

# SMS specifications for sprint 1

## HTML 5 project

V4.0

Author: Ayman Maat

Date: 06.07.2012



# Release note

Wireframe pack lives in:

TF Dropbox : OWD\_UX\_Working/THE NEW WORLD/02\_Sprints/3\_Contacts\_Messages/1\_Interaction/wireframes/05 SMS  
Mozilla Dropbox : OWD\_Moz\_share/Mozilla/App SMS  
Jira : Open Web Device / OWD-1380  
Mozilla's wiki : <https://wiki.mozilla.org/Gaia/SMS>

## new wireframes

- none

## updated wireframes

### **SMS : message thread listing**

- annotation 04 updated to include failed messages

## deleted wireframes

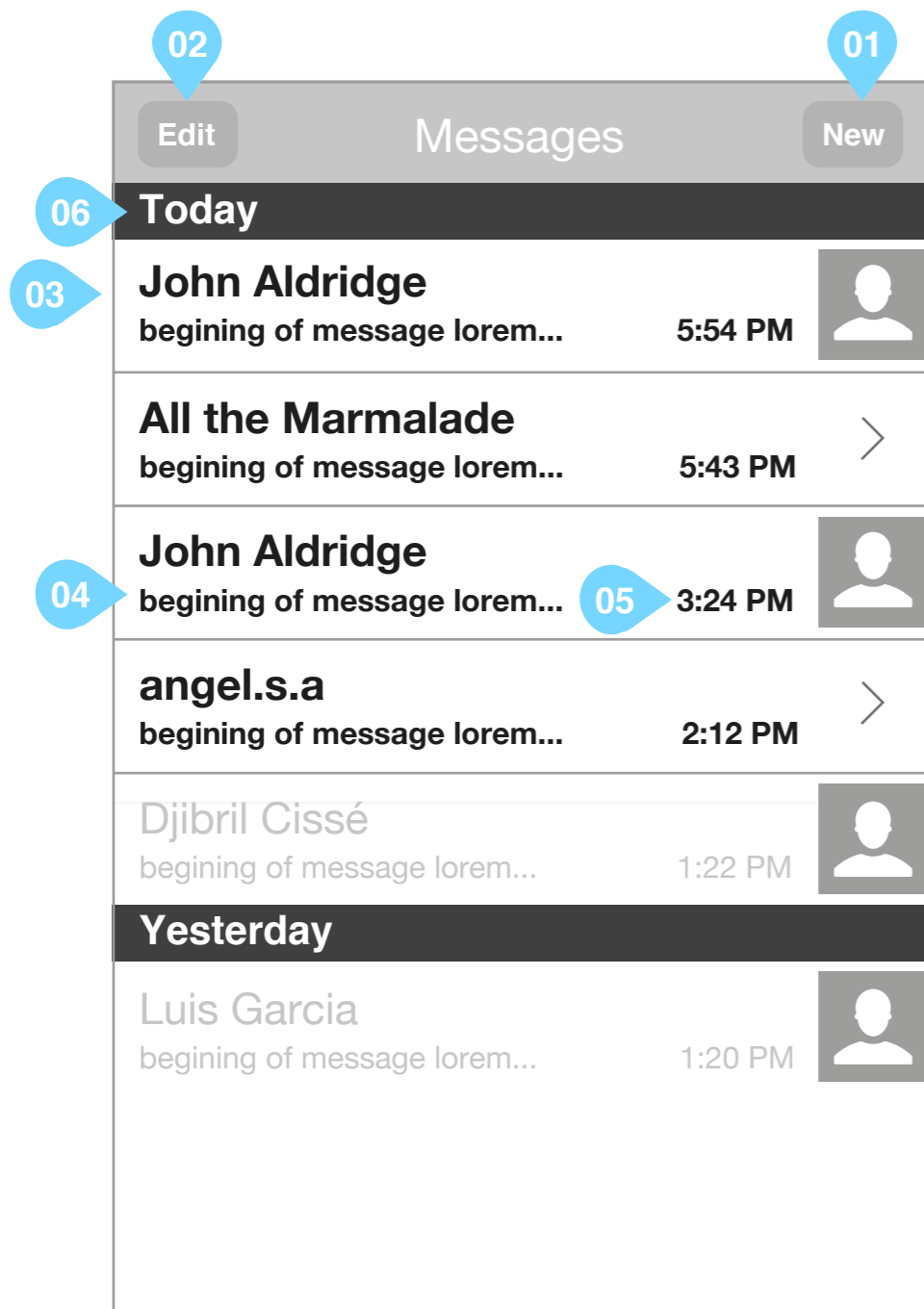
- none

# Table of Contents

SMS : message thread listing	5
SMS : message thread	6
SMS : existing message composer	7
SMS : new message composer	8
SMS : contact list	9
SMS : contact list phone numbers	10
SMS : message thread listing edit mode	11
SMS : message thread edit mode	12
SMS : message failure options	13
SMS : service unavailable	14

# Comms app (basic comms)

## SMS : message thread listing



Wireframe illustrating content that is contained under the SMS tab:

### annotation

01 Compose new message button

upon tap

opens black SMS composer. refer to wireframe 'SMS : new message composer'

02 Edit button

if there is not content in the list make the edit button incative

upon tap

changes list area to editable mode. refer to wireframe 'SMS : message thread listing edit mode'

03 message module

highlighted text indicates that the thread contains unread incoming message

upon tap

takes the use to the message thread. refer to wireframe 'SMS : message thread'

04 message preview

- truncated text of last message in thread

- if a message has failed to be sent replace meassage preview with 'Message failed to send'

05 timestamp

timestamp of last message in message thread

06 time labelling

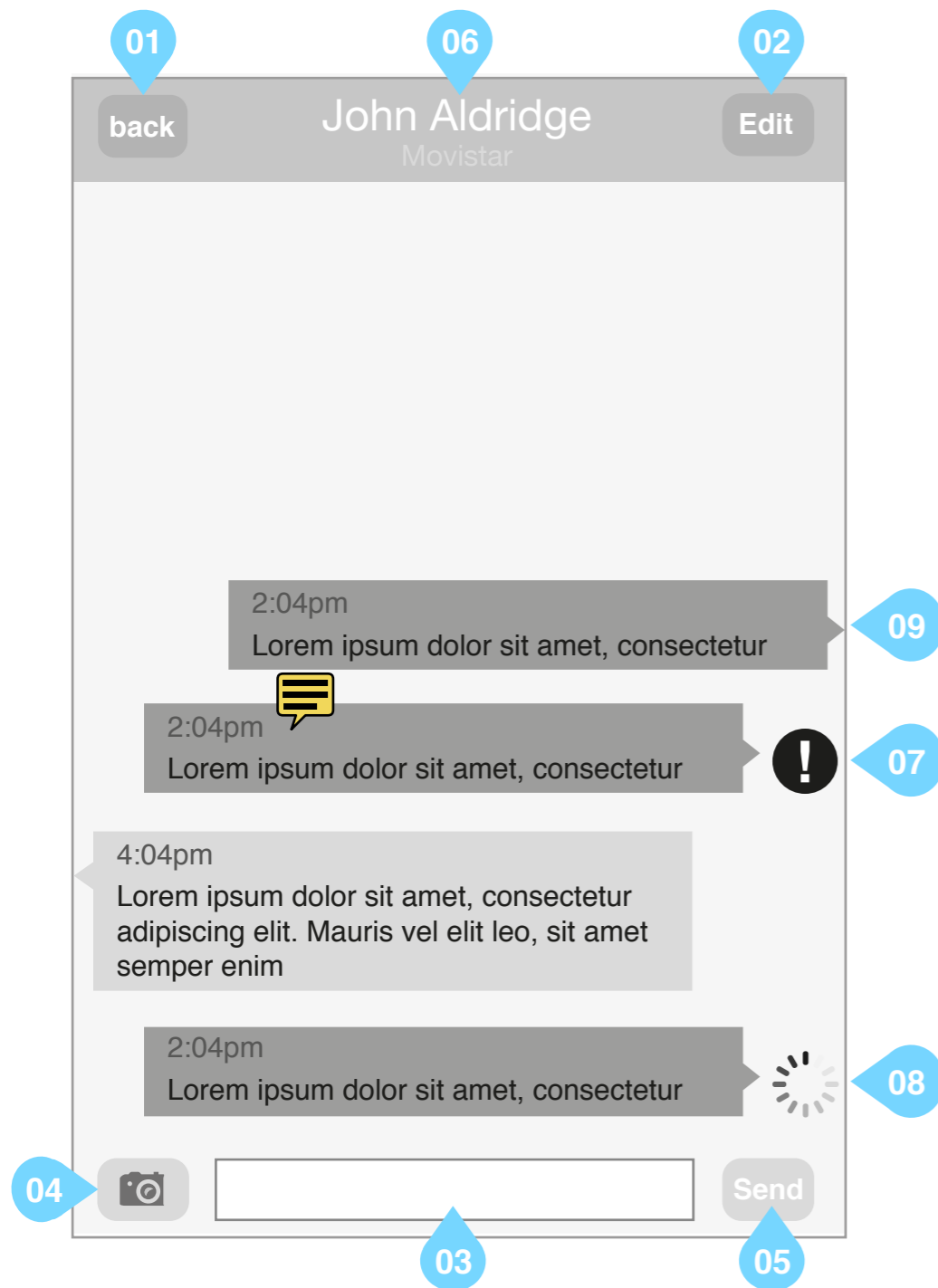
for time labelling specification refer to wireframe 'call log' in wireframe pack 'HTML5\_Dialer\_Contacts\_20120620\_R2S1'

### first time / empty state

When the list is empty output 'no messages recorded, why not start communicating now'

# Comms app (basic comms)

## SMS : message thread



Wireframe illustrating an SMS message thread:

### annotation

01 Back button

upon tap

takes the user back to the interface from which the thread was called

02 Edit button

upon tap

changes list area to editable mode. refer to wireframe 'SMS : edit mode'

03 textfield

upon tap

opens keyboard so that user can post to thread. refer to wireframe 'SMS : existing message composer'

04 add picture button

05 send button

06 contact name

upon tap

takes the user to the contacts profile. refer to wireframe : 'contact detail'

07 indication of failure in message sending

upon tap

takes the user to wireframe 'SMS : message failure options'

08 sending activity indicator

- upon successful sending of a message indicator vanishes and message right indents to enforce its sending

- if the message has not been sent because the service is unavailable the indicator remains until service becomes available and message is sent.

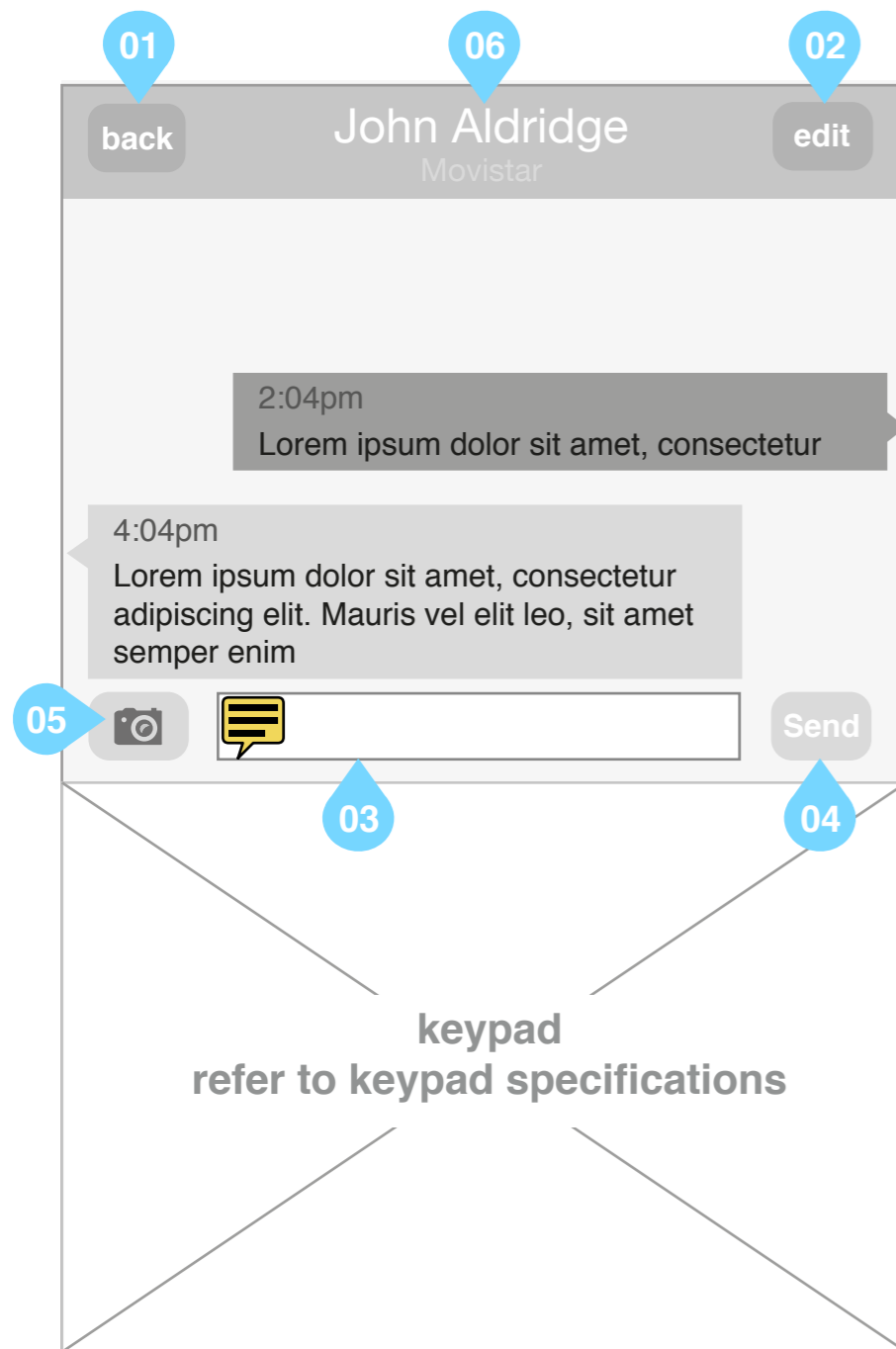
09 successfully sent message

### first time / empty state

When the list is empty output 'no messages recorded, why not start communicating now'

# Comms app (basic comms)

## SMS : existing message composer



Wireframe illustrating an SMS message thread:

### annotation

01 Back button

upon tap

takes the user back to the interface from which the thread was called

02 Edit button

upon tap

- keyboard is closed

- changes list area to editable mode. refer to wireframe 'SMS : edit mode'

03 textfield

plays back to user what they have written prior to posting on message thread

04 Send button

submits content of text field to message thread

input errors to communicate to end user upon selecting send:

- content typed into text field exceeds limit

- memory assigned for message storage is close to limit

- memory assigned for message storage exceeds limit

- service is unavailable. refer to wireframe : 'SMS : service unavailable'

05 add photo button - version 2 feature

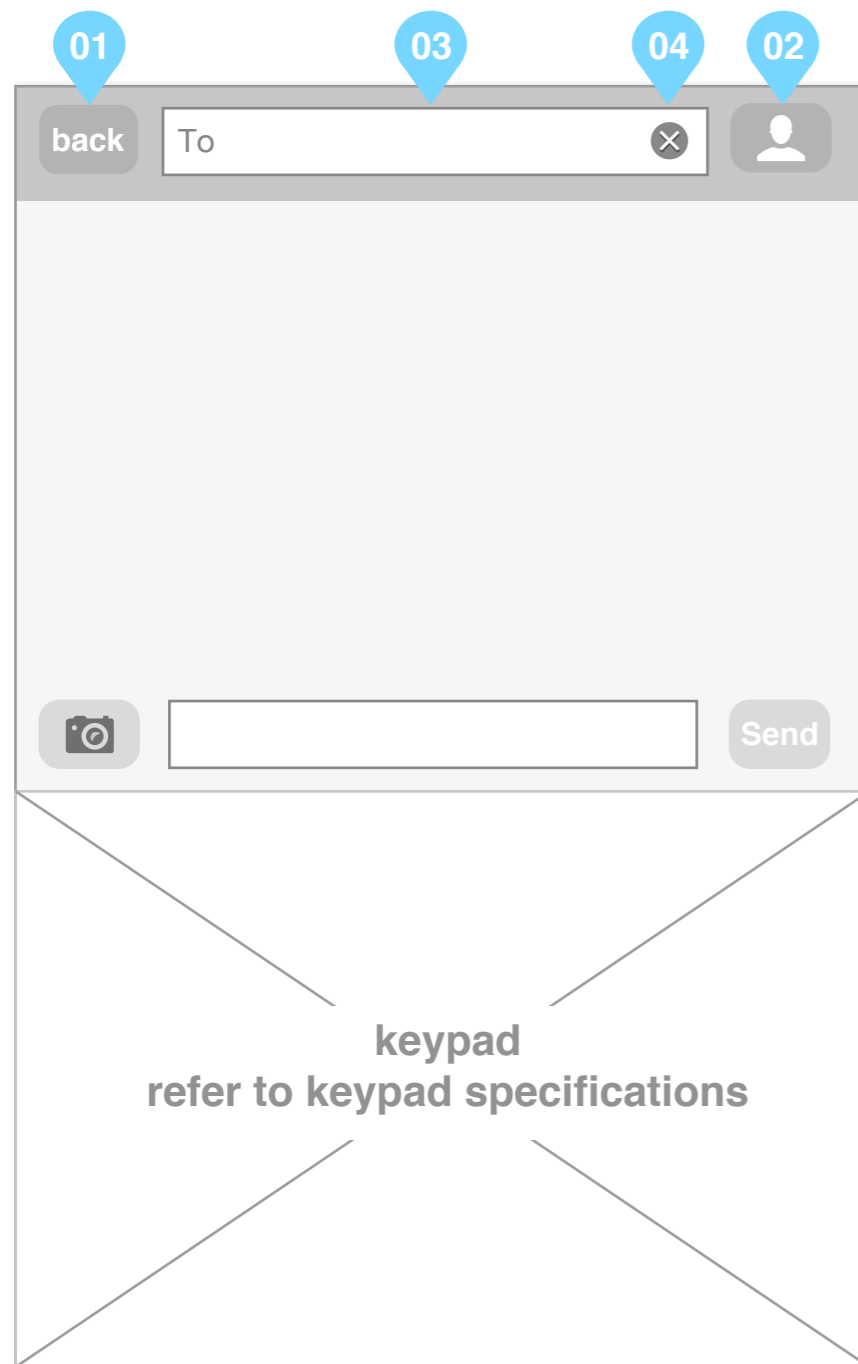
06 contact name

upon tap

takes the user to the contacts profile. refer to wireframe : 'profile : contact detail'

# Comms app (basic comms)

## SMS : new message composer



Wireframe illustrating an SMS message thread:

### annotation

#### 01 Back button

##### upon tap

takes the user back to the interface from which the thread was called

#### 02 add contact button

if content is already added to the textfield in annotation 03 make add contact button inactive

##### when active upon tap

- contact list is opened. refer to wireframe 'contact list : all'

#### 03 To Textfield

when a recipient is added to the SMS via annotation 02 write name in textfield. otherwise:

##### upon tap

- text box becomes active, keyboard remains alphabetical

##### upon type

- upon typing a string live search results of string to contact name matches show in messages pane.

##### input errors to communicate to end user upon typing:

- no search results returned.

#### 04 call to action to delete content of textfield

##### upon tap

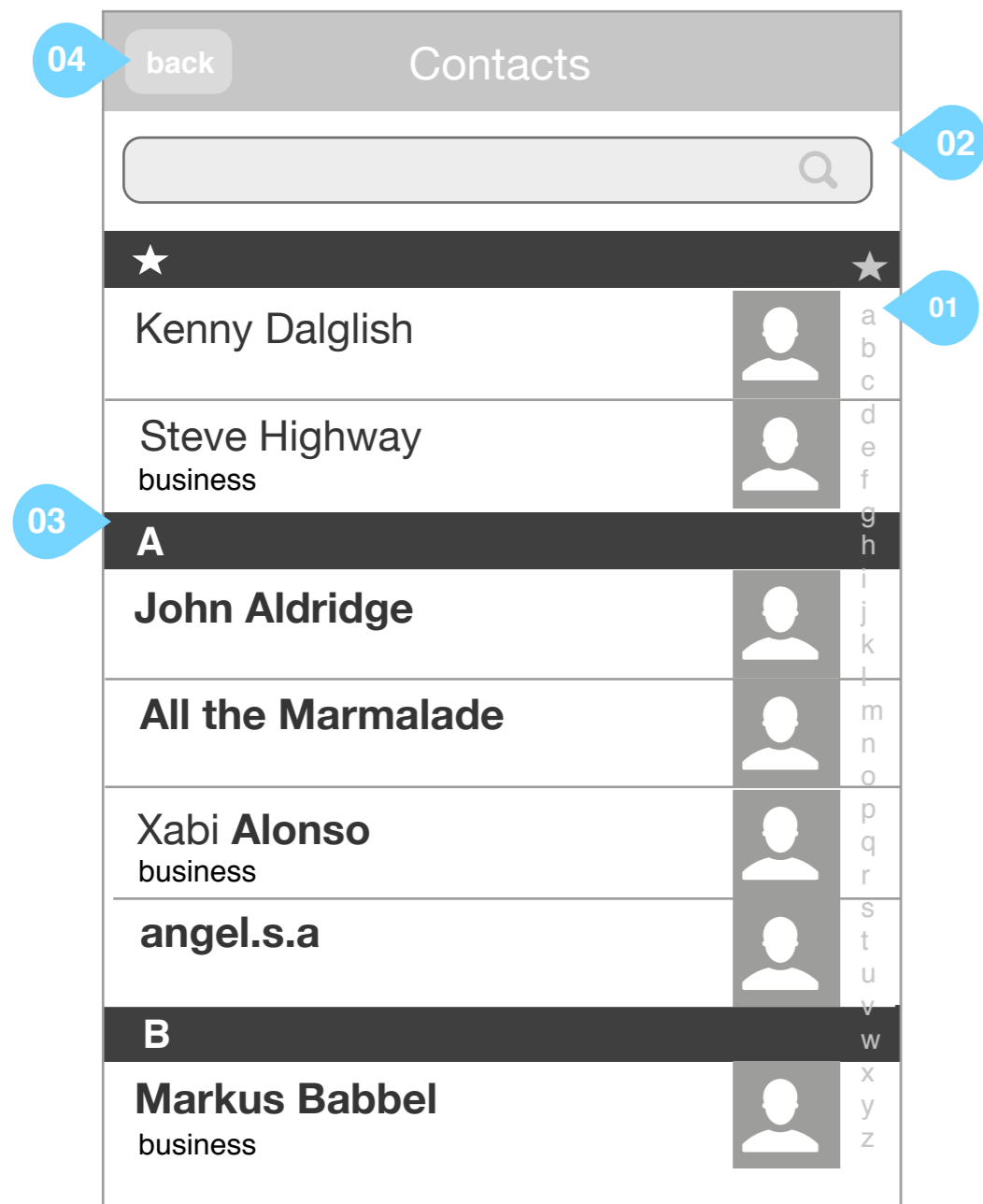
content of text field is cleared

For all other functional specifications refer to wireframe 'SMS : existing message composer'



# Comms app (basic comms)

## SMS : contact list



Wireframe illustrating the contact list when accessed through the SMS tab

### annotation

01 vertical index 

**upon tap**  
takes the user to the position in the contacts list starting with the selected letter

02 Opens search facility allowing the user to search for a contact within the contacts list

03 Contact module including content of 'business' field if used

**upon tap**  
adds the user to the SMS conversation

04 back button

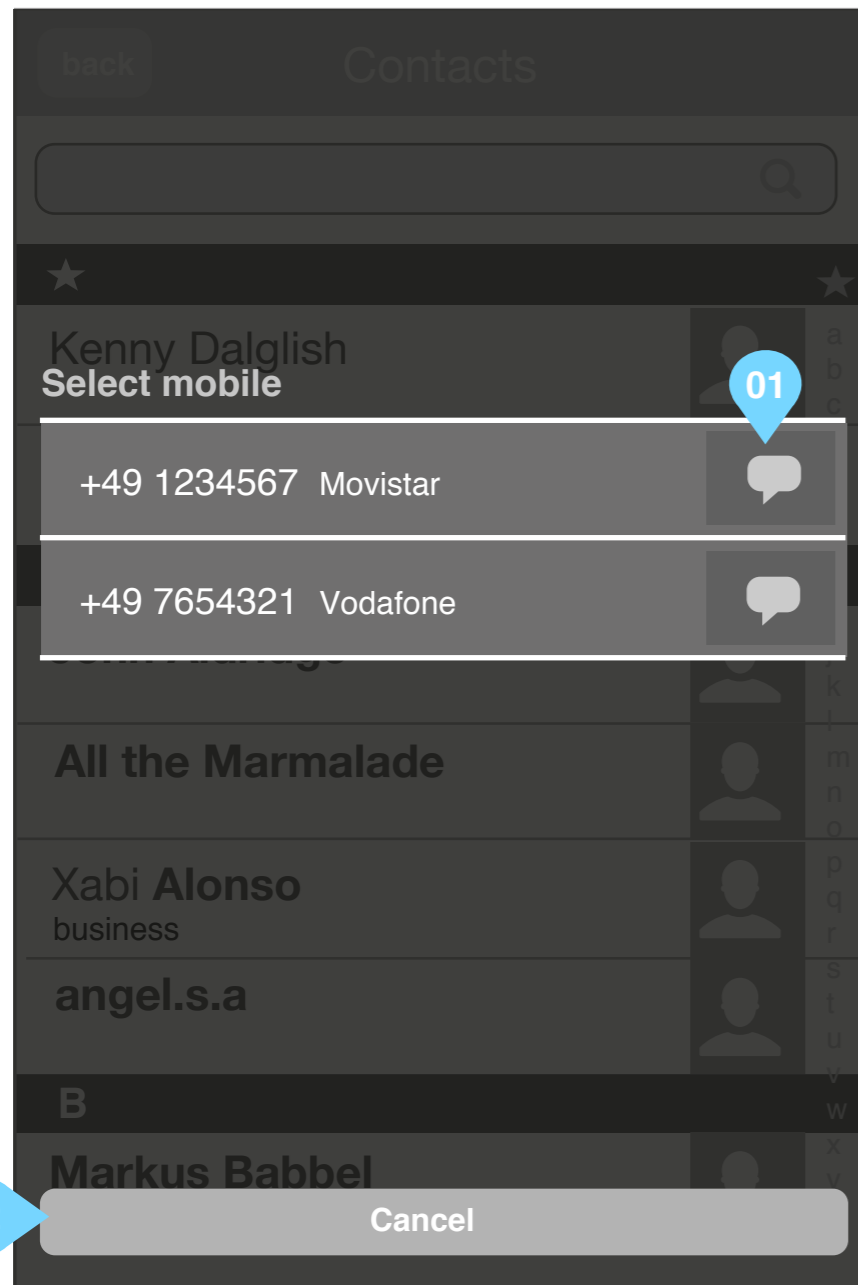
**upon tap**  
user is returned to wireframe 'SMS : new message composer'

### first time / empty state

- Suggest presenting in page invitation to import contacts or add new contacts.
- Make edit button inactive.

# Comms app (basic comms)

## SMS : contact list phone numbers



Wireframe illustrating reply options to a communications cluster in the timeline

### annotation

01 call to action to add contact to the SMS using the desired phone number

#### upon tap

if existing SMS conversation already exists:

- go to existing message thread. refer to wireframe 'SMS : message thread'

if there is no existing SMS conversation already existing:

- go to new message composer with the 'TO' field prepopulated with the contacts name. refer to wireframe 'SMS : new message composer'

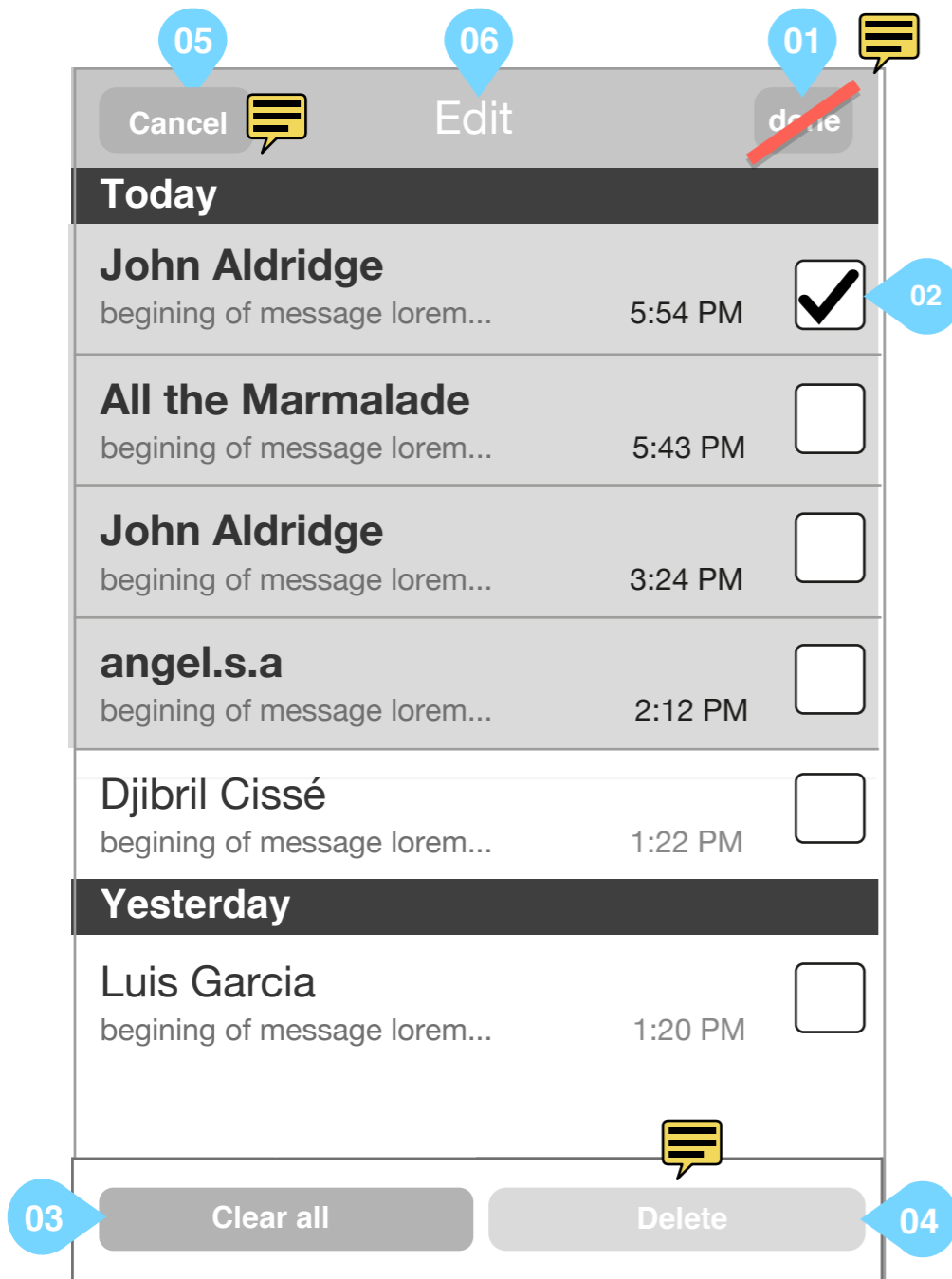
02 cancel

#### upon tap

closes overlay and returns the user back to their original view

# Comms app (basic comms)

## SMS : message thread listing edit mode



Wireframe illustrating the messages listing in edit mode. The timeline in edit mode allows the user to:

- remove selected messages
- clear all messages from all contacts

### annotation

01 done button

upon tap

- any edits to the timeline are committed
- editable mode is exited and the user is returned to the messages listing view
- label of button changes to 'edit'

02 checkbox to select a message to delete

upon tap

checkboxes become ticked

03 clear all messages button

upon tap

dialogue to confirm deletion of all messages from SMS is opened. refer to wireframe 'profile : timeline delete all communications confirmation'

04 delete selected messages button

button is inactive until the user selects a checkbox

upon tap

deletes all selected messages

05 cancel button

upon tap

- any edits to the message thread listing are discarded
  - editable mode is exited and the user is returned to the message thread listing view.
  - label of 'Done' button changes to 'new'
  - label of 'cancel' button changes to 'edit'
  - Header text is replaced by the word 'Messages'
- refer to wireframe 'SMS : message thread listing'

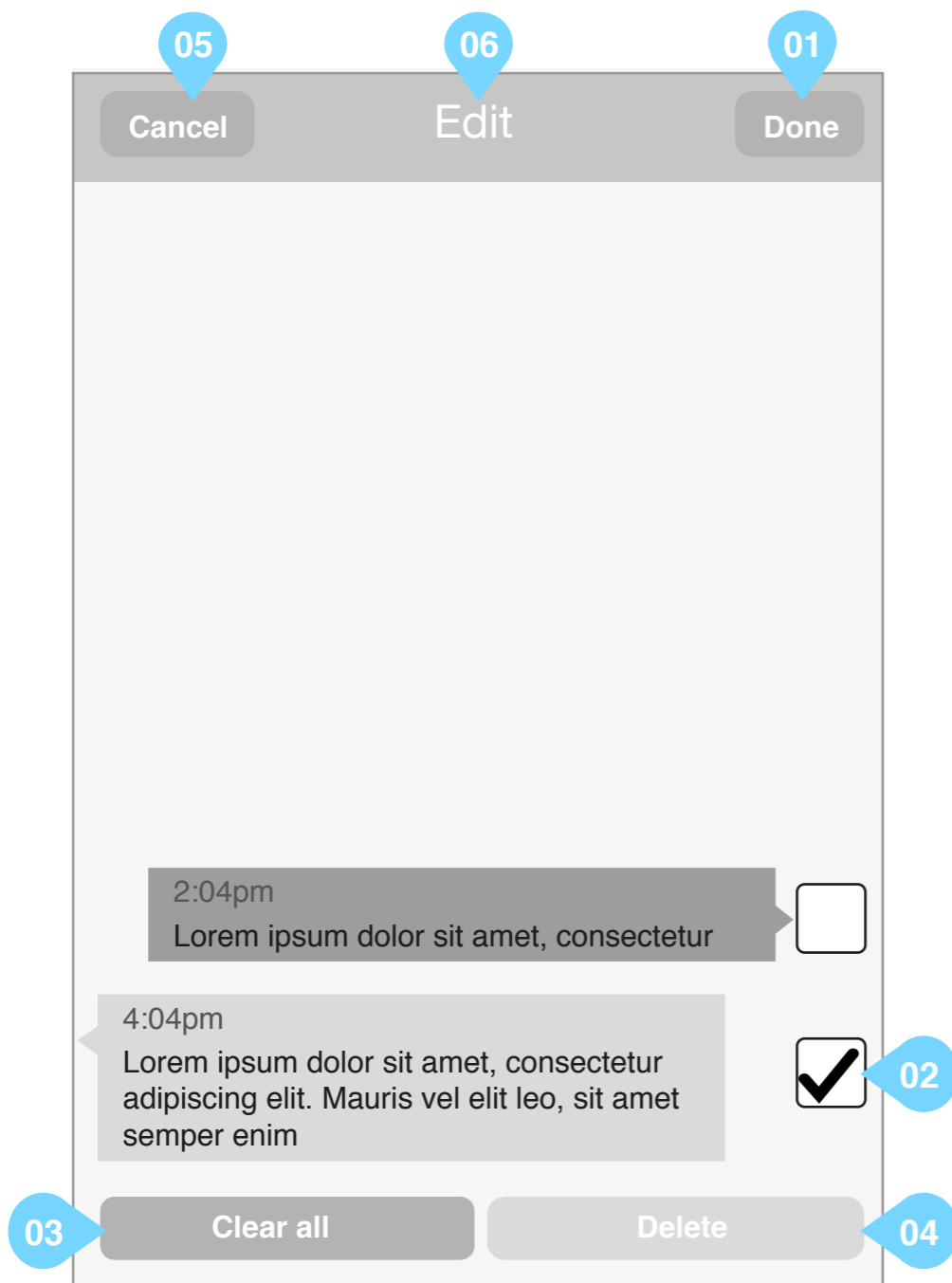
06 header text

upon selection of a checkbox

replace 'edit' with '<n> selected' where <n> is the number of checkboxes selected

# Comms app (basic comms)

## SMS : message thread edit mode




Wireframe illustrating the SMS thread in edit mode. The edit mode allows the user to:

- remove selected messages
- clear all messages in the thread

### annotation

#### 01 done button

##### upon tap

- any edits to the thread are committed
- editable mode is exited and:
  - if messages remain in the thread and the user has entered edit mode from wireframe 'SMS : message thread' user is taken to wireframe 'SMS : message thread'
  - if messages remain in the thread and the user has entered edit mode from wireframe 'SMS : existing message composer' user is taken to wireframe 'SMS : existing message composer'
  - if no messages remain in the thread user is taken to wireframe 'SMS : new message composer' 

#### 02 checkbox to select a message to delete

##### upon tap empty checkbox

checkboxes become ticked

##### upon tap ticked checkbox

checkboxes become empty

#### 03 clear all messages button

##### upon tap

dialogue to confirm deletion of all messages from selected contact is opened. refer to wireframe 'profile : timeline delete all communications confirmation'

#### 04 delete selected messages button

button is inactive until the user selects a checkbox

##### upon tap

deletes all selected messages

#### 05 cancel button

##### upon tap

- any edits to the message thread are discarded
- editable mode is exited and the user is returned to the message thread view.
- label of 'Done' button changes to 'edit'
- label of 'cancel' button changes to 'back'
- Header text is replaced by name and carrier of contact refer to wireframe 'SMS : message thread'

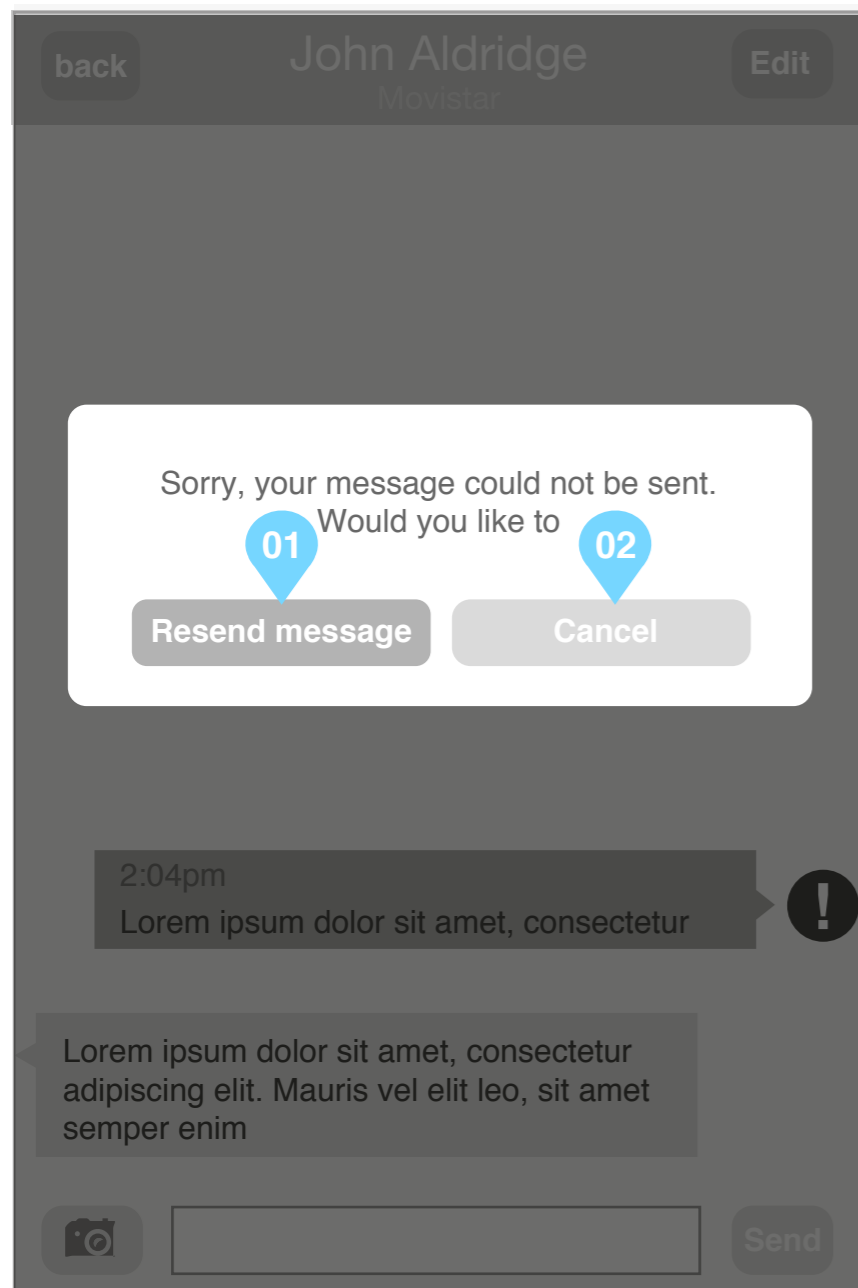
#### 06 header text

##### upon selection of a checkbox

replace 'edit' with '<n> selected' where <n> is the number of checkboxes selected

# Comms app (basic comms)

## SMS : message failure options



Wireframe illustrating the options when an SMS has failed to send:

### annotation

01 resends message

02 cancels the message

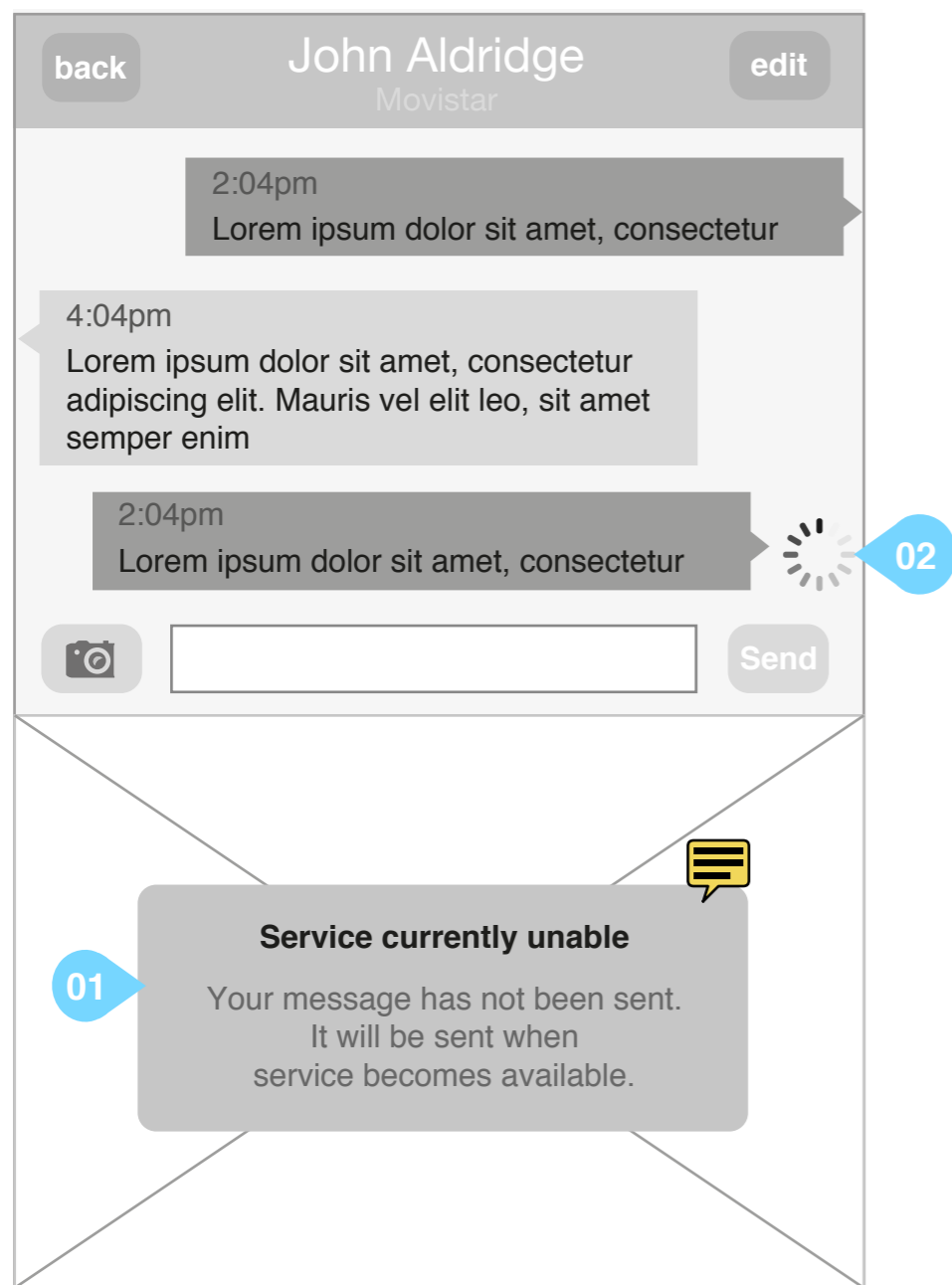
upon tap

changes list area to editable mode. refer to wireframe 'SMS : edit mode'



# Comms app (basic comms)

## SMS : service unavailable



Wireframe illustrating a sent SMS message when the service is unavailable

### annotation

- 01 temporary message overlays keyboard informing the user that the service is unavailable and that the message has not been sent
- 02 sending activity indicator remains present when service is unavailable

# Thank you

Ayman Maat  
aymanmaat@hotmail.com