

Nathan F, Jason L, Boris T and Stefan Z.
CIS330 Final Project
WineSquare Project Write-up

1. Web Application Overview

WineSquare is a web application that allows users to “check-in” at locations where they are enjoying fine wines. In addition to this check-in functionality, the application is a space where users can log in, check the activity of all users on the site through a recent activity “news ticker”, see “badges” earned from drinking certain amounts/ types of wines, and get personalized recommendations on wines to try based on past check-ins of the user and preferences of other users on the site. We are currently at 29 users and are looking forward to continued growth.

2. Implemented Features

We were able to pack some fun features into our WineSquare application. They include:

- Signup system involving error correction to prevent mismatched passwords, duplicate users, invalid fields, etc.
- A badges system (10 in all, collect them all!) that reward users based on total number of check-ins, types of wines tried, and frequency of check-ins. We keep track of which badges each user has obtained and dynamically generate the resulting profile and badges pages.
- Map integration throughout the web app; using Google’s Map API and the Javascript tool GMap we added a dynamic system to view wine origin on the wine page and a method to “Visualize your history” of check-ins from your profile page
- A separate location awareness system that is able to track and store your location through the browser for check-ins. In addition to visualizing this data, users can see the location of their last check-in from their profile page
- A recommendations engine that looks at previous wines tasted and history of other users to determine wines (based on region) that the user would enjoy.
- Dynamically created, individual profiles displaying check-in history, totals for check-ins, wines tasted and Badges collected, and recommendations
- Search capabilities for all of the wines in our catalog (over 2000 in total). Searching for a wine will take you to its individualized page showing all the details provided by Freebase.

Please see attached for screenshots

WineSquare

Sign Up or [Log In](#)

Please enter your information so we can get you started on WineSquare.

First Name*

Last Name*

Email Address*

Password*

Confirm Password*

Gender*

☒ Male

☐ Female

Birthday*

Month Day Year

Photo

[Choose File](#) | No file chosen

[Join »](#)

Signup Page

WineSquare

Search Wines... Profile Logout

Jason Lucibello

Please change any information you would like to be updated.

First Name*

Last Name*

Email Address*

New Password*

Confirm Password*

Gender*

☒ Male

☐ Female

Birthday*

Month Day Year

Photo


[Choose File](#) | No file chosen

[Update Profile »](#) [Delete Profile »](#)

Profile Edit

WineSquare

Search Wines... Profile Logout



Jason Lucibello

Last drank at: 3723 Locust Walk, The Wharton School, University of Pennsylvania, Philadelphia, PA 19104, USA

10 Total Check-ins 9 Wines Tasted 4 Badges Collected


History

Visualize my history!

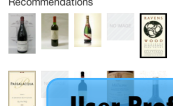
- 2005 Bella Alexander Valley Late Harvest Zinfandel
Dark, super-ripe fruit, with flavors of superlumps, caramel, cola, and spice. It has a long, lingering, mouth-coating finish that lasts forever. It is the perfect finishing touch with cheeses or a dark, rich chocolate dessert.
- 2005 Poggiarellino Rosso di Montalcino
The Wine Spectator called it: "Simple, with cherry and blackberry aromas and flavors. Medium-bodied, with a fresh finish. Drink now. 345 cases made." (08/07)

Badges

See All




Recommendations



User Profile

WineSquare

Search Wines... Profile Logout



2005 Poggiarellino Rosso di Montalcino

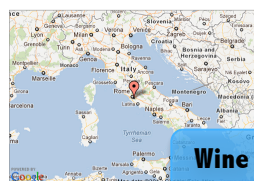
Location: Italy

Vintage: 2005

Wine Producer: Poggiarellino

Percent Alcohol: 13.5 %

Description: The Wine Spectator called it: "Simple, with cherry and blackberry aromas and flavors. Medium-bodied, with a fresh finish. Drink now. 345 cases made." (08/07)



Wine Page

WineSquare

Search Wines... Profile Logout

Check-In to WineSquare!

You were drinking:

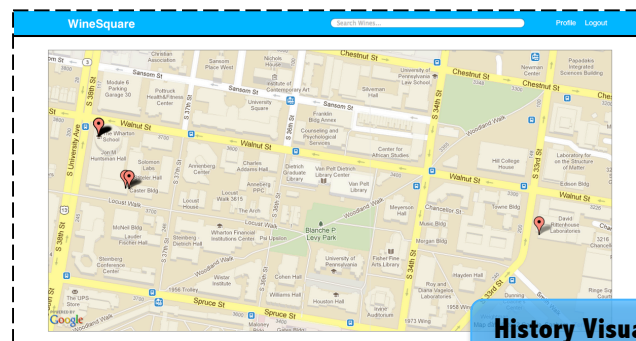
2004 Domaine Prieur Brunet Jehanne de Chantal

At: 3723 Locust Walk, The Wharton School, University of Pennsylvania, Philadelphia, PA 19104, USA

On: Thursday, May 03, 2012 at 19:50 PM

[Drink! »](#)

Check-Ins







History Visualizer

WineSquare

Search Wines... Profile Logout

Jason's Badges

These are all the badges that Jason has unlocked by drinking wine! Click on a badge to learn more about it, including how to earn one for yourself!


Drink More! Drink More! Drink More! Drink More!

Earned Badges

WineSquare

Search Wines... Profile Logout

See All Badges



The Winstar

That's 10 wines you've consumed!

Congratulations! Here's a Gold Star. Now get back out there!

Earned by Jason Lucibello on 2012-05-03 19:29:37

Badge Page

3. Changes to our Original Schema

Below is our original schema design that we submitted. While we did not drastically alter our schema design, we made a few pointed changes to provide functionality that we hadn't thought of during our original brainstorm. Additionally, we eliminated Milestones (what would have been like "Mayorships" in FourSquare) because we realized that the FourSquare model didn't fit perfectly with our idea for WineSquare. While FourSquare's focus is on the location where people check in, our app focuses more on the wine people are drinking rather than the location where they are drinking it. Instead, we integrated the idea of "milestones" with our badges, and created the sense that badges were goals to reach (such as drinking a certain number of unique wines or drinking a certain number of wines within a given period of time).

Original Project Schemas

- *User* (uid, first_name, last_name, username, photo, birthday)
- *Wine* (wid, year, picture, description, country, percent_alcohol, variety)
- *Drank* (uid, wid, time, location)
 - uid references User, wid references Wine
- *Badge* (title, icon)
- *Has* (uid, wid, title, time)
 - uid references User, title references Badge
- *Milestone* (title, icon)
- *Earned* (uid, title, time)
 - uid references User, title references Milestone

Constraints not captured by Schema:

- Only one user can hold a badge at one time for a unique win (Mayor, Senior drinker, etc)

Now, below is our finalized, actual schema design for our database.

Project Schemas

- *User* (user, password, first_name, last_name, sex, photo, birthday)
- *Wine* (wid, name, descr, pic, vintage, alcohol, producer, country)
- *Drank* (uid, wid, time, location)
 - uid references User, wid references Wine
- *Badge* (title, req, isUnique, subtitle, descrip, photo)
- *HasBadge* (uid, title, time)
 - uid references User, title references Badge

Changes to note:

- “Milestones” entity, “Earned” relationship both eliminated/omitted as previously mentioned above
- “Has” relationship renamed to “HasBadge”
- “HasBadge” no longer requires ‘wid’, since it’s just a matter of a user having one of the given badges.
- “User” entity changed slightly to account for password, etc. Also, ‘user’ attribute, which was originally ‘uid’, is just their email address, which was how we kept track of each user/ what we assigned as each user’s unique attribute.
- “Badge” entity has more attributes.
 - ‘req’ is the number of checkins needed to unlock the badge.
 - ‘isUnique’ is a boolean that represents whether or not the requirement requires unique wines or just plain number of checkins.
 - ‘subtitle’ is the text to be dynamically placed right under the title on the badge page. Similarly, ‘descrip’ is the paragraph or so of text to be inserted dynamically on the badge page for the given badge.
- “Wine” entity is changed slightly to account for the most the freebase attributes that had content.

4. Interesting Design Decisions

- PHP + MySQL and phpMyAdmin - Since we wanted to build something interesting, entertaining and useful, plus we wanted to learn something new, we decided to use PHP as our dynamic backend. Some of the nice features of PHP are that it has a huge library of functions that makes things really simple and connecting to an SQL database was not a problem at all. We use a remote server on which we have the database and we connect it to WineSquare through the php’s mysql library. On the server side, we have the application phpMyAdmin which allow us to manage our database very easily by providing full access to all of the tables’ properties and records. This simplified our work immensely. The choice was definitely a good one because writing the SQL queries was very simple - one only had to write strings with the queries, pass the parameters and then use the `mysql_query()` function. Also, the things usually worked out smoothly and we did not encounter serious bugs. At the same time we learned a lot from doing the project and enjoyed doing something fun like this.
- Location awareness - Using the Google Maps API, we use a user’s IP Address to find their location (the user is first asked to confirm that he would like to share his location). Every time the user opens the “drink”/check-in page, his location is determined and stored along with the other information. Besides simply reporting where each user’s check in was in his recent activity, we added the option to “visualize a user’s history.” After pressing this link, you’re taken to a map with pins at each of the given user’s past “drinking” locations.

- Profile Pages: The profile pages for each user are populated through profile.php. Each time we access a profile, we establish some sort of 'user' whose page it is. If we pass in a parameter to the URL through PHP, that means we load that person's profile page and populate it through queries to various tables based on that ID/email address. If not (i.e. it's just <http://pennquiz.com/winesquare/profile.php>), then that means the 'user' is whoever is logged in, and the page is populated from their email address.
- Home Page "News Ticker": We wanted to find a way that users could interact with each other on the site and see what their friends were drinking and where. Adding the "News Ticker" on the homepage allowed users to see what the WineSquare community was drinking as a whole in a simple way. Other potential implementations of this that we considered were global user statistics of wines consumed or an introduction of the concept of "friends".
- Recommendations: An intelligence element of our project was wine recommendations. We recommend wines to the user (who is logged in) based on wines that other users have drunk that have locations or years in common with the user's drinking history. This is done through one query, and we populate the page based on the results. We display the picture of the wines of interest in a grid format as clickable links to their pages, like we do for badges.
- The Wine Page: In the wine page we experimented with different methods of conceptualizing the data Freebase provided us on wines. We settled on a UI that allowed the user to quickly access the information on the wine (as opposed to a drop down tab a user could click for more information) and a cover photo that gave the impression we were taking the wine "off the rack" to examine and find out more information on it.
- Check-in - The check-in page is probably the most important in the whole application. It was the first on which we experimented with our autocomplete functionality. Basically, on search we pull all the information about the wines from the database and then pack them into an array that we place in the autocomplete box. From there on each key press, the search bar gets refreshed and displays wines that match the typed string. Then when the user picks a wine, it automatically gets inserted in the data field and then a user can *Drink!* it which creates a record of the event in our database.

5. Work Distribution

Each of us contributed to the success of WineSquare.

Stefan, who is the member of the group with backend experience in PHP, was the backend workhorse, showing us how to navigate PHP syntax, designing the check-in functionality and setting up the database information from FreeBase. Stefan took the lead on the PHP for this project.

Jason handled creation of the badges, the design of the wine page, the Google Maps APIs, and image manipulation across the application. Jason worked on the HTML, CSS, some of the PHP and Photoshop work.

Boris worked on the recommendation system, UI design, history visualization, signups, location detection (Google maps API), and profile editing/deletion. He worked on the HTML, CSS, JQuery, and some of the PHP.

Nathan partnered with Boris on UI, creating many of the HTML/CSS pages, and headed dynamic page creation of all users. He worked on HTML, CSS, and some of the PHP.

6. Conclusion

We had a blast designing WineSquare & creating a working application using skills we've gained this semester. Thanks for giving us the freedom to choose our project and we hope that you enjoy it. Come drink with us!

Sincerely,

The WineSquare Team