Web and mobile app requirements document

Introduction:

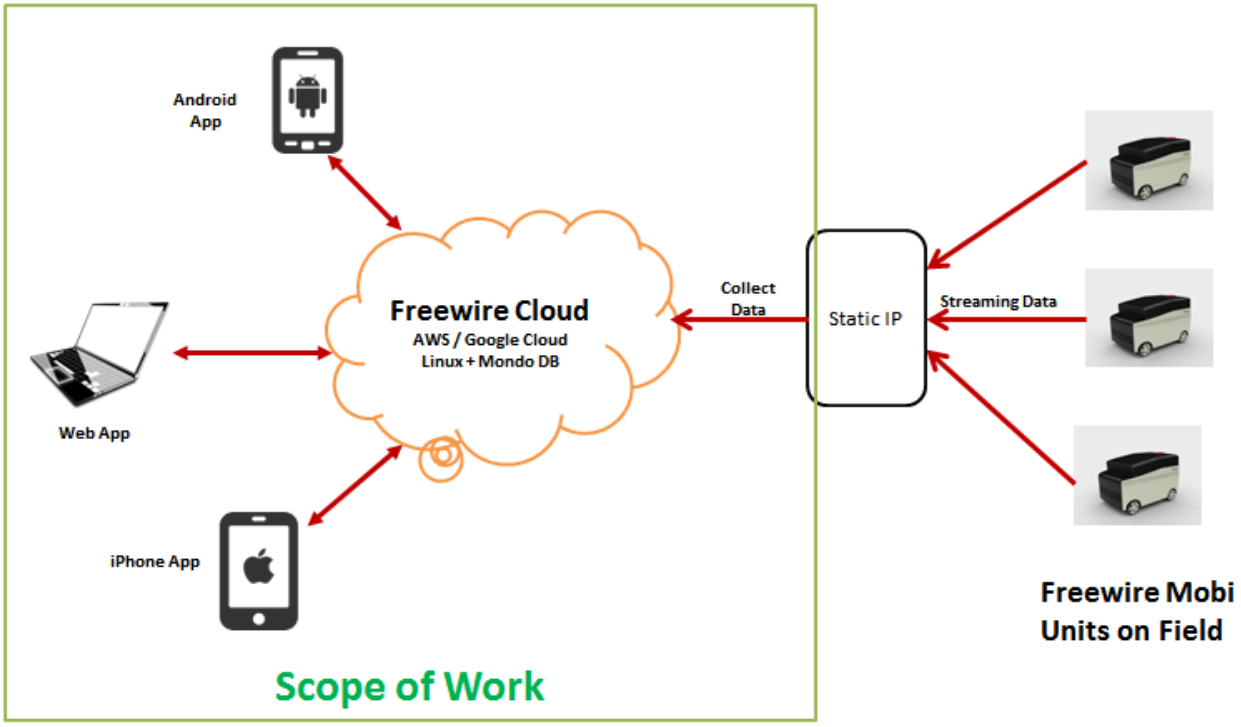
Freewire Technologies, Inc (www.freewiretech.com) based at San Francisco, CA is developing a scalable solution to the challenges of a large-scale Electric vehicle charging infrastructure at office locations and workplaces. Freewire is developing *Freewire Mobi* which is a mobile charging solution for electric vehicles that solves many issues associated with stationary electric vehicle charging stations.

Freewire will provide several Mobis (Electric Vehicle charging units that are mobile and have stored energy) and an attendant to workplaces where employees drive in with Electric Cars (such as Nissan Leaf, Chevy Volt, Fiat 500e etc). Freewire contracts directly with the companies/workplaces that provide free EV charging as a perk to their employees.

The freewire attendant will move these units in the parking lot via a joystick to the pre-registered vehicles that need to be charged. The attendant will plug in the charger to the car and will move to the next car when the charging completes. One attendant will manage about 5 Mobi units.

When a Mobi unit is depleted of the charge, the attendant will then move it inside the building to plug in for next charge.

Scope of Work



The work is for end to end development, testing and documentation for the Freewire Software platform. The scope of work includes

1. Technical design of Freewire cloud solution (AWS or Google Cloud), software design, database design, android, iOS & web add design.
2. Wireframing, design and development of Freewire Android, iOS and web app as per the use cases described in this document.
    a. Please note that the screens provided below are not going to be the exact scope. There may be some new screens needed as the design is completed. (although the functionality defined is not expected to change much. We will also share the psd files for the screens).
    b. The look and feel of the app needs to be redesigned and UX improved. The finish required needs to be of this professional quality and colors such as #3, # 8, # 24, # 26 in the link here. (http://thinknewdesign.co.uk/30-beautiful-mobile-ui-examples/)
3. Administration and setup of AWS or Google Cloud.
4. Development of Freewire software and database (mongo DB, but open to suggestions) on AWS or Google Cloud.
5. Development of interface with a predefined static IP to collect Freewire Mobi streaming data.
6. Unit testing and integration testing of the solution.
7. Documentation.

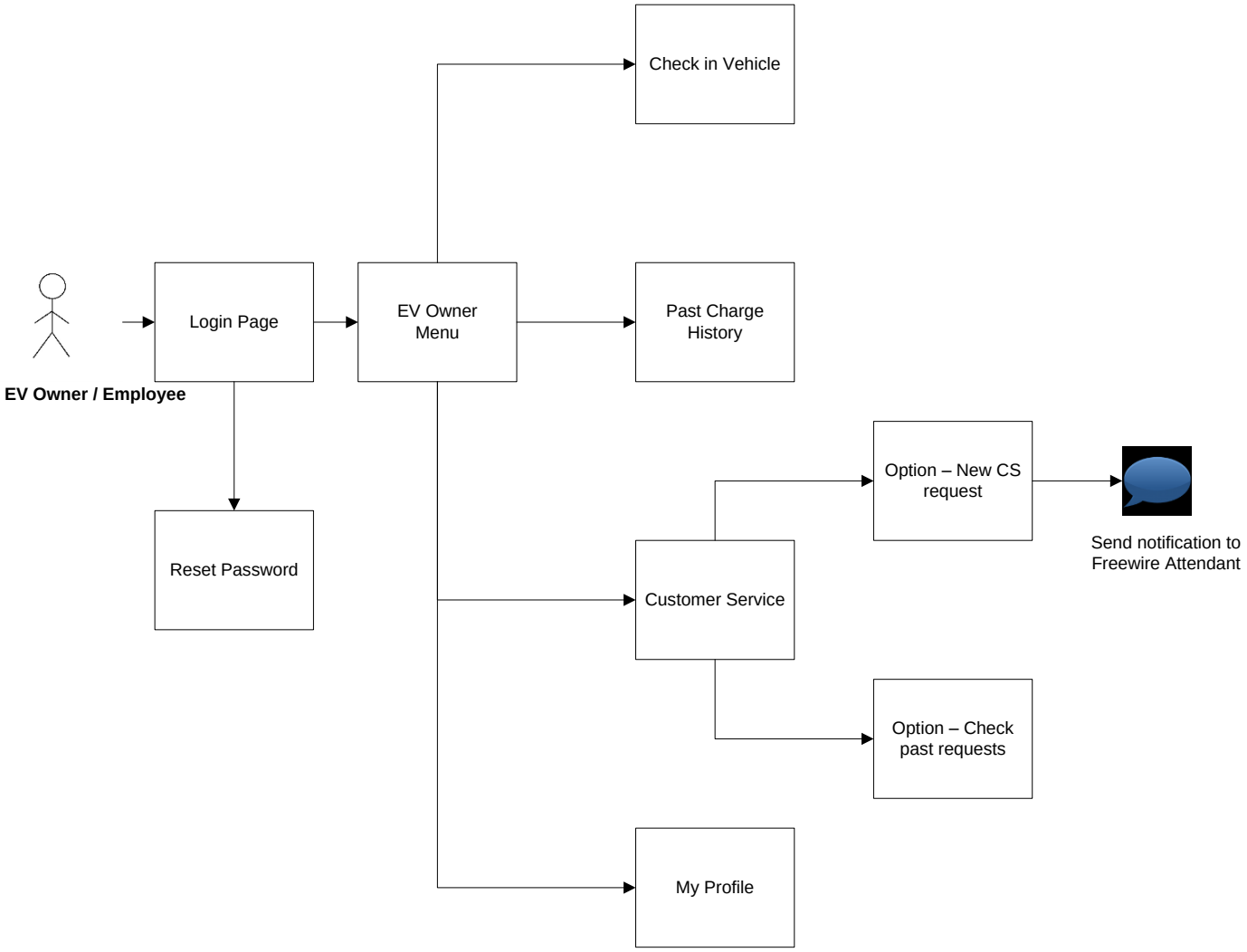
Functional Requirements

USERS

There are 4 types of users.

1. **Electric vehicle owner / Employee:** This user group represents people who drive electric cars and work at the company that have contracted the Freewire services. These employees are selected to participate in the Freewire charging program. The EV owner has access to view and transact data related to her own electric car only.
2. **Freewire Attendant:** The Freewire employee who moves the Freewire Mobi units around in the parking lot and charges the electric cars of the EV owner. This user can access the data from all EV owners for a specific client only.
3. **Client Administrator:** The Company's main person responsible to ensure that the employees are serviced in accordance with the service engaged. This user can access the data from all EV owners for a specific client only.
4. **Freewire Administrator:** This is a super user and has privileges to see and view data across all clients.

EV Owner use cases



Use Case – Login and forgot password (applicable for all user types)

Description: Simple login screen with option to save login and password and a link to reset password. The user name and password is pre-defined by the administrator and emailed to the user in advance. The reset password (sample screen not available) link takes the user to another screen with a button that will send an email to the freewire admin to reset the password for the user. Freewire admin will reset password manually and email the user directly. On the login screen, the Freewire logo is displayed.

When the user opens the app, the user name and password is auto saved. If the user successfully logs in, a menu screen opens up. There are 4 menu screens based on the user type. After the login, both freewire logo [can be hardcoded] and the client logo [Client Logo is saved in the client profile described later. Client logo is not applicable for Freewire Admin because they are not linked to a client] is displayed. The apps also shows a picture of the user [user has the option to upload her picture from the my profile option. If she does not load her picture then a default icon is showed] and greets her as "Welcome *Amanda Jones @Client*" [user name comes from the user table First name + last name. The client name comes from the client table]. The screen presents menu for the EV Owner [based on the user type in the user

table].

Use Case – Check In (EV Owner Only)

Description: The check in process means that the EV driver has arrived at workplace parking lot and has parked her electric car, and she wants to let freewire know where she has parked at.

When and EV Owner parks in the parking lot, she logs into her Freewire Mobile app and checks in. The Mobile app records the coordinates of the location and records it on the parking lot map / list.
Check in feature is also available via the web. After check in the App reminds user to keep the charging port unlocked. The check in is added to the list of the vehicles to be charged during the day.

Mobile Experience:

- The check in page opens up a map of a parking lot and shows a blinking tag over the map based on the GPS location. The objective of the check-in is to let EV driver tell the app where she parked the car.
- The screen advises the user to drag the tag on the screen to place it where the user parked it at.
- After the user places the car at the new spot, she clicks a button to check it in.

- At the completion of the check in, the system confirms that the check in information has been received. She will be notified when the charging session begins.

After the "check in" menu options

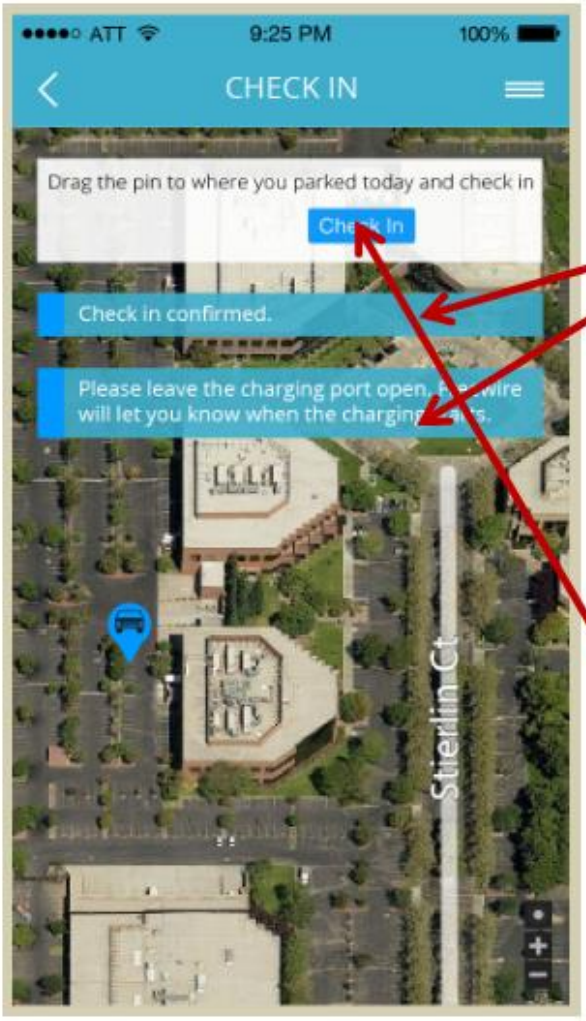
This map is actually an image. This image is a copy from bing maps and is shows the parking lot of the client. This is uploaded when the client record is set up.

The user is able to scroll the map up and down and sideways to the extent of the image limits.

The user touches the pin icon and moves it over the map to the general area where she parked at. When the user leaves it the pin stays. The user can drag and move it again, until she checks in.

When the user is done placing the pin, she checks in by pushing this button.

The user is also able to zoom in or out of the image from these buttons or by pinching.



When the user completes the check in, the system responds by confirming the checkin and advising the user to keep the charging ports open.

Please combine the 2 blue boxes. There is no need to have them separate.

After the check in is complete, this box is not needed. Please remove it. (Later this will be used to let user move their cars from their current location, but it is not needed in the first release)

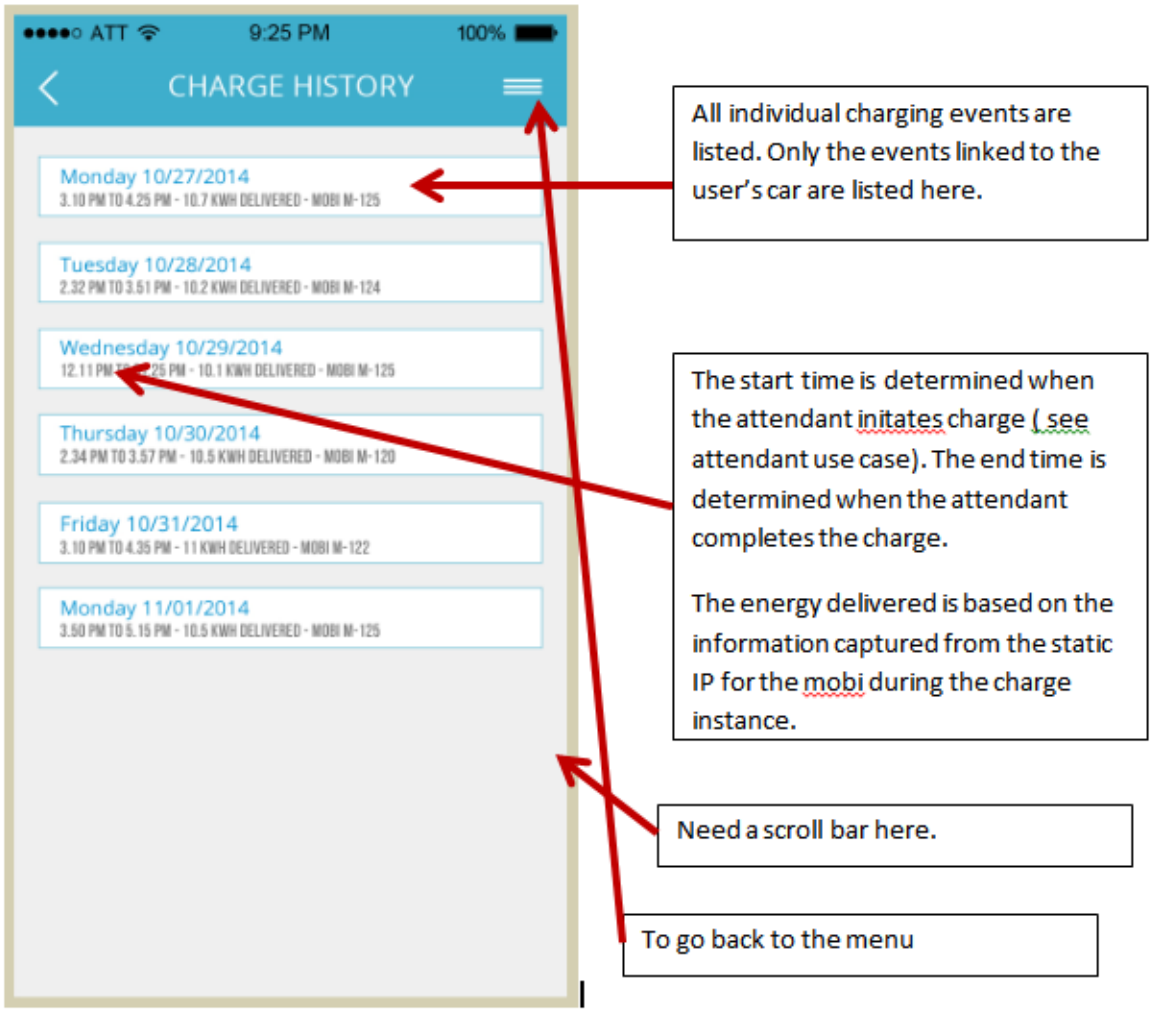
Use Case – Charge history

Description : EV Owner / Employee checks the history of all the charge events from the past. The screen shows her the She can see the actual time when the charge times of charge, charge delivered etc

Mobile Experience:

* The user selects the "Past car charging events" option

* The objective of this option is to let the user know the past charging events from freewire Mobi. Display the following or similar information on the next screen.

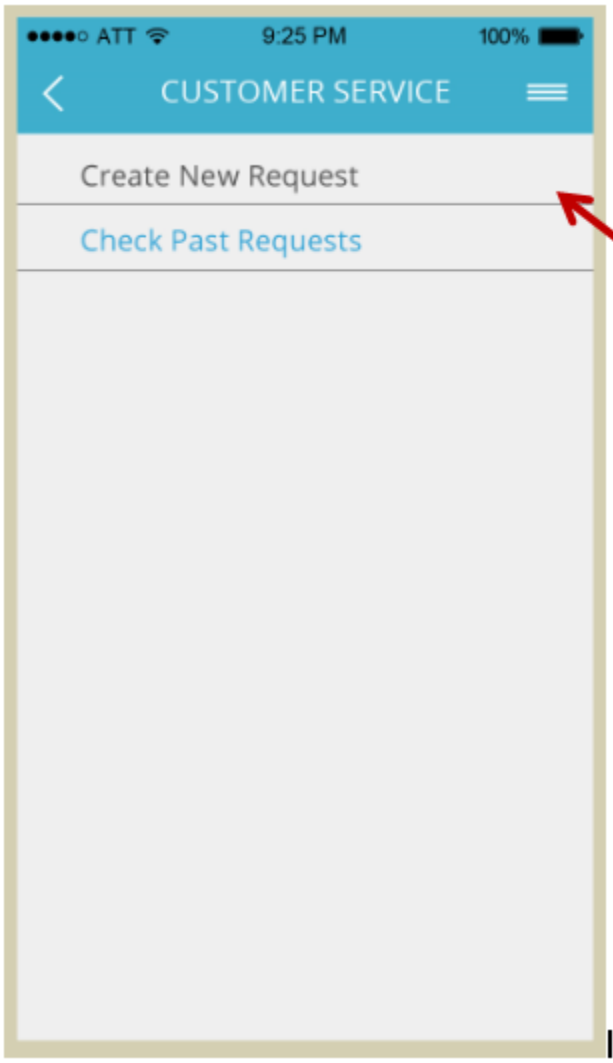


Use Case – Create Customer Service Request

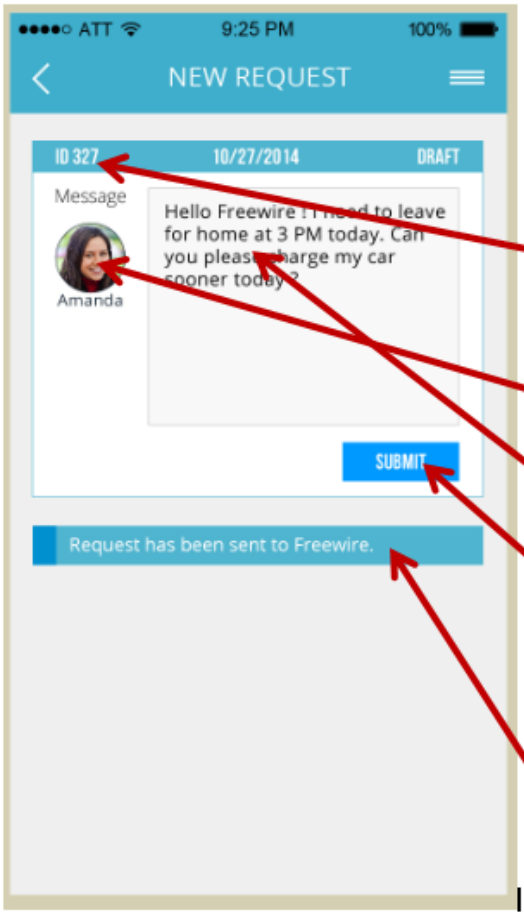
Description : This allows an EV owner to create a service request to send a message to the Freewire Attendant or check status of an existing service request.

Mobile Experience:

* After login, the user selects the "Customer Service Request" option
* Next Screen opens an option to create a new request or check a list of past requests.
* The user clicks on a new request and the app opens up a text box to type. The App also creates a Request number (unique across the entire Freewire network) , a date timestamp, and status.
* After typing, the user clicks a submit button. The app pops up a message that the CS request has been sent to Freewire.
* If the client administrator logs in, she gets to see all the CS requests from the client.
* The user or client admin can change the status of a request.



Sub menu after the user selects the Customer service from the main menu



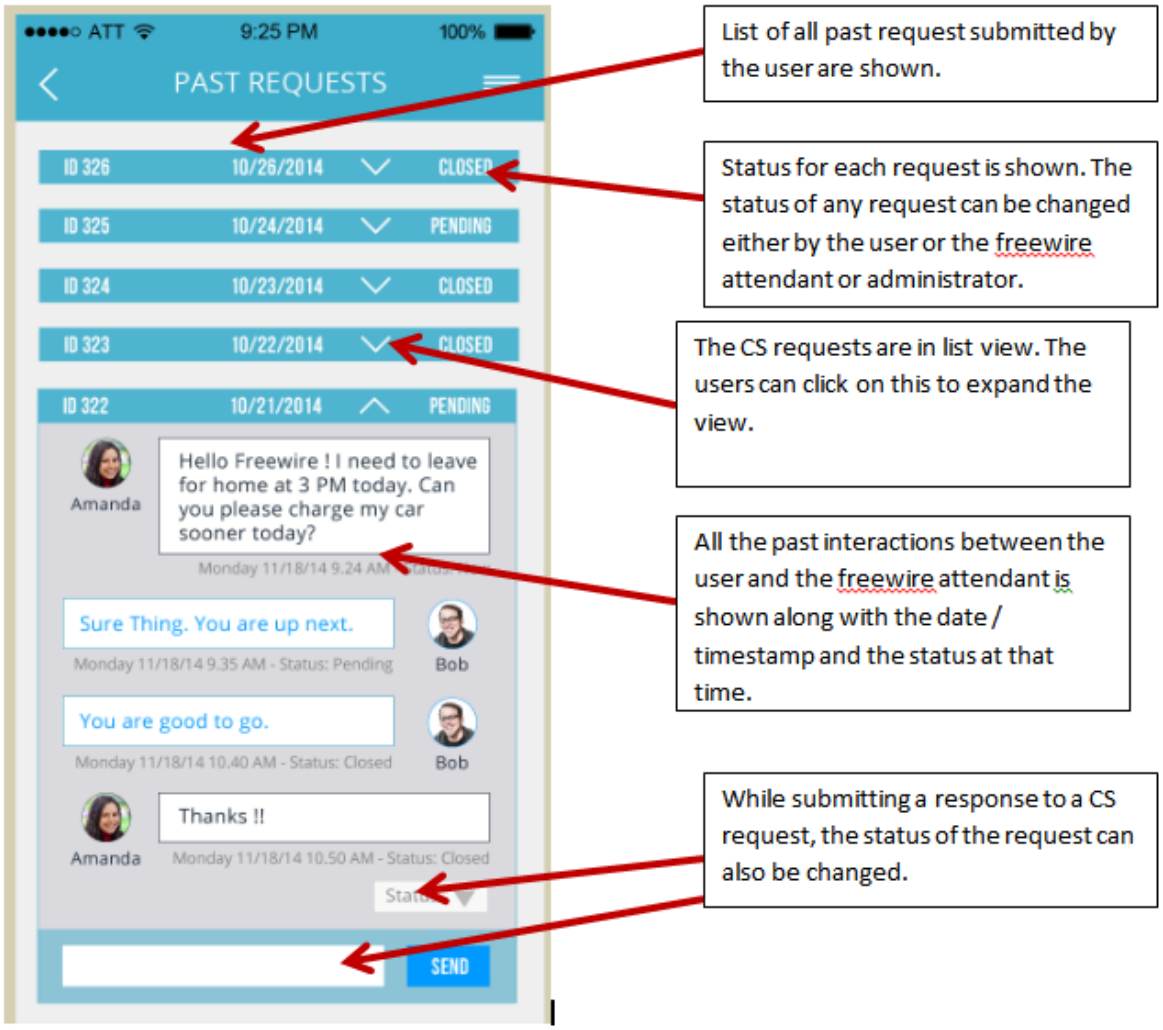
System automatically generates a new Customer service number for all new requests.

User's picture from the user profile

Free text.

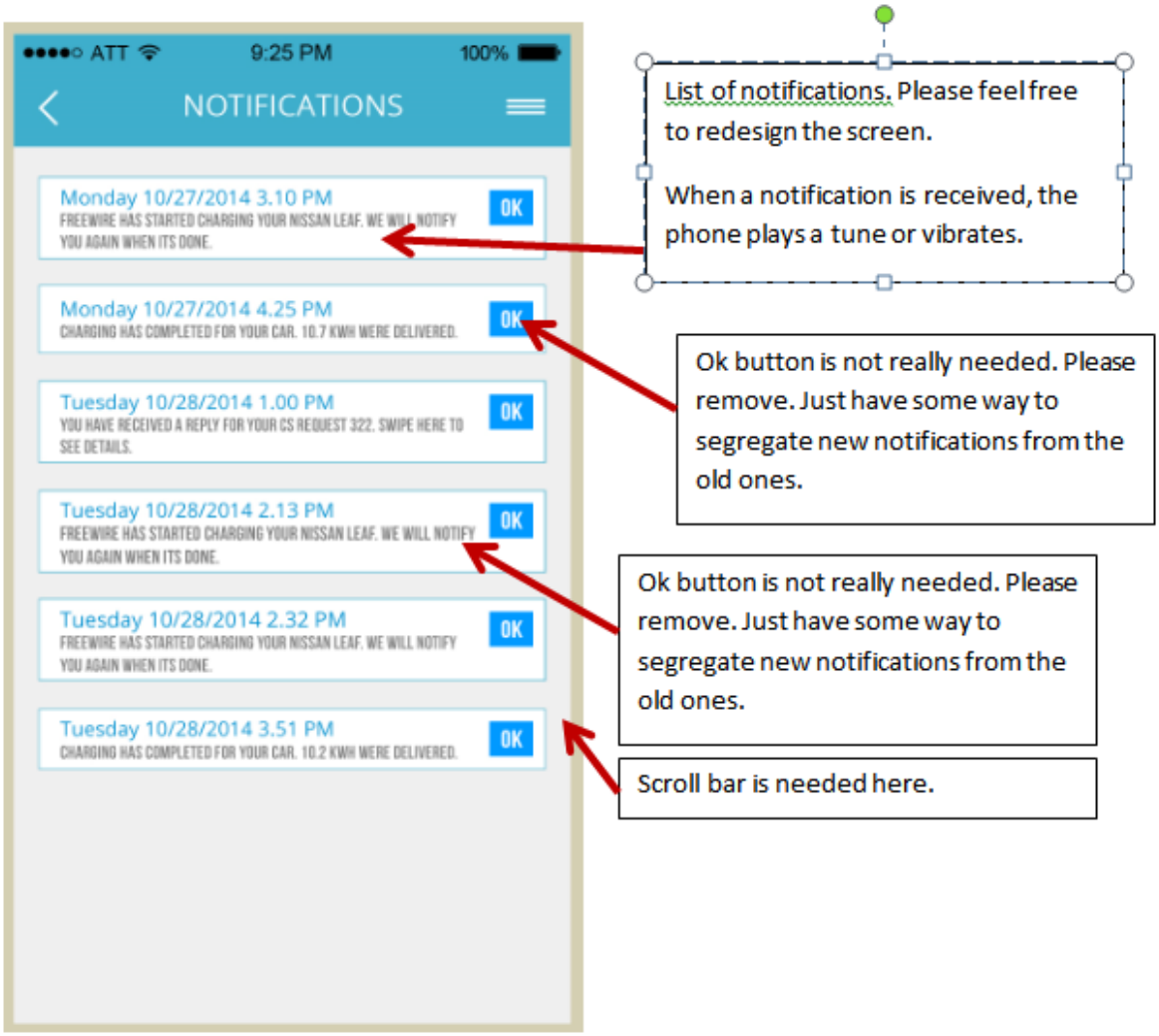
Submit button to create a new request. This button disappears when the text is blank or if it has been submitted.

This shows only after the text has been submitted.



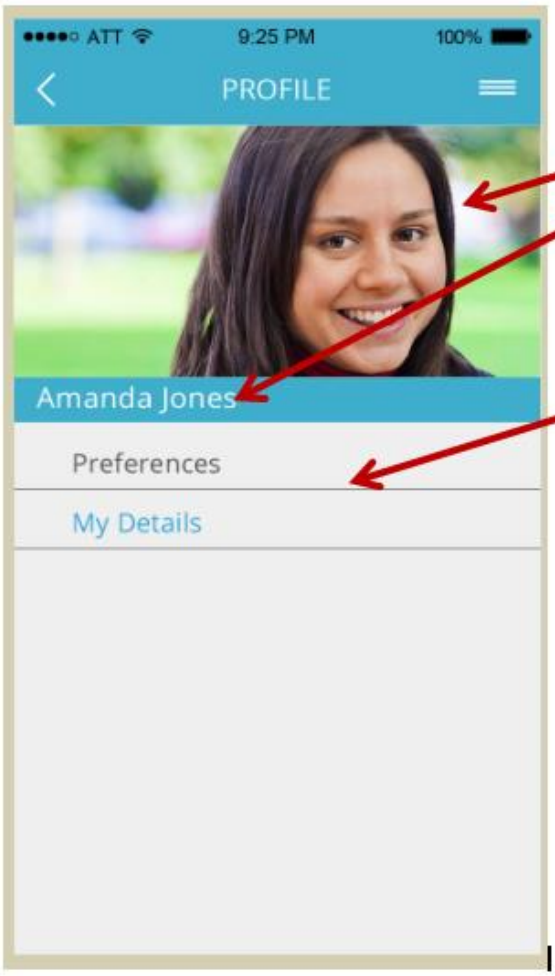
Use Case –Notifications

Description : When freewire attendant initiates the charge on a vehicle, he triggers a notification via a mobile device for the EV Owner / Employee. Owner's phone beeps with the alert. Owner has the ability to respond by OK button or by a text back to the attendant. A notification is also generated when a user receives a response to a CS request.



Use Case – My profile

The user is able to change some information about herself. The information is first populated by the freewire admin from the Create Users screen.

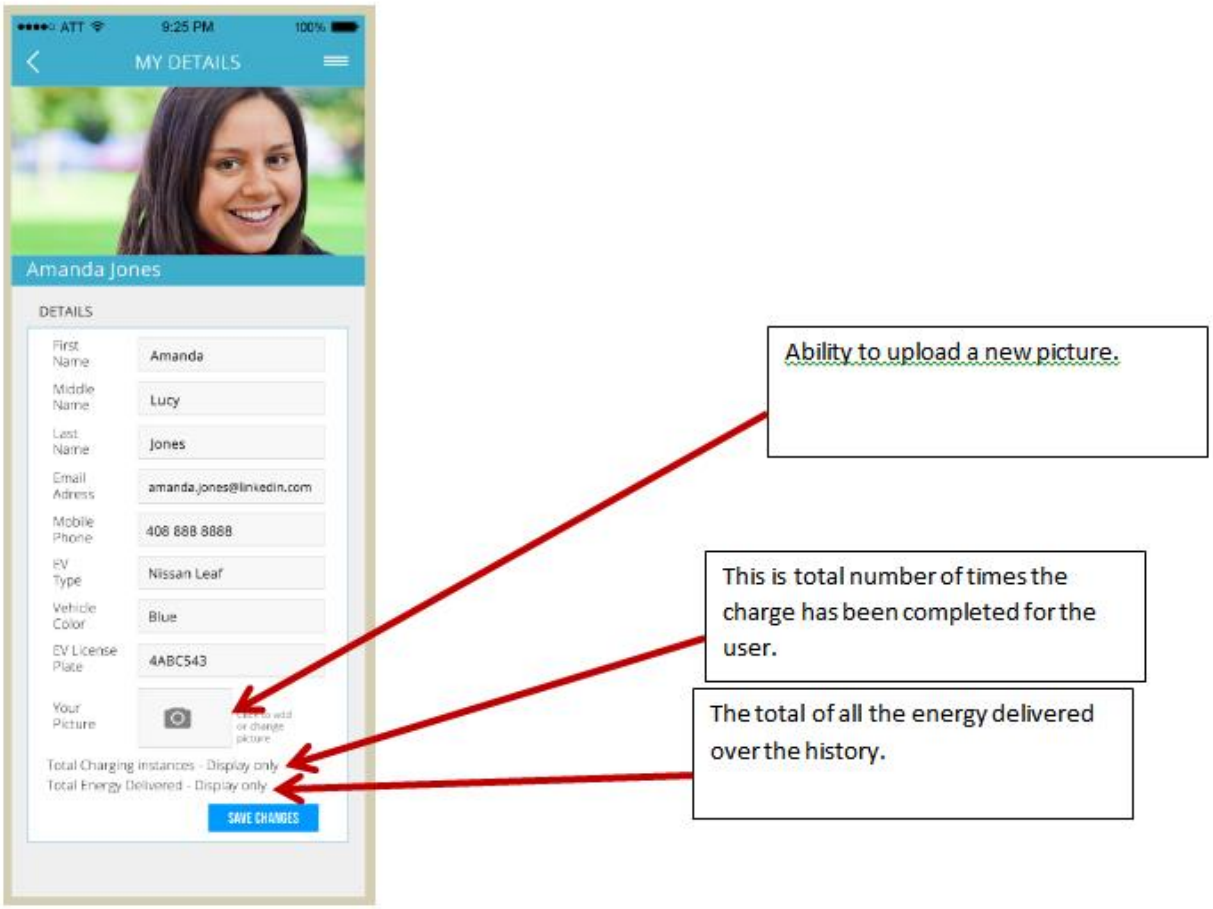


Picture and name of the user based on profile

Sub menu for profile menu option.

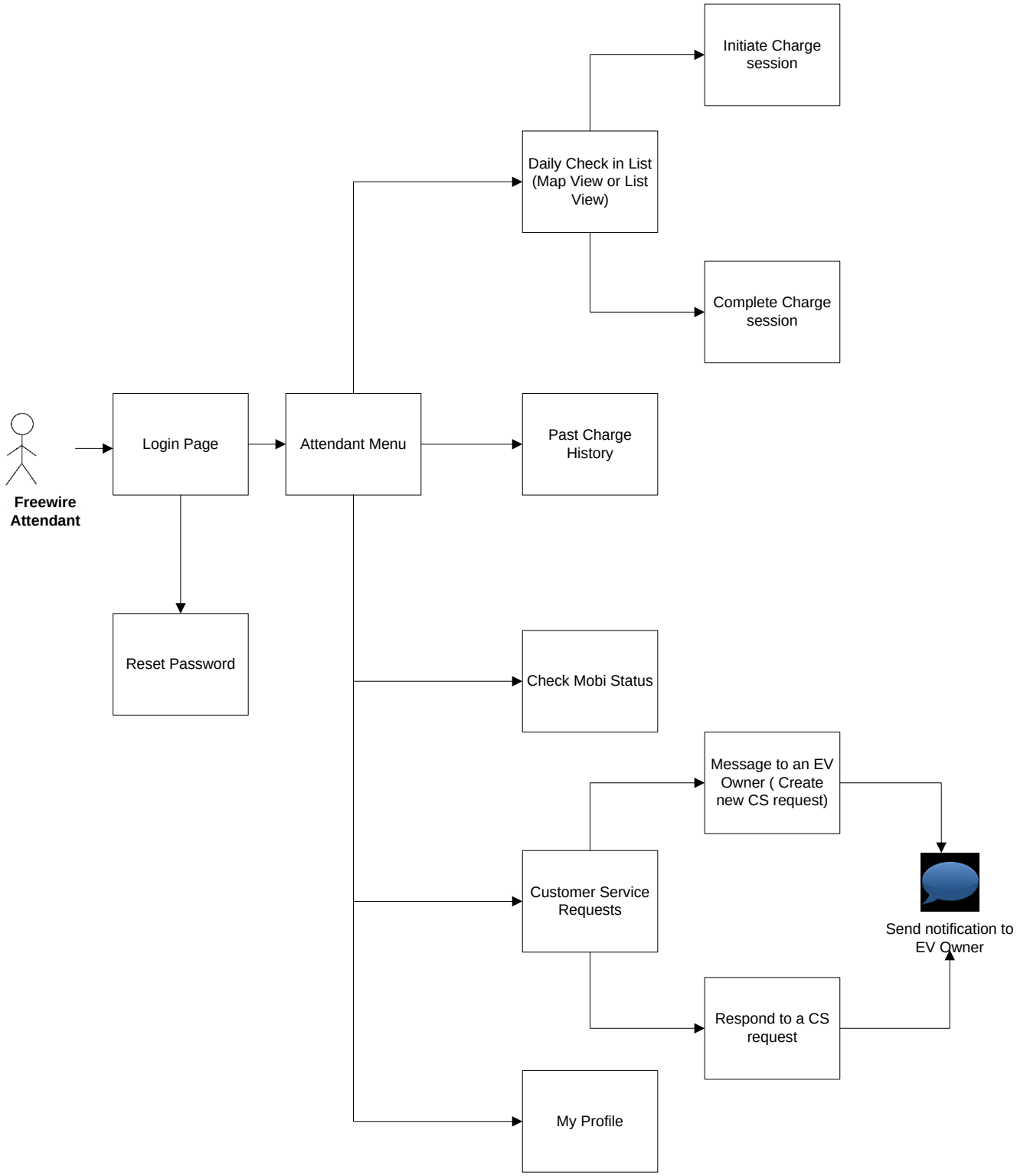
Ability to change current password.

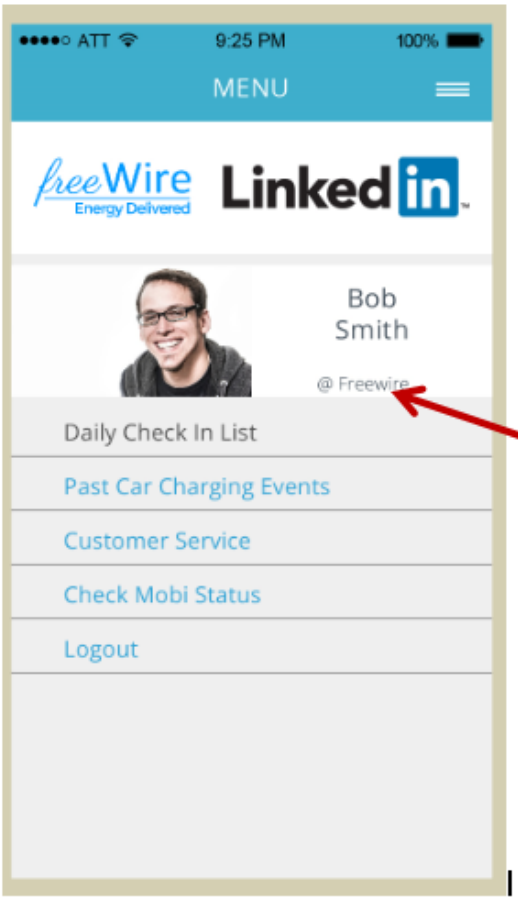
Set notifications on or off. (This only turns off the active notification via a beep or vibration. The notifications and history is still maintained in the notifications menu.


ATT
9:25 PM
100%
MY DETAILS
Amanda Jones
DETAILS
First Name
Amanda
Middle Name
Lucy
Last Name
Jones
Email Adress
amanda.jones@linkedin.com
Mobile Phone
408 888 8888
EV Type
Nissan Leaf
Vehicle Color
Blue
EV License Plate
4ABC543
Your Picture
Total Charging instances - Display only
Total Energy Delivered - Display only
SAVE CHANGES
Ability to upload a new picture.
This is total number of times the charge has been completed for the user.
The total of all the energy delivered over the history.

I

Freewire Attendant Use Cases

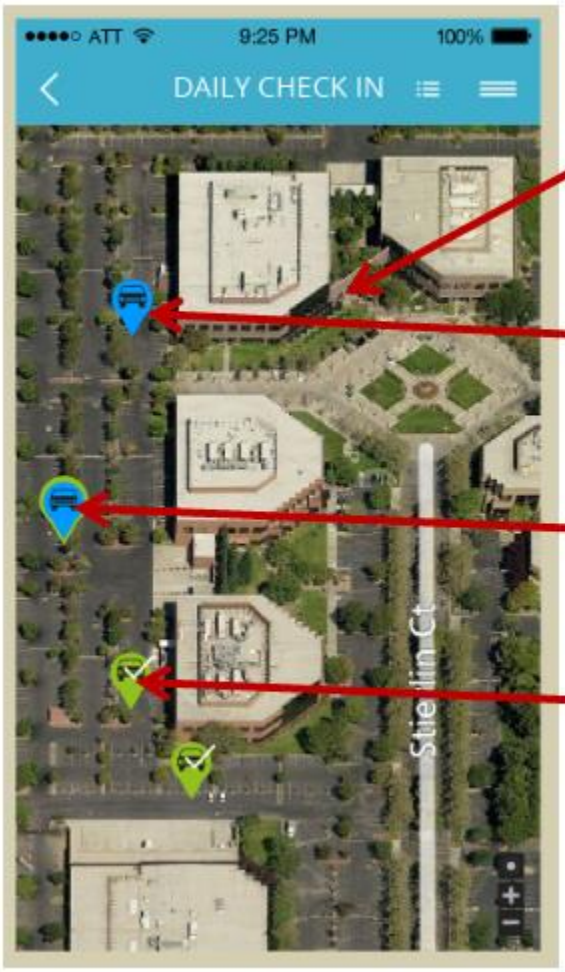




The system determines the Freewire attendant based on the user type.

Daily check in list

The freewire attendant uses this screen to find all the users that have checked in and where they are parked at.



When the freewire attendant selects the daily check in page, he sees the same map.

All the vehicles that have checked in for the day are visible as tags.

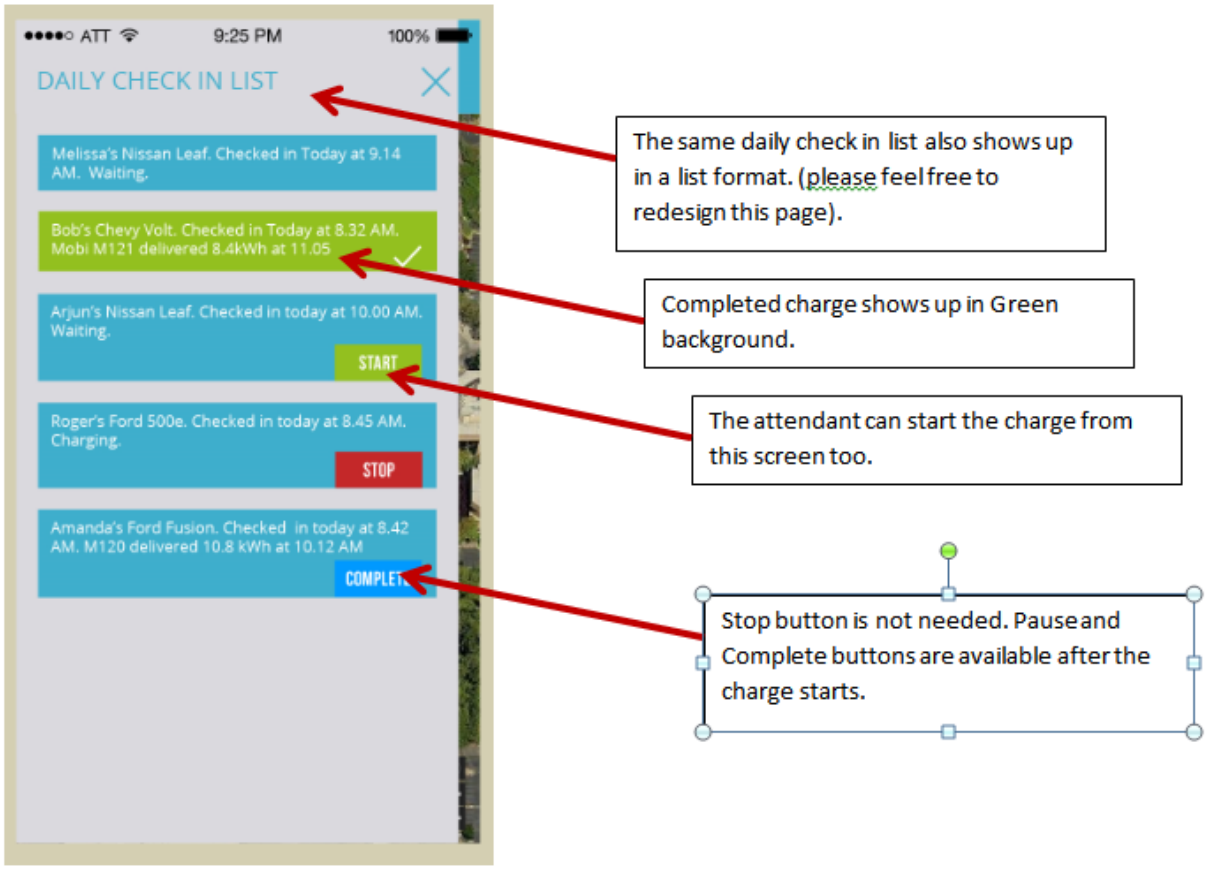
The vehicles being charged are shown with blinking heartbeat

The vehicles being charged are shown with blinking heartbeat

By touching the tag, a box opens up with some more details about the car and

After the start button is pushed, the start button disappears and 2 other buttons need to show up. Pause button and Complete button. (add those buttons and remove the stop button).

The Tag icon starts a background green heartbeat when the vehicle is being charged. (any other way to indicate the charging action would be fine too as long as it looks good.)



Use Case – Daily charge schedule

Description : Attendant checks the list of the check ins to find the vehicles that need to be charged. The check in list lets the attendant know which vehicles have already been charged and which ones are not, along with where they are parked at. The app allows him to look at list view or a map view. Attendant identifies the next vehicle to be charged.

.

Mobile Experience:

* When the attendant opens the app, the user name and password is auto saved. He hits the login button.
* The screen also provides 3 options for the freewire attendant.
  - Daily Check In list
  - Past car charging events
  - Customer Service Request
  - Check Mobi Status

* The attendant selects the "Daily Check in list" option
* The app shows a map view and a list view of the checked in vehicles. It also includes the vehicles that have already been charged today and the ones that are waiting.
* By clicking on a car check in record, it shows who does the car below to and where is it parked in the lot. What time was checked in etc.

Use Case – Initiate Charge Session

Description : Attendant locates the vehicle to be charged and plugs in Mobi to start the charging action. He then opens up the app on his device to send a notification to the EV Driver with optional comments.

.

Mobile Experience:

Initiate charge session is an option available when the attendant clicks on a car check in record. The attendant initiates a charge after he plugs in a charger to the vehicle.
* When the attendant clicks the "initiate charge" and confirms it, the app sets the status of the check in as charging.
* The app also triggers an alert to the user that the charge has been initiated.

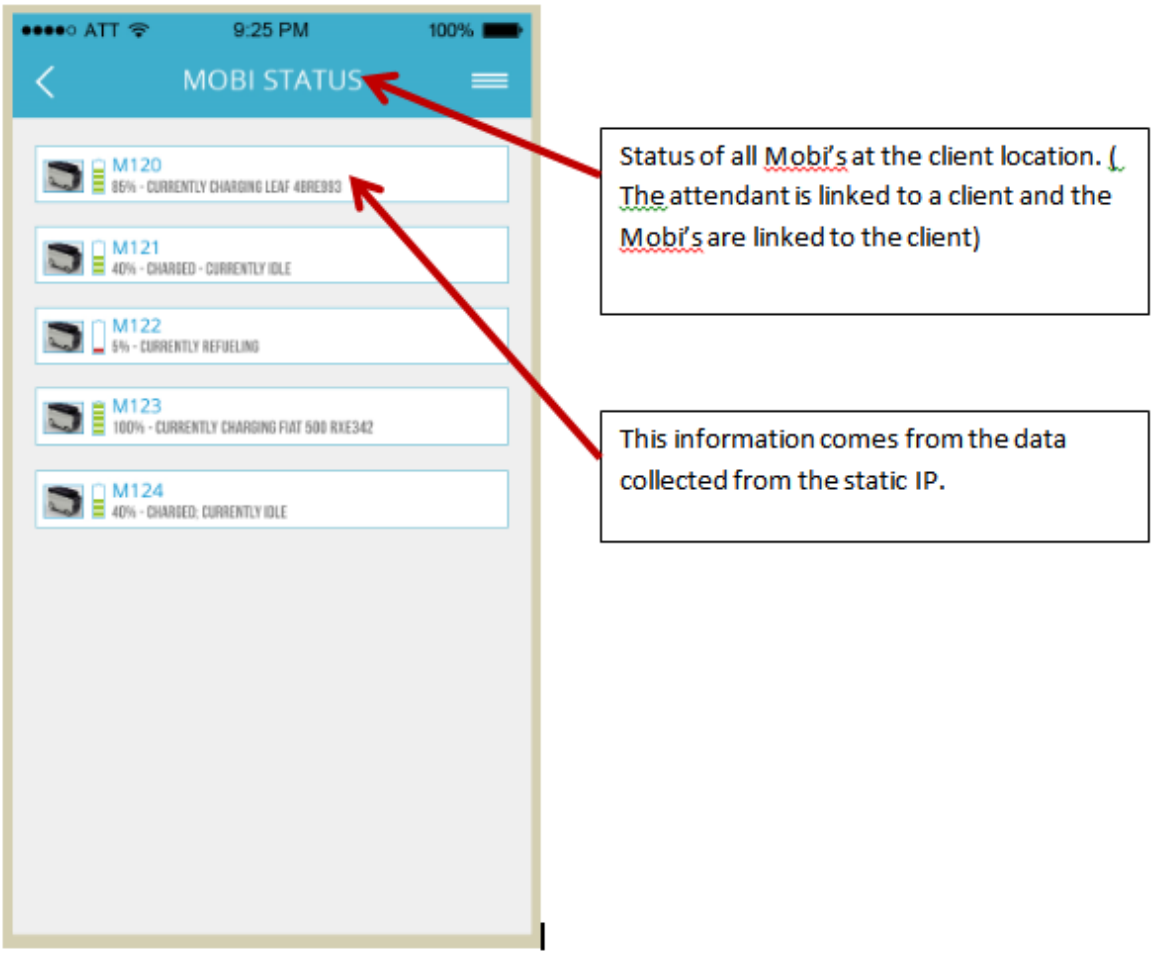
Use Case – Complete Charge Session

Description : Freewire attendant completes the charging session and sends a notification along with optional comments for the EV Owner

Complete charge session is an option available when the attendant clicks on a car check in record that is already in the charging status. The attendant initiates a charge after he unplugs the charger from the vehicle.
* When the attendant clicks the "Complete charge" and confirms it, the app sets the status of the check in as completed.
* The app also triggers an alert to the user that the charge has been completed.

Use Case – Check Mobi Statuses

Description : Typically one attendant will oversee multiple Mobi units. This feature allows him to look at the state of each mobi -
- Which Mobi is currently charging a car
- Which Mobi is idle
- Which Mobi is done charging but needs to be unplugged
- Which Mobi is getting refilled (Getting itself charged)



Mobile Experience:

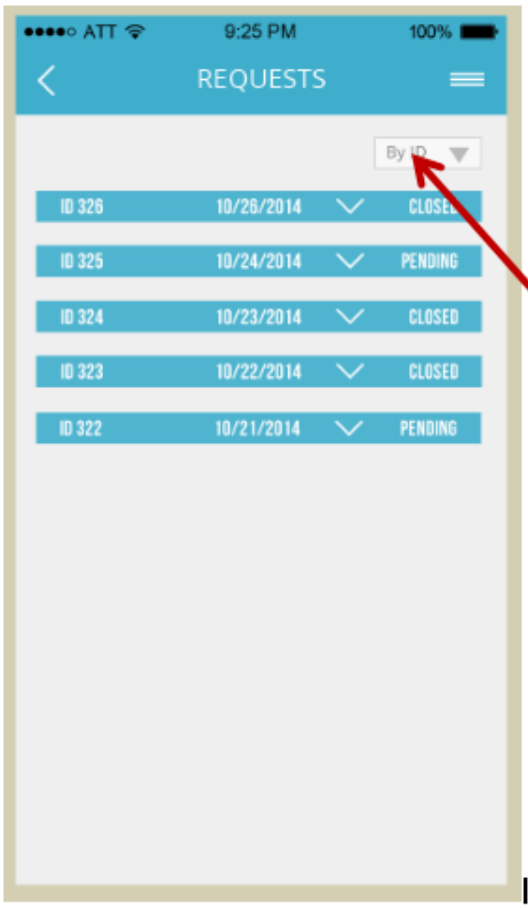
The screen shows are list of all Mobis at the client site and their status. Something similar to the following.

Use Case – Respond to CS request

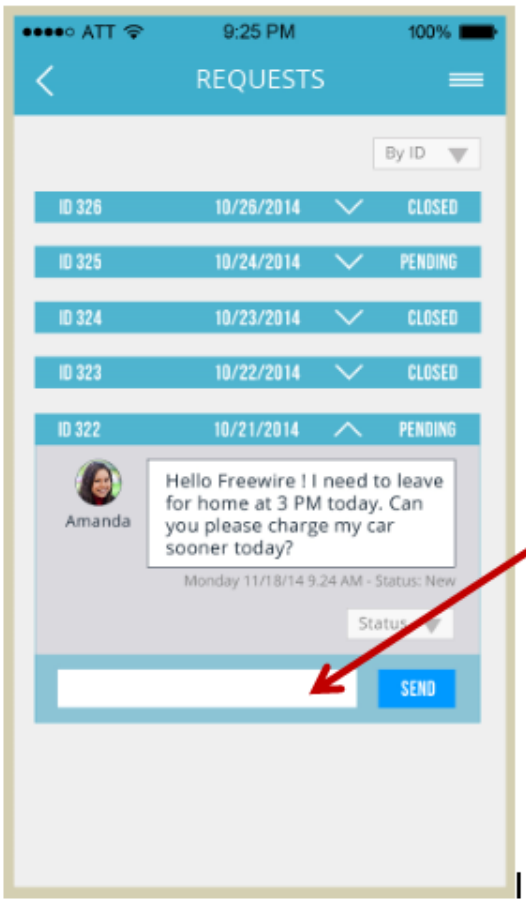
Description : This allows the freewire attendant to respond to a service request.

Mobile Experience:

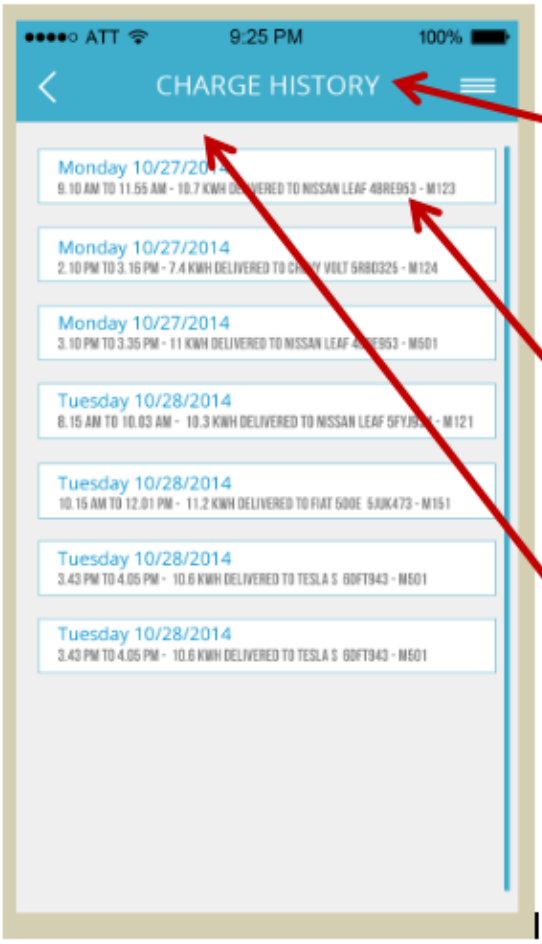
* When a CS request is submitted, the attendant gets an alert from the app about a new submission. The same alert is trigerred when a new message is posted for an existing app.
* The attendant types up a response and sents it back to the user.
* The attendant may change the status of the request.



Attendant can see all the CS requests for the client. He can sort it by ID, by date, by user or by status.



Use Case – Check charge history

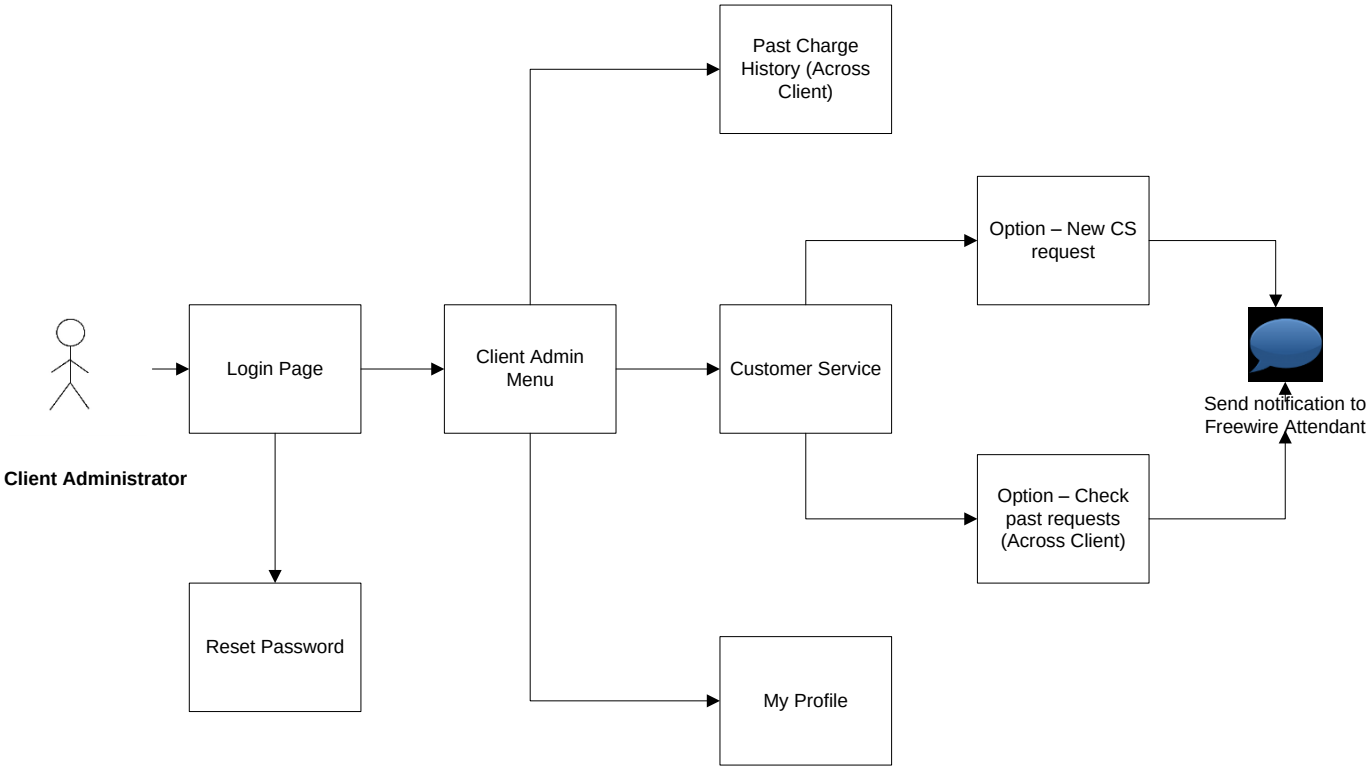


Attendant can see all the charge instances for the client (unlike the EV owner who can see only the charge events of her car only).

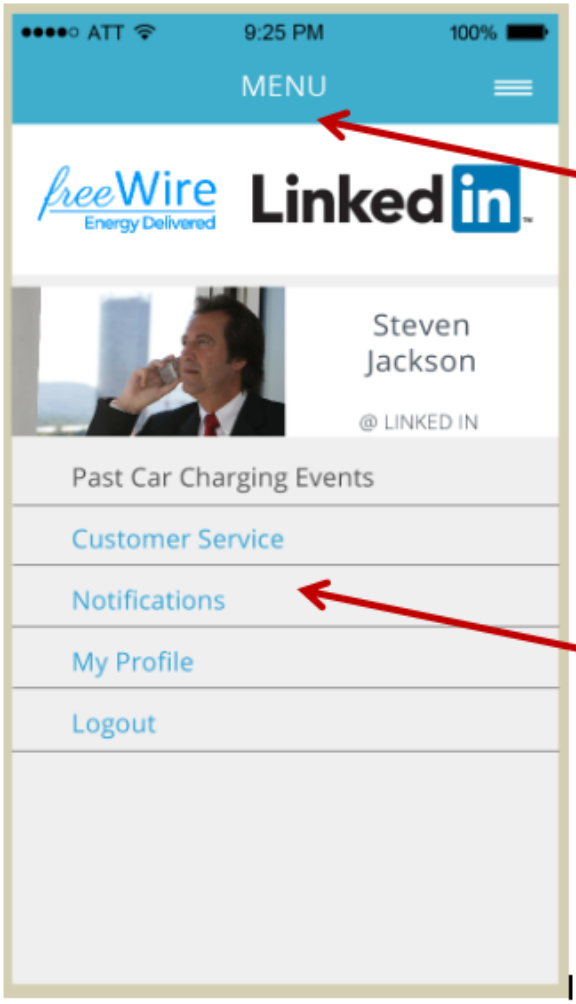
The actual car information is also available for the attendant.

Need some ability to sort by Day (current view) or sort by the Mobi (the M123 etc number here) or sort by the car license plate number.

Client Administrator use cases :



Client administrator is a customer employee who is responsible to make sure that the EVs are charged daily as per the agreed contract. He gets the similar access as the Freewire attendant except he has access to only Customer service and Past History functions for the client location. The Client admin does not have check in feature.



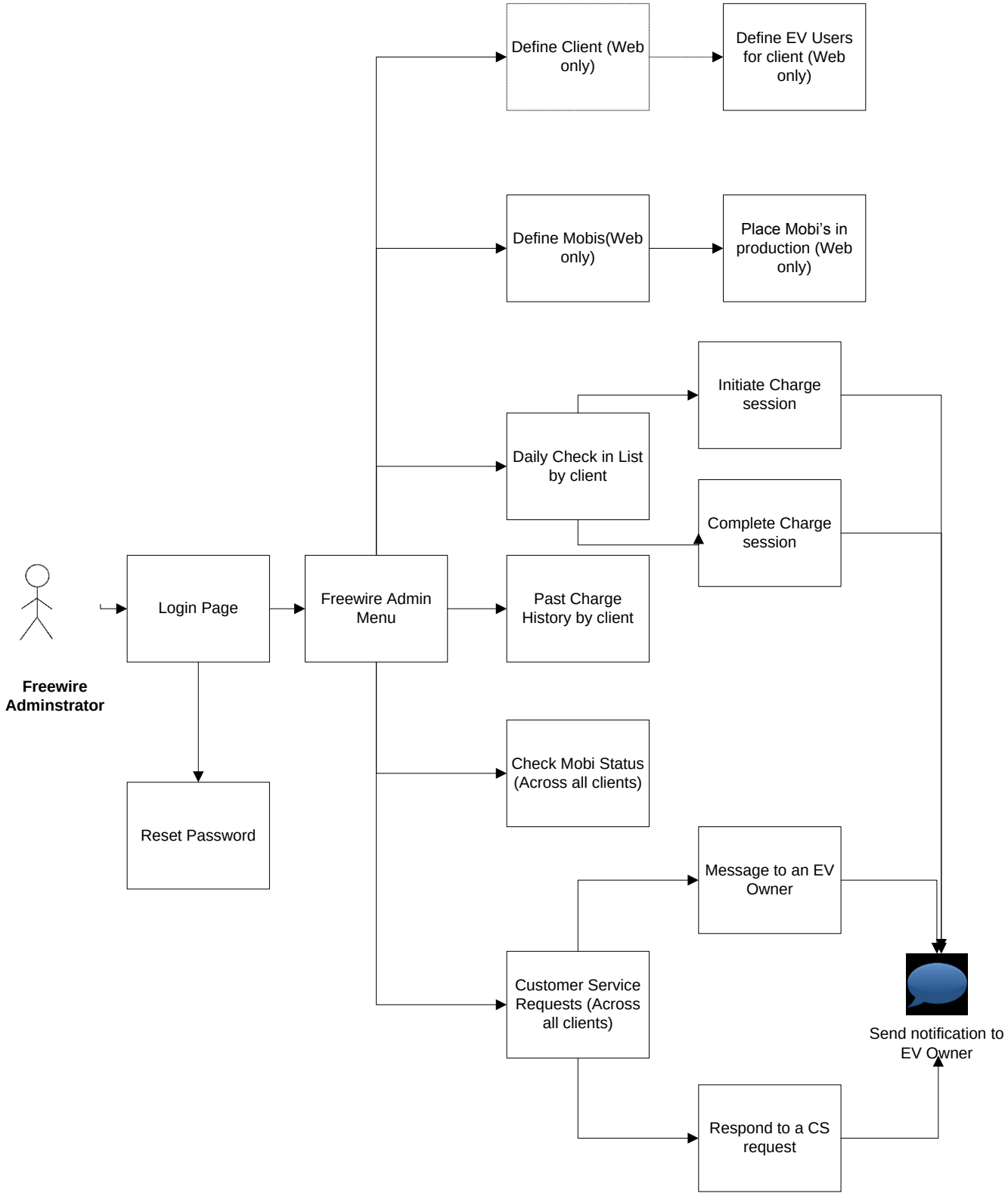
Menu for the Client Administrator

Notifications are not needed. Please remove

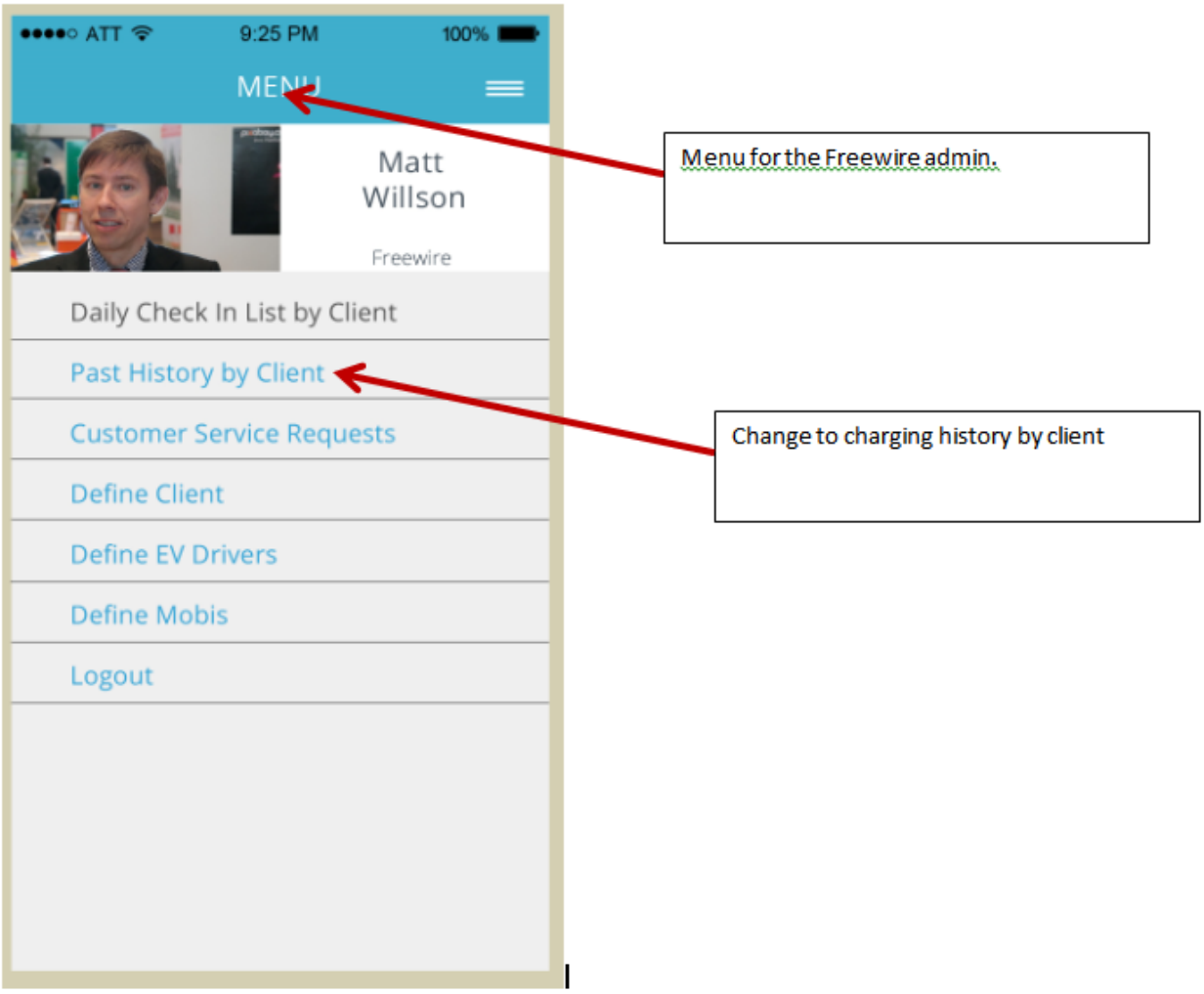
The customer service, past car charging events and my profile page are similar to the Freewire Attendant and are not repeated here.

Freewire Administrator use cases:

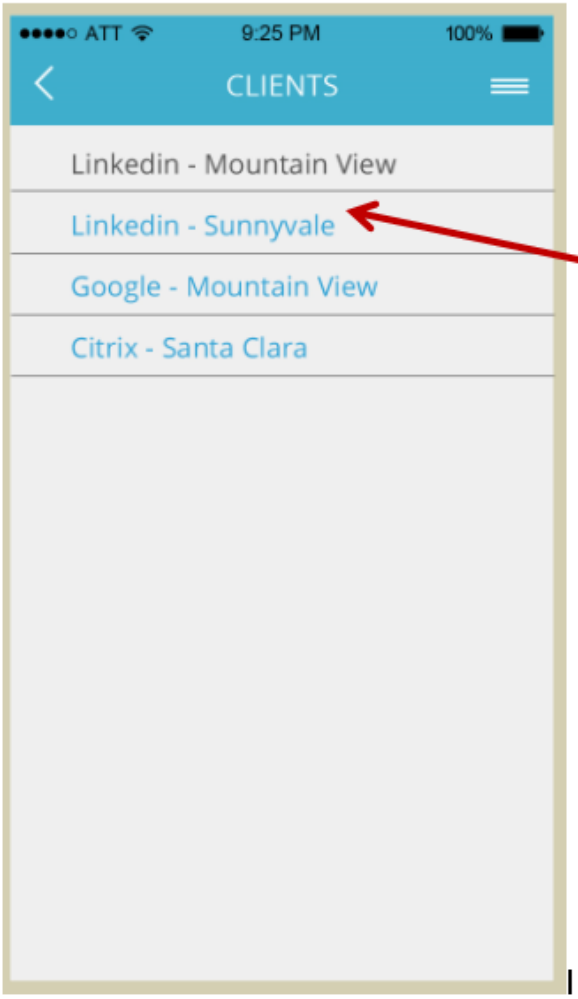
This is a super user + system administrator and has full access to do everything. In addition, he also has the ability to define Client, Define EV owner / users, Define Mobi units.



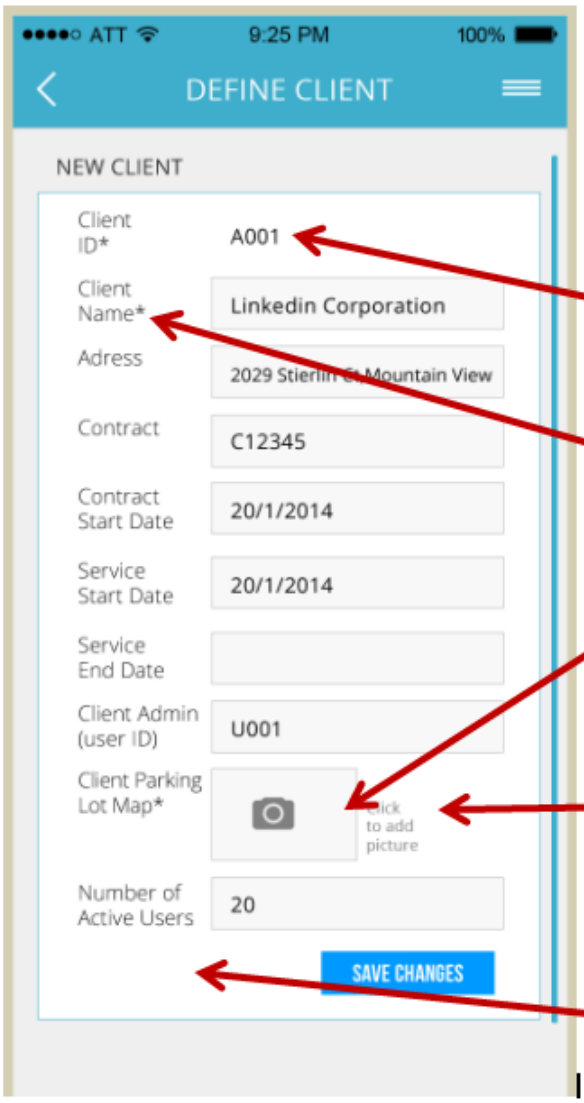
Use Case : Define Client – The objective of this is for the admin to define the client in the Freewire system. The admin can also view all the clients in the system and change the settings as needed.



The freewire admin can see the check in list, CS request and Charging history same as the Freewire attendant, except the admin has access to all clients. So when the admin selects any of those menu options, a sub menu opens up which lists all the clients in the system . The admin selects the client and then enters the relevant screen.



Use Case – Define Client : The admin defines client in the system. In addition to other details, he also uploads the map of the parking lot and the logo of the client.



Client ID is automatically generated by the system.

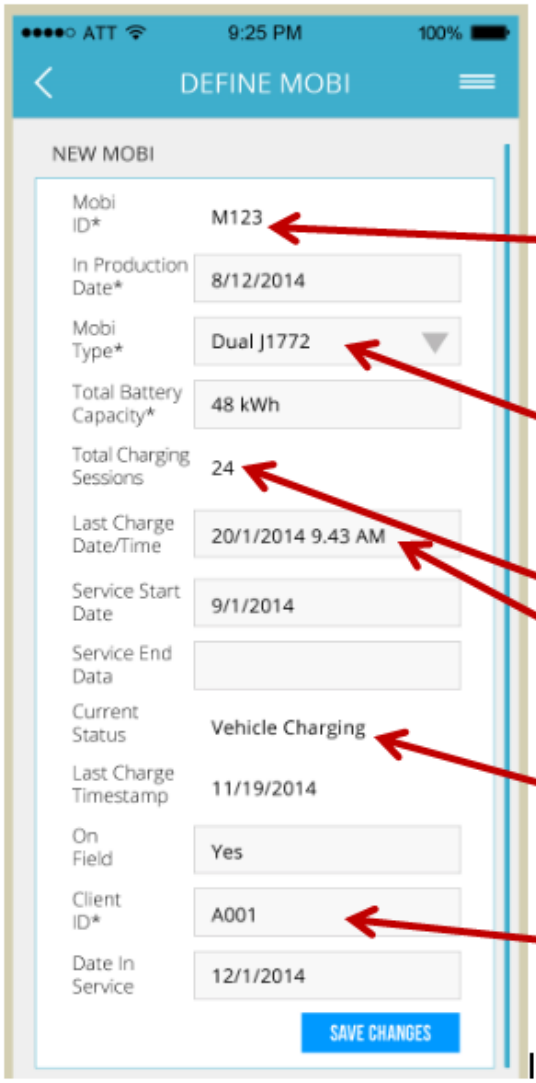
Mandatory fields have *

The admin uploads the map of the client's parking lot as an image. This map shows up when the EV owner checks in.

Need another field to add the client's logo.

Need a button to delete record. Deletion is only possible if no EV owner or Mobi is linked to the client

Use Case : Define Mobi



Mobi ID is entered value. Please change this field to open.

There are 4 types of Mobi.

- J1772
- Dual J1772
- Chademo
- Sae Combo

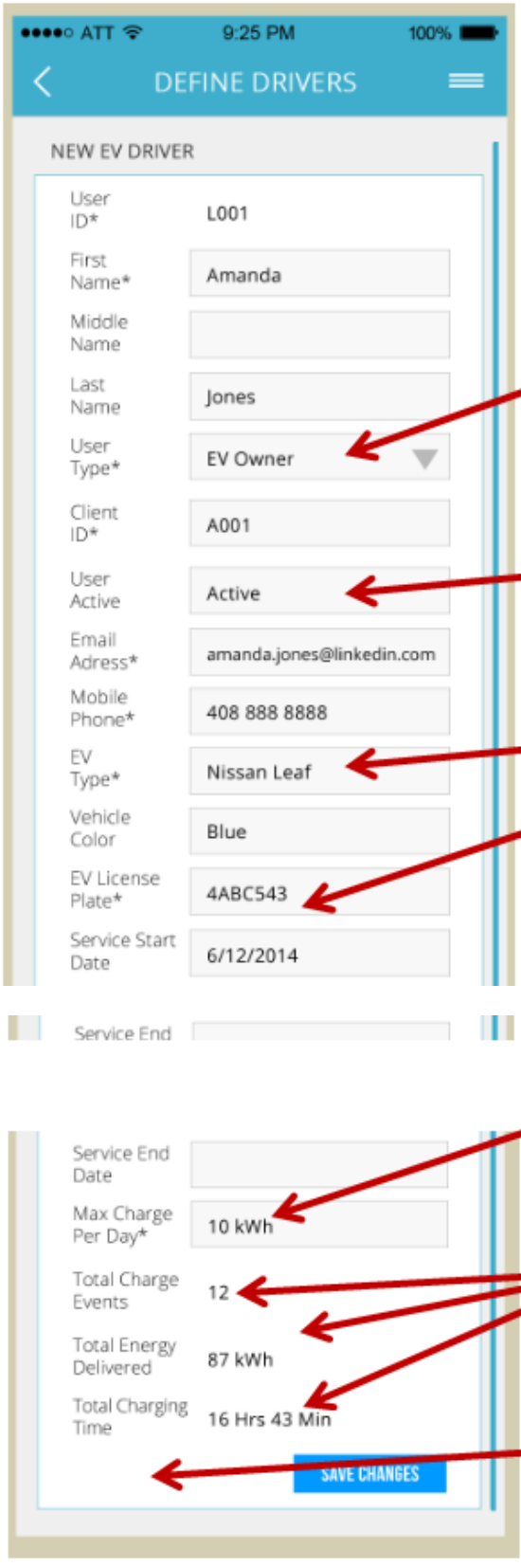
Calculated by the system

Calculated by the system. Please change the field to gray

From the data feed collected from the static IP

This is not a mandatory field. Please change. The mobi is linked to a client from this field.

Use Case : Define Users



There are 4 types of user types.

- EV Owner
- Freewire Attendant
- Client Administrator
- Freewire Administrator

Change this field to a drop down. The possible values are

- Active
- Inactive

EV type / EV license plate is only mandatory for EV owner

This is the allocated charge for the EV owner / day. Beyond this the freewire system generated a notification for the attendant

These values are stored in the system based on the charging events.

Need to add a button to delete record. A user can only be deleted if there is no charge history or check in linked to the user.

Notifications.

Following notifications are triggered by the app itself.

1. EV Driver / Employee :
    a. When the charge is started by the attendant
    b. When the charge is completed by the attendant
    c. When a response to a CS request is received.
2. Freewire Attendant :
    a. When a charge is completed for a vehicle.
    b. When a new CS request or an update to an existing CS request is received.
    c. When the total charge amount reaches the daily charge allowance in the EV driver profile.

Database tables

Following are some suggested tables, but these are by no means indicative of the final DB requirements. These are just some suggestions for the DB and program logic design.

Table - Client				
Field	**Mandatory**	**Description**	**Comments**	**Example**
Client ID	Yes	Unique id for the Freewire's Customer	Generated by the system.	A001
Client Name	Yes	Company Name	Manually Entered	Linkedin Corporation
Address	No	Address of the service location	Manually Entered	2029 Stierlin Ct, Mountain View
Contract	No	Contract / SOW Number	Manually Entered	C12345
Contract Start Date	No	Freewire Contract Date	Manually Entered	20/1/2014
Service Start Date	No	The agreed start date of the service	Manually Entered	5/1/2014
Service End Date	No	The date service ended	Manually Entered	4/30/2017
Client Administrator (user ID)	No	The Freewire user id of the client	From table User	U001

		administrator		
Client Parking lot Map	Yes	Default parking lot map linked to the Client. JPEG ?	All users from this client will see this map on the check-in screen	TBD
Number of Active Users	No	Number of active Freewire users based on the contract	The system will not allow active users for the client that exceeds this limit	20

Table - User				
Field	**Mandatory**	**Description**	**Comments**	**Example**
User ID	Yes	Unique user id	Generated by the system.	L001
First Name	Yes	First Name	Manually Entered / Editable by user This name is used by the App the greet after login and for the Customer service forms	Amanda
Middle Name	No	Middle Name	Manually Entered / Editable by user	
Last Name	Yes	Last Name	Manually Entered / Editable by user	Jones
User Type	Yes	There are 4 possible user types - EV Owner - Client Administrator - Freewire Attendant - Freewire Admin	A user can have one or many user types.	EV Owner
Client ID	Yes if the user type is - EV Owner - Client Administrator - Freewire Attendant	Unique id for the Freewire's Customer	From the table - Client	A001
User Active	Yes if the user type is - EV Owner	Indicates if the user is an active member of the freewire program. Values : Active or Inactive	The inactive users are not allowed the check in action	Active

Email Address	Yes	Email address of the user	Manually Entered / Editable by user. This will is where email notifications will be sent.	amanda.jones@tbd.com
Mobile Phone	Yes	The mobile phone number of the user where the app will be initiated	Manually Entered / Editable by user	408 888 8888
EV Type	Yes if the user type is - EV Owner	The electric vehicle that the user drives	Manually Entered / Editable by user	Nissan Leaf
Vehicle Color	Yes if the user type is - EV Owner	The color of the EV	Manually Entered / Editable by user	Blue
EV License Plate	Yes if the user type is - EV Owner	The license plate of the EV	Manually Entered / Editable by user. If it's a new vehicle, then NEW will be used.	4ABC543
Service Start Date	No	The date when the user joins the freewire program	The user will not be allowed to check in before this date. Field is updated only by Freewire Admin. Shows up as uneditable on user profile	6/12/2014
Service End Date	No	The date when the user leaves the freewire program	The user will not be allowed to check in after this date. Field is updated only by freewire admin	6/10/2017
Max Charge permitted / Day	Yes if the user type is - EV Owner	The maximum charge that the user is allowed to get each day. When this limit is reached, it will trigger an alert to the freewire admin to unplug the charger from the vehicle	Not visible to the user	10 kWh
Total Charge Events	No	The number of times the vehicle has been charged by the freewire program	Visible but not editable by the user. The program increments this value by 1 each time the freewire attendant triggers a charge initiate notification for the vehicle	12

Total Energy Delivered	No	This is the total of the energy delivered to the vehicle	Visible but not editable by the user. The program increments this value by the actual charge amount for the charge event	87 kWh
Total Charging time	No	Total time spent charging the vehicle	Visible but not editable by the user. The program increments this value by the charge intiate - charge complete notifications triggered by the freewire attendant	16 Hrs 43 Min

Table - Mobi				
Field	**Mandatory**	**Description**	**Comments**	**Example**
Mobi ID	Yes	Unique serial number for the Freewire's Mobi Unit	Manually Entered	M123
In production date	Yes	Date when the Mobi is placed in service	Manually Entered	8/12/2014
Mobi Type	Yes	3 possible Values - J1772 - Dual J1772 - Chademo	Manually Entered	Dual J 1772
Total Battery Capacity	Yes	Max battery capacity of the unit	Manually Entered	48 kWh
Total Charging Sessions	No	Total number of charging events the Mobi has been engaged in	Calculated by system based on the charge initiate notifications triggered from the Mobi	24
Last Charge date / time	No	The date and time stamp when the last 'state of charge' = 'Grid Charging' was received	Based on the Mobi state of charge data captured from the broadcast source	20/1/2014 9.43 AM
Service Start Date	No	Date when the Mobi is placed in service	Manually Entered	9/1/2014
Service End Date	No	The date service ended	Manually Entered	
Current State	No	Based on the last received state. 4 Possible values - True Off/Deep Sleep - True Idle - Grid Charging - Vehicle Charging	Based on the Mobi state of charge data captured from the broadcast source	Vehicle Charging

Last state timestamp	No	Date and timestamp	Based on the Mobi state of charge data captured from the broadcast source	11/19/2014
On Field	No	Indicates if the Mobi is linked to a client	Boolean - Yes or No	Yes
Client ID	Yes if In Service is yes	Client ID	Manually entered. From table Client	A001
Date in Service	Yes if In Service is yes	Date when the Mobi is placed in service at client	Manually Entered	12/1/2014

Table - Check-In				
Field	**Mandatory**	**Description**	**Comments**	**Example**
User ID	Yes	User ID of the EV driver checking in	Captured from the Check-in action done by the user. Refer table User	L001
Check in Date and Time	Yes	Timestamp of the check in event	Captured from the Check-in action done by the user.	11/19/2014 9.14 AM
Check in Location	Yes	The exact location of the check in	Captured from the Check-in action done by the user. Format : TBD	TBD
Service Completed	No	This field indicates if the charge service was completed by the attendant for the check-in Boolean : Yes / No	Captured based on the service completed event done by the attendant for the check-in record	Yes
Charge initiated	No	Timestamp of the charge intiate event for the check-in	Captured based on the Charge initiated event trigerred by the attendant	11/19/2014 12.32 PM
Charge completed	No	Timestamp of the charge completion event for the check-in	Captured based on the Charge complete event trigerred by the attendant	11/19/2014 3.02 PM
Attendant ID	No	The user ID of the attendant charging vehicle	Captured by the Mobile app action of the attendant	U023
Mobi ID	No	The Mobi ID of the Mobi Charger serving the check in	Captured by the charge completed event trigerred by the attendant	M123

Charge Delivered	No	The actual charge delivered by the Mobi	Based on the Mobi state of charge data captured from the broadcast source	10.5 kWh

Table - Customer Service Request Header				
Field	**Mandatory**	**Description**	**Comments**	**Example**
Request ID	Yes	Unique id generated by the system for each CS request	The Id is auto generated.	CS001
Created By	Yes	User Id of the person who created it	Captured from the create new CS request action	U001
Client ID	Yes	System captures the client id linked to the user ID of the person creating it	Captured from the create new CS request action	A001

Table - Customer Service Request Lines				
Field	**Mandatory**	**Description**	**Comments**	**Example**
Request ID	Yes	Unique id generated by the system for each CS request	The Id is auto generated.	CS001
Record type	Yes	New or Update	Indicates if this is a new CS request or and update to existing one	Update
Timestamp	Yes	Timestamp of the update		11/21/2014 9.15 AM
Status	Yes	Status types - New - In Process - Closed	When the user creates a new request, the system sets the status as NEW. When the Attendant replies to it, then the status changes to In Process. When it is resolved, the status is changed manually by the Attendant to CLOSED.	In Process
Message	Yes	The actual text message typed		Hi Freewire. Can you please charge my car sooner today ? I need to leave early. Thanks.'