Software INC.

**Software Design Document**

**Play My Song**

Mike Schmitt Rodny Joseph Karl Kraus Haseeb Saadut

**Version 1.0**

**September 30, 2016**

1. **Introduction**

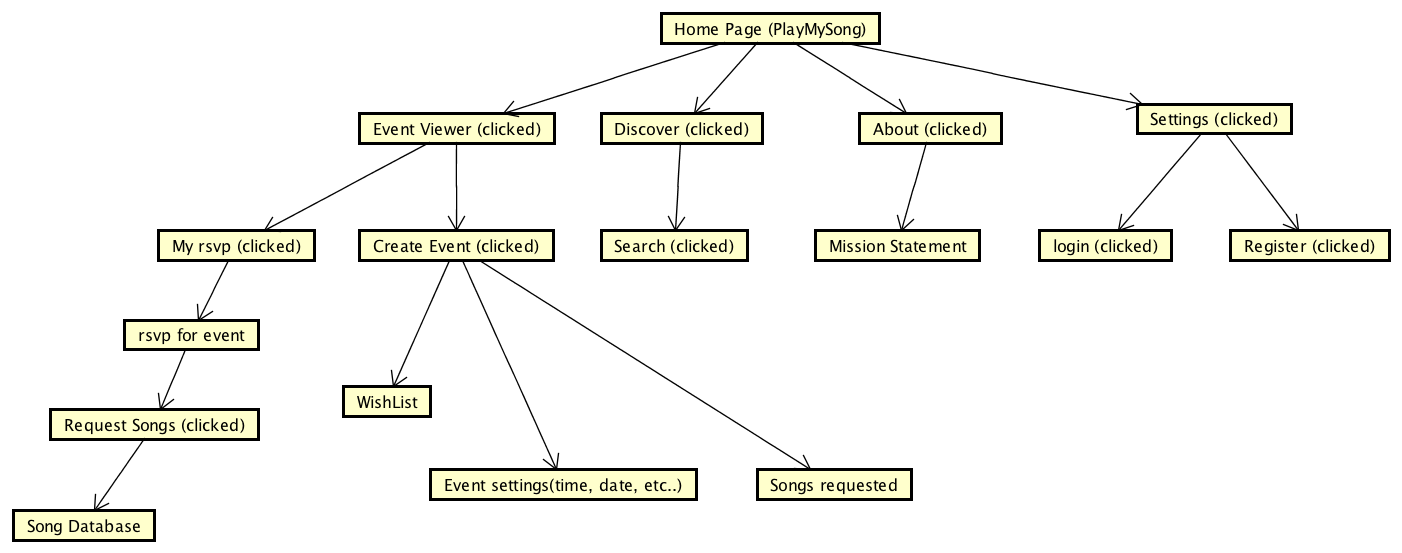
The purpose of this software design document is to demonstrate the overall design structure of the Play My Song application. This is intended to provide insight to the system’s structure and components. The topics covered in this document will include the following (but in no particular order):

* Introduction: Brief summary with Site Map
* User interface design
* Data and flow design
* Database design
* Constraints, limitations and restrictions designs
* Test cases and expected results
* Structure (internal, global and temporary)
  1. **Goals and Objectives**

Play My Song is an application that is used for creating events and requesting songs at an event. Along with the general design principles, this application needs to do the following:

* Pinpoint an exact location for an applicable party (map algorithm, search button)
* Include social media icons for following, liking, etc.
* Universal account for multiple users
* Show statistics for songs and DJ popularity, venues visited, etc.
* Have an app icon in the navigation bar (all pages)
* Queue song requests
  1. **Client-side and Server-side designs**

Play My Song is composed of two primary components: a **client-side system** that executes tasks like the ones above and a **server-side system** that communicates to a database:

Core Features

**Descriptions: In No Particular Order**

1. Settings, login and User Registration

* Users will be able to login or register
* Also allows user to adjust his/her account settings and preferences accordingly

1. Create Event and Wish-list

* Adds an event using a name, description, location, start/end time
* Designation for private or public with a randomly generated code
* Only the DJ users can edit events particulars (time, location etc)
* DJ’s can add to and modify Wish-list

1. Discover

* User can look up events (past, present and future)
* Includes featured events and local events

1. Search Events

* User can look for events using a built-in map algorithm
* Provides a short list of events within a certain radius of miles
* Pairs up with featured events, meaning they will be common hits
* Local events may be excluded

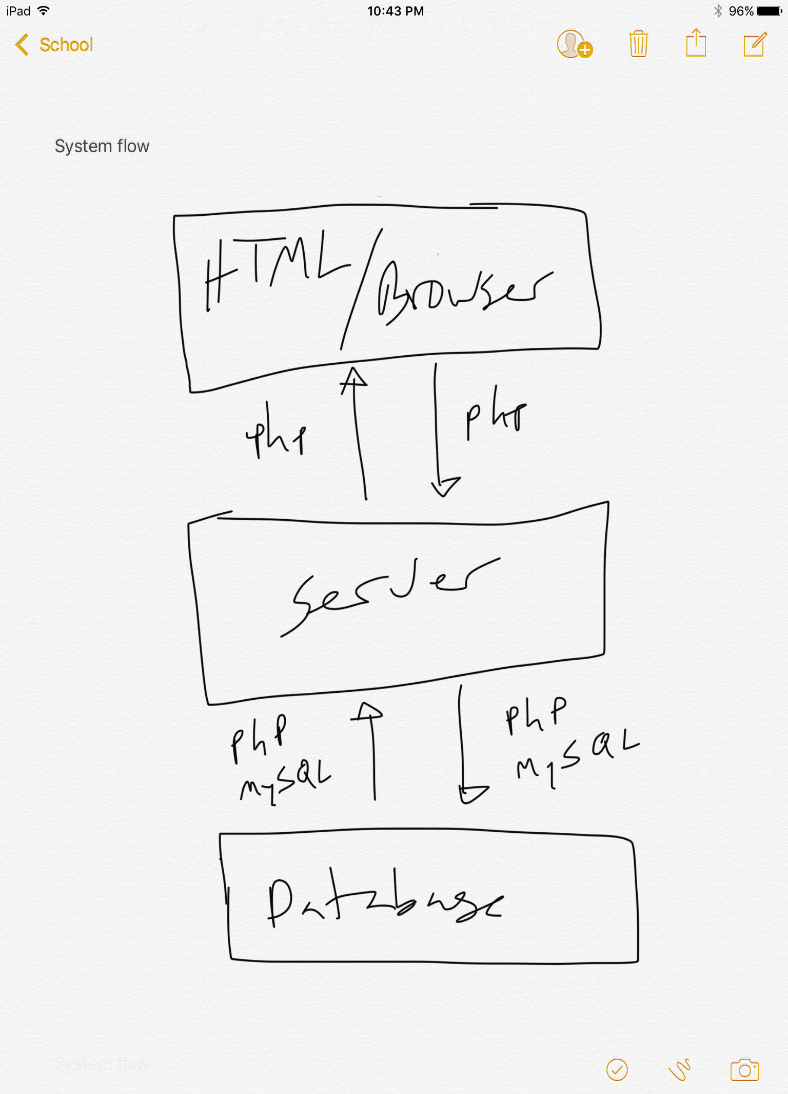
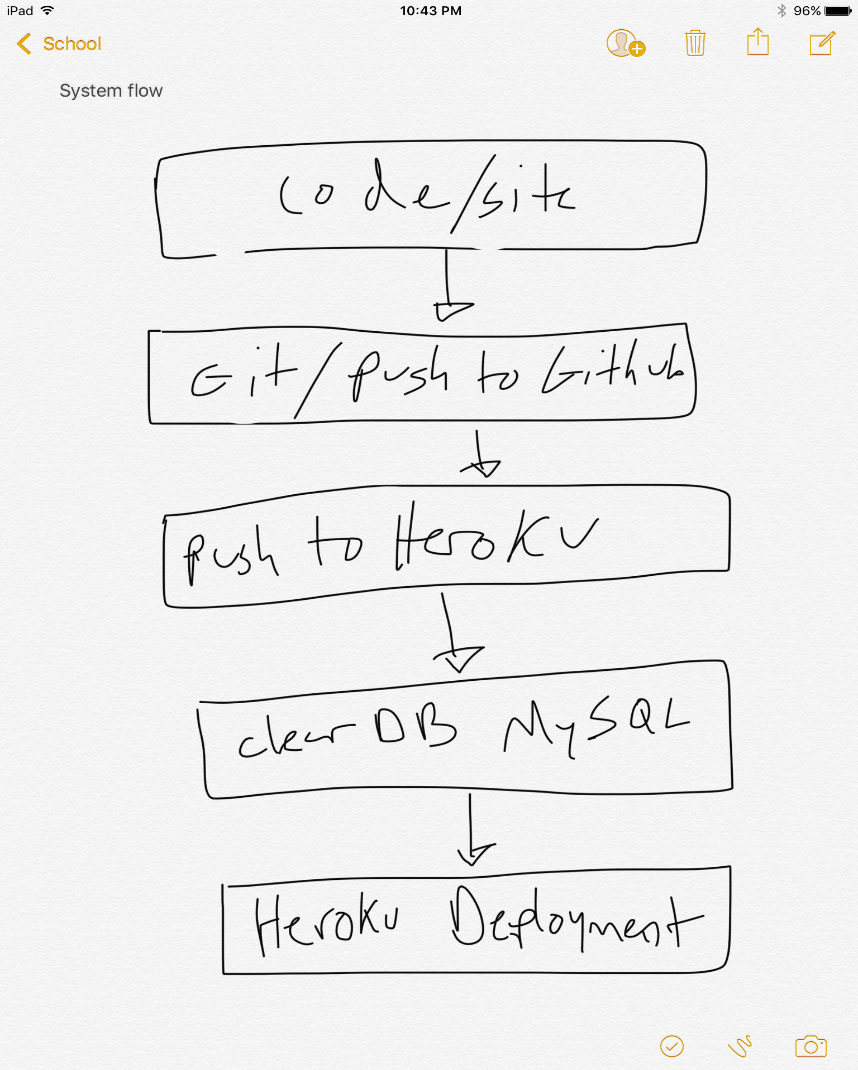
1. Song Requests

* User is taken to a page to submit their song and DJ will evaluate it for popularity
* DJ may accept, reject or “mark the song as lame”

1. About and mission statement

* Outlines our mission and also creator information/contact us (optional)

**Server-side Communication**



Additional Features

Any features that are not part of this design document will be implemented in another version.

* 1. **Intended Audience and Summary**

This software design document (SDD for short) is mainly created for individuals on the development team of this product rather than a general audience (SRS). This party includes the developers and team manager (one person in this audience could be one and the same). This doesn’t necessarily need to be read exclusively in order. When a question needs to be answered in relation to a specific section, a reader can jump to that corresponding set of passages. Here is a brief glimpse at the document’s contents:

*Part 1: Introduction*

* *A summary of the project: goals and objectives, scope, system details, major constraints*

*Part 2: Data Design*

* *Data structures and flow patterns*
* *Database details*

*Part 3: (Architectural and Component-Level Design)*

* *Describes Play My Song class by class*

*Part 4: User Interface Design*

* *Details relating to the graphical user interface (GUI) structure*
* *Mainly what the product will look like*

*Part 5: Restrictions, Limitations and Constraints*

* *Discusses the general constraint(s) weighed upon the project*

*Part 6: Testing Issues*

* *Lists out the numerous test cases, expected results and other relevant information*
* *Anyone interested in the software testing process should consult this section*

*Part 7: Appendices*

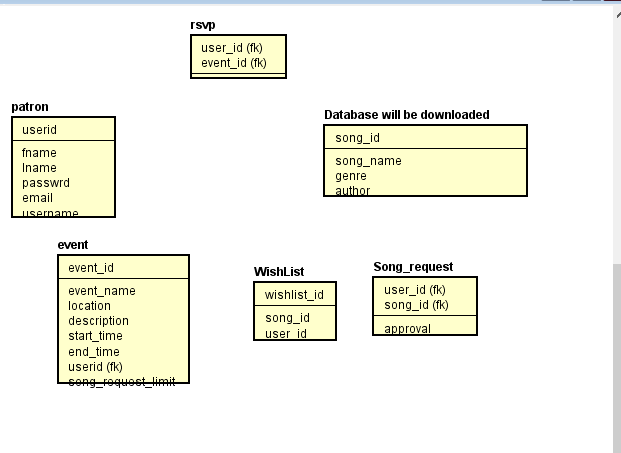
* *Our two models that illustrates how the product is laid out*

1. **Data Design**

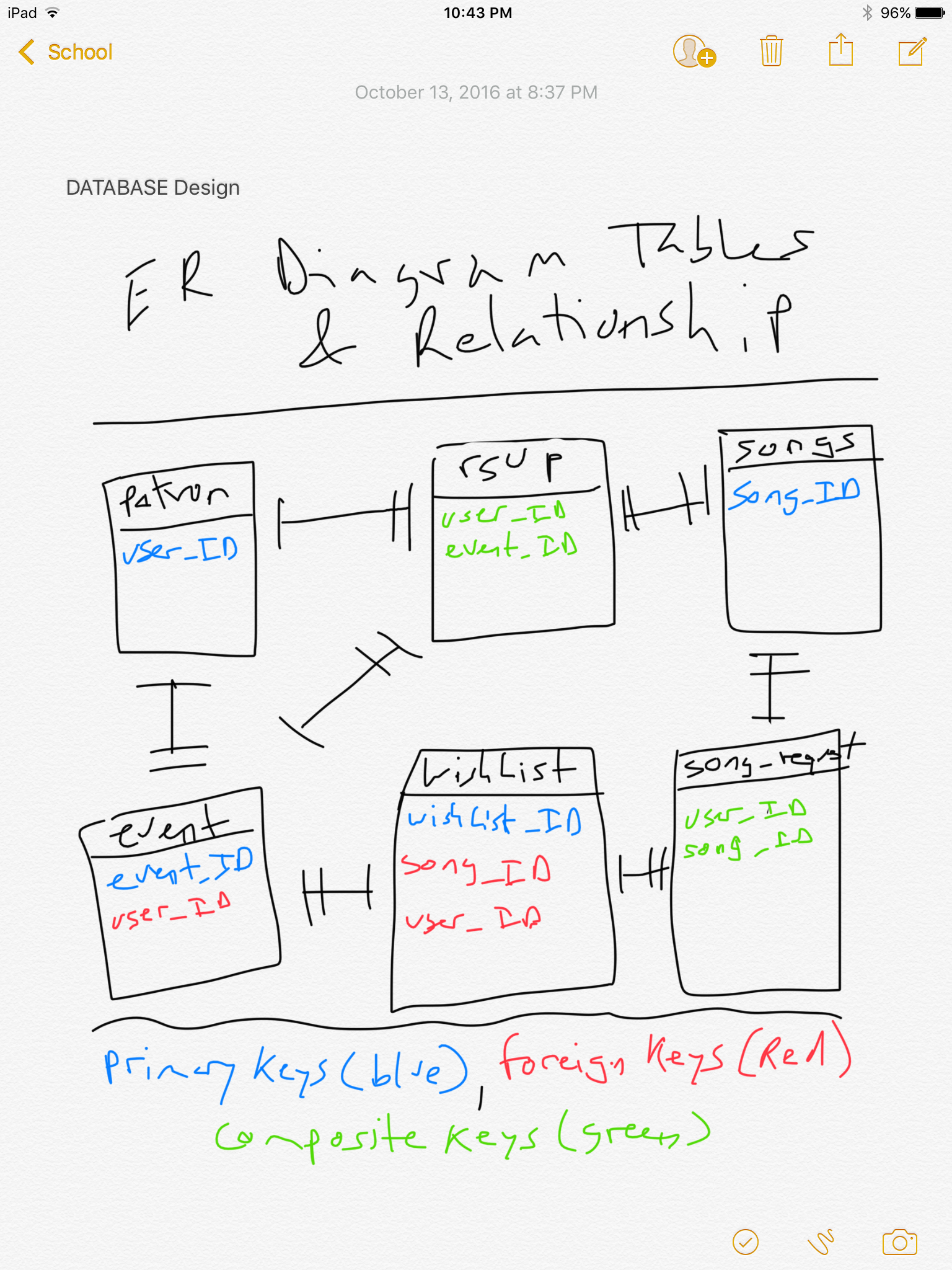
**2.1 Internal Software Data Structure**

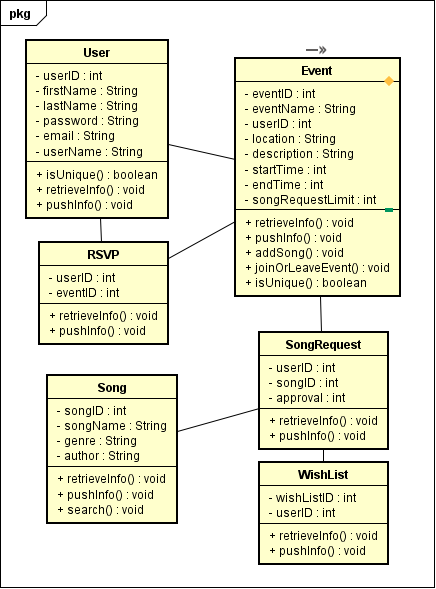
This application’s internal structure is divided into two parts, as mentioned above. There is a client-side and server-side.

The client-side will be based on the data received from the classes in the architectural/component design subsection further into the document. They will be formatted into a UML Diagram and Sequence Diagram (see *Appendices*). The data on the client will be requested from the server at system initialization and refreshed when applicable based on the user’s actions. Data structure in the server will be housed in three databases: user, song, event. The language for the database will be MySQL. The server will be implemented using the PHP language



**ER Diagram demonstrating required Database tables**

 **ER Diagram Relations Demonstrated:**

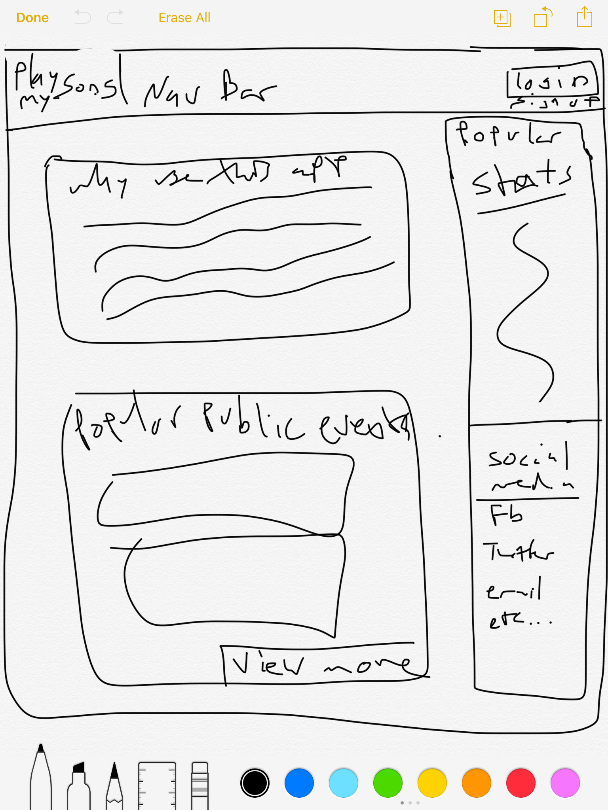
**3. ARCHITECTURAL & COMPONENT-LEVEL DESIGN**

**4.1 DESCRIPTION OF THE USER INTERFACE**

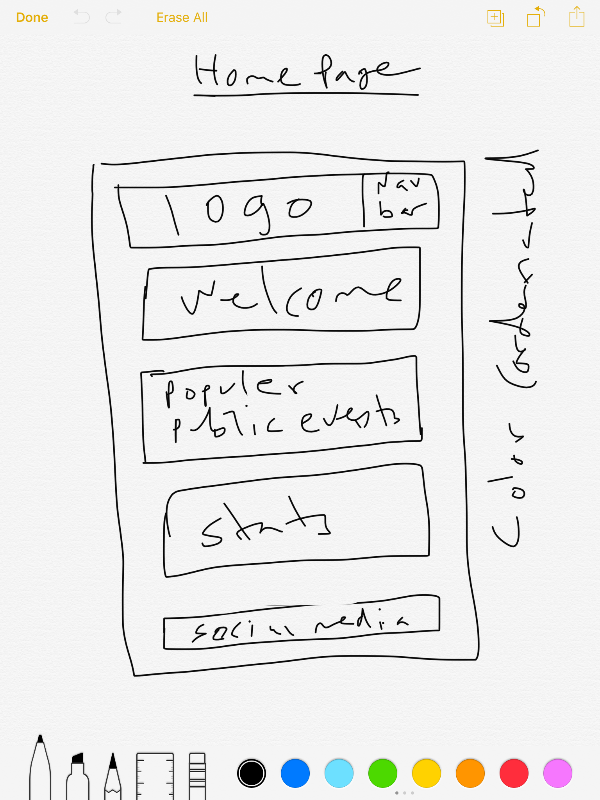
Pages will contain navigation bar, and the rest of the body/content. These components will be arranged in such a way that the user will be able to quickly grasp the purpose of each function and perform whatever task the user wants to accomplish, within the software scope

FOR FIRST TIME AND RETURNING USERS

Desktop View: HOMEPAGE



Mobile view: HOMEPAGE



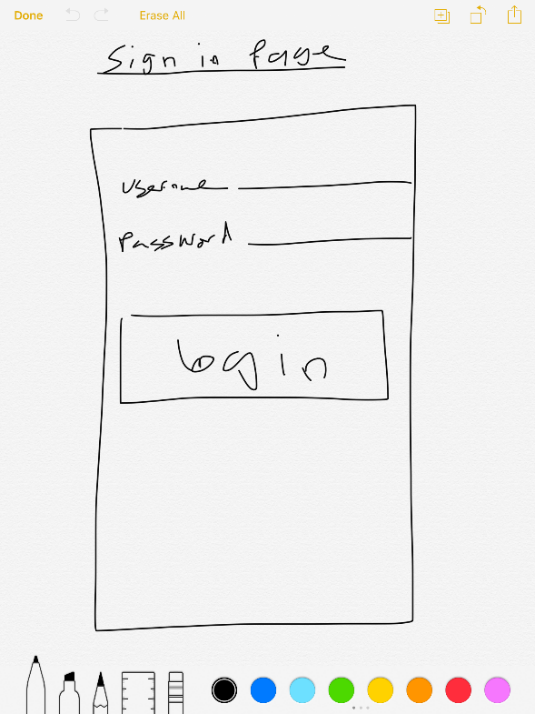
The Homepage is the screen page that the user first sees when they go to the site. It shows popular events, stats, and login/signup options

FOR FIRST TIME USERS

Desktop View: Sign-in Page



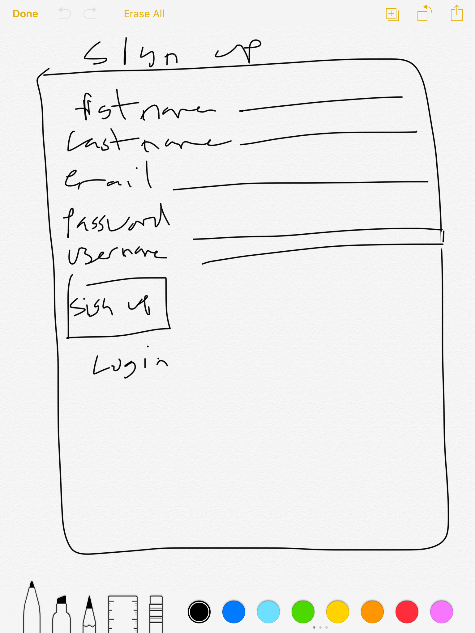
Mobile View: Sign-in Page



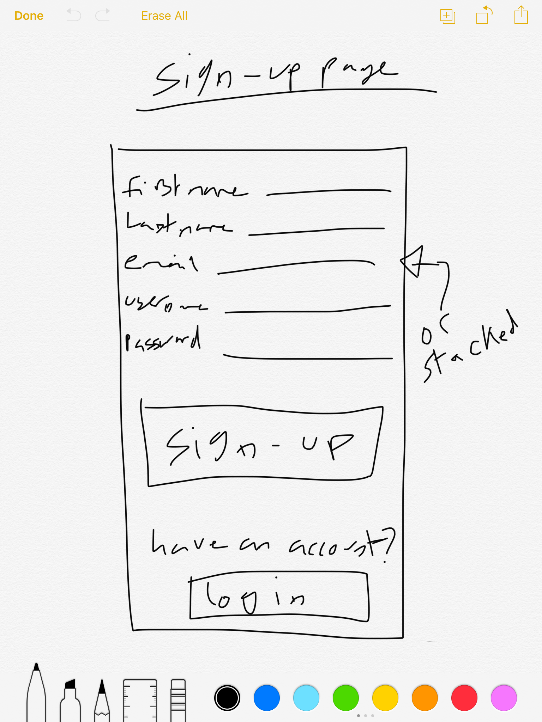
The login page will allow the user to log into their account where they will have additional access which as creating events or requesting songs

FOR FIRST TIME USERS

Desktop view: Sign-up Page



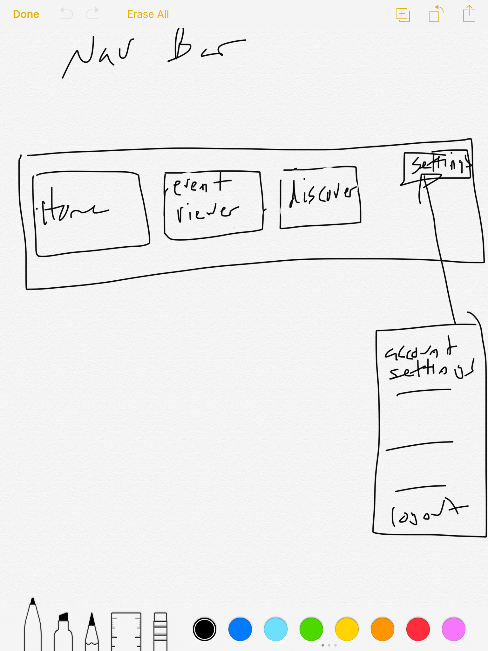
Mobile View: Sign-up Page



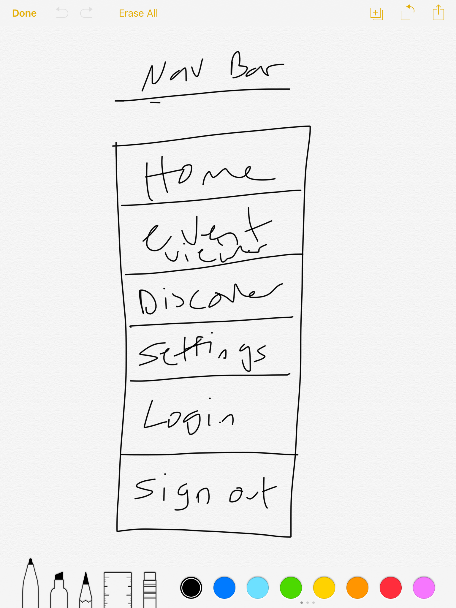
The signup page will allow the user to create an account.

FOR FIRST TIME AND RETURNING USERS

Desktop View: Navigation



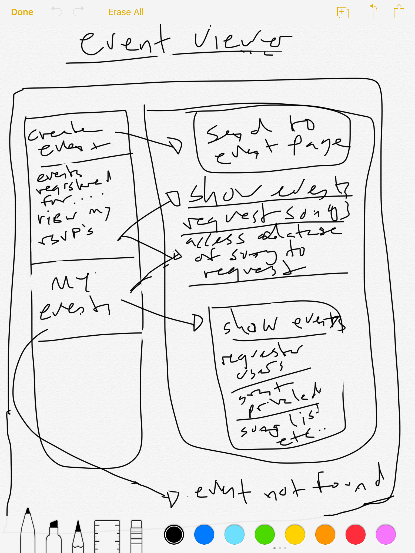
Mobile View: Navigation



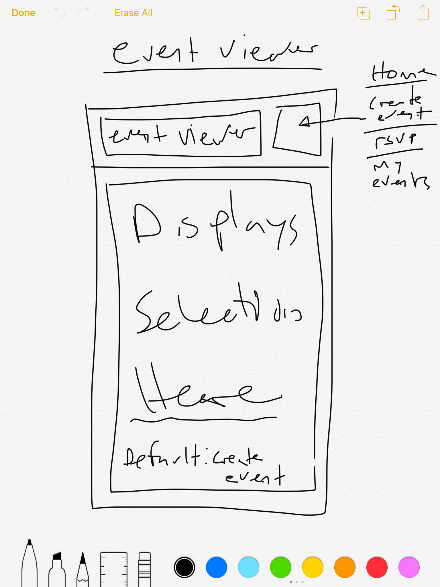
The navigation bar will allow user to navigate to different pages on the site

FOR REGISTERED USERS

Desktop View: Event Viewer



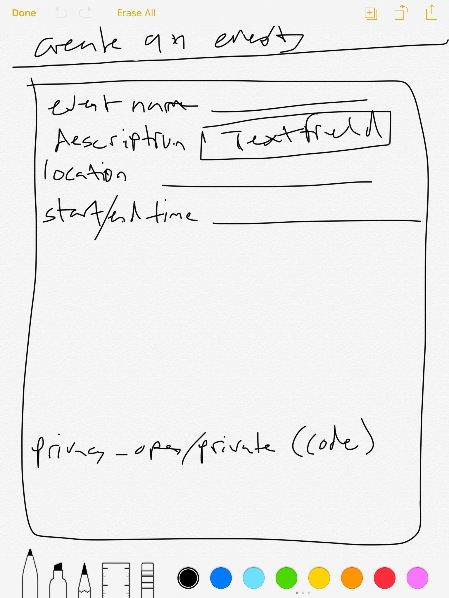
Mobile View: Event Viewer



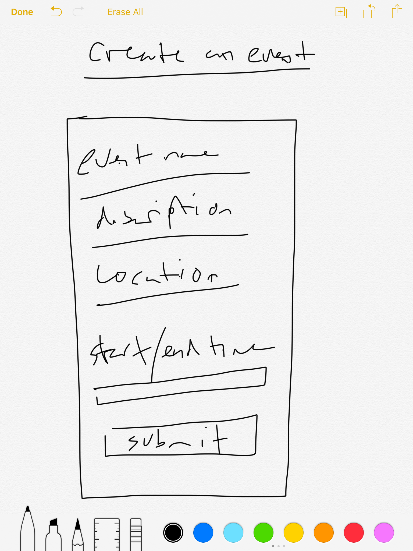
Event Viewer is a navigation bar tab that allows the user to create events, view rsvp’s, and view events that they’ve created

FOR REGISTERED USERS

Desktop View: Create Event



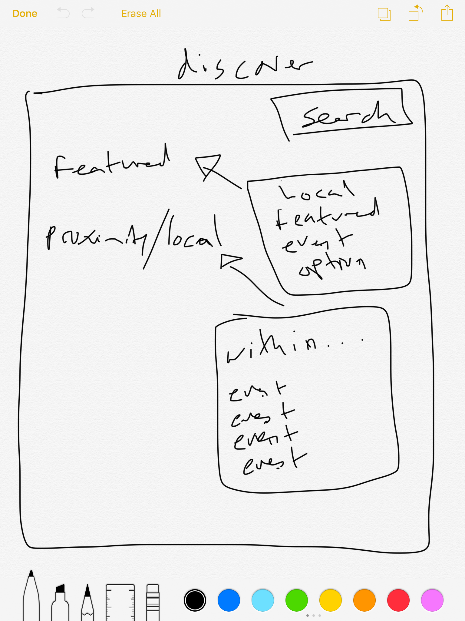
Mobile View: Create Event



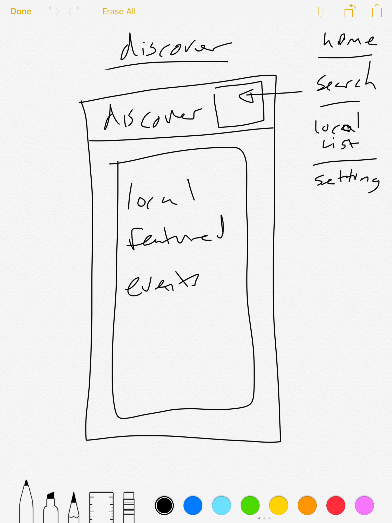
When the user selects create event from the View Events page, they will be sent to page. There they can create an event and specify location and start/end date and time.

FOR REGISTERED USERS

Desktop View: Discover



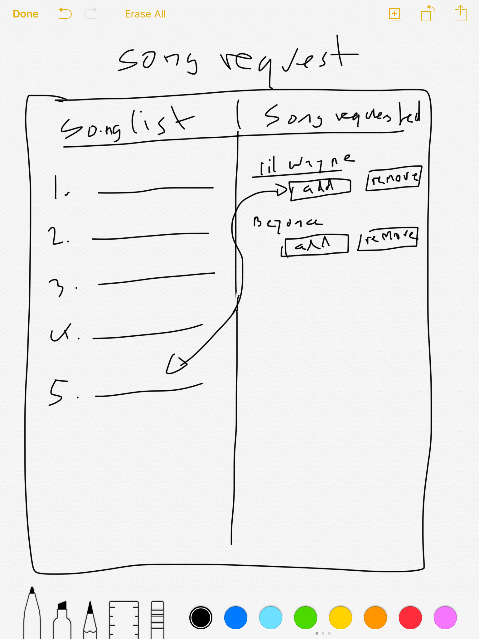
Mobile View: Discover



Discover tab shows local featured events. User can also search for an event by area code and user will be able to search events within a specified distance

FOR REGISTERED USERS

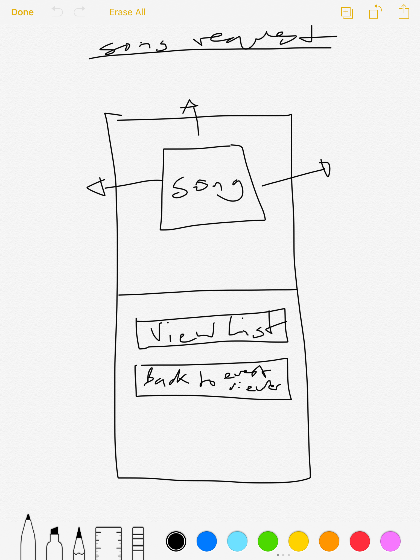
Desktop View: Song Request



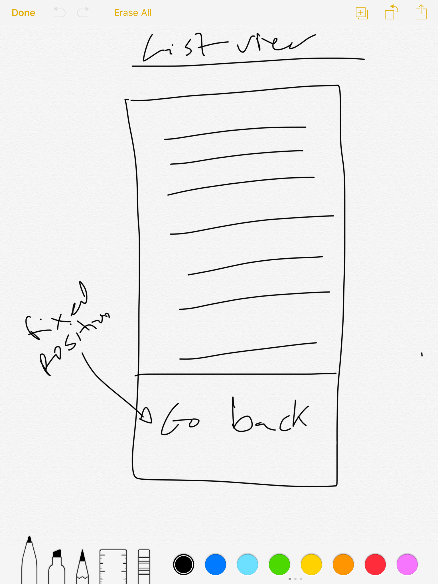
SEE Mobile View below for description

FOR REGISTERED USERS

Mobile View: Song Request



Mobile View: Song Request



Song request page for desktop view will show songs requests to the left and song list to the right whereas mobile view will show song request with ability to swipe left, right, or up. It will have option to go to list view which shows the song-list

4.3 COMPONENTS AVAILABLE

* We will use HTML, CSS, Some JavaScript and PHP.
* HTML Forms
* CSS Media Query
* JavaScript/PHP methods, save, login/sign-in and on-click features

**5. RESTRICTIONS, LIMITATIONS, AND CONSTRAINTS**

As a result of the highly modular design and organization of data – as well as unlimited expansion potential on the server side – implementing these optional features at a later date would be arguably easier than incorporating them into the first design.

A constraint and limitation of users is that they will need internet access to use the site. This is not a downloadable mobile app so users will not be able to save anything directly to their mobile devices or access features on the site as one would on a mobile device. Everything is done by going to the URL of the site.

Another constraint is that users who do not have an account will not be able to request songs or create an event. They will not be able to register for events or manage requested songs. Registered users will have a specified amount of song request privileges.

**6.2 Performance Bounds-** Since this application will be holding many users, execution time is negligible for local actions. For the server, it should have an acceptable performance ability when negotiating between server and client. In this application, PHP or Ruby on rails will suffice. As for keeping the amount of events in my events, about 10 will do. Only by increasing bandwidth of server will allow more events.

**6.3 Critical Systems**- The database has to be accurate in both events in case someone is searching for them and the location of the events. The database must be updated correctly and contain no corrupted or false data. Extensive testing has to be done to make sure the server and client implementation is valid. Since this program requires request and send information to server, it is important to have both ends working. Testing procedures will call to make sure information is passed correctly.

**6.4 Testing Cases-**

|  |  |
| --- | --- |
| Feature | Cases |
| Sign up to the app | Connection successful. New account, username and password and email sent to the server. |
|  | Connection failed. New account, username and password and email sent to the server. |
| Account log in | Correct information input. |
|  | Incorrect information input. |
| Event viewer | Create events- Adds event based on parameters |
|  | Events registered for- Client will return the events the user signed up for |
|  | View RSVP |
|  | My events |
|  | Event not found- Client will return nothing if the search does not match the criteria |
| Discover- search event | Event found using parameters |
|  | Event not found |
| Song request | Add song/Remove song- User is taken to page to submit their song and DJ will then accept or reject it |
| Event modification | DJ users can edit events |
| Performance | Time all actions to make sure device runs smoothly |

