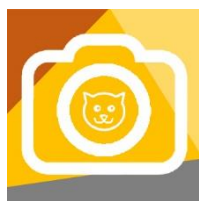


CATSHARE



Mica Zäch & Luca Hanimann
Überbetrieblicher Kurs (M335)
Version 1.0

Inhaltsverzeichnis

1	Ausgangslage.....	3
2	Definition der App.....	3
3	Nutzergruppe.....	3
4	Funktionalität.....	3
4.1	Must-haves	3
4.2	Nice-to-haves.....	3
5	Technisches Grobkonzept.....	4
5.1	Plugins	4
5.1.1	Kamera	4
5.1.2	Ortung.....	4
5.1.3	Speicherung	4
5.2	Umgebung	4
5.3	Frontend.....	4
5.4	Backend.....	4
6	Mock-Up.....	5
6.1	Screen 1.....	5
6.2	Screen 2.....	5
6.3	Screen 3.....	5
6.4	Farbkonzept.....	5
7	Umsetzung	6
7.1	Systemdarstellung	6
7.2	Graphical User Interface	6
7.3	Plugins	6
7.3.1	Kamera	6
7.3.2	Ortung.....	6
7.3.3	Speicherung	6
7.4	Probleme.....	7
7.5	Bugs.....	7
7.6	Tools	7
7.6.1	Flutter.....	7
7.6.2	Visual Studio Code	7
7.6.3	Dart.....	7
8	Bedienungsanleitung.....	8
8.1	Anforderungen	8
8.2	Anleitung	8
9	Testfälle.....	9
10	Schlusswort.....	10
11	Quelle.....	10
12	Abbildungsverzeichnis	10

1 Ausgangslage

Wir sind auf die Idee gekommen, da wir gerne Katzen mögen, besitzen und wir eine Native-App zu diesem Thema entwickeln wollen. Wir wollen Katzenbilder im Internet fördern und mit dieser App ein Tool dafür schaffen.

2 Definition der App

Mit dieser App können Nutzer Katzen erkennen, diese fotografieren und anschliessend zum Foto Text und den Standort hinzufügen und diese teilen.

3 Nutzergruppe

Katzenliebhaber, welche gerne Bilder von ihren Katzen machen wollen und diese teilen. Diese App richtet sich auf alle Altersgruppe und Geschlechter.

4 Funktionalität

4.1 Must-haves

- Es können Fotos gemacht werden
- Fotos in der Galerie zu speichern
- Text zum Foto hinzufügen
- Standort zum Foto hinzufügen

4.2 Nice-to-haves

- Erkennung von Katzen. Auf der Kamera wird in Echtzeit angezeigt, ob es sich beim fokussierten Objekt oder Lebewesen um eine Katze handelt.
- Globaler Feed
- Theme-Farbe ändern

5 Technisches Grobkonzept

5.1 Plugins

5.1.1 Kamera

Um die Kamera des Smartphones mit der App benutzen zu können, verwenden wir hier ein Flutter Plugin. (camera.dart)

Zum Hochladen von Bilder verwenden wir image_picker.dart.

5.1.2 Ortung

Für die genaue Standortbestimmung verwenden wir hier ein Flutter Plugin. (geolocator.dart)

5.1.3 Speicherung

Zur Speicherung der Fotos auf dem Smartphone verwenden wir ein Flutter Plugin. (path.dart und path_provider.dart)

Zum Teilen von erstellten Bilder benutzen wir esys_flutter_share.dart.

5.2 Umgebung

Das Projekt wird als Native-App realisiert. Dazu verwenden wir Flutter. Als Programmierumgebung verwenden wir hier Visual Studio Code.

5.3 Frontend

Mit links und rechts Wischen, kann zwischen den Screens gewechselt werden. Im «Feed» Tab kann die Option „Foto hochladen“ gewählt werden. Die App besitzt ausserdem noch ein Eingabefeld, um das geschossene Foto zu beschriften.

5.4 Backend

Die Fotos werden in der Galerie des Smartphones und in der Local Storage der App gespeichert um sie auch im Feed anzeigen zu können. Zudem greift die App auf den Ortungsdienst um den Standort des Benutzers zu bestimmen. Der Standort wird bei einer Änderung überprüft und aktualisiert.

6 Mock-Up

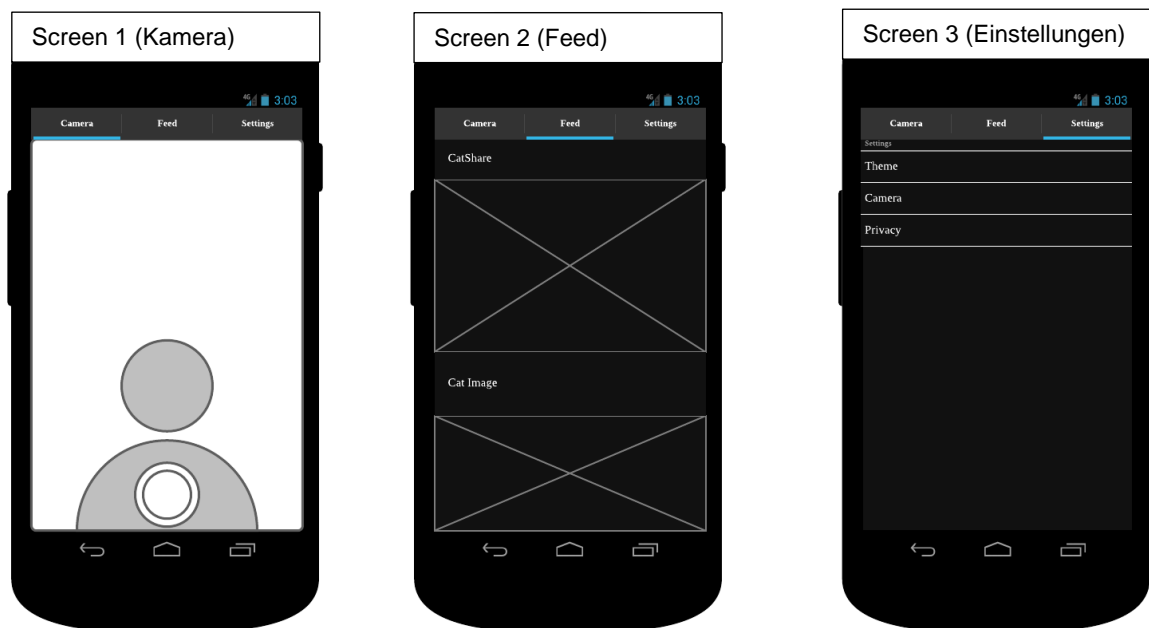


Abbildung 1 Mock-Up

6.1 Screen 1

Auf diesem Screen können Fotos der Katze gemacht werden. Danach wird der Benutzer aufgefordert ein Text hinzuzufügen und bekommt die Option, den Standort hinzuzufügen.

6.2 Screen 2

Hier befindet sich der Feed, in welcher, nebst der Galerie, alle gemachten Fotos abgespeichert sind. Es können auch Bilder aus der Galerie hochgeladen werden.

6.3 Screen 3

Im dritten Screen befinden sich noch Einstellungen, in welcher der Benutzer das Theme, die Kameraeinstellungen und Privatsphäre ändern kann.

6.4 Farbkonzept

Die Farben sind warm gehalten und sollen ein Wohlgefühl übermitteln.



Abbildung 2 Farbkonzept

7 Umsetzung

7.1 Systemdarstellung

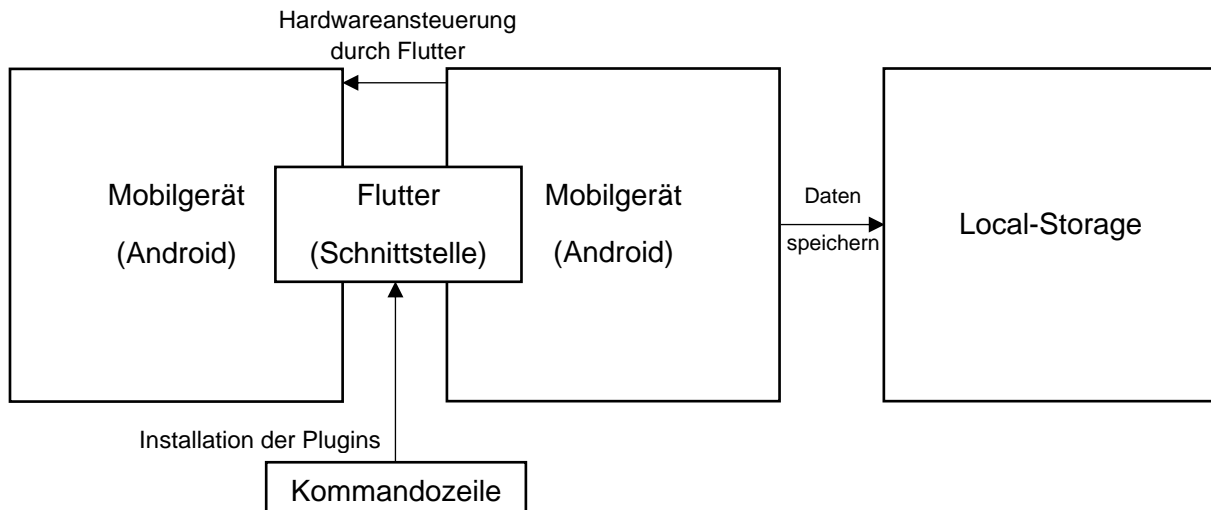


Abbildung 3 Systemdarstellung

7.2 Graphical User Interface

Das ganze Design, die Navigation, das Wischen zwischen den verschiedenen Screens wurde mit Dart und Flutter realisiert. Für die Buttons haben wir `material.dart` verwendet.

Das Framework besteht aus Flutter. Die Widgets und Designelemente stammen von der `material`-Bibliothek.

7.3 Plugins

7.3.1 Kamera

Wir verwenden `camera.dart` für den Kamerazugriff. Dieses erstellt das Kamera-Widget, welches wir in den vorgesehenen Screen eingefügt haben.

7.3.2 Ortung

Für die Ortung haben wir `geolocator.dart` verwendet. Von ihr haben wir eine ganz einfache Methode genommen, welche die Position in Längen- und Breitengrad ausgeben kann. Bei einer Aufnahme erfasst die App den Standort des Smartphones, jedoch kann sie keinen Standortinformation von einem hochgeladenen Bild erfassen.

7.3.3 Speicherung

Die Bilder, Texte, Standorte werden direkt im Cache der App gespeichert. Dazu diente uns `pathprovider.dart` und `path.dart`.

Für die Teilen-Funktion haben wir einen vorgefertigten Button vom Plugin `esys_flutter_share.dart` verwendet.

7.4 Probleme

Unser Hauptproblem war die Zeit. Da wir unter anderem eine neue Sprache, sowie neues Framework ausgewählt haben, mussten wir Zeit mit dem auseinandersetzen investieren.

Wir haben uns im Nachhinein betrachtet, zu viele Ziele gesetzt, welche in dieser zur Verfügung gestellten Zeit nicht realisierbar waren.

Die Sprache Dart und das Framework Flutter, welches wir für dieses Projekt ausgewählt haben, ist relativ neu, sodass es erst wenig Ressourcen zum Recherchieren gibt.

7.5 Bugs

Bug	Beschreibung
Verschwinden des Bildes	Nach einem erstellten Bild samt Text und Standort verschwindet es im Feed, sobald nach links oder rechts auf andere Screens gewischt wird.

Tabelle 1 Bugs

7.6 Tools

7.6.1 Flutter

Flutter verwenden wir für die Schnittstelle zwischen der App und den Hardwarefunktionen des Smartphones. Wir brauchen Flutter um auf die Kamera, den GPS-Sensor zu zugreifen und für das Speichern von Bildern in der Galerie des Smartphones.



Abbildung 4
Logo Flutter

7.6.2 Visual Studio Code

Visual Studio Code ist eine plattformübergreifende, freier Quelltext-Editor. Wir verwenden es als Programmierumgebung für die Entwicklung dieser App.



Abbildung 5
Logo VS Code

7.6.3 Dart

Dart ist eine Programmiersprache, die von Google entwickelt wird. Die Sprache wurde als generelle Vielzweck-Programmiersprache entworfen, die insbesondere eine moderne Alternative zur Verwendung in Webbrowsern darstellen soll.



Abbildung 6
Logo Dart

8 Bedienungsanleitung

8.1 Anforderungen

Um die App benutzen zu können, muss das Smartphone ein Android Betriebssystem haben und optimalerweise auch eine aktive GPS-Verbindung.

8.2 Anleitung

Nach dem Öffnen der App, befindet man sich in der Kamera. Nun kann ein Foto gemacht werden. Danach kann das Bild geteilt werden.

Im Feed kann ein Bild gemacht, als auch Hochgeladen werden. Dazu klicken Sie auf den roten Button mit dem Plus auf dem Bild.

Sobald ein Bild gemacht oder Hochgeladen wurde, fordert die App eine Texteingabe.

Wenn diese gespeichert ist, erscheint im Feed nun das Bild samt Text und Standort.

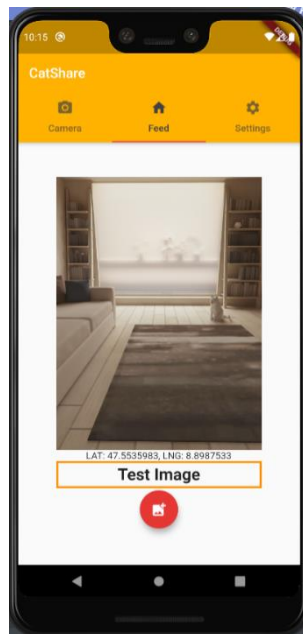


Abbildung 8 Feed

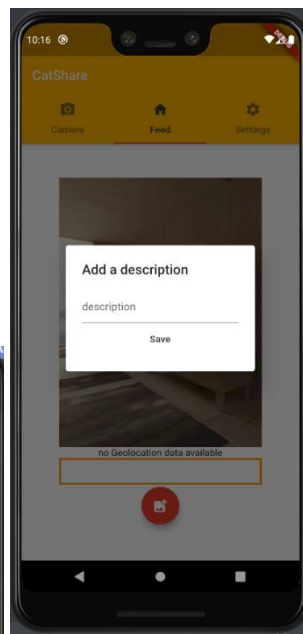


Abbildung 9 Texteingabe



Abbildung 7 Kameraansicht

9 Testfälle

Getestet von:	Mica Zäch, Luca Hanimann
Getestete Version:	Version 1.0
Verwendete Frameworks:	Flutter, Dart
Plattform:	Phone
Datum:	02.10.2020

Was wird getestet?	Was wird erwartet?	Wie wird getestet?	Was ist das Resultat?	Beschreibung
Grafische Benutzeroberfläche				
Die Navigationsbar wird geprüft.	Man befindet sich auf Screen 1 (Kamera)	Öffnet die App		
Die Navigationsbar wird geprüft.	App wechselt zu Screen 2	Wischt einmal nach rechts		
Theme Farbe	Theme Farbe ändert sich auf Grau	Wischt zum rechten Screen (Settings/Screen 3), wählt Themes und dann Grau aus.		Aus Zeitgründen noch nicht implementiert.
Texteingabe	Im Feed wird zum Bild den Text «Testtext» angezeigt..	Erstellt ein Foto und gibt «Testtext» ein.		
Kamera				
Kamerafunktion/ Fotografieren	App zeigt in Echtzeit Bild	Bewegt das Smartphone und beobachtet		
Kamerafunktion/ Fotografieren	App lässt ein Foto machen	Tippt auf den Kreis in der Mitte unten.		
Standort				
Standortbestimmung	Am Foto wird der Standort «Frauenfeld» angezeigt.	Erstellt ein Foto, gibt beim Text «Testfall Standortbestimmung» ein.		Es gibt Längen- und Breitengradzahl aus.
Speicherung				
Sicherung in der Galerie	Das Foto wird in der Galerie des Smartphones gespeichert.	Erstellt ein Foto, gibt beim Text «Testfall Galerie» ein und wechselt zur Galerie App.		Aus Zeitgründen noch nicht implementiert.
Sicherung im Feed	Das Foto wird samt Text und Standort im Feed angezeigt.	Erstellt ein Foto, gibt beim Text «Testfall Feed» ein.		
Sicherung im Feed	Das Foto soll samt Text und Standort im Feed angezeigt werden. Auch bei einer Schliessung der App	Erstellt ein Foto, gibt beim Text «Testfall Feed» ein. Schliesst die App. Öffnet diese wieder und wischt einmal nach rechts in den Feed.		Fehler noch nicht gefunden. Aus Zeitgründen noch nicht implementiert.

Tabelle 2 Testfälle

10 Schlusswort

Es war eine sehr interessante Woche und eine tolle Erfahrung. Die Teamarbeit hat gut funktioniert. Eine neue Sprache und Framework zu benutzen war vielleicht nicht die allerbeste Idee, trotzdem konnten wir darin Fuss fassen und in diesen Umständen (u.a. der gegebenen Zeit) eine solide App verwirklichen. Wir nehmen uns in Zukunft zu Herzen, nicht allzu viele und grosse Ideen zu planen.

11 Quelle

Flutter:

<https://flutter.dev/>

Geolocator:

<https://pub.dev/packages/geolocator>

Share:

https://pub.dev/packages/esys_flutter_share

Camera:

<https://pub.dev/packages/camera>

<https://flutter.dev/docs/cookbook/plugins/picture-using-camera>

Path Provider:

https://pub.dev/packages/path_provider

Path:

<https://pub.dev/packages/path>

Image Picker:

https://pub.dev/packages/image_picker

12 Abbildungsverzeichnis

Abbildung 1 Mock-Up	5
Abbildung 2 Farbkonzept.....	5
Abbildung 3 Systemdarstellung	6
Abbildung 4 Logo Flutter	7
Abbildung 5 Logo VS Code	7
Abbildung 6 Logo Dart.....	7
Abbildung 7 Kameraansicht.....	8
Abbildung 9 Feed	8
Abbildung 8 Texteingabe.....	8
Tabelle 1 Bugs	7
Tabelle 2 Testfälle	9