ReShift

by Sheldon

Project 3 Phase 1
Individual Design Document

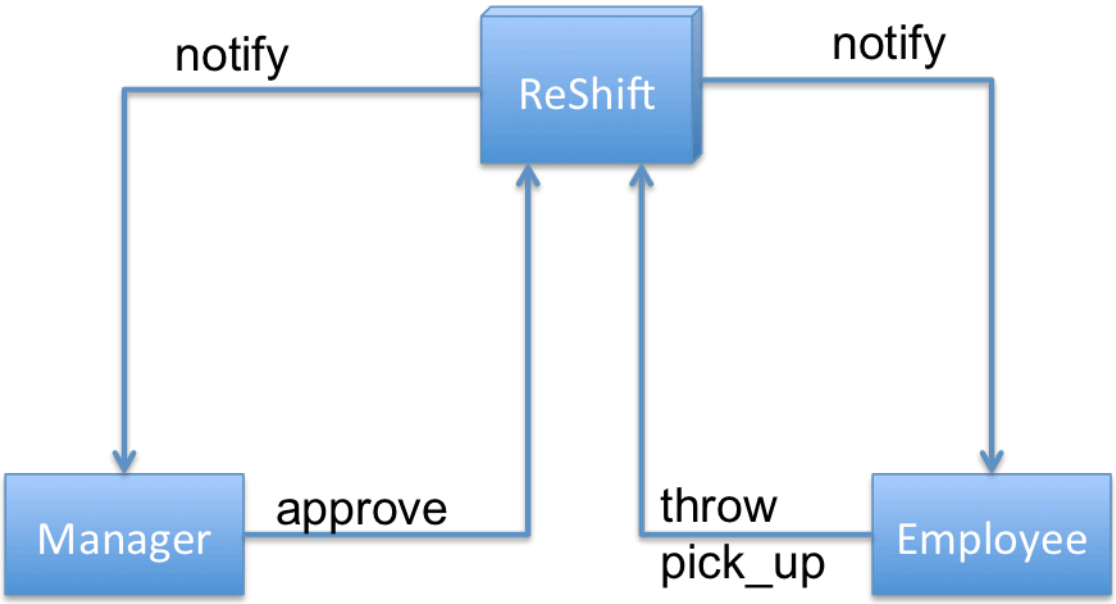
I. Motivation:

ReShift is a web application that helps employees in a shift-based job better organize, pick up and exchange shifts. This app also enables managers of these shift-based jobs to not have to worry about the issues behind their employees not being available for their shift. In many large-scale companies that have a high shift turnover rate, shifts are usually exchanged by a phone call or email to a manager, who then finds some one to replace the absent employee. There is no efficient solution to how this exchange is handled. ReShift can change that.

Key Purposes:

- **Organize all of the information pertaining to a shift based job is in one place.** The big issue for a number of shift-based jobs is the unorganized nature behind exchanging.
- **Centralize the exchange of shifts to one place.** This is *the* hub of all shift exchange.
- **Efficiently display picked up and thrown shifts.** Paper schedules and phone calls to managers are a thing of the past.

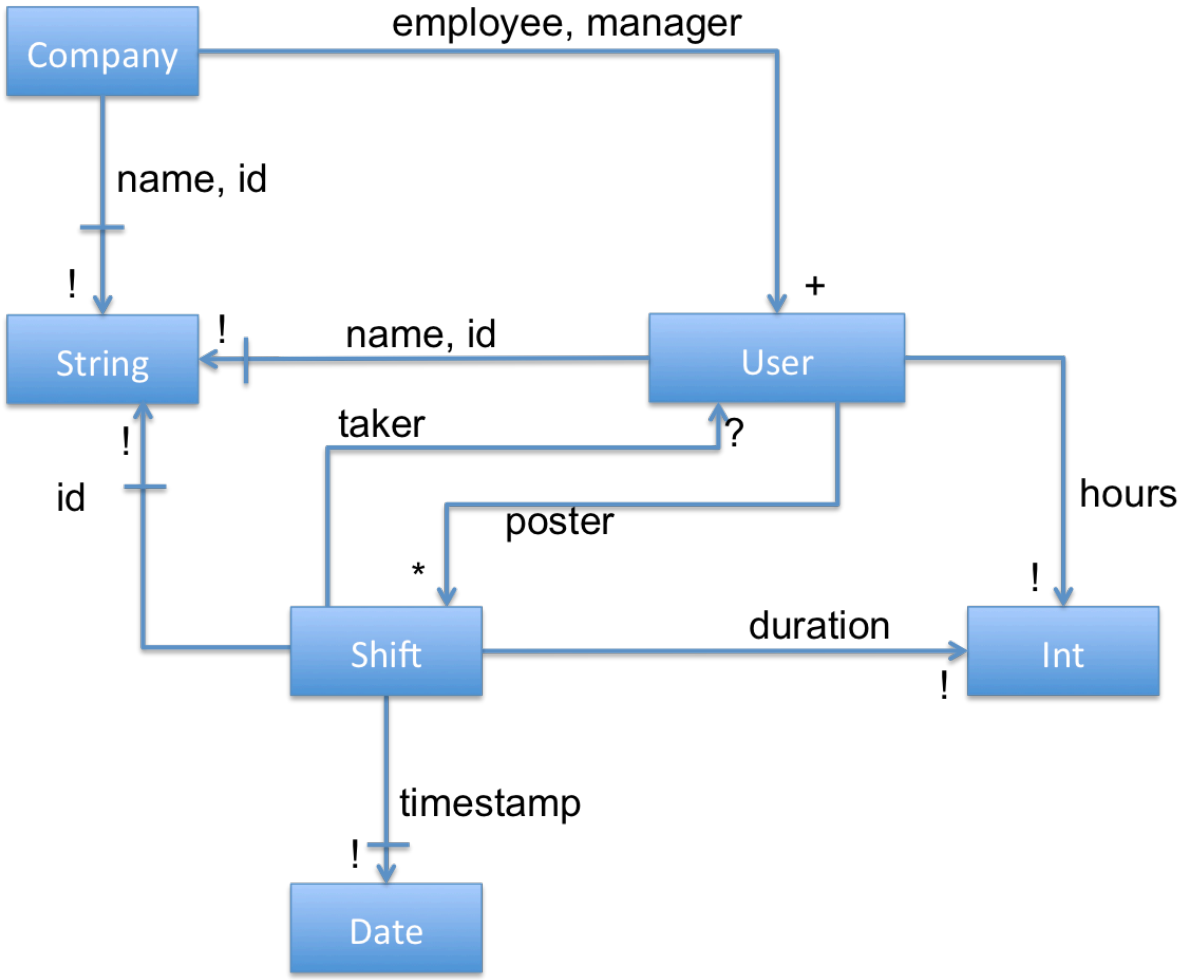
II. Context Diagram:



III. Concepts:

- **Shift** – This is the set start time and duration for a slot of time an employee can work.
- **Throwing** – Users that cannot work the specified shift that they are supposed to have the option to "throw" the shift. This notifies all other employees that the shift is now available to be covered by someone else.
- **Pick-Up** – In contrast to "throw", "pick-up" allows the user to claim the given shift. Users can claim as many shifts as are available.
- **Employee** – The main user of ReShift. "Employees" have the ability to view the times that they work and can decide – if the shift is available – to pick up the shift. They also have the ability to give up their shifts "throw" their shifts.
- **Manager** – A secondary user of ReShift. "Manager" has the ability to add new shifts and assign which employees work the given shift. They also approve of the hours and can assign pay when the week is over.
- **Company** – Within ReShift a "company" is defined as the set of employees that belong to the same time schedule. The employees within this "company" all have access to view who is working when and can "pick-up" any available shift that is "thrown" by another employee. The idea of "company" restricts employees of different companies to pick up and throw shifts that do not pertain.

IV. Data Model:



V. Design Challenges:

How do employees pick up shifts? This is a problem because this raises the issue of how is the representation of a shift being claimed done.

Solutions:

- There is an attribute of the shift object that keeps the ID of the user that claimed it
- An additional object holds the interaction of picking up and dropping of shifts

➔ The simplest and most succinct version is creating an attribute that is changed to null when a shift is available. Once it is claimed then the attribute is