

# Nithya Panchangam

---

Vedic Calendar

HARIHARAN KRISHNAMURTHY

Nov '21

# Agenda

---

- *Panchangam – Schools of Thought*
- *Proposal to Integrate Hindu Calendar (2 Approaches)*
- *Proposal to make VedicCalendar Flexible (2 Approaches)*
- *Nithya Panchangam Software Architecture*
- *Interfaces to VedicCalendar Java Library*
- *Nithya Panchangam App Screenshots*
- *References*

# Panchangam – Core Aspects

## Ayanamsa

- Planetary Positions
- Sidereal
- Sunrise
- Sunset

Systems / Modes:  
❖ Chitrapaksha  
❖ Lahiri  
❖ Krishnamurthi

## Calendars

- Luni-solar
- Solar
- Lunar

Systems / Modes:  
❖ Vakyam  
❖ Surya Siddhanta  
❖ Drik Ganitham

## Dina Vishesham

- Pancha-Angam
- Rules to determine vishesham

Systems / Modes:  
❖ Sauramaanam  
❖ Chaandramaanam  
❖ Amanta / Purnimanta

## Panchangam – Schools of Thought (Contd.)

---

*Panchangam calculation requires knowledge of two important heavenly bodies – Ravi (Sun) and Chandra (moon).*

***Three schools of Thought (could be more!)***

- ***Vakyam*** is an ancient system where planetary motions are described in simple sentences (hence the *vakya*). *Vakya Panchangam* is computed based on *Slokas* or *Vakyas* handed down to the practitioners of this school.

## Panchangam – Schools of Thought (Contd.)

---

- *Surya Siddhantham* mentions that one should observe the sky and make necessary corrections to planetary formulae (*Bija Samskar*) in order to make an accurate Panchangam.
- The modern astronomical method of *Drik Ganitham* or *Drik Siddhantham* allows validation of the calculated planetary positions using rigorous mathematical equations resulting in quite precise positions of the planets.

*(Nithya Panchangam supports all three as configurable options!)*

# Proposal to Integrate Hindu Calendar

---

## *Architectural Considerations*

- *Which Operating System(s) should be supported?*
- *Which language(s) should be supported?*
- *Where & how to archive & version control the repository for future enhancements?*
- *How to accommodate different schools of thought for Panchangam calculations?*
- *How to provide a uniform & unified interface despite the inherent differences?*

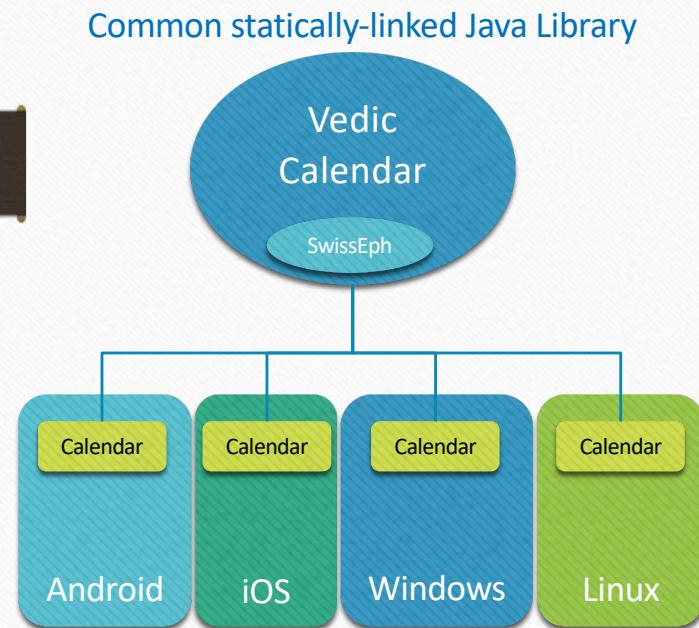
# Proposal to Integrate Hindu Calendar (Contd.)

---

## *Architectural Considerations*

- *Simple & Scalable*
- *Portable & Easy to Migrate to newer platforms*
- *Easy to Maintain*
- *Flexible*
- *Extendable*

# Proposal to Integrate Hindu Calendar (Approach – 1)



## *VedicCalendar as a single-source Java Library*

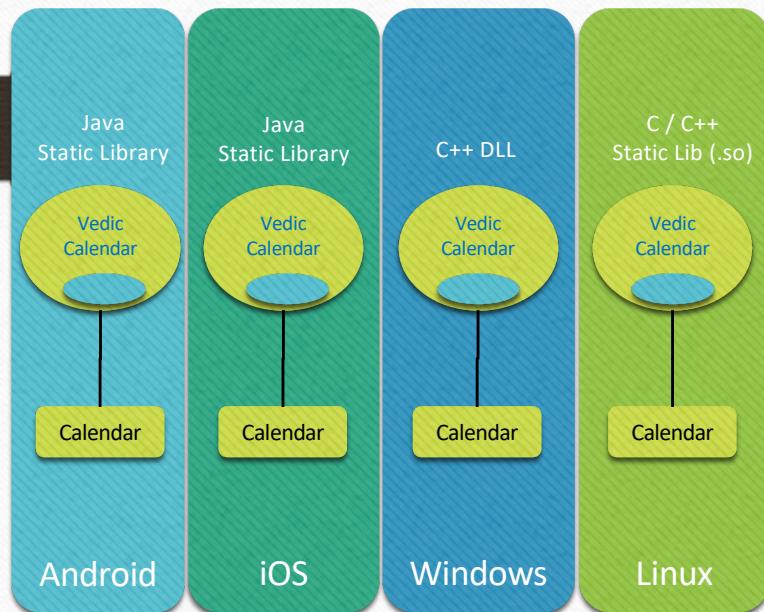
### Pros

- ***Simple & Easy-to-maintain*** in the long run
- ***Extendable*** on newer platforms with limited efforts
- ***Easy to port & migrate*** to newer platforms
- ***Object-model flexible*** to extend to customize Panchangam calculations

### Cons

- ***Can be extended to Java-based platforms only***

# Proposal to Integrate Hindu Calendar (Approach – 2)



*VedicCalendar specific for each native platform*

## Pros

- *Independent language, linking mechanism as per platform needs (plug-ins / DLL / dynamic libraries etc)*

## Cons

- *Difficult to maintain object-model to customize Panchangam calculations*
- *Not easily extendable / portable on newer platforms*
- *Repository for each native platform is difficult to maintain in the long run*
- *Newer features need specific effort to port on each newer platform*

# Proposal to make VedicCalendar Flexible (Approach – 1)

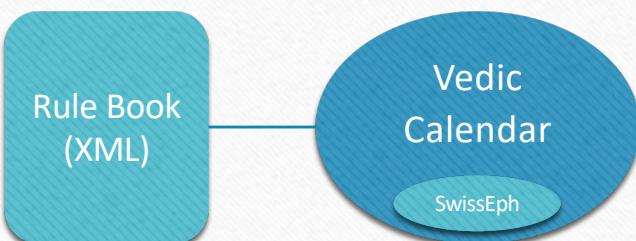
**Derive child class from VedicCalendar for custom calculations**

**Pros**

- *Uniform interface to OS / platforms*
- *Extendable to configure calculations as per one's own sampradhaya / parampara*

**Cons**

- *New paradigm / grammar needs to be created to configure all formula*
- *Can get complex & cryptic at times*



VedicCalendar does not support this as of now!

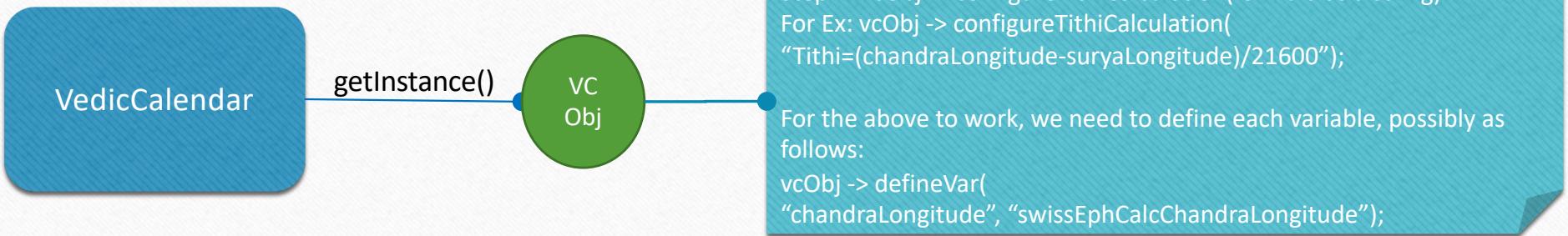
## Proposal to make VedicCalendar Flexible (Approach – 1)

### ***Flexible Interfaces – An Example!***

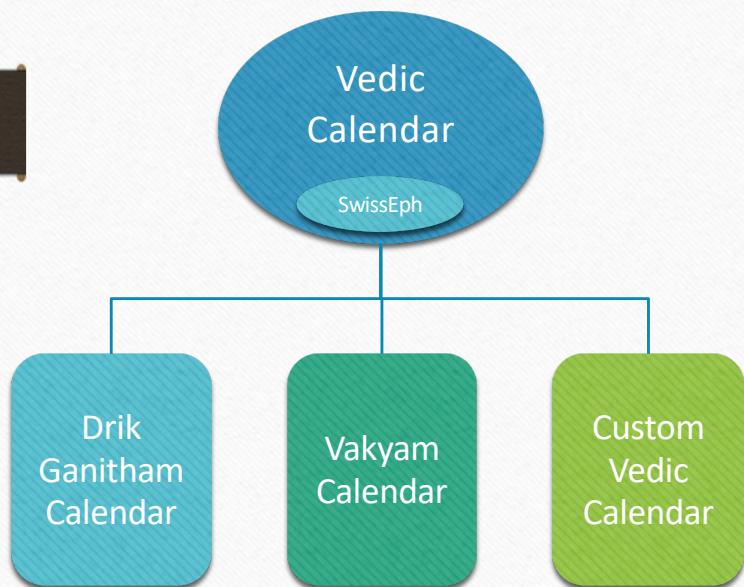
Tithi is calculated from the distance between Chandra and Ravi.

$$\text{Tithi} = (\text{Chandra Longitude} - \text{Surya Longitude}) / 360^\circ$$

Now how to configure the above calculation in VedicCalendar? (This is just a proposal. Need to think through!)



# Proposal to make VedicCalendar Flexible (Approach – 2)



**Derive child class from VedicCalendar for custom calculations**

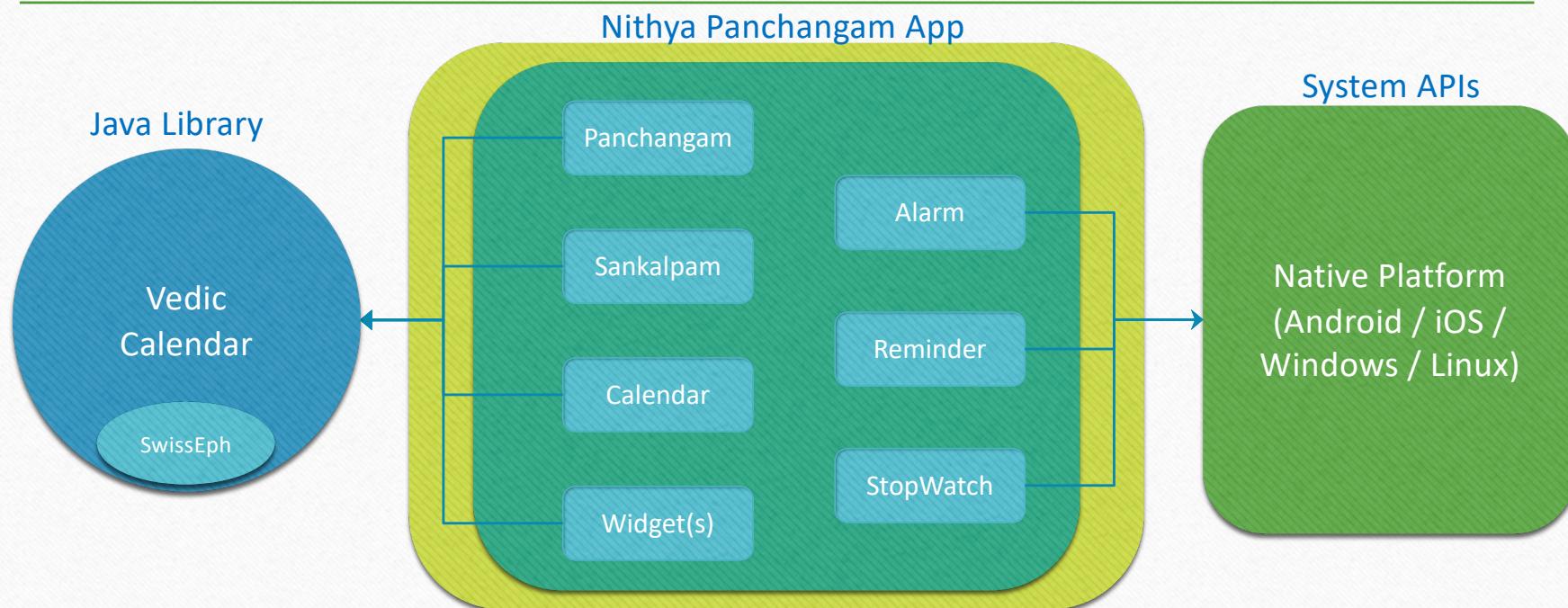
## Pros

- *Each sampradhaya / parampara has full powers to own & drive their own calculations*

## Cons

- ***Difficult to maintain in the long run***
- ***Each platform-owned Calendar(s) need to be modified to use the derived object for custom implementation***

# Nithya Panchangam Software Architecture



# Interfaces to VedicCalendar Library



- **Panchangam Calculations (Primary)**  
(Tithi, Vaasaram, Nakshatram, Yogam, Karanam)
- **Panchangam Calculations (Secondary)**  
(Samvatsaram, Ayanam, Ritu, Maasam, Paksham, Raasi)
- **Panchangam Calculations (Misc)**  
(Lagnam, Horai, Amruthathi Yogam, Shuba Kaalam,  
Rahu Kaalam, Yamagandam, Gulika Kaalam)
- Dina Vishesham Calculations

# Interfaces to VedicCalendar Library (*Contd.*)

---

## Panchangam Primary Interfaces

1. getThithi()
2. getNakshatram()
3. getYogam()
4. getKaranam()
5. getVaasaram()

## Panchangam Supplementary Interfaces

1. getSunrise()
2. getSunset()
3. getSamvatsaram()
4. getAyanam()
5. getRithu()
6. getSauramaanamMaasam()
7. getChaandramaanamMaasam()
8. getPaksham()
9. getRaasi()

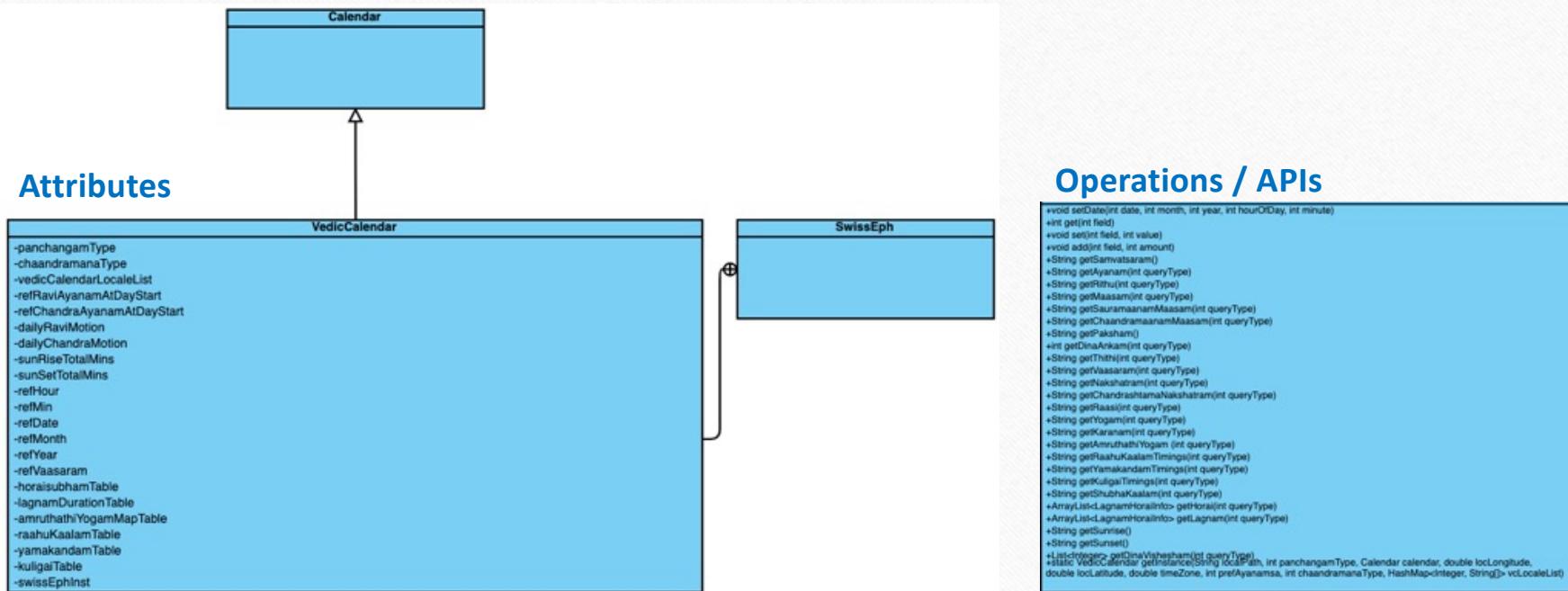
## Panchangam Misc Interfaces

1. getDinaVishesham()
2. getChandrashtamaNakshatram()
3. getLagnam()
4. getHorai()
5. getAmruthathiYogam()
6. getShubhaKaalam()
7. getRaahuKaalamTimings()
8. getYamakandamTimings()
9. getKuligaiTimings()

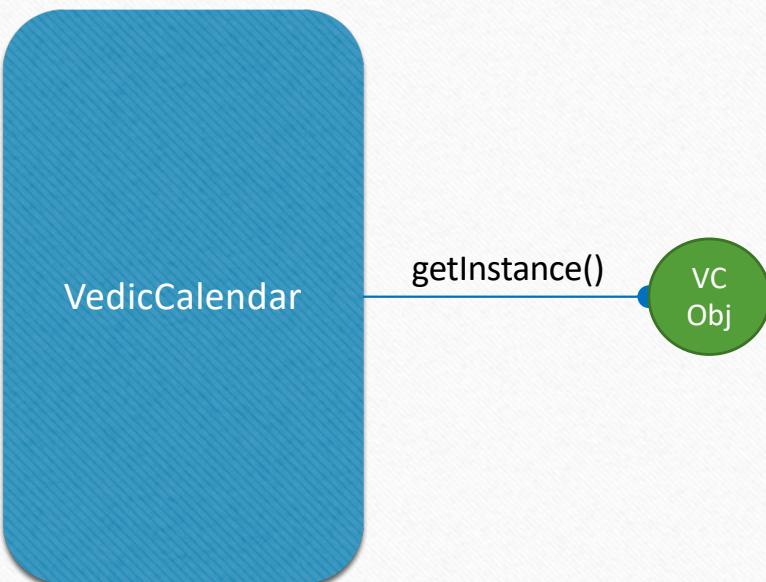
This can be improved based on feedback!

## Class Diagram

# Interfaces to VedicCalendar Library (Contd.)



# Interfaces to VedicCalendar Library (*Contd.*)



## Parameters to **getInstance()** static API:

- [1] "Path" to "app\_assets" directory
- [2] Panchangam Type  
PANCHANGAM\_TYPE\_DRIK\_GANITHAM /  
PANCHANGAM\_TYPE\_TAMIL\_VAKHYAM /  
PANCHANGAM\_TYPE\_TELUGU\_PANCHANGAM /  
PANCHANGAM\_TYPE\_KANNADA\_PANCHANGAM
- [3] Calendar object to indicate date
- [4] Longitude of the location
- [5] Latitude of the location
- [6] Timezone of the location
- [7] Ayanamsa Type (AYANAMSA\_CHITRAPAKSHA / AYANAMSA\_LAHIRI)
- [8] Chaandramaana Type  
CHAANDRAMAANAM\_TYPE\_AMANTA /  
CHAANDRAMAANAM\_TYPE\_PURNIMANTA
- [9] HashMap<Integer, String[]> - string arrays that contain locale-specific texts for each Panchangam field.

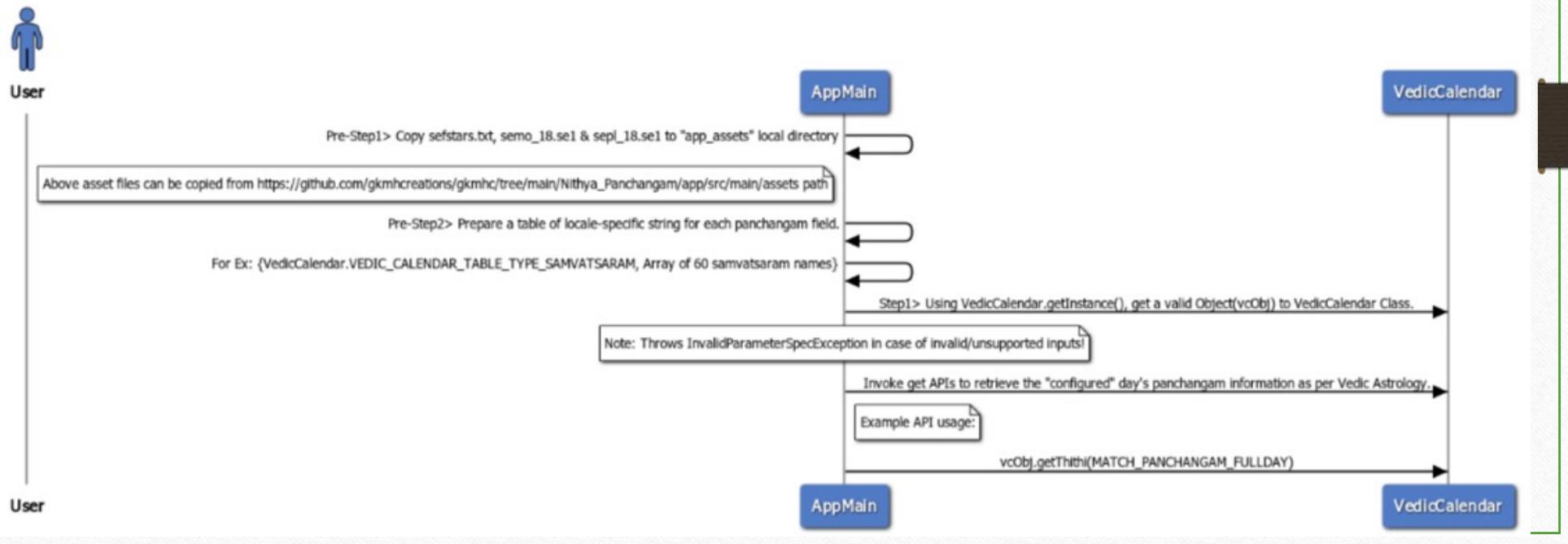
# Interfaces to VedicCalendar Library (Contd.)

17 Panchangam fields for which locale-specific array definitions are expected as input to getInstance().

For ex: refer buildVedicCalendarLocaleList() in MainActivity.java (Check References slide)

- `VEDIC_CALENDAR_TABLE_TYPE_SAMVATSARAM`
- `VEDIC_CALENDAR_TABLE_TYPE_AYANAM`
- `VEDIC_CALENDAR_TABLE_TYPE_RITHU`
- `VEDIC_CALENDAR_TABLE_TYPE_SAURAMANA_MAASAM`
- `VEDIC_CALENDAR_TABLE_TYPE_CHAANDRAMANA_MAASAM`
- `VEDIC_CALENDAR_TABLE_TYPE_PAKSHAM`
- `VEDIC_CALENDAR_TABLE_TYPE_THITHI`
- `VEDIC_CALENDAR_TABLE_TYPE_SANKALPA_THITHI`
- `VEDIC_CALENDAR_TABLE_TYPE_RAASI`
- `VEDIC_CALENDAR_TABLE_TYPE_NAKSHATHRAM`
- `VEDIC_CALENDAR_TABLE_TYPE_SANKALPA_NAKSHATHRAM`
- `VEDIC_CALENDAR_TABLE_TYPE_YOGAM`
- `VEDIC_CALENDAR_TABLE_TYPE_KARANAM`
- `VEDIC_CALENDAR_TABLE_TYPE_VAASARAM`
- `VEDIC_CALENDAR_TABLE_TYPE_DHINAM`
- `VEDIC_CALENDAR_TABLE_TYPE_HORAI`
- `VEDIC_CALENDAR_TABLE_TYPE_AMRUTATHI_YOGAM`

# Interfaces to VedicCalendar Library (Contd.)



# Interfaces to VedicCalendar Library (Contd.)

## *Input parameter for some APIs:*

**MATCH\_PANCHANGAM\_FULLDAY – For Ex: Panchami (14:05) > Sashti**

**MATCH\_SANKALPAM\_EXACT – For Ex: Sashti (based on current time)**

**MATCH\_PANCHANGAM\_PROMINENT – For Ex: Sashti (based on prominent thithi during the day!)**

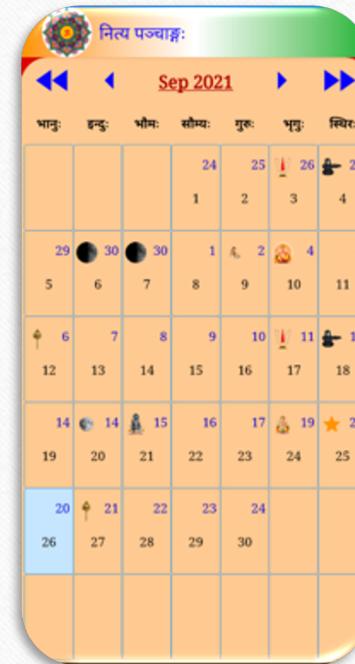
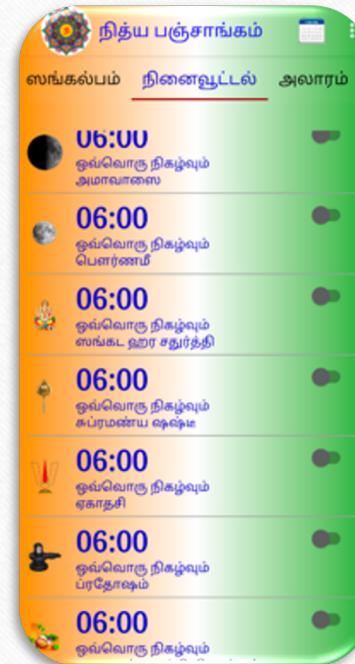
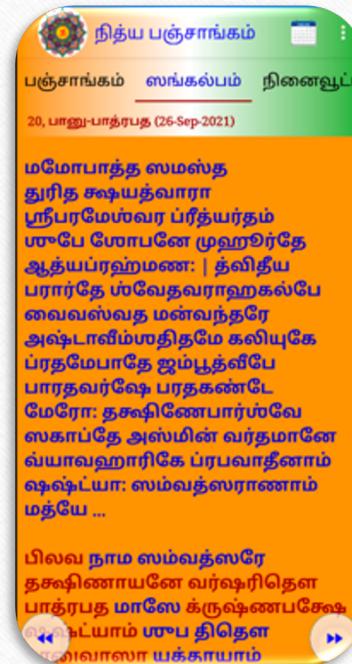
## *Sample Usage:*

```
vcObj.getAyanam(VedicCalendar.MATCH_PANCHANGAM_FULLDAY)  
vcObj.getDinaAnkam(VedicCalendar.MATCH_PANCHANGAM_FULLDAY);  
vcObj.getSauramaanamMaasam(VedicCalendar.MATCH_PANCHANGAM_FULLDAY);  
vcObj.getHorai(VedicCalendar.MATCH_SANKALPAM_EXACT);  
vcObj.getDinaVishesham(VedicCalendar.MATCH_PANCHANGAM_PROMINENT);
```

Check References slide for links to sample & UT files.



# Nithya Panchangam Android App Screenshots



# References

---

- *Algorithm for the Panchangam calculations is referred from Karanam Ramkumar's guidelines below:*  
<https://fdocuments.in/document/panchangam-calculations.html>
- *Sample reference to SwissEph library for getting Longitude & Latitude for Ravi & Moon and Udhaya Lagnam:*  
<http://th-mack.de/download/contrib/Vedic Houses.java>
- *Github link to VedicCalendar Implementation:*  
[https://github.com/gkmhcreations/gkmhc/blob/main/Nithya\\_Panchangam/app/src/main/java/com/gkmhc/utils/VedicCalendar.java](https://github.com/gkmhcreations/gkmhc/blob/main/Nithya_Panchangam/app/src/main/java/com/gkmhc/utils/VedicCalendar.java)
- *Github link to sample UT file for reference:*  
[https://github.com/gkmhcreations/gkmhc/blob/main/Nithya\\_Panchangam/app/src/androidTest/java/com/gkmhc/vedanta/nithya\\_panchangam/VedicCalendarUnitTest.java](https://github.com/gkmhcreations/gkmhc/blob/main/Nithya_Panchangam/app/src/androidTest/java/com/gkmhc/vedanta/nithya_panchangam/VedicCalendarUnitTest.java)
- *Github link to MainActivity.java (to refer to sample implementation of buildVedicCalendarLocaleList()):*  
[https://github.com/gkmhcreations/gkmhc/blob/main/Nithya\\_Panchangam/app/src/main/java/com/gkmhc/vedanta/nithya\\_panchangam/MainActivity.java](https://github.com/gkmhcreations/gkmhc/blob/main/Nithya_Panchangam/app/src/main/java/com/gkmhc/vedanta/nithya_panchangam/MainActivity.java)
- *Google Android Playstore store link to download "Nithya Panchangam" App:*  
[https://play.google.com/store/apps/details?id=com.gkmhc.vedanta.nithya\\_panchangam&hl=en\\_IN&gl=US](https://play.google.com/store/apps/details?id=com.gkmhc.vedanta.nithya_panchangam&hl=en_IN&gl=US)  
Note: "Nithya Panchangam" is distributed under GNU GPL:  
<http://www.gnu.org/licenses/old-licenses/gpl-2.0.html>

Thank you

---