

Activity Planner Using SAP OpenUI5 Application Framework

Abstract

Aims/ objectives: To design a calendar component for scheduling activities which reduces the space required for representation. Design a Gantt chart with hierarchical activities which gives better visualization and understanding.

Study design: Cross-sectional study.

Place and Duration of Study: Department of Computer Engineering, Pune Institute of Computer Technology, Pune, Maharashtra, India, between July 2016 and June 2017.

Methodology: We studied different calendar components available in the market including Microsoft Outlook Calendar and Google Calendar and identified drawbacks of them and then develop a generalized calendar control which schedules activities in the calendar and also develop the Gantt chart for activities by using the SAP OpenUI5 application framework. Also the classroom scheduling algorithm is included to schedule the activities on the activity planner. The proposed algorithm reduces the time for scheduling by taking the number of rooms in the classroom scheduling algorithm as constant, and only taking the number of activities as the input to the algorithm.

Results: Proposed calendar component schedules activities by using the classroom scheduling algorithm. Also using the OpenUI5 framework, calendar component can be designed which provides elegant and user friendly interface. Also the Gantt chart is designed with hierarchy of events which makes easy understanding of activities.

Conclusion: 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.

Keywords: Calendar Component, Gantt chart, MVC, SAP OpenUI5

2010 Mathematics Subject Classification: 53C25; 83C05; 57N16

1 Introduction

Planner or scheduler is used to make the easy management of activities. Planner arranges the activities in the form of graphics that enables the user to understand activities. Generally planners use the time parameter to display the activities and the relations among them. Calendar is one of the most effective tools used for planning purposes. Calendar is one of the most effective tools for planning purposes. Calendar supports different views according to the duration like daily view, weekly view, monthly view, yearly view, etc. We can get a different perspective of activity from different views. Therefore calendars are used for the planning or scheduling purpose. The planners are not new, as people are using them for long time.

Many planners are available in the market those provide solutions to the scheduling problems. But they lack certain features like Gantt chart or management of activities which makes them tedious to use. Great user interface and minimum used space make users, attract to the application. Scheduling is the integral part of any planner which schedules activities in a more efficient way. But many calendar planners dont provide these functionalities. Also, many calendar planners dont provide the Gantt chart.

Without Gantt chart, it is difficult to keep track of activities and fails to provide basic understanding of the schedule. Graphical representation of the scheduling of the activities is essential for understanding of events. So user interface has much more weight in the calendar component design. Without the activity planner, 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. Therefore, it is needed to develop the calendar activity planner with Gantt chart which is much more space efficient and has a user friendly interface.

Schedulers or planners have a large number of applications in the industries, manufacturing processes and also in the universities. So the greater user interface and optimized scheduling become the application useful. Also the Gantt chart can be used to track the completed and remaining activities. Using analysis of the Gantt chart, organizations can decide the method of work that needs to be done to complete the work within deadlines.

In the proposed work, the calendar component uses the classroom scheduling algorithm to schedule the activities in the calendar. By taking the start and end time of the activity and the enrollment of the activity we can schedule an activity in the calendar. In this algorithm, different enrollment for different rooms are used to schedule activities according to the capacities. Also collision in the activity schedule is resolved by assigning the activity to the next higher enrolled available room. The proposed work makes use of SAP OpenUI5 application framework to develop the calendar component which provides the rich libraries of the user interface to develop the user friendly application. SAP OpenUI5 is based on the model-view-controller (MVC) software design pattern to develop the product. Our system provides the basic views, including the hourly, daily, weekly and monthly view. The proposed work aims at providing the Gantt chart to show the scheduling, because the Gantt chart helps to set realistic time frames and makes it easy to understand the scheduling sequence. For evaluation, the system under consideration is compared with the other calendar components and using the Gantt chart, we can easily trace the activity and the schedule of the activity.

1.1 Literature Survey

Organizations like Microsoft (14) and Google (15) have developed the calendar components, i.e. Microsoft Outlook Calendar and Google Calendar respectively. These calendars provide facility of

creating the calendar activity and schedule it according to users' convenience. We can create, edit, delete and even move the activity using the controls. But they lack in scheduling of resources as per the system requires. They also lack Gantt chart. Google calendar is more user centric rather than activity centric such that it doesn't consider the resources required for the activity but only start and end date of the activity. Therefore, Google calendar cannot be used for the product development schedules, but can only use for user's personal daily schedules. Microsoft Outlook Calendar is useful for official use but still it lacks the certain features like it only adds activity as single object which has nothing to do with the current product development. It means an activity is the independent entity has nothing to do with other activities. Therefore, it is difficult to construct the Gantt chart in such case. Resource allocation is provided in 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 does not support a Gantt chart view. 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 institute developed Frontier Calendar does not support the Gantt chart view. Then bryntum.com develops the Bryntum Calendar (18) 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 fullcalendar (19) provides the dynamic scheduling of the activities. Table 1 shows the gap identification of all the calendar products.

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 proposed work for arranging the room for any activity and schedule the activities in a more optimized way. A work (3) proposed by Zhejun Liu shows the optimal way to arrange the classroom to students in the university. Another work (1) 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 systems needs in consideration. This method takes the systems needs and the number of sizes of resource and assigns a resource to a system.

There are many web applications that run on the model-view-controller (MVC) design pattern, but there are some things that lacks 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 (11) of connectors in distributed MVC so that entire MVC connector becomes loosely coupled. they designed the "connector stack" that is layered structure consisting of transport, dependent and presentation layer.

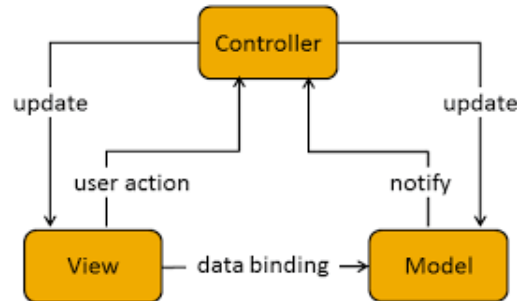
2 Methodology

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

Table 1: Gap Identification of Calendar Components

Sr. No.	Product	Developed By	Drawbacks
1	Microsoft Outlook	Microsoft Corporation	supports schedule view but does not support Gantt chart view and hierarchy of activities.
2	Google Calendar	Google	Doesn't support Gantt Chart View.
3	NCSU Calendar	North Carolina State University	supports daily view only, and implementation estimate for the monthly view is 4 to 5 months. Doesn't support Gantt chart view.
4	WD Calendar	web-delicious.com	Does not support Gantt chart view, the control is built using PHP, and this requires a database table to be setup for the calendar to be used.
5	Frontier Calendar	MIT	Does not support Gantt Chart view.
6	Bryntum Calendar	bryntum.com	Primarily a Gantt chart calendar but does not collaborate with other views like day, week or month. It uses Ext JS internally.

Figure 1: MVC Architecture



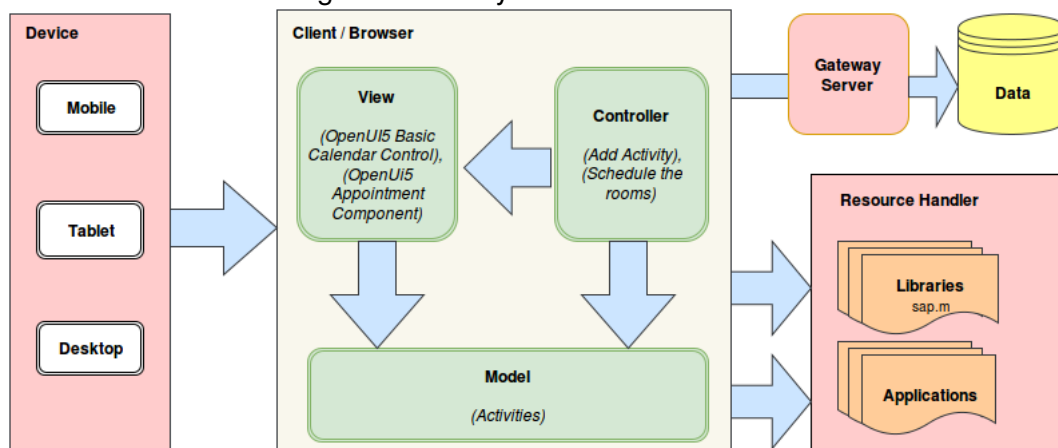
request sent by user sent from the view is sent to the controller part. The controller is used to perform the methods invoked by the view. After performing actions on the data, the controller the view and model about data updating. The basic structure of MVC is shown in the Figure 1. The model stores 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 models state after any updating of the data occurs. And also model can present the updated presentation to the view.

2.1 System Architecture

First, let us look at the System Architecture of activity planner and discuss the relevant design aspects. Then the scheduling algorithm is discussed.

The architecture diagram is given in the figure 2. As indicated in the architecture diagram, 4 basic components are there are device, client/browser, data and resource handler. The device component may have the devices like mobile, tablet, desktop which are used to show the activity planner. The supporting routines are embedded in this module. The device will interact with the client/browser component. The client/browser component is facilitated with the MVC architecture

Figure 2: Activity Planner Architecture



pattern. The model is the central component of the pattern, which directly manages the data, logic and rules of the application. In Activity Planner, the data considered is set of activities. The view is the output representation of the information, for which activity planner provides the OpenUI5 calendar component and the rendered activity component. The controller accepts the input and converts it into the commands for the model and view.

The data used here consists of set of activities stored in JSON format and can be accessed by creating the JSON object in the view. And finally resource handler provides the libraries in the proposed system which are sap.m and sap.ui.commons which provide all necessary controls require to design the application.

To schedule the activities in the activity planner, proposed solution uses the classroom scheduling algorithm. In this algorithm, inputs are startTime, endTime and enrollment of the activity to be added as the input and the allocatedRoom as the output. Input also contains the set of start, end dates of the activities and set of room allocated to that activity. This algorithm first checks the enrollment of the activity to be added and try to allocate the suitable room by checking the capacity of the each room. If that room is not available, then it allocates the next highest possible room to the activity. If all the rooms are unavailable at that time, then it throws the error of activity collision. The classroom scheduling is described on the page 106

This algorithm assigns the room for added activity in the Activity Planner in the $O(n)$ time where 'n' is the number of activities already added.

In this algorithm, inputs are startTime, endTime and enrollment of the activity to be added as the input and the allocatedRoom as the output. Input also contains the set of start, end dates of the activities and set of room allocated to that activities. This algorithm first checks the enrollment of the activity to be added and try to allocate the suitable room by checking the capacity of the each room. If that room is not available then it allocates the next highest possible room to the activity. If all the rooms are unavailable at that time, then it throws the error of activity collision.

This algorithm assigns the room for added activity in the Activity Planner in the $O(n)$ time where 'n' is the number of activities already added.

3 Results and Discussion

Activity Planner uses the Classroom Scheduling algorithm which runs in the $O(m * n)$ time where, m is a number of classes and n is the number of activities. We can consider the constant number of classes to reduce the complexity of the algorithm, from which the algorithm runs in the $O(n)$ time only.

3.1 Performance Testing with Graph

Performance of Activity Planner is checked with the help of timeline facility of the Chromium Browser in Ubuntu 16.04 LTS. The load time for Activity Planner is shown as a Bar Graph in figure 3. Here, all the loading files are HTML files, scripting files are under section 'Scripting'. All the media files are under 'Rendering'. Stylesheets are under 'Painting' and all the miscellaneous resources comes under 'other' section. In this figure, total time taken by the Activity Planner to load is **2.88 seconds** on localhost and same for the Google Calendar is **3.67 seconds** online. Out of which the largest time is taken by the scripting of application in both cases. The percentage time taken for page loading operations for each Activity Planner and Google Calendar is shown in the table 2. 80% of total load time is spent on the scripting of the application which processes all the JavaScript involved in the application in case of Activity Planner. But in case of Google Calendar, it spends only 58.11%. Webpage remains idle in case any script files require any action from the browser about frequent updating of content. But in case of OpenUI5, idle time is less compared to Google Calendar. As

Algorithm 1: Classroom Scheduling Algorithm

Input : $startTime, endTime, enrollment, room1 = 25, room2 = 50, room3 = 75, room4 = 100$

Output: $allocatedRoom$

Set : $\{st\}, \{en\}, \{roomsAllocated\}$

```

1 if  $enrollment < 1$  or  $enrollment < 100$  then
2   |  $allocatedRoom \leftarrow \emptyset$ 
3 else if  $enrollment \leq room1$  then
4   |  $allocate(1)$ 
5 else if  $enrollment \leq room2$  then
6   |  $allocate(2)$ 
7 else if  $enrollment \leq room3$  then
8   |  $allocate(3)$ 
9 else if  $enrollment \leq room4$  then
10  |  $allocate(4)$ 
11 Procedure ( $allocate(roomNo)$ )
12 for  $index = 1, 2, 3, \dots$  in  $st$  do
13   | if  $roomsAllocated_{index} = roomNo$  then
14     | if ( $startTime < st_{index}$  and  $endTime < en_{index}$ )
15       | or ( $startTime > st_{index}$  and  $endTime > en_{index}$ ) then
16         |  $allocatedRoom \leftarrow roomNo$ 
17         |  $st \leftarrow st \cup startTime$ 
18         |  $en \leftarrow en \cup endTime$ 
19         |  $roomsAllocated \leftarrow roomsAllocated \cup allocatedRoom$ 
20       | end
21     | else if  $roomNo \neq 4$  then
22       |  $allocate(roomNo + 1)$ 
23     | else
24       |  $allocatedRoom \leftarrow \emptyset$ 
25     | end
26 end

```

Figure 3: Load Time of Activity Planner

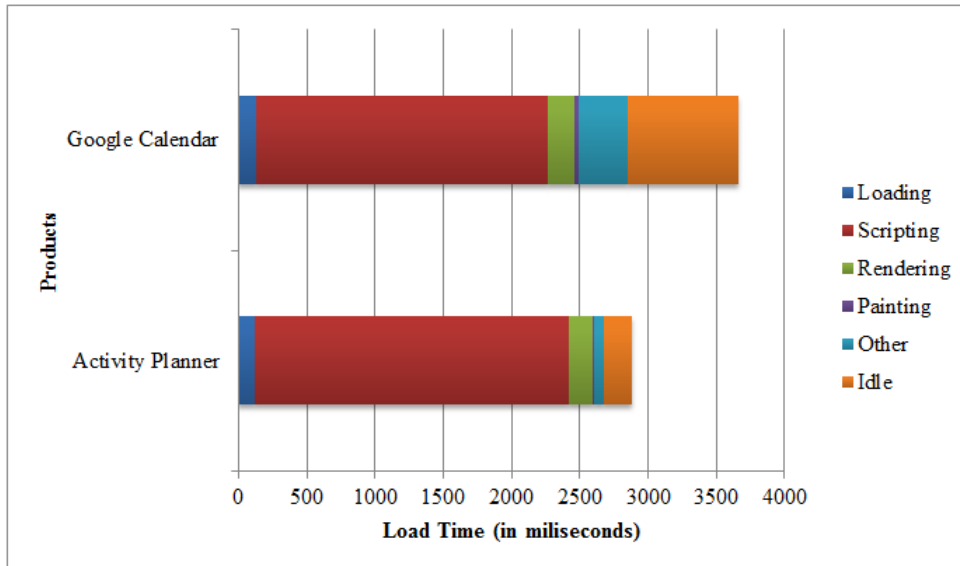


Table 2: Performance Evaluation

	Activity Planner	Google Calendar
Loading	4.19%	3.53%
Scripting	80%	58.11%
Rendering	5%	5.43%
Painting	0.2%	0.9%
Other	2.75%	9.82%
Idle	7.06%	22.13%

OpenUI5 uses the most of the user interface as a part of the web page, it takes less time for loading as compared with the Google Calendar.

The miscellaneous files are generated very less in Activity Planner as compared with the Google Calendar which reduces the time for loading of the application. So, as OpenUI5 reduces the painting workload, but it has a disadvantage for the scripting time. As OpenUI5 involves a large amount of JavaScript libraries, it takes nearly more than three forth of the total loading time that of Google Calendar.

4 CONCLUSIONS

Conculsions by developing the activity planner is as follows:

- a Many calendars in the market does not provide the combined views of all the activities and also the Gantt chart including the Microsoft Outlook and Google Calendar.

- b** We have developed the Calendar Activity Planner using the OpenUI5 framework which makes the use of javascript libraries to develop the Calendar control and also we can plot the Gantt chart.
- c** We have also added the classroom scheduling algorithm to schedule the activity in the calendar.
- d** By sorting the activities in the runtime we can achieve optimal space required for the activities for display purpose in the planner.
- e** Also by adding Gantt chart, user can understand the activity sequence and the hierarchy of the activities.

References

- [1] Chao Wang, Aili Wang, Xuehai Zhou, "A Classroom Scheduling Service for Smart Classes" in *IEEE Transactions on Services Computing*, 2015. p. 155-164.
- [2] Jaime Raigoza, Rushi Thakkar, "Browser Performance of JavaScript Framework, SAPUI5 & jQuery" in *International Conference on Computational Science and Computational Intelligence*, 2016. p. 1420-1421.
- [3] Zhejun Liu, "Optimal model for university classroom assignment to reduce traffic volume" in *Innovative Computing Technology (INTECH), 2013 Third International Conference*, 2013. p. 375-378.
- [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" in *Intelligent Systems, GCIS'09. WRI Global Congress*, 2009. p. 379-384.
- [5] Yan Hao, Bibo Tu, Jianfeng Zhan, Dan Meng, "PWP: a cluster Web portal based on MVC" in *Eighth International Conference on High-Performance Computing in Asia-Pacific Region*, 2005. p. 239-243.
- [6] Shu-qiang Huang, Huan-ming Zhang, "Research on Improved MVC Design Pattern Based on Struts and XSL", in *International Symposium on Information Science and Engineering*, 2008, pp. 451-455.
- [7] Yonghong Chen, Xiwei Xu, Liming Zhu, "Web Platform API Design Principles and Service Contract", in *19th Asia-Pacific Software Engineering Conference*, 2012. p. 877-886.

- [8] Florian Haupt, Frank Leymann, Cesare Pautasso, "A Conversation Based Approach for Modeling REST APIs", in *12th Working IEEE/IFIP Conference on Software Architecture*, 2015. p. 165-174.
- [9] Li Li, Wu Chou, "Designing Large Scale REST APIs Based on REST Chart", in *IEEE International Conference on Web Services*, 2015, p. 631-638.
- [10] S. H. Maes, R. Hosn, J. Kleindienst, T. Macek, T. V. Raman, L. Seredl, "A DOM-based MVC multi-modal e-business" in *IEEE International Conference on Multimedia and Expo*, 2001, p. 897-900.
- [11] Hamid Mcheick, Yan Qi, "Dependency of components in MVC distributed architecture", in *24th Canadian Conference on Electrical and Computer Engineering(CCECE)*, 2011, p. 691-694.
- [12] Frolin S. Oscariza, Karthik Pattabiraman, Ali Mesbah, "Detecting Inconsistencies in JavaScript MVC Applications", in *IEEE/ACM 37th IEEE International Conference on Software Engineering*, 2015, p. 325-335.
- [13] Yanfang Wang, Chunyan Guo, Lei Song, "Architecture of E-COMmerce Systems Based on J2EE and MVC Pattern", in *International Conference on Management of e-Commerce and e-Government*, 2009, p. 284-287.
- [14] support.office.com, "Introduction to Outlook Calendar", 2017. [Online] Available: <https://support.office.com/en-us/article/Introduction-to-the-Outlook-Calendar-bc7ce542-72a7-4bc5-8c59-5bb5700620e4>
- [15] support.google.com, "Calendar Help", 2017. [Online] Available: <https://support.google.com/calendar>
- [16] opeui5.hana.ondemand.com, "Essentials of OpenUI5", 2017. [Online] Available: <https://openui5.org/features.html>
- [17] tutorialspoint.com, "MVC Architecture", 2017. [Online] Available: https://www.tutorialspoint.com/struts_2/basic_mvc_architecture.htm
- [18] bryntum.com, "Bryntum Calendars Demo", 2017. [Online] Available: <https://www.bryntum.com/examples/gantt-latest/examples/calendars/>
- [19] fullcalendar.io, "FullCalendar-JavaScript Event Calendar", 2017. [Online] Available: <https://fullcalendar.io/>