677 Design Documentation

Nishant Raj, Noel Varghese, Rishika Bharti

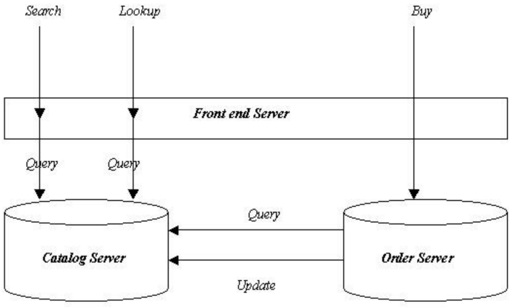
14 March 21

1 System Design

This project aims to design a two-tier web application for an online book store called Pygmy.com. The bookstore sells the following books.

Item Number	Book	Topic
1	How to get a good grade in 677 in 20 minutes a day.	Distributed Systems
2	RPCs for Dummies.	Distributed Systems
3	Xen and the Art of Surviving Graduate School.	Graduate School
4	Cooking for the Impatient Graduate Student.	Graduate School

The two tiers of the application are the front-end and back-end and each tier consists of various components as depicted below in the image.



1.1 Frontend

The front-end application is designed to support user-requests in the form of three operation which are listed below.

- ***search(topic)*** Using the search operation, the user of the web-application can specify a topic and the operation will query the servers in the back-end to retrieve all the books thta come under that particular topic. The returned output contains the title and the item number associated with the title.

- ***lookup(item_number)*** This operation allows the user to pass a item number that belongs to a particular book and the front-end then returns the details associated with the book having the item number. The details include the number of books of that type available in stock and the unit cost of the book.

- ***buy(item_number)*** The buy operation allows the user to specify an item number that thye wish to purchase.

  All the aforementioned operations that hit the front-end server query the servers that are set up on the back-end.

1.2 Backend

The back-end consists of two components which are the catalog server and order server.

1.3 Catalog Server

The catalog server maintains the catalog for the various books that are sold in the bookstore. For every entry, the catalog server stores the number of items in stock, the cost and the topic of the book in the database.

1.4 Order Server

The order server maintains a list of all orders received for the books that the users wish to buy from the catalog.

The operations from the front-end query different servers on the backend. The search and lookup functions trigger a request on the catalog server. We can query items from catalog based by two ways which are through the topic of the books and by the item number. The catalog server also supports the update operation which allows us to update the cost of a particular item or the increase or decrease the stock of tha