-Workflow gewährleisten. Ausserdem erklärte ein jeder den anderen stets jegliche Funktion, welche Fragen aufwarfen oder ziemlich komplex waren, so dass jedes Teammitglied den Code nachvollziehen konnte.

**Negativ:**

Das einzig Negative, das zu verkennzeichnen ist, war die Problematik mit Git. Es war zwar möglich Inhalte auf Git zu pushen, jedoch war es uns nicht möglich den lokalen Ordnerinhalt mit dem Repository Inhalt zu überschreiben. Dadurch mussten wir grössere Änderungen per Teams Chat einander zusenden.

* **Wie seid ihr mit dem Endergebnis zufrieden?**

Schlussendlich wurden alle Funktionalitäten fast wie geplant umgesetzt. Die App Funktionalitäten sind vollumfänglich verfügbar und funktionieren. Dazu realisierten wir eine schönes Farbdesign, welches zum Inhalt unserer App passt. Wir sind stolz darauf, dass wir trotz Herausforderungen im Entwicklungsprozess unsere Ziele erreicht haben. Besonders zufrieden sind wir mit dem ansprechenden blauen Farbdesign, das nicht nur visuell ansprechend ist, sondern auch den Inhalt und die Benutzerfreundlichkeit unserer App unterstützt. Durch genügende Tests und Anpassungen konnten wir die App stabil und benutzerfreundlich gestalten. Insgesamt sind wir sehr zufrieden mit dem Endergebnis und den Erfahrungen, die wir während dieses Projekts gesammelt haben.

* **Was habt ihr gelernt?**

Wir haben die Grundlagen und Techniken der mobilen App Entwicklung mit React Native gelernt, dazu auch die Nutzung von Komponenten, Hooks und Navigation. Wir haben effizientes Datenmanagement durch den Einsatz von AsyncStorage gemacht und benutzerfreundliche Oberflächen gestaltet. Ausserdem haben wir gelernt, systematisch Fehler zu identifizieren und zu beheben, was unsere Debugging-Fähigkeiten verbessert hat.

* **War alles vorhanden oder was fehlte noch?**

Wir haben die History-Seite abgeändert. Statt einer grafischen Darstellung in Form von Balken wie wir es im Mockup dargestellt haben, gibt es jetzt einfach eine Liste. Dies wurde gemacht, weil die Umsetzung der Grafiken zu kompliziert war und wir zu wenig Zeit hatten.