Pamukkale Univ Muh Bilim Derg, 24(2), 238-245, 2018

Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi

Pamukkale University Journal of Engineering Sciences

Mobil cihazlar kullanılarak kapalı alanlarda navigasyon sistemi

Indoor navigation system with using mobile devices

*Oğuzhan KAYIŞ[1], Yusuf ÇAKMAK[2], Semih UTKU[3]**

[1,2,3]Bilgisayar Mühendisliği Bölümü, Mühendislik Fakültesi, Dokuz Eylül Üniversitesi, İzmir, Türkiye.
oguzhankayis@gmail.com, yusuf.cakmak@windowslive.com, semih@cs.deu.edu.tr

Geliş Tarihi/Received: 11.10.2016, Kabul Tarihi/Accepted: 12.06.2017
doi: 10.5505/pajes.2017.15010
* Yazışılan yazar/Corresponding author
Araştırma Makalesi/Research Article

Öz

Gündelik hayatta, açık veya kapalı alanlarda belirli bir yer bulmaya olan ihtiyaç için, farklı çözümler üretilmektedir ve farklı teknolojiler kullanılmaktadır. GPS teknolojisi açık alanlarda birçok farklı amaç için kullanılırken, kapalı alanlardaki yer bulma problemi için henüz genel bir çözüm/teknoloji bulunmamaktadır. Kapalı alanlar için geliştirilen yaklaşımların en temel problemleri, kapalı alanların modellenmesi ve kullanıcının konumunun belirlenmesi sürecidir. Bu çalışmamızda kapalı alan navigasyon sistemine genel bir çözüm modeli önerilmektedir. Çalışma kapsamında, binaların modellenebileceği bir web uygulaması ve modellenen binaların 2B düzlemde gözlemlenebileceği bir mobil uygulama geliştirilmiştir. Kapalı alanlarda yol bulma problemini A algoritmasını kullanarak çözmeyi amaçlayan bu çalışmamızda ayrıca bina modelleme işlemi için de bir araç sunulmaktadır.*

Anahtar kelimeler: Web tabanlı modelleme, Kapalı alan navigasyon, A* algoritması

Abstract

In our daily lives, different solutions are produced and technologies are used for the need to find a specific place in indoor or outdoor areas. While GPS technology is used for many different purposes in outdoor areas, here is not yet a general solution/technology for locating the problem in indoor areas. The most basic problems of approaches developed for in indoor areas are the modeling of indoor areas and the determination of the user's location. In this study, we propose a general solution model for indoor area navigation system. A web application was developed for modeling buildings in 2D plane. Besides, a mobile navigation application was implemented for indoor areas. Mobile application are intended to minimize the loss of time during finding places in indoor areas for users. For this purpose, the shortest path algorithm (A algorithm) for mobile application has been added to the study.*

Keywords: Web based modeling, Indoor navigation, A* algorithm

1 Giriş

Dünya üzerinde nüfusu bir milyondan fazla olan dört yüzün üzerinde şehir bulunmaktadır. Gelişmiş ülkelerde nüfusun %70'inin kentsel alanlarda yaşadığı bilinmektedir [1]. Bu nüfus değişiminin ve hareketinin bir sonucu olarak özellikle kentsel alanlarda yaşayanların günlük hayatlarını kolaylaştırmak amacıyla navigasyon sistemleri zamanla daha sık kullanılır hale gelmiştir. GPS (Küresel Konum Belirleme Sistemi) teknolojisi ile birlikte, uydu tabanlı navigasyon sistemleri hayatımıza girmiş ve yaygın olarak kullanılmaya başlanmıştır. Savunma sanayi, deprem analizleri, işaretleme, arama kurtarma, araç takip sistemleri, uçak takip sistemleri gibi birçok modern uygulamada GPS kullanılmaktadır. Tüm bu kullanım alanları, özellikle son yirmi yıllık zaman dilimi içerisindeki gelişmelerle GPS'in birçok sektörde etkinleştiğini göstermektedir [2].

Uydular ve mobil yol araçları kullanılarak dış alanların haritalanması mümkündür ve detaylı yol bilgileri rahatça elde edilebilmektedir. Günlük hayatımızda GPS kullanan araçlardaki navigasyon sistemleri, bizlere her gün adres tarifi yapmaktadır. Bu çözümler gerek zaman, gerekse maliyet olarak bize katkı sağlamaktadır. Tüm bu kullanım alanlarına ve artılarına rağmen; günümüzde GPS'in açık alanlardaki (outdoor) navigasyon başarısına yaklaşan herhangi bir kapalı alan (indoor) navigasyon sistemi bulunmamaktadır. Hastane, kamu binası, alışveriş merkezleri gibi farklı kapalı alanlarda yer arama için yaşadığımız zorluklar düşünüldüğünde; kapalı alanlar için geliştirilecek başarılı bir yaklaşımın ya da çözümün hayatımızı açık alanlardaki çözümler gibi olumlu olarak etkileyeceğini düşünebiliriz [3].

Kapalı alanlar için oluşturulan herhangi bir navigasyon sisteminin, GPS teknolojisin açık alanlar için olan sistemlerdeki başarısını yakalayamaması, oluşturulan sistemlerin küresel anlamda uygulanamamasıyla doğrudan ilintilidir. Kapalı alanların haritalanması veya modellenmesi için, binaların yapılarından dolayı GPS'den farklı çalışmalara ihtiyaç duyulmaktadır. Kapalı alanlar için yapılacak çalışmaların mutlaka genel kullanılabilir bir modelleme aracını içermesi gerekmektedir. Öyle ki oluşturulacak çözümler, içlerinde küresel anlamda kabul edilebilir, bina modelleme problemine çözüm oluşturabilecek bir modelleme aracını bulundurmazlar ise, küresel olarak da uygulanamaz hale gelmeleri kaçınılmazdır. Kapalı alan navigasyon sistemleri konusunda henüz bir standart olmasa da bazı yaklaşımlar mevcuttur.

Günümüzde insanlar kapalı alanlarda yol bulma probleminin çözümüne yardımcı olacak, yazılımdan bağımsız çözümler üretmişlerdir. Hastaneler, alışveriş merkezleri, kamu kurumlarının binaları gibi büyük yapıları düşündüğümüzde birçoğunun girişinde kat planlarının bulunduğunu panolar ve yol tarifi yapan küçük haritalar görürüz. Bu yöntem ulaşılacak noktaya giden yolu kısaltmak için sıklıkla kullanılsa da uygulamanın taşınabilir olmaması ve haritanın oluşturulmasının uzun zaman alışı gibi nedenlerden dolayı kullanışlı bir çözüm değildir. Bu yöntemler kapalı alanlarda yer bulma probleminin çözümü için sıklıkla kullanılıyor olsalar da çeşitli yazılım destekli yöntemler de geliştirilmiştir.

Kapalı alan navigasyon sistemleri temelde iki farklı alanda yapılan çalışmaların birleşiminden oluşur. Bu alanlar; modelleme ve model üzerinde yön bulma işlemleridir. Bu noktada, farklı uygulamalar bu iki konuyu farklı şekillerde

incelemişler ve çözümler geliştirmişlerdir. Örneğin bina modellenmesi için geliştirilen temel fikir, binanın kat planları şeklinde küçük parçalara bölünüp, bu şekilde bir haritalama oluşturmaktır. Mevcut birçok uygulama iki boyutlu modelleme üzerine oluşturulmuştur. Bu alandaki popüler uygulamaların bir karşılaştırılmasının yapıldığı tablo sonuç kısmında sunulmuştur.

2 Önceki çalışmalar ve mevcut durum

Literatürde yapılan çalışmaların çoğunda navigasyon sistemi ile birlikte konumlandırma işleminin de bulunduğunu görmekteyiz. Konum bilgisi için her ne kadar çeşitli donanımlar kullanılsa da büyük binalarda kullanılacak donanım ürünlerinin maliyetinden dolayı bu yöntemler henüz bir standart haline gelememişlerdir.

1999 yılında bir grup bilgisayar bilimci tarafından yürütülen bir çalışmada kapalı alanlar modellenirken; dış dünyadaki gibi uydu tabanlı fotoğraf bilgisi içeren ve gerçekçi modellemeler yerine, önemli yapıtaşlarının (asansörler, tuvaletler, merdivenler vb.) ikonlarla belirlenmesinin daha iyi çözümler üreteceği belirtilmiştir [4].

Bina modelleme işlemi kapalı alan navigasyon konusuyla doğrudan bağlantılı olsa da robotik konusunda da robotların izleyeceği yolun modellenmesi kapsamında bağlantılıdır. Robotik alanında yayımlanmış bir makalede haritalama işleminde, iki boyutlu dikdörtgen bölümlere ayrılmış bir yaklaşımla topolojik modelleme yaklaşımı karşılaştırılmıştır [5]. Çalışmada, dikdörtgen bölümlere ayrılmış haritalama işleminin kolay uygulanabileceği, bakım yapılabileceği, en kısa yol gibi fonksiyonların da bu yapıya kolayca uygulanabileceği sonucuna ulaşılmıştır. Dolayısıyla bu durum da oval yapılardaki binaların modellenmesi gerçeklikten uzak sonuçlar vermektedir.

Birçok çalışmada modelleme işleminin gerçek görüntüye yakınlık ve kolay uygulanabilirlik parametreleri doğrultusunda; kolay uygulanabilirlik tarafına yakın olması gerektiği savunulmaktadır. Bu konuda Jung ve diğ.'nin Wi-Fi fingerprint kullanılarak harita oluşturulması konusunda yapmış olduğu bir çalışmada; model oluşturma sürecinde karmaşık bütünsel modeller yerine dikdörtgen şekillerden oluşan bir benzerinin oluşturulmasının yeterli olduğu sonucuna ulaşılmıştır [3].

Modelleme işleminin bir diğer karar verilmesi gereken kısmı ise modellerin kullanıcıya nasıl sunulacağı bölümüdür. Bu noktada günümüzde mobil teknolojisi ön plana çıkmaktadır. Ancak, bina modelleri oldukça büyük görseller olabilmekteyken taşınabilir mobil cihazların ekran boyutları aynı oranda yeteri kadar büyük değildirler. Bu alanda yapılan bir çalışmada; Chittaro küçük boyutlardaki telefon ekranlarında görselleştirme işleminin nasıl daha etkili ve verimli olacağı üzerine önerilerde bulunmuştur [6].

Navigasyon sisteminin, model üzerinden yön bulma bölümü ise yer tarifini kapsayan ve gidilecek yerin görsel bir sunumunun bulunduğu alandır. Bu aşamada kullanıcılara gidecekleri yere olan en kısa mesafeyi gösterecek "en kısa yol bulucu" algoritma kullanılmalıdır. Kapalı alanlarda en kısa yol bulma problemi literatürde çokça işlenen ve çözüm aranan bir problemdir.

Mohri [7] bina modelini bir grafik yapısına dönüştürerek bir çalışma gerçekleştirmiştir. Çalışmasında, gidilebilecek bölümleri grafiğin düğümleri gibi tasarlamış ve bu grafik üzerinde en kısa yol bulma algoritmasının uygulanmasını göstermiştir. Bu çalışma da ayrıca bu işlemin verimlilik analizleri de yer almaktadır. 2011 yılında Pendyala [8] tarafından yürütülen diğer bir çalışmada da yine navigasyon sistemi için grafik tabanlı bir çözümü önerilmiştir. Çalışma da dikdörtgen bölümler şeklindeki bir yapının nasıl grafik şekline dönüştürüleceği hakkında bir yol haritası sunulmuştur. Aynı zamanda ayrıntılı arama algoritmasıyla (exhaustive search) en kısa yol bulma işlemini garanti hale getirmiştir. Önerilen çözümde, bina modelleri içerisinde gidilebilecek alanlar ve engel olarak tanımlanan öğeler oluşturulmuştur. Çalışmada kullanılan yol bulma algoritması bu engelleri de dikkate alacak şekilde çözüm üretmektedir. 2013 yılında yayınlanan başka bir çalışmada da yine engellerin olduğu bir grafik harita üzerinde en kısa yol bulma işlemi incelenmiştir [9]. Dikdörtgen bölümlerden oluşan yapı, grafiğe çevrilmiş ve en kısa yol bulma işlemi için A* algoritmasından faydalanılmıştır. Andrade [10] grafik üzerinde kısa yol problemine yeni bir formülasyon ekleyerek bir çalışma yayınlamıştır. Çalışmada üç farklı yöntem incelenmiş ve karşılaştırılmıştır. İlk yöntemde grafik, kapsayan ağaç veri yapısına (spannig tree) dönüştürülerek en kısa yol bulma algoritmaları uygulanmıştır. İkinci yöntemde, numerik işlemlere dayalı ve daha az değişkenin kullanıldığı bir model oluşturulmuştur. Son yöntemde ise akış tabanlı kompakt modeli kullanarak Steiner gezgin satıcı problemi (Travelling Salesman Problem) ile ilişkilendirmiştir. Andrade çalışma sonucunda düğüm sayısı yüzlerce olduğunda da gerçekten verimli bir çözüm ürettiği için numerik tabanlı modeli önermiştir.

Literatürdeki diğer çalışmalarda da navigasyon sistemi ile beraber donanım destekli konumlandırma üzerinde çalışılmıştır. Çalışmalarda birçok teknolojinin verimli bir şekilde kapalı alanlarda kullanılamadığına değinilmiştir [11]. Mautz, çalışmasında tam olarak konum bulma işleminin kapalı alanlarda açık alanlarda çok daha zor olduğunun altını çizmektedir. Çalışmada bu durumun temel sebepleri; uydu sinyallerinin kapalı alanlarda kullanılamaması ve iç mekânların mimari olarak küçük birimlerden oluşması olarak gösterilmiştir. Zheng ve diğ. 2016 yılında üç boyutlu (3B) kapalı konumlandırma sistemi üzerinde bir çalışma gerçekleştirmişlerdir [12]. Çalışma da, 3B gösterimi ile bir kişinin ya da cihazın konumunu belirlemek amacıyla düşük maliyetli olduğu öne sürülen Mikro-Elektro-Mekanik Sistemi (MEMS) sunulmuştur. Farklı sensörler (jiroskop, ivmeölçer, barometre) kullanılarak konum bilgisine ulaşılan bu çalışmada, sensör bilgileri ilişkili ve birlikte kullanılmıştır.

Wang ve diğ.'de [13] RFID teknolojisi kullanılarak kapalı konumlandırma sistemi üzerinde çalışmışlardır. Konumlandırma bilgisi için kullandıkları teknoloji ucuz bir çözüm olmasına karşın, bu sistem de bir binanın harita bilgisinin oluşturulması zor bir süreçtir. Tüm bu çalışmalar kapalı alanlarda yapılan navigasyon sistemleri için farklı çözüm arayışlarının devam ettiğini göstermektedir. Yapılacak çözümün modelleme ve yön bulma süreçlerini birlikte barındıracak bir çözüm olması gerekmektedir. Ayrıca çözümün mobil bir uygulama ile kullanıcılara sunulması ve yaygınlaştırılması hedeflenmelidir. Bu çalışmamız ile kapalı alan yön bulma konusunda bir çözüm üretmeyi amaçlamaktayız.

Günümüzde kapalı alan navigasyon sistemleri, harici donanımsal çözümler kullanarak ya da mobil cihazlardaki sensörler vasıtasıyla konumlandırma sorununu çözmeye çalışmışlardır. Diğer taraftan bir başka sınıflandırma da

içerisinde bina modellemek için bir araç bulunduran ve bulundurmayanlar olarak yapılabilir. Bir çizim aracını da içerisinde bulunduran sistemler küresel ölçekte daha fazla kullanıcıya hitap etmektedir. Bu alanda piyasayı domine eden herhangi bir teknoloji ya da yazılım olmasa da bazı ürünler ön plana çıkmıştır.

Google şirketinin kapalı alan navigasyon sistemi için Google Indoor Maps adında bir yaklaşımı bulunmaktadır [14]. Bu yazılım bu çalışmaya konu olan benzer bir şekilde kat planları 2B düzlemde mobil cihaza yansıtarak kullanıcıya bina modelini sunmaktadır. Uygulama çevrimiçi modda çalışır. Bina modelleme işlemi genellikle profesyonellerce yapılmasını gerektirmektedir. Kullanıcıların kendi binalarını modelleyebilecekleri basit bir araç da çözüm içerisinde bulunmaktadır.

Bu alanda çalışan ve popüler olan başka bir şirket de IndoorAtlas'dır. Şirketin önerdiği çözümde binaların manyetik alanını incelenerek harita bilgisi alınmaktadır. Bu bilginin binanın modellenmesinde kullanılması hedeflenmektedir. Ayrıca bu alanda çalışan mobil yazılım geliştiriciler için de aynı yöntemle çalışan bir platform ile kullanıcılara sunulmaktadır [15].

Android markette de içerisinde çeşitli alışveriş merkezi, hastane gibi büyük binaların modellerinin bulunduğu birçok özel uygulama bulunmaktadır. Bu uygulamalardan bazıları da tıpkı bu çalışmada ortaya çıkardığımız yazılım gibi çevrimdışı modda çalışabilmektedir. Ancak bina modellerinin statik olarak kaydedildiği bu uygulamalar yalnızca kısıtlı bir alandaki binaların modelleri içerdiğinden küresel ölçekte yaygın olamamışlardır [16]. Yukarıda belirtilen uygulamaların dışında, bina modelleme işlemini mobil cihaz üzerinde çizimle yapan, uygulamalar da bulunmaktadır [17].

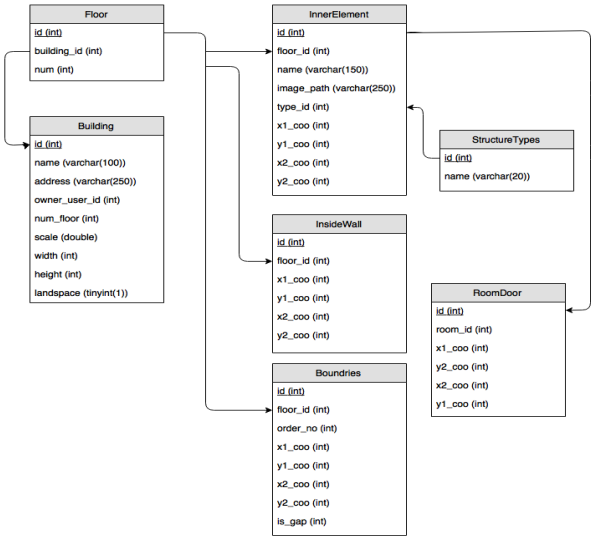
Şuan farklı yönleriyle ön plana çıkan kapalı alan navigasyon sistemleri bulunsa da henüz genel bir çözüm bulunmamaktadır. Kapalı alanlar günümüzde, insanların bilmedikleri bir konumu ararken çok zaman kaybettikleri bölgelerdir. Bu zaman kaybını en aza indirgenmesi ve bu alanda henüz bir standart olmayışı çalışmamızın temel motivasyon kaynağıdır.

3 Navigasyon sistemi

Geliştirdiğimiz sistem web tabanlı bir modelleme ortamı ile modelleri kullanan mobil tabanlı iki temel uygulamadan oluşmaktadır. Web uygulaması, kullanıcıların binaları modelleyebilecekleri bir ortam sunmaktadır. Web Uygulama aracılığıyla oluşturulan ve kaydedilen bina modelleri, Android tabanlı bir mobil uygulamayla kullanıcılara sunulmaktadır. Mobil uygulama aynı zamanda kapalı alanlarda yol bulma sürecini hızlandıracak bir algoritma (Algoritma-2) ile desteklenmiştir. Çalışmamızın içerisinde bina modelleme bölümü önemli bir yer tutmaktadır. Bu çalışma temelde iki farklı bölümden oluştuğu için iki farklı kullanıcı kitlesine hitap etmektedir. Alışveriş merkezleri, hastaneler ve kamu kurumlarının binalardan sorumlu kimseler binaların modellendiği web uygulamamızın muhtemel kullanıcıları olurken, her gün bu binaları ziyaret eden binlerce insan da mobil uygulamamızın hitap ettiği kullanıcı kitlesini oluşturmaktadır.

Bina modellerinin bir kere telefona indirilmesinin ardından mobil uygulamanın çevrimdışı olarak çalışabilmesi, mobil uygulamanın en önemli özelliklerinden biridir.

Oldukça düşük yer kaplayan model bilgileri, akıllı telefonlarda JSON (JavaScript Object Notation) formatında depolanmaktadır. JSON formatı diğer bir veri iletim formatı olan XML (Extensible Markup Language) yapısına göre daha az yanıt süresi (response time) ve ayrıştırma zamanı (parsing time) olduğu için tercih edilmiştir [18]. Bir standart oluşturmak amacıyla tasarlanan JSON tipindeki model, kullanıcının bina görseline ulaşmak istemesiyle kısa bir süre içerisinde iki boyutlu düzlemde gösterilir. Bu yapı herhangi bir binanın modellemesinde de küresel anlamda uygulanabilecek şekilde oluşturulmuş ve bina içindeki alanlar kalıtsal bir yapının parçası olacak şekilde tanımlanmıştır. Bu objenin kalıtsal yapısının detayları Şekil 1'de verilen diyagramda gözlemlenebilir.



Şekil 1: Bina bilgisinin depolandığı obje modeli.

Önerdiğimiz yapıda bir bina objesi; içerisinde farklı sayılarda katlar bulundurabilir. "Bina" objesinin, "binanın sınırları" objesi ile bağlantısı kat planı üzerinden sağlanmaktadır. Bu sayede katların farklı boyutlarda olabilmesi sağlanmıştır. Geçmişte yapılan ve incelediğimiz çalışmalardan da elde edilen veriler ışığında, Dikdörtgen temelli olarak bina modellemenin uygun olacağı sonucuna ulaşılmıştır. Bu model yapısıyla yol bulma problemi gerçekleştirilmektedir. Bu nedene kat sınırları için iki çapraz köşe noktasının koordinatları kaydedilmektedir. Her kat objesinde; iç mekânda kullanılan duvarlar ve oda benzeri yapılar bulunmaktadır. Bu yapılar oluşturulan çözümde iç elemanlar (inner elements) olarak tanımlanmıştır. İç elemanlar odalar, asansörler, merdivenler gibi nesnelerin; koordinatları, hangi kata ait oldukları, isimleri ve resimleri gibi ortak özellikleri barındırmaktadır. Bu nedenle tek bir obje ile temsil edilmişlerdir. Bu yapıların çizim için tiplerinin ayrıca belirlenmesi için başka bir tablo ile bağlanmışlardır (StructureTypes). Kısa yol bulma algoritması bu iç elemanlar arasında uygulanmıştır. Bağlantının sağlanacağı düğüm noktalılarını belirlemek amacıyla da bu elemanların bilgileri ayrıca bir nesnenin (RoomDoor) içinde belirtilmişlerdir.

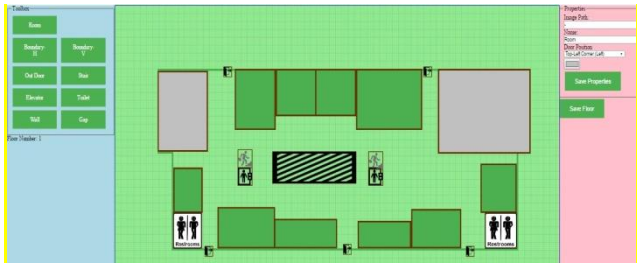
Büyük binaların modellenmesini oldukça düşük zamanlarda tamamlanmasını sağlayan bir çizim aracı içerisinde bulunduran çalışmamız, uygulama çıktısı olarak bir obje ürettiğinden, kolay modellenme özelliği ile iç mekân modelleme sistemi problemine bir yenilik sunmaktadır.

3.1 Bina modelleme uygulaması

Katların birbirinden farklı yüzey alanlarına sahip olabildiği bu sistem, popüler çizim uygulamalarının yapısı örnek alınarak tasarlanmıştır [19],[20]. Ancak bina modeli tamamen birebir ölçekte değil bazı varsayımlar ve kabullenmeler üzerine oluşturulmuştur. Yol bulma işlemi için dikdörtgen temelli bir yapının ikonlarla belirtilmesinin yeterli olacağı fikri üzerinden bir tasarım yapılmıştır. Temelde bir çizim alanı ve nesnelerin içerisinde bulunduğu bir çizim aracından oluşmaktadır. Kolay kullanılabilirlik özelliğini ön planda tuttuğumuz uygulamadaki çizim aracı öğeleri Şekil 2'de gösterilmiştir. Öğelerin çizim alanındaki görsellerinin bulunduğu örnek çizim de Şekil 3'te sunulmuştur.



Şekil 2: Çizim uygulamasının çizim aracı öğeleri.



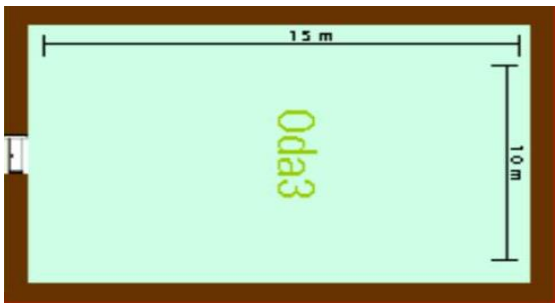
Şekil 3: Örnek bir çizim.

Bina modellemek amacıyla sisteme giren kullanıcının, çizim araçlarını kullanmadan önce topolojik yapının oluşması amacıyla, binadaki kat sayısını girdi olarak belirtmesi gerekmektedir. Kullanıcı binadaki kat sayısı bilgisini girdikten sonra zemin kattan başlayarak tüm katları çizer, çizilen kat bilgileri mevcut oturum (session) üzerinde depolanır ve kullanıcın son kat planını da çizilmesiyle birlikte bilgi veri tabanına gönderilir.

Bunun yanında uygulamanın açılış ekranında ölçeklemede kullanılmak üzere bir sabit, binanın ismi ve adres bilgilerinin girilmesi gerekmektedir. Binanın gerçek ölçüleri, web tarayıcı üzerinden girilen modelin ekran ölçüleri ve mobil uygulamada ilgili modelin gösterileceği ekranın ölçüleri birbirinden farklıdır. Gerçek ölçülerin farklı çözünürlüklere sahip mobil uygulamalarda gösterimi sırasında ölçünün korunması ve tutarlı olması gerekmektedir. Bu amaçla bina modeli oluşturulurken bina ölçüm değerleri (metre ve piksel değerleri), çizilen binaya ait sabit bir değer ile tutulmaktadır. Scale unit olarak adlandırdığımız bu sabit değer, çizim uygulamasında piksel başına düşen uzunluk oranını metre cinsinden belirlemek amacıyla kullanılmaktadır. Önerilen değer 0,1 olarak belirlenmiştir. Girdi olarak alınan bu değer mobil ekranın boyutu bilgisi de kullanılarak oluşturulan oranla boyuttan bağımsız bir tasarım sağlamaktadır. Bina içi alanların gerçek uzunluk bilgilerinin hesaplanması işleminin yapıldığı fonksiyonun sözde (pseudo) kodu aşağıda verilmiştir. Odaların uzunluk bilgisinin mobil uygulamada gösterimi Şekil 4'te örneklenmiştir.

```
1. function scale (int wb, int hb, double scaleUnit )
2.  // wb: width browser   hb: height browser
2.  SizeObj cr // cd:current room
3.  // wb * scaleUnit = 1 m-
4.  read JSON into building
5.  LOOP at building
6.        w ← building-room-width
7.        h  ← building-room-height
7.        cr.with ← w / (wb * scaleUnit)
8.        cr.height ← h / (hb * scaleUnit )
9.        add cr into sizeList.
10.  ENDLOOP.
```

Algoritma-1: Uzunluk bilgisinin ölçeklenmesi.



Şekil 4: Odaların ölçekleme işleminin mobil uygulamada görünümü.

Sisteme değerleri girdikten sonra çizim alanının etkin hale gelmesiyle tıpkı popüler çizim uygulamalarında olduğu gibi, kullanıcı sürüklenebilir nesneleri çizim alanına taşıyabilir. Çizim alanının içinde nesnelerin boyutlarını değiştirebilir. Ayrıca oda nesnesi için mobil uygulamada kullanılmak üzere; odanın ismi, rengi gibi özellik bilgilerini de ekleyebileceği bir özellik ekranı bulunmaktadır. Kullanıcının çok katlı olarak tanımlanmış binalar için tek tek kat planlarını çizmesiyle ekranda görsel olarak bulunan yapı ilişkisel veri tabanına bir Bina ID'si üzerinden kaydedilir. Bu aşamadan sonra yapılan isteklerde bu ID kullanılarak bina bilgisini içeren JSON'a ulaşılabilir.

3.2 Mobil uygulama

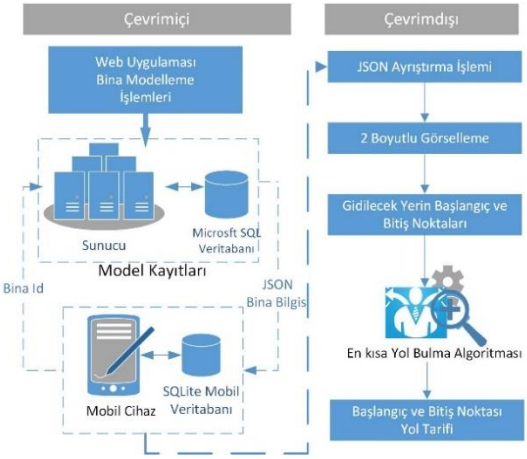
Android işletim sistemli cihazlarda kullanılmak üzere oluşturduğumuz mobil uygulama ilk yüklemede bir kayıt ekranı ile açılır. Başta bu tip bir uygulamada kullanıcı girişi anlamsız gibi görünse de; çalışmamız içerisinde kullanıcın binalar arası ve bir bina içerisinde arattığı konumları olası bir veri işleme süreçlerinde kullanılmak üzere kaydetmektedir. Kayıt işleminin ardından kullanıcı veri tabanında bulunan bina çizimlerinin olduğu sayfa ile karşılaşır.

Listview şeklinde oluşturulan bu sayfanın custom adapteri içerisinde bina isimleri ve bina ID bilgileri bulunmaktadır. Kullanıcın herhangi bir bina ismine tıklamasıyla adapterin o üyesinin ID bilgisi sunucuya gönderilen isteğe eklenir ve ilişkili JSON beklenir. Bu aşamadan sonra tüm sistem çevrimdışı modda çalışmaktadır. Gelen bilgi mobil cihazın dış (external) alanına kaydedilir. Sistem bina ID'si ile eşleşecek şekilde oluşturulur. Kullanıcıya mobil cihazda depolanan bina,

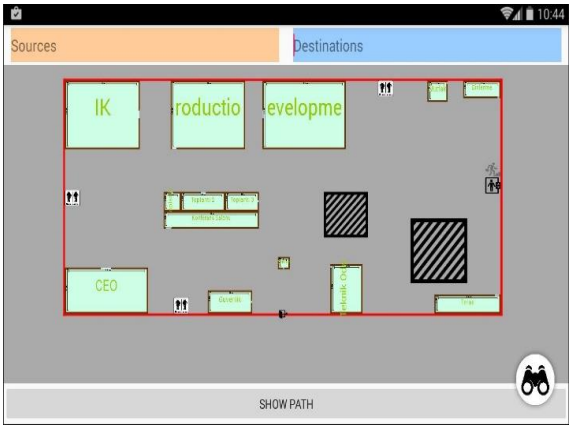
binalardan oluşan en son kullanılan yerler isimli bir liste ekranına sunulur. Kullanıcı bu listeden bir kayıt seçtiğinde seçilen kaydın ID bilgisi kullanılarak mobil cihazın veri tabanına sorgu yapılır. İlgili JSON objesi çizimlere dönüştürülmek üzere hazır hale gelir.

Mobil cihazda SQLite dosya tabanlı veri tabanı içerisinde kayıtlar saklanmaktadır. Mobil veri tabanına yapılan sorgu her bina seçiminde yalnızca tek bir kayıt ile sınırlı olmaktadır. Şekil 1'de bulunan nesne modeli aynı zamanda mobil cihazda bir bina bilgisinin bina ID bazında nasıl saklandığını da göstermektedir. Sistemde sadece bina içi nesnelerin 2B düzlemdeki pozisyon bilgileri saklanmaktadır. Mobil cihazda gerekli alan ortalama büyüklükteki bir alışveriş merkezi modeli için 3 kilobayt seviyesindedir.

JSON objesi kullanıcının bina çizimini görmek istemesi durumunda obje ayrıştırılarak küresel değişkenler içerisine kaydedilir. Ayrıştırma işleminin ardından tüm gidilebilecek noktaların koordinatları kaydedilmiştir. Bu bilgi hem sistemde girdi alırken hem de çıktı verirken hızlı bir sonuç almak için kullanılmaktadır. Sistemin genel işleyişi Şekil 5'te gösterilmiştir. Kullanıcının konum bilgisi için herhangi bir donanım kullanmadığımızdan dolayı sistemde gerekli olan hedef ve kaynak pozisyonları bilgisi kullanıcıdan girdi olarak alınmaktadır. Şekil 6'da gösterilen mobil uygulamanın ekran görüntüsünde üst bölümde kaynak ve hedef noktaları için iki adet ön tahminli yazı giriş ögesi bulunmaktadır. Ayrıştırılan JSON objesinden alınan bilgiler kullanılarak, kaynak noktasının alındığı kutucukta kullanıcının bulunduğu kattaki tüm gidilebilecek noktalar hazır halde sunulmaktadır.



Şekil 5: Sistemin genel işleyiş şeması.



Şekil 6: Mobil uygulamanın ekran görüntüsü.

Hedef noktası içinse kullanıcıya binadaki gidilebilecek tüm alanlar kat bilgisinden bağımsız olarak olası hedefler olarak tanımlanmıştır. Kullanıcının başlangıç veya hedef noktasını seçmesi ile birlikte ikonlar alanların üzerine yerleştirilir. Böylece kullanıcı gideceği yerin neresi olduğunu gözlemleyebilir. Eğer hedef noktası gösterilen katta değilse kullanıcıya gitmesi gereken kat bilgisi ayrıca ekrana yazılır. Bu durumda kullanıcının bulunduğu kat ile hedef kata ulaşmasını sağlayacak yapı (asansör ya da merdivenden hangisi yakında) seçilir ve toplam yol kullanıcı konumundan asansör/merdivene ve oradan hedef noktasına olmak üzere ikiye bölünür. Sistem çıktı olarak ekrana ikonlar yerleştirse de iki nokta arasında yol bilgisi isteyen kullanıcılar için de bir algoritma kullanmaktadır. Şekil 5'te görülen ekranda en alt bölümdeki butona tıklanması ile birlikte dikdörtgen bölümler şeklindeki sistem için bir A* algoritması uygulanmıştır. Algoritma ve çalışma prensibi başlık 3.3'te açıklanmıştır.

3.3 En kısa yol bulucu

Çalışmamızda nesnelerin pozisyon bilgileri için çizim alanı (canvas) üzerindeki piksel değerleri kullanılmıştır. Bu sebeple bina grid tabanlı bir haritaya dönüştürülmüştür. En kısa yol probleminin çözümünde grafik tabanlı farklı kısa yol bulucu algoritmalar [21] bulunsa da A* algoritması bu yapıdaki bir haritada kolay uyarlanabilir olduğu ve mobil cihazlarda verimli çalışması nedeniyle seçilmiştir. A* algoritması bir döngü ile her durumda olası tüm seçenekleri çeşitli metriklerle kontrol etmektedir. Algoritma her aşamada en kısaya götüreceği tahmin edilen yol ile çalışmaya devam eder. Bu metrikler G, F ve H maliyetleridir.

H maliyeti incelenen noktadan varış noktasına olan uzaklığı tahmin etmek için kullanılan bilgiyi hesaplamakta kullanılmaktadır. G maliyeti bir sonraki noktanın belirlenmesi için kullanılan değerdir. F maliyeti G ve H maliyetlerinin toplamı ile hesaplanır. Algoritmanın ilk adımı olarak başlangıç ve bitiş noktalarının koordinatları belirlenir. Bu durum temelde iki olasılığı ortaya çıkarır. Başlangıç ve bitiş noktası belirleme işlemi önceden tanımlı objeler (gezilebilir alanlar, odalar, asansörler vb.) üzerinden sisteme girildiğinden, program girilen alanların hangi katta olduğunu ilk anda tespit edebilmektedir. Eğer başlangıç ve bitiş noktaları aynı katta ise her ikisine de birer ikon eklenir ve en kısa yol bulucu algoritma çalıştırılır. Aksi halde başlangıç noktasından, başlangıç noktasının bulunduğu kattaki merdiven veya asansör için algoritma çalıştırılır ve bir yol çizilir. Kullanıcının o noktaya geldiği aksiyonu ile beraber, gideceği kat planı ekrana getirilir. Ardından merdiven veya asansör noktasından bitiş noktasına olan yolu bulması için algoritma ikinci kez çalıştırılır.

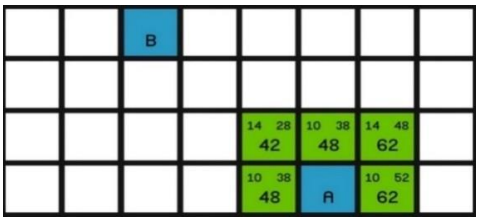
A* algoritmasının uygulanmasında her adımda bir sonraki adıma geçiş için bir parametre gerekmektedir. Yol bulucu algoritmada açıklandığı gibi yolun seçilebilmesi için bir uzaklık bilgisi hesaplanmaktadır. Komşu node'lar içinden uzaklığı en az olan seçilerek yol bulunduğundan hangi node'un daha yakın olduğu bilgisi için bir uzaklık formülasyonu gerekmektedir. Bunun için gerekli olan heuristic formülü aşağıda verilen Manhattan uzaklığı olarak seçilmiştir[22].

$$d(x,y) = |x1 - x2| + |y1 - y2| \quad (1)$$

İki nokta arasında uzaklık hesaplamasında kullandığımız bu yöntemde doğrudan koordinatları işleyip, olası köklü sayılarla CPU'yu yormak yerine tam sayı değerleri kullanılmıştır. Çapraz olarak yol uzaklığı hesaplamasına da izin veren

Manhattan uzaklık formülünü uygularken yatayda ya da dikeyde bir birimlik hareket için maliyetimizi 10 artırırken çapraz bir hareket içinse $10\sqrt{2}$ sayısının yaklaşık değeri olarak 14 sayısı kullanılmıştır.

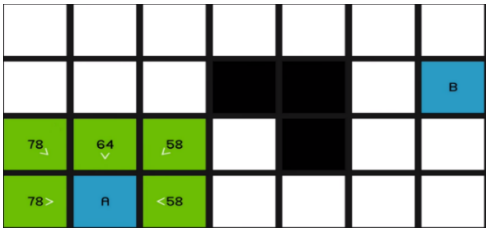
Örnek gösterim Şekil 7'de bulunmaktadır. Mavi hücreler başlangıç ve bitiş noktaları için tanımlanmıştır. A noktası başlangıç ve B noktası da bitiş noktası olarak belirlenmiştir. İlk adımda hesaplanan maliyetler gösterilmiştir. Yeşil hücreler bitiş noktasına ulaşmak için kullanılabilecek olası yerleri gösterir. A noktasının beş komşusu da bu küme içerisindedir. Yeşil hücrelerde yazan değerler hesaplanan maliyetleri göstermektedir. Üst sol köşedeki değer G maliyetini, üst sağ köşedeki değer H maliyetini ve ortada büyük puntolu değer de F maliyetini göstermektedir. Şekil 7'de A noktasının sol üst komşusu için, G maliyeti 14 olarak atanmış çünkü A noktasının çaprazda bir birim uzaklığında bulunmaktadır. H maliyeti, B noktasına çapraz olarak iki birim uzaklıkta konumlandığı için 28 olarak hesaplanmıştır. F maliyeti de bu iki maliyetin toplamı olarak formülüze edildiğinden 42 olarak belirlenmiştir. Bütün komşuluklar bu şekilde hesaplanmaktadır.



Şekil 7: A* Algoritmasının uygulanması.

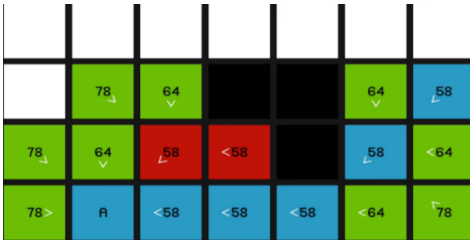
En düşük F maliyetine sahip komşu nokta bir sonraki adım olarak belirlenir ve yığın içerisine kaydedilir. Bu işlem hedef noktasına ulaşıncaya kadar devam eder. Algoritma içerisinde engeller bulunmayan bir haritada oldukça verimli olarak çalışmaktadır. Ancak, çalışmamızda harita üzerinde iki nokta arası odalar, duvarlar ve hatta boşluklar gibi yolun geçemeyeceği bölgeler bulunmaktadır. Bu problemi aşmak için çalışmamızda n*m boyutunda (n ve m canvas boyutlarını temsil eder) bir harita oluşturulmuştur. Bu harita içerisinde 1 ve 0 değerleri barındırmaktadır. Gidilemeyecek engel teşkil eden alanlar 1, yolun geçebileceği alanlar 0 olarak tanımlanmıştır.

Şekil 8'de engel içeren bir durum gösterilmektedir. A noktasından B noktasına erişim içerisinde engel durumuna göre backtrack yapılarak alternatif yol üzerine hesaplamalar ve ilerleme gerçekleştirilmiştir. Backtrack için gereken parent ilişkisi tanımlanmıştır. Eğer eşit değerlere sahip olan noktalardan biri seçilmiş ve hedef noktaya ulaşımda maliyet artmışsa backtrack yapılarak F değerleri güncellenmiştir. Engel ile karşılaşılması durumunda da F değeri güncellenmiş ve alternatif yollar üzerinde hesaplamalar gerçekleştirilmiştir. Bu işlem hedef noktasına ulaşıncaya kadar devam etmiştir. İncelenen örnek için bulunan yol Şekil 9'da gösterilmektedir.



Şekil 8: Komşu noktaların parent ilişkisi ve engel içeren harita.

Sistemin işleyişinde iki farklı liste yapısı kullanılmıştır. Bu listeler open ve closed list olarak isimlendirilmiştir. Open list işlenecek noktaları depolamaktadır. Şekil 9'da gözlemlenen yeşil hücreler bu listeye eklenmektedir. Closed list ise şuana kadar çoktan işlenmiş noktaları depolamak için kullanılmıştır. Şekil 9'daki kırmızı hücreler bu listenin elemanlarıdır.



Şekil 9: Yol bulma işleminin sonucunda noktaların gösterimi.

Algoritmanın çalışmasında ilk olarak başlangıç hücresi open liste eklenmiştir. Bu işlemle birlikte sonuç bulunana kadar bir döngü başlatılmaktadır. Döngünün sonlanma koşulu open list öğesinin boş olması olarak belirlenmiştir. Döngü içerisinde en küçük F maliyetine sahip hücre işlenen hücre olarak seçilerek, open list içerisinden silinmektedir. İşlenen hücrede closed list içerisine kaydedilir. Bu aşamada işlenen hücrenin hedef noktası olup olmadığı kontrolü yapılır. Eğer değilse komşu noktalar kullanılarak işlem devam eder. Algoritmanın pseudo kodu aşağıda verilmiştir.

```
1   OPEN // işlenecek hücrelerin kümesi
2   CLOSED // işlenmiş hücrelerin kümesi
3   add the start node to OPEN //ilk hücre eklenir
4   LOOP
5   current = node in OPEN with the lowest f_cost
6   remove current from OPEN
7   add currrent to CLOSED
8   IF current is target node //yol bulundu
9   return.
10  foreach neighbour of current node
11  IF neighbour is not traversable
12  or neighbour is in CLOSED list
13  skip to next neighbour
14  IF new_path to neighbour is shorter OR
15  neighbour is not in OPEN
16  set f_cost of neighbour
17  set parent of neighbour to current
18  IF neighbour is not in OPEN
19  add neighbour to OPEN.
20  END LOOP.
```

Algoritma-2: En kısa yol bulucu.

Algoritmanın hedef hücreyi döndürmesi ile çizim işlemi mobil uygulamada başlamaktadır. Parent child ilişkisi kullanılarak başlangıç noktasından bitiş noktasına kadar yol çizilir. Farklı bir durum olarak da eğer seçilen hedef noktası başlangıç ile aynı katta değilse öncelikle farklı bir kata gidilebilecek bir nesneye (asansör ya da merdivenden hangisi daha yakınsa) erişim yol çizilir. Hedef katta da o araçtan hedef noktasına ayrı bir yol çizilerek kullanıcıya sunulur.

4 Deneysel çalışmalar

Mobil uygulama testleri üzerine yapılan araştırmaların çoğu mobil kullanılabilirlik testine odaklanmıştır. Mobil uygulamalar için hizmet kalitesi (QoS) gereksinimleri; tepki süresi, kullanılabilirlik, verimlilik ve ölçeklenebilirlik konularına odaklanmaktadır [27]. Çoğu araştırmalarda mobil

cihazlarda kullanılan web servislerinin QoS'sini değerlendirmek için üçüncü parti yazılımların kullanılması önerilmektedir [28]. Sistemin işleyişinde hizmet veren servislerin vereceği ortalama yanıt süresi (response time) değerlendirilmek için JMeter [29] uygulaması ile bir simulasyon yapılmıştır. J Meter web servislerinin yük testini değerlendirmek için Java platformunda geliştirilmiş ve açık kaynaklı olarak sunulan bir uygulamadır. Test sırasında İntel Core i7 işlemcili ve 8 Gb RAM konfigurasyonunda bir sunucu kullanılmıştır. Simulasyon ile web hizmetlerine saniyede eş zamanlı istekler gönderilerek tepki süreleri ölçülmüştür. Eş zamanlı 100 isteğe ulaşıldığı anda 370 ms'lik yanıt süresine ulaşıldığı deneyler sonucunda elde edilmiştir. Şekil 10'da bu erişim bilgileri gösterilmektedir. Kullanıcı sayısı çok arttığı durumlarda daha fazla isteği paralel olarak ele almak mümkündür.



Şekil 10: Web servisler ortalama yanıt süresi.

Deneysel çalışmalar neticesinde, sistemi besleyen web servislerinin eş zamanlı kullanıcı erişiminde çok kısa bir süre tepki verdiği görülmektedir. Kullanılacak güçlü bir sunucu ile veya yük dengeleyici seçeneğini kullanarak daha fazla isteğin eş zamanlı işlenmesi sağlanabilir.

Çalışmamız sırasında uygulamamızın özelliklerini aşağıdaki maddeler halinde gösterebiliriz;

- Çalışmamız ile kapalı alanlarda binaların modellenebileceği web uygulaması ve modellenen binaların 2B düzlemde gözlemlenebileceği mobil uygulama geliştirilmiştir. Bütünleşik bir çözüm oluşturulmuştur,
- Literatürde ve yapılan çalısmalardaki farklı örnekler referans alınarak uygulama kolaylığı ve işlem süresi nedeniyle çalışmamızda A* algoritması tercih edilmiştir,
- Kat planları arasında oluşacak olan eksik tasarımlarda yol bulma algoritması hatalı yada sorunlu çalışabilmektedir. Bu nedenle kullanıcıların uygulama içerisinde tüm kat planlarını doğru olarak oluşturması gerekmektedir,
- Sistem 2B tasarım üzerinden gerçekleştirilmiştir. 3B nesnelerin tasarımı desteklenmemektedir. Bu nedenle 3B modeller ile çalışamamaktadır. Sistemin 3B olarak çalışabilmesi için web ortamında geliştirmiş olduğumuz çizim aracının geliştirilmesi gerekmektedir,
- Uygulamamız içerisinde, kat planları geliştirmiş olduğumuz çizim ortamı ile çizilmesi durumunda doğru çalışabilir. Sistemin esnekliği açısından farklı çizim uygulama formatlarına entegre olabilmesi için çalışmalar devam etmektedir.

5 Sonuçlar ve gelecek çalışmalar

Çalışmamızın sonucunda kullanıcıların bina modellemekte kullanabilecekleri bir web uygulaması ve modellenen binaların kat planlarının gözlemlenebileceği bir mobil uygulama ortaya çıkarılmıştır. Android cihazlar için geliştirilen bu uygulama içerisinde en kısa yol bulucu bir bölüm de bulundurmaktadır. Uygulamanın içerisinde ayıca kullanıcıların binalar arası ve bina içerisindeki girdi ve çıktı konum bilgileri veri tabanına kaydedilmektedir. Bu bilgiler ileriki aşamalarda bu alandaki ticari veya akademik veri işleme çalışmalarında kullanılabilecek durumdadır.

Çalışmamız bu alanda yer alan birçok çalışmadan farklı en temel yönü; tüm yaklaşımın herhangi bir dış donanımdan destek alınmaksızın oluşturulmuş olmasıdır. Mevcut uygulamalarda Tablo 1 de görüldüğü gibi farklı donanımlara bağımlı çözümler gösterilmiştir. Donanım bağımsız olarak geliştirmiş olduğumuz uygulamanın diğer bir özelliği de kendi bünyesinde birlikte çalışabilen çizim yazılımınında bulunmasıdır. Geliştirmiş olduğumuz çözüm ile 2-Boyutlu çizimler desteklenmektedir. Aynı zamanda çözümümüz çevrimdışı modda çalışma yapısı ile de diğer çözümlere göre farklılık göstermektedir. Mevcut durum başlığı altından örnek verilen uygulamalar ve çalışmamızın bir karşılaştırılması Tablo 1'de sunulmaktadır.

Uygulamamız bina modellemek amacıyla tasarlanmış olsa da; esnek tasarımı sayesinde market raflarının ve ürünlerinin modellenmesi, çeşitli sinema konferans salonu gibi binaların oturma alanlarının modellenmesi ya da orientering sporundaki haritalama işleminin gerçekleştirilmesinde de kullanılabilir.

Tablo 1: İlgili uygulamaların karşılaştırılması.

		Jeomanyetik Teknoloji	Wifi Kullanımı	Bluetooth Kullanımı	Çizim Aracı	Çevrimdışı Mod	2B Çizim	3B Çizim
1	Çalışmamız				√	√	√	
2	Google IM [14]	√	√	√	√		√	
3	Indoor Atlas [15]	√		√			√	
4	Fast Mall [16]					√	√	
5	Floor Plan Creator [17]				√		√	
6	Indoora [23]		√	√			√	
7	Matterport [24]					√		√
8	Insoft [25]		√	√	√	√	√	
9	Gipstech [26]	√	√	√		√	√	

Çeşitli yapı taşlarının tanımlı olduğu sistemde araç ve yaya yolu ayrımını yapan bir nesne bulunmamaktadır. En kısa yol bulucu algoritmanın böyle bir ögeye uygun olarak uygulanması ile birlikte iç mekân navigasyon sistemimiz, kapalı alanlar gibi üniversite yerleşkeleri gibi açık alanlar için de uyarlanabilir hale gelecektir.

Sistemde odaların uzaklıkları metre cinsinden kaydedilmektedir. Uygulamanın ileriki süreçlerinde eklenecek sesli yol tarif bölümü ile özellikle görme engelli ve yaşlı insanların kullanabileceği bir sürüm da planlanmaktadır.

6 Kaynaklar

[1] Population Reference Bureau. "Human Population Data". http://www.prb.org (09.10.2016).

[2] Farrell J, Barth M. "The global positioning system and inertial navigation". McGraw-Hill Professional, 1999.

[3] Jung S, Lee S, Han D. "A crowdsourcing-based global indoor positioning and navigation system". *Pervasive and Mobile Computing*, 31(1), 94-106, 2016.

[4] Puikkonen A, Sarjanoja A, Haveri M, Huhtala J, Häkkilä J. "Towards designing better maps for ındoor navigation-experiences from a case study, *MUM". 8th International Conference on Mobile and Ubiquitous Multimedia*, Cambridge, United Kingdom, 22-25 November 2009.

[5] Thrun S. "Learning metric-topological maps for indoor mobile robot navigation". *Artificial Intelligence,* 99(1), 21-71, 1997.

[6] Chittaro L. "Visualizing information on mobile devices". *Journal Computer Archive,* 39(3), 40-45, 2007.

[7] Mohri M. "Semiring frameworks and algorithms for shortest-distance problems" *Journal of Automata, Languages and Combinatorics*, 7(3), 321-350, 2002.

[8] Pendyala V. "Euclidean Shortest Path with Obstacles". Article, Indiana State University, Terre Haute, Indiana, 14.12.2011.

[9] Anbuselvi R, Phil M. "Path Finding Solutions for Grid Based Graph". *Advanced Computing: An International Journal,* 4(3), 51-60, 2013.

[10] Andrade RC. "New formulations for the elementary shortest-path problem visiting a given set of nodes". *European Journal of Operational Research,* 254(3), 755-768, 2016.

[11] Mautz R. "Overview of current indoor positioning system". *Geodesy and Cartography,* 35(1),18-22, 2009.

[12] Zheng L, Zhou W, Tang W, Zheng X, Peng A, Zheng H. "A 3D indoor positioning system based on low-cost MEMS sensors". *Simulation Modelling Practice and Theory,* 65, 45-56, 2016.

[13] Wang C, Wu F, Shi Z, Zhang D. "Indoor positioning technique by combining RFID and particle swarm optimization-based back propagation neural network" *Optik-International Journal for Light and Electron Optics*, 127(17), 6839-6849, 2016.

[14] Google. "Google Indoor Positing System". https://www.google.com/maps/about/partners/indoormaps/ (09.10.2016).

[15] Indoor Atlas. "Indoor Atlas". https://www.indooratlas.com/ (09.10.2016).

[16] Google Play Store, "FastMall, Andoroid Application". https://play.google.com/store/apps/details?id=com.mindsmack.fastmall (09/10/2016).

[17] Google Play Store. "Floor Plan Creator, Android Application". https://play.google.com/store/apps/details?id=pl.planmieszkania.android&hl=tr (09.10.2016).

[18] Rodrigues C, Afonso J, Tome P. "Mobile application webservice performance analysis: Restful services with json and xml". *ENTERprise Information Systems*, 220, 162-169, 2011.

[19] G2 Crowd Grid. "Best Building Design and Building Information Modeling (BIM) Software". https://www.g2crowd.com/categories/building-design-and-building-information-modeling-bim (03.03.2017).

[20] Jung SH, Lee G, Han D. "Methods and tools to construct a global indoor positioning system". *IEEE Transactions on Systems*, *Man, and Cybernetics: Systems*, 99, 1-14, 2017.

[21] Rohila K, Gouthami P, Priya M. "Dijkstra's shortest path algorithm for road network". *International Journal of Innovative Research in Computerand Communication Engineering,* 2(10), 6139-6144, 2014.

[22] Paul E. Black, "Manhattan distance", in Dictionary of Algorithms and Data Structures. https://www.nist.gov/dads/HTML/manhattanDistance.html (26.02.2017)

[23] Jensen CS, Lu H, Yang B. "Indoor-A new data management frontier". *IEEE Computer Society Data Engineering Bulletin.*, 33(2), 12-17, 2010.

[24] Li H. "User Interface of Indoor Mapping" based on Project TANGO". *International Conference on Indoor Positioning and Indoor Navigation*, At Alcalá de Heranes, Spain, 4-7 October 2016.

[25] Indoor Navigation. "Indoor Positioning and Indoor Tracking by infsoft". https://www.infsoft.com/, (25.02.2017).

[26] Petrich A. "GiPSTech and the Competition Over İndoor Positioning Technology". http://www.gipstech.com, (28.02.2017).

[27] Anand S, Naik M, Harrold MJ, Yang H. "Automated concolic testing of smartphone apps". *In Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering*, 59(1-11), ACM, New York, USA, 11-16 November 2012.

[28] Gao J, Bai X, Tsai WT, Uehara T. "Mobile application testing: a tutorial". *Computer*, 47(2), 46-55, 2014.

[29] Jing Y, Lan Z, Hongyuan W, Yuqiang S, Guizhen C. "JMeter-based aging simulation of computing system". *In Computer, Mechatronics, Control and Electronic Engineering (CMCE), 2010 International Conference*, Changchun, China, 24-26 August 2010.