

Dare You Go



Owais Musa | Muhammad Watad | Baraa Hleihil | Khaled Fahoum

Project advisor: Kfir Lev-Ari

Networked Software Systems Laboratory, Technion

Back-end Team



Front-end Team



What will we talk about?

- What is “Dare You Go”?
- Use case Example
- Features
- Front-end & User Interface
- Solution Architecture
- Design Details
- Tests
- Development Environment
- The Future
- Live Demo

What is “Dare You Go”?

A social ‘dare-proof’ game

- Android application
- The user opens a dare, other users can participate and respond with proofs
- The winner is determined by the user who opened the dare

What is “Dare You Go”? (2)

- Dare (challenge): a well defined task with deadline.
 - It can be text, video or image.
 - Each dare can have unlimited number of proofs and comments
- Proofs are likeable. The more your proof is liked, the higher you get in the leaderboard.

Use Case Example

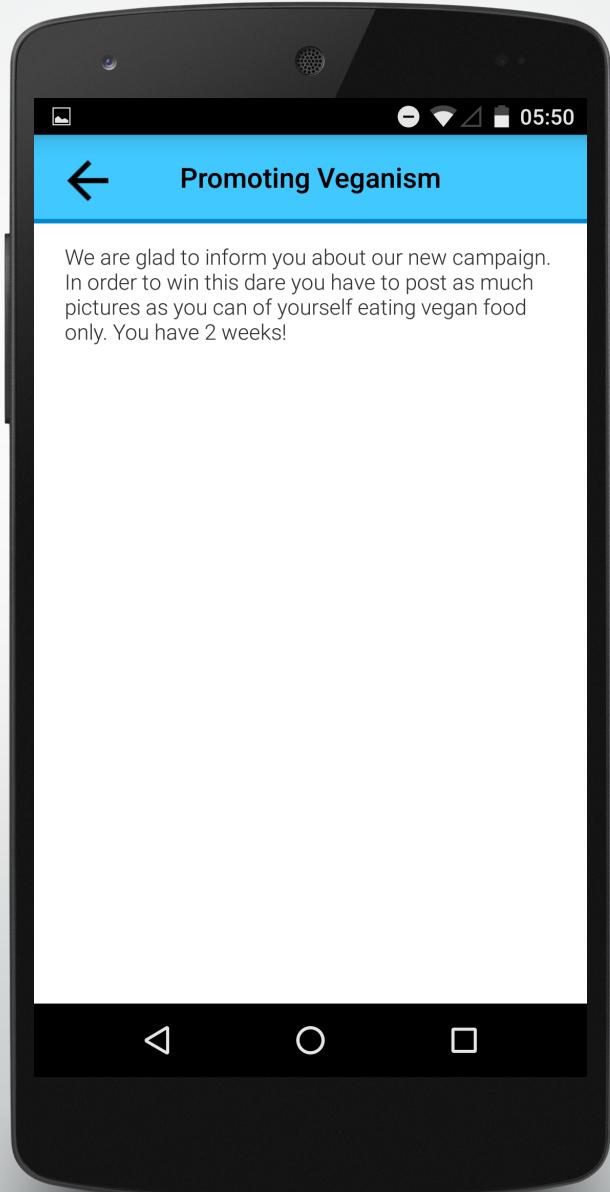
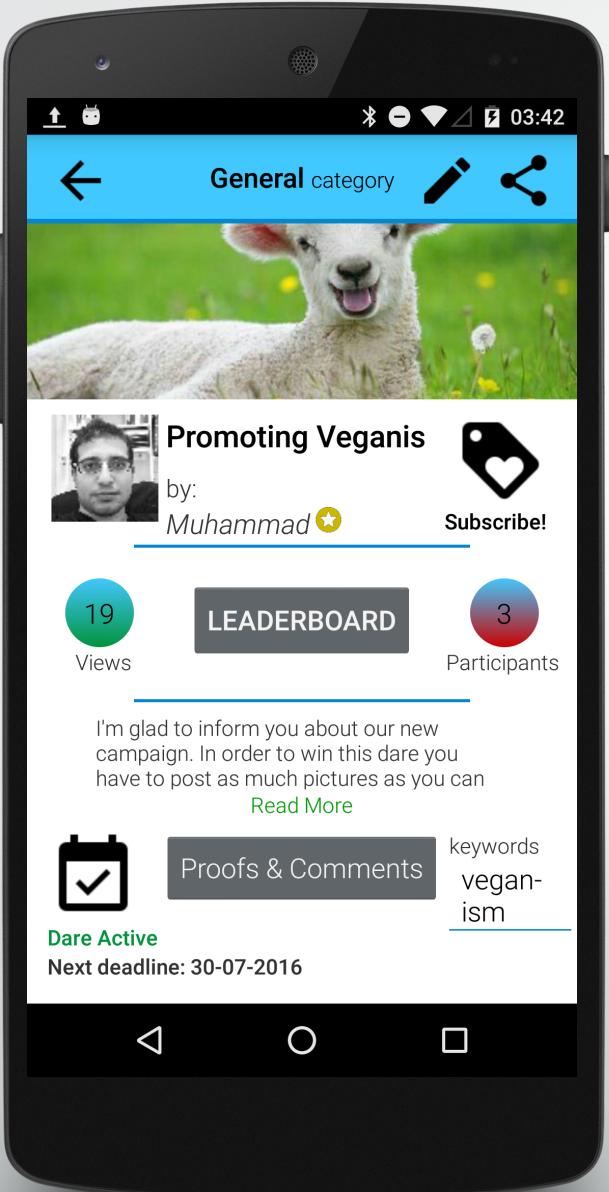
How can I
promote veganism
and animal rights?

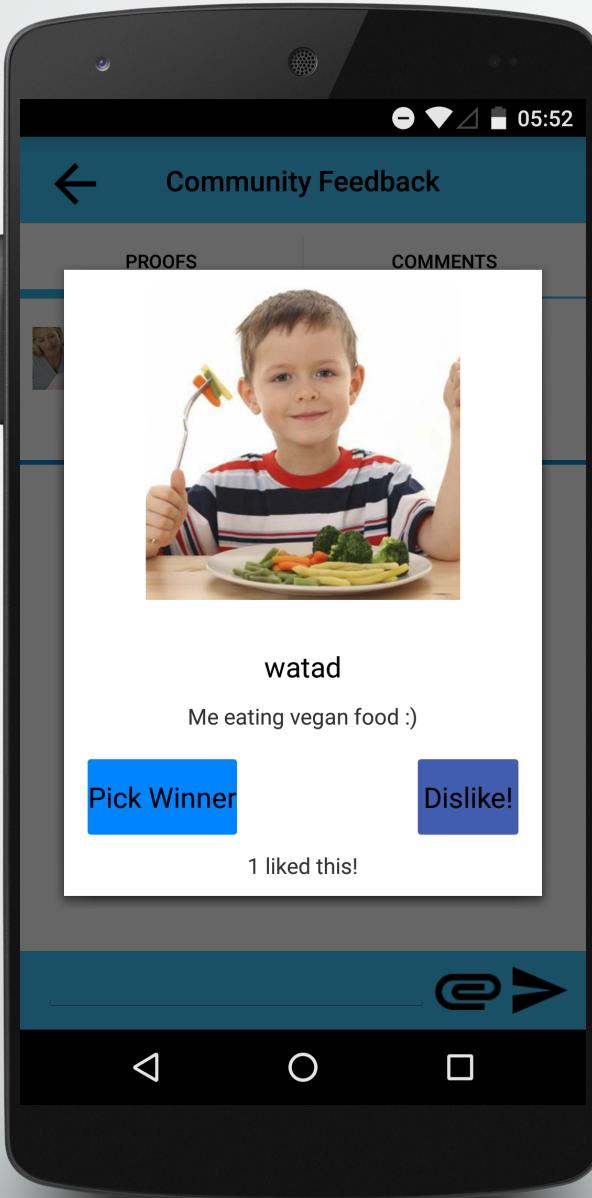




Lets challenge
people!





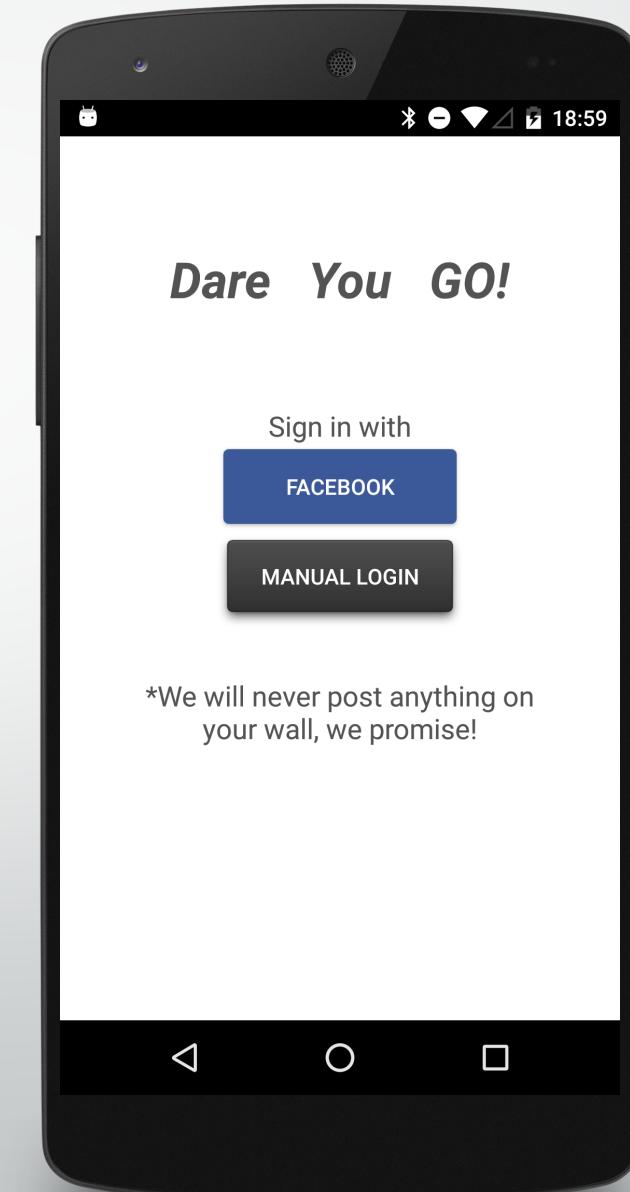


Features

Creating User

To start using the application you need to sign up:

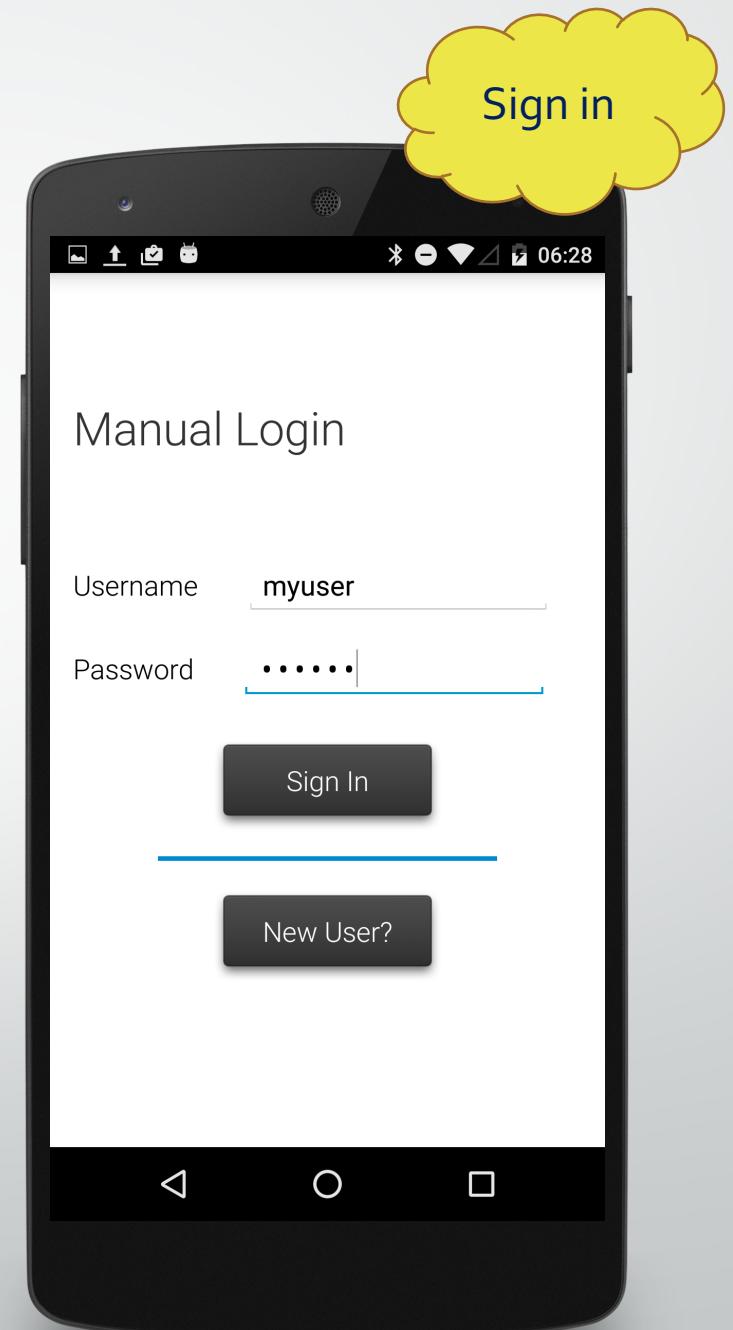
- Manually, with a new Parse user or
- Using your Facebook account



Creating User

To start using the application you need to sign up:

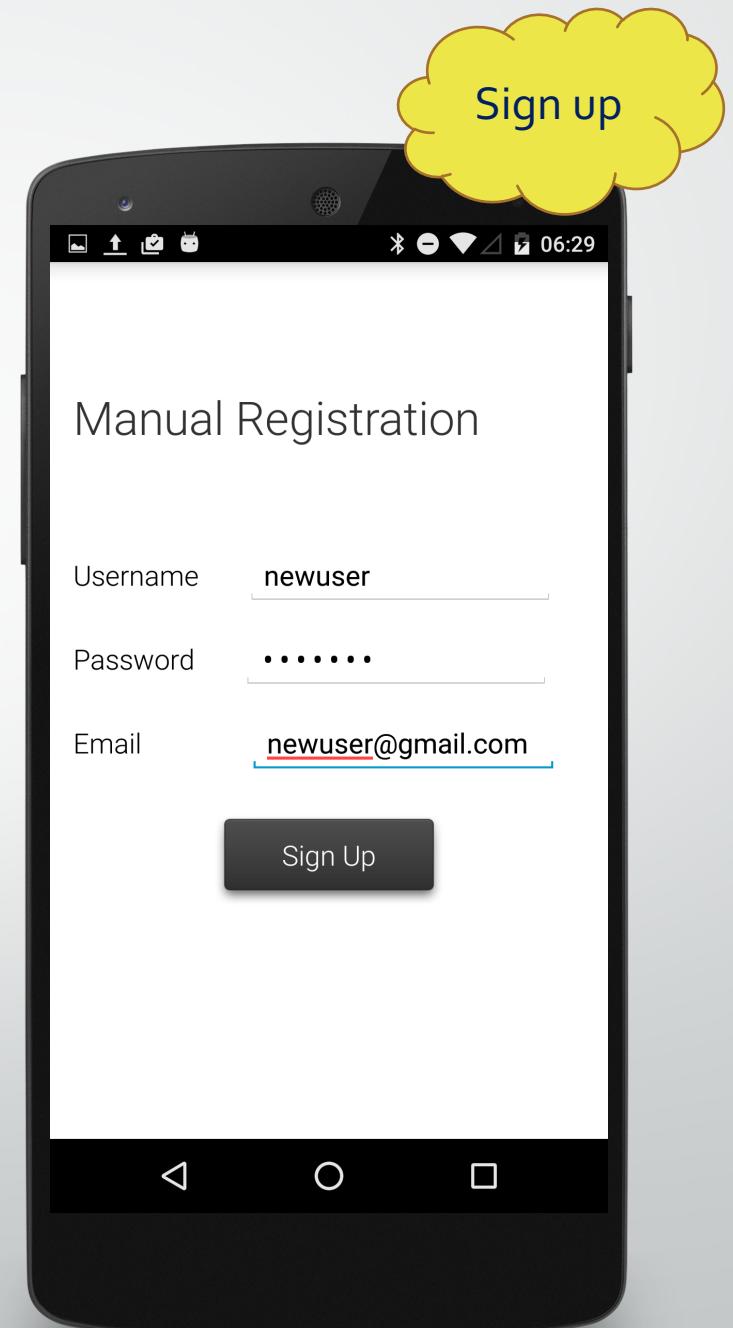
- Manually, with a new Parse user or
- Using your Facebook account



Creating User

To start using the application you need to sign up:

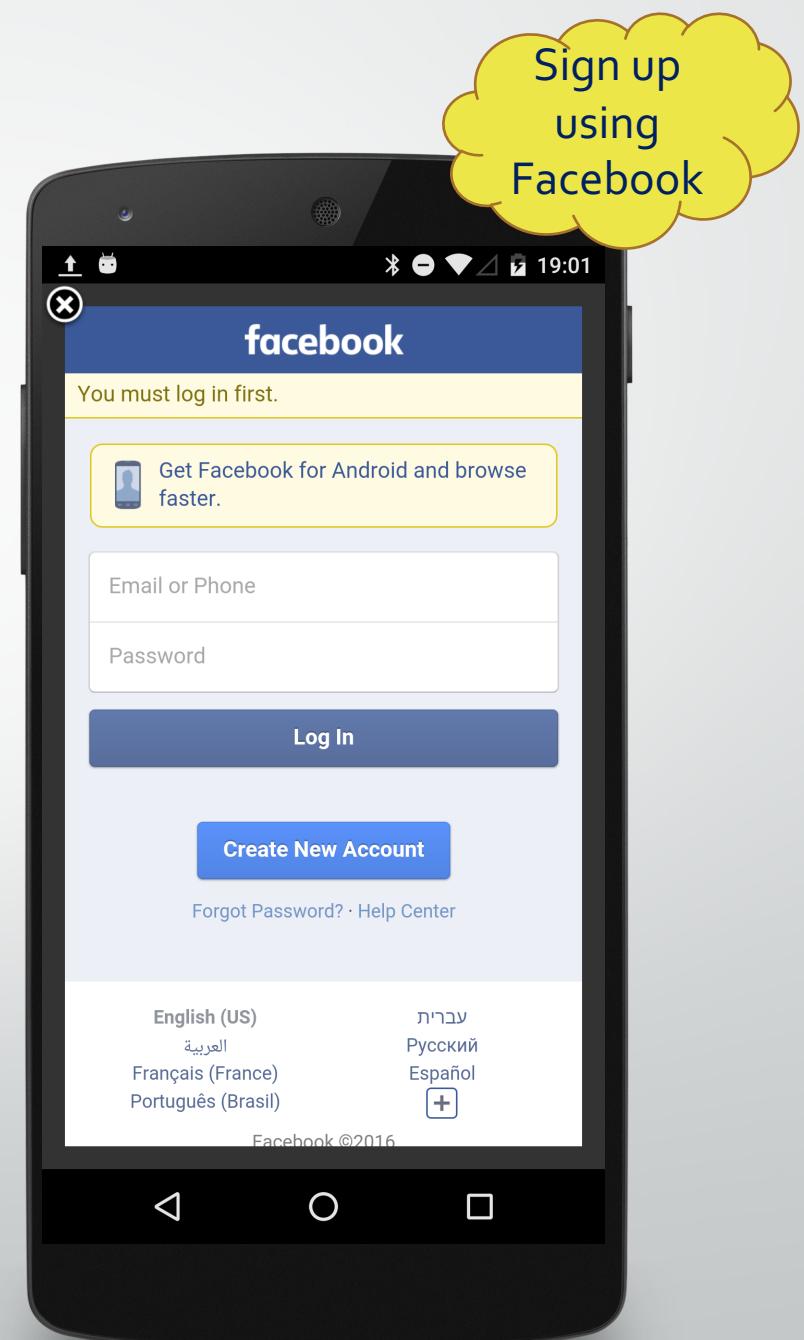
- Manually, with a new Parse user or
- Using your Facebook account

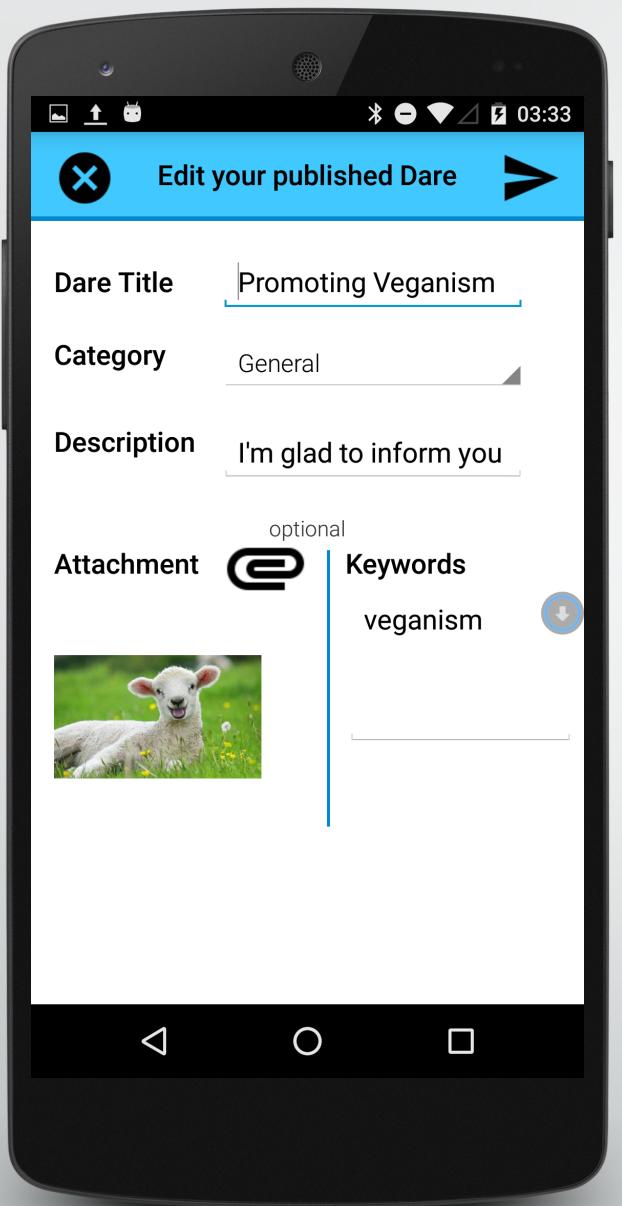


Creating User

To start using the application you need to sign up:

- Manually, with a new Parse user or
- Using your Facebook account





Creating Dare

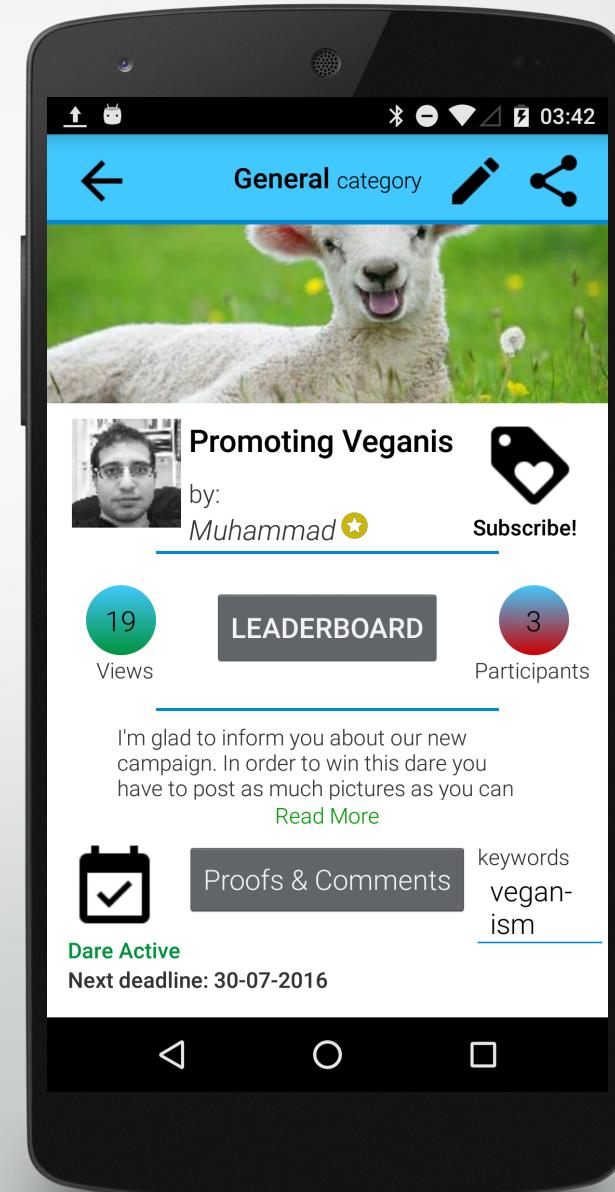
Dare includes:

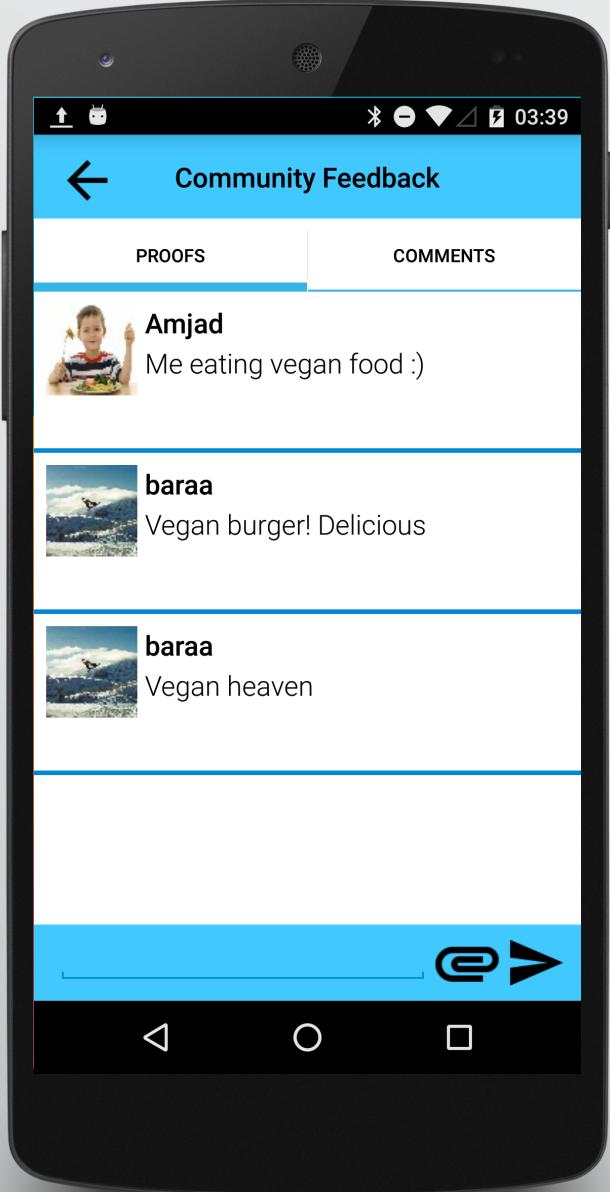
- Title
- Category
- Description
- Deadline
- Attachment
- Keywords

Dare View

Dare includes:

- Title
- Category
- Description
- Deadline
- Attachment
- Keywords



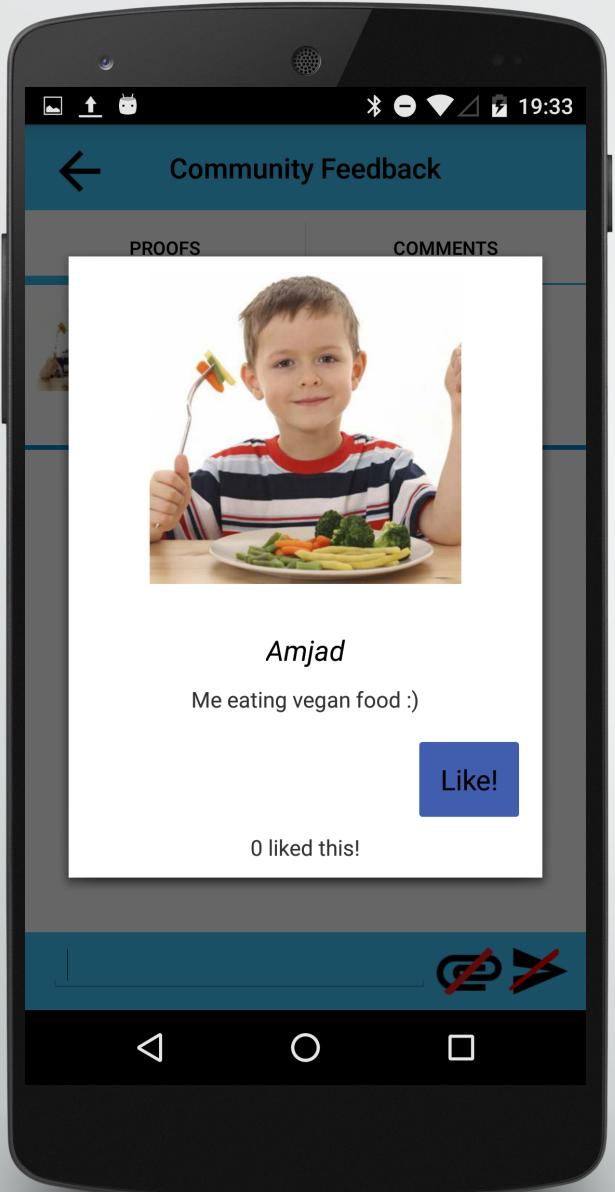


Adding proofs

- Users can participate and post proofs
- Proofs can be liked!

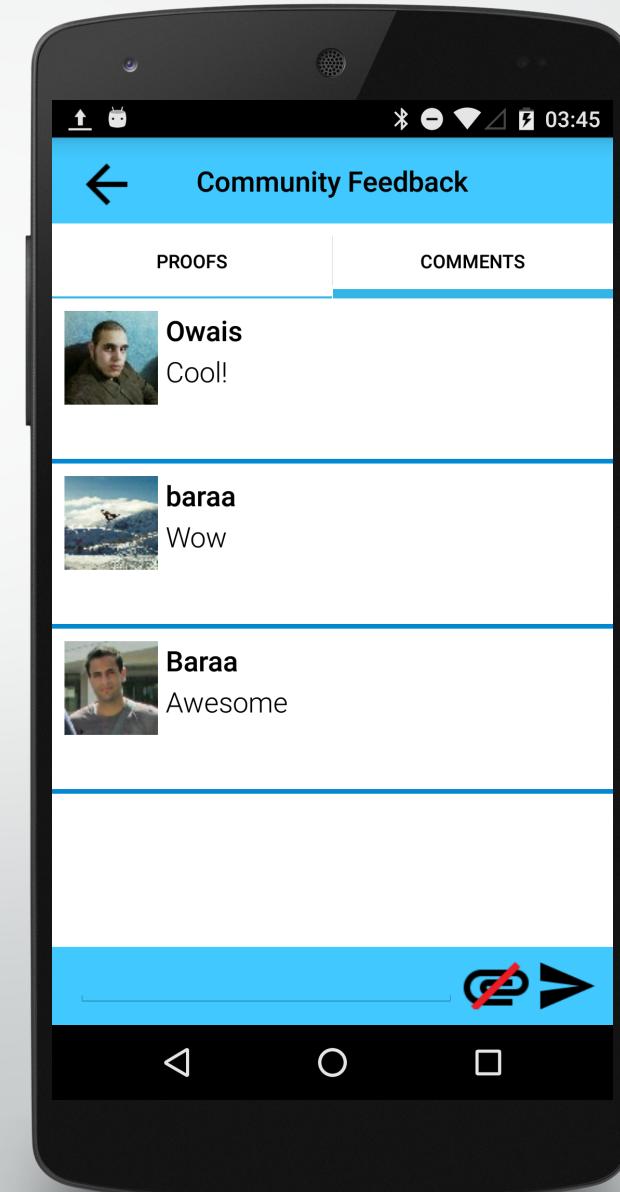
Adding proofs

- Users can participate and post proofs
- Proofs can be liked!

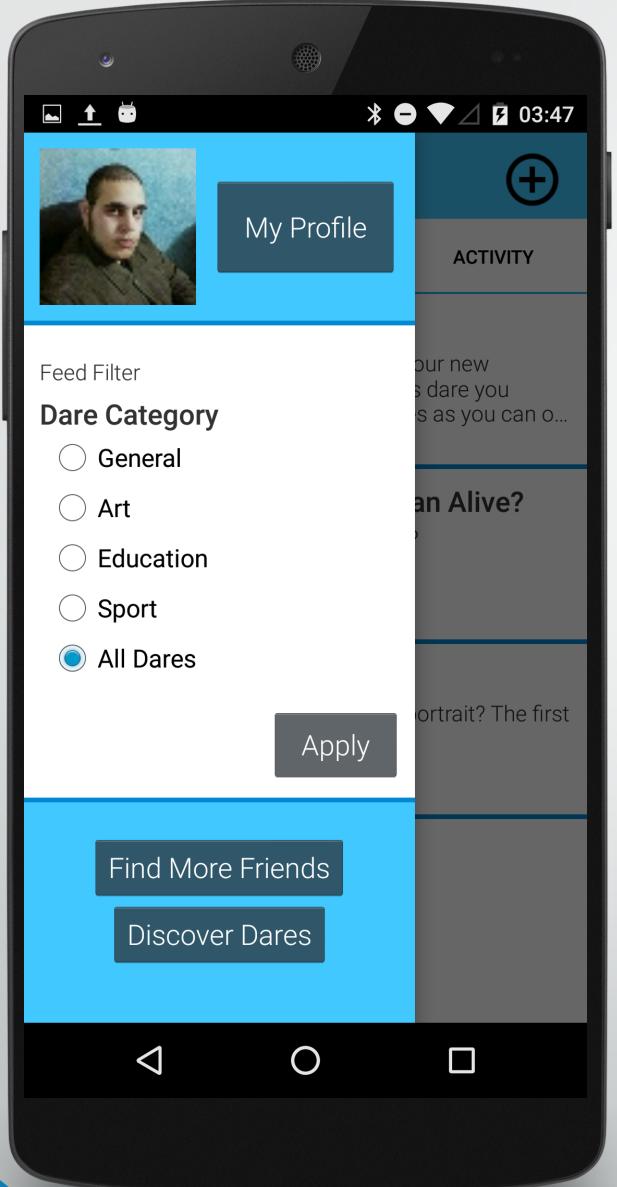


Adding comments

Users can add comments
and express their thoughts

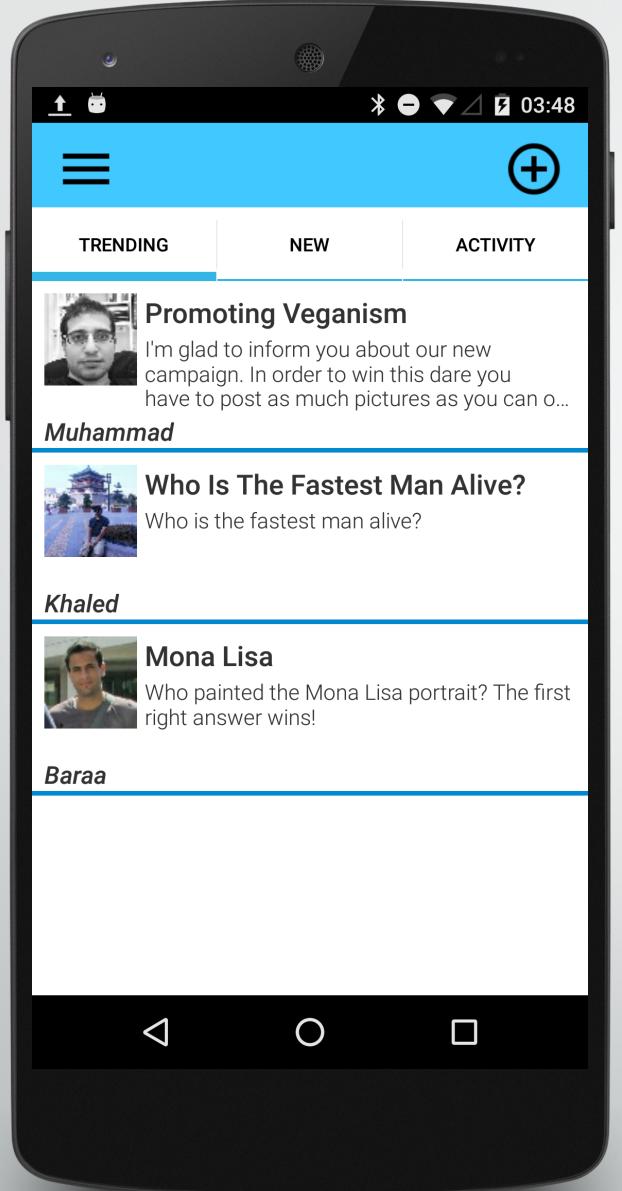


Dares Filtering and Sorting



User can:

- Choose which dares he is interested in
- See trending and new dares
- See latest activity



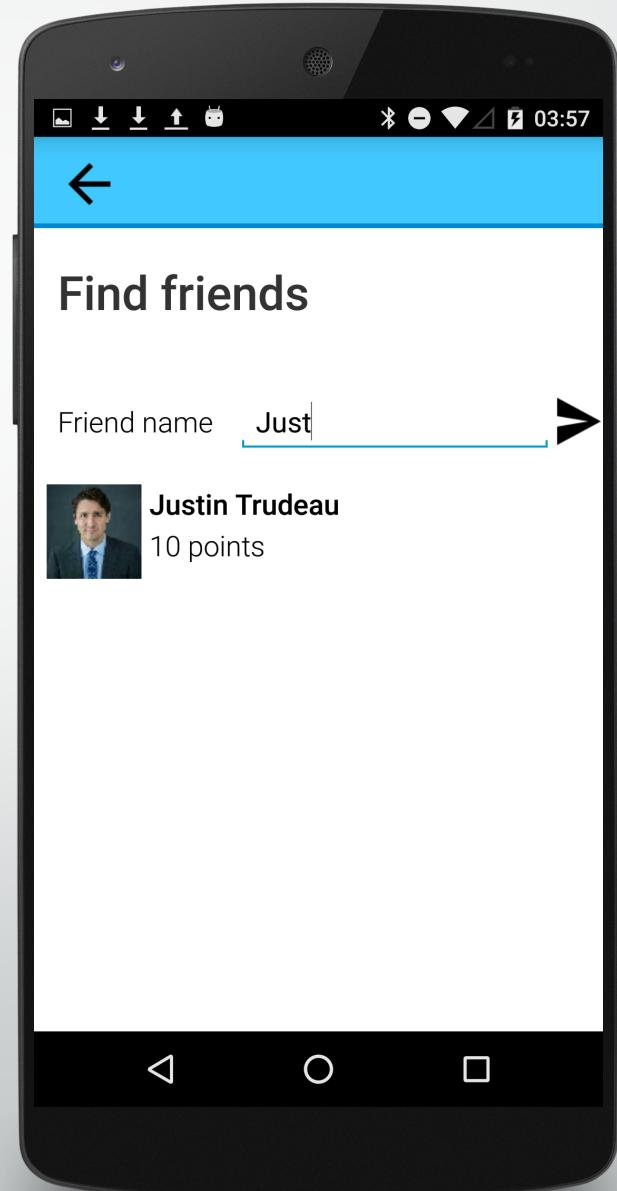
Dares Filtering and Sorting

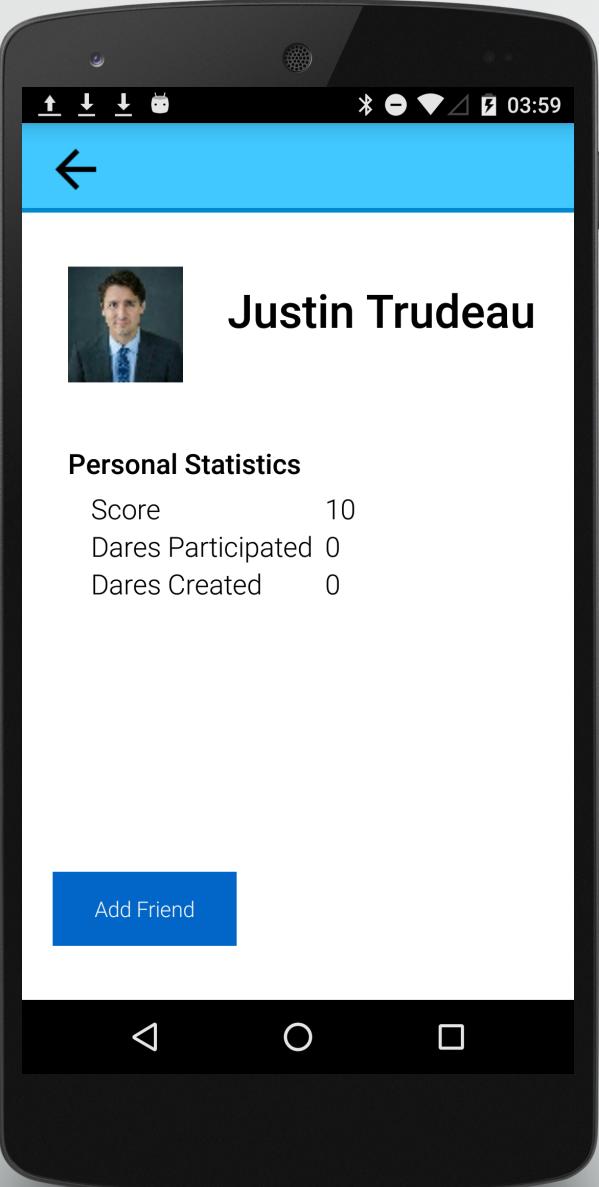
User can:

- Choose which dares he is interested in
- See trending and new dares
- See latest activity

Find Friends

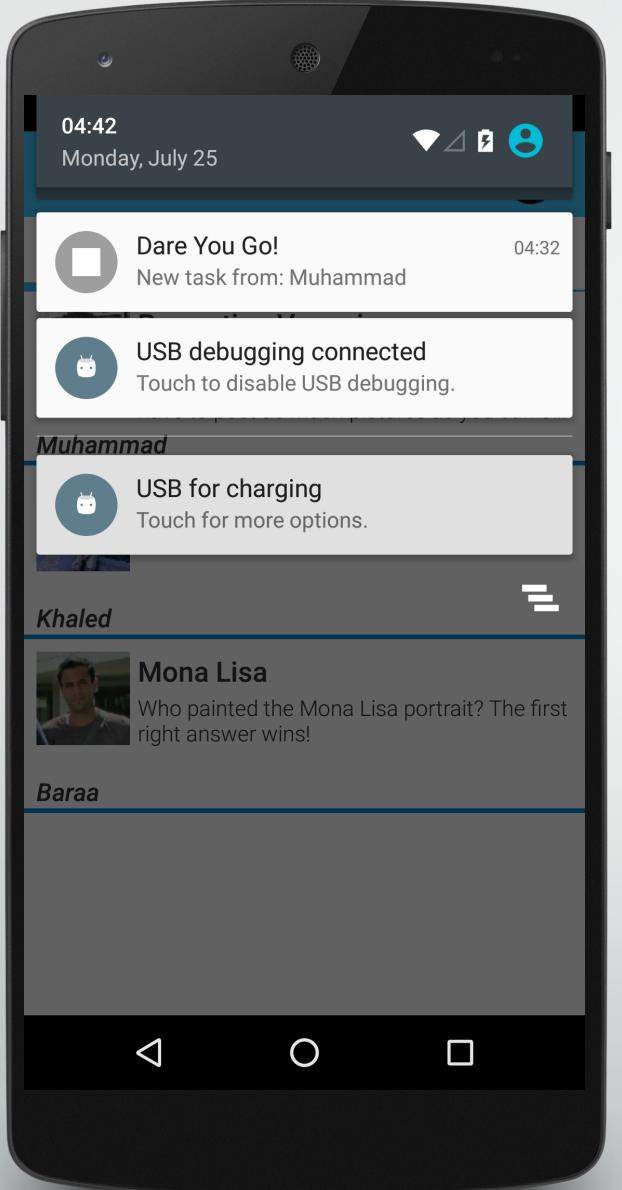
- Users can find other users simply!





Adding friends

- Users can add others as friends:
 - Get updated about new dares
- Users can unfriend other users
 - Will not be updated any more

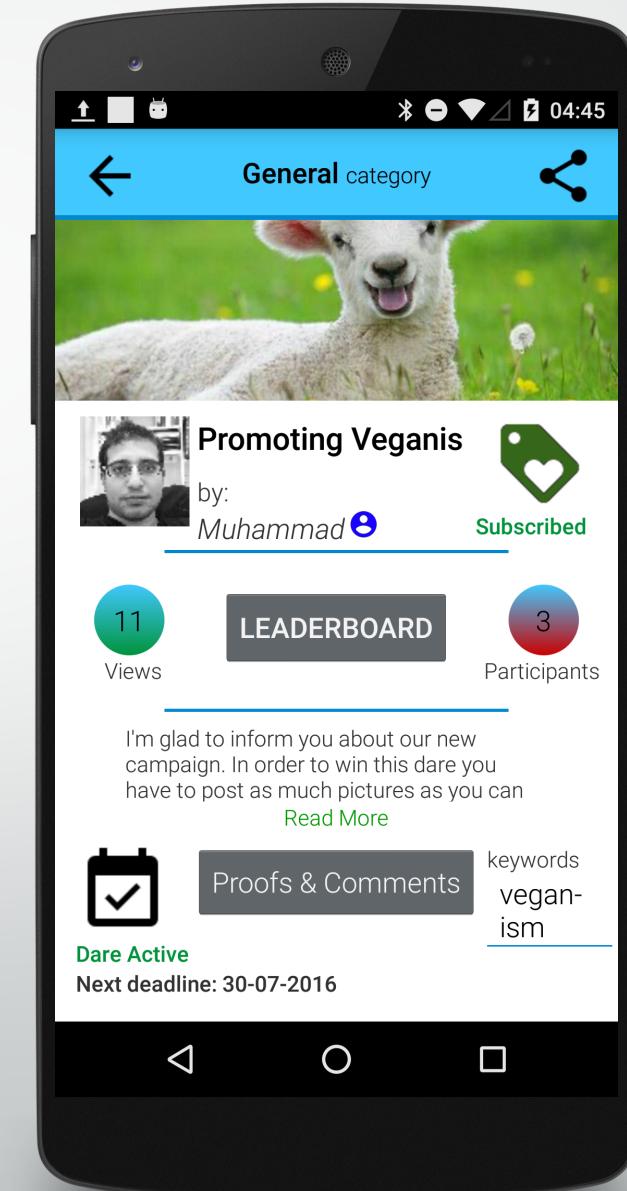


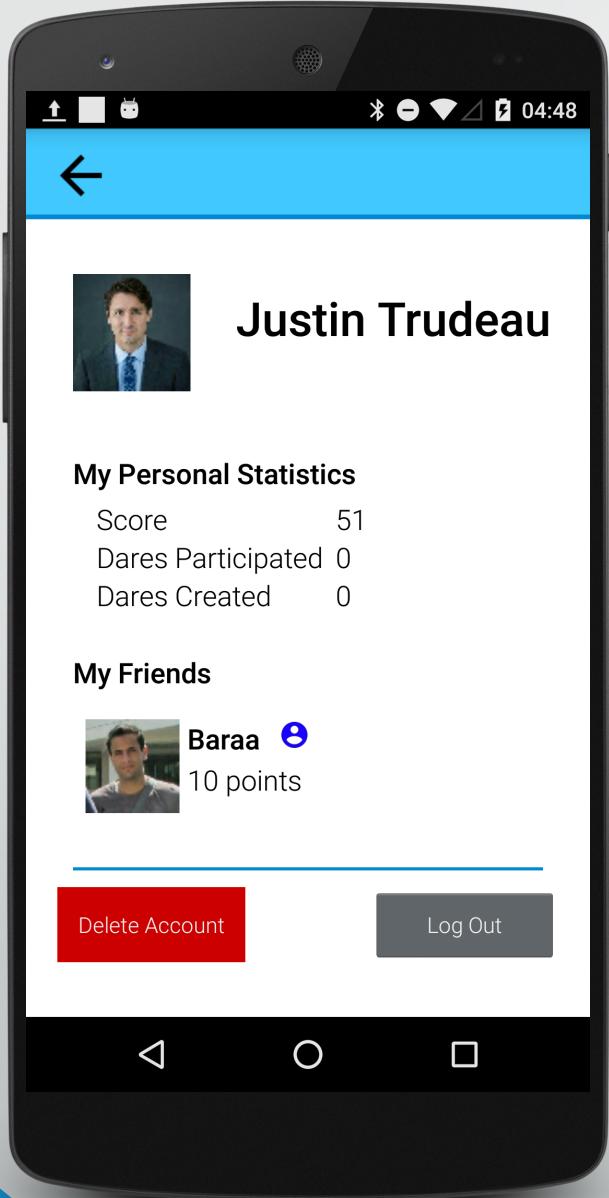
Adding friends

- Users can add others as friends:
 - Get updated about new dares
- Users can unfriend other users
 - Will not be updated any more

Subscribe to Tasks

Tasks which the user is subscribed to appear in the Activity tab.



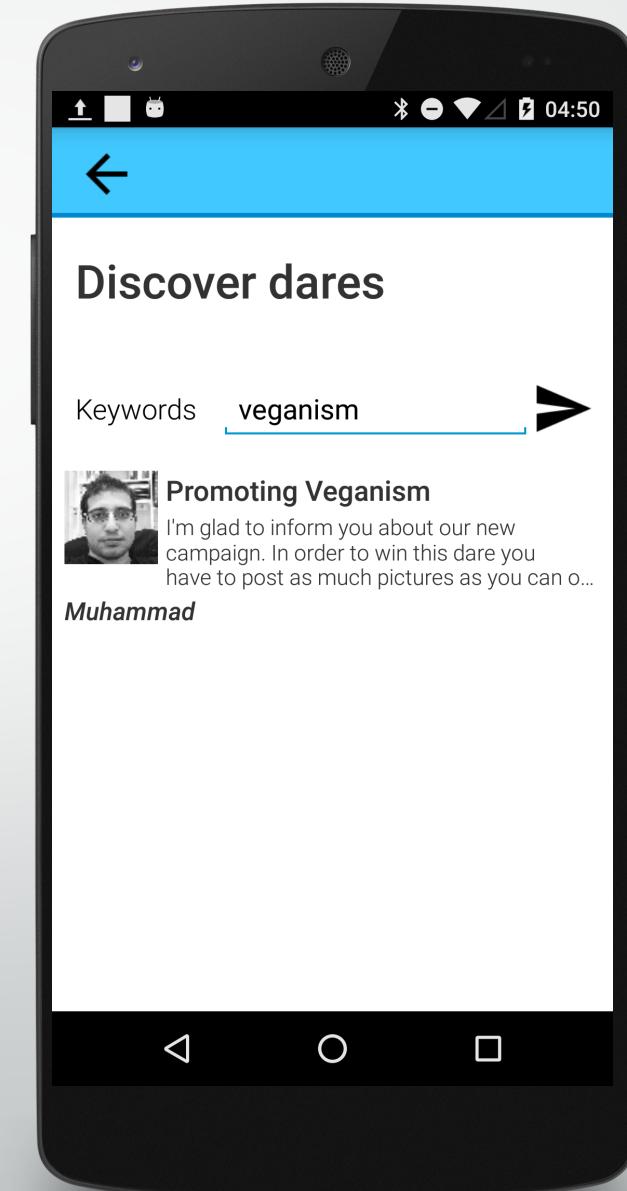


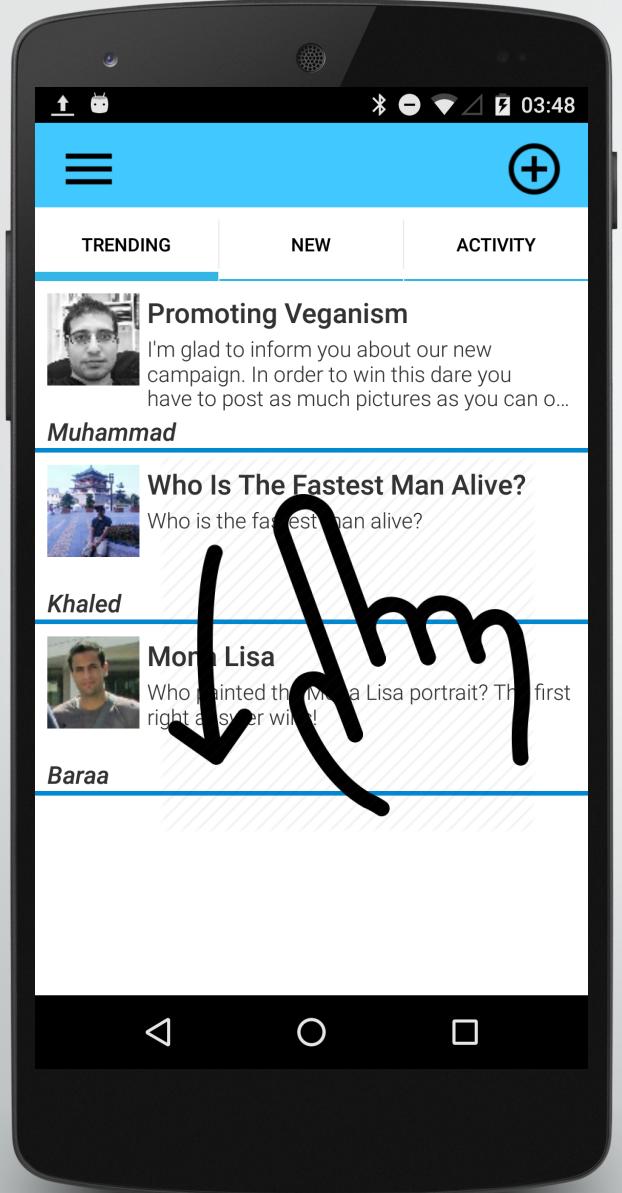
User Profile

Basic information about the user

Keywords

Discover dares using keywords



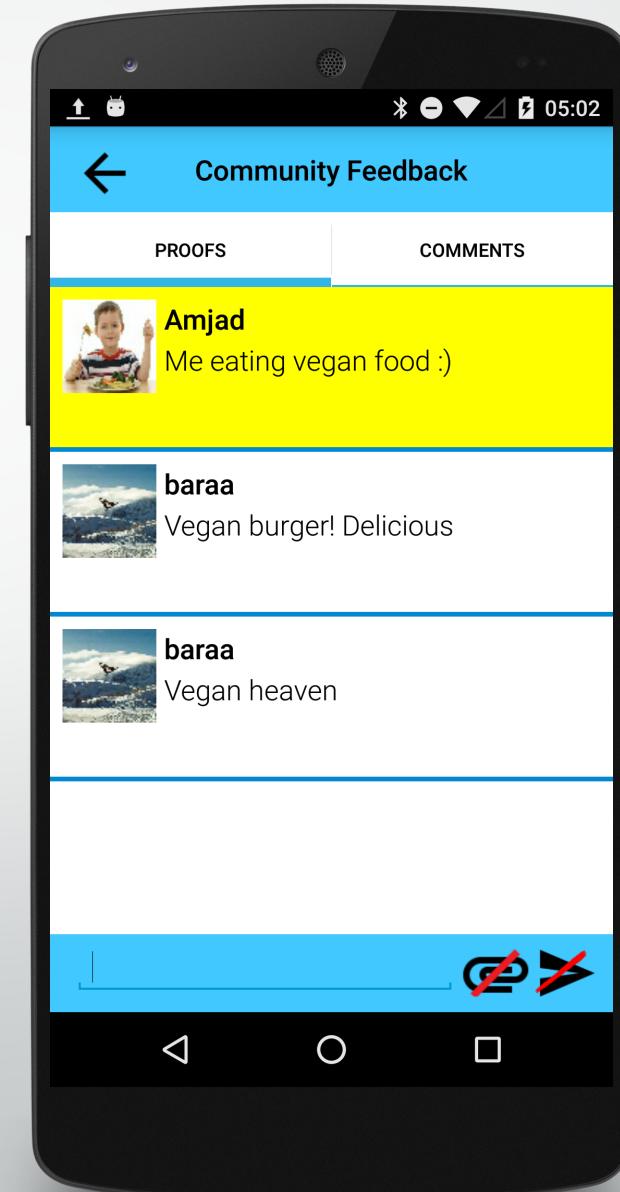


Data updating

- Lists are updated automatically
- Swipe down to load new data

Picking a Winner

- A proof can be picked as a winner by the user who opened the dare



Retrieve dares in batches

- Dares are loaded gradually when needed
- Batches of 10 dares each time

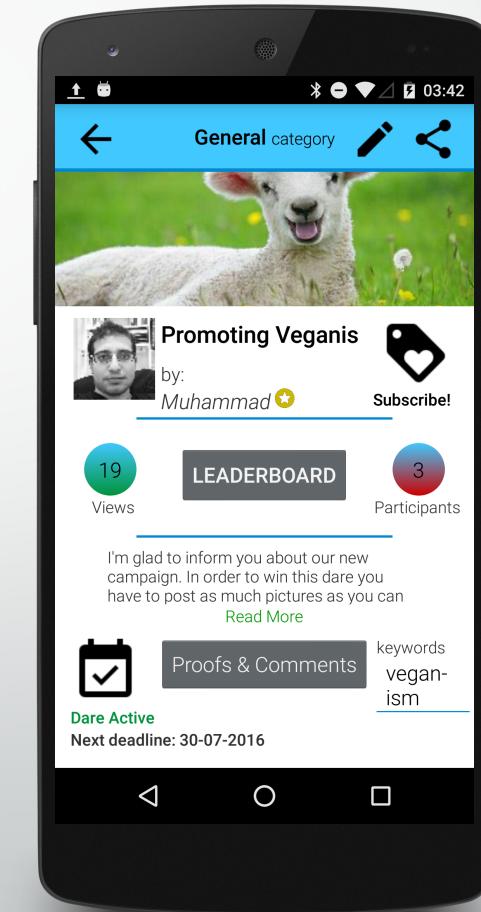
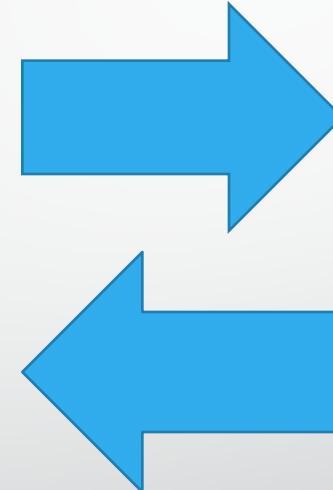
Front-end & User Interface

- Not all set and get!
 - Multithreading for blocking tasks and dealing with their async
 - Optimizations for better UX
 - Multithreading, Compressing user photos being sent to Parse, Resizing photos with max dimension larger than 4k, Resizing profile photos for faster display in lists
 - Overcoming parse file size limitation + out of memory error when dealing with large files
 - using many android components
 - handlers and loopers, drawer , tabs, imageviews , lists , video player, intents for starting activities/fragments and sharing, etc..
 - Creativity and design choices!

Solution Architecture



Server

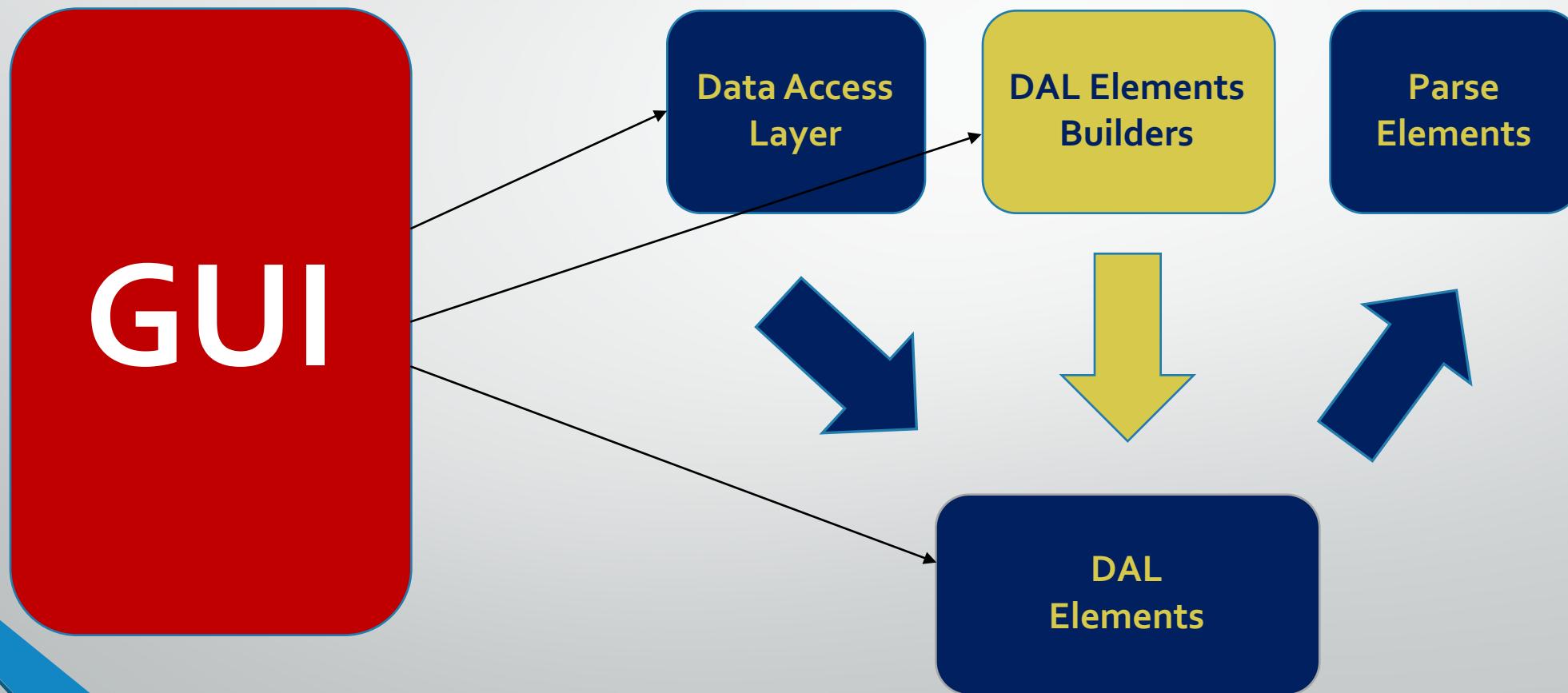


End Users

Design Details - Parse

- Backend solution for mobile application
- Eliminates the need for writing server code or maintaining servers
- Push notifications
- Users management
- Facebook integration
- Asynchronous data management

Design Details – High Level Design



Design Details – High Level Design

GUI



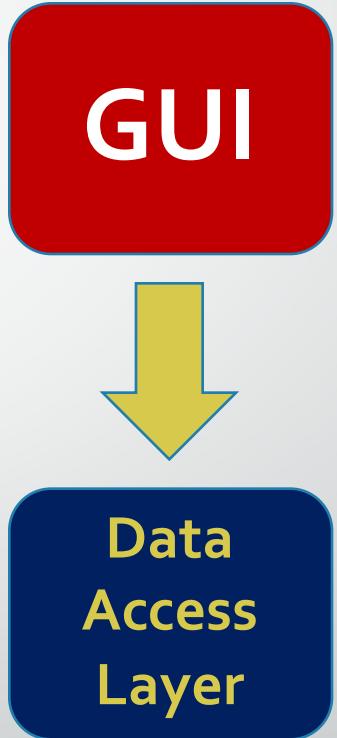
Parse
Elements

Design Details - DAL

- Stands for Data Access Layer.
- A wrapper built on top of Parse.
- Its purpose is to encapsulate the underlying storage and provide statically typed elements for GUI.

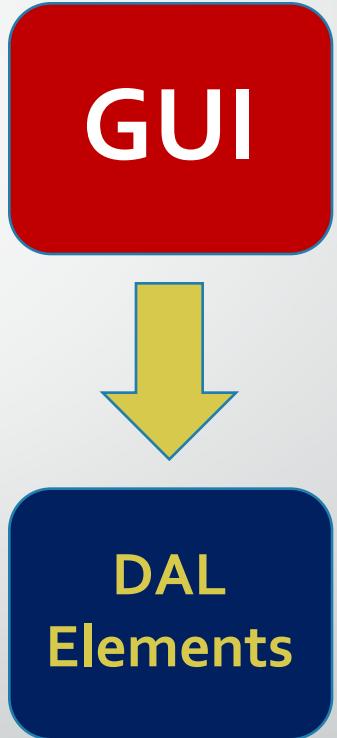
Design Details - DAL

- Single entry point for DAL component.
- Responsible for application scope queries, such as log in, sign up, listening on general events ...
- Distinguish between DEV mode and PRODUCTION mode, as a convenient way to develop without caring about handling real context such as real users.



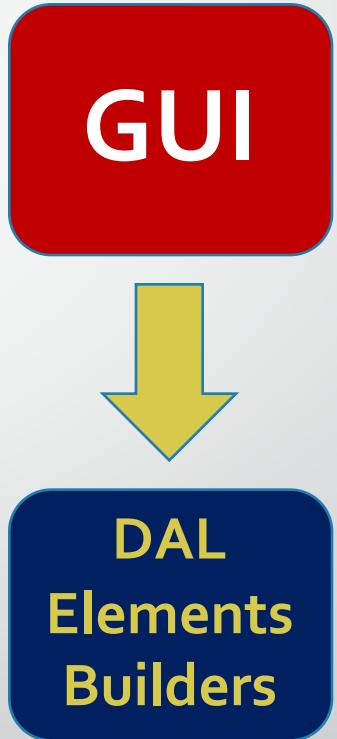
Design Details - DAL

- Statically typed wrappers for parse elements.
- Each DAL element reflects one GUI element.
 - Task, Comment, User ...
- `Task.getTitle()` instead of `parseTask.get("title")`
- You can set callbacks to listen on some events
 - `Task.setOnChangeListener(listener)`
 - Introduce `DalContainer<DAL element>` to listen on events related to a set of elements (e.g. notify me when there is a new task to display)

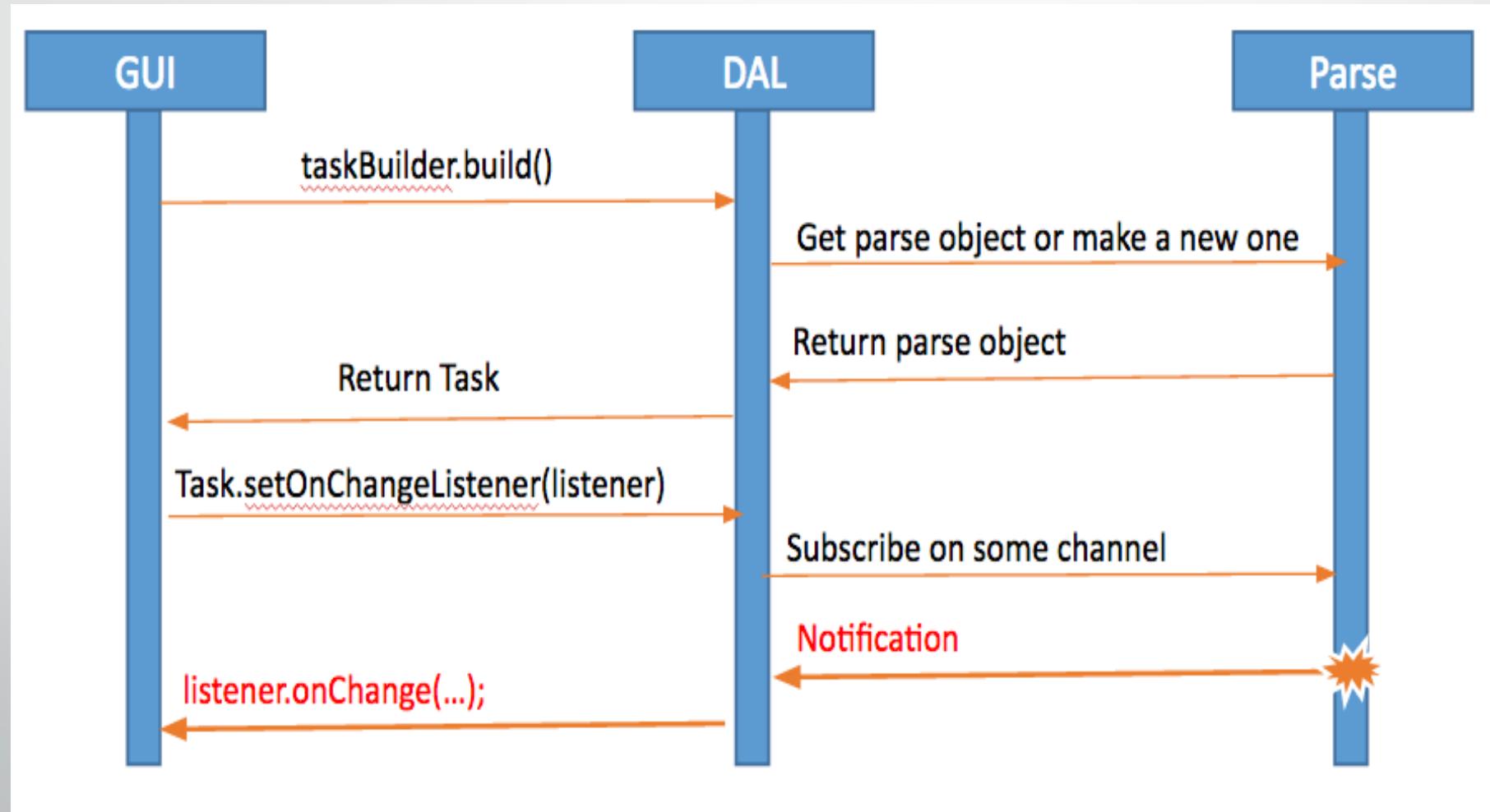


Design Details - DAL

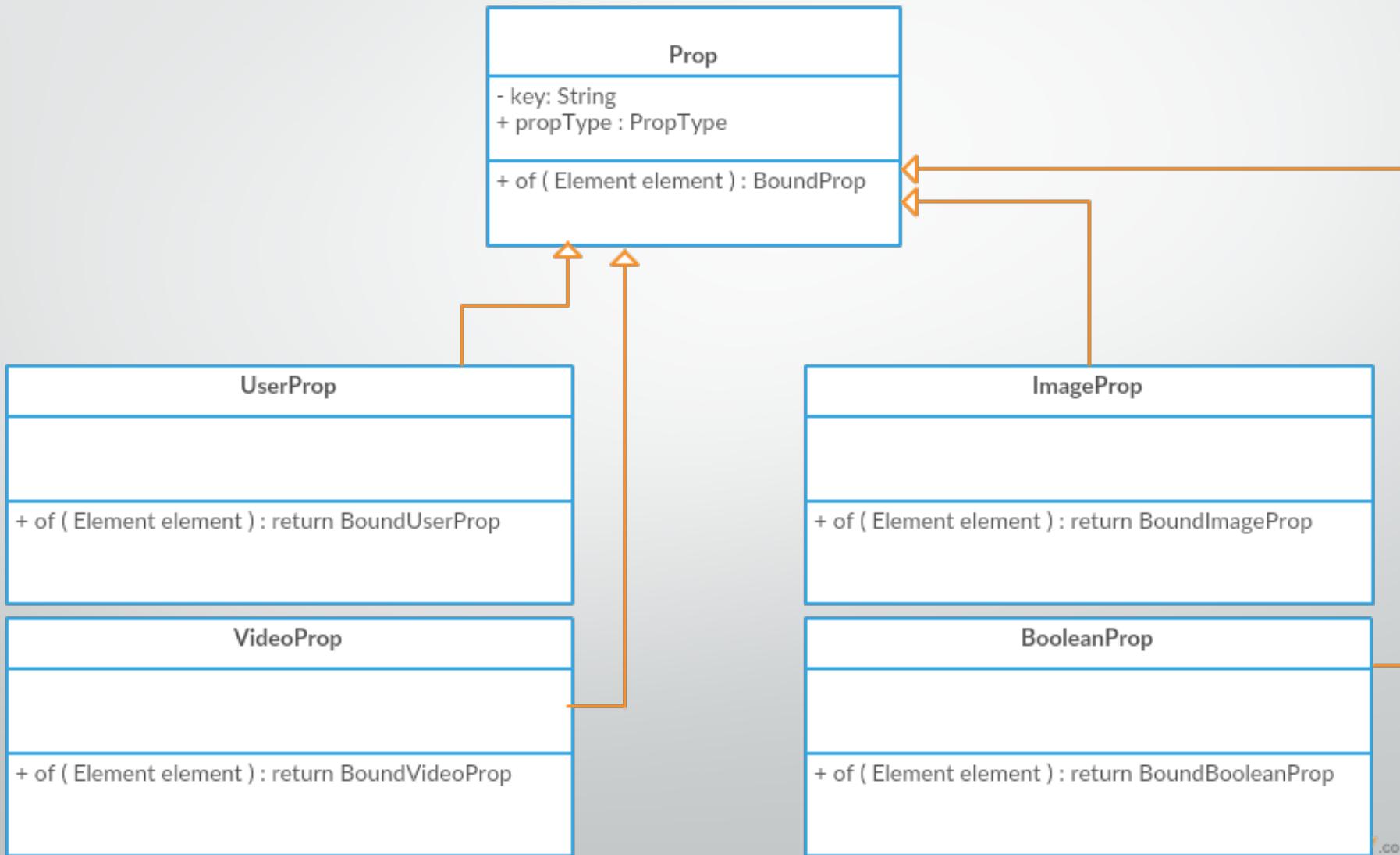
- A convenient way to create new DAL elements.
- Use Builder design pattern with fluent API
 - `new TaskBuilder(user, title, /* other required properties */).setDescription(desc).setContent(cont)./*other optional properties*/.build();`
- Prevent mistakes
 - Task cannot be with no title!
- Hide implementation details
 - Each DAL element needs to hold a Parse element!



Design Details – Notifications

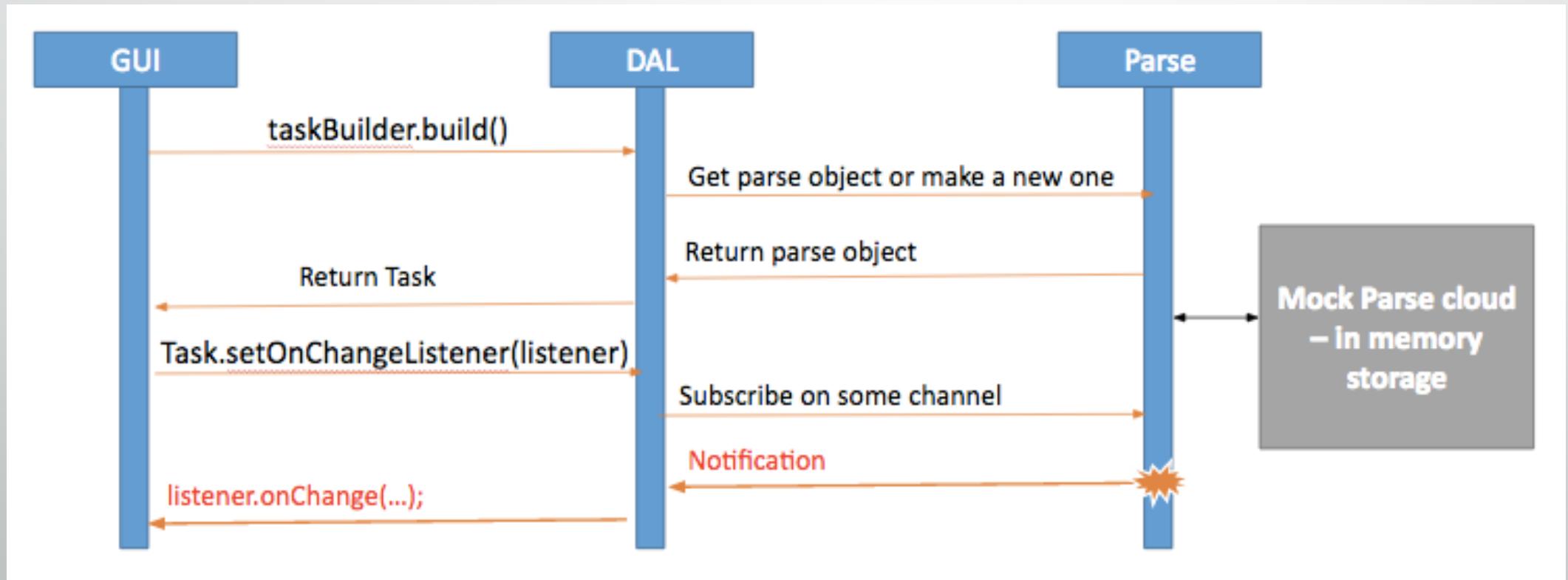


Design Details - Properties

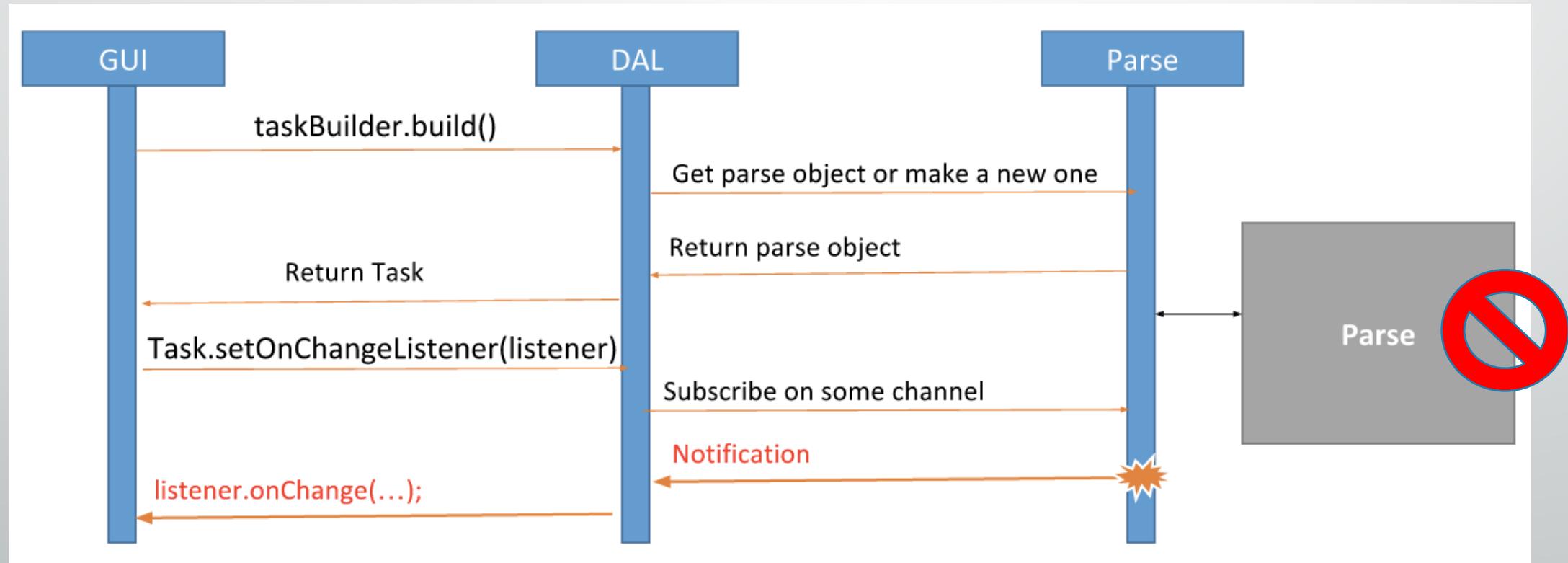


Tests

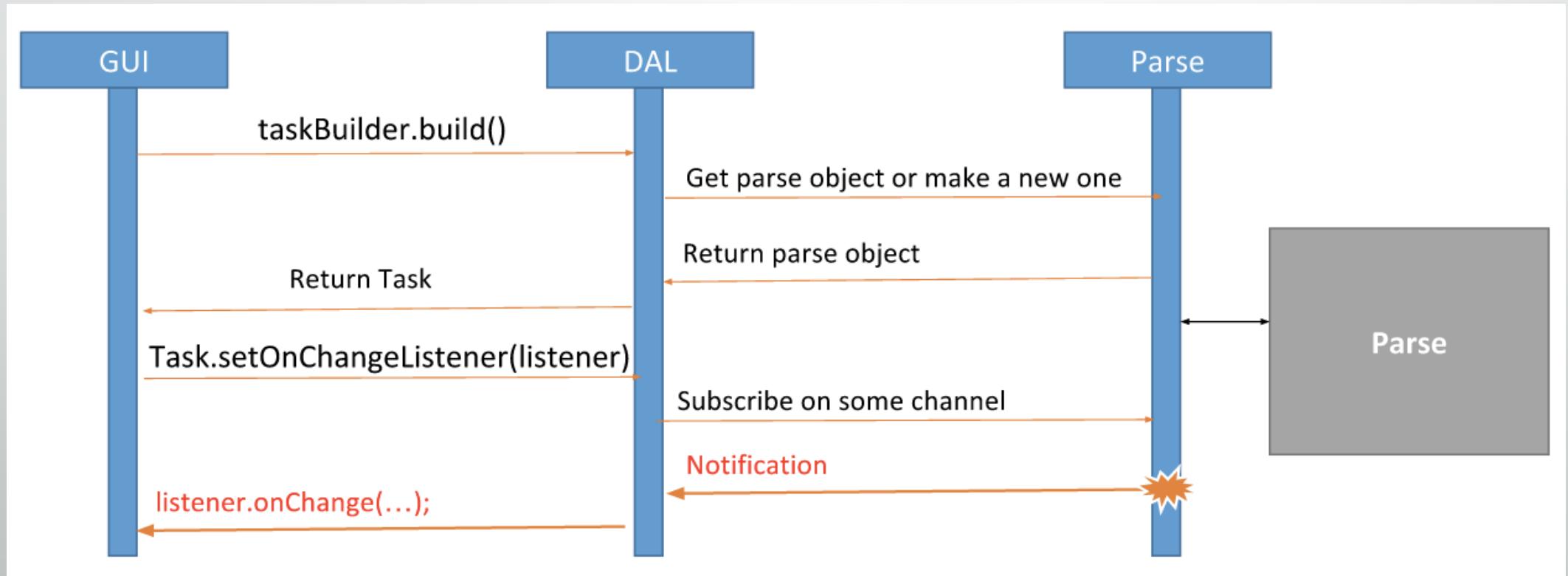
Take 1 – Unit Tests with Mockito



Take 2 – Robolectric



Take 3 – UI-less Integration Tests



Development Environment

- The code was written in Java
- Eclipse
- Version control: Git
 - Used for issues management
 - SourceTree as Git client

The Future

- Find dares which are close to the user
- Database migration
- More integration in Social networks (e.g. sharing on Facebook)
- Web application
- iOS application



Live Demo



Thanks!