SALN Digitalization App

Lucas Miguel V. Agcaoili, Aeron Dann P. Peñaflorida, Miguel A. Guiang

LAMig

Client: Associate Prof. Rommel Feria

Group Blog: https://shirokamiqq.github.io/cs191-LAMig-GroupBlog/

Course Details: CS 191 WFR1 2526A - Associate Prof. Ligaya Leah Figueroa

Customer Statement of Requirements

I. Background and Motivation

The SALN (Statements of Assets, Liabilities and Net Worth) is a required document that government officials must submit annually. It serves as a declaration of their assets (what they own), liabilities (what they owe), and total net worth, while also disclosing any financial connections or business interests. Given this, the SALN promotes transparency, accountability, and integrity in the public office, which serves as a safeguard against corruption. Currently, government officials still have to go through a tedious process of manually filling out forms either through handwriting or electronically using spreadsheets. Thus, there is a need for an application to make filling up SALNs more efficient.

II. Proposed Solution / Project Idea

The proposed solution is a SALN Progressive Web App that aims to streamline the generation of the SALN by allowing users to input their information through either completing a form digitally through the app or uploading a JSON. The app will then generate the input into a properly formatted PDF file. Since it is a progressive web app, it means that the web app must be easy to access, installable, available offline, background-capable, and responsive to any device on which it is installed. Additionally, the inputted information will only be temporarily stored, after which it will be deleted. This ensures user privacy when filling out sensitive information.

III. Users/Stakeholders

a. **Practitioners** - The practitioners of this software project will be Migz, Lucas, and Dann, who are primarily responsible for carrying out the development tasks, software project paperwork, and communication with the client.

b. **Customers -** The main customer of this software project is Prof. Rommel Feria, who will provide the project requirements, set expectations for deliverables, and give feedback to the project at every scheduled meeting.

c. **End Users -** The end users for this project are government employees who need to file their SALN.

IV. Requirements/Features

To register for an account, users will be asked to provide an email address. This will be used for Two-Factor Authentication during registration and for every login instance. After the user provides their email, they will be sent 4 random words as OTP, which they need to input to verify their account.

Once the user logs in, they will be taken to the dashboard with two buttons: one that brings them to the SALN page, and one that lets them view a saved SALN form that they filled out within the past 5 days.

Once the user goes to the SALN page, they can either fill out a form on the site or upload a JSON file with all the information they need to fill out the form. They can upload their signature to the site or sign it using their mouse or touchscreen.

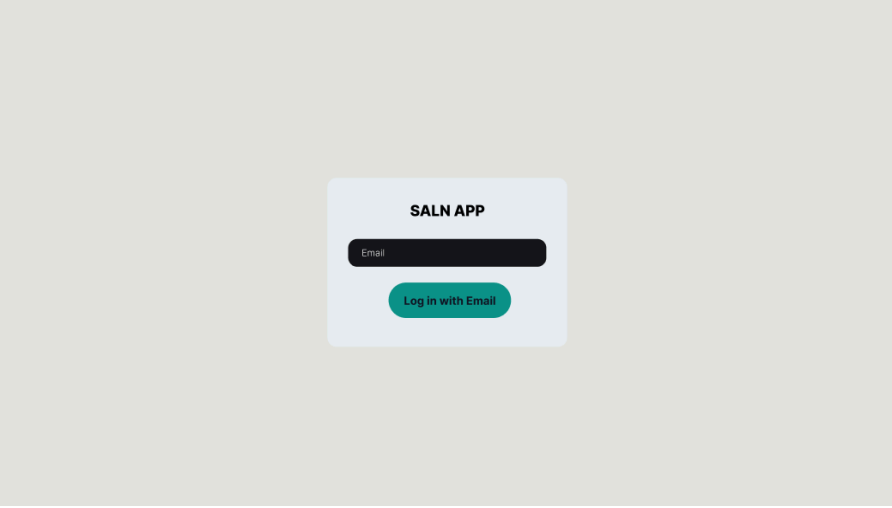
Upon completion of a SALN form, the user may generate a PDF of the completed form with the information they filled in. They then have the option to download or save the information they uploaded.

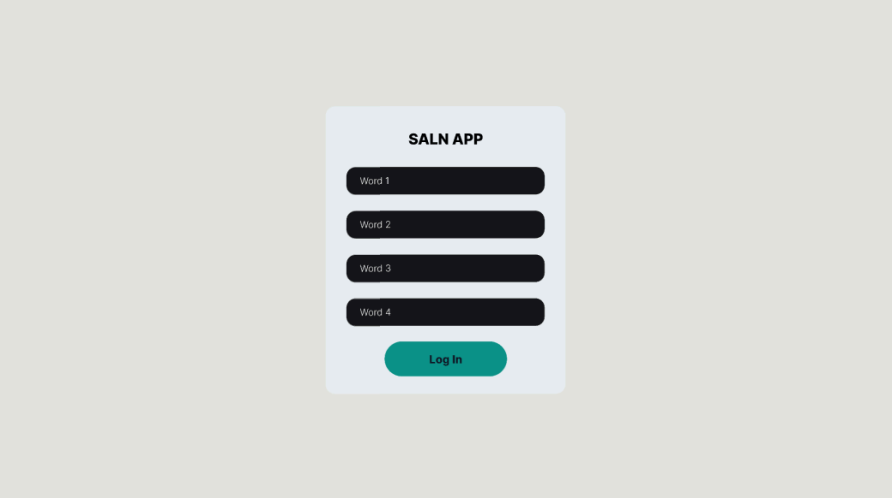
The app also allows the user to delete their account or remove any SALN form data. Lastly, SALN data older than 5 days will be automatically deleted from the database.

In essence, the app will be a progressive web application that allows users to create accounts to track the status of their SALN. Once the user requests to fill out their SALN, they will be able to access a digitalized interface similar to the SALN. There will be automatic computation for total assets and liabilities based on the inputs from the user, and a dropdown menu to allow them to add as many family members or boxes as necessary. Information gathered by the app will be stored and encrypted in a secure database.

V. Preliminary UI Design

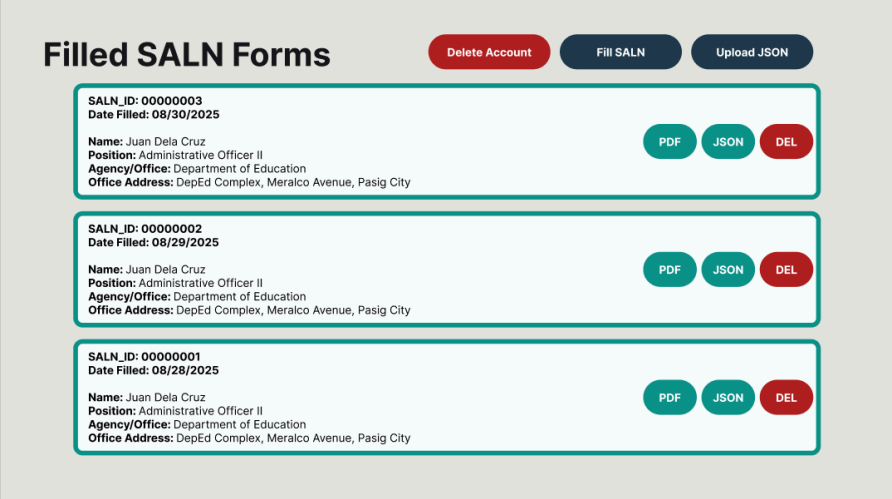
Upon opening the app, the user will start at the registration/login page that asks for an email address. Once the user enters an email, they will also have to enter the four random words sent to that email as a verification step.



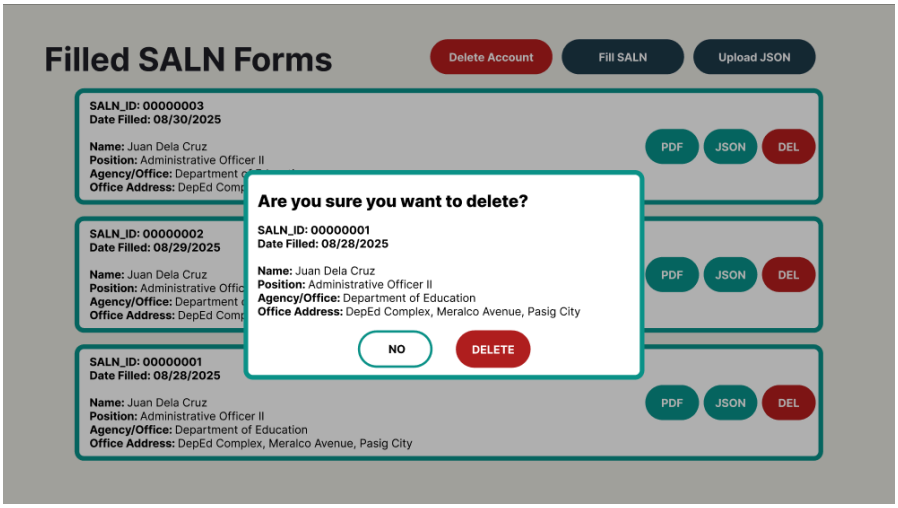


Upon logging into the app, the user will see a dashboard listing all the SALN forms he has completed in the past 5 days. For each of the user's filled SALN forms, there would be

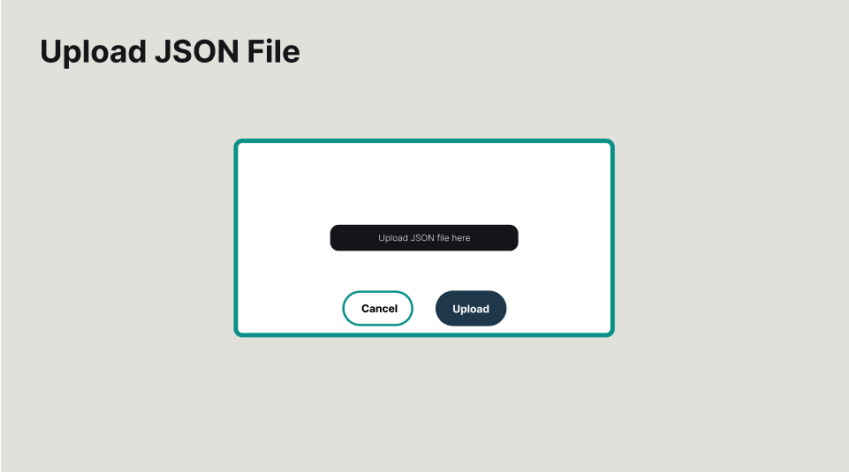
buttons to delete it, to generate a PDF version of it, and to export its data in a JSON file. Also, the user would see buttons to fill up a new SALN form, upload a JSON file containing SALN form data, and to delete his account along with its associated SALN data.



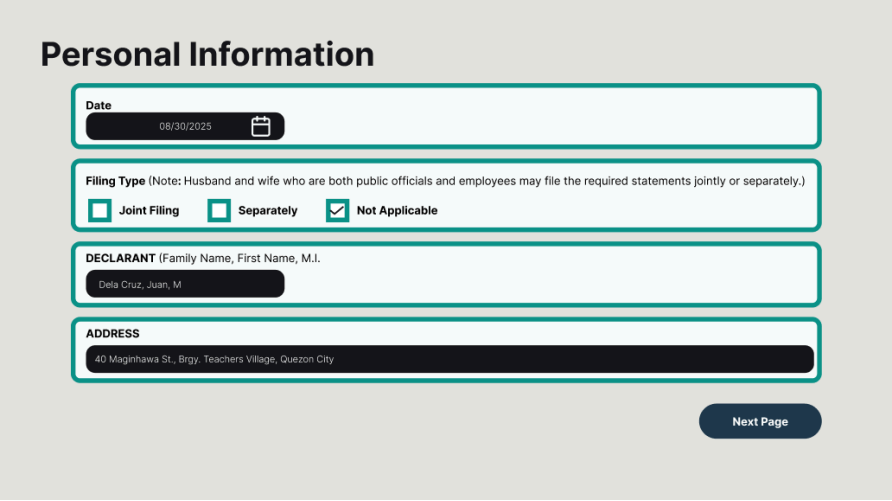
Should the user choose to delete his account or to delete a filled SALN form, he would see an alert asking for confirmation before the action is completed.

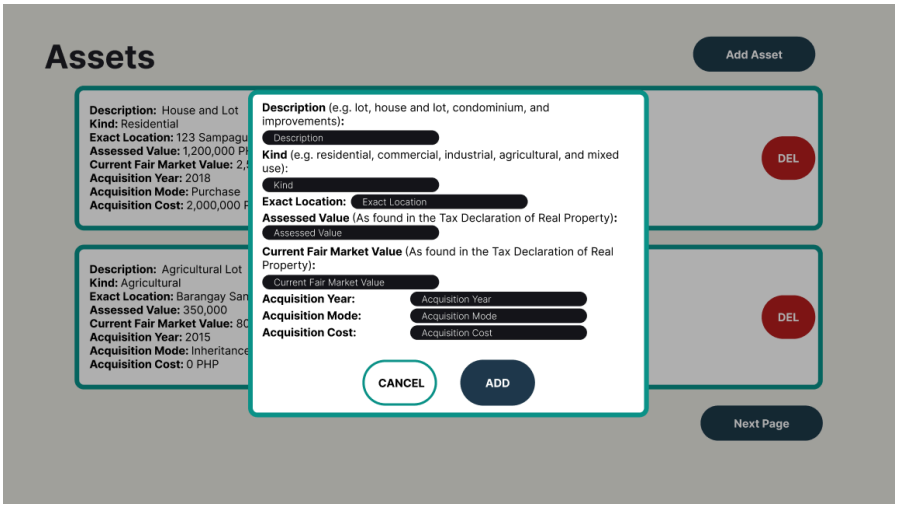


Should the user choose to upload a JSON file, they'll be sent to a page that prompts them to upload a JSON file.



Should the user choose to fill in a SALN form, they will be sent to a responsive version of the SALN Form. To reduce clutter, the form will be divided into several sections, each having its own page. Additionally, tables in the original SALN form will be displayed by setting each entry to card view. This would allow for a more responsive representation of the tables. Should a user wish to add a new row (or card) to the table, a modal will appear prompting the user to enter the information for the card.





Should the user choose to generate a PDF or export a JSON file of a filled SALN form, clicking the appropriate button would automatically export the file into the device's files.

Plan of Work

I. Estimation of Costs and Resources

- Using the WebApp software project resource estimation technique as described by Roetzheim, we have identified the following function points for our project:

    a. Inputs - The SALN app will have the following inputs

        I. User Login Page: Account credentials, four-word OTP

        II. SALN Form Pages: Data to be provided for the SALN accomplished through either filling out the online form or uploading a JSON file

        III. Miscellaneous: Deleting user account, removing SALN forms filled less than 5 days ago

    b. Outputs - The SALN app will have the following outputs

        I. Static pages: Login page UI, Upload JSON page UI, Empty SALN Forms UI, Modal UI for adding rows

        II. Dynamic web page scripts: Form pages with different data and Dashboard page

        III. Reports: Generated SALN Form and JSON data exported

    c. Tables - The data to be stored in the app are the user account information and the SALN data. All of this information will be stored in an encrypted database.

    d. Interfaces - The SALN app will have the following interfaces:

I. Email Verification Interface - The app sends the user an email with four random words for verification purposes.

II. JSON Import/Export Interface - The user can import a JSON file to create a SALN, and the app can also provide a JSON file for the SALN data.

III. PDF Export Interface - The user can export the SALN as a PDF file.

e. Queries - These are the queries that the app will need:

I. Checking if the user is trying to make an account with an email already registered in the database.

II. Checking if the entered credentials are correct.

III. Fetching the user's SALN data from the last 5 days

IV. Checking if the JSON file uploaded is valid or if the file is in JSON format.

For the Backend, the platform will use Linux, PHP, and Laravel. The Frontend will use HTML, CSS, and the Marko JavaScript framework. Using all of these services is free. The cloud server hosting will be done using Digital Ocean. It is estimated that around $6 per month is required, which can be covered using the $200 student plan, which is valid for one year.

- Using the process-based estimation technique, we have the following table

Dev Team Tasks	%Effort (Percentage of TIme)	Total Time(In Weeks)
Communication and Client Meetings	4.76%	1
Requirements Gathering and Analysis	19.05%	4
Frontend Programming	9.52%	2
Backend Programming	9.52%	2
Hosting	4.76%	1
Testing	28.57%	6
Maintenance	23.81%	5
Total	100%	21

Based on this, a majority of the effort will be spent on testing, maintenance, and requirements gathering and analysis. Minimal time will be spent on programming and hosting.

II. Identification of Schedules

Dev Team Tasks include Software Development documents needed for CS 191 and CS 192. Client Meetings are required at least once a month. For client meetings, there will be two dates in parentheses next to the agenda: Preferred Meeting Date and Due Date of Meeting Agenda. Client Meetings are most likely to be held on Mondays due to that being common available free time of all stakeholders, despite the due dates for the agenda of that meeting to be

due on a later day during the week. However, client meetings could be postponed to some time later that week.

There are weeks allocated for Project Development. Every Monday after a Project Development week, the Dev Team will send a progress report to the client.

For a general overview, the first half of the schedule will be used for the functionality of the SALN App, and then the second half will be used to turn the app into a Progressive Web App.

Week / Dates	Dev Team Tasks	Client Meetings
Aug 25 - Aug 29	**Group Blog & Project Plan** (Due Aug 29)	
Sep 1 - Sep 5	System Requirements Document Pt 1	
Sep 8 - Sep 12	System Requirements Document Pt 1	**Discuss SRD** (Sep 8)
Sep 15 - Sep 19	**SRD Pt 1** (Due Sep 17); System Requirements Document Pt 2	
Sep 22 - Sep 26	**SRD Pt 2** (Due Oct 3)	**Finalize SRD** (Oct 1)
Sep 29 - Oct 3	Frontend (Dashboard/Homepage) **PROJECT IMPLEMENTATION START**	
Oct 6 - Oct 10	Frontend (SALN Form)	Progress Report
Oct 13 - Oct 17	Backend (Database Setup & APIs)	Progress Report
Oct 20 - Oct 24	Backend (Data Encryption)	Progress Report
Oct 27 - Oct 31	Host on Digital Ocean	Progress Report
Nov 3 - Nov 7	Write Tests	**Software Release v.1** (Nov 3; Due Nov 5)
Nov 10 - Nov 14	Write Tests; Bug Fixes	**Get v.1 comments** (Nov 10)
Nov 17 - Nov 21	Software Test Report	
Nov 24 - Nov 28	**Software Test Report** (Due Nov 26)	
Dec 1 - Dec 5	Bug Fixes	**Closing CS 191; Improvements for CS 192** (Dec 1)
Dec 8 - Dec 12	BUFFER	

Week / Dates	Task	Client Meetings
Jan 19 - Jan 23	Project Dev Based on v.1 Comments	
Jan 26 - Jan 30	Project Dev Based on v.1 Comments	Progress Report
Feb 2 - Feb 6	Software Maintenance Plan	
Feb 9 - Feb 13	**Software Maintenance Plan** (Due Feb 11)	**Software Release v.2** (Feb 9; Due Feb 13)
Feb 16 - Feb 20	Write Tests; Bug Fixes	**Get v.2 Comments; Project Evaluation by Client** (Feb 16, Due Feb 20)
Feb 23 - Feb 27	Project Dev Based on v.2 Comments	
Mar 2 - Mar 6	**Software Test Report** (Due Mar 6)	Progress Report
Mar 9 - Mar 13	Write Tests; Bug Fixes	**Software Release v.3** (Mar 9, Due Mar 13)
Mar 16 - Mar 20	Write Tests; Bug Fixes	**Get v.3. Comments** (Mar 16, Due Mar 20)
Mar 23 - Mar 27	Project Dev Based on v.3 Comments	
Mar 30 - Apr 3	**Software Test Report** (Due Apr 3)	Progress Report
Apr 6 - Apr 10	Write Tests; Bug Fixes	**Software Release v.4; Project Turnover; User Training** (Apr 6, Due Apr 10)
Apr 13 - Apr 17	NONE [HOLY WEEK]	NONE [HOLY WEEK]
Apr 20 - Apr 24	Maintenance	
Apr 27 - May 1	Maintenance	
May 4 - May 8	Maintenance; **Project Presentation** (Due May 8)	
May 11 - May 15	Maintenance;	**Project Documentation** (Due May 15)
May 18 - May 22	Maintenance	

III. Project Metrics

Since the product is a progressive web app, this project will use WebApp Project Metrics to keep track of the project's status.

As the digital SALN form's status changes depending on user inputs, one of the project metrics to be used will be the number of dynamic web pages. This will be the project metric to keep track of the project status during the Frontend phase.

Additionally, since the app will make use of a database to store users' SALN Form answers, the number of persistent data objects must also be a metric. This will be the project metric to keep track of the project status during the Backend Phase.

Aside from the persistent data objects, the number of executable functions will also be used as a metric. These include functions like sending emails for verifications, inserting and deleting SALN data from the database, and exporting SALN data to PDF or JSON format.

IV. Effort Breakdown

As shown in the Identification of Schedules chapter, the project implementation could be broken down into three phases: Frontend, Backend, and Deployment. For the sake of each member experiencing being a team lead, for each phase, one of the three members of the development team would act as the team lead for the phase, while the other two would assist. As the team lead, they are mostly responsible for brainstorming the necessary elements of that phase.

For the Frontend Phase, Lucas would be the team lead. In essence, the team lead for the Frontend would brainstorm the layout of the app. This includes how the app would look, what buttons and input elements are needed, and how the frontend elements of the app would improve

the user's experience. Particularly, he would brainstorm the elements of the dashboard and how the SALN Form could be represented as a digital form on the app.

For the Backend Phase, Miguel would be the team lead. The team lead for the Backend would brainstorm how user accounts and form answers would be represented in the database and would also set up the database. The Backend Team Lead would also be in charge of brainstorming the executable functions needed for the frontend to be connected with the database, to send emails for verification, and to export SALN data.

For the Hosting Phase, Dann would be the team lead. The main task of the Hosting Team Lead would be to configure a cloud server that is able to host the server-side computations. Additionally, the Hosting Team Lead would be in charge of the Web App's deployment alongside securing a Secure Sockets Layer (SSL) Certificate.

As for coordinating the integration, the Back End Team lead will take charge of this since both Frontend and Hosting would make use of the work products from the Backend. As for writing unit and integration tests, all members will share that responsibility.

For turning the web app into a progressive web app, the Back End Team Lead would take charge, due to most of the processes being in the backend.

Project Risks

Risk	Category	Probability	Impact	RMMM
Learning new technologies to the Dev Team might cause delays	Staff Size and Experience	70%	2	Team Leads for each phase should focus more on learning about the technologies for their phase.
Web Server may crash due to limitations of DigitalOcean Droplet purchased.	Development Environment	50%	3	Scope of the Project defines the expected number of users.
Meetings may be postponed due to schedule conflicts.	Stakeholder Characteristics	50%	3	Meetings between Dev Team and Client could be postponed to some other time in the week. Dev Team may not have all members present.
Dev Team might see delays in schedule due to academic workload	Stakeholder Characteristics	80%	2	CS 191 actually gives sessions just for project implementation.
Data Encryption may not meet expectations.	Technology to be built	60%	1	Allot time to learn about encryption. Clarify data privacy requirements with Client.

Impact Values: 1-Catastrophic; 2-Critical; 3-Marginal; 4-Negligible

References

Karol Ilagan, R. D. M. (2020, October 18). *A citizen's guide to where and how to get a saln*. PCIJ.org. https://pcij.org/2020/10/18/a-citizens-guide-to-where-and-how-to-get-a-saln/