

Unit -3.

[Mobile User interface Design].

1. Effective use of screen Real Estate:

Screen real estate refers to the amount of space that is available on a screen. In e-learning, it is not available space on the computer screen that concern us - it is the space on the slides we design and use to teach the material that we find to be most important.

2. Principles to use screen real estate effectively.

- i) Use white space, but not too much
- ii) Narrow the margins
- iii) Use graphics and figures carefully
- iv) Use audio functions
- v) Place closed captioning strategically
- vi) Use a responsive e-learning design

[Understanding Mobile Information Design]

1. **Mobile Application Development:** Process of creating software applications that run on a mobile device, and a typical mobile application utilizes a network connection to work with remote computing resources.

- Mobile development process involves creating installable software bundles (codes, binaries, assets etc), implementing backend services such as data access with an API, and testing the application on target device.

→ **Device platforms:** Device platform refers to operating system on which these mobile applications run.

- Popular platform: Android and iOS.
- Developers create mobile application platform using different programming languages and software development tools.
- Android app can be developed using Java, Kotlin, or ATT, while iOS apps can be developed using Swift or Objective-C.

2. Approaches to building mobile Applications.

- i) **Native App Development:** This involve developing separate applications for each mobile platform, such as android and iOS. Developers use platform-specific programming languages, tools and development environment to create apps that are optimized for each platform.
- Native development allows developers to take full advantage of the unique features and capabilities of each platform, resulting in high performance apps with a better user experience.

ii) Hybrid App Development: It involves building a single app that can run on multiple platforms, using web technologies such as HTML, CSS, and Javascript.

- Hybrid apps are wrapped in a native container that enables them to access device-specific features, such as camera, GPS and contacts. Hybrid app development offers a faster development cycle and a cost effective solution for building mobile apps that can run on multiple platforms.

iii) Cross platform app development: It is similar to hybrid app development, but it uses cross platform frameworks and tools that allow developers to build apps that run natively on multiple platforms, including Android, iOS and Windows.

- Cross-platform frameworks such as React Native and Xamarin use a single codebase that can be shared across platforms, resulting in a faster development cycle and reduced costs.

iv. Progressive Web Apps (PWA) :- Web applications that can be accessed via a web browser but offer a mobile app-like experience.

- PWAs are designed to work on any device, regardless of the platform, and can be installed on the homescreen of a mobile device.
- PWAs offer a fast and responsive user experience and do not require installation or updates.

3. Mobile Application Development Lifecycle

- a) Planning
- b) Analysis
- c) Design
- d) Development
- e) Testing
- f) Deployment
- g) Maintenance.

→ Interlinked core components of a mobile application

: Front-end and backend.

a) Front-end: It is the part of the applications that user interact with directly, including the user interface (UI) and user experience (UX) design. The frontend is responsible for presenting information to the user, receiving user input, and displaying the result of user actions.

b) Back-end: The backend refers to the server side infrastructure that supports the applications functionality. It is responsible for managing data storage, processing user requests, and providing the necessary services to support the frontend of the application.

Key components of backend:

- a) Server
- b) Database
- API's
- Security
- Performance
- Scalability

→ How Frontend communicates to backend.

The frontend of a mobile application communicates to backend through APIs. The backend provide a set of API that the frontend can use to retrieve data, perform actions and interact with other systems.

When the user interact with the frontend of the mobile application, the application sends a request to the backend API. The backend then processes the request, retrieves the necessary data, and sends a response back to the front end. The front end then uses the response to update the user interface and display the result of the users actions.

For eg: lets say a user opens a mobile application and wants to view a list of products. The frontend of the application sends a request to the backend API, asking for a list of products from the database and sends the list back to the front end. The front end then uses the data to update the product list on the user interface, displaying the products for the user to view.

→ Why developers use cloud backend.

i) Scalability: Cloud based backends can be easily scaled up or down to handle changes in application usage or traffics. Cloud providers offer infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS) solutions that can be easily scaled up or down based on the application needs. This allow developers to quickly adjust resources to accomodate spikes in traffic or usage without having to make significant changes to the application infrastructures.

ii. Cost-effectiveness: Eliminate the need for developers to purchase and maintain their own servers and hardware.

iii) Security: Includes security measures like encryption and data backup.

iv) Flexibility: Allow developers to use a wide range of programming languages, frameworks, and tools to build and deploy their applications.

4 Different ways to develop Mobile apps:-

- i) 1st party Native App development
- ii) Progressive web development
- iii) Cross-platform Application.

i) 1st party Native App development:-

- These types of apps normally run in the native devices, that is run only in the OS that it is specifically designed for it.

Eg: A retail company wants to improve the in-store shopping experience for its customers. They develop a 1st party native app that allows customer to:

- Browse the store's inventory and product information.
- Create a shopping list
- Scan barcodes to view product information and reviews.
- Locate items in the store using an interactive map.
- This app is only available to the company's customers and can only be used in their physical stores. The app is designed to integrate with the company's existing systems, such as inventory management and point of sale systems.
- This app is developed by the retail company for its own use, to improve the in-store customer experience, increase sales and gain insights from the customer's behaviour.
- In this eg: the retail company is the 1st party, and the app is a native app, because it is developed for specific platform (ios or Android) and can take full advantage of the device capabilities and features.

→ Advantages.

- High performance.
- Access to all features and APIs.
- It is a large community to solve errors.
- Updates available on same day.

→ Disadvantages

- Low speed.
- Doesn't support open source.

iii) Progressive web Application:

Eg: A news website wants to provide its users with a better mobile experience. They develop a progressive web app that:

- Allows users to access the website offline by storing content on the user's device.
- Send push notification to users to alert them of breaking news.
- Can be installed on the user's home screen like a native app.
- Provides a fast and smooth browsing experience.
- Has a responsive design that adapts to different screen sizes.
- Users can access the PWA by visiting the website on their mobile browser. They are prompted to install the PWA on their home screen, which allows them to access the website offline and receive push notification.
- In this example, the news website is the 1st party and the app is a progressive web app because it can be accessed through a web browser and can be installed on the user's device like a native app. It also allows users to access the content offline and have a fast and smooth experience.

→ Advantages:

- Development is fast as same code is used for Android, iOS, web applications.
- No installation required.

→ Disadvantages:

- Don't have access to all features, so UX is not that good.
- iOS does not support all the features of PWA.
- The UI for development is bespoke i.e. the buttons, edit texts needs to be programmed which was not necessary for the 1st party native apps.

iii) Cross-platform Application.

- Eg: A project management company wants to create a project management tool that can be used by teams on different platforms. They develop cross-platform application that:
- Can be used on windows, Mac, iOS and Android devices
 - Allows users to create and assign tasks, set deadlines, and track progress.
 - Integrates with popular tools such as Google calendar and Trello.
 - Has a user-friendly interface that works seamlessly across all platforms.
 - The application can be downloaded from the company's website or from different app stores such as App store, Google play store, Microsoft store and Mac App store, depending on the platform.
 - This eg: illustrates how the company developed a project management tool that can be used by teams on different platforms, windows, Mac, iOS and Android, which is a cross platform application. It allows team to collaborate and manage

their products seamlessly, regardless of the platform they use.

→ Advantages.

- Development speed is high. Same code is used for android and ios.
- Maintenance cost is low as the errors and updates as to be encountered only once.

→ Disadvantages.

- Slow code performance with limited tool Availability.
- Limited user experience i.e. These apps does not have access to Native only features.

* Mobile Applications Eg : Whatsapp, Instagram, Uber, YouTube.

5. Pros of Mobile Application.

- i) Offline access
- ii) Extensive capabilities
- iii) Personalisation → get preferences, creating account etc.
- iv) Brand promotion opportunities.
- v) User experience
- vi) Speed performance
- vii) Functionality.

6. Cons of Mobile App.

- i) Support and maintenance
- ii) Cost
- iii) Compatibility
- iv) Use of user's resources

7. Websites:

- Collection of web pages that are linked with each other where the web pages share the single domain name.
- Websites are published on any web server and this website is publicly accessible.
- Websites can be accessed on any digital platforms supported with search engines or web browsers. for eg: a web browser site can be accessed on a mobile, in a computer, and in tablet.
- Websites commonly referred to as simply sites only.
- They are particularly dedicated to any topic like these are developed for various purposes like social networking, education, news, business organizations, entertainment etc.
- Eg: Netflix, Instagram, Amazon, YouTube etc.

→ Pros:

- i) Wide Reach
- ii) Cost-effective
- iii) Convenient
- iv) Improved credibility
- v) Branding.

→ Cons:

- i) Technical issue
- ii) Initial Investment
- iii) Time-consuming
- iv) Competition
- v) Accessibility.
- vi) No offline access

8. Advantages of Mobile Applications over website.

- i) Better user experience
- ii) Offline functionality
- iii) Improved performance
- iv) Personalisation
- v) Increased engagement
- vi) Access to device features

[chapter-3.2]

[Mobile Interface Design Tools].

1. User Interface tools:- Tools are software programs or framework that allow developers and designers to create and design user interfaces or applications or websites.

2. UI Design tools:-

i) Sketch:- Vector graphics editor for designing user interfaces, with features like symbols and artboards that make it easy to create reusable UI elements.

- Shorthand and math operators to speed up the design process.
- Variable and openType fonts for infinite control over your interface typography.
- Cross platform tools for real-time collaboration, feedback sharing and developer hand off.

ii) Adobe XD:- UX design software that allows designers to create wireframes, high fidelity mockups, and interactive prototypes.

- Vector based drag and drop editor tool with unlimited artboards and smart guides that help you to align different objects and elements.
- 3D Transforms
- Voice prototyping, enabling you to create voice commands.

- iii) UXpin: End to end platform capable of delivering polished, interactive prototypes - no coding skill required.
- Built in libraries for iOS, Google Material Design, Bootstrap and User Flows, full of ready to use interactive elements, colours, text styles and icons.
 - Built in contrast checker and colour blindness simulator to help you ensure your designs are as accessible and inclusive as possible.
- iv) Figma: Web based design tools that enable real-time collaboration among designers and developers, making it easy to design and prototype UIs.
- Modern pen tool that allows to draw in any direction with vector network.
 - Auto layout for easy responsive design.
 - Accessible libraries with ready-made assets which you can drag and drop in your design files.
- v) In-vision studio:
- Vector drawing tool for lightning-fast screen design.
 - Adaptive layout for responsive design; enabling you to quickly and easily adjust and scale your designs to fit any screen size.
 - Built-in animation capabilities, including smart-swipe transitions, timeline editing and auto-layer linking.

3. Web Development: Web development involves coding, designing, and operating web applications and websites. The programming languages needed for web development are HTML, CSS, Javascript. Web development can further be divided into 3 categories.

i) Front-End web development

ii) Back-End web development

iii) Full-Stack web development

Web development involves the use of various frameworks, libraries, and tools such as Angular, React, Vue.js, Django, Ruby on Rails, and more, which help developers to build websites faster and more efficiently.

4. Difference between

Mobile Apps

i) Mobile apps are developed for specific platform, such as iOS or Android.

ii) They are downloaded and installed via an app store such as Google play store and Apple store and have access to system resources, such as GPS and the camera.

iii) Mobile app may work offline.

iv) Mobile Apps are faster.

v) Mobile apps have more safety and security.

Web Apps.

ii) Web apps are accessed via the internet browser and will function according to device you are viewing them on.

iii) Web apps are not native to a particular system and there is no need to be downloaded or installed.

iii) It needs active internet connection.

iv) Web apps are slower.

v) less secure.

5. Web apps:- Software application that is accessed through a web browser over the internet. Built using web technologies such as HTML, CSS and Javascript.

→ Pros.

→ Cons.

- i) Platform-independent
- ii) Easy to update
- iii) Low development cost
- iv) Easy to maintain
- v) Scalable.
- vi) Limited functionality
- vii) Performance limitations
- viii) Security concerns
- ix) Browser compatibility
- x) Lack of offline access.

6. Responsive design and adaptive design.

Responsive design is an approach where the website layout and design adapt to the screen size of the device being used to view it. The website responds to changes in the device screen size, adjusting the placement and size of images, text, and other elements to optimize the user experience. This approach typically involves using flexible grids, images and CSS media queries to adjust the website layout and design for different screen sizes.

Adaptive design, on the other hand, involves creating multiple versions of a website, each designed specifically for a different device type or screen size. This approach detects the type of device being used and serves up the appropriate version of the website, optimized for that device. This approach can provide a more tailored experience for users on each device type, but can be more time consuming and costly to develop multiple version of the websites.

7. Dedicated Mobile apps: Software applications designed specially for mobile devices such as smartphones or tablets. Unlike web apps, which are accessed through a web browser, dedicated mobile apps are installed directly onto a user's mobile device and can be accessed without an internet connection.

8. HTML 5 application: HTML 5 applications, also known as HTML 5 web apps, are software applications that are built using web technologies such as HTML, CSS and Javascript, and are accessed through a web browser.

→ Features:

- Cross platform compatibility
- Reduce development time and cost
- Easy deployment
- Scalability
- Flexibility

9. What is Android?

Android is a mobile operating system based on a modified version of the Linux kernel and other open-source software, designed primarily for touchscreen mobile devices such as smartphones and tablets.

10. Android components.

i) Activities: Deals with the UI and the user interactions to the screen. An Android application typically consists of multiple activities that interact with each other.

Public class Mainactivity extends Activity { }

ii) Services: Component that runs in the background to perform long running operations. It doesn't have user interface and can run even

if the user switches to other app.

public class MyService extends Service { all code for service }.

iii) Broadcast receivers: Component that receives and responds to system wide broadcast announcements. For eg: a broadcast receiver can be used to receive a notification when the battery is low.

public class MyReceiver extends BroadcastReceiver { public void onReceive(Context context, Intent intent) {} }

iv) Content provider: Component that manages a shared set of app data. Allows different app to access and share the same data, such as contact or media files.

public class MyContentProvider extends ContentProvider { public void onCreate() {} }

i) How to create an Android project.

ii) Open Android Studio and click on "Start a new Android project" on the welcome screen or go to File > New > New project.

iii) Choose "Phone and tablet" template and select the type of app you want to create, such as "empty activity" or a "basic activity".

iv) Give your project a name and select the programming language you want to use, such as Java or Kotlin.

v) Choose the minimum API level your app will support. This determines the oldest version of android that your app can run on.

vi) Choose the target SDK version, which is the highest version of android app that your app is designed to run on.

vii) Choose the device form factors you want to support, such as phone or tablet.

viii) Choose the default orientation of your app like portrait or landscape

ix) Click "Finish" to create the project.