Turnip

Team 2 - Project Backlog

Abigail Alderson, Cole Johnson, Keegan Irby, Kevin Cardona, Kyle Pollina

Problem Statement

Despite the presence of various event platforms currently available, event hosts and guests are often left without a simple platform to privately share content central to a specific occasion. With Turnip, hosts will be able to send invitations, view accepted/unaccepted invites, and create an event page that clearly displays all the details. Most importantly, guests and hosts will be able to easily upload exclusive media that Turnip curates into unique event memory albums.

Background Information

Audience

Millennials are bombarded with new technology every day, which makes them rather adaptive when it comes to new software. This, combined with a calendar full of parties, weddings, and other outings creates the perfect audience for Turnip. A simple platform with familiar assistive applications (like Facebook) can be easily picked up by this young crowd, which allows for easy onboarding and a greater likelihood of consistent use.

Similar Platforms

Several social media platforms today provide a variety of experiences, and while these applications aim to give people the power to connect, they all focus on different types of interaction. For example, Facebook is designed for long term interaction, Snapchat only allows temporary content for 24 hours, and Instagram is fashioned for following people, not events.

Limitations

Many of the existing applications mentioned present a variety of features to the user; but often times, a user still has accounts on at least two of these platforms. By providing a

service to users that combines the streaming capabilities of Facebook, simplicity of use on Snapchat, and permanence of content on Instagram, Turnip will create a unique platform that users can easily adopt to lessen the desire to bounce between multiple applications.

Functional Requirements:

1) As a user, I would like to...
    a) create an account on Turnip.
    b) login to my Turnip account.
    c) access my account on different devices.
    d) change my account information including my name, password, and email.
    e) upload content to a current event stream.
    f) comment and interact with other users on event posts.
    g) export content from Turnip to Facebook, Instagram or other platforms.
    h) view content "memory albums" of past events I've been involved with.
2) As a host, I would like to...
    a) invite my Facebook friends to my event.
    b) display clear details about my event.
    c) edit content on my event's stream.
    d) customize my event page to suit my event's theme.
    e) allow users to add music to my Spotify playlist.
    f) set up my payment solution (Venmo, Square) for event entrance.
    g) view guest account details.
3) As a guest, I would like to...
    a) RSVP to an event.
    b) view event details.
    c) tag other attendees in content.
    d) view detailed directions to the event based on where I am located.
    e) pay host for event entrance.
    f) view host account details.

Non Functional Requirements

Architecture

Our team plans to develop Turnip with a separate frontend and backend, which offers several advantages. It will allow our team to easily and equally split the tasks we need to complete and will also allow for easier deployments when pushing to production. Additionally, if we decide to make the app suitable for a platform other than web, having a pre-built API will make that task much easier. For the backend, we will develop a RESTful API with Node.js, employ MongoDB for our data storage, and use Express.js for routing. The frontend will be developed with AngularJS and will make requests to our API.

Usability

Usability and little to no onboarding is very important for Turnip. The minimization of hindrances allows users to quickly and easily upload content and return back to the event environment. After all, it should be more important for people to experience the event than to spend time on their phone. That being said, extensive user testing and research will be crucial to design the optimal UI/UX for the application.

Security

We want to make sure that each event page is only accessible by people who have been invited to attend the event. Users will be given specific permissions depending on what their role for the event is. If they are a host, they will have access to administrative powers like removing posts and adding/removing guests from an event. To achieve this, all API requests will need to be authenticated before the requests can be executed.

Integrations

To make Turnip more feature rich, we will take advantage of a few API's.. We will use:

- Spotify's API to allow guests to add songs to an event-specific playlist.
- Venmo's API to let guests pay the host (should the host deem necessary).
- Google Maps API to provide guests with directions to the event.