



Lập trình hướng đối tượng

THỊ TRẦN THÚ CƯƠNG

Nguyễn Tuyết Ngân – Phạm Văn Hưng | K57 Máy tính và Khoa học thông tin | January 1, 2014

Lời cảm ơn

Chúng em xin chân thành cảm ơn những bài giảng bổ ích và chi tiết về lập trình hướng đối tượng trên lớp của thầy Hiếu, cũng như những giờ thực hành của thầy Hùng và thầy Chí, đặc biệt là thầy Chí vì đã luôn tận tình chỉ bảo và trả lời những câu hỏi của chúng em trong suốt quá trình thực hiện.

Mục lục

I. Phát biểu bài toán	4
1. Bài toán	4
2. Các đối tượng cần quản lý	4
3. Các chức năng người chơi có thể sử dụng	4
4. Các chức năng chương trình có thể quản lý	5
II. Nội dung	6
1. Mục tiêu báo cáo	6
2. Phạm vi bài toán	6
3. Tài liệu tham khảo	6
4. Cấu hình tối thiểu	6
III. Phân tích và thiết kế	8
1. Chức năng và nhiệm vụ của hệ thống	7
2. Cơ chế	7
3. Phân tích Cơ sở dữ liệu và các đối tượng của chương trình	8
1. Người chơi	8
2. Loại thú cưng	9
3. Thú cưng	9
4. Túi vật phẩm	10
4. Phân tích chi tiết các chức năng	11
1. Đăng ký	11
2. Đăng nhập	12
3. Bán đồ	13
4. Nhà riêng	14
5. Trung tâm thú cưng	15
a. Mua mới	16
b. Mua mới	16
6. Nhà của thú cưng	17
7. Cài đặt	18
8. Túi cá nhân	19
9. Cửa hàng	20
10. Trung tâm trò chơi	20

11. Quán đồ uống	21
12. Chương trình quản lý	22
IV. Các hướng phát triển, mở rộng	23
1. Thêm các chức năng cho hệ thống hiện tại.....	23
2. Mở rộng	23
Kết luận.....	24

I. Phát biểu bài toán

1. Bài toán

Trên Internet, không khó để tìm kiếm hình ảnh liên quan đến một con vật, hay một nhân vật trong truyện tranh, trong phim hoạt hình. Với một kho hình ảnh vô cùng lớn và dễ thương, đôi khi người thích chúng cũng muốn có một nhân vật cho riêng mình để chăm sóc.

Xuất phát từ ý tưởng đó, dựa vào kho hình động của các nhân vật trong truyện tranh, phim hoạt hình Nhật Bản vô cùng đồ sộ tại vnsharing.net, chúng em đã quyết định viết nên một chương trình để biến những hình động này thành thú cưng của riêng mỗi người chơi trên nền một trò chơi riêng biệt. Đó chính là “**Thị trấn thú cưng**”.

Nội dung của trò chơi gồm có các chức năng chính:

- _ Mua bán các thú cưng.
- _ Nuôi thú: cho ăn để các thú cưng của mình lên level, tắm rửa, đổi tên, nựng thú để tăng chỉ số vui vẻ.
- _ Lưu trữ và cho phép người chơi thay đổi thông tin cá nhân.
- _ Cho phép người chơi tham gia vào các minigame để kiếm thêm tiền mua thú cưng và vật phẩm.

Ngoài ra, còn có chương trình quản lý nằm ngoài trò chơi dành cho người quản trị để xem và thay đổi thông tin của các người chơi, cũng như xuất các report cần thiết.

2. Các đối tượng cần quản lý

- _ Người chơi
- _ Các thú cưng của người chơi đó
- _ Các vật phẩm mà người chơi có
- _ Các loại thú cưng

3. Các chức năng người chơi có thể sử dụng

- _ Đăng ký tài khoản mới.
- _ Đăng nhập trò chơi.
- _ Thay đổi thông tin cá nhân.

- _ Mua và bán các thú cưng.
- _ Mua và bán các vật phẩm của game.
- _ Chăm sóc thú cưng.
- _ Chơi các minigame để kiếm tiền mua thú cưng và vật phẩm.

4. Chức năng của chương trình quản lý

- _ Xem, thay đổi thông tin người chơi và xóa một người chơi.
- _ Xem danh sách các thú cưng của từng người chơi, gọi về các thú cưng đã bỏ đi của người chơi.
- _ Xuất ra report.

II. Nội dung

1. Mục tiêu báo cáo

- _ Mô tả bài toán, chi tiết các chức năng và cách thức hoạt động của chương trình.
- _ Cho người đọc cái nhìn tổng quát về đề tài.

2. Phạm vi bài toán

- _ Trò chơi nuôi thú cưng dành cho các người chơi và chương trình quản lý đơn giản cho phép xuất ra report.

3. Tài liệu tham khảo

- [1] Trần Trọng Hiếu, *Các slide bài giảng trên lớp*, ĐHKHTN – ĐHQGHN, 2013.
- [2] Ngô Văn Chí, *Các bài giảng thực hành và chương trình ví dụ*, ĐHKHTN – ĐHQGHN, 2013.
- [3] <http://docs.oracle.com/javase/tutorial>
- [4] <http://stackoverflow.com>
- [5] <http://sourceforge.net/>, 2 class **AnimatedIcon** và **GifCoder**.

4. Cấu hình tối thiểu

a) Phần mềm

- _ Hệ điều hành: Windows hoặc Linux.
- _ Java 6.
- _ Microsoft SQL Server 2005.

b) Phần cứng

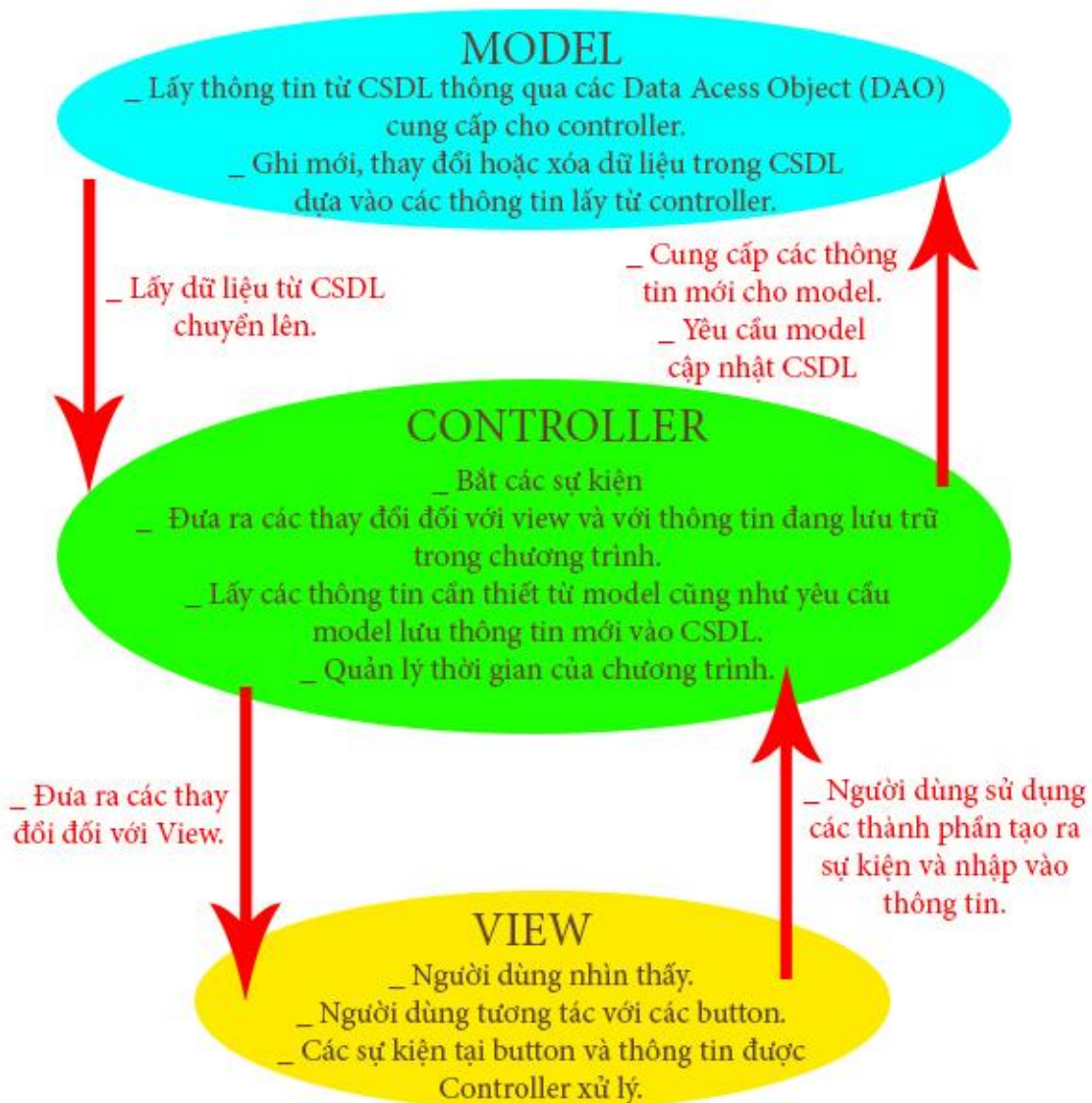
- _ CPU: Celeron D 2.04GHz.
- _ RAM: 512MB
- _ Ổ cứng: 10GB.

III. Phân tích và thiết kế

1. Chức năng và nhiệm vụ của hệ thống

- _ Cung cấp một giao diện đồ họa cho phép người chơi sử dụng các chức năng của trò chơi.
- _ Lưu trữ những thay đổi vào cơ sở dữ liệu.

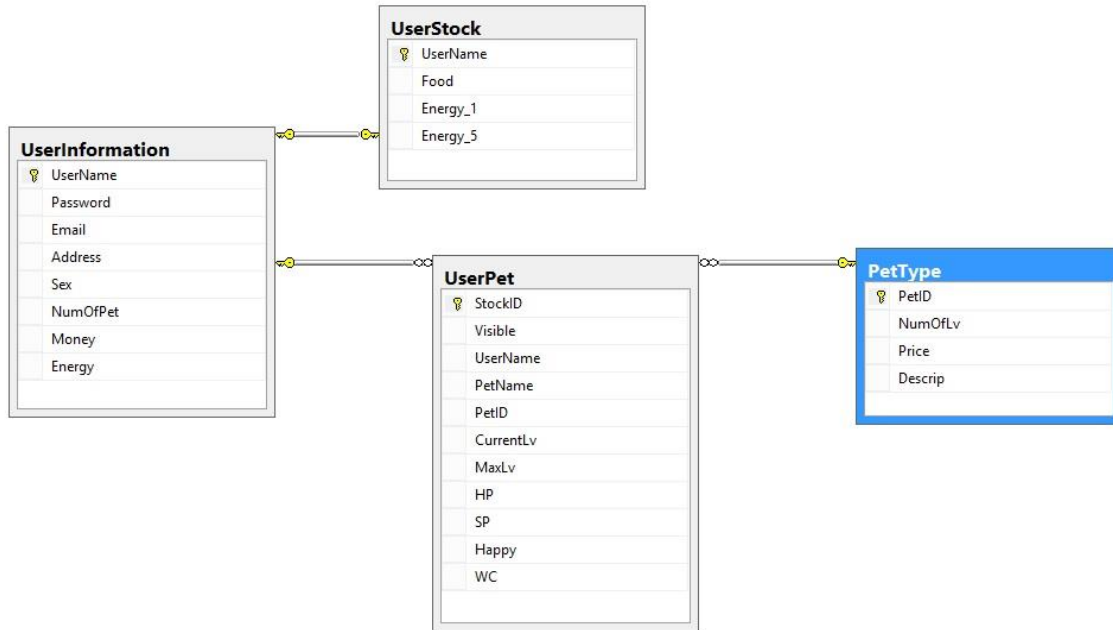
2. Cơ chế



Hình 1. Hệ thống hoạt động theo mô hình MVC

Bằng mô hình MVC, các lớp trong chương trình được chia ra theo từng chức năng tương ứng: Các lớp trong View chứa phần đồ họa để hiển thị, Model chứa Connection, các DAO, các lớp mang phép toán logic ứng với từng class trong View và các đối tượng của chương trình, Controller chứa phần điều khiển và xử lý tương ứng.

3. Phân tích Cơ sở dữ liệu và các đối tượng của chương trình



Hình 2. Mô hình thực thể

Chương trình làm việc với 4 đối tượng chính:

1. Người chơi (tương ứng với **UserInformation**).
2. Loại thú cưng (tương ứng với **PetType**).
3. Thú cưng (tương ứng với **UserPet**).
4. Túi vật phẩm của người chơi (tương ứng với **UserStock**).

1. NGƯỜI CHƠI

Lớp người chơi mang theo thông tin tương ứng về từng người chơi:

_ **UserName**: có dạng chuỗi kí tự, do người chơi tự chọn khi đăng ký tài khoản. Tên đăng nhập không thể thay đổi và là khóa tương ứng trong bảng **UserInformation**.

_ **Password**: có dạng chuỗi kí tự, do người chơi đặt để đăng nhập hệ thống, người chơi có thể chỉnh sửa mật khẩu sau khi đã đăng nhập thành công vào game.

_ **Email, Address, Sex**: các thông tin cá nhân có thể để trống, tương ứng có kiểu chuỗi kí tự với Email và địa chỉ, kiểu logic với giới tính (true: giới tính Nam, false: giới tính Nữ).

_ NumOfPet: Số lượng thú cưng người chơi có, do hệ thống tự động cập nhật, có kiểu số nguyên.

_ Money: Có kiểu số nguyên, là tiền trong game dùng để mua thú cưng và các vật phẩm. Người chơi không thể tự chỉnh sửa số tiền của mình mà khi đăng ký tài khoản sẽ được cấp cho một số tiền mặc định. Người chơi có thể kiếm thêm tiền thông qua việc chơi các minigame. Controller sẽ quản lý số tiền này và thay đổi nó cũng như yêu cầu Model lưu lại vào CSDL tùy theo hoạt động của người chơi.

_ Energy: Có kiểu số nguyên, mỗi người chơi sẽ có một chỉ số năng lượng mà tối đa là một giá trị cố định của chương trình. Khi chơi các minigame, tùy vào nội dung game, người chơi sẽ tốn một phần năng lượng tương ứng. Năng lượng này sẽ tự động tăng lên theo thời gian, tùy vào số thú cưng mà người chơi có.

Trong chương trình, thông tin người chơi được miêu tả bởi lớp **User**, lớp này có các thuộc tính tương ứng với các cột trong CSDL và các phương thức cho phép controller gán, thay đổi và lấy các giá trị từ thuộc tính của lớp này. Riêng Energy có thêm các phương thức cho phép tăng hoặc giảm 1, khi tăng hoặc giảm sẽ kiểm tra điều kiện để giá trị không vượt quá mức cho phép hoặc nhỏ hơn 0.

2. LOẠI THÚ CƯNG

Loại thú cưng mang theo thông tin của từng loại thú cưng:

_ PetID: số hiệu phân biệt loại thú cưng, có kiểu số nguyên.

_ NumOfLv: số cấp độ mà loại thú cưng đó có, có kiểu số nguyên.

_ Price: giá tiền của loại thú cưng, có kiểu số nguyên.

_ Descrip: vài dòng miêu tả về loại thú cưng đó, có kiểu chuỗi kí tự.

Trong chương trình, thông tin loại thú cưng được miêu tả bởi lớp **PetType**, lớp này có các thuộc tính tương ứng với các cột trong và các phương thức cho phép controller gán, thay đổi và lấy các giá trị từ thuộc tính của lớp này.

Các hình ảnh về từng loại thú cưng được lưu trong thư mục của project mã nguồn, đường dẫn có dạng "...\\petID\\level.gif", trong đó mỗi loại thú cưng sẽ có một thư mục riêng mang tên PetID tương ứng, trong đó chứa các file tương ứng với từng cấp độ của loại thú cưng, đánh số từ 1 cho tới NumOfLv.

3. THÚ CƯNG

Mỗi thú cưng sẽ mang theo thông tin và các chỉ số.

_ StockID: mã số của thú cưng, dùng để phân biệt mỗi thú cưng là duy nhất. StockID có kiểu số nguyên, do SQL tự tăng lên mỗi lần một thú cưng mới được thêm vào bảng.

_ Visible: trạng thái, cho biết thú cưng này còn thuộc về Người chơi hay đã bỏ đi, có kiểu logic.

- _ **UserName**: tên người chơi sở hữu thú cưng này, có kiểu chuỗi kí tự.
- _ **PetName**: tên của thú cưng này, có kiểu chuỗi kí tự.
- _ **PetID**: số hiệu của loại thú cưng tương ứng, có kiểu số nguyên.
- _ **CurrentLv**: cấp độ hiện tại của thú cưng. Tùy vào cấp độ hiện tại, hình ảnh của cấp độ sẽ được hiển thị, có kiểu số nguyên.
- _ **MaxLv**: có giá trị đúng bằng **NumOfLv** của loại thú cưng tương ứng, có kiểu số nguyên.
- _ **HP**: chỉ số HP của thú cưng, có kiểu số nguyên, giảm xuống theo mỗi lần đăng nhập của người chơi. Khi người đăng nhập 5 lần liên tục mà không cho thú cưng này ăn, thú cưng sẽ bỏ đi (thuộc tính **Visible** thay đổi). HP sẽ tăng lên mỗi lần người chơi cho thú cưng ăn. Nếu HP đã đạt giá trị tối đa thì khi cho ăn sẽ không tăng nữa.
- _ **SP**: chỉ số biểu thị mức độ tiến hóa của thú cưng, có kiểu số nguyên. Khi thú cưng chưa đạt tới **MaxLv** thì mỗi lần SP đạt tới các giá trị 10, 20, 30, 40, **CurrentLv** của thú cưng sẽ tăng.
- _ **WC**: chỉ số vệ sinh của thú cưng, có kiểu số nguyên. Mỗi lần người chơi cho thú cưng ăn, chỉ số vệ sinh sẽ tự động tăng. Khi chỉ số vệ sinh đạt max mà người chơi vẫn tiếp tục cho ăn thì chỉ số vui vẻ sẽ giảm.
- _ **Happy**: chỉ số vui vẻ của thú cưng, có kiểu số nguyên. Khi người chơi nựng thú cưng, chỉ số vui vẻ sẽ tăng. Khi chỉ số vui vẻ về 0, thú cưng sẽ bỏ đi.

Trong chương trình, thông tin người chơi được miêu tả bởi lớp **UserPet**, lớp này có các thuộc tính tương ứng với các cột trong CSDL và các phương thức cho phép controller gán, thay đổi và lấy các giá trị từ thuộc tính của lớp này. Riêng các chỉ số có thêm các phương thức cho phép tăng hoặc giảm 1, khi tăng chỉ số sẽ đồng thời kiểm tra điều kiện để tăng các chỉ số khác của thú cưng, đồng thời kiểm tra điều kiện để các chỉ số không vượt quá mức cho phép hoặc nhỏ hơn 0.

4. TÚI VẬT PHẨM

Mỗi túi vật phẩm sẽ mang thông tin về các vật phẩm người chơi có.

- _ **UserName**: tên người chơi sở hữu túi vật phẩm, có kiểu chuỗi kí tự.
- _ **Food**: số thức ăn cho thú cưng mà người chơi có, có kiểu số nguyên.
- _ **Energy_1**: số vật phẩm tăng 1 Energy mà người chơi có, có kiểu số nguyên.
- _ **Energy_5**: số vật phẩm tăng 5 Energy mà người chơi có, có kiểu số nguyên.

Trong chương trình, thông tin người chơi được miêu tả bởi lớp **UserPet**, lớp này có các thuộc tính tương ứng với các cột trong CSDL và các phương thức cho phép controller gán, thay đổi và lấy các giá trị từ thuộc tính của lớp này.

4. Phân tích chi tiết các chức năng

Mỗi màn hình hiển thị trong chương trình sẽ có 1 lớp View tương ứng để viết phần hiển thị, và 1 lớp Controller để viết các sự kiện, tùy theo chức năng sẽ có lớp Model để viết các phép toán logic và phần tương tác với DAO tương ứng.

DAO sẽ thông qua một Connection duy nhất để kết nối đến CSDL. Các View và Controller tương ứng với từng frame trong chương trình đều được tạo ra một lần duy nhất dưới dạng static, sử dụng Singleton Coding Pattern để sử dụng một View duy nhất cho mọi lần hiển thị tới màn hình đó, nhằm tiết kiệm bộ nhớ và giảm thời gian tải game.

1. ĐĂNG KÝ

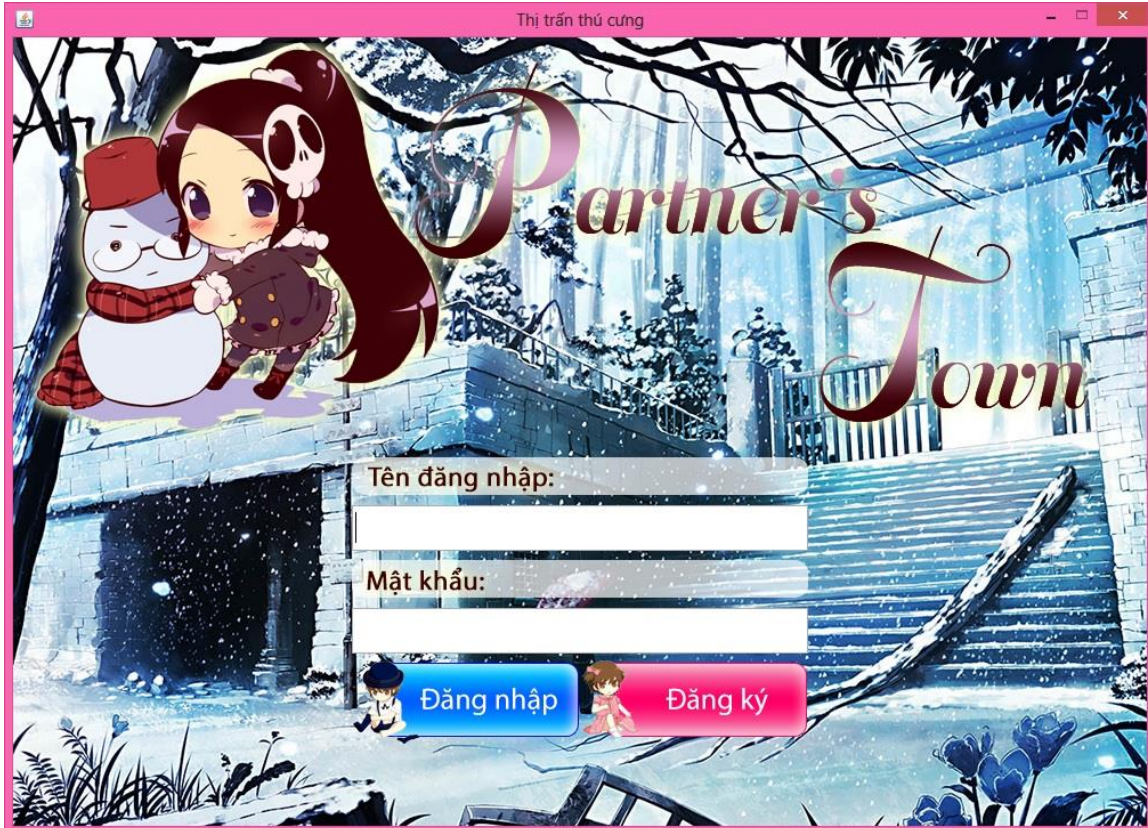
Hình 3. Đăng ký

Người chơi có thể mở cửa sổ đăng ký từ cửa sổ đăng nhập đầu tiên. Khi bấm “Nhập lại”, các trường sẽ được xóa trắng. Khi người chơi bấm “Hoàn tất”, các điều kiện sau sẽ được kiểm tra:

- _ Các trường ở mục Thông tin tài khoản không được bỏ trống.
- _ Tên đăng nhập này không trùng với các tên đăng nhập có sẵn trong CSDL.
- _ Tên đăng nhập không chứa dấu cách và phải nằm trong khoảng từ 6 đến 32 ký tự.
- _ Mật khẩu không chứa dấu cách và phải nằm trong khoảng từ 6 đến 16 ký tự.
- _ Mật khẩu và mật khẩu gõ lại phải trùng nhau.

Nếu các điều kiện đều thỏa mãn, một đối tượng **User** và một đối tượng **UserStock** mới sẽ được tạo, model sẽ thêm User này vào bảng **User Information**, thêm UserStock vào **UserStock** và Dialog đăng ký sẽ tự biến mất, người chơi có thể đăng nhập bằng tài khoản vừa tạo. Ngược lại, nếu một trong số các điều kiện trên không thỏa mãn, chương trình sẽ đưa ra thông báo tương ứng.

2. ĐĂNG NHẬP



Hình 4. Đăng nhập

Màn hình đăng nhập sẽ cho phép người chơi nhập vào tên đăng nhập và mật khẩu. Controller sẽ lấy thông tin nhập vào, đưa cho model kiểm tra với dữ liệu lưu trong CSDL. Khi người chơi nhấn nút “Đăng nhập”, nếu đúng, các thông tin tương ứng với người chơi này, bao gồm User, UserStock và List<UserPet> sẽ được lưu vào một lớp **Information** trong Controller, trong đó bao gồm các thuộc tính tương ứng với các đối tượng, và màn hình sẽ chuyển sang bản đồ. Nếu sai, chương trình sẽ đưa ra thông báo.

Khi đăng nhập thành công, Controller sẽ bắt đầu một Timer được tạo ra để cứ 2 phút lại tăng Energy của người chơi 1 lần, người chơi không thể thay đổi hay truy cập tới Timer này.

Khi người chơi nhấn “Đăng ký”, Dialog đăng ký sẽ hiện ra.

3. BẢN ĐỒ



Hình 5. Bản đồ

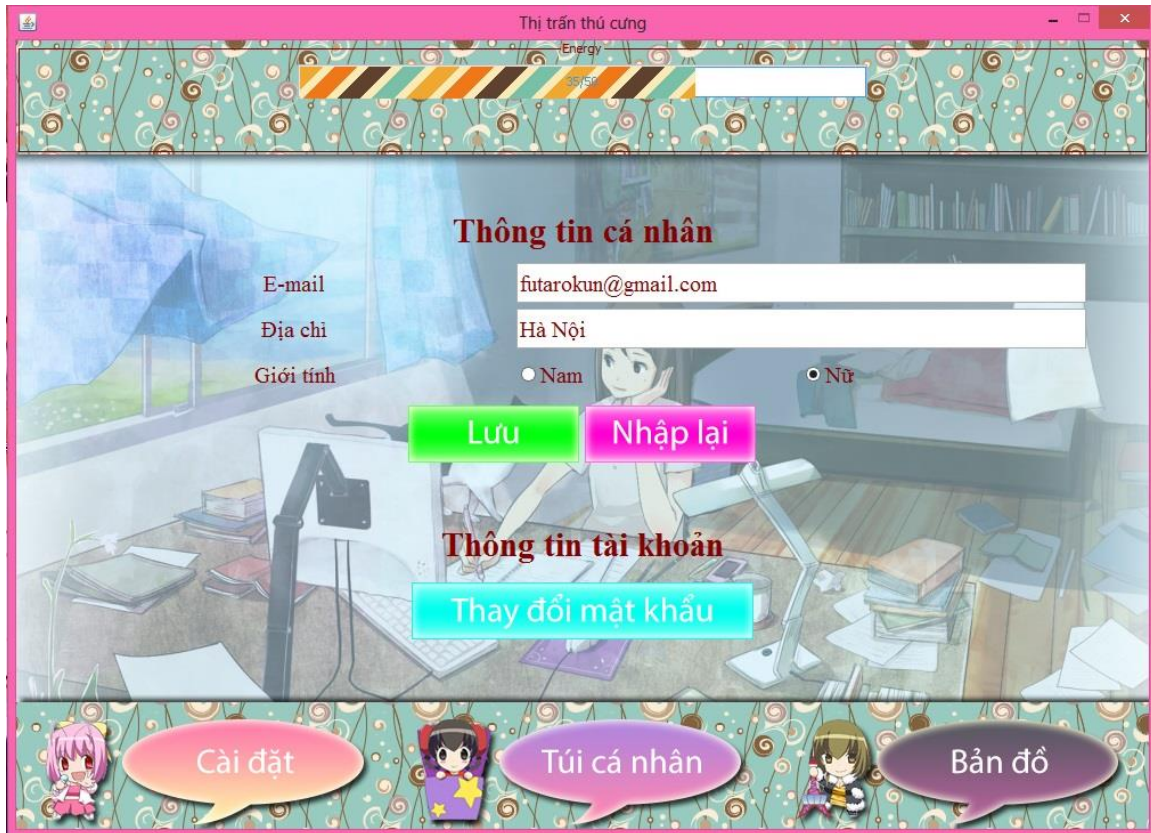
Trên màn hình sẽ gồm 7 thành phần chính:

_ Thanh Energy: thể hiện số Energy mà người chơi đang có.

_ Các nút dẫn tới các màn hình khác:

- + Home: thông tin cá nhân.
- + Partner Center: mua và bán thú cưng.
- + Partner's House: ngăn chứa thú cưng, cho phép người chơi chăm sóc thú cưng của mình.
- + Shop: nơi mua vật phẩm.
- + Game Center: chơi các minigame.
- + Drinks: hướng dẫn chơi game dành cho tân thủ.

4. NHÀ RIÊNG



Hình 6. Nhà riêng

- _ “Lưu”: lấy thông tin từ các textfield do người chơi nhập vào, lưu vào lớp Information. Model sẽ lấy thông tin từ Information để cập nhập vào CSDL.
 - _ “Nhập lại”: thêm thông tin từ lớp Information vào các textfield.
 - _ “Thay đổi mật khẩu”: Dialog “Thay đổi mật khẩu” sẽ xuất hiện.
 - _ “Cài đặt”: Dialog “Cài đặt” xuất hiện.
 - _ “Bản đồ”: Trở về bản đồ ban đầu.
- _ Dialog thay đổi mật khẩu:**
- + Kiểm tra xem mật khẩu hiện tại có đúng hay không.
 - + Kiểm tra xem mật khẩu mới có chưa dấu cách và nằm trong khoảng từ 6 đến 16 kí tự hay không.
 - + Kiểm tra xem mật khẩu nhập lại có trùng với mật khẩu mới hay không.



Hình 7. Thay đổi mật khẩu

Nếu 3 điều kiện trên đều đúng, nút “Lưu” sẽ lưu lại mật khẩu mới vào Information, và Model sẽ lấy thông tin từ Information cập nhật xuống cơ sở dữ liệu.

5. TRUNG TÂM THÚ CỪNG



Hình 8. Trung tâm thú cưng

- _ “Mua mới”: chuyển tới màn hình “Mua mới”.
- _ “Bán lại”: chuyển tới màn hình “Bán lại”.
- _ “Bản đồ”: quay về “Bản đồ” game.

a) Mua mới



Hình 9. Mua mới

Thông qua model, controller sẽ lấy thông tin về tất cả các loại thú cưng hiện có trong CSDL và hiển thị lên view.

_ “Mua”: Controller sẽ kiểm tra xem có dòng được chọn không và số tiền còn đủ để mua thú cưng đó hay không.

Nếu đúng, một đối tượng **UserPet** mới sẽ được tạo với giá trị trường “UserName” là username hiện tại đang lưu ở lớp Information và giá trị “PetID” là PetID của PetType đã được chọn. UserPet mới này sẽ được Model thêm vào bảng UserPet của CSDL, sau đó Information sẽ tải lại List<UserPet> của User theo danh sách model cập nhật lại từ CSDL.

Nếu sai, chương trình sẽ đưa ra thông báo.

_ “Trở lại”: Quay lại “Trung tâm Thú cưng”.

b) Bán lại

Controller sẽ lấy một List<UserPet> bao gồm các UserPet có thuộc tính trạng thái Visible = true từ List<UserPet> tại Information để hiển thị.

_ “Bán”: Nếu không có dòng nào được chọn, chương trình sẽ đưa ra thông báo. Nếu có, model sẽ xóa dòng có StockID trùng với StockID của dòng được chọn, sau đó controller sẽ cập nhật lại thông tin tại Information theo thông tin lấy từ CSDL, cộng tiền cho người chơi và thay đổi lại bảng đang hiển thị.



Hình 10. Bán lại

_ “Bản đồ”: trở lại “Trung tâm thú cưng”.

6. NHÀ CỦA THÚ CƯNG

Controller sẽ lấy từ Information một List các UserPet có giá trị Visible = true.

Mỗi tab của view hiển thị một thú cưng bao gồm các thành phần:

_ Hình ảnh: hiển thị hình ảnh của level hiện tại của thú cưng đó.

_ Tên của thú cưng.

_ SP.

_ Hình nền: tùy vào chỉ số WC của thú cưng, sẽ hiển thị hình nền tương ứng.

_ Emotion Icon: biểu tượng hiển thị tùy theo chỉ số Happy.

Các nút chức năng:

_ “Cho ăn”: Nếu người chơi còn thức ăn, controller sẽ trừ thức ăn của User trong Information, sau đó tăng SP, HP, WC và Happy của Pet và model sẽ cập nhật thay đổi vào view. Sau đó từ thông tin ở Information mà view sẽ đưa ra thay đổi tương ứng tùy theo chỉ số mới.



Hình 11. Nhà thú cưng

_ “Nựng”: tăng chỉ số Happy của thú cưng trong User ở Information, sau đó cập nhật lại CSDL và view.

_ “Tắm”: reset chỉ số WC của thú cưng về 0, lưu lại trong CSDL và thay đổi hình nền.

_ “Đổi tên”: Hiện Dialog đổi tên cho phép nhập vào tên mới của thú cưng, lưu lại vào CSDL và thay đổi tên đang hiển thị theo tên mới.

7. CÀI ĐẶT

Thông qua một biến logic của chương trình biểu diễn trạng thái của việc phát nhạc: có hoặc không, Setting Dialog sẽ hiển thị biểu tượng có nhạc/không nhạc tương ứng. Bấm vào Button này, người chơi có thể tắt nhạc hoặc bật lại nhạc nền của Game.

Ngoài ra, cửa sổ cài đặt cũng mang thông tin Credit của nhóm chúng em.



Hình 12. Cài đặt

8. TÚI VẬT PHẨM:

Dialog hiển thị số vật phẩm hiện có của người chơi, cho phép sử dụng 2 loại vật phẩm tăng Energy, nguyên lý hoạt động giống như các phần trước: Controller lưu thông tin vào Information và model cập nhật thông tin vào CSDL theo Information.



Hình 13. Túi vật phẩm

9. SHOP

Cho phép người chơi mua các vật phẩm, bằng cách bắt sự kiện bấm nút và kiểm tra điều kiện xem có đủ tiền hay không để tăng giá trị tương ứng của UserStock. Nếu không đủ tiền, chương trình sẽ đưa ra thông báo.



Hình 14. Cửa hàng

10. TRUNG TÂM TRÒ CHƠI

Thông qua việc kiểm tra điều kiện Energy có đủ hay không, chương trình sẽ mở các minigame tương ứng lên hoặc đưa ra thông báo. Trong các minigame sẽ có nút cho sự kiện tăng tiền cho người chơi khi trò chơi kết thúc.

Tại Game Center có 3 minigame:

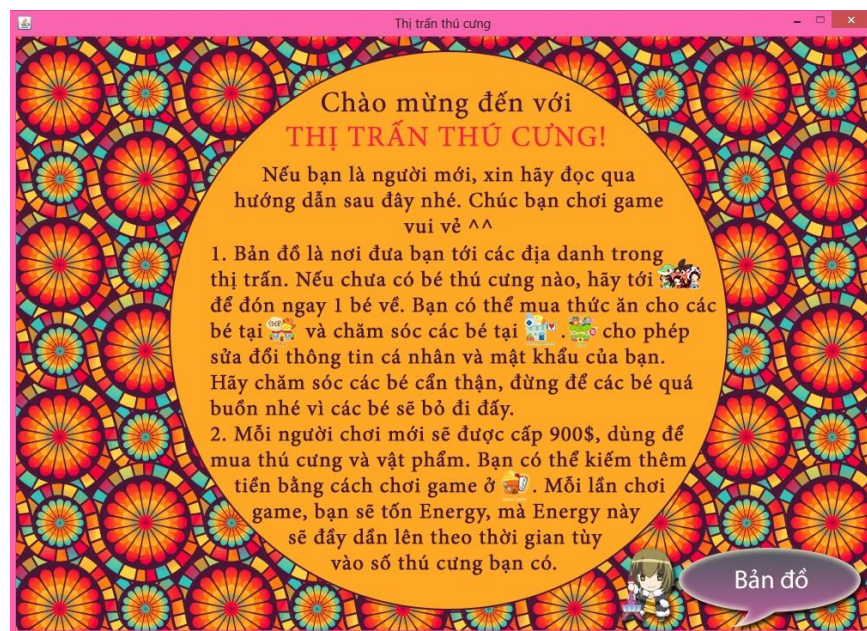
- _ Flip Flop: trò chơi tìm cặp giống nhau.
- _ Fruit Plus: trò chơi tính cộng.
- _ Toilet Rush: đưa người xếp hàng vào đúng phòng cần thiết.



Hình 15. Trung tâm trò chơi

II. QUÁN ĐỒ UỐNG

Hiển thị hướng dẫn thu gọn dành cho người chơi.



Hình 16. Quán đồ uống

12. CHƯƠNG TRÌNH QUẢN LÝ

Là một Project tách biệt, hoạt động theo nguyên lý tương tự như trò chơi chính và cho phép xuất ra report cần thiết.



Hình 17. Màn hình chính của chương trình quản lý

IV. Các hướng phát triển, mở rộng

Thời gian và khả năng có hạn không cho phép chúng em mở rộng phần mềm nhiều hơn, nhưng sau đây là một số hướng phát triển:

1. Thêm các chức năng cho hệ thống hiện tại

- _ Cho phép người chơi trả tiền để nhận lại thú cưng đã bỏ đi.
- _ Thêm các minigame và loại thú cưng mới, các loại thú cưng đặc biệt chỉ cho phép mua khi đạt một số điều kiện, ...
- _ Thêm các tính năng và chỉ số mới cho thú cưng, mở thêm phòng tập để tăng các chỉ số.

2. Mở rộng để sử dụng game trong mạng LAN

- _ Thay đổi cài đặt Connection để người chơi trong cùng mạng LAN có thể truy cập tới CSDL ở một máy chủ.
- _ Thêm các chức năng: hòm thư, thêm bạn, cho phép các người chơi khác nhau có thể liên lạc với nhau.

Với hệ thống hiện tại, có rất nhiều các khác nhau để phát triển, nâng cấp, nhưng trên đây là một số hướng khả thi để thực hiện mà chúng em muốn làm khi có thêm thời gian. Mong muốn lớn hơn nhưng rất khó thực hiện, là tới một lúc nào đó có thể phát triển đề tài đầu tay của mình thành 1 Game Online thực thụ có lẽ cần nhiều thời gian và kiến thức hơn nữa, mà chúng em sẽ cần phải cố gắng thật nhiều và không ngừng học hỏi để có thể làm được.

Kết luận

Thời gian làm đề tài này đã cho chúng em rất nhiều kinh nghiệm bổ ích, cũng như nhiều cảm xúc khác nhau: hào hứng khi thấy code của mình chạy tốt, thấy hạnh phúc khi nghĩ ra những cách giải quyết cho vấn đề của mình và choáng ngợp vì những thứ Java có thể làm, cũng như cả những cảm xúc buồn, bức bối và mất kiên nhẫn khi gặp lỗi, hay những khi code không hoạt động như mình đã dự đoán phải tìm sự trợ giúp từ Internet và từ các thầy, tuy nhiên đó là những lúc học hỏi được nhiều nhất.

Chúng em không thể nào hoàn thành được đề tài này nếu như không có những bài giảng và sự chỉ dẫn tận tình của thầy, cũng như sự giúp đỡ và cổ vũ của bạn bè trong suốt thời gian vừa qua.

Thời gian làm đề tài cũng như bài làm cũng chỉ vừa kết thúc, tuy nhiên đó mới chỉ là khởi đầu cho những quãng đường khó khăn nhưng cũng tràn đầy những điều mới mẻ hấp dẫn kế tiếp. Chúng em rất mong nhận được những lời góp ý và chỉ dẫn nghiêm khắc nhưng tận tình để có thể hoàn thiện đề tài và kiến thức của bản thân nhiều hơn nữa, cũng như mong chờ tới những đề tài tiếp theo.

Một lần nữa, chúng em xin chân thành cảm ơn.