Project #4: Warehouse Management System

Team: Can You C My Screen?

CSCI 1110-01: Intro to C Programming

Dr. Reza Sadeghi

Thursday, April 28th, 2021

1

Outline

Team Description

Arely J. Parra López - Allowing the user to view, request, and save items from the Warehouse Management System as well as providing the admin and guest user an exit function.

Alexander Vita - Adding, deleting, and editing items with varied details (i.e., Type, Stored Time, ID, etc) from Warehouse Management System.

Kamryn Hammond - Creating login page for admin and guest users to enter username & password for Warehouse Management System.

Team Description (cont.)

Jarred Crystal - Generating user-friendly software that provides a welcome page, a menu of all functions that users have access to, and tabular format of all requested information from the Warehouse Management System.

Sophie Ross - Allowing admin to view the list of borrowing requests as well as accept/reject borrowing requests made by guest users in the Warehouse Management System.

System Description

◎ **Greeted by a WMS Welcome Page with registration or login.**

◎ **If logging in as an Admin, individual is able to:**
- View a list of usernames & passwords
- Search for usernames & passwords in the system
- Change usernames/passwords for admin & guest
- Add/Remove admin & guest
- Add/Remove/Edit items from the WMS
- View/Search for items from the WMS
- View list of borrowing requests from guest users
- Accept/Deny borrowing requests from guest users

System Description (cont.)

◎ **If logging in as Guest User, individual is able to:**
  - ○ Search for items in the WMS
  - ○ Make a list of favorite items from the WMS
  - ○ Request items to borrow/buy
  - ○ View item's borrowing history

◎ **All users can exit from the WMS when in the Login Menu, Admin Menu, or the Guest Menu.**

Functions Description

Crystal:

◎ **int loginMenu(void)**
- Allows for new registration of users and the selection of either the admin or guest login prompt

◎ **void main(void)**
- Provides the main interface to the login menu dictating which function will be executed following user input

◎ **int adminMenu(void)**
- Provides the menu options within the admin menu

◎ **void admin(void)**
- Provides the main interface to the admin menu dictating which function will be executed following user input

Functions Description (cont.)

Crystal:

◎ **int guestMenu(void)**
  ○ Provides the menu options within the guest menu
◎ **void guestMain(void)**
  ○ Provides the main interface to the guest menu dictating which function will be executed following user input

Functions Description (cont.)

Hammond:

◎ **int* adminLogin(void)**
  - ○ Allows admin to search for their login from the WMS inventory.

◎ **int* guestLogin(void)**
  - ○ Allows guest users to search for their login from the WMS inventory using their private encryption key.

◎ **void view(void)**
  - ○ Provides an organized view of all the Admin & Guest User's encryption keys, usernames, and passwords.

◎ **int* search(void)**
  - ○ Allows the Admin to search the WMS LogIn.txt file for a specific encryption key, username, and password.

Functions Description (cont.)

Hammond:

◎ **void edit(void)**
  ○ Allows the Admin to select a specific user from the WMS and edit their login information.

◎ **void add(void)**
  ○ Allows the Admin to add a new user to the WMS with an encryption key, username, and password.

◎ **void delete(void)**
  ○ Allows the Admin to select and delete a user from the WMS.

Functions Description (cont.)

Vita:

◎ **void addItems (void)**
- ○ Admin can add an item to the inventory

◎ **int* searchItems(void)**
- ○ Admin can search for an item within the inventory

◎ **void editItems (void)**
- ○ Admin can edit an items characteristics

◎ **void deleteItems(void)**
- ○ Admin can delete an item from the inventory

◎ **void viewItems(void)**
- ○ Admin can view all items within the inventory

Functions Description (cont.)

Ross:

◎ **Void acceptDeny(void)**
  ○ Allows Admin to accept or deny borrowing request list from guest users.
◎ **Void viewBorrowing(void)**
  ○ Allows Admin to view all borrowing requests from guest users.

Functions Description (cont.)

Parra López:

◎ **void registration(void)**
- New individuals can create a new accounts within the WMS.

◎ **int* searchWMS(void)**
- Guest can search for an item from the WMS inventory.

◎ **void makeList(void)**
- Guest can make a list of favorite items from the WMS inventory.

◎ **void viewList(void)**
- Guest can view their list of favorite items.

◎ **void requestItems(void)**
- Guest can request to borrow/buy an item.

◎ **void viewHistory(void)**
- Guest can view the borrowing/purchase history of items.

Variables Description

Crystal:

◎ **int adminMenu(void)**
- ○ int input;

◎ **void admin(void)**
- ○ int option;
- ○ char quit;

◎ **int loginMenu(void)**
- ○ int input;

◎ **int guestMenu(void)**
- ○ int input;

◎ **void guestMain(void)**
- ○ int task;
- ○ char quit;

◎ **void main(void)**
- ○ int choice
- ○ char quit;

Variables Description (cont.)

Hammond:

◎ **int* adminLogin(void)**
◎ **int* guestLogin(void)**
◎ **int* search(void)**
  ○ static int output[2]
  ○ char key[100]
  ○ char input[100]
  ○ FILE *myfile
  ○ char myString[100]
  ○ int line=0
  ○ int find=0
  ○ int fieldNumber=0
  ○ char quit

◎ **void view(void)**
  ○ struct entity
  ○ char encrypt[20]
  ○ char username[20]
  ○ char password[20]
  ○ struct entity new
  ○ char myString[100]
  ○ int pline=0

Variables Description (cont.)

Hammond:

◎ **void edit(void)**
- int* place=search()
- int line=place[1]
- int find=place[0]
- char username[20], password[20]
- FILE *invfile
- FILE *temp
- char myString[100]
- int pline=0
- char quit

◎ **void add(void)**
- struct entity
- char encrypt[20]
- char username[20]
- char password[20]
- struct entity new
- FILE *myfile
- char quit

Variables Description (cont.)

Hammond:

◎ **void delete(void)**
- ○ int* place=search()
- ○ int line=place[0]
- ○ int find=place[0]
- ○ FILE *originalfile
- ○ FILE *tempfile
- ○ char myString[100]
- ○ int pline=0
- ○ char quit

Variables Description (cont.)

Vita:

◎ **void addItems (void)**
- struct entity
- char Type...[100]
- struct entity new
- FILE *invfile
- char quit

◎ **int* searchItems(void)**
- static int output[2]
- char key[100]
- char input[100]
- FILE *invfile
- char myString[100]
- int line=0
- int find=0
- int fieldNumber=0
- int pline=0

Variables Description (cont.)

Vita:

◎ **void editItems (void)**
- ○ int* place=searchItems()
- ○ int find = 0
- ○ int line = 0
- ○ char Type….[100]
- ○ FILE *invfile
- ○ FILE *tempfile
- ○ char myString[100]
- ○ int pline=0
- ○ char quit

◎ **void deleteItems(void)**
- ○ int* place=searchItems()
- ○ int find=place[0]
- ○ int line=place[1]
- ○ FILE *invfile
- ○ FILE *tempfile
- ○ char myString[100]
- ○ int pline=0
- ○ char quit

Variables Description (cont.)

Vita:

◎ **viewItems(void)**
- ○ FILE *invfile
- ○ struct entity
- ○ char Type….[100;
- ○ struct entity new
- ○ char Type….[100];
- ○ char myString[100000]
- ○ int pline=0

Variables Description (cont.)

Ross

- **void viewBorrowing(void)**
  - FILE myfile*
  - struct entity
  - char Type...[100]
  - struct entity new
  - char myString[10000]
  - int pline = 0

- **void acceptDeny(void)**
  - struct entity
  - char ID[20]
  - char accept[20]
  - char deny[20]
  - FILE myfile*
  - char quit

Variables Description (cont.)

Parra López:

◎ **void registration(void)**
- struct entity
- char encrypt[20]
- char username[20]
- char password[20]
- struct entity new
- FILE *myfile
- char quit

◎ **int* searchWMS(void)**
- static int output[2]
- char ID[100]
- char input[100]
- FILE *myfile
- char myString[100]
- int line=0
- int find=0
- int fieldNumber=0
- char quit

Variables Description (cont.)

Parra López:

◎ **void viewList(void)**
◎ **void viewHistory(void)**
  - FILE *myfile
  - struct entity
  - char Type...[100]
  - struct entity new
  - char myString[10000]
  - int pline=0

◎ **void makeList(void)**
◎ **void requestList(void)**
  - struct entity
  - char Type...[100]
  - char Length[100]
  - struct entity new
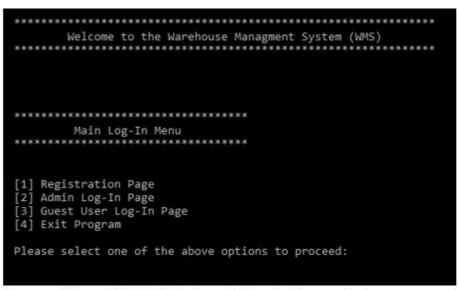 - FILE *myfile
  - char quit

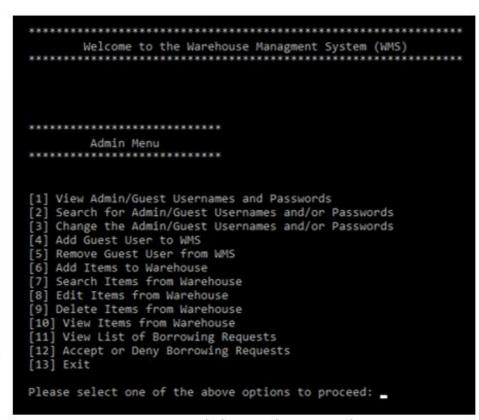# Figures

Figure 7: WMS Main Log-in Menu in Command Prompt

Figure 8: WMS Admin Menu in Command Prompt

Figures (cont.)

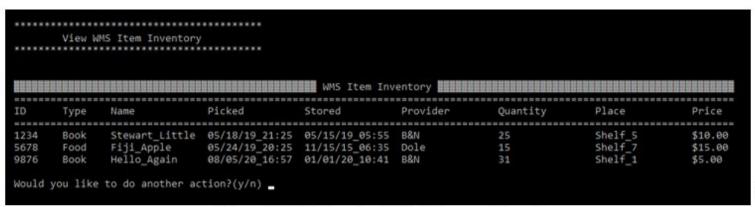*Figure 9:* WMS Admin/Guest User Encryption Key, Username, & Password Table in Command Prompt

Figure 10: WMS Inventory Table in Command Prompt

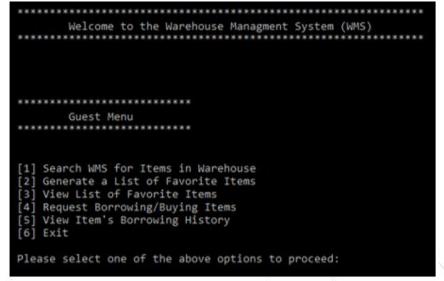*Figure 11:* WMS Guest User's Borrowing Requests Table in Command Prompt

Figures (cont.)

Figure 12: WMS Guest User Menu in Command Prompt

Figure 13: WMS Guest User's Favorite Items List Table in Command Prompt

Figure 14: WMS Guest User's View of Item History Table in Command Prompt

References

1. Fischer, A. E., Fischer, M. J., & Eggert, D. W. (2016, August 30). *Applied Introductory C Programming - University of New Haven*. Chapters from 2nd edition. http://eliza.newhaven.edu/apc/.

2. Sadeghi, R. (2020, December 14). *RezaSadeghiWSU/LMS*. GitHub. https://github.com/RezaSadeghiWSU/LMS.

3. Sadeghi, R. (2020, December 15). *RezaSadeghiWSU/Calendar-Management-System*. GitHub. https://github.com/RezaSadeghiWSU/Calendar-Management-System.

4. Sadeghi, R. (2021). passwordSystem.c. West Haven; Reza Sadeghi.

Thank You