

SWE1-Projekt

Team09

Oliver Wendt -	41033
Emircan Aslan -	40568
Florian Radtke -	40640
Dominic Wittop -	40644
Kevin Ziegler -	41111

Inhaltsverzeichnis

Einleitung	3
Aufgabenstellung	3
Wer hat das Projekt bearbeitet?	3
Unsere Idee.....	3
Unser Projekt	4
Modellierung unserer Anwendung.....	5
UML-Anwendungsfalldiagramme	5
Was ist ein Anwendungsfalldiagramm und was sind die Funktionen?	5
UML-Aktivitätsdiagramme.....	7
UML-Klassendiagramm	9
Wireframe	10
Was ist ein Wireframe?	10
Umsetzung.....	11
Prototyp.....	13
Screenshots.....	13
Rechner.sh	13
Login.sh	16
Fazit	17
Literaturverzeichnis	17

Einleitung

Aufgabenstellung

Modelliert im Team ein kleines IT-System (z.B. einen Aufgabenplaner, ein System zur Berechnung des digitalen CO₂-Fußabdrucks eurer Teammitglieder oder ähnliches) mit verschiedenen UML-Modellen (mindestens mit Anwendungsfalldiagramm, Aktivitätsdiagrammen mit Schwimmbahnen und Objekten, Klassendiagramm) und mit einem User Interface-Prototypen, der zumindest eine Ausgabe auf einer html-Seite simuliert (unter Verwendung von HTML/CSS und Bash-Skripten oder Java).

Wer hat das Projekt bearbeitet?

Das Projekt wurde vom gesamten Team09 bearbeitet. Zu den Mitgliedern gehören: Oliver Wendt, Emircan Aslan, Florian Radtke, Dominic Wittop und Kevin Ziegler. Die Zusammenarbeit und Kommunikation fanden hauptsächlich Online statt. Allerdings gab es auch einige Sitzungen vor Ort in der Hochschule.

(Dominic Wittop)

Unsere Idee

Wie wir bereits durch mehrere Institutionen in Erfahrung bringen konnten, verbreitet sich das Thema Adipositas immer mehr in Europa und der Welt. “[...] Übergewicht und Adipositas sind wichtige Merkmale zur Beschreibung des Gesundheitszustands einer Bevölkerung. In den letzten Dekaden hat die Prävalenz von Übergewicht und Adipositas weltweit zugenommen und ist daher von bedeutender PublicHealth-Relevanz.”¹ Deshalb haben wir, als Team 09, es und als Aufgabe gemacht diesem entgegenzuwirken. Unserer Grundidee ist ein einfacher, aber doch sehr effektiver Kalorienrechner, der es den Nutzern ermöglichen soll, sich Ziele entsprechend der eigenen Essgewohnheiten zu setzen.

¹ RKI, 2017

Unser Projekt

(Florian Radtke)

Für viele ist es schwer, den Überblick über Kalorienverbrauch zu behalten und die eigenen Gewichtsziele zu verfolgen. Deswegen möchten wir mit Projekt diese Leute mit einer einfachen und kostenlosen Möglichkeit unterstützen, ihre persönlichen Gewichtsziele zu erreichen und ihnen dabei eine Übersicht über ihre Kalorienveränderung der letzten Woche geben. Der Nutzer bekommt hierbei die Möglichkeit, regelmäßig und freiwillig seine Daten aufzuzeichnen, was auch als "Self-Tracking" bezeichnet wird.²

Bei der Website muss man sich zunächst einmal anmelden, damit die Daten über einen Zeitraum von einer Woche gespeichert und wieder aufgerufen werden können. Dabei wird der input-type password benutzt, damit das Passwort während der Anmeldung nicht angezeigt wird. Mit der Anmeldung wird anschließend für jeden neuen Nutzer automatisch ein Ordner erstellt, in dem seine Daten gespeichert werden. Nach der Anmeldung hat der Nutzer die Möglichkeit über einen Zeitraum von einer Woche verschiedenen Daten zu persönlichen Informationen (z.B. Gewicht, Alter Geschlecht) Aktivitäten und Mahlzeiten, sowie sein persönliches Ziel anzugeben, ob er zum Beispiel abnehmen oder zunehmen möchte. Anhand dieser Daten wird dann der Grundumsatz, sowie die Anzahl an verbrannten und aufgenommenen berechnet und angezeigt. Außerdem wird mit Hilfe der Berechnung eine tägliche Empfehlung für das Erreichen der Zielkalorien gegeben. Anschließend kann dann am Ende der Woche eine Wochenbilanz erstellt werden, welche zeigt, ob und wann man sein Ziel erreicht hat und eine Übersicht über die Werte der letzten Woche gibt.

Für das Projekt haben wir dabei nur HTML, CSS und Bash-/CGI-Skripte benutzt. Außerdem haben wir die Aufgaben auf alle Leute verteilt, sodass jeder etwas in dem Bereich machen kann, indem er am besten ist, aber trotzdem jeder noch etwas dazu beiträgt.

² Vgl. Christian Schöttler, Self-Tracking zwischen Emanzipation und digitaler Überwachung, 23.02.2024

Modellierung unserer Anwendung

UML-Anwendungsfalldiagramme

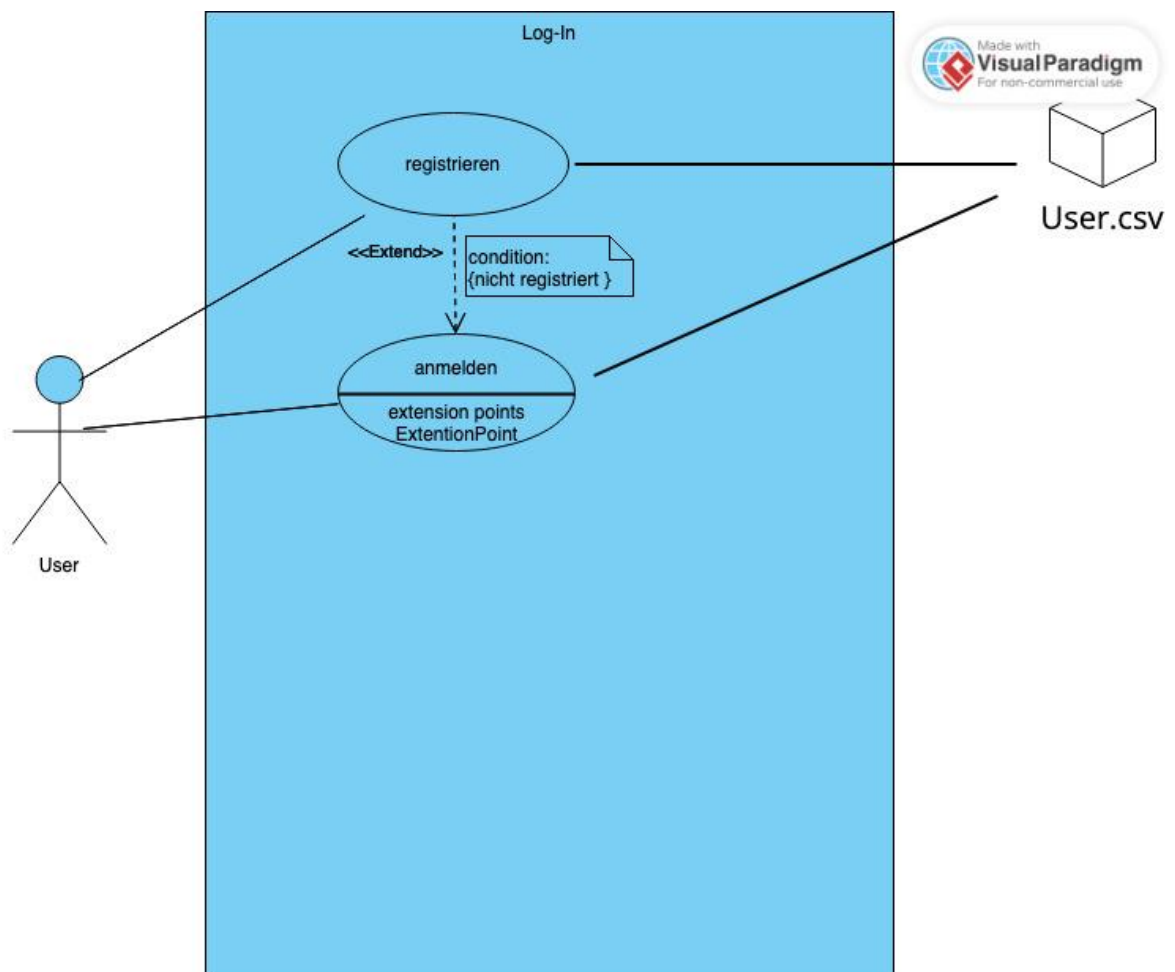
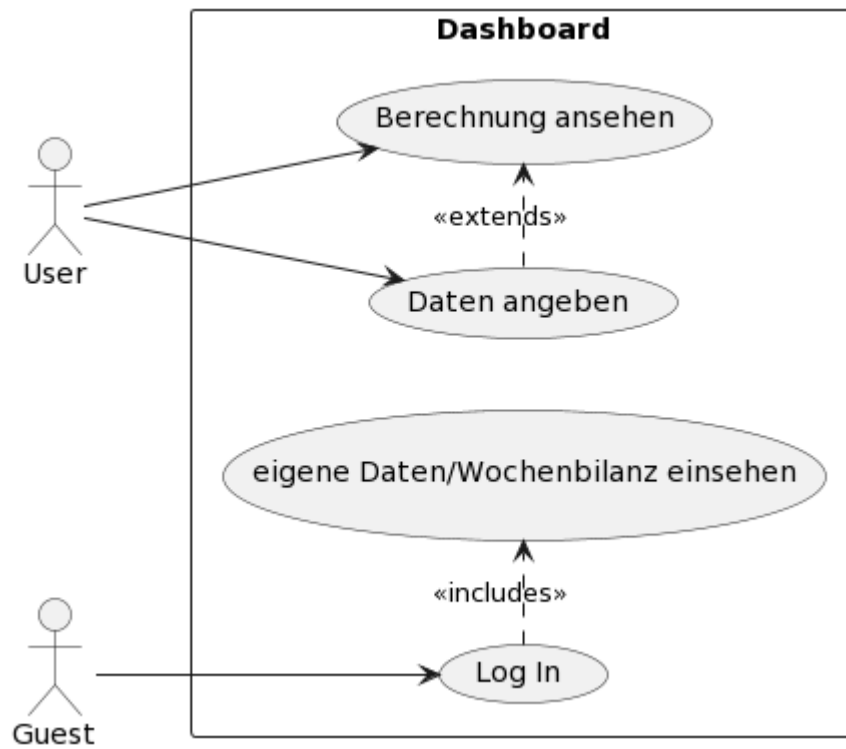
(Dominic Wittop)

Was ist ein Anwendungsfalldiagramm und was sind die Funktionen?

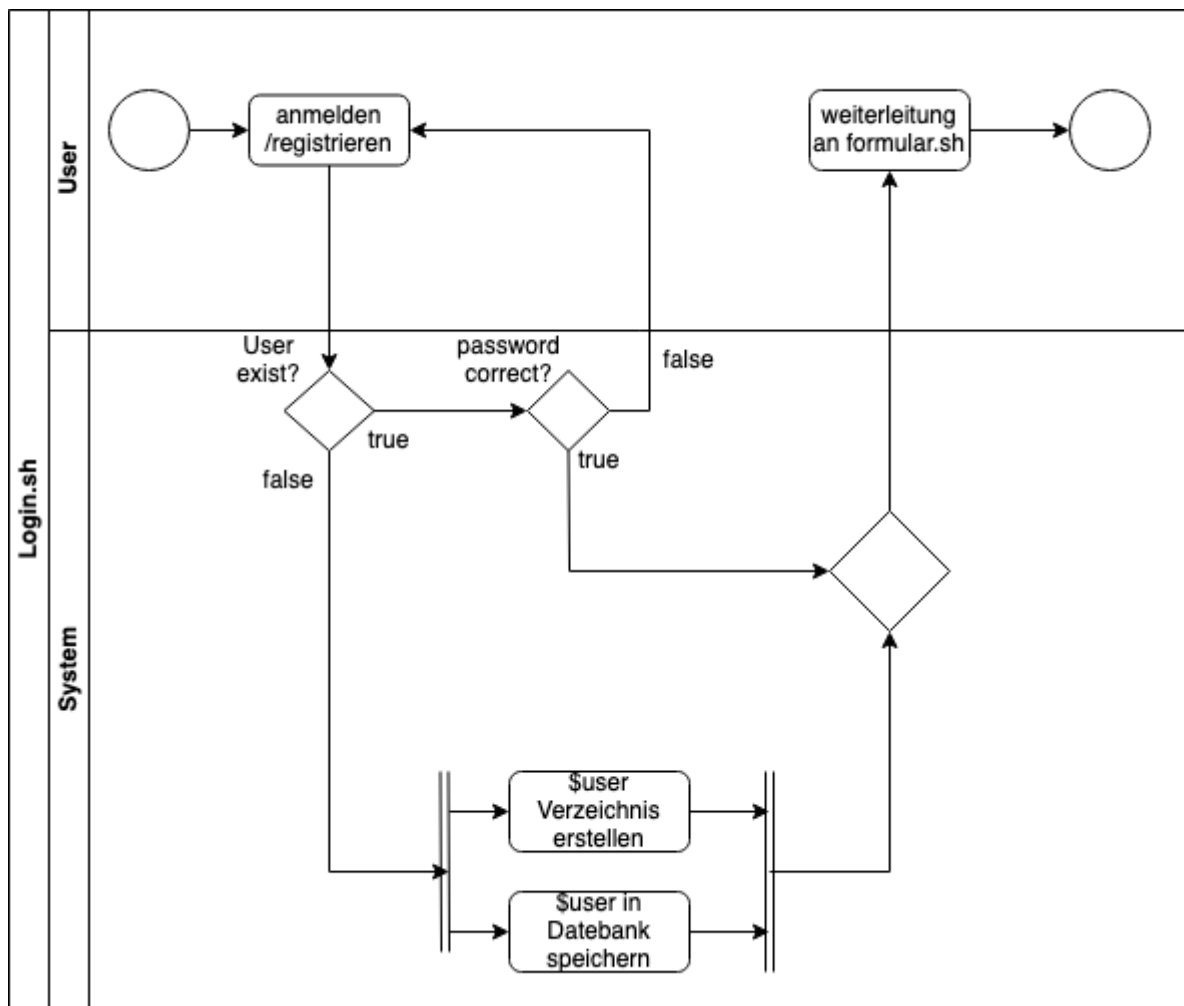
Anwendungsfalldiagramme sind Modelle, die in der Softwareentwicklung verwendet werden, um die möglichen Aktionen des Nutzers vereinfacht und grafisch darzustellen. “Anwendungsfalldiagramme dienen zur Beschreibung der allgemeinen Funktionen und des Geltungsbereichs eines Systems. Diese Diagramme können außerdem zur Identifikation der Interaktionen zwischen dem System und seinen Akteuren verwendet werden. Die Anwendungsfälle und Akteure in Anwendungsfalldiagrammen beschreiben, welche Operationen das System ausführt und wie die Akteure es verwenden, jedoch nicht, wie das System intern arbeitet”³. Der Hauptgrund, für die Verwendung solcher Diagramme ist die Erleichterung des Verständnisses über die Anforderungen des Systems und welche Funktionen es mit sich bringt.

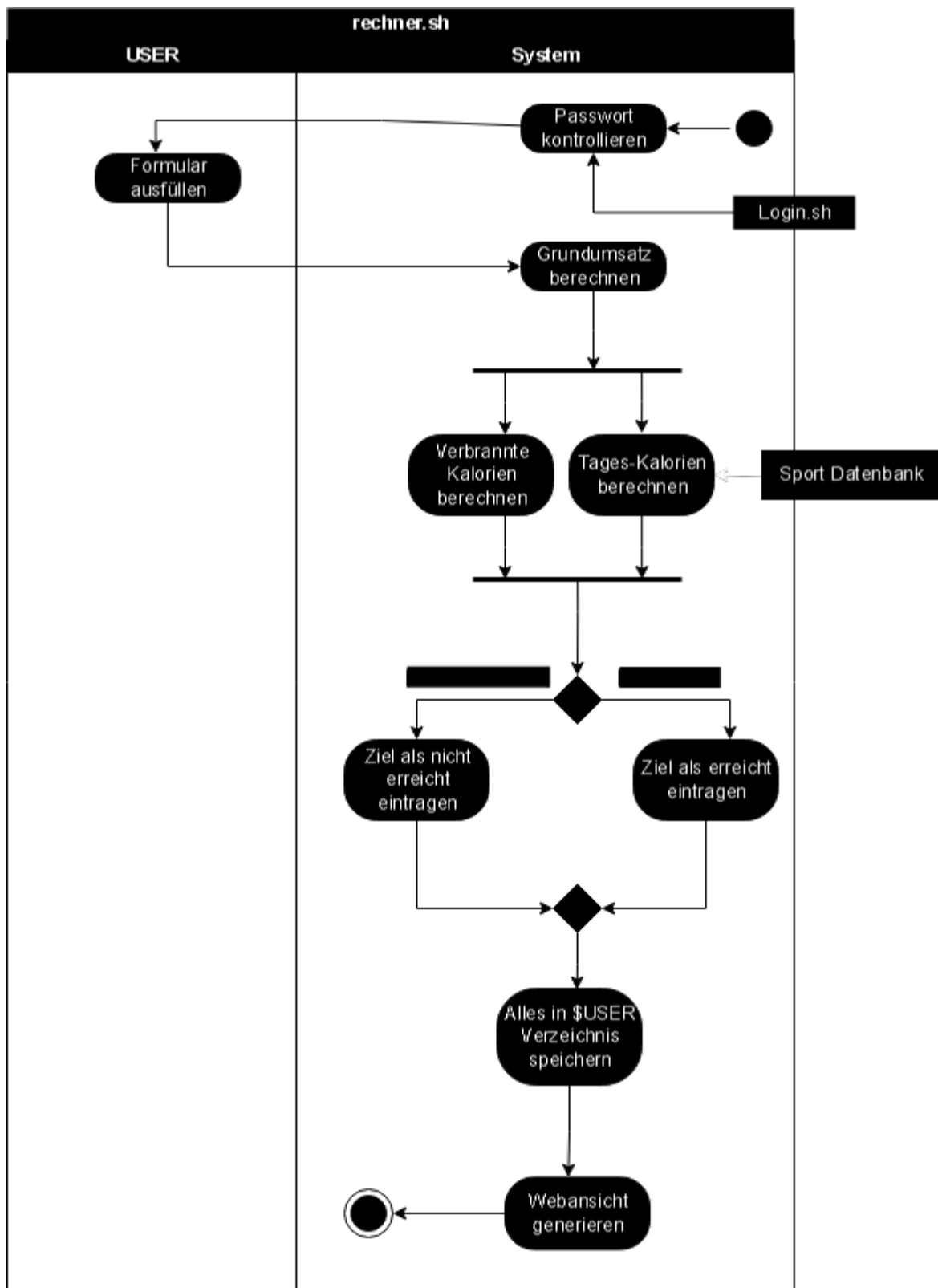
Unser Anwendungsfalldiagramm dient zur Veranschaulichung, wie der Nutzer mit unserer Webseite interagiert. Die beiden Akteure unseres Diagramms sind zum einen der Guest vor dem Log-in, und zum anderen der User selbst, sobald er sich eingeloggt hat. Durch das Einloggen oder Erstellen des eigenen Accounts kann man dann die Berechnung vom eigenen Kalorienverbrauch starten. Der User muss, um zur Berechnung fortzufahren, das Geschlecht, Alter, Körpergewicht, Ziel, die Größe, Dauer, letzten Mahlzeiten und die Sportart, die der Nutzer tätigt, angegeben haben. Sobald ein Account erstellt wurde, kann der User jederzeit erneut auf seine Daten zugreifen und diese verändern. Die Daten aus den Anwendungsfällen werden dann in die csv-Datei geschrieben und zur Berechnung weitergegeben.

³ IBM, Anwendungsfalldiagramme

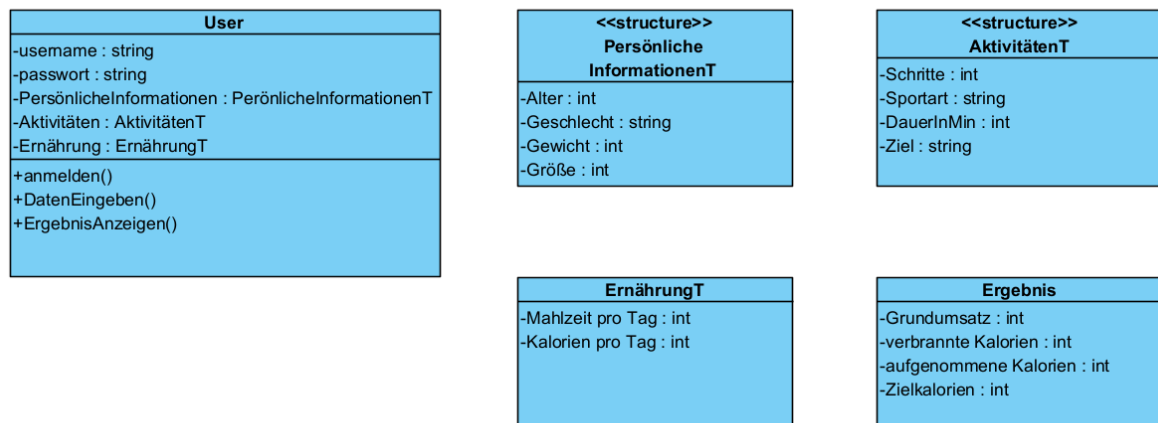


UML-Aktivitätsdiagramme





UML-Klassendiagramm



(Florian Radtke)

In Klassendiagrammen werden die verschiedenen Objekte, des Projektes und ihre Eigenschaften dargestellt. Die Klasse ist dabei der abstrakte „Obergriff für Dinge (Objekte), die gemeinsamen Eigenschaften haben.“⁴ Eigenschaften hingegen sind Attribute, wie zum Beispiel Position, Größe oder Farbe, welche unterschiedlich Ausprägung oder Werte haben. Bei der Größe können diese beispielsweise beliebige Zahlen wie 1 oder 12 sein, welche dann durch den Datentyp `int` dargestellt werden. Außerdem können Klassen auch bestimmte Operationen zugeordnet werden.

Es gibt hier die die Klasse `User`, welcher die Attribute `username`, `passwort`, `persönliche Informationen`, `Aktivitäten` und `Ernährung` hat und die Operationen `anmelden`, `Daten eingeben`, sowie `Ergebnis anzeigen`, nutzen kann. Drei der Attribute werden anschließend in den elementaren Klassen `Persönliche Informationen`, `Aktivitäten` und `Ernährung` genauer definiert. Die Klassen, `Persönliche Informationen` und `Aktivitäten` sind hierbei eine `structure`, da sie sich aus mehreren Typen zusammensetzen. Außerdem gibt es die Klasse `Ergebnis`, welches die Attribute `Grundumsatz`, `verbrannte Kalorien`, `aufgenommene Kalorien` und `Zielkalorien` besitzt.

⁴ Prof. Dr. Ulrike Erb, SWE 1 – Modellierung UML-Klassendiagramme Notationen und Beispiele S.11

Wireframe

(Kevin Ziegler)

Was ist ein Wireframe?

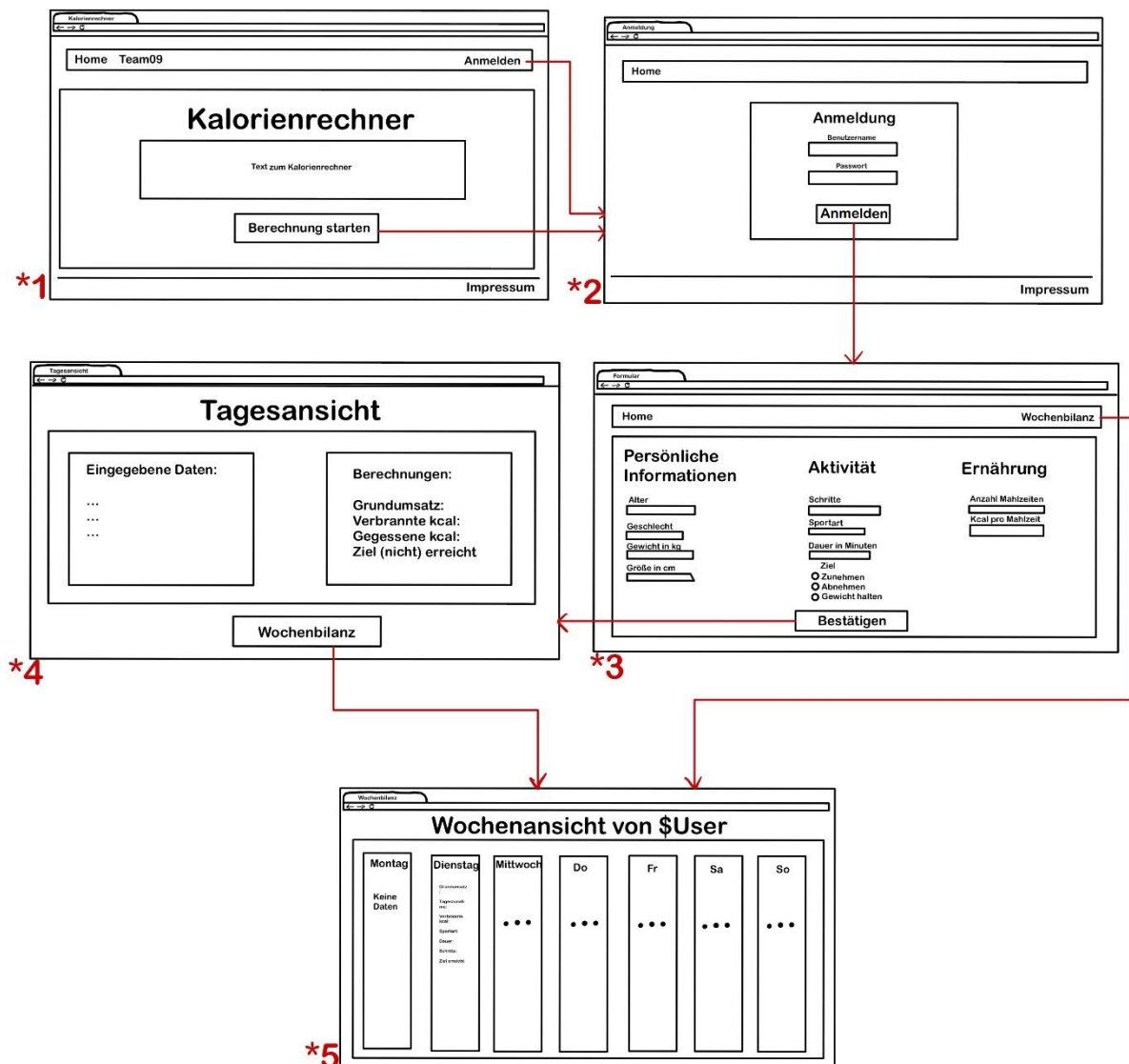
Mit einem “Wireframe[...] [kann man] die Struktur der Bedienoberfläche, insbesondere die Aufteilung in einzelne Screens und Navigationsmöglichkeiten [zeigen], ohne schon genau auf visuelle Aspekte wie Farben, Größen, etc. Einzugehen.”⁵

Sie zeichnen sich unter anderem durch die skizzenhafte Darstellung, niedrige Detailgenauigkeit und einem benutzerzentrierten Ansatz aus.

Es ist ein wichtiger Bestandteil der Umsetzung eines Projektes, da man so im Vorhinein schon die grobe Struktur (in unserem Fall, der Website) modellieren kann. So hat man bereits valide Anhaltspunkte, an welchen man sich bei der tatsächlichen Programmierung und Gestaltung der Website orientieren kann. Dies ist zum einen zeitsparend und ermöglicht zum anderen eine effiziente Zusammenarbeit im Team, da eine Struktur vorgegeben wird, an welche sich gehalten werden muss und somit weniger Unstimmigkeiten und Differenzen entstehen.

⁵ Dehlinger J, Dixon J (2011) Mobile application software engineering: Challenges and research directions. Workshop on Mobile Software Engineering

Umsetzung



(Kevin Ziegler)

Das Wireframe-Diagramm stellt die einzelnen Seiten unseres Prototyps da. Zu Beginn landet man auf der Landingpage (siehe Bild *1). Von dort aus hat man die Möglichkeit zur Team-Website zu gelangen, sich direkt anzumelden oder die Seite des Impressums (der Hochschule) aufzurufen. Wenn man sich dazu entscheidet, sich anzumelden, gelangt man auf die Anmeldeseite (siehe Bild *2). Dort muss man sich mit seinen Userdaten anmelden bzw. registrieren. Hat man dies erledigt, geht es weiter zum Formular (siehe Bild 3*). Dieses ist aufgeteilt in drei Abschnitte: Persönliche Informationen, Aktivität und Ernährung. Von dieser Seite aus hat man die Möglichkeit: seine Informationen einzutragen, oder direkt zur Wochenansicht (dazu gleich mehr) zu gelangen. Wenn man seine Informationen einträgt und mit dem "Bestätigen"-Button bestätigt, gelangt man zur Tagesansichtsseite (siehe Bild *4). Dort werden im linken Block die angegebenen Informationen und im rechten Block die berechneten Daten angezeigt. Von dort aus hat man über einen Button auch die Möglichkeit, auf die Wochenansichtsseite (siehe Bild *5)

zu gelangen. Auf dieser Seite wird für jeden Wochentag (Mo-So) eine Box generiert. In diesen Boxen werden nun die gespeicherten Daten des Users der letzten Tage angezeigt. Hat der User an einem Tag keine Daten angegeben, zeigt die jeweilige Box “Es liegen keine Daten für diesen Tag vor” an.

Prototyp

Screenshots

Rechner.sh

```
38 if [ -f "$USER_data/$USER-Mon.csv" ]; then
39 #grundumsatz|daily_calo|burned|sport|sport_minutes|steps|ausgabe|
40 grundumsatz=$(cat "$USER_data/$USER-Mon.csv" |cut -d';' -f2)
41 daily_calo=$(cat "$USER_data/$USER-Mon.csv" |cut -d';' -f3)
42 burned_calo=$(cat "$USER_data/$USER-Mon.csv" |cut -d';' -f4)
43 sport=$(cat "$USER_data/$USER-Mon.csv" |cut -d';' -f5)
44 sport_minutes=$(cat "$USER_data/$USER-Mon.csv" |cut -d';' -f6)
45 steps=$(cat "$USER_data/$USER-Mon.csv" |cut -d';' -f7)
46 goal=$(cat "$USER_data/$USER-Mon.csv" |cut -d';' -f8)
47 echo "<p>Grundumsatz: $grundumsatz kcal</p>"
48 echo "<p>Tageszunahme: $daily_calo kcal</p>"
49 echo "<p>verbrannte Kalorien: $burned_calo kcal</p>"
50 echo "<p>Sportart: $sport</p>"
51 echo "<p>Dauer in Minuten: $sport_minutes</p>"
52 echo "<p>Sie sind an diesem Tag $steps Schritte gelaufen</p>"
53 if [[ $goal == "lose" ]]; then
54 echo "<p>Das Ziel wurde an diesem Tag nicht erreicht</p>"
55 else
56 echo "<p>Das Ziel wurde an diesem Tag erreicht</p>"
57 fi
58 else
59 echo "<p>Es liegen keine Daten für diesen Tag vor</p>"
60 fi
```

```

60 #Berechnungen - Grundumsatz
61 if [[ "$GENDER" == "maennlich" ]]; then
62   grundumsatz=$(echo "scale=2; 88.362 + (13.397 * $WEIGHT) + (4.799 * $HEIGHT) - (5.677 * $AGE)" | bc)
63 elif [[ "$GENDER" == "weiblich" ]]; then
64   grundumsatz=$(echo "scale=2; 447.593 + (9.247 * $WEIGHT) + (3.098 * $HEIGHT) - (4.330 * $AGE)" | bc)
65 fi
66 echo "<p>Dein Grundumsatz sind: $grundumsatz kcal</p>"
67 #Berechnungen - Sport
68 if grep -q "$SPORT" "$Sportarten_db"; then
69   calo_minutes=$(grep "$SPORT" "$Sportarten_db" | cut -d';' -f2)
70   sport_calo=$(echo "scale=2;$calo_minutes * $SPORT_MINUTES" | bc)
71 else
72   echo "<p>ERROR</p>"
73 fi
74 echo "<p>Du hast mit $SPORT: $sport_calo kcal verbrannt</p>"
75 ALL_MEALS=$(echo "($MEALS * $MEALS_CALORIES)" | bc)
76 echo "<p>Du hast $ALL_MEALS kcal heute eingenommen</p>"
77 #Berechnung - Tageswert mit Bezug auf Sport
78 burned_calo=$(echo "$grundumsatz + $sport_calo" | bc)
79 daily_calo=$(echo "((($grundumsatz - ($MEALS * $MEALS_CALORIES)) - $sport_calo)" | bc)
80 echo "<p>Du hattest heute eine Kalorien verbrennung von: $burned_calo kcal</p>"
81
82 # Goal ausgabe
83 if [[ "$GOAL" == "zunehmen" ]]; then
84   if [ "$daily_calo" -gt "0" ]; then
85     #Ist gut
86     echo "<p>Ziel erreicht</p>"
87     ausgabeGoal=win
88   else
89     echo "<p>Ziel nicht erreicht</p>"
90     echo "<p>Du müsstest die Differnz von $daily_calo kcal zu dir nehmen, damit sich kein Defizit bildet</p>"
91     ausgabeGoal=lose
92     # Nicht ziel
93   fi
94 elif [[ "$GOAL" == "abnehmen" ]]; then
95   if [ "$daily_calo" -ge "0" ]; then
96     # Nicht gut
97     echo "<p>Ziel nicht erreicht</p>"
98     echo "<p>Du müsstest $daily_calo kcal verzichten, damit du keinen Überfluss an Kalorien zu haben</p>"
99     ausgabeGoal=lose
100   else
101     # gut
102     echo "<p>Ziel erreicht</p>"
103     ausgabeGoal=win
104   fi
105 else #Gewicht halten
106   if [ "$daily_calo" -ge "$((0 - 200))" ] && [ "$daily_calo" -le "$((0 + 200))" ]; then
107     # gut
108     echo "<p>Ziel erreicht</p>"
109     ausgabeGoal=win
110   else
111     echo "<p>Ziel nicht erreicht</p>"
112     echo "<p>Um dein Gewicht zu halten muss du die Differnz von $daily_calo ausbauen.</p>"
113     ausgabeGoal=lose
114     # schlecht
115   fi
116 fi

```

(Oliver Wendt)

Unser Skript "Rechner.sh" ist hauptsächlich für die Durchführung von Berechnungen verantwortlich. Es empfängt die Werte aus unserem Formular (siehe Bild...) und zeigt die eingegebenen Daten an. Anschließend wird der Grundumsatz anhand der Harris-Benedict-Formel für Männer und Frauen berechnet. Diese Formel wurde gewählt, da sie eine relativ hohe Genauigkeit aufweist, mit Abweichungen von lediglich $\pm 10\%$. Laut einer Studie „Abweichung von $\pm 10\%$. Bei den Probanden mit Übergewicht schätzten die Formeln zu 70, 52, 64 % den richtigen Wert innerhalb des 10 % Bereichs ab.“⁶ Aufgrund ihrer Zuverlässigkeit haben wir uns für die Verwendung dieser Formel entschieden.

⁶ Zoller, Tim. "Überprüfung viel verwendeter Formeln zur Abschätzung des Energie Grundumsatzes, der maximalen Herzfrequenz und der maximalen Sauerstoffaufnahme bei Schweizer Probanden mit unterschiedlichen anthropometrischen Voraussetzungen." (2018). (S.12)

Daraufhin wird aus unserer Kalorien-Sportdatenbank der entsprechende Kalorienverbrauch pro Minute entnommen. Basierend darauf wird der tägliche Kalorienverbrauch ermittelt sowie die Gesamtkalorien, die während des Tages verbrannt wurden. In unserem Formular wird auch das angestrebte Ziel angegeben, sei es Abnehmen, Zunehmen oder das zu Gewicht halten. Dies wird anhand der täglichen Kalorien berechnet, wobei ein Zusammenspiel aus Grundumsatz, Mahlzeiten und verbrannten Kalorien beim Sport berücksichtigt wird. Beim Zunehmen müssen die Kalorien positiv sein, beim Abnehmen negativ. Für das Gewicht Halten haben wir einen Puffer von 200 Kalorien eingeführt, um sicherzustellen, dass dieses Ziel erreichbar ist und ein Gefühl des Erfolgs vermittelt wird.

Die ermittelten Daten werden anschließend nur in den nutzerspezifischen Verzeichnissen gespeichert und verarbeitet, damit eine wöchentliche Bilanz erstellt werden kann.

Siehe Aktivitätsdiagramm von Rechner.sh (Seite 8), graphische Darstellung der Umsetzung.

Login.sh

```
1 #!/usr/bin/env bash
2 echo "Content-Type: text/html"
3 echo
4 echo
5 file="/usr/lib/cgi-bin/swe-prototyp/login/login.csv"
6 user_log="/usr/lib/cgi-bin/swe-prototyp/user-info"
7 username=$(echo "$QUERY_STRING" | cut -d'&' -f1 | cut -d'=' -f2)
8 password=$(echo "$QUERY_STRING" | sed 's/.*password=//g')
9 #Überprüft ob der username in der file gespeichert ist
10 if grep -q "$username" "$file"; then
11     check=$(grep "$username:" "$file" | sed 's/.*:.*//g')
12     #Überprüft ob das übergebene password mit dem in der .csv übereinstimmt
13     if [[ "$password" == "$check" ]]; then
14         /usr/lib/cgi-bin/swe-prototyp/formular.sh $username $password
15     else
16         echo "<html><head><meta http-equiv='refresh' content='0;URL=https://informatik.hs-bremerhaven.de/docker-step2023-team-09-web/swe/falsepw.html'></head></html>"
17     fi
18 else
19     #legt einen neuen user an
20     echo "$username:$password" >> "$file"
21     mkdir "$user_log/$username"
22     /usr/lib/cgi-bin/swe-prototyp/formular.sh $username $password
23 fi
```

(Emircan Aslan)

In unserer login.sh wird die Anmeldung/Registrierung für unser System durchgeführt.

Beim Anmeldevorgang wird als erstes überprüft, ob der angegebene Benutzername in unserer login.csv hinterlegt ist. Ist dies der Fall, wird als nächstes überprüft ob das zugehörige Passwort, mit dem in der Datenbank hinterlegten übereinstimmt und der Nutzer wird angemeldet. Andernfalls wird dem Nutzer mitgeteilt, dass das angegebene Passwort falsch ist.

Wenn es sich bei dem Nutzer um einen neuen Benutzer handelt und sein Benutzername nicht in unserer Datenbank auftaucht, so wird dieser mit seinem Benutzernamen und dem Passwort in der Datenbank hinterlegt und angemeldet. Zudem wird in unserer Datenbank ein Verzeichnis für den Nutzer erstellt, wo im späteren Verlauf seine Eingaben gespeichert werden.

Zur Vermeidung von Schwachstellen überprüfen wir ähnlich wie bei der Anmeldung bei jeder Weiterleitung zur nächsten Seite, noch einmal Benutzernamen und Passwort.

„Identifizierung von Sicherheitslücken kann in verschiedenen Stadien des Software-Entwicklungsprozesses und -Lebenszyklus durchgeführt werden.“⁷ Sollte sich im Verlauf der Nutzung diese ändern und nicht mehr übereinstimmen, so wird der Nutzer zum Anmelde Fenster geleitet.

Siehe Aktivitätsdiagramm von Login.sh (Seite 7), graphische Darstellung der Umsetzung.

⁷ SSAKAL, Peter; POHL, Hartmut. Entwicklungshelfer und Stresstester: Tool-gestützte Identifizierung von Sicherheitslücken in verschiedenen Stadien des Softwarelebenszyklus. KES: die Zeitschrift für Informationssicherheit, 2010, 26. Jg., Nr. 2, S. 63-66

Fazit

Die Umsetzung des Projekts stieß auf verschiedene Herausforderungen, aber auch auf erfolgreiche Meilensteine. Leider stellte sich die Implementierung der Datensicherung und Passwortverschlüsselung als nicht machbar heraus, was potenzielle Sicherheitsrisiken für das System aufwirft. Des Weiteren traten Probleme bei der Datenverarbeitung auf, insbesondere in Bezug auf Berechtigungen, was zu erheblichen Verzögerungen führte. Trotz dieser Hürden wurde die Weiterleitung an das nächste Skript erfolgreich implementiert, was einen wichtigen Fortschritt darstellt.

Die Idee einer allgemeinen Datenbank für Lebensmittel zur besseren Kalorienverfolgung konnte jedoch nicht realisiert werden, was möglicherweise die Funktionalität des Endprodukts beeinträchtigt. Zusätzlich stellten sich die grafische Umsetzung in UML-Diagrammen und die Suche nach relevanten Zitaten, als problematisch heraus, was die Kommunikation und wissenschaftliche Basis des Projekts beeinträchtigt hatten.

Trotz dieser Hindernisse wurde konstruktiv am Prototyp gearbeitet, und das Projekt wurde erfolgreich bis zum aktuellen Produktstand umgesetzt. Besonders positiv war die reibungslose Zusammenarbeit trotz räumlicher Distanz. Insgesamt zeigt sich, dass trotz einiger Schwierigkeiten bedeutende Fortschritte erzielt wurden und das Projekt auf einem soliden Fundament steht.

Literaturverzeichnis

- [1] RKI, 2017
- [2] Christian Schöttler, Self-Tracking zwischen Emanzipation und digitaler Überwachung, 23.02.2024
- [3] IBM, Anwendungsfalldiagramme
- [4] Prof. Dr. Ulrike Erb, SWE 1 – Modellierung UML-Klassendiagramme Notationen und Beispiele S.11
- [5] Dehlinger J, Dixon J (2011) Mobile application software engineering: Challenges and research directions. Workshop on Mobile Software Engineering
- [6] Zoller, Tim. "Überprüfung viel verwendeter Formeln zur Abschätzung des Energie Grundumsatzes, der maximalen Herzfrequenz und der maximalen Sauerstoffaufnahme bei Schweizer Probanden mit unterschiedlichen anthropometrischen Voraussetzungen." (2018). (S.12)
- [7] SSAKAL, Peter; POHL, Hartmut. Entwicklungshelfer und Stresstester: Tool-gestützte Identifizierung von Sicherheitslücken in verschiedenen Stadien des Softwarelebenszyklus. KES: die Zeitschrift für Informations-Sicherheit, 2010, 26. Jg., Nr. 2, S. 63-66