CS102	**Fall 2019/20**	Project Group	G1
Assistant:	**Mustafa Can Çavdar**		

Criteria	TA/Grader	Instructor
Presentation		
Overall		

~ ExToNext ~

javangels

İlke KAŞ

Zeynep Büşra ZİYAGİL

Bilgehan AKCAN

Selahattin Cem ÖZTÜRK

Lara FENERCİOĞLU

Umut Ege ÖZDEMİR

Detailed Design Report

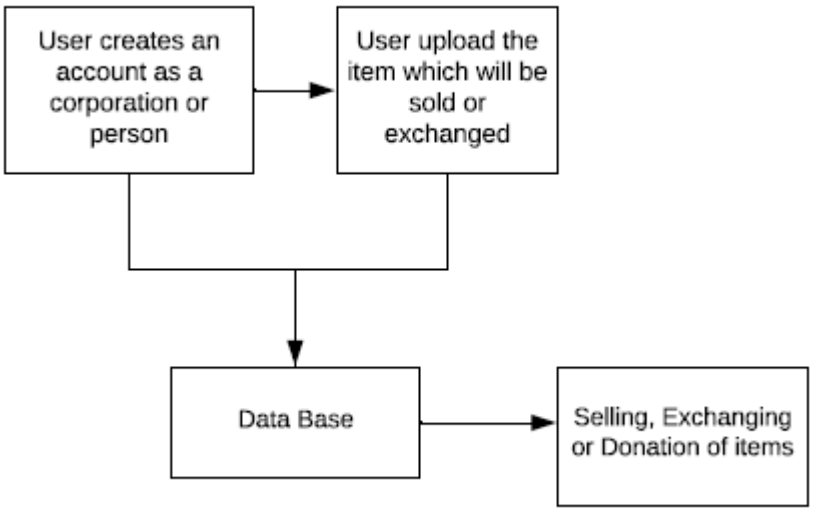
(version 2.0)

25 December 2019

1. Introduction

People want to close out their unnecessary items since they do not use them anymore. Also, people sometimes buy items they will not use in the future. Because of that there is a huge consumption problem. For that reason, to avoid unnecessary consumerism, there is a need for a social network program that will be a desktop application. This platform will enable people to exchange and/or sell their items. Moreover, this platform will also allow people to donate their items to needy people or foundations. Items will be added to the program by the users. According to the seller's wish, items can be exchanged with another item, can be sold or donated. Eventually, the platform will be available for all people who want to share their items. Hopefully, our social network platform will help a lot of people to find what they're looking for. On the following pages, there will be a diagram showing the basic organization of the application with a brief explanation and a UML class diagram about core model of the application.

2. System Overview



ExToNext is a desktop application. Our program mostly uses javax.swing and java.awt packages in order to provide graphical user interface. Also, databases are used to ensure the persistent data.

Our program enables people to buy, sell or donate items through some buttons that call some specific actions. Above, there is a diagram which shows the basic organization of the project.

3. Core Design Details

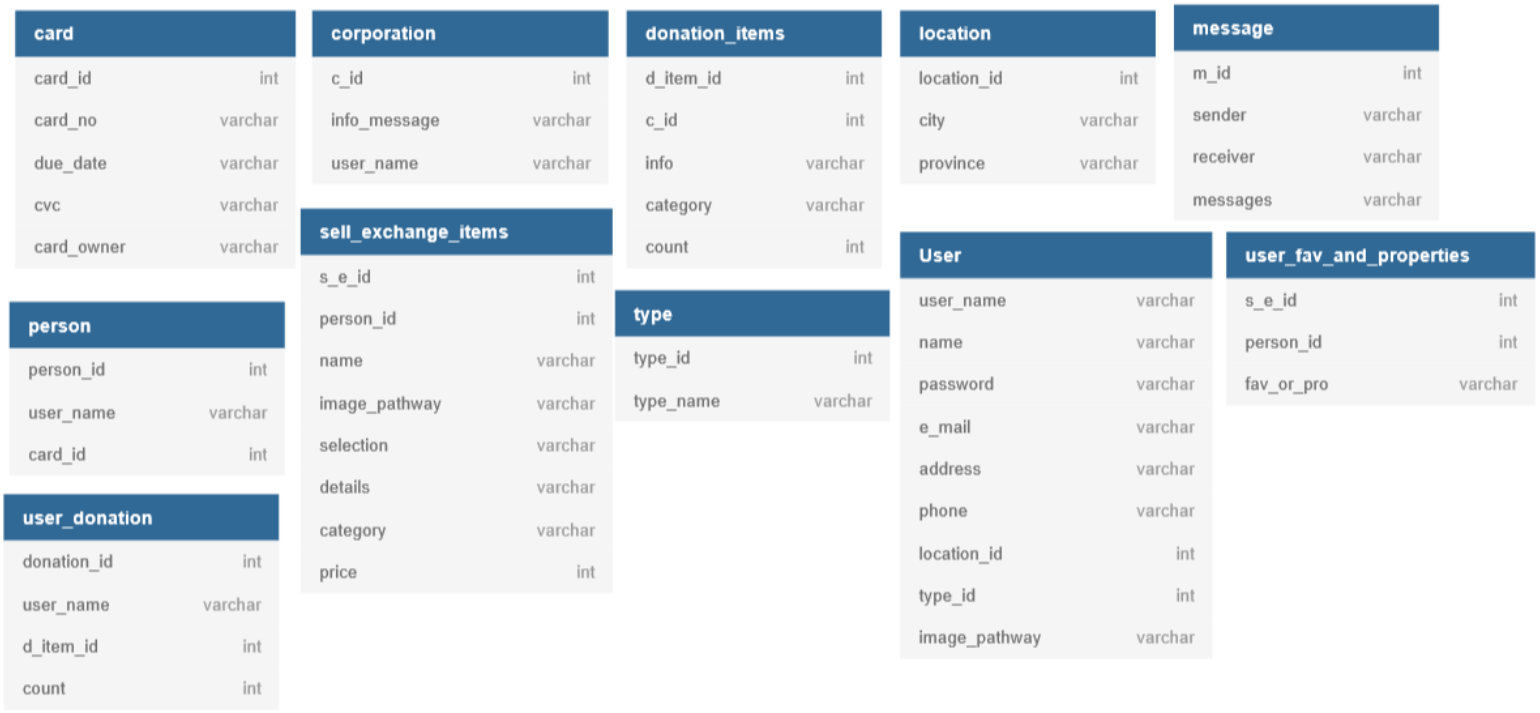
This is our database connector class. We use those methods in GUI.

DBConnector

-connection: Connection

+DBConnector()
+isAlreadySignUp(String userName): boolean
+personOrCorporation(String userName): int
-createCard(String cardNo, String dueDate, int cvc, String cardOwner): void
-addOneToPerson(String userName, int cardId): void
-createLocation(String city, String province): void
-addToUser(String userName, String name, String password, String email, String address, String phone, int locationId, int typeId, String image): void
+public boolean createCorporationUser(String userName,String city, String province,String name, String password, String email, String address, String phone, String image, String infoMessage): boolean
+boolean createPersonUser(String cardNo, String dueDate, int cvc, String cardOwner,String userName, String city, String province,String name, String password, String email, String address, String phone, String image): boolean
+userImage(String userName): String
+itemImage(int itemId): String
+getCardId(String cardNo, String dueDate, int cvc, String cardOwner): int
+getLocationId(String city, String province): int
-addOneToCorporation(String infoMessage, String userName): void
-addWishList2(int c_id, String info, String category,int count): void
+addWishList1(String userName, String info, String category, int count): void
+listDonationSearchBar(String inputcat,String input): ArrayList<String>
+listDonationOnlySearchBar(String input): ArrayList<String>
+donationItemDetail(int donationItemId): ArrayList<String>
+donate(int donationItemId,String userName, int count): void
-updateCount(int donationItemId): void
-addToUserDonations(String userName, int d_item_id, int count): void
+afterDonation(int donationItemId): ArrayList<String>
+seeCorporationInfo(String userName): ArrayList<String>
+donationsIGet1(String userName): ArrayList<String>
-donationsIGet2(int corpId): ArrayList<String>
+myDonations(String userName): ArrayList<String>
+getReceivers(String sender): ArrayList<String>
+showMessages(String sender, String receiver): ArrayList<String>
+sendMessage(String sender, String receiver, String messages): void
+seePersonInfo(String userName): ArrayList<String>
+myUploads1(String userName): ArrayList<String>
-myUploads2(int personId): ArrayList<String>
+myProperties(String userName): ArrayList<String>
+myFavorites(String userName): ArrayList<String>
+sellListBuySearchCat(String input, String category): ArrayList<String>
+sellListBuySearch(String input): ArrayList<String>
+exchangeListBuySearchCat(String input, String category): ArrayList<String>
+exchangeListBuySearch(String input): ArrayList<String>
-itemInfo2(int seId): ArrayList<String>
+itemInfo1(String image): ArrayList<String>
-personInfo2(int seId): ArrayList<String>
+personInfo1(String image): ArrayList<String>
+uploadSellItem1(String userName,String name, String image, String details, String category, int price): void
-uploadSellItem2(int personId, String name, String image, String details, String category, int price): void
+uploadExchangeItem1(String userName,String name, String image, String details, String category): void
-uploadExchangeItem2(int personId, String name, String image, String details, String category): void
+getPersonId(String userName): int
+getseId(String image): int
+addToFavorites(String userName, String image): void
-turnFavoriteToProperty(int personId, int seId): void
-addToProperty(int personId, int seId): void
+buyAndExchange(String userName, String image): void
-removeFromItems(int seId): void
+logIn(String userName, String passWord): boolean

We don’t have a model class in Java but we treated our database defaultdb as a model. This is our tables uml on defaultdb database:



4. Task Assignment

4.1 İlke Kaş

- Will work on the UI Design
- Log In and Sign Up pages (UI Design)
- Corporation Account Main Page (UI Design)
- Personal Account Main Page (UI Design)
- Creating Wish List Page (UI Design)
- Will create the Person class
- Will create the Corporation class
- Will work on DataBaseConnector

4.2 Zeynep Büşra Ziyagil

- Will work on UI Design
- Chats Page (UI Design)
- Resulting Items Page (UI Design)
- Detailed Page for Item (UI Design)
- Upload Item Page (UI Design)

- Will create the DonationItem class
- Will create the SellExchangeItem class

4.3 Bilgehan Akcan

- Will create the test classes
- Will create the Card class
- Will create the Location class
- Will work on the UI Design
- Resulting Donations Page (UI Design)
- Donated Item Page (UI Design)

4.4 Selahattin Cem Öztürk

- Will create the DonationItem class
- Will create the SellExchangeItem class
- Will work on DataBaseConnector

4.5 Lara Fenercioğlu

- Will work on the User class
- Will create the Person class
- Will create the Corporation class
- Will work on the UI Design
- Will create the test classes
- Will work on DataBaseConnector

4.6 Umut Ege Özdemir

- Will work on the UI Design
- Searching Page (UI Design)
- Resulting Donations Page (UI Design)
- Wish List Page (UI Design)
- Donated Item Page (UI Design)
- Evaluation Page (UI Design)