

# Bachelor Project Open-Items

- Greetings, thank you all for coming today, Lucas Jung
- Present my bachelor project: Development of an Open-Source and Cross-Platform Client for a Collaborative List Application
- During the presentation try to explain all those words (in blue), for the time being let's call it Open-Items as it is the name of the application

## Plan

- First a deep dive into the actual project ...
- Then a live demonstration of the application
- Then a quick overview of what is around the project (meta) ...

## Project Presentation

### Problem Space

The idea for building this project arises from two observations

- examples: shopping lists, todo lists, films to watch, project ideas, new year resolutions, trips planning, gifts ideas, ...
- Current solutions are not so good
- Class current solutions in 2 fundamental issues
  - Usability / Not easy to use (includes a lot of different things)
    - \* Too many features (calendar integrations, contacts, looking to be the best tool to fit all uses, end up being the right one for none)
    - \* Not available on all platforms (only apple or only android or only on desktop computer and not on mobile devices)
    - \* Not offline first: need to create an account and to have an internet connection, which is not always the case when you need your lists (groceries, underground, outside in the wild, ...), privacy concerns (big corporations) with most sensitive data (questions you need to ask to your doctor, list of medicines you take, things you buy or want, ...) invaluable data in those kind of applications for data brokers and advertisers
  - Proprietary / Closed source Software problems vs. Open-Source (often called Free-Software, free not only as in price but also as in freedom)
    - \* Explain main close source vs open source problems
      - Cost / Freemium / Ads or Subscriptions
      - Security (no data collection)
      - Transparency issues, even if they tell you they don't collect your data, you have no way to check as the code is not public, where as with open source it is (anybody with some kinds of software development background can check)
      - Vendor lock-in (making it intentionally hard to switch app once they are in the system)
      - Actually let me tell you a quick story of how I really decided to start building the solution: check-list application (decent in features and UI but closed source) fully changed their UI with not possibility to fall back to older version (open-source would have solved this with fork)
      - Community contributions, everyone can help and contribute to the code when open-source (add functionalities you want or fix bugs), if disagreements allowed fork the project (create your own version from now on)
      - Self-hosting (server at home)
    - \* Why isn't every Software application Open-Source ? It has one big disadvantage, and it lies on the developer: hard to get funding (donations, B2B). which is a big downside, but I can't think of another disadvantage and certainly not from a user/consumer standpoint

### Open-Items Solution

Open-Items aims to be a solution to all those previous problems, and provide the best tool to store your thoughts.

- Those who know me well know how much I like lists, I have a list for almost everything in my life. That makes me very qualified to identify what heavy list users want and need in order to build the best list application solution possible for this problem.
- Aims to be like a good old pen and paper based list system (means without too many features) while empowering users with technology to allow quality of life features on top of it
  - Like device synchronisation
  - Collaborative lists
  - Easier modification (edits, delete, archive, ...)
  - Better organisation (separate accounts, move lists/sublists, transfer lists between accounts, import and export to JSON commonly used format + no vendor lock-in)
  - Search across all your lists (or specific search)
  - ... And much more
- Goal is to be very easy to use with a simple yet powerful interface
- Offline first
  - No need to have an internet connection to use Open-Items (for basic uses)
- Open-Source (with option to self-host) with all the benefits discussed earlier
- Being Cross-platform and working everywhere
- Link to the full project specifications document to know more details

## Bachelor Project contribution

- First thing was to evaluate deeply the problem space with the different issues that I wanted to solve with this application and the problem there are in the existing solutions
- Then I needed to establish those fully detailed project specifications and define a clear goal for the Bachelor
  - The whole Open-Items project as presented before is way too big to code by 1 person in 1 semester => semester project has the focus on starting building the application only “client” part of the title (no server yet so no data synchronization or collaboration or online accounts) fully offline if you prefer. So client-side only version of the app + design part
  - So the work I did on Open-Items during this semester is all the fundamental work and building the huge basis to grow upon later, it is in fact very technical (except for the designing part) a lot of code was written and 3/4 of my time on this project was about writing the code, we can just take a look at the technical stuff so you can get what I was actually working on during this semester project
  - To achieve this I used the Dart + Flutter (add a few details)
    - \* Data modelling and reactivity system which was probably the hardest part and it looks like that
    - \* Picture of code with data modeling interactions. “So I don’t expect any of you to understand any of what is show in this code :)”
    - \* Diagram of the code models interactions. “This is the diagram representation of the data modeling classes.”
    - \* “However, the goal of this presentation is to give a higher level view of the whole project rather than spending time to explain and details what is under the hood.”
    - \* “Those were just interesting to see how things are really done behind the scenes and it can give a glimpse of what I spent my time on to get to the result you will in a few minutes.”

## Demonstration

I will now show the result of the work I did this semester. I will present the WEB version but this application (as it is built using Flutter) can be easily made available for all major platforms (android, apple, windows, linux, ...)

- Can observe and talk about the design phase of the project while I show the app, “every little design detail was a long reflexion” (colors, font, pop-ups, ...)
- First things first, an account in the offline-first and self-hosting world is not exactly what everybody is used to with regular proprietary applications (like social media or online shopping solutions)
- Offline first implies there is a difference between online and offline accounts
  - An offline account means that no data is saved anywhere else than on the device it was created on, benefits like fast (no need to confirm email or even communicate any additional information, just give it a name and go), no spy / reinforced privacy); disadvantage (no backup, no synchronisation, ...)
  - An online account is similar to what everybody is used to on their application meaning they can log in

- from different devices and have their data synced
  - The key point is that the user is given the CHOICE to chose what they want to use (between an offline and online account)
  - They can have multiple accounts of different types and decide on a list per list basis of what list they want synced and what list they want to keep private for themselves on their own device
  - Users can also change their mind and transfer their list from account (offline) to account (online) whenever they want
  - For the moment, as said earlier only offline account will work as the server code is not yet written
- Explain forms validation
- Explain infinite list nesting

## Challenges Encountered

- Cross platform was a hard thing to maintain, everything should work everywhere is not at all granted when building software (platform have different requirements and ways to work that are vastly different)
  - Especially with database and how to store and persist data when you close the application (database cross platform work)
- Model establishment was probably the hardest thing in the project so far as it is hard to establish the correct data structures and relationships between components such that everything works as intended
- I also tried to think about the future server side integration and built everything around that so that when I start developing the server side code it can integrate nicely with the already available application and not have to re-write everything to include server
- Data validation was also a topic where I struggled a little, it is always hard in software to validate data that a user provides as you have to assume the user can and will provide anything and random stuff and you need to point them toward the correct format that your application is expecting (as we saw in demo) it is also hard to provide a clean UI to explain to the user what they did wrong (that is the reason why most only forms that you have to fill in are ugly and frustrating)
- Initially I intended to fit even more features about the Open-Items client application during this bachelor project.
  - I underestimated a bit the amount of work that was required to build all the fundamental basis to grow upon.
  - I had already dedicated an enormous amount of time to this project (honestly probably more than I should have) so I could not integrate more features during this semester.
  - However, now that all of this is coded and ready, it will be much easier to add functionalities in the future (on top of that everything is ready to adopt new features).

## Statistics

Let's comment a few numbers about the project.

- The programming part represents about 7'000 lines of code spanned across 96 source code files
- I contributed a total of 87 commits on the GitHub repository
- As it's very rare to write the correct code on the first try there is an actual total of 20'000 lines that were changed in the code base across the duration of the semester.
- All of this work represents about 230 hours of work.
- There were a total of 14 people that tested the application in its current state to try it out and provide feedback
- I also wrote a more technical document of 38 pages that explains and details the architectural choices behind the development.

## Future of Open-Items

- I will carry on the development of Open-Items in my free-time or maybe in a future semester project at EPFL
- Items custom ordering
- Import / Export functionalities
- Search

- (other client related features)
- Server side implementation, huge amount of work
  - Enables Device sync
  - Collaborative lists
- I really think that the application has a bright future ahead and hope it will be useful to people looking to this kind of solution.

## Conclusion

Key takeaways of the presentation:

- Bachelor project allowed me to work and produce the first working version of Open-Items
- I really enjoyed coding the app
- Designed to be efficient
- Follows the Open-Source philosophy and will be available on most platforms and devices
- New features are to be expected in the coming years

learn more about the project on the GitHub repository.

## End

Find this presentation online to go over parts that I might have gone too fast over or to follow links to the parts you were interested in

## Thanks

- Thank Dr. Jamila Sam and Dr. Barbara Jobstmann for making this project possible by supervising my work during the semester
- Thank audience for their time/attention (and questions)
- Now I would be happy to answer any questions you might have about the projet or the presentation !