Team Three: Project Design

Constance Sturm, Michele Cook, Chelsea Nieves, Valerie Rudich, and Hoi Lam Wong

University of Maryland Global Campus

CMSC 495-7382: Capstone in Computer Science

Professor David Castillo

December 5, 2023

Team Three: Project Design

Table of Contents

1.0 Introduction

The purpose of the design plan is to demonstrate how the application will be implemented and developed. This plan design will show all aspects of the application including the format and tools that will be used to implement different functions. Each part of the application will be fully described including the different prompts, the user interaction, how the application processes commands, and what the desired result should be. The project design should fully show what each part of the application will include, as well as, an overview of how the code will be structured. The project will be written in python and will use tools and libraries such as a tkinter for GUI , random generator, buffered reader to parse .txt files, SQLite3 for database storage, and bcrypt for encrypting user passwords.

2.0 Requirements Analysis

2.1 Functional

The application requires that the user makes selections from a GUI menu. The menu will be fully described with notes about how the application should function. There will be a menubar applied to the rules and regulations aspect of the program which will be fully described with coding references. The random generator that is used for the "Random" option and other categories to display a random fortune which will be described with code references and tested for randomness. The different fortune .txt documents will be described with examples of what the fortunes will look like for each category and how the buffered reader will work in the program. Any messages for the user will be demonstrated in the project design.

2.2 Non-functional

Non-functional aspects include the welcome message which will be displayed using labels in the GUI. The testing will also be a non-functional but important aspect of this application.

2.2 Constraints

3.0 Use Cases

Actor Profile: User who desires to gain a deeper insight into their future using the program. A user can be new to the system or have prior experience with the program. This program is suitable for all ages with the use of a computer.

Use Case 1.0: View Main Menu

Actor: User
Description: Displays the main menu user. The menu includes options for "Play", "Login", and "Register" if the user is not signed in. If user is signed in, the menu displays "Get a Fortune" and "View Past Fortunes"
Preconditions: The program has been started
Sequence of events (scenario):
1. The user is welcomed to the program
2. The user is prompted to choose from the following menu.
   a. Play As A Guest
   b. Login
   c. Register
3. The program then takes the user to the Fortune Menu if "Play As A Guest" is selected.
4. The program takes the user to the Login Form if "Login" is selected
5. The program takes the user to the Registration Form if "Register" is selected

Postconditions: The user is then transferred to another menu or prompt based on their selection.

Use Case 2.0: View Rules

Actor: User
Description: The user wants to view the rules for using the fortune teller program.
Preconditions: The program has started and the user is presented with the main menu.
Sequence of events (scenario) :
1. The user selects the 'Rules' option from the menu bar at the top of the main menu.
2. The program displays the rules for using the fortune teller.
3. The user clicks a button to close the rules window.

Postconditions: The user has viewed the rules and is navigated back to the main menu.

Use Case 3.0: Exit Program

Actor: User

Description: The user intends to exit the fortune teller program
Preconditions: The program has been launched, and the user is presented with the main menu.

Sequence of events (scenario) :
1. The user hover over the 'Exit' option from the menu bar.
2. The 'Exit' option display a dropdown menu with option "Exit Program"
3. The user clicks "Exit Program"
4. The program ceases to run.

Alternate Flow
    3a. The user chooses to stay in the program either by hovering or clicking away from the 'Exit' option.
      b. The user is presented with the main menu where the user can make another selection.

Postconditions: The program exits or returns to the main menu based on the user's choice.

Use Case 4.0: View Fortune

Actor: User
Description: The user wishes to receive a fortune based on a specific category of their choice.
Preconditions: The program has started and the user is presented with the main menu.
Sequence of events (scenario) :
1. The user chooses either "Play as Guest" if not logged in, or otherwise "Get a Fortune" in the main menu
2. The program displays a menu with options to choose a fortune category, including Love, Career, General, Health, and Surprise Me (random).
3. The user clicks the button that corresponds to the fortune that they'd like to view.
4. The program displays a new window that displays the fortune message from the selected category.
5. If user is not logged in, he/she is prompts to choose between:
   - "Choose Another Fortune"

Alternate Flow
    5a. If user is logged in, he/she is prompts to choose between:
   - "Choose Another Fortune"
   - "Save Fortune"

Postconditions: The user has received a fortune from their selected category and can either choose another fortune, save their fortune (if logged in), or close the displaying window.

Use Case 4.1: Choose Another Fortune

Actor: User
Description: The user, after viewing a fortune, decides to choose another fortune from a different category of their choice.

Preconditions: The user has already selected and viewed a fortune from a specific category using the Fortune Teller program.

Sequence of events (scenario) :
1. After viewing a fortune, the user is always prompted with the option to"Choose Another Fortune".
2. The user selects "Choose Another Fortune"
3. The fortune message display window is closed.
4. The program displays the fortune menu with options to choose a fortune category, including Love, Career, General, Health, and Surprise Me (random).

Postconditions: The user is presented with the menu of fortune categories again and can select a different category to view another fortune.

Use Case 4.2: Save Fortune

Actor: User

Description: The logged in user, after viewing a fortune, decides to save the displayed fortune to their fortune history.

Preconditions: The user has already selected and viewed a fortune from a specific category using the Fortune Teller program.

Sequence of events (scenario) :
1. After viewing a fortune, if the user is logged in, the user is prompted with the option to"Choose Another Fortune" and "Save Fortune".
2. The authorized user selects "Save Fortune"
3. The window with the fortune message is closed and a new window appears to confirm the fortune is saved.
4. The user can close out the new window by clicking button "Close"
5. The main menu and fortune menu remains open.

Postconditions: The fortune is saved to the user's account and the user can select another category from the fortune menu or select another action from the main menu.

Use Case 5.0: User Register

Actor: User

Description: The guest user decides to register a new account for use in the program.

Preconditions: The program has started, and the guest user (who is not logged in) is presented with the main menu for guest users.

Sequence of events (scenario) :
1. The user is prompted to choose from the following guest main menu.

      a) Play As A Guest

      b) Login

      c) Register

2. The user selects "Register" from the main menu.
3. The user is presented with a new window with the Register form.
4. The user fills out the form with valid inputs.
5. The user clicks on "Submit" to submit the form.
6. The user is presented with a new window displaying a registration confirmed message.
7. The user is asked to login with the new information at the main menu.

Postconditions: The user has registered an account with the information they provided.

Use Case 6.0: User Login

Actor: User
Description: The guest user decides to login in the program.
Preconditions: The program has started, and the guest user (who is not logged in) is presented with the main menu for guest users.
Sequence of events (scenario) :

1. The user is prompted to choose from the following guest main menu.

      a) Play As A Guest

      b) Login

      c) Register

2. The user selects "Login" from the main menu.
3. The user is presented with a new window with the Login form.
4. The user fills out the form with valid username and matching password
5. The user clicks on "Submit" to submit the form.
6. The user is presented with a new main menu with updated options, 'Get Fortune' and 'View Past Fortune', and a welcome back message.

Postconditions: The user is logged into their account and is presented with a new window that includes options to get fortune and view past fortune.

Use Case 7.0: View Past Fortunes

Actor: User
Description: The logged in user decides to view the fortunes they have received and selected to save in the past.
Preconditions: The program has started, and the user (who is logged in) is presented with the main menu for authorized users.
Sequence of events (scenario) :

1. The user is prompted to choose from the following menu for authorized users.

      a) Get a Fortune

      b) View Past Fortunes

2. The user selects "View Past Fortunes" from the main menu.
3. The program fetches all the entries from the past fortunes table that match with the current user's username.
4. The user is presented with a new window that includes a table with their previous received fortunes that they have saved.

Postconditions: The user is presented with a new window that displays the list of previous fortunes that they received and saved.

Use Case 8.0: User Signout

Actor: User
Description: The logged in user decides to save the displayed fortune to their fortune history.
Preconditions: The user has already selected and viewed a fortune from a specific category using the Fortune Teller program.
**Sequence of events (scenario) :**
5. The logged in user selects the 'Sign Out' option from the menu bar at the top of the main menu.
6. The program displays a new window asking the user to confirm the sign out action.
7. The user clicks 'Yes' to sign out.
8. The main menu display for authorized users is now hidden.
9. User is presented with main menu for guest user with the options to
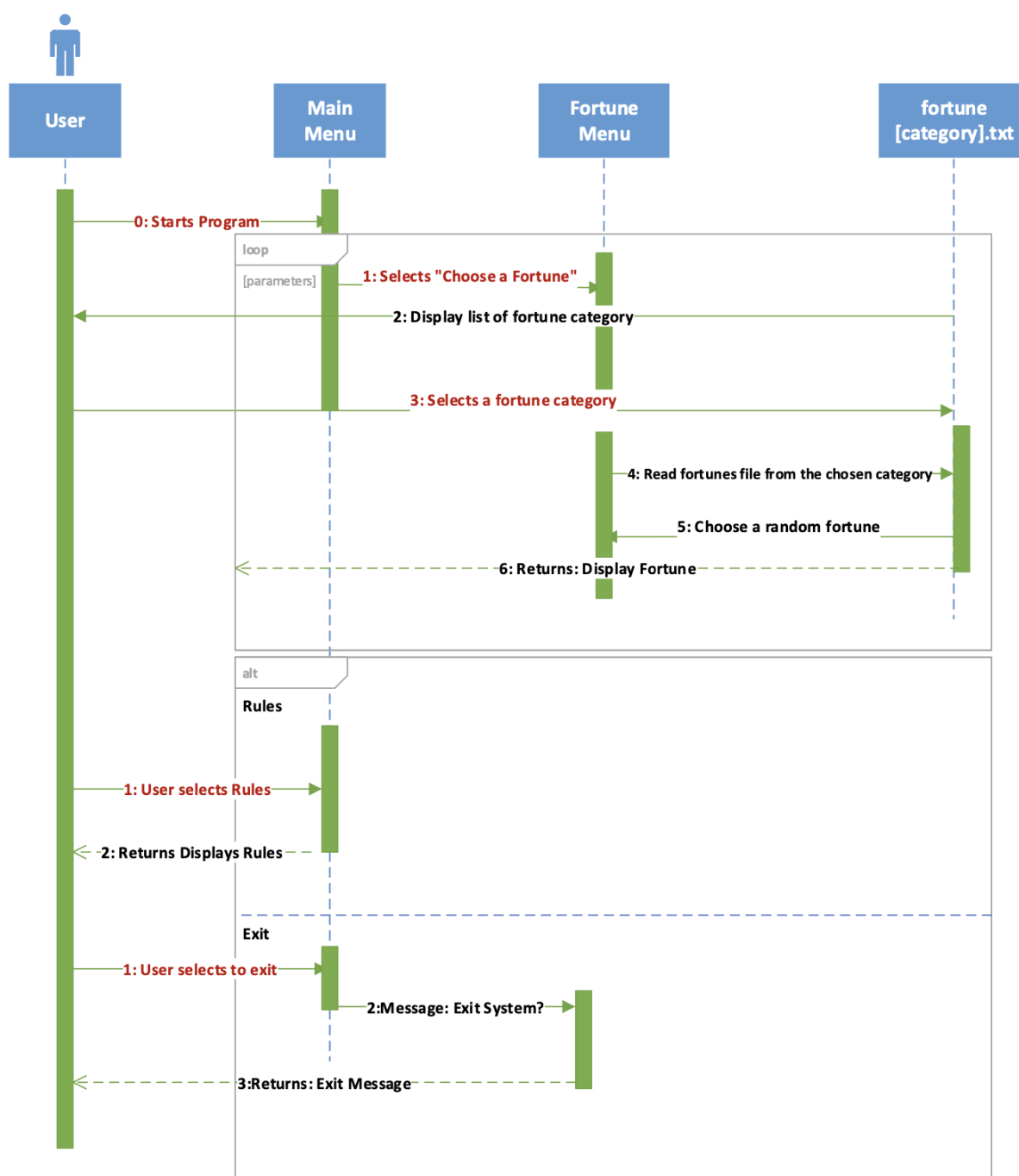   a) Play As A Guest
   b) Login
   c) Register

Postconditions: The user is now signed out and can no longer save and view past fortunes.
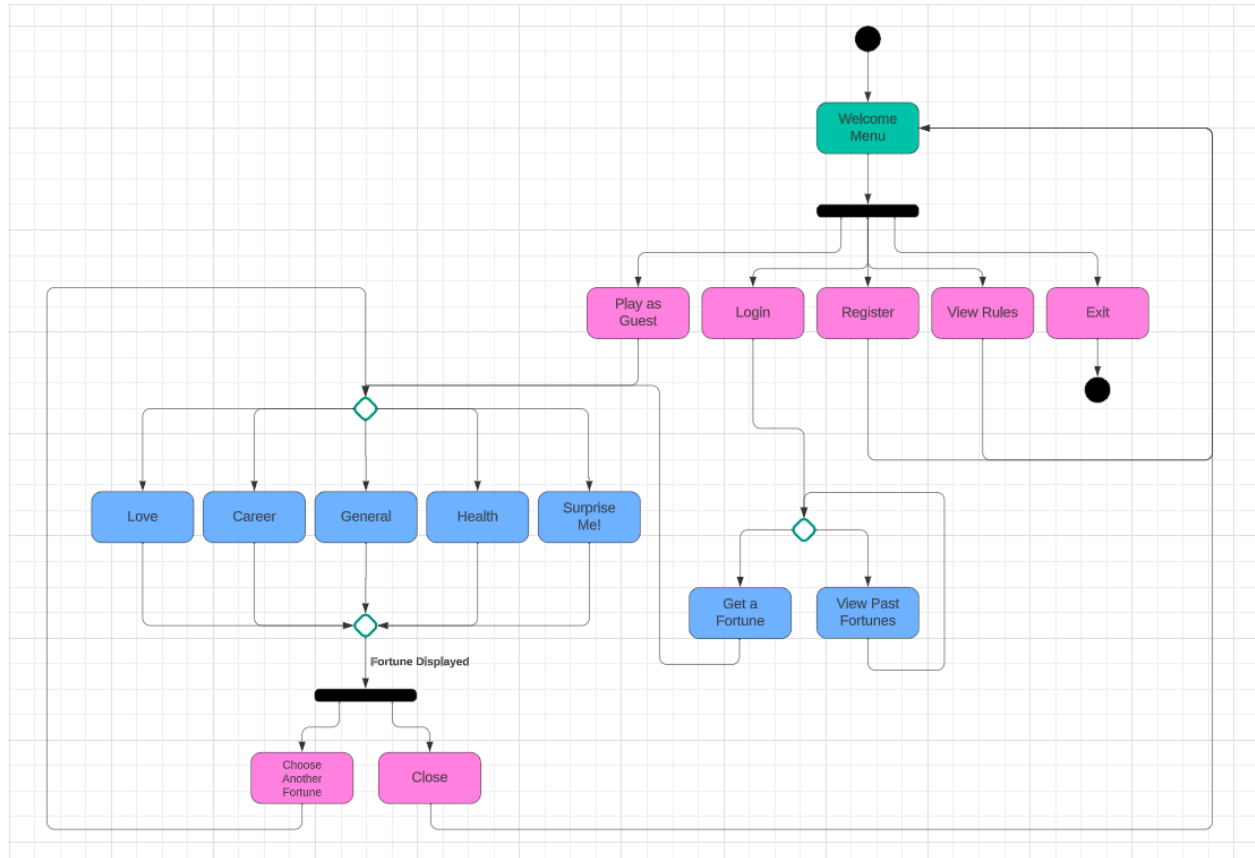
4.0 Diagrams

4.1 Use Case Diagram
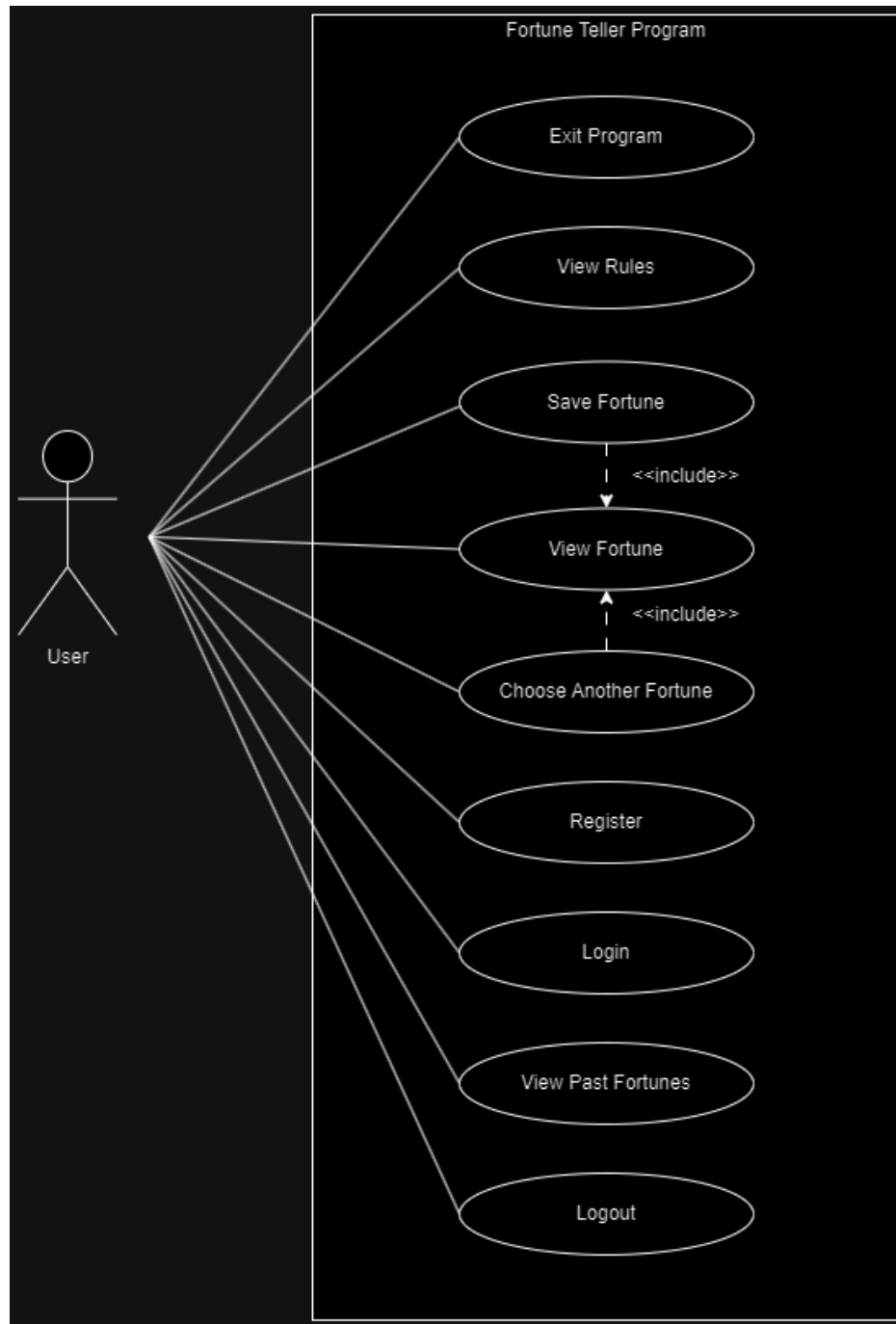
4.2 Sequence Diagram

4.2 Activity Diagram

https://lucid.app/lucidchart/1c4de69c-f614-44f1-9b50-86716c1ebc02/edit?viewport_loc=-672%2C-219%2C2399%2C1386%2C0_0&invitationId=inv_ab70aa7a-fded-4da0-bbac-2a9351d09eab

5.0 Text file

The program will be reading from .txt files for the different fortune categories to display a fortune to the user based on their selection. The text files are also grouped together inside a folder named 'texts' for better organization.

5.1 Example

love_fortune.txt
This is a love fortune
This is a 2nd love fortune
This is a 3rd love fortune
...

general fortune.txt
This is a general fortune
This is a 2nd general fortune
....

health_fortune.txt
This is a health fortune
This is a 2nd general fortune
..

career_fortune.txt
This is a career fortune
This is a 2nd career fortune
..

The random option will read a fortune from all the .txt files and chooses one randomly.
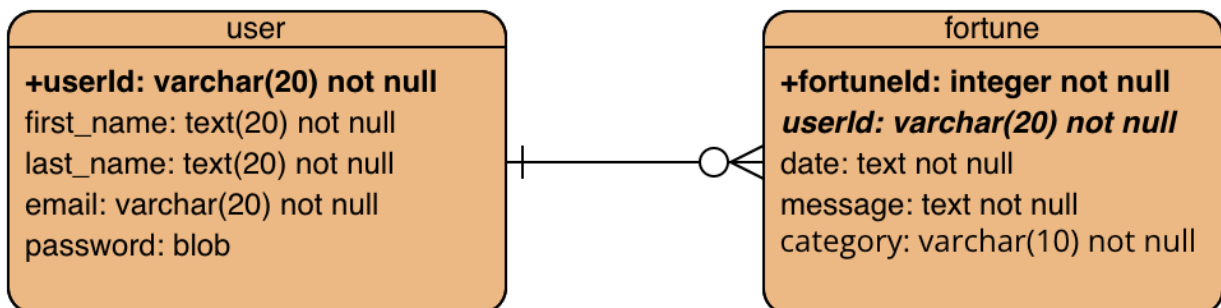
5.2 Purpose

The multiple text files *[category]_fortune.txt* will represent a collection of fortunes within a specific category. For example, *love_fortune.txt* will hold the fortune messages from the category, Love. Each category has their own message getter function which accesses different text files according to the user's input. A separate function would then pick a random fortune message within the category. For random category selection, multiple text files will be read and a fortune is then picked.

5.3 Constraints

The file name must follow the naming convention "[category]_fortune.txt" where [category] is the name of the fortune category the user has chosen. For example, love fortunes are stored in the text file "love_fortune.txt". Each fortune message is separated with a delimiter, serving as a distinct marker that separates one fortune from another within the text file. This allows efficient parsing and retrieval of individual fortunes. With that said, the character that serves as the delimiter cannot appear within the fortune message itself.

To achieve this, we have decided to utilize the Python built-in function 'splitlines()', which automatically appends the special character "\n" to the end of each line, serving as an inconspicuous delimiter. This format not only improves readability but also makes it easier to add new fortune messages.

6.0 Entity- Relationship Diagram

user
+userId: varchar(20) not null
first_name: text(20) not null
last_name: text(20) not null
email: varchar(20) not null
password: blob

fortune
+fortuneId: integer not null
userId: varchar(20) not null
date: text not null
message: text not null
category: varchar(10) not null

7.0 Conclusion

The project will have user interaction using a GUI menu to play as a guest or login/register as a returning user. The rules selection will be displayed with the menu bar selection. The fortune menu will be displayed with the incorporation of buttons to choose the fortune that they would like to display. The returning users will have the option to save their fortunes to a database to be able to view them again at another time. The returning users will also have the option to view their previously saved fortunes from the main menu. Sign out is available to logged in users anytime in the menu bar of the main menu. The categories for the fortunes involve love, career, health, general, and random. The fortunes will have a random generator that will choose from the different .txt files to choose a fortune for the user. With the exit the program option the user will be prompted if they really want to exit the program. If they want to exit they will then have a goodbye message displayed and the program will end.