

Recomminders: Efficient (re)Discovery

Concept Videos Report

Asad Khaliq

Team Manager

Edwin Park

Development

Ji Park

Design

Quentin Perrot

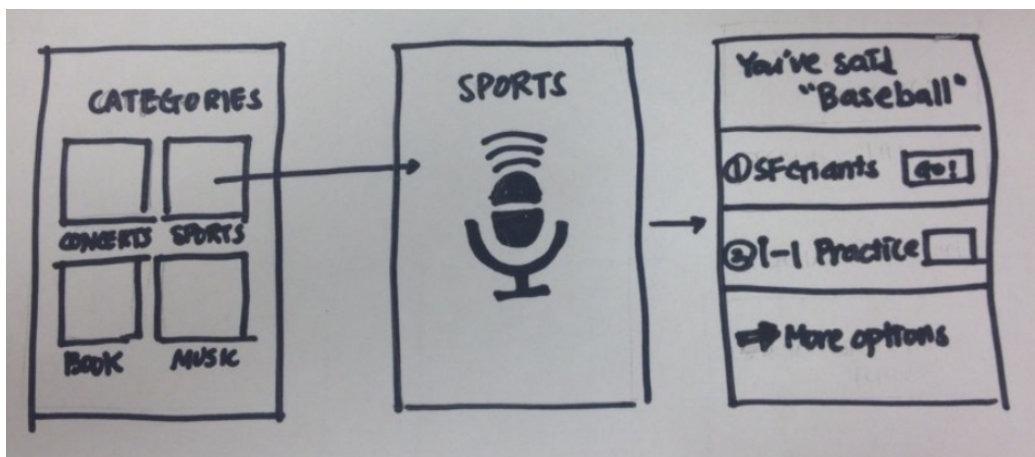
Documentation
User Testing

Problem and Solution Overview

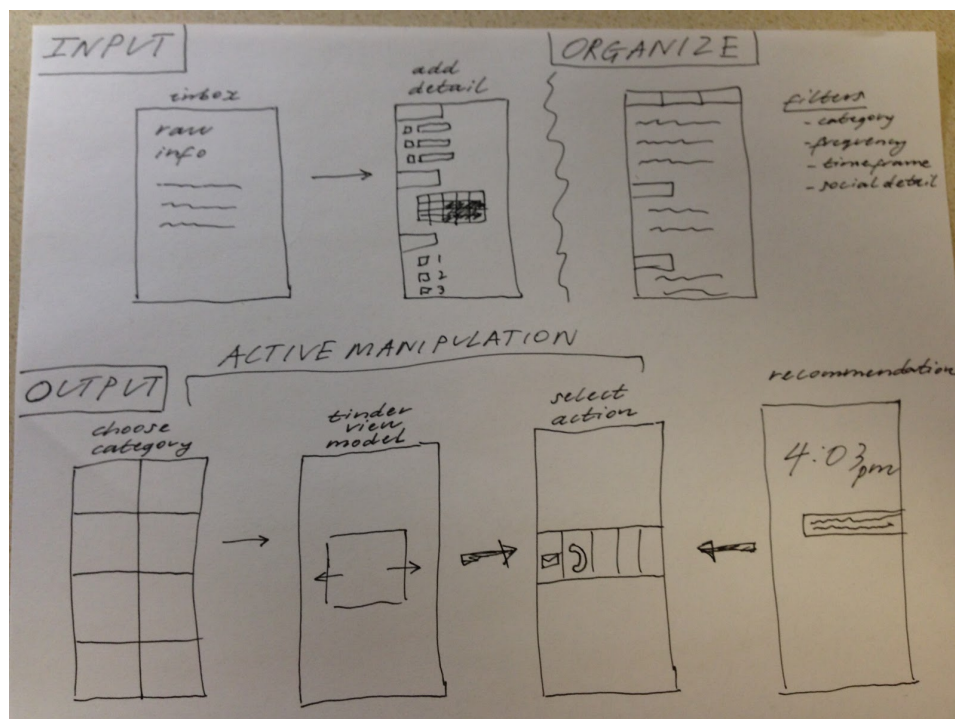
Our hectic lives mean we can't always focus on discovering the experiences around us as efficiently as we'd like. Our current methods of reminder are disconnected and inefficient. By recording and organizing our aspirations to discover in a centralized location, we can then efficiently utilize smart and location-based technology to help us revisit things we'd like to see and do at the right time and place. This technology will be able to sync with your calendar and track your movements, thus understanding when and where a recommendation would be appropriate and likely to be accepted.

UI Design Sketches

Interface 1: Spontaneous Speech Interface



Interface 2: More Active Features



Selected Interface Design

After careful deliberation, we chose our second interface design. Our thought process is described below.

Initial Desire for our Product

Our group's initial desire was to create a product that a user had to interact with very little, thereby maintaining a strong feeling of spontaneity when you receive a recommendation. Our first interface outlines a product that uses speech recognition to input recommendations, bypassing the need to input by hand. The product engine would then recommend at an opportune time, and once again, there would be minimal interaction. Many of our interviewees were busy people, and so in this context, a seamless interaction between phone and user would be a main selling point for us as it did not waste any of their time and functioned in the background.

Challenges and Problems with Speech-based Interface

This 'seamless' design presented certain challenges and defied some of the behavioral data we received through interviews. The main challenge this interface presented was: if you are given an incomplete recommendation through speech (a recommendation without enough detail for the engine to recommend at the right time and place), then the design failed. Given this problem, we needed two things: (1) a quick, easy, and fluid way of inputting recommendations (by speech, text or any other means) and (2) a way of 'detailing' recommendations so the engine can then manipulate them. Using our contextual inquiry data, we realized people usually look at their sticky notes and their notes application when they have a little downtime. Using this behavioral trait - looking at your sticky notes and changing them when you have downtime - we decided to give this action its own screen. This solves our engineering problem and mimics how today's user interacts with his/her data.

Secondly, a 'seamless' design bypassed the ability to go through inputted recommendations. In a search to make this design run in the background, we made the conscious decision to eliminate the ability to view your data in an 'organizer.' Bypassing this element increased the spontaneity of each recommendation, but once again went against some of our contextual inquiry data. During interviews, we found that people would not only value being recommended things at the right time and place, but be able to *actively* go through their data and plan things independently. As a result, our chosen interface incorporates a screen that allows the user to filter through recommendations and explore them further.

Why did we choose our current interface?

What is preserved in our chosen interface is the instant recommendation that arrives as a notification on your lock-screen. This element of spontaneity, we deemed, was very important to both our designs. Our chosen interface covers more use cases that were delineated during our contextual inquiry, and at the same time, maintains an option for

spontaneity. With this design, users are able to input and view their information easily and in a centralized location; each function of the app is designed to be used in specific environments with special constraints (time being a very important constraint); allows the user to organize their own data as they do in real life; allows users to both plan events on their own and be spontaneous.

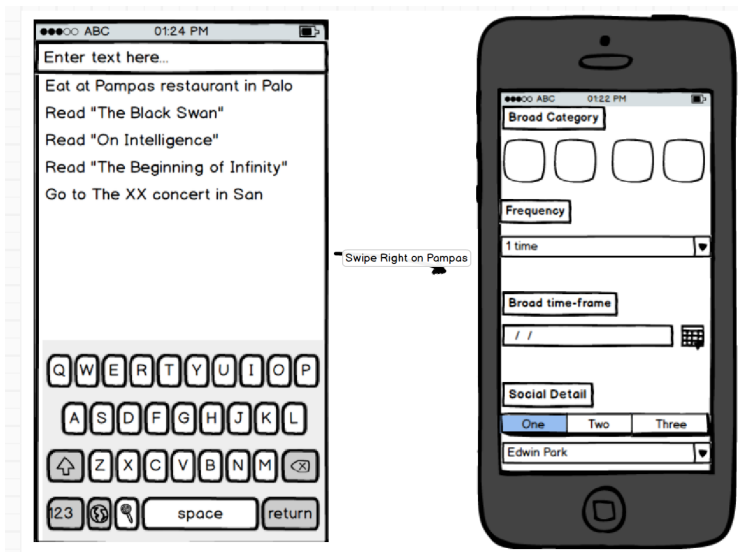
Functionality Table

Feature	Functionality	Context of Use
<i>Input recommendations</i>	Easily enter recommendations you receive many times a day into a fluid and quick interface	Anywhere and anytime. Take your phone out of your pocket, type a few things in, and you're done.
<i>Detail recommendations</i>	Easily annotate a recommendation to include additional detail	When you have downtime, quickly fill in some detail. This could be in the bathroom, in the car or in bed.
<i>View your data</i>	Seeing your detailed recommendations in a fluid and changeable way: apply filters, move things around etc.	Again, when you have downtime is best. Or when you want to plan an event and need some inspiration.
<i>Receive automatic notifications</i>	At the right time and place, receive a recommendation to do a thing you've always wanted to do.	Although we desire for recommendations to be spontaneous, they should occur at a good moment for you, in order to maximize the chance of you completing it.

UI Scenarios Storyboards

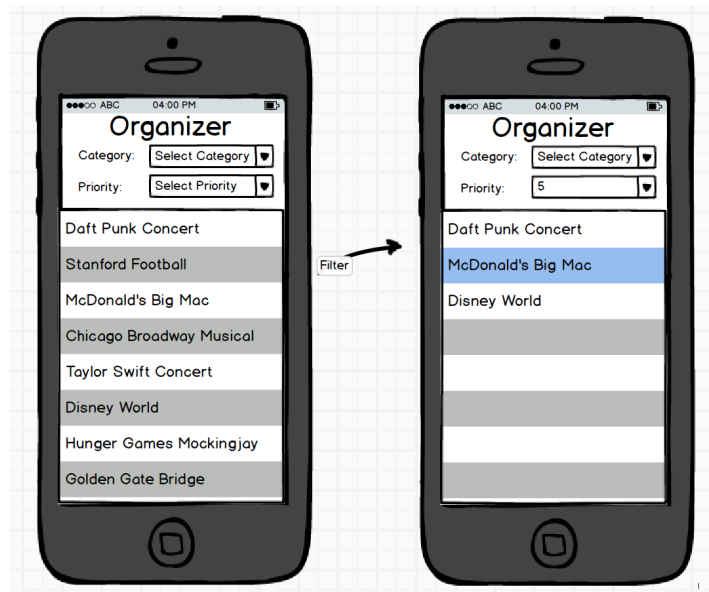
Task 1: Recording the Information (simple complexity)

When users are recommended something by a friend or see something they want to do, then they input the information briefly into the application. When users have more time, then they will organize and 'detail' their recommendation further, so the input can be used by the application's engine. For each list item in the left screen, a user will need to detail it as seen on the right.



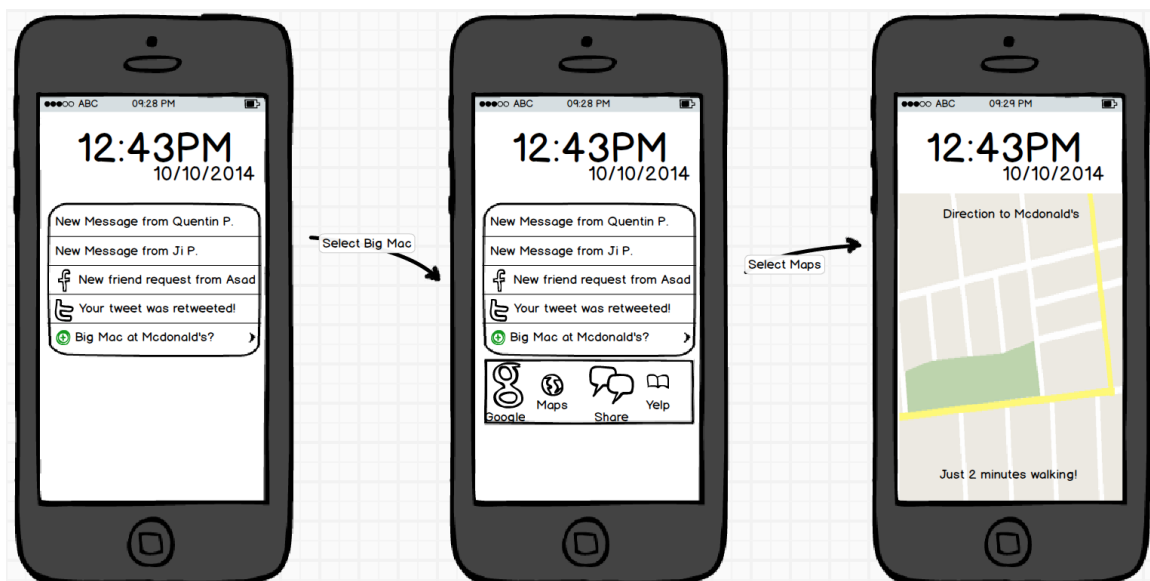
Task 2: Organizing and Viewing Data (moderate complexity)

Users are able to look through their detailed data in this organizer. Example functionality is a filter system. Users will be able to sort through their list by category, frequency, time frame, social detail etc. This is a quick and clean way of viewing data in a centralized location.

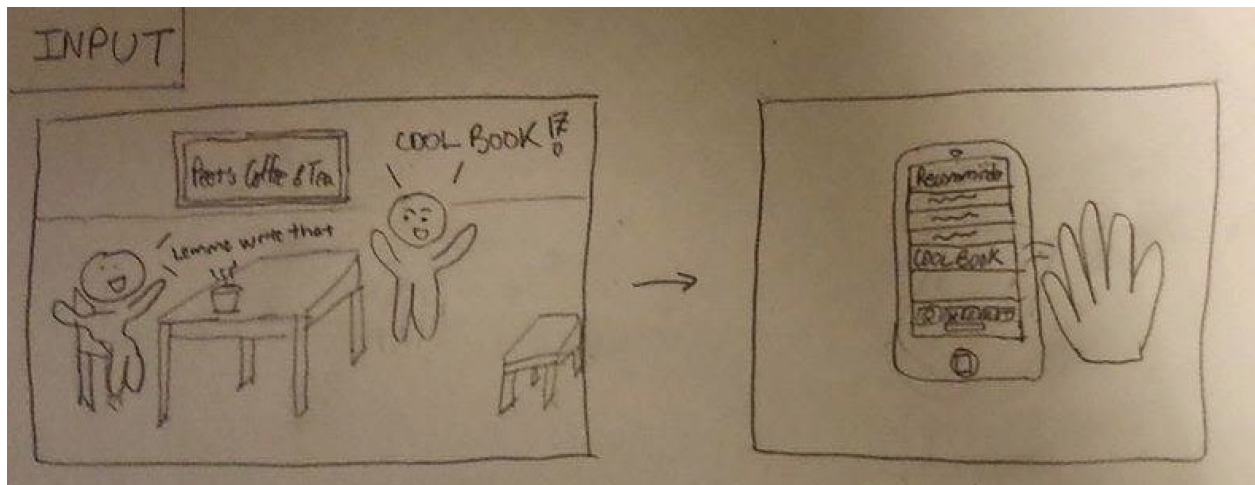


Task 3: Interacting with Automatic App Recommendations (complex)

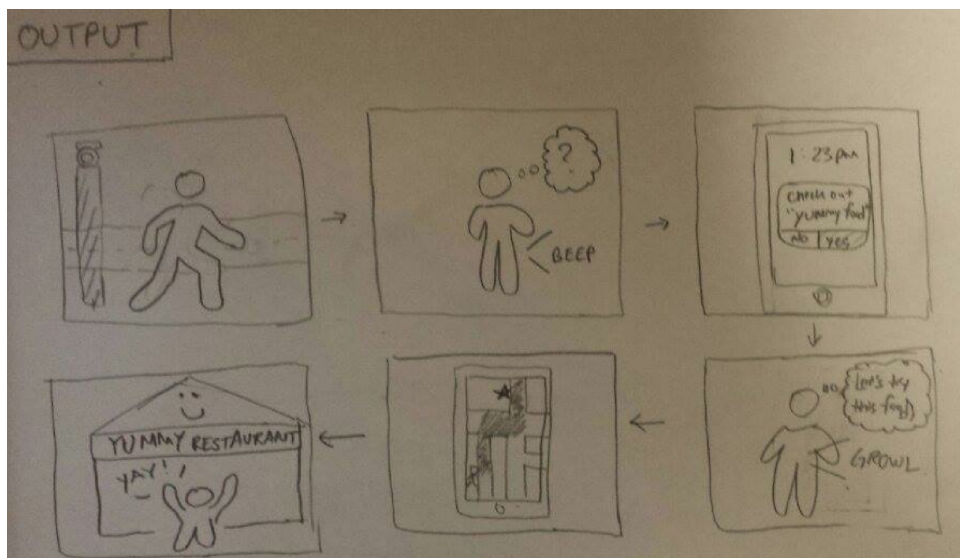
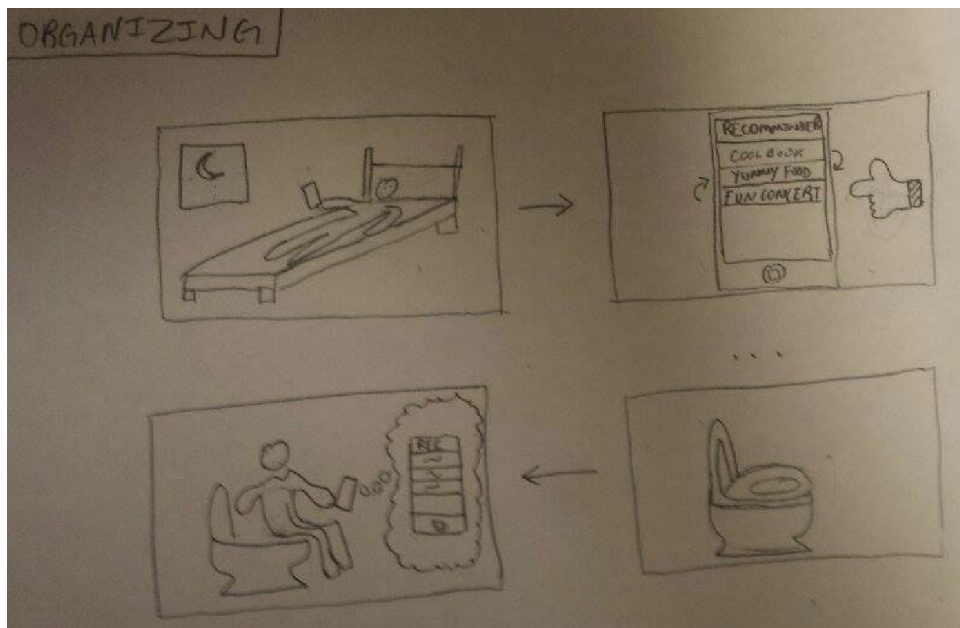
Users can receive notifications - reminders - of events that they wish to do, at the right time and place. In this scenario, users receive a notification at the right time and place, and are able to act on it - they are able to look up more information, use their maps, share with friends or even look for reviews.



Video Planning Storyboards



n



Concept Video Description

Key difficulties

Delivering a concise storyline without getting bogged down in the details was a major issue since our story had to be simple enough to be easy to follow (for someone who did not know about our idea and concept) yet still be compelling enough to make viewers want to use the product and buy into our idea. Since we did not have a UI - or concept - fleshed out fully, we had to make sure that our context and tasks were interesting and compelling enough.

Minimising the use of dialogue was a key problem since dialogue would easily provide exposition and allow us to set the scene, but comes with a set of drawbacks whilst also being hard to understand and follow if not done right. All our scenes were filmed with dialogue and we had to selectively edit the final video to ensure that we minimised dialogue yet still delivered a story that was easy to follow. Despite a fairly good result, we felt that there was still some room to further minimise dialogue.

What worked well

Having three broad tasks that naturally follow each other in our application/concept idea allowed us to cleanly break the video into three separate sections with small title text announcing each section. This meant the story was easier to follow than it otherwise would have been and allowed us to demonstrate a few key scenes pertaining to each of the three tasks as opposed to condensing each task into one scene and therefore limiting the amount of possible use-cases we could show.

Having a soundtrack (instead of dialogue or background noises) that was well-chosen meant that the video had just the right overarching tone. Even simple scenes such as walking out of a cafe or into a new restaurant were made to look meaningful and augmented the sense of adventure and exploration in the video.

How long did it take

Storyboarding, brainstorming, ideating and sketching possible UI ideas	5-6 hours
Shooting all video content	3-4 hours
Editing the video and refining	3 hours
Total	11-13 hours

Video Link: <http://vimeo.com/109111206>