



Universität Regensburg

Philosophische Fakultät III
Sprach- , Literatur- und Kulturwissenschaften
Institut für Information und Medien, Sprache und Kultur (I:IMSK)
Lehrstuhl für Medieninformatik

Einführung in die Anwendungsprogrammierung
Modul: MEI-M03.3
Sommersemester 2017

Travelnote

Dein Reisetagebuch.

Andreas Obermeier
1751433
Informationswissenschaft, Medieninformatik
3. Semester B.A.

Dr. Gräßmann-Straße 8
92249 Vilseck
Tel.: 0176 91355967
Email: andreas.obermeier@stud.uni-regensburg.de

Michael Wexler
1680736
Informationswissenschaft, Medieninformatik + FKN
8. Semester B.A.

Hubertusstraße 15
92369 Sengenthal
Tel.: 0172 8249869
Email: michael.wexler@stud.uni-regensburg.de

Abgegeben am 23.09.2017

Inhalt

1	Kurzbeschreibung	5
2	Technische Voraussetzungen	6
3	Problemstellung und Lösungsvorschlag	7
4	Design & Implementierung	8
4.1	User Interface	9
4.2	Implementierung	15
5	Testing	18
6	Finaler Zustand und Ausblick	19
6.1	Finaler Zustand	19
6.2	Sinnvolle Erweiterungen	19
7	Projektmanagement	21

Abbildungen

Abbildung 1: Loginbildschirm der Anwendung	9
Abbildung 2: Overview-Bildschirm	10
Abbildung 3: Neuen Tagebucheintrag erstellen	10
Abbildung 4: Tagebuch Übersicht.....	11
Abbildung 5: Tagebuch Menü	11
Abbildung 6: Tagebuch Lösch-Dialog	12
Abbildung 7: Profil	12
Abbildung 8: Gespeicherte Standorte.....	13
Abbildung 9: Karte	13
Abbildung 10: Karte mit Standpunkt	14
Abbildung 11: Neues Galeriebild	14
Abbildung 12: Bildergalerie	15

1 Kurzbeschreibung

Gerade junge Menschen und Studenten sind in ihrer freien Zeit viel auf Reisen und erleben dort viel. Mit der App *Travelnote* vergessen Sie keine wichtigen Erlebnisse mehr. *Travelnote* ist eine Android-App mehr der Sie einfach und intuitiv Ihre Erlebnisse festhalten und mit andern teilen können, auf einer Karte können Sie zudem erkennen, wo auf der Welt Sie schon überall waren. Nach einem einfachen Login per E-Mail und Passwort oder über eine bereits von Ihnen genutzte App, können Sie Ihre Erlebnisse in Form eines Tagebucheintrages festhalten, eine Fotogalerie erstellen und Orte auf einer Karte markieren. Mit *Travelnote* können sie einfach ihre Erlebnisse versenden, ihre persönliche Reisekarte betrachten und in Erinnerungen schwelgen.

Grundlegende Features von *Travelnote*:

- Persönliches Benutzerprofil (mit Facebook-Login)
- Tagebuch erstellen & per E-Mail versenden
- Orte auf der Karte markieren und so eine Sammlung erstellen
- Erstellen eines persönlichen Fotoalbums

Travelnote stellt dem Nutzer eine Urlaubs-App zur Verfügung, welche alle grundsätzlichen Funktionen erfüllt. Der Nutzer benötigt dann nicht mehr verschiedene Applikationen, um seinen Anforderungen gerecht zu werden, sondern nur noch *Travelnote*.

2 Technische Voraussetzungen

Travelnote ist eine reine Mobil-Anwendung, die ohne eigene Server-Infrastruktur funktioniert. Für den Login wird allerdings eine Internetverbindung benötigt. Der Login erfolgt über eine Firebase-Datenbank und kann mit Facebook verknüpft werden. Für den Betrieb der Anwendung reicht eine manuelle Installation der Anwendung auf einem Android-Gerät aus. Um Tagebuch-Einträge per E-Mail versenden zu können, muss auf dem Gerät eine E-Mail Applikation eingerichtet werden. Dies kann auch beim erstmaligen Durchführen der Funktion gemacht werden. *Travelnote* kann auch auf Tablets verwendet werden. Die Geräte benötigen mindestens die Version 4.0 (IceCreamSandwich). Programmiert wurde die Applikation auf der neusten Version 8.0 (Oreo).

3 Problemstellung und Lösungsvorschlag

Wenn man im Urlaub seine Erlebnisse festhalten will, dann benötigt man dafür viele verschiedene Apps. Man braucht eine Tagbuch-App, um seine Erlebnisse zu dokumentieren, man muss sich ein Album erstellen, wenn man nur die Urlaubsfotos haben will. Dieses ist dann aber immer in der Gerätegalerie gespeichert. Man benötigt eine Map-Applikation, um seine Standorte zu markieren. Und wahrscheinlich muss man sich überall noch separat anmelden. Um dies zu erleichtern versuchen wir mit *Travelnote* eine Android-App zu bauen, die all diese Aufgaben erfüllt. Die App soll einen einfachen persönlichen Login besitzen. Die Grundlegenden Funktionen der App fassen die oben genannten Aufgaben zusammen. Es kann ein Tagebuch erstellt werden mit Ort & Datum. Dieses kann auch per E-Mail versendet werden. Des Weiteren kann der Nutzer auf einer Karte seinen Standort/Orte markieren. Diese werden ebenfalls gespeichert und können auch gesammelt betrachtet werden. Zudem kann man aus der Gerätegalerie Fotos zu einem Album zusammenfügen, welches ebenfalls in der App gespeichert wird. Die App ist perfekt, um nach einem anstrengenden oder erholsamen Urlaubstag abends im Hotelzimmer die gewonnenen Eindrücke zu sammeln.

4 Design & Implementierung

Die Anwendung beinhaltet insgesamt fünf Activities und 5 Fragments. Bei erster Öffnung wird die `MainActivity` geöffnet. Diese dient zum Login. Die Activity überprüft, ob ein Nutzer bereits eingeloggt ist. Falls nicht, kann er sich per E-Mail oder/und per Facebook einloggen. Falls er schon eingeloggt ist, wird er sofort zur `StartActivity` geleitet. Diese eine Menüleiste, welche die verschiedenen Fragments aufruft. Bei Start ist immer das `OverviewFragment` aktiv. In diesem Fragment werden drei Buttons dargestellt. Diese führen zu `NewDiaryEntryActivity`, `NewImageEntryActivity` und zum `MapFragment`. In ersterer kann ein neuer Tagebucheintrag erstellt werden, in zweiterer ein Foto zum Album hinzugefügt werden und in letztere wird die Karte gezeigt und es können Orte markiert werden. Benutzt man in der Menüleiste der `StartActivity` die Icons, dann wird das aktuelle Fragment ersetzt. Drückt man den „Buch-Icon“, dann erscheint das `DiaryFragment`. Hier werden erstellte Tagebucheinträge angezeigt. Man erhält durch eine Optionenleiste die Möglichkeit das gesamte Tagebuch zu löschen oder per E-Mail zu versenden. Weiter kann man, wenn man kurz auf einen einzelnen Eintrag drückt, diesen einzelnen Eintrag per Mail versenden. Drückt man länger, kann man den einzelnen Eintrag löschen. In den letzteren drei Varianten muss man die Aktion vorher in einem `AlertDialog` bestätigen. Das Fragment enthält zudem einen „Neuen Eintrag“-Button. Durch ihn gelangt man zur `NewDiaryEntryActivity`.

Betätigt man in der Menüleiste den „Karten-Icon“ wird das aktuelle Fragment durch das `MapFragment` ersetzt. Dort wird eine Google-Map gezeigt. Man kann hier seinen eigenen Standort über zwei Möglichkeiten markieren. Einmal über die Optionen und, wenn man auf den Marker klickt. Zudem kann man hier einen Ort eingeben und diesen dann per Button-Klick markieren. In den Optionen des Fragments kann man zudem noch alle seine bisherigen Standorte auf der Karte markieren & seine Standorte löschen.

Drückt man in der Menüleiste das „Profil-Icon“, wird das aktuelle Fragment durch das `SettingsFragment` ersetzt. Hier kann man sich ausloggen. In diesem Fall wird man wieder zur `MainActivity` geführt, man muss sich neu einloggen. Ebenfalls kann man im Profil eine Liste seiner bisherigen Standorte betrachten und einzelne per Druck(Klick) löschen.

Drückt man in der Menüleiste den „Galerie-Icon“, wird das aktuelle Fragment durch das `GalleryFragment` ersetzt. Hier werden die gespeicherten Bilder mit Titel und Ort angezeigt. Zudem gibt es in den Optionen des Fragments die Möglichkeit das Album zu löschen.

Die Anwendung beinhaltet zudem eine Klasse `DatabaseHelper`. Hier wird eine SQL-Datenbank initiiert. Sie beinhaltet zwei Tabellen, jeweils für das Tagebuch und die Karte. Des Weiteren gibt es drei Adapter-Klassen, `DiaryCursorAdapter`, `GalleryBaseAdapter` und `LocationCursorAdapter`. Diese dienen zur Darstellung der in den Datenbanken gespeicherten Daten. Zusätzlich gibt es noch die `ImageDBHelper`-Klasse, eine extra Datenbank, welche ein Objekt der `CustomImage`-Klasse speichert.

Da die Online-Datenbank noch nicht funktioniert, kann der Nutzer sich zwar auf mehreren Geräten anmelden, seine Datenbank wird aber nicht übertragen.

Sollte im Verlauf der Nutzung die Internetverbindung abbrechen wird ein Disconnect-Bildschirm eingeblendet. Man kann die App dann nicht weiter nutzen.

4.1 User Interface

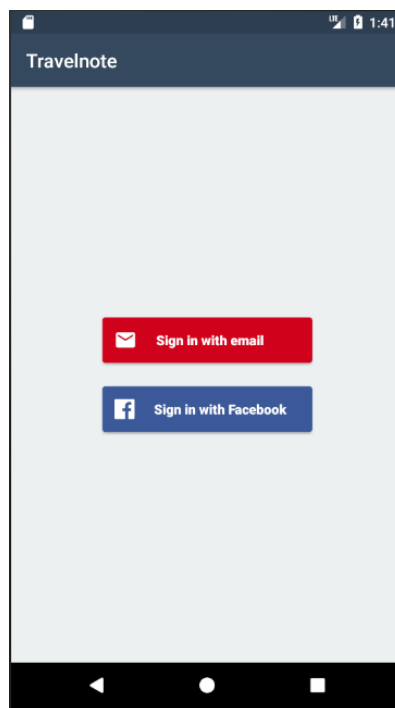


Abbildung 1: Loginbildschirm der Anwendung

Hier kann sich der Nutzer mit den zwei Möglichkeiten anmelden.



Abbildung 2: Overview-Bildschirm

Hier wird in der `StartActivity` das `OverviewFragment` angezeigt. über die Schaltflächen – die den Großteil des Bildschirms einnehmen – gelangt der Nutzer zu den jeweiligen Activities. Im unteren Bereich ist die Menüleiste sichtbar (v.l.n.r.: Home, Tagebuch, Galerie, Karte, Profil). Drückt man den „Neuer Tagebucheintrag“-Button gelangt man zu folgendem Screen:

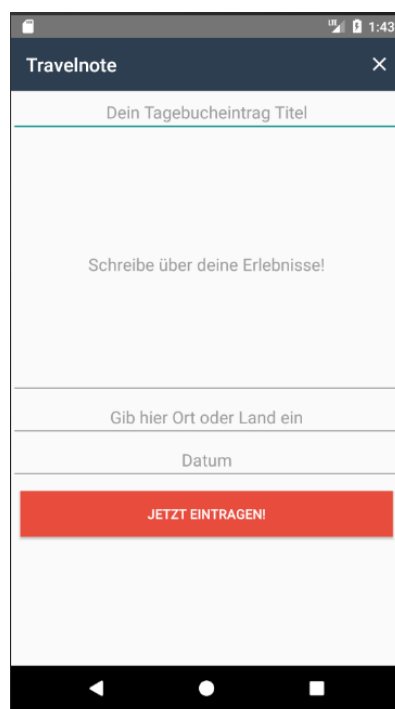


Abbildung 3: Neuen Tagebucheintrag erstellen

Hier kann man für den Eintrag Titel, Text, Ort und Datum (über einen DatePicker) eingeben. Mit „Jetzt Eintragen!“ fügt man den Eintrag zur Datenbank hinzu.

Betätigt man in Abbildung 2 in der Menüleiste das Tagebuch-Icon, öffnet sich das `DiaryFragment`.

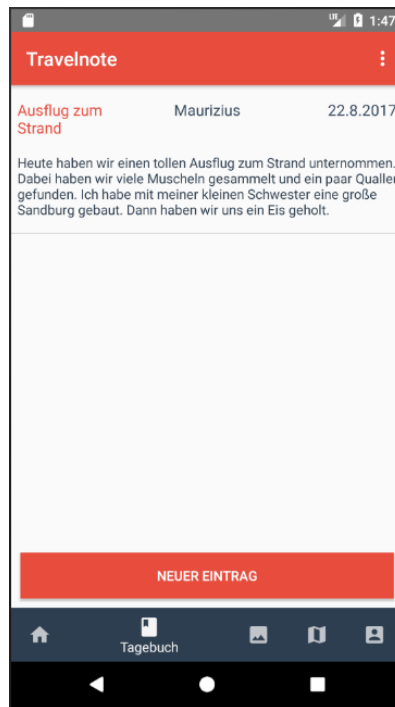


Abbildung 4: Tagebuch Übersicht

Mit Menü

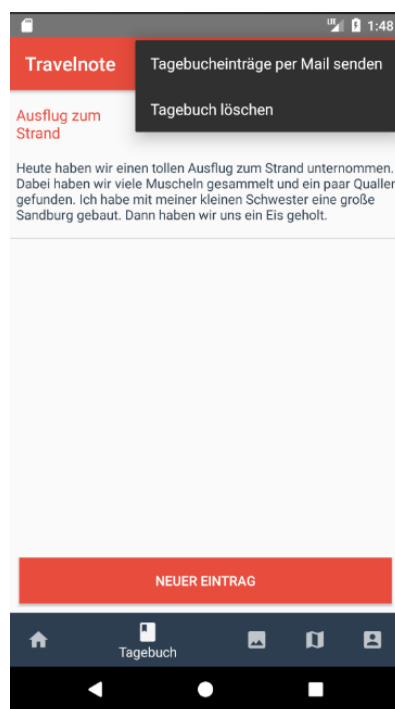


Abbildung 5: Tagebuch Menü

und Lösch-Dialog:

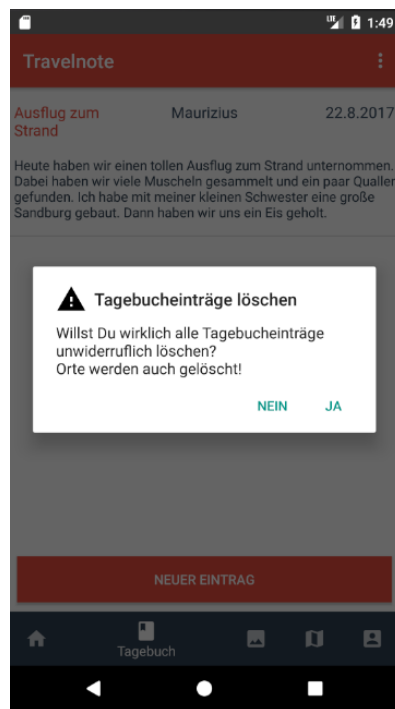


Abbildung 6: Tagebuch Lösch-Dialog

Betätigt man in der Menüleiste das Profil-Icon gelangt man zum Profil:

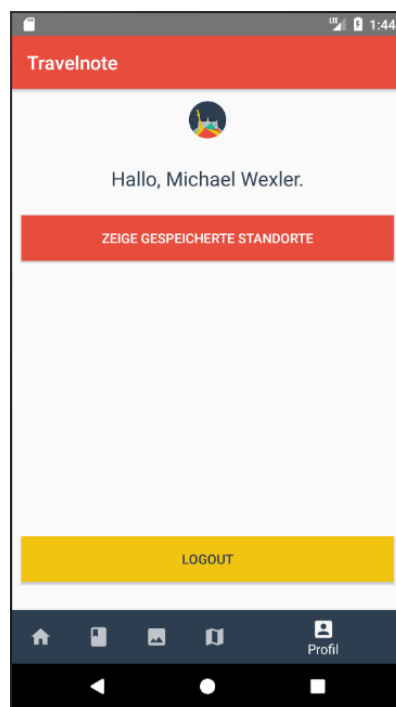


Abbildung 7: Profil

Hier kann er sich ausloggen oder seine gespeicherten Standorte betrachten.

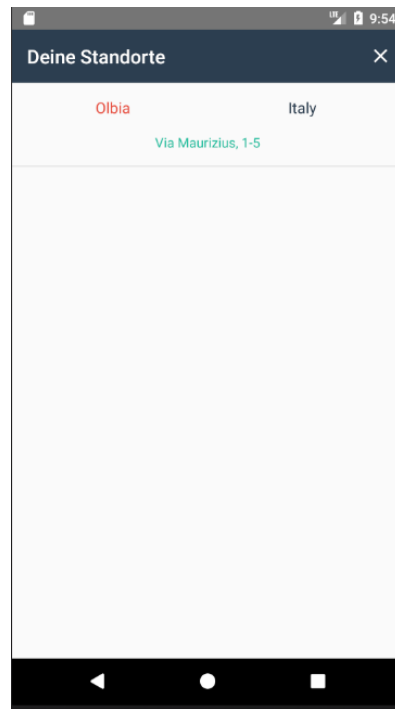


Abbildung 8: Gespeicherte Standorte

Drückt man im Menü das „Karte“-Icon oder im OverviewFragment den „Zur Karte“-Button, wird das aktuelle Fragment durch das MapFragment ersetzt:

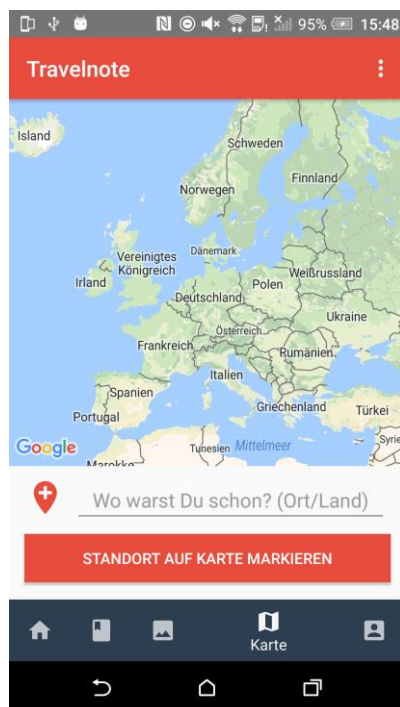


Abbildung 9: Karte

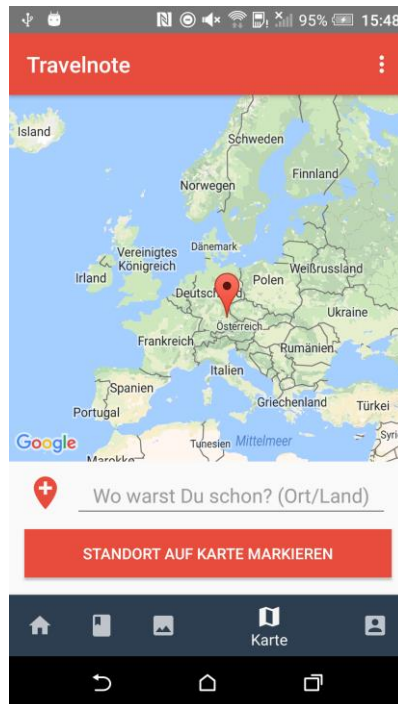


Abbildung 10: Karte mit Standpunkt

Hier kann der Nutzer seinen aktuellen Standort markieren, alle Nutzer-Standorte anzeigen (funktioniert noch nicht) und eigene Standorte löschen.

Drückt man auf dem `OverviewFragment` den „Neue Galerie“-Button öffnet sich die `NewImageEntryActivity`. Hier kann der Nutzer ein neues Bild mit Titel und Ort zur Galerie hinzufügen.

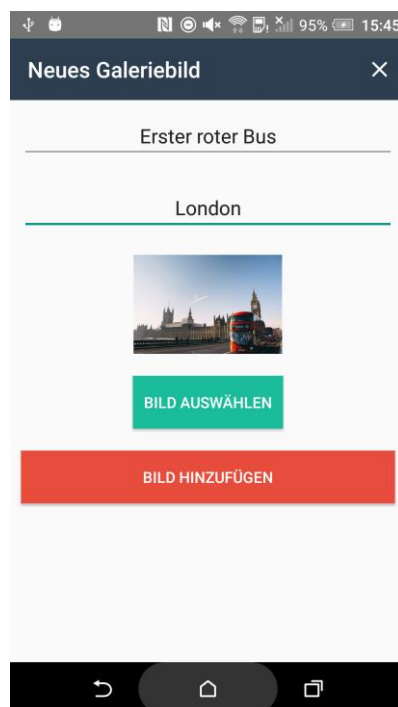


Abbildung 11: Neues Galeriebild

Wenn man im Menü das „Foto“-Icon anklickt, wird das aktuelle Fragment durch das `GalleryFragment` ersetzt. Hier kann der Nutzer seine Bilder betrachten

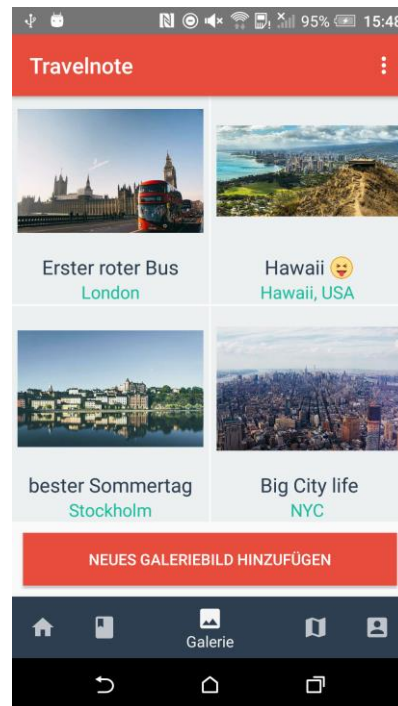


Abbildung 12: Bildergalerie

4.2 Implementierung

Die wichtigste Aufgabe in der Applikation ist es die vom Nutzer eingegeben oder ausgewählten Daten zu speichern. Daher ist die Verwendung einer Datenbank essentiell. Deshalb ist die `DatabaseHelper`-Klasse die wichtigste Klasse der Applikation. Sie wird von fast allen Activities und Fragmenten benutzt. In ihr werden per `SQLiteOpenHelper` zwei Tabellen erstellt. Die Map-Tabelle beinhaltet Latitude & Longitude, UserID & Username und einen Origin-Wert. Die Klasse hat die Funktionen neue Koordinaten zur Datenbank hinzuzufügen, die Koordinaten aller User abzugreifen, die aktuelle Position auf der Karte zu entfernen und alle Koordinaten des Nutzers zu entfernen. Die zweite (Tagebuch-)Tabelle enthält den Titel des Eintrags, den Eintrag an sich, den Ort, das Datum, UserID und Username. Hier stehen folgende Funktionen zur Verfügung: Neuen Beitrag zur Datenbank hinzufügen, die Einträge des Nutzers löschen, und auf einen bestimmten Beitrag zugreifen. Die Datenbank für die Galerie wird in der `ImageDBHelper`-Klasse erstellt. Sie beinhaltet, einen Titel, einen Ort, ein Bild und die UserID. Hier werden Objekte der `CustomImage`-Klasse gespeichert. Ein Objekt dieser Klasse besitzt ein Bild (`byte[]`), einen Titel (String) und eine Ort (String).

Allgemein wurde für die meisten Funktionen der Datenbank-Klasse SQL-Befehle wie z.B. „`SELECT * FROM „ + TABLE_DIARY_ENTRIES + „ WHERE _id = „ id;`“ verwendet und per „`rawQuery()`“ ausgeführt.

Zwei Adapter (Map & Tagebuch) wurden anders wie in den Folien des Kurses mit einem Cursor versehen, um mit diesem die Inhalte der jeweiligen Datenbank-Tabelle abzugreifen.

Da der Login eine Online-Datenbank benötigt, damit sich ein Nutzer auf verschiedenen Geräten anmelden kann, wird der Login über einen Firebase-Authentikator abgewickelt. Um dann in den jeweiligen Fragmenten und Activities den aktuellen Nutzer abzufragen wird das `FirebaseUser`-Objekt verwendet. Mit der `getUid()`- und der `getDisplayName()`-Funktion wird immer der aktuelle Nutzer festgestellt. Da jede Datenbanktabelle auch immer den Nutzer speichert, werden auch in den Fragmenten die Daten des eingeloggten Nutzers abgerufen.

Im `MapFragment` wird die Karte gezeigt. Hier wird die `GoogleMap` des `GooglePlayServices` benutzt. Um die einzelnen Standorte zu verwalten und die aktuelle Position festzustellen werden hier auch der `LocationManager`, `LocationListener` und der `Geocoder` verwendet. (Abbildung 9)

Die Tagebucheinträge werden alles als `String` in der Datenbank gespeichert. Für die Darstellung der Einträge innerhalb des `DiaryFragment` wurden ein Adapter sowie ein Layout entworfen, der die Datenbank mit einem `ListView` verknüpft und die relevanten Inhalte darstellt. (Abbildung 4) Das Anzeigen der Einträge wird über eine `AsyncTask` durchgeführt, um bei einer höheren Datenmenge nicht die anderen Funktionen des Fragments zu unterbinden.

Im `GalleryFragment` werden die in der `ImageDBHelper`-Datenbank gespeicherten `CustomImage`-Objekte dargestellt. Auch hier wird die Anzeige über eine `AsyncTask` durchgeführt. (Abbildung 12)

Auch in der `ShowLocationListActivity` die durch betätigen des „Zeige gespeicherte Standorte“-Buttons im `SettingsFragment` werden ebenfalls über eine Adapter und einem `ListView` dargestellt. (Abbildung 8)

Zu Beginn der Implementierung wurden alle Bereiche (Karte, Galerie und Tagebuch) noch in eigenen Activities eingerichtet. Dies führte bei Einfügen der `BottomNavigation`,

also der Menüleiste zu Problemen. Ziel war es, die Menüleiste so gut wie immer sichtbar zu haben. Deshalb wurden die einzelnen Activities verworfen und eine Fragment-Struktur implementiert, die alle in der `StartActivity` liegen. Durch Nutzereingabe (Drücken der Menüpunkte) wird zwischen den Fragmenten gewechselt. Nur für die Eingabe der Tagebucheinträge und dem Hinzufügen von Bildern zum Album wurden extra Activities erstellt. Auch jetzt wurde erst die universelle `DatabaseHelper`-Klasse erstellt, um von ihr alle zwei nötigen Tabellen zu verwalten. Durch die Verwendung der SQL Syntax konnten einige Befehle sinngemäß auf andere Tabellen übertragen oder kopiert werden. Weniger Probleme gab es beim Einfügen der GoogleMap, das Speichern von Standorten und Anzeigen auf der Karte stellten sich schon als schwieriger heraus. Beim Versuch den `Place Autocomplete`-Service der Google Places API zu verwenden und die Orte dann auch direkt zu speichern, kam es auch immer wieder zu Fehlern. Deswegen wurde diese Idee wieder verworfen und `Geocoder` und `LocationManager` benutzt.

Probleme gab es bis zum Schluss mit der Galerie. Es führte immer wieder zu Problemen, mehrere Bilder vom Gerät zu erhalten und diese in einer Form zu speichern, die dann im `GalleryFragment` wieder als Bilder angezeigt werden konnten. Es wurde dann eine Tabelle aus der `DatabaseHelper`-Klasse entfernt und eine extra Datenbank für die Bilder erstellt. Zusätzlich wurde eine extra Objektklasse implementiert, die ein Bild, einen Titel und einen Ort enthält.

Dem Login ist ein Facebook-Login beigelegt, der aber nicht alleine reicht, um die App zu benutzen. Man benötigt also dennoch einen Firebase-Login. Der Facebook-Login hat bisher wenig nutzen, da die entsprechende Share-Funktion nicht umgesetzt wurde.

5 Testing

Der Usability-Test wurde noch durchgeführt, als die drei Bereiche Karte, Tagebuch und Galerie (noch nicht voll funktionsfähig) noch in separaten Activities implementiert waren. Ein Menü gab es nur auf dem Start-Bildschirm in Form von Buttons. Dies war ein großer Kritikpunkt der Tester, die gerne ein Universelles Menü gehabt hätten. Deshalb wurde dann die komplette bestehende Activity-Architektur verworfen und auf Fragmente umgestellt. Weiter gab es Kritik am Design. Das lag vor allem daran, dass das Layout der Buttons zu uneinheitlich war. Dies wurde behoben durch einheitliche Größen und Hintergrundfarbe. Auch die Farben im restlichen Layout, z.B. Titelleisten wurden angepasst. Auch stellten wir hier fest, dass beim Anzeigen von vielen Daten eine Verzögerung erfolgt. Dies konnte durch den Einsatz der AsyncTask verbessert werden.

6 Finaler Zustand und Ausblick

6.1 Finaler Zustand

Die Applikation konnte wie geplant umgesetzt werden. Die Kernfeatures sind soweit implementiert. Das Map-Tagging funktioniert problemlos und erlaubt es sowohl die aktuelle Position zu markieren, als auch einen anderen Ort. Das Tagebuch ist auch komplett fertig und beinhaltet Ort & Datum. Diese werden in der Datenbank gespeichert. Die Ortsliste wird zudem gespeichert. Der E-Mail-Export des Tagebuchs ist auch implementiert. Es können sowohl einzelne Einträge, als auch das ganze Tagebuch gesendet werden. Dem Fotoalbum kann immer nur ein Foto gleichzeitig hinzugefügt werden, mit Titel und Ort.

Die Zusatzfeatures funktionieren teilweise. Es gibt zwar einen Facebook-Login, aber noch keine Share-Funktion. Der Anfang der Online-Datenbank ist mit dem Firebase-Login zwar implementiert, der Aufwand der Datenbankimplementierung überstieg jedoch unsere Kapazitäten. Das Feature Bilder zu favorisieren wurde nicht umgesetzt, da die Implementierung der Galerie an sich größere Probleme bereitete.

In dem abgegebenen Projekt sind die Map Release Keys (noch) nicht eingetragen, da es dann immer zu Fehlern und Abstürzen kam.

6.2 Sinnvolle Erweiterungen

Grundsätzliche Erweiterungen sind die Features die im bisherigen Zustand noch fehlen. Einmal wäre das die Möglichkeit Einträge, Bilder oder Orte per Facebook zu teilen, um dem bisher funktionslosen Facebook-Login einen Nutzen zu geben. Des Weiteren fehlt das Zusatzfeature Bilder zu favorisieren. Dafür könnte die Datenbank so umgeschrieben werden, dass es eine neue Spalte mit einem Boolean gibt, der speichert, ob ein Bild favorisiert ist oder nicht. Das Layout könnte dann so verändert werden, dass favorisierte Bilder einen goldenen Rahmen erhalten oder einen Stern im Eck haben. Weiter müssen im Tagebuch Orte immer sehr genau angegeben werden, damit der richtige Ort gespeichert wird. (Bsp.: Eingabe: Mauritius; Gespeichert: Via Maurizius, Italien) Hier könnte man ein Dropdown-Menü implementieren. Dies wurde mit dem `Place Autocomplete`-Service versucht, lies sich aber noch nicht richtig umsetzen. Zudem könnte man im Ta-

gebuch noch die Funktion einbauen, einzelne Beiträge nachträglich zu bearbeiten. Weiter werden Bilder noch direkt gespeichert, anstatt der Datenpfad. Dies verlangsamt die App und macht das ganze etwas träge. Es sollten nur der Datenpfade gespeichert werden, die Umwandlung dieser zurück zu Bildern stellte sich allerdings als äußerst hartnäckiges Problem dar. Weiter wäre es sinnvoll, wenn der Galerie auch mehrere Fotos gleichzeitig hinzugefügt werden können.

Die vorgestellte Android-App kann auch in weitere Richtungen sinnvoll erweitert werden, die den Benutzer dazu motivieren, die App weiterhin oder überhaupt zu nutzen. So könnte *Travelnote* zum Beispiel um eine Funktion erweitert werden, die es Nutzern erlaubt Tipps zu einem bestimmten Ort zu hinterlegen, die dann den Nutzern in der Gegend angezeigt werden, damit diese ihre Reisen entsprechend gestalten können. Eine weitere Erweiterung wäre die Implementierung einer Möglichkeit von öffentlichen Tagebucheinträgen, die dann entweder mit dem Nutzer verknüpfte Freunde oder, je nach Einstellung, alle zu sehen bekommen. Weiterhin wäre es sinnvoll die App um eine Kalenderfunktion zu erweitern, die eingetragene Reisen, Tagebucheinträge und Orte verknüpft und weiterhin in bestehende (System-)Kalender integriert werden könnte.

7 Projektmanagement

Die interne Abstimmung verlief zumeist über WhatsApp und in Person an der Universität. Als Plattform für das Projekt wurde Git verwendet mit dem Webhoster Github.

Die Aufgaben im Projekt wurden folgendermaßen verteilt/bearbeitet:

Aufgabe	Verantwortlicher
Login (Firebase)	AO
Datenbank	AO, MW
Tagebuch	AO, MW
Karte	AO, MW
Galerie	AO, MW
Layout Entwurf	AO
Layout Anpassungen	AO, MW
Design	AO, MW
Dokumentation	MW

AO = Andreas Obermeier

MW = Michael Wexler