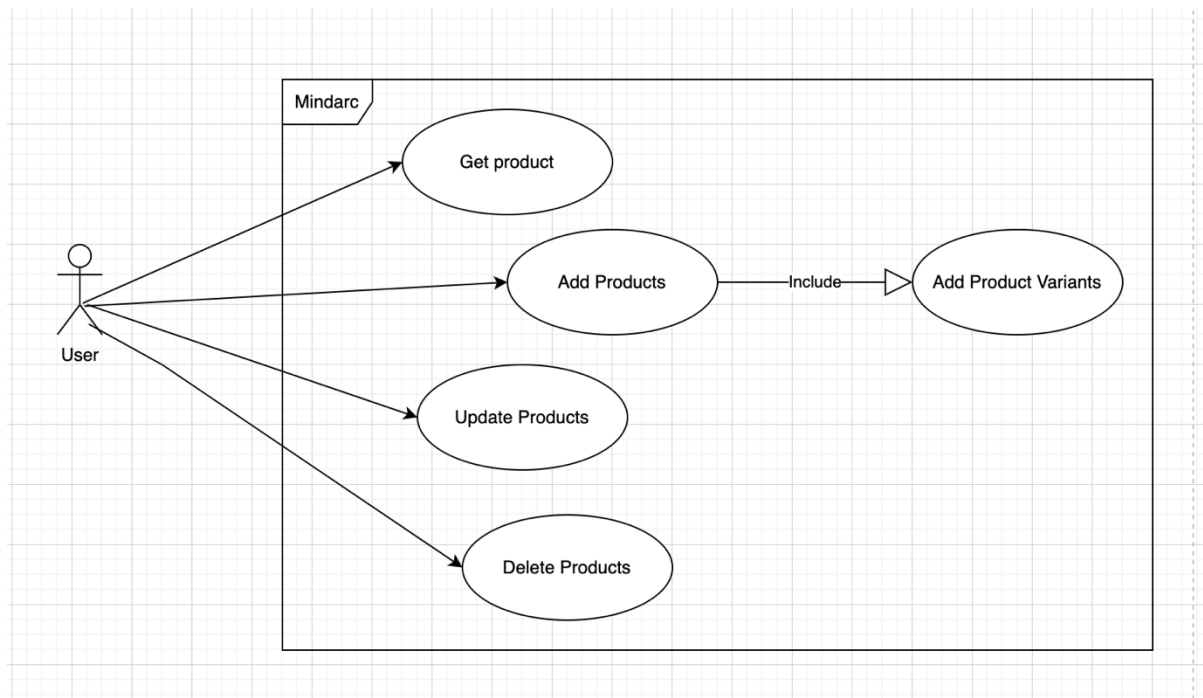


MINDARC ASSESSMENT

USE CASE

We have a main actor in our system, User/Tenant



A user can conduct some activity in the Mindarc Product Management System

- Get Product(s)
- Add product(s)
- Update product(s)
- Delete product(s)

CLASS DIAGRAM

Product: The central part of the application

Variant: The additional information added to identify different variants of a product in the system

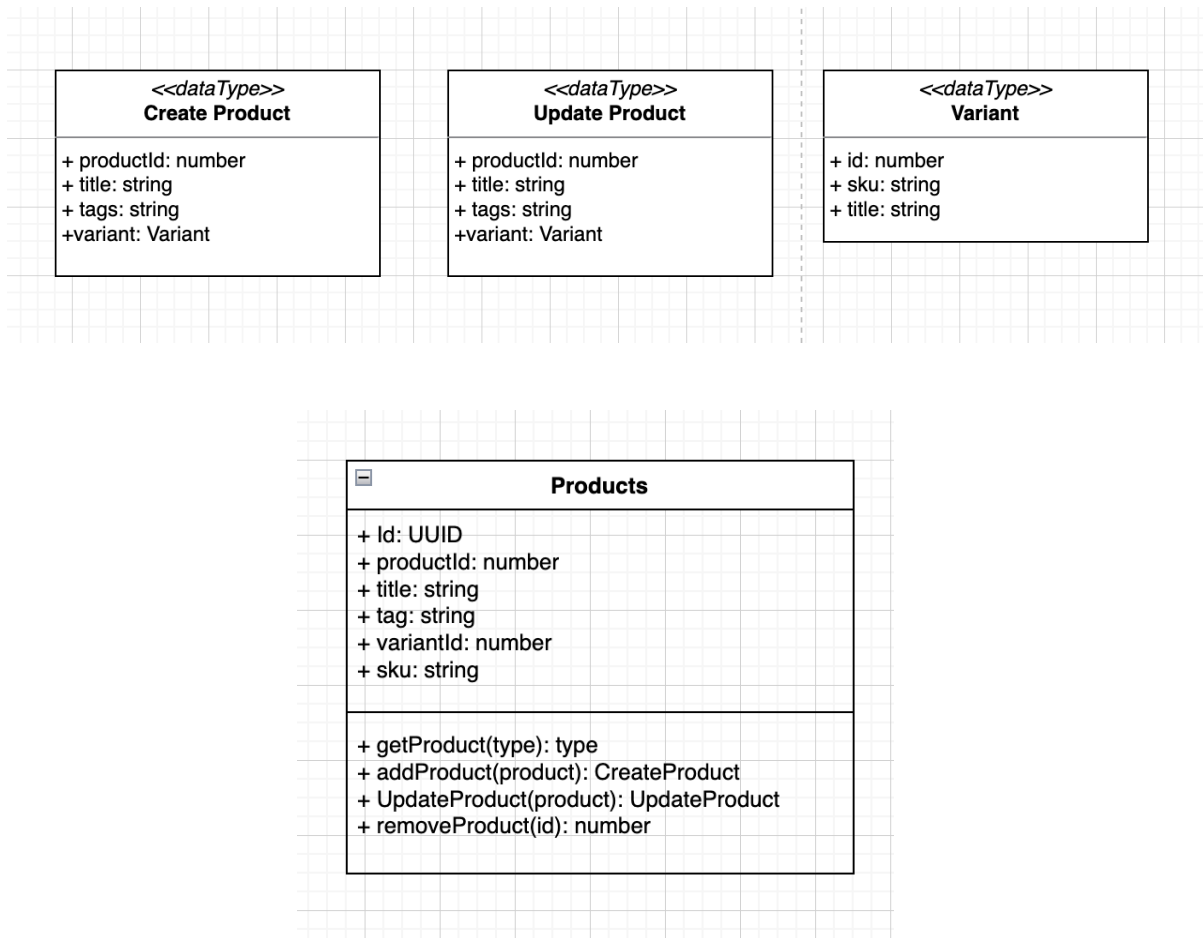
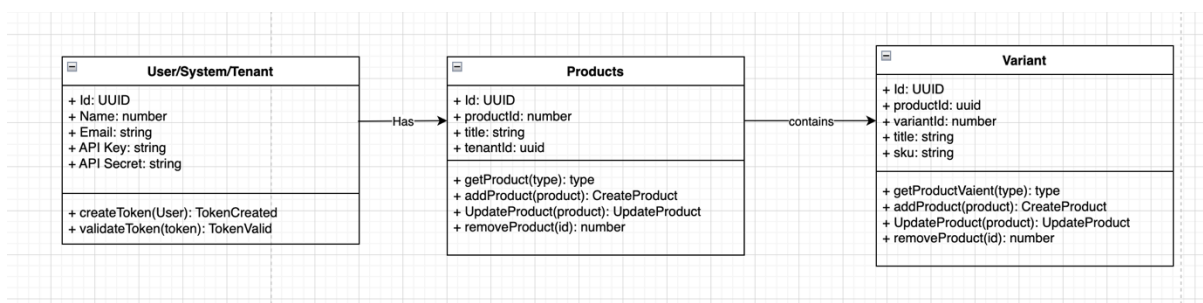


FIGURE 1: CLASS DIAGRAM FOR EXISTING APPLICATION

Future Improvement

- Added new features to register users/tenants into the system.
- Only valid tenants/users are able to create/get/update/delete from the system.
- Hard delete needs to be replaced by soft delete by having a column `isDeleted` in the Product and Variants entity.
- Break down the Product and Variant into 2 different tables, making it easier to handle data changes in the future.



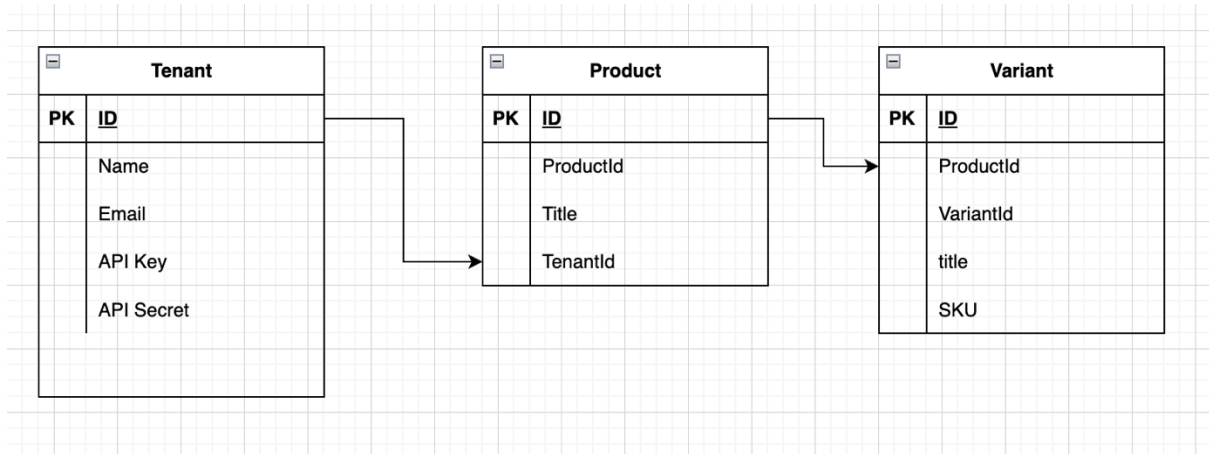


FIGURE 2: ENTITY DIAGRAM FOR FUTURE IMPROVEMENT

APPLICATION IMPLEMENTATION

DB Credential

DATABASE_HOST=aws.connect.psdb.cloud

DATABASE_NAME=mindarc

DATABASE_USERNAME= hb8vm7fga8o4ucmdxyn3

DATABASE_PASSWORD=pscale_pw_CN1y2Kn0VV3ewpU9DSKb0MqGGpZQKULdkWZlqYCDTpC

Implementation

- **API:** According to the assessment, 5 APIs are implemented with different roles. I have added another Post API which allows the user to post new products by entering JSON into the body while the other one by entering the URL.

GET	/api/products	Create a product by entering url	▼
POST	/api/products	Create products	▼
PUT	/api/products	Update products	▼
POST	/api/products/url	Create products by entering url	▼
DELETE	/api/products/{product_id}	Delete products	▼

FIGURE 3: IMPLEMENTED APIS

- Error Handling

- Created a template for error response to client-side so the front end can easily handle it.

```
{
  "isSuccess": false,
  "error": {
    "statusCode": 500,
    "message": "Could not retrieve data from endpoint. Please try again later!",
    "error": "Internal Server Error"
  },
  "timestamp": "2023-09-24T22:15:59.697Z",
  "path": "/api/products?url=test"
}
```

FIGURE 4: CUSTOM ERROR MESSAGE WITH DEFAULT TEMPLATE

- User friendly message
- Unit Testing
- Deployment: Using Docker Compose to deploy the application