CS 5520
MOBILE APPLICATION DEVELOPMENT

Week 1

NEDA CHANGIZI

Today's Learning Outcomes

- Explain the course logistics
- Set up the tools used in this course
- Compare and contrast different approaches to mobile development

Who Am I?

SFU SIMON FRASER UNIVERSITY

- B.Sc. in Electrical Engineering, M.Sc. In Computer Science
- 8 years professional experience in software industry
- Have been teaching at British Columbia Institute of Technology (BCIT) since Jan 2019
- Started teaching at Northeastern University in Spring 2022.
- Office hours - Please make an appointment by emailing me/messaging me on Teams: ne.changizi@northeastern.edu
  - Mon 1:30 – 2:30
  - Wed 11 – 12
  - Thurs 11:30 – 12:30
- When sending me an email:
  - Include “CS 5520” in your email subject line
  - Include your full name
  - A clear question
  - Allow 2 business days to reply

Course Outline

- Flipped learning:

https://www.washington.edu/teaching/topics/engaging-students-in-learning/flipping-the-classroom/

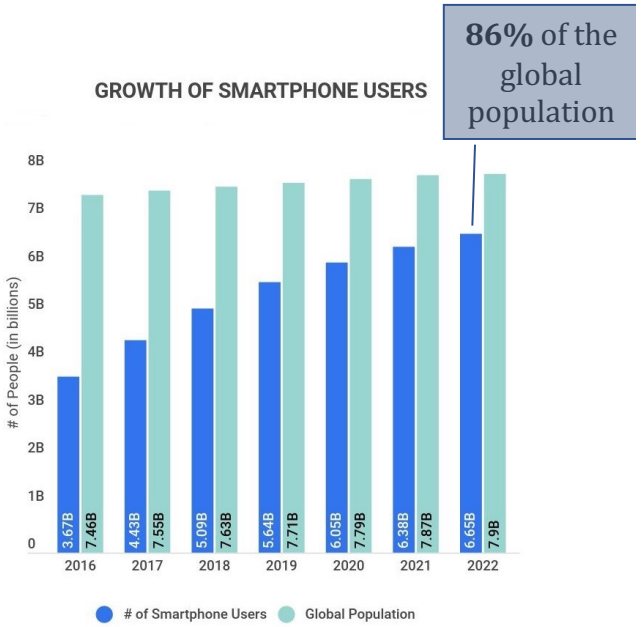
- Evaluation Criteria

Criteria	%	Comments
Quizzes	15	Due before every class as well as at the beginning of some classes.
Assignment 1	20	
Assignment 2	25	
Assignment 3 - Group Project	40	2-3 members

Mobile Apps Stats

- The Apple App Store has 1.96 million apps available for download.
- There are 2.87 million apps available for download on the Google Play Store.
- As of 2021, roughly **90%** of mobile time is spent using apps.

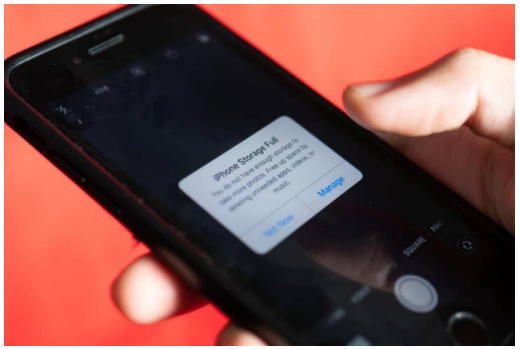
https://buildfire.com/app-statistics/

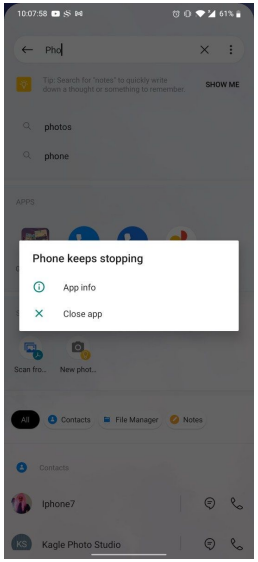


https://www.zippia.com/advice/smartphone-usage-statistics/

Mobile Application Development

- Creating software applications that run on a mobile device and utilizes a network connection to work with remote resources.
  - Does mobile app development only need front-end implementation?
  - What are some challenges/limitations of writing code to run on mobile apps?

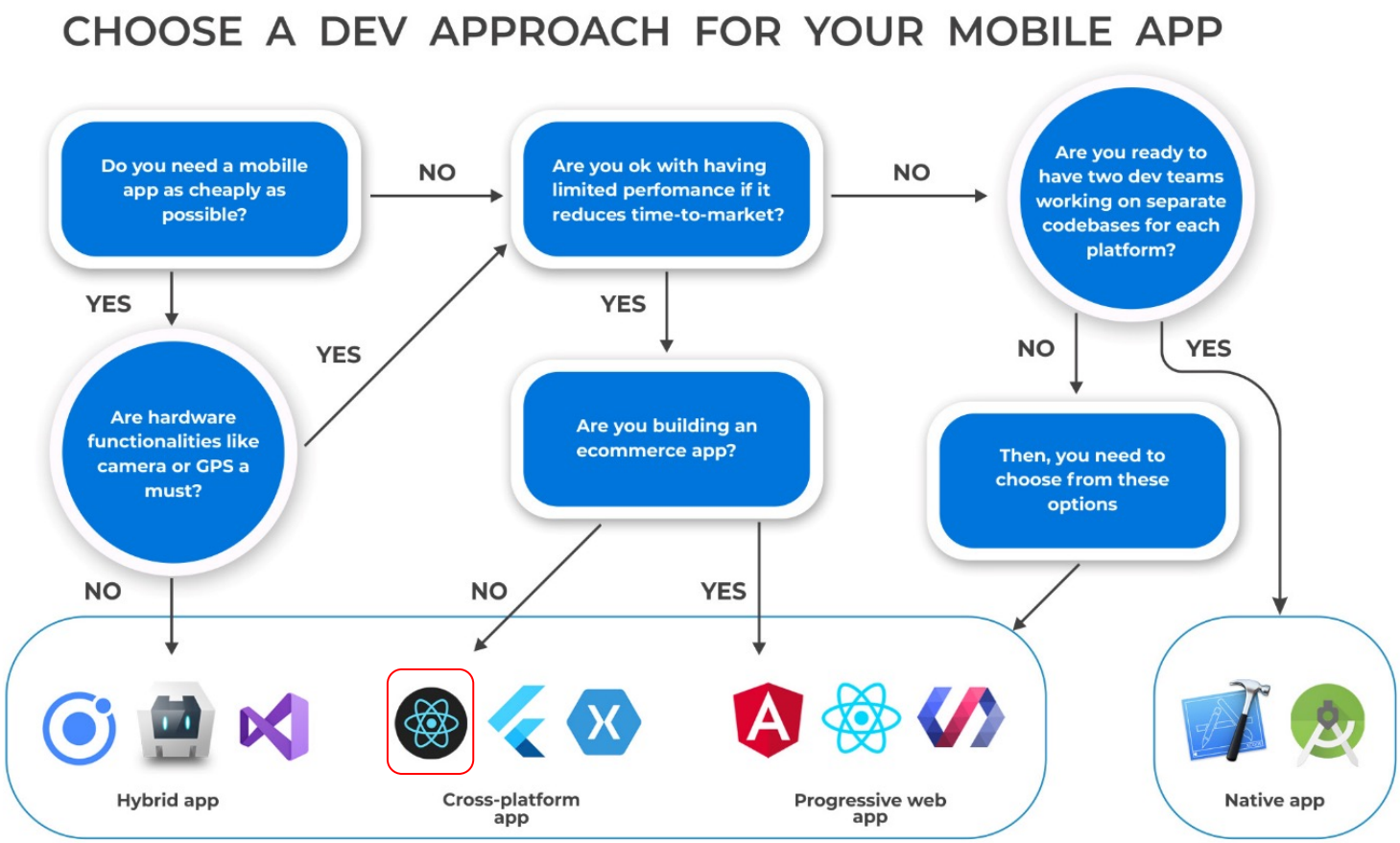




Mobile Application Development – Approaches

- Alternatives for Building Mobile Apps
  - **Native Mobile Applications** (Objective-C/Swift, Java/Kotlin)
  - **Cross-Platform Native Mobile Applications** (React Native, Xamarin, Flutter)
  - **Hybrid Mobile Applications** (CORDOVA, ionic. PhoneGap)
  - **Progressive Web Applications** (HTML, CSS, JS)

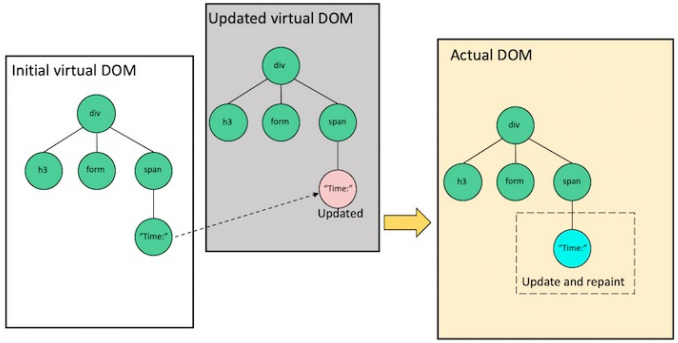
- Discuss some pros and cons of different approaches.
- What are some points to consider when choosing a mobile app development approach?



https://www.3kits.com/blog/mobile-app-development-company

What is React?

- A JavaScript library for building user interfaces
  - Components: reusable modules that renders a part of our app
  - Declarative: declares how the UI (components' states) looks given different triggers
  - Re-rendering UI on change is *optimized*
    - Virtual DOM : Only updates parts of the page that are affected by the update
    - Batch update mechanism

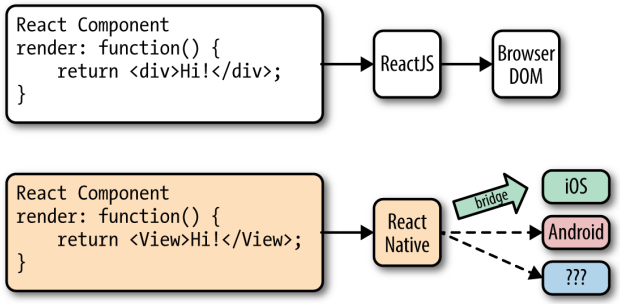


https://blog.logrocket.com/virtual-dom-react

What is React Native?

"Learn once, write anywhere"

- An open-source framework for building Android and iOS applications
  - It uses React.js library and the app platform's native capabilities
    - The code is written in JavaScript and JSX, but the user gets native app experience

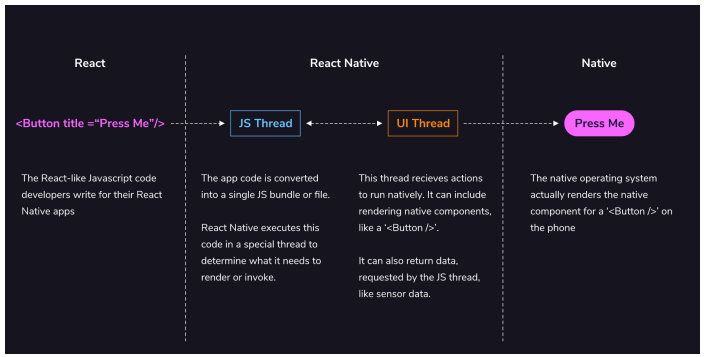


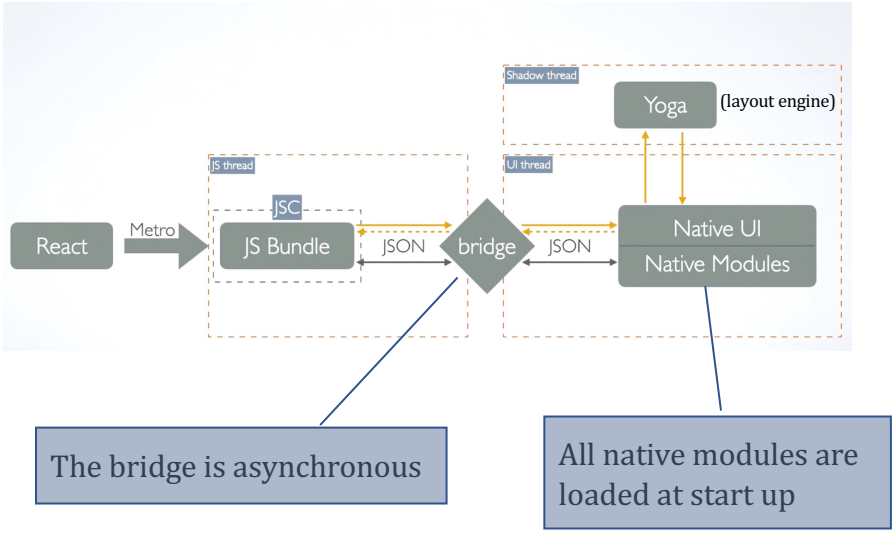
https://www.oreilly.com/library/view/learning-react-native/9781491929049/ch02.html

- It was released by Facebook in 2015.
- Who's using React Native?

React Native – bridge

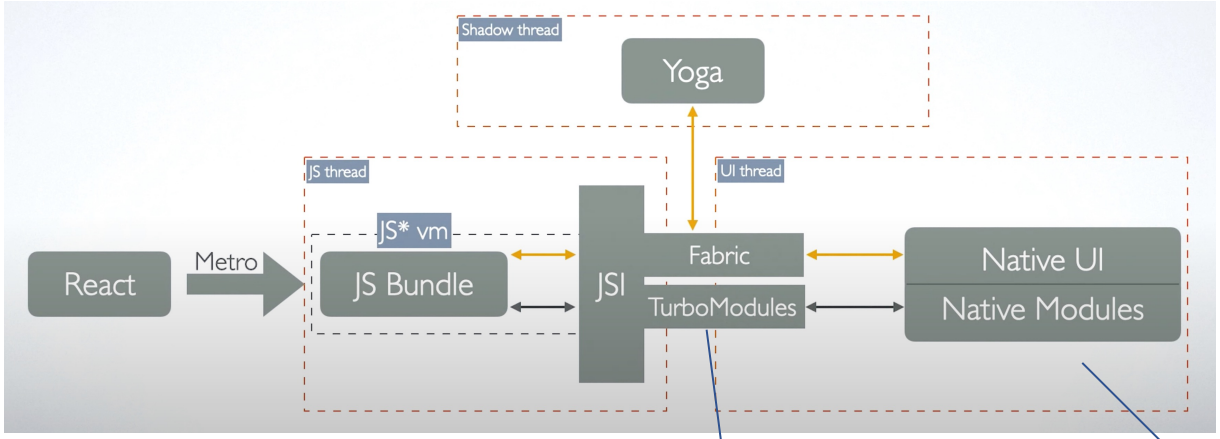
- The communication overhead between the JavaScript thread and the Native UI thread which happens over the React Native bridge can affect the app performance.





New Architecture

- JavaScript Interface (JSI), which is in C++, enables through awareness and direct communication between the JS side and the native side.



The New Architecture is Here...What now?

Direct reference between the two realms

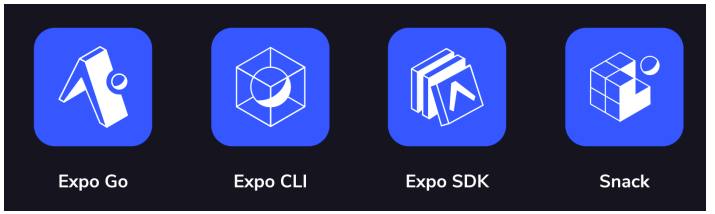
We only load native modules when they are needed

React Native – Advantages and Disadvantages

- Advantages:
  - Better performance/user experience compared to hybrid apps
  - Access to platform's specific capabilities
  - Easy update push to already installed apps
  - One codebase

- Disadvantages:
  - Slower than purely native apps (because of React Native bridge)
  - Fast-paced growing framework

Expo



- A framework and platform that helps you get started with React Native very fast
  - Limitations

- Expo CLI: command-line tool to serve your project. Requirements:
  - Node.js
  - Git
  - Watchman, required only for macOS or Linux users

- Expo SDK is a modular set of packages that provide access to native APIs, like Camera or Notifications.

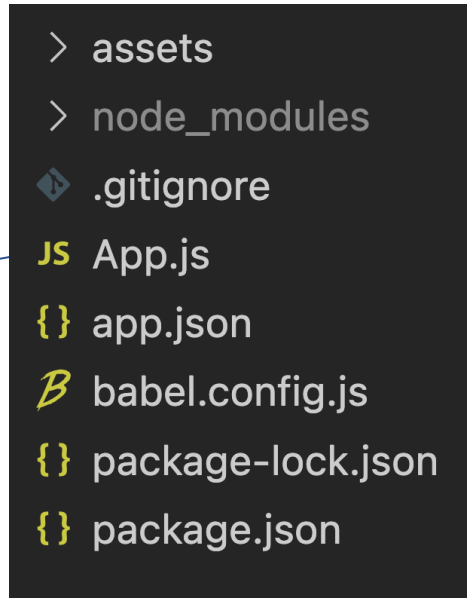
- To try out React Native directly in the browser without installing any tools, you can try out Snack (in-browser editor)

Create a new React Native project

```
npx create-expo-app <name of your project>
```

- Open the new folder that was created for your project in an IDE
  - e.g. Visual Studio Code

Entry point of the app;
The root component

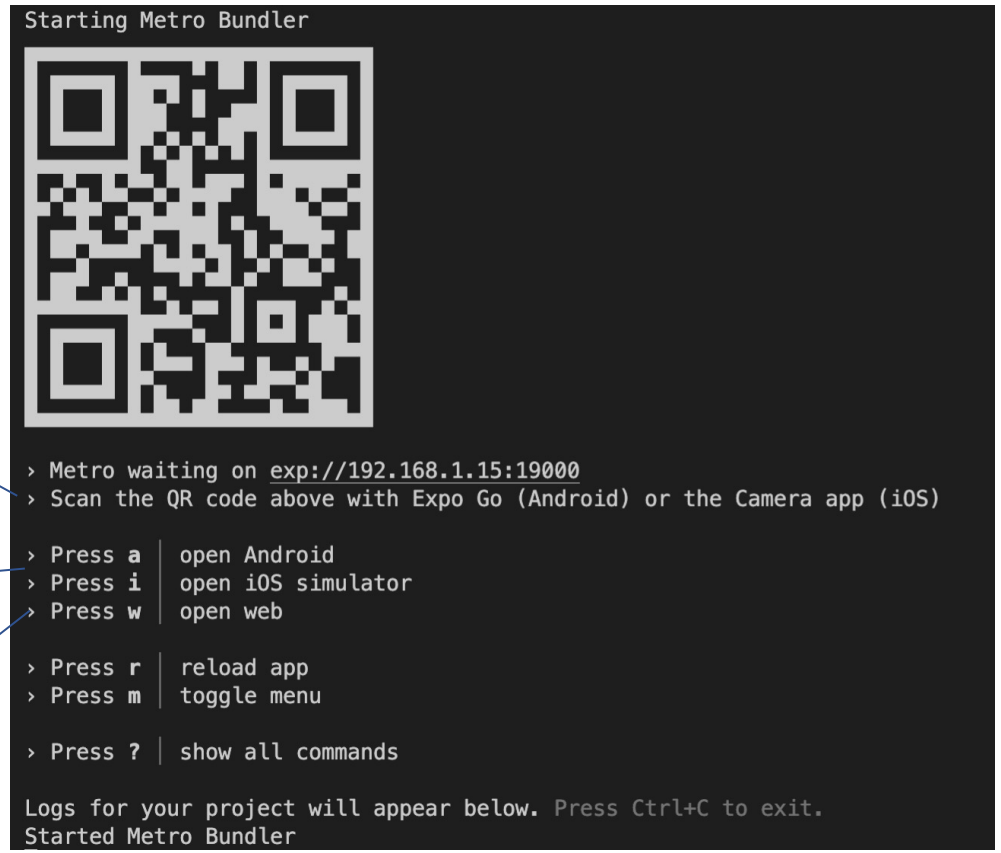


- What is the purpose of package.json?
- Run this in a terminal in the folder of your project
  - This will start a local development server

```
npx expo start
```

Running the application

- Expo CLI terminal UI:



A mobile client called Expo Go

iOS simulator (on macOS) or Android Emulator
Have these installed before the next class

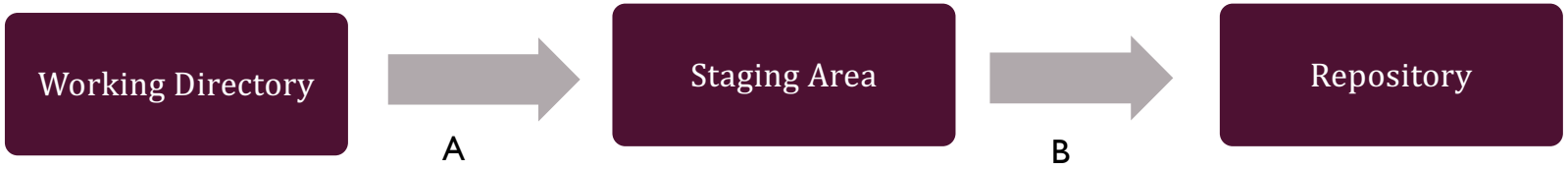
Any web browser

Version Control – git

- Verify your git installation: (terminal on Mac, PowerShell on Windows)

```
git --version
```

- What is the purpose of having a repository in Git? How can you create one?
- What is the purpose of making a commit in Git? What are the steps to make one?
- What commands are used in A and B?



- Are these areas 3 separate physical spaces?

Configure Git Settings

- Run the following commands in your PowerShell/terminal:

```
git config --global user.name "Your Name"
git config --global user.email "youremail@yourdomain.com"
```

- Once you have done the above run the below to confirm

```
git config --list
```