Sports Inviter Bot - "*Sports invitation made easy! "*

Sends out multiple Google Calendar invites with loaded goodies on upcoming games.

Problem to Solve:

To make life easier for people organizing and / or part of teams involved in various outdoor team sports/games (e.g. cricket, baseball, soccer, football, field hockey etc.) to plan and prepare in advance to participate, track, notify and report on 'who', 'what', 'when' and 'where' information of different games in a league or tournament.

About the project:

The Sports inviter bot will read inputs excel files on team, schedule, player & venue details and compile a Google calendar invite with event / game details, weather details and player trip details for the players of different teams involved in the game or tournament.

Value:

Reduce the time taken and manual effort on repetitive tasks to communicate game details, improve quality by removing manual error and increasing stakeholder satisfaction in whole process of participating in the games.

Benefits:

- Game Organizers:
  Improve team's participation in the tournament/league. Having all the games schedules in a Calendar will help them better organize and communicate changes to the games.
  For instance, assume a tournament has 5 different divisions A, B, C, D & E and each division has 10 teams. Having 5 google calendar each for league will help the Organizer to access different data of the tournament.

- Team Captains/Mangers:
  Team captains don't not have to manually share game details and track each team member's availability.

- Team members (including captains) :
  Single source of information on all upcoming games including venue, weather and trip details in one place and automatic reminder for the game.

Input files:

1) Teams.xlsx : Excel with 2 columns a) Team names & b)Tournament name [the team has registered]
2) Schedule.xlsx: Excel with game schedule details for a tournament which has a) Game date b) Start and end time c) Teams playing d) Venue name
   Note: This sheet can be created by scrapping the webpage if the tournament has a website.
3) Venues.xlsx: Excel will have a) Venue name, b) address, c) zip and d) google map link
4) Players.xlsx: Excel sheet with a)player name, b)address and c)email id

Trigger:

Once a week (Any day in a week) – Idea is to send calendar invites only 1 invite per game for the team for the upcoming weekend.

Assumptions:

1) The games are scheduled only during weekends

2) Players travel to the venue in car

High level Process:

1. Teams who needs to request the notification service must register the names in "Teams.xlsx" and "Player.xlsx"
2. The bot will loop through this list and process one team at a time (transaction item).
3. Checks if the team has an upcoming match this weekend in the tournament / league schedule.
4. Collects the Game details like opponent name, venue name, start and end time.
5. Uses the Venue name to find the Venue details like address & zip from Venue input file.
6. The bot then finds the Weather condition like rain, wind & temperature details of the venue on the game day using open weather map API.
7. The bot also calculates the Trip details like distance and duration of travel for each player from the origin to the venue using google map API.
8. Finally, the bot compiles all the above information in a Google calendar invite and sends it out to each of the all the player email id's and updates the transaction status back in "Teams.xlsx" with Success or Failure.
9. The bot then picks the next team (transaction) and continues the same process until all the teams in the transaction list are exhausted.
10. Team manager or captain can track player availability based on the acceptance of the request and players will be notified with all the information of the upcoming game from one calendar invite.

Framework used:

Re-Framework is used to process different transactions for different teams (it uses the transaction from "Teams.xlsx" and not queue). Each row in Teams.xlsx is a transaction and for each team all these modules are executed and the transaction status of "Success! Calendar Invite sent" or "Failed! Calendar Invite not sent" is updated back in "Teams.xlsx" once the transaction is completed.
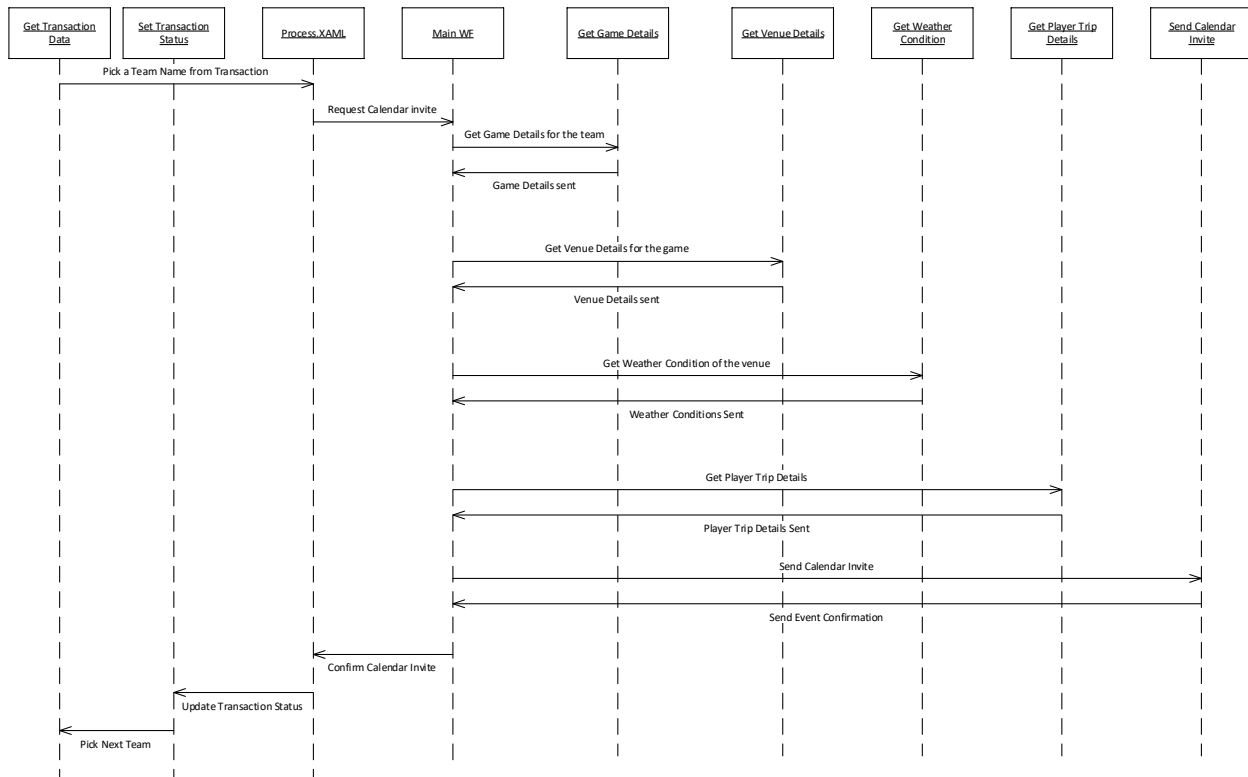
Application used:

The entire process flow depicted in the "Bot" swim lane is automated with UiPath Studio and GSuite Calendar Component (UiPath Go!).

The venue Weather details are received from https://openweathermap.org/api and use Google Location for player Trip details from https://developers.google.com/maps/documentation/directions/start

Modules:

The users can use the entire flow to notify game details to the team members or just pick reusable XAML files to A) Find weather condition based on zip b) Find the distance and duration of trip from origin to destination & c) send just Google calendar invite to list of emails ids with event title and details.

4.1.1 Get Game Details from Schedule Module

- Input: Team name
- Steps: use the team name and get the game details from Schedule.xlsx
- Output: team 1 , team 2, venue name, game date, start and end time

4.1.2 Find Venue Address and Zip Module
- Input: Venue name
- Steps: Use the venue name and find the venue address from Venues.xlsx
- Output: address, zip & google map link

4.1.3 Get Weather conditions based on Zip Module (Open Weather map API) [Reusable]
- Input: Game date, Venue zip , Weather API end point & Weather API Key
- Steps: Open Weather map will use the venue zip and compile the weather condition of the venue in the game date (from 6 A.M to 9 P.M)
- Output: weather condition on Game date on the venue

4.1.4 Calculate Trip details for players Module (distance and duration using Google Location API) [Reusable]

- Input: Origin address , destination address, Google API key, Google Location API endpoint
- Steps: Google Location API will calculate the distance and duration based on the origin and destination addresses
- Output: Distance and Duration

4.1.5 Send Calendar invite module (Google Gsuite Calendar API) [reusable]

- Input: Invitees email id's, Event Title , Event Description, Date, start and end time, location, Google API Key, Google client ID and client Secret
- Steps: Gsuite UiPath activity will use all the inputs to send out the Calendar invite and returns a confirmation in form of Event ID.
- Output: Google Calendar Event ID

How to run?

1. Download the project from GitHub
2. Check the prerequisite.pdf from the downloaded project documents folder and perform the steps like updating UiPath packages needed, input folder setup, and API keys and authentication registrations and generation.
3. Update the Teams.xlsx, schedule.xlsx, Venue.xlsx & Players.xlsx for respective teams [please makes sure you use the same format for all the data types including the start and end time & try to avoid blanks]
4. Open the Data\ config.xlsx and check the Google endpoint & API key, Oauth secret and Open Weather map end point & API keys are updated
5. Run the program and check the calendar.google.com of the account you used to get the google API key. All the Calendar invites will be available there.

Supporting documents:

Process design document, Pre requisites, UiPath Project file, Workflows are available in GitHub code repository.