Contact Management System

JAVAC

Htet Oo Thu (662115504)

Phone Myat Ko Ko (662115512)

Wai Yan Moe Lwin (662115516)

**Project Plan:**

* Member roles
  + Project Manager (PM) - Htet Oo Thu (662115504)
  + Software Programmers - Wai Yan Moe Lwin (662115516)

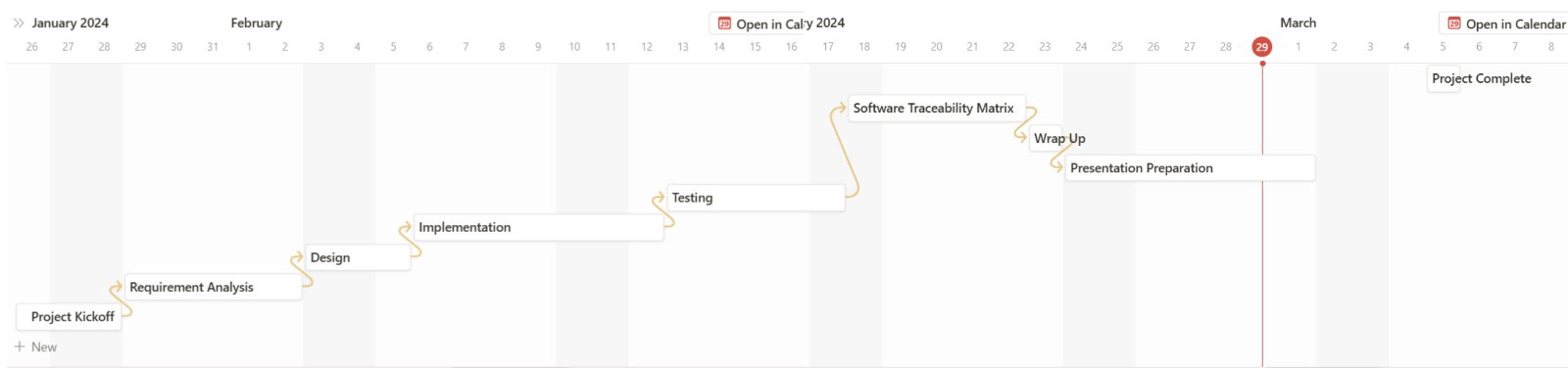
Phone Myat Ko Ko (662115512)

* + Software Testers - Wai Yan Moe Lwin (662115516)

Phone Myat Ko Ko (662115512)

* Program scope
  + To create a simple contact management system.
  + In this system, user can
    - Add contacts
    - View contact details
    - Update information
    - Delete contacts
  + Methodology – Incremental Model
  + Project Timeline – 26 Jan – 5 March
  + Assumptions
    - This system is console-based program
    - The system uses csv file as data base
* Schedule and Gantt chart

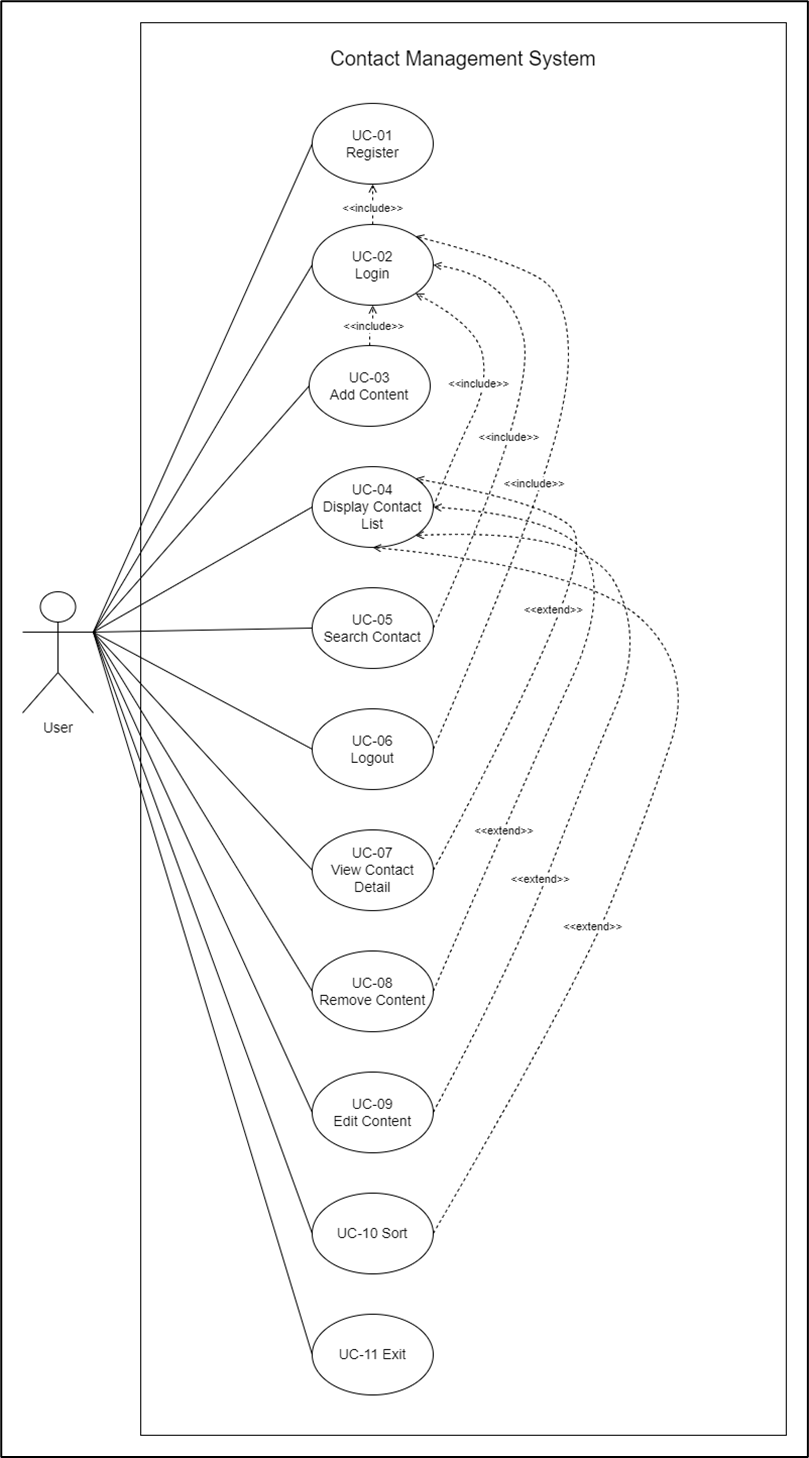
[Link to Notion Project Schedule](https://www.notion.so/Gantt-chart-305fa68dd01a48edb9c241e8cf153c95?pvs=4)



* Technologies and Tools
  + Technologies used to develop the program – JDK, VS Code
  + Project Management Tools – Notion
  + Diagram Tools – draw.io
  + Document and Version Control Tools – MS 365 Word, GitHub.
  + Collaboration and Communication Tools – MS team

**Requirements:**

* Type of users
  + The users who want to save and manage their contacts.
* Use case diagram and description.



Use case diagram

* Functional requirements

[URS-01]: User can register a new account with username and password.

[SRS-01] The system shall display prompts to choose 1(Login), 2(Register), 0(Exit).

[SRS-02] The system shall display prompts to enter username and password.

[SRS-03] The System shall create a new account with username and password given by the user.

[SRS-04] The system shall redirect to the “Login/Register” prompts.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use Case ID | UC-01 | | | | |
| Use Case Name | Register | | | | |
| Created By | JAVAC | | Last Update By | |  |
| Date Created | 01/02/2024 | | Last Revision Date | |  |
| Actors | User | | | | |
| Description | User can register a new account with username and password | | | | |
| Trigger | User enters “2” in “Login/Register” page | | | | |
| Preconditions | - | | | | |
| Use Case Input Specification | | | | | |
| Input | Type | Constraint | | | Example |
| Username | String | The value cannot be empty | | | KoKo |
| Password | String | The value cannot be empty | | | Password123 |
| Post Conditions | User account will be created, and user will be redirected to “Login/Register” page | | | | |
| Normal Flows | User | | | System | |
|  | 2. User enters “2”.        4. User enters username and password. [A1: User inputs empty value] | | | 1. The system shall display prompts to choose login or register.    3. The system shall display prompts to enter username and password.        5. The System shall create new account with username and password given by the user.    6. The system shall redirect to the “Login/Register” prompts. | |
| Alternative Flow | [A1: User inputs empty value]   1. The system shall display the message “Username/Password cannot be empty!!!” below the password input prompt. 2. Go back to normal flow 3. | | | | |
| Exception Flow | - | | | | |
| Assumption | - | | | | |
|  |  |  |  |  |  |

[URS-02]: User can login to the existing account with username and password.

[SRS-05] The system shall display prompts to choose 1(Login), 2(Register), 0(Exit).

[SRS-06] The system shall display prompts to enter username and password.

[SRS-07] The system shall redirect to the “Home Menu” prompts and load contact file of respective user.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use Case ID | UC-02 | | | | |
| Use Case Name | Login | | | | |
| Created By | JAVAC | | Last Update By | |  |
| Date Created | 01/02/2024 | | Last Revision Date | |  |
| Actors | User | | | | |
| Description | User can login to the existing account with username and password | | | | |
| Trigger | User enters “1” in “Login/Register” page | | | | |
| Preconditions | - | | | | |
| Use Case Input Specification | | | | | |
| Input | Type | Constraint | | | Example |
| Username | String | The value cannot be empty | | | KoKo |
| Password | String | The value cannot be empty | | | Password123 |
| Post Conditions | User will be redirected to “Home Menu” page | | | | |
| Normal Flows | User | | | System | |
|  | 2. User enters “1”.        4. User enters username and password.  [A1: User inputs empty value]  [A2: User inputs incorrect username or password] | | | 1. The system shall display prompts to choose login or register.    3. The system shall display prompts to enter username and password.            5. The system shall redirect to the “Home Menu” prompts and load contact file of respective user. | |
| Alternative Flow | [A1: User inputs empty value]   1. The system shall display the message “Username/Password cannot be empty!!!” below the password input prompt. 2. Go back to normal flow 3.     [A2: User inputs incorrect username or password]   1. The system shall display the message “Username or Password is incorrect!!!” below the password input prompt. 2. Go back to normal flow 3. | | | | |
| Exception Flow | - | | | | |
| Assumption | User has already registered account. | | | | |
|  |  |  |  |  |  |

[URS-03]: User can create a new contact with name, phone number, email (optional), and address (optional).

[SRS-08] The system shall display prompts to choose 1(Add Contact), 2(Display Contact-List), 3(Search Contact), 0(logout).

[SRS-09] The system shall display prompts to enter name, phone number, email (optional), and address (optional).

[SRS-10] The System shall create a new contact and add it to the user contact list.

[SRS-11] The system shall redirect to the “Home Menu” prompts.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use Case ID | UC-03 | | | | |
| Use Case Name | Add Contact | | | | |
| Created By | JAVAC | | Last Update By | |  |
| Date Created | 01/02/2024 | | Last Revision Date | |  |
| Actors | User | | | | |
| Description | User can create a new contact with name, phone number, email (optional), and address (optional). | | | | |
| Trigger | User enters “1” in “Home Menu” page | | | | |
| Preconditions | User have logged in | | | | |
| Use Case Input Specification | | | | | |
| Input | Type | Constraint | | | Example |
| Name | String | The value cannot be empty | | | KoKo |
| Phone Number | String | The value cannot be empty | | | 091-23456789 |
| Email (optional) | String | - | | | [koko@gmail.com](mailto:koko@gmail.com) |
| Address (optional) | String | - | | | Green Nimman |
| Post Conditions | Contact will be created and added to the user’s contact list.  User will be redirected to “Home Menu” page. | | | | |
| Normal Flows | User | | | System | |
|  | 2. User enters “1”.          4. User enters name, phone number, email (optional), and address (optional).  [A1: User inputs empty value for name or phone number]  [A2: User inputs existing contact name] | | | 1. The system shall display prompts to choose 1(Add Contact), 2(Display Contact-List), 3(Search Contact), 0(logout).    3. The system shall display prompts to enter name, phone number, email (optional), and address (optional).                5. The System shall create a new contact and add it to the user contact list.    6. The system shall redirect to the “Home Menu” prompts. | |
| Alternative Flow | [A1: User inputs empty value for name or phone number]   1. The system shall display the message “Contact Name or Phone Number cannot be empty!!!” below the password input prompt. 2. Go back to normal flow 3.     [A2: User inputs existing contact name]   1. The system shall display the message “This contact name has already existed. Update existing contact or create a new contact with different name” below the password input prompt. 2. Go back to normal flow 3. | | | | |
| Exception Flow | - | | | | |
| Assumption | - | | | | |
|  |  |  |  |  |  |

[URS-04]: User can view contact list.

[SRS-12] The system shall display prompts to choose 1(Add Contact), 2(Display Contact-List), 3(Search Contact), 0(logout).

[SRS-13] The system shall display a contact list.

[SRS-14] The System shall redirect “Contact Menu”.

[SRS-15] The system shall redirect to the “Home Menu” prompts.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use Case ID | UC-04 | | | |
| Use Case Name | Display Contact List | | | |
| Created By | JAVAC | Last Update By | |  |
| Date Created | 01/02/2024 | Last Revision Date | |  |
| Actors | User | | | |
| Description | User can view contact list. | | | |
| Trigger | User enters “2” in “Home Menu” page | | | |
| Preconditions | User have logged in | | | |
| Post Conditions | Contact list will be displayed.  “Contact Menu” will be displayed. | | | |
| Normal Flows | User | | System | |
|  | 2. User enters “2”.            5. User enters “0”. | | 1. The system shall display prompts to choose 1(Add Contact), 2(Display Contact-List), 3(Search Contact), 0(logout).    3. The system shall display a contact list.  [A1: Contact list is empty]    4. The System shall redirect “Contact Menu”.    6. The system shall redirect to the “Home Menu” prompts. | |
| Alternative Flow | [A1: Contact list is empty]   1. The system shall display the message “Empty Contact List”. 2. The system shall redirect to “Home Menu”. | | | |
| Exception Flow | - | | | |
| Assumption | - | | | |
|  |  |  |  |  |

[URS-05]: User can search contact by name.

[SRS-16] The system shall display prompts to choose 1(Add Contact), 2(Display Contact-List), 3(Search Contact), 0(logout).

[SRS-17] The system shall display prompts to enter keywords.

[SRS-18] The system shall display a contact list related to keyword.

[SRS-19] The System shall redirect “Contact Menu”.

[SRS-20] The system shall redirect to the “Home Menu” prompts.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use Case ID | UC-05 | | | | |
| Use Case Name | Search Contact | | | | |
| Created By | JAVAC | | Last Update By | |  |
| Date Created | 01/02/2024 | | Last Revision Date | |  |
| Actors | User | | | | |
| Description | User can search contact by name. | | | | |
| Trigger | User enters “3” in “Home Menu” page | | | | |
| Preconditions | User have logged in | | | | |
| Use Case Input Specification | | | | | |
| Input | Type | Constraint | | | Example |
| Keyword | String | - | | | Ko |
| Post Conditions | Search results will be displayed.  “Contact Menu” will be displayed. | | | | |
| Normal Flows | User | | | System | |
|  | 2. User enters “3”.      4. User enters keyword            7. User enters “0”. | | | 1. The system shall display prompts to choose 1(Add Contact), 2(Display Contact-List), 3(Search Contact), 0(logout).    3. The system shall display prompts to enter keywords.    5. The system shall display a contact list related to keywords.    6. The System shall redirect “Contact Menu”.    8. The system shall redirect to the “Home Menu” prompts. | |
| Alternative Flow | - | | | | |
| Exception Flow | - | | | | |
| Assumption | - | | | | |
|  |  |  |  |  |  |

[URS-06]: User can logout.

[SRS-21] The system shall display prompts to choose 1(Add Contact), 2(Display Contact-List), 3(Search Contact), 0(logout).

[SRS-22] The system shall save contact lists in the database.

[SRS-23] The System shall display “Logged Out”.

[SRS-24] The system shall redirect to the “Login/Register” prompts.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use Case ID | UC-06 | | | |
| Use Case Name | Logout | | | |
| Created By | JAVAC | Last Update By | |  |
| Date Created | 01/02/2024 | Last Revision Date | |  |
| Actors | User | | | |
| Description | User can logout | | | |
| Trigger | User enters “0” in “Home Menu” page | | | |
| Preconditions | User have logged in | | | |
| Post Conditions | User will be redirected to “Login/Register” page | | | |
| Normal Flows | User | | System | |
|  | 2. User enters “0”. | | 1. The system shall display prompts to choose 1(Add Contact), 2(Display Contact-List), 3(Search Contact), 0(logout).    3. The system shall save contact lists in the database.    4. The System shall display “Logged Out”.    5. The system shall redirect to the “Login/Register” prompts. | |
| Alternative Flow | - | | | |
| Exception Flow | - | | | |
| Assumption | - | | | |
|  |  |  |  |  |

[URS-07]: User can view the detailed information of the specific contact.

[SRS-25] The system shall display prompts to choose 1(View Contact Detail), 2(Remove Contact), 3(Update Contact Information), 4(Sort), 0(Back).

[SRS-26] The system shall display prompts to enter ID.

[SRS-27] The system shall display a “Contact Name”, “Phone Number”, “Email” and “Address”.

[SRS-28] The System shall redirect “Contact Menu”.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use Case ID | UC-07 | | | | |
| Use Case Name | View Contact Detail | | | | |
| Created By | JAVAC | | Last Update By | |  |
| Date Created | 01/02/2024 | | Last Revision Date | |  |
| Actors | User | | | | |
| Description | User can view the detailed information of the specific contact. | | | | |
| Trigger | User enters “1” in “Contact Menu” page | | | | |
| Preconditions | UC-04 or UC-05 | | | | |
| Use Case Input Specification | | | | | |
| Input | Type | Constraint | | | Example |
| ID | Integer | ID must be in the range of 1 to the number of contacts saved. | | | 2 |
| Post Conditions | Display “Contact Name”, “Phone Number”, “Email” and “Address”. | | | | |
| Normal Flows | User | | | System | |
|  | 2. User enters “1”.      4. User enters ID.  [A1: ID is out of ranged]  [A2; User enter character]      6. User press “Enter”. | | | 1. The system shall display prompts to choose 1(View Contact Detail), 2(Remove Contact), 3(Edit Contact), 4(Sort), 0(Back).    3. The system shall display prompts to enter ID.    5. The system shall display a “Contact Name”, “Phone Number”, “Email” and “Address”.      7. The System shall redirect “Contact Menu”. | |
| Alternative Flow | [A1: ID is out of ranged]   1. The system shall display the message “Invalid ID”.   [A2; User enter character]   1. The system shall display the message “Enter only number”. | | | | |
| Exception Flow | - | | | | |
| Assumption | - | | | | |
|  |  |  |  |  |  |

[URS-08]: User can delete the specific contact.

[SRS-29] The system shall display prompts to choose 1(View Contact Detail), 2(Remove Contact), 3(Update Contact Information), 4(Sort), 0(Back).

[SRS-30] The system shall display prompts to enter ID.

[SRS-31] The system shall display the alert message “Are you sure about that? y/n.

[SRS-32] The system shall remove the selected contact.

[SRS-33] The system shall display the message “Contact Deleted”.

[SRS-34] The System shall redirect “Contact Menu”.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use Case ID | UC-08 | | | | |
| Use Case Name | Remove Contact | | | | |
| Created By | JAVAC | | Last Update By | |  |
| Date Created | 02/02/2024 | | Last Revision Date | |  |
| Actors | User | | | | |
| Description | User can delete the specific contact. | | | | |
| Trigger | User enters “2” in “Contact Menu” page. | | | | |
| Preconditions | UC-04 or UC-05 | | | | |
| Use Case Input Specification | | | | | |
| Input | Type | Constraint | | | Example |
| ID | Integer | ID must be in the range of 1 to the number of contacts saved. | | | 2 |
| confirm | char | Must be ‘y’ or ‘n’. | | | y |
| Post Conditions | The selected contact is deleted. | | | | |
| Normal Flows | User | | | System | |
|  | 2. User enters “2”.      4. User enters ID.  [A1: ID is out of ranged]  [A2: ID is invalid]        6. User enter ‘y’.  [A3: User enter ‘n’]  [A4: User enters other character] | | | 1. The system shall display prompts to choose 1(View Contact Detail), 2(Remove Contact), 3(Edit Contact), 4(Sort), 0(Back).    3. The system shall display prompts to enter ID.        5. The system shall display the alert message “Are you sure about that? y/n”.        7. The system shall remove the selected contact.  8. The system shall display the message “Contact Deleted”.    9. The system shall redirect “Contact Menu”. | |
| Alternative Flow | [A1: ID is out of ranged]   1. The system shall display the message “Invalid ID”.   [A2; User enter character]   1. The system shall display the message “Enter only number”.   [A3: User enter ‘n’]   1. The system shall redirect to the “Contact Menu”.   [A4: User enters other character]   1. The system shall display the message “Invalid input”. 2. Go back to normal flow 5. | | | | |
| Exception Flow | - | | | | |
| Assumption | - | | | | |
|  |  |  |  |  |  |

[URS-09]: User can edit the specific contact.

[SRS-35] The system shall display prompts to choose 1(View Contact Detail), 2(Remove Contact), 3(Update Contact Information), 4(Sort), 0(Back).

[SRS-36] The system shall display prompts to enter ID.

[SRS-37] The system shall display prompt to choose which field to edit 1(Name), 2(Phone Number), 3(Email), 4(Address), 0(Back).

[SRS-38] The system shall update the selected field of the contact.

[SRS-39] The system shall display the message “[field] Edited”.

[SRS-40] Go back to normal flow 5.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use Case ID | UC-09 | | | | |
| Use Case Name | Edit Contact | | | | |
| Created By | JAVAC | | Last Update By | |  |
| Date Created | 02/02/2024 | | Last Revision Date | |  |
| Actors | User | | | | |
| Description | User can edit the specific contact. | | | | |
| Trigger | User enters “3” in “Contact Menu” page. | | | | |
| Preconditions | UC-04 or UC-05 | | | | |
| Use Case Input Specification | | | | | |
| Input | Type | Constraint | | | Example |
| ID | Integer | ID must be in the range of 1 to the number of contacts saved. | | | 2 |
| Post Conditions | The contact information will be changed. | | | | |
| Normal Flows | User | | | System | |
|  | 2. User enters “3”.      4. User enters ID.  [A1: ID is out of ranged]  [A2; ID is invalid]          6. User enter 1-4.  [A3: User enter 0]  [A4: User enter other than 0-4] | | | 1. The system shall display prompts to choose 1(View Contact Detail), 2(Remove Contact), 3(Edit Contact), 4(Sort), 0(Back).    3. The system shall display prompts to enter ID.        5. The system shall display prompt to choose which field to edit 1(Name), 2(Phone Number), 3(Email), 4(Address), 0(Back).      7. The system shall update the selected field of the contact.    8. The system shall display the message “[field] Edited”.    9. Go back to normal flow 5. | |
| Alternative Flow | [A1: ID is out of ranged]   1. The system shall display the message “Invalid ID”.   [A2; User enter character]   1. The system shall display the message “Enter only number”.   [A3: User enter 0]   1. The system shall redirect to the “Contact Menu”.   [A4: User enter other than 0-4]   1. The system shall display “Invalid input”. | | | | |
| Exception Flow | - | | | | |
| Assumption | - | | | | |
|  |  |  |  |  |  |

[URS-10]: User can edit the specific contact.

[SRS-41] The system shall display prompts to choose 1(View Contact Detail), 2(Remove Contact), 3(Update Contact Information), 4(Sort), 0(Back).

[SRS-42] The system shall display prompts to choose 1(Sort by Name) and 2(Sort by Date Created).

[SRS-43] The system shall sort accordingly.

[SRS-44] The system shall redirect to “Contact Menu”.

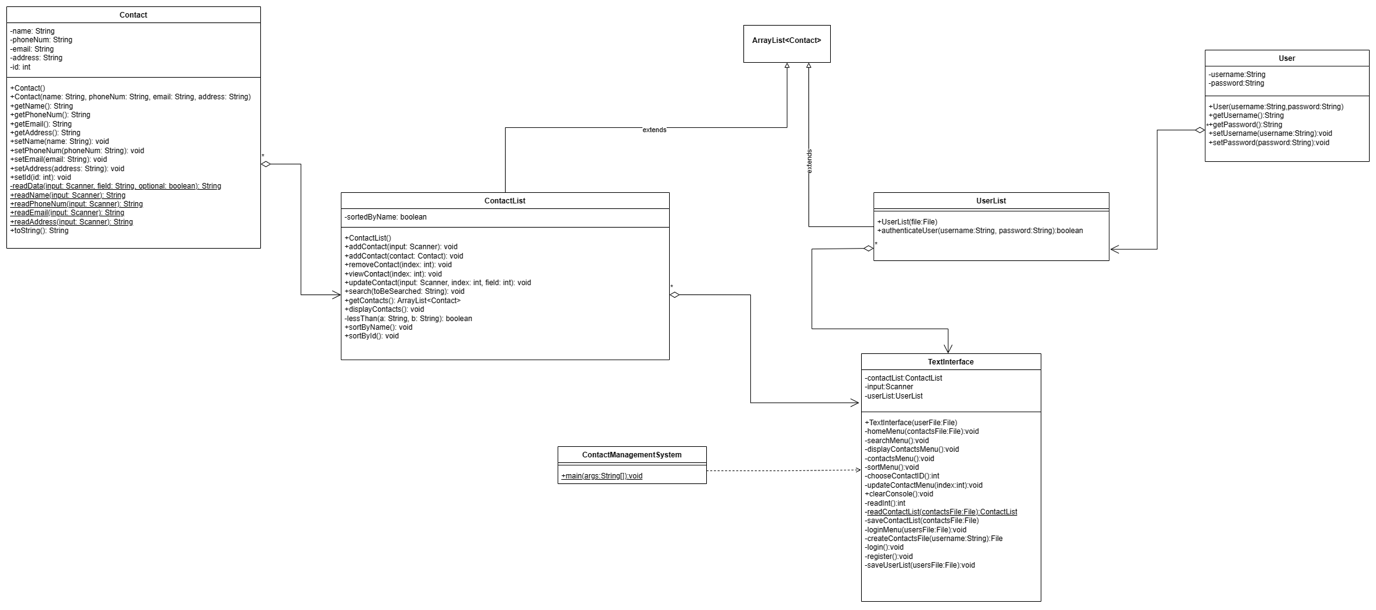
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use Case ID | UC-10 | | | | |
| Use Case Name | Sort | | | | |
| Created By | JAVAC | | Last Update By | |  |
| Date Created | 02/02/2024 | | Last Revision Date | |  |
| Actors | User | | | | |
| Description | User can sort the contact. | | | | |
| Trigger | User enters “4” in “Contact Menu” page. | | | | |
| Preconditions | UC-04 or UC-05 | | | | |
| Use Case Input Specification | | | | | |
| Input | Type | Constraint | | | Example |
| Sort type | Integer | Sort type must be 1 or 2. | | | 1 |
| Post Conditions | The contact list will be sorted.  The system shall redirect to contact menu. | | | | |
| Normal Flows | User | | | System | |
|  | 2. User enters “4”.          4. User enters 1 or 2.  [A1: User enter number except 1 or 2] | | | 1. The system shall display prompts to choose 1(View Contact Detail), 2(Remove Contact), 3(Edit Contact), 4(Sort), 0(Back).    3. The system shall display prompts to choose 1(Sort by Name) and 2(Sort by Date Created).        7. The system shall sort accordingly.    8. The system shall redirect to “Contact Menu”. | |
| Alternative Flow | [A1: User enter number except 1 or 2]   1. The system shall display the message “Invalid input”. | | | | |
| Exception Flow | - | | | | |
| Assumption | - | | | | |
|  |  |  |  |  |  |

[URS-11]: User can exit the program.

[SRS-41] The system shall display prompts to choose 1(Login), 2(Register), 0(Exit). [SRS-42] The program will be terminated.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use Case ID | UC-11 | | | |
| Use Case Name | Exit | | | |
| Created By | JAVAC | Last Update By | |  |
| Date Created | 02/02/2024 | Last Revision Date | |  |
| Actors | User | | | |
| Description | User can exit the program. | | | |
| Trigger | User enters “0” in “Login/ Register” page. | | | |
| Preconditions | - | | | |
| Post Conditions | The program will be terminated. | | | |
| Normal Flows | User | | System | |
|  | 2. User enters “0”. | | 1. The system shall display prompts to choose 1(Login), 2(Register), 0(Exit).      3. The program will be terminated. | |
| Alternative Flow | - | | | |
| Exception Flow | - | | | |
| Assumption | - | | | |
|  |  |  |  |  |

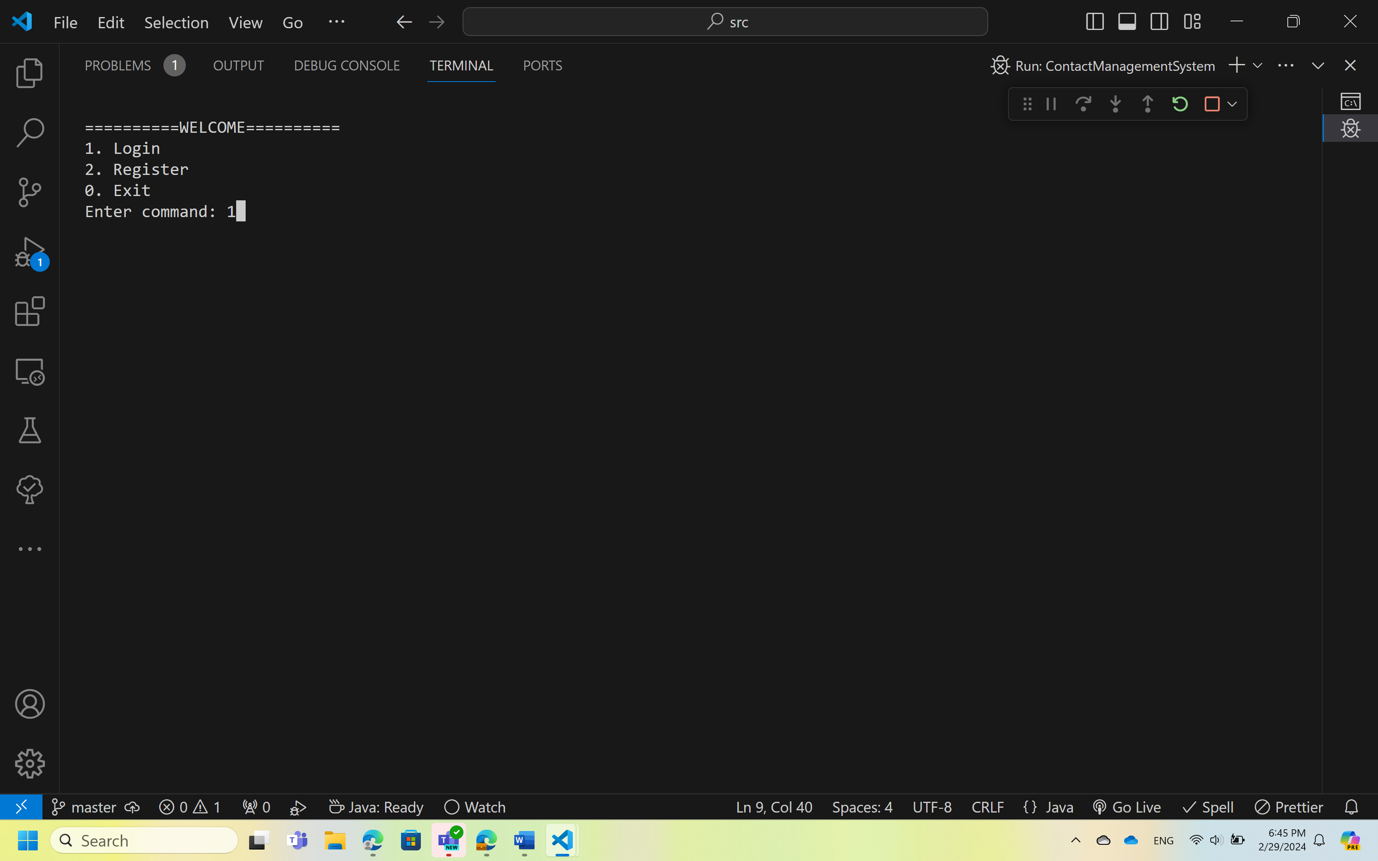
**Design:**

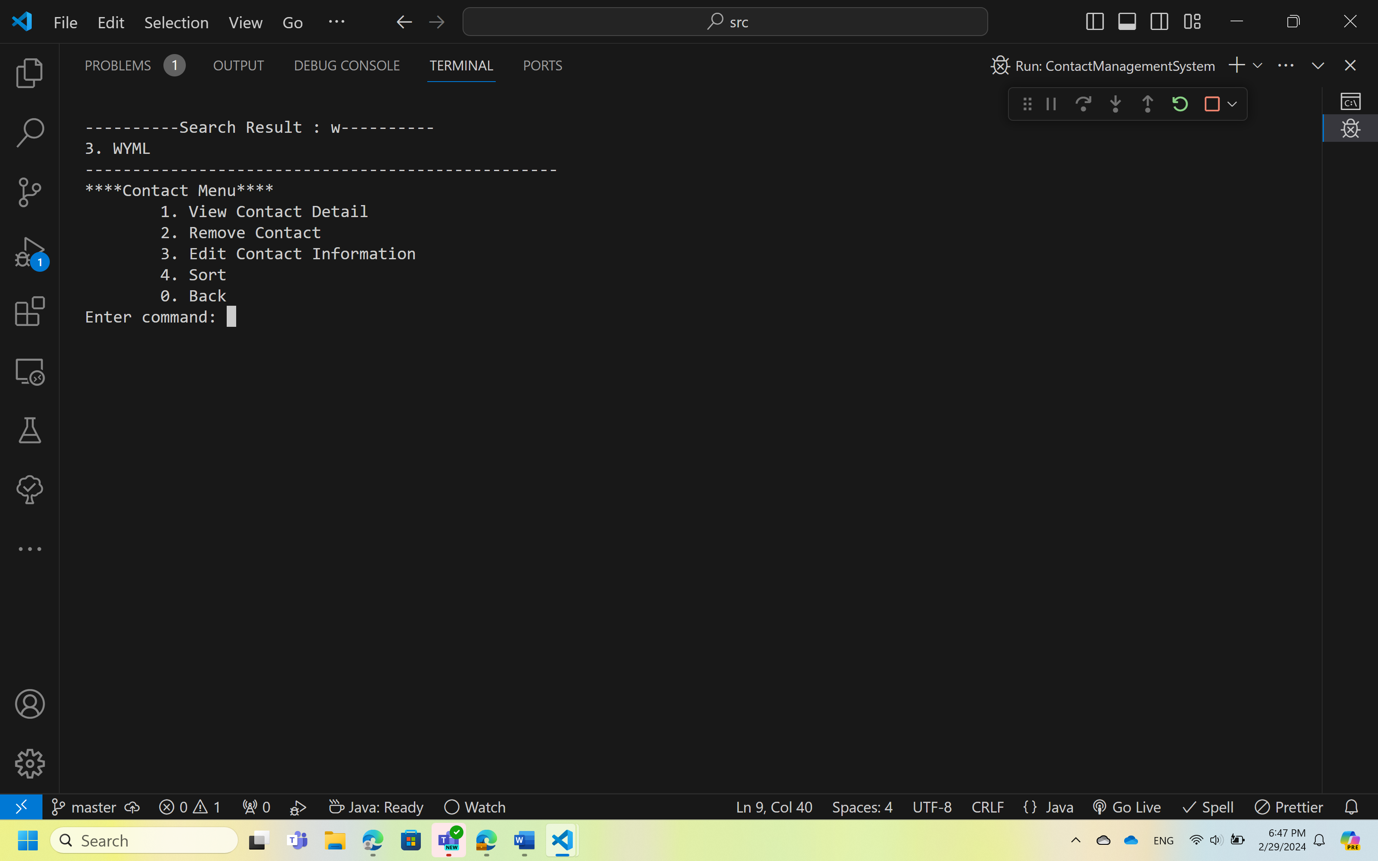
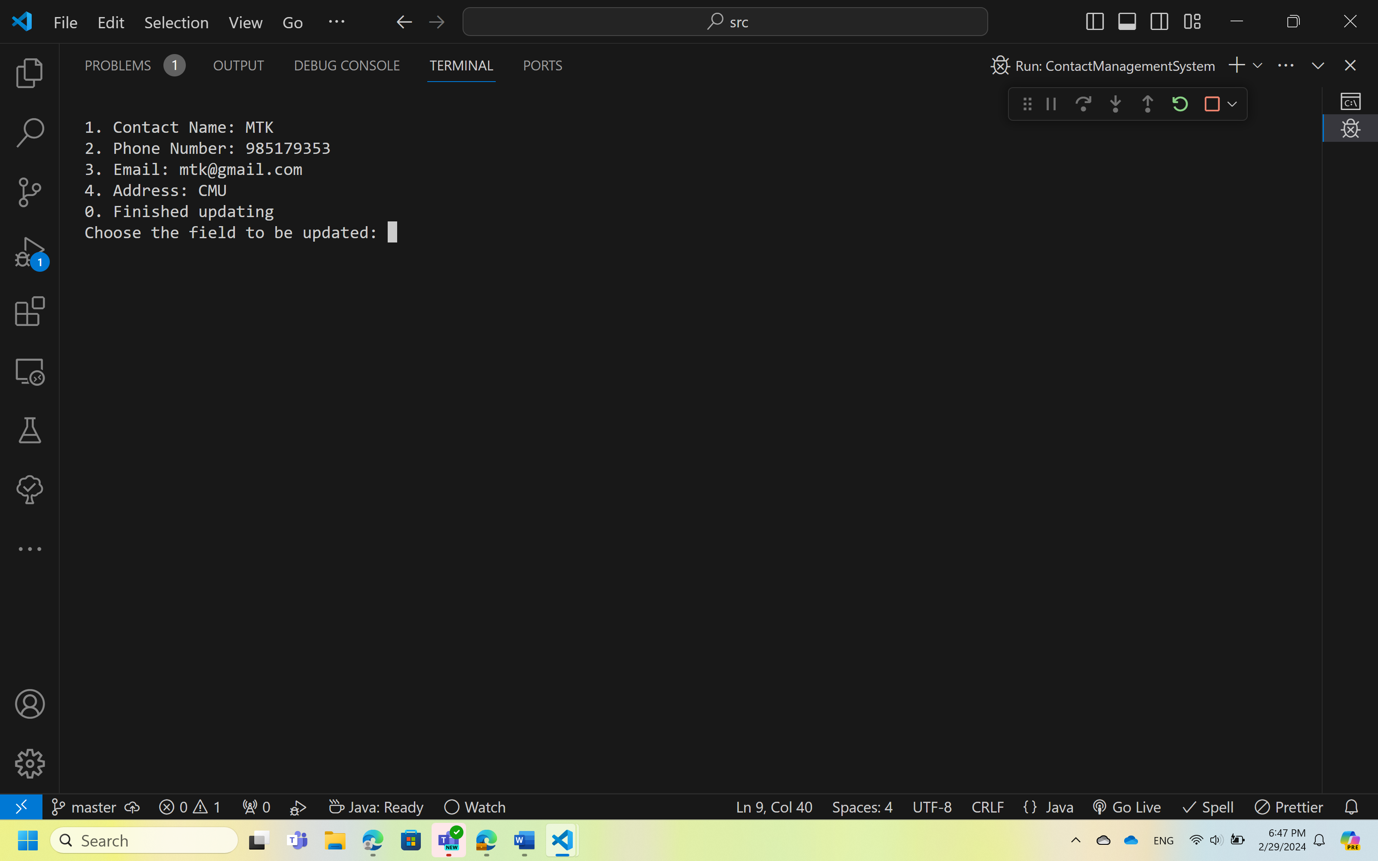
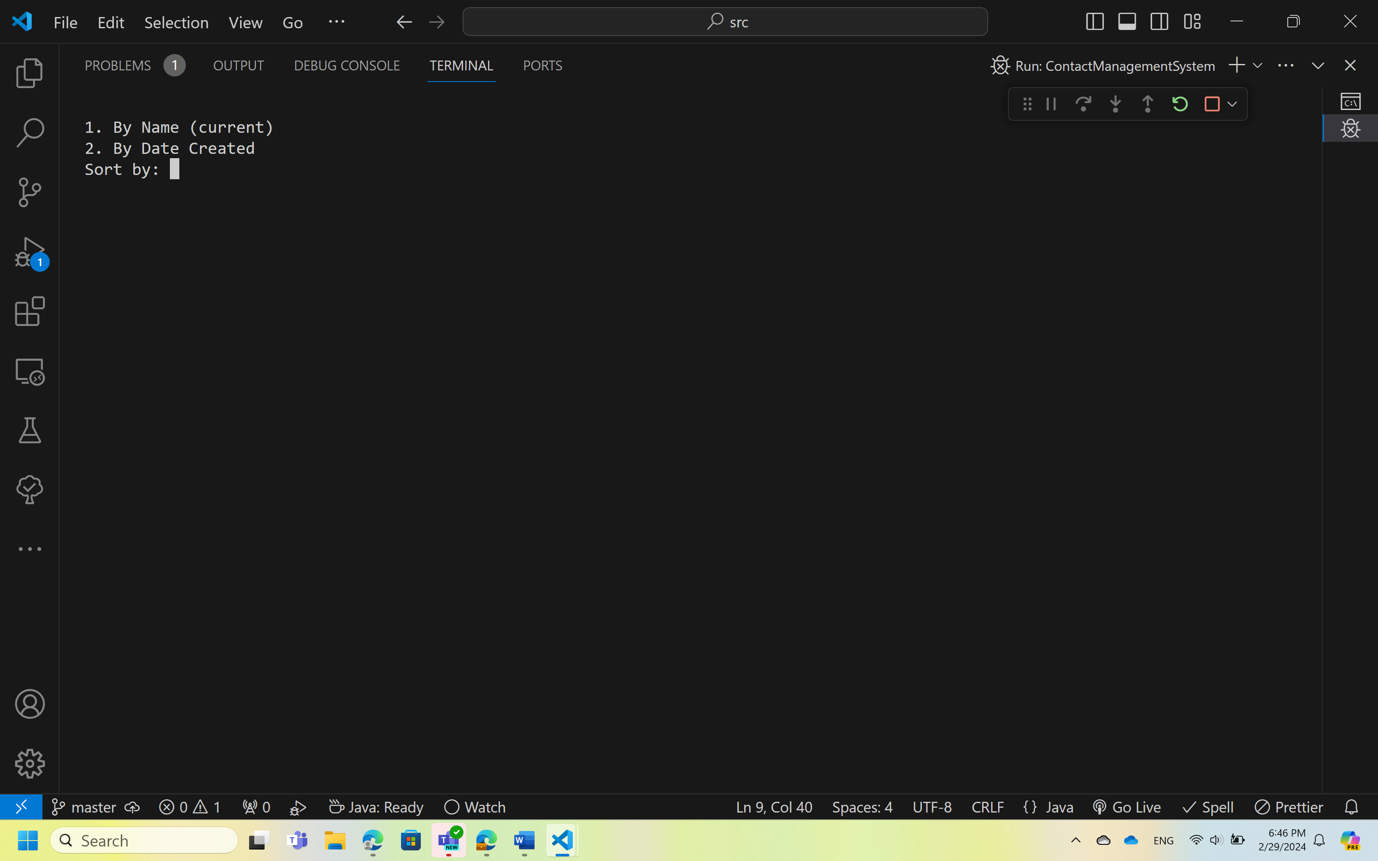
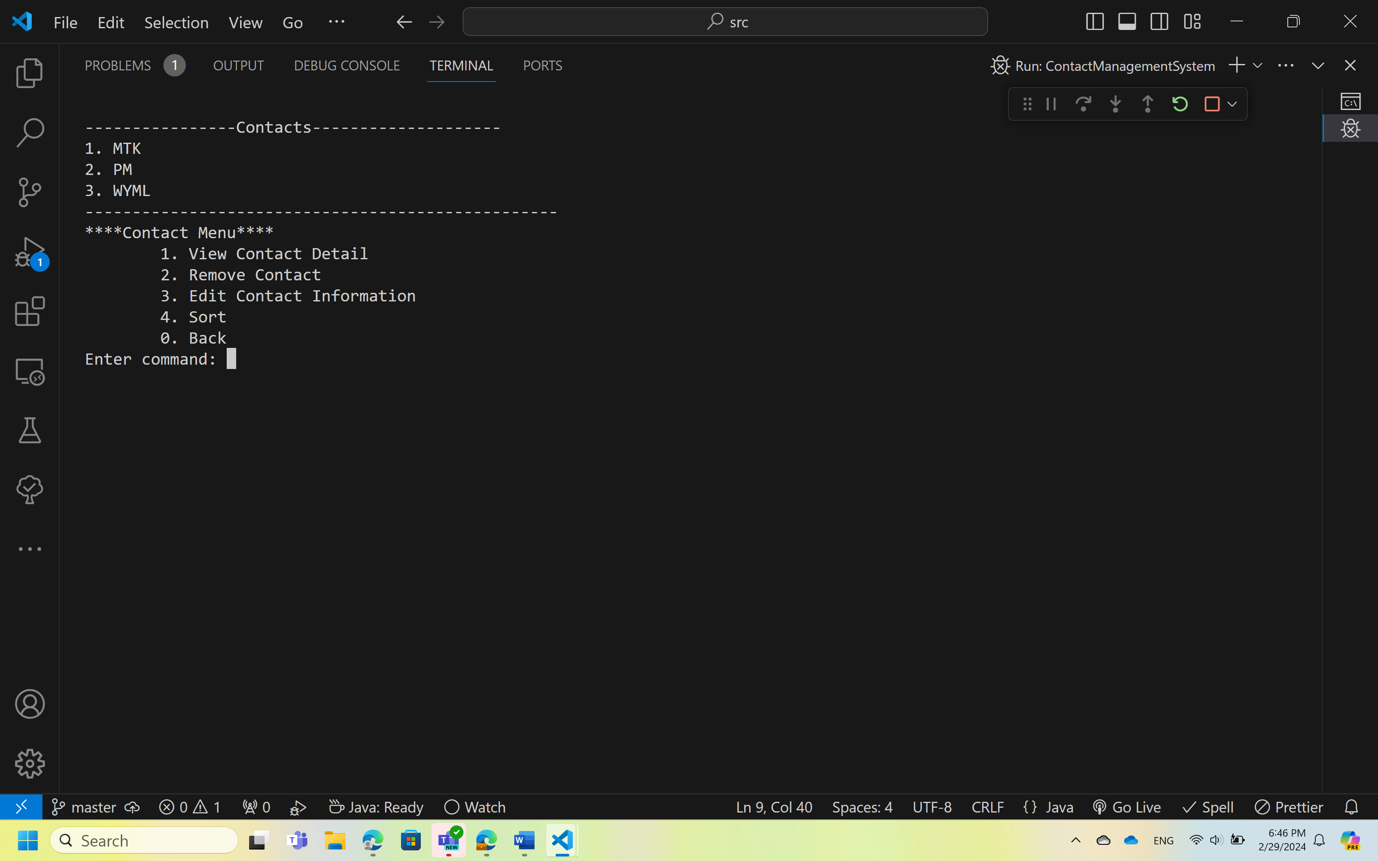
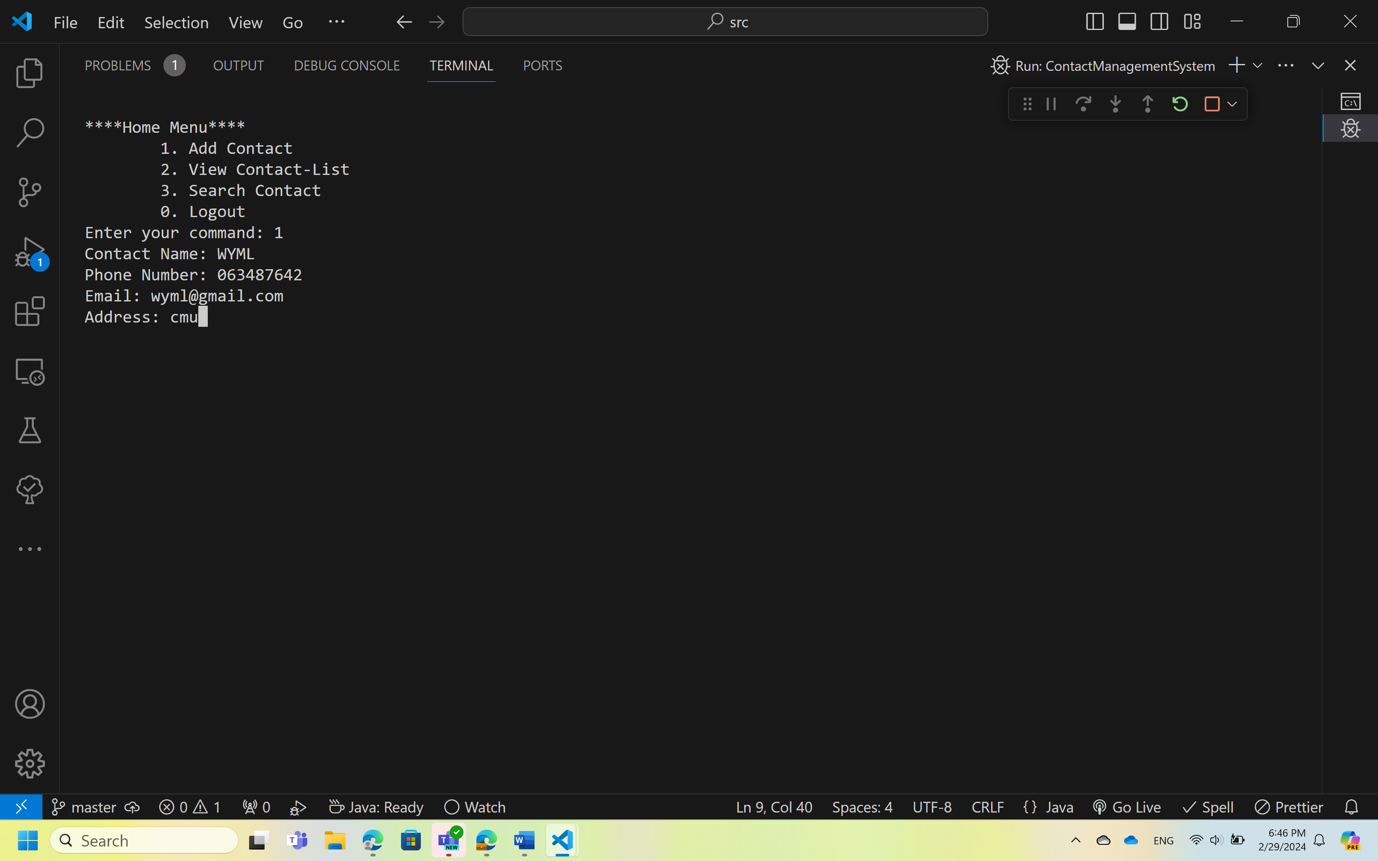
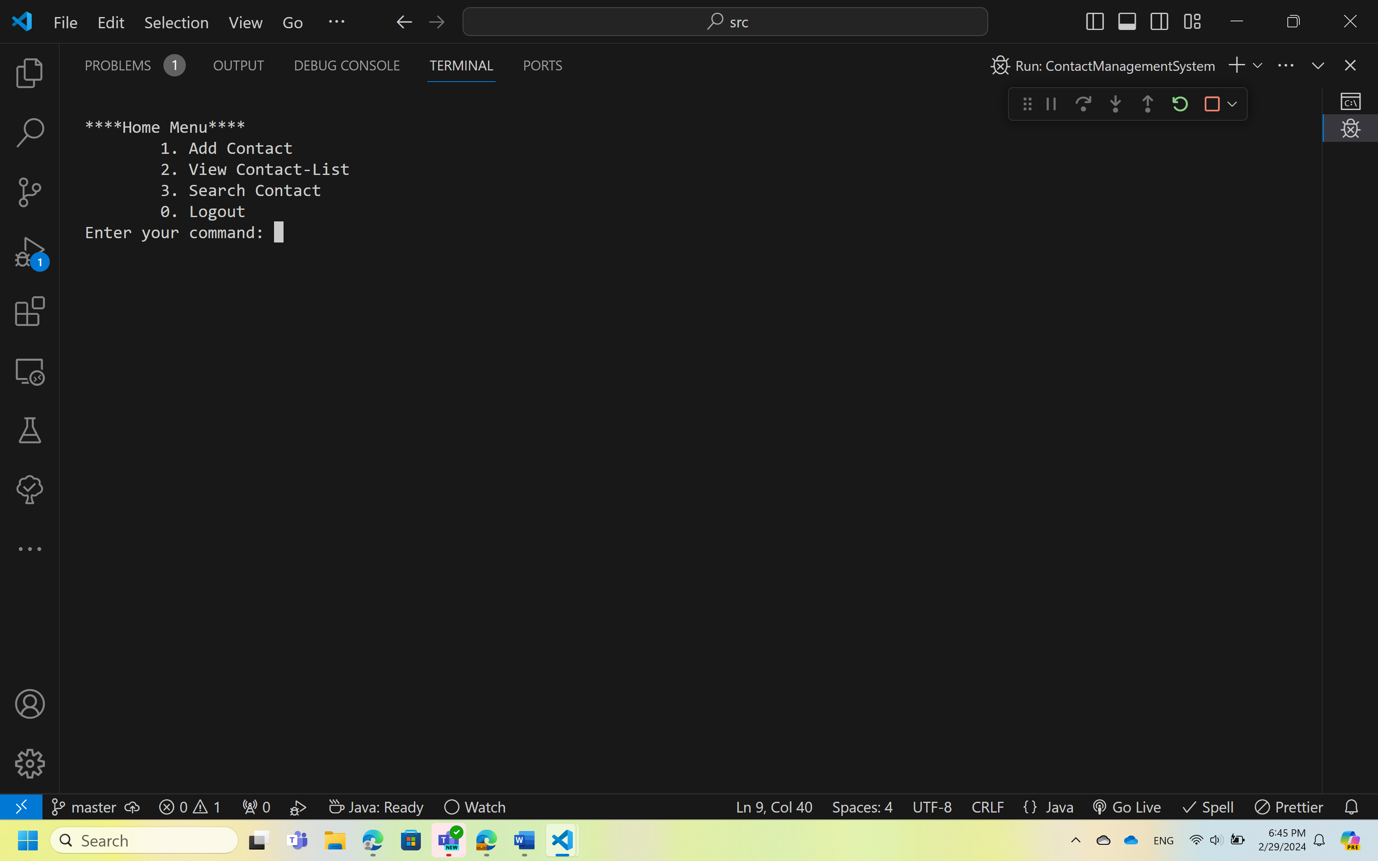
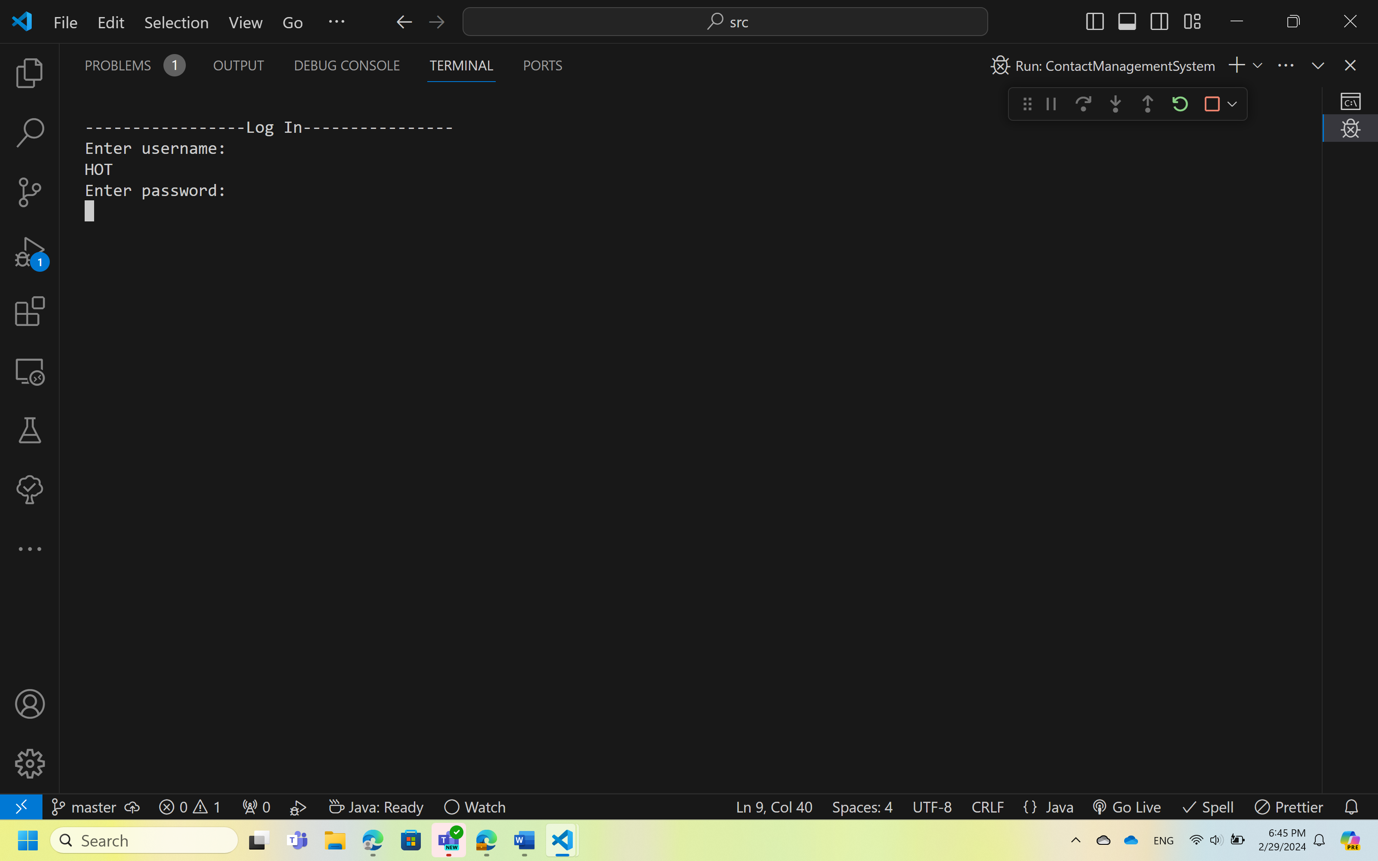


[Link to Class Diagram](https://github.com/WaiYanMoeLwin/ContactManagementSystem-CMU-SE-Project/blob/main/design/ClassDiagram.png)

**Implementation:**

* Screenshots of the JAVA text-based program:





[Link to Project Repository](https://github.com/WaiYanMoeLwin/ContactManagementSystem-CMU-SE-Project.git)

**Test plan of Unit testing and Feature testing:**

**Unit Testing Plan**

**Class - User**

Unit to be tested: getUsername(): String

Related Requirements: SRS-6

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Case Description | Input Data | Expected Outcome |
| UTC-1 | Test getting username | “Htet Oo Thu” | “Htet Oo Thu” |
|  |  |  |  |

Unit to be tested: getPassword(): String

Related Requirements: SRS-6

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Case Description | Input Data | Expected Outcome |
| UTC-2 | Test getting password | “12345” | “12345” |
|  |  |  |  |

Unit to be tested: setUsername(String username): void

Related Requirements: SRS-6, SRS-3

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Case Description | Input Data | Expected Outcome |
| UTC-3 | Test setting username | “Htet Oo Thu” | “Htet Oo Thu” |
|  |  |  |  |

Unit to be tested: setPassword(String password): void

Related Requirements: SRS-6, SRS-3

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Case Description | Input Data | Expected Outcome |
| UTC-4 | Test setting password | “12345” | “12345” |
|  |  |  |  |

**Class – UserList**

Unit to be tested: authenticateUser(String username, String password): boolean

Related Requirements: SRS-6

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Case Description | Input Data | Expected Outcome |
| UTC-5 | Verify correct username and password return true  Precondition: User already registered | Username: “Htet Oo Thu”  Password: “12345” | True |
| UTC-6 | Verify non-existing username and correct password return false  Precondition: User already registered | Username: “Wai Yan”  Password: “12345” | False |
| UTC-7 | Verify correct username and incorrect password return false  Precondition: User already registered | Username: “Htet Oo Thu”  Password: “54321” | False |
|  |  |  |  |

**Class- Contact**

Unit to be tested: getName(): String

Related Requirements: SRS-13, SRS-18, SRS-27, SRS-37, SRS-42

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-8 | Test getting contact name | “Wai Yan” | “Wai Yan” |
|  |  |  |  |

Unit to be tested: getPhoneNumber(): String

Related Requirements: SRS-27, SRS-37

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-9 | Test getting phone number | “061234567” | “061234567” |
|  |  |  |  |

Unit to be tested: getEmail(): String

Related Requirements: SRS-27, SRS-37

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-10 | Test getting email | “[waiyanmoelwin\_w@cmu.ac.th](mailto:waiyanmoelwin_w@cmu.ac.th)” | “[waiyanmoelwin\_w@cmu.ac.th](mailto:waiyanmoelwin_w@cmu.ac.th)” |
|  |  |  |  |

Unit to be tested: getAddress(): String

Related Requirements: SRS-27, SRS-37

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-11 | Test getting address | “Student Dorm” | “Student Dorm” |
|  |  |  |  |

Unit to be tested: getId(): int

Related Requirements: SRS-27, SRS-37

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-12 | Test getting id which is to note order of creation | 1 | 1 |
|  |  |  |  |

Unit to be tested: setName(String name): void

Related Requirements: SRS-10, SRS-38

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-13 | Test setting contact name | “Phone Myat” | “Phone Myat” |
|  |  |  |  |

Unit to be tested: setPhoneNumber(String phoneNumber): void

Related Requirements: SRS-10, SRS-38

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-14 | Test setting phone number | “061765432” | “061765432” |
|  |  |  |  |

Unit to be tested: setEmail(String email): void

Related Requirements: SRS-10, SRS-38

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-15 | Test setting email | “[someemail@gmail.com](mailto:someemail@gmail.com)” | “[someemail@gmail.com](mailto:someemail@gmail.com)” |
|  |  |  |  |

Unit to be tested: setAddress(String address): void

Related Requirements: SRS-10, SRS-38

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-16 | Test setting address | “CMU” | “CMU” |
|  |  |  |  |

Unit to be tested: setId(int id): void

Related Requirements: SRS-10

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-17 | Test setting id | 7 | 7 |
|  |  |  |  |

Unit to be tested: readData(Scanner input, String field, boolean optional): String

Related Requirements: SRS-9, SRS-37

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-18 | Test reading string data from console printing prompt | input:“Some String”  field: “field”  optional : false | System.out-“field: ”  “Some String” |
| UTC-19 | Test reading empty string from console while optional is false | input:“”  field: “field”  optional : false | System.out-“field: ”  System.out-“field cannot be empty.”  (prompt again) |
| UTC-20 | Test reading empty string from console while optional is true | input:“”  field: “field”  optional : false | System.out-“field: ”  “” |

Unit to be tested: readName(): String

Related Requirements: SRS-9, SRS-37

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-21 | Test reading contact name from console printing prompt | “Phone Myat” | “Phone Myat” |
|  |  |  |  |

Unit to be tested: readPhoneNum(): String

Related Requirements: SRS-9, SRS-37

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-22 | Test reading contact Phone number from console printing prompt | “0912345” | “0912345” |
|  |  |  |  |

Unit to be tested: readEmail(): String

Related Requirements: SRS-9, SRS-37

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-23 | Test reading contact email from console printing prompt | “[phone@gmail.com](mailto:phone@gmail.com)” | “[phone@gmail.com](mailto:phone@gmail.com) |
|  |  |  |  |

Unit to be tested: readAddress(): String

Related Requirements: SRS-9, SRS-37

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-24 | Test reading contact address from console printing prompt | “Chiang Mai” | “Chiang Mai” |
|  |  |  |  |

**Class-TextInterface**

Unit to be tested: saveUserList(File userFile): void

Related Requirements: SRS-3

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-25 | Test saving registered user to the users database. | - | User’s credentials are saved to the database. |
|  |  |  |  |

Unit to be tested: chooseContactID(): int

Related Requirements: SRS-14

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-26 | Test reading contact id from the console prompt | 1 | 1 |
|  |  |  |  |

Unit to be tested: clearConsole(): void

Related Requirements: SRS-8,SRS-11

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-27 | Testing clearing the console after going to one page from another. | - | The console is clear and empty |
|  |  |  |  |

Unit to be tested: readInt(): int

Related Requirements:SRS-5, SRS-8,SRS-11,SRS-42

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-28 | Test reading integer from the console | 1 | 1 |
|  |  |  |  |

Unit to be tested: readContactList(): ContactList

Related Requirements: SRS-7

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-29 | Test reading contact list from the contact file. | - | Contact List |
|  |  |  |  |

Unit to be tested: createContactsFile(): File

Related Requirements: SRS-9, SRS-37

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-30 | Test creating file to store contacts | - | File |
|  |  |  |  |

**Class- ContactList**

Unit to be tested: lessThan(String a, String b): boolean

Related Requirements: SRS-43

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-31 | Test if the method returns expected Boolean result from comparing two strings | String a: ‘a’  String b:’b’ | True |
|  |  |  |  |

Unit to be tested: addContact(Scanner input): void

Related Requirements: SRS-10

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-32 | Test constructing a new contact object. | Name= “Phone”  Phone number= “093231”  Email= [p@gmail.com](mailto:p@gmail.com)  Address= “CMU” | Contact with input properties is created. |
|  |  |  |  |

Unit to be tested: addContact(Contact contact): void

Related Requirements: SRS-10

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-33 | Test new valid contact is added to the contact list. | Name= “Phone”  Phone number= “093231”  Email= [p@gmail.com](mailto:p@gmail.com)  Address= “CMU” | Contact is added to the contact list. |
| UTC-34 | Test a contact with existing name cannot be added to the contact list. | Name= “Phone”  Phone number= “093231”  Email= [p@gmail.com](mailto:p@gmail.com)  Address= “CMU” | The message “This contact name has already existed.”  is shown. |

Unit to be tested: removeContact(int index): void

Related Requirements: SRS-32

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-35 | Test removing the specified contact from the contact list. | Id=1 | Contact with Id 1 is removed from the contact list. |
|  |  |  |  |

Unit to be tested: viewContact(int index): void

Related Requirements: SRS-27

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-36 | Test viewing the specified contact’s details. | Id=1 | The details of contact whose Id is 1 is shown. |
|  |  |  |  |

Unit to be tested: updateContact(Scanner input, int index, int field): void

Related Requirements: SRS-38

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-37 | Test updating the specified contact’s name. | Index=1  Field=1 | Contact’s name is updated. |
| UTC-38 | Test updating the specified contact’s Phone number | Index=1  Field=2 | Contact’s phone number is updated. |
| UTC-39 | Test updating the specified contact’s email. | Index=1  Field=3 | Contact’s email is updated. |
| UTC-40 | Test updating the specified contact’s address. | Index=1  Field=4 | Contact’s address is updated. |
| UTC-41 | Test that user cannot update with invalid field input. | Index=1  Field=6 | The message “Invalid input” is shown. |
|  |  |  |  |

Unit to be tested: search(String keyword): void

Related Requirements: SRS-17, SRS-18

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-42 | Test searching the contact with keyword. | keyword = “Ph” | The list of contacts which has names starting with “Ph” is shown. |
| UTC-43 | Test searching the contact with non-existing keyword. | Keyword= “ Zi” | The message “No such contact” is shown. |
|  |  |  |  |

Unit to be tested: displayContacts(): void

Related Requirements: SRS-13

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-44 | Test displaying the contact list. | - | The contact list is shown. |
|  |  |  |  |

Unit to be tested: sortByName(): void

Related Requirements: SRS-43

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-45 | Test sorting the contact list by name in ascending order. | - | The contact list is sorted by name. |
|  |  |  |  |

Unit to be tested: sortById(): void

Related Requirements: SRS-43

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| UTC-46 | Test sorting the contact list by Id in ascending order. | - | The contact list is sorted by Id. |
|  |  |  |  |

**Feature Testing Plan**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome |
| FTC-1 | Verify the user can access the registration feature.  Precondition: User is already on the Welcome page.  Steps:   1. User enters “2”. | - | Prompt for username and password will be shown. |
| FTC-2 | Verify the user can register with correct input.  Precondition: User is already on the Welcome page.  Steps:   1. User enters “2”. 2. Enter username and password. | Username = “HOT”  Password = “123!@” | User is added to the user list. |
| FTC-3 | Verify the user cannot register with duplicate name.  Precondition: User is already on the Welcome page.  Steps:   1. User enters “2”. 2. Enter duplicate username. | Username = “HOT” | The message “Username already exists” will be shown. |
| FTC-4 | Verify the user cannot register with empty name.  Precondition: User is already on the Welcome page.  Steps:   1. User enters “2”. 2. Enter empty username. | Username = “” | The message “Username cannot be empty.” will be shown. |
| FTC-5 | Verify the user cannot register with empty password.  Precondition: User is already on the Welcome page.  Steps:   1. User enters “2”. 2. Enter correct username and empty password. | Username = “HOT”  Password = “” | The message “Password cannot be empty” will be shown. |
| FTC-6 | Verify the user can access login feature.  Precondition: User is already on the Welcome page.  Steps:   1. User enters “1”. | - | Prompt for username and password will be shown. |
| FTC-7 | Verify the user can login with correct input.  Precondition: User is already on the Welcome page.  Steps:   1. User enters “1”. 2. Enter username and password. | Username = “HOT”  Password = “123!@” | User logged in.  Home Menu is shown. |
| FTC-8 | Verify the user cannot login with empty username.  Precondition: User is already on the Welcome page.  Steps:   1. User enters “1”. 2. User enters empty username. | Username = “” | The message “Username cannot be empty.” will be shown. |
| FTC-9 | Verify the user cannot login with empty password.  Precondition: User is already on the Welcome page.  Steps:   1. User enters “1”. 2. User enters username and empty password. | Username: “HOT”  Password: “” | The message “Password cannot be empty” will be shown. |
| FTC-10 | Verify the user cannot login with incorrect credentials.  Precondition: User is already on the Welcome page.  Steps:   1. User enters “1”. 2. User enters incorrect username and/or incorrect password. | Username: “TOH”  Password: “abc12” | The message “Incorrect username or password” will be shown. |
| FTC-11 | Verify the user can access add contact features.  Precondition: User already logged in.  Steps:   1. User enters “1”. | - | Prompt for contact name will be shown. |
| FTC-12 | Verify the user cannot add contact with empty contact name.  Precondition: User already logged in.  Steps:   1. User enters “1”. 2. User enters empty contact name. | Contact name: “” | The message “Contact name cannot be empty” will be shown. |
| FTC-13 | Verify the user cannot add contact with duplicate contact name.  Precondition: User already logged in.  Steps:   1. User enters “1”. 2. User enters duplicate contact name. | Contact name: “Bae” | The message “This contact name already existed. Update existing contact or create a new contact with different name.” will be shown. |
| FTC-14 | Verify the user cannot add contact with empty phone number.  Precondition: User already logged in.  Steps:   1. User enters “1”. 2. User enters contact name and empty phone number. | Contact name:  “Chit Lay”    Phone No.: “” | The message “Phone number cannot be empty” will be shown. |
| FTC-15 | Verify the user cannot add contact with valid contact information.  Precondition: User already logged in.  Steps:   1. User enters “1”. 2. User enters valid contact name, phone number, email and address. | Contac name:  “Ko Ko”    Phone No:  “098-7654321”    Email:  “[koko123@gmail.com](mailto:koko123@gmail.com)”    Address: Chiang Mai | A new contact with given information will be saved. |
| FTC-16 | Verify the user can access view contact list.  Precondition: User already logged in.  Steps:   1. User enters “2”. | - | Contact list and contact menu will be shown. |
| FTC-17 | Verify the user can access search contact feature.  Precondition: User already logged in.  Steps:   1. User enters “3”. | - | Prompt for keyword will be shown. |
| FTC-18 | Verify the user can search contact starting with keyword.  Precondition: User already logged in.  Steps:   1. User enters “3”. 2. User enters keyword. | Keyword= “k” | Contact names starting with “k”/ “K” will be shown |
| FTC-19 | Verify that the user will be informed there is no such contact when there is no contact starting with keyword.  Precondition: User already logged in.  Steps:   1. User enters “3”. 2. User enters keyword. | Keyword = “x” | The message “No such contact.” will be shown. |
| FTC-20 | Verify that the user can log out.  Precondition: User already logged in.  Steps:   1. User enters “0” | - | Welcome Page will be shown. |

|  |  |  |  |
| --- | --- | --- | --- |
| FTC-21 | Verify that the user can access the view contact details feature.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “1”.. | - | Prompt for contact ID will be shown. |
| FTC-22 | Verify that the user can view the existing contact details.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “1”.. 2. User enters valid ID. | ID = “4” | Contact name, Phone number, Email, and Address for contact with ID 4 will be shown. |
| FTC-23 | Verify that the user cannot view the contact details using invalid ID.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “1”. 2. User enters invalid ID. | ID = “420”  ID = “-2” | The message “Invalid ID” will be shown. |
| FTC-24 | Verify that the user can access the remove contact feature.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “2”. | - | Prompt for contact ID will be shown. |
| FTC-25 | Verify that the user can remove the existing contact.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “1”. 2. User enters valid ID. | ID = “4” | Contact with ID 4 will be deleted. |
| FTC-26 | Verify that the user cannot remove the contact using invalid ID.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “1”. 2. User enters invalid ID. | ID = “420”  ID = “-2” | The message “Invalid ID” will be shown. |
| FTC-27 | Verify that the user can access the edit contact feature.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “3”. | - | Prompt for contact ID will be shown. |
| FTC-28 | Verify that the user can edit the existing contact.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “3”. 2. User enters valid ID. | ID = “4” | Prompt to choose which field to edit will be shown. |
| FTC-29 | Verify that the user can edit contact name.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “3”. 2. User enters valid ID. 3. User chooses “1”. 4. User enters new name. | ID = “5”  Field = “1”  Name: “Ma Ma” | Contact name for contact with ID “5” will be changed to “Ma Ma”. |
| FTC-30 | Verify that the user can edit Phone number.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “3”. 2. User enters valid ID. 3. User chooses “2”. 4. User enters new phone number. | ID = “5”  Field = “2”  Phone Number:  “092-3456789” | Phone number for contact with ID “5” will be changed to “092-3456789”. |
| FTC-31 | Verify that the user can edit email.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “3”. 2. User enters valid ID. 3. User chooses “3”. 4. User enters new email. | ID = “5”  Field = “3”  Email: “[mama834@gmail.com](mailto:mama834@gmail.com)” | Email for contact with ID “5” will be changed to “[mama834@gmail.com](mailto:mama834@gmail.com)” |
| FTC-32 | Verify that the user can edit the address.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “3”. 2. User enters valid ID. 3. User chooses “4”. 4. User enters new address. | ID = “5”  Field = “4”  Address: “Bangkok” | Address for contact with ID “5” will be changed to “Bangkok”. |
| FTC-33 | Verify that the user cannot edit the contact using invalid ID.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “3”. 2. User enters invalid ID. | ID = “420”  ID = “-2” | The message “Invalid ID” will be shown. |
| FTC-34 | Verify that the user can access the sort of feature.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “4”. | - | Prompt to choose sort by name and sort by date will be shown. |
| FTC-35 | Verify that the user can sort contact list by name.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “4”. 2. User enters “1”. | - | Contact list will be sorted by name. |
| FTC-36 | Verify that the user can sort contact list by date.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “4”. 2. User enters “2”. | - | Contact list will be sorted by date. |
| FTC-37 | Verify that the user can exit the program.  Precondition: User is already on the Welcome page.  Steps:   1. User enters “0”. | - | Program will be terminated. |

**Test record of Unit testing and Feature testing:**

**Unit Testing Record**

**Class - User**

Unit to be tested: getUsername(): String

Related Requirements: SRS-6

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Case Description | Input Data | Expected Outcome | Actual Outcome | Result |
| UTC-1 | Test getting username | “Htet Oo Thu” | “Htet Oo Thu” | “Htet Oo Thu” | Pass |
|  |  |  |  |  |  |

Unit to be tested: getPassword(): String

Related Requirements: SRS-6

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Case Description | Input Data | Expected Outcome | Actual Outcome | Result |
| UTC-2 | Test getting password | “12345” | “12345” | “12345” | Pass |
|  |  |  |  |  |  |

Unit to be tested: setUsername(String username): void

Related Requirements: SRS-6, SRS-3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Case Description | Input Data | Expected Outcome | Actual Outcome | Result |
| UTC-3 | Test setting username | “Htet Oo Thu” | “Htet Oo Thu” | “Htet Oo Thu” | Pass |
|  |  |  |  |  |  |

Unit to be tested: setPassword(String password): void

Related Requirements: SRS-6, SRS-3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Case Description | Input Data | Expected Outcome | Actual Outcome | Result |
| UTC-4 | Test setting password | “12345” | “12345” | “12345” | Pass |
|  |  |  |  |  |  |

**Class – UserList**

Unit to be tested: authenticateUser(String username, String password): boolean

Related Requirements: SRS-6

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Case Description | Input Data | Expected Outcome | Actual Outcome | Result |
| UTC-5 | Verify correct username and password return true  Precondition: User already registered | Username: “Htet Oo Thu”  Password: “12345” | True | True | Pass |
| UTC-6 | Verify non-existing username and correct password return false  Precondition: User already registered | Username: “Wai Yan”  Password: “12345” | False | False | Pass |
| UTC-7 | Verify correct username and incorrect password return false  Precondition: User already registered | Username: “Htet Oo Thu”  Password: “54321” | False | False | Pass |
|  |  |  |  |  |  |

**Class- Contact**

Unit to be tested: getName(): String

Related Requirements: SRS-13, SRS-18, SRS-27, SRS-37, SRS-42

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual Outcome | Result |
| UTC-8 | Test getting contact name | “Wai Yan” | “Wai Yan” | “Wai Yan” | Pass |
|  |  |  |  |  |  |

Unit to be tested: getPhoneNumber(): String

Related Requirements: SRS-27, SRS-37

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual Outcome | Result |
| UTC-9 | Test getting phone number | “061234567” | “061234567” | “061234567” | Pass |
|  |  |  |  |  |  |

Unit to be tested: getEmail(): String

Related Requirements: SRS-27, SRS-37

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual Outcome | Result |
| UTC-10 | Test getting email | “[wai\_w@cmu.ac.th](mailto:wai_w@cmu.ac.th)” | “[wai\_w@cmu.ac.th](mailto:wai_w@cmu.ac.th)” | “[wai\_w@cmu.ac.th](mailto:wai_w@cmu.ac.th)” | Pass |
|  |  |  |  |  |  |

Unit to be tested: getAddress(): String

Related Requirements: SRS-27, SRS-37

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual Outcome | Result |
| UTC-11 | Test getting address | “Student Dorm” | “Student Dorm” | “Student Dorm” | Pass |
|  |  |  |  |  |  |

Unit to be tested: getId(): int

Related Requirements: SRS-27, SRS-37

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual Outcome | Result |
| UTC-12 | Test getting id which is to note order of creation | 1 | 1 | 1 | Pass |
|  |  |  |  |  |  |

Unit to be tested: setName(String name): void

Related Requirements: SRS-10, SRS-38

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual Outcome | Result |
| UTC-13 | Test setting contact name | “Phone Myat” | “Phone Myat” | “Phone Myat” | Pass |
|  |  |  |  |  |  |

Unit to be tested: setPhoneNumber(String phoneNumber): void

Related Requirements: SRS-10, SRS-38

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual Outcome | Result |
| UTC-14 | Test setting phone number | “061765432” | “061765432” | “061765432” | Pass |
|  |  |  |  |  |  |

Unit to be tested: setEmail(String email): void

Related Requirements: SRS-10, SRS-38

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual Outcome | Result |
| UTC-15 | Test setting email | “[someemail@gmail.com](mailto:someemail@gmail.com)” | “[someemail@gmail.com](mailto:someemail@gmail.com)” | “[someemail@gmail.com](mailto:someemail@gmail.com)” | Pass |
|  |  |  |  |  |  |

Unit to be tested: setAddress(String address): void

Related Requirements: SRS-10, SRS-38

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual Outcome | Result |
| UTC-16 | Test setting address | “CMU” | “CMU” | “CMU” | Pass |
|  |  |  |  |  |  |

Unit to be tested: setId(int id): void

Related Requirements: SRS-10

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual Outcome | Result |
| UTC-17 | Test setting id | 7 | 7 | 7 | Pass |
|  |  |  |  |  |  |

Unit to be tested: readData(Scanner input, String field, boolean optional): String

Related Requirements: SRS-9, SRS-37

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual Outcome | Result |
| UTC-18 | Test reading string data from console printing prompt | input:“Some String”  field: “field”  optional : false | System.out-“field: ”  “Some String” | System.out-“field: ”  “Some String” | Pass |
| UTC-19 | Test reading empty string from console while optional is false | input:“”  field: “field”  optional : false | System.out-“field: ”  System.out-“field cannot be empty.”  (prompt again) | System.out-“field: ”  System.out-“field cannot be empty.”  (prompt again) | Pass |
| UTC-20 | Test reading empty string from console while optional is true | input:“”  field: “field”  optional : false | System.out-“field: ”  “” | System.out-“field: ”  “” | Pass |

Unit to be tested: readName(): String

Related Requirements: SRS-9, SRS-37

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual Outcome | Result |
| UTC-21 | Test reading contact name from console printing prompt | “Phone Myat” | “Phone Myat” | “Phone Myat” | Pass |

Unit to be tested: readPhoneNum(): String

Related Requirements: SRS-9, SRS-37

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual Outcome | Result |
| UTC-22 | Test reading contact Phone number from console printing prompt | “0912345” | “0912345” | “0912345” | Pass |
|  |  |  |  |  |  |

Unit to be tested: readEmail(): String

Related Requirements: SRS-9, SRS-37

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual Outcome | Result |
| UTC-23 | Test reading contact email from console printing prompt | “[phone@gmail.com](mailto:phone@gmail.com)” | “[phone@gmail.com](mailto:phone@gmail.com) | “[phone@gmail.com](mailto:phone@gmail.com) | Pass |
|  |  |  |  |  |  |

Unit to be tested: readAddress(): String

Related Requirements: SRS-9, SRS-37

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual Outcome | Result |
| UTC-24 | Test reading contact address from console printing prompt | “Chiang Mai” | “Chiang Mai” | “Chiang Mai” | Pass |
|  |  |  |  |  |  |

**Class-TextInterface**

Unit to be tested: saveUserList(File userFile): void

Related Requirements: SRS-3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual  Outcome | Result |
| UTC-25 | Test saving registered user to the users database. | - | User’s credentials are saved to the database. | User’s credentials are saved to the database. | Pass |
|  |  |  |  |  |  |

Unit to be tested: chooseContactID(): int

Related Requirements: SRS-14

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual Outcome | Result |
| UTC-26 | Test reading contact id from the console prompt | 1 | 1 | 1 | Pass |
|  |  |  |  |  |  |

Unit to be tested: clearConsole(): void

Related Requirements: SRS-8,SRS-11

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual Outcome | Result |
| UTC-27 | Testing clearing the console after going to one page from another. | - | The console is clear and empty | The console is clear and empty | Pass |
|  |  |  |  |  |  |

Unit to be tested: readInt(): int

Related Requirements:SRS-5, SRS-8,SRS-11,SRS-42

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual Outcome | Result |
| UTC-28 | Test reading integer from the console | 1 | 1 | 1 | Pass |

Unit to be tested: readContactList(): ContactList

Related Requirements: SRS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual Outcome | Result |
| UTC-29 | Test reading contact list from the contact file. | - | Contact List | Contact List | Pass |
|  |  |  |  |  |  |

Unit to be tested: createContactsFile(): File

Related Requirements: SRS-9, SRS-37

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual Outcome | Result |
| UTC-30 | Test creating file to store contacts | - | File | File | Pass |
|  |  |  |  |  |  |

**Class- ContactList**

Unit to be tested: lessThan(String a, String b): boolean

Related Requirements: SRS-43

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual  Outcome | Result |
| UTC-31 | Test if the method returns expected Boolean result from comparing two strings | String a: ‘a’  String b:’b’ | True | True | Pass |
|  |  |  |  |  |  |

Unit to be tested: addContact(Scanner input): void

Related Requirements: SRS-10

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual  Outcome | Result |
| UTC-32 | Test constructing a new contact object. | Name= “Phone”  Phone number= “093231”  Email= [p@gmail.com](mailto:p@gmail.com)  Address= “CMU” | Contact with input properties is created. | Contact with input properties is created. | Pass |
|  |  |  |  |  |  |

Unit to be tested: addContact(Contact contact): void

Related Requirements: SRS-10

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual  Outcome | Result |
| UTC-33 | Test new valid contact is added to the contact list. | Name= “Phone”  Phone number= “093231”  Email= [p@gmail.com](mailto:p@gmail.com)  Address= “CMU” | Contact is added to the contact list. | Contact is added to the contact list. | Pass |
| UTC-34 | Test a contact with existing name cannot be added to the contact list. | Name= “Phone”  Phone number= “093231”  Email= [p@gmail.com](mailto:p@gmail.com)  Address= “CMU” | The message “This contact name has already existed.”  is shown. | The message “This contact name has already existed.”  is shown. | Pass |

Unit to be tested: removeContact(int index): void

Related Requirements: SRS-32

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual  Outcome | Result |
| UTC-35 | Test removing the specified contact from the contact list. | Id=1 | Contact with Id 1 is removed from the contact list. | Contact with Id 1 is removed from the contact list. | Pass |
|  |  |  |  |  |  |

Unit to be tested: viewContact(int index): void

Related Requirements: SRS-27

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual  Outcome | Result |
| UTC-36 | Test viewing the specified contact’s details. | Id=1 | The details of contact whose Id is 1 is shown. | The details of contact whose Id is 1 is shown. | Pass |
|  |  |  |  |  |  |

Unit to be tested: updateContact(Scanner input, int index, int field): void

Related Requirements: SRS-38

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual  Outcome | Result |
| UTC-37 | Test updating the specified contact’s name. | Index=1  Field=1 | Contact’s name is updated. | Contact’s name is updated. | Pass |
| UTC-38 | Test updating the specified contact’s Phone number | Index=1  Field=2 | Contact’s phone number is updated. | Contact’s phone number is updated. | Pass |
| UTC-39 | Test updating the specified contact’s email. | Index=1  Field=3 | Contact’s email is updated. | Contact’s email is updated. | Pass |
| UTC-40 | Test updating the specified contact’s address. | Index=1  Field=4 | Contact’s address is updated. | Contact’s address is updated. | Pass |
| UTC-41 | Test that user cannot update with invalid field input. | Index=1  Field=6 | The message “Invalid input” is shown. | The message “Invalid input” is shown. | Pass |
|  |  |  |  |  |  |

Unit to be tested: search(String keyword): void

Related Requirements: SRS-17, SRS-18

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual  Outcome | Result |
| UTC-42 | Test searching the contact with keyword. | keyword = “Ph” | The list of contacts which has names starting with “Ph” is shown. | The list of contacts which has names starting with “Ph” is shown. | Pass |
| UTC-43 | Test searching the contact with non-existing keyword. | Keyword= “ Zi” | The message “No such contact” is shown. | The message “No such contact” is shown. | Pass |
|  |  |  |  |  |  |

Unit to be tested: displayContacts(): void

Related Requirements: SRS-13

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual Outcome | Result |
| UTC-44 | Test displaying the contact list. | - | The contact list is shown. | The contact list is shown. | Pass |
|  |  |  |  |  |  |

Unit to be tested: sortByName(): void

Related Requirements: SRS-43

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual Outcome | Result |
| UTC-45 | Test sorting the contact list by name in ascending order. | - | The contact list is sorted by name. | The contact list is sorted by name. | Pass |
|  |  |  |  |  |  |

Unit to be tested: sortById(): void

Related Requirements: SRS-43

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual  Outcome | Result |
| UTC-46 | Test sorting the contact list by Id in ascending order. | - | The contact list is sorted by Id. | The contact list is sorted by Id. | Pass |
|  |  |  |  |  |  |

**Feature Testing Record**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case# | Test Description | Input Data | Expected Outcome | Actual Outcome | Result |
| FTC-1 | Verify the user can access the registration feature.  Precondition: User is already on the Welcome page.  Steps:   1. User enters “2”. | - | Prompt for username and password will be shown. | Prompt for username and password will be shown. | Pass |
| FTC-2 | Verify the user can register with correct input.  Precondition: User is already on the Welcome page.  Steps:   1. User enters “2”. 2. Enter username and password. | Username = “HOT”  Password = “123!@” | User is added to the user list. | User is added to the user list. | Pass |
| FTC-3 | Verify the user cannot register with duplicate name.  Precondition: User is already on the Welcome page.  Steps:   1. User enters “2”. 2. Enter duplicate username. | Username = “HOT” | The message “Username already exists” will be shown. | The message “Username already exists” will be shown. | Pass |
| FTC-4 | Verify the user cannot register with empty name.  Precondition: User is already on the Welcome page.  Steps:   1. User enters “2”. 2. Enter empty username. | Username = “” | The message “Username cannot be empty.” will be shown. | The message “Username cannot be empty.” will be shown. | Pass |
| FTC-5 | Verify the user cannot register with empty password.  Precondition: User is already on the Welcome page.  Steps:   1. User enters “2”. 2. Enter correct username and empty password. | Username = “HOT”  Password = “” | The message “Password cannot be empty” will be shown. | The message “Password cannot be empty” will be shown. | Pass |
| FTC-6 | Verify the user can access login feature.  Precondition: User is already on the Welcome page.  Steps:   1. User enters “1”. | - | Prompt for username and password will be shown. | Prompt for username and password will be shown. | Pass |
| FTC-7 | Verify the user can login with correct input.  Precondition: User is already on the Welcome page.  Steps:   1. User enters “1”. 2. Enter username and password. | Username = “HOT”  Password = “123!@” | User logged in.  Home Menu is shown. | User logged in.  Home Menu is shown. | Pass |
| FTC-8 | Verify the user cannot login with empty username.  Precondition: User is already on the Welcome page.  Steps:   1. User enters “1”. 2. User enters empty username. | Username = “” | The message “Username cannot be empty.” will be shown. | The message “Username cannot be empty.” will be shown. | Pass |
| FTC-9 | Verify the user cannot login with empty password.  Precondition: User is already on the Welcome page.  Steps:   1. User enters “1”. 2. User enters username and empty password. | Username: “HOT”  Password: “” | The message “Password cannot be empty” will be shown. | The message “Password cannot be empty” will be shown. | Pass |
| FTC-10 | Verify the user cannot login with incorrect credentials.  Precondition: User is already on the Welcome page.  Steps:   1. User enters “1”. 2. User enters incorrect username and/or incorrect password. | Username: “TOH”  Password: “abc12” | The message “Incorrect username or password” will be shown. | The message “Incorrect username or password” will be shown. | Pass |
| FTC-11 | Verify the user can access add contact features.  Precondition: User already logged in.  Steps:   1. User enters “1”. | - | Prompt for contact name will be shown. | Prompt for contact name will be shown. | Pass |
| FTC-12 | Verify the user cannot add contact with empty contact name.  Precondition: User already logged in.  Steps:   1. User enters “1”. 2. User enters empty contact name. | Contact name: “” | The message “Contact name cannot be empty” will be shown. | The message “Contact name cannot be empty” will be shown. | Pass |
| FTC-13 | Verify the user cannot add contact with duplicate contact name.  Precondition: User already logged in.  Steps:   1. User enters “1”. 2. User enters duplicate contact name. | Contact name: “Bae” | The message “This contact name already existed. Update existing contact or create a new contact with different name.” will be shown. | The message “This contact name already existed. Update existing contact or create a new contact with different name.” will be shown. | Pass |
| FTC-14 | Verify the user cannot add contact with empty phone number.  Precondition: User already logged in.  Steps:   1. User enters “1”. 2. User enters contact name and empty phone number. | Contact name:  “Chit Lay”    Phone No.: “” | The message “Phone number cannot be empty” will be shown. | The message “Phone number cannot be empty” will be shown. | Pass |
| FTC-15 | Verify the user cannot add contact with valid contact information.  Precondition: User already logged in.  Steps:   1. User enters “1”. 2. User enters valid contact name, phone number, email and address. | Contac name:  “Ko Ko”    Phone No:  “098-7654321”    Email:  “[koko123@gmail.com](mailto:koko123@gmail.com)”    Address: Chiang Mai | A new contact with given information will be saved. | A new contact with given information will be saved. | Pass |
| FTC-16 | Verify the user can access view contact list.  Precondition: User already logged in.  Steps:   1. User enters “2”. | - | Contact list and contact menu will be shown. | Contact list and contact menu will be shown. | Pass |
| FTC-17 | Verify the user can access search contact feature.  Precondition: User already logged in.  Steps:   1. User enters “3”. | - | Prompt for keyword will be shown. | Prompt for keyword will be shown. | Pass |
| FTC-18 | Verify the user can search contact starting with keyword.  Precondition: User already logged in.  Steps:   1. User enters “3”. 2. User enters keyword. | Keyword= “k” | Contact names starting with “k”/ “K” will be shown | Contact names starting with “k”/ “K” will be shown | Pass |
| FTC-19 | Verify that the user will be informed there is no such contact when there is no contact starting with keyword.  Precondition: User already logged in.  Steps:   1. User enters “3”. 2. User enters keyword. | Keyword = “x” | The message “No such contact.” will be shown. | The message “No such contact.” will be shown. | Pass |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| FTC-20 | Verify that the user can log out.  Precondition: User already logged in.  Steps:  User enters “0” | - | Welcome Page will be shown. | Welcome Page will be shown. | Pass |
| FTC-21 | Verify that the user can access the view contact details feature.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “1”.. | - | Prompt for contact ID will be shown. | Prompt for contact ID will be shown. | Pass |
| FTC-22 | Verify that the user can view the existing contact details.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “1”.. 2. User enters valid ID. | ID = “4” | Contact name, Phone number, Email, and Address for contact with ID 4 will be shown. | Contact name, Phone number, Email, and Address for contact with ID 4 will be shown. | Pass |
| FTC-23 | Verify that the user cannot view the contact details using invalid ID.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “1”. 2. User enters invalid ID. | ID = “420”  ID = “-2” | The message “Invalid ID” will be shown. | The message “Invalid ID” will be shown. | Pass |
| FTC-24 | Verify that the user can access the remove contact feature.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “2”. | - | Prompt for contact ID will be shown. | Prompt for contact ID will be shown. | Pass |
| FTC-25 | Verify that the user can remove the existing contact.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “1”. 2. User enters valid ID. | ID = “4” | Contact with ID 4 will be deleted. | Contact with ID 4 will be deleted. | Pass |
| FTC-26 | Verify that the user cannot remove the contact using invalid ID.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “1”. 2. User enters invalid ID. | ID = “420”  ID = “-2” | The message “Invalid ID” will be shown. | The message “Invalid ID” will be shown. | Pass |
| FTC-27 | Verify that the user can access the edit contact feature.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “3”. | - | Prompt for contact ID will be shown. | Prompt for contact ID will be shown. | Pass |
| FTC-28 | Verify that the user can edit the existing contact.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “3”. 2. User enters valid ID. | ID = “4” | Prompt to choose which field to edit will be shown. | Prompt to choose which field to edit will be shown. | Pass |
| FTC-29 | Verify that the user can edit contact name.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “3”. 2. User enters valid ID. 3. User chooses “1”. 4. User enters new name. | ID = “5”  Field = “1”  Name: “Ma Ma” | Contact name for contact with ID “5” will be changed to “Ma Ma”. | Contact name for contact with ID “5” will be changed to “Ma Ma”. | Pass |
| FTC-30 | Verify that the user can edit Phone number.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “3”. 2. User enters valid ID. 3. User chooses “2”. 4. User enters new phone number. | ID = “5”  Field = “2”  Phone Number:  “092-3456789” | Phone number for contact with ID “5” will be changed to “092-3456789”. | Phone number for contact with ID “5” will be changed to “092-3456789”. | Pass |
| FTC-31 | Verify that the user can edit email.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “3”. 2. User enters valid ID. 3. User chooses “3”. 4. User enters new email. | ID = “5”  Field = “3”  Email: “[mama834@gmail.com](mailto:mama834@gmail.com)” | Email for contact with ID “5” will be changed to “[mama834@gmail.com](mailto:mama834@gmail.com)” | Email for contact with ID “5” will be changed to “[mama834@gmail.com](mailto:mama834@gmail.com)” | Pass |
| FTC-32 | Verify that the user can edit the address.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “3”. 2. User enters valid ID. 3. User chooses “4”. 4. User enters new address. | ID = “5”  Field = “4”  Address: “Bangkok” | Address for contact with ID “5” will be changed to “Bangkok”. | Address for contact with ID “5” will be changed to “Bangkok”. | Pass |
| FTC-33 | Verify that the user cannot edit the contact using invalid ID.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “3”. 2. User enters invalid ID. | ID = “420”  ID = “-2” | The message “Invalid ID” will be shown. | The message “Invalid ID” will be shown. | Pass |
| FTC-34 | Verify that the user can access the sort of feature.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “4”. | - | Prompt to choose sort by name and sort by date will be shown. | Prompt to choose sort by name and sort by date will be shown. | Pass |
| FTC-35 | Verify that the user can sort contact list by name.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “4”. 2. User enters “1”. | - | Contact list will be sorted by name. | Contact list will be sorted by name. | Pass |
| FTC-36 | Verify that the user can sort contact list by date.  Precondition: User is already in view contact list feature or search contact feature.  Steps:   1. User enters “4”. 2. User enters “2”. | - | Contact list will be sorted by date. | Contact list will be sorted by date. | Pass |
| FTC-37 | Verify that the user can exit the program.  Precondition: User is already on the Welcome page.  Steps:   1. User enters “0”. | - | Program will be terminated. | Program will be terminated. | Pass |

**Traceability matrix record:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| UC | URS | SRS | UTC | FTC |
| UC-1 | URS -1 | SRS -1  SRS -2  SRS -3  SRS -4 | UTC -3  UTC –4  UTC-25 | FTC -1  FTC -2  FTC -3  FTC -4  FTC -5 |
| UC-2 | URS -1 | SRS -5  SRS -6  SRS -7 | UTC -1  UTC -2  UTC -3  UTC -4  UTC -5  UTC -6  UTC –7  UTC-29 | FTC -6  FTC -7  FTC -8  FTC -9  FTC -10 |
| UC -3 | URS -3 | SRS -8  SRS -9  SRS -10  SRS -11 | UTC -13  UTC -14  UTC -15  UTC -16  UTC -17  UTC -18  UTC -19  UTC -20  UTC -21  UTC -22  UTC -23  UTC –24  UTC-32  UTC-33  UTC-34 | FTC -11  FTC -12  FTC -13  FTC -14  FTC -15 |
| UC -4 | URS -4 | SRS -12  SRS -13  SRS -14  SRS –15 | UTC -8  UTC –26  UTC-44 | FTC -16 |
| UC -5 | URS -5 | SRS -16  SRS -17  SRS -18  SRS -19  SRS -20 | UTC –8  UTC-42  UTC-43 | FTC -17  FTC -18  FTC -19 |
| UC -6 | URS -6 | SRS -21  SRS -22  SRS -23  SRS -24 |  | FTC -20 |
| UC -7 | URS -7 | SRS -25  SRS -26  SRS -27  SRS -28 | UTC -8  UTC -9  UTC -10  UTC -11  UTC –12  UTC-36 | FTC -21  FTC -22  FTC -23 |
| UC -8 | URS -8 | SRS -29  SRS -30  SRS -31  SRS -32  SRS -33  SRS -34 | UTC-35 | FTC -24  FTC -25  FTC -26 |
| UC -9 | URS -9 | SRS -35  SRS -36  SRS -37  SRS -38  SRS -39  SRS -40 | UTC -8  UTC -9  UTC -10  UTC -11  UTC -12  UTC -13  UTC -14  UTC -15  UTC -16  UTC -18  UTC -19  UTC -20  UTC -21  UTC -22  UTC -23  UTC -24  UTC –25  UTC-37  UTC-38  UTC-39  UTC-40  UTC-41 | FTC -27  FTC -28  FTC -29  FTC -30  FTC -31  FTC -32  FTC -33 |
| UC -10 | URS -10 | SRS -41  SRS -42  SRS -43  SRS -44 | UTC -8  UTC -25  UTC –28  UTC-31  UTC- 45  UTC –46 | FTC -34  FTC -35  FTC -36 |
| UC -11 | URS -11 | SRS -45  SRS- 46 |  | FTC -37 |
|  |  |  |  |  |

**Project wrap-up**

We learned about the importance of documentation in software engineering. It is also important to understand the requirements and clearly specify them. As we have to code as a team, we need to write code that is readable and understandable. We need to design the program to lower level so that both of us know what we need to write.

Successes:

* We implemented the required features with relatively low bugs that we passed most of the test cases at the first time.
* We have completed the project in time.

Areas for improvement:

* We could have improved our source code to be cleaner and more consistent, so it is easier to read and maintain.