



Carcass

by Mathias Wöß

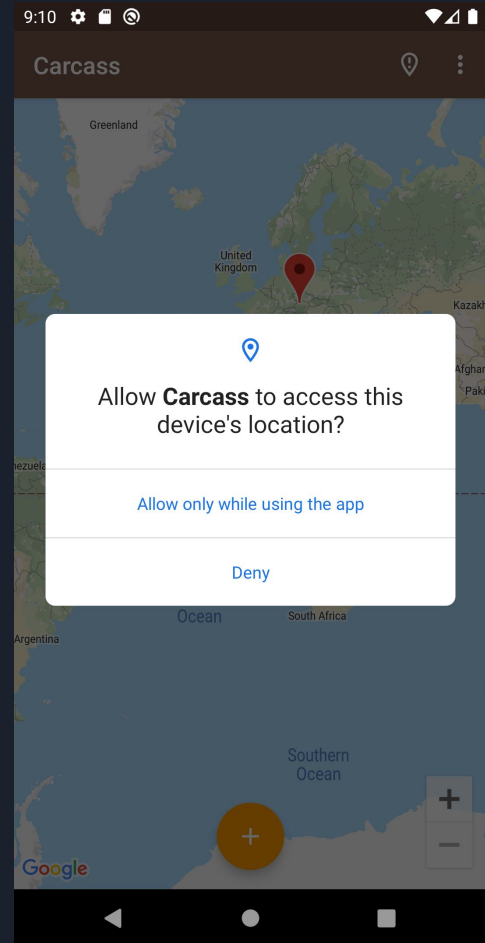


Idea

- Report dead animal body locations
- Inform people and authorities
- Keep neighbourhood clean
- Help getting rid of the carcasses
- Provide optional picture
- Anonymous
- Real time updates

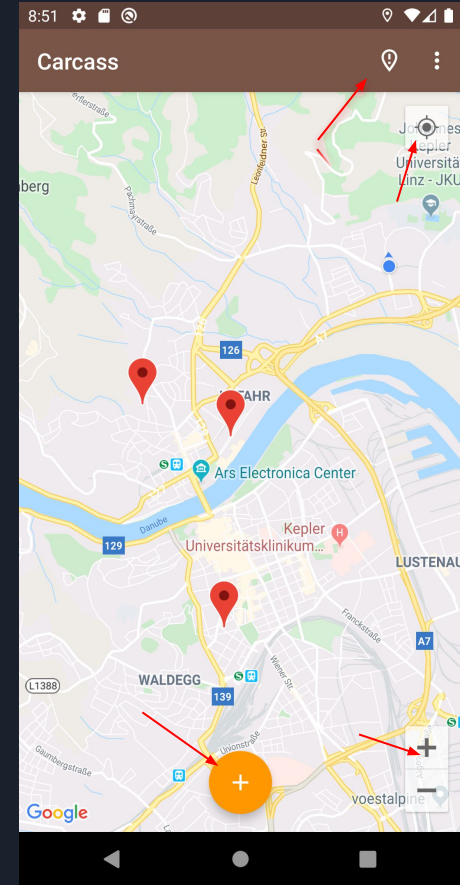
Permissions

- Ask user to provide location permission



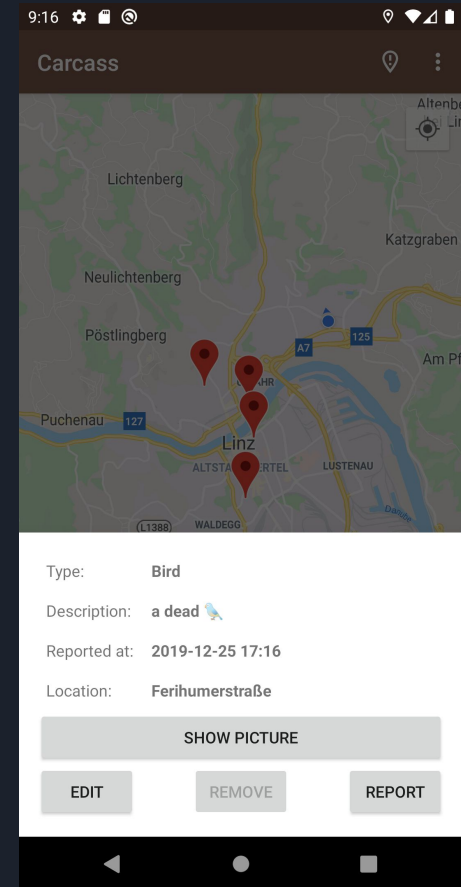
Main UI

- Google Maps integration
- Firebase DB for data
- Show current location
- FloatingActionButton to add
- ActionBar to show/hide flagged Markers
- Center Location
- Zoom-Buttons



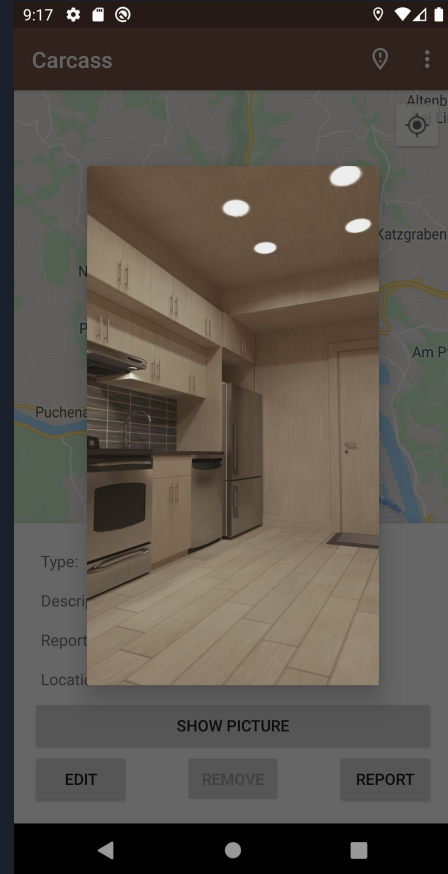
Details

- BottomSheetFragment
- Details about
 - Type
 - Description
 - location and
 - Time of reporting
- Buttons
 - Edit
 - Remove
 - Report/Flag
 - Show picture
 - Because not always pleasant



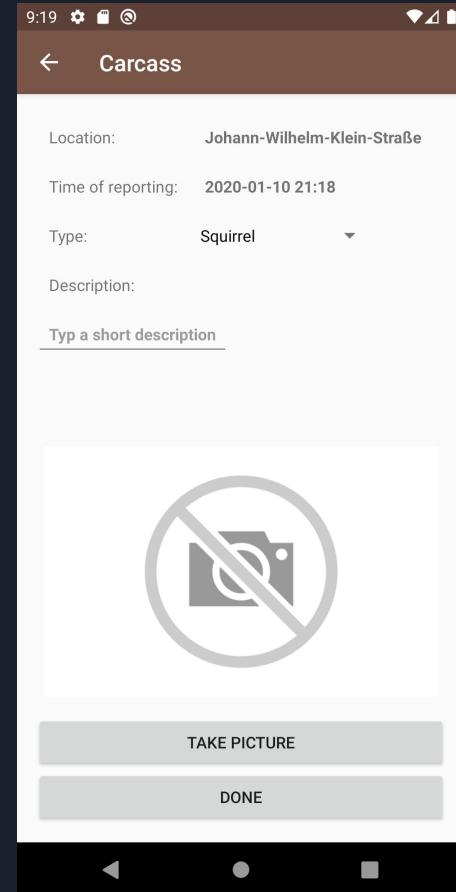
Show picture

- Get from server
- Show as overlay
- Easily dismissable
- On demand loading
 - Data saving



Add

- Add new entries at current location
- Capture time of reporting
- Choose kind/type
- Optional description
- Add picture
 - Camera Intent



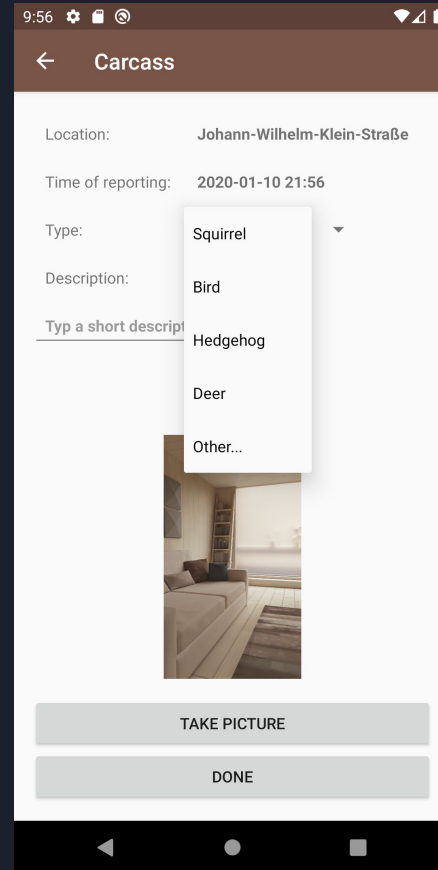
The screenshot shows a mobile application interface for adding a carcass report. The title bar at the top is brown and contains a back arrow, the title "Carcass", and status icons for time (9:19), settings, notifications, and battery. The main form area is white and contains the following fields:

- Location:** Johann-Wilhelm-Klein-Straße
- Time of reporting:** 2020-01-10 21:18
- Type:** Squirrel (with a dropdown arrow)
- Description:** A text input field with the placeholder text "Typ a short description".

Below the description field is a large square area containing a camera icon with a diagonal slash through it, indicating that a picture can be added. At the bottom of the form are two buttons: "TAKE PICTURE" and "DONE". The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps icons.

Select type

- Choose type of animal
 - Data from server
- Offer “Other...” entry
 - Add new types



The screenshot shows a mobile application interface for reporting a carcass. The title bar at the top is brown with a back arrow and the text "Carcass". The status bar above it shows the time 9:56 and various icons. The form fields are as follows:

- Location:** Johann-Wilhelm-Klein-Straße
- Time of reporting:** 2020-01-10 21:56
- Type:** A dropdown menu is open, showing options: Squirrel, Bird, Hedgehog, Deer, and Other... (highlighted).
- Description:** A text input field with the placeholder "Typ a short description".
- Image:** A placeholder image showing a modern interior with a sofa and a window.

At the bottom, there are two buttons: "TAKE PICTURE" and "DONE". The Android navigation bar is visible at the very bottom.

Other...

- Define new type
- Show modal bottom sheet dialog
- Open keyboard and focus

The screenshot shows a mobile application interface. At the top, the status bar displays the time 9:23 and various icons. Below it, a dark header bar contains a back arrow and the title 'Carcass'. The main form area has a light gray background and contains the following fields:

- Location:** Johann-Wilhelm-Klein-Straße
- Time of reporting:** 2020-01-10 21:18
- Type:** Other... (with a dropdown arrow)
- Description:** A text input field with the placeholder text 'Typ a short description'.

A modal bottom sheet is currently open, displaying a white background with the text 'Please specify the name' and a 'SAVE' button in the top right corner. Below the text is a text input field. At the bottom of the screen, a keyboard is visible, showing a standard QWERTY layout with a search icon, a microphone icon, and a green checkmark button on the right.

Other...

- Check duplicates
- Only define new
- Synchronize with database

9:24

← Carcass

Location: Johann-Wilhelm-Klein-Straße

Time of reporting: 2020-01-10 21:18

Type: Other...

Description:

Typ a short description

Please specify the name SAVE

Type already exists

Deer

> Deer | 🦌 | Dear | 🎤

q w e r t y u i o p

a s d f g h j k l

⬆ z x c v b n m ⬆

?123 , 😊 . ✓

Edit

- Update existing entries
- Pre-Fill data
- Load picture from server if exists
 - Show cropped
- Add/update missing/existing picture
- Provide missing/update existing description
- Change type

7:36

← Carcass


Current location: Amphitheatre Parkway

Time of reporting: 2019-12-07 15:33

Type: Deer ▼

Description:

A dead deer. Probably was killed by a car crash.



EDIT PICTURE

DONE

Database

2 major nodes:

- animalTypes
 - Name
- Carcasses
 - Type
 - Location
 - Description
 - Date (Epoch format)
 - URL to picture
 - Flagged



Storage

- Keep full sized images
- Update on edit
- Remove if marker removed
- Link to entry by “url” field

The screenshot shows the Google Cloud Storage web interface. The address bar displays the path: `gs://inspiring-team-257815.appspot.com > images > -LvVm6q`. A blue button labeled "Upload file" is in the top right. Below the address bar is a table listing files:

<input type="checkbox"/>	Name	Size	Type	Last modified
<input type="checkbox"/>	JPEG_20191207_153414_818369958051_24.jpg	139.58 KB	image/jpeg	Dec 7, 2019

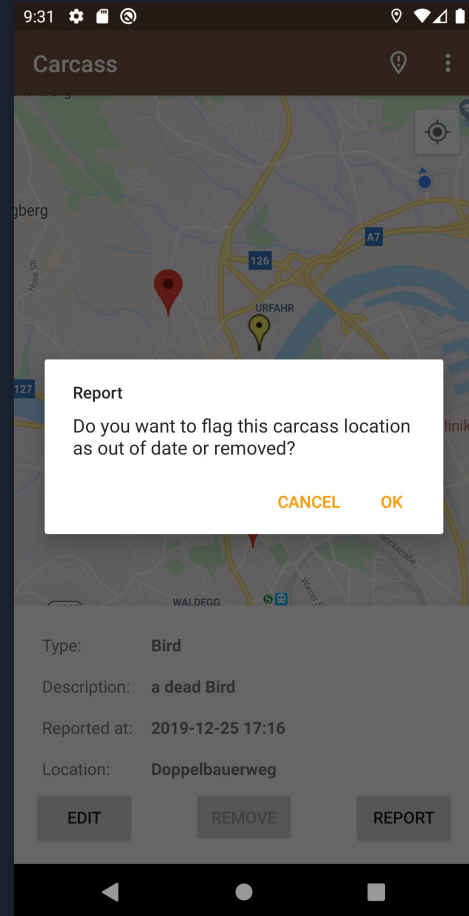
To the right of the table, a preview of the selected file is shown. The preview includes a thumbnail of a checkered board on a stand. Below the thumbnail, the following metadata is displayed:

- Name: [JPEG_20191207_153414_818369958051_24.jpg](#)
- Size: 142,927 bytes
- Type: image/jpeg
- Created: Dec 7, 2019, 3:34:26 PM
- Updated: Dec 7, 2019, 3:34:26 PM

At the bottom, there are two expandable sections: "File location" and "Other metadata".

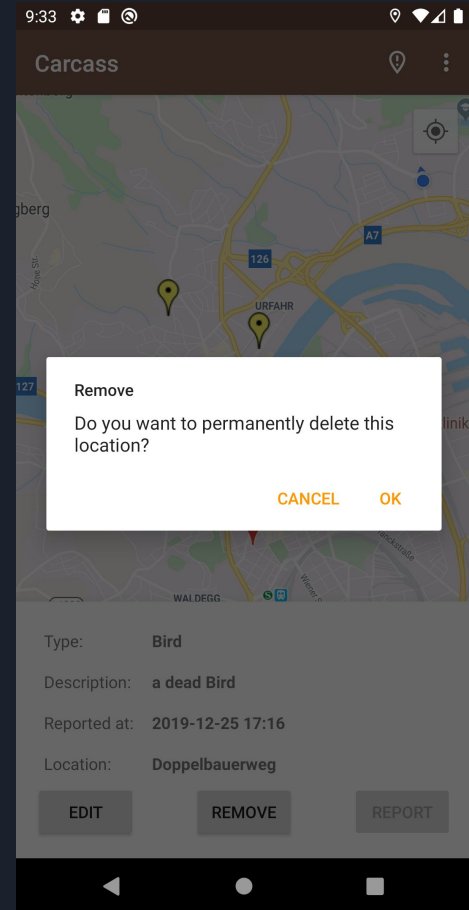
Flag entry

- Provide option to flag marker
 - Out of date
 - Removed
 - Inappropriate
- Visualize by different icon
- Reset flag by editing entry again



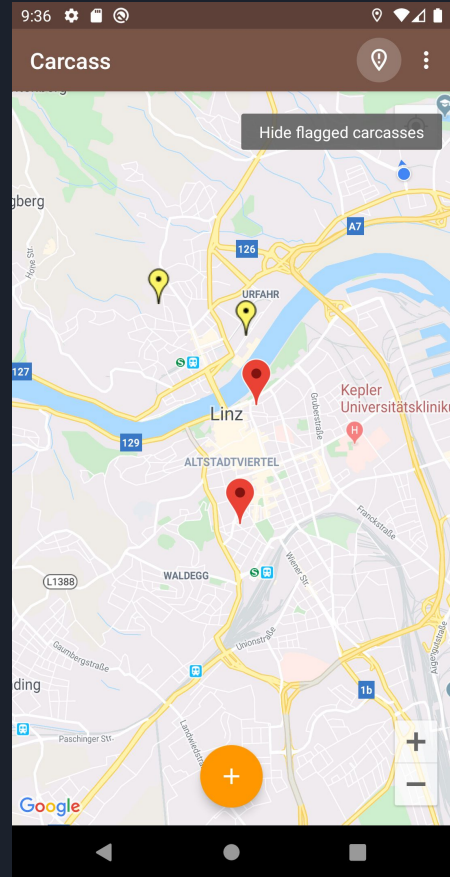
Remove entry

- Only when flagged
- Do not provide at once
 - 1 day
 - 1 week
 - ...
 - Give chance to update
- Remove button disabled until then



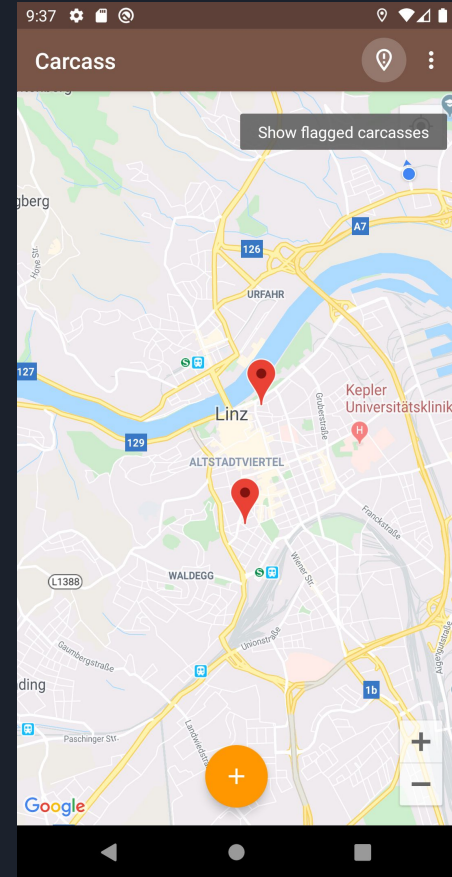
Hide flagged markers

- Action bar icon
- Show or hide, depending on state
- Filter “useless” information
- Persist in state bundle



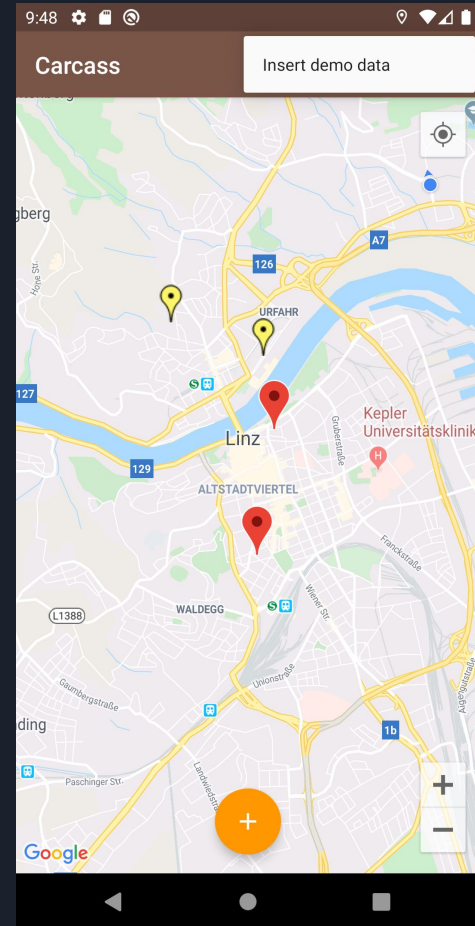
Hide flagged markers

- Action bar icon
- Show or hide, depending on state
- Filter “useless” information
- Persist in state bundle



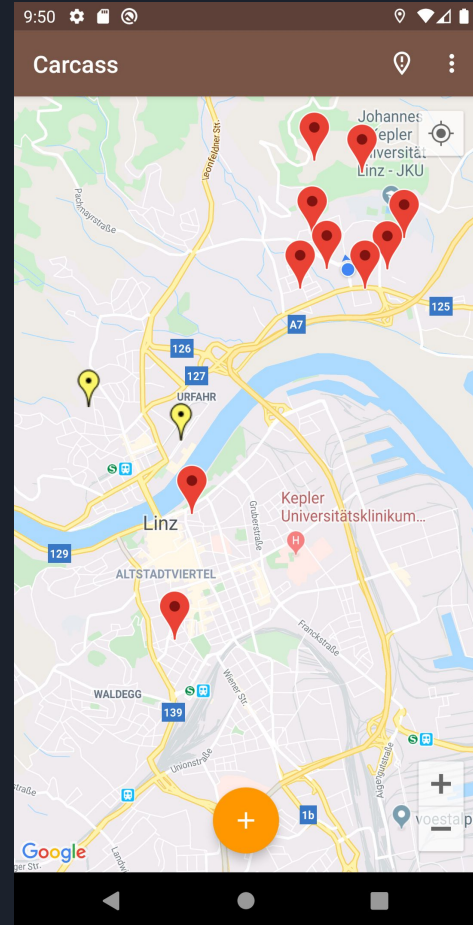
Insert Demo Data

- Debug function to test app
- Adds random markers around current location
- Random type and description



Insert Demo Data

- Debug function to test app
- Adds random markers around current location
- Random type and description





Live demo in emulator



Complications and Challenges

- Firebase DB integration
- Json representation of all data and parsing
- Additional Firebase Storage integration for binary (non-string) data (images)
- Kotlin (new language)
- Maps API
- Camera functionality and image processing
- Android lifecycle
- Location updates



Thank you!

It was fun developing on Android!

Questions?