

MakeSafe

*Developing a Mobile Application to Assist Probationers with
Better Decision Making*



Developer Guide

V2.1

COMPUTER SCIENCE DEPARTMENT

TEXAS CHRISTIAN UNIVERSITY

May 5 2015

Revision Signatures

Sign below if you a) have read the entire document b) found the information within to be accurate and c) are certain the document is free of grammatical and spelling errors.

Name	Signature	Date
Marcus Beal		
Matt Butz		
David James		
Zack Morris		

Revision History

Version	General Description of Changes	Date
V1.0	Initial Draft	2/27/15
V2.0	Final Publication	4/23/15
V2.1	Final Revisions	5/5/15

Table of Contents

REVISION SIGNATURES.....	II
REVISION HISTORY	III
1 INTRODUCTION	1
1.1 PURPOSE.....	1
1.2 OVERVIEW	1
1.3 HOW TO GET STARTED	1
2 OVERALL DESCRIPTIONS.....	2
2.1 PRODUCT PERSPECTIVE	2
2.2 DEVELOPMENT ENVIRONMENT	3
3 SYSTEM ARCHITECTURE.....	4
3.1 SYSTEM OVERVIEW.....	4
3.2 SESSION STRUCTURE	5
3.3 JSON STRUCTURE	6
3.4 INPUT FOLDER (STAYSAFE) FILE STRUCTURE	7
4 CODE LAYOUT.....	8
4.1 STAYSAFE CODE STRUCTURE.....	8
4.2 STAYSAFE CLASS HIERARCHY.....	13
4.3 KEEPSAFE CODE STRUCTURE	14
5 DATA EXPORT STRUCTURE.....	16
5.1 OUTPUT DATA VARIABLES	16
6 GLOSSARY OF TERMS.....	17

1 Introduction

1.1 Purpose

The purpose of this document is to provide developers with the appropriate information so that they can modify the StaySafe and KeepSafe applications. It includes a general overview of the application and how the entire system interacts with different aspects of the software.

1.2 Overview

Section 2 – Overall Description: Brief detail of the project origins and concept

Section 3 – System Architecture (description with diagram): Interactions between application components

Section 4 – Code Layout: Class and method structure

Section 5 – Data Export Structure: File system and naming conventions

Section 6 – Glossary of Terms: Glossary of terms used in document

1.3 How to Get Started

These are the files and software necessary to begin the StaySafe process.

- StaySafe.json – The input file for the StaySafe application
- StaySafe.apk – The actual application that must be installed on the tablets
- StaySafe – The folder on the tablet that holds all of the images, audio, and video files for the application
- KeepSafe.jar – A stand-alone Java application that can update the StaySafe.json file
- Android Tablet – StaySafe only runs on Android and was designed specifically for 10” tablets
- Eclipse Luna v4.4.0 with the Android Development Toolkit v23.0.3

2 Overall Descriptions

2.1 Product perspective

The MakeSafe senior project, sponsored by IBR, developed the StaySafe application to meet IBR's design requirements. Major objectives of the project included developing the StaySafe app application for Android, saving research data (e.g., user selections, usage statistics, and user survey responses), and creating the version-publishing application. The KeepSafe software tool provides the researchers with sustainability, a means of publishing future versions of the application without further developer involvement.

There are three major pieces to the project. For point of clarification they are:

- MakeSafe – The name of the overall project
- StaySafe – The application for android developed by the MakeSafe Team
- KeepSafe – A stand-alone Java Application used to edit dynamic content

Tablet application:

StaySafe is the tablet application for Android that provides a simple, intuitive, interface for a population with either limited education or limited computer literacy. There will be twelve sessions lasting approximately 10-15 minutes each. The sessions are divided into three distinct templates each of which has a similar structure. The first session is an initial demonstration followed by eight WORK-IT sessions and three participant choice. Further information is depicted in Section 3.2 of this document. The eight WORK-IT sessions, while following a standard template, will differ in terms of text, audio, and video content.

Research Assistant laptop:

The Research Assistant will have a Windows or Mac based laptop to serve as a mechanism to administer, manage, and track the completed application sessions. Session interaction data will be downloaded from the tablet to the laptop. The data will be added to a controlled database by the IBR staff for subsequent analysis.

KeepSafe:

KeepSafe is a Java application that can run on any windows based laptop running Java version 8. KeepSafe allows for the dynamic content in the StaySafe application to be updated. KeepSafe outputs a JSON file which can be read by the StaySafe application.

2.2 Development Environment

StaySafe will use agile methodologies in order to meet the requirements agreed upon between the TCU IBR team and the Computer Science student development team.

Programming Environment

- MAC or Windows PC running:
 - Eclipse Luna v4.4.0 with the Android Development Toolkit v23.0.3
 - Excel 2010
 - JSON editor v3.1.1 www.jsoneditoronline.org
- Android tablet running OS 4.4.2 API 19 or higher

Support Environment

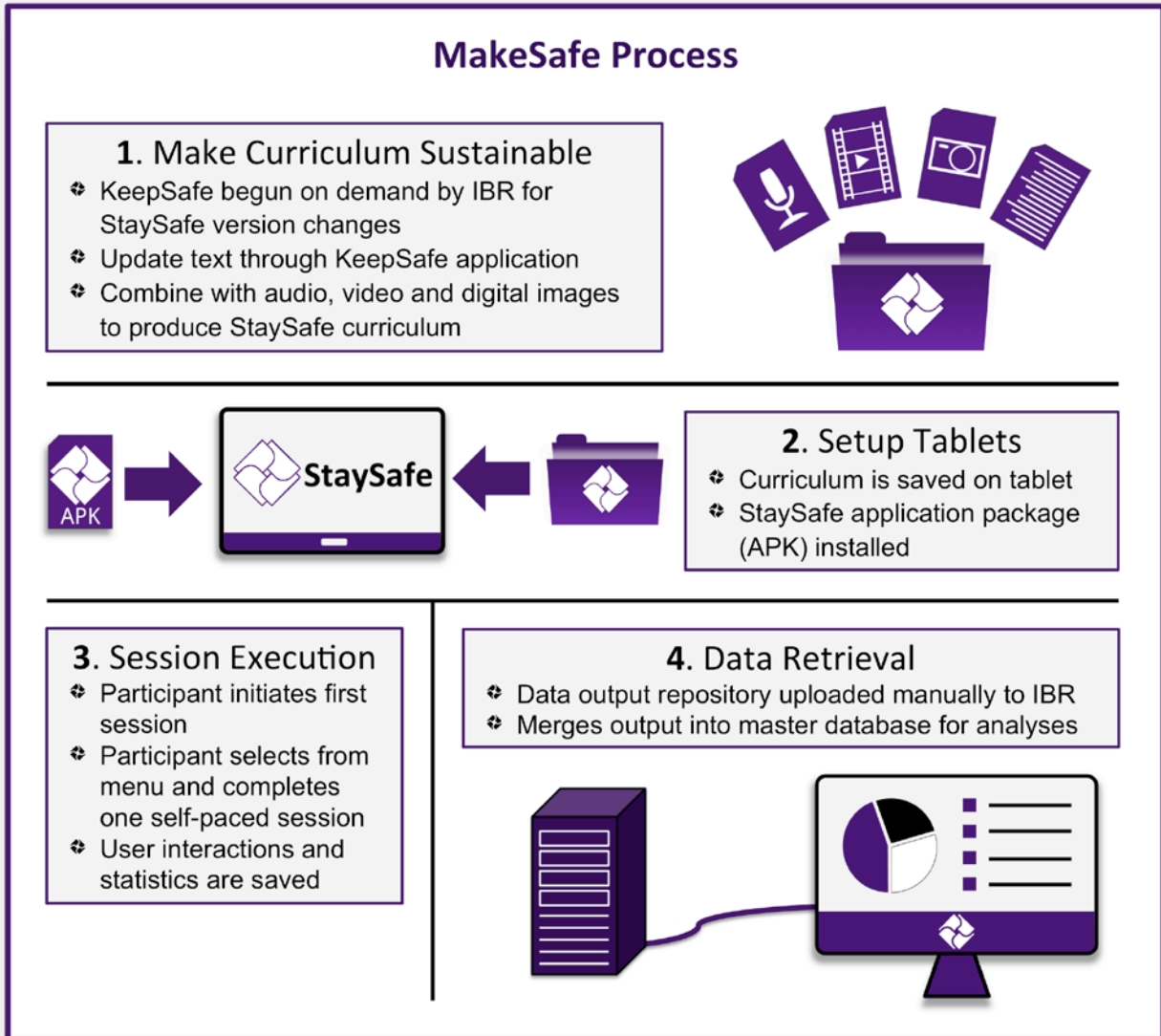
- Core FTP Lite 1.3c
- FileZilla
- Java Version 8 update 45 on the machine running KeepSafe

Other Software

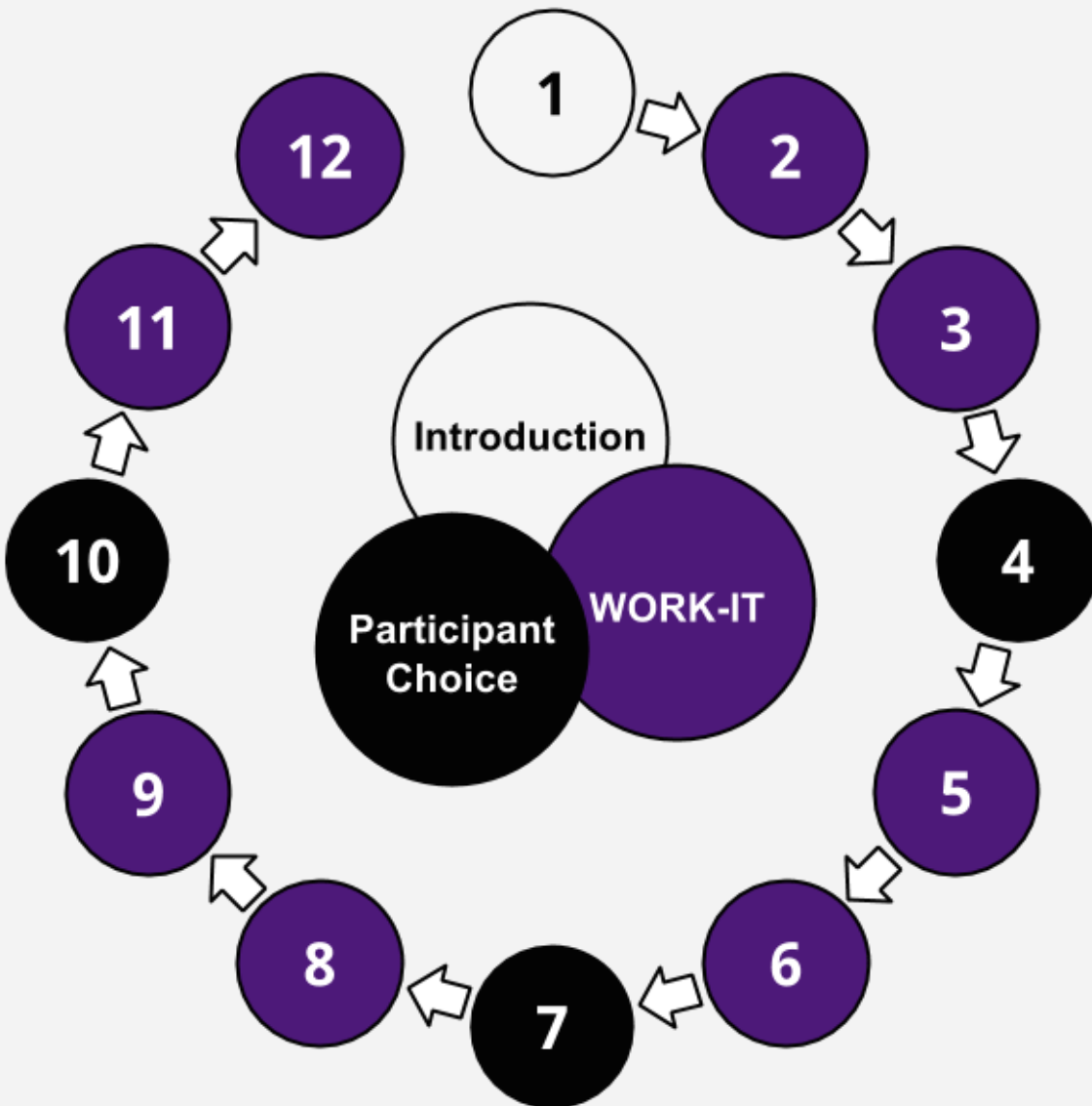
- Microsoft Office 2010
- Google Drive
- Adobe Photoshop CS6
- Microsoft Visio 2013
- Notepad++ 6.5.1

3 System Architecture

3.1 System Overview



3.2 Session Structure



Session Descriptions:

- Twelve sessions lasting approximately 10-15 minutes each.
- Three distinct templates each of which has a similar structure.
 - Template 1 –(Intro) An initial demonstration.
 - Template 2 – (WORK-IT) Eight WORK-IT sessions.
 - Template 3 – (PARTICIPANT CHOICE) Three participant choice sessions.

3.3 JSON Structure

This section defines the structure of the JSON file read in by the application. It encompasses 1600+ different values and is generated by the KeepSafe application.

```
"Version":
"Template1":
  "People":
    "ProblemXX":
      "Options": []
      "Game": []
      "AffectedPersonsFacts": []
      "OptionsFacts": []
      "SurveyQuestions": []
      "consequences": ""
      "choice": ""
    "ProblemChoices": []
  "Places":
    "ProblemChoices": []
  "Things":
    "ProblemChoices": []
  "Template1SL2": ""
  "Template1SL3": ""
"Template2":
  "Things":
    "ProblemXX":
      "APTtitle": ""
      "Options": []
      "Game": []
      "AffectedPersonsFacts": []
      "OptionsFacts": []
      "SurveyQuestions": []
      "consequences": ""
      "choice": ""
    "ProblemXX": {}
    "ProblemXX": {}
    "ProblemXX": {}
    "ProblemChoices": []
  "People": {}
  "Places": {}
"Template3": {
  "Session4":
    "Game": []
    "fact1Body": ""
    "fact1Title": ""
    "SurveyQuestions": []
    "fact2Title": ""
    "fact2Body": ""
  "Session7": {}
  "Session10": {}
```

3.4 Input Folder (StaySafe) File Structure

Below is the general structure of the input folder that the StaySafe application reads in. This includes audio, video, images and text required for the StaySafe application to function normally. The structure of the file system is crucial to the StaySafe application functionality.

Root Storage

/StaySafe

/VoiceOver – Folder containing the audio files for the application. Each audio file must follow the proper naming convention in order for the application to be able to read it in.

/Template2L10.wma

/Template2SL11.wma

/...

/Videos – Folder containing the video files for the application. Each video file must follow the proper naming convention in order for the application to be able to find the video.

/Template3SL5Session7.mp4

/Template3SL5Session4.mp4

/...

/Template2SL3PT3.mp4

/Template2SLPT4.mp4

/...

/Images – Folder containing the image files for the application. The images must follow the proper naming convention in order for the application to be able to display them.

/Template3SL5S10.png

/Template3SL5S4.png

/...

/Template2SL4PT4.png

/Template2SL4PT3.png

/...

/Data – The Folder where the application writes out the user interaction data for IBR to analyze.

/StaySafe.json (See Section 3.3)

4 Code Layout

4.1 StaySafe Code Structure

Every class within the StaySafe package performs unique tasks. All classes are Android Fragments with the exception of the Login, Template1, Template2, and Template3 classes that are Android Activities. Furthermore, the ErrorFragment and the ExitUtility are a special type of Fragment known as a DialogFragment. In this section a general overview of the functionality of all classes are provided. The classes are designed to be used interchangeably across templates and can even be altered by the calling activity to alter the base appearance of the screen displayed by the class. Each class has a corresponding layout file that loads the user interface required.

Affected - The Affected Fragment is responsible for displaying the affected persons screen to the user. The affected persons screen consists of 4 affected persons and 4 corresponding facts. When the user selects an affected person, it is then highlighted and the affected person fact is then displayed next to the affected person. If the affected person is then de-selected, the highlight is removed and the affected person fact is grayed out but still visible to the user. Data collected here includes which affected persons were selected, the time to select them, and the total time the user spent on the screen.

Begin - The Begin Fragment is responsible for displaying the Begin screen to the user. This screens only function is to provide a stop-point between the Login screen and the first screen the user interacts with as a part of the application. The user is prompted to press the next button to start the application. There is no functionality within this screen.

Best -The Best Fragment is responsible for displaying the Best Options screen to the user. The best options screen consists of 2 options the user selected on the previous screen. When the user selects an option, it is then highlighted. If the other options are then selected afterwards, that option is highlighted and the previous highlight is removed. Data collected here includes which option was selected, the time to select it, and the total time the user spent on the screen.

ErrorFragment - The ErrorFragment DialogFragment is responsible for displaying error messages to the user. Once the error message is displayed, the user may dismiss the fragment by pressing the OK button or simply pressing anywhere on the screen other than the DialogFragment itself. Custom error messages are supported. Simply pass the desired string as an argument in the constructor of this class.

ExitUtility - The ExitUtility DialogFragment allows the administrative user to exit back to the beginning of the application without having to go through the rest of the application in its entirety. Pressing and holding the StaySafe logo in the upper-left corner for 5 seconds brings up the ExitUtility. The administrative user then types in the password, originally 1873, and is brought back to the Login screen. It should be noted that all previous data collected is deleted after the administrative user press the Restart button on the DialogFragment. The user may dismiss the fragment by pressing the Cancel button or simply pressing anywhere on the screen other than the DialogFragment itself.

Facts - The Facts Fragment is responsible for displaying the Facts screen to the user. The facts screen consists of two pieces of dynamic text, a dynamic video and a dynamic picture. The user has the option to watch the video but is not required. The next button is always visible as there is no required steps before the user can move on. Data collected here includes the time to press the play button for the video and the total time the user spent on the screen.

Game - The Game Fragment is responsible for displaying the game screen to the user. A game consists of 7 questions with 2 answers each. This Fragment reads in the game question and their answers dynamically from the reader passed to the class in its constructor. The answers are then randomized. To begin the game screen, the user must press the go button. When pressed the running man moves to the 1st question. As the man arrives at the question location, the question and answers become visible. The user must then select an answer. Appropriate user feedback is shown to the user indicating if they made the right decision. After the feedback has been given, the running man then moves to the next question. The process is repeated until question 7. When the running man moves after answering the 7th question, he runs to the finish line and the final animation occurs. Once the ending animation is complete, the next button is presented to the user to move to the next screen in the current cycle. Data collected here includes all answers chosen, how many correct answers were chosen, how long it took the user from the beginning of the screen to make an answer selection for all 7 questions, and the total time the user spent on the screen.

Imagine - The Imagine Fragment is responsible for displaying the Imagine screen to the user. The imagine screen consists of one of the options the user selected on the previous screens and three done buttons. The user is prompted to think about the option and then press the done button, moving them onto the next part of the exercise. If a done button is not pressed after a period of time, the user is prompted to press the button and continue on with the exercise. Data collected here includes the time to press each done button and the total time the user spent on the screen.

Login - The Login Activity is responsible for displaying the login screen for the administrative user to enter in probationer data into the application. Data collected here includes Site ID, Client ID, Session Number, RA ID, and Date. The application then verifies that both client ids match. Once all the fields have been entered and the administrative user presses the next button, this activity then loads a new activity corresponding to the session number entered in the Session Number field. (Either Template1, Template2, or Template3)

Menu - The Menu Fragment is responsible for displaying the possible problem choices to the user. When the user selects a problem theme area (People, Places, or Things) the problems included in that theme are shown to the user. The user then touches a problem choice to continue with the application. Data collected here includes which Theme and Problem were selected as well the users time to do so. The last timing element is the total time the user spent on the Menu screen

Menu3 - The Menu3 Fragment is responsible for communicating with the Template3 class which Template 3 screen should be displayed to the user. Additionally, the Menu3 Fragment displays the navigation page to the user that allows the users to decide what Template 3 Activity is loaded next. When the user selects a possible Template 3 screen (Video, Facts, or Game) the Template3 class then loads the new screen. On subsequent visits, this screen has the ability to remove functionality to the buttons presented to the user. This removing of functionality guides the user to different Template 3 screens that they have not already completed within the current session. This screen also has the capability of changing its base appearance by loading in different layouts while keeping the base logic the same.

OnSSTimerListener - An interface designed to facilitate the interactions of the SSTimer class (interface user) and the classes that instantiated it. This interface is only used to allow code snippets to be run after a certain time frame has elapsed.

Options - The Options Fragment is responsible for displaying the options screen to the user. The options screen consists of 4 options and 4 corresponding facts. When the user selects an option, it is then highlighted and the option fact is then displayed underneath the option. If the option is then de-selected, the highlight is removed and the option fact is grayed out but still visible to the user. The user must select two options before proceeding. If the user selects another option after selecting two, the first option they selected is de-selected. Data collected here includes which options were selected, the time to select them, and the total time the user spent on the screen.

RAInstruction - The RAInstructions Fragment is responsible for signaling to the administrative user that a session has been completed and instruct them on how to transfer research data off the tablet. This screen has no functionality.

Rate - The Rate Fragment is responsible for displaying the rate screen to the user. The rate screen consists of the two options selected on the previous screen as well as check boxes under each option. When the user selects an option after the question has appeared, the checkbox corresponding to that question number is set to visible. The user must select an option for every question before continuing. Data collected here includes which options were selected for each question, the total number selected for each question, the time to select each option for each question, and the total time the user spent on the screen.

RateVideo - The RateVideo Fragment is responsible for displaying the rate video screen to the user. The user is prompted to choose whether or not the choice made in the previous Video was good or bad. Once a choice has been made, some possible consequences of this choice are displayed to the user. After consequences have been shown for a pre-determined amount, the next button is displayed to the user indicating all required aspects of this screen have been completed. Data collected here includes if the user said the choice was good or bad, the time to do so, and the total time spent of this screen.

Restate - The Restate Fragment is responsible for displaying the problem selected by the user. There is no functionality on this screen. Data collected here includes the time the user spent on the screen.

Review - The Review Fragment is responsible for displaying the WORK-IT review to the user. The animations begin at the start of the screen and "slide" down into their position. Once one animation has completed the next animation commences. The W has a 3 second delay while the rest of WORK-IT has a two second delay. Data collected here includes the time the user spent on the screen.

SSCallback - An interface designed to facilitate the interactions of the StaySafe Fragments and their current Template's Activity. This interface allows Fragments to display errors, control the next buttons availability to the user, save data to be output, animate the "floating hand", the ability for the application to be restarted and finally to output the accumulated user data. Additionally, this interface allows the Menu3 Fragment to signal to Template3 what screen should be loaded in next, to show the routing buttons to the user associated with Template 3, and cancel any routing previously instantiated within Template 3.

SSJsonReader - The SSJsonReader is the class that the StaySafe application will use to read in the dynamic text for the curriculum. This class also functions to retain pertinent information that can be used through the session but does not necessarily require file I/O. This class should only be instantiated once for a given session. The instance of the class should be passed from screen to screen. This will ensure no data loss or repetitive calls thus resulting in minimal I/O for the application. The JSON file that this class reads from should be placed in the root directory of the tablet and must be titled StaySafe.json

SSTimer - The SSTimer class functions as a utility to the StaySafe application to facilitate all time data. Furthermore, this class uses the Android CountdownTimer to enable the StaySafe application to perform actions after a certain amount of time has elapsed.

Stop - The Stop Fragment is responsible for signaling to the user that the session has completed. The administrative user is then prompted to enter in a pass code to progress the application past this screen. If an incorrect pass code was entered, it is signaled to the user and the screen does not progress.

Survey - The Survey Fragment is responsible for displaying the dynamic survey questions to the user. This screen loads the questions related to the problem selected on the Menu screen. When the user makes a selection/response, the question the response was made from becomes grayed out and the response made turns green indicating selection and the remaining responses are grayed out. The user has the options to select different responses for the same question by replacing the previous selection with a newer one by pressing a different response than currently selected. Data collected here includes which responses were selected, the timing to select it, and the time the user spent on the screen.

Talk - The Talk Fragment is responsible for displaying the "persons the user can talk to", to the user. When the user selects a person, it is then highlighted. If the person is then de-selected, the highlight is removed. Data collected here includes which person was selected, the timing to select it, and the time the user spent on the screen.

Template1 - The activity is an extension of the Template2 activity. This class loads the base layout for Template 1 and then proceeds to load in fragments for the rest of the application. Furthermore, this class performs high level actions such as restarting the StaySafe application, showing errors to the user, allowing the administrative user to exit the application early, saving data and then outputting that data. This application also creates an instance of the SSJsonReader class and the proceeds to pass that instance throughout the run of the application. This class extends Template2 by allowing the use of a "floating hand". This hand allows that application to demonstrate functionality of the application by moving the hand and then simulating screen presses.

Template2 - Template 2 is the base activity for the StaySafe application. This class loads the base layout for Template 2 and then proceeds to load in fragments for the rest of the application. Furthermore, this class performs high level actions such as restarting the StaySafe application, showing errors to the user, allowing the administrative user to exit the application early, saving data and then outputting that data. This application also creates an instance of the SSJsonReader class and the proceeds to pass that instance throughout the run of the application.

Template3 - This activity is meant to break up the usual Template2 process for the user. This activity is run during the 4th, 7th and 10th run of the application. This class loads the base layout for Template 3 and then proceeds to load in fragments for the rest of the application. Furthermore, this class performs high level actions such as restarting the StaySafe application, showing errors to the user, allowing the administrative user to exit the application early, saving data and then outputting that data. This application also creates an instance of the SSJsonReader class and the proceeds to pass that instance throughout the run of the application. This class is responsible for interfacing with the Menu3 Fragment and determining what screen should be displayed next. This class saves what screens the user has taken a part of in this session.

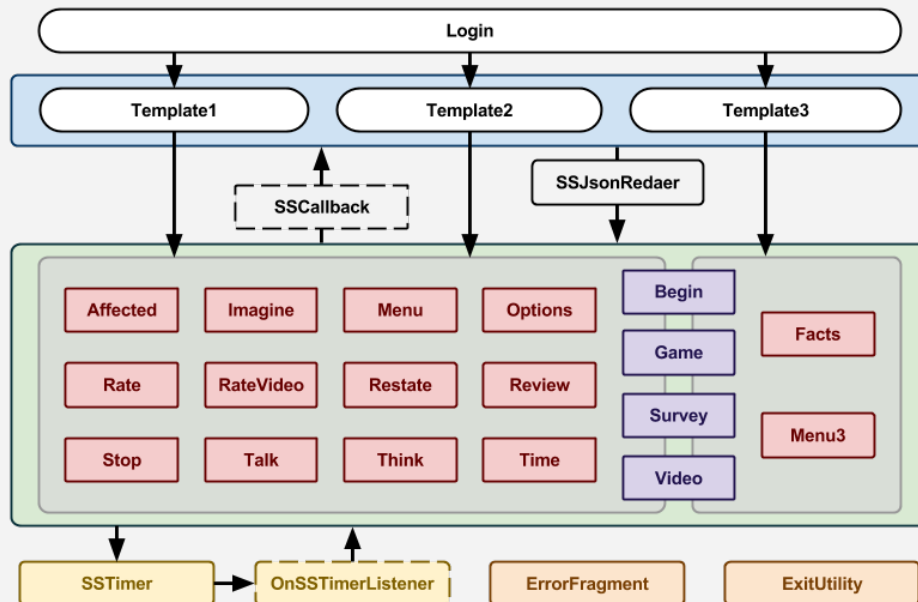
Think - The Think Fragment is responsible for displaying the think screen to the user. The think screen consists of one of the options the user selected on the previous screens and three done buttons. The user is prompted to think about the option and then press the done button, moving them onto the next part of the exercise. If a done button is not pressed after a period of time, the user is prompted to press the button and continue on with the exercise. Data collected here includes the time to press each done button and the total time the user spent on the screen.

Time - The Time Fragment is responsible for displaying the time screen to the user. The time screen consists of the spinning animation at the top of the screen. Data collected here is the total time the user spent on the screen.

Video - The Video Fragment is responsible for displaying a video to the user. The video being displayed depends on the template and selections made by the user. Generally, this method waits on the voice over to finish and then allows the user to press the play button and begin the video. For Template3 the user can press the play button instantly. Data collected here includes when the play and next buttons were pressed.

4.2 StaySafe Class Hierarchy

The diagram below depicts the hierarchical structure used by the StaySafe application. The Login class launches one of the three template Activities. Each of the template activities, surrounded in blue, then proceeds to interchange Fragments as desired. Fragments are within the green box. Template1 and Template2 pull fragments from the same subset of total Fragments. Template3 pull fragments from the rest of the set. The Begin, Game, Survey, and Video Fragments are used by all activities. Each template Activity passes an instance of the SSJsonReader class to all of the Fragments called by the Activity. Every Fragment can communicate upwards to its parent Activity through the SSCallback interface. All Fragments can create instances of the SSTimer class provided they implement the OnSSTimerListener interface. This allows the Fragments to implement timers as needed. Finally the ErrorFragment and ExitUtility are DialogFragments and can be called by any Activity.



4.3 KeepSafe Code Structure

KSError.java – Used for displaying error messages to the user

KSEException.java – Exception handling for try catch

KeepSafe.java – KeepSafe Package

MainFrame.java – Displays the startup frame for initial user interaction. Calls the other java classes for each of the templates.

SaveJSON.java – Executes all save functions throughout the application

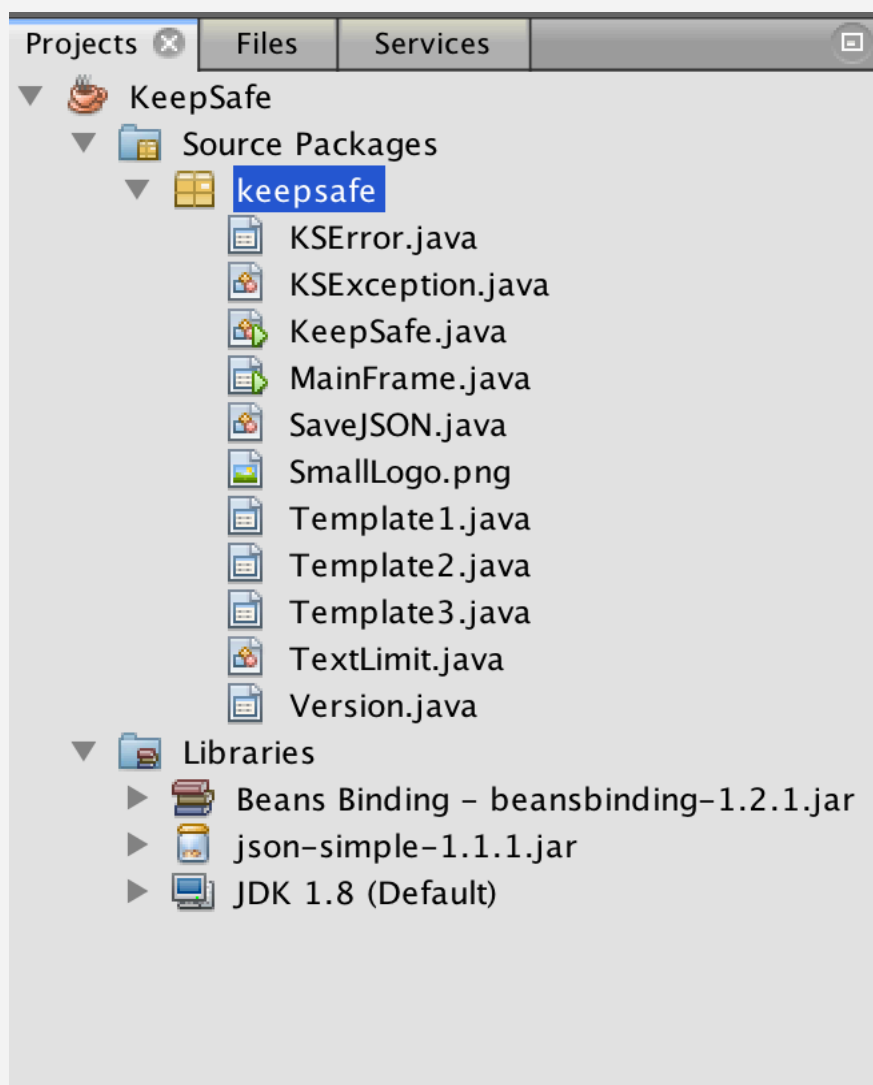
Template1.java – Template 1 text fields, formats, interactions

Template2.java – Template 2 text fields, formats, interactions

Template3.java – Template 3 text fields, formats, interactions

TextLimit.java – Sets the text box limits to the specifications

Version.java – Version control window



5 Data Export Structure

5.1 Output Data Variables

Templates 1 and 2

Variable Names		
Site	SE_1	TI_12_Time1
Clientid	SE_2	TI_12_Time2
Site_Clientid	SE_3	TI_12_Time3
RAID	SE_4	TI_12_Next
Session	SE_5	TI_13_Choice
MM	SE_6	TI_13_Next
DD	SE_7	TI_14_Done1
YYYY	TI_1_Video	TI_14_Done2
SES_Date	TI_1_Next	TI_14_Done3
Version	TI_2_Theme	TI_14_Next
CH_2_Theme	TI_2_Problem	TI_15_Next
CH_2_Problem	TI_2_Next	TI_16_Time1
CH_4_Choice	TI_3_Video	TI_16_Time2
CH_7_Choice1	TI_3_Next	TI_16_Time3
CH_7_Choice2	TI_4_Choice	TI_16_Time4
CH_7_Choice3	TI_4_Next	TI_16_Time5
CH_7_Choice4	TI_5_Next	TI_16_Time6
CH_8_Choice	TI_6_Next	TI_16_Time7
CH_9_Choice1	TI_7_CHOICE1	TI_16_Next
CH_9_Choice2	TI_7_CHOICE2	TI_18_Next
CH_9_Choice3	TI_7_CHOICE3	TI_17_Time1
CH_9_Choice4	TI_7_CHOICE4	TI_17_Time2
CH_12_Choice1	TI_7_Next	TI_17_Time3
CH_12_Choice2	TI_8_Choice	TI_17_Time4
CH_12_Choice3	TI_8_Next	TI_17_Time5
CH_12_NumberA	TI_9_CHOICE1	TI_17_Time6
CH_12_NumberB	TI_9_CHOICE2	TI_17_Time7
CH_13_Choice	TI_9_Next	TI_17_Next
CH_16_Choice1	TI_10_Done1	
CH_16_Choice2	TI_10_Done2	
CH_16_Choice3	TI_10_Done3	
CH_16_Choice4	TI_10_Next	
CH_16_Choice5	TI_11_Done1	
CH_16_Choice6	TI_11_Done2	
CH_16_Choice7	TI_11_Done3	
CH_16_Total	TI_11_Next	

Template 3

Variable Names	
Site	SE_1
Clientid	SE_2
Site_Clientid	SE_3
RAID	SE_4
Session	SE_5
MM	SE_6
DD	SE_7
YYYY	TI_4_choice
SES_Date	TI_4_next
Version	TI_16_Time1
CH_3_Facts	TI_16_Time2
CH_3_Game	TI_16_Time3
CH_3_Video	TI_16_Time4
CH_16_Choice1	TI_16_Time5
CH_16_Choice2	TI_16_Time6
CH_16_Choice3	TI_16_Time7
CH_16_Choice4	TI_16_Next
CH_16_Choice5	TI_17_Time1
CH_16_Choice6	TI_17_Time2
CH_16_Choice7	TI_17_Time3
CH_16_Total	TI_17_Time4
	TI_17_Time5
	TI_17_Time6
	TI_17_Time7
	TI_17_Next
	TI_19_next
	TI_19_video

CH – The choice for the screen
 # – The corresponding screen number
 TI – The timing data for a particular piece of data
 SE – Session Evaluation. Used on the survey screen

6 Glossary of Terms

CS: TCU's Computer Science department

GUI: Graphical User Interface

HBV: Hepatitis B Virus

HCV: Hepatitis C Virus

HIV: Human Immunodeficiency Virus

HTML: Hypertext Markup Language

IBR (Institute for Behavioral Research): The Institute of Behavioral Research (IBR) was established in 1962 by Saul B. Sells to conduct research on personality structure, personnel selection, social interactions, and organizational functioning. See more at www.ibr.tcu.edu.

IDE: Integrated Development Environment

JSON: JavaScript Object Notation

PO: The individual probation officer the probationer will be meeting with.

RA: The individual Research Assistant who will be on-site administering the StaySafe therapy to the probationers.

SD: Security Digital

SDK: Software Development Kit

WiFi: Refers to the wireless communication of network data.

WORK-IT: This is an acronym that depicts the model that we use to flow through template 2.

W: What's the problem? Who will be affected by your decision? Who can help with this decision?

O: Think about your Options.

R: Rate your Options.

K: Knowing what decision to make.

I: Imagine how you will turn your choice into action.

T: Time to Test the results.