

Artırılmış Gerçeklik ile Ders Programı Sistemi

School Timetable System With Augmented Reality

Kaan Toraman, Eda Obuz, Hacer Sueda Efe

*Bilişim Sistemleri Mühendisliği
Teknoloji Fakültesi - Kocaeli Üniversitesi
İzmit, Kocaeli /Türkiye*

kaan41tor@gmail.com , edaobuz4@gmail.com , suedaeefe@gmail.com

Özet

Bu rapor Unity'de geliştirilen okullar için bir ders programı sistemi mobil uygulaması projesine ait problem tanımı, yapılan araştırmalar, akış şeması, veritabanı, kullanıcı arayüzü, projenin backendi ve karşılaşılan sorunlar ilgili bilgiler içermektedir. Bu uygulamada ayrıca artırılmış gerçeklik kullanılmıştır, bir sınıfa ait ders programı sınıfın adı AR kamerası aracılığıyla taratıldığında görüntülenebilmektedir. Projede C# PHP ve SQL kullanılmıştır.

Abstract

This report contains information about the problem definition, researches, flowcharts, database, user interface, background of the project and problems encountered for a school timetable system mobile application project developed in Unity. Augmented reality is also used in this application, a class schedule can be viewed when the name of the class is scanned through the AR camera. C#, PHP and SQL were used in the project.

1. Problem Tanımı

Bu projeyi geliştirme amacımız okullardaki ders programı ayarlama sorununa bir çözüm üretmektir. Eğitim kurumlarındaki ders programlarını ayarlamak çoğu zaman zorlu olabilir, öğretmen ve sınıfın müsaitlik durumlarını göz önünde bulundurmak ve çakışan ders saatlerini engellemek oldukça karmaşık bir süreçtir. Klasik yöntemlerle bu süreci yönetmek zor ve zaman alıcı bir hale gelmektedir. Bu yüzden geliştirdiğimiz bu uygulama ile okulların ders programlarını daha etkili bir şekilde düzenlemelerine yardımcı olmayı amaçladık.

Geliştirdiğimiz uygulama, öğretmenlerin ve sınıfların müsaitlik durumlarını takip ederek programda mümkün olan aralıklara ders eklemelerine olanak tanır. Aynı zamana denk gelen dersler için çakışma kontrolü yapar

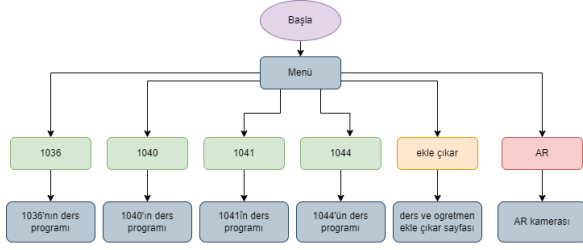
ve izin vermez. Bu sayede, ders programı oluşturulurken bir çok problemin önüne geçilerek daha düzenli bir program oluşturulabilir. Ayrıca proje eklemiş olduğumuz artırılmış gerçeklik teknolojisiyle sınıfların ismi kameradan taratılarak programa görüntülenebilmektedir, bu sayede telefon üzerinden sadece kamera kullanılarak kolay bir şekilde güncel programa erişim sağlanabilmektedir.

2. Yapılan Araştırmalar

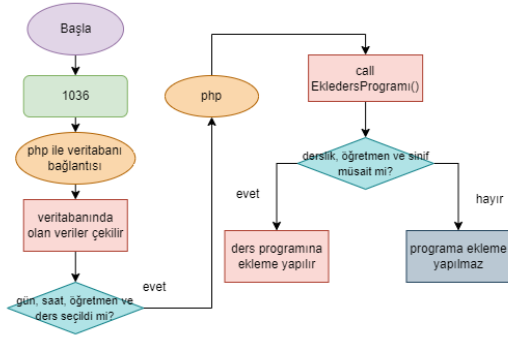
Proje gereklilikleri doğrultusunda geliştirmek istediğimiz sistem için hangi araçları ve hangi veritabanı yönetim sistemini kullanabileceğimizle ilgili kapsamlı bir araştırma süreci gerçekleştirildi. Öncelikle web mi mobil geliştirmek istediğimize karar vermemiz gerekti, bu süreçteki araştırmalarımız sonucunda ve artırılmış gerçeklik kullanma kararımızın kesinleşmesi sebebiyle en uygun geliştirme ortamının Unity olduğuna karar verildi.

Proje gerekliliklerine uygun olarak, graph boyama tekniğini daha uygun bir şekilde kullanabilmek adına bir graph veritabanı seçeneği değerlendirildi. Bu aşamada, başlangıçta popüler bir tercih olan Neo4j kullanılmaya karar verildi fakat bu sistemle ve kullanılan cypher diliyle daha önce çalışmadığımız için konuyla ilgili detaylı bir araştırma süreci gerçekleştirdik. Fakat sonucunda Unity ve Neo4j arasındaki bağlantıyı tam anlamıyla sağlayamadık. Bu zorlukların üstesinden gelmek adına projenin gereksinimlerine daha uygun bir çözüm olarak MySQL veritabanı seçeneği gündeme geldi. MySQL ile ilgili gerekli araştırmalar sonucunda Unity ile MySQL arasındaki bağlantı sağlanarak projenin devamında başarılı bir entegrasyon elde edildi. Bu değişiklik, projenin daha sağlıklı ve verimli bir şekilde ilerlemesine olanak sağladı.

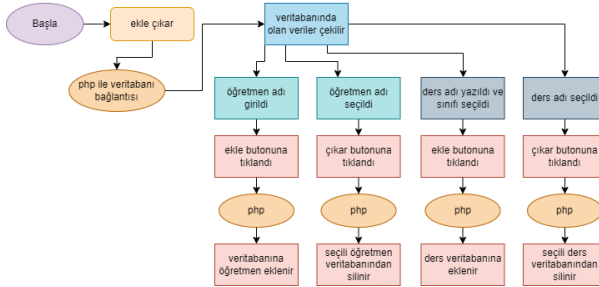
3. Akış şeması



Şekil 1: Uygulama Akış Diyagramı



Şekil 2 : Ders Programı Akış Diyagramı



Şekil 3 : Ders/Öğretmen Ekle Çıkart Akış Diyagramı

4. Kullanıcı Arayüzü (UI)

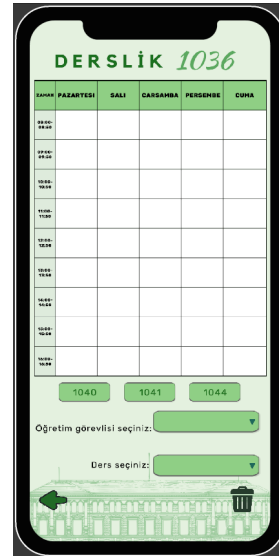
Uygulamanın Ana Sayfasının UI'ı için bir game manager içerisine canvas eklendi ardından canvasın içine image elementi eklendi, bu image elementinin source image'inin tasarımı Canva kullanılarak yapıldı. Resmi unity de kullanabilmek için resim Unity'ye import new asset seçilerek eklendikten sonra texture type sprite olarak ayarlandı. Bu işlemten sonra source image kısmına bu resim sürüklenip bırakıldı. Arka planın ardından butonlar eklendi, butonların UI'ı yine Canva'da tasarlandı internetten background removal tooları kullanılarak bu tasarımlar buton olarak kullanıma uygun hale getirildi.



Şekil 4 : Ana Menü UI

Dersliklerin UI'ı için de benzer bir yaklaşım sergilendi. Canvas'ın üstüne yine bir background image ayarlandı. Ders programı için bir tablo görüntüsü yapıldı bu görüntü eklendi, bu tabloda belirlenen yerlerin her biri için birer buton eklendi bu butonlar kullanılarak tabloya veri ekleme işlemleri gerçekleştirildi. Ardından seçimler için dropdownlar eklendi bu dropdownların arka planları ve ok elementleri internetten hazır asset olarak alındı.

Her dersliğe ait ders programı sayfasından diğer dersliklerin ders programlarına erişimin kolayca sağlanabilmesi amacıyla her sayfaya ders programları arası iletimi sağlayan butonlar eklenmiştir.

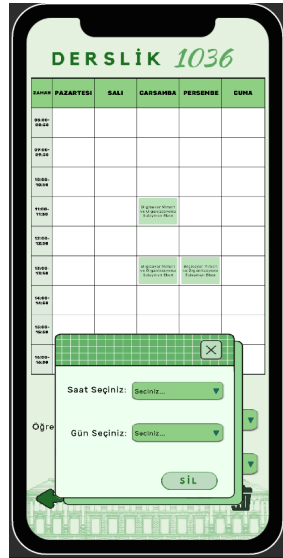


Şekil 5 : Ders Programı UI



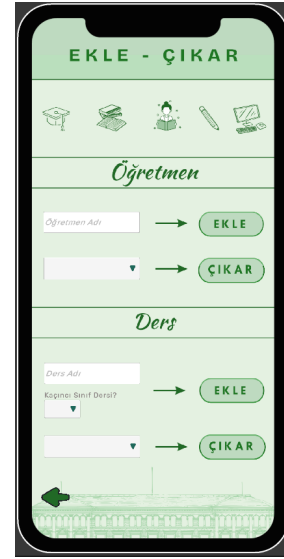
Şekil 6 : Ders Programı UI

Ders programına eklenmiş bir dersi silmek amacıyla ekranın sol alt kısmına bir çöp kutusu ikonu eklendi. Bu ikona tıklandığında bir pop-up aktif olmaktadır. Silinmek istenen dersin saat ve gün bilgileri girilerek silme işlemi tamamlanmaktadır.



Şekil 7 : Ders Programı UI

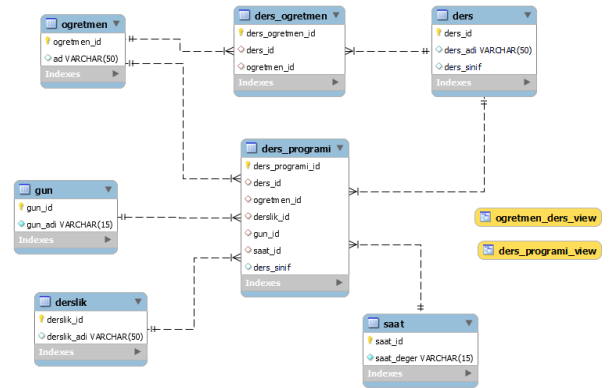
Ekle çıkar sayfası için de ana sayfada yapılan işlemler tekrarlandı, Canvadan arka plan tasarımı yapıldı, eklendi. Butonlar da ana sayfada kullanılan butonlarla aynı şeklindedir.



Şekil 8 : Ders/Öğretmen Ekle Çıkar UI

5. Veritabanı

Projede MySQL veritabanı yönetim sistemi kullanıldı. Oluşturulan veritabanında temel olarak 7 adet tablo 2 adet view bulunmaktadır, veritabanı içerisinde ders_programı tablosuna veri eklemek için ayrıca bir stored procedure yazılmıştır. Veritabanına ait ilişki diyagramı şekilde verilmiştir.



Şekil 9 : Veritabanı İlişki Diyagramı

Öğretmen tablosunda öğretmene ait id ve isim verileri tutulmakta; ders tablosunda id, dersin adı ve dersin ait olduğu sınıf (yani 1. sınıf, 2. sınıf dersi anlamında) verileri tutulmakta; derslikte yine benzer şekilde id ve dersliğin adı tutulmakta. Normalde ders tablosu ve öğretmen tablosunu bir öğretmen_id foreign key'i aracılığıyla bağlamak düşünüldü böylece ders tablosunda dersi veren öğretmenin id'si tutulacaktı fakat üniversite ortamında bir dersi sadece bir öğretmen vermediğinden ötürü aralarında bire bir bir ilişki kurmak yanlış olurdu, bu yüzden ayrıca ders_ogretmen isminde bir tablo daha oluşturulup dersi hangi öğretmenin verdiğine ait bilgiler

burada tutuldu bire çok ilişki sağlanmış oldu. Ders programı tablosunda ise bütün bu bilgiler birleştirildi, dersin adı, dersi veren öğretmen, dersin verildiği derslik, dersin ait olduğu sınıf, ayrıca dersin verildiği gün ve saat bilgileri. Bu gün ve saat bilgileri için de 2 adet tablo daha oluşturuldu, başlangıçta bu verileri ders_programı tablosunda birer string şeklinde tutmak denendi, fakat bu durumda ders programını gün veya saate göre sıralamak mümkün olmadı, dolayısıyla ayrı tablolarda bu bilgileri id'ler ile tutup ders programını çağırırken order by komutuyla id'leri sıralamak mümkün kılındı.

Başta bahsedilen procedure'un yazılma amacı ders programına ders eklerken bazı müsaitlik durumlarını kontrol etmesidir. Procedure sırasıyla öğretmen, derslik ve sınıfın müsaitlik durumunu kontrol etmek üzere ayarlanmıştır. Yani öğretmenin ders eklenmek istenen gün ve saatte başka bir dersi varsa procedure buna izin vermemektedir, aynı şekilde diyelim ki 1. sınıfların pazartesi saat 9 da dersi varsa başka bir öğretmen o gün o saatte kendi müsaitse ve müsait olan bir dersliği seçse bile 1. sınıflar o zaman diliminde müsait olmadığı için 1. sınıfların programına herhangi bir ekleme yapamamaktadır. Eğer 3 durum da müsaitlik sağlıyorsa programa bunu eklemektedir.

```
BEGIN
DECLARE ogretmen_count INT;
DECLARE derslik_count INT;
DECLARE sinif_count INT;

-- ilgili öğretmenin belirli bir günde ve saatte başka dersi var mı kontrol et
SELECT COUNT(*) INTO ogretmen_count
FROM ders_programi
WHERE ogretmen_id = p_ogretmen_id AND gun_id = p_gun_id AND saat_id = p_saat_id;

-- ilgili dersliğin belirli bir günde ve saatte başka dersi var mı kontrol et
SELECT COUNT(*) INTO derslik_count
FROM ders_programi
WHERE derslik_id = p_derslik_id AND gun_id = p_gun_id AND saat_id = p_saat_id;

-- ilgili sınıfın belirli bir günde ve saatte başka dersi var mı kontrol et
SELECT COUNT(*) INTO sinif_count
FROM ders_programi dp
JOIN ders d ON dp.ders_id = d.ders_id
WHERE d.ders_sinif = (SELECT ders_sinif FROM ders WHERE ders_id = p_ders_id)
AND dp.gun_id = p_gun_id AND dp.saat_id = p_saat_id;

-- Her üç kontrol de başarılıysa yeni ders programını ekle
IF ogretmen_count = 0 AND derslik_count = 0 AND sinif_count = 0 THEN
INSERT INTO ders_programi (ders_id, derslik_id, ogretmen_id, gun_id, saat_id)
VALUES (p_ders_id, p_derslik_id, p_ogretmen_id, p_gun_id, p_saat_id);
SELECT 'ders programına eklendi.' AS result;
ELSE
SELECT 'çakışan bir veri mevcut.' AS result;
END IF;
END //
```

Şekil 10 : Müsaitlik kontrolü için yazılan prosedür

2 adet de view tablosu oluşturulmuştur. Bu tablolarda gösterilen verilerin bazı join işlemleri kullanılarak birden fazla tablodan çekilmesi gerektiği için, her seferinde bu uzun sorguların yazılması önlenmesi için bu veriler view haline getirilmiştir, select * from view şeklinde veriler kolaylıkla görüntülenebilmektedir. Bunlardan biri ders_programi_view, normal ders_programi tablosuna veri girilirken foreign_keyler kullanıldığından dolayı idler aracılığıyla veri giriliyor bu durumda tablo çekildiğinde karşımıza sadece sayılar geliyor, bu

istenmeyen durumu ortadan kaldırmak için diğer tablolardan bu idlere ait isimler çekilerek daha okunabilir bir tablo oluşturulmuştur. Benzer şekilde ogretmen_ders_view view'ı da hangi öğretmenin hangi dersi verdiğini göstermek içindir.

```
168 CREATE VIEW ders_programi_view AS
169 SELECT
170     gun.gun_adl AS gun,
171     saat.saat_deger AS saat,
172     ders.ders_adl AS ders,
173     ders.ders_sinif AS sinif,
174     derslik.derslik_adl AS derslik,
175     ogretmen.ad AS ogretmen
176 FROM
177     ders_programi
178 JOIN gun ON ders_programi.gun_id = gun.gun_id
179 JOIN saat ON ders_programi.saat_id = saat.saat_id
180 JOIN ders ON ders_programi.ders_id = ders.ders_id
181 JOIN derslik ON ders_programi.derslik_id = derslik.derslik_id
182 JOIN ogretmen ON ders_programi.ogretmen_id = ogretmen.ogretmen_id
183 ORDER BY
184     ders_programi.gun_id,
185     ders_programi.saat_id;
```

gun	saat	ders	sınıf	derslik	ogretmen
Pazartesi	08:00	Veritabanı Yönetim Sistemleri	2	1036	Serdar Solak
Pazartesi	11:00	Ayrık Matematik	3	1040	Süleyman Eken
Salı	08:00	Algoritma ve Programlama	1	1040	Yavuz Selim Fathoglu
Carsamba	09:00	Bilgisayar Mimari ve Organizasyonu	3	1040	Hall Yigit
Perembe	12:00	Yazılım Geliştirme Lab.	3	1041	Süleyman Eken

Şekil 11 : Ders programı için view tablosu

6. Kodlama

6.1 AR ile Görsel Tarama

Proje, Unity oyun motorunda mobil cihazlar için artırılmış gerçeklik (AR) kullanılarak geliştirildi. Bu geliştirme sürecinde "Vuforia Engine" AR teknolojisi kullanıldı. Vuforia Engine, görüntü tanıma, nesne tanıma ve işaretlenmiş görsellerin algılanması gibi özelliklere sahiptir. Bu özellikler, kullanıcıların AR uyumlu cihazlarını kullanarak gerçek dünya ortamında dijital içerikler oluşturmalarına veya görsel etkileşimlerde bulunmasına olanak sağlar. Kullanıcılar, Vuforia'nın sağladığı bu özellikler sayesinde çevrelerini zenginleştirebilir ve interaktif deneyimler yaşayabilirler.



Şekil 12 : Vuforia

Projede, Vuforia veritabanı içerisinden derslikleri temsil etmek üzere görseller belirlendi. Bu görseller, Unity ortamına aktarıldıktan sonra her biri için "ImageTarget"lar oluşturuldu. Bu Image Target'ların içlerine üç boyutlu dikdörtgenler eklendi ve tarama işlemi sonrasında kullanıcıların karşısına çıkacaklar. Bu düzenleme, ilgili dersliklerin programlarına yönlendirme yapabilmek amacıyla gerçekleştirildi. Bu yönlendirmeler

sayesinde, kullanıcılar istedikleri derslikle ilgili bilgilere ulaşabilecekler.



Şekil 13: Derslik görseli tarama

6.2 Veritabanı ile Bağlantı

6.2.A PHP Dosyaları

Unity oyun motoruna bir SQL veritabanı entegrasyonu gerçekleştirmek, projenin veri yönetimi için güçlü ve esnek bir altyapı oluşturmayı hedeflemektedir. Ancak, bu entegrasyon çeşitli zorlukları da beraberinde getirmektedir. İlk olarak, Unity'nin hafif ve performansa odaklı bir oyun motorudur. SQL gibi ağır bir veritabanı yönetim sisteminin (DBMS) kullanılması, projede performans düşüşlerine sebep olabilir. Ayrıca, SQL veritabanları çoğunlukla sunucu-tabanlıdır, bu da projenin bağımsızlığını etkiler ve ek sunucu gereksinimini ortaya çıkarır. Bütün bu durumlara rağmen projede PHP kullanılarak Apache bir sunucu üzerinden veritabanına bağlantı sağlandı.

Veritabanı bağlantısı PHP yardımıyla sağlanmıştır. Veritabanı üzerinden yapmak istenilen bütün oluşturma, silme, güncelleme ve okuma işlemleri PHP kodları ile sağlanmıştır. Proje içerisinde 16 adet PHP kod dosyası vardır. Bunlardan bir tanesi veritabanı ile olan bağlantıyı, 4 tanesi veri eklemeyi, 3 tanesi veri silmeyi ve geriye kalan 8 tanesi ise veri okumayı sağlamaktadır.

Veritabanıyla bağlantı kurulmasını sağlayan PHP kod dosyası içerisinde \$servername, \$DBusername, \$DBpassword ve \$dbname değişkenleri bulunur ve bunlar sırasıyla veritabanı sunucusu adı, kullanıcı adı, şifre ve bağlanılacak olan veritabanının adını içerir. \$conn değişkeni, mysqli sınıfından bir nesneye atanarak veritabanı bağlantısı oluşturur. Eğer bağlantı hatası oluşursa, die() fonksiyonu ile birlikte bağlantı hatası mesajı ekrana yazdırılır ve program sonlandırılır.

```
C:\Users> kaan4 > OneDrive > Belgeler > GitHub > YazLab1-2 > php > veritabanliAR > conn.php
1  <?php
2  $servername="localhost";
3  $DBusername="root";
4  $DBpassword="";
5  $dbname="dersprogrami";
6
7  $conn=new mysqli($servername, $DBusername,$DBpassword,$dbname);
8
9  if($conn-> connect_error)
10 {
11     die("1". $conn->connect_error);
12 }
13
```

Şekil 14: conn.php Dosyası

Verileri veritabanına eklemeyi sağlayan PHP kod dosyaları içerisinde; ilk olarak, conn.php dosyasıyla bir veritabanı bağlantısı oluşturulur. Unity C# dosyalarından gelen veriler "\$_POST" ile alınır ve ilgili değişkenlere atanır. Sonrasında, \$sql adında bir SQL sorgusu oluşturulur ve "INSERT INTO" komutuyla belirtilen tablolara C# dosyalarından gelen değerler eklenir. Try-catch yapısı, olası hataları ele alır. \$sql sorgusu çalıştırılır ve eğer bir hata oluşursa, if koşulu içinde işlenir ve hata mesajı olarak geri döner. Bağlantı hatası veya SQL sorgusunun başarısız olması durumunda geri dönen mesajları ekrana yazdırılır. Başarılı bir şekilde kayıt eklenirse, "Başarılı" mesajı geri döndürülür ve veritabanı bağlantısı kapatılır.

```
C:\Users> kaan4 > OneDrive > Belgeler > GitHub > YazLab1-2 > php > veritabanliAR > dersEkle.php
1  <?php
2  require "conn.php";
3
4
5  $dersAdi=$_POST["ders_adi"];
6  $dersSinif=$_POST["ders_sinif"];
7
8
9  $sql="INSERT INTO ders(ders_adi,ders_sinif)VALUES('".$dersAdi."','".$dersSinif."')";
10
11 |
12 |
13 |
14 |
15 |
16 |
17 |
18 |
19 |
20 |
21 |
22 |
23 |
24 |
25 |
26 |
27 |
28 |
29 |
30 |
31 |
32 |
33 |
34 |
35 |
36 |
37 |
38 |
39 |
40 |
41 |
42 |
43 |
44 |
45 |
46 |
47 |
48 |
49 |
50 |
51 |
52 |
53 |
54 |
55 |
56 |
57 |
58 |
59 |
60 |
61 |
62 |
63 |
64 |
65 |
66 |
67 |
68 |
69 |
70 |
71 |
72 |
73 |
74 |
75 |
76 |
77 |
78 |
79 |
80 |
81 |
82 |
83 |
84 |
85 |
86 |
87 |
88 |
89 |
90 |
91 |
92 |
93 |
94 |
95 |
96 |
97 |
98 |
99 |
100 |
101 |
102 |
103 |
104 |
105 |
106 |
107 |
108 |
109 |
110 |
111 |
112 |
113 |
114 |
115 |
116 |
117 |
118 |
119 |
120 |
121 |
122 |
123 |
124 |
125 |
126 |
127 |
128 |
129 |
130 |
131 |
132 |
133 |
134 |
135 |
136 |
137 |
138 |
139 |
140 |
141 |
142 |
143 |
144 |
145 |
146 |
147 |
148 |
149 |
150 |
151 |
152 |
153 |
154 |
155 |
156 |
157 |
158 |
159 |
160 |
161 |
162 |
163 |
164 |
165 |
166 |
167 |
168 |
169 |
170 |
171 |
172 |
173 |
174 |
175 |
176 |
177 |
178 |
179 |
180 |
181 |
182 |
183 |
184 |
185 |
186 |
187 |
188 |
189 |
190 |
191 |
192 |
193 |
194 |
195 |
196 |
197 |
198 |
199 |
200 |
201 |
202 |
203 |
204 |
205 |
206 |
207 |
208 |
209 |
210 |
211 |
212 |
213 |
214 |
215 |
216 |
217 |
218 |
219 |
220 |
221 |
222 |
223 |
224 |
225 |
226 |
227 |
228 |
229 |
230 |
231 |
232 |
233 |
234 |
235 |
236 |
237 |
238 |
239 |
240 |
241 |
242 |
243 |
244 |
245 |
246 |
247 |
248 |
249 |
250 |
251 |
252 |
253 |
254 |
255 |
256 |
257 |
258 |
259 |
260 |
261 |
262 |
263 |
264 |
265 |
266 |
267 |
268 |
269 |
270 |
271 |
272 |
273 |
274 |
275 |
276 |
277 |
278 |
279 |
280 |
281 |
282 |
283 |
284 |
285 |
286 |
287 |
288 |
289 |
290 |
291 |
292 |
293 |
294 |
295 |
296 |
297 |
298 |
299 |
300 |
301 |
302 |
303 |
304 |
305 |
306 |
307 |
308 |
309 |
310 |
311 |
312 |
313 |
314 |
315 |
316 |
317 |
318 |
319 |
320 |
321 |
322 |
323 |
324 |
325 |
326 |
327 |
328 |
329 |
330 |
331 |
332 |
333 |
334 |
335 |
336 |
337 |
338 |
339 |
340 |
341 |
342 |
343 |
344 |
345 |
346 |
347 |
348 |
349 |
350 |
351 |
352 |
353 |
354 |
355 |
356 |
357 |
358 |
359 |
360 |
361 |
362 |
363 |
364 |
365 |
366 |
367 |
368 |
369 |
370 |
371 |
372 |
373 |
374 |
375 |
376 |
377 |
378 |
379 |
380 |
381 |
382 |
383 |
384 |
385 |
386 |
387 |
388 |
389 |
390 |
391 |
392 |
393 |
394 |
395 |
396 |
397 |
398 |
399 |
400 |
401 |
402 |
403 |
404 |
405 |
406 |
407 |
408 |
409 |
410 |
411 |
412 |
413 |
414 |
415 |
416 |
417 |
418 |
419 |
420 |
421 |
422 |
423 |
424 |
425 |
426 |
427 |
428 |
429 |
430 |
431 |
432 |
433 |
434 |
435 |
436 |
437 |
438 |
439 |
440 |
441 |
442 |
443 |
444 |
445 |
446 |
447 |
448 |
449 |
450 |
451 |
452 |
453 |
454 |
455 |
456 |
457 |
458 |
459 |
460 |
461 |
462 |
463 |
464 |
465 |
466 |
467 |
468 |
469 |
470 |
471 |
472 |
473 |
474 |
475 |
476 |
477 |
478 |
479 |
480 |
481 |
482 |
483 |
484 |
485 |
486 |
487 |
488 |
489 |
490 |
491 |
492 |
493 |
494 |
495 |
496 |
497 |
498 |
499 |
500 |
501 |
502 |
503 |
504 |
505 |
506 |
507 |
508 |
509 |
510 |
511 |
512 |
513 |
514 |
515 |
516 |
517 |
518 |
519 |
520 |
521 |
522 |
523 |
524 |
525 |
526 |
527 |
528 |
529 |
530 |
531 |
532 |
533 |
534 |
535 |
536 |
537 |
538 |
539 |
540 |
541 |
542 |
543 |
544 |
545 |
546 |
547 |
548 |
549 |
550 |
551 |
552 |
553 |
554 |
555 |
556 |
557 |
558 |
559 |
560 |
561 |
562 |
563 |
564 |
565 |
566 |
567 |
568 |
569 |
570 |
571 |
572 |
573 |
574 |
575 |
576 |
577 |
578 |
579 |
580 |
581 |
582 |
583 |
584 |
585 |
586 |
587 |
588 |
589 |
590 |
591 |
592 |
593 |
594 |
595 |
596 |
597 |
598 |
599 |
600 |
601 |
602 |
603 |
604 |
605 |
606 |
607 |
608 |
609 |
610 |
611 |
612 |
613 |
614 |
615 |
616 |
617 |
618 |
619 |
620 |
621 |
622 |
623 |
624 |
625 |
626 |
627 |
628 |
629 |
630 |
631 |
632 |
633 |
634 |
635 |
636 |
637 |
638 |
639 |
640 |
641 |
642 |
643 |
644 |
645 |
646 |
647 |
648 |
649 |
650 |
651 |
652 |
653 |
654 |
655 |
656 |
657 |
658 |
659 |
660 |
661 |
662 |
663 |
664 |
665 |
666 |
667 |
668 |
669 |
670 |
671 |
672 |
673 |
674 |
675 |
676 |
677 |
678 |
679 |
680 |
681 |
682 |
683 |
684 |
685 |
686 |
687 |
688 |
689 |
690 |
691 |
692 |
693 |
694 |
695 |
696 |
697 |
698 |
699 |
700 |
701 |
702 |
703 |
704 |
705 |
706 |
707 |
708 |
709 |
710 |
711 |
712 |
713 |
714 |
715 |
716 |
717 |
718 |
719 |
720 |
721 |
722 |
723 |
724 |
725 |
726 |
727 |
728 |
729 |
730 |
731 |
732 |
733 |
734 |
735 |
736 |
737 |
738 |
739 |
740 |
741 |
742 |
743 |
744 |
745 |
746 |
747 |
748 |
749 |
750 |
751 |
752 |
753 |
754 |
755 |
756 |
757 |
758 |
759 |
760 |
761 |
762 |
763 |
764 |
765 |
766 |
767 |
768 |
769 |
770 |
771 |
772 |
773 |
774 |
775 |
776 |
777 |
778 |
779 |
780 |
781 |
782 |
783 |
784 |
785 |
786 |
787 |
788 |
789 |
790 |
791 |
792 |
793 |
794 |
795 |
796 |
797 |
798 |
799 |
800 |
801 |
802 |
803 |
804 |
805 |
806 |
807 |
808 |
809 |
810 |
811 |
812 |
813 |
814 |
815 |
816 |
817 |
818 |
819 |
820 |
821 |
822 |
823 |
824 |
825 |
826 |
827 |
828 |
829 |
830 |
831 |
832 |
833 |
834 |
835 |
836 |
837 |
838 |
839 |
840 |
841 |
842 |
843 |
844 |
845 |
846 |
847 |
848 |
849 |
850 |
851 |
852 |
853 |
854 |
855 |
856 |
857 |
858 |
859 |
860 |
861 |
862 |
863 |
864 |
865 |
866 |
867 |
868 |
869 |
870 |
871 |
872 |
873 |
874 |
875 |
876 |
877 |
878 |
879 |
880 |
881 |
882 |
883 |
884 |
885 |
886 |
887 |
888 |
889 |
890 |
891 |
892 |
893 |
894 |
895 |
896 |
897 |
898 |
899 |
900 |
901 |
902 |
903 |
904 |
905 |
906 |
907 |
908 |
909 |
910 |
911 |
912 |
913 |
914 |
915 |
916 |
917 |
918 |
919 |
920 |
921 |
922 |
923 |
924 |
925 |
926 |
927 |
928 |
929 |
930 |
931 |
932 |
933 |
934 |
935 |
936 |
937 |
938 |
939 |
940 |
941 |
942 |
943 |
944 |
945 |
946 |
947 |
948 |
949 |
950 |
951 |
952 |
953 |
954 |
955 |
956 |
957 |
958 |
959 |
960 |
961 |
962 |
963 |
964 |
965 |
966 |
967 |
968 |
969 |
970 |
971 |
972 |
973 |
974 |
975 |
976 |
977 |
978 |
979 |
980 |
981 |
982 |
983 |
984 |
985 |
986 |
987 |
988 |
989 |
990 |
991 |
992 |
993 |
994 |
995 |
996 |
997 |
998 |
999 |
1000 |
1001 |
1002 |
1003 |
1004 |
1005 |
1006 |
1007 |
1008 |
1009 |
1010 |
1011 |
1012 |
1013 |
1014 |
1015 |
1016 |
1017 |
1018 |
1019 |
1020 |
1021 |
1022 |
1023 |
1024 |
1025 |
1026 |
1027 |
1028 |
1029 |
1030 |
1031 |
1032 |
1033 |
1034 |
1035 |
1036 |
1037 |
1038 |
1039 |
1040 |
1041 |
1042 |
1043 |
1044 |
1045 |
1046 |
1047 |
1048 |
1049 |
1050 |
1051 |
1052 |
1053 |
1054 |
1055 |
1056 |
1057 |
1058 |
1059 |
1060 |
1061 |
1062 |
1063 |
1064 |
1065 |
1066 |
1067 |
1068 |
1069 |
1070 |
1071 |
1072 |
1073 |
1074 |
1075 |
1076 |
1077 |
1078 |
1079 |
1080 |
1081 |
1082 |
1083 |
1084 |
1085 |
1086 |
1087 |
1088 |
1089 |
1090 |
1091 |
1092 |
1093 |
1094 |
1095 |
1096 |
1097 |
1098 |
1099 |
1100 |
1101 |
1102 |
1103 |
1104 |
1105 |
1106 |
1107 |
1108 |
1109 |
1110 |
1111 |
1112 |
1113 |
1114 |
1115 |
1116 |
1117 |
1118 |
1119 |
1120 |
1121 |
1122 |
1123 |
1124 |
1125 |
1126 |
1127 |
1128 |
1129 |
1130 |
1131 |
1132 |
1133 |
1134 |
1135 |
1136 |
1137 |
1138 |
1139 |
1140 |
1141 |
1142 |
1143 |
1144 |
1145 |
1146 |
1147 |
1148 |
1149 |
1150 |
1151 |
1152 |
1153 |
1154 |
1155 |
1156 |
1157 |
1158 |
1159 |
1160 |
1161 |
1162 |
1163 |
1164 |
1165 |
1166 |
1167 |
1168 |
1169 |
1170 |
1171 |
1172 |
1173 |
1174 |
1175 |
1176 |
1177 |
1178 |
1179 |
1180 |
1181 |
1182 |
1183 |
1184 |
1185 |
1186 |
1187 |
1188 |
1189 |
1190 |
1191 |
1192 |
1193 |
1194 |
1195 |
1196 |
1197 |
1198 |
1199 |
1200 |
1201 |
1202 |
1203 |
1204 |
1205 |
1206 |
1207 |
1208 |
1209 |
1210 |
1211 |
1212 |
1213 |
1214 |
1215 |
1216 |
1217 |
1218 |
1219 |
1220 |
1221 |
1222 |
1223 |
1224 |
1225 |
1226 |
1227 |
1228 |
1229 |
1230 |
1231 |
1232 |
1233 |
1234 |
1235 |
1236 |
1237 |
1238 |
1239 |
1240 |
1241 |
1242 |
1243 |
1244 |
1245 |
1246 |
1247 |
1248 |
1249 |
1250 |
1251 |
1252 |
1253 |
1254 |
1255 |
1256 |
1257 |
1258 |
1259 |
1260 |
1261 |
1262 |
1263 |
1264 |
1265 |
1266 |
1267 |
1268 |
1269 |
1270 |
1271 |
1272 |
1273 |
1274 |
1275 |
1276 |
1277 |
1278 |
1279 |
1280 |
1281 |
1282 |
1283 |
1284 |
1285 |
1286 |
1287 |
1288 |
1289 |
1290 |
1291 |
1292 |
1293 |
1294 |
1295 |
1296 |
1297 |
1298 |
1299 |
1300 |
1301 |
1302 |
1303 |
1304 |
1305 |
1306 |
1307 |
1308 |
1309 |
1310 |
1311 |
1312 |
1313 |
1314 |
1315 |
1316 |
1317 |
1318 |
1319 |
1320 |
1321 |
1322 |
1323 |
1324 |
1325 |
1326 |
1327 |
1328 |
1329 |
1330 |
1331 |
1332 |
1333 |
1334 |
1335 |
1336 |
1337 |
1338 |
1339 |
1340 |
1341 |
1342 |
1343 |
1344 |
1345 |
1346 |
1347 |
1348 |
1349 |
1350 |
1351 |
1352 |
1353 |
1354 |
1355 |
1356 |
1357 |
1358 |
1359 |
1360 |
1361 |
1362 |
1363 |
1364 |
1365 |
1366 |
1367 |
1368 |
1369 |
1370 |
1371 |
1372 |
1373 |
1374 |
1375 |
1376 |
1377 |
1378 |
1379 |
1380 |
1381 |
1382 |
1383 |
1384 |
1385 |
1386 |
1387 |
1388 |
1389 |
1390 |
1391 |
1392 |
1393 |
1394 |
1395 |
1396 |
1397 |
1398 |
1399 |
1400 |
1401 |
1402 |
1403 |
1404 |
1405 |
1406 |
1407 |
1408 |
1409 |
1410 |
1411 |
1412 |
1413 |
1414 |
1415 |
1416 |
1417 |
1418 |
1419 |
1420 |
1421 |
1422 |
1423 |
1424 |
1425 |
1426 |
1427 |
1428 |
1429 |
1430 |
1431 |
1432 |
1433 |
1434 |
1435 |
1436 |
1437 |
1438 |
1439 |
1440 |
1441 |
1442 |
1443 |
1444 |
1445 |
1446 |
1447 |
1448 |
1449 |
1450 |
1451 |
1452 |
1453 |
1454 |
1455 |
1456 |
1457 |
1458 |
1459 |
1460 |
1461 |
1462 |
1463 |
1464 |
1465 |
1466 |
1467 |
1468 |
1469 |
1470 |
1471 |
1472 |
1473 |
1474 |
1475 |
1476 |
1477 |
1478 |
1479 |
1480 |
1481 |
1482 |
1483 |
1484 |
1485 |
1486 |
1487 |
1488 |
1489 |
1490 |
1491 |
1492 |
1493 |
1494 |
1495 |
1496 |
1497 |
1498 |
1499 |
1500 |
1501 |
1502 |
1503 |
1504 |
1505 |
1506 |
1507 |
1508 |
1509 |
1510 |
1511 |
1512 |
1513 |
1514 |
1515 |
1516 |
1517 |
1518 |
1519 |
1520 |
1521 |
1522 |
1523 |
1524 |
1525 |
1526 |
1527 |
1528 |
1529 |
1530 |
1531 |
1532 |
1533 |
1534 |
1535 |
1536 |
1537 |
1538 |
1539 |
1540 |
1541 |
1542 |
1543 |
1544 |
1545 |
1546 |
1547 |
1548 |
1549 |
1550 |
1551 |
1552 |
1553 |
1554 |
1555 |
1556 |
1557 |
1558 |
1559 |
1560 |
1561 |
1562 |
1563 |
1564 |
1565 |
1566 |
1567 |
1568 |
1569 |
1570 |
1571 |
1572 |
1573 |
1574 |
1575 |
1576 |
1577 |
1578 |
1579 |
1580 |
1581 |
1582 |
1583 |
1584 |
1585 |
1586 |
1587 |
1588 |
1589 |
1590 |
1591 |
1592 |
1593 |
1594 |
1595 |
1596 |
1597 |
1598 |
1599 |
1600 |
1601 |
1602 |
1603 |
1604 |
1605 |
1606 |
1607 |
1608 |
1609 |
1610 |
1611 |
1612 |
1613 |
1614 |
1615 |
1616 |
1617 |
1618 |
1619 |
1620 |
1621 |
1622 |
1623 |
1624 |
1625 |
1626 |
1627 |
1628 |
1629 |
1630 |
1631 |
1632 |
1633 |
1634 |
1635 |
1636 |
1637 |
1638 |
1639 |
1640 |
1641 |
1642 |
1643 |
1644 |
1645 |
1646 |
1647 |
1648 |
1649 |
1650 |
1651 |
1652 |
1653 |
1654 |
1655 |
1656 |
1657 |
1658 |
1659 |
1660 |
1661 |
1662 |
1663 |
1664 |
1665 |
1666 |
1667 |
1668 |
1669 |
1670 |
1671 |
1672 |
1673 |
1674 |
1675 |
1676 |
1677 |
1678 |
1679 |
1680 |
1681 |
1682 |
1683 |
1684 |
1685 |
1686 |
1687 |
1688 |
1689 |
1690 |
1691 |
1692 |
1693 |
1694 |
1695 |
1696 |
1697 |
1698 |
1699 |
1700 |
1701 |
1702 |
1703 |
1704 |
1705 |
1706 |
1707 |
1708 |
1709 |
1710 |
1711 |
1712 |
1713 |
1714 |
1715 |
1716 |
1717 |
1718 |
1719 |
1720 |
1721 |
1722 |
1723 |
1724 |
1725 |
1726 |
1727 |
1728 |
1729 |
1730 |
1731 |
1732 |
1733 |
1734 |
1735 |
1736 |
1737 |
1738 |
1739 |
1740 |
1741 |
1742 |
1743 |
1744 |
1745 |
1746 |
1747 |
1748 |
1749 |
1750 |
1751 |
1752 |
1753 |
1754 |
1755 |
1756 |
1757 |
1758 |
1759 |
1760 |
1761 |
1762 |
1763 |
1764 |
1765 |
1766 |
1767 |
1768 |
1769 |
1770 |
1771 |
1772 |
1773 |
1774 |
1775 |
1776 |
1777 |
1778 |
1779 |
1780 |
1781 |
1782 |
1783 |
1784 |
1785 |
1786 |
1787 |
1788 |
1789 |
1790 |
1791 |
1792 |
1793 |
1794 |
1795 |
1796 |
1797 |
1798 |
1799 |
1800 |
1801 |
1802 |
1803 |
1804 |
1805 |
1806 |
1807 |
1808 |
1809 |
1810 |
1811 |
1812 |
1813 |
1814 |
1815 |
1816 |
1817 |
1818 |
1819 |
1820 |
1821 |
1822 |
1823 |
1824 |
1825 |
1826 |
1827 |
1828 |
1829 |
1830 |
1831 |
1832 |
1833 |
1834 |
1835 |
1836 |
1837 |
1838 |
1839 |
1840 |
1841 |
1842 |
1843 |
1844 |
1845 |
1846 |
1847 |
1848 |
1849 |
1850 |
1851 |
1852 |
1853 |
1854 |
1855 |
1856 |
1857 |
1858 |
1859 |
1860 |
1861 |
1862 |
1863 |
1864 |
1865 |
1866 |
1867 |
1868 |
1869 |
1870 |
1871 |
1872 |
1873 |
1874 |
1875 |
1876 |
1877 |
1878 |
1879 |
1880 |
1881 |
1882 |
1883 |
1884 |
1885 |
1886 |
1887 |
1888 |
1889 |
1890 |
1891 |
1892 |
1893 |
1894 |
1895 |
1896 |
1897 |
1898 |
1899 |
1900 |
1901 |
1902 |
1903 |
1904 |
1905 |
1906 |
1907 |
1908 |
1909 |
1910 |
1911 |
1912 |
1913 |
1914 |
1915 |
1916 |
1917 |
1918 |
1919 |
1920 |
1921 |
1922 |
1923 |
1924 |
1925 |
1926 |
1927 |
1928 |
1929 |
1930 |
1931 |
1932 |
1933 |
1934 |
1935 |
1936 |
1937 |
1938 |
1939 |
1940 |
1941 |
1942 |
1943 |
1944 |
1945 |
1946 |
1947 |
1948 |
1949 |
1950 |
1951 |
1952 |
1953 |
1954 |
1955 |
1956 |
1957 |
1958 |
1959 |
1960 |
1961 |
1962 |
1963 |
1964 |
1965 |
1966 |
1967 |
1968 |
1969 |
1970 |
1971 |
1972 |
1973 |
1974 |
1975 |
1976 |
1977 |
1978 |
1979 |
1980 |
1981 |
1982 |
1983 |
1984 |
1985 |
1986 |
1987 |
1988 |
1989 |
1990 |
1991 |
1992 |
1993 |
1994 |
1995 |
1996 |
1997 |
1998 |
1999 |
2000 |
2001 |
2002 |
2003 |
2004 |
2005 |
2006 |
2007 |
2008 |
2009 |
2010 |
2011 |
2012 |
2013 |
2014 |
2015
```

```

<?php
require "conn.php";

$ders = $_POST["ders_id"];

$sql = "DELETE FROM ders
      WHERE ders_id = '$ders'";

try {
    $result = $conn->query($sql);

    if ($result === false) {
        die("Sorgu hatası: " . $conn->error);
    } else {
        echo "Başarılı bir şekilde silindi.";
    }
} catch (Exception $e) {
    die("Hata: " . $e->getMessage());
}

$conn->close();
?>

```

Şekil 16: dersSil.php Dosyası

PHP dosyalarında son olarak verileri veritabanı üzerinden okumayı sağlayan PHP kod dosyaları içerisinde; ilk olarak, conn.php dosyasıyla bir veritabanı bağlantısı oluşturulur. Sonrasında \$sql değişkenine atanmış olan "SELECT" sorgusuyla belirtilen tablodaki, istenilen verilere erişilir. Sonrasında, sorgu çalıştırılır ve eğer dönen satır sayısı 0'dan büyükse, while döngüsü içinde her bir satırın verilerini mesaj olarak geri döner. Eğer sorgu sonucunda hiçbir satır dönmezse, yani veritabanında kayıt yoksa, "0 result" mesajı geri döner.

```

C:\Users\kaan4> OneDrive > Belgeler > GitHub > YazLab1-2 > php > veritabanlilar > dersler.php
1 <?php
2 require "conn.php";
3
4
5
6 $sql="SELECT ders_id, ders_adi, ders_sinif FROM ders";
7 $result = $conn->query($sql);
8 if($result->num_rows > 0)
9 {
10     while($row = $result->fetch_assoc())
11     {
12         echo "<table border='1'><tr><td>". $row["ders_id"]. "</td><td>". $row["ders_adi"]. "</td><td>". $row["ders_sinif"]. "</td></tr></table>";
13     }
14 }
15 else
16 {
17     echo "0 result";
18 }

```

Şekil 17: dersler.php Dosyası

6.2.B C# Dosyaları

Projede, veritabanıyla bağlantı kuran ve içinde metodlar bulunan iki C# kod dosyası yer almaktadır. Bu dosyalardan biri veri göndermek için kullanılırken diğeri ise veri okumak için yazılmıştır. Veritabanı bağlantısını sağlayan bu dosyalar, projenin işlevselliğini temsil eder ve veri işlemlerini gerçekleştirmek için kullanılır. Birinci dosya, verileri göndermek için gerekli olan fonksiyonları içerirken, diğer dosya verilerin okunmasını sağlayan metodları içermektedir.

Veritabanından verileri okumamızı sağlayan kod dosyasındaki metodlar UnityWebRequest.Get metoduyla belirtilen URL'e bir istek oluşturulur ve yield return www.SendWebRequest() ile istek tamamlanması

beklenir. Ardından, www.isNetworkError ve www.isHttpError ile hata olup olmadığı kontrol edilir. Eğer bir hata oluşmuşsa konsola bilgi mesajı yazdırır. Başarılı bir istek durumunda, istekten gelen veri www.downloadHandler.text ile alınır ve
 işaretiyle göre bu veri parçalanarak belirtilen diziye atanır.

```

1 bagvuru
public IEnumerator getDersProgrami()
{
    UnityWebRequest www = UnityWebRequest.Get(server_url + "/dersprogrami.php");
    yield return www.SendWebRequest();
    if (www.isHttpError || www.isNetworkError)
    {
        Debug.Log("Baglanti Hatası");
    }
    else
    {
        string s = www.downloadHandler.text;
        //ilk ders_programi_id
        //ikinci ders_id
        //üçüncü derslik_id
        //dördüncü gun_id
        //besinci saat_id
        //altıncı ders_sinif
        dersProgramlari = s.Split("<br>");

        foreach (string za in dersProgramlari)
        {
            za.Trim();
        }
    }
}

```

Şekil 18: Veri okuma kod örneği

Veritabanına veri göndermemizi yani silme, güncelleme ve ekleme işlemlerini yapmamızı sağlayan kod dosyasındaki metodlarda İlk olarak, server_url değişkeni, istek gönderileceği sunucunun URL'sini içerir. Ardından, bir fonksiyon tanımlanır. Bu fonksiyon, verilen parametrelerle birlikte belirtilen URL'ye POST isteği gönderir. Bu URL, server_url'de belirtilen ana URL'ye belirtilen uzantıyı ekler. sendPostRequest fonksiyonu, bu isteği gerçekleştirir. Bu fonksiyon, belirtilen URL'ye POST isteği yapar ve parametre olarak verilen verileri isteğe ekler. Bu fonksiyon, geri dönen cevaba göre Task<bool> türünde bir değer döndürür. İsteğin başarılı olup olmadığını belirtmek için bool bir değer döndürür.

```

static class HttpDestinyManager
{
    readonly static string server_url = "localhost:88/veritabanlilar";

    //dersProgrami
    public static async Task<bool> dersProgramiEkle(string p_ders_id, string p_derslik_id, string p_ogretmen_id, string p_gun_id, string p_saat_id)
    {
        string url = $"{server_url}/dersprogramiEkle.php";

        return await sendPostRequest(url, new Dictionary<string, string>()
        {
            { "p_ders_id", p_ders_id },
            { "p_derslik_id", p_derslik_id }, //Post ile belirtilen php dosyasına gönderilen değerler
            { "p_ogretmen_id", p_ogretmen_id },
            { "p_gun_id", p_gun_id },
            { "p_saat_id", p_saat_id }
        });
    }

    //dersEkle
    public static async Task<bool> dersEkle(string ders_adi, string ders_sinif)
    {
        string url = $"{server_url}/dersEkle.php";

        return await sendPostRequest(url, new Dictionary<string, string>()
        {
            { "ders_adi", ders_adi }, //Post ile belirtilen php dosyasına gönderilen değerler
            { "ders_sinif", ders_sinif },
        });
    }
}

```

Şekil 19: Veri gönderme kod örneği

6.3 Uygulama

Proje, Unity oyun motoru kullanılarak geliştirilmiş olduğu için uygulama C# programlama dili ile kodlanmıştır. Proje dosyaları arasında çeşitli işlevlere sahip bir dizi kod dosyası bulunmaktadır. Proje içerisinde

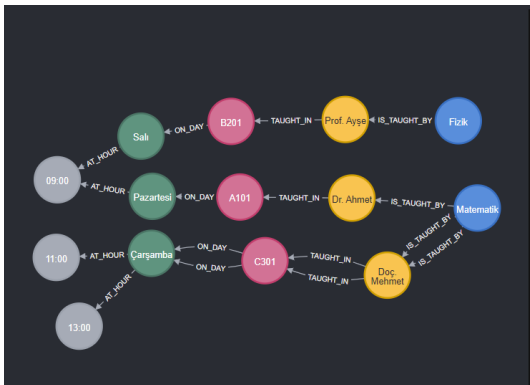
veritabanına ekleme ve çıkarma işlemlerinin yapılmasını sağlayan “EkleCikarManager”, taratılan resimlerin butonlarının kontrolünü sağlayan “Clickable”, veritabanından ders programlarının okunmasını ve yazdırılmasını sağlayan “GetTimeTables”, derslerin birbirleri ile çakışmasını ve uygunluk durumlarının kontrol edilerek veri tabanına eklenmesini sağlayan “TimetableButtons”, PlayerPrefslerin kontrolünün örneği olan “PlayerPrefsController”, veritabanında bulunan ders programlarından ders silmemiyi sağlayan “DeleteLesson”, seçilebilir verileri kullanılmasını sağlayan “DropdownDatas”, sahne geçişlerini sağlayan “HomeController” ve veri gönderme işlemlerinin örneklerinin bulunduğu “PostManager” kod dosyaları projede bulunmaktadır.

7. Karşılaşılan Problemler

Projenin ilk aşamalarında proje isterlerinde graph boyama yöntemiyle projenin gerçekleştirilmesi istendiği için daha uygun olduğu düşünülerek neo4j veritabanı yönetim sistemi kullanılmıştır. Yine aslında Mysql kullanarak oluşturulan veritabanı mantığındaki tablolar yerine nodelar yani düğümler kullanılarak ilişkiler sağlanmıştır. Örneğin pazartesi bir düğüm, salı bir düğüm, matematik dersi bir düğüm şeklinde, bunlar sorgularla birbirine bağlanmıştır, bu nodeların aralarında ilişkiler tanımlanmıştır örneğin at_hour, on_day, taught_in gibi, taught_in o dersin hangi sınıfta öğretildiğini belirler şeklindedir.

MERGE
(course)-[:IS_TAUGHT_BY]->(instructor)-[:TAUGHT_IN]->(classroom)-[:ON_DAY]->(day)-[:AT_HOUR]->(hour) şeklinde ilişkiler tanımlanmıştır.

Şekilde demo amaçlı oluşturulan neo4j veritabanı verilmiştir.



Şekil 20: Neo4j veritabanı yapısı

Bu veritabanına veri eklerken projede asıl kullanılmış olunan procedure mantığında bir kod yazılmıştır veri eklerken müsaitliği ve çakışmaları kontrol etmesi amacıyla. O kod da şekilde verilmiştir.

```
WITH (instructor_id:3, course_id:1, classroom_id:3, day_id:3, hour_id:3) AS params
MATCH (instructor:Instructor {instructor_id: params.instructor_id})
WHERE NOT (instructor)-[:TAUGHT_IN]->(:Classroom)-[:ON_DAY]->(:Day {day_id: params.day_id})
-[:AT_HOUR]->(:Hour {hour_id: params.hour_id})
WITH params, instructor
MATCH (classroom:Classroom {classroom_id: params.classroom_id})
WHERE NOT (classroom)-[:ON_DAY]->(:Day {day_id: params.day_id})-[:AT_HOUR]
->(:Hour {hour_id: params.hour_id})
WITH params, instructor, classroom
WITH params, instructor, classroom
MATCH (course:Course {course_id: params.course_id}),
      (day:Day {day_id: params.day_id}),
      (hour:Hour {hour_id: params.hour_id})
MERGE (course)-[:IS_TAUGHT_BY]->(instructor)-[:TAUGHT_IN]->(classroom)
-[:ON_DAY]->(day)-[:AT_HOUR]->(hour)
RETURN true
```

Şekil 21: Neo4j müsaitlik kontrolü kodu

Fakat Neo4j ile Unity arasında veri alışverişi konusunda birçok problemle karşılaşıldı, dolayısıyla projeye Mysql veritabanı kullanılarak devam etme kararı alındı.

8. Sonuç

Belirlenen problem ve amaç doğrultusunda Unity üzerinde artırılmış gerçeklik kullanılarak bir ders programı uygulaması geliştirilmiştir. Bu uygulama ile eğitim kurumlarında karşılaşılan ders programı oluşturma problemine bir çözüm getirilmiştir, AR kullanılarak da konuya farklı bir yönden yenilik getirilmiştir.

Uygulama kolay kullanılabilir, kullanıcı dostu bir arayüz ile tasarlanmıştır.

Uygulamadaki menüden dersliklerin ders programına erişim sağlanabilmektedir. Bu ders programlarına öğretmen ve bir zaman dilimi seçildiğinde eğer öğretmen, derslik ve sınıf üçünün de durumu müsaitse programa ekleme yapılabilmektedir. Anlık olarak eklenen ders görüntülenebilmektedir.

Menüden aynı şekilde ders ve öğretmen ekleme çıkarma sayfasına erişim bulunmaktadır. Bu sayfadan veritabanına öğretmen ve ders eklenip silinebilmektedir.

Menüden son olarak AR kamerasının kullanıldığı sayfaya erişim bulunmaktadır. Bu sayfada derslikler için önceden oluşturulmuş olan dersliğin isminin bulunduğu kartlar taratılarak o dersliğe ait ders programı görüntülenebilmektedir.

Referanslar

- <https://developer.vuforia.com/> [24.12.23]
- <https://neo4j.com/docs/> [24.12.23]
- <https://learn.microsoft.com/en-us/sql/relational-databases/graphs/sql-graph-architecture?view=sql-server-ver16> [25.12.23]