

NOTE ON WATER BALANCE APP

Prepared By:- Ashish Wankhade, Parth Gupta, Swapnil Patil, Vidyadhar Konde,
CSE, IIT Bombay
Date:30 October 2019

This document describes the Water Balance App and its component in short. Section I discuss water balance and its component while section II gives an illustration of Water Balance App. Complete water balance model is explained in POCRA Report on description of water balance framework [1].Link for downloading the app explained is at referece [2].

Introduction:-

The app computes point level water balance based on SWAT model. Application takes minimal inputs like Farm location, cropping pattern, irrigation is done (if any) and delivers results about the farm status in the form of values like Water Deficit, Rainfall, Runoff, AET, Soil Moisture, etc. The output results are given for both Crop and Monsoon End. It is likely to be beneficial for

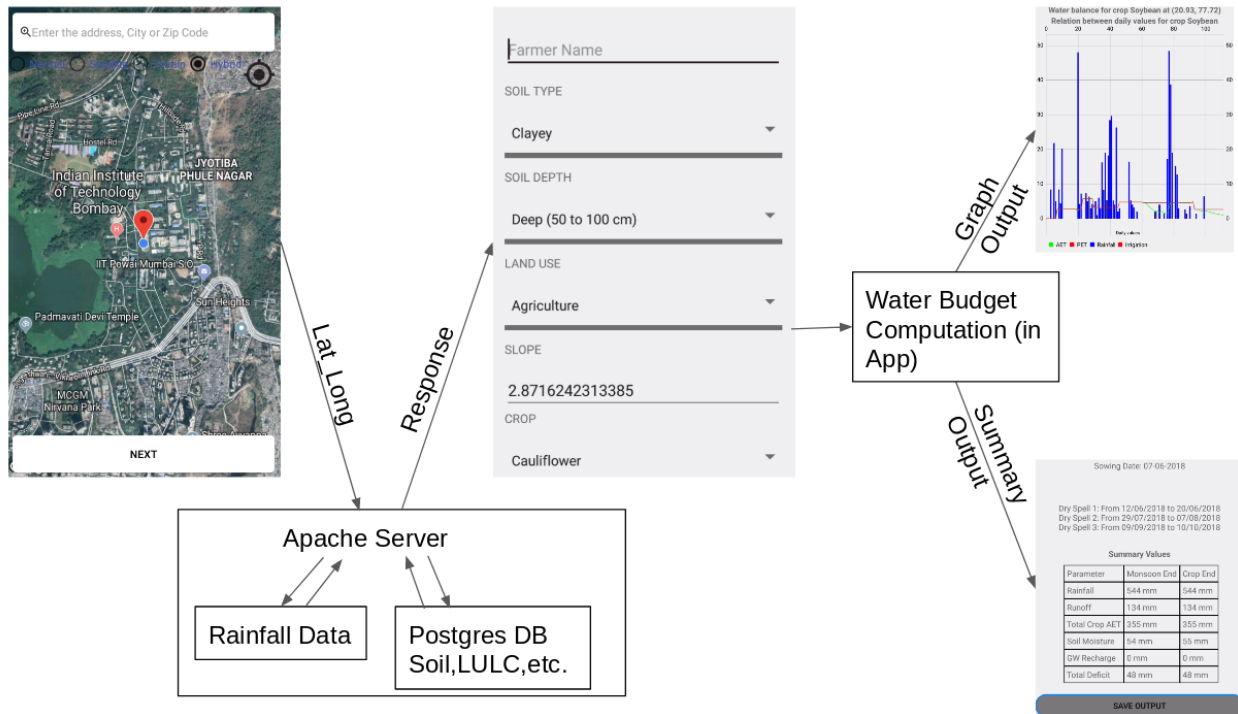


Figure 1: Context Flow Diagram

Krishi-Sahayak and to the administrators for village level planning.

Figure 1 represents the flow in which the outputs are generated based on given agriculture-related inputs. Initially, farm location is taken as input from the user (can be Krishi-Mitra, administrator, etc) with Google map/satellite as a reference. This location in terms of latitude and longitude is fed as input to the server and respective values for that corresponding farm are fetched from the server. Those values comprise of Soil Type, Soil Depth, Land Use, and Slope.

Then additional information like farmer name, Crop, Irrigation count and amount of irrigation is taken as input from the user. This set of input is then fed to the water budget model calculator in the android application and respective graphs and summary values are computed for both monsoon and crop end. The output value primarily comprises of AET, Water Deficit, Rainfall, Runoff, Soil Moisture, and Ground Water Recharge.

SECTION I

Water Budget Concepts:-

The hydrological cycle forms the basis of the water budget. Its key components include precipitation, surface runoff, stored infiltration, storage surface water, ground and water discharge, evapotranspiration.

Evapotranspiration(ET)

The combination of two separate processes whereby water is lost on the one hand from the soil surface by evaporation and on the other hand from the crop by transpiration is referred to as evapotranspiration (ET). Evaporation and transpiration occur simultaneously and there is no easy way of distinguishing between the two processes.

Reference evapotranspiration (ET_o)

The evapotranspiration rate from a reference surface, without short of water, is called the reference crop evapotranspiration or reference evapotranspiration and is denoted as ET_o. The reference surface is a grass reference crop with specific characteristics. FAO Penman-Monteith method is recommended as the sole standard method for the definition and computation of the reference evapotranspiration.

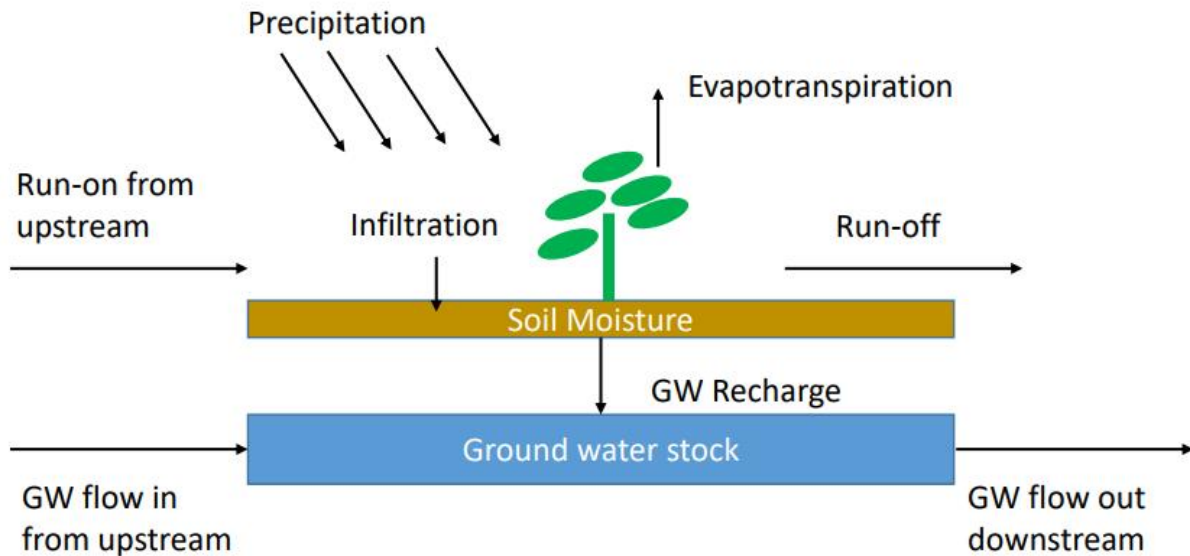


Figure 2: Water balance components

Crop evapotranspiration (ET_c)

The crop evapotranspiration differs distinctly from the reference evapotranspiration (ET_o) as the ground cover, canopy properties and aerodynamic resistance of the crop are different from grass. The effects of characteristics that distinguish field crops from grass are integrated into the crop coefficient (K_c). In the crop coefficient approach, crop evapotranspiration is calculated by multiplying ET_o by K_c. So,

$$ET_c = K_c * ET_o$$

Single crop coefficient approach (K_c)

The K_c predicts ET_c under standard conditions. This represents the upper envelope of crop evapotranspiration and represents conditions where no limitations are placed on crop growth or evapotranspiration due to water shortage, crop density, or disease, weed, insect or salinity pressures. The ET_c predicted by K_c is adjusted if necessary to non-standard conditions, ET_c adj, where any environmental condition or characteristic is known to have an impact on or to limit ET_c.

Saturation

During a rain shower or irrigation application, the soil pores will fill with water. If all soil pores are filled with water the soil is said to be saturated. There is no air left in the soil. It is easy to determine in the field if the soil is saturated. After the rain or the irrigation has stopped, part of the water present in the larger pores will move downward. This process is called drainage or percolation.

Field capacity

Field Capacity is the amount of soil moisture or water content held in the soil after excess water has drained away and the rate of downward movement has decreased. This usually takes place 2-3 days after rain or irrigation in previous soils of uniform structure and texture. After the drainage has stopped, the large soil pores are filled with both air and water while the smaller pores are still full of water. At this stage, the soil is said to be at field capacity. At field capacity, the water and air contents of the soil are considered to be ideal for crop growth.

Permanent wilting point

The water stored in the soil is taken up by the plant roots or evaporated from the topsoil into the atmosphere. If no additional water is supplied to the soil, it gradually dries out. The dryer the soil becomes, the more tightly the remaining water is retained and the more difficult it is for the plant roots to extract it. At a certain stage, the uptake of water is not sufficient to meet the plant's needs. The plant loses freshness and wilts; the leaves change colour from green to yellow.

Finally, the plant dies. The soil water content at the stage where the plant dies is called the permanent wilting point. The soil still contains some water, but it is too difficult for the roots to suck it from the soil.

Total Available water

As the water content above field capacity cannot be held against the force of gravity and will drain and as the water content below wilting point cannot be extracted by plant roots, the total available water in the root zone is the difference between the water content at field capacity and wilting point.

Readily Available Water

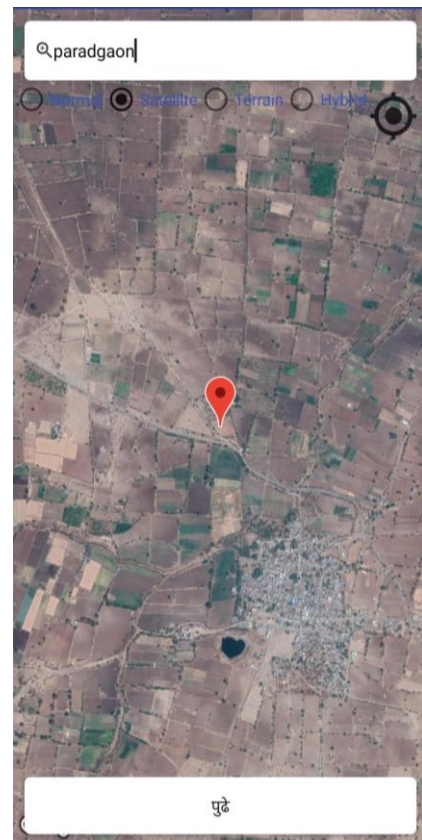
Although water is theoretically available until the wilting point, the crop water uptake is reduced well before the wilting point is reached. Where the soil is sufficiently wet, the soil supplies water fast enough to meet the atmospheric demand of the crop, and water uptake equals ET_c. As the

soil water content decreases, water becomes more strongly bound to the soil matrix and is more difficult to extract. The fraction of TAW that a crop can extract from the root zone without suffering water stress is the readily available soil water. $RAW = p \text{ TAW}$. RAW is the readily available soil water in the root zone [mm], p average fraction of Total Available Soil Water (TAW) that can be depleted from the root zone before moisture stress (reduction in ET) occurs. The factor p differs from one crop to another. A value of 0.50 for p is commonly used for many crops.

SECTION II

This section discusses the steps for using the app.

Step 1) Start the app



Step 2) Select the location

Different Maps for Farm Location Input can be viewed

- Normal Map
- Satellite Map
- Terrain Map
- Hybrid Map

User has to select a farm and click on Next Button.

ram

मातीचा प्रकार

Clayey

मातीची सखोलता

Very deep (> 100 cm)

जमिनीचा वापर

Agriculture

उतार

3.00316262245178

पीक

Soybean

पावसाचे वर्ष

2018

मान्सून समाप्ती तारीख

10/10/2018

प्रती सिंचन पाणी (अंदाजे mm मध्ये)

35

सिंचनांची संख्या

0

तपशील माहिती रेसेसिंग

चालवा

Figure 3 Inputs to App (set A)

Step 3) Set the inputs(Set A)

- User has to give different inputs such as crop, soil type, soil depth, land use, the slope of the land.
- User has to select an appropriate year to run water balance app.
- Also, additional watering details except rainfall are important. These details are taken as input in the model.
- A number of watering, watering amount and watering dates need to be set up.
- Figure 4 describes inputs set to app.
- In set A inputs, Number of irrigations applied by user is zero.

Step 4) Save the results

- The app generates graph and summary table
- Figure 4 describes output of App for input set A
- The summary table consists summary of different parameters like rainfall, soil moisture, runoff, AET
- Two graphs are generated one with daily values and other with cumulative values. Graph typically explains relation between all the crop related parameters
- In daily graph, value of each parameter is plotted day wise while in cumulative graph daily values are added together and plotted day wise
- In day wise graph PET shows water required for crop in red color line while AET shows actual water consumption in green color line. Difference of these two values is actual crop deficit.
- In cumulative table run off is described with pink line while ground water recharge is described with green line.
- From summary table user can understand values both at monsoon and crop end.
- It shows there is total deficit of 100mm.As user has not applied any external watering this deficit is very high and can be covered with manual water application

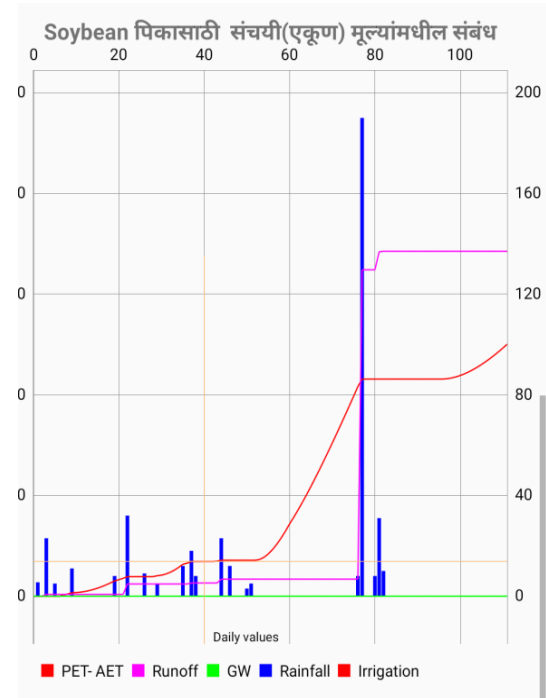
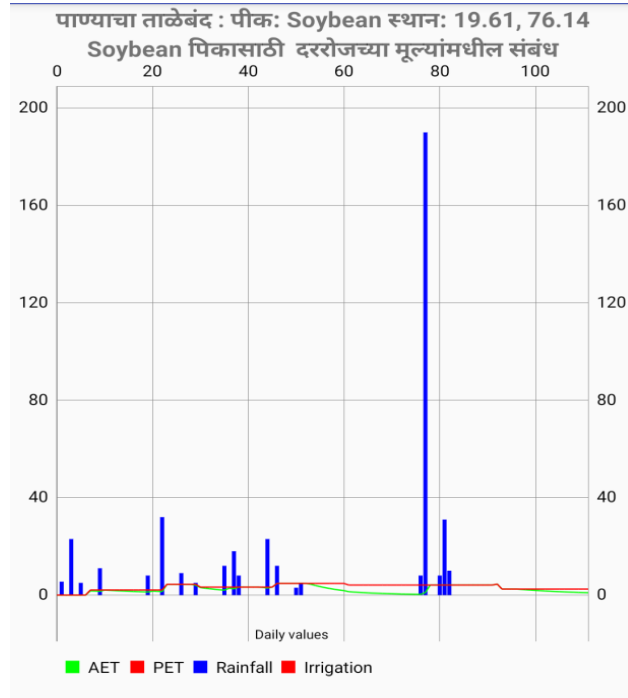
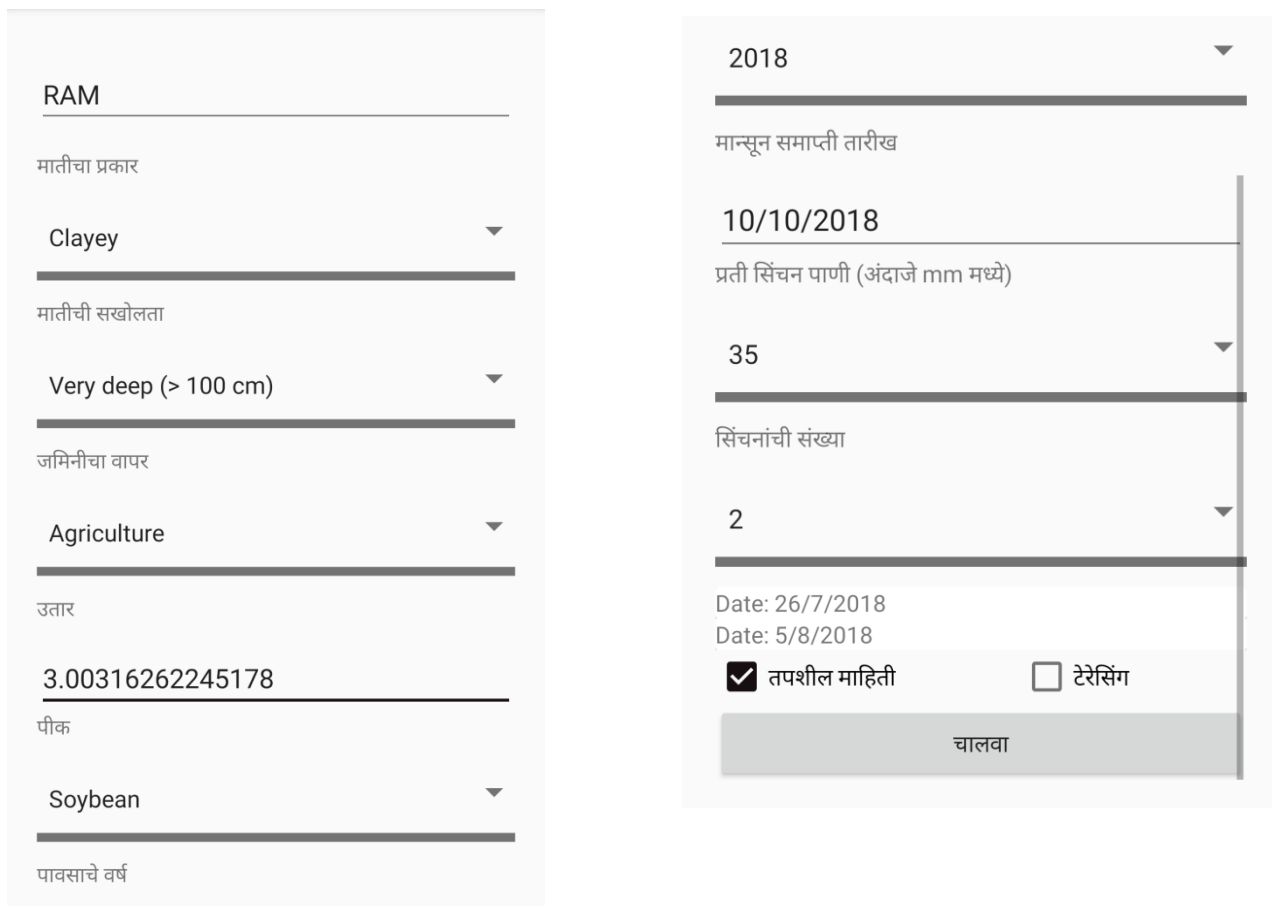


Figure 4 Output from App for Set A

Step 3) Set the inputs (Set B)

- In this part we are going to see example when user has applied irrigation manually.
- Now user has to set two important inputs if external irrigation is applied
 - I. Number of watering applied
 - II. dates of watering per application
- Notice, here all the other parameters are kept same as set A except number of watering.
- Dates of watering are selected as 26 July and 5 August 2018.
- Figure 5 shows inputs to app set B



The image displays two panels of an application interface for setting irrigation parameters (Set B).

Left Panel (Crop and Soil Details):

- RAM** (Crop Name)
- मातीचा प्रकार** (Soil Type): Clayey
- मातीची सखोलता** (Soil Depth): Very deep (> 100 cm)
- जमिनीचा वापर** (Land Use): Agriculture
- उतार** (Slope): 3.00316262245178
- पीक** (Crop): Soybean
- पावसाचे वर्ष** (Rainfall Year):

Right Panel (Irrigation Details):

- Year:** 2018
- मान्सून समाप्ती तारीख** (Monsoon End Date): 10/10/2018
- प्रती सिंचन पाणी (अंदाजे mm मध्ये)** (Water per irrigation in mm): 35
- सिंचनांची संख्या** (Number of irrigations): 2
- Date:** 26/7/2018 and 5/8/2018
- Manual Irrigation:** तपशील माहिती (Manual Irrigation details)
- Telesing:** टेरेसिंग (Telesing)
- चालवा** (Apply)

Figure 5 Inputs to App (Set B)

Step 4) Save the results(Set B)

- Figure 6 shows output from app for set B. As manual watering is applied by user deficit is expected to drop.
- Earlier it was 100 mm, Now with two watering each of 35 mm deficit dropped to 51mm
- Notice the change in soil moisture for input set A and set B is by 6mm for crop end
- from 70 mm water applied externally AET is increased by 49mm. Runoff occurred by 15 mm. Soil moisture improved by 6mm.
- Summary tables from figure 6 and 4 can distinguish difference in deficit due to manual watering by user.

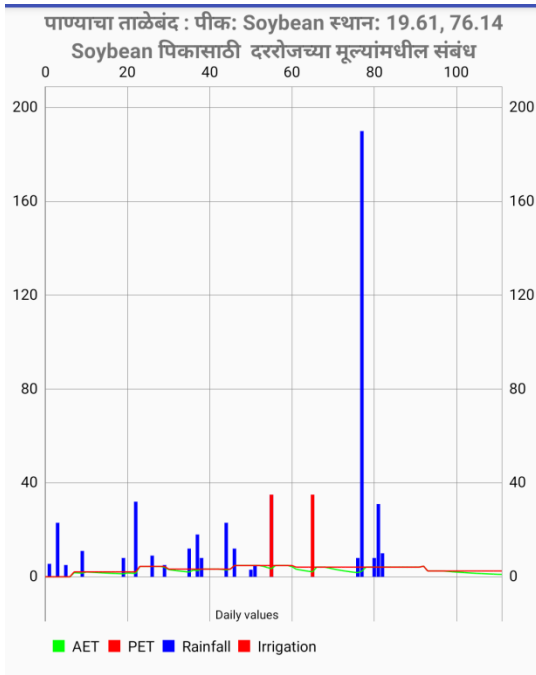
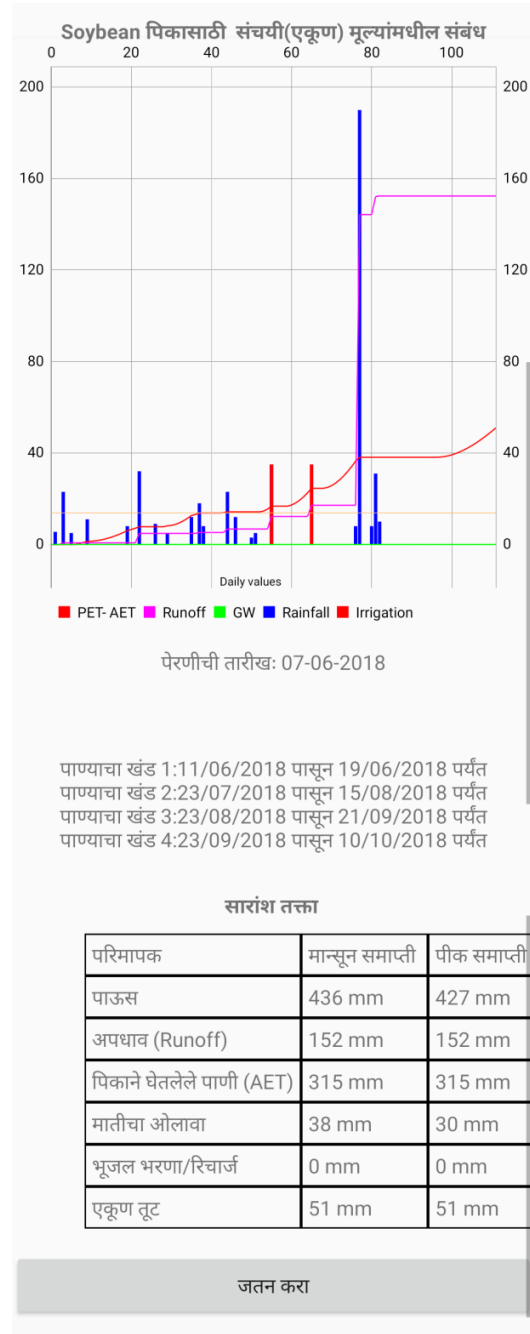


Figure 6 Output from App Set B



PocraApp Findings.

(Ashish wankhade)

1. The rainfall data starts with 01/06/YYYY i.e year span is from 1 June to 31 May.
2. In the skymet api, sometimes data is missing for few circles.
3. The code repository is not on git.
4. Sometimes the VM allocated to backend for the app i.e in host-4 PostgressApp (125), the VM runs out of space. The error logs n /var/log/apache2 are created in GBs.
5. The rainfall data is stored on a yearly basis in files on server.
6. Skymet have few more circles than what the app currently supports. To incorporate the same, we need to define the geometry around the latitude and longitude of that circle. Or we can add them in existing circles based on closet distance.
7. To daily sync the recent data, one script (or cronjob) will be running on the server which will automatically append the latest data at appropriate files.

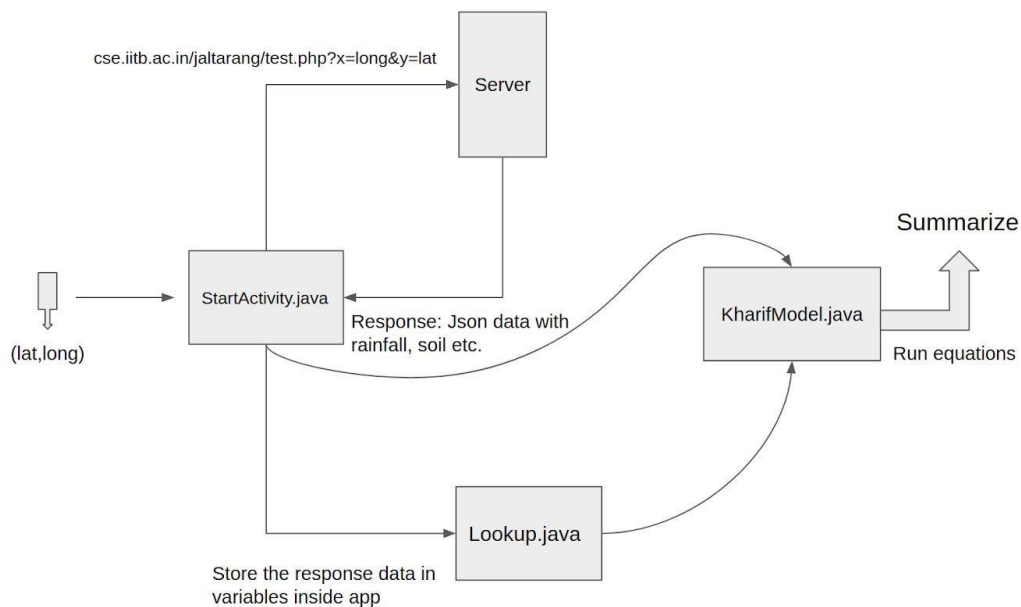


Fig.7 Basic structure of the Android app.

Improvements in PoCRA android application:-

1. Dynamic ET calculation for the year 2019.
2. Integration of weather data from skymet api.
3. User Interface improvement in summary page graphs and layout.
4. User Interface improvement in user form.
5. Server configuration for automatically fetching data from skymet api.
6. Creation and population of appropriate database tables to store data from skymet api.
7. Automatic scripts written on server to fetch data from skymet. This script runs at 01:00 everyday.
8. Date label on X-axis added for the graphs created on summary page.
9. Different irrigation methods
10. Sowing date option

```
void dynamicEt(Double[] temp_avg, Double[] temp_min, Double[] temp_max, String district_ta_circle, Double lat, Double longi){
    double[] et0 = {7.51, 7.51, 7.51, 7.51, 7.51, 7.51, 7.51, 7.51, 7.51, 7.51, 7.51, 7.51, 7.51, 7.51, 7.51, 7.51, 7.51, 7.51, 7.51, 7.51, 7.51};
    // double Ra = 1; //needs to be calculated
    int numberOfDays = temp_avg.length;
    for(int i=0; i<numberOfDays; i++){
        int M = Calendar.getInstance().get(Calendar.MONTH);
        int D = Calendar.getInstance().get(Calendar.DATE);
        int j = (275*(M/9)-30+D)-2;
        if(M < 3)
            j+=2;
        int year = Calendar.getInstance().get(Calendar.YEAR);
        if(year%4 ==0 && year%100 != 0 && M>2)
            j++;

        int J = j+1; //given equation for J.
        J = j;
        double dr = 1 + 0.033*Math.cos((2*3.14*J)/365);
        double Gsc = 0.082;
        double phi = (lat*3.14)/180;
        double tan_phi = Math.tan(phi);
        double delta = 0.489*Math.sin((2*3.14*J)/365-1.39);
        double omega_s = Math.acos(-(tan_phi*Math.tan(delta)));
        double Ra = (24*60*Gsc*dr*(omega_s*Math.sin(phi)*Math.sin(delta)+Math.cos(phi)*Math.cos(delta)*Math.sin(omega_s)))/3.14;
        double Tmean = temp_avg[i];
        double Tmax = temp_max[i];
        double Tmin = temp_min[i];
        et0[i] = 0.0023*(Tmean+17.28)*Math.pow((Tmax-Tmin), 0.5)*Ra*0.408;
    }
    for(int i=numberOfDays; i<et0.length; i++){
        et0[i]=0.0;
    }
    dynamic_et.put(district_ta_circle, et0);
}
```

Fig 8. Code for ET calculation which takes parameters from skymet api

Server scripts for scraping and populating db :

```
import psycopg2 import
datetime, json import
requests import os from
datetime import date try
:
connection = psycopg2.connect(user = "postgres",password = "postgres",host = "localhost",database="gisdata")
```

```

cursor = connection.cursor()
# Print PostgreSQL Connection properties
print ( connection.get_dsn_parameters(),"\n")

# Print PostgreSQL version
cursor.execute("SELECT version();") record =
cursor.fetchone() print("You are connected
to - ", record,"\n") except (Exception,
psycopg2.Error) as error :
    print ("Error while connecting to PostgreSQL", error) st_file =
open('/var/www/html/Data2/last_scrap_date.txt') file_content =
st_file.read() st_file.close() ex_date = file_content.split("-") s_dt
= datetime.date(int(ex_date[0]), int(ex_date[1]), int(ex_date[2]))
s_dt = s_dt + datetime.timedelta(days=1) today = date.today() yr =
s_dt.year if s_dt.month<6: yr =yr-1
payload = {'uname':183050078,'passwd':'nothing.is.plan977314'
} internet = "https://internet.iitb.ac.in/index.php" p =
requests.post(internet,data=payload) while s_dt < today:
    retries=0 url = "https://www.skymetweather.com/media/apiawscurrentdata-
mahavedh.php?date="+str(s_dt) print(url)
    while retries<10:
        try: page = requests.get(url, auth=('pocra',
'pocra@2k18')) s = str(page.json())
break except: retries = retries+1
        pass
    if retries>=10:
        break
    print(s) s =
s.replace('\','\"') s =
s.replace('None','-1')
data = json.loads(s)
    for station in data['allStationsHourlyData']:
        x = station['longitude']
y = station['latitude']
district = station['district']
taluka = station['taluka']
circle = station['circle']
print(x,y,circle) min_temp =
[]
    for data in station['stationdata']:

```

```
min_temp.append(data['data'][0])    if
len(min_temp) != 0:
    minimum_temp = min(min_temp)
else:
    minimum_temp=-9999
max_temp = []
for data in station['stationdata']:
```

```

        max_temp.append(data['data'][1])
if len(max_temp)!=0:
    maximum_temp = max(max_temp)
else:
    maximum_temp=-9999
    total_temp = 0.0
count = 0
    for data in station['stationdata']:
total_temp += data['data'][2]
count +=1
    if total_temp != 0.0:
        avg_temp = total_temp/count
else:
    avg_temp=-9999
    rainfall_day = 0
    for data in station['stationdata']:
rainfall_day += data['data'][-1]
    #sql = "INSERT INTO skymet_data (geom,district,taluka,circle,lat,long)
VALUES(ST_GeometryFromText('MULTIPOINT(%s %s)',4326),%s,%s,%s,%s,%s)"
    #cursor.execute("INSERT INTO public.skymet_data (geom,district,taluka,circle,lat,long)
VALUES(ST_GeometryFromText('MULTIPOINT(%s %s)',4326),%s,%s,%s,%s,%s)",(x,y,district,taluka,circle,y,x))
#cursor.execute("INSERT INTO skymet_data (geom, circle1, taluka, district, lat, longi, temp_min,
temp_max,temp_avg, rainfall) VALUES(ST_GeometryFromText('MULTIPOINT(%s
%s)',4326),%s,%s,%s,%s,%s,ARRAY[%s],ARRAY[%s],ARRAY[%s],ARRAY[%s])", (x,y, circle, taluka, district, y, x, minimum_temp, m
aximum_temp, avg_temp, rainfall_day))
    if s_dt.day == 1 and s_dt.month == 6:
        cursor.execute("INSERT INTO skymet_data (skyid,temp_min, temp_max, temp_avg, rainfall,year)Select id as
skyid, '{}' as temp_min, '{}','{}','{}',%s::numeric from skymet_api where circle1=%s' AND district=%s' AND
taluka=%s'",(yr,circle,district,taluka))
        #cursor.execute("INSERT INTO skymet_data (skyid,temp_min, temp_max, temp_avg, rainfall,year)
VALUES(Select skyid from skymet_api where circle1=%s,district=%s,taluka=%s,
'{}','{}','{}','{}',%s)",(circle,district,taluka,yr))
        cursor.execute("UPDATE skymet_data SET
temp_min=array_cat(temp_min,ARRAY[%s::numeric]), temp_max=array_cat(temp_max,ARRAY[%s::numeric]),
temp_avg=array_cat(temp_avg,ARRAY[%s::numeric]),rainfall=array_cat(rainfall,ARRAY[%s::numeric]) WHERE year=%s
AND skyid IN (Select id from skymet_api where lat=%s::numeric AND longi=%s::numeric)"
,(minimum_temp,maximum_temp,avg_temp,rainfall_day,yr,y,x))
        connection.commit()
        st_file = open(
'/var/www/html/Data2/last_scrap_date.txt','w')
        st_file.write(str(s_dt))
        st_file.close()
        s_dt = s_dt
+ datetime.timedelta(days=1)

connection.close()

```


UI changes in PoCRA android app:-

Farmwater

ashish

मातीचा प्रकार
Gravelly clay

मातीची सखोलता
Moderately deep (25 to 50 cm)

जमिनीचा वापर
Agriculture

उतार
3.00316262245178

पीक
Cotton

पावसाचे वर्ष
2019

मान्सून सुरुवात तारीख
15/06/2013

मान्सून समाप्ती तारीख
10/10/2019

सिंचनाची संख्या
2

तारीख
Date: 15/8/2019

प्रती सिंचन पाणी
40

सिंचन प्रकार
Flood

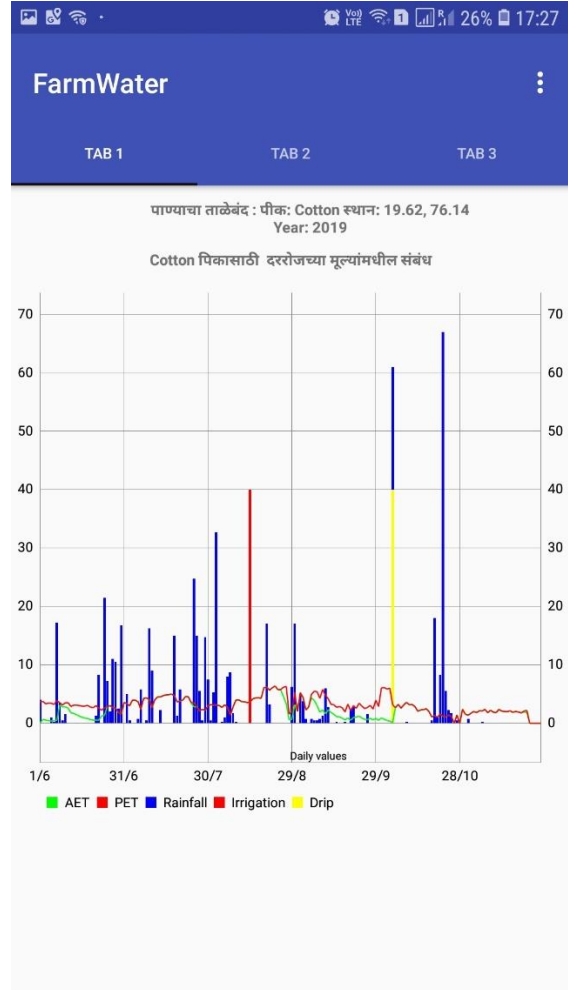
Date: 5/10/2019

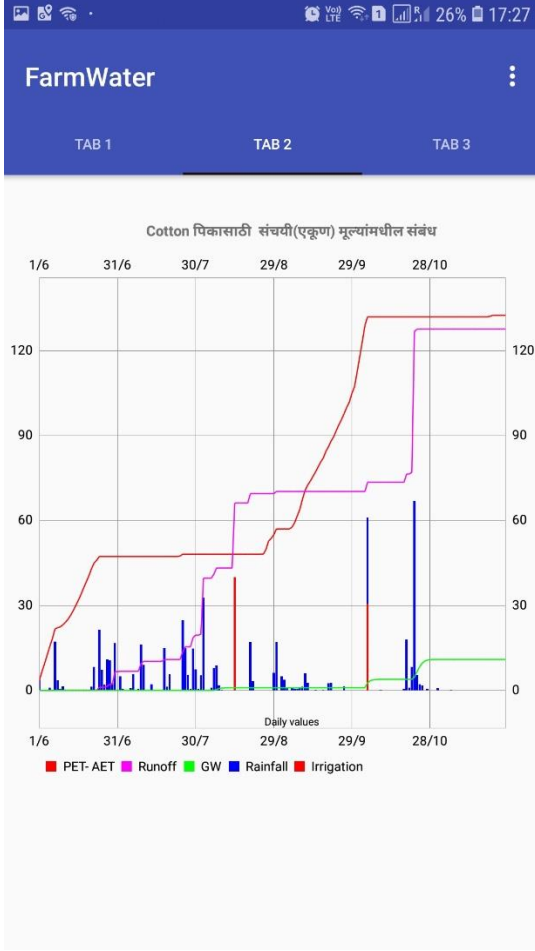
40

Drip

तपशील माहिती ट्रेसिंग

चालवा





FarmWater

TAB 1 TAB 2 TAB 3

पेरणीची तारीख: 15-06-2013

सिंचनाची तारीख [15/8/2019, 5/10/2019]
 पाण्याचा खंड 1:11/06/2019 पासून 20/06/2019 पर्यंत
 पाण्याचा खंड 2:11/08/2019 पासून 20/08/2019 पर्यंत
 पाण्याचा खंड 3:27/09/2019 पासून 04/10/2019 पर्यंत

सारांश तक्ता

परिमाणवक	मान्सून समाप्ती	पौक समाप्ती
वाऊस	401 mm	507 mm
अपघाव (Runoff)	74 mm	128 mm
पिकाने घेतलेले पाणी (AET)	346 mm	425 mm
मातीचा ओलावा	61 mm	24 mm
भूजल भरणा/रिचार्ज	4 mm	11 mm
एकूण तूट	132 mm	132 mm

जतन करा

References:-

1] POCRA Report on description of water balance framework

<https://www.cse.iitb.ac.in/~pocra/Month2Report.pdf>

2] Link for downloading App

https://www.cse.iitb.ac.in/~pocra/Android_app/pocra_final.apk

<https://drive.google.com/file/d/1ekpNTto99aP-q5-QTEIL1tEwGq8nx8Twq/view?usp=sharing>