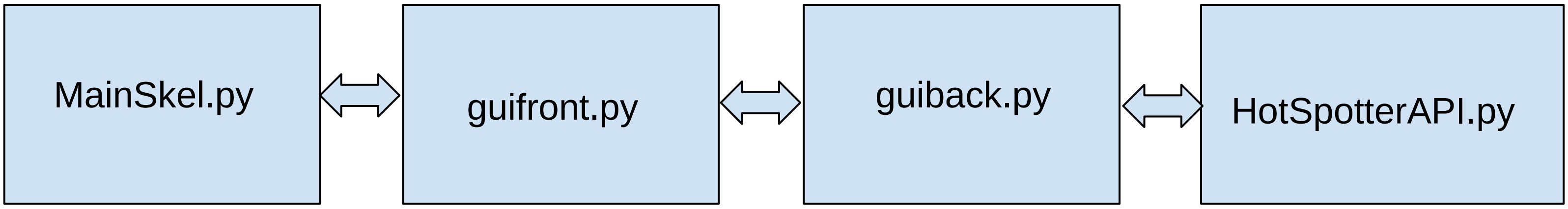
Graphical User Interface
Interface Control Document (ICD)

Created 3/11/17

Author: Matthew Dioso

- **Scope**
  This document details the updated design of the HotSpotter User Interface by ECE 17.7. It also describes processes in which commands in the User Interface are called within the HotSpotter API
- **Concepts**
  The HotSpotter User Interface is drawn out in the file MainSkel.py; the file populates a blank window using features defined within MainSkel.py (/hotspotter/hsgui/_frontend). Guifront.py (/hotspotter/hgui) adds functionality to the features defined in MainSkel.py by communicating with guiback.py (/hotspotter/hsgui). Within guiback.py is where calls to the HotSpotter API (hotspotter/hotspotter/HotSpotterAPI.py) are made. Any new functionalities added to HotSpotter should be placed within HotSpotter API and follow this method of access.



- **Interface**
  Example call: python main.py
  - Launches HotSpotter, should reflect any visual changes made in MainSkel.py

  GUI Updates:
  - Color of main window was changed to RGB(25, 25, 70) as specified in Panthera's branding documentation
  - resources_MainSkel.qrc (/hotspotter/hsgui/_frontend)
    - Qt resource file that contains references to image files displayed in HotSpotter
  - Image Table Buttons
    - Import Image file(s) (should be Import Image directory)
    - Save Database
    - AutoChip
  - Chip table Buttons
    - Save Database (will change to Save Query Results)
    - AutoQuery

New Functionalities:

- autochip(back) (in guiback.py)
  - Back is a reference to current *hs* instance
  - Creates path to template directory (image_directory/templates)
    - Fpath = get_work_directory() +’/templates’
  - Calls autochip from HotSpotter API with fpath as parameter
  - Populates chip table with newly generated chips
- autoquery(back) (in guiback.py)
  - Not yet implemented as of 3/11/2017
  - Two calls to HotSpotter API
    - Recognition on all chips
    - Chip association to determine individuals