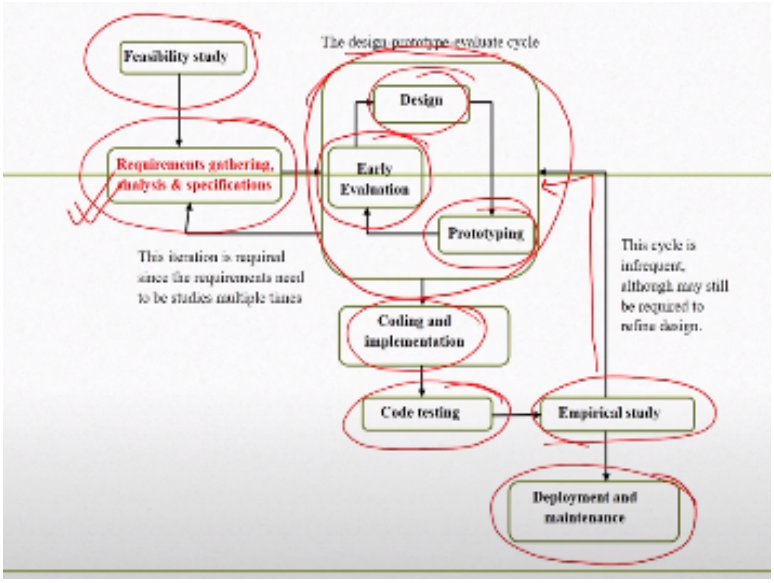
Design & Implementation of Human – Computer Interfaces
Prof. Dr. Samit Bhattacharya
Department of Computer Science & Engineering
Indian Institute of Technology, Guwahati

Module No # 02
Lecture No # 10
Case Study (Other Requirement Gathering)

Hello and welcome to NPTEL MOOCS course on design and implementation of human computer interfaces we are going to continue with the case study in this lecture number 9. So we started the case study in the previous lecture on an application development and in this lecture we are going to continue with the case study. So before we go further we will first have a look at where we are in this course.

So we started discussion on interactive software development life cycle. As repeatedly we have discussed there are many stages which form the life cycle. These stages are performed in a particular sequence and sometimes the stages form a loop.

(Refer Slide Time: 01:55)



So if you may recollect the stages first one is feasibility study which anyway we are not focusing, on second is requirement gathering analysis and specification, third, is basically a set of sub stages performed in a loop manner. So the sub stages are design sub stage, prototyping sub

stage, and quick evaluation of prototyping that is another sub stage. So together they form a cycle which we are calling the design prototype evaluates cycle.

So through this cycle we do 2 things one is we design the interface as well as interaction and also we design the system. Now once the system is designed then we go to the coding and implementation stages this is followed by the testing stage where we test the code or the program. Another testing stage follows which is empirical user testing or usability testing through this testing we get to know usability of the end product.

And occasionally it may lead to returning back to this cycle design prototype evaluation cycle if we find some problems in the usability testing phase but that is typically very infrequent. And once we are certain about the efficacy of the code as well as usability of the product we go for the next test that is that is deployment and maintenance. So among these stages we are currently covering the requirement, gathering, analysis and specification stage.

We have already learned what we mean by requirement gathering analysis and specification. To recap so there are 2 things involved one is requirement for whom and then how to specify those requirements? First question is for whom we are collecting the requirement? In typical software development process requirements can be obtained through interviews with clients or customers as well as based on experience of the designers.

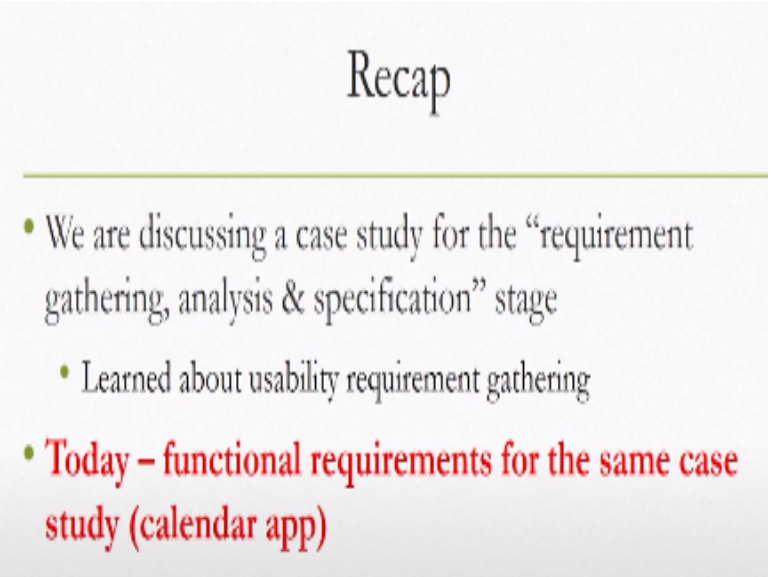
However in this case along with requirements provided by the clients we also need to know the requirements of the actual end users. So here we are distinguishing between the clients and the end users. End users are layman users who need not have any technical expertise but they just want to use the system for achieving certain goals. So when we said requirement gathering so we focused on these 2 aspects.

First of all to gather requirements from end users which is likely to lead to usable systems these requirements we are referring to as usability requirements. To gather those usability requirements what we did we have gone through one particular method called contextual inquiry method. So if we follow this approach and implement it then we are likely to find out some requirements that lead to better usability.

Other thing is so usability requirements are nonfunctional requirements from the point of view of a system developer there is another category of requirements which typically comes through interaction with clients as well as through prior experience of the designer.

Now these are called functional requirements where we actually get to know the requirements and get to specify those requirements in the form of functions with clearly defined input, output and some optional description of the functions. We have briefly looked at what these functions are how they can be specified and what they refer to in the context of a system development.

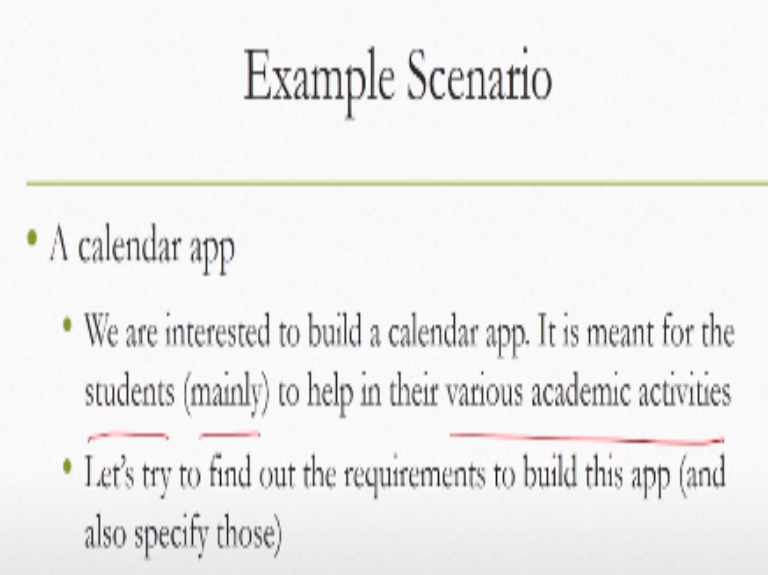
(Refer Slide Time: 06:48)



Now let us come back to our case study so we are currently discussing on a case study pertaining to the development of a calendar application or app. Now this application is primarily meant to be developed for students who are expected to use it for academic purpose. So our user group is fixed students more specifically college going students or students enrolled for higher studies. And the work context is also fixed that is academic environments which include classrooms which includes PR discussion groups and so on.

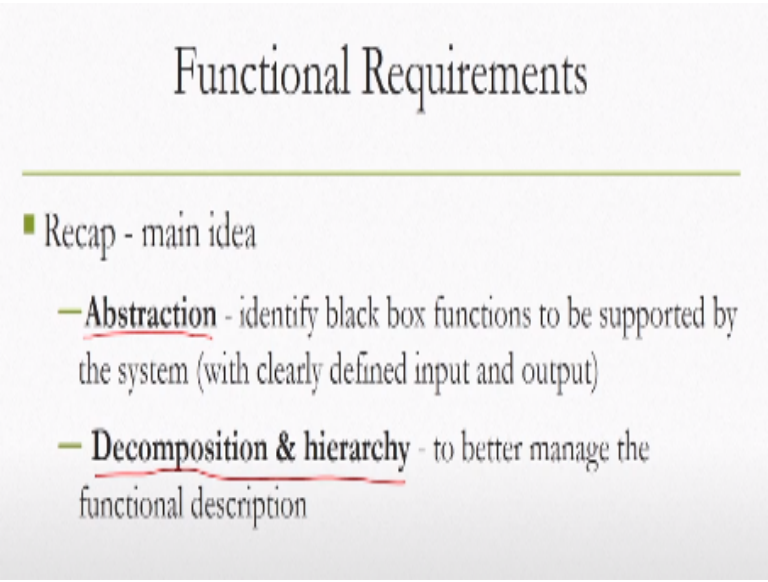
And if these are specified then what we are discussing is? How to find out the requirements both non-functional and functional requirements? So in the previous lecture we have discussed how to find out the usability requirements through contextual inquiry. In this lecture we will try to see how to find out functional requirements which can be specified in the form of functions.

(Refer Slide Time: 08:20)



So calendar app is our example application development scenario and as I said this meant for students primarily students enrolled for higher studies in colleges or universities and the app is meant to help them in their various academic activities. So their work setting is also specified given this specifications let us try to find out what are the functional requirements for this particular application?

(Refer Slide Time: 09: 02)



Now before we start discussion on functional requirement let us quickly review the basic ideas. So when we are talking of functional requirements essentially we are trying to find out functions with specific input, output and description. Now these functions are not to be specified in details

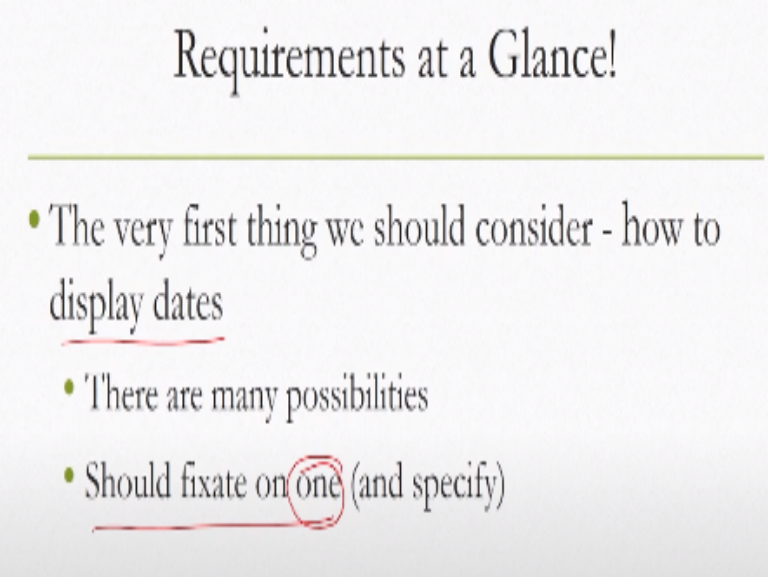
instead we will assume that the functions are black boxes whatever input is given somehow it is processed and generates the desired output.

So we will assume that that output is generated the function is designed in such a way rather than focusing on how to design the function so that the output is generated from the input. So this black box assumption of function is what is called abstract way of looking at the function or abstraction. So this abstraction is very important we think of the functions in a very abstract way.

Second thing is decomposition and hierarchy so when we are talking of functions we should not think of complex functions instead we try to divide functions into sub functions up to the level wherever we feel it necessary. And we should strive to create a hierarchy of such functions and sub functions. As we have seen in one of the earlier lectures that hierarchy actually makes things easier to understand and implement in subsequent stages.

So decomposition of a function into sub functions and creating a hierarchy is very much desirable along with the way to conceive the functions or the abstractions. So both are important and should be kept in mind while we are going to find out functional requirements.

(Refer Slide Time: 11:01)

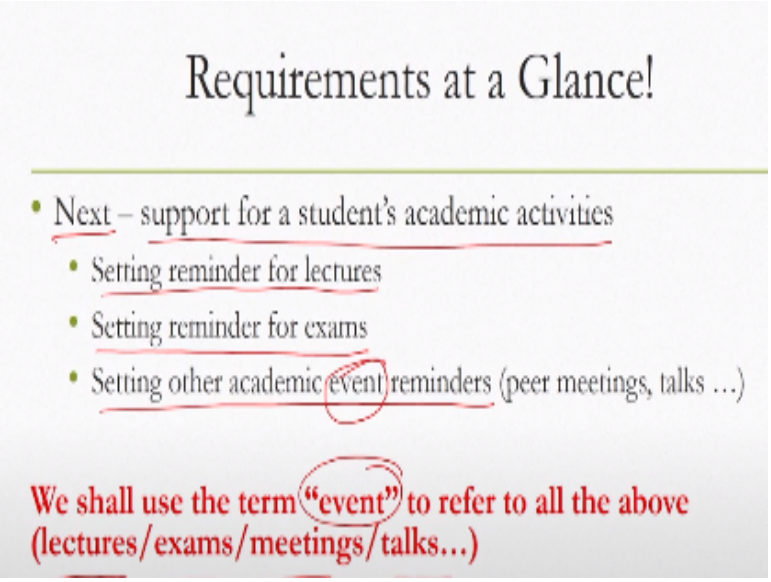


Now let us see what can be the functional requirements for this particular app. So we will here rely on our intuition although we may think of this intuition as coming from interview with clients or customers who have told us to build the system. It may also come from the past

experience of the developer or the development team. So whatever be the case we can still identify few functions for this particular first of all the app should display dates in specific manner.

So; first requirement is how to display the dates in the calendar app which is suitable for specified user group in specified work setting. Now there are many possibilities in which we can display these dates. Our objective is to find out one of those possibilities and then specify that particular possibility.

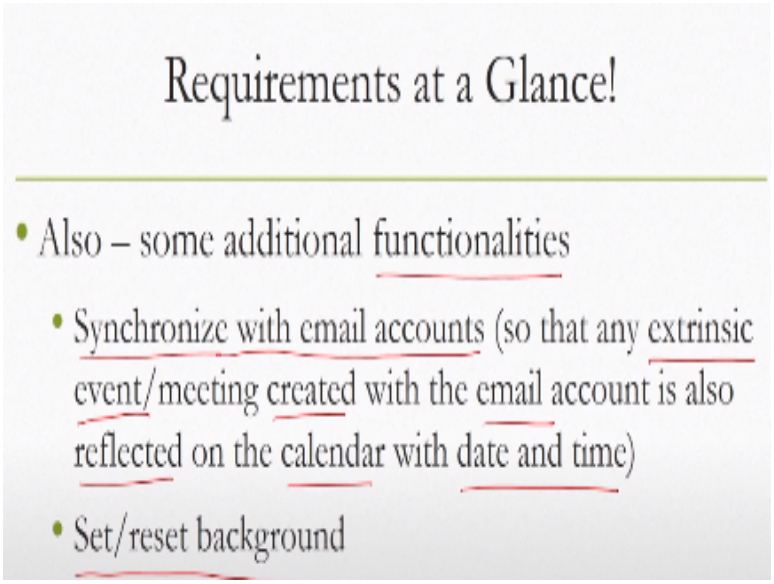
(Refer Slide Time: 12:18)



So we will see how it can be done next obvious requirement is support for a student's academic activities. Now this may include setting reminders for lectures, setting reminder for exams, setting other academic event reminders. Such as peer meetings, talk's seminars so on and so forth now here the term event should be noted carefully. In subsequent part of this lecture I am going to use the term event to actually indicate a host of things including lectures exams meetings talks.

So everything will be termed as event without any further disambiguation and that is good enough for our purpose.

(Refer Slide Time: 13:16)

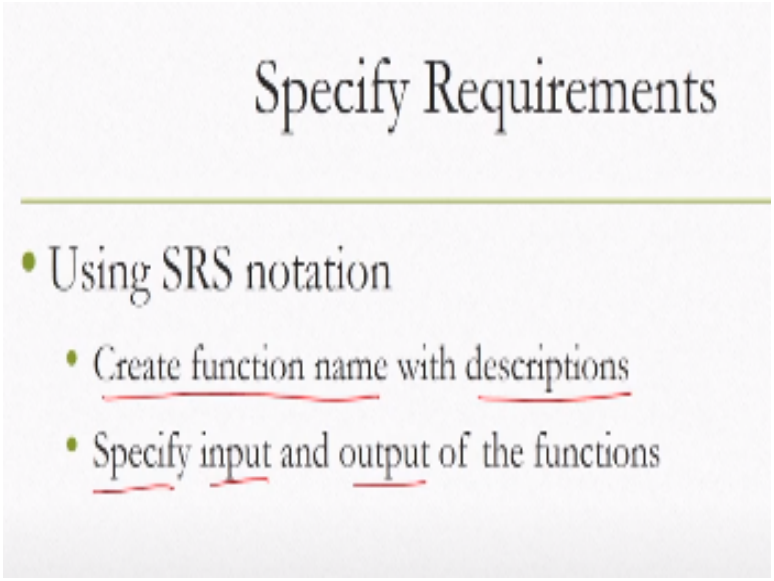


Now these are some obvious requirements there may be some other requirements or some additional functionalities. Synchronization with email may be one such functionality so in this functionality what is required is synchronize with email accounts of the user. So that if there is any extrinsic event created with the e-mail account then it is automatically reflected on the calendar with specified date and time.

So even if you are not explicitly using the calendar but saved some specific events with your email then the app should be able to fetch that information from account and then display it on the calendar. There may be another cosmetic function that is set or reset the application background for aesthetic purpose. So user may be given the freedom to set or reset the background so that is another requirement that we can think of.

So once the requirements are identified our next task is to specify them. Now for specification earlier we have seen that we can use a notation for specification and the resulting document that we get is called SRS or software requirement specification document.

(Refer Slide Time: 14:56)

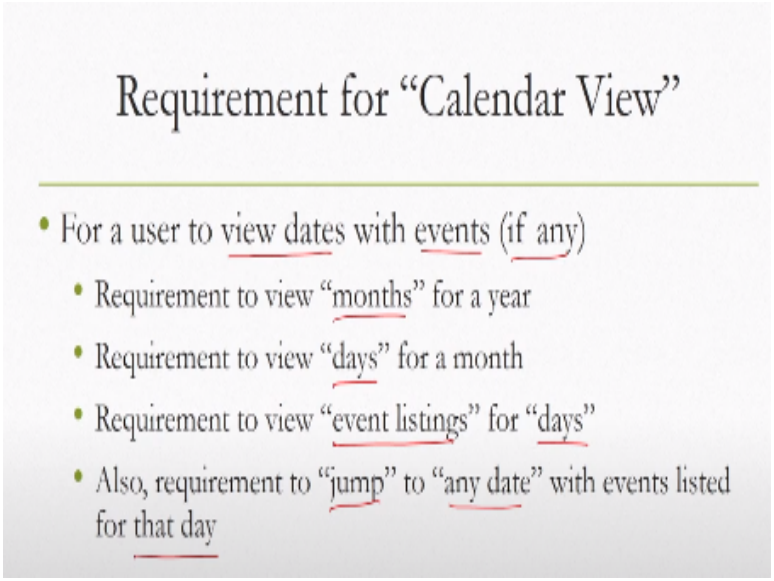


So the notation involves creation of function name with description and clearly specifying the input and output of the functions. So for each function we need to do these things clearly specifies input, output as also provides some name and description. So; for the functions that we have identified for the calendar app. Let us see how we can create the SRS document so first we will focus on how to specify the functions that we have identified.

(Refer Slide Time: 15:37)

So from this specification we will get the SRS document for the calendar app now let us try to see how it is or how it can be done?

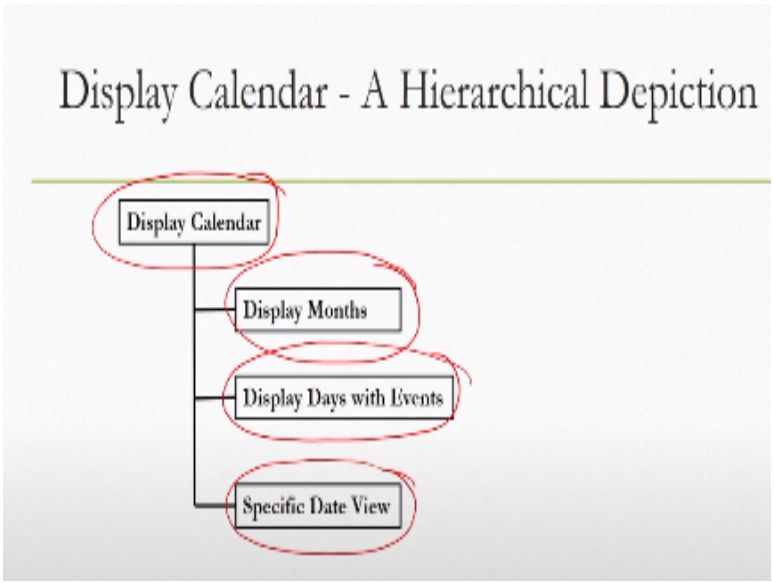
(Refer Slide Time: 15: 48)



So we will first have a look at requirements earlier we have discussed it informally let us now formally discuss the requirements. Now for the requirement for a user to view dates with events if there are any such events for the specific date. Then we can have further requirements like requirement to view months, for a year requirement, to view days for a month and requirement to view event listings for days for each day or a special chosen day what are the events that are listed against it.

We may also need to jump to any date from the current date with events listed for that day so this may be one additional feature or function that should be available with our calendar app.

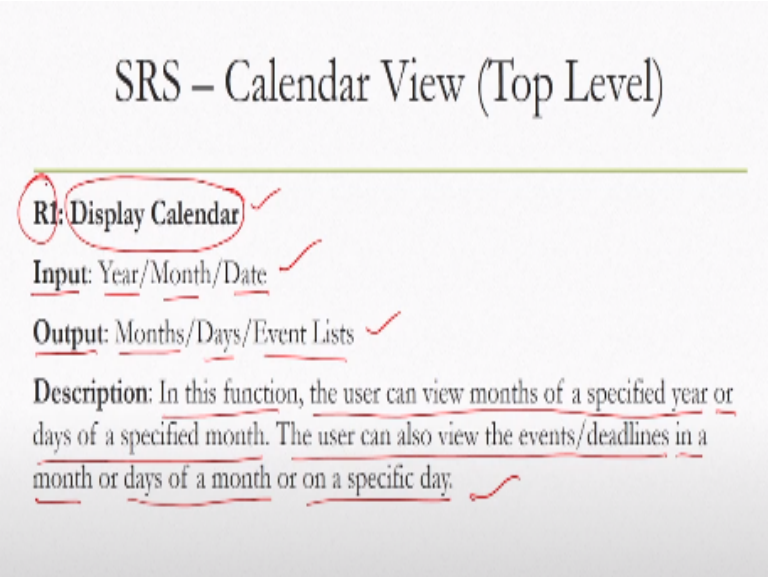
(Refer Slide Time: 16:59)



So accordingly we can create a functional hierarchy for all these requirements. So at the top level of the hierarchy we have display calendar function which is the top level function then under it comes sub function display months, display days with events and specific date view that is the last requirement jumping to any date. So if we jump how to display that particular date note here the abstraction.

So we still have not discussed anything on how to implement these functions rather we are only focusing on what are the functions and how they can be organized in the form of a hierarchy.

(Refer Slide Time: 17:54)

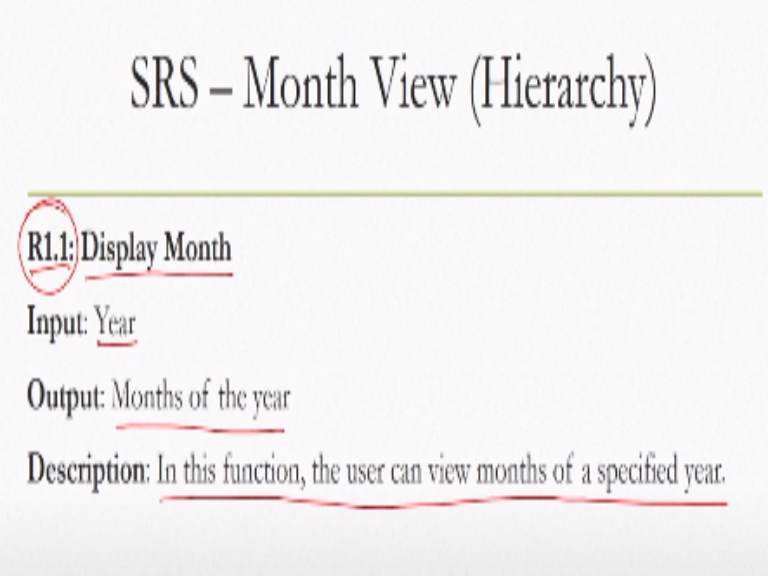


Now let us see how we can specify it? So since display calendar is the top level function. So we give it a name display calendar along with this label showing the level of the hierarchy. And for this top level function we provide an input that is year month or date as input and there is an output which is specified which is corresponding months days or lists of events for a day. So if we are inputting on then the output will be the months of the year.

If the input is month then the output then the output will be days of the month if the input is a day then the output will be the list of events for that day. There can be a description also but as I said earlier this can be optional but ideally it should be there to clarify the purpose of the functions. For example in this case we can have a description like in this function the user can view months of a specified year or days of a specified month.

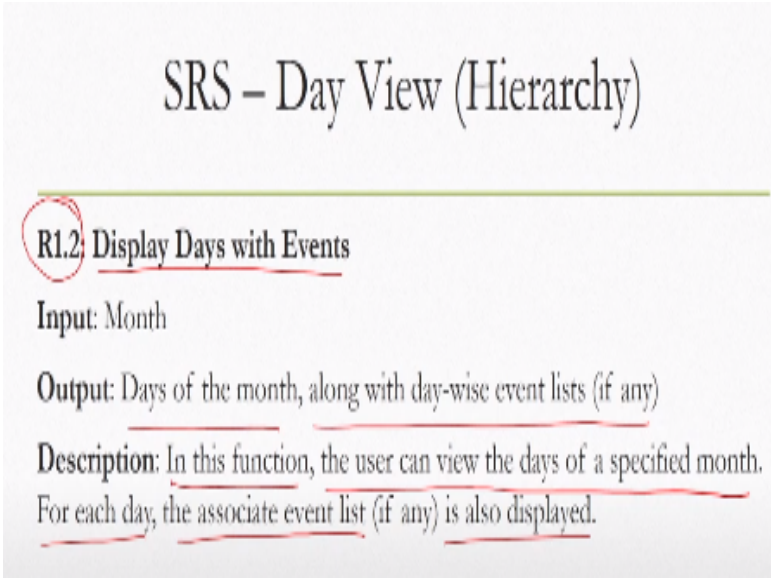
The user can also view the events deadlines in a month or days of a month or on a specific day. So this is the purpose of the function why we are designing it to serve this purpose so description gives us that idea of the purpose. So here we have all these components name, input, output and description and note that name is having a label this label is representing the level of the hierarchy.

(Refer Slide Time: 20:06)



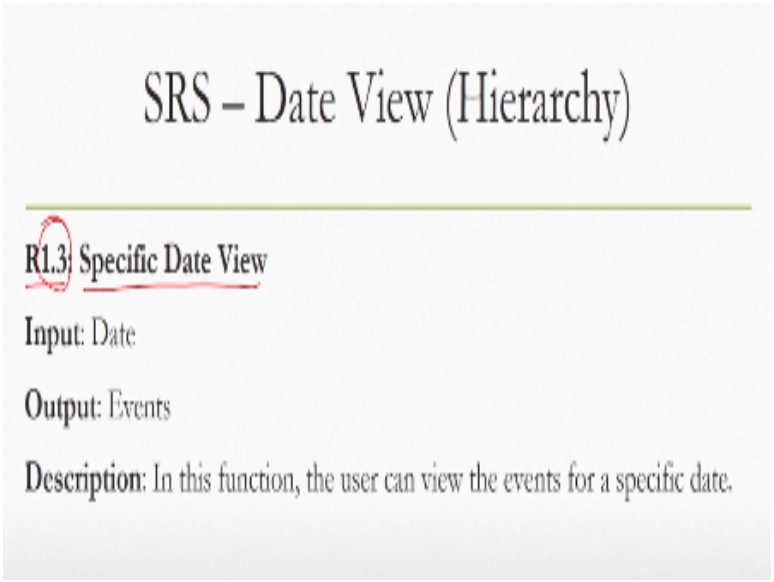
Now below the top level function there are 3 sub functions that we have mentioned in the hierarchy. First sub function is display month so again here we have one name with label which indicates the level of the hierarchy. Since this comes under display calendar function so we are giving it a level 1.1 that means under 1 this is the first sub function. So in this case the input is year, the output is months of the year and description is in this function the user can view months of a specified year that is the description so it is as simple as that.

(Refer Slide Time: 21:18)



Then we will move to the next sub function that is display days with events. Since; it is the second sub function as per our hierarchical display so we are giving it a level 1.2. Now in this case input is month and output is days of the month along with day wise event lists if there are any such events. Description in this function the user can view the days of a specified month for each day the associated event list is also displayed if there is any such event so this is our second sub function in the hierarchy.

(Refer Slide Time: 22:11)

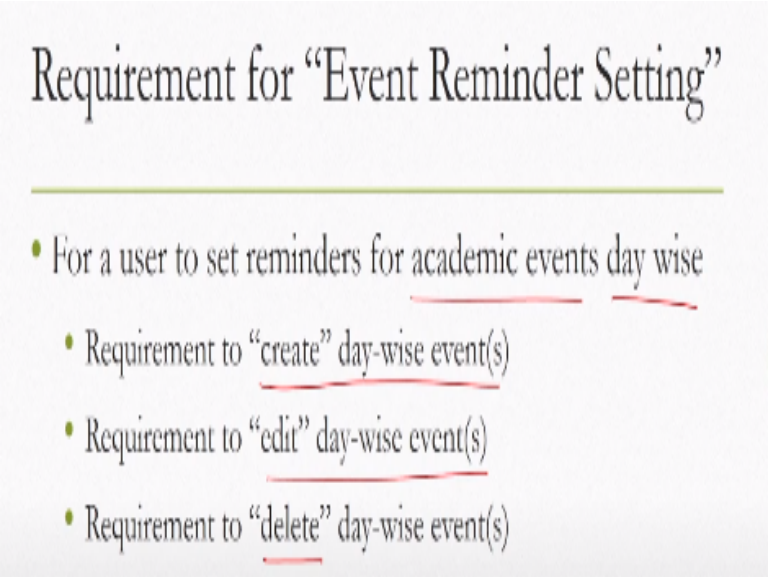


Now there is third sub function accordingly we have given it a level 1.3 and the name of the sub function is specific date view as usual. So under this sub function we will have input output and

description. Let us see what is the input? What is the output? And what is the description for this sub function? Input is date, output is events and description is in this function the user can view the events for a specific date.

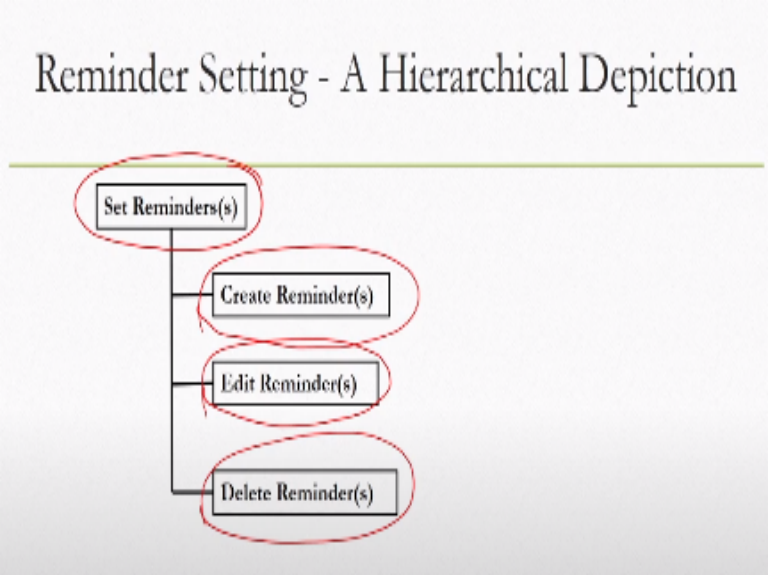
So if there is no event then it will not show anything if there are events then the list of events will be shown. So that was about one requirement let us move to the next requirement for event reminder setting.

(Refer Slide Time: 23:12)



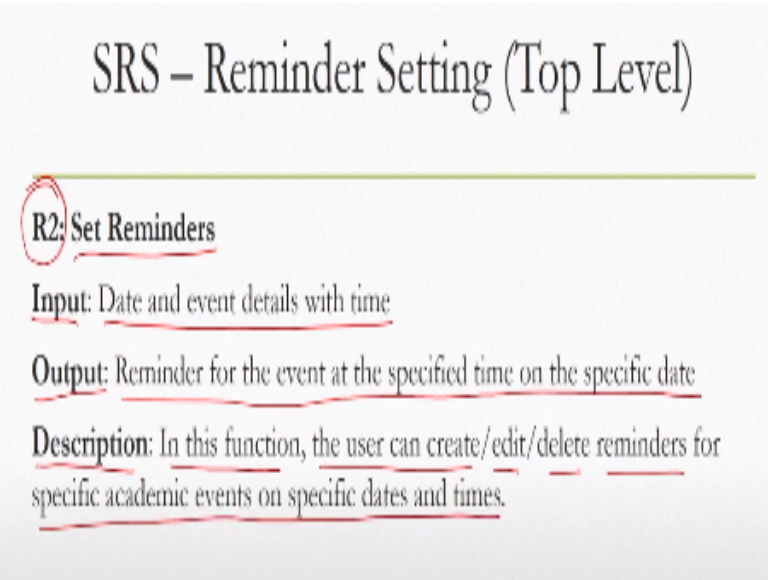
So for a user to set reminders for academic events day wise what we need? We need to create device events, we need to edit device events and we need to delete device events.

(Refer Slide Time: 23:45)



So these 3 sub functions should be good enough to represent this particular requirement of setting reminders for academic events. So let us see how they can be arranged in the form of a hierarchy so that is quite straight forward. So top level we have set reminders as the top level function then below it sub functions come first is create reminders, then edit reminders and then delete reminders. So this hierarchy looks similar to the previous hierarchy that we have already covered.

(Refer Slide Time: 24:45)

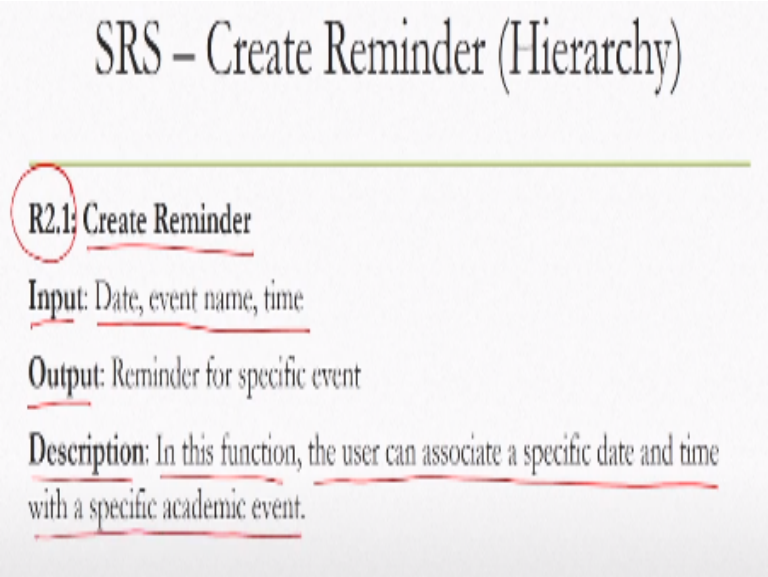


Now let us create the corresponding SRS or specify these functions in the form of textual description as we have seen earlier. First is the top level function so since it is another top level

function so we have given it the level R 2 the previous top level function was R 1 and the name is set reminders. Now as per requirement so we have to provide the input which in this case is date and event details with time.

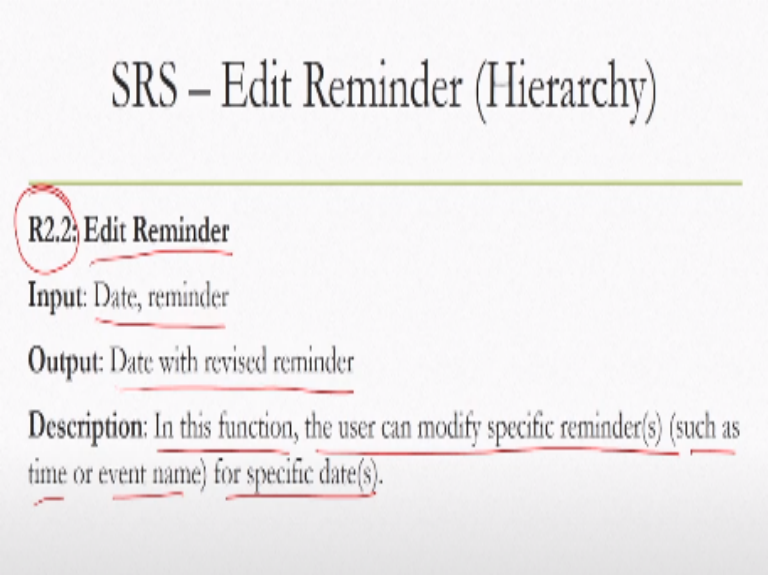
And we need to specify the output which is reminder for the event at the specified time on the specific date. Finally there is this description which says that in this function the user can create, edit, delete reminders for specific academic events on specific dates and times. So this top level function talks about all the three sub functions create, edit and delete and then under the sub functions will specify which of these activities are applicable.

(Refer Slide Time: 26:09)



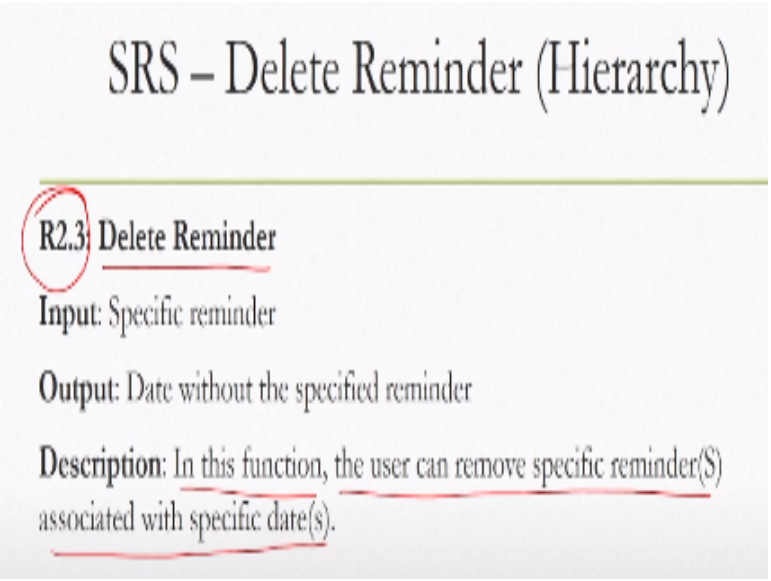
So as per the hierarchy we have set create reminder as the first sub function so accordingly we have given it a level 2.1. 2 because it comes under the second top level function and 0.1 or 2.1 because it is the first sub function of the second top level function. Then there is this input which is date event name and time together constitute the input output is reminder for specific event. And description is in this function the user can associate a specific date and time with a specific academic event this is the description for this sub function.

(Refer Slide Time: 27:09)



The second sub function under top level function is edit reminder so we have given it a label R2.2. In this we have input as date and reminder output as date with revised reminder and description as in this function the user can modify specific reminders such as time or event name for specific dates. So this is the second sub function under the second top level function.

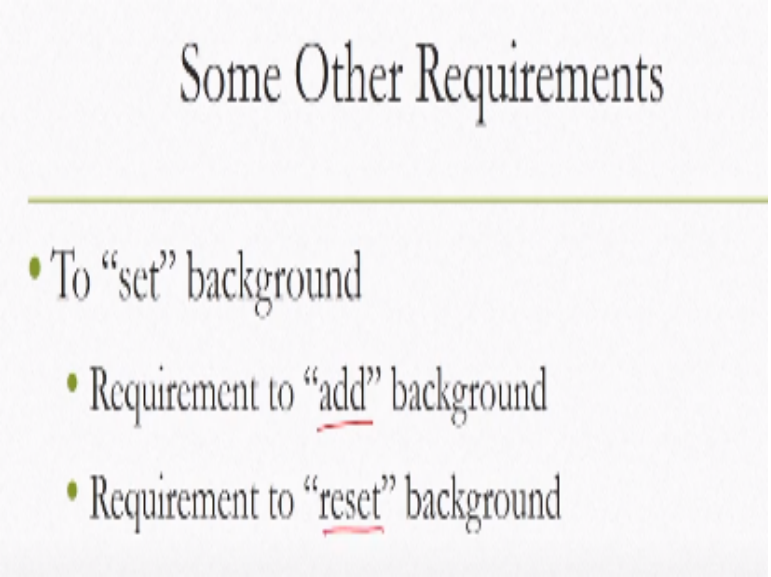
(Refer Slide Time: 27:53)



The third sub function is delete reminder so we have given it a level R2.3 for the same reason that we have mentioned earlier. Here input is specific reminder, output is date without the specific reminder and description says in this function the user can remove specific reminders

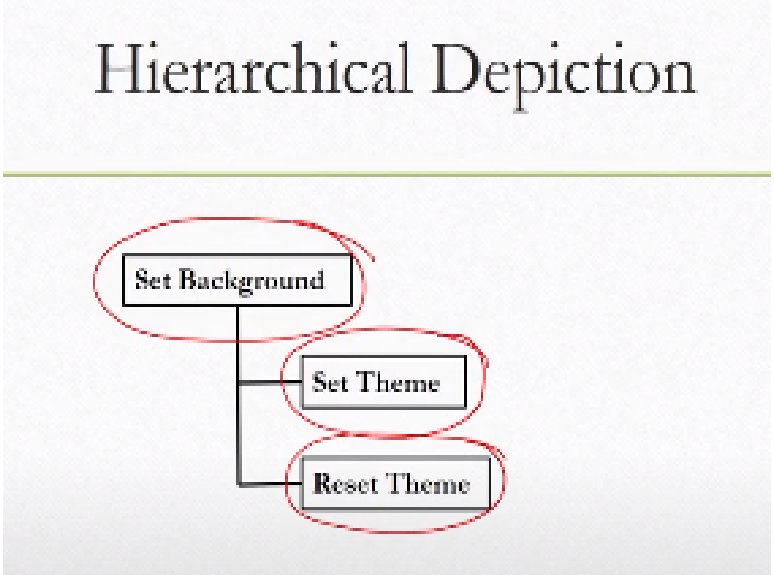
associated with specific dates. So that is the description for this third sub function of the second top level function.

(Refer Slide Time: 28:43)



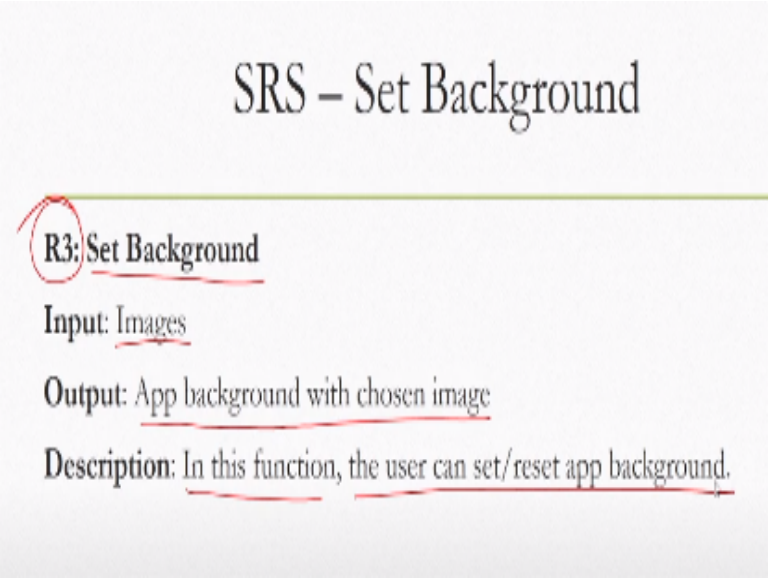
We also mentioned some other functions for example set the background user can set the background to make it look better quote unquote better as per his or her aesthetic judgment. So in order to have this function we need to be able to add background and we need to be able to reset background both are requirements for setting background task.

(Refer Slide Time: 29:20)



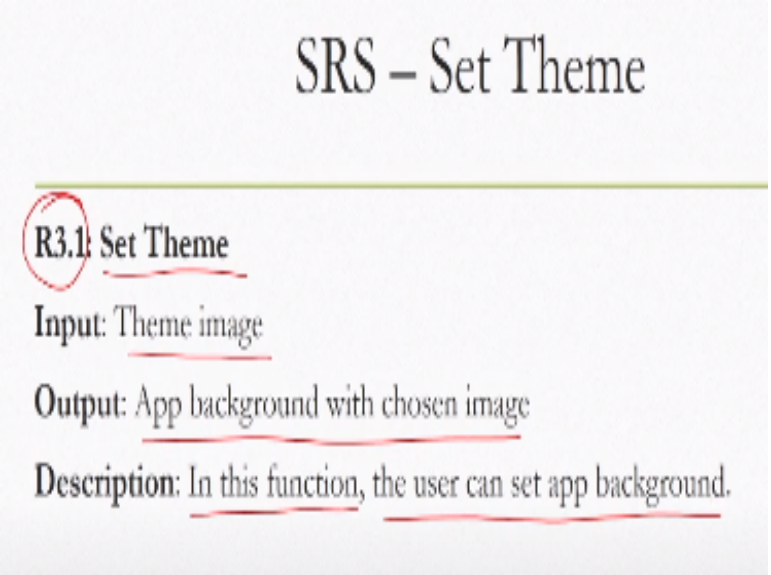
So accordingly we can create yet another hierarchy of tasks or functions and sub functions. So top level function is set background and under this comes set theme and reset theme so these 2 can be considered to be 2 sub functions for the top level function set background. So let us quickly have a look at how to specify these functions and sub functions using the SRS notation?

(Refer Slide Time: 30:11)



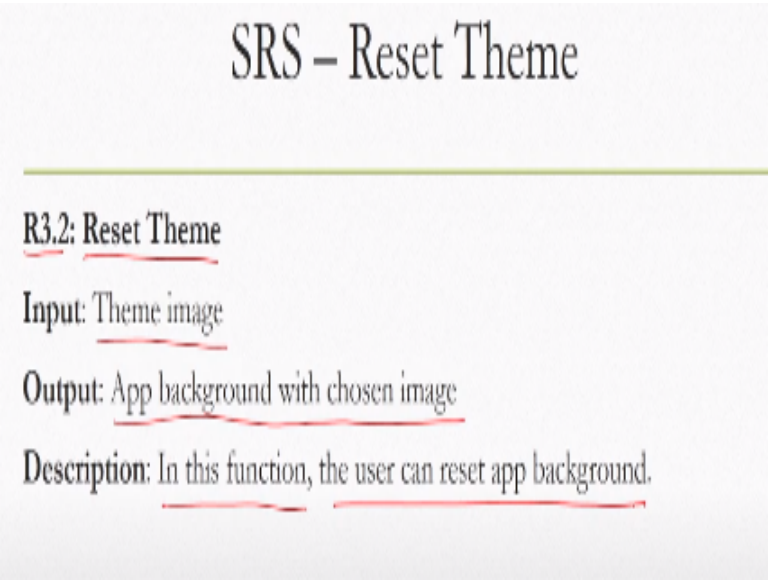
So top level function is set background name given is set background and label is R3 because this is the third top level function we are discussing. Its input is image or set of image output is tab background with choose an image. So ideally its input should be one image and output should be the app background with chosen image so the image is the background. We can write as its description text like in this function the user can set or reset app background. So both the activities can be written in the description of the top level function.

(Refer Slide Time: 31:16)



Then we will specify the lower level of the hierarchy the first sub function that is set theme so we are giving it a label 3.1 because it is the first sub function of the third top level function. Now here input is theme image output is app background with choose an image and we can have a description as in this function the user can set app background. So this can be a simple description of the function.

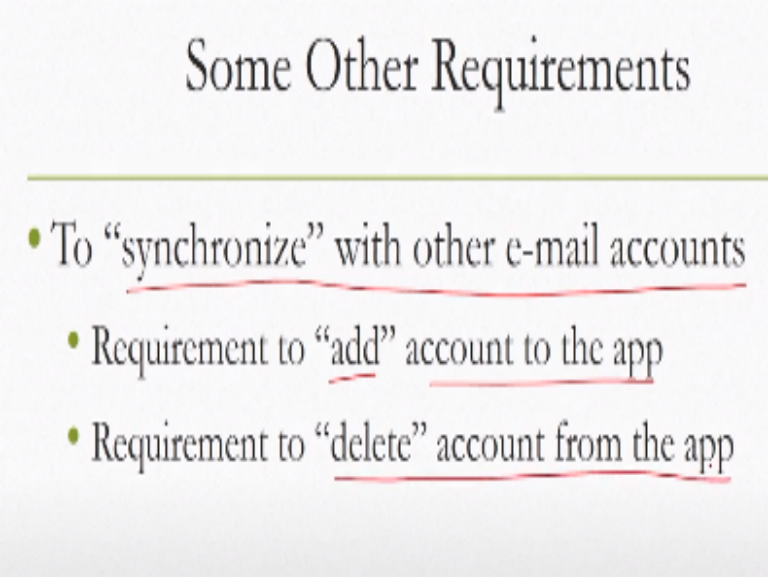
(Refer Slide Time: 31:50)



Then for the second sub function we can have a label R3.2 and the name as reset theme it can have an input again as a theme image in the form of a theme image and output will be the background with the chosen image. So in both the cases input output is similar in nature only the

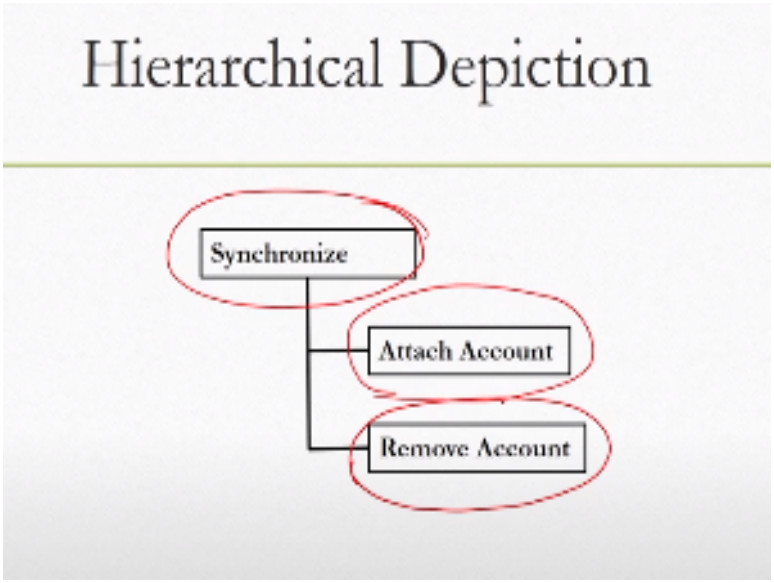
content varies description is also similar in this function the user can reset app background so we can have similar way to specify input, output and description for this two sub function under the top level function.

(Refer Slide Time: 32:42)



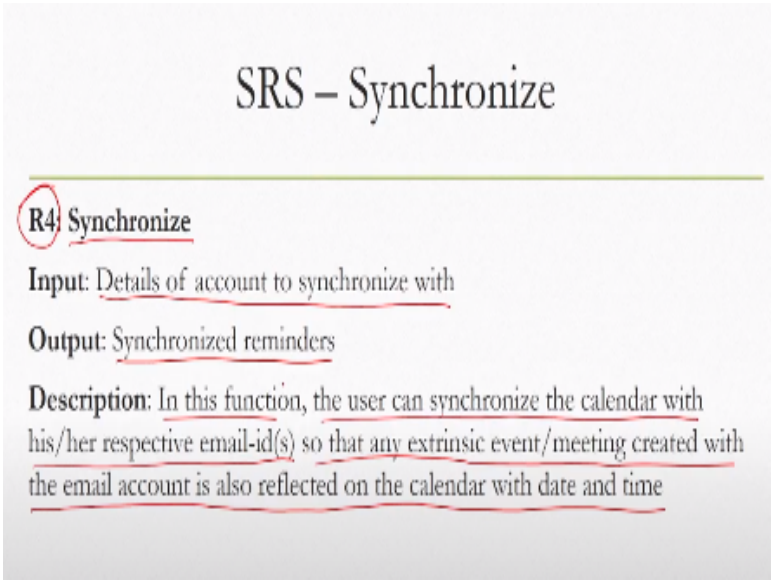
Setting background is one requirement then we can have one more requirement which we discussed earlier that is synchronize with other email accounts. So that if something is listed and linked to this account then it automatically comes to the calendar display. In order to achieve that we have further requirements one is to add account to the application second one is to delete account from the application. So I may need to remove the synchronized account so some facilities should be provided for that also.

(Refer Slide Time: 33:23)



So we can create in a very simple manner hierarchy so top level function is synchronized under this comes 2 sub functions one is attach account and the second is remove account. So the hierarchy can be created in a similar manner in which we have created the earlier hierarchy and once this is created we go for creation of the specification document or the SRS part for these functions and sub functions.

(Refer Slide Time: 34:13)



So how the SRS should look for this top level as well as the sub functions for the top level function we can give it a label R4 because this is the fourth top level function and the name remains the same synchronize. Then we have input as details of account to synchronize with

output as synchronized reminders. And description in this function the user can synchronize the calendar with his or her respective email ids.

So that any extrinsic event meeting created with the email account is also reflected on the calendar with date and time. Slightly longer description however it conveys the purpose of the function. Now again take a pause and have a look at what we are doing just for pointing that out again we are specifying a function. But nowhere we are talking about what this function should be doing only we are giving it a name so that is abstraction and a purpose.

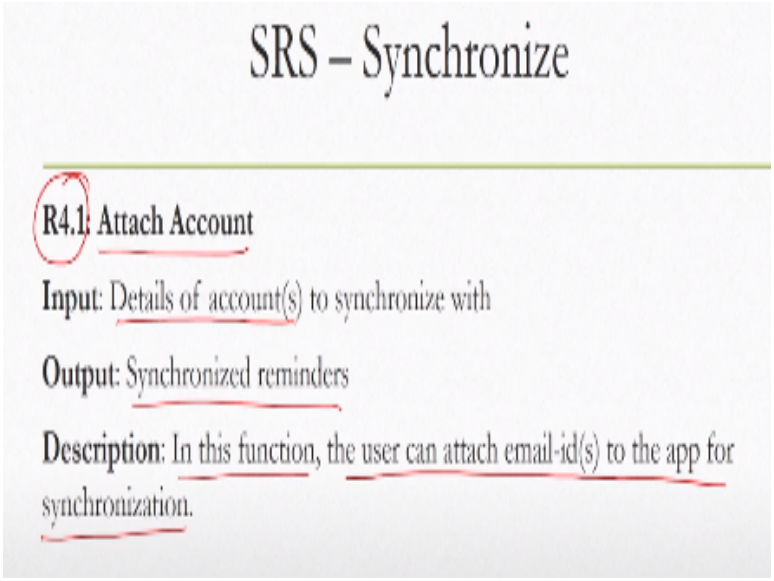
That is again enforcing the abstraction idea then we are specifying input and output now while specifying we are not specifying the exact nature of the input that is also important. Again input and output is also specified in an abstract and flexible manner like here details of account is an input. Now how to provide that detail should it be a textual string should it be an alphanumeric string should it be uni-code compliant string should it be in some other format.

These details we are not providing and we are not interested also because those are implementation details so those are not part of this stage of requirement specification. That is very important for you to keep in mind that in requirement specification we do not bother about implementation details such as in which format to specify and how to process the information? So neither we are interested in finding out the algorithm for this function to implement nor, we are interested in specifying the exact input and output format.

Rather we will keep it as abstract and flexible as possible so that later on designer can make a choice depending on availability of platforms, resources, manpower and other things. So synchronized reminders, is output but we are not specifying what it means exactly to be having an output called synchronized reminders. So it simply states that the reminders are now synchronized with the display and they are now visible on the display.

But what that exactly means and how that is done is not what we are specifying here rather only the, what should be done what is expected?

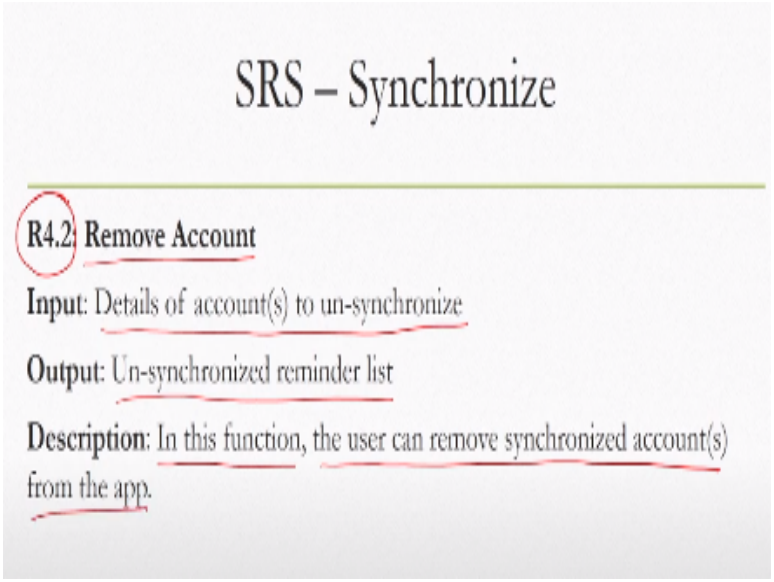
(Refer Slide Time: 37:29)



Now under the top level function there are 2 sub functions that we have seen one is attach account. So accordingly we have given it a level 4.1 because this is the first sub function and a name attached account in this the input is details of account or maybe more than one accounts that we need to synchronize with the app and the output is the synchronized reminders.

A description here can be simpler than top level function in this function the user can attach email ids to the app for synchronization. This can be simple description of the purpose of the function.

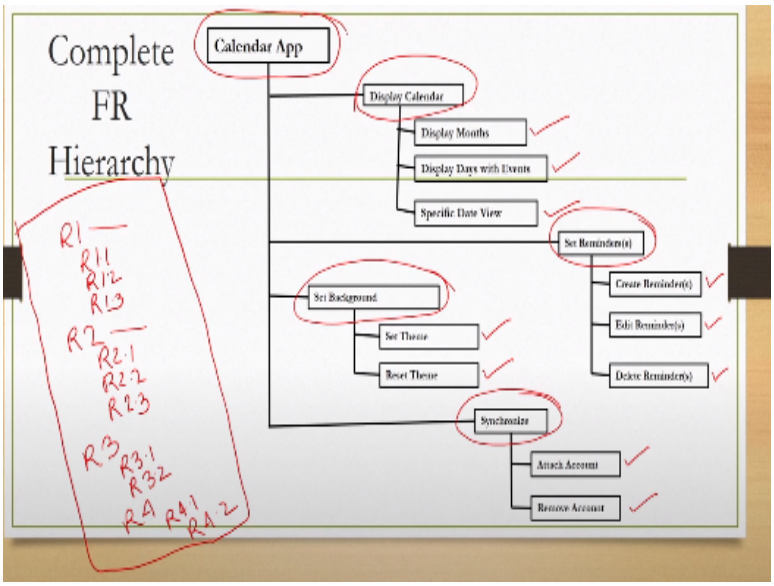
(Refer Slide Time: 38:25)



Then comes the other sub function that is remove account so the label given is 4.2 indicating that it is the second sub function for the fourth top level function and the name is remove account. So the input is details of accounts to be removed or unsynchronized and output is unsynchronized reminder list. That means now the display should not contain events that is linked to the account which is unsynchronized and the purpose again can be simply stated as in this function.

The user can remove synchronized accounts from the app so that events attached to that account does not reflect on the display.

(Refer Slide Time: 39:25)

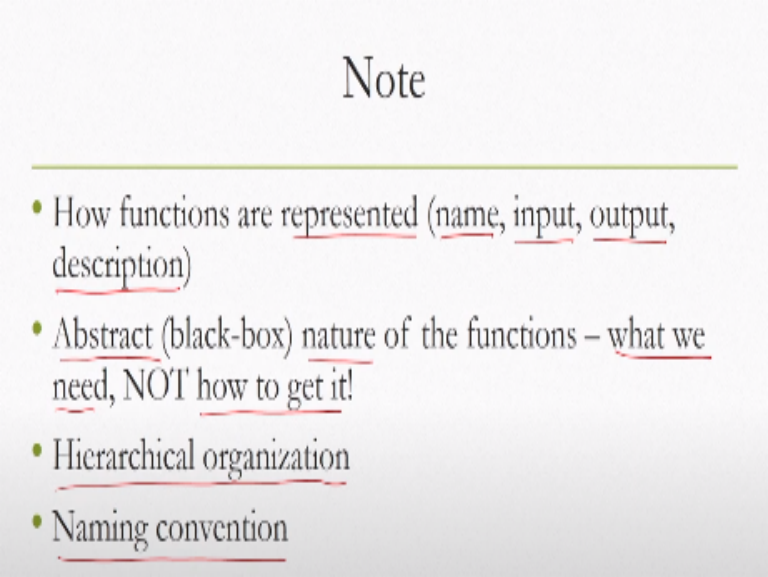


So then how the complete hierarchy looks like let us have a look so we have top level function as calendar app that is the entire top level function. Under this we have one top level function display calendar under which we have three sub functions display months, display days with events and display date view specific date. Then we have set reminders as another top level function under, which there are sub functions create reminder, edit reminder, delete reminder.

Third top level function is set background so under this top level function there are 2 sub functions set theme and reset theme. And finally this fourth top level; function synchronize under which we have attach account and remove account. So effectively what we can think of as the SRS should contain R1 then R1.1, R1.2, R1.3, then R2, R2.1, R2.2, R2.3, then R3, R3.1, R3.2 and finally R4. Under which R4.1 and 2 now for each of these of course we have to specify like R1 or R1.1 we have to specify the input output and description.

So once that is done this entire thing will constitute the SRS for the calendar app which will specify the functional requirements for the SRS to re-emphasize the point. So these are only considering the functional requirements we have not yet bothered about specifying the usability requirements that we have discussed in the previous lecture. Now once we put in the textual forms of the functional requirements that will constitute our whole SRS for the calendar.

(Refer Slide Time: 42:13)



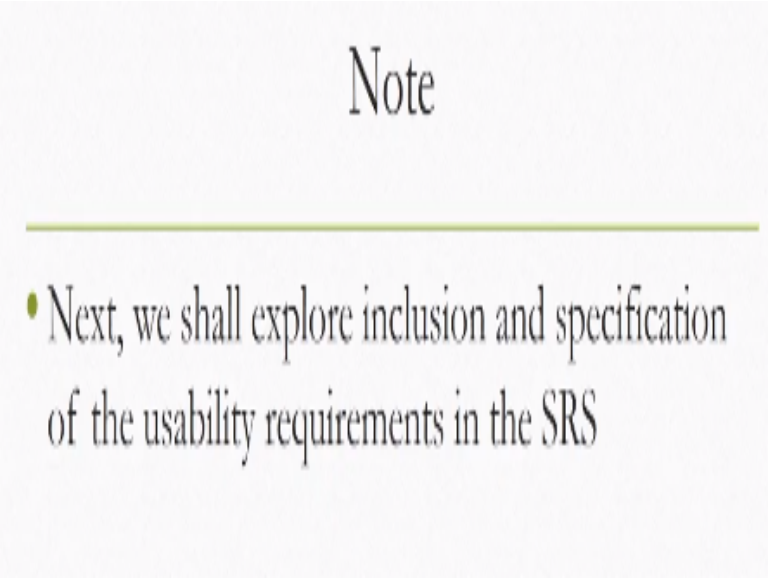
So again here you should keep in mind and note in this lecture how we have represented the functions in SRS namely kind of name given sort name 1 or 2 words. Then input specification, output specification and description for functions at each level then the abstract nature of the way we have identified the functions. So essentially our focus is what we need rather than how to achieve the requirement how to do it?

So our objective is only to specify functions which we need rather than how to get what we need or the algorithm for the functions so that is not required at this stage. And then hierarchical organization of the functions so how the functions are organized in the form of a hierarchy? And also the conventions that we used for naming the functions that we already mentioned earlier labeling convention, names and naming convention.

So as I said we have not discussed anything about usability requirements in this particular lecture here we concentrated more on how to specify the functional requirements? First of all how to

identify and how to specify? In the next lecture what we are going to do is to see how the usability requirements that we have discussed in the previous lecture can be incorporated as part of the SRS. So that we can implement it as functions and at the same time ensure that the system is usable.

(Refer Slide Time: 44:16)



How to do that we will see in the next lecture. So with that we come to the end of this lecture here we continued our discussion on the case study and we discussed specification of functional requirements for the calendar app. In the next lecture we are going to expand the SRS document by incorporating usability requirements as well. And how to do that because usability; is a non-functional requirement how we can convert it to functional requirement.

And apart from such conversions is there any other way to make use of usability requirements? Because usability requirements, is not necessarily convertible to functional requirements. So; how to make use of the usability requirements? That we have identified through contextual inquiry that also we are going to discuss along with how and if we can convert it to functional requirements. That is all for this lecture thank you hope you have learned and looking forward to meet you in the next lecture soon thank you and goodbye.