

# MIT RScanVP - GR6

Added by [Matthew Falk](#), last edited by [Matthew Falk](#) on May 17, 2012 21:13

## User Testing

### Design

- The final design of our interface keeps all of the functionality from our previous prototypes but aims to simplify the workflow to make the entire process as quick and smooth as possible.
- The home screen has been changed in response to heuristic evaluation comments, as it was originally too wide to be seen on many modern devices. We changed the format of the home screen to include a marquee instead of a ribbon of images so that the view will not be as wide. This was much cleaner and allowed the page to be displayed cleanly on all devices that we tested.

RScanVP is the quick and easy way to add RSVPs to your event posters.

**Design QR Code**

Choose a color for your QR code:

Color:  Hex:  Size:  Style:

QR Code: 

When you are ready, click the QR, keep and download your QR code.

[Back](#) [Done](#)

**Bob's UI Lecture**

Tue 5/1 3-4 PM  
2-123

**Alert**  
15 minutes before

[Add To Calendar](#)

RSVPs to Date

1.00  
0.75  
0.50  
0.25  
0.00

[Remember](#)

[Get Started](#)

To create a code: IE 9+, Firefox 3+, Chrome 3+, Safari 3+

To view a code: iPhone with Barcode Scanners that opens in Safari. (AT&T Scanner works) Other phones/methods coming soon.

6.813 UI Designs 2012


- After clicking the get started button on the home page the user is presented with an initial form to fill in the event information that the attendees will see after scanning the QR code they are creating. Unlike in earlier prototypes

where all input fields were identical in size, the event description input has been expanded into a multi row text area to be more consistent with the fact that event descriptions can often be several sentences. In response to heuristic evaluations that showed our time entry field had usability issues, this area was redesigned to include a drop down to allow users to select times without having to manually type them in. Error checking was also implemented to alert users if their date or time entries were invalid.

The screenshot shows a mobile application interface for 'RScanVP'. At the top is the logo 'RScanVP' in a blue, stylized font. Below the logo is a section titled 'Enter Event Details' in bold black text. The main content area has a blue background with white text and input fields. It includes: 'Event Title' (text input), 'Event Description' (multi-row text area), 'Location:' (text input), 'Date:' (text input), and 'Time:' (two text inputs labeled 'Start' and 'End'). To the right of the date and time inputs is a QR code. At the bottom of the form are two buttons: 'Cancel' and 'Next'. Below the form is a footer that reads '6.813 UI Designs 2012'.

- After inputting event details users are brought to a page where they can customize the color and size of the QR code and then download it. Previous prototypes had divided QR customization and QR download into two separate pages. Feedback during heuristic evaluation indicated that the download page, which previously showed as sequence of instructions about downloading the code and how to use it, was overly complicated for the task. We therefore decided to streamline the process. We added text field for the user to enter their email address as well as a text field containing the user's link to the admin page, to the QR code design page, and removed the additional page altogether.

**Choose a color for your QR code:**



Red:

Green:

Blue:

**Choose a size for your QR code:**

Height/Width:  inches

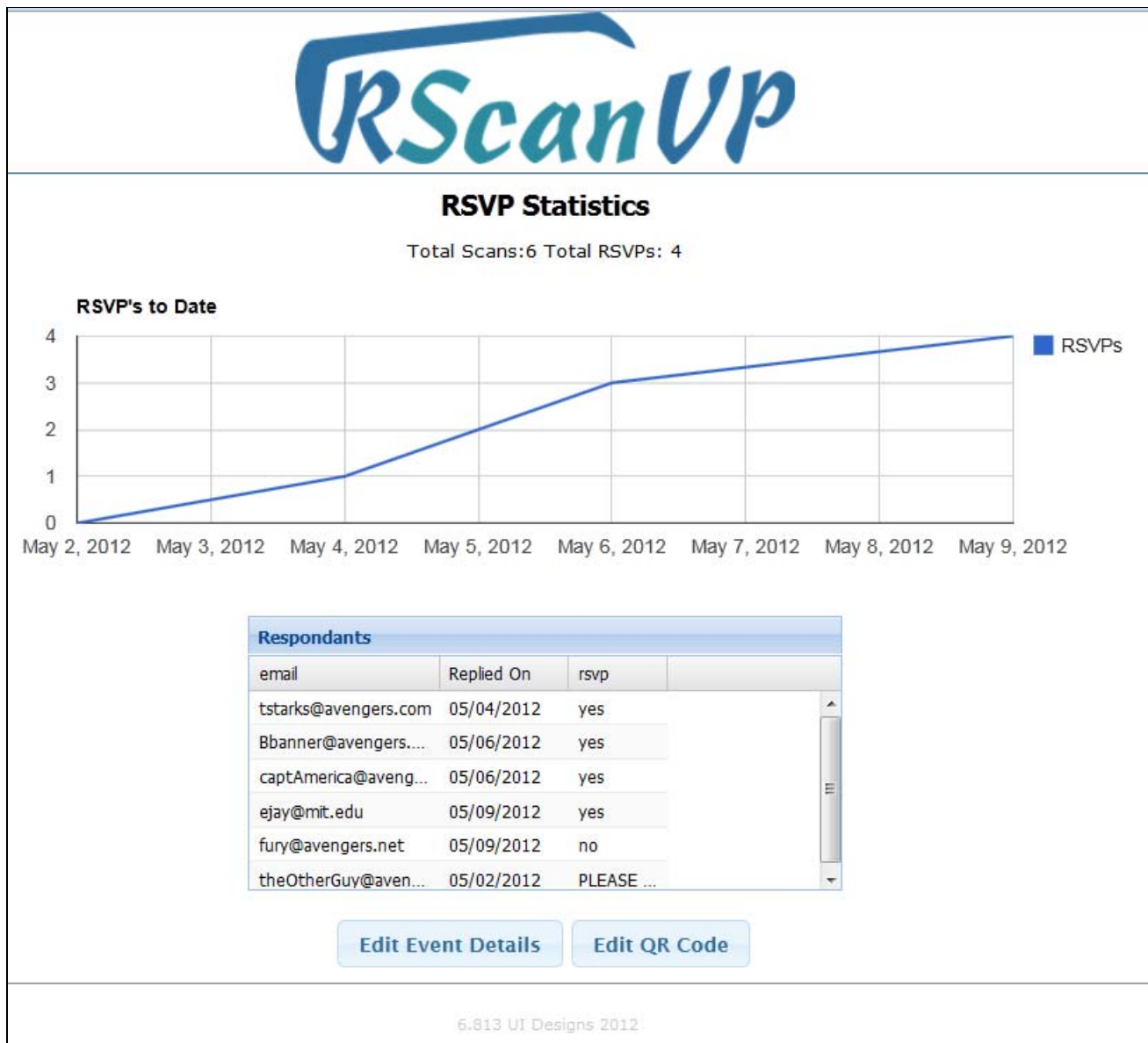
**Access your Event Responses:**

Email:

Optional: You will receive an email with the following link.

Admin Link:

- The admin page where users can view RSVP statistics remained the same functionality wise throughout the iterations into our final design. It allows users to view a table of all those responded as well as a chart showing response numbers versus time. However, the layout of the elements on the page was updated to be internally consistent with the rest of the site. The main elements were centered, with the navigation buttons to the other parts of the site moved to the bottom.



- The heuristic evaluation noted a lack of feedback for sending RSVP responses. Therefore, we updated the view page to indicate to the user when a response message had been sent to the system.

## Implementation

- We use our own server to host the website. We use PHP, JavaScript, AJAX, JQuery, and HTML to implement the site.
- The back end of the site provides a MySQL database that stores data provided when users use the site. Whenever a user creates a QR code for a particular event. All of the details for that event are stored in the database along with enough information to generate the QR code. The user is allowed to re-edit QR codes and update their values in the database.
- The front end of our website features a simple and usable design that allows users to enter information pertaining to a particular event. Our website creates a QR code that gives potential attendees a simple way to RSVP to these events by scanning the QR code (hence the name RScanVP). The user is presented with another page of development that allows them to recolor the QR code and also re-size it within visible constraints.
- QR codes are regenerated each time a new size or color is set. This was simpler than recoloring the image in

JavaScript and thus we chose this way of doing it. However, based on user testing, this was a bad idea because users get confused when the QR code keeps changing.

## Implementation Fixes

Based on feedback from testing we fixed several issues with RScanVP

- We redesigned the homepage to better fit on narrower devices.
- We added a drop down for times to the event details page.
- We added a poster to the event details page to communicate the poster metaphor.
- We decided to take away the "Download" button altogether. This makes it a little harder for users to acquire the QR code that has been generated. We made this decision so that the user was not redirected away from the site at any time because we could not figure out any other way to do it.
- We decided to add red text indicating an error in text fields so that users would know why their submissions were being rejected. Our user testing indicated that this was a good idea but would be more helpful if it were more visible and possibly located next to the field that had the error.

## Evaluation

### Getting the users

- We decided to use pairs of users to test our project. As the project involves someone creating a QR code and then another person accessing that QR code and responding to an RSVP, we figured it made more sense to use separate users and give them focused tasks instead of making them perform tasks as multiple people. We gathered three pairs of people where each pair consisted of an MIT poster maker (dorm, school, and sport) and an random MIT student. By getting three different types of poster makers and a small group of random MIT students we feel that this represents our target user population very well. The poster makers are exactly the types of users we were hoping to build out website for and the MIT students fit the role of event attendees very well.

### Briefing the Users and User Tasks

- After getting the users who would participate in the test, we told them that we created a website to help with the tracking of results for advertised events.
  - Poster Maker - You are in charge of publicizing for an event. Your first task is to use our site to create a QR code that prospective attendees can scan to send an RSVP for the event. Later on, you will be tasked with viewing the statistics for your event.
  - Prospective Attendee - Your task is simply to scan a QR code, RSVP 'yes', and choose to get an email reminder/add it to your calendar.

### Usability Problems and Non-Implemented Solutions

- User did not know what a QR code was - Major - We could add more information about what a QR code is to the homepage.
- Download Obscurity - Catastrophic - Users were unsure of how to actually download the QR code. There was a note saying to right-click and download the image, but users got stuck because the term "download" was not consistent with the supplied option of "Save Image As...". In order to fix this, we could simply change the line of text telling the user how to download and make the syntax consistent. Further, we could simply email the image to the user when we email them the Admin link so that they do not have to worry about downloading it. However, this means that supplying an email would not longer be optional.

- **Size Restriction - Minor** - The user is restricted to certain sizes when re-sizing the QR image. We limit the size of the image to be between 1 and 5 inches; however, we never alert the user of this until they made the mistake of supplying an out of range size. We can fix this by putting an indicator that we only accept certain sizes. Further, we could fix this by having the size be a slide rule. Thus, the user can slide to the exact allowed size that they want but will have no way of getting outside those bounds. This would also supply them with sufficient feedback as to what sizes are allowed.
- **Recolor Confusion - Major** - Users were a little confused that every time they clicked "Set Color" a new QR code appeared. As mentioned in the Implementation Section, we did this on purpose. To avoid this, we could have made re-size and re-color happen on the QR images themselves rather than trying to create a new QR image each time the button was clicked. As a simpler, but not desired, solution we could have written a note to the user explaining that the codes would continually change but still work, so they should not worry that the images keep changing.
- **Not Download Error - Major** - Users felt that downloading the QR image was important enough that there should be a safety precaution in place to check if the user actually downloaded the image. When a user clicks done on the Design page, they should be prompted with a pop-up box asking if they downloaded the image. This solves itself and would help users not to miss the line of text telling them to grab the image.
- **Redundant Color - Cosmetic** - One user thought that after choosing a color, he needed to also click the "Set Color" button to confirm the color of the QR code. He found this slightly unnecessary and thus redundant. Of course, this was not actually necessary, but he did not know that. We could solve this by letting the user know that the image displayed on the page is actually the QR image that they will download. This will help clarify that changing the color is a permanent action and no further actions are required to set the color.
- **AM/PM Confusion - Major** - The radio button on the phone is confusing because two of the users (possible attendees) were unsure of whether or not they were going to the event. They had trouble deciphering whether or not "Yes" was the currently selected option. We could fix this by simply providing a better radio button that makes it much more clear which option is being chosen.
- **Error Text - Minor** - As noted by the TA, error text was far from the fields that had the error. This was not mentioned in the user testing because the screen that we used for user testing happened to be larger than the one used for GR5 demoing. Thus, it was clear to users when the text appeared that it had appeared.
- **User did not know if they had a QR code scanner - Catastrophic** - We could ask poster makers to insert instructions on how to download a QR scanner. We could ask poster makers to also provide a text URL for them to enter.
- **Color took a while to change - Minor** - We could indicate that the page was processing while the new QR code was loading.
- **Didn't hit set size - Minor** - The user typed in a size, but did not hit "Set size". We could automatically set the size when the box loses focus.
- **The user did not want to give their email - Minor** - We could add a message saying their email will only be used once.
- **Network slow - Out of Scope** - The user was confused when the internet network was slow; little you can do about this.
- **User surprised could see other RSVPs - Out of Scope** - Security was not a requirement for this project.

## Reflection

- The main thing that we learned over the course of the semester is that we are not our user. That is, even if we think something is simple, obvious, or clear, this does not mean that the user will feel the same way. We have a lot of preconceived notions about what things should do because we actually created them, the users of our

system will not be in that position and are thus much more likely to be confused.

- There is no one right method of testing. We caught different things that needed to be fixed when doing heuristic evaluation vs paper prototyping for example. Heuristic evaluation was much better for finding things that needed to be visually fixed. It was very helpful to get someone else's opinions on what the website was missing in terms of display and aesthetics. Where as, paper-prototyping was better at providing us with functionality problems. User testing was very helpful in drawing a line between what we thought vs. what the user thought and how we perceived the site as a whole. For us, the most useful tests were those where users tried to play with the system and commented on wherever they got stuck or confused (i.e. paper-prototyping and user testing).
- If we were doing this project over again, we would focus on providing more options for how things are done in paper-prototyping. For instance, we would present three options for QR image re-sizing. One a text box, one a drop down box, and one with a slide-rule. Basically, we would have preferred to add a step into the iterative design process where we did something like AB Testing to determine which designs were optimal for the most users.
- Doing this project again, we would focus a lot more on testing with a lot of users (as opposed to less than 5). With this many users it was hard to get a general feel for how they all felt together, but rather their individual concerns, which may or may not have been necessary or realistic.
- Finally we would have liked to do some smaller rounds of user testing, not just on the final implementation, but on some of the modular sections as they were completed. While the overall experience for the user cannot be tested without an almost fully complete implementations, several individual sections of the application are significant enough to stand on their own and warrant user testing. In this way, we could catch usability problems earlier on in each of the parts and iterate on them, before pulling all of the tasks together into a test of the overall final design and experience.

---

## Get Help

[User Guide](#)

[Confluence Help](#)

[Knowledge Base](#)

[Training](#)

[Contact the Help Desk](#)

[Request a Wiki Space](#)

## Resources

[Terms of Service](#)

[Supported Browsers](#)

[MIT Touchstone](#)

[Stellar](#)

[WebSIS](#)