SOFTWARE REQUIREMENTS SPECIFICATION FOR THE “VAPOR” PROGRAM

By Kyle Harvey

Introduction

Purpose

The Vapor program is a game archive submission platform to allow the submission of games to be archived much easier. Publishers of games that would like their game to be archived for historical purposes can use the Vapor program to submit their game for submission, edit their submissions, remove previous submissions and view the current list of submissions by them and others.

Program Description

The Vapor program is used by archiving services in the realm of games to quickly manage large collections of games both in terms of maintaining file structure of the archive itself and in terms of keeping track of the information associated with each program. Submitters of each game must have an account to submit and all information for a submission is associated with that account. There will also be customers in case a publisher of a game wants to have the game available for sale and the customer account will also allow the Vapor program to keep track of downloads concerning the free games also.

User Classes and Characteristics

### Customer Account:

Used to manage an account for a customer who might want to buy/get a game from the archive.

### Submitter Account:

Used to manage the information for each submitter of a game.

### Game:

Used to manage the information for each submission.

### Sysadmin:

Used internally in the program itself to handle major functionality of the archive itself. File operations and statistics retrieval etc.

Operating Environment

Desktop environment, specifically windows.

Design and Implementation Constraints

Simple interactive gui. Hidden behind a login screen for security reasons and to manage permissions easier.

# External Interface Requirements

User Interfaces

### Login Screen:

Will allow user to either login with a registered account (Customer or Submitter account located in the database) or take user to a register account screen to create a new account

### CRUD:

For account management and submission management there will be individual forms/windows that allow user to Create Read Update and Delete relevant records.

Software Interfaces

The program will directly connect to running SQL server service to manage the database on host computer. Connection string info will have to be stored outside program for easy configuration. Preferably not stored at all, for security purposes connection will have to occur through start-up page.

Domain Model

The domain model contains the “User”, the “Vapor” program, the “SQL Database” as well as the “Archive of Games” which is the filesystem containing the games organized into relevant file structure.

System Features (Use Cases)

In general, there are 3 main Use case categories. (1.) There is the sysadmin who will manage the files, accounts and database. (2.) There is the Submitter who manages their own submissions to the archive and some basic account management in regards their account info. (3.) And there is the Customer who may access the archive of Games to request a copy of a game or buy a copy if the game is priced.

### (1.) Sysadmin

1. Add/Delete accounts from database records.
2. Add/Delete games from database records.
3. Edit the associated info for an account.
4. Edit the associated info for a game.
5. Display info for multiple accounts or a single account.
6. Display info for multiple games or a single game.

### (2.) Submitter

1. Add/Delete games from database records.
2. Edit the associated info for submitters account.
3. Edit the associated info for submitters game.
4. Display info for All or one of Submitters games.

### (3.) Customer

1. List available games and read info about each game.
2. Use Vapor to buy or get for free any game available.
3. Game will be provided through a URL or file path name

# Other Non-functional Requirements

Performance Requirements

Must be scalable to 1000’s or more games/accounts with easy to adjust code in case requirements change.

Security Requirements

Protections against SQL injection, weak password, and plain text connection or other info being stored in product

# Other Requirements

1. The “Vapor” program should have at least 3 Forms / Windows or you could have one window with multiple tabs.
2. the “Vapor” program database design should include at least 2 tables with primary and foreign key relationship
3. The project needs to be well tested and robust in operation