## Scenarios

1. Miles visits Feeder for the first time and takes it for a “test drive”.
2. Miles uploads his OPML from Google reader on Feeder.
3. Miles is on a website and decides that he wants to subscribe to the content. He clicks on the Feeder browser button and is able to subscribe to the feed on Feeder.
4. Miles has subscribed to some feeds for some days but has not read a single feed on Feeder yet. He signs into Feeder and is able to see all his subscribed feeds. He is able to click on a specific feed to start reading items in that feed.
5. Miles has read feeds on Feeder at least once before. He signs into Feeder and is able to see all his subscribed feed. He also sees a “personalized items” link along with other feed links. Upon clicking that link he is able to read personalized items.
6. Miles starts using the service regulary and alternates between reading “personalized feeds” and individual feeds.
7. As a regular user, Miles has subscribed to a lot of different feeds and wants to take a look at all his subscribed feeds so he can unsubscribe from them. He decides to unsubscribe from a few.
8. Ed has scheduled the service to crawl at regular intervals.
9. Ed has scheduled the service to learn at regular intervals.
10. Ed has a monitoring system that analyzes logs.
11. Ed has a maintenance system that regularly archives old feeds and creates space in the db

## Feature Areas

1. View info about subscribed feeds
2. Add manually to subscribe feeds
3. Add OPML to subscribed feeds
4. Add via button to subscribe feeds
5. Remove from subscribed feeds
6. Read subscribed feeds
7. Read personalized feeds
8. Discover new feeds
9. Crawl feeds
10. Classify feeds
11. Archive old feed items
12. Analyze logs

## Crawl functional specification

Crawls feeds in the db and add items to the db.

1. There are no users in the system - no side effects.
2. There is a user who has not subscribed to any feeds – no side effects
3. There is a user who has subscribed to some of the feeds crawled which had new valid content – user should see new items appear.
4. There is a user who has subscribed to some of the feeds crawled which had no new valid content – user should see no items.

Classify classifies new items based on user’s reading habits.

1. There are no users in the system – Classify is a no-op.
2. There is a user who has not established a reading behavior because they have not read any items – Classify is a no-op.
3. There is a user who did not have any new items either because he did not subscribe to any feeds or the feeds that he subscribed to did not have any new valid items – Classify is a no-op.
4. There is a user who has established a normal reading behavior because they have read some items in the past – Classify does its job.
5. There is a user who has established a reading behavior of not liking anything they read because they have never clicked on any item they read – Classify will classify all new items as “uninteresting”.
6. There is a user who has established a reading behavior of liking everything they read because they have always clicked on all items they ever read – Classify will classify all new items as “interesting”.

Learn learns the users reading behavior.

1. There are no users in the system – Learn is a no-op.
2. There is a user who has not established any reading behavior because they have not read any items – Learn is a no-op.
3. There is a user who has a normal reading behavior – Learn does its job.
4. There is a user who has a reading behavior of not liking anything they read – Learn does its job with p(uninteresting) = 1.
5. There is a user who has a reading behavior of liking everything they ever read – Learn does its job with p(interesting) = 1.
6. There is a user who has some unread items along with a past reading history – Learn does its job on the read items and ignores the unread items.