**RoutIne TrackIng System**

**İsmail Tuna Seydişehirli, 20200108006**

**Emirhan Yılmaz, 20200108052**

**Bilge Kaan Altuntaş, 20200108039**

**Enes Mirza Kaşıkçı, 20200108046**

BIP 2012 Bitirme Projesi Final Raporu

Danışman:

Emrah Sarıçiçek



Piri Reis Üniversitesi, Denizcilik Meslek Yüksekokulu, Bilgisayar Teknolojileri Bölümü, Bilgisayar Programcılığı Programı

[2023]

# ÖZ

Proje temel olarak bilgisayarda kullanılan uygulamaları gözlemleyip bunların türüne göre veritabanına kaydedip telefon uygulaması aracılığıyla paylaşabilen, uyarı ve gözlem uygulaması. Projenin amacı uygulama kullanımını gözetleyip kaydetmek ve sağlıksız olabilecek davranışlar farkedildiğinde uyarı mesajı göndermek, aynı zamanda kullanıcılara kendi kullanım alışkanlıklarını gözlemleme fırsatı tanıyarak bir kullanıcı bilinci oluşturmak. Proje öncelikle bilgisayar uygulaması ile sistemde çalışan ana uygulamaları okur ve bunları kullanıcının bilgisayar üzerinde efektif bir şekilde izleyip kontrol edebilmesini sağlar.

# İÇERİK

[ÖZ ii](#_Toc67560619)

[İÇERİK iii](#_Toc67560620)

[**1.** **GİRİŞ** 4](#_Toc67560623)

[**1.1** **Problemin Tanımı** 4](#_Toc67560624)

[**1.2** **Alternatif Çözümler** 4](#_Toc67560625)

[**1.2.1** **Çözümlerin Açıklanması** 4](#_Toc67560626)

[**1.2.2** **Olabilirlik Analizi (Feasibility Analysis)** 5](#_Toc67560627)

[**1.2.3** **Değerlendirme ve Karar** 5](#_Toc67560628)

[**2.** **SİSTEM ANALİZİ** 5](#_Toc67560629)

[**2.1** **Projenin Tanımı ve Kapsamı** 6](#_Toc67560630)

[Aktörler, context diyagramı 6](#_Toc67560631)

[**2.2** **Gereksinim Analizi** 6](#_Toc67560632)

[**2.2.1** **İşlevsel Gereksinimler** 6](#_Toc67560633)

[**2.2.2** **İşlevsel Olmayan Gereksinimler** 7](#_Toc67560634)

[**2.3** **Use Case Analizi** 7](#_Toc67560635)

[Use case diyagramları 7](#_Toc67560636)

[**2.4** **Süreç Analizi (Process Modelling)** 8](#_Toc67560637)

[**3.** **SİSTEM TASARIMI** 8](#_Toc67560638)

[**3.1** **Sistem Mimarisi Tasarımı (Architectural Design)** 8](#_Toc67560639)

[Yazılım ve donanım mimarisi 8](#_Toc67560640)

[**3.2** **Kullanıcı Arayüzü Tasarımı (User Interface Design)** 8](#_Toc67560641)

[**3.3** **Program Tasarımı** 9](#_Toc67560642)

[**3.4** **Veri Modelleme ve Veritabanı Tasarımı** 9](#_Toc67560643)

[**4.** **KODLAMA** 9](#_Toc67560644)

[**5.** **TEST** 10](#_Toc67560645)

[**5.1** **Test Senaryosunun Belirlenmesi** 10](#_Toc67560646)

[**5.2** **Birim Testler** 11](#_Toc67560647)

[**5.3** **Bütünleşik Testler** 11](#_Toc67560648)

[**5.4** **Sistem Testi** 11](#_Toc67560649)

[**5.5** **Kullanıcı Testleri** 1](#_Toc67560650)1

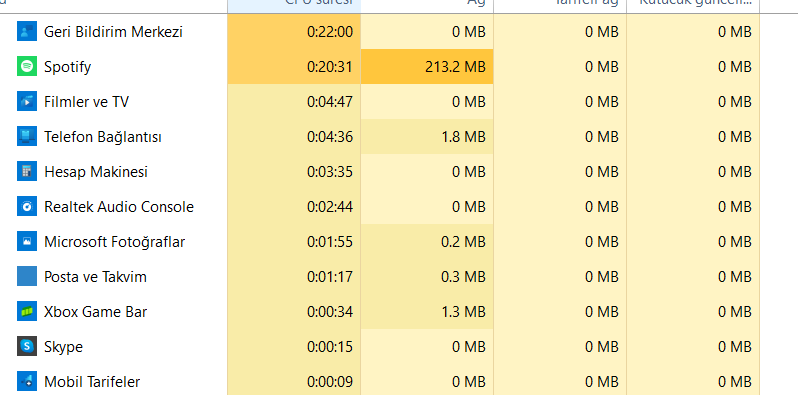
[**6.** **UYGULAMAYA HAZIRLIK VE UYGULAMAYA KOYMA** 12](#_Toc67560651)

1. **GİRİŞ**
   1. **Problemin Tanımı**

Projenin ele aldığı problem, bilgisayar ve uygulamalarının denetimsiz kullanılıp sağlıksız bağımlılıklara dönüşmesi ve kullanıcıların işlem geçmişini inceleyemediğinden dolayı bilinç kazanamaması.

* 1. **Alternatif Çözümler**

**-**Halihazırda işletim sistemlerinde bulunan görev yöneticilerinde sayaçların görünür, daha doğru ve ulaşılabilir olması.



1.2.a: genel olarak saatlerce çalışan bir bilgisayarda eksik veri olması…

-Uygulamalarda, kullanım uyarıları veya sınır ayarları olması.

-Genel kullanıcı eğitimleri verilmesi.

* + 1. **Çözümlerin Açıklanması**

-Uygulama ve işletim sistemi taraflı çözümlerin şirketler ve geliştiriciler için market veya kâr kaybetmeye sebep olabilir. Bu yüzden yapılmaması veya üstüne gidilmemesi kendilerince anlaşılabilir.

-Genel kullanıcı eğitimi tarafında ise toplumun eğitimlere karşı tepkisi veya eğitimlere katılımı değişken olup kesin sonuçlar alınmasını engeller.

* + 1. **Olabilirlik Analizi (Feasibility Analysis)**

-Projenin teknik açıdan değerlendirilmesi yapıldığında:

Kullanılan metodlara ve arayüzlere aşina olunduğundan dolayı risklerin çoğu giderilmiştir. Projemiz, orta büyüklükte bir proje olup 4 kişi tarafından çalışıldığı için temel olarak büyük bir risk taşımıyor. Kullandığımız donanımsal ve yazılımsal kaynaklar yeterli olduğundan dolayı teknik bir risk taşımıyor.

-Projenin ekonomik açıdan değerlendirilmesi yapıldığında:

Proje yapım aşamasında temel olarak bir harcama çıkartmasa, projenin aktif çalışabilmesi için gerekli olan ortalama fiyat yıllık 42$ ya da 840TL. Proje, başlangıçta bağış veya çok ufak bir ücret karşılığı satılacak olsa bile 1 yıl içerisinde kâra geçmeye hazırdır. Uzun vadede zarara geçmeyeceği planlanıyor. Market üzerinde benzer bir proje sık karşılaşılmadığı için “marketanalizi” yapmak odukça zor fakat negatif bir çıkarımda bulunulmadı.

-Projenin operasyonel açıdan değerlendirilmesi yapıldığında:

Proje, işleyişi anlamında hizmet verilecek kitle spesifik ama küçük olmayan bir topluluktur. Projenin öğrenim ve tam kullanıma geçiş süreci, yapısından dolayı çok kısa ve özdür. Kullanacak olan kitlenin herhangi bir alanda sıkıntı çektiğinde ulaşabilecekleri kişiler uygulamanın arayüzünde değişmez bir şekilde bulunacaktır. Farklı seviyelerde teknoloji deneyimi olan insanlar için kolay ulaşılabilir bir yapıda olacaktır.

-Projenin zaman açısından değerlendirilmesi yapıldığında:

Proje, dört kişilik bir grup tarafından, aksaklıklar göz önüne alınınca, ortalama 4 ayda bitmesi planlanıyor. Proje için gerekli olan süre içerisinde bitmeyip aksaması riski aşırı derecede küçüktür. Grubun plan aşamasındaki görev dağılımı ile bu riskler minimal düzeye indirilmiştir.

* + 1. **Değerlendirme ve Karar**

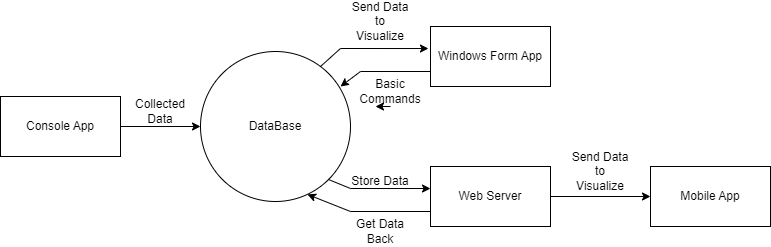
Proje “Olabilirlik Analizi” tamalandığında, seçtiğimiz çözümün uygun olduğunu ve taşıdığı risklerin getirileri ile karşılaştırıldığında kabul edilebilir olduğu görülüyor. Sonuç olarak, proje taşıdığı kâr şansı ve yapımı sırasında kazandırdığı bilgi ve deneyim açısından bakılınca en ideal tercihtir.

1. **SİSTEMANALİZİ**
   1. **Projenin Tanımı ve Kapsamı**

* Proje temel olarak üç alt parçadan oluşur, bunlar; Konsol Uygulaması, Windows Form Uygulaması, Veri Tabanıdır, ileride güncellemeler ile birlikte Web Sunucusu ve Mobil Uygulama da aralarına girecektir.
* Konsol Uygulaması temel olarak sistemde görsel olarak daha az yer kaplarken, asıl amacı sisteme daha az kaynak yükü bindirmek ve proje sisteminin bütününün sürekli açık kalmasını engellemek. Farklı kütüphaneler ve temel Windows işlevi olan “tskschdl” yani görev zamanlayacı ile tam efekt gösterecektir.
* Windows Form Uygulaması bize Konsol Uygulamasından verileri görselleştirerek gözü yoran sayısal verilerden, göze hitap eden modellere dönüştürmeyi amaçlar. Diğer bir işlevi ise eskiden depolanan verilerin tekrar gösterimini sağlamak, ileride gelen güncellemeler ile gönderme işlemi de yapıcaktır.
* Veri Tabanı Konsoldan toplanan verileri başlıklar altında düzenli şekilde depolar ve hem Windows Form hem de Mobil Uygulamalarda kullanılması için hazırda tutar. Çıktı formatı sayesinde iki uygulama tarafından da okunabilir.

Ayrıca ilerideki eklencek parçalara bakıldığında:

* Web Sunucusu, “Rest Api” mimarisi ile yalnızca veri taşıma işlemi yapıp görsel herhangi amacı bulunmamakta, ilerleyen zamanlarda komut iletisi de sağlamayı planlıyoruz...
* Mobil Uygulama, Web Sunucusundan çektiği verileri okuyup kullanıcıya bilgisayarda yaptığı eylemleri görme ve denetleme şansı tanıyor, işlevsel olarak yalnızca Veri Tabanından gelen verileri okuyup görselleştirmek amacında olsa da ileride küçük çaplı uzaktan kontrol de sağlanmaya çalışılacak…

2.1.a Context Diagram:

* 1. **Gereksinim Analizi**
     1. **İşlevsel Gereksinimler**

-Konsol Uygulamasının daimi olarak çalışıyor, her 15 dakikada bir sistem üzerindeki verileri okuyor ve Veri Tabanına kaydediyor olmalı.

-Günlük olarak, belirlenen saatte Veri Tabanındaki son işlem tabloları Web Sunucusuna aktarılmalı.

-Bilgisayar arayüzü kişilerce istenen boyutta ayarlanabilmeli.

* + 1. **İşlevsel Olmayan Gereksinimler**

-Projenin kullanılabilmsei için Windows OS kullanılması gerekmekte.

-Proje, bilgisayar açık oldukça işlevsel olarak çalışmalı.

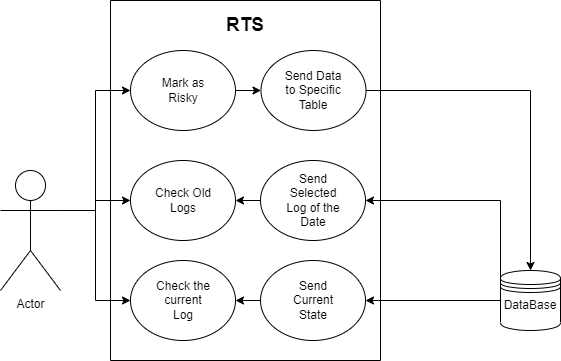
-Kayıt uygulaması çıktıları beklenen yere depoluyor ve “.db” programına erişebiliyor olması

-Sistem işlemleri Bilgisayar tarafından engellenmemeli.

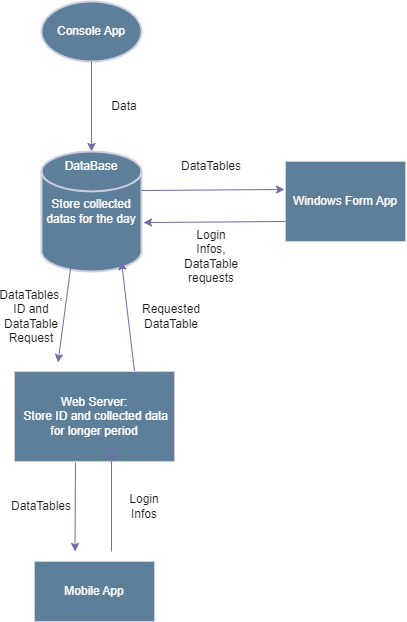
* 1. **Use Case Analizi**

Sistem temelinde otonom bir yapıya sahip olduğundan dolayı sistemin işlemlerinin başarı olması, kullanıcıya değil sisteme bağlıdır. Kullanıcının işleme etkisi ilk kurulumda güvenlik izini vermesine bağlı.

2.3.a Use Case Diagram:



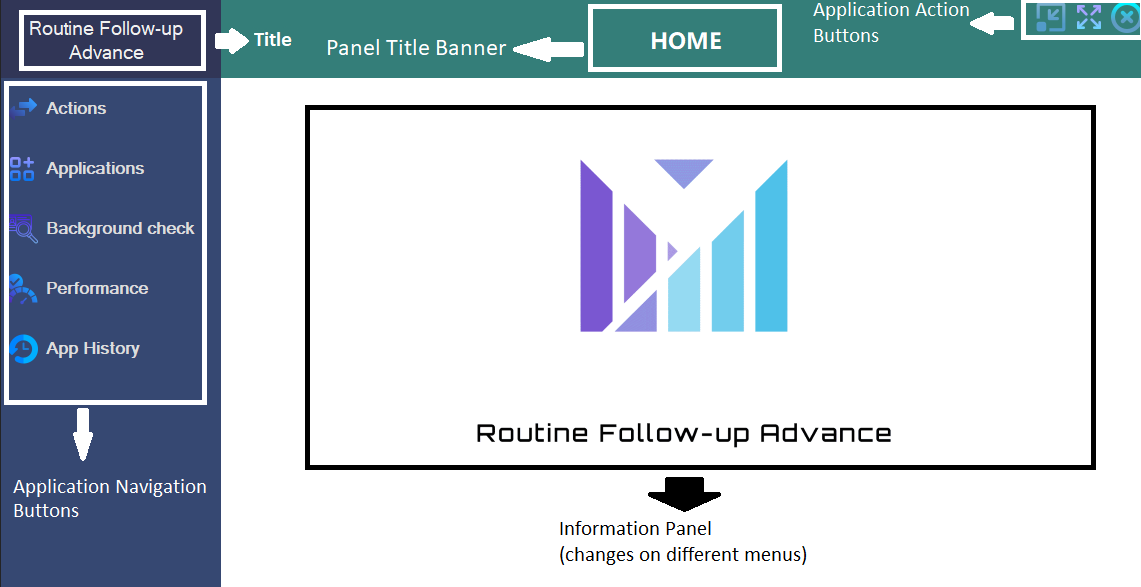
* 1. **Süreç Analizi (Process Modelling)**



1. **SİSTEM TASARIMI**
   1. **Sistem Mimarisi Tasarımı (Architectural Design)**

Proje, “Client-Based” Mimarisi üzerine kuruludur. Temel olarak sunucuya yalnızca veri depolama sorumluluğu verip, istemci tarafında sunum, uygulama ve veri erişim mantıklarının tamamı işlenir.

* 1. **Kullanıcı Arayüzü Tasarımı (User Interface Design)**

****

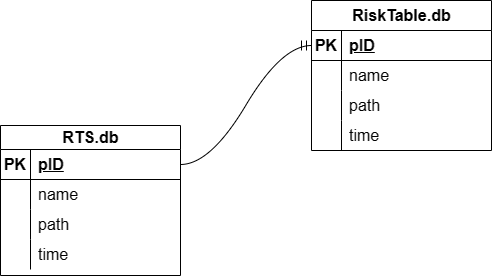
* 1. **Program Tasarımı**

Konsol uygulaması, programı veritabanı ile bağlanmasını ve her gün belirli saatte dışa aktarılıp depolanmasını sağlayan bir programdır. 2 parçadan oluşur birisi 15 dakikada bir very çekip “.db” programına gönderir ve diğeri her gün “.db” den very çekip tarih ile damgalayıp kaydeder dosyaya.

Form uygulaması, temel olarak veritabanından veri çekmeye yarayan temel bir modül ve açık başlıklarla ayrılan bölümleri olan bir görsel yapıya sahip.

Veritabanı, temel olarak uygulamaların temel bilgilerini depolayan “RTS” ve “RiskTable” olmak üzere ikifarklı tablodan oluşup, “RiskTable”, “RTS’den” veri alıyor.

* 1. **Veri Modelleme ve Veritabanı Tasarımı**



1. **KODLAMA**

**Console App:**

Python programlama dilinin, basit ve anlaşılabilir olması ve fazla sayıda kütüphane esnekliği olması sebebiyle “Console App” tarafında uygun seçilmiştir. SQlite3 ve psutil kütüphanelerinden faydalanan ve temel olarak SQL “.connect” ve “.cursor” komutları ile bilgisayarda toplanmaya başlanan bilgiler depolanır ve sistem içinde kontrolü sağlanır.

**Örneğin:**

import psutil

import sqlite3

#Database bağlantısını gerçekleştirmek için gerekli komut dizini

def create\_database():

    conn = sqlite3.connect('RTS.db')

    cursor = conn.cursor()

    #Tablo kontrol et ve yoksa yeni oluştur...

    cursor.execute('''

        CREATE TABLE IF NOT EXISTS RTS (

            pid INTEGER PRIMARY KEY AUTOINCREMENT,

            name TEXT,

            path TEXT,

            time INTEGER,

        )

    ''')

    conn.commit()

    conn.close()

“psutil” kütüphanesi ile birlikte bilgisayar üzerinde gerçekleşen “process” komutlarına ulaşım sağlanabiliyor. Bu kütüphane sayesinde spesifik alanlarla filtrelenmiş uygulamaların hem ID hem isim hem de kurulu olduğu patikanın adresini de gözler önüne sermiştir.

Örneğin:

def getRunningApps():

    apps = []

    PathsToExclude = ['\\Windows\\', '\\Program Files\\', '\\AppData\\', '\\Windows Defender\\']

    AppNames = set()

    for proc in psutil.process\_iter(['pid', 'name', 'exe']):

        if proc.info['name'].endswith('.exe') and not any(path in proc.info['exe'] for path in PathsToExclude):

            if 'CrashHandler' not in proc.info['name']:

                if proc.info['name'] not in AppNames:

                    AppNames.add(proc.info['name'])

                    apps.append(proc.info)

    return apps

**Windows Form App:**

C#' dilinin kullanım amacı, Windows üzerinde çalışan güçlü ve verimli uygulamalar oluşturmaktır. Windows platformunda çeşitli yazılım projeleri geliştirmek için kullanılan bir dil olması yönünden Masaüstü uygulaması geliştirmek için uygun seçilmiştir.

Kotlin'in kullanım amacı, Android uygulamalarının geliştirilmesini daha kolay, hızlı ve güvenli hale getirmektir. Geliştiricilerin yatkınlığına göre mobil uygulama bu tarz yazılmıştır.

1. **TEST**
   1. **Test Senaryosunun Belirlenmesi**

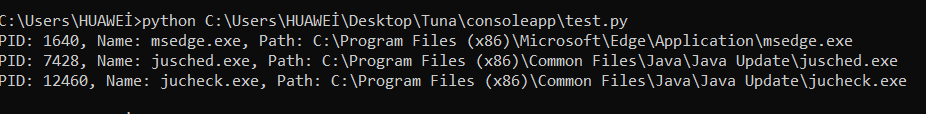
Sıradan günlük kullanımı ve eğlence uygulamalarının aktif kullanıldığı durumlar hedef kitlenin genel kullanımını andıracak şekilde gerçekleşecektir.

* 1. **Birim Testler**

Consol App:

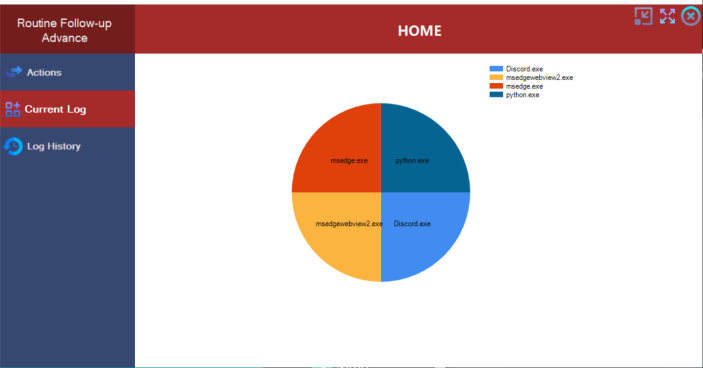
Beklendiği üzere filtreleri efektif olarak işleyip doğru sonuçları veriyor ve test amaçlı kullanılan lokal bir veri-tabanına veri işleyebiliyor…

Program aktifleştiği sürede bilgisayara ekstra yükü gözle görülür bir etki yaratmıyor...



Windows Form App:

Beklendiği üzere istenilen verileri kolaylıkla ulaşıp Veri-Tabanı üzerinden tabloları çağırabiliyor.

* 1. **Bütünleşik Testler**
* Uygulama çalıştırıldığında herhangi bir hata ile karşılaşılmadı ve temel olarak akıcı bir kullanımı vardı.
* Uygulamalar arası iletişim sırasında hata vermeyip, bekletme problem yaratmadı.
* Konsol uygulaması aktifleşme aralığında, genel kullanım için herhangi bir problem olmadığı gözlemlendi.
  1. **Sistem Testi**
* Senaryo dahilinde hem oyun oynanması hem de günlük kullanımda bilgisayarın ne kadar etkilendiği ve toplanan verilerin sağlığı göz önüne alındığında, uygulama hem sağlıklı hem de efektif olduğu gözlemlendi.
* Arka planda çalışan konsol uygulaması her aktifleştiğinde CPU kullanıp, bu kullanımının kullanımı etkilemiyecek ölçüde olduğu gözlemlendi.
  1. **Kullanıcı Testleri**

Geliştirici grup içerisinde yapılan testlerde hem konsol uygulamasaı hem de ana uygulama sorunsuz şekilde, beklenen çıktıları verkmektedir.

PowerShell ile farklı bilgisayarlarda kurulmlarda hata alınmamıştır.

VeriTabanı yerel bağlantılar uygulandığında ya da internet sunucusuna bağlanırken farklı bilgisayarlarda efktifliğinde farklılıklar oluyor.

1. **UYGULAMAYA HAZIRLIK VE UYGULAMAYA KOYMA**

* Uygulama ücretsiz ve yalnızca bağış seçeneği ile dağıtıma çıkacaktır.
* Uygulamanın bir “RTS.rar” sistemi ile dağıtılıyor olması ve yalnızca PowerShell kullanımı ile son kurulum işlemi de yapılabilecektir.
* Kurulum belgesi de “RTS.rar” dosyasının içinde yer alıcaktır.
* Bu dosyalar Github üzerinden erişilebilir ve Veri-Tabanı güncellemelri yine oradan yapılabilir olacak.