**GitHub Username**: ishan1604

# NotifShredder

## Description

Since the number of apps on your phone are growing day by day the number of notifications from all the sources are growing day by day. It becomes very mundane a task to swipe away the unwanted notifications when a lot of them are there. It could be from your SMS app about marketing/sales messages, it could be from your GMAIL regarding promotional emails etc., it could be from AMAZON regarding new offers etc. and the list goes on.

So the problem I identified was that each and every app has the capability of sending notifications that might be irrelevant for a user.

NotifShredder will come to the rescue by auto dismissing the notifications that you are most likely to. It will train itself with the help of your usage patterns. It will allow users to also customize their experience about which apps they never want NotifShredder to mingle with and which apps they always want it dismiss the notifications from.

## Intended User

Because it is a general life/productivity hack it can be used by anyone who owns an Android phone. Targeting all the users in the world who ever bought an android and will be buying an android.

## Features

The main features of my app:

- Allows people to choose which apps to permanently block notifications from.
- Auto dismisses the notifications on user's behalf for the apps selected above.
- It also trains itself to dismiss notifications on users behalf on the basis of training data that it receives. For Example: If user constantly dismisses the notifications for Promotional SMSes and Emails after a certain threshold training inputs we'll automatically do that for him.

## 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 Photoshop or Balsamiq.

**HOME SCREEN**

- **On boarding slides**
- **Google+ Signin**

**ZERO-DAY PERSONALISATION SCREEN**

- **Show a list of type of notifications that may be annoying for a user.**
- **Ask user to select 1 or more from that list.**
- **Use that as your initial training data to take decisions on behalf of the user.**

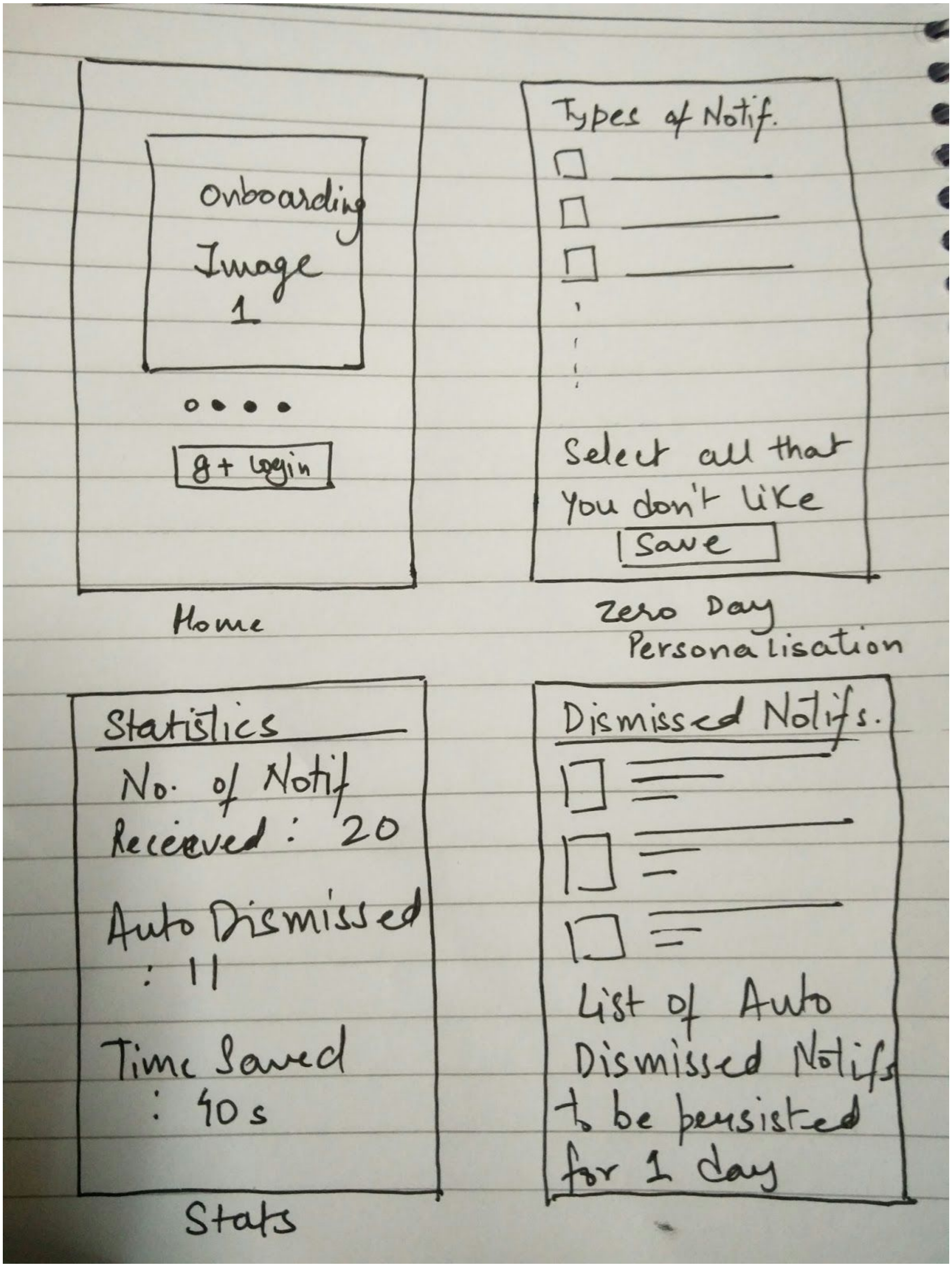
**STATS SCREEN**

- **Show user how many notifications he received today.**
- **How many were auto dismissed.**
- **How much time we saved for him today and in lifetime etc.**

**DISMISSED NOTIFICATIONS LIST**

- **We will save the list of notifications dismissed by our app in the last 24 hours.**
- **So that any time a user wants to go through that he can do that, so actually he never misses the notification and can trust the app.**

- **He can also tell the app if a notification that was auto dismissed by us, was an important one so that we can update our predictor for him.**



# Key Considerations

### How will your app handle data persistence?

I will build my own content provider to persist and query the list of notifications that arrive on the phone.

### Describe any corner cases in the UX.

Cannot think of any corner case at the moment. If anything comes to my mind during the development phase I'd readily handle that.

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

I will be using Picasso for showing images, Parse for persisting some data online, Google Analytics and Google+ Signin dependencies, Gson for Json Parsing

# Next Steps: Required Tasks

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

## Task 1: Project Setup

- Project setup
- Initialize the build.gradle with dependencies required
- sync all repositories
- create a signing key

## Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for Home Screen
- Build UI for Zero Day Personalisation Screen
- Build UI for Statistics Screen

- Build UI for Dismissed Notifications Screen

## Task 3: Complete Home Screen

- Build On boarding View Pager
- Implement google+ login
- Persist auth token after successful login

## Task 4: Build Personalisation Screen

Create a listview that shows some predefined data set for a user to choose from. Persist the selected values.

## Task 5: Implement Notification Dismissal Algorithm

A service will run in the background that will keep on listening for notifications as and when they arrive it will take action accordingly based on the algorithm supplied to dismiss or keep a notification and the training data stored in the app.