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

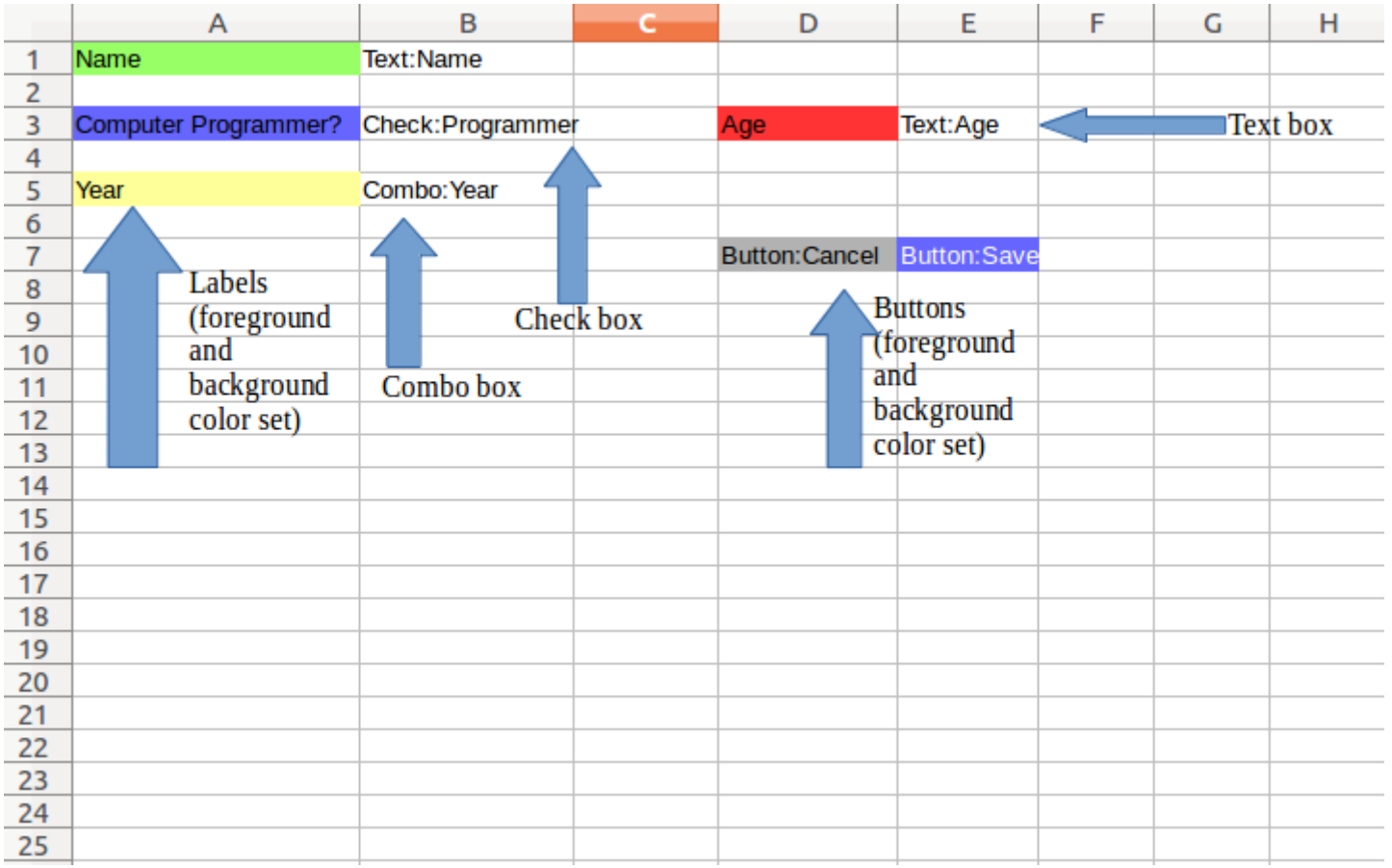
Method (1)

- 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

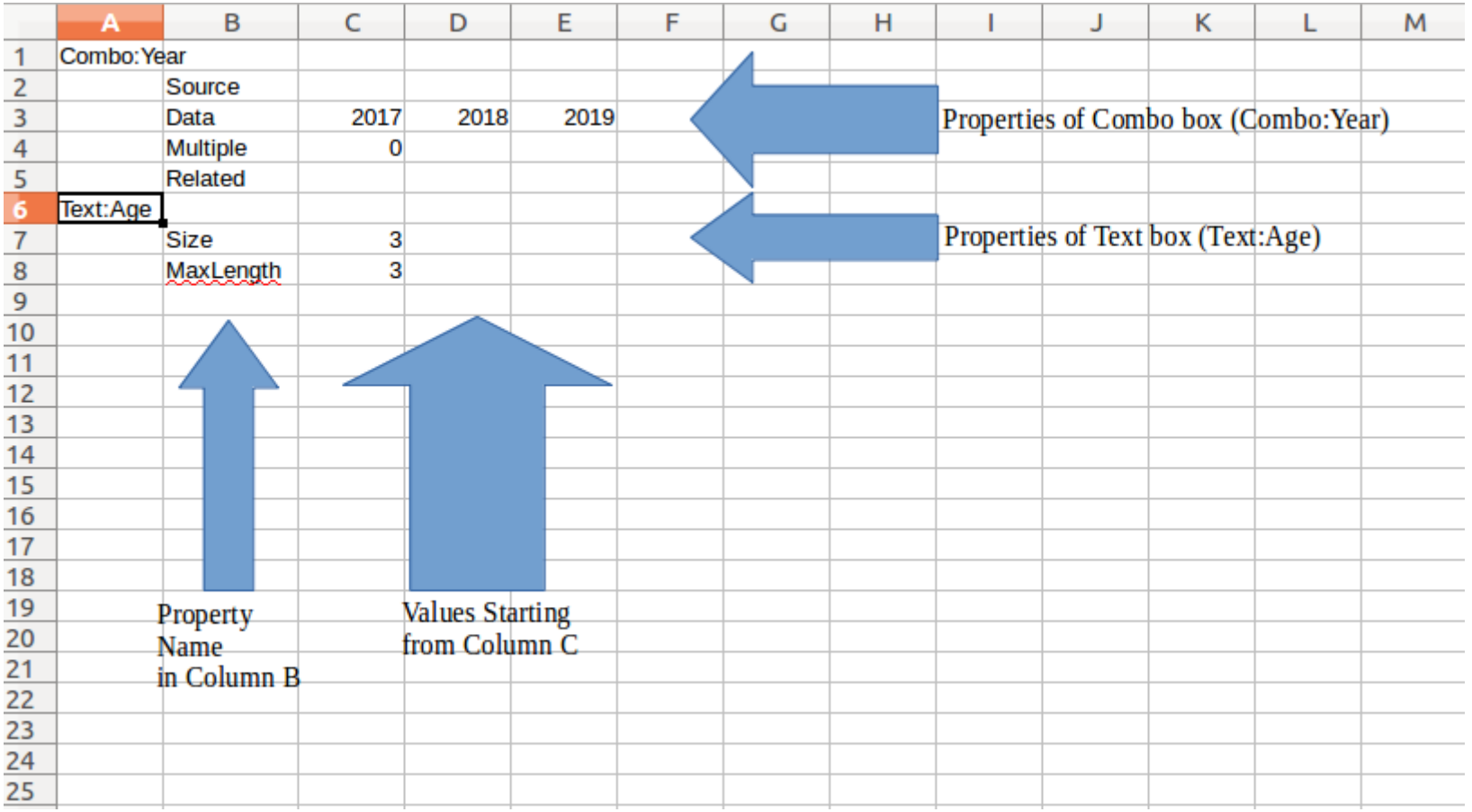
Method (2)

- 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

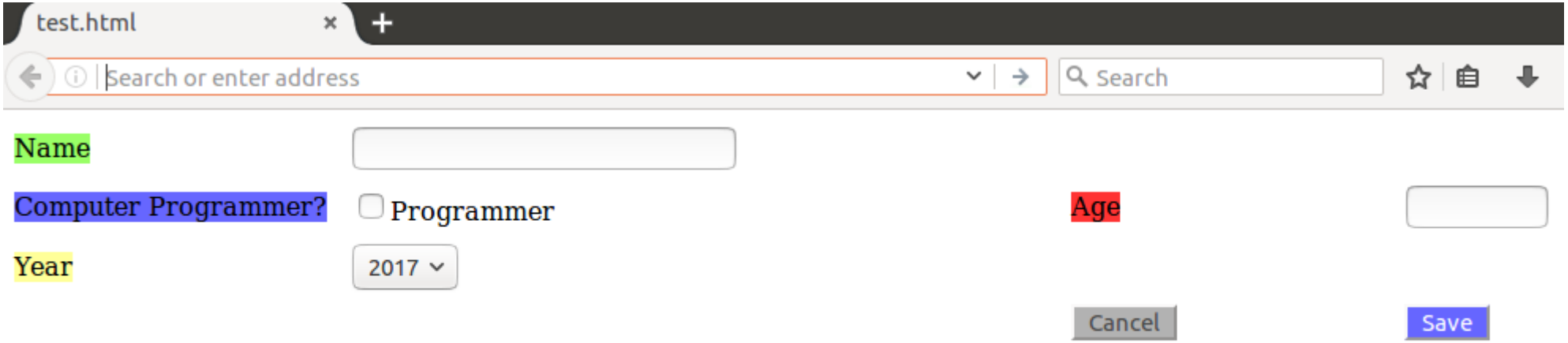
Example: User Interface Layout



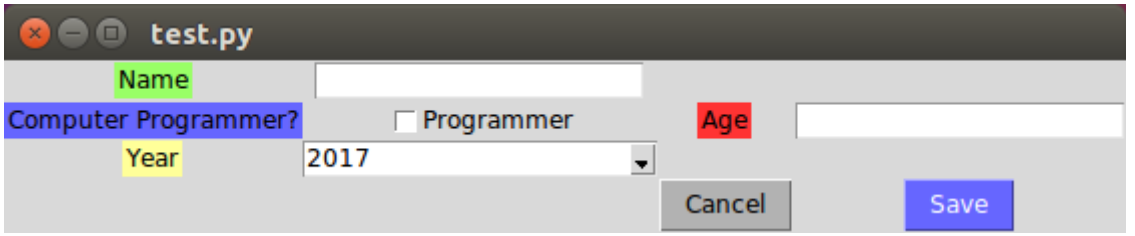
Example: Component Properties



Example: Generated Web Page



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