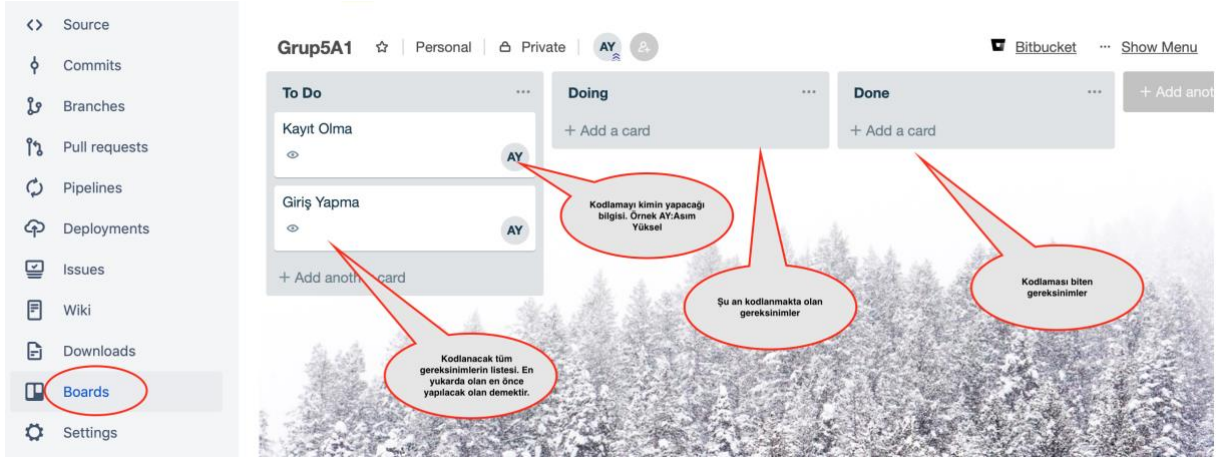
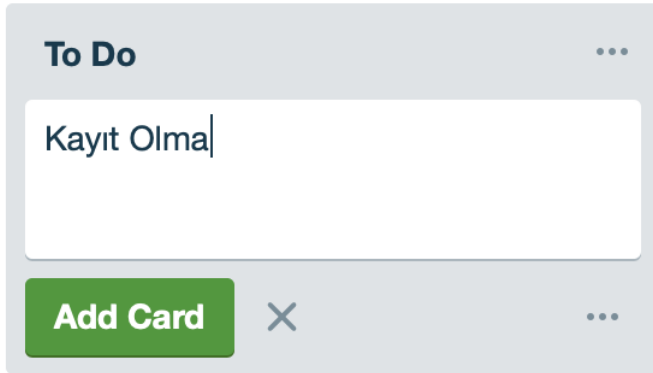


ÖDEV-4

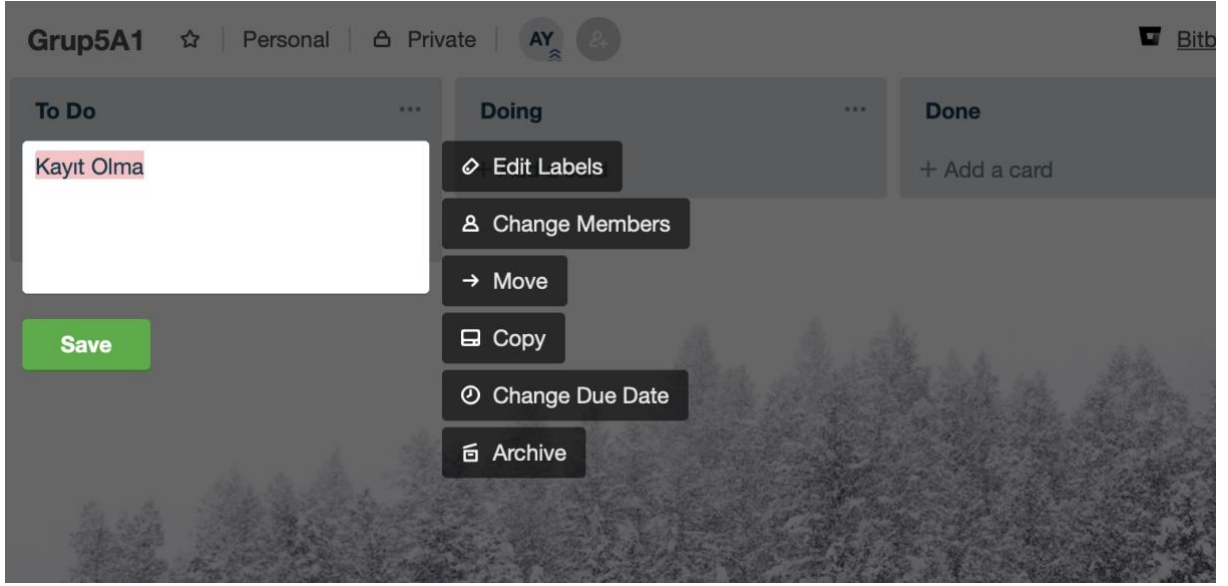
1) Artık kodlama kısmına geçmiş bulunmaktayız. Bu süreci iterasyonlar halinde yapacağız. Her aşamada çalışan bir kod göstereceksiniz. Proje minimum 2 iterasyonda bitmeli. Yani bana minimum 2 defa proje göstermelisiniz. İterasyon süresini kendiniz belirleyiniz. Tasarım kısmında önermiş olduğunuz her bir gereksinim bu iterasyonlarla koda dönüşecek. Gereksinimleri koda dönüştürürken önceliklendirme yapacaksınız. İlk olarak hangi gereksinimleri kodlayacağınızı belirleyeceksiniz. Hangi gereksinimlerin yapılacağını, şu an yapılıyor olduğunu ya da bittiğini Bitbucket Boards kısmından yapacaksınız. Aşağıda Boards kısmına ait bir görsel mevcut. Sizin adınıza bu kısmı aktifleştirdim.



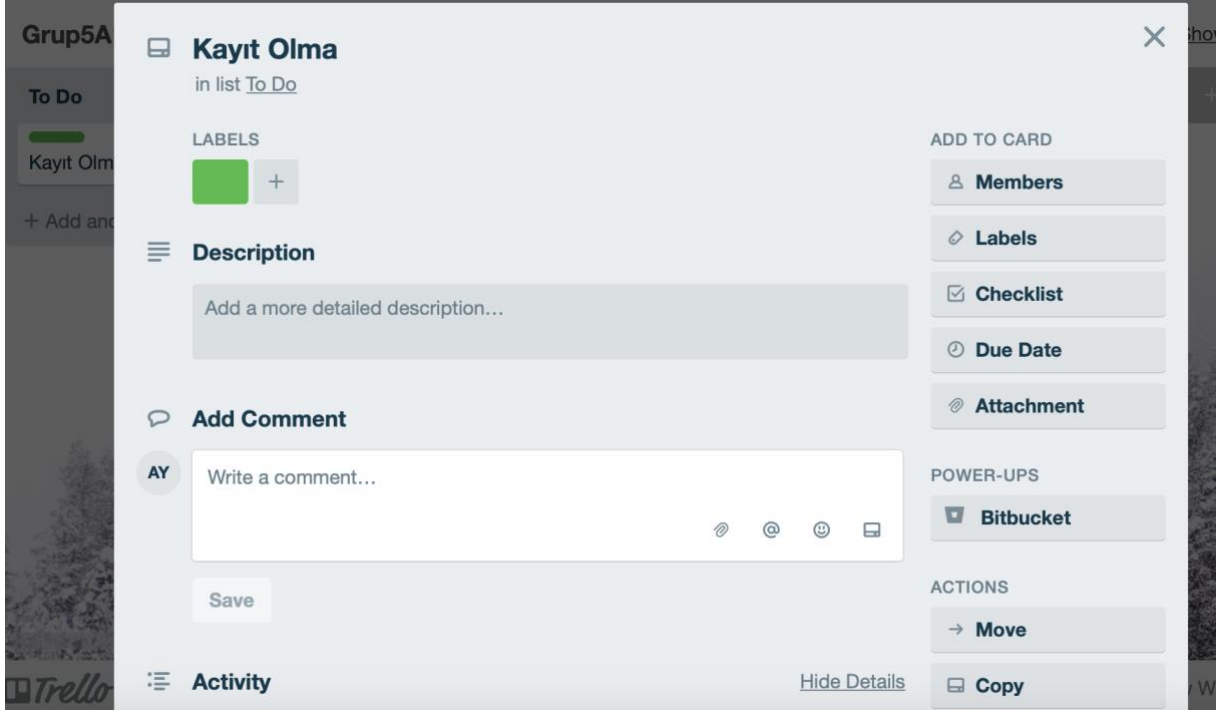
To Do Kısmı: Sisteminizde kodlanacak tüm gereksinim isimlerinin listesini içerir. Artı tuşuna (Add a Card) basarak gereksinim ekleyebilirsiniz ve gereksinimimize bir isim verebilirsiniz. Yeşil renkli Add Card tuşuna bastığınızda görev eklenir.



Görevi ekledikten sonra görev isminin en sağında bulunan kalem tuşuna bastığınızda aşağıdaki gibi hızlı işlemler menüsü karşınıza gelir. Edit labels ile renkli etiketler oluşturabilirsiniz. Change members diyerek bu görevi kimin yapacağını atayabilirsiniz. Move tuşu ile görevi diğer kısımlara aktarabilirsiniz. Copy ile görevin bir kopyasını oluşturabilirsiniz. Change Due Date diyerek görevin ne zaman tamamlanması gerektiğini ayarlayabilirsiniz. Archive tuşuna hiçbir zaman basmıyorsunuz.



Aynı zamanda gereksimin üzerine tıklayarak daha detaylı işlem yapabileceğiniz aşağıdaki pencere üzerinde de aynı işlemleri yapabilirsiniz.



Doing Kısmı: Kodlamaya başladığınız, üzerinde çalıştığınız gereksinimlerin listesini içerir. Bu kısımdaki görevler benim verdiğim tarihe kadar kodlaması bitecek olan gereksinimlerdir. Doing kısmını To Do kısmından Move tuşu ile taşıyarak dolduruyorsunuz. O hafta içinde kaç tanesini yapabileceğinize siz karar vereceksiniz.

Done Kısmı: Doing kısmında yapılacaklar yapıldığında buraya taşıyorsunuz. Bu taşıma işlemi kodları gösterim tarihinde bana gösterdikten sonra taşıyorsunuz. Gösterimde ben onaylamadan asla done kısmına taşıma yapmıyorsunuz. Gösterim bittikten sonra benim tamam olmuş dediğim kısımları buraya taşıyorsunuz. Dönem sonunda burada tüm gereksinimlerin ismi yer almış olacak.

2) Her iterasyon bittiğinde aşağıdaki gibi bir sayfa oluşturulacak. İsteyen video yerine ekran görselleri de koyabilir. Metin olarak yapılanları açıklayabilir.

Title * 1. İterasyon

Content * ? Edit

Bu İterasyonda Gerçekleştirilen Gereksinimler

- Kayıt Olma
- Giriş Yapma
- Blah Blah Blah

Bu İterasyonda Görev Alan Grup Üyeleri

- Ali
- Veli
- Selami

İterasyon Videosu

Bu kısma bu iterasyon sonucunda oluşan ürünle ilgili tanıtım videosu eklenecek. Bu aşamada neler yaptığınızı uygulamaya üzerinde göstererek anlatacaksınız. Grup üyelerinden sesi güzel olan birisi anlatım yapabilir :) Videoyu çektikten sonra Youtube'a atıp link verebilirsiniz.

3) İterasyonlarda isteyen önce Front-End'i, sonra API'yi, daha sonra Back-End'i yaparak ilerleyebilir (Örnek: İlk iterasyonda Front-End tasarlanır. 2.'de tüm API metotları yazılır ve POSTMAN ile Test Edilir. 3. İterasyonda Front-End API'ye bağlanır ve Backend halledilir.) İsteyen'de 3'ünü bir arada yapacak şekilde gidebilir (Örnek: 1. İterasyonda Kayıt ol sayfası tasarlanır. Kayıt ol API'si yazılır. Kayıt Ol sayfası API'ye bağlanarak Backend bitirilir. 2. İterasyonda yine aynı şekilde Giriş Yap Sayfası Diye gider).

4) Kullanıcı Dokümanı: Wiki'de kullanıcı dokümantasyonu isimli bir sayfa oluşturulmalı. Proje bittikten sonra kullanıcının sisteminizi, menüleri nasıl kullanacağına dair bilgileri, arayüz görsellerini içeren sayfadır. Yani kullanıcı için bir yardım dokümanıdır, kullanıcı kılavuzudur. Google da "kullanıcı dokümanı" ifadesiyle aratabilirsiniz.

Örnek:

- <https://v2.sambapos.org/tr/content/kullanici-dokumantasyonu>

5) Geliştirici Dokümanı: Sisteminizi geliştirmeye devam edecek, üzerine yeni özellikler ekleyecek geliştiriciler için neyin nasıl yapılacağını, hangi metotların ne iş yaptığını, API'lerin nasıl kullanılacağını tanımlayan doküman (Get Started Doc da denebilir).

Örnekler:

- <https://organikapi.com/documents/>
- <https://devcenter.heroku.com/articles/getting-started-with-java>
- <https://ituuucsd1605.readthedocs.io/en/latest/developer/mert.html#universite-sinifinin-yapisi-ve-kurucu-fonksiyonu>

Notlar:

- Final sınav tarihine kadar vaktiniz var. Projenizin son teslim tarihi dersin final günü saat 17:00'a kadardır.
- İterasyonlarınızı göstermek için önceden randevu alın.
- Kodlama aşamasında görev paylaşımına karışmayacağım. Tüm grup üyelerinin onaylayacağı, sonradan ben fazla yaptım sen az

yaptın, tüm projeyi ben yaptım demeyecek şekilde bir görev dağılımı yapınız.

- Kodlama bittiğinde tek parça bir ürün göstermeniz gerekiyor. Grup içi anlaşmazlıklardan dolayı herkes kendi yaptığı yeri getirip göstermemeli. Ayrı ayrı yapıyor olsanız bile son aşamada her şeyi birleştirip tek bir proje göstereceksiniz.
- Grubun ahengini bozan, grubun projeyi bitirmesini engelleyen, üzerine düşeni yapmayan grup üyelerini tespit edersem dersten bırakacağım.
- Angular vb. framework kullanma zorunluluğunu da kaldırıyorum. Yapanları ekstra puanla ödüllendireceğim.
- Proje başında belirlemiş olduğunuz tüm gereksinimleri kodlamış olmanız gerekiyor.
- Eğer Hocam biz REST API yazmak istemiyoruz direk veritabanına bağlayacağız dersiniz bu bölümü atlayabilirsiniz. Ancak 15 puan kesilir (85 üzerinden değerlendirilir). Ya da 2. Alternatif Mobil cihazların kullanacağı REST API ayrı yapılır. POSTMAN ile çalıştığını gösterirsiniz. Ana uygulamanız doğrudan veritabanına bağlanabilir. Bu durumda herhangi bir puan kırılmaz.
- Dönem başında söylediğim gibi 2FA, Facebook ve Google ile giriş işlemlerinin kodlanması zorunlu. Bunların 1 tanesinin bile eksik olması durumunda değerlendirme 85 üzerinden yapılacaktır.
- Görev dağılımında veritabanı ve tabloları oluşturma işleminin birlikte yapılması önerilir. Diğer kısımlar bireysel görevler şeklinde atanabilir. Front End'i bir gruba/kişiye Backend'i bir gruba/kişiye atayabilirsiniz.
- Kodlanmamış bir gereksinim varsa görevi kimin yapmadığına bakmaksızın tüm gruptan puan kırılacaktır. Dolayısıyla birbirinizi işi yapmaya teşvik edin, zorlayın.
- Mazereti olanlar dışında diğer grup üyeleri her iterasyon bitiminde odama gelerek **çalışır bir proje** göstermeli.

Değerlendirme:

- Her iterasyonda çalışan bir kod ortaya çıkmış mı?
- Boards'da Doing kısmındaki o iterasyon için kodlanması gereken gereksinimler tamamlanmış mı?
- Müşteri iterasyonu onayladığında Doing kısmındakiler Done kısmına geçmiş mi?

- Her bir iterasyon eşit puan değerindedir. Dolayısıyla 80 puanlık kod kısmı yaptığınız iterasyon sayısına eşit olarak bölünecektir. Proje minimum 2 iterasyonda bitmeli. Yani 2 iterasyondan önce proje bitti diye gelmiyorsunuz.
- Son ürün bitbucket'e push edilmiş mi?
- Son ürün yerelde ve bir alan adı altında çalışıyor mu?
- Tüm gereksinimler kodlanmış mı?
- Sunum kısmını odamda yapacağım. Projelerini bitirenler bir randevu tarihi almalılar ve grupça odama gelmelidir. Projenizi gösterdikten sonra hatalarınızı değiştirme imkânı verilmeyecektir ve notunuz değişmeyecektir. Ona göre eksiksiz ve projeniz hem lokalde hem de domain de çalışır şekilde gelin.