Lost and Found App: Breaking the Wall of Lost Items on Campus

Name: Dang Quynh Chi

Student ID: 1038057045

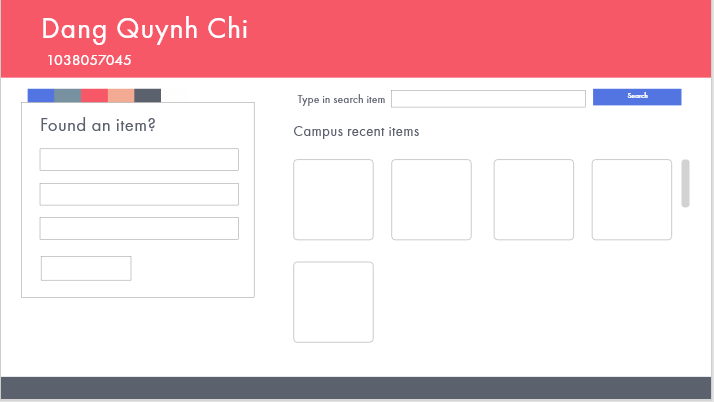
# Summary of Program

Every day at your school, valuable items are misplaced - phones, wallets, laptops and more. Without an effective system to record and reunite lost items with their owners, these items often go missing forever.

But now there is a solution – Lost and Found, a centralized lost and found database designed specifically for schools. It allows students and staff to easily log any found items, search for missing belongings, and get credit for returning lost property.

By cutting out disorganized paper records and manual searches, the program brings order and efficiency to the chaotic process of reuniting owners with their lost items. No more rummaging through lost boxes hoping for the best.

Sensitive item details are stored using encryption and salting, so you can feel safe logging expensive devices without fear of theft. User logins are also protected by passwords or accessed by google sign in ( fpt mail )



# Required Roles

Describe each of the classes, interfaces, and any enumerations you will create. Use a different table to describe each role you will have, using the following table templates.

Class: Item:Identifiable object

|  |  |  |
| --- | --- | --- |
| Responsibility | Notes | Type details |
| location | The location of the item lost | Location.Name string |
| Item | (time, name, User founder, status) | Constructor |
| time | The time it was lost | String |
| type | Type of the item | String |
| finder | The id of the finder | String |
| owner | The id of the owner | String |
| status | Display the status of the item | String |
| Status property | Get the status ( by user) | string Status |
| AStatus property | Get and set the status(admin) | string AStatus |

Class: Inventory

|  |  |  |
| --- | --- | --- |
| Responsibility | Notes | Type details |
| Inventory | (id)  List<Item> = new List<Item> |  |
| items | List of type Item | List<Item> |

Class: GameObjects

|  |  |  |
| --- | --- | --- |
| Responsibility | Notes | Type details |
| Identifiable | (List string ) adds ids | constructor |
| ident |  | List<string> |
| AreYou | If contains id, return | bool |

Class: User

|  |  |  |
| --- | --- | --- |
| Responsibility | Notes | Type details |
| name | Name of the user |  |
| id | Student Id |  |
| name | Return name |  |

Class: Location

|  |  |  |
| --- | --- | --- |
| Responsibility | Notes | Notes |
| name | Name of the locaction | string |
| floor | The floor of the location | string |
| description | Location description | string |
| inventory | List of the items in the location | List<Item> |

Abstract Class: Command

|  |  |  |
| --- | --- | --- |
| Responsibility | Type details | Notes |
| Command | (string[] id) :base(id) | constructor |
| Execute | Execute(ILocationList location, string[] text) |  |

CommandFactory

|  |  |  |
| --- | --- | --- |
| Responsibility | Type details | Notes |
| Execute | MakeCommand(string[] text) | method |

Class: LookCommand :Command

|  |  |  |
| --- | --- | --- |
| Responsibility | Notes | Type details |
|  |  |  |
| Execute | Look for item | Method string |
| Look at | If the inventory exists, returns string | Method string |
| Fetch | Return IhaveInventory | Method IhaveInventory |

Class: Post

|  |  |  |
| --- | --- | --- |
| Responsibility | Notes | Type details |
| Execute | Upload item, if successful returns a string | Method string |
| Filter | If the inventory exists, cancels the operation and returns a string | Method string |

Class: TakeCommand:Command

|  |  |  |
| --- | --- | --- |
| Responsibility | Notes | Type details |
| Execute | Move item, if successful returns string | Method string |
| Look at | If the inventory exists, returns the inventory | Method Inventiory |
| Fetch | Return IhaveInventory | Method IhaveInventory |

Interface: ILocationList

|  |  |  |
| --- | --- | --- |
| Responsibility | Type details | Notes |
| Command | (string[] id) :base(id) | constructor |
| Execute | Execute(ILocationList location, string[] text) |  |

LocationList

|  |  |  |
| --- | --- | --- |
| Responsibility | Type details | Notes |
| LocationList | () | constructor |
| AddLocation | (Location loc) | Add location into list |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Class Diagram

Provide an initial design for your program in the form of a class diagram.

# Sequence Diagram

Provide a sequence diagram showing how your proposed classes will interact to achieve a specific piece of functionality in your program.