International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 07 | July 2022 www.irjet.net p-ISSN: 2395-0072

Planner Application Based on Customer Relationship Management

Parakh Shah[1], Sushmitha N[2]

[1,2] *Department of Information Science and Engineering, R V College of Engineering, Bangalore, India*

Abstract – *This project involves building a calendar or planner-like view for the managers. It should enable the management level staff to perform various everyday level tasks and improve productivity. The project aims to make day-to-day tasks like seeing a list of all to-dos for their team easier. The aim is to aggregate data from multiple Customer Relationship Management (CRM) sub-modules. This data needs to be shown and visualized in an easily digestible, calendar grid view. All tasks against a lead should be shown in an organized manner to improve viewability. Everyday tasks and dealership traffic should be visualized in the planner. There must be weekly, bi-weekly, and monthly views available at the minimum. Various things need to be learned from the existing code. This will help in developing new modules and integrating them into existing systems. A learning-based project with a similar tech stack was given to us and then the project CRM planner was developed.*

Key Words: **Customer Relationship Planner (CRM), Automotive Retail Cloud (ARC), Software as a Service (SaaS), User Interface**

1. INTRODUCTION

The dealer module was already there when the initial project for the CRM module began. There were various sub-modules for leads, tasks, and appointments in this CRM module. An employee must therefore examine each module to see his tasks. It was challenging for a manager to keep track of the responsibilities given to each employee who reported to him. To address the issues the staff members are experiencing and the managers, a decision was made to create a CRM planner. A tool called CRM (Customer Relationship Management) is used to control relationships with current and potential customers. The CRM module is designed specifically for auto dealers to manage their leads and customers.

CRM users can view all of their tasks and appointments, including interacting with customers, in a calendar-like view. The CRM planner consists of a month-view, a weekly view, and a three-day view. The tasks that have been allocated to the users will be simpler for them to see. The responsibilities and appointments of the workers who report to him can be seen by the manager. Additionally, the boss will have the ability to plan his own and other employees' responsibilities and appointments.

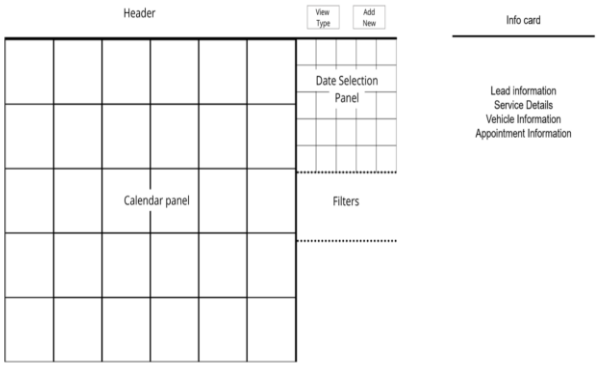

Fig 1 - Prototype of Planner UI

The objectives of Planner include building a planner/calendar for managers to keep track of appointments and tasks created for different services. It also focuses to develop a service for clients to schedule appointments and complete them through different means like call, text, mail, and message and to schedule appointments and tasks at any given time and any given day, which can be managed by the different views in the application namely Day View, Three Day View, Week View, Month View, and, lastly, to make it easier to find a certain activity by utilizing different criteria for both service appointments and sales assignments.

The modernized form of CRM is partly a result of the increased use of technology in maintaining and maximizing the value of information. The planner also uses a technology stack specifically related to information technology, which is used to build the project and use them as products for the future.

1.1 LITERATURE SURVEY

In paper [1], It is well acknowledged that there is a growing market for tools and apps that operate more quickly. JavaScript frameworks have several benefits, including ease of use and straightforward framework interaction. In [1], Vue.js and React.js are contrasted. In [2], ReactJS is a part-based library that is conveyed for the advancement of intelligent UIs. Right now, it holds the title of the most popular front-end JS library. It integrates the view (V) layer of the M-V-C (Model View Controller) design.

It is supported by Facebook, Instagram, a network of independent designers, and several organizations. Respond fundamentally makes it possible to create complex, large-scale online applications that can change their content without necessitating new page revivals.

It aims to provide improved customer experiences while blazingly quickly and powerfully developing online applications. ReactJS can likewise coordinate with other JavaScript libraries or structures in MVC, for example, AngularJS.

In paper [3] Individual aspects in web applications, such as typefaces, text weight, colors, backgrounds, width, and height, are all affected by the style. The use of CSS Modules and CSS in JS, as well as how to include and use CSS files in React, are all covered in this chapter. The simplest method for styling a React user interface is including one or more CSS files in the HTML file that launches React. When using traditional CSS stylesheets, CSS Modules offer some of the benefits of using JavaScript style objects. There are numerous third-party libraries available for JS that implement CSS.

In [4] This senior thesis' major goal was to investigate a front-end JavaScript library for creating online and mobile applications. Presently, there are numerous front-end application development frameworks and libraries. One of them is the ReactJS library, one of the newest web technologies. It's been demonstrated to be the quickest rendering library. It focuses on the view component of the MVC paradigm and is extensively used for the creation of large-scale applications. It was created by Facebook for internal use, and compared to other technologies, it has proven to be a quick and effective library.

The primary objective of this chapter, as stated in paper [5], is to complete the React web application by adding URL routing and the remaining components before preparing to deploy the application in a container. The TypeScript compiler parameters used in this chapter are listed in Table 20-1 for your convenience. In this chapter, URL routing and the last few components needed to prepare React web applications for container deployment are added. The TypeScript compiler parameters used in this chapter are listed in Table 20-1 for your convenience. According to a paper [6], the majority of higher education institutions (HEIs) have silos of scattered procedures, which further exacerbates the ambiguity and conflict surrounding customer relationship management (CRM), wishes, expectations, and demands. (DEN). To maximize resource effect and reduce organizational disruption, HEIs should practically map this DEN to existing processes, roles, events, activities, channels, and technologies (PRE-ACTs) within the institution. The research also analyses the use of additional practical approaches that must be taken into account to ensure that strategic HEI CRM DEN is properly captured and that the needs are adequately matched to actual HEI operations.

Using reference in [8], it first relates to the CIDOC-CRM standard, as well as the data model and description vocabularies. Knowledge graphs can be enabled by a variety of technologies, two examples being a graph database and an object graph mapping library. The case study is the Portuguese National Archives, Torre do Tombo, and the major goal is to develop a system for document description and retrieval that complies with CIDOC-CRM and is accessible to both professionals and members of the general public. The Arch Onto ontology and its incorporation into the Arch Graph knowledge graph served as the primary data model for archive records when it was first designed. The creation of applications made possible by the graph and the translation of existing records to a richer representation will both contribute to the goal of a semantic archival information system.

2. METHODOLOGY

Figure 1 shows a block diagram representing the entire flow of the proposed work. Figure 2 shows the high-level design for the project. A system called customer relationship management (CRM) enables one to oversee all of one's business's interactions and relationships with both present and potential clients. The goal is clear: to foster stronger business relationships. A CRM system helps organizations maintain relationships with customers, streamline operations, and boost profitability.

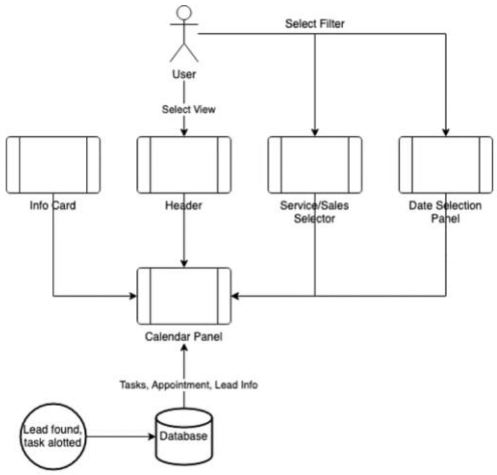

Fig 2 - System Architecture - HLD

The implementation of this project proceeds in different phases -

1. To implement the first task, the tools used mainly include the Full Calendar Module:

- The project makes use of the Full Calendar package to help build the application fluently.
- React JavaScript is elegantly integrated with Full Calendar.
- It provides a component that exactly matches the functionality of Full Calendar's standard API.
- By instructing Full Calendar's core package to start rendering with React virtual DOM nodes rather than the react nodes it typically uses, it becomes a "true" React component.

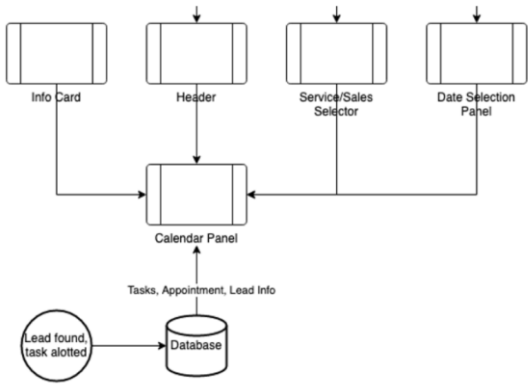

Fig 3 - System Design - Calendar Panel

2. For the implementation of functionalities and modules i.e., the second category:

- Once the Full Calendar component is working properly, it makes use of the design document to implement the changes that are mentioned and also adds new functionality if any exists to suit the customer's needs
- This is achieved using different code changes using React and Redux state management tools. And the versions of the modules are managed using bitbucket and git.
- Every module is followed by preproduction and release and accordingly, the rest of the changes are made to the project considering the circumstances.

3. For Integration of Spring Boot React Full-Stack Architecture:

- The next step is to combine the React backend with the Spring Boot back-end along with the local data to carry out the project.
- The project uses React as the front-end stack and Spring Boot as the Back-end stack.

4. **Adding Filters -**

- As shown in Figure 2, the main dashboard/planner consists of various views and filters associated with it.
- Sales appointments and service tasks can be created through both the Add New button on the top right corner and the on Click functionality of any of the dates and time slots as shown.

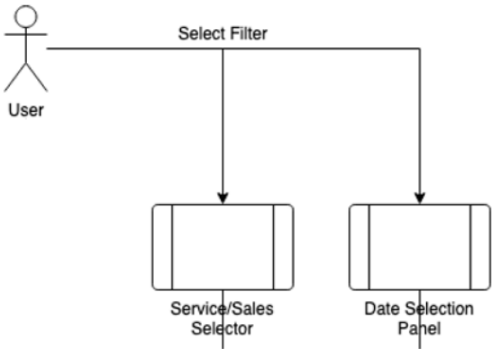

Fig 4 - System Design - Filters and Date Selection Panel

5. Data Retrieval -

- Data retrieval becomes easy with the use of various filters provided in the bottom right corner consisting of both the sales appointments and service tasks filters.
- There are multiple views namely Day, Three Day, Week, and Month views, and the slots are divided into Days and time slots respectively.

6. Customer Service -

The next step is to deliver the Planner to different clients and incorporate the changes that they suggest and maintain the optimization and availability at all times.

3. RESULTS AND DISCUSSIONS

The design aims to aggregate data from multiple CRM sub-modules. This data can be from different leads which consist of appointments for a test drive or showroom visits for purchase/inquiry. The data can also be different tasks which can comprise responding to pending customer queries or pending emails and calls. Lastly, the appointments are mainly divided into two types: Sales Appointments and Service Appointments.

This data needs to be shown and visualized in an easily digestible, calendar grid view. All tasks against a lead should be shown in an organized manner to improve viewability. Everyday tasks and dealership traffic should be visualized in the planner. There must be weekly, bi-weekly and monthly views available at the minimum. Presenting CRM as an important and time-saving tool will inspire people to utilize it on a wide range of occasions. Managers must make CRM software a vital medium for making or recording all everyday contacts by the Sales staff. Make sure the sales infrastructure understands that CRM may help them keep current on scheduled activities and is more than just a tracking tool.

Software is used in almost all of the organization's processes to replace physical labor. The problem is that none of these systems are connected, and no data is exchanged between them. When creating CRM software for a company, it is essential to understand the needs of both the employees and the business. The expectations of all personnel levels, whether they are Management Users, Sales Representatives, Service Executives, or Marketing Managers, must be met by a successful CRM installation.

4. CONCLUSION

All firms that have senior management make decisions regarding CRM deployment due to the financial and operational changes that will result from the CRM's implementation. The entire team must be informed of this decision and the subsequent process, and senior management must support the plan throughout the implementation stage. A lack of communication among the staff members may lead to insecurities and mental hurdles.

The major future enhancements that the project includes are adding more relevant filters to the application. Currently, the application deals with only three types of filters. One is the default filter and one contains different filter values from both Sales Tasks and Service Appointments. To improve the user experience and to increase the speed of data retrieval, we can add more relevant filters to the application. The second major future enhancement is the integration of an already existing scheduler with the Planner - Next task in future enhancements is the Integration of the CRM Scheduler with the Planner. Schedulers do an overall different task of scheduling tasks and working hours of different people and make the process more efficient for the manager. The integration of both of them will lead to a great improvement in user experience and customer satisfaction.

REFERENCES

[1] C. M. Novac, O. C. Novac, R. M. Sferle, M. I. Gordan, G. BUJDOSó and C. M. Dindelegan, “Comparative study of some applications made in the Vue.js and React.js frameworks”, 16th International Conference on Engineering of Modern Electric Systems (EMES) - 2021

[2] Archana Bhalla, Shivangi Garg, Priyangi Singh, “PRESENT DAY WEB-DEVELOPMENT USING REACTJS”, International Research Journal of Engineering and Technology (IRJET) - 2020

[3] Minnick, Chris, “Styling React”, Beginning React JS Foundations Building User Interfaces with ReactJS –2022

[4] Naimul Islam Naim, “ReactJS: An Open-Source JavaScript Library for Front-end Development”, Metropolia University of Applied Sciences Bachelor of Engineering Information Technology Thesis 30 - May 2020

[5] Adam Freeman, “Creating a React App- Part 2”, Essential TypeScript 4, From Beginner to Pro - 2021

[6] Basel Khashab, Stephen R. Gulliver, Rami M Ayoubi, Carolyn strong, “Analysing enterprise resources for developing CRM framework in higher education institutions”, Scoping and aligning CRM strategy in higher education institutions: practical steps, Journal of Strategic Marketing - 2020

[7] A Terminanto, R Hidayat and A N Hidayanto, “Implementation of enterprise resource planning using Odoo module sales and CRM”, IOP Conf. Series: Materials Science and Engineering 277 - 2017

[8] Inês Koch, Nuno Freitas, Cristina Ribeiro, Carla Teixeira Lopes & João Rocha da Silva, “Knowledge Graph Implementation of Archival Descriptions Through CIDOC-CRM”, International Conference on Theory and Practice of Digital Libraries - 2019

[9] Vipul Kaushik, Kamal Gupta, Deepali Gupta (2018). React Native Application Development

[10] Tim A. Majchrzak, Comprehensive Analysis of Innovative Cross-Platform App Development Frameworks.

[11] M. Miškuf and I. Zolotová, "Architecting React Applications with Flux," 2020 IEEE 13th International Symposium on Applied Machine Intelligence and Informatics (SAMI), Herlany, 2015, pp. 193-197

[12] Talhaoui Y., Kohtamäki M., Rabetino R. (2019) Business Intelligence—Capturing an Elusive Concept. In: Kohtamäki M. (eds) Real-time Strategy and Business Intelligence. Palgrave Macmillan, Cham.

[13] Krill, P., May 15, 2014, "React: Making faster, smoother UIs for data-driven Web apps,"

[14] Hemel, Z., June 03, 2013, "Facebook's React JavaScript User Interfaces Library Receives Mixed Reviews,".

[15] Dawson, C., 25 Jul 2014, "JavaScript's History and How it Led to ReactJS,".

[16] Silva, E., December 21, 2017, "How to Make create-react-app work with a Node Back-end API,".