CSCI110
Spring Session 2013

Assignment 3
PHP and MySQL
14 marks

Complete "Exercises 3 and 4" before attempting this assignment.

Aim

This assignment introduces you to the basics of server side scripting (PHP) and database usage.

The assignment should be completed in the laboratory using the Ubuntu (Linux) environment.

Objectives

On completion of this assignment and its associated exercises, you will be able to:

- Create multi-script web applications.
- Utilize MySQL for persistent data storage.
- Perform simple data manipulation operations using SQL.
- Write scripts in the PHP scripting language.

The assignment includes some limited CSS styling and Javascript, building on your previous experience with these technologies.

The assignment description follows standard Software Engineering (SE) practices with "requirements", "use cases", "storyboard walkthrough", and some limited UML (Unified Modelling Language) diagrams. This is to give you some experience with these SE elements prior to your further studies in CSCI204, CSCI205, and CSCI222.

Overview

You have to create a simplified web-fronted shop that allows users to place orders, and that keeps records of its activities in tables in a MySQL database.

This particular web-shop sells gift items associated with Apple's iPhone, iPod, and iMac products to the insanely rich with no better use for their money.

For this assignment, you need only create the "customer view". (Handling of payments is not a part of your modelled system – a small web-site such as this would do best to sub-contract financial transactions to PayPal.) Administration of the

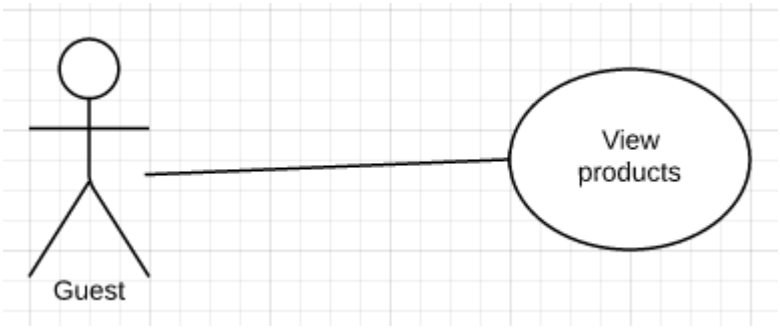
MySQL data tables needed for the site will be done using MySQL Workbench (or the Netbeans client) rather than through a web interface.

Requirements

1. The system shall maintain a database of products that are available for sale through the web-front store. Each product has a title, description, image, price, and optionally a discount on that price.
2. The system shall enable visitors to the web site to view a tabular display of icons for the products on sale, and link through these icons to view details of products.
3. The system shall enable registration by those desiring to purchase items. Persons registering will create accounts with their email address, name and physical address.
4. The system shall allow registered users to purchase (multiple copies) of any product; purchases are recorded.
5. The system shall provide convenient log-in and log-out mechanisms for registered users.
6. The system shall "personalise" web pages presented to registered users.

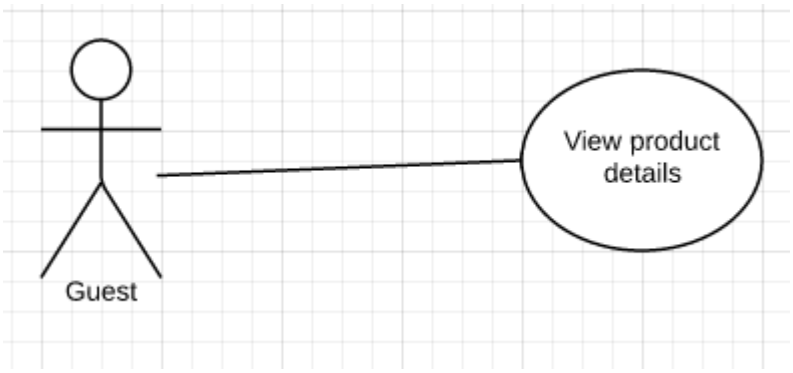
Use Cases

Guest view products



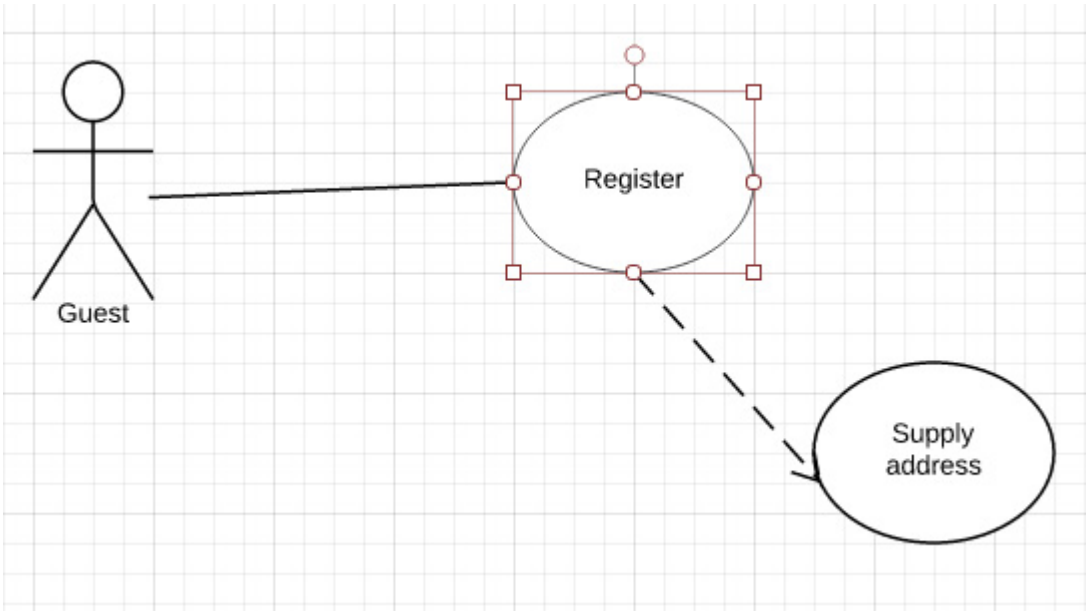
Any internet user may view a display showing all the products available from the store.

Guest view product details



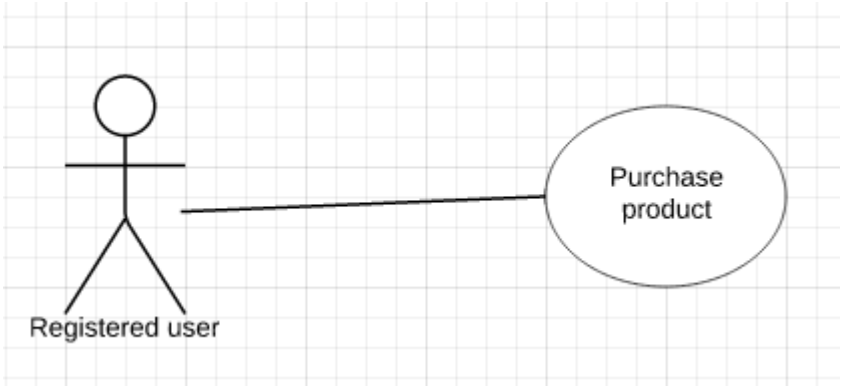
The display of products available at the store provides links to more detailed information on each item

Guest registers with store



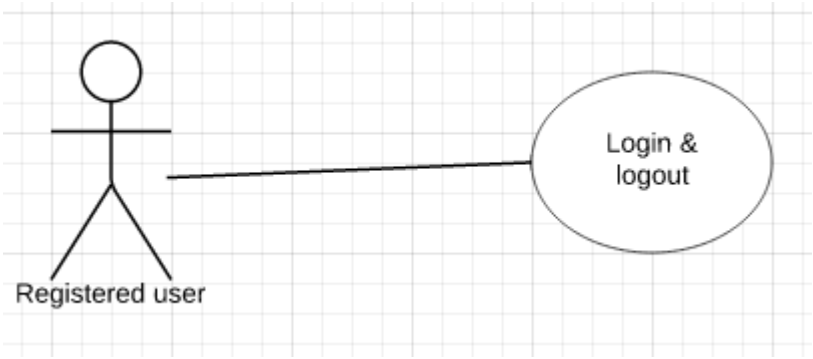
A guest must register before he/she can actually order product. Registration is a two step process. In the first step, the guest supplies identification data – email address, and name. In the second step, physical address data are entered (street address, city, state, and postcode).

Registered user purchases item



A registered user can elect to purchase (multiple copies) of any chosen product.

Registered member – login and logout

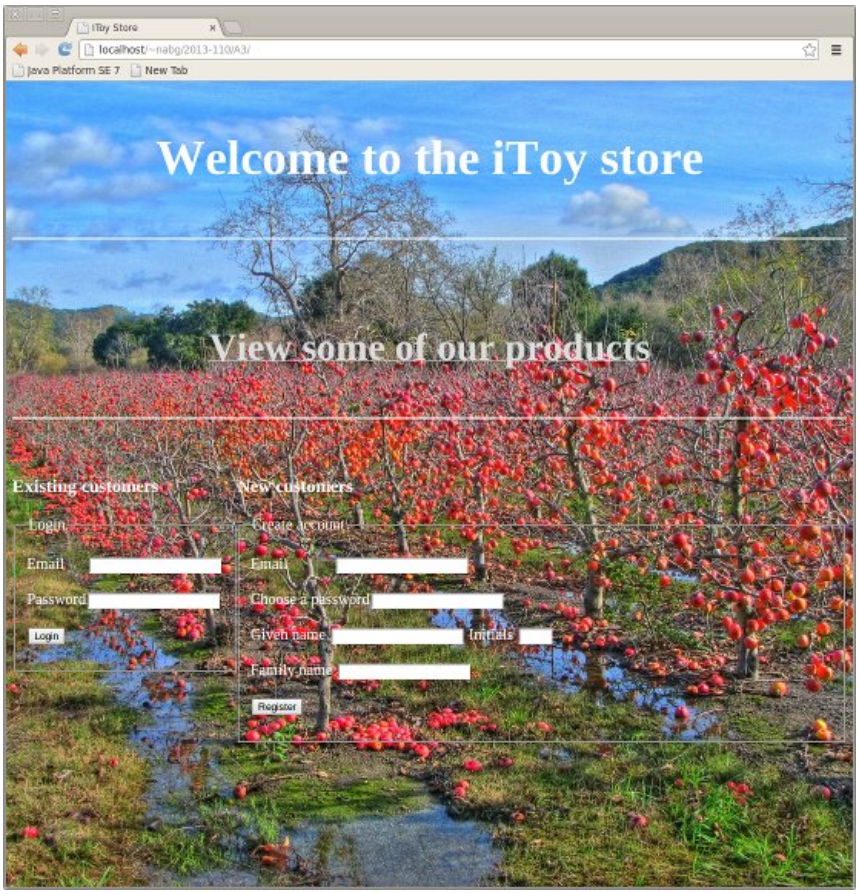


A person who has already registered with the site will use a login option (and logout option) on return to the site.

"Storyboard"/Walkthrough

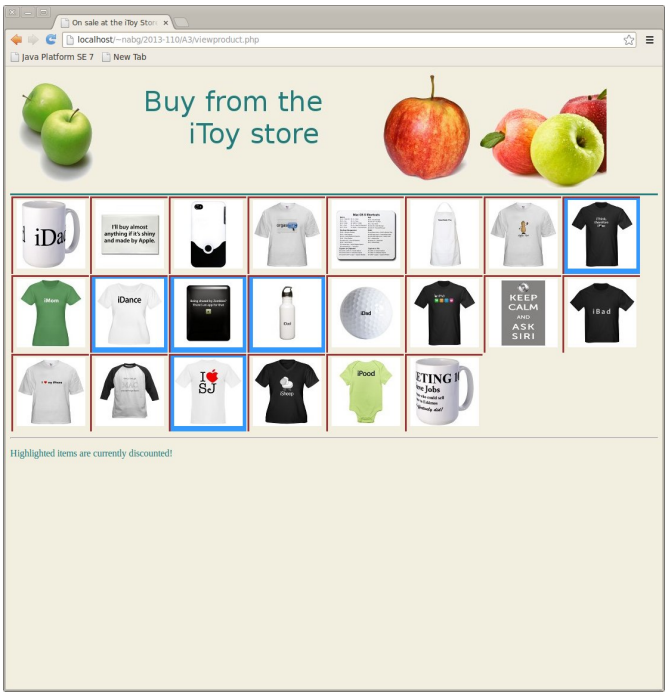
Following the specification of "use cases", the next step in the development of an interactive program is often the creation of some form of "storyboard", or "walkthrough", or "preliminary user manual" that illustrates how the program is expected to work.

"Index" page (front page of web site):



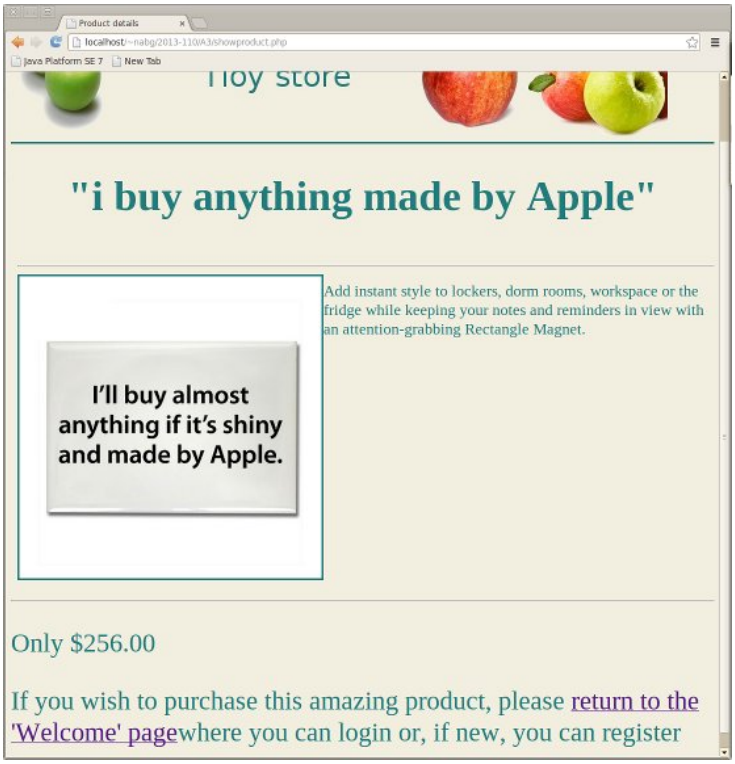
The front page should have a link to the display of products and forms that allow an existing customer to login, or a new customer to create an account.

"View products on sale" page (guest):



The "view products available" page should show a table of image links. The table is generated dynamically from data in the system's "products" data table. It shows icon images for each product; if a product is currently discounted, the icon will be highlighted.

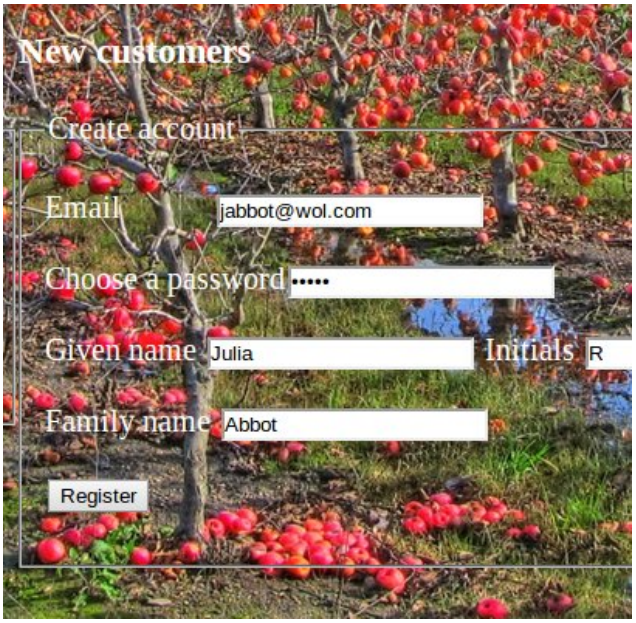
"View product details" page (guest):



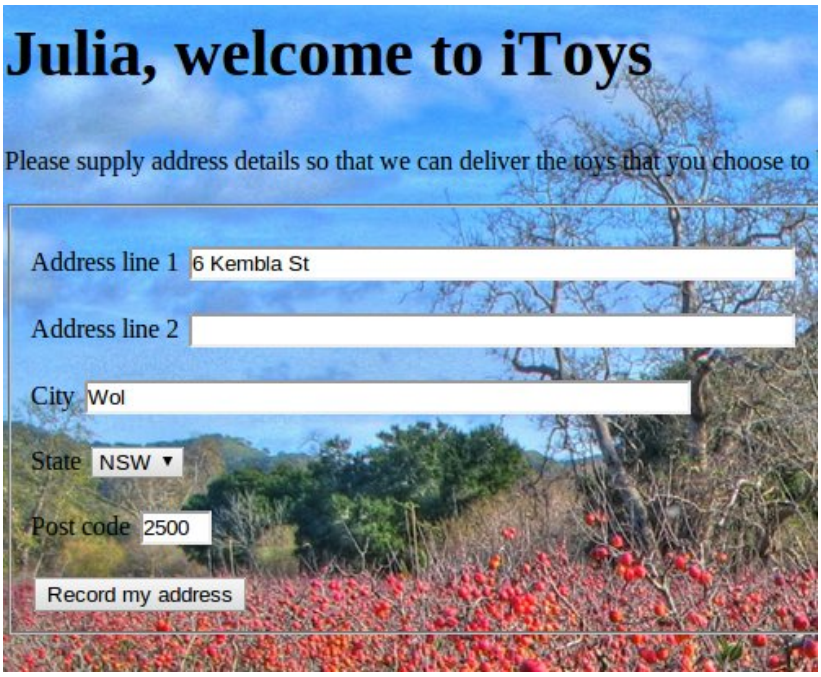
The "view product details" page shows the title, image, and description for a chosen product along with the price. (If a discount currently applies to a product, the normal price and current discounted price are shown – this feature is illustrated in a later

screenshot.) If the product details page is being viewed by a "guest", then a link will be supplied at the bottom of the page that will take the user back to the index page where the login and register forms are located.

Register



A new customer who wishes to buy from the iToy store will complete the registration form and then supply address data (address line 2 is optional):



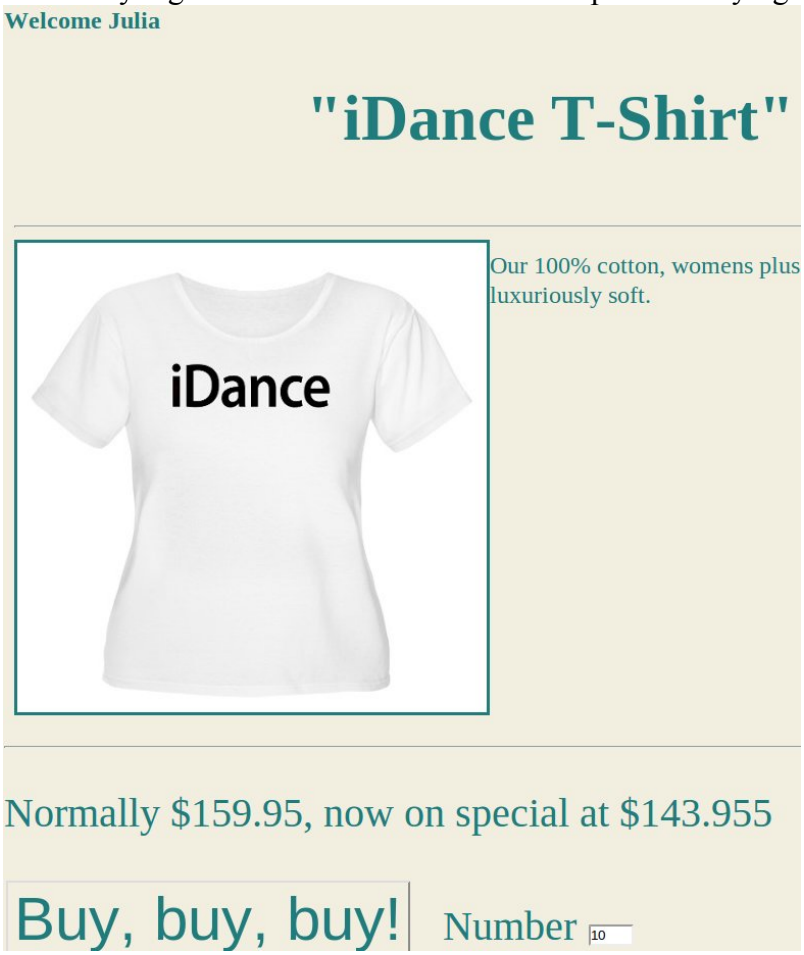
(Registration data and address data are to be checked with Javascript on submission, and further checked by the processing script. The checks are simple. All required fields must contain input; the characters strings shouldn't contain HTML markup or Javascript attack code; state and post code should be consistent etc. If the data fail the

Javascript checks, submission is prevented. The server side code will simply re-display the data entry form if invalid data are found.)

If registration is completed successfully, the new user is shown the products page:

The products display page is "personalized" – simply an extra greeting.

The newly registered user will now have the option of buying any selected product:

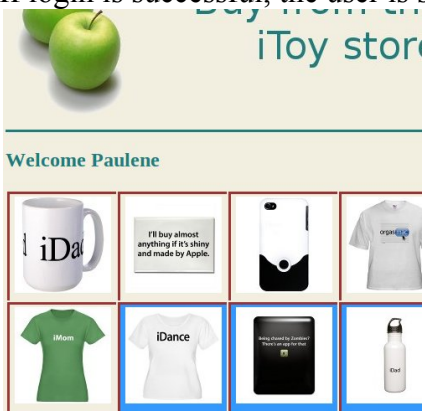


Login

A visitor who already has an account with the iToys store will use the login form:

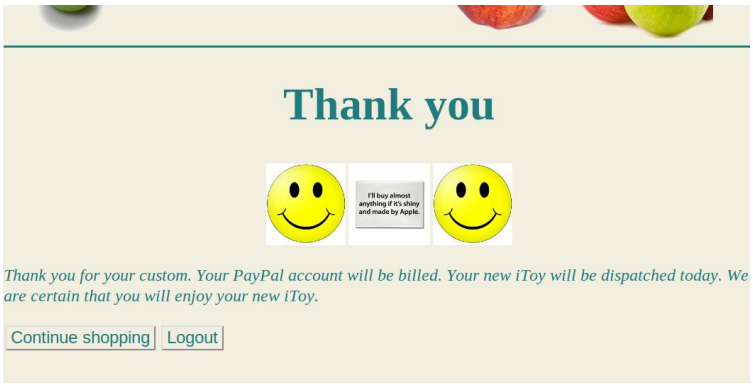
If the combination of name and password is invalid, the system shall show an error page:

If login is successful, the user is shown the products page;



She can purchase items:

The system generates a simple response page thanking the user for their custom; the page has options to continue shopping or logout:

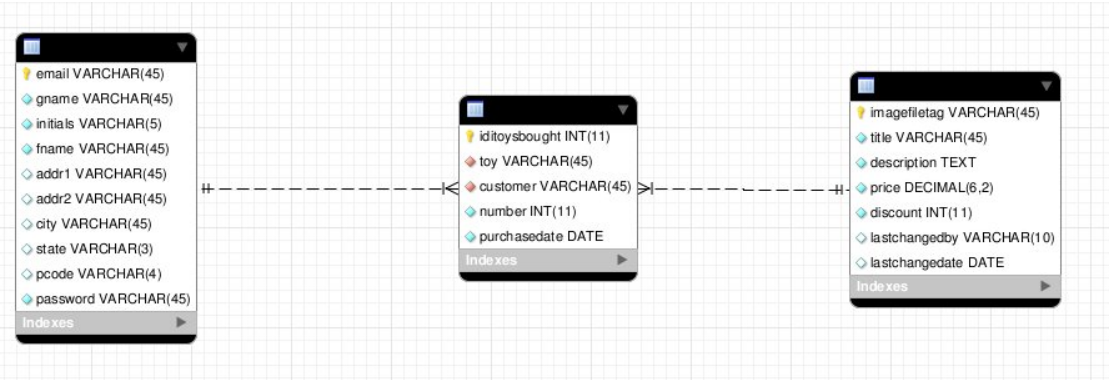


The logout option terminates the on-line session:

Data model

The next stage in developing an application is typically refinement of the "data model". The "data model" is the conceptual structure for the database tables, and where appropriate program language classes, that are used for persistent data.

The following diagram was created in the ER modelling component of MySQL's "WorkBench" – it suggests the basic form of the persistent data that will be needed for this application.



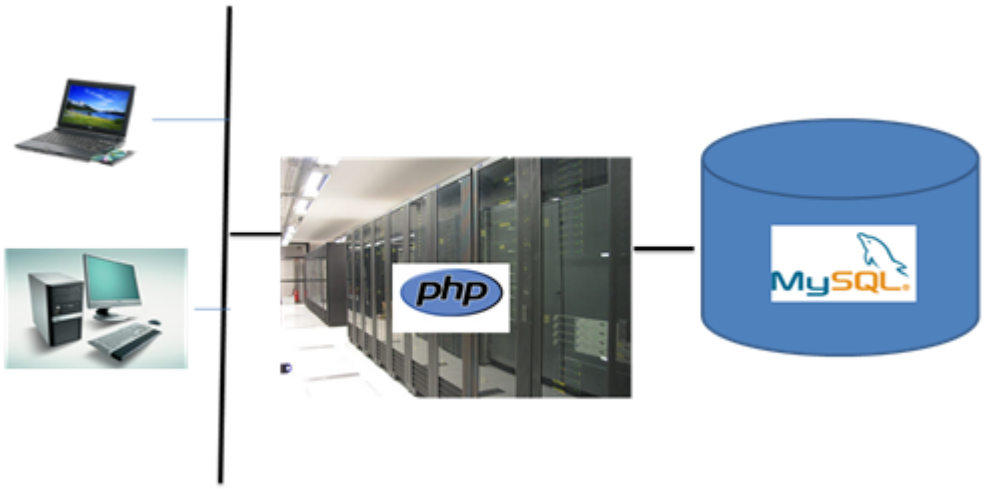
You should write an SQL script that creates your tables. You will be able to run that script on any of the computers to recreate the tables in the local instance of MySQL. Your "createtables.sql" script should be included in your final report.

It's probably easiest for you to use MySQL's table editor to build up a version of your table definition – just cut and paste its generated SQL into your script. (You should specify INNO database – it's the only one that enforces foreign key constraints.)

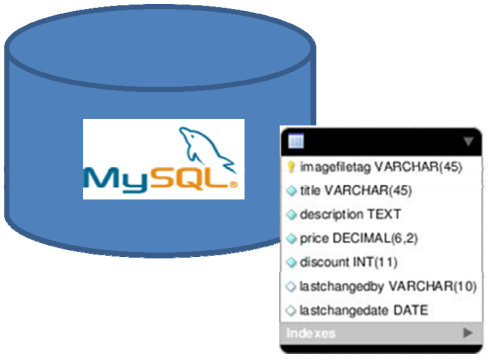
Architecture

The next system's development task is refining the architecture.

The high-level architecture is easy! It's the same for all web-applications:

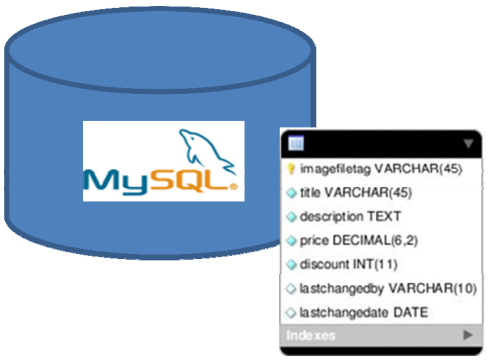


At a more detailed level, *before you start coding,* you should simply clarify the workings of each of the PHP scripts that make up your application.



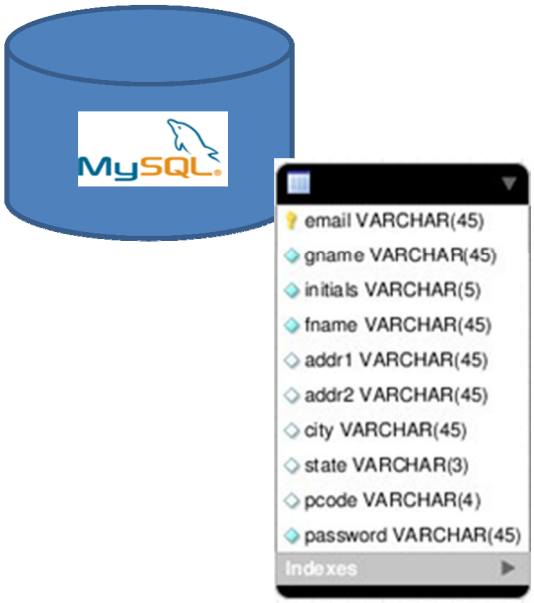
ViewProduct:

Get: Display a form that allows user to select a product that they would like to view in detail.
The form consists of a set of <input type= 'image' …/> elements. It is composed using a "select imagefiletag from itoyproducts" query.



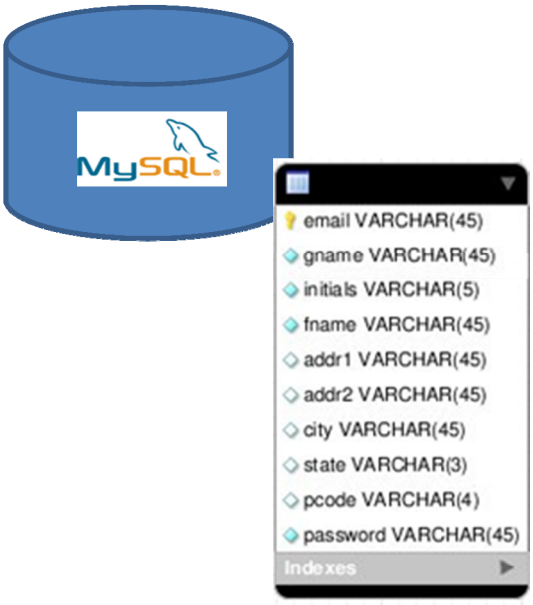
ShowProduct:

Get: Display product details, and if user is registered a form that can be used to purchase (multiple instances) of a chosen product.



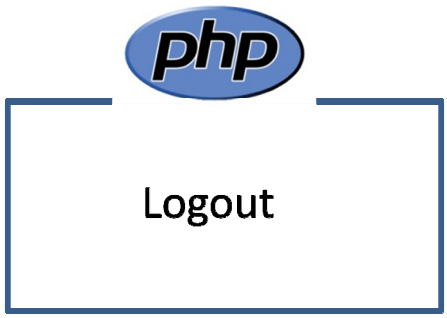
CustomerLogin:

Post: Check supplied user name and password against data in tables.



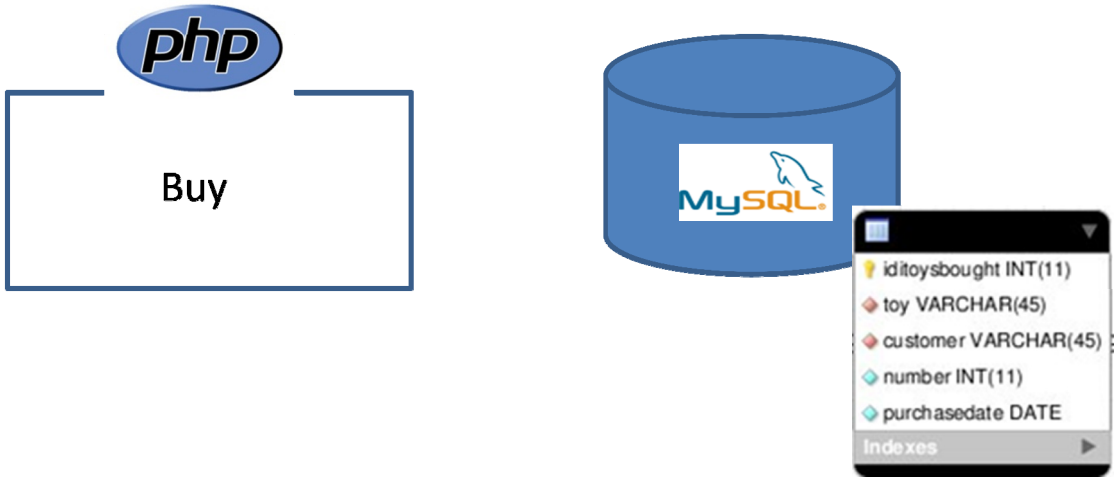
CustomerRegister and CustomerAddress:

Post: Together these scripts will handle the creation of a new customer record in the iToysCustomers table.



Logout:

Get: Terminates current session.



Buy:

Post: Adds record to idtoysbought table.

Data

Really, it's *your responsibility* to create the data sets that will be employed when building and testing your application!

But part of it has been made easy for you. Data that you can use to create your products table has been put in /share/cs-pub/110/A3 on banshee.

It's best to create a few initial records in your itoyscustomers table with an SQL script – creating more entries through your PHP code once this is working.

Assignment report

You do not submit your web pages!

You are marked on a report that you write. This report presents your web site in the form of screen-shots, the Javascript file, the style sheet, the PHP scripts, your SQL scripts along with supplementary explanation and comment. Code listings should be included with proper formatting – use the "Print as HTML" option in NetBeans to get formatted versions that can be pasted into a report. (Do use the HTML route, don't paste in screen shots of code.) The report should be prepared on a word processor and converted to PDF format. The Open Office word processor on Ubuntu Linux has a "print to PDF" option. There should be Word to PDF converters on the Windows OS. Ubuntu has a snapshot tool (in its accessories menu). Screenshots should be scaled down when pasted into reports.

Reports should start by a clear definition of the task (and don't just cut and paste from this document). They should belong in a portfolio of work that you can present to potential employers.

Submission

Prepare your report and convert to PDF format as the file A3.pdf.

The report should be submitted electronically via the `turnin` system. For this assignment you submit your assignment via the command:

```
turnin -c csci110 -a 3 A3.pdf
```

Late submissions would be submitted as:

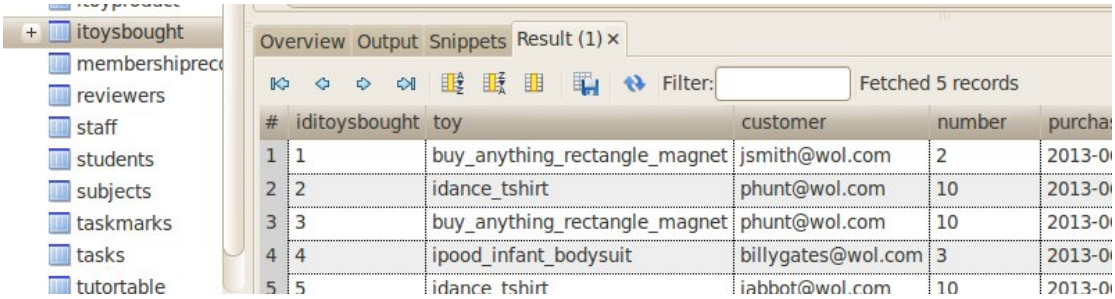
```
turnin -c csci110 -a 3late A3.pdf
```

The program `turnin` only works when you are logged in to the main banshee undergraduate server machine. As in assignment 1, you must transfer your report to the banshee machine using some form of ftp. Then from an Ubuntu workstation in the lab, you must open a terminal session on the local machine, and then login to banshee via ssh and run the `turnin` program.

Marking

The assignment is worth 14 marks total.

- Appearance of report: *1 mark*
  Cover, index, proper sectioning, brief explanation of task in each section, images, neatness of included code, …
  A good structure would be
  - Title, and index
  - Overview
  - Demonstration of working site – screen shots forming a "walkthough" similar to that shown above.
  - Database aspects.
  - Summary of NetBeans project
  - Structured presentation of code – each script presented with brief commentary.
- Appearance of web site: *3 marks*
  CSS code and its usage, form layout, data presentation, etc. (and the minor Javascript component)
- Database aspects: *3 marks*
  Create table scripts etc; evidence for database being created and data loaded.
  You can use screenshots of MySQL Workbench views of tables to show that data have been inserted.



#	iditoysbought	toy	customer	number	purcha
1	1	buy_anything_rectangle_magnet	jsmith@wol.com	2	2013-0
2	2	idance_tshirt	phunt@wol.com	10	2013-0
3	3	buy_anything_rectangle_magnet	phunt@wol.com	10	2013-0
4	4	ipood_infant_bodysuit	billygates@wol.com	3	2013-0
5	5	idance_tshirt	jabbot@wol.com	10	2013-0

6	sheila@wol.com	Sheila	S	Smith	5/55 Kembla St	NULL	Wol	NSW	2500	dd
7	billygates@wol.com	William	F	Gates	16/28 Cliff Road	NULL	Wol	NSW	2500	42
8	jabbot@wol.com	Julia	R	Abbot	6 Kembla St	NULL	Wol	NSW	2500	ee

- Scripts: *7 marks*
  Scripts must be accompanied by evidence of their working – screen shots, table listings showing evidence of table changes.
  - View of all products, and detailed view of a product – *3 marks*
  - Registration of new user – *2 marks*
  - Login, logout of registered user, and personalization of pages – *1 mark*
  - Record a sale – *1 mark*

The "summary of the NetBeans project" is just a screen shot like the following along with any necessary explanatory commentary: