

Pemrograman Mobile Cross Platform

Modal, Loading, and Action Sheet Controller

OVERVIEW

Finishing this module, students will be able to:

- Able to create and utilize modal, loading, and action sheet controllers.
- Able to implement sliding options on ion-item components

This module is divided into several parts. OVERVIEW explains the overall objective of this module and how to understand its contents. MATERIALS NEEDED and SOFTWARE REQUIREMENTS shows the requirements. ACTIVITIES section explains how to do various things, as explained in class. Finally, the TASK describes the tasks that students need to finish and get graded in the corresponding module.

MATERIALS NEEDED / REFERENCES

- This module
- Materials from the class (PowerPoint slides, etc.) – *dari kelas teori*
- Ionic Framework Official Documentation

SOFTWARE REQUIREMENTS

- Visual Studio Code
- Android Studio or XCode
- Google Chrome

ACTIVITIES

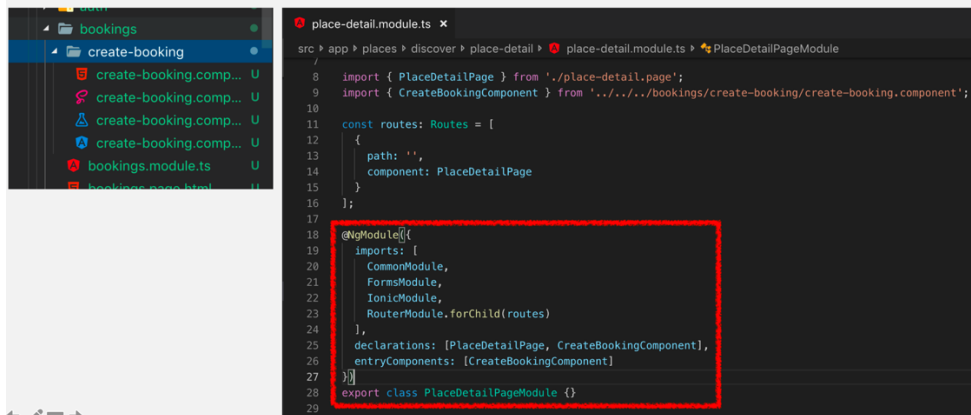
Modal Controller

Modal Controller

<https://ionicframework.com/docs/api/modal>

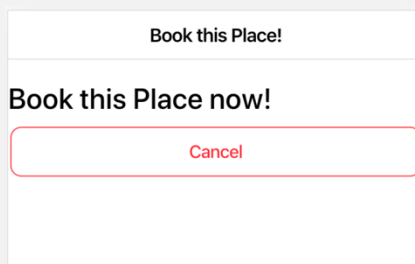
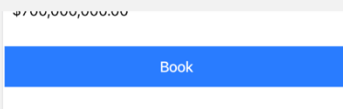
1. Create the component for the modal
>> `ionic generate component bookings/create-booking`
2. Declare the component in the `places/discover/place-detail/place-detail.module.ts` file on **declaration** and **entryComponents**
3. Create the button to show the modal
4. Present the modal using **ModalController**

1. Create the component for the modal
>> `ionic generate component bookings/create-booking`
2. Declare the component in the `places/discover/place-detail/place-detail.module.ts` file on **declaration** and **entryComponents**



3. Create the button to show the modal
4. Present the modal using **ModalController**

```
<ion-button expand="full" (click)="bookPlace()">Book</ion-button>
```



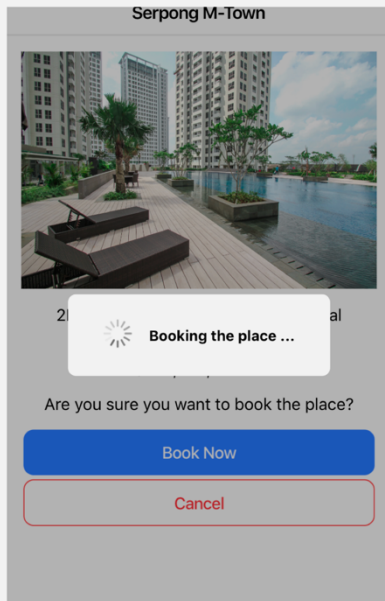
```
export class PlaceDetailPage implements OnInit {
  place: Place;
  constructor(
    private router: Router,
    private navCtrl: NavController,
    private route: ActivatedRoute,
    private placesSvc: PlacesService,
    private modalCtrl: ModalController) {}

  bookPlace() {
    this.modalCtrl.create({ component: CreateBookingComponent })
      .then(modalElement => {
        modalElement.present();
      });
  }
}
```

Loading Controller

Loading Controller

<https://ionicframework.com/docs/api/loading-controller>



```
constructor(private modalCtrl: ModalController,  
private loadingCtrl: LoadingController) { }
```

```
bookThisPlace() {  
  // this.isLoading = true;  
  this.loadingCtrl.create({  
    keyboardClose: true,  
    message: 'Booking the place ...'  
  })  
  .then(loadingEl => {  
    loadingEl.present();  
    setTimeout(() => {  
      loadingEl.dismiss();  
      this.modalCtrl.dismiss({ message: 'booked!' },  
        'confirm');  
    }, 2000);  
  });  
}
```

Action Sheet Controller



Action Sheet Controller

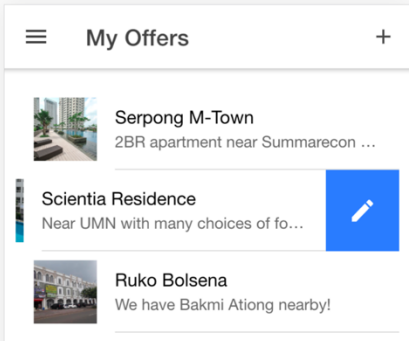
```
async bookPlace() {  
  const actionSheet = await this.actionSheetCtrl.create({  
    header: 'Book Place',  
    buttons: [{  
      text: 'Book w/ Random Date',  
      handler: () => {  
        this.modalCtrl.create({ component: CreateBookingComponent,  
          componentProps: { selectedPlace: this.place } })  
          .then(modalElement => {  
            modalElement.present();  
            return modalElement.onDidDismiss();  
          })  
          .then(resultData => {  
            console.log(resultData);  
          });  
      }  
    }, {  
      text: 'Cancel',  
      role: 'cancel',  
      handler: () => {  
        console.log('Cancel clicked');  
      }  
    }  
  ]  
});  
  await actionSheet.present();  
}
```

Sliding Items

Sliding Items

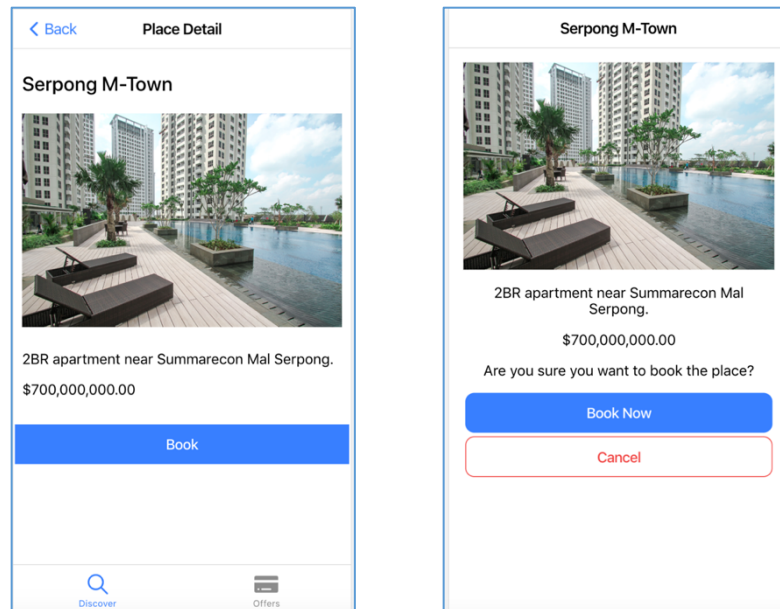
```
<ion-item-sliding *ngFor="let p of loadedPlaces" #slidingItem>
  <ion-item-options side="end">
    <ion-item-option (click)="editOffer(p.id, slidingItem)">
      <ion-icon name="create" slot="icon-only"></ion-icon>
    </ion-item-option>
  </ion-item-options>
  <ion-item [routerLink]="['/', 'places', 'tabs', 'offers', p.id]">
    <ion-thumbnail slot="start">
      <ion-img [src]="p.imageUrl"></ion-img>
    </ion-thumbnail>
    <ion-label>
      <h2>{{ p.title }}</h2>
      <p>{{ p.description }}</p>
    </ion-label>
  </ion-item>
</ion-item-sliding>
```

```
editOffer(id: string, slidingItem: IonItemSliding) {
  slidingItem.close();
  this.router.navigate(['/', 'places', 'tabs', 'offers', 'edit', id]);
}
```

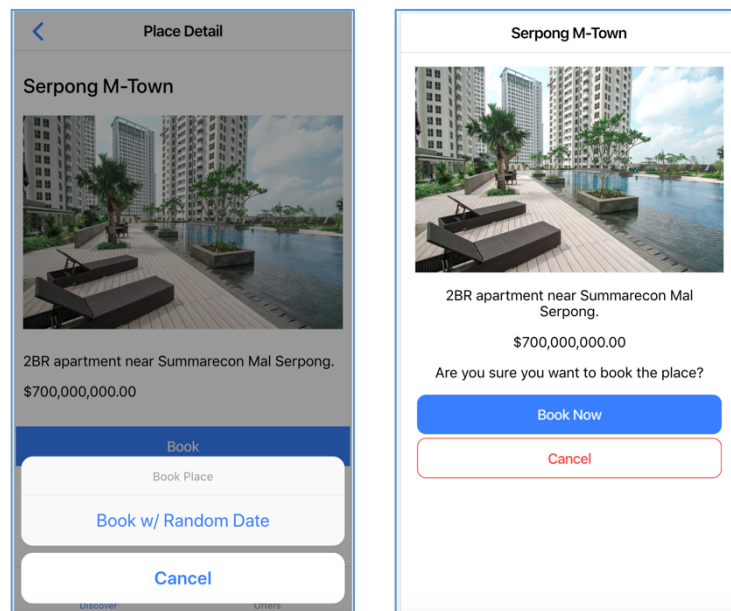


TASKS *(deadline today)*

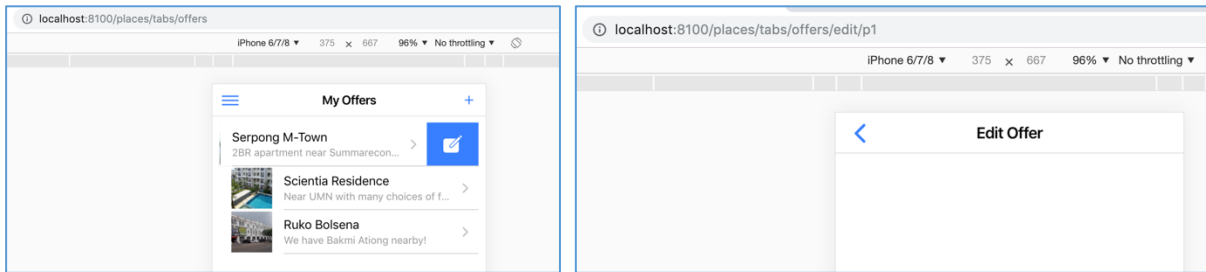
1. Continue working on the Ionic Places App from last week. When the user opens the Place Detail Page, provide a button to book the place which if clicked will open a Modal that enables the user to book the place.



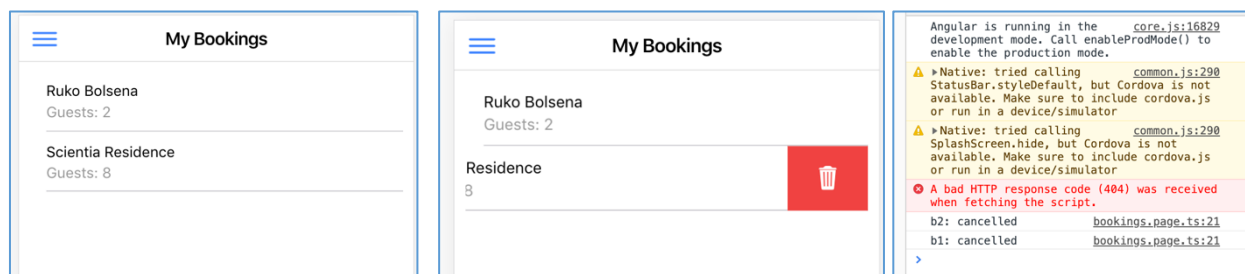
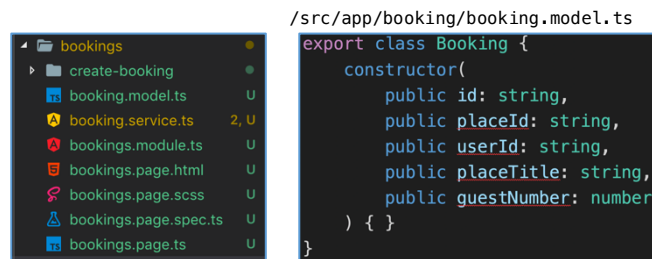
2. Create an action sheet which pops up when the Book button is clicked.



3. Create the swipable offers with sliding items, as shown below, which will redirect the page to the Edit Offer Page along with the ID passed as the parameter.



4. Create the booking page along with the Booking model and service



5. Create the delete functionality in the Booking Service so that the trash icon is clicked, the corresponding booking will be deleted.

