

Utah State University

DigitalCommons@USU

---

All Graduate Plan B and other Reports

Graduate Studies

---

5-2014

## A Customer Data Acquisition Application for Android

Richard William Preece  
*Utah State University*

Follow this and additional works at: <https://digitalcommons.usu.edu/gradreports>



Part of the [Computer Sciences Commons](#)

---

### Recommended Citation

Preece, Richard William, "A Customer Data Acquisition Application for Android" (2014). *All Graduate Plan B and other Reports*. 397.

<https://digitalcommons.usu.edu/gradreports/397>

This Report is brought to you for free and open access by the Graduate Studies at DigitalCommons@USU. It has been accepted for inclusion in All Graduate Plan B and other Reports by an authorized administrator of DigitalCommons@USU. For more information, please contact [digitalcommons@usu.edu](mailto:digitalcommons@usu.edu).



A CUSTOMER DATA ACQUISITION APPLICATION FOR ANDROID

by

Richard William Preece

A report submitted in partial fulfillment  
of the degree requirements for the degree

of

MASTER OF SCIENCE

in

Computer Science

Approved:

---

Dr. Vladimir Kulyukin  
Major Professor

---

Dr. Daniel Watson  
Committee Member

---

Dr. Nicholas Flann  
Committee Member

UTAH STATE UNIVERSITY  
Logan, Utah

2014

Copyright © Richard Preece 2014  
All Rights Reserved

## **Abstract**

This report discusses the development process and walkthrough of an application that is intended to aid businesses with two issues that they face on a consistent basis: customer data acquisition and eliminating expiring inventory.

In developing an application that can quickly and effortlessly acquire contact information from a customer, businesses can continue to offer deals that are of interest to customers. This can help to use inventory that would otherwise go to waste, as well as increase customer engagement by providing incentives to become a repeat customer.

## Acknowledgements

First and foremost, I would like to thank my wife, Lauren, and my daughter, Claire, for the love and support they have offered throughout my education at Utah State University. I would also like to thank all other family that has helped me through my years as a student. Your support was invaluable to me and my success as a student.

I would like to thank Dr. Vladimir Kulyukin for the many hours spent working with me to reach the necessary steps to achieve this goal as well as the classes that I took from him. I could not have accomplished this work without him.

Many thanks are needed for my committee members Dr. Dan Watson and Dr. Nick Flann. Their input and guidance was most appreciated on the work presented in this report.

Last of all I would like to thank Utah State University for the opportunity to attend school at such an excellent institution.

Richard Preece

## Table of Contents

<b>Abstract .....</b>	<b>iii</b>
<b>Acknowledgements .....</b>	<b>iv</b>
<b>Table of Contents .....</b>	<b>v</b>
<b>Table of Tables .....</b>	<b>vii</b>
<b>Table of Figures .....</b>	<b>viii</b>
<b>Chapter 1 .....</b>	<b>1</b>
<b>Introduction.....</b>	<b>1</b>
<b>Chapter 2 .....</b>	<b>4</b>
<b>Other Work .....</b>	<b>4</b>
Introduction .....	4
Why Android? .....	4
Previous Android Application.....	5
Concurrent iOS Application .....	5
<b>Chapter 3 .....</b>	<b>7</b>
<b>Related Work .....</b>	<b>7</b>
Introduction .....	7
The Ottawa Hospital.....	7
Kony.....	8
Malaria Control.....	8
Cloud Computing .....	8
Business Intelligence and Customer Data Analytics .....	9
Cross Platform Consistency.....	11
<b>Chapter 4.....</b>	<b>13</b>
<b>Application Development.....</b>	<b>13</b>
Hardware Used.....	13
Software Used.....	14
Use Cases.....	15
Business Log In.....	15
Business Log Out.....	17
Customer Check-in.....	18
Access Settings.....	20
View Database Contents.....	21
View and Modify Catchall Group.....	22
Manually Sync Data with Server .....	23
View and Modify Sync Settings .....	24
Delete Database Contents .....	25
Pick Winner.....	26
<b>Application Walkthrough .....</b>	<b>28</b>
Assumptions.....	28
Business Login Page .....	28
Business Selection Page .....	29
Customer Check-in Screen.....	31
Local Administration Options .....	38

<b>Database Schema</b> .....	<b>49</b>
<b>Additional Features</b> .....	<b>52</b>
Google Cloud Messaging .....	52
Rooted Devices .....	52
<b>Online Communications</b> .....	<b>54</b>
<b>Chapter 5</b> .....	<b>56</b>
<b>Summary</b> .....	<b>56</b>
<b>References</b> .....	<b>57</b>
<b>Appendices</b> .....	<b>59</b>
<b>Appendix A – Question and Answer with Jon Parrish</b> .....	<b>59</b>

## **Table of Tables**

Table 1 - Online Server Communications.....	54
---------------------------------------------	----



## Table of Figures

Figure 1 – Check-ins Per Day .....	10
Figure 2 - Offer Creation Page.....	10
Figure 3 - Nexus 10.....	13
Figure 4 - Galaxy Tab 2 7.0.....	14
Figure 5 - Business Log In Use Case Diagram.....	15
Figure 6 - Business Log Out Use Case Diagram.....	17
Figure 7 - Customer Check-in Use Case Diagram.....	18
Figure 8 - Access Settings Use Case Diagram.....	20
Figure 9 - View Database Contents Use Case Diagram .....	21
Figure 10 - View and Modify Catchall Group Use Case Diagram .....	22
Figure 11 - Manually Sync Data with Server Use Case Diagram .....	23
Figure 12 - View and Modify Sync Settings Use Case Diagram .....	24
Figure 13 - Delete Database Contents Use Case Diagram .....	25
Figure 14 - Pick Winner Use Case Diagram .....	26
Figure 15 - Initial Login Screen.....	28
Figure 16 - Login Screen With Auto-filled Credentials.....	29
Figure 17 - Business Selection .....	30
Figure 18 - Downloading Assets .....	31
Figure 19 - Main Check-in Screen.....	32
Figure 20 - Input Methods .....	33
Figure 21 - Phone Number Check-in .....	33
Figure 22 - Email Address Check-in.....	34
Figure 23 - An Invalid Phone Number .....	35
Figure 24 - A Valid Phone Number.....	35
Figure 25 - Validating Customer with Server .....	36
Figure 26 - Redemption Options.....	37
Figure 27 - Finishing a Check-in .....	38
Figure 28 - Default Settings Page .....	39
Figure 29 - Always store check-ins locally .....	40
Figure 30 - Require Login Password to Edit Settings .....	41
Figure 31 - Owner Password Required for Settings Access .....	42
Figure 32 - Allowing Device to Function Without Internet Connection.....	43
Figure 33 - Device Forced Into Offline Mode .....	44
Figure 34 - Data Viewer.....	44
Figure 35 - Catch-All Groups.....	45
Figure 36 - Sync Data .....	46
Figure 37 - Database Options .....	47
Figure 38 - Raffle Winners .....	48
Figure 39 - AES Schema .....	49
Figure 40 - Encrypted Schema .....	49
Figure 41 - User Schema.....	50
Figure 42 - Businesses Schema .....	50
Figure 43 - Offline Check-ins Schema .....	51

Figure 44 - Application Asking for Root Permission ..... 52  
Figure 45 - Rooted Device ..... 53

# Chapter 1

## Introduction

CityGro is a Salt Lake City based company that specializes in customer data acquisition for companies in 17 states and 3 countries. Originally, they were based in Cache County, Utah under the name BlueCache. Their business model was a subscription based discount card service where a customer added businesses that interested them, and in turn CityGro would email the customers when a business had special deals and offers.

In order to reach a larger audience, BlueCache shed their name, and became CityGro and moved their headquarters to Salt Lake City to expand to markets in the Salt Lake Valley. CityGro's business model began to change while an employee was speaking to a manager of a grocery store who had thrown away thousands of dollars in inventory. He was asked why he couldn't simply give the inventory away, to which the manager replied that he had no way to contact his customers.

Initially CityGro attempted to increase data acquisition by using methods such as texting in, loyalty cards, QR codes, and web based forms. They found that the adoption rate was lower than expected, and decided to try something new. This began the idea that a business could collect customer info such as phone numbers and email addresses at the point of check out. The idea was that the customer was most excited about the product at that moment, and they wanted to capitalize on the excitement felt by the customers. CityGro built an iOS

application to accomplish this, and both customers and businesses quickly began to embrace the idea.

They soon realized that they needed to add features and expand to Android to support their customer's needs, but CityGro did not employ anybody that was knowledgeable in Android development. At this point they decided to contact Utah State University to get help with the application. While the Android application was under development, a similar iOS application was being developed with corresponding features and interfaces. Both applications were designed to have a consistent look and feel while being familiar to users of both platforms.

Instead of selling products directly to the consumer, CityGro now places a tablet computer in a store and gives them access to web based analytics and tools for a monthly fee. This allows customers to check-into the device at the store by using personal information, and then allows the business to use that information as they see fit.

CityGro estimates that an individual business will see anywhere from five to 500 check-ins per day, with a typical business averaging 2,000 check-ins per year. CityGro also estimates that data acquisition is increased tenfold in businesses that employ their devices, and CityGro has received more than 450,000 check-ins to date.

The application developed and described in this paper has two main goals, first is to give a customer an easy way to provide contact information to a

business and receive benefits for doing so. The second is to make customer provided data easily accessible and usable by the business.

## Chapter 2

### Other Work

#### Introduction

This application is built on previous work from many people both within CityGro and in the world of Computer Science. CityGro has had a previous Android application as well as iOS applications. It also relies on work from people studying fields related to cloud computing, business intelligence and customer data analytics.

During the course of developing this application, considerations had to be made for cross platform development to keep the interfaces consistent while taking advantage of the native application abilities.

#### Why Android?

The original kiosk application for CityGro was written for the iOS operating system. CityGro realized the need to introduce an Android application for many reasons. First, Android has seen an intense adoption rate over the past few years.

- 2.8% market share in Q2 2009 (Canalys, 2009)
- 57 % market share in Q3 2011 (70% in the U.S.) (Canalys, 2011)
- 78.1% market share in Q4 2013 (IDC, 2014)

Second, Android manufacturers often offer cheaper devices than comparable iOS devices. Since CityGro provides the device for the business, cheaper devices provide larger profit margins.

Third, Android is backed by a large community of open source developers that aim to make the operating system smoother. This application is able to take advantage of the work of such developers when the device user has been granted root access. This allows for extra control of the device, and enabled more features described later.

Fourth, CityGro allows resellers in various locations to license their software. Some resellers requested the ability to use Android based software rather than software for other mobile operating systems.

### **Previous Android Application**

CityGro had created an Android application in the past, but this application was aimed at providing coupons to a customer's phone rather than being used as an in-store kiosk to gather customer data. These applications have since been removed from the application store and are no longer compatible with CityGro's servers.

The application saw more than 850 users through its lifespan, and even though it is no longer usable, it is still installed on more than 150 devices. This application allowed businesses to reach out to customers in basic ways to offer deals that may bring customers back to their store. It was eventually abandoned as CityGro moved towards placing devices in stores, rather than using devices owned by customers.

### **Concurrent iOS Application**

CityGro was in the beginning stages of developing an application for iOS that was intended to be the mirror of the Android application being developed.

This allows the same platform to be found on multiple operating systems across a wide variety of devices. Application features are kept in sync with each other to ease server side code complexity and reduce special cases for each operating system.



## **Chapter 3**

### **Related Work**

#### **Introduction**

Mobile applications play a role in many modern businesses. Use of mobile devices has skyrocketed over the past few years, and it is only expected to increase. It is estimated that by the year 2016 there will be approximately 126 million tablets, and 257 million smartphones in use in the United States (Schadler & McCarthy, 2012).

With a market that will grow to this size, it is obviously a prime area for growing your business. This chapter will discuss ways in which mobile computing has helped to increase customer engagement and improve a business's bottom line.

#### **The Ottawa Hospital**

The Ottawa Hospital worked with IBM to create a tablet based system for controlling their daily processes. The purpose was to take work out of the doctor's hands, and make it an intuitive process.

Similar to the application built for CityGro, The Ottawa Hospital's application works largely with a cloud to avoid keeping sensitive data on the device. The device intermittently erases data from the device so as to not hold customer information for long periods of time (IBM, 2013).

## **Kony**

Kony is another example of a company building cloud based mobile applications. Instead of focusing on medical applications, Kony has decided to focus on mobile banking. Through use of the application, it was determined that while the average user would check their bank account balance once or twice per day, those using mobile devices would check balances up to 5 times per day (Kony, 2013).

## **Malaria Control**

CityGro offers businesses the ability to communicate with customers in various ways such as emails and text messages. Text messaging has been proven to be a very effective means of communication. For instance, malaria control in Africa has long been an issue, due in part to a lack of communication with authorities.

With the spread of cellular networks in Africa, researchers found that text messaging was a cost effective way to give critical information to residents. They were able to give information such as treatment options, available medications, and post treatment information. Researchers found that such a method of communication was an effective way to spread information reliably (Zurovac, Talisuna, & Snow, 2012).

## **Cloud Computing**

This application takes advantage of cloud computing techniques. Cloud computing offers many advantages to relieve stress on low powered mobile devices. It offers cost effective means for storing and retrieving customer data,

while allowing scaling and maintainability. Cloud computing offers the ability to give massive amounts of data to a central location, and then let individual devices glean useful data when needed (Rao, Sasidhar, & Kumar, 2012).

Since no single device can, or is allowed, to hold all customer data and no mobile device is capable of performing analytics on the data, they instead all feed into a central server. These cloud capabilities have become essential, and the applications aren't able to properly function unless the cloud server regularly accessible.

### **Business Intelligence and Customer Data Analytics**

Lately, an area of interest in business has been analyzing customer data. Entire companies have been created and sold for billions of dollars that were solely based on customer analytics (Adobe Systems Incorporated, 2009). Businesses such as Google are interested in customer data analyzing to deliver content and ads appropriately. In fact, it is estimated that 97% of companies whose revenue exceeds \$100 million are engaged in some sort of business analytics.

With such a large market for customer data, and analytics of data, it is no wonder that CityGro has attempted to gain entry into this field. When data is uploaded to the server from a device, there are numerous analytics available to the companies within seconds. These analytics range from the number of times per day a customer has checked in, to the time of day.

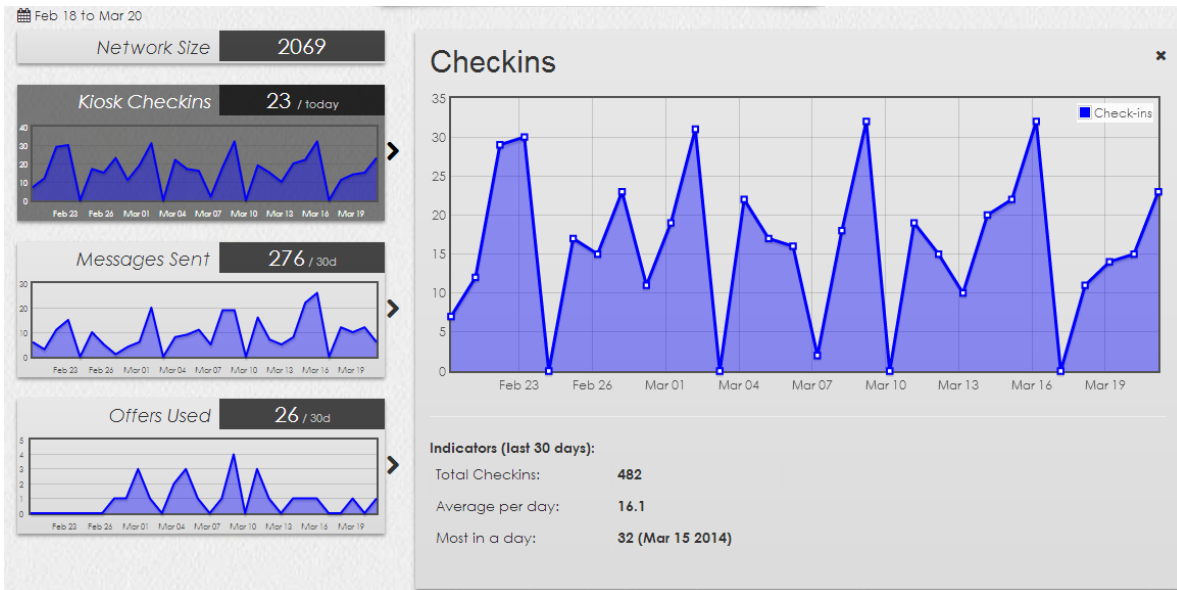


Figure 1 – Check-ins Per Day

They also offer data on enticement messages sent to customers, and the rate at which customers respond or opt out of messaging. They give companies the ability to customize the look and feel of the kiosk, as well as the process of requesting data from a customer. Here companies can create offers to entice customers to return to their store and purchase more products.

Name	Description	Kiosk	ID
Get 1 FREE (up to \$5 value) after purchase of 12	Get 1 cup FREE after you purchase 12. (up to \$5 value free)	Kiosk	ID: #148107
Buy 1 Get 1 FREE!!! Today only!	Buy 1 get 1 free! Equal or lesser value free.	Kiosk	ID: #319439
Today Only Offer	\$2.00 OFF ANY PURCHASE OF \$5.00 OR MORE.	Kiosk	ID: #319705

Figure 2 - Offer Creation Page

## **Cross Platform Consistency**

A difficult issue in any cross platform application is the consistency found in the application. There are guidelines that can be followed to make sure that applications look and feel similar to each other, but it is difficult to achieve (Heitkötter, Hanschke, & Majchrzak, 2013).

iOS offers an advantage to development because devices offered follow specific aspect ratios and sizes. This means that an application is guaranteed to look similar across devices. Android, however, does not have any specific guidelines for screen sizes and resolutions, and therefore support of devices is broken down into categories and developers are expected to ensure compatibility.

Cross platform consistency was achieved in two ways. First, the application was restricted to 7-inch and 10-inch devices. This fell into the “large” and “xlarge” categories of Android devices. It was much easier to guarantee a similar look and feel on these two devices by using density independent pixels to ensure that screen elements were similar sizes on various screen resolutions. The pixilation of assets was also avoided by making many elements on the screen native instead of images included in the application.

Second, the application uses native Java code for the bulk of its time. The application, however, takes advantage of web views for the times that customers are checking in and redeeming rewards. A web view allows the application to run html code in a native setting without opening a web browser. Since most modern

web browsers interpret markup languages similarly, it enables an identical experience between devices for those periods of time.

## Chapter 4

### Application Development

#### Hardware Used

The two main platforms that this application is aimed at are a 10-inch tablet, and a 7-inch tablet. All development was done on a 10-inch Nexus 10 and a 7-inch Galaxy Tab 2 7.0



Figure 3 - Nexus 10



Figure 4 - Galaxy Tab 2 7.0

## Software Used

This application was developed using Eclipse Indigo<sup>1</sup> with the Android Software Development Kit<sup>2</sup>. The Crashlytics<sup>3</sup> error reporting tool was also used to watch for any software issues that occur outside of development. This application is also only compatible with Android version 4.0+ (API level 16+).

---

<sup>1</sup> Eclipse Indigo can be found at <http://www.eclipse.org/indigo/>

<sup>2</sup> The Android Software Development Kit can be found at <https://developer.android.com/sdk/index.html?hl=sk>

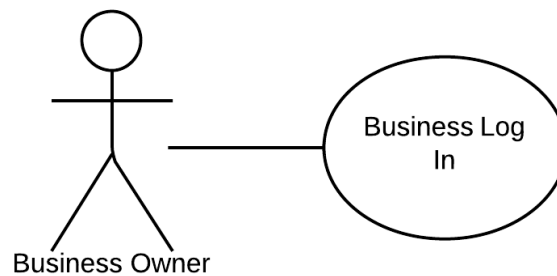
<sup>3</sup> Crashlytics can be found at <http://try.crashlytics.com/>



## Use Cases

This application is based on a specific set of use cases. The walkthrough of the application will show how each use case was implemented. They are as follows:

### ***Business Log In***



**Figure 5 - Business Log In Use Case Diagram**

#### PRIMARY ACTOR:

Business owner

#### GOAL:

To log in as a business and allow customers to then check-in

#### PRE-CONDITIONS:

The application must be running, with no other businesses logged in

#### NORMAL FLOW:

The business owner enters a username and password (See Figure 15)

Username and password are verified with a server, or with local data

The business owner is logged in, and may choose which business to use

(See Figure 17)

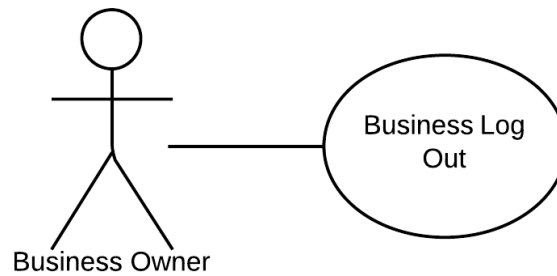
The application downloads assets (See Figure 18)

The application is ready for customers to check-in (See Figure 19)

#### EXCEPTIONS:

Business owner enters incorrect credentials

## ***Business Log Out***



**Figure 6 - Business Log Out Use Case Diagram**

### **PRIMARY ACTOR:**

Business owner

### **GOAL:**

To end the current logged in session on the device

### **PRE-CONDITIONS:**

The application must be logged in and waiting for a customer to check-in,  
or be on the settings page

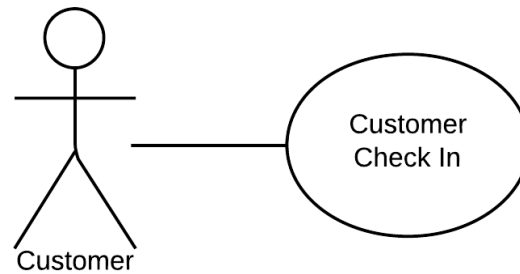
### **NORMAL FLOW:**

If on the check-in page, the business owner must perform the predefined  
gesture

If on the settings page, the business owner presses the Log Out button

(See Figure 28)

## **Customer Check-in**



**Figure 7 - Customer Check-in Use Case Diagram**

### PRIMARY ACTOR:

Customer

### GOAL:

To check-in to the business

### PRE-CONDITIONS:

The application must be logged in and waiting for a customer check-in

### NORMAL FLOW:

The customer enters the appropriate credentials (See Figure 24)

The customer presses the check-in button

The application retrieves any customer specific info from the server (See Figure 25)

The customer enters personal information, if needed

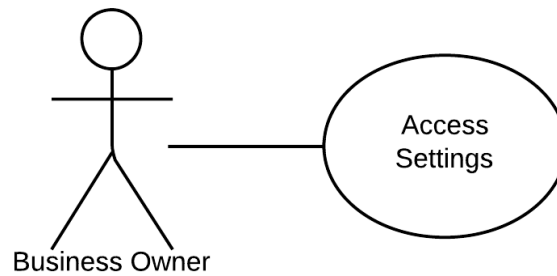
The customer redeems offers, if applicable (See Figure 26)

The customer finishes check-in (See Figure 27)

EXCEPTIONS:

The device is offline

## ***Access Settings***



**Figure 8 - Access Settings Use Case Diagram**

### **PRIMARY ACTOR:**

Business owner

### **GOAL:**

Access the settings page

### **PRE-CONDITIONS:**

The application must be logged in and waiting for a customer check-in

### **NORMAL FLOW:**

The business owner performs the predefined gesture

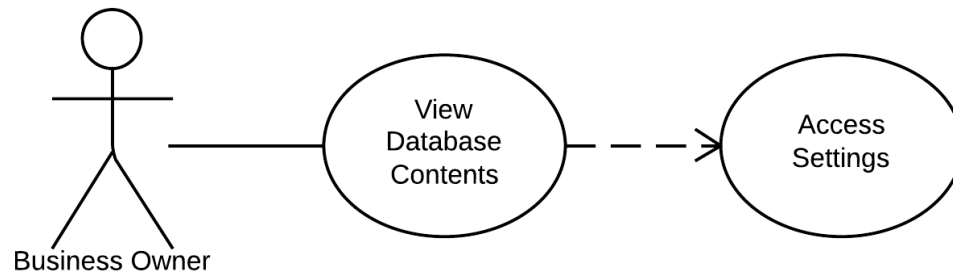
The business owner enters any required credentials

The settings page is displayed and the business owner may exit at any time (See Figure 28)

### **EXCEPTIONS:**

Incorrect credentials are entered

## ***View Database Contents***



**Figure 9 - View Database Contents Use Case Diagram**

### **PRIMARY ACTOR:**

Business owner

### **GOAL:**

View the data held in the local SQLite database

### **PRE-CONDITIONS:**

The settings page must be visible

### **NORMAL FLOW:**

The business owner presses the "View Data" button

Data is shown to the user (See Figure 34)

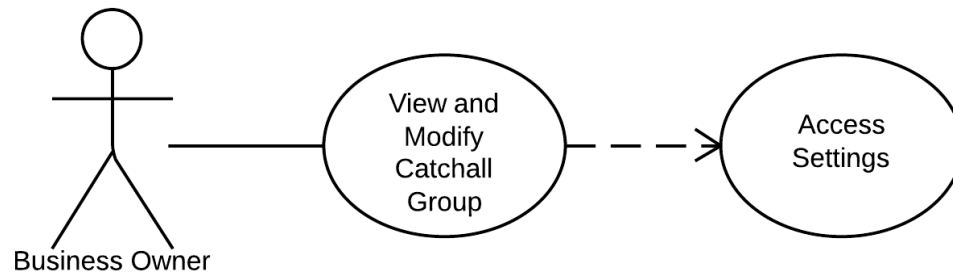
The business owner may attempt to re-sync data

The business owner presses the "OK" button

### **EXCEPTIONS:**

No data in the SQLite database

## ***View and Modify Catchall Group***



**Figure 10 - View and Modify Catchall Group Use Case Diagram**

### PRIMARY ACTOR:

Business owner

### GOAL:

View the current catchall group and optionally edit the catchall group

### PRE-CONDITIONS:

The settings page must be visible

### NORMAL FLOW:

The business owner presses the "Catchall Group" button

The business owner can see the selected catchall group (See Figure 35)

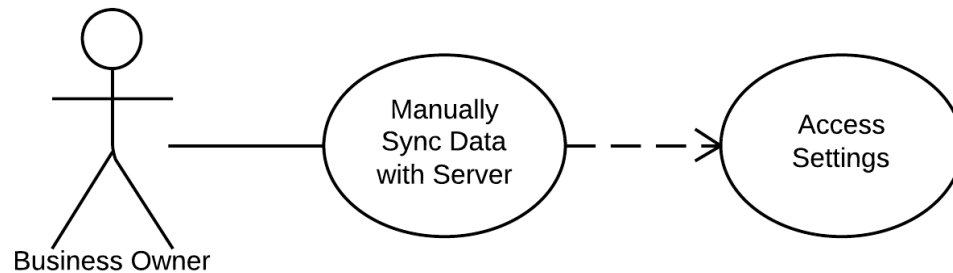
The business owner may edit the selected catchall group

### EXCEPTIONS:

The business owner cancels out of the catchall group dialog window



## ***Manually Sync Data with Server***



**Figure 11 - Manually Sync Data with Server Use Case Diagram**

### PRIMARY ACTOR:

Business Owner

### GOAL:

Sync data with the remote server

### PRE-CONDITIONS:

The settings page must be visible, and un-synced data is present on device

### NORMAL FLOW:

The business owner presses the "Sync Data" button

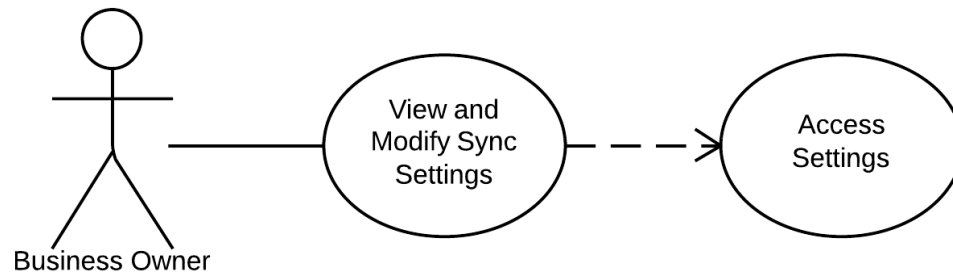
A visual indicator shows that data is being synced (See Figure 36)

Once finished, the last sync date is updated to the current time

### EXCEPTIONS:

Device isn't online

## ***View and Modify Sync Settings***



**Figure 12 - View and Modify Sync Settings Use Case Diagram**

### PRIMARY ACTOR:

Business Owner

### GOAL:

View the sync settings, and optionally modify the settings

### PRE-CONDITIONS:

The settings page must be visible

### NORMAL FLOW:

The business owner presses the "Sync Settings" button

The business owner may edit the days to hold data in the database (See  
Figure 37)

The owner may save, or cancel changes

## Delete Database Contents

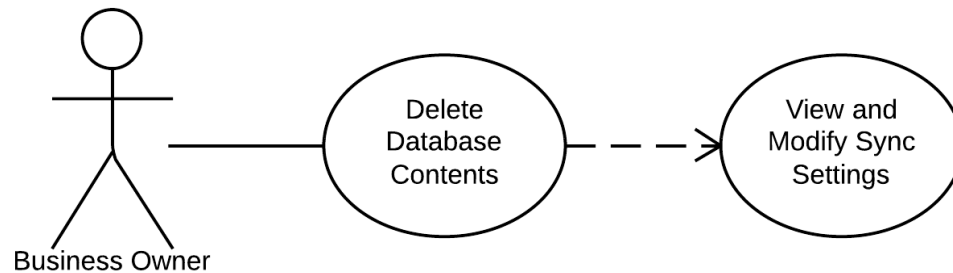


Figure 13 - Delete Database Contents Use Case Diagram

### PRIMARY ACTOR:

Business Owner

### GOAL:

Delete all check-in records from database

### PRE-CONDITIONS:

The sync settings dialog must be visible

### NORMAL FLOW:

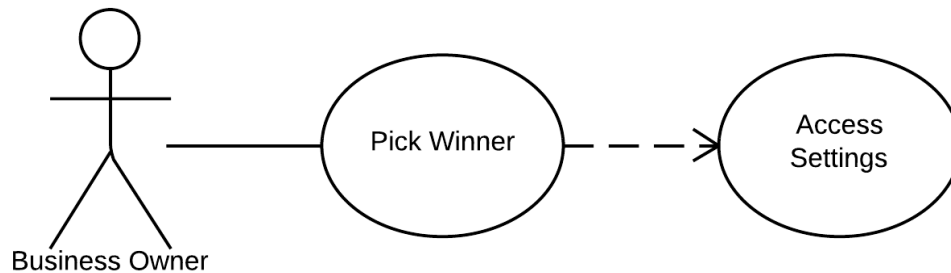
The business owner presses the “Clear Database” button

The message “Press 5 more time(s) to clear database” shows as confirmation (See Figure 37)

The business owner presses the “Clear Database” button 5 more times

### EXCEPTIONS:

The sync settings dialog is closed before database is cleared

**Pick Winner**

**Figure 14 - Pick Winner Use Case Diagram**

**PRIMARY ACTOR:**

Business Owner

**GOAL:**

Pick a winner from check-in objects during a range of time

**PRE-CONDITIONS:**

The settings page must be visible, with at least 1 check stored on the device

**NORMAL FLOW:**

The business owner presses the “Pick Winner” button

The business owner enters a date range that includes at least one check-in (See Figure 38)

The business owner presses the “Pick Winner” button

The system selects a random winner from the eligible candidates

**EXCEPTIONS:**

The user cancels out of the pick winner dialog

## Application Walkthrough

### Assumptions

This application can only be used by people with an active account on CityGro's website. This walkthrough assumes that the Business Owner has previously created an account, and set up business details that a Customer can check-into while using the application.

### Business Login Page

The Business Owner must log in before any Customer. This is to ensure that all data is submitted to the appropriate online account. The first time the application is started, the Business Owner is presented with the login screen (See Figure 15).

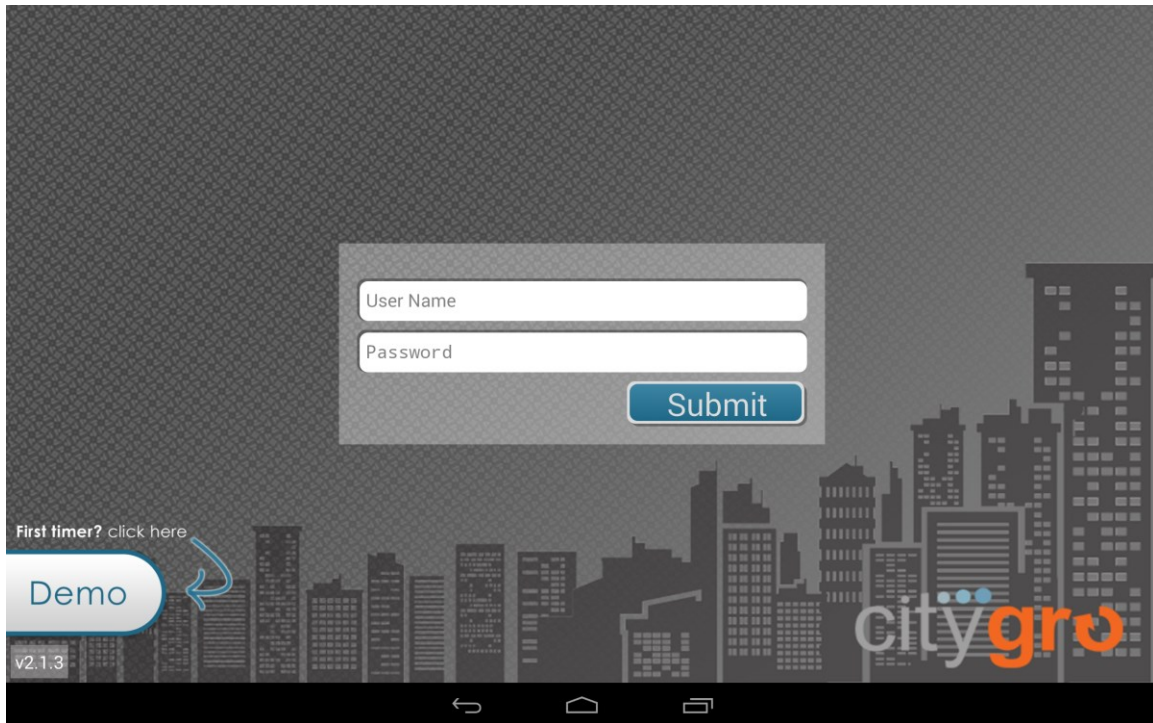


Figure 15 - Initial Login Screen

The Business Owner may now enter their credentials. The application defaults to saving the most recently entered and validated credentials. They are saved in the database, and encrypted using 256-Bit AES Symmetric Key Encryption. The next time the application is started, the username and password are decrypted and placed into the User Name and Password fields (See Figure 16).

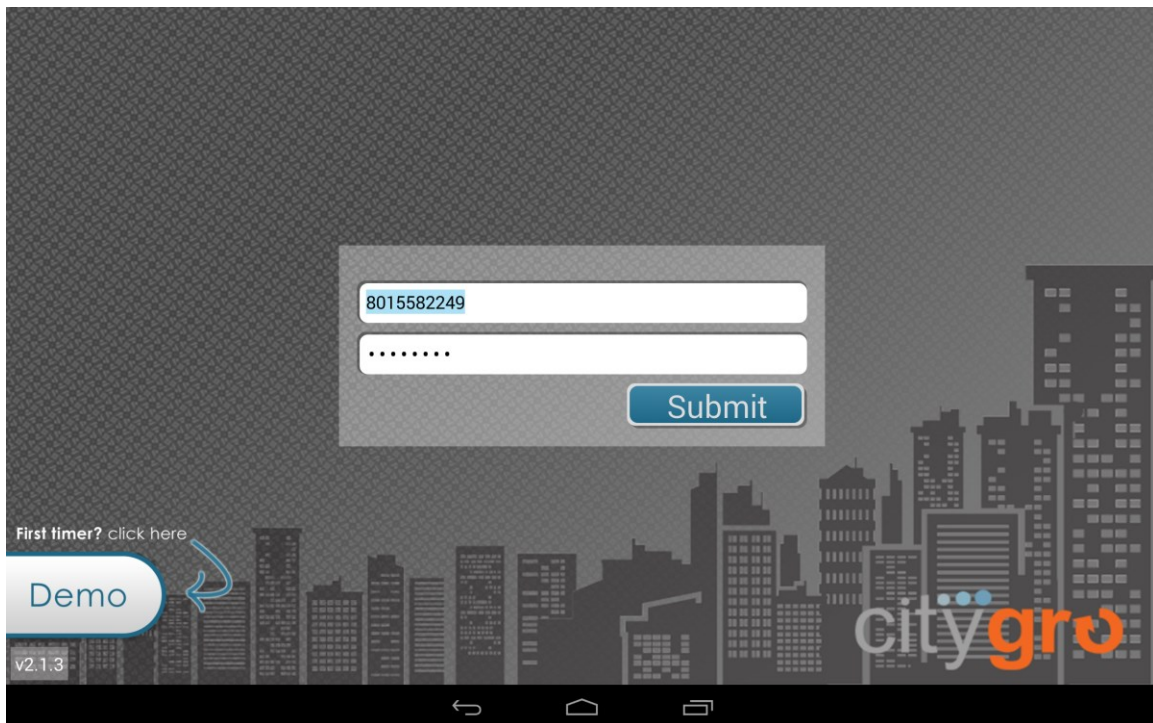
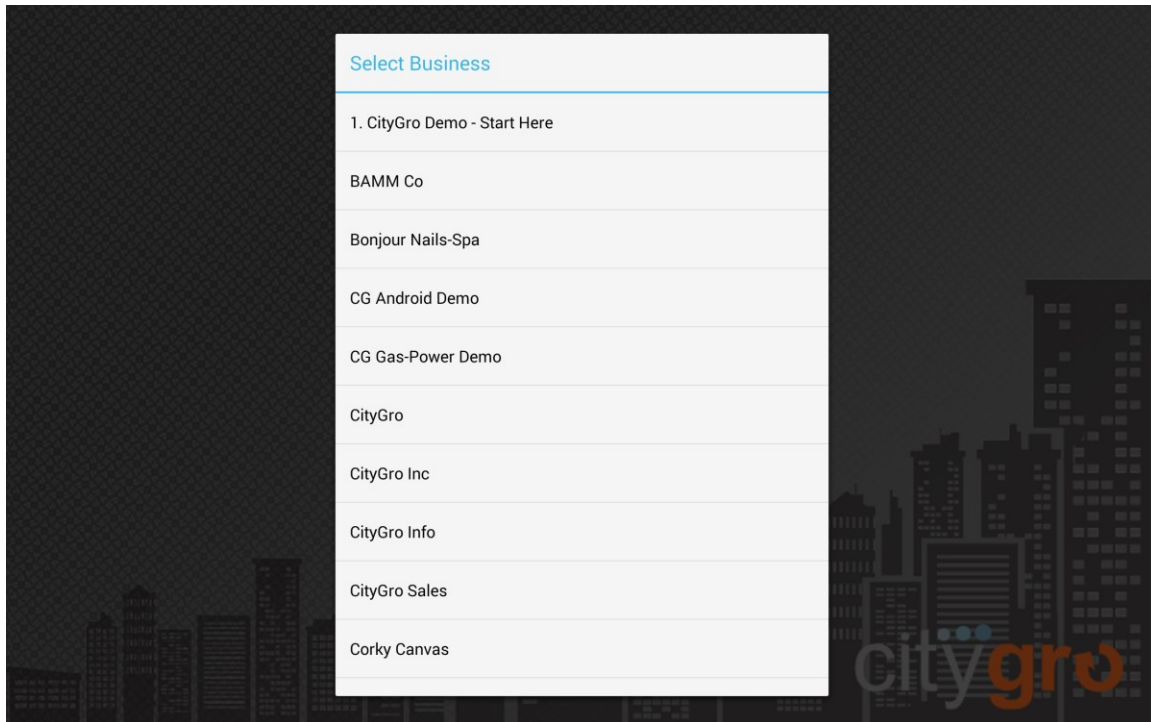


Figure 16 - Login Screen With Auto-filled Credentials

### **Business Selection Page**

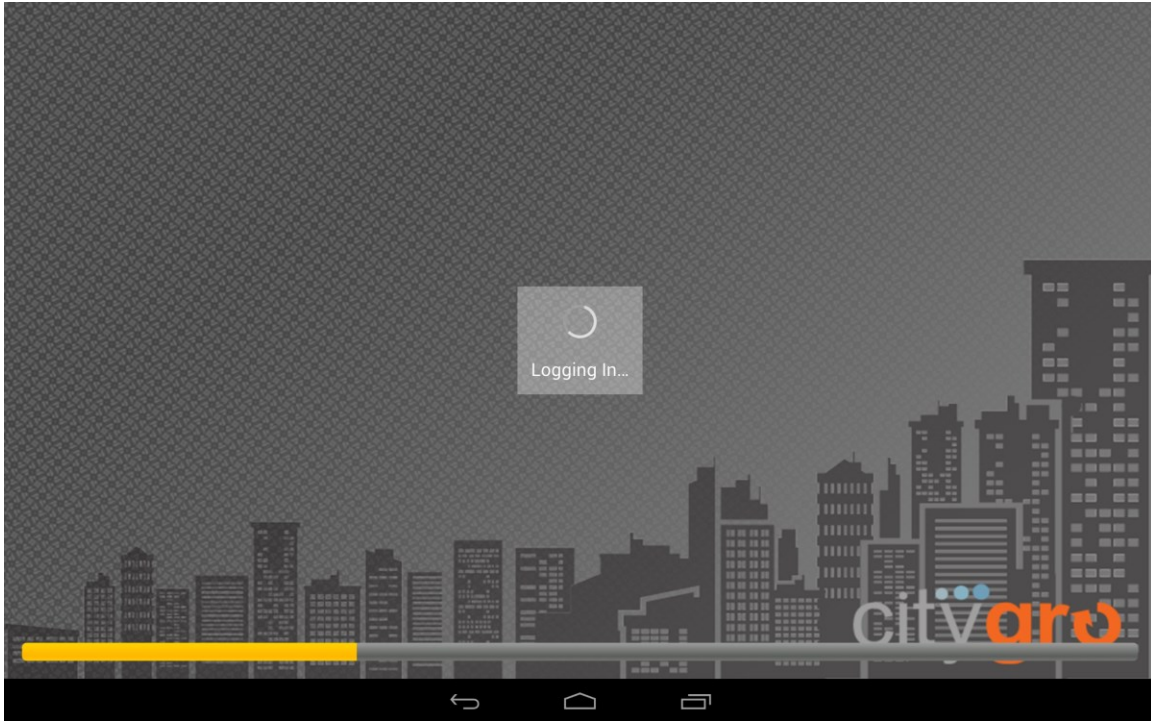
A Business Owner can be validated either online, or through saved credentials. Once a Business Owner is validated, they are asked to select a business that has been previously been set up online (See Figure 17). If the owner only has a single business, then no choice is given.



**Figure 17 - Business Selection**

Next the application downloads assets that are needed for that business. These are things such as company logos, backgrounds, and any other image used. These are all specified in a JSON object returned from the server upon logging in. This object is saved in the database in case the device is not online the next time the owner tries to log in.





**Figure 18 - Downloading Assets**

### **Customer Check-in Screen**

At this point, the tablet is ready for use by customers. There are additional customization options that will be discussed later. The customer is presented with the check-in screen (See Figure 19).

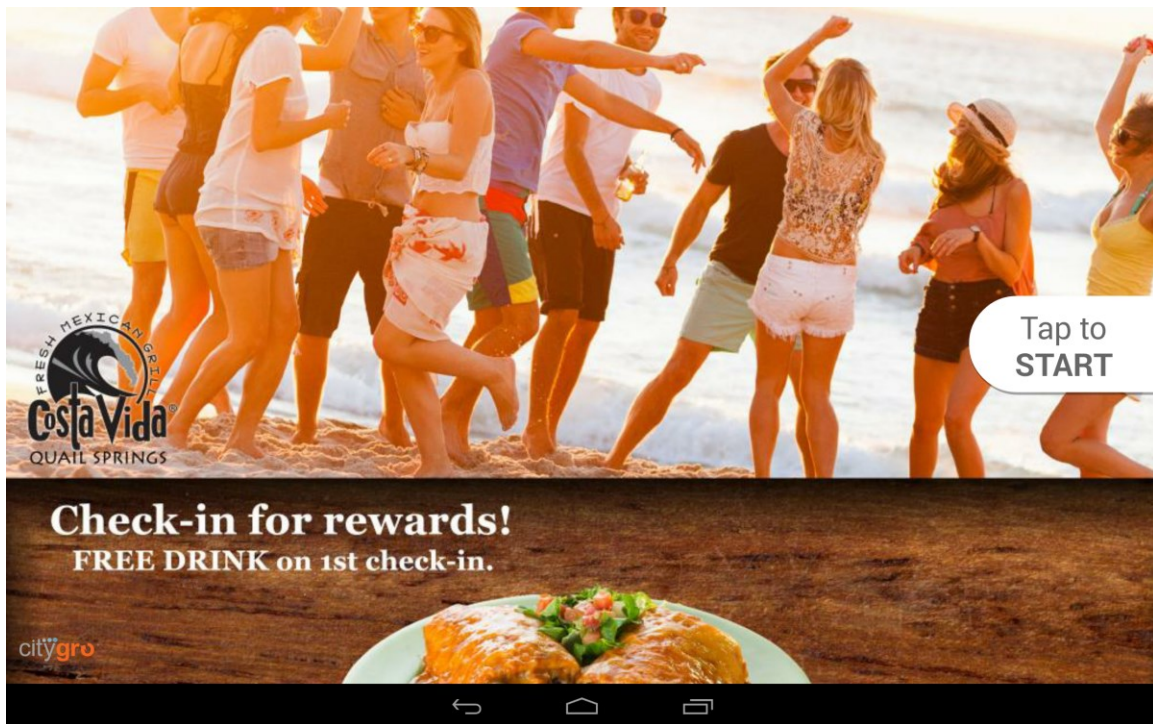


Figure 19 - Main Check-in Screen

This screen is highly customizable. Business owners may decide what key identifiers a Customer is able to use when checking in such as phone number, email address, or generic numeric and alphanumeric identifiers. If more than one option is supplied, the Customer can switch between input methods. If more than two are supplied, the Customer is prompted to select an input method (See Figure 20).

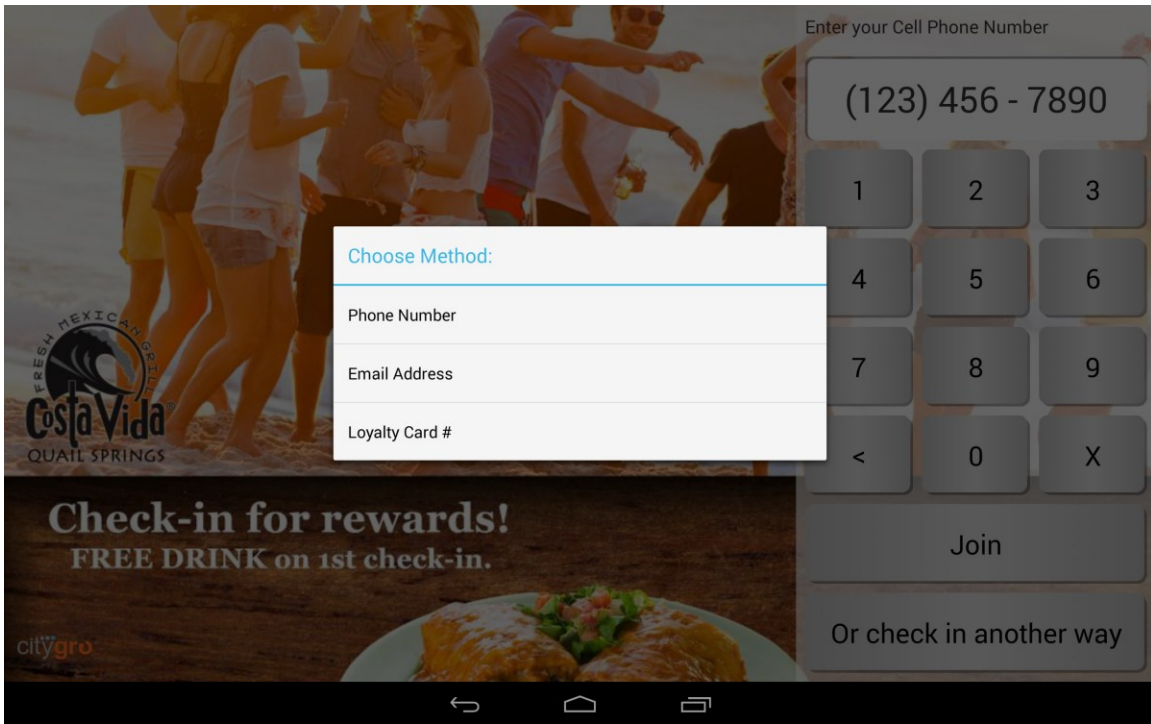


Figure 20 - Input Methods

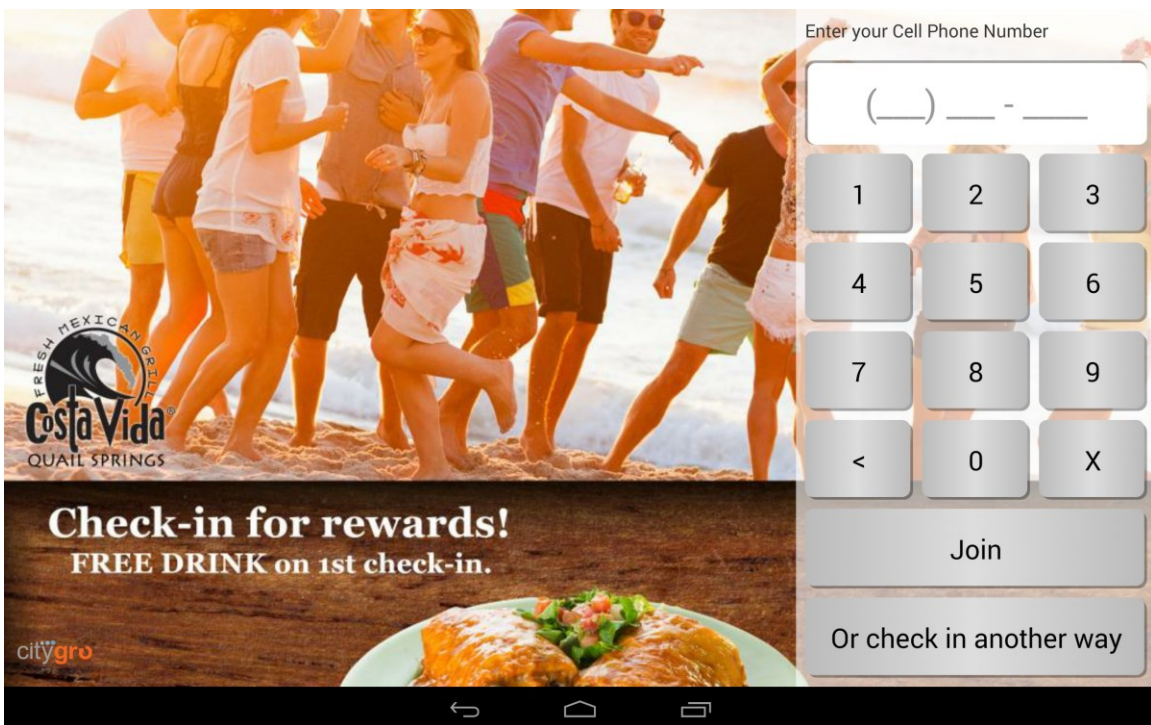


Figure 21 - Phone Number Check-in

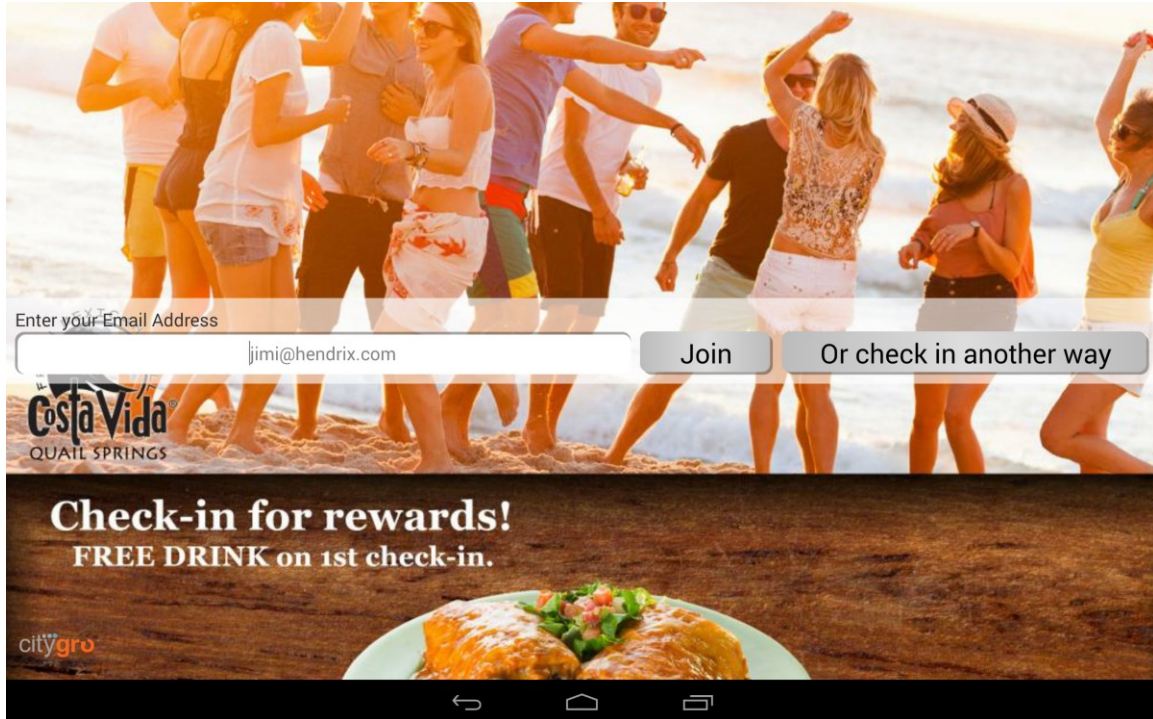


Figure 22 - Email Address Check-in

All input methods can be optionally validated with regular expressions. If an invalid identifier is entered, the Customer is alerted to their error (See Figure 23).

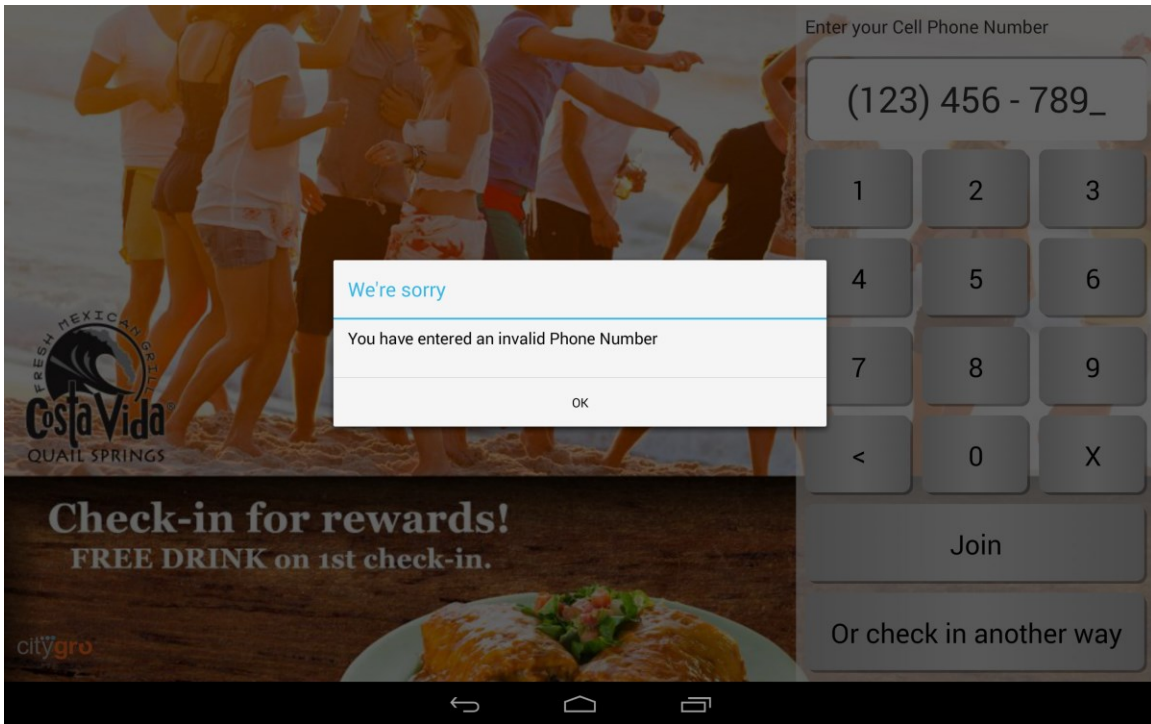


Figure 23 - An Invalid Phone Number

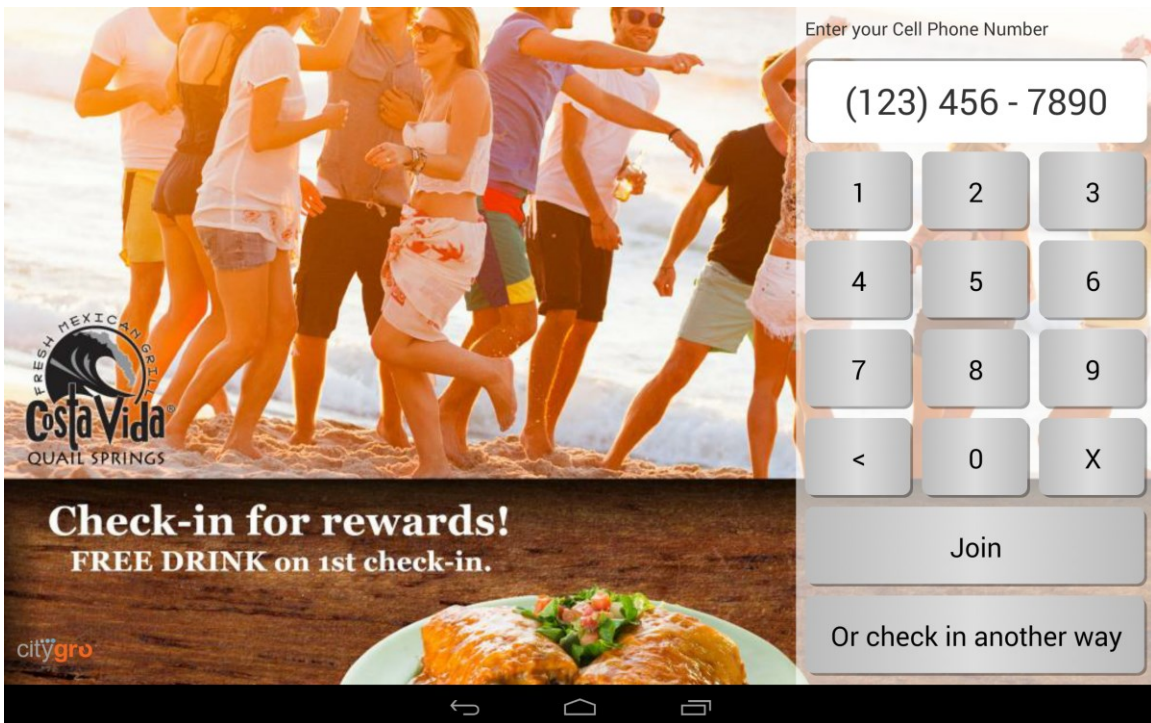


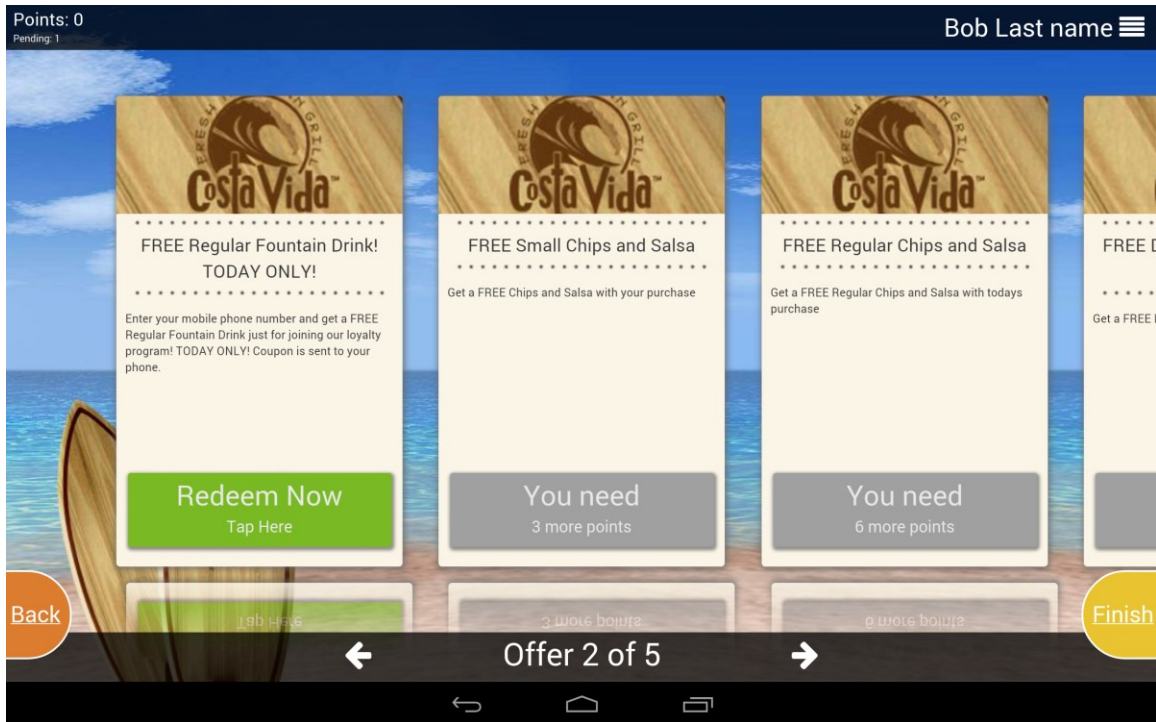
Figure 24 - A Valid Phone Number

If the Customer enters a valid identifier, they are validated with the server (See Figure 25). If no server is available, then the Customer is asked to enter information.



Figure 25 - Validating Customer with Server

The loading screen is shown for a minimum of 1 second. The business owner can specify this time online.



**Figure 26 - Redemption Options**

The Customer is presented with the option to redeem offers (See Figure 26). The offer in green is available immediately, whereas the grayed out offers are only available when the Customer has checked in multiple times. The Business Owner has the ability to determine the timeout required when checking into a business to ensure a customer is not checking in multiple times per day. When the 'Finish' button is pressed, the Customer is taken back to the check-in screen, via a splash screen (See Figure 27), where the process starts over.

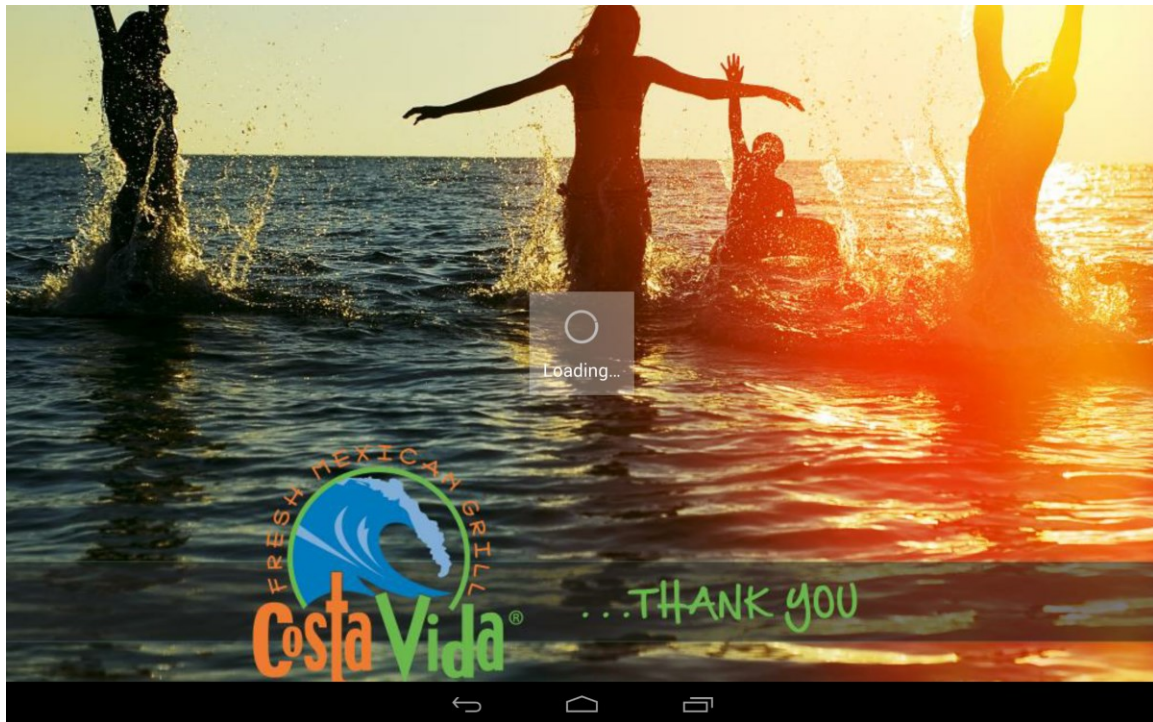


Figure 27 - Finishing a Check-in

At this point the finished check-in JSON object is sent to the server, and optionally stored on the device indefinitely. If no server is available, then the data is stored locally to be uploaded at a later time.

Businesses can specify other options to be available to the Customer on the check-in screen such as a business logo, a dedicated incentive message, or a custom terms and conditions link, which opens a web view to that location if the device is online.

### Local Administration Options

The application features some administration options that are stored locally and unique to a particular business. This means that if a device is shared between businesses, the settings remain unique to each individual business. The



administration options (See Figure 28) are accessed from the Customer check-in screen by drawing a circle on the background image.

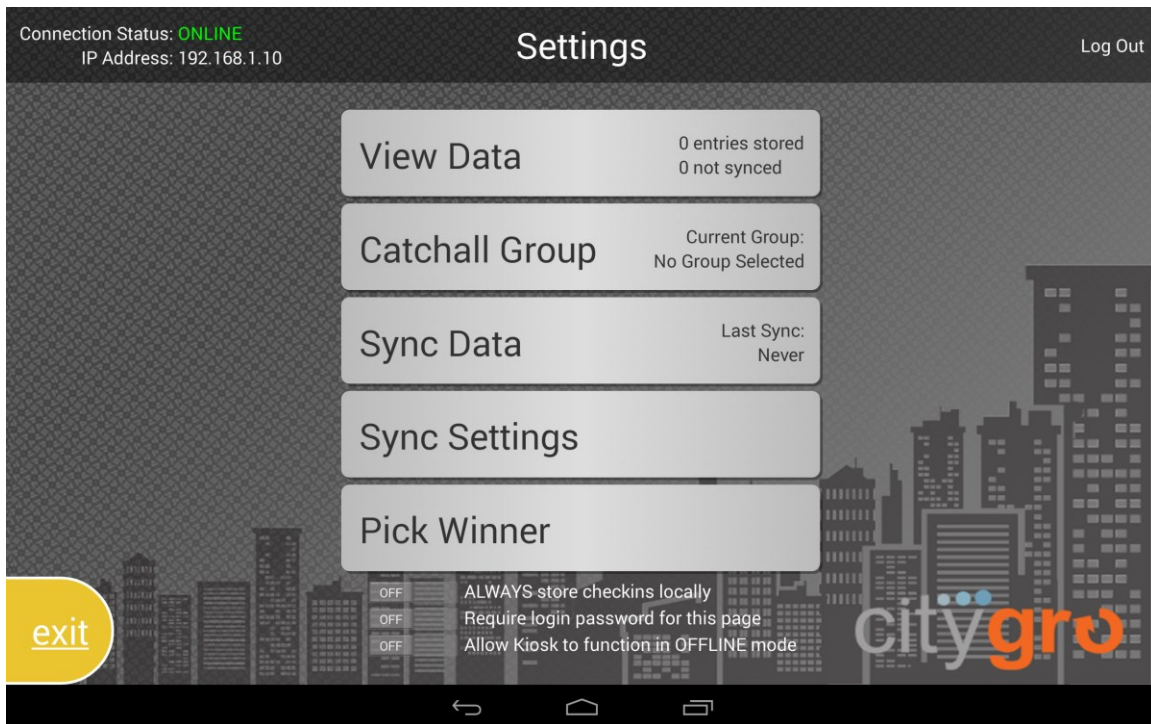
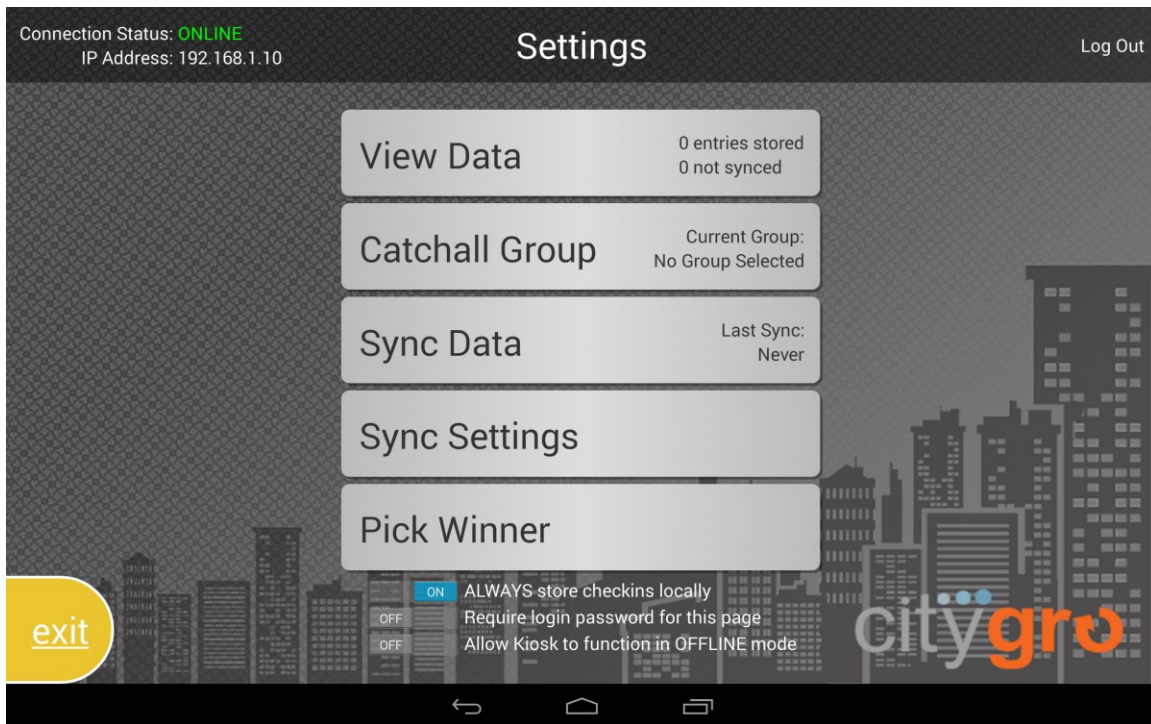
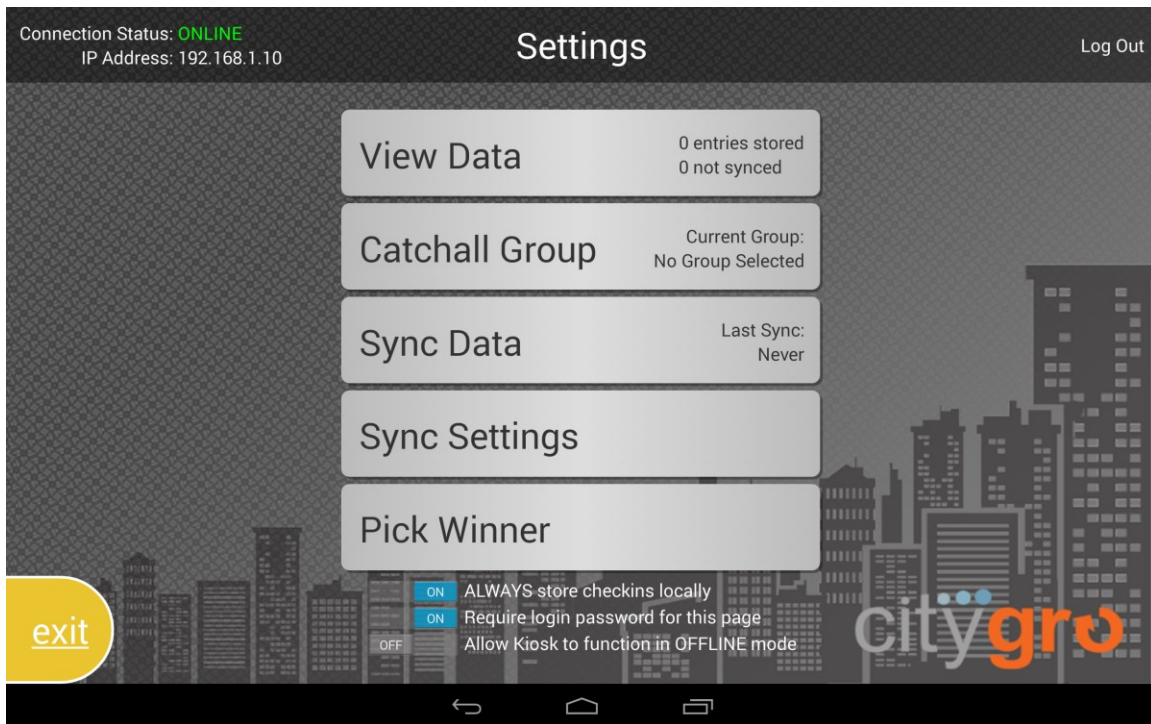


Figure 28 - Default Settings Page



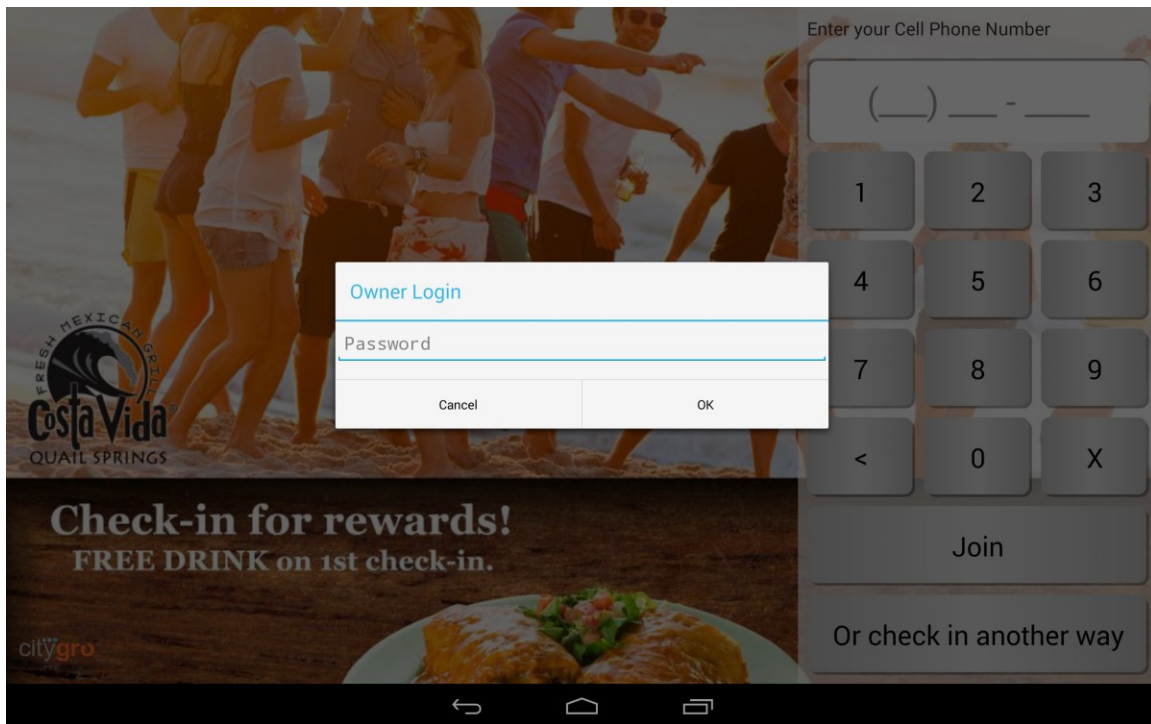
**Figure 29 - Always store check-ins locally**

When the “Always store check-ins locally” option is selected (See Figure 29), all check-in objects are stored locally on the device. This is useful when holding raffles since normally check-in objects are only stored long enough to guarantee that they end up on the server.



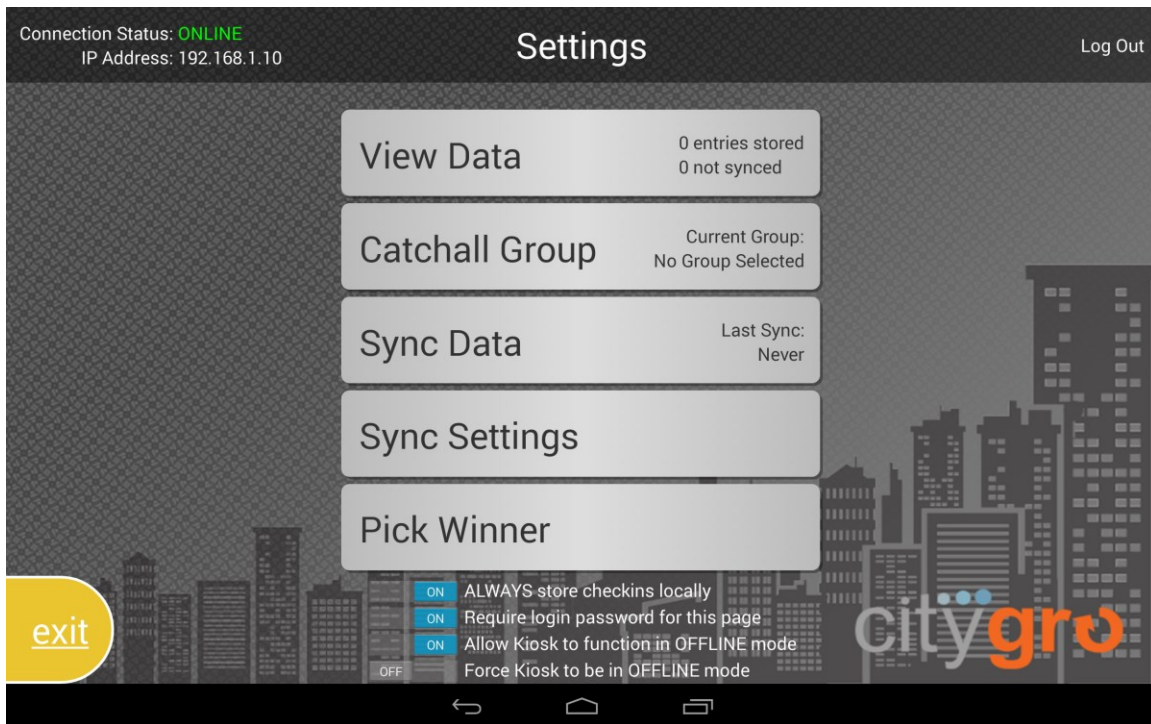
**Figure 30 - Require Login Password to Edit Settings**

The Business Owner can lock the settings page to make it available to only those who have access to the account password of the Business Owner. This guarantees that customers that know how to access the settings page can access no sensitive data.



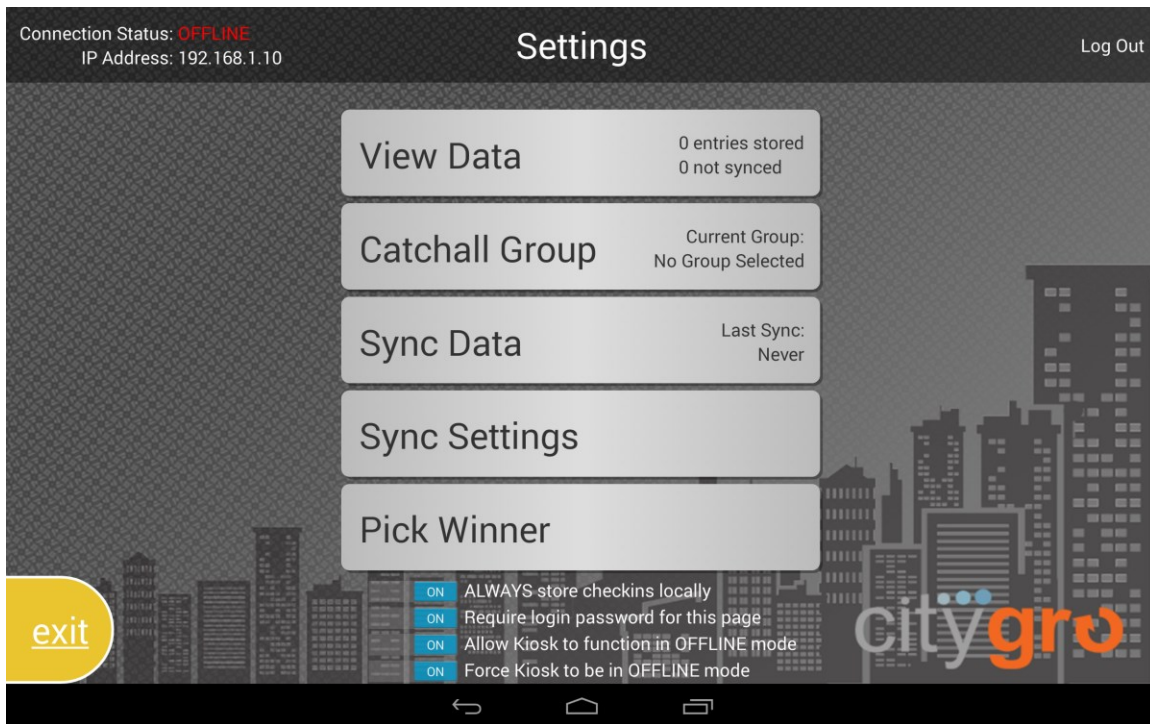
**Figure 31 - Owner Password Required for Settings Access**

If the password is required, the Business Owner is prompted to enter the password before they are allowed to access the settings page (See Figure 31). This password is validated against the locally held and encrypted password instead of validating with the server again.



**Figure 32 - Allowing Device to Function Without Internet Connection**

Normally, a Customer is not allowed to check-in if there is no current internet connection. Some businesses choose to not give internet access to the tablet during normal operation to save bandwidth and instead sync all check-ins at the end of the day. This allows the tablet to appear to function normally even if no active internet connection is available. If this option is selected (See Figure 32), a new option becomes available that can force the device into an offline mode.



**Figure 33 - Device Forced Into Offline Mode**

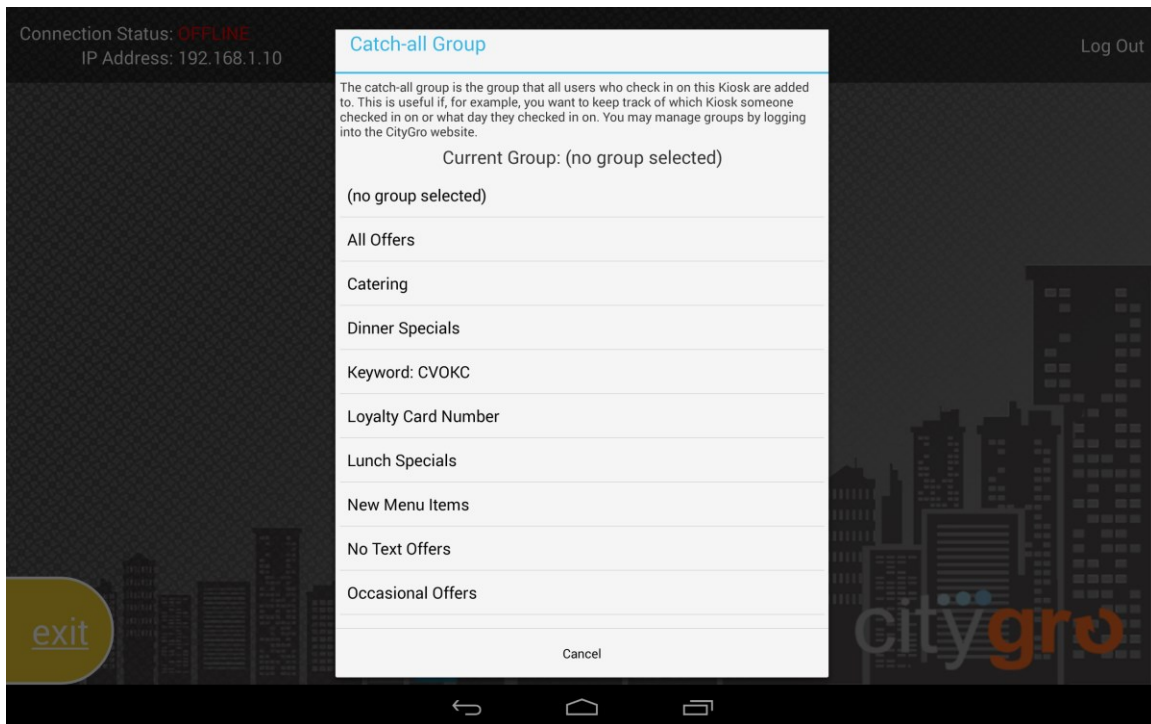
If an active internet connection is available, some Business Owners may still not want constant internet usage by customers. By forcing the device to act in offline mode (See Figure 33), it guarantees that the device does not attempt to reach the server during normal operation. A Business Owner may still use the available internet for syncing, however.

Business ID	Checkin Time	Available User Info
111427	02/10/2014 @ 20:44:51	Richie Preece: Phone Number: 11234567890

**Figure 34 - Data Viewer**

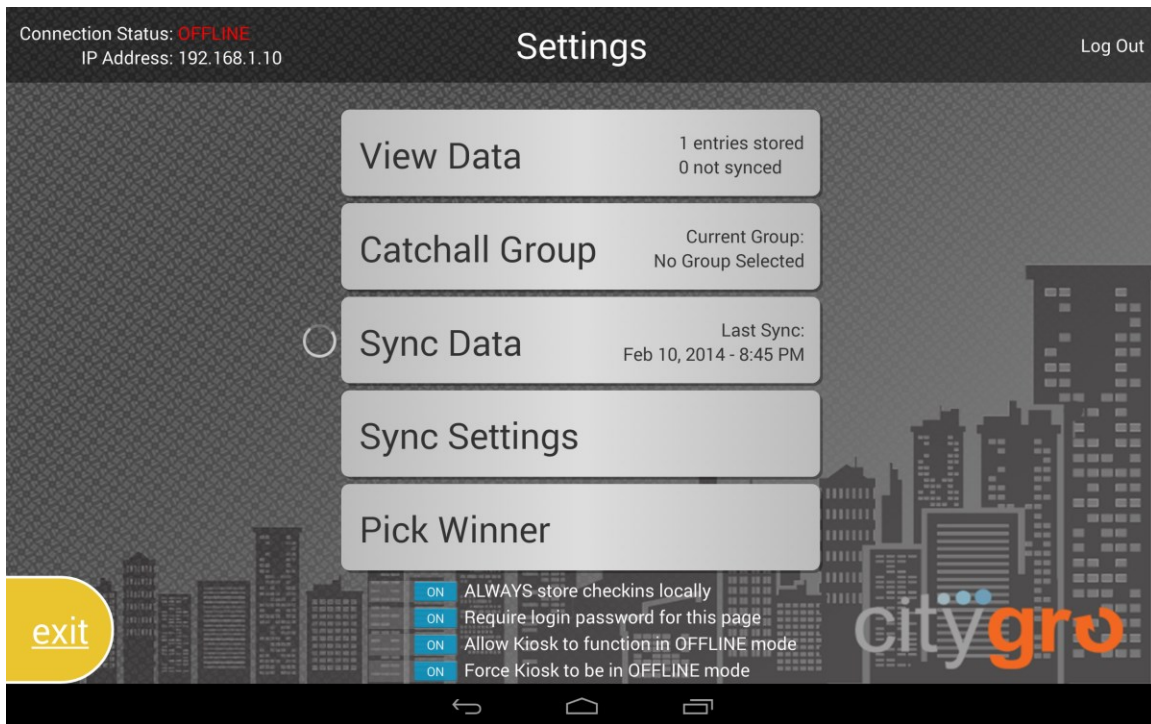
Business Owners may wish to see what data has been collected in the database (See Figure 34). Any data that is stored locally is shown to the

Business Owner. It gives the current sync status, and allows a Business Owner to re-sync the data.



**Figure 35 - Catch-All Groups**

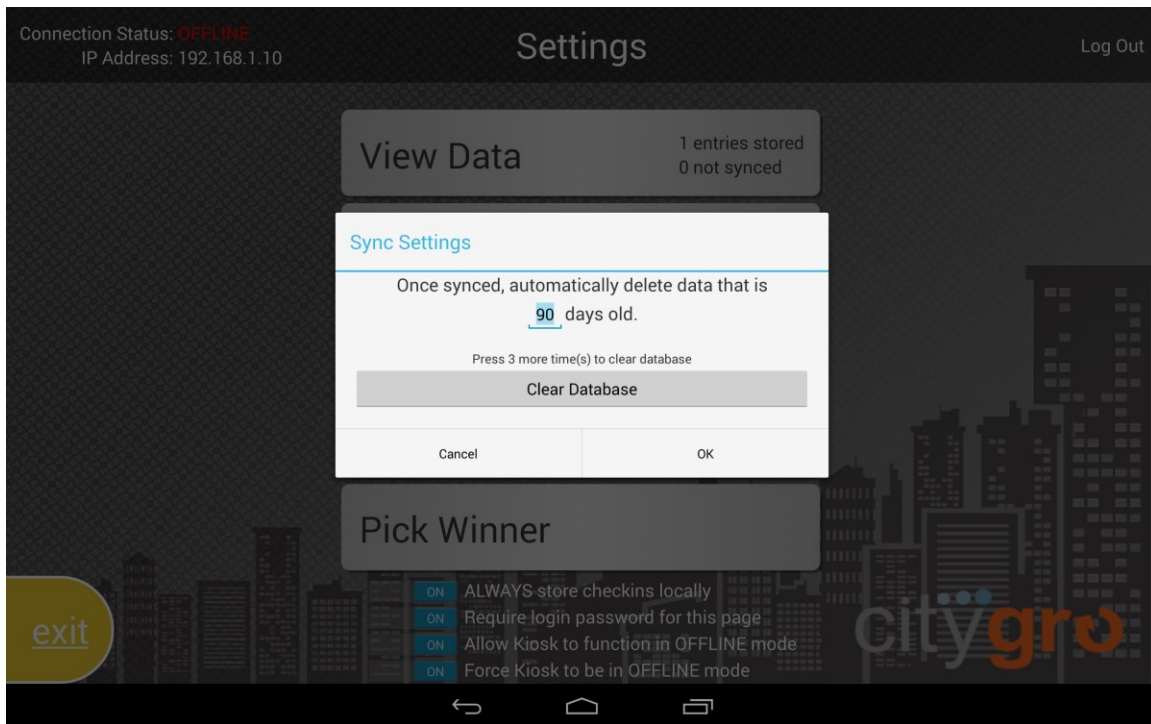
Catch all groups are used if a Business Owner wishes to group all check-ins during a period of time into a common group. These groups are created online, and selected by clicking on the 'Catchall Group' button.



**Figure 36 - Sync Data**

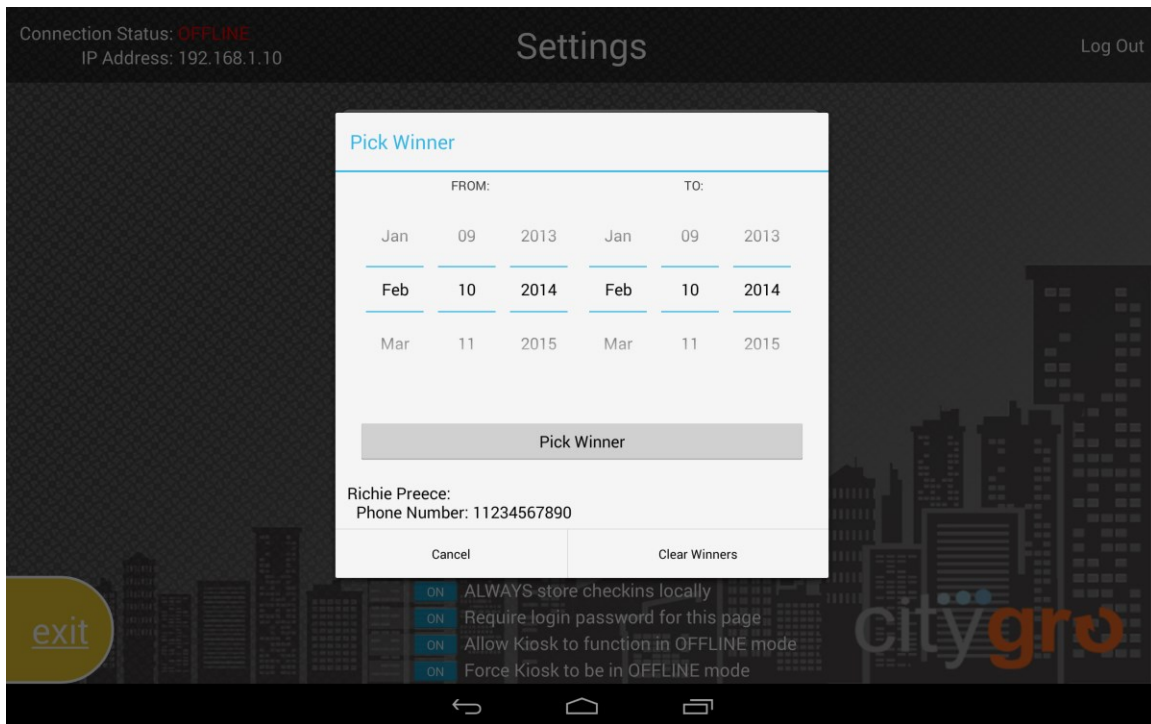
The 'Sync Data' button allows a Business Owner to upload check-in objects to the server in bulk, instead of one by one. It shows a loading spinner while syncing, and gives the time of the last sync (See Figure 36).





**Figure 37 - Database Options**

The Business Owner may specify the age at which to automatically delete records from the database (See Figure 37). Additionally, they may clear the entire database by pressing the 'Clear Database' button 5 times. A simple message appears warning the Business Owner of the impending action.



**Figure 38 - Raffle Winners**

The application offers the ability to pick a winner from a group of customers checking in over a period of time (See Figure 38). It limits the eligible customers to those that checked in from 12:00AM on the first day to 11:59PM on the last day. Additionally, it limits to those that checked in to the business currently logged in, in case there are records from another business present on the device. When a winner is chosen, their contact info appears and that check-in instance becomes ineligible to win again.

## Database Schema

This application takes advantage of the built in SQLite database tools. SQLite is appropriate for applications since the database is private to the application, and cannot be accessed by other applications, or users using shell access from a computer. Therefore the data stored in the database remains private and secure.

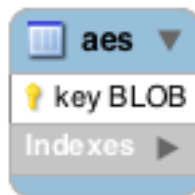


Figure 39 - AES Schema

An AES key is automatically created when the tablet first runs, and is stored in the database for encrypting and decrypting stored usernames and passwords. All encryption is handled in the database connector, and no encrypted data leaves the single encryption handling class. Any auto-fill data is first decrypted before being passed out of the database handler.

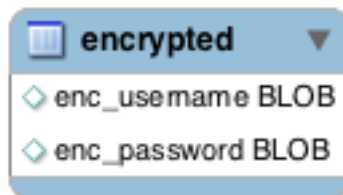
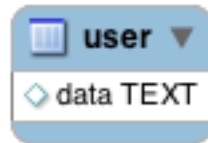


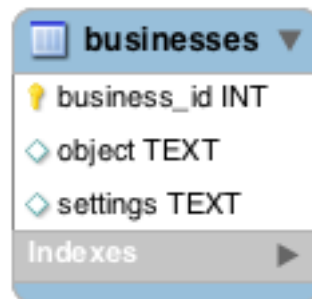
Figure 40 - Encrypted Schema

The encrypted table is used to store any credentials safely. The encryptions is merely an extra layer of defense as the data should not be accessible outside of the application anyways.



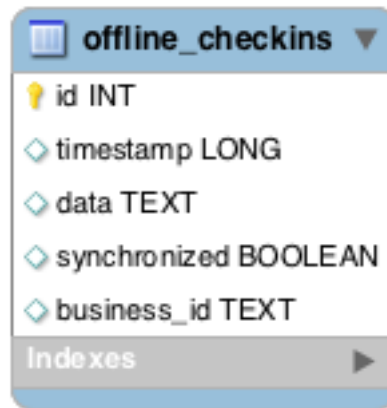
**Figure 41 - User Schema**

The user table stores the object returned when a business owner logs into the device. It holds the businesses available to the business owner as well as some other information. It is replaced with each successful online log in, and is referenced if the device is offline and correct credentials are entered.



**Figure 42 - Businesses Schema**

The businesses table stores the object returned when a business owner selects a business to use for customer check-ins. It contains a large of information including what check-in identifiers are allowed and what assets are used by the business. It is replaced with each successful online log in, and is referenced if the device is offline and the business owner is authenticated.



**Figure 43 - Offline Check-ins Schema**

Any permanently stored check-in object is held in this database. It gives a creation timestamp for deletion purposes, the JSON object, the associated business id, and finally a Boolean indicator of whether or not the object has been synchronized with the server.

## Additional Features

### Google Cloud Messaging

The check-in process takes advantage of a web view, which can be customized by the business owner. Should the business owner wish to change settings or upload new assets, or should CityGro alter the html, the device would need to reload the assets. This can be done through a push message system that tells the device to reload all assets.

### Rooted Devices

This application also takes advantage of devices that have been 'rooted' or, which allow root access to the user of the tablet. If the device has been given root access, the application will ask the user to access the root privileges when starting up.

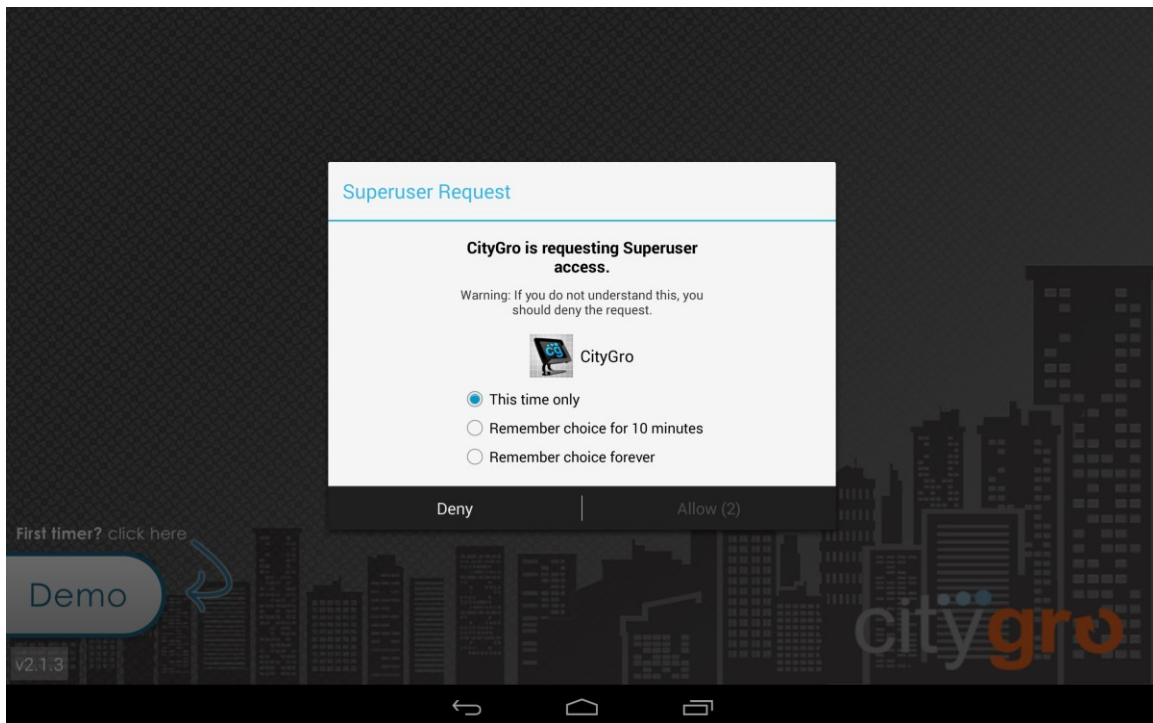


Figure 44 - Application Asking for Root Permission

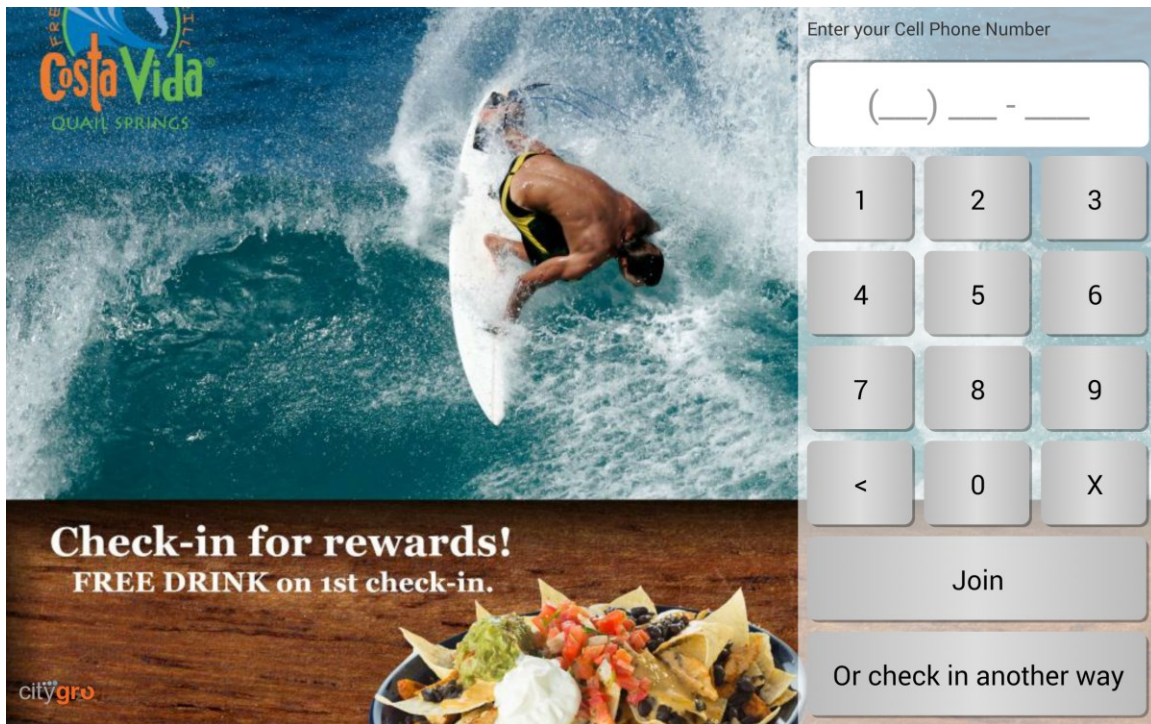


Figure 45 - Rooted Device

On a rooted device, the soft navigation bar disappears after the business owner logs in. This makes exiting the application impossible unless you are logged out of the application.

## Online Communications

The application has a limited set of online communications with a remote server housed and maintained by CityGro. The instances where Internet communications are attempted are:

- A business owner logging into the device
- A business owner selecting a business to use (this may be done automatically if there is only one associated business)
- The system downloading assets
- A customer begins a check-in on the device
- A customer finishes a check-in on the device
- A manual sync is requested

The application allows for the device to "... function in OFFLINE mode" as well as "Force ... to be in OFFLINE mode". The following table describes when the device communicates to a server over the Internet:

Internet Available?	Yes		
	Normal	Allowed Offline	Forced Offline
Logging In	YES	YES	YES
Selecting Business	YES	YES	YES
Downloading Assets	YES	YES	YES
Begin Check-in	YES	YES	NO
Finish Check-in	YES	YES	NO
Manual Sync	YES	YES	YES

Table 1 - Online Server Communications



In reality, the device communicates with the server during most normal operation. The only time it doesn't communicate with the server is for customer operations when forced offline.

## Chapter 5

### Summary

This application was developed over a period of approximately 6 months. It is currently being used by approximately 12 customers, and is still in a beta testing mode.

The application originally had two main goals, the first was allowing customers to easily and quickly check-in with their contact information, which was accomplished by always being on and waiting for customer data to be entered. The customer simply needs to enter a simple identifier and answer a few questions to redeem offers.

The second goal was to allow business to access and use the data provided by customers. With a simple setting, all data is stored on the device to be accessed by authorized users. This data can be used in limited ways locally, or when synced to a remote server, can be used in many other ways using the website provided by the company.

## References

- Adobe Systems Incorporated. (2009, October 23). *Adobe acquires Omniture Software*. Retrieved from <http://www.adobe.com/investor-relations/omniture-acquisition.html>
- Canalys. (2009, August 17). *Smart phones defy slowdown*. Retrieved from <http://www.canalys.com/newsroom/smart-phones-defy-slowdown>
- Canalys. (2011, October 31). *HTC takes the lead in the US smart phone market*. Retrieved from <http://www.canalys.com/newsroom/htc-takes-lead-us-smart-phone-market>
- Chen, H., Chiang, R. H., & Storey, V. C. (2012). Business Intelligence and Analytics: From Big Data to Big Impact. *MIS Quarterly*.
- Heitkötter, H., Hanschke, S., & Majchrzak, T. A. (2013). Comparing Cross-Platform Development Approaches for Mobile Applications. *Web Information Systems and Technologies*, 120-138.
- IBM. (2013). *The Ottawa Hospital: Mobile Puts Patients First*. Somers, New York, United States of America.
- IDC. (2014, February 12). *Android and iOS Continue to Dominate the Worldwide Smartphone Market with Android Shipments Just Shy of 800 Million in 2013, According to IDC*. Retrieved from <http://www.idc.com/getdoc.jsp?containerId=prUS24676414>
- Kony. (2013). *When mobile banking experiences count: Maximizing customer value at a major retail bank*. Orlando, Florida, United States of America.

Rao, N. M., Sasidhar, C., & Kumar, V. S. (2012). Cloud Computing Through Mobile Learning. *arXiv*.

Schadler, T., & McCarthy, J. C. (2012). *Mobile is the New Face of Engagement*. Cambridge: Forrester Research, Inc.

Zurovac, D., Talisuna, A. O., & Snow, R. W. (2012). Mobile Phone Text Messaging: Tool for Malaria Control. *PLoS Medicine*.

## Appendices

### Appendix A – Question and Answer with Jon Parrish

Q: Would you give a brief overview of how CityGro came to be?

A: CityGro stems back to a conversation I had with a grocery store owner who had just thrown away thousands of dollars of inventory that went bad. I asked him why he didn't give it away at cost to all of his customers and in return he asked me how in the world he would communicate a deal like that to all of his customers before the food spoiled completely. From there we started building difference tools to help capture customer data (primarily contact info) and enhance communication.

Q: How did the idea of a check-in kiosk start?

A: We started trying all kinds of different techniques to capture information (text ins, registering loyalty cards, QR codes, web forms, etc.) but adoption rates weren't very high... so we developed a way people could enter info right at checkout... when they are most hyped on the product.

Q: How many customers are using the Android version of the Kiosk?

A: Not many... perhaps a dozen or so. It's still in beta mode...

Q: What kind of statistics could you provide on Android/iOS kiosk usage?

A: We get anywhere from 5 to 500 check-ins per day. The average business will get 2000 check-ins per year. I'm sure Ben has a better answer for this. One of the neat things is that our kiosks have increased data acquisition rates more than 10 fold! We've had over 450,000 check-ins to date.

Q: Is there anything else that you think would be helpful for a report?

A: Maybe a spiel on automated communication based off a check-in? Or that we are now in over 17 states and 3 countries.