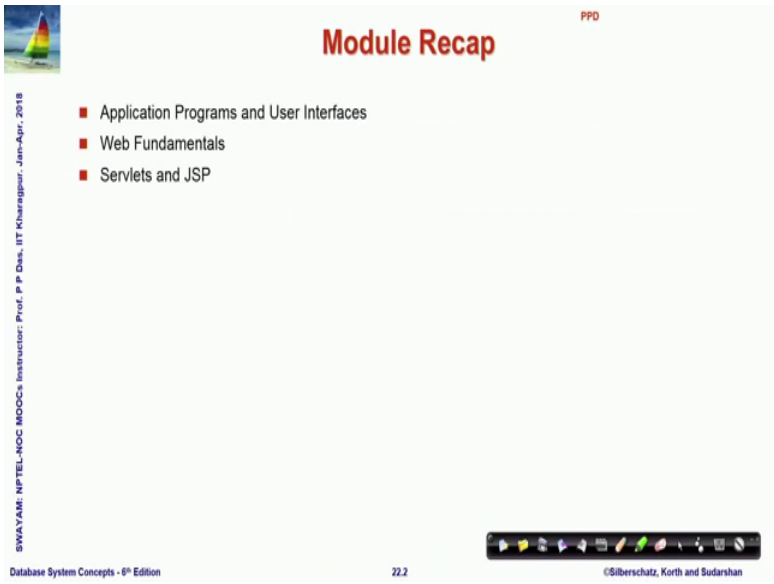
Database Management System
Prof. Partha Pratim Das
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture – 22
Application Design and Development (Contd.)

Welcome to module 22 of database management systems, we have been discussing application, design and development this is the second part of that discussion.

(Refer Slide Time: 00:27)

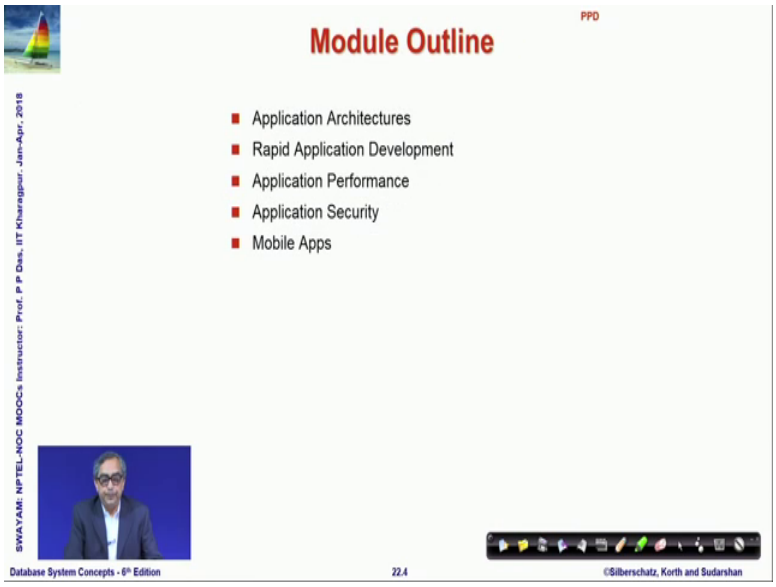


In the last module, we have taken a quick look at the application programs and the user interfaces, looked at the fundamental notions of web and specifically the servlets and JSP.

(Refer Slide Time: 00:38)

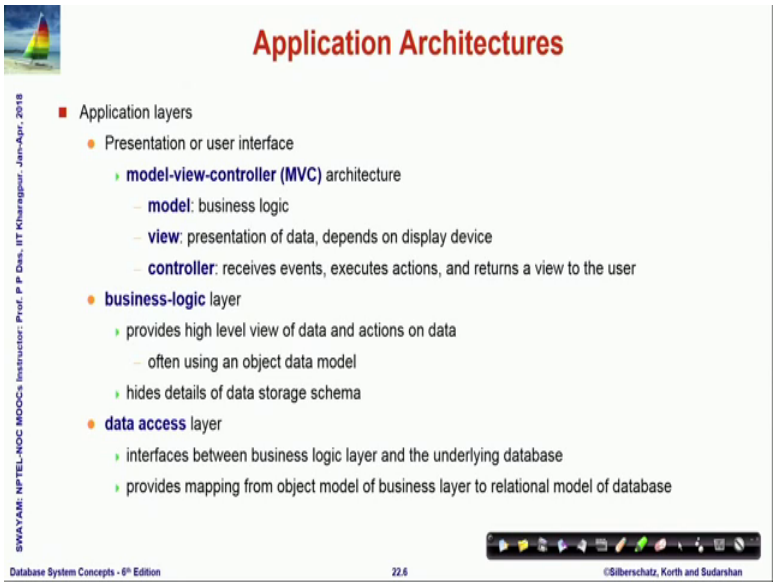
In the current module, we would like to understand the 3 tier architecture in little bit more detail, and explore quickly take a look into the rapid application development processes what kind of help is available, for quickly develop applications and then, we briefly we will look into the issues in terms of an applications performance and it is required security, and at the end we will discuss how a mobile app is similar to such web based database applications? And how they are different?

(Refer Slide Time: 01:14)



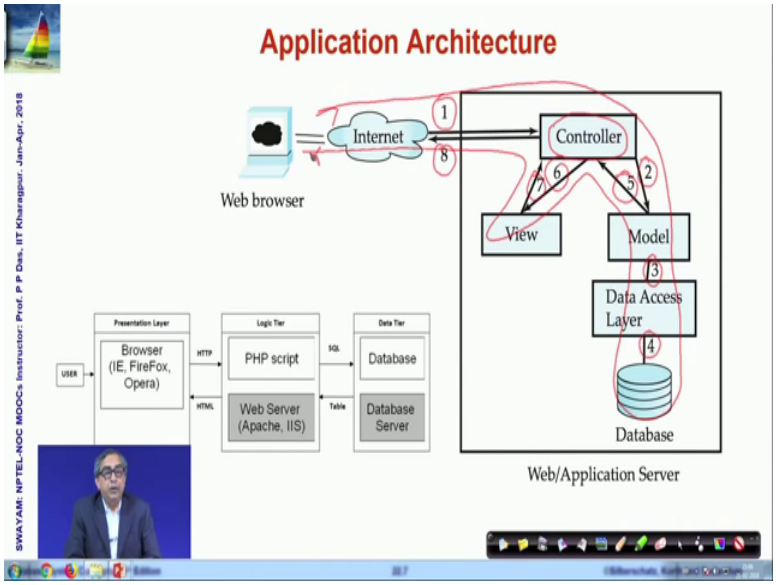
So, this is the outline the 5 parts.

(Refer Slide Time: 01:19)



So, in terms of the application architecture again the presentation layer, or the user interface, business logic layer, and the data access layer, the frontend, the middle layer and the backend. Now, in the presentation layer or the user interface it is typical that applications follow, what is now known as MVC architecture, model view control architecture where, the model is the the is kind of the business logic that, that is implemented in terms of the frontend information. View is the actual presentation of the data, that HTML and controller is one who, receives different events execute actions and so on and then, we will go into the other layers.

(Refer Slide Time: 02:09)



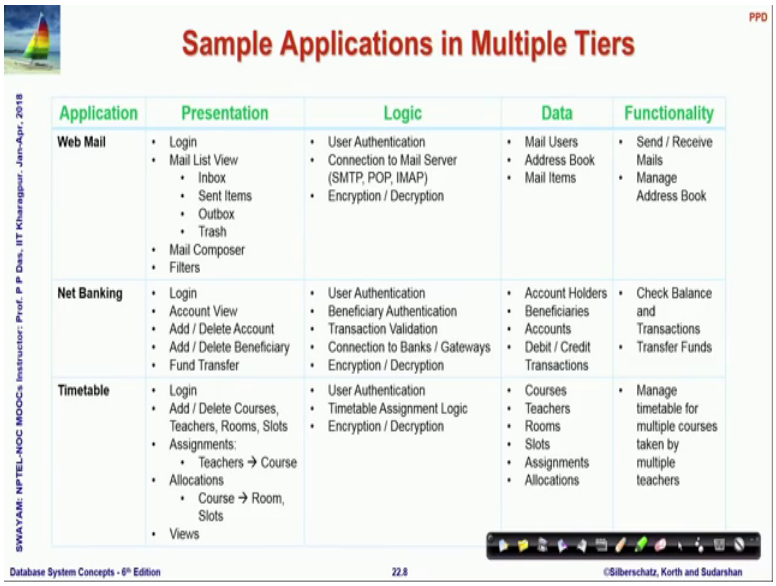
So, here let us try to understand this flow. So, there is a request in the web browser say, to the service. So, this is the sequential now, let us say we are trying to log in to Gmail. So, in one we send a form HTML form which, has the username and the password and possibly encrypted, that comes to the controller ah. So, which basically controls the different events.

So, the controller knows that, it has to now decide whether, this what actions are required in terms of this input data. So, it is sends it to the model which, is the business logic here. So, the business logic model knows now well. So, at this there is an application which, deciphers the business logic which says that, business logic required here is we have a password and we have a user names now, we have to decide whether this user is a valid user and whether, he or she can be allowed to login. So, the model has to check on the user data, the user id and password data and therefore, it has to come from a database. So, it passes on this request to the data access layer, the data access layer in turn access the database.

So, you can think of data access layer is something like a SQL query layer where, you have formed a query select etc., from etc., where, user id is equal to PPD, password is equal to XXX, the database depending on what is found in the database is back to the model, and the model then sends it to the control it is says ok, this is what I have found. So, this is the result of the data that result of the request that has been prepared. So, it is says that well this is have been found and therefore, we have extracted the mails in the inbox that existed, or it is says that the authentication is not possible. So, it plugs in a error message and sends it to the controller, controller now knows that a response has to be framed. So, the controller sends it to the view of the MVC, the view we prepare the HTML that needs to go back. So, view prepares the HTML and sends it back to the controller. So, controller now has the response which it is sends back to the web browser through the internet, and we get to see that well now, my inbox and mails are all here.

So, this is a complete flow of starting from here, going through this, coming back, going here, going back here, is the is the whole rout of the request, response that goes over the HTTP in a typical web or application scenario, that is the there is a way this application architecture is expected to work.

(Refer Slide Time: 05:13)



So, here I have created a small table, showing you the different activity is that happens at the presentation logic and data layer of different common applications like, web mail like, Google. So, at the presentation layer you will do things like, log in, mail list, view inbox, sent item, outbox so on, mail composer. So, we can write mails filters of checking at different mails. So, all the all these happens.

So, for example, if you talk about filters they might often happen in the java script itself, that trans within the browser, the logic the business logic will do user authentication, connection to mail server because, a mails have to come from a different server, they are not they may not be setting typed in terms of the of the database itself.

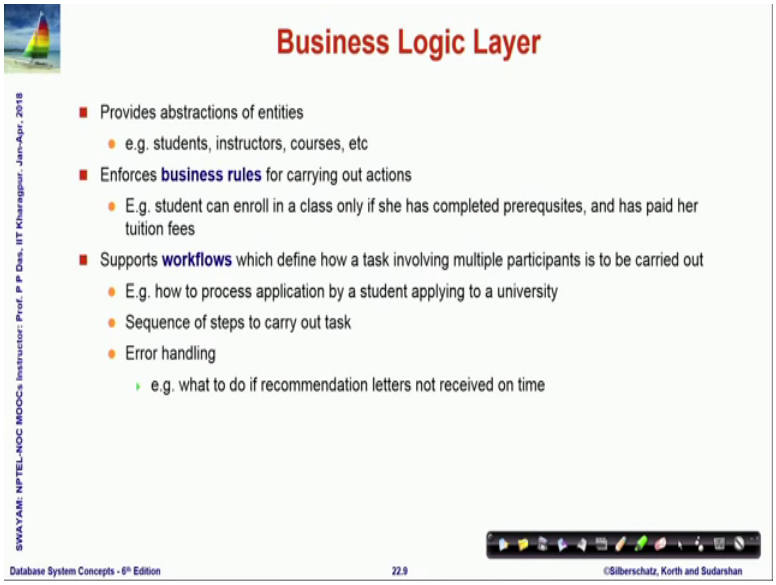
And then user encryption, decryption and the data side you will have different tables to represent the mail users, the users like you and me all who are users of the Gmail, the address book of each for each one of us the mail items and so on, and that will give us the functionality of send, receive mails, managing address book and so on. So, similarly I have listed an application for net banking which can where, we can check balance and do transactions transfer funds, or a timetable where you can manage time table for multiple courses taken by multiple teachers and so on.

So, you can if you think about an application then, you should be able to moderately map it is a functionality across the presentation logic and data layer. So, that you know what you want to do at the clients, and which is which is at the presentation layer, or what you

want at the absolute packet which is on the data, what tables and all the databases that you want to maintain, and the business logic of how actually this application will give you the result, how actually it will?

So, that is something where, you will have a whole lot of complex logic that might come in.

(Refer Slide Time: 07:18)

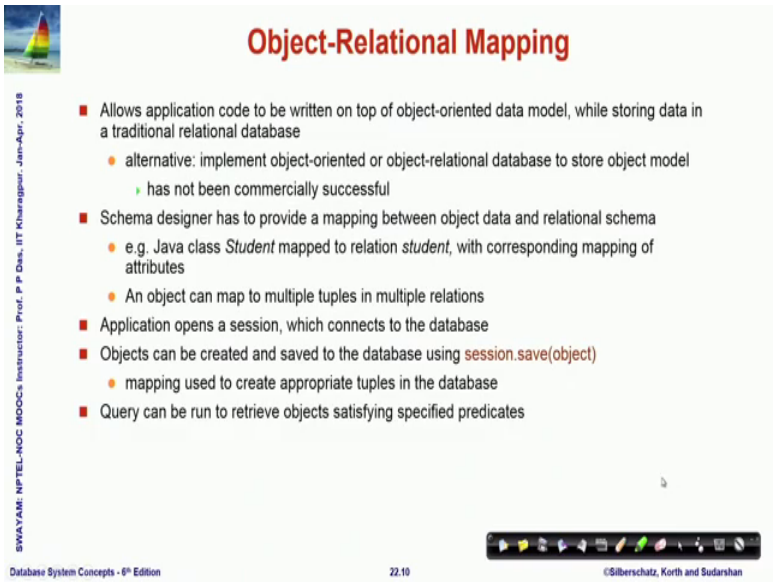


So, coming to the business logic layer specifically, that provides abstraction of that will provide abstraction of various entities, students, instructors, courses, mails, your accounts, balance and so on, and that will enforce business tools for carrying out this. So, a student can enroll in a class only if she has completed prerequisites, you can transfer funds from one account to the other, provided, you have the authority to transfer, provided you have enough funds in the account that is going to get debited and so on. So, those are the different business tools, which will be employed by the business logic layer, and it will support a work flow which defines how a task will be carried out in terms of the multiple participants, and remember that all participants may not actually be human beings, they could be some could be human being some could be other machines or other applications as well.

So, it gives you the work flow. So, you can any of the 3 application that I just mentioned, everywhere you can find that there is a work flow, if you are want to check a mail then, there is a steps that you need to do go to the inbox, chose the particular mail item, get the

body and then if you want to reply to that, select that, the submit that, you get a get a new form when you write the reply, and within that form you get the original copy of the original mail and so on. So, all these kind of work flows will be supported by the business logic layer.

(Refer Slide Time: 08:47)



Now, certainly you can understand that at the frontend, if we if we talk about what are different you know languages and models, that we are working within the frontend naturally our language is HTML tries for presentation, embedded with java script, at the backend in the data it is the database and the SQL query. So, it is a relational model, but what happens in between? What happens in the business layer which connects them? Now, naturally business layer you could write complex business tools.

For example, in the time table application we will have a complex algorithm, I know to find out what allocation of class rooms and slots, are feasible for assignments of teachers to courses availability of rooms and so on. So, often the business logic layer, the this tier would be convenient to write in some typical common high-level language like, C plus plus or java, and naturally you would know from your experience of software engineering that, if you have a object based language then, that will be a very convenient to do that.

So, which means that, if you have say some entity as a student in your relational database then, most likely in your business logic, which is a java code you will have a class called

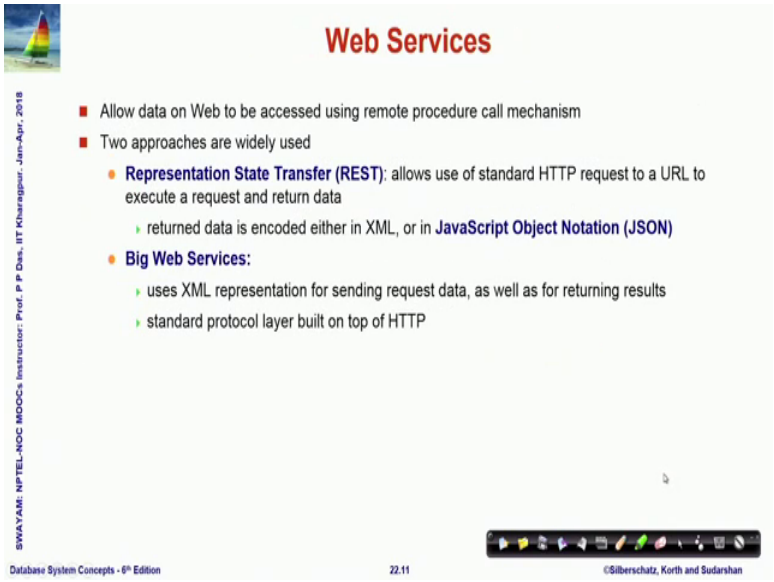
student. So, the relation student is in the relational model, and your class student is in the object based model, and you will need to define these in terms of certain mapping of the attributes, which we had shown when we talked about embedding of a languages, and this is what is commonly known as a object relational mapping.

So, that objects can map to multiple tuples, in multiple relations and can be viewed in a different way. So, so this mapping itself we could create a virtual view in the business logic layer, in the business logic language that you are looking at. Of course, they have been attempts to create a models, which are relational models which are also object oriented those are called object relational databases, some of them have been successful, but not really commercially successful.

So, we continue to work with SQL, and the relational database kind of things in the database level, and some kind of a high-level language like, C plus plus, java and the object-based model in the middle tarred in the in the business logic, and continue to do the object relational mapping for solving the problems. So, you can here, I have given the points of what happens? How the objects get created? Application opens a session, which connects to the database because, we need to get the data from the database, they can be objects that can be created safe to the database.

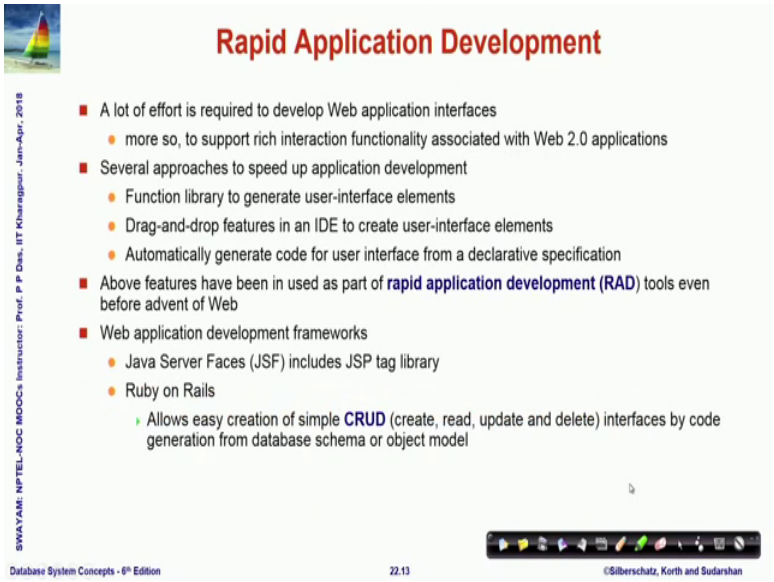
They can be extracted from the database; new objects can be created and mapping use to create a appropriate tuples in the database. So, it is a 2-way traffic that will keep on happening where, the business logic layer will continue to see entities as objects whereas, the database layer will continue to see them, as tuple in the relational database.

(Refer Slide Time: 12:07)



So, there are also several web services, that can be used and a you may be getting familiar with that, we will not go deep into this, but I will just mention that, web services a mechanism through which, you can access a data from remote server using what is known as a remote procedure call, and today very common approach for this is called rest representation state transfer, which allow standard HTTP request to a URL to execute a request and return data, and a several of the web services are based on that, and are the are big web services which you must have heard of, but I just mentioned it at this point because it is contextual, but we will not get into those in this course really. Now, coming to how do you actually develop applications?

(Refer Slide Time: 13:05)

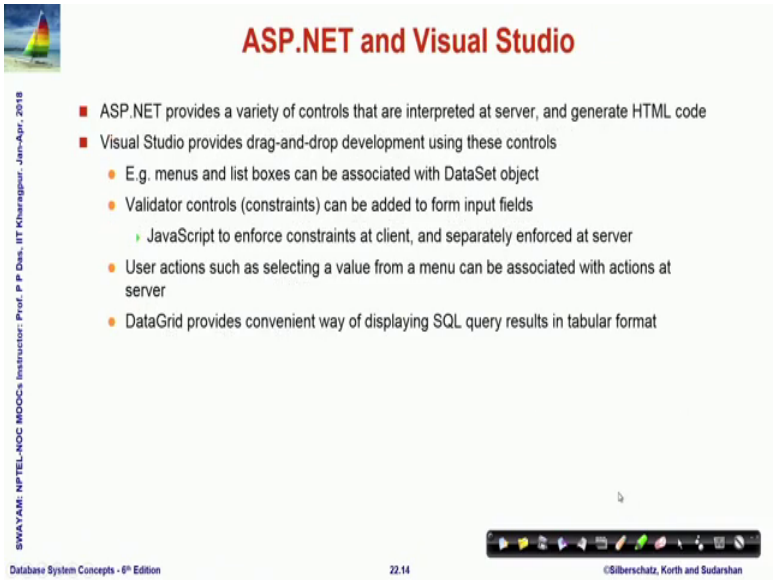


Now, it is not a easy process that is a lot of refer require to develop web applications, you need to support the functionality that of current day web. So, you need several approaches to speed up applications. So, you this is in parallel to if you think of how? What has been done to speed up development processes, development applications for different say C applications, or C plus plus, or java applications.

So, one approach naturally is to variety of function library, which can help you easily get user interface elements like buttons, checkboxes, radio drag and drop features in the IDE, IDE stands for integrated development environment like, visual studio, front page these kind of which can use a, which can create user interference elements easily you can automatically generate code for user interface, and these are all parts of rapid application development tools, and some frameworks are very popular, this is for primarily the java server faces or JSF is a framework where, you can rapidly develop fill in all the requirements of the different layers, in a web based database application in other very popular is ruby on rails, which allows easy creation of simple crud create, read, update, delete.

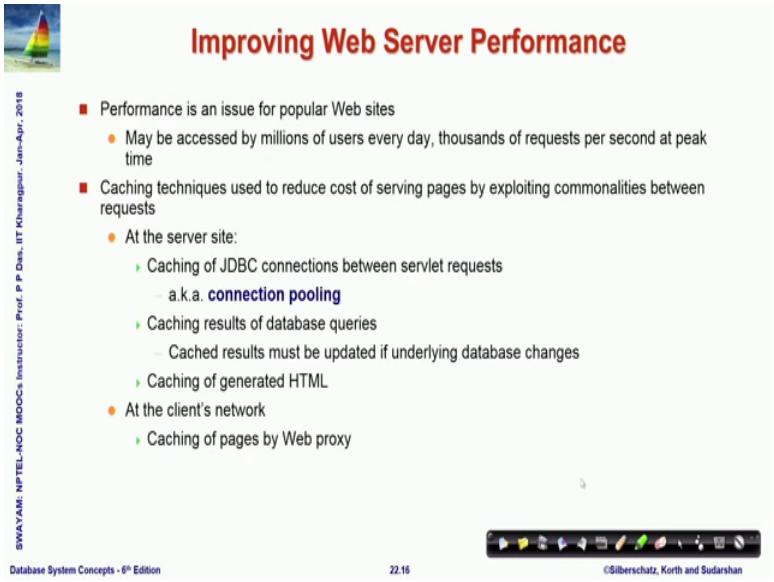
So, if you look at database applications and common applications will all applications will at least need to do this it lead to create data, read data, update data, delete data. So, you can do that quickly with ruby on rails. So, if you are locating into really the development of database applications, get yourself familiar with this rapid application development processes.

(Refer Slide Time: 14:53)



There are different other frameworks as well, another very popular and widely used framework is Microsoft specific, this is unfortunately not portable because, it is proprietary of Microsoft where, you can use the the dot net framework of Microsoft, which helps you with lot of an a resources and libraries which are already provided, and like we talked about JSP, we can use ASP dot net here active server page in the dot net framework, which provides a whole lot of controls and you have a very nice powerful id in terms of visual studio, high were being proprietary these need licenses and you need to pay for that. So, many a times, many developers may not be able to afford it or like it for that reason. Naturally, as you designed applications and create that, you will have to be careful about it is performance because certainly we all want very fast results.

(Refer Slide Time: 15:52)



So, if I log in to my Gmail application, and I would expect that as soon as I press the submit button with a couple of seconds, my inbox will be displayed on my browser, I am not ready to wait for 2 minutes, 3 minutes, 5 minutes for doing that.

So, while developing the application you will have to make an estimate of how often it will be used? How many users will use that every day and so on? how many, what is your expected heat rate? That is, when the maximum number of users are trying to use this then, what is the you know estimated number of request per second? That will come to your server. And to improve performance their different kinds of techniques can be used, the significant of them is called caching, caching is nothing but if you expect to request to be similar ah. So, that they are results should be similar then, after the first request besides sending the response back, you actually keep a local copy of that.

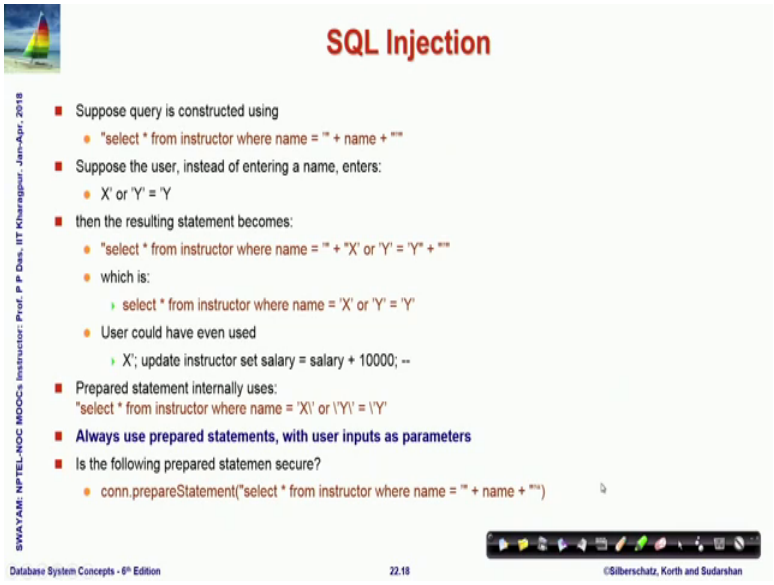
So, that if a similar request come in future, you can you may not re compute it, you can just send that caches copy. So, it can be done in terms of verity of JDBC connections called connection pooling, it can be done in terms of database queries, caching of generated, HTML and so on.

It can be done at the clients network side also by caching pages by web proxy; obviously, if you are using caching to improve performance, you will have to understand lot more of the web dynamics in depth because, certainly if you cache then there is a possibility, that somebody is asking is sending a request for which, the response would not be the same as what it was, when the last time the response was computed and cached. So, if

you send the cached information back then, you may be giving a dated information. So, possibly say for example, if you are checking at the net banking transaction, you have making a fund transfer and checking your account transactions after that one transfer, you would not expect a cached page whereas, if you are looking at the website of a say IIT Kharagpur then, it will be to cache that because, it is not expected to change very frequently. So, these are different factors. So, we do not have it in the scope to going to different issues of, how to improve performance? And what are the different challenges?

But I just want that you to be sensitive about these issues. Other very deep you know concerned, deep requirement about applications are the security of applications, there are this is a very involved topic, and there are several issues that are involved.

(Refer Slide Time: 18:48)

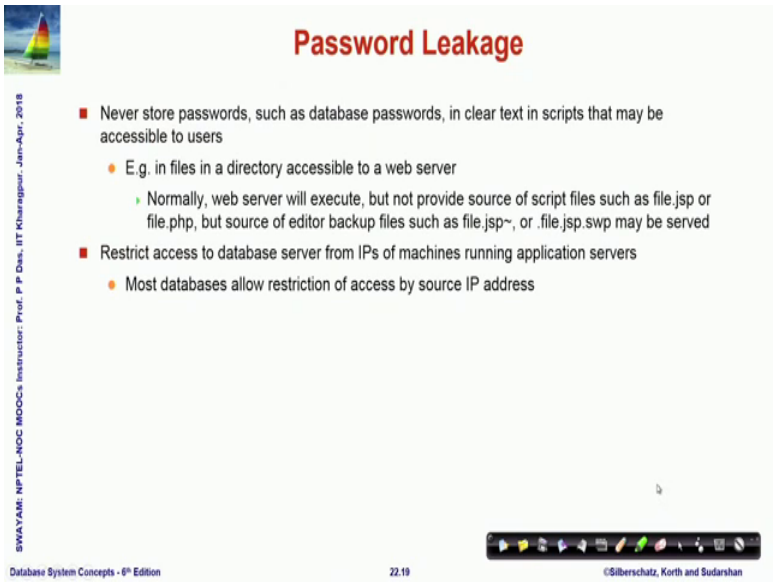


So, and I am talking about only a few of them because, the many of them require knowledge about several other fields particularly, in the field of security and an encryption and so on. So, one that is very common in terms of SQL query is known as a SQL injection where, based on I mean there is if you are expecting some inputs to come ah, and fill up certain parts of the SQL query then, you will have to be careful that, user should not be able to give such input say such strings.

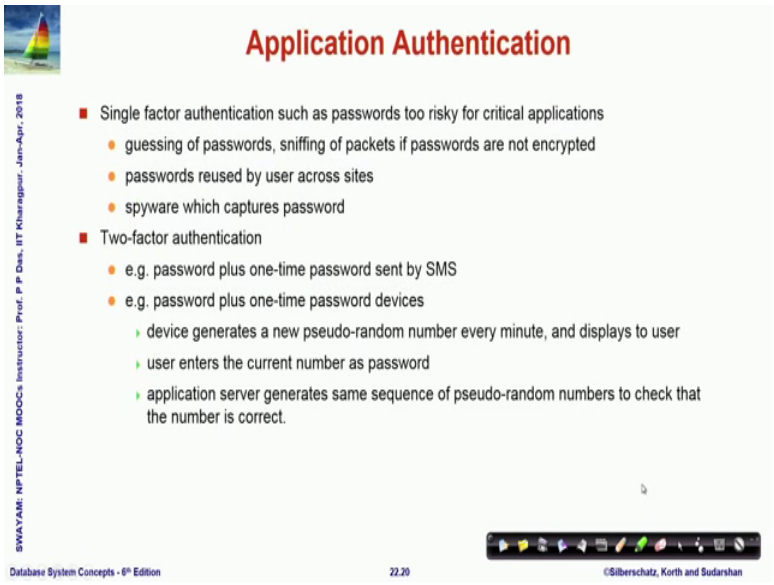
So, that the query actually mean something different, this query may be which was just a query to read something, may update something, or give some different result. So, here I have just briefly highlighted some of those issues, there are other security issues like,

leakage of password if there are important things which are based on a single password then, use the password gets compromised because, it is shared or it is broken in a middle by some hacker and so on then, you will have a lot of risks involved.

(Refer Slide Time: 19:37)



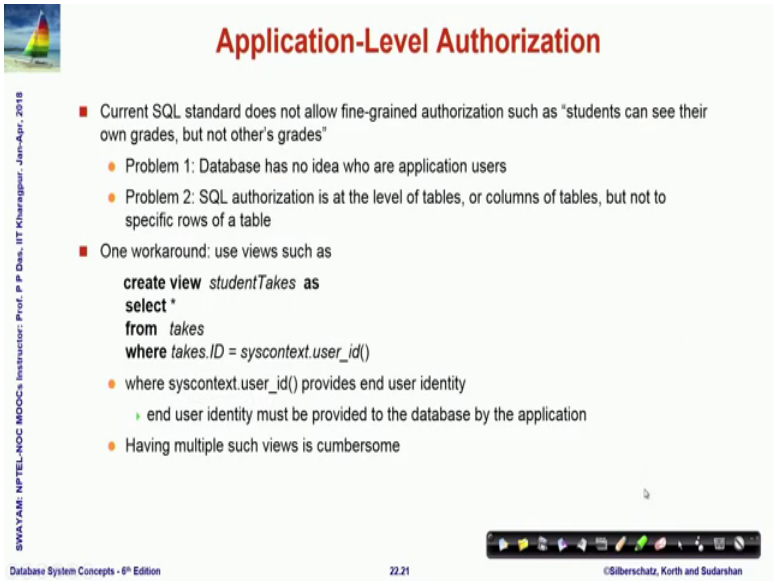
(Refer Slide Time: 19:59)



So, you should be you will need to make sure that, you do not store passwords you just store encrypted forms of that in which encrypted forms, you have must have observed for the last couple of years that, in many cases earlier you could just log in. So, giving just one password was enough, but now in many transactions for example, if you are logging

into a net banking application then, often you will be asked to provide some additional key information, or you will be asked to do a authentication by OTP one-time password and and so on. So, these are common because, you know it is risky to work with just a single authentication, and you will find that some net banking applications for example, if you are doing a fund transfer, then they actually require 2 additional password authentication, one is a special password for fund transfer, and then possibly we will be asked to go through an OTP. So, depending on the criticality of your application and the potential vulnerability of the application, your authentication mechanism will have to be appropriately designed.

(Refer Slide Time: 21:09)



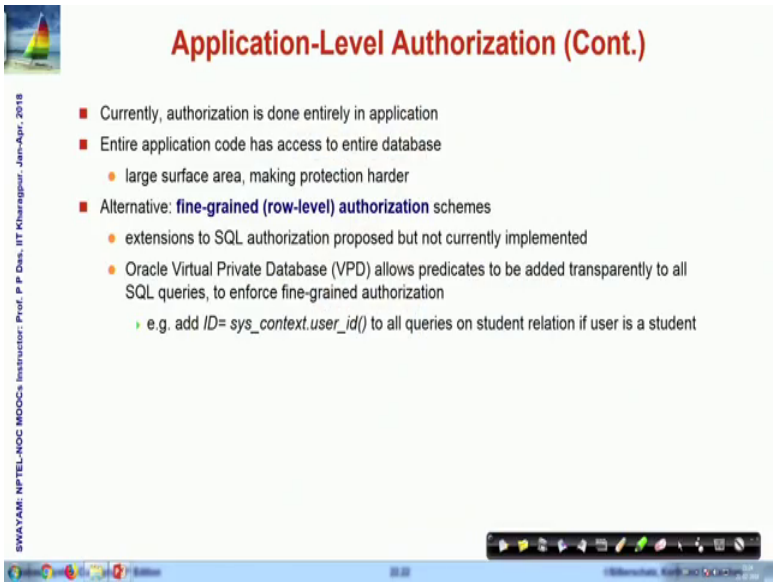
There are issues in terms of security, in terms of the application level also, for example, if you if you are looking at a at a student application where, you want to say that the student, every student can see his or her own grade, but they should not be able to see the grade of others. Now, how do you implement this? Now, 2 issues are one is the database cannot implement, this database cannot provide this as a part of access control because, database has no idea about who the application users are, and the second if you recall the way, we talked about authorization in SQL, that is in terms of tables or columns of the tables, but SQL has no mechanism to create authorization for rows of the table.

So, this will have to be handled, in terms of what is known as at the application layer. So, this is where you are. So, what this is saying that you are created a view where, you are

created a view student text where, to do this you will have to read you are saying where, text dot id is equal to this function called is sys context dot user id. So, sys context is an object naturally, dot user I d this function called. So, what this function gives you call to the function gives you is an information about the end user identity, and that identity has to match, the identity that exist in the text I d for the result to be computed. So, this will ensure that depending on who is actually the current user, the same view will be evaluated for different results.

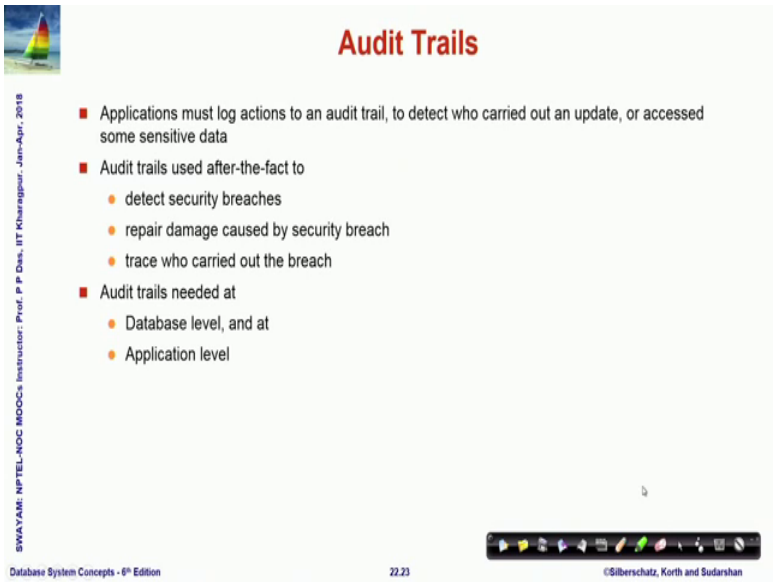
So, this is no not a not a very sound this is not a very comfortable situation because, here the authorization is not being implemented in the database level, but is being implemented at the application level, but that is way things a because, most of the fine grained authorization currently, is done entirely in the application level only a extension to SQL authorization, at for similar you know fine graining and so on has been proposed, but they have not been implemented because, there are several issues of implementing where there several issues of how do you represent? How do you module? And most importantly, these have potentials of slowing down the database, query process and significantly.

(Refer Slide Time: 23:08)



So, often this is not a preferred mechanism either. Another way to ensure security is to keep audit trail, trail must log actions into it.

(Refer Slide Time: 23:46)

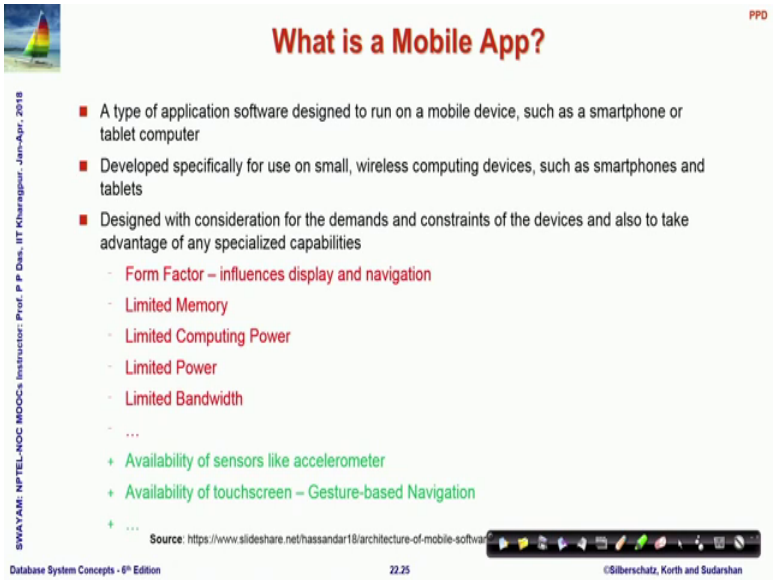


So, where you write down actions in terms of who carried out and update, or who access some sensitive data, or who did a delete and so on. So, audit trails can be used later on if the need arises to detect, if some security breach that happen, or if some damage has been caused by the security breach, that can be corrected and so on create a trace.

So, auditing is necessarily a very required, in a very required activity for any data-based application and audit trail had a good mechanism for that. So, when you develop applications you have to consider has to whether, you would like to support audit trail of course, if you support audit trail then, somewhat your application will get slowed down, it will require more disk to keep that trail because, every transaction every details you will get logged in to the audit trail, but it is very, very important for critical applications like, net banking where currently it is mandated every transaction that, you do every action that you do on your account, through the net banking is trailed in the banks end.

So, that if later there is a there is some dispute, there is some breach has found then, the same can be recovered and trace back to the actions, that you are actually taken ah. At the end of this module let me take a quick look into the mobile apps.

(Refer Slide Time: 25:32)

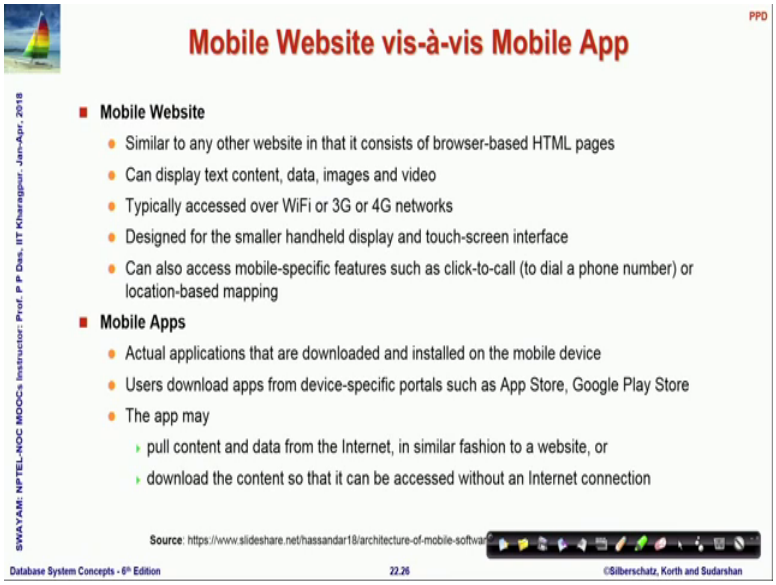


These are somewhat different, but it is very important to today to take a notion of the mobile app. So, it is a type of a application software, which is designed to run on a mobile devise. So, it this is not necessarily mean that, it has to be a smart phone, it could be a tablet or you know some small device, which typically is handled and carried around . So, it is typically specifically for use on small wireless computing devices, and it is designed with considerations for demands and coincident of the device. So, how are how is a different? If I want to use the application or want to have an application which does my task, but I want to do it as a mobile app, I want to do it for a small wireless handled device. Certainly, there are if we if we since the small device it has lot of constraints, there are lot of demands of the device, compare to if we were using a web browser in a desktop or a laptop where, you have lot of resources.

And also on the this is this is on the one kind of the restriction side and on the other side, these devices have certain capabilities, which your desktop or laptop may not have and therefore, it may be you may be able to do more interesting application using this mobile devices, and there could be really interesting mobile apps, and as you all must be using today use 10s of if not 100s of mobile apps for different applications, many of which actually support a database at the backend. So, these are in red are some of the negatives, which are restrictive in terms of a mobile application the first thing is form factor, which is the look the aspect ratio the size and the style of the device, which influences display and influences the way you navigate. In a desk of application you will typically use a mouse, or a ruler and keyboard to navigate, but that is not possible or that is not easy in terms of a mobile device. So, you are navigation may happen in in a different way.

The presentation itself may happen in a different way, you have a very limited display area. So, you might want to stack multiple responses one after the other whereas, the in a in a web application running on a desktop, you would probably have shown them on different tabs side by side, or would have just shown them side by side on the display. Then, mobile devices typically have limited memory, they have limited computing power, very importantly most of them run on battery and therefore, they have one limited power available. So, you are applications will lead to be power optimized, which is not the case with the normal web based application, you have a wireless connection, so which may be limited in bandwidth based on your connectivity and that time, and so on. So, you while developing a mobile app corresponding to a possible web application, your considerations would be very different. So, if I say that a bank say, HDFC bank has a net banking application, which runs on the browser. And it is also having a mobile app, which runs on say the android phone then, the their requirements and their style of solutions will have to be very, very different. And at the same time if I talk about a mobile app then, the mobile devices often have features which desktop do not have for example, you have a number of senses available like, accelerometer and so on which you can make to advantage for example, we I mean in in many smart phones, if we just rotate the screen, rotate that device, then the display self-rotate automatically, which we use we use some some kind of an accelerometer inputs for that, you have a touch screen which can allow a wide range of gesture based navigation, which is typically not possible in desktop and most of the laptops.

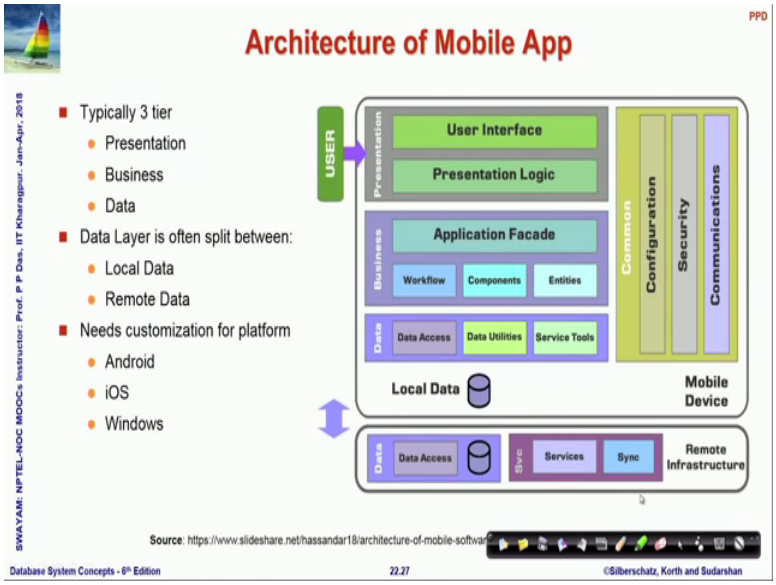
(Refer Slide Time: 29:48)



So, having said that, naturally there are there are 2 aspects of mobile apps as well. So, when we talking about mobile apps, we have talking about stand alone mobile apps, there are other ways of developing applications also for example, we can do a typical web-based application, but we can use a website which is specifically designed catering to the mobile devices. So, these send back pages, which are smaller in size are stack differently and so on. So, in contrast to that a mobile app are actually, applications that are downloaded and runs on the system.

So, these are all different possibilities and even though mobile website is also becoming popular, but certainly mobile apps are very, very common in terms of a majority of critical applications of data bases that we have.

(Refer Slide Time: 30:40)



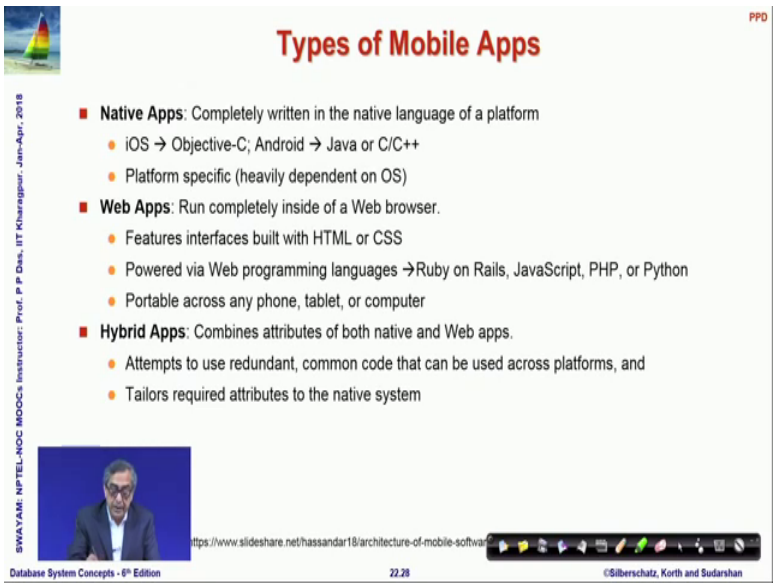
So, this is the typical architecture of a mobile app as you can see again here, that you have the 3 layers presentation, business and data the only difference being now, since you have a device, which can go anywhere and you have a connectivity.

So, this is this is what shows the connectivity, this is the device side and this is the pure backend or your service provider side. So, since your connectivity is not may not be very strong, you try to create all the layers of a presentation, business, as well as data on the mobile phone itself, but naturally all the data you will not have on your device ah, you are checking mails on the Gmail naturally, you will not have all the data on your Gmail.

So, what you do is you use the connection to connect to the remote database provided by the service, but primarily most of this layer of this presentation, business and a small part of the data layer are all realized within the phone itself, or within the mobile device itself. So, the data layer is split in this case, which is very different from how you do the web-based applications? And specifically, it might need customizations based on the kind of platform, you are using whether you have using android, iOS and windows, and these are all custom solutions for that and you could also note that, there are different types of mobile apps one large class is known as native application where, which is completely written in the native language of the platform.

So, for if you are doing an iOS mobile app then you will write it in object C, objective C if you are doing an android one you will write it in C or C plus plus or java is platform specific whereas, there is another class of mobile apps, which are known as web apps which run completely inside the web browser, so much like the way java scripts work.

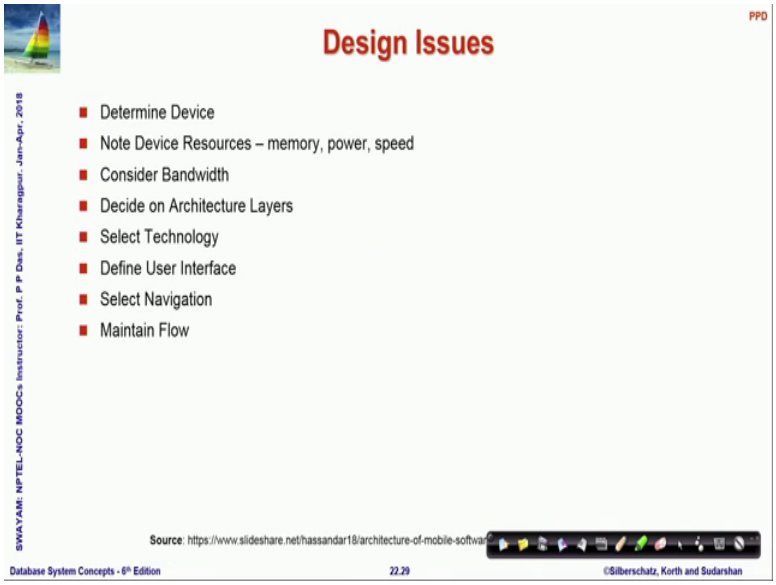
(Refer Slide Time: 32:12)



So, they feature interfaces built with HTML and the style shades, and they have powered by different web programming languages like, ruby on rails java script to PHP and so on. And certainly there is a third kind, when you combine the attributes of both native and wave application and you try to you know. So, you can very easily understand, that if your application is a native one then it is not portable across devices, this is not an iOS application is not run on android and so on.

If it is a web app app, then it is portable because it is portable across different phone tablet or computer because, you are using generic technologies. So, you could do a hybrid app also where, you could use redundant or common code, which is usable across platform and add some tailor functionality in terms of the native system. So, these are the typical kinds of mobile apps that.

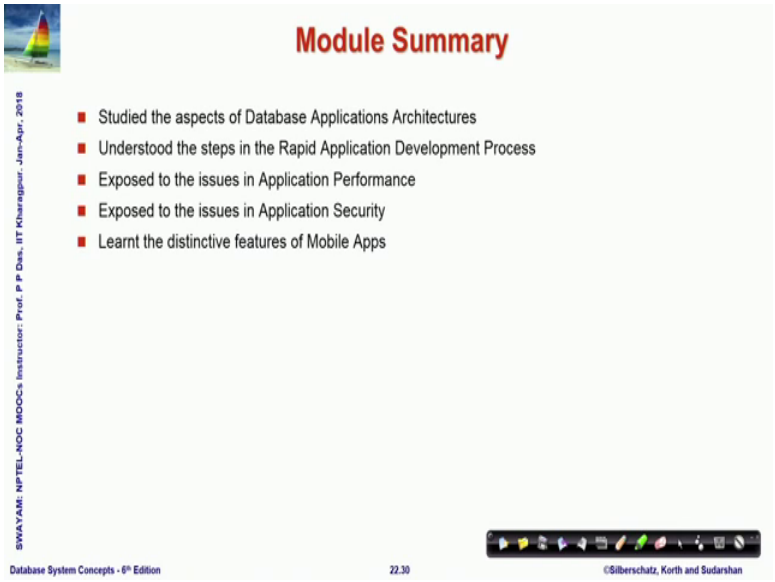
(Refer Slide Time: 33:50)



You might be developing and there a several factors to considered in the design issue, you have to first decide on the device, you have to take specific note of the typical resources, that you will your device will support memory, power, speed. You have to consider what should be the band width should it be 2 g, 3 g, 4 g or wireless you know LAN, what kind of connections you would expect? Decide on the layers in the architecture, select the technology based on the device choice and the other factors define the user interface, navigation and maintain the work flow.

So, these are very variety I will not go into details of this, but just wanted to give a glimpse of the fact, that in today's time while you are talking about database applications then it is a very reality, that you will create that as a mobile app.

(Refer Slide Time: 34:32)



So, in this module to summarize your study aspects of database application architecture, understood the steps of rapid development, and took a quick look into the issues of application performance security, and what it takes to? What are the distinctive features of a mobile app for database applications?