

Interface Specification

Interface Specification

frontend-database

Interface 1 frontend::Login ==> database::Login
Interface 2 frontend::Location ==> database::Location
Interface 3 frontend::Store ==> database::Store
Interface 4 frontend::HuntedStore ==> database::HuntedStore
Interface 5 database::HuntedStore ==> frontend::HuntedStore
Interface 6 frontend::Item ==> database::Item
Interface 7 frontend::Map ==> database::Map
Interface 8 frontend::Feedback ==> database::Feedback
Interface 9 frontend::Feedback ==> database::Feedback
Interface 10 frontend::Customer ==> database::Customer
Interface11 database::Customer ==> frontend::Customer
Interface 12 frontend::Registration ==> database::Registration

algorithm-database

Interface 13 database::Store ==> algorithm::Store
Interface 14 database::Customer ==> algorithm::user
Interface 15 database::Item ==> algorithm::recommendation algorithm
Interface 16 algorithm::recommendation algorithm ==> database::

web-database

Interface 17 web::Registration ==> database::Registration
Interface 18 database::Registration ==> web::Registration
Interface 19 database::Login ==> web::Login
Interface 20 web::Management ==> database::Administrator
Interface 21 database::Administrator ==> web::Management
Interface 22 database::Analytics ==> web::Analytics
Interface 23 database::Store ==> web::StoreOwner
Interface 24 web::StoreOwner ==> database::Store
Interface 25 web::StoreInfo ==> database::Store
Interface 26 web::User ==> database::Store
Interface 27 database::Store ==> web::User
Interface 28 database::Feedback ==> web::Feedback
Interface 29 database::Item ==> web::Item
Interface 30 web::Item ==> database::Item
Interface 31 database::Store ==> web::Location
Interface 32 database::Item ==> web::Statistics

Appendix:

object::Location
object::Item
object::Store
object::Feedback
object::User
object::Statistics
object::EngagementData

update from v.4 to v.5:

frontend-database

Interface 1 frontend::Login ==> database::Login

```
{
  "InterfaceId": {
    "type": "integer",
    "minimum": 1,
    "maximum": 32
  },
  "CurrentUser": {"type": "string"},
  "UserName": {
    "type": "string",
    "minLength": 1
  },
  "Password": {
    "type": "string",
    "minLength": 1
  }
}
```

return with

```
{
  "InterfaceId":{
    "type": "integer",
    "minimum": 1,
    "maximum": 32
  },
  "CurrentUser": {
    "type": "string",
    "minLength": 1
  },
  "MatchToken": {"type": "boolean"}
}
```

- Tip1: InterfaceId means the id of interface that we use to identify the interface index and json content. We have 32 interfaces in total so the InterfaceId will range from 1 to 32.
- Tip 2: CurrentUser is None if this is an anonymous visitor, otherwise it is the UserName.
- Description of the interface: When the user logs in, this interface is called. On the front end, the user needs to input the Username and Password, which are then sent to the database. The database performs verification, and the result of the verification is indicated by the returned match_token.

Interface 2 frontend::Location ==> database::Location

```
{
  "InterfaceId":{
    "type": "integer",
    "minimum": 1,
    "maximum": 32
  },
  "CurrentUser": {
    "type": "string",
```

```

        "minLength": 1
    },
    "CurrentLocation": {
        "type": "object::Location",
        "properties": "Illustrated in appendix"
    }
}

```

- Description of the interface: When the user needs data related to location information, such as viewing a map, they need to provide the CurrentLocation to the database.

Interface 3 frontend::Store ==> database::Store

```

{
    "InterfaceId":{
        "type":"integer",
        "minimum": 1,
        "maximum": 32
    },
    "CurrentUser": {
        "type": "string",
        "minLength": 1
    },
    "StoreName": {
        "type": "string",
        "minLength": 1
    }
}

return with

{
    "InterfaceId":{
        "type":"integer",
        "minimum": 1,
        "maximum": 32
    },
    "CurrentUser": {
        "type": "string",
        "minLength": 1
    },
    "StoreList": {
        "type": "array",
        "items": {
            "type": "object::Store",
            "properties": "Illustrated in appendix"
        },
        "minItems": 0,
        "uniqueItems": true
    }
}

```

- Tip1: This provides the function of querying the store information given the store name.

- Description of the interface: When a user wants to inquire about store information, they provide the StoreName to the database. Since the StoreName provided by the user might be incomplete or there may be stores with the same name, the database should return a list of stores that meet the criteria.

Interface 4 frontend::HuntedStore ==> database::HuntedStore

```
{
  "InterfaceId":{
    "type":"integer",
    "minimum": 1,
    "maximum": 32
  },
  "CurrentUser": {
    "type": "string",
    "minLength": 1
  },
  "HuntedStoreIdList": {
    "type": "array",
    "items": {
      "type": "integer",
      "minimum": 0
    },
    "minItems": 0,
    "uniqueItems": true
  }
}
```

- Tips1: HuntedList is like a "Browsing history".
- Description of the interface: When a user browses new stores, they need to provide the HuntedList to the database for updating the content stored in the database.

Interface 5 database::HuntedStore ==> frontend::HuntedStore

```
{
  "InterfaceId":{
    "type":"integer",
    "minimum": 1,
    "maximum": 32
  },
  "CurrentUser": {
    "type": "string",
    "minLength": 1
  },
  "HuntedStoreIdList": {
    "type": "array",
    "items": {
      "type": "integer",
      "minimum": 0
    },
    "minItems": 0,
    "uniqueItems": true
  }
}
```

- Tips1: This is for displaying the HuntedList.
- Description of the interface: When a user needs to view their browsing history, the database provides information about the HuntedStoreIdList to the front end.

Interface 6 frontend::Item ==> database:Item

```
{
  "InterfaceId":{
    "type":"integer",
    "minimum": 1,
    "maximum": 32
  },
  "CurrentUser": {
    "type": "string",
    "minLength": 1
  },
  "ItemName": {
    "type": "string",
    "minLength": 1
  }
}

return with

{
  "InterfaceId":{
    "type":"integer",
    "minimum": 1,
    "maximum": 32
  },
  "CurrentUser": {
    "type": "string",
    "minLength": 1
  },
  "ItemList": {
    "type": "array",
    "items": {
      "type": "Item",
      "properties": "Illustrated in appendix"
    },
    "minItems": 0,
    "uniqueItems": true
  }
}
```

- Tips1: This provides the function of querying the item information given the item name.
- Description of the interface: When a user wants to inquire about item information, they provide the ItemName to the database. Since the ItemName provided by the user might be incomplete or there may be items with the same name, the database should return a list of items that meet the criteria.

Interface 7 frontend::Map ==> database::Map

```
{
  "InterfaceId":{
    "type":"integer",
    "minimum": 1,
    "maximum": 32
  },
  "CurrentUser": {
    "type": "string",
    "minLength": 1
  },
  "MyLocation": {
    "type": "object::Location",
    "properties": "Illustrated in appendix"
  },
  "RequestType": {
    "type":"integer",
    "minimum": 1,
    "maximum": 2
  },
}

return with

{
  "InterfaceId":{
    "type":"integer",
    "minimum": 1,
    "maximum": 32
  },
  "CurrentUser": {
    "type": "string",
    "minLength": 1
  },
  "StoreList": {
    "type": "array",
    "items": {
      "type": "object::Store",
      "properties": "Illustrated in appendix"
    },
    "minItems": 0,
    "uniqueItems": true
  }
}
```

- Description of the interface: RequestType=1 means this a request searching for nearby stores provided with "MyLocation". RequestType=2 means this is a request for a list recommended stores. The user provides the RequestType and MyLocation to the database to obtain the needed StoreList.

Interface 8 frontend::Feedback ==> database::Feedback

```
{
  "InterfaceId":{
    "type":"integer",
    "minimum": 1,
    "maximum": 32
  },
  "CurrentUser": {
    "type": "string",
    "minLength": 1
  },
  "StoreId": {
    "type": "integer",
    "minimum": 0
  },
  "comment": {
    "type": "string",
    "minLength": 0
  },
  "rating": {
    "type":"integer",
    "minimum": 1,
    "maximum": 10
  }
}
```

- Tip1 : Feedback is a rating that evaluate the recommendation results.
- Tip2 : "rating" should be a positive integer from 1 to 10.
- Description of the interface: Users evaluate the store recommendations provided by the recommendation system, specifically including comments and ratings, which are transmitted to the database.

Interface 9 frontend::Feedback ==> database::Feedback

```
{
  "InterfaceId":{
    "type":"integer",
    "minimum": 1,
    "maximum": 32
  },
  "CurrentUser": {
    "type": "string",
    "minLength": 1
  },
  "Item": {
    "type":"object::Item",
    "properties": "Illustrated in appendix"
  },
  "comment": {
    "type": "string",
    "minLength": 0
  },
  "rating": {
```

```

        "type": "integer",
        "minimum": 1,
        "maximum": 10
    }
}

```

- Description of the interface: Similar to Interface8. Users evaluate the recommended item results from the recommendation system, specifically including comments and ratings, which are transmitted to the database.

Interface 10 frontend::Customer ==> database:: Customer

```

{
  "InterfaceId": {
    "type": "integer",
    "minimum": 1,
    "maximum": 32
  },
  "CurrentUser": {
    "type": "string",
    "minLength": 1
  },
  "UserName": {
    "type": "string",
    "minLength": 1
  },
  "Password": {
    "type": "string",
    "minLength": 1
  },
  "Birthday": {
    "type": "date"
  },
  "Interests": {
    "type": "array",
    "items": "integer",
    "minimum": 0
  },
  "CurrentLocation": {
    "type": "object::Location",
    "properties": "Illustrated in appendix"
  }
}

```

- Tip1: This is used to update customer information stored in the database.
- Description of the interface: When a user wishes to update their information, they provide the database with new details such as Birthday, Interests, and CurrentLocation. It should be noted that for security reasons, before updating the information, the user needs to re-enter their Username and Password for verification.

Interface11 database::Customer ==> frontend:: Customer

```
{
  "InterfaceId":{
    "type":"integer",
    "minimum": 1,
    "maximum": 32
  },
  "CurrentUser": {
    "type": "string",
    "minLength": 1
  },
  "UserName": {
    "type": "string",
    "minLength": 1
  },
  "Password": {
    "type": "string",
    "minLength": 1
  },
  "Birthday": {
    "type": "date"
  },
  "Interests": {
    "type": "array",
    "items": "integer",
    "minimum": 0
  },
  "CurrentLocation":{
    "type": "object::Location",
    "properties": "illustrated in appendix"
  }
}
```

- Tip1: This is used to show the customer information on the screen. Password should be set to "None".
- Description of the interface: The database provides the current user with relevant information about other users, without the need to enter a password.

Interface 12 frontend::Registration ==> database::Registration

```
{
  "InterfaceId":{
    "type":"integer",
    "minimum": 1,
    "maximum": 32
  },
  "CurrentUser": {
    "type": "string",
    "minLength": 1
  },
  "UserName": {
    "type": "string",
    "minLength": 1
  }
}
```

```

    },
    "Password": {
        "type": "string",
        "minLength": 1
    }
}

return with
{
    "InterfaceId":{
        "type":"integer",
        "minimum": 1,
        "maximum": 32
    },
    "CurrentUser": {
        "type": "string",
        "minLength": 1
    },
    "successful_token": {"type": "boolean"}
}

```

- Tip1: this is used for registration and to check whether the UserName is unique.
- Description of the interface: Users register by entering their Username and Password on the front end, which are then transmitted to the database. The database checks if the same Username already exists and returns a successful_token to the front end to indicate whether the registration was successful or not.

algorithm-database

Interface 13 database::Store ==> algorithm::Store

```

{
    "InterfaceId":{
        "type":"integer",
        "minimum": 1,
        "maximum": 32
    },
    "CurrentUser": {
        "type": "string",
        "minLength": 1
    },
    "Items": {
        "type": "array",
        "items": {
            "type": "object::Item",
            "properties": "Illustrated in appendix"
        },
        "minItems": 1,
        "uniqueItems": true
    },
}

```

```

    "Location": {
      "type": "object::Location",
      "properties": "Illustrated in appendix"
    }
  }
}

```

Interface 14 database::Customer ==> algorithm::user

```

{
  "InterfaceId":{
    "type":"integer",
    "minimum": 1,
    "maximum": 32
  },
  "CurrentUser": {
    "type": "string",
    "minLength": 1
  },
  "UserId": {
    "type": "integer",
    "minimum": 0
  },
  "UserData": {
    "type": "array",
    "items": {
      "type": "object::Customer",
      "properties": "Illustrated in appendix"
    },
    "minItems": 1,
    "uniqueItems": true
  }
}

```

Interface 15 database::Item ==> algorithm::recommendation algorithm

```

{
  "InterfaceId":{
    "type":"integer",
    "minimum": 1,
    "maximum": 32
  },
  "CurrentUser": {
    "type": "string",
    "minLength": 1
  },
  "ItemData": {
    "type": "array",
    "items": {
      "type": "object::Item",
      "properties": "Illustrated in appendix"
    },
    "minItems": 1,
    "uniqueItems": true
  }
}

```

```
}
```

Interface 16 algorithm::recommendation algorithm ==> database::

```
{
  "InterfaceId":{
    "type":"integer",
    "minimum": 1,
    "maximum": 32
  },
  "CurrentUser": {
    "type": "string",
    "minLength": 1
  },
  "RecommendedStores": {
    "type": "array",
    "items": {
      "type": "object::Item",
      "properties": "Illustrated in appendix"
    },
    "minItems": 1,
    "uniqueItems": true
  }
}
```

web-database

Interface 17 web::Registration ==> database::Registration

```
{
  "InterfaceId":{
    "type":"integer",
    "minimum": 1,
    "maximum": 32
  },
  "CurrentUser": {
    "type": "string",
    "minLength": 1
  },
  "UserName": {
    "type": "string",
    "minLength": 1
  },
  "HashedPassword": {
    "type": "string",
    "minLength": 1
  }
}
```

Interface 18 database::Registration ==> web::Registration

```
{
  "InterfaceId":{
    "type":"integer",
    "minimum": 1,
    "maximum": 32
  },
  "CurrentUser": {
    "type": "string",
    "minLength": 1
  },
  "UserName": {
    "type": "string",
    "minLength": 1
  },
  "HashedPassword": {
    "type": "string",
    "minLength": 1
  }
}
```

Interface 19 database::Login ==> web::Login

```
{
  "InterfaceId":{
    "type":"integer",
    "minimum": 1,
    "maximum": 32
  },
  "CurrentUser": {
    "type": "string",
    "minLength": 1
  },
  "UserName": {
    "type": "string",
    "minLength": 1
  },
  "HashedPassword": {
    "type": "string",
    "minLength": 1
  },
}
```

Interface 20 web::Management ==> database::Administrator

```
{
  "InterfaceId":{
    "type":"integer",
    "minimum": 1,
    "maximum": 32
  },
  "CurrentUser": {
    "type": "string",
```

```

    "minLength": 1
  },
  "UserData" : {
    "type": "object::User",
    "properties": "Illustrated in appendix"
  },
  "StoreInfo" : {
    "type": "object::Store",
    "properties": "Illustrated in appendix"
  }
}

```

Interface 21 database::Administrator ==> web::Management

```

{
  "InterfaceId": {
    "type": "integer",
    "minimum": 1,
    "maximum": 32
  },
  "CurrentUser": {
    "type": "string",
    "minLength": 1
  },
  "UserData": {
    "type": "object::User",
    "properties": "Illustrated in appendix"
  },
  "StoreInfo": {
    "type": "object::Store",
    "properties": "Illustrated in appendix"
  }
}

```

Interface 22 database::Analytics ==> web::Analytics

```

{
  "InterfaceId":{
    "type": "integer",
    "minimum": 1,
    "maximum": 32
  },
  "CurrentUser": {
    "type": "string",
    "minLength": 1
  },
  "UserEngagementData": {
    "type": "object::EngagementData",
    "properties": "Illustrated in appendix"
  },
  "Advice": {
    "type": "string",
    "miniumLength": 0
  }
}

```

```
}
```

- Tip1 : Advice is the overall advice proposed by LLM to improve the whole system.

Interface 23 database::Store ==> web::StoreOwner

```
{
  "InterfaceId":{
    "type":"integer",
    "minimum": 1,
    "maximum": 32
  },
  "CurrentUser": {
    "type": "string",
    "minLength": 1
  },
  "Statistics": {
    "type": "object::Statistics",
    "properties": "Illustrated in appendix"
  }
}
```

Interface 24 web::StoreOwner ==> database::Store

```
{
  "InterfaceId":{
    "type":"integer",
    "minimum": 1,
    "maximum": 32
  },
  "CurrentUser": {
    "type": "string",
    "minLength": 1
  },
  "StoreInfo": {
    "type": "object::StoreInfo",
    "properties": "Illustrated in appendix"
  }
}
```

Interface 25 web::StoreInfo ==> database::Store

```
{
  "InterfaceId":{
    "type":"integer",
    "minimum": 1,
    "maximum": 32
  },
  "CurrentUser": {
    "type": "string",
    "minLength": 1
  },
  "StoreName": {
    "type": "string",
    "minLength": 1
  }
}
```

```

    },
    "Location" : {
        "type": "object::Location",
        "properties": "Illustrated in appendix"
    },
    "Items": {
        "type": "array",
        "items": {
            "type": "object::Item",
            "properties": "Illustrated in appendix"
        },
        "minItems": 1,
        "uniqueItems": true
    }
}

```

Tip1: The UserName and Password of the store owner should be stored in "StoreInfo" class in database.

Interface 26 web::User ==> database::Store

```

{
    "InterfaceId":{
        "type":"integer",
        "minimum": 1,
        "maximum": 32
    },
    "CurrentUser": {
        "type": "string",
        "minLength": 1
    },
    "UserName": {
        "type": "string",
        "minLength": 1
    },
    "HashedPassword": {
        "type": "string",
        "minLength": 1
    }
}

```

Interface 27 database::Store ==> web::User

```

{
    "InterfaceId":{
        "type":"integer",
        "minimum": 1,
        "maximum": 32
    },
    "CurrentUser": {
        "type": "string",
        "minLength": 1
    },
    "UserName": {
        "type": "string",

```



```

        "minLength": 1
    },
    "HashedPassword": {
        "type": "string",
        "minLength": 1
    }
}

```

Interface 28 database::Feedback ==> web::Feedback

```

{
    "InterfaceId":{
        "type":"integer",
        "minimum": 1,
        "maximum": 32
    },
    "CurrentUser": {
        "type": "string",
        "minLength": 1
    },
    "Feedback": {
        "type": "object::Feedback",
        "preproperties": "Illustrated in appendix"
    }
}

```

Interface 29 database::Item ==> web::Item

```

{
    "InterfaceId":{
        "type":"integer",
        "minimum": 1,
        "maximum": 32
    },
    "CurrentUser": {
        "type": "string",
        "minLength": 1
    },
    "Item": {
        "type": "object:Item",
        "preproperties": "Illustrated in appendix"
    }
}

```

Interface 30 web::Item ==> database::Item

```

{
    "InterfaceId":{
        "type":"integer",
        "minimum": 1,
        "maximum": 32
    },
    "CurrentUser": {
        "type": "string",

```

```

        "minLength": 1
    },
    "Item": {
        "type": "object:Item",
        "properties": "Illustrated in appendix"
    }
}

```

Interface 31 database::Store ==> web::Location

```

{
    "InterfaceId":{
        "type":"integer",
        "minimum": 1,
        "maximum": 32
    },
    "CurrentUser": {
        "type": "string",
        "minLength": 1
    },
    "Location": {
        "type": "object:Location",
        "properties": "Illustrated in appendix"
    }
}

```

Interface 32 database::Item ==> web::Statistics

```

{
    "InterfaceId":{
        "type":"integer",
        "minimum": 1,
        "maximum": 32
    },
    "CurrentUser": {
        "type": "string",
        "minLength": 1
    },
    "CustomerVisits": {
        "type": "integer"
    },
    "Feedbacks": {
        "type": "array",
        "items": {
            "type": "object::Feedback",
            "properties": "Illustrated in appendix"
        },
        "minItems": 1,
        "uniqueItems": true
    }
}

```

Appendix:

object::Location

```
{
  "latitude": {
    "type": "number"
  },
  "longitude": {
    "type": "number"
  },
  "country" : {
    "type": "string",
    "minLength": 0
  },
  "state" : {
    "type": "string",
    "minLength": 0
  },
  "city": {
    "type": "string",
    "minLength": 0
  },
  "street": {
    "type": "string",
    "minLength": 0
  },
  "number": {
    "type": "string",
    "minLength": 0
  },
  "floor": {
    "type": "string",
    "minLength": 0
  },
  "zipcode": {
    "type": "string",
    "minLength": 0
  }
}
```

object::Item

```
{
  "ItemId": {
    "type": "integer",
    "minLength": 0
  },
  "ItemName": {
    "type": "string",
    "minLength": 1
  },
  "ItemPrice": {
    "type": "number",
    "minimum": 0
  }
}
```

```

    },
    "ItemDescription": {
      "type": "string",
      "minLength": 0
    },
    "ItemImage": {
      "type": "base64",
      "minLength": 0
    },
    "ItemStoreId": {
      "type": "integer",
      "minLength": 0
    },
    "ItemStoreName": {
      "type": "string",
      "minLength": 1
    },
    "customerVisits": {
      "type": "integer"
    }
  }
}

```

- Tip1: Same items in different stores have different ItemId, which means ItemId is unique.

object::Store

```

{
  "storeId": {
    "type": "integer",
    "minLength": 0
  },
  "storeName": {
    "type": "string",
    "minLength": 1
  },
  "location": {
    "type": "object:Location",
    "properties": "Illustrated in appendix"
  },
  "items": {
    "type": "array",
    "items": {
      "type": "object::Item",
      "properties": "Illustrated in appendix"
    },
    "minItems": 1,
    "uniqueItems": true
  },
  "StoreDescription": {
    "type": "string",
    "minLength": 0
  }
}

```

object::Feedback

```
{
  "Item": {
    "type": "object::Item",
    "properties": "Illustrated in appendix"
  },
  "comment": {
    "type": "string",
    "minLength": 0
  },
  "rating": {
    "type": "integer",
    "minimum": 1,
    "maximum": 10
  },
  "UserName": {
    "type": "string",
    "minLength": 1
  }
}
```

object::User

```
{
  "userId": {
    "type": "integer",
    "minimum": 0,
  },
  "UserEmail": {
    "type": "email"
  },
  "UserName": {
    "type": "string",
    "minLength": 1
  },
  "HashedPassword": {
    "type": "string",
    "minLength": 1
  }
}
```

object::Statistics

```
{
  "customerVisits": {
    "type": "integer"
  }
  "feedbacks": {
    "type": "array",
    "items": {
      "type": "object::Feedback",
      "properties": "Illustrated in appendix"
    },
  },
}
```

```

    "minItems": 0,
    "uniqueItems": true
  }
}

```

object::EngagementData

```

{
  "AvgRating": {
    "type": "number",
    "minimum": 0
  },
  "SelectedComment": {
    "type": "array",
    "items": {
      "type": "string"
    },
    "minItems": 1,
    "uniqueItems": true
  }
}

```

- Tip1: AvgRating is the average rating of a particular store.
- Tip2: SelectedComment is the low scored comment by user selected as reference for refining the system

update from v.4 to v.5:

1. Use JSON SCHEMA to make it more formal.
 1. Follow [English Version](#) / [Chinese Version](#).
 2. Make some slight changes for "object" type for readability.
 3. Add more restrictions on the data formats like the minimum value of integer or the minimum length of string. However, these restrictions may **not be precise before each PM from each group checking them**.
2. Remove the comma after the last item in the JSON.
3. Set unique interface ID.
4. Unified the naming convention to use PascalCase consistently. (except for the formats in JSON SCHEMA)