

# **Examination – CS 300**

March 10, 2022

12:00 --- 1:50PM

## **DIRECTIONS**

1. This examination is open book and open notes.
2. Do all of your work in the exam packet.
3. The completed exam packet must be submitted as a Word or PDF file through Canvas by 2PM.
4. Begin by typing your name and last 4 digits of your PSU ID in the spaces below.

Name\_\_\_\_Juhwan Lee\_\_\_\_\_,    PSU ID (Last 4 digits) \_\_\_\_3246\_\_\_\_.

## Problem Statement

Suppose that you are being asked by your local food bank to create an on-line sign-up system for weekly food pick-up. You are given some initial requirements.

A customer goes to the food bank website and click the sign-up link. The sign-up page will collect the following information from the customer: Name, Phone, and Email, and allow the customer to pick an available time slot in the week to ensure that social distancing can be maintained for that time slot, and a food box will be ready when the customer shows up. Once the customer signs up, the sign-up is recorded in the food bank database and a confirmation is emailed to the customer including the time slot and the address of the food bank. The customer will then go to the food bank during the selected time slot to pick up.

**(Notes:** How the system is used by the food bank staff to access the sign-up records is beyond the scope of this exam.)

1. Please identify the WHAT, WHO, and WHY for this software product.

WHAT: food bank on-line sign-up system for weekly food pick-up

WHO: food bank customer

WHY: to ensure that social distancing can be maintained for that time slot, and a food box will be ready when the customer shows up

2. Please identify the personas and write the user stories for this software product.

**Persona**

Ted is a 41-year-old man living in poverty. Because Ted is struggling financially, he is unprotected in the current world of COVID-19. Social distancing is very important to Ted because if he gets COVID-19 he will not be able to get proper treatment. Fortunately, he has access to a computer and can make a living using the food bank.

**User story**

With the food bank online sign-up system, Ted can set a safe date for receiving food, and if he goes to the food bank on the specified date, there will be no crowds and he will be able to pick up his food while maintaining social distancing.

3. Please use the input/action/output template to describe the features of this software product.

Feature 1: Sign up

Input: Name, Phone, and Email, User ID, Passwords

Action: All the data goes to proper data socket

Output: Sign-up process complete, account stored in database, move to pick time slot page

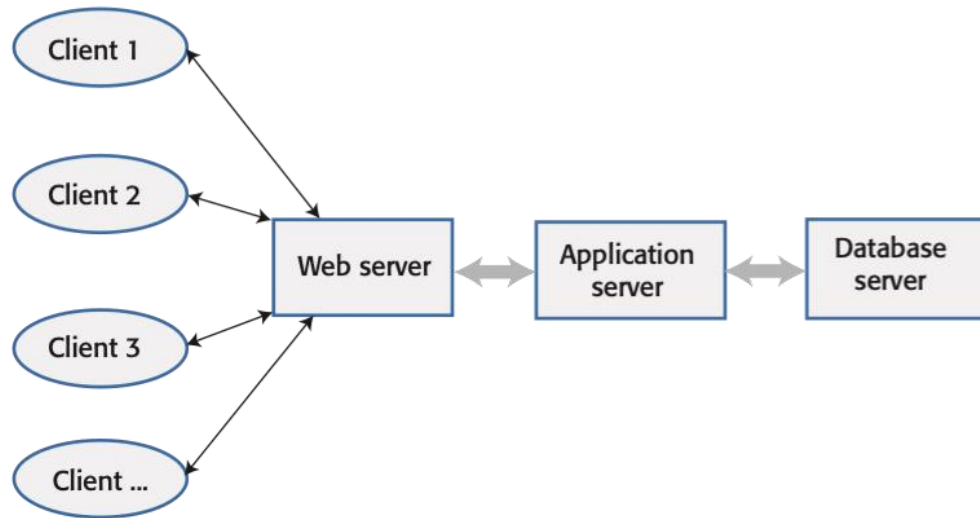
Feature 2: Pick time slot

Input: Click on the desired date

Action: Check if the date is available and if available, the date will be reserved

Output: Confirmation email will be sent, display “the food box will be ready on this day, please show up”, database will be reordered

4. Using a diagram, please show how this software product can be implemented using a multi-tier client-server architecture.



5. Please explain how you would deploy this software product on a cloud platform such as Amazon AWS, i.e., what would be the cloud software architecture for this product.

It is recommended that the food bank online sign-up system be deployed as a cloud base. This is because all users who use this application do not need to install the application and can use the service over the Internet through the web. Software as a service (SaaS) is the right type for this type of application, and the advantage is that it is easy to manage and can reduce development and maintenance costs. And by using microservices architecture, program readability can be improved, and complexity can be reduced.

6. Please design the backend of this software product following the micro-service architecture.

**Backend**

The backend design of the food bank online signup system should use a shared database. Shared Databases provide a synchronized storage location accessible by multiple concurrent users. The reason for using a shared database is that all food boxes are shared by all customers, and when one customer makes a reservation for a food box with a date, the result of this should influence other customers.

**Micro-service architecture**

User Interface → Reserve food box on certain date → Database



7. Please suggest, giving reasons, appropriate forms of security for this software product.

**Password based**

In the case of the food bank online signup system, password-based authentication would suffice the situation. Because it provides the product with basic security, no one else can access it without providing the required ID and password.

8. Instead of using a database system, please implement a simple data structure that can support the create, retrieve, update, and delete of the customer signup records. Provide both the interface and implementation of this data structure. (Note: Pseudo code is OK.)

```
#include <iostream>
#include <cctype>
#include <cstring>
using namespace std;

class signup
{
public:
    signup()
    {
        name = "";
        year = 0;
        month = 0;
        day = 0;
    }
    ~signup()
    {
        name = "";
        year = 0;
        month = 0;
        day = 0;
    }
    int create()
    {
        cin >> new_signup.name;
        cin.ignore(100, '\n');
        cin >> year;
        cin.ignore(100, '\n');
        cin >> month;
        cin.ignore(100, '\n');
        cin >> day;
        cin.ignore(100, '\n');
        return 1;
    }
}
```

```

int retrieve(signup &to_retrieve)
{
    to_retrieve.name = name;
    to_retrieve.year = year;
    to_retrieve.month = month;
    to_retrieve.day = day;
    return 1;
}

```

```

private:
    string name;
    int year;
    int month;
    int day;
};

```

```

struct node
{
    signup data;
    node *next;
};

```

```

class list
{
public:
    list()
    {
        head = NULL;
    }
    ~list()
    {
        if (!head)
        {
            return;
        }
        destroy(head);
    }
}

```

```

void destroy(node *&head)
{
    if (!head)
        return;
    destroy(head->next);
    delete head;
    head = NULL;
}

int update(signup &newdata)
{
    return update(head, newdata);
}

int update(node *&head, signup &newdata)
{
    if (!head)
    {
        head = new node;
        head->data = newdata;
        head->next = NULL;
        return 1;
    }
    return update(head->next, newdata);
}

int remove(string toremove)
{
    if (!head)
        return 0;
    return remove(head, toremove);
}

int remove(node *&head, string toremove)
{
    if (!head)
        return 0;
    if (data.name == toremove)
    {
        node *temp = head->next;
        delete head;
    }
}

```

```

        head = temp;
        return 1;
    }
    return remove(head->next, toremove);
}

private:
    node *head;
};

```

9. Please provide the unit tests for the above data structure.

```

int main()
{
    list signup_list;

    signup user_1;
    signup user_2;
    signup user_3;
    signup user_4;
}

```

```

if(user_1.create() == 1)
    cout << "create test 1 passed" << endl;
else
    cout << "create test 1 failed" << endl;
if(user_2.create() == 1)
    cout << "create test 2 passed" << endl;
else
    cout << "create test 2 failed" << endl;
if(user_3.create() == 1)
    cout << "create test 3 passed" << endl;
else
    cout << "create test 3 failed" << endl;
if(user_4.create() == 1)
    cout << "create test 4 passed" << endl;
else
    cout << "create test 4 failed" << endl;

user_1.retrieve(user_2);
user_3.retrieve(user_4);

if(user_1 == user_2)
    cout << "retrieve test 1 passed" << endl;
else
    cout << "retrieve test 1 failed" << endl;
if(user_3 == user_4)
    cout << "retrieve test 2 passed" << endl;
else
    cout << "retrieve test 2 failed" << endl;

if(signup_list.update(user_1) == 1)
    cout << "update test 1 passed" << endl;
else
    cout << "update test 1 failed" << endl;
if(signup_list.update(user_2) == 1)
    cout << "update test 2 passed" << endl;
else

```

```

        cout << "update test 2 failed" << endl;
    if(signup_list.update(user_3) == 1)
        cout << "update test 3 passed" << endl;
    else
        cout << "update test 3 failed" << endl;
    if(signup_list.update(user_4) == 1)
        cout << "update test 4 passed" << endl;
    else
        cout << "update test 4 failed" << endl;

    if(signup_list.remove(user_1) == 1)
        cout << "remove test 1 passed" << endl;
    else
        cout << "remove test 1 failed" << endl;
    if(signup_list.remove(user_2) == 1)
        cout << "remove test 2 passed" << endl;
    else
        cout << "remove test 2 failed" << endl;
    if(signup_list.remove(user_3) == 1)
        cout << "remove test 3 passed" << endl;
    else
        cout << "remove test 3 failed" << endl;
    if(signup_list.remove(user_4) == 1)
        cout << "remove test 4 passed" << endl;
    else
        cout << "remove test 4 failed" << endl;

    return 0;
}

```

10. Please provide the tests you will develop to test the features of this software product.

Test 1: Sign up test

Input: Name, Phone, and Email, User ID, Passwords

Expected Output: Sign-up process complete, account stored in database, move to pick time slot page

Test 2: Pick time slot test

Input: Click on the desired date

Expected Output: Confirmation email will be sent, display “the food box will be ready on this day, please show up”, database will be reordered

(This is the last page of the exam.)