**Group 8 (Software Sharks)**

Requirements Analysis

**-- PROBLEM --**

During the last few years, multiple members of our group have experienced poor scheduling techniques used for our various places of employment. It is a rather daunting task to balance everything that pertains to each employee such as availability and requesting off in order to formulate an accurate schedule. Additionally, to add an extra headache to the scheduling manager, they also have to create multiple schedules for various positions that are held. Managers already have enough work to attend to with daily problems that occur at the workplace, so we believe we can alleviate some of that stress with the implementation of Schedule Shark!

**-- REQUIREMENTS ANALYSIS --**

**User Requirements**

**Primary: Kaitlin Anderson**

**Secondary: Jeremy Warden**

* User Login
  + Employee
    - Server
    - Bartender
    - Busser
    - Food-Runner
    - Cashier
    - Hostess/Host
    - Supervisor
    - Manager
* Give Availability
  + Employee
    - Days of availability
    - Time of availability
* Request Time off
  + Employee
* Accept Request off
  + Manager
* View Schedule
* Edit Schedule
  + Manager
* Contact other employees
  + Similar employees & Managers
* Registration approval
  + Manager

**System Requirements**

**Primary: Jeremy Warden**

**Secondary: Kaitlin Anderson**

* **LAMP STACK (STORED ON AZURE)**
  + Linux
    - Virtual Machine is powered by Linux, creating a safe environment for us to utilize the resources necessary to run our application
  + Apache
    - Web server where we will be hosting our web application
  + MySQL Database
    - Our web based application will be database driven, using user data in order to function properly
  + Python/PHP
    - We will be communicating between our controller and our model with a server side scripting language

**Functional Requirements**

**Primary: Han Chen**

**Secondary: Josh Lewis**

* **User Login**
  + On correct input, advances user to site.
  + On incorrect input, allows user to try again or change password.
* **Give Availability**
  + Store Employees availability
  + Use information for generating Schedule
  + Edit availability
* **Request Time off**
  + Send request off dates to manager in order for approval
  + Store date on approval
  + Use information for generating schedule
* **View Schedule**
  + Each type of employee will have ability to view the corresponding schedule
* **Edit Schedule**
  + Managers should be able to make changes to the schedule
* **Contact Other Employees**
  + Managers should be able to mass e-mail employees.
  + Employees should be able to contact similar employees, as well as their manager.

**Non-Functional Requirements**

**Primary: Josh Lewis**

**Secondary: Han Chen**

* The program should be stable, it should have an incredibly low failure rate.
* The program should be fast and efficient, responding in under a minute to queries and requests.
* The program should recover gracefully from incorrect inputs and from system outages.
* Database should be able to handle large amount of data and simultaneous requests.
* System should be secure, not just anybody can register for an account in the system, you must be invited.
* Should work on multiple web platforms including, IOS and Android web browsers.