Laboratory of Web Application Development
Lecturer: MSc Alfredo Salazar Vélez

Team Project Specification Document

I. Introduction

This document includes the specification of the project developed by Pythoneers for MA Stucco. inc.. All the agreements and specifications are presented in this document, which both Pythoneers, represented by Alejandro Zamudio and MA Stucco, inc, represented by Brenda Alanis, from now on known as The Client, commits to.

This document is intended to serve as a guide both to The Client and to Pythoneers by describing the objectives of the project, the web application, and the impacts of its implementation on the target organization. Also, the document represents a formal requirements specification, which both Pythoneers and The Client agreed that the project will be developed by Pythoneers from January to May 2017. It also serves as an implementation manual for Pythoneers to follow without further clarification. The document clearly states the accorded requirements with The Client, among other things, so as to help Pythoneers in the process of implementation. Confidentiality of this document is extended to Brenda Alanis, Pythoneers and Alfredo Salazar, from now on known as The Teacher.

The web application hereby described has the sole purpose of supporting the everyday operational tasks of The Client's organization, as well as serving as an organizational tool for the administrative party within such organization. The development of the website intends to serve as a tool, that when used, increases the efficiency of the operational tasks and reduces money loses within the organization.

II. Project Objectives

The project at hand has several expected outcomes, one of which is the web application described in this document. It is the project's main objective to increase the efficiency of all the operational-level tasks and to improve the organization and assignment of them. It is also expected that the project will produce an improvement in the organizational department within The Client's company. These objectives are expected to be achieved through the development of a tool that enables these improvements. Such a tool is the web application in term.

The project should and will deliver such web application, and the continuous use of it will bring, in time (mid-term), a decrease in money loses and an increase in productivity.

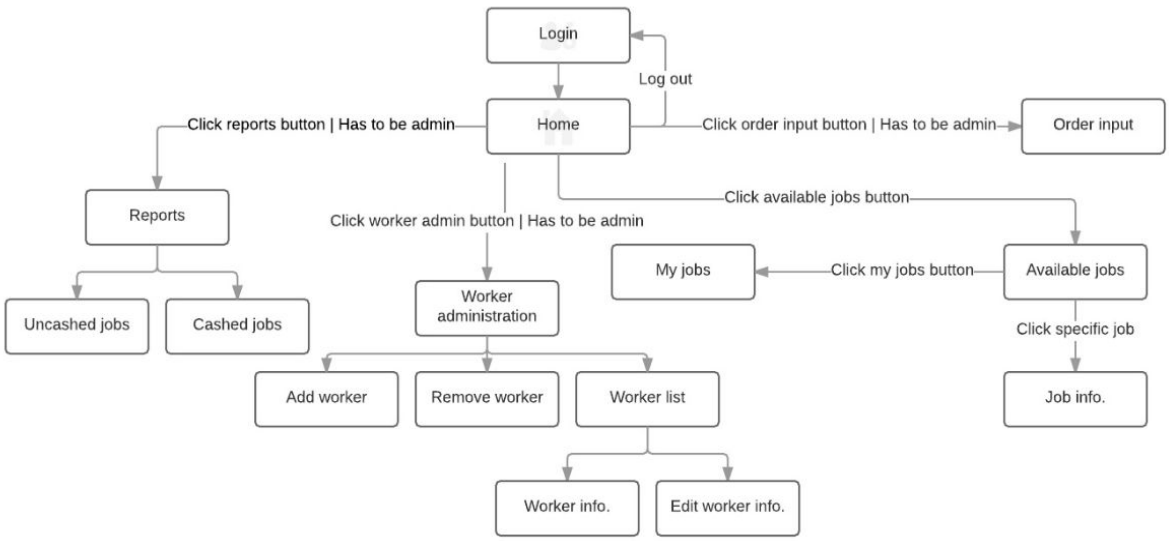
III. Success Criteria

The success of this project would increase, in a greater way, the accuracy of the production line; while minimizing delays and decreasing waste, incomplete orders and dead times at installations. This would also improve customer service, reliability of the company and an efficient way to produce the product.

Laboratory of Web Application Development
Lecturer: MSc Alfredo Salazar Vélez

Team Project Specification Document

IV. Site Map



IV.I Content of the web application

Login: user will have to login in order to use the web application. Users can not sign up on their own, they have to be manually registered by an administrative user.

Home page: the home page will have 4 buttons order input, reports, worker administration and available jobs.

- Order input will take you to the order input page, only the administrator will be able to access this page.
- Reports : will take you to report page, only the administrator will be able to access this page.
- Worker administration: will take you to worker administration page, only the administrator will be able to access this page.
- Available jobs: depending on the worker and user profile they will be able to see which job is available and has not been done, while the admin can see all the still pending jobs.

Order input: it will take you to a page where the administrator will fill out a form with the

specifications about the job.

Reports: reports page will have show a summary of the jobs that have been done, it will show uncashed jobs and cashed jobs. There will be a filter option available for searching specific dates or specific jobs.

- Uncashed jobs: it's the summary of the jobs that have not yet been cashed, once they are cashed the administrator will click a button and move it to cashed jobs.
- Cashed jobs: summary of all the jobs that have been cashed in.

Worker administration: the administrator will be able to edit worker's info.

- Add worker: where the administrator manually adds a worker.
- Worker list: shows a list of all the workers.
    - Worker info: shows all the information that is required from the workers.
        - Delete worker: administrator can erase a worker.
        - Edit worker info: administrator can edit a worker's info.

Available jobs: it's the page that will show all the available jobs, for the administrator the will able to see all the jobs still pending to be done. While the workers will only be able to see the jobs they can work on. There are different types of jobs there are 2 main processes that are foam and cast they both have a different process, the workers that work on foam have to cut the foam, spray the foam and install the foam, the workers will be able to see the available jobs and accept the jobs and when they are done it will have to report it in the web page as done, same for the cast jobs.

- My jobs: will show all available jobs and the jobs that the worker is currently working on, when a worker accepts a job then it will no longer be on available jobs and until the work is finished and reported by the worker.
- Job info: shows all the information about the job and has a button for the worker to accept the job.

V. Functional Specification

- When a user first tries to access the web application, he/she is presented with the login page. In this page the user must input the required credentials to be able to access the web application. Only administrative users have a registered account by default, all the other users (workers) have to be manually registered by these administrative users.
- Once a user has entered the required credentials and they're validated and he/she is authenticated, the user is presented with the home page of the web app. This page contains a set of buttons that lead to other pages, but the ones that are shown depends on the type of user that has accessed the web app. The overall set of buttons are as follows: Order input button, Reports button, Worker administration button, and Available jobs button.
- The Order input button takes the user (it has to be an administrative one) to the order input page. This page presents a form than enables the user to input the information of a certain new order, and then the form is submitted and processed for back-end storage.
- The Reports button takes the user (it has to be an administrative one) to the reports page. This page presents a list of all finished jobs in the organization, and can be filtered to show only the ones that are already cashed in or the ones that haven't been cashed. It also includes a search bar that enables the user to search specific jobs in this pool of finished jobs.

Team Project Specification Document

- The Worker administration button takes the user (it has to be an administrative one) to the worker administration page. This page presents a list of all current workers in the organization and a button that enables the user to add a new worker via a form. If the clicks a specific worker in the list, the web app presents the user with a new page that contains the information of the selected worker, and shows two buttons, one that allows the user to edit the worker's information via a form, and the other one that enables the user to delete the worker from the system. The only way to delete a worker is if such worker doesn't have assigned jobs in that specific moment.
- The Available jobs button takes the user (it works both for administrative users and normal users) to the available jobs page. This page basically shows a list of all the available jobs at the current time, but it works contextually. The list contains certain jobs depending on the user that has been authenticated. For example, if an administrative user is authenticated, the page will show a list of all the current jobs in the organization, but if a normal user (worker) is authenticated, the page will show a list of available jobs, if any, that fall into such worker's specialty. This means that certain users (workers) will be able to see only certain jobs. If the user selects a job in the list, the information of such job will be displayed in a new page and if the job hasn't been taken, the user (worker) can accept the job. Once the a job has been accepted by a user (worker), it will not appear anymore in the available job list, but it will appear in the My jobs list of the user that accepted it. The My jobs is a button that takes the user (worker) to another page that displays a list of all the current jobs that the user (worker) has accepted and is working on so far.

VI. Technical Specification

- All the pages and functionality of the web app are described in the Technical Specification section. This section will only discuss the technical aspects of the web application in term.
- The technology that's going to be used in the development of the web app is Django both for front-end and back-end.
- Django is based on python and a specific template language defined by Django, so these are the scripting languages that are going to be used.
- For front-end, besides the template language, some HTML and CSS will be used, along side with frameworks like Bootstrap.
- For back-end, besides python, SQLite3 is going to be used for the database implementation and a specific Django framework will be used for the definition of the schema. This definition is still based on python.
- All the control of versions of the web app will be managed using git through Github.
- The web app directory structure is as follows:
    - MyWebApp
        - templates
        - WebApp
            - migrations
            - static
            - templatetags
            - __init__.py
            - admin.py
            - apps.py
            - models.py
            - tests.py
            - views.py

Laboratory of Web Application Development
Lecturer: MSc Alfredo Salazar Vélez

Team Project Specification Document

- MyWebApp
  - __init__.py
  - settings.py
  - urls.py
  - wsgi.py
- manager.py
- db.sqlite3

VII. Content Plan

VII.I Version Control

The software for version control that's going to be used is Git, supported by the website Github to establish a remote repository for every team member to access and update accordingly. This remote repository in Github is going to be a private one so as to protect the software designed and developed for The Client.

VII. II Responsibilities

Alejandro Zamudio is the leader of Pythoneers and will also function as project manager, he will be in charge of making sure development meets the requirements and specifications, in addition to participating in product development.

Melissa Figueroa is the main contact with The Client and will be in charge of keeping abreast of the two parts of the work that is performed besides participating in the development of the product.

Jorge Márquez and Evert Salinas will focus on product development and product design.

VII.III Timelines and Milestones

The development team will hold weekly meetings to see product progress and determine next short-term goals. Meetings with the client were planned if necessary at the time required.

Sprint 1 period: February 6th - February 20th
Sprint 1- 1.1: Create design and styling guide
Sprint 1- 1.2: Create ER diagram
Sprint 1- 1.3: Develop Django database models
Sprint 1- 1.4: Plan automated testing suite for each component
Sprint 1- 1.5: Plan testing suit for client system testing

Sprint 2 period: March 21st - March 7th
Sprint 2- 2.1: Enhance Django's default User model
Sprint 2- 2.2: Develop Login Django view (component)

Team Project Specification Document

Sprint 2- 2.3: Develop Login Django template
Sprint 2- 2.4: Develop Home Django view (component)
Sprint 2- 2.5: Develop Home Django template
Sprint 2- 2.6: Develop Reports Django view (component)
Sprint 2- 2.7: Develop Reports Django template

Sprint 3 period: March 8th - March 29

Sprint 3- 3.1: Develop Worker Administration Django view (component)
Sprint 3- 3.2: Develop Worker Administration Django template
Sprint 3- 3.3: Develop Order Input Django view (component)
Sprint 3- 3.4: Develop Order Input Django template
Sprint 3- 3.5: Develop Available Jobs Django view (component)
Sprint 3- 3.6: Develop Available Jobs Django template

Sprint 4 period: March 30th - May 3rd

Sprint 4- 4.1: Develop concurrency handling mechanism for job acceptance and availability
Sprint 4- 4.2: Develop automated testing suite for each component
Sprint 4- 4.3: Develop testing suit for client system testing

VIII. Testing Plan

The testing plan will be divided in two phases:

- Whitebox testing: This phase will test the functionality of every developed component, including alternate routes of execution, depending on the kind of user doing the testing (worker user or admin user). This tests are part of an automated testing suite developed in Django.
    - Login (any user): This test is focused on authentication of users and non-users. The test will try to login using random non-existent credentials and also with credentials of a current user. The former should not proceed to the Home page and the latter should.
    - Home (any user): This test is focused on authorization of users to access certain functionalities of the web app. The test will try to access different pages going from the Home page while using both a worker user and an admin user. The admin user attempt should allow the test to go anywhere while the worker user one should allow the test only to access the Available Jobs page.
    - Order Input (admin user): This test is focused on the creation of new orders/jobs. The test will try to create a new job using wrong input and another using correct input. The former should not allow the test to create the job and the latter should.
    - Reports (admin user): This test is focused on the validation of the functionality of the reports list. It tests that the reports list indeed shows uncashed jobs and cashed ones when prompted. The test also validates the functionality of the search bar.
    - Worker Administration (admin user): This test includes a set of 5 subtests from the same context.
        - Add worker (admin user): This test is focused on the creation of new worker users. The test will try to create a new worker user using wrong input and another using correct input. The former should not allow the test to create the user and the latter should.
        - Remove worker (admin user): This test is focused on the deletion of a current worker user. The test will try to delete a user that is assigned to a job and a

user that has no job assigned. The former should not allow the test to delete the user and the latter should.

- Worker list (admin user): This test is focused on the validation of the functionality of the worker list. It tests that the worker list indeed shows all current workers. It also tests that when a specific worker is clicked in the list, a new page with the worker's info should be displayed.
- Worker info (admin user): This test is focused on the validation of the functionality of the Worker Info page. It tests that this page shows indeed the correct info of the specified worker.
- Edit worker info (admin user): This test is focused on the validation of the functionality of the edit worker info page. The test will try to edit a worker's info poorly and then correctly. The former should not allow the test to edit the worker's info and the latter should. It will also test that the changes to the info indeed took place.

- Available Jobs (any user): This test is focused on the validation of the functionality of the Available Jobs page. It tests that this page shows indeed the correct list of available jobs depending on the type of user and the status of such. The admin users should be able to see all current jobs. The worker users should be able to see only the jobs that fall into their specified status/category. This test also includes two subtests.
  - My jobs (worker user): This test is focused on the validation of the functionality of the My Jobs page. It tests that this page indeed shows the correct list of jobs that are assigned to the authenticated worker user.
  - Job Info (any user): This test is focused on the validation of the functionality of the Job Info page. It tests that this page indeed shows the correct info of the specified job. It also tests that if a worker user accepts/takes a job, that if this job at the moment is unassigned, then the job is assigned to the authenticated user.

- Blackbox testing: This phase will test the functionality of the system from the perspective of the user. A series of incremental prototypes are going to be given to The Client for testing purposes, and The Client will use the web app both with admin users and worker users and will report their findings for further development. This testing will occur beginning with the completion of sprint 2.

IX. Appendix

Meeting Notes

18 January 2017

With The Client:

- The client explained the processes that they use at the company; they are Foam cutting and installing and also Cast moulding and installing.
- It's important that they are able to see in the report which worker did the cutting and which did the installation.
- The Client mentioned that it is important that the web application can be accessed by mobile.
- The Client also mentioned that filtering the report by date or other attributes is

Team Project Specification Document

necessary.

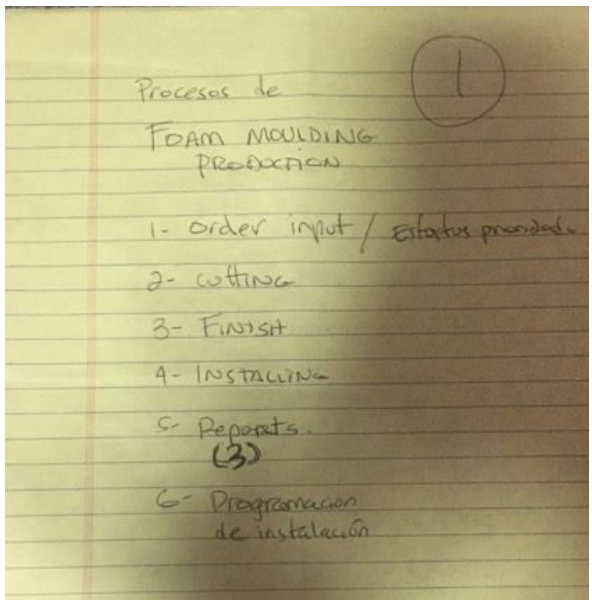
- The Client mentioned that it would be a good addition if the reports can be downloaded as an Excel document.

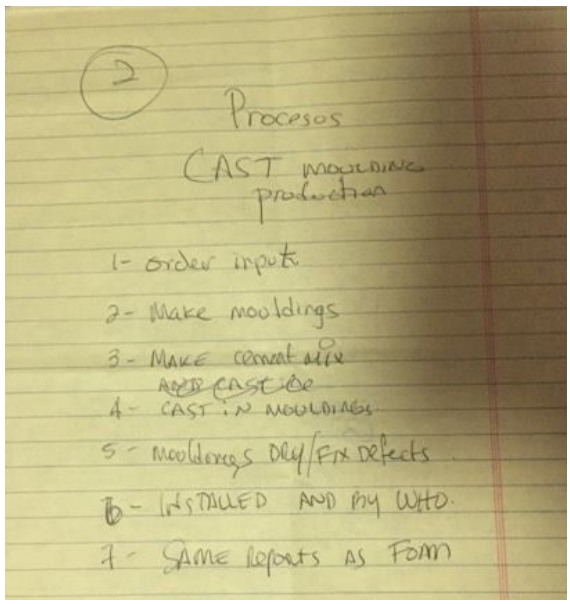
Action items:

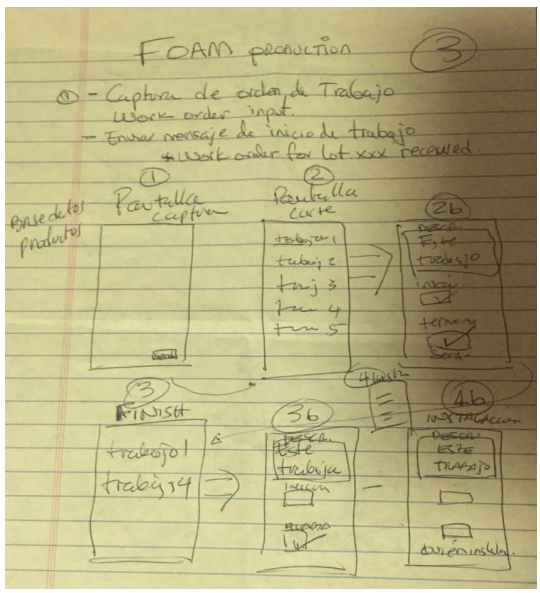
- Create a facebook group so we can talk.
- Work on the SRS depending on your assignment.

Notes

Some pictures of some of the notes made by The Client when they were explaining the project.

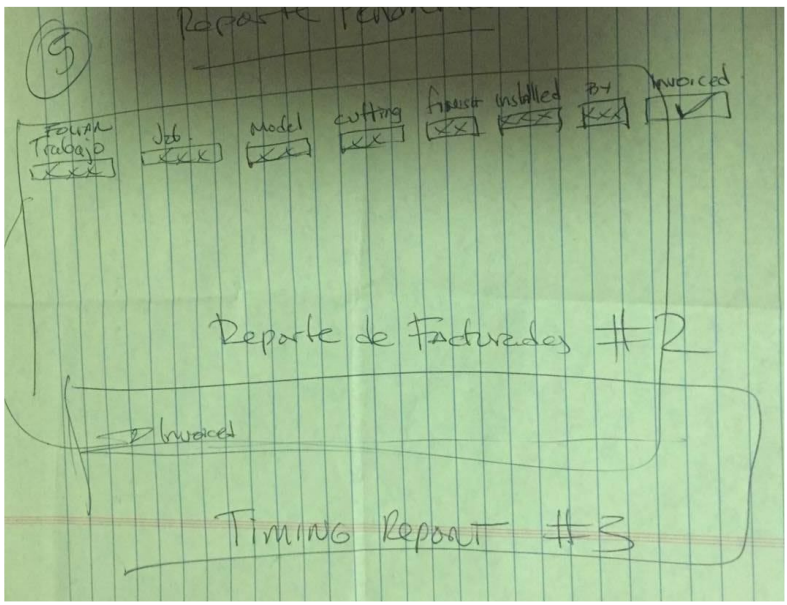






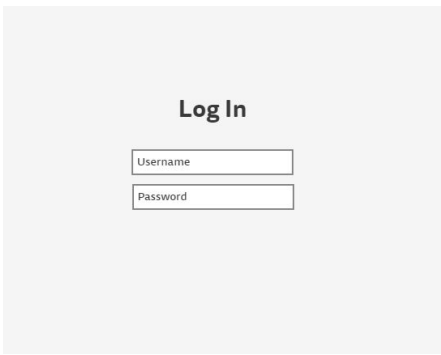
Laboratory of Web Application Development
Lecturer: MSc Alfredo Salazar Vélez

Team Project Specification Document



Design Concepts

Login



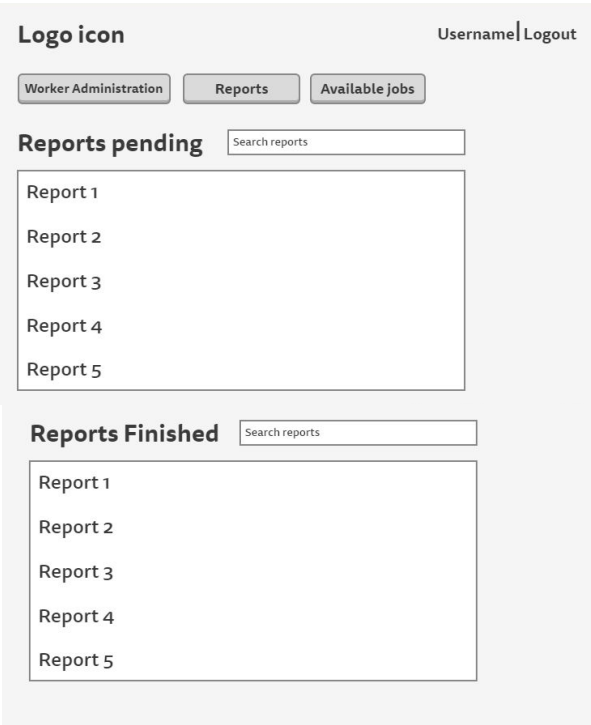
Home(admin)

Laboratory of Web Application Development
Lecturer: MSc Alfredo Salazar Vélez

Team Project Specification Document

Home(worker)

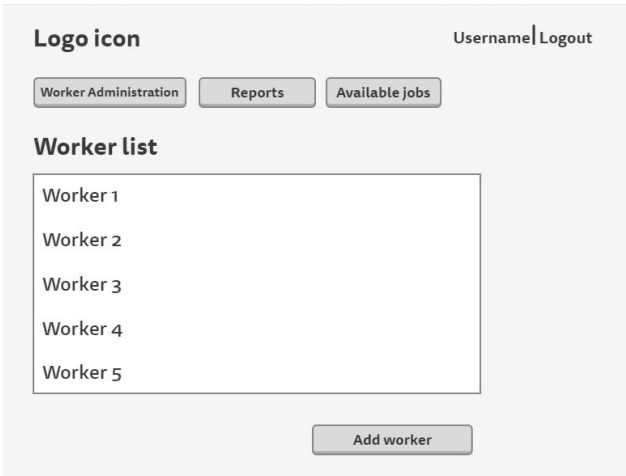
Reports



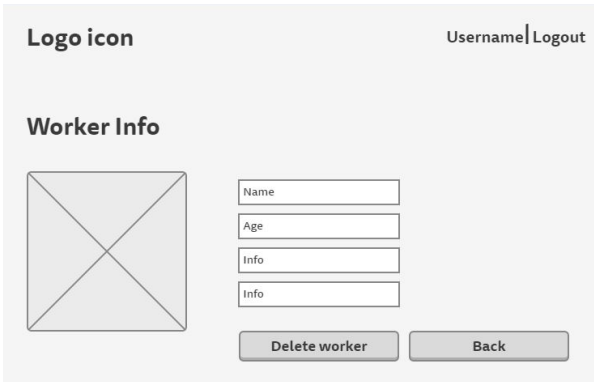
Laboratory of Web Application Development
Lecturer: MSc Alfredo Salazar Vélez

Team Project Specification Document

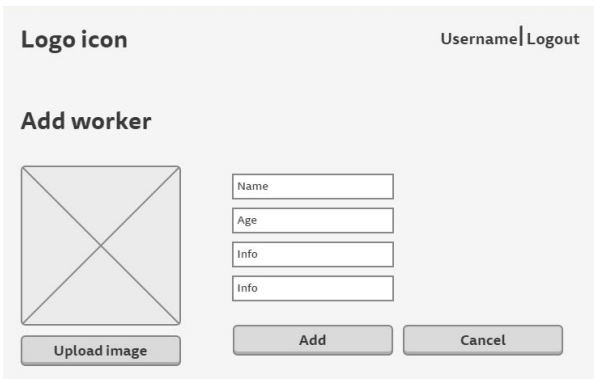
Worker Administration



Worker info



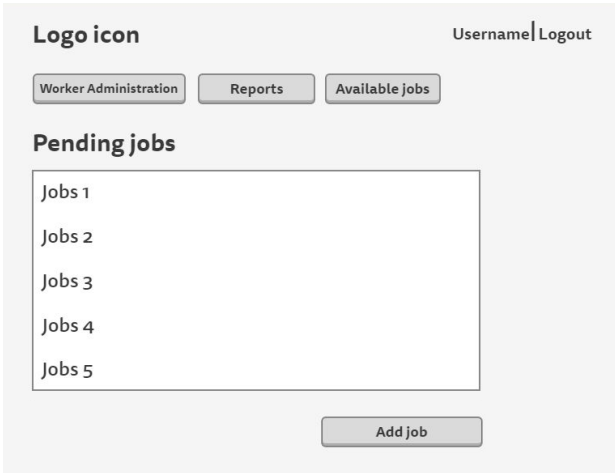
Add worker



Laboratory of Web Application Development
Lecturer: MSc Alfredo Salazar Vélez

Team Project Specification Document

Jobs(admin)



Jobs(workers)

Job info

Laboratory of Web Application Development
Lecturer: MSc Alfredo Salazar Vélez

Team Project Specification Document

Prototyping

There's going to be 3 prototypes in total, each one presented after a specific sprint has terminated.

- Prototype 1: This prototype is the result of sprint 2 and contains full functionality for the Login, Home and Reports components.
- Prototype 2: This prototype is the result of sprint 3 and contains full functionality for Worker Administration, Order Input and Available Jobs components.
- Prototype 3: This prototype is the result of sprint 4 and contains full functionality for all components and the concurrency handling mechanisms.

Assumptions

NA

Glossary of Terms

NA

Laboratory of Web Application Development
Lecturer: MSc Alfredo Salazar Vélez

Team Project Specification Document

X. Client and Team Leader Signature Page

With the signature of this document, Pythoneers commits to maintain a strict confidentiality of the information received by The Client and to use it solely for the purpose that is established in this document. Likewise, Pythoneers are responsible to give back and document, key, or account that is given to them during the development of the project.

The Client is responsible to give any information agreed above in this document, necessary for the successful completion of this project. Also, the client commits to follow up all the process for the development of the project and be available for any given question or clarification from the team.