Assignment #2

Worth: 10% of final grade

Account Ticketing System

Milestone	Worth	Due Date	Submission Required
1	*10%*	*(Suggested Target: November 16th)*	*NO*
2	**40%**	**November 26th by 23:59 EST (end of Week 11)**	**YES**
3	*10%*	*(Suggested Target: November 30th)*	*NO*
4	**40%**	**December 3rd by 23:59 EST (end of Week 12)**	**YES**

Introduction

Assignment 2 is a continuation of Assignment 1 Milestone 4 and will complete the account ticketing system. The ticketing component will be added among other improvements. Additional business and data validation rules will be required to ensure a higher quality data store is maintained including security and overall application functionality. Agents will have more features available including ticket management and data persistence (using files) so when data is modified, the changes are retained and can be reloaded when the application is later restarted.

For assignment 2, you will be provided with a set of generalized instructions and given **more freedom to create your own solution** (this means creating your own functions, macro's, and deciding in what file they should be placed). **However, you must use the data types and functions that are explicitly stated).**

Preparation

Download or clone the Assignment 2 (**A2**) from https://github.com/Seneca-144100/IPC-Project
In the directory: A2/MS1 you will find the Visual Studio project files ready to load. Open the project (**a2ms1.vcxproj**) in Visual Studio.

IMPORTANT

- ***You will need to copy your work from Assignment 1 Milestone 4 and add all the header and source code files to the a2ms1 Visual Studio project before continuing.***
- ***Remember to update the comments at the top of each file to reflect Assignment 2 Milestone 1.***
- ***DO NOT ADD the a1ms4.c file.***

Milestone – 1 *(Worth 10%, Target Due Date: November 19th)*

Milestone – 1 does not require a submission and does not have a specific deadline, however, you should target to have this part completed no later than **November 19th** to ensure you leave enough time to complete Milestone – 2 which must be submitted and is due **November 26th**.

Milestone-1 includes the main function and should not be modified (***a2ms1.c***). This main will do some preliminary data type testing by creating some test data using the ***new data types*** and then launch the application logic by calling the ***applicationStart*** function accordingly.

This milestone focuses on accommodating some ***new data types*** and extending/improving on data validation routines including implementing additional business logic (rules and conditions for data).

Specifications

New Data Types

You will need to create three (3) new data types in this milestone ("**Message**", "**Ticket**", and "**AccountTicketingData**") which will complete what is needed for this application.

Review the ***a2ms1.c*** file (more specifically the "***main***" and "***populateTickets***" functions) to learn more about the field information used in these new data types based on the data being assigned.

The "**Message**" and "**Ticket**" data types will need to be defined in **a new header file "*ticket.h*"** (don't forget to apply the *safeguarding* technique as described in Assignment 1).

Message type

- The **Message** data type has three members. The 1st member is a single character representing the account type of the author of the message (same as used in the ***Account*** type). The 2nd member represents the display name for a given user (same as used in the ***Account*** type) and should be able to accommodate 30 printable characters. The 3rd member represents the message details and should be sized to accommodate 150 printable characters.
- Create meaningful member names for each.

Ticket type

- The **Ticket** type has six (6) members.
  1. Unique number for a ticket.
  2. Customer account number related to the ticket.
  3. A ticket status indicator where 0 represents closed and 1 represent active (still open)
  4. A subject line (like an email subject) that should accommodate up to 30 printable characters.
  5. A counter that represents the number of messages associated with the ticket.
  6. An array of **Message** types that should be able to store up to 20 messages.
- Create meaningful member names for each.

AccountTicketingData type

- This new data type is provided for you below and should be placed in the existing header file **"accountTicketingUI.h".** This type will be used to help simplify the passing of data between key functions with more efficiency and readability.

```
struct AccountTicketingData
{
    struct Account* accounts;   // array of accounts
    const int ACCOUNT_MAX_SIZE; // maximum elements for account array

    struct Ticket* tickets;     // array of tickets
    const int TICKET_MAX_SIZE;  // maximum elements for ticket array
};
```

- Review the ***a2ms1.c*** file to see how this is instantiated and used.

Application Logic Entry-Point

The function parameters for "***applicationStart***" will need to be modified so it receives just one argument which is a pointer to the new "***AccountTicketingData***" type.

- Update the necessary function definition to use the new argument accordingly (after reading and implementing the menu changes below).

Menu Modifications

menuAgent

- The function used for the **agent** main menu "***menuAgent***" currently has three (3) parameters but will require modification. Since the accounts array and its maximum size information are now members of the new data type "***AccountTIcketingData***", the existing first two parameters can be replaced with a pointer to the new "***AccountTicketingData***" type. This means this function should now only have two (2) parameters.
- Update the necessary function definition to use the new argument accordingly.
- The **menuAgent** function should be modified to display five (5) more menu options (5 – 9). Selecting any of these new options will display a temporary notice that the feature is not currently available. Review the sample output for details.

Data Validation and Business Rules

You should **apply system library functions** like the ***character analysis*** and ***manipulators*** you have recently learned about to help enforce data validation and business rules where appropriate (review your code and apply where necessary).

New Account

- The application currently prompts the user for an account number when creating a new **Account**. This is not ideal and needs to be replaced with an **auto-generated account number** based on the next increment of the highest number found in the accounts data set.
  - Upgrade your process for creating a new account so the account number is automatically assigned before getting user input for the remaining data. The account number should also be **displayed as part of the title/banner** (see below sample, the **50600** was automatically assigned).
  - Prompting for a new account should therefore start with the account type like this:

```
New Account Data (Account#:50600)
----------------------------------------
Enter the account type (A=Agent | C=Customer):
```

User Login

- Enhance the validation for obtaining the **UserLogin** member that stores the ***login identifier*** and do not allow any **whitespace characters** (spaces and tabs etc.). Below is an example of an attempt to enter whitespace characters:

```
User Login Data Input
----------------------------------------
Enter user login (10 chars max): my login
ERROR:  The user login must NOT contain whitespace characters.
```

```
Enter user login (10 chars max): my      login
ERROR:  The user login must NOT contain whitespace characters.
Enter user login (10 chars max):
```

- Enhance the validation for obtaining the **UserLogin** member that stores the ***password*** to enforce the password meets the new criteria (see example below):

```
Enter the password (must be 8 chars in length): password
SECURITY: Password must contain 2 of each:
          Digit: 0-9
          UPPERCASE character
          lowercase character
          symbol character: !@#$%^&*
Enter the password (must be 8 chars in length): aaAA#$12
```

Note: "**aaAA#$12**" is valid because it meets the password validation criteria.

Person

- Enhance the **Person** data input processes (when adding new or updating) so that entered values for the ***country*** member are stored as all UPPERCASE characters (the user should be able to enter lowercase characters and you will convert it to uppercase accordingly).

A2-MS1: Sample Output

```
==============================================
Account Ticketing System - Login
==============================================
1) Login to the system
0) Exit application
----------------------------------------------

Selection: 1

Enter your account#: 50008

AGENT: Will Smith (50008)
==============================================
Account Ticketing System - Agent Menu
==============================================
1) Add a new account
2) Modify an existing account
3) Remove an account
4) List accounts: detailed view
----------------------------------------------
5) List new tickets
6) List active tickets
7) List closed tickets
8) Add a new ticket
9) Manage a ticket
----------------------------------------------
0) Logout

Selection: 1

New Account Data (Account#:50600)
----------------------------------------
Enter the account type (A=Agent | C=Customer): A
```

```
Person Data Input
----------------------------------------
Enter the person's full name (30 chars max): Agent Chris
Enter birth year (current age must be between 18 and 110): 1999
Enter the household Income: $240750.11
Enter the country (30 chars max.): england

User Login Data Input
----------------------------------------
Enter user login (10 chars max): Has Space
ERROR:  The user login must NOT contain whitespace characters.
Enter user login (10 chars max): Neil234
Enter the password (must be 8 chars in length): 12345678
SECURITY: Password must contain 2 of each:
          Digit: 0-9
          UPPERCASE character
          lowercase character
          symbol character: !@#$^&*
Enter the password (must be 8 chars in length): pa55WD!d
SECURITY: Password must contain 2 of each:
          Digit: 0-9
          UPPERCASE character
          lowercase character
          symbol character: !@#$^&*
Enter the password (must be 8 chars in length): pa55WD&!

*** New account added! ***

<< ENTER key to Continue... >>[ENTER]

AGENT: Will Smith (50008)
==============================================
Account Ticketing System - Agent Menu
==============================================
1) Add a new account
2) Modify an existing account
3) Remove an account
4) List accounts: detailed view
----------------------------------------------
5) List new tickets
6) List active tickets
7) List closed tickets
8) Add a new ticket
9) Manage a ticket
----------------------------------------------
0) Logout

Selection: 4

Acct# Acct.Type Full Name       Birth Income      Country    Login      Password
----- --------- --------------- ----- ----------- ---------- ---------- --------
30001 CUSTOMER  Silly Sally      1990   150000.10 CANADA
50599 AGENT     Fred Flintstone  1972  2250400.22 AFRICA     agent1     y*b*#*@*
30004 CUSTOMER  Betty Boop       1978   250800.74 INDIA
50008 AGENT     Will Smith       1952  2350600.82 U.S.A.     agentJ     T***2*t*
20020 CUSTOMER  Shrimpy Shrimp   2000   350500.35 KOREA
50600 AGENT     Agent Chris      1999   240750.11 ENGLAND    Neil234    p*5*W*&*
```

```
<< ENTER key to Continue... >>[ENTER]

AGENT: Will Smith (50008)
==============================================
Account Ticketing System - Agent Menu
==============================================
1) Add a new account
2) Modify an existing account
3) Remove an account
4) List accounts: detailed view
----------------------------------------------
5) List new tickets
6) List active tickets
7) List closed tickets
8) Add a new ticket
9) Manage a ticket
----------------------------------------------
0) Logout

Selection: 2

Enter the account#: 50600

Update Account: 50600 (Agent Chris)
----------------------------------------
1) Update account type (current value: A)
2) Person
3) Login
0) Done
Selection: 3

User Login: Neil234 - Update Options
----------------------------------------
1) Password
0) Done
Selection: 1

Enter the password (must be 8 chars in length): juMP1!*&
SECURITY: Password must contain 2 of each:
          Digit: 0-9
          UPPERCASE character
          lowercase character
          symbol character: !@#$^&*
Enter the password (must be 8 chars in length): juMP1!*9

User Login: Neil234 - Update Options
----------------------------------------
1) Password
0) Done
Selection: 0

Update Account: 50600 (Agent Chris)
----------------------------------------
1) Update account type (current value: A)
2) Person
3) Login
0) Done
Selection: 2
```

```
Person Update Options
----------------------------------------
1) Full name (current value: Agent Chris)
2) Household Income (current value: $240750.11)
3) Country (current value: ENGLAND)
0) Done
Selection: 3

Enter the country (30 chars max.): romania

Person Update Options
----------------------------------------
1) Full name (current value: Agent Chris)
2) Household Income (current value: $240750.11)
3) Country (current value: ROMANIA)
0) Done
Selection: 0

Update Account: 50600 (Agent Chris)
----------------------------------------
1) Update account type (current value: A)
2) Person
3) Login
0) Done
Selection: 0

AGENT: Will Smith (50008)
==============================================
Account Ticketing System - Agent Menu
==============================================
1) Add a new account
2) Modify an existing account
3) Remove an account
4) List accounts: detailed view
----------------------------------------------
5) List new tickets
6) List active tickets
7) List closed tickets
8) Add a new ticket
9) Manage a ticket
----------------------------------------------
0) Logout

Selection: 4

Acct# Acct.Type Full Name       Birth Income      Country    Login      Password
----- --------- --------------- ----- ----------- ---------- ---------- --------
30001 CUSTOMER  Silly Sally      1990   150000.10 CANADA
50599 AGENT     Fred Flintstone  1972  2250400.22 AFRICA     agent1     y*b*#*@*
30004 CUSTOMER  Betty Boop       1978   250800.74 INDIA
50008 AGENT     Will Smith       1952  2350600.82 U.S.A.     agentJ     T***2*t*
20020 CUSTOMER  Shrimpy Shrimp   2000   350500.35 KOREA
50600 AGENT     Agent Chris      1999   240750.11 ROMANIA    Neil234    j*M*1***

<< ENTER key to Continue... >>[ENTER]

AGENT: Will Smith (50008)
```

```
==============================================
Account Ticketing System - Agent Menu
==============================================
1) Add a new account
2) Modify an existing account
3) Remove an account
4) List accounts: detailed view
----------------------------------------------
5) List new tickets
6) List active tickets
7) List closed tickets
8) Add a new ticket
9) Manage a ticket
----------------------------------------------
0) Logout

Selection: 5

Feature #5 currently unavailable!

<< ENTER key to Continue... >>[ENTER]

AGENT: Will Smith (50008)
==============================================
Account Ticketing System - Agent Menu
==============================================
1) Add a new account
2) Modify an existing account
3) Remove an account
4) List accounts: detailed view
----------------------------------------------
5) List new tickets
6) List active tickets
7) List closed tickets
8) Add a new ticket
9) Manage a ticket
----------------------------------------------
0) Logout

Selection: 6

Feature #6 currently unavailable!

<< ENTER key to Continue... >>[ENTER]

AGENT: Will Smith (50008)
==============================================
Account Ticketing System - Agent Menu
==============================================
1) Add a new account
2) Modify an existing account
3) Remove an account
4) List accounts: detailed view
----------------------------------------------
5) List new tickets
6) List active tickets
7) List closed tickets
8) Add a new ticket
```

```
9) Manage a ticket
----------------------------------------------
0) Logout

Selection: 7

Feature #7 currently unavailable!

<< ENTER key to Continue... >>[ENTER]

AGENT: Will Smith (50008)
==============================================
Account Ticketing System - Agent Menu
==============================================
1) Add a new account
2) Modify an existing account
3) Remove an account
4) List accounts: detailed view
----------------------------------------------
5) List new tickets
6) List active tickets
7) List closed tickets
8) Add a new ticket
9) Manage a ticket
----------------------------------------------
0) Logout

Selection: 8

Feature #8 currently unavailable!

<< ENTER key to Continue... >>[ENTER]

AGENT: Will Smith (50008)
==============================================
Account Ticketing System - Agent Menu
==============================================
1) Add a new account
2) Modify an existing account
3) Remove an account
4) List accounts: detailed view
----------------------------------------------
5) List new tickets
6) List active tickets
7) List closed tickets
8) Add a new ticket
9) Manage a ticket
----------------------------------------------
0) Logout

Selection: 9

Feature #9 currently unavailable!

<< ENTER key to Continue... >>[ENTER]

AGENT: Will Smith (50008)
==============================================
Account Ticketing System - Agent Menu
==============================================
1) Add a new account
2) Modify an existing account
3) Remove an account
4) List accounts: detailed view
----------------------------------------------
5) List new tickets
6) List active tickets
7) List closed tickets
8) Add a new ticket
9) Manage a ticket
----------------------------------------------
0) Logout

Selection: 0

### LOGGED OUT ###

==============================================
Account Ticketing System - Login
==============================================
1) Login to the system
0) Exit application
----------------------------------------------

Selection: 0

Are you sure you want to exit? ([Y]es|[N]o): Y

==============================================
Account Ticketing System - Terminated
==============================================
```

Milestone – 1 Submission

1. ***This is a <u>test submission</u> for verifying your work only*** – no files will be submitted to your instructor.

2. Upload (file transfer) your all header and source files:
   `a2ms1.c`
   `account.c`
   `account.h`
   `accountTicketingUI.c`
   `accountTicketingUI.h`
   `commonHelpers.c`
   `commonHelpers.h`
   `ticket.h`

3. Login to matrix in an SSH terminal and change directory to where you placed your source code.

4. Manually compile and run your program to make sure everything works properly:

   ```
   gcc -Wall a2ms1.c account.c accountTicketingUI.c commonHelpers.c -o ms1 <ENTER>
   ```

   If there are no error/warnings are generated, execute it: **`ms1`** `<ENTER>`

5. Run the submission command below (replace **profname.proflastname** with **your professors** Seneca userid and replace **NAA** with your section):

   ```
   ~profName.proflastname/submit 144a2ms1/NAA_ms1 <ENTER>
   ```

6. Follow the on-screen submission instructions.

Milestone – 2 *(Worth 20%, Due Date: November 26TH)*

Milestone – 2 will involve refinements to be made to any appropriate code that would benefit from using the string library (string.h). In addition, the login procedure and requirements will be enhanced to include more robust authentication. This will involve the prompting of the user for their account number, login identifier, and password to be validated before being given access to the system (only three (3) attempts will be permitted and if not successful, will be returned to the login menu). Lastly, support for creating, viewing, and managing of ticket information will be added.

Milestone-2 includes the ***main*** function which should not be modified (***a2ms2.c***). The ***main*** function will populate account and ticketing data to be used in testing the changes and new features before handing off the process to your business logic, starting in the ***applicationStart*** function.

Specifications

Reminder

You will be provided with a set of generalized instructions and given more freedom to create your own solution and unless otherwise explicitly stated, you should **create your own functions and macro's where appropriate, including deciding in what file they should be placed**.

String Library

- Review all your code and upgrade where necessary to use functions available from the string library. Functions you should be considering can be any of the following (but no others):

  strlen, strcpy, strcat, strcmp, strncat, strncmp, strncpy, strchr, strrchr

Business Rules and Logic Modifications

Login Process

- The login process currently only prompts for an account number to permit access to the system. This must be changed to incorporate more robust authentication. This will now include prompting for the following:

  - Account number (Note: Only agent type accounts can login)
  - User login identifier
  - Password

- The combined validation of all these pieces of information will determine if the user can have access to the system. If all three pieces of information combined match with the system data then the application should launch the agent main menu.
- Only three (3) attempts are permitted. If the 3rd attempt does not match the records for the provided account and user information, the user should be returned to the starting menu.
- Review the sample output carefully to see how the process should work when invalid account numbers, and/or invalid user login identifiers, and/or invalid passwords are entered.
- Note: You do not want to disclose to the user which of the three (3) fields were incorrect – doing so helps hackers determine where they have guessed correctly! You will also want to avoid disclosing the field length limits as this is very useful information to a hacker.

Remove an Account (agent menu option #3)

- Removal of an account should also remove any tickets (set to a safe empty state) that are associated with the account.
- The result message should be updated to include the number of tickets removed:

```
*** Account Removed! (? tickets removed) ***
```

  Note: Replace ? with the number of tickets that were removed.

Viewing Tickets

- Currently, the agent main menu options #5-#7 to list new, active, and closed tickets indicates the feature is currently unavailable. This must now be replaced with the functionality to display all the appropriate tickets for the respective views (new, active, or closed).

- **New tickets** are determined based on two field values. The ticket status must be open (will have a value of 1) and there will be only one (1) message.

  Note: Tickets will always have at least one message, and the first message will always be associated with a customer type (the first message describes the ticket issue).

- **Active tickets** are determined based on the status being open (will have a value of 1) and this listing will include new tickets.

- **Closed tickets** are determined based on the status being closed (will have a value of 0).

- All ticket listing views will include six (6) pieces of information:
    - Ticket number
    - Account number
    - Customer's full name *****

      *****<u>Hint</u>: see "note" above regarding ticket construction. You may assume the 1st message in a ticket will be authored by a customer.
    - Ticket status
    - Subject line
    - Number of messages
- Review the sample output to see how this should work including what content should be displayed and the desired tabular format.
- You will notice, after displaying the main ticket summaries, the user should have the option to enter a specific ticket number to view the messages related to that specific ticket.

  <u>Note</u>: Only the tickets that are listed may be entered otherwise display the error message:
  ```
  ERROR: Invalid ticket number.
  ```
- Review the sample output to see how the contents of the messages should be displayed.
- When the user enters a zero (0), control will be returned to the ticket menu.

<u>Add New Ticket (menu option #8)</u>

Currently, the agent main menu option #8 indicates the feature is currently unavailable. This must now be replaced with the functionality to create a new customer ticket.

- The system must **automatically determine** the next available **ticket number** (follows the same logic applied in the generation of the next account number).
  <u>Note</u>: If the ticketing system has reached the maximum allowable number of tickets, the following error message should be displayed and then return to the main menu:
  ```
  ERROR: Ticket listing is FULL, call ITS Support!
  ```
- All new tickets are immediately set to the "ACTIVE" status.
- The customer's account number must be associated with the new ticket being created.

  <u>Note</u>: Agent type account holders are not permitted to have tickets, therefore, if the account number entered belongs to an agent type, the following error message should be shown:
  ```
  ERROR: Agent accounts can't have tickets!
  ```
- The main **subject** of the ticket must be entered that concisely summarizes the purpose of the ticket (the problem).
- It is mandatory an initial **message** be entered that details the reason for the ticket.
  <u>Hint</u>: You can guarantee and assume that all tickets will have at least one message and that the first message will be associated to the customer.
- The account type and respective full name must be recorded with the message.

Manage a Ticket (menu option #9)

Currently, the agent main menu option #9 indicates the feature is currently unavailable. This must now be replaced with the functionality to manage a ticket.

- The agent must enter a valid ticket number to be modified. If it is invalid display an error message:
  ```
  ERROR: Invalid ticket number.
  ```
- The agent may modify the ticket in four (4) possible ways:
  1. Add a CUSTOMER message
  2. Add an AGENT message

     Attempting to add a message to a closed ticket should display the error:
     ```
     ERROR: Ticket is closed - new messages are not permitted.
     ```

     Attempting to add a message beyond the maximum message limits should display the error:
     ```
     ERROR: Message limit has been reached, call ITS Support!
     ```

  3. Close the ticket (only if it is currently active)
     - When the ticket is already in a closed state, an error message should be displayed:
       ```
       ERROR: Ticket is already closed!
       ```
     - The agent should be prompted to confirm their action
     - Display a confirmation message when the operation is complete

  4. Re-open a ticket to make it ACTIVE again
     - When the ticket is already in an active state, an error message should be displayed:
       ```
       ERROR: Ticket is already active!
       ```
     - The agent should be prompted to confirm their action
     - Display a confirmation message when the operation is complete
- Review the sample output to see how the ticket management options should be presented.

A2-MS2: Sample Output (LONG)

The LONG submission option will qualify you to **potentially earn a maximum of 100% (A+)**.

Please review the file "***A2MS2-LONG-OUTPUT.pdf***" (located in the **sub-directory**: "***sample-output***") which should be used for this submission option. Also available, is a text file "***a2ms2_4-long-inputs.txt***" which contains only the user inputs and can be used to help automate testing your work by copying and pasting it into your command window.

A2-MS2: Sample Output (SHORT)

The SHORT submission option will qualify you to **potentially earn a maximum of 70% (B)**.

Please review the file "***A2MS2-SHORT-OUTPUT.pdf***" (located in the **sub-directory**: "***sample-output***") which should be used for this submission option. Also available, is a text file "***a2ms2_4-short-inputs.txt***" which contains only the user inputs and can be used to help automate testing your work by copying and pasting it into your command window.

Reflection (Worth 20%, Due Date: Nov. 26th)

Academic Integrity

It is a violation of academic policy to copy content from the course notes or any other published source (including websites, work from another student, or sharing your work with others).

Failure to adhere to this policy will result in the filing of a violation report to the Academic Integrity Committee.

Instructions

- Create a text file named "**reflect.txt**" and record your answers to the questions below in this file.
- Answer each question in sentence/paragraph form unless otherwise instructed.
- A minimum **350** overall word count is required (does NOT include the question or any sample code) and no more than **600.**
- Whenever possible, it is expected you will substantiate your answers with a brief example to demonstrate your view(s).

1. What is your favourite string function from the string library that you have used in this application? Describe why?

2. Milestones 1 & 2 required you to create some additional functions. List all the **new** function **prototypes** you added. For each new function you created, briefly describe why you created it and include what module/library you put it in and why you added it to that specific module.

Reflections will be graded based on the published rubric:
https://github.com/Seneca-144100/IPC-Project/tree/master/Reflection%20Rubric.pdf

Milestone – 2 Submission

1. Upload (file transfer) your all header and source files including your reflection:
   - `a2ms2.c`
   - `account.c`
   - `account.h`
   - `accountTicketingUI.c`
   - `accountTicketingUI.h`
   - `commonHelpers.c`
   - `commonHelpers.h`
   - `ticket.h`
   - `ticket.c`
   - `reflect.txt`

2. Login to matrix in an SSH terminal and change directory to where you placed your source code.

3. Manually compile and run your program to make sure everything works properly:
   ```
   gcc -Wall a2ms2.c account.c ticket.c accountTicketingUI.c commonHelpers.c -o ms2 <ENTER>
   ```
   If there are no error/warnings are generated, execute it: ***ms2*** *<ENTER>*

4. There are TWO options (a or b below) for submission. Decide which **ONE** is best for you. Each option has its own submission command (see below). Replace **profname.proflastname** with **your professors** Seneca userid and replace **NAA** with your section accordingly:

   a. **LONG** version, for a potential **maximum grade of 100% (A+)**.
      Use the sample output file "***A2MS2-LONG-OUTPUT.pdf***":

      ```
      ~profName.proflastname/submit 144a2ms2Long/NAA_ms2Long <ENTER>
      ```

      ===== OR =====

   b. **SHORT** version, for a potential **maximum grade of 70% (B)**.
      Use the sample output file "***A2MS2-SHORT-OUTPUT.pdf***":

      ```
      ~profName.proflastname/submit 144a2ms2Short/NAA_ms2Short <ENTER>
      ```

5. Follow the on-screen submission instructions.

Milestone – 3 *(Worth 10%, Target Due Date: November 30th)*

Milestone – 3 does not require a submission and does not have a specific deadline, however, you should target to have this part completed no later than **November 30th** to ensure you leave enough time to complete Milestone – 4 which will be due **December 3rd**.

The last major component to be added will involve the implementation of persistent storage of the account and ticketing data. The system will require the functionality to load account and ticket information from text files (the storing of account and ticket information to files will be required for Milestone-4).

Milestone-3 includes the main function and, like previous milestones, should not be modified (***a2ms3.c***). The provided main requires you to develop two functions that are responsible for reading data from two text files: one for the account data and the other for the ticket data. Once the data is read from the files and stored into memory, the main will launch the application logic by calling the ***applicationStart*** function.

Specifications

The application will need to add support for ***reading*** data from text files – there are two (2) text files involved:

accounts.txt

The "**accounts.txt**" file is responsible for storing all the system's account data. Each member of the Account data type is represented in the data and delimited using a tilde (~) symbol. Review the contents of this file carefully. (See the file included with this project.)

Hint: Review how the data is stored differently based on the account type!

tickets.txt

The "**tickets.txt**" file is responsible for storing all the system's ticket data (including messages). Each member of the Ticket data type is represented in the data and delimited using a pipe (|) symbol. Review the contents of this file carefully – particularly how it stores the related messages. (See the file included with this project.)

Application Start

- The starting routine of the application (from main) must *read* from the ***accounts.txt*** and ***tickets.txt*** files to prepare the system for operation. Review the main() code in ***a2ms3.c*** and develop the necessary functions that are called which populate the Account and Ticket type array's.

Warning
It is possible the data files can have more records in them than your application is designed to store. You must apply the necessary logic to ensure you don't attempt to over-stuff your arrays.

Testing

After coding the required functions to load the data from the text files, you should be able to run the application and produce the same results as shown in the "A2-MS3: Sample Output" section.

A2-MS3: Sample Output

```
##########################################################################
Starting Account Ticketing System....
   Loading account data... (15 accounts loaded)
   Loading ticket data...  (14 tickets loaded)
##########################################################################


==============================================
Account Ticketing System - Login
==============================================
1) Login to the system
0) Exit application
----------------------------------------------

Selection: 1

Enter the account#: 50008
User Login        : agentJ
Password          : TT*&21tt

AGENT: Will Smith (50008)
==============================================
Account Ticketing System - Agent Menu
==============================================
1) Add a new account
2) Modify an existing account
3) Remove an account
4) List accounts: detailed view
----------------------------------------------
5) List new tickets
6) List active tickets
7) List closed tickets
8) Add a new ticket
9) Manage a ticket
----------------------------------------------
0) Logout

Selection: 4

Acct# Acct.Type Full Name       Birth Income      Country    Login      Password
----- --------- --------------- ----- ----------- ---------- ---------- --------
30001 CUSTOMER  Silly Sally      1990   150000.10 CANADA
50599 AGENT     Fred Flintstone  1972  2250400.22 AFRICA     agent1     y*b*#*@*
30004 CUSTOMER  Betty Boop       1978   250800.74 INDIA
50008 AGENT     Will Smith       1952  2350600.82 U.S.A.     agentJ     T***2*t*
30020 CUSTOMER  Shrimpy Shrimp   2000   350500.35 KOREA
34000 AGENT     Xyla Yates       1991    61907.58 GREECE     Cherokee   E*1*d*&*
50007 CUSTOMER  Chaney Kinney    1963    22288.09 SLOVENIA
30014 AGENT     Hanae Horn       1999    35403.36 SPAIN      Keiko      R*5*r*&*
40021 CUSTOMER  Kane Lancaster   1951    77711.60 PORTUGAL
35035 CUSTOMER  Honorato Banks   1999    83024.91 HONDURAS
42042 CUSTOMER  Dexter Martin    1932    40187.20 GUAM
35049 CUSTOMER  Buck Odom        1990    60494.16 LESOTHO
44056 CUSTOMER  Craig Mcknight   1961    91914.61 BAHAMAS
34063 CUSTOMER  Jeffrey Gills    1989    27746.17 SRI LANKA
50070 AGENT     Wylie Pollard    1990    61384.65 ALBANIA    Lara       E*4*v*&*

<< ENTER key to Continue... >>[ENTER]

AGENT: Will Smith (50008)
==============================================
Account Ticketing System - Agent Menu
==============================================
1) Add a new account
2) Modify an existing account
3) Remove an account
```

```
4) List accounts: detailed view
----------------------------------------------
5) List new tickets
6) List active tickets
7) List closed tickets
8) Add a new ticket
9) Manage a ticket
----------------------------------------------
0) Logout

Selection: 5

------ ----- --------------- ------ ------------------------------ --------
Ticket Acct# Full Name       Status Subject                        Messages
------ ----- --------------- ------ ------------------------------ --------
080599 30001 Silly Sally     ACTIVE No power/does not turn on          1
040599 35049 Buck Odom       ACTIVE Power Issue                        1
------ ----- --------------- ------ ------------------------------ --------

Enter the ticket number to view the messages or
0 to return to previous menu: 0

AGENT: Will Smith (50008)
==============================================
Account Ticketing System - Agent Menu
==============================================
1) Add a new account
2) Modify an existing account
3) Remove an account
4) List accounts: detailed view
----------------------------------------------
5) List new tickets
6) List active tickets
7) List closed tickets
8) Add a new ticket
9) Manage a ticket
----------------------------------------------
0) Logout

Selection: 6

------ ----- --------------- ------ ------------------------------ --------
Ticket Acct# Full Name       Status Subject                        Messages
------ ----- --------------- ------ ------------------------------ --------
060001 30004 Betty Boop      ACTIVE Frequent Disconnects               5
080599 30001 Silly Sally     ACTIVE No power/does not turn on          1
080004 30020 Shrimpy Shrimp  ACTIVE My head hurts!                     3
080020 30020 Shrimpy Shrimp  ACTIVE It's broken/does not work          5
030530 30004 Betty Boop      ACTIVE Does not respond to command...    17
080204 30001 Silly Sally     ACTIVE It's very messy!                   2
040599 35049 Buck Odom       ACTIVE Power Issue                        1
040001 40021 Kane Lancaster  ACTIVE Connectivity Problem               5
040530 40021 Kane Lancaster  ACTIVE Not doing what it's told...       17
040004 35049 Buck Odom       ACTIVE Causes bodily harm!                3
040204 50070 Wylie Pollard   ACTIVE It's very messy!                   2
------ ----- --------------- ------ ------------------------------ --------

Enter the ticket number to view the messages or
```

```
0 to return to previous menu: 40204

================================================================================
040204 (ACTIVE) Re: It's very messy!
================================================================================
CUSTOMER (Wylie Pollard):
   It this supposed to be so messy?

AGENT (Will Smith):
   It's a slime blaster - so yes, it is supposed to be VERY messy!

<< ENTER key to Continue... >>[ENTER]

------ ----- --------------- ------ ------------------------------ --------
Ticket Acct# Full Name       Status Subject                        Messages
------ ----- --------------- ------ ------------------------------ --------
060001 30004 Betty Boop      ACTIVE Frequent Disconnects               5
080599 30001 Silly Sally     ACTIVE No power/does not turn on          1
080004 30020 Shrimpy Shrimp  ACTIVE My head hurts!                     3
080020 30020 Shrimpy Shrimp  ACTIVE It's broken/does not work          5
030530 30004 Betty Boop      ACTIVE Does not respond to command...    17
080204 30001 Silly Sally     ACTIVE It's very messy!                   2
040599 35049 Buck Odom       ACTIVE Power Issue                        1
040001 40021 Kane Lancaster  ACTIVE Connectivity Problem               5
040530 40021 Kane Lancaster  ACTIVE Not doing what it's told...       17
040004 35049 Buck Odom       ACTIVE Causes bodily harm!                3
040204 50070 Wylie Pollard   ACTIVE It's very messy!                   2
------ ----- --------------- ------ ------------------------------ --------

Enter the ticket number to view the messages or
0 to return to previous menu: 0

AGENT: Will Smith (50008)
==============================================
Account Ticketing System - Agent Menu
==============================================
1) Add a new account
2) Modify an existing account
3) Remove an account
4) List accounts: detailed view
----------------------------------------------
5) List new tickets
6) List active tickets
7) List closed tickets
8) Add a new ticket
9) Manage a ticket
----------------------------------------------
0) Logout

Selection: 7

------ ----- --------------- ------ ------------------------------ --------
Ticket Acct# Full Name       Status Subject                        Messages
------ ----- --------------- ------ ------------------------------ --------
070533 30004 Betty Boop      CLOSED Nothing happens...                 1
064611 30020 Shrimpy Shrimp  CLOSED It gets hot and smokes             4
044611 35049 Buck Odom       CLOSED Unit is burning up                 4
------ ----- --------------- ------ ------------------------------ --------
```

```
Enter the ticket number to view the messages or
0 to return to previous menu: 0

AGENT: Will Smith (50008)
==============================================
Account Ticketing System - Agent Menu
==============================================
1) Add a new account
2) Modify an existing account
3) Remove an account
4) List accounts: detailed view
----------------------------------------------
5) List new tickets
6) List active tickets
7) List closed tickets
8) Add a new ticket
9) Manage a ticket
----------------------------------------------
0) Logout

Selection: 0

### LOGGED OUT ###

==============================================
Account Ticketing System - Login
==============================================
1) Login to the system
0) Exit application
----------------------------------------------

Selection: 0

Are you sure you want to exit? ([Y]es|[N]o): Y

==============================================
Account Ticketing System - Terminated
==============================================
```

Milestone – 3 Submission

1. ***This is a test submission for verifying your work only*** – no files will be submitted to your instructor.
2. Upload (file transfer) your all header and source files:

```
a2ms3.c
account.c
account.h
accountTicketingUI.c
accountTicketingUI.h
commonHelpers.c
commonHelpers.h
ticket.h
```

ticket.c

3. Login to matrix in an SSH terminal and change directory to where you placed your source code.

4. Manually compile and run your program to make sure everything works properly:

```
gcc -Wall a2ms3.c account.c ticket.c accountTicketingUI.c commonHelpers.c -o ms3 <ENTER>
```

If there are no error/warnings are generated, execute it: `ms3 <ENTER>`

5. Run the submission command below (replace **profname.proflastname** with **<u>your professors</u>** Seneca userid and replace **NAA** with your section):

```
~profName.proflastname/submit 144a2ms3/NAA_ms3 <ENTER>
```

6. Follow the on-screen submission instructions.

<u>Milestone – 4</u> *(Worth 20%, Due Date: December 3rd)*

This will be the final milestone for the account ticketing system application. In this milestone, you will complete the implementation of persistent storage of the account and ticketing data process. The system will require the functionality to save account and ticket information to text files.

This is your last chance to refine your work! The completion of this milestone is a culmination of weeks of hard work, and you should celebrate it by polishing it with all the refinements you can think of and make it a piece of work you can be proud of. Therefore (and to be eligible to receive maximum marks), it is expected you will thoroughly review your code and…

1. Implement all necessary changes as per your instructor's feedback from previous milestones.
2. Beautify your code so it is easy to read and maintain, which includes:
   - Consistent formatting of code (indentation and line-spacing).
   - Use of sufficient comments that concisely describe critical sections of logic to maximize the understanding of your code where the code is otherwise not quickly easy to read.
   - Applying best practices for variable, function, and parameter naming.
   - Following the style guidelines as demonstrated by the course notes, video examples, previous tests quizzes, and your professor's examples.
3. Remove any violations of the single-entry single-exit principle (see course notes regarding the use of flags to control logic flow).
4. Apply as best you can, the design principles for modularity and structured design as described in the course notes.

<u>Specifications</u>

When an **<u>agent</u>** log's out of their session and returns to the main login menu, the current state of the ***accounts array*** must be saved to the ***accounts.txt*** file (<u>recreates</u> the file). The current state of the ***tickets array*** must also be saved to the ***tickets.txt*** file (<u>recreates</u> the file) and displays the number of records saved to each file. Review the sample output where an agent log's out to see how this should work.

Suggestion

Be sure to copy the provided original "***accounts.txt***" and "***tickets.txt***" files from the **sub-directory:** "***master-data-files***" to your source code directory after each execution of your program for ***retesting***. This will be required now that you will be modifying these files after each execution! Always work from a fresh state with each execution of your program!

You may also want to look into creating a "***.bat***" (Windows) or ***bash*** (Linux/Mac OS) script file that will automate copying the files for you!

Note: The solution data text files ("***accounts-long-final.txt***" or "***account-short-final.txt***" and "***tickets-long-final.txt***" or "***tickets-short-final.txt***") are provided for you with this milestone's project located in the **sub-directory**: "***sample-output***" so you can compare and confirm you have successfully written the records correctly (use the appropriate files that match the submission type you plan to use).

A2-MS4: Sample Output (LONG)

The LONG submission option will qualify you to **potentially earn a maximum of 100% (A+)**.

Please review the file "***A2MS4-LONG-OUTPUT.pdf***" (located in the **sub-directory**: "***sample-output***") which should be used for this submission option. Also available, is a text file "***a2ms2_4-long-inputs.txt***" which contains only the user inputs and can be used to help automate testing your work by copying and pasting it into your command window.

Confirm your resulting "***accounts.txt***" and "***tickets.txt***" files against the provided solution text files "***accounts-long-final.txt***" and "***tickets-long-final.txt***" (located in the **sub-directory**: "***master-data-files***").

A2-MS4: Sample Output (SHORT)

The SHORT submission option will qualify you to **potentially earn a maximum of 70% (B)**.

Please review the file "***A2MS4-SHORT-OUTPUT.pdf***" (located in the **sub-directory**: "***sample-output***") which should be used for this submission option. Also available, is a text file "***a2ms2_4-short-inputs.txt***" which contains only the user inputs and can be used to help automate testing your work by copying and pasting it into your command window.

Confirm your resulting "***accounts.txt***" and "***tickets.txt***" files against the provided solution text files "***accounts-short-final.txt***" and "***tickets-short-final.txt***" (located in the **sub-directory**: "***master-data-files***").

Reflection (Worth 20%, Due Date: December 3rd)

Academic Integrity

It is a violation of academic policy to copy content from the course notes or any other published source (including websites, work from another student, or sharing your work with others).

Failure to adhere to this policy will result in the filing of a violation report to the Academic Integrity Committee.

Instructions

- Create a text file named "**reflect.txt**" and record your answers to the questions below in this file.
- Answer each of the 3 questions below in sentence/paragraph form unless otherwise instructed.

1. NOT including the mandatory functions stated in the specifications from Assignments 1 and 2, **LIST all** the functions (use the prototypes) **you** created. Categorize them by module/library in the following sequence (if you did not create any functions for a given module/library, simply state "***None developed***"):

   a) commonHelpers.h
   b) account.h
   c) ticket.h
   d) accountTicketingUI.h

2. Identify **two (2) major components** (features) from the assignment that you **enjoyed** developing and detail why you found it enjoyable. Your answer must be at least 200 words but no more than 300.

3. Identify **two(2)** major components (features) from the assignment that you **disliked** developing and detail why you found it so unlikable. Your answer must be at least 200 words but no more than 300.

NOTE: The submission process cannot be one of those reasons!

Reflections will be graded based on the published rubric:
https://github.com/Seneca-144100/IPC-Project/tree/master/Reflection%20Rubric.pdf

Milestone – 4 Submission

1. Upload (file transfer) your all header and source files including your reflection:
   ```
   a2ms4.c
   account.c
   account.h
   accountTicketingUI.c
   accountTicketingUI.h
   commonHelpers.c
   commonHelpers.h
   ticket.h
   ticket.c
   accounts.txt
   tickets.txt
   reflect.txt
   ```

2. Login to matrix in an SSH terminal and change directory to where you placed your source code.

3. Manually compile and run your program to make sure everything works properly:
   ```
   gcc -Wall a2ms4.c account.c accountTicketingUI.c commonHelpers.c ticket.c -o ms4 <ENTER>
   ```
   If there are no error/warnings are generated, execute it: **`ms4`** *`<ENTER>`*

4. There are <u>TWO options</u> (a or b below) for submission. Decide which **<u>ONE</u>** is best for you. Each option has its own submission command (see below). Replace **profname.proflastname** with **<u>your professors</u>** Seneca userid and replace **NAA** with your section accordingly:

   a. **<u>LONG</u>** version, for a potential **maximum grade of 100% (A+)**.
      Use the sample output file "***A2MS4-LONG-OUTPUT.pdf***":

      ```
      ~profName.proflastname/submit 144a2ms4Long/NAA_ms4Long <ENTER>
      ```

      ===== OR =====

   b. **<u>SHORT</u>** version, for a potential **maximum grade of 70% (B)**.
      Use the sample output file "***A2MS4-SHORT-OUTPUT.pdf***":

      ```
      ~profName.proflastname/submit 144a2ms4Short/NAA_ms4Short <ENTER>
      ```

5. Follow the on-screen submission instructions.