A Survey on Activity Planner using OpenUI5 Framework

Suhas Bhattu[1]

P.G. Student, Department of Computer Engineering, Pune Institute of Computer Engineering, Pune,

Maharashtra, India[1]

ABSTRACT: Calendar component is the essential need for any organization for arranging and scheduling activities, planning meetings. Many calendar products and schedulers are available in the market that provides basic calendar functionality but still lack in certain features like optimized scheduling, Gantt chart for effective visuals for the hierarchy of the events. These features improve the overall performance and user interface. This work proposes the marketing activity calendar component using the OpenUI5 developed by SAP. Using OpenUI5's inbuilt JavaScript modules and libraries and the model-view-controller (MVC) software design pattern, new features like Gantt chart is possible to develop and also the scheduling algorithms can also be applied to schedule the events.

KEYWORDS: Calendar Component, Gantt chart, MVC, SAP OpenUI5.

I. INTRODUCTION

Scheduling of events is the most important and the crucial point in the development of any production process. Allocation of resources according to their availability and as per their requirement, scheduling has to be done in the most effective and optimized way. Scheduling is applied in various fields, including the manufacturing processes, software development, and in much system software including the operating system. Thus Scheduling remains to be most challenging research topics in the development field. Scheduling is also the important part of the marketing activities involved in the product development. Activities such as the setting up meetings dates, product design dates, deciding product delivery dates are involved in the marketing activities. In so arranging the activities according to their priority, and requirement of resources is the most challenging task.

The calendar component is the automated calendar that renders the added activities on the calendar according to their start and end dates. The calendar component is responsible for accepting the inputs from the user as an activity to be added to the calendar. After that component arranges the activities according to their start and end dates on the date panel of the calendar. There are many calendar planners available in the market now, which provide multiple functionalities like providing the different views like the monthly, daily and weekly view which gives the user different aspects about the activities in the different time frames. These calendar components make the planning very efficient and elegant, such that any user can understand the activities according to their duration. The user interface should be the most important, but most ignored part in these calendar components. The good user interface practice not only attracts customers but also makes the system data flow easier to understand. Therefore the basic views of the calendar should represent the activity in a user-friendly way. There is one more facility that most of the calendar planner providers don't deliver is the Gantt chart and the hierarchy of the events. Gantt chart helps us to visualize activities in the real-time frames [4]. So users can see all activities according to their time frames in a single chart and then it is easy to compare and analyze activities. Also, the Gantt chart reduces the complexity of scheduling of activities by representing them in great visual form. Without Gantt chart, it is difficult to analyze the activities by using only basic views of the calendar. Also, it is difficult to keep track of activities without Gantt chart. Difficulty in the understanding of the schedule is increased without the Gantt chart.

Graphical representation of the scheduling of activities is essential for the understanding of the events. So the user interface has much more weight in the calendar component design. Without the marketing calendar, it is difficult to

manage and schedule all the activities and deadlines manually. Also by making the facility of the Gantt chart, it is easy to convey more information than in the textual form.

In this work, we propose the system that schedules the activities according to the resources and also provide the Gantt chart for activities and sub activities planning. We are developing a generalized control which can be used across multiple solutions. This generalized control adds the activity in the calendar and then it is displayed in all the views on the calendar. We are using the SAP OpenUI5 open-source application framework. It is basically a JavaScript library which is based on the model-view-controller (MVC) software design pattern to develop the product. The Gantt chart facility can be added to the product by making the use of existing modules in OpenUI5 and also scheduling of activities is done to reduce the time and to utilize the more resources. Our system provides the basic views along with the Gantt chart and scheduling of the activities. Even though there are many calendar planners available in the market, this product provides all the required functions that any organization or any end user desires.

The rest of the paper is organized as follows. Section 2 gives the brief overview of related work. Section 3 introduces the system. Section 4 introduces the architecture. Finally, a conclusion is presented in the last section.

II. RELATED WORK

Organizations like Microsoft and Google have developed the calendar components, i.e. Microsoft Outlook Calendar[10] and Google Calendar[11] respectively. These calendars provide the facility of creating the calendar activity and schedule it according to user's convenience. We can create, edit, delete and even move the activity using the controls. But they lack in giving the scheduling of resources as per the system requires. There lacks a Gantt chart in both of these calendars. Google calendar is more the user centric rather than the activity-centric such that it doesn't consider the resources required for the activity, but only the start and end time of activity. Therefore the Google calendar cannot be used for the organization's product development schedules, but only can use for user's personal daily schedules. Microsoft Outlook Calendar is effective for official use, but still, it lacks the certain features like this calendar only add activity as a single object which has nothing to do with the current product development. It means an activity is an independent entity which has nothing to do with other activities. Therefore, it is difficult to construct the Gantt chart in such case. Resource allocation is provided the Outlook Calendar on the first come first serve basis.

There are also other calendar component products developed by other organizations and universities. NCSU(North Carolina State University) students, developed the calendar that supports the daily view only but not the other views. And also the estimate for the monthly view is 4 to 5 months. But it doesn't support a Gantt chart view. The web-delicious.com developed the WD Calendar does not support the Gantt chart view, the control is built using PHP, and requires a database table to be set up for the calendar to be used. The MIT developed Frontier Calendar does not support the Gantt chart view. Then bryntum.com develops the Bryntum Calendar develops primarily a Gantt chart calendar but does not collaborate with other views like day, week, or month. It uses Ext JS internally. Then fullcalendar.io developed all the views, but Gantt chart. But the advantage is that it also provides the scheduling of the activities. Also, the fullcalendar provides the dynamic scheduling of the activities. Table-1 shows the gap identification of all the calendar products.

There are many web applications that run on the model-view- controller (MVC) design pattern, but there are some things that lack in the MVC pattern. If we separate all the components of products, then though there is simplicity in understanding the logic, but the debugging of the code becomes very difficult. Also, there are many disadvantages of MVC patterns, i.e. it is not suitable for small applications, MVC affects its performance and design. MVC is bound by some fixed logic, developers cannot cross those limits and work within the boundaries only and the connectors of the MVC pattern are tightly coupled. In 2011, Hamid Mcheick and Yan Qi proposed a proposed a layered model [1] of connectors in distributed MVC so that entire MVC connector becomes loosely coupled. They designed the "connector stack" that is a layered structure consisting of transport, dependent and presentation layer.

Scheduling of resources is also the most important topic in the research field. Resources should be assigned to the machines or systems in an optimized way to reduce the time required for the operations. Scheduling increases the resource utilization by matching supply and demand. There is a scheduling requirement in our proposed work in arranging the room for any activity and schedule the activities in a more optimized way. A work [2] proposed by Zhejun Liu shows the optimal way to arrange the classroom to students in the university. Another work [3] proposes a method of service-oriented-architecture (SOA) it schedules a resource with the help of backspace design space

exploration and greedy algorithms. Another work [4] proposed by the Ching-Ming Hsu, Hui-Mei Chao uses a heuristic driven process coupled with two fitness functions which satisfy the system's needs in consideration. This method takes the system's needs and the number of sizes of resource and assigns a resource to a system.

III. SYSTEM ARCHITECTURE

This section describes the architecture of the calendar component designed using the OpenUI5 application framework [12]. Fig. 1 shows the block diagram of the proposed system. The system has four components basically. The Device, Client (or Browser), Resource Handler and Data.

*These components are discussed in the following sections:*

A. *Device :* Proposed solution works on any device including mobile, tablet, or desktop. One of the properties of OpenUI5 is that it runs on almost any browser and it is responsive to all devices.

B. *Client/Browser:* The browser runs the application that is based on the Model-View-Controller to take the request from the device. When any device sends the request to the client, the view fetches libraries required to perform the application. So the OpenUI5 basic calendar component is fetched first when we load the application and then the appointment component is rendered on that calendar component. Then the Controller performs the computation require to generate the output of the user's request. In our system, then adding the activity on the calendar, perform scheduling of activities is the computational part and it is done in the controller. All title and author details must be in single-column format and must be centered. The model part is used to fetch the data from the database to perform the operations on. This model contains the set of activities in the JSON format which consists of the name of activity, start and end date of an activity, room required for the activity.

C. Resource Handler : Libraries used in the proposed system are sap.m and sap.ui.commons which provide the all necessary components requires to design the application. But sap.m library provides most of the control of the products. It is the leading library for OpenUI5 which can be used for both mobile and desktop applications. And also the already developed components and applications can be fetched from the applications of the OpenUI5. Such that the calendar component is already provided in the OpenUI5 library so we can work on that component to develop our product.

D. Data : Data used by the product is stored in the databaseand it can be accessed by the Gateway server.
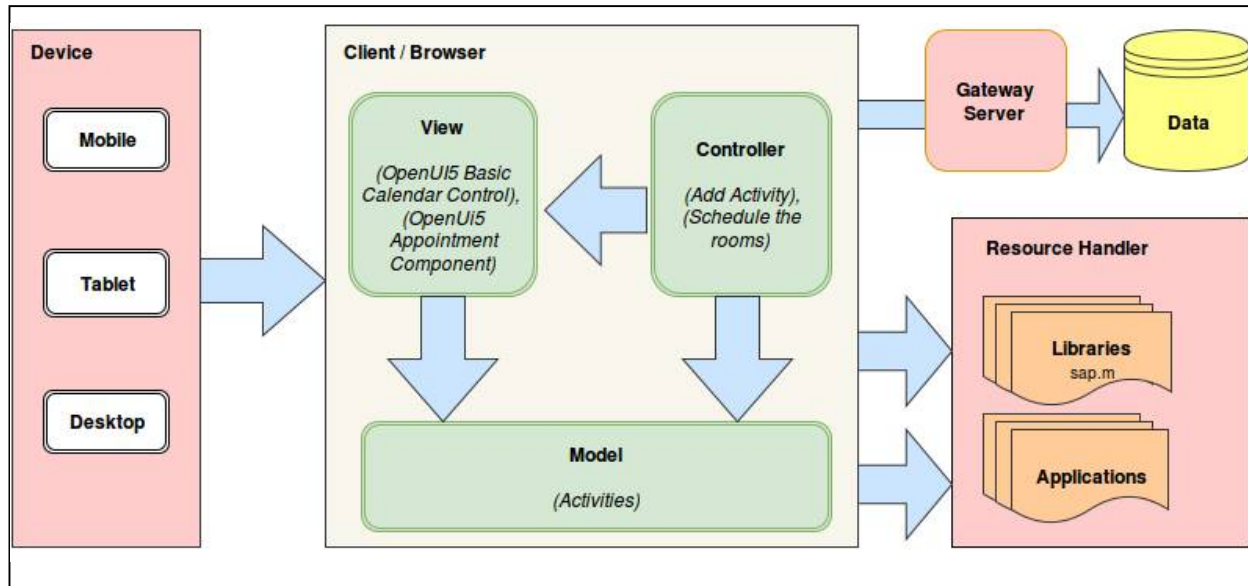
IV. SYSTEM OVERVIEW

The basic architecture of the system is based on the Model-View-Controller software design pattern. This pattern makes a user interface, computation and data part separate from each other. Because of this, the debugging of the application becomes easy and also it is simple to understand the data flow during running time.

In the MVC pattern, the view part is where the user interaction occurs. The request sent by user sent from the view is sent to the controller part. The controller is used to perform the methods invoked by the view. After performing actions on the data, the controller the view and model about data updating. The basic structure of MVC is shown in the Fig. 2. The model stores the activity data that is fetched from the commands from the controller and it is also displayed in the view. And then view contains the basic calendar component which contains the user interface of the calendar and adds activity. The controller can communicate with both model and view. The controller can update the model's state after any updating of the data occurs. And also model can present the updated presentation to the view.
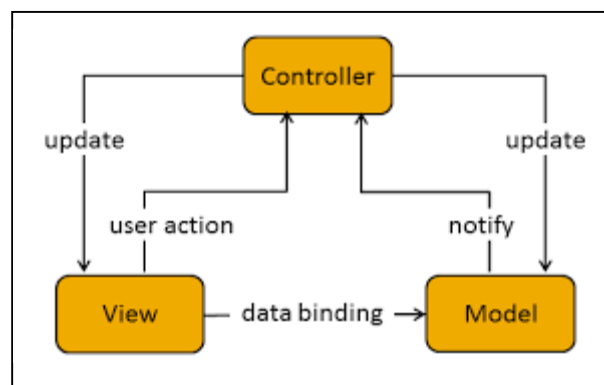
MVC can be also represented as a sequence diagram as shown in Fig. 3. The client can send requests to the view and view converts that request to a function call. That function call is passed to the controller and the controller performs the action. Then after the execution, controller confirms action to the model and model replies with the notify change

event. After controller gets a reply from the model, it replies views with the updated information, and the end user can see through the view. The same analogy is applicable for any MVC pattern model.

V. CONCLUSION

Calendar scheduler helps us to manage and schedule the activities during any product development. Many calendars don't support the Gantt chart view. Also, the scheduling of the resources is also not done in many calendars. Our work makes the use of SAP OpenUI5 application framework for the use of calendar component and the Gantt chart component. This calendar supports the scheduling of the resources and the different views. Our future work includes the render the activity on the calendar control provided by the OpenUI5 and then develops the technique to position the event on the calendar so that it uses the optimal space and also it is re-sizable.

REFERENCES

1. Hamid Mcheick, Yan Qi, Dependency of components in MVC Distributed Architecture, Electrical and Computer Engineering(CCECE), 2011 24th Canadian Conference, 2011.
2. Zhejun Liu, Optimal Model for University Classroom Assignment to Reduce Traffic Volume, Third International Conference on Innovative Computing Technology (INTECH 2013), 2013, pp.375-378.
3. Chao Wang, Aili Wang, Xuehai Zhou, A Classroom Scheduling Service for Smart Classes, IEEE Transactions on Services Computing, 2015.
4. Chin-Ming Hsu, Hui-Mei Chao, A Student-Oriented Class-Course Timetabling Model with the Capabilities of Making Good Use of Student Time, Saving College Budgets and Sharing Departmental Resources Effectively, Intelligent Systems, 2009. CGIS '09. WRI Global Congress, 2009.
5. Yan Hao, Bibo Tu, Jianfeng Zhan, Dan Meng, PWP: a Cluster Web Portal based on MVC, Eighth International Conference on High-Performance Computing, 2005.
6. Shu-qiang Huang, Huan-ming Zhang, Research on Improved MVC Design Pattern Based on Struts and XSL,International Symposium on International Science and Engineering, 2008.
7. Yonghong Chen, Xiwei Xu, Liming Zhu, Web Platform API Design Principles and Service Contract, 19th Asia-Pacific Software Engineering Conference, 2012.
8. Florian Haupt, Frank Leymann, Cesare Pautasso, A conversation based approach for modeling REST APIs, 12th Working IEEE/IFIP Conference on Software Architecture, 2015.
9. Li Li, Wu Chou, Designing Large Scale REST APIs Based on REST Chart, IEEE International Conference on Web Services, 2015. support.office.com, "Introduction to Outlook Calendar", 2016.
10. Online] Available: https://support.office.com/en-us/article/Introduction-to-the-Outlook-Calendar support.google.com, "Calendar Help", 2016. [Online] Available: https://support.google.com/calendar
11. opeui5.hana.ondemand.com, "Essentials of OpenUI5", 2016. [Online] Available: https://openui5.org/features.html
12. sap.com, "Model View Controller(MVC)", 2016. [Online] Available: https://help.sap.com/saphelpn uiaddon10/helpdata/en/content.htm