

# Service-Oriented Computing Team 6 Project Proposal

## Mashup Homepage (tentative project name)

10-6-2021

**Hongjin Yu**

**Nahed Abdelgaber**

### Overview

We aim to create a customizable homepage for users that allows users to create their own mashups. Users will be able to choose from different data sources / APIs and display the data on their homepage. A stretch goal would be to also allow for scripting capabilities, so that users can combine and transform data from different sources. The product is targeted towards non-technical users, users will not need to know or care about any technologies under the hood: servers, API communications, databases, authentication, html/css, etc. In terms of ease of use, think of a facebook page. In terms of customizability, think of a physical desktop, where the user can put anything anywhere.

### Background

First observation:

Facebook, Youtube, Netflix, Instagram etc, push content to users and decide what users should see based on algorithm and user's past history.

This has several problems:

1. Algorithms will give the user 'junk food' style content, politically polarizing content, content that satisfies the user's short term compulsions but usually don't facilitate long term personal growth. This is further exacerbated by the trend of content creators chasing the algorithms so their content becomes more sensational and less wholesome overtime.
2. When algorithm make mistakes and give the user what they do not want, there is usually no way for user to direct content explicitly. Some services allow for users to provide feedback to the algorithm, but rules are always opaque.
3. Targeted ads and such are based off of user data collection, which have a multitude of problems. Privacy, unwanted products, exposing embarrassing life circumstances etc.
4. Users often want certain content to always show, and/or prioritize their content. For example they have a best friend and they will always leave a reply to that friend's posts. Or they want a section of their page to report the current score of a sports match. Or they want to track covid numbers / stock index etc. Current algorithm curated content will often push irrelevant content to the top and users must 'hunt' for content they really want.

Second observation:

Users will often have several favorite websites / platforms that they visit and switch between every day. However, most of the content on the website is noise to the user, and they really only care about a small fraction of the content.

- To solve the frequently used website problem, users have used the following strategies:
  - Pinning browser tabs
  - Having the sites bookmarked and reopen them every time manually / automatically on browser start
  - Browser plugins that take snapshots of the sites
- The above strategies are ad hoc and are not using a tool optimized for the requirement.
- There is no way for users to filter out noise.

### Third observation:

Many platforms have an ecosystem of software and services that provide the user with great value. However the software on each platform is always slightly different, with slightly different features and UI. The user has to context switch and relearn the same software multiple times. Take the simplest example, a weather app on windows and android. Both convey the same information yet each one has a different layout, and different ways to navigate.

## What we aim to solve

The above observations suggest that there is a niche in the market that is not being fulfilled, the product should meet the following requirements:

- Self directed.
  - Users are fed up by being fed. Users should be able to choose, with granularity what content they want to see.
  - Self directed also serves as a noise filter. Users constantly being fed by what the algorithm thinks will interest them will indeed keep their eyes glued to the screen in the short term, but will often lead to burnout in the long term.
- Content aggregation.
  - Today, users' attention is becoming fragmented. Users need to 'hunt' for the same content they want every day (opening up several different sources, navigating, filtering out noise).
  - Content should come to the user, when the user wants it, in the format the user wants, without additional baggage.
  - User should feel like a president or a CEO. Information flowing to them should be filtered and organized as if by a personal secretary. Instead, users today feel like slaves with multiple masters. Platforms ensnare users with social, psychological and biological pressures, and users are compelled to hunt for information on different platforms, with different rules and bombarded with noise.
- Fully customizable layout and priorities.
  - User should be able to set content priority instead of letting AI choose.
  - User organizes things the way they want to, and they stay that way.
- Unified experience between platforms.
  - Don't force the user to context switch to different platforms.

## What the project is

- Each user is given a 'homepage'.
- Each homepage has a number of boxes.
- User chooses what API / data source each box connects to.
- Present APIs to non-technical users in a easy to digest manner.
- User gets to enjoy personalized, self directed content from a myriad of sources in a single place.

## Similar concepts and why this is different

### Android widgets

- There is a widget for everything that shows you information you want on your android home screen. For example the author has one that directly controls his audio book player, and another that tracks stocks / cryptocurrencies.
- This is the most similar technology to what we want.
- We are different because we are using web technology, so the user will have a unified experience across platforms. e.g. logging in to the website with their phone / pc / mac / smart TV will show the same (responsive) content. Switching devices will not interrupt user experience.
- Android widgets is more limited as it must fit in one page and must adhere to android UI conventions.
- Android widgets is really out of place living in the same space as apps.

### Windows start menu and widgets

- Same concept as android widgets
- Platform dependent
- Poor ecosystem. No developers -> no users -> no developers -> no users -> ...
- We are using web technology which already has a good ecosystem.

### Browser homepage plugins

- Various browsers have plugins that allow the homepage to be customized to show snapshots of favorite websites.
- The website snapshots are limited in usefulness. A shrunken down snapshot of a website does not usually convey information a user really wants.

### Social media aggregation websites

- These are usually geared toward content creators to keep track / push to all sites, and for data analytics.
- Not geared toward content consumers.
- Not easy to use.

### RSS Feeds

- Not user friendly to set up
- Requires software

## What the deliverable will look like

### Minimum viable product

The scope of the project is very broad, but since we are building a site that allows users to create their own mashups, we can start small and build out from there. The very simplest version would be to just have one single API / data source the user can choose from (weather probably)

Login screen

Homepage

### **Boxes / iframes**

- Homepage will have boxes that show content from different APIs sources.
- User controls what data source each box connects to.

### **Stretch goals**

- Adding more API's to choose from
- Connecting to social media
- Connecting to Ad services and allow user to choose Ad content, user gets paid by Ads (not a far cry since many android are basically this)
- Multiple pages for each user. Work, play, etc.
- Boxes can be rearranged

### **Long stretch goals**

- Mashups usually provide additional value beyond just providing raw data.
- Allow users to connect data from different sources and allow for onsite scripting capabilities to combine data with all the power of a scripting language.
- Intuitive and non-technical visual scripting.
- User can train AI and direct AI explicitly instead of AI implicitly collecting data and constructing models. Don't discard AI, but give users the power to choose.