Idea Analysis Portal

PRODUCT DESIGN DOCUMENT

TEAM 87

Introduction

The aim of the project is to build a web portal to provide a platform for any organization's employees to post any new product idea and get their peer's reviews and detailed analysis using statistical and sentiment analysis techniques.

Critical User Journey

Login and Dashboard

- The portal is intended to provide a platform to discuss and analyse new product ideas within an organisation. Hence the first page on entering the portal will be a login portal where the user needs to authenticate using one's company mail id. No other feature is accessible without user authentication.
- On authentication, the user will be directed to the dashboard.
- The dashboard will have an option where the user can add a new product idea. He will be able to provide an idea title, idea explanation and description, associated links, upload demo image. He will be required to provide a category for the product like sustainability, privacy, etc. with an option of placing multiple categories to categorize the product idea. This point is covered in detail in the Post New idea section.
- The user will also be able to view all of one's previous idea submissions with a link on each which will direct it to the statistics page for further insights on reviews on each of one's previous ideas.

  Mocks:

  https://www.figma.com/file/uEyBCfWrvbQ4fPtKezvpQz/Login-Page
  https://www.figma.com/file/QGqU2DxOimzNStOMu3WMf5/Dashboard
  https://www.figma.com/file/fPCvtC44DkuILriIpWxFgm/Update-profile-form
  https://www.figma.com/file/mYZHjZmsSG6VZo7HPW4IQJ/Idea-Submission

Post New Idea

i) The user comes to this section from his dashboard by clicking on “Post New Idea” button.
ii) This page gives the user the option to add his idea title and description. These will be text boxes.
iii) He can also add images/videos. This will prompt him to select a file from his computer.
iv) He can choose the category his product belongs to.

Mocks

https://www.figma.com/file/MK2vbU2tnV6NEFKnUPRqhc/PostNewIdea

Common Feed

i) The user comes to the common feed through the common navbar link. Here he can see cards for ideas based on the date of posting.
ii) There will an option to filter products based on category.
iii) Each of the card takes the user to the Idea Description page on clicking on it.
v) The user can also access the navbar through which it can logout or go to his dashboard.

Mocks

https://www.figma.com/file/MrqozFGcI36Qf1KsBRbTng/CommonFeed?node-id=0%3A1

Idea Description

i) This section contains the description of the idea and includes photos and videos related to the idea.
ii) User can fill a survey for the product.
iii) Users can upvote/downvote the idea by clicking on the upvote/downvote button
iv) This section contains all the comments that are posted about the idea in the review section
v) The user can comment on the idea using comment button in the review section

Mocks

https://www.figma.com/file/nqUmh6BF5Fi0IKr06kXQjR/Idea-Description-Page

Comment Page and Survey Page

i) This section contains a form that asks the user to input the comment about the idea. This input is required.
ii) It also takes other input from the user like the age group to which the idea would be most useful, if any competitors of the idea are present. This information is optional for the user.
iii) After clicking submit, the comment gets displayed in the review section and statistic analysis page gets opened

Mocks

https://www.figma.com/file/B6MLwcz6J7CtH47QCbCoj3/Comment-form

Statistical Analysis

i) The user comes to this section of an idea via two routes:

- The user who submitted the idea through a personal dashboard.
- The user who submits a review after the review is submitted(to avoid human bias)

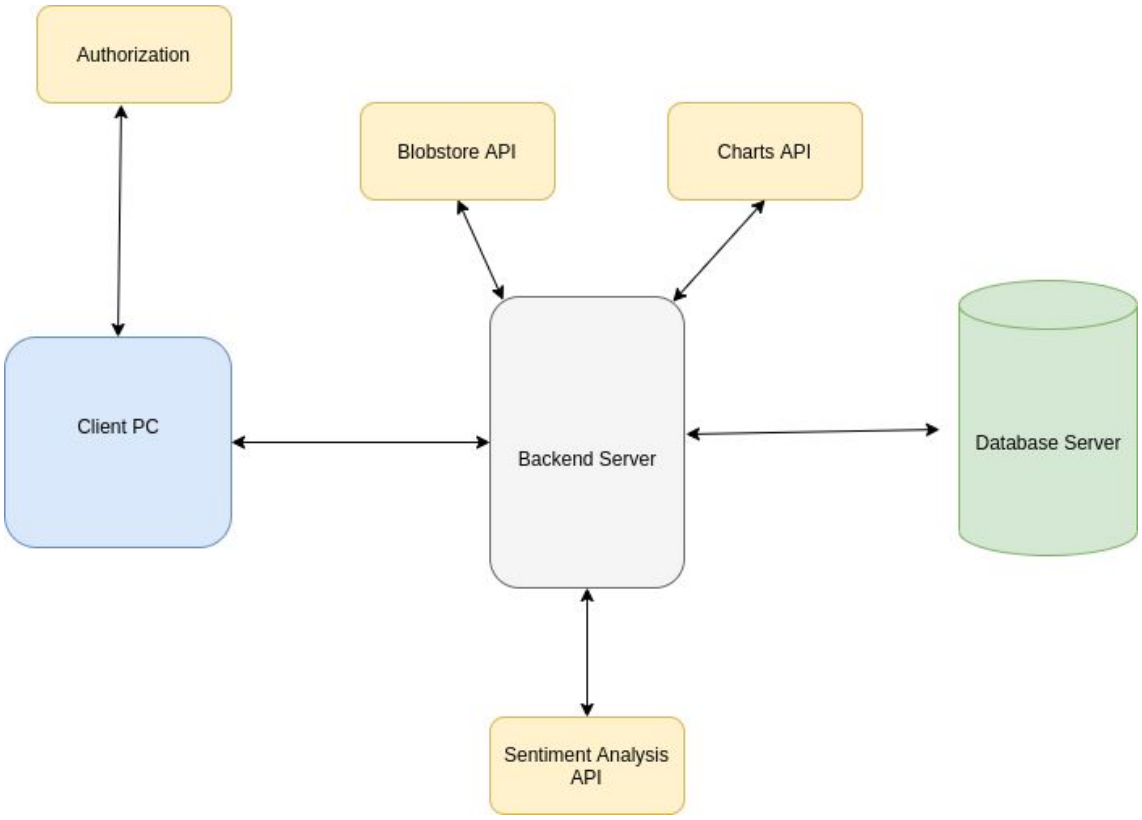
ii) The page contains a number of charts based on the data via reviews.

Mocks

https://www.figma.com/file/uS2G0o7fb9gLGJIzhJbq4K/Stats-Page

System Design

HLD



Components and Infrastructure

1) Frontend: Using HTML and CSS. Javascript is used to interact with the backend to fetch and post data.
2) Database: NoSQL (Firestore in Datastore Mode)
3) Backend: Java servlets will be used to interact with the database for post and get queries.
4) The website will be deployed and hosted using AppEngine.
5) Public APIs used:
    (1) Authentication: Any user will be required to authenticate oneself using company email to use the portal.
    (2) Blobstore: To store profile images of each user, image and video for each product.
    (3) Sentiment Analysis: To find a score (between -1 and 1) depicting relative positive/negative sentiment associated with the comment.

(4) Charts: To show all the info about the idea in the form of charts. The data will be taken from the database and the Charts API will just convert the data into charts.

APIs exposed from Backend

1) **Login**:
    a) postLoginInformation:
        (i) Extract email and password from the login form.
        (ii) Authenticates using the Authentication API.
        (iii) Adds user to database if not already added.
        (iv) Set a user login variable to be accessible throughout the project.
2) **Dashboard**:
    a) getPreviousIdeas:
        i) Based on the user logged in, fetch the ideas from the database for the particular user sorted on the basis of time.
        ii) For each idea, the data will be image and text description as it is only a brief display.
        iii) Return a json object of the same.
    b) getUserData:
        i) Based on the user logged in, fetch the user data from the database for the particular user.
        ii) Return Email, Name, DOB, Team information, ImageUrl.
    c) postUserData:
        i) Extract DOB, Team Information, Image from Update data form.
        ii) Use blobstore to generate url for the image.
        iii) Update DOB, Team Information, ImageUrl in database for the particular user.
    d) postNewIdea:
        i) Extract the following information from post a new idea form: Name, Description, Category, Demo Video, Image.
        ii) Use Blobstore to get Image and VIdeo Url
        iii) Add a new entry with the following fields: Name, Description, Category, Timestamp, ImageUrl, VideoUrl, Useridin the Product Idea database.
3) **Common Feed:**
    a) gotoIdeaDescription:

i) On clicking any of the cards, we send its key as input parameter and redirect to the page of the idea description. ThisAPI will be used in cards as well as carousel on the page.

b) filterBy:
   i) The options here allows to filter ideas by category.

4) Idea Description and Comments:

a) getIdeaDescription:
   i) Fetch data from the Product Idea database
   ii) Text description, images, videos, previous comments, categories, upvotes and downvotes about the idea will be displayed

b) postComment:
   i) Extract comment, age group, competitors, domain from the comment form
   ii) Store the extracted information in the database corresponding to the idea

c) IncreaseUpvote:
   i) Increases the upvote attribute in the Product Idea database by one

d) GetUpvote
   i) Fetch the updated upvotes

e) IncreaseDownvote
   i) Increases the downvote attribute in the Product Idea database by one

f) GetDownvote
   i) Fetch the updated downvotes

5) Post Idea Form Page:

a) PostIdeaSubmission: Fetches the data in the form and the images and videos and sends it to the backend. The backend sends this info to the database server.

6) **Statistics Page:**
    a) getLikedDataTable
        i. retrieves the frequencyArray for sentiment scores from database
        ii. creates and returns DataTable from the data retrieved with range (of sentiment score) and frequency

    b) getSuggestions
        i. retrieves a list of suggestions and extracts keywords using TF-IDF.

    c) getAgeGroupDataTable
        i) retrieves the frequencyArray for different age group from database
        ii) creates and returns DataTable from the data retrieved with “Age-Group” and “Frequency” fields.

    d) drawLikedChart
        i. Prepare the DataTable by calling getLikedDataTable with “State” and “Percent” fields.
        (State= Like/Dislike)
        ii. Customize the chart
        iii. Draw the chart
    e) drawPercentLikedChart
        i. Prepare the data by calling getLikedDataTable
        ii. Customize the chart
        iii. Draw the chart
    f) drawAgeChart
        i. prepare data by calling getAgeGroupDataTable
        ii. Customize and draw pie chart

Stats Computation

1) The **sentiment analysis** returns the score of each comment individually. We will calculate this score and normalise it to 10 whenever a comment is made. Advantage of using this approach is that we don’t have to keep a separate periodic thread and users can view the real time impact of their review which will improve the user experience.

2) Accordingly, we will calculate the **final rating** (which is the average of all comments for that idea), when the statistics page is loaded.

3) Depending on normalised score we will update the count of the specific array location of **Percentage of user liking/disliking.**
Eg: if score is 0-2, we will update arr[0], if score is 2-4 we will update arr[1] and so on. While displaying we will display it as a percentage of total comments in a chart.

4) **Upvotes and downvotes** will be calculated when stats page is loaded. So we will display them on a chart

5) For **age groups**, based on user responses, similar to point iv, we will calculate and display the percentage of suitable age groups on a chart based on user answers.

6) We will also use TF-IDF to extract important keywords from suggestions like competitors information, etc.

Database Schema

- **User**:
  (1) Email: String
  (2) Name: String
  (3) DOB: Date
  (4) Team Name: String
  (5) ImageUrl: String
- **ProductIdea**:
  (1) Name: String
  (2) Author ID: Key
  (3) Date Time of Issue: TImestamp
  (4) Category: String
  (5) ImageUrl: String
  (6) Description: String

- **Comment:**
  (1) AuthorId : key
  (2) ProductId : key
  (3) Text: String
  (4) Suggestion: String

(5) Sentiment Score: double
(6) Timestamp: timestamp

- **Survey:**
  (1) AuthorId : key
  (2) ProductId : key
  (3) AgeGroupCount : int (representing index of age group)
- **Vote:**
  (1) AuthorId : key
  (2) ProductId : key
  (3) voteValue : integer (-1 or 1)

Apart from this, one entity type will be created by Blobstore to store the images and corresponding URLs.