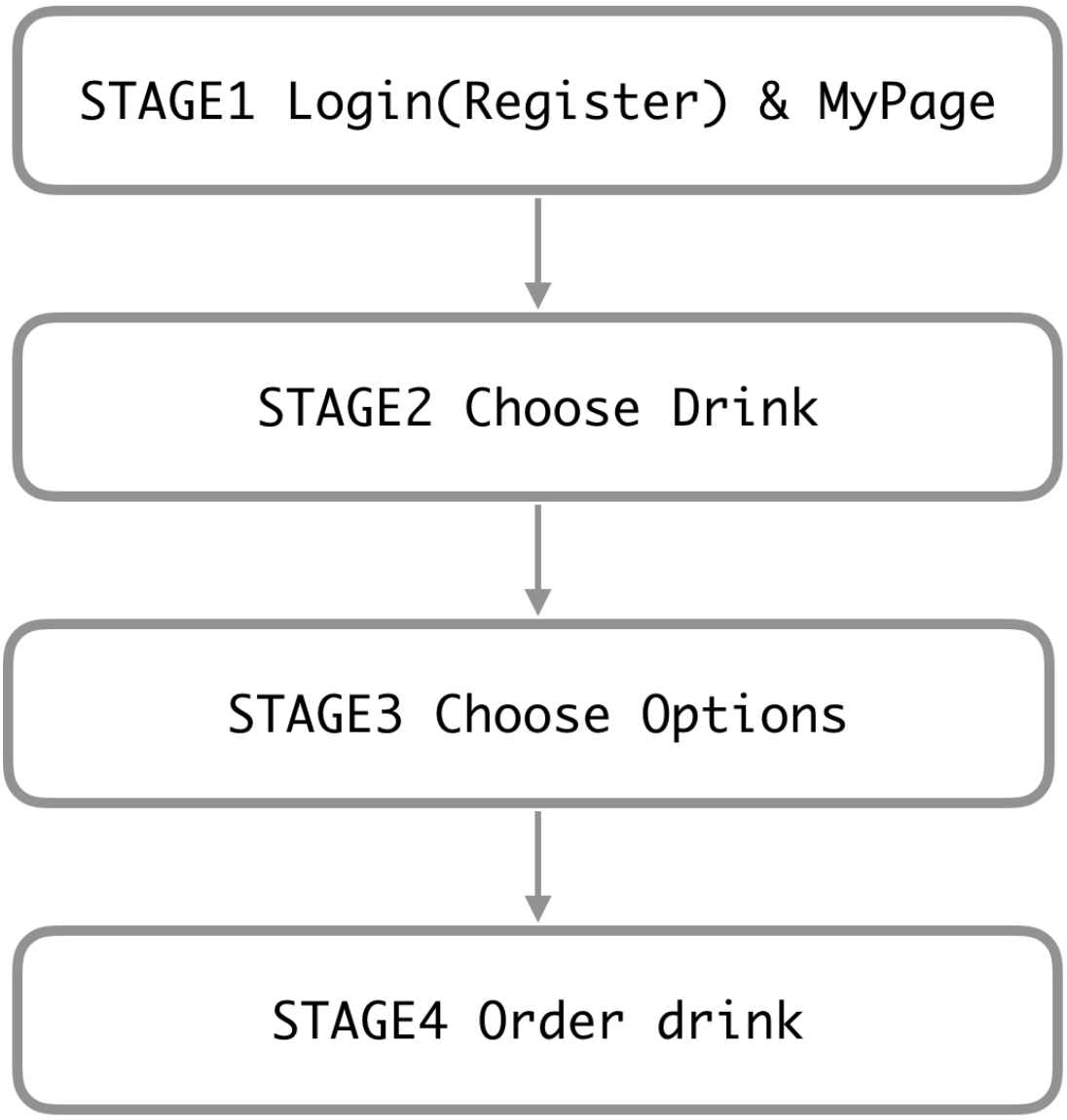
Customized Drink Recommendation

Scenarios and DML

Team No	**13**
Team Name	**DBzara**
Team Member	**Yoo ByeongHeon**
	Kim SaeNa
	Han BoKyung
	Kim HyoRim

I. Overall Scenario



II. Detail Scenario

1) STAGE 1 Login(or Register) & MyPage

Use Case: Register

Actor : User

Precondition: Program compile

Scenario:

Input Events from User	*System Events and Responses*
#1. On the first screen, the user clicks the Register button. #2. User enters the ID and clicks the Verify ID button. #3. After the user fills in all the fields of the frame, click on the Register button.	#1. Change the frame to the register page. #2. If the same id is found after checking the database for the same id, a warning message is sent. If the same id does not exist, change the id verify flag. #3.Verify that the fields are all filled, verify that the id check flag matches the condition, and then send the query to register the user.

Example Queries

#1

- **SELECT** user_id **FROM** userList **WHERE** user_id="asdf";

#2

- **INSERT INTO** userList **VALUES(**"asdf", "asdf", "1","M","ad"**)**;

Use Case: Login

Actor : User

Precondition: User is already registered.

Scenario:

Input Events from User	***System Events and Responses***
#1. User enters id and password.	#1. After confirming registered id and pass word, save User id and go to main page.

Example Queries

#1

- **SELECT** user_id, user_name, age, gender **FROM** userList **WHERE** user_id="BestJC" **AN D** passwd="BestJC";

Use Case: Show Picked List

Actor : User

Precondition: User is already registered.

Scenario:

Input Events from User	***System Events and Responses***
#1. Click PickedList on 'MyPage'	#1. Show tuple from PickedList whose user_id is same with this user.
#2. Actor click 'close' button	#2. Close window

Example Queries

#1

- **select** b.drink_id, a.drink_name FROM drink a **inner join** pickedList b on a.drink_id=b.drink_id **inner join** userList c on c.user_id=b.user_id WHERE (c.user_name='user_name')

Use Case: Show list of items which was ordered by user

Actor : User

Precondition: User is registered in this system

Scenario:

Input Events from User	***System Events and Responses***
#1. Click 'Ordered List' in My page	#1. Show the list user ordered in the past
#2. User click 'close' button	#2. Close 'Ordered List' page

Example Queries

#1

- **SELECT** user_id FROM userList **WHERE** user_name=user_name(variable) ;
- **SELECT** b.drink_name, c.option1, c.option2, c.option3, d.count **FROM** drink b **inner jo in** orderedList d on b.drink_id=d.drink_id **inner join** chooseList c on c.choose_id=d.c hoose_id WHERE d.user_id=userID;

Use Case: User add new disease to his disease list

Actor : User

Precondition: User is already registered

Scenario:

Input Events from User	***System Events and Responses***
#1. User click 'Add Disease' button on My Page #2. User inserts disease id to text filed and click 'INSERT' button	#1. System shows the list of all diseases and the filed to insert new disease #2. System adds the disease user inserted to disease list.

Example Queries

#1

- **SELECT** * FROM diseasetable
  : To show the list of all disease that user can add.
- **SELECT** user_id FROM userList WHERE user_name='user_name(variable)'
  : The User's id is needed when adding new disease to the sufferList

#2

- **INSERT** INTO sufferList(user_id, disease_id) **VALUES**('user_id(var)', 'AddDisID(value of disease id)')

Use Case: User checks his own list of disease

Actor : User

Precondition: User is already registered in this system.

Scenario:

Input Events from User	***System Events and Responses***
#1. User clicks 'Check Disease' Button on My Page	#1. System bring the list of diseases he is suffering

Example Queries

#1

- **SELECT** disease_id, disease_name **FROM** diseasetable **WHERE** disease_id **IN** (**SELECT** disease_id **FROM** sufferList **WHERE** user_id='user_id')

Use Case: User deletes his disease after he got better

Actor : User

Precondition: User is already registered in this system

Scenario:

Input Events from User	***System Events and Responses***
#1. User clicks 'Delete Disease' button on My Page & inserts disease id which is going to be deleted	#1. System deletes the tuple which has user id, disease id.

Example Queries

#1

- **DELETE FROM** sufferList **WHERE** disease_id='disease_id' **AND** user_id='user_id'

Use Case: User changes his password

Actor : User

Precondition: User is already registered in this system

Scenario:

Input Events from User	***System Events and Responses***
#1. User clicks 'Change Password' button #2. User insert new password and press 'change' button	#1. System shows present password and the text field to get new password #2. System updates new password

Example Queries

#1

- **SELECT** passwd **FROM** userList

#2

- **UPDATE** userList **SET** passwd='newPW' **WHERE** user_id='user_id'

2) STAGE 2 Choose Drink

Use Case: Search for drink by cafe

Actor : User

Precondition: User logged in.

Scenario:

Input Events from User	***System Events and Responses***
#1. User clicks 'order' button.	#1. System shows a next page. The next page are included 'cafe' button, 'new' button, and 'Drink Recommend' button.
#2. User clicks 'cafe' button.	#2. System shows next page. The next page are included 'hisbeans' button, 'EDIYA' button, 'Mom's cafe' button, and 'Applen In the Tree' button
#3. User selects one of four cafe he/she wants.	#3. System shows a list of drinks.

Example Queries

#1

- **SELECT** * FROM drink **INNER JOIN** drinkandcafe **ON** drink.drink_id = drinkandcafe.drink_id **WHERE** drinkandcafe.cafe_id = 1;(Hisbeans)
- **SELECT** * FROM drink **INNER JOIN** drinkandcafe **ON** drink.drink_id = drinkandcafe.drink_id **WHERE** drinkandcafe.cafe_id = 2;(EDIYA)
- **SELECT** * FROM drink **INNER JOIN** drinkandcafe **ON** drink.drink_id = drinkandcafe.drink_id **WHERE** drinkandcafe.cafe_id = 3;(Mom's cafe)
- **SELECT** * FROM drink **INNER JOIN** drinkandcafe **ON** drink.drink_id = drinkandcafe.drink_id **WHERE** drinkandcafe.cafe_id = 4;(Apple In the Tree)

Use Case: Search for new drinks

Actor : User

Precondition: User logged in.

Scenario:

Input Events from User	***System Events and Responses***
#1. User clicks 'order' button. #2. User clicks 'new' button.	#1. System shows a next page. The next page are included 'cafe' button, 'new' button, and 'Drink Recommend' button. #2. System shows a list of drinks which is released in last three months.

Exceptional Scenarios :

- If you do not have a freshly released drink in the last three months, nothing will pop up.

Example Queries

#1

- **SELECT** drink_name, price, temparature **FROM** drink **WHERE** release_date **LIKE** '%18/5%' **OR** release_date **LIKE** '%18/4%' **OR** release_date **LIKE** '%18/3%' ;

Use Case: Search for drinks that meet the conditions user choosed

Actor : User

Precondition: User logged in.

Scenario:

Input Events from User	***System Events and Responses***
#1. User clicks 'order' button.	#1. ystem shows a next page. The next page are included 'cafe' button, 'new' button, and 'Drink Recommend' button.
#2. User clicks 'Drink Recommand' button.	#2. The system displays a list of drinks that meet the conditions you have selected.

Exceptional Scenarios :

- If there is no drink that meets all the conditions you have selected, nothing will pop up.
- If you select the checkbox and do not select the radio button next to it, no results will be displayed.
- If you select the checkbox, select the radio button next to it, and uncheck the box, the selection will not be reflected.

Example Queries

#1 one condition

- [price]
- **SELECT * FROM** drink **WHERE** drink.price >=1500 **AND** drink.price <= 5000
- [season]
- **SELECT * FROM** drink **JOIN** recipe **ON** recipe.drink_id = drink.drink_id **JOIN** ingredientslist **ON** recipe.ingredient_name = ingredientslist.ingredient_name
  WHERE season = 'Spring'
- [age]

- **SELECT * FROM** drink **INNER JOIN** orderedList **ON** orderedList.drink_id = drink.drink_id **INNER JOIN** userList **ON** orderedList.user_id = userList.user_id **WHERE** userList.age >= 10 **AND** userList.age < 20 **GROUP BY** orderedList.drink_id **ORDER BY** count DESC
- [gender]
- **SELECT * FROM** drink **JOIN** orderedList **ON** orderedList.drink_id = drink.drink_id **JOIN** userList **ON** orderedList.user_id = userList.user_id **WHERE** userList.gender = 'M' **GROUP BY** orderedList.drink_id **ORDER BY** count DESC
- [caffeine/decaffein]
- **SELECT * FROM** drink **JOIN** recipe **ON** recipe.drink_id = drink.drink_id **JOIN** ingredientslist **ON** recipe.ingredient_name = ingredientslist.ingredient_name **WHERE** recipe.ingredient_name = 'Espresso';

#2 two condition

- [season, age]
- **SELECT * FROM** drink **JOIN** recipe **ON** recipe.drink_id = drink.drink_id **JOIN** ingredientslist **ON** recipe.ingredient_name = ingredientslist.ingredient_name INNER JOIN orderedList **ON** orderedList.drink_id = drink.drink_id **INNER JOIN** userList ON orderedList.user_id = userList.user_id **WHERE** season = 'Fall' **AND** userList.age >= 10 **AND** userList.age < 20 **GROUP BY** orderedList.drink_id **ORDER BY** count **DESC**
- [season, gender]
- **SELECT * FROM** drink **JOIN** recipe **ON** recipe.drink_id = drink.drink_id **JOIN** ingredientslist **ON** recipe.ingredient_name = ingredientslist.ingredient_name JOIN orderedList **ON** orderedList.drink_id = drink.drink_id **JOIN** userList **ON** orderedList.user_id = userList.user_id **WHERE** season = 'Summer' **AND** userList.gender = 'M' **GROUP BY** orderedList.drink_id **ORDER BY** count **DESC**

#3 three condition

- [season, caffeine, age]
- **SELECT * FROM** drink **JOIN** recipe **ON r**ecipe.drink_id = drink.drink_id **JOIN** ingredientslist **ON** recipe.ingredient_name = ingredientslist.ingredient_name **INNER JOIN** orderedList **ON** orderedList.drink_id = drink.drink_id **INNER JOIN** userList ON orderedList.user_id = userList.user_id **WHERE** season = 'Summer' **AND** userList.age >= 20

AND userList.age < 30 **AND** recipe.ingredient_name = 'Espresso' **GROUP BY** ordere dList.drink_id **ORDER BY** count **DESC**

#4 four condition

- [season, age, gender, caffeine]
- **SELECT * FROM** drink JOIN recipe **ON** recipe.drink_id = drink.drink_id **JOIN** ingredi entslist **ON** recipe.ingredient_name = ingredientslist.ingredient_name **JOIN** orderedLi st **ON** orderedList.drink_id = drink.drink_id **JOIN** userList **ON** orderedList.user_id = userList.user_id **WHERE** season = 'Summer' **AND** userList.gender = 'F' **AND** userList. age >= 40 **AND** userList.age < **50 AND** recipe.ingredient_name = 'Espresso' **GROUP BY** orderedList.drink_id **ORDER BY** count **DESC**

#5 five condition

- [price, season, age, gender, caffeine]
- **SELECT * FROM** drink **JOIN** recipe **ON** recipe.drink_id = drink.drink_i **JOIN** ingredie ntslist **ON** recipe.ingredient_name = ingredientslist.ingredient_name JOIN orderedList **ON** orderedList.drink_id = drink.drink_id **JOIN** userList **ON** orderedList.user_id = us erList.user_id **WHERE** season = 'Winter' **AND** drink.price >=1500 **AND** drink.price < = 5000 **AND** userList.gender = 'F' **AND** userList.age >= 50 **AND** userList.age < 60 **A ND** recipe.ingredient_name = 'Espresso' **GROUP BY** orderedList.drink_id **ORDER BY** count **DESC**

Use Case: Pick drinks

Actor : User

Precondition: The system shows a beverage list.

Scenario:

Input Events from User	***System Events and Responses***
#1 User clicks a drink in the list #2 User clicks 'Pick' button	 #2. The System add the drink to user's pric ked list.

Exceptional scenario :

- If the user's selected drink is already in the list, it will not be added.
- If you do not select a drink and press the pick button, you will be prompted to selec t a drink.

Example Queries

#1

- **INSERT INTO** pickedList **VALUES** (15, 27);

3) STAGE 3 Choose Options

Use Case: User choose option by their own preference

Actor : User

Precondition: drink is already determined.

Scenario:

Input Events from User	***System Events and Responses***
#1. User chooses options without consideri ng their disease.	#1. After user selects the option, the system shows the final choice by combining the sel ected options.
#2. User chooses an option considering thei r own disease.	#2. When a user selects options, the system gives them a choice of options taking into account their disease. And then, it presents the final choice with the selected options.

Example Queries (Duplicated queries are not described.)

#1

- **SELECT** choose_id **FROM** chooseList **WHERE** option1='caramelSyrup';
- **SELECT** choose_id **FROM** chooseList **WHERE** option1='caramelSyrup' **AND** option2=' cinamonSyrup';
- **SELECT** choose_id **FROM** chooseList **WHERE** option1='caramelSyrup' **AND** option2=' cinamonSyrup' **AND** option3='hazelnusyrup';

#2

- **SELECT** option_name **FROM** optionList;
- **SELECT** disease_id **FROM** myDisease;
- **SELECT** option_name **FROM** optionCauseList **WHERE** disease_id=1;
- **SELECT** * **FROM** optionList **WHERE** option_name **<>** 'shot' **AND** option_name **<>** 'ci nnamonSyrup';
- **CREATE OR REPLACE VIEW** myDisease **AS SELECT** disease_id **FROM** sufferList **WHE RE** user_id=1;

Use Case: User choose option among the recommend list

Actor : User

Precondition: drink is already determined

Scenario:

Input Events from User	***System Events and Responses***
#1. User is recommended option list without considering their disease and choose one of them.	#1. Depending on the drink, the system shows up to five combinations of the most selected options so far. (At this scenario, the user's disease is not considered.)
#2. User is recommended option list considering their disease and choose one of them.	#2. Depending on the drink, the system shows up to five combinations of the most selected options so far, taking into account the user's disease.

Example Queries (Duplicated queries are not described.)

#1

- **SELECT** choose_id from **(SELECT** * **FROM** orderedList **ORDER BY** count **DESC)t WHERE** drink_id=4 **LIMIT** 5;

#2

- **SELECT** choose_id **FROM** chooseIdView;
- **SELECT** disease_id **FROM** myDisease;
- **SELECT** option_name **FROM** optionCauseList **WHERE** disease_id=1;
- **SELECT** choose_id **FROM** tempChooseListView **WHERE** option1<>'shot' **AND** option1<>'cinnamonSyrup' **AND** option2<>'shot' **AND** option2<>'cinnamonSyrup' **AND** option3<>'shot' **AND** option3<>'cinnamonSyrup'
- **SELECT** option1, option2, option3 **FROM** chooseList **WHERE** choose_id=3;
- **SELECT** drink_name **FROM** drink **WHERE** drink_id =3;
- **CREATE OR REPLACE VIEW** descOrderedListView **AS SELECT** * **FROM** orderedList **ORDER BY** count **DESC**;
- **CREATE OR REPLACE** chooseIdView **AS SELECT** choose_id **FROM** descOrderedListView **WHERE** drink_id=3;

- **CREATE OR REPLACE VIEW** myDisease **AS SELECT** disease_id **FROM** sufferList **WHERE** use r_id=1;
- **CREATE OR REPLACE VIEW** tempChooseListView **AS SELECT** choose_id, option1, option2, option3, **FROM** chooseList **WHERE** choose_id=1 **OR** choose_id=5;

4) STAGE 4 Order Drink

Use Case: Select option and Check disease and Order drink

Actor : User

Precondition: drink(name) is already determined

Scenario: If the user has both a disease filter and option selection

Input Events from User	***System Events and Responses***
#1. User selects the Temperature/Size corresponding to the previously selected drink. #2. If user wishes to know if the beverage ingredients are dangerous to his or her illness, selects the check box "Do you want to consider disease?". #3. User clicks the option selection button. #4. The user clicks the Order button to confirm the order information selected by the user. #5. The user who has confirmed all the information clicks the COUNT button.	#1. The drink_id is determined according to the user's selected Temperature / Size and drink_name. #2. The query finds the ingredients of the selected drink corresponding to the user's disease and displays a warning message if there is a match. #3. Call **STAGE 3 Choose Options** #4. Change the frame from setOrderPage to setOrderInfo Page. Display user selection options and beverage information on the frame. #5. If the user selects the drink ID and option id as the first choice, it sends the INSERT query. If it is the type that has already been selected, it sends the UPDATE query to save the user's choice.

Example Queries

#1

- **SELECT** drink_id, price, temparature, size **FROM** drink **WHERE** drink_name="EarlGrey Tea";

#2

- **CREATE OR REPLACE VIEW** myDisease **AS SELECT** disease_id **FROM** sufferList **WHERE** user_id="1"
- **SELECT** disease_id **FROM** myDisease
- **SELECT** Ingredient_name **FROM** causeList **WHERE** disease_id=1
- **SELECT** Ingredient_name **FROM** causeList **WHERE** disease_id=13
- **SELECT** Ingredient_name **FROM** causeList **WHERE** disease_id=14
- **SELECT** Ingredient_name **FROM** causeList **WHERE** disease_id=7
- **CREATE OR REPLACE VIEW** ingredientListView **AS SELECT** ingredient_name **FROM** recipe **WHERE** drink_id=10

#4.

- **SELECT IF(**"null" **IN (**option1**),** 0**,** option1**) as** option1, **IF(**"null" **IN (**option2**),** 0**,** option2**) as** option2**,IF("**null**" IN (**option3**),** 0**,** option3**) as** option3 **FROM** chooseList **WHERE** choose_id=4;
- **SELECT** ingredient_name **FROM** recipe Where drink_id=10;
- **SELECT SUM(IF(**"null" **IN (**option1**),** 0, 1**))+SUM(IF(**"null" **IN (**option2**),** 0**,** 1**))+SUM(IF(**"null" **IN (**option3**),** 0, 1**)) as** sum **FROM** chooseList **WHERE** choose_id=4;

#5

- **SELECT** count **FROM** orderedList **WHERE** user_id="1" **and** drink_id=10 **and** choose_id=4;
- **INSERT INTO** orderedList **VALUES**("1", 10, 4, 1); //First Choice
- **UPDATE** orderedList **SET** count=2 **WHERE** user_id="1" **and** drink_id=10 **and** choose_id=4; //Second Choice