



# Events4You

All your events in one place



# Goals until today

## Implementing download feature

Cached data gets stored persistent, so there is no extra network traffic

The stored events will be removed if the events ending date is exceeded



# Goals until today

## Enhanced profile management

The profiles are now stored on and managed by the server

It is possible to create new profiles / sign up

Auto login for the last logged in user

You can skip login and use an anonymous profile



# Goals until today

## Implementing the search function

Search is available and the results are popping up as an extra tab, which will be closed if deselected

Search results are generated by the tags of an event

The closer events which contain searched tags are more relevant and appear on top



## Further work

Communication with actual server

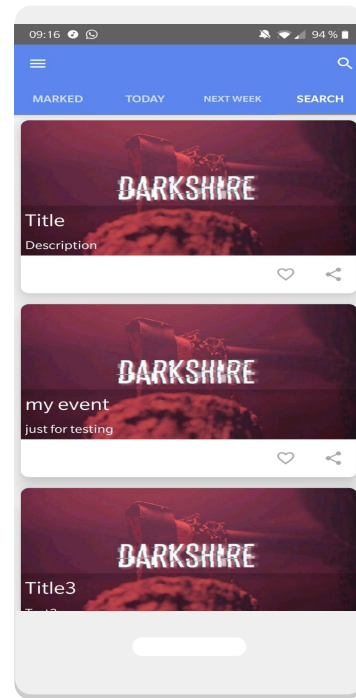
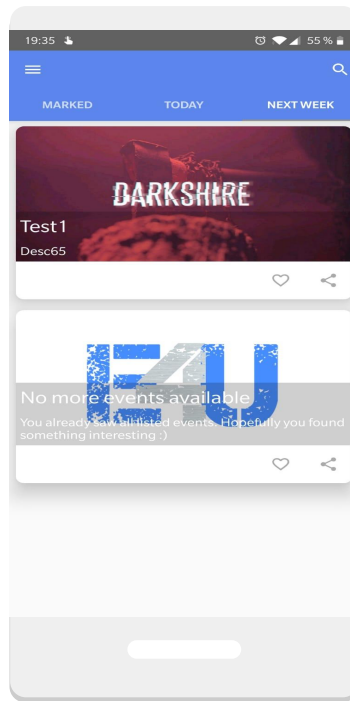
Creating events is now possible

# New layout frames

## Search tab

4th tab which opens when the user searches for an event

closes if the user swipes to another tab or changes view to anything else (e.g. settings page)

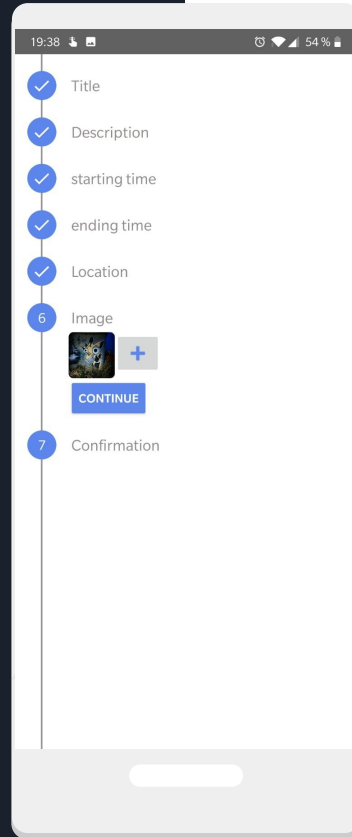


# Layout frame

Create new events

Asks for all necessary information to create an event

Creates event and sends it to the server



Usability common practice like

IME Action “DONE” on single line inputs

Image thumbnails which expand to full size by clicking

Material date and time picker



# Adaptation concept overview

Location Based

Network connectivity

decides bulk size

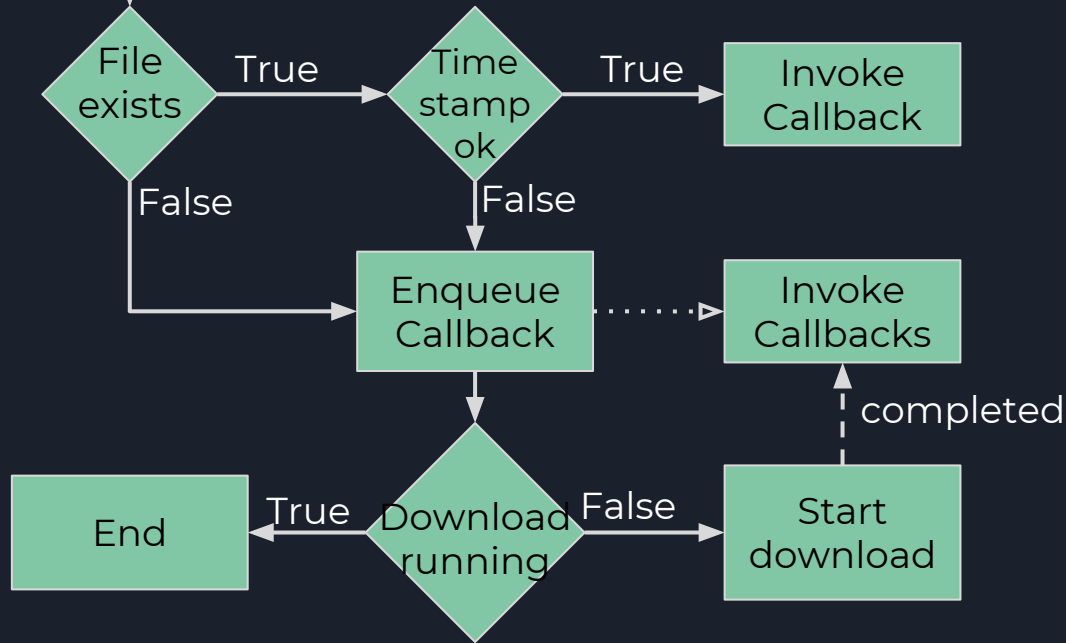
Indexing event object

Download persistent caching



# Connectivity - Batching

Download URL  
to File





## Connectivity - Index (Traffic reduction)

Request “index” file instead of all events

Binary representation of “header-fields” of events:

**ID** (UUID; 16 bytes),

**Start/End-time** (2 \* long; 2 \* 8 bytes),

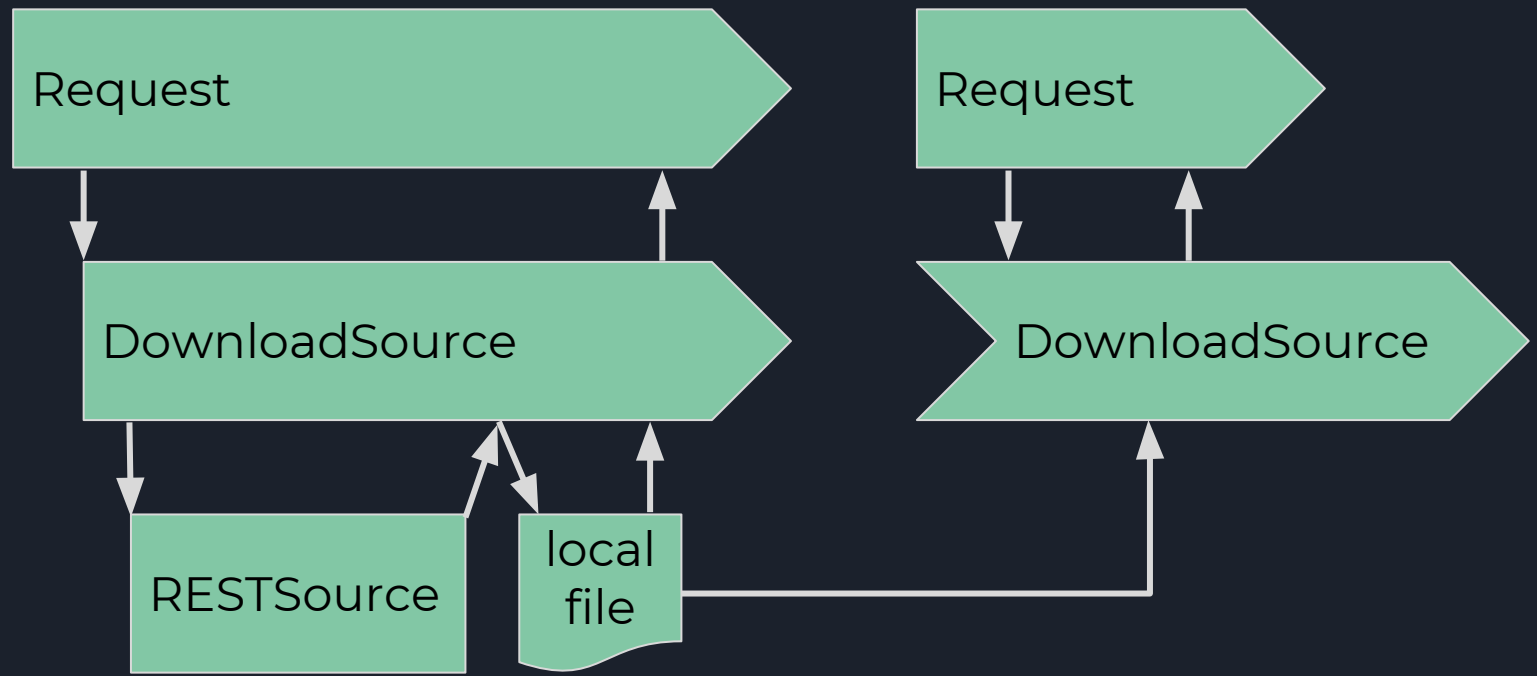
**Location** (2 \* float; 2 \* 4 bytes)

=> 40 bytes per card

Sort based on headers from index

Download details in sorted order as needed  
(respecting bulk-rules)

# Availability - Persistence





## Lessons learned

Cleartext HTTP-connections require permission in the AndroidManifest

Network communication cannot be run on the UI-Thread

Busy waiting becomes fine if it's delayed

Using logging on the correct levels helps with debugging



## Lessons learned

Flushing streams before closing them is a **very** good idea

Disable auto-login after logout

Running code can reduce bugs dramatically

Mock it 'till you write it

“git commit -a -m” is a thing



# Lessons learned

Single Activity Application is a good design

Fragments inside fragments require another fragmentmanager

Adding views at runtime can cause gravity issues

Gesture controlled views inside gesture controlled views are difficult to handle correctly



## Lessons learned

Asynchronicity is a curse and a blessing at the same time

Race conditioning threads are no fun to debug

Stackoverflow is such a help (!)

Deadlines can be pretty surprising ; )



## Further goals

Implementing the blacklisting feature again

Get some profile / user security going

Get some server security

Build a crawler to get real events

Build notification system

Advanced settings

Improve search results

Build a recommendations tab

Enhance profiles

Implement social networking, so you can link artist in the events or see what friends are interested in

Personalize the app UI

...