Choosing Conference Papers

PairCon

Report 2

Client
Kilian Weinberger
kqw4@cornell.edu

Team
James Russo
Justina Chen
Mahak Garg
Mutahir Kazmi
Sahana Peters
Yao-Chuan Chang

Table of Contents

Requirements and Preliminary Design

Requirements

General Overview

Purpose and Scope

The purpose of PairCon is to offer a platform for people who attend research conferences to find the meetings and talks that they would be most interested in. PairCon aims to tailor to a user's preferences and maximize their attendance by saving a lot of time to search on each conference website and read all different papers. PairCon tries to achieve this by providing a recommendation system to members that suggests what papers at conferences they may benefit from based on their research interests. Members can manage their schedule and be notified of upcoming lectures, aiding them in a convenient way to navigate through a conferences and meetings. One the other hand, conference organizers can upload published papers, provide updates and notify attendees of changes to any schedules through one platform. PairCon's scope of usage extends to all research conference attendees and conference organizers.

Objectives and Goals

Milestone 1

The Goal of first milestone 1 to do the feasibility study to decide how we are going to build the system which best fits our client's requirement. We first define the scope of our project and what are client's necessary or good-to-have requirements. Based on these needs, we conduct following analysis to determine how we are going to build the system environment.

1) Define scope of software system
2) Preliminary Requirement Analysis
    a) Objectives
    b) Business Consideration
    c) User role and responsibilities
3) Technical Analysis
    a) Statement of functionality
    b) Security and user capabilities
    c) Non-functional requirements
    d) Other functions
4) Planning and Resource Analysis
    a) Development process
    b) Suggested deliverables
    c) Gantt chart & time line
5) Visibility plan
6) Risk analysis

Milestone 2

The Goals of milestone 2 was to modify the UI Mockups based on the feedback from the client as well as when new requirements were recognized. By this time, the backend is at a stage that the product is a MVP (minimum viable product). The functionalities and features in place are:

1) **PDF Scraping**: The user can give a URL with the list of its publication pdfs as input and there is a pdf scraper module that scrapes the pdfs and convert them to text and stores the text.
2) **Recommendation System**: A baseline paper similarity algorithm based on TF/IDF and cosine similarity is in place. This gives similarity scores between the conference papers and attendee papers which can be used for giving recommendations.
3) **Sign Up/Login System**: The login and sign up system for the attendees is in place.
4) **Landing Page:** The landing page was also created as a part of this milestone.

But the journey to achieving milestone 2 was not without its obstacles. While working on this milestone some of the requirements specified during the feasibility analysis changed and some new requirements were identified. We also had a learning curve to the technology(Ruby on Rails).

1) **Google Scholar API:** When we started, we decided with the client that we will use the Google scholar page of each attendee to get his publications. While working on creating a pdf scraper for the website, we identified that there is no public Google Scholar API available that can be used to get the attendee's publications. We also tried to scrape the Google Scholar page but this is against Google's ToS, and is illegal. So, the requirement changed during this milestone and we decided with the client the input type can be changed to a personal publication page, which has links to all the pdfs.
2) **Admin Page:** A new requirement that was identified during the milestone 2 phase was the need of another user role "admin". Earlier we decided that creating a conference can be open to anyone logged in the system, but then we realised that it would create too much chaos if anyone could create conferences. There was a need of admin role who can approve the request of someone wanting to be conference organizer. So, we talked to the client and agreed on proving some kind of admin control. We will be creating a admin dashboard which will help an "admin" see the organizer requests and approve them.
3) **Technology Learning Curve**: The technology chosen by us is Ruby on Rails(ROR), which a major part of the team has not worked with. Rails is a full stack framework with a lot of concepts to learn at the same time. So, there was a learning curve while using ROR to create the system. Also, some of the people were not aware of the git versioning system that we are using to keep track of changes, so this was another learning curve.

Current Systems

There is currently no system available which can give you recommendations about what lectures to attend in a conference. The current system is to look at the papers being presented and the schedule on a conference specific web page or app created by the conference organizers and decide which one to attend by reading the description/paper. This takes attendee's time and sometimes leads to them missing a lecture that might be of interest to them.

Proposed System Requirements

Overview

There are three key pieces of requirements in our project: Functional requirements focuses on building functions that connects to the database and recommendation system. Usability requirement consists of accessibility and mobile-friendly interface for academic users; Non-functional requirement aims to enhance security, performance and maintainability of the system.

Functional Requirements

The system should be able to perform the following functions :

User Interface

Conference Information	User must be able to see the information about a conference like conference name, organizer, location, URL and schedule.
Recommendation Table	If a user joined a conference they should be able to view their recommendations for that particular conference
Conference list	A user should be able to view the list of conference they have joined.
Buttons	Buttons for joining a conference, creating a conference, editing and saving a conference

Functionalities

Login/SignUp	Users should be able to sign up and once signed up can login to the system and land on the dashboard page.
Edit a User	User should be able to edit their information.
Create/Edit a Conference	A conference organizer should be able to create and edit a conference's information
Search a Conference	A user should be able to search a conference.
Get recommendations for a conference	An attendee should be able to get the recommendations for a conference based on similarity score between his publications and conference papers.
Admin organizer approval	The admin should be able to approve the request from a user to become an organizer.
Join a Conference	A user should be able to join a conference.
Become an organizer	An attendee should be able to request the admin to become an organizer.

Data

The database has various fields for the data we need to store which will be discussed in detail in later sections.

User	First name, last name, email, role(attendee, organizer, admin), publication url, password, gender, phone number
Conference	Name, location , url, organizer_id, paper_ids, start date, end date
Paper	Title, link, text, author
Similarity	Paper_id1, paper_id2, similarity core

Usability Requirements

Accessibility: The landing page and dashboard layout need to be compatible across typical user's web browsers. The typical user is an academic—professors, students, and researchers. We will focus on developing compatibility with Chrome first, and if time allows ensure compatibility with other browsers such as Safari, Firefox, and Internet Explorer. With the prevalence of users who access the Internet through smartphones and other mobile sites, we will need to make sure our system is mobile-compatible as well.

Salience and Ease of Use: In order to make the system intuitive to use, the dashboard layout needs to be classic and familiar. The primary use-case, involving creating a profile, registering for a conference, and generating recommendations needs to be straightforward too.

Feedback: Because the automated recommendation system requires time to scrape PDFs and compute distances between papers, there are delays in between when the user creates a profile or registers for a conference and receives recommendations. This delay needs to be conveyed to the user, so that they know the process is successfully running in the background. The user should also be notified when the process is complete.

Effective Recommendations: Recommendation quality is only one component of an effective recommendation system (of course, this needs to be intact). Additionally, users need an explanation of the recommendations in order to trust them. The system needs to display list of keywords on which the PDFs were matched, in order to justify the recommendation

Non-functional Requirements

Security: The system aims to maintain different user roles to ensure confidentiality of the conference organizers and the attendees. An attendee cannot modify conference details and a conference organizer cannot modify an attendee's profile. An admin approves a request from an organization to become a conference organizer. Passwords that are stored in the database are encrypted.

Scalability: The system has a goal of reaching at least 3000 conference users. To enable this expansion, we intend to host the server on elastic cloud services like Heroku.

Maintainability: To ensure that the system is maintainable and as a project continues to be improved, we have enforced stringent code quality rules. We are following an MVC architecture and using standard practice of Ruby on Rails to keep our software readable.

Responsive Compatibility: The system should work on on both handheld and desktop devices. We have chosen responsive themes and have been simultaneously testing and

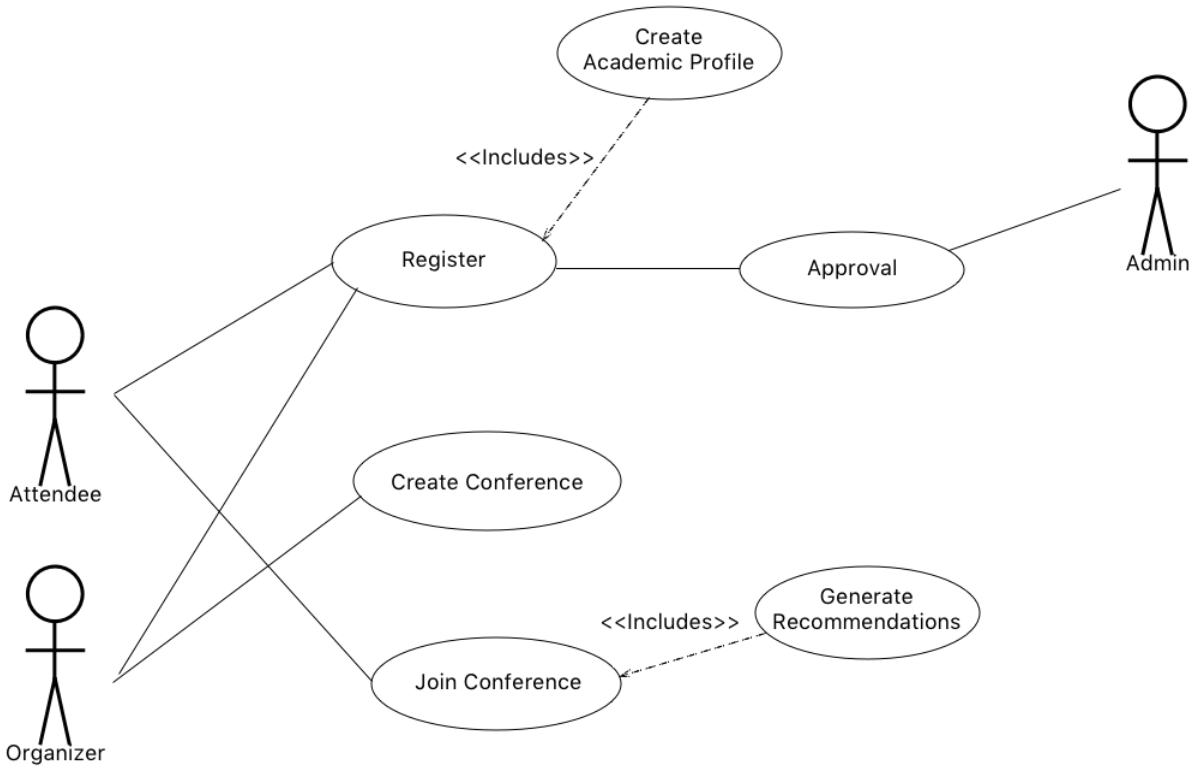
creating interfaces that would provide seamless usage on all devices with different screen resolution and operating systems like Windows and MacOS.

Backups: Currently our system uses Git as a software version control system and a repository for our code and data. Once our application is deployed, we would also enact a process to ensure that regular backups of the cloud instance would be taken.

Requirement Models

Use Case Diagrams and Specifications

Use Case Diagram



Scenario 1: Conference Attendee Signs Up and Creates Profile

Individual: Jane Doe, a post-doc studying UX and Human-Computer Interaction at University of Michigan. She's using PairCon for the first time.

Equipment: Jane's personal laptop--a MacBook Pro running Chrome

Scenario

1. JD turns on laptop, opens Google Chrome, and navigates to the PairCon URL
2. JD clicks the "sign-up" link, and inputs her information into the form: first and last name, email address, and password. She clicks the "Sign up" button.
3. The PairCon Dashboard launches. There is a banner/notification that notifies her to complete her user profile. Simultaneously, JD receives an email notification that her profile has been successfully created.
4. JD clicks on the chevron by the default user avatar. She clicks on "User Settings"
5. A form opens. JD's first and last name, and email address are displayed. Jane clicks on the default avatar, and a modal pops up. She chooses to upload a profile picture from her computer.
6. There is also a field to submit her academic publication profile website URL, and also a button to open another form to fill her publications manually. JD opts submit her academic profile URL. She pastes the link and clicks "submit." A notification window pops up, informing her that the system is downloading her publications to complete her user profile.
7. With her profile complete, JD clicks on the "Home" link on the left menu bar. The "Home" dashboard reloads.
8. JD sees a notification that tells her that her academic profile is still being generated.
9. JD logs out.
10. JD receives an email notifying her when her profile generation is complete.

Scenario 2: Conference Attendee Registers for a Conference

Individual: Jane Doe, a post-doc studying UX and Human-Computer Interaction at University of Michigan. She's created her profile and registering for the upcoming CHI conference. This is her first time using PairCon.

Equipment: Jane's personal laptop--a MacBook Pro running Chrome

Scenario

1. JD turns on laptop, opens Google Chrome, and navigates to the PairCon URL
2. On the PairCon landing page, JD enters her email and password.
3. The PairCon Dashboard launches. There is a banner/notification that notifies her that her user profile is complete.

4. JD clicks the “Join a Conference” button. A search bar opens up.
5. She types “CHI” into the search bar and presses enter. The results load, and she finds the matching conference. She clicks on the icon
6. The icon brings her to the internal conference page. JD clicks the “plus” icon to add the conference to her conference list, which shows up on her left-hand menu bar under “Conferences” when she refreshes the page.
7. JD then presses the “Generate Recommendations” button. She sees a banner notifying her that the system is running, and her recommendation list should be up shortly.
8. JD logs out of PairCon.
9. JD receives and email notifying her that her recommendation list for CHI has been processed.
10. JD logs back into PairCon and clicks on the “CHI 2018” link in the “Conference” section of her menu. She see’s her list of generated recommendations with the name of the paper, the authors, and the paper keywords.
11. JD logs out of PairCon.

Scenario 3: Conference Organizer Signs Up and Creates Profile

Individual: Jane Doe, an Event Manage for the International Machine Learning Society. She’s using PairCon for the first time.

Equipment: Jane’s personal laptop--a MacBook Pro running Chrome

Scenario

1. JD turns on laptop, opens Google Chrome, and navigates to the PairCon URL
2. JD clicks the “sign-up” link, and inputs her information into the form: first and last name, email address, and password. She clicks the “Sign up” button.
3. The PairCon Dashboard launches. There is a banner/notification that notifies her to complete her user profile. Simultaneously, JD receives an email notification that her profile has been successfully created.
4. JD then navigates to the chevron where she clicks the “Become an Organizer” link
5. A form opens where JD can input the name of the organization and the conference email id that would be used. She then can click submit
6. On submission, a popup is displayed that her request to become an organizer is in process and she would receive an email confirmation shortly.
7. If she is confirmed as an organizer of the conference, an email with a link will be sent to her.
8. Once JD clicks on the link, she can view the conference she is an organizer of, on her dashboard along with any other conference she may attend.

Scenario 4: Conference Organizer Creates a Conference

Individual: Jane Doe, an Event Manager for the International Machine Learning Society.
She is creating the conference supported by them – the International Conference for Machine Learning (ICML)

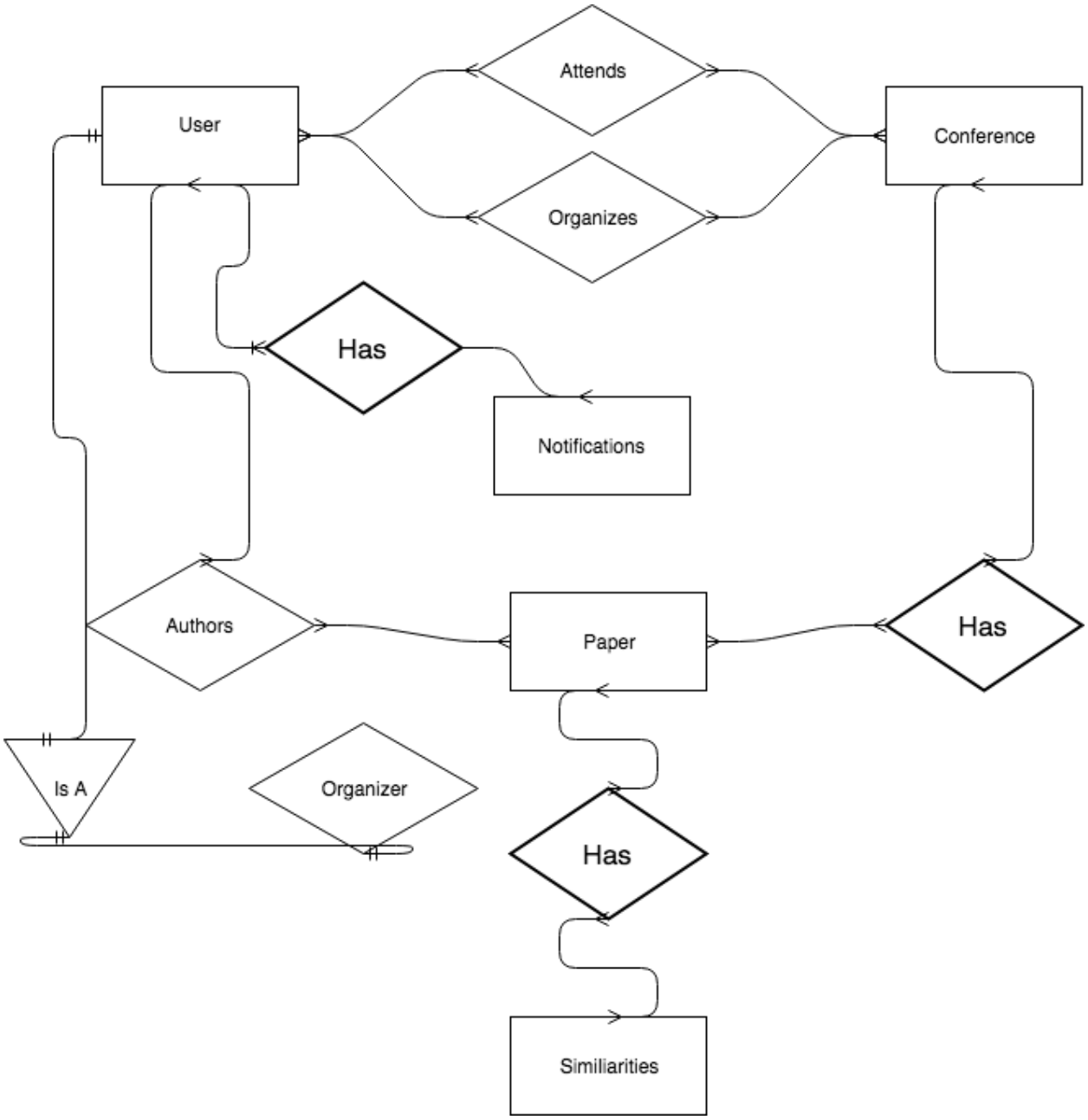
Equipment: Jane's personal laptop--a MacBook Pro running Chrome

Scenario

1. JD turns on laptop, opens Google Chrome, and navigates to the PairCon URL
2. On the PairCon landing page, JD enters her email and password.
3. The PairCon Dashboard launches. There is a banner/notification that notifies her that her user profile is complete.
4. As she is also registered as a conference organizer, she can navigate to the conference organizer dashboard.
5. JD clicks the "Add a conference" button.
6. A form opens up where JD can then enter the name of the conference, the date, location, the research field and the number of papers that might be presented. There is also an option for JD to upload all the possible papers that might be showcased at the conference through a csv file.
7. If JD has not created a csv file previously, she may download the template of the csv file and then input the papers and their speakers accordingly.
8. Once she clicks the submit, the conference will then be generated.
9. The conference will then be displayed on her dashboard as an upcoming event.
10. JD logs out of PairCon.

Data Models: ER Diagram

Simplified ER Diagram



The main entity of our ER diagram is the **User** entity. This entity will keep track of users of our website and have attributes such as a *user_id*, *password*, *user personal information*, and other user specific information to keep track of users. The **User** entity will have the relationships **attends** and **organizes** to the entity **Conference**. The **Conference** entity will keep track of our

academic conferences and have the attributes of *id*, *start date* and *end date*, *location*, *url*, and any other conference specific information. **Users** will be able to “attend” conferences, indicating that they are interested in going to the conference. Once a **User** has been approved by an admin to be an organizer they will be able to **organize** conferences which entails creating and editing conferences for which they are the organizer. Finally **Users** will be able to become **Organizers** in order to create and edit **Conferences.** This entity will needed to be approved by an admin before they can create **Conferences** but will allow for more functionality once approved.

A **User** also has **Notifications**. This allows us to notify users of different events such as a new conference being created they may be interested in, whether they have been approved for creating conferences, etc. **Notifications** will have the attributes of an id, message, date, and whether or not it has been read.

The other main entity of our database is the **Paper** entity. This entity has the attributes of a *paper id*, the *title* of the paper, *pdf link* to the paper, the *hash* of the paper so we can tell if it’s already been added quickly, and the path to the text version of the paper stored on our server. A **User** can author many **Papers** and **Papers** can have many authors. **Conferences** can also have many **Papers** presented at them and many **Papers** can be presented at multiple conferences. Finally **Papers** have many **Similarities** to other **Papers**. A **Similarity** has the attributes of two *paper ids*, a *similarity score*, and a *hash* unique to those two papers. This setup allows us not to have to recompute similarities for papers we have already computed the similarities for.

The actual implementations of these entities and relationships as tables will be further detailed in a later section discussing our database implementation.

Preliminary Design

User Interface

The landing page and the dashboard are the two main components of the system interface. Within the dashboard, there are several different subsections—the header bar, the left-hand menu bar, and the dynamic content at the center.

Landing Page

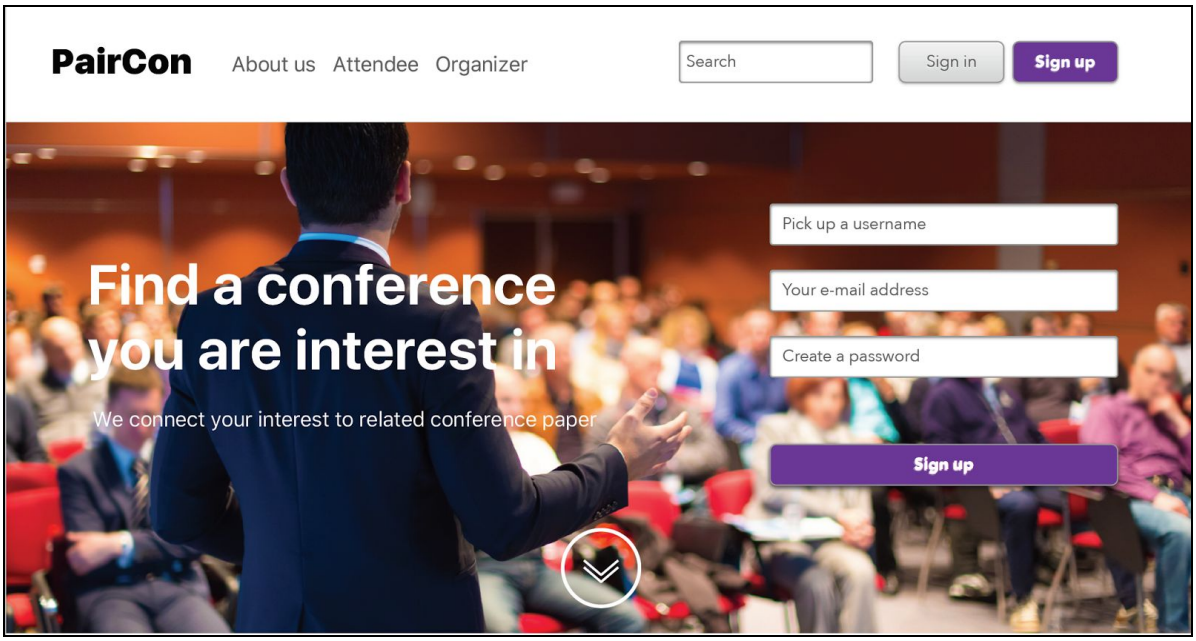
In our landing page, we adopted a single page design which allows users to easily scroll down the page and find all the information they need on one single page. We also took into account responsiveness and mobile-friendliness as two key features when designing the landing page.

Since a large number of our users are academics who are constantly in meetings and looking at information on their phones, we thought it was important to ensure proper responsive and mobile friendly design.

The landing page is comprised of three main sections; the header bar, main content, and a footer bar as follow:

Header Bar

The header bar is to help users to navigate through the website and get the most relevant content out of it. In our design, we tried to keep the navigation bar simple in order to prevent our users from being overwhelmed by too much irrelevant information. We also provide search functionality in the header which allows users to quickly search for existing conferences.

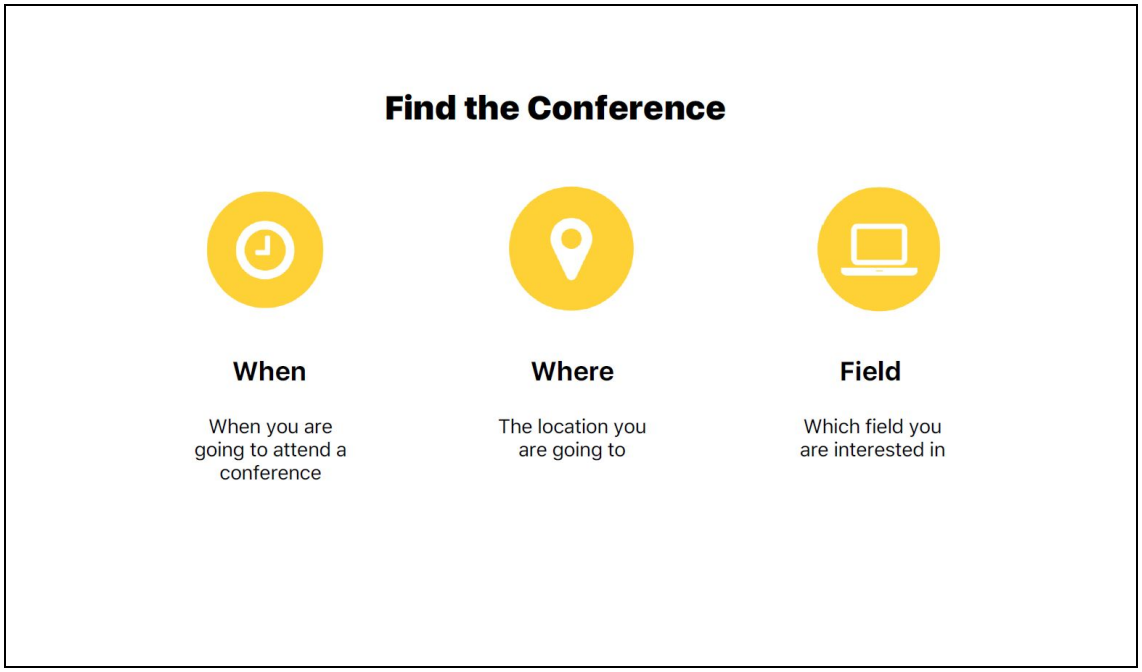


Generally there will be two types of users “conference attendees” and “conference organizers”, we designed the “Sign in” and “Sign up” link in our header which allows users to quickly log into their personal page. For the first time user, there is simple and precise text to explain what PairCon does and a quick sign up section next to it.

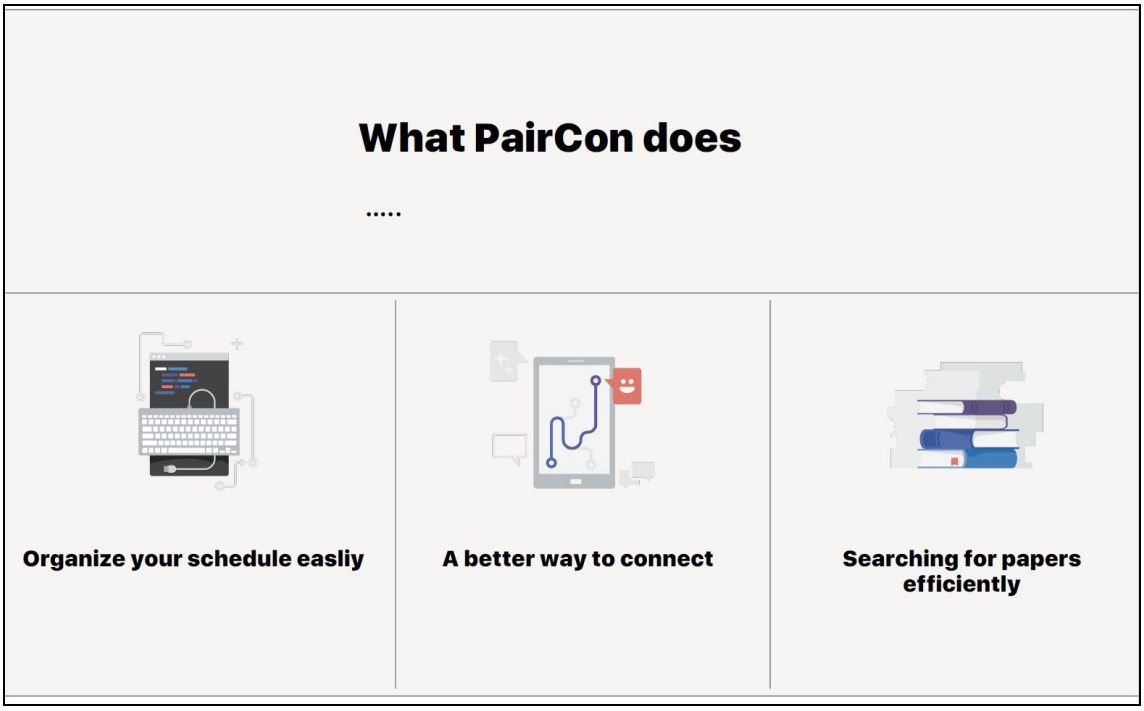
Main Content

The design idea of behind the main content section is to let our users immediately immerse themselves in the opening lines of a story when they scroll down the page. Simplicity is the key behind the landing page, allowing users to easily get to necessary sections. We attempt to tell a compelling store by showing them why they should choose PairCon to help them find a better conference to attend. The following is our story line for the main content:

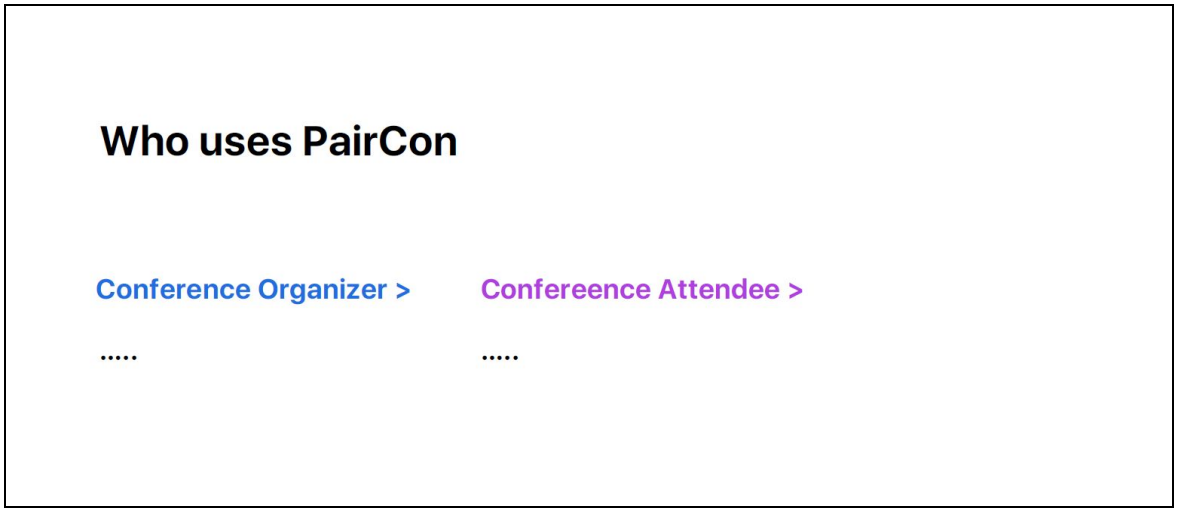
Section 1: How you can take an action



Section 2: What features PairCon has



Section 3: Who the users are and the information board



Footer Bar

The idea of designing a footer is to help visitors by adding information and providing help service at the bottom of the web pages. So our footer bar consists of three parts: the legal section on the left side, with contact and help sections on the right side. In the middle, users can see our logo for PairCon as follows:.

We placed the legal section on the left side of the footer because it is important for our users and we want it to be clearly available. Also when a user signs up for our website they will have to sign our Terms of Service Agreement. We do this because we would like to make sure every participant's privacy is guaranteed and users of our website will follow terms and condition to use our pairing algorithm.

On the right side, we will provide links for users to contact us for further information and customer service when our users need help. Each link will create a partial view in separate page when clicked on.

Dashboard

The dashboard is comprised of three main sections; the header bar, the left-hand menu bar, and a center section with dynamic content. The header bar and left-hand menu bar are mostly static for the user.

The system dashboard has the same format and theme for all three users (Administrator, Organizer, and Attendee), with slight differences to account for different functionalities and needs that each user requires. Additionally, the color scheme for the Administrator dashboard is darker, because the Administrator has significantly different responsibilities and privileges. The Administrator is deals with other users' privileges and roles, rather than working directly with conferences.

Header Bar

The header bar has a search bar for users to search for existing conferences. It also has the user profile picture (or default avatar if the user doesn't upload a profile picture). Next to the user profile picture is a chevron, which opens a settings menu on-click. This is the same for the three users.

Below the header bar is a banner with any important notifications, and other functional buttons specific to the type of user. The Attendee only sees a button to "Join a Conference," while the Organizer also sees a button to "Create a Conference." The Administrator banner has "Manage Organizers" and "Manage Conferences" buttons instead.

Left Menu Bar

The left menu bar allows the user to interact with the main functionality of the system. Under the "Home" section, the user sees a menu link for their main dashboard page.

The Administrator sees two dashboard options—the main dashboard and the administrative dashboard. The main dashboard allows the Administrator to interact with conferences as an Organizer or Attendee. The administrative dashboard allows the Administrator to adopt their administrative role over other users. This set-up allows the Administrator to both use the PairCon system and handle other users, within one login.

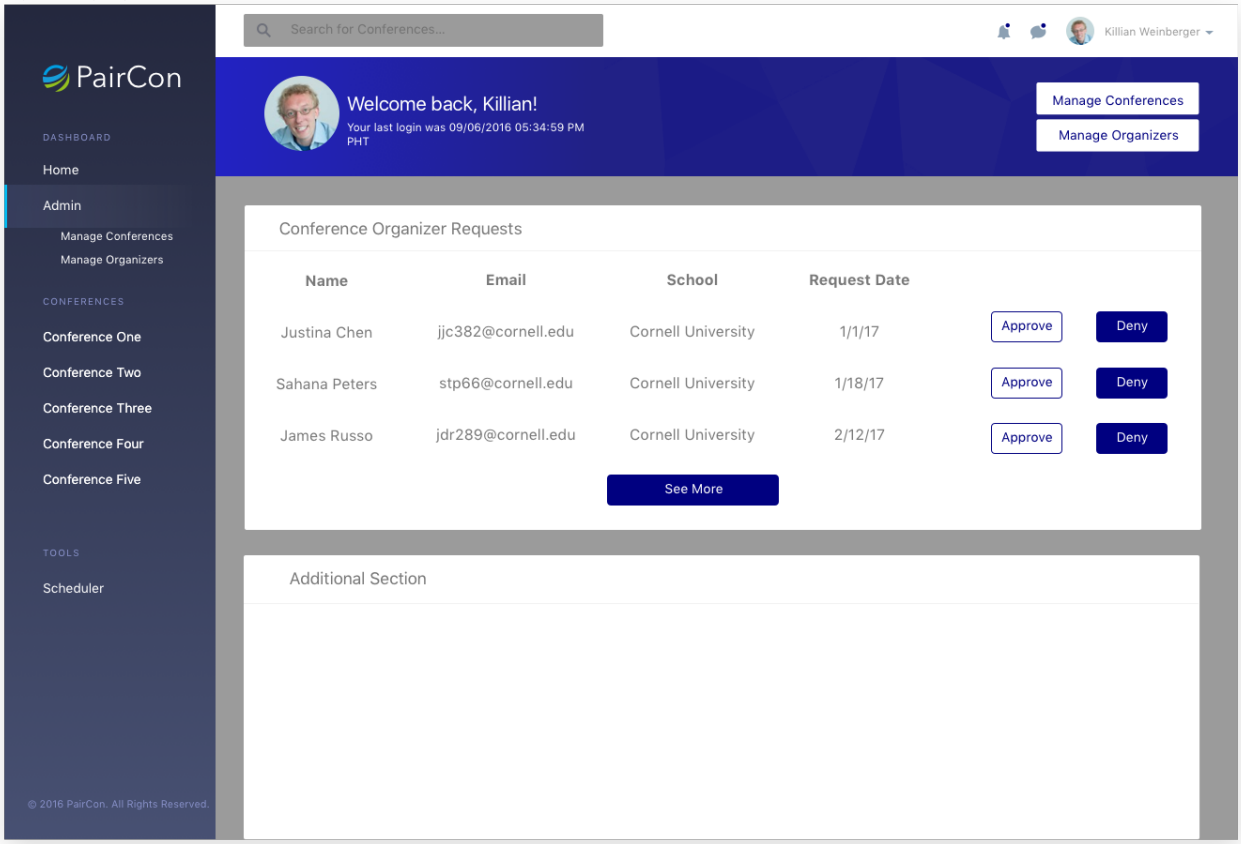
In the second part of the left menu bar, under "Conferences," the user sees the list of conferences that they have either organized or are attending. They can click on each conference in the list to load more details on the conference in the dynamic content section.

Main Content
This section is dynamic, changing constantly depending on what “page” the user selects.

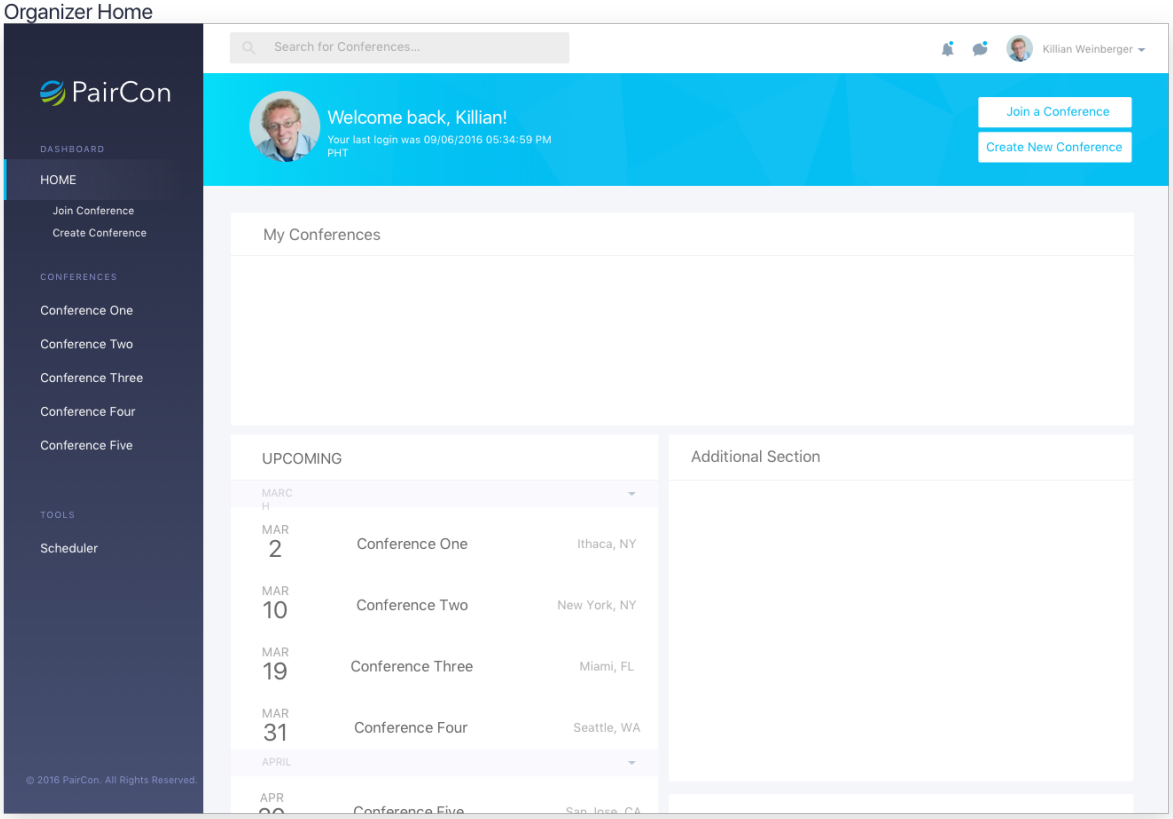
Dashboard: The home dashboard contains an overview of the user’s conference information. The top section includes the list of conferences that the user has joined on PairCon, with concise conference details (conference dates, main sponsor, location). Another section includes a calendar with upcoming conferences, which the user can use to discover upcoming conferences that they might be interested in attending.

The administrative dashboard has a similar layout to the home dashboard, but allows the Administrator to handle Organizer requests. They see a list of Attendees who have requested organizer privileges, along with key information about these Attendees (Organization, email, and request date). The Administrator can verify each request, and approve or deny each with corresponding buttons for each request.

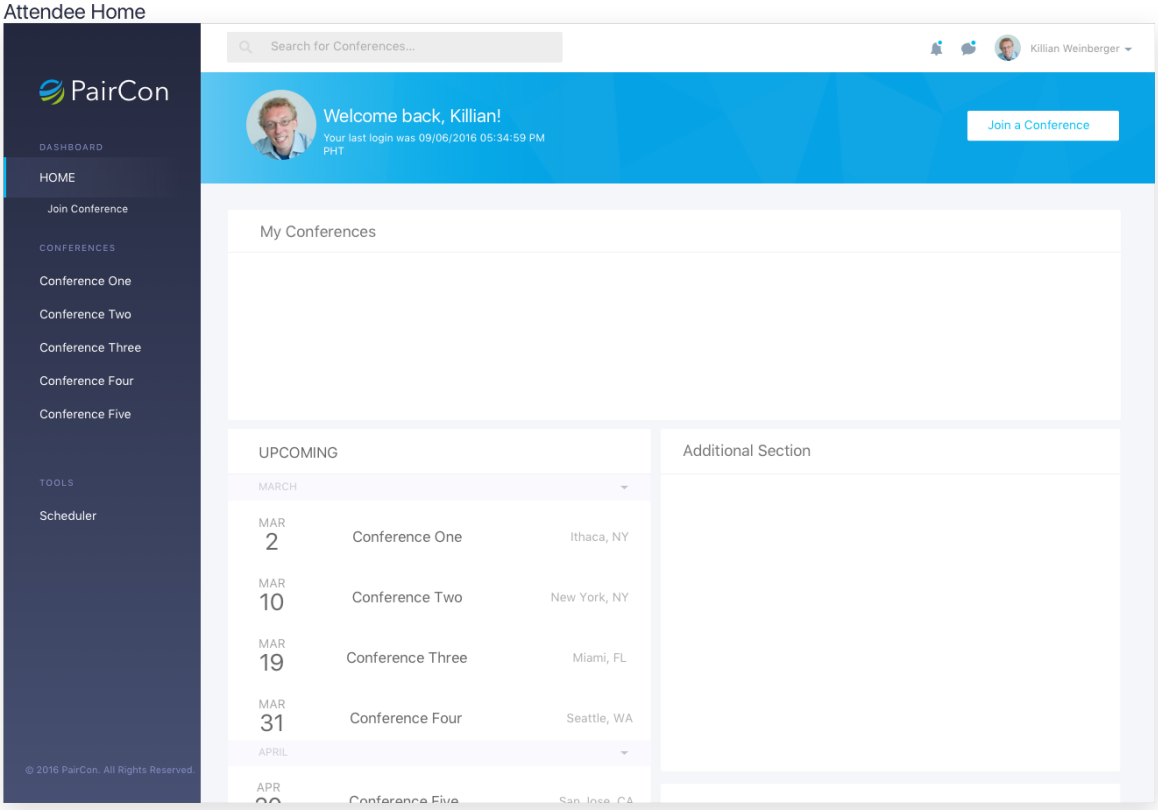
Admin Home



Organizer Home



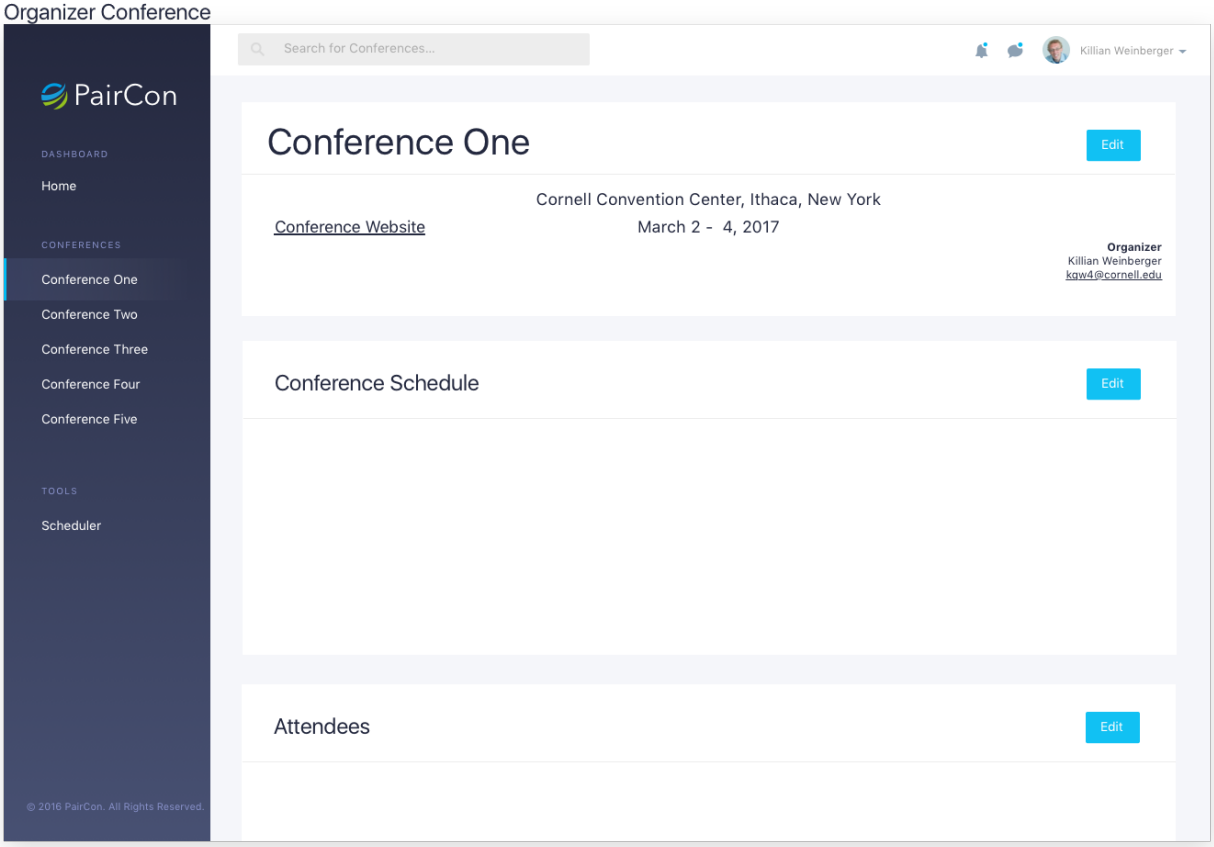
Attendee Home



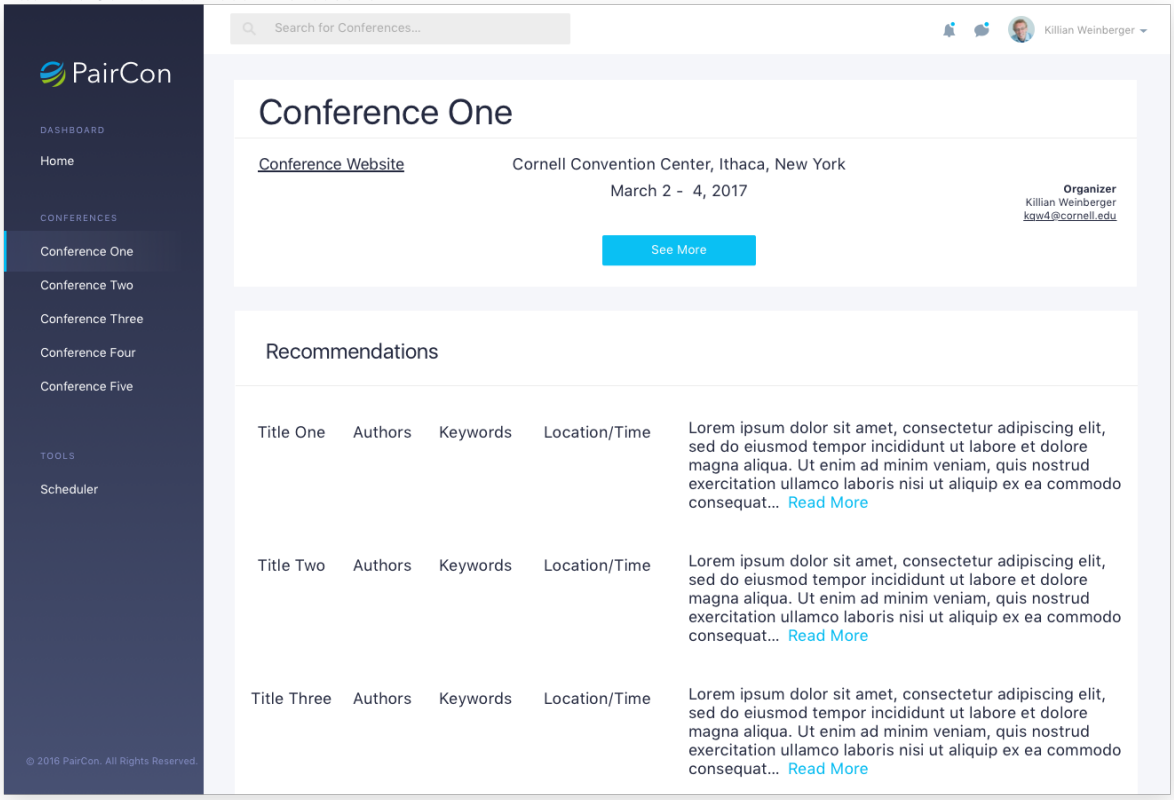
Conference Page: When an Attendee clicks on a conference that they have joined/created, they see a section with key conference information (conference website link, name, location, date, and contact), and their list of recommendations underneath if these have already be generated. If the recommendations have not been generated, the Attendee will see a button that they can click to generate recommendations. The conference information section can be expanded with a "See More" button, to display a description of the conference, a PDF download of the conference schedule, and a PDF of the conference map.

The Organizer sees similar content on the conference pages as the attendee, but they have options to edit the conference information section and can see the list of attendees and the list of top recommendations.

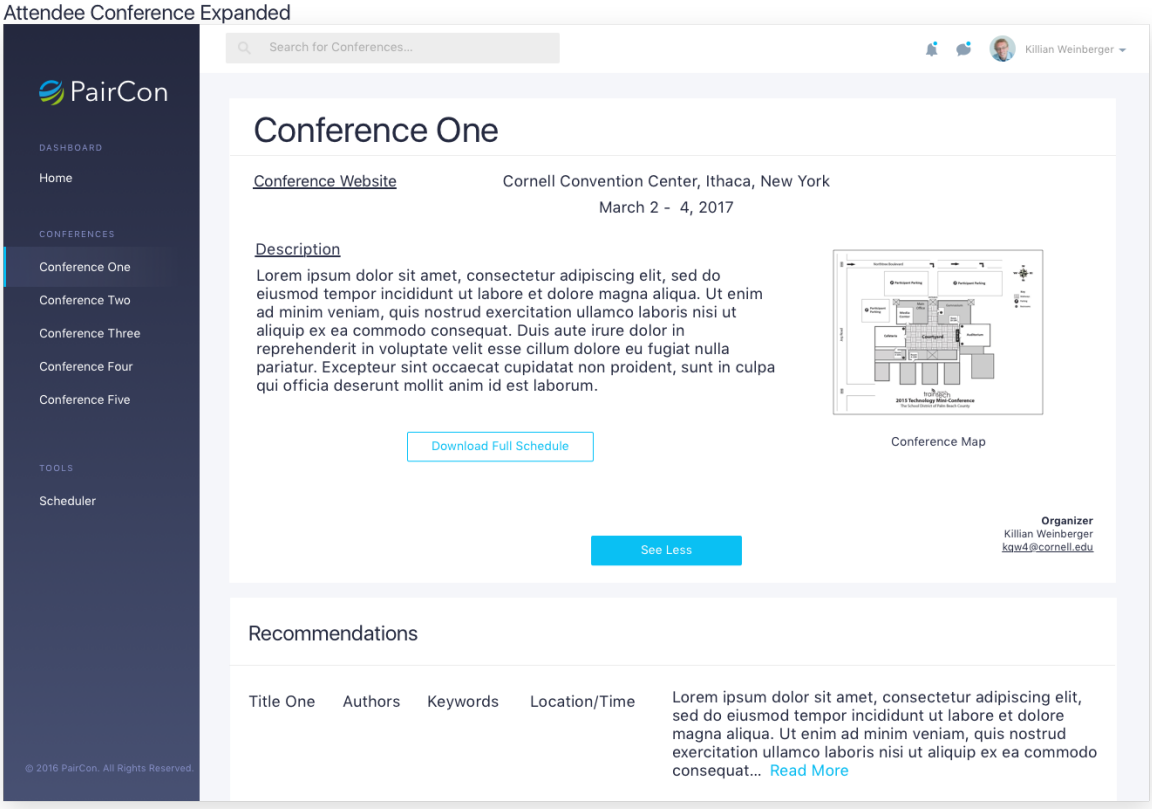
Organizer Conference



Attendee Conference Recommendations



Attendee Conference Expanded

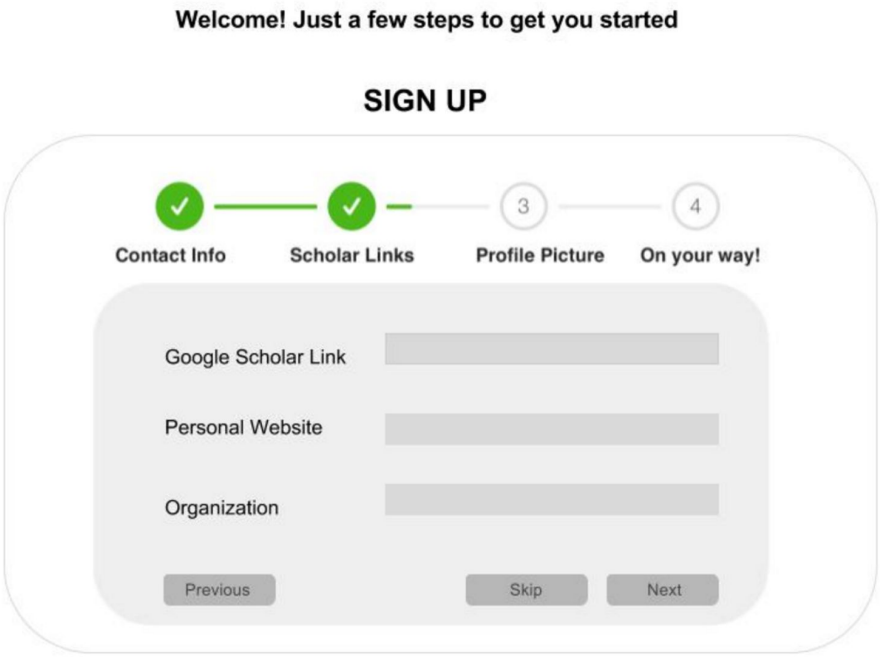


Login System

If we put the signup process in one page it could be long and painful, and users may have to fill out a lot of information in one single page. Because of this we decided to divide the sign-up process into several steps:

- Contact information
- Scholar links
- Profile picture
- Others

A progress bar along the top section informs the user how much progress they have made and how much further they must go to complete the sign-up process.



System Design

Choice of Technology

The whole system is divided into four parts and thus these different parts of the overall system played an important role in determining different technologies. Additionally, some of the members had experience in one technology over the others and this was another factor that was used to determine which technology to use.

The team is using the following technologies:

Website:

- Ruby on Rails 5
- Javascript / JQuery
- HTML5
- CSS 3

Database:

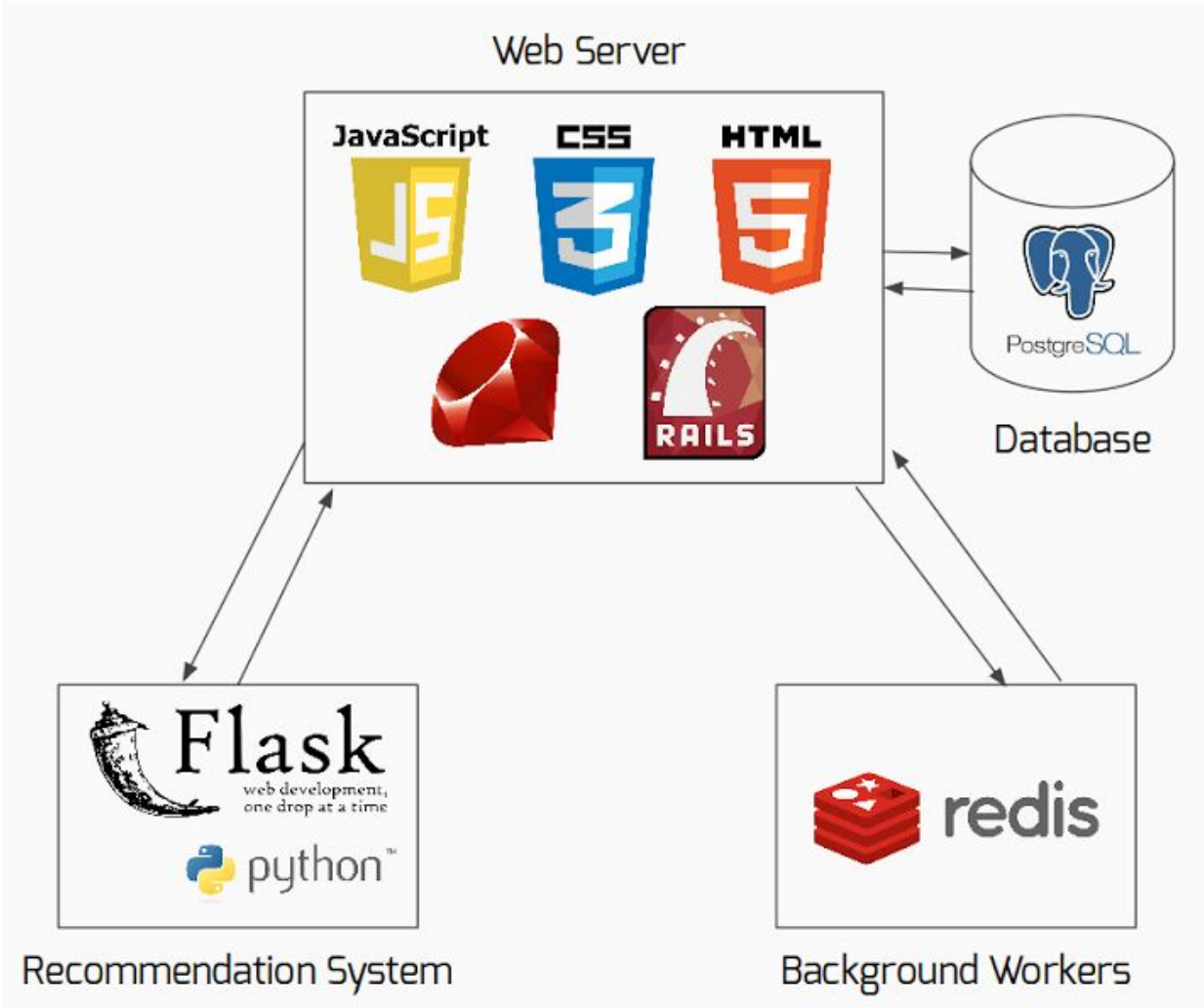
- Postgresql

Recommendation System:

- Flask
- Python

Background Processors:

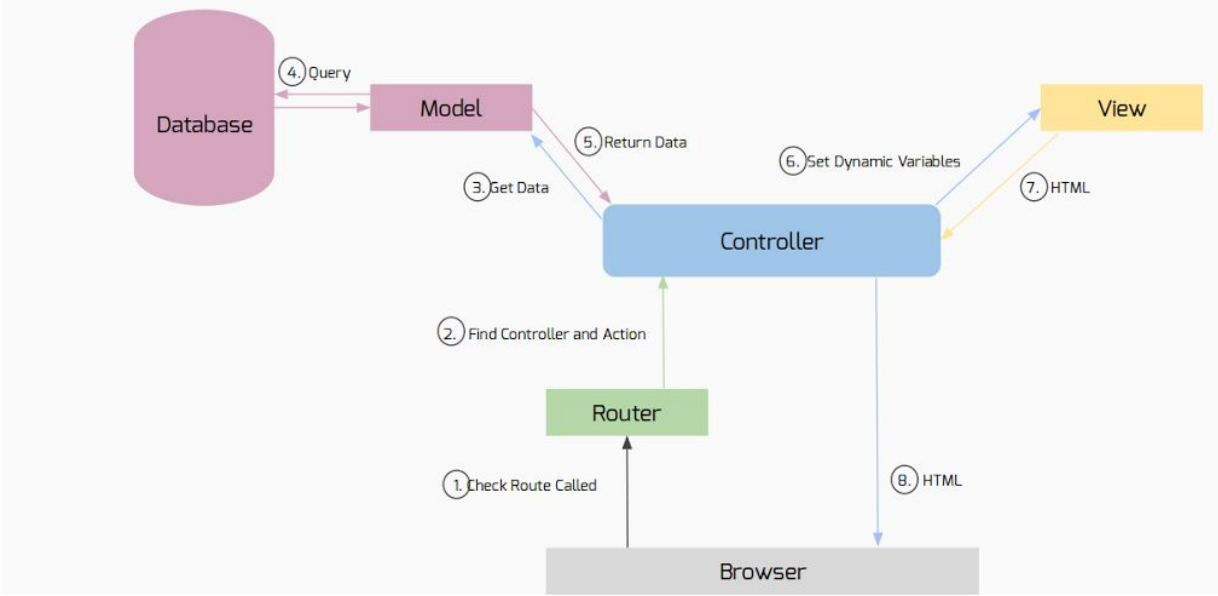
- Redis / Sidekiq



The web server provides the actual interface that organizers and other people interested in the system interact with. The web server communicates with the database layer which is running on PostgreSQL. Upon requesting for recommendation, the system interacts with another web

server running in Flask. This system is used to provide results in the form of JSON which is then rendered by the web server. The architecture of the recommendation system was suggested by the client due to the ease of implementing various machine learning techniques and vast amount of APIs through it. The web server then uses Redis with Sidekiq to run background jobs such as scraping web pages etc.

Model View Controller



Ruby on Rails follows a MVC architecture which is described below:

The Model

- Contains data for the application
- Contains state of the application
- Contains all business logic
- Provides queried results in the form of objects via Object Relational Mapping

The View

- Generates the user interface which presents data to the user
- Passive, i.e. doesn't do any processing
- Views work is done once the data is displayed to the user.
- Many views can access the same model for different reasons

The Controller

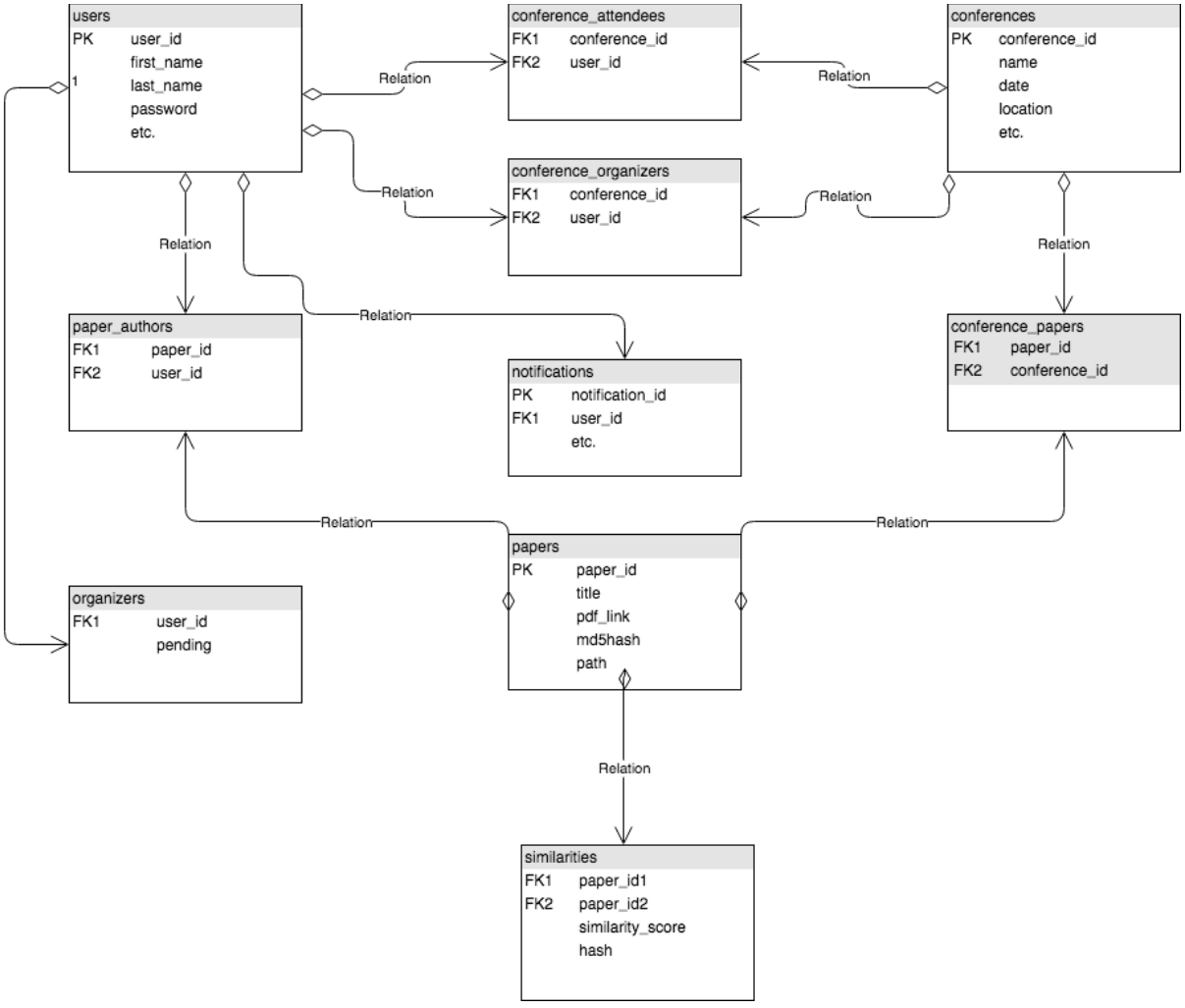
- Receive events from the outside world
- Interact with the model
- Displays the appropriate view to the user

Database Design

The first figure below shows our proposed database schema based on our previous ER diagram and second figure below shows the proposed table definitions. We have broken each of our entities and relations into individual tables in order to keep the information normalized and relational. We did not make any big changes from the ER diagram to the actual database design except adding in necessary attributes to our database tables. We have a larger number of attributes for our user table than originally intended, because Rails has robust built-in functionality that utilizes these attributes. Consequently, we created these attributes to use along with Rails. The rest of the tables have much fewer attributes as demonstrated and described in the ER diagram section.

This may not be the final table designs, as we are constantly adding functionality to our site and adding attributes and tables.

Proposed Database Schema



Proposed Database Tables