



SuperBlog in GO

Projekt

für die Prüfung zum

Bachelor of Science

des Studienganges Angewandte Informatikan der
Dualen Hochschule Baden-Württemberg Mosbach

von

3818468 (INF15B), 6985153 (INF15A), 9875672 (INF15B)

Abgabedatum	6. Januar 2018
Bearbeitungszeitraum	5. Semester
Gruppen ID	42
Gutachter der Dualen Hochschule	Prof. Dr. Helmut Neemann

Inhaltsverzeichnis

Abkürzungsverzeichnis	iii
Abbildungsverzeichnis	iv
1 Architektur	1
1.1 GO	1
1.2 Hypertext Markup Language	1
1.3 Cascading Style Sheet	3
2 Anwenderdokumentation	4
3 Betrieb	6
4 Aufgabenverteilung der Teammitglieder	9
4.1 3818468	9
4.2 6985153	9
4.3 9875672	9

Abkürzungsverzeichnis

CSS Cascading Style Sheet

HTML Hypertext Markup Language

JSON JavaScript Object Notation

UUID Universally Unique Identifier acronym plural CSS [CSS] Cascading Style Sheet

Abbildungsverzeichnis

1.1	Übersicht der Architektur	2
3.1	Startseite des SuperBlog	6
3.2	Anzeige eines einzelnen Beitrags	7
3.3	Bestätigung zum Löschen eines Artikels	7
3.4	Login-Seite	8
3.5	Änderung des Passworts	8

1 Architektur

Die Architektur des Projektes kann in die drei Teile „[HTML](#)“, „[CSS](#)“ und „GO-Logik“ geteilt werden. Die genannten Teile sind in Abbildung 1.1 grafisch dargestellt. Die Abbildung verdeutlicht den Zusammenhang der einzelnen Teile, die im Folgenden kurz erläutert werden.

1.1 GO

Die Logik des Projektes ist in der Klasse „main.go“ implementiert. Abbildung 1.1 zeigt einen Überblick über alle verwendeten Variablen, Typen und Funktionen. Die main-Funktion der Klasse initialisiert beim Start des Projektes zunächst die Variablen mit den beim Start übergebenen oder per Default eingestellten Flags. Anschließend werden die notwendigen Hypertext Markup Language ([HTML](#))-Dateien für den Blog geparsed, sodass die Daten in die [HTML](#)-Templates eingefügt werden können. Danach wird der Webserver gestartet, der dann die geparsten Templates und Cascading Style Sheet ([CSS](#))-Dateien ausliefert.

1.2 Hypertext Markup Language

Der [HTML](#)-Teil des „SuperBlogs“ setzt sich aus den Dateien „index.html“, „loeschen.html“, „login.html“, „neu.html“, „profil.html“ und „template.html“ zusammen. Um ein einheitliches Design gewährleisten zu können, greifen diese Dateien auf ein [CSS](#) zurück. Dabei parsed jede Seite zunächst die Datei „theme.css“ und anschließend „style.css“. Die Funktion beider [CSS](#)-Dateien wird im folgenden Kapitel 1.3 genauer erläutert.

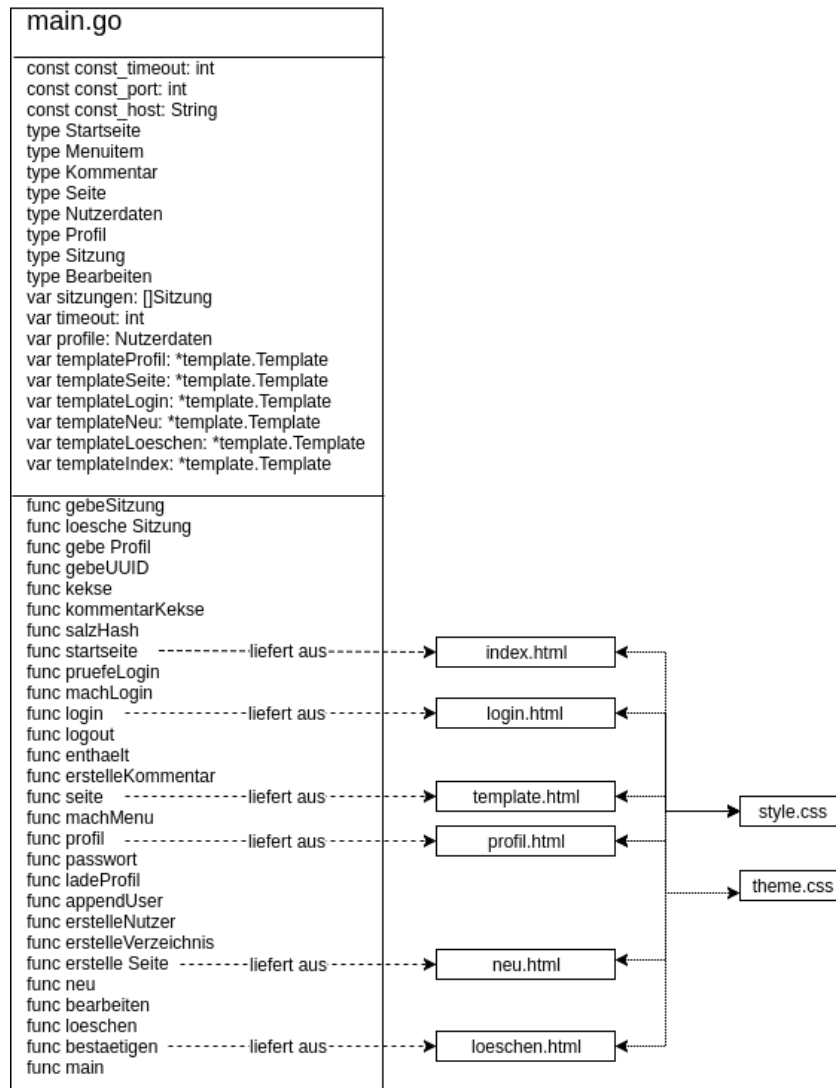


Abbildung 1.1: Übersicht der Architektur

1.3 Cascading Style Sheet

Die Darstellung und Farbgebung des „SuperBlogs“ kann mithilfe eines Cascading Style Sheets angepasst werden. Die Seite ist für ein einfaches Theming konzipiert, welches in der Datei „theme.css“ im Ordner „css“ verändert werden kann. Dort werden die in der „style.css“-Datei verwendeten Parameter, wie beispielsweise die Farben der Weboberfläche, definiert.

Für eine ansprechende Darstellung wurde ein StyleSheet erstellt, welches den Ansätzen des von Google geprägten Material Design folgt. Da das Farbschema in der Datei „theme.css“ angepasst werden kann, ist die farbliche Abgrenzung zu Googles originalem Material Design gegeben.

2 Anwenderdokumentation

Für den Betrieb von SuperBlog wird das Programm mit „go run /src/de/vorlesung/projekt/42/main.go [Optionen]“ zur Ausführung gebracht. Dabei können verschiedene Flags verwendet werden. Die möglichen Flags werden mit der Option „-h“ auf der Kommandozeile angezeigt. Folgende Flags existieren:

-h

Zeigt eine Übersicht aller setzbaren Flags an.

-host string

Setzt die Host-URL des Webservers.

-nutzer

Legt neue Benutzer mit gewünschtem Nutzernamen und Passwort an.

-Port int

Ändert den Port für den Webserver (Default: 8000).

-timeout int

Ändert den Timeout von Sitzungen in Minuten (Default: 15).

Nachdem das Programm erstmals über die Konsole gestartet und die gewünschten Flags gesetzt wurden, kann der Blog über „https://Adresse:Port“ (Default: https://localhost:8000) erreicht werden. Wird nun ein neuer Blogeintrag erstellt, so wird automatisch ein Ordner „seite“ im aktuellen Verzeichnis angelegt. In diesem werden auch die zukünftig erstellten Blogeinträge gespeichert. Dabei wird jeder Beitrag in einer eigenen JavaScript Object Notation (**JSON**)-Datei gespeichert, die als Namen eine eigene Universally Unique Identifier (**UUID**) erhält (z.B: „192137141451325523553.json“). Der Inhalt dieser Datei gliedert sich wie folgt:

-Menu

dient lediglich zur Anzeige des Menüs, wird nicht mitgespeichert

-Dateiname

UUID der Datei

-Titel

Titel des Blogeintrags

-Inhalt

Inhalt des Blogeintrags

-Datum

Datum der Beitragserstellung

-Autor

Autor des Eintrags

-Kommentare

Kommentare zum Blogeintrag (Autor, Datum und Inhalt jedes einzelnen Kommentars)

-Bearbeitet

Datum, wann der Blogeintrag bearbeitet wurde

Auch die Profile der einzelnen Nutzer des Blogs werden in einer [JSON](#)-Datei gespeichert. Diese befindet sich jedoch nicht im Ordner „seite“ sondern im aktuellen Verzeichnis. In dieser Datei werden zu jedem erstellten Profil die folgenden Informationen gespeichert:

-Menu

dient lediglich zur Anzeige des Menüs, wird nicht mitgespeichert

-Name

Name des Nutzers

-Passwort

gehashtes Passwort des Nutzers

-Meldung

dient lediglich zur Anzeige einer Meldung beim Login-Screen, wird nicht mitgespeichert

3 Betrieb

Nachdem der Blog, wie im vorherigen Kapitel [Anwenderdokumentation](#) beschrieben, gestartet und die gewünschten Flags gesetzt wurden, kann dieser verwendet werden. Der Nutzer gelangt zunächst auf die Startseite (s. Abbildung 3.1), die in „index.html“



Abbildung 3.1: Startseite des SuperBlog

implementiert ist. Auf der Startseite werden jeweils die ersten 1000 Zeichen aller im Blog vorhandenen Posts, die durch registrierte User erstellt wurden, angezeigt. Ist bisher kein Eintrag erstellt worden oder wurde soeben der letzte vorhandene Eintrag gelöscht, so erscheint eine Meldung, dass aktuell kein Blogeintrag vorhanden ist. Sind jedoch Einträge verfügbar, werden diese, absteigend nach Erstelldatum sortiert, angezeigt. Ein gewünschter Beitrag kann gelesen werden, indem die Überschrift des Eintrags ausgewählt wird.

Der Nutzer gelangt dadurch automatisch auf diesen Blogeintrag (durch „template.html“

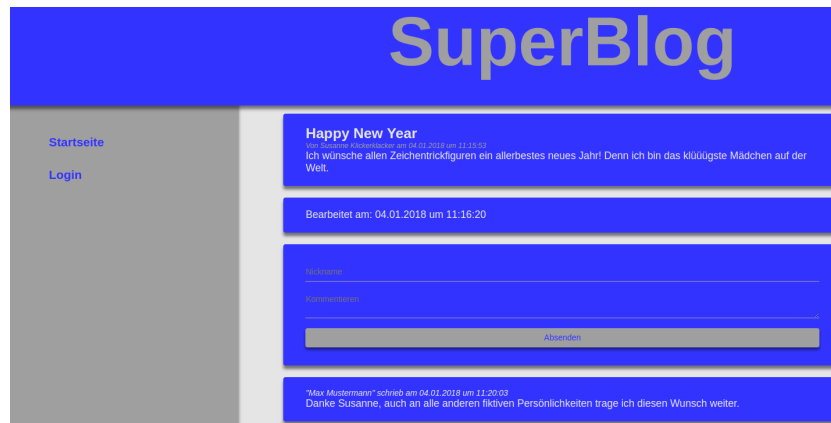


Abbildung 3.2: Anzeige eines einzelnen Beitrags

realisiert) und kann nun den vollen Beitrag lesen und kommentieren. Wie in Abbildung 3.2 dargestellt, wird zunächst der Beitrag selbst angezeigt. Wurde dieser von dem User, der ihn erstellt hatte, bearbeitet, so wird dies unter dem Beitrag angezeigt. Des Weiteren gibt es die Möglichkeit, Kommentare zum Beitrag zu schreiben und die von anderen Lesern zu lesen. Die Kommentare werden dabei chronologisch geordnet, sodass der aktuellste oben ist. Wurde bereits im Vorfeld ein Beitrag kommentiert, so wird der im Cookie gespeicherte Nutzernamen bereits automatisch beim Kommentieren eingetragen.

Will ein Nutzer seinen eigenen Beitrag ändern, so muss dieser ebenfalls zunächst auf den zu ändernden Beitrag navigieren. Von dort aus besteht die Möglichkeit über den Menüeintrag „Artikel bearbeiten“ auf die Bearbeitungsseite (realisiert durch „neu.html“ mit automatisch eingefügtem Text des Beitrags) zu gelangen.

Soll ein Beitrag komplett gelöscht werden, so kann, ebenfalls von der Beitragsseite aus, der Menüeintrag „Artikel löschen“ gewählt werden. Um einen Beitrag nicht unbeabsichtigt löschen zu können, ist eine Bestätigung des Löschvorgangs erforderlich, die in Abbildung 3.3 sichtbar ist.



Abbildung 3.3: Bestätigung zum Löschen eines Artikels

Für das Erstellen, Bearbeiten und Löschen von Artikeln ist ein Login notwendig. So können Beiträge nur von dem User bearbeitet und gelöscht werden, der diesen auch erstellt hat. Die Login-Seite („login.html“) kann über das Menü „Login“ erreicht werden. Der Login wird durch eine einfache Seite, auf der der Nutzernamen und das Passwort



Abbildung 3.4: Login-Seite

eingetragen werden müssen, dargestellt (s. Abbildung 3.4). Will ein User sein Passwort ändern, so muss sich dieser zunächst im Blog anmelden und kann es anschließend über Menüeintrag des eigenen Namens ändern (durch „profil.html“ realisiert). Um das Passwort erfolgreich ändern zu können muss das neue Passwort zweimal und das alte Passwort einmal eingegeben werden (s. Abbildung 3.5).

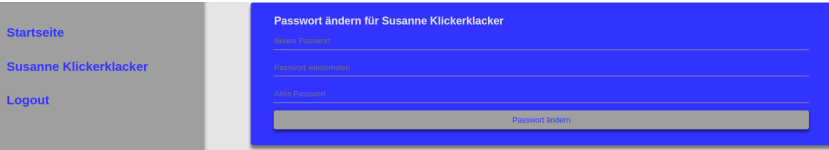


Abbildung 3.5: Änderung des Passworts

4 Aufgabenverteilung der Teammitglieder

4.1 3818468

Der Hauptaugenmerk lag in der Implementierung der GO-Logik und der initialen Erstellung der Web-Applikation. Darüber hinaus wurde Unterstützung bei den GO-Tests sowie bei der Vermittlung des Quellcodes geleistet.

4.2 6985153

Das Aufgabengebiet fokussierte sich auf das Projektmanagement inklusive der Terminierung und Verteilung von Aufgaben innerhalb der Gruppe. Des Weiteren gehörte zum Tätigkeitsbereich das nachvollziehbare Dokumentieren und Kommentieren des Quellcodes sowie die Erstellung dieses Dokuments.

4.3 9875672

Über die Unterstützung im GO-Quellcode hinaus lag der Fokus vor allem bei der Erstellung von sinnvollen Testfällen und der Qualitätsprüfung des Programmcodes. Außerdem gehörte die Erstellung eines ansprechenden Web-Designs zum Aufgabenfeld.