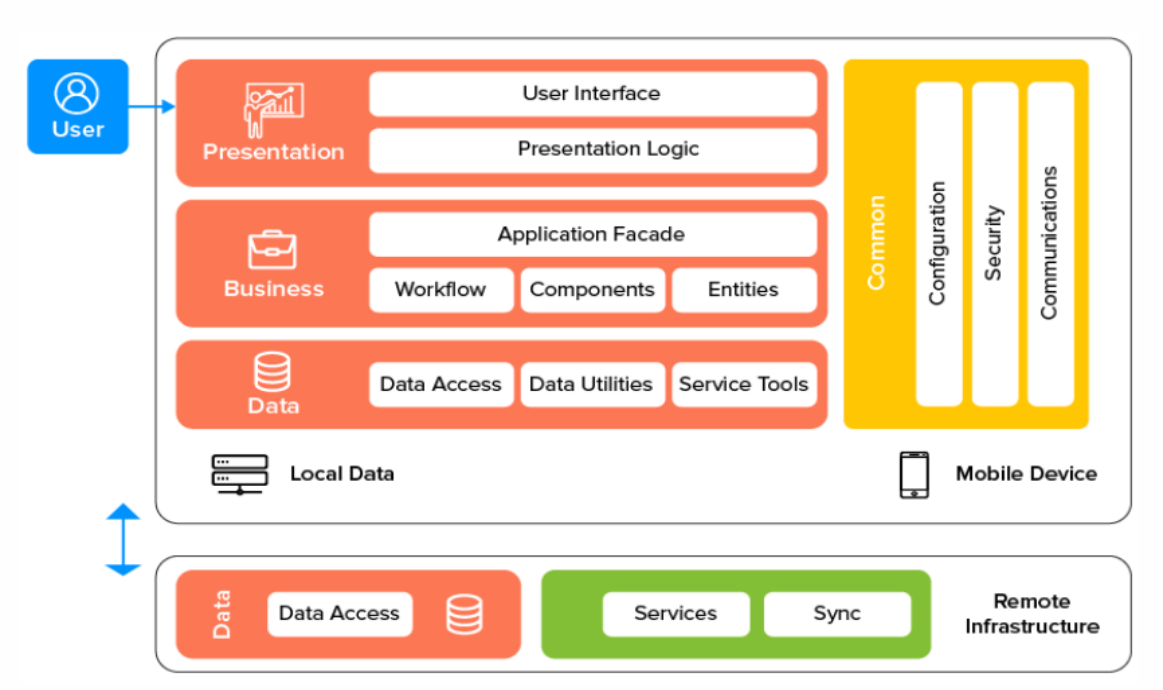
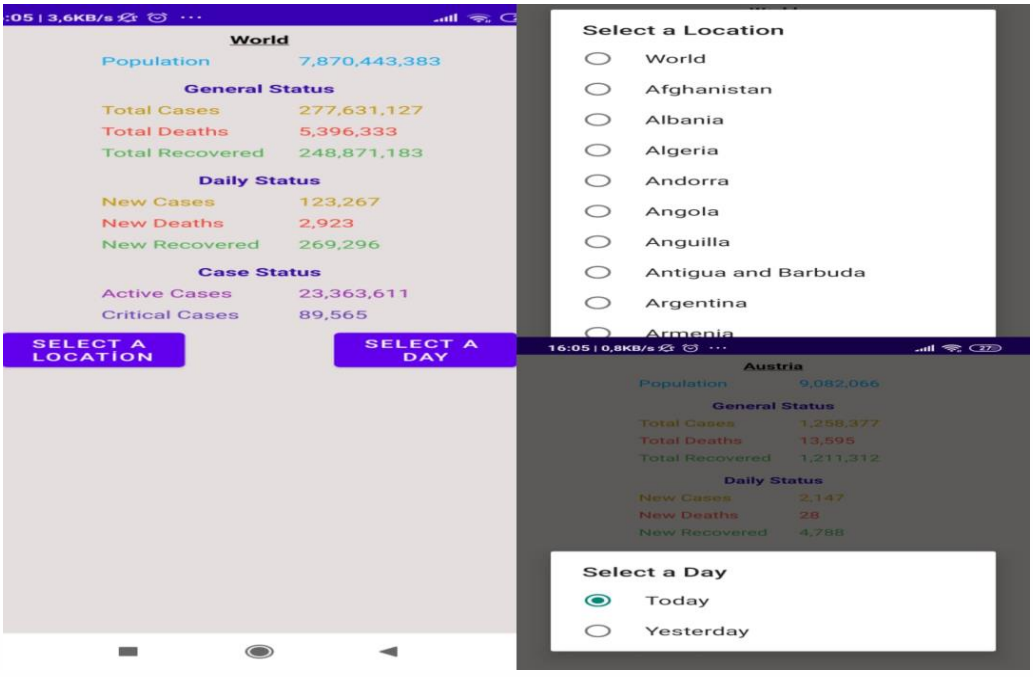
UYGULAMA MİMARİ TASARIMI

Bir uygulama mimarisi , bir işte kullanılan uygulamaların birbirleriyle ve kullanıcılarla nasıl etkileşimde bulunduklarına odaklanan davranışlarını tanımlar.



Sunum Katmanı



İş Katmanı

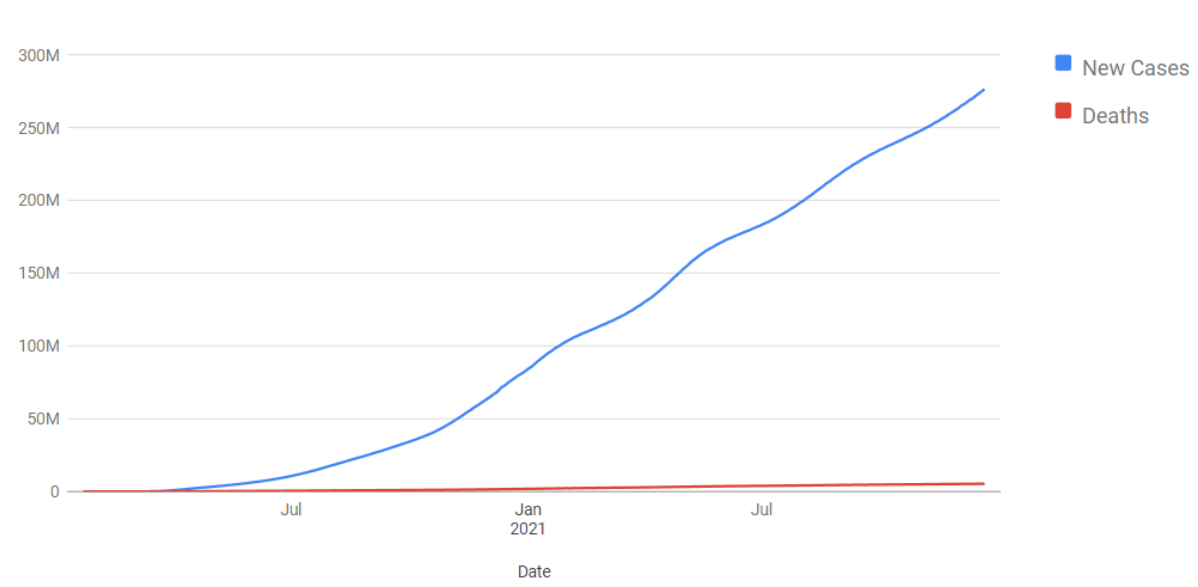
```
btnLocation.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View arg0) {
        if (countryNamesList.size() == 0) {
            Country.initializeCountryNames(ctx, countryNamesList, new VolleyCallBack() {
                @Override
                public void onSuccess() {
                    countryNamesSequence = countryNamesList.toArray(new CharSequence[countryNamesList.size()]);
                    AlertDialog.Builder ad = new AlertDialog.Builder( context: MainActivity.this);
                    ad.setCancelable(false);
                    ad.setTitle("Select a Location");
                    ad.setSingleChoiceItems(countryNamesSequence, checkedCountriesItem, new DialogInterface.OnClickListener()
                        @Override
                        public void onClick(DialogInterface dialogInterface, int arg1) {
                            checkedCountriesItem = arg1;
                            country = new Country(countryNamesSequence[arg1].toString());
                            boolean yesterday = false;
                            if(checkedDaysItem == 1)
                                yesterday = true;
                            country.initializeCovidData(ctx,yesterday, new VolleyCallBack() {
                                @Override
                                public void onSuccess() { setTexts(country); }
                            });
                            dialogInterface.cancel();
                        }
                    });
                    ad.show();
                }
            });
        }
        else
        {
            AlertDialog.Builder ad = new AlertDialog.Builder( context: MainActivity.this);
            ad.setCancelable(false);
            ad.setTitle("Select a Location");
            ad.setSingleChoiceItems(countryNamesSequence, checkedCountriesItem, new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialogInterface, int arg1) {
                    checkedCountriesItem = arg1;
                    country = new Country(countryNamesSequence[arg1].toString());
                    boolean yesterday = false;
                    if(checkedDaysItem == 1)
```

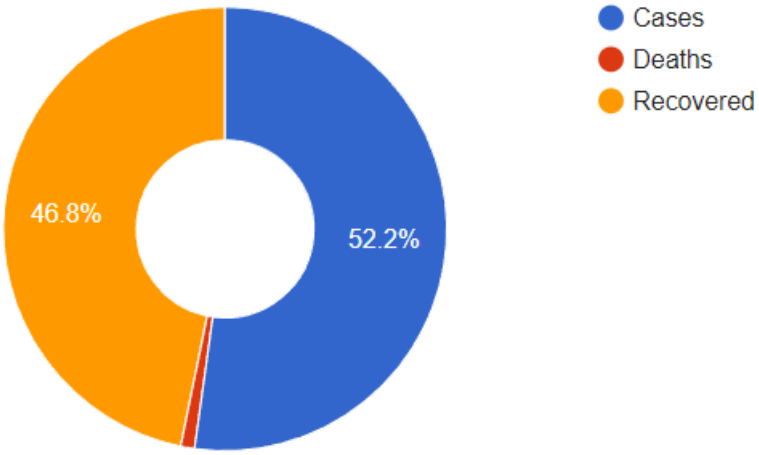
Veri Katmanı

```
void initializeCovidData(Context context,boolean yesterday, final VolleyCallBack callBack)
{
    if(!InternetConnection.checkConnection(context)) {
        Toast.makeText(context, text: "Connection Error", Toast.LENGTH_LONG).show();
        return;
    }

    String url;
    if(name.equals("World"))
        url = "https://disease.sh/v3/covid-19/all?yesterday=" + yesterday;
    else
        url = "https://disease.sh/v3/covid-19/countries/" + name + "?yesterday=" + yesterday;
    JsonObjectRequest request = new JsonObjectRequest(Request.Method.GET, url, jsonRequest: null,
            new Response.Listener<JSONObject>() {
                @Override
                public void onResponse(JSONObject response) {
                    try {
                        population = response.getLong( name: "population");
                        covidData.totalCases = response.getInt( name: "cases");
                        covidData.todayCases = response.getInt( name: "todayCases");
                        covidData.totalDeaths = response.getInt( name: "deaths");
                        covidData.todayDeaths = response.getInt( name: "todayDeaths");
                        covidData.totalRecovered = response.getInt( name: "recovered");
                        covidData.todayRecovered = response.getInt( name: "todayRecovered");
                        covidData.activeCases = response.getInt( name: "active");
                        covidData.criticalCases = response.getInt( name: "critical");

                        callBack.onSuccess();
                    } catch (JSONException e) {
                        Toast.makeText(context, text: "Reading Data Error", Toast.LENGTH_LONG).show();
                    }
                }
            }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            try{
                Toast.makeText(context, text: "Error Code : " + error.networkResponse.statusCode, Toast.LENGTH_LONG).show();
            }catch (Exception e){
                Toast.makeText(context, text: "Connection Error", Toast.LENGTH_LONG).show();
```


300M
250M
200M
150M
100M
50M
0
Jul
Jan
2021
Jul
Date
New Cases
Deaths


46.8%
52.2%
Cases
Deaths
Recovered