**Software Sharks (Group 8)**

**Schedule Shark**

**Requirements Analysis**

**Authors:**

Kaitlin Anderson

Jeremy Warden

Josh Lewis

Han Chen

03/16/2016

**Version: 0.01**

**Table of Contents**

The Problem……………………………………………………1

User and System Requirements………………………………...4

Functional and Non-Functional Requirements…………………6

Use Case Diagram……………………………………………...8

Activity Diagram……………………………………………….9

ERD…………………………………………………………...10

Glossary…..…………………………………………………..11

**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.

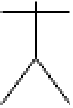
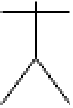
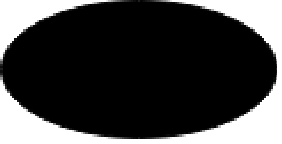
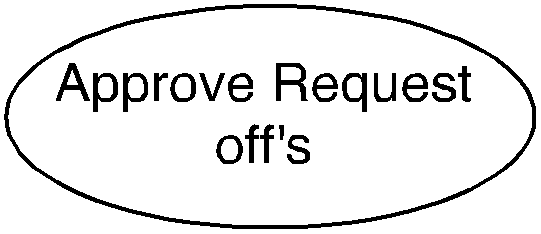
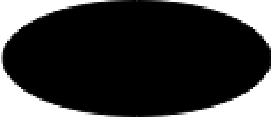
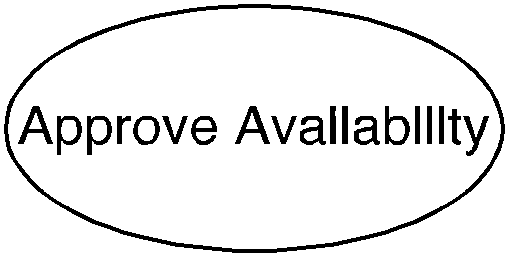
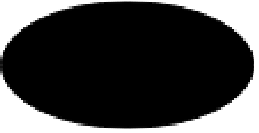
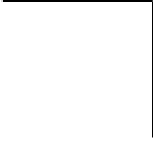
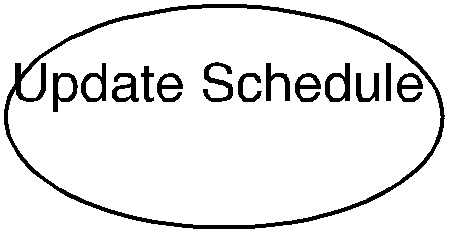
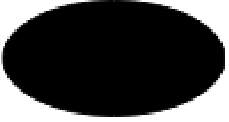
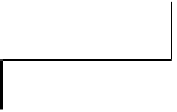
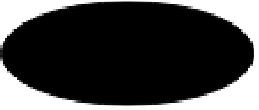
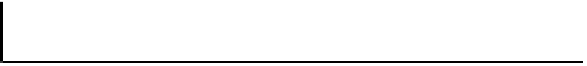
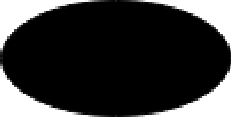
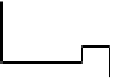
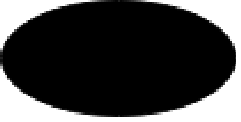
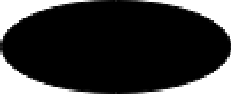
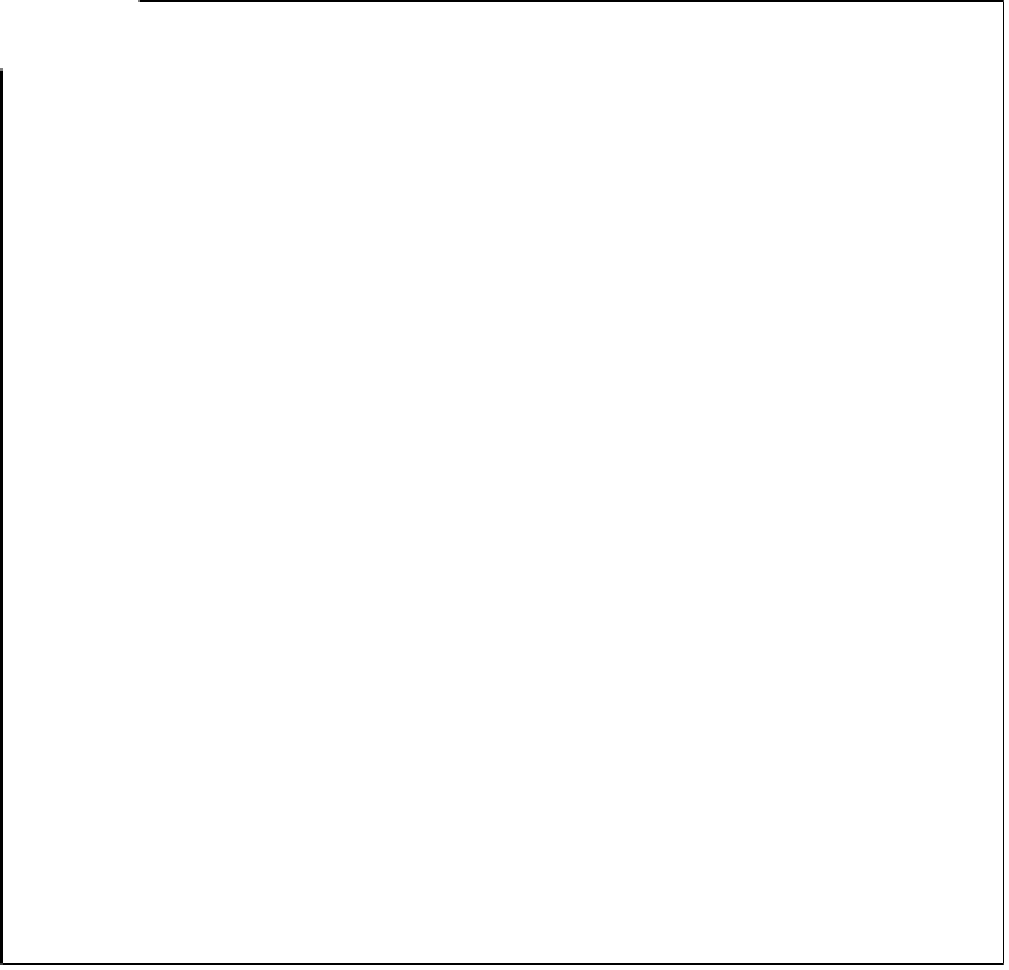
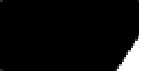
**Non-Functional Requirements**

**Primary: Josh Lewis**

**Secondary: Han Chen**

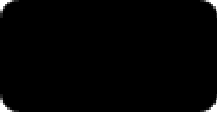
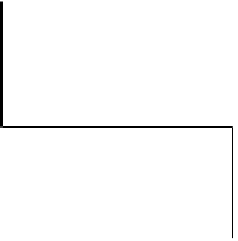
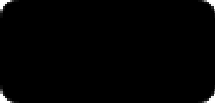
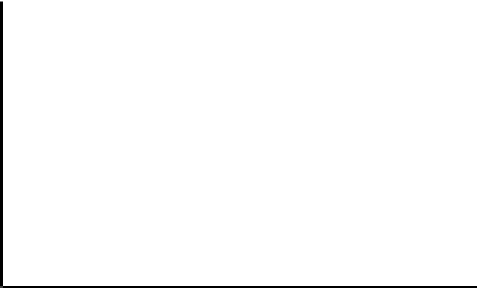
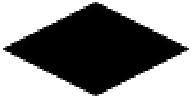
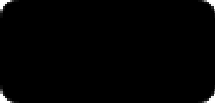
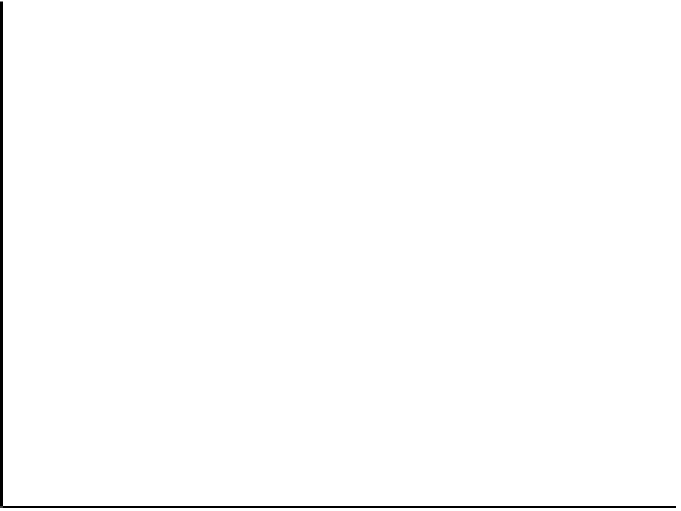
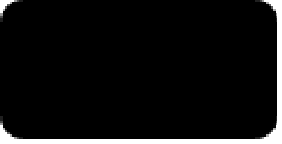
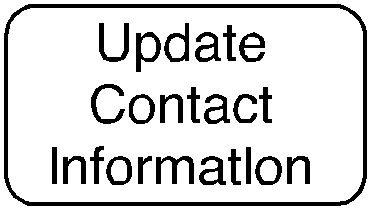
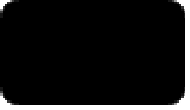
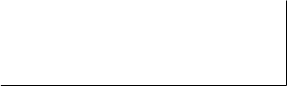
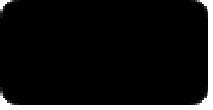
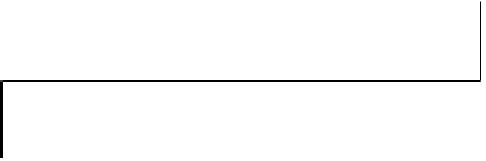
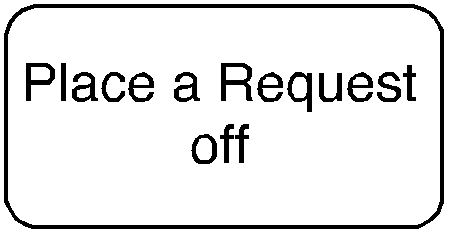
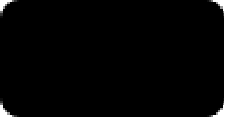
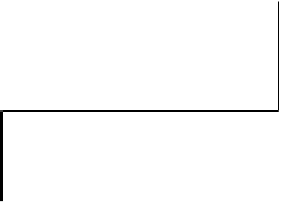
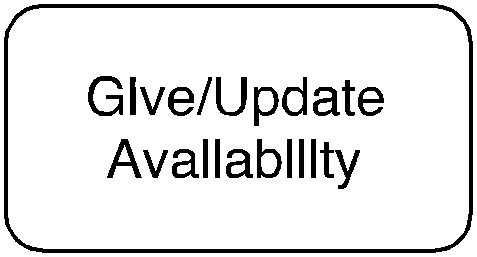
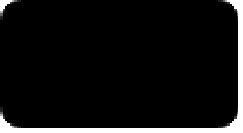
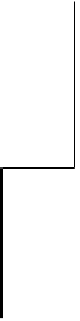
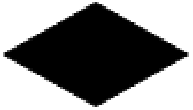
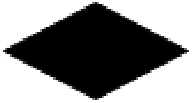
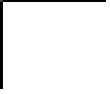
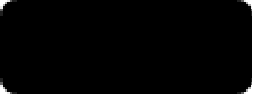
* The program should be stable, it should have an incredibly low failure rate.
* The program should be fast and efficient, responding in under a minute to queries and requests.
* The program should recover gracefully from incorrect inputs and from system outages.
* Database should be able to handle large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

Use Case Diagram



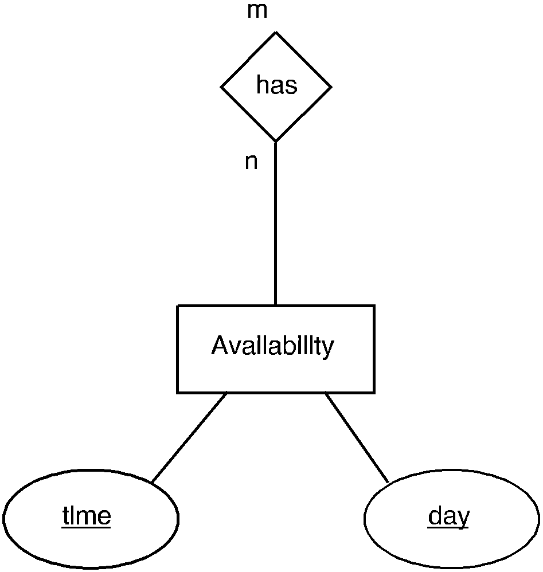
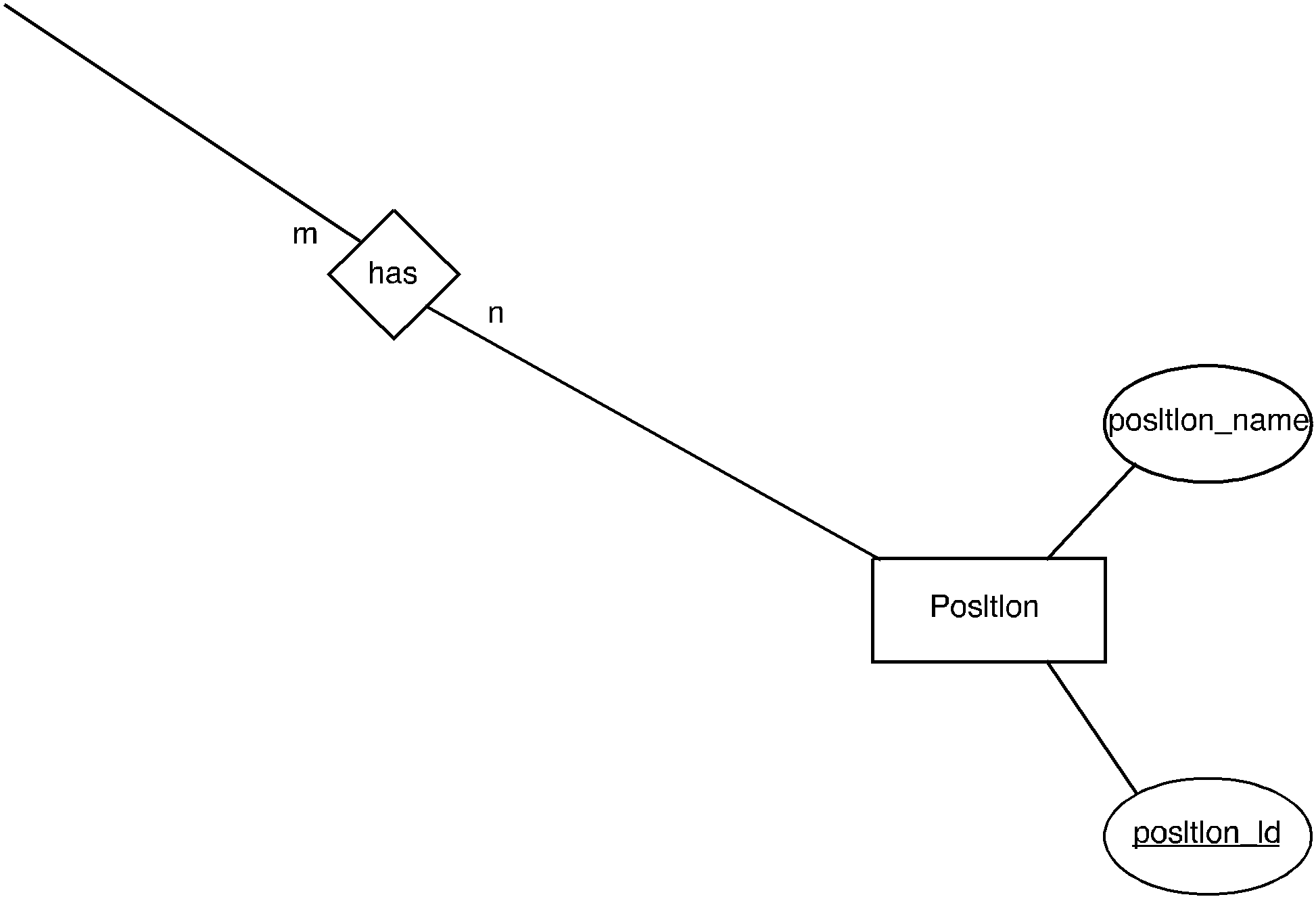
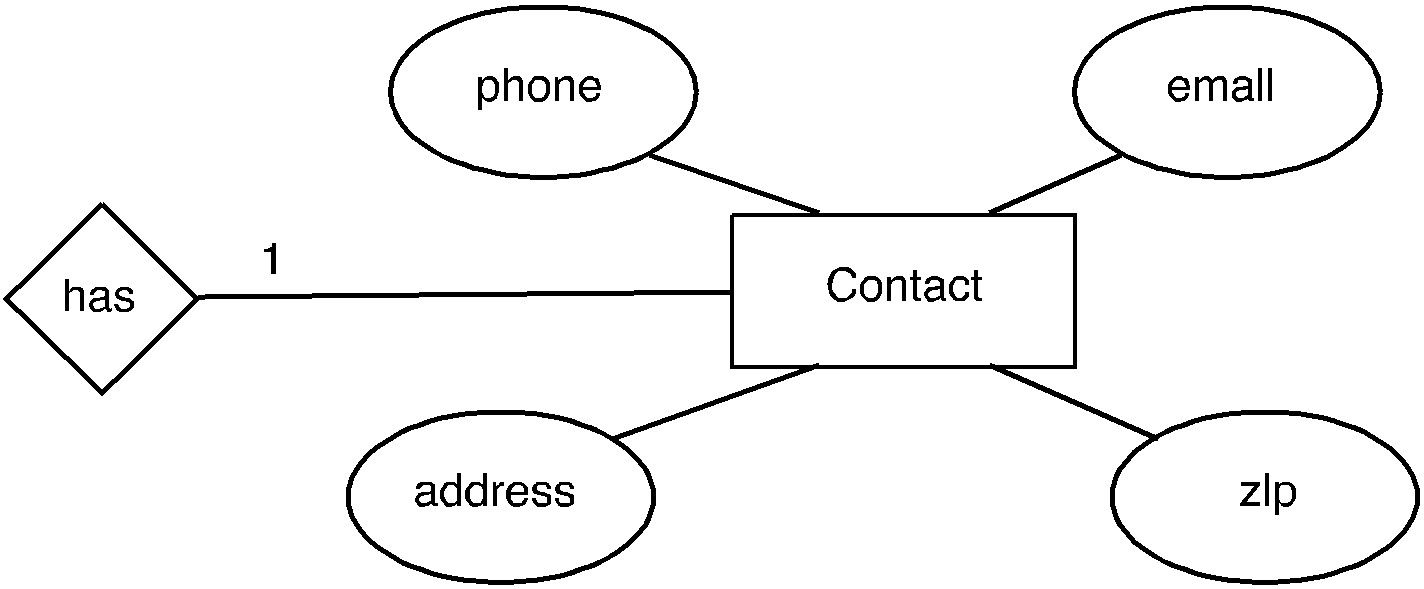
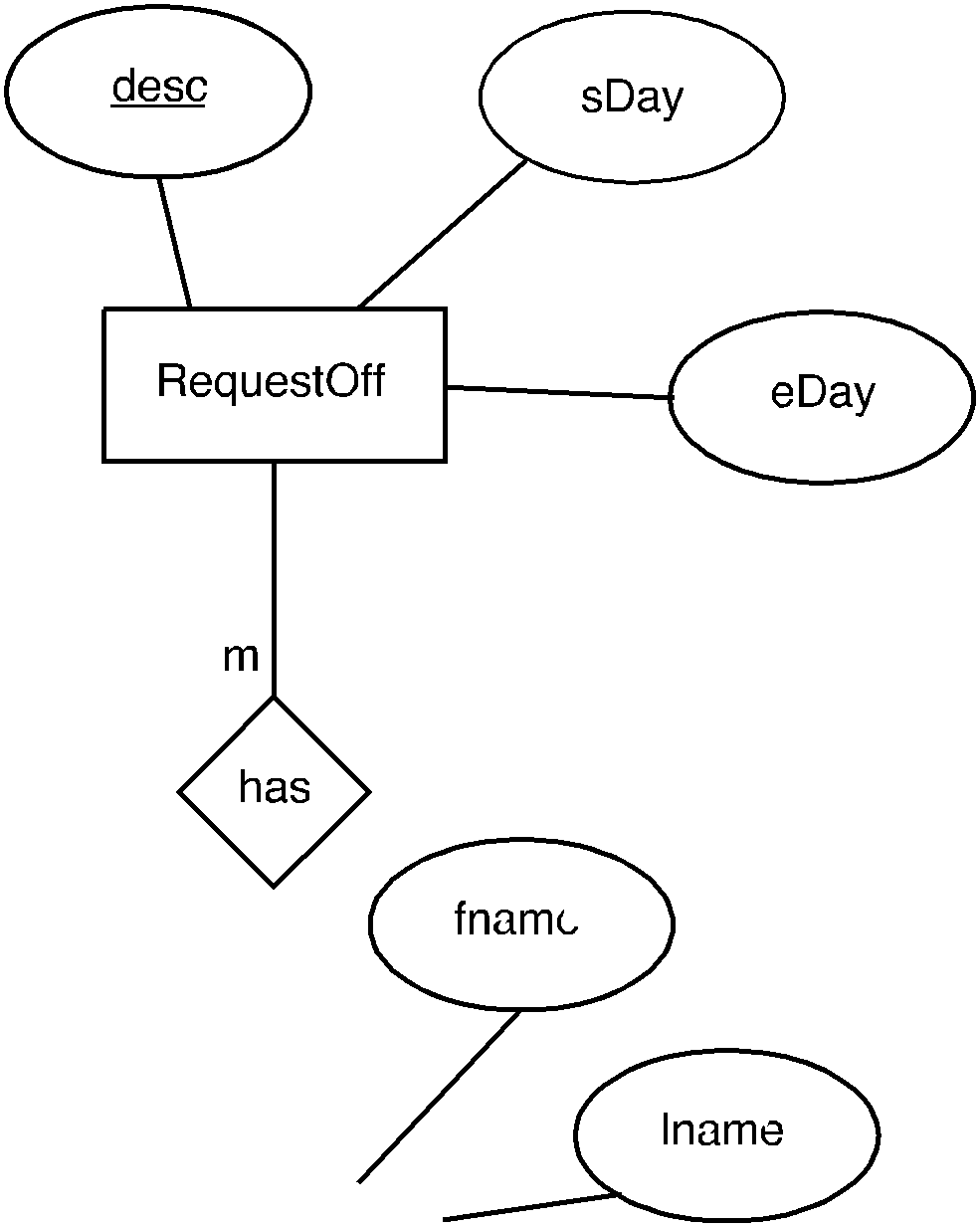
Employee

Manager



Activity Diagram

ERD



ChangeLog

|  |  |  |  |
| --- | --- | --- | --- |
| # | Date | By | Description |
| 01 | 03/10/2016 | All | Sprint one meeting: decide how tasks are divvied up |
| 02 | 03/15/2016 | Kaitlin Anderson &  Jeremy Warden | Create user requirements |
| 03 | 03/15/2016 | Jeremy Warden &  Kaitlin Anderson | Create system requirements |
| 04 | 03/15/2016 | Josh Lewis &  Han Chen | Create functional requirements |
| 05 | 03/15/2016 | Han Chen &  Josh Lewis | Create non-functional requirements |
| 06 | 03/15/2016 | Jeremy Warden | Cerate DDL, User Case |
| 07 | 03/15/2016 | Kaitlin Anderson | Create ERD |
| 08 | 03/17/2016 | Josh Lewis | Integrate the documents and diagrams, create table of contents |
| 09 | 03/17/2016 | Han Chen | Create change log and glossary |

**-- GLOSSARY –**

**Schedule**

A list of employees, and associated information, for example, position, working time, responsibilities for a given time period.

**User Requirements**

What the users expect the software to be able to do. The user requirements can be used as a guide to planning cost, timetables, milestones, testing, etc.

**System Requirements**

In order to work efficiently, all computer software needs certain hardware components or other software resources to be present on a computer. These prerequisites are known as system requirements and are often used as a guideline as opposed to an absolute rule.

**Functional Requirements**

It essentially specifies what the system should do. It specifies a behavior or function, for example, display the name, available time and edit the employees’ information, etc.

**Non-functional Requirements**

It essentially specifies how the system should behave and that it is a constraint upon the systems behavior. One could also think of non-functional requirements as quality attributes for of a system.