**AIMS: AN INTERNET MEDIA STORE**

TEST PLAN FOR INFORMATION DELIVERY OF CUSTOMER

Subject: ITSS Software Development

Group Number 9

| **Name** | **Student ID** | **Role** |
| --- | --- | --- |
| Lê Xuân Hiếu | 20215201 | Team Leader |
| Nguyễn Hà Hiếu | 20215202 | Member |
| Nguyễn Mạnh Hiếu | 20215203 | Member |
| Nguyễn Văn Hiếu | 20215204 | Member |
| Phạm Trung Hiếu | 20215205 | Member |

Hanoi, June 2024

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| 1 | Instructions | |
| 2 | Overall Description | |
| 3 | Testing Approach | |
| 4 | Unit Testing Details | |
|  |  | |

1. **Introduction**
   1. **Object**

Validation and Verification: It aims to validate the functionality, correctness of the DeliveryInfo class in managing shipping information within a software system.

It ensures that the class handles exceptions (e.g., EmptyFieldsException, InvalidEmailException, InvalidDateException….) appropriately when invalid data is encountered.

* 1. **Scope**
* Check the EmptyFieldsException exception

Input: One or more information fields are left blank.

Expectation: Throws an EmptyFieldsException with the message "All fields must be filled".

* Check for InvalidNameException

Input: Name field whose length exceeds 50 characters.

Expectation: Throws an InvalidNameException with the message "Your name is invalid".

* Check the InvalidEmailException exception

Input: Invalid email (does not contain "@mail.com" or "@gmail.com").

Expectation: Throws an InvalidEmailException with the message "Your email is invalid".

* Check for InvalidNumberException exception

Input: Invalid phone number (not 10 digits or not starting with 0).

Expectation: Throws an InvalidNumberException with the message "Your phone number is invalid".

* Check for InvalidDateException exception

Input: Invalid express delivery date (before current date).

Expectation: Throws an InvalidDateException with the message "The delivery date must be after today".

1. **Overall description**
   1. **Genneral overview**

* Usecase

Actors:

* Customer: Provides delivery information such as name, email, phone number, address, and instructions.
* System: Validates the provided information and handles exceptions.

Use Cases:

Submit Delivery Info: Customer submits delivery information.

Validate Delivery Info: System validates the submitted delivery information.

Handle Invalid Data: System handles any invalid data by throwing appropriate exceptions.

* Activity
* **Input Delivery Info**: Customer inputs their delivery information.
* **Check Empty Fields**: System checks if any fields are empty.
* **Validate Name**: System validates the name.
* **Validate Email**: System validates the email.
* **Validate Phone**: System validates the phone number.
* **Check Rush Order**: System checks if it is a rush order.
* **Validate Rush Time**: If it's a rush order, the system validates the rush time.
* **Complete Validation**: If all validations pass, the system accepts the delivery information.
* **Handle Exceptions**: If any validation fails, the system throws the appropriate exception

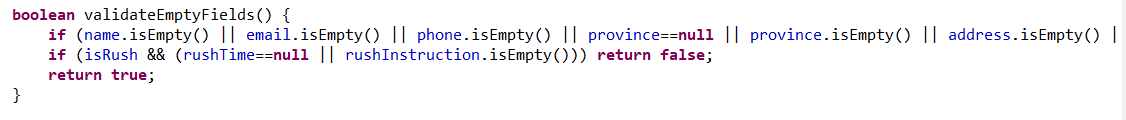
1. **Testing approach**

Replace the test cases according to the initial conditions set and check

Check for common exceptions (empty, incorrect format, ....)

In this case we test the ValidateInfo class by trying some common cases and testing with junit5 library

1. **Unit test detail**
   1. **testValidateEmptyFields()**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Test data | Value | Expected Results | Actual Results |
| 1 | Input information with “email” is empty | "", "hieu", "0123456789", "Hà Nội", "123 ",  "Leave at door", **true**, LocalDate.now(), "instructions" | System return  EmptyFieldsException("All fields must be filled") | * Function return “false” * System return EmptyFieldsException("All fields must be filled") |
| 2 | Input information with “name” is empty | "hi@mail.com", "", "0123456789", "Hà Nội", "123 ",  "Leave at door", **true**, LocalDate.now(), "instructions" | System return  EmptyFieldsException("All fields must be filled") | * Function return “false” * System return   EmptyFieldsException("All fields must be filled") |
| 3 | Input information enough all fields | "test@gmail.com", "hieu", "0123456789", "Hà Nội",  "123", "Leave at door", **true**, LocalDate.now(),  "instructions" | Continue checking process with other fields | * Function return “true” * Continue checking process with other fields |

* 1. **testValidateName()**

A computer code with text

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Test data | Value | Expected Results | Actual Results |
| 1 | Input name with length >50 character | "ten nay dai vai cádjhfjshfjsdhjfhsjdfhjsdhfjd dfhsdjfsfhjsdhfjkshfhsjkdhfks" | System return  InvalidNameException("Your name is invalid"); | * Function return “false” * System return InvalidNameException("Your name is invalid"); |
| 2 | Input name with length <50 character | “hieu” | Continue checking process with other fields | * Function return “true”   Continue checking process with other fields |

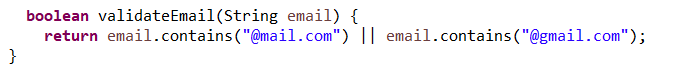
* 1. **testValidatePhone()**

A screen shot of a computer code

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Test data | Value | Expected Results | Actual Results |
| 1 | String number length =10 includes all digits | “0123456789” | Continue checking process with other fields | * Function return “true” * Continue checking process with other fields |
| 2 | String number length != 10 | “123456” | System return  InvalidNumberException("Your phone number is invalid") | * Function return “false” * System return   InvalidNumberException("Your phone number is invalid") |
| 3 | numeric string includes non-alphanumeric characters | "phone12345" | System return  InvalidNumberException("Your phone number is invalid") | * Function return “false” * System return   InvalidNumberException("Your phone number is invalid") |

* 1. **testValidateEmail()**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Test data | Value | Expected Results | Actual Results |
| 1 | String email contains  ”@gmail.com” | test@gmail.com | Continue checking process with other fields | * Function return “true” * Continue checking process with other fields |
| 2 | String email contains  ”@mail.com | “tetssthd@mail.com” | Continue checking process with other fields | * Function return “true” * Continue checking process with other fields |
| 3 | String email do not contains “@gmail.com”  ”@mail.com” | test@udsfsdfsdfsn.com | System return  InvalidEmailException("Your email is invalid") | * Function return “false” * System return   InvalidEmailException("Your email is invalid") |

* 1. **testValidateTime()**

A black text on a white background

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Test data | Value | Expected Results | Actual Results |
| 1 | Date input after currrent time | LocalDate.now() | Continue checking process with other fields | * Function return “true” * Continue checking process with other fields |
| 2 | Date input before current time | LocalDate.now().minusDays(1)) | InvalidDateException("The delivery date must be after today") | * Function return “false” * InvalidDateException("The delivery date must be after today") |

* 1. **testValidateRushAddress()**

A computer code on a white background

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Test data | Value | Expected Results | Actual Results |
| 1 | Province input is “hà nội” or “hồ chí minh” | “HÀ NỘI” | Continue checking process with other fields | * Function return “true” * Continue checking process with other fields |
| 2 | Province input is not “hà nội” or “hồ chí minh”current time | “đà nẵng” | validateRushAddress return “false” | validateRushAddress return “false” |