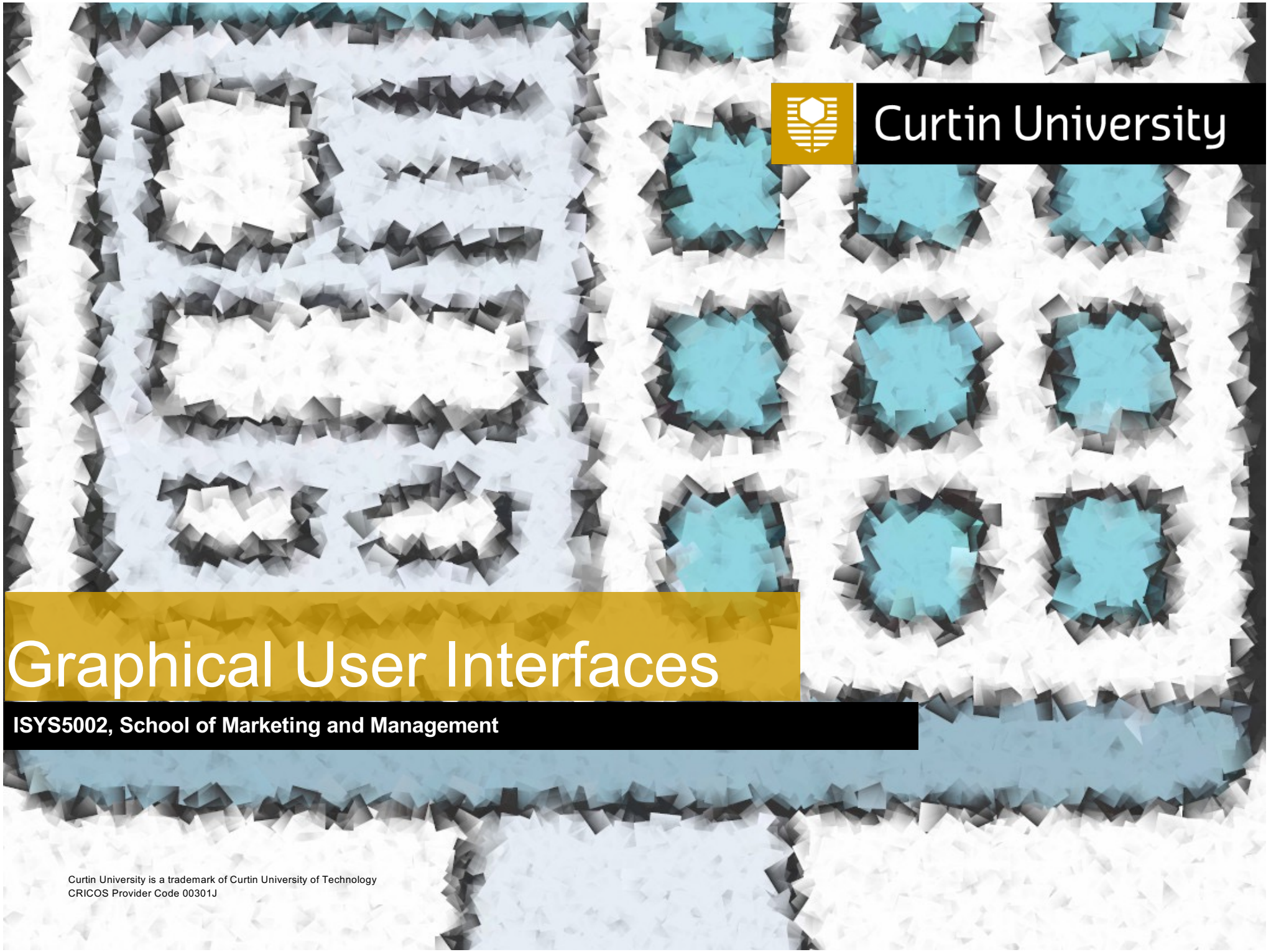


Graphical User Interfaces

ISYS5002, School of Marketing and Management

ELECTRONIC WARNING NOTICE FOR COPYRIGHT STATUTORY LICENCES

Curtin University

I acknowledge the traditional custodians of the land on which I work and live, and recognise their continuing connection to land, water and community. I pay respect to elders past, present and emerging.

Today

- Discuss User Interfaces
- Explain different interaction styles
- User support which should be built-in to user interfaces
- See how to create an ‘app’ using Python notebooks

The User Interface

- System users often judge a system by its interface rather than its functionality
- A poorly designed interface can cause a user to make catastrophic errors
- Poor user interface design is the reason why so many software systems are never used

Graphical User Interface

Characteristic	Description
Windows	Multiple windows allow different information to be displayed simultaneously on the user's screen.
Icons	Icons different types of information. On some systems, icons represent files; on others, icons represent processes.
Menus	Commands are selected from a menu rather than typed in a command language.
Pointing	A pointing device such as a mouse is used for selecting choices from a menu or indicating items of interest in a window.
Graphics	Graphical elements can be mixed with text on the same display.

Advantages

- They are easy to learn and use. Users without experience can learn to use the system quickly.
- The user may switch quickly from one task to another and can interact with several different applications.
- Information remains visible in its own window when attention is switched.
- Fast, full-screen interaction is possible with immediate access to anywhere on the screen

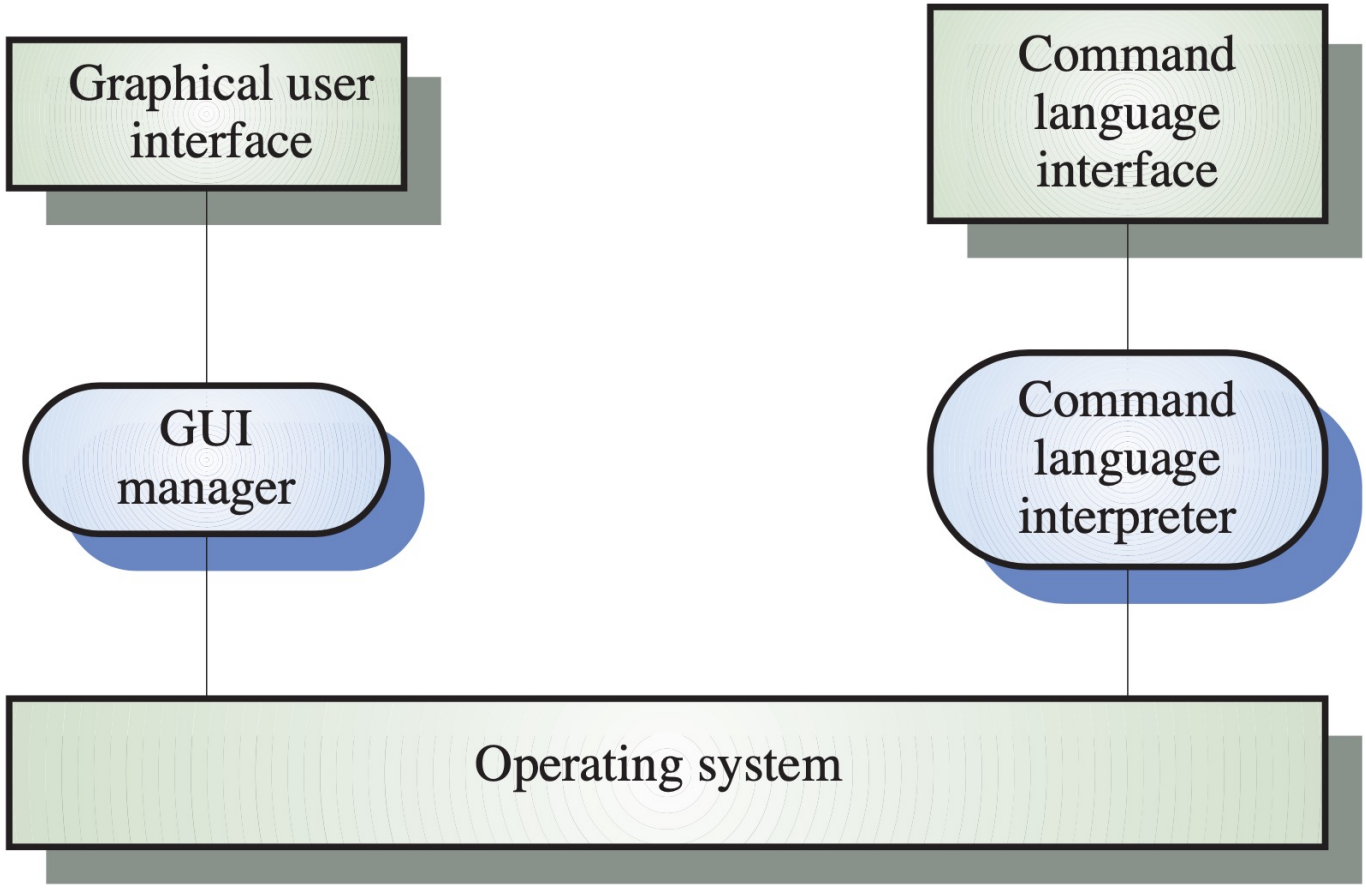
UI Design Principles

- User familiarity
- Consistency
- Minimal surprise
- Recoverability
- User guidance
- User diversity (Accessibility – ISYS3004)

Interaction Styles

- Direct manipulation (video games)
- Menu selection (general software)
- Form fill-in (loan application, stock control)
- Command language (operating systems)
- Natural language (Siri, Hey Google)

Modern Systems - Multiple Interfaces



User Support

- Guidance (Tool Tips)
- Help System
  - I want information
  - I need help!
- Error Messages
- User Documentation

GUI - History

- Punch Cards
- Text-Based (Teletype)
- Text-Based (Monitor)
- Sketch Pad (Ivan Sutherland's light pen)
- Stanford Online System (hyperlinks)
- Xerox PARC (first mouse)
- ….

Common Widgets

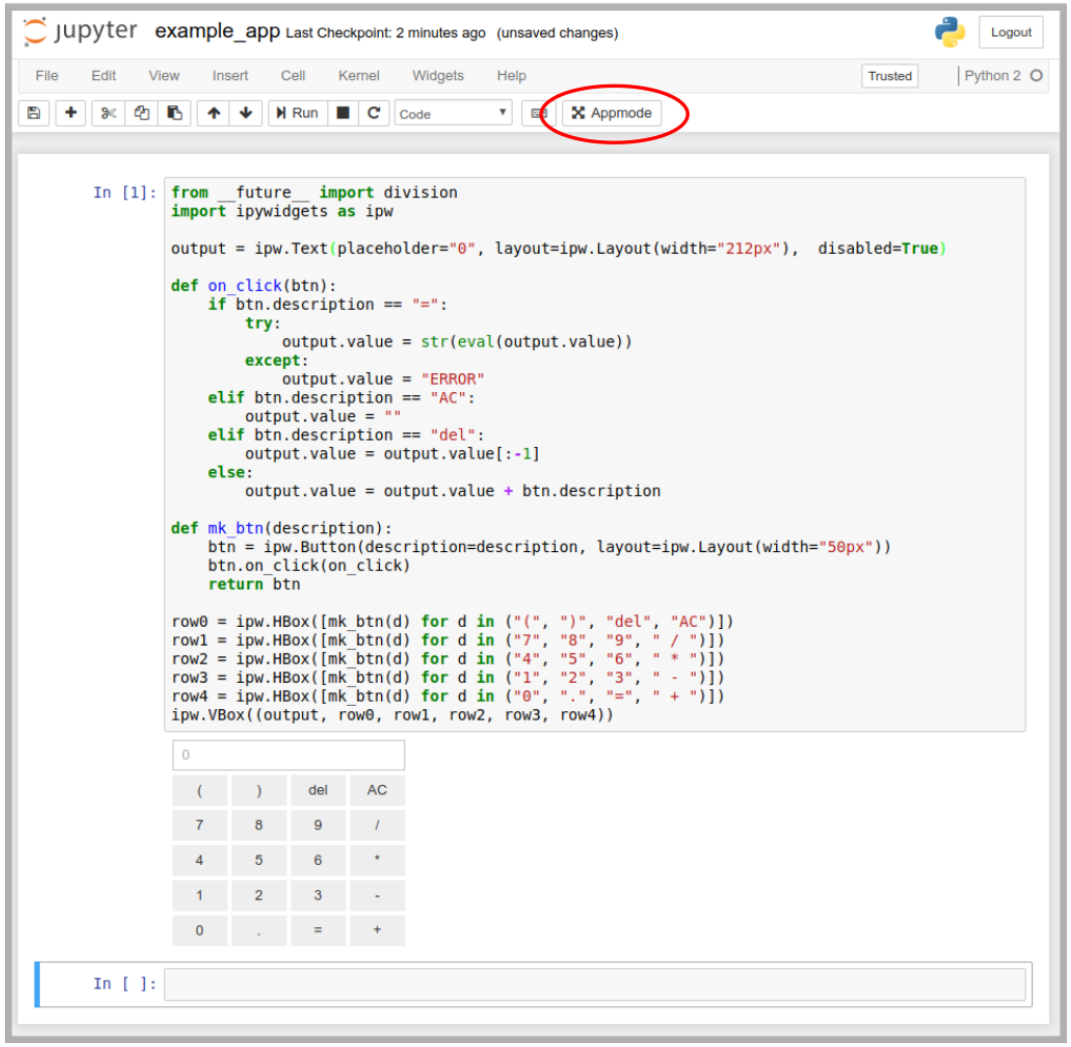
- Buttons
- Radio buttons
- Check Boxes
- Text Boxes
- Sliders
- Date Pickers
- Colour Pickers
- Tabs
- Canvas
- Plots/Charts

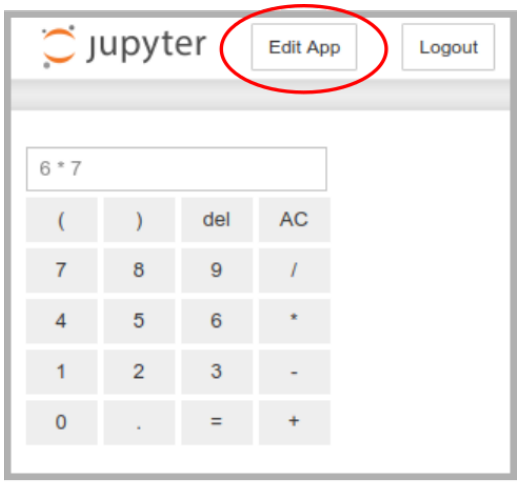
Curtin University

App (in a notebook)

- **Change variables**
- **Use input()**
- **Colab forms**
- **Widgets**
  - **Forms**
  - Dashboard
- Share
  - Module/Package (Nbdev -> PyPi)
  - Interactive (Binder, NBViewer)
  - Report (Slides, HTML, PDF)

App in a notebook





Curtin University

App (from a notebook)

- Native GUI (**tkinter**, pyQT, wxPython, PySide)

  Cross platform can be difficult

- Web Page (**cherrypy**, bottle, flask, Django, pyscript)

  Need to know some HTML and CSS

- Web App (**Anvil**, Streamlit, Volia, Dash, Panel)
- Native Cross Platform App (Kivy, Beeware)

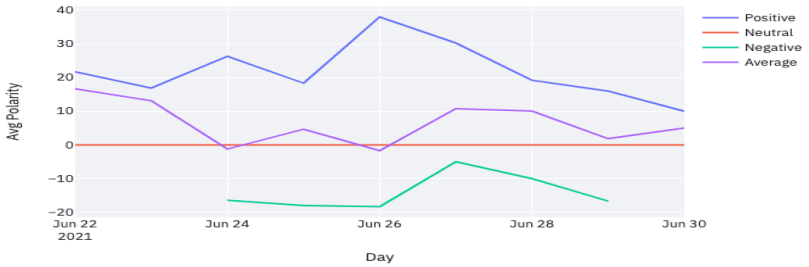
App from a Notebook. (streamlit)

Brandwatch

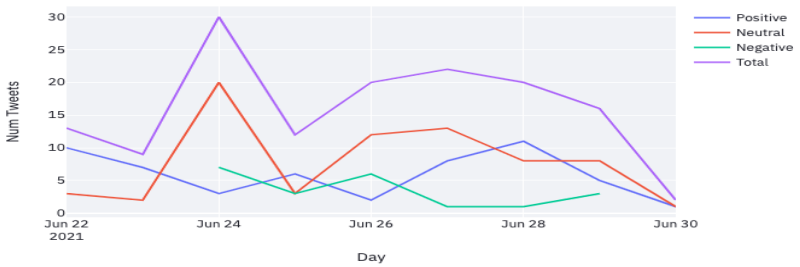
Search Twitter

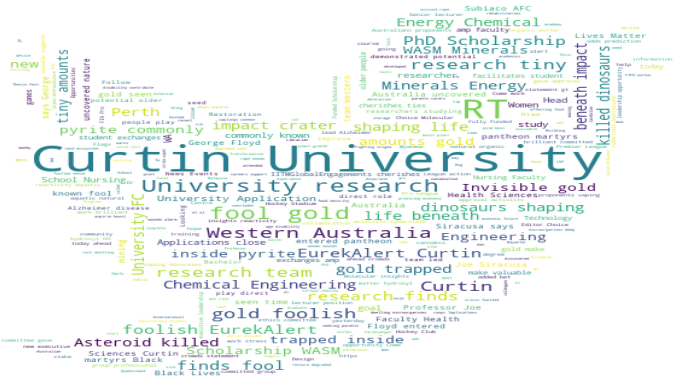
Curtin University

Sentiment over Time

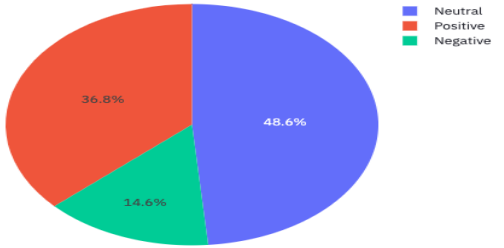


Tweets per Day



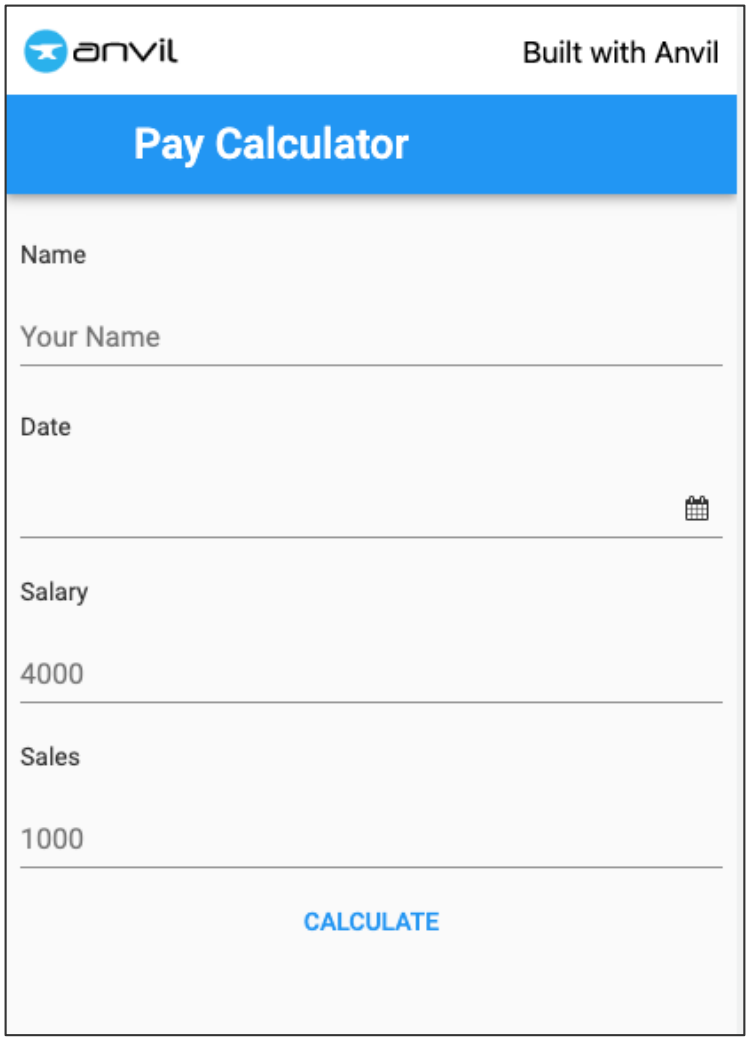


Sentiment Proportions



Curtin University

App from a Notebook (anvil)



Curtin University

Can you

- Suggest some general design principles for user interface design
- Explain different interaction styles
- Describe the user support which should be built-in to user interfaces
- Outline some of the ways create an ‘app’ using notebooks