Pascal Affentranger & Nicolas Meier

TBZ | i-ch 355

Notenverwaltungstool

i-ch 355 - Studienwoche tbz



Inhaltsverzeichnis

[1. Projektbeschrieb 2](#_Toc454530084)

[2. Anforderungen 2](#_Toc454530085)

[2.1. Erreicht: 2](#_Toc454530086)

[3. Zusatzanforderungen 2](#_Toc454530087)

[3.1. Erreicht: 2](#_Toc454530088)

[3.2. Nicht Erreicht: 2](#_Toc454530089)

[4. Use-Cases 3](#_Toc454530090)

[4.1. Use-Case Diagramm 3](#_Toc454530091)

[4.2. Use-Case Beschrieb 4](#_Toc454530093)

[5. Gui-Skizze 5](#_Toc454530094)

[6. Klassenübersicht 6](#_Toc454530095)

[6.1. Ordnerübersicht 6](#_Toc454530096)

[6.2. Klassenbeschreibung 6](#_Toc454530097)

[7. ERD 7](#_Toc454530098)

[7.1. Semester: 7](#_Toc454530099)

[7.2. Subject: 7](#_Toc454530100)

[7.3. Grade: 7](#_Toc454530101)

[8. Testfälle 8](#_Toc454530102)

[9. Screenshots 9](#_Toc454530103)

[10. Zusammenarbeit 10](#_Toc454530105)

[11. Fazit 10](#_Toc454530106)

# Projektbeschrieb

Für das Modul 335, welches wir infolge der Studienwoche in Zürich an der TBZ besuchen, müssen wir eine Mobileapplikation auf Android erstellen und diese dokumentieren.

Unsere Idee war es, eine Applikation zu erstellen, welche die Schulnoten verwaltet, da wir selbst unsere Schulnoten monatlich weiterleiten müssen. Da wir jedoch noch keine digitale Lösung dafür haben, wollten wir dies in diesem Modul Realisieren, da wir die Noten nicht schriftlich erfassen und daher oftmals das Datum oder sogar die Note einer Prüfung vergessen. Wir haben uns das so vorgestellt, dass der Benutzer mehrere Semester erstellen und benennen kann. Zu jedem Semester können verschiedenste Fächer angelegt werden. Sobald ein Fach angelegt ist, kann man dem Fach Noten hinzufügen. Wird dies gemacht, so ist im Anschluss auf der Übersicht der Fächer der jeweilige Notenschnitt ersichtlich. Noten, Fächer sowie Semester können auch gelöscht werden, falls man dafür keine Verwendung mehr hat.

# Anforderungen

### Erreicht:

* In der App kann man auf dem Startscreen per Button ein Semester mit gewünschtem Titel hinzufügen, welches anschliessend dort angezeigt wird.
* Für jedes angelegte Semester können in einem weiteren Fenster mehrere Schulfächer mit dem gewünschten Titel gespeichert werden.
* Für jedes angelegte Schulfach können in einem weiteren Fenster mehrere Noten mit dem Datum der Prüfung und dem Notenwert hinzugefügt werden.
* Jedes Semester kann gelöscht werden. Dabei werden die dazugehörigen Fächer und Noten ebenfalls gelöscht.
* Jedes Fach kann gelöscht werden. Dabei werden die dazugehörigen Noten ebenfalls gelöscht.
* Jede Note eines Fach kann gelöscht werden. Anschliessend wird der Notenschnitt neu berechnet.

# Zusatzanforderungen

### Erreicht:

* Die App enthält auf jedem Screen einen eigenen Return-Button, welcher den User auf den vorherigen Screen zurückschickt.
* Falls der User ein Semester, ein Fach, oder eine Note löschen möchte, so öffnet sich ein Pop-Up, welches eine Bestätigung des Users verlangt, ob er den Datensatz wirklich löschen möchte.
* Um das Datum der Note einzugeben wird nicht ein Eingabefeld, sondern ein Kalender angezeigt, wo man das Datum bequem auswählen kann.

### Nicht Erreicht:

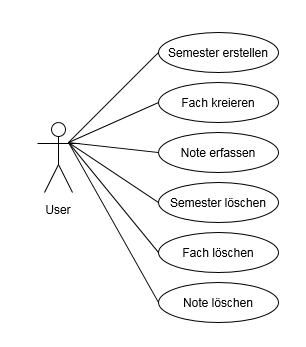
* Semester, Fächer und Noten können bearbeitet werden ohne sie zu löschen.
* Der Note kann eine Bemerkung hinzugefügt werden.

# Use-Cases

### Use-Case Diagramm

# 

Notenverwaltung



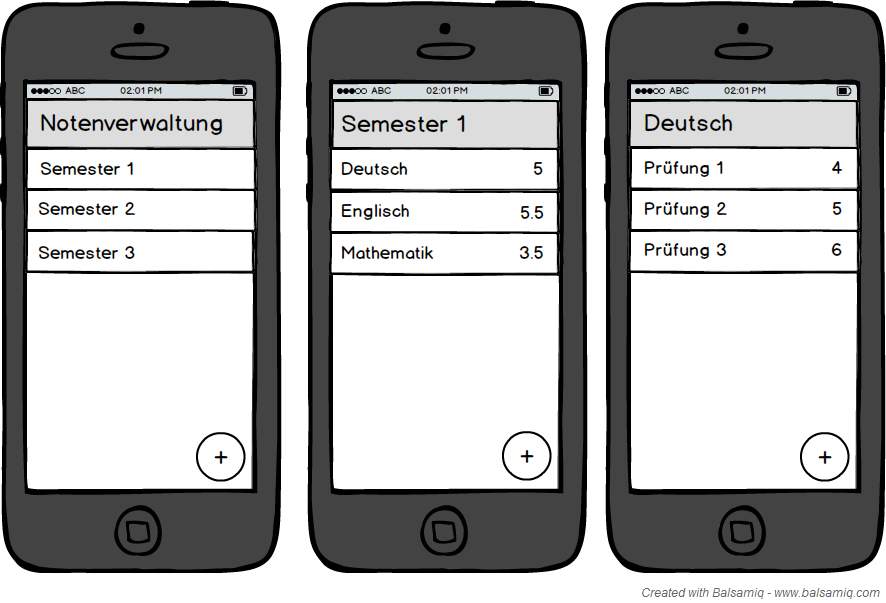
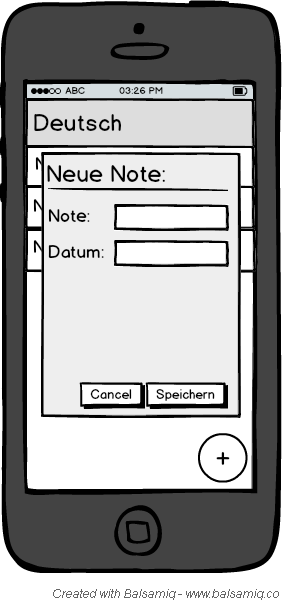
### Use-Case Beschrieb



# Gui-Skizze

Dies sind die drei Hauptoberflächen der Applikation. Die erste Grafik ist das Hauptmenü. Falls nun auf ein Semester geklickt wird, kann man über das zweite Fenster die einzelnen Fächer verwalten. Wählt man nun ein Fach an, so werden in einem Neuen Menü die einzelnen Prüfungen aufgelistet. In jedem Menü besteht die Möglichkeit, einzelne Einträge zu löschen.

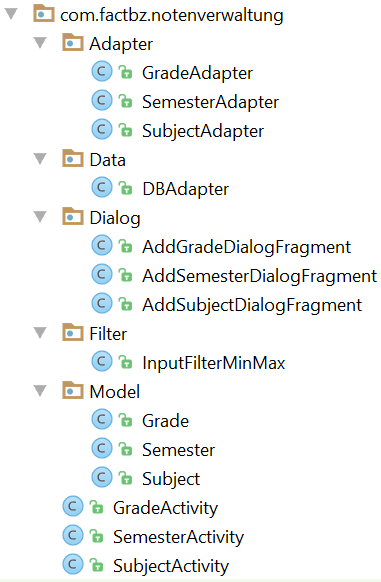
Sobald der + Button betätigt wird, um ein Semester, ein Fach oder eine Note anzulegen, so wird sich ein neues Fenster öffnen, wo man anschliessend die nötigen Informationen mitgeben kann. Bestätigt man nun die Eingaben, so werden die Daten in der Datenbank gespeichert.



# Klassenübersicht

* 1. Ordnerübersicht:

Da wir sehr viele Klassen haben, haben wir sie in folgende Ordner aufgeteilt

* **Adapter**: Alle Klassen die Daten formatieren
* **Data**: Klassen die Daten speichern
* **Dialog**: Alle DialogFragment Klassen
* **Filter**: Alle Input Filter
* **Model**: Alle Model
* **Activity**: Alle Activity (Controller) Klassen
  1. Klassenbeschreibung:
* **GradeAdapter**:

Stellt eine Note dar. Formatiert das Datum und die Note.

* **SemesterAdapter**:

Stellt ein Semester dar.

* **SubjectAdapter**:

Stellt ein Fach dar. Berechnet den Durchschnitt aller Noten.

* **DBAdapter**:

Erstellt die Datenbank und führt SQL aus

* **AddGradeDialogFragment**:

Erstellt einen AlertDialog. Beim Klick auf das Datumsfeld kommt ein Datepicker. Die Note kann nur einen Wert zwischen 1 und 6 haben (Siehe -> InputFilterMinMax).

* **AddSemesterDialogFragment**:

Erstellt einen AlertDialog. Es wird keine spezielle Überprüfung ausgeführt.

* **AddSubjectDialogFragment**:

Erstellt einen AlertDialog. Es wird keine spezielle Überprüfung ausgeführt.

* **InputFilterMinMax**:

Filter der Zwei Floats, Min, Max, als Input hat. Überprüft den Input auf seine Gültigkeit.

* **Grade**:

Model für Note mit Datum, Note und Id.

* **Semester**:

Model für Semester mit Name und Id.

* **Subject**:

Model für Subject mit Name und Id.

* **GradeActivity**:

Controller für Noten Anzeige implementiert alle Gui funktionen.

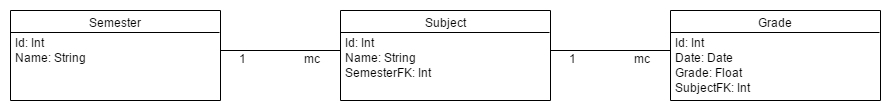
* **SemesterActivity**:

Controller für Semester Anzeige implementiert alle Gui funktionen.

* **SubjectActivity**:

Controller für Fächer Anzeige implementiert alle Gui funktionen.

# ERD



### Semester:

Diese Tabelle wird dafür verwendet, um ein Semester zu speichern. Der User muss den Titel des Semesters angeben, welcher dann auch in der Übersicht angezeigt wird. Ein Semester hat keine oder mehrere Fächer.

* **id:** Primärschlüssel zur eindeutigen Identifikation.
* **Name:** Der Semestertitel, welcher vom User gewählt werden kann.

### Subject:

Die Subject Tabelle wird dafür verwendet, um die Fächer eines Semesters zu erstellen. Der User muss den Titel des Schulfachs angeben, welcher dann auch in der Übersicht angezeigt wird. Jedes Fach gehört zu genau einem Semester. Jedes Fach hat keine oder mehrere Noten.

* **id:** Primärschlüssel zur eindeutigen Identifikation.
* **Name:** Der Titel des Schulfachs. Kann vom User gewählt werden.
* **SemesterFK:** Fremdschlüssel des Semester-Tables.

### Grade:

Die Grade Tabelle wird dafür verwendet, um die Noten eines Fachs zu speichern. Der User muss das Datum der Prüfung oder der Note angeben, sowie die Note selbst. Die Note wird anschliessend dafür verwendet, um in der Schulfächerübersicht den Durchschnitt pro Fach zu berechnen. Jede Note gehört genau zu einem Schulfach.

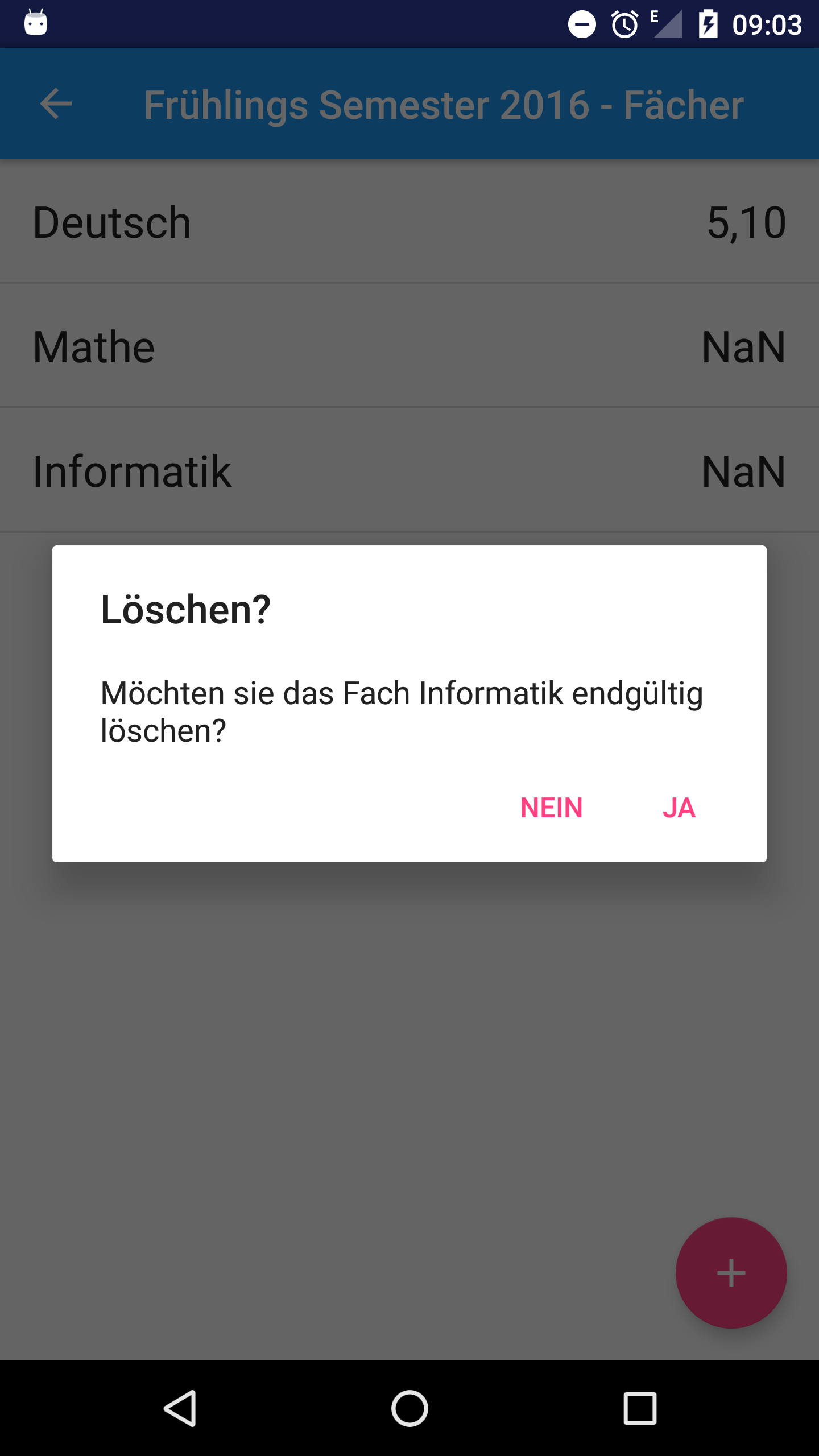
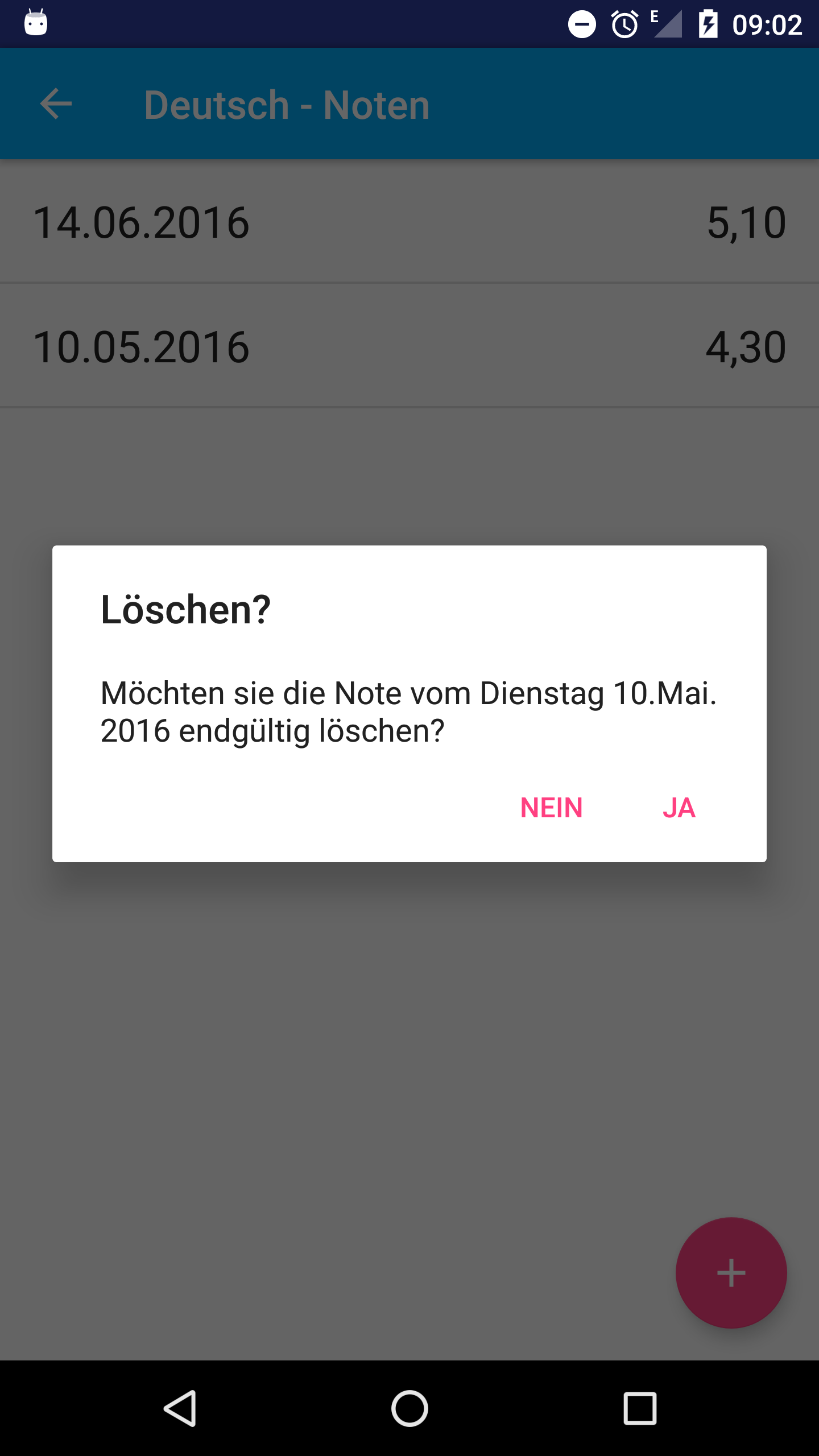
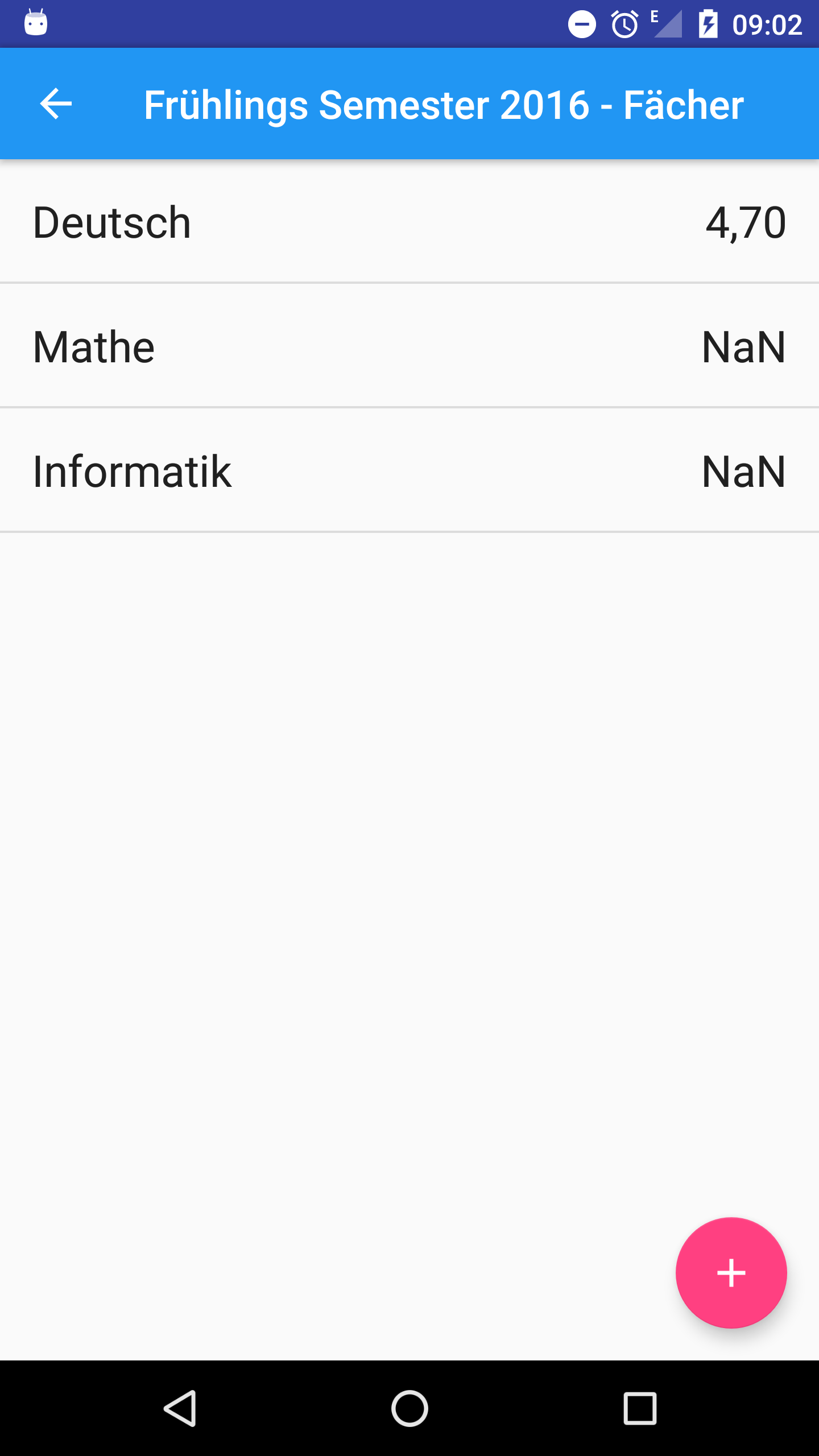
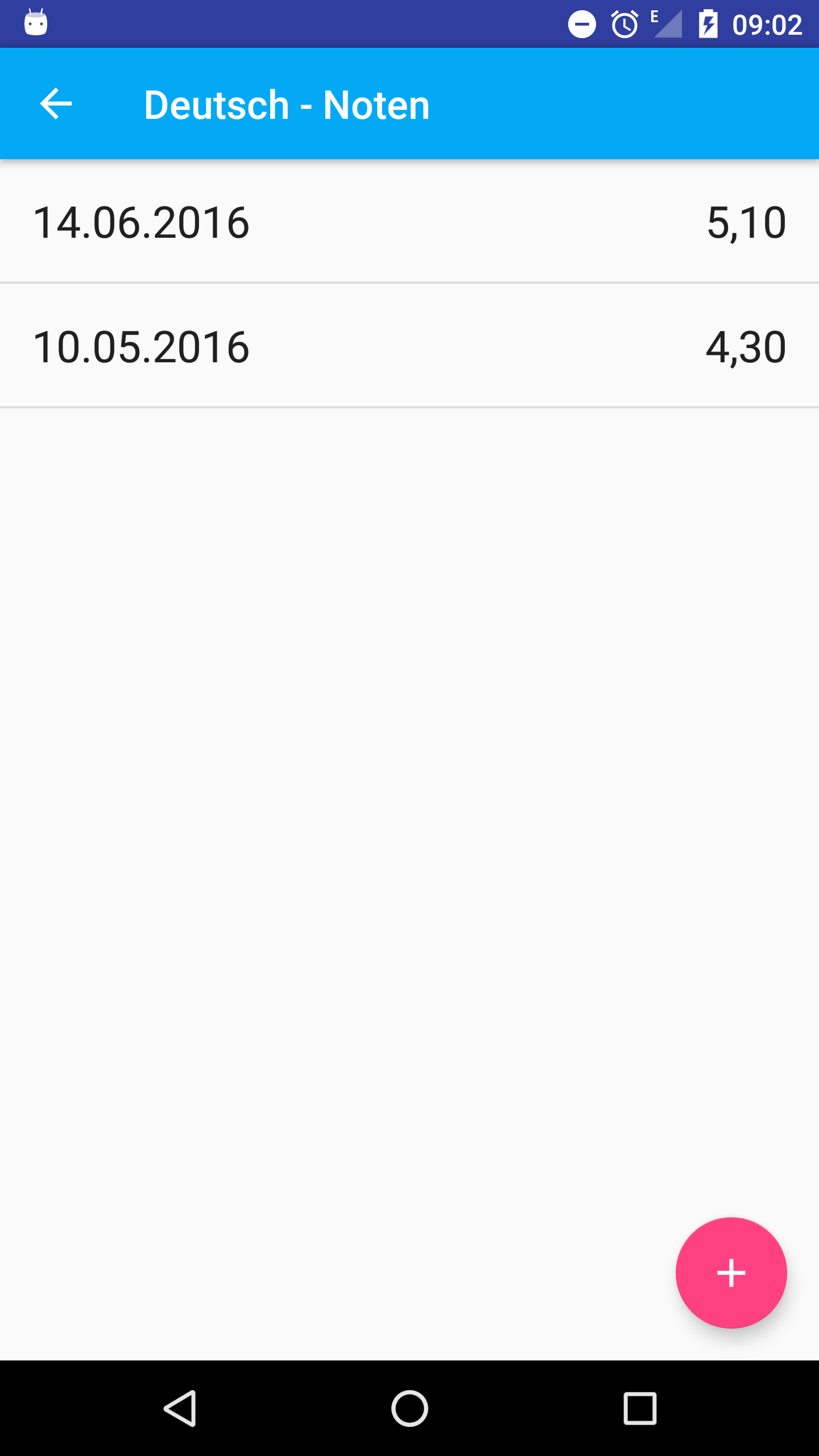
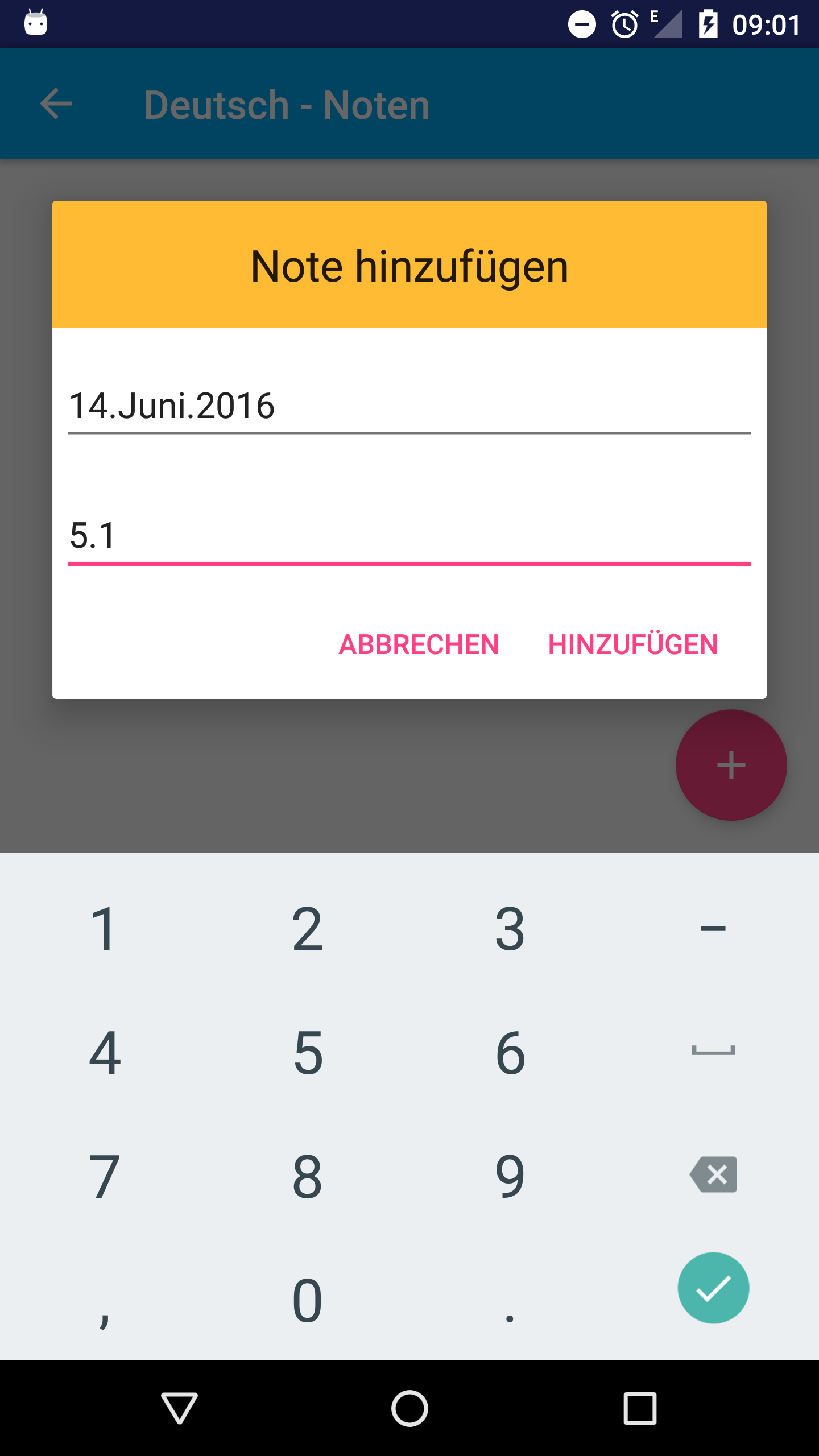
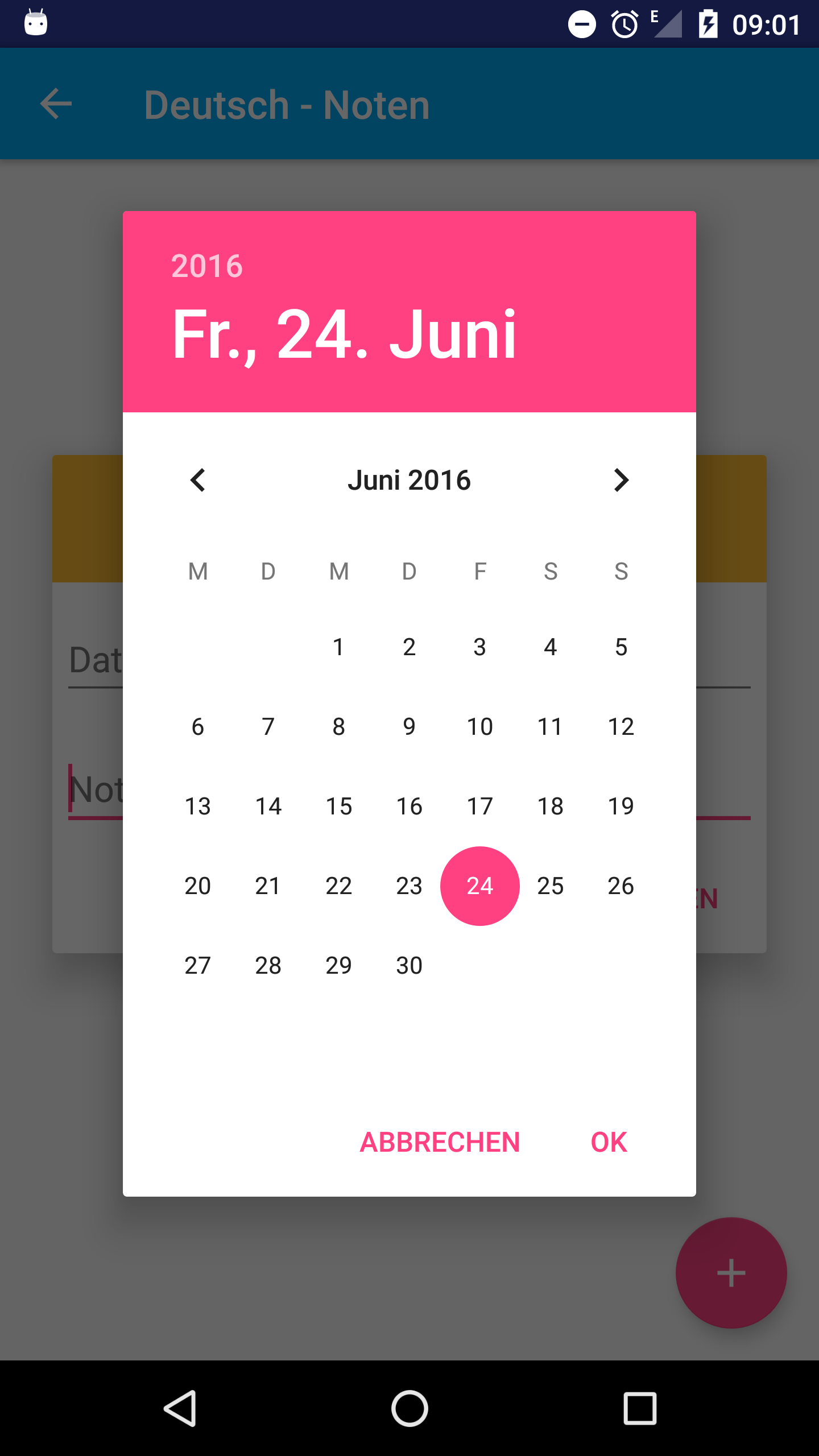
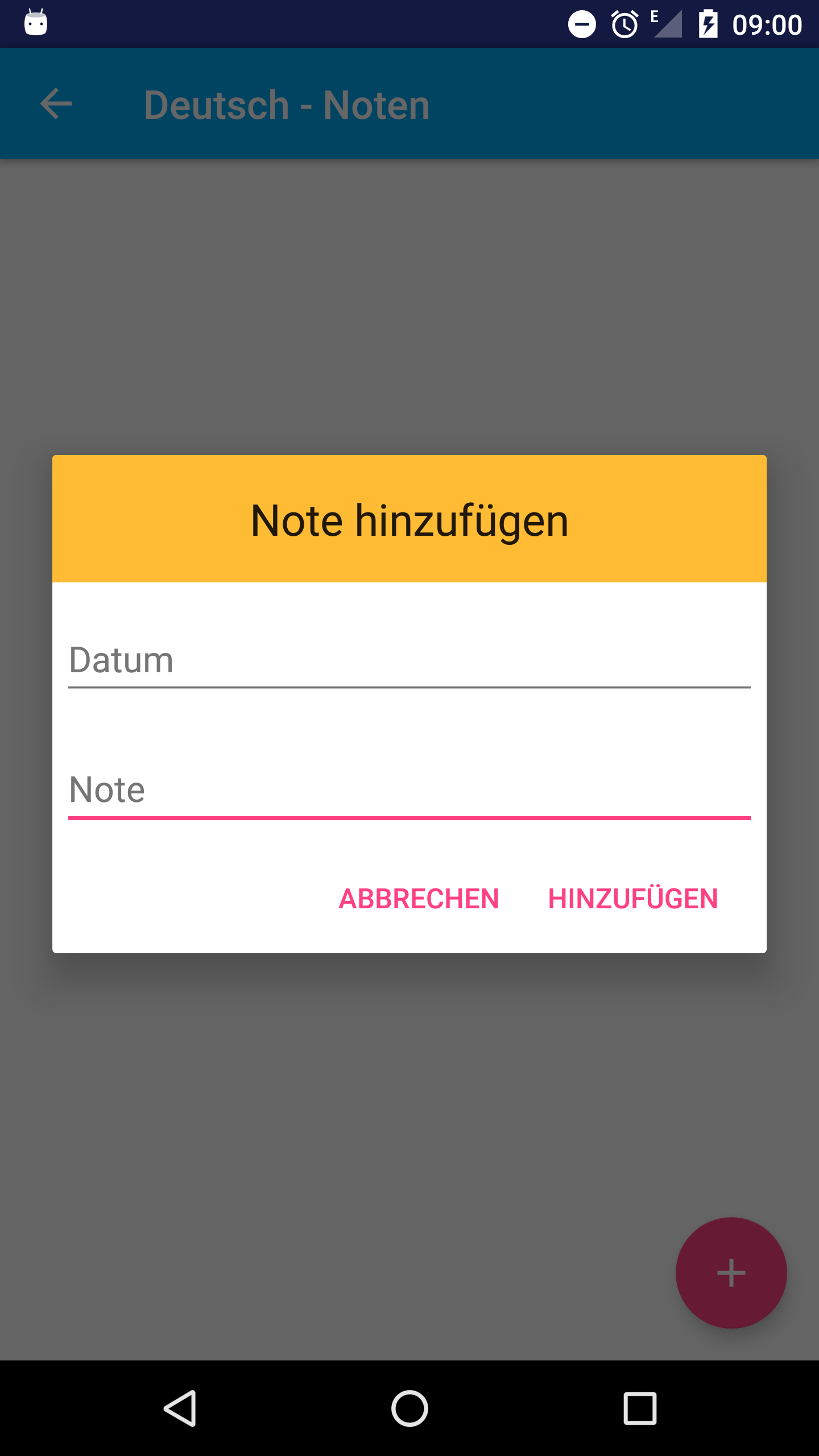
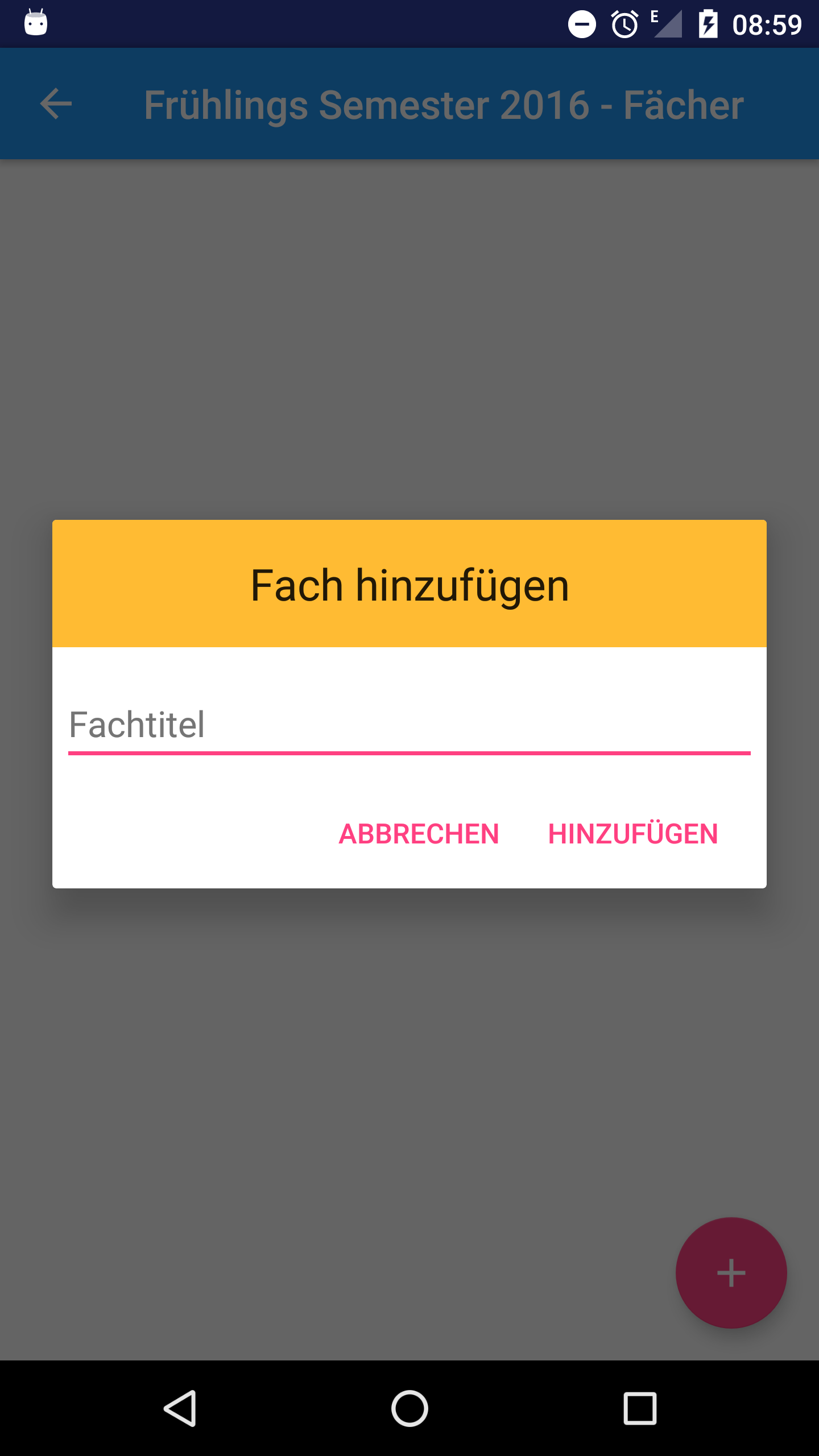
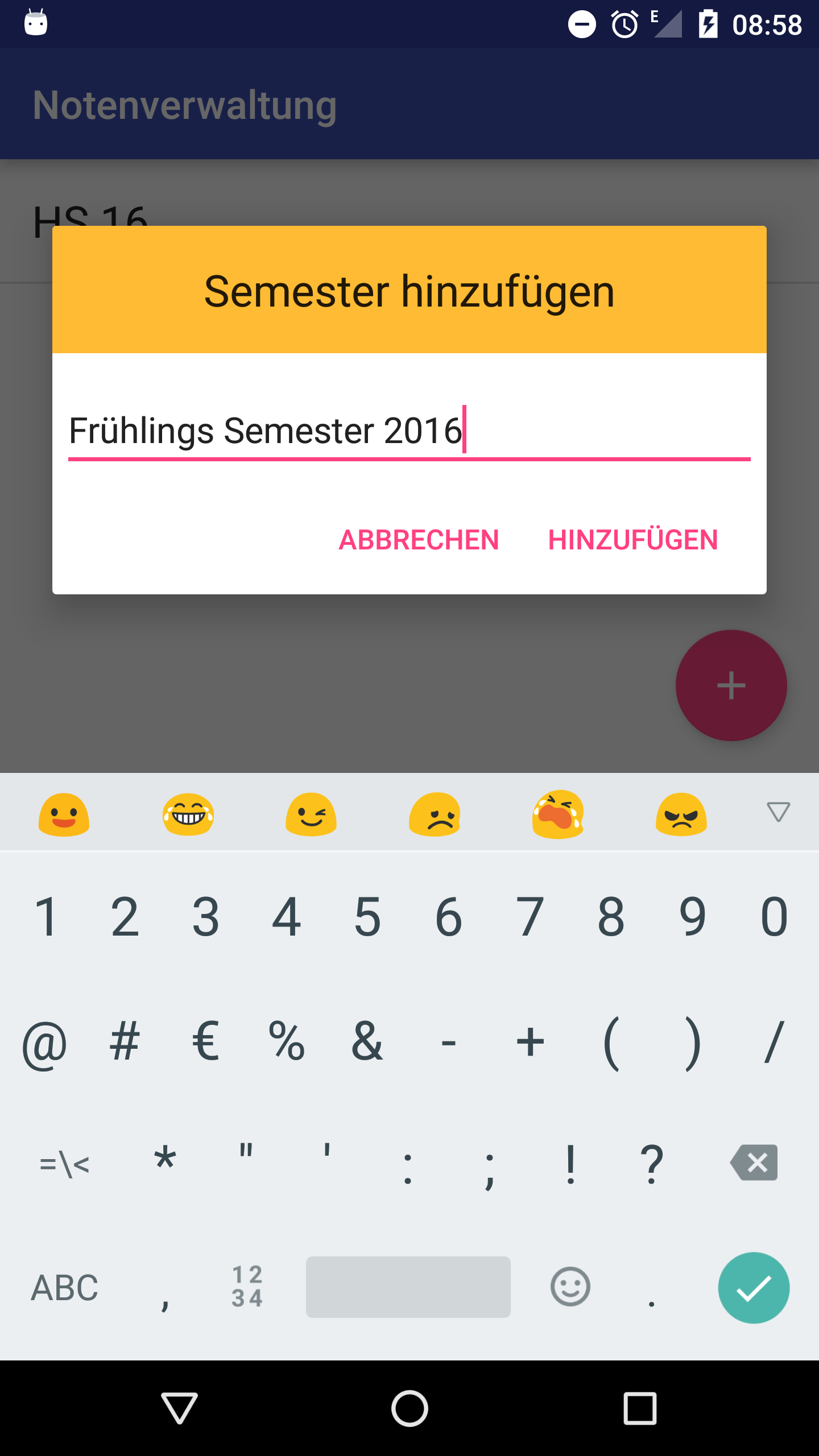
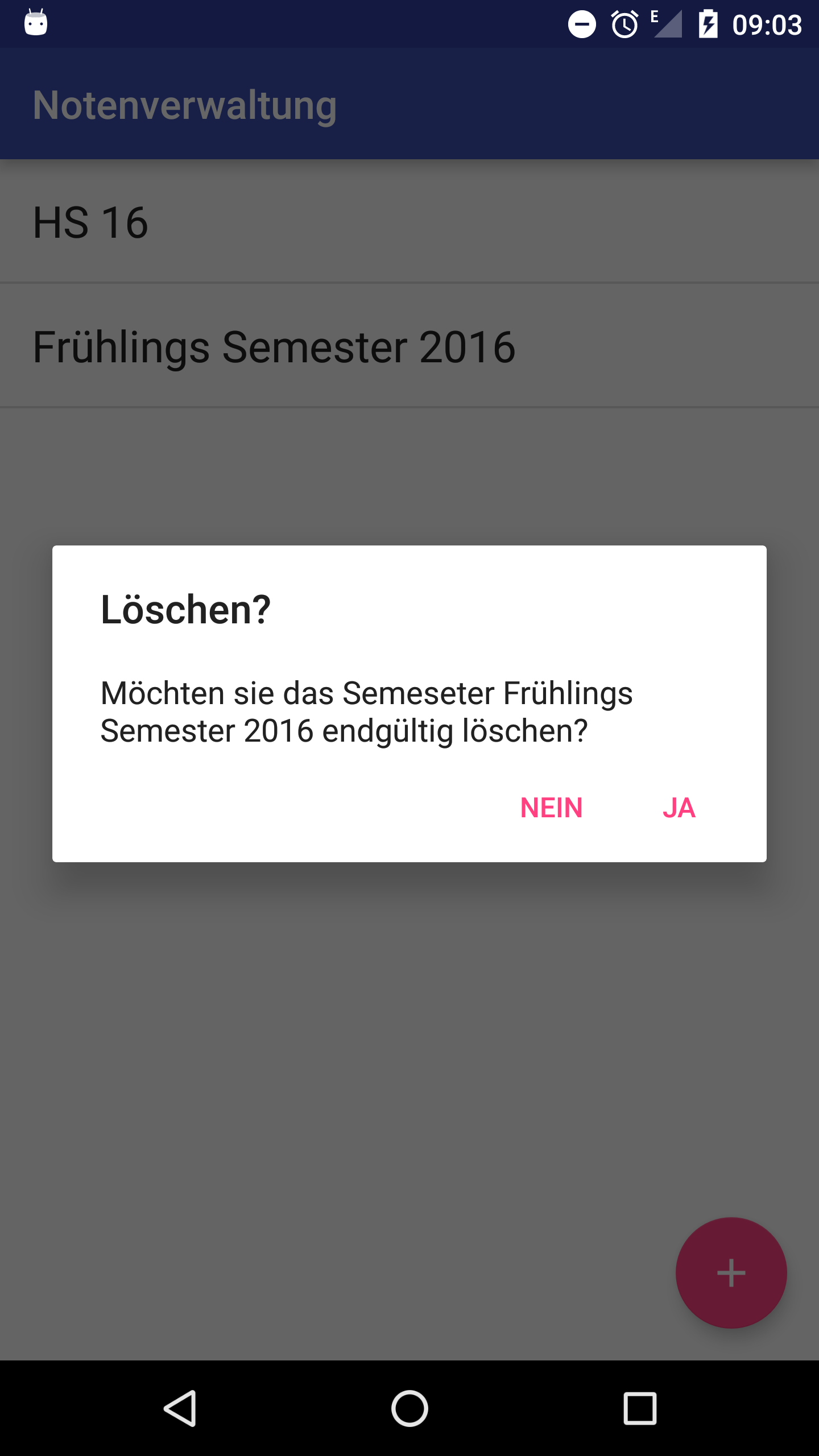
* **id:** Primärschlüssel zur eindeutigen Identifikation.
* **Date:** Datum der Note/der Prüfung. Dient als Anzeigename.
* **Grade:** Die erhaltene Note. Wird gebraucht um den Durchschnitt auszurechnen.
* **SubjectFK:** Fremdschlüssel des Subject-Tables

# Testfälle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nr. | Activity | Eingabe | Erwartetes Ergebnis | Tatsächliches Ergebnis | Resultat |
| 1 | Semester | Klick auf + Button | Dialog erscheint | Dialog erscheint | Bestanden |
| 2 | Semester | Dialog ausgefüllt und gesendet. | Daten in Datenbank gespeichert | Daten in Datenbank gespeichert | Bestanden |
| 2 | Semester | Langes Drücken auf Semester | Löschdialog erscheint | Löschdialog erscheint | Bestanden |
| 3 | Semester | Bestätigen des Löschvorgangs | Daten werden rekursiv gelöscht | Daten werden rekursiv gelöscht | Bestanden |
| 4 | Semester | Klick auf ein Semester | Neues Fenster mit Fächern des Semesters erscheint | Neues Fenster mit Fächern des Semesters erscheint | Bestanden |
| 5 | Subject | Klick auf + Button | Dialog erscheint | Dialog erscheint | Bestanden |
| 6 | Subject | Dialog ausgefüllt und gesendet. | Daten in Datenbank gespeichert | Daten in Datenbank gespeichert | Bestanden |
| 7 | Subject | Langes Drücken auf Fach | Löschdialog erscheint | Löschdialog erscheint | Bestanden |
| 8 | Subject | Bestätigen des Löschvorgangs | Daten werden rekursiv gelöscht | Daten werden rekursiv gelöscht | Bestanden |
| 9 | Subject | Klick auf ein Fach | Neues Fenster mit Noten des Fachs erscheint | Neues Fenster mit Noten des Fachs erscheint | Bestanden |
| 10 | Subject | Klick auf Return Button | Man gelangt zurück zur Semesterübersicht | Man gelangt zurück zur Semesterübersicht | Bestanden |
| 11 | Grade | Klick auf + Button | Dialog erscheint | Dialog erscheint | Bestanden |
| 12 | Grade | Dialog ausgefüllt und gesendet. | Daten in Datenbank gespeichert | Daten in Datenbank gespeichert | Bestanden |
| 13 | Grade | Langes Drücken auf Fach | Löschdialog erscheint | Löschdialog erscheint | Bestanden |
| 14 | Grade | Bestätigen des Löschvorgangs | Daten werden gelöscht | Daten werden gelöscht | Bestanden |
| 15 | Grade | Klick auf Datumsfeld | Es erscheint ein Kalender zur einfachen Datumseingabe | Es erscheint ein Kalender zur einfachen Datumseingabe | Bestanden |
| 16 | Grade | Klick auf Return Button | Man gelangt zurück zur Fachübersicht | Man gelangt zurück zur Fachübersicht | Bestanden |

# Screenshots

# 



# Zusammenarbeit

Wir haben mit GitHub gearbeitet, was für uns sehr viele Vorteile hatte. Vor allem das Mergen war sehr Hilfreich. Wir haben die Arbeit an der App wie folgt Aufgeteilt (via GitHub kann jede Zeile einer Person zugeordnet werden):

* **Nicolas Meier**: ERD, Design (Storyboard, Views), Programmieren (Alle Adapter, Alle Filter, Alle Models, Semester Dialog und der Haupteil der Activities)
* **Pascal Affentranger**: Use Case, Dokumentation, Datenbank, Programieren (Datenbankanbindung, Subject & Grade Dialog und ein Teil der Activities)

# Fazit

Wir konnten alle Grundanforderungen erfüllen, leider konnten wir nicht alle Zusatzanforderungen umsetzen. Bei der Entwicklung gab es keine grossen Probleme. Wir haben während dem Programmieren viel Neues gelernt. Die App ist in einem Fertigen zustand, sodass wir sie im Play Store veröffentlichen könnten.

Projekt auf GitHub: <https://github.com/factbz/Notenverwaltung>