Y0l0vv3bd3V$69's Homework 4

CSE 134b 2015

November 20, 2015

Team members: Alexie Sousa, Gil Olaes, Jason Tan, Max Takano, and Sam Marks

Table of Contents

Table of Contents 2
Important! Read Before Grading 3
Our technology stack 5
Changes to the Original Design 9
How We Meet the Grading Criteria 12
Contributions by Member 14
Android screenshots 15

Important! Read Before Grading

These are critical notes that we recommend reading before evaluating our site.

These notes will explain some things that might be confusing and will provide a way to evaluate the website correctly.

Logging in

Make sure to use a syntactically valid email when trying to sign up for the first time of the form xxxx@xx.xx eg. (a@a.com).

Hosted Site (http://cse134b-fa15.co.nf/src/list.html).

Our notifications integration OneSignal requires the site to be hosted to function properly. Loading the site locally will throw an error when attempting to subscribe to notifications on the list.html page. Please view the hosted site at: http://cse134b-fa15.co.nf/src/list.html when evaluating notifications. The rest of the site is up to date and may be graded here as well, although it is up to you.

Testing Notifications

If you would like an easier way to test notifications than waiting for our once a day reminder message, we made a page which can be found in the top level of our repo called: notificationsTest.html. From here you can immediately push out a notification to see it in action.

CSS/HTML Disclaimer

Although we did make significant efforts to clean up the original projects CSS and HTML (such as giving all files consistent indentation and removing tons of inefficient code). There are still some inefficient patterns and HTML practices such as wrapping many things with <label> or <input> that we didn't change so that we could focus on improving the overall look and of course reactive UI and server components. We should not be marked for the few things left behind of the last project, especially since we cleaned about 80% of it.

Advance Day button

There is a button at the top left of the list.html page that we included which would not be shown in an actual deployment of the project. This is for your ease of grading because it allows you to simulate what would happen when the app advances a day without actually having to wait a day. **Of course note that our app actually does perform all of this logic on our parse server when the day actually advances in cloud/main.js.** This is just a handy time saver we thought to include.

Troubleshooting

Please email maxytakano@gmail.com if you find anything in our submission confusing or something appears to be broken. We verified our submission on as many devices and browsers as possible so everything should be functioning.

Our technology stack

Front End

Our front end was entirely done using purely css, HMTL, and javascript. We did not use jquery or any extraneous libraries to assist with our client side's UI focused javascript. Everything was done natively to avoid bloating our code base with libraries and to ensure a fast reactive site. We decided to work off of the template project, enhancing and cleaning it up to experience and learn what it feels like to be a client side engineer given a website which must be improved.

Database

For our database, we used a integration with Parse for these key reasons:

- **User accounts were straight-forward and robust.**
  - The user must enter a VALID email address and password upon sign-up to create an account which we felt was adequate login validation for this assignment.
  - Upon sign-in or log-in, the user is cached into the browser. The user will even be logged in immediately upon future visits to the app.
  - Setting up the data base one we had the users was also easy, we were able to associate tables with rows and columns for the various data needed to be stored in the database.
- **It allowed us to implement asynchronous CRUD operations:**
  - **Create: Add a habit.**
    - Requires non-empty, unique (no current habits may have similar names) title. Throws validation error.
    - Able to select and/or upload image icon. By default, first image on list is used.
    - By default, no weekly frequencies are selected. Requires at least one frequency selected. Throws validation error.
    - By default, daily frequency is 1.
    - Notes are optional and can be viewed in the list page.
    - The user may opt out and not add a habit via the Cancel button.

- **Read: Habits List**
  - Dynamically builds the view of the habit list of the use.. Habits are segregated into two categories: active and inactive. Inactive habits are habits that do not apply for the day (e.g. weekly frequency is only on Mondays, today is Tuesday, therefore, the habit is inactive).
  - For each habit on the list, a blue cog icon next to its title will reveal the Edit and Delete buttons.
- **Update: Edit a habit.**
  - Prefills out the form with current existing information of the habit.
  - User has the same functionalities as Create operation and has the same validation requirements with one exception:
  - If user modifies the habit's title, the old habit with the old title will be deleted. If user does not modify the habit's title, the old habit will be updated instead.
  - The user may opt out and void any changes on this page by clicking the Cancel button.
- **Delete: Delete a habit.**
  - On the habit list, a user may reveal the delete button under the blue cog button. Deletion is as simple as clicking the delete button and confirming the deletion operation.
  - Deleted habits are removed forever.
  - Scheduled jobs allow for advancing the day (updating habits upon a new day). These jobs automate the updating of each habit of each user in the database.
- **It allowed for scheduled jobs**
  - Scheduled jobs allow for advancing the day (updating habits upon a new day). These jobs automate the updating of each habit of each user in the database.

Here is a schema detailing how our database was implemented.

[USER]:

id: String - implied

username: String - required

email: String - required

password: String - required

list: String - required - foreign key LIST(id)

[LIST]

id: String - implied

habits: Array of HABIT - required

edit_id: String - foreign key HABIT(habit_id)

[HABIT]

id: String - implied

habit:_id: String - required

icon: {

icon_src: String - required,

icon_id: String - required

} - required

weekly_freq: Array of Boolean - required

daily_freq: Integer - required

notes: String

completed_days: Integer - required

total_days: Integer - required

current_streak: Integer - required

longest_streak: Integer - required

completed_today: Integer - required

Notifications

For our notifications we decided to integrate with OneSignal. OneSignal allowed us to do all the things promised by Roost, except for free:

- Users can opt into receiving (pseudo-)realtime notifications by clicking on the “subscribe” button on the list view
- Once opted in, users will periodically receive notifications sent from our back end reminding them to fill in their habits.
- A test HTML page (notificationTest.html) has been created to allow you to send notifications to all subscribed users. Just click the solitary button on the page in order to send notifications to everyone that is subscribed.
- Behavior by device/browser:
  - Internet Explorer: No functionality due to IE’s inability to display desktop notifications, as they don’t have an implementation of the notifications API
  - Firefox: No functionality, as our third party technology is not compatible with Firefox except on experimental nightly builds. In client notifications are possible when the website is open, but this functionality was deemed too trivial to implement.
  - Chrome: Full functionality. After opting in, notifications will pop up as long as the Chrome browser is open (regardless of what pages are open). If Chrome is not open when a notification is sent, it will be displayed the next time Chrome is launched.
  - Android: Full functionality on Android’s Google Chrome browser. When opted in, notifications will appear as notifications on the user’s task bar immediately after they are sent by the server.
  - iOS: Not compatible unless the user downloads an application, which we deemed too much of a hassle to implement.
- Integrated with it Parse. Notification is pushed through OneSignal; Parse schedules these notifications as well as modifying their contents.

Changes to the Original Design

While our philosophy on this project was to keep the original designers intent so as to simulate a real world client side engineering job, we needed to make some major changes to allow the app to make sense. We stayed true enough to the design that designers would say “ah that is exactly what we meant!” as was brought up in lecture, while at the same time actually being a functioning habit application. Below are justifications for the changes we were forced to make.

General Changes

- Combined the welcome and login page, ultimately removing the welcome page. Welcome page was functionally similar to the login page and served no further purpose.
- When the user first signs up or logs in, instead of being taken to the add habit page the user is taken to the list page. This is more intuitive and simple then the original design because it teaches the user the the list page is their primary page. To highlight this point, we added a note teaching the user how to add a habit by adding an arrow and messaging pointing to the previously confusing (+) button.
- Color scheme: We changed the color scheme to be more blue centric since the gray was too light and didn’t work well with the light blue background. We took the darker blues from the (+) button in the original implementation.
- Interface changes: We added a log out button in the top right corner along with a greeting with the user’s email address.

Add Habit

- Habit Icons Animations: We took out the scaling animation for the habit icons. The selected icon no longer resizes to 1.3x the original size and doesn’t scale up while hovering over it. We found the animation to be bulky and distracting, especially when the selected icon switches since it displaces the margins all around; instead, a dull blue border appears during hover and a dark blue border permanently surrounds the selected icon. The exception is the “add an image” button, where the image outline turns from gray to black.

- Image Upload/Default Icons: The upload image icon was changed to say “add an image” to be more intuitive that it is a custom upload button. The upload function will allow users to pick an image file from their files to use as an icon. The image will be displayed as a new icon appended to the beginning of the list and automatically selected. All icons are saved to the database and the user can use it again later. We also added a couple more default image choices for users to pick from as well.
- Smart Defaults: There are smart defaults now for the Habit Icons and Daily Frequency to make it more convenient. By default, the first Habit Icon and a Daily Frequency of 1 is selected.
- Validations: Validations has been added to items without smart defaults. If the user tries to save a habit without adding a Habit Title or picking Weekly Frequencies, error messages will appear, prompting the user to fill in the missing fields. This way, the habits are created properly with all fields completed.
- Save/Cancel Buttons: The “save” button was changed to dark blue to match the new color scheme and a cancel button was added for when the user changes their mind during a habit creation or edit. We also added a color darken animation on the buttons for aesthetic and to show the hover event.

Habit List

- Advance Day Button: This was added as a debugger to see the functionality of the program. Clicking it will move on to the next day. **Again this button is for grading purposes only**. This would **not** be seen by the user normally.
- Edit panel: We moved the edit and delete icons into an Edit Panel in the top because we didn’t want it cluttering the Habit window and the user doesn’t need to see those icons at all times. It’s a small blue cog button next to the Habit Title that rolls out to reveal the edit and delete buttons when clicked. Clicking the cog button again will roll the buttons back in. For now, editing a habit will append it to the end of the list and also reset all the statistics associated with it.
- Weekly Frequency: Buttons showing which days the user selected in blue have been added right below the Habit Icon so the user can see which days they have the habit for.

- Thumbs Buttons: We changed the check button back to thumbs up and thumbs down buttons since a single check mark was not sufficient and didn't account for failed habits. We made them green and red, respectively, to denote their purpose. When hovered over, the colors darken slightly. Once a habit is completed or failed, the buttons will gray out and become un-clickable.
- Notes Button: A section for the option notes field was added since there wasn't one before. It's also pop-out. When the user clicks the notes button, the box will extend down to show the notes written for that particular habit. It will be blank if the user entered nothing.
- Statistics Copy: We changed the messages displayed on each Habit module and when the user performs an action to minimize confusion and clutter of information. We decided to show what mattered in an intuitive manner. The top portion now only shows the ratio of days completed to total days attempted. If the user forgets to input their completion or failure and advance day is triggered, this ratio will automatically count the previous day as a failure. The bottom will always show the counter for the day, and the counter will update as the user clicks the thumbs up button for successfully completing a habit. If the user clicks the thumbs down, they automatically fail for the day and the message changes to a failure notice with some words of encouragement. We moved the message about the record of days completed and how many days completed in a row to when the user completes a habit for a day.
- Inactive Habits: Habits are split into two sections: habits for the day and habits not active for the day. The top section is which habits scheduled for the current day according to the weekly frequency inputted by the user when creating the habit. Habits not scheduled for the day will be displayed below in another section and cannot be validated for the day, though deletion and edit are still available for it. As advance day is triggered, habits will switch in between the sections according to whether they are active for that day or not.

How We Meet the Grading Criteria

- 10pts for CRUD functions
  - We completed all CRUD functionality and even implemented it for notes and adding images
- 5pts for validations
  - HTML is fully validated. JS is mostly validated; JS may fail validation for trivial cases warnings are mainly thrown because the js linters do not account for a web/html environment
- 20pts for UI focused JavaScript
  - We implemented many improvements, making everything consistent and reactive. There are tons of subtle animations such as the notes/options sliding out and cleaner hover animations (buttons become more opaque instead of less opaque).
- 20pts for Notifications solution including any app modifications
  - Our notifications system which was explained in the first section actually pushes out notifications. Additionally it is robust enough to pop notifications up when the user opens a browser and even works on mobile devices.
- 10pts for overall code style and repo cleanliness
- Refactored given design and code base. Fully commented. Code is structured well with correct indentation and code spacing. No unused files in the repo.
- 10pts for browser conformance and exception handling
- HTML validates. All HTML, CSS, and vanilla JS technologies were researched and were made sure to operate on all browsers (Safari, Chrome, Mozilla) before use.
- Notifications operates on Chrome, Safari, Android and iOS. They do not work on Mozilla Firefox due to alpha stage development on OneSignal.
- We have included user validation (user must be logged in to view add.html/list.html; they will be redirected to log in page). Forms have been validated for all user input cases. CRUD operations have been secured in terms of race-conditions (If user spams the Thumbs Up button, it will only apply once).

- 5pts for README file
    - detailed report. seen here

Contributions by Member

Alexie Sousa

Worked on the redesign of the UI

Gil Olaes

Back end cloud code and desktop notifications

Jason Tan

Log In/Sign out/CRUD operations with Parse integration, habit list population, form validation

Max Takano

Log In/Sign out/CRUD operations with Parse integration, habit list population, form validation. Implemented a ton of the reactive UI.

Sam Marks

Did some research on parse

Android screenshots