

**FxChange**

**Old Stuff Exchange for FPTU Student**

– Ho Chi Minh, May 2023 –

**Table of Contents**

[I. Overview 4](#_Toc96516286)

[1. Code Packages 4](#_Toc96516287)

[2. Database Design 4](#_Toc96516288)

[a. Database Schema 4](#_Toc96516289)

[b. Table Description 4](#_Toc96516290)

[II. Code Designs 5](#_Toc96516291)

[1. <Feature/Function Name1> 5](#_Toc96516292)

[a. Class Diagram 5](#_Toc96516293)

[b. Class Specifications 5](#_Toc96516294)

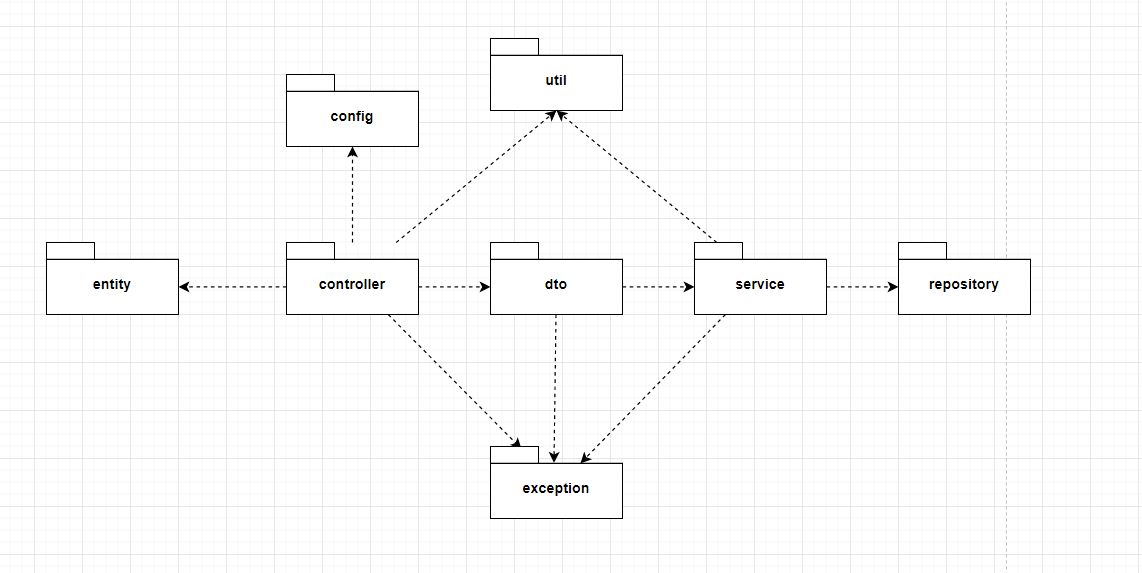
[c. Sequence Diagram(s) 5](#_Toc96516295)

[d. Database queries 6](#_Toc96516296)

[2. <Feature/Function Name2> 6](#_Toc96516297)

# I. Overview

## 1. Code Packages



***Package descriptions***

|  |  |  |
| --- | --- | --- |
| **No** | **Package** | **Description** |
| 01 | config | - This package contains classes related to configuration of the Spring framework and other third-party libraries.  - ClassName + “Config” (eg: AppConfig.java) |
| 02 | controller | contains classes that handle and route incoming HTTP requests  - ClassName + “Controller” (eg: DashboardController.java) |
| 03 | entities | - contains entity classes which represent the data model of the application.  - ClassName + “Entity” (eg: UserEntity.java) |
| 04 | service | - contains service classes which implement the business logic of the application.  - ClassName + “Service” (eg: UserService.java) |
| 05 | dto | - contains data transfer objects which are used to transfer data between layers, contain the data that is necessary for the operation being performed.  - ClassName + “DTO” (ex: UserDTO.java) |
| 06 | exception | - contains exception classes which are thrown when errors occur during request processing.  - ClassName + “Exception” (eg: CreateNewProductException.java) |
| 07 | repository | - contains repository interfaces which handle data access and storage directly to database.  - ClassName + “Repository” (eg: UserRepository.java) |

## 2. Database Design

### a. Database Schema

### 

### 

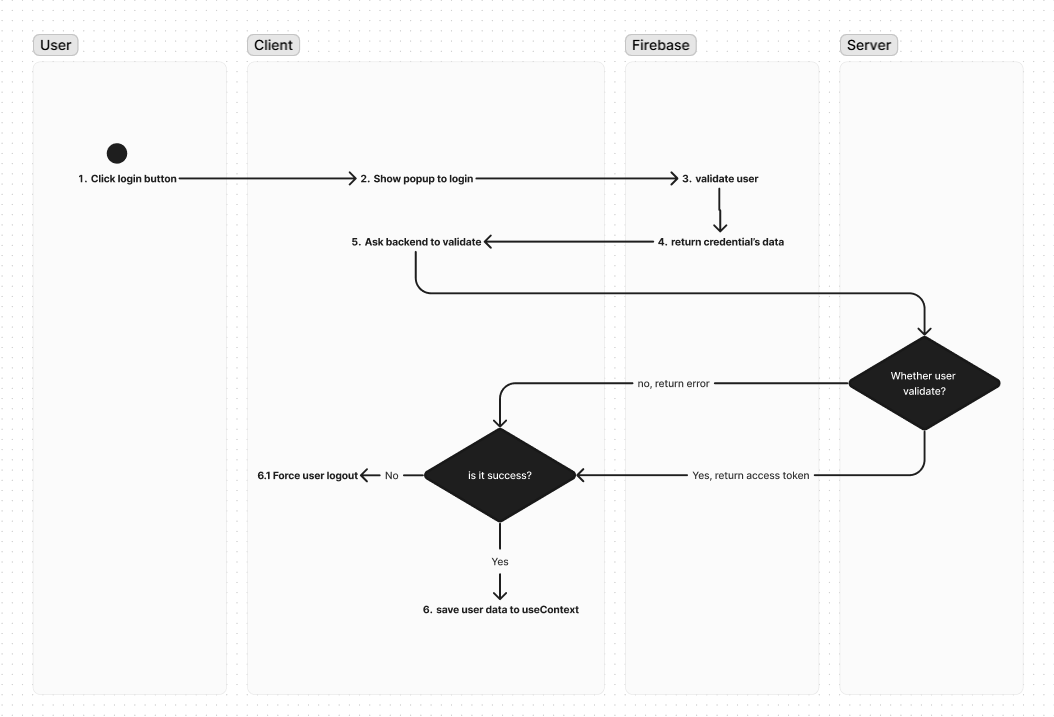
### b. Table Description

|  |  |  |
| --- | --- | --- |
| **No** | **Table** | **Description** |
| 01 | users | - contains user’s information (name, address, phone,…)  - primary key: \_id |
| 02 | stuffs | - contains stuff’s information  - primary key: \_id  - foreign key: category\_id, type\_id, author\_id |
| 03 | categories | - stuff’s category  - primary key: \_id |
| 04 | types | - stuff’s type (exchange, market)  - primary key: \_id |
| 05 | tags | - stuff’s tag (size, color,…)  - primary key: \_id |
| 06 | transactions | - contains stuff’s transaction, final result when exchange, trade, auction.  - primary key: \_id  - foreign key: customer\_id, stuff\_id |
| 07 | roles | - system’s role, aims to authorization  - primary key: \_id |
| 08 | permissions | - list permission corresponding for each role  - primary key: \_id  - foreign key: resource\_url\_id |
| 09 | resource\_url | - all routes of application  - primary key: \_id |
| 10 | notifications | - contains all the notification of specific action corresponding for each user on system.  - primary key: \_id  - foreign keys: user\_id, target\_id |
| 11 | comments | - contains all comments on system.  - primary key: \_id  - foreign keys: user\_id, stuff\_id |
| 12 | point\_history | - store history changing points of user  - primary key: \_id  - foreign keys: user\_id |

# II. Code Designs

## Sequence diagram

1. Login



1. Exchange stuff

A picture containing text, diagram, screenshot, parallel

Description automatically generated

1. Buy stuff

A diagram of a flowchart

Description automatically generated with low confidence

1. Private chat

A picture containing text, diagram, design

Description automatically generated