

The International Conference on Advanced Management
and Information Technology Services 2017

Multi-platform Application User Interface Design based on Spreadsheet

Authors:

Noprianto

Benfano Soewito, Ph.D

Dr. Ford Lumban Gaol

Prof. Bahtiar Saleh Abbas, Ph.D

Presented by:

Benfano Soewito, Ph.D

User Interface

- What users directly see and feel
 - Users interact with an application through its user interface
 - Types: command-line interface (CLI), text-based user interface (TUI), graphical user interface (GUI), etc.

User Interface Development

- What programmers spend a lot of time on
 - Decades ago: programmers drew the components character-by-character or pixel-by-pixel
 - Then: a set of user interface components were available and ready to use
 - Then: graphical user interface builders → Designing user interface was like drawing, only probably much easier

Multiple Platforms

- As technologies advanced, new platforms were born
 - Programmers are busier when they target more platforms
 - Single code base, at the user interface part: sometimes hard or not applicable to produce an application that is run on many platforms

Proposed Multi-platform User Interface Design

- User interface model design
- A model that put emphasis on multi-platform output
- Design only once and codes for multiple platforms are simpler to be generated
- Assume no specific user interface toolkit
- Based-on spreadsheet

Spreadsheet

- Have been around us for almost four decades
- Interesting because of:
 - Its computational techniques
 - Table-oriented user interface
 - Provide a good foundation for user interface modeling

Method (1)

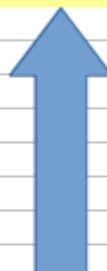
- First worksheet represents a user interface layout
- Second worksheet holds properties of each user interface component in the layout
- Third worksheet should contain related or helper code if any

Method (2)

- User Interface Layout:
 - Map spreadsheet table into grid-like layout
 - User interface component: using simple, descriptive, text-based definition
 - Component styling: make use of cell styling (background color, font color, etc)
- Component properties:
 - In second sheet
 - Reference to component: column A
 - Property: column B
 - Value: starting from column C

Example: User Interface Layout

	A	B	C	D	E	F	G	H
1	Name	Text:Name						
2								
3	Computer Programmer?	Check:Programmer		Age	Text:Age			
4								
5	Year	Combo:Year						
6								
7				Button:Cancel	Button:Save			
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								



Labels
(foreground
and
background
color set)



Combo box



Check box



Buttons
(foreground
and
background
color set)



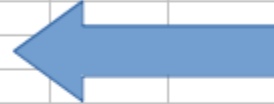
Text box

Example: Component Properties

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Combo:Year												
2		Source											
3		Data	2017	2018	2019								
4		Multiple	0										
5		Related											
6	Text:Age												
7		Size	3										
8		<u>MaxLength</u>	3										
9													
10													
11													
12													
13													
14													
15													
16													
17													
18													
19		Property											
20		Name											
21		in Column B											
22													
23													
24													
25													



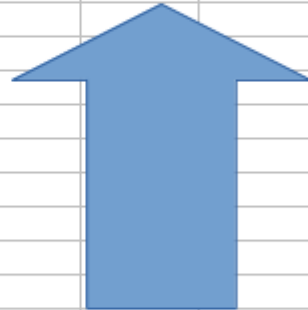
Properties of Combo box (Combo:Year)



Properties of Text box (Text:Age)

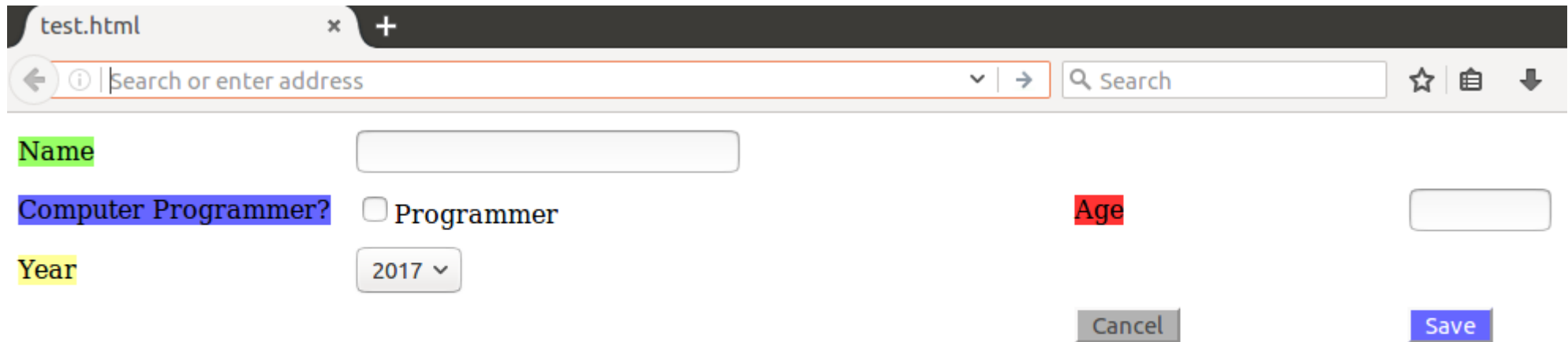


Property
Name
in Column B



Values Starting
from Column C

Example: Generated Web Page

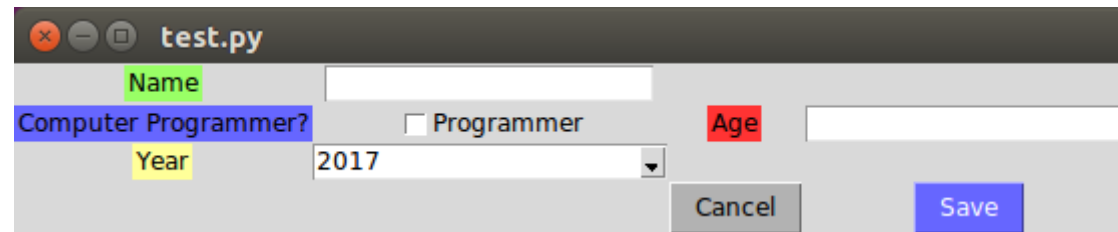


The screenshot shows a web browser window with a single tab titled 'test.html'. The address bar contains the text 'Search or enter address' and a search icon. The main content area displays a form with the following elements:

- Name**: A text input field.
- Computer Programmer?**: A checkbox followed by the text 'Programmer'.
- Year**: A dropdown menu showing '2017'.
- Age**: A text input field.
- Buttons**: 'Cancel' and 'Save' buttons.

The labels 'Name', 'Computer Programmer?', 'Year', and 'Age' are highlighted with colored backgrounds (green, blue, yellow, and red respectively).

Example: Generated Python Application (Tk Toolkit)



Implementation

- Python 2 application
 - Using Openpyxl library to work with Office Open XML spreadsheet
- Free/open source software
 - Download: <http://noprianto.com>

Experiments and Result

Respondents	Respondent Group	Using their own method or tool (development time in minutes)	Proposed Method (development time in minutes)	Comparison (percent)
1	A	25	20	80.0%
2	A	40	23	57.5%
3	A	38	15	39.8%
4	A	50	24	48.0%
5	B	105	28	26.7% → Best
6	B	65	30	46.1%
7	B	90	29	32.2%
8	B	48	30	62.5%

Respondent groups: (A) familiar with user interface development for multiple platforms
(B) Not familiar

Conclusion

- Proposed method: quicker user interface prototyping, for multi-platform applications
- Limitations:
 - Cell merging
 - Gravity of user interface components
 - Text alignment, both horizontal and vertical
 - Enabled/disabled state emulation