Projektplanung Mobile App TaskTimer

Projektidee

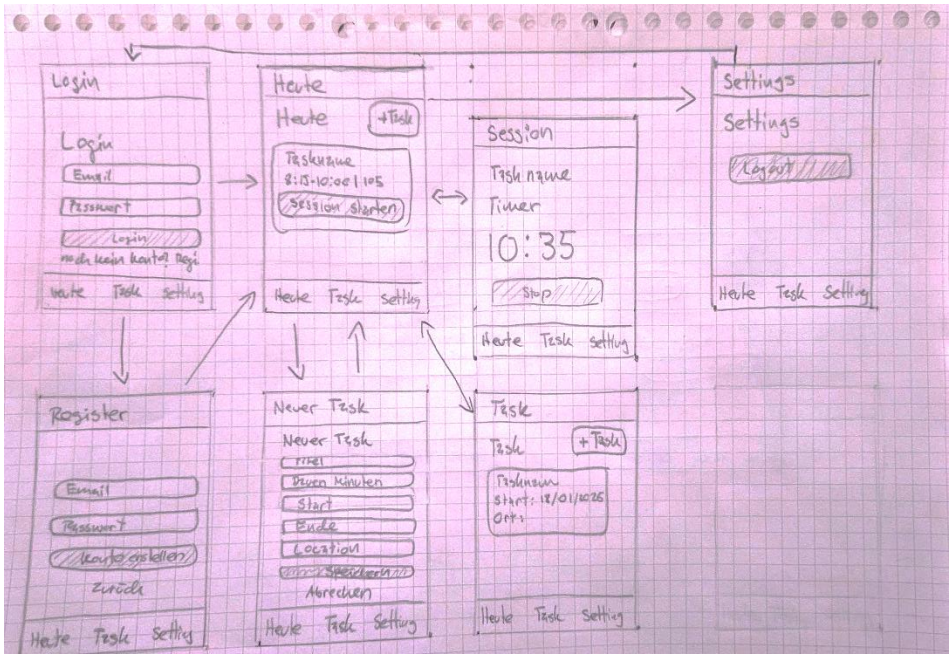
Die App TaskTimer hilft Nutzern dabei, ihre täglichen Aufgaben (Tasks) zu planen und zu bearbeiten.
Jede Aufgabe hat:

- einen Titel
- eine geplante Dauer
- eine Start- und Endzeit
- Datum
- optional einen Standort (GPS)

Beim Start einer Aufgabe wird einen Timer angezeigt, den man Stoppen und wieder fortsetzen kann.
Beim Öffnen der App sind die Task die für Heute geplant sind im separatem Tab ersichtlich.

Die App verwendet:

- Authentifizierung (Login/Registrierung)
- persistente Datenspeicherung (Firebase Firestore)
- Sensoren/Aktoren (GPS & Timer)



Ablauf der Screen-Folgen

1. App startet → Login
2. Login erfolgreich → Heute / Login → Registrieren → Heute
3. Heute→ Task auswählen oder neuen Task erstellen
4. Task starten → Session mit Timer
5. Session beenden → zurück zu Heute
6. Heute→ Logout → Login

Auflistung aller Funktionalitäten

Funktionalitäten

- **Auth**: Login, Register, Logout (Firebase Auth)
- **Storage**: Tasks speichern und laden (Firestore)
- **Sensor**: GPS Location für Tasks
- **Zeit/Sensor**: Systemzeit für "Tasks heute"
- **Aktor**: Timer (Session) zeigt Sekunden/Minuten live

Authentifizierung

- Benutzerregistrierung
- Benutzerlogin
- Logout
- Zugriff nur für eingeloggte Benutzer

Persistente Speicherung (Firestore)

- Speichern von Tasks
- Laden der Tasks pro Benutzer
- Jeder Benutzer sieht nur seine eigenen Daten

Sensoren / Aktoren

Sensor : GPS (Location)

- Ermittlung des aktuellen Standorts
- Speicherung des Standorts beim Task

Aktor : Timer

- Start/Stop Timer während einer Session
- Anzeige der verstrichenen Zeit in Sekunden

Zeit- & Taskverwaltung

- Geplante Start- und Endzeiten
- Geplante Dauer eines Tasks
- Filterung der Tasks nach heutigem Datum

Testplan (Anwendungsfälle als Testfälle)

Testumgebung

- Gerät: Android oder iOS Smartphone mit **Expo Go**
- Internetverbindung aktiv (Firebase)
- Firebase: Authentication (Email/Passwort) aktiv, Firestore aktiv, Rules gesetzt
- App Start: npx expo start --tunnel --clear

Testfälle

T1 – App startet korrekt & Redirect

Voraussetzung: App installiert/gestartet, kein User eingeloggt
Schritte:

1. App öffnen
2. Beobachten, welcher Screen erscheint

Erwartetes Ergebnis: App leitet auf Login Screen weiter

T2 – Registrierung

Voraussetzung: Kein Account vorhanden
Schritte:

1. Auf Registrieren gehen
2. Email + Passwort eingeben
3. Konto erstellen klicken

Erwartetes Ergebnis: Account wird erstellt, User ist eingeloggt, App zeigt Heute Screen

T3 – Login

Voraussetzung: Account existiert
Schritte:

1. Email + Passwort eingeben
2. Login klicken

Erwartetes Ergebnis: Login erfolgreich, App zeigt Heute Screen

T4 – Logout

Voraussetzung: User ist eingeloggt
Schritte:

1. Settings öffnen
2. Logout drücken

Erwartetes Ergebnis: User wird abgemeldet, App zeigt wieder Login Screen

T5 – Task erstellen ohne Location

Voraussetzung: User eingeloggt
Schritte:

1. + Task öffnen
2. Titel, Dauer, Start/Ende eingeben
3. Speichern

Erwartetes Ergebnis: Task wird gespeichert und erscheint in Tasks Liste (und in Heute)

T6 – Task erstellen mit Location (GPS Permission)

Voraussetzung: User eingeloggt, Standortberechtigung noch nicht vergeben
Schritte:

1. + Task öffnen
2. Location hinzufügen drücken
3. Permission erlauben
4. Speichern

Erwartetes Ergebnis: Location wird gespeichert, Task zeigt Standortdaten

T7 – Permission abgelehnt (GPS)

Voraussetzung: User lehnt Location-Permission ab
Schritte:

1. Location hinzufügen drücken
2. Permission ablehnen

Erwartetes Ergebnis: App zeigt Fehlermeldung (Permission abgelehnt), Task kann nicht ohne Location gespeichert werden.

T8 – Heute-Filter (Tasks nur für heute)

Voraussetzung: Es gibt Tasks mit Startdatum heute und an anderen Tagen
Schritte:

1. Heute-Tab öffnen

Erwartetes Ergebnis: Es werden nur Tasks mit Startdatum heute angezeigt

T10 – Session starten (Timer)

Voraussetzung: Heute-Tab zeigt mindestens einen Task
Schritte:

1. Session starten drücken

Erwartetes Ergebnis: Session Screen öffnet sich, Timer startet bei 00:00 und zählt hoch

T11 – Session stoppen

Voraussetzung: Timer läuft
Schritte:

1. Stop drücken

Erwartetes Ergebnis: App navigiert zurück auf Heute Screen, Timer stoppt (Screen verlassen)

T12 – Datensicherheit: User sieht nur eigene Tasks

Voraussetzung: Zwei Accounts (User A und User B)
Schritte:

1. User A erstellt Task
2. Logout, dann Login als User B

Erwartetes Ergebnis: User B sieht nicht die Tasks von User A

Lösungskonzept (Entwicklung, Framework, App-Typ, Aufbau)

Entwicklung & App-Typ

Die App wird mit Expo (React Native) und Expo Router entwickelt.
Programmiersprache: TypeScript (.tsx).
Es handelt sich um eine Hybrid-App / Cross-Platform App, weil ein gemeinsamer Codebase für Android und iOS genutzt wird. Die App läuft während der Entwicklung in Expo Go.

Wichtige Komponenten:

- **Expo Router** für Navigation (Tabs + weitere Screens)
- **React Components** für UI (Screens, Buttons, Inputs)
- **Context (AuthProvider)** zur Verwaltung des Login-Status
- **Firebase SDK** für Authentifizierung und Datenbankzugriff
- **Firestore Snapshot Listener** für Live-Updates der Task-Liste

Nutzung der Aktoren/Sensoren, Storage, Auth, Local Storage

*Authentifizierung (Firebase Authen**tica**tion)*

- Nutzer registrieren und loggen sich mit Email/Passwort ein
- Nach Login wird der User-Status im AuthContext gespeichert (onAuthStateChanged)
- Logout entfernt die Session und führt zurück zum Login Screen
- **Warum:** Dadurch hat jeder User seine eigenen Daten und ist eindeutig identifizierbar (userid/uid).

Persistente Speicherung (Firebase Firestore)

- Tasks werden in der Collection tasks gespeichert
- Jeder Task enthält ownerUid, damit Daten pro Nutzer getrennt sind
- Tasks werden über onSnapshot geladen (Live-Updates)

Datenmodell Task (Beispiel):

- title: string
- plannedMinutes: number
- startAt: Date/Timestamp
- endAt: Date/Timestamp
- location: {lat, lng} oder null
- ownerUid: string

Sicherheitsregeln:

- Nur eingeloggte User dürfen auf Firestore zugreifen
- Jeder User darf nur seine eigenen Tasks lesen/ändern/löschen (ownerUid == uid)

Sensor 1: Standort (GPS) – expo-location

- Beim Erstellen eines Tasks kann der User Location hinzufügen drücken
- Die App fragt Permission an
- Bei Zustimmung wird die aktuelle Position gespeichert
- Bei Ablehnung kann der Task ohne Standort gespeichert werden
  Nutzen: Tasks können einem Ort zugeordnet werden

Aktor 2: Timer / Session (UI-Aktor)

- Beim Start einer Session öffnet sich ein Timer-Screen
- Ein setInterval erhöht jede Sekunde die Zeit
- Anzeige als MM:SS
- Mit Stop endet die Session
  Nutzen: Nutzer sieht live, wie lange er an der Aufgabe arbeitet.

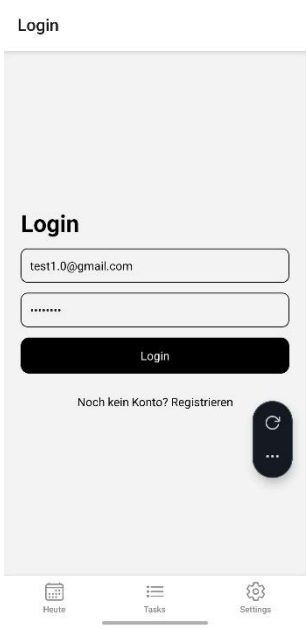
Mobile App – Mobile App programmieren

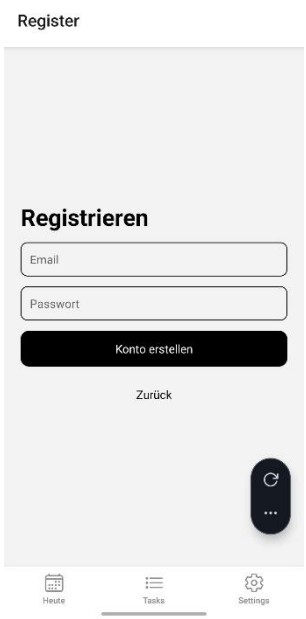
Ich habe das Projekt anhand meines Storyboards und der vorgegebenen Anforderungen erfolgreich umgesetzt. Die App bietet die Möglichkeit, sich zu registrieren und anschließend anzumelden. Nach dem Login gelangt man auf die Seite "Heute", auf der alle Tasks angezeigt werden, die für den aktuellen Tag geplant sind.

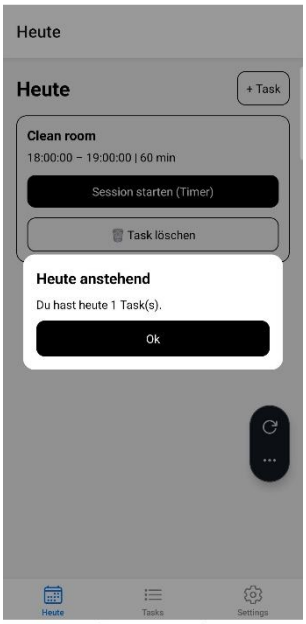
Zusätzlich gibt es die Seite "Tasks", auf der alle erstellten Tasks übersichtlich aufgelistet sind. Auf beiden Seiten "Heute" und "Tasks", können neue Tasks erstellt werden. Das dafür verwendete Formular wurde wie geplant umgesetzt und entspricht dem Aufbau aus dem Storyboard. Auch das Layout der App wurde nach dem Storyboard gestaltet.

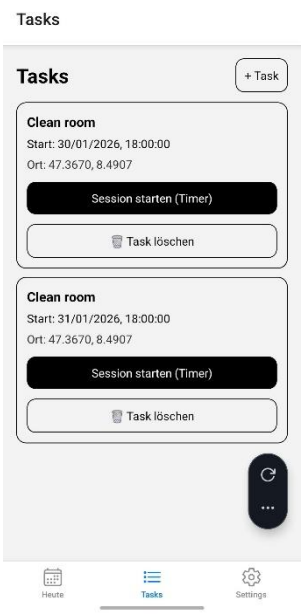
Während der Umsetzung habe ich festgestellt, dass eine Löschfunktion in der Planung gefehlt hat. Deshalb habe ich zusätzlich einen **Delete-Button** eingebaut, mit dem Tasks und

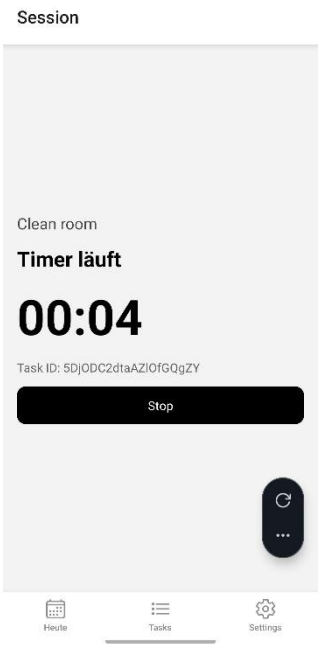
Sessions gelöscht werden können. Diese Erweiterung macht die App benutzerfreundlicher und erleichtert die Verwaltung der Tasks.

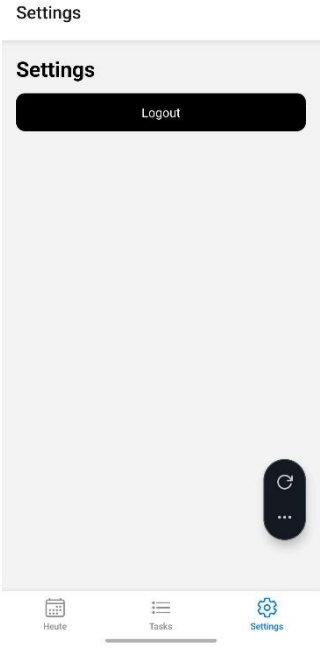












Mobile App – Mobile App publizieren

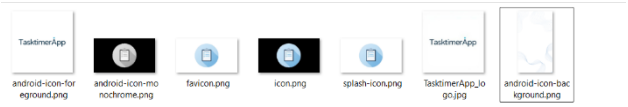
1. App final Test

Starte die App noch einmal um die Funktionalität vor dem publizieren zu überprüfen:

- Login funktioniert
- Tasks erstellen
- Tasks löschen
- Session starten

2. App Icons Anpassen

Überprüfen und falls nötig die App Icons anpassen und ersetzten. Im app.json den Pfad anpassen.



3. EAS prüfen / einrichten

Im Projektordner Terminal öffnen und folgendes im Terminal eingeben:

- npm i -g eas-cli

```
PS C:\Users\User\IdeaProjects\Modul335\workspace\TaskTimerApp> npm i -g eas-cli
/
```

- eas login -> loggin with your Expo Account

```
PS C:\Users\User\IdeaProjects\Modul335\workspace\TaskTimerApp> eas login
Log in to EAS with email or username (exit and run eas login --help to see other login options)
√ Email or username ... dmc.141592@gmail.com
√ Password ... ************
Logged in
```

- eas build:configure -> bestätigen das einEAS project erstellt werden soll und anschliessend auswählen auf welchen Platforms (Hier habe ich beides gewählt deshalb steht dort «ALL». Es wurde ein eas.json erstellt.

```
PS C:\Users\User\IdeaProjects\Modul335\workspace\TaskTimerApp> eas build:configure
EAS project not configured.
√ Would you like to automatically create an EAS project for @dmc.121/TaskTimerApp? ... yes
√ Created @dmc.121/TaskTimerApp: https://expo.dev/accounts/dmc.121/projects/TaskTimerApp on EAS
√ Linked local project to EAS project ad97cd13-ea7e-4784-a9cf-bd386ec70e6d
💡The following process will configure your iOS and/or Android project to be compatible with EAS
 revert them at any time.

√ Which platforms would you like to configure for EAS Build? » All
```

4. APK-Build konfigurieren

Öffne eas.json und prüfe, ob APK eingestellt ist. Falls nicht füge unter production folgenden code hinzu.

```
"production": {
  "autoIncrement": true,
  "android": {
    "buildType": "apk"
  }
}
```

5. APK erstellen (Publizieren)

Jetzt bauen wir die App, folgenden Befehl in den Terminal eingeben und einen Adroid Application ID auswählen, bei build Android Keystore (Y) auswählen:

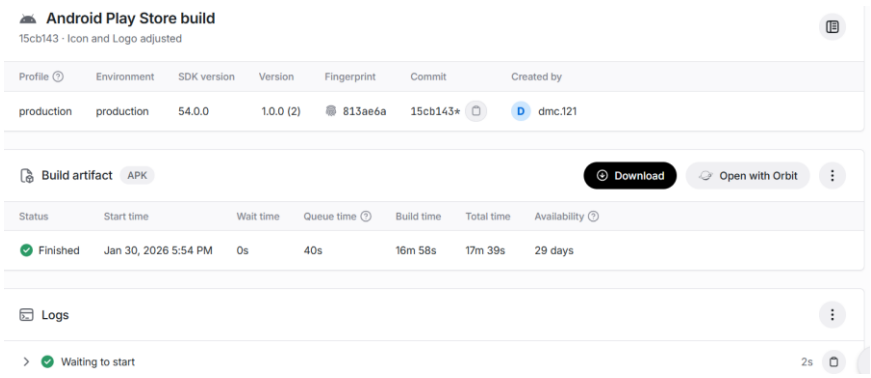
```
PS C:\Users\User\IdeaProjects\Modul335\workspace\TaskTimerApp> eas build --profile production --platform android
Resolved "production" environment for the build. Learn more: https://docs.expo.dev/eas/environment-variables/#setting-the-environment-for-your-builds
No environment variables with visibility "Plain text" and "Sensitive" found for the "production" environment on EAS.

Android application id Learn more: https://expo.fyi/android-package
√ What would you like your Android application id to be? ... com.dmc.tasktimer
No remote versions are configured for this project, versionCode will be initialized based on the value from the local project.
√ Incremented versionCode from 1 to 2.


√ Using remote Android credentials (Expo server)
√ Generate a new Android Keystore? ... yes
√ Created keystore

Compressing project files and uploading to EAS Build. Learn more: https://expo.fyi/eas-build-archive
√ Compressed project files 1s (1.3 MB)
√ Uploaded to EAS 1s
√ Computed project fingerprint

See logs: https://expo.dev/accounts/dmc.121/projects/TaskTimerApp/builds/5aff21b2-4fb5-415c-a2ea-daa8c90f911c

Waiting for build to complete. You can press Ctrl+C to exit.
/ Build in progress...
```

Was passiert:

- Expo baut die App in der Cloud
- Man bekommt einen Download-Link
- Ergebnis: eine fertige .apk Datei

6. APK herunterladen & testen

- APK herunterladen
- https://expo.dev/artifacts/eas/ceJ65qDHpCwbPhUahXga91.apk



- auf Android installieren
- prüfen:
    - App Icon sichtbar
    - Splash Screen sichtbar

- App startet korrekt

Mobile App gemäss Testplan überprüfen

Testfall	Beschreibung	Durchführung (Kurz)	Erwartetes Ergebnis	Tatsächliches Ergebnis	Status
T1	App startet korrekt & Redirect	App ohne Login gestartet	Weiterleitung zum Login-Screen	Login-Screen wurde angezeigt	Erfolgreich
T2	Registrierung	Registrierung mit E-Mail & Passwort	Account wird erstellt, Heute-Screen wird angezeigt	Account erstellt, Weiterleitung auf Heute	Erfolgreich
T3	Login	Login mit bestehendem Account	Login erfolgreich, Heute-Screen sichtbar	Login erfolgreich, Heute-Screen sichtbar	Erfolgreich
T4	Logout	Logout über Settings	Weiterleitung zum Login-Screen	Login-Screen wurde angezeigt	Erfolgreich
T5	Task erstellen ohne Location	Task mit Titel & Zeit erstellt	Task erscheint in Tasks & Heute	Task korrekt gespeichert & angezeigt	Erfolgreich
T6	Task erstellen mit Location	Location hinzugefügt & Permission erlaubt	Location wird gespeichert	Standortdaten korrekt gespeichert	Erfolgreich
T7	GPS Permission abgelehnt	Permission abgelehnt	Fehlermeldung, Task ohne	Task ohne Location nicht	Erfolgreich

			Location speicherbar	gespeichert werden	
T8	Heute-Filter	Heute-Tab geöffnet	Nur heutige Tasks sichtbar	Nur heutige Tasks sichtbar	Erfolgreich
T10	Session starten	Session starten gedrückt	Timer startet bei 00:00	Timer startete korrekt	Erfolgreich
T11	Session stoppen	Stop gedrückt	Rückkehr zum Heute-Screen	Navigation korrekt durchgeführt	Erfolgreich
T12	Datensicherheit	User A -> Logout -> User B	User B sieht keine Tasks von User A	Keine fremden Tasks sichtbar	Erfolgreich