|  |  |
| --- | --- |
| Application documentation  Courtesy of Andrew Su | Documentation for classes and methods in Android Application for the Control Unit. Includes use cases, internal design, and examples.    Application Specific Control Unit for Neilsen and DOD Prosthetics Projects |

Contents

[JSon 1](#_Toc505604800)

[JSonManager 1](#_Toc505604801)

[JSonManagerException 2](#_Toc505604802)

[Application Activities 3](#_Toc505604803)

[MainActivity 3](#_Toc505604804)

[FrontLayout 3](#_Toc505604805)

[ChannelLimits 4](#_Toc505604806)

[Fragments 5](#_Toc505604807)

[Channel 5](#_Toc505604808)

[ChannelFragmentAdapter 6](#_Toc505604809)

[ChannelFragmentListener 7](#_Toc505604810)

# JSon

## JSonManager

File saving and loading protocol for userData using JSon format

**Note:** using the sdcard for storage also further external storage of data onto other devices and systems such as a dedicated server, cloud, or locked computer

public saveData(AppCompatActivity activity, PatientSettings data)

calls createDirectory()

creates a valid filepath to the file the data will be save to

**Requires:** sdcard must be inserted/active

permission WRITE\_EXTERNAL\_STORAGE must be enabled

**Result:** file has correctly been saved with patient data to filepath –

Filepath protocol: */sdcard/hybrid\_app/userData#.txt*

**Ex:** JSonManager.saveData(activityContext, user);

**public readData(AppCompatActivity** activity**, Context** context**)**

throws JSonManagerException

* Cannot open file from specified filepath
* Input stream reader of the file couldn’t not read all lines from file source
* Could not use JSon protocol to convert file to object

**Requires:** file sdcard must be inserted/active

file for user must exist in directory

permission READ\_EXTERNAL\_STORAGE must be enabled

**Result:** returns the PatientSettings object for user data retrieved from file

meant to retrieve patient data saved by saveData() method

**Ex:** PatientSettings userData = JSonManager.readData(this, this.getApplicationContext());

**private createDirectory()**

Creates the necessary folders/directories before returning full filepath prefix

Makes call to WRITE\_EXTERNAL\_STORAGE to create directory paths

Allows for appending */sdcard/hybrid\_app/….* to filepath

## JSonManagerException

# Application Activities

## MainActivity

Activity representing first screen loaded when starting app; can be used as title/info screen

uses activity\_main layout and only has one button to go to **FrontLayout**

**protected onCreate(Bundle** savedInstanceState**)**

uses standard AppCompatActivity process

attaches **OnClickListener** to button

## FrontLayout

Activity acting a switch screen to move to next functional activities

moves to **GraphList**, **UserList**, and **ChanneLimits** activies

**protected onCreate(Bundle** savedInstanceState**)**

uses standard AppCompatActivity process

attaches **OnClickListener** to buttons to start to respective activities

**public void onActivityResult** **(int** requestCode, **int** resultCode, **Intent** data**)**

when child activity from startActivityForResult() is ended

if the child activity has been ended without error *(RESULT\_OK)*, then will get the userData returned by the child activity and store it in the local scope

## ChannelLimits

activity that allows setting and saving of all the individual channel limits in the user/patient’s settings; communicates with **ChannelFragmentAdapter** and **ChannelFragment** attached to it

**protected onCreate(Bundle** savedInstanceState**)**

uses standard AppCompatActivity process

initializes userFileName if not already provided (*calls* *loadData(****String?*** *userFilename)* )

attaches ChannelAdapter and sliding TabLayout to ViewPager

**public dispatchTouchEvent(MotionEvent** event)

listener for a press-down on the device touch-screen

used to exit the keyboard setting when a press is made outside the keyboard

**public loadData()**

calls JSonManager.readData(AppCompatActivity, Context)

sets the userData stored in the activity to the loaded data from default file

**public PatientSettings loadData(String userFileName)**

same as loadData(), but can load non-default file for specific patient/user

# Fragments

## Channel

representation of a single channel for the stimulus pattern

**Public Channel(double** amplitude**, double** amplitudescale**, double** pulsewidth**, double** pulsewidthscale**)**

Constructor that sets the internal variables of amplitude and pulsewidth;   
modifies input amplitude/pulsewidth by the scale through division (value/scale)

if no modification needed, input amplitudescale and pulsewidthscale will be 1

**Public Channel(double** amplitude, **double** pulsewidth)

Constructor with default amplitude and pulsewidth scale of 1

**Ex:** return new Channel(amplitudeBar.getProgress(), amplitudeBar.getMax(), pulseWidthBar.getProgress(), pulseWidthBar.getMax());

## ChannelFragmentAdapter

Adapter attaches ChannelFragment references to the ChannelLimits class Creates and tracks all ChannelFragments used PatientSettings data to initialize fragment variables

**public int getCount**()

returns the number of fragments/tabs/items created and referenced in the Adapter

**public Fragment** **getItem(int** position)

standard getItem method used during creation/attaching of an Adapter to AppActivity

**ChannelFragment.newInstance** used for creation of each Fragment

**public CharSequence getPageTitle(int** position**)**

method called to attach the title of the tab for each fragment created

**public Object instantiateItem(ViewGroup** container, **int** position)

Extension of standard Adapter instantiateItem method to allow automatic method to create a fragment while saving the Tags for each fragment for future reference

**public void save()**

converts all local amplitudes and pulsewidth values to PatientSettings format to the local *user* variable, then saves the PatientSettings data to JSon file

calls **JSonManager.saveData** method to save data to file

**public void quit()**

quits out of the adapter with desired data

**public void setAmplitudeValue(int** position, **double** value)

sets amplitude values of selected channel

**public void setPulsewidthValue(int** position, **double** value)

sets pulsewidth values of selected channel

## ChannelFragmentListener

static based class that contains listeners and attached action protocols for views contained in ChannelFragment

**public SeekBarListener(TextView** target, **Callable<Void>** function)

attaches reference to related TextView and ‘function’ to current SeekBar;

Currently only uses OnStopTrackignTouch method response

Whenever SeekBar tracking is stopped, TextView’s value is updated to SeekBar’s current progress position and attached ‘function’ is called

**public EditorActionListener**(**SeekBar** target, **Callable<Void>** function)

attaches reference to SeekBar and ‘funciton’ to current TextView

**onEditorAction** standard listener for listening for start and finish typing in text editing

**matchTarget**  helper method to change SeekBar progress value to *adjustTargetValue*

**adjustTargetValue**  checks and parses the TextView for a numerical representation to place into SeekBar progress

**public ButtonListener(Executable** ex**)**

attaches a 'function' executable to specific clickable button

**onClick** calls attached function