

Entwicklung einer Planer Web-Applikation

1. Zielbestimmung

1.1 Muss-Kriterien

1.2 Wunschkriterien

1.3 Abgrenzungskriterien

2. Produkteinsatz

2.1 Anwendungsbereiche

2.2 Zielgruppe

2.3. Betriebsbedingungen

3. Produktumgebung

3.1 Software

3.2 Hardware

4. Produktfunktionen

4.1 Benutzerfunktionen

4.1.1 Benutzerkennung

4.1.2 Persönliche Daten

4.1.3 Persönliche Konfiguration

5. Architektur der Anwendung

6. Nutzung der Anwendung und Funktionalität

1. Zielbestimmung

Die Planer-App namens „plannly“ ist eine erweiterte Notizen-App zur Planung verschiedener Aufgaben und eine grundsätzliche Plattform zum Festhalten von Notizen. Dies soll erleichtert werden durch Stichpunkte („Bulletpoints“ oder „Bullets“) als kleinste Bestandteile, die in Listen geordnet werden. Zusätzlich bieten tägliche Listen und ein Kalender die Möglichkeit der Planung von bestimmten Tagen.

1.1 Muss-Kriterien

- Der Benutzer kann sich in der Web-Applikation registrieren und seine Anmeldedaten werden online gespeichert.
- Der Benutzer kann Listen erstellen, die er beliebig benennen, umbenennen und löschen kann.
- Der Benutzer kann Bulletpoints erstellen, welche die folgenden Anforderungen erfüllen müssen:
 - Ein Bullet(point) ist grundsätzlich ein kleines Textstück mit Informationen, welches an einer festen Position in der Liste steht.
 - Bullets besitzen einen Typ (Dash, Bullet, To-Do, Star), wobei Dash und Bullet lediglich einen kosmetischen Zweck erfüllen, To-Dos die Erstellung einer Checkliste erlauben und Stars als globale „Lesezeichen“ zum Hervorheben von einzelnen Bullets dienen.
 - Der Typ jederzeit kann beliebig mit einem Auswahlfenster oder einem Tastenkürzel geändert werden.
 - Jeder Bulletpoint besitzt neben der vertikalen Position auch eine bestimmte Einrückung (Indent), wodurch man einfach hierarchische Strukturen in einer Liste erzeugen kann.
 - Mit einem kleinen Menü kann Bullets beliebig formatieren, in andere Listen verschieben und löschen.
- Jeder Benutzer besitzt eine Liste für jeden Tag im Jahr und kann dort Aufgaben und Notizen erstellen.
- Der Benutzer kann seine Listen mit anderen Benutzern teilen, auch wenn diese kein Benutzerkonto in der App besitzen.
- Alle Listen, Bullet-Points, Notizen und Login-Informationen werden in einer SQL-Datenbank online gespeichert und dort verarbeitet, wodurch sie unabhängig vom Gerät mit einer Internetverbindung von überall erreichbar sein sollen.

1.2 Wunschkriterien

- Der Benutzer kann seine Listen mit anderen Benutzern teilen und damit zusammen mit anderen Benutzern an einer Liste in Echtzeit arbeiten. (Funktion zur Kollaboration)
- Der Benutzer kann in einer Suchleiste nach Stichwörtern suchen, welche in einer oder mehreren seiner Listen enthalten sein können.
- Der Benutzer kann das Erscheinungsbild (Theme) der Applikation von hell auf dunkel stellen.
- Der Benutzer kann Tags erstellen und einen oder mehrere auf einmal einem Bulletpoint anhängen, um diesen außerhalb von nur Listen zu kategorisieren.

1.3 Abgrenzungskriterien

Die App kann momentan nur Text speichern, keine Dateien bzw. Bilder und Videos, da es ohne weitere Kosten nicht möglich ist, einen zuverlässigen Datenspeicher in der Cloud zu finden (wie z.B. Amazon S3). Die Benutzerkonten haben momentan wenig Funktionalität und könnten noch erweitert werden (z.B. Verwaltung, um Änderungen zu erlauben; Profilbilder; 2-Faktor-Authentifizierung).

2. Produkteinsatz

2.1 Anwendungsbereiche

Einzelpersonen verwenden diese Web-Applikation, um sich Notizen oder To-Dos für den Alltag zu erstellen, um wichtige Gedanken im Verlaufe des Tages nicht zu vergessen. Zusätzlich können Personen eine Liste erstellen, die mit einer Gruppe geteilt wird, und in der Zukunft kann hier auch die Möglichkeit einer Kollaboration an einer Liste bestehen.

2.2 Zielgruppe

Personengruppen jeden Alters, die leicht Dinge vergessen oder sich einfach eine Strukturierte Liste ihrer Gedanken erstellen möchten bzw. Gedanken planen möchten und z.B. mit einem bestimmten Datum im Kalender in Verbindung setzen möchten.

2.3. Betriebsbedingungen

Betriebsdauer: täglich, 24 Stunden am Server; je nach Nutzung im Browser

Der Server muss selten gewartet werden, kann jedoch zur Zeit der Wartungsarbeiten für wenige Minuten unerreichbar sein

Falls nötig, ist der Administrator zur Entfernung von expliziten Inhalten verantwortlich

3. Produktumgebung

Das Produkt ist weitgehend unabhängig vom Betriebssystem, sofern folgende Produktumgebung vorhanden ist.

3.1 Software

Benutzer

- Client = ein aktueller Web-Browser

Server

- Debian 10 (Linux)
- Generierte JAR-Datei mit Spring (Boot) und Vaadin 24
- JDK 19
- Tomcat Web Server
- Nginx (reverse-proxy)
- Certbot (mit SSL-Zertifikat von LetsEncrypt)
- PostgreSQL Datenbank bei Supabase
- Testen auf eigenem Rechner möglich (lokal, nur Tomcat)

Entwicklungsumgebung

- IntelliJ, verschiedene Betriebssysteme (Windows, MacOS, Linux)
- JDK 19 oder 20
- Apache Maven zur Verwaltung von externen Paketen

3.2 Hardware

Client

- Internetfähiger Rechner bzw. Smartphone (Ausführen als progressive Web-App möglich)

Server

- Testen auf eigenem Rechner oder VPS-Server in der Cloud

4. Produktfunktionen

4.1 Benutzerfunktionen

4.1.1 Benutzererkennung

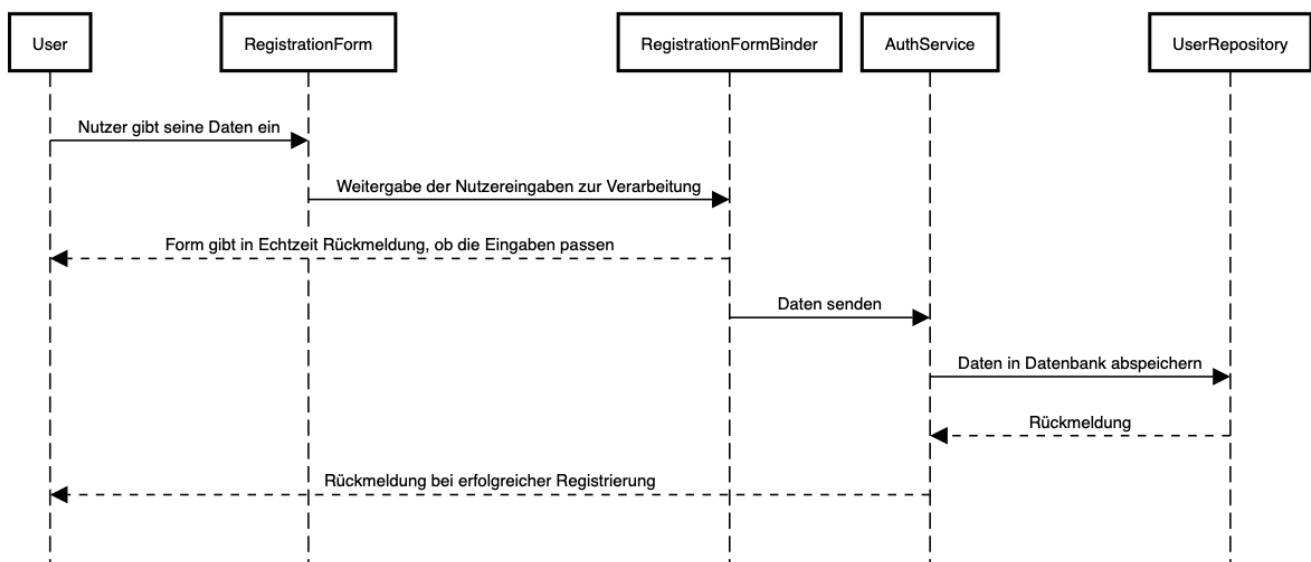
Ein im System registrierter Nutzer kann das System erst nutzen, wenn er angemeldet ist.

Registrieren: Ein beliebiger Internet-Benutzer kann sich über die Login-Seite des Systems schnell und bequem registrieren lassen. Zum Registrieren sind mindestens folgende Angaben erforderlich:

- gewünschter Benutzername
- gewünschtes Passwort
- E-Mail-Adresse (intern noch nicht verwendet, später jedoch für das Zurücksetzen von Passwörtern usw. nützlich)

Folgendes Sequenzdiagramm beschreibt den groben Prozess:

Sign-Up-Prozess



Die Registrierung ist erfolgreich, wenn der Benutzername und das Passwort innerhalb des Systems jeweils eindeutig sind, dies und weitere Abgleichung geschieht im RegistrationFormBinder. Mit dem erfolgreichen Abschließen des Registrierungs Vorgangs kann sich der Benutzer nun anmelden.

Der Anmeldeprozess wird vom Spring Security Framework übernommen, wobei ein Großteil der Logik für den Programmierer selbst nicht direkt sichtbar ist (Logik wie eine „Black Box“). Lediglich Konfiguration und Code zur Überbrückung zur Kompatibilität mit den Entitys sind sichtbar.

4.1.2 Persönliche Daten

Die persönlichen Daten des Benutzers sind in einer Datenbank gespeichert, die auf dem Server des Providers selbst verschlüsselt sind, jedoch für Administratoren frei zugänglich sind. Somit liegt zurzeit das volle Vertrauen im Betreiber, in der Zukunft könnten jedoch auch die Daten mit zusätzlicher Verschlüsselung für Administratoren unlesbar gemacht werden.

4.2 Bulletpointfunktionen

Bullets sind kleine Textstücke mit Informationen und die kleinsten „Bausteine“ in der App. Ein Nutzer kann die Eigenschaften aus den Musskriterien beliebig ändern.

4.3 Listenfunktionen

Eine Liste ist eine Ordnungsstruktur von Bullets, mit einer Position und einer Einrückung jedes Bullets. Es gibt zwei Arten von Listen:

4.2.1 Normale Listen

Normale Listen können nach Belieben von einem Nutzer in der Seitennavigation erstellt werden, und besitzen grundsätzlich einen Namen, einen festen Besitzer und intern ein Erstellungsdatum. Zusätzlich kann ein Nutzer eine Liste als öffentlich einstellen, sodass auch Nutzer ohne Benutzerkonto darauf zugreifen können. Eine Änderung des Namens oder die Löschung normaler Listen ist jederzeit möglich.

4.2.2 Tägliche Listen

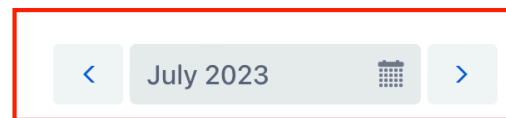
Tägliche Listen sind grundsätzlich eine Erweiterung der normalen Listen, mit einigen Besonderheiten. Sie werden für jeden Tag im Kalender automatisch generiert, wenn sie zum ersten Mal aufgerufen werden. Der Name ist festgelegt (exaktes Datum) und kann nicht geändert werden. Außerdem können tägliche Listen nicht gelöscht und auf öffentlich gesetzt werden.

Mit der festen Zugehörigkeit zu einem Datum lassen sich so für jeden Tag To-Do Listen oder ähnliches erstellen, die man vor allem für die Planung in die Zukunft benutzen kann, aber auch zum Abrufen von Dingen aus der Vergangenheit.

Das Datum der aktuell angezeigten täglichen Liste lässt sich entweder mit einfachen Knöpfen ändern oder mit dem Klick in der Kalender-Ansicht oder dem kleinen Kalender-Widget.

Pfeile und Kalender-Widget:

Friday, 07. July



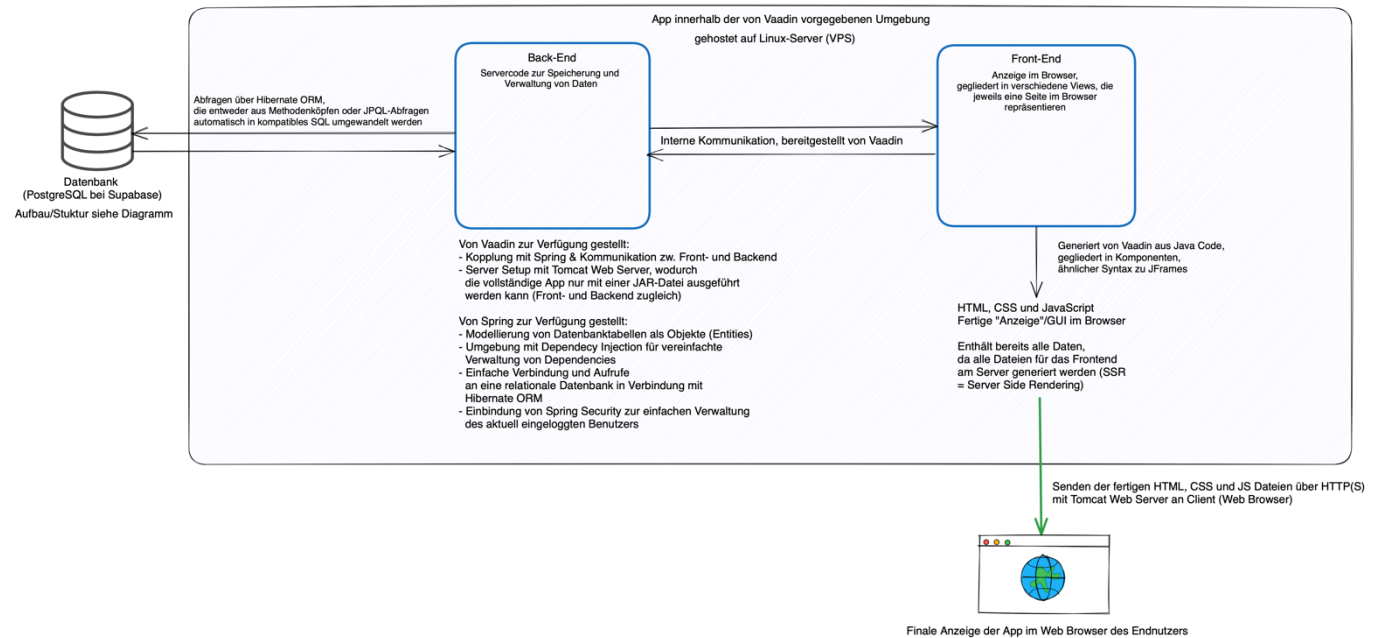
Vollständige Kalender-Ansicht: (hier können in Zukunft für jeden Tag Events angezeigt werden)

Calendar							< Today >	
Ma.	Di.	Wo.	Do.	Vr.	Sa.	So.		
W26 26	27	28	29	30	1	2		
W27 3	4	5	6	7	8	9		
W28 10	11	12	13	14	15	16		
W29 17	18	19	20	21	22	23		
W30 24	25	26	27	28	29	30		
W31 31	1	2	3	4	5	6		

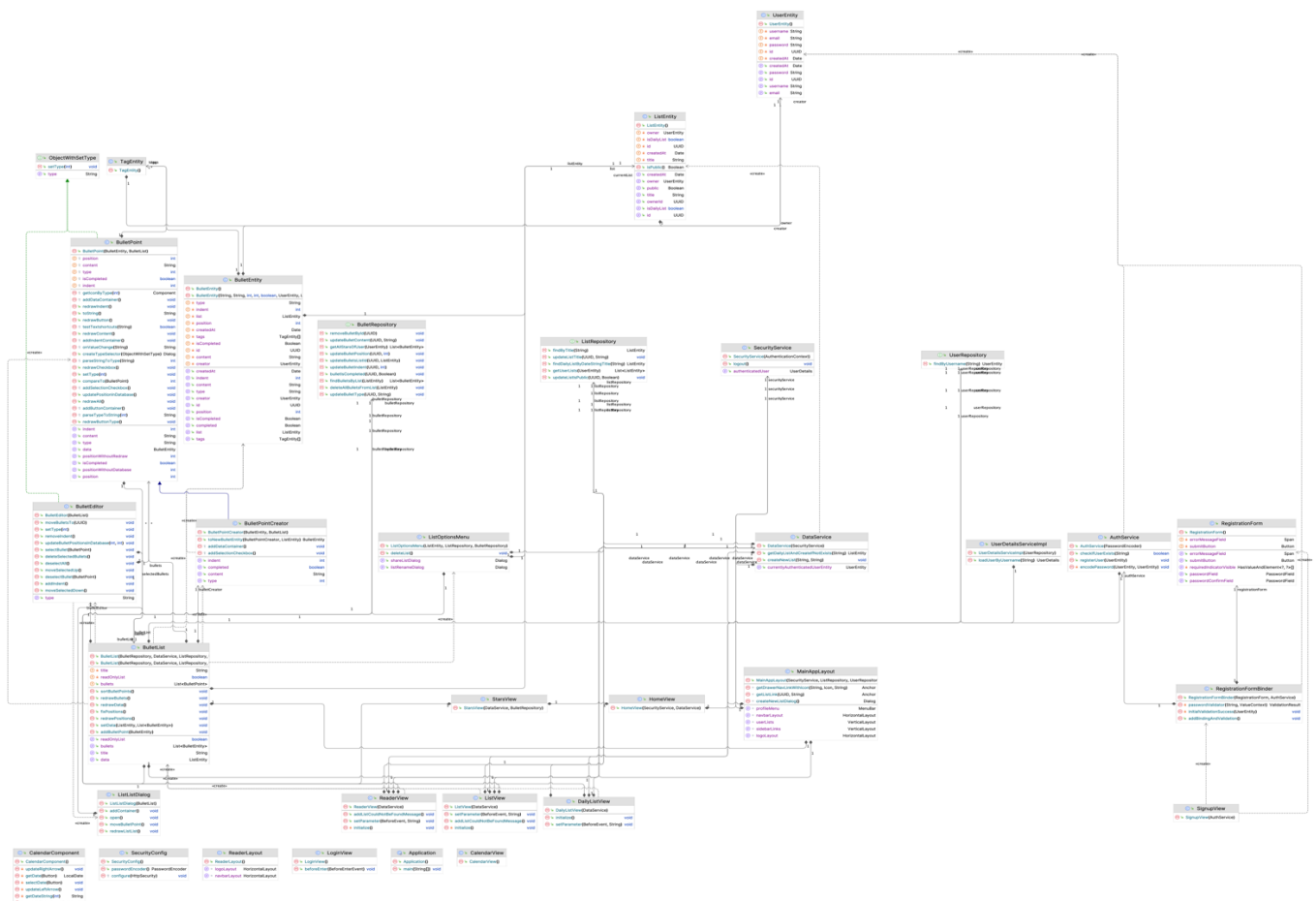
5. Architektur der Anwendung

Die grundsätzliche Architektur lässt sich einteilen in Front- und Backend in der von Vaadin vorgegebenen App-Umgebung, ein Datenbankserver und der Client (ein Web-Browser), der aus der Vaadin-Umgebung die fertigen Dateien zur Anzeige der App erhält.

Folgendes Diagramm zeigt dies detailliert und erläutert zusätzlich die Rolle der verschiedenen Bestandteile:

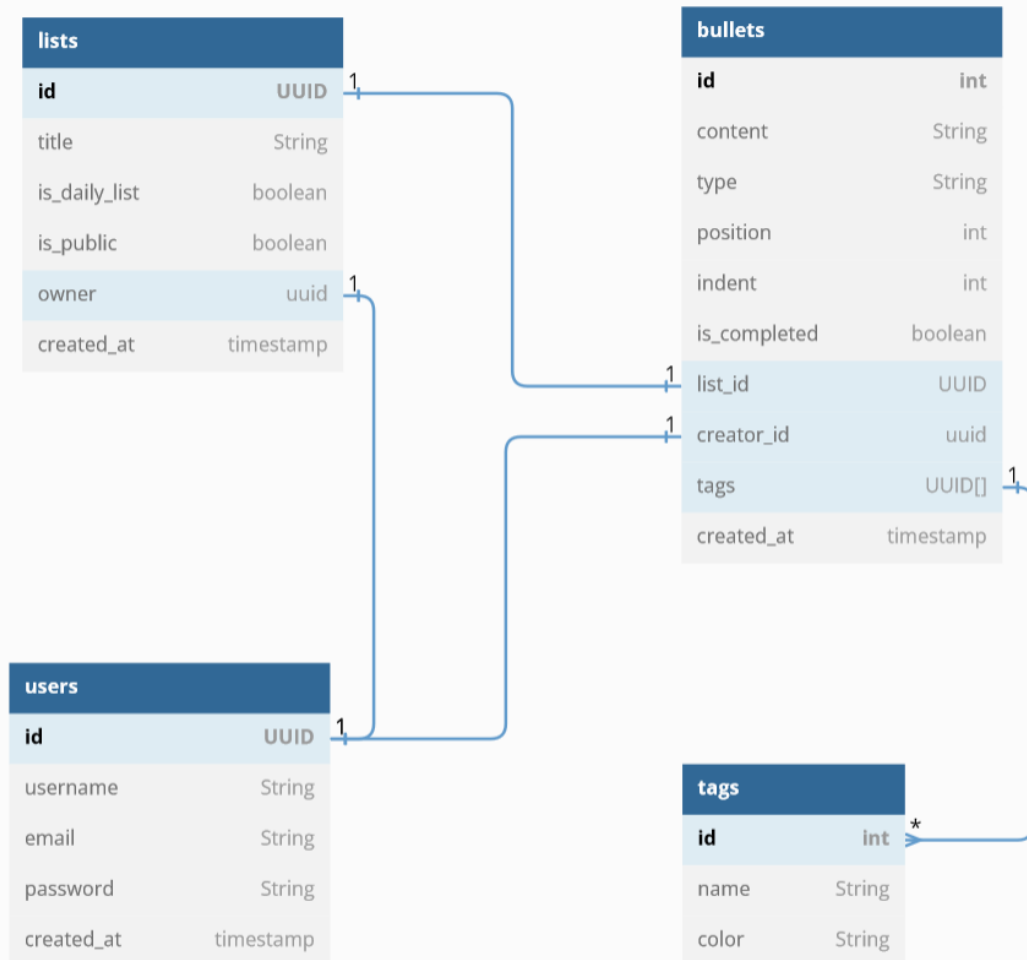


Da Front-End und Back-End intern gekoppelt sind, lassen sich diese in einem Klassendiagramm nicht direkt trennen. Trotzdem ist ein Klassendiagramm sinnvoll, da es alle verschiedenen Klassen und ihre Verbindungen anzeigt: (Anmerkung: Viele Verbindungen entstehen indirekt durch Dependency Injection)



Da das Klassendiagramm zu groß ist und ein hohes Zoom-Level nötig ist, lässt sich eine Version in voller Auflösung im Projektordner finden.

Die Datenbank ist relational und im Quellcode als Klassen/Objekte, die sogenannten Entitys modelliert, in der Theorie und am Datenbankserver sieht sie wie im folgenden Diagramm aus: (wobei die Tabelle für Tags momentan noch nicht in Verwendung ist, jedoch in der Zukunft für die Tags wichtig sein wird)



6. Nutzung der Anwendung und Funktionalität

Die finale Version der App ist mit einem Internet-fähigen Gerät öffentlich zugänglich, unter der Adresse <https://planner.trobo.tech>

Beim ersten Aufruf der Webseite wird man auf die Login-Seite weitergeleitet. Wenn man noch kein Benutzerkonto besitzt, lässt sich dieses auf der verlinkten „Signup“-Seite erstellen.

Login-Seite mit verlinkter Seite zur Account-Erstellung:

Welcome to plannly

Log in

Username •

Password •

Log in

[Forgot password](#)

[Don't have an account yet? Sign up](#)

Auf der „Signup“-Seite lässt sich einfach ein Benutzerkonto erstellen:

Create an account

Username •

maxmustermann

Email •

mustermann@gmx.de

Password •

••••••••

Confirm password •

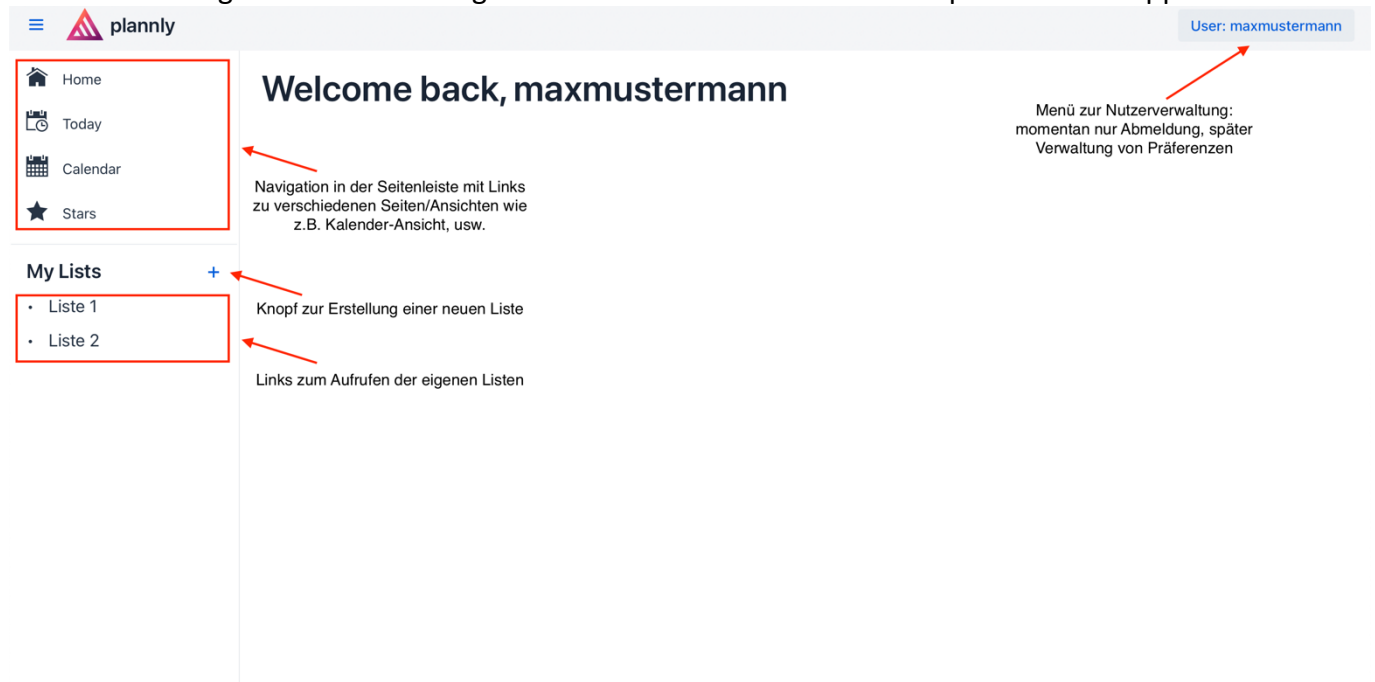
••••••••

Create Account

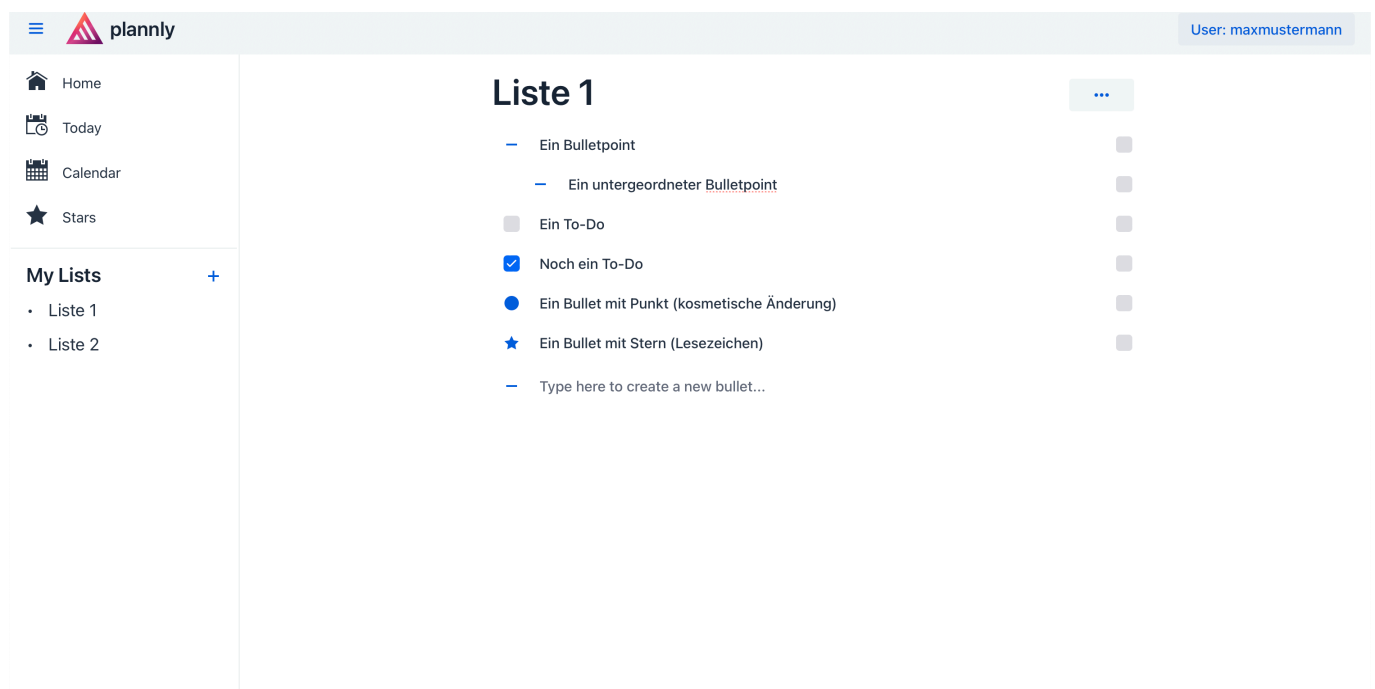
[Already have an account? Log in](#)

Nach der erfolgreichen Erstellung eines Benutzerkontos wird der Nutzer auf die Login-Seite weitergeleitet, auf der er sich anmelden kann, um Zugriff auf die App zu erhalten.

Nach einer erfolgreichen Anmeldung befindet sich der Nutzer in der Hauptansicht der App:



Das Herzstück der App sind die Listen, die folgendermaßen aussehen können:



Die Benutzeroberfläche ist vollständig auf Englisch, jedoch bietet das Vaadin-Framework die Möglichkeit an, eine App nachträglich auch in andere Sprachen zu übersetzen. Aus zeittechnischen Gründen wurde eine Übersetzung auf Deutsch ausgelassen, kann aber später hinzugefügt werden.

Alle weiteren Funktionen der App können unter dem oben angegebenen Link selbst ausprobieren und/oder werden in der Präsentation gezeigt.