P3 Defence

Optimizing Smartphone Interruptions through a Smart Notification System

Vivek Paul Joseph
IxD, IDC

Guide:
Prof. Jayesh Pillai

P3 Defence • Now

Introduction

Research Problem

P3 Defence • In 1 min

Primary/Secondary Research

CLEAR ALL

Introduction

- Smartphone notifications are a major source of interruptions

Image Source: https://goo.gl/images/ipFe7B

Introduction

- Notifications popping up during inopportune moments break the flow of work

Image Source: https://goo.gl/images/ipFe7B

Interruption Payoff Scale

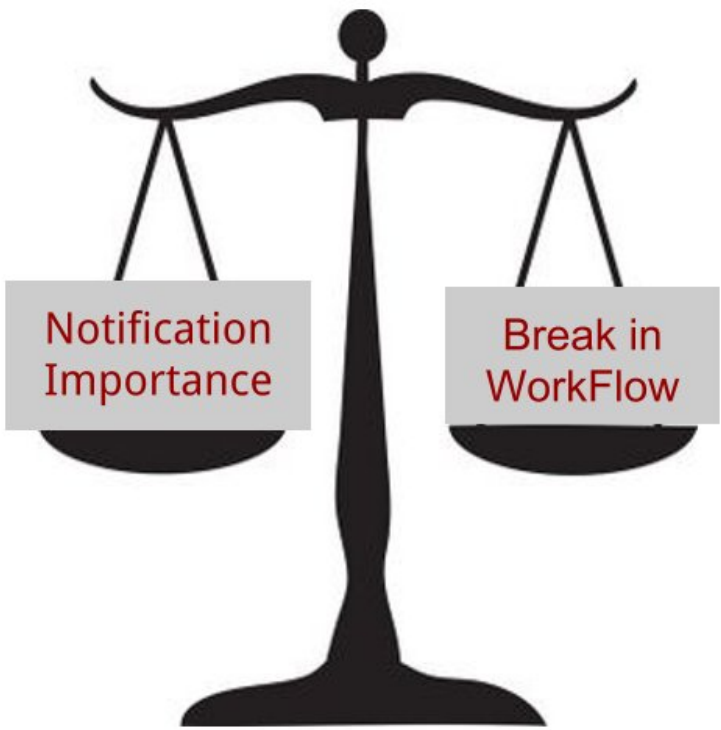


Image Source:
http://www.clipartkid.com/images/5/scales-clip-art-V7SKqo-clipart.jpg
https://goo.gl/images/HeKvv8

Interruption Payoff Scale

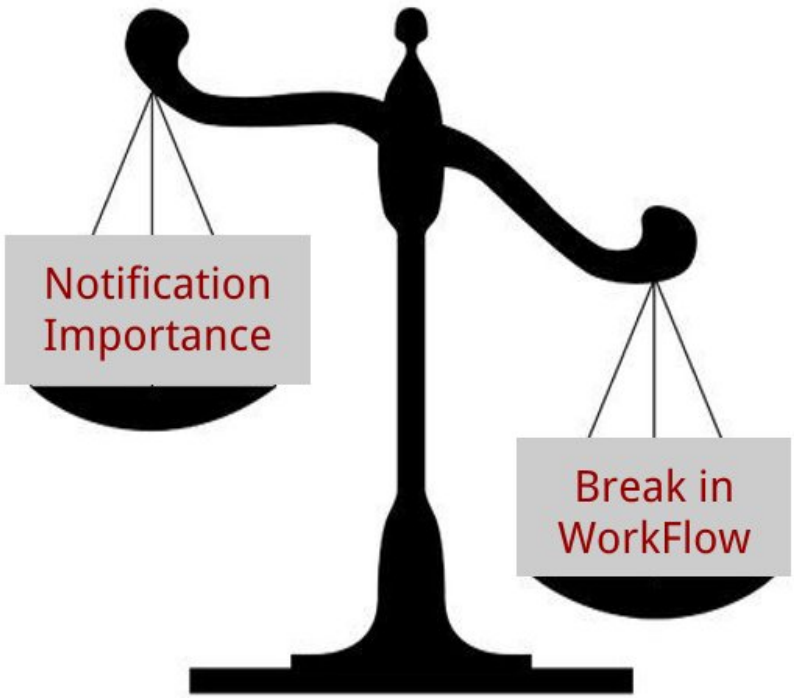


Image Source: http://cliparts.co/cliparts/8i6/oBp/8i6oBppLT.jpg
https://goo.gl/images/HeKvv8

Objective

Reduce the number of interruptions generated by smartphones

Objective

Reduce the number of interruptions generated by smartphones

Through

Objective

Reduce the number of interruptions generated by smartphones

Through

A personalised smart notification management system that learns from user behaviour and screens low priority interruptions

Approach: Preliminary Research

- Unstructured interviews with users
  - Is there a problem?

Approach: Secondary Research

- Reviewed Existing methods to control notifications
- Reviewed Research Papers on:
  - Disruptive effects of smartphone notifications
  - Interruption Management for smartphones

Monday, May 1 2017

P3 Defence • Now

Primary Research

Interviews & Data Collection

P3 Defence • In 5 mins

Design Implications

CLEAR ALL

Primary Research: Interviews

- Semi-structured interviews:
  - Users' subjective opinion of how many notifications they receive daily
  - Users' subjective opinion of whether notifications seem intrusive
  - How they manage notifications?

Primary Research: Data Collection

- Platform in scope: Android

Primary Research: Data Collection

- Input required for Notification Manager Algorithm
  - Parameters, whose values can help predict relevance of notification
  - Pre-classified data from users (Training data)

Primary Research: Selection of Parameters

- Preliminary list based on what can be measured by android

Primary Research: Selection of Parameters

- Preliminary list based on what can be measured by android
  (and could help predict relevance of the notification)

Primary Research: Selection of Parameters

- Preliminary list based on what can be measured by android

- App/Package Name
- Date of receipt
- Time of Receipt
- Time of Consumption
- Priority of message
- Text Content of the notification
- Title of notification
- User Activity on receipt of notification
- User Activity on consumption of notification
- User state at the point of receipt of notification
- Was the user using the phone at the point of receipt of notification
- User Location on notification receipt
- User Location on notification consumption
- User response

Primary Research: Selection of Parameters

- Preliminary list based on what can be measured by android
- Filter out unviable parameters
  - User Activity, location: Resource intensive

- App/Package Name
- Date of receipt
- Time of Receipt
- Time of Consumption
- Priority of message
- Text Content of the notification
- Title of notification
- User Activity on receipt of notification
- User Activity on consumption of notification
- User state at the point of receipt of notification
- Was the user using the phone at the point of receipt of notification
- User Location on notification receipt
- User Location on notification consumption
- User response

Primary Research: Selection of Parameters

- Preliminary list based on what can be measured by android
- Filter out unviable parameters

- App/Package Name
- Date of receipt
- Time of Receipt
- Time of Consumption
- Priority of message
- Text Content of the notification
- Title of notification
- User response

Primary Research: Data Collection

- Develop a Data Collection Tool

Primary Research: Data Collection

- Develop a Data Collection Tool
  - To be installed in the user’s smartphone

Primary Research: Data Collection

- Develop a Data Collection Tool
- Developed in Android Studio

Image Source:
http://www.eliteisinc.com/wp-content/uploads/2015/11/android-studio.png

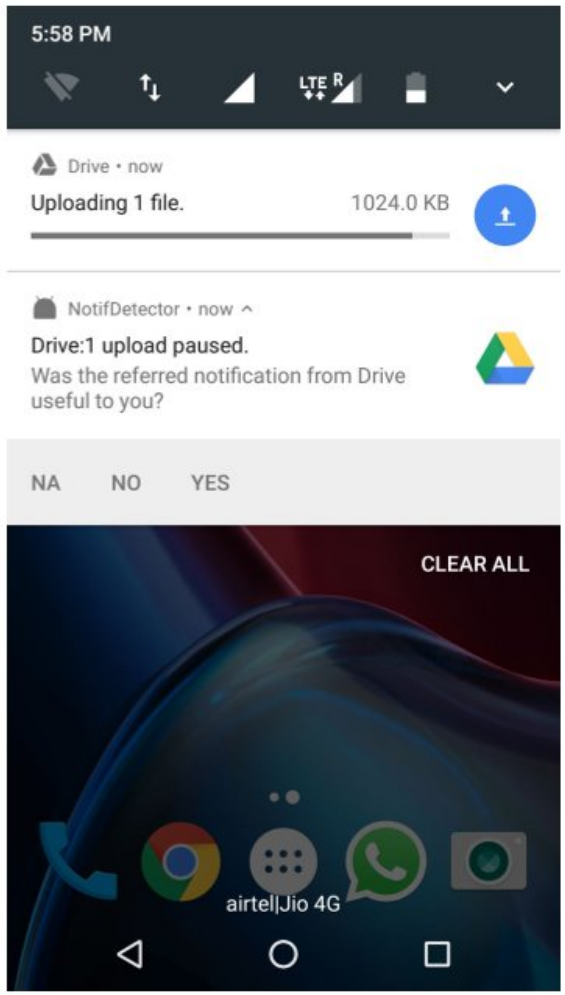
Primary Research: Data Collection

- Develop a Data Collection Tool (DCT)
- Developed in Android Studio
- DCT installed on 8 users' devices
  - 7 users - 1 week
  - 1 user - 4 weeks

Image Source:
http://www.eliteisinc.com/wp-content/uploads/2015/11/android-studio.png

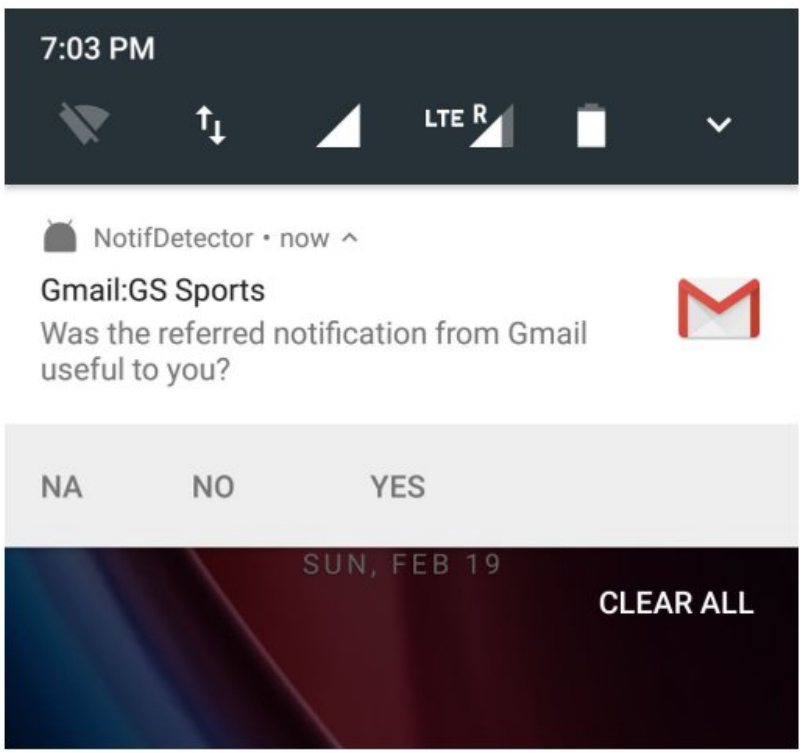
Data Collection: Tool Development

- User response collection
  - Thru a notification from Data Collection Tool



Data Collection: Tool Development

- User response collection
  - Thru a notification from Data Collection Tool
- Notification contains:
  - App name
  - Notification title
  - Response buttons



Primary Research: Recap

- Semi-Structured Interviews
- Data Collection Tool

P3 Defence • Now

Design Implications

Derived from primary and secondary research

P3 Defence • In 2 mins

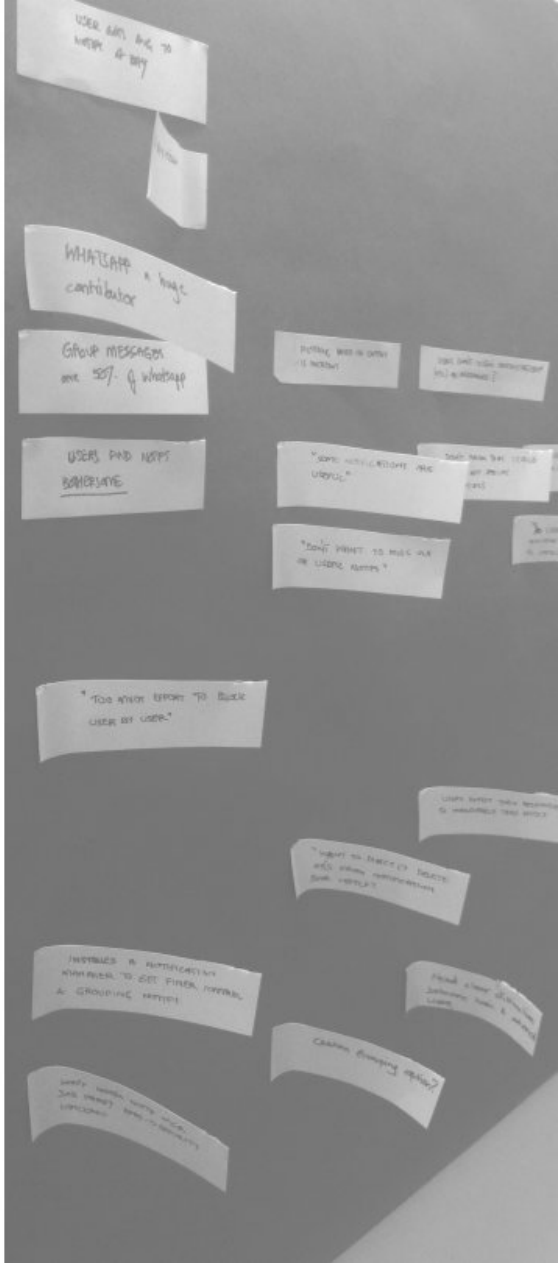
Design

P3 Defence • In 10 mins

Evaluations

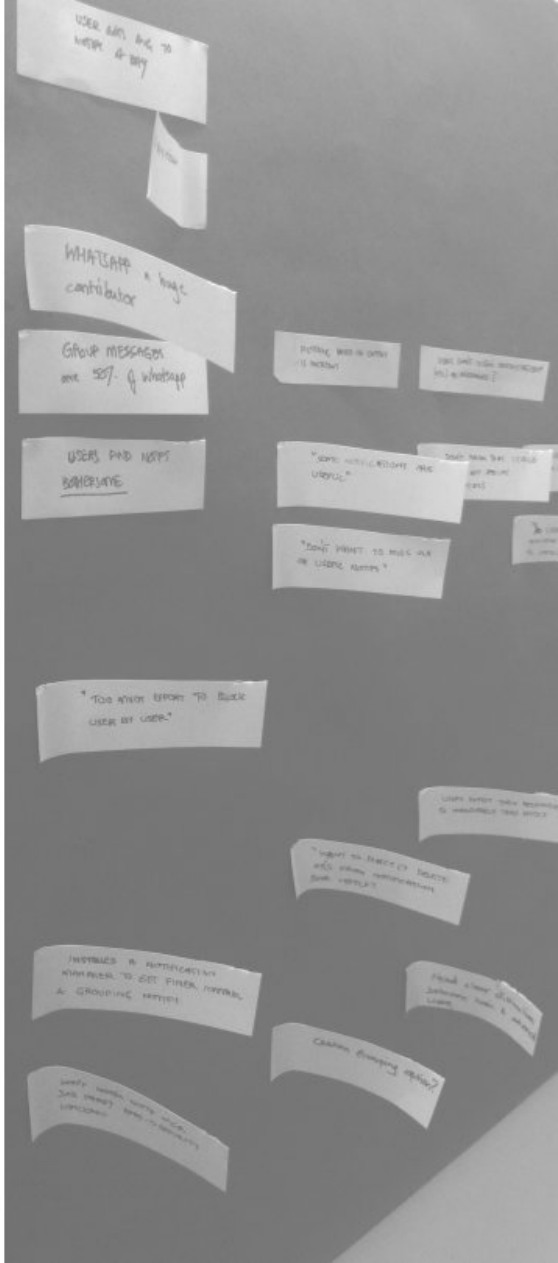
Design Implications:

- Types of Users:
  - Type 1: Zero Customizer
  - Type 2: Moderate Customizer
  - Type 3: Heavy Customizer
  - Type 4*: Cluster Blocker



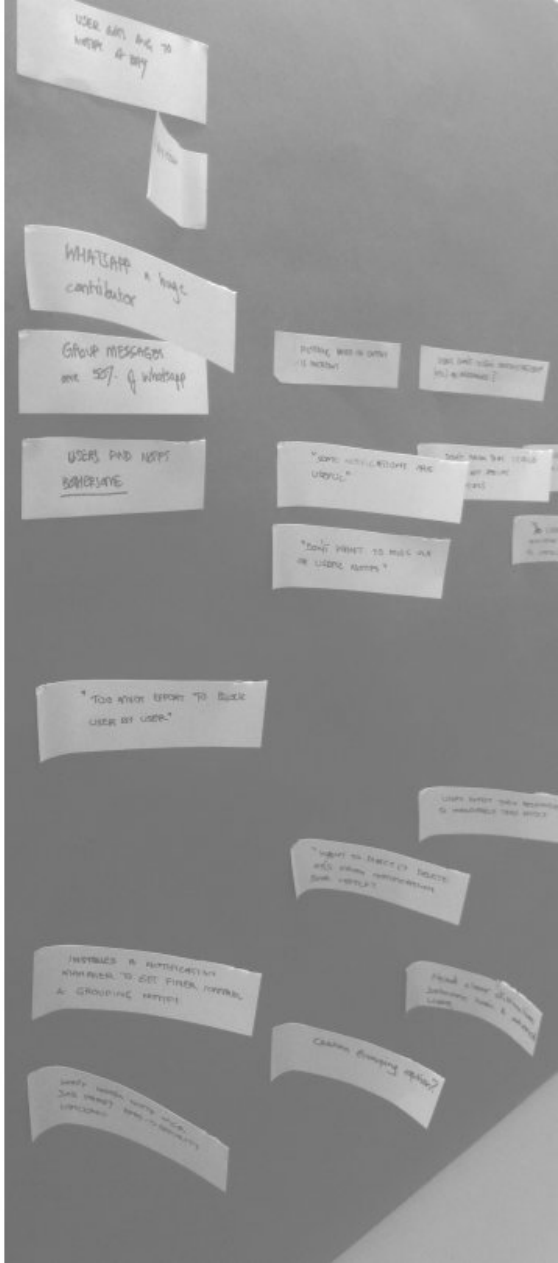
Design Implications:

- Types of Users:
  - Type 1: Zero Customizer
  - Type 2: Moderate Customizer
  - Type 3: Heavy Customizer
  - Type 4*: Cluster Blocker
- **Cater to each user type**



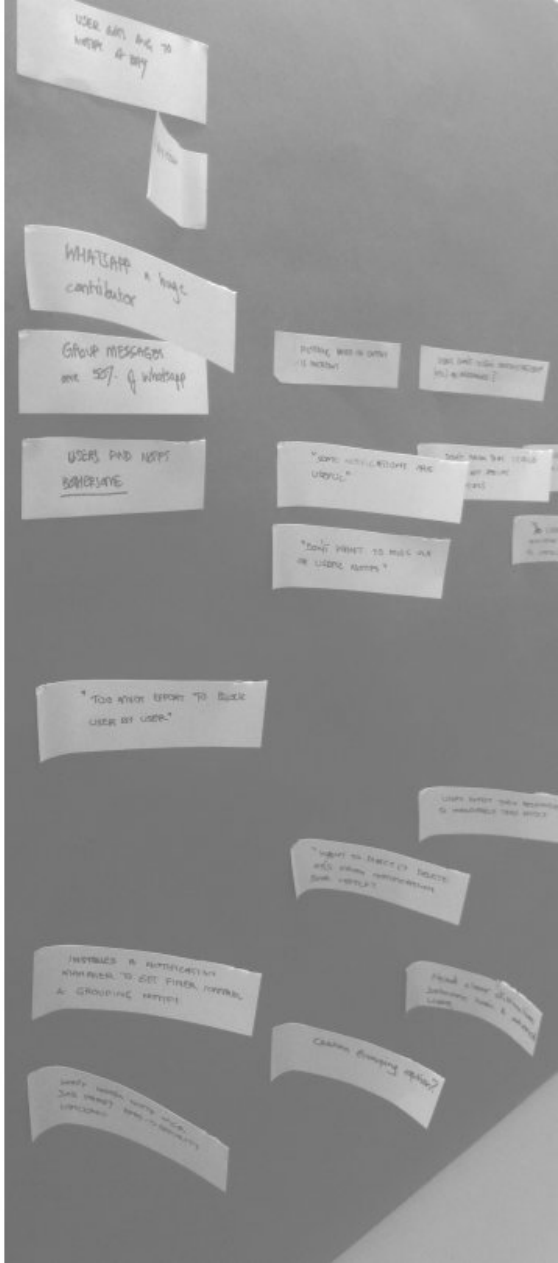
Design Implications:

- Types of Users:
  - Type 1: Zero Customizer
  - Type 2: Moderate Customizer
  - Type 3: Heavy Customizer
  - Type 4*: Cluster Blocker
- Cater to each user type
- **Filter. Don’t block.**



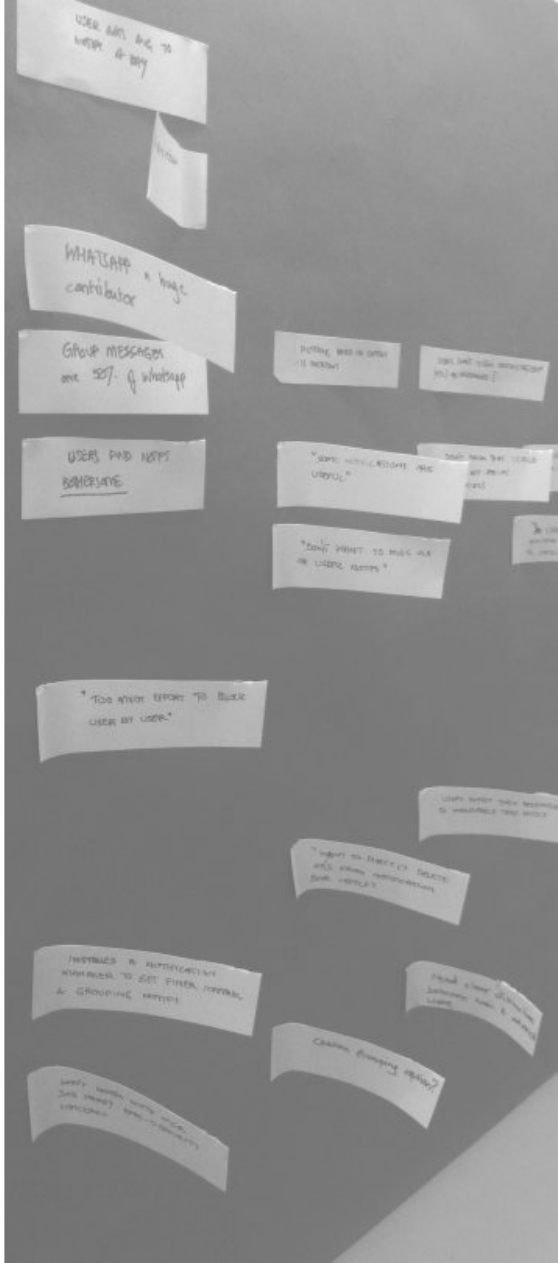
Design Implications:

- Types of Users:
  - Type 1: Zero Customizer
  - Type 2: Moderate Customizer
  - Type 3: Heavy Customizer
  - Type 4*: Cluster Blocker
- Cater to each user type
- Filter. Don’t block.
- **Prompt Feedback to user actions**
  - People stop using a feature in the absence of visible effect of their actions



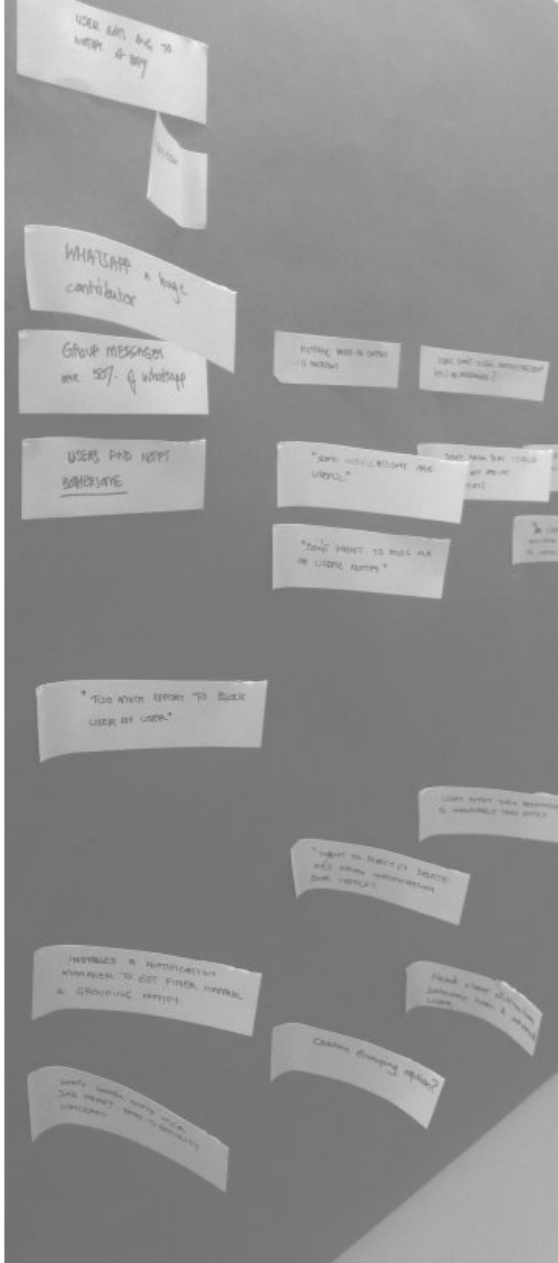
Design Implications:

- Types of Users:
  - Type 1: Zero Customizer
  - Type 2: Moderate Customizer
  - Type 3: Heavy Customizer
  - Type 4*: Cluster Blocker
- Cater to each user type
- Filter. Don’t block.
- Prompt Feedback to user actions
- **Tackle F.o.M.O**



Design Implications:

- Types of Users:
  - Type 1: Zero Customizer
  - Type 2: Moderate Customizer
  - Type 3: Heavy Customizer
  - Type 4*: Cluster Blocker
- Cater to each user type
- Filter. Don’t block.
- Prompt Feedback to user actions
- Tackle F.o.M.O
- **Delay delivery to opportune moments**



Monday, May 1 2017

P3 Defence • Now

Design

Design of system and interactions

P3 Review • In 7 mins

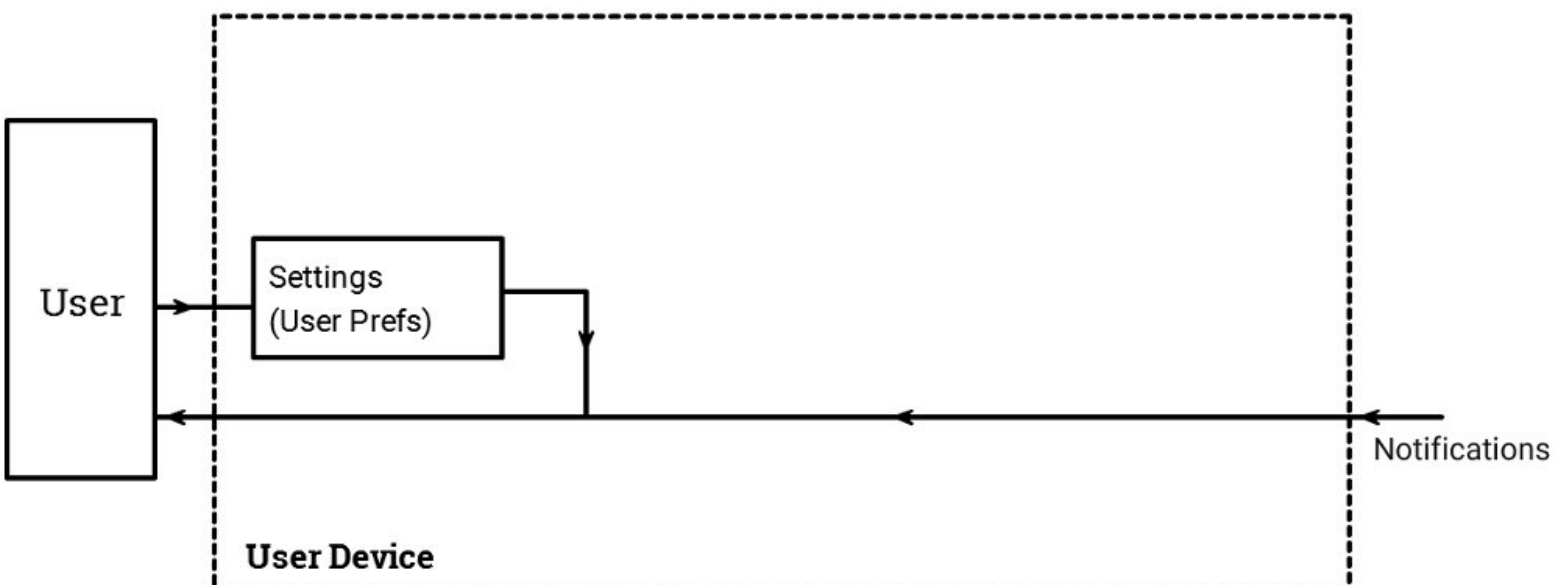
Evaluation

P3 Review • In 12 mins

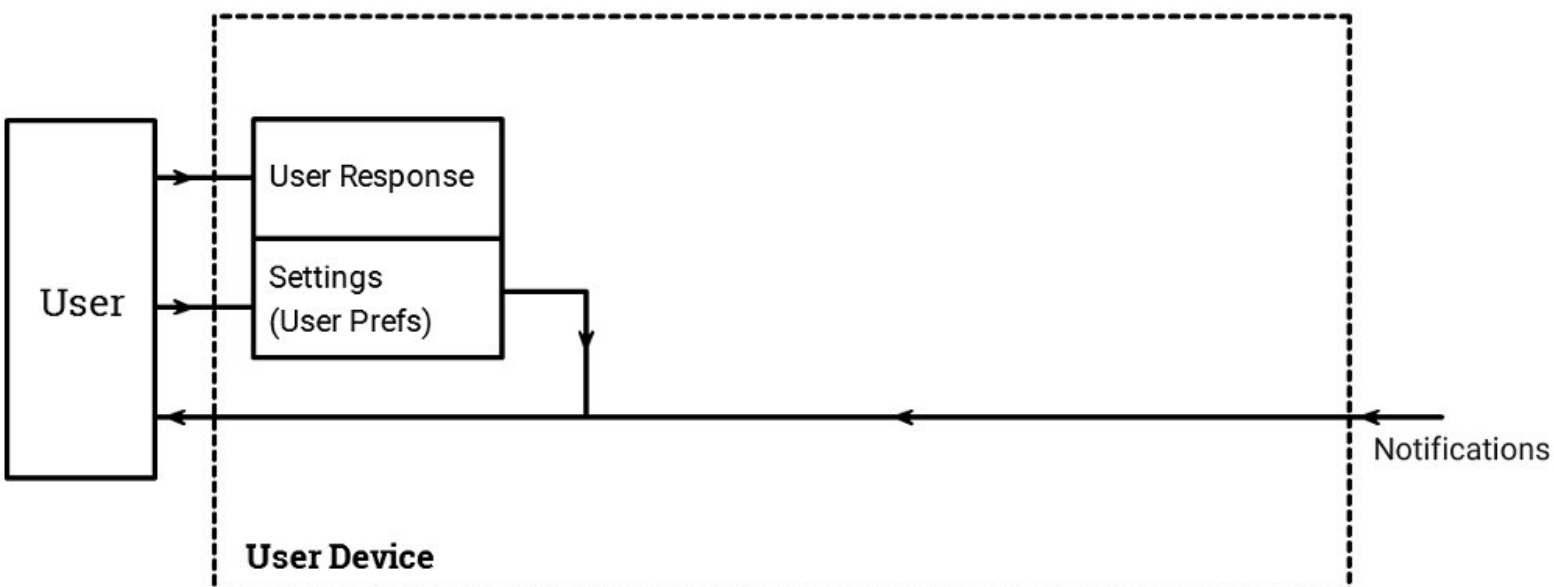
Conclusion

CLEAR ALL

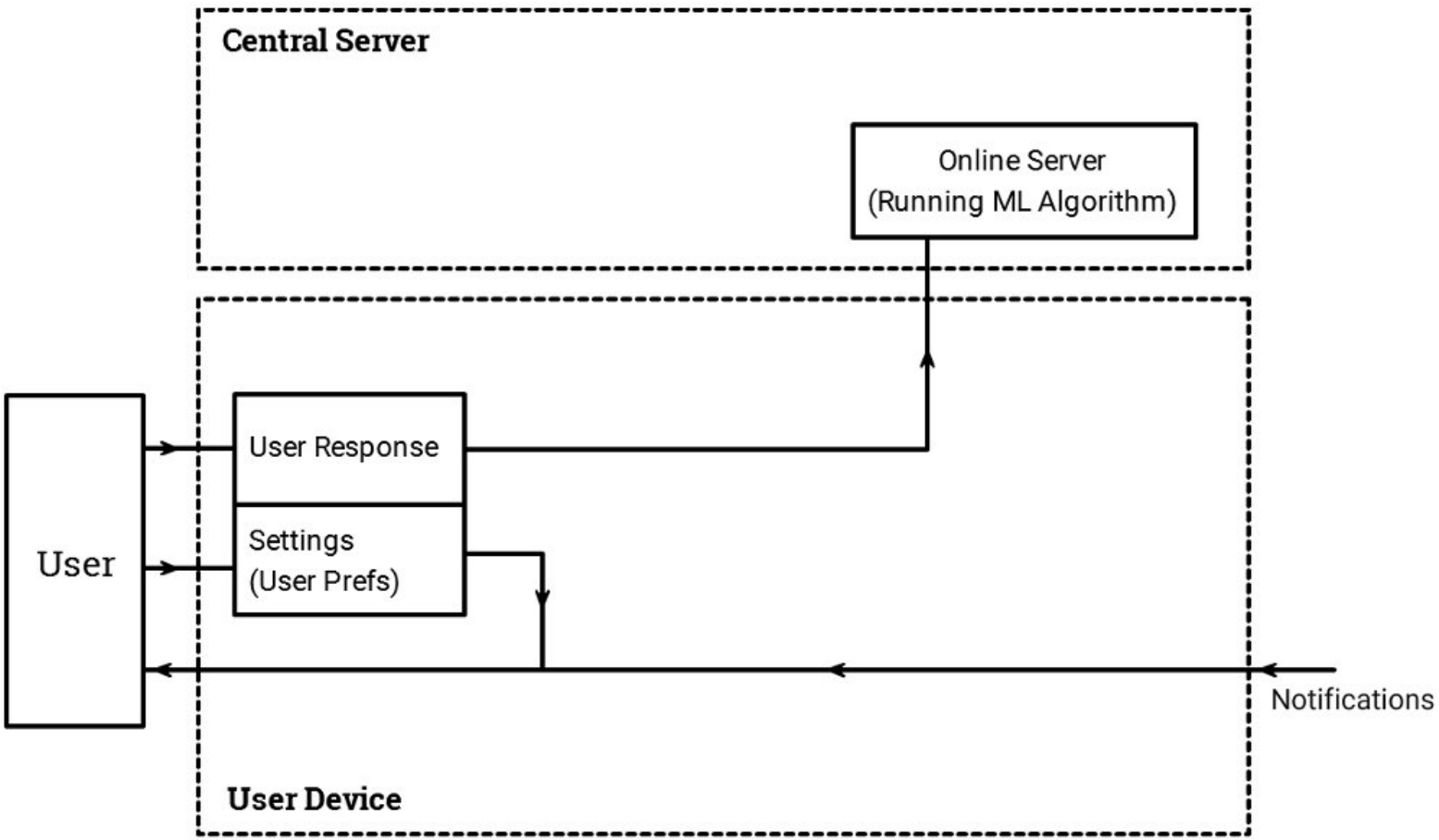
Design: Current System



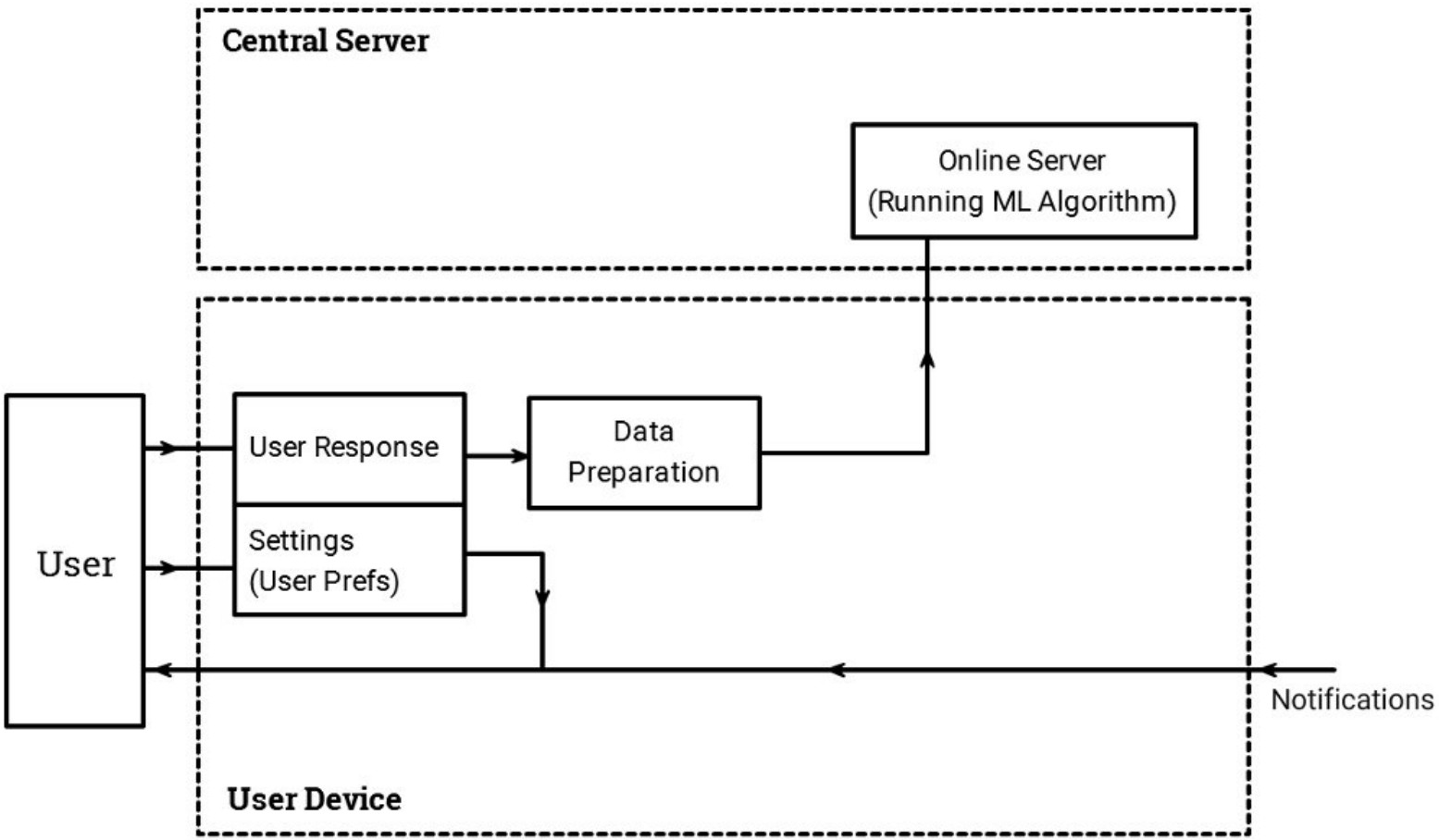
Design: Proposed System



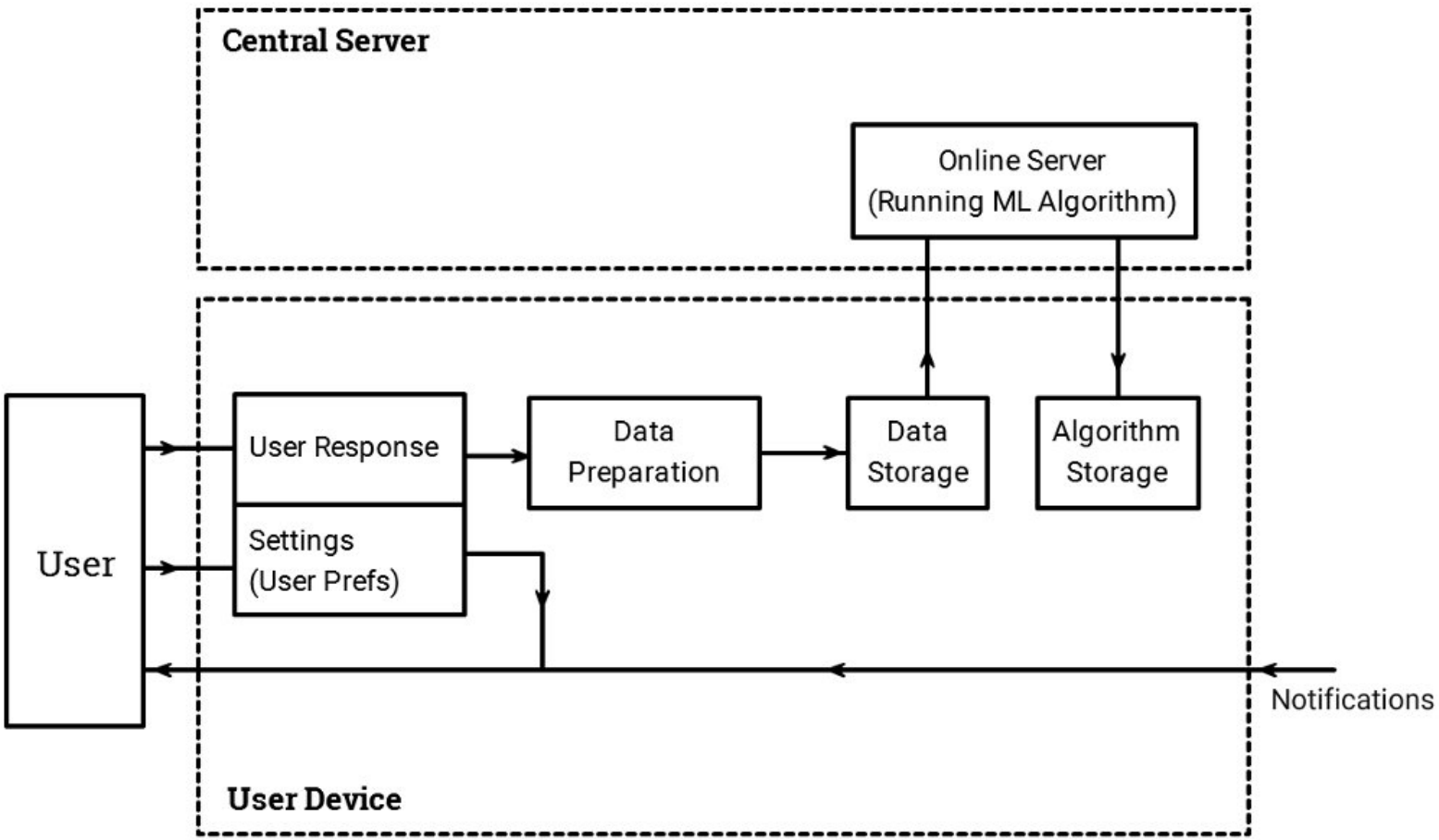
Design: Proposed System



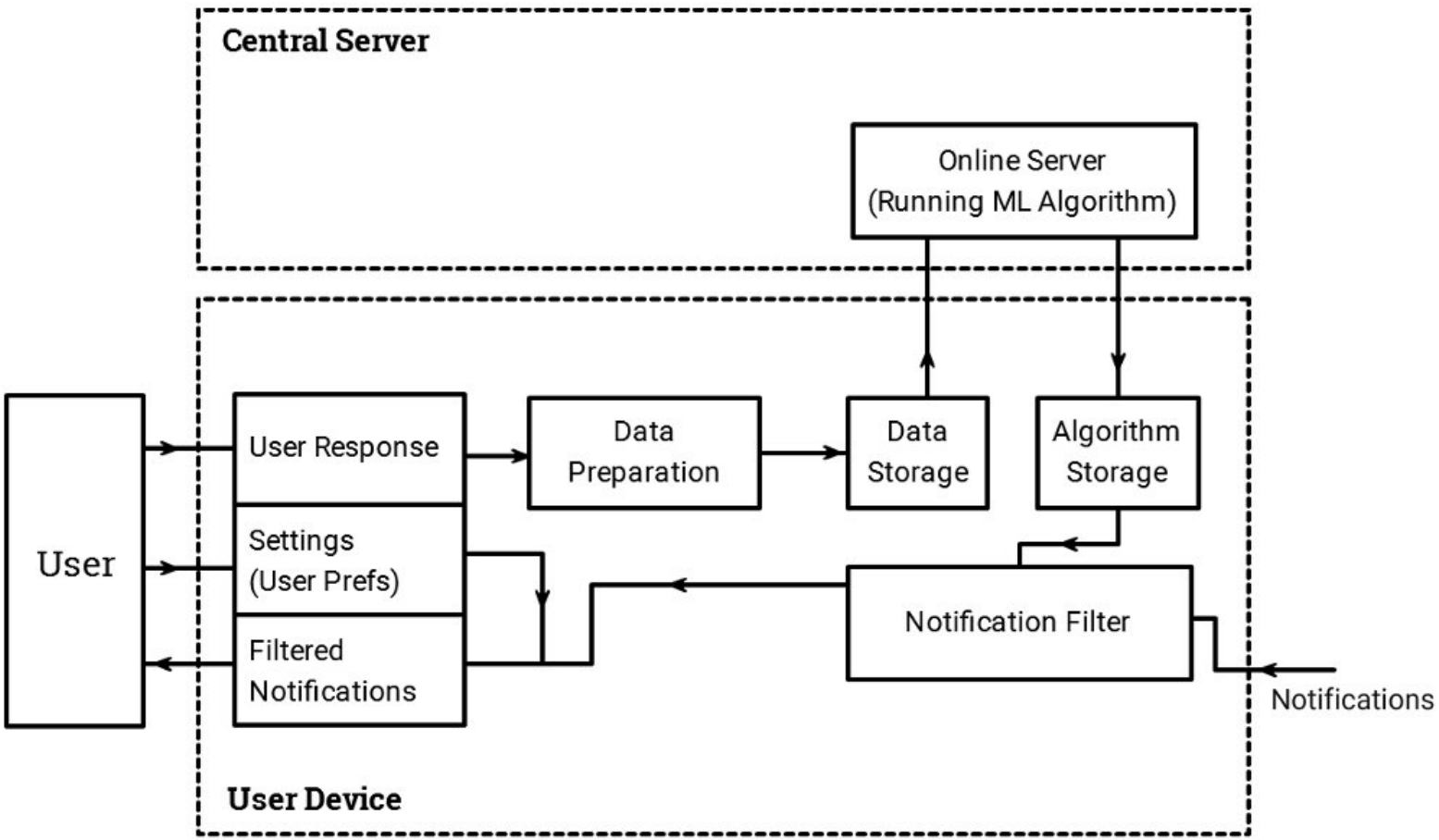
Design: Proposed System



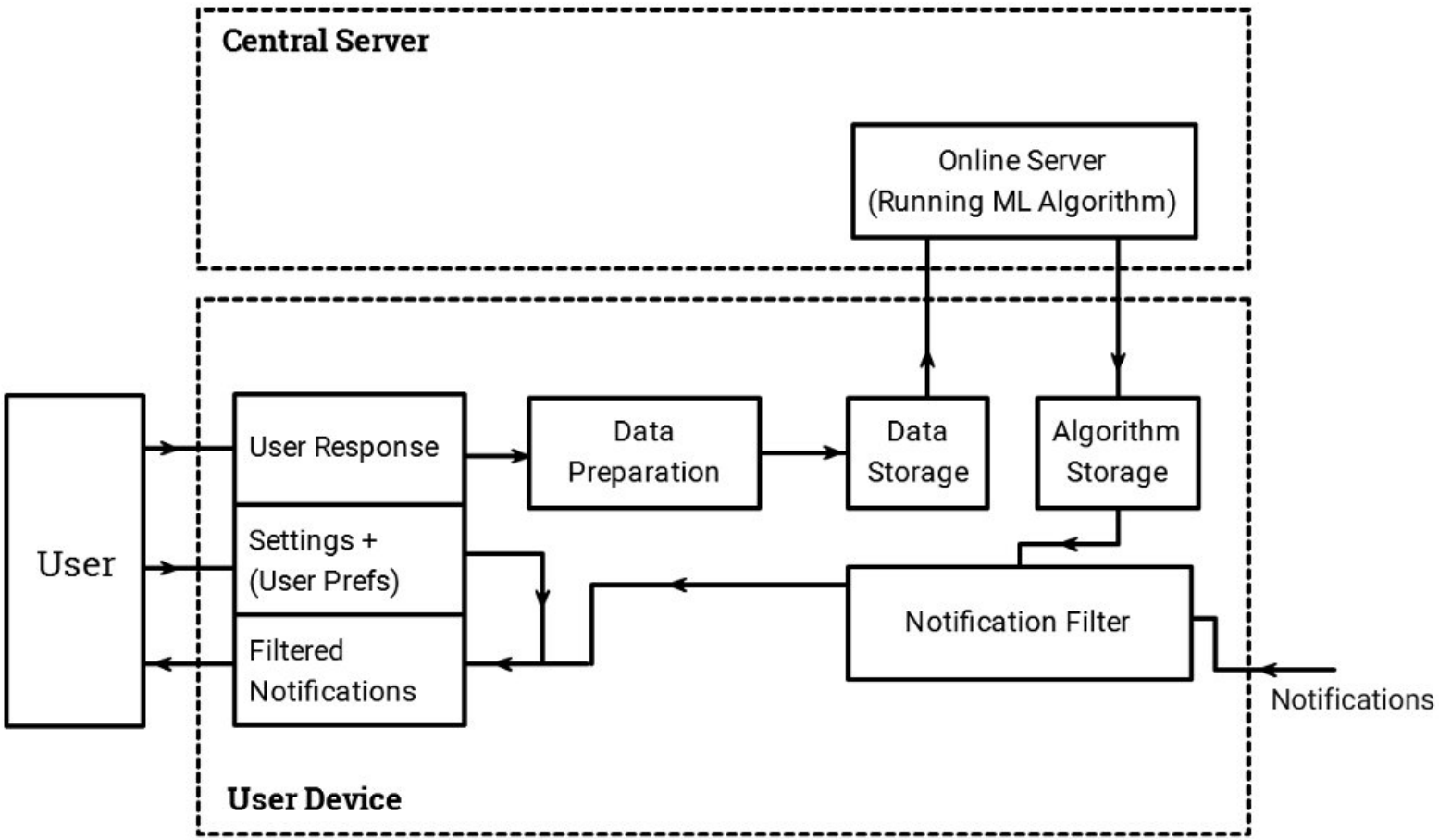
Design: Proposed System



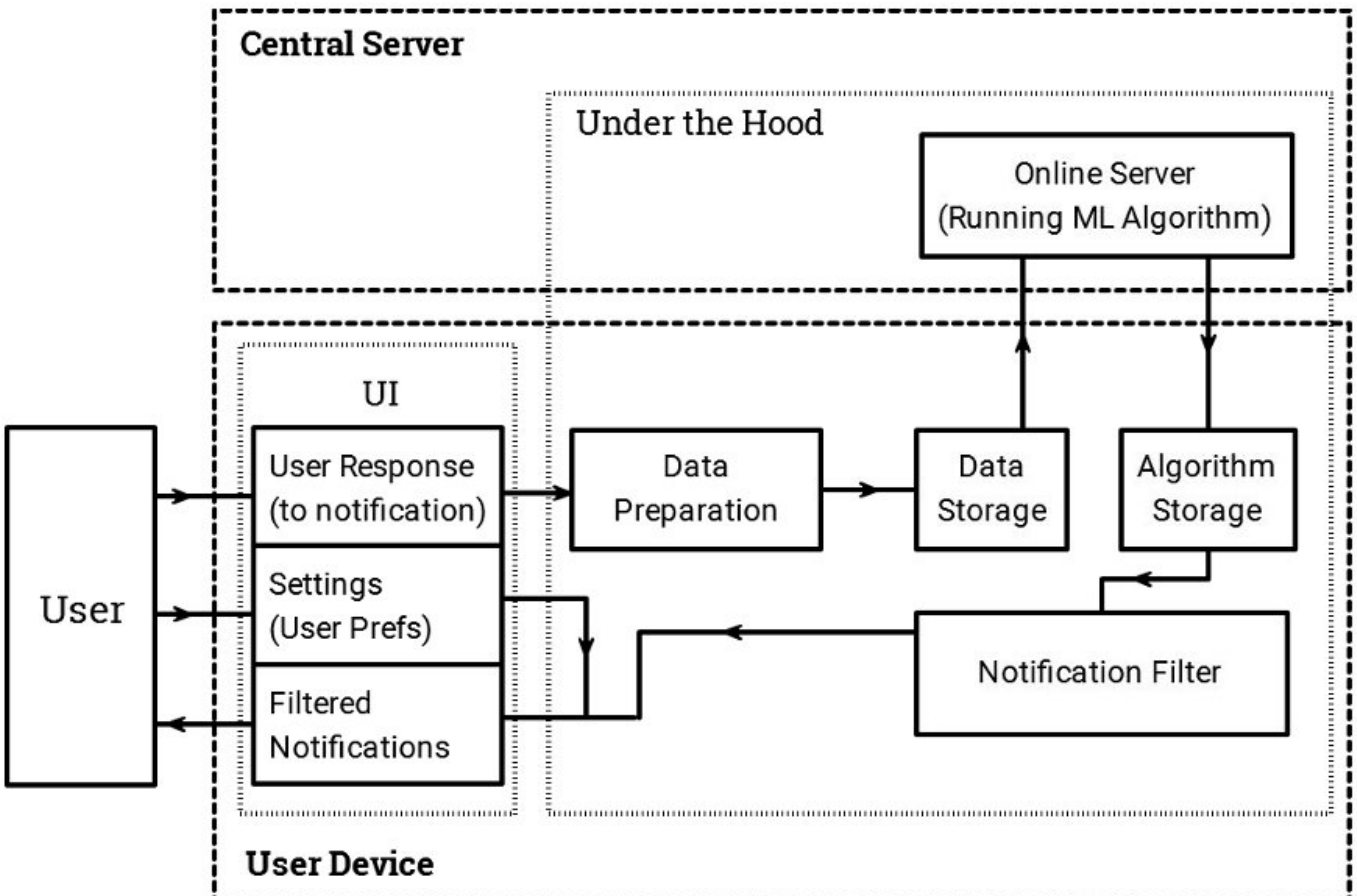
Design: Proposed System



Design: Proposed System



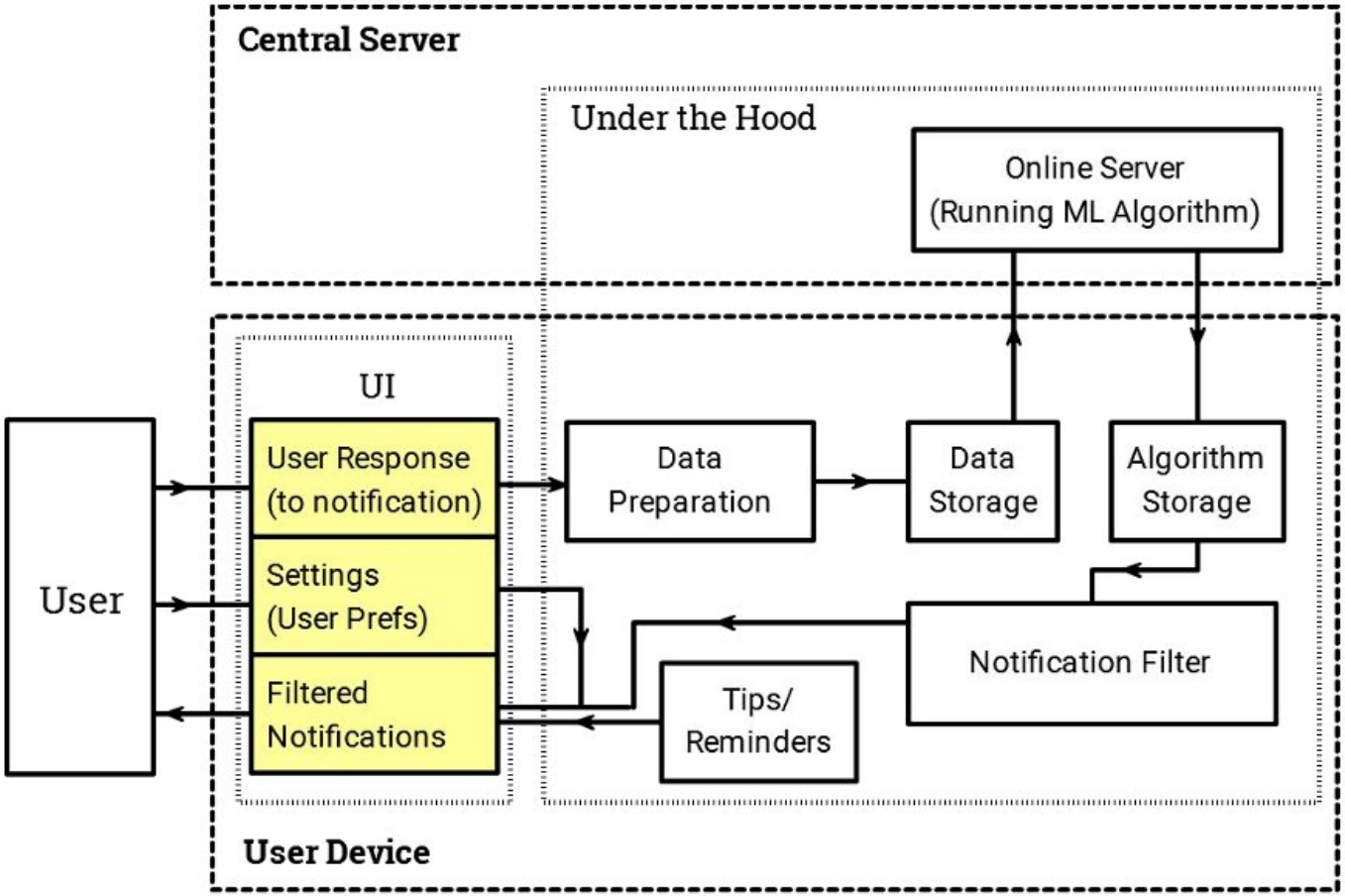
Design: Overall System



Design: Prototyping Approach

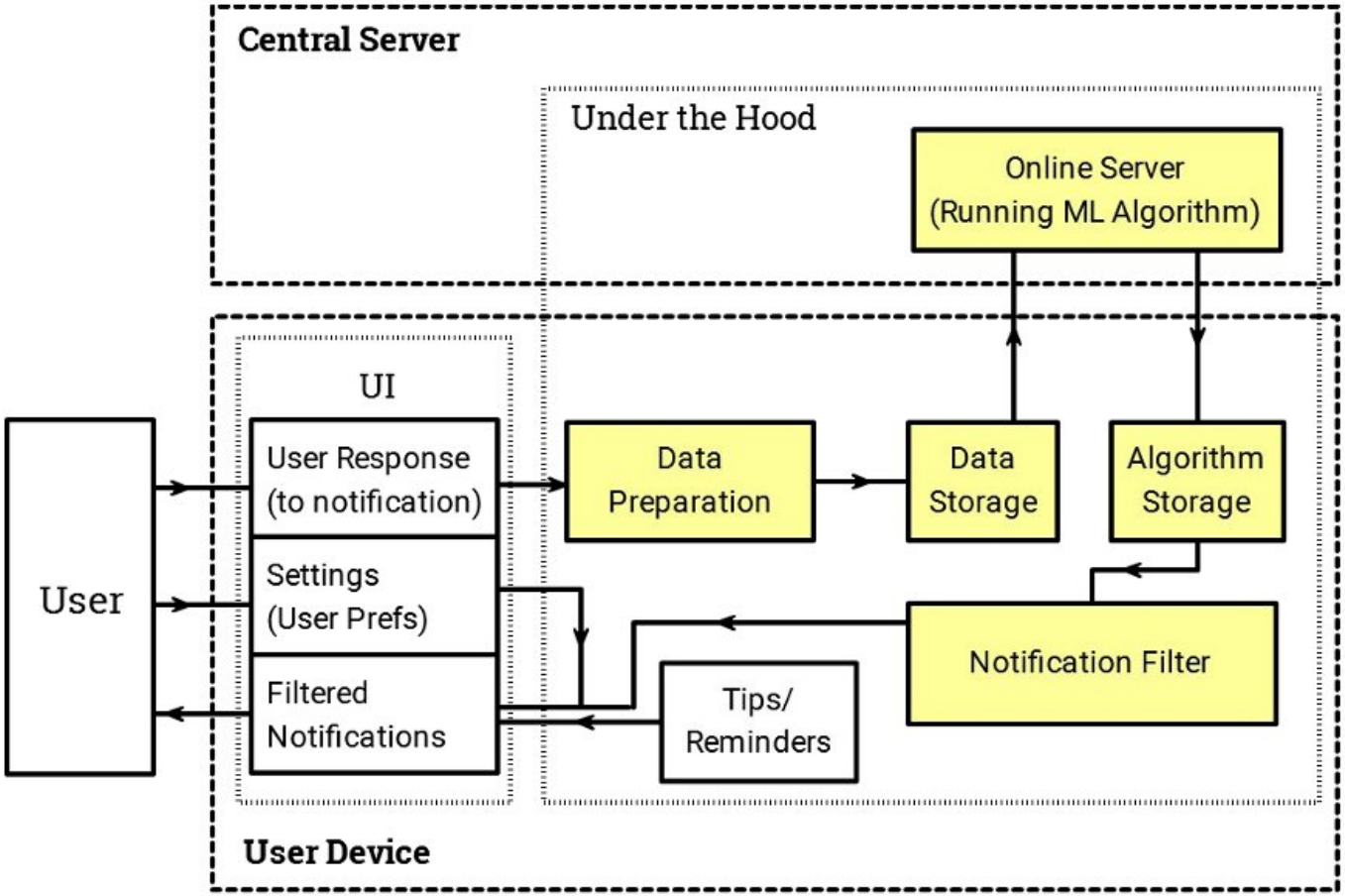
- UI Prototype
  - Demo
  - Due to restrictions on modifying interaction at OS level
- Functioning System Prototype
  - For evaluating performance of system

Design: Prototyping Approach

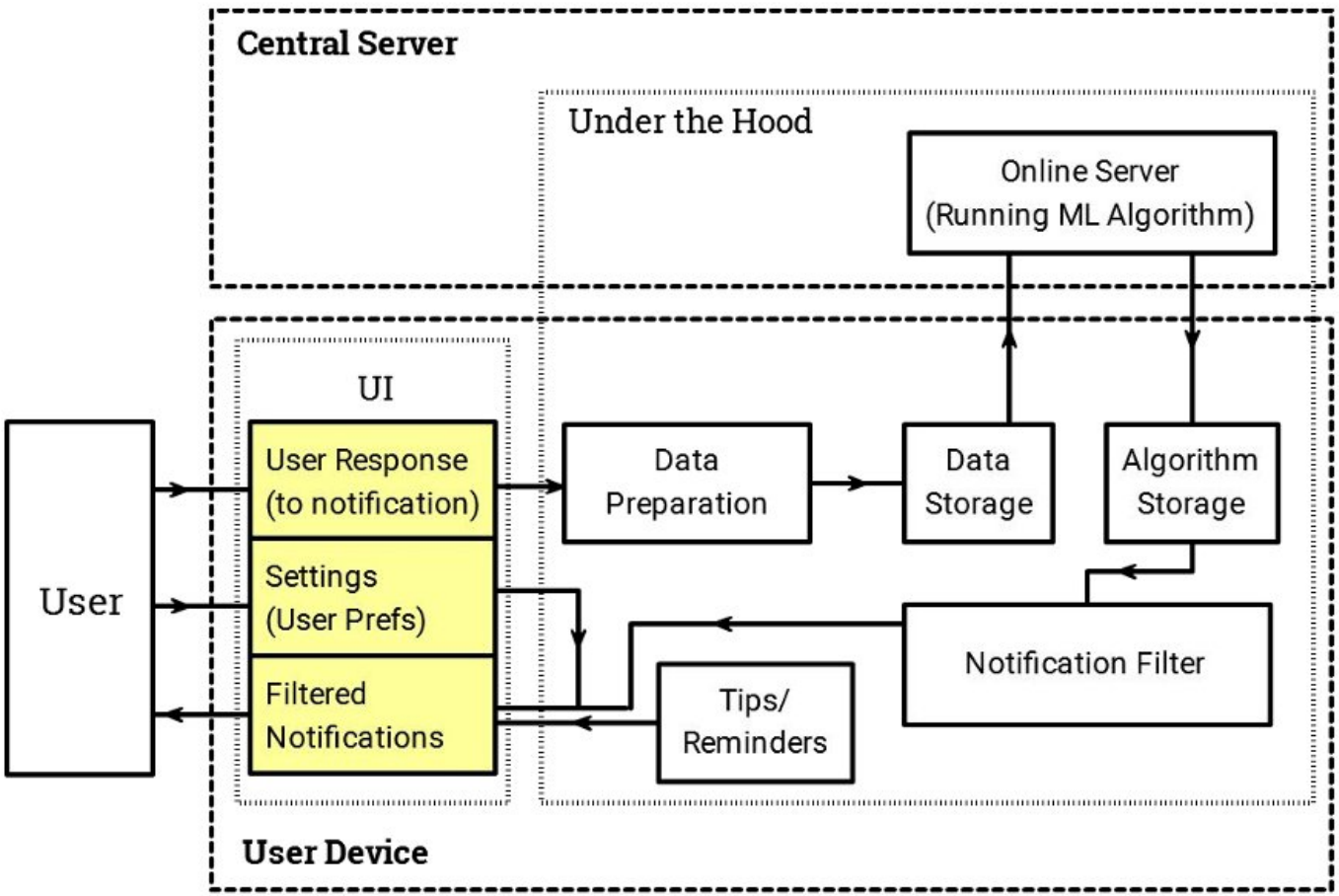


Modules Included in UI Prototype -

Design: Prototyping Approach



Design: UI


Central Server
Under the Hood
Online Server
(Running ML Algorithm)
UI
User
User Response
(to notification)
Settings
(User Prefs)
Filtered
Notifications
Data
Preparation
Data
Storage
Algorithm
Storage
Notification Filter
Tips/
Reminders
User Device

Design: UI

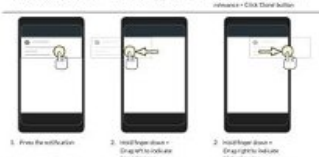
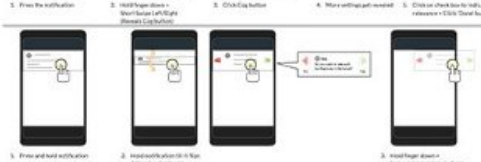
- User Response (Interaction)
- Filtered Notification Presentation
  - Interactions
  - Visuals (Layout and Icon)
- Fine Controls (Flow + Interactions)

Design - UI: User Response

- Function: Provide user with quick and easy method to communicate their opinion on the relevance of a notification
- Guidelines:
  - Minimum no. of actions
  - Minimum Cognitive Load

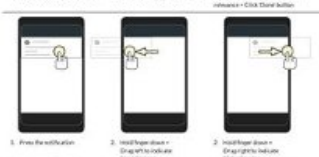
Design - UI: User Response : Ideation

- Function: Provide user with quick and easy method to communicate their opinion on the relevance of a notification
- Guidelines:
  - Minimum no. of actions
  - Minimum Cognitive Load

INTERACTION	EFFORT	COGNITIVE LOAD *(indicative)*	
	Moderate (3.5)	High	
	Low (1.5)	High	
	High (4.5)	Low	
	High (4.5)	High	
	Low (2)	Low	

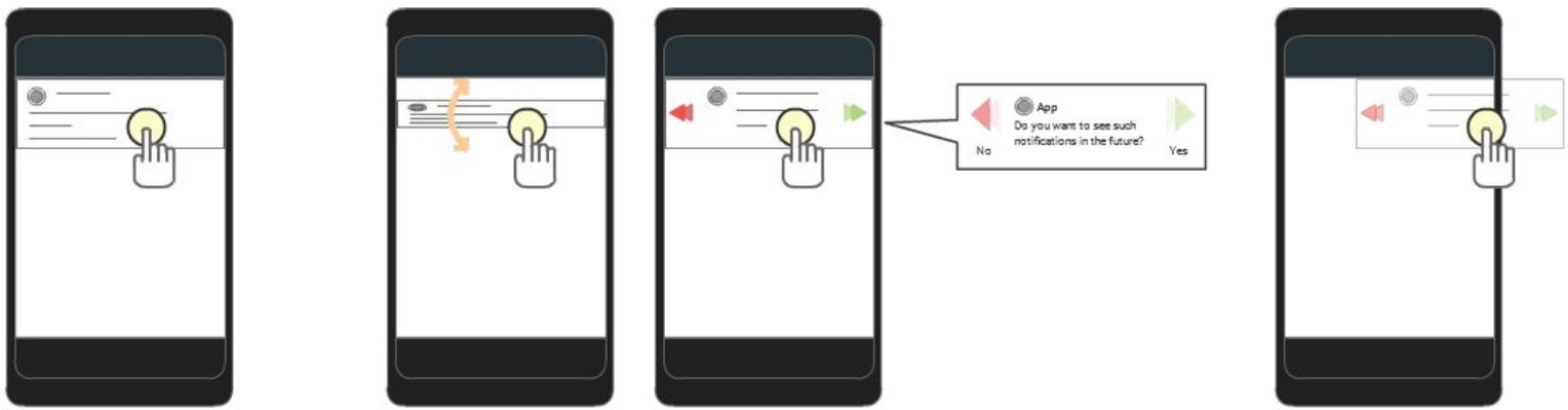
Design - UI: User Response : Ideation

- Function: Provide user with quick and easy method to communicate their opinion on the relevance of a notification
- Guidelines:
  - Minimum no. of actions
  - Minimum Cognitive Load

INTERACTION	EFFORT	COGNITIVE LOAD *(indicative)*
	Moderate (3.5)	High
	Low (1.5)	High
	High (4.5)	Low
	High (4.5)	High
	Low (2)	Low

Design - UI: User Response

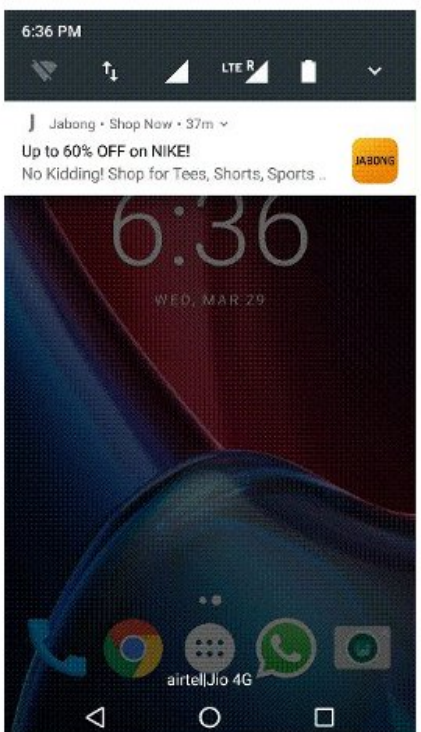
- Selected Interaction: Based on **effort** and **cognitive load** on user



1. Press and hold notification
2. Hold notification till it flips (Vibration feedback)
3. Hold finger down + Swipe right or left to indicate high/low relevance

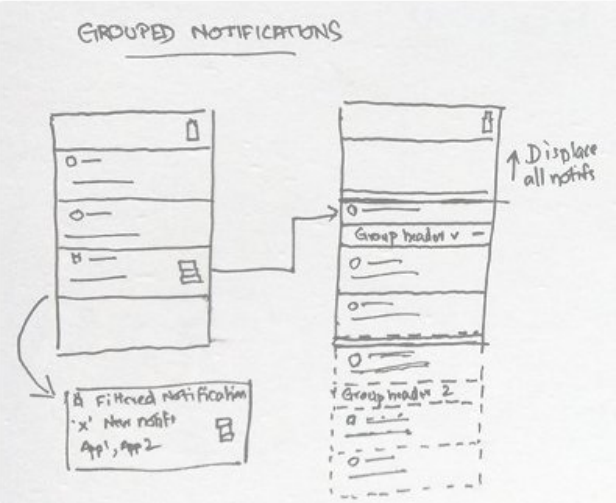
Design - UI: User Response

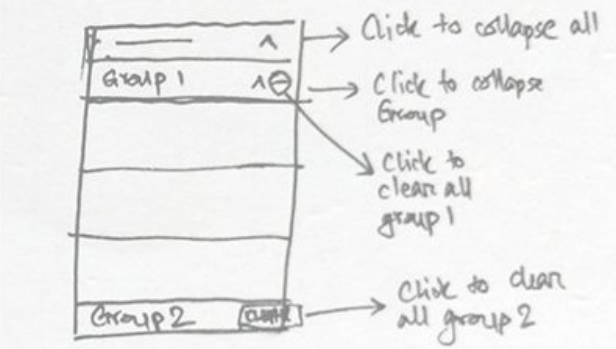
- Selected Interaction



Design - UI: Presentation

- Potentially >40 notifications per day
  - Grouping for easier consumption
- On notification bar over separate screen
  - To maintain user conceptual models
- Flow designed and screens wireframed

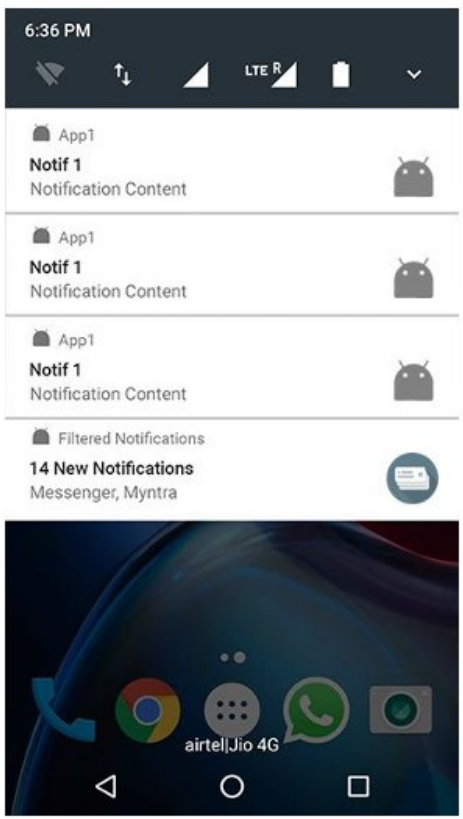


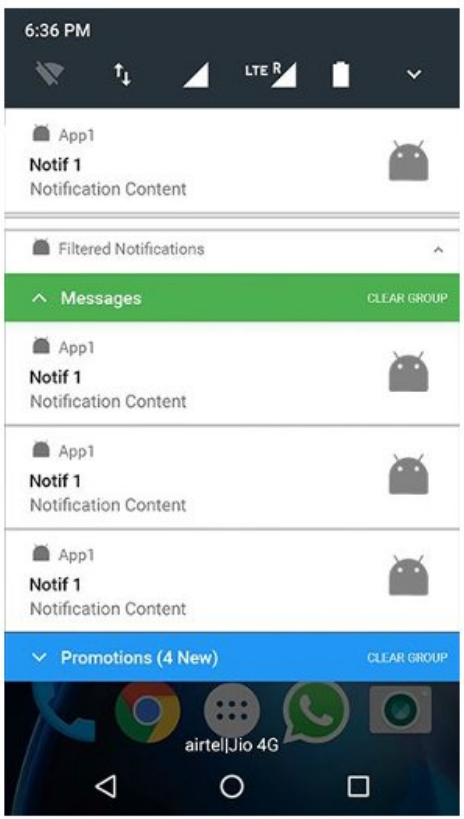


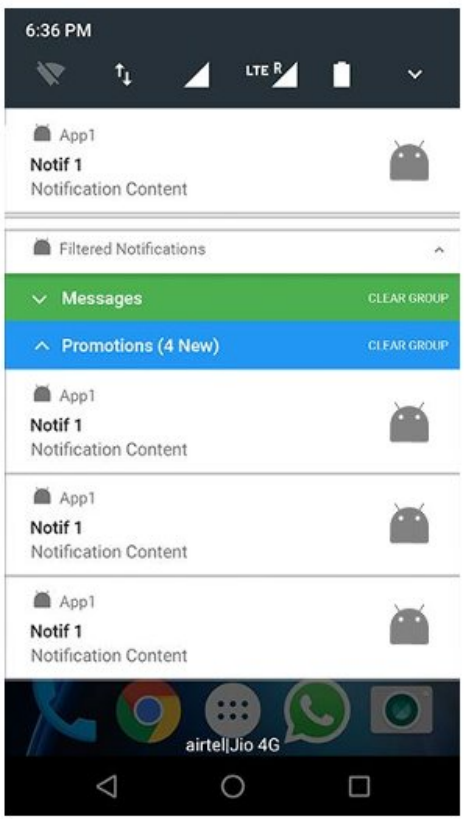
Design - UI: Presentation

- Icon Designed for
  “filtered+grouped” notification

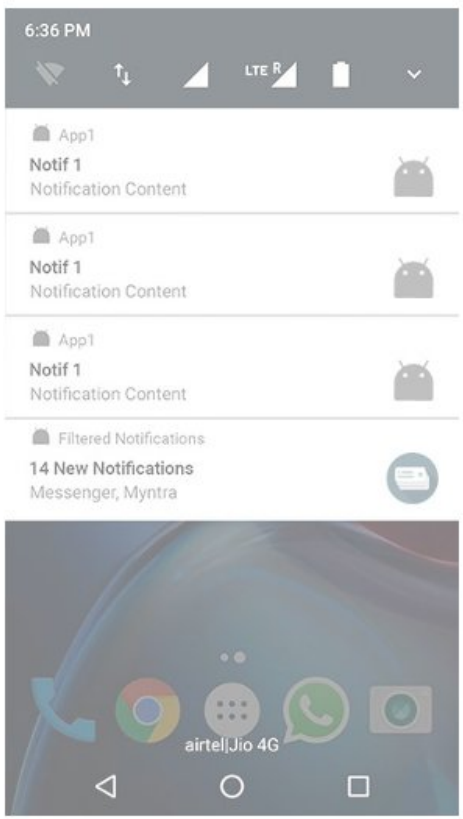
Design - UI: Presentation

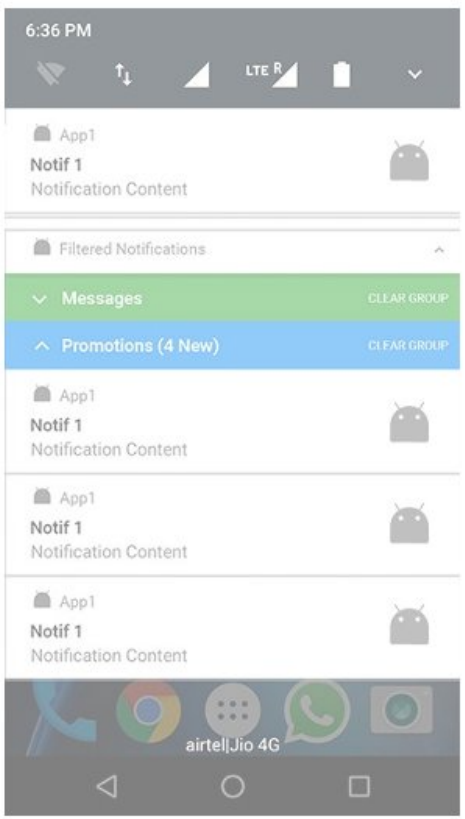






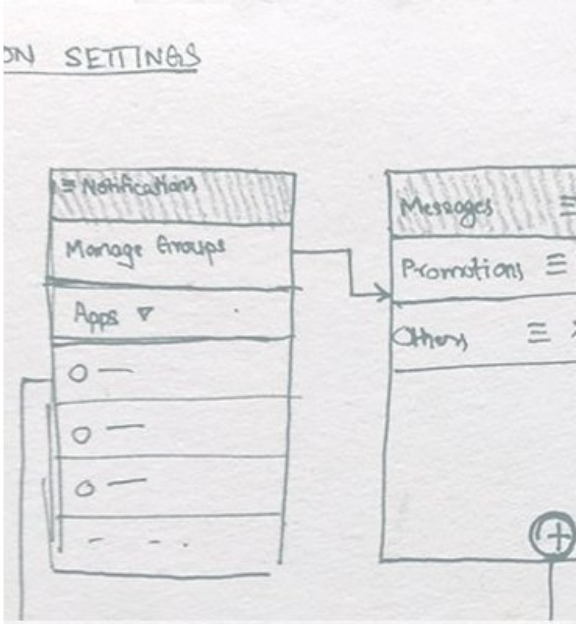
Design - UI: Presentation

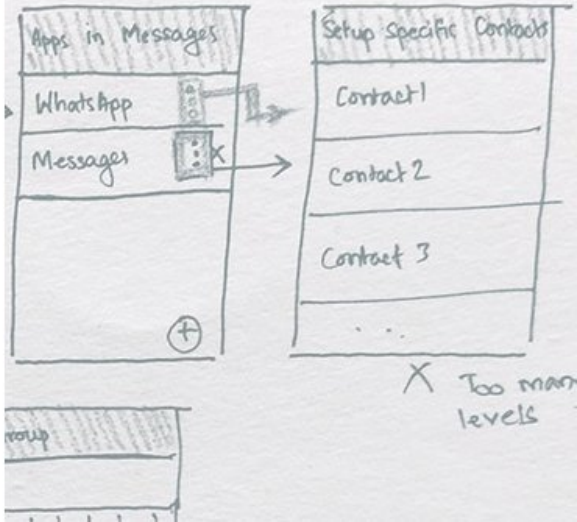




Design - UI: Fine Controls

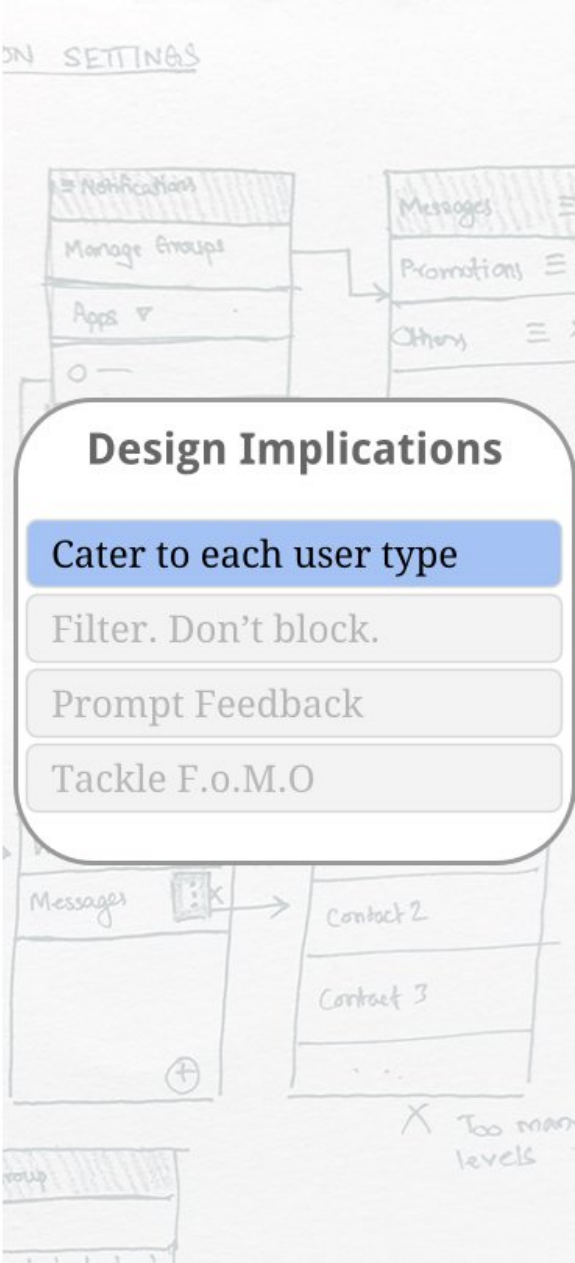
- Guidelines:
  - New options shouldn’t clash with existing conceptual model of users
  - Provide user with means to tweak filtering to the last detail





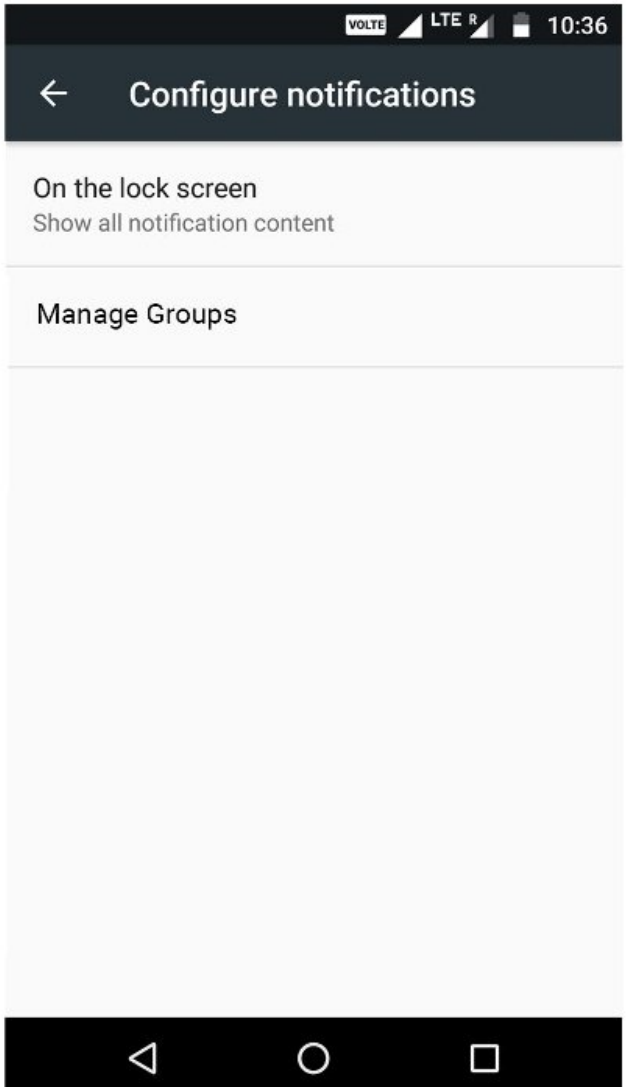
Design - UI: Fine Controls

- Guidelines:
  - New options shouldn’t clash with existing conceptual model of users
  - Provide user with means to tweak filtering to the last detail



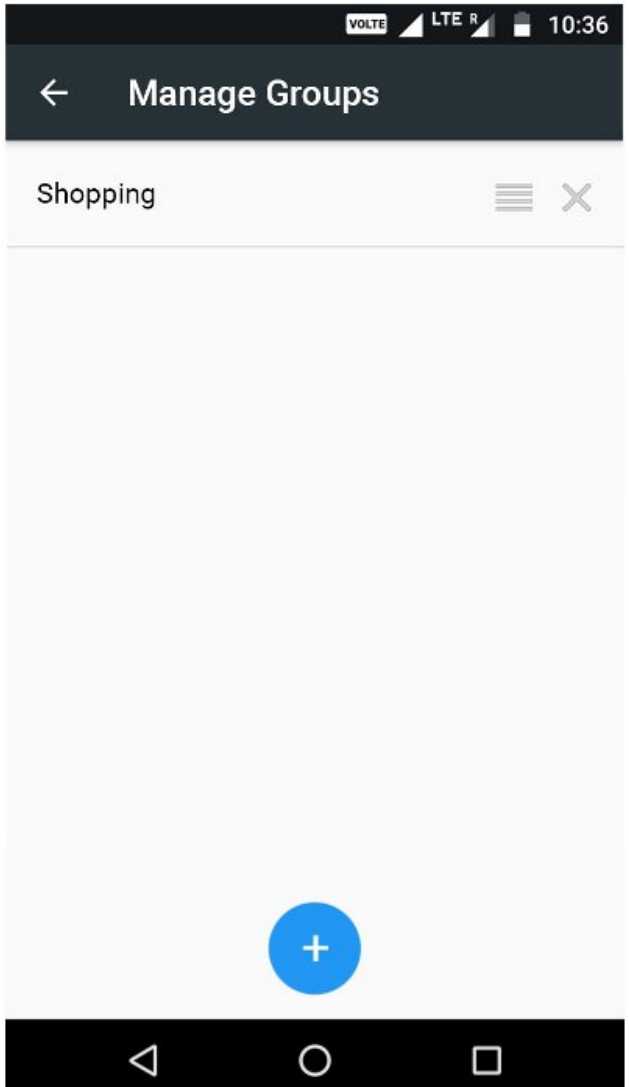
Design - UI: Fine Controls

- Customization options:
  - **Managing Groups**
    - Targeting Type 2 & 3 users



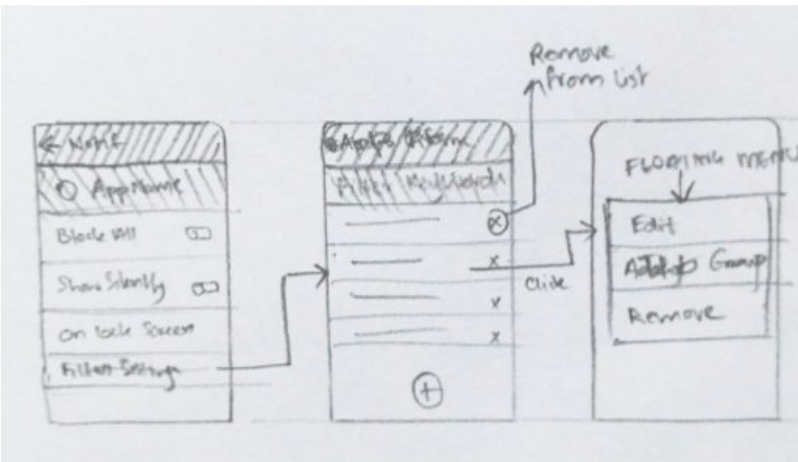
Design - UI: Fine Controls

- Customization options:
  - **Managing Groups**
    - Targeting Type 2 & 3 users



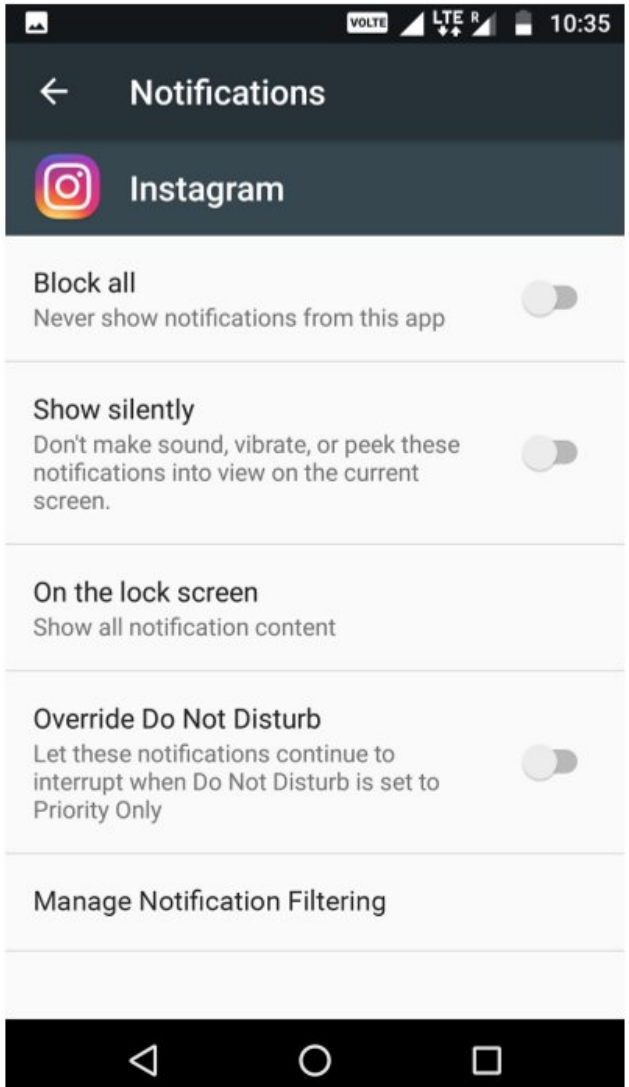
Design - UI: Fine Controls

- Customization options:
  - Managing Groups
  - **Controlling the filter keywords**
    - For Expert users (Type 3 users)



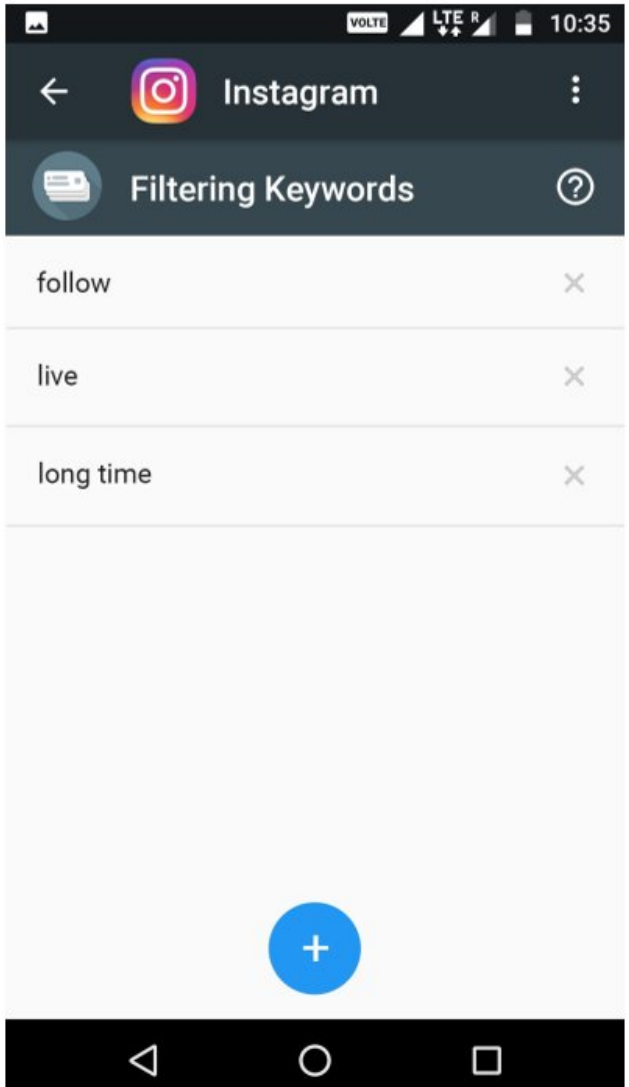
Design - UI: Fine Controls

- Customization options:
  - Managing Groups
  - **Controlling the filter keywords**
    - For Expert users (Type 3 users)



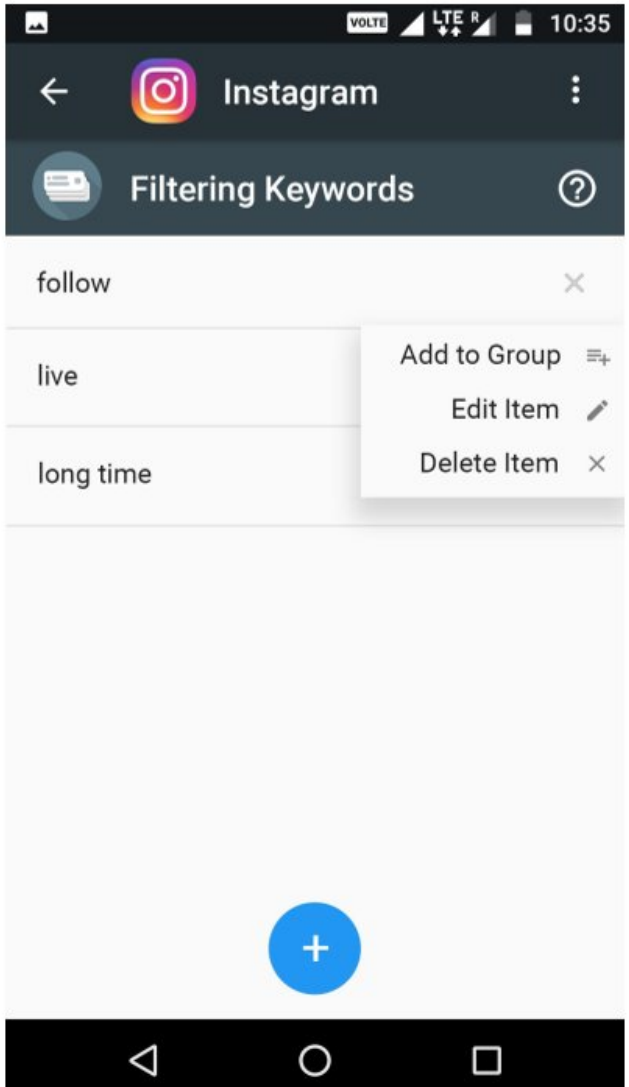
Design - UI: Fine Controls

- Customization options:
  - Managing Groups
  - **Controlling the filter keywords**
    - For Expert users (Type 3 users)

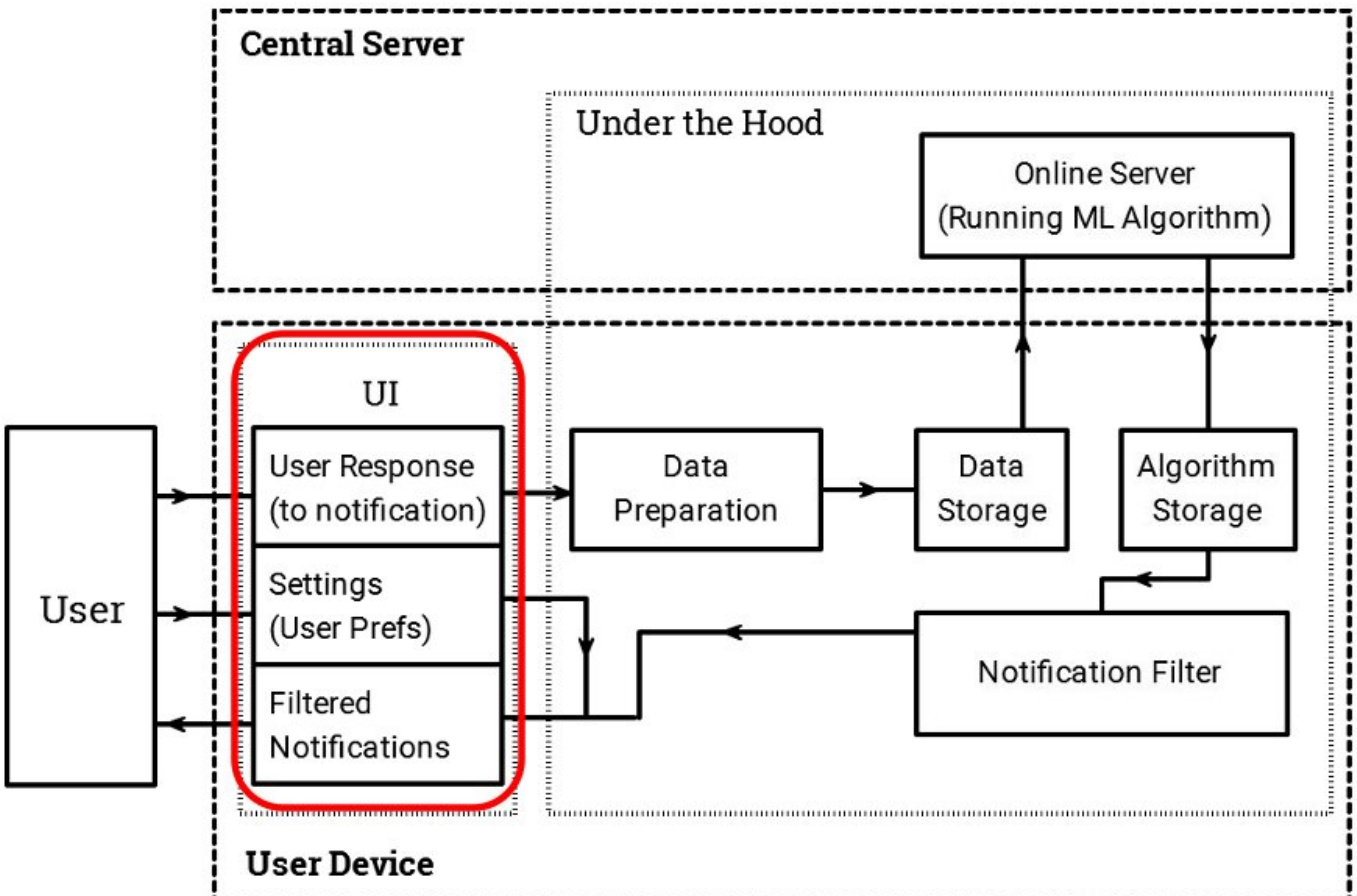


Design - UI: Fine Controls

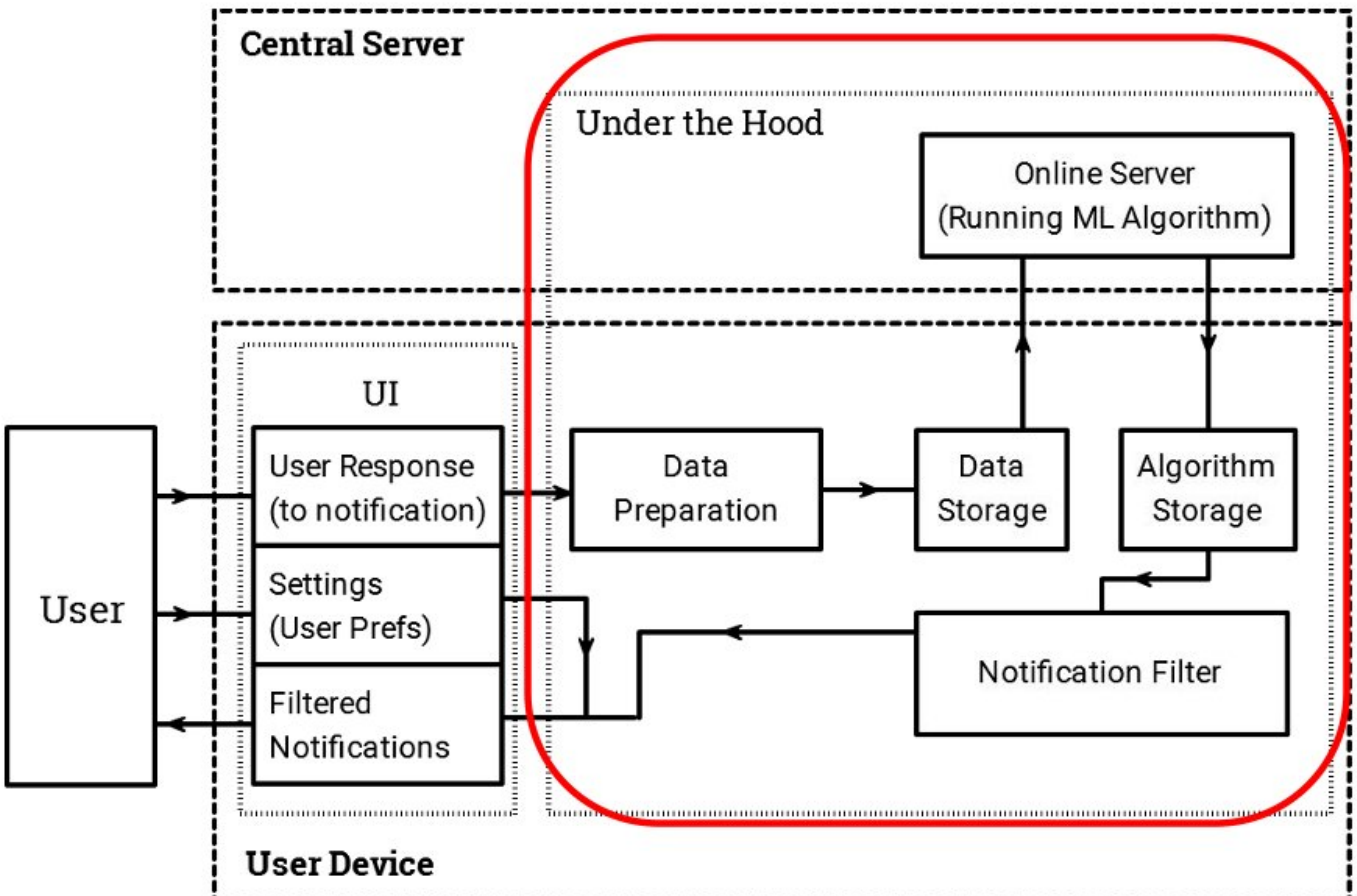
- Customization options:
  - Managing Groups
  - **Controlling the filter keywords**
    - For Expert users (Type 3 users)



Design: UI


Central Server
Under the Hood
Online Server
(Running ML Algorithm)
UI
User
User Response
(to notification)
Settings
(User Prefs)
Filtered
Notifications
Data
Preparation
Data
Storage
Algorithm
Storage
Notification Filter
User Device

Design: System



Design - System: Data Preparation

- Selection of variables to track
  - Should capture essence of user preferences

SELECTED VARIABLES

- App/Package Name
- Date of receipt
- Time of Receipt
- Time of Consumption
- Priority of message
- Text Content of the notification
- Title of notification
- User response

Design - System: Data Storage

- Guidelines
  - Quell Data Security concerns
  - Data (internet) Usage Minimization

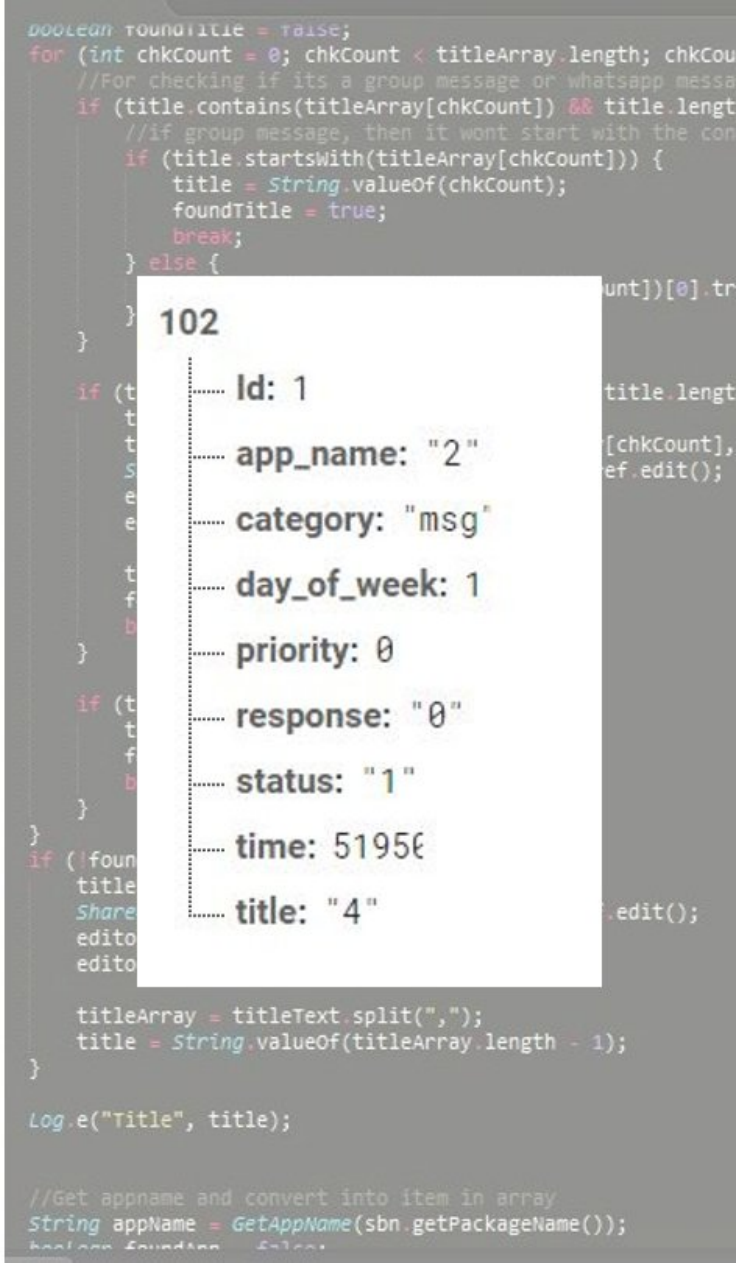
```
boolean foundTitle = false;
for (int chkCount = 0; chkCount < titleArray.length; chkCou
    //For checking if its a group message or whatsapp messa
    if (title.contains(titleArray[chkCount]) && title.lengt
        //if group message, then it wont start with the con
        if (title.startsWith(titleArray[chkCount])) {
            title = String.valueOf(chkCount);
            foundTitle = true;
            break;
        } else {
            title = title.split(titleArray[chkCount])[0].tr
        }
    }

    if (titleArray[chkCount].contains(title) && title.lengt
        titleArray[chkCount] = title;
        titleText = titleText.replace(titleArray[chkCount],
        SharedPreferences.Editor edit = sharedPref.edit();
        edit.putString("titleList", titleText);
        edit.apply();

        title = String.valueOf(chkCount);
        foundTitle = true;
        break;
    }

    if (title.equals(titleArray[chkCount])) {
        title = String.valueOf(chkCount);
        foundTitle = true;
        break;
    }
}
if (!foundTitle) {
    titleText = titleText + "," + title;
    SharedPreferences.Editor editor = sharedPref.edit();
    editor.putString("titleList", titleText);
    editor.apply();

    titleArray = titleText.split(",");
    title = String.valueOf(titleArray.length - 1);
}

Log.e("Title", title);


//Get appname and convert into item in array
String appName = GetAppName(sbn.getPackageName());
```

Design - System: Data Storage

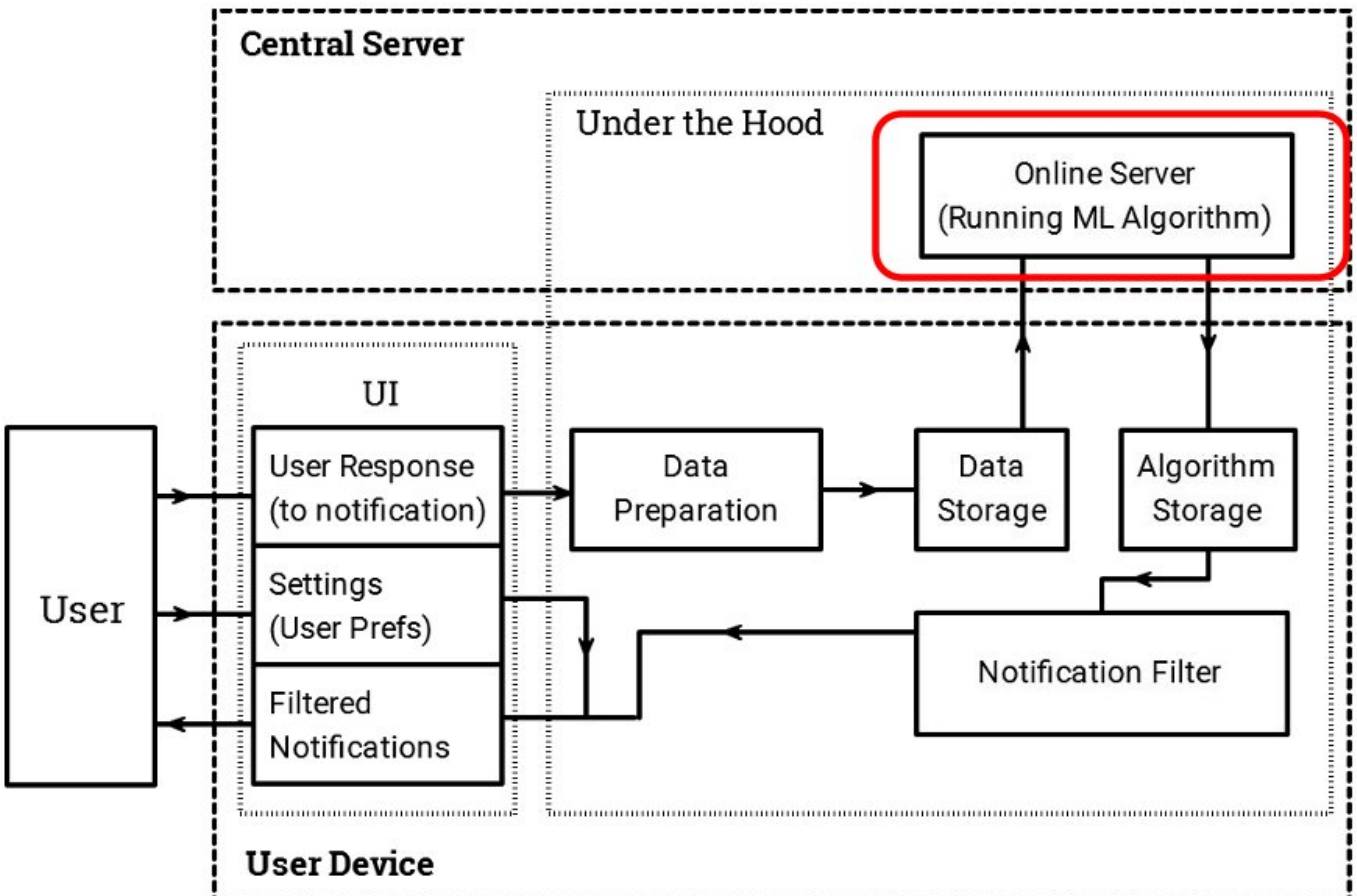
- Guidelines
  - Quell Data Security concerns
  - Data (internet) Usage Minimization

Design - System: Data Storage

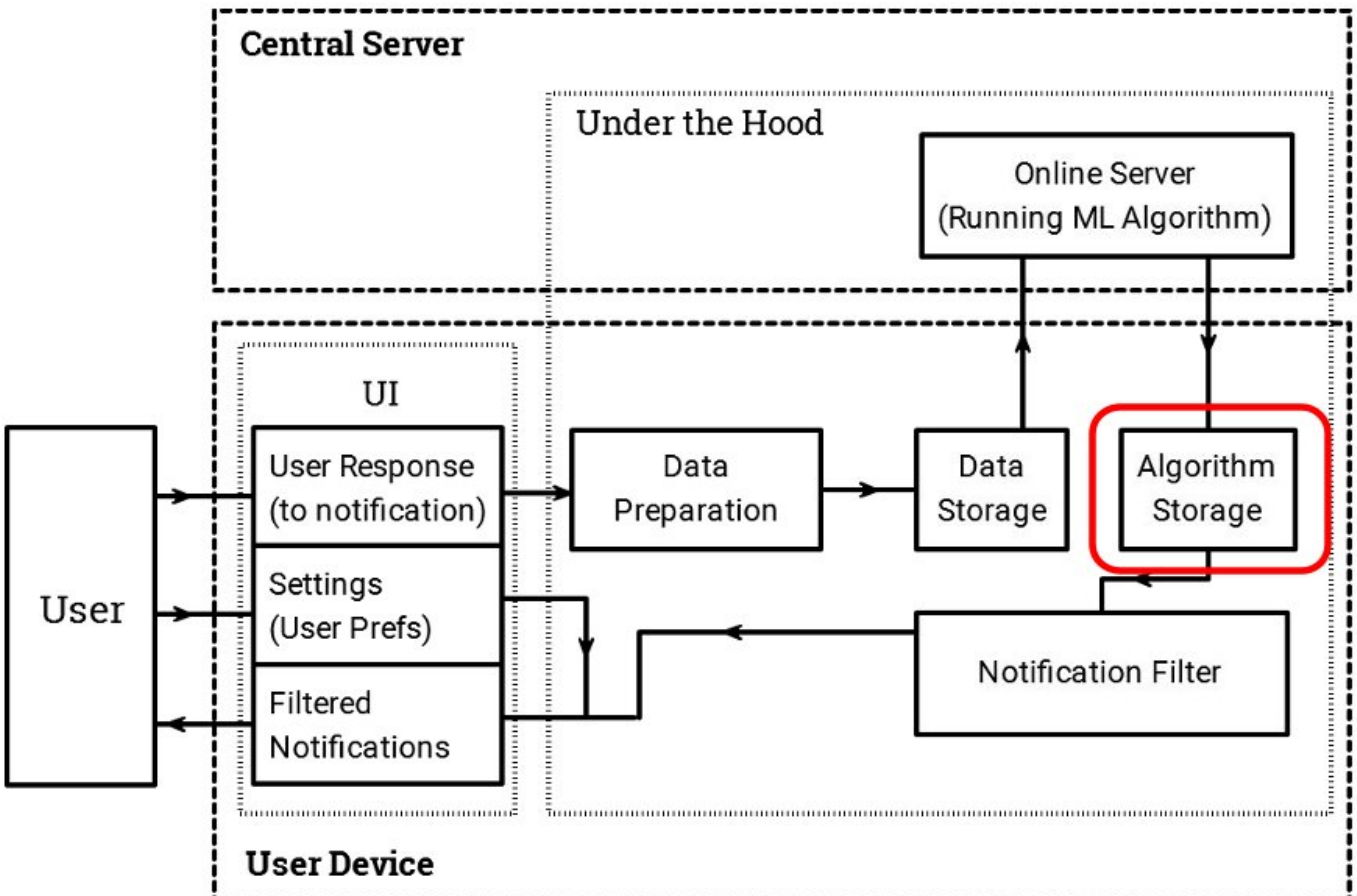
- Guidelines
  - Quell Data Security concerns
  - Data (internet) Usage Minimization



Design - System: Online Server

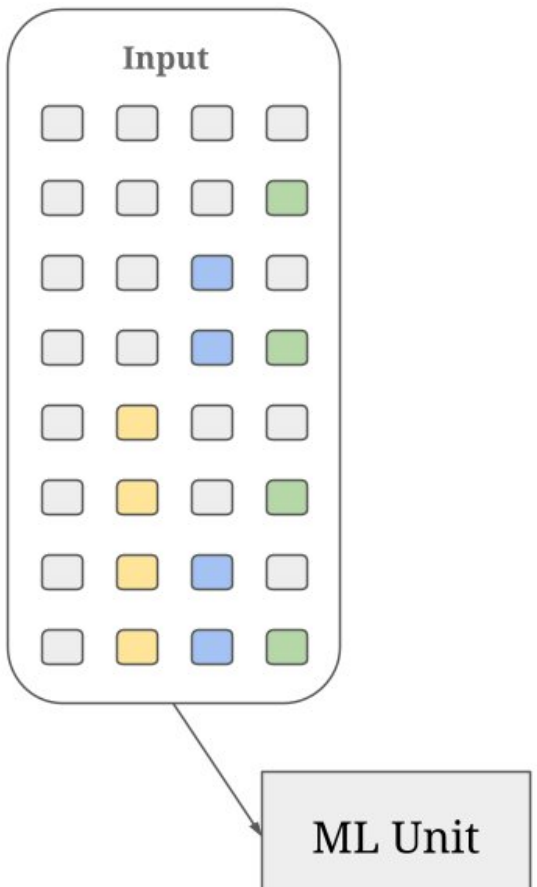


Design - System: Algorithm Export



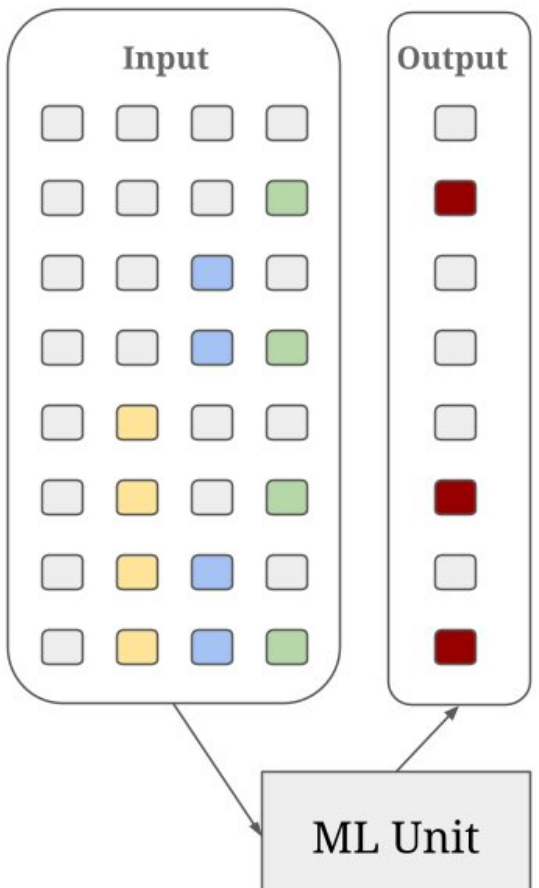
Design - System: Algorithm Export

- User not always online
  - Need to have filtering occur within user device
- Method:
  - All possible permutations of features fed into algorithm



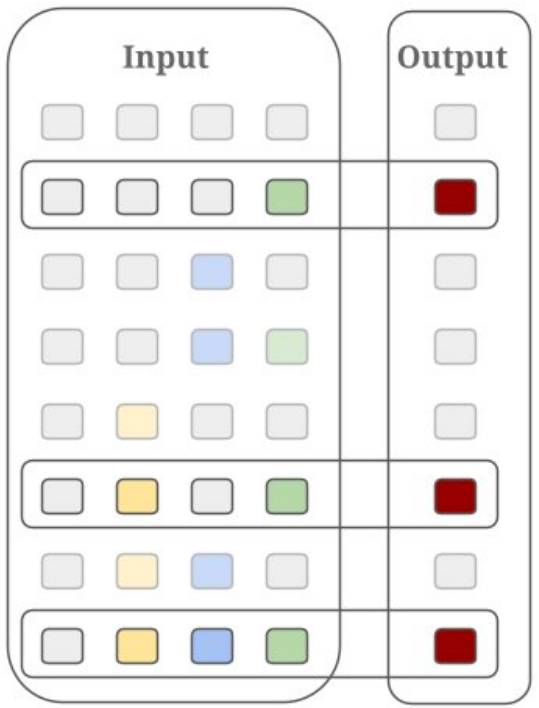
Design - System: Algorithm Export

- User not always online
  - Need to have filtering occur within user device
- Method:
  - All possible permutations of features fed into algorithm
  - Permutations that are marked irrelevant are extracted & exported to user device



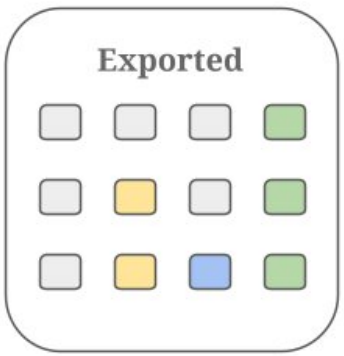
Design - System: Algorithm Export

- User not always online
  - Need to have filtering occur within user device
- Method:
  - All possible permutations of features fed into algorithm
  - Permutations that are marked irrelevant are extracted & exported to user device

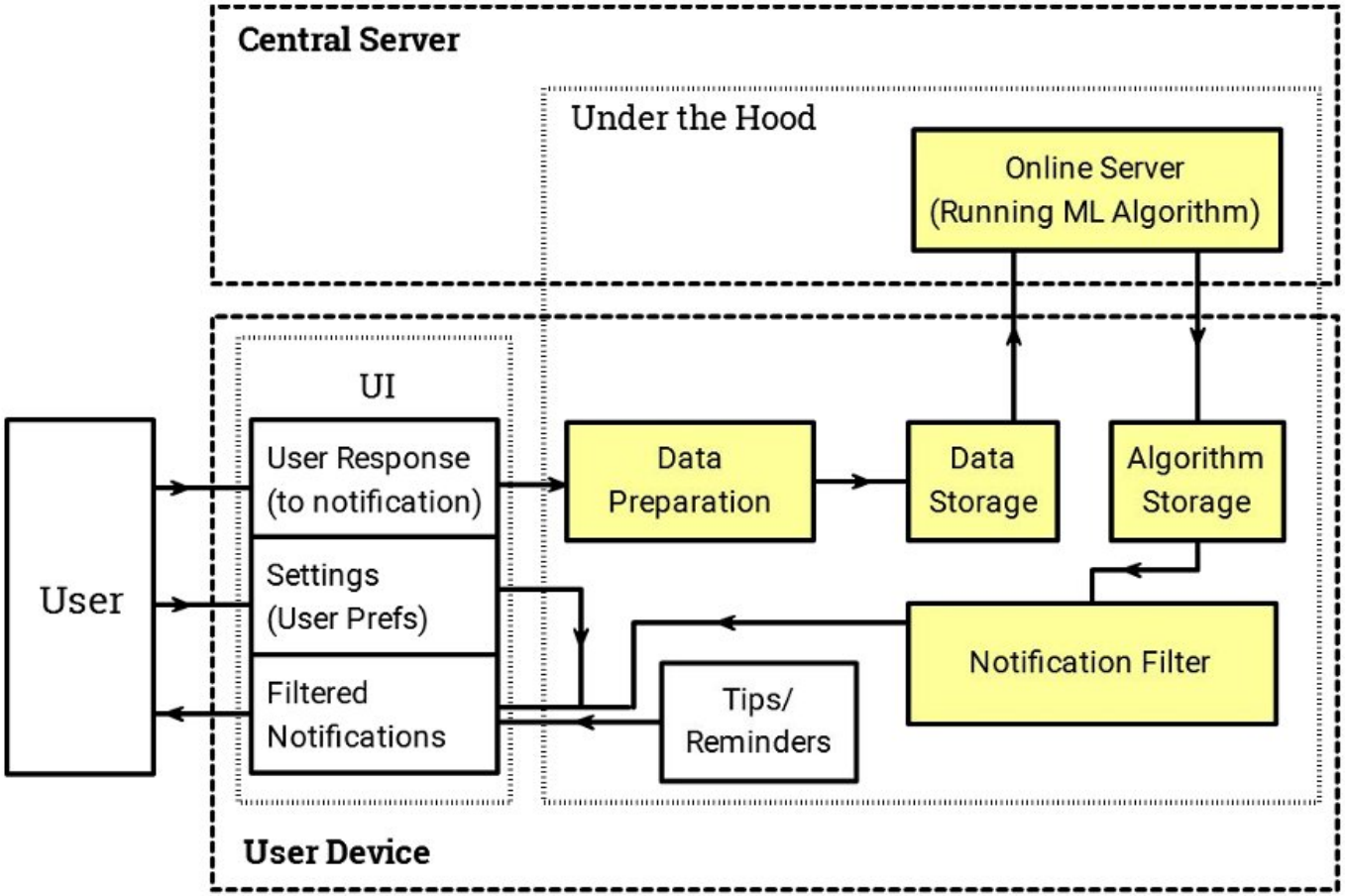


Design - System: Algorithm Export

- User not always online
  - Need to have filtering occur within user device
- Method:
  - All possible permutations of features fed into algorithm
  - Permutations that are marked irrelevant are extracted & exported to user device

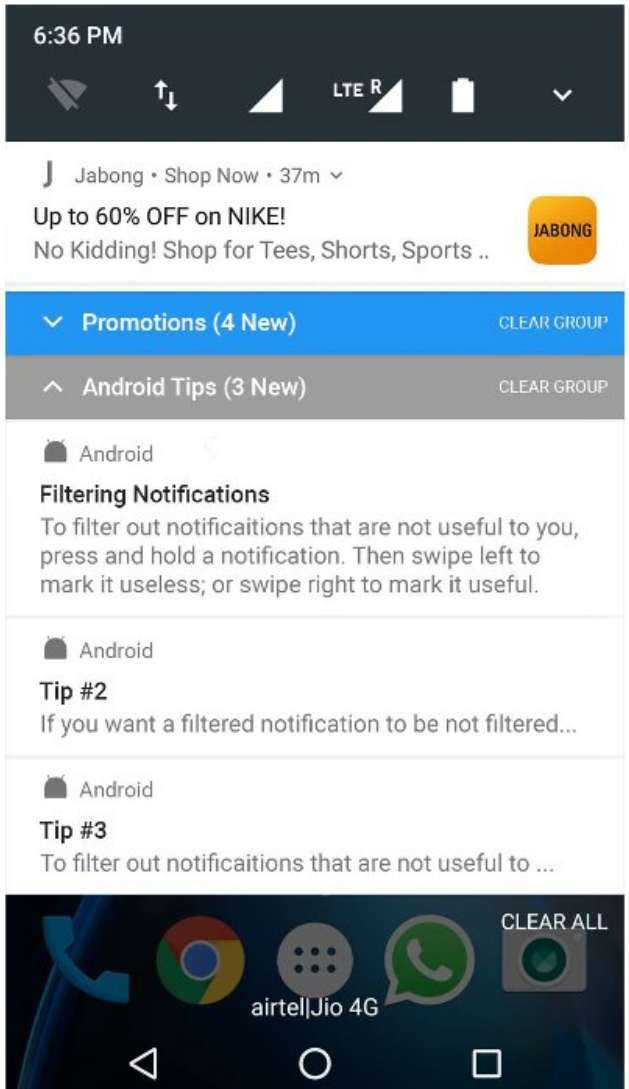


Design: System Prototype

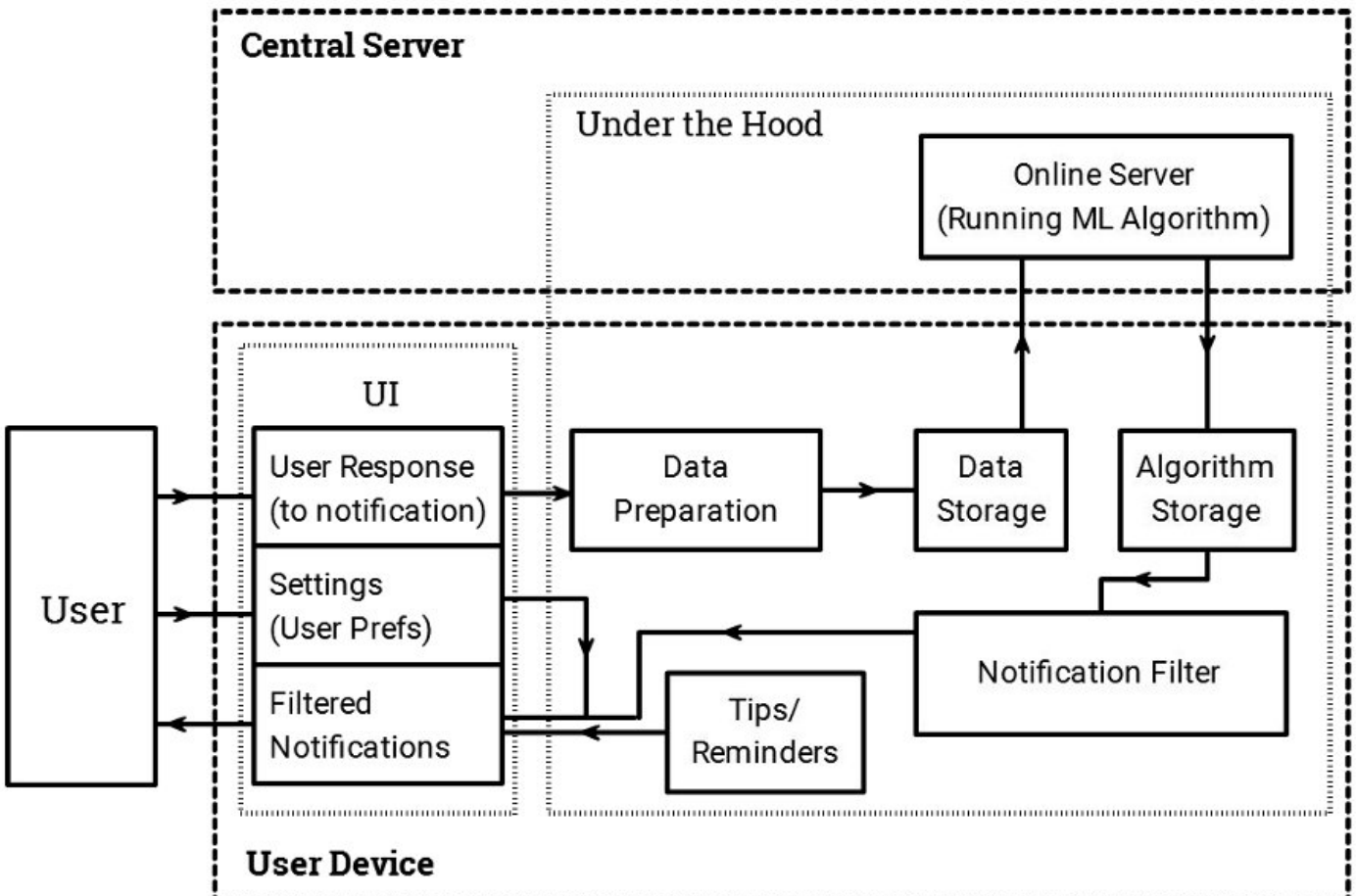


Design: Reminders/Tips

- Help user utilize features in android
- Based on user behaviour
  - Usage frequency of feature



Design: Reminders/Tips



Design: Delayed Delivery

- Timings such that there is minimum break in flow for user
- Possible options:
  a. At a specific time of day
     - Based on initial usage behaviour
  b. Just after a phone call
  c. Just after user quits an app
     - Specifically leisure apps

Monday, May 1 2017

P3 Defence • Now

Evaluation

Design & Evaluation of Algorithm

P3 Review • In 5 mins

Conclusion

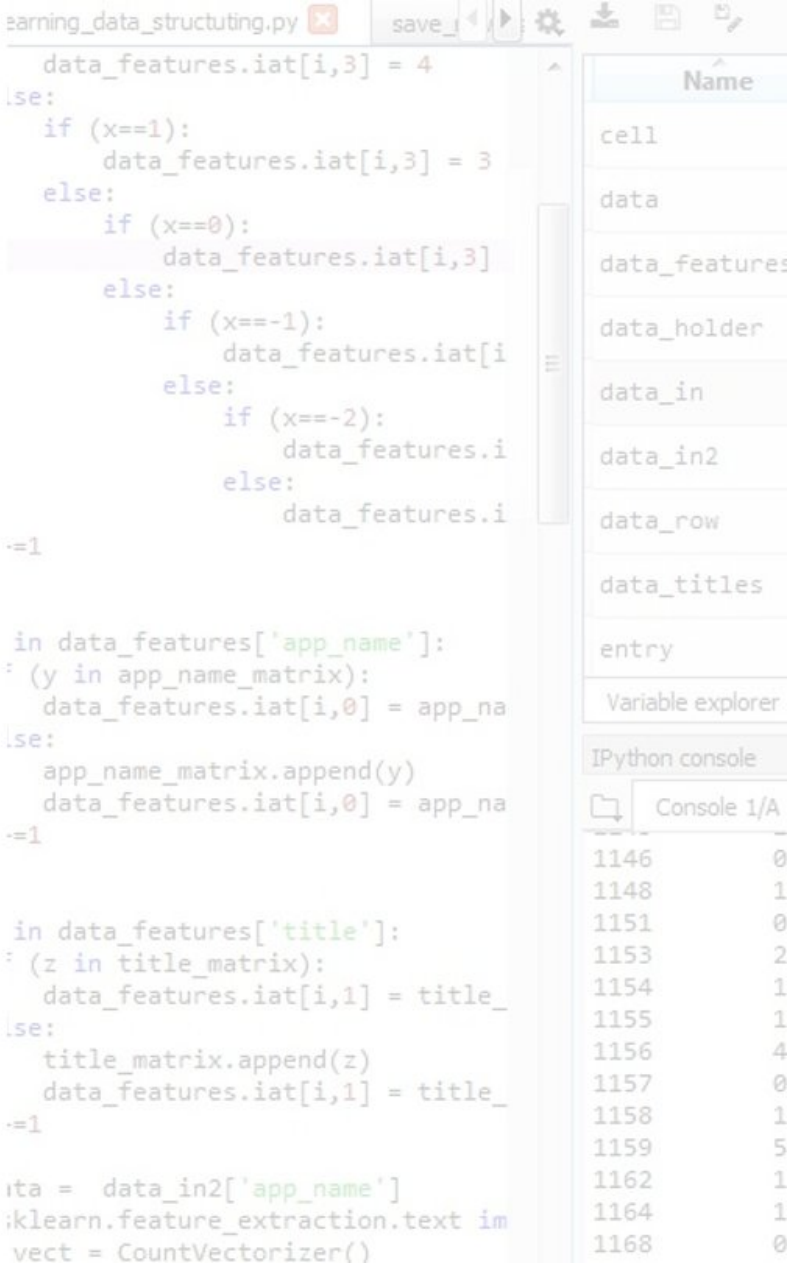
P3 Review • In 7 mins

Thank You

CLEAR ALL

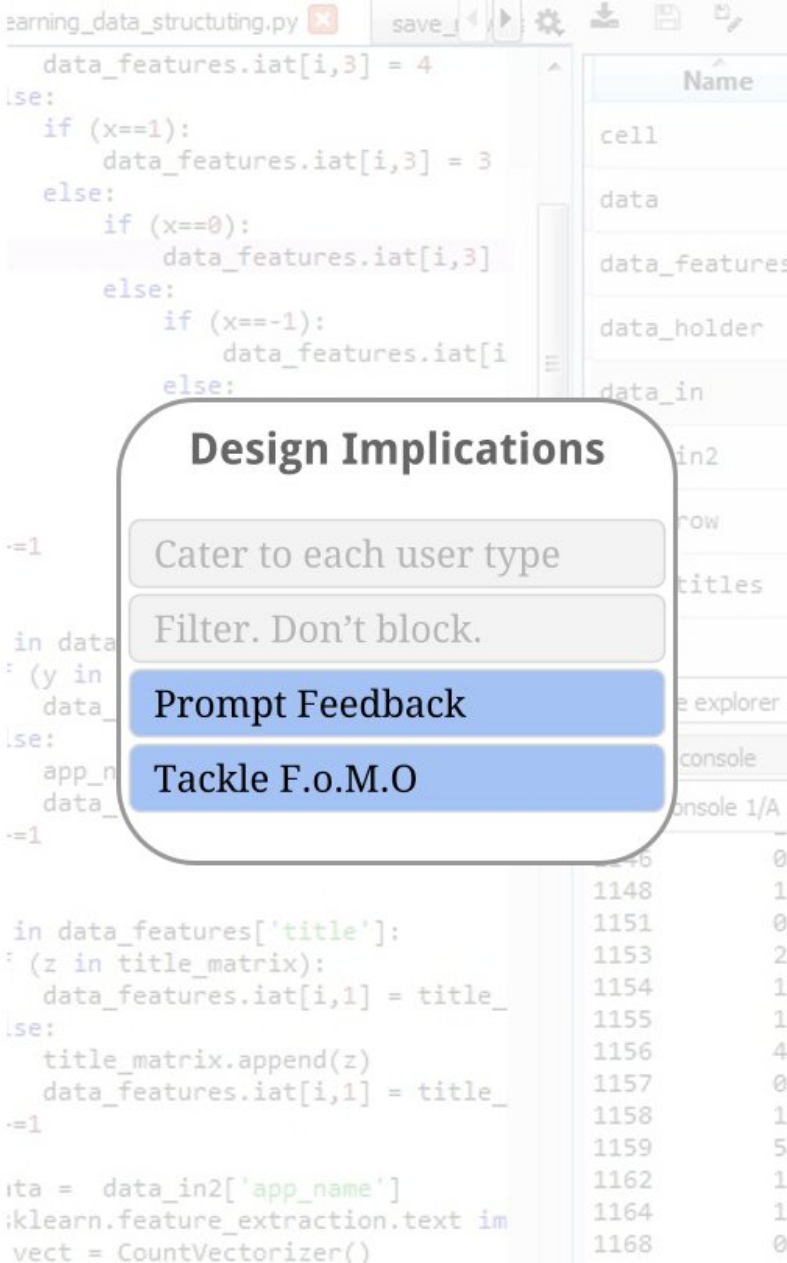
Evaluation: Key Metrics

- % of irrelevant messages filtered
  - Directly feeds into aim of the project (Reduce no.of interruptions)
- % of Relevant messages filtered(incorrectly)
  - Should be minimum
- Number of filtered items with low number of data points
  - Initial stages of use

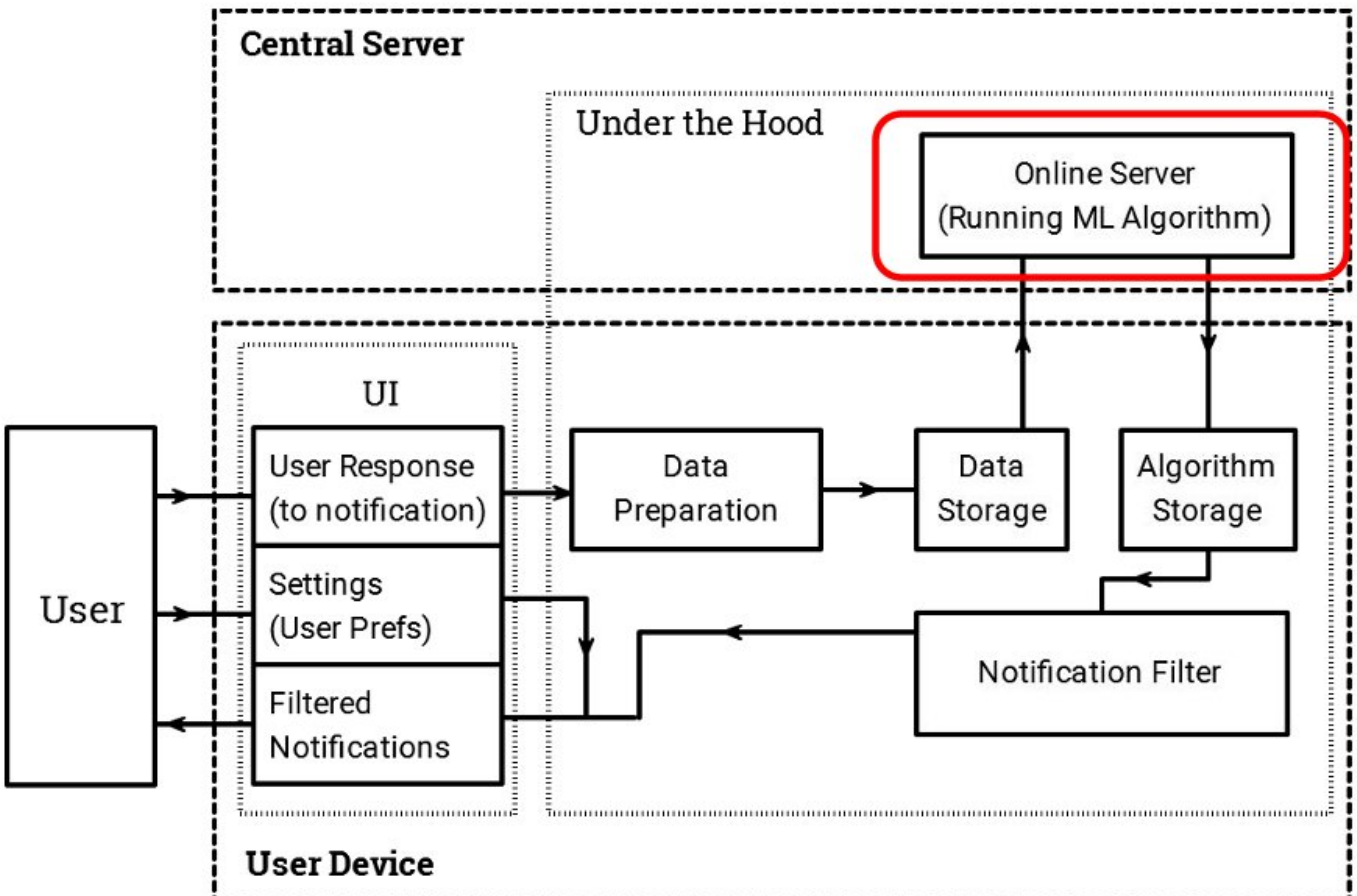


Evaluation: Key Metrics

- % of irrelevant messages filtered
  - Directly feeds into aim of the project (Reduce no.of interruptions)
- % of Relevant messages filtered(incorrectly)
  - Should be minimum
- Number of filtered items with low number of data points
  - Initial stages of use



Design/Evaluation: Algorithm



Design/Evaluation: Algorithm

- Iterative (design/evaluation) process

Design/Evaluation: Algorithm

- Development Details
  - Platform: Python
  - Scikit Learn ML Library

Design/Evaluation: Algorithm

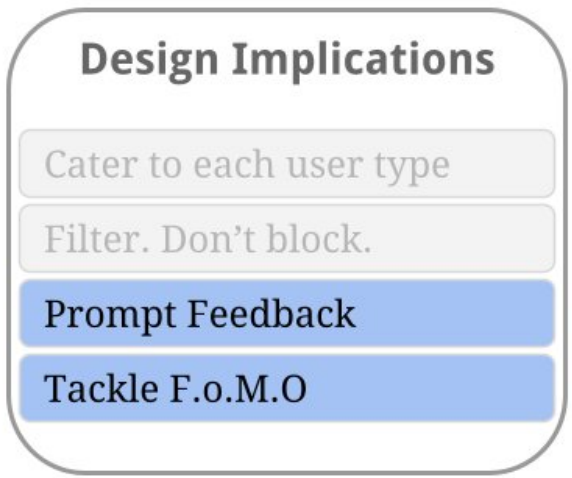
- Main sections of filtering algorithm
  - Data prep
  - ML Method Selection
  - ML Algorithm Export

Design/Evaluation: Algorithm

- Main sections of filtering algorithm
  - Data prep
  - **ML Method Selection**
  - ML Algorithm Export

Design/Evaluation: Algorithm

- Guideline:
  - **No relevant message should be filtered**
    - Else, user builds a habit of checking filtered messages regularly (F.o.M.O)
  - **User should see prompt effect(filtering)**

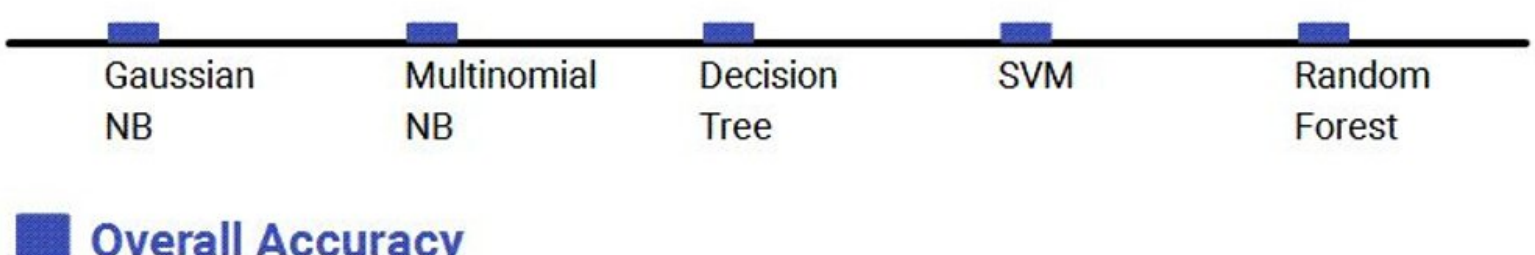


Design/Evaluation: Algorithm

- Methods tried:
  - Naive Bayesian Classification
    - Gaussian
    - Multinomial
  - Support Vector Machine
  - Decision Tree
  - Random Forest
- Primary data used for training and evaluation
  - 60/40 split

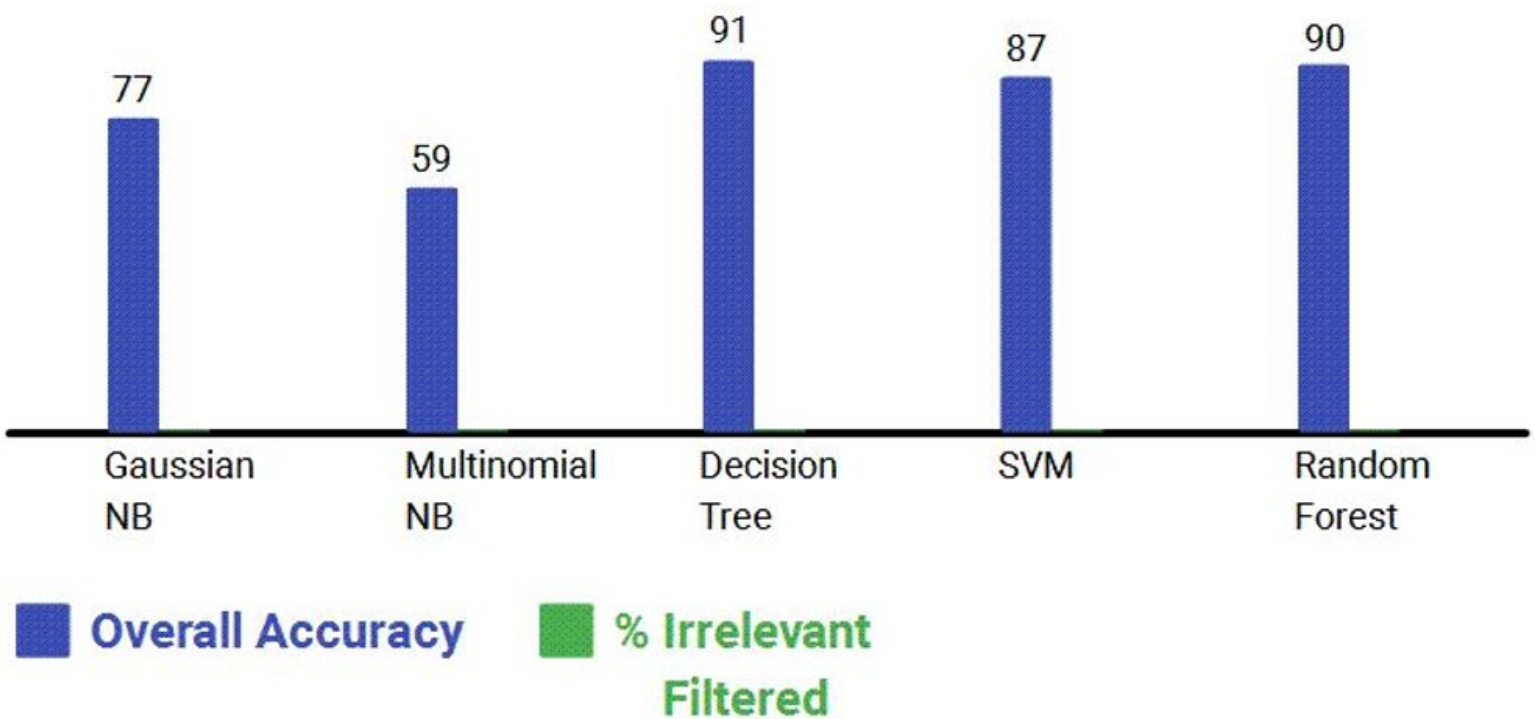
Design/Evaluation: Algorithm

700 Training Points
463 Test Points



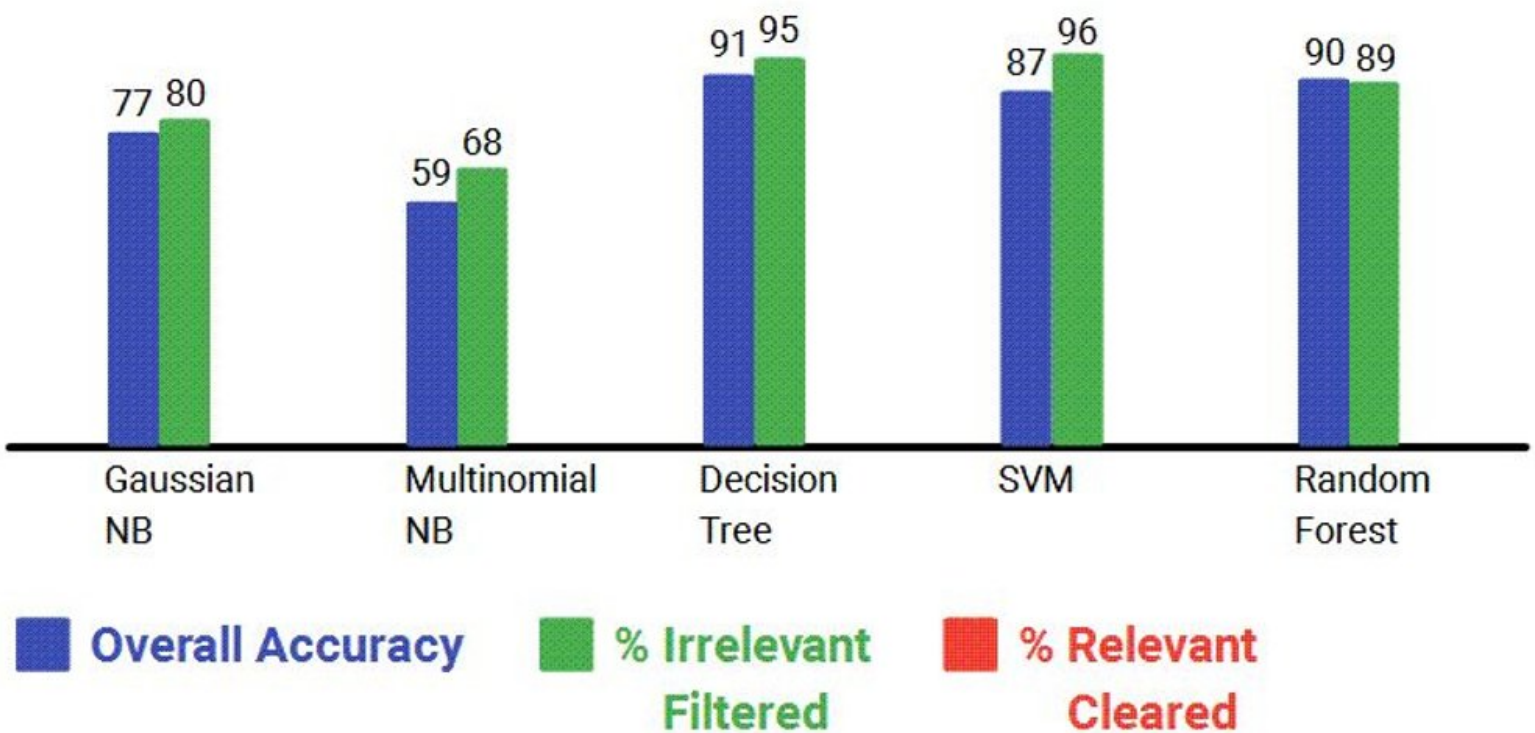
Design/Evaluation: Algorithm

700 Training Points
463 test points



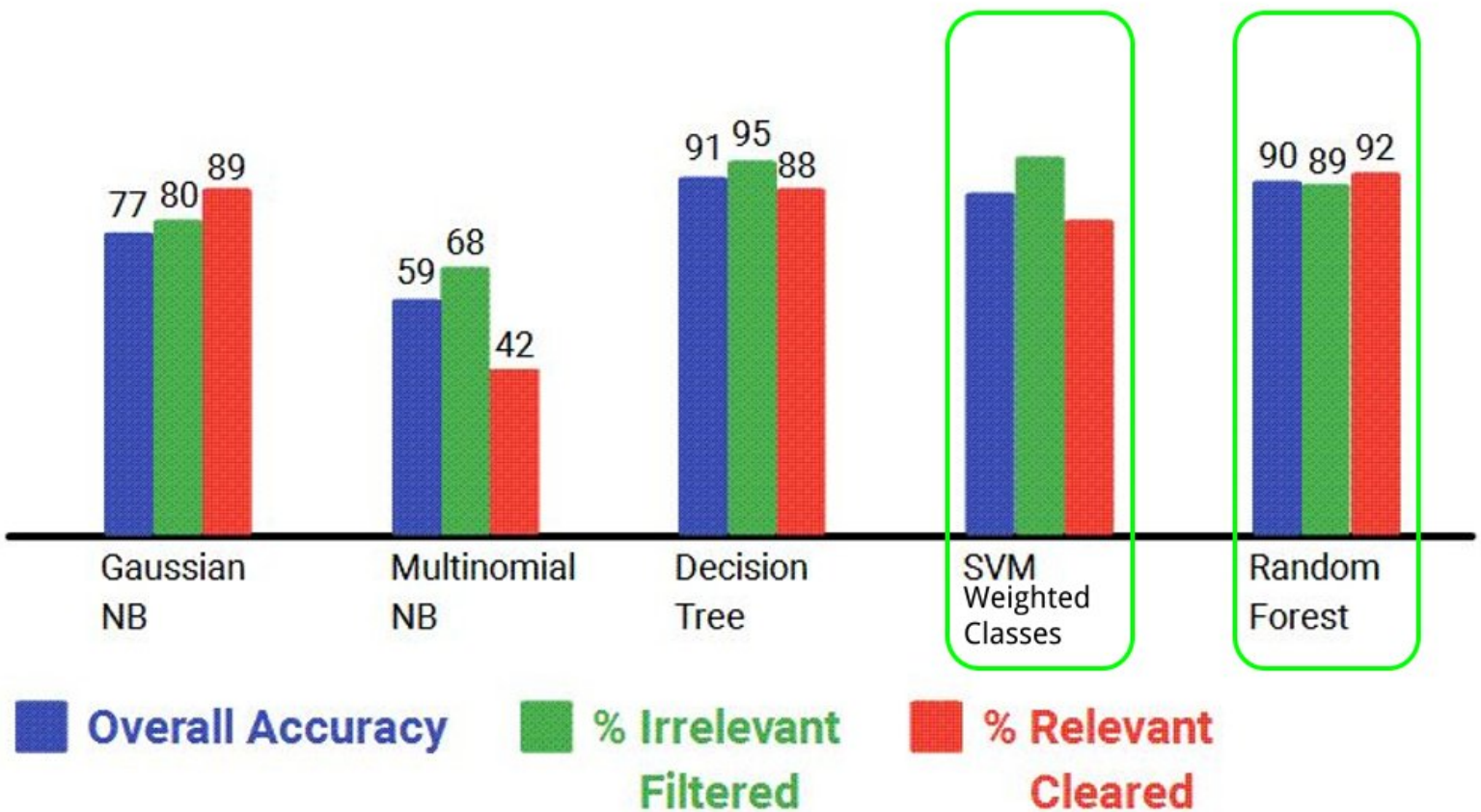
Design/Evaluation: Algorithm

700 Training Points
463 test points



Design/Evaluation: Algorithm

700 Training Points
463 test points

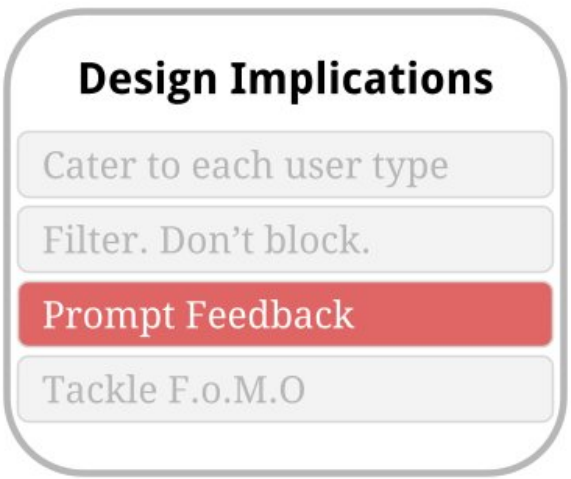


Design/Evaluation: Algorithm

- Early stages of use:
  - Low number of data points
  - Only ML : No filtering

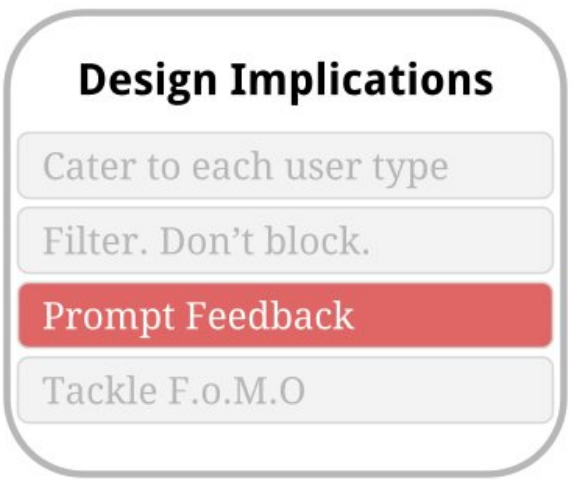
Design/Evaluation: Algorithm

- Early stages of use:
  - Low number of data points
  - Only ML : No filtering



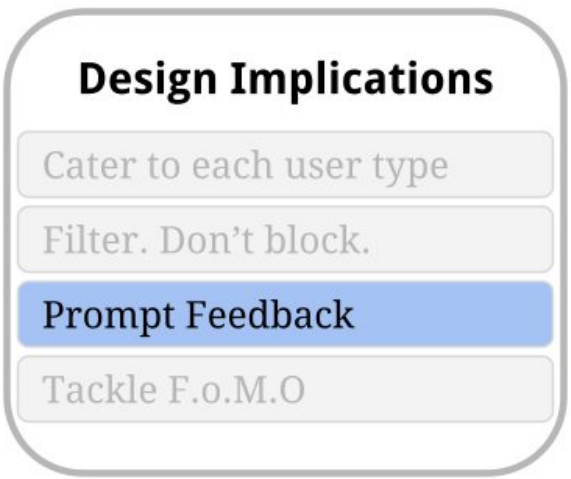
Design/Evaluation: Algorithm

- Early stages of use
- Add-on code
  - Checks for repeated notifications
  - Classifies as last user response



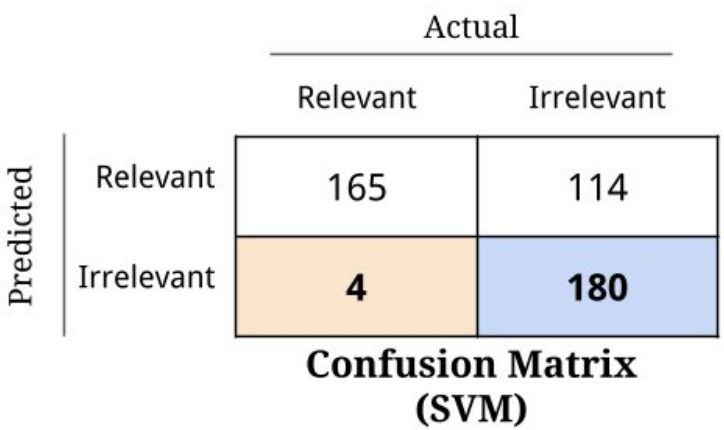
Design/Evaluation: Algorithm

- Early stages of use
- Add-on code
  - Checks for repeated notifications
  - Classifies as last user response
- ML + Add-on code:
  - Filtering starts right away
  - De-activate Add-on after sufficient data points recorded



Evaluation: Performance

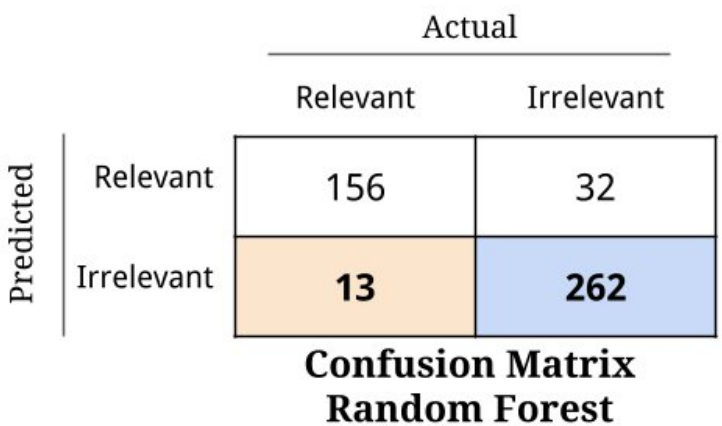
- Performance based on 18 days of training data:
- **SVM (Weighted Classes)**
  - % of irrelevant messages filtered
    - **61.22** % (180/294)
  - % of Relevant messages filtered(incorrectly)
    - **2.8** % (4/169)



Confusion Matrix (SVM)

Evaluation: Performance

- Performance based on 18 days of training data:
- **Random Forest**
  - % of irrelevant messages filtered
    - **89.11** % (262/294) (+27.89%)
  - % of Relevant messages filtered(incorrectly)
    - **7.7** % (13/169) (+4.9%)



Confusion Matrix Random Forest

Evaluation: Performance

- Over a period of 12 days, 180 unwanted notifications filtered
- **15** less interruptions per day

Evaluation: Performance

- Over a period of 12 days, 180 unwanted notifications filtered
- 15 less interruptions per day
- Room for improvement
    - Would get better with time

Monday, May 1 2017

P3 Defence • Now

Conclusion

P3 Defence • In 2 mins

Thank You

CLEAR ALL

Conclusion

- Against the aim of the project (reduce low priority interruptions), smart notification system (SNS) achieved **61(89)%** reduction with **18** days' data. With a false positive rate of **2.8(7.7)%**

- New interactions designed for user-system communication
- UI demo prototype developed

- A functional system prototype designed, developed

Future Scope

- SNS as part of Android OS/ iOS
  - Better evaluation with new interactions applied
- More variables tracked for better filtering performance
- Tweaking ML algorithm for better performance

P3 Defence • Now

Thank You

Existing Methods
For Managing interruptions

- Switch off the phone or put it away

Existing Methods For Managing interruptions

- Switch off the phone or put it away
- Block specific apps:
  - Via OS settings
  - Via 3rd party apps

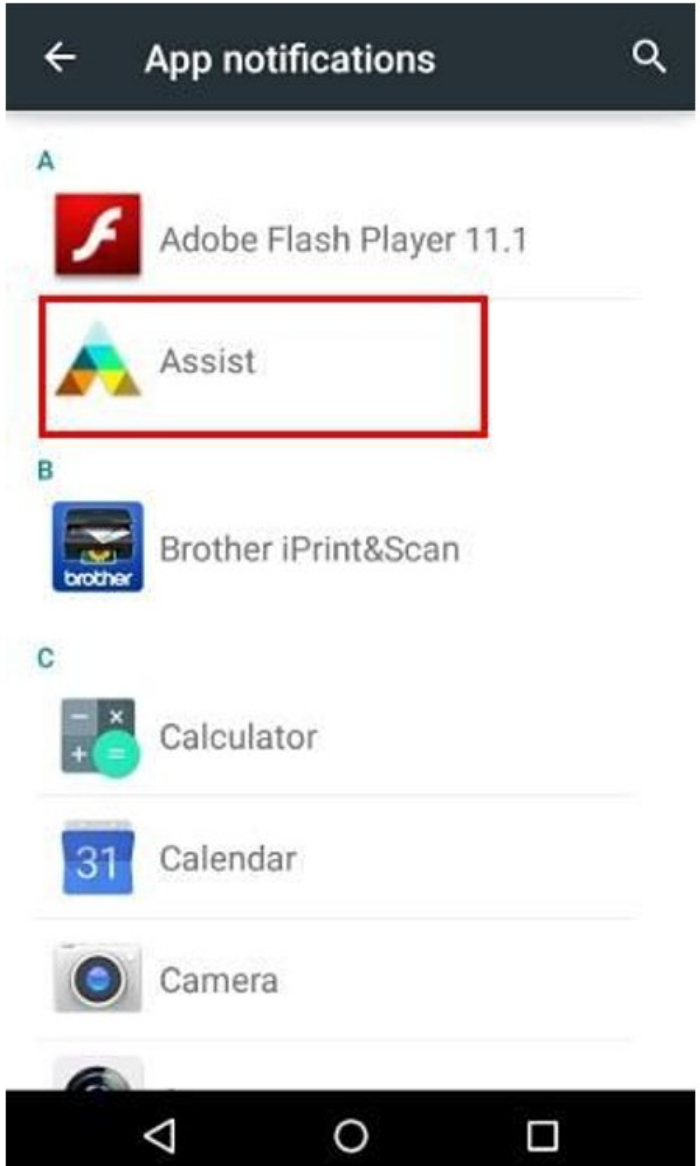


Image Source:
nhttp://gadgetguideonline.com/android/lollipop/how-to-use-android-lollipops-notification-and-interruption/

Existing Methods For Managing interruptions

- Switch off the phone or put it away
- Block specific apps:
  - Via OS settings
  - Via 3rd party apps

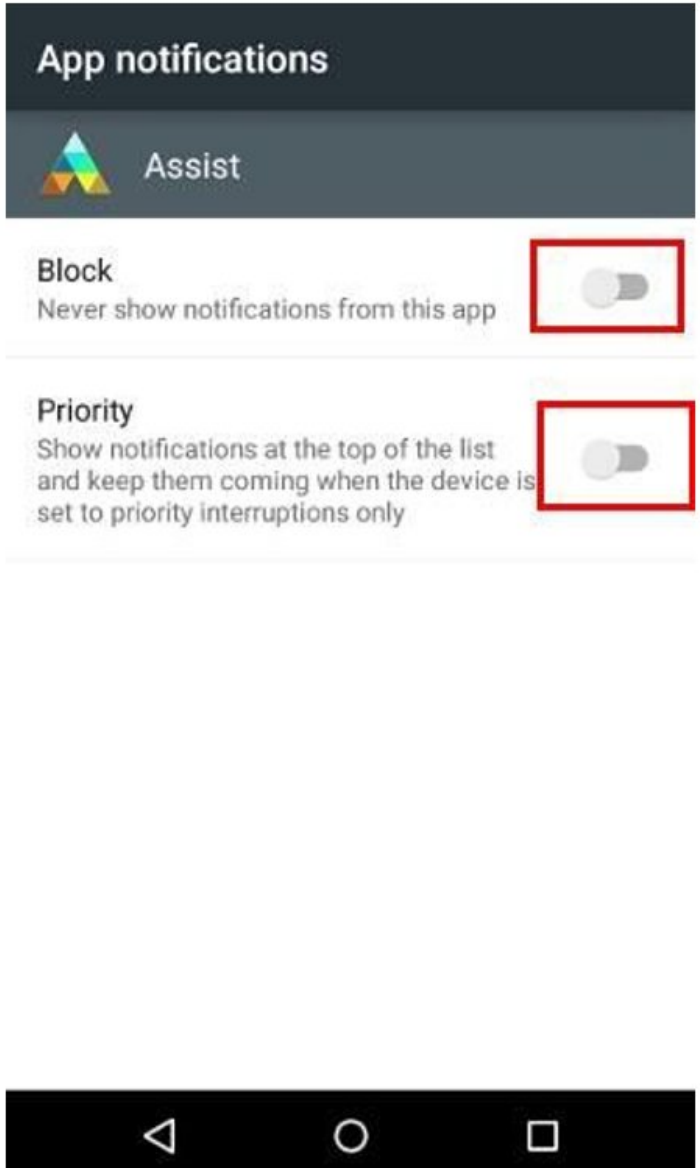


Image Source:
nhttp://gadgetguideonline.com/android/lollipop/how-to-use-android-lollipops-notification-and-interruption/

Existing Methods For Managing interruptions

- Switch off the phone or put it away
- Block specific apps
- Priority/DnD modes
  - OS supported (Android Lollipop and above)
- 3rd Party apps

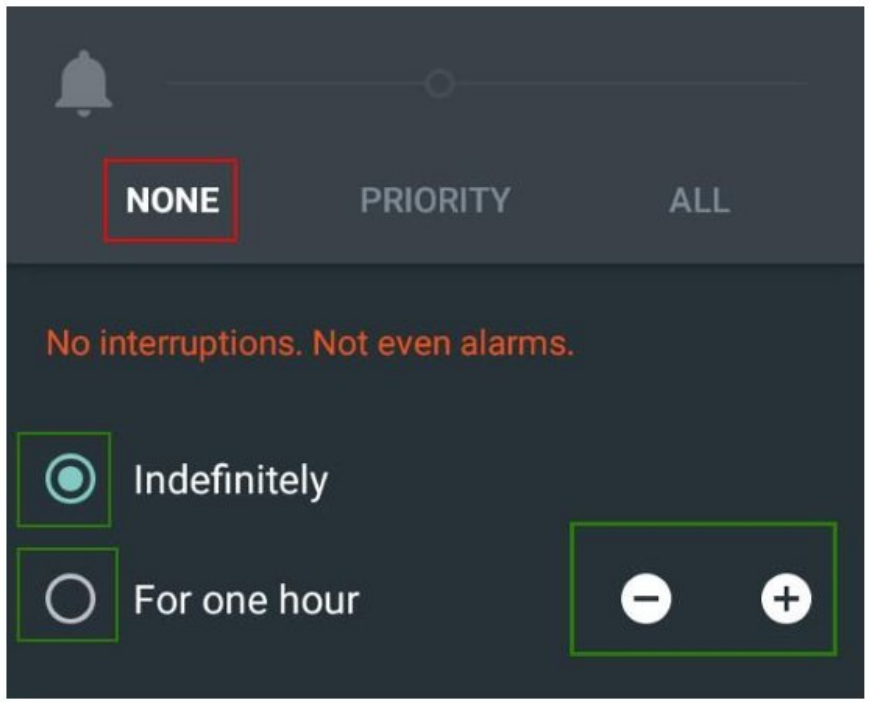


Image Source: nhttp://gadgetguideonline.com/android/lollipop/how-to-use-android-lollipops-notification-and-interruption/

Existing Methods
For Managing interruptions

- Drawbacks:
  - User may miss out on important/urgent interruptions

Existing Methods
For Managing interruptions

- Drawbacks:
  - User may miss out on important/urgent interruptions
  - User is required to know of the existence of such methods

Existing Methods
For Managing interruptions

- Drawbacks:
  - User may miss out on important/urgent interruptions
  - User is required to know of the existence of such methods
  - There is no fine control (except for phone calls/messages - which requires manual setting up)

Existing Methods
For Managing interruptions

- **User may not be aware of the effect interruptions have on their productivity**