

건강을 위한 식단관리 애플리케이션 Wealthy Health

허세린(19017093)

수원대학교 ICT융합대학 컴퓨터학부 컴퓨터sw학과

목차

1. 서론

2. 시스템 설계 및 구현

2-1. 설계 단계 선정

2-2. 각 단계별 세부사항

2-2-1. analysis

2-2-2. showinfo

2-2-3. recipe

2-3. 예상 User Interface

3. 결론

요약

집 밖에서의 활동이 줄어든 이 시대에 맞는 건강관리 애플리케이션을 제작하고자 하였으며, 애플리케이션 속에 들어갈 프로그램들을 구현하였다.

이 프로그램을 통해 코로나 시대의 건강관리에 관한 사용자들의 수요를 충족시켜 줄 수 있을 것으로 예상된다.

1. 서론

건강한 삶을 위해 많은 사람들은 다양한 노력을 하며 살아간다. 정기적으로 건강검진을 받고, 헬스장에 가 운동을 하고 식단을 관리하기도 한다. 코로나가 장기화 되면서 이러한 건강관리를 집에서도 이어나가고자 하는 사람들이 많다는 것을 알 수 있었고, 활동량이 줄어들다 보니 자연스럽게 집에서 관리할 수 있는 방안들에 관심이 높아졌다는 것 또한 알 수 있었다. 건강관리에 도움이 되고자 하는 애플리케이션들이 시중에 많이 상용화되어 있는데, 이 애플리케이션들을 보면 식단을 기록하는 일종의 '일기'의 용도로 쓰이는 것들이 대부분이었다. 건강한 식단을 잘 짤 수 있는 경험과 지식이 있는 사람들에게는 이것으로 충분하겠지만, 그렇지 않은 사람들이 대부분이므로 기록 기능을 가진 애플리케이션만으로는 부족하다고 생각하였다.

이에 따라, 개인에게 맞는 식단을 직접 제공해주는 기능이 있으면 사람들이 더욱 편리하게 이용할 수 있을 것 같았기 때문에 식단 기록 애플리케이션에서 더 나아가 식단을 짜주고, 근본적인 건강을 관리해주는 애플리케이션을 구상하게 되었다. 또한, 이를 응용하여 일상의 식단을 변화시켜 호전시키거나 예방할 수 있는 병들에 대한 정보와 식단을 제공한다면 유저들의 요구를 더욱 명확히 충족시킬 수 있고 활용성도 더 다양해질 것이라고 예상한다.

이번 프로젝트를 통해서는 질병을 진단할 수 있고 간단한 정보를 제공하며 이에 따른 식단과 조리법을 제공하는 것 까지 개발 가능하였다. 개발 가능 부분을 넘어서는 내용이지만, 병원에

서 하는 건강검진 결과를 연동시키거나 유저들이 더 전문적인 건강수치 관련 질문들에 답하면서 병을 진단할 수 있고, 이에 따라 적합한 병원을 거리 순으로 추천해주는 것 또한 유익할 것이라고 생각한다. 병원 기록과 연동시킨다면 환자(유저) 측에서는 계속해서 같은 병원에서 지속적으로 관리 받을 수 있다는 장점이 있고, 제공 받은 식단 및 조리법으로 큰 힘을 들이지 않고 하나의 애플리케이션만으로도 한 끼 식사 준비가 가능하다는 장점이 있다. 병원 측에서는 한 번 유치한 고객(환자)를 지속적으로 진단 및 진료가 가능하기 때문에 병원에 이득이 될 것으로 예상되고, 환자들이 집에서도 혼자 관리할 수 있는 능력이 향상되므로 환자 관리에 용이하다는 장점을 가지게 된다.

이러한 바람을 담아 기획하게 된 이 애플리케이션의 명칭은 Wealthy Health 이고, 이는 '부유한 건강'이라는 의미이다. 우리의 건강이 조금 더 풍요롭고 잘 관리될 것이라는 의미에서 'wealthy' 라는 명칭을 사용하게 되었다.

2. 시스템 설계 및 구현

2.1 설계단계 선정

프로그램을 구현하기에 앞서, 체계적으로 필요한 부분을 빠짐없이 작성하기 위해 부분적으로 나누어 진행하였다. 크게 세 파트로 나누었는데 analysis, showinfo, recipe로 구성되어있다. analysis 라는 부분은 또, 두 개의 파일로 나뉘

어 정보를 수집하는 부분과 수집한 정보를 바탕으로 진단 및 보기 편리하도록 정리하는 부분으로 구성되어 있다. showinfo는 정보를 제공하기 위한 부분이며, recipe는 총 세부분으로 구성되어 있는데 식단을 짜기 위한 부분이다.

이렇게 전체 프로그램을 파트별로 나누어 진행하였는데, 코드 작성 순서 또한 출력될 때의 순서대로 analysis단계부터 진행하였다.

2.2각 단계별 세부사항

2.2.1 analysis

건강검진 항목에 들어가는 항목을 제일 먼저 생각하였다. 제일 기본적인 이름부터 시작하여 신장과 몸무게, 혈압 수치와 혈당량 등의 정보를 얻기 위해 변수를 지정해 놓고 사용자 입력값으로 받을 수 있도록 설계하였다. 사용자들이 정보를 입력한 뒤에는 그 정보를 기반으로 몇 가지 발생가능한 병들을 진단해 보았다. 혈압 수치가 140을 초과하면 고혈압, 90 미만이면 저혈압으로 측정되며 혈당량이 120 이 넘으면 당뇨로 진단되게끔 설계하였다. 또한, 저체중과 과체중을 진단할 수 있도록 bmi 식을 사용하여 신장과 몸무게를 이용하여 비만도를 확인할 수 있게끔 하였다. me라는 리스트를 미리 정의해 두고, 병들이 진단되면 그 리스트에 추가하고 출력을 해주어 사용자가 자신의 건강상태를 확인할 수 있도록 하였다.

```
myname=""
myage=0
myheight=0
myweight=0
me=[]
myblood_pressure=0
myblood_sugar=0

myname = input("성함을 입력해주세요 : ")
print("안녕하세요~", myname, "님", "WnWealthy Health 입니다!!")

high_blood_pressure = None
low_blood_pressure = None
diabete = None

myblood_pressure = int(input("혈압수치를 입력해주세요 : "))
myblood_sugar = int(input("혈당수치를 입력해주세요 : "))

high_blood_pressure = myblood_pressure > 140
low_blood_pressure = myblood_pressure < 90
diabete = myblood_sugar > 120

myheight = int(input("신장을 입력해주세요 : "))
myweight = int(input("체중을 입력해주세요 : "))
height = myheight/100
mybmi = myweight/(height*height)
if (mybmi <= 18.5) :
    print("저체중입니다.")
    me.append("저체중")

if (mybmi > 18.5 and mybmi <= 24.9) :
    print("정상체중입니다.")

if (mybmi > 24.9 and mybmi <=29.9) :
    print("과체중입니다.")
    me.append("과체중")
```

▲analysis 의 일부분

2.2.2 showinfo

analysis 부분에서 리스트에 입력된 병명 중 그 병에 대해 확인하고 싶은 부분이 있다면 사용자가 선택하여 정보를 확인할 수 있도록 하였다. while (True) 문을 사용하여 반복하면서 자신의 리스트에 속해있는 병명이 아니더라도 확인하고 싶다면 확인할 수 있도록 작성하였다. 사용자가 확인하고 싶은 병명에 번호를 붙여 while문 속의 if문에서 분류하였고 하나만 확인하고 끝나지 않을 수 있으므로 continue로 계속 이어지게끔 하였고, 잘못 입력될 수 있는 부분 또한 고려하여 작성하였다.

2.2.3 recipe

이 부분이 Wealthy Health 프로그램을 작성하는 데 있어 가장 중요하고 핵심적인 부분이라고 할 수 있다. 입력받은 정보를 이용하여 직접적으로 식단은 물론 그 조리법까지 제공해주는 부분이다.

처음 이 부분을 작성하려 했을 때, 식재료들을 어떻게 분류하여 출력할 것인지에 대해 많은 고민의 시간을 거쳤다. 처음 기획했던 의도는 식재료들을 영양성분, 도움이 되는 병명, 색깔을 기준으로 하여 분류하고자 하였으나 구현하는 데에 한계가 있음을 느껴, 원재료가 아닌 완성된 하나의 요리를 제시하는 것으로 변경하였다. 이렇게 하여 조리법을 제공하기에도 훨씬 수월하고 복잡하지 않은 코드를 사용하여 원하는 출력 값을 만들어낼 수 있었다.

recipe 파트에서 첫 부분은 병명마다 한 끼 식사에 들어갈 수 있는 요리들을 모아 분류해 놓은 부분이다. 탄수화물 부분, 단백질 부분, 비타민 등의 명칭을 이용하여 각 병명마다 적게는 1개 많게는 4개의 리스트를 생성하였다.

```
list_highbptan = ["현미밥", "검은콩밥", "조밥"]  
list_highbpdan = ["생선", "삶은계란", "두부구이", "가지구이", "표고버섯볶음"]  
list_highbpgood = ["양파", "굴", "인삼차", "생강차"]
```

▲recipe-01의 일부분

recipe파트의 두 번째 부분은 이 리스트에 있는 요리들 중 1~2개를 출력해내는 것이다. 이미 영양소별로 분류가 되어 있는 상태이기 때문에 랜덤하게 출력하는 코드를 작성하였다. 이 부분도 앞서 작성했던 것과 동일하게 while문을 이용하였고, 원한다면 모든 부분의 식단을 제공

받을 수 있도록 하였다. 원하는 식단(예를 들어 과체중용 식단, 고혈압용 식당 등)을 번호로 선택하면 그에 맞는 리스트가 랜덤적으로 생성된 후, 출력되고 continue되어 원할 때까지 출력해볼 수 있게끔 구성하였다.

```
if select2 == 2 :  
    print("\n포만감을 주는 음식들과 열량이 낮은 음식들을 선택해야 합니다.")  
    sampleList5 = random.sample(list_overweighttan, 1)  
    sampleList6 = random.sample(list_overweightdan, 1)  
    sampleList7 = random.sample(list_overweightfib, 1)  
    print(sampleList5)  
    recipe.append(sampleList5)  
    print(sampleList6)  
    recipe.append(sampleList6)  
    print(sampleList7)  
    recipe.append(sampleList7)  
    continue
```

▲recipe-02의 일부분

마지막 부분은 조리법을 제공하는 부분이다. 조리법은 인터넷에 검색해본 것과 경험에서 나온 조리법들을 제공하였고, 기본적인 밥 종류에 대해서는 조리법을 제공하지 않았다.

```
print(recipe)  
print("조리법을 확인해주세요.")  
while True :  
    print("\n조리법을 보고자 하는 요리의 번호를 선택해주세요.\n")  
    select3 = int(input("1.고등어구이\n2.양념갈비\n3.계란말이\n4.가지무침\n5.오이무침"))  
  
    if select3 == 1 :  
        print("<고등어구이>\n")  
        print("1.물기를 최대한 제거해주세요.\n")  
        print("2.밀가루나 부침가루를 묻혀 가루는 살살 털어내고 준비해주세요.\n")  
        print("3.기름을 넉넉하게 두른 팬에서 세볼로 1차 튀기듯 구워주세요.\n")  
        print("4.두번째 튀기는 정도로 졸내야 살이 망가지지 않고 좋아요.\n")  
        print("5.가장자리가 노릇노릇할때 뒤집으면 됩니다.\n")  
        continue
```

▲recipe-03의 일부분

2-3.예상 User Interface

이 프로그램을 만약 애플리케이션으로 구현 가능하다면 다음과 같이 UI를 만들고 싶다.

▲정보를 수집하는 부분의 창

이러한 테마를 이용하여 showinfo 부분, recipe 부분을 간단하게 디자인해보았다.

▲showinfo 부분

▲recipe 부분

3.결론

본 프로젝트에서는 평소 느끼던 생활 속 불편함을 해결하기 위해 건강관련 프로그램을 제작하였다. 정보를 수집하고 제공하며, 식단을 제공하는 것으로 구성되어 있으며, 현대의 언택트 시대에서의 다이어트나 건강관리에 효과적일 것이다. 병원에서 받는 건강검진 결과나 인바디 측정 결과 등을 연동하는 기술을 접목한다면 사업성을 더욱 키울 수 있을 것으로 예상된다.