Ilya Levin

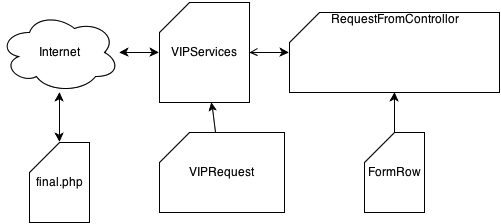
Final Project

1. Design (this is basically beefing up your project proposal)

The VIP app allows a consumer to get on a “VIP List” when going to a venue such as a dance club. The application allows the consumer to let a promoter know the name of the person, how many guests do they have, and whether or not they would like a VIP table.

The promoter will then receive an email with the VIP request and will work with the venue to accommodate the request. Then the promoter can contact the consumer and let them know whether they were able to fulfill their request or not.

1. Implementation notes: specifically:  
   - Which APIs/Frameworks did you use - either OS supplied or -*especially important* - 3rd party (e.g. cocos2d, etc)  
   - Documentation of all classes uses, preferably with a class hierarchy or UML(optional) diagram. (UML is optional, diagram isnt')



Frameworks used:

Grand Central Dispatch

RequestFromControllor – This class is responsible for controlling the user interface and interaction with the VIPService.

FormRow – This class inherits from UITableViewCell. This class simply hides the touch keyboard should the user touch anywhere inside a UITableViewCell.

VIPService – This class is responsible for the interaction between the RequestFromController(the UI) and final.php page (the backend) which is located on the internet.

VIPRequest – This is the data model of the VIP request. It is used to store the values of a VIP request when the user fills out the form.

Final.php – This is a php script that accepts a “VIP” request through URL parameters and sends an email to the promoter.

1. **Use cases you've tested - this is super important**

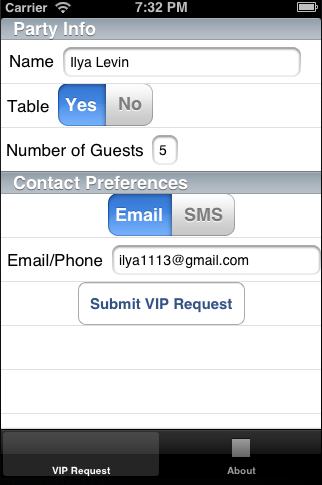
|  |  |
| --- | --- |
| **Use Case** | **Expected Results** |
| A user clicks an input field then clicks on a FormRow (UITableViewCell). | Keyboard should hide after a tap in a FormRow. |
| Fill out form and submit | Should get an alert that message has been sent. Then you should receive an email in your inbox. See below on how to config the app to send to your email. |
| Fill out form and submit without internet connection. Note: The way I simulated this is by blocking all requests from the simulator in my firewall settings. | Should get an alert there is no connection. |
| Fill out an invalid email in the contact section | Should receive an error message. |

1. **quick and dirty user's guide. Screenshots welcome.**

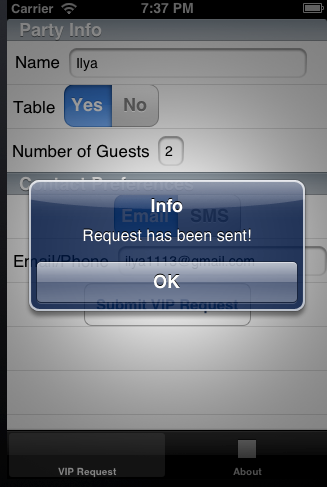
**How to config app**

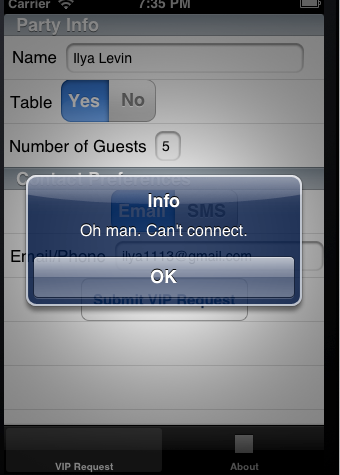
1. Open config.plist
2. Change email address to your own
3. Save and deploy a new build

1) Fill out request

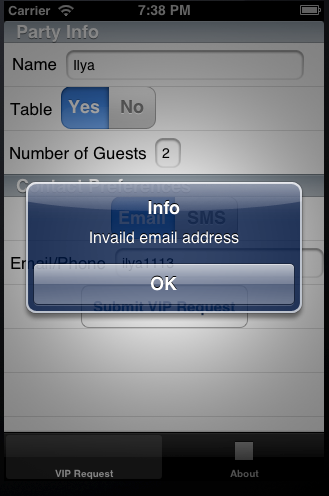


2) Wait for the response back from the server

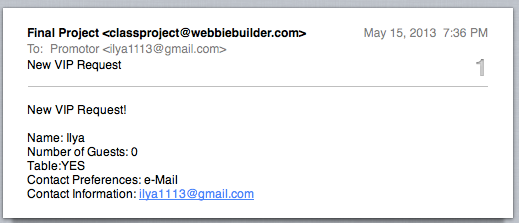




Email address validation:



Once a form is filled out you should get an email in the email address that you setup in the plist.



1. Specific challenges faced, if any, as well as workarounds/solutions used

Trying to understand how Grand Central Dispatch works was a bit difficult but extremely useful.

In the user interface I used a static UITableView, but it took me a while to figure out how to get to the data that the user filled in.

Learning how to store objects is very complex and lots of different ways. Spent a lot of time trying to figure out how to store the last 10 objects an array and save it to a file.

1. Responsibilities of each team member (unless you're a 1-man team, obviously)

Ilya Levin– Built all the classes and php script

Cheralathan N.T – QA and unit testing

1. What you've learned from all this, and**your** overall impression (yes, we want your feedback) of the final project.

Overall I learned a lot about iOS and am much more familiar with it then ever before. I was also able to practice my software engineering skills and I hope I did a good job at abstraction. Learning how to create my own delegate, Grand Central Dispatch, and the static table view was very useful and interesting to me. I also got familiar with BitBucket and Git as a source control management in xCode.