MULTI TRADE

Aric Rachmany - 205357619

Tal kabaso – 204613863

MultiTrade Web site

1. Introduction

This sdd file will describe our project technologies and practices that we chose to use

In our project, we combine several techniques and programming languages together to create a comfortable and efficient user environment as well as an efficient, convenient and operational server side.

The main idea of the project is to create a comfortable and user-friendly environment that will allow it to exchange objects with different people by an algorithm that ensures that every person who participates only profit from it.

In the documents describing our project, we chose to use "tender" to describe the group and the process of replacement.

1. SYSTEM OVERVIEW

See **design and functionality explanation.pdf** In order to expect the functionality and website design

1. SYSTEM ARCHITECTURE

In our project we chose to use python as our main programming language in the project because it is very convenient and flexible and allows to perform operations relatively easily compared to other programming languages.

In the server side we use "flask" framework that allows as to connect between the html files and the client side into one working system. In order to use the functionality on the web pages.

we use JS to improve the user experience and the convenience and clarity of the system capabilities.

Our data about the users and another meta data will be saved with the help of the familiar database "MongoDB".

Spectator

Participant in a tender

Join

Finding a tender on the site by searching for existing tenders.

Enter to existing tender

Open a tender

Register to the site and filling personal information, you can add personal items that can be exchanged now or later in personal settings

Share link with pwd

Share link with pwd

Closed tender

Closed tender

Open tender

Open tender

Join with pwd

Closed tender

Open tender

Join via link

Share link

Share link

Choose personal item for replacement

Once the number of participants who entered compatible with tender requirement or it is time for the tender to start, participants will be asked to rate the other participants' items from 1 to n

The algorithm starts as soon as everyone has finished ranking

The algorithm ends with participants seeing the results of the tender.



1. DATA DESIGN

In our project we must deal with a lot of user related information such as login details, personal details, items, previous offers, tenders in which he participated and more.

We chose to use MongoDB which allows to save the data like Jason rather than sql, so there are many functions that are linked to MongoDB that allow us to manipulate the data and use it quickly and conveniently that allows us to perform all the calculations and algorithms that makes our results more accurate and satisfy all users.

**Some entities and objects:**

ITEM will store:

Name, description, type and image.

USER will store:

Email, password in order to login.

And another contact details like phone number and city which that optional.

* Each item will be associated with user.

TENDER will store:

Description of the tender type if it’s a specific tender.

Participants that will user and specific item for each user that he chose to enter with to the tender.

Start time and end time which the participants must be aware to.

1. COMPONENT DESIGN

See **design and functionality explanation.pdf** in order to see what each button will do.

Another algorithm and the most important one is the "trading cycles" algorithm which allows us to bring each user an item in order of priority so that the user does not lose from entering the tender.

Initialize a directed graph: The nodes are the people and items, there is a line which attach between the people and the item he wants the most, and from the items to their owners.

A. Find a directed circle in the graph.

B. Perform the swap in a cycle.

C. The nodes who participating in the swap are deleted from the graph.

D. Update the lines between the people and the items according to their prefers and the items who stayed in the graph.

E. Repeat steps A to D until the graph is empty.

The algorithm can work for many users whenever there is a cycle in the graph which has x people and x items which each wants the other's items. Will make the swap and will continue searching for more cycles. In the "worst case" for a participant, he will stay with his current item. At the end of the algorithm, each participant will know if which product he received from the auction.

The "trading cycle's algorithm" maintains the following features:

1. The algorithm stops.

2. Truthful - It is better for each participant to tell the truth in order to have a better chance of getting the product he wants.

3. Individual rational - user who participated in the process will not lose because he can rate his item.

4. Pareto efficiency (only that strong preference relations).

1. **HUMAN INTERFACE DESIGN** – see **design and functionality explanation.pdf** In order to expect the functionality and user experience.