

SOFTWARE REVIEW:

# Building a Low-Fidelity Prototype with Justinmind

BY MICHAEL OPSTEEGH | *STC Senior Member*

ONE OF THE FOUNDATIONS of an effective usability test is selecting the right tools. All of your careful planning, participant selection, and test development can be undone if your prototype fails during testing. A malfunctioning prototype may cause test participants to be confused, question whether they completed a task, or spend valuable test time remarking on the limitations of the prototype. Any of these situations can skew key metrics, such as memorability, learnability, perception of errors, and satisfaction, and can ultimately distort your test results.

There are, of course, many ways to mitigate a prototype failure, like beta testing or falling back on paper prototypes, but selecting the right tool for developing your prototype is the best first step. The Justinmind Prototyper ([www.justinmind.com](http://www.justinmind.com)) provides a flexible platform for developing interactive, low-fidelity prototypes for your usability tests. Justinmind, however, has some limitations, so it may not be suitable for all of your projects.

## About Justinmind

The Justinmind Prototyper allows you to develop functional wireframes for websites, desktop programs, and mobile apps. Simply drag controls, text boxes, and other elements onto the canvas, and add an action, a calculation, or other functionality (see Figure 1).

The primary focus of Justinmind seems to be collaboration between UX designers and developers, rather

than usability research. The built-in tools are designed to enable you to share your prototype with stakeholders and solicit feedback through comments. Having said that, Justinmind makes concessions for usability testing by integrating with such third-party services as Validately, UsabilityTest, and Google Analytics (see Figure 2). To connect your project to these services, you'll need to share your project in the Justinmind application, and then go to your Justinmind online account, locate your project, and select the Usability Testing Tools option from the Action drop-down.



Figure 2. Online sharing enables you to connect to usability testing tools.

Although Justinmind offers the ability to connect to these usability testing services, the limitations in the actual

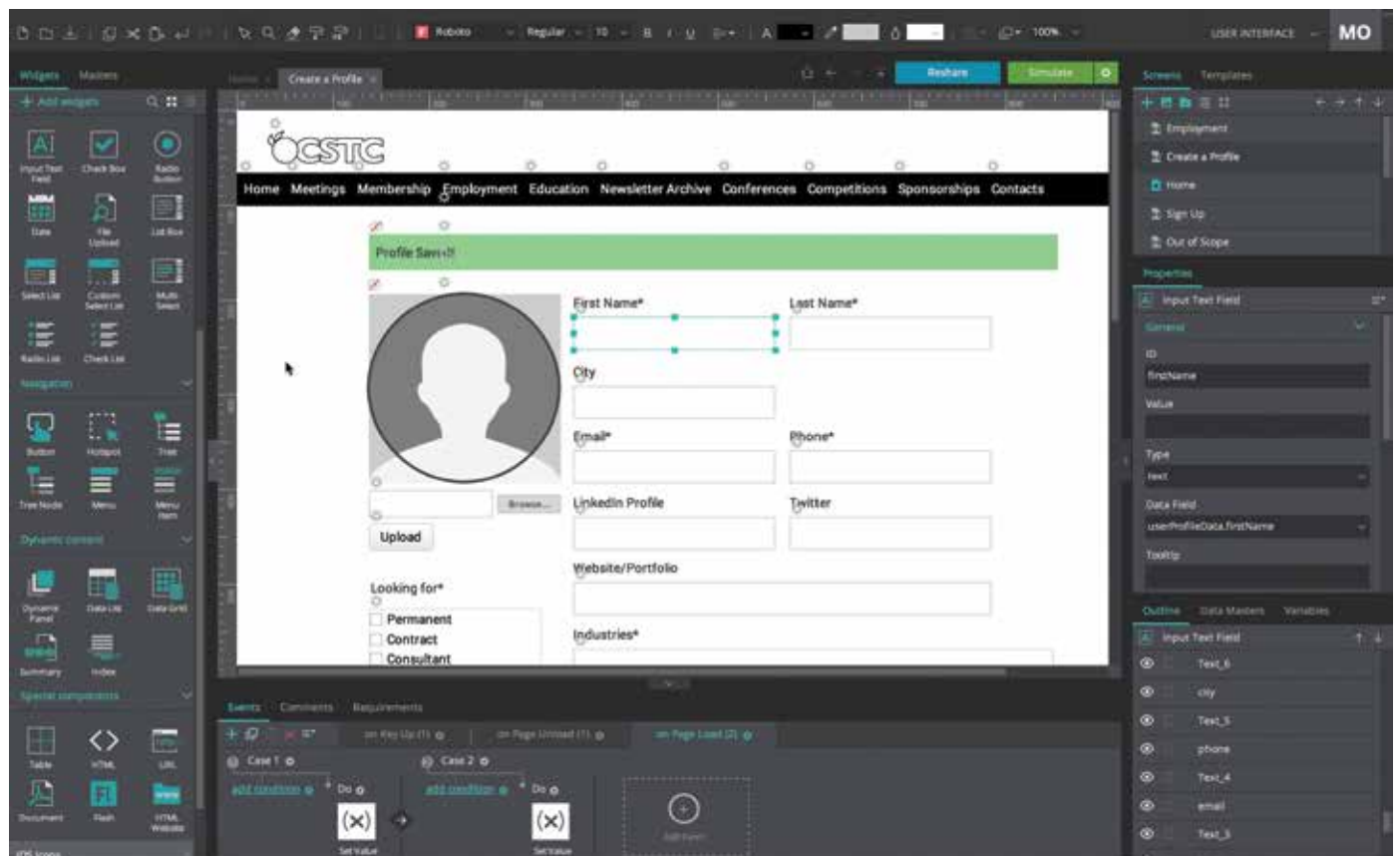


Figure 1. Justinmind Prototyper drag-and-drop interface.

prototypes make it clear that the real intent of the software is to create light-weight, proof-of-concept prototypes for presentations where the user input is predictable.

## What's Great about Justinmind

Justinmind's drag-and-drop interface enables you to quickly drop interface elements, called widgets, onto the canvas to create a wireframe in minutes. The robust library of interface elements means you won't spend a lot of time developing your own from scratch. If you need to add interactivity, drag actions to elements on the canvas to make those elements functional. Additionally, Justinmind makes it easy to create a small database for use with your prototype.

## Drag-and-Drop Interface Elements and Actions

Build a prototype quickly by dragging interface elements from the widget library to the canvas. Need a link to take the user to another page in the site? Simply drag that page to the menu or button, and the navigation action is created (see Figure 3).

## Widgets and Interface Elements

Justinmind has a robust library of interface controls, or widgets, right out of the box. The basic library includes a vast array of text boxes and buttons and more advanced controls, like a file-upload element and a date picker. You can build navigation menus, navigation trees, radio lists, and more. If you're developing a mobile app, you can download additional widget libraries for controls specific to iOS and Android interfaces.

## Dynamic Content

Your interface isn't going to be much use for testing if your participants don't have any data to work with. Justinmind's Data Master feature enables you to create a small database for your users to interact with. This is a handy feature for test scenarios that require looking up, creating, or editing data records (see Figure 4).

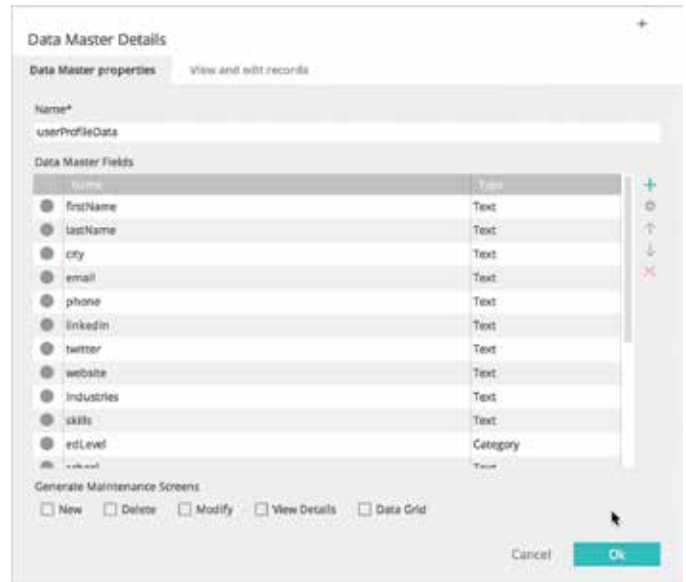


Figure 4. Justinmind enables you to create a small database so participants can interact with data in the prototype.

Want to create a form quickly? Drag your completed data master to the canvas, and Justinmind will create the

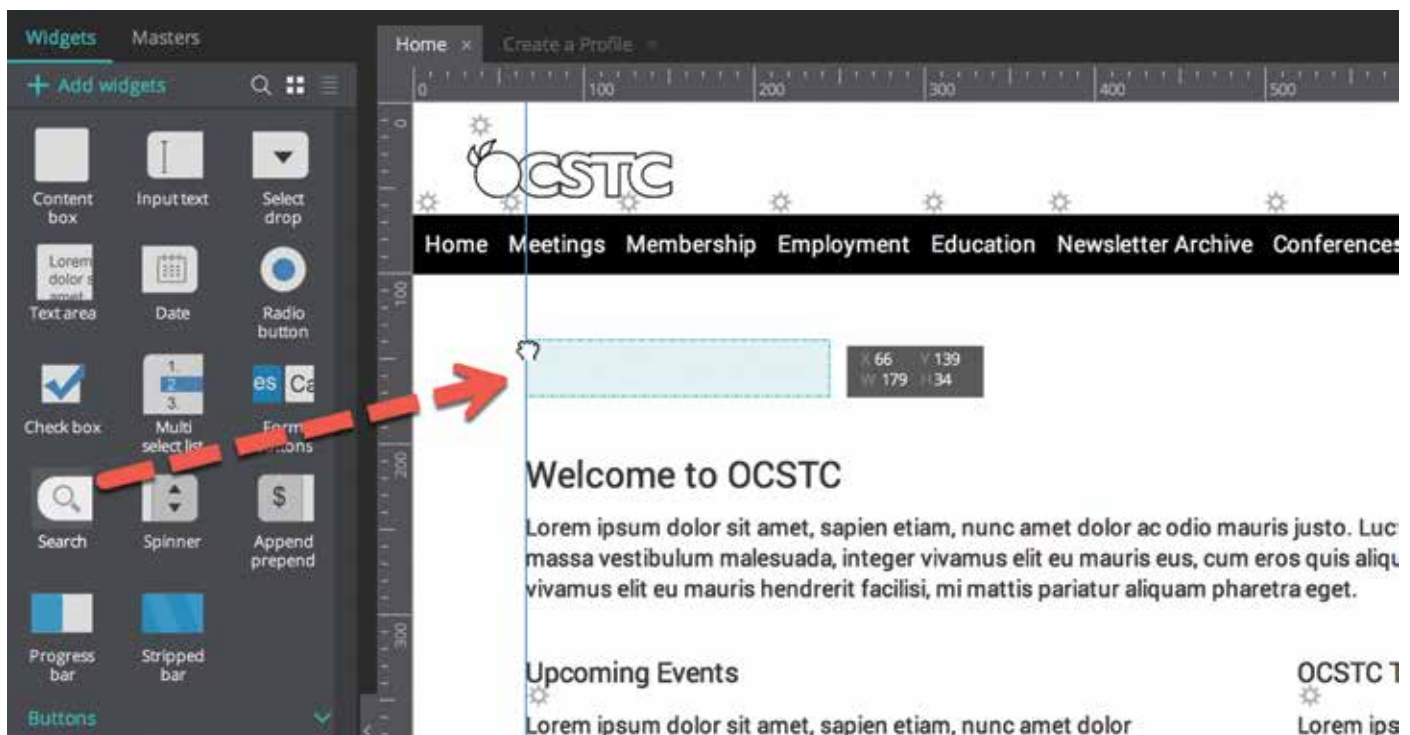


Figure 3. Drag items from the library to place them on the canvas. In this case, the Search element is dropped onto the canvas.

form for you. If you have an existing form, drag elements from the Data Master list to the fields in your interface to connect the fields to the database.

If you have a small amount of data for users to enter, you might dispense with the Data Master and use the Variable feature instead. Variables ensure that data entered on one page will appear on another page. However, this is only feasible if your user will interact with only one record and not have to add, modify, or delete any records. For example, users editing a personal profile page would only edit their own data, so this situation could rely on variables.

## What Could Be Better

While Justinmind makes it easy to build an interface and add basic actions and data elements quickly, some basic functions aren't easy to program, some controls don't behave as expected, and there are unknown limits to the data that can be entered.

### Tab Order

Justinmind's Prototyper doesn't give you the ability to set the tab order globally or on a page by page basis. By this I mean that you can't click the fields to set the order in which the fields come into focus as the user presses the Tab key. In fact, by default, pressing the Tab key in the simulation does nothing. This is striking, because tabbing through forms is a common action among users. To enable the Tab key to move the focus from field to field, you must add an action to each field individually—a time-consuming process for long forms (see Figure 5).

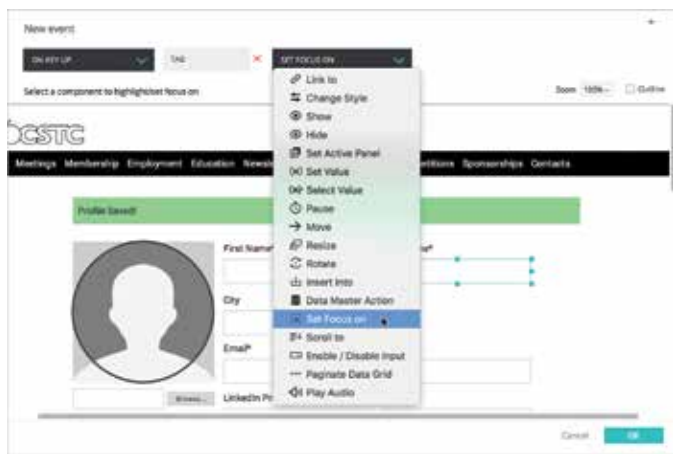


Figure 5. You must add an action to each field and select where you want to set the focus when the user presses the Tab key.

## Controls Don't Always Play Nice with Data

Text variables retain the information entered in text boxes; however, selection controls, like drop-down menus, radio lists, check lists, and list boxes don't retain their selections by default. This is problematic when your test scenarios include tasks that require the user to return to the form to edit their data. You can add some complicated actions to some of the controls so they display the user's selections

when the page loads again, but some controls, like the drop-down menu, don't seem to respond to these actions. These situations tend to cause confusion among participants ("Hmmm. I thought I selected that,"), which indicates that they might feel they did something wrong.

## Unknown or Unexplained Limitations

The Justinmind Prototyper seems to enforce some undocumented limitations on data entry. The text area box does not wrap text to the next line when participants reach the width of the text frame. Instead, participants see only one long line of text. The text area box also does not accept more than 255 characters. This is a concern during the testing, because participants comment on not being able to enter additional data. This limitation, which reminds participants that they are working only in a prototype, seems to disengage the participants momentarily from the task at hand and lowers their satisfaction with the prototype.

## Conclusions and Recommendations

A friend recommended Justinmind Prototyper to me recently when I mentioned that I was building a "paper" prototype using hidden layers and form fields in Adobe Acrobat. On his recommendation, I signed up for a 30-day trial and downloaded the software. I'm glad that I did. Even after taking the time to learn this new software, I believe I built the working prototype much more quickly—and with a better result—than I would have by repurposing a familiar tool to build it.

Having said that, the shortcomings mentioned in this article all arose during my usability testing. These quirks in the prototype caused user confusion and required the evaluator to intervene and explain that these issues were limitations in the prototype. These issues took the users' focus away from the tasks and drew their attention to the prototype itself, likely causing some variation in the click-rate and time-on-task data collected. The errors heightened users' perceptions of errors in the product and lowered overall satisfaction.

I recommend the Justinmind Prototyper for usability tests where participants will interact with limited amounts of data. Justinmind will help you to develop a highly interactive, data-driven prototype quickly; however, you'll need to beta test your prototype rigorously prior to conducting your evaluation. **i**

MICHAEL OPSTEEGH (michael.opsteegh@eyefinity.com) has been a technical writer in the software and financial services industries since 2004. He is currently a Senior Technical Writer for Eyefinity, which supports eyecare professionals with industry-leading software and services. He's a Lecturer in the professional writing program at California State University, Long Beach, and a PhD student at Texas Tech University. He holds a master's degree in English and a certificate in technical and professional communication. He is also a Certified Technical Professional Communicator (CPTC).