Dokumentation ToDo-App

Hendrik Mallonn | IT22 | Abgabe: 13.03.2025

Anforderungen an das Projekt

- Es lassen sich ToDos erstellen und in einer Liste darstellen
- Einträge können einen Titel und eine optionale Notiz erhalten
- Einträge können hinzugefügt, als erledigt gekennzeichnet und bearbeitet werden
- Bereits erledigte Einträge lassen sich wieder als nicht erledigt kennzeichnen

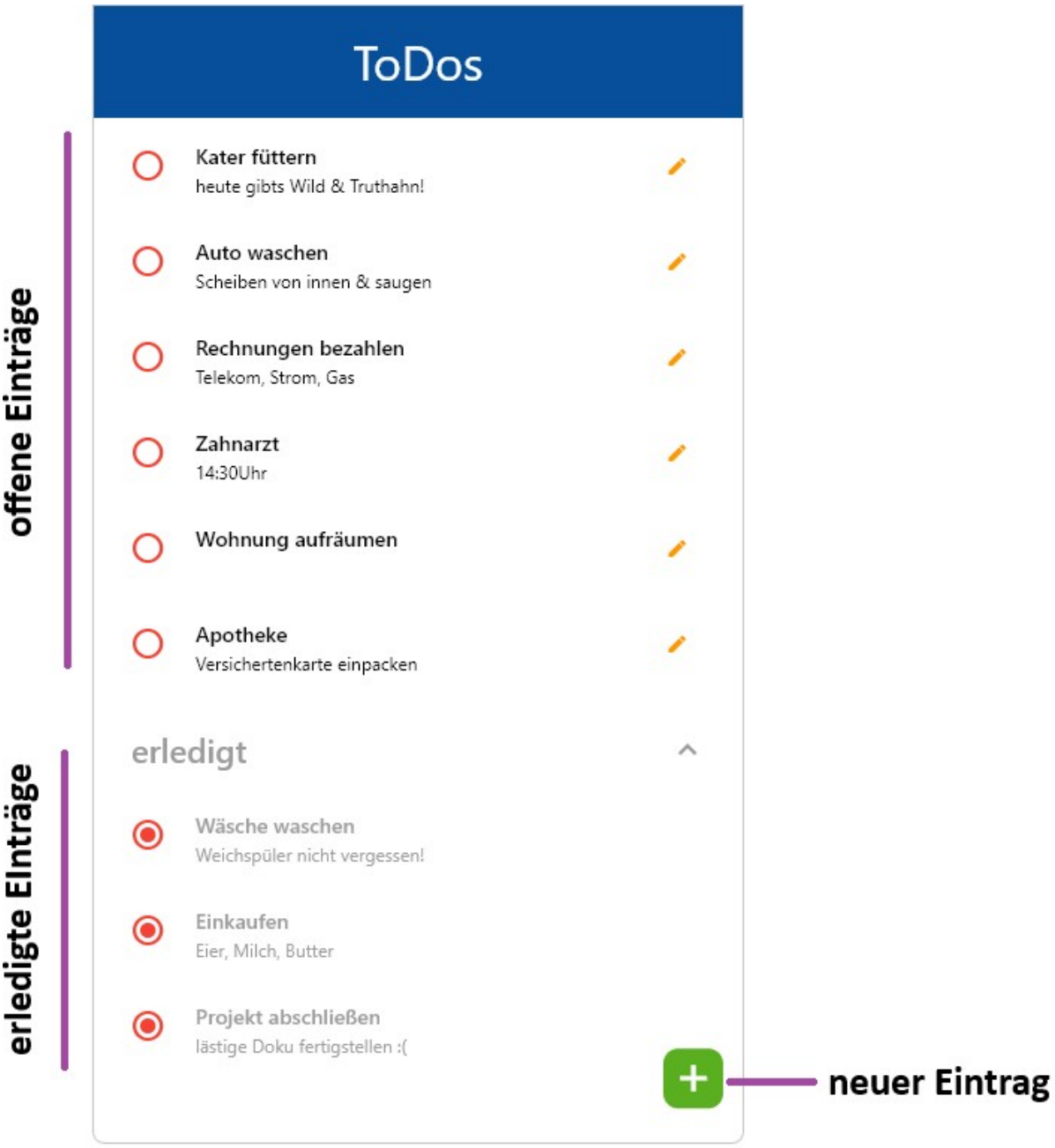
Beschreibung des Projekts

Im Rahmen eines Schulprojektes wird eine einfache ToDo-App in Flutter entwickelt. Flutter stellt ein auf Googles Dart-basierendes Framework dar, welches die Möglichkeit bietet, Apps zeitgleich als Web-, Windows-, macOS-, Android und iOS-Anwendung zu programmieren, ohne übermäßige Anpassungen für die jeweilige Version vornehmen zu müssen. Die ToDos werden klassisch als Liste dargestellt und zwischen ‚noch offen' und ‚bereits erledigt' separiert. Die grundlegenden Funktionen umfassen das Hinzufügen neuer ToDos inkl. optionaler Notiz, das Erledigen, das Bearbeiten sowie die Möglichkeit, ein bereits erledigtes ToDo wieder in den Status ‚offen' zu versetzen.

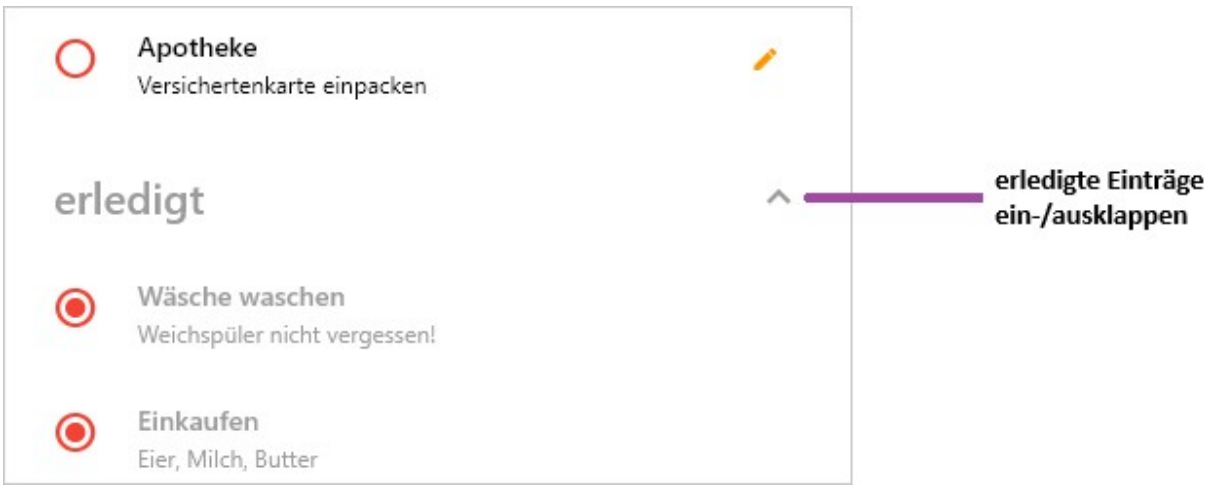
Die Daten werden auf dem ausführenden Gerät gespeichert, es bedarf keiner Internetverbindung oder sonstiger Verbindung zu einer externen Datenbank oder den Zugriff auf Dateien und/oder Ordnern des Betriebssystems.

Benutzerinterface und –Handbuch

Hauptbildschirm



Der Hauptbildschirm zeigt eine Listendarstellung aller Einträge der ToDo-Liste. Im oberen Bereich der App sind die offenen, noch zu erledigenden und im unteren Bereich des Bildschirms die bereits erledigten Einträge zu finden. Durch einen Klick oder Tap auf den grünen Button in der unteren, rechten Ecke des Bildschirms, wird die Eingabemaske für einen neuen Eintrag aufgerufen.



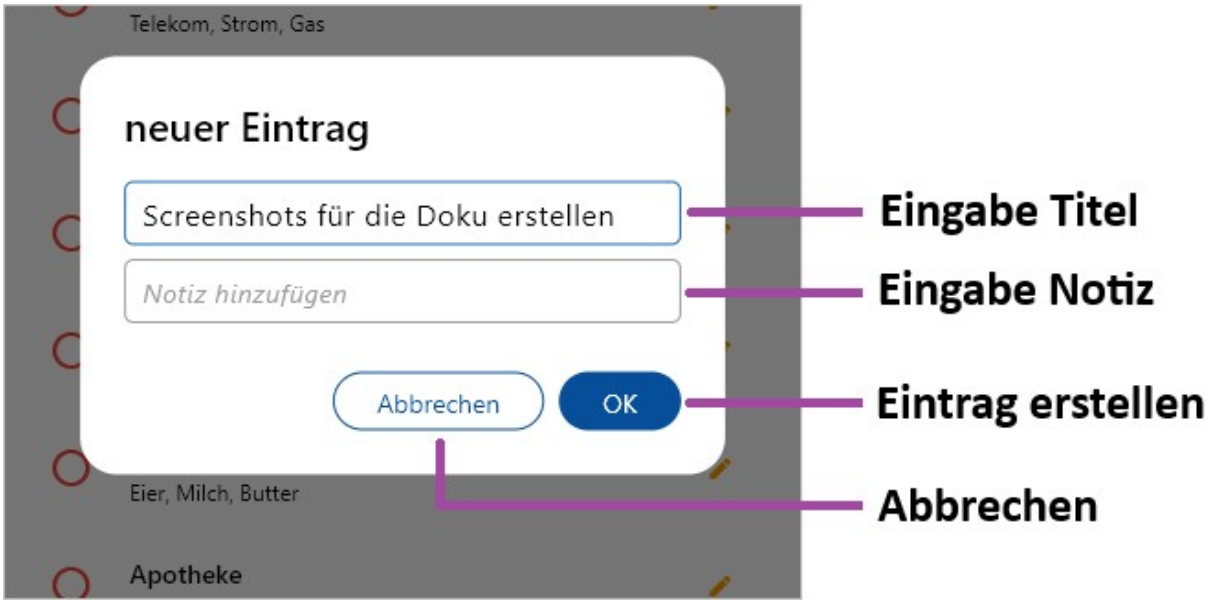
Mit einem Klick/Tap auf den kleinen Pfeil lassen sich die erledigten Einträge ein- oder ausklappen.

Listeneintrag im Detail

Die einzelnen Listenelemente zeigen den Titel und die Notiz – falls beim Erstellen hinzugefügt – an. Der Indikator auf der linken Seite zeigt an, dass das ToDo noch zu erledigen ist. Mit einem Klick oder Tap auf diesen, markiert der Benutzer das ToDo als „erledigt“. Dadurch wird er in den unteren Bereich verschoben.

Der Edit-Button auf der rechten Seite bietet die Möglichkeit, den Titel und/oder die Notiz noch einmal zu ändern und den Eintrag erneut zu speichern.

Eintrag hinzufügen/bearbeiten



Mit einem Klick/Tap auf den grünen „+“-Button in der unteren, rechten Ecke des Bildschirms, wird die Eingabemaske zum Erstellen eines neuen ToDo-Eintrages aufgerufen. Hier hat der Benutzer die Möglichkeit, einen Titel (Pflichtfeld) und eine dazugehörige Notiz (optional) einzugeben. Mit einem Klick/Tap auf den „OK“-Button wird der neue Eintrag der Liste hinzugefügt. Ein Klick/Tap auf den „Abbrechen“-Button bricht den Vorgang ab und schließt den Popup. In diesem Fall wird kein neuer Eintrag erstellt.

Sollte der Benutzer den „OK“-Button drücken, ohne einen gültigen Titel angegeben zu haben, erscheint am unteren Bildschirmrand eine Aufforderung, dies zu tun.

Projektablauf

Der Projektablauf wurde in mehrere Phasen unterteilt. Aufgrund des relativ geringen Umfangs des Projektes, ist es nicht möglich, jeden einzelnen Schritt mit einer genauen Zeitangabe zu versehen. Generell hat sich das Projekt wie folgt unterteilt:

- Themenfindung
- Planung
    - Wie genau soll das Projekt realisiert werden
    - Festlegen der Programmiersprache / des Frameworks
    - User Interface / User Experience
- Realisierung
    - Erstellen eines groben Frontends
    - Funktionen für die Benutzerschnittstelle
    - Implementieren von Variablen und Funktionen, die der Verarbeitung der Daten dienen
- Test der Funktionen und des User-Interfaces

Realisierung / Programmierung

Die Hauptklasse der App stellt *TodoPage()* dar. Diese beinhaltet den Code der gesamten Benutzerschnittstelle und weiterführenden Widgets und Klassen. Beim Initialisieren der Seite wird in der Funktion *_loadTodos()* aufgerufen, welche dafür sorgt, dass zuvor gespeicherte Einträge geladen werden. Diese Einträge werden mit Hilfe des Pakets „*shared_preferences*“ direkt in der App gespeichert. Hier wurde bewusst auf die Verwendung von externen Dateien oder gar Datenbanken verzichtet, da hier nur kleine Mengen nicht komplexer Daten gespeichert werden müssen.

Der Body von *TodoPage()* enthält einen *ListView.builder*, welche eine Liste von *TodoItem()*-Objekten erzeugt. Dabei wird jedes ToDo in einem *ListTile*-Widget dargestellt.

Die Klasse *TodoPageState()* implementiert außerdem Methoden zum Hinzufügen (*_addTodo()*), Bearbeiten (*_editTodo()*), Erledigen (*_completeTodo()*) und Widerrufen der Erledigung (*_uncompleteTodo()*) von ToDos. Jede dieser Methoden aktualisiert den aktuellen State der Seite und speichert die ToDos mit Hilfe von der Klasse TodoStorage, welche in der Datei todo_storage.dart zu finden ist.

Die Klasse TodoItem bekommt die Parameter *note* und *todo* übergeben, welche aus der Eingabe der Textfelder aus dem Widget *inputPopup()* kommen. Dieses wird durch die *onPressed*-Funktion vom Widget *newTodoButton()* aufgerufen.

Testverfahren

Funktionen

- Erstellen eines *TodoItem()*
- *_addTodo()*: neuen Eintrag hinzufügen und prüfen, ob dieser in der Liste erscheint
- *_completeTodo()*: Eintrag erledigen und prüfen, ob der Eintrag in die erledigten ToDos verschoben wird
- *_uncompleteTodo()*: bereits abgeschlossenen Eintrag widerrufen und prüfen, ob der Eintrag wieder in die Ursprungsliste verschoben wird
- *_editTodo()*: Inhalt eines bestehenden Eintrages ändern und prüfen, ob diese übernommen wurden

User-Interface

- Prüfen, ob alle Objekte richtig in den Listen dargestellt werden
- Prüfen, ob alle UI-Elemente wie erwartet funktionieren
- Prüfen, ob beim Klicken jedes Buttons die richtige Funktion aufgerufen wird
- Scrollverhalten der Seite
- Dynamisches Layout beim Ändern der Größe des Fensters

Fazit

Generell verlief das Projekt ohne weitere Komplikationen oder große Probleme. Dies liegt zum größten Teil wohl daran, dass ich mich für die Realisierung für das Framework „Flutter“ entschieden habe, welches ich in meinem Ausbildungsbetrieb täglich nutze. Meine Erfahrung mit diversen Besonderheiten und Lösungsansätzen hat mir dabei erheblich geholfen und zu einem reibungslosen Ablauf beigetragen.

Das Projekt wurde weitgehend wie geplant realisiert. Lediglich die Idee des Einsatzes einer externen Datenbank oder das Speichern in eine externe Datei wurde verworfen, da dies für die kleine Menge an Daten absolut überdimensioniert war und sich eine simplere Implementierung zur Speicherung auf direkter App-Ebene anbot.

Da sowohl der Umfang als auch die Komplexität des Projekts überschaubar blieben, war keine aufwendige Strukturierung oder Vorplanung erforderlich. Stattdessen konnte ich erfolgreich nach dem Prinzip „einfach drauf los“ arbeiten. Allerdings wurde dabei deutlich, dass ich für zukünftige Projekte – insbesondere mit steigender Komplexität – ein wesentlich höheres Augenmerk auf eine strukturierte Herangehensweise legen werde.