1. PROBLEM STATEMENT

- Emails relevant to bookkeeping (invoices, statements, etc)
- Emails relevant to servers (server down, monthly uptime report)
- SPAM or irrelevant emails (Viagra, watch, loto). While mailbox has tools some always pass.
- Emails that can wait (news, magazines, etc. Nice to read but not essentials to productive work)
- Email that should be associated to a task, project or client
- As per Email as a first-class citizen: "Instead of one gigantic mail store, we should have a number of smaller ones that make sense to one's workflow (ex.: around projects, tasks, clients, etc.) and that can easily be shared and prioritized."

2. RELATED WORK ANALYSIS

Platform	**Algorithm**
NextCloud	Gaussian Naive Bayes
Google	Logistic Regression and Neural Networks
Outlook	?
Yahoo	?

2.1. Findings So Far

- Many research papers suggest Support Vector Machines (SVM) to outperform the rest of the ML algorithms for email classification. However, the leading platforms have not adopted it. Some more research will be required that will focus on,
    - Data set to be used
    - Supervised or Unsupervised Learning (*I think Supervised is better*)
    - Tiki compatibility
- SVM algorithm had the longest training time
- SVM algorithm with optimized parameters had the highest accuracy score
- Naive Bayes algorithm had the quickest predicting time
- Important Stuff to Keep in Mind
    - NextCloud claims to have used the local data to ensure user's privacy. Explore this more and find out how, and why?

- Reference links:

  - https://nextcloud.com/blog/nextcloud-mail-introduces-machine-learning-for-priority-inbox/
  - https://www.sciencedirect.com/science/article/pii/S2405844018353404
  - https://towardsdatascience.com/the-best-machine-learning-algorithm-for-email-classification-39888e7b1846
  - https://slidetodoc.com/a-study-of-supervised-spam-detection-applied-to/

3. PROPOSED STRATEGY SO FAR

Goal: Classify emails based on Projects or anything **user wants to filter or classify emails on.**

- The result of Auto Classification and Auto Filters is same. Though, the classification model can learn and adapt according to user's feedback, whereas if the filtered result is wrong in Auto Filters, then the user will simply have to live with that or perform manual alterations to the generated folder, or the code might have to get altered.
- There will be three kinds of folders:
  - By default: Inbox, Sent, Drafts, Trash, etc.
  - Made by users
  - Made by Machine Learning algorithms

4. IMPLEMENTATION STRATEGY

4.1. Tasks for Phase 1

1. UI Designing
   1. Wireframing
   2. Mockups
2. Find and finalize the best Searching tool

4.2. Tasks for Phase 2

1. Text Sanitization Mechanism
2. Tokenizer and Stemmers

4.3. Tasks for Phase 3

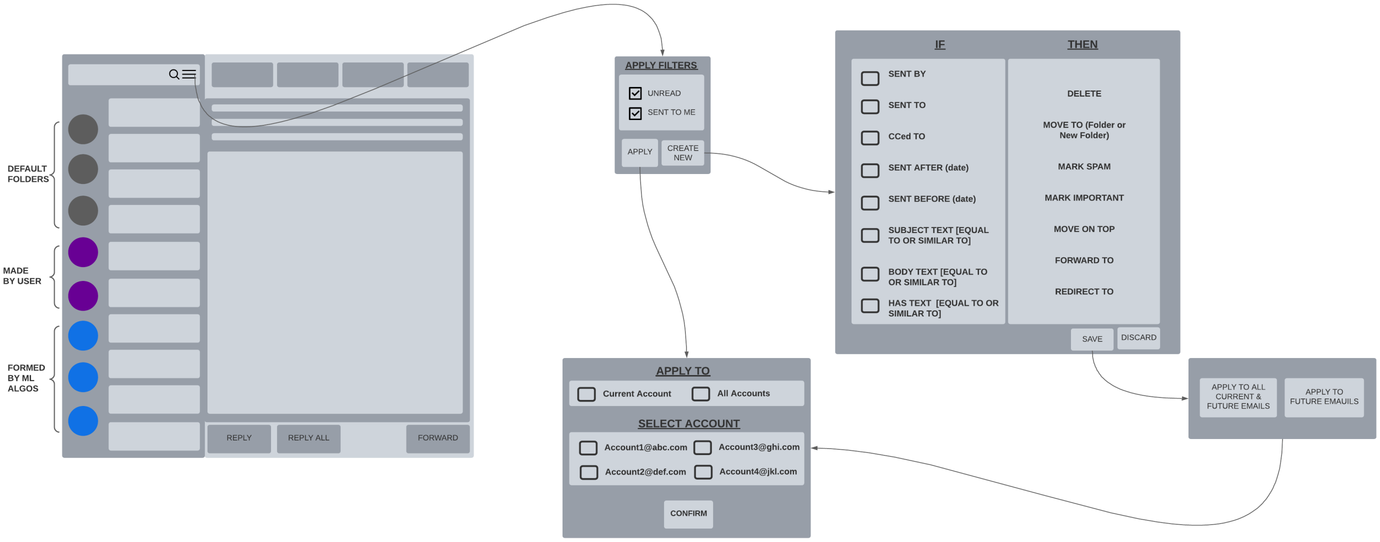
1. Make a list of top 3 best algorithms/classifiers (that are able to learn)
2. Explore and finalize best Bag of Words technique
3. Data set preparation,
    1. Get email data from Cypht
    2. Data preprocessing
4. Train model
5. Evaluate and compare the best model
6. Integrate the best model in Cypht for email classification

5. PHASE 1 - MANUAL FILTERS

5.1. Expected Features

1. Ability to apply the provided filters, such as,
    1. Unread
    2. Sent to me
    3. ?
2. Ability to apply customized filters (like Outlook's rules)
3. Ability to save customized filters

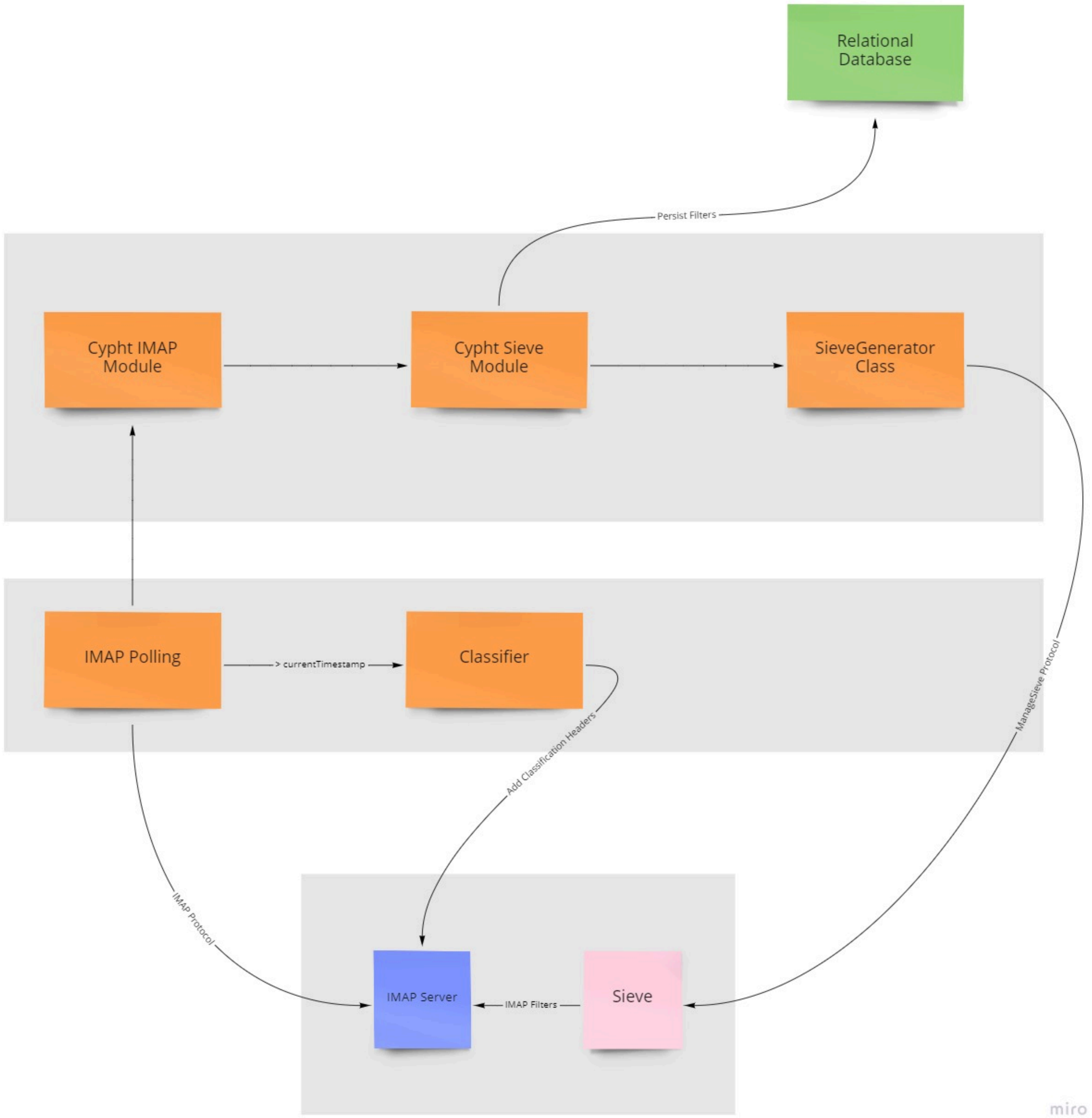
5.2. Wireframe



https://lucid.app/lucidspark/2323e28f-d365-4d03-ac56-8e4a46c8cec6/edit?invitationId=inv_0010e449-6e4a-4c84-93c0-b9212c140a9c

5.3. Tools and Technology

Sieve and **Ingo** will be used to apply the email filtering. Ingo is the "Email Filter Rules Manager", started as a frontend for the Sieve filter language. Following diagram shows the high-level architecture of the proposed solution.



The server side filtering will be done by Sieve. The ***managesieve*** service will make it possible to connect with Ingo to create, edit, enable and delete filter rules. For more details, refer to https://www.skrilnetz.net/server-side-mail-filtering-with-horde-ingo-and-sieve/
For more details on Sieve, refer to http://sieve.info/
For more details on Ingo, refer to https://github.com/horde/ingo and https://www.horde.org/apps/ingo

6. PHASE 2 - AUTO FILTERS

6.1. Expected Features

1. Filter spam email
2. Filter important emails
3. Integrate an external third-party spam filtering tool? (reference: https://www.quora.com/What-major-email-providers-filter-spam-well)

7. PHASE 3 - AUTO CLASSIFICATION

7.1. Expected Features

1. Filter spam email (point 1-4 are like Google's)
2. Filter important emails
3. Filter updates
4. Filter promotions
5. Make folders based on projects, clients
6. Learn and adapt on user's feedback and emails that they mostly search