|  |  |
| --- | --- |
|  | TutorialOn  Gaba, Vince Harley, D.  He, Zhiyang  Manuel, Lester, N. |

1. Table of Contents

[A Objectives 1](#_Toc526352768)

[B Scope and Features 2](#_Toc526352769)

[B.1 Scope and Feature # 1 2](#_Toc526352770)

[B.2 Scope and Feature # 2 2](#_Toc526352771)

[C Limitations 3](#_Toc526352772)

[C.1 Limitation # 1 3](#_Toc526352773)

[C.2 Limitation # 2 3](#_Toc526352774)

[D System Requirements 4](#_Toc526352775)

[D.1 Hardware Requirements 4](#_Toc526352776)

[D.1.1 Processor 4](#_Toc526352777)

[D.1.2 Memory 4](#_Toc526352778)

[D.1.3 Storage 4](#_Toc526352779)

[D.2 Software Requirements 4](#_Toc526352780)

[D.2.1 Operating System 4](#_Toc526352781)

[D.2.2 Software Requirements 4](#_Toc526352782)

[E Class Diagram 5](#_Toc526352783)

[E.1 {Name} Class Diagram 5](#_Toc526352784)

[F Class Definition 6](#_Toc526352785)

[F.1 {Name} Class Definition 6](#_Toc526352786)

[G User Manual 8](#_Toc526352787)

[G.1 Peer Adviser Login 8](#_Toc526352788)

[G.2 Advisee Login 9](#_Toc526352789)

[G.3 Administrator Login 11](#_Toc526352790)

[G.4 Peer Adviser Signup 12](#_Toc526352791)

[G.5 Advisee Signup 13](#_Toc526352792)

[G.6 Administrator Signup 14](#_Toc526352793)

[H Unit Test 15](#_Toc526352794)

[H.1 Advisee Login Without Logged-in Peer Adviser 15](#_Toc526352795)

[H.2 Wrong Peer Adviser Login 15](#_Toc526352796)

[H.3 Correct Peer Adviser Login 15](#_Toc526352797)

[H.4 Wrong Advisee Login 15](#_Toc526352798)

[I References 16](#_Toc526352799)

[I.1 Reference # 1 16](#_Toc526352800)

[I.2 Reference # 2 16](#_Toc526352801)

List of Figures

[Figure F.1.a: {Name} Class Diagram 5](#_Toc526352802)

[Figure G.1.a: Peer Adviser Login 8](#_Toc526352803)

[Figure G.2.a: Advisee Login 9](#_Toc526352804)

[Figure G.3.a: Administrator Login 11](#_Toc526352805)

[Figure G.4.a: Peer Adviser Signup 12](#_Toc526352806)

[Figure G.5.a: Advisee Registration 13](#_Toc526352807)

[Figure G.6.a: Administrator Registration 14](#_Toc526352808)

[Figure H.a: TutorialOn in terminal 15](#_Toc526352809)

[Figure H.1.a: Advisee Login Without Logged-in Peer Adviser 15](#_Toc526352810)

[Figure H.2.a: Wrong Peer Adviser Login 15](#_Toc526352811)

[Figure H.3.a: Correct Peer Adviser Login 15](#_Toc526352812)

[Figure H.4.a: Wrong Advisee Login 15](#_Toc526352813)

# Objectives

{Objective Paragraph 1}

{Objective Paragraph 2}

{Objective Paragraph 3}

# Scope and Features

## Scope and Feature # 1

Scope and Feature # 1 details…

## Scope and Feature # 2

Scope and Feature # 2 details…

# Limitations

## Limitation # 1

Limitation to “Scope and Feature # 1” detail # 1

Limitation to “Scope and Feature # 1” detail # 2

## Limitation # 2

Limitation to “Scope and Feature # 2” detail # 1

Limitation to “Scope and Feature # 2” detail # 2

# System Requirements

## Hardware Requirements

### Processor

Processor Core Count : {Value}

### Memory

Memory Capacity : {Value} {Unit (e.g. GB)}

### Storage

Disk Size : {Value} {Unit (e.g. GB)}

## Software Requirements

### Operating System

Operating System Name : {Value}

Architecture Type : {Value}

Edition : {Value}

### Software Requirements

Software Application # 1 : {Value}

Software Application # 2 : {Value}

Software Application # 3 : {Value}

# Class Diagram

## {Name} Class Diagram

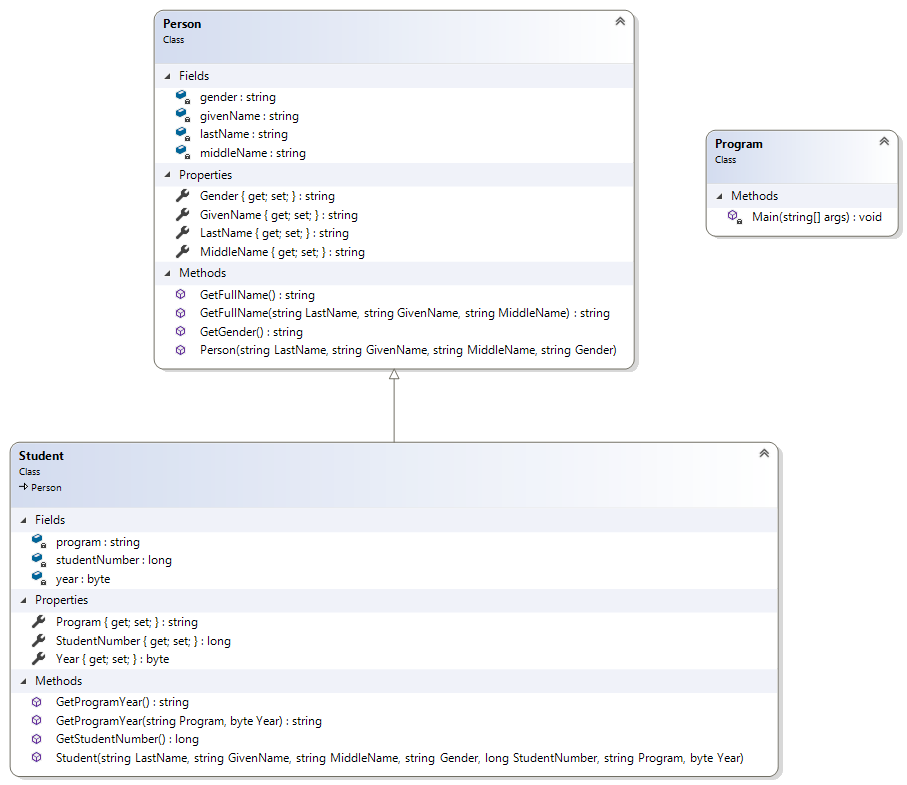


Figure F.1.a: {Name} Class Diagram

# Class Definition

## {Name} Class Definition

|  |  |
| --- | --- |
| **Name:** | Person |
| **Type:** | Class |
| **Description:** | Used to store and get person information. |
| **Fields:** | personID : int |
| name : string |
| **Properties:** | PersonID { get; set; } : int |
| Name { get; set; } : string |
| **Methods:** | Person(int PersonID, string Name) |
| GetID() : int |
| GetName(string LastName, string GivenName) : string |

|  |  |
| --- | --- |
| **Name:** | Person |
| **Member Name:** | Person |
| **Type:** | Constructor |
| **Description:** | Used as a constructor for the “Person” class. |
| **Parameters:** | int PersonID |
| string Name |
| **Return Value:** | None |
| **Return Type:** | None |

|  |  |
| --- | --- |
| **Name:** | Person |
| **Member Name:** | GetID |
| **Type:** | Method |
| **Description:** | Used to get ID of person. |
| **Parameters:** | None |
| **Return Value:** | ID |
| **Return Type:** | int |

|  |  |
| --- | --- |
| **Name:** | Person |
| **Member Name:** | GetFullName |
| **Type:** | Method |
| **Description:** | Used to get full name of person based from last and given name. |
| **Parameters:** | string LastName |
| string GivenName |
| **Return Value:** | Name |
| **Return Type:** | string |

|  |  |
| --- | --- |
| **Name:** | Student |
| **Type:** | Class |
| **Description:** | Used to store and get student information. |
| **Fields:** | program : string |
| studentNumber : long |
| year : byte |
| **Properties:** | Program { get; set; } : string |
| StudentNumber { get; set; } : long |
| Year { get; set; } : byte |
| **Methods:** | GetStudentNumber() |
| GetProgramYear(string Program, byte Year) : string |

|  |  |
| --- | --- |
| **Name:** | Student |
| **Member Name:** | Student |
| **Type:** | Constructor |
| **Description:** | Used as a constructor for the “Student” class. |
| **Parameters:** | string Program |
| long StudentNumber |
| byte Year |
| **Return Value:** | None |
| **Return Type:** | None |

|  |  |
| --- | --- |
| **Name:** | Student |
| **Member Name:** | GetStudentNumber |
| **Type:** | Method |
| **Description:** | Used to get student number of student. |
| **Parameters:** | None |
| **Return Value:** | StudentNumber |
| **Return Type:** | string |

|  |  |
| --- | --- |
| **Name:** | Student |
| **Member Name:** | GetProgramYear |
| **Type:** | Method |
| **Description:** | Used to get program and year level of student. |
| **Parameters:** | string Program |
| byte Year |
| **Return Value:** | ProgramYear |
| **Return Type:** | string |

# User Manual

## Peer Adviser Login

Enter student number (10-digit).

Click “Login”.

Enter password.

Click “Login”.

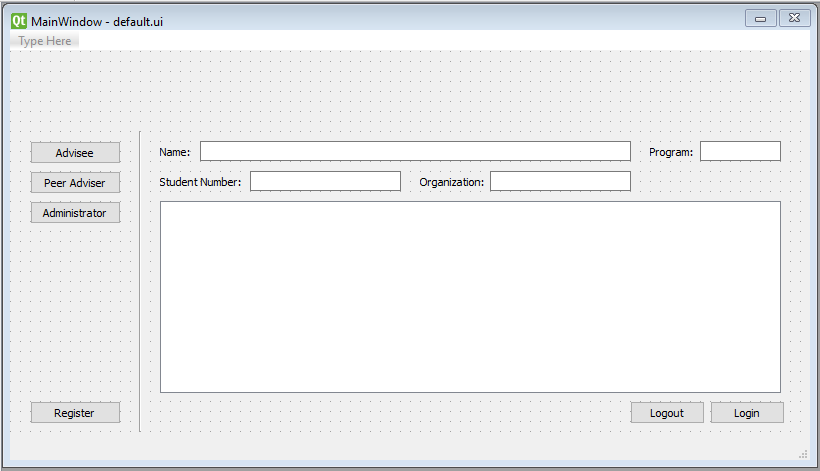


Figure G.1.a: Peer Adviser Login

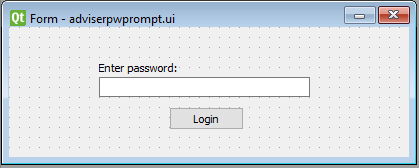


Figure G.1.b: Peer Adviser Login Password Prompt

## Advisee Login

Enter student number (10-digit).

Choose a field (Mathematics, Chemistry, Physics, or Professional Course)

Select a course code from the drop-down list.

If “Others” is selected, enter the course code.

Click the “…” button to choose a peer adviser from the logged in list.

Select a peer adviser.

Click “Select”.

Click “Login”.

Note: No advisee can login if there is no logged-in peer adviser.

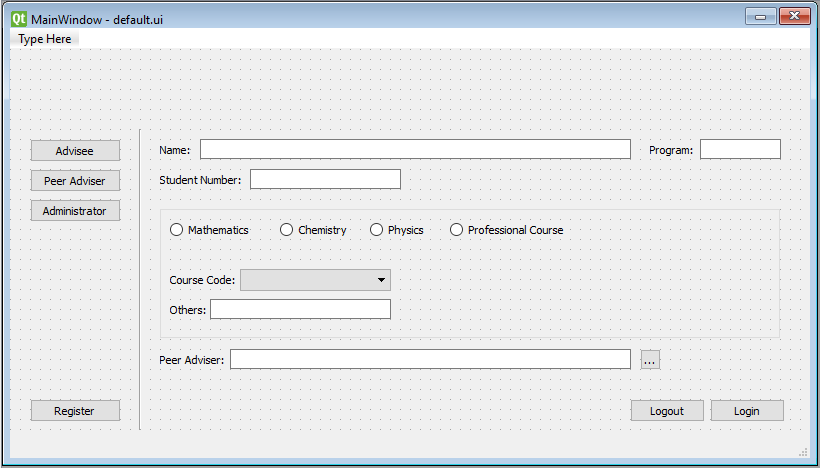


Figure G.2.a: Advisee Login

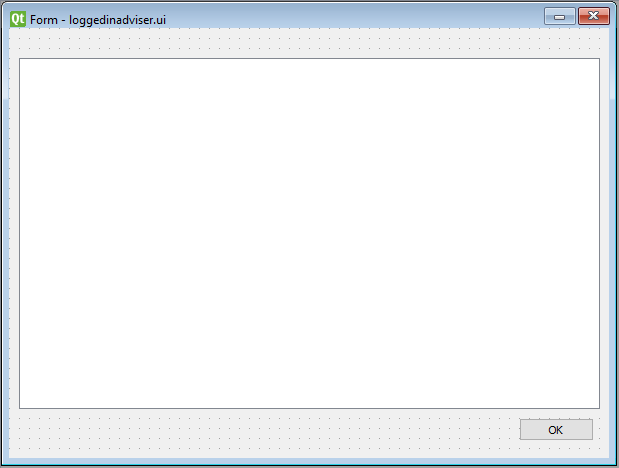


Figure G.2.b: Logged-In Peer Advisers

## Administrator Login

Enter username.

Enter password.

Click “Login”.

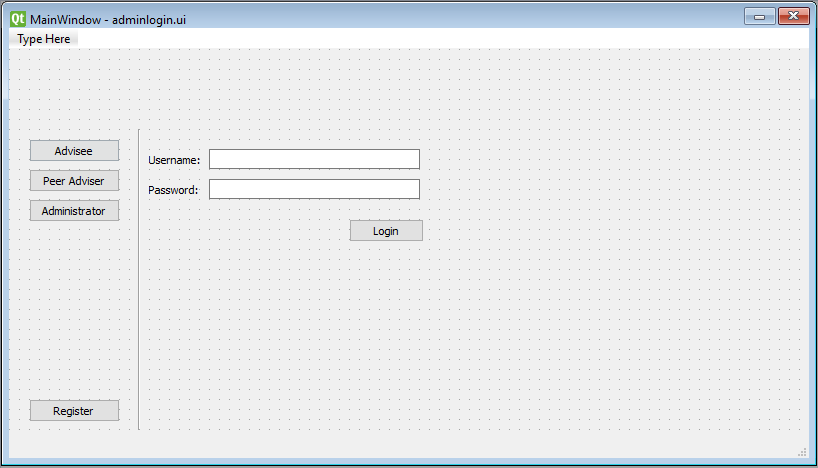


Figure G.3.a: Administrator Login

## Peer Adviser Signup

Fill-up fields.

Click “Register”.

Note: Only administrators are capable of creating peer adviser’s account.

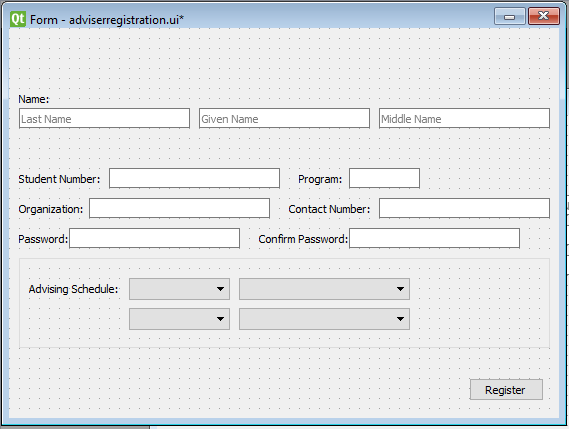


Figure G.4.a: Peer Adviser Signup

## Advisee Signup

Fill-up fields.

Click “Register”.

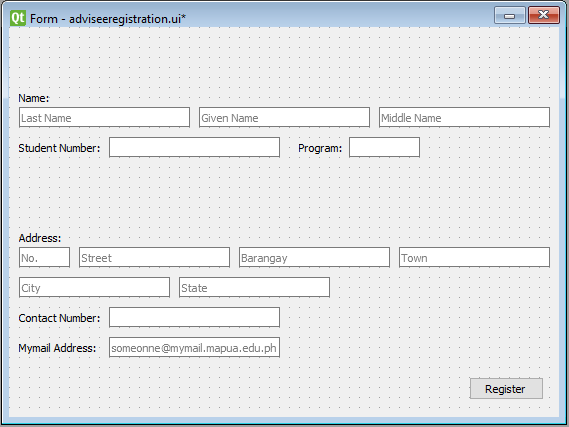


Figure G.5.a: Advisee Registration

## Administrator Signup

Fill-up fields.

Click “Register”.

Note: Only administrators can create administrator account.

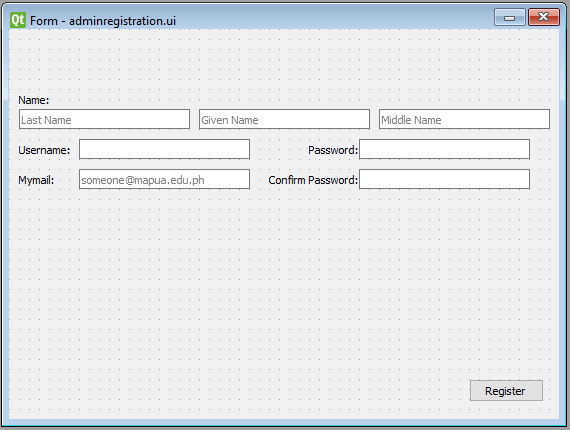


Figure G.6.a: Administrator Registration

# Unit Test

The proponents had test cases in logging in users using the terminal.

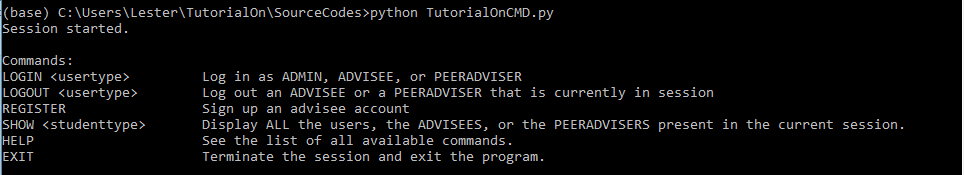


Figure H.a: TutorialOn in terminal

## Advisee Login Without Logged-in Peer Adviser

No advisee can login if no peer adviser logged-in.



Figure H.1.a: Advisee Login Without Logged-in Peer Adviser

## Wrong Peer Adviser Login

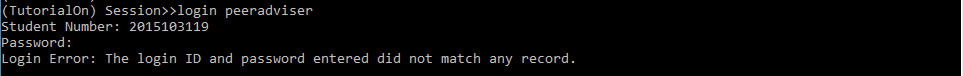


Figure H.2.a: Wrong Peer Adviser Login

## Correct Peer Adviser Login

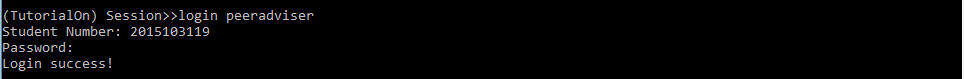


Figure H.3.a: Correct Peer Adviser Login

## Wrong Advisee Login

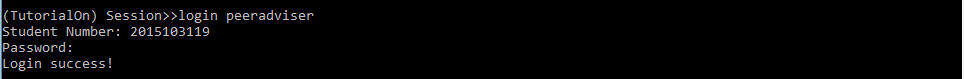


Figure H.4.a: Wrong Advisee Login

# References

## Reference # 1

Reference Title : {Value}

Reference Author : {Value}

Reference Page Source : {Value}

## Reference # 2

Reference Title : {Value}

Reference Author : {Value}

Reference Page Source : {Value}

