# **Stevens Marketplace-Database Proposal**

## **Github:**

<https://github.com/liaoyufu-leo/StevensMarketPlace_CS_546_Group16_FinalProject.git>

## **Group Members:**

* Aditya Jayadevan Menon
* Balakishore Kongara
* Yash Gandhi
* YuFu Liao
* Venkat Sai Kusumurthy

## **User Collection**:

The user collections store data of users. Users will be able to register a new account, log in to their account, and update their information. It contains user\_id, account, password, nickname, gender, address, cart.

user = {

"user\_id": ObjectId(),

"account": "yliao10@stevens.edu",

"password": "hash",

"nickname": "Yufu Liao",

"gender": "male",

"address":{

"street": "501 Jackson",

"apt": "Apt 120w",

"city": "Hoboken",

"state": "New Jersey",

"zipCode": "07030"

},

"cart": [

item\_id

]

}

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| user\_id | ObjectId | A globally unique identifier is created automatically to represent the user. |
| account | string | A unique Stevens education email, the user uses this to log in.  Formate: “[xxx@stevens.edu](mailto:xxx@stevens.edu)”; more than 5 preview chars. |
| password | string | The hash of the user’s password,  Formate: more than 10 char; must have upper char and lower char; at least one special char. |
| nickName | string | Preferred name of the user.  Formate: not more than 30 characters |
| gender | string | Preferred gender of the user.  Format: “male/female/other” |
| address | string | The address of the user.  Formate: |
| cart | array | An array to store items’ \_id(string) for the user’s collection.  Formate: [ “item\_id”, ….] |

## **Item Collection**:

The item collection stores data of items published by sellers, users will be able to search items, view items; sellers are able to publish items, update items’ descriptions, and withdraw items. It contains item\_id, seller\_id, status, date, title, price, photos, description.

item = {

"item\_id": ObjectId(),

"seller": "yliao10@stevens.edu",

"status": "selling/sold/withdrawn",

"date": "MM/DD/YY HH:MM:SS",

"title": "Lenovo Laptop",

"price": 800,

"photos": [

"/photos/xxx.jpg"

],

"description": "This is a fresh laptop",

"comments": []

}

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| item\_id | ObjectId | A globally unique identifier is created automatically to represent the item. |
| seller | string | A globally unique identifier comes from the seller’s information. |
| status | string | A string represents the status of an item.  Formate: one of “selling, sold, withdrawn”; |
| date | string | A date format string stores the date and time of when the item is published.  Formate: “MM/DD/YY HH:MM:SS” |
| title | string | The title is edited by the seller  Formate: not more than 30 words |
| price | number | The price of the item  Formate: number >0 |
| photos | array | An array to store photos uploaded by sellers, and elements in the array is the relative link.  Format: [“/photos/xxxx.jpg”,....] |
| description | string | The description of the item.  Formate: not more than 1000 words |
| comments | array | Details are below. |

## **Comment Collection**:

This collection stores the data of all comments made on different products. It contains comment\_id, commenter, content, date.

comment = {

"\_id": ObjectId(),

"commenter": "xxxxx@stevens.edu",

"date": 2021-11-22T21:07:24.074Z,

"content": content,

}

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| \_id | ObjectId | A globally unique identifier created automatically  to represent the comment |
| item\_id | ObjectId | The item user comment |
| commenter | ObjectId | The user’s account of the person who made the comment |
| date | string | ISODate |
| content | string | It is a string that contains the content that the user commented. |

## **Transaction Collection**:

The Transaction Collection collects all the data of the transactions made on different products. It contains trasaction\_id, date, type, status, value.

transaction = {

"transaction\_id": ObjectId(),

"item\_id": ObjectId(),

"seller": "xxxxx@stevens.edu",

"buyer": "xxxxx@stevens.edu",

"date": 2021-11-22T21:07:24.074Z,

"price": 123,

"payment": {

"type": "credit card",

"card number": "1234 1234 1234 1234",

"valid date": "09/26",

"security code": "123"

},

"status": "completed"

}

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| \_id | ObjectId | A globally unique identifier comes from the transaction information. |
| item\_id | ObjectId | The transaction for which item |
| seller | string | The user who sell the time |
| buyer | string | The user who bought the item |
| date | string | A date format string stores the date and time of when the transaction created for the item.  Formate: “MM/DD/YY HH:MM:SS” |
| Price | number | The price of iten |
| payment | Object | The payment stores the method user pay for the item.  "cash"|"credit card" |
| status | string | This will give us the status of the transaction.  "pending" ,"delivered","refund" |

## **Message Collection**:

The Message collection collects all the data of messages that's been sent or received by users. It contains message\_id,users,messages.

message = {

"message\_id": ObjectId(),

"users": [account,account],

"messages": [

{ "sender": account, "date": "MM/DD/YY HH:MM:SS", "content": "string" }

]

}

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| message\_id | ObjectId | A globally unique identifier comes from the messages made. |
| users | ObjectId | It will store the id of both the users messaging |
| messages | Array | An array that stores the messages of the two users.  Formate: [{ "sender": account, "date": "MM/DD/YY HH:MM:SS", "content": "string" }] |

## 