
GitHub Username: https://github.com/danielrust

NoZeroDays

Description

I don't like To-do apps. I don't like most productivity apps. I have found that the best way for me to accomplish many things at once is to take small bites out of a goal; the best way to do that is just simply do a little bit each day. The worst scenario is to take a zero day on a goal…. That is, don't do anything. This breaks habits and pushes the goal out of our mind further and further. This app allows the user to list their long term goals that they would like to work towards. Throughout the day, they can indicate that they did work towards that goal, and therefore not take a zero day. The app keeps track of streaks when users continually work towards goals and not take zero days.For instance, if one of your goals was to read a certain book, you would indicate in the app that you wanted to "read certain book"; the idea is that you could at least just read one page to avoid taking a zero day. If you wanted to exercise or lose weight, you could at least just work out 10 mins. I find that if I have a way to be accountable, at the end of the day I could look and see if there was something that I could do, however minimal, so that I would not have to take a zero day on a goal. Just do SOMETHING. Doesn't have to be big.

Intended User

The intended user is anyone who feels like there is no kind of scheduling or productivity app that helps them out. We all have goals or extra things we'd like to do in life, but its hard to keep track of everything. This is for those people who feel most comfortable accomplishing bite size chunks of goals on a daily basis, as a way of developing good lasting habits.

Features

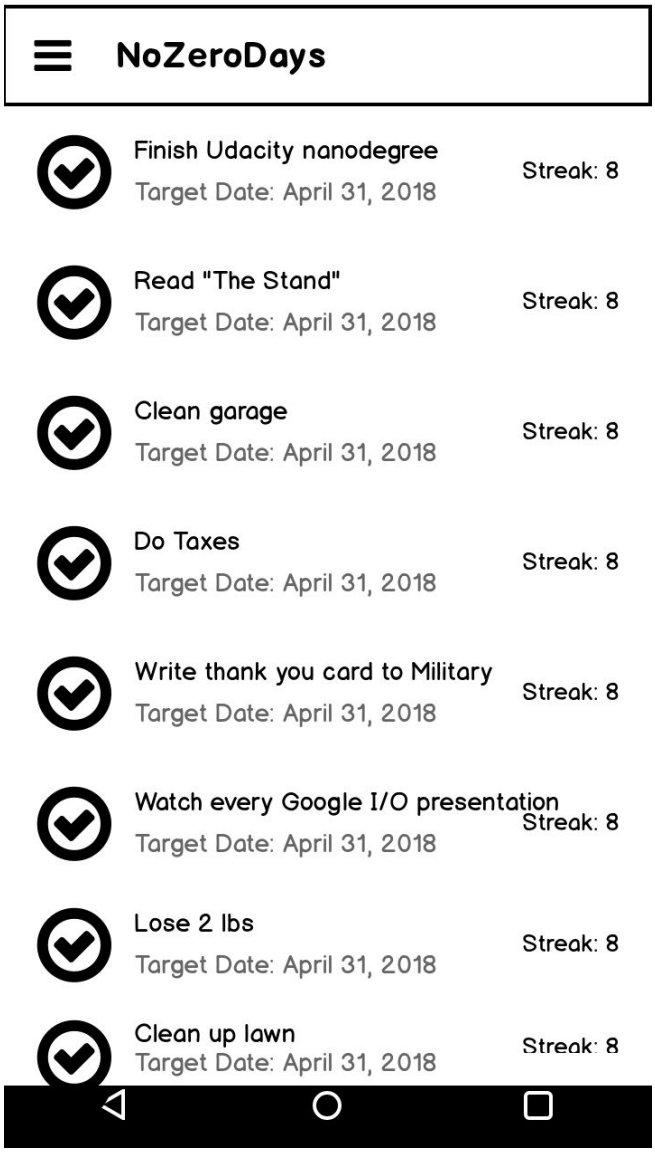
This app will allow users to:

- Add goals in list format to work towards
- Indicate that they have accomplished their goal for that day
- See their progress on a goal (see their streak of nozerodays taken)
- Be reminded by notifications of their goals and not to take a no zero day

User Interface Mocks

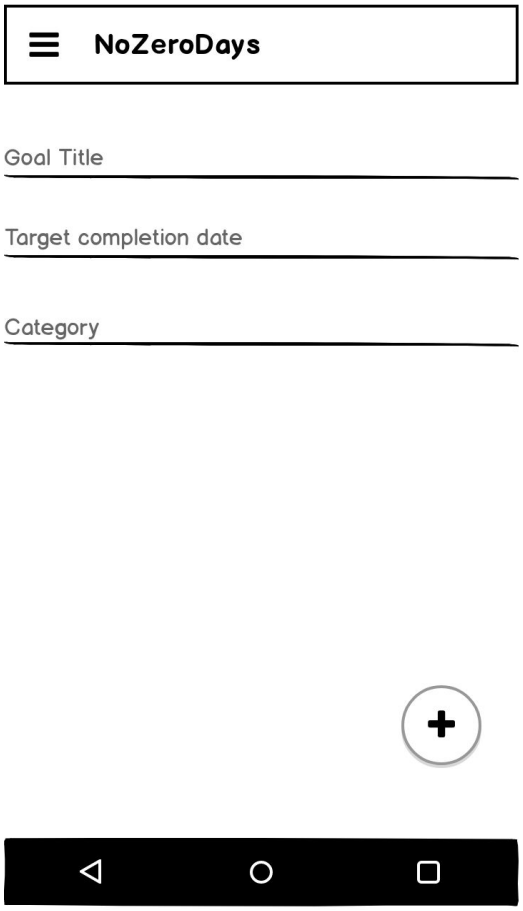
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1



This is the main activity that the user will see upon first opening the application. This is the itemized list of all the users nozeroday goals. The icon on the left will be greyed out until the user taps on it indicating they have worked towards that goal that day. Upon tapping, the icon will turn to green. The streak will also increment. The user is able to long-touch/tap on each item to edit them if needed. There will be a FAB allowing the users to add more goals as needed. Swiping the item to the left will complete the item, swiping to right will give option to delete it.

Screen 2



This is where the user will add a goal to complete. The user inputs the title, the target completion date (via the native time/date picker), and then the self-created category this goal applies to, if wanted.

Key Considerations

How will your app handle data persistence?

This app will be use both Firebase and content provider/sqlite

Describe any edge or corner cases in the UX.

This app flow is fairly simple. The only navigation tricks needed to be accounted for would be to access the app from a push notification.

Describe any libraries you'll be using and share your reasoning for including them.

I will probably use threetenbp to handle the time formatting and storing. I will also use Butterknife or the native databinding libraries to handle view binding. Should any REST calls be necessary, I will use Retrofit to simplify that. And of course, I will leverage Dagger 2 for dependency injection.

Describe how you will implement Google Play Services or other external services.

The two google play services that will be used by this app are Firebase Authentication and Cloud Messaging. Should the user desire notifications, I will use firebase's push notification service. The user will also be able to login with several different authentication options using firebase.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

First, I will need to setup the content provider or firebase data persistence service to store the user's goal information. This would include mapping out the data model for the goals to be stored. Once this service is set up, I will create the activity to insert the goal into the database. Upon success, I will then create the main UI that will pull on the goals from the database and display in a recyclerview. I will implement Dagger with it's associated components early on in the process as well.

Task 2: Implement UI for Each Activity and Fragment

- The main activity will utilize a linear layout oriented recycler view. I will create this view and use databinding to retrieve goals from the repository.
- I will then create the add goal activity, which will take the user-input information and store it in the repository
- I will create another activity that will show all the inactive goals that have been deleted or completed

- The mainactivity will feature a mechanism to long-touch on a goal to edit it. This will take the user to a “Edit goal” activity.
- I will create a Settings activity with preferences regarding notifications and such.

Task 3: Implement notifcations

- Using firebase, I will create a system that will push notifications (if requested by the user) that will remind the user to avoid taking a zero day on a goal.This will fire at a time to be decided by the user.
- Create associated intents to allow user to access app easily from notifications.

Task 4: Setup search

- Using content provider, I will provide a search autocomplete feature that will allow the user to easily type and search their previously made goals. This will be down using an asynctask to easily communicate with the repository

---

Submission Instructions

- After you’ve completed all the sections, download this document as a PDF [File → Download as PDF]
    - Make sure the PDF is named “**Capstone_Stage1.pdf**”
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
- Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”