



HALIÇ ÜNİVERSİTESİ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

BMY549 - İleri Mobil Programlama

Final Raporu

Enes Cevik

23152010013@ogr.halic.edu.tr

İçindekiler

Ödev İçeriği	2
Uygulama Raporu	3
Uygulama Ekran Görüntüleri	6

Ödev İçeriği

- MVVM mimarisi ile ve ViewBinding kullanarak bir android **uygulaması geliştirilerek ve yüzyüze olarak** sunulacaktır.
- Proje 3 ekrandan oluşacaktır ve aşağıda linki bulunan api ile iletişim kurularak disney filmleri listelenecektir.
<https://gist.github.com/skydoves/aa3bbbf495b0fa91db8a9e89f34e4873/raw/a1a13d37027e8920412da5f00f6a89c5a3dbfb9a/DisneyPosters.json>
 - E1 - Splash Ekranı:
 - CounterDownTimer, Thread gibi yöntemlerle 3 saniye bu ekran gösterilip, 3 saniye sonunda E2 açılıp bu ekran ölecektir.
 - Tasarım size bırakılmıştır.
 - E2 - Disney Filmleri Listesi:
 - Bu ekranın çalışma şekli şöyle olacaktır; internet var ise internetten veriler çekilecek, eğer internet yok ise lokalde saklanan veriler basılacaktır. İnternet var ise ve lokalde verilerin bir kopyası saklanmadı ise veriler lokale de eklenecektir. Eğer hem internet yok hem de daha önceden lokale bir veri saklanmadı ise liste yerine ürünlerin yüklenmediğinde dair bir tasarım (Empty View) basılacaktır.
 - Burada ilgili karakterler **RecycleView** içerisine **CardView** olacak şekilde listelenecektir.
 - İstekler retrofit2 ile atılıp asenkronizasyon için Coroutine kullanılacaktır.
 - CardView içerisinde; id, poster resmi, karakter adı, playTime verileri olmalıdır.
 - Üst tarafında bir **searchview** olacak ve metin değıştikçe listeyi filtreleyecektir. Büyük küçük harfe duyarlı olacaktır.
 - Bu ekranda **geri tuşuna basılınca** kullanıcıya çıkmak isteyip istemediği sorulacak ve çıkışa basılırsa uygulama kapanacaktır. İptale basıldığında ise dialog kapatılacaktır.
 - Liste yüklenene kadar ve ProgressDialog ile lütfen bekleyiniz ibaresi gösterilecektir.
 - Bu ekranda üst tarafta bir sıralama butonu ve liste türü değıştirme butonu olacaktır.
 - Liste türü değıştir butonu tek buton olacaktır ve basıldıkça listeleme türü değışecektir.
 - Sıralama butonuna basıldığında ise **alert dialog ile liste açılacak**, listede isme göre artan ya da isme göre azalan sıra ile ve id ye göre artan azalan sıra ile listeleme yapılacaktır.
 - İlgili iteme tıklanınca detay verisi de eklenerek E3 ekranı açılacaktır.
 - E3 - Detay Ekranı:
 - İlgili item bu ekrana taşındıktan sonra Apideki diğer detayların tümü bu ekranda gösterilecek ve **nested scrool** olacaktır.

Uygulama Raporu

repo: <https://github.com/enescevik/IleriMobilOdev>

- Uygulama dili kotlin seçildi.
- Uygulamayı çalıştıracak min android api sürümü 32 olarak belirlendi.
- Uygulama id'si olarak com.halic.ilerimobilfinal kullanıldı.
- Uygulama içerisindeki tüm string ifadeler resource altındaki strings.xml altına eklendi.
- AndroidManifest.xml dosyasında tema olarak cihazın varsayılan temasını alması ve belirtilmedikçe appBar eklenmemesi için Theme.AppCompat.NoActionBar teması kullanıldı.
- Api erişimi ve local storage işlemleri için AndroidManifest.xml dosyasında cihaz izinleri eklendi:

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```
- build.gradle.kts dosyasına aşağıdaki pluginler eklendi:
 - **kotlin-kapt:** Cihazda SQLite üzerinde veri saklamak için kullanılan dao ve database sınıflarının otomatik üretilmesini sağlamak için eklendi.
 - **kotlin-parcelize:** Ana liste ekranından detay ekranına movie detayı iletmek için eklendi. Aktiviteler arası veri taşıma yöntemlerinden biridir.
- build.gradle.kts dosyasına aşağıdaki kütüphaneler (dependencies) eklendi:

```
//retrofit (HTTP istekleri atmak için eklendi)
implementation("com.squareup.retrofit2:retrofit:2.9.0")
implementation("com.squareup.retrofit2:converter-gson:2.9.0")

//kotlin coroutines (asen kron kod geliştirmek için eklendi)
implementation("org.jetbrains.kotlinx:kotlinx-coroutines-core:1.7.3")
implementation("org.jetbrains.kotlinx:kotlinx-coroutines-android:1.7.3")

//lifecycle viewmodel (MVVM yapısı kurmak için eklendi)
implementation("androidx.lifecycle:lifecycle-viewmodel-ktx:2.6.2")
implementation("androidx.lifecycle:lifecycle-livedata-ktx:2.6.2")

//room (Local storage SQLite işlemleri için eklendi)
implementation("androidx.room:room-runtime:2.6.1")
kapt("androidx.room:room-compiler:2.6.1")

//glide (ImageView kontrolüne URL ile resim bind etmek için eklendi)
implementation("com.github.bumptech.glide:glide:4.16.0")

//swiperefreshlayout (Ana ekrandaki listede yukarıda aşağı sürüklemeye yapınca refresh yapması için eklendi)
implementation("androidx.swiperefreshlayout:swiperefreshlayout:1.2.0-alpha01")
```
- mipmap klasöründeki ikon değiştirilerek uygulamaya default ikon harici yeni bir ikon belirlendi.
- Resource altına menu klasörü eklenerek ana ekran menüsü için xml hazırlandı, ikonları drawable resource altına eklendi:
 - **Ara:** Movie arama yapmak için searchview getirir.
 - **Sırala:** Sıralama yapması için kullanıcıya dialog açar.
 - **Listeleme türü:** Listeleme türü seçmesi için kullanıcıya dialog açar.
- Default gelen MainActivity.ktx silindi ve uygulamanın ilk açılışında gelmesi için SplashScreen boş bir aktivite olarak eklendi:
 - AndroidManifest.xml dosyasında oluşan SplashScreen tag içine ilk açılışta gelecek ekran olduğu belirtmek için MAIN ve LAUNCHER filtresi eklendi.
 - Tasarım arkaplanı geçişli bir görünüm olması için drawable resource altına splash_screen_gradient.xml eklendi.

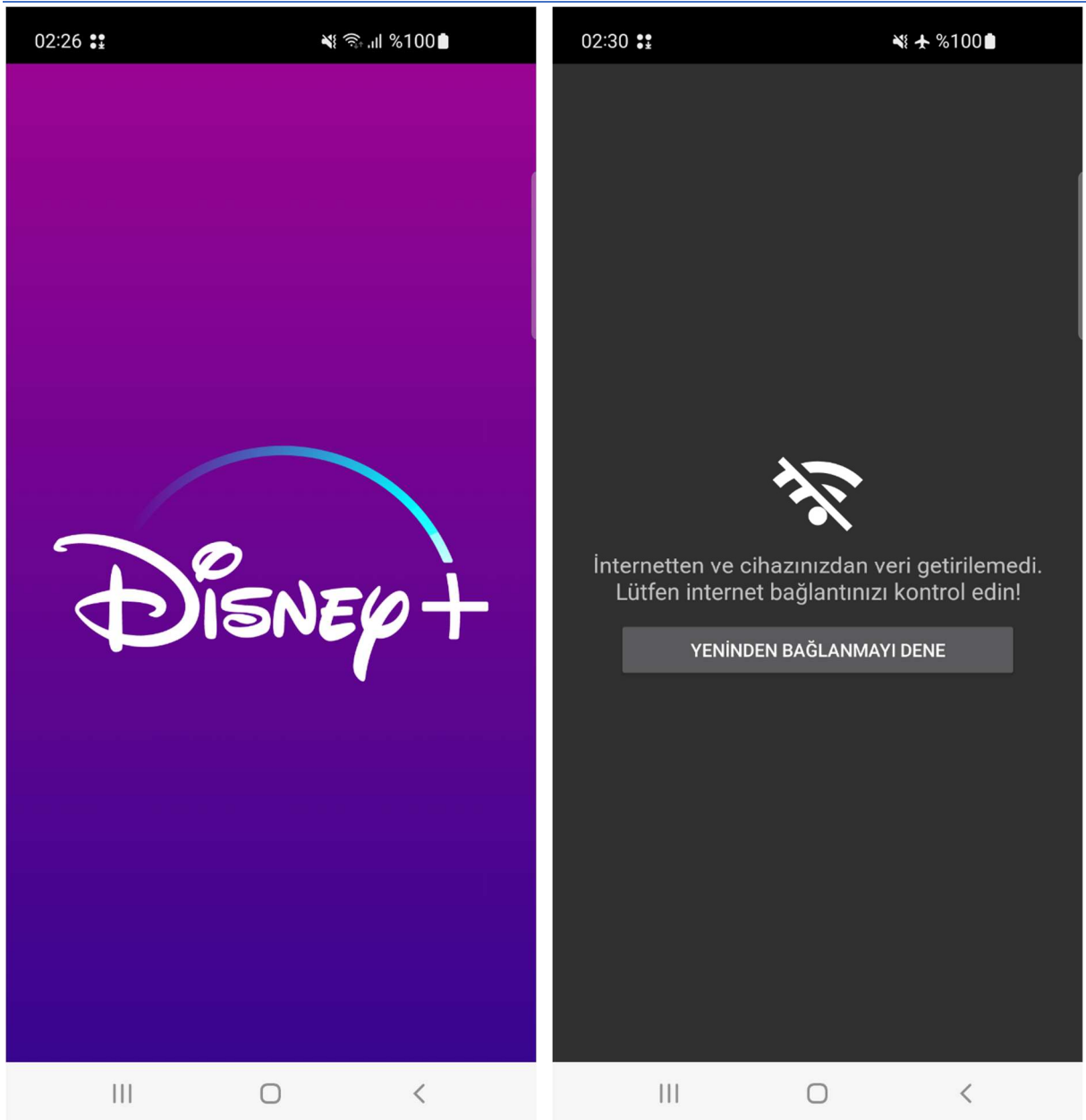
- Tasarımda ortalı olarak gelecek bir ImageView eklendi ve göstereceği resim drawable resource altına eklendi (disney_logo).
 - Kod sayfasına 3sn bekleyip ana ekranı (MainActivity) açması için Looper eklendi ve finish() komutu ile uygulamanın activity history'den silinmesi sağlandı.
 - Kod sayfasına geri tuşuna basılınca işlem yapılmaması için `onBackPressedDispatcher.addCallback { }` eklendi.
 - Ana ekran için MainActivity eklendi:
 - AndroidManifest.xml dosyasında oluşan MainActivity temasındaki NoActionBar ifadesi appbar gelmesi için kaldırıldı.
 - Ana ekran tasarlandı:
 - Yukarıdan aşağı sürükleme ile refreh yapmak için SwipeRefreshLayout eklendi.
 - Tüm kontrolleri relative olması için RelativeLayout kullanıldı.
 - Movie listesi için RecyclerView kullanıldı ve varsayılan olarak GridLayoutManager görünümünde spanCount=2 (2 sütun) olacak şekilde ayarlandı.
 - Lütfen bekleyiniz yazısı için TextView ve dönen animasyonu için ProgressBar eklendi.
 - Component adında bir klasör eklendi ve altına geri tuşuna basıldığında gösterilecek dialog kontrolü (CustomExitDialog) komponent olarak eklendi, böylece bu dialog ortak kullanıma sunulmuş oldu ve istenen ekranda burası çağrılarak dialog gösterilebilecek.
 - İnternet yok ise ve local storage henüz veri kaydedilmediyse uyarı ekranı gelmesi için EmptyDataActivity eklendi:
 - Ekrana ortalı olarak wifi_off ikonu eklendi
 - İnternet yok mesajı TextView ile eklendi.
 - İnterneti kontrol edip bağlantı var ise ana ekranı dönmesi için buton eklendi.
 - extensions klasörü eklendi ve altına NetworkExtensions sınıfı interneti kontrolü herkesin yapabilmesi için ortak bir yere eklendi.
 - Kod sayfasına NetworkExtensions sınıfındaki hasInternet çağrılarak bağlantı kontrolü yapıldı, bağlantı var ise ana sayfaya yönlendirme yapıldı.
 - Geri tuşuna basılırsa kullanıcıya ana ekrandaki gibi davranan dialog eklendi.
 - Disney apisinden verileri çekmek için;
 - model klasörü eklendi ve altına MovieModel sınıfı apiden gelen veri tiplerine göre hazırlandı; ayrıca detay ekranına bu sınıfı ileteceğimiz için Parcelable' dan türetili.
 - apis klasörü eklendi ve altına MovieApi sınıfı oluşturuldu, retrofit kütüphanesi bu api sınıfını kullanarak isteklerde bulunacak, bu yüzden URL' den listeyi getirmesi için bir tane GET metodu eklendi.
 - İnternette veriler çekildikten sonra local veri tabanına yazması için;
 - persistence klasörü eklendi ve altına;
 - MovieDao sınıfı eklendi, bu sınıf içerisinde yapılcak işlemler; select, insertAll olarak hazırlandı.
 - Local storage SQLite üzerinde oluşacak veri tabanı bilgilerini barındıran MovieDatabase sınıfı eklendi.
 - entities klasörü eklendi ve veri tabanında tutulacak MovieEntity tablosunu işaret eden sınıf oluşturuldu.
- Not:** MovieModel ve MovieEntity ayrı olmasının sebebi, veri tabanı sınıflarını tasarım sınıflarından yalıtmasıdır.
- exntesions klasörü altına MovieModel ve MovieEntity map etmesi için MapExtensions sınıfı eklendi.

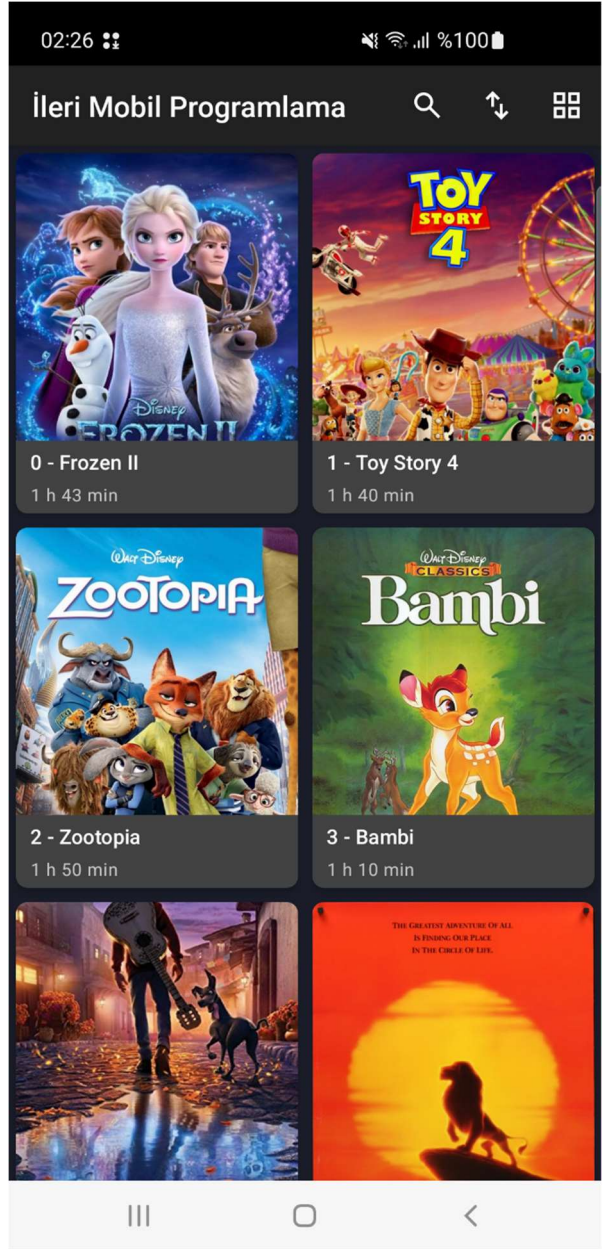
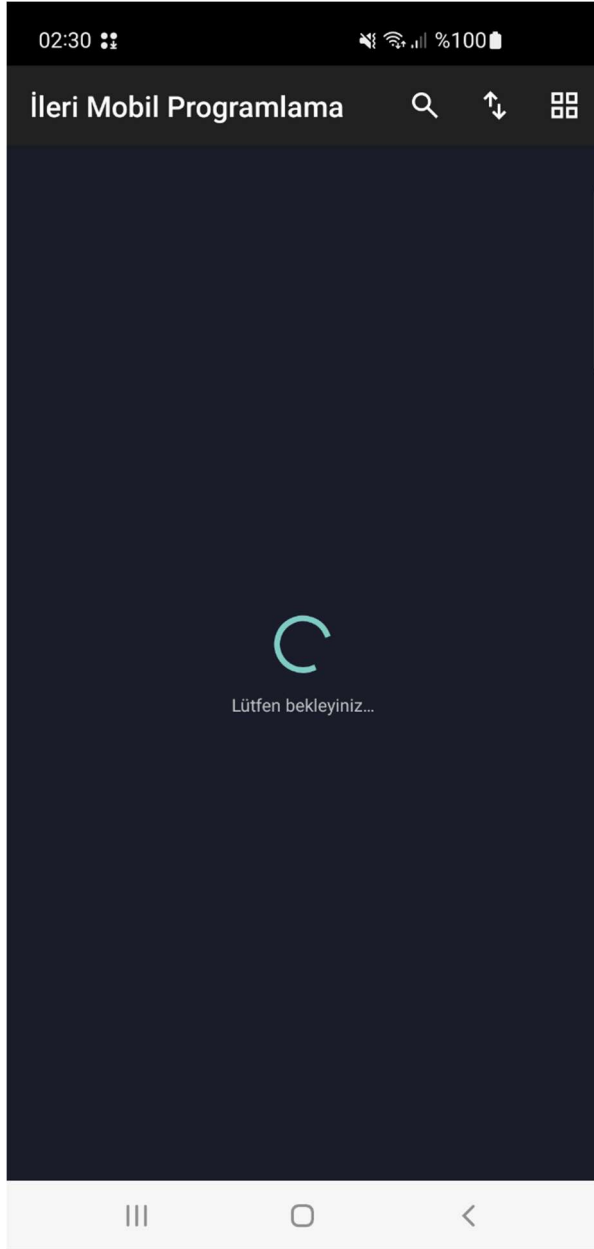
- MVVM ile ana ekran işlemlerini yapabilmek için;
 - view_model klasörü oluşturuldu ve altına MovieViewModel sınıfı eklendi.
 - ViewModel sınıfında ana ekrana verileri getirmesi için fetchMovies metodu eklendi:
 - İlk kural olarak internet kontrolü yapıldı, internet var ise apiden veriler çekilde ve local storage gelen veri kaydedildi.
 - Eğer internet yok ise local storage veriler getirildi.

Not: Burada internetten ve local storage' den veri olmama durumu kontrol edilmedi, onu ana ekranda burayı çektikten sonra kontrol ediyoruz ve duruma göre uyarı ekranına yönlendiriyoruz.
- Ana ekranda verileri listelemek, aramak ve sıralamak için;
 - adapters klasörü eklendi ve altına RecyclerView.Adapter' dan türeyen MovieListAdapter sınıfı oluşturuldu.
 - Adapter sınıfında constructor parametresi olarak movie listesi ve click eveni alınıyor, sınıf intı olunca movie listesini RecyclerView.Adapter' da kullanmak için local bir değişken set edildi.
 - Adapter altındaki ViewHolder nested sınıfında bindData metoduna gelen bir movie item için önce layout resource altına card_layout eklendi ve sonra burada verileri tasarıma bind edildi; resim URL ImageView bind etmek için Glide kütüphanesini kullanıldı. Card tıklandığında event tetiklenmesi için card listener set edildi.
 - Adapter sınıfına arama yapmak için getFilter metodu eklendi (*bunun için önce adapter sınıfı Filtirable sınıftan da türetildi*) ve burada gelen arama değerine göre filtreleme kodu yazıldı.
 - Adapter sınıfına sıralama yapmak için setSortType metodu eklendi ve bu metoda gelecek sortType enum parametresine göre sıralama yapıp adapter refresh etmesi sağlandı.
- Detay ekranı için MovieDetailActivity eklendi.
 - onCreate metodunda gelen movie verisine göre ekran bind edildi.
- Ana ekran kod sayfasında;
 - onCreate metodunda;
 - Geri yapıldığında kullanıcının çıkmak isteyip istemedini soracak dialog getirildi ve basılan butona göre (Evet/Hayır) uygulamanın kapanıp kapanmayacağı belirtildi (*onBackPressedDispatcher.addCallback { customExitDialog() }*)
 - ViewModelProview MovieViewModel olarak tanımlandı ve fetchMovie metodu çağrıldı:
 - Veriler yok ise (internet ve localden veri gelmedi ise) EmptyDataActivity açıldı,
 - Verileri var ise;
 - movieListAdapter tanımlandı ve tasarımdaki RecyclerView nesnesine adapter set edildi ve card tıklandığında detay ekranı açması sağlandı.
 - Lütfen bekleyiniz (loading) gizlendi.
 - RecyclerView gösterildi.
 - Yukarıdan aşağı sürükleyince refresh yapması için swipeRefresh.setOnRefreshListener eklendi.

- onCreateOptionsMenu metodu override edildi ve arama için searchView gelecek şekilde geliştirme yapıldı, ayrıca arama yaptıkça adapter sınıfındaki arama metodu ile listede arama yaptırıldı.
- onOptionsItemSelected metodu menuden seçilen öğeye göre işlem yapılması için override edildi:
 - Sıralama tıklandığında gelecek dialog için layout resource altına sort_type_dialog.xml eklendi ve dialogda seçilen tipe göre adapter sınıfındaki sıralama metodu ile sıralama yaptırıldı.
 - Liste tipi her tıklandığında ana ekrandaki görünümün 2 sütun ve 1 sütun arasında geçiş yaparak ekrandaki liste görünümünü ve listType ikonunu değiştirmesi sağlandı.

Uygulama Ekran Görüntüleri





02:29
100%

0 - Frozen II

(2019, 1 H 43 MIN)

Frozen II, also known as Frozen 2, is a 2019 American 3D computer-animated musical fantasy film produced by Walt Disney Animation Studios. The 58th animated film produced by the studio, it is the sequel to the 2013 film Frozen and features the return of directors Chris Buck and Jennifer Lee, producer Peter Del Vecho, songwriters Kristen Anderson-Lopez and Robert Lopez, and composer Christophe Beck. Lee also returns as screenwriter, penning the screenplay from a story by her, Buck, Marc E. Smith, Anderson-Lopez,

III
O
<

02:29
100%

0 - Frozen II

(2019, 1 H 43 MIN)

Frozen II, also known as Frozen 2, is a 2019 American 3D computer-animated musical fantasy film produced by Walt Disney Animation Studios. The 58th animated film produced by the studio, it is the sequel to the 2013 film Frozen and features the return of directors Chris Buck and Jennifer Lee, producer Peter Del Vecho, songwriters Kristen Anderson-Lopez and Robert Lopez, and composer Christophe Beck. Lee also returns as screenwriter, penning the screenplay from a story by her, Buck, Marc E. Smith, Anderson-Lopez, and Lopez,[2] while Byron Howard executive-produced the film.[a] [1] Veteran voice cast Kristen Bell, Idina Menzel, Josh Gad, Jonathan Groff, and Ciarán Hinds return as their previous characters, and they are joined by newcomers Sterling K. Brown, Evan Rachel Wood, Alfred Molina, Martha Plimpton, Jason Ritter, Rachel Matthews, and Jeremy Sisto.

Konusu:

King Agnarr of Arendelle tells a story to his young children, Elsa and Anna, that their grandfather, King Runeard, established a treaty with the neighboring tribe of Northuldra by building a dam in their homeland, the Enchanted Forest. However, a fight occurs, resulting in Runeard's death. The battle enrages the elemental spirits of Earth, Fire, Water, and Air of the forest. The spirits disappear and a wall of mist traps everyone in the Enchanted Forest. Young Agnarr barely escapes due to the help of an unknown savior.

III
O
<

