

PMV Mini

Background and Requirements

First Solar wanted to replace an existing application (PV Tools) that had become outdated and was no longer being maintained. It was a simple application designed to upload .csv files to a database with background processing that validated module serial numbers and checked for duplicate scans. The basic requirements for my project were to mimic that application's functionality, but increase efficiency and ease of use. We wanted to add the ability to visualize and edit data before uploading it, provide user feedback on potential errors in the data, allow users to edit data already uploaded to the database, and improve overall performance without dramatically changing the underlying infrastructure (i.e., the database tables and labeling practices used for storage of processed scans and processing history records).

Development Stack and Reasoning

I used the Vue.js JavaScript framework for the client-side codebase and the .NET Web API framework for the server-side codebase of this application. One important reason for both choices was the desire to be able to merge this application into a larger application (PMV) being built concurrently by a separate development team—this also provided me with valuable resources to turn to with questions I ran into while using these tools for the first time. While .NET is First Solar's established choice for server-side development and database interactions, Vue.js is less prevalent, but was chosen specifically for use in PMV. Ultimately, I deemed the reusable template/component-based structure of Vue.js and simplicity of adding Node.js libraries like Vuex (state management), Vuetify (UI), and Axios (http requests) a great fit for the needs of the application and my programming skillset (no prior experience with JavaScript frameworks, but a basic understanding of HTML/CSS/JavaScript and strong object-oriented background).

Takeaways

This project taught me a lot about software development in the context of a larger business. It surprised me how much of my time went towards tasks like configuring DEV/QA environments as opposed to simply writing/testing code, and how seemingly small obstacles in these processes could result in large delays. In the future, I would take more care to ensure the path to production is ready throughout the development process, seeing as that is the point at which the software's value-added is finally realized. I would also make an effort to get more direct feedback on my code itself, especially when using new languages. I definitely appreciated the opportunities I had to get feedback on the design and functionality of my application, as well ask questions about the tools I was using, and I think formal code-reviews from co-workers with more development experience than me would further improve my quality of work.

Code Excerpts and Application Screenshots

PMV Mini

Upload Scans































BOX/PALLET SCANS

MODULE SCRAP

DATA CORRECTIONS

Search

Search

Serial Number ↓	Latitude	Longitude	Elevation	Site Name	Scan Time	Edit/Delete	Status
5386961	30.995424	-102.281351	703		7/12/2016 9:31	 	
5386961	30.995424	-102.281351	703	East Pecos Solar		 	
5509211	30.995497	-102.281035	684	East Pecos Solar	7/12/2016 10:22	 	
5509211	30.99549	-102.281085	689	East Pecos Solar	7/12/2016 10:34	 	
5509211	30.995494	-102.281031	685	East Pecos Solar	7/12/2016 10:22	 	
5509211	30.995468	-102.281053	691	East Pecos Solar	7/12/2016 10:34	 	
5509211	30.995468	-102.28106	688	East Pecos Solar	7/12/2016 10:23	 	
5509211	30.995464	-102.281057	687	East Pecos Solar	7/12/2016 10:23	 	
5508998	30.99582	-102.280946	697	East Pecos Solar	7/12/2016 10:37	 	
5508998	30.995827	-102.28094	697	East Pecos Solar	7/12/2016 10:37	 	

Rows per page: 10 1-10 of 579

© 2018 First Solar For any issue/help with the system, please contact [Service Desk](#)

PMV Mini

DATA PARAMETERS

Search

Select parameters for the data you wish to analyze.

Data Parameters

Data Type*
Box/Pallet Scans

Site Name
AVRA

Read Time
Start Date*
2012-07-31
End Date*
2012-08-07

2012
Tue, Jul 31

July 2012

S M T W T F S

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31

LOAD CANCEL





















© 2018 First Solar For any issue/help with the system, please contact [Service Desk](#)

PMV Mini

DATA PARAMETERS

7/31/2012

Search

Serial Number	Latitude	Longitude	Elevation	Site Name	Scan Time ↑	Edit	Status
04035475	32.373679	-111.2804	615	AVRA	7/31/2012 12:46		
03972880	32.373962						
04008357	32.37408						
04037638	32.373825						
04008425	32.374198						
04035697	32.374333						
04001829	32.374472	-111.280424	612	AVRA	7/31/2012 12:53		
04035216	32.374608	-111.28038	612	AVRA	7/31/2012 12:54		
04035231	32.374701	-111.280413	612	AVRA	7/31/2012 12:54		
04035462	32.374816	-111.28041	611	AVRA	7/31/2012 12:54		

Rows per page: 10 1-10 of 20

© 2018 First Solar For any issue/help with the system, please contact [Service Desk](#)

Edit Scan

Site Name

Scan Time
7/31/2012 12:46

Field cannot be left blank.

CANCEL SAVE

Vue.js (Client)

```
BoxScans.vue x ServerError.vue UploadTab.vue UploadProcessing.vue DataCorrections.vue JS globalFedUploads.js JS GlobalFedUploadService.js
1 <template>
2 <div>
3   <server-error
4     v-if="serverError"
5   ></server-error>
6   <upload-processing
7     v-else-if="scans.length == 0 && boxProcessing"
8     :name="name"
9     :statusMessage="statusMessage"
10  ></upload-processing>
11   <upload-tab
12     v-else
13     ref="child"
14     :name="name"
15     :headers="headers"
16     :scans="scans"
17     :errorChecks="errorChecks"
18     :warningChecks="warningChecks"
19     :processing="boxProcessing"
20     :statusMessage="statusMessage"
21     @update-scans="updateScans"
22     @post-scans="postBoxScans"
23     @reset-scans="scans = []"
24     @get-status="getBoxScansProcessingStatus"
25   ></upload-tab>
26 </div>
27 </template>
28
29 <script>
30 import { mapGetters, mapActions } from 'vuex';
31 import serverError from './Error/ServerError';
32 import uploadProcessing from './Upload/UploadProcessing';
33 import uploadTab from './Upload/UploadTab';
34
35 export default {
36   data: () => ({
37     name: 'Box/Pallet Scans',
38     headers: [
39       { text: 'Serial Number', value: 'num' },
40       { text: 'Latitude', value: 'latitude' },
41       { text: 'Longitude', value: 'longitude' },
42       { text: 'Elevation', value: 'elevation' },
43       { text: 'Site Name', value: 'site' },
44       { text: 'Scan Time', value: 'date' },
45       { text: 'Edit/Delete', value: 'edit', sortable: false, align: 'center' },
46       { text: 'Status', value: 'scanStatus', sortable: false },
47     ],
48     scans: [],
49     errorChecks: [
50       // check for missing site name/scan time
51       scan => !scan.site.data.length,
52       scan => !scan.date.data.length,
53     ],
54     warningChecks: [
55       // check for missing serial num/lat/long/elevation
56       scan => !scan.num.data.length || scan.num.data === '0',
57       scan => !scan.latitude.data.length || scan.latitude.data === '0',
58       scan => !scan.longitude.data.length || scan.longitude.data === '0',
59       scan => !scan.elevation.data.length || scan.elevation.data === '0',
60     ],
61   ]),
62   computed: {
63     ...mapGetters('globalFedUploads', ['boxProcessing', 'boxCompletionPercentage', 'serverError']),
64     statusMessage() {
65       return `${this.boxCompletionPercentage}% Complete`;
66     },
67   },
68   methods: {
69     ...mapActions('globalFedUploads', ['getBoxScansProcessingStatus', 'postBoxScansToTempData', 'postBoxScansToDatabase']),
70     updateScans(lines) {
71       this.loadTableData(lines).then(() => {
72         this.$refs.child.$refs.child2.updateScanStatusesFrom(0);
73       }, (msg) => {
74         this.$refs.child.showAlert('Invalid Number of Fields', msg, 'error', 'error');
75       });
76     },
77     loadTableData(lines) {
78       return new Promise((resolve, reject) => {
79         // Check for typical Box Scan file characteristics
80         if (this.isValid(lines)) {
81           // Determine whether first line contains headers
82           let start = 0;
83           if (lines[0].split(',')[0].includes('Scanner')) {
84             start = 1;
85           }
86           // Read in csv data ignoring headers and last line (blank)
87           for (let i = start; i < lines.length - 1; i += 1) {
88             const line = lines[i].split(',');
89             // Scan data listed in same order as header data
90             const scan = {
91               num: { data: line[1].trim(), visible: true, index: 0 },
92               latitude: { data: line[2].trim(), visible: true, index: 1 },
93               longitude: { data: line[3].trim(), visible: true, index: 2 },
94               elevation: { data: line[4].trim(), visible: true, index: 3 },
95               site: { data: line[5].trim(), visible: true, label: 'Site Name', index: 4 },
96               date: { data: line[6].trim(), visible: true, label: 'Scan Time', index: 5 },
97               scanner: { data: line[0].trim(), visible: false, index: 6 },

```

.NET Web API (Server)

```
BoxScan.cs  BoxController.cs  BoxLogic.cs  BoxScanData.cs  BoxData.cs  BoxMap.cs  PVAppDataModel.edmx [Diagram1]  Web.config
GlobalFedComponents
2 references
42 public BoxScan[] AddItemsToTemp(BoxScan[] scans)
43 {
44     BoxScanData.ClearTempData();
45     return BoxScanData.AddTempBoxScans(scans);
46 }
47
48 public int PostToDatabase(string user)
49 {
50     using (IBoxData boxData = new BoxData())
51     using (IScannedDataProcessingHistoryData scannedDataProcessingHistoryData = new ScannedDataProcessingHistoryData())
52     {
53         BoxScanData.GetEDWData();
54         IEnumerable<string> duplicateSerialNums = boxData.FindDuplicateSerialNums(BoxScanData.BoxSerialNums);
55         List<BoxScan> edwErrorScans = new List<BoxScan>();
56         List<BoxScan> globalFedErrorScans = new List<BoxScan>();
57         List<BoxScan> validScans = new List<BoxScan>();
58         int total = BoxScanData.BoxScans.Length;
59
60         foreach (BoxScan sc in BoxScanData.BoxScans)
61         {
62             sc.user = user;
63             if (sc.watts == null)
64             {
65                 edwErrorScans.Add(sc);
66             }
67             else if (duplicateSerialNums.Contains(sc.num) || validScans.Exists(scan => scan.num == sc.num)) // account for duplicates within the data being uploaded
68             {
69                 globalFedErrorScans.Add(sc);
70             }
71             else
72             {
73                 validScans.Add(sc);
74             }
75         }
76
77         Debug.WriteLine("edwErrors: " + edwErrorScans.Count);
78         Debug.WriteLine("globalFedErrors: " + globalFedErrorScans.Count);
79         Debug.WriteLine("validScans: " + validScans.Count);
80     }
81 }
```

```
BoxScan.cs  BoxController.cs  BoxLogic.cs  BoxScanData.cs  BoxData.cs  BoxMap.cs  PVAppDataModel.edmx [Diagram1]  Web.config
GlobalFedComponents
2 references
80
81 int batchCount = 0;
82 foreach (BoxScan scan in validScans)
83 {
84     batchCount++;
85     GlobalFedStatus.BoxScansPercentage = (int)((double)batchCount / total) * 100;
86     Debug.WriteLine(batchCount);
87     Debug.WriteLine(GlobalFedStatus.BoxScansPercentage);
88     bool success = true;
89     try
90     {
91         boxData.AddItem(scan);
92     }
93     catch (Exception ex)
94     {
95         success = false;
96         saveToProcessingHistory(scan, "Error", "Exception processing uploaded record with serial number: " + scan.num + ". Error: " + ex.Message, batchCount, total, scannedDataProcessingHistoryData);
97     }
98     if (success)
99     {
100         saveToProcessingHistory(scan, "Complete", "", batchCount, total, scannedDataProcessingHistoryData);
101     }
102     if (batchCount == validScans.Count || batchCount % BATCHSIZE == 0)
103     {
104         boxData.SaveChanges();
105     }
106 }
107 foreach (BoxScan scan in edwErrorScans)
108 {
109     batchCount++;
110     GlobalFedStatus.BoxScansPercentage = (int)((double)batchCount / (double)total) * 100;
111     saveToProcessingHistory(scan, "Error", "Could not get MES DW information for this serial number: " + scan.num, batchCount, total, scannedDataProcessingHistoryData);
112 }
113 foreach (BoxScan scan in globalFedErrorScans)
114 {
115     batchCount++;
116     GlobalFedStatus.BoxScansPercentage = (int)((double)batchCount / (double)total) * 100;
117     saveToProcessingHistory(scan, "Complete", "Duplicate Scan.", batchCount, total, scannedDataProcessingHistoryData);
118 }
119
120 BoxScanData.ClearTempData();
121 return batchCount;
122 }
123
124 private void saveToProcessingHistory(BoxScan scan, string status, string message, int batchCount, int total, IScannedDataProcessingHistoryData scannedDataProcessingHistoryData)
125 {
126     ProcessedScan processed = new ProcessedScan
127     {
128         num = scan.num,
129         raw = scan.raw,
130         status = status,
131         message = message,
132         user = scan.user,
133     };
134     scannedDataProcessingHistoryData.AddItem(processed);
135     if (batchCount == total || batchCount % BATCHSIZE == 0)
136     {
137         scannedDataProcessingHistoryData.SaveChanges();
138     }
139 }
140 }
```