15-237 Term Project
dcermina, nikhilj, shegde

Documentation (For submission on Friday 12/7)

**Summary:**
Our application offers a platform for couples in long distance relationships to stay in touch and communicate. The two people in the couple link their accounts to each other, and are given access to a dual profile page and a 'Playground'. Nobody else has access to these pages.

The profile page allows users to post quick information that the other person may want to view (such as profile pictures, their current mood, and their current status), upcoming events like birthdays and anniversaries, and couple statistics (time spent dating, etc.).

The playground page allows users to post texts, pictures, audio recordings, drawings, and user locations. These functions are easily accessible through tabs at the bottom of the screen.

The application is intended to be for mobile (iOS/Android).

**Features [for the end user]:**
- Profile page for both users:
    - Status, mood, profile pictures
    - Birthday/anniversary notifications [swipable off screen]
    - Couple statistics
        - Time spent dating
        - Next time seeing each other [date]
        - Days until next time seeing each other [countdown]
- Playground page:
    - Text Messages
    - Post Photos
        - Camera
        - From phone gallery
    - Post audio
        - From phone recording
    - Draw on canvas and send picture
        - Choose colors
        - Choose width of brush
    - Post location
        - Posts google map snippet on card
    - All messages stored on database, retrieved on login

# Design Process, Iterative Design, and User Testing:

## Section 1: The Design Process/Iterative Design

The application was designed with constant feedback from potential users of the app (i.e. People in relationships, some in long distance relationships).
The design of the application started off with a barebones jQuery mobile interface with a default theme, as in Figure 1.

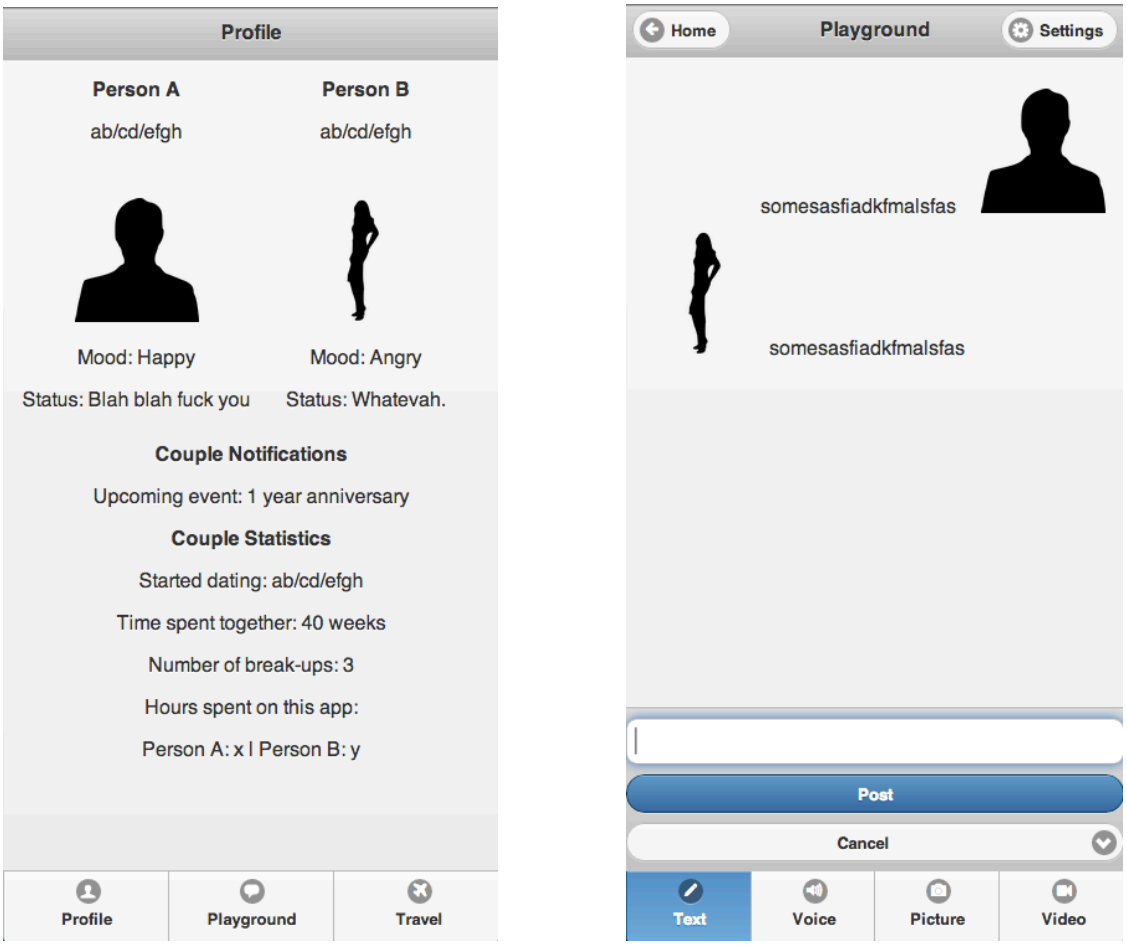



**Fig. 1 : Version 1**

The next iteration of the design was inspired by the slick “card” interface of Google Now, where different snippets of information are displayed on different cards. Around this time, we started using custom jQuery mobile themes to more closely relate our application to the color scheme of “Cherry”, and the concepts that relate to the name (intimacy, excitement, desire). This can be seen in figure 2.

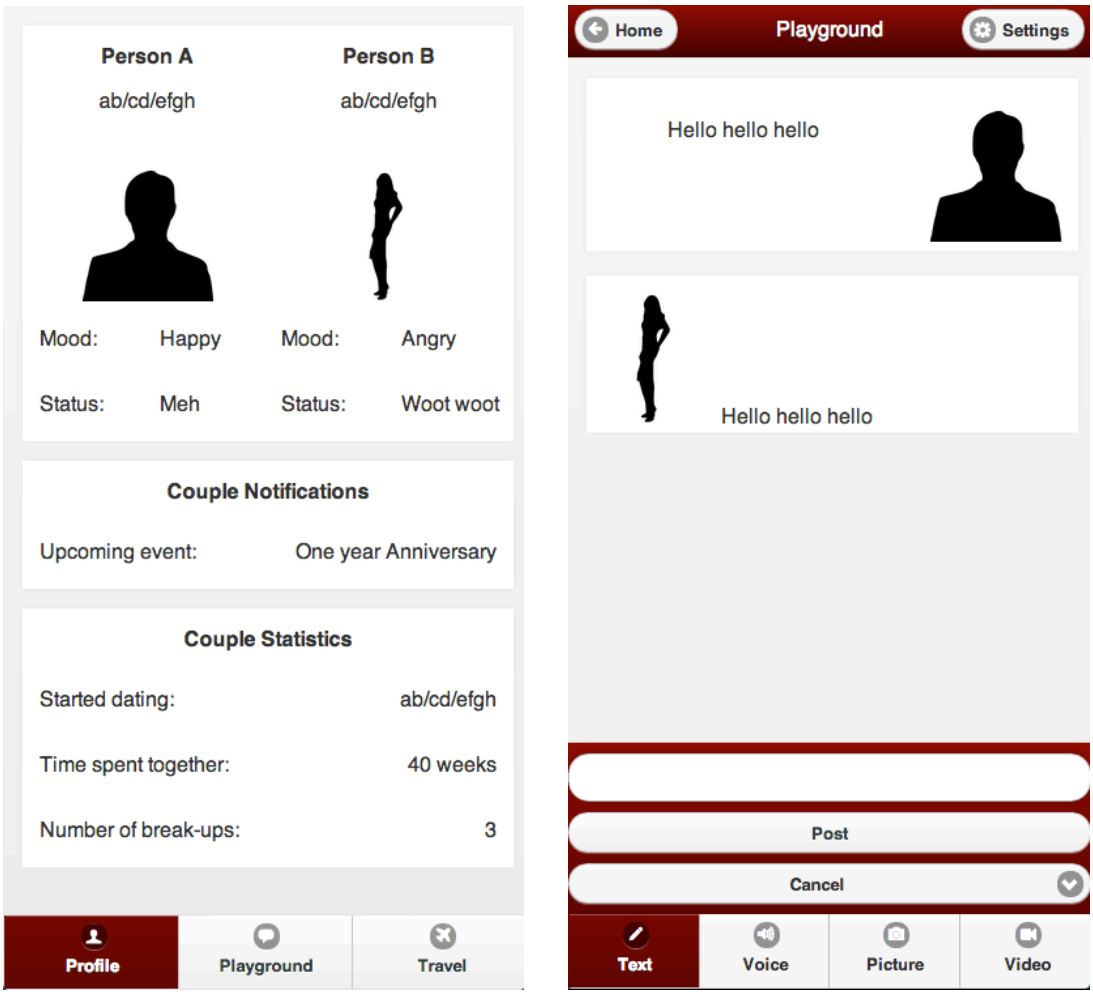



**Fig. 2 : Version 2**

During the course of our project, the travel site whose API we had initially planned on using for the travel section closed their public API service. Moreover, we got much more invested in the timeline (playground) component than we had initially anticipated. After discussing this with our mentor Brandon, we decided to scrap the travel section and focus instead mainly on the playground component. After getting some user and designer feedback on the playground section (namely on the button positioning, timestamps, and chat window proportions), the designed evolved into what can be seen in Figure 3.

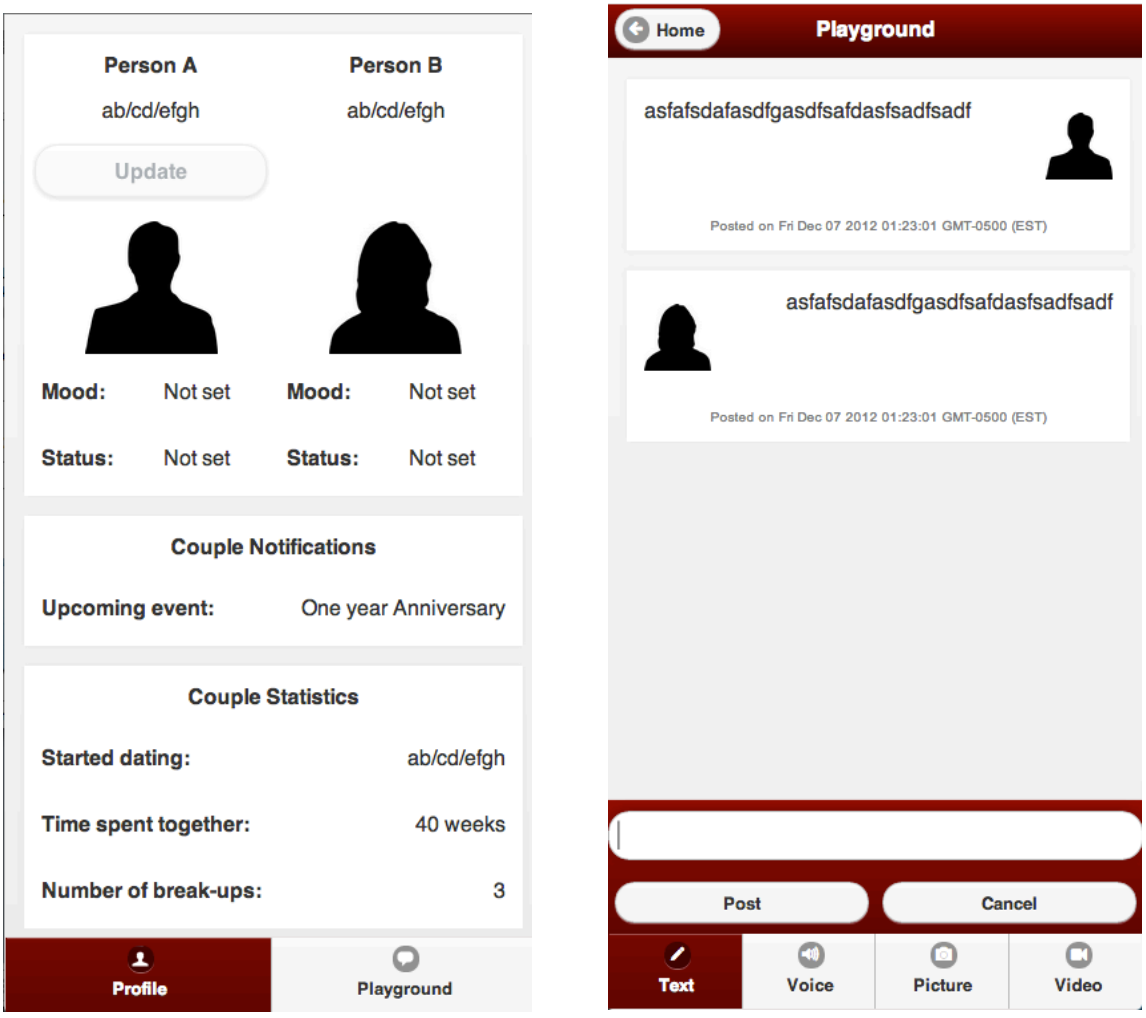
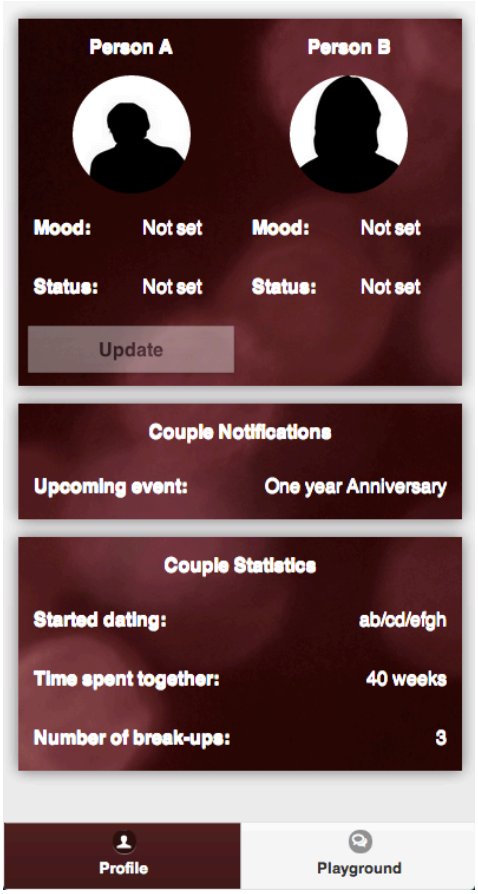
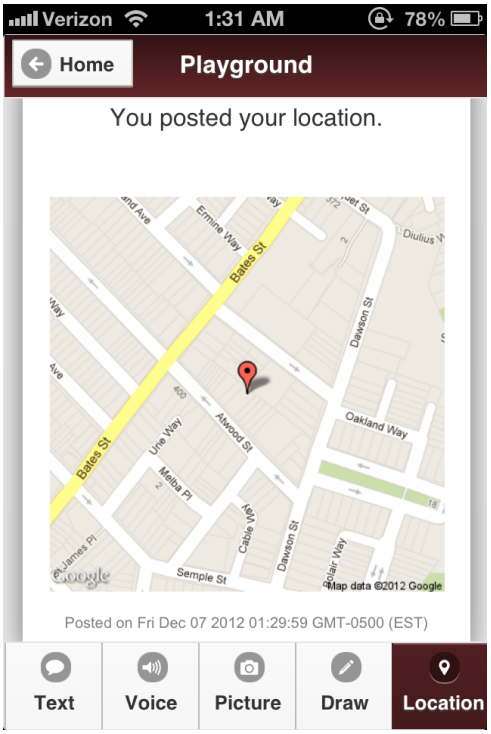
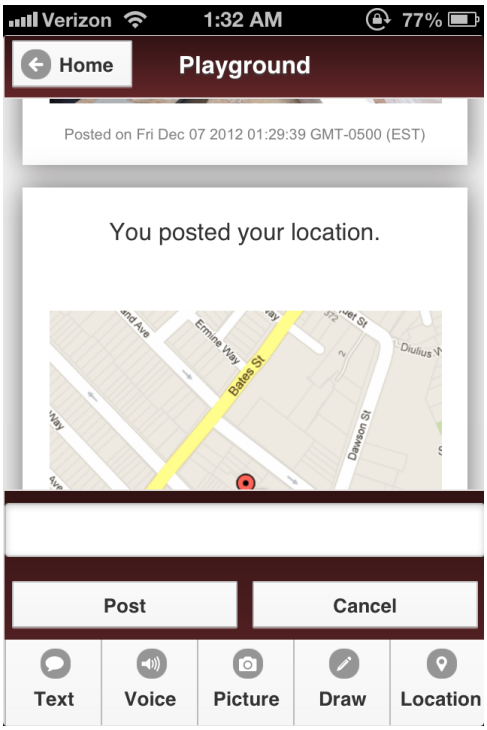


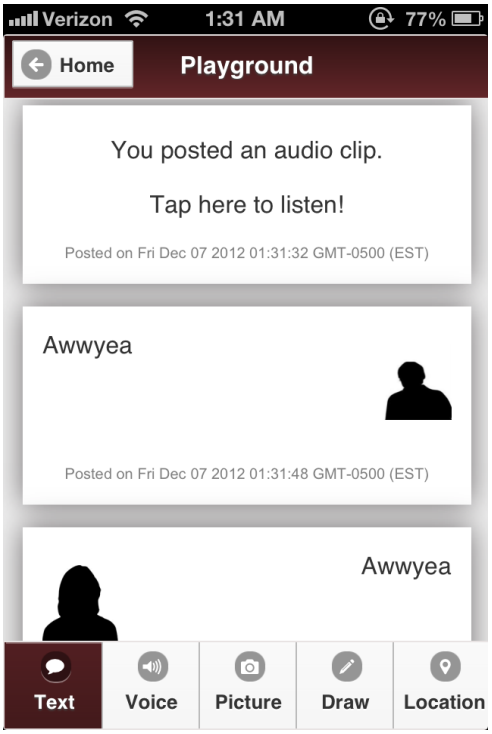

**Fig. 3 : Version 3**

For the next (and final, for the moment) iteration, we worked extensively with a designer to create a background in Photoshop to add character and charm to a nearly monochromatic interface. This was important to do since this application is meant to appeal to people in relationships, and the UI plays a significant role in keeping up the appeal for both parties involved. The idea behind the darker red was to create a sense of heady luxury through its similarity to the color of wine. The inspiration behind the rounded profile picture displays is the UI from Path, which in general has a clean and intuitive interface. Refer to the images of Version 4 below for examples.




**Version 4 (Current)**

## Section 2: User Testing and Feedback

User testing was done with the following people:

- People from class (Akash Kulkarni, Anish Phophaliya): **Version 2**
    - Suggested adding something to indicate swipable surfaces
        - Work in progress
    - Suggested uploading media from gallery to playground
        - Implemented (v4)
- Mishq Laliwala (Sophomore Art Major in relationship): **Version 3**
    - Suggested less round buttons
        - Implemented (v4)
    - Suggested a darker red theme
        - Implemented (v4)
    - Suggested moving the Update status button on homepage
        - Implemented (v4)
- Rupa Patel (Junior Information Systems Major): **Version 3**
    - Suggested blurred background image for cards
        - Briefly implemented, but got inspiration to iterate towards the current version 4 design
    - Suggested Path UI for inspiration
        - Implemented circular profile pictures
- Sherry Chiang (Biology): **Version 4**
    - She thought it was 'cute'
    - She likes the cherry theme
    - She likes the play on words


### Acknowledgement:

Meghna Raghunathan (Junior Design Major in long distance relationship): **v3 –> v4**

- Provided feedback on theme/overall look and helped iterate from version 3 to version 4
- Worked with us on photoshop design for version 4


**Note: The iterative design process is still underway, and the designs shown above may not necessarily reflect the final design presented on the 14th.**

Elements Used:

1. Canvas:

    - Canvas that can be drawn on by user through the playground. Users can choose brush color, brush width, and tap to drop shapes (possibly more)

2. CSS (including a reasonably range of techniques, such as reset, pseudo-selectors, fixed and fluid layout, transitions and animations, etc)

    - Custom CSS for the entire body of the application [card system], custom fonts, selectors, multiple classes for several elements, transparency, more to come

3. DOM manipulation

    - Profile page updated based on selections [mood, status, profile pic] and server-side information [details for the other user, notifications card, stats card]. Playground page updated based on user posts [every kind] and server-side info [reload all old cards on page load]

4. jQuery

    - Used throughout app to add event listeners to DOM (tap, swipe, etc), to animate (fade ins for cards), for DOM manipulation (appends, hides, etc.) for form validation, and more.

5. jQuery Mobile or Sencha Touch or equivalent (including your own high-quality custom-made UI widgets)

    - The application was built using the jQuery Mobile framework. Multi-page template, dialogs, popup boxes, a custom icon set [Github: /commadelimited/jQuery-Mobile-Icon-Pack], custom theme [themeroller], extensive use of grids

6. PhoneGap

    - App works on Cordova for iOS and Android. Allows for camera use, microphone use, location services, and local gallery access.

7. AJAX (or similar) client (consume an API)

    - All information collected on the client side is sent to the backend using AJAX calls to the backend server, and it is then stored on the database. Moreover, the client can use AJAX calls to request data from the backend server.

8. AJAX (or similar) server (provide an API)

    - The server responds to calls from the client using AJAX calls so that it can serve data back to the client from the MongoDB database. Moreover, it sends confirmation responses through AJAX calls to the client when data is requested to be stored.

9. node.js (with or without express)

    - The entire server-side code is written in node.js, where data gets stored as events are fired

10. server-side databases

    - We used MongoDB with Mongoose to store user data (including playground messages), and GridFS to store large files such as images and audio.