

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
2. Tasarım	1
2.1. UML Diyagramının İncelenmesi	1
2.2. Proje Ortamının Hazırlanması	3
2.3. Görev Bölümü.....	4
2.3. Sınıflara İlişkin Bilgiler	6
2.4. Menünün Oluşturulması	7
2.5. Programın Çalıştırılması	9
2.6. Bitbucket Kullanımı.....	17
2.7. Doxygen Kullanımı.....	18
3. Sonuçlar ve Değerlendirme	20

TABLÖLER LİSTESİ

Tablo 1: UML Diyagramındaki Sınıflar Arası İlişkiler (1. Aşama)	2
Tablo 2: UML Diyagramındaki Sınıflar Arası İlişkiler (2. Aşama)	3
Tablo 3: Görev Bölümü	5
Tablo 4: Menü UML Diyagramındaki Sınıflar Arası İlişkiler.....	8
Tablo 5: Yetkili Robot Operatörü Bilgileri.....	9

RESİMLER LİSTESİ

Resim 1: generic.dll Kütüphanesi Yükleme Hatası Görüntüsü	4
Resim 2: Webots Uygulamasında Robot ile Bağlantı Sağlanması.....	4
Resim 3: Örnek .h Ekran Görüntüsü	6
Resim 4: Örnek .cpp Ekran Görüntüsü	6
Resim 5: Menü UML Diyagramı	7
Resim 6: Uygulama Login Ekranı Görüntüsü.....	9
Resim 7: Main Menü Ekran Görüntüsü	10
Resim 8: Connection Menüsü Ekran Görüntüsü (1. Aşama).....	10
Resim 9: Connection Menüsü Kullanımı Ekran Çıktısı	10
Resim 10: Connection Menüsü Ekran Görüntüsü (2. Aşama).....	11
Resim 11: Operate Robot Menüsü Ekran Görüntüsü	11
Resim 12: Forward Seçeneği Ekran Görüntüsü.....	11
Resim 13: Webots Simülasyon Robot İleri Hareket Ekran Görüntüsü	12
Resim 14: Turn Right Seçeneği Ekran Görüntüsü	12
Resim 15: Webots Simülasyon Robot Sağa Dön Hareketi Ekran Görüntüsü.....	13
Resim 16: Sensör Menüsü Ekran Görüntüsü (1. Aşama).....	13
Resim 17: Sensör Menüsü Kullanımı Ekran Görüntüsü.....	13
Resim 18: Sensör Menüsü Ekran Görüntüsü (2. Aşama).....	14
Resim 19: Path Menüsü Ekran Görüntüsü (1. Aşama)	14
Resim 20: Path Menü Kullanımı Ekran Görüntüsü	15
Resim 21: Path Menüsü Ekran Görüntüsü (2. Aşama)	15
Resim 22: Record Menü Kullanımı Ekran Görüntüsü (1. Aşama)	16
Resim 23: Record Menüsü Ekran Görüntüsü (2. Aşama).....	16
Resim 24: Bitbucket Repository ve Branchleri	17
Resim 25: Bitbucket Örnek Branch Ekran Görüntüsü	18
Resim 26: Örnek Doxygen html Ekran Görüntüsü	19

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.

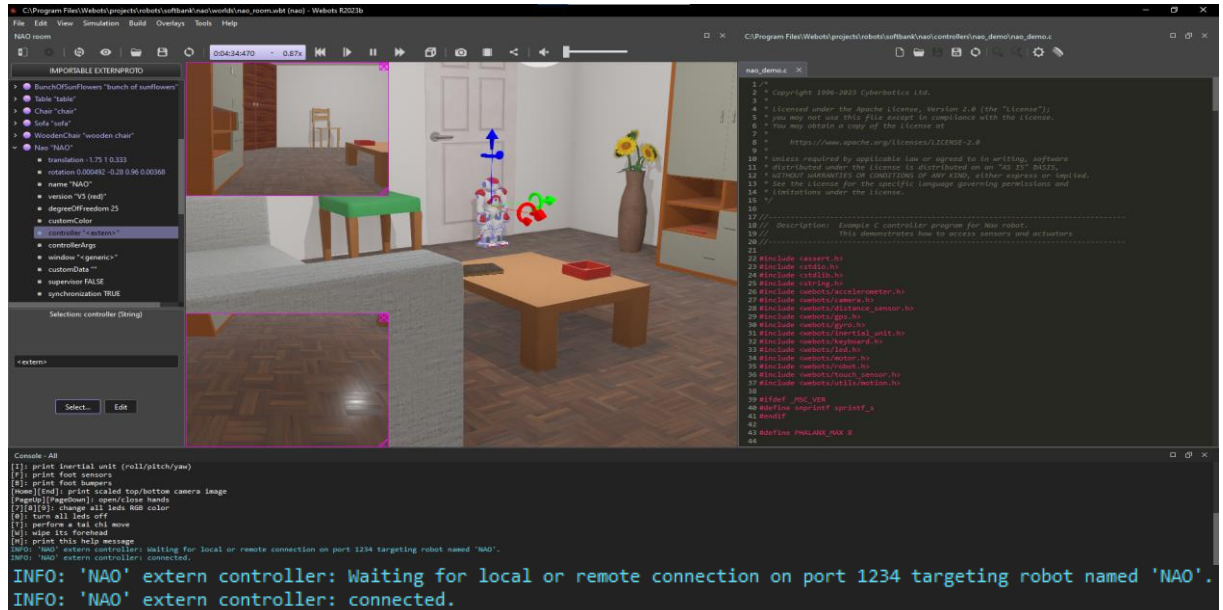
C:\Program Files\Webots\resources\projects\plugins\robot_windows\generic dizininde bulunan generic.dll dosyası <Proje_Adi>\x64\Debug dizinine kopyalanmasına rağmen konsol ekranında hata alınmaktadır. Resim 1’de bu hataya ilişkin görüntü yer almaktadır.

Resim 1: generic.dll Kütüphanesi Yükleme Hatası Görüntüsü

```
Error: Belirtilen mod³l bulunamad².  
(dynamic library)  
Error: failed to load C:/Program Files/Webots/resources/projects/plugins/robot_windows/generic/generic.dll library  
Robot connected.
```

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. Menü’nün 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

```
/**
 * @file Encryption.h
 * @Author Gurkan Kayi (152120191092@ogrenci.ogu.edu.tr)
 * @date December, 2023
 * @brief Encryption class to be used in robot applications to encrypt and decrypt data.
 *
 * Encryption class is used to encrypt and decrypt data. It has encrypt and decrypt functions.
 * It has static functions to encrypt and decrypt data taken as integer.
 */

#ifndef ENCRYPTION_H
#define ENCRYPTION_H

class Encryption {
public:
    ///! Encrypts the given number.
    static int encrypt(int number);
    ///! Decrypts the given number.
    static int decrypt(int number);
};

#endif // ENCRYPTION_H
```

Resim 4: Örnek .cpp Ekran Görüntüsü

```
/**
 * @file Encryption.cpp
 * @Author Gurkan Kayi (152120191092@ogrenci.ogu.edu.tr)
 * @date December, 2023
 */

#include "Encryption.h"

int Encryption::encrypt(int number) {
    ///! @param digit1 is an integer variable to store the first digit of the number.
    ///! @param digit2 is an integer variable to store the second digit of the number.
    ///! @param digit3 is an integer variable to store the third digit of the number.
    ///! @param digit4 is an integer variable to store the fourth digit of the number.
    int digit1, digit2, digit3, digit4;

    ///! Extracting digits.
    digit1 = number / 1000;
    digit2 = (number / 100) % 10;
    digit3 = (number / 10) % 10;
    digit4 = number % 10;

    ///! Encryption algorithm.
    digit1 = (digit1 + 7) % 10;
    digit2 = (digit2 + 7) % 10;
    digit3 = (digit3 + 7) % 10;
    digit4 = (digit4 + 7) % 10;

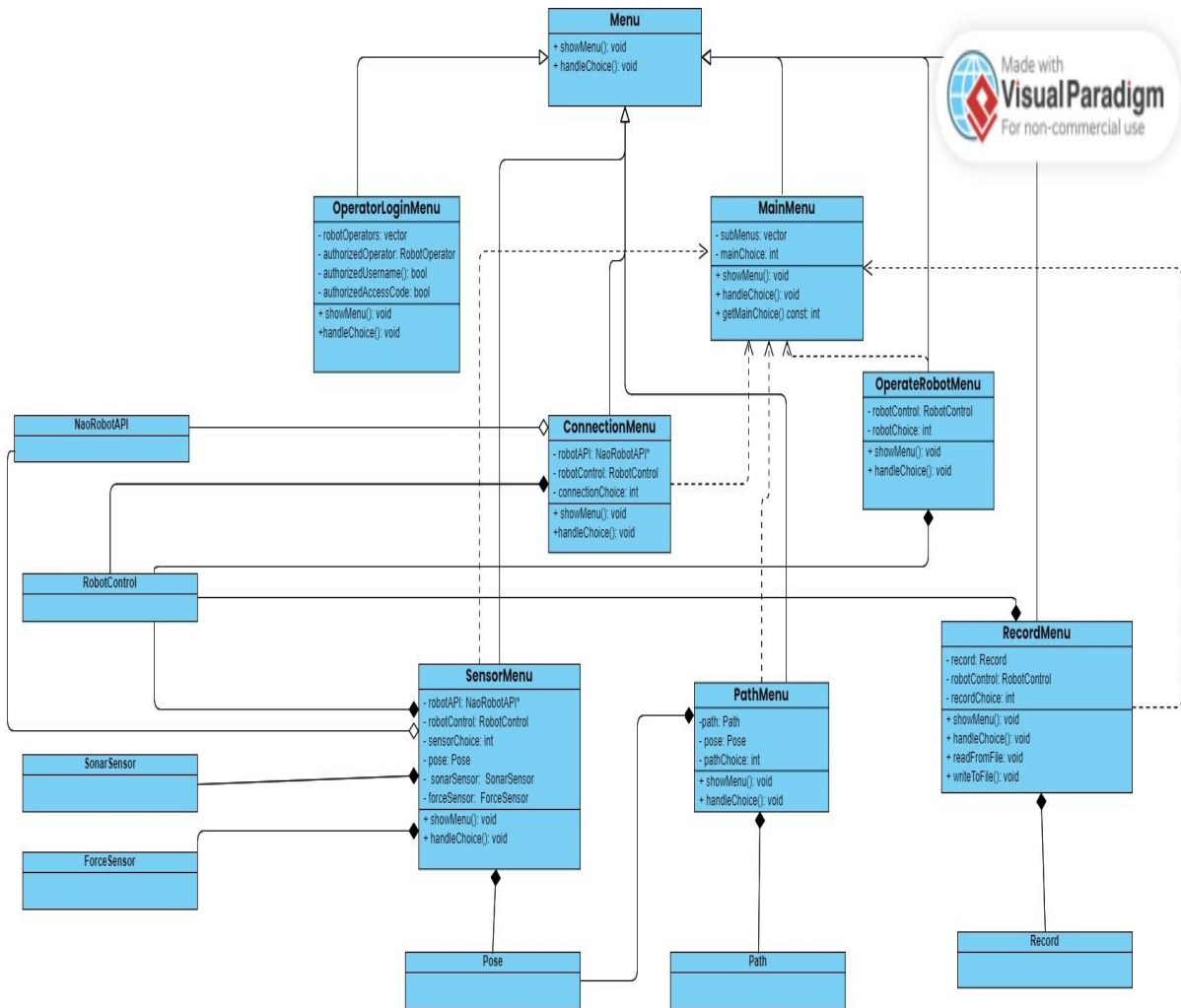
    ///! Swapping digits.
    ///! @param encryptedNumber is an integer variable to store the encrypted number.
    int encryptedNumber = digit3 * 1000 + digit4 * 100 + digit1 * 10 + digit2;

    return encryptedNumber;
}
```

2.4. Menü Oluşturulması

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

Resim 5: Menü 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 çalışma ekran görüntüsü bulunmaktadır.

Resim 6: Uygulama Login Ekranı Görüntüsü

```
-----
Welcome to Webots Robot Control Application
-----

Operator Login Menu
-----
Please enter your username and access code to use the app!

Please enter your username: gur26
Please enter your access code: ****

Operator authorized. Access granted.
Operator Informations:
Name: Gurkan
Surname: Kayi
Access State: 1
```

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

```
=== Main Menu ===
1. Connection
2. Operate Robot
3. Sensor
4. Path
5. Record
6. Exit
Enter your choice: _
```

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

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

```
=== Connection Menu ===
1. Connect Robot
2. Disconnect Robot
3. Back to Main Menu
Enter your choice: _
```

Connection menüsü kullanımı aşağıda yer almaktadır.

Resim 9: Connection Menüsü Kullanımı Ekran Çıktısı

```
Enter your choice: 1
Error: Belirtilen mod³l bulunamad².
(dynamic library)
Error: failed to load C:/Program Files/Webots/resources/project

Robot connected.

=== Connection Menu ===
1. Connect Robot
2. Disconnect Robot
3. Back to Main Menu
Enter your choice: 2

Robot disconnected.

=== Connection Menu ===
1. Connect Robot
2. Disconnect Robot
3. Back to Main Menu
Enter your choice: 2

Robot is already disconnected. Please connect the robot first.

=== Connection Menu ===
1. Connect Robot
2. Disconnect Robot
3. Back to Main Menu
Enter your choice:
```

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)

```
Robot connected.

=== Connection Menu ===
1. Connect Robot
2. Disconnect Robot
3. Open Access
4. Close Access
5. Back to Main Menu
Enter your choice: 3

Enter access code: 1234
Access granted.

=== Connection Menu ===
1. Connect Robot
2. Disconnect Robot
3. Open Access
4. Close Access
5. Back to Main Menu
Enter your choice: 4

Enter access code: 1234
Access closed.
```

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

Resim 11: Operate Robot Menüsü Ekran Görüntüsü

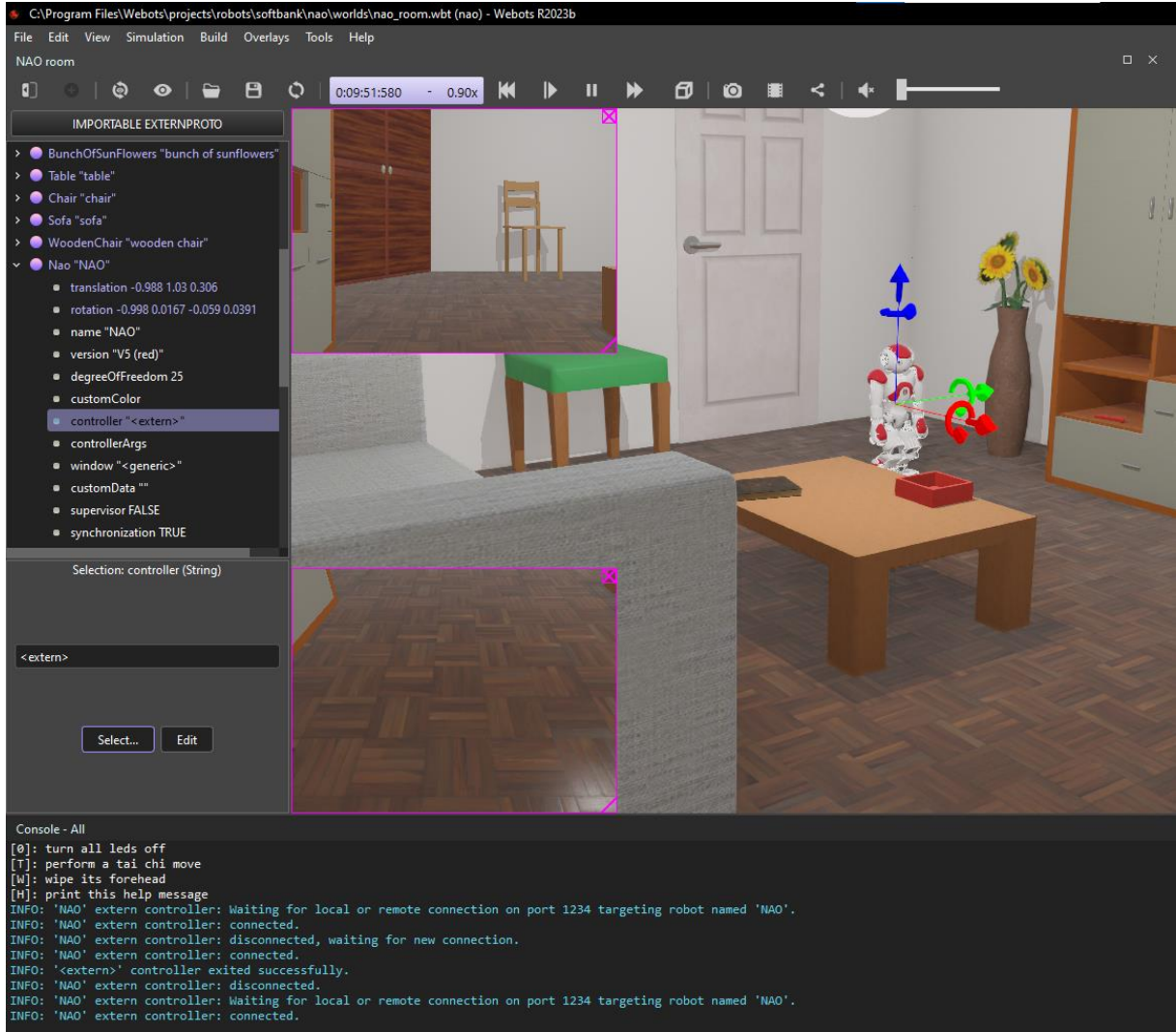
```
=== Operate Robot Menu ===
1. Move Forward
2. Move Backward
3. Move Left
4. Move Right
5. Turn Left
6. Turn Right
7. Stop
8. Back to Main Menu
Enter your choice: _
```

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

Resim 12: Forward Seçeneği Ekran Görüntüsü

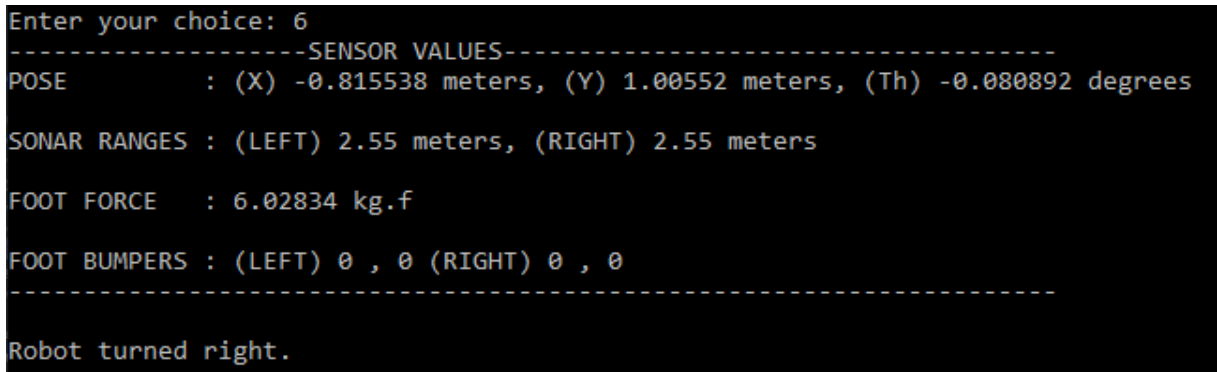
```
Enter your choice: 1
-----SENSOR VALUES-----
POSE          : (X) -1.74418 meters, (Y) 1.00251 meters, (Th) 2.36537 degrees
SONAR RANGES  : (LEFT) 2.55 meters, (RIGHT) 2.55 meters
FOOT FORCE     : 5.7373 kg.f
FOOT BUMPERS  : (LEFT) 0 , 0 (RIGHT) 0 , 0
-----
Robot moved forward.
```

Resim 13: Webots Simülasyon Robot İleri Hareket Ekran Görüntüsü

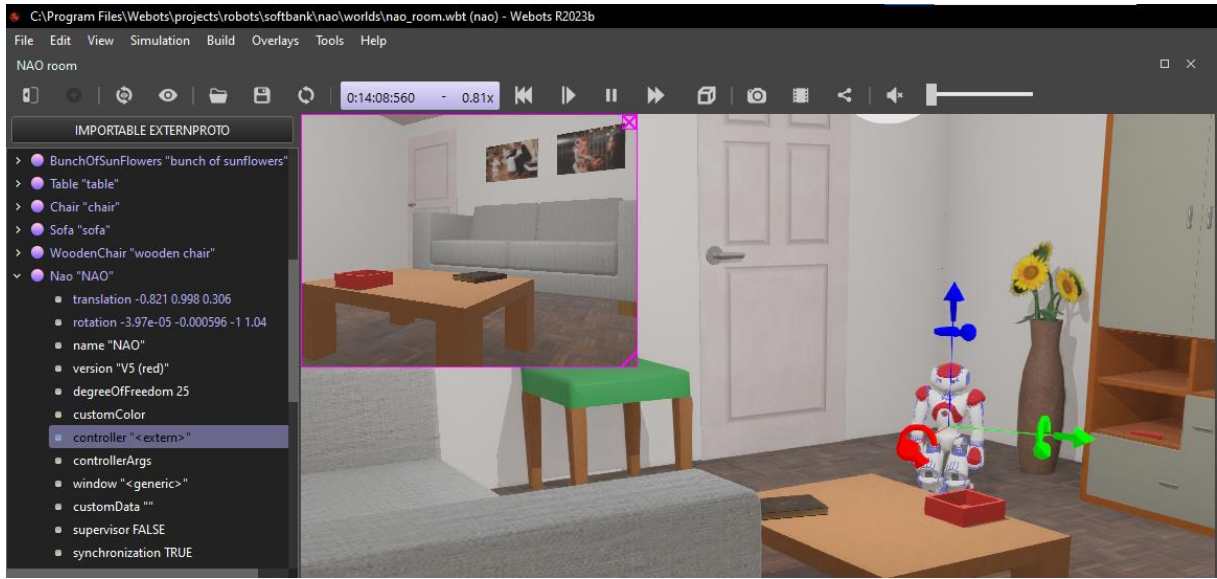


Ya da sağa dön seçildiğinde oluşan durum aşağıda yer almaktadır.

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)

```
=== Sensor Menu ===
1. Display Pose
2. Display Sonar Ranges
3. Display Foot Force
4. Display Foot Bumpers
5. Back to Main Menu
Enter your choice:
```

Resim 17: Sensör Menüsü Kullanımı Ekran Görüntüsü

```
Enter your choice: 1
Pose: (X) -0.820724 meters, (Y) 0.99777 meters, (Th) -37.935 degrees

=== Sensor Menu ===
1. Display Pose
2. Display Sonar Ranges
3. Display Foot Force
4. Display Foot Bumpers
5. Back to Main Menu
Enter your choice: 2
Sonar Ranges: (LEFT) 2.55 meters, (RIGHT) 2.55 meters

=== Sensor Menu ===
1. Display Pose
2. Display Sonar Ranges
3. Display Foot Force
4. Display Foot Bumpers
5. Back to Main Menu
Enter your choice: 3
Foot Force: 6.05775 kg.f
```


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

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

```
=== Sensor Menu ===
1. Display Pose
2. Display Sonar Ranges
3. Display Foot Force
4. Display Foot Bumpers
5. Record Sensor
6. Print Sensor
7. Back to Main Menu
Enter your choice: 6

Enter the sensor type to print: dssSS
Error: Sensor type not found.
```

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)

```
=== Path Menu ===
1. Add Pose
2. Print Path
3. Get Pose at Index
4. Remove Pose at Index
5. Insert Pose at Index
6. Back to Main Menu
```

Resim 20: Path Menü Kullanımı Ekran Görüntüsü

```
=== Path Menu ===
1. Add Pose
2. Print Path
3. Get Pose at Index
4. Remove Pose at Index
5. Insert Pose at Index
6. Back to Main Menu
Enter your choice: 1

Enter x for the pose: -1.27686
Enter y for the pose: 1.00851
Enter th for the pose: 0.0496028

Pose added to the path.

=== Path Menu ===
1. Add Pose
2. Print Path
3. Get Pose at Index
4. Remove Pose at Index
5. Insert Pose at Index
6. Back to Main Menu
Enter your choice: 2

Path printed:
(-1.27686, 1.00851, 0.0496028)
```

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

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

```
=== Path Menu ===
1. Add Pose
2. Print Path
3. Get Pose at Index
4. Remove Pose at Index
5. Insert Pose at Index
6. Add to Path
7. Clear Path
8. Back to Main Menu
Enter your choice: 6

Path size before adding: 0
Path size after adding: 1

=== Path Menu ===
1. Add Pose
2. Print Path
3. Get Pose at Index
4. Remove Pose at Index
5. Insert Pose at Index
6. Add to Path
7. Clear Path
8. Back to Main Menu
Enter your choice: 7

Path size before clear: 1
Path size after clear: 0
```

Record menüsünde 1. aşamada dosyadan okuma ve dosyaya yazdırma işlemleri yapılabilir. Read From File denilerek önceden oluşturulmuş dosyadan okuma 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)

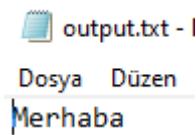
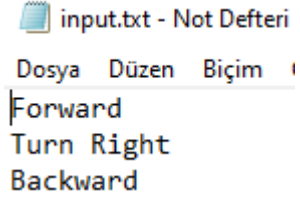
```
=== Record Menu ===
1. Read From File
2. Write To File
3. Back to Main Menu
Enter your choice: 1

Enter the file name to read from: input.txt
Read from file: Forward
Read from file: Turn Right
Read from file: Backward

=== Record Menu ===
1. Read From File
2. Write To File
3. Back to Main Menu
Enter your choice: 2

Enter the file name to write to: output.txt
Enter data to write to the file: Merhaba

Write to file successful.
```



Program durdurulduğunda ya da konsol ekranı kapatıldığında robot bağlantısı kesilmektedir.

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

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

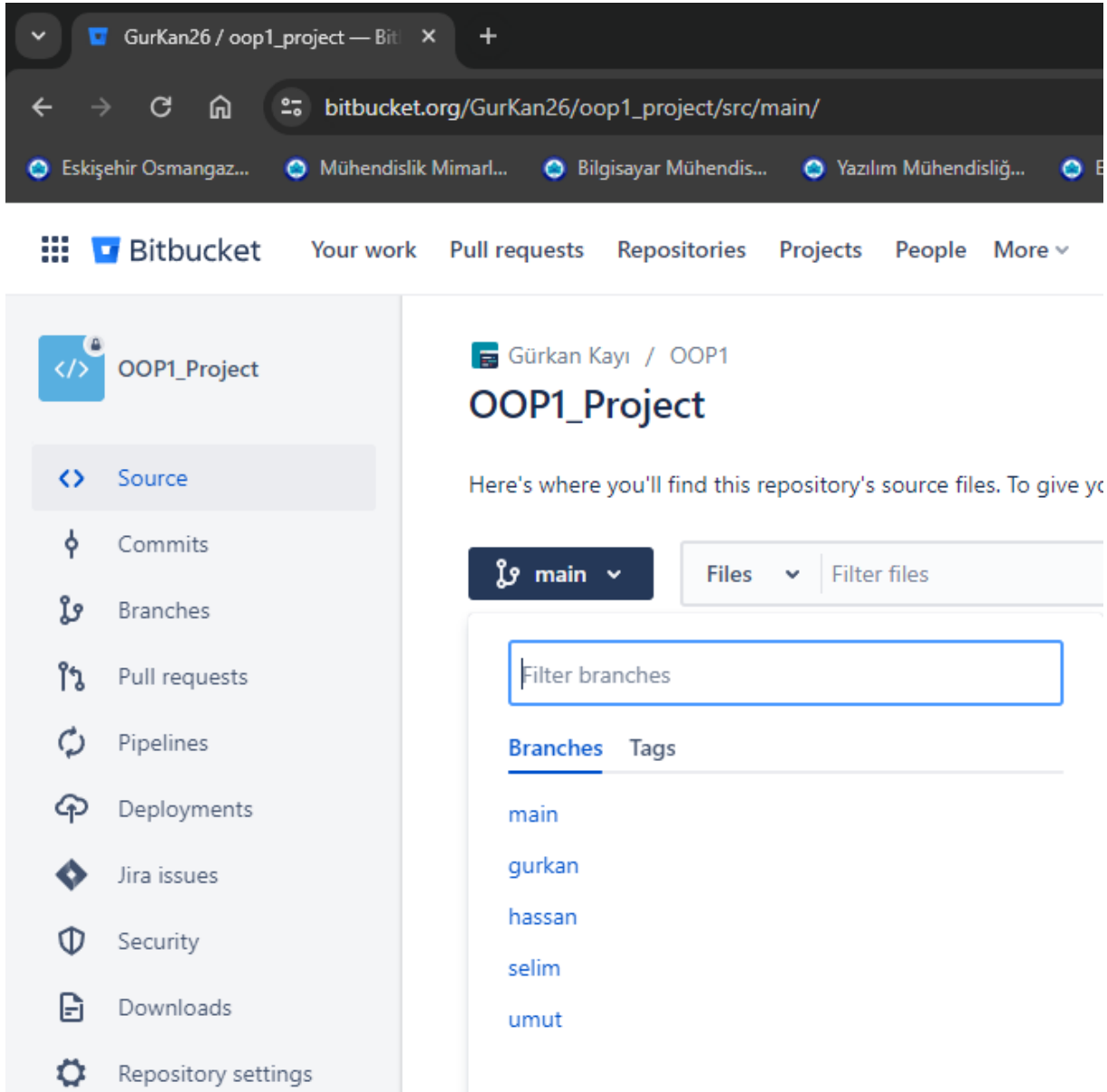
```
=== Record Menu ===
1. Read From File
2. Write To File
3. Record Path
4. Back to Main Menu
Enter your choice: 3

Enter the file name to write to: out.txt
Path recorded to file successfully.
```

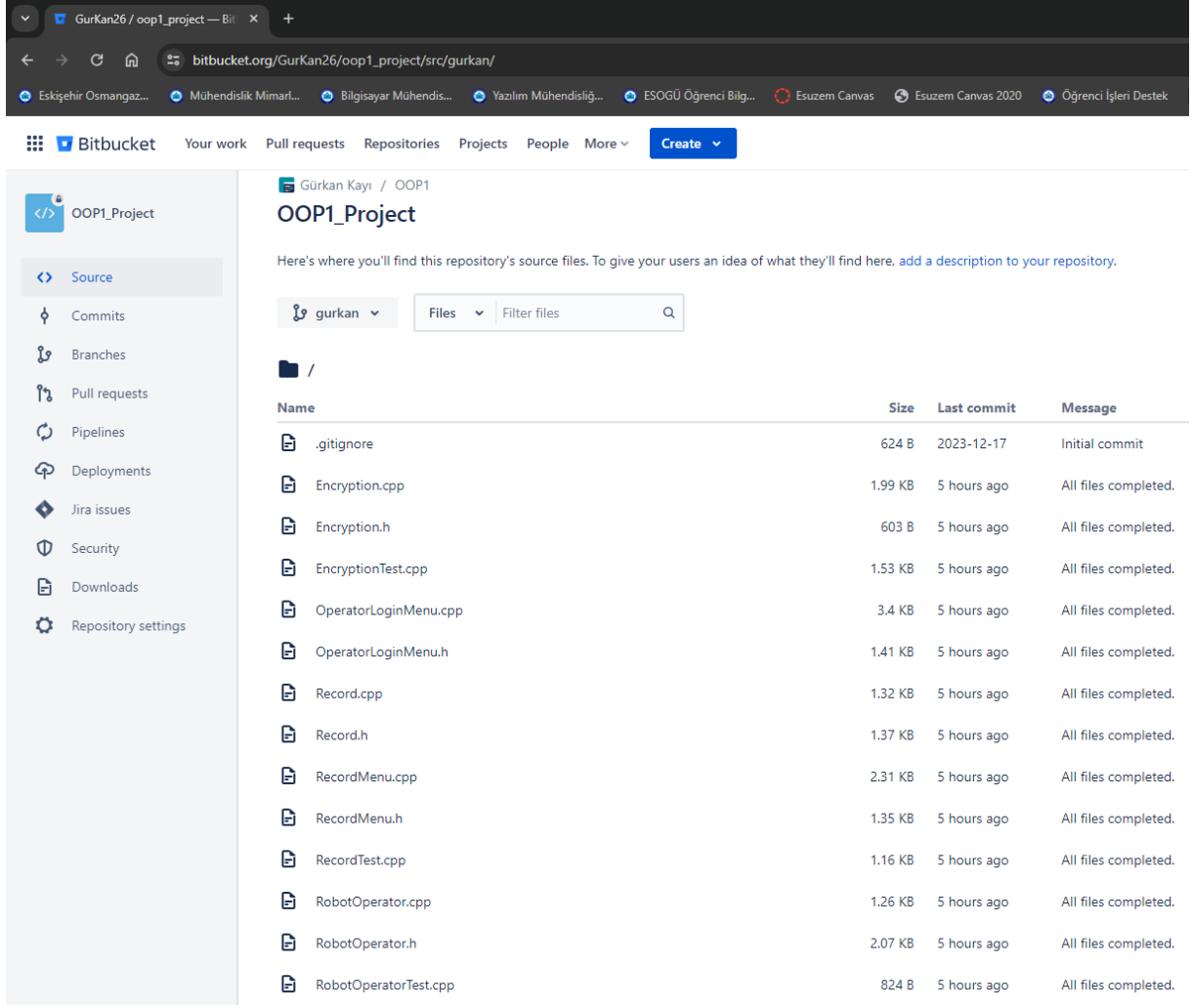
2.6. Bitbucket Kullanımı

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

oop1_project

Main Page	Classes ▾	Files ▾	
Encryption Class Reference			
#include <Encryption.h>			
Static Public Member Functions			
static int encrypt (int number) Encrypts the given number.			
static int decrypt (int number) Decrypts the given number.			
Member Function Documentation			
◆ decrypt()			
int Encryption::decrypt (int number)			
Decrypts the given number.			
Parameters			
digit1 is an integer variable to store the first digit of the number.			
digit2 is an integer variable to store the second digit of the number.			
digit3 is an integer variable to store the third digit of the number.			
digit4 is an integer variable to store the fourth digit of the number.			
Extracting digits.			
Decryption algorithm.			
Swapping digits.			
Parameters			
decryptedNumber is an integer variable to store the decrypted number.			

3. Sonular ve Deęerlendirme

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 yürütölmeye özen gösterilmiştir. Proje verilmeden önce grup oluşturulması aşamasında grup üyeleri arasında rahatça iletişim sağlanabilmesi için mesajlaşma uygulaması üzerinden grup kurulmuştur. Proje verildięi 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 olduęu 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 olduęu görölmektedir. Bir taraftan nesne tabanlı programların 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 simölasyon ile bağlantı sıkıntıları yaşanabilmektedir. Yine de bu hata genel olarak uygulamanın kullanılmasına engel bir durum teşkil etmemektedir.

Genel olarak proje sayesinde önemli kazanımlar sağlandıęı söylenebilir.