

Software Requirements Specification

Project Information

Project: Agro-Climatic Advisory Portal - Bicol (ACAP-BICOL) / ACAP 1.0

Release Number: 4.0

Date Updated: September 6, 2023

Attached worksheets:

- ACAP 1.0 -- Use Case Suite v4.0
(*ACAP_1.0_Use_Case_Suite_4.0.pdf*)
- ACAP 1.0 -- Feature Set v4.0
(*ACAP_1.0_Feature_Set_4.0.pdf*) – ON-GOING DRAFT

Introduction

The Agro-Climatic Advisory Portal Bicol (ACAP-BICOL) is an online web application for disseminating weather and climate information of the Adaptation and Mitigation Initiative in Agriculture (AMIA) Villages in the Bicol Region. It follows a hybrid type of three-tier web application architecture consisting of a website for the client layer, the Firestore database for the database layer, REST APIs running on a NodeJS server, and several client-side logic for the business layer.

The portal aims to guide extension workers in creating and disseminating relevant and tailored advisories and recommendations for farmers and fisherfolk that will help them address the impacts of climate change thru the use of its online-accessible tools for linking smart automatic bulletins PDF creation linking with PAGASA weather forecast data, the centralized process of making information available for general viewing or download and by relaying PAGASA weather-related data for laymanized public viewing in a readily-accessible and timely manner.

This document discusses ACAP's high-level components, use cases, and features.

Use Cases

The use cases are organized into four categories with subcategories.

I. Use Cases by Functional Area

ACAP divides use cases according to their functional specification, identified into seventeen (17) subcategories:

- Login and user authentication component, which has the use cases for user login authentication;
- Accounts management component, which has use cases for administering the ACAP admin users;
- PDF bulletins generator component, which has use cases for drafting, rendering, and publishing crop recommendations PDF bulletins;
- Public web page components, comprising of several publicly-accessible and viewable pages which has use cases for displaying and visualizing the Bicol CIS data in public-viewable web pages such as the:
 - Home page component, which has use cases for visualizing the information of the Bicol AMIA villages in an interactive web map
 - ACAP services page component, which has use cases for displaying the PAGASA 10-day, seasonal, and severe weather forecast data
 - Cropping calendar component, which has use cases for displaying interactive crop calendar data on-site and allowing the PDF bulletins generator component to use cropping calendar data as a reference
 - Public seasonal recommendations generator, which has uses cases for generating seasonal crop recommendations in English and Tagalog languages for public viewing preview
 - Public 10-day farm outlook and weather advisory generator, which has use cases for generating 10-day farm outlook and weather advisory crop recommendations in English and Tagalog for public viewing preview
 - Bulletins (PDF downloads) component, which has requirements for organizing and displaying generated PDFs on-site for download
 - Site Search, which has use cases for allowing public website viewers to search for keywords from the website and list URL links to matching results
- Admin (private) web page components, comprising of privileged admin access to web pages for managing the ACAP website, such as the:
 - Crop recommendations management component, which has use cases for creating, previewing, and deleting admin-created seasonal, 10-day, and severe weather farm outlook and weather advisory crop recommendations PDF bulletins and reports
 - Seasonal weather forecast component, which has use cases for administering the Seasonal Weather data by uploading PAGASA's seasonal weather forecast Excel file and displaying the uploaded data on-site
 - Phonebook component, which has use cases for managing contacts in a phonebook
 - SMS management component, which has use cases of previewing the crop recommendation, sending SMS logs, and lastly, initiating sending the SMS text
 - ACAP settings component, which has use cases for manually updating the site-wide seasonal weather forecast, 10-day weather forecast, and severe weather advisory data thru manual encoded data input concerning PAGASA data and

semi-automatic site data updates via Excel files upload for other relevant weather data that are not yet possible to have automatic updates

- User Profile management component, which has use cases for allowing an Administrator to update their password
- Cron Jobs for Automatic Weather Forecast Data Syncing to ACAP's Database, which has use cases for regularly and automatically syncing PASAGA's 10-day weather forecast Excel files, El Nino/La Nina page, and Severe Tropical Cyclone Bulletins page contents to ACAP's database on scheduled time intervals.
- Representational State Transfer Application Programming Interface (REST API) Endpoints, which has use cases for allowing signed-in admins to mutate ACAP data over HTTPS for: bulletins, phonebook, reports, users, and PAGASA weather forecast (seasonal, 10-day, and special), sending SMS, and for generating bulletin PDFs
- Data Processing and Upload Scripts, which has use cases for cleaning, pre-processing, formatting, and uploading raw data to the Firestore database; and for seeding the Firestore database with default data

II. Use Cases by Stakeholder

- a. Four (4) types of stakeholders can access or interact with the website portal: the public viewer, administrator, super administrator, and the system administrator. Use cases are divided according to the stakeholders' roles and key needs.

III. Use Cases by Priority

- A. Use cases are partitioned into three priorities; essential, expected and desired.

IV. Use Cases by Business Object and Actors

- A. There are eleven (11) business objects defined that interact with the actors; user records, crop recommendations records, weather forecast records, cropping calendar records, recommendations report logs, (site-wide) PDF bulletins, contacts, SMS, public-viewable website pages, administrator-accessible web pages, and super-administrator accessible web pages.

Details

Actors are described as follows:

- All
 - Stakeholders that can view and access the public-viewable website pages over the network
 - Key Needs:
 - Convenient access to the website's content over the network

- Public Viewer

- A regular web surfer not signed in on the website as Administrators or Super Administrators.
- These stakeholders can view and make assessments on ACAP's public website pages and are restricted access to the ACAP login pages.
- Key Needs:
 - Convenient access to the website's content over the network

- Administrator

- These stakeholders can create reports and PDF bulletins while referencing the site's current weather forecast data. Administrators can send SMS text versions of generated reports to their contacts on ACAP's SMS Management page.
- Key Needs:
 - Account security
 - Convenient automatic access to the site-wide weather data, incorporated into the PDF bulletin and report
 - Add, view, and delete own seasonal and 10-day weather farm outlook crop recommendations report logs
 - Generate and automatically upload site-wide crop PDF bulletins for public download from own reports
 - Create new contacts in own Phonebook
 - Send the SMS text version of reports to their Contacts list
 - Update and view the site-wide weather data (seasonal rainfall forecast, 10-day weather forecast, and special weather forecast)

- Super Administrator

- Super administrators have the responsibility for the housekeeping of Administrator users
- Key Needs:
 - Account security

- Create, view, update, delete and restrict access to Administrator user accounts
- System Administrator
 - System administrators are responsible for initializing and configuring ACAP to work with their region. They will also be responsible for deploying the frontend and backend applications to the recommended cloud services, other Platform-as-a-Service (PaaS), or their internal infrastructure.
 - System administrators are ideally highly-experienced software developers familiar with ACAP's overall architecture.
 - Key needs:
 - Account security
 - Convenient network access, especially during deployment and troubleshooting logs from the cloud services dashboard

Functional Requirements

The software provides the same set of use cases and features. Super Administrators can create Administrator users who can log in to the site and update or delete Admin user accounts. Administrators can generate crop recommendations, which trigger the creation of site-wide final PDF bulletins. These bulletin PDFs are then made available on the Bulletins downloads web page. Administrators can also manually update the site's weather data and send SMS texts of generated crop recommendations (currently containing placeholder text) to Contacts in their Phone Books. Regular public site viewers can view weather data on the public-accessible weather services web page and view partial-complete seasonal and 10-day crop recommendations on the Crop recommendations page.

Non-Functional Requirements

Usability requirements

- The system must ensure user-friendly use case steps to help users use the system. It will provide step-by-step procedures for each use case and feature.

- The user interfaces should be as familiar as possible to users who have used other similar climate services web portals. E.g., we will follow the UI guidelines for naming menus, buttons, dialog boxes, and UI layout whenever possible.
- The front-end website should follow a responsive web design layout and patterns. Users should be able to view the website in a user-friendly manner on most major desktop and mobile web browsers and best viewed on the Google Chrome web browser.
- The front-end website should load fast. It should not lag, freeze, or become unresponsive while loading or when users view or use the various web page features.
- The 1-pager bulletin PDFs should have selectable and searchable text content and not just be made up of an image/picture file. The bulletin PDF file size should be around 500 KB – 1 MB (max) to allow network-friendly downloads for viewing or printing. It should only have at most (1) one page, regardless of text content length.
- Users should see the latest PAGASA 10-day, seasonal, or severe cyclone weather forecast from the ACAP Services web page. Since we are currently unsure of when PAGASA exactly updates its weather forecast data from its side (in exact hours), ACAP should display the date and time when it last synced the weather forecast data with PAGASA (either through Cron jobs or manual update by Administrators) for reference based on observations.
- Users should have a strong network connection, ranging from 4G to Wifi, to have a smoother user experience viewing and loading data from the website.

Reliability and up-time requirements

*The system should be able to host a moderate amount of Administrator users (~50+) and medium size of weather and user/site data not exceeding 1GB at a stable phase.

The server should have a 99.9% up-time with at most 45 minutes of downtime. We expect the website, server, and ACAP services to be accessible, especially if severe cyclones are in the Philippines.

Safety requirements

ACAP utilizes stable technologies to build the system. The application, currently deployed on the standard plan (free-tier) cloud services, expects minimum human safety hazards during the maintenance and operation of the system since cloud-based applications allow flexible access to its components over the cloud.

Security requirements

- The website prompts users to input their super admin-provided username and password to control Admin site access.
- Only Super Administrators can create authentic new users and edit or delete existing Administrator accounts.
- Only Administrators can configure the site's weather data, create crop recommendations PDF bulletins and reports, and send SMS text versions of generated recommendations to contacts.
- Sensitive data like API keys, phone numbers, emails, and user-generated reports should be securely stored and encrypted. Secure Firestore Security Rules should be observed to prevent leaking of this sensitive information to the project's public Firebase REST API.
- The frontend website should not be able to perform "write" operations to the Firestore database or Storage. Write access to the Firestore database should only be through secure, authenticated HTTP requests to the NodeJS backend REST API.
- The REST API should reject HTTP requests that do not have or have invalid Bearer Authorization tokens from the front end. Signing into the ACAP admin pages automatically generates fresh, valid Bearer Authorization tokens using the Firebase Authentication service.
- HTTP requests accessing the NodeJS REST APIs should have minimal to thorough validation.
- Operations requiring legitimate data, like generating bulletin PDFs, should retrieve (crop recommendations, cropping calendar, and weather forecast) data from the Firestore database and not from user input.
- We will observe various software exploits to avoid and ensure the reliability and security of the site, for example, XSS, etc.
Details:
 - Passwords will be 4-16 characters long and alphanumeric.
 - We will use encrypted communications (SSL) for this website.

Performance and scalability requirements

The system should be able to perform well, given Firebase's generous Spark plan (free) pricing tier. The website should be accessible on GitHub Pages, and the Individual (free tier) plan Render account for the NodeJS backend server should respond promptly to HTTP requests. Firebase Authentication can support 100+ user accounts, while a pay-as-you-go pricing subscription for the Firestore database and Cloud Firestore Storage may be required if site data exceeds Firebase's Spark plan limits.

The NodeJS server backend, where ACAP's other core server components reside (currently hosted on an Individual plan (free-tier) Render server), can be flexibly moved to more powerful bare metal OS servers or similar Platform-as-a-Service (PaaS) cloud services.

Minimal database and server optimizations will be done during the maintenance and support of the system if needed but require expert reviews regarding overall server networking and security, especially if the system moves to a bare metal OS server. Dedicated expert advice on server/database management is in place should a decision for migrating ACAP to use local databases (remove Firebase and Render cloud services) be made.

ACAP's infrastructure, comprised of several cloud services (Firebase, Render, GitHub) currently using standard (free-tier) pricing plans, can scale pretty well by upgrading to paid plans.

Maintainability and upgradability requirements

Maintainability is our ability to make changes to the product over time. We will address this by anticipating several types of change and by carefully documenting our design and implementation.

The system has plans for future releases, which will happen when new system features are requested. It will require updating the initial design and architecture of the current stable running system and implementing the code changes.

Should a future release be requested, we must be able to deploy and upgrade with minimal downtime and service disruption.

Business life-cycle requirements

The business life-cycle of a product includes everything that happens to that product over several years, from the initial release, through important but infrequent use cases, until product retirement. The product's main life-cycle requirements are listed below.

Details:

- The product should be stable. It should use systems and software architectural design patterns for web applications that will live until the technologies and tools used to build it are considered obsolete. It will be scheduled for updates until an upgrade is critically needed.

System hardware requirements

ACAP relies on the standard plan (free-tier) hardware specifications of the following cloud services discussed in more detail in the *Cloud Infrastructure References Pricing Estimates* table. These services are limited but upgradable to paid subscription plans when deemed necessary.

- **GitHub** – for storing project source codes and revision tracking
- **GitHub Pages** – for website UI hosting
- **GitHub Actions** – for deploying to development and production environments and for running Cron jobs
- **Firebase**
 - Firebase Authentication – User authentication, registration, login
 - Firebase Cloud Storage – for storing and hosting PDF files
 - Firestore Database – NoSQL database for storing user and site data
- **Render** – Web Services for hosting a NodeJS backend with custom-created API endpoints for managing miscellaneous background processes.

Cloud Infrastructure References Pricing Estimates

	Standard Plan	Upgraded Plan	Recommendations
GitHub	Price: (none) Unlimited public/private repositories Unlimited GitHub Pages website hosting Size limit: 1GB (each published sites and source repositories) Bandwidth limit: 100GB/month (soft bandwidth limit) 2,000 CI/CD minutes/month	Price: \$44 per user/year (Team Pricing Plan) Includes everything in Standard Plan plus: - 3,000 CI/CD minutes/month - Protected branches - Environment deployment branches and secrets More information on https://github.com/pricing	Standard Plan is ok and more than enough to use
Render (please also consider the Heroku alternative, below)	FREE Price: \$0 per account/month Web Services - RAM: 512MB	STARTER Price: \$7 per instance/month Web Services: - RAM: 512GB	The FREE plan is alright to use with minor setbacks: <i>creating bulletins take more than 1-2 minutes if no one previously created</i>

	<ul style="list-style-type: none"> - CPU: Shared - Running hours/mo: 750 hrs - Bandwidth: 100 GB - Always on: Sleeps after 15 min of inactivity, otherwise always on depending on your remaining monthly free dyno hours. - Build Minutes/mo: 500 min 	<ul style="list-style-type: none"> - CPU: 0.5 - Running hours/mo: unlimited - Bandwidth: 500GB - Always on: Does not sleep after 15 min of inactivity <p>STANDARD Price: \$25 per instance/month</p> <p>Web Services</p> <ul style="list-style-type: none"> - RAM: 2GB - CPU: 1 - Running hours/mo: unlimited - Bandwidth: 500GB - Always on: Does not sleep after 15 min of inactivity <p>More information on https://render.com/pricing#compute (see the Services section)</p>	<p><i>a bulletin in the past 15 minutes</i>. The service will become unusable if it exceeds the default 750 hours total running time per month, but this is unlikely to happen since only 1 or 3 admins will create bulletins or send SMS, which needs the web service.</p> <p>The minimal upgraded STARTER plan at \$7/mo is enough to ensure that the web service is always turned on (does not sleep), have an unlimited running hours per month and a slightly higher RAM and CPU for processing power, most used when creating PDF bulletins.</p> <p>The upgraded STANDARD at plan at \$25/mo is most ok to use if the the system will require more RAM and CPU to speed up it's internal processes.</p> <p>Running the backend on Render's double containers looks complex. <i>We recommend using Heroku for simplicity and less strenuous CPU runtime</i> (see the Heroku pricing estimates below)</p>
Firestore Database (Firebase)	<p>Price: \$0/month (No-cost Spark Plan)</p> <p>Stored data: 1 GB</p>	<p>Price: \$0.18/month per excess of 1 GB (Blaze Plan)</p> <p>Stored data (minimum): 1 GB Stored data (maximum): 2 GB</p> <p>More information on https://firebase.google.com/pricing</p>	<p>The No-cost Spark plan looks sufficient to store and manage ACAP's raw data.</p> <p>The maximum 2 GB data storage offered by the Blaze plan looks like a good alternative for storing data from regions</p>

			with lots of municipalities.
Firestore Authentication (Firestore)	Price: \$0/month (No-cost Spark Plan) 50k monthly active users	Price: (Blaze plan, none) The Blaze plan's feature that ACAP uses is already included in the Spark plan.	Standard Plan
Firestore Storage (Firestore)	Price: \$0/month (No-cost Spark Plan) GB stored: 5GB / day GB downloaded: 1 GB / day	Price: \$0.026 per GB in excess of 5 GB (Blaze Plan) Minimum stored GB: 5 GB (price covered in Spark Plan) Minimum stored GB: 10 GB (price: \$0.13) More information on https://firebase.google.com/pricing	The standard plan is ok to use for storing around ~1 MB bulletin PDFs but the Blaze plan 10 GB storage looks like a good alternative for buffer size.
Semaphore SMS Service	Price: (n/a)	Pricing: 1,000 credits: P560 5,000 credits: P2,800.00 10,000 credits: P5,600.00 25,000 credits: P14,000.00 More information on https://semaphore.co/	Each credit is worth 1 SMS text containing 160 characters. The 1,000 credits for P560 looks sufficient enough to use per region per year. The 5,000 credits for P2,800.00 look like a good buffer if region admins are expecting to send SMS to 100+ users regularly.

*Note: Subscribing a Firebase service (i.e., Firestore database) to Firebase Blaze Plan will automatically subscribe all Firebase services to their designated Blaze Plans pricing

	Standard Plan	Upgraded Plan	Recommendations
Heroku	Price: (n/a)	Plan: Eco Price: \$5 for 1000 hours per month RAM: 512 MB Number of process types: 2 Always on: Sleeps after 30 mins of inactivity, otherwise always on depending on your remaining monthly Eco dyno hours.	ACAP initially used Heroku's standard plan (Hobby Plan), which was dissolved last September 2022 after Heroku stopped offering standard plans. Their standard plan became the current Eco plan. We recommend using Heroku

		<p>Plan: Basic Price: \$0.01 per hour (max \$7 per month) RAM: 512 MB Number of process types: 10 Always on: Does not sleep after 30 mins of inactivity, otherwise always on depending on your remaining monthly Eco dyno hours.</p> <p>Plan: Standard 1X Price: \$0.03 per hour (max \$25 per month) RAM: 512 MB Number of process types: unlimited Always on: Does not sleep after 30 mins of inactivity, otherwise always on depending on your remaining monthly Eco dyno hours.</p> <p>More information on https://www.heroku.com/pricing https://www.heroku.com/dynos</p> <p>*Note: A detailed comparison of the (2) Heroku Upgraded Plans (Basic and Standard 1X) is available on link for more information.</p>	<p>rather than Render.com (if possible) because installing the OS-level packages needed by ACAP's PDF renderer does not result in complex backend running times, which might lead to slow performance.</p> <p>We recommend subscribing to the Eco plan, or to the Basic plan.</p>
--	--	--	---

Cloud Infrastructure Standard Pricing Plan (viewing by Frontend and Backend)

	Website Frontend	NodeJS Backend	Firebase Project (includes Firestore Database, Firebase Authentication and Firebase Storage)
Hosting	GitHub Pages https://amia-cis.github.io	Render (Individual plan) https://amia-cis.onrender.com/	Google Firebase
Pricing References	https://github.com/pricing#compare-features https://docs.github.com/en/pages/getting-started-with-	https://render.com/docs/free#free-web-services	https://firebase.google.com/pricing

	github-pages/about-github-pages		
Standard (Free) tier Pricing Plan features	<p>Pages and Wikis are always available for public repositories.</p> <p>Size limit: 1GB (published sites and source repositories)</p> <p>Bandwidth limit: 100GB/month (soft bandwidth limit)</p>	<p>RAM: 512MB</p> <p>Running hours/mo: 750 hrs</p> <p>Bandwidth: 100 GB</p> <p>Always on: Sleeps after 15 min of inactivity, otherwise always on depending on your remaining monthly free dyno hours.</p>	<p>Firestore Database Email/Password usage: unlimited</p> <p>Firestore Database Stored Data: 1GB Network egress (usage): 10GB/month Document writes: 20K writes/day Document reads: 50K reads/day Document deletes: 20K deletes/day</p> <p>Firestore Storage GB Stored: 5GB GB Downloaded: 1GB/day Upload Operations: 20K/day Download Operations: 20K/day</p>

Developer/Programmer Laptop Hardware Specifications for ACAP Software Development

OS: Windows 11 (prefer) or Windows 10, Mac/Linux or any OS that can run NodeJS

RAM: 8 GB - 16 GB

Processor: Intel/AMD/Ryzen (if Windows)

Graphics: (any)

Storage: @least 512GB SSD + and HDD (optional)

Display: FHD 1920x1080 IPS (prefer)

System software requirements

The web server must have the following:

- A web server that can host static HTML, CSS, and JavaScript for hosting the website's UI.

The backend server cloud hosting, usually a Platform-as-a-Service (PaaS), must be capable of running the following:

- A NodeJS web server for running background processes and hosting ACAP's APIs.
- NPM scripts as Cron jobs in case GitHub Actions for running the Cron jobs used for automatic weather forecast data syncing to the database are unavailable

The database, storage, and user authentication uses a Firebase spark plan project.

*Please view the **Cloud Infrastructure References Pricing Estimates** table for more details

Application program interfaces (APIs)

The frontend website uses the Firebase Web SDK and standard JavaScript APIs for NextJS (React) for building its UI. These are open-source and can easily be acquired. NodeJS, also a JavaScript runtime for servers, is used to create and run the backend REST APIs while using the Firebase Admin SDK for NodeJS.

The custom REST APIs are accessible over a secure SSL connection and require a signed-in admin's authentication.

The Firebase APIs are accessible as the client (website or NodeJS backend) uses the correct Firebase-generated API keys and service account JSONs.

Data import and export requirements

The system will store all data in a Firestore NoSQL database, where other trusted client programs over the web can access it with secure authentication or authorization. Custom NPM scripts run through the command line will facilitate raw data pre-processing, cleaning, and uploading to the Firestore database.