Naveen Krishnamurthi - naveenk1@stanford.edu
Kevin Miller - kmiller4@stanford.edu
Danial Shakeri - dshakeri@stanford.edu
GitHub Repo: https://github.com/Gasparila/cs194
01/13/2015

Project Proposal

Description of the Project:

The purpose of our project is to develop payroll software that uses timecard data to accurately compute payment information for employees in a manner that is compliant with California labor laws and regulations. Our team envisions a web platform that seamlessly integrates with a variety of different forms of timecard software, and allows for effortless access of payroll calculations from the cloud. Employers that take advantage of our service will use our software to provide information about employee work hours, wages, etc. for a given pay period. Our system will then consolidate this information, and generate a pay statement for each employee that can be easily and securely accessed online.

Need for the Product:

There are around seven hundred thousand businesses in California that have multiple employees. Each of these businesses must pay their employees and do so in a manner complying with California labor laws. Laws governing pay, including how to pay overtime and how long records must be kept, are far stricter in California than any other state. For instance, in California, pay statements must be saved for all employees going back four years. This contrasts against the regulations of almost every other state, which tend to require that records must be saved only for current employees, and only for a period of six months. Another example can be seen in the case of how overtime is accrued and calculated, which can be complicated for employees who work in multiple positions. In this environment, there is a demonstrated need for payroll software that can handle all of these calculations for employers, who should not be required to compute it themselves.

Our payroll software will be useful for all businesses in California that employ multiple employees and do not want to do payroll calculations by hand. Essentially, an employer would provide us with their timecard data, and we would then be responsible for all time calculations and creating pay statements that the employer could access at their leisure. Our team will work with companies to provide accurate pay statements and have records that are easily accessible by both the employers and authorized employees. This is a huge convenience for employers, and is not currently offered by our market competitors.

The need for our product is also a result of the fact that even though there are at least three major payroll companies, **none** of them currently comply will all of California's regulations. For a more detailed explanation of this problem, please see the section below entitled "Discussion of Competing Products." The result is that there is currently a large market demand for payroll software that is inexpensive, easy to use, and compliant with all California

regulations. We plan to meet this market need, and create a product that provides a valuable service for all California employers.

Potential Audience:

The main audience for our product consists of all businesses that operate in California, given that our payroll system is designed to be compliant with California payroll laws and regulations. In the future, however, we plan to modify our software in order to make it compliant with all state and/or federal regulations, thereby allowing us to meet the demands of businesses all over the United States.

Discussion of Competing Products:

There are a number of different companies that make payroll software. The chart below, taken from qbgarage.com, has information about the cost of some of these products as well as the services they offer (more data can be found at http://www.qbgarage.com/payroll-service-comparison):

January 9, 2011 Description (note 1)	Intuit Online Payroll Basic	Intuit Online Payroll Plus	QuickBooks Basic Payroll	QuickBooks Enhanced Payroll	QuickBooks Assisted Payroll	Paychex Flexible	Paychex Small Biz Paperless	ADP Compliance w/Pay Conv
Employees (note 7)	10	10	10	10	10	10	10	10
% employees using direct deposit	100%	100%	100%	100%	100%	100%	100%	100%
Pay periods per year	52	52	52	52	52	52	52	52
Payroll frequency	Weekly	Weekly	Weekly	Weekly	Weekly	Weekly	Weekly	Weekly
Monthly fee (note 2)	$ 38.50	$ 52.50	$ 14.53	$ 21.82	$ 58.65	$ 169.00	$ 119.00	$ 311.57
Per employee/pay period fee	N/A	N/A	N/A	N/A	$ 1.00	N/A	N/A	N/A
Direct deposit fee per deposit	$ -	$ -	$ 1.25	$ 1.25	$ 1.25	Included	Included	Included
Additional discount (note 2)	$ 55.02	$ 97.02						
Monthly fee for 1 year (note 2)	$ 462.00	$ 630.00	$ 174.30	$ 261.80	$ 703.80	$ 2,028.00	$ 1,428.00	$ 3,738.80
Check fees	Note 1	Note 1	Note 1	Note 1	Note 1	Note 5	N/A	Included
Per employee/pay period fee	N/A	N/A	N/A	N/A	$ 520.00	N/A	N/A	N/A
Direct deposit fees	$ -	$ -	$ 650.00	$ 650.00	$ 650.00	$ 780.00	Included	Included
Setup fee	N/A	N/A	N/A	N/A	$ 49.00	$ 50.00	Included	$ 200.00
Prepare W2's (note 3)	Note 4	Note 4	N/A	Note 4	$ 82.50	$ 97.50	Included	$ 84.00
Tax preparation/filing fees	Note 4	Note 4	N/A	Note 4	Included	$ 754.00	Included	Included
QuickBooks interface	Included	Included	Included	Included	Included	$ 180.00	Note 6	Note 6
Total annual cost	$ 462.00	$ 630.00	$ 824.30	$ 911.80	$ 1,956.30	$ 3,659.50	$ 1,428.00	$ 3,822.80

Intuit's Online Payroll Basic and QuickBooks' Basic Payroll both offer the exact same service that we hope to provide: an interface that allows employers to submit timecard data and receive pay statements. They do not provide additional features such as actually paying the employees, computing tax and social security withholdings, or printing checks and/or statements. We definitely plan to add these features in the future, but they are well beyond the scope of what we can accomplish in ten weeks, and therefore are not primary goals for our current project.

We hope to compete with these products in a number of different ways. First, we intend to offer a product that is significantly cheaper than our market competitors. Even the cheapest

option, Intuit's Online Payroll Basic, costs about $4 per employee per month, whereas more expensive options can cost as much as $30 per employee per month. With the current cost of running servers and storing data (both extremely cheap), we could easily charge a small fraction for the same service. Current payroll software is vastly overcharging employers, and there is significant demand for a less expensive option. However, price is the secondary motivation for why our product could be superior to competitors.

Our product's primary differentiating factor relates to compliance with California labor laws. We spoke with an attorney who is an expert on California labor law, and she specifically noted that she isn't aware of a single payroll software provider that currently complies completely with all California payroll regulations. She also knows of many companies who use ADP that are actively searching for alternatives due to lawsuits caused as a result of ADP's failings. According to this expert, ADP does not have the infrastructure to store employee pay records back 4 years, and often miscalculates overtime pay due to the different way that California calculates overtime in comparison to the rest of the country. We plan to create a product that is completely compliant with all California laws from the start, which will in turn allow us to offer a product to companies that helps shelter them from legal liability.

Major Technologies Used:

Our project is a web-based application that will be constructed using the Django framework. Django is a Python-based free and open source web framework that empowers developers to more easily build powerful database-driven websites. Our choice to use Django is a consequence of both its popularity, and our team's familiarity with the Python programming language. Django also has a number of useful built-in security features (ie. CSRF protection), and employs an object-relational mapper (ORM) that makes database interactions substantially less complicated.

The pages that are served from our Django application will be styled using the Bootstrap CSS and JS framework. Bootstrap is an extremely popular front-end framework that is used for building cross-platform applications that are responsive and aesthetically powerful. Incorporating Bootstrap into our project will in turn make it substantially easier for our application to display information to users in a form that appears clean and professional, and will ensure site functionality on mobile devices.

Building this application will require collaboration between our different team members, which necessitates the use of a version control system. Version control systems allow developers to track the changes that are made by different collaborators on a project, and reverse certain changes where appropriate. We have chosen to use Git, an extremely popular open source revision control system that all of our team members are relatively familiar with. The Git repository that contains our application's code and assets will be stored on GitHub, which was chosen simply because it is the world's most popular web-based Git repository hosting service.

This Git repository will then be deployed to Heroku- the service that our team will be using to host our web application. Heroku is a cloud based server solution that allows developers to easily manage and scale deployments of web applications. Our team has chosen to use Heroku because it scales easily, offers free web application hosting initially, and operates seamlessly with Django applications. Furthermore, Django's ORM allows it to be database-agnostic, which means that using Heroku's SQL database offering (Heroku Postgres) only requires making minor modifications to our application's settings.

Resource Requirements:

N/A. Our application will be developed using freely available/open-source software and hosting services. For this reason, we will not need to acquire any special resources to develop this application. To be more specific, we plan to develop on a free Heroku server that will provide us with access to a Heroku Postgres database with the space we need.

Potential Approaches:

Our team's research into current payroll software offerings highlighted the need for a system that complied with California labor laws/regulations, and also allowed for easy and secure online viewing of pay statements. One potential approach that we considered was developing a payroll system that was tightly integrated with our own time-keeping software. This would entail developing a web interface with an authentication system that would enable employers to directly enter timekeeping data onto our site. However, we decided to forego this approach, since we wanted our payroll software to integrate easily with existing timekeeping software that was currently in use. Our choice to use a REST API thereby allows our platform to remain as flexible as possible, and also lets employers submit payroll information through automated means.

Assessment of Risks:

The nature our of our application's use of employee data entails a number of risks. First, security is an extremely important concern, since vulnerabilities in our application could allow unauthorized users to modify pay statements and gain access to confidential information. Our team plans to address this issue by testing for common vulnerabilities, and following best practices with regards to secure application design. Second, changing the schema of our database down the road to add additional functionality can be rather difficult. We plan to preempt this issue by designing our schema with the functionality that we intend to support in mind, and using Django's ORM to help handle any necessary modifications. Finally, site reliability is a crucial concern, since employers rely on our service for generating statements that will be used to pay employees. For this reason, we must thoroughly test our application to ensure that is both functional and robust. We also plan to take measures to ensure that our payment data can be

stored for an appropriate amount of time (in order to remain compliant with California payroll laws), and can be recovered in the event of a database crash/failure.

Next Steps:

Our first step will be to consult further with legal experts in order to outline the exact California regulations that our payroll software must comply with. This involves identifying the specific employee information that will be needed to generate pay statements, determining how to calculate wages in accordance with California law, etc. This information will allow us to better understand how to build our database's schema. It will also help us design the structure of our REST API, in a manner that allows our product to be incorporated easily into employers' existing payroll processes. We also intend to speak with different businesses in order to identify other problems that they experience with their current payroll software providers. These various steps will thereby allow us to begin building a novel payroll product that uniquely complies with California employment legislation, and offers various competitive advantages that allow it to outperform other market alternatives.