

Sechi Mobile PoC – iOS - Deployment Guide

Revision History

Author	Revision Number	Date
Karol Szafrński	1	16.05.2014

Deployment Instructions	3
1. Organization of Submission	3
2. Application Setup	3
2.1 Ant .Properties File	3
2.2 Build Scripts Setup	3
3. Database Setup	3
4. Deployment Instructions	3
5. Starting	3
6. Verification	3
7. Resource Contact List	3

Deployment Instructions

1. Organization of Submission

All sources are in root directory of archive. Application is using RestKit library, and it's headers must be available to include for Sechi application.

All images are available in sechi/Resources folder.

Configuration file is sechi/values.plist.

2. Application Setup

RestKit library is already included. Nothing more is needed.

3. Ant .Properties File

Property Name	File property is stored	Example Setting
SyncInterval	Sechi/values.plist	10
APIUrl	Sechi/values.plist	http://my-heroku-sechi.herokuapp.com/

3.1. Build Scripts Setup

No special scripts were used here.

4. Database Setup

Application is using CoreData to persist data when offline. No special setup needed.

5. Deployment Instructions

Just build and run in xcode needed.

6. Starting

Just build and run in xcode needed.

7. Verification

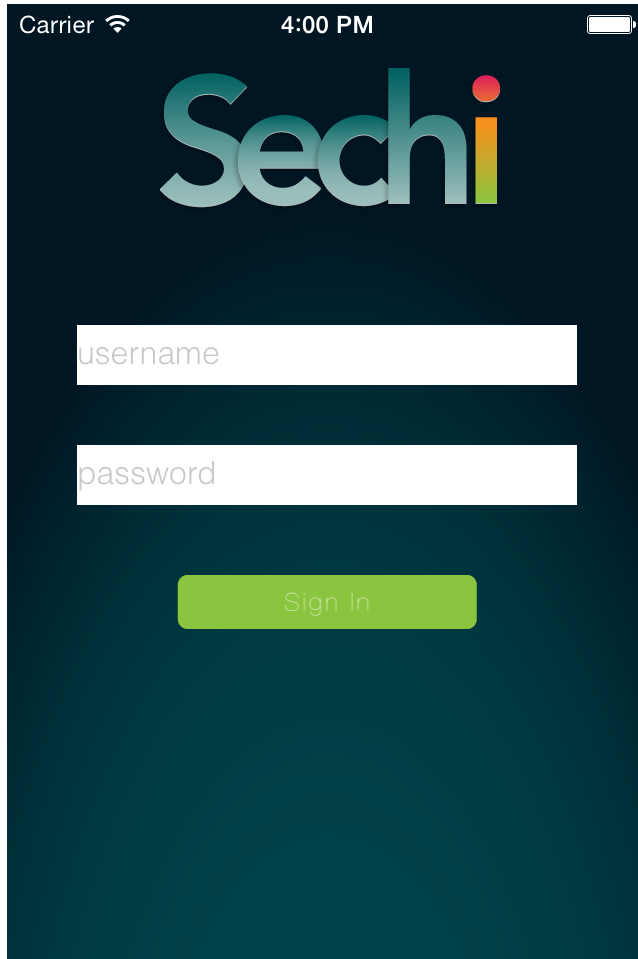
Sync is performed for each kind of object after clicking button in main menu. **Important** because of API methods missing for update and save a new objects. All changes are lost after next sync. Someone wrote about it in comments. That's why it was probably better to leave the sync performed after pressing signup button, but it's changed here, as it was marked in 2.2.2, sync will overwrite existing data. Normally changes are send to the server first, than new rows are downloaded. But that workflow is broken here, cause of missing api methods.

App is displaying objects that `isdeleted` field is other than 'true'. That's the only reason why some objects

are not visible in app.

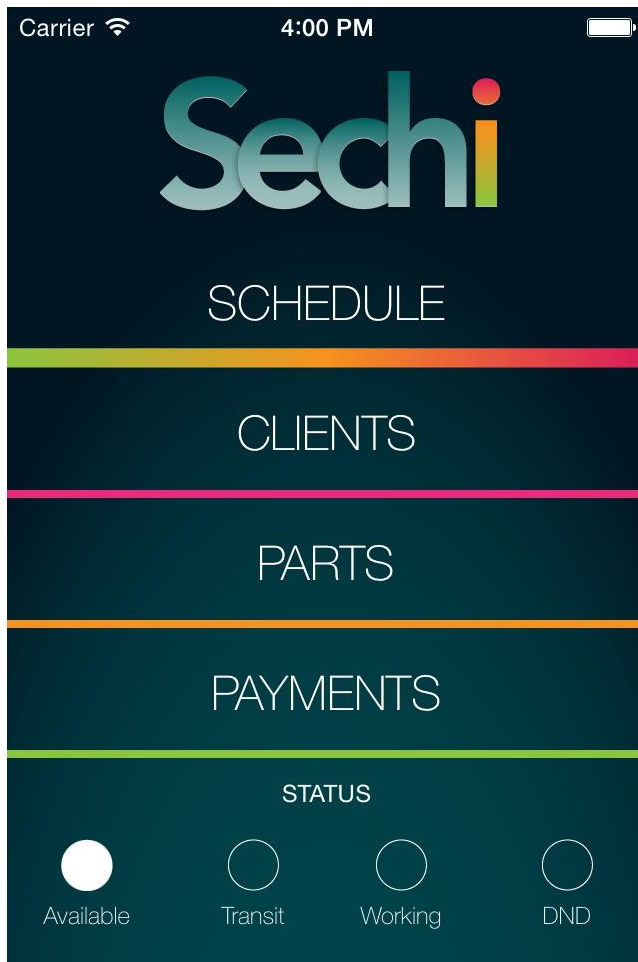
7.1 Sign in screen

- You can only press sign in button here. This view is only used to show how login process can work.



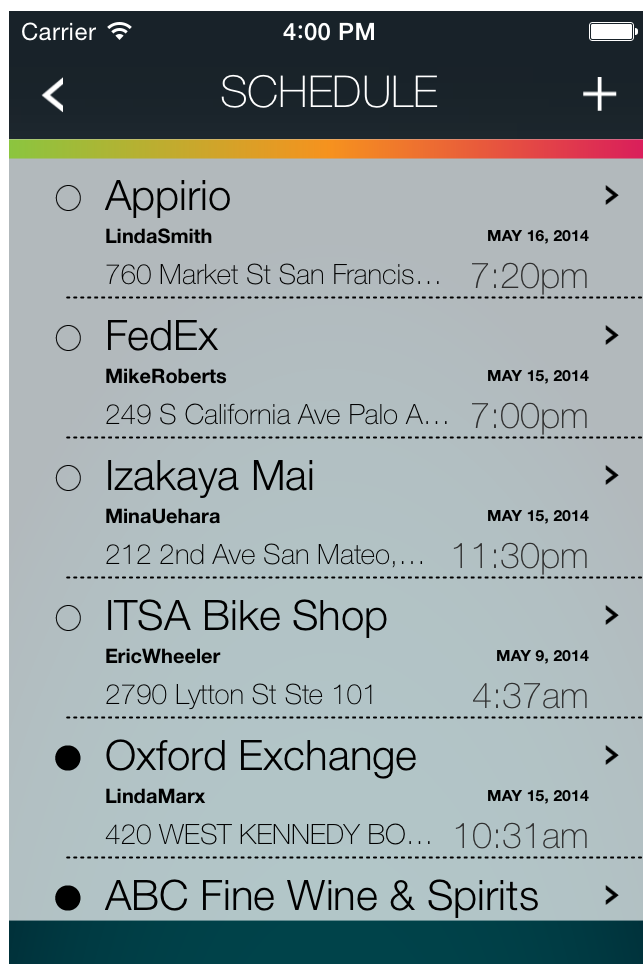
7.2 Main menu screen

- Select menu position to view object's list
- Change current status - it's saved and restored after app relaunch



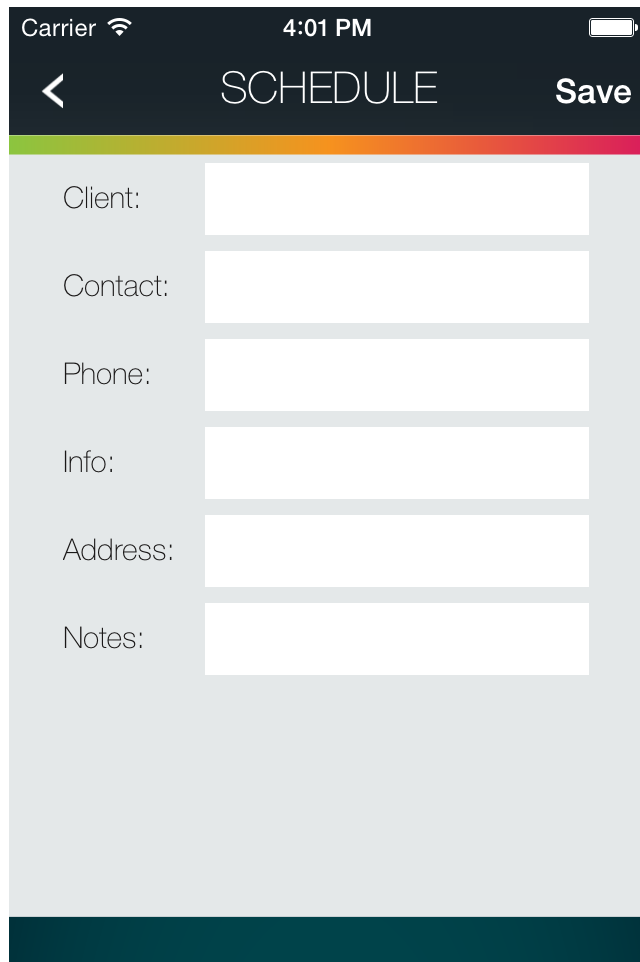
7.3 Schedule list

- Circle at the left indicates job status. It's filled black when completed or empty otherwise.
- Swipe left to show delete button. Confirmation needed.
- Tap plus button to add a new row.
- Rows visible here are downloaded from API and refreshed every time the view is loaded (view opened from main menu)



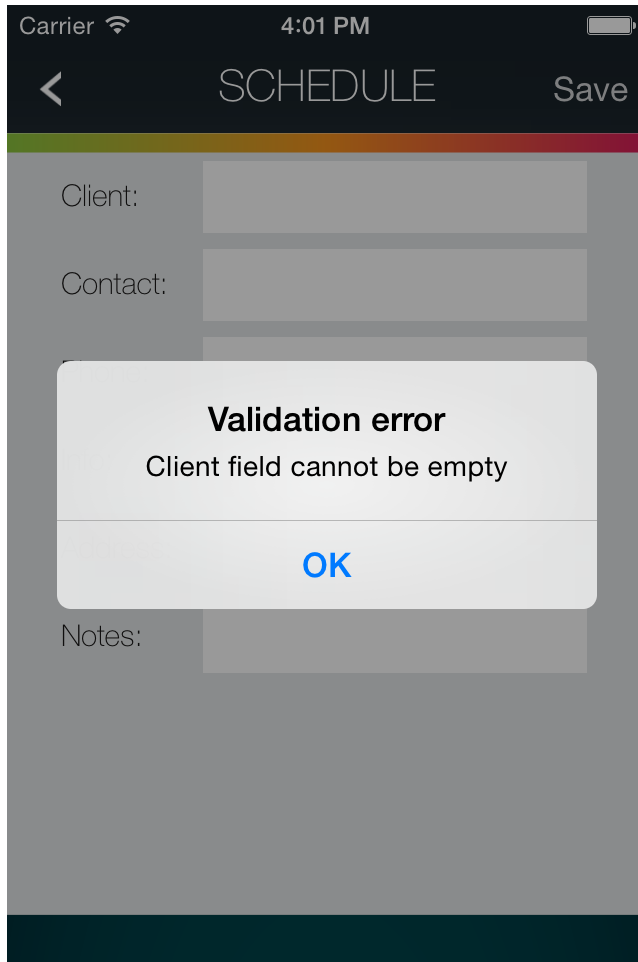
7.4 Add new job

- You can add a new job on this screen.
- If any of the text fields will be empty there will be a validation error and job wont be saved.



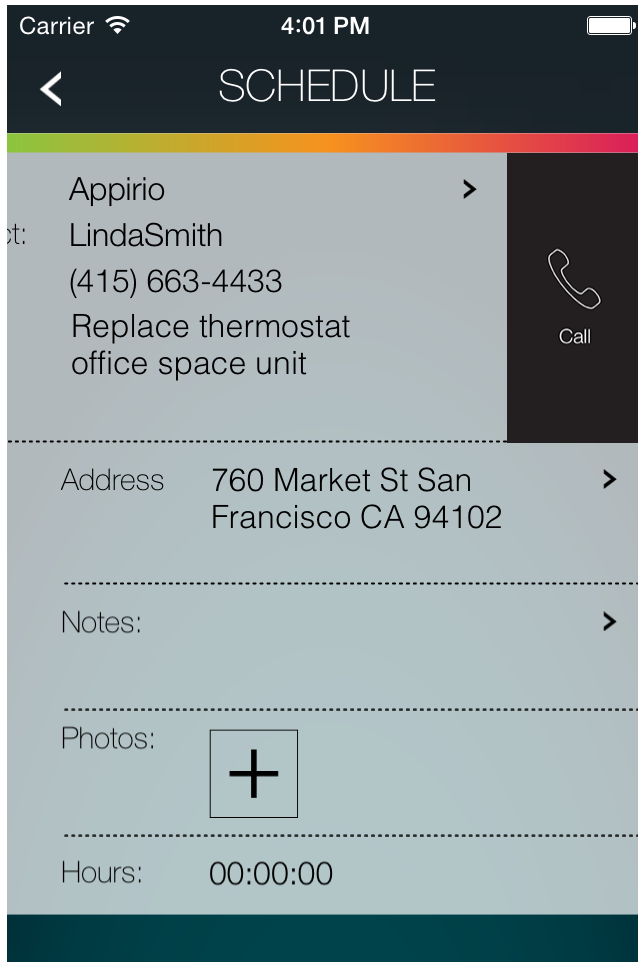
A screenshot of a mobile application interface for a 'SCHEDULE' form. The top status bar shows 'Carrier' with a signal icon, '4:01 PM', and a battery icon. Below this is a dark header bar with a back arrow, the title 'SCHEDULE', and a 'Save' button. A horizontal rainbow-colored bar separates the header from the form fields. The form consists of six white input fields with light gray borders, each preceded by a label: 'Client:', 'Contact:', 'Phone:', 'Info:', 'Address:', and 'Notes:'. The bottom of the screen is a solid dark teal bar.

Client:	<input type="text"/>
Contact:	<input type="text"/>
Phone:	<input type="text"/>
Info:	<input type="text"/>
Address:	<input type="text"/>
Notes:	<input type="text"/>



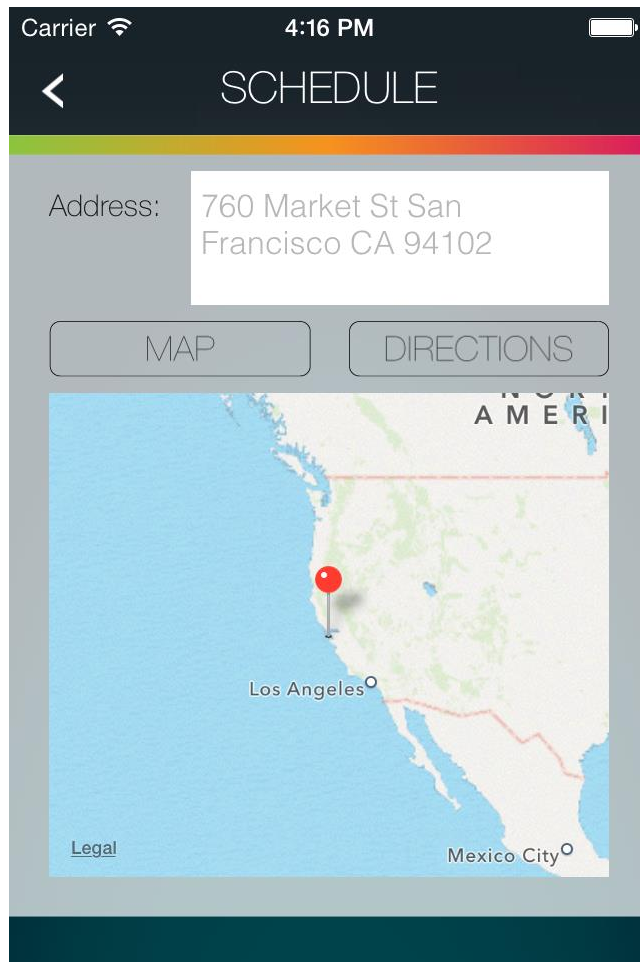
7.5 Single schedule view

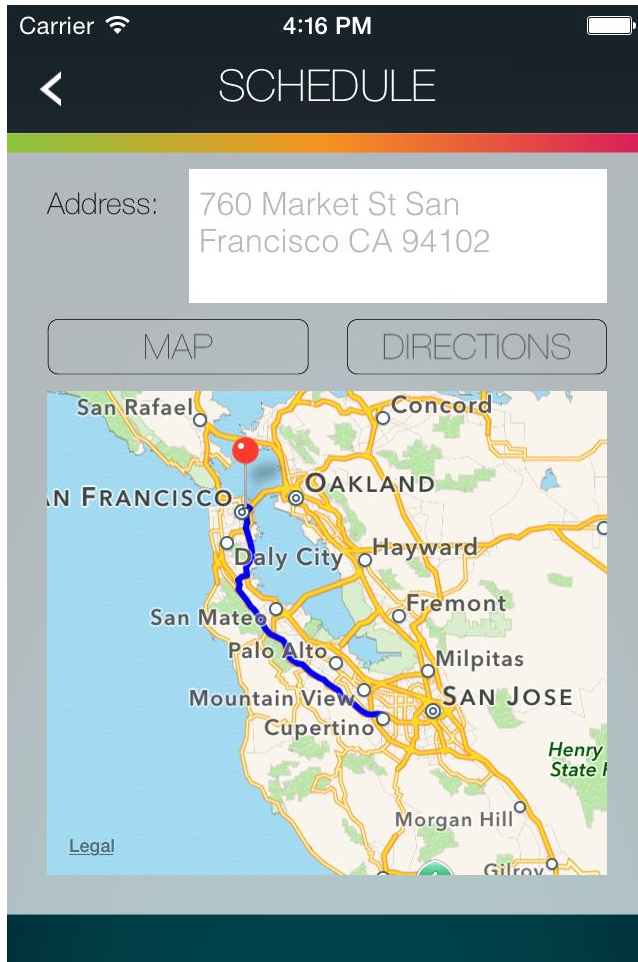
- Swipe first cell to show call button, after tapping it will show phone dialer
- Swipe second cell to show map button, it will display 7.6 view.
- Tap any of the first three cells to show edit form. Any text field cannot be left empty. Only exception is when field was empty in response from API and you didn't change it.
- You can also add a photo here from library or camera, by tapping plus button near to the "Photos" label.
- Start and Complete buttons are corresponding to hours label in a way described in topcoder forum thread: <http://apps.topcoder.com/forums/?module=Thread&threadID=819004>



7.6 Job address view

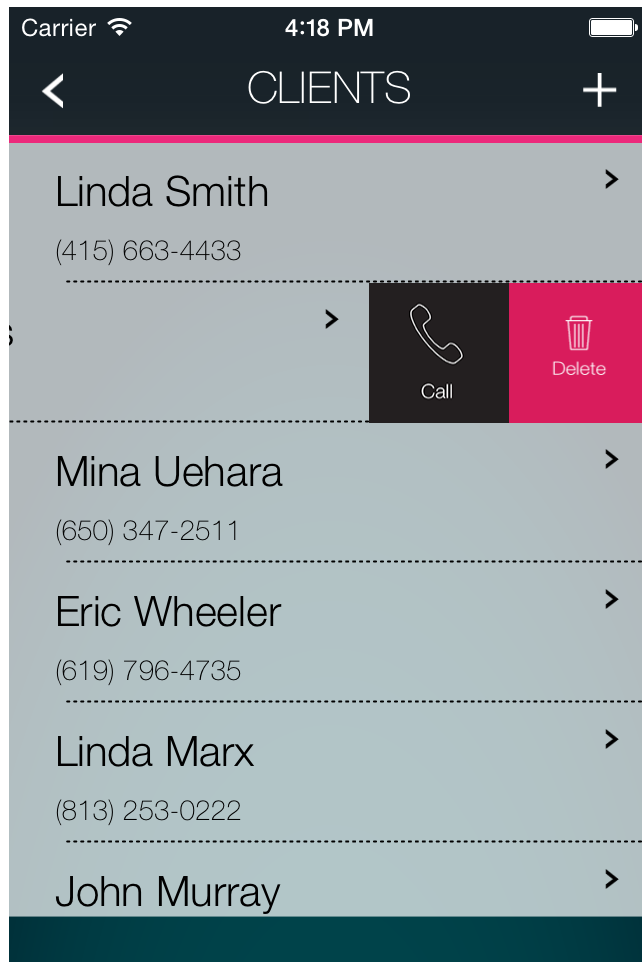
- This view will show address from text field on map, and after pressing directions button there will be also visible a route from current location to pinned point.
- If you change the address, it will be refreshed on map after pressing "Map" button.
- Any changes you make here are not saved! To edit address just tap on it on previous screen - do not swipe.
- Location access for this app must be enabled in settings.





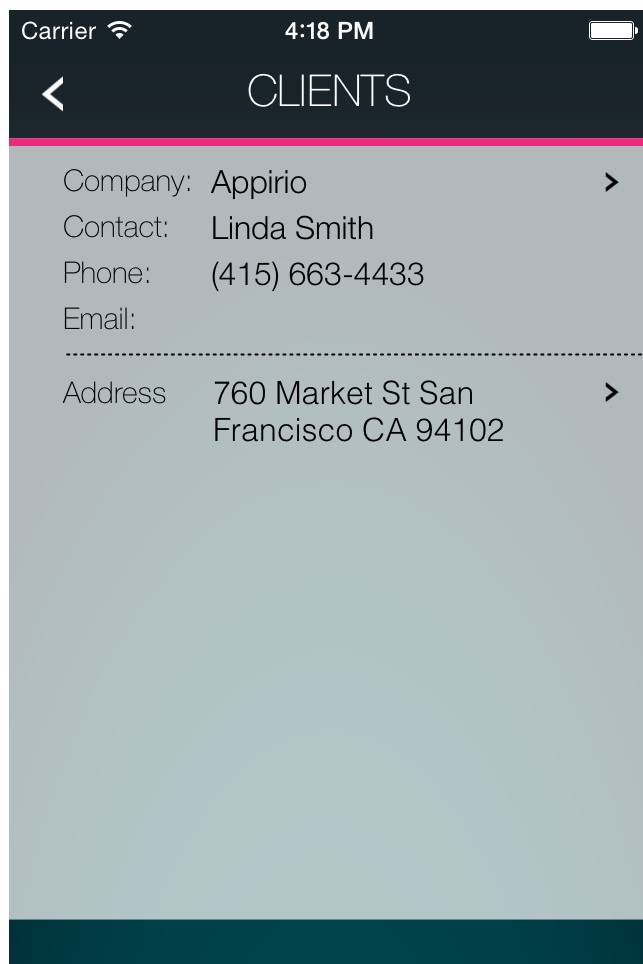
7.7 Clients list view

- View available after selecting "Clients" from main menu.
- After swiping on the cell there's available option to call or delete a client.
- Rows visible here are downloaded from API and refreshed every time the view is loaded (view opened from main menu)
- Tap plus button to add a new client. Every text field has to be filled, otherwise there will be a validation error.



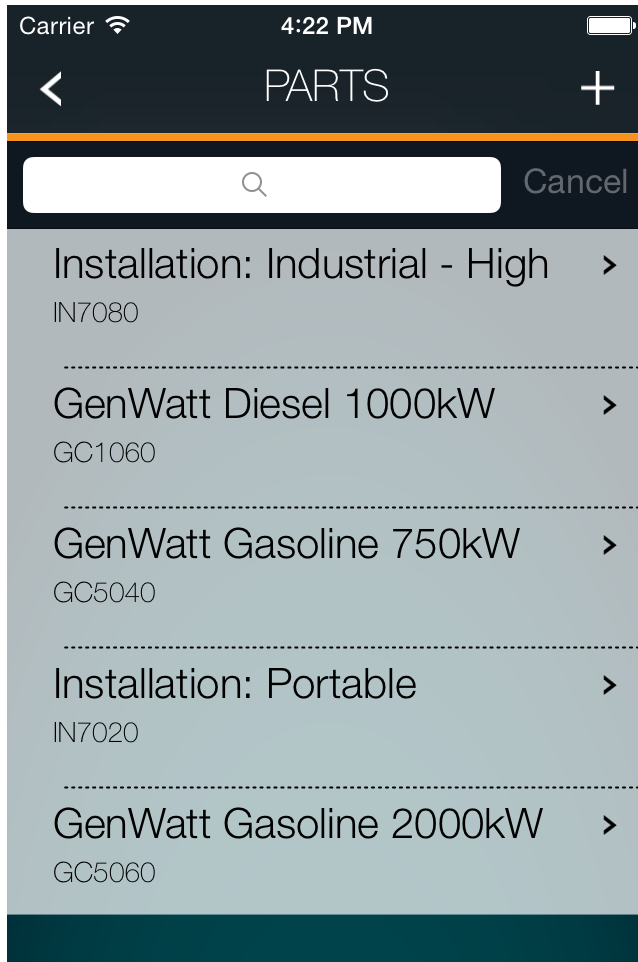
7.8 Single client view

- Tap any cell to edit it's content. Any text field cannot be left empty. Only exception is when field was empty in response from API and you didn't change it.
- Swipe first cell to show call button which displays a dialer if number is properly formatted.
- Swipe second cell to display map view working similarly to the 7.6 Job address view



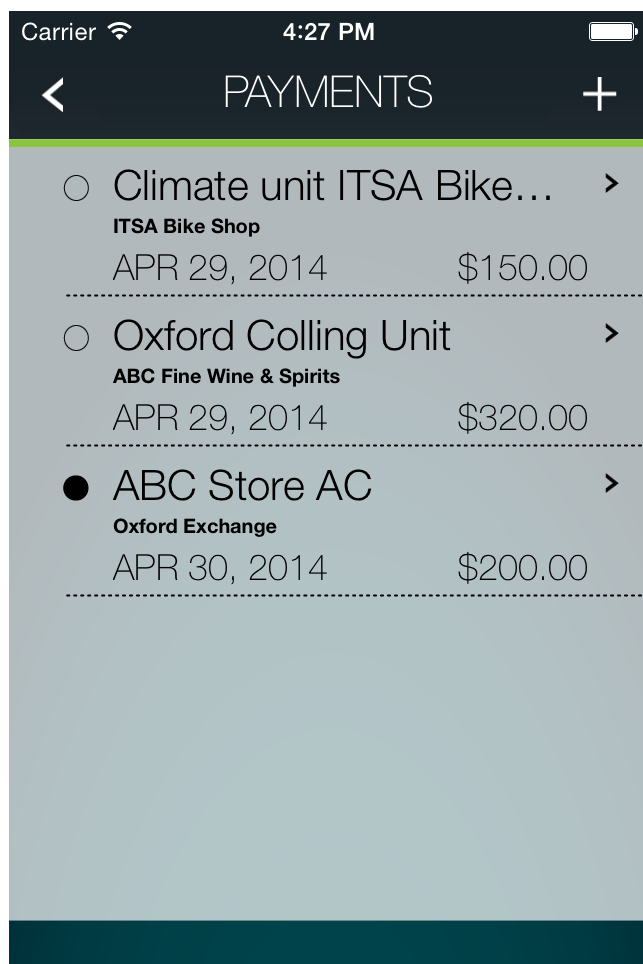
7.9 Parts list view

- View visible after selecting "Parts" from main menu.
- Rows visible here are downloaded from API and refreshed every time the view is loaded (view opened from main menu)
- After changing contents of UISearchBar text field there's api call that in future will return filtered rows. UISearchBar is visible after dragging table down.
- Tap plus button to display barcode scanner. Code 39 type codes are supported. When code is read it's value is pushed to UISearchBar. To cancel scanning process just tap on the screen.
- Nothing more can be done here. You can only browse the list.



7.10 Payments list view

- Visible after selecting "Payments" from main menu
- Circle on the left indicates if payment status is completed or not.
- Rows visible here are downloaded from API and refreshed every time the view is loaded (view opened from main menu)
- Tap plus button to add a new payment. Every text field has to be filled, otherwise there will be a validation error.
- Swipe any cell left to show delete button. Confirmation needed before deleting.
- Tap any cell to go to 7.11 view.



7.11 Single payment view

- Tap a cell to edit its content. Any text field cannot be left empty. Only exception is when field was empty in response from API and you didn't change it.
- Tap complete button to change payment status. If payment status is completed this button will not be visible.

The screenshot shows a mobile application interface for a payment screen. At the top, the status bar displays 'Carrier' with a signal icon, the time '4:27 PM', and a battery icon. Below this is a dark header bar with a back arrow on the left and the title 'PAYMENTS' in the center. A green horizontal line separates the header from the main content area. The main content area has a light gray background and contains the following fields: 'Amount' with the value '150' and a right-pointing chevron; 'Client:' with the value 'ITSA Bike Shop'; 'Job:' with the value 'Climate unit ITSA Bike S...'; 'Date:' with the value 'Apr 29, 2014'; and 'Notes:' which is empty. Below these fields is a rounded rectangular button with the text 'COMPLETE'. A dashed horizontal line is positioned below the button. The bottom of the screen features a dark teal footer bar.

8. Resource Contact List

Name	Resource Email
Karol Szafrński	karolszafranski@gmail.com