Delivery on Demand System

The Problem to Solve

One of the inconvenience situations for faculties or students in UW Tacoma would be the case that some important items or equipment for a class room or office are missed or broken, so the students or faculties cannot take full advantage of the facilities the school provided. It could be a minor problem for some people, but actually, it is not for the most of students and faculties. They expect all the items or equipment the place should have to be there.

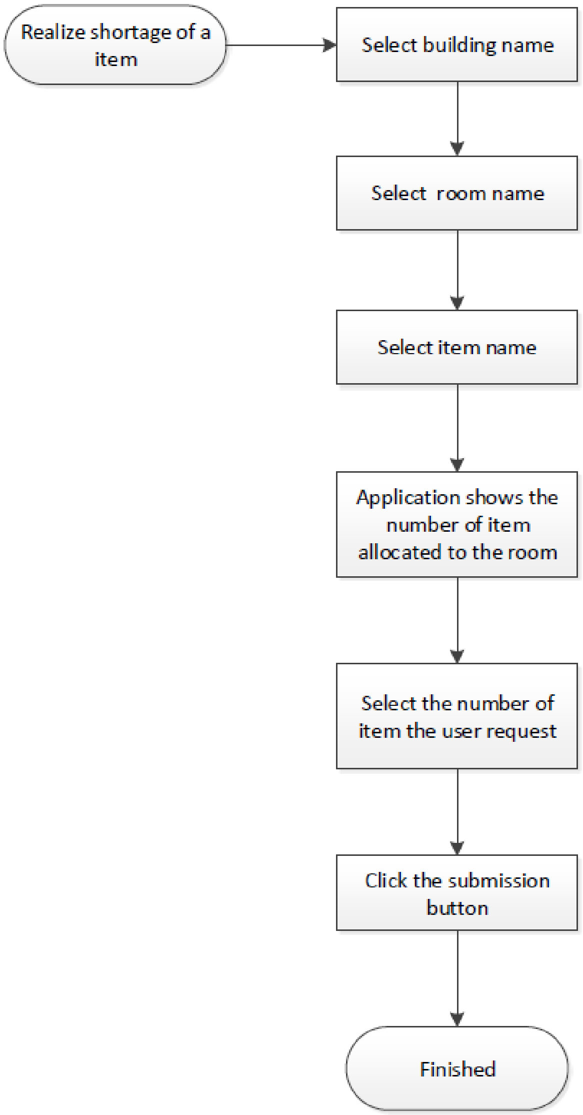
Suggesting Solution

One of the undeniable and apparent ways to make a college prosper is to make the students and employees of the college satisfied, but it is not always easy. However, locating the items which are supposed to be a particular place would be the one of simple and easy ways to make the college prosper. Moreover, if the college decides what kind of items and when they need them to buy depending on the demand history, it would allow more efficient management of their resources.

For that reason, I would like to suggest a Delivery on Demand System which consists of three parts. First part, students or employees in UW Tacoma can request certain absent or broken items which are supposed to be there in particular places (especially rooms) on their demand. The below diagram will help you understand how the application is supposed to be used for the end user.

Delivery on Demand

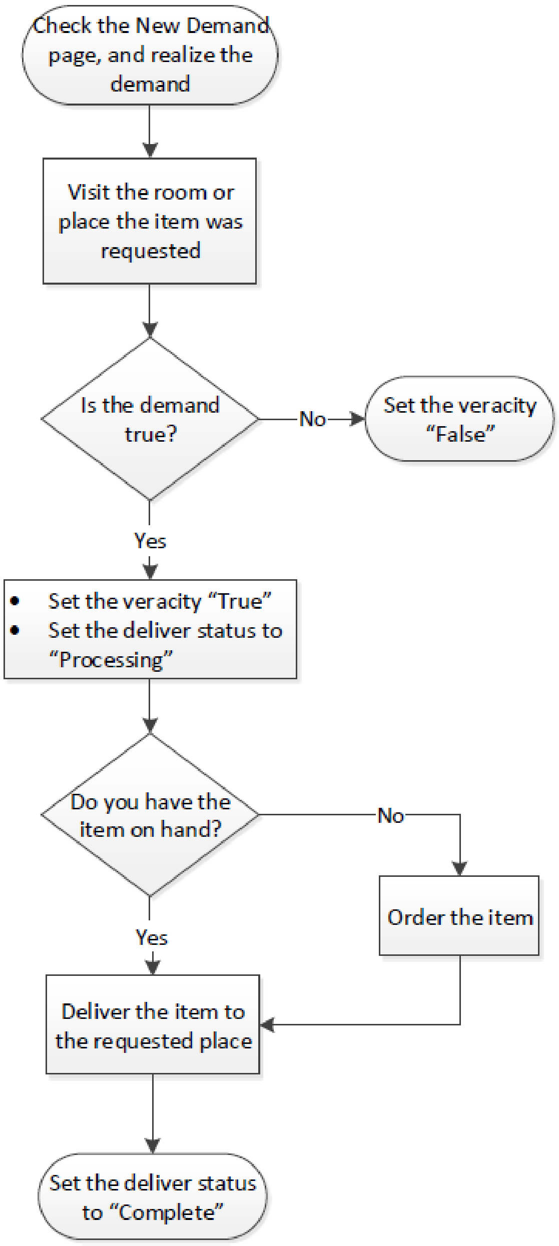
Item Demand Application Flow Chart


Realize shortage of a item
Select building name
Select room name
Select item name
Application shows the number of item allocated to the room
Select the number of item the user request
Click the submission button
Finished

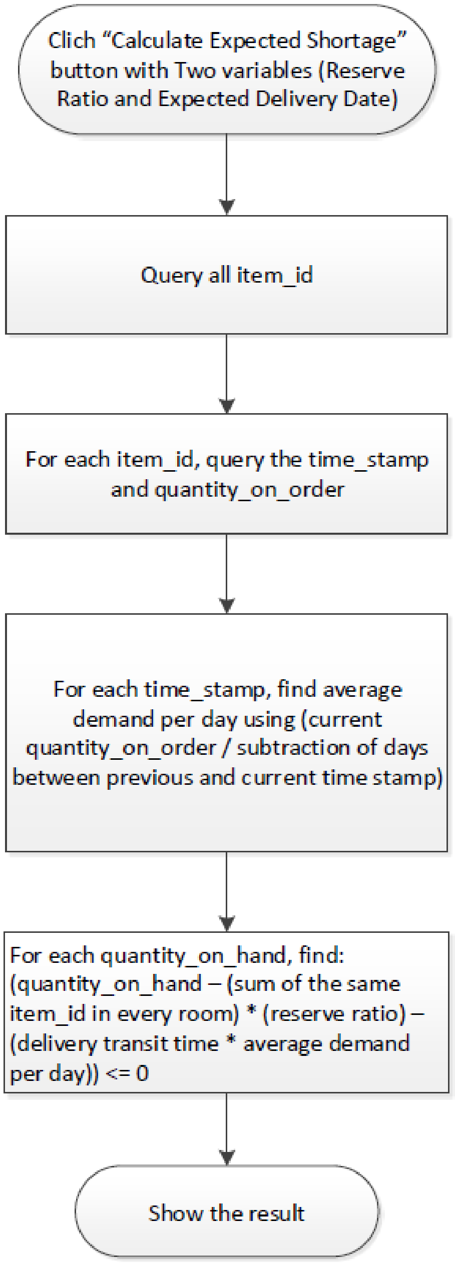
Second, the staff who takes responsibility of managing items and equipment of the school will check the demand last day using the Delivery on Demand system, then physically go to the location to check whether the demand is true or not. If the demand is true, locate the items from school storage. The below diagram will help you to understand the concept.

Delivery on Demand

New Demand Flow Chart


Check the New Demand page, and realize the demand
Visit the room or place the item was requested
Is the demand true?
No
Set the veracity "False"
Yes
Set the veracity "True"
Set the deliver status to "Processing"
Do you have the item on hand?
No
Order the item
Yes
Deliver the item to the requested place
Set the deliver status to "Complete"

The last part of the system provides certain anticipation mechanism to expect the shortage ahead of it really happens depending on the demand history. The calculation will be done with several information: the total number of each type of items located in the whole campus plus the number of items on their hand, reserve ratio of the stock, and expected delivery date from the item store (or the date they want to check). The below diagram will also help you to understand the concept.


Clich “Calculate Expected Shortage” button with Two variables (Reserve Ratio and Expected Delivery Date)
Query all item_id
For each item_id, query the time_stamp and quantity_on_order
For each time_stamp, find average demand per day using (current quantity_on_order / subtraction of days between previous and current time stamp)
For each quantity_on_hand, find: (quantity_on_hand – (sum of the same item_id in every room) * (reserve ratio) – (delivery transit time * average demand per day)) <= 0
Show the result

Delivery on Demand

Calculate Expected Shortage System

Flow Chart

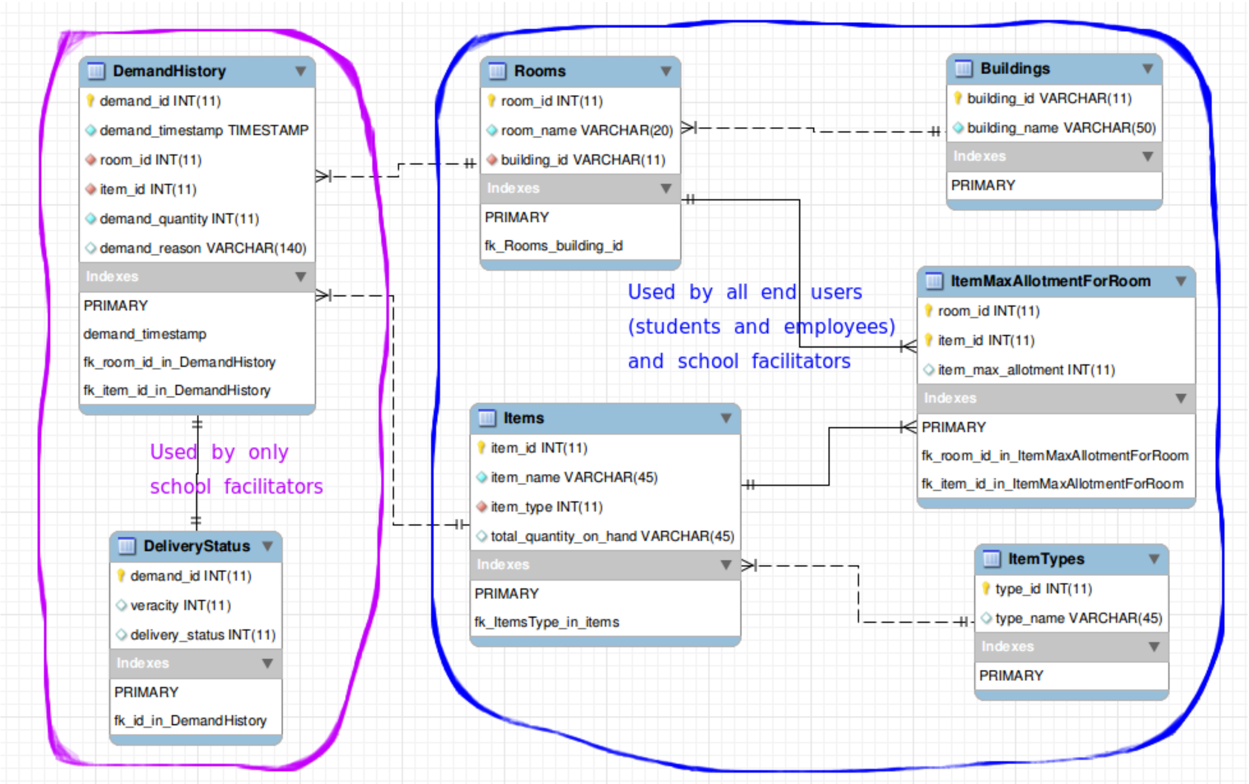
Technical specification

Ubuntu 12.04 Tomcat7 Server at Amazon Web Service using EC2 server, MySQL 5.5, MySQL Workbench for EER diagram, Spring Framework MVC3 for server side programming, Hibernate for ORM in Spring Framework, and Twitter Bootstrap 7 for client side programming.

Milestones

1. Section 2, 3: Finish the normalization on the whole attributes needed for the application and draw the ER diagram (done by Feb 16, 2013)
2. Section 4: Populate the database (done by Feb 18, 2013)
3. Section 5, 6: Code the front end of the application (done by March 21, 2013)

Enhanced Entity Relation (EER) Diagram



Used by all end users (students and employees) and school facilitators

Used by only school facilitators

Index Design

Most of the queries the application uses are on primary and foreign keys in each table, so the indexes are on the primary and foreign keys. Only exception is the index on the demand_timestamp in DemandHistory table. The reason was to implement the function: calculate expected shortage. The calculation will be on certain period of time (e.g. 2013/01/01 00:00:00 to 2013/03/22 00:00:00), so an index is needed for fast fetch. However, I realized that there can theoretically be more than a row having a same timestamp, but it would not be happening in real world because of the latency of the network and transaction of the database. For that reason, I thought that it would better have the demand_timestamp as id column, but it was late to change the schema.

Future Work

I have not finished the function: calculating the expected shortage, but this project is not only for the class final project, but also I want to propose the idea to the IT department in UWT because I'm working as a software developer in internship for the IT department. I'm implementing the function before end of this week, and might be working on authentication, authorization and security parts of the application, mainly for learning Spring Framework purpose.

Comment for the class

It was very good time for me to take the database class. I do not still know many parts of the database, I might learn about the tip of the big iceberg, but I'm very interested in developing an application, and eager to make a good application for my start up in the future. The all the concepts I learned in the class were very helpful to expend my knowledge in software designing.