Requirements Specifications

Al Jazeera Multimedia Translation Tool - Innovations Lab, Kosovo

Revision 1.0  
Author Soud Hyder  
Date 22nd Nov 2012

4

Table of contents

[1.0 Functional Requirements](#h.hi6rpau8z6th)

[1.1 Authentication](#h.yf3ix124adj)

[1.2 Text Translation](#h.nua9v5wz22uh)

[1.3 Audio Auditions](#h.svpm9vngp0jq)

[1.4 User Management](#h.83arunykmlyg)

[1.5 Social Features](#h.6wy834udk0on)

[2.0 Technical Requirements](#h.7ml674ghxs9o)

[2.1 REST API](#h.d37sj1nwhydx)

[2.2 Modularisation](#h.ea4ejylnslev)



# 1.0 Functional Requirements

## 1.1 Authentication

* Web interface authentication and user account administration
* Rights management to control data access to respective data sets\*
* Key based authentication for API calls\*

## 1.2 Text Translation

* Text input of scripts
* Uploading and reading of script documents in common formats e.g .txt, .doc and .docx
* Crowdsourced translation of scripts, scripts should be broken down into sizeable chunks (e.g paragraphs) and distributed
* Review of translated chunks by editors and compilation of final scripts
* Processing of text subtitle files of approved translated script in common formats e.g. SBV, SRT

## 1.3 Audio Auditions

* Approved scripts should be publicly viewable as a “challenge”
* Auditions per approved script to be recorded and displayed on the platform via Soundcloud
* Crowd will vote on best audition which will be reviewed by an editor
* Most voted and best quality to be selected by editor and remixed with the original video file to produce a translated version of the original video in the target language

## 1.4 User Management

* Interface to administer users and roles
* Roles
  1. Administrator (full access)
  2. Editor (access to assigned task)
  3. Translator (access to publicly available translation tasks)

## 1.5 Social Features

* Ability to translators to share updates or call to actions on Facebook and Twiiter
  1. ‘I have just helped to translate this video <http://transl.ie/xyz>’
  2. ‘@person you would be able to translate this video <http://transl.ie/xyz>’
* Facebook Social Graph Integration
  1. Trigger actions on newsfeed
     + ‘Soud Hyder has just translated a segment of this video <http://translie.ie/xyz>’
     + ‘ Soud Hyder just received the Master Translator Badge’
  2. Social app/widget on Facebook
     + Allow translators to collaborate with friends on tasks
     + Allow translators to invite friends to work on tasks
* Voting
  1. Upvoting on a translation
  2. Downvoting on a translation
* Gamification
  1. Badges - <https://github.com/mozilla/openbadges>

# 2.0 Technical Requirements

## 2.1 REST API

A REST wrapper with a system generated key will be required to receive & expose data via REST protocols. Actual data sets to be read will be defined by the administrator. The API should have the following features -:

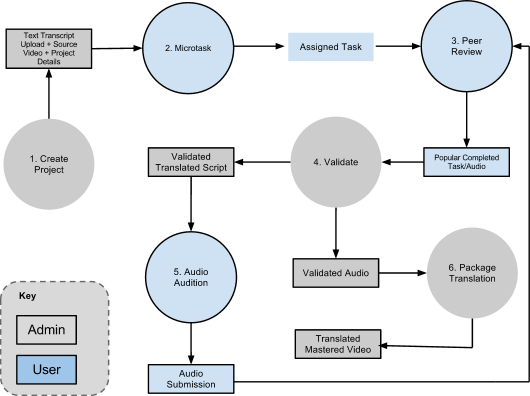
1. Parameters
   1. Task ID
   2. Sub Task ID
   3. Data
2. Authentication
   1. A system generated key should expose specific data sets

## 2.2 Modularisation

System should be modularised and built on a Service Oriented Architecture

# 3.0 Schematic Diagrams

## Context Diagram



**1. Create Project**

The administrator will create a new project and define the following parameters

* Project Name
* Project Description
* Translate From Language
* Translate To Language
* Translations (number of translations for each task)
* Micro-tasking options (break text by number of sentences, paragraphs, pages)
* Assign Editors
* Hashtags (text entry, official hashtags for the campaign)

**2. Microtask**

Once the project has been activated, a public url should be instantiated. Taskers/Translators can be anonymous or members of the system. This interface will allow members of the public to view the task queue and iterate through them with the following options.

* Translate (enter text and submit)
* Skip
* Shout Out (if the text is difficult, share with specific friend on Facebook or Tweet out for help, for tweets, official hashtag/s should be added)

**3. Peer Review**

The peer review process will allow users on the system to look through the submitted tasks (text/audio) and vote for them as follow. (We can start with anonymous voting based on browser sessions hoping that the law of averages will balance any false positives or negatives)

* Vote Up
* Vote Down
* Comment

**4. Validation**

Editors would go through the translated tasks, the ones with higher votes should surface higher or be highlighted. If more than one editor is assigned, a simple majority should be used, for 50-50 cases, a vote by a super editor should take precedence. Options as follows. Once all submissions for a project have been translated, the super editor will look approve and a text file of the script translation will be generated

* Accept (removes from translation queue)
* Reject (returns to translation queue and resets vote counter)
* Approve (function for super editors once all micro tasked processes have been successfully completed)

**5. Audio Auditions**

This will allow users to record/send links of their auditions, functions as follows

* Record
* Upload file

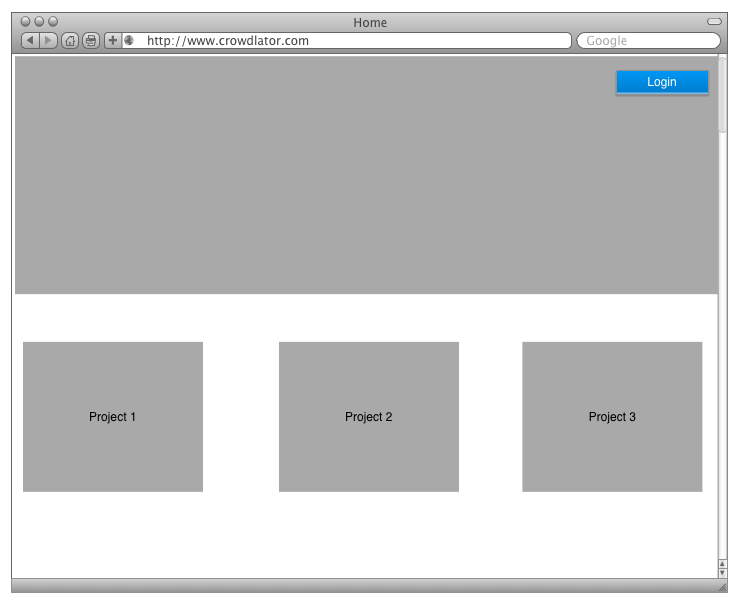
**6. Package Translation**

This will take the approved audio audition file and digitally re-master it with the source video, creating a digitally mastered file of the translated video. Options as follows

* Download
* Post to You Tube
* Comment

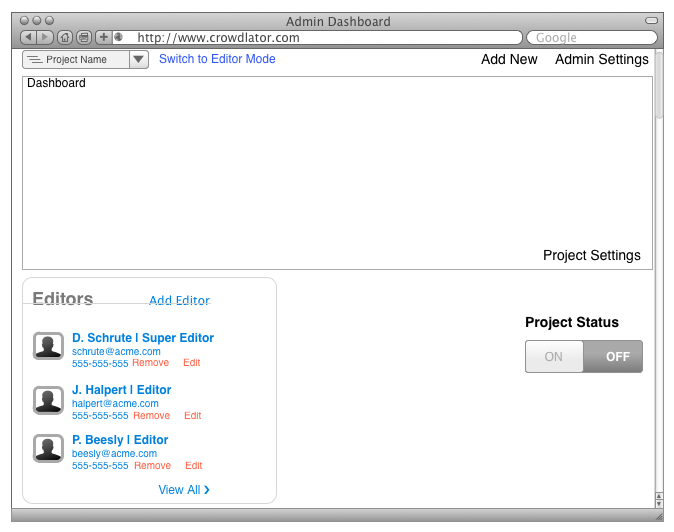
4.0 Wireframes

**1. Home/Index Page**



The homepage should allow users to login and should list current projects that are live. Pagination should be used to avoid cluttering the page. Project details should have a brief summary of the project and a link to allow users to access the project

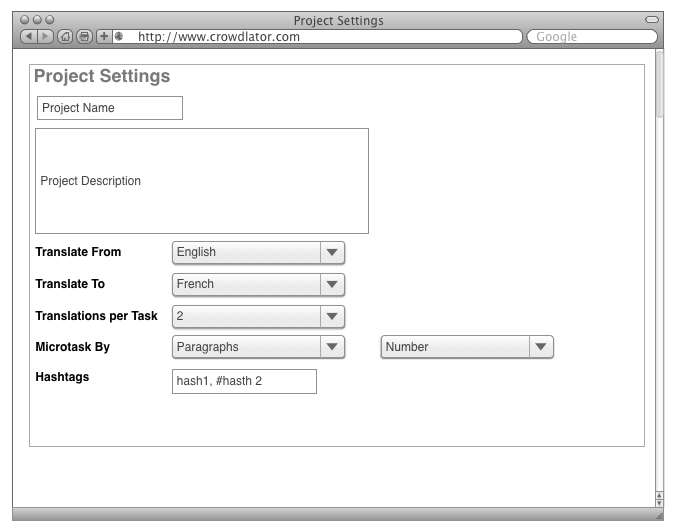
**2. Admin Dashboard**



The administrator dashboard will allow the administrator to iterate through current project and add new projects, it should also cater for the following -:

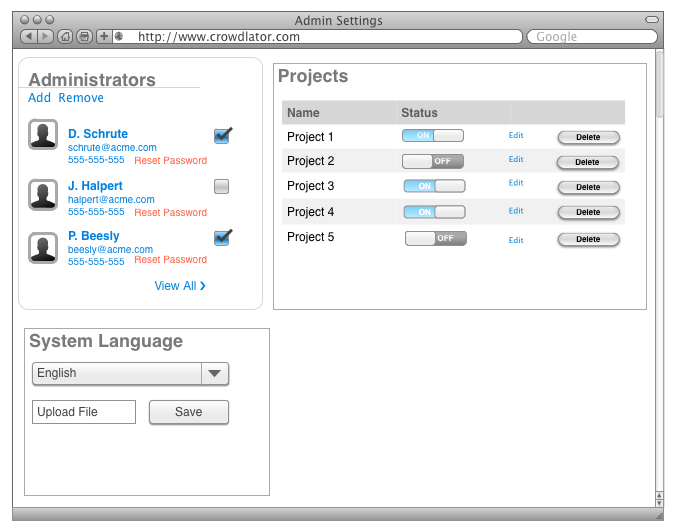
* Edit editor (privileges/editor type, contact information)
* Project Status (On - live/listed, Off - unlisted)
* Project Settings
* Dashboard; shows a summary of current project including -:
  + % Complete
  + Most active translators
  + Translation rate (hourly, daily)
* Switch to Editor Mode (switch current admin view to super editor view)

**3. Project Settings**



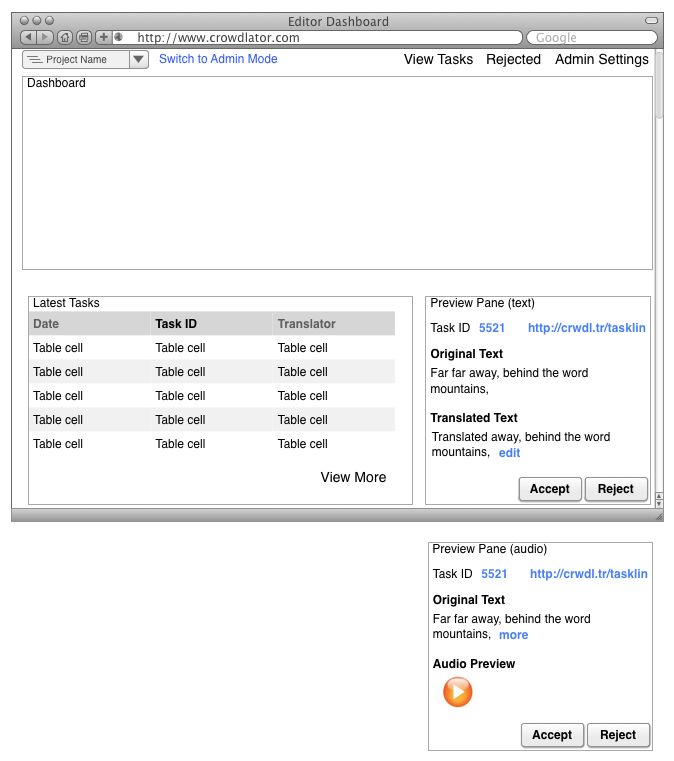
The project settings page should allow the administrator to edit meta data attached to the project, title, description and micro tasking options

**4. Admin Settings**



The administrator settings page should allow the administrator to manage other administrators on the system as a well a global console of all projects attached the instance with options to edit, switching and deleting. It should also allow the administrator to change the system language or upload a new language pack based on the i10 standards

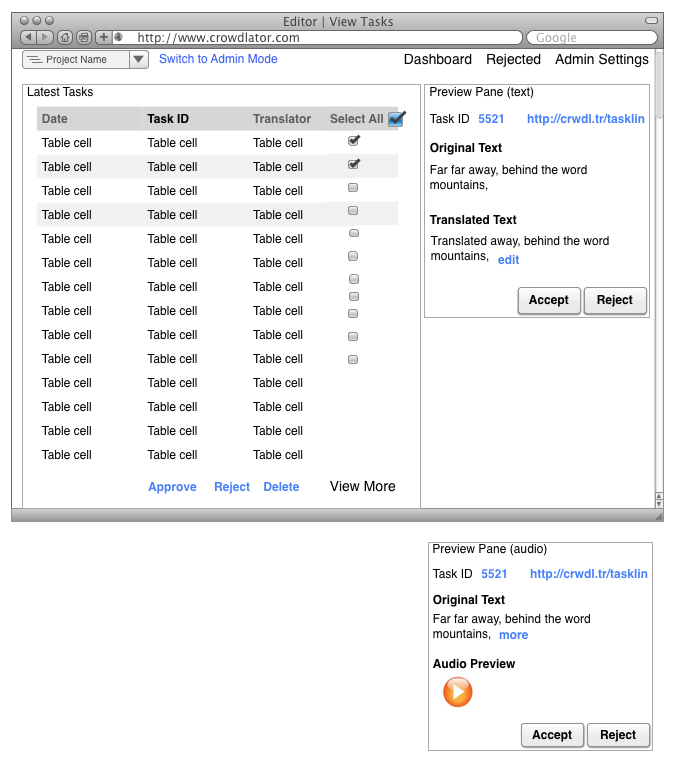
**5. Editor Dashboard**



The editor dashboard should allow editors to view and or operate the following information/functions

* Dashboard; shows a summary of current project including -:
  + % Complete
  + Most active translators
  + Translation rate (hourly, daily)
  + Current Task Stage
    - Text Translation
    - Audio Auditions
* Latest Tasks (table of latest completed tasks)
  + When row is clicked, task details should load on preview pane
* Preview Pane
  + Editor should be able to accept or reject task
* Admin Mode
  + Should be enabled for system administrators
  + When clicked it should open the admin dashboard screen
* Admin Settings
  + Should be enabled for system administrators
  + When clicked it should open the admin settings screen
* Rejected
  + It should open the rejected tasks page and filter by current project
* Project Name
  + It should allow editors to iterate through projects they have access to

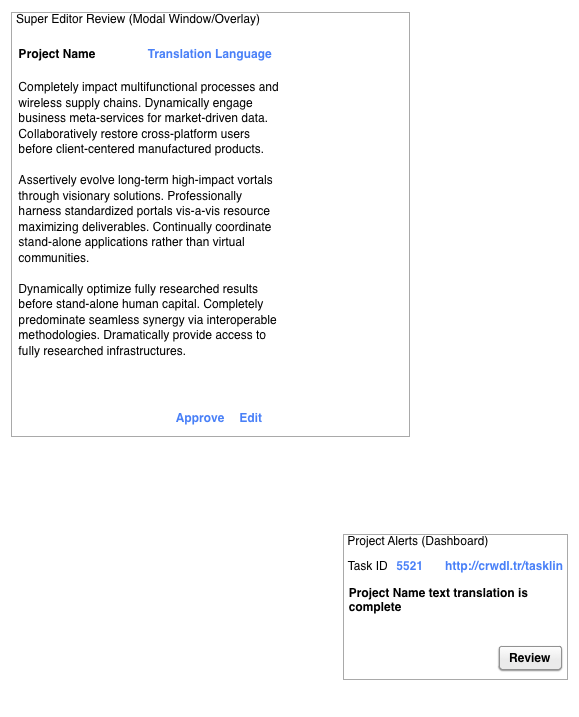
**5. Editor View Tasks**



The Editor View Tasks should allow editors to view a list of submitted tasks for the current project. It should have the following functions

* Project Selector
  + Specific Projects
  + All Projects (list on a *Last In First Out* basis - like an email mailbox)
* Preview Pane
  + Editor should be able to accept or reject task
* Admin Mode
  + Should be enabled for system administrators
  + When clicked it should open the admin dashboard screen
* Bulk Processing
  + Allow user to select/deselect taks
  + Approve/Reject/Delete with user confirmations via a dialog
    - e.g “Are you sure you want to permanently delete 24 tasks?”

**6. Super Editor Approval**



The Super Editor Approval should show up as a message on the dashboard and an activated menu item in the Super Editor View of the system. Once processed, these should be dequeued/deactivated from the front-end.These should list the completed project/s. Once “Review” is clicked a modal window should appear with the pieced up translation of the source with the following options -:

* Approve
  + Approve text as is and submits it for the next stage of the process (i.e. audio auditions)
* Edit
  + Makes the text editable and it should prompt the user for confirmation upon saving. “Save” and “Cancel” buttons should be instantiated on the UI once a user clicks “Edit”