

Use Cases:
Delivery Tracker

Mark Whitney
Dec. 7, 2009

Table of Contents

Use Case 1: Powering the Device	3
Figure 1.1 – Device top view	5
Figure 1.2 – Device side/bottom view	6
Use Case 2: Log-in	7
Figure 2.1 – Log-in screen	9
Figure 2.2 – Log-in screen with error message	9
Use Case 3: Calling up List of Deliveries	10
Figure 3.1 – Deliveries List	11
Use Case 4: View Map of Delivery Locations	12
Figure 4.1 – Map screen	13
Use Case 5: View Current Location	14
Figure 5.1 – Map screen	15
Use Case 6: Record Package Pick-up	16
Figure 6.1 – Pick-up screen	19
Figure 6.2 – Pick-up screen with scanner message	19
Figure 6.3 – Pick-up screen with error message about scanner	19
Figure 6.4 – Pick-up screen with customer-signature box.	19
Use Case 7: Record Package Drop-off	20
Figure 7.1 – Drop-off screen	23
Figure 7.2 – Drop-off screen with scanner message.	23
Figure 7.3 – Drop-off screen with customer-signature box.	23
Use Case 8: Record Failed Pick-up Attempt	24
Figure 8.1 – Pick-up screen with "No" answer	25
Figure 8.2 – Pick-up screen with choice of reasons	25
Figure 8.3 – Deliveries List with data on failed pick-up	26
Figure 8.4 – Deliveries List with data on several failed pick-ups	26
Use Case 9: Record Failed Drop-off Attempt	27
Figure 9.1 – Drop-off screen with "No" answer	28
Figure 9.2 – Drop-off screen with choice of reasons	28
Figure 9.3 – Drop-off screen with list of failed pick-ups	29
Use Case 10: View Delivery Record	30
Use Case 11: View Alerts from Headquarters	31
Figure 11.1 – Alert message	33
Use Case 12: Accept/Decline Requests from Headquarters	34
Use Case 13: Back-up Menu	36
Figure 13.1 – Back-up menu	37
Use Case 14: Log-out	39
Figure 14.1 – Log-out screen	40

Use Case 1: Powering the Device

Description: The Delivery Tracker is a handheld device that allows a person using a bicycle to access and record information about deliveries being made.

Actor: Bicycle delivery driver.

Precondition: The user is holding the device, which is turned off.

Post-condition: The user is viewing the log-in screen.

Primary Scenario: The user presses the recessed power switch with the stylus.

Primary Task Flow:

1. The user grasps the stylus and extends the retractable wire.
2. User inserts writing end of stylus into recess that contains the power switch.
3. User presses power switch.
4. Screen lights up to signal power is on.
5. Log-in page loads.

Alternate Scenario 1: 1. The stylus is missing.

Alternate Task Flow 1: 1. The user performs the same steps outlined in the primary scenario using a spare, unattached stylus carried as a spare.

Alternate Scenario 2: 1. The stylus is missing and the user has no spare.

Alternate Task Flow 2: 1. The user performs the same steps outlined in the Primary Scenario using a ball-point pen or similar device.

Technical Specs: **Device Construction**

1. The Delivery Tracker body is made of metal with an outer lining of shock-absorbent, water-resistant rubber.

2. The power switch is recessed to avoid being triggered by accidental bumping of the device. The power switch is padded with soft rubber to protect it from damage by improper use of the stylus or other object.
3. The touch screen allows finger-tip scrolling through lists and across maps, and finger-tip pressing of the log-in and menu screens, but requires use of the stylus for entry of customer signatures.
4. The touch screen is protected by thin, replaceable scratch-resistant covers which do not interfere with the touch-screen functionality.
5. The hard plastic stylus is attached to the device with a retractable, spring-wound wire that allows no part of the wire to dangle outside the frame of the device.

Figure 1.1 – Top view of Delivery Tracker

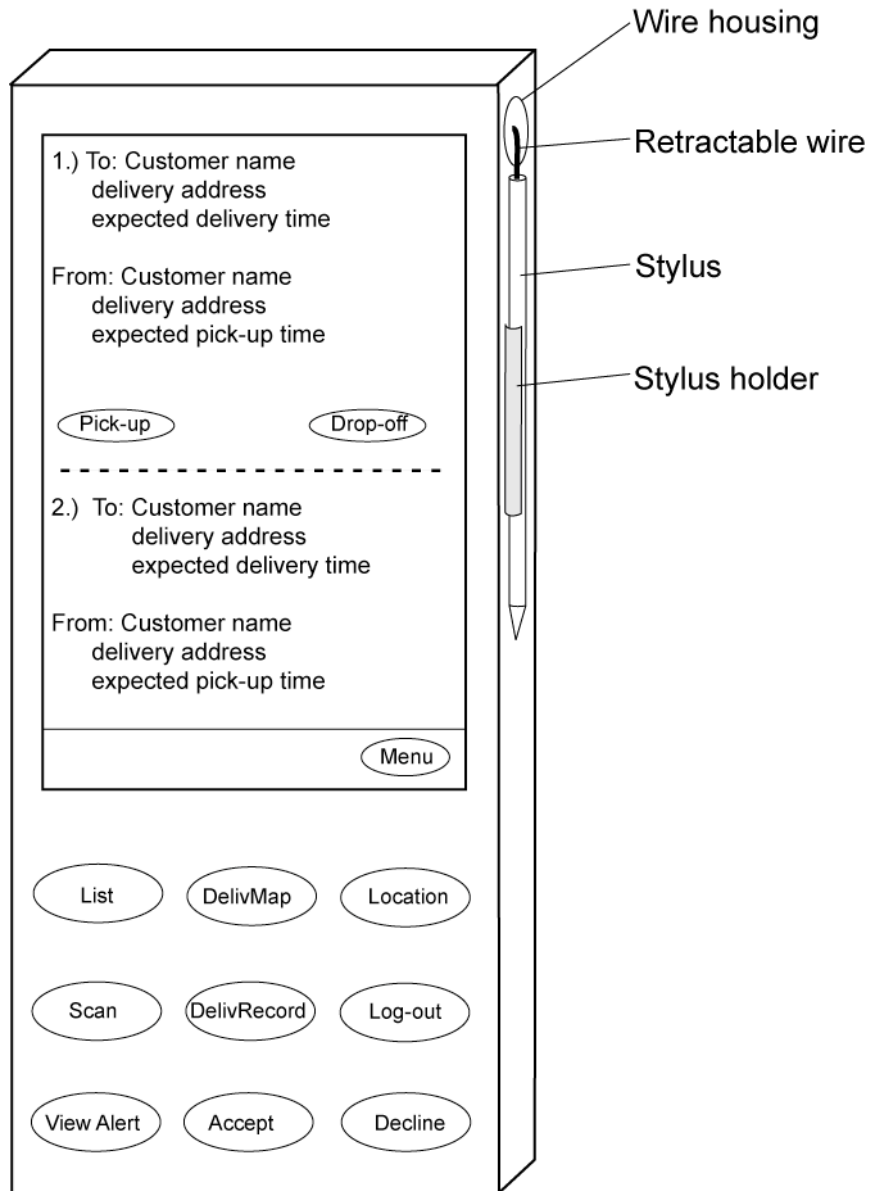
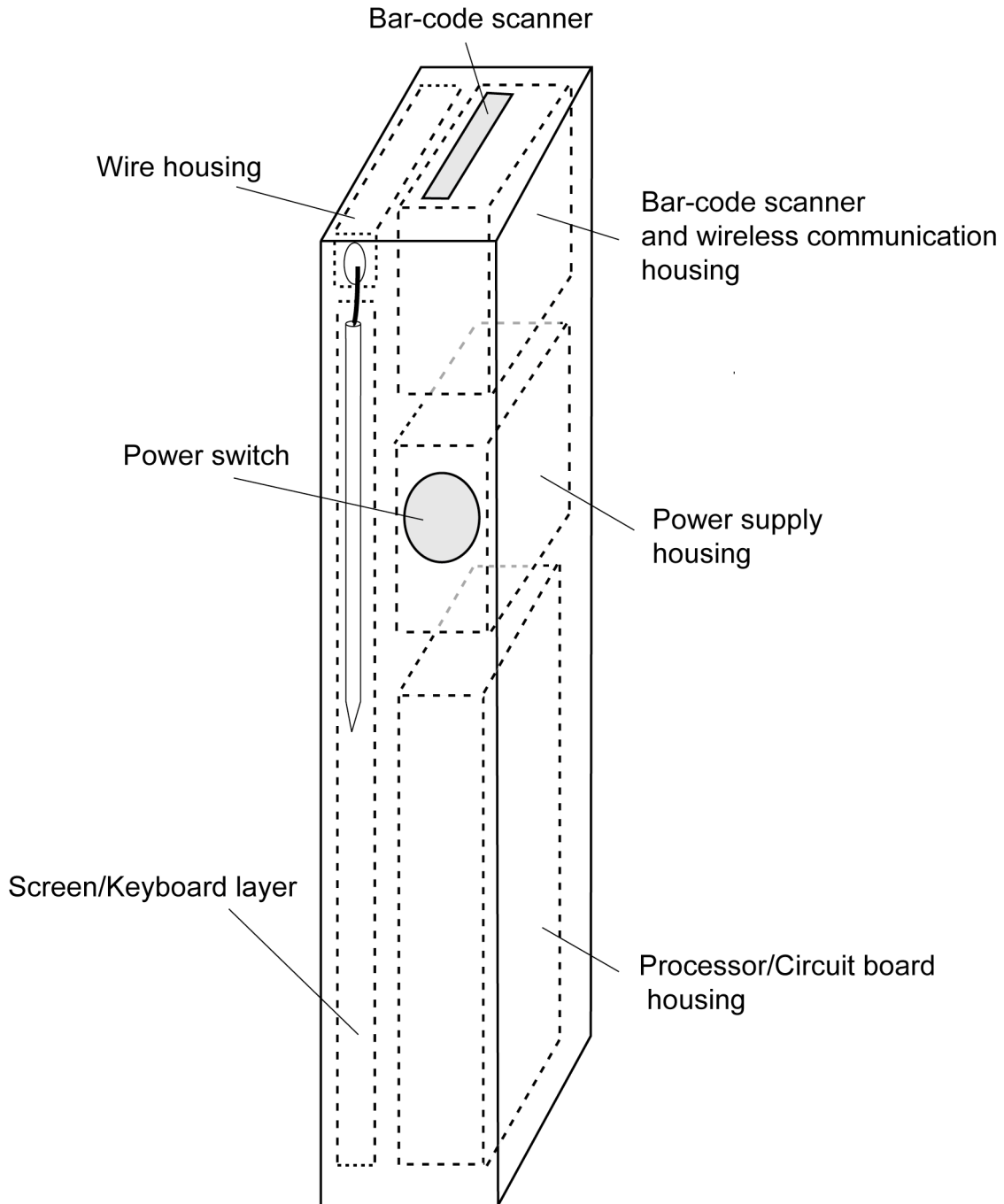


Figure 1.2 – Side view of Delivery Tracker



Use Case 2: Log-in

Description: The Delivery Tracker has a log-in function that requires the user to select an identification code and password from successive tables of numerals on the touch screen.

Actor: Bicycle delivery driver.

Precondition: The user is viewing the log-in screen.

Post-condition: The user is viewing the welcome screen.

Primary Scenario: The user enters a five-digit identification code followed by a three-digit password.

Primary Task Flow:

1. The user enters five numerals in sequence.
2. Device auto-submits the entry.
3. Password screen loads.
4. The user enters three numerals in sequence.
5. Device auto-submits the password entry.
6. Welcome screen loads.

Alternate Scenario 1: The user enters an identification number that doesn't correspond to a valid identification code in the database.

Alternate Task Flow 1:

1. The user enters five numerals in sequence.
2. Device auto-submits the entry.
3. Error message loads.
4. User re-enters identification code.
5. Device auto-submits the entry.
6. Password screen loads.

Alternate Scenario 2: The user enters a password that doesn't correspond with the identification code previously entered.

Alternate Task Flow 2:

1. The user enters three numerals in sequence.
2. Device auto-submits the entry.

3. Error message loads.
4. User re-enters password.
5. Device auto-submits the entry.
6. Welcome screen loads.

Technical Specs:

AJAX

1. A script detects and records the five numerals entered for the identification code and checks the value against current valid values in the database. If valid, the script loads the password screen. If the value is not valid, the script loads an error message.
2. A script records the three numerals entered for the password and checks that value against the current valid password for the identification code in the database. If valid, the script loads the welcome screen. If the value is not valid, the script loads an error message.
3. A script records the number of attempts made to enter the identification code, and if five failed attempts are made, the script locks out the user and displays an error message. The device will not unlock until instructions to do so are sent to it by headquarters. If the device remains idle for 30 minutes after locking up, a script will detect this and shut down the power.
4. A script records the number of attempts made to enter the password, and if five failed attempts are made, the script locks out the user and displays an error message. The device will not unlock until instructions to do so are sent to it by headquarters. If the device remains idle for 30 minutes after locking up, a script will detect this and shut down the power.
5. A script detects the user re-entering the user name or password and unloads the error message.

Display

1. The identification code and password screens show the numerals 0-9 across two lines, each line containing five numerals, centered.
2. The welcome screen contains the text: "Welcome (User Name)" in the center and a "Menu" button in the bottom right corner.

3. The error message contains an icon and the text of the error message centered on the screen.
4. The icon appearing in error messages consists of a triangle containing an exclamation point.

Error Messages

1. "The identification code you entered is not in use. Please re-enter your identification code."
2. "The password you entered is not valid for your identification code. Please re-enter your password."
3. "This function is locked for security reasons. If you are a valid device user, call XXX-XXX-XXXX for assistance."

Welcome Message

1. "Welcome (user name)"

Figure 2.1 – Log-in screen for identification code

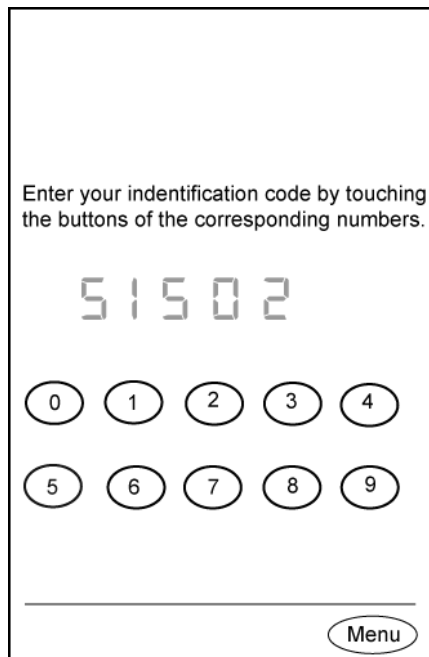
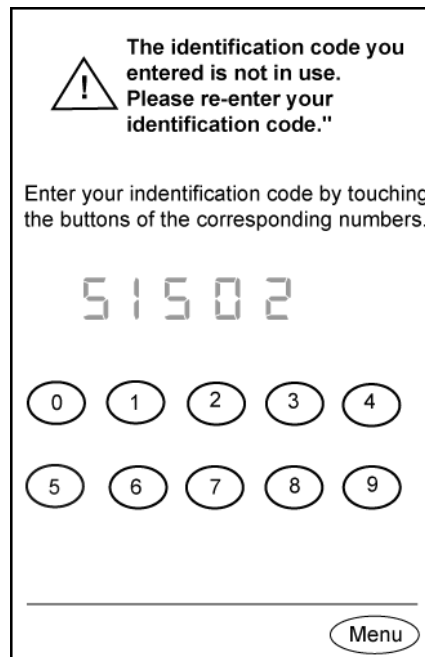


Figure 2.2 – Log-in screen with error message



Use Case 3: Calling up Deliveries List

Description: The device has a function that lists pending deliveries in the order they are to be made.

Actor: Bicycle delivery driver.

Precondition: The user is viewing the log-in or other screen.

Post-condition: The user is viewing the list of deliveries.

Primary Scenario: The user presses a soft key to call up the list.

Primary Task Flow:

1. The user presses the soft key labeled "List."
2. The list of current orders loads.

Alternate Scenario: The user presses the "List" key, but nothing happens.

Alternate Task Flow:

1. The user touches the "Menu" button.
2. A back-up menu screen loads (see Use Case 11).
3. User touches the "List" button with the stylus.
4. The list of current orders loads.

Technical Specs: **AJAX**

1. A script prioritizes deliveries in a next-job orientation as set by headquarters. The script detects new files received via wireless communication and reads a priority code that is part of each file's meta data. The script enters the new data in the list, reorders the list by delivery time and priority code, and loads the updated list.
2. A script detects the user's press on the screen and scrolls up or down based on the user's movement.
3. Completed deliveries are removed from the list and entered in the delivery record file (see Use Case 10).

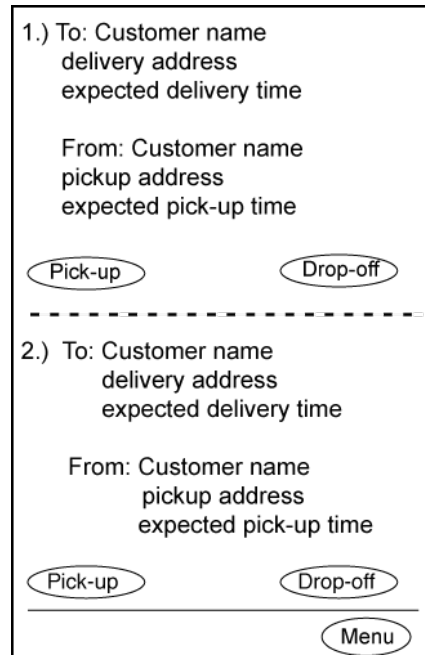
Display

1. The list scrolls vertically.
2. The list screen will contain a dividing line near the bottom and the "Menu" button in the bottom right corner.

List content

1. Name of customer/business sending delivery.
2. Pick-up address.
3. Expected time of pickup.
4. Name of customer/business receiving delivery.
5. Drop-off address.
6. Expected time of delivery.
7. A button labeled "Pick-up" at left and a button labeled "Drop-off" at right.
8. Dashed line.
9. First line of next entry, etc.

Figure 3.1 – Deliveries List



Use Case 4: View Map of Delivery Locations

Description: The device has a function that updates and displays a map of locations for pending deliveries, and the order in which they are to be made.

Actor: Bicycle delivery driver.

Precondition: The user is viewing the delivery list or some other screen.

Post-condition: The user is viewing a map.

Primary Scenario: The user presses a soft key to display the map.

Primary Task Flow:

1. The user presses the soft key labeled "DelivMap."
2. The map screen loads, showing the pick-up and drop-off locations for the first delivery on the list. User may, if desired, use finger to scroll around map.
3. User touches the "Next" button on touch-screen.
4. The map screen loads, showing the pick-up and drop-off locations for the second delivery on the list.
5. User touches "Next" up to three more times to display data about the first five deliveries on the list.
6. User presses "Clear," then "Next" to return to the next (or first) scheduled delivery.

Alternate Scenario: The user presses the "DelivMap" key, but nothing happens.

Alternate Task Flow:

1. The user touches the "Menu" button.
2. A back-up menu screen loads (see Use Case 11).
3. User touches the "DelivMap" button.
4. The map screen loads, showing the pick-up and drop-off locations for the first delivery on the list.
5. User picks up with No. 3 in the Primary Task Flow.

Technical Specs: AJAX

1. A script queries the database for the current list of deliveries based on times and priority codes, compiles it and displays a Google map graphically identifying the pick-up and drop-off locations of the next (or first) delivery.
2. A script detects the user touching the "Next" button on the screen and loads a map identifying the pick-up and drop-off locations for each subsequent delivery. The script also detects the user touching the "Clear" button on the screen and clears all delivery locations.
3. A script detects the user's press on the screen and scrolls based on the user's movement.

Display

1. Text at the top of the screen says "Location of next delivery," "Location of second delivery," etc., up to and including "fifth delivery."

Below that is the map, followed by a "Clear" button on the left and a "Next" button on the right, then a dividing line and a "Menu" button in the lower left corner.

2. The map scrolls in any direction.

Figure 4.1 Delivery map

Use Case 5: View Current Location

Description: The device has a function that displays a map showing the delivery driver's current location.

Actor: Bicycle delivery driver.

Precondition: The user is viewing a map of delivery locations or another screen.

Post-condition: The user is viewing a map showing his/her current location.

Primary Scenario: The user accesses the map with a soft key.

Primary Task Flow:

1. The user presses the soft key labeled "Location."
2. A Google map loads graphically identifying user's location as well as the pick-up and drop-off locations for the next delivery on the list.
3. User, if desired, touches "Next" button under map and the map reloads identifying the pick-up and drop-off locations for the first and second deliveries on the list.
4. User may use "Next" button to add data for up to five deliveries, marked numerically in order from the list.
5. User presses "Clear" button.
6. Map reloads showing user's current location and the pick-up and drop-off locations for the next delivery.

Alternate Scenario: The user presses the "Location" soft key but nothing happens.

Alternate Task Flow:

1. The user touches the "Menu" button.
2. A back-up menu screen loads (see Use Case 13).
3. User touches the "Location" button.
4. The map screen loads, showing user's current location and pick-up and drop-off locations for the next delivery.
5. User picks up with No. 3 in the Primary Task Flow.

Technical Specs: **AJAX**

1. A script regularly obtains data about the device's location from Global Position System technology housed with

- processor/circuit board, loads a Google map surrounding that location and displays an icon marking that location.
2. The script that queries and compiles the updated list and locations using GPS technology (Use Case 5, AJAX note No.1) adds icons identifying the pick-up and drop-off locations for the first delivery.
 3. A script detects the user touching the "Next" button on the screen and reloads a map identifying the user's current location, as well as the pick-up and drop-off locations for each subsequent delivery. The script accommodates data for the first five deliveries on the list.
 4. A script detects the user's press on the screen and scrolls based on the user's movement.

Display

1. Text at the top of the screen says "Current location."
Below that is the map, followed by a "Clear" button on the left and a "Next" button on the right, then a dividing line and a "Menu" button in the lower left corner.
2. The map scrolls in any direction.

Figure 5.1 – Current Location map



Use Case 6: Record Package Pick-up

Description: The device has functions to confirm that a package has been picked up.

Actor: Bicycle delivery driver.

Precondition: The user has accepted the package and is viewing the list of deliveries.

Post-condition: The user is viewing the updated list of deliveries.

Primary Scenario: A signature is required to confirm pick-up, but bar-code data is not.

Primary Task Flow:

1. The user calls up the updated Delivery List.
2. User touches "Pick-up" button.
3. Pick-up screen loads.
4. User touches "Yes" button ("Is package available?").
5. User touches "No" button ("Bar-code needed?").
6. Pick-up screen reloads with text field for signature.
7. Customer enters signature with stylus, clicks "Done" button next to text box.
8. Screen reloads with added "Confirm" button.
9. User (delivery person) touches "Confirm" button.
10. Delivery list reloads with text "Pick-up: hh:mm EST" replacing "Pick-up" button.

Alternate Scenario 1: Bar-code data is required.

Alternate Task Flow 1:

1. The user calls up the updated Delivery List.
2. User touches "Pick-up" button.
3. Pick-up screen loads.
4. User touches "Yes" button ("Is package available?").
5. User touches "Yes" button ("Bar-code needed?").
6. Pick-up screen reloads with message "Scanner activated."
7. User presses soft key labeled "Scan."
8. User scans bar code. Pick-up screen reloads with text field for signature.

9. Customer enters signature with stylus, clicks "Done" button next to text box.
10. Screen reloads with added "Confirm" button.
11. User (delivery person) touches "Confirm" button.
12. Delivery list reloads with text "Pick-up: hh:mm EST" replacing "Pick-up" button.

Alternate Scenario 2: Bar-code data is required, but nothing happens when user presses soft key for scanner.

- Alternate Task Flow 2:*
1. The user calls up the updated Delivery List.
 2. User touches "Pick-up" button.
 3. Pick-up screen loads.
 4. User touches "Yes" button ("Is package available?").
 5. User touches "Yes" button ("Bar-code needed?").
 6. User presses soft key labeled "Scan" but nothing happens.
 7. User touches "Menu" button.
 8. Users touches "Scan" button.
 9. User picks up with No. 8 in Alternative Task Flow 1.

Technical Specs:

AJAX

1. A script detects the user's touch on the "Pick-up" button and loads the Pick-up screen.
2. A script detects the user's touch on the "Yes" or "No" buttons associated with the question "Is the package available?" If the user chooses yes, the script records that choice. If the user chooses no, the script records that choice and reloads the Pick-up page with the list of reasons the package is unavailable (see Use Case 8).
3. A script detects the user's touch on the "Yes" or "No" buttons associated with the question "Bar-code needed?" If the user chooses no, the script records that choice. If the user chooses yes, the script records that choice, triggers activation of the scanner, receives confirmation the scanner is activated and reloads the Pick-up page with the message "Scanner activated."

4. A script detects the bar-code data sent by the scanner, records it and stores it in a secure database along with other data about the delivery. The script reloads the Pick-up page with the text box for the customer's signature displayed.

If the scanner doesn't successfully read the bar code, the script reloads the Pick-up page with an error message.

5. A script detects if the scanner has operated for 30 seconds without picking up a valid code, shuts off the scanner, and reloads the screen with an error message.
6. A script detects the customer's signing and touching the "Done" button with the stylus, and loads the updated Delivery List with "Pick-up hh:mm EST" displayed.
7. A script detects the user touching "Confirm," records the current time and date, and reloads the secondary confirmation page displaying the time of pick-up.

Display

1. The initial Pick-up screen contains the text title "Pick-up," followed by two questions, each with "Yes" and "No" buttons.
2. The Pick-Up screen that loads if the user selects "Yes" to "Bar-code needed?" is identical to No. 1 except for the added message "Scanner activated."
3. The "Yes" and "No" buttons will consist of styled oval buttons labeled "Y" and "N."
4. The Pick-Up screen that loads if the user selects "No" to "Bar-code needed?" is identical to No. 1 except for the added text box and "Done" button for the customer's signature.
5. The customer signature box will stretch nearly the entire width of the screen, with the "Done" button below it.
6. The "Done" button will consist of a stylized oval button labeled "Done."
7. A dividing line and the "Menu" button appear near the bottom of the screen.

Messages

1. "Scanner activated."
2. "No bar code data was detected. Press "Y" to try again or "N" to proceed without bar-code."

Figure 6.1 – Pick-up screen



Figure 6.2 – Pick-up screen with scanner message

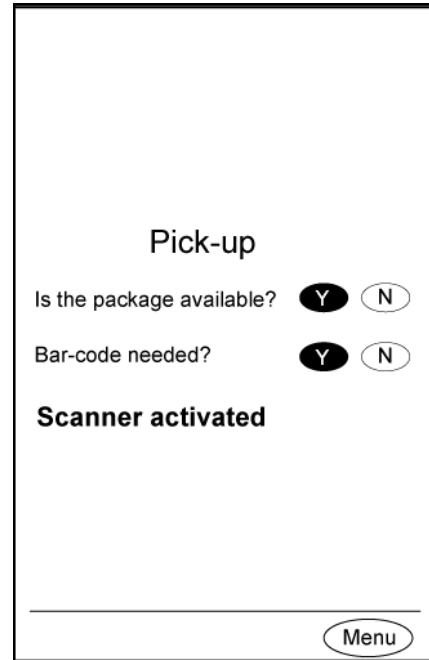


Figure 6.3 – Pick-up screen with error message about scanner

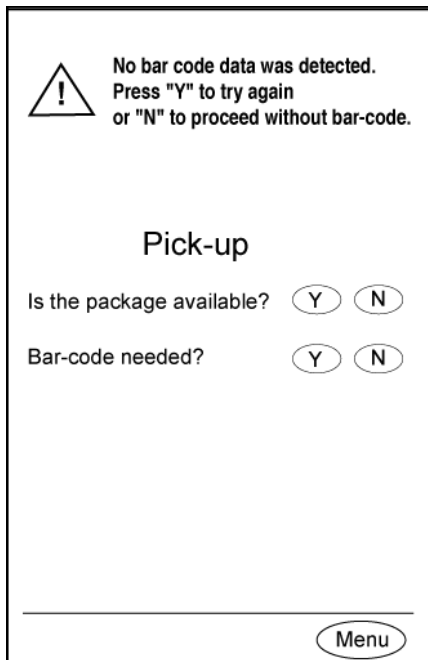
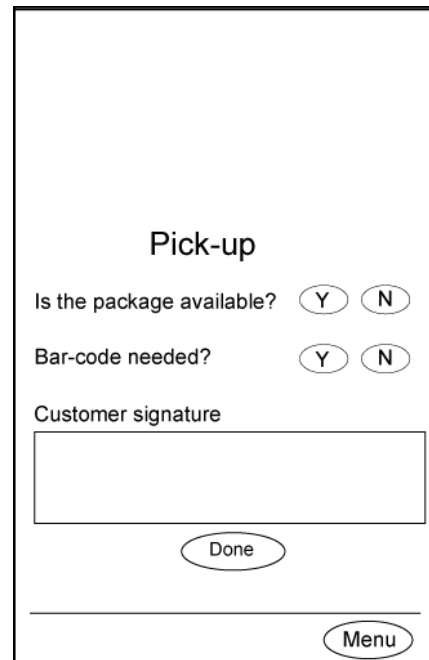


Figure 6.4 – Pick-up screen with customer signature box



Use Case 7: Record Package Drop-off

Description: The device has functions to confirm that a package has been dropped off.

Actor: Bicycle delivery driver.

Precondition: The user has handed over the package and is viewing the list of deliveries.

Post-condition: The user is viewing the updated list of deliveries.

Primary Scenario: A signature is required to confirm drop-off, but bar-code data is not.

Primary Task Flow:

1. The user calls up the updated Delivery List.
2. User touches "Drop-off" button.
3. Drop-off screen loads.
4. User touches "Yes" button ("Was drop-off successful?").
5. User touches "No" button ("Bar-code needed?").
6. Drop-off screen reloads with text field for signature and "Done" button.
7. Customer enters signature and touches "Done" button next to text box with stylus.
8. Drop-off screen reloads with added "Confirm" button.
9. User (delivery person) touches "Confirm" button.
10. Delivery list reloads with text "Drop-off: hh:mm EST" replacing "drop-off" button.

Alternate Scenario 1: Bar-code data is required.

Alternate Task Flow 1:

1. The user calls up the updated Delivery List.
2. User touches "Drop-off" button.
3. Drop-off screen loads.
4. User touches "Yes" button ("Was drop-off successful?").
5. User touches "Yes" button ("Bar-code needed?").
6. Drop-off screen reloads with message "Scanner activated."

7. User presses soft key labeled "Scan."
8. User scans bar code. Screen reloads with text field for stylus signature and "Done" button.
9. Customer enters signature and touches "Done" button with stylus.
10. Drop-off screen reloads with added "Confirm" button.
11. User touches "Confirm" button.
12. Delivery List loads with text "Drop-off: hh:mm EST" replacing "Drop-off" button.

Alternate Scenario 2: Bar-code data is required, but nothing happens when user presses soft key for scanner.

- Alternate Task Flow 2:*
1. The user calls up the updated Delivery List.
 2. User touches "Drop-off" button.
 3. Drop-off screen loads.
 4. User touches "Yes" button ("Was drop-off successful?").
 5. User touches "Yes" button ("Bar-code needed?").
 6. Drop-off screen reloads with "Scanner activated." message.
 7. User presses soft key labeled "Scan" but nothing happens.
 8. User touches "Menu" button on screen.
 9. Menu page loads.
 10. User touches "Scan" button.
 11. User picks up with No. 8 in Alternative Task Flow 1.

Technical Specs:

AJAX

1. A script detects the user's touch on the "Drop-off" button and loads the primary confirmation screen.
2. A script detects the user's touch on the "Yes" and "No" buttons associated with "Was drop-off successful?" If the user chooses yes the script records that choice and the time and date it was made. If the user chooses no, the script reloads the Drop-off page with the choices of reasons.

3. A script detects the user's touch on the "Yes" or "No" buttons associated with the question "Bar-code needed?" If the user chooses no, the script records that choice. If the user chooses yes, the script records that choice, triggers activation of the bar-code scanner, receives confirmation the scanner is activated and reloads the Drop-off page with the message "Scanner activated."
4. A script detects the bar-code data from the scanner, records it and stores it in a secure database along with other data about the delivery. The script reloads the Drop-off page with the text box for the customer's signature and a "Done" button displayed.
If the scanner doesn't successfully read the bar code, the script reloads the Pick-up page with an error message.
5. A script detects if the scanner has operated for 30 seconds without picking up a valid code, shuts off the scanner, and reloads the screen with an error message.
6. A script detects the customer's signing and touching the "Done" button with the stylus, and reloads the Drop-off page with a "Confirm" button displayed.
7. A script detects the user's touch on the "Confirm" button, records the time and date, and loads the updated Delivery List.

Display

1. The customer signature box will stretch nearly the entire width of the screen, with the "Done" button below it.

Messages

1. "Scanner activated."
2. "No bar code data was detected. Press "Y" to try again or "N" to proceed without bar-code."

Figure 7.1 – Drop-off screen

Drop-off

Was drop-off successful? Y N

Bar-code needed? Y N

Menu

Figure 7.2 – Drop-off screen with scanner message

Drop-off

Was drop-off successful? Y N

Bar-code needed? Y N

Scanner activated

Menu

Figure 7.3 – Drop-off screen with customer signature box

Drop-off

Was drop-off successful? Y N

Bar-code needed? Y N

Customer signature

Done

Menu

Use Case 8: Record Failed Pick-up Attempt

Description: The device has functions that let the user record a pick-up attempt in which the sender was not at the pick-up address at the pick-up time, or which failed for some other reason.

Actor: Bicycle delivery driver.

Precondition: The user is at the pick-up address at the scheduled time, but the recipient is unable to pick up the package.

Post-condition: The user is viewing the updated list of deliveries.

Primary Scenario: The user records the pick-up attempt as failed and gives a reason for the failure.

Primary Task Flow:

1. The user presses the "List" soft key.
2. The Delivery List screen loads, with the currently attempted delivery first.
3. The user presses the "Pick-up" button for the delivery.
4. The Pick-up screen loads.
5. User clicks "No" button ("Is package available?").
6. Pick-up page reloads with four choices.
7. User presses a button for one of the choices.
8. Delivery list loads with data about failed pickup.

Technical Specs: **AJAX**

1. A script detects the user's touch on the "Pick-up" button and loads the Pick-up page.
2. A script detects the user's touch on the "Yes" and "No" buttons associated with "Is package available?" If the user chooses no, the script reloads the Pick-up page with the choices of reasons displayed.
3. A script detects the user's choice of a reason, records that choice and loads the updated Delivery List with data about the failed pick-up time and date.
4. A script will record subsequent failed attempts and adds the data about time and date of other failed attempts to the updated Delivery List as described in No. 3 above.

Display

1. The "Yes" and "No" buttons consist of styled oval buttons labeled "Y" and "N."
2. Radio buttons are used to select a reason for the failed pick-up attempt.

Figure 8.1 – Pick-up screen with "No" answer

Pick-up

Is the package available? Y N

Bar-code needed? Y N

Menu

Figure 8.2 – Pick-up screen with choice of reasons

Pick-up

Why is the package unavailable?

No-one at address

Package not ready

Customer changed mind

Other

Menu

Figure 8.3 – Deliveries List with data on failed pick-up

1.) To: Customer name
delivery address
expected delivery time

From: Customer name
pickup address
expected pick-up time

Failed pick-up
hh:mm EST Drop-off

2.) To: Customer name
delivery address
expected delivery time

From: Customer name
pickup address
expected pick-up time

Pick-up Drop-off

Menu

Figure 8.4 – Deliveries List with data on several failed pick-ups

1.) To: Customer name
delivery address
expected delivery time

From: Customer name
pickup address
expected time of pick-up

Failed pick-up
hh:mm EST Drop-off

Failed pick-up
hh:mm EST

Failed pick-up
hh:mm EST

2.) To: Customer name
delivery address
expected delivery time

Menu

Use Case 9: Record Failed Drop-off Attempt

Description: The device has functions that let the user record a drop-off attempt in which the recipient was not at the drop-off address at the drop-off time, or which failed for some other reason.

Actor: Bicycle delivery driver.

Precondition: The user is at the drop-off address at the scheduled time, is unable to drop off the package.

Post-condition: The user is viewing the updated list of deliveries.

Primary Scenario: The user records the drop-off attempt as failed and gives a reason for the failure.

Primary Task Flow:

1. The user presses the "List" soft key.
2. The Delivery List screen loads, with the currently attempted delivery first.
3. The user presses the "Drop-off" button for the delivery.
4. The Drop-off screen loads.
5. User clicks "No" button ("Was drop-off successful?").
6. Pick-up page reloads with four choices.
7. User presses a button for one of the choices.
8. Delivery list loads with data about failed pickup.

Technical Specs:

AJAX

1. A script detects the user's touch on the "Drop-off" button and loads the Drop-off page.
2. A script detects the user's touch on the "Yes" and "No" buttons associated with "Is package available?" If the user chooses no, the script reloads the Drop-off page with the choices of reasons displayed.
3. A script detects the user's choice of a reason, records that choice and loads the updated Delivery List with data about the failed drop-off time and date.

4. A script records subsequent failed attempts and adds the data about time and date of other failed attempts to the updated Delivery List as described in No. 3 above.

Display

1. The "Yes" and "No" buttons consist of styled oval buttons labeled "Y" and "N."
2. Radio buttons are used to select a reason for the failed pick-up attempt.

Figure 9.1 – Drop-off screen with "No" answer

Drop-off

Was drop-off successful? Y N

Bar-code needed? Y N

Menu

Figure 9.2 – Drop-off with choice of reasons

Drop-off

Why is the package unavailable?

No-one at address

Package not ready

Customer changed mind

Other

Menu

**Figure 9.3 – Drop-off screen
with list of failed drop-offs**

1.) To: Customer name delivery address expected delivery time	
From: Customer name pickup address expected time of pick-up	
Pick-up hh:mm EST	Failed drop-off hh:mm EST
	Failed drop-off hh:mm EST
	Failed drop-off hh:mm EST

2.) To: Customer name delivery address expected delivery time	
<hr/>	
Menu	

Use Case 10: View Delivery Record

Description: The device has a function that compiles and displays a record of completed deliveries, as well as failed pick-up and drop-off attempts.

Actor: Bicycle delivery driver.

Precondition: The user is viewing any screen.

Post-condition: The user is viewing the Delivery Record screen.

Primary Scenario: The user wants to check the status of various deliveries.

Primary Task Flow:

1. The user presses the soft key labeled "DelivRecord."
2. The Delivery Record screen loads.

Alternate Scenario: The user presses the "DelivRecord" soft key but nothing happens.

Alternate Task Flow:

1. The user touches the "Menu" button.
2. The back-up menu screen loads (see Use Case 13).
3. User touches the "DelivRecord" button.
4. The Delivery Record screen loads.

Technical Specs:

AJAX

1. A script detects the user's pressing the soft key and loads the Delivery Record screen.
2. A script detects the user's touching the "DelivRecord" button in back-up menu and loads the Delivery Record screen.

Display

1. The Delivery Record screen is similar to the Deliveries List, contains all deliveries attempted during the day, and can be scrolled.

Use Case 11: View Alerts from Headquarters

Description: The device can receive alerts and other messages from headquarters, and display those messages if the user chooses to view them.

Actor: Bicycle delivery driver.

Precondition: The user is viewing any screen.

Post-condition: The user is viewing an alert message.

Primary Scenario: Staff at headquarters want to ask the user if he/she can quickly add a new delivery and rearrange the Delivery List.

Primary Task Flow:

1. The user hears a beeping tone which signals an alert, or views an alert message that appears at the top of whatever screen is being viewed.
2. User presses soft key labeled "View Alert."
3. Alert message screen loads.
4. User reads the message.
5. If the message does not require a reply, the user touches the "Close" button. If the message requires a reply, see Use Case 12.
6. The updated Deliveries List loads.

Alternate Scenario: The user presses the "View Alert" soft key but nothing happens.

Alternate Task Flow:

1. The user touches the "Menu" button.
2. The back-up menu screen loads (see Use Case 13).
3. User touches the "View Alert" button.
4. Alert message screen loads.
5. User picks up with No. 4 in Primary Task Flow.

Technical Specs: **AJAX**

1. A script detects the arrival of an alert message via

wireless communication and triggers a beeping tone and an alert message.

2. A script detects the user's pressing the "View Alert" soft key and loads alert message screen.
3. A script detects the user's touching the "View Alert" button in back-up menu and loads the alert message screen.

Display

1. The alert message will be displayed above the title of whatever page the user is viewing.
2. One of two screens will be displayed after the user presses "View Alert," depending on whether the alert message requires a reply or not.
3. The screen for messages not requiring replies will contain the text of the message, a "Close" button, and a dividing line near the bottom of the page followed by the "Menu" button in the bottom right corner.
4. The screen for messages requiring replies will contain the text of the message and a dividing line near the bottom of the page followed by the "Menu" button in the bottom right corner.

Messages

1. "Can you divert to (address) immediately to pick up a Priority 1 package?"
2. "Can you divert to (address) at hh:mm to pick up a Priority 2 package?"
3. "Can you divert to (address) sometime today to pick up a Priority 3 package?"
4. "Pick-up of Delivery No. X has been cancelled. Update Delivery List."
5. "Drop-off of Delivery No. X has been cancelled. Return package to sender at earliest convenience."
6. "Drop-off of Delivery No. X has been cancelled. Return package to sender immediately."
7. "Pick-up of Delivery No. X has been postponed. Update Delivery List."
8. "Drop-off of Delivery No. X has been postponed. Update Delivery List."

Figure 11.1 – Alert message

ALERT:
Can you divert to (address)
immediately to pick up a
Priority 1 package? Close

1.) To: Customer name
delivery address
expected delivery time

From: Customer name
pickup address
expected pick-up time

Pick-up Drop-off

2.) To: Customer name
address

Menu

Use Case 12: Accept/Decline Requests from Headquarters

Description: The device has a function that lets the user accept or decline an alert/request from headquarters, and specify a limited number of reasons for declining.

Actor: Bicycle delivery driver.

Precondition: The user is viewing an alert message.

Post-condition: The user is viewing the updated Deliveries List.

Primary Scenario: The user accepts the request.

Primary Task Flow:

1. The user reads the alert message.
2. User presses the soft key labeled "Accept."
3. Updated Deliveries List loads.

Alternate Scenario 1: The user declines the request.

Alternate Task Flow 1:

1. The user reads the alert message.
2. User presses the soft key labeled "Decline."
3. Updated Deliveries List loads.

Alternate Scenario 2: The user presses the "Accept" soft key, but nothing happens.

Alternate Task Flow 2:

1. The user reads the alert message.
2. User presses the "Accept" soft key but nothing happens.
3. User touches "Menu" button on screen.
4. Back-up menu loads.
5. User touches "Accept" button.
6. Updated Deliveries List loads.

Alternate Scenario 3: The user presses the "Decline" soft key, but nothing happens.

- Alternate Task Flow 3:*
1. The user reads the alert message.
 2. User presses the "Decline" soft key but nothing happens.
 3. User touches "Menu" button on screen.
 4. Back-up menu loads.
 5. User touches "Decline" button.
 6. Updated Deliveries List loads.

Technical Specs:

AJAX

1. A script detects the user's pressing the "Accept" soft key and loads the updated Deliveries List.
2. A script detects the user's touching the "Accept" button in back-up menu and loads the updated Deliveries List.
3. A script detects the user's pressing the "Decline" soft key and loads the updated Deliveries List.
4. A script detects the user's touching the "Decline" button in back-up menu and loads the updated Deliveries List.

Use Case 13: Back-up Menu

Description: The device has an alternate menu offering touch-screen access to all the functions of the soft keys, in case any or all of the soft keys become damaged or otherwise fail to function.

Actor: Bicycle delivery driver.

Precondition: The user is viewing any page.

Post-condition: The user is viewing a screen of buttons corresponding to the soft keys.

Primary Scenario: One or more soft keys do not work.

Primary Task Flow:

1. The user touches the "Menu" button in the bottom right corner of any screen.
2. The back-up menu screen loads.
3. The user touches the button for the desired function.
4. The appropriate screen loads.

Technical Specs:

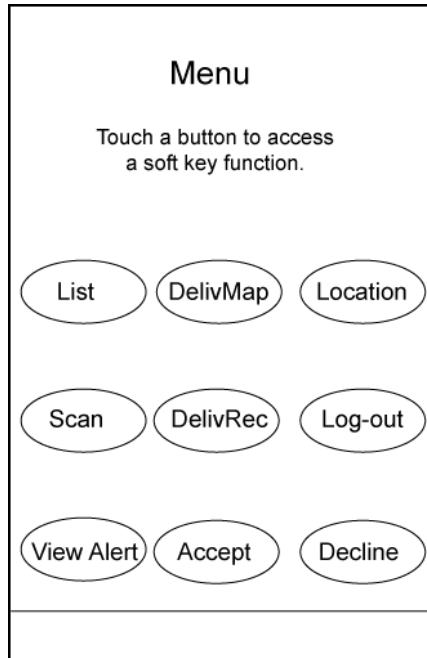
AJAX

1. Scripts will detect the user touching one of the various buttons and load the appropriate page or perform the appropriate function.

Display

1. The buttons and labels will appear in the same matrix-like sequence in which the soft keys are arranged.

Figure 13.1 – Back-up menu



Use Case 14: Log-out

Description: The device has a function that allows the user to log out by pressing a soft key.

Actor: Bicycle delivery driver.

Precondition: The user is viewing any screen.

Post-condition: The user is viewing the log-in screen and about to turn off the power or hand it off to another user.

Primary Scenario: The user logs out and turns off the power.

Primary Task Flow:

1. The user presses the soft key labeled "Log out."
2. Log-out screen loads.
3. User touches "Yes" button.
4. Screen loads asking if user wants to turn off device.
5. User touches "Yes" button.
6. Device shuts off power.

Alternate Scenario 1: The user logs out and but does not turn off power.

Alternate Task Flow 1:

1. The user presses the soft key labeled "Log out."
2. Log-out screen loads.
3. User touches "Yes" button.
4. Screen loads asking if user wants to turn off device.
5. User touches "No" button.
6. Log-in screen loads.
7. User logs in later or a new user logs in immediately.

Alternate Scenario 2: Nothing happens when the "Log out" soft key is pressed.

Alternate Task Flow 2:

1. The user presses the soft key labeled "Log out," but nothing happens
2. User touches "Menu" button on screen.
3. Log-out screen loads.
4. User picks up with No. 3 in Primary Task Flow.

Technical Specs:

AJAX

1. A script detects the user's pressing the "Log out" soft key and loads the log-out screen.
2. A script detects the user's touching the "Log out" button in the back-up menu and loads the log-out page.
3. A script detects the user's touching the "Yes" button in the log-out screen, ends the user's session and loads the log-in page.
4. A script detects the user's touching the "No" button in the log-out screen, and reloads the previous page.
5. A script detects the user's pressing the power switch and loads the screen asking if the user is sure he/she wants to turn off the device.

The script detects the user touching the "Yes" button and triggers the power shutting down.

The script detects the user touching the "No" button and reloads the log-in page.

Messages

1. "Do you wish to log out?"
2. "Do you wish to turn off the device?"

Figure 14.1 – Log-out screen

