‘To Do’ list for ExpenseReport Project

* Low
  + Consider to use email\_field instead of text\_field for email field in users/views/\_form.html file
  + Account\Settings doesn’t work
  + Input controls move to next row when there is an error on the form (with FF and explorer only, chrome is fine)
  + Decide whether we need to provide title for all pages
  + Code fragment in users.js.coffeescript file duplicated. Refactor it.
  + Switch to PostgreSQL database for production environment
  + Check indexes for all the tables and add if necessary
  + Money fields (advance pay) should display cents in the forms
  + Create link from ExpenseDetail to Expense
  + Add feature to logically delete a user
  + Check code for SQL injection (see <http://guides.rubyonrails.org/security.html#sql-injection>)
  + Pressing ‘Enter’ submits the form. Find out is this correct behavior.
  + Improve the layout of Expense attributes in ExpenseDetails screen (add owner name)
  + In MS Explorer disabled and readonly field look differently
  + Send emails asynchronously
* Medium
  + Remove all id fields from attr\_accessible
  + Implement SSL in production (see description in Rails tutorial in [Section 7.4.4](http://ruby.railstutorial.org/chapters/sign-up#sec:deploying_to_production_with_ssl).)
  + Find out whether we can use SecureRandom module to generate random password for newly created users
  + Implement server side processing for dataTables
  + Identify records which cannot be deleted or edited. Add the functionality for it
  + Implement “remember me” and “reset password” features
  + There is no ‘Please select’ line in select controls in Edit mode (New mode is fine, manager field in user is fine, too)
  + Get Exchange rate from web service
  + Redefine redirect\_to directories for after transitions
* High
  + Check for maximum column length in all grids
  + Check text field lengths in the database and forms
  + Disable edit button in appropriate expense view screen
  + Add change owner utility function
  + Add change manager utility function