



SIT32004

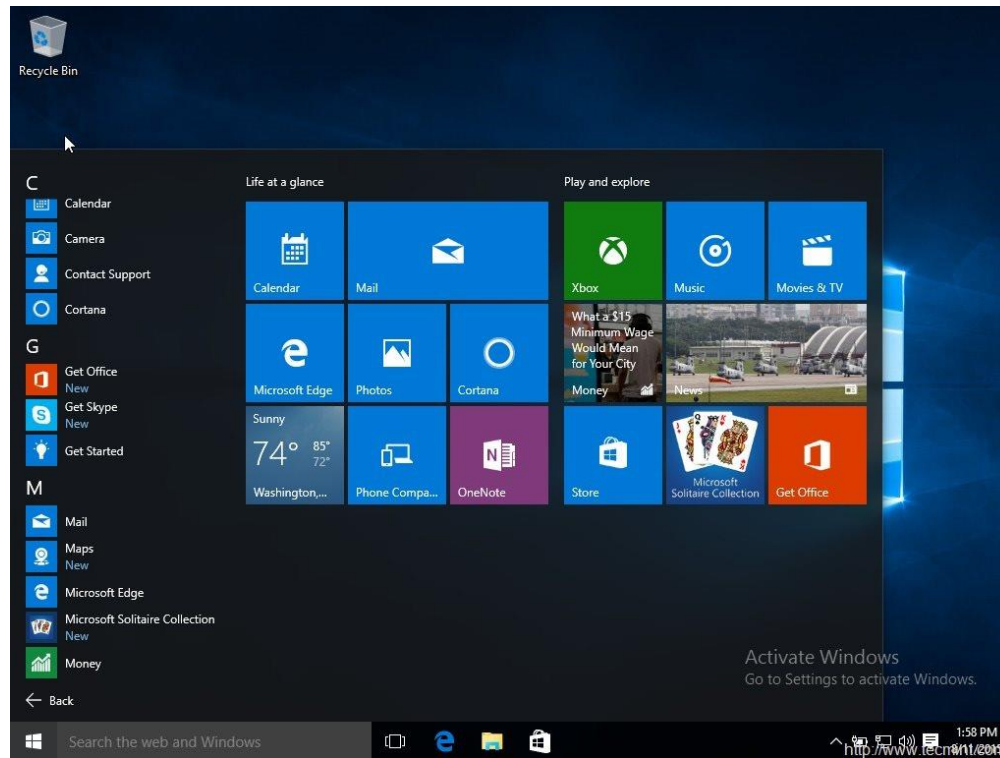
ICT Application Development

Lecture 01
Introduction



Application

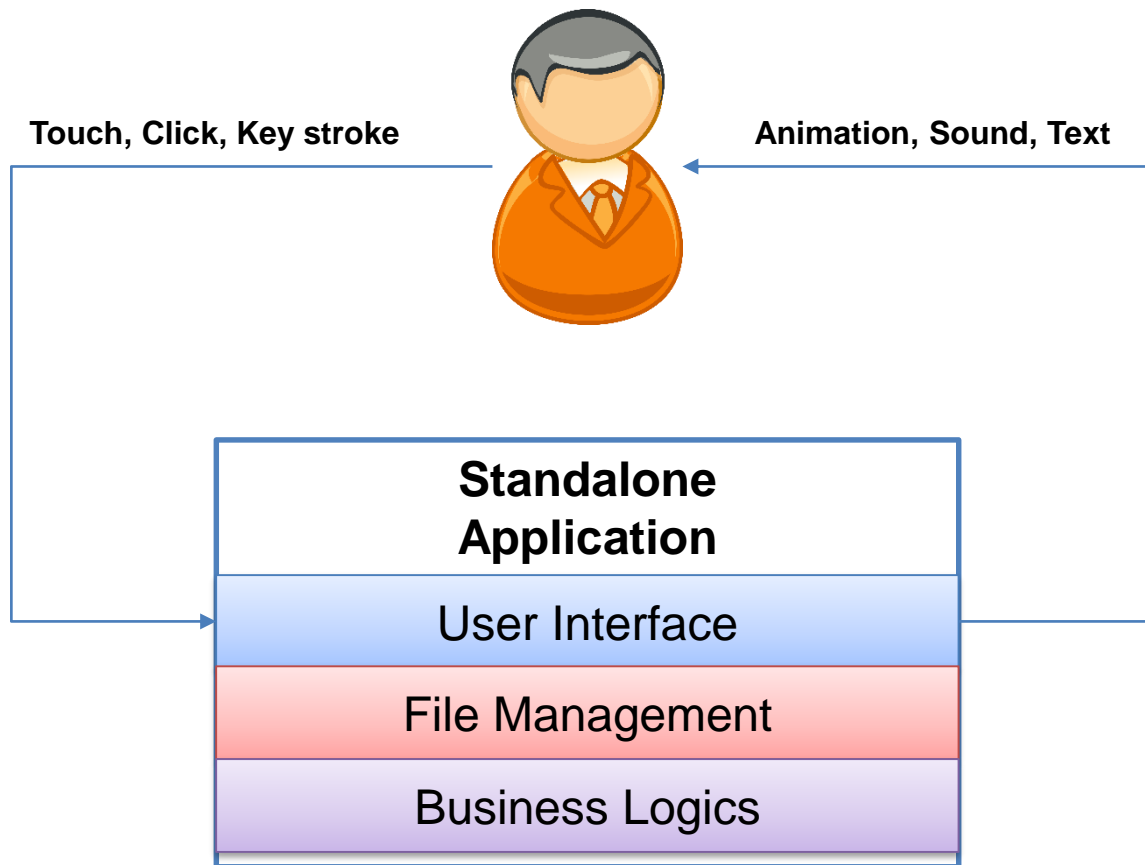
- What is computer application?
 - Definition
 - » **Application** is any material, product or a program which is **designed for endgame-user to use**. Application may refer to **computer software** designed to **help the user to perform specific tasks [1]**



List of Windows Application

Types of Application: Application's Perspective (1/2)

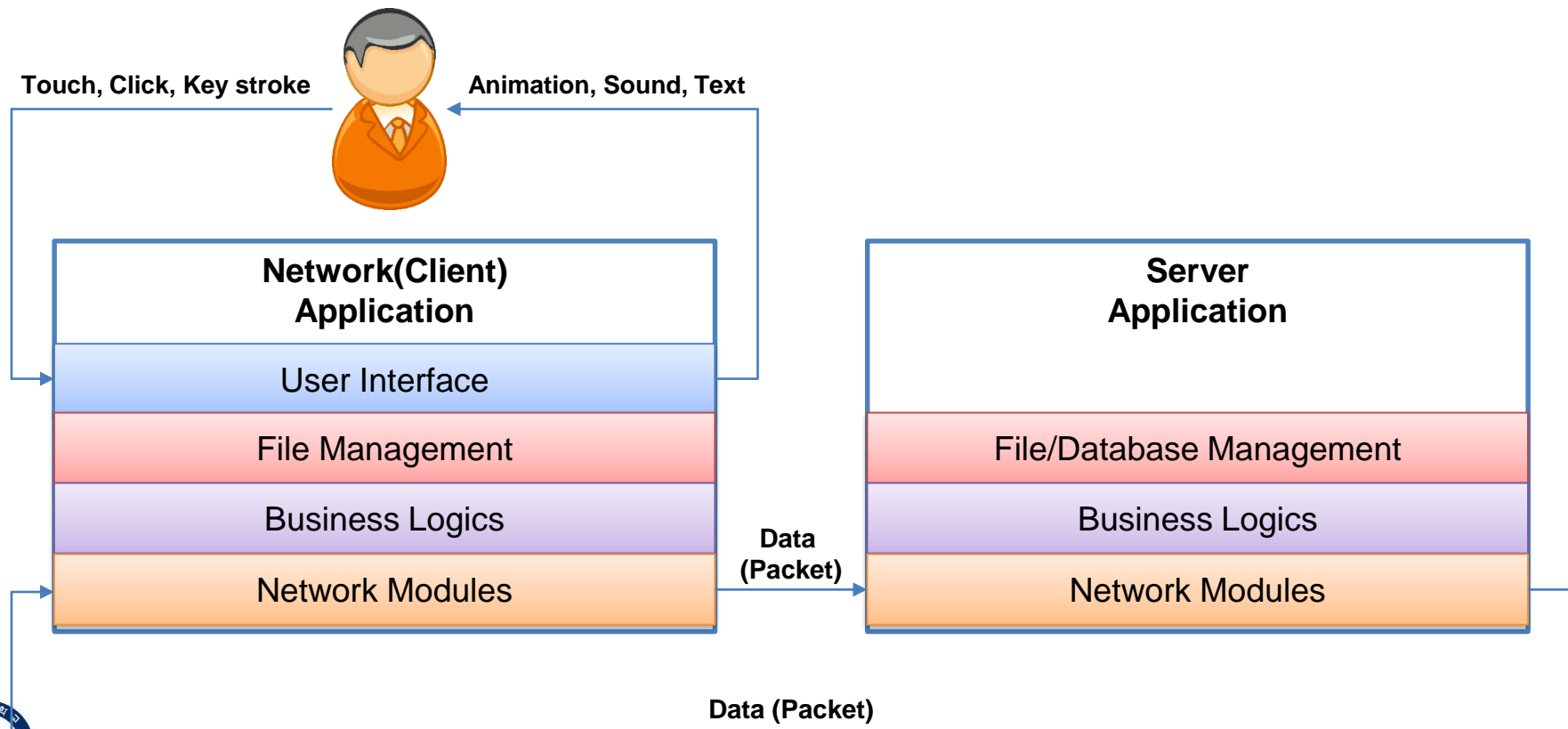
- Standalone Application
 - The application does not interact with other application
 - The application handles users action by itself



Types of Application: Application's Perspective (2/2)

- Network(Client) Application

- The application may handle users' actions by itself and may send the users' to the server
- The Client may not have great computation power, but the Server has great computation power to cover several clients' requests



Types of Application: Hardware's Perspective

- Desktop Application vs Mobile Application
 - Characteristics of Desktop Application
 - » High computation power
 - » May not use the network features
 - » Usage: Office work, Games
 - » Input Device: Keyboard, Mouse, Mic.
 - » Output Device: Monitor, Speaker
 - Characteristics of Mobile Application
 - » Low computation power
 - » Network enabled
 - » Usage: SNS, Casual Games, etc.
 - » Input Device: Camera, Touch Screen, Pen, Mic.
 - » Output Device: Screen, Speaker, Motor



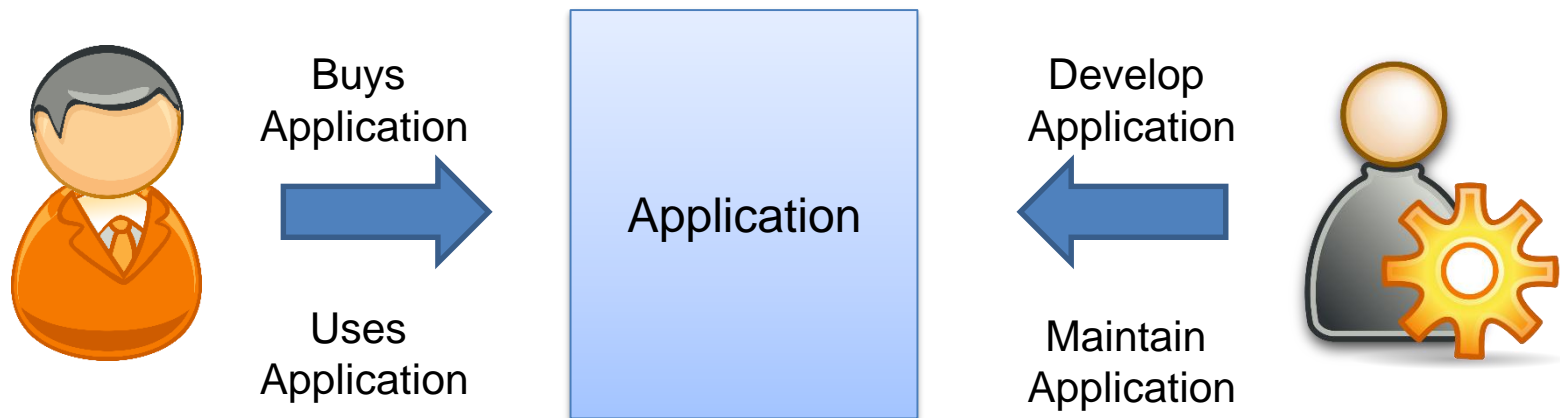
Desktop Application



Mobile Application

Users

- Types of users
 - Customers [2]
 - » A customer (sometimes known as a client, buyer, or purchaser) is the recipient of a good, service, product
 - Developer [3]
 - » One who programs computers or designs the system to match the requirements of a user



What is good application – Customer's Perspective (1/3)

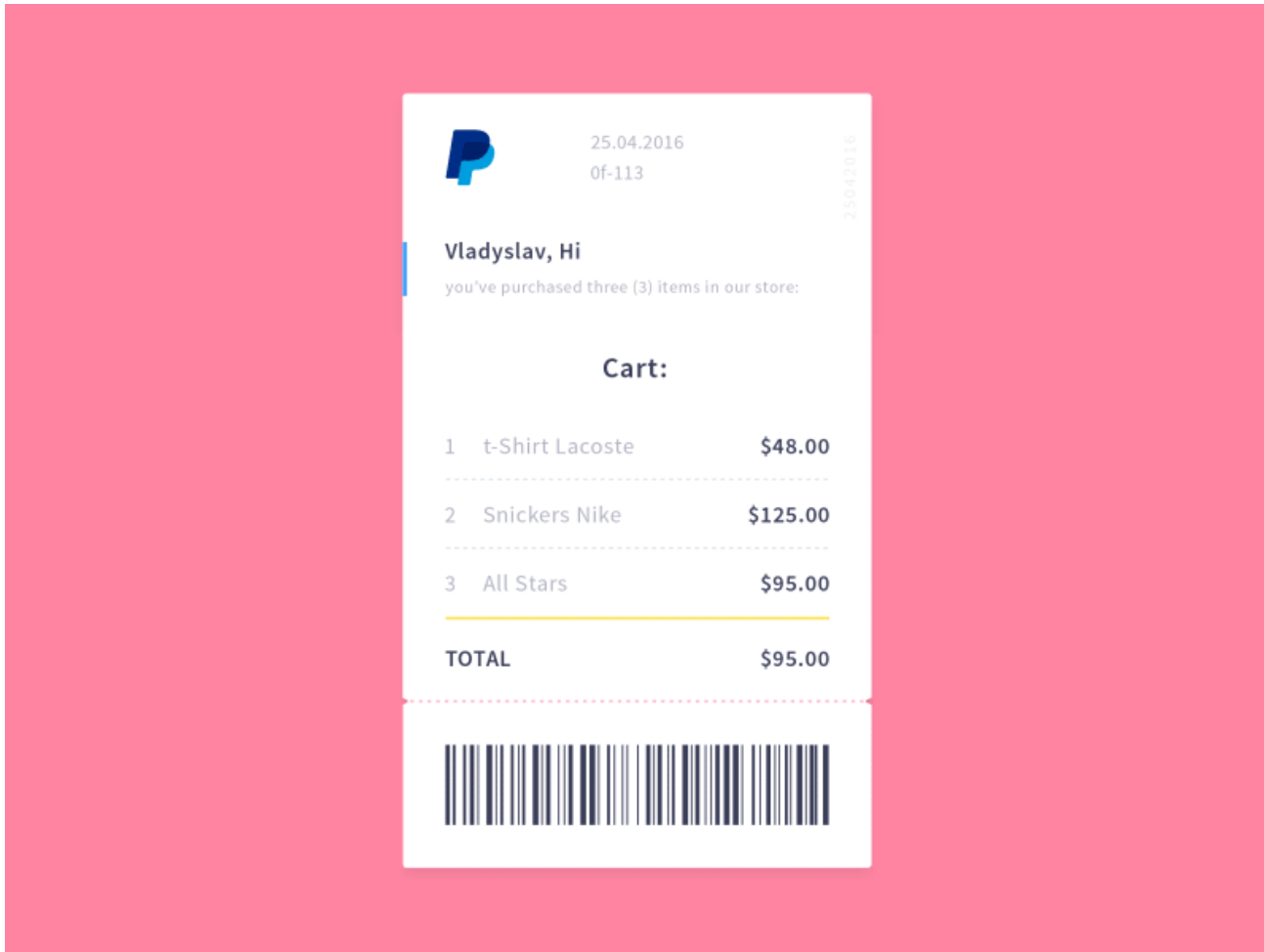
- Good Performance
 - User does not want to wait
 - If your application terminates a task in a few second, you can make application to finish the task in minutes easily

Factorial.py
<pre># calculate 10! import time def factorial(num): if num == 1: return 1 else: return factorial(num -1) * num value = 10 start = time.clock() print("10!= " + str(factorial(value))) end = time.clock() print("Elapsed Time:" + str(end - start) + " seconds")</pre>
<pre>10!= 3628800 Elapsed Time:0.0001524 seconds</pre>

Factorial2.py
<pre># calculate 10! import time def factorial(num): if num == 1: return 1 else: print("Caluating " + str(num) + "!") return factorial(num -1) * num value = 10 start = time.clock() print("10!= " + str(factorial(value))) end = time.clock() print("Elapsed Time:" + str(end - start) + " seconds")</pre>
<pre>Caluating 10! Caluating 9! Caluating 8! Caluating 7! Caluating 6! Caluating 5! Caluating 4! Caluating 3! Caluating 2! 10!= 3628800 Elapsed Time:0.0014502 seconds</pre>

What is good application – Customer's Perspective (2/3)

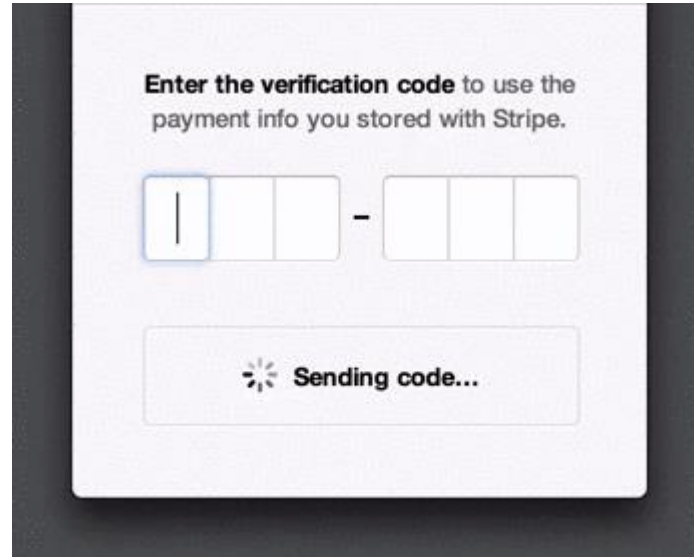
- Consider design and user experience at the same time



Author/Copyright holder: Vladyslav Tyzun. Copyright terms and license: Fair Use.

What is good application – Customer's Perspective (3/3)

- Consider design and user experience at the same time
 - Do animation *purposefully*,



Author/Copyright holder: Stripe, Inc. Copyright terms and license: Fair Use.

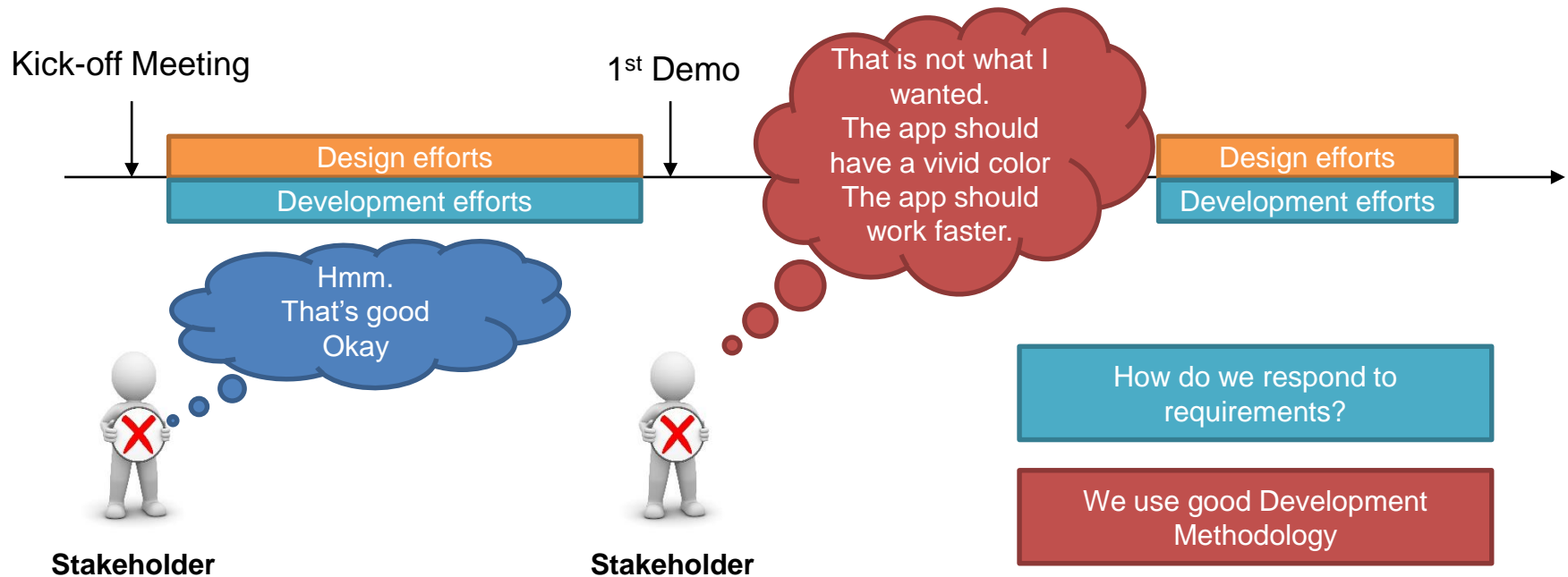
Best Practice

- Understand what your users need
 - Deliver that information
- If you've got lots of information to convey, try using visuals instead of text
- Always label your links
 - User does not like mystery links
- Avoid adding any kind of friction to user action
 - Unless they're meant to dissuade the action
- Animation is like cursing
 - If you overuse it, it loses all its impact

What is good application – Developer's perspective

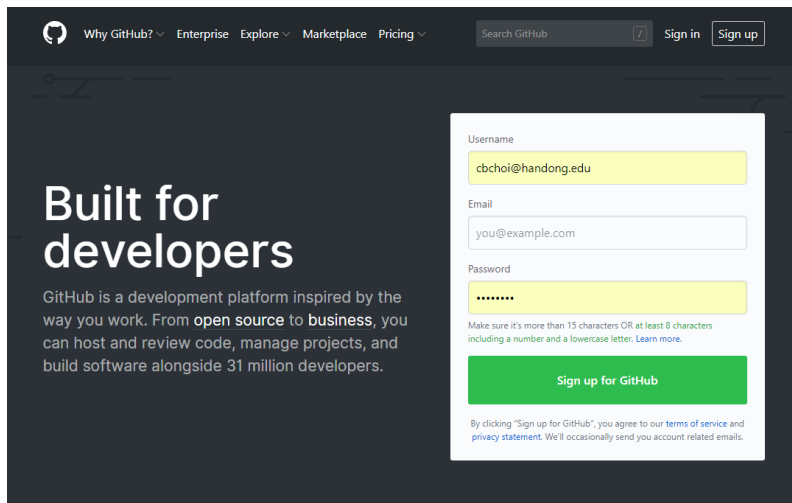
- Development Methodology
 - Not to re-do your efforts
 - People don't know what they want until you show it to them

Project Timeline

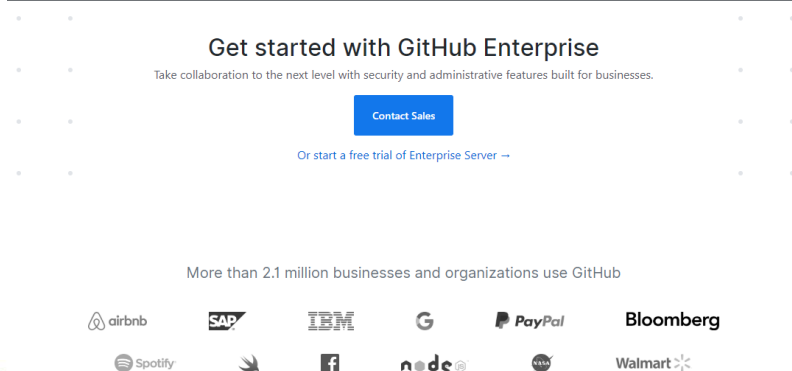


What is good application – Developer's perspective

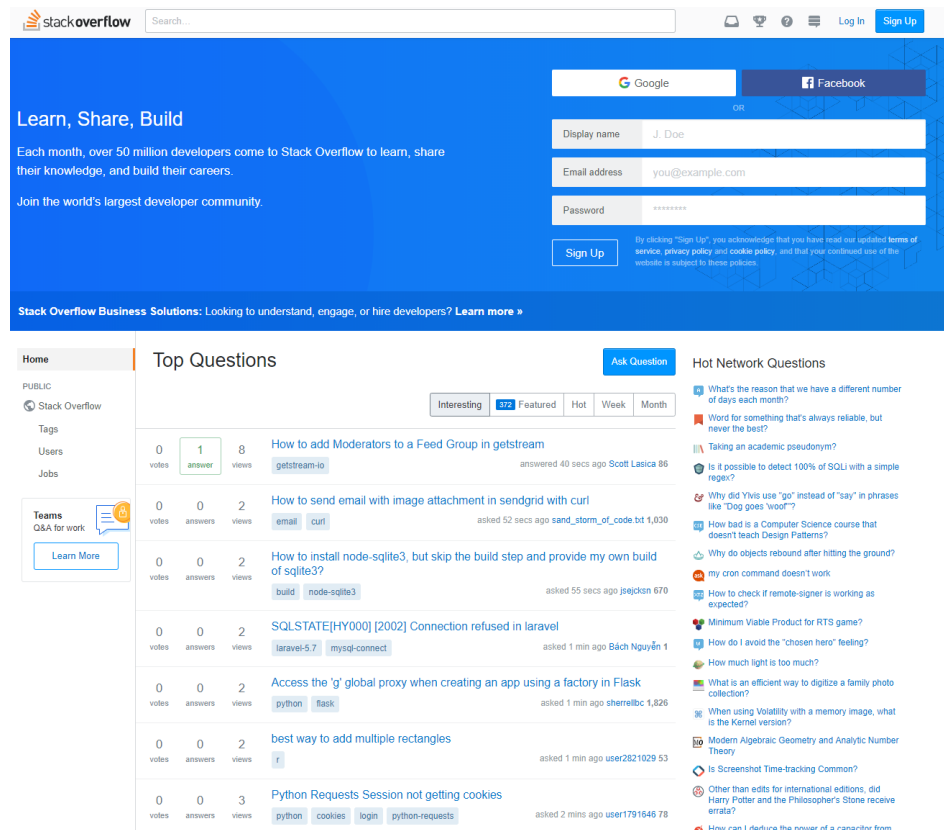
- How to collaborate with others
 - You cannot develop an application by yourself in general
 - There are several tools to help collaboration with others



The image shows the GitHub sign-up page. It features a dark header with navigation links like 'Why GitHub?', 'Enterprise', 'Explore', 'Marketplace', and 'Pricing'. A search bar and 'Sign in'/'Sign up' buttons are also present. The main content area has the heading 'Built for developers' and a subtext explaining GitHub's mission. Below this is a sign-up form with fields for 'Username' (filled with 'cbchoi@handong.edu'), 'Email' (filled with 'you@example.com'), and 'Password' (filled with '*****'). A green 'Sign up for GitHub' button is at the bottom. A small disclaimer at the bottom of the form states: 'By clicking "Sign up for GitHub", you agree to our terms of service and privacy statement. We'll occasionally send you account related emails.'



This section promotes GitHub Enterprise. It starts with the heading 'Get started with GitHub Enterprise' and a subtext: 'Take collaboration to the next level with security and administrative features built for businesses.' Below this is a blue 'Contact Sales' button. Further down, it says 'Or start a free trial of Enterprise Server --'. At the bottom, it states 'More than 2.1 million businesses and organizations use GitHub' and lists logos for various companies including Airbnb, SAP, IBM, Google, PayPal, Bloomberg, Spotify, Facebook, Node.js, Walmart, and others.



The image shows the Stack Overflow homepage. It has a blue header with the 'stackoverflow' logo, a search bar, and links for 'Log In' and 'Sign Up'. Below the header is a section titled 'Learn, Share, Build' with a description: 'Each month, over 50 million developers come to Stack Overflow to learn, share their knowledge, and build their careers. Join the world's largest developer community.' There is a 'Sign Up' button and a disclaimer. Below this is a section for 'Stack Overflow Business Solutions'. The main content area is divided into 'Home' (with links to PUBLIC, Stack Overflow, Tags, Users, Jobs) and 'Top Questions'. The 'Top Questions' section lists several questions with their respective vote counts, answer counts, and view counts. For example, the first question is 'How to add Moderators to a Feed Group in getstream' with 0 votes, 1 answer, and 8 views. The 'Hot Network Questions' section is also visible on the right side.



github.com

stackoverflow.com

12 / 17

Considerations for developing an application

- Considerations
 - You should analyze the user's requirements
 - You should choose the hardware
 - You should choose the operating system
 - You should choose the programming language
 - You should choose the types of the application
 - You should consider about the design and the user's experience

Windows 10 on ARM

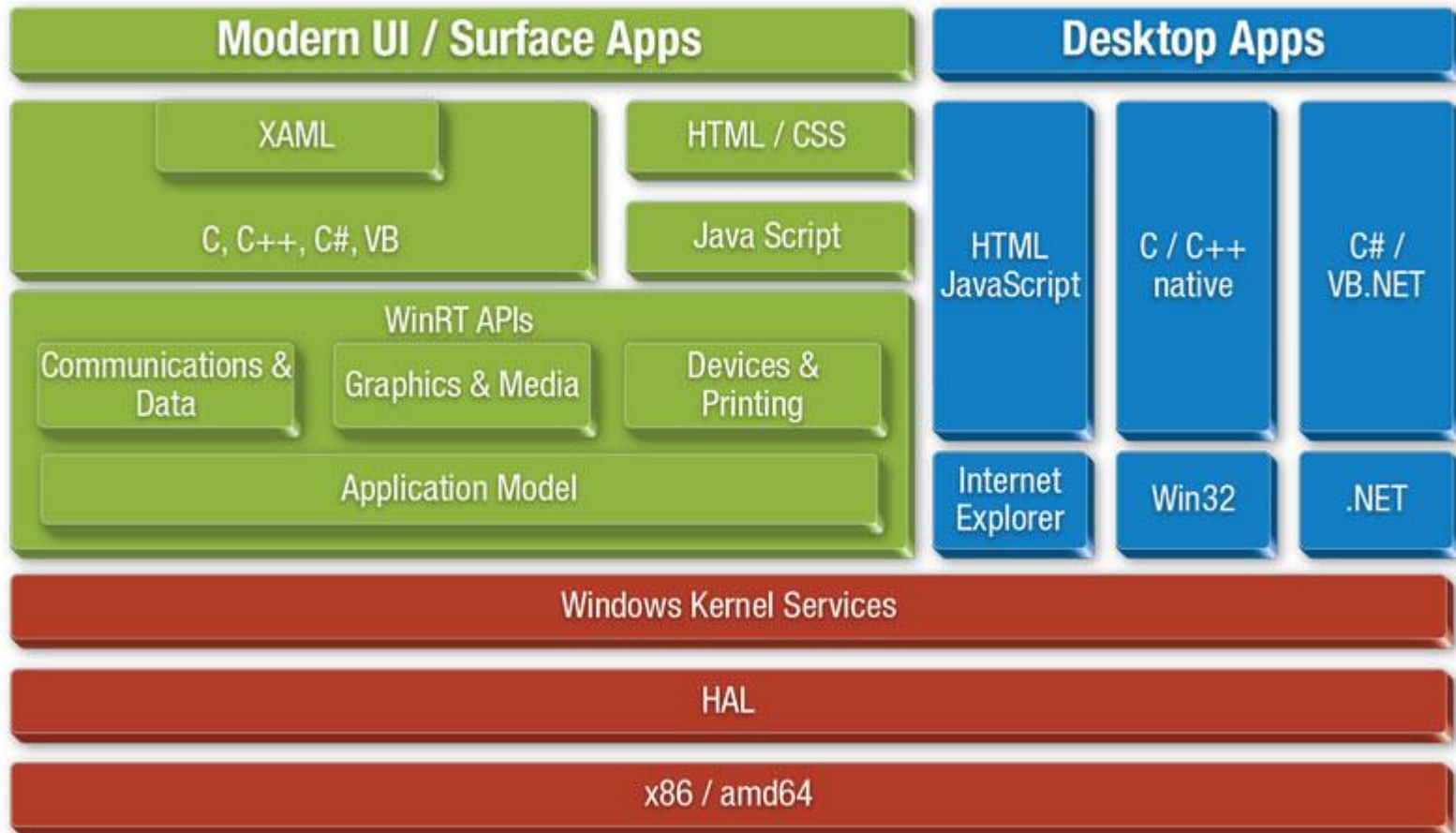
Windows 10 on PC

Same OS runs on different CPU



If CPU changes, you cannot run existing programs

Architecture of an application



Windows 8 architecture

Python

- General-purpose, high-level programming language
- Object-oriented programming language
- Supports event
- Easy-to-learn with clear and expressive syntax



Kivy: A Cross-Platform Python UI Library

- Open source Python library for rapid development of applications that make use of user interfaces, such as multi-touch apps.
 - Cross platform
 - » Linux, Windows, OS X, Android, iOS, and Raspberry Pi
 - » You can run the same code on all supported platforms
 - Business Friendly
 - » Kivy is 100% free to use
 - » You can use it in a commercial product
 - GPU Accelerated
 - » Graphics engine is built over OpenGL ES 2, using a modern and fast graphics pipeline

Reference

- [1] Application, available from <https://en.wikipedia.org/wiki/Application> accessed 24th Feb. 2019
- [2] Customer, available from <https://en.wikipedia.org/wiki/Customer> accessed 24th Feb. 2019
- [3] Developer, available from <https://en.wikipedia.org/wiki/Developer> accessed 24th Feb. 2019