

Yüksek doğrulukta kaba inşaat kalemlerinin metrajını hesaplayan YBM tabanlı prototip yazılımın geliştirilmesi

Development of BIM-based prototype software for the accurate quantity take-off calculation of rough construction items

Faruk ERGEN¹ , Önder Halis BETTEMİR^{*2} 

¹Inönü Üniversitesi, Mühendislik Fakültesi, İnşaat Mühendisliği Bölümü, 44280, Malatya

• Geliş tarihi / Received: 18.05.2022

• Kabul tarihi / Accepted: 15.10.2022

Öz

Yapı Bilgi Modellemesi (YBM) kullanımı inşaat sektöründe önemli ölçüde yaygınlaşmıştır. Bununla birlikte küçük ve orta ölçekli yükleniciler YBM yazılımlarının getireceği maliyet ile YBM kullanımına adapte olabilmek için gerekli olan personel eğitimi ve iş alışkanlığı değişimi gereksiniminden dolayı YBM kullanımına uzak kalmaktadır. Bu çalışmada YBM yazılımlarının temel fonksiyonlarından olan 3 Boyutlu görselleştirme ve kaba inşaatın metraj hesaplama işlemlerini gerçekleştirebilen bir yazılım geliştirilmiştir. Python programlama dili ile uyumlu Tkinter Kütüphanesi, Python tabanlı Ursina 3B oyun motoru ve SQLite3 veri tabanı uygulaması kullanılarak insan müdahalesi olmadan beton, betonarme kalıbı ile kalıp iskelesi metrajlarını hesaplayıp 3B görselleştirebilen yazılım geliştirilmiştir. Geliştirilen yazılım çok fazla işleve sahip olmadığı için kullanımı kolay kullanıcı ara yüzlerine sahiptir. Bu sayede özel bir personel eğitimi gerektirmeden 2 Boyutlu çizimlere sadece kat yüksekliği gibi derinlik verilerinin girilmesi ile yapının 3 Boyutlu görselleştirilmesi yapılabilmektedir. Buna ek olarak beton, betonarme kalıbı ve kalıp iskelesi metrajları da otomatik biçimde çıkarılmaktadır. Elle yapılan metraj hesabı ile karşılaştırıldığında kalıpta %0.04 oranında sapma olduğu tespit edilmiştir. Yapılan literatür taramasında Revit ile elde edilen metraj verilerinin daha fazla sapabildiği görülmüştür. Geliştirilen YBM yazılımını kullanacak küçük ölçekli yükleniciler daha hassas ve hızlı metraj hazırlayarak daha doğru biçimde maliyet analizi yapma imkânına sahip olacaklardır.

Anahtar kelimeler: 3B görselleştirme, Metraj, Python, Tkinter, YBM.

Abstract

Utilization of BIM among the construction sector has become widespread significantly. However, small and medium scale contractors are reluctant to implement BIM as a consequence of the additional cost and the staff adaptation training requirements as well as major changes at work habits. In this study software, capable of executing the 3-Dimensional visualization and preparation of quantity take-off which are basic functions of BIM software, is developed. Quantity take-off computations of concrete, formwork for the reinforced concrete, and scaffolding for the formwork as well as 3D visualization tasks are executed without human intervention by utilizing Tkinter library which is compatible with Python, Ursina game engine, and SQLite3 database applications. The developed software is not versatile since it has very simple graphical user interfaces. For this reason, 3-Dimensional visualization can be achieved by the definition of depth data such as height of the floor without a special training of the staff. Moreover, quantity take off of concrete, formwork of the reinforced concrete and scaffolding for the formwork can be obtained automatically. The obtained formwork quantity take-off amount is compared with the manual quantity take-off and the comparison revealed that the automated quantity take-off deviates only 0.04%. The literature review illustrate that Revit may not achieve the aforementioned accuracy. The small-scale contractors would have the opportunity of execution of more precise cost analysis by preparing more accurate and fast quantity take-off.

Keywords: 3D Visualization, Quantity take-off, Python, Tkinter, BIM.

* Önder Halis BETTEMİR; onder.bettemir@inonu.edu.tr

1. Giriş

1. Introduction

İnşa edilen binaların özgün olması sonucunda her bir inşaat projesinde teknik çizimlerin hazırlanması, metrajların hesaplanması, statik hesaplarının yapılması, iş programının oluşturulması ve projenin inşa edilmesi ile ilgili uğraşların baştan yapılması gereklidir. Belirtilen faaliyetler farklı disiplinlerde uzmanlaşmış profesyonellerin birlikte çalışmasını gerektirmektedir. Ayrıca proje süreci boyunca taraflar arasındaki veri akışı da çok iyi yönetilmelidir. Bununla birlikte inşaat sektörü doğası gereği çok rekabetçidir ve bu nedenle inşaat firmaları inşaat işlerini hızlı ve ekonomik biçimde yürütmek zorundadır. Bu hedefler proje paydaşları üzerinde ciddi baskı yaratmakta ve işlerin yürütülmesi insan hatalarına açık hale gelmektedir.

Yapı Bilgi Modellemesi (YBM) inşaat firmaları açısından proje paydaşları arasında iş birliğini arttırmak ve veri akışını yönetmek için uygun bir araçtır. YBM, inşaat sektöründe birçok hatayı önleme potansiyeline sahip bir araç olarak görülmektedir (Zuppa vd., 2009). YBM yazılımları, bir binanın tasarımı, analiz edilmesi ve test edilmesi süreçlerinde makine tarafından okunabilen parametrik nesnelere kullanarak binaların sanal modellerini derleme yeteneğine sahiptir (Sacks vd., 2010). Bu sayede geçmişte ancak çok sayıda yazılım kullanılarak yapılabilen işler sadece bir YBM yazılımı ile yürütülebilmektedir. Teknik çizimlerin hazırlanması, inşaat kalemlerinin metrajlarının hesaplanması ve inşaat işlerinin iş programının yapılması geçmişte farklı yazılımların kullanımını gerektirmekteydi. İnşaat işlerinin yönetiminin tek bir yazılımla gerçekleştirilebilir hale getirilebilmesi yüklenici, mühendis ve mimarlar ile diğer proje paydaşları için birçok fayda sağlayacaktır. Bu faydalar yazılımlar arasında hatalı veri aktarımının yapılabilme riskinin önlenmesi, yazılımlar arasında oluşabilecek versiyon uyumsuzluklarının ortadan kaldırılması, hazırlanan belgelerin tekrar hazırlanma zorunluluğunun ortadan kaldırılması, iş programının daha kolay ve hızlı hazırlanması olarak özetlenebilir (Barlish & Sullivan, 2012).

YBM yazılımlarının belirtilen faydaları dolayısı ile dünya genelinde birçok yazılım firması YBM yazılımlarının üretilmesi veya YBM yazılımları üzerinde çalışan eklentilerin geliştirilmesi üzerine odaklanmıştır. YBM yazılımı geliştirilmesi önemli teorik ve uygulama bilgisine sahip olmanın yanı sıra yüksek miktarda sermaye de gerektirmektedir. Ayrıca YBM uygulamaları blok zincir üzerinde veri toplama, saklama ve işleme özelliklerini de barındırdıkları için bu alanda oluşabilecek bir tekelleşme inşaat sektöründeki büyük ölçekli yükleniciler dâhil birçok yükleniciye ait kâr oranları, ihale teklif bedeli ve benzeri bilgilerin belirli veri merkezlerinde saklanması riskini içerecektir. Bu nedenle önemli verileri barındıran YBM yazılımlarının ekonomi için çok önemli olan inşaat sektörüne ait verilerin gizliliği konusunda çok önemli bir yeri bulunmaktadır. Buna ek olarak yeni ortaya çıkan dijital ikiz yaklaşımını gerçekleştirebilen yazılımların geliştirilmesi ve bu konuda ilerleme kaydederek yol gösterici konuma geçebilmek önemlidir (Sacks vd., 2020). Belirtilen koşulların dikkate alınması sonucunda bu çalışmanın motivasyonunu oluşturan YBM yazılımı geliştirilmesi amaçlanmıştır. Bu çalışmada YBM yazılımı geliştirme sürecinde geline aşama anlatılmakta ve karşılaşılan hesaplama güçlüklerinin nasıl aşıldığı açıklanmaktadır.

Makalenin ikinci bölümünde YBM üzerine yapılan çalışmaların kısa bir özeti sunulmuş, üçüncü bölümde geliştirilen YBM yazılımının özellikleri ve modelleme detayları açıklanmış, dördüncü bölümde ise yapılan bir vaka çalışması gösterilmiştir. Elde edilen bulgular değerlendirilip tartışma bölümünde ise yazılımın uygulanabilirliği ve geliştirilmesi gereken yönleri tartışılmış ve sonuç kısmı ile makale sonlandırılmıştır.

2. Bilimsel yazın taraması

2. Literature review

YBM kullanımının inşaat sektörüne sağladığı faydalar sayesinde YBM yazılımlarının yeteneklerinin geliştirilmesi üzerine teorik ve uygulamaya yönelik çalışmalar önemli ölçüde hız kazanmıştır. Kalfa (2018) YBM'nin uygulama alanlarını akademik ve endüstriyel çalışmalar ile sertifika ve dokümantasyona yönelik çalışmalar olarak sınıflandırmıştır. Omaran vd. (2019) yaşam döngü maliyet analizini YBM yazılımına bir oyun motoru kullanarak entegre etmişlerdir. İnşaat sürecinin 4 Boyutlu simülasyonunu yapıp kullanım sürecini modelleyerek inşa edilen tesisin işletme giderlerini tahmin etmişlerdir. Uygulamayı kullananlarla bir anket çalışması düzenlemiş ve ankete katılanlar belirgin bir biçimde geliştirilen uygulamanın faydalı olduğu sonucuna varmışlardır. Olsen & Taylor (2017) YBM kullanımının faydalarını tespit etmek için anket düzenlemiş ve en çok fayda sağlayan alanın YBM ile elde edilen metrajın manuel hesaplamalara göre daha

hızlı ve daha doğru olması olduğunu belirlemişlerdir. Ayrıca yeteri kadar erken yapılması durumunda YBM risk değerlendirme çalışmasının gerçekleştirilmesinde de yardımcı olabilmektedir.

[Delgado vd. \(2020\)](#) Arttırılmış Gerçeklik (AG) ve Sanal Gerçeklik (SG) tekniklerinin Mimarlık, Mühendislik ve İnşaat (MMİ) sektörleri tarafından uygulanmasını incelemişlerdir. Dört tartışma grubu kurarak AG ve SG teknolojilerinin mevcut durumlarının faydaları, karmaşıklıkları, zorlukları ve yetersizliklerini ele almışlardır. MMİ sektörlerinin AG ve SG kullanımının sunduğu imkânlardan tam olarak faydalanamadığı sonucuna varılmıştır. [Ozcan-Deniz \(2019\)](#) 18 inşaat şirketi ile anket yaparak SG tekniğinin olası uygulama alanlarını incelemiştir. [Sarı & Pekerçli \(2020\)](#) Türkiye, ABD ve İngiltere’de YBM uygulamaları üzerindeki yasal durumları incelemişlerdir. [Temel & Başağa \(2020\)](#) Endüstri Temel Sınıfları (ETS) dosya formatı kullanılarak aktarılan verilerin yönetmeliklere uygunluğunu incelemişlerdir. Çalışma sonunda YBM yazılımlarının oluşturdukları modellerin yönetmelik ve şartname ihlallerini insan müdahalesi olmadan belirlemede yeterli olduğunu belirlemişlerdir.

YBM ve SG entegre edilmiş sistemler iş güvenliği ve işçi sağlığı eğitiminde ve iş kazalarının önlenmesinde de kullanılmaktadır. [Dhalmahapatra vd. \(2021\)](#) mobil vinç kullanımına bağlı olarak oluşabilecek iş kazalarını önlemek için eğitim amaçlı bir deneysel araç geliştirmişlerdir. [Albahbah vd. \(2021\)](#) SG üzerine detaylı bir literatür çalışması yapmış ve görselleştirmenin inşaat sektörü açısından iş sağlığı ve güvenliği, çakışmaların tespiti ve yapım yönetimi eğitimi alanlarında faydalı olma potansiyelinin bulunduğunu belirlemişlerdir.

Adobe Acrobat Reader yazılımının 9 ve ileri sürümleri 3 Boyutlu model eklenmesini desteklemektedir. [Nadeem vd. \(2015\)](#) Autodesk 3DSMax yazılımında 3 boyutlu modeller çizip “pdf” dosya uzantısı olarak kaydetmiş ve Adobe Acrobat Reader yazılımının 3 boyutlu modelleri destekleme özelliğini kullanarak ücretsiz bir yazılım üzerinde ve cep telefonları da dâhil olmak üzere 3 Boyutlu inşaat çizimlerinin yakınlaşma, uzaklaşma ve döndürme işlemlerini gerçekleştirmişlerdir. [Sacks & Barak \(2008\)](#) 3 Boyutlu modellemenin tasarım ve çizim hatalarını azalttığını, daha kısa teslim süresi sağladığını, verimi arttırdığını ve maliyetleri azalttığını belirtmişlerdir.

[Millman & Aivazis \(2011\)](#) Python programlama dilinin mühendislik problemlerinin çözümüne yönelik yazılım geliştirmede giderek daha uygun bir platform olduğunu belirtmişlerdir. Karmaşık sayılarla işlem yapılabilmesi, verilerin grafiksel gösterimi ve görselleştirme kabiliyetinin Python programlama dilinin giderek yaygınlaşmasında etkili olduğunu belirtmişlerdir. [Quraishi & Dhapekar \(2021\)](#) Python programlama dilinin zorluk düzeyini incelemiş ve inşaat sektörünün hesap tabloları kullanarak gerçekleştirdiği yapım yönetimi uygulamaları ve şantiye yönetimi uygulamalarının kolaylıkla Python dilinde inşaat mühendisleri tarafından kodlanabileceğini ifade etmişlerdir. [Sarvade & Pore \(2021\)](#) betonarme kiriş tasarımı yapabilen bir yazılımı Python programlama dilinde geliştirmişlerdir. Yazılımın geliştirilmesinde Python programlama dili ile uyumlu NumPy, SciPy, Pandas ve Matplotlib kütüphaneleri kullanılmıştır.

İnşaat işlerinin metraj hesaplamaları insan hatasına açıktır ve elle yorucu düzeyde hesaplama yapılmasını gerektirir. [Bettemir \(2018\)](#) kazı, beton ve betonarme kalıbı metrajlarının minha hesaplama süreçlerindeki iş yükünü hafifletmek için bir yöntem ve hesaplama süreci önermiştir. [Bettemir vd. \(2019\)](#) ile [Bettemir & Yücel \(2021\)](#) metraj hesaplamalarında yapı elemanlarının temas ettiği uzunluk, alan ve hacim değerlerini bina geometrisine bağımlı biçimde hesaplamışlardır. [Bettemir & Bulak \(2022\)](#) metraj hesaplamalarını aks birleşimlerini daha fazla özel durumu kapsayacak şekilde formülleştirerek tüm binalar için uygulanabilir duruma getirmişlerdir. [Khosakitchalart vd. \(2018, 2019a, 2019b, 2019c\)](#) çalışmalarında Dynamo yazılımının 1.3.3.4111 sürümünü duvar ve yer kaplama malzemelerinin metraj hesaplama başarısını kendi geliştirdikleri yöntemle karşılaştırmışlardır. YBM üzerinde gerçekleştirilen metraj hesaplamaları elle yapılan metraj hesaplamalarından çok daha kısa sürede tamamlanmaktadır. Buna karşın hesap hatası yapılmamışsa elle yapılan metrajlar tam sonuca sahiptir ve daha doğrudur. [Yoon vd. \(2020\)](#) isim vermeden 3 farklı ticari YBM yazılımı ile metraj hesaplamış ve yazılımların donatı metrajında tam doğru değerden sırası ile %-7.9, %-27.2 ve %-15.8 hatalı değerler sunduğunu belirlemişlerdir. Buna ek olarak betonda %-10 kalıpta ise %4.3 mertebesinde hatalı metraj sonuçları elde edilmiştir. [Pratoom & Tangwiboonpanich \(2016\)](#) Revit 2016 versiyonunu kullanarak donatı metrajı hesaplatmışlar ve gerçek değer Revit 2016’nın sunduğu değerden %17.76 daha fazla olduğunu belirlemişlerdir. [Sherafat vd. \(2022\)](#) Revit’te gerçekleştirilen donatı metrajının %-6 mertebesinde hatalı olduğunu vaka çalışmasında belirlemiştir. YBM yazılımları arasında YBM projelerinin okunabilmesini sağlayan ETS formatının proje paydaşları arasında iş birliğini önemli derecede artırmasına rağmen otomatik metraj hesaplamada %25’e kadar hata yapabileme potansiyeli olduğunu tespit

etmişlerdir. Hatanın büyük bir kısmının YBM yazılımları arasında ETS formatındaki dosyaların paylaşılması ve okunması sırasında bazı elemanların geometrik pozisyon bilgilerinin hatalı kaydedilmesi sonucu oluştuğu belirlenmiştir.

Bayram (2020) Autodesk Revit ve Nemetschek Allplan yazılımlarının sunduğu metraj değerlerini incelemiş ve 20 cm kalınlığındaki duvar metrajında değerlerin %4'ün üzerinde saptığını belirlemiştir. Ergen & Bettemir (2022) Türk inşaat sektörü tarafından yaygın olarak kullanılan 2 yazılımın kalıp ve beton metraj değerlerini karşılaştırmış ve %4 mertebesinde hatalı sonuçlar verebildiklerini belirlemiştir. Lee vd. (2014) yer döşemesi iş kaleminin metraj ve maliyet hesaplamalarını insan müdahalesiz biçimde yapan bir sistem geliştirmişlerdir. Odanın kullanım amacı, karonun türü, boyutları, malzeme türü gibi iş koşullarının sisteme tanıtılması ile önceden uzman görüşleri doğrultusunda eğitilmiş sistem işçilik ve malzeme maliyeti tahmini yapmaktadır. Choi vd. (2015) binaların ön tasarım safhasında metraj hesaplayan bir sistem geliştirmişlerdir. OpenBIM yazılımına ETS verileri aktararak verinin doğruluğu denetlenmiş ve metraj hesaplamaları gerçekleştirilmiştir. Liu vd. (2016) metraj hesaplamalarında ontoloji tabanlı semantik bir yaklaşım geliştirerek nesnelerin birbirleri arasındaki ilişkilerin tanımlanması ile daha yüksek doğrulukta sonuçlar elde etmişlerdir. Kurulan ilişkilerde duvarın kapı pencere gibi başka nesnelere barındırabileceği tanımlanarak minhalar daha doğru hesaplanabilmiştir. Kim vd. (2022) yapının tanımlanması sırasında oluşturulan modellerdeki çelişkileri incelemiş ve bunların metraj üzerindeki etkisini araştırmışlardır. Uygulanan inşaat tekniğine uyumsuz biçimde oluşturulan kompozit duvar ve prefabrik elemanların metraj değerlerinin %6 mertebesinde sapabileceği belirlenmiştir. Liu vd. (2022) metraj hesaplamalarında semantik bir model geliştirerek YBM yazılımlarının metraj hesaplaması üzerine mevcut durumlarındaki yetersizliklerin önemli bir kısmını gidermişlerdir. Döşeme-duvar, döşeme-kiriş, döşeme-kolon ve döşeme duvar birleşimlerinin geometrisi ve kesişim alanları tanımlanmış ve daha yüksek doğrulukta metraj hesaplamaları gerçekleştirilmiştir.

Farklı inşaat yöntemi uygulamak da metraj değerini etkileyebilmektedir. Bir kaya temel kazısının gerçekleştirilmesi için hidrolik kırıcı veya ripper bıçağı kullanımı bu duruma örnek olarak gösterilebilir (Bettemir vd., 2021). Belirtilen durum göz önüne alındığında metrajın YBM ile hazırlanması farklı inşaat yöntemlerinin uygulanmasının karşılaştırılmasında yardımcı olabilecektir.

Mevcut durumda YBM yazılımları yapısal ve mimari elemanların karmaşık biçimdeki kesişimlerini tam olarak belirleyememektedir. Bu nedenle oluşan kesişim alanı veya hacmi kabullenmelerle hesaplanmakta ve metraj sürecindeki minhaların bir kısmı kabullenmeler yardımıyla belirlenmektedir. Bunun sonucunda hesaplanan metraj gerçek değerden sapabilmektedir. Kolon-kiriş, kolon-döşeme ve kiriş-döşeme kesişimleri basitleştirme yapılan kesişim hesaplamalarına örnek olarak gösterilebilir. Kesişen elemanların tespit edilebilmesi kadar, kesişimin olduğu kısmın da tespit edilmesi minhanın doğru hesaplanabilmesi için önemlidir. Belirtilen sorunun çözümü için yazılım firmaları ve akademi yoğun biçimde çalışmaktadır. Bu çalışmada Python programlama dili ve bu dille uyumlu biçimde 3 Boyutlu görselleştirme yapabilen Ursina oyun motoru ve Tkinter Kütüphanesi kullanılarak YBM özelliklerine sahip bir yazılım geliştirilmiştir. Geliştirilen yazılım 3 Boyutlu görselleştirme yapabilmekte ve beton, kalıp ve kalıp iskelesi metrajlarını yüksek doğrulukta hesaplayabilmektedir. Bu sayede literatür ve uygulamadaki önemli bir boşluğu doldurulması hedeflenmiştir.

3. Materyal ve metod

3.1. Material and method

Python programlama dili içerisinde bulunan Tkinter Kütüphanesi, Python tabanlı Ursina 3B oyun motoru ve SQLite3 veri tabanı kullanılarak YBM tabanlı inşaat planlaması ve 3B görselleştirme işlemlerini gerçekleştiren bir yazılım geliştirilmiştir. Yazılımın geliştirme süreci çizim ara yüzünün tasarlanması, çizim komutlarının alınması, veri tabanı oluşturulması ve veri tabanının yönetilmesi, metraj ve maliyet hesaplamalarının yapılması ve 3 Boyutlu görselleştirme aşamalarından oluşmaktadır.

3.1. Çizim ara yüzünün oluşturulması

3.1.1. Preparation of graphical user interface for drawing

Tkinter, Python programlama dili için standart olarak kullanılan bir Grafiksel Kullanıcı Arayüzü (GKA) kütüphanesidir. Çapraz platform olan Tkinter Kütüphanesinin Windows işletim sistemi dışında MacOS ve Unix gibi birçok işletim sisteminde çalışabilme özelliği güçlü ve sağlam ara yüzler oluşturma imkânı sağlamaktadır. Bu sayede Tkinter Kütüphanesinin kullanımı giderek yaygınlaşmaktadır.

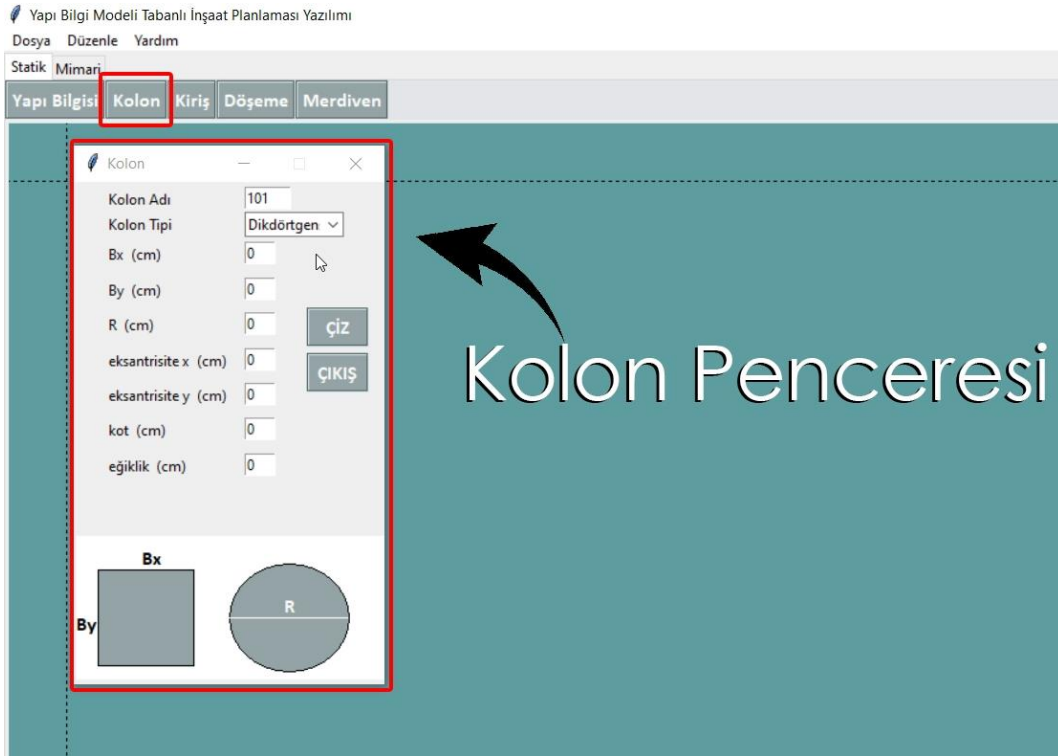
Tkinter kütüphanesindeki *Tk()* sınıfı ile geliştirilen yazılımın ana penceresi oluşturulmaktadır. Oluşturulan ana pencere üzerinde çizim ara yüzü, butonlar ve diğer araçların yerleri belirlendikten sonra *Frame()* sınıfı kullanılarak araçlar gruplandırılmaktadır. *Frame()* sınıfı, farklı araçların ana penceredeki konumlarına göre düzenlenmesi ve gruplandırılması işlemlerinin gerçekleştirilmesi için çerçeve oluşturmaktadır. Şekil 1'de butonları bir arada tutan çerçevenin oluşturulmasını sağlayan kod parçacığı sunulmuştur. Tkinter kütüphanesinde geometri yönetimini gerçekleştirmek için *pack()*, *grid()* ve *place()* metotları kullanılmaktadır.

```
buttonFrame = Frame(tab1, width=1535, height=40, bd=0, bg="#E5E7E9")
buttonFrame.grid(row=0, column=0, sticky="nsew")
```

Şekil 1. *Frame()* sınıfı kullanılarak çerçeve oluşturulması
Figure 1. *Frame formation by the Frame() class*

Çizim ara yüzünü oluşturmak için üzerinde çizim yapılabilen tuvali düzenleyen *Canvas()* sınıfı kullanılmıştır. Çizgi, dikdörtgen, çokgen, yay gibi birçok şekli çizmek için içerisinde metotlar bulundurmaktadır. *Canvas()* sınıfı ile oluşturulan çizim ara yüzü penceresinde ekran kaydırma, pan, işlemi yapılabilmesi için *Canvas()* sınıfı içerisindeki *scrollregion* komutu kullanılmıştır.

Çizilecek yapı elemanlarının özelliklerinin kullanıcı tarafından girilmesini sağlamak için açılan (pop-up) pencereler, *TopLevel()* sınıfı ile oluşturulmuştur. *Button()* sınıfı kullanılarak butonlar oluşturulmuştur. Şekil 2'de tıklandığında ilgili yapı elemanını tanımlamak için kullanıcı ara yüzü penceresini açan butonlar gösterilmiştir.



Şekil 2. Yapı elemanlarından kolona ait özelliklerin girilmesini sağlayan pencere
Figure 2. *Window providing the data entrance of the column structural element*

Kolona ait özelliklerin girilmesini sağlayan ve Şekil 2'de sunulan penceredeki metinler *Label()* sınıfı ile oluşturulmuştur. *Label()* sınıfı, resim ya da metinlerin pencere içerisinde görüntülenmesini sağlamaktadır. Kolon penceresindeki metin kutuları ise *Entry()* sınıfı ile oluşturulmuştur. *Entry()* sınıfı, kullanıcı tarafından girilen veriler için girdi kutuları sağlamaktadır. Şekil 3'te kolon penceresine ait *Label()* ve *Entry()* sınıflarının kullanıldığı bir kod parçacığı sunulmuştur.

```
label_column_name = Label(column_data_popup, text="Kolon Adı")
label_column_name.place(relx=0.10, rely=0.01)

entry_column_name = Entry(column_data_popup, textvariable=kolon_ad, width=6)
entry_column_name.place(relx=0.55, rely=0.01, relwidth=0.15)
```

Şekil 3. Label() ve Entry() sınıflarının kullanımı

Figure 3. Utilization of Label() and Entry() classes

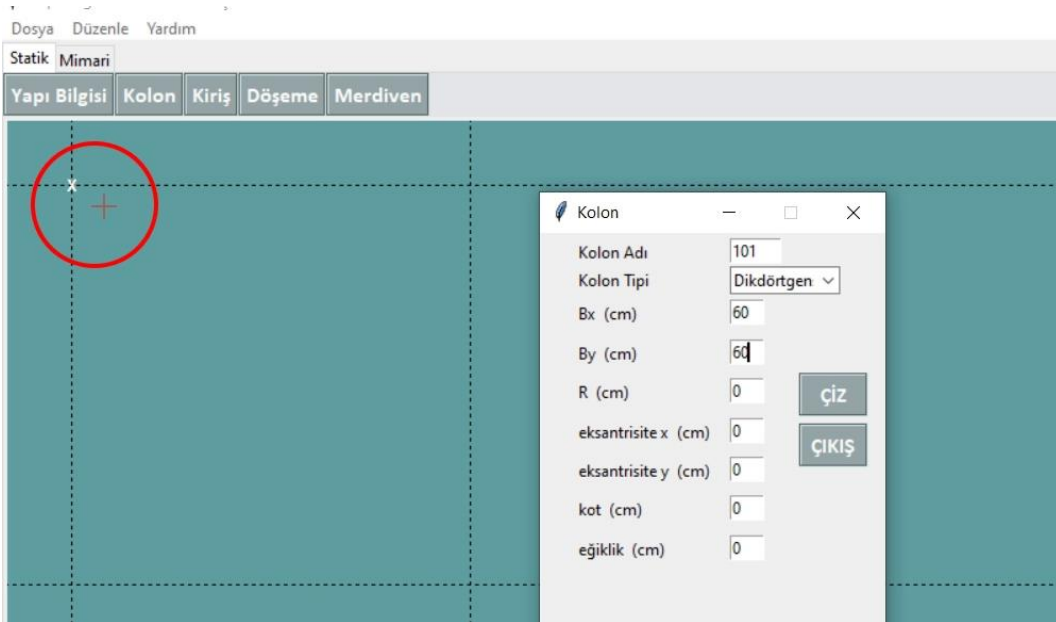
Entry() sınıfları ile oluşturulan girdi kutularındaki değerleri saklamak için değişkenlerden yararlanılır. Örneğin tamsayı saklamak için *IntVar()* sınıfı kullanılırken, dizi ya da karakter saklamak için *StringVar()* sınıfı kullanılmaktadır. Değişkenler aracılığıyla bellekte saklanan değerlerin çağırılması için *get()* metodundan yararlanılır. Örneğin Şekil 3'te kolon_ad değişkenini *kolon_ad.get()* komutu ile çağrılabilir.

3.2. Çizim komutlarının işlenmesi

3.2. Transaction of drawing commands

Çizim için oluşturulan “Çiz” butonuna basıldıktan sonra çizimi gerçekleştiren fonksiyon çağrılmaktadır. Python dilinde fonksiyon oluşturmak için fonksiyon adının önünde “def” anahtar sözcüğü kullanılır. Çizilecek elemanın şekline göre *Canvas()* sınıfına ait komutlar kullanılır. Örneğin dikdörtgen kolon elemanının 2B gösterimi için *create_rectangle()* komutu kullanılmıştır. Komutlar yalın haliyle değil elemana göre özelleştirilmiş haliyle kullanılmıştır. Çizim fonksiyonunun fare tıklamaları ile gerçekleştirilmesi için *bind()* komutu kullanılmıştır. *bind()* komutu, belirli bir işlemin belirli bir pencerede fare ve klavye düğmelerini kullanarak gerçekleştirilmesini sağlamaktadır.

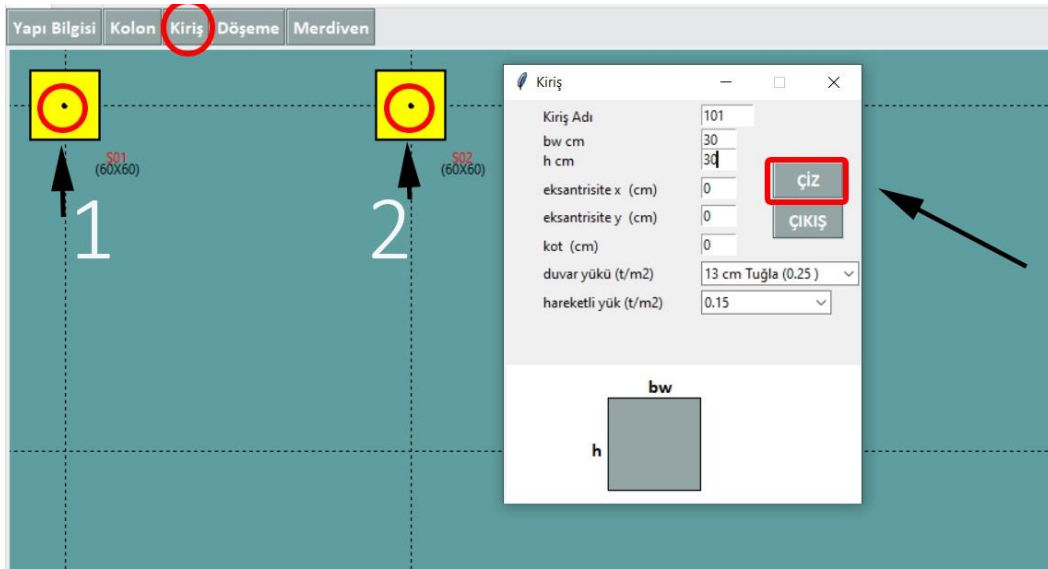
Çizilecek yapısal eleman ile ilgili gerekli öznitelik bilgileri Şekil 2’de sunulan ara yüz kullanılarak girildikten sonra “Çiz” butonuna basılarak girilen veriler kaydedilir. Ardından çizim ekranı üzerindeki akslar yardımıyla elemanın çizileceği yere farenin sol tuşu ile tıklanarak çizim gerçekleştirilir. Kolonun çizimle tanımlanması sırasında çizimin gerçekleştirilmesi için tek fare tıklaması yeterli iken giriş gibi elemanlar için 2 tıklama noktası gerekmektedir.



Şekil 4. Kolon çizimi için tanımlanan veriler

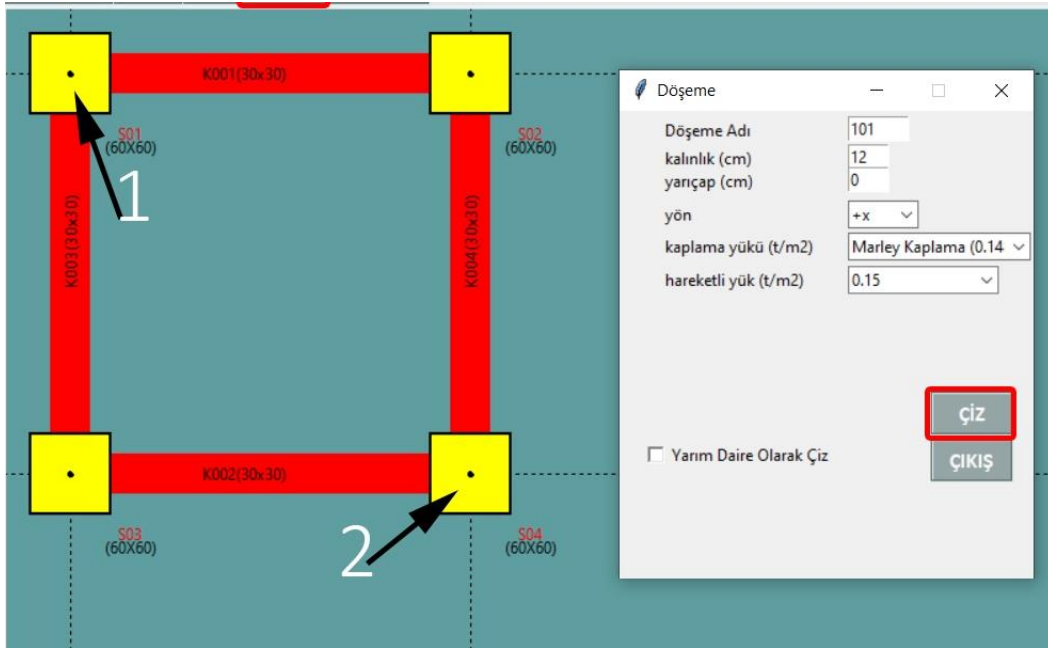
Figure 4. Defined data for the drawing of column

Kiriş elemanını çizmek için “Kiriş” butonuna basıldıktan sonra açılan pencerede kirişe ait özellikler girilir ve “Çiz” butonuna basılır. Ardından Şekil 5’te gösterildiği gibi sırasıyla 1 ve 2 numaralı düğüm noktalarına sol tıklanarak kirişin çizimi gerçekleştirilir.



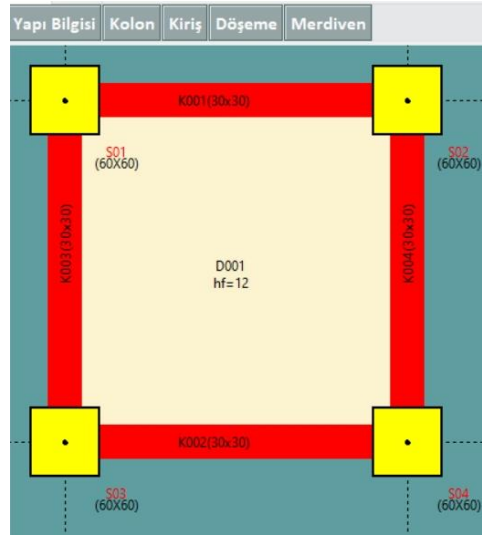
Şekil 5. Ara yüz ekranında 1 ve 2 ile ifade edilen kısımlara sol tıklanarak kirişin çizilmesi
Figure 5. Drawing of beam by left clicking the points 1 and 2 at the drawing interface

Döşeme elemanını çizerek tanımlayabilmek için döşemeyi taşıyacak olan tüm kolon ve kirişler Şekil 2, 4 ve 5'te ifade edildiği biçimde tanımlanmalıdır. Döşemenin özneliklerini tanımlamak için "Döşeme" butonuna basıldıktan sonra açılan pencerede döşemeye ait özellikler girilir ve "Çiz" butonuna basılır. Ardından Şekil 6'da gösterildiği gibi sırasıyla 1 ve 2 numaralı düğüm noktalarına farenin sol tuşu ile basılır. Böylece döşeme elemanının çizimi gerçekleştirilir.



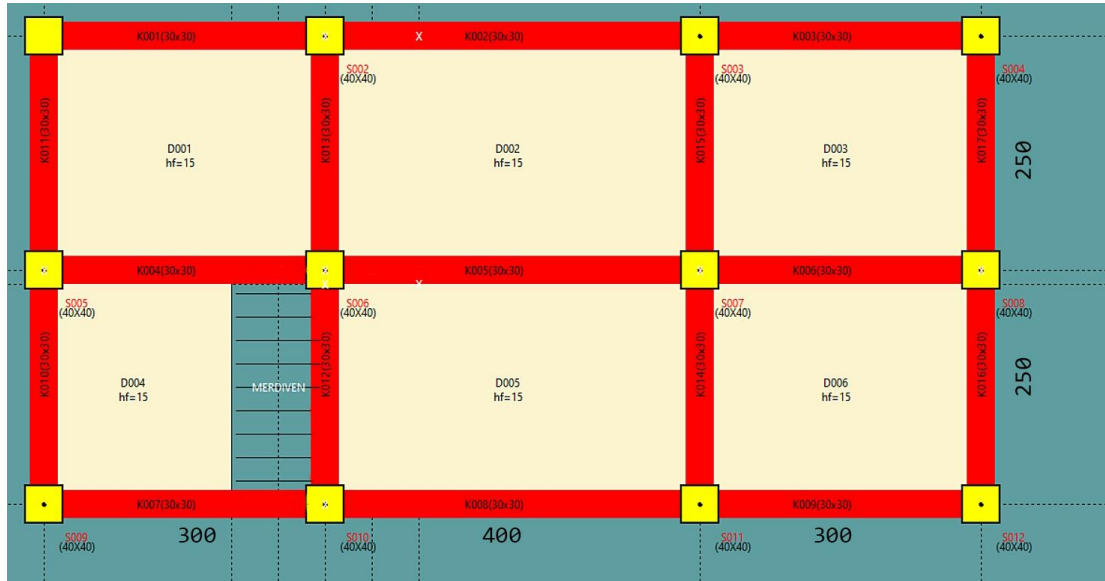
Şekil 6. Döşeme özelliklerinin girilmesini sağlayan döşeme ara yüz ekranı
Figure 6. Graphical user interface which provides the definition of slab attributes

Veri girişinin tamamlanması ile Şekil 7'de gösterilen döşeme çizilmiş olur. Çizimi gerçekleştirilen elemanlar birer nesne olarak Python'da oluşturulan bir liste içerisine eklenmektedir. Kolon, kiriş ve döşeme yapısal elemanları için farklı listeler oluşturulmuştur. Aynı sınıfa ait elemanlar aynı liste içerisine eklenmektedir. Çizilen bir elemanı silmek için eleman üzerinde sağ tıklanarak çıkan menüden "Sil" seçeneğine tıklanır. 2B çizim ekranında silinen eleman ayrıca özneliklerini saklayan liste içerisinde de silinmektedir.



Şekil 7. Döşemenin çizilerek tanımlanmasının gerçekleştirilmesi
Figure 7. Definition of slab by drawing

Çizilecek elemanların doğru noktaya konumlanması için fare ile düğüm noktasına yaklaşıldığında kullanıcıyı bilgilendirme amaçlı metinle uyarı verilmektedir. Uyarı çıktığında çizimi gerçekleştirmek için farenin sol tuşuna basıldığında çizim tam olarak istenilen noktaya göre çizilecektir. Kullanıcının bilgilendirilmesi sayesinde çizimler daha az hata ile gerçekleştirilmekte ve hesaplamalar daha doğru sonuç vermektedir. Geliştirilen YBM yazılımı ile çizilen yapının kalıp planı Şekil 8’de sunulmuştur.



Şekil 8. 2 katlı yapının birinci kat planının gösterimi
Figure 8. Illustration of floor plan of the two-story building

3.3. Veri tabanı oluşturulması ve veri tabanının yönetilmesi

3.3.1. Formation and management of database

Yapısal elemanların öznitelik bilgilerinin saklanması, sorgulanması, silinmesi ve değiştirilebilmesi için yazılıma ilişkili bir veri tabanının olması gereklidir. Geliştirilen programda veri tabanı oluşturmak için *sqlite3* veri tabanı motoru kullanılmıştır. *SQLite* açık kaynak kodlu veri tabanı motorudur. Yapılandırılmış verilerin saklandığı veri tabanları YBM yazılımları için çok büyük bir öneme sahiptir. YBM tabanlı geliştirilen programda çizilen yapı elemanların özellikleri, yapı bilgileri vb. nitelik bilgileri *sqlite3* ile oluşturulan veri tabanında kaydedilmektedir. Veri tabanına kaydedilen veriler daha sonra metraj hesaplamalarında ve yapının 3B modelinin oluşturulmasında kullanılmaktadır.

Şekil 9’da çizimi gerçekleştirilen döşemelere ait öznelik verilerinin saklandığı veri tablosu sunulmuştur. Döşemelerin başlangıç ve bitiş noktaları, döşeme kalınlığı, kaçınca kata ait olduğu gibi bilgiler veri tabanında saklanmaktadır.

	id	Xbas	Ybas	Xbit	Ybit	hf	dosemeNo	dosemeAd	dosemeTipi	dosemeCapi	mevcutKat
	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre
1	1	50	50	300	350	12	2392	001	0	0	0
2	2	50	50	300	350	12	100	101	0	0	1
3	3	300	50	600	350	12	2479	002	0	0	0
4	4	300	50	600	350	12	103	102	0	0	1
5	5	600	50	850	350	12	2553	003	0	0	0
6	6	600	50	850	350	12	106	103	0	0	1
7	7	300	350	600	600	12	2621	005	0	0	0
8	8	300	350	600	600	12	109	105	0	0	1
9	9	600	350	850	600	12	2684	006	0	0	0
10	10	600	350	850	600	12	112	106	0	0	1
11	11	50	350	200	600	12	2878	004	0	0	0
12	12	50	350	300	600	12	115	104	0	0	1

Şekil 9. sqlite3 ile oluşturulan döşeme veri tablosunda saklanan veriler

Figure 9. Data table of the data of slab stored at the database formed by sqlite3

Python *sqlite3* ile veri tabanı oluşturmak için *connect()* metodunu kullanarak *sqlite3* ile etkileşime geçilmektedir. Ardından veri tabanında işlem yapabilmek için *cursor()* metodu kullanılarak bir nesne oluşturulmaktadır. Veri tabanında yapılacak işlem *execute()* metodu ile yerine getirilmektedir. Açılan her veri tablosu işlem tamamlandıktan sonra *close()* metodu ile kapatılmalıdır aksi takdirde bellekte yer kaplamaya devam eder. Şekil 10’da kiriş elemanı için “beam.db” adında bir veri tablosu oluşturan örnek kodlama gösterilmektedir. SQLite veri tabanı motoru Windows, MacOS, Android işletim sistemlerinde sorunsuz biçimde çalışabilmesi her platformda çalışabilecek bir YBM yazılımı geliştirilebilmesi açısından önemli bir avantajdır.

```
def beam_data():
    con_3 = sqlite3.connect("beam.db")
    cur_3 = con_3.cursor()
    cur_3.execute(
        "CREATE TABLE IF NOT EXISTS beam (id INTEGER PRIMARY KEY, \
        Gx text, Gy text, Genislik text, Derinlik text, Uzunluk text, Gxbas text, Gybas text, \
        Gxbit text, Gybit text, KirisEx text, KirisEy text, kirisNo text, kirisAd text, mevcutKat text)"
    )
    con_3.commit()
    con_3.close()
```

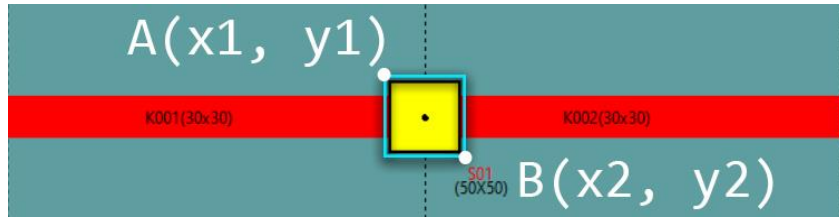
Şekil 10. sqlite3 ile veri tablosu oluşturma

Figure 10. Formation of data table by sqlite3

3.4. Çizilen elemanların komşuluklarının tespit edilmesi

3.4. Detection of the neighborhood of the drawn elements

Metrajın yüksek doğrulukta gerçekleştirilebilmesi için birbirlerine komşu olan yapısal elemanların belirlenmesi gerekmektedir. Komşu elemanların birbirlerine hangi konumda temas ettikleri ve temas yüzeyinin boyutu hesaplanmalıdır. Ayrıca döşemede kolon dışı olması gibi durumların tespit edilebilmesi için bir taşıyıcı elemanın başka bir taşıyıcı elemanın sınırlarını tanımlayan poligonun sınırlarının içine geçip geçmediğinin tespit edilmesi ve iç içe geçen elemanların ne kadar boyutta örtüştüğünün belirlenmesi gereklidir. Şekil 11’de S01 kolonuna saplanan K001 ve K002 kirişleri gösterilmektedir. Bu kirişlerin hangi kolona saplandığı ve kolonun kalıp alanı hesaplanırken ne kadar alan düşülmesi gerektiğini YBM yazılımının hesaplayabilmesi gereklidir. Ayrıca Şekil 8’de gösterildiği gibi kolonlar döşeme içine geçebilmektedir. Döşemelerin beton ve kalıp metrajlarını doğru hesaplayabilmek için kolonun ne kadarının döşeme sınırlarına geçtiğinin hesaplanması gereklidir.



Şekil 11. S01 kolonu ve komşu kirişlerin gösterimi

Figure 11. Illustration of the S01 column and the neighboring beams

Tkinter kütüphanesinin sınıflarından biri olan *Canvas()* sınıfına ait *find_overlapping()* ve *find_closest()* fonksiyonları yapı elemanlarının komşuluklarını tespit etmek için kullanılmıştır. Bu metotlar arasında en iyi sonucu veren metot yapı elemanının durumuna ve sonucun doğruluğuna göre tespit edilmiştir. Şekil 11’de gösterilen S01 kolonuna komşu olan elemanları tespit etmek için öncelikle kolona ait veri tabanından S01 kolonunun koordinat bilgileri çekilir. Elde edilen koordinat bilgileri doğrultusunda S01 kolonuna komşu olan elemanları tespit etmek için kolonun ebatlarından biraz daha büyük bir alan için önceden tanımlı bir tampon değeri kullanılarak oluşturulur. Şekil 11’de S01 kolonuna ait komşu elemanları tespit edebilmek için (x_1, y_1) ve (x_2, y_2) noktalarından geçen bir dikdörtgen alan oluşturulur. Ardından “*find_overlapping(x1,y1,x2,y2)*” fonksiyonu ile ilgili parametreler girilerek oluşturulan dikdörtgen alanı içerisinde arama yapılır. Arama sonuçlarında birden çok eleman tespit edilebilir. Bu yüzden istenilen elemanı tespit edebilmek için çizim aşamasında çizilen elemanlara “tag” parametresi verilmektedir. Örneğin kirişler için “tag” parametresi “tag=kiriş” olarak verilmiştir. “tag” parametresine göre eşleştirme yapılarak istenilen eleman türünde komşu elemanlar tespit edilmektedir.

3.5. Metraj ve maliyet hesabı

3.5. Quantity take-off and cost computations

Geliştirilen YBM yazılımı kalıp, beton ve kalıp iskelesinin metrajını hesaplayabilmektedir. Ayrıca tanımlanan rayiç değerleri kullanarak yaklaşık maliyet hesaplamaktadır.

3.5.1 Kalıp metrajı

3.5.1. Quantity take-off of formwork

Kalıp metrajı hesabı için izlenen işlem adımları aşağıda özetlenmiştir;

1. Yapı elemanlarının kolon-kiriş, kolon-döşeme ve kiriş-döşeme komşulukları tespit edilir.
2. Komşulukların temas alanları hesaplanır.
3. Hesaplanan temas (minha) alanları ilgili elemanın brüt kalıp metrajı sonucundan çıkartılarak elemanın kalıp metrajı hesaplanır.
4. İşlem 1-2-3 kalıp planındaki tüm elemanlar için uygulanır.
5. Tüm elemanların kalıp metrajı toplanarak toplam kalıp metrajı elde edilir.

3.5.2 Beton metrajı

3.5.2. Quantity take-off of concrete

Beton metrajı hesabı için ilgili elemanların ebatları veri tabanından çekilir. Kolon ve kirişlerin beton metrajı genişlik, uzunluk ve yükseklik değerlerinin çarpılması ile elde edilir. Döşemelerin beton metrajında ise kolonların komşuluklarının hesaplanması gereklidir. Belirtilen işlemin akış şeması aşağıda verilmiştir.

1. Kolon-döşeme komşulukları tespit edilir.
2. Komşulukların temas alanları hesaplanır.
3. Hesaplanan temas (minha) alanları ilgili döşemenin brüt alanından çıkarılır.
4. Elde edilen alan döşeme kalınlığı ile çarpılır ve beton miktarı elde edilir.
5. İşlem 1-2-3-4 tüm döşemelere uygulanır.

3.5.3 Kalıp iskelesi metraji

3.5.3. Quantity take-off of scaffolding for the formwork

Kalıp iskelesi betonarme kalıplarının taze betonun ağırlığını taşıyamaması nedeniyle kalıpları desteklemek için inşa edilen iskeledir. Kalıp iskelesi metraji beton dökülen bölgede desteklenmesi gereken kalıplar için kurulacak olan iskelenin hacminin hesaplanmasını gerektirir. Kalıp iskelesi metraji için hesaplanacak olan katın yüzey alanı ile kat yüksekliği çarpılır. Çarpım sonucundan ilgili katın beton metraji çıkartılarak kalıp iskelesi kurulacak kesimin hacmi hesaplanır. Belirtilen işlemin akış şeması aşağıda verilmiştir.

1. Kolon-döşeme-kiriş yapı elemanlarının yüzey alanı hesaplanır.
2. Alan değeri kat yüksekliği ile çarpılır.
3. Hesaplanan hacimden beton hacmi çıkarılır ve kalıp iskelesi miktarı elde edilir.

3.5.4 Yaklaşık maliyet hesabı

3.5.4. Computation of the approximate cost

Metraj değerlerinin, metraj kalemlerinin birim fiyatları ile çarpılması sonucu ilgili iş kaleminin yaklaşık maliyeti hesaplanır. İnşaat sektöründe yaygın olarak Türkiye Cumhuriyeti Çevre, Şehircilik ve İklim Değişikliği Bakanlığı tarafından açıklanan birim fiyatlar kullanılmaktadır. Fakat yüklenici inşaatın zorluk derecesine, inşaatın yapıldığı bölgeye göre inşaatı daha farklı birim maliyete yapabilir. Bu nedenle iş paketlerinin birim fiyatları kullanıcı tarafından Şekil 12’de gösterilen grafik ara yüz ile tanımlanmaktadır.

TARIF	BİRİM FİYAT (TL)
HAZIR BETON	260
PLYWOOD KALIP	70
KALIP İSKELESİ	15

Şekil 12. Birim fiyat tanımlama ara yüz penceresi

Figure 12. Graphical user interface for the definition of unit prices

Pencere üzerinde bulunan “Birim Fiyat” butonuna basıldığında açılan ara yüz ekranına ilgili tarifler için rayiç bedelleri kullanıcı tarafından girilir. Ardından “Metraj Hesabı” butonuna basılarak metraj ve maliyet hesabı yapılır. Sonuçlar Rapor şeklinde yeni bir pencere açılarak kullanıcıya sunulur.

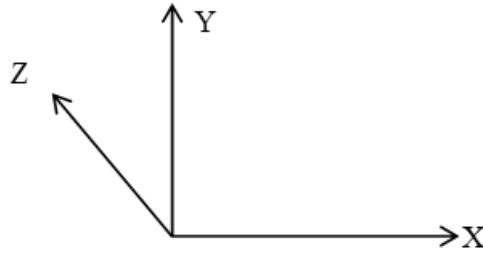
3.6. Yapının 3 boyutlu görselleştirilmesi

3.6. 3-Dimensional visualization of the structure

Geliştirilen program ile taşıyıcı sistemi çizilen bir yapının 3 Boyutlu Modelini oluşturmak için Python programlama dilinde Ursina Oyun Motoru kullanılmıştır. Python 3.6 ve üzeri sürümlerde uyumlu biçimde çalışan Ursina, açık kaynak kodlu bir oyun motorudur. 3 Boyutlu görselleştirmenin sağlıklı biçimde yapılabilmesi için koordinat sisteminin tutarlı biçimde tanımlanması gereklidir. Grafik motorlarının koordinat sistemi tanımlamaları yapı elemanı ve çizim programından aktarılan koordinatların dönüştürme işleminden sonra kullanılabilmesini sağladığı için genellikle sol el kuralına uymaktadır. Ursina oyun motorunun koordinat tanımlaması Şekil 13’te sunulmuştur.

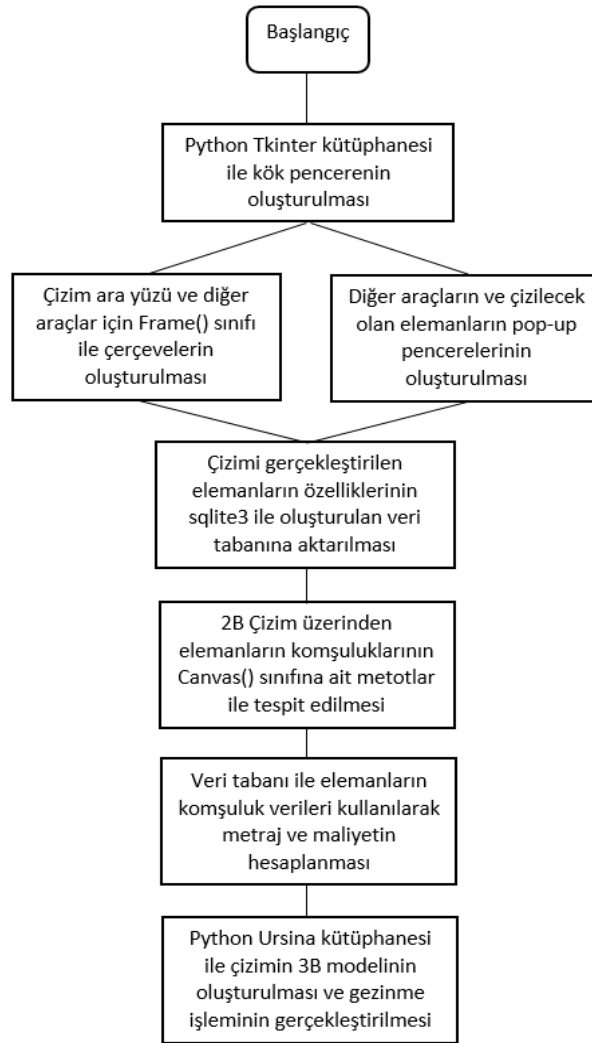
Yapının elemanlarının 3B Modelini oluşturmak için yapı elemanlarının alanları ve kalınlıkları veri tabanından sorgulama ile elde edilir. Ardından Ursina kütüphanesindeki *Entity()* sınıfı kullanılarak ilgili elemanın 3B Modeli oluşturulur. 3B Modeli oluşturulan yapının içerisinde gezinme işlemi için *FirstPersonController()* sınıfından bir FPS nesnesi oluşturulur. Oluşturulan nesnenin hareket edebilmesi için x, y ve z parametrelerine bir tam sayı değeri girilmelidir. Klavyeden “W, A, S, D, Boşluk” tuşlarını kullanarak oluşturulan nesneyi

hareket ettirerek yapı içerisinde gezinilebilir. TBM yazılımının geliştirme sürecinin akış şeması Şekil 14'te sunulmuştur.



Şekil 13. Ursina koordinat sistemi tanımının gösterimi

Figure 13. Illustration of the definition of Ursina coordinate system



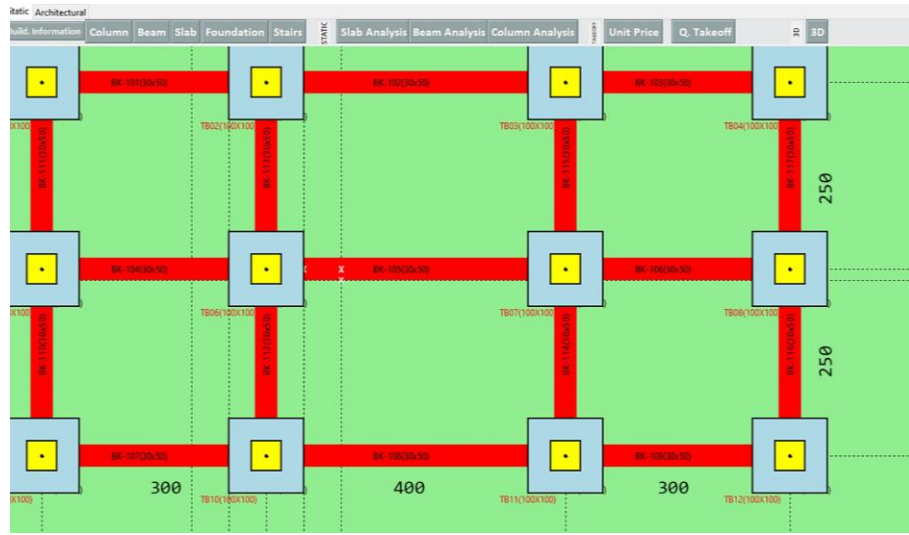
Şekil 14. YBM yazılımı geliştirme sürecinin akış şeması

Figure 14. Illustration of the flowchart of the BIM software development process

4. Vaka analizi

4. Case study

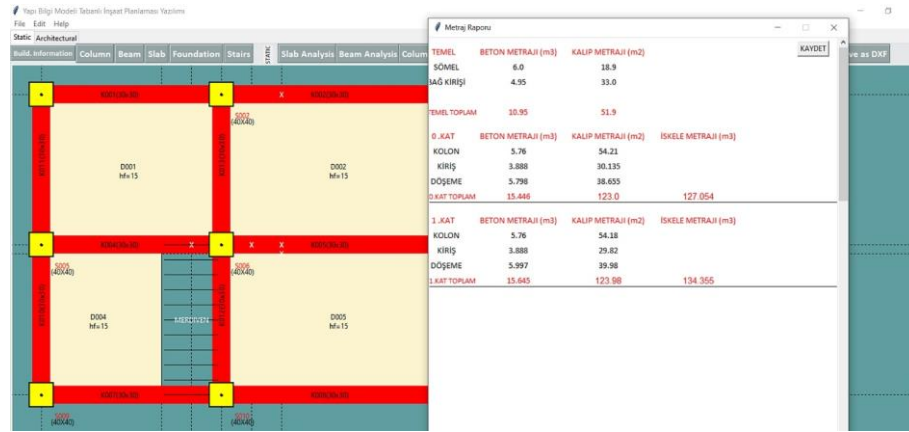
Geliştirilen YBM özelliklerine sahip yazılımın yetenekleri iki katlı bir hipotetik betonarme projesi üzerinde denenmiştir. Şekil 8'de gösterilen kat planı çizilmiş, ayrıca yapının tekil temeli ve bağ kirişlerinin Şekil 15'te sunulduğu gibi tasarım çizimleri gerçekleştirilmiştir.



Şekil 15. Yapının tekil temel planının gösterimi

Figure 15. Illustration of single footing plan of the two story building

Geliştirilen yazılım yöntem kısmında açıklanan algoritmaları koşturarak kalıp, beton ve kalıp iskelesi metrajları hesaplanmıştır. Yazılım tarafından hazırlanan metraj raporu Şekil 16'da sunulmuştur.



Şekil 16. Metraj raporu

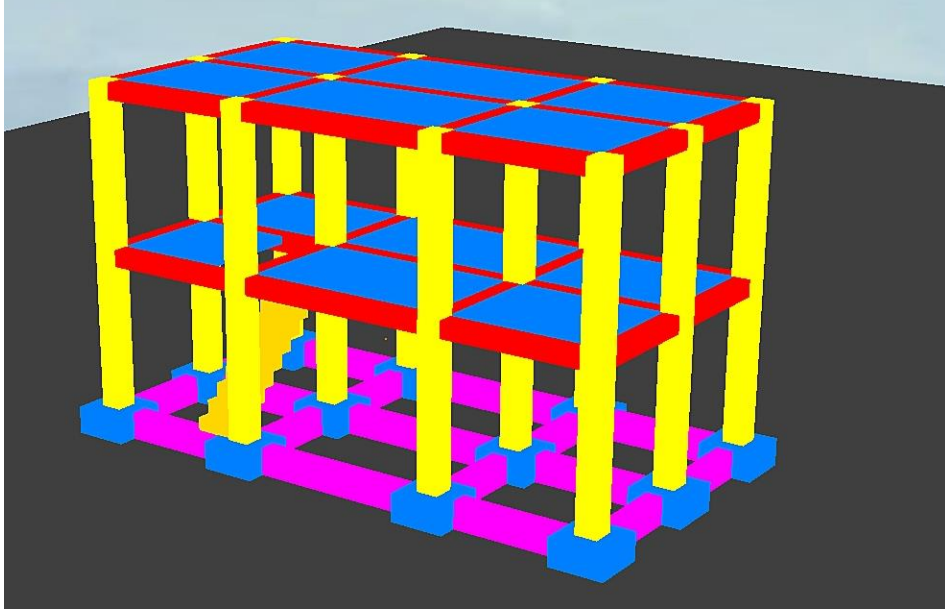
Figure 16. Quantity take-off report

Yazılımın hazırladığı metraj raporunda merdivenlerin kalıp ve beton değerleri bulunmamaktadır. Merdivenlerin metrajı dâhil edilmesi ileri çalışma olarak gerçekleştirilecektir. Merdiven dışındaki elemanların metraj değerleri el ile Türkiye Cumhuriyeti Çevre, Şehircilik ve İklim Değişikliği Bakanlığı tarafından belirtilen Birim Fiyat usulü sözleşmelerde ödemeye esas teşkil eden metraj hesaplama yöntemleri ile elde edilmiştir. Kontrol amaçlı yapının beton, kalıp ve kalıp iskelesi metrajları elle hesaplanmıştır. Yazılımın sunduğu metraj değerleri elle yapılan metraj hesaplamaları ile bire bir aynı sonucu vermektedir.

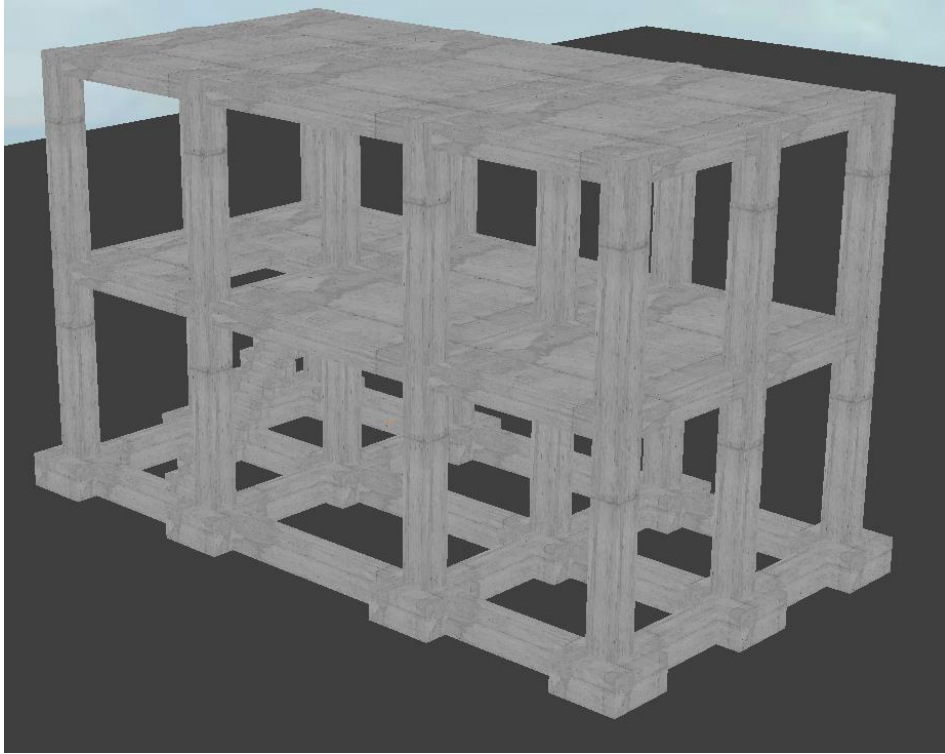
Metraj hesaplamaları sırasında kolon-kiriş, döşeme-kiriş ve kolon-döşeme komşuluklarının tespit edilmesi gerekmektedir. Vaka analizi olarak incelenen örnek projenin her katında 12 adet kolon, 17 adet kiriş ve 6 adet döşeme bulunmaktadır. İncelenen bina göreceli olarak +küçük bir yapı olsa da tipik bir müstakil konut boyutlarındadır. Veri girişi biçiminden dolayı kolon-döşeme komşuluklarının tespitine gerek kalmamaktadır. Çünkü döşemelerin tanımlanması, kendisini çevreleyen kolonlarla gerçekleştirildiği için kolon-döşeme komşulukları herhangi bir sorgulama yapmadan belirlenmektedir. Kolon-kiriş komşulukları ise her kolona sapanan kirişin çift döngü ile önce tüm kolonların ardından tüm kirişlerin taranması ile tespit edilmektedir. Kolona sapanan kiriş sayısı kesin bir rakam olmayacağı için sapanan kiriş sayısı ne olursa olsun tüm döngü tüm kirişler için çevrilmektedir. Vaka analizinde $12 \times 17 \times 2 = 408$ adet kolon ve kirişin temas edip etmediği sorgulaması yapılarak kolon-kiriş komşulukları sorgulanmıştır. Sorgu sayısının 2 ile çarpılmasının nedeni

kirişlerin başlangıç ve bitiş düğüm noktası olmak üzere 2 adet düğüm noktasına sahip olması ve her 2 düğüm noktasının da karşılaştırılmasıdır. Kiriş-döşeme komşuluğunun tespiti ise tüm kirişlerin başlangıç ve bitiş düğüm noktasının aynı döşemeyi çevreleyen düğüm noktaları arasında olması durumunun sorgulanması ile gerçekleştirilir. Bu şekilde $6*17*2 = 204$ adet sorgulama yapılarak kiriş-döşeme komşulukları tespit edilmiştir.

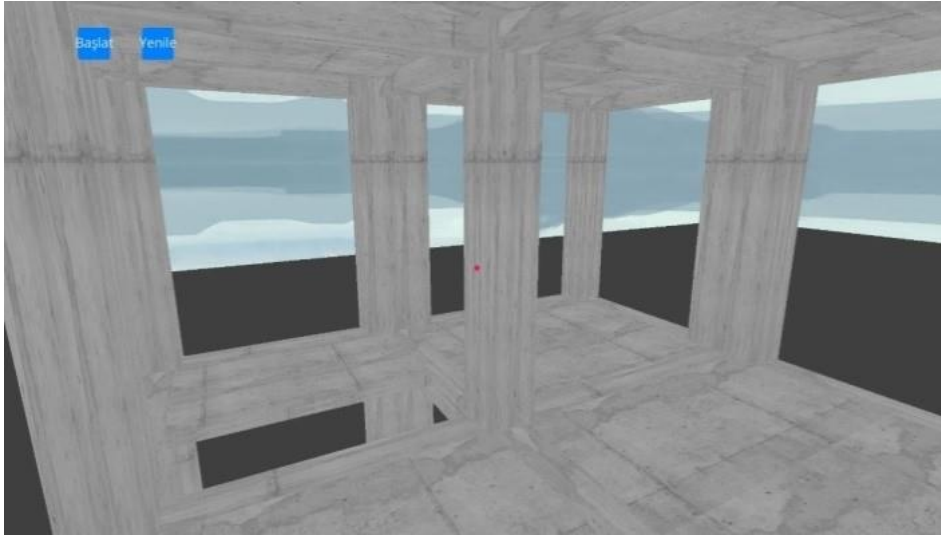
Metraj hesaplamaları gerçekleştirilen yapının 3 Boyutlu gösterimi gerçekleştirilmiş ve Şekil 17 ve 18’de sunulan beton dokusu eklenmiş ve eklenmemiş görüntüler elde edilmiştir. Ayrıca binanın içinde yürüme görüntüsü Şekil 19’da sunulmuştur.



Şekil 17. Binanın beton dokusuz 3B modeli
Figure 17. 3D model of the building without concrete texture

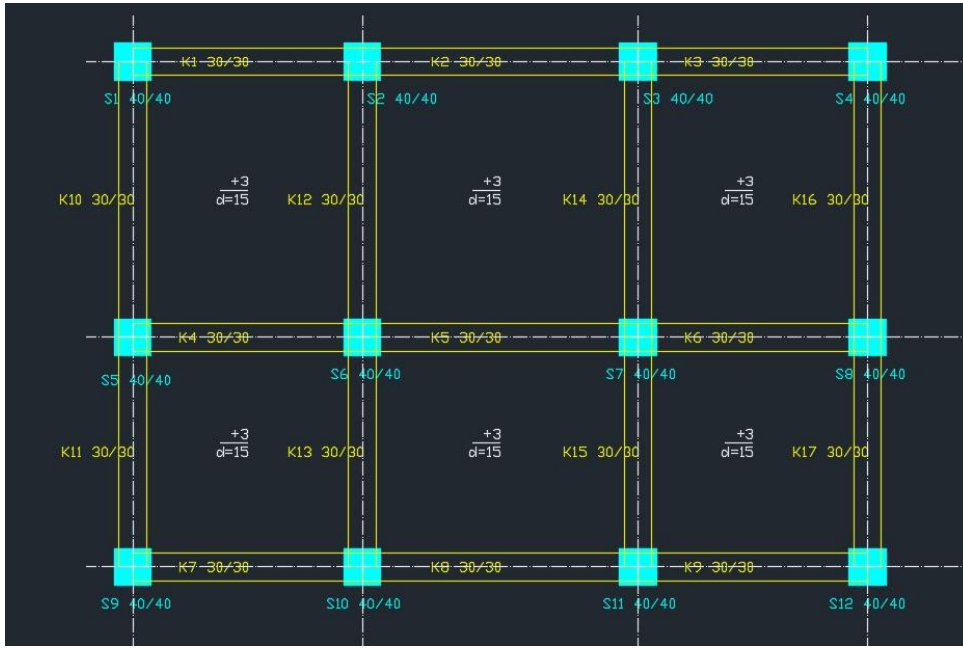


Şekil 18. Binanın beton dokulu 3B modeli
Figure 18. 3D model of the building with concrete texture



Şekil 19. Yapının içinde beton dokulu hali ile gezinme
Figure 19. Walk around the building with concrete texture

Çizim dosyası dxf uzantısı olarak dışarı aktarılabilmekte ve ticari bilgisayar destekli çizim veya yapı bilgi modellemesi yazılımları tarafından açılabilir. Bu sayede diğer ticari yazılımlarla veri aktarımının yapılabilmesi sağlanmıştır. Oluşturulan çizimler dxf dosya uzantısı olarak kaydedilmiş ve AutoCAD yazılımı ile açılmıştır. Elde edilen görünüm Şekil 20’de sunulmuştur.



Şekil 20. Çizimlerin dışı aktarılıp başka ticari yazılımla açılması
Figure 20. Extraction of the drawings and opening them with another commercial software

Vaka analizinde incelenen yapının temel, zemin katı ve birinci kat için kaba inşaat iş kalemlerinin metrajı hesaplanmış ve elde edilen değerler Tablo 1’de sunulmuştur. Donatı metrajının hesaplanması bu çalışmanın kapsamı dışındadır.

Elde edilen metraj sonuçları elle hesaplanarak kontrol edilmiş ve sapma miktarları Tablo 2’de sunulmuştur. Aynı yapı metraj hesaplaması yapan yerel 2 yazılımla incelenmiş ve yazılımların sunduğu metraj değerlerinin sapma oranları Tablo 2’de sunulmuştur. İncelenen yazılımlar kalıp iskelesi metrajı hesaplamadıkları için bu iş kaleminin karşılaştırması yapılamamıştır.

Tablo 1. Vaka analizi metraj sonuçları**Table 1.** Results of quantity take-off of case study problem

Eleman yeri	Eleman	Metraj Kalemleri		
		Beton (m ³)	Kalıp (m ²)	Kalıp iskelesi (m ³)
Temel	Sömel	6.000	18.90	
Temel	Bağ Kiriş	4.950	33.00	
Temel	Toplam	10.950	51.90	
Zemin Kat	Kolon	5.760	54.21	
Zemin Kat	Kiriş	3.888	30.14	
Zemin Kat	Döşeme	5.798	38.66	
Zemin Kat	Toplam	15.446	123.00	127.054
Birinci Kat	Kolon	5.760	54.18	
Birinci Kat	Kiriş	3.888	29.82	
Birinci Kat	Döşeme	5.997	39.98	
Birinci Kat	Toplam	15.645	123.98	134.355

Elde edilen metraj sonuçları elle hesaplanarak kontrol edilmiş ve sapma miktarları Tablo 2’de sunulmuştur. Aynı yapı metraj hesaplaması yapan yerel 2 yazılımla incelenmiş ve yazılımların sunduğu metraj değerlerinin sapma oranları Tablo 2’de sunulmuştur. İncelenen yazılımlar kalıp iskelesi metrajı hesaplamadıkları için bu iş kaleminin karşılaştırması yapılamamıştır.

Tablo 2. Geliştirilen yazılım ile yerel yazılımların metraj hata oranlarının karşılaştırılması**Table 2.** Comparison of the quantity error rates of the developed software and local software

Hata Oranı (%)			
Metraj kalemi	Geliştirilen yazılım	Yerel yazılım 1	Yerel yazılım 2
Beton	0.00%	-0.301%	0.019%
Kalıp	0.01%	3.795%	-0.246%

5. Bulgular

5. Results

Bu çalışmada geliştirme aşamasında olan bir YBM yazılımının mevcut durumu vaka analizi çalışması ile denenmiş ve sonuçları paylaşılmıştır. Geliştirilmekte olan YBM yazılımı mevcut haliyle başarılı biçimde 3 Boyutlu görselleştirmeyi gerçekleştirebilmektedir. Buna ek olarak kesin doğrulukta beton, kalıp ve kalıp iskelesi metrajlarını hesaplayabilmektedir. Yazılım ekranda donma olmadan bina içinde gezinme imkânını sunmaktadır. Görselleştirme özelliği sayesinde tasarım hatalarını belirleme, çakışma analizi ve sunum amaçlı kullanıma potansiyeli bulunmaktadır. Fakat yazılımın ticari özelliği kazanabilmesi için önemli iyileştirmelerin yapılması gereklidir.

Geliştirilme aşamasında olan yazılım kullanılarak yapılan beton, kalıp ve kalıp iskelesi metrajları ayrıca manuel olarak hesaplanmış ve birebir aynı sonuçların elde edildiği görülmüştür. Yazılımda kullanılan kesişim tanımlama algoritması kolon-kiriş, kolon-döşeme ve döşeme-kiriş kesişimlerini tanımlamakta ve kesişim alanlarını doğru biçimde hesaplamaktadır. Bu sayede kalıp, beton ve kalıp iskelesi metrajları doğru biçimde hesaplanabilmektedir.

Örnek çalışmada taşıyıcı sistem düzgün geometrili olacak şekilde tasarlanmıştır. Döşemenin veya kolonların daire şeklinde olması durumunda kesişim alanları tam doğru biçimde belirlenememekte ve metraj değerlerinde gerçek değerlerden küçük miktarlarda sapmalar olmaktadır. Belirtilen eksikliğin giderilmesi ileri çalışma olarak gerçekleştirilecektir.

6. Tartışma

6. Discussion

Geliştirilmekte olan YBM yazılımının mevcut durumunda duvar, sıva, boya, yer döşemesi, doğrama ve çatı çizimlerinin yapılabilmesi, belirtilen iş kalemlerinin 3 Boyutlu görselleştirmesinin yapılması ve belirtilen kalemlerinin inşaat aşamalarında bina içinde yürünmesi özellikleri bulunmamaktadır. Belirtilen özelliklerin yazılıma kazandırılması ileri çalışma olarak tasarlanmaktadır.

Mevcut haliyle beton, kalıp ve kalıp iskelesi metrajları hesaplanabilmektedir. Fakat donatı, duvar, sıva, boya, yer döşemesi, doğrama ve çatı metrajları hesaplanmamaktadır. Mimari elemanların metrajını yüksek doğrulukta hesaplayabilmek için çok daha kapsamlı biçimde komşulukların belirlenip çakışma analizinin yapılması gereklidir. Belirtilen metraj kalemlerinin hesaplanabilmesi ileri çalışma olarak tasarlanmaktadır.

YBM yazılımları yapısal elemanlara çok gelişmiş düzeyde öznitelik verilerinin tanımlanabilmesine olanak vermektedir. Geliştirilmekte olan yazılımın mevcut durumunda fiyat bilgisi ve betonun mekanik özellikleri tanımlanabilmektedir. Öznitelik verisi olarak renk, tedarikçi, kullanılan malzemelerin özellikleri, işçilik gereksinimleri, ısı geçirgenliği, garanti süresi ve benzeri verilerin girilebilmesi, saklanabilmesi, silinebilmesi ve değiştirilebilmesi özelliklerinin kazandırılması tasarlanmaktadır.

İnce inşaat işlerinin metrajı ve görselleştirmelerinin yapılması, iş programının kritik yol ile oluşturulması, takvimli iş programının oluşturulması, inşaat sırasında günlük harcanan malzeme ve yakıt gibi kalemlerin muhasebesinin tutulması ileri çalışma olarak yazılıma eklenecektir. Belirtilen araçların eklenmesi ile geliştirilen yazılımın yetenekleri ve kullanım alanları arttırılabilecek ve daha faydalı olabilecektir.

Metraj hesaplanabilmesi için gerçekleştirilen komşuluk analizlerinde toplamda binin altında sorgulama yapılmıştır. Gerçekleştirilen her sorgulama işlemi 2 adet tamsayının eşit olup olmadığının karşılaştırmasından ibarettir. Bu işlemin hesap yükü günümüz bilgisayarlarının hesaplama hızı dikkate alındığında oldukça düşüktür. Fakat sorgu sayısı yaklaşık olarak binanın aks sayısının karesi ile aynı hızda artmaktadır. Bu durumda Türkiye’de inşa edilen standart bir apartman dikkate alındığında on bin ile yüz bin arasında karşılaştırma yapılarak tüm yapısal elemanların komşulukları tespit edilebilecektir. Bu işlemin hesap yükü ve bellek gereksinimi vasat bir bilgisayarın kapasitesi dikkate alındığında sorunsuz biçimde gerçekleştirilebileceği görülmektedir. Yazılımın eksik olan özelliklerin giderilmesi ile işlevsel ve ticari değeri olabilen yerli bir YBM yazılımı geliştirilmiş olacaktır.

7. Sonuç

7. Conclusion

Bu makalede ilk örnek YBM yazılımının geliştirilebilmesi için yapılan çalışmalar özetlenmiştir. Kaba inşaatın metraj hesaplamasında kullanılan algoritma yüksek doğrulukta kesişim noktalarını tespit etmektedir. Bu sayede geliştirilen yazılım betonarme bir binanın beton, kalıp ve kalıp iskelesi metrajlarını Tablo 2’de yapılan karşılaştırmada sunulduğu üzere mevcut yazılımlardan daha yüksek doğrulukta yapabilmektedir. Ayrıca binanın kaba inşaatının 3 boyutlu görüntüsü oluşturulup binanın içinde görüntüde donma ve takılma olmadan sanal gezinti yapılabilir. Bu çalışmada geliştirilen yazılımın yeteneklerinin iyileştirilmesi için ince iş kalemlerinin de metrajının yüksek doğrulukta hesaplanması ve 3 Boyutlu görüntülerinin oluşturulması üzerinde durulacaktır. Ayrıca yazılıma daha fazla öznitelik bilgilerinin tanımlanmasını, saklanmasını, sorgulanmasını ve işlenmesini sağlayacak veritabanının oluşturulması, işçilik gereksinimlerinin hesaplanması, iş programının oluşturulması ve stok takibinin yapılması özelliklerinin eklenmesi de olumlu olacaktır.

Yazar katkısı

Author contribution

Bu araştırmada; Ergen, kodlamaların gerçekleştirilmesi; Bettemir, metraj hesaplama algoritmasının oluşturulmasını gerçekleştirmiştir.

Etik beyanı*Declaration of ethical code*

Bu makalenin yazarları, bu çalışmada kullanılan materyal ve yöntemlerin etik kurul izni ve / veya yasal-özel izin gerektirmediğini beyan etmektedir.

Çıkar çatışması beyanı*Conflicts of interest*

Yazarlar herhangi bir çıkar çatışması olmadığını beyan eder.

Kaynaklar*References*

- Albahbah, M., Kıvrak, S., & Arslan, G. (2021). Application areas of augmented reality and virtual reality in construction project management: A scoping review. *Journal of Construction Engineering Management & Innovation*, 4(3), 151-172, <https://doi.org/10.31462/jcemi.2021.03151172>.
- Barlsh, K., & Sullivan, K. (2012). How to measure the benefits of BIM—a case study approach. *Automation in construction*, 24, 149-159.
- Bayram, S. (2020). Yapı bilgi modellemesi (YBM) kapsamında geleneksel metraj ile yazılımın karşılaştırılması, *Yapı Bilgi modelleme*, 2(2), 58-65.
- Bettemir, Ö. H. (2018). Development of spreadsheet based quantity take-off and cost estimation application. *Journal of Construction Engineering Management & Innovation*, 1(3), 108-117, <https://doi.org/10.31462/jcemi.2018.03108117>.
- Bettemir, Ö.H., Gündüz, E., Akkurt, O., Hilal, E., & Arslan, M.A. (2019). İnşaat işlerinin iş programına bağlı nakit akışı değişkenliğinin saptanması ve düzenlenmesi. *Mühendislik Bilimleri ve Tasarım Dergisi*, 7(1), 211-223, <https://doi.org/10.21923/jesd.459948>.
- Bettemir, Ö.H. & Yücel, T. (2021) Zaman maliyet ödünleşim probleminin en az insan müdahalesi ile oluşturulup çözülmesi. *Uludağ Üniversitesi Mühendislik Fakültesi Dergisi*, 26(2), 461-480, <https://doi.org/10.17482/uumfd.869234>.
- Bettemir, Ö.H. & Bulak, Ö. (2022). İnşaat sürecinin iş çizelgelemesi, yönetimi ve optimizasyonu. *Teknik Dergi*, 33(6), 12945-12986, <https://doi.org/10.18400/tekderg.981601>.
- Bettemir, Ö. H., Özdemir, E., & Sarıcı, D. E. (2021). Selection of construction machine by considering time-cost trade-off problem. *Journal of Construction Engineering Management & Innovation*, 4(3), 173-186, <https://doi.org/10.31462/jcemi.2021.03173186>.
- Choi J., Kim H. & Kim I. (2015). Open BIM-based quantity take-off system for schematic estimation of building frame in early design stage. *Journal of Computational Design and Engineering*, 2(1):16-25, <https://doi.org/10.1016/j.jcde.2014.11.002>.
- Delgado J.M.D., Oyedele L., Demian P., & Beach T. (2020). A research agenda for augmented and virtual reality in architecture, engineering and construction. *Advanced Engineering Informatics*, 45, 101-122. <https://doi.org/10.1016/j.aei.2020.101122>.
- Dhalmahapatra K., Maiti J., & Krishna O.B. (2021). Assessment of virtual reality-based safety training simulator for electric overhead crane operations. *Safety Science*, 139, 105241, <https://doi.org/10.1016/j.ssci.2021.105241>.
- Ergen, F., & Bettemir Ö.H. (2022). Development of BIM software with quantity take-off and visualization capabilities. *Journal of Construction Engineering Management & Innovation*, 5(1), 01 – 14, <https://doi.org/10.31462/jcemi.2022.01001014>.
- Kalfa, S. M. (2018). Building information modeling (BIM) systems and their applications in Turkey. *Journal of Construction Engineering Management & Innovation*, 1(1), 55-66.

- Khosakitchalert C., Nobuyoshi Yabuki, & Fukuda T. (2019b). Automatic concrete formwork quantity takeoff using building information modeling. *Proceedings of the 19th International Conference on Construction Applications of Virtual Reality (CONVR2019)* pp. 21-28, 13 – 15, November.
- Khosakitchalert, C., Yabuki, N., & Fukuda, T. (2018). The accuracy enhancement of architectural walls quantity takeoff for schematic BIM models. In *ISARC, Proceedings of the International Symposium on Automation and Robotics in Construction*, Berlin, Germany.
- Khosakitchalert, C., Yabuki, N., & Fukuda, T. (2019a). BIM-based wall framing calculation algorithms for detailed quantity takeoff, *Proceedings of the 4th International Conference on Civil and Building Engineering Informatics*, 251-258, Sendai, Miyagi, Japan.
- Khosakitchalert, C., Yabuki, N., & Fukuda, T. (2019). Improving the accuracy of BIM-based quantity takeoff for compound elements. *Automation in Construction*, 106, 1-20, <https://doi.org/10.1016/j.autcon.2019.102891>.
- Kim S., Chin S., & Kwon S. (2019). A discrepancy analysis of BIM-based quantity take-off for building interior components. *Journal of Management in Engineering*, 35(3):05019001.
- Lee S.K., Kim K.R., & Yu J.H. (2014). BIM and ontology-based approach for building cost estimation. *Automation in Construction*, 41, 96-105, <https://doi.org/10.1016/j.autcon.2013.10.020>.
- Liu H., Cheng J.C., Gan V.J. & Zhou S. (2022). A knowledge model-based BIM framework for automatic code-compliant quantity take-off. *Automation in Construction*, 133:104024, <https://doi.org/10.1016/j.autcon.2021.104024>.
- Liu H., Lu M., & Al-Hussein M. (2016). Ontology-based semantic approach for construction-oriented quantity take-off from BIM models in the light-frame building industry. *Advanced Engineering Informatics*, 30(2), 190-207.
- Millman, K. J., & Aivazis, M., (2011). Python for scientists and engineers. *Computing in Science & Engineering*, 13(2), 9-12.
- Nadeem, A., Wong, A. K., & Wong, F. K. (2015). Bill of quantities with 3D views using building information modeling. *Arabian Journal for Science and Engineering*, 40(9), 2465-2477.
- Olsen D., & Taylor J. M. (2017). Quantity take-off using building information modeling (BIM), and its limiting factors. *Procedia Engineering*, 196:1098-1105.
- Omaran, S. M., Alghamdi, A. A., Alharishawi, S. C., & Hains, D. B. (2019). Integrating BIM and game engine for simulation interactive life cycle analysis visualization. In *Computing in Civil Engineering 2019: Visualization, Information Modeling, and Simulation*, 120-128, American Society of Civil Engineers, Reston, VA.
- Ozcan-Deniz, G. (2019). Expanding applications of virtual reality in construction industry: A multiple case study approach. *Journal of Construction Engineering Management & Innovation*, 2(2), 48-66.
- Pratoom, W., & Tangwiboonpanich, S. (2016). A comparison of rebar quantities obtained by traditional vs bim-based methods. *Suranaree Journal of Science and Technology*, 23(1), 5-10.
- Quraishi A, & Dhapekar N. K. (2021). Applicability of python in civil engineering, *International Research Journal of Engineering and Technology (IRJET)*, 8(1), 554-556.
- Sacks, R., Brilakis, I., Pikas, E., Xie, H. S., & Girolami, M. (2020). Construction with digital twin information systems. *Data-Centric Engineering*, 1, e14, 1-26, <https://doi.org/10.1017/dce.2020.16>.
- Sacks, R., Koskela, L., Dave, B. A., & Owen, R. (2010). Interaction of lean and building information modeling in construction. *Journal of construction engineering and management*, 136(9), 968-980.
- Sacks R., & Barak R. (2007). Impact of three-dimensional parametric modeling of buildings on productivity in structural engineering practice. *Automation in Construction*, 17, 439-449, <https://doi.org/10.1016/j.autcon.2007.08.003>.
- Sari, R., & Pekerçli, M. K. (2020). An investigation of comparison and evaluation of official BIM documents released in the USA, UK and Turkey. *Journal of Construction Engineering Management & Innovation*, 3(1), 67-84.

- Sarvade S.M. & Pore S.M. (2019). Use of python programming for interactive design of reinforced concrete structures, *National Conference on Exploring New Dimensions in Teaching Learning for Quality Education, Maharashtra, India*, 160–163.
- Sherafat, B., Taghaddos, H., & Shafaghat, E. (2022). Enhanced automated quantity take-off in building information modeling enhanced automated quantity take-off in building information modeling. *Scientia Iranica*, 29(3), 1024-1037.
- Temel, B., & Basaga, H. (2020). Investigation of IFC file format for BIM based automated code compliance checking. *Journal of Construction Engineering Management & Innovation*, 3(2), 113-130, <https://doi.org/10.31462/jcemi.2020.02113130>.
- Yoon, J. D., Cho, H. S., Lee, J. H., Shin, J. Y., & Kim, E. S. (2020). A comparison of quantity take-offs of RC structures based on BIM. *Journal of the Computational Structural Engineering Institute of Korea*, 33(1), 35-44.
- Zuppa, D., Issa, R. R., & Suermann, P. C. (2009). BIM's impact on the success measures of construction projects. *In Computing in Civil Engineering*, 503-512.