pecan

Team 20: Design Document

Abid Kaisani, Abir Shukla, Pranav Vasudha, Vedant Nevetia

Purpose

There are already many easy to use survey platforms out there; however, the goal of Pecan is to have the same easy to use vibe along with the additional feature of data analysis. Data analysis is extremely important in a day where data trends are becoming an ever increasing powerful tool in understanding how things work. Giving ease of access to this feature with options to download the data in tables and graphs in file formats such as .xlsx (Excel Workbook) allows us to put the power at the fingertips of the user; this grants users the ability to store, analyze data, or even feed it into Machine Learning making the possibilities endless.

Functional Requirements Include:

1. **Creating a User Account**
    - Grants ability for any user to be able to login and manage their account
    - Any user who creates an account will gain access to be able to send out surveys to any group of people
    - Users will have the ability to create set groups of people that makes it easier to send multiple surveys to the same sets of people
2. **Surveys**
    - Any amount of surveys can be created and managed by any particular user
    - Can be created/generated with base template designs
    - Users can customize their own template designs or modify already existing template designs when making surveys
3. **Survey Management**

- Users can adjust the date that any open survey that they created expires on
- Users have the option to view how many survey participants have submitted the survey, along with specifics on which participants excluding anonymous participants

4. **Survey Participants**
    - Have the option to submit a survey anonymously and securely even without having an account
    - Have the option to share a survey that was received
    - Have the option to block certain users from sending them surveys if they make an account
    - Can create an account to view a list of surveys that they have answered in the past
5. **Centralized Survey Hub**
    - Allows for users to view all surveys sent out by them in a centralized view
    - Allows for users to view all surveys received in a centralized view along with the expiration data on all of them
6. **Notifications**
    - All users have the option to set how often, on which platform, or if they receive notifications or not for new/expiring surveys that have not yet been answered
7. **Data Analysis**
    - All users will have ease of access to data file downloads
    - Users have the option to change which variables have accordance to which axis along with formatting options of the graph types that the data is displayed in

Design Issues

- **Issue 1: How should users be allowed to access recorded data?**
  - Option A: User credentials can be used to access data collected from a survey created using the same credentials.
  - Option B: Provide user with a link when a survey has been created that allows user to access the collected data.
  - Decision: **Option B**. User credentials will be utilized to ensure that users are not allowed to edit already existing forms or create illegitimate survey forms using an organization/individuals' identity. Additionally, the use of hyperlinks to a page that would host the users collected data seemed significantly intuitive given how easy it makes to share surveys amongst individuals and organizations. Furthermore, considering the scenario where the user has lost his/her data retrieving link, we will be creating an additional system that would retrieve the original link from our servers based on the information provided by the user. When considering Option A, it seems unintuitive to restrict survey data to just one account considering the constant need of everyday individuals to share data. Additionally, comparing the convenience that user data retrieval links introduce to the restrictions of user account specific data retrieval we chose to go forward with user data retrieval links for the retrieval of collected data.
- **Issue 2: What management and allocation methodology will be applied for accounts of multiple users of the same organization?**
  - Option A: Each user is provided with their own set of credentials regardless of the organization they are affiliated with.
  - Option B: Every organization is allocated one user account and the user credentials are then shared amongst members of the organization using our tool.

- Decision: **Option A**. Considering that each individual regardless of which organization they are afflicted with would be using our services simply to assess collected data. Consequently, it seems necessary to provide each individual rather than each organization a separate set of user credentials. It also ensures that supervisors are easily monitor that the correct surveys are available for their targeted individuals. Additionally, it would ensure that the changes made by an individual can be accounted for since they would be able to make them only using their credentials. Hence, we agreed on multiple sets of credentials being the better alternative out of the two.

- **Issue 3: Which platform will the backend be built upon?**
  - Option A: Python (Flask)
  - Option B: NodeJS
  - Decision: **Option A**. Python provides a simpler mechanism to manipulate data. Additionally, most of our team members are well versed when to comes to working with python and Flask. NodeJS has an extremely volatile nature, it is constantly changing. Furthermore, only a quarter of our team members were proficient in working with NodeJS. Considering these our two options and the characteristics that each one include into our workflow, we decided to work with python given the time it would save that could be reallocated into improving the product instead of learning a new platform to work with.

- **Issue 4: What number of surveys should an organization be allowed to create and edit simultaneously?**
  - Option A: Infinite number of survey creation and deletion by one user simultaneously.
  - Option B: Limited restrictions on the survey creation and deletion done by one user simultaneously. (5-10)

- Option C: Extremely limited restrictions on the survey creation and deletion done by one user simultaneously. (1-4)
- Decision: **Option B**. Considering the three options, Option B seemed to most realistic since it is highly unlikely that an organization will be creating or editing more than 10 surveys cumulatively. Hence, Option A was neglected. Option B was preferred over option C since the number made available was considered to be extremely restricting by our team. Consequently, option B seemed to be the most reasonable choice out of the three options available.

- **Issue 5: Which platform should the application be initially designed for?**
  - Option A: The application will initially be web-based.
  - Option B: The application will initially be designed for mobile platforms.
  - Decision: **Option A**. Considering that most mobile operating systems now have complex web views that can easily interact with any web-based application without any issue it seems to be a better choice to make a web application. Additionally, if the web application is made mobile friendly the web views will be able to handle interactions significantly improved manner. Furthermore, initial mobile based applications can easily be released if the web application is designed to be mobile friendly. Consequently, we decided to focus on the web application for now.

- **Issue 6: How should users be able to access surveys?**
  - Option A: Allow surveys to only be accessible by links generated once the survey has been created.
  - Option B: Allow users to access surveys through a survey feed that allows users to answer newly created surveys.

- Decision: **Option B**. Our product is entirely based on the availability of surveys to our users. Option A does allow users to get access to surveys but this approach seems restrictive since the user outreach would be limited. The limitation to the outreach of a certain survey is primarily caused by the fact that the base of users would have to be reached and it is highly unlikely that an organization will be able to reach all our users who opt answer surveys. Option B offers greater user outreach simply because every user will be able to view the same surveys and every survey will be made available to every user. This feed will be available to users through an email sent on regular time intervals based on the users' preference. Consequently, given the superiority that the option B has we decided to go with option B.

- **Issue 7: What is the user workflow model going to be?**
  - Option A: Continuous, the user can continue creating/editing or answering a survey from where they left off.
  - Option B: Momentary, the users will lose their progress if survey creation or editing in not completed during one instance.
  - Decision: **Option A**: Option A will allow users to have a sense of continuity in their workflow additionally, this feature will not be difficult to implement into our product as it does not require us to build another sub-product. Additionally, the ability to continue any survey from where it was left of seems to provide additional value to our user which would most definitely be a factor for the users to utilize our services as opposed to our competitors. Option B would ensure that there are no data corruptions and all data entered is complete and so would result in better data integrity quite specifically because a certain set of credentials will be used by an entire organization. Considering the low probability of a survey left uncompleted by a user if not able to continue from where they left off, we decided to

implement option A into our product to make the users' workflow significantly more convenient.

- **Issue 8: Dealing with uncompleted survey data when representing data.**
  - Option A: Include uncompleted form data when quantifying statistical data
  - Option B: Ignore uncompleted form data when quantifying statistical data
  - Decision: **Option B**. Option A would suggest that we include whatever information we can collect through our surveys to the user who created the survey but this would include data that might have been entered considering a certain context and due to the uncomplete nature of the data the context cannot be derived which would leave the data unviable. This inclusion of incomplete data would thus include discrepancies into our system. Option B if uncompleted data is ignored then we would have a decreased amount of data points to generate graphical representation of data. Nonetheless, this would ensure that the data entered by users, ideally, is accurate and so would increase the credibility of the gather data using our product. Evaluating our two choices we decided to pick Option B.
- **Issue 9: Which database platform should we use?**
  - Option A: MongoDB
  - Option B: Parse
  - Option C: Firebase
  - Decision: **Option C**. Option A suggest that we work with MongoDB as our database platform for this project. Given the nature of our product we noticed that it seemed unnecessary for our product to use such a sophisticated backend service. Option B suggests

that we use Parse as our database platform. Parse although simple and is easy to work with. Our team noticed a scarcity of individuals who had experience working with Parse and so were reluctant to learn to work with Parse. Option C suggest that we work with Firebase. Firebase is known to make development faster and majority of our team members had prior experience with working with Firebase. Given that the majority of our team preferred Firebase to be their primary database platform solution with similar characteristics to that of Pecan, we considered Option C to be our best option out of the three.

- **Issue 10: Which front end framework will we be using?**
  - Option A: Angular
  - Option B: Riot
  - Option C: React
  - Decision: **Option A**. Considering the three frameworks that we picked to work with Option A seemed to be the one which our team was most proficient in working with. Even though the other two their benefits to offer we considered the time being saved by working with a familiar framework capable of being allocated to another portion of Pecan.
- **Issue 11: What kind of a design scheme will Pecan use?**
  - Option A: Responsive
  - Option B: Adaptive
  - Decision: **Option A**. Pecan initially will be a web application and will be available to other platforms through the web as opposed to having a dedicated application for each platform. Therefore, it seems more reasonable for us to opt towards a responsive design scheme

since it would ensure a reliable product that individuals would appreciate whilst using Pecan. Consequently, Pecan will be a web application that considers the environment it's being run on and responds to it accordingly.

- **Issue 12: How will users receive notifications for new surveys?**
  - Option A: Email
  - Option B: In-app Notifications
  - Option C: Text Messages
  - Decision: **Option A**. When considering option C we noticed that most individuals tend not to go through their texts after a short period of time regarding subjects such as surveys or irrelevant quizzes. Additionally, there is a probability that due to the immense amount of text messages sent users would treat our messages as spam. We could allow users to set up a certain duration after which they could be notified about new surveys being available for them or the progress of the outreach of the surveys they created. Option B would allow us to not interfere with the users' day. When the user would begin using our system our notifications will provide a higher values and so the user is likely to interact with the notifications. Option A suggests that we simply use Emails to interact with our user base in regards to notifying them with newly available surveys. Organizations could use this as a way to spread the survey through their own and our mailing lists. Additionally, users could select a time interval by which they want to be notified about new surveys. Considering the intuitiveness that Option A provides alongside its seamless nature we settled on Option A.

Design Issues

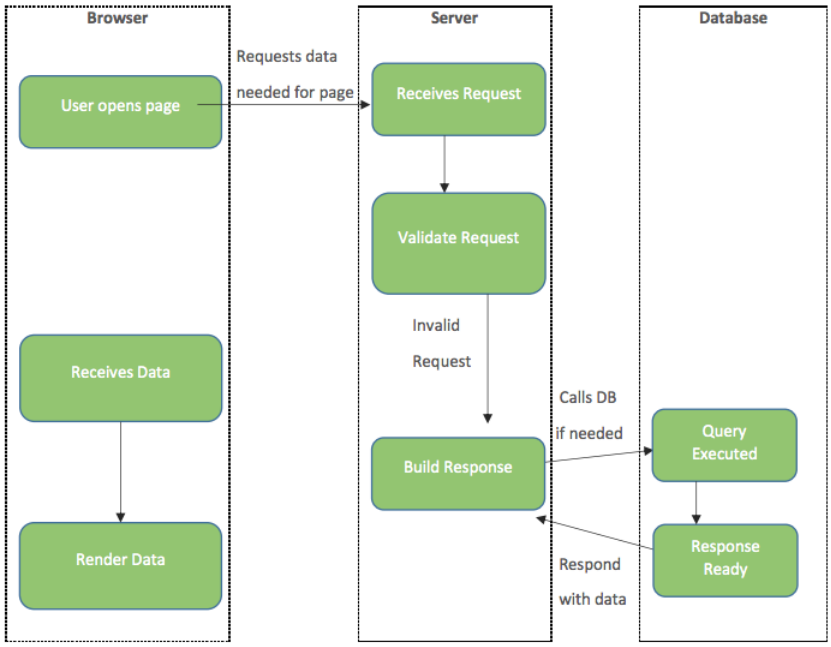
High Level Overview

Our Project will use the client-server model. The server will respond to the client requests, and the client will parse and render the response data using the Flask python based micro framework. The figure below demonstrates the high level overview of the system.

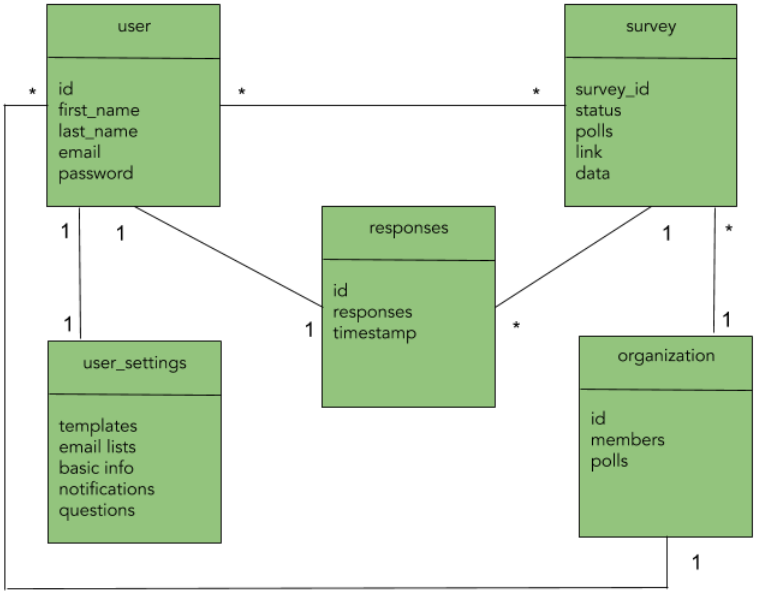
1. Client
    - Client send HTML from request to server API using method "POST"
    - Client receives a response from Server
    - Response data is interpreted and rendered using Flask
2. Server
    - Receives and handles all client requests
    - Validates requests before processing
    - Queries database and returns appropriate response to client (error message, webpage, etc.)
3. Database
    - Database stores all system information, such as users (and all information pertaining to them), current open surveys, and survey responses.

Flow of Events

The diagram below shows a typical flow of events. The sequence begins with a user opening the web application through their browser. If the client requires data from the server in order to complete the page, the app will ask for data from the user and the client will send that data to the server. The server will first validate the request taking out risk of users without certain permission to gain or modify data that they are not supposed to. If Client HTML Form Request API Response Client Client DB Query Query Response the request if valid, the server will generate a response, communicating with the database if needed, and will return the response to the client. If the request is invalid, then the client is trying to access information they are not allowed to, so an error response will be generated and returned to the client. The client will then parse the request and return the appropriate webpage with the appropriate data for the user to view and/or download.



Design Details



Description

1. User
   (a) User is a generic user, capable of creating surveys and gathering data from the surveys
   (b) Contains name, email, password and list of all surveys created and answered by this user
2. Organization
   (a) Contains multiple *members*, representing members of the organization
   (b) Contains a list of all surveys created members of the organization on behalf of the organization
3. Survey

(a) Contains a list of all questions to be asked by the survey

(b) Contains a unique identifier

(c) Users may create multiple surveys under their account and distribute the survey via the *link*

(d) Surveys contain a status to indicate whether they are open or closed

4. Responses

   - Includes answers to survey polls and the time they were answered

5. User Settings

(a) Includes settings and things unique to every user

(b) Custom survey templates created by the user

(c) Custom email lists made by the user to make it quicker to send out surveys to multiple users

(d) Default data format to obtain the data in

(e) Notification settings for surveys, etc.

(f) Basic information about users that can be easily plugged into surveys should those questions be asked

User Authentication

Authentication will be handled for the most part by Flask and Firebase, as shown in the diagram below. We will be routing all user login requests to the server, where Flask will take care of bundling the user and password combinations and sending them for validation to Firebase. A brief overview of the steps will be as follows:

1. User requests access to a certain restricted page

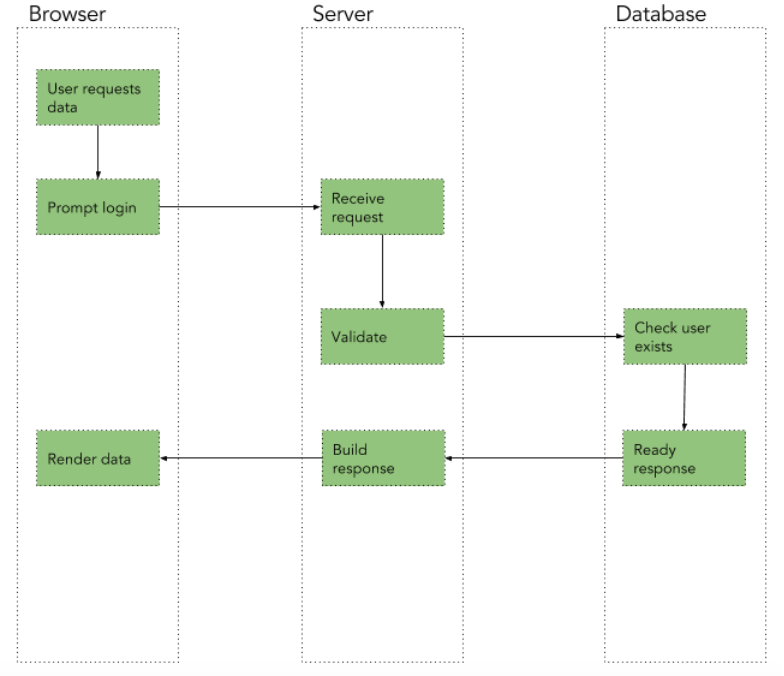
2. User prompted to login

3. User's credentials are taken from the form, and Flask queries the Firebase database to authenticate user

4. If the login was valid, the database will respond with the corresponding data, which Flask will then use and render the appropriate data

5. If the login was invalid, the database will again respond with the appropriate data, which will be handled by Flask and the corresponding data will be rendered to the user

6. All future data requests will be made within the user's context, and validated to make sure there is no unauthorized access

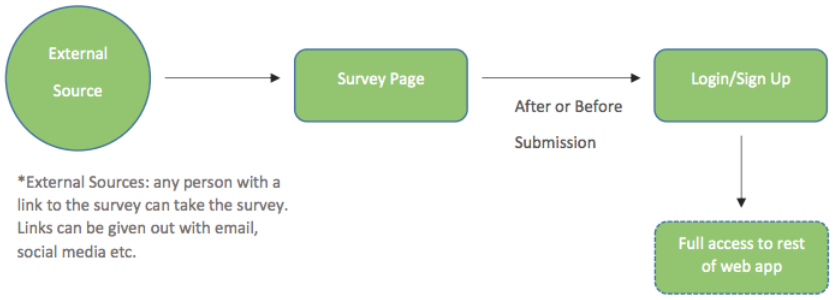


Navigation Flow Map

Our navigation focuses on ease of access on all pages. Initially the user would have to login/or create an account providing a small amount of information for the application to best serve theme, then the user will be brought to the homepage. The sidebar allows users to access any page from any page in just

one click, this makes navigation both easy and efficient, while keeping a simple and minimalistic design. The homepage allows users to navigate to new/recent surveys, "create a survey", "survey analytics", and "user settings" in one click. Surveys will be sorted by both topic and time of publish, other surveys can be reached either by link or by search. The survey, and add survey pages will ask users for information in a simplistic and aesthetic manner, and the "survey analytics" page will display and provide data in the same manner.

For Non-Users:



For Users:

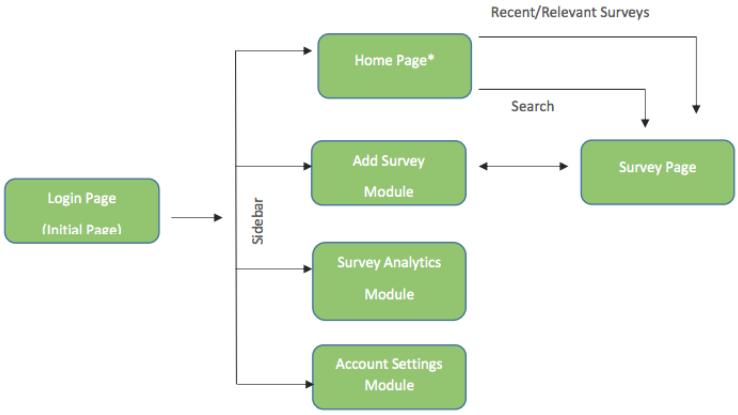
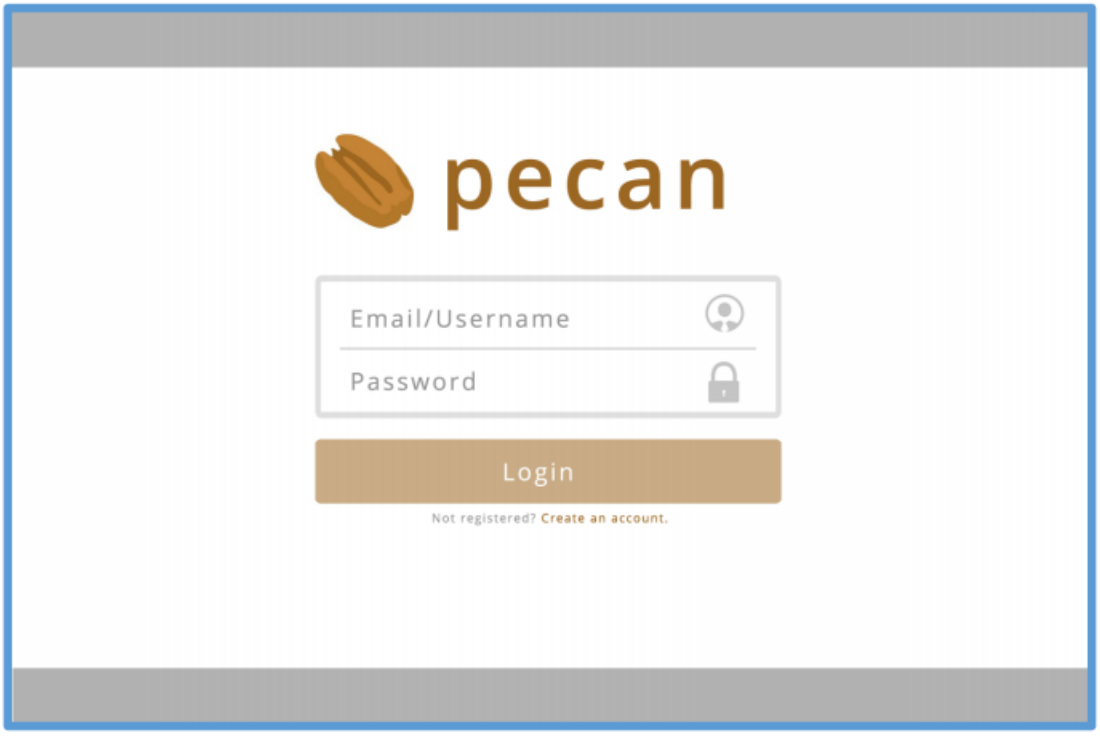


Diagram illustrates sidebar feature and ease of access, along with one-way/two-way direction for navigation flow.

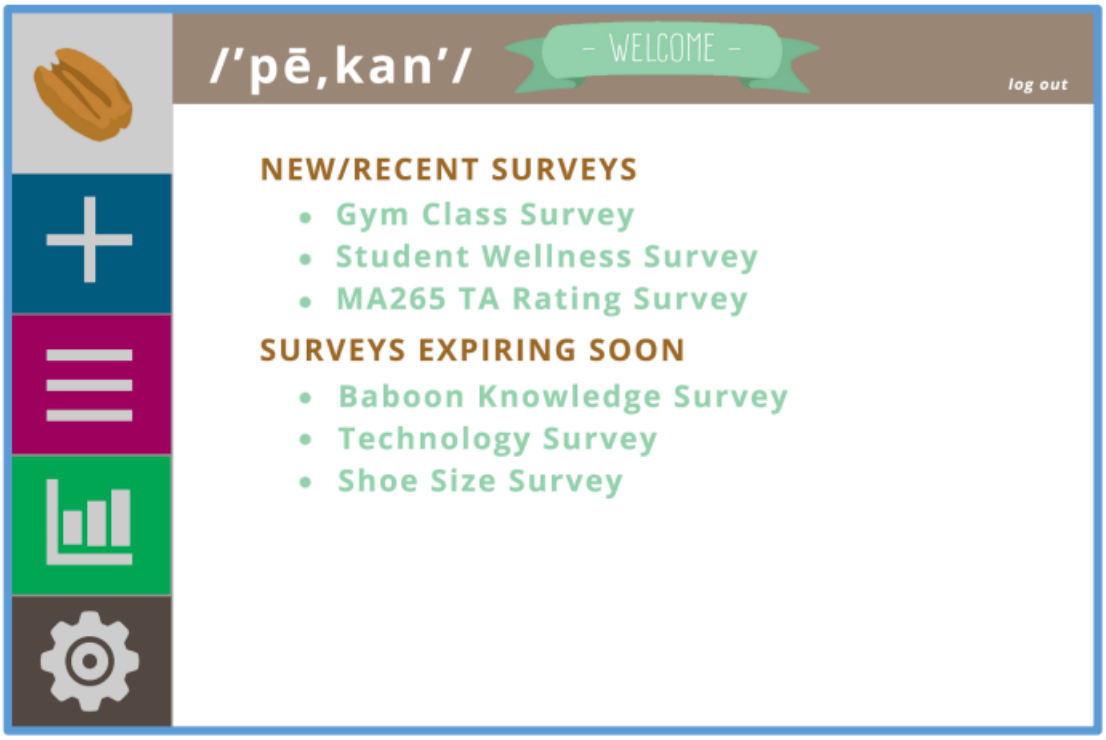
UI Mockups

Our UI mockups are based around the concept of Material Design and takes a minimalistic design approach. The most important part of our UI is responsiveness and ease of use in order to ensure user satisfaction. Our primary colors are light brown, dark brown and a light green that acts as an accent color. This schema can be seen throughout the fake text and layouts shown below. As for the main side menu the emphasis was on trying to make the individual buttons stand out from each other as they represent all of the major components of pecan. Using the Law of Past Experience we decided to choose icons that are familiar to most people and are often used in modern software design to mean certain things such as the add and menu button. This side bar stays consistent across the entire web app and only changes to highlight that a certain element/page was selected in to provide click feedback. *Note: the UI mockups are using a temporary typography set

User Authentication Screen on Website:

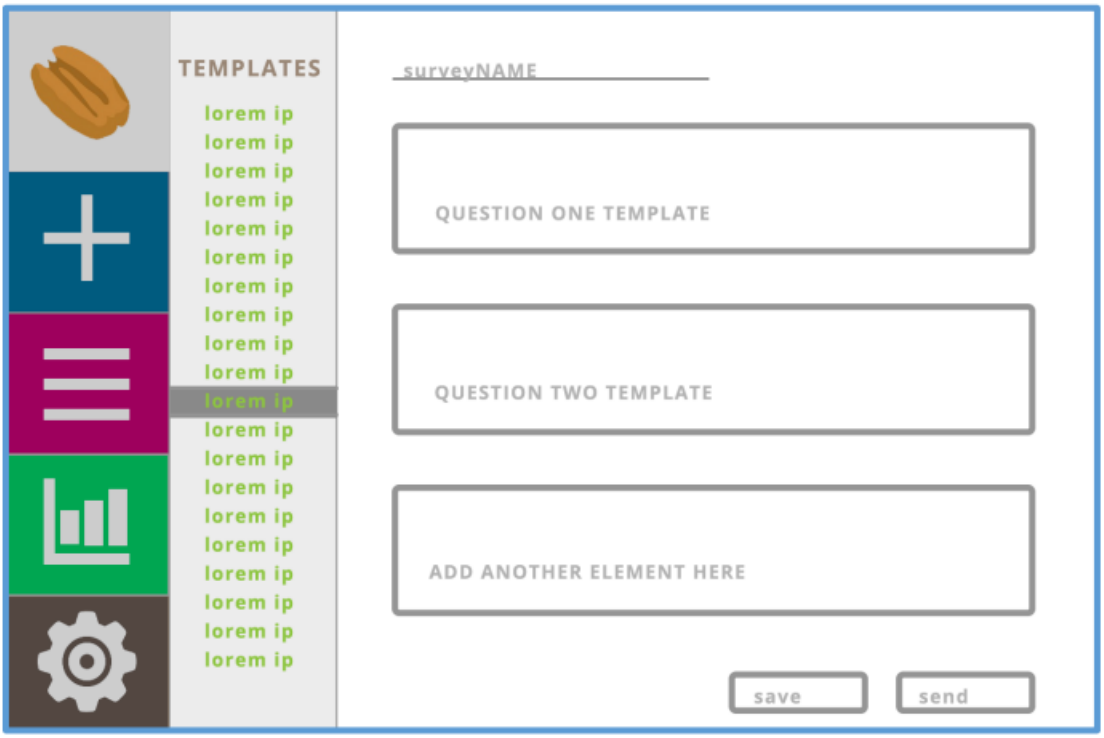


Homepage



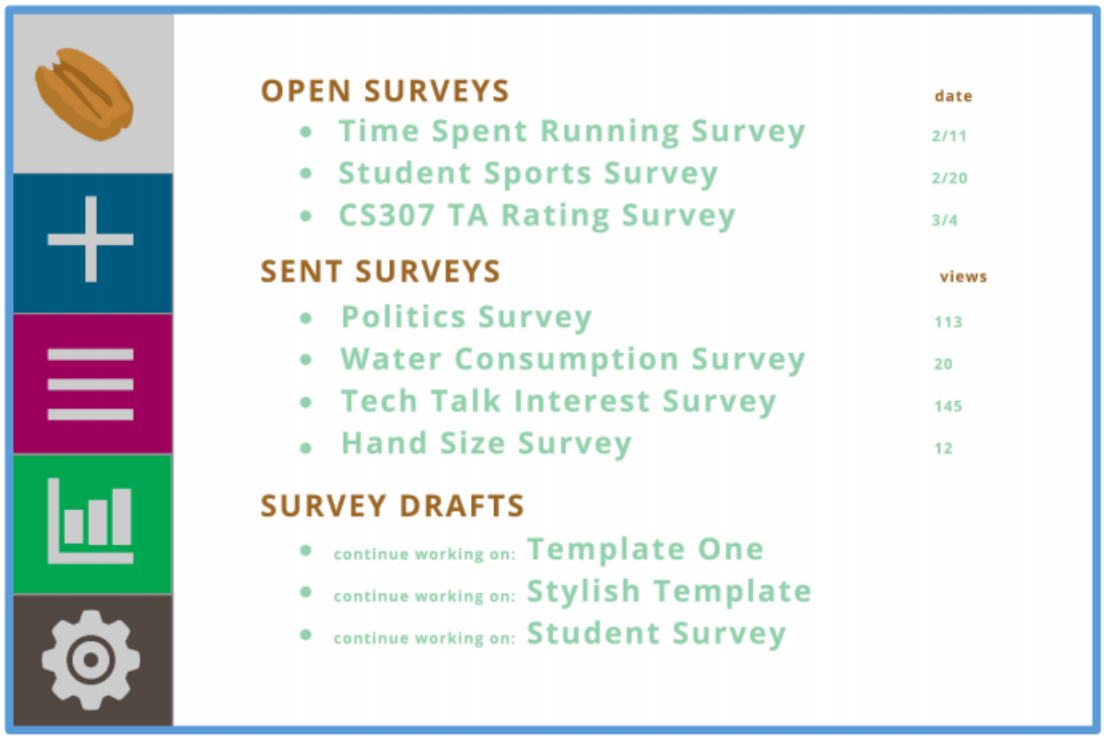
The purpose of the home page is to allow for users to quickly check up on any new and recent surveys that they may have received and also check up on surveys that are about to expire without any navigation. This view gives the user a brief overview on all of the surveys available and can be thought of as a more condensed and streamlined version of the HUB view shown in a diagram below.

Create New Survey Page



The New Survey Page is the page where a user would go to create a survey. All of the available templates are listed out on the left hand side and one of the options that is not shown in the mockup would be an option to create your own template and choose your color combination/background images. The 'add another element button' gives the user access to a list of all the elements that are available. The page also gives you the option to save your template as a draft or send it out if it is ready.

Survey Hub Page



The *Survey Hub* page as mentioned in the home page description is where a user would go in order to access and view all of his/her surveys in one centralized place along with information about it such as the number of views on it or the date that a certain survey expires. This view is also where any survey drafts that you previously saved are kept; in order to continue to work on any of them, all a user would have to do is click on the name of the draft to open it up.

Data Analysis Page

The Data Analysis Page provides a place for users to view their past surveys and also go back over the analysis and graphs for any prior survey that they created.

Settings Page

The Settings Page is where notifications, passwords and all other user info can be accessed and changed. The user can access all of their saved templates here, as well as edit or delete them. They can also control their email lists on this page, adding or removing emails from a particular list. The user can also edit their basic information, and will be provided with various settings to fine tune their experience while using our product.