江西理工大学 信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING

Mobile application development

移动应用开发

Lecture 09:

APP Inventor _Example

Review the example and introduce some blocks

Dr Ata Jahangir Moshayedi

Prof Associate ,

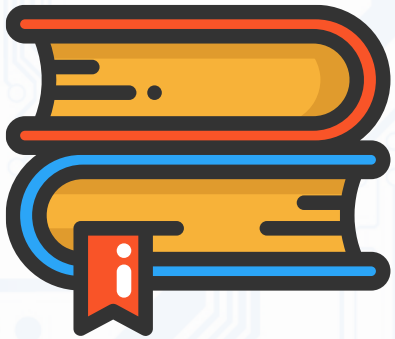
School of information engineering Jiangxi university of science and technology, China

EMAIL: ajm@jxust.edu.cn

Autumn _2021

江西理工大學 信息工程学院

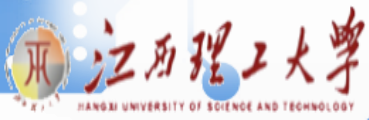
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



MOBILE APPLICATION DEVELOPMENT

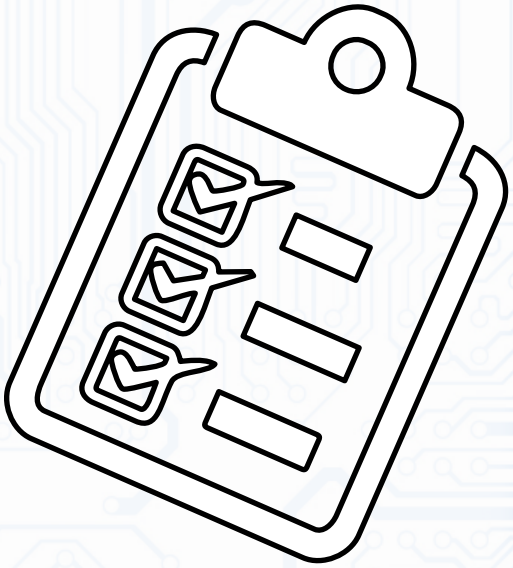
LECTURE 09: APP Inventor _Example

Review the example and introduce some blocks



App Inventor

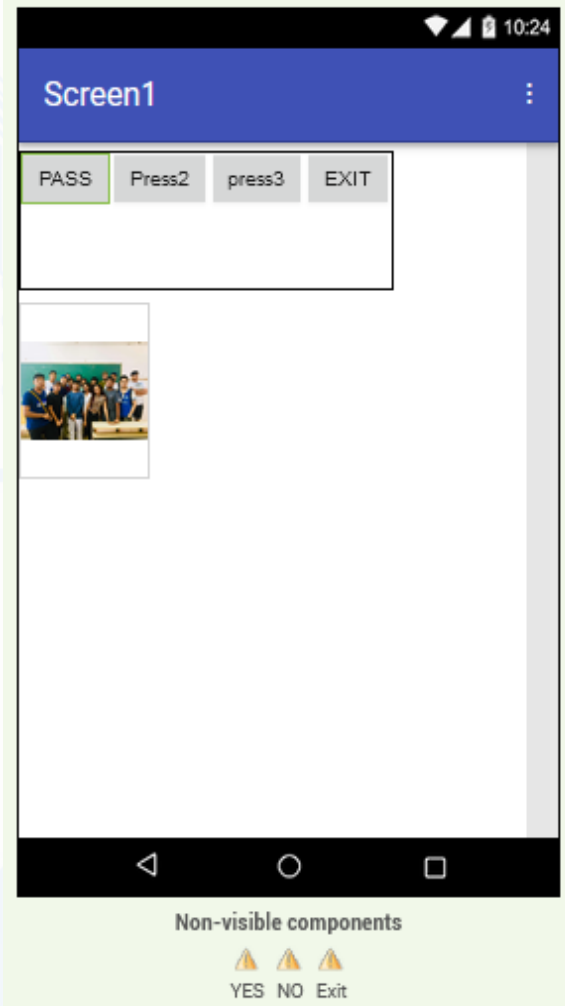
Agenda



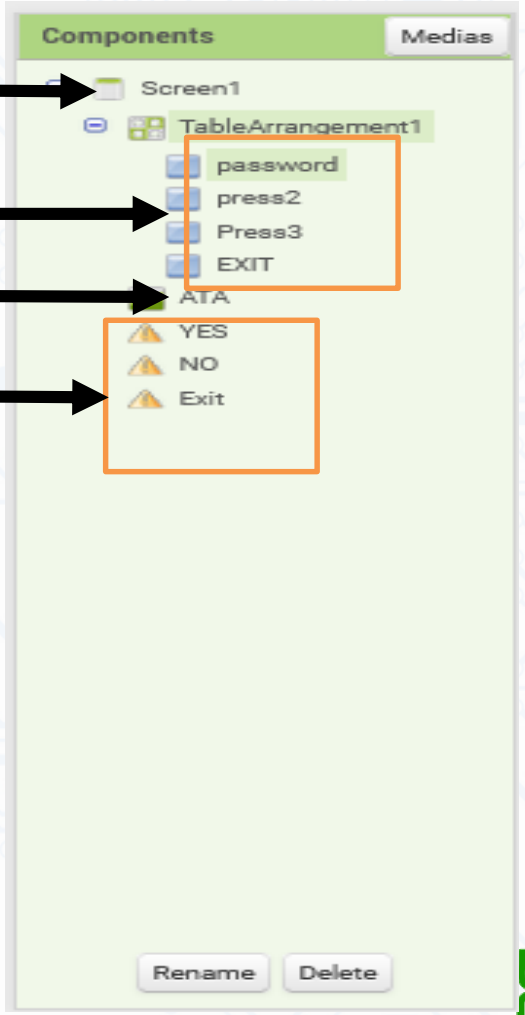
- **Review the example and introduce some blocks**
  - **MIT App Inventor Text Blocks**
  - **MIT App Inventor Notifier**
  - **MIT App Inventor Variable Blocks**
  - **MIT App Inventor Logic Blocks**

- We will find each needed block in this example and describe blocks and part to design an app
- Let us see our example

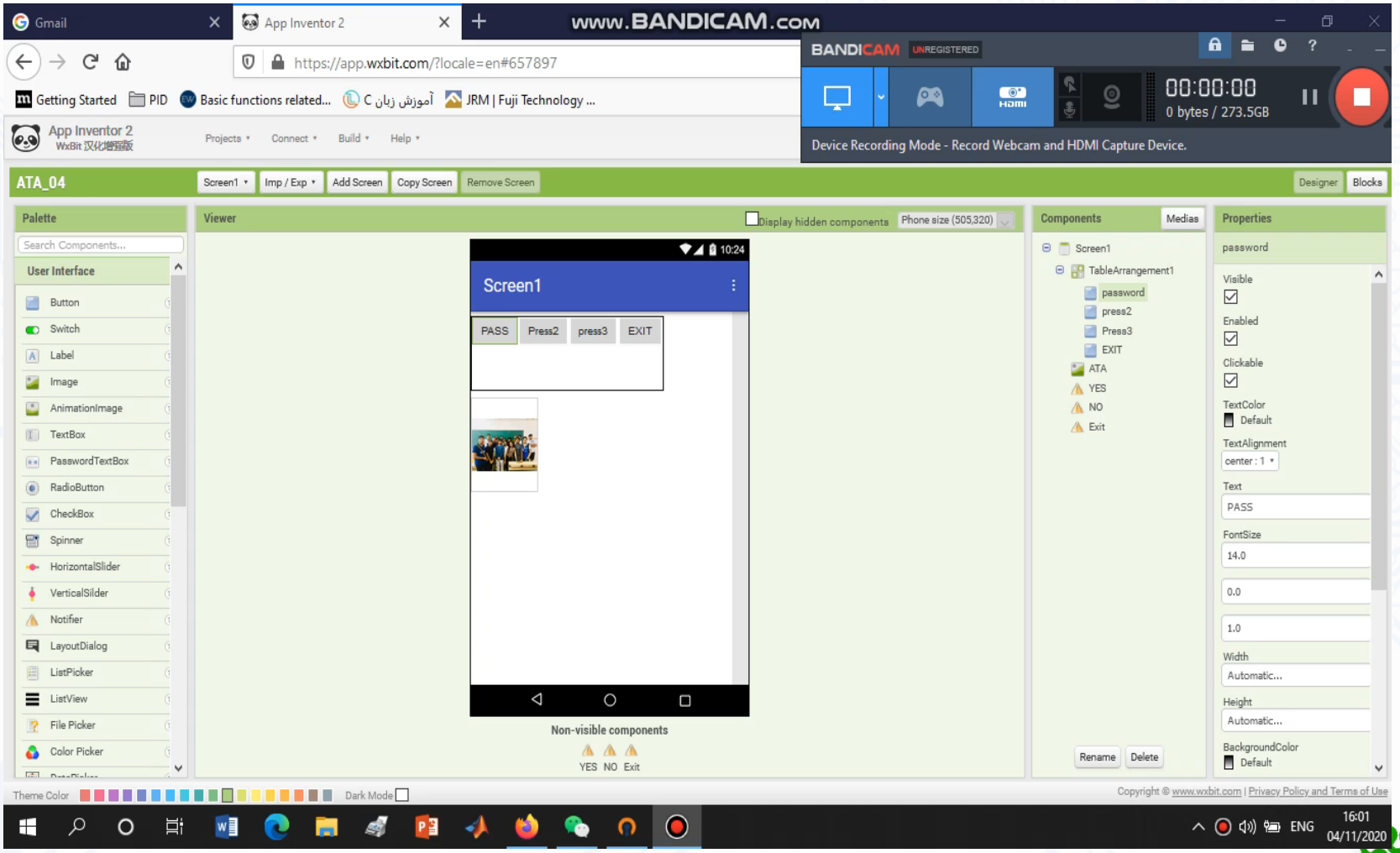
Project part in Designer section



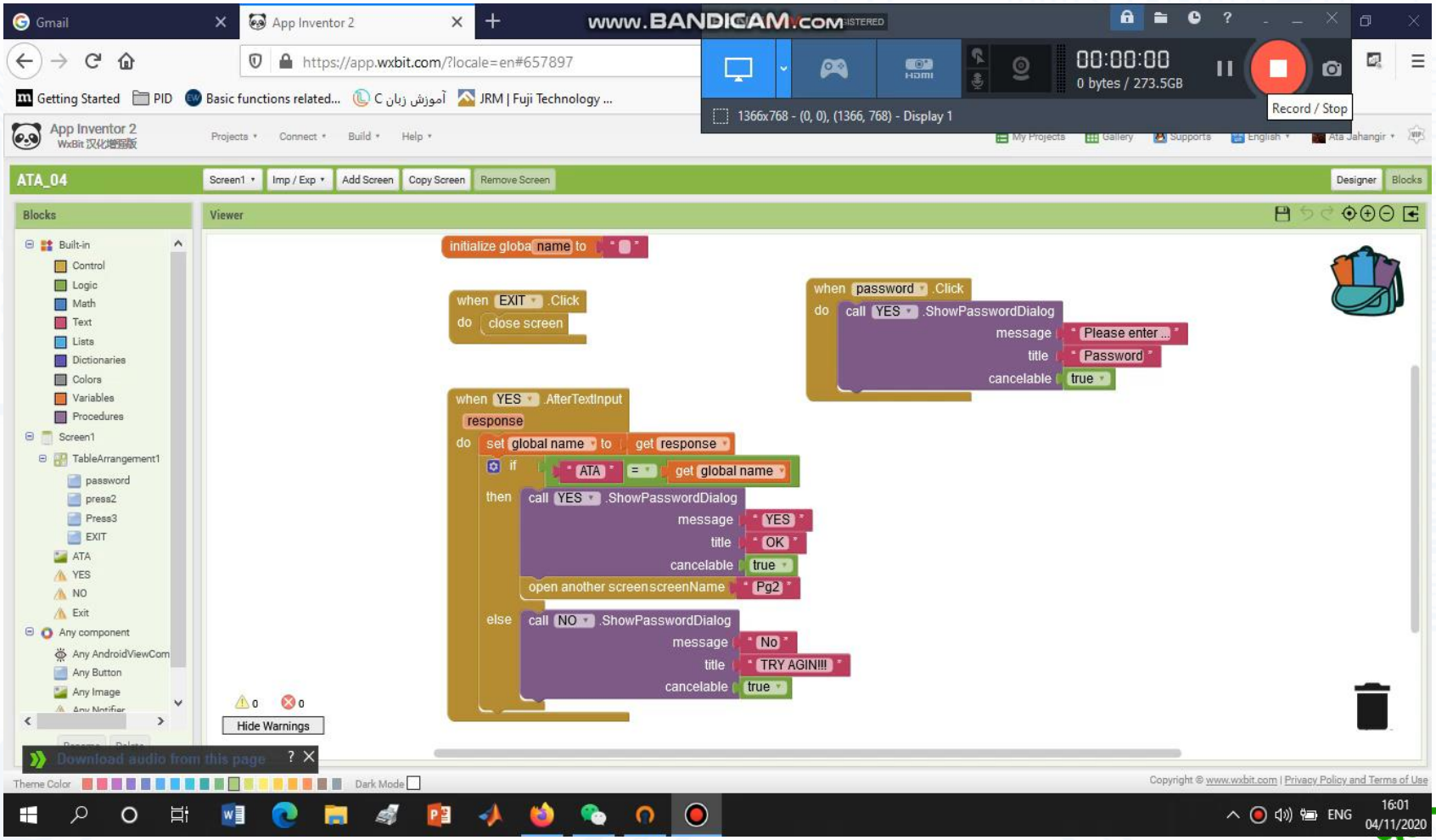
- @Screen 1
- Layout arrangement
- Buttom *4
- Image
- Notification section



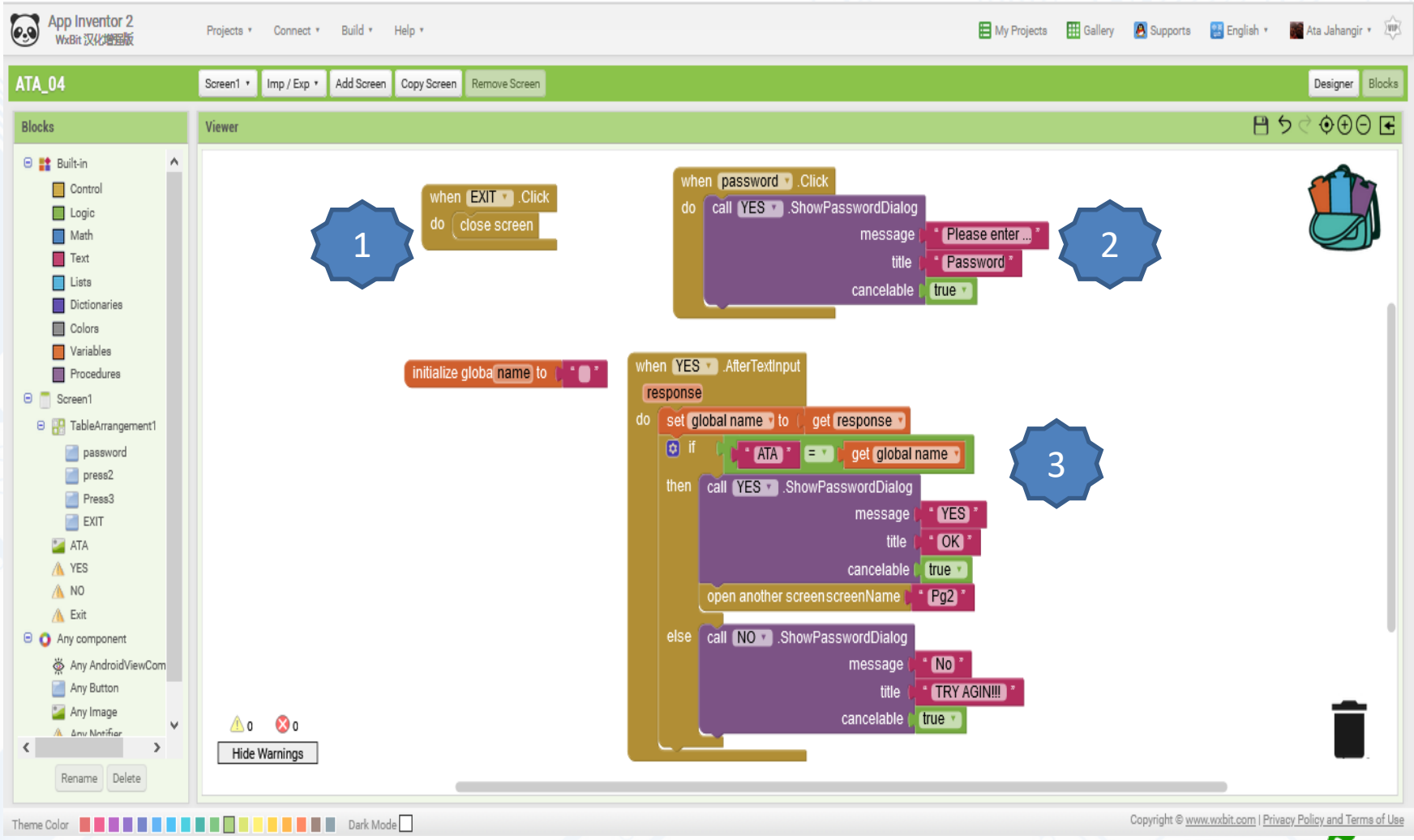
The example …..


Gmail
App Inventor 2
www.BANDICAM.com
https://app.wxbit.com/?locale=en#657897
BANDICAM UNREGISTERED
00:00:00
0 bytes / 273.5GB
Device Recording Mode - Record Webcam and HDMI Capture Device.
App Inventor 2
Projects
Connect
Build
Help
ATA_04
Screen1
Imp / Exp
Add Screen
Copy Screen
Remove Screen
Designer
Blocks
Palette
Viewer
Components
Properties
Screen1
PASS
Press2
press3
EXIT
Non-visible components
YES NO Exit
Rename
Delete
Theme Color
Dark Mode
Copyright © www.wxbit.com | Privacy Policy and Terms of Use
ENG
16:01
04/11/2020

App Designs Steps……



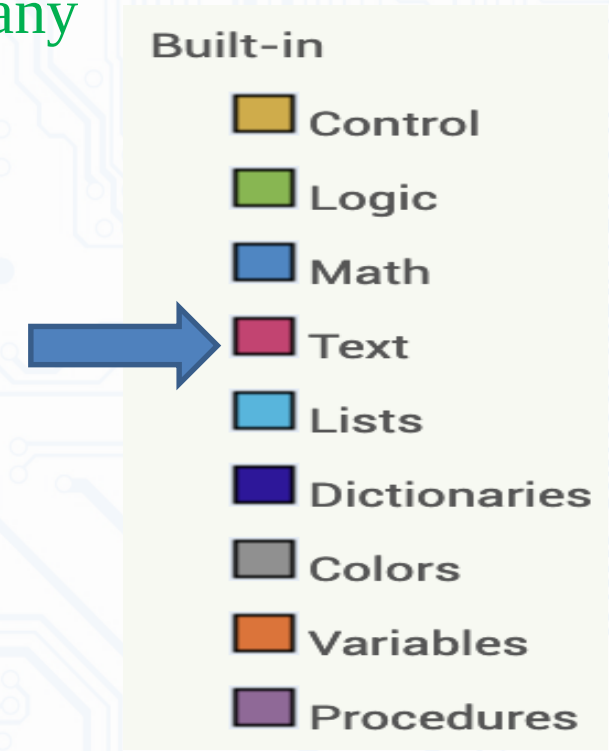
Project part in Block section ……



MIT App Inventor Text Blocks

- string
- join
- length
- is empty
- compare texts
- trim
- upcase
- downcase
- starts at
- contains
- contains any
- contains all

- split at first
- split at first of any
- split
- split at any
- split at spaces
- segment
- replace all
- obfuscated text
- is a string?
- reverse
- replace all mappings



MIT App Inventor Text Blocks

” “ (string block)

Contains a text string.

This string can contain any characters (letters, numbers, or other special characters).

On App Inventor it will be considered a Text object.

join

- Appends all of the inputs to make a single string.
- If no inputs, returns an empty string

length

- Returns the number of characters including spaces in the string.
- This is the length of the given text string.

is empty

- Returns whether or not the string contains any charaacters (including spaces).
- When the string length is 0, returns true otherwise it returns false.

MIT App Inventor Text Blocks

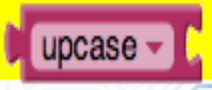
compare texts < > = ≠

- Returns whether or not the first string is lexicographically <, >, =, or ≠ the second string depending on which dropdown is selected.
- A string a considered lexicographically greater than another if it is alphabetically greater than the other string.
- Essentially, it would come after it in the dictionary. All uppercase letters are considered smaller or to occur before lowercase letters. cat would be > Cat.

trim

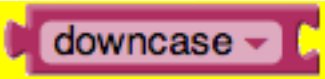
- Removes any spaces leading or trailing the input string and returns the result.

upcase



- Returns a copy of its text string argument converted to all upper case.

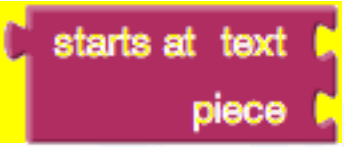
downcase



- Returns a copy of its text string argument converted to all lower case.

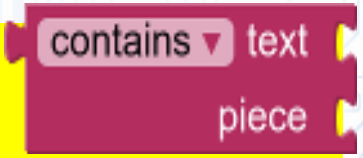
MIT App Inventor Text Blocks

starts at



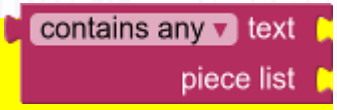
Returns the character position where the first character of *piece* first appears in text, or 0 if not present. For example, the location of *ana* in *havana banana* is 4.

contains



Returns true if piece appears in text; otherwise, returns false.

contains any



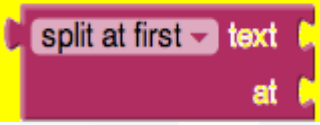
- Returns true if any of the pieces in the piece list appear in text; otherwise, returns false.
- This block can be obtained by changing the dropdown on the contains block

contains all

- Returns true if all of the pieces in the piece list appear in text; otherwise, returns false.
- This block can be obtained by changing the dropdown on the contains block.

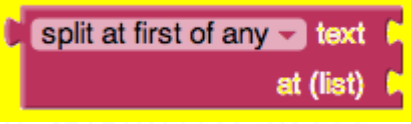
MIT App Inventor Text Blocks

split at first



Divides the given text into two pieces using the location of the first occurrence of at as the dividing point, and returns a two-item list consisting of the piece before the dividing point and the piece after the dividing point. Splitting apple,banana,cherry,dogfood with a comma as the splitting point returns a list of two items: the first is the text apple and the second is the text banana,cherry,dogfood. Notice that the comma after apple doesn't appear in the result, because that is the dividing point.

split at first of any

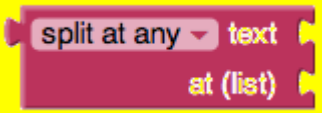


- Divides the given text into a two-item list, using the first location of any item in the list at as the dividing point.
- Splitting i love apples bananas apples grapes by the list [ba,ap] would result in a list of two items the first being i love and the second ples bananas apples grapes.

split

- Divides text into pieces using at as the dividing points and produces a list of the results.
- Splitting one,two,three,four at , (comma) returns the list (one two three four). Splitting one-potato,two-potato,three-potato,four at -potato, returns the list (one two three four).

split at any



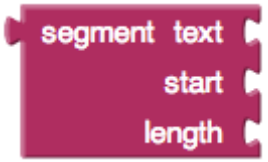
- Divides the given text into a list, using any of the items in at as the dividing point, and returns a list of the results.
- Splitting appleberry,banana,cherry,dogfood with at as the two-element list whose first item is a comma and whose second item is rry returns a list of four items: (applebe banana che dogfood).

MIT App Inventor Text Blocks

split at spaces

Divides the given text at any occurrence of a space, producing a list of the pieces.

segment



Extracts part of the text starting at start position and continuing for length characters.

replace all

Returns a new text string obtained by replacing all occurrences of the substring with the replacement.

Replace all with She loves eating. She loves writing. She loves coding as the text, She as the segment, and Hannah as the replacement would result in Hannah loves eating. Hannah loves writing. Hannah loves coding.

obfuscated text

Produces text, like a text block. The difference is that th etext is not easily discoverable by examining the app's contents.

Use this when creating apps to distribute that include confidential information, for example, API keys.

Warning: This provides only very low security against expert adversaries.

MIT App Inventor Text Blocks

is a string?

Returns true if *thing* is a text object, otherwise false.

reverse

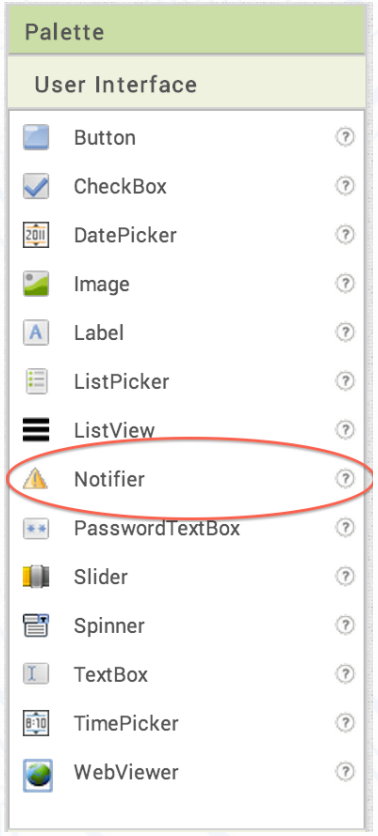
- Reverse the given text. For example, “reverse” would become “esrever”.

replace all mappings

Given a dictionary of mappings as input, replaces the key entries in the text with the corresponding values in the dictionary. Returns the text with the mappings applied.

MIT App Inventor Notifier

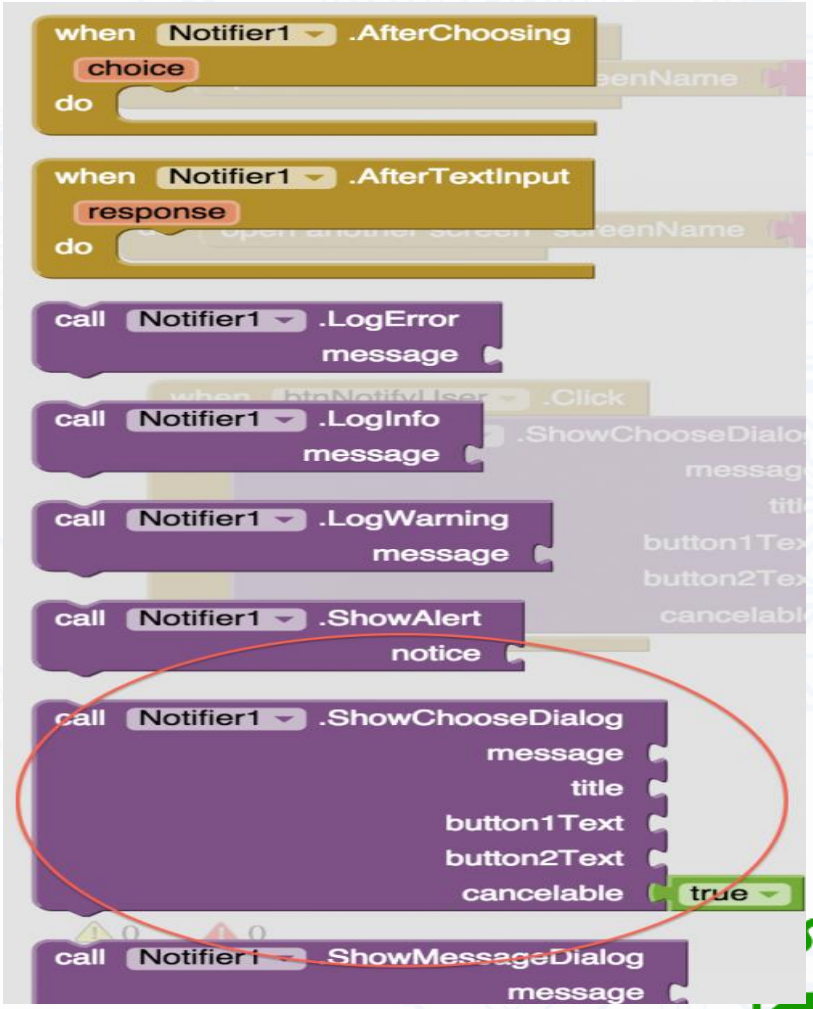
To create a pop up warning dialog, use the *Notifier* control that appears in the Designer's Palette:



Drag the Notifier icon into the app user interface design – the Notifier is used as a non-visible component, just below the user interface:

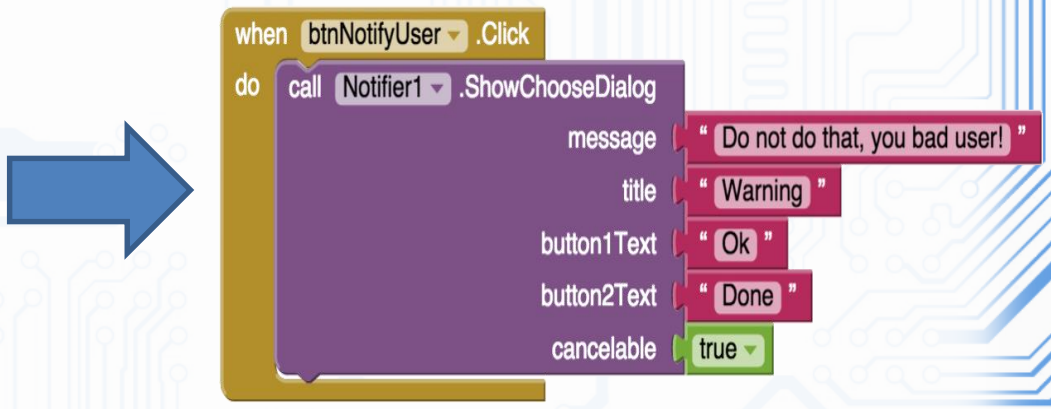
MIT App Inventor Notifier

- Switch to the Blocks view, click on the Notifier1 block to view the available methods.
- partial list of methods is shown here – the item circled in red is the one we will use in this example:



MIT App Inventor Notifier

In response to an on screen button press (*btnNotifyUser's Click* event), the app displays the dialog message:

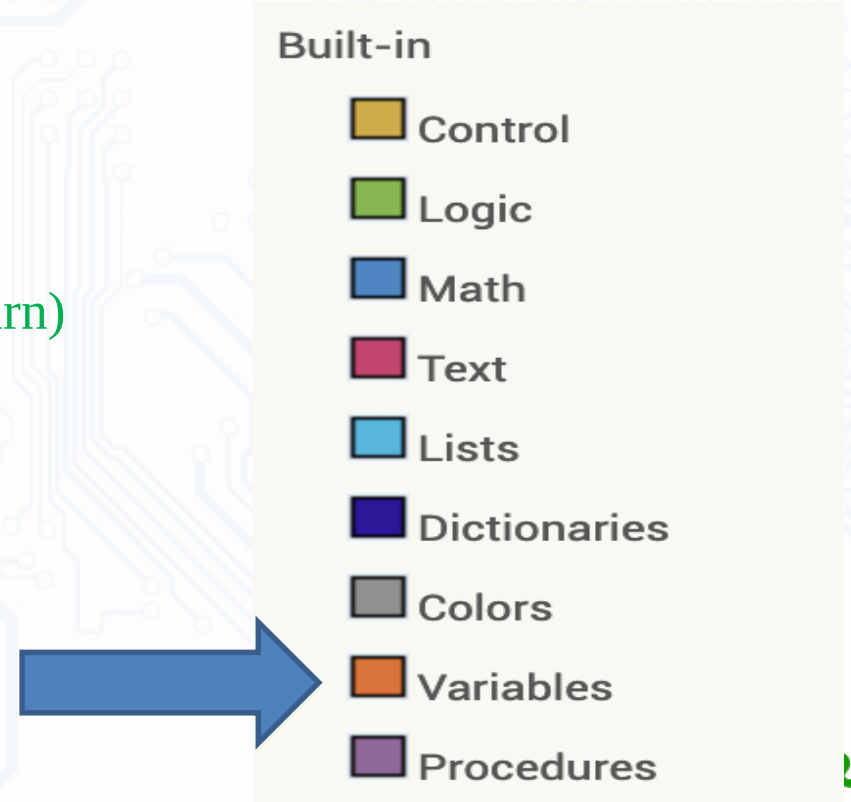


- This block can be inserted anywhere a processing block is allowed such as the result of an if-then-else conditional block.
- You can also change the text or background color of the app to something more interesting than the "black and white" default colors shown in this example.
- ***Notifier* provides a quick and easy way to display short alert messages to the user of the app.**

How to make the notifer

MIT App Inventor Variable Blocks

- There are five main types of variable blocks:
  - initialize global name to
  - get
  - set
  - initialize local name to in (do)
  - initialize local name to in (return)



MIT App Inventor Variable Blocks

initialize global name to

- This block is used to create global variables. It takes in any type of value as an argument.
- Clicking on *name* will change the name of this global variable.
- Global variables are used in all procedures or events so this block will stand alone.

- Global variables can be changed while an app is running and can be referred to and changed from any part of the app even within procedures and event handlers.
- can rename this block at any time and any associated blocks referring to the old name will be updated automatically.

MIT App Inventor Variable Blocks

get

This block provides a way to get any variables you may have created.

set

- This block follows the same rules as get.
- Only variables in scope will be available in the dropdown.
- Once a variable *v* is selected, you can attach a block to give *v* a new value.

initialize local name to - in (do)

This block is a mutator that allows you to create new variables that are only used in the procedure you run in the DO part of the block. This way all variables in this procedure will all start with the same value each time the procedure is run.
NOTE: This block differs from the block described below because it is a DO block.
You can attach *statements* to it. Statements *do* things.
That is why this block has space inside for statement blocks to be attached.
You can rename the variables in this block at any time and any corresponding blocks elsewhere in your program that refer to the old name will be updated automatically

MIT App Inventor Variable Blocks

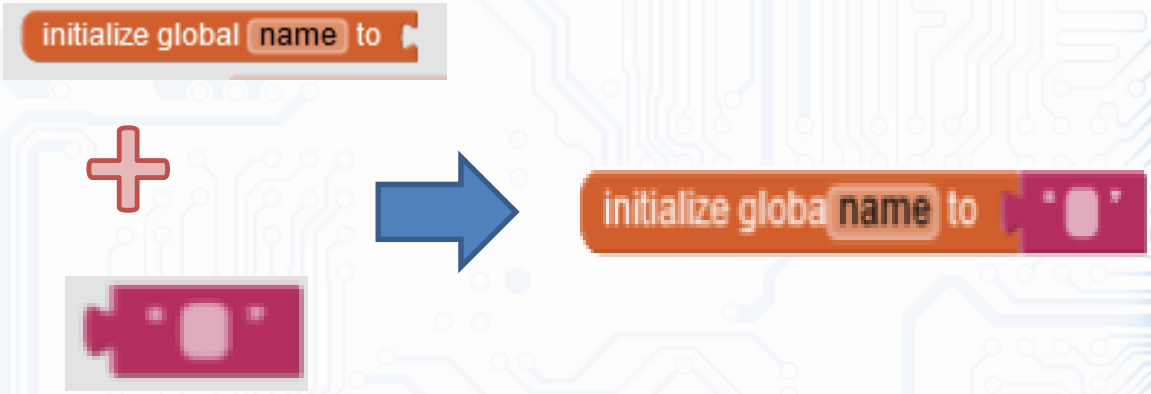
initialize local name to - in (return)

This block is a mutator that allows you to create new variables that are only used in the procedure you run in the RETURN part of the block.
This way all variables in this procedure will all start with the same value each time the procedure is run.

NOTE: This block differs from the block described above because it is a RETURN block.

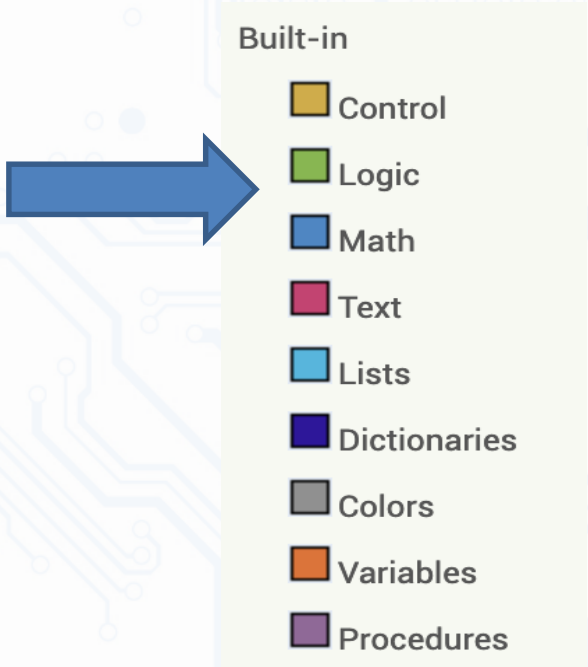
- **You can attach *expressions* to it. Expressions *return* a value. That is why this block has a socket for plugging in expressions.**
- You can rename the variables in this block at any time and any corresponding blocks elsewhere in your program that refer to the old name will be updated automatically

Variable section



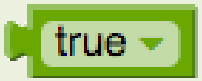
MIT App Inventor Logic Blocks

- There are Seven main types of variable blocks:
  - True
  - False
  - = {#=}
  - Not
  - ≠ {#not=}
  - And
  - or



MIT App Inventor Logic Blocks

true



- Represents the constant value true. Use it for setting boolean property values of components, or as the value of a variable that represents a condition.

false

- Represents the constant value false. Use it for setting boolean property values of components, or as the value of a variable that represents a condition.

= {#=}

- **Tests whether its arguments are equal.**

Two numbers are equal if they are numerically equal, for example, 1 is equal to 1.0.

Two text blocks are equal if they have the same characters in the same order, with the same case. For example, banana is not equal to Banana.

Numbers and text are equal if the number is numerically equal to a number that would be printed with that text. For example, 12.0 is equal to the result of joining the first character of 1A to the last character of Teafor2.

Two lists are equal if they have the same number of elements and the corresponding elements are equal.

Acts exactly the same as the = block found in Math

not

•Performs logical negation, returning false if the input is true, and true if the input is false.

MIT App Inventor Logic Blocks

≠ {#not=}

Tests to see whether two arguments are not equal.

or

Tests whether any of a set of logical conditions are true. The result is true if one or more of the tested conditions are true.

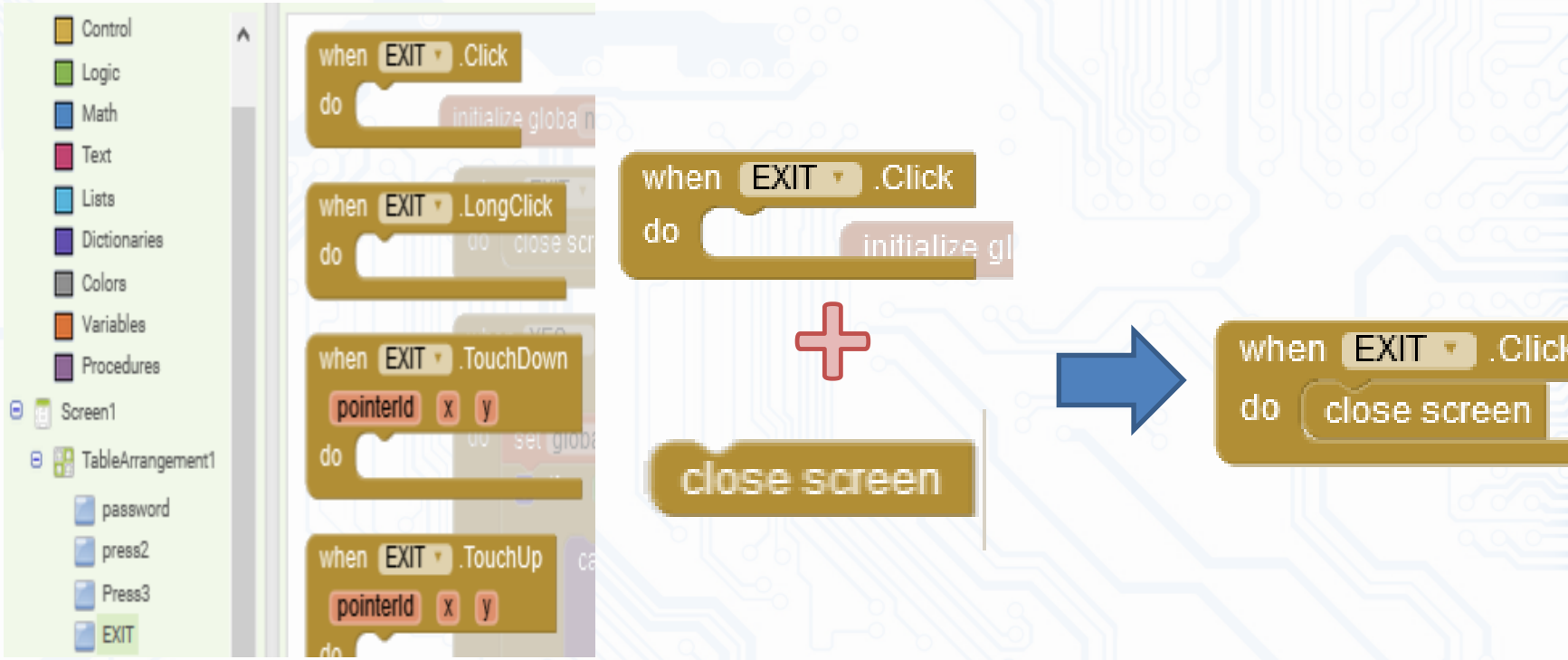
The number of tests can be expanded using the mutator.

The conditions are tested left to right, and the testing stops as soon as one of the conditions is true. If there are no conditions to test, then the result is false.

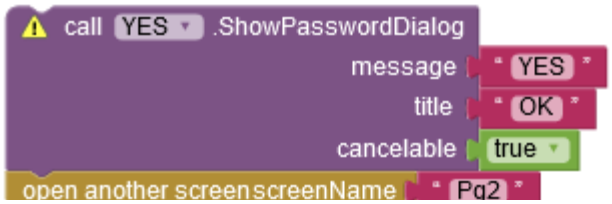
and

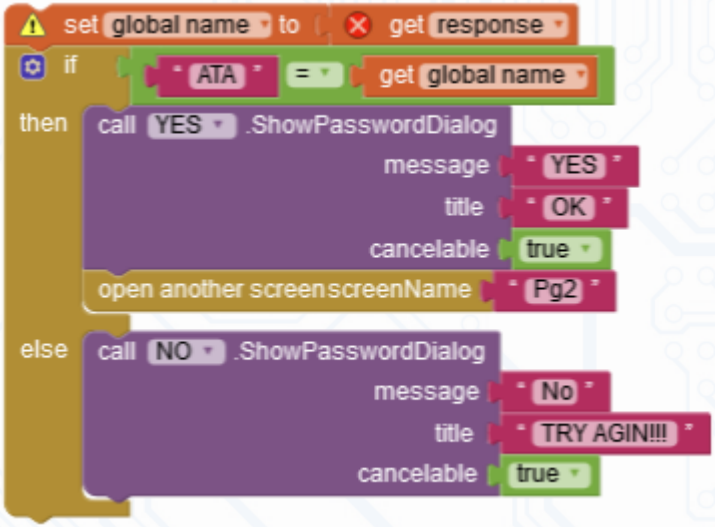
- Tests whether all of a set of logical conditions are true. The result is true if and only if all the tested conditions are true. The number of tests can be expanded using the mutator.
- The conditions are tested left to right, and the testing stops as soon as one of the conditions is false. If there are no conditions to test, then the result is true. You can consider this to be a logician's joke.

Bottom block actions



All needed part for our example

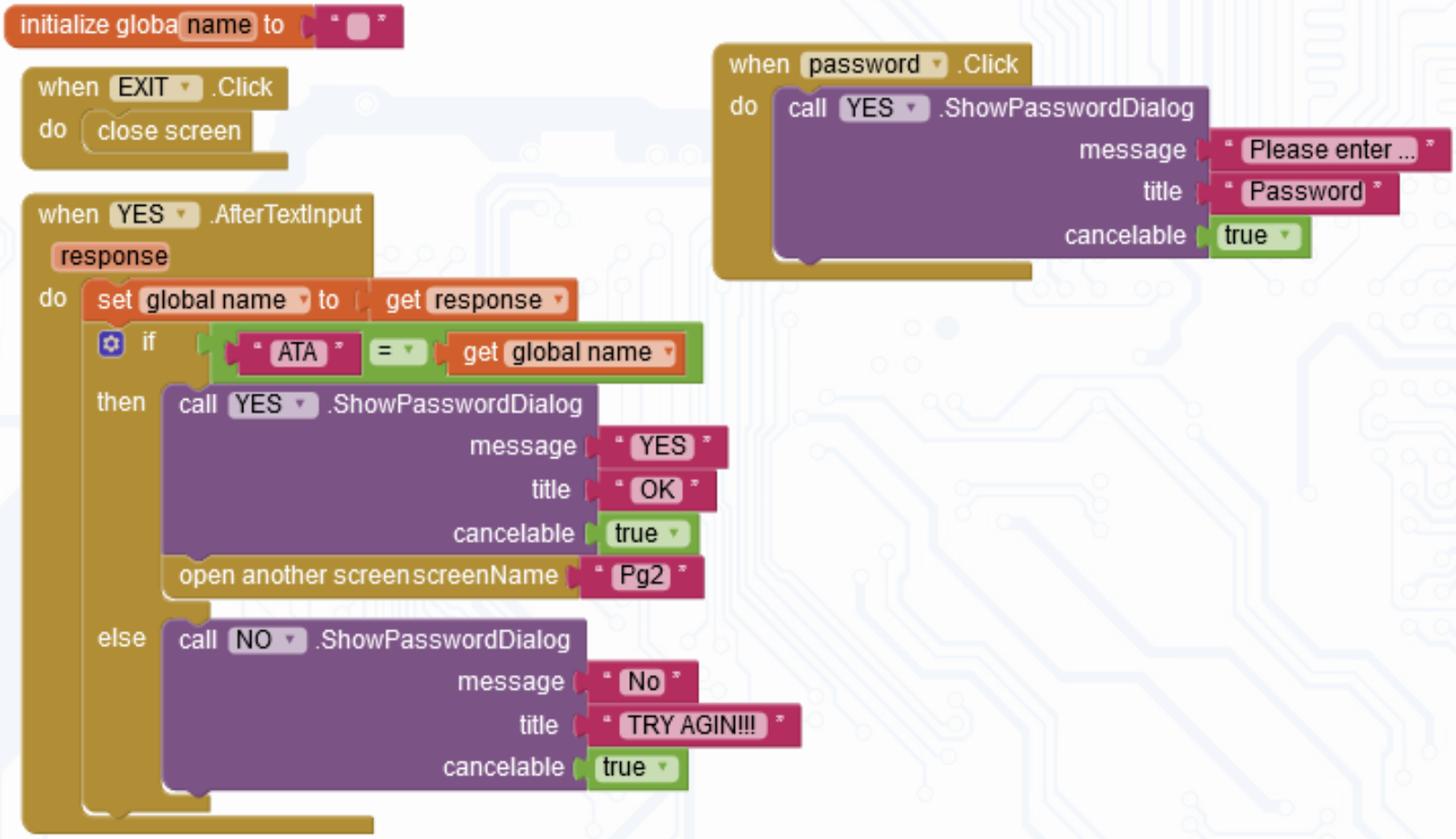




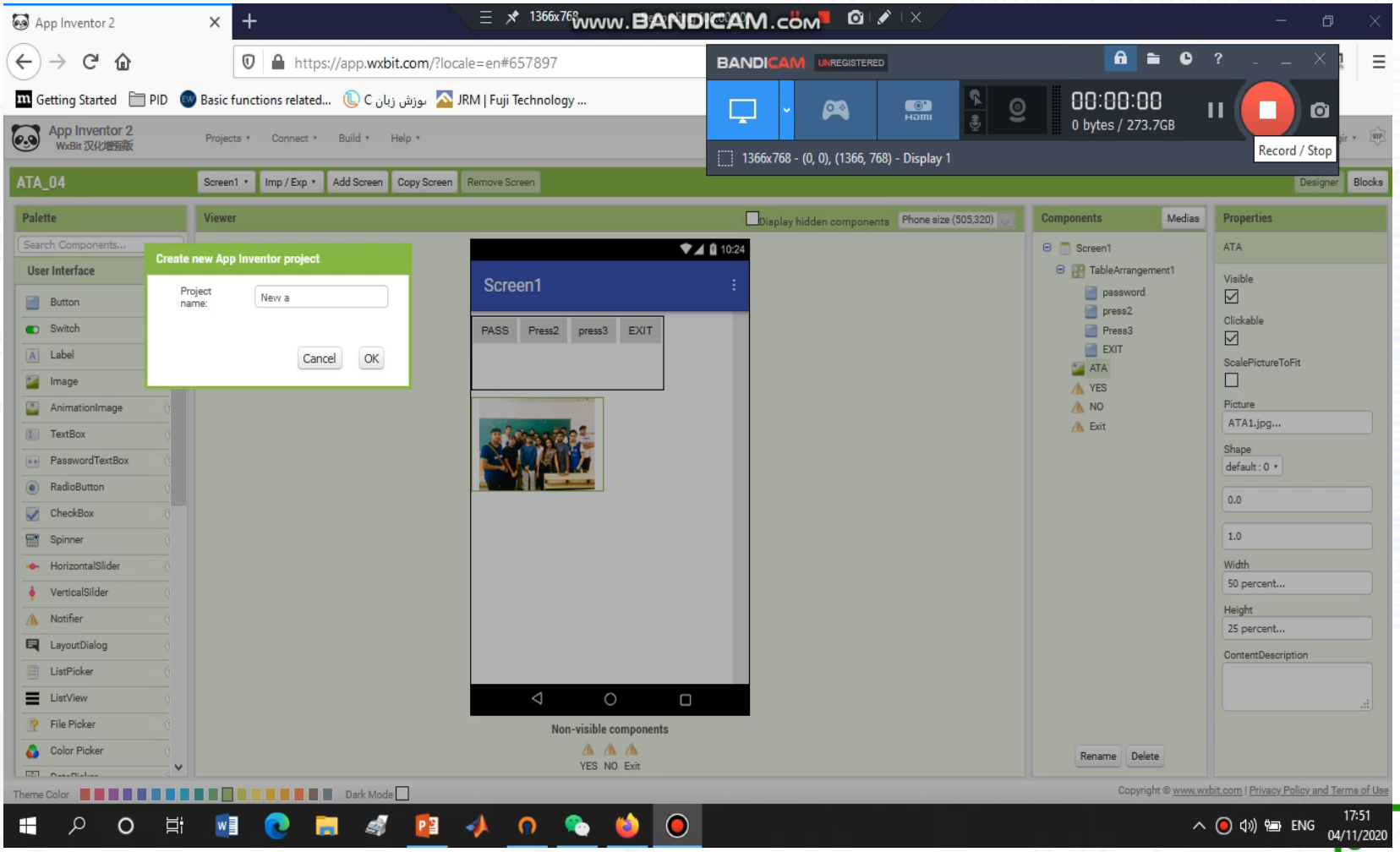
江西理工大學 信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING

App Inventor

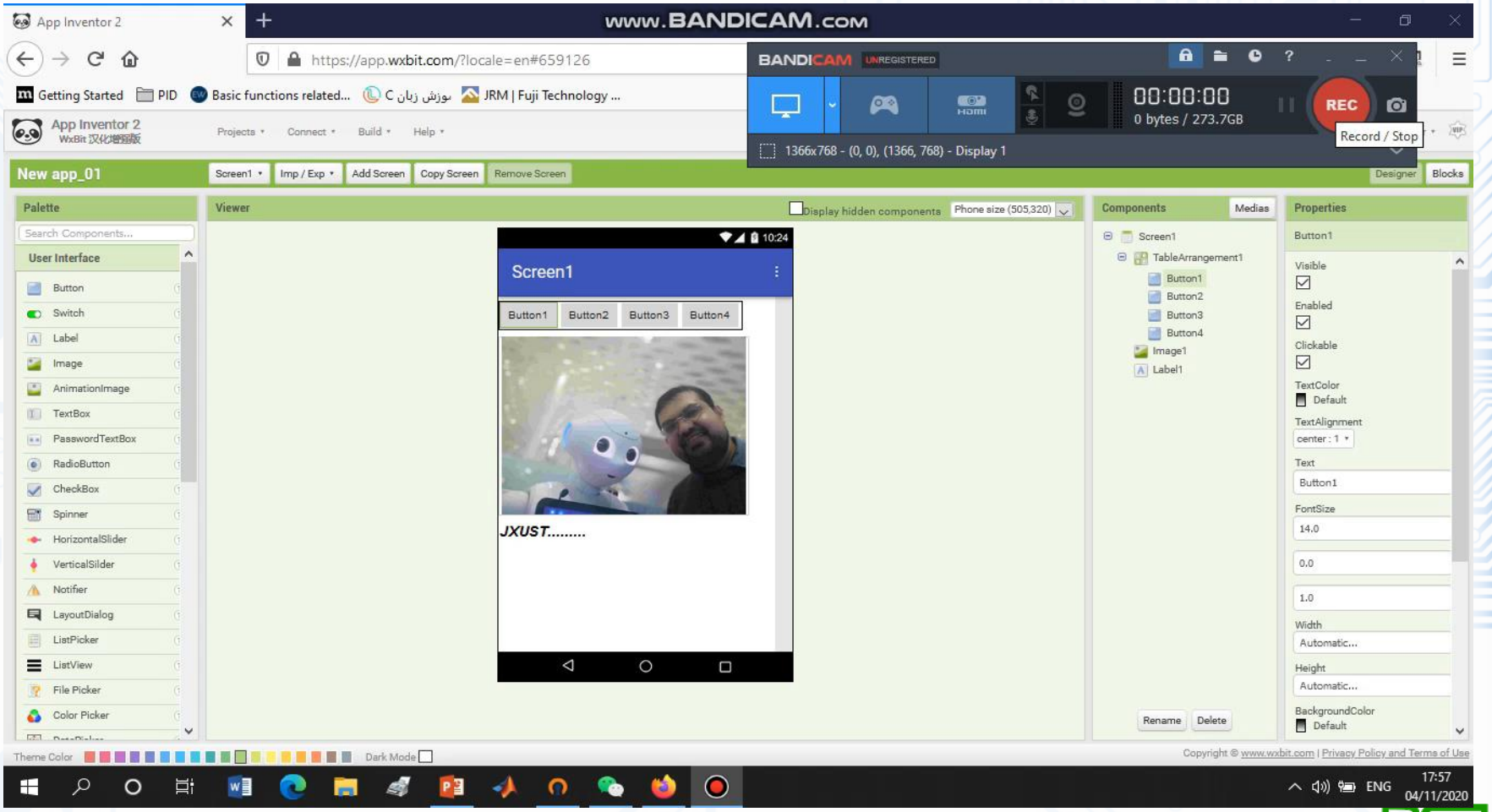
All needed part for our example



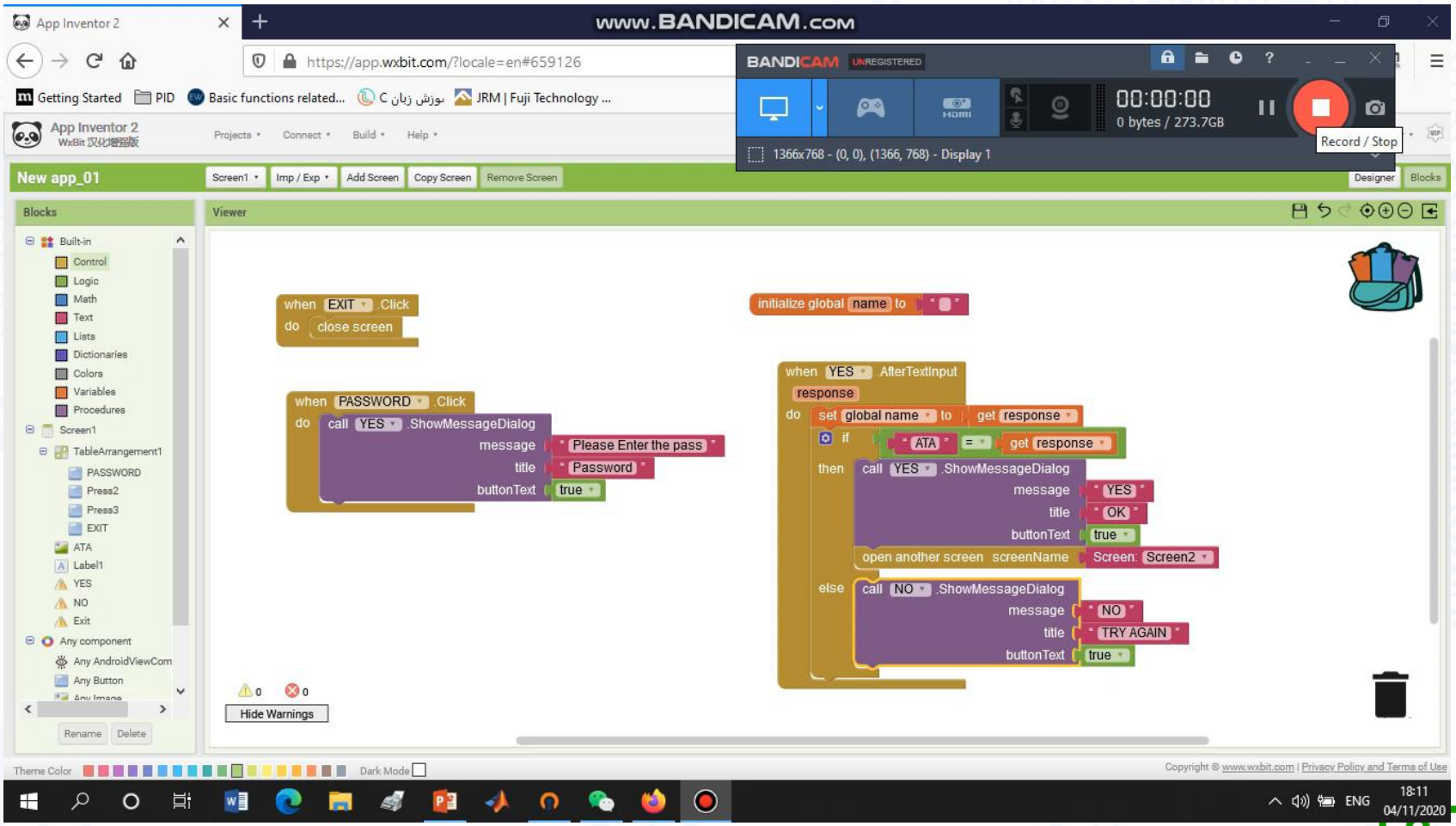
App Designs Steps



App Designs Steps

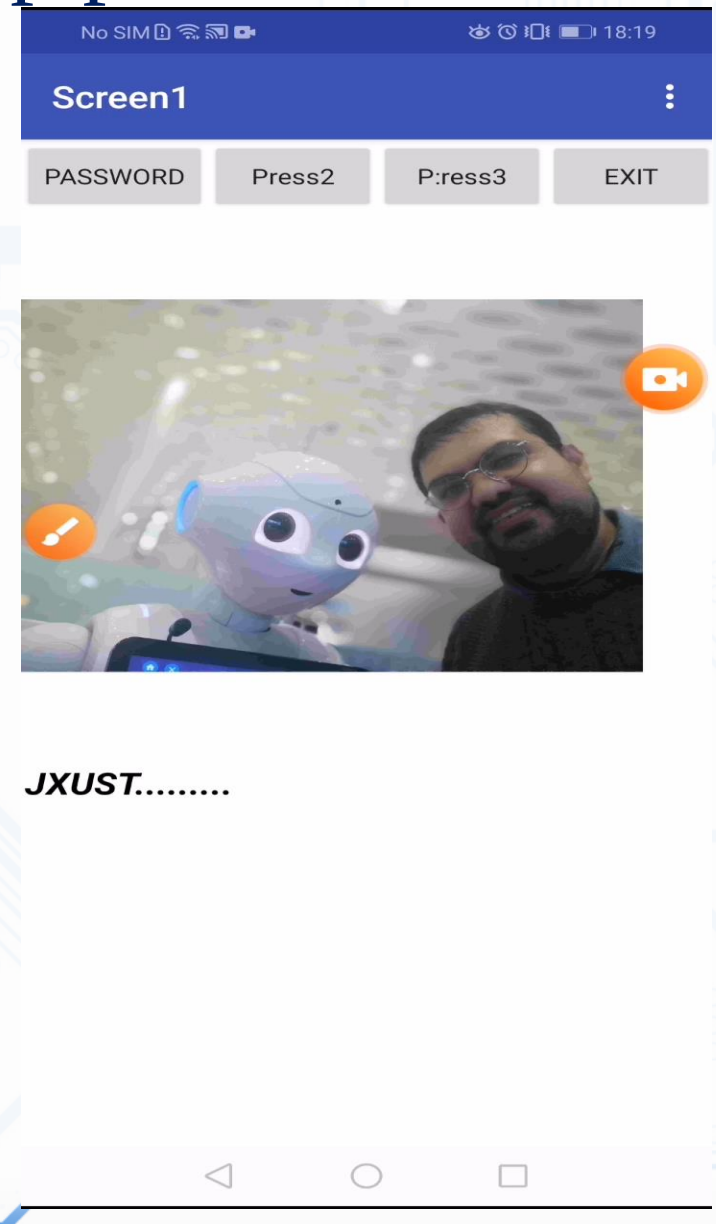


App Designs Steps

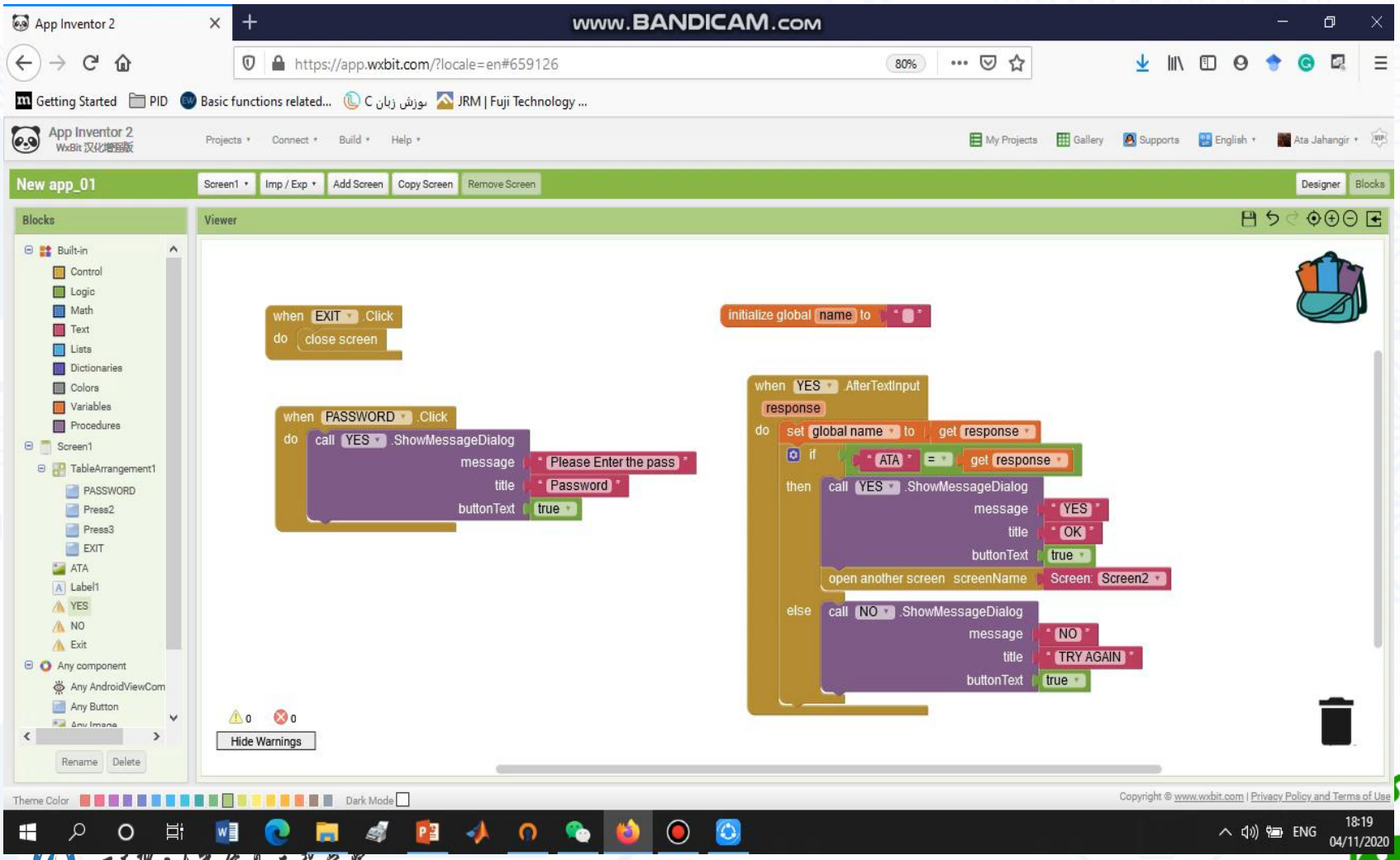


Designed app performance

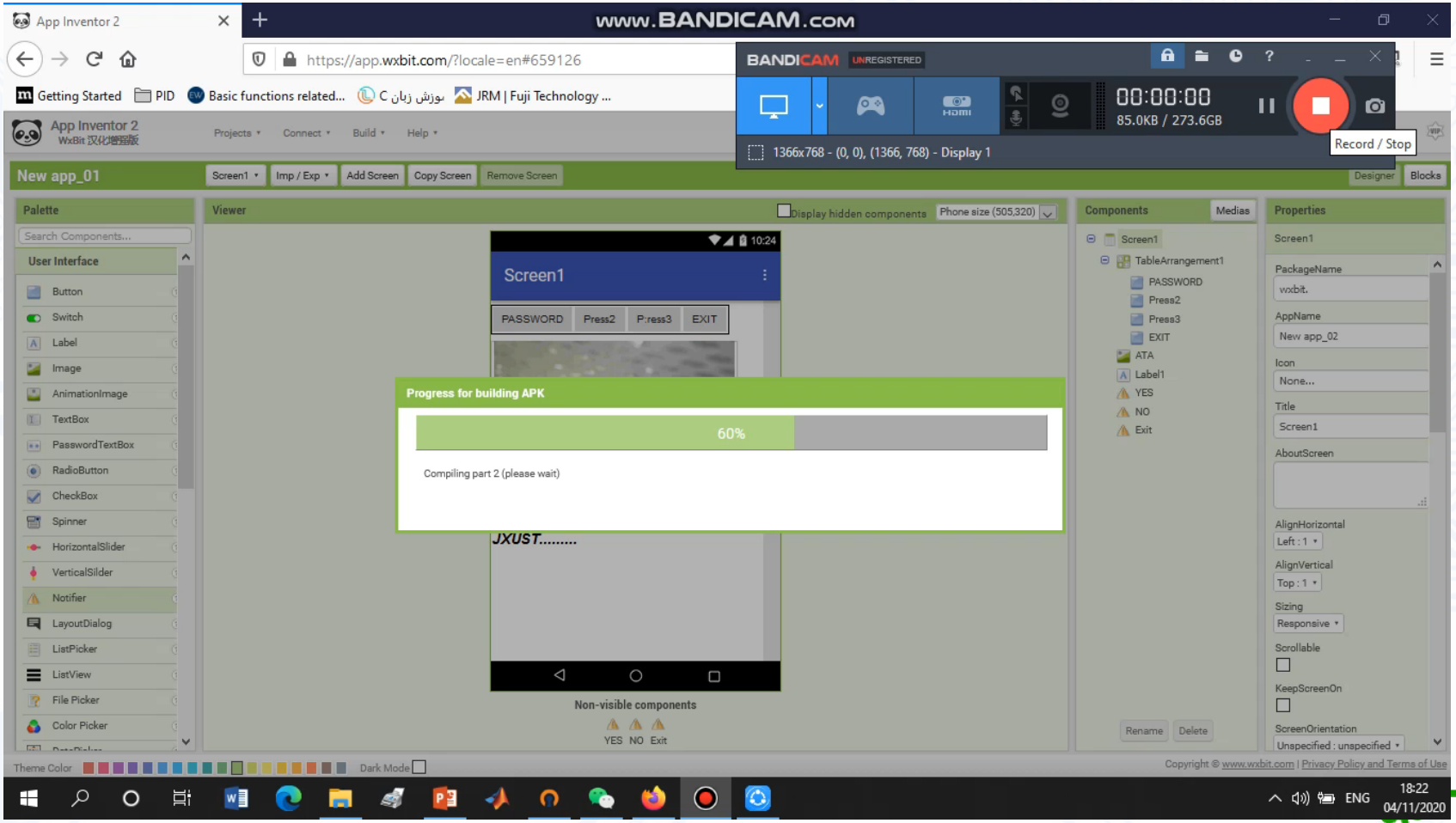
There is one problem



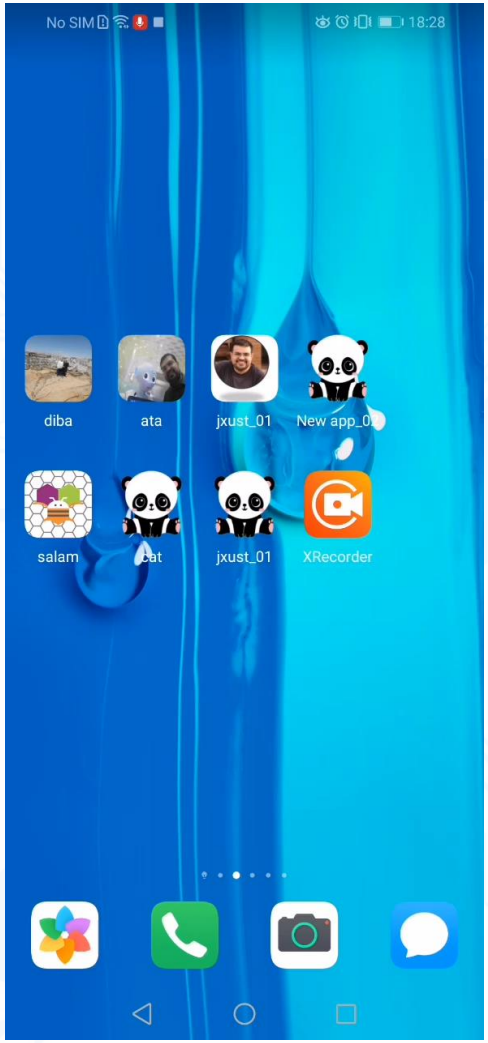
App Designs Steps



App Designs Steps



Final App



How we can make this design better ?

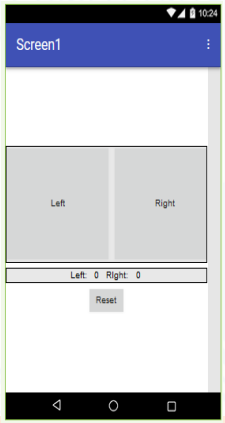
you should write the pass in caps how we can make the case independent of caps or small?

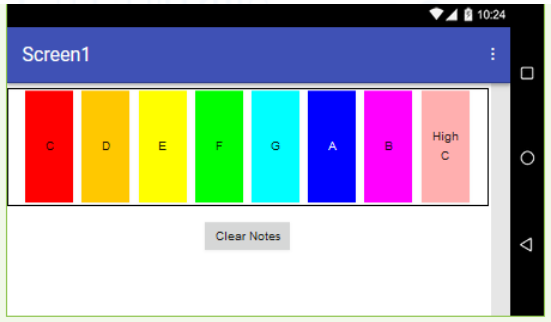
Student Task_7

1.Repaet the example in the ppt and report and sort the problem

Whats your prefer name for this app?

2. Design and follow two example shown on task book 1, you should send the design process and working add clips inside your ppt





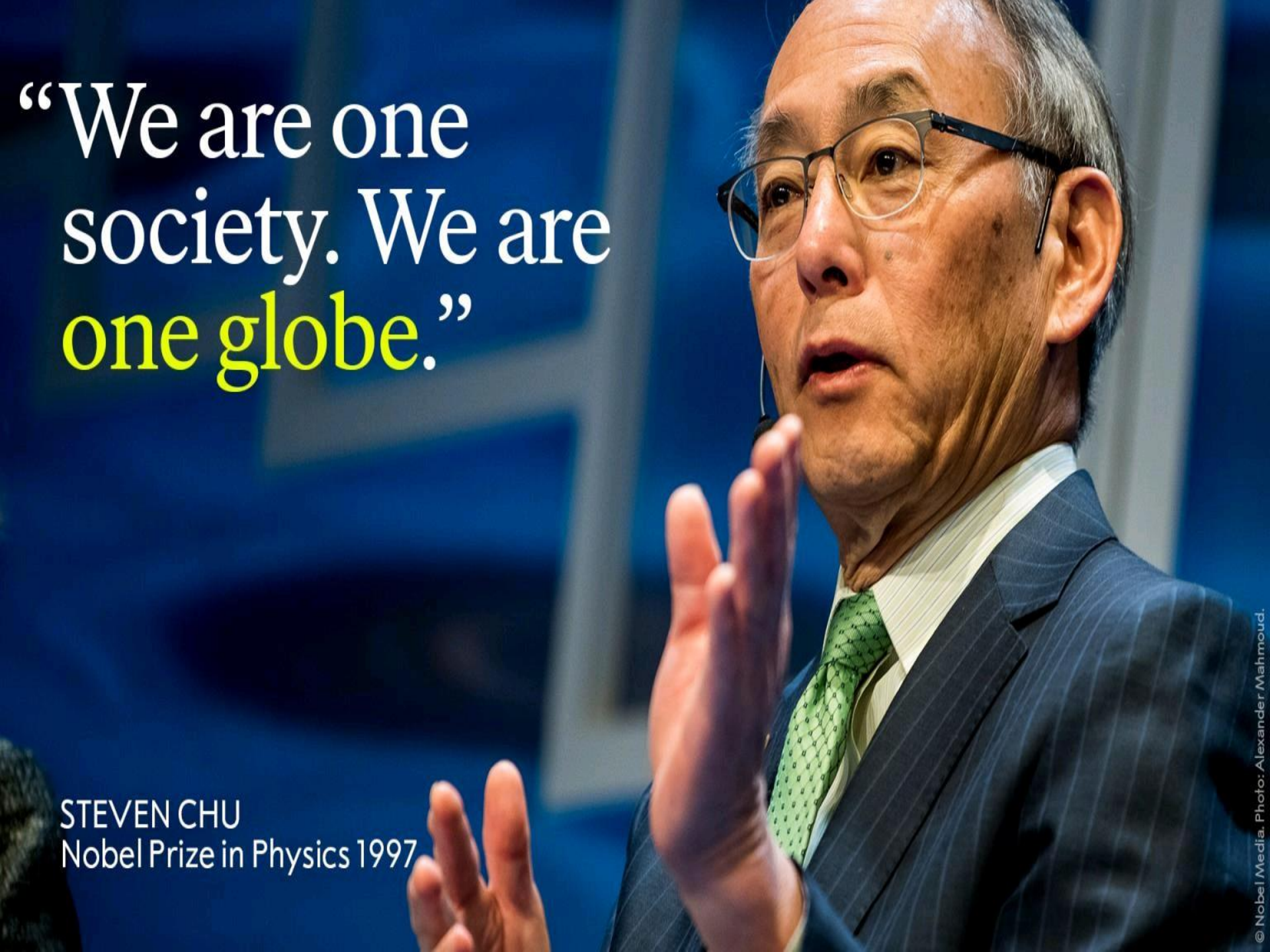
Next lecture

- **Just MOOC**
- **Your file should have this format of name**

  <Task number><student name><Student ID>.ppt

Reference

- **https://coldstreams.com/appinventor/2014/09/08/display-warning-and-alert-box-messages-in-app-inventor-apps/**
- **https://appinventor.mit.edu/explore/content/alertme.html**
- **Teaching with AppInventor** http://appinventor.mit.edu/explore/teach.html
  AppInventor Tutorials:
  http://appinventor.mit.edu/explore/ai2/tutorials.html
- **Sounds** http://www.soundbible.com
- **App Inventor:** http://appinventor.googlelabs.com/
- **Appinventor.org:** http://www.appinventor.org/
- **Wolber, Abelson et al. text:** http://www.appinventor.org/text2011
- **Group:** http://groups.google.com/group/app-inventor-instructors
- **Wolber course:** http://appinventor.org/course-in-a-box
- **Morelli course:** http://turing.cs.trincoll.edu/~ram/cpsc110/


"We are one society. We are one globe."
STEVEN CHU
Nobel Prize in Physics 1997
© Nobel Media. Photo: Alexander Mahmoud.

江西理工大学
Jiangxi University of Science and Technology
信息工程学院
School of information engineering

Digital Image Processing

THANK YOU

"BE HUMBLE. BE HUNGRY. AND ALWAYS BE THE HARDEST WORKER IN THE ROOM."