|  |
| --- |
| **Projenin Adı ve Süresi** |
| **Adı**: Finansal UI Komponentleri ve Akıllı Entegrasyon Çözümleri.  **Süresi** :  **AR-GE İş Gücü :** |

|  |
| --- |
| **Anahtar Kelimeler** |
| WPF, Telerik UI, SignalR, Redis, Entegrasyon, Spring, MyBatis, Oracle, EHCache |

|  |
| --- |
| **Proje Özeti** |
| Bu projeyle finansal dünyada en sık karşılaşılan 2 genel probleme çözüm bulmak amaçlanmıştır. Bu problemlerden ilki önyüzde kullanılan ekran bileşenlerin birçok değişik frameworkte standart veri tabanı aplikasyonları ve benzeri problemlerin çözümü için tasarlanmış olmasıdır. Aslında ihtiyaç kendi beslemesini push teknolojileriyle yapabilen ve hızlı değişen dataya ayak uydurabilecek performanslı ve de özellikle finansal datayı kullanıcıdan daha kolay yakalayabilen ekran bileşenleri geliştirmektir. Bir diğer problem de gelişen teknolojik ensturmanlarla beraber entegrasyon noktalarındaki aşırı artışı yönetebilecek konfigürasyona dayalı hızlı entegrasyon modelidir. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | |  | |  |  |  | |  |  |  |  | |  |  |  |  | |  | |  | |  |  |  |  | | | | |  | |  |  |  | |  | |  |  |  | | |  |  | |  |
| **Hedeflenen Kazanım ve Sonuçlar** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Mevcut finans yazılımları incelendiğinde genellikle kullanılan ekran bileşenlerinin üzerinde geliştirilen frameworke ait standart bileşenler olduğu gözlemlenmektedir. Finans dünyasının yapısı çok hızlı ve değişkendir. Özellikle saniyelerin karar vermekte etkili olduğu birçok uygulama mevcuttur. Bunlara örnek olarak Hazine yazılımlarını da gösterebiliriz. Bu bağlamda, buradaki ihtiyaçlar da çok farklılaşmaktadır. Datanın belli bir hızda değişkenliğinin gözlemlenebilmesinden, işlem girişinde kullanılan ekran komponenlerinin ihtiyaca yönelik olarak customize edilebilmesine veya belli bir yükteki datayı daha kolay analiz edebilmeye yaraması gerekliliği mevcuttur. Bu gibi ihtiyaçlar ulusal ve uluslararası piyasalardan kaynak arama yönündeki faaliyetlerin maliyetlerini dolaylı yoldan etkilemektedir.  Projenin bu ayağıyla beraber WPF üzerine yazılan Telerik UI ile geliştirilmiş standart ekran bileşenleri kullanılarak finansal ihtiyaçlara için customize edilmiş ekran bileşenleri yazılacaktır. Bu komponentler ayrıca SignalR yardımıyla push teknolojilerinin kullanılmasına olanak sağlayacaktır. Bu bileşenler değişik projeler tarafından tekrar tekrar yazılmaya olan ihtiyacı gidereceğinden, uygulama geliştirme maliyetlerini ve süresini düşürecektir. Ayrıca kullanıcıların karar mekanizmalarını yürütmesi sırasında daha hızlı hareket edebilmesini sağlayacaktır.  Projenin diğer ayağı ile ise yine finans dünyasında ihtiyaç hissedilen çoklu entegrasyon noktalarını konuşturmada harcanan zamanı ve maliyetleri azaltmaya yöneliktir. Sadece finans dünyasında değil malum bilgi teknolojileri sistemlerinde artık birçok sistemin bir araya gelerek entegre olması gerekiyor. Bu çoğu zaman ESB üzerinden alanların birbirine eşleşmesiyle sağlanıyor ve bu da çoğu zaman maliyetli ve esneklikten uzak bir şekilde gerçekleşiyor. Burada asıl problemse bu dosya boylarının çok büyük haller alması ve gerçekte bu datanın bir kısmının sadece gerçek ihtiyaç olmasına rağmen bütün dosyanın parse edilerek harcanan zaman kaybıdır. Projenin bu ayağıyla beraber zamanın çok kıymetli olduğu finansal sistemler içinde event modelli ve konfigürasyona dayalı bir entegrasyon sistemi kurularak datanın hızlı bir şekilde sistemlere generic olarak entegre edilmesi sağlanacaktır. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | |  | |  |  |  | |  |  |  |  | |  |  |  |  | |  | |  | |  |  |  |  | | | | |  | |  |  |  | |  | |  |  |  | | |  |  | |  |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Ar-Ge Niteliği** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | Redmound Enterprise Framework v2 sunacağı hazır, kolay entegre edilebilen ve özelleştirilebilen esnek modülleri aracılığıyla kurumsal bir web uygulamasının gereksinimlerinin hızlı bir şekilde karşılanmasına olanak sağlayacaktır. Mevcut açık kaynak kodlu ve lisanslı ürünler doğrudan hazır bir altyapı sunmamaktadır. Redmound projemiz ve Spring, Jboss Seam, RichFaces, ICEFaces, PrimeFaces vb. diğer birçok framework sadece uygulama geliştirecek bir yapı sunmaktadır. Uygulama geliştiriciler sunulan bu imkanlar ile kendi ihtiyaçlarına göre yetkilendirme, kullanıcı yönetimi, performans izleme ve iş akışlarını gerçekleştirmektedirler. Redmound Enterprise Framework projemiz bu yapıya ek olarak standartları belirlenmiş, ortak kullanılabilecek modüller sunmaktadır. Redmound Enterprise Framework v2 projemiz ile kurumsal web tabanlı yazılım geliştirme aşamasında esnekliği azaltmadan konfigürasyon yükünü azaltarak (Convention over Configuration) geliştiricileri karmaşık yapılandırma işlemlerinden kurtaran geliştirici/geliştirme verimliliğini önemli derecede arttıracak ek araçlar, özellikler ve modüller sağlanacaktır.  Kullanıcıların ve kullanıcı yetkilerinin admin arayüzleri ile dinamik olarak yönetimi web uygulamaları için standart bir gereksinimdir. Tekrarlı olarak ekranların, süreçlerin ve ilişkili altyapının kurgulanması projeler için ciddi bir maliyet oluşturmaktadır. Projelerde harcanan zaman ve efor değerlendirildiğinde Redmound Enterprise Framework v2 sunacağı özelleştirilebilir ekranlar ve altyapı ile birlikte uygulama geliştirme sürecini hızlandıracak, yazılım geliştirme ekiplerine/süreçlerine çeviklik kazandıracaktır.  Kurumsal web uygulamaları üzerinden verilerin oluşturulması, sunulması, güncellenmesi ve silinmesi amacıyla her bir veri tipi özelinde tekrarlı olarak model, servis ve ekran tasarlama/geliştirme süreçleri işletilmektedir. İşletilen bu süreçler neticesinde elde edilen yapıların birçok ortak özellikleri bulunmaktadır. Redmound Enterprise Framework v2 projemiz bünyesinde tasarlanacak olan eclipse sihirbazı ile veri tabanı modeli üzerinden standardize edilmiş ve talepler doğrultusunda kolaylıkla özelleştirilebilen modeller, servisler ve ekranlar otomatik olarak oluşturulacaktır (scaffolding). Bu sayede uygulama geliştiricilerin tekrar eden bu süreçler yerine işletilecek olan iş süreçleri üzerine yoğunlaşmalarına imkan sağlanacaktır. Hataya açık tekrar eden süreçlerin otomatize ve standardize edilmesiyle test ve geliştirme aşamalarında zaman ve efor kazancı elde edilecektir.  Geliştirme süreçlerinin hızlandırılması, uzmanlık gereksiniminin azaltılması ve kurumsal web uygulamalarının hayata geçiş sürelerinin kısaltılması amacıyla uygulamalar tarafından genel olarak ihtiyaç duyulan ekranlar belirlenecek ve kolaylıkla özelleştirilebilen taslak sayfalar oluşturulacaktır. Redmound Enterprise Framework v2 projemiz kapsamında tasarlanacak olan eclipse sihirbazı ile uygulama geliştiriciler ihtiyaç duydukları ekranları kolaylıkla projelerine dahil edebileceklerdir.  Projenin geliştirme kolaylığı ve bakım yapılabilirliği açılarından kritik önem taşıyan proje yapısının belirlenmesi ve oluşturulması ihtiyaçları geliştirilecek olan prototipler ile karşılanacaktır. Bu kapsamda farklı ihtiyaç kümeleri özelinde maven prototipleri(archetype) oluşturulacaktır. Prototip kullanımı vasıtasıyla proje başlangıcında elde edilecek hız ile projeler daha kısa sürede hayata geçirilebilecektir.  Kurumsal web uygulamaları sahip oldukları özellikleri dış sistemlerin kullanımına açmak amacıyla bünyelerinde yüksek sayıda servis barındırmaktadır. Aktif olarak kullanımda olan servislerin belirlenmesi, bağımlılıkların/entegrasyon noktalarının kolaylıkla saptanabilmesi ve yeni bir servis geliştirileceği durumda ihtiyaçları karşılayacak servislerin olup olmadığının kolaylıkla saptanabilmesi için geliştirilen servislerin standartlara uygun olarak kayıt altına alınması önemli bir ihtiyaç olarak ortaya çıkmaktadır. Bu kapsamda Redmound çatısı ile geliştirilen uygulamalara kolaylıkla entegre edilebilen bir servis kayıt modülü tasarlanacak ve geliştirilecektir.  "Üye girişi" fonksiyonu geliştirilen sistemlerin büyük bir çoğunluğu için ana gereksinimler arasında yer almaktadır. Bu ana gereksinimin karşılanması amacıyla genellikle hali hazırda kullanılan ldap - active directory ya da SSO sistemi ile entegrasyon sağlanması gerekmektedir. Sosyal medya vasıtasıyla "üye girişi" gerçekleştirimi de sıklıkla tercih edilen yöntemlerdendir. Üyelik sistemlerinin geliştirilen uygulamalara eklenmesinden doğan tekrarlı efor ve zaman kaybının önüne geçilmesi amacıyla ihtiyaçlar doğrultusunda kolaylıkla konfigüre edilebilen ek bileşenler geliştirilecektir.  Gelişen web teknolojileri ve artan kullanıcı ihtiyaçları/beklentileri ile birlikte istemci tarafında uygulanan javascript tabanlı çözümlerin kompleksitelerinin artması istemci tarafı bağımlılık yönetimini bir ihtiyaç haline getirmektedir. Redmound Enterprise Framework v2 projesi kapsamında IoC pirensibinin bir gerçekleştirimi olan bağımlılık enjeksiyonu(dependency injection) tasarım şablonunun istemci tarafında uygulanması ile bileşenler arası ilişki yönetimi sorumluluğu framework tarafından yerine getirilecektir. DI tasarım şablonunun istemci tarafında uygulanması kod tekrarının önlendiği, bakım yapılabilirliği ve test edilebilirliği yüksek ve daha modüler bir ön yüz katmanı elde edilmesini sağlayacaktır. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | **Yenilikçi Yönü** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | Redmound Enterprise v2, hem daha önce tamamlamış olduğumuz Redmound ve Redmound Enterprise projelerini bir adım daha ileri götürmesi hem de piyasadaki benzer ürünlerin cevap veremediği bazı teknik eksiklikleri gidermesi bakımından yenilikler içermektedir:  • Entegre Html, CSS, Javascript birleştirme, sıkıştırma ve kod karıştırma süreçleriyle rendering aşamasında | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Sayfa (4 / 8) | | | | | | | | | | | | | | | | |  | |  |  |  | |  |  |  |  | |  |  |  |  | |  | |  | |  |  |  |  | | | | |  | |  |  |  | |  | |  |  | | | | | | | |
| **Ar-Ge Niteliği** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Proje, birçok değişik uygulama tarafından ortak bir altyapı çerçevesinde kullanılacağından Ar-Ge niteliği taşımaktadır.  Pazar araştırıldığında proje için kullanılacak bu custom ekran bileşenlerinin tanımlanan teknolojilerle geliştirilmesi bu alanda denenmemiş bir yöntemdir. Proje süreci boyunca kullanıcılara “user interaction” testleri uygulanıp datanın en verili şekilde yakalandığına emin olunacaktır. Ayrıca bu altyapıyla geliştirilen push method sürelerinin takibi ve ekrana yansıma süreleri ölçülecektir.  Müşteri ihtiyaçlarına ve tercihlerine göre bu bileşenler üzerinden değişimler yapılabilmesine olanak sağlayan bir yapı da kurulacaktır.  Diğer taraftan da kurulan bu entegrasyon yapısı aslında sadece finans dünyasında değil, tüm SOA uygulamalarında kullanılabilecek esneklik sağlayacak şekilde düşünülmüştür. Bu sebeple oluşan bu yapı, diğer benzer projelerde de entegrasyon sürelerini ciddi miktarda düşürecektir. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  | |  | | |  |  | | | | |  | | | | |  | |  | |  | | | | |  |  |  |  | |  | | |  |  | |  | | | |  |  | | |  | |
| **Yenilikçi Yönü** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Günümüzde önyüz teknolojilerine yönelik bir çok değişik component kütüphaneleri olmasına rağmen bu spesifik ihtiyacı karşılayacak tanımlanmış teknoloji yapısında bir ürün bulunmamaktadır. Özellikle bu customizable finansal ekran kütüphaneleri kullanıcı deneyimini baştan değiştirerek kullanım kolaylığı ve zaman kazanımını sağlayacaktır. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  | |  | | |  |  | | | | |  | | | | |  | |  | |  | | | | |  |  |  |  | |  | | |  |  | | | | | | | | | | | |

|  |
| --- |
| **Proje Detayı** |
| Malum olduğu üzere finans dünyası artık IT uygulamalarıyla beraber iç içe yürümektedir. Burada gerçek ihtiyaç daha hızlı aksiyon alarak piyasada fark yaratmaya çalışmaktır. Çünkü bu dünya için hız her şeydir ve maliyetler saniyelik düzeyde değişmektedir. Datanın gerçek zamanlı push teknolojileriyle gösterilmesi, sisteme hızlı entegre edilmesi ve datanın en esnek ve hızlı şekilde kullanıcıdan yakalanması çok büyük önem taşımaktadır. Bunu yaparken de uygulama maliyetlerini en az seviyeye çekmek birçok IT sistemi geliştiricisi için çok gerekli olmaktadır. Bu projeyle beraber hem entegrasyon maliyetleri azalacaktır hem de fark yaratacak önyüz bileşenleri kullanıma alınacaktır. |
|  |
|  |
| **Ekonomik Değeri** |
| Finans kurumlarının ihtiyacı olan gerek bu customize edilmiş bileşenleri kullanarak gerekse event modelli entegrasyon yapısını kullanarak zaman ve maliyetlerin azaltıldığı bir ürün hedeflenmiştir. |