

Project 3 Report :

PhonBk

Design of a smarter context based
Phonebook visualization for mobile
devices

Akshay Kore

Interaction Design
M.Des (2014-16)

Guide: **Prof. Anirudha Joshi**

Industrial Design Centre

Indian Institute of Technology, Bombay

Approval Sheet

The Interaction Design Project II entitled “Data Visualization for Mumbai” by Akshay Kore, Roll Number 146330007 is approved, in partial fulfillment of the Master in Design Degree in Interaction Design at the Industrial Design Centre, Indian Institute of Technology Bombay

Guide :



Internal :



Chairperson :


26/7/16

External :



Declaration

I declare that this written document represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/ source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.



Akshay Kore

146330007

Industrial Design Centre,
Indian Institute of Technology Bombay

May 2016

Acknowledgements

I would like to thank Prof. Anirudha Joshi for his guidance and support through the tenure of this project. I am grateful to Prof. Venkatesh Rajamanickam, Prof. Girish Dalvi, Prof. Ravi Poovaiah at IDC for their valuable feedback and suggestions.

I would also like to thank Mr.Suresh Chande from Microsoft Research for his feedback at different stages of the project. I am grateful to Prasad Ghone, Jayati Bandyopadhyay, Dileep Mohanan, Indrajeet Roy and Sagar Yende for their support and feedback.

Thank you to The Library. The Internet. Stackoverflow and Quora.

Abstract

With the advent of smartphones and social media it is common to have over a one thousand contacts in one's phonebook. There is a problem of information design for management of contacts since the modern phonebook contains phone numbers as well as details for other mediums of communication such as email, Facebook, WhatsApp, etc. Despite the advancement in computing power of our phones, our phonebooks have been slow to adapt. Most current phonebooks are based on the same design of the phonebook for a feature phone. In the vast sea of contact information, how does one find which are the most relevant contacts?

Data from Primary studies, user studies and secondary research points heavily towards the Pareto Principle which suggests that only 20% of the information is important. The same principle can be applied to the task of contact visualization as well. Importance of a contact can be determined from usage patterns of calling, location information as well as personal details of the contact. This context aware data can be used for contact suggestion. It was found in the primary and secondary research that certain features of the phonebook such as favorites and groups are redundant and not many users seem to be using them.

A preliminary design was created which augmented the structure of the existing phonebook with a contact suggestion widget. The suggestions in the widget are updated periodically based on the user's location, time, day and date. A heuristic evaluation of the idea presented the need to use realistic data. Problems were found with navigation in the existing phonebook and the text input based search of contacts was found to be redundant. It was also thought that the suggestions widget might be beneficial in scenarios when a very small screen real estate is available in the case of wearable devices. The phonebook could also encourage exploration of contacts.

The Phonebook is a ubiquitous application in the smartphone and redesign of the phonebook need not be a complete overhaul. A final idea was developed through a series of smaller design interventions in the navigation, microinteractions and the user interface of the phonebook application. The concept named 'PhonBk' was designed around the goal of finding the most important contact at the moment. This is done by means of a suggestion widget in the home screen of the application, a frequency based suggestion in the contact list and an advanced navigation feature which enables filtering and sorting of the call log.

An evaluation was conducted with 7 users to check for the accuracy of suggestion of contacts and the advanced navigation feature was evaluated for its usability. It was found out that the suggestions of contacts was fairly accurate, however the usability of the same needs to be tested. In case of the advanced navigation feature, it was found that users faced great difficulty in the usability. The advanced navigation feature was redesigned based on the inputs from the usability evaluation.

The PhonBk project is an attempt to re-design the phonebook application which is already ubiquitous across all current smartphones. The project attempts to make important incremental improvements in the current phonebook by means of smarter suggestions based on the context of the user, through an advanced sorting and navigation feature and other design interventions in the visualization as well as organization of contacts. The end goal is to make improvements in the fundamental task of finding the most relevant contact at any moment.

Table of Contents

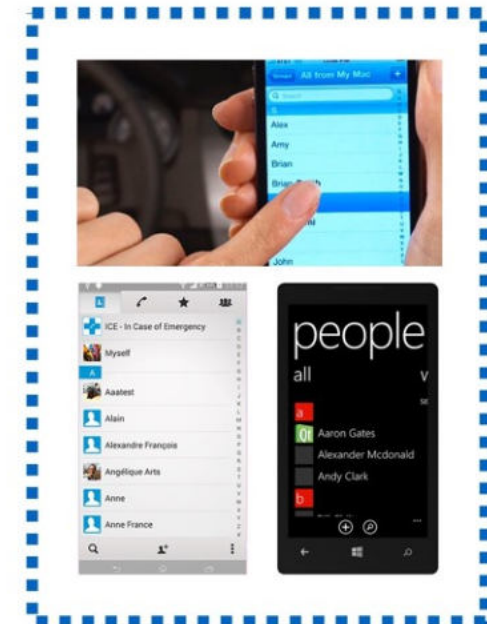
Introduction	8	Final Design Idea - PhonBk	44
		Information Architecture	45
Secondary Research	9	Design Details	46
Usability of the contact list	9		
Pareto Principle : 80/20 Rule	9	Evaluation	58
Contact Importance	10	Suggestion mechanism	58
Communication Medium Recommendation	14	Advanced sort and filter	59
Case Studies	15	Re-Design post evaluation	63
Non-conventional Case Studies	16		
Conventional Case Studies	19	Conclusion	65
Overview of Case Studies	29		
User Studies	30	References	66
Redefined Design Brief	32		
Preliminary Design Idea	33		
Contextual Prediction Widget	33		
Heuristic Evaluation	35		
Design Ideas	36		
Design Ideas - Views	37		
Design Ideas - Navigation	40		

Introduction

These days, it is normal to have over 1000 contacts in a phonebook which include phone numbers, emails and other data. We are in touch with some of these contacts frequently. We use many others rarely. The current visualization of the phonebook data is quite old, and is essentially based on a feature phone. This project is about creating a solution that makes use of high resolution touchscreens and sensors in mobile devices to enhance the user experience.

The design of the phonebook has largely remained the same over the past years. Navigation and finding contacts is still based on Alphabetical Lists, Favorites, Call Logs and in some instances frequency based defaults. Our devices have become exponentially smarter, but not our phonebooks. The term 'Mobile devices' has been loosely defined. Contact lists are one of the most frequently used

applications on mobile devices. Users are reluctant to modify or delete already created contacts and the device's increased storage capabilities prevent the need to do so. Contact lists thereby become increasingly large sometimes measuring more than a thousand entries. The phonebook application is fairly ubiquitous across mobile devices. A non-linear design process has been followed to come up with a design intervention and then conducting an evaluation with the users. The end solution is the design of a smarter phonebook that provides information about the most important and relevant contacts through suggestions and recommendations as well as advanced filtering and navigation features.



Secondary Research

The following page shows different ideas and strategies that have the potential for making the Phonebook smarter. The ideas range from design of the User Interface to Designing algorithms, using machine learning techniques to solving navigation and User Experience problems.

Usability of the contact list

Mobile devices often store a large repository of contact information which aggravates the problem of accessing the desired contact quickly. The users have to deal with fairly lengthy interactions often resulting in frustration due to errors¹.

The contact list application is arguably one of the most frequent function of the mobile device². It was found in a paper by Andreas et al. that users would avoid searching for a contact manually (by typing in the search box) and instead look at the call list first, particularly if it was someone they remembered having talked to recently. However this may not always be the best strategy to adopt as shown by Bocker and Suwita in their 1999 paper that although almost all users had no issues finding and calling a given contact from the phone book (94% success), this rate dropped significantly to 73% when asked to find and call the same contact from a call list³.

Pareto Principle : 80/20 Rule

The Pareto principle⁴ states that out of all the available information, only about 20% of it is valuable. Out of all the contacts in the phonebook, only a handful of contacts are important at any given point of time. The number of important contacts at any given

moment generally ranges from 10 -20 contacts, however different studies suggest a different number. Overall, only around 20% of the contacts out of the entire contact list is important for the average person. In a paper by Whittaker et al. 17 users were asked to rate the importance of their email contacts and found out that only 19% of these contacts were rated as important (Pareto Principle)⁵. A study of calling patterns conducted by Vassilios et al. with 28 subjects showed that 46% of the contacts were not used in the previous 6 months or never used⁶. The Pareto principle arises again in the study with approximately 19% of the contacts being used at least once weekly.

Another paper by Jung et al. investigates the efficiency of accessing and differentiating important contacts⁷. They found that users reacted positively to being able to access quickly the top 10 contacts in terms of communication frequency in addition the contacts whose birthday / anniversary was coming up. Their findings align with the hypothesis for the need for differentiation of important contacts.

A paper by Vassilios et al. on examining the relationship between the size of the contact list and the number of important contacts contained therein found no statistically significant correlations between the parameters⁸. The number of important contacts at a particular moment remains relatively steady despite the increase in the size of the contact list. According to the findings in the paper after analysis of real user data, quick access to approximately 10 important contacts needs to be provided.

Contact Importance

User Defined Importance

Most contact book applications on smartphones have a feature of assigning a favorite/star tag to a contact. Favorites have become a ubiquitous feature for the contact book application. A study by Vassilios et al. showed that for most users who use the favorite feature, the percentage of outgoing calls to these favorites is significantly high⁹. For 60% of the users in the experiment that used the favorites feature, more than 50% of the outgoing calls were to these starred contacts. This reinforces the significance of personal preference in predicting the importance of a contact however there were a number of candidates who did communicate very much with their starred contacts on the phone.

Users often find clever ways of segregating important information, such as placing non-important information in a collection or placing the letter 'a' at the beginning of a filename so it would be placed on top of the list and is thus quickly accessible¹⁰. Some users use numbers to impose different levels of hierarchy (eg. 1mummy, 2prasad)¹¹. Apart from user definition, contacts can also be segregated based on their importance derived from patterns of calling along with the context of the user.

Recency and Longevity

With mobile devices providing a very large storage space to store contact details, users quickly build large contact repositories, which include many contacts that are needed only temporarily or are very infrequently used as well as unrelated items such as PINs, passwords and other information¹². Frequency of interaction is a strong indicator

of the importance of a contact¹³. Recency of the interaction is also a prime factor for deriving importance. Important contacts should have interacted with the user more recently than unimportant ones¹⁴. Important contacts should interact over longer periods than unimportant ones¹⁵.

The more recent the interaction, the more long term the interaction, the more important is the contact. Recency may indicate temporary importance while longevity indicates certain importance. In a paper by Whittaker et al. the hypotheses suggests based on lab experiments and findings that the more the interaction with a contact, the more important the contact is and recent communication is more likely to come from an important contact¹⁶. Call duration is another important parameter that is useful for prediction tasks¹⁷.

Reciprocity

It was found that important contacts have a greater reciprocity than unimportant contacts¹⁸. Lack of reciprocity indicates that the contact is not an important one. In cases where the contact sends a message to the user but never receives any reply back, the contact is unimportant.

Context Awareness

The importance of a particular contact may also depend on the current context of the user. Some contacts are important in certain locations, at certain times, often a combination of both. The importance may also depend on the closeness or relatedness of a contact to another contact. All this information can often be coupled with recency and longevity patterns to find out who is important to me right now. This data can be used to predict the most important contacts and visualize the same for the phonebook.

Bayesian networks are one of the efficient tools to handle context awareness. Jun-Ki et al. trained Bayesian Networks using user schedules, location, statistical patterns, anniversary data to create a recommender system for a smart phonebook¹⁹. A method of using a context tree was employed to achieve the desirable result. The proposed method expands each user context into a tree. When a user does an action on the mobile device, his context is accumulated in as a frequency of nodes in a context-tree. The node with the highest frequency/score is recommended to the user. Increase in context-history would increase the accuracy of the system.

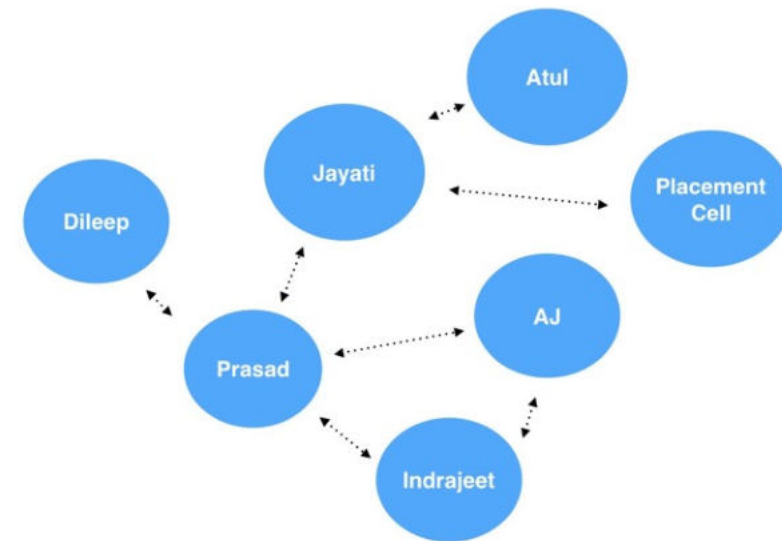
Improving the contact list using context awareness was also examined by Oulasvirta et al. The major finding was that context awareness may be used to present relevant information to the user while leaving the ultimate control over the course of action, thereby augmenting decision making²⁰.

Predicting the next important contact

This idea derives inspiration from the Phoneman paper²¹ where you predict the next possible action after the current action. For example, after using the phonebook the next action would probably be using the calendar or creating a reminder. In similar fashion, the contact data reveals patterns in calling certain individuals. After calling a specific person, I am prone to make the next call / interaction with a specific group of people, in the same fashion as creating bayesian networks. This data can be used to predict and suggest the next possible contact.

Contact Information

The smartphone has the potential of storing large amount of metadata for each contact. This opens a range of possibilities in design, prediction, learning and a number of other fields. This information becomes very important to design context awareness for the phonebook.



PREDICTING THE NEXT IMPORTANT CONTACT BASED ON CURRENT INTERACTION. ONE CAN OBSERVE GROUPS BEING FORMED IN THE SCHEMATIC DIAGRAM

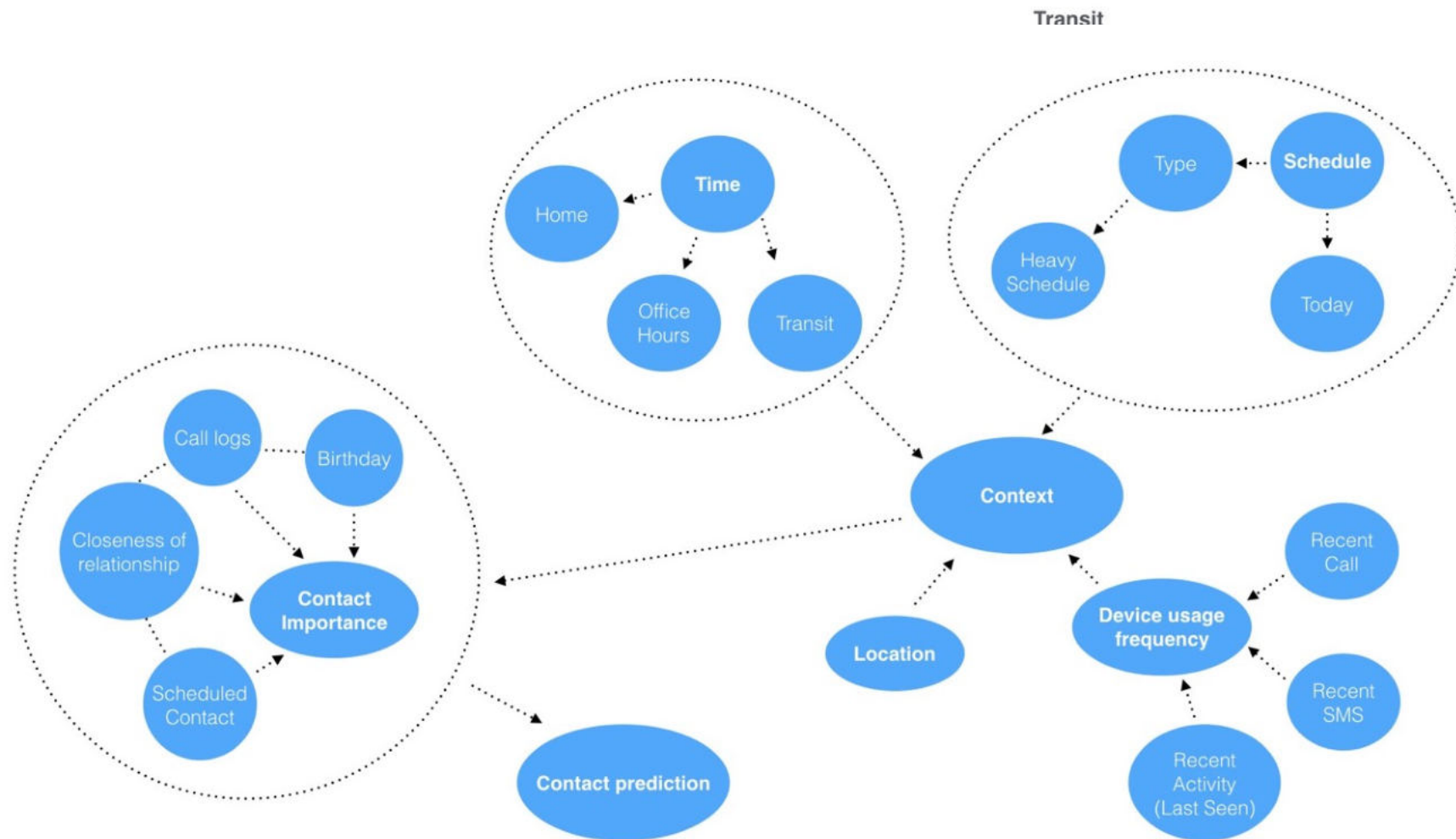
Types of contact information that currently exist in major Phonebook applications to improve context awareness of the phonebook app:

General

- Name
- Surname
- Sex
- Carrier
- Important Dates (Birthdate, Anniversary)

Demographic

- Expertise (Doctor, Architect)
- Social Position (Sir, HOD, Ma'am)



DIFFERENT TYPES OF USER CONTEXTS IN THE CONTACT PREDICTION MECHANISM

User Defined

- Favorites
- Schedule (Meeting, Dinner)
- Relationship (Mom, Dad)

Modes of Communication

- Phone Numbers
- SMS
- WhatsApp
- Email IDs
- Facebook
- Twitter
- Website
- Blogsite

Context

- Physical (London, Mumbai)
- Cognitive (Busy, Available)
- Social (With 'prasad')
- Temporal Location (Time, Date, Timezone)
- Interactional (Used phone 10mins ago)
- Co-ordinates (Latitude, Longitude)

Communication Medium Recommendation

Xiwei Zhuang et al. describe a method of recommending the most appropriate medium to communicate with a particular contact²². It is based on using data of user habits, schedule and phone status (on silent, last interaction) of the person being communicated with. The user of the application gets to know the status of the person being contacted with. These methods however raise a number of privacy and security concerns for contacts in the phonebook.

This type of recommendations for the communication medium to be used can also be generated based on the user's context, patterns of calling, carrier information, location rather than extracting data from the person being called. For example, the system would detect the carrier information and location of the caller and person being called (based on phone number) and suggest what medium to use. If it is an international number, it could suggest that an email or a WhatsApp call would be a cheaper option. In case of dual sim mobile phones, it could recommend which sim to use based on both the parties' carrier information and the current data plan. However, communication medium recommendation is also dependent on the current context of rates of phone calls vs charges for the internet. For example, a WhatsApp or Skype calling is cost effective over an International phone call currently although this was not the case some years ago.

Case Studies

The following section discusses different types of contact management and phonebook applications. The case studies reveal common patterns in different phonebook applications discussed at the end of the section.

Non-conventional Case Studies

A critical analysis of non-mainstream, non-standard applications. These applications in many cases are parts of research projects.

Conventional Case Studies

A critical analysis of mainstream applications used by at least 1,00,000 people. These include device specific default phonebook applications as well as popular downloadable phonebook applications like Truecaller.

Non-conventional Case Studies

ContactMap²³

Contact Map is a social desktop application enabling organization of the users' contacts in groups. The application uses people as a primary unit of interaction rather than messages in the case of traditional applications. The Visual Interface replicates many functions of a shared physical workplace. It creates a personal social desktop that represents the people who are central to the user's work and personal life. It is a visualization of important contacts and their relationships.

Evaluation of the tool indicated that users performed certain tasks like map construction and identifying important contacts better on the ContactMap tool over traditional email systems. However, it was felt that the ideal interface should contain both the person centric as well as the message centric view. The following statements point to the tradeoffs in using the ContactMap approach.

Pros

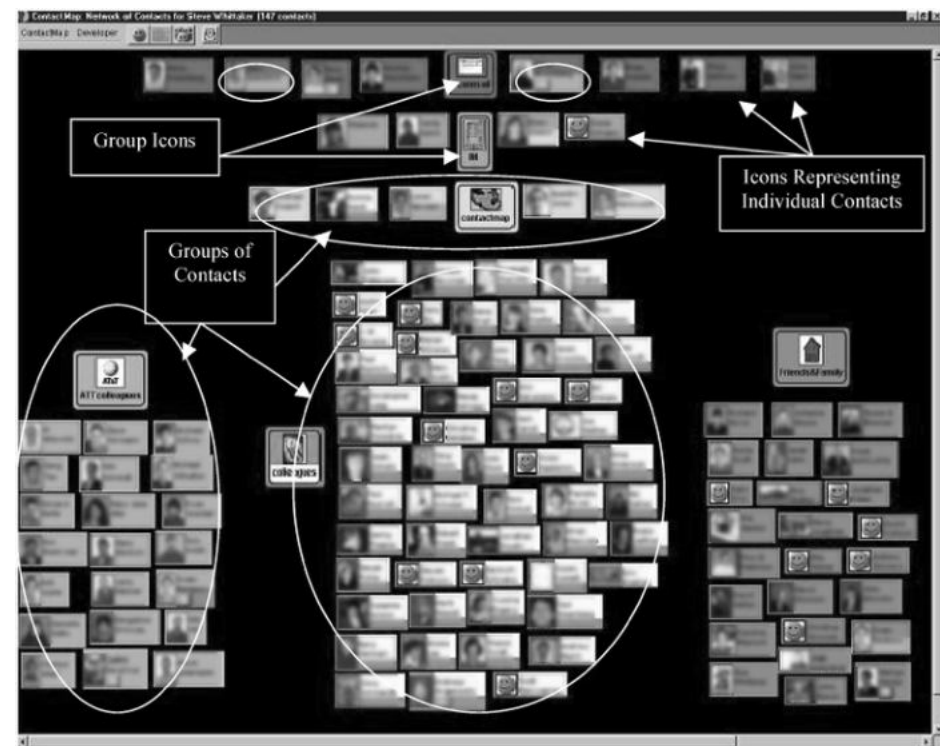
- ContactMap is a great way of externalizing the contact list as a visualization.
- Users can easily see an overview of all the contacts.
- Users' contact groups are easily viewable and it is a great tool for exploration.

Cons

- The ContactMap does not present any actionable information to the user.

"Seeing contacts is really useful for doing things related to projects or tasks, but if I want to see recent messages or just quickly respond to a message, then Netscape is better."

"It's great to see messages related to people, but I still need the time-based Inbox view to give me an overall sense of what's happened in the last hour or two."



CONTACTMAP : GROUPS OF CONTACTS CREATED BY EMAIL PATTERNS

Calchas²⁴

Calchas is a mobile contact retrieval interface which attempts to contextually predict contacts which the user is likely to access. The prediction is done based on recency and longevity patterns. The time span for recency and frequency windows are user defined along with the number of contacts to be predicted. It was found that presentation of order of contacts (random, alphabetically arranged or chronological) does not play a significant role in the usability of the predicted list. There is no strong correlation between the order in which the contacts are arranged and find-ability of the contact.

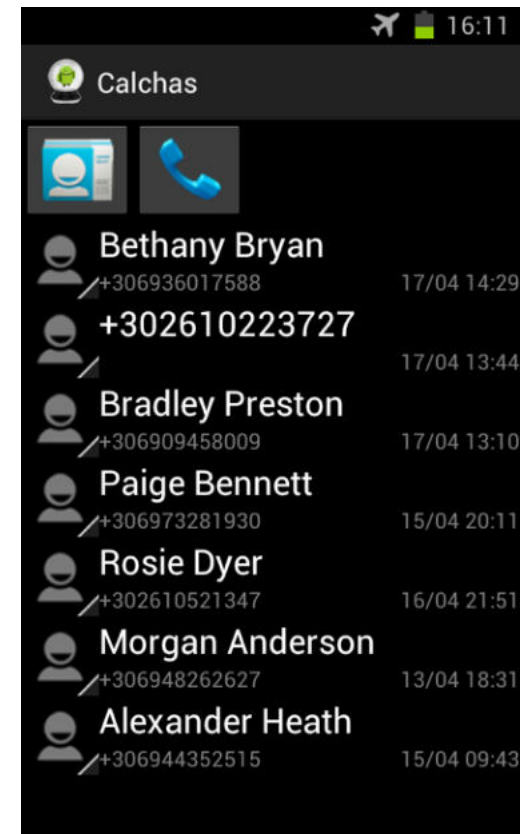
Pros

- Calchas is great step forward in contact suggestion and does a good job suggesting important contacts. This method of suggestion could readily be used with certain improvements in the granularity of the data used.
- The application uses the concept of unremarkable computing²⁵ well, where it does not give the user a reason for the suggestion of a contact. Thereby making the application slightly simpler to use despite additional screens. This principle of unremarkable computing is a good way of indicating the contacts suggested without burdening the user with the cognitive load of reasons for prediction in most cases.

Cons

- The author's have not worked on the user interface for the contact book prediction. There is a haphazard arrangement of contacts.
- It only uses recency and longevity patterns and misses out on the use of some important metadata like user relationships, location, relatedness, etc.

- The recency and longevity time spans are to be defined by the user. This is not the best design decision, since users might not know what is a good time span for accurate predictions.
- The application does not take into account the reduction of the importance of certain contacts overtime.



THE LISTING OF SUGGESTED CONTACTS IS RANDOM. HOWEVER, THIS DOES NOT AFFECT THE USABILITY OF THE APPLICATION.

ContextContacts²⁶

ContextContacts is an application developed in an effort to redesign the contact book. The application is based on presenting information of several meaningful contexts of the user's contacts. The information of profile, current status (busy, available, using phone right now) if indicated up front to help the user in deciding the best medium to be used to contact the person.

Pros

- The design intervention augments the existing phonebook application without creating a separate application.
- The application brings the data forward which could be one of the low priority goals of the phonebook.

Cons

- The application gives rise to a number of security and privacy concerns since it takes sensitive data of the phone status from the users' contacts. The application might be used for spying purposes. The amount of data presented for a contact does not justify its utility.

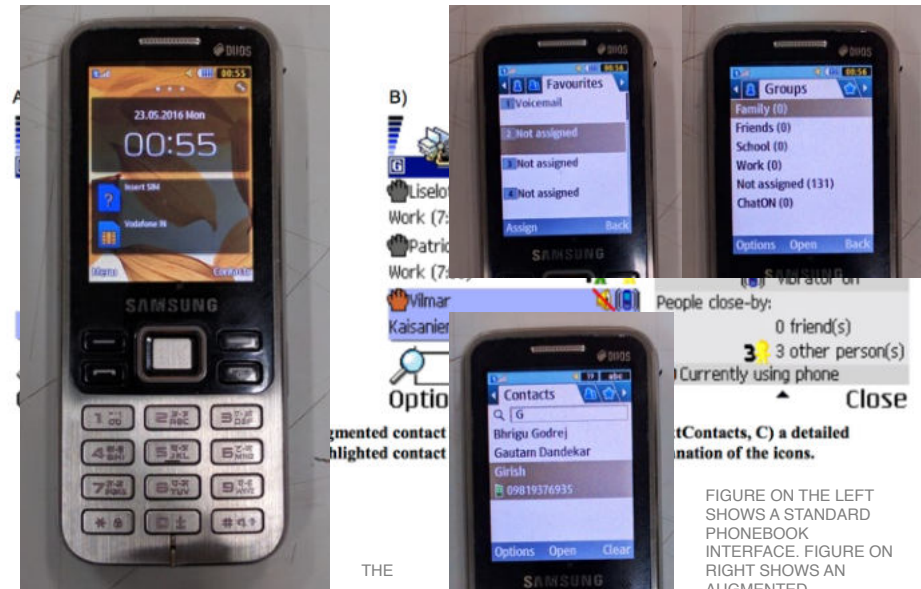
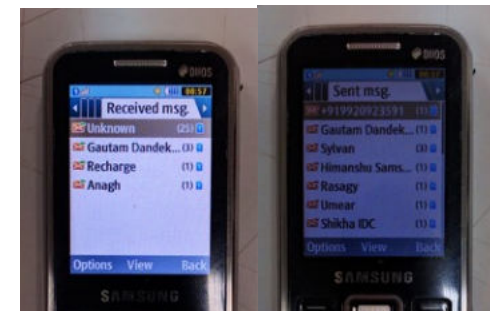
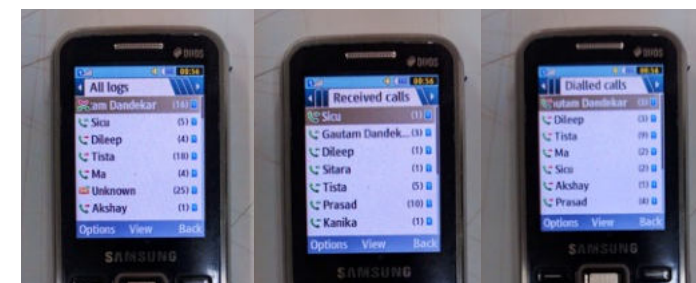


FIGURE ON THE LEFT SHOWS A STANDARD PHONEBOOK INTERFACE. FIGURE ON RIGHT SHOWS AN AUGMENTED



MESSAGES ARE ACCESSED FROM WITHIN THE CONTACT BOOK



CALL LOGS ARE ACCESSED FROM A SEPARATE APPLICATION

Conventional Case Studies

Samsung Duos Phone App

The phonebook application is made for a feature phone.

General Information

Phone Type : Feature Phone

Input Type : Physical Keypad

OS : Tizen OS

Ordinary Features

- Text Input based Search
 - Alphabetical List of contacts
 - Favorites
 - Call Logs (Chronological list)
-

Special Features

- Call logs have are divided into All, missed, outgoing and incoming calls.
 - Call logs are accessed from a separate application and are not embedded in the phonebook application.
-

Cons

- The phonebook for a feature phone is slowly losing relevance.

IOS Phonebook App

The phonebook application is made for devices running IOS.

General Information

Phone Type : Smart Phone

Input Type : Touchscreen Keypad

OS : Apple IOS

Ordinary Features

- Text Input based Search
- Alphabetical List of contacts
- Favorites
- Call Logs (Chronological list of the last 100 calls)
- Keypad

Special Features

- Call logs have are divided into All and missed calls which is a good way to bring important information forward. In this case missed calls.
- Siri Suggestions widget can be accessed from the home screen dialogue. The widget suggests 5 frequent contacts in the default state and 10 frequent contacts in the expanded state. The suggestions widget easily enables access to frequent contacts quickly and is a good design intervention to the existing phonebook.

- The alphabetical list is pleasantly redundant in nature. The list is alphabetically sorted both by contacts' names and surnames which makes is highly usable.
- Users can filter the phonebook for inclusion of emails. This is a handy feature in cases where users want to declutter the contact list.

Cons

- Redundant features like voicemail in the maintab.



REDUNDANCY IN THE CONTACT LIST



SIRI SUGGESTIONS : MINIMIZED SHOWS A FREQUENCY BASED SUGGESTION OF CONTACTS



SIRI SUGGESTIONS : MAXIMIZED

Windows Phonebook

The phonebook application is made for devices running the Windows Phone OS.

General Information

Phone Type : Smart Phone

Input Type : Touchscreen Keypad

OS : Windows Phone

Ordinary Features

- Text Input based Search
- Alphabetical List of contacts
- Favorites
- Call Logs
- Keypad

Special Features

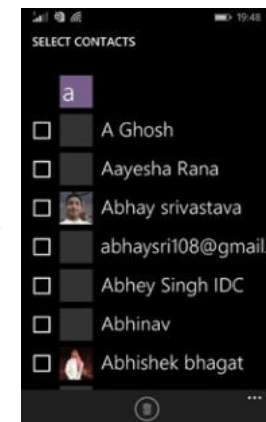
- Grid of alphabets for quick access of contacts and easy navigation. This organization reduces the number of scrolls needed to navigate alphabetical lists. and can be used as a standard phonebook feature.
- The phonebook gives an overview of all the user's social networks. Integration of social networks is also a direction in which phonebooks ought to go in.

Cons

- The Grid of alphabets takes the user to a list, thereby defeating the purpose of having a grid. The same action can easily be performed by means of an alphabet scroll.



GRID OF ALPHABETS FOR



ALPHABETICAL LIST



OVERVIEW OF THE USER'S SOCIAL NETWORKS

ASUS Phonebook

The phonebook application is made for ASUS smartphones running a custom Android OS.

General Information

Phone Type : Smart Phone

Input Type : Touchscreen Keypad

OS : Android (Custom)

Ordinary Features

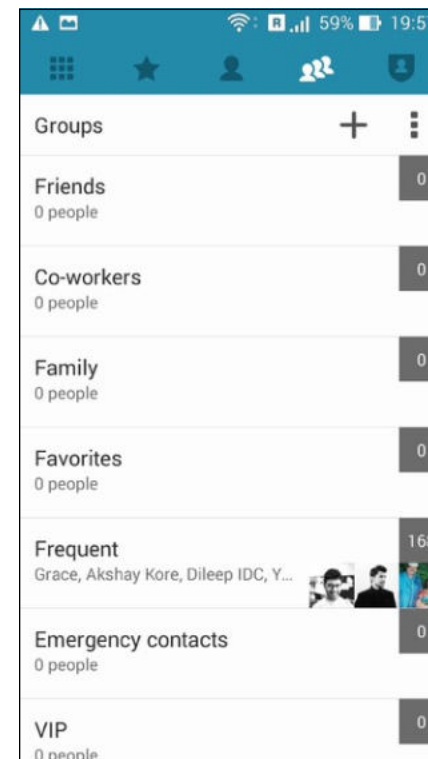
- Text Input based Search
- Alphabetical List of contacts
- Favorites
- Call Logs
- Keypad

Special Features

- Groups is a special tab.

Cons

- Blocked calls in a separate tab is brought upfront. This is an unimportant feature that can easily be hidden deep in the information architecture.
- The groups special tab is not completely pre-populated and the application expects the user to create groups. Most of the groups are empty.
- The groups tab contains a frequent contact group that is pre-populated. This feature could be shallow in the information architecture.



THE PHONEBOOK AUTOMATICALLY
CREATES A GROUP OF FREQUENT
CONTACTS

Samsung Phonebook

The phonebook application is made for Samsung smartphones running Tizen OS.

General Information

Phone Type : Smart Phone

Input Type : Touchscreen Keypad

OS : Tizen OS

Ordinary Features

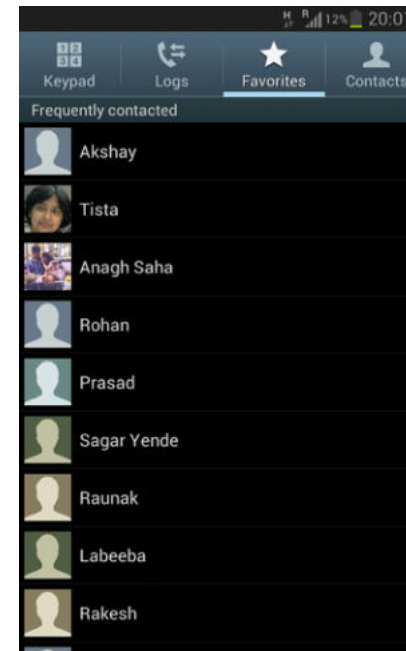
- Text Input based Search
- Alphabetical List of contacts
- Favorites
- Call Logs
- Keypad

Special Features

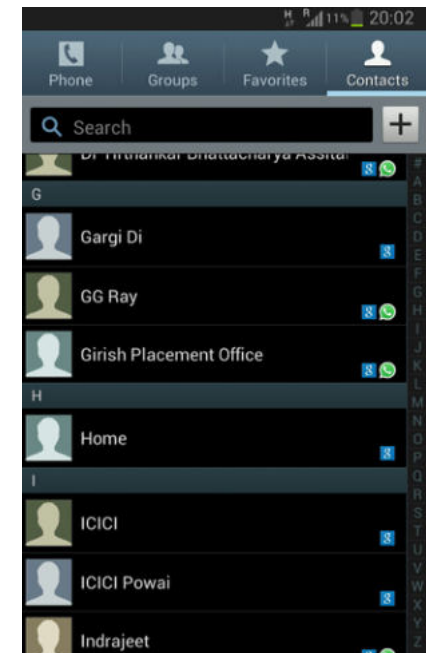
- The contact list element shows upfront what information the contact element contains. This is beneficial to the user in deciding the medium of communication.
- Favorites Tab is pre-filled with frequent contacts and is almost never empty. This feature enables the favorites tab to be contextual without providing additional settings.
- The Phonebook Log shows a list of calls as well as messages.

Cons

- The contacts tab changes the modality of the application without informing the user. The 'keypad' and 'log' tabs are replaced by 'phone' and 'groups' respectively.



PRE-POPULATED FAVORITES



THE TOP TAB CHANGES WHEN THE USER ENTERS THE CONTACTS MODE.

Truecaller

Truecaller²⁷ is a popular cross platform phonebook application. It based on the idea of crowdsourcing contact information and revealing information of unsaved contacts to the user. It is not a default application.

General Information

Phone Type : Smart Phone

Input Type : Touchscreen Keypad

OS : Apple IOS, Android and Windows Phone

Ordinary Features

- Text Input based Search
- Alphabetical List of contacts
- Favorites
- Keypad

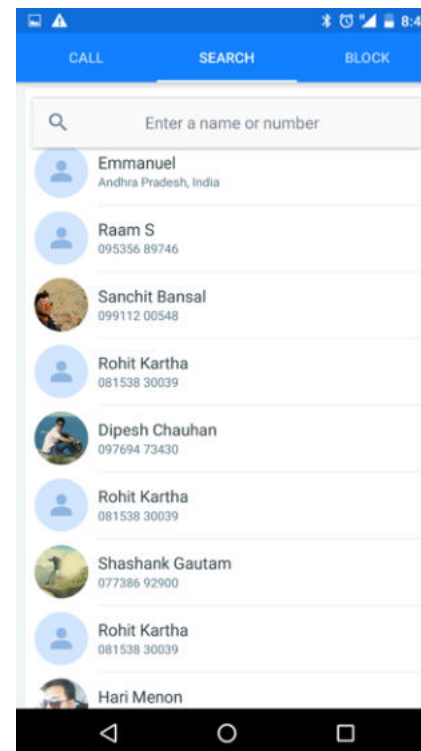
Special Features

- Searching for unknown numbers as a result of an extensive crowdsourced database. Truecaller is a good example using an online database of contacts for showing transparency.
- Warning indication of spam when a call arrives. The crowdsourced data is cleverly used to indicate types of callers.
- It provides a call log with information of numbers that have not been saved.

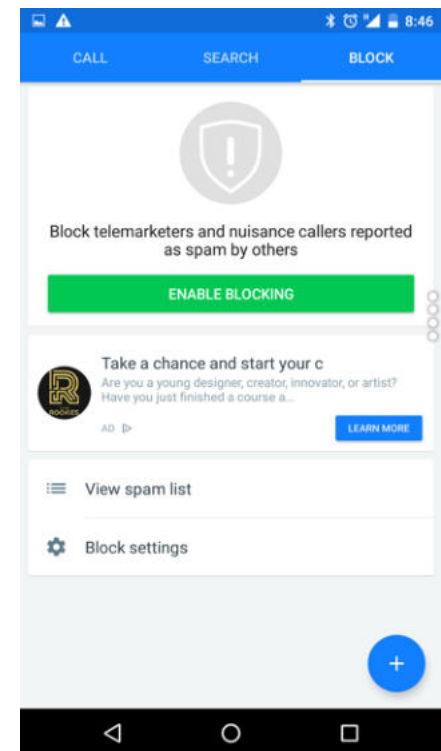
- Busy / Available indicators for contacts. Showing contacts' context is an idea that can be borrowed.

Cons

- Advertisements in call logs makes it more of a utility in case of the free version than a replacement for the phonebook.



CROWDSOURCE SEARCH OF CONTACTS. A CONTACT NUMBER IS ALMOST NEVER DEVOID OF ITS IDENTITY



ADVERTISEMENTS AT DIFFERENT TOUCHPOINTS IN THE APPLICATION

Drupe Phonebook

The Drupe²⁸ phonebook application is made for devices running the Android OS. This is not a default application.

General Information

Phone Type : Smart Phone

Input Type : Touchscreen Keypad

OS : Android OS

Ordinary Features

- Text Input based Search
- Alphabetical List of contacts

Name of Phonebook	Call Log, Alphabetical Lists, Favorites, Text Input Search	Call Frequency based suggestions	Auto groups formation	Other	Cons
Feature Phone (TIzen OS)	Yes				
IOS	Yes	Yes			
Android Lollipop	Yes	Yes			
Asus ZenPhone	Yes		Yes		Most auto created groups are empty
Windows Phone	Yes			Grid List, Social Book encourages exploration	
Samsung Phonebook	Yes				
True caller	Yes			Crowdsourced phone directory	
Drupe Caller	Yes	Yes		Quick access to multiple modes of communication, notification for special days	

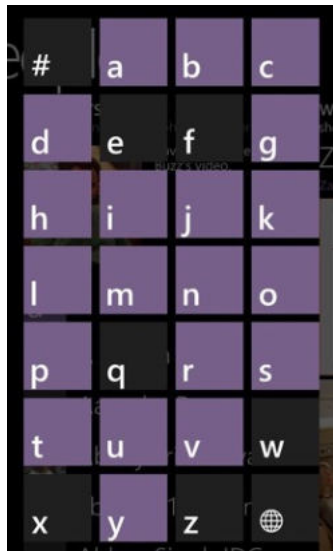
COMPARISON CHART OF CONVENTIONAL CASE STUDIES

- Favorites
- Call Logs
- Keypad

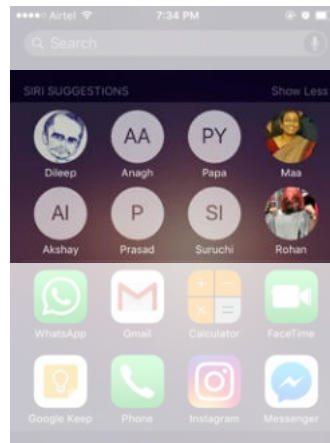
- Customizable background
- Notifications for special events like a contact's birthday, anniversary, etc. These can be used to suggest relevant contacts.

Special Features

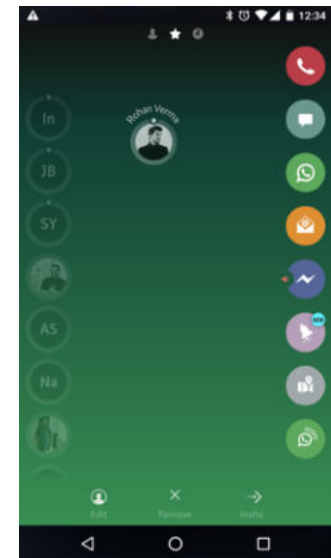
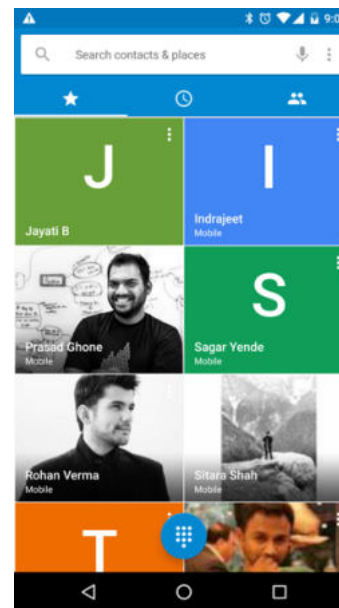
- Quick access to multiple modes of communication available.
- Clever use of long press and swipe gestures within the phonebook application.
- Always available widget on the homescreen.



ALPHABETICAL GRID BASED NAVIGATION IN WINDOWS MOBILE



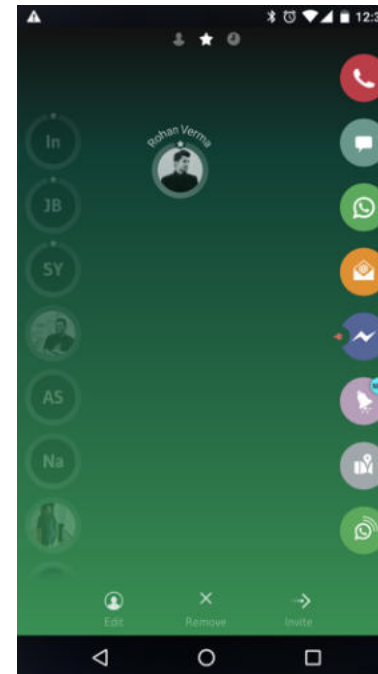
CALL FREQUENCY BASED SUGGESTIONS IN IOS (LEFT) AND ANDROID MARSHMALLOW PHONEBOOK (RIGHT)



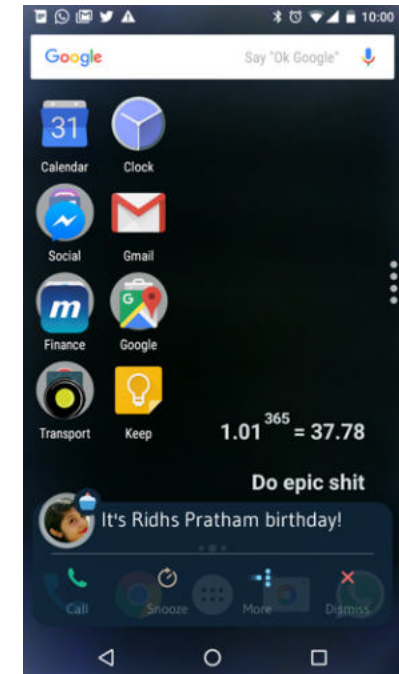
SPECIAL INTERACTION OF SELECTING AN APPROPRIATE COMMUNICATION MEDIUM IN THE DRUPET CALLER

Cons

- Clunky User Interface
- The application shows too much information and lacks smart defaults. The user is expected to know what modes of communication are available for a contact.



USER HAS TO DRAG A CONTACT TO ONE OF THE COMMUNICATION CHANNELS TO USE THE MEDIUM



PUSH NOTIFICATIONS FOR SPECIAL EVENTS. THE CALL TO ACTION IS VISIBLE IRRESPECTIVE OF THE MEDIUM OF COMMUNICATION BEING AVAILABLE.

Overview of Case Studies

A study of different conventional and non-conventional phonebook/contact management applications revealed a number of interesting insights and patterns followed in the design of the phonebook app.

- Some of the applications such as Truecaller and ContextContacts raise a number of privacy issues due to collection of data from sources other than the user of the application itself. Data for other contacts is revealed.
- Most phonebook applications still follow the same pattern of text input based search, alphabetical list of contacts, favorites, call logs, etc. with minor user interface tweaks.
- The favorites feature is one of the most underutilized options in the phonebook. A number of phonebook applications namely the Android Lollipop phonebook and Drupe Caller do a good job of pre-populating favorites.
- Some phonebook applications such as the Android Lollipop and the IOS phonebook provide a frequency based suggestion of contacts. The IOS phonebook provides suggestions outside the bounds of the phonebook app in a widget on the home screen while Google phonebook does it within the phonebook app.
- Only the Drupe phonebook suggests events and provides a call to action for the same. However, it suggests irrespective of whether the medium of communication (phone number) is available.

User Studies

The phonebook is a ubiquitous application used by all users of a mobile phone. It was decided early on, that the problems and solutions to a universal application such as a phonebook would be better solved if there is no particular user persona in the design considerations.

A user study was conducted with a total of 11 users of different age, profession and socio-economic status which included 5 students, 1 teacher, 2 working professionals, 2 office support staff and a company driver. A method of contextual inquiry was used to conduct the study. The goal was to find out pain points or clever techniques used in organizing contacts. These would be later extrapolated into design problems or solutions. The user study revealed a number of insights particularly in the area of contact storage patterns and use of special functions like favorites and groups.

Students	5
Teacher	1
Working Professionals	2
Office Support Staff	2
Company Driver	1
Total	11

Favorites are not a favorite

Of the total number of users interviewed, only 3 saved contacts as favorites out of which one user had accidentally saved a contact as a favorite which he did not bother to remove from his favorites. A glance of contacts who saved favorites revealed that the number of favorite contacts ranged between 1 to 5. A peek at the phonebooks of additional 10 users revealed that none of the 10 users used the favorites feature of the phonebook application.

How do people store contact information?

There were certain repeating patterns found in the contacts storage. These were thought to be not very useful in the beginning. Different users stored the same contact with different names and information. For example the contact of the same person named 'Anirudha Joshi' was stored as 'Prof Anirudha', 'Prof Joshi', 'Anirudha Joshi IDC' etc. by different people. One can however generate groups from the semantic structure of the different contact names. These contact names often contain substantial amount of semantic data which can be used in cases of context awareness.

There were certain peculiarities in contact storage patterns :

- Majority of the users stored contacts of relationships as the relationships themselves (Mom, Dad, Aunt, Uncle).
- Some phone numbers were identified from their locations. In this case often the location was a suffix (eg. Richa London, Sushant Bangalore).
- Close contacts are often saved only as their first names (Rohan, Prasad).

- Some users saved place names such as shop, restaurant names as contacts.
- Many contacts had formed special groups. For example, a user saved a few contacts with a conference name he met the contact in as a suffix (CITD Ram).
- Majority users saved a number of contacts indicating the contact's position in the society. These were characterized by words such as Sir, HOD, Senior, Junior, etc. as prefix or suffixes.
- A number of contacts saved contacts with their professions indicated in the contact name (Dr. Podar, Ar. Shweta).
- Users belonging to a lower socio-economic background and low levels of literacy like the support staff and the company driver used clever ways of storing important contacts. They saved these contacts with a number as a prefix (eg. '1sunita') for the contact to appear the first in the contact list.

The contact patterns found in the user studies are however varied and messy in nature. Some contacts saved their father's or wife's contact numbers as 'boss' or 'love' which does not fall under particular categories. One user with a higher socio-economic background who employs multiple drivers saves or renames the day's driver's contact as 'driver of the day'. Analysis of contact names based of semantic patterns would be a daunting task when in fact a call log analysis would yield much better results with low messiness.

Redefined Design Brief

The availability of good computing capabilities of smartphones and a large amount of data available for contacts enables a great number of possibilities. The phonebook is a ubiquitous application used by all smartphone users irrespective of the operating system differences. In fact the phonebook is one of the de-facto components for any phone. The primary use of the phone still remains to find a contact. However, searching for a contact can be done through multiple ways apart from the standard text input based search, alphabetical lists, call logs and favorites.

Proposed goals for a contact book application

Enable Quick Access to Important Contacts

The primary use of the contact book is to search relevant contacts at the moment. This can be achieved through a phonebook application that suggests contacts based on contextual awareness i.e. using data for call log patterns, location, etc. The end result would be achieved through design interventions in the following areas in the contact book:

- Visualization of Contacts
- Navigation

Encourage Exploration

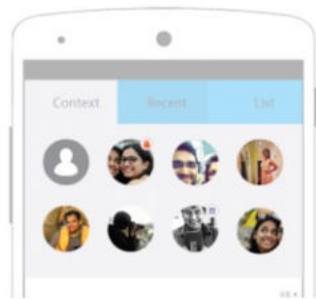
The Phonebook is a quick access application for majority of the smartphone users. However, a variety of metadata associated with contacts may lead of a number of insights about the interactions. Patterns can be revealed by means of data visualizations to enable and encourage deeper explorations of the users' contacts.

Preliminary Design Idea

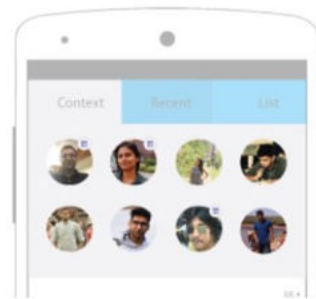
Contextual Prediction Widget

One of the early ideas consisted of creating a widget in the existing standard phonebook structure that predicts a set of contacts based on the context of the user. This idea mainly employs bayesian networks and machine learning as mentioned in the secondary research to predict which contacts are most important at the moment. Different types of contacts would be predicted during the different times of the day. Each predicted contact has a tag (birthday, anniversary, meeting scheduled) to let the user know why a particular contact was predicted.

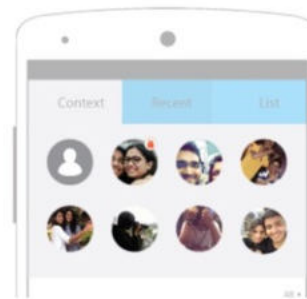
Contextual Prediction Bar



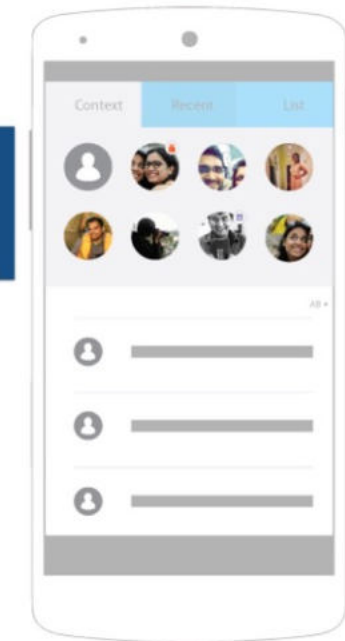
Morning / Home



Afternoon / Office

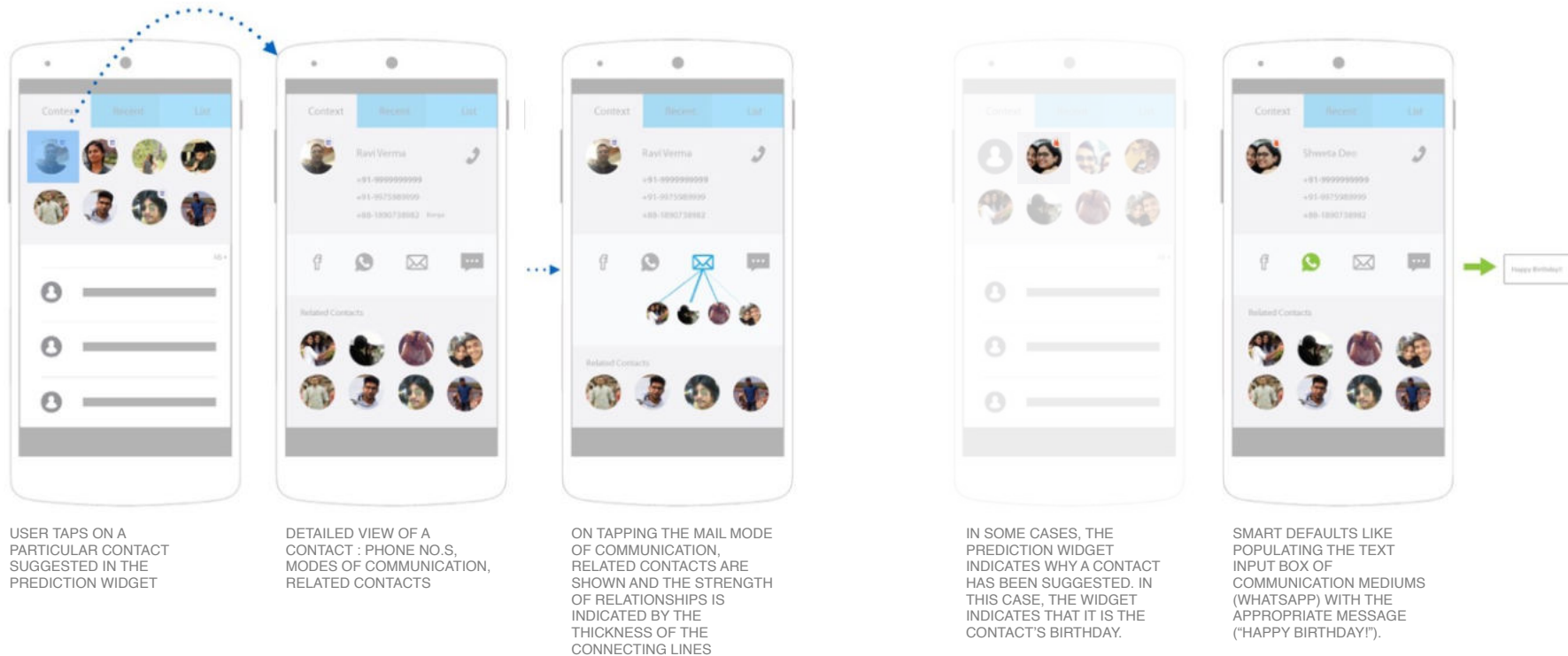


Evening / Home



DIFFERENT TYPES OF CONTACTS PREDICTED DURING DIFFERENT TIMES OF THE DAY

Additional design features include a contact view which shows contacts related to the particular contact and smart defaults (like sending a birthday wish) based on the contact's information.



CONTACT DETAILED VIEW

Heuristic Evaluation

A heuristic evaluation was conducted with a group of 3 interaction design students. The aim was to find any usability faults in the contextual prediction widget. The feedback showed that the idea of prediction although intriguing was not found to be fool proof.

The heuristic evaluation led to the following insights :

Using Real Data

The proposed solution was based on imaginary data and predictions shown acted as placeholders. It was thought that using real or realistic data about a user would generate richer insights to prove the usability of the predictions generated.

Smaller Screen Real Estate use cases

An accurate prediction would improve the speed of accessing relevant contact information. However, the idea becomes more interesting in cases where there is not much screen real estate available or the user cannot reach the system to access contact information i.e. wearable devices, car dashboards, etc.

Navigation of Contacts

Prediction in the proposed solution at the moment is possible only in cases where the information of a contact is stored. What happens in

cases where contact information is not stored and the user only remembers the context(time, day, date, location, etc.) in which the call was made. An advanced navigation of the call history might be an interesting solution.

Redundancy in text input based search

One of the users with a generic name (Akshay, Sagar, Prasad, etc.) expressed a concern that whenever his father wants to call him he has to type the son's name completely since there are a lot of people with the same name. A standard text input based search of the contact book is a good solution for accessing infrequent contacts. However, in cases where a frequent contact needs to be accessed, typing the name of the contact every time starts becoming a redundant activity.

Relationship Exploration

The use of a phonebook application is often a quick activity involving a few taps on the smartphone. Users mainly use the phonebook to make a call, for messaging or to share contact information. With so much data about a contact being available, the phonebook can also be used as means to explore relationships of the users with their contacts through insights about patterns of calling and messaging.

Design Ideas

The contextual prediction widget idea presented earlier is based on secondary research and the author's judgement. However, the heuristic evaluation revealed a need for using realistic data. A data for a period of three months (from December 2015 to February 2016) was captured from the author's call and email log.

Following data was collected :

Email

- Incoming Messages
- Outgoing Messages
- Contacts in CC
- Day
- Date
- Time

Calls

- Incoming Calls
- Outgoing Calls
- Day
- Date
- Time

An analysis of the call and email log revealed patterns in the data that led to generation of ideas and solutions. The ideas tackle different issues of visualization, overview and navigation problems

The design ideas are categorized into two groups :

- Views
- Navigation

Design Ideas - Views

These ideas mainly involve design of microinteractions that aid in access of contact information.

Histogram View

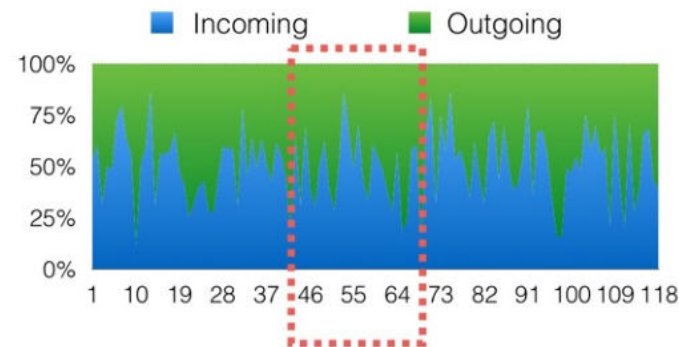
The distribution of emails over a period of 3 months is plotted on a histogram. The histogram showed data for incoming and outgoing calls for each day. Taking cue from the MIT Media Lab's Immersion project, groups of interactions were created using data for incoming messages, outgoing messages and people in cc. These groups were re-named based on the event (IndiaHCI Conference) or type of group.

However, this type of a histogram view of the email interactions although intriguing does not help in solving the problem of quick access to contacts.

Indrajeet	incoming call with	2016-01-18	01:48:23	00:00:30
Prasad Ghone	incoming call with	2016-01-17	21:45:16	00:00:27
Prasad Ghone	outgoing call with	2016-01-17	21:23:03	00:01:26
Aaji	outgoing call with	2016-01-17	21:22:55	00:00:05
Gargi Karve New	incoming call with	2016-01-17	20:48:40	00:00:07
Gargi Karve New	incoming call with	2016-01-17	20:23:42	00:00:01
Gargi Karve New	outgoing call with	2016-01-17	19:18:05	00:00:00
Gargi Karve New	incoming call with	2016-01-17	18:34:15	00:00:03
Gargi Karve New	outgoing call with	2016-01-17	18:29:51	00:00:00
Pooja Raut	incoming call with	2016-01-17	17:34:40	00:00:12
Pooja Raut	incoming call with	2016-01-17	17:29:01	00:02:43
Pooja Raut	incoming call with	2016-01-17	17:20:40	00:00:49
Gargi Karve New	outgoing call with	2016-01-17	16:29:56	00:01:01
Mummy	outgoing call with	2016-01-17	14:40:49	00:00:00
Mummy	outgoing call with	2016-01-17	14:40:28	00:00:00
Gargi Karve New	outgoing call with	2016-01-17	13:52:33	00:00:28
Ruchita Chandsarkar	outgoing call with	2016-01-17	13:52:25	00:00:00
Ruchita Chandsarkar	incoming call with	2016-01-17	13:49:12	00:00:25
Gargi Karve New	incoming call with	2016-01-17	13:47:37	00:00:18
Aditi Shirodkar	outgoing call with	2016-01-17	13:40:34	00:01:03
Gargi Karve New	outgoing call with	2016-01-17	13:39:39	00:00:00
Gargi Karve New	incoming call with	2016-01-17	13:32:55	00:00:41
Ruchita Chandsarkar	incoming call with	2016-01-17	13:06:15	00:00:28

EMAIL LOG EXTRACTED AS A CSV FILE

Contact Histogram - Incoming vs Outgoing



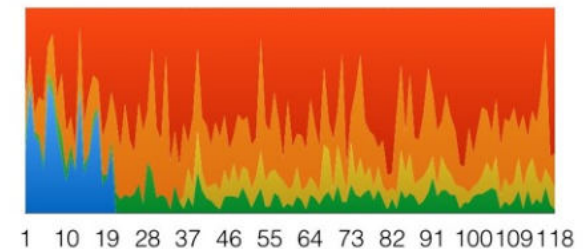
Contact List:

Aj Sir	01/02/16	11:00
Prasad Ghone	01/02/16	11:00
Indrajeet Roy	01/02/16	11:00
Atul Shukla	01/02/16	11:00
Jayati	01/02/16	11:00

A HISTOGRAM VIEW OF THE EMAIL LOG DATA OVER A PERIOD OF 3 MONTHS, OUTGOING AND INCOMING MESSAGES ARE SEPARATED.

Groups

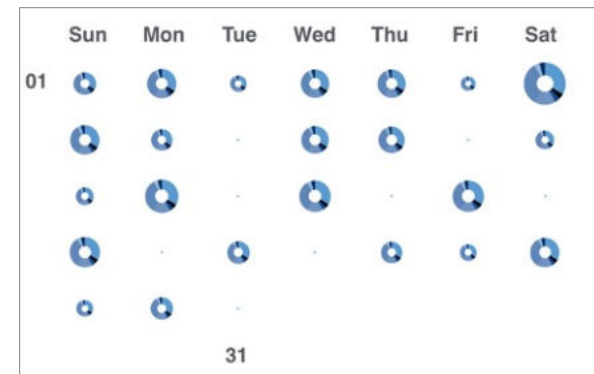
■ HCI ■ P3 ■ CSMVS
■ Placement Work ■ Other



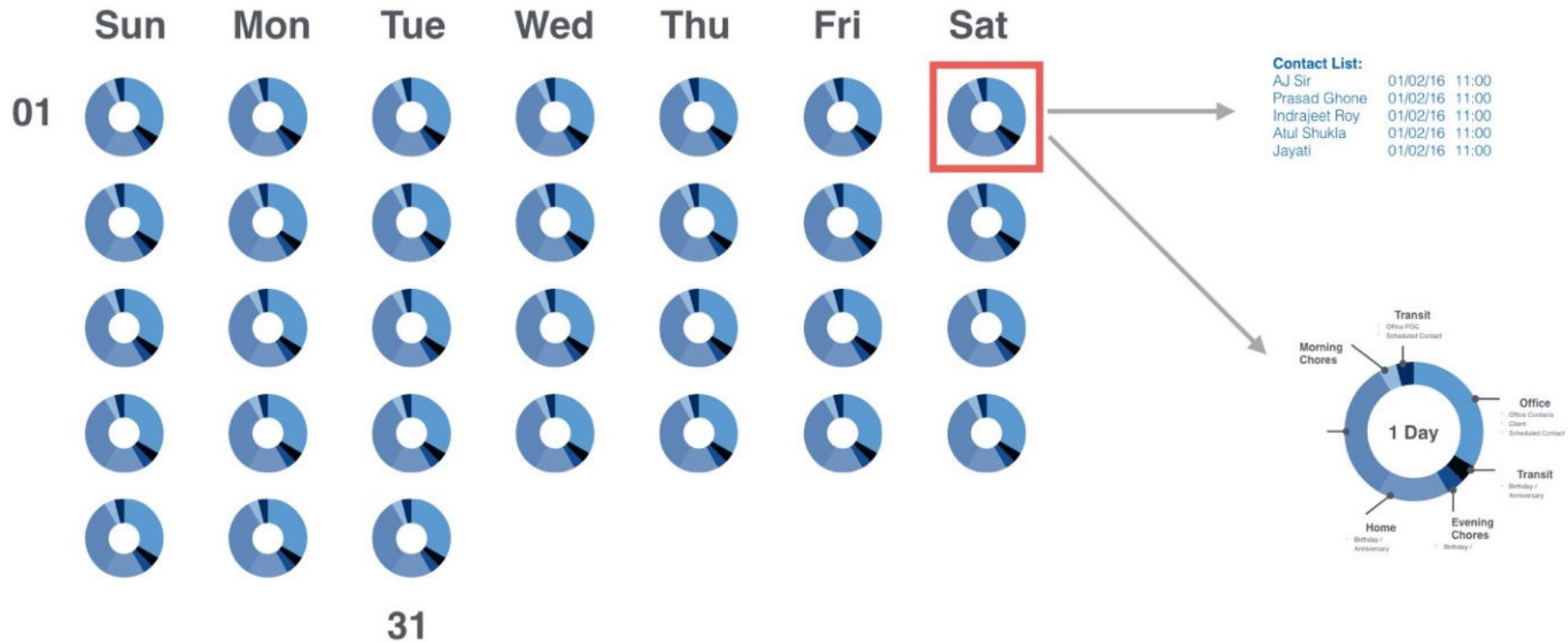
GROUPS ARE GENERATED BASED ON INTERACTIONS BETWEEN CONTACTS AND THE PEOPLE IN CC. GROUPS ARE RENAMED MANUALLY AND REVEALED PATTERNS. EACH GROUP IS INDICATIVE OF PARTICULAR EVENTS DURING THE TIME SPAN.

Calendar View

Taking cues from visualizations generated by fitness applications (heart rate, sleep patterns, etc). The 24hr prediction concept mentioned earlier was expanded into a calendar view. Navigation of contacts would be done through this month long / year long view of interactions. This also was an interesting visualization idea and parts of it were thought to be useful. However, the subsequent visualization is not actionable in nature.



VISUALIZATION OF A CONTACT'S CALLING PATTERNS FOR A MONTH. THE SIZE OF THE CIRCLE INDICATES THE NUMBER OF CALLS. DOT SHOWS THAT NO CALLS HAVE BEEN MADE OR RECEIVED FROM THE CONTACT.



VIEW OF AN INDIVIDUAL DAY. GROUPS FORMED BASED ON CONTEXTUAL AWARENESS. SCHEMATIC

CALENDAR VIEW OF CALL INTERACTIONS. EACH PIE CHART IS INDICATIVE OF A DAY'S CALL LOG. DIVISIONS ARE INDICATIVE OF TIMES THE USER CALLED.

Design Ideas - Navigation

These ideas mainly tackle problems of navigation and visualization of the phonebook utility beyond alphabetical and chronological lists.

Finding contacts based on past experience

How many times does it happen that you are trying to find a contact you saved (or worse, you did not save it!) but you forget the name. You probably know other information like when had you called the contact, where you were when you called, who else did you call. In other words, you know the context of the call. This idea proposes an additional navigation of the phonebook by means of advanced filters. This idea can be explained better by means of scenarios. This method proposes to find a particular contact through the recollection of context or past experience.

The method uses the following parameters for context recollection :

- Day
- Date or Date Span
- Time Span
- Location
- People

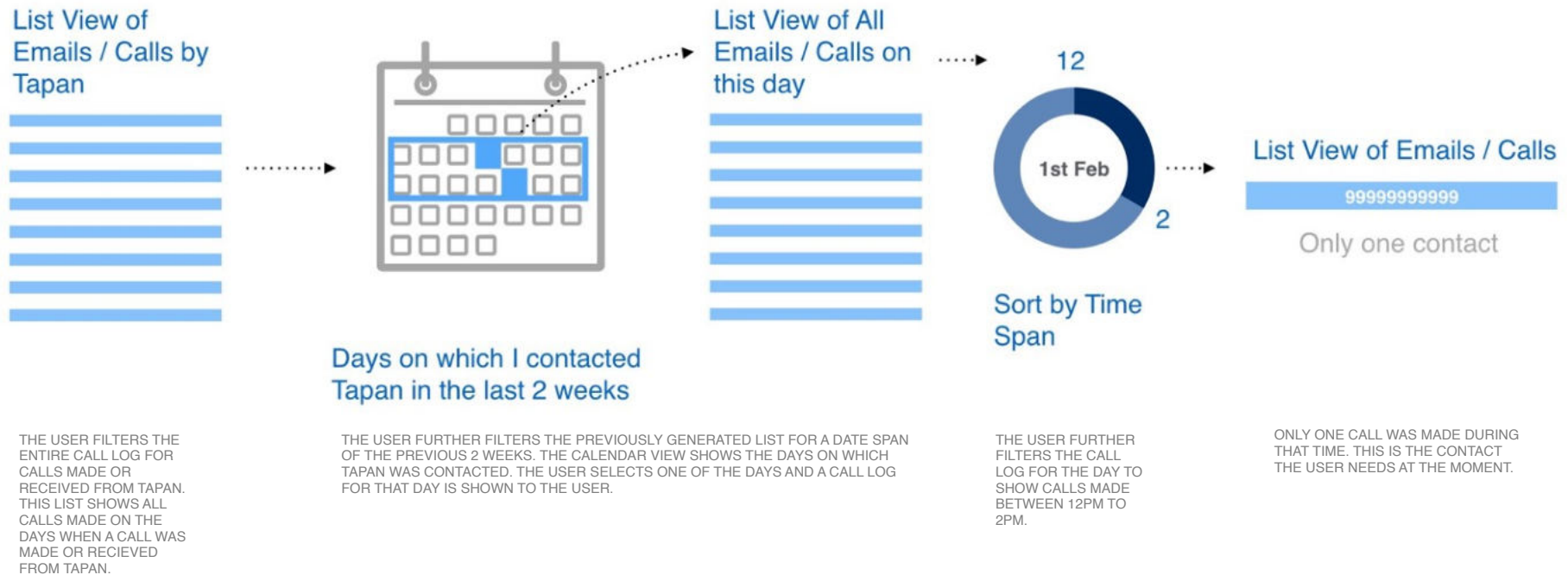


SCENARIO 2 : SEARCH BASED ON DAY AND TIME INFORMATION

Scenario 3: Search based on People/s, Date Range, Day, Time

Tapan had called me to give his clothes to the Dhobi between 4-5 am some 2 weeks back. I called the Dhobi between 12-2 pm on the same day. I do not have the Dhobi's number and urgently need it.

Solution : In this case, the user would filter the days on which Tapan had called him. If there are more number of such days, then the user would filter further for a time span between 4am to 5am. This would for example, lead to only two days that have such a pattern. The user would look at the log of these two days and filter further for time spans between 12pm to 2pm. This advanced result would lead to only one number matching the filters used.

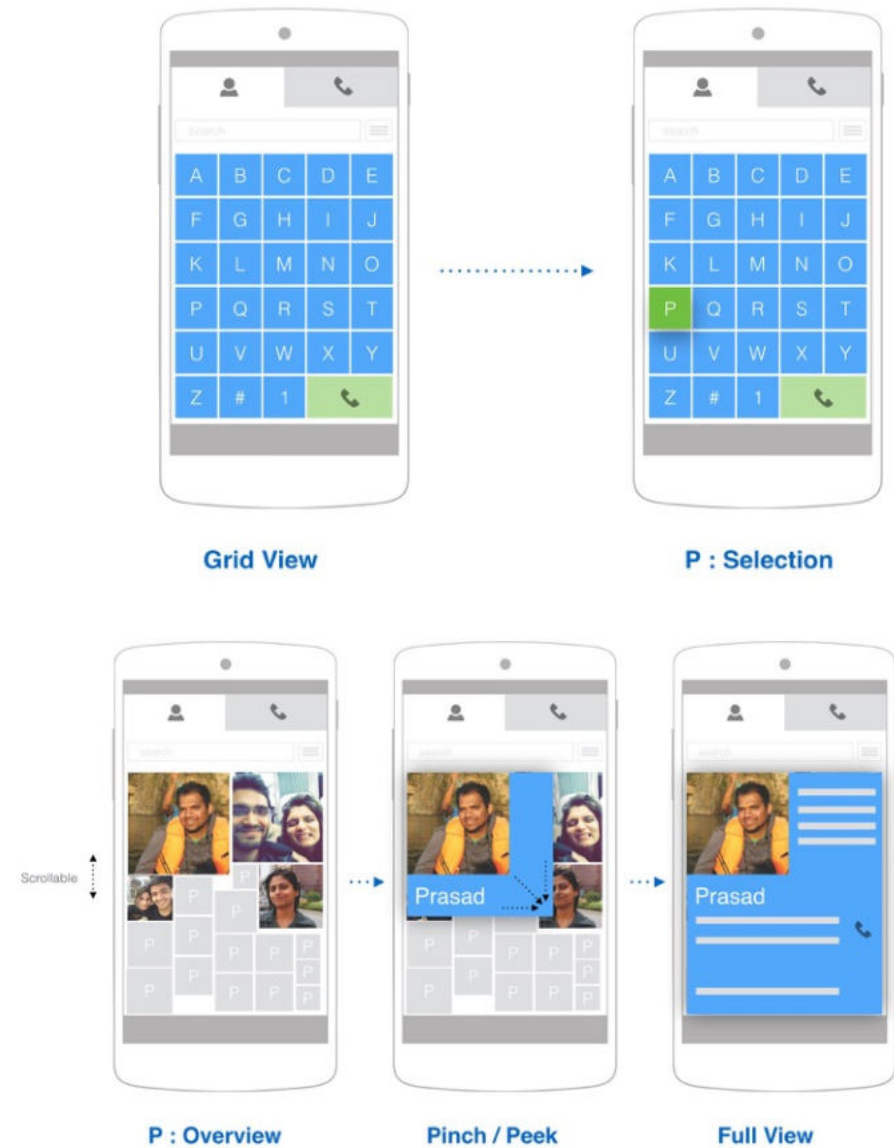


SCENARIO 3 : SEARCH BASED ON PEOPLE, DATE RANGE, TIME RANGE

ContactGrid

One of the users with a generic name (Akshay, Sagar, Prasad etc.) said that whenever his father wants to call him he has to type the son's name completely since there are a lot of people with the same name. This process is unnecessarily redundant since his father is not tech-savvy and has not saved any favorites. The following idea germinates from this problem.

The ContactGrid as the name suggests consists of a simple grid of alphabets for navigation. On selecting a particular alphabet, the screen shows a grid of people with names starting with the alphabet (this is given preference) or containing the alphabet. The size of each element in the grid is determined by the importance of the contact based of frequency, recency and longevity patterns discussed earlier. In the above mentioned scenario, the user's parent would in most cases see the son's contact as a prominent contact thereby saving the number of redundant clicks in search and typing the name.



GRID OVERVIEW BASED ON
CALLING FREQUENCY.

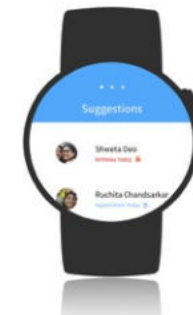
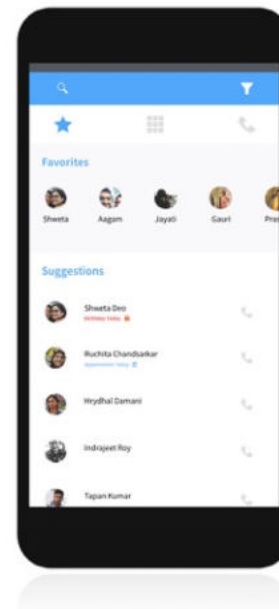
Final Design Idea - PhonBk

A critical analysis of the secondary research, user studies, preliminary design ideas, heuristic evaluation and design ideas reveals a number of pieces of ideas and microinteractions of the final idea taking shape.

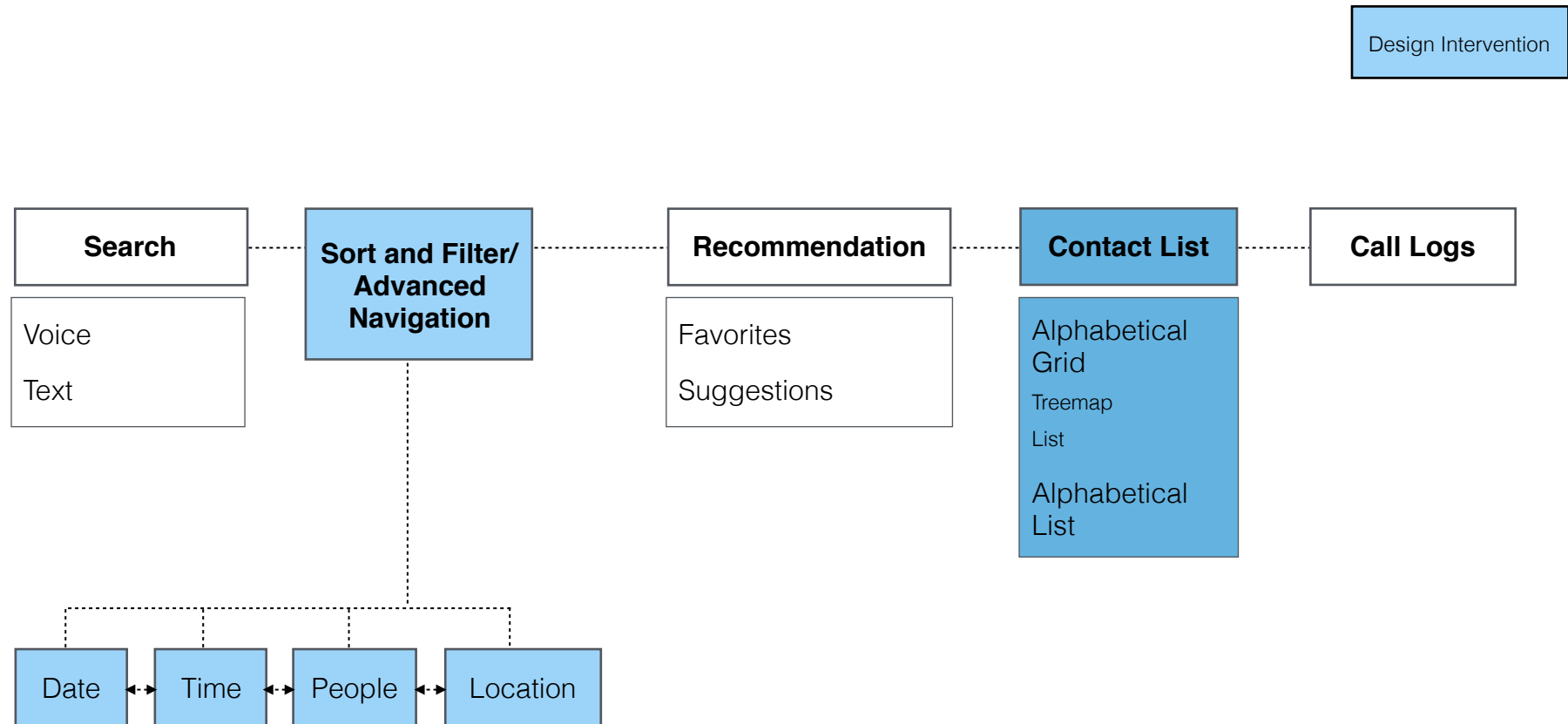
The key findings and ideas that led to the final design are summarized below :

- Suggesting contacts based on the context of the user. Derived from secondary research and the idea of the contextual prediction widget.
- In most cases the user does not need to know why a particular contact has been suggested/predicted.
- Contextual prediction becomes very useful in cases where the screen real estate available is very small or the user is unable to access the device physically eg. wearables, car dashboards.
- User defined favorites is not a very widely used feature. This would decide the hierarchy of favorites utility.
- The users could be given insights about their contacts. These could be hidden deeper in the information architecture.
- Giving the users an alternative to standard navigation routes (list of call logs, alphabetical lists) would be a good idea. This could be done by means of advanced filtering and sorting.
- There is a substantial redundancy in text input based search which could be augmented through smarter visualizations of contacts as in the case of ContactGrid.
- All ideas should augment the existing structure of the phonebook with call logs and alphabetical lists without replacing them. However, one could look at the gradual replacement of these features with the advancement of the proposed feature.

PhonBk is a cross device phonebook application. The application consists of the standard favorites, alphabetical list view and call log features along with design interventions in the form of an advanced sort and filter function, grid based navigation and recommendations based on contextual awareness.



Information Architecture



INFORMATION ARCHITECTURE FOR SMARTPHONES



INFORMATION ARCHITECTURE FOR WEARABLES

Design Details

Fixed Elements

Search

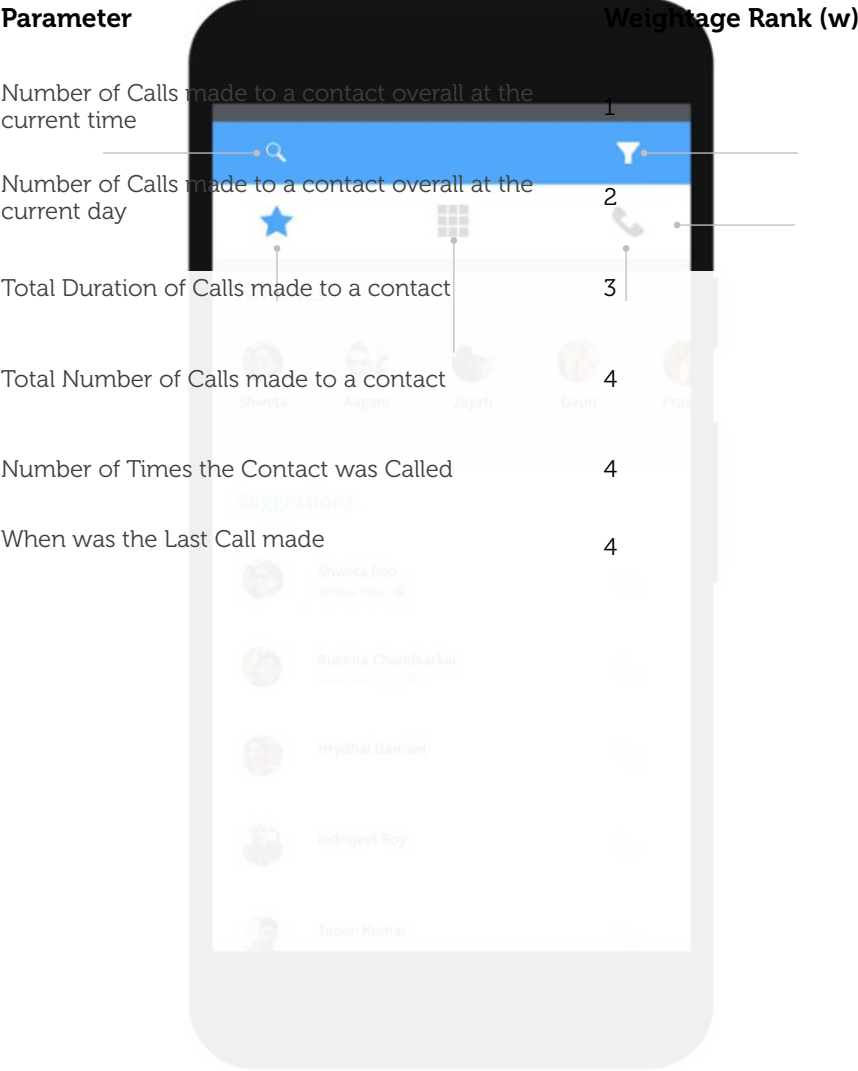
- Voice
- Text

Sort + Filter (sort by)

- Date
- Time
- People
- Location

Navbar

- Recommendation
- Contact List
- Call Logs



Recommendation Panel

The recommendation screen is the first encounter with the application. It consists of the following elements :

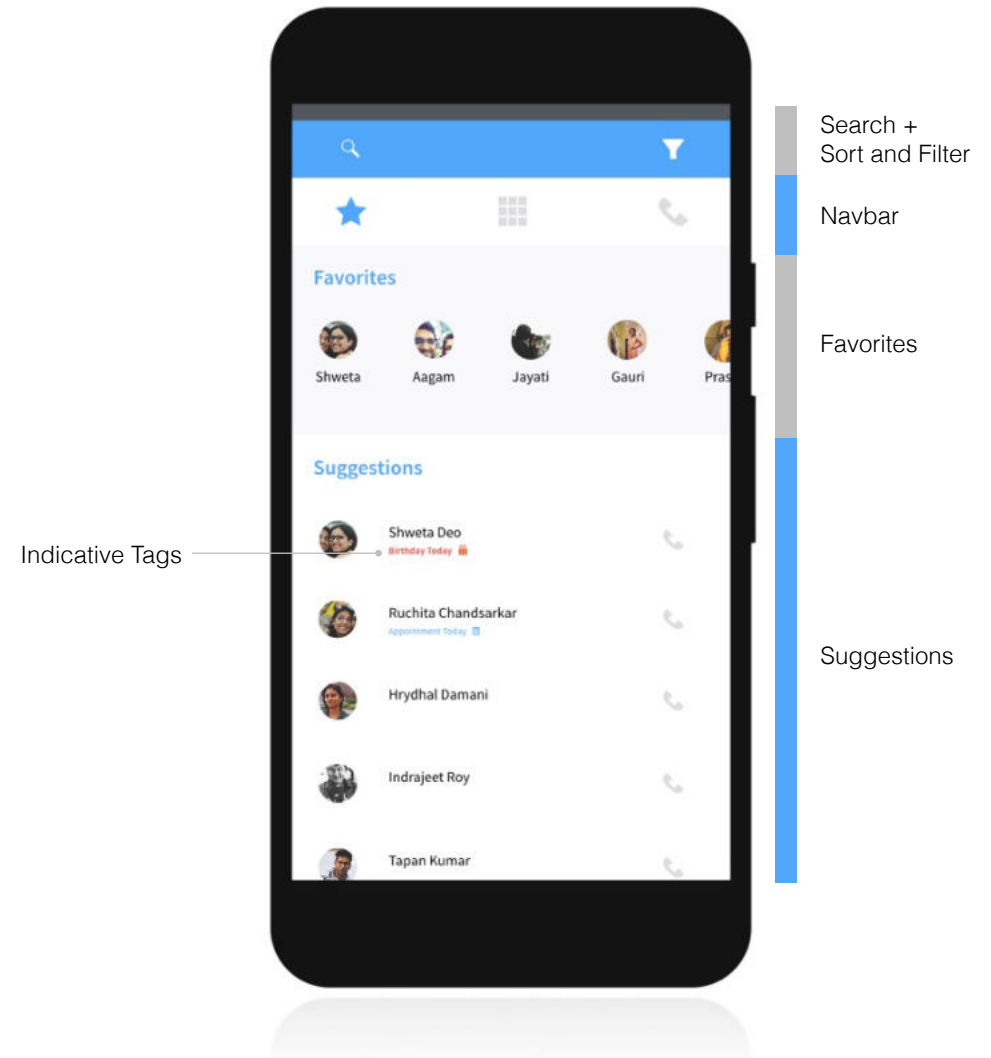
Favorites

It was found in the user studies that favorites is not a frequently used function. Users on an average saved 0-5 contacts as favorites. Most users did not make use of the favorites functionality. The favorites tab is thus designed to occupy a smaller screen real estate.

Suggestions

The suggestions panel occupies 2/3rd's of the screen space. The suggestions panel consists of 10 contact elements. The elements show indicative tags for why a particular contact has been suggested (indicative tags for birthday's, anniversaries, meetings, etc).

Favorites and Suggestions are one of the three major components of PhonBk app for wearables apart from search.



PhonBk Suggestions Algorithm

The suggestions in the suggestion panel are generated by a method of assigning an index value in a database to each contact. The top 10 contacts with the highest index values are shown. The index value is generated by taking into consideration the following parameters :

- Number of Calls made to a contact overall at the current time
- Number of Calls made to a contact overall at the current day
- Total Duration of Calls made to a contact
- Total Number of Calls made to a contact
- Number of Times the Contact was Called
- When was the Last Call made

Each parameter is assigned a weightage and the following equation is used to generate the index value.

$$I = (w1 * (1 - (ThisMomentTotalCallsRank / \maxRank))) + (w2 * (1 - (ThisDayTotalCallsRank / \maxRank))) + (w4 * (1 - (TimesContactedRank / \maxRank))) + (w2 * (1 - (NumberCallsMadeRank / \maxRank))) + (w3 * (1 - (TotalCallDurationRank / \maxRank))) + (w4 * (1 - (LastTimeContactedFromToday / SixMonths)))$$

Where w1, w2, w3, w4 is the weightage rank. The values of which are indicated in the adjoining table.

This method of generating index values is different from the one used by Jun-Ki et al where user schedules, location, statistical patterns, anniversary data were used to create a recommender system based on a context tree²⁹. In the case of the 'PhonBk' Suggestions, a simple indexing and ranking based on the parameters and weightage provided is employed. The 'PhonBk' method is somewhat similar to the PageRank³⁰ Algorithm used by Google Search to rank website urls.

Contact List

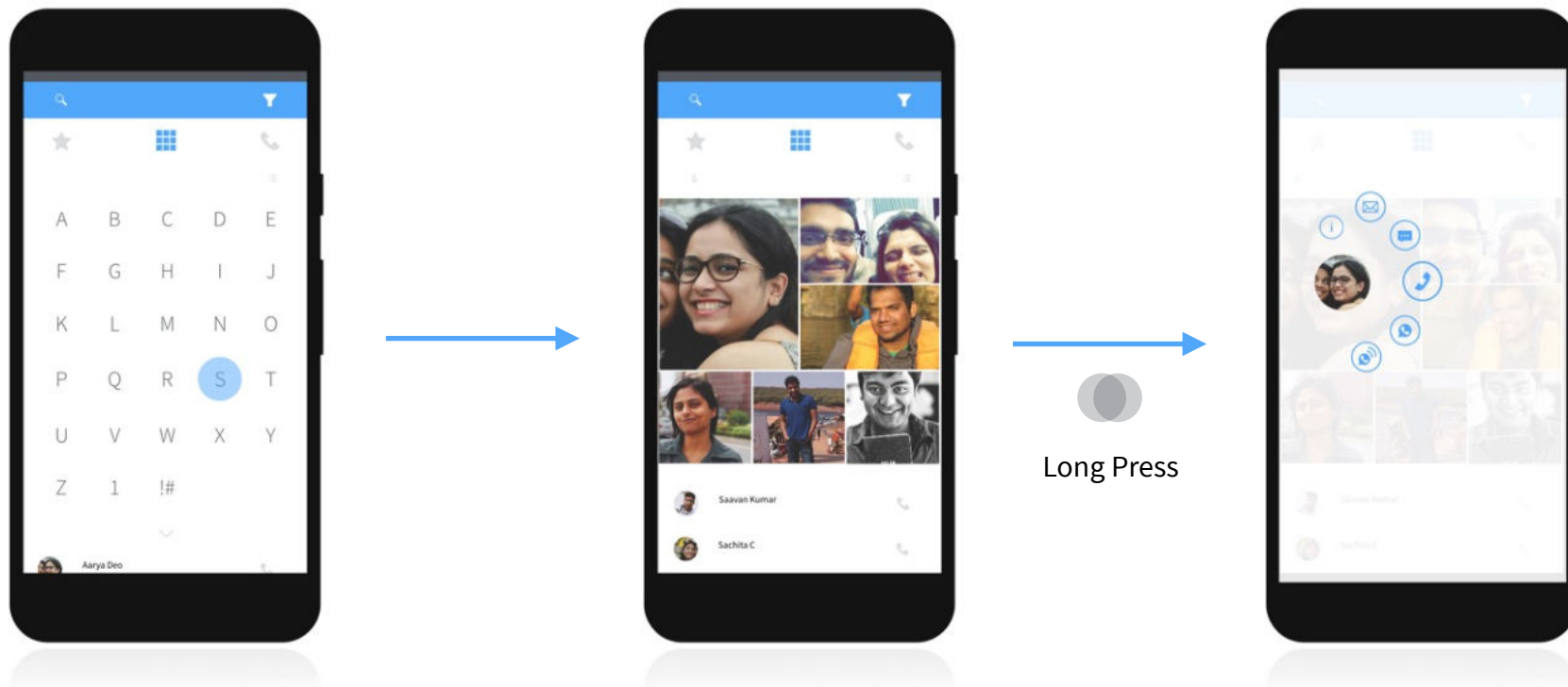
The contact list consists of two modes, mainly the ContactGrid Mode and the Standard List mode. The Standard List is an alphabetically arranged list of contacts.

Contact Grid

This is the default mode where the interaction begins with a grid of alphabets. When the user taps on an alphabet a schematic treemap

view of the top 10 contacts for quick access starting with the selected letter is shown. The location and size of the treemap grid remains constant despite changes in the calling frequency. This is to ensure that the schematic grid is imbibed in the user's muscle memory. A constant change in size and position of top contacts might lead to ambiguity and confusion.

Scrolling below reveals an alphabetically arranged list.



THE USER ON TAPPING A LETTER IS TAKEN TO A PARTIAL TREEMAP VIEW OF THE TOP 10 MOST FREQUENT CONTACTS WITH NAMES STARTING FROM THE LETTER SELECTED. THIS GESTURE REDUCES THE NUMBER OF TAPS NEEDED TO ACCESS A FREQUENT CONTACT TO TWO OVER THE REDUNDANT TASK OF SEARCHING WITH A TEXT INPUT SEARCH.

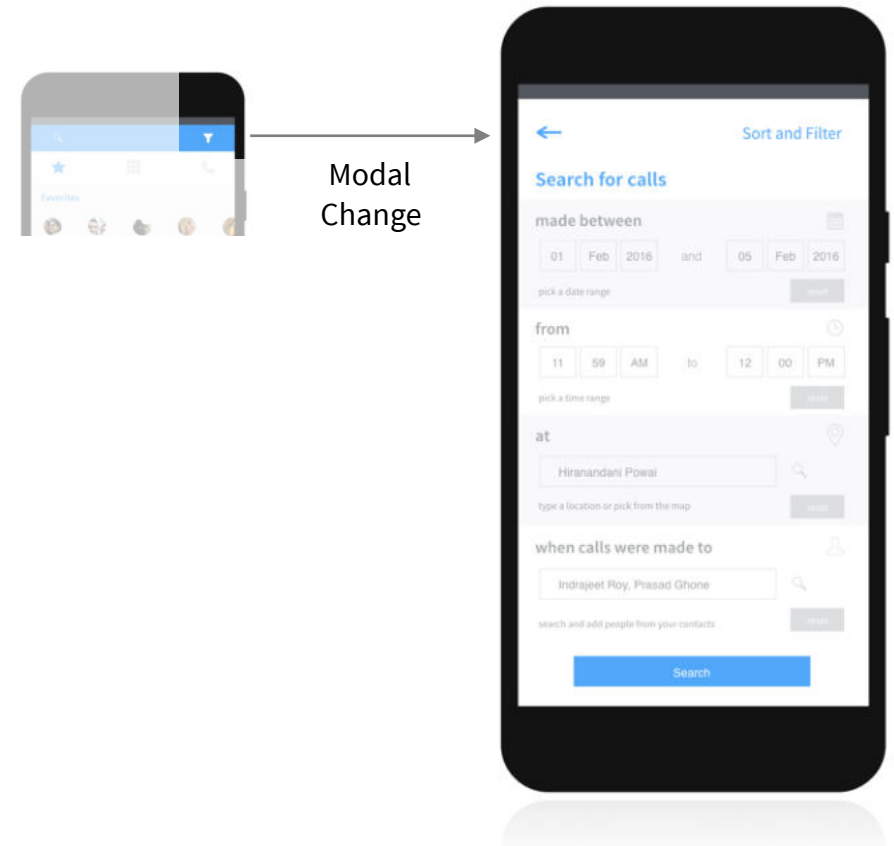
Advanced Navigation

The advanced navigation feature can be accessed from the filter button in the top right corner of the application. The terms 'advanced navigation' and 'sort and filter' have been used interchangeably. Use of the advanced navigation feature is presumed to be an infrequent action and hence brings about a modal change³¹ in the user interface. The top bar is lost and is replaced by a back button.

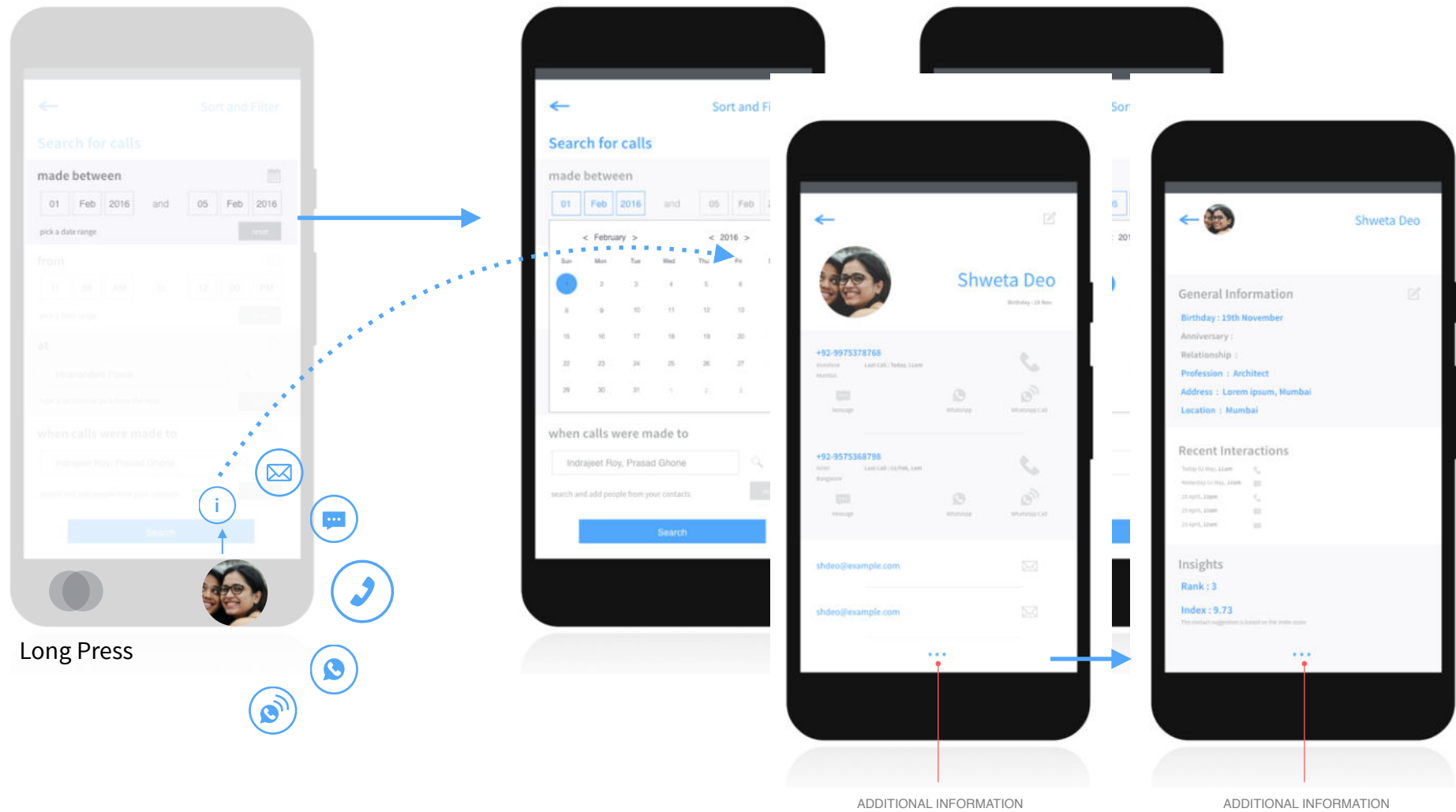
Microcopy

The user's first interaction on reaching the 'sort and filter' mode is with an input form with choices. Taking cues from the Budge app's screen for 'to do' reminders³² and to make the form interesting and easily understandable, the form is broken down into a simple sentence in english i.e. Search for calls made between (date) and (date) from (time) and (time) at (location) when calls were made to (person/people from the contact list). The form also brings forward the underlying logic of the algorithm used for the advanced navigation feature.

Search for calls made between
(this date) and (this date) from
(this time) and (this time) at
(this location) when calls were
made to (this person or
people).



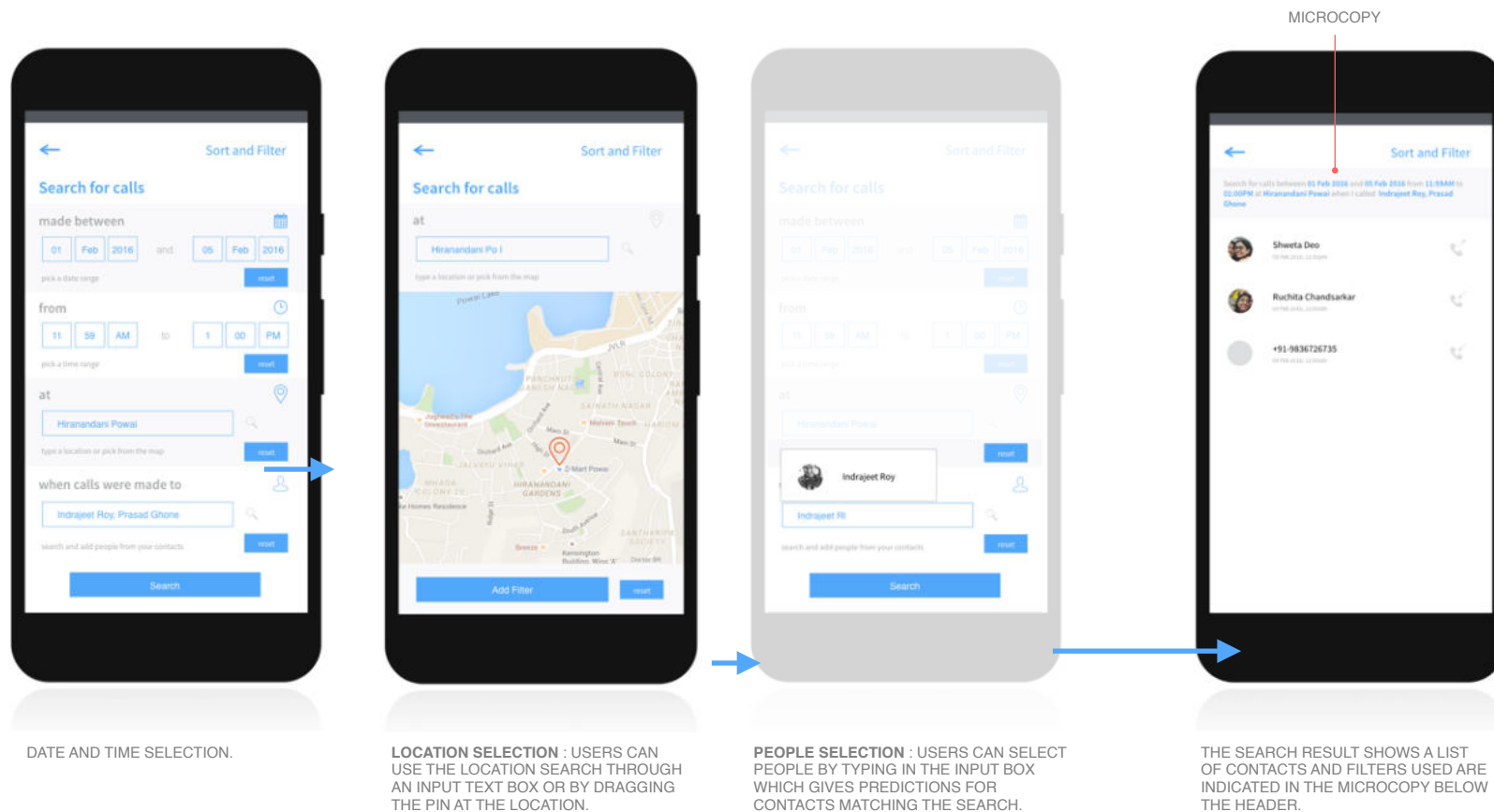
THE SORT AND FILTER IS SHOWN AS A MICROCOPY IN THE FORM OF A SENTENCE IN PLAIN ENGLISH.



Scenario

“I had made a call to Rakesh in the first week of February early in the afternoon. Rakesh is an important lead who works at Microsoft at a senior level. I need to get in touch with an engineering team member for a project I am working on. Sadly, I did not save Rakesh’s number when he called me. I was busy having lunch with Prasad and Indrajeet at Hiranandani Gardens, Powai.”

Solution : In my call log of 1265 calls, I use the PhonBk app’s advanced navigation feature to search for calls made between 12pm to 1pm in the first week of February which yields 136 calls. I further filter the search based on the location from which the call was made i.e. Hiranandani Gardens, Powai which filters the list to 12 calls and the people whom I might have called that day i.e. Indrajeet and Prasad. This gives me a list of 3 calls, of which the two contacts have been previously saved and the unsaved phone number is probably the one I am looking for.



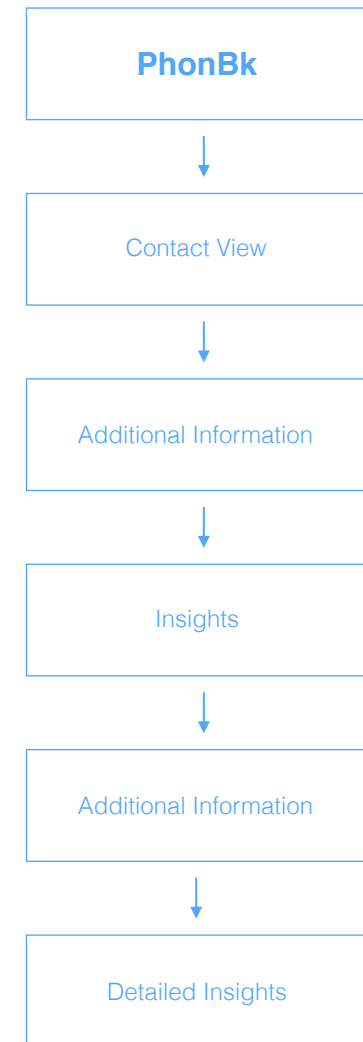
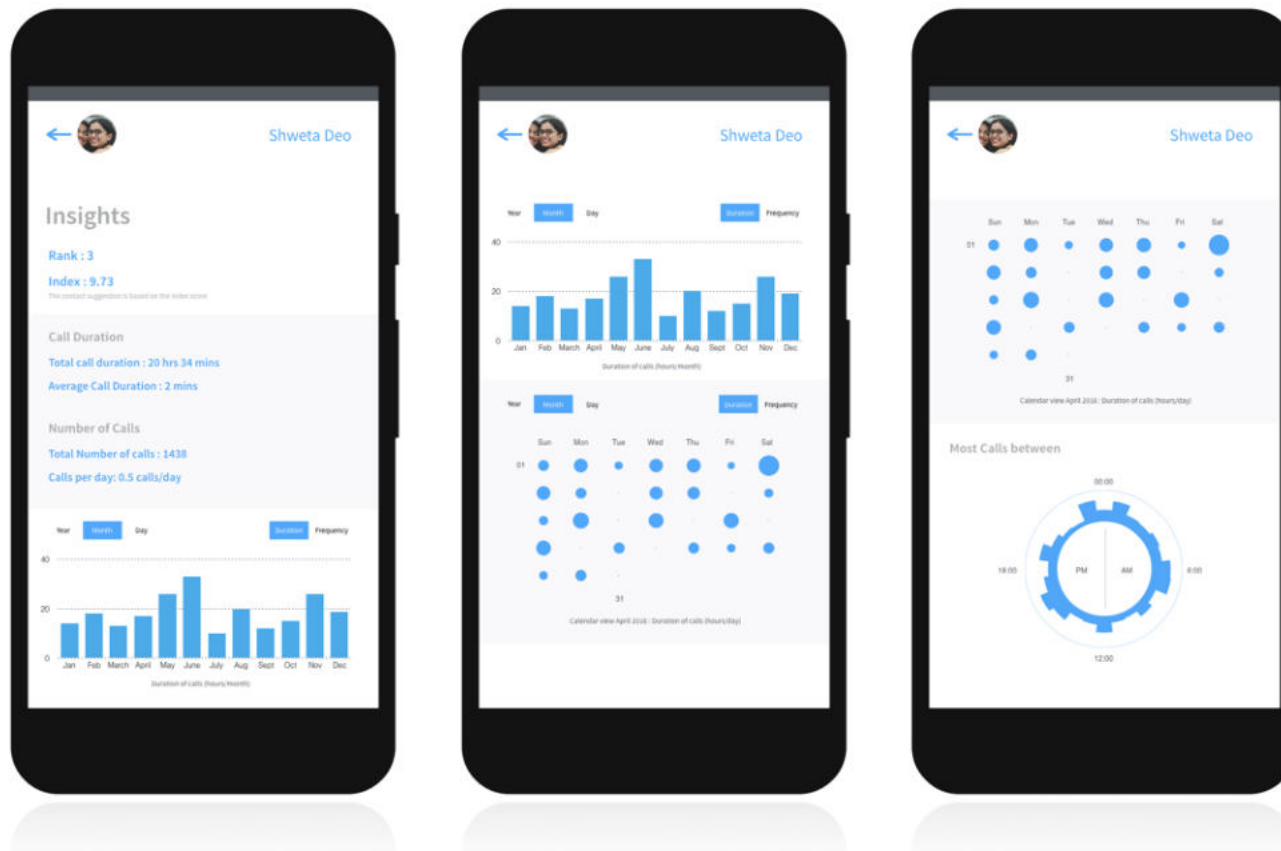
Contact View

The contact view can be accessed by a long press gesture on a contact icon and using the circular menu to select the ‘i’ icon. One can also tap directly on the picture in the contact element to access this view.

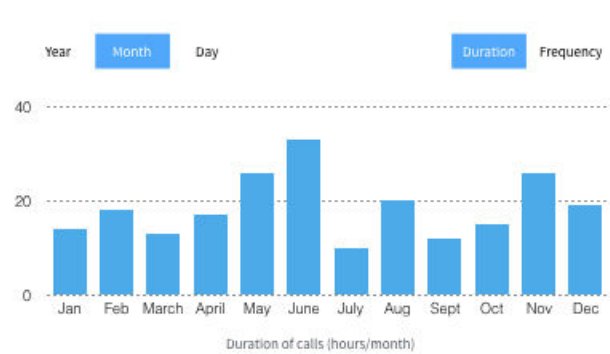
The contact view shows an overview of the contact information like phone numbers, email ids associated with a contact. It also presents general information, 5 recent interactions and insights about the contact in view.

Insights

Insights are a special feature in the PhonBk application to encourage exploration of contacts and their relationships. The exploration is meant to be serendipitous and hence the feature is hidden deep within the information architecture. The Insights feature enables exploration by a series of visualizations and rankings as well as index data of a contact.



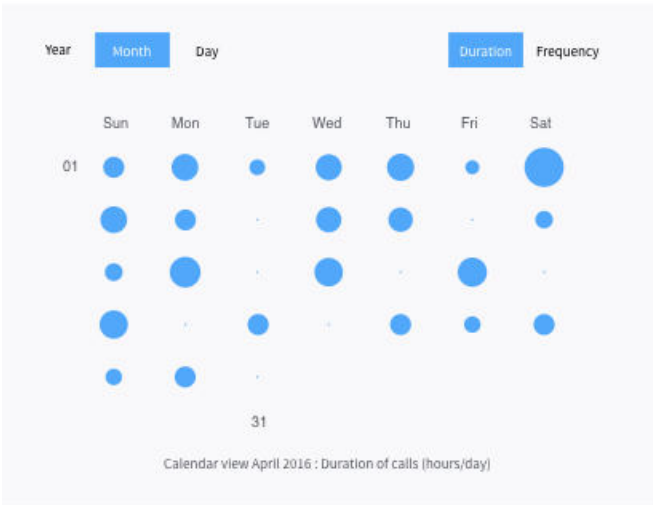
THE INSIGHTS FEATURE IS HIDDEN DEEP WITHIN THE APPLICATION ARCHITECTURE



HISTOGRAM VIEW OF INTERACTIONS WITH A CONTACT. THE USER CAN FILTER THE HISTOGRAM BASED ON THE DURATION OF CALLS OR THE FREQUENCY. THE RANGE BASED ON YEAR, MONTH OR DAY CAN BE VIEWED. IT ENABLES THE USER TO VIEW TRENDS IN CALLING PATTERNS.



RADIAL BAR SHOWING THE INTERACTIONS WITH A PARTICULAR CONTACT OVER A 24 HOUR PERIOD. IN THIS CASE, THE USER HAS INTERACTED WITH THE CONTACT MOST BETWEEN 10-22 PM



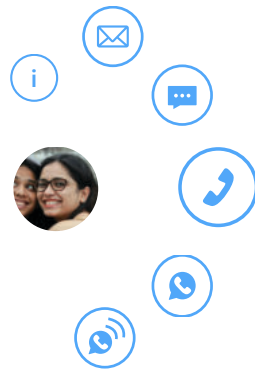
AREA MAP OF INTERACTIONS IN A CALENDAR VIEW. THE LARGER THE CIRCLE, THE MORE NUMBER OF INTERACTIONS OR MORE THE DURATION DEPENDING ON THE CURRENTLY SELECTED VIEW.

VISUALIZATIONS IN THE INSIGHTS FEATURE

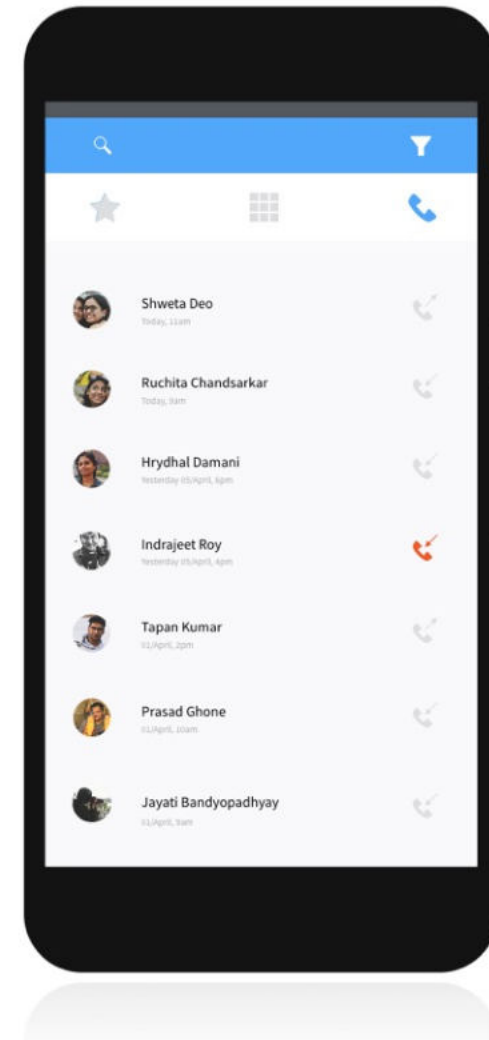
Other

The PhonBk Application also consists of standard features like a call log. In case of the call log, only missed calls are highlighted.

A circular menu pops up when a contact picture is long pressed. The circular menu consists of modes for quick access to the dialer, messenger, email and WhatsApp. The locations of the menu options with respect to the position of the finger on the screen is constant to imbibe the feature location in the user's muscle memory.



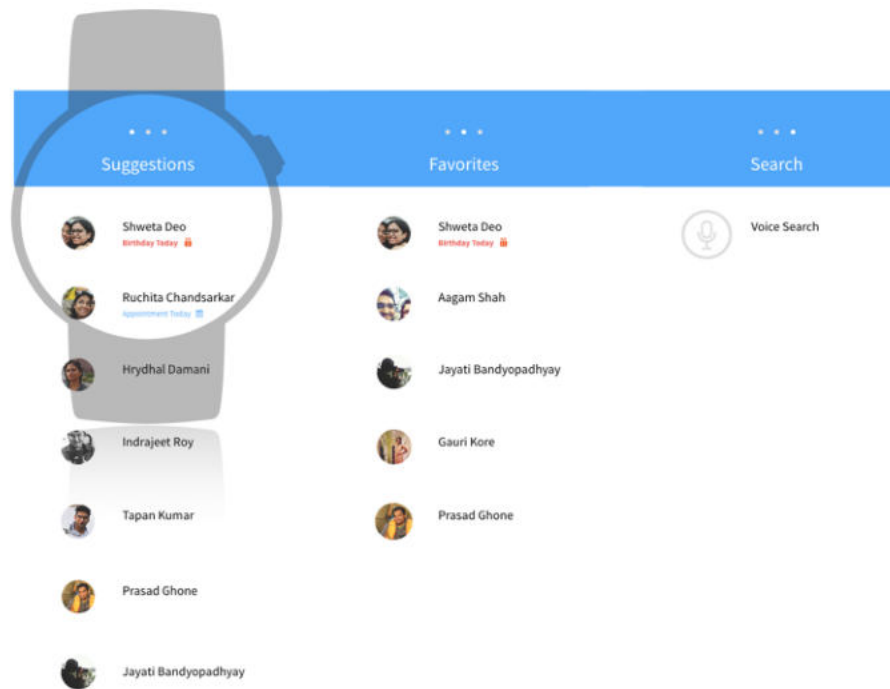
LONG PRESS MENU



CALL LOG SCREEN

PhonBk for Wear

The Suggestions feature of the phonebook along with favorites is a good solution for quick access of important contacts for mobile devices with a medium to large screen real estate available. However, these features become even more useful in cases where the screen real estate is limited and the user is not always able to access the entire Phonebook. A wearable device in the form of a smartwatch is a preferred use case for these features. A smartwatch does not have enough screen real estate for users to scroll or search through an entire phonebook with more than 1000 contacts.



5 Second Rule

The design principles for android wear suggest, that any action on the wearable device should not require the user more than 5 seconds to perform. A typical use of a wearable app is less than 5 seconds³³.

Hence, the PhonBk App for Android wear consists on only 3 functions for quick access of relevant contacts i.e. Suggestions, Favorites and Voice Search.

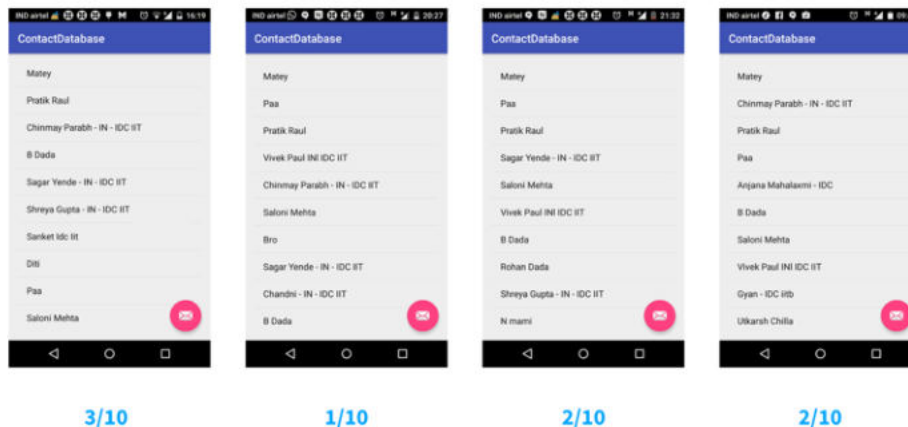
Evaluation

The design is to be evaluated for the accuracy of the suggestion mechanism and the usability of the advanced sorting and filters. The evaluation was done for a smartphone form factor used in a portrait mode.

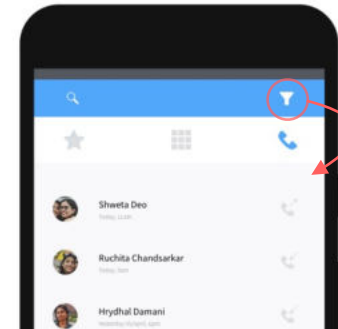
Suggestion mechanism

Evaluation Goal

To check the accuracy of the contact suggestion at different times of the day. To see whether relevant contacts are suggested.



A USER'S TOP 10 CONTACT SUGGESTIONS DURING DIFFERENT TIMES OF THE DAY. THE FIGURE INDICATES THE NUMBER OF INACCURATE SUGGESTIONS OUT OF 10. LEFT TO RIGHT, MORNING, AFTERNOON, EVENING, NIGHT



ADVANCED FILTER SHOULD BE LOWER IN HIERARCHY SINCE IT IS NOT A REGULAR USE FEATURE

Protocol

7 users were asked to use a custom suggestion application and open it during different time intervals for 2 days. The custom suggestion application was tested 56 times. The users were asked the relevancy of the contacts suggested at different times of the day. The users were asked to note down the number of relevant contacts out of the 10 contacts suggested i.e. contacts they would be prone to call at that time. Generic timings are used, namely morning, afternoon, evening and night. The underlying logic of the suggestions application was not revealed.

Findings

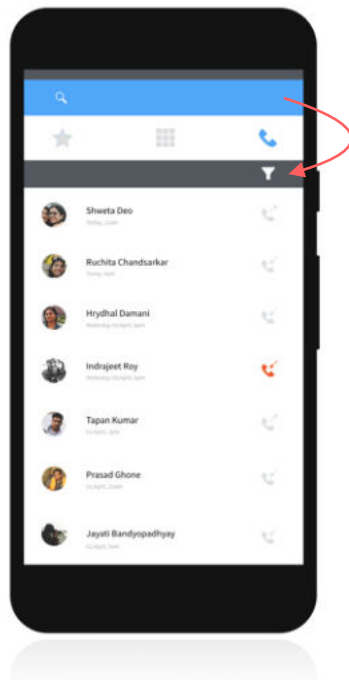
- For 5/7 users, the most accurate suggestions were during the daytime (2-7pm) since most calls were made or received during this time.
- The most irrelevant suggestions were shown during the late night (3-5am) due to lack of enough calls made or received during these times.

- It is important for the call logs to be older than 2 months for more relevancy in the suggestions, since the suggestion is based on the number of calls made over a longer period of time.
- The suggestions did not change overtime for one particular user since the user had deleted his call log.
- On an average 2 irrelevant contacts were suggested during each use.

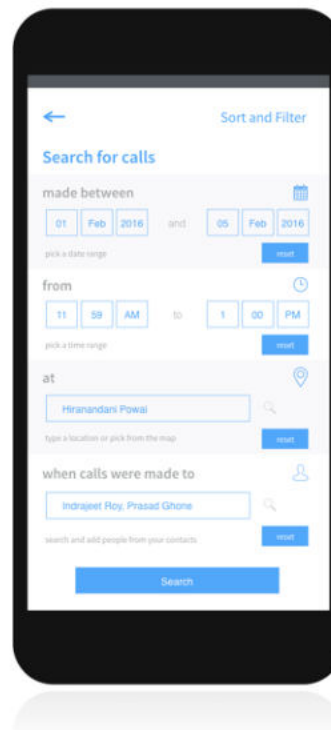
Advanced sort and filter

Evaluation Goal

To check the usability of the advanced sort and filtering utility and the usability of the microcopy. Are the advanced sorting and filtering features reasonably usable.



THE SORT AND FILTER UTILITY IS NO LONGER GLOBAL. IT IS LOCATED UNDER THE CALL LOG TAB



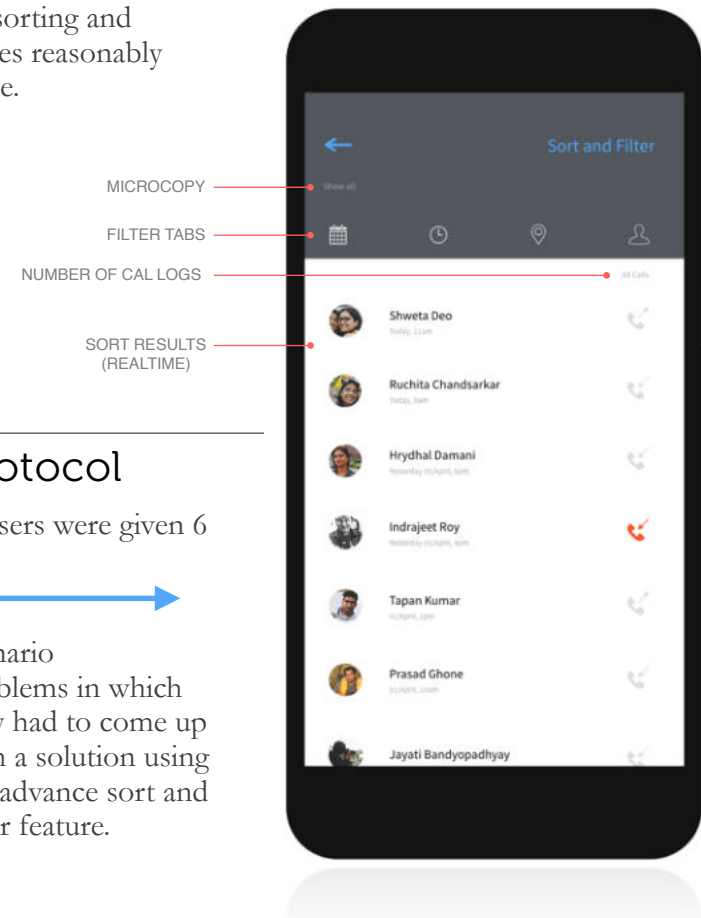
OLD DESIGN

Protocol

7 Users were given 6



scenario problems in which they had to come up with a solution using the advance sort and filter feature.



RE-DESIGN POST EVALUATION

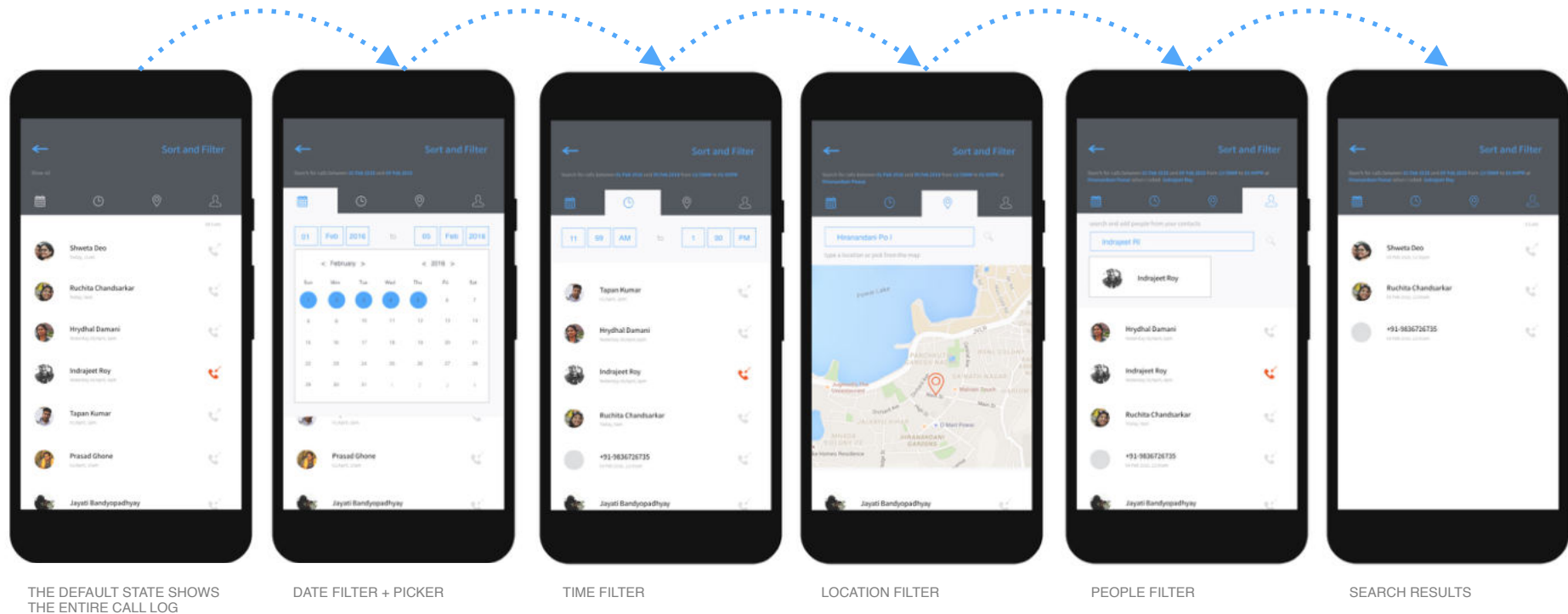
The feature is tested for its usability through a think aloud protocol. The participants would have to use the application while continuously thinking out loud, or verbalizing their thoughts as they move through the interface or perform the tasks given. The usability test is conducted with multiple users. Each user is given a single type of a smartphone form factor to use the tool. Following are the tasks that the users were asked to perform.

Task 1 : Using the date filter

The user is asked to find the call log list between two dates. This is an early success task and familiarize the user with the sort and filter interface.

Task 2 : Using the location filter

The user is asked to find all calls made at IDC IIT Bombay. This task is given to check the usability of the location filter.



Task 3 : Using the people filter

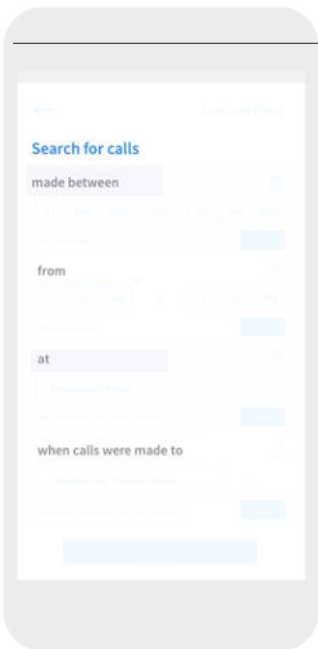
The user is asked to find the call log of days when calls were made to two specific contacts. This is a slightly complex task to familiarize with the people filter.

Task 4 : Using a combination of location and date filters

The user is asked to use a combination of location and date filters.

Task 5 : Using a combination of location, date and people filters

The user is asked to find list of calls made at Powai on a day/s when a call was made to a particular contact. This is a slightly complex task to perform. The task also checks the familiarity with using the location and people filters.



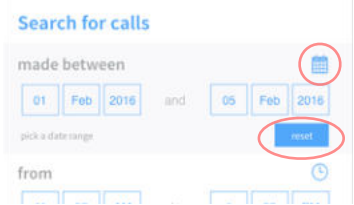
MOST USERS WERE NOT ABLE TO GUESS THE SENTENCE CONSTRUCTOR FOR THE QUERY.

Task 6 : Using a combination of all filters

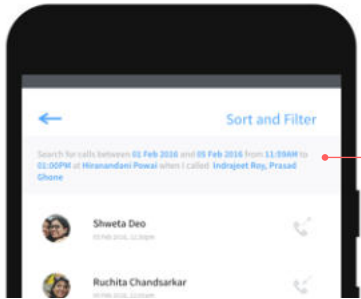
The user is asked to find all calls made in the last month at Powai between 1pm to 8pm when a call was made to Jayati. This task has the highest levels of complexity for the PhonBk application with the need to use all the functionality in sort and filter.

Findings

- Filter is for filtering the call log list, it should be in the call log list and not global.
- Unclear affordances for buttons and icons in the advanced navigation feature
- Use of range terms like last day, last week, last hour, etc for filtering by date and time instead of only the date filter would be beneficial.
- 6/7 users could not guess the sentence constructor for query.
- None of the users could guess the microcopy.
- Lack of immediate feedback of the call logs filtered was found to be a major hurdle in using the utility.



BUTTONS AND ICONS ARE NOT CLEARLY DIFFERENTIATED I.E. UNCLEAR AFFORDANCES



MICROCOPY

BUTTONS AND ICONS ARE NOT CLEARLY DIFFERENTIATED I.E. UNCLEAR AFFORDANCES

THE MICROCOPY WAS NOT EASILY PERCEIVED OF AS A FILTER BUTTON

Re-Design post evaluation

The findings from revealed certain immediate design alterations mainly related to feedback and redesign of affordances in the advanced sort and filter utility. The Advanced sort and filter utility has been redesigned with tab menus for filter options and the user gets a realtime sorted list as soon as any filter is in an active stage.

Scenario

“I had made a call to Rakesh in the first week of February early in the afternoon. Rakesh is an important lead who works at Microsoft at a senior level. I need to get in touch with an engineering team member for a project I am working on. Sadly, I did not save Rakesh’s number when he called me. I was busy having lunch with Prasad and Indrajeet at Hiranandani Gardens, Powai.”

Solution : In my call log of 1265 calls, I use the PhonBk app’s advanced navigation feature to search for calls made between 12pm to 1pm in the first week of February which yields 136 calls. I further filter the search based on the location from which the call was made i.e. Hiranandani Gardens, Powai which filters the list to 12 calls and the people whom I might have called that day i.e. Indrajeet and Prasad. This gives me a list of 3 calls, of which the two contacts have been previously saved and the unsaved phone number is probably the one I am looking for.

The following diagrams demonstrate the workflow.

Conclusion

The Phonebook application is fairly ubiquitous and enables what appears to be a small but important function of a phone, the function of enabling communication. Despite this importance, our phonebooks have not really improved drastically.

Smartphones with good computing capabilities have become fairly ubiquitous. These devices have the potential of collecting and storing large amounts of data about contacts, patterns of communication and other types of metadata such as location, context of the user and contact, relationship, anniversaries, etc. In a number of cases, smartphones already store this data. The 'PhonBk' application tries to embed intelligence in the phonebook by means of suggestions and enables novel ways of interacting with our contacts.

The prototype application would be deployed and a longitudinal evaluation would be conducted for the suggestion mechanism and the user interface for both the smartphone as well as the smartwatch form factor. Other possibilities of integrating social networks and enabling matching of contacts based on proximity or social contexts could be explored. The possibility of revealing the context of the contacts present and the ethical issues surrounding the same could also be looked at. The suggestion algorithm would be further improved by integrating social media information, context of contacts and based on the user feedback. Despite the future goals, the primary goal of the phonebook application is to enable the user to find the most relevant contact at any moment. The redesign of the Phonebook is in a sense, a billion people problem and any amount of time or effort saved, would give an exponential value to all users.

References

- ¹ Stefanis, V., Plessas, A., Komninos, A., & Garofalakis, J. (2012). Patterns of usage and context in interaction with communication support applications in mobile devices. Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services - MobileHCI '12. doi: 10.1145/2371574.2371579
- ² Komninos, A., & Liarokapis, D. (2009). The use of mobile contact list applications and a context-oriented framework to support their design. Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services - MobileHCI '09. doi:10.1145/1613858.1613951
- ³ Böcker, M., Suwita, A. 1999. Evaluating the Siemens C10 mobile phone – Beyond “Quick and Dirty” Usability Testing Proceedings of HFT’99, May 1999, Copenhagen, Denmark.
- ⁴ "Pareto Principle". Wikipedia. N. p., 2016. Web. 26 April 2016.
- ⁵ Whittaker, S., Jones, Q., Terveen, L. Contact Management: Identifying Contacts to Support Long- Term Communication. In *Proc. of CSCW 2002* , (2002), 216-225.
- ⁶ Stefanis, V., Plessas, A., Komninos, A., & Garofalakis, J. (2012). Patterns of usage and context in interaction with communication support applications in mobile devices. Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services - MobileHCI '12. doi: 10.1145/2371574.2371579
- ⁷ Jung, Y., Anttila, A., Blom, J. 2008. Designing for the Evolution of Mobile Contacts Applications. In *Proc. MobileHCI'08*, ACM Press (2008), pp. 449-452.
- ⁸ Stefanis, V., Plessas, A., Komninos, A., & Garofalakis, J. (2012). Patterns of usage and context in interaction with communication support applications in mobile devices. Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services - MobileHCI '12. doi: 10.1145/2371574.2371579
- ⁹ Stefanis, V., Plessas, A., Komninos, A., & Garofalakis, J. (2012). Patterns of usage and context in interaction with communication support applications in mobile devices. Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services - MobileHCI '12. doi: 10.1145/2371574.2371579
- ¹⁰ Jones, W., Phuwanartnurak, A. J., Gill, R., Bruce, H. Don't take my folders away! Organizing personal information to get things done. In *CHI '05 extended abstracts on human factors in computing systems*, ACM Press (2005), 1505-1508.
- ¹¹ Stefanis, V., Plessas, A., Komninos, A., & Garofalakis, J. (2012). Patterns of usage and context in interaction with communication support applications in mobile devices. Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services - MobileHCI '12. doi: 10.1145/2371574.2371579

- ¹² Stefanis, V., Plessas, A., Komninos, A., & Garofalakis, J. (2012). Patterns of usage and context in interaction with communication support applications in mobile devices. Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services - MobileHCI '12. doi:10.1145/2371574.2371579
- ¹³ Stefanis, V., Plessas, A., Komninos, A., & Garofalakis, J. (2012). Patterns of usage and context in interaction with communication support applications in mobile devices. Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services - MobileHCI '12. doi:10.1145/2371574.2371579
- ¹⁴ Whittaker, S., Jones, Q., & Terveen, L. (2002). Contact management. Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work - CSCW '02. doi:10.1145/587078.587109
- ¹⁵ Whittaker, S., Jones, Q., & Terveen, L. (2002). Contact management. Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work - CSCW '02. doi:10.1145/587078.587109
- ¹⁶ Whittaker, S., Jones, Q., Terveen, L., 2002. Contact Management: Identifying Contacts to Support Long-Term Communication. In Proceedings of CSCW 2002 (New Orleans, LA November 16–20, 2002), 216-225
- ¹⁷ Jahanbakhsh, K., King, V., & Shoja, G. C. (2012). Predicting human contacts in mobile social networks using supervised learning. Proceedings of the Fourth Annual Workshop on Simplifying Complex Networks for Practitioners - SIMPLEX '12. doi:10.1145/2184356.2184366
- ¹⁸ Whittaker, S., Jones, Q., Terveen, L., 2002. Contact Management: Identifying Contacts to Support Long-Term Communication. In Proceedings of CSCW 2002 (New Orleans, LA November 16–20, 2002), 216-225
- ¹⁹ Min, J., Kim, H., & Cho, S. (2008). Social and Personal Context Modeling for Contact List Recommendation on Mobile Device. 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology. doi:10.1109/wiat.2008.367
- ²⁰ Oulasvirta, A., Raento, M., Tiita, S. 2005. ContextContacts: Re-Designing Smartphone's Contact Book to Support Mobile Awareness and Collaboration. In Proc. MobileHCI'05, ACM Press (2005), pp. 167-174.
- ²¹ Millen, D., & Henderson, D. (2000). PhoneMan. CHI '00 Extended Abstracts on Human Factors in Computing Systems - CHI '00. doi:10.1145/633292.633379
- ²² Zhuang, X., Sun, Y., & Wei, K. (2014). SMOCoR: A Smart Mobile Contact Recommender Based on Smart Phone Data. 2014 IEEE 38th Annual Computer Software and Applications Conference. doi:10.1109/compsac.2014.47
- ²³ Whittaker, S., Jones, Q., Nardi, B., Terveen, L., Creech, M., Isaacs, E., & Hainsworth, J. (2002). ContactMap. Proceedings of the 2002 ACM on Computer Supported Cooperative Work Video Program - CSCW '02. doi:10.1145/616706.616714
- ²⁴ Stefanis, V., Komninos, A., Plessas, A., & Garofalakis, J. (2013). An interface for context-aware retrieval of mobile contacts. Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services - MobileHCI '13. doi:10.1145/2493190.2494432

- ²⁵ Tolmie, P., Pycock, J., Diggins, T., Maclean, A., & Karsenty, A. (2002). Unremarkable computing. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems Changing Our World, Changing Ourselves - CHI '02. doi:10.1145/503376.503448
- ²⁶ Oulasvirta, A., Raento, M., & Tiitta, S. (2005). ContextContacts. Proceedings of the 7th International Conference on Human Computer Interaction with Mobile Devices & Services - MobileHCI '05. doi:10.1145/1085777.1085805
- ²⁷ Truecaller. (n.d.). Retrieved February 1, 2016, from <https://www.truecaller.com/>
- ²⁸ Drupe for Android- Contacts and dialer. (n.d.). Retrieved May 24, 2016, from <http://www.getdrupe.com/>
- ²⁹ Min, J., Kim, H., & Cho, S. (2008). Social and Personal Context Modeling for Contact List Recommendation on Mobile Device. 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology. doi:10.1109/wiat.2008.367
- ³⁰ "Pagerank". Wikipedia. N. p., 2016. Web. 26 April 2016.
- ³¹ Saffer, D. (2013). Microinteractions. O'Reilly.
- ³² Saffer, D. (2013). Microinteractions. O'Reilly.
- ³³ Design Principles for Android Wear. (n.d.). Retrieved May 24, 2016, from <https://developer.android.com/design/wear/principles.html>