****

**BALIKESİR ÜNİVERSİTESİ**

**İLERİ PROGRAMLAMA DERSİ**

**JAVA TABANLI SOHBET UYGULAMASI**

**GEREKSİNİM ANALİZ RAPORU**

**1. Proje Tanımı**

Bu proje, kullanıcıların odalara katılarak mesajlaşmalarını sağlayan bir chat uygulaması geliştirmeyi amaçlamaktadır. Uygulama, paralel programlama tekniklerini kullanarak eşzamanlı çoklu kullanıcı desteği sağlayacak, mesajları veritabanına kaydedip geçmiş mesajları kullanıcıya gösterecek ve kullanıcı dostu bir arayüze sahip olacaktır.

Chat uygulaması, Java tabanlı olacak ve PostgreSQL veritabanı yönetim sistemi ile çalışacaktır. Arka planda bir Socket server çalışacak, client tarafında ise Java GUI (örneğin, Swing veya JavaFX) kullanılacaktır.

**2. Gereksinimlerin Tanımı**

**2.1 Fonksiyonel Gereksinimler**

**2.1.1 Kullanıcı Kayıt ve Giriş Sistemi**

* Kullanıcılar uygulamaya kaydolabilecek ve giriş yapabilecek.
* Kullanıcı bilgileri veritabanında saklanacak (PostgreSQL).
* Giriş yapan kullanıcılar, sistemde aktif olarak tanımlanacak ve birden fazla kullanıcı aynı anda giriş yapabilecek.

**2.1.2 Oda Yönetimi**

* Kullanıcılar yeni bir sohbet odası oluşturabilecek veya mevcut odalara katılabilecek.
* Odalar şifreli veya şifresiz olabilir.
* Her odaya ait mesajlar PostgreSQL veritabanında saklanacak.
* Odaya katılan kullanıcılar, odadaki önceki mesajları görebilecek.

**2.1.3 Mesajlaşma Sistemi**

* Kullanıcılar bir odadayken mesaj gönderebilecek.
* Mesajlar anında tüm katılımcılara iletilecek.
* Gönderilen her mesaj, PostgreSQL veritabanına kaydedilecek ve timestamp (zaman damgası) ile saklanacak.
* Mesajlar odaya sonradan katılan kullanıcılara gösterilecek.

**2.1.4 Arayüz**

* Kullanıcı dostu bir grafik arayüz (GUI) olacak.
* Java Swing veya JavaFX ile görselleştirilmiş bir arayüz kullanılacak.
* Arayüzde:
  + Oda oluşturma,
  + Odaya katılma,
  + Mesajları görüntüleme ve gönderme işlevleri olacak.

**2.1.5 Bağlantı ve Çoklu Kullanıcı Desteği**

* Birden fazla kullanıcı aynı anda sisteme giriş yapabilecek ve farklı odalarda ya da aynı odada mesajlaşabilecek.
* Her kullanıcının mesajları, paralel programlama ile eşzamanlı olarak işlenecek.
* Kullanıcılar odadan çıkmadan önce gönderilen tüm mesajları görebilecek.

**2.2 Fonksiyonel Olmayan Gereksinimler**

**2.2.1 Performans**

* Uygulama aynı anda en az 50 kullanıcının bağlanabileceği şekilde optimize edilecek.
* Mesaj gönderim ve alma işlemleri gerçek zamanlı olacak.

**2.2.2 Güvenlik**

* Kullanıcı kayıt ve giriş işlemlerinde şifreleme kullanılacak (örneğin, BCrypt ile parolalar hash edilecek).
* Şifreli odalara giriş için güçlü parola kontrol mekanizması olacak.
* Tüm kullanıcı bağlantıları SSL ile güvenli hale getirilecek.

**2.2.3 Veri Yönetimi**

* PostgreSQL veri tabanı kullanılacak.
* Veri tabanı mesaj geçmişini, kullanıcı bilgilerini ve oda bilgilerini tutacak.
* Veri tabanı ile iletişimde paralel programlama ile eşzamanlı veri işlemi yapılacak.

**2.2.4 Taşınabilirlik**

* Uygulama platform bağımsız olacak. Windows, macOS ve Linux işletim sistemlerinde çalışabilecek.

**3. Teknik Gereksinimler**

**3.1 Yazılım Gereksinimleri**

* **Programlama Dili:** Java (versiyon 11+)
* **Socket Programlama:** Java Sockets API
* **Paralel Programlama:** Java ExecutorService, Thread, synchronized bloklar, CompletableFuture
* **Veri tabanı:** PostgreSQL (versiyon 12+)
* **JDBC**: PostgreSQL JDBC driver ile veritabanı bağlantısı.
* **Arayüz:** Java Swing veya JavaFX ile GUI geliştirme.
* **Şifreleme:** BCrypt (şifreleme işlemleri için)
* **IDE:** IntelliJ IDEA, Eclipse, NetBeans (tercihe bağlı)

**3.2 Donanım Gereksinimleri**

* Minimum **RAM**: 4GB (Tercihen 8GB)
* Minimum **İşlemci**: 2 çekirdekli işlemci (tercihen 4 çekirdekli)
* **Depolama**: 500MB disk alanı

**3.3 Veri tabanı Gereksinimleri**

* **PostgreSQL Veri tabanı**: Mesajların ve kullanıcı bilgilerin saklanması için.
* **Veri tabanı Yapısı**:
  + **users** tablosu: Kullanıcı bilgilerini (kullanıcı adı, şifre) saklar.
  + **rooms** tablosu: Odaların bilgilerini tutar.
  + **messages** tablosu: Her odada gönderilen mesajları saklar (mesaj, kullanıcı, zaman damgası).

**4. Paralel Programlama Kullanım Alanları**

**4.1 Client Bağlantı Yönetimi**

* Her client, bir thread havuzu kullanılarak bağımsız bir şekilde sunucuya bağlanacak. Bu sayede birden fazla kullanıcı eşzamanlı olarak mesajlaşabilecek.

**4.2 Mesaj Alma ve Gönderme**

* Her kullanıcının mesajları, paralel olarak işlenecek. Kullanıcı mesaj gönderdiğinde bu mesaj diğer kullanıcılara anında iletilecek ve aynı anda veri tabanına kaydedilecek.

**4.3 Veritabanı İletişimi**

* Mesajların ve kullanıcı hareketlerinin veri tabanına kaydedilmesi ve okunması işlemleri, paralel olarak gerçekleşecek. Veri tabanı işlemleri sırasında, çoklu kullanıcı veri yazma ve okuma sırasında "deadlock" gibi durumlar önlenecek.

**4.4 Sunucu Performansı**

* Sunucu, ExecutorService ile birden fazla client bağlantısını aynı anda işleyebilecek. Paralel programlama teknikleri, yüksek performans sağlamak için kullanılacak.

**5. Veri tabanı Tasarımı**

**5.1 Tablolar**

**5.1.1 users (Kullanıcı bilgilerini saklar)**

|  |  |  |
| --- | --- | --- |
| **Alan adı** | **Veri tipi** | **Açıklama** |
| id | SERIAL | Birincil anahtar |
| username | VARCHAR(50) | Kullanıcı adı |
| password | VARCHAR(255) | BCrypt ile hashlenmiş şifre |

**5.1.2 rooms (Sohbet odalarını saklar)**

|  |  |  |
| --- | --- | --- |
| **Alan adı** | **Veri tipi** | **Açıklama** |
| id | SERIAL | Birincil anahtar |
| name | VARCHAR(50) | Oda adı |
| password | VARCHAR(255) | (Opsiyonel) Şifrelenmiş oda parolası |

**5.1.3 messages (Mesajları saklar)**

|  |  |  |
| --- | --- | --- |
| **Alan adı** | **Veri tipi** | **Açıklama** |
| id | SERIAL | Birincil anahtar |
| room\_id | INT | Yabancı anahtar (rooms.id) |
| user\_id | INT | Yabancı anahtar (users.id) |
| message | TEXT | Mesaj içeriği |
| timestamp | TIMESTAMP | Mesajın gönderilme zamanı |

**6. Risk Analizi**

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk** | **Olasılık** | **Etki** | **Önlem** |
| Çoklu kullanıcı desteğinde performans sorunları | Orta | Yüksek | Thread havuzları ve optimizasyon |
| Veri kaybı | Düşük | Yüksek | Veri tabanı işlemlerini doğru yönetme |
| Bağlantı güvenliği | Orta | Yüksek | SSL/TLS kullanımı, güçlü şifreleme |
| Veri tabanı kilitlenmeleri | Orta | Orta | Paralel programlama ile deadlock önlenmesi |

**7. Sonuç**

Bu gereksinim analizi raporu, chat uygulamasının geliştirilmesinde izlenecek yol haritasını ortaya koymaktadır. Uygulama, Java programlama dili kullanılarak paralel programlama tekniklerinden tam anlamıyla faydalanacak ve PostgreSQL veri tabanı ile veri yönetimi sağlanacaktır. Kullanıcı dostu bir arayüz ve güvenli bir sistem altyapısı ile yüksek performanslı bir chat uygulaması geliştirilmesi hedeflenmektedir.