iResume

Milestone 7

Final Report

Adewale Adekoya I Saige Liu I Rui Ma I Mark Walle I Huiying Wang

April 5, 2016

SENG 310 - *iResume* Project Milestone 7

Adewale Adekoya, Saige Liu, Rui Ma, Mark Walle, Huiying Wang

2016-04-05

Table of Contents

Adewale Adekoya, Saige Liu, Rui Ma, Mark Walle, Huiying Wang

2016-04-05

Problem Description and Motivation

The process of making resumes is generally an iterative exercise that adjust and expands upon content from your pre-existing resume versions. Content creation is usually only done after working at a new job, gaining new skills, publishing new works, or achieving accreditations. Steps in creating a new resume version typically involve finding the last resume you wrote, updating it with short sections of new content or content drawn from other versions, adjusting that content to best match the requirements of a potential position, saving one editable-format copy and exporting a static PDF copy for distribution. In each of these steps, time and energy that may have already been spent in previous iterations of the same task is spent. We search through past resumes to find the ideal content for the job, and we retrace the same sequence of steps virtually every time. When we store the resumes, we often must make a choice between accessibility, security, and risk of physical media failure. Even document portability can become a nuisance as we download new software or transition between operating systems. Eventually, numerous copies of your resume exist in various formats, scattered across an assortment of storage devices, and every version has only minor differences from any other.

Resume writing occurs when you are looking for a job, which is already a stressful enough process for many people. If a sufficiently well-developed application were developed to handle the burden of contending with the disorganized, time-consuming, and repetitive aspects of resume writing, then we would have more time to focus on applying to jobs in a timely and less stressful way.

Motivated by these problems, and by the urgency of looming co-op placements, our team focused on developing prototypes for a web-based application that would be have a target focus on resume creation, editing, storing, and distribution which would provide accessibility without compromising security. The outcome of this effort is *iResume*.

Existing Solutions

The solution we have prototyped implements the concept of version control as is implemented in Git; and in cloud document storage, such as GitHub. These systems, while integral to modern programmers and software development companies remain largely inaccessible to the average user due to barriers in technological fluency. Our solution abstracts away the complexity of repository management, and requires no special skills or syntax. As a result, iResume has far fewer features than most version control systems; but the benefit of this is that average users will only experience fast and easy ways to recall, edit, and save new iterations of their content, without needing to understand how it happens.

LinkedIn is a social platform where users are, in essence, making their resume content available to some public or partially-public audience. LinkedIn itself doesn't have effective tools to rapidly prototype and save new versions of traditional paper resumes, which, by-and-large, is still the expected method of applying for positions (even in across technology sectors). In the course of building this tool, our team did find that at one time LinkedIn Labs had an application called Resume Builder. This software is described in LinkedIn blogs and describes an API reading in resume content from a LinkedIn profile that has authorized it and parsing the content into a generic template; however, at the time of this writing, the DNS for resume.linkinlabs.com is no longer resolving. We could not find descriptions of this application maintaining a library of previously generated resumes. What it did demonstrate to us was that bridging our service through LinkedIn's RESTful API to their existing profile is a viable option to scrape content into iResume.

There is a small handful of web based tools that offer some similar features as iResume. We examined some of the most popular: Resumonk, Visual CV, and ResumeGenius. While each offers some of the features iResume does, none offer every feature; and many are frustrating user experiences, having incomplete, buggy, or unintuitive interfaces and workflows. Some offer resume library and building functions, but do not allow rapid rebuilding of resumes based on previously composed resume versions. Worst of all, they aggressively push paid subscription services in order to accomplish the minimum assortment of features which would actually make them useable (such as template selections, exceeding length limitations, storing more than one resume in a library, removing branding).

The Proposed Solution

Our solution to the problem stated above would be a web-based application that focuses on resume creation, editing, storing and also provides the distribution and accessibility without compromising security. The solutions to the problem would include a system that:

- Allows users to build a resume by reusing block components
- Eliminates document clutter in the user's storage space
- Provides version control on their resumes
- Creates a unified environment to manage, edit, create and publish their resumes.

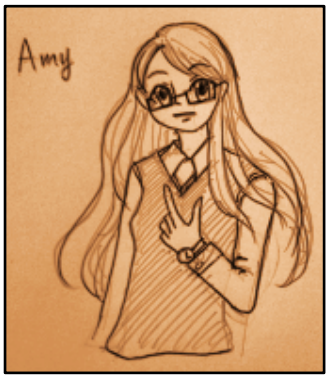
iResume would be filled with content from the user either through manual entry or extraction from their LinkedIn page. The user would have the option to create a new template or choose from a set of curated templates. If they choose from the set of curated templates, iResume would successfully parse the information into the template. This would allow the user to easily manipulate the content of the resume because the templates are organized into block components. A simple drag and drop of a content section will change an entire section of the resume. Content sections would retain version control, allowing the users to rapidly recall subsections or their entire resume.

Adewale Adekoya, Saige Liu, Rui Ma, Mark Walle, Huiying Wang

2016-04-05

Personas

Our personas were developed early in the semester and were fine tuned by the results of requirements gathering, and use case design and user testing. The end result were personas that we used to demonstrate use cases in the final prototypes of iResume. The personas we developed were Amy and Josh, who were both Uvic co-op students, but who each had their own reasons and objectives for using the iResume software.



Amy

Amy, a 20 year old student at the University of Victoria. Now she is in her second year studying a combined major program of Health Information Science and Computer Science. She just started the co-op program this semester, and has written a basic resume for computer science co-op.

Josh

Josh, a fourth year Mechanical Engineering student at the University of Victoria is on the verge of completing his degree and currently needs one more co-op work term to graduate. Josh knows that it takes a lot of effort to get a job offer, he has a lot of experience when it comes to writing resumes and applying to various companies.

Scenarios and Use Cases

Amy and Josh each had a scenario and use case developed for them that would let them satisfy unique goals with the use of the software. The primary difference between them was that Amy would be a first time user, while Josh would have used the software to a point where he had many resumes stored within the system. With that being the case, each use case demonstrated different aspects of iResume.

Adewale Adekoya, Saige Liu, Rui Ma, Mark Walle, Huiying Wang

2016-04-05

Amy's Scenario

Amy wants to apply for almost all the jobs posting on the co-op website, she has to first create a resume and make changes to fit in different positions. In addition, being new to the process of developing new resumes and applying for multiple positions, Amy would like to to have a system to help her build new resumes. She went for iResume, and wanted to play around with the system and test out different functions. To seize more opportunities, Amy has posted her resume onto some public websites such as LinkedIn, which included her Email address and phone number, in order to get more responses from companies. However, after a few days, she received a huge amount of unsolicited phone calls. Thus, she pays utmost attention to privacy protection when she is under job hunting. She would love to have an App that can help her store the old version of resume without changing any format, and help to protect privacy.

Amy's Use Case

description of Amy's Use Case IN THE FINAL PROJECT.

1. Amy creates an account, enter username and password, besides, click the checkbox of "dynamic password" to sign in.
2. Two input boxes appears below the checkbox of "dynamic password" to ask Amy to enter cell phone number and the dynamic password.
3. Amy enters cell phone number first and click "send" button, then she enters the dynamic password that the system sent to his cell phone.
4. Then system tells her verifying successfully.
5. The system loads to iResume's main interface.
6. Amy clicks on Menu button
7. She clicks on the "Setting" button to check the setting function, and then closes the window.
8. She clicks on the "Profile" button to check the profile function and then closes it.
9. She clicks on add New button. The system popped out several options: to import or create resume information (Import from LinkedIn, Upload from file, Start from Scratch)
10. She chooses one of the resume template and creates a new one.
11. She saves the current editing resume into the system
12. She clicks "Export" and select whether export a PDF or DOC version of the resume.
13. The browser downloads the PDF and the download appears in the footer area of her browser;
14. She clicks the downloaded file to preview the version of his resume created using iResume.

Josh's Scenario

Josh saves different copies of his resume on his PC, he boots up his trusty PC and opens multiple tabs, each tab is a copy of a resume with different skill sets and experience. Josh has undergone this process every year to get his last 3 work terms. He would love a system that simplifies this process, a system that allows him to manage his resume in a single sit down, a system that provides privacy protection, also a system that would be relatively easy to use. in anticipation that his last work term job hunt would not be too much of a hassle.

Josh's Use Case

1. Sign in using the account that already exists (joshjobseeker1992);
2. After logging in, a landing screen presents him with a library of previously created resumes.
3. He selects the first resume, and clicks "view";
4. He selects the resume he wants to edit and presses the "Edit" button on the file browser window;
5. After the system loads that resume, Josh deletes the Objective section.
6. He moves the mouse on the Work Experience section, and to edit that section button.
7. The system brings Josh to the editing window. He clicks the version selection button.
8. Then Josh reviewed history version.
9. Josh choses an earlier copy of the the second model and selects to edit its content.
10. Then Josh saves the version he edited
11. After saving, Josh wants to insert an objective
12. Josh choses a recently written Objective from the options, and saves it into place on the resume.
13. Josh is presented with the resume again, now having a new objectives section appearing on top, and the earlier selected work experience section.
14. Then Josh closes the page, returning to the library.
15. At last, Josh logout.

Evolution of the Prototype

The prototype changed dramatically from the initial inception to the final project's prototype. The original prototype was built around a split horizontal screen, intended for

exclusive use in a browser with a wide aspect ratio. Figure 6.1 shows the split screen view, which was central to the design of iResume.

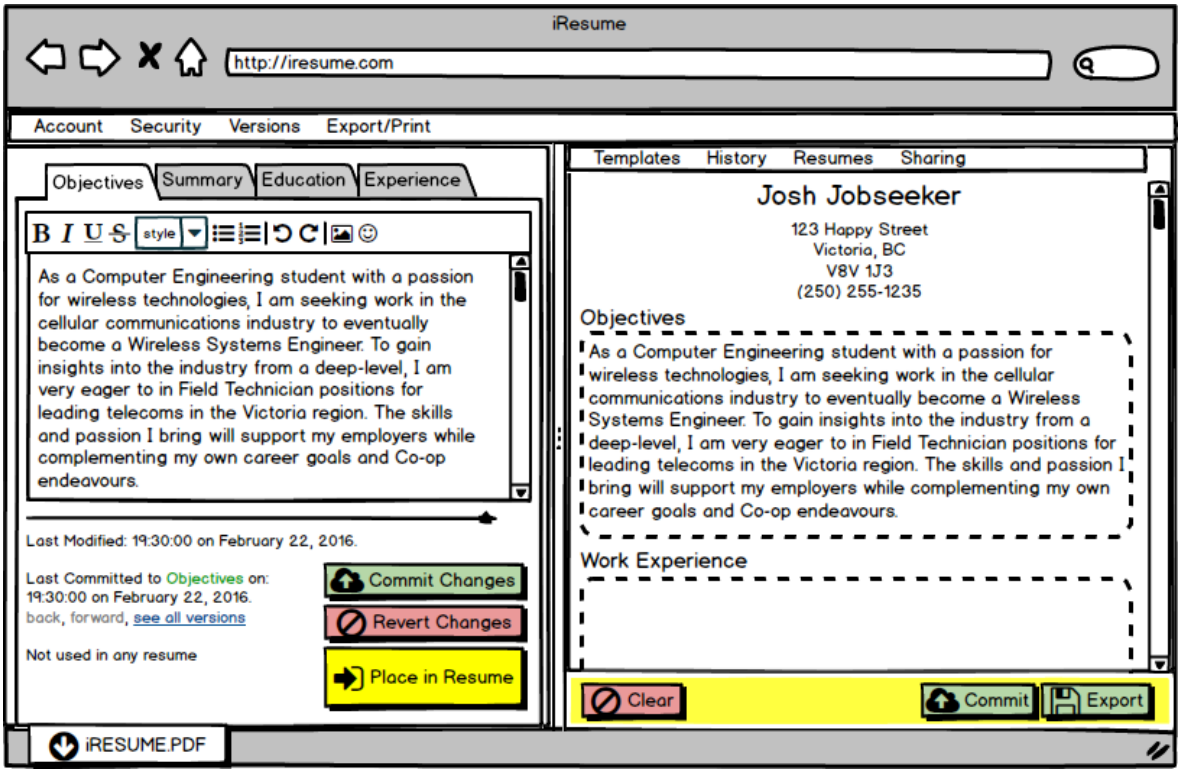


Figure 6.1 - The first low fidelity prototype showing the horizontal split view separating editing options on the left from editing options on the right.

The problems with this interface were revealed in user testing and feedback form cognitive walkthrough evaluations. The most obvious problem is the limitation of this view to a wide aspect ratio, which makes the service unsuitable to the responsive design required for presentation on small screen and mobile devices which are most commonly used in vertical alignment.

The next problem was found when we tested cognitive walkthroughs. There was confusion and mistakes made around how to interact with the assortment of buttons on the screen. It wasn't clear to users what "commit" intended; and even just selecting the correct button took too long for the user to find, and was often preceded by several mis-selections by the user. Essentially, the prototype's design had not considered constraints in design, instead it attempted to include as many functions as possible in a single screen.

Another concern with the low-fidelity prototype was, at that point, we had not yet envisioned a way that users would access previously generated resumes or sections. After considering this, we developed a interface for Josh to select a previous version of one of his

work experiences, which allowed for a scrolling selection of earlier iterations of the content to edit as a new iteration, or just to insert into the resume template (see Figure 6.2).

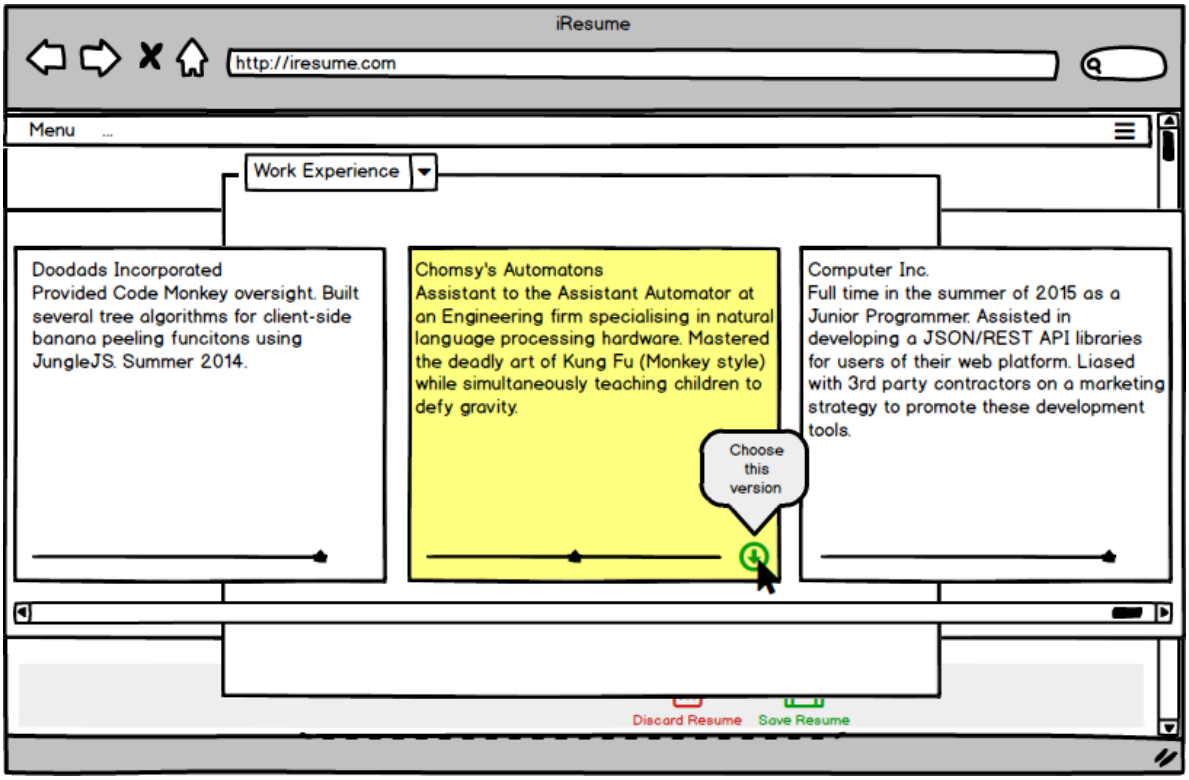


Figure 6.2 - The low-fidelity version of section selection. After choosing, the selected section would be inserted under the “Work Experience” section of the template shown in Figure 6.1.

The use case itself evolved significantly over the course of this project, informed by feedback from user testing and peer evaluation. Initially, Josh was to sign in as a returning user and upload a Microsoft Word version of an existing resume. iResume would parse the uploaded document into the relevant sections and reformat the presentation into a selected template. The idea required a “hand waving” explanation of how iResume would actually parse the uploaded document. In reality, the variability between styling and presentation of users source files would be so great as to make such a parsing engine extremely difficult to develop. It would only practically work if the resume were originally written in a markup language with predictable section names.

As a result, we decided that for the final prototypes, Josh wouldn’t import a resume from his computer, but instead would edit one already existing in the system. This would also require us to create the resume library interface, which we were not able to prototype in low-fidelity due to time constraints, but regardless, the feedback on the final prototype was positive. Figure 6.3 shows the library view seen after Josh signs into iResume.

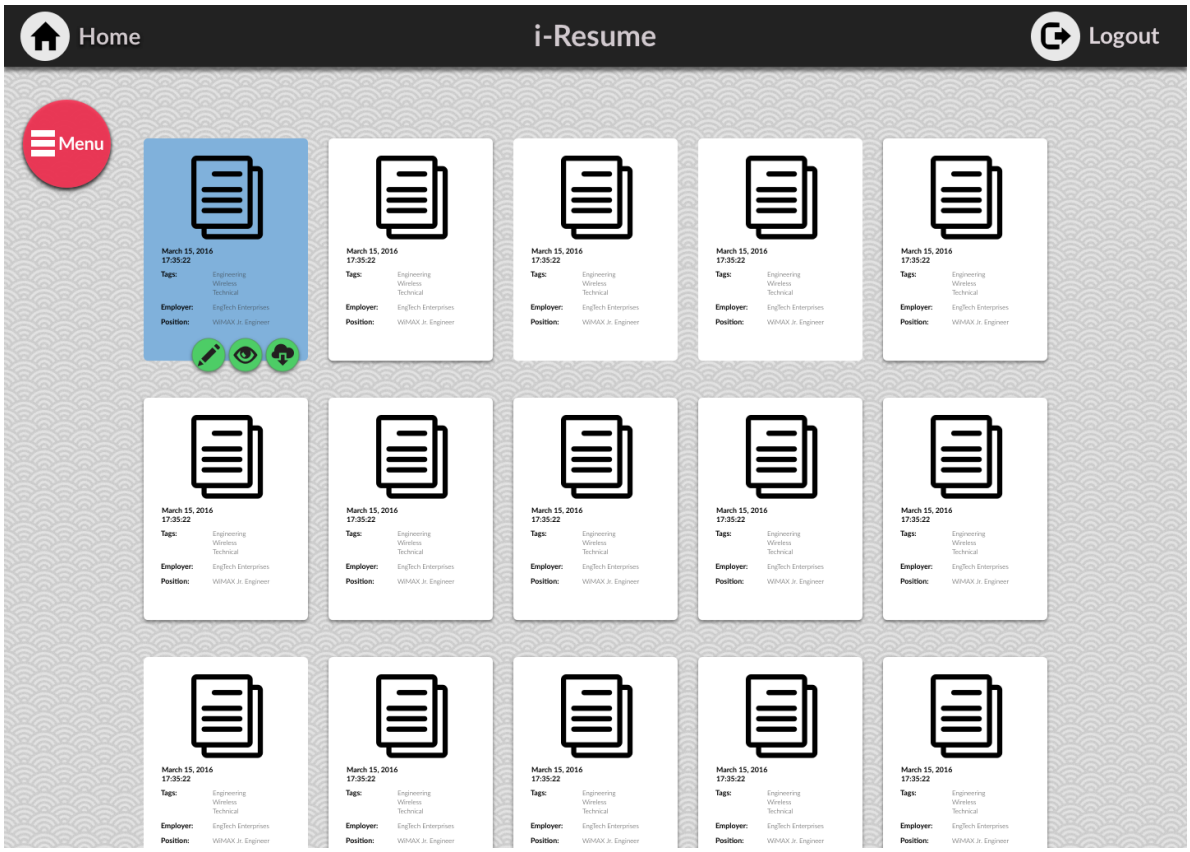


Figure 6.3 - The resume library in the high-fidelity prototype, developed to accommodate the changes made to to Josh's use case.

Figure 6.3 also hints at the departure from the original design horizontal split scheme, as well as the transition from written buttons to iconographic ones. It also demonstrates a cursor hover-effect on one of the existing resumes; hovering over presents immediate options to the user, and are constrained to the single file by virtue of their disappearance when the cursor is not overtop the file. This single panel view carried on into the editing and templating presentations, the low-to-high prototype evolution of which is shown between Figures 6.4 and 6.5, respectively.

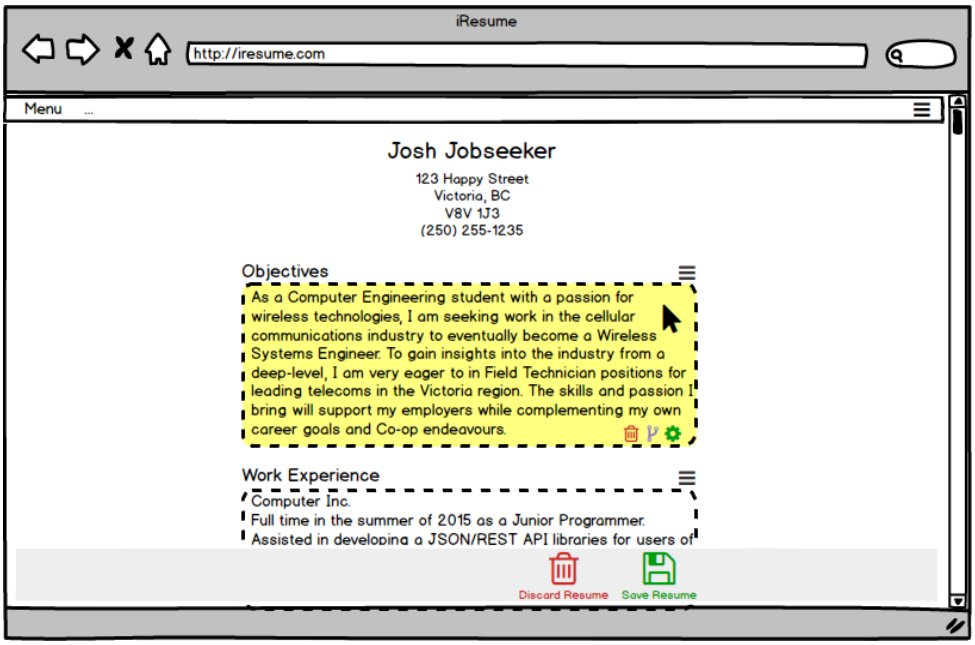


Figure 6.4 - The low-fidelity prototype of an unsplit interface, showing hover options on a section.

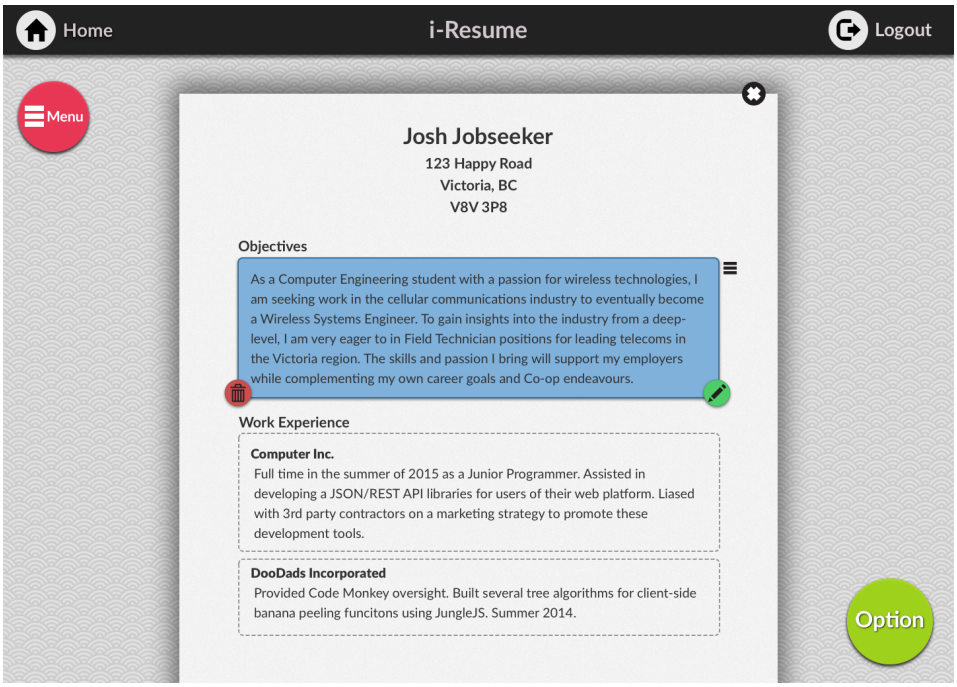


Figure 6.5 - The high-fidelity prototype representation of the same screen in Figure 6.4.

High Fidelity Prototypes

We developed a high fidelity prototype for each use case. Amy's shows an new user, and Josh's shows a returning and heavy user. Figure 7.1 is the login block used in each use case.

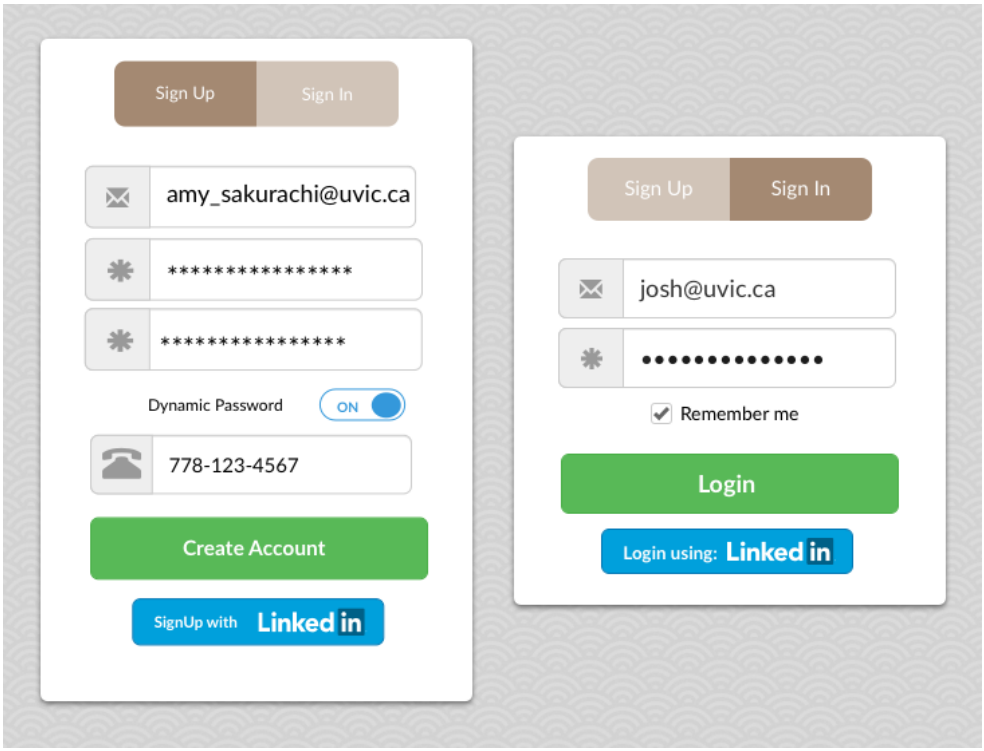


Figure 7.1 - The Sign Up and Sign In panels shown together, after completion.

The library, and resume template screen have already been demonstrated in the Evolution of the Prototype section.

Editing is done by section instead of as a single document. This allows iResume to manage each section under version control. The editing of a single section of Josh's resume is shown in Figure 7.2. Selecting from pre-existing content was accomplished through a swimlane style interface with versions of each section accessible as a chronological list beneath each content section. The overlay appears after selecting Figure 7.2's purple button indicating with the versions icon. This selection interface is presented in Figure 7.3.

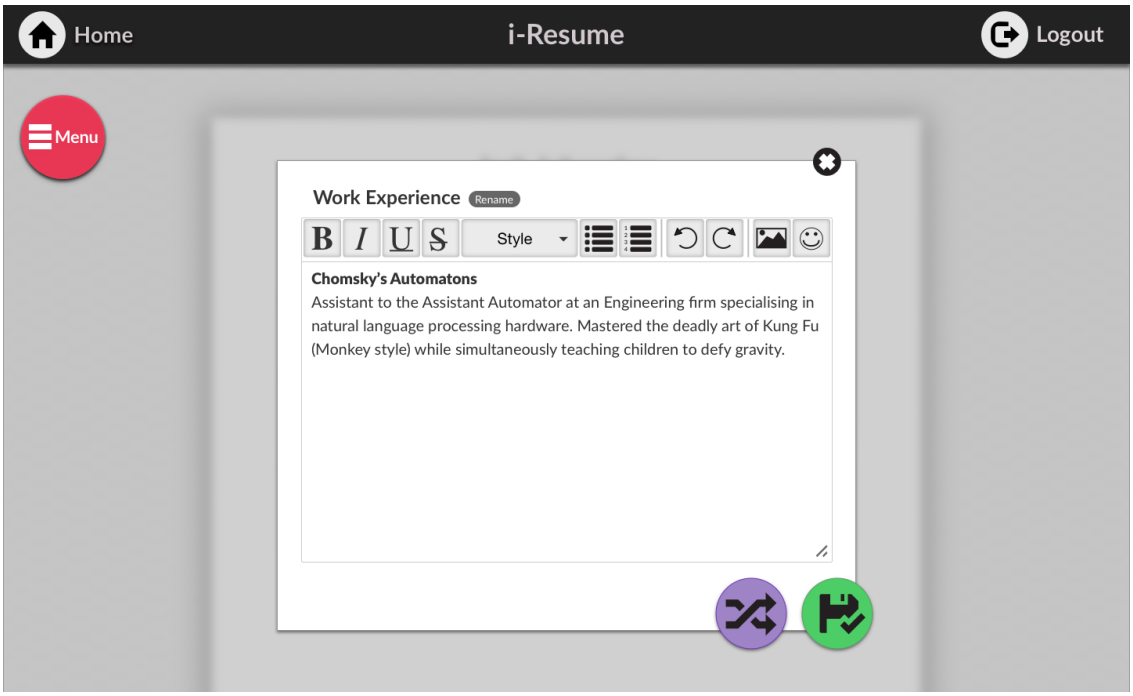


Figure 7.2 - Editing a section of Josh's Resume, presented as an overlay in the resume templating screen. The editor also shows options to select a different version and save the edits, which closes the overlay and inserts the section into the resume.

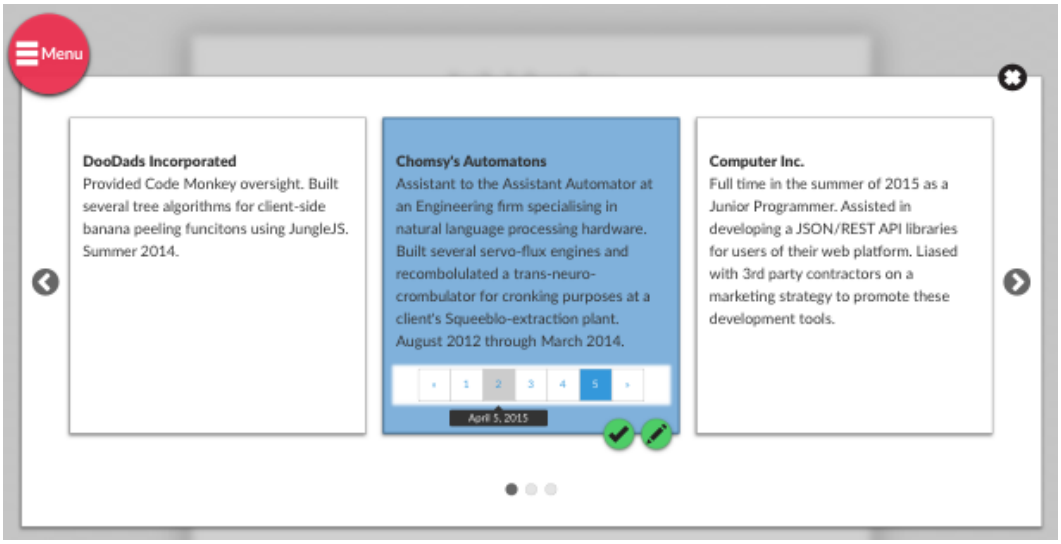


Figure 7.3 - A cropped view of swimlane interface version selection overlay and hover options

Future Work

Even in the high-fidelity prototype, minor differences are apparent between each use case. This was an artifact of multiple designers attempting to simultaneously and rapidly create the prototypes required, highlighting the difficulties professional design firms must encounter in maintaining quality control in large teams of designers. In future, Sketch should be approached with the team in mind.Currently, Adobe's team features (such as linked objects and shared asset files) are more developed than Sketch, but using a traditional shared workflow, symlinks, or finding the right Sketch plugins to improve teamwork should make this easier. In addition to unifying all prototype assets and deciding upon common styles to implement consistently throughout the product, we would also focus on the language and personality iResume expresses.

Our video presentation described the goals of iResume and proceeded with a cognitive walkthrough of Amy's use case. It would be good, in the future, to generalize the presentation of the product, to market features rather than focus on a use case.

After we completed the high fidelity, we still did some test and research let participants to use our web-system. The process reflect some issues we need to continue to promote. To be specific, when selecting variants of a section, as in Figure 8.1.

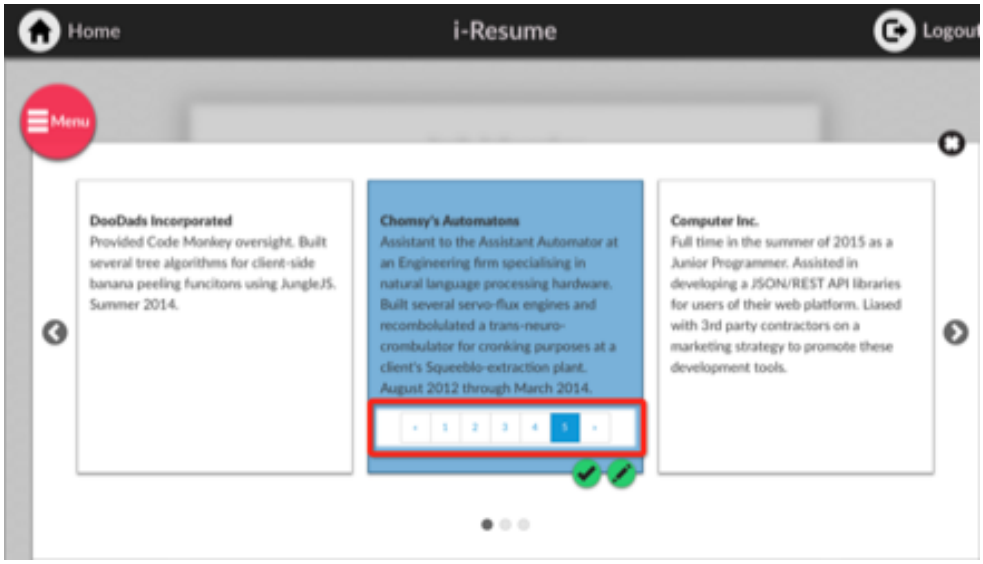


Figure 8.1 - Highlighting the version selection options underneath a section.

When users encountered the overlay shown in Figure 8.1, before our participants moved their mouse, they were both confused about the numbers listed. They wondered if

these numbers represented pages of the content, or other versions? We need to improve the design of this part, like when the mouse point point to the number, there will be a thumbnail to show the changes of that version.

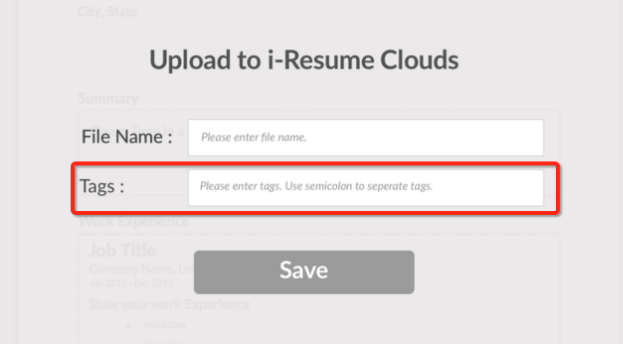


Figure 8.2 - When saving a resume and choosing meta-data characteristics.

In Figure 8.2, the Tags is empty. It needs users to fill in keywords based on their satisfaction. Some of our participants propose that the Tags function can be settled by our web-system, which depends on changes that users made, and some high frequency words. And tags can afford user option to enhance.

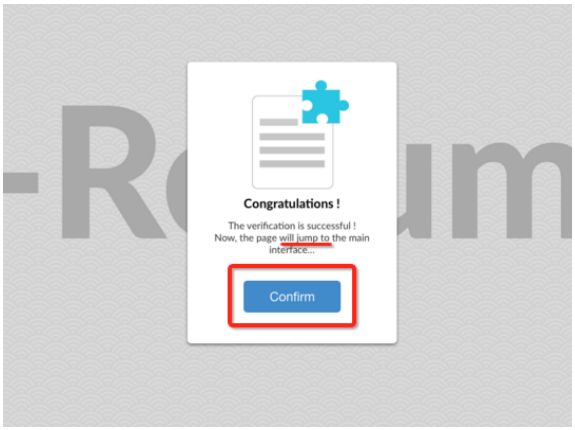


Figure 8.3 - Verification message after successfully saving and before loading library.

For the Figure 8.3, most of our participants hesitate when they operate on this step. As the underline words of “...will jump to ...”, they thought this page will jump to another one automatically. Thus, they just keep waiting. We may need to revise it as an automatic one or a manual one when we design in the future, in order to eliminate misunderstanding.

There is one of our participants suggests that we should add a push function. Because the web-system is mainly about resume editing according to different job. Thus, we can add reminder function of “due day of the resume ” or “deadline of the job application”. This function can combine our resume system with user experience together.

We will reexamine the whole design process from users' perspective instead of designers' perspective. For example, if we're a (potential) co-op student, which kind of function do we need to see the most in i-Resume web-system. Based on this idea, we will integrate some functions, and add more functions inside the web in the future.

Lessons Learned

Challenges and Solutions

The first challenge our group had to overcome was to decide a project topic. We had an disagreement with two topics: one was iResume, and the other one is an online travel planner. As the discussion went no decision, we asked instructor and TA for help and advice, and we finally made a decision. To coordinate with the team better, we did a self and team evaluation. According to the evaluation, we predicted challenges we might meet in the future, and concluded things we can improve.

Another challenge was to improve our prototype based on our user observation. According to the user observation, we found out that the user get confused when first coming into the editing page. There are too many information on the interface. In addition, the color of the buttons did not really popped out. Then we discussed the observation with our team. We concluded that all of the users we observed had the same confusion, and the problem occurred at the same place. Based on that, we decided to change the interface completely, make it more simple and less information on the same page to reduce user confusion.

Moving on to build prototypes from low to high fidelity, since two of us use OS X and others use Windows, we found it difficult to use a same software to build high fidelity prototype. We decided to use two softwares at first, Sketch and Photoshop. However, since we could not find same plugins for Photoshop, and unify the UI using different softwares, we decided to only use Sketch. Also, learning to use Sketch was also a challenge. We went through a couple online tutorials and helped each other to master the software.

Interesting and Valuable Things about the Project

- Learned how to cooperate with team members, share ideas, solve disagreement and then reach a consensus.

- Learned to use the prototype design softwares such as Balsamiq (low fidelity prototype) and Sketch (high fidelity prototype) to develop a easy understanding prototype.
- Learned using Invision to present the prototype in an interactive way. The user can communicate with the prototype.
- Learned how to use software Any Recorder (recording prototype video) to record the demonstration of the prototype.
- Learned the basic process of design and develop a application considering the Human-Computer-Interaction principles.
- Learned how to develop a imaginary persona, persona, and actual use case to predict and lead the process a user might be using the application.
- Learned how to hold a user observation to gain feedback from real users to help finding deficiency and improving the prototype.

Links To Prototypes and Video

Interactive versions of the final prototypes are served through InVision;
Amy - Amy's use case
Josh - Josh's use case
The video presentation of iResume (which uses Amy’s Use Case) is available here.