
Software Project Task Manager

Prof. Dr. Agnès Voisard
Muhammed-Ugur Karagülle

31. Juli 2023

Team:

Mohamad Weam Albunni
Redwan Albunni
Hussam Al Hallaq
Yousha khadra

Contents

1	Vorwort:	2
2	Einleitung:	2
3	Einführung in die Domäne und Motivation:	3
4	Related Work:	3
5	Zielsetzung:	5
6	Implementierung:	6
6.1	Model:	6
6.2	View:	8
6.2.1	HomePage:	8
6.2.2	MonthlytasksPage:	9
6.2.3	WeeklyTasksPage:	10
6.2.4	AddTaskPage:	10
6.2.5	EditTaskPage:	10
6.2.6	RegistrationPage:	11
6.2.7	LoginPage:	11
6.3	Controller:	12
6.3.1	TaskController:	12
6.3.2	ThemeService:	12
6.3.3	NotifyHelper:	12
7	Evaluation:	15
8	Fazit:	15

1 Vorwort:

Sehr geehrte Leserinnen und Leser, mit großer Freude präsentieren wir Ihnen die vorliegende Dokumentation unseres Softwareprojekts, dem Taskmanager-App.

Diese App wurde im Rahmen unseres Studiums an der Freien Universität Berlin entwickelt und ist das Ergebnis intensiver Arbeit und engagierter Teamarbeit. Wir sind stolz darauf, Ihnen unser Projekt präsentieren zu können und hoffen, dass Sie in dieser Dokumentation wertvolle Einblicke und Informationen finden werden.

Unser Ziel bei der Entwicklung des Taskmanager-App war es, eine innovative Lösung zur effizienten Organisation und Verwaltung von Aufgaben zu schaffen. Wir waren von der Bedeutung des Taskmanagements im persönlichen Leben, der Arbeit und dem Projektmanagement überzeugt und erkannten die Herausforderungen, die viele Menschen bei der Aufgabenverwaltung haben. Basierend auf dieser Motivation begannen wir mit der Konzeption und Umsetzung der App.

2 Einleitung:

Die Einleitung dieser Dokumentation bietet Ihnen einen umfassenden Überblick über das Taskmanager-App-Projekt. Wir werden Ihnen die zentralen Aspekte und Ziele unseres Projekts vorstellen und Ihnen einen Einblick in den Entwicklungsprozess geben.

Wir beginnen mit einer detaillierten Beschreibung der Domäne des Taskmanagements und erläutern die Bedeutung dieser Thematik im persönlichen Leben, der Arbeit und dem Projektmanagement. Dabei zeigen wir auf, wie eine effektive Aufgabenverwaltung zur Steigerung der Produktivität und Erreichung von Zielen beitragen kann.

Anschließend gehen wir auf die Motivation hinter der Entwicklung des Taskmanager-App-Projekts ein. Wir erläutern die Herausforderungen, denen wir begegnet sind und die uns dazu motiviert haben, eine Lösung zu entwickeln, die den individuellen Anforderungen und Präferenzen der Benutzer gerecht wird. Insbesondere betonen wir die Rolle der bedingten Planung bei der Aufgabenorganisation und wie dies in unserer App umgesetzt wurde.

Im nächsten Abschnitt befassen wir uns mit verwandten Arbeiten im Bereich des Taskmanagements. Hier stellen wir Ihnen inspirierende Beispiele und bewährte Praktiken anderer Taskmanagement-Apps vor, die uns bei der Entwicklung unseres Projekts inspiriert haben. Wir zeigen auf, wie wir innovative Funktionen und Lösungen in unsere eigene Anwendung integriert haben, um den individuellen Anforderungen gerecht zu werden.

Abschließend formulieren wir unsere Zielsetzung klar und präzise. Wir beschreiben die Hauptbeiträge unseres Projekts, wie die Implementierung von benutzerdefinierten Erinnerungen und wiederkehrenden Aufgaben, und betonen den Mehrwert, den unser Taskmanager-App den Benutzern bietet. Wir sind zuversichtlich, dass unsere App einen positiven Einfluss auf die Produktivität und Organisation der Benutzer haben wird.

Wir laden Sie ein, diese Dokumentation weiterzulesen und einen detaillierten Einblick in unser Taskmanager-App-Projekt zu erhalten. Wir hoffen, dass Sie von den präsentierten Informationen profitieren und einen Einblick in unsere Arbeit gewinnen können.

3 Einführung in die Domäne und Motivation:

Die Domäne des Taskmanagements spielt eine entscheidende Rolle in verschiedenen Bereichen wie dem persönlichen Leben, der Arbeit und dem Projektmanagement. Es ermöglicht den Benutzern, ihre Aufgaben effektiv zu planen, zu verfolgen und zu verwalten, um ihre Produktivität zu steigern und ihre Ziele erfolgreich zu erreichen. Die Fähigkeit, Aufgaben zu organisieren und den Überblick zu behalten, ist für eine effiziente Arbeitsweise unerlässlich.

Im persönlichen Leben kann das Taskmanagement dazu beitragen, den Alltag besser zu strukturieren und wichtige Aufgaben nicht zu vergessen. Es ermöglicht den Benutzern, ihre Zeit optimal einzuteilen und Prioritäten zu setzen, um ihre persönlichen Ziele zu erreichen. Ob es darum geht, Haushaltsaufgaben zu erledigen, persönliche Projekte umzusetzen oder Freizeitaktivitäten zu planen, ein effektives Taskmanagement unterstützt dabei, den Tag zu organisieren und Stress zu reduzieren. In der Arbeitswelt ist das Taskmanagement von entscheidender Bedeutung, um Projekte erfolgreich umzusetzen und den Arbeitsablauf effizient zu gestalten. Das Festlegen von Zielen, das Planen von Aufgaben, das Festlegen von Fristen und das Verfolgen des Fortschritts sind entscheidende Schritte, um Projekte termingerecht abzuschließen. Eine strukturierte und organisierte Arbeitsweise ermöglicht es den Mitarbeitern, ihre Aufgaben effizient zu erledigen und Engpässe zu vermeiden.

Auch im Projektmanagement ist das Taskmanagement von großer Bedeutung. Die erfolgreiche Umsetzung von Projekten erfordert eine genaue Planung und Aufgabenverwaltung. Das Festlegen von Meilensteinen, das Zuweisen von Aufgaben an Teammitglieder, das Verfolgen des Projektfortschritts und das rechtzeitige Identifizieren von Engpässen sind entscheidend für den Erfolg eines Projekts. Ein effektives Taskmanagement ermöglicht es den Projektmanagern, den Überblick über die Aufgaben und den Fortschritt zu behalten und gegebenenfalls Anpassungen vorzunehmen.

Die Entwicklung unseres Taskmanager-Apps wurde von diesen Herausforderungen und Problemen motiviert. Wir erkannten den Bedarf an einer leistungsstarken Softwareanwendung, die es Benutzern ermöglicht, ihre Aufgaben effizient zu organisieren und zu verwalten, unabhängig von der spezifischen Domäne. Unsere Motivation besteht darin, eine umfassende Lösung anzubieten, die den individuellen Bedürfnissen und Präferenzen gerecht wird und Benutzern dabei hilft, ihre Produktivität zu steigern und ihre Ziele erfolgreich zu erreichen.

In den folgenden Abschnitten dieser Dokumentation werden wir detailliert auf die Related work, Implementierung, Evaluation und Einschränkungen unseres Taskmanager-App-Projekts eingehen. Wir sind zuversichtlich, dass unsere App einen Mehrwert für die Benutzer bietet und eine effektive Lösung für die Aufgabenverwaltung in verschiedenen Domänen darstellt.

4 Related Work:

Bei unserer eingehenden Untersuchung verwandter Arbeiten im Bereich des Taskmanagements für den persönlichen Gebrauch haben wir verschiedene Quellen herangezogen und uns intensiv mit mehreren führenden Apps auseinandergesetzt. Hierbei haben wir nicht nur Informationen aus den offiziellen Websites der Apps bezogen, sondern auch Testversionen heruntergeladen und selbst ausführlich getestet. Durch diese umfassende Herangehensweise konnten wir wertvolle Erkenntnisse gewinnen und bewährte Praktiken identifizieren, die in der Branche erfolgreich angewendet werden.

Eine der Quellen, die wir konsultiert haben, ist die App **Todoist**. Todoist ist eine äußerst beliebte und weit verbreitete Taskmanagement-App, die sowohl für den persönlichen als auch für den beruflichen Gebrauch entwickelt wurde. Wir haben die Funktionalitäten von Todoist analysiert, darunter die Möglichkeit, Aufgaben zu priorisieren, Kategorien zuzuweisen, Termine festzulegen und Erinnerungen zu erhalten. Darüber hinaus haben wir uns mit der benutzerfreundlichen Oberfläche und der nahtlosen Integration der App auf verschiedenen Plattformen vertraut gemacht. Indem wir Todoist als Inspirationsquelle nutzen, konnten wir bewährte Praktiken und erfolgreiche Ansätze für die Aufgabenverwaltung verstehen und darauf aufbauen.

Eine weitere Quelle, die wir eingehend untersucht haben, ist die App **Any.do**. Any.do bietet ähnliche Funktionen wie Todoist und ermöglicht Benutzern, ihre Aufgaben zu organisieren, zu priorisieren und zu verfolgen. Besondere Merkmale von Any.do sind die Möglichkeit der Spracheingabe, das Anhängen von Dateien und die nahtlose Zusammenarbeit durch das Teilen von Aufgaben mit anderen Benutzern. Wir haben uns mit der intuitiven Benutzeroberfläche und der einfachen Handhabung der App vertraut gemacht und deren Einfluss auf die Benutzererfahrung analysiert.

Zudem haben wir die von Microsoft entwickelte Taskmanagement-App **Microsoft To Do** eingehend studiert. Diese App zeichnet sich durch ihre nahtlose Integration in andere Microsoft-Produkte wie Outlook, OneNote und Teams aus. Benutzer können ihre Aufgaben einfach organisieren, synchronisieren und mit anderen teilen. Wir haben uns mit der vertrauten Benutzeroberfläche von Microsoft To Do vertraut gemacht und die Vorteile der nahtlosen Integration in das Microsoft-Ökosystem untersucht.

Durch unsere umfassende Recherche und die ausführliche Erprobung dieser Apps konnten wir tiefgreifende Einblicke gewinnen und bewährte Praktiken im Bereich des Taskmanagements für den persönlichen Gebrauch identifizieren. Die gewonnenen Erkenntnisse haben es uns ermöglicht, unsere eigene Taskmanagement-App mit innovativen Funktionen und Lösungen zu entwickeln, die den spezifischen Anforderungen und Präferenzen der Benutzer gerecht werden. Wir haben bewährte Praktiken aufgegriffen und gleichzeitig unsere eigenen einzigartigen Ansätze und Verbesserungen integriert, um eine qualitativ hochwertige und effektive Lösung zu schaffen.

5 Zielsetzung:

Unsere Zielsetzung bei der Entwicklung der Taskmanager-App war es, eine umfassende Lösung für die individuelle Aufgabenverwaltung zu schaffen, die den spezifischen Bedürfnissen und Präferenzen der Benutzer gerecht wird. Wir hatten das klare Ziel vor Augen, eine Anwendung zu entwickeln, die Benutzern dabei hilft, ihre Aufgaben effektiv zu organisieren, den Überblick zu behalten und ihre persönlichen Ziele erfolgreich zu erreichen. Eine zentrale Zielsetzung unseres Projekts war es, den Benutzern die Möglichkeit zu geben, benutzerdefinierte Erinnerungen für ihre Aufgaben festzulegen. Wir wollten sicherstellen, dass Benutzer keine wichtigen Termine oder Fristen verpassen und ihre Aufgaben rechtzeitig erledigen können.

Durch die Implementierung von Erinnerungsfunktionen können Benutzer spezifische Datum- und Uhrzeitangaben für Erinnerungen festlegen, um an bevorstehende Aufgaben erinnert zu werden. Diese personalisierten Erinnerungen helfen den Benutzern, ihre Aufgaben effizient zu planen und zu organisieren.

Ein weiteres Ziel unseres Projekts war es, die Möglichkeit zur Erstellung wiederkehrender Aufgaben anzubieten. Wir erkannten, dass viele Aufgaben regelmäßig und in bestimmten Zeitintervallen erledigt werden müssen. Deshalb wollten wir unseren Benutzern die Möglichkeit geben, Aufgaben so einzurichten, dass sie automatisch zu bestimmten Zeitpunkten wiederholt werden. Durch die Einrichtung wiederkehrender Aufgaben werden Benutzer kontinuierlich daran erinnert, diese regelmäßigen Aufgaben auszuführen und somit eine konstante Produktivität zu gewährleisten.

Des Weiteren war es uns ein Anliegen, eine benutzerfreundliche und intuitive Benutzeroberfläche zu schaffen. Wir wollten sicherstellen, dass unsere App leicht verständlich ist und die Benutzer in der Lage sind, ihre Aufgaben schnell und einfach hinzuzufügen, zu bearbeiten und zu verwalten. Durch eine übersichtliche Darstellung der Aufgabenliste, klare Anweisungen und eine intuitive Navigation haben wir darauf geachtet, dass die App benutzerfreundlich und zugänglich ist. Neben diesen konkreten Zielen war es uns wichtig, eine qualitativ hochwertige Softwarelösung zu entwickeln, die stabil, zuverlässig und skalierbar ist. Wir legten Wert auf eine robuste Architektur und einen sauberen Code, um eine langfristige Wartbarkeit und Erweiterbarkeit der App sicherzustellen. Außerdem haben wir umfangreiche Tests durchgeführt, um sicherzustellen, dass die App reibungslos funktioniert und den Benutzeranforderungen gerecht wird.

Insgesamt war unsere Zielsetzung, eine leistungsstarke und effektive Taskmanager-App zu entwickeln, die den Benutzern dabei hilft, ihre Aufgaben effizient zu organisieren und erfolgreich abzuschließen. Durch die Erfüllung dieser Ziele wollten wir eine positive Auswirkung auf die Produktivität, Organisation und das allgemeine Wohlbefinden der Benutzer erzielen.

6 Implementierung:

Die Task-Management-App implementiert das Model-View-Controller (MVC)-Muster, das eine klare Trennung von Datenmodellen (Model), Benutzeroberfläche (View) und Geschäftslogik (Controller) ermöglicht. Dieses Muster fördert eine strukturierte und gut organisierte Architektur, die die Wartbarkeit, Skalierbarkeit und Testbarkeit der Anwendung verbessert. Die App besteht aus den folgenden Komponenten, die jeweils ihre eigenen Verantwortlichkeiten haben:

6.1 Model:

Das Model in der Task-Management-App spielt eine zentrale Rolle bei der Verwaltung und Darstellung der Daten. Es besteht aus der Task-Klasse, die die Informationen einer Aufgabe repräsentiert. Die Task-Klasse enthält verschiedene Attribute, um die Eigenschaften einer Aufgabe zu speichern:

ID: Eine eindeutige Kennung, die jede Aufgabe identifiziert. Die ID wird verwendet, um Aufgaben eindeutig zu identifizieren und sie in der Datenbank oder anderen Speichermedien zu verwalten.

Titel: Der Titel einer Aufgabe beschreibt kurz, worum es in der Aufgabe geht. Zum Beispiel könnte der Titel "Einkaufsliste" lauten.

Beschreibung: Die Beschreibung einer Aufgabe ermöglicht eine detailliertere Darstellung der Aufgabenanforderungen oder -hinweise. Sie kann zusätzliche Informationen enthalten, die für das Verständnis und die Bearbeitung der Aufgabe wichtig sind.

Abschlussstatus: Ein Boolean-Wert, der angibt, ob die Aufgabe abgeschlossen wurde oder nicht. Wenn der Abschlussstatus auf "true" gesetzt ist, bedeutet dies, dass die Aufgabe erledigt wurde.

Datum: Das Datum, an dem die Aufgabe erledigt werden soll oder erledigt wurde. Dies ermöglicht es dem Benutzer, Aufgaben nach Datum zu organisieren und anzuzeigen.

Start- und Endzeit: Eine Aufgabe kann eine geplante Start- und Endzeit haben, die den Zeitraum angibt, in dem die Aufgabe erledigt werden sollte.

Farbe: Eine Farbzuzuordnung für die Aufgabe, die es dem Benutzer ermöglicht, Aufgaben visuell zu kennzeichnen und zu gruppieren. Dies kann bei der schnellen Identifizierung und Unterscheidung von Aufgaben helfen.

Erinnerungszeit: Die Erinnerungszeit ist der Zeitpunkt, zu dem der Benutzer eine Benachrichtigung über die Aufgabe erhalten möchte. Dies hilft dem Benutzer, wichtige Aufgaben nicht zu vergessen.

Wiederholungsmodus: Der Wiederholungsmodus gibt an, ob die Aufgabe regelmäßig wiederholt werden soll. Es kann verschiedene Optionen wie täglich, wöchentlich, monatlich oder benutzerdefiniert geben.

In der Task-Klasse wird die 'fromJson'-Methode verwendet, um ein Task-Objekt aus JSON-

Daten zu erstellen, und die 'toJson'-Methode kann verwendet werden, um ein Task-Objekt in JSON zu konvertieren. Diese Methoden ermöglichen den einfachen Austausch von Daten zwischen dem Model und anderen Komponenten der App, wie der Datenbank oder der Benutzeroberfläche.

wir haben die DBHelper-Klasse als Schnittstelle zur Datenbank implementiert, sie ermöglicht das Speichern, Abrufen, Aktualisieren und Löschen von Aufgaben. Sie verwendet eine lokale Datenbank, um die Aufgaben persistent zu speichern. Dabei nutzt sie das sqflite-Plugin, das eine SQLite-Datenbank in Flutter unterstützt. Hierfür haben wir ein Entity-Relationship-Modell **figure 6.1** erstellt, das die Datenstruktur der Aufgaben und ihre Beziehungen definiert und effizient speichert.

Die DBHelper-Klasse enthält Methoden wie insertTask, getTasks, updateTask und deleteTask, um auf die Datenbank zuzugreifen und die Aufgaben zu verwalten.

Die insertTask-Methode ermöglicht das Hinzufügen einer neuen Aufgabe zur Datenbank. Sie nimmt ein Task-Objekt entgegen, konvertiert es in das entsprechende Datenbankformat und fügt es der Datenbank hinzu.

Die getTasks-Methode ruft alle Aufgaben aus der Datenbank ab. Sie gibt eine Liste von Task-Objekten zurück, die die gespeicherten Aufgaben repräsentieren. Dabei werden die Daten aus der Datenbank gelesen und in Task-Objekte umgewandelt.

Die updateTask-Methode ermöglicht das Aktualisieren einer vorhandenen Aufgabe in der Datenbank. Sie nimmt ein Task-Objekt entgegen, sucht die entsprechende Aufgabe in der Datenbank und aktualisiert die zugehörigen Daten.

Die deleteTask-Methode ermöglicht das Löschen einer Aufgabe aus der Datenbank. Sie nimmt ein Task-Objekt entgegen, sucht die entsprechende Aufgabe in der Datenbank und entfernt sie.

Der TaskController arbeitet eng mit der DBHelper-Klasse zusammen, um die Aufgabenliste abzurufen und zu aktualisieren. Wenn beispielsweise der TaskController eine neue Aufgabe hinzufügt, ruft er die insertTask-Methode in der DBHelper-Klasse auf, um die Aufgabe in der Datenbank zu speichern. Wenn der TaskController Aufgaben abrufen, verwendet er die getTasks-Methode der DBHelper-Klasse, um die Aufgaben aus der Datenbank abzurufen und sie dem TaskModel hinzuzufügen.

Durch die Verwendung der DBHelper-Klasse wird die Persistenz der Daten gewährleistet, sodass die Aufgaben auch nach dem Schließen und erneuten Öffnen der App erhalten bleiben. Die DBHelper-Klasse spielt eine wichtige Rolle bei der Verbindung zwischen dem Model und der lokalen Datenbank und unterstützt die Datenverwaltungsfunktionen der Task-Management-App. Sie ermöglicht eine effiziente Speicherung und Abfrage von Aufgaben und trägt zur Zuverlässigkeit und Skalierbarkeit der App bei. Das Model spielt eine entscheidende Rolle bei der Datenverwaltung und -darstellung in der Task-Management-App. Es ermöglicht das Erstellen, Speichern, Laden und Aktualisieren von Aufgaben und bietet eine klare Datenstruktur, um eine konsistente und zuverlässige Erfahrung für den Benutzer zu gewährleisten.

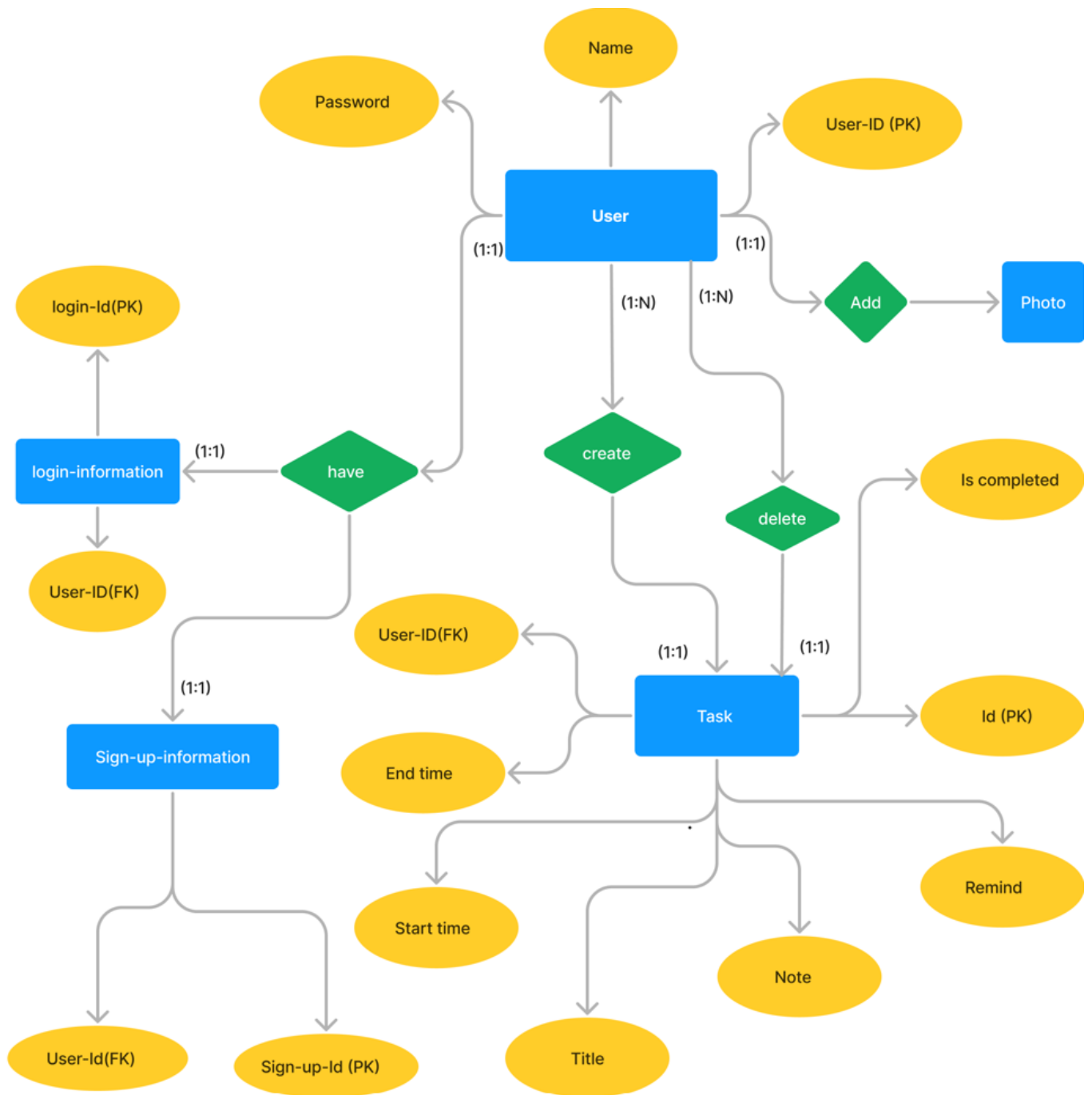


Figure 6.1: ERM

6.2 View:

Die View-Komponente ist für die Benutzeroberfläche der App verantwortlich und besteht aus verschiedenen Flutter-Seiten, die dem Benutzer verschiedene Ansichten und Interaktionen bieten:

6.2.1 HomePage:

Die HomePage ist die Hauptseite der App und bietet dem Benutzer eine Übersicht über die Aufgaben für den ausgewählten Tag. Sie besteht aus verschiedenen Komponenten:

AppBar: Die AppBar enthält die Schaltfläche zum Wechseln des App-Themas, mit der der

Benutzer zwischen dem hellen und dem dunklen Modus wechseln kann. Zusätzlich gibt es eine Schaltfläche zur Navigation zur monatlichen Aufgabenansicht, die den Benutzer zur `MonthlytasksPage` weiterleitet. Die `AppBar` stellt eine konsistente Navigationsleiste bereit, die auf allen Seiten der App angezeigt wird.

AddTaskBar: Die `AddTaskBar` ist eine Komponente, die das aktuelle Datum anzeigt und dem Benutzer ermöglicht, neue Aufgaben hinzuzufügen. Wenn der Benutzer auf die `Add-Task` Schaltfläche klickt, wird die `AddTaskPage` geöffnet, auf der er weitere Details zur Aufgabe eingeben kann.

DateBar: Die `DateBar` ist eine Komponente, die es dem Benutzer ermöglicht, den ausgewählten Tag zu ändern. Sie enthält normalerweise einen Kalender oder eine Datumswähler-Komponente, mit der der Benutzer das Datum auswählen kann. Wenn der Benutzer das Datum ändert, werden die Aufgaben für den entsprechenden Tag in der `ShowTasks`-Komponente aktualisiert und angezeigt.

ShowTasks: Die `ShowTasks`-Komponente enthält die Liste der Aufgaben für den ausgewählten Tag und zeigt sie durch Aufgabenkacheln (`TaskTiles`) an. Jede Aufgabenkachel zeigt den Titel, die Beschreibung, die Start- und Endzeit sowie den Status der Aufgabe an. Der Benutzer kann auf eine Aufgabenkachel klicken, um weitere Details zur Aufgabe anzuzeigen oder die Aufgabe als abgeschlossen zu markieren. Wenn eine Aufgabe als abgeschlossen markiert wird, wird ihr Status aktualisiert und die Kachel wird visuell hervorgehoben, um anzuzeigen, dass sie erledigt ist.

Die `HomePage` bietet dem Benutzer eine intuitive Benutzeroberfläche, um Aufgaben für den ausgewählten Tag anzuzeigen, neue Aufgaben hinzuzufügen und vorhandene Aufgaben zu bearbeiten oder als erledigt zu markieren.

6.2.2 `MonthlytasksPage`:

Die `MonthlytasksPage` ist eine Seite, die dem Benutzer eine monatliche Aufgabenübersicht bietet. Sie ermöglicht es dem Benutzer, Aufgaben für den gesamten Monat anzuzeigen und durch den Kalender zu navigieren, um Aufgaben für bestimmte Tage auszuwählen. Die `MonthlytasksPage` besteht normalerweise aus den folgenden Komponenten:

AppBar: Die `AppBar` enthält die Schaltfläche zur Rückkehr zur `HomePage`. Sie bietet eine konsistente Navigationsleiste für die App.

Calendar: Der Kalender ist eine Komponente, die den Benutzer durch den aktuellen Monat navigieren lässt. Der Benutzer kann auf ein bestimmtes Datum im Kalender klicken, um die Aufgaben für diesen Tag anzuzeigen.

ShowTasks: Die `ShowTasks`-Komponente zeigt die Aufgaben für den ausgewählten Monat und das ausgewählte Datum an. Sie verwendet Aufgabenkacheln (`TaskTiles`), um die Aufgabeninformationen anzuzeigen. Der Benutzer kann auf eine Aufgabenkachel klicken, um weitere Details zur Aufgabe anzuzeigen.

Die `MonthlytasksPage` ermöglicht es dem Benutzer, Aufgaben für einen bestimmten Monat anzuzeigen und zwischen den Monaten zu navigieren, um Aufgaben für verschiedene Tage anzuzeigen.

6.2.3 WeeklyTasksPage:

Die WeeklyTasksPage ist eine Seite, die dem Benutzer eine wöchentliche Aufgabenübersicht bietet. Sie ermöglicht es dem Benutzer, Aufgaben für eine bestimmte Woche anzuzeigen und durch den Kalender zu navigieren, um Aufgaben für bestimmte Tage auszuwählen. Die WeeklyTasksPage besteht normalerweise aus den folgenden Komponenten:

AppBar: Die AppBar enthält die Schaltfläche zur Rückkehr zur HomePage. Sie stellt eine konsistente Navigationsleiste für die App bereit.

Calendar: Der Kalender ist eine Komponente, die den Benutzer durch die aktuelle Woche navigieren lässt. Der Benutzer kann auf ein bestimmtes Datum im Kalender klicken, um die Aufgaben für diesen Tag anzuzeigen.

ShowTasks: Die ShowTasks-Komponente zeigt die Aufgaben für die ausgewählte Woche und das ausgewählte Datum an. Sie verwendet Aufgabenkacheln (TaskTiles), um die Aufgabeninformationen anzuzeigen. Der Benutzer kann auf eine Aufgabenkachel klicken, um weitere Details zur Aufgabe anzuzeigen oder die Aufgabe als zu löschen.

Die WeeklyTasksPage ermöglicht es dem Benutzer, Aufgaben für eine bestimmte Woche anzuzeigen und zwischen den Wochen zu navigieren, um Aufgaben für verschiedene Tage anzuzeigen.

6.2.4 AddTaskPage:

Die AddTaskPage ist eine Seite, die es dem Benutzer ermöglicht, eine neue Aufgabe hinzuzufügen und die Details wie Titel, Beschreibung, Datum, Start- und Endzeit, Farbe, Erinnerungszeit und Wiederholungsmodus festzulegen. Die AddTaskPage besteht normalerweise aus den folgenden Komponenten:

Formularfelder: Die AddTaskPage enthält verschiedene Formularfelder, in die der Benutzer die Informationen für die neue Aufgabe eingeben kann. Dazu gehören normalerweise Textfelder für den Titel und die Beschreibung der Aufgabe, ein Datumsauswahlfeld für das Datum der Aufgabe, Auswahlfelder für die Start- und Endzeit, Farbwähler für die Aufgabenfarbe, ein Erinnerungszeitfeld und Optionen für den Wiederholungsmodus.

Schaltfläche zum Speichern: Nachdem der Benutzer die Aufgabeninformationen eingegeben hat, enthält die AddTaskPage eine Schaltfläche zum Speichern der Aufgabe. Wenn der Benutzer auf diese Schaltfläche klickt, wird die eingegebene Aufgabe dem Model hinzugefügt und die Aufgabenliste wird aktualisiert.

Die AddTaskPage ermöglicht es dem Benutzer, neue Aufgaben hinzuzufügen und die Details für die Aufgaben festzulegen.

6.2.5 EditTaskPage:

Die EditTaskPage ist eine Flutter-Seite, auf der Benutzer Aufgaben bearbeiten können. Sie enthält Eingabefelder für den Titel und die Notizen der Aufgabe sowie Dropdown-Listen für das Datum, die Start- und Endzeit, die Erinnerungszeit und den Wiederholungsmodus. Die ausgewählten Werte werden im TaskController gespeichert und ermöglichen es dem Benutzer, die Änderungen zu speichern.

Die Seite verwendet den TaskController, um auf die Aufgabenliste zuzugreifen und Datenbankoperationen durchzuführen. Die Textfelder für den Titel und die Notizen verwenden TextEditingController, um den Inhalt zu steuern. Es gibt auch TextButton, um das Datum, die Start- und Endzeit auszuwählen, sowie Dropdown-Listen für die Erinnerungszeit und den Wiederholungsmodus.

Vor dem Speichern der Änderungen erfolgt eine Validierung der Zeit, um sicherzustellen, dass die Zeitdauer positiv ist und gültige Kombinationen von Start- und Endzeit verwendet werden. Bei ungültigen Eingaben wird dem Benutzer eine Fehlermeldung angezeigt. Nach dem Speichern werden die Änderungen in der Datenbank aktualisiert und die Aufgabenliste neu geladen.

Die EditTaskPage bietet eine benutzerfreundliche Möglichkeit, Aufgaben zu bearbeiten, und stellt sicher, dass die Änderungen korrekt gespeichert werden.

6.2.6 RegistrationPage:

Die RegistrationPage ist eine Seite, die es dem Benutzer ermöglicht, sich in der App zu registrieren. Sie enthält normalerweise Formularfelder, in die der Benutzer seinen Benutzernamen, und sein Passwort eingeben kann. Nachdem der Benutzer seine Registrierungsinformationen eingegeben hat, kann er auf eine Schaltfläche zum Absenden klicken, um seine Registrierung abzuschließen. Die RegistrationPage ermöglicht es dem Benutzer, sich in der App zu registrieren und ein Benutzerkonto zu erstellen.

6.2.7 LoginPage:

Die LoginPage ist eine Seite, auf der sich der Benutzer in der App anmelden kann. Sie enthält normalerweise Formularfelder, in die der Benutzer seinen Benutzernamen und sein Passwort eingeben kann. Nachdem der Benutzer seine Anmeldeinformationen eingegeben hat, kann er auf eine Schaltfläche zum Anmelden klicken, um auf sein Benutzerkonto zuzugreifen. Die LoginPage ermöglicht es dem Benutzer, sich in der App anzumelden und auf sein Benutzerkonto zuzugreifen.

Jede Seite hat ihre eigene visuelle Darstellung, die durch die Verwendung von Flutter-Widgets und vordefinierten Stilen erstellt wird. Die Seiten bieten Benutzerinteraktionen wie das Klicken auf Schaltflächen, das Auswählen von Daten und das Eingeben von Text.

6.3 Controller:

Der Controller in der Task-Management-App ist für die Geschäftslogik verantwortlich und besteht aus verschiedenen Klassen, die als Vermittler zwischen dem Model und der View fungieren. Jede dieser Klassen hat spezifische Verantwortlichkeiten und ermöglicht die Interaktion und Kommunikation zwischen den verschiedenen Komponenten der App.

6.3.1 TaskController:

Der TaskController ist einer der Hauptcontroller in der App und übernimmt die Verwaltung der Aufgabenliste. Er ist für die Durchführung verschiedener Aktionen wie dem Hinzufügen, Abrufen, Aktualisieren und Löschen von Aufgaben verantwortlich. Der TaskController enthält eine `taskList`, die eine `RxList<Task>` ist und die Liste der Aufgaben enthält. Diese `RxList` ermöglicht es, Änderungen an der Liste automatisch zu erkennen und die Benutzeroberfläche entsprechend zu aktualisieren.

Der TaskController bietet eine Reihe von Methoden, um mit den Aufgaben zu interagieren. Die `addTask`-Methode ermöglicht es, eine neue Aufgabe zur Liste hinzuzufügen. Dabei wird das übergebene Task-Objekt in der Datenbank gespeichert und zur `taskList` hinzugefügt. Die `getTasks`-Methode ruft die Aufgaben des Benutzers aus der Datenbank ab und aktualisiert die `taskList` entsprechend. Die `deleteTask`-Methode ermöglicht das Löschen einer Aufgabe aus der Datenbank und der `taskList`. Die `markTaskCompleted`-Methode markiert eine Aufgabe als abgeschlossen, indem der Abschlussstatus in der Datenbank und in der `taskList` aktualisiert wird.

Der TaskController arbeitet eng mit dem Model, insbesondere der Task-Klasse, zusammen, um die Aufgaben effizient zu verwalten. Er verwendet auch die Datenbank-Schnittstelle, wie die `DBHelper`-Klasse, um Datenbankoperationen durchzuführen und sicherzustellen, dass die `taskList` immer auf dem neuesten Stand ist. Der TaskController ermöglicht es der App, die Aufgabenliste zu organisieren, zu aktualisieren und Änderungen in Echtzeit darzustellen.

6.3.2 ThemeService:

Der ThemeService ist dafür verantwortlich, das Erscheinungsbild der App zu verwalten und dem Benutzer die Möglichkeit zu geben, zwischen dem hellen und dem dunklen Modus zu wechseln. Der ThemeService speichert den aktuellen Theme-Modus und aktualisiert das Design der App entsprechend. Dies ermöglicht es Benutzern, das Erscheinungsbild der App an ihre Vorlieben anzupassen und eine angenehme Benutzererfahrung zu gewährleisten.

Der ThemeService arbeitet mit den Flutter-Theming-Funktionen und -Paketen zusammen, um das Design der App anzupassen. Er ermöglicht es Benutzern, zwischen verschiedenen Farbpaletten, Hintergründen und Schriftarten zu wählen. Der ThemeService stellt sicher, dass das ausgewählte Theme in der gesamten App konsistent angewendet wird und dass Änderungen sofort sichtbar sind.

6.3.3 NotifyHelper:

Der NotifyHelper ist für das Planen und Anzeigen von Benachrichtigungen in der App verantwortlich. Er verwendet das Flutter-local-notifications-Plugin, um Benachrichtigungen für anstehende Aufgaben anzuzeigen.

Der NotifyHelper ermöglicht es Benutzern, an wichtige Aufgaben erinnert zu werden und keine Fristen oder Termine zu verpassen.

Der NotifyHelper bietet Funktionen zum Planen von Benachrichtigungen basierend auf den Startzeiten der Aufgaben. Wenn eine Aufgabe hinzugefügt oder aktualisiert wird, verwendet der NotifyHelper die Informationen aus dem Task-Objekt, um eine Benachrichtigung für die entsprechende Aufgabe zu planen. Die Benachrichtigungen enthalten den Titel, die Beschreibung und die Startzeit der Aufgabe.

Wenn eine Benachrichtigung empfangen wird und der Benutzer daraufklickt, zeigt der NotifyHelper einen Dialog an, der den Benutzer über die Aufgabe informiert und ihm die Möglichkeit gibt, die Details der Aufgabe anzuzeigen. Der NotifyHelper arbeitet eng mit dem TaskController zusammen, um Informationen über anstehende Aufgaben zu erhalten und Benachrichtigungen entsprechend zu planen.

Der Controller in der Task-Management-App ermöglicht es, die Geschäftslogik der Anwendung effizient zu verwalten.

Durch die Trennung der Verantwortlichkeiten zwischen dem Model, dem View und dem Controller wird eine klare Struktur erreicht, die Wartbarkeit, Skalierbarkeit und Testbarkeit fördert. Der Controller stellt sicher, dass die Benutzeraktionen richtig verarbeitet werden, das Model und die View synchronisiert werden und eine nahtlose Interaktion zwischen den Komponenten ermöglicht wird. Durch die Verwendung des MVC-Musters wird der Code der App besser strukturiert und wartbarer. Das Model ist für die Datenverwaltung zuständig, die View für die Darstellung der Benutzeroberfläche und der Controller für die Geschäftslogik und die Interaktionen zwischen Model und View.

Dieses Muster ermöglicht auch eine einfache Skalierung der App, da neue Funktionen und Seiten leicht hinzugefügt oder vorhandene geändert werden können, ohne den gesamten Code zu beeinträchtigen. Es fördert eine saubere Trennung der Verantwortlichkeiten und erleichtert die Wartung, Erweiterung und Testbarkeit der App. Zur besseren Veranschaulichung haben wir auch ein Flussdiagramm **figure.6.2** erstellt, das den Ablauf der App deutlich darstellt.

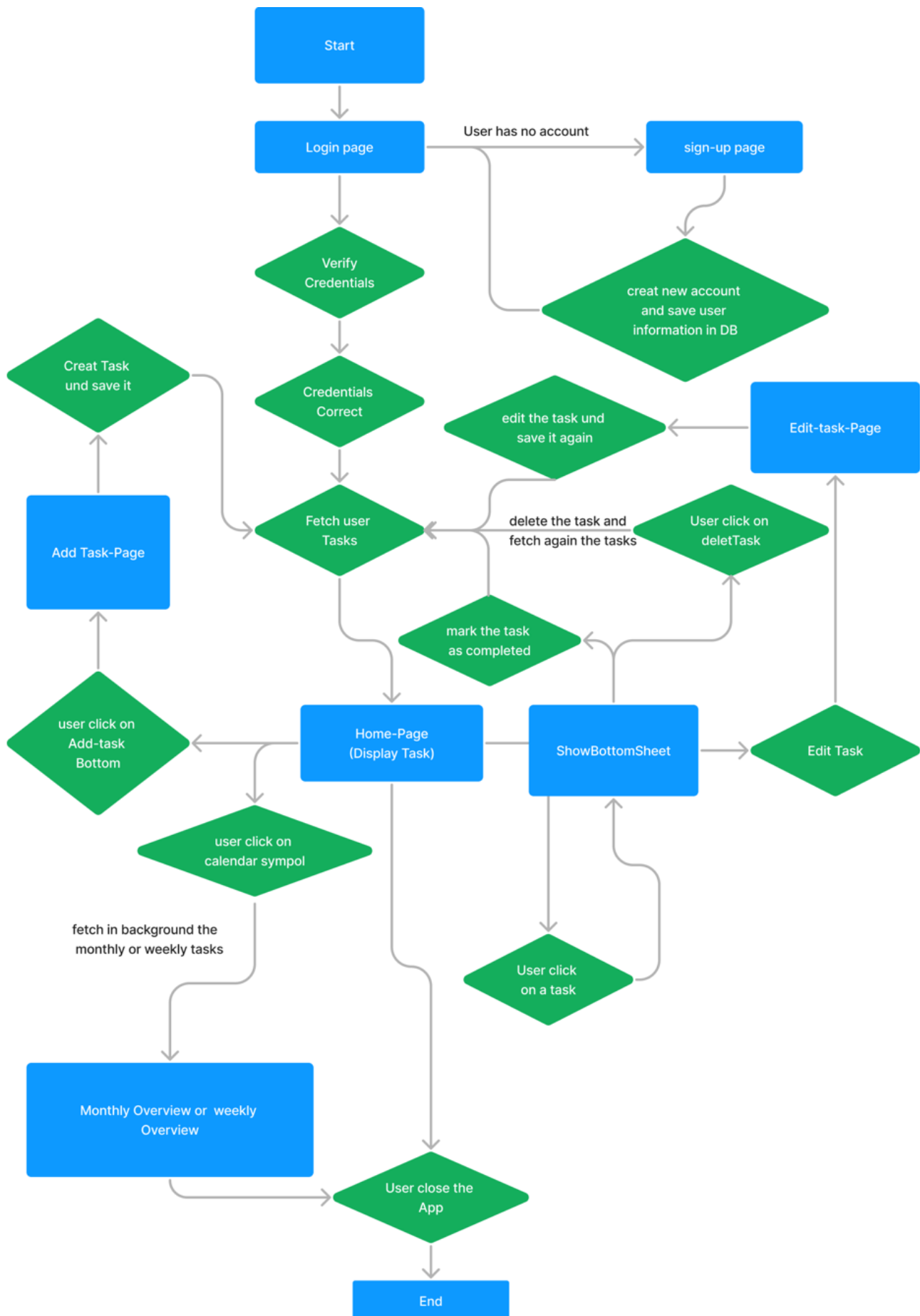


Figure 6.2: Flussdiagramm

7 Evaluation:

Die Evaluation unseres Taskmanager-Apps umfasste eine gründliche Überprüfung der Qualität, Leistung und Benutzerfreundlichkeit der Anwendung. Wir haben dabei verschiedene Testverfahren angewendet, um sicherzustellen, dass unsere App den hohen Standards entspricht, die wir uns gesetzt haben.

Ein wichtiger Teil der Evaluation waren umfangreiche Integrationstests. Dabei haben wir die reibungslose Zusammenarbeit aller Komponenten der App überprüft. Dazu gehörte unter anderem die Interaktion mit der Datenbank, um sicherzustellen, dass Aufgaben korrekt hinzugefügt, bearbeitet und gelöscht werden können und dass alle Änderungen synchronisiert werden. Darüber hinaus haben wir die korrekte Funktionalität aller Seiten und Funktionen getestet. Dies umfasste die Überprüfung, ob Benutzer Aufgaben problemlos anzeigen, filtern und sortieren können. Durch diese umfangreichen Integrationstests konnten wir sicherstellen, dass unsere App in Bezug auf Funktionalität und Datenverarbeitung einwandfrei funktioniert.

Des Weiteren haben wir Unittests durchgeführt, um die einzelnen Funktionen und Methoden in der App isoliert zu testen. Dadurch konnten wir sicherstellen, dass jede Komponente ordnungsgemäß arbeitet und die erwarteten Ergebnisse liefert. Wir haben verschiedene Szenarien simuliert und überprüft, ob Aufgaben korrekt hinzugefügt, bearbeitet und gelöscht werden, ob die Datenverarbeitung und -validierung fehlerfrei erfolgt und ob Benutzereingaben ordnungsgemäß verarbeitet werden. Durch diese Unittests konnten wir potenzielle Fehler oder Inkonsistenzen in der App identifizieren und beheben, um sicherzustellen, dass sie zuverlässig und stabil funktioniert.

Insgesamt hat die Evaluation gezeigt, dass unser Taskmanager-App erfolgreich die gesteckten Ziele erreicht hat. Die App arbeitet zuverlässig, bietet eine benutzerfreundliche Benutzeroberfläche und erfüllt die Anforderungen der Benutzer effektiv. Durch umfangreiche Integrationstests und Unittests konnten wir sicherstellen, dass alle Komponenten der App reibungslos zusammenarbeiten und die gewünschten Funktionen erfüllen. Das Feedback der Testbenutzer hat uns geholfen, die App weiter zu verbessern und die Benutzerfreundlichkeit kontinuierlich zu optimieren.

8 Fazit:

Abschließend möchten wir die wichtigsten Erkenntnisse und Ergebnisse unseres Taskmanager-Apps zusammenfassen. Unser Ziel war es, eine praktische und leistungsstarke Lösung für die individuelle Aufgabenverwaltung zu entwickeln, die Benutzern dabei hilft, ihre Aufgaben effektiv zu organisieren und zu verwalten. Unsere intensive Arbeit hat zu einem vielversprechenden Ergebnis geführt.

Unser Taskmanager-App zeichnet sich durch eine intuitive und benutzerfreundliche Benutzeroberfläche aus. Wir haben großen Wert daraufgelegt, dass die App einfach zu bedienen und leicht verständlich ist. Durch die klare und übersichtliche Darstellung der Aufgabenliste und der relevanten Informationen können Benutzer ihre Aufgaben schnell hinzufügen, bearbeiten und verwalten. Komplexe Menüs und überladene Funktionen wurden vermieden, um eine stressfreie Erfahrung bei der Aufgabenverwaltung zu gewährleisten.

Bei der Evaluation unseres Taskmanager-Apps haben wir verschiedene Testverfahren angewendet, um die Qualität, Leistung und Benutzerfreundlichkeit der Anwendung zu überprüfen. Durch umfangreiche Integrationstests konnten wir sicherstellen, dass alle Komponenten der App reibungslos zusammenarbeiten. Zudem haben wir Unittests durchgeführt, um die einzelnen Funktionen und Methoden in der App isoliert zu testen. Dabei konnten wir potenzielle Fehler identifizieren und beheben, um eine zuverlässige und stabile Funktionalität sicherzustellen.

Es ist jedoch wichtig, die Einschränkungen unseres Taskmanager-Apps zu beachten. Eine der Limitationen besteht in der Geräteabhängigkeit, da die App auf ein bestimmtes Gerät beschränkt ist und keine Cloud-Synchronisierung unterstützt. Zudem bietet die App derzeit keine integrierten Funktionen für die Zusammenarbeit oder das Teilen von Aufgaben mit anderen Benutzern. Diese Limitationen können die Flexibilität und Zusammenarbeit einschränken.

Für zukünftige Entwicklungen und Verbesserungen können wir verschiedene Aspekte in Betracht ziehen. Die Integration zusätzlicher Funktionen wie das Teilen von Aufgaben oder die Vereinfachung der Anbindung an Cloud-Speicherdienste könnte die Benutzererfahrung erweitern. Eine verbesserte Zusammenarbeit und die Möglichkeit, die App über verschiedene Geräte hinweg zu nutzen, könnten die Effizienz und Flexibilität der Aufgabenverwaltung weiter steigern.

Unser Taskmanager-App bietet jedoch auch herausragende Merkmale, die es von anderen Anwendungen abheben. Die intuitive Benutzeroberfläche, die einfache Handhabung und die effiziente Organisation der Aufgaben sind Aspekte, die unsere App zu einer angenehmen und stressfreien Lösung machen. Wir sind zuversichtlich, dass unsere App einen positiven Einfluss auf die Produktivität und Organisation der Benutzer haben wird.

Abschließend möchten wir betonen, dass wir mit unserem Taskmanager-App eine bedeutende Entwicklung in der individuellen Aufgabenverwaltung erreicht haben. Unsere intensive Arbeit und die umfassende Evaluation haben dazu beigetragen, eine praktische und benutzerfreundliche Lösung zu schaffen. Wir sind stolz auf das Ergebnis und hoffen, dass unsere App den Benutzern dabei hilft, ihre Aufgaben effektiv zu organisieren und ihre Ziele erfolgreich zu erreichen.