

COMP4342

Group Project

Group19

Cheung Sui Wing
Lau Man Chun
Cheng Chi Kit
Kwong Chun Him

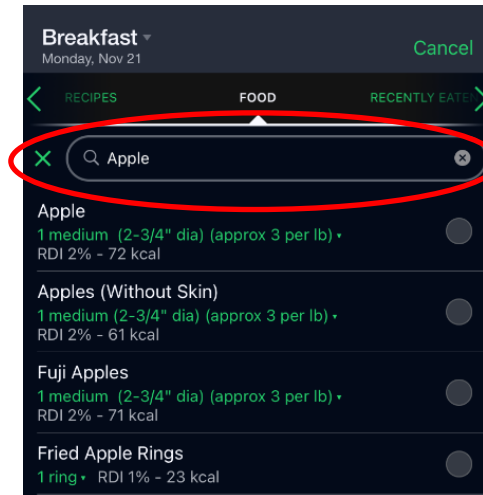
21027547D
21027257D
21028079D
21028468D



Overview

- Diet application
- For user to record daily calories absorbed
- Existing application example:

- User need to input the food one by one to search

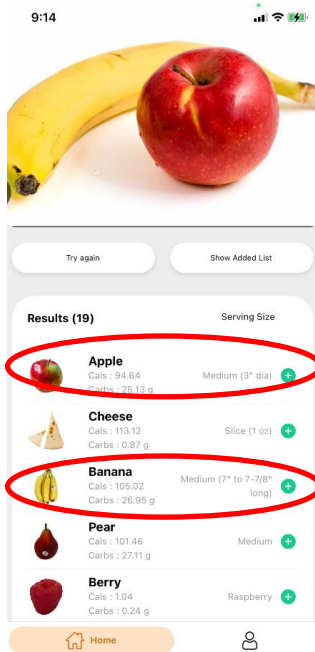


Market app: FatSecret



Overview

- Not convenient if you need to input so many food
- Our App:



← Take/pick a picture of the food

← Get the result by using food recognize



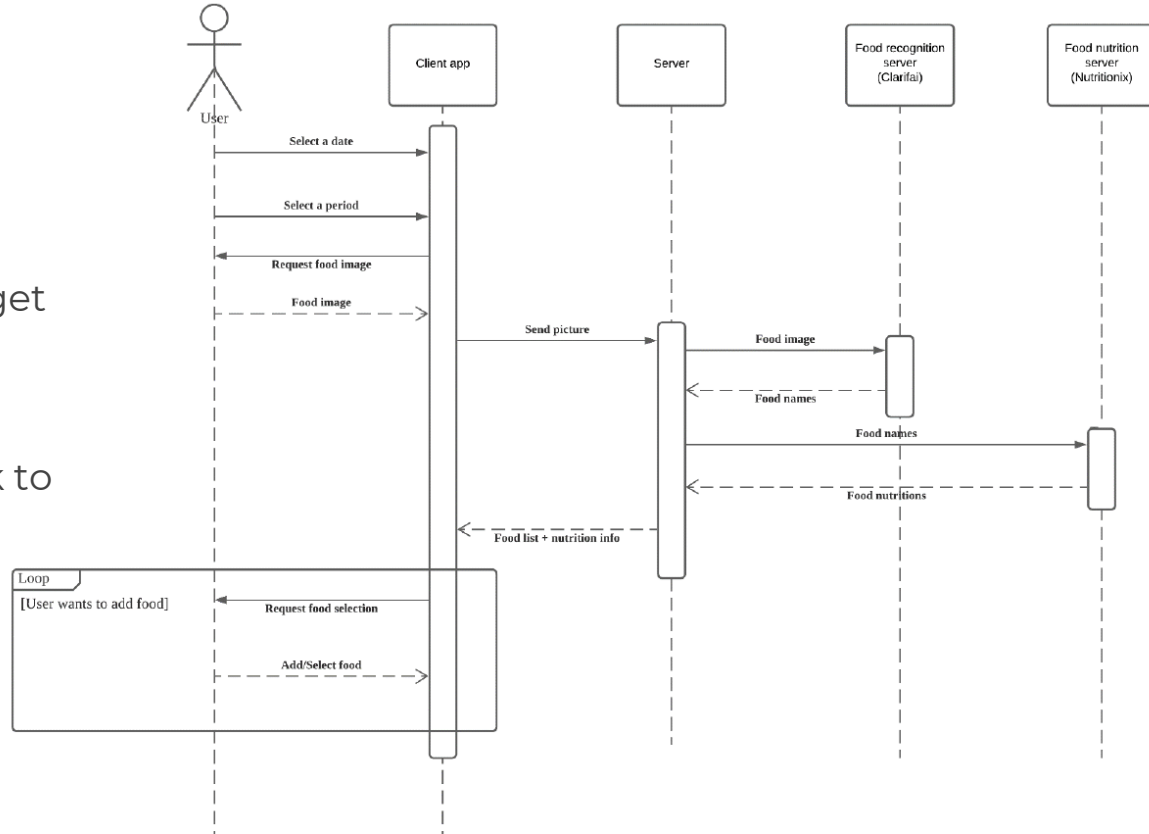
Application logic

- We **do not** train our model or collect food data
- Instead, we use two API:
 - Clarifai → provide free food recognize service
 - Nutritionix → provide nutrition data for food



Application logic (main function)

1. Send picture → backend server
2. Server send picture → Clarifai (get food recognize result)
3. Server send the food result → nutritionix (get nutrition data)
4. Reformat the data → send back to user



$$T_1 = \ell_1 + 273$$

$$333\text{K}, T_2 = t$$

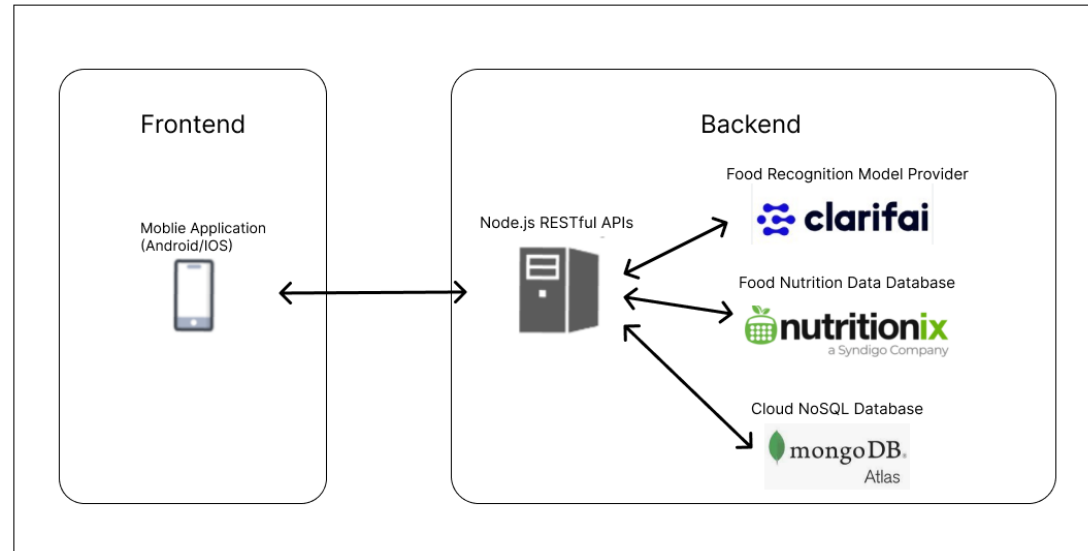
$$73 = 293\text{K}$$

$$x_0$$

$$x_0 + \Delta x$$

Software architecture

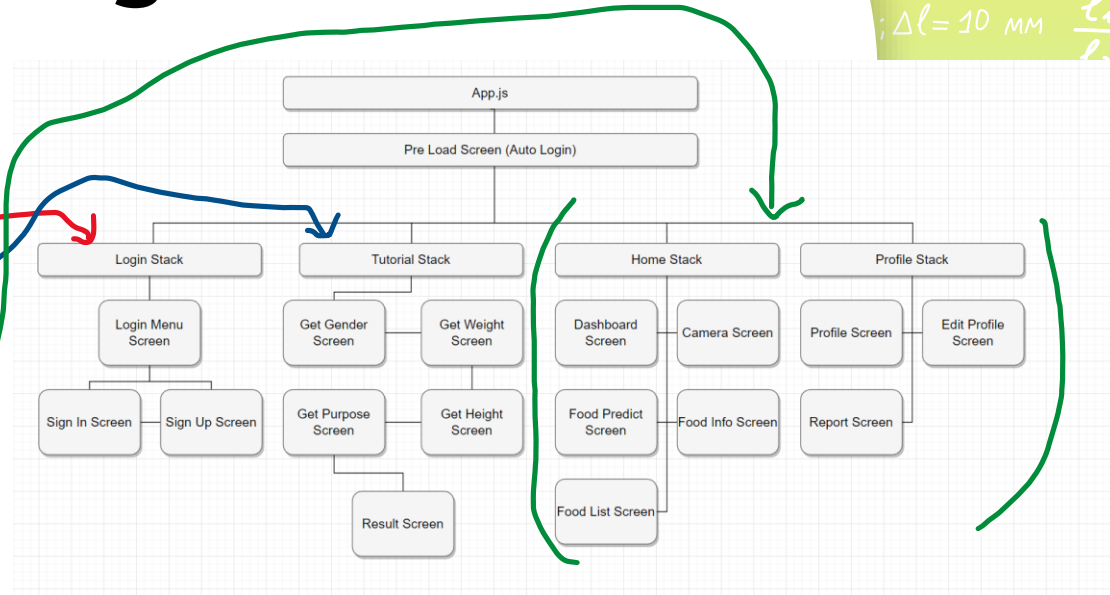
- Frontend → Mobile Application
- Backend → Restful server
 - With Clarifai API
 - Nutritionix API
 - mongoDB Atlas



Client-side design

- React Native expo
- (IOS 13.0+ / Android 5.0 API 21+)
 1. **Signin/signup**
 2. **If New Account, → Go Tutorial Screen**
 3. **If Not New Account, → Go Home Screen**
- Store the “user token” for auto login only
- Application-side encryption → hash the password before sending it to the server.

P.S Tutorial Screen is to lead the new user to input personal information to calculator BMR



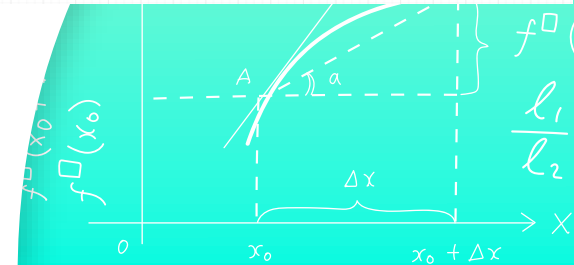
All Screen in our app

$$T_1 = t_1 + 273$$

$$333K, T_2 = t$$

$$73 = 293K$$

$$\Delta l = 10 \text{ mm} \quad \frac{l_1}{l_2}$$



Server-side design

- Node.js as the base
- Express.js → develop RESTFul API
- Mongoose.js → manage the connection of MongoDB atlas
- Java Web Token → protect our route

```
> router.post('/',upload.single('FoodImage'), async (req, res, next) =>{...
```

- 1 route for food recognition

```
/* create user
URL:localhost:3000/user/signup
Method: POST
body:
{
  "username": "test",
  "password": "12345"
}
*/
router.post('/signup', (req, res) =>{...
})

/* login
URL:localhost:3000/user/login
Method: POST
body:
{
  "username": "test",
  "password": "12345"
}
*/
router.post('/login', (req, res) =>{...
})

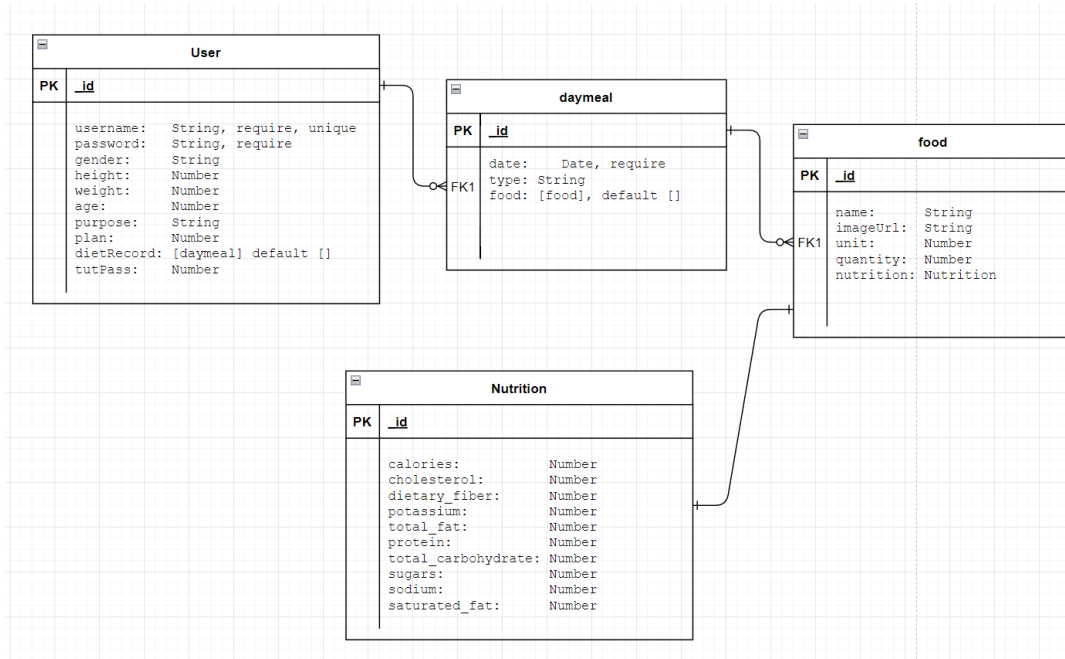
/* verify user by token
URL:localhost:3000/user/verify
Method: POST
body:
{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
*/
router.post('/verify',AuthToken, (req,res)=>{...
})

/* edit
URL:localhost:3000/user/edit
Method: POST
body:
{
  "user": "{
    _id : xxxxxxxxxxxxxx
    username: xxx
    xxx: xxx
    ...
  }",
  "token": xxxxxxxxxxxx
}
*/
router.post('/edit',AuthToken, (req,res)=>{...
})
```

- 4 route for user (Signin/ Signup/ verify/ edit)

Database

- Using MongoDB Atlas
- Data model:



Database

- How the data look:

```
_id: ObjectId('63773a83a7f43bbb19168aef')
username: "ArthasKeW"
password: "$2b$10$IiwSb0Y/KBatxVVLtGBQuHtrHSH9VeLnZ9mkqkDammlvgTlSNs26"
gender: "Male"
height: 170
weight: 75
age: 22
purpose: "Increase"
plan: 1
tutPass: 1
✓ dietRecord: Array
  ✓ 0: Object
    date: 2022-11-18T00:00:00.000+00:00
    type: "breakfast"
    ✓ food: Array
      ✓ 0: Object
        name: "rice"
        imageUrl: "https://nix-tag-images.s3.amazonaws.com/784_thumb.jpg"
        unit: "cup"
        quantity: 1
        > nutrition: Object
          _id: null
        > 1: Object
          _id: ObjectId('63773adca7f43bbb19168af4')
      > 1: Object
      > 2: Object
      > 3: Object
      > 4: Object
      > 5: Object
      > 6: Object
      > 7: Object
      > 8: Object
      > 9: Object
      > 10: Object
      > 11: Object
      > 12: Object
    __v: 0
```

$$T_1 = t_1 + 273$$

$$333K, T_2 = t_2 + 273$$

$$73 = 293K$$

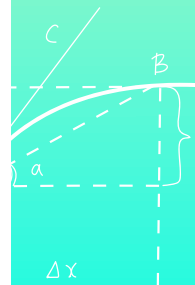
$$\Delta l = 10 \text{ mm} \quad \frac{l_1}{l_2}$$

$$= 0.036, \epsilon_1 = 3,6$$



$$tg$$
$$y$$

$$\Delta l_n + \Delta l_0$$



$$y =$$
$$f(x)$$

$$\frac{l_1}{l_2}$$

0

x_0

$x_0 + \Delta x$

Demo + Q&A

