Contents

[**Report** 2](#_Toc88653997)

[**1, Chụp hình cái app khi hiện bảng tạo căn hộ thành công và code của cái chức năng bảng đó** 2](#_Toc88653998)

[**2, Là chụp hình cái bảng properties các mục điền để thông tin để tạo nhà và code cho cái bảng điền thông tin đó** 4](#_Toc88653999)

[**3, Là chụp hình cái nơi chứa db của mình (Trên web firebase) và code liên quan tới firebase** 6](#_Toc88654000)

[**A, Add:** 6](#_Toc88654001)

[**B, Get List** 7](#_Toc88654002)

[**C, Remove** 8](#_Toc88654003)

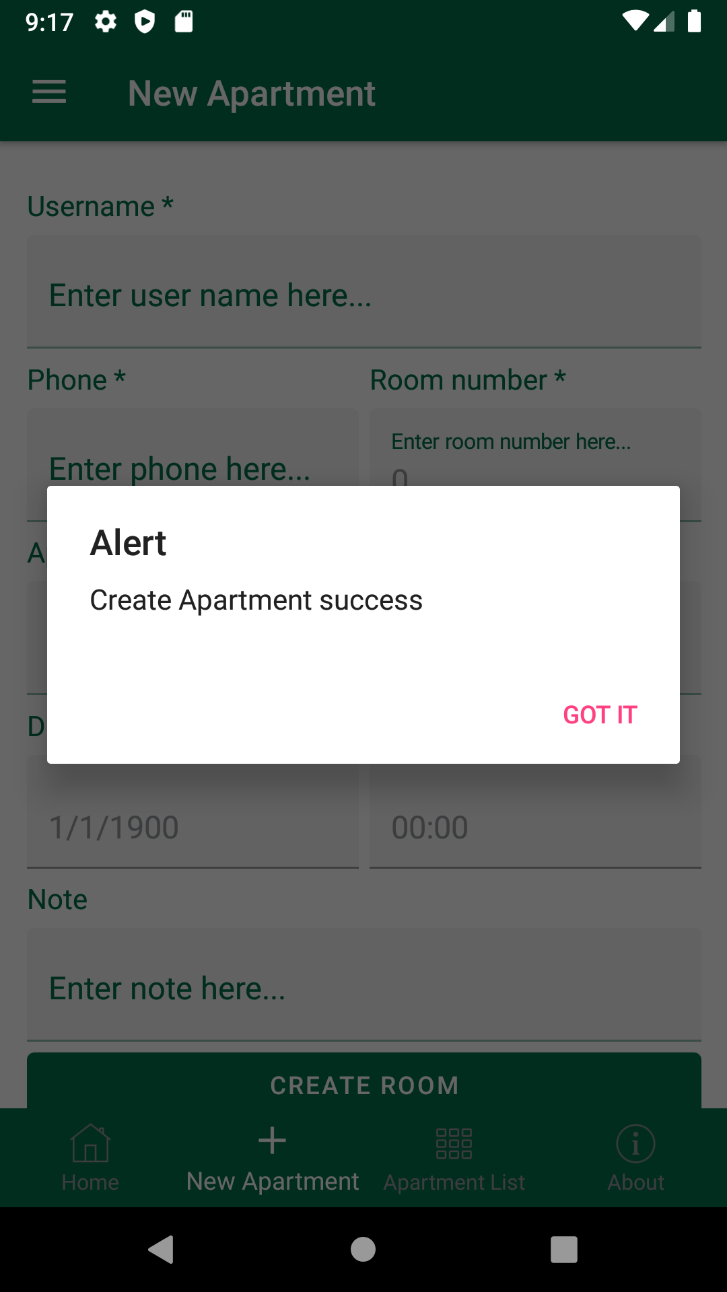
[**D, Filter** 8](#_Toc88654004)

[**E, Filter page** 9](#_Toc88654005)

[**4, Là chụp về việc tạo căn hộ mới mà bị chặn lại do validation và code liên quan tới việc này** 10](#_Toc88654006)

**Report**

**1, Chụp hình cái app khi hiện bảng tạo căn hộ thành công và code của cái chức năng bảng đó**



Code View:

<Button BackgroundColor="{StaticResource Green}" TextColor="White"

Text="Create Room" Command="{Binding CreateCommand}"/>

Code ViewModel:

public ICommand CreateCommand => new Command(async () =>

{

if (string.IsNullOrEmpty(Apartment.Username?.Trim())

|| string.IsNullOrEmpty(Apartment.Phone)

|| Apartment.Price == 0 || Apartment.RoomNumber == 0

|| Index == -1

|| Apartment.DateHire.Date == DateTime.MinValue.Date)

{

await Shell.Current.DisplayAlert("Alert", "Value can not be empty or null", "Got it");

}

else

{

try

{

Apartment.City = (City)Index;

var itemExist = (await FirebaseDatabase.Child("Apartments")

.OnceAsync<Item>()).FirstOrDefault(x => x.Object.RoomNumber == Apartment.RoomNumber

&& x.Object.City == Apartment.City);

if (itemExist != null)

{

await Shell.Current.DisplayAlert("Alert", "Room is booked", "Got it");

}

else

{

Apartment.Id = Guid.NewGuid().ToString();

await FirebaseDatabase.Child("Apartments")

.PostAsync(Apartment);

Index = -1;

Apartment = new Item();

await Shell.Current.DisplayAlert("Alert", "Create Apartment success", "Got it");

}

}

catch (Exception ex)

{

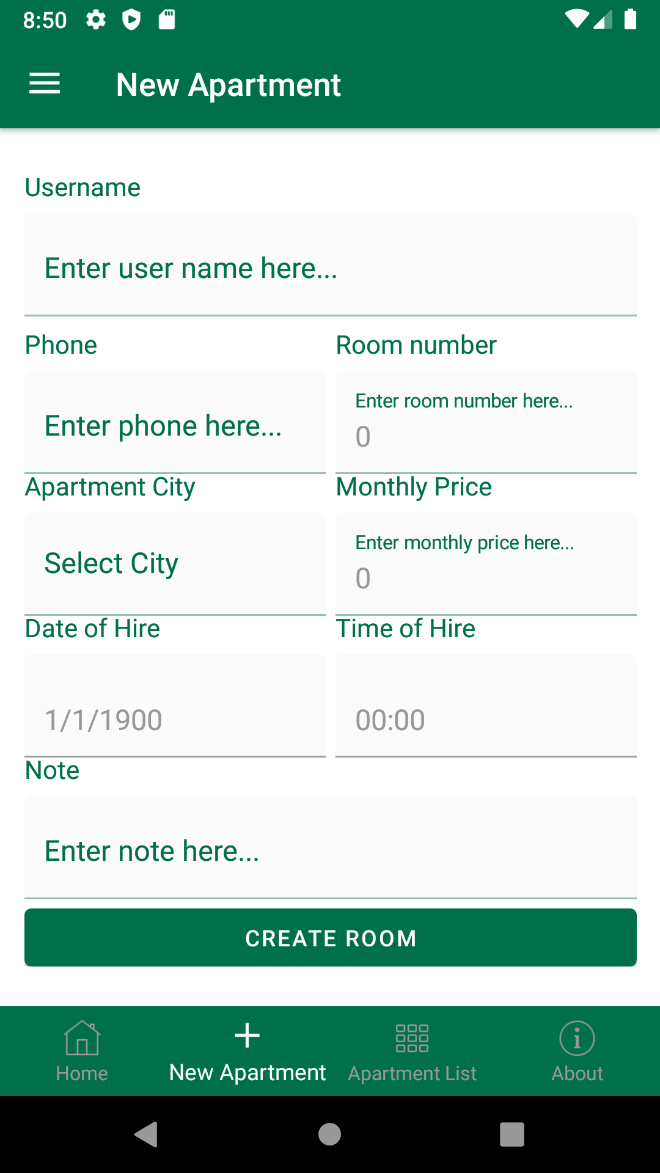
Debug.WriteLine(ex.Message);

}

}

});

**2, Là chụp hình cái bảng properties các mục điền để thông tin để tạo nhà và code cho cái bảng điền thông tin đó**



Code View:

<ScrollView>

<StackLayout Margin="15,25">

<StackLayout>

<Label Text="Username \*" FontSize="Subtitle" TextColor="{StaticResource Green}"/>

<Editor Placeholder="Enter user name here..." PlaceholderColor="{StaticResource Green}"

TextColor="{StaticResource LightGray}" Keyboard="Text" Text="{Binding Apartment.Username}"/>

</StackLayout>

<Grid ColumnDefinitions="\*,\*">

<StackLayout>

<Label Text="Phone \*" FontSize="Subtitle" TextColor="{StaticResource Green}"/>

<Editor Placeholder="Enter phone here..." PlaceholderColor="{StaticResource Green}"

TextColor="{StaticResource LightGray}" Keyboard="Telephone" Text="{Binding Apartment.Phone}"/>

</StackLayout>

<StackLayout Grid.Column="1">

<Label Text="Room number \*" FontSize="Subtitle" TextColor="{StaticResource Green}"/>

<Editor Placeholder="Enter room number here..." PlaceholderColor="{StaticResource Green}"

TextColor="{StaticResource LightGray}" Keyboard="Numeric" MaxLength="3" Text="{Binding Apartment.RoomNumber}"/>

</StackLayout>

</Grid>

<Grid ColumnDefinitions="\*,\*">

<StackLayout>

<Label Text="Apartment City \*" FontSize="Subtitle" TextColor="{StaticResource Green}"/>

<Picker ItemsSource="{Binding Cities}" Title="Select City" TextColor="{StaticResource LightGray}" TitleColor="{StaticResource Green}"

FontSize="18" SelectedIndex="{Binding Index}"/>

</StackLayout>

<StackLayout Grid.Column="1">

<Label Text="Monthly Price \*" FontSize="Subtitle" TextColor="{StaticResource Green}"/>

<Editor Placeholder="Enter monthly price here..." PlaceholderColor="{StaticResource Green}"

TextColor="{StaticResource LightGray}" Keyboard="Numeric" Text="{Binding Apartment.Price}"/>

</StackLayout>

</Grid>

<Grid ColumnDefinitions="\*,\*">

<StackLayout>

<Label Text="Date of Hire \*" FontSize="Subtitle" TextColor="{StaticResource Green}"/>

<DatePicker TextColor="{StaticResource LightGray}" Date="{Binding Apartment.DateHire}"/>

</StackLayout>

<StackLayout Grid.Column="1">

<Label Text="Time of Hire" FontSize="Subtitle" TextColor="{StaticResource Green}"/>

<TimePicker TextColor="{StaticResource LightGray}" Format="HH:mm" Time="{Binding Apartment.TimeHire}"/>

</StackLayout>

</Grid>

<StackLayout>

<Label Text="Note" FontSize="Subtitle" TextColor="{StaticResource Green}"/>

<Editor Placeholder="Enter note here..." PlaceholderColor="{StaticResource Green}"

TextColor="{StaticResource LightGray}" Text="{Binding Apartment.Note}" Keyboard="Text"/>

</StackLayout>

<Button BackgroundColor="{StaticResource Green}" TextColor="White"

Text="Create Room" Command="{Binding CreateCommand}"/>

</StackLayout>

</ScrollView>

**3, Là chụp hình cái nơi chứa db của mình (Trên web firebase) và code liên quan tới firebase**



Code ViewModel:

**A, Add:**

try

{

Apartment.City = (City)Index;

var itemExist = (await FirebaseDatabase.Child("Apartments")

.OnceAsync<Item>()).FirstOrDefault(x => x.Object.RoomNumber == Apartment.RoomNumber

&& x.Object.City == Apartment.City);

if (itemExist != null)

{

await Shell.Current.DisplayAlert("Alert", "Room is booked", "Got it");

}

else

{

Apartment.Id = Guid.NewGuid().ToString();

await FirebaseDatabase.Child("Apartments")

.PostAsync(Apartment);

Index = -1;

Apartment = new Item();

await Shell.Current.DisplayAlert("Alert", "Create Apartment success", "Got it");

}

}

catch (Exception ex)

{

Debug.WriteLine(ex.Message);

}

**B, Get List**

async Task ExecuteLoadClassroomsCommand()

{

IsBusy = true;

try

{

Apartments.Clear();

var list = (await FirebaseDatabase.Child("Apartments")

.OnceAsync<Item>()).Select(x => new Item

{

Username = x.Object.Username,

City = x.Object.City,

DateHire = x.Object.DateHire,

Note = x.Object.Note,

Phone = x.Object.Phone,

Price = x.Object.Price,

RoomNumber = x.Object.RoomNumber,

TimeHire = x.Object.TimeHire,

});

if(Apartments.Count == 0)

{

foreach (var item in list)

{

Apartments.Add(item);

}

}

ExecuteFilterItems();

}

catch (Exception ex)

{

Debug.WriteLine(ex);

}

finally

{

IsBusy = false;

}

}

**C, Remove**

try

{

var item = (await FirebaseDatabase.Child("Apartments")

.OnceAsync<Item>()).Where(x => x.Object.Id == aparment.Id).FirstOrDefault();

await FirebaseDatabase.Child("Apartments").Child(item?.Key).DeleteAsync();

}

catch (Exception ex)

{

Debug.WriteLine(ex.Message);

}

**D, Filter**

Task TextChanged(string username)

{

// username

var un = username?.Trim().ToLower();

if (!string.IsNullOrEmpty(un))

{

var list = Apartments.ToList();

list = list.Where(x => x.Username.Trim().ToLower().Contains(un)).ToList();

Apartments.Clear();

foreach (var item in list)

{

Apartments.Add(item);

}

}

return Task.CompletedTask;

}

**E, Filter page**

void ExecuteFilterItems()

{

if (FilterApartment == null) return;

var list = Apartments.ToList();

// username

var un = FilterApartment.Username?.Trim().ToLower();

if (!string.IsNullOrEmpty(un))

{

list = list.Where(x => x.Username.Trim().ToLower().Contains(un)).ToList();

}

// phone

var phone = FilterApartment.Phone?.Trim();

if (!string.IsNullOrEmpty(phone))

{

list = list.Where(x => x.Phone == phone).ToList();

}

// roomnumber

var room = FilterApartment.RoomNumber;

if (room != 0)

{

list = list.Where(x => x.RoomNumber == room).ToList();

}

// city

var city = FilterApartment.City;

if (city != null)

{

list = list.Where(x => x.City == city).ToList();

}

Apartments.Clear();

foreach (var item in list)

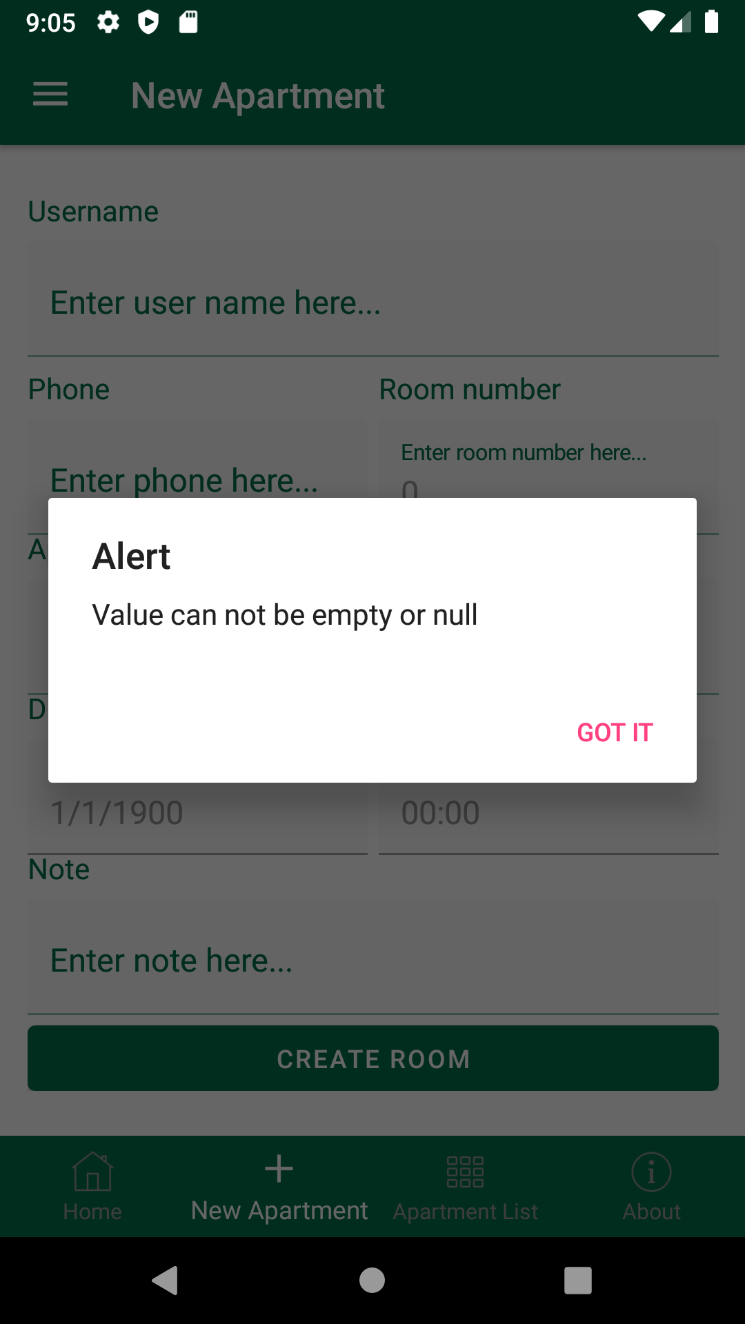
{

Apartments.Add(item);

}

}

**4, Là chụp về việc tạo căn hộ mới mà bị chặn lại do validation và code liên quan tới việc này**



Code ViewModel:

if (string.IsNullOrEmpty(Apartment.Username?.Trim())

|| string.IsNullOrEmpty(Apartment.Phone)

|| Apartment.Price == 0 || Apartment.RoomNumber == 0

|| Index == -1

|| Apartment.DateHire.Date == DateTime.MinValue.Date)

{

await Shell.Current.DisplayAlert("Alert", "Value can not be empty or null", "Got it");

}