**T.C.**

**ESKİŞEHİR OSMANGAZİ ÜNİVERSİTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**2023-2024 Eğitim-Öğretim Yılı**

**Güz Dönemi**

**152113017 – NESNE TABANLI PROGRAMLAMA I**

**PROJE RAPORU**

**Proje Başlığı**

“İnsansı Robot Denetim Sistemi”

**Grup 11**

152120191092 - Gürkan KAYI (Grup Lideri)

152120181073 - Hassan MOSHTAQ

152120191006 - Umutcan ÇAKMAK

152120191023 - Selim Can YAZAR

**Dersin Hocası**

Prof. Dr. Metin ÖZKAN

**OCAK 2024**

İÇİNDEKİLER

[1. Giriş 1](#_Toc155726042)

[2. Tasarım 1](#_Toc155726043)

[2.1. UML Diyagramının İncelenmesi 1](#_Toc155726044)

[2.2. Proje Ortamının Hazırlanması 3](#_Toc155726045)

[2.3. Görev Bölümü 4](#_Toc155726046)

[2.3. Sınıflara İlişkin Bilgiler 6](#_Toc155726047)

[2.4. Menünün Oluşturulması 7](#_Toc155726048)

[2.5. Programın Çalıştırılması 9](#_Toc155726049)

[2.6. Bitbucket Kullanımı 17](#_Toc155726050)

[2.7. Doxygen Kullanımı 18](#_Toc155726051)

[3. Sonuçlar ve Değerlendirme 20](#_Toc155726052)

**TABLOLAR LİSTESİ**

[**Tablo 1: UML Diyagramındaki Sınıflar Arası İlişkiler (1. Aşama)** 2](#_Toc155726054)

[**Tablo 2: UML Diyagramındaki Sınıflar Arası İlişkiler (2. Aşama)** 3](#_Toc155726055)

[**Tablo 3: Görev Bölümü** 5](#_Toc155726056)

[**Tablo 4: Menü UML Diyagramındaki Sınıflar Arası İlişkiler** 8](#_Toc155726057)

[**Tablo 5: Yetkili Robot Operatörü Bilgileri** 9](#_Toc155726058)

**RESİMLER LİSTESİ**

[**Resim 1: generic.dll Kütüphanesi Yükleme Hatası Görüntüsü** 4](#_Toc155726154)

[**Resim 2: Webots Uygulamasında Robot ile Bağlantı Sağlanması** 4](#_Toc155726155)

[**Resim 3: Örnek .h Ekran Görüntüsü** 6](#_Toc155726156)

[**Resim 4: Örnek .cpp Ekran Görüntüsü** 6](#_Toc155726157)

[**Resim 5: Menu UML Diyagramı** 7](#_Toc155726158)

[**Resim 6: Uygulama Login Ekranı Görüntüsü** 9](#_Toc155726159)

[**Resim 7: Main Menü Ekran Görüntüsü** 10](#_Toc155726160)

[**Resim 8: Connection Menüsü Ekran Görüntüsü (1. Aşama)** 10](#_Toc155726161)

[**Resim 9: Connection Menüsü Kullanımı Ekran Çıktısı** 10](#_Toc155726162)

[**Resim 10: Connection Menüsü Ekran Görüntüsü (2. Aşama)** 11](#_Toc155726163)

[**Resim 11: Operate Robot Menüsü Ekran Görüntüsü** 11](#_Toc155726164)

[**Resim 12: Forward Seçeneği Ekran Görüntüsü** 11](#_Toc155726165)

[**Resim 13: Webots Simülasyon Robot İleri Hareket Ekran Görüntüsü** 12](#_Toc155726166)

[**Resim 14: Turn Right Seçeneği Ekran Görüntüsü** 12](#_Toc155726167)

[**Resim 15: Webots Simülasyon Robot Sağa Dön Hareketi Ekran Görüntüsü** 13](#_Toc155726168)

[**Resim 16: Sensör Menüsü Ekran Görüntüsü (1. Aşama)** 13](#_Toc155726169)

[**Resim 17: Sensör Menüsü Kullanımı Ekran Görüntüsü** 13](#_Toc155726170)

[**Resim 18: Sensör Menüsü Ekran Görüntüsü (2. Aşama)** 14](#_Toc155726171)

[**Resim 19: Path Menüsü Ekran Görüntüsü (1. Aşama)** 14](#_Toc155726172)

[**Resim 20: Path Menü Kullanımı Ekran Görüntüsü** 15](#_Toc155726173)

[**Resim 21: Path Menüsü Ekran Görüntüsü (2. Aşama)** 15](#_Toc155726174)

[**Resim 22: Record Menü Kullanımı Ekran Görüntüsü (1. Aşama)** 16](#_Toc155726175)

[**Resim 23: Record Menüsü Ekran Görüntüsü (2. Aşama)** 16](#_Toc155726176)

[**Resim 24: Bitbucket Repository ve Branchleri** 17](#_Toc155726177)

[**Resim 25: Bitbucket Örnek Branch Ekran Görüntüsü** 18](#_Toc155726178)

[**Resim 26: Örnek Doxygen html Ekran Görüntüsü** 19](#_Toc155726179)

# 1. Giriş

Projede Webots isimli robot simülasyon programına bağlantı sağlanarak, yazılan kodlar ile robotun yönlendirilmesi beklenmektedir. Proje konusu, 3B robot simülatöründeki robotun denetimi (hareket ettirilmesi ve sensörlerinden veri alınması) için yazılım geliştirmektir. Bu amaçla proje dosyaları arasında verilen NaoRobotAPI.h, NaoRobotAPI.lib ve robotun hareketlerinin gözlemlenmesini sağlayan RobotController.cpp’ye ek olarak proje föyünde yer alan UML diyagramında gösterilen RobotControl, ForceSensor, SonarSensor, BumperSensor, Pose, Path, Node, Record, RobotOperator ve Encryption sınıflarına ait .h ve .cpp dosyaları ile test dosyaları yazılması istenmektedir. Bu kısımlar tamamlandıktan sonra ise bu sınıfların kullanıldığı bir Menü tasarımı yapılarak bu menüye ait sınıfların yazılması gerekmektedir.

Grup üyelerinin ortak görüşü ile Gürkan Kayı grup lideri olarak seçilmiştir. Grup üyelerinin yazdıkları kodları birbirleri ile paylaşmaları ve kod yönetimi için ödev föyünde Bitbucket uygulaması kullanılması istenmiş, grup lideri tarafından açılan repository’de her grup üyesinin ismine branch oluşturularak kodlar bu branch’lere aktarılmıştır. Sonrasında tüm kodlar birleştirilerek proje ortaya çıkarılmıştır.

Kod dosyalarının dökümantasyonu için de Doxygen uygulamasında kullanılacak şekilde yorum, bilgi satırları düzenlenmiştir. Doxygen uygulaması ile html formatında dökümantasyon dosyaları oluşturulmuştur.

Yukarıda kısaca özetlenen uygulama ve yapılanlara ilişkin bilgilere bu raporda yer verilmektedir.

# 2. Tasarım

## 2.1. UML Diyagramının İncelenmesi

Proje föyünde yer alan UML diyagramında kodlanması istenen 1. aşamada 10 adet sınıf bulunmaktadır. 2. aşamada 4 sınıf daha eklenmiş ve bazı sınıflar arası ilişkiler değişmiş ya da güncellenmiştir. Grup üyeleri arasında dengeli ve adaletli bir paylaşım yapılması düşünülerek 1. aşamada 10 sınıf 4 kişi arasında paylaştırılmış, ek olarak bu sınıflara ait test dosyaları ve menü bölümleri de sınıfları oluşturan üyelere bırakılmıştır. 2. aşamada ise yeni 4 sınıf 4 kişiye paylaştırılmıştır. Görev bölümüne ilişkin detaylı bilgilere tablo 3’ten ulaşılabilir.

Kodlamaların yapılabilmesi için UML diyagramında bulunan sınıflar arasındaki ilişkilerin doğru bir şekilde anlaşılması gerekmektedir. UML diyagramında bulunan sınıflar arasında şu ilişkiler yer almaktadır:

**Tablo 1: UML Diyagramındaki Sınıflar Arası İlişkiler (1. Aşama)**

|  |  |  |
| --- | --- | --- |
| **Sınıf Adı** | **İlişkinin Adı** | **İlişki İçinde Olduğu Sınıf(lar)** |
| NaoRobotAPI | - | - |
| RobotControl | Aggregation | NaoRobotAPI |
| Composition | Pose |
| ForceSensor | Aggregation | NaoRobotAPI |
| SonarSensor | Aggregation | NaoRobotAPI |
| BumperSensor | Aggregation | NaoRobotAPI |
| Pose | - | - |
| Path | Composition | Node |
| Realization/Implementation | Pose |
| Node | Composition | Pose |
| Record | - | - |
| RobotOperator | - | - |
| Encryption | Realization/Implementation | RobotOperator |

**Kaynak:** <https://tugrulbayrak.medium.com/uml-class-diyagramlari-4c3bb7e9cc4c> (Erişim tarihi: 18.12.2023)

**Tablo 2: UML Diyagramındaki Sınıflar Arası İlişkiler (2. Aşama)**

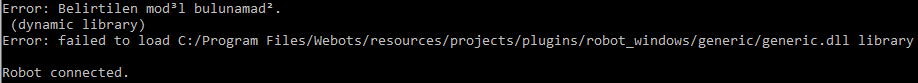
|  |  |  |
| --- | --- | --- |
| **Sınıf Adı** | **İlişkinin Adı** | **İlişki İçinde Olduğu Sınıf(lar)** |
| NaoRobotAPI | - | - |
| NaoRobotInterface | Aggregation | NaoRobotAPI |
| Inheritance | RobotInterface |
| RobotInterface | Composition | Pose |
| Aggregation | SensorInterface |
| RobotControl | Composition | RobotInterface |
| Composition | SensorInterface |
| Composition | Pose |
| Composition | Path |
| Composition | Record |
| Composition | RobotOperator |
| SensorInterface | - | - |
| NaoRobotSensorInterface | Inheritance | SensorInterface |
| ForceSensor | Inheritance | NaoRobotSensorInterface |
| SonarSensor | Inheritance | NaoRobotSensorInterface |
| BumperSensor | Inheritance | NaoRobotSensorInterface |
| Pose | - | - |
| Path | Composition | Node |
| Realization/Implementation | Pose |
| Node | Composition | Pose |
| Record | - | - |
| RobotOperator | Realization/Implementation | Encryption |
| Encryption | - | - |

## 2.2. Proje Ortamının Hazırlanması

Proje ortamı hazırlanırken ödev föyü ve ek-1 dosyasından faydalanılmıştır. Webots versiyon R2023b indirilerek kurulmuştur. Visual Studio 2022 programında gerekli ayarlar yapıldıktan sonra, proje paketi içinde verilen ve Webots dizininde bulunan dosyalar ilgili dizinler altına kopyalanmıştır.

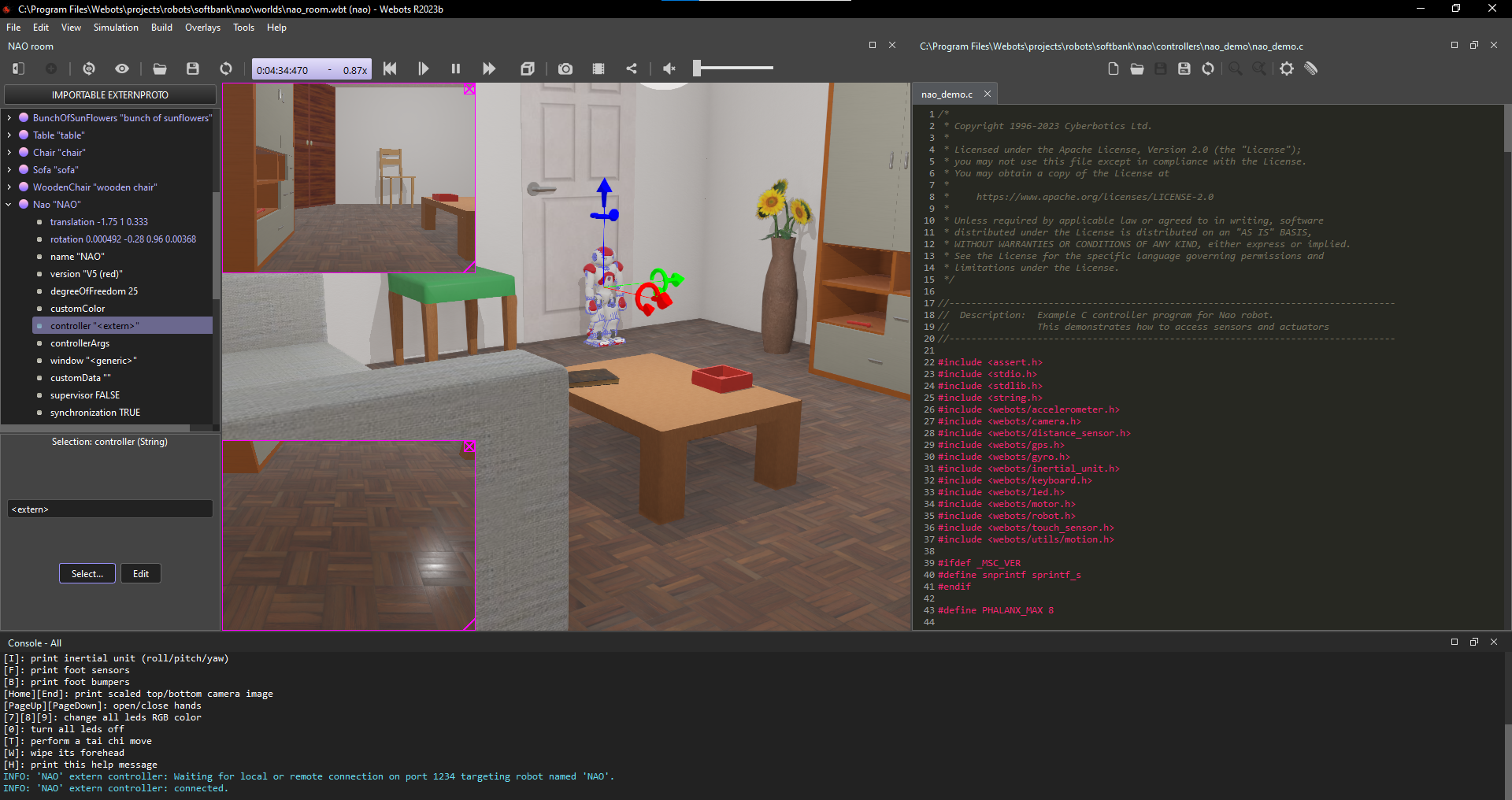
C:\Program Files\Webots\resources\projects\plugins\robot\_windows\generic dizininde bulunan generic.dll dosyası <Proje\_Adi>\x64\Debug dizinine kopyalanmasına rağmen konsol ekranında hata alınmaktadır. Resim 1’de bu hataya ilişkin görüntü yer almaktadır.

**Resim 1: generic.dll Kütüphanesi Yükleme Hatası Görüntüsü**



Bu hatanın giderilmesi için Webots programı kaldırılarak yeniden kurulmuş, oluşan dosyalar ilgili dizinlere kopyalanarak tekrar denenmiş ancak yine aynı hata alınmaya devam edilmiştir. Alınan bu hataya rağmen robot ile bağlantı kurulmakta ve robot komutlara yanıt vermektedir. Kodlamaya başlanmadan önce RobotController.cpp dosyası çalıştırılarak robotun nasıl çalıştığı ve komutlara göre hareketleri anlaşılmaya çalışılmış, sonraki kodlamalarda bundan yararlanılmıştır. Resim 2’de Webots uygulaması ile robotun bağlandığına dair görüntüye yer verilmektedir.

**Resim 2: Webots Uygulamasında Robot ile Bağlantı Sağlanması**

## 2.3. Görev Bölümü

UML diyagramındaki ilişkiler göz önünde bulundurularak yazılacak kodların grup üyeleri arasında paylaşımında ilişki içerisinde olan sınıflar dikkate alınmıştır. Buna göre yapılan görev bölümü tablo 3’te gösterilmektedir. Tabloda menü ve diğer görevlendirmelere ilişkin de bilgiler bulunmaktadır. Test dosyaları sınıfları yazan üyeler tarafından oluşturulmuştur. Menünün tasarlanması ortak olarak yapılmış olup sonrasında menü UML’i buna göre oluşturulmuştur.

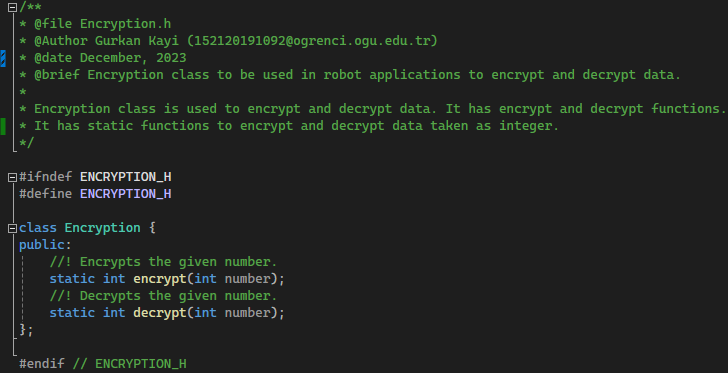
**Tablo 3: Görev Bölümü**

|  |  |
| --- | --- |
| **Takım Üyesi** | **Görevleri** |
| Gürkan Kayı | Record class (h, cpp)  RecordTest  RobotOperator class (h, cpp)  RobotOperatorTest  Encryption class (h, cpp)  EncryptionTest  OperatorLoginMenu class (h, cpp)  RecordMenu class (h, cpp)  NaoRobotInterface class (h, cpp)  Dosyaların Dökümantasyonu  Proje Raporunun Hazırlanması |
| Hassan Moshtaq | BumperSensor class (h, cpp)  BumperSensorTest  ForceSensor class (h, cpp)  ForceSensorTest  Menu class (h, cpp)  MainMenu class (h, cpp)  SensorMenu class (h, cpp)  SensorInterface (h, cpp)  NaoRobotSensorInterface (h)  Dosyaların Dökümantasyonu  Menu UML’nin Hazırlanması |
| Umutcan Çakmak | RobotControl class (h, cpp)  RobotControlTest  Pose class (h, cpp)  PoseTest  Node class (h, cpp)  NodeTest  ConnectionMenu class (h, cpp)  RobotInterface (h, cpp)  Dosyaların Dökümantasyonu |
| Selim Can Yazar | SonarSensor class (h, cpp)  SonarSensorTest  Path class (h, cpp)  PathTest  OperateRobotMenu class (h, cpp)  PathMenu class (h, cpp)  Main.cpp  SensorInterface (h, cpp)  NaoRobotSensorInterface (h)  Dosyaların Dökümantasyonu |

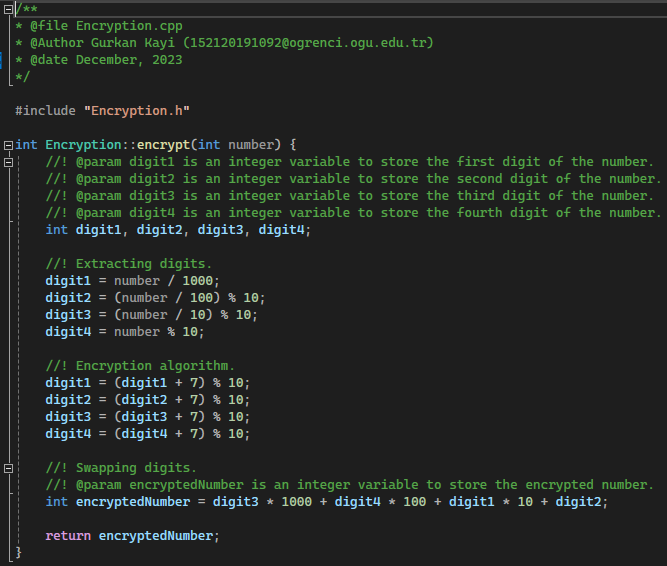
## 2.3. Sınıflara İlişkin Bilgiler

Sınıflarda kullanılan attribute ve method’lara ilişkin ayrıntılı açıklamalar kodların yanında oluşturulan yorum satırları ve bilgilerde bulunmaktadır. Örnek olması açısından resim 3’te ve 4’te bir sınıfın görüntülerine yer verilmektedir.

**Resim 3: Örnek .h Ekran Görüntüsü**



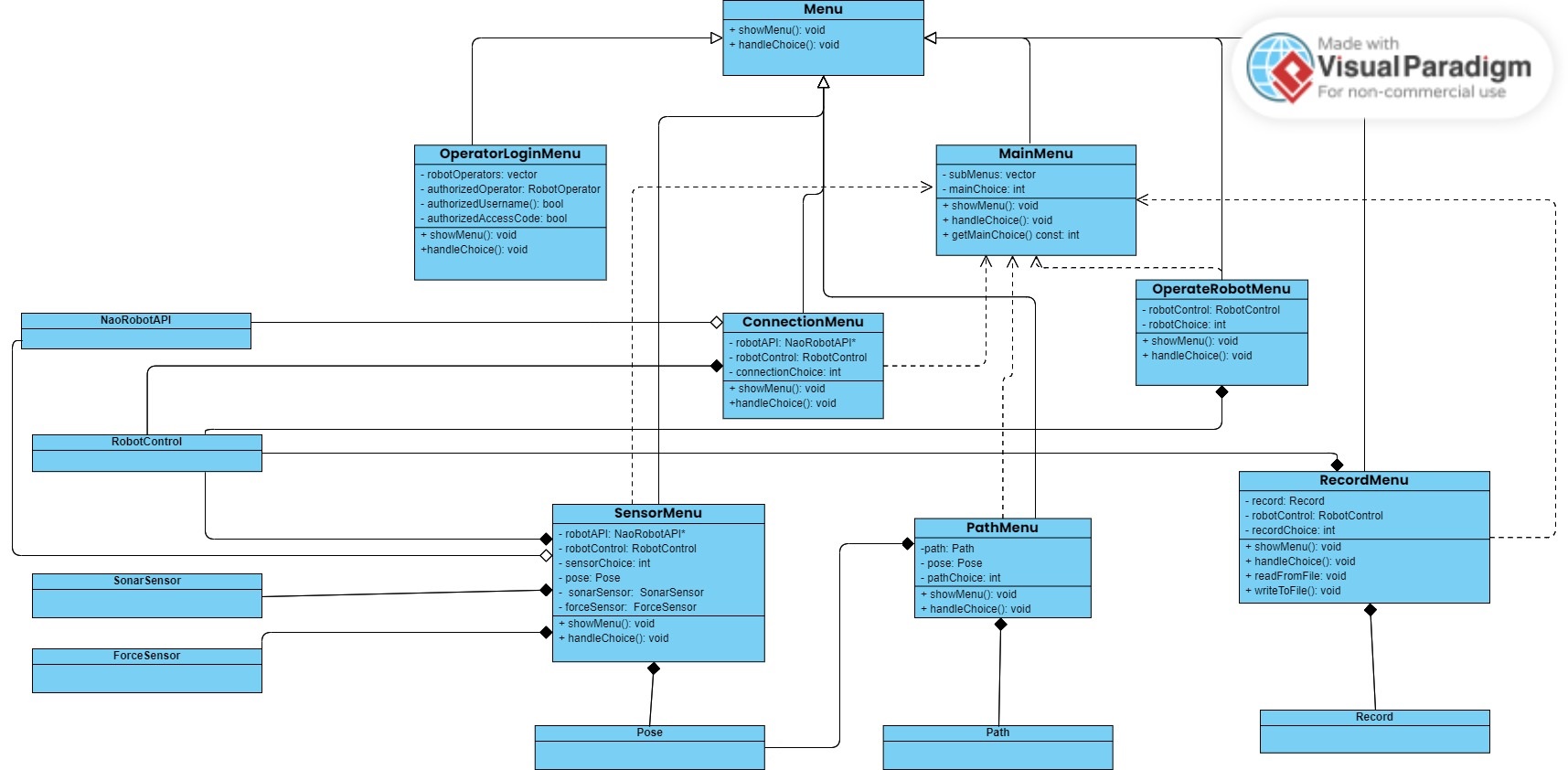
**Resim 4: Örnek .cpp Ekran Görüntüsü**



## 2.4. Menünün Oluşturulması

Sınıflar kodlandıktan ve test edildikten sonra menü tasarlanmıştır. Ödev föyünde istendiği şekilde ve örnek verilen menü tasarımındaki gibi metin tabanlı ve kullanımı kolay bir menü hazırlanmaya gayret gösterilmiştir. Menüde mümkün olduğunca fazla sınıfa ait bilgilerin yer aldığı bölümler tasarlanmıştır. Menünün tasarımına ilişkin resim 5’te yer alan UML diyagramı hazırlanmıştır.

**Resim 5: Menu UML Diyagramı**



Menü UML diyagramının daha detaylı görülebilmesi için jpg formatındaki dosya ayrıca zip klasörü içine de konulmuştur.

Menüdeki sınıflar arasında yer alan ilişkiler tablo 4’te bulunmaktadır.

**Tablo 4: Menü UML Diyagramındaki Sınıflar Arası İlişkiler**

|  |  |  |
| --- | --- | --- |
| **Sınıf Adı** | **İlişkinin Adı** | **İlişki İçinde Olduğu Sınıf(lar)** |
| Menu | - | - |
| OperatorLoginMenu | Inheritance | Menu |
| MainMenu | Inheritance | Menu |
| ConnectionMenu | Inheritance | Menu |
| Realization/Implementation | MainMenu |
| Aggregation | NaoRobotAPI |
| Composition | RobotControl |
| OperateRobotMenu | Inheritance | Menu |
| Realization/Implementation | MainMenu |
| Composition | RobotControl (1. Aşama)  NaoRobotInterface (2. Aşama) |
| SensorMenu | Inheritance | Menu |
| Realization/Implementation | MainMenu |
| Aggregation | NaoRobotAPI |
| Composition | RobotControl |
| Composition | SonarSensor |
| Composition | ForceSensor |
| Composition | Pose |
| PathMenu | Inheritance | Menu |
| Realization/Implementation | MainMenu |
| Composition | Pose |
| Composition | Path |
| RecordMenu | Inheritance | Menu |
| Realization/Implementation | Pose |
| Composition | Record |

Menu sınıfı diğer menülerin atası durumundadır ve bu menüde virtual fonksiyonlar kullanılmıştır. Böylece alt menüler için standart oluşturulmuştur. Menüler arasındaki bu ilişkilere dayalı görev bölümünde belirtildiği şekilde paylaşım yapılarak menü(ler) oluşturulmuştur.

## 2.5. Programın Çalıştırılması

Programı çalıştırmak için oluşturulan Main.cpp dosyasında önce testler çağrılmaktadır.

Sonra 4 adet grup üyesi için giriş bilgileri (operatör yetkilendirmesi) yapılmaktadır. Giriş bilgileri tabloda bulunmaktadır.

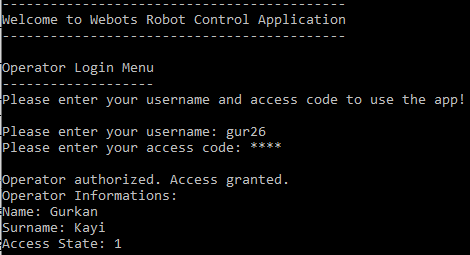
**Tablo 5: Yetkili Robot Operatörü Bilgileri**

|  |  |  |  |
| --- | --- | --- | --- |
| **Operatör Adı** | **Soyadı** | **Kullanıcı Adı** | **Şifresi** |
| Gurkan | Kayi | gur26 | 1234 |
| Hassan | Moshtaq | Has26 | 5678 |
| Umutcan | Cakmak | umcan | 1357 |
| Selim Can | Yazar | secan | 2468 |

Uygulamayı kullanmak için kullanıcı adı ve şifre istenmektedir. Bu 4 yetkili operatör bilgileri ile uygulamaya giriş yapılabilir. Yine yeni operatörler eklenebilir.

Şifre girilirken şifrenin görünmemesi için asterisk “\*” karakterler kullanılmıştır. Bu işlemler OperatorLoginMenu sınıfında yapılmaktadır. Resim’de uygulama çalıma ekran görüntüsü bulunmaktadır.

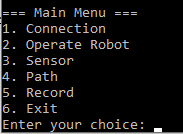
**Resim 6: Uygulama Login Ekranı Görüntüsü**



Kullanıcı adı ya da şifre yanlış girildiğinde bilginin tekrar girilmesi istenmektedir. Şifre asterisk karakterler ile görüntülenmesinin yanında encryption sınıfı sayesinde föyde istenildiği şekilde şifrelenmektedir.

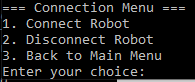
Kullanıcı adı ve şifre doğru ise Main menü kısmına ulaşılmaktadır. Main Menu 5 alt menü ve çıkış seçeneklerine sahiptir.

**Resim 7: Main Menü Ekran Görüntüsü**



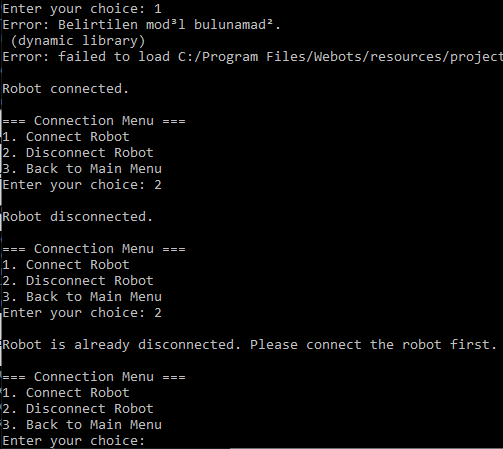
Connection menüsünde 1. aşamada 3 seçenek vardır. Burada robot bağlantısı sağlanmakta veya robot bağlantısı kesilmektedir.

**Resim 8: Connection Menüsü Ekran Görüntüsü (1. Aşama)**



Connection menüsü kullanımı aşağıda yer almaktadır.

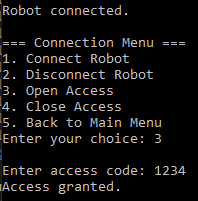
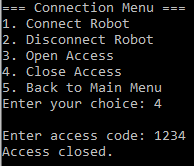
**Resim 9: Connection Menüsü Kullanımı Ekran Çıktısı**



Robot zaten bağlı veya bağlı değil ise uyarı verilmektedir.

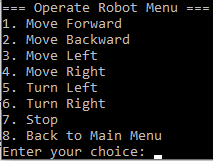
Connection menüsünde 2. aşamada 5 seçenek vardır. Buraya Open ve Close Access seçenekleri eklenmiştir.

**Resim 10: Connection Menüsü Ekran Görüntüsü (2. Aşama)**

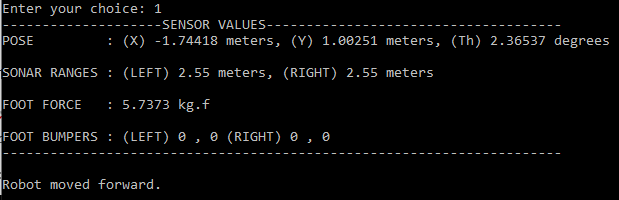
Operate Robot menüsünde 8 seçenek bulunmaktadır. Robot ile bağlantı sağlandıktan sonra bu menüden robot hareket ettirilmektedir.

**Resim 11: Operate Robot Menüsü Ekran Görüntüsü**

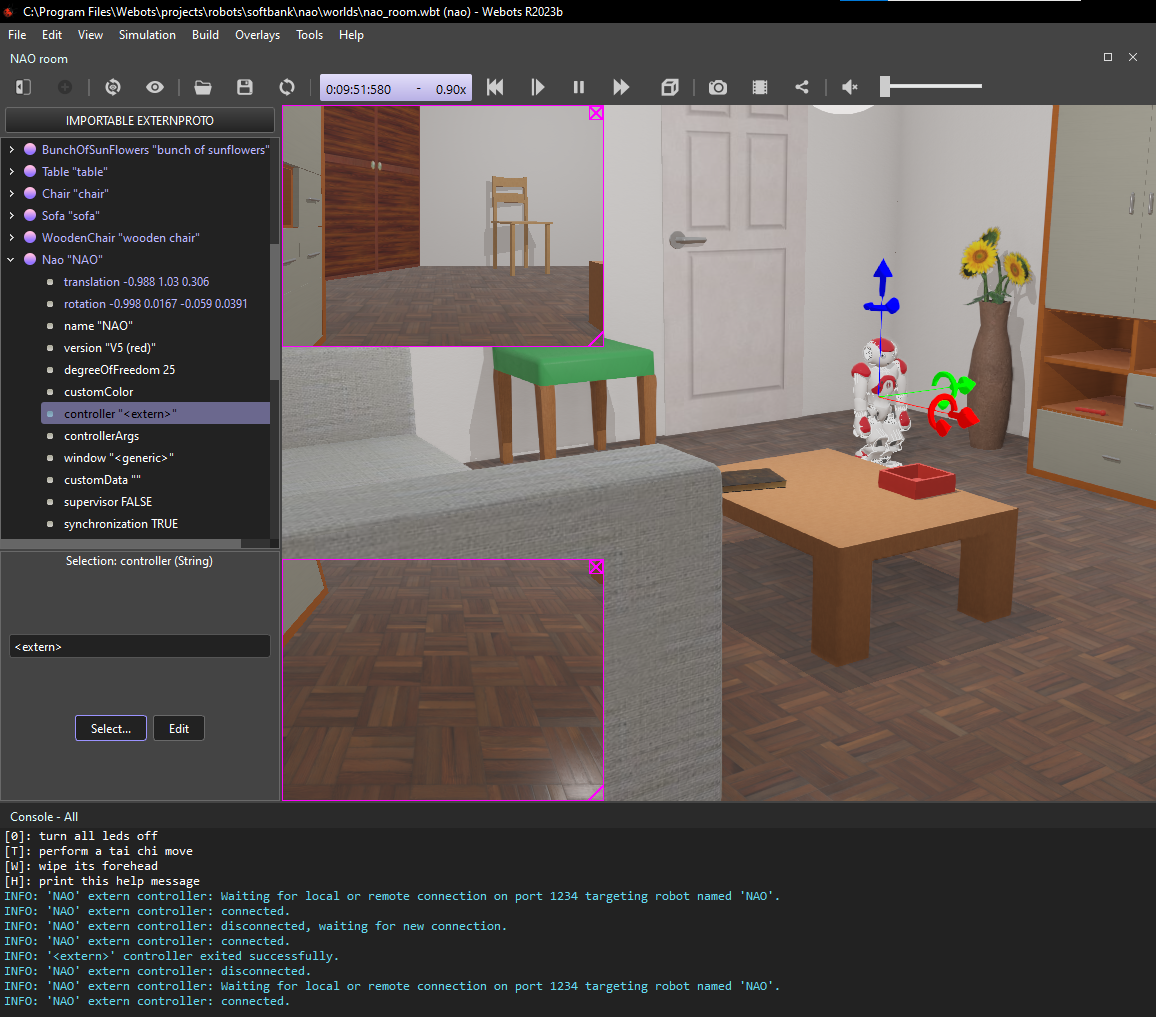


Örneğin ileri seçeneğine basıldığında robot ileri hareket etmekte ve bilgileri ekrana yazdırılmaktadır.

**Resim 12: Forward Seçeneği Ekran Görüntüsü**

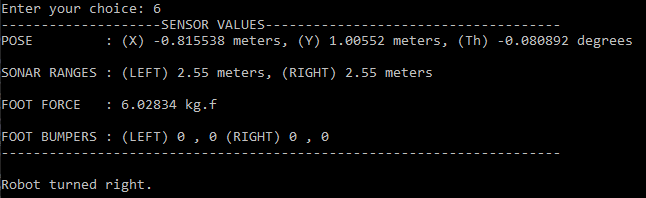


**Resim 13: Webots Simülasyon Robot İleri Hareket Ekran Görüntüsü**

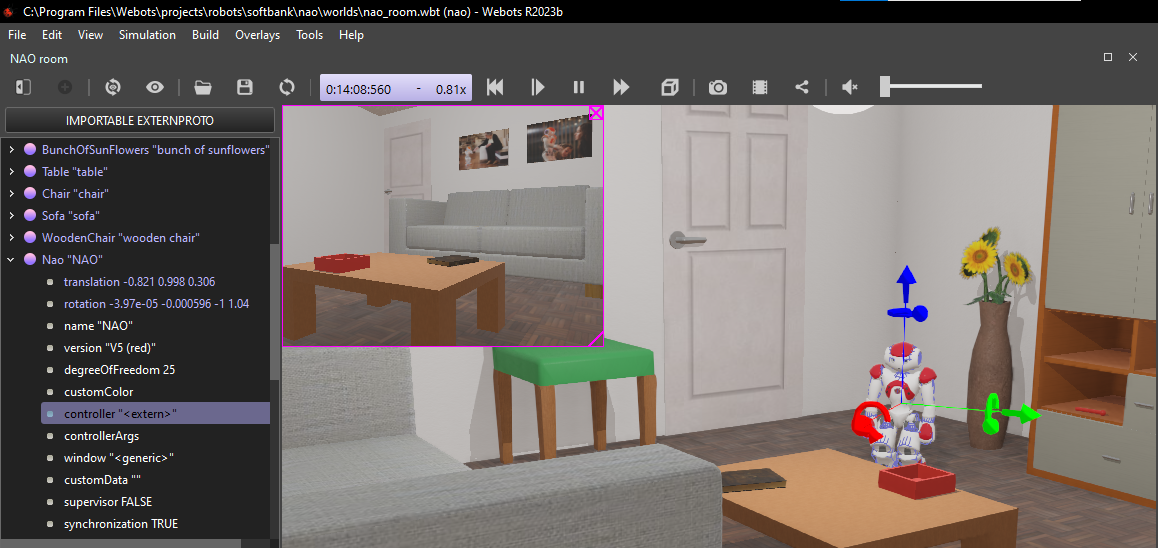


Ya da sağa dön seçildiğinde oluşan durum aşağıda yer almaktadır.

**Resim 14: Turn Right Seçeneği Ekran Görüntüsü**

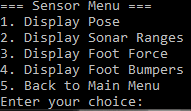


**Resim 15: Webots Simülasyon Robot Sağa Dön Hareketi Ekran Görüntüsü**

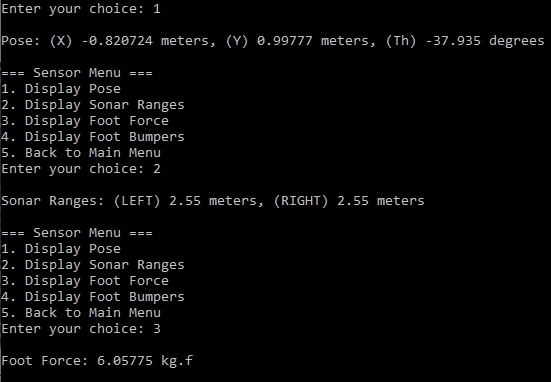


Sensör menüsünde 1. aşamada 5 seçenek bulunmaktadır. Burada mevcut konumun sensör değerleri görüntülenmektedir.

**Resim 16: Sensör Menüsü Ekran Görüntüsü (1. Aşama)**

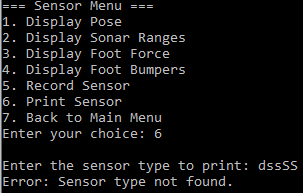


**Resim 17: Sensör Menüsü Kullanımı Ekran Görüntüsü**



Sensör menüsü 2. Aşamada 7 seçenek bulunmaktadır. Record ve Print Sensor seçenekleri buraya eklenmiştir.

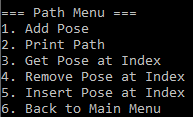
**Resim 18: Sensör Menüsü Ekran Görüntüsü (2. Aşama)**



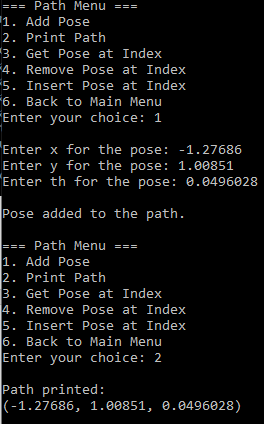
Path menüsünde 1. aşamada 6 seçenek yer almaktadır. Örneğin Add Pose tıklanıp x, y ve th değerleri girildiğinde bu değerler listeye eklenmekte ve daha sonra Print Path denilerek görüntülenebilmektedir.

Yine birden fazla indekste bulunan değerler Get Pose at Index komutu sayesinde görüntülenmektedir. Remove Pose at Index ile girilen indekste yer alan değerler silinmekte ve son olarak Insert Pose at Index komutu ile de girilen indekse değerler eklenmektedir. Path boş ise uyarı yazısı çıkmaktadır.

**Resim 19: Path Menüsü Ekran Görüntüsü (1. Aşama)**

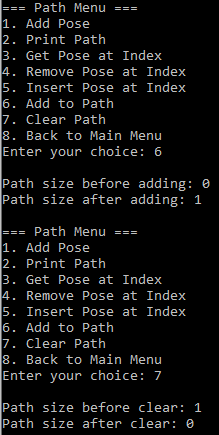


**Resim 20: Path Menü Kullanımı Ekran Görüntüsü**



Path menüsü 2. Aşamada 8 seçenekten oluşmaktadır. Add to Path ve Clear Path seçenekleri buraya eklenmiştir.

**Resim 21: Path Menüsü Ekran Görüntüsü (2. Aşama)**



Record menüsünde 1. aşamada dosyadan okuma ve dosyaya yazdırma işlemleri yapılabilmektedir. Read From File denilerek önceden oluşturulmuş dosyadan okuam yapılabilir, Write To File seçilerek de dosyaya yazdırma yapılabilir. Komut satırında yapılan işlemler ve ekran görüntüleri resimde gösterilmektedir.

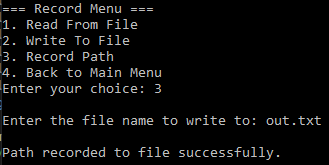
**Resim 22: Record Menü Kullanımı Ekran Görüntüsü (1. Aşama)**

|  |  |
| --- | --- |
|  |  |

Program durdurulduğunda ya da konsol ekranı kapatıldığında robot bağlantısı kesilmektedir.

Record Menüsünde 2. Aşamada 4 seçenek vardır. Record Path seçeneği buraya eklenmiştir.

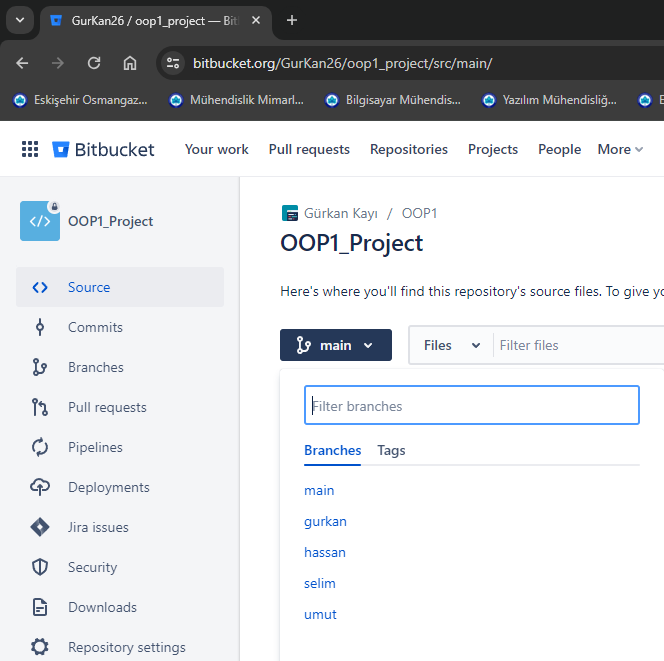
**Resim 23: Record Menüsü Ekran Görüntüsü (2. Aşama)**



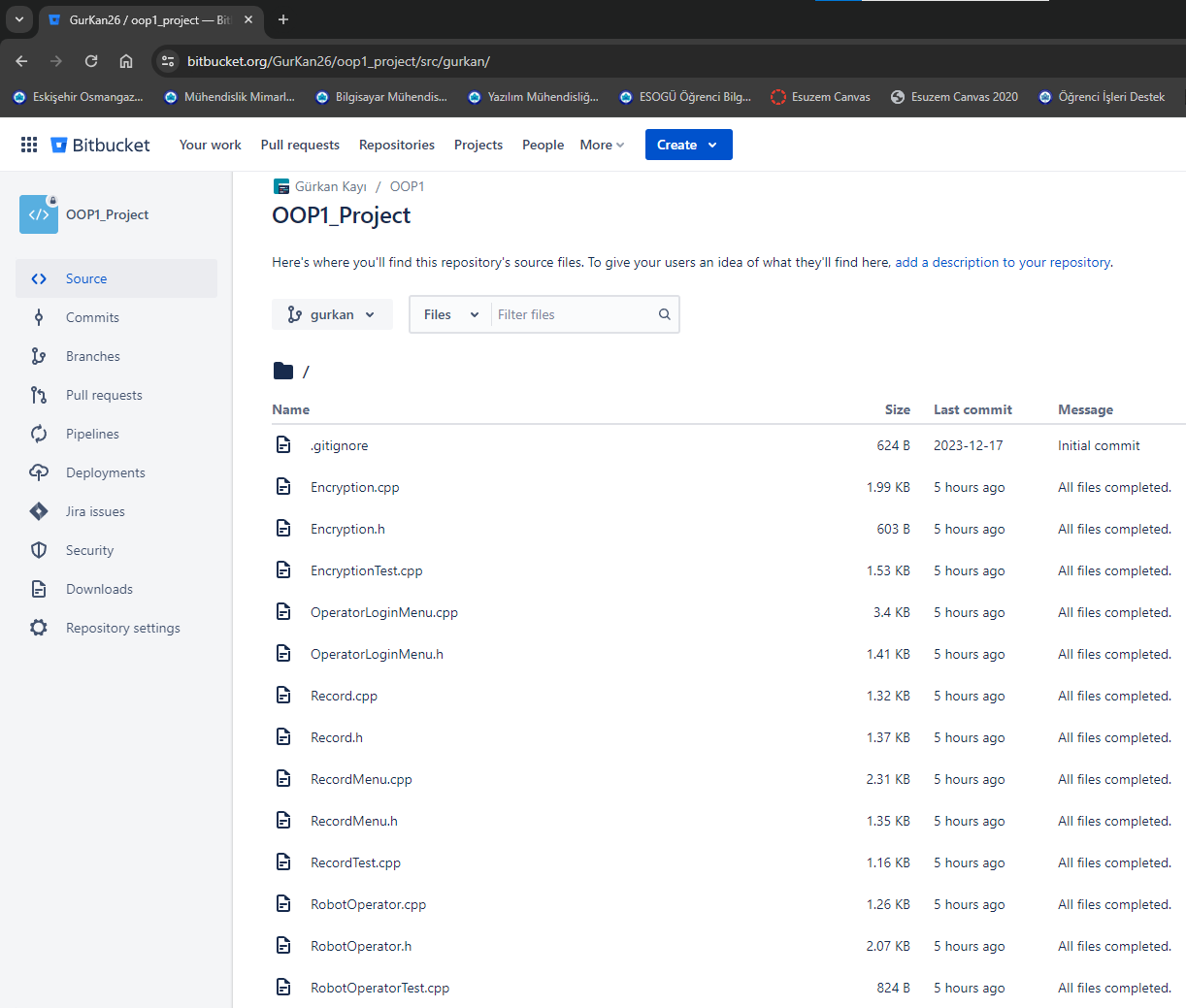
## 2.6. Bitbucket Kullanımı

Bitbucketta oluşturulan repo ve branchlere ilişkin ekran görüntüsü resimde gösterilmektedir. Ayrıca örnek branch görüntüsüne de yer verilmektedir.

**Resim 24: Bitbucket Repository ve Branchleri**



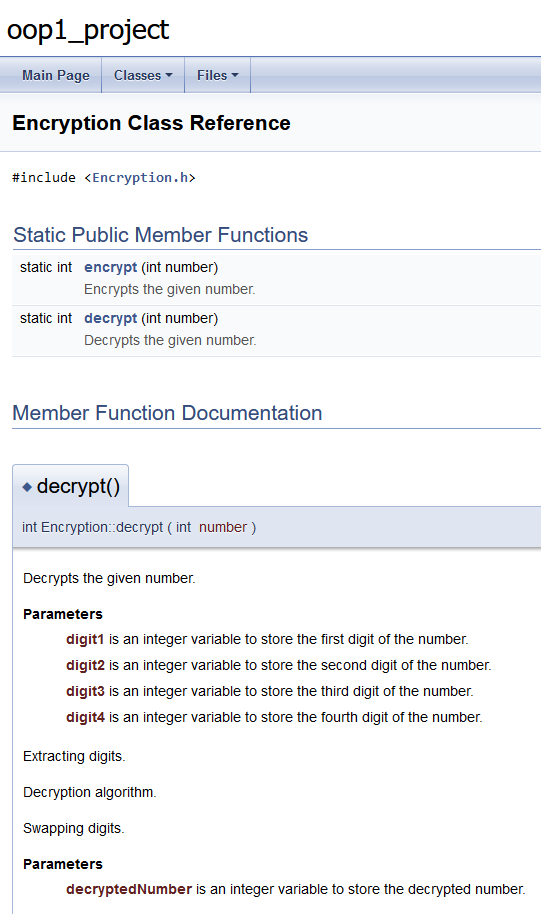
**Resim 25: Bitbucket Örnek Branch Ekran Görüntüsü**



## 2.7. Doxygen Kullanımı

Kodların dökümantasyonu için Doxygen uygulaması kullanılmış ve html formatında dökümantasyon belgeleri oluşturulmuştur. Resimde örnek html ekran görüntüsü bulunmaktadır.

**Resim 26: Örnek Doxygen html Ekran Görüntüsü**



# 3. Sonuçlar ve Değerlendirme

Projede nesne tabanlı yaklaşıma uygun olacak şekilde sınıflar oluşturmaya ve sınıflar arasında inheritance, composition, aggregation, realization/implementation gibi ilişkiler gözetilerek kod yazılmaya çalışılmıştır. Grup üyeleri nezdinde nesne tabanlı yapıların (abstraction, inheritance, polymorphism, templates, exception handling, etc) kullanılmaya çalışılarak kod yazılması temel amaç olmuştur.

Grup üyeleri arasında dengeli bir dağılım, uyum, işbirliği ve haberleşme yürütülmeye özen gösterilmiştir. Proje verilmeden önce grup oluşturulması aşamasında grup üyeleri arasında rahatça iletişim sağlanabilmesi için mesajlaşma uygulaması üzerinden grup kurulmuştur. Proje verildiği günden itibaren yaklaşık 6 saati bulan 5-6 online toplantı düzenlenmiş, toplantılarda föyde istenen-belirtilen çalışmalara ilişkin fikir alışverişinde bulunulmuş ve görev paylaşımı yapılarak projenin ilerlemesi sağlanmıştır.

Bitbucket uygulamasının kullanımı sayesinde iş takibi düzenli bir şekilde yapılabilmiştir. Bu bakımdan bu tür uygulamaların kullanımının kod yazılması sürecinde proje takibi açısından son derece önemli olduğu değerlendirilmektedir.

Doxygen kullanımı ile projeye sonraki zamanlarda göz atıldığında ya da proje dışından kişiler kodlara baktığında kolayca hatırlanabilmesi-anlaşılabilmesi sağlanmaktadır. Dökümantasyon açısından bu uygulamanın kullanımı kesinlikle önerilmektedir.

Projenin artıları ve çıktılarının oldukça önemli olduğu görülmektedir. Bir taraftan nesne tabanlı programlanın temelleri esas alınarak oluşturulan sınıf, nesne gibi varlıklar sayesinde uzun ve karışık kodlama yerine sade, anlaşılır ve basit kodlama yapılmaktadır. Diğer taraftan grup çalışması yapılarak mesleki hayata dair önemli tecrübeler kazanılmaktadır. Öte yandan da Bitbucket, Doxygen gibi yazılımcıların başucu uygulamalardan ikisinin kullanımı pekiştirilmektedir.

Projede olumsuz bir durum olarak generic.dll kütüphanesi kullanımında alınan hataya yer verilebilir. Webots uygulaması kaldırılıp tekrar kurulmasına ve ilgili dosyalar ilgili dizinlere yeniden kopyalanmasına rağmen bu sorun ortadan kaldırılamamıştır. Belki de bu sebepten uygulamada bazen simülasyon ile bağlantı sıkıntıları yaşanabilmektedir. Yine de bu hata genel olarak uygulamanın kullanılmasına engel bir durum teşkil etmemektedir.

Genel olarak proje sayesinde önemli kazanımlar sağlanıldığı söylenebilir.