1) High-Level Product Description
    a. Topic Area:
        Print 3D Models from Anywhere
    b. Functional Scope:
        Show real-time status of 3D printer by visualizing sensor data/ camera stream. Allow browsing and printing of existing 3D models saved to disk.
    c. What makes it useful:
        Allows 3D printing hobbyists to leave their printer physically unattended while they monitor its state remotely.
    d. Minimum Features for the App to Be Viable:
        - View Livestream of Printer Bed
        - View Actual/Target Temperatures of Bed/Hotend
        - View Current Printer Status
        - View Models saved to Disk
            o Give option to print
    e. Additional Features that would be Nice to Have:
        - View Time-lapses of Prior Prints
        - View Historical Statistics such as Print Time and Filament Usage
        - Preheat printer for certain materials
        - Print models from online sources such as Thingiverse

2) Kits and iOS Functionality Needed beyond UIKit
    - JSON API Calls for Access to 3D Printer
    - AVKit for displaying time-lapses

3) Functionality
    a. Navigation: UITabViewController, UINavigationController
        The application will be centered around a tab-based navigation system to allow quick switching between tasks.
        - Overview : Show small camera feed, printer status, relevant heating temperatures
        - Control : Show enlarged camera feed, have relevant safety controls (motors off, fan on, stop print)
        - Files : Show saved files on disk
            - File Detail View : Show rendered preview (if available), file metadata and allow print option for the selected file. This will be implemented through the use of a navigation controller.
        - Printers : Configure Saved Printers & Choose Current Printer
            - Printer Detail View : Show and allow editing of printer metadata. This will be implemented through the use of a navigation controller.
    b. Persistence:
        - NSUserDefaults
            To be used for storing printer configurations locally (See PrinterConfig in Data Models)
        - Remote Web-based API - Accessing Printer(s)
            Used for communicating with printer, API data will reflect target and current states, will be serialized into Swift objects (See Data Models)
    c. Background tasks off the Main Thread:
        - Poll for Printer State in Background
            Update temperature, camera feed, and printer status continually to ensure quick response as user browses the app.
    d. Custom views/images

- MJPEG Camera stream, live feed from 3D Printer Bed (https://github.com/freedom27/MjpegStreamingKit)
- Reusable UIViews, also allows for easier importing data from models
  Ex:
  - TemperatureView for repeatedly shown Temperature Data
  - BasicAuthView for repeatedly used Authentication Input
  - MJPEGCameraStreamView to wrap camera streaming and loading symbol together for easy reuse

e. Physical Sensing such as Acceleration or GPS
- Not as applicable for this type of app.

4) Basic Data Models

4.1) Local Data Models

- PrinterConfig
  - Stored Locally on Device
  - Properties:
    - (URL) : Printer Address
    - (CameraConfig) : Camera Config
    - (BasicAuthentication?) : Printer HTTP Basic Auth
    - (PrinterAuthentication?) : Printer HTTP Access Control
- BasicAuthentication, only needed to get past wholly secured printers
  - Stored Locally on Device as part of PrinterConfig Object
  - Properties:
    - (String) : HTTP Basic Auth Username
    - (String) : HTTP Basic Auth Password
- PrinterAuthentication, only needed for certain privileged operations
  - Stored Locally on Device as part of PrinterConfig Object
  - Properties:
    - (String) : API Access Key
- CameraConfig, can be manually configured or use defaults
  - Stored Locally on Device as part of PrinterConfig Object
  - Will initially be identical to printer auth
  - Properties:
    - (URL) : Camera Address
    - (BasicAuthentication?) : Camera HTTP Basic Auth

4.2) Remote Data Models

- CurrentPrinterState, shows current status of a tool
  - Stored remotely on Printer, Serialized from JSON to Swift Data Model
  - Fetched on regular interval, and sends global notification on change
  - Properties:
    - ([TemperatureData]) : Tool Temperature Data
    - (String) : Current State of Printer
    - (JobState?) : Current State of Print Job, if Printing/Paused
  - Methods:
    - (Void) : reloadData() : reloads state from printer, to be called from background tasks off the main thread
- TemperatureData, shows current temperature of a tool
  - Properties:
    - (String) : Tool Name
    - (Double) : Actual/Current Temperature
    - (Double) : Target Temperature
- JobState, shows current status of a print job

        - Properties:
            - (String) : File Name
            - (Int)   : Print Time Spent
            - (Int)   : Estimated Print Time Left
            - (Double) : Estimated Percent Complete
    - PrintableModels, shows printable models
        - Stored remotely on Printer, Serialized from JSON to Swift Data Model
        - Properties:
        - ([PrintableFileData]) : Files
        - Methods:
        - (Void)   : reloadData() : reloads files from printer
    - PrintableModelData, shows printable model
        - Properties:
            - (String) : Name
            - (String) : Path
            - (String) : Location
            - (String) : File Type
            - (Integer) : Size (bytes)
            - (Integer) : Upload Date (UNIX timestamp)
        - Methods:
            - (Void)   : print() : Selects and begins printing the file

Links:
https://github.com/freedom27/MjpegStreamingKit

http://docs.octoprint.org/en/master/api/printer.html#retrieve-the-current-printer-state

http://docs.octoprint.org/en/master/api/datamodel.html#temperature-data

http://docs.octoprint.org/en/master/api/datamodel.html#job-information

http://docs.octoprint.org/en/master/api/files.html#retrieve-all-files

http://docs.octoprint.org/en/master/api/files.html#issue-a-file-command