Design Document

Group: Dec14-05

TV Remote Scheduling App. For Android™

Brianna Tegeler (Leader), Cimone Wright (Key-Idea), Dallas Thola (Communicator), and Josh Carritt (Web Master)

Version	Date	Author	Change
1.0	03/05/14	BCT	Initial Document
1.1	03/08/14	CLW	Compiling some individual contributions
1.2	03/12/14	BCT	Compiling rest of individual contributions
1.3	03/14/14	CLW	Formatting Issues and add missing components

Table of Contents

1 Introduction

1.1 PROBLEM STATEMENT

Television remotes typically have between 30 – 50 buttons. Setting a television with these remotes is impossible to do for many elderly people. Television programming schedules are often difficult to read or comprehend by many elderly individuals even if they have the mental and physical capability to select channels with a TV remote.

There are some very simple TV remotes that are marketed to support the geriatric population but many elderly people even have difficulty using these and they lack the programmability required to set up a pre-determined program schedule throughout the day or the week. Many people in nursing homes spend much of their day watching television yet they have a difficult time selecting television programming that meets their interests.

1.2 SOLUTION

To create a television controller and interface that allows for weekly television programming selection. This controller will autonomously change the channel based on the information programmed. This device will be geared toward elderly, physically and/or mentally impaired individuals.

1.3 SCOPE

Create a basic working prototype of the android application by the end of spring 2014.

1.4 DEFINITIONS, ACRONYMNS, ABBREVIATIONS

Term	Description
LIRC	Linux Infrared Remote Control
STB	Set Top Box
USB	Universal Serial Bus

1.5 REFERENCES

Irdroid™ user manual:

http://www.irdroid.com/wpcontent/uploads/2011/10/Irdroid_Users_Manual_1.0.pdf

Nexus 7 user manual:

http://nexusmanual.com/

---

2 System Design

2.1 SYSTEM REQUIREMENTS
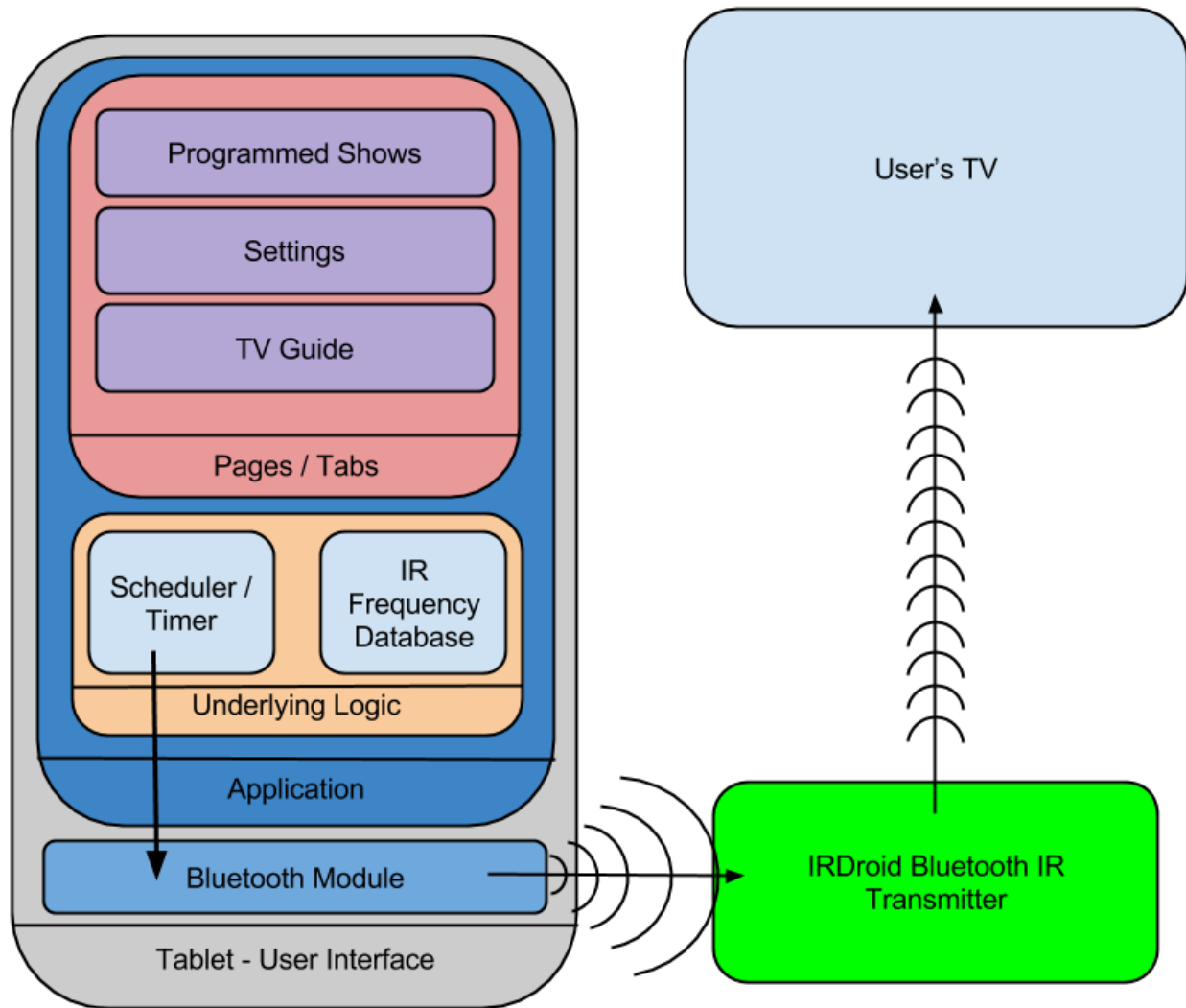
2.1.1 Functional Requirements

- Wireless Internet access
- Each module must be connected to the correct activity
- TV guide or cable data must be able to parse through in order for the user to be able to select a program to watch.
- Storage of user input must stay on the device until the user deletes it.
- Autonomously change channel
- Data must be available before channel programming can occur

- Limit access to authorized users

2.1.2   Non-Functional Requirements

- The hardware/software should change the channel fast.

- The buttons/text should be large for ease of use.

- The software should auto connect to hardware with Bluetooth.

- The hardware/software should be able to run 24/7

- The app should be able to automatically find the right TV code for the TV set up.

- The software should pull up the correct TV guide based on area.

2.2 FUNCTIONAL DECOMPOSITION

2.2.1   Overall Functionality
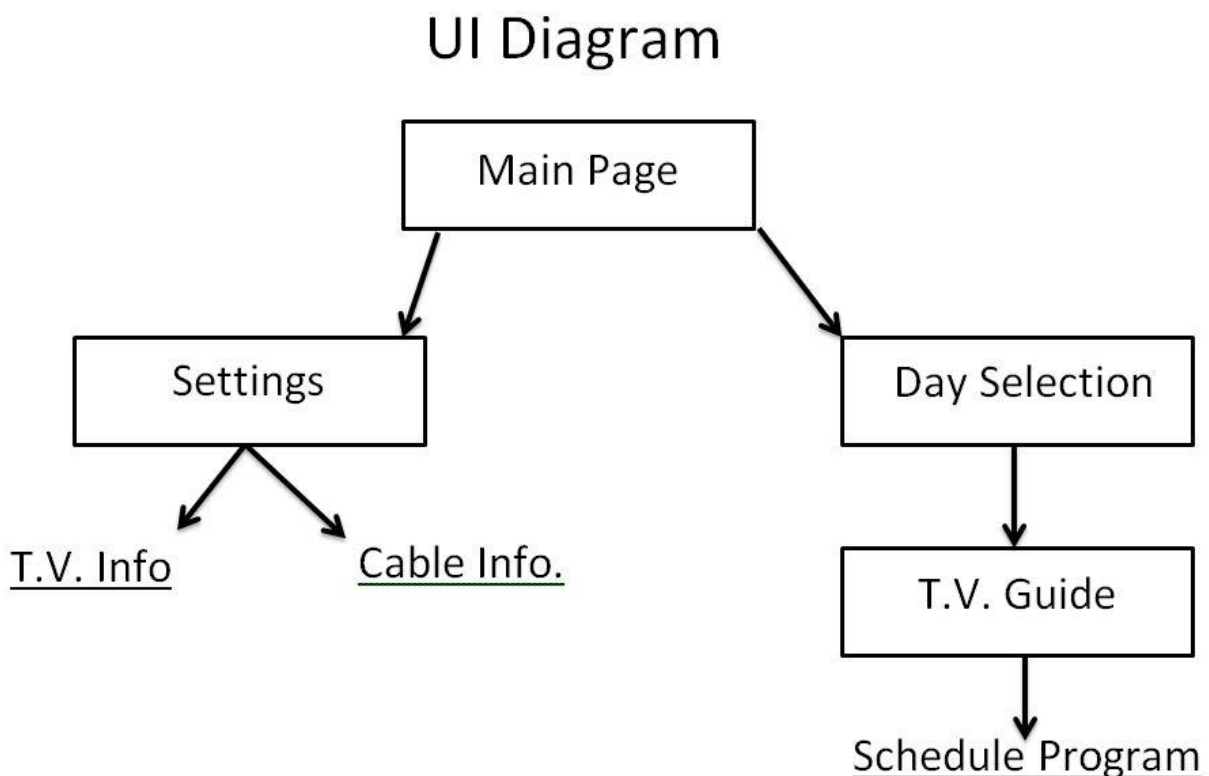
2.2.2   Sub-functions
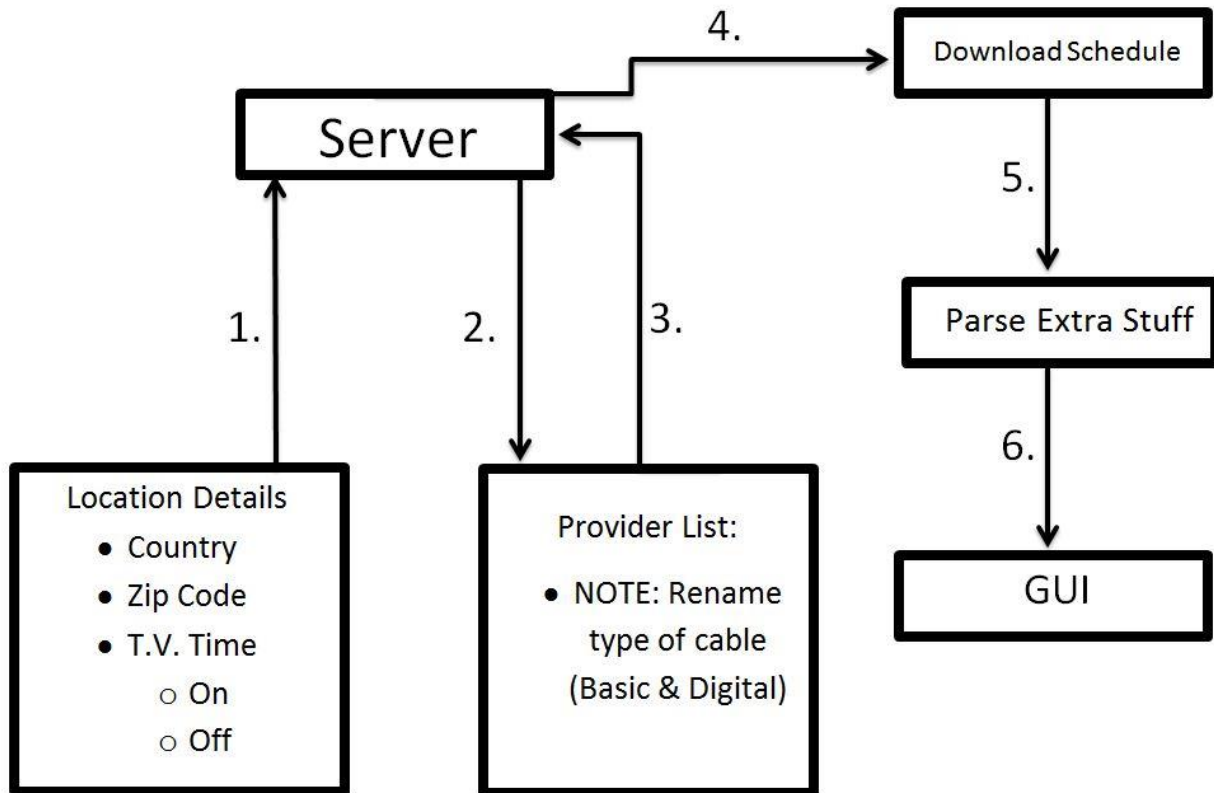
2.3 SYSTEM ANALYSIS

2.3.1 Sub-Function Processes

2.3.1.1 XML Parsing

To obtain the Television programming schedule we are employing the help of an application known as MC2XML which provides location and provider specific programming schedules for up to fourteen days. The program outputs an XML file containing much information we do not need so there arises the problem of how to extract the necessary information and inject it into our app. We solve this problem with a server side bulk parse. That will parse the critical information into memory and write a new, much smaller file to be sent to the app. This new file can then be opened by the app and read in much faster and the information can be displayed quickly.

2.3.1.2 Scheduler

The heart of the project is the scheduler which is where the app keeps track of when to switch the channel and turn the television on or off. This will be built mainly around interrupts, as they are able to wait to run a portion of code for a set amount of time. With one of these set for each channel change we assure that the channel is changed at the correct time.

2.3.1.3 IR-Droid Functionality

With the decision to use the IR-Droid V2.0 we are also able to inherit the functionality that is already programmed to work with the device. The Developer of the IR-Droid has released open-source code for interfacing with the device as well as basic remote functionality. The IR-Droid is able to control the televisions using the LIRC database which contains the frequency codes for all major brands of televisions. Allowing the device to change the frequencies that it is sending to be able to tailor to many different televisions.

2.3.1.4 Setup and Settings

The application we are developing will require initial setup by the user, including setting location, television programming provider, and when to turn the TV on and off. These settings will allow the app to interface with the server to obtain the correct programming information as quick as possible.

2.3.2   Relation of Requirements and Functions/Sub-Functions

2.3.2.1 XML Parsing

The parser interacts both with the settings page and the TV Guide page, as the parser needs to pull in the TV on and time off times to be able to clip the guide into as small of a display as possible so the user is not overwhelmed with more information than needed, and then once the data is received and loaded, the parser passes the data to the TV Guide page to be displayed for the user to select which programs to watch.

2.3.2.2 Scheduler

The scheduler interacts with the TV Guide page and the settings page. The TV Guide page acts as a user friendly control for the scheduler. When a show is selected to be watched. The scheduler creates a timer to change the channel at the correct time to watch the specified show. The settings page is used to know when to schedule the TV on and off functions so the program does not attempt to change the channel when the TV is off.

2.3.2.3 IR-Droid Functionality

The IR-Droid code interacts with the main page of the app that functions as a very basic remote control allowing the user to turn the tv on or off, or change the channel up or down. The Scheduler also uses the IR-Droid code. When a timer goes off and the app needs to change a channel it calls the IR-Droid with the appropriate channel number and the IR-Droid code uses the Bluetooth interface to send the correct signal to the IR-DROID V2.0 to send the IR signal to change the channel on the Television.

2.3.2.4 Settings

The Settings page interacts with the XML Parser to set the location, provider, and time information to tailor the experience to the user's needs. The Settings page interacts with the scheduler to set the boundaries of when timers can be set as well as when to power on/off the television. The IR-Droid also uses the Settings page to obtain the brand information for the television so that the correct IR frequencies can be sent to the television.

3 Detailed Design

3.1 INPUT/OUTPUT SPECIFICATIONS

3.1.1   Main Module

- IR/Bluetooth connection

3.1.2   Schedule Module

- Day

- Parser information

- List of programs scheduled by the user currently

3.1.3   Settings Module

- Wireless

  - Network
  - Password (if applicable)

- TV Information

  - TV brand
  - TV model

- Cable

  - Country
    - State
    - Zip code
  - Provider
    - Cable package

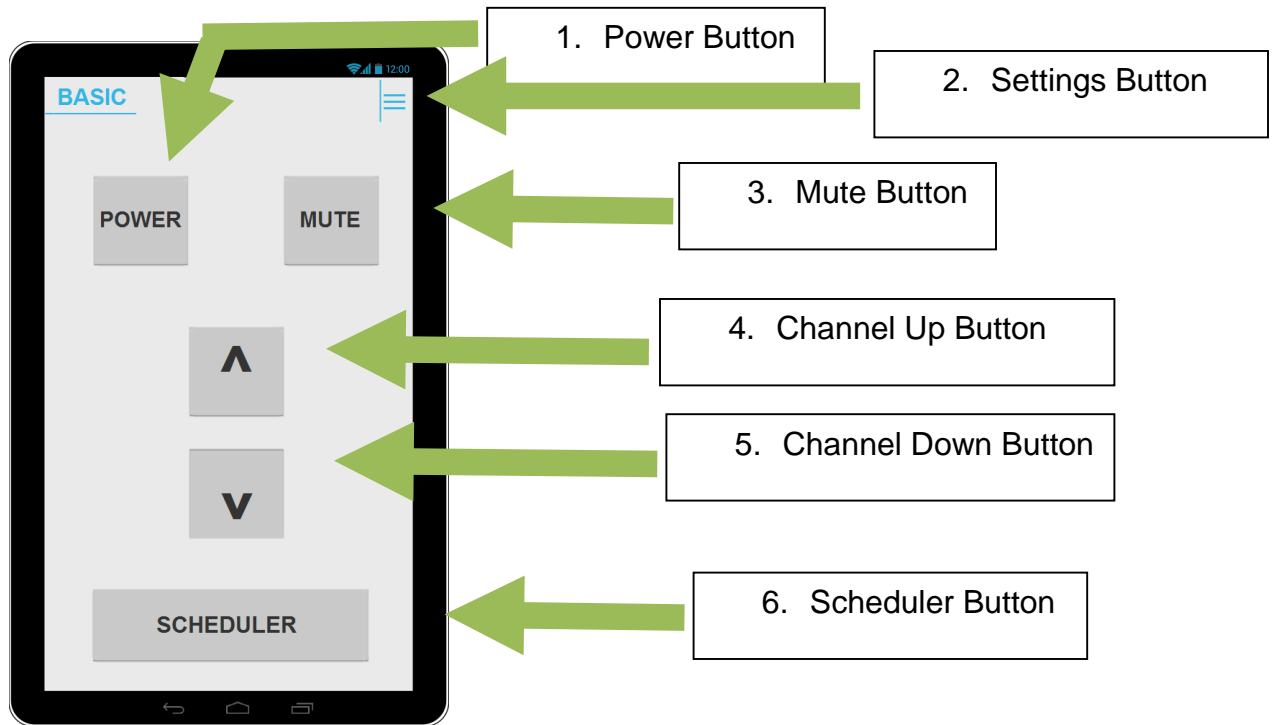## 3.2 USER INTERFACE SPECIFICATION

Figure 1: Main Page

Main Page: This is the first page the user will see when opening the android application. Power button allows the user to turn the television on and off. Mute button allows the user to turn the sound on and off with the button of a button. Channel Up allows the user to increases the channel number by increments of 1. Channel down allows the user to decrease the channel number by increments of 1. Scheduler button takes the user to the scheduler page. Settings button takes the user to the settings page.
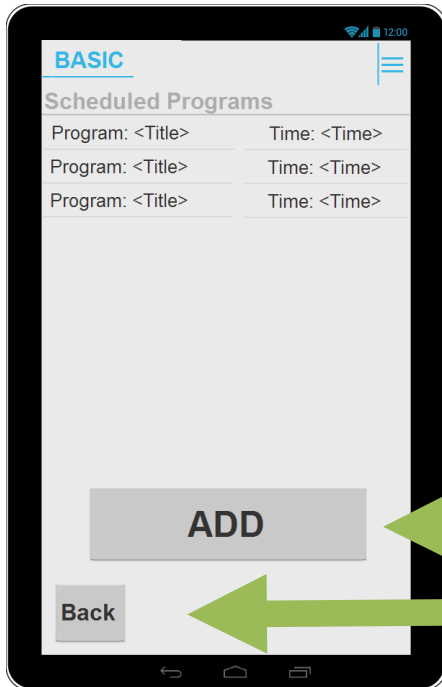
Table 1: Sample TV Guide

Figure 2: Schedule Page

Schedule Page: This page will display the program titles and times of what will be showing for the day selected. Eventually, this screen will look similar to Table 1. The user will be able to click on the program and push the add button to add the program to their personal schedule. The personal schedule is a list of programs chosen by the user, the device will automatically change the channel when something on the list of personal programs is ready to start. The back button will take direct the user back to the main page.
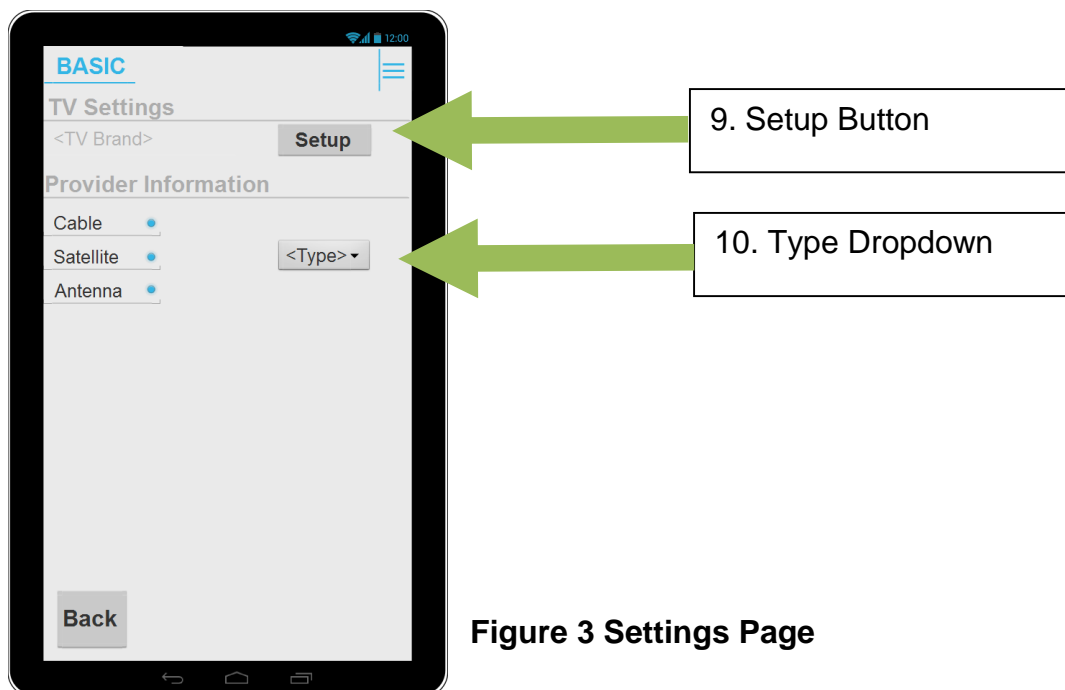

Figure 3 Settings Page

Settings page: This page is used to setup and/or change the television information or cable information. When the setup button is pushed a dialog box will pop up allowing the user to select the brand and type of television that the device will be connected to. The type dropdown will populate a list of possible cable providers and the type of cable (basic, digital, or satellite, etc…) for the user to select.

3.3 HARDWARE AND SOFTWARE SPECIFICATIONS

3.3.1   Hardware Specification

- Irdroid™: A device that sends information through infrared and Bluetooth.

- Tablet: A compact computer with multiple sensors, a display, and single battery unit.

Figure 4: Irdroid flat view

Figure 5: Irdriod side view

3.3.1.1 Supported Remotes:

- <u>LIRC Remotes (see a list)</u>
- Samsung
- LG
- Motorola
- Sony
- Panasonic
- Philips
- Communication interface – Bluetooth (A2DP protocol)
- Remote control range 30 meters for Bluetooth (LOS)
- Cool Laser Cut enclosure

3.3.1.2 Module specs:

- Power supply 12V DC
- Bluetooth specification v2.1 + EDR
- Bluetooth Class II device
- Bluetooth Audio Profile (A2DP)
- IR LED wavelength 940nm

3.3.1.3 Package Contents:

- Irdroid v2.0 – 1pcs
- User Manual – 1pcs
- Irdroid Handbook – 1pcs
- Power supply cable – 1pcs

3.3 SOFTWARE SPECIFICATION

There will be an android application that to control the television and channel the channel when the scheduled program is within 30 seconds of starting. A website will complement application to allow remote assistance in configuring which shows will be watched when.

3.4 TEST PLAN

3.4.1 Test Environment

3.4.1.1 JUnit

JUnit is a Java testing environment to test whether the modules for a program are working correctly.

3.4.2 Test Design for Checking Wireless Access

3.4.2.1 Function/Requirement Tested

- Wireless Internet access

3.4.2.2 Procedure

- Check the wireless flag on the hardware

3.4.2.3 Evaluation of Test Results

- True = Wireless Access granted
- False = Wireless Access denied

3.4.3 Test Design for Module Connectivity

3.4.3.1 Function/Requirement Tested

- Autonomously change channel
- Each module must be connected to the correct activity

3.4.3.2 Procedure

- Run application and test each utility to see if they work correctly with their descriptions

3.4.3.3 Evaluation of Test Results

- Passed = All utilities work correctly with their descriptions
- Failed = All or some utilities do not work correctly with their descriptions

3.4.4 Test Design for Parser

3.4.4.1 Function/Requirement Tested

- TV guide or cable data must be able to parse through in order for the user to be able to select a program to watch
- Data must be available before channel programming can occur

3.4.4.2 Procedure

- Give the parser fake data and then run to make sure it gives out the data needed for the application

3.4.4.3 Evaluation of Test Results

- Passed = All fake data was parsed correctly
- Failed = All or some fake data was parsed incorrectly

3.4.5   Test Design for TV Selection

3.4.5.1 Function/Requirement Tested

- The app should be able to automatically find the right TV code for the TV setup

3.4.5.2 Procedure

- Run application's TV setup program in the settings page and test to see if the TV selected on the application matches the TV

3.4.5.3 Evaluation of Test Results

- Passed = TV setup program selected the correct TV
- Failed = TV setup program selected the incorrect TV

3.5 IMPLEMENTATION ISSUES

3.5.1.1 Accessing TV Guides

Getting access to all the cable providers' television guide is not as easy as we thought it would be. Many providers require you to pay to get that information. We have found a database that gives us this information for free, but does not include every cable providers. This database includes the major distributors, which is sufficient for now.

3.5.1.2 Irdroid shipment date

Irdroid shipment/arrival is unknown and we have contacted the company, but have not received any response. We need this device to be able to change the channel using the application we are developing on the nexus 7 tablet. If this device does not arrive by the

---

time we return from spring break will design and implement our own version of this device.

3.5.1.3   Android eclipse environments

All the team members have our eclipse setup and configured differently. This is a problem because every time someone pushes to the repo and someone else pulls from the repo, eclipse will throw a ton of errors. To solve this problem we will be making a standard configuration for all members to follow.