CS2102 Database Project Report

Carolend

DONG SHAOCONG, HE XINYI, LIU YULIN, WANG ZEXIN

Department of Mathematics

National University of Singapore

AY2017/18 Semester 1

Abstract

Acknowledge

We would like to thank Associate Professor Bressan Stephane for his helpful supervision throughout the course of this project.

Contents

1 Introduction

In this project, we are required to build a stuff sharing website. the system allows people to borrow or lend stuff that they own (tools, appliances, furniture or books) either free or for a fee. Users advertise stuff available (what stuff, where to pick up and return, when it is available, etc.) or can browse the available stuff and bid to borrow some stuff. The stuff owner or the system (your choice) chooses the successful bid. Each user has an account. Administrators can create, modify and delete all entries.

1.1 Developing Specifications

After seeing through the relevant products specifications and the website requirements, we decided to use PHP as back end programming language, $Javascript$ as front end developing language, $MySQL$ as our database. We used $Laravel$, which is the most popular web application framework for PHP.

From the project requirement, Eloquent ORM in built in Laravel to access and manipulate the database is not allowed. Therefore, anything related to database management, access, and manipulation is down by importing PHP $mysqli$ library and execute raw SQL queries.

To develop this web application, we utilise the Model View Controller (MVC) framework.

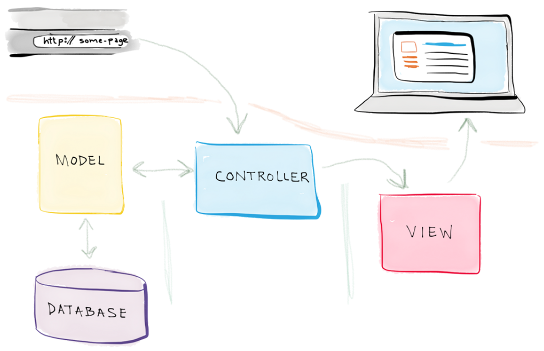

Figure 1: Model View Controller (MVC)

We use CSS Scaffolding in Laravel to take care of the views. Buttons and redirections from the user side are implemented in $Javascript$. The models and controllers are written in PHP. The specific $MySQL$ database we used it *root* user's database named *blog*. The website is still hosted on `localhost:8000` only.

From our modelling for this problem, there are two types of users. The admin users will have different interfaces and unlimited access to all entries. From the user management button in the admin user's profile page. All the users information can be modified and deleted. New users can be added on this panel as well.

2 Database Design

2.1 Entity-Relationship Diagram

2.2 Entities

Users

Attribute	Domain
email	VARCHAR(64)
username	VARCHAR(64)
password	VARCHAR(64)
mobile	INT(8)
address	VARCHAR(128)
points_available	INT(3)
credit_rating	NUMERIC
created_at	TIMESTAMP

Items

Attribute	Domain
name	VARCHAR(64)
avatar	VARCHAR(256)
owner	VARCHAR(64)
description	TEXT
available	VARCHAR(5)
created_at	TIMESTAMP

Posts

Attribute	Domain
item	VARCHAR(64)
title	VARCHAR(64)
location	VARCHAR(128)
description	TEXT
start	TIMESTAMP
end	TIMESTAMP
created_at	TIMESTAMP

Bids

Attribute	Domain
bidder	VARCHAR(64)
post	VARCHAR(64)
status	CHAR(7)
points	INT(3)
created_at	TIMESTAMP

Loans

Attribute	Domain
bid	VARCHAR(64)
post	VARCHAR(64)
start	TIMESTAMP
end	TIMESTAMP
comments	TEXT
status	VARCHAR(8)
created_at	TIMESTAMP

2.3 Relational Schema

2.4 Schema Functions

3 SQL Queries

3.1 Simple Queries

3.2 Aggregate Queries

3.3 Nested Queries

3.4 Queries using INNER JOIN

3.5 Queries using EXISTS

3.6 Queries using set operations

3.7 Insertions, Deletions and Updates

3.8 Stored Procedures and Triggers

4 Web Interface Design

4.1 Login Page

5 Sample table and diagram insertion

StrikePrice	Closed-form formula	Ordinary Monte Carlo	Control Variate
105	10.0022021172	10.005252032814727	10.005252032814751
110	8.02638469385	8.0166707887876427	8.016670788787664
115	6.37924904693	6.3659464432937911	6.3659464432937733
120	5.02541348179	5.0148870431746415	5.0148870431746184
125	3.92690420603	3.9241698581683577	3.9241698581683728
130	3.04592058431	3.044679765401427	3.0446797654014213
135	2.34679877596	2.3310307686205207	2.3310307686205225
140	1.79723400902	1.8055531375480696	1.8055531375480751
145	1.36889248498	1.3630119824610754	1.3630119824610734
150	1.03756650489	1.0254470920297398	1.0254470920297361
155	0.783018613011	0.77819086371547286	0.77819086371547308
160	0.588637155719	0.5924848566970754	0.59248485669707973
165	0.440997057228	0.44342591182216823	0.4434259118221664
170	0.329392108384	0.32449718396190719	0.32449718396190735
175	0.245381782063	0.2462392632801686	0.24623926328016907
180	0.182377553986	0.17995020687354496	0.1799502068735449
185	0.13528073067	0.13478417666883458	0.13478417666883413
190	0.100175092579	0.099449778570450814	0.099449778570450523
195	0.0740722950356	0.074452813719303179	0.074452813719303304
200	0.0547050187389	0.051631534936688268	0.051631534936688074

Figure 2: Effect of control variate in pricing European call options

6 Conclusion

correct way of citing something:[1]

References

[1] Yves Hilpisch, *Python for Finance*. O'Reilly Media, 2015.

[2] Paul Glasserman, *Monte Carlo methods in financial engineering*. Springer, 2010.

[3] Daniel Duffy, *Finite difference methods in financial engineering: A partial differential equation approach*. John Wiley&Sons, 2006.

[4] Mark Broadie, Paul Glasserman, Steven Kou, *A Continuity Correction for Discrete Barrier Options*. Mathematical Finance, 1997.