In this assignment you will be implementing a simple WebApp with AngularJs based on mockup provided by designer.

The purpose for this exercise is for you to demonstrate your ability to understand the use-case, develop your solutions and think independently.

Please be sure to read these instructions thoroughly before you begin.

Instructions

- Knowledge of the JS/AngularJS development is assumed and necessary to complete the assignment.
- You are expected to work on your own machine. It is allowed to use any IDE or OS you in favor of.
- This is an open assignment - while the instructions give significant guidance, they are not a step-by-step tutorial. You may and should use any resource available to you including search engines, message boards, web sites or anything else. We want to see how well you can solve problems, not just use what you already know.
- Your code should be written as to meet quality in professional working settings. Commenting, decomposition and error handling are a few of the things you should consider while generating your solution.

Problem

- Please implement the given mockup page (mockup.zip) with following use cases. Other requirements are specified at bottom session.

Use Case

Background

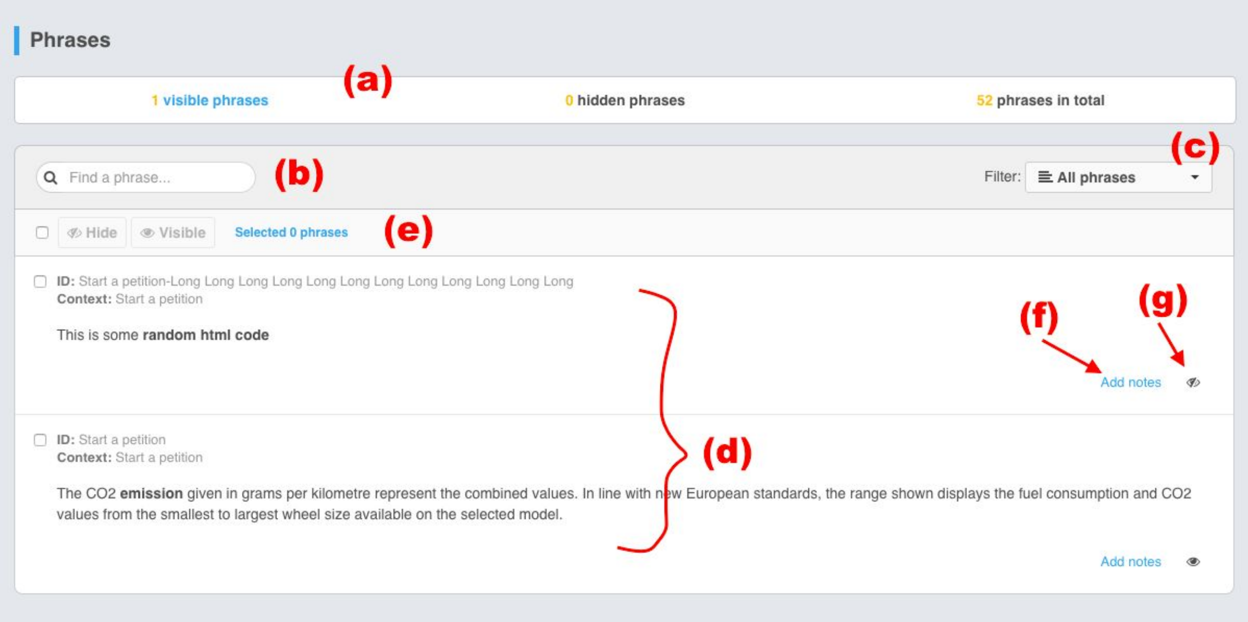
Clients provide phrases (strings) to our platform for translations. It’s a page to manage those phrases.

Details

1. Each phrase only has one single status, either as Visible(Default) or Hidden
2. User is able to change phrase status freely.
3. User is able to bulk update phrases status.
4. User is able to add notes to each phrase.
5. User is able to filter phrase by keyword

   a. Filter will apply to phrase id, context, value, and notes

6. User is able to filter phrase by status



1 visible phrases 0 hidden phrases 52 phrases in total

Selected 0 phrases

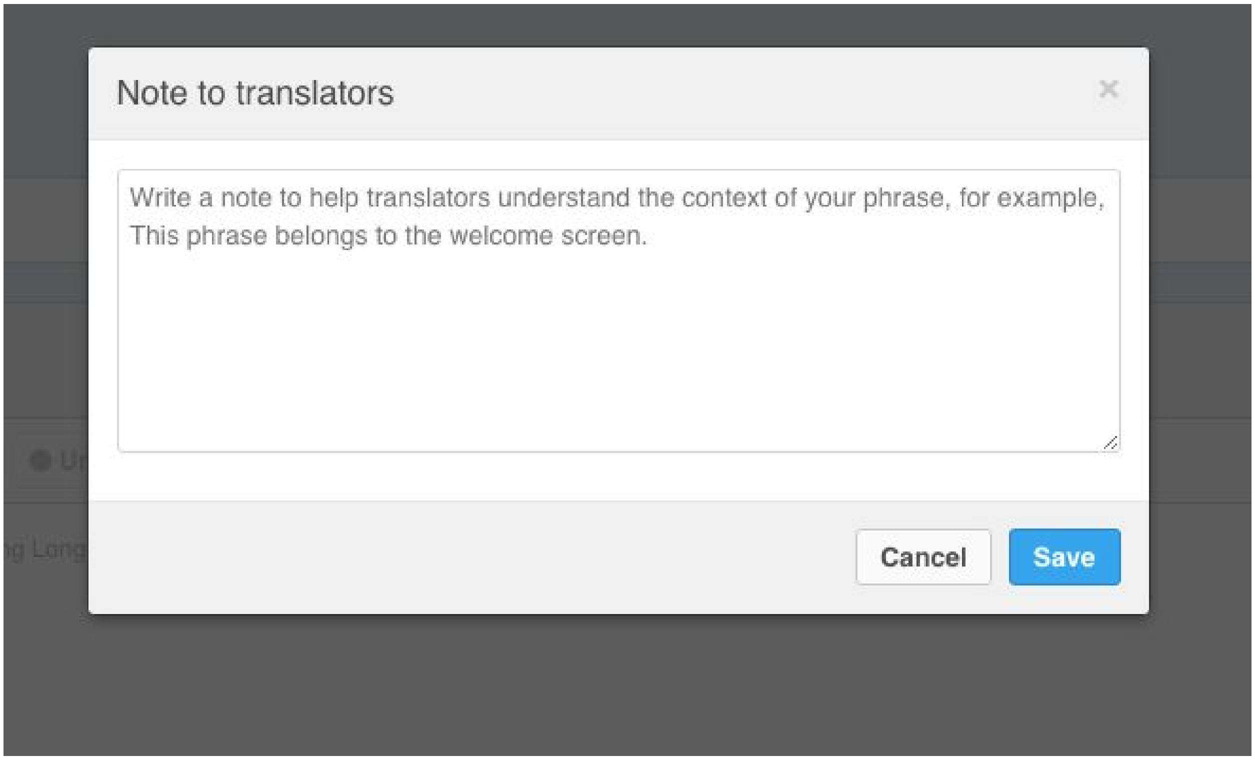
ID: Start a petition-Long Long Long Long Long Long Long Long Long Long Long Long
Context: Start a petition

This is some **random html code**

ID: Start a petition
Context: Start a petition

The CO2 **emission** given in grams per kilometre represent the combined values. In line with new European standards, the range shown displays the fuel consumption and CO2 values from the smallest to largest wheel size available on the selected model.

a. Phrases statistic bar
   i. Display phrases count for different status
   ii. Filter phrase by corresponding status after user click
b. Keyword search
   i. Order by phrase id
   ii. Search phrase which contained keyword
   iii. Field : id, context, value, notes
c. Status Filter
   i. Filter phrase by corresponding status
d. Phrase list
   i. Display phrase id, context and value
   ii. Render the list by JSON data source
   iii. Assume the list contained 1000 items
   iv. Only phrase value field allow html-styled text
e. Action bar
   i. select all checkbox to choose all phrases
   ii. 4 status button to change selected phrase status
      1. Disable the button when no phrase selected
   iii. Display selected phrases count
f. Add notes button
   i. Attach free text notes to phrase.
   ii. Model will popup and saved notes should be loaded if any



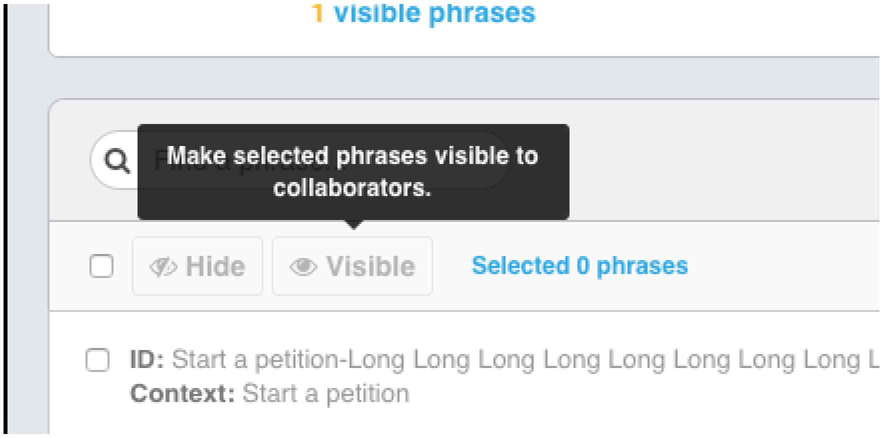
Note to translators

Write a note to help translators understand the context of your phrase, for example, This phrase belongs to the welcome screen.

   iii. Click Save button will save note and Cancel button won’t
   iv. Notes should be stored locally
   v. It can be assumed user will input Plain text only.

g. Status icon
   i. Display the phrase current status
   ii. Update instantly after status changed.

h. Tooltip



Make selected phrases visible to collaborators.

ID: Start a petition-Long Long Long Long Long Long Long Long L
Context: Start a petition

   i. Buttons and links has tooltip. Don’t miss them.

Requirements

  - Support Chrome v40 or above
  - Use AngularJs 1.5.x
  - No jQuery dependency / code, Angular’s jqLite is allowed
  - Allowed to add any libraries / frameworks.
  - Allowed to modify the HTML structure to fulfill your coding architecture
  - Don’t modify the provided css files
  - Not required to use Bootstrap
  - Keep the layout as original as possible
  - Assume the data list will serve for 1000 items and take performance into considerations
  - Store the user input locally (localstorage / webSQL / cookies …)
  - Hard-coded data are allowed, not necessary to handle the data-fetching via network.
- Using following items are a plus
  - ES6/7
    - Be sure your code are working on Chrome v40
    - JS compiler allowed
  - Webpack
  - Any dependency management tools
  - JS lint tools and style guide
- You are allowed to research for solutions and that would be taken into account.
- Please test your program thoroughly, in later stage of interview potentially you would be followed up with further problems using changed input or questions on your design.

Deliverables and Time Limit

- You will be expected to email us a zip file that contains source code that can compile and run your solution.
- Also provide a instruction to initialize your works if any. Like how to install the dependencies
- There is no strict time limit for this assignment except for the deadline specified. Though we will consider how long it took for us to receive your solution, you should feel free to take however long you feel is necessary to solve the problem to your satisfaction.

Submission

Please reply to the email attaching the zipped files.