

Use Case Suite

Project Information

Project: ACAP-BICOL (ACAP 1.0) – Rice Crop Manager Advisory Service (RCMAS) Collaboration

Release Number: 1.0

Date updated: September 12, 2023

Description: This document discusses how Collaborator users may interact with the modified ACAP Bicol (ACAP 1.0) system. Describing the user-system interaction aims to provide methods of visualizing the system architecture, managing scopes, and establishing requirements.

*These use cases assume a working ACAP 1.0 system is in place.

Other use cases that interact directly or indirectly with the collaboration project are described in more detail in the attached ACAP 1.0 – Use Case Suite v4.0 (Lite version) document ([acap 1.0 use case suite v4.0 lite.pdf](#)) for reference.

Use Cases by Functional Area

- **PAGASA Weather Forecast REST API Endpoints**
 - UC-00: Seasonal Weather Forecast API
 - UC-01: 10-Day Weather Forecast API
 - UC-02: Special Weather Forecast API
- **PAGASA Historical Weather Forecast REST API Endpoints**
 - UC-03: Historical Seasonal Weather Forecast API

- UC-04: Historical 10-Day Weather Forecast API
- UC-05: Historical Special Weather Forecast API

- **PAGASA Historical Weather Forecast Data Management**

- UC-06: Archiving and storage of Historical Seasonal Weather Forecast
- UC-07: Archiving and storage of Historical 10 Day Weather Outlook Forecast
- UC-08: Archiving and storage of Historical Special Weather Forecast
- UC-09: Cron job – Auto delete outdated Historical Seasonal Weather Forecast
- UC-10: Cron job – Auto delete outdated Historical 10-Day Weather Outlook Forecast
- UC-11: Cron job – Auto delete outdated Historical Special Weather Forecast

- **Client Authorization**

- UC-12: Create API keys
- UC-13: Authorize API keys

Use Cases by Stakeholder

Collaborator

- UC-00: Seasonal Weather Forecast API
- UC-01: 10-Day Weather Forecast API
- UC-02: Special Weather Forecast API
- UC-03: Historical Seasonal Weather Forecast API
- UC-04: Historical 10-Day Weather Forecast API

- UC-05: Historical Special Weather Forecast API

stakeholder. This list shows only the relevant stakeholders interacting directly with the collaboration project.

System Administrator

- UC-12: Create API keys
- UC-13: Authorize API keys
- UC-06: Archiving and storage of Historical Seasonal Weather Forecast
- UC-07: Archiving and storage of Historical 10 Day Weather Outlook Forecast
- UC-08: Archiving and storage of Historical Special Weather Forecast
- UC-09: Cron job – Auto delete outdated Historical Seasonal Weather Forecast
- UC-10: Cron job – Auto delete outdated Historical 10-Day Weather Outlook Forecast
- UC-11: Cron job – Auto delete outdated Historical Special Weather Forecast

• System Administrator

- This user oversees setting up the system and deploying the front-end and back-end to their respective cloud environments.
- System Administrators are also responsible for creating and authorizing API keys in the API endpoints.

• Administrator

- Users can log in to the website and perform administrative actions such as creating crop recommendation bulletin PDFs and manually updating the PAGASA weather forecast data using ACAP tools.

Stakeholders

The collaboration project has the same stakeholders as ACAP 1.0, including the new Collaborator

• Collaborator

- Authorized users who can make HTTP requests to the new PAGASA REST API endpoints using API keys

Use Cases by Business Object and Actor

BO/Actor	Collaborator	Administrator	System Administrator	System
Weather Forecast	view	Set update	Set update	set update
API Key	use		create	
PAGASA REST API	HTTP request (GET)		authorize	

UC-00: Seasonal Weather Forecast API

Summary:	Enables Collaborators to fetch the latest active PAGASA seasonal weather forecast from ACAP's database
Priority:	Desired
Use Frequency:	Sometimes
Direct Actor/s:	Collaborator
Stakeholder/s:	Collaborator, System Administrator

Prerequisite/s:	1. Authorized API Keys (UC-12, UC-13)
Main Success Scenario:	1. Create a (GET) HTTP request to the PAGASA Seasonal Weather Forecast API in the deployed API URL: https://acap-rcmas.vercel.app/api/weatherforecast 2. Attach the following GET query parameters in the request: <ol style="list-style-type: none"> type=seasonal province=Camarines Sur

	<p>c. key=(YOUR_API_KEY)</p> <p>3. View the success 200 API response results.</p> <p><i>(Please view the PAGASA REST API usage documentation for more information at https://acap-rcmas.vercel.app/)</i></p>
Alternative Scenario:	<p>1. The collaborator may supply incorrect GET query parameters for:</p> <ol style="list-style-type: none"> type=seasonal province=Camarines Sur <p>2. The collaborator may supply an invalid API key for the query parameter:</p> <ol style="list-style-type: none"> key=(YOUR_API_KEY) <p>3. Collaborators will receive a 403 or 500 error response if they supply invalid GET query parameters like described in the alternative scenario steps #1 or #2.</p> <p>4. The REST API endpoint hosted in Vercel may become inaccessible whenever Vercel experiences downtimes.</p>
Notes and Questions:	

UC-01: 10-Day Weather Forecast API

Summary:	Enables Collaborators to fetch the latest active PAGASA 10-day weather forecast from ACAP's database
Priority:	Desired
Use Frequency:	Always (daily)
Direct Actor/s:	Collaborator
Stakeholder/s:	Collaborator, System Administrator
Prerequisite/s:	1. Authorized API Keys (UC-12, UC-13)
Main Success Scenario:	<p>1. Create a (GET) HTTP request to the PAGASA Seasonal Weather Forecast API in the deployed API URL: https://acap-rcmas.vercel.app/api/weatherforecast</p> <p>2. Attach the following GET query parameters in the request:</p> <ol style="list-style-type: none"> type=tenday province=Camarines Sur key=(YOUR_API_KEY) <p>3. View the success 200 API response results.</p> <p><i>(Please view the PAGASA REST API usage documentation for more information at https://acap-rcmas.vercel.app/)</i></p>
Alternative Scenario:	<p>1. The collaborator may supply incorrect GET query parameters for:</p> <ol style="list-style-type: none"> type=tenday province=Camarines Sur

	<p>2. The collaborator may supply an invalid API key for the query parameter:</p> <ol style="list-style-type: none"> key=(YOUR_API_KEY) <p>3. Collaborators will receive a 403 or 500 error response if they supply invalid GET query parameters like described in the alternative scenario steps #1 or #2.</p> <p>4. The REST API endpoint hosted in Vercel may become inaccessible whenever Vercel experiences downtimes.</p>
Notes and Questions:	

UC-02: Special Weather Forecast API

Summary:	Enables Collaborators to fetch the latest active PAGASA special weather forecast from ACAP's database
Priority:	Desired
Use Frequency:	Rarely
Direct Actor/s:	Collaborator
Stakeholder/s:	Collaborator, System Administrator
Prerequisite/s:	1. Authorized API Keys (UC-12, UC-13)
Main Success Scenario:	<p>1. Create a (GET) HTTP request to the PAGASA Seasonal Weather Forecast API in the deployed API URL: https://acap-rcmas.vercel.app/api/weatherforecast</p> <p>2. Attach the following GET query parameters in the request:</p> <ol style="list-style-type: none"> type=special province=Camarines Sur key=(YOUR_API_KEY) <p>3. View the success 200 API response results.</p> <p><i>(Please view the PAGASA REST API usage documentation for more information at https://acap-rcmas.vercel.app/)</i></p>
Alternative Scenario:	<p>1. The collaborator may supply incorrect GET query parameters for:</p> <ol style="list-style-type: none"> type=special province=Camarines Sur <p>2. The collaborator may supply an invalid API key for the query parameter:</p> <ol style="list-style-type: none"> key=(YOUR_API_KEY) <p>3. Collaborators will receive a 403 or 500 error response if they supply invalid GET query parameters like described in the alternative scenario steps #1 or #2.</p> <p>4. The REST API endpoint hosted in Vercel may become inaccessible</p>

	whenever Vercel experiences downtimes.
Notes and Questions:	

UC-03: Historical Seasonal Weather Forecast API

Summary:	Enables Collaborators to fetch one or more past records of PAGASA seasonal weather forecast data from ACAP's database
Priority:	Desired
Use Frequency:	Optional
Direct Actor/s:	Collaborator
Stakeholder/s:	Collaborator, System Administrator
Prerequisite/s:	1. Authorized API Keys (UC-12, UC-13) 2. Historical seasonal weather forecast data management in UC-06, UC-09
Main Success Scenario:	<ol style="list-style-type: none"> Create a (GET) HTTP request to the PAGASA Historical Seasonal Weather Forecast API in the deployed API URL: https://acap-rcmas.vercel.app/api/weatherforecast/archives Attach the required GET query parameters in the request: <ol style="list-style-type: none"> type=seasonal province=Camarines Sur key=(YOUR_API_KEY) Attach the GET query parameters to the request to retrieve a set of (6) seasonal weather forecasts from a starting month and year. <ol style="list-style-type: none"> month_start={month code} year={4-digit year} Attach the GET query parameters to the request to retrieve a specified month's available seasonal forecasts concerning the month(s) of its upload: <ol style="list-style-type: none"> month={month code} year={4-digit year} View the success 200 API response results. <p><i>(Please view the PAGASA REST API usage documentation for more information at https://acap-rcmas.vercel.app/)</i></p>
Alternative Scenario:	1. The collaborator may supply incorrect GET query parameters for: <ol style="list-style-type: none"> type=seasonal province=Camarines Sur

	<ol style="list-style-type: none"> The collaborator may supply an invalid API key for the query parameter: <ol style="list-style-type: none"> key=(YOUR_API_KEY) Collaborators will receive a 403 or 500 error response if they supply invalid GET query parameters like described in the alternative scenario steps #1 or #2, or main success scenarios steps #3 or #4. The REST API endpoint hosted in Vercel may become inaccessible whenever Vercel experiences downtimes.
Notes and Questions:	

UC-04: Historical 10-Day Weather Forecast API

Summary:	Enables Collaborators to fetch one or more past records of PAGASA 10-day weather forecast data from ACAP's database
Priority:	Desired
Use Frequency:	Optional
Direct Actor/s:	Collaborator
Stakeholder/s:	Collaborator, System Administrator
Prerequisite/s:	1. Authorized API Keys (UC-12, UC-13) 2. Historical 10-day weather forecast data management in UC-07, UC-10
Main Success Scenario:	<ol style="list-style-type: none"> Create a (GET) HTTP request to the PAGASA 10-Day Historical Weather Forecast API in the deployed API URL: https://acap-rcmas.vercel.app/api/weatherforecast/archives Attach the required GET query parameters in the request: <ol style="list-style-type: none"> type=tenday province=Camarines Sur key=(YOUR_API_KEY) Attach the GET query parameter to the request to retrieve (1) one or more records of past 10-day weather forecast at a specified date(s): <ol style="list-style-type: none"> date_created={YYYY/MM/DD} date_created={comma-delimited values of YYYY/MM/DD} Attach the GET query parameter to the request to retrieve (1) one or more records of past 10-day weather forecast from a specified start date and end date:

	<p>a. date_created_range={YYYY/MM/DD D, YYYY/MM/DD}</p> <p>5. View the success 200 API response results.</p> <p><i>(Please view the PAGASA REST API usage documentation for more information at https://acap-rcmas.vercel.app/)</i></p>
Alternative Scenario:	<p>1. The collaborator may supply incorrect GET query parameters for:</p> <p>a. type=tenday</p> <p>b. province=Camarines Sur</p> <p>2. The collaborator may supply an invalid API key for the query parameter:</p> <p>a. key=(YOUR_API_KEY)</p> <p>3. Collaborators will receive a 403 or 500 error response if they supply invalid GET query parameters like described in the alternative scenario steps #1 or #2, or main success scenarios steps #3 or #4.</p> <p>4. The REST API endpoint hosted in Vercel may become inaccessible whenever Vercel experiences downtimes.</p>
Notes and Questions:	

UC-05: Historical Special Weather Forecast API

Summary:	Enables Collaborators to fetch one or more past records of PAGASA special weather forecast data from ACAP's database
Priority:	Desired
Use Frequency:	Optional
Direct Actor/s:	Collaborator
Stakeholder/s:	Collaborator, System Administrator
Prerequisite/s:	<p>1. Authorized API Keys (UC-12, UC-13)</p> <p>2. Historical 10-day weather forecast data management in UC-08, UC-11</p>
Main Success Scenario:	<p>1. Create a (GET) HTTP request to the PAGASA 10-Day Historical Weather Forecast API in the deployed API URL: https://acap-rcmas.vercel.app/api/weatherforecast/archives</p> <p>2. Attach the required GET query parameters in the request:</p> <p>a. type=special</p> <p>b. province=Camarines Sur</p> <p>c. key=(YOUR_API_KEY)</p> <p>3. Attach the GET query parameter to the request to retrieve (1) one or more</p>

	<p>records of past 10-day weather forecast at a specified date(s):</p> <p>a. date_created={YYYY/MM/DD}</p> <p>b. date_created={comma-delimited values of YYYY/MM/DD}</p> <p>4. Attach the GET query parameter to the request to retrieve (1) one or more records of past 10-day weather forecast from a specified start date and end date:</p> <p>a. date_created_range={YYYY/MM/DD D, YYYY/MM/DD}</p> <p>5. View the success 200 API response results.</p> <p><i>(Please view the PAGASA REST API usage documentation for more information at https://acap-rcmas.vercel.app/)</i></p>
Alternative Scenario:	<p>1. The collaborator may supply incorrect GET query parameters for:</p> <p>a. type=tenday</p> <p>b. province=Camarines Sur</p> <p>2. The collaborator may supply an invalid API key for the query parameter:</p> <p>a. key=(YOUR_API_KEY)</p> <p>3. Collaborators will receive a 403 or 500 error response if they supply invalid GET query parameters like described in the alternative scenario steps #1 or #2, or main success scenarios steps #3 or #4.</p> <p>4. The REST API endpoint hosted in Vercel may become inaccessible whenever Vercel experiences downtimes.</p>
Notes and Questions:	

UC-06: Archiving and storage of Historical Seasonal Weather Forecast

Summary:	Enables the system to store the latest seasonal weather forecast data into the historical database collection.
Priority:	Desired
Use Frequency:	Sometimes
Direct Actor/s:	System
Stakeholder/s:	Administrator
Prerequisite/s:	1. Latest active SEASONAL weather forecast data (see the ACAP 1.0 UC-21)
Main Success Scenario:	<p>1. The system fetches the "current" active SEASONAL weather forecast data from the database into temporary storage.</p> <p>2. The system's process for uploading and validation of an Administrator-</p>

	<p>uploaded PAGASA's seasonal weather forecast Excel file (ACAP 1.0 – UC-21) finishes without errors, overwriting the latest active seasonal weather forecast data.</p> <p>3. If step #2 finishes successfully, store the previous seasonal weather forecast data in the historical database collection.</p>
Alternative Scenario:	<p>1. If the main success scenario step #2 fails with validation or other errors:</p> <ol style="list-style-type: none"> Skip storing the previous seasonal weather forecast data in the historical database collection.
Notes and Questions:	

UC-07: Archiving and storage of Historical 10-Day Weather Forecast

Summary:	Enables the system in which the Cron jobs are running (or System Administrators) to store yesterday's 10-day weather forecast data into the historical database collection.
Priority:	Desired
Use Frequency:	Always (daily)
Direct Actor/s:	System (GitHub Actions cron)
Stakeholder/s:	System Administrator
Prerequisite/s:	1. Latest active 10-DAY weather forecast data (see the ACAP 1.0 UC-26, UC-64)
Main Success Scenario:	<ol style="list-style-type: none"> The system fetches the "current" active 10-DAY weather forecast data from the database and stores it in the historical database collection by its "date_created" field. The system's Cron job for fetching, validating, and uploading PAGASA's latest 10-day weather forecast Excel files (ACAP 1.0 – UC-64) finishes download and validation and uploads without errors, overwriting the latest active 10-day weather forecast data.
Alternative Scenario:	<ol style="list-style-type: none"> If the main success scenario step #1 fails with validation or other errors: <ol style="list-style-type: none"> Continue fetching, validation, and uploading the latest PAGASA 10-day weather forecast Excel files in the main success scenario step #2. A System Administrator initiates running the cron job from GitHub Actions.

	3. A System Administrator initiates running the cron job from their development machine.
Notes and Questions:	

UC-08: Archiving and storage of Historical Special Weather Forecast

Summary:	Enables the system in which the Cron jobs are running (or System Administrators) to store the last active special weather forecast data into the historical database collection.
Priority:	Desired
Use Frequency:	Always (daily every 2 hours)
Direct Actor/s:	System (GitHub Actions cron)
Stakeholder/s:	System Administrator
Prerequisite/s:	1. Latest active SPECIAL weather forecast with typhoon data (see the ACAP 1.0 UC-28, UC-66)
Main Success Scenario:	<ol style="list-style-type: none"> The system fetches the "current" active SPECIAL weather forecast data from the database and stores it in the historical database collection by its "typhoon_name," "year," and "bulletin_no" fields. <ol style="list-style-type: none"> If historical data containing the same "typhoon_name," "year," and "bulletin_no" fields do not yet exist, the script proceeds to store the active severe cyclone weather forecast data in the historical collection. The system's Cron job for fetching (web-scraping), validating, and uploading the latest tropical cyclone information from PAGASA's Tropical Cyclone web page (ACAP 1.0 – UC-66) finishes download and validation and uploads without errors, overwriting the latest active severe cyclone weather forecast data.
Alternative Scenario:	<ol style="list-style-type: none"> If the main success scenario step #1 fails with validation or other errors: <ol style="list-style-type: none"> Continue fetching, validation, and uploading the latest PAGASA 10-day weather forecast Excel files in the main success scenario step #2. If the main success scenario step #1 also finds existing records of similar "typhoon_name," "year," and "bulletin_no" fields in the historical database collection: <ol style="list-style-type: none"> Continue fetching, validation, and uploading the latest PAGASA 10-

	<p>day weather forecast Excel files in the main success scenario step #2.</p> <ol style="list-style-type: none"> 2. A System Administrator initiates running the cron job from GitHub Actions. 3. A System Administrator initiates running the cron job from their development machine.
Notes and Questions:	

UC-09: Cron job – Auto delete outdated Historical Seasonal Weather Forecast

Summary:	Enables the system in which the Cron jobs are running (or System Administrators) to delete outdated seasonal weather forecast data from the historical database collection.
Priority:	Desired
Use Frequency:	Sometimes
Direct Actor/s:	System (GitHub Actions cron)
Stakeholder/s:	System Administrator
Prerequisite/s:	1. Outdated historical SEASONAL weather forecast data – older than (6) six months (UC-06)
Main Success Scenario:	<p>There are (2) two options for deleting outdated historical seasonal weather forecast data.</p> <p>SYSTEM (GitHub Actions cron)</p> <ol style="list-style-type: none"> 1. The system checks the historical database collection for outdated seasonal weather forecast data (older than (6) six months) every 1st day of the month at around midnight (Philippine time). 2. The system deletes outdated historical seasonal weather forecast data if it finds any. <p>SYSTEM ADMINISTRATOR</p> <ol style="list-style-type: none"> 1. System Administrators can initiate checking and deleting outdated seasonal weather forecast data by: <ol style="list-style-type: none"> a. Running the system cron script from GitHub Actions. b. Running the cron script from their development machines.
Alternative Scenario:	1. If the main success scenario step #1 fails with validation or other errors. Or does not find outdated seasonal weather forecast data:

	a. Nothing is changed or modified in the historical weather forecast collection.
Notes and Questions:	

UC-10: Cron job – Auto delete outdated Historical 10-Day Weather Forecast

Summary:	Enables the system in which the Cron jobs are running (or System Administrators) to delete outdated 10-day weather forecast data from the historical database collection.
Priority:	Desired
Use Frequency:	Always (daily)
Direct Actor/s:	System (GitHub Actions cron)
Stakeholder/s:	System Administrator
Prerequisite/s:	1. Outdated historical 10-DAY weather forecast data – older than (3) three months (UC-07)
Main Success Scenario:	<p>There are (2) two options for deleting outdated historical 10-day weather forecast data.</p> <p>SYSTEM (GitHub Actions cron)</p> <ol style="list-style-type: none"> 1. The system checks the historical database collection for outdated 10-day weather forecast data (older than (3) three months) every day between 9:00 AM – 12:00 PM after running the Cron job for fetching, parsing, and uploading the latest PAGASA 10-day weather forecast Excel files (see ACAP 1.0 – UC-64 for more information). 2. The system deletes outdated historical 10-day weather forecast data if it finds any from step #1. <p>SYSTEM ADMINISTRATOR</p> <ol style="list-style-type: none"> 1. System Administrators can initiate checking and deleting outdated seasonal weather forecast data by: <ol style="list-style-type: none"> a. Running the system cron script from GitHub Actions. b. Running the cron script from their development machines.
Alternative Scenario:	1. If the main success scenario step #1 fails with validation or other errors. Or does not find outdated 10-day weather forecast data: <ol style="list-style-type: none"> a. Nothing is changed or modified in the historical weather forecast collection.

Notes and Questions:	
-----------------------------	--

UC-11: Cron job – Auto delete outdated Historical Special Weather Forecast

Summary:	Enables the system in which the Cron jobs are running (or System Administrators) to delete outdated special weather forecast data from the historical database collection.
Priority:	Desired
Use Frequency:	Always (every 2 hours)
Direct Actor/s:	System (GitHub Actions cron)
Stakeholder/s:	System Administrator
Prerequisite/s:	1. Outdated historical SPECIAL weather forecast data – older than (3) three months (UC-08)
Main Success Scenario:	<p>There are (2) two options for deleting outdated historical SPECIAL weather forecast data.</p> <p>SYSTEM (GitHub Actions cron)</p> <ol style="list-style-type: none"> 1. The system checks the historical database collection for outdated special weather forecast data (older than (3) three months) every (2) two hours after running the Cron job for web-scraping, parsing, and uploading the latest PAGASA severe cyclone (special) weather forecast data from their Tropical Cyclone webpage (see ACAP 1.0 – UC-66 for more information). 2. The system deletes outdated historical special weather forecast data if it finds any from step #1. <p>SYSTEM ADMINISTRATOR</p> <ol style="list-style-type: none"> 1. System Administrators can initiate checking and deleting outdated seasonal weather forecast data by: <ol style="list-style-type: none"> a. Running the system cron script from GitHub Actions. b. Running the cron script from their development machines.
Alternative Scenario:	<ol style="list-style-type: none"> 1. If the main success scenario step #1 fails with validation or other errors. Or does not find outdated special weather forecast data: <ol style="list-style-type: none"> a. Nothing is changed or modified in the historical weather forecast collection.

Notes and Questions:	
-----------------------------	--

UC-12: Create API Keys

Summary:	Enables System Administrators to create API Keys
Priority:	Desired
Use Frequency:	Rarely
Direct Actor/s:	System Administrator
Stakeholder/s:	System Administrator, Collaborator
Prerequisite/s:	
Main Success Scenario:	<p>(Note: There are no streamlined processes or guidelines for creating API keys for now).</p> <ol style="list-style-type: none"> 1. The System Administrator creates API keys comprising of random characters by: <ol style="list-style-type: none"> a. Using 3rd party tools or software. b. Typing random text from the keyboard. 2. After step #1, the System Administrator saves the generated API key into a text file or Google Drive for safe-keeping.
Alternative Scenario:	
Notes and Questions:	

UC-13: Authorize API Keys

Summary:	Enables System Administrators to allow REST API access to Collaborators with valid API keys
Priority:	Desired
Use Frequency:	Rarely
Direct Actor/s:	System Administrator
Stakeholder/s:	System Administrator, Collaborator
Prerequisite/s:	API keys (UC-12)
Main Success Scenario:	<ol style="list-style-type: none"> 1. The System Administrator successfully generates an API key (from UC-12). 2. The System Administrator includes the API key into the system's list of valid API keys and restarts the system. 3. Collaborators making HTTP requests to the PAGASA REST API endpoints using valid API keys should receive expected response data after step #2.
Alternative Scenario:	
Notes and Questions:	