**Feasibility Study: Foodinfo**

**Requirements**

Operational:

* The system’s content provider part should be able to work on any Web browser with minimum of 1366x768 pixel resolution.
* The system’s end-user part should be able to work on Android devices which has minimum 4.2 Android version and higher.
* The system’s web service part should be able to work with json format.

Performance:

* The system should be available for use 24 hours per day, 365 days per year.
* The system must be scalable in the future for meet the performance needed.

Security:

* All users must access the system with their own password and login information.
* The passwords of the users must be hashed before saving database.
* Content providers must need an authorization from the administration for log in to the system.

Quality assurance:

* The system should be able to show and store the picture of the product minimum 250x250 pixels.
* The system should be able to show and store the contents of the product, barcode number of the product, explanation and details of the product, storage condition of the product, using instructions of the product and energy, sugar, fat rate, saturated fat rate and salt etc. health-related information.
* The system on the android should be able to read the barcode on the product using the integrated camera on the phone.
* The system on the android must open the related product page when reads the barcode.

Cultural and Political:

* The system on the android must provide 3 system language as Turkish, English and Polish.
* The system on the android should be able to read user’s device language on first start of the system and if the language exist it must work in that language.
* The system on the android should be able to provide user to change system language.
* The system on the android must provide scalable language list for user on settings menu and chosen language must be saved as default product detail language.
* On first system start, the system should be able to provide a default product detail language list for user.

**Identifying Components:**

**Android Studio**

+ A lot of source code , provides android emulator , perfect code completion, appearance, everthing settled and less bug

- System Requirements

**Rad Studio**

+ Gives you alternative coding language (C++)

- New and has a lot of bugs , less source code

**Xamarin**

+ Gives you alternative coding language (C#) , growing fast , product of Microsoft

- New and has a lot of bugs , less source code

**Unity**

+ A lot of graphical utilities

- Main purpose of unity is Game Development

**Eclipse**

- code completion , UI , it started to lose users to other alternatives

+ It's Initial release in 2001

**Risks:**

**1- ) Inherent Schedule Flaw**

We have a specific time for each project. Therefore, the planning of the time should be done well. There are some methods for avoiding Inherent Schedule Flaw - Little's law, Monte Carlo method etc.

### 2- ) **Requirement inflation**

### The Pareto princeple (also known as the80/20 rule)

### if we implement pareto princeple for software, we can say that used properties is about % 20. So we should concentrate MVP (minimum viable product**)**

### **3- ) Employee/Students turnover**

### It is impossible to avoid this risk.Even for Google, leaving employees is one of the biggest problems. To avoid this risk, it is necessary to reduce the collection of information in one person as much as possible.

### We need to create an information repository and give information to each other.

### **4- ) Specification breakdown**

If this happens, the project ends and is therefore very important.

F.e. We are writing the program first as Android in Android Studio. But there is uncertainty about ios. There are environments where you can write programs for both android and ios at the same time. F.e. If you are using unity, you can get output for both android and ios. But in this case, the performance may experience difficulties.

According to my previous experience, if the same program is written in android studio and unity, apk of unity takes up a lot more space. Also, unity is a game engine. The android studio has much more documentation, open source code, and tutorial. If you want to write an android application, android studio is the most recommended environment. Other side, there is noone who developed native ios application in group.

**The uncertainty about how to write ios is the greatest risk.**

### **5- ) Poor productivity**

### Good communication is essential to avoid this problem. Also we always should check our business planning.

**Competitors:**

**Codecheck:**

**-** Has a Barcode-Scanner and shows ingredients, but is rather feasible for diet plans.

- Our Application is supposed to show all ingredients and make it possible to read them in any

language.

- Our Application is supposed to be more open to make it possible to widen the amount of

information.

**Open Food Facts:**

- Has a Barcode-Scanner and shows ingredients.

- No Language option

- Our Application is supposed to be more open to make it possible to widen the amount of

information.