

Team: Jennifer Baumann, Louis Bouddhou, David Morris, Roberto Kingsley

Title: Font'n: Android Font Maker

Project Summary: Build an app that allows a user to draw letters and create a font from the collection of characters that they can then save and use.

Project Requirements:

Business Requirements

There are no business requirements.

User Requirements

ID	Requirement	Document	User	Priority
UR-01	Users can draw a character in a custom font.		User	Critical
UR-02	User can undo and redo steps in drawing current character.			Low
UR-03	Users can erase parts of a drawing.			Low
UR-04	User can copy parts of a character for use in a later character.			NTH
UR-05	User can restart entire letter			High
UR-06	User can create parts of kerning table			NTH

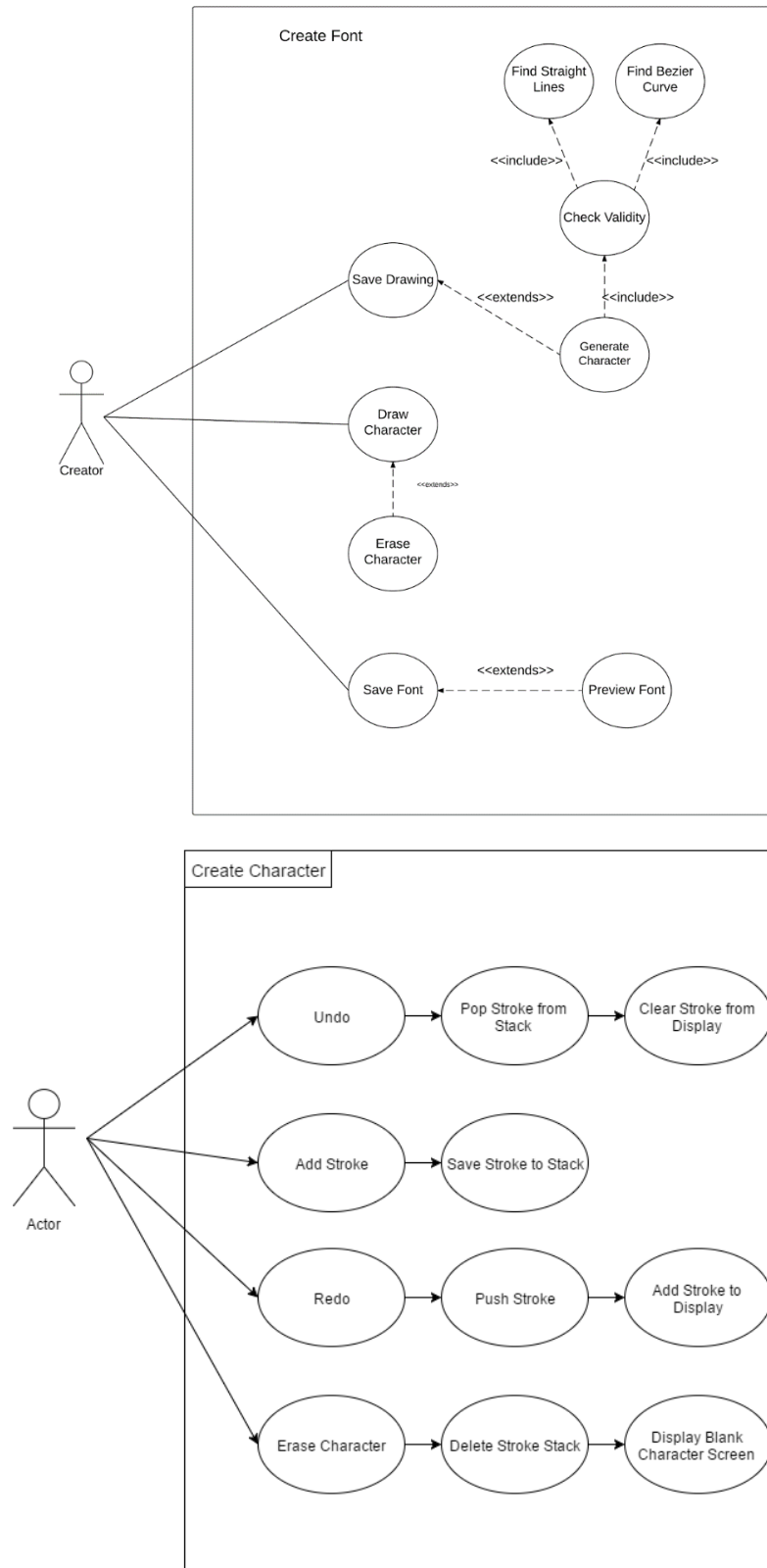
Functional Requirements

ID	Requirement	Document	User	Priority
FR-01	The app must be able to export the user-created font as a TrueType font.			High
FR-02	App must must identify bezier curves and straight lines in user-inputted font data.			High
FR-03A	Fonts should generate a kerning table for every user font.			NTH
FR-03B	Fonts should support ligatures.			NTH
FR-04A	User generated font can be viewed in a table of all characters			High
FR-04B	Character can be selected from table to edit			High

Non-Functional Requirements

ID	Requirement	Document	User	Priority
NF-01	App must be available in the google play store			Medium
NF-02	The app should be able to store 5 fonts.			Medium
NF-03	The app should not crash at a higher rate than 1 crash / 20 minutes.			NTH

Users and Tasks:



Use Case Name:	Create Character																
Use Case ID:	UR-01																
Priority:	Critical																
Short Description:	User draws characters in their font.																
Actor:	Creator																
Preconditions:	Creator Has Opened App and pressed "Create Font"																
Postconditions:	System appends new font/file to fonts array and display shadow character screen																
Main course of Action:	<table> <tr> <th></th><th>Actor Action</th><th>System Response</th></tr> <tr> <td></td><td>Creator presses "Create Font"</td><td>Appends empty font/file to fonts array</td></tr> <tr> <td></td><td>Creator adds strokes to new character</td><td>Display/Clears strokes accordingly</td></tr> <tr> <td></td><td>Creator Presses "Next"</td><td>Commit Character to font/file</td></tr> <tr> <td></td><td></td><td></td></tr> </table>			Actor Action	System Response		Creator presses "Create Font"	Appends empty font/file to fonts array		Creator adds strokes to new character	Display/Clears strokes accordingly		Creator Presses "Next"	Commit Character to font/file			
	Actor Action	System Response															
	Creator presses "Create Font"	Appends empty font/file to fonts array															
	Creator adds strokes to new character	Display/Clears strokes accordingly															
	Creator Presses "Next"	Commit Character to font/file															
Alternate Course(s) of action:	Creator chooses to "See Fonts"																
Expected State(s):	All characters of the font must be visible in the character grid																
Exceptional Course(s) of Action:	If Bezier curves cannot be identified in character drawn by the creator																

Use Case Name:	Restart Character		
Use Case ID:	UR-05		
Priority:	High		
Short Description:	Creator can restart the character at any time		
Actor:	Creator		
Preconditions:	The creator has drawn a character and pressed the Erase Character Button		
Postconditions:	Display a new character screen		
Main course of Action:			
		Actor Action	System Response
	1	Creator chooses to create a font	Displays shadow character screen
	2	Creator adds stroke(s)	Saves stroke(s) in stack
	3	Creator presses “Erase Character”	Delete stroke stack
		Display blank character screen	
Alternate Course(s) of action:			

Use Case Name:	Undo/Redo Stroke																			
Use Case ID:	UR-02																			
Priority:	Low																			
Short Description:	Creator can undo and redo strokes when creating character																			
Actor:	Creator																			
Preconditions:	The creator has drawn strokes and pressed the redo or undo button																			
Postconditions:	Display character screen with previous stroke gone																			
Main course of Action:	<table> <tr> <th></th><th>Actor Action</th><th>System Response</th></tr> <tr> <td>1</td><td>Creator chooses to create a font</td><td>Displays character screen</td></tr> <tr> <td>2</td><td>Creator adds stroke(s)</td><td>Saves stroke(s) in stack</td></tr> <tr> <td>3</td><td>Creator presses "Undo"</td><td>Pop stroke off stack</td></tr> <tr> <td>4</td><td></td><td>Clear stroke</td></tr> <tr> <td>5</td><td></td><td>Display previous strokes</td></tr> </table>			Actor Action	System Response	1	Creator chooses to create a font	Displays character screen	2	Creator adds stroke(s)	Saves stroke(s) in stack	3	Creator presses "Undo"	Pop stroke off stack	4		Clear stroke	5		Display previous strokes
	Actor Action	System Response																		
1	Creator chooses to create a font	Displays character screen																		
2	Creator adds stroke(s)	Saves stroke(s) in stack																		
3	Creator presses "Undo"	Pop stroke off stack																		
4		Clear stroke																		
5		Display previous strokes																		
Alternate Course(s) of action:	<table> <tr> <th></th><th>Actor Action</th><th>System Response</th></tr> <tr> <td>3b</td><td>Creator presses "Redo"</td><td>Push Stroke</td></tr> <tr> <td>4b</td><td></td><td>Display Stroke</td></tr> </table>			Actor Action	System Response	3b	Creator presses "Redo"	Push Stroke	4b		Display Stroke									
	Actor Action	System Response																		
3b	Creator presses "Redo"	Push Stroke																		
4b		Display Stroke																		
Note(s):	If user presses redo has never undone a stroke then system changes nothing																			

Use Case Name:	Save Font																
Use Case ID:	FR-01																
Priority:	High																
Short Description:	User saves characters to file.																
Actor:	User (saving their font)																
Preconditions:	User must have already completed drawing a font.																
Postconditions:	User must have been able to preview the font before finally saving the font to a file.																
Main course of Action:	<table> <tr> <th></th><th>Actor Action</th><th>System Response</th></tr> <tr> <td>1</td><td>Click Next Character button on last character available.</td><td>Take user to font preview screen after last character has been drawn.</td></tr> <tr> <td>2</td><td>Review all generated characters visually for quality</td><td>Display all generated characters for review</td></tr> <tr> <td>3</td><td>Click "Finalize Font"</td><td>Move to next page which shows file location</td></tr> <tr> <td>4</td><td>Review final screen with file location</td><td></td></tr> </table>			Actor Action	System Response	1	Click Next Character button on last character available.	Take user to font preview screen after last character has been drawn.	2	Review all generated characters visually for quality	Display all generated characters for review	3	Click "Finalize Font"	Move to next page which shows file location	4	Review final screen with file location	
	Actor Action	System Response															
1	Click Next Character button on last character available.	Take user to font preview screen after last character has been drawn.															
2	Review all generated characters visually for quality	Display all generated characters for review															
3	Click "Finalize Font"	Move to next page which shows file location															
4	Review final screen with file location																
Alternate Course(s) of action:	<table> <tr> <th></th><th>Actor Action</th><th>System Response</th></tr> <tr> <td>1</td><td>Click Next Character button on last character available.</td><td>Take user to font preview screen after last character has been drawn.</td></tr> <tr> <td>2</td><td>Review all generated characters visually for quality</td><td>Display all generated characters for review</td></tr> <tr> <td>3</td><td>Select a character to edit</td><td></td></tr> </table>			Actor Action	System Response	1	Click Next Character button on last character available.	Take user to font preview screen after last character has been drawn.	2	Review all generated characters visually for quality	Display all generated characters for review	3	Select a character to edit				
	Actor Action	System Response															
1	Click Next Character button on last character available.	Take user to font preview screen after last character has been drawn.															
2	Review all generated characters visually for quality	Display all generated characters for review															
3	Select a character to edit																
Expected State(s):	File saved to a location, location printed to screen for user to retrieve.																
Exceptional Course(s) of Action:	User never clicks to finalize character list.																

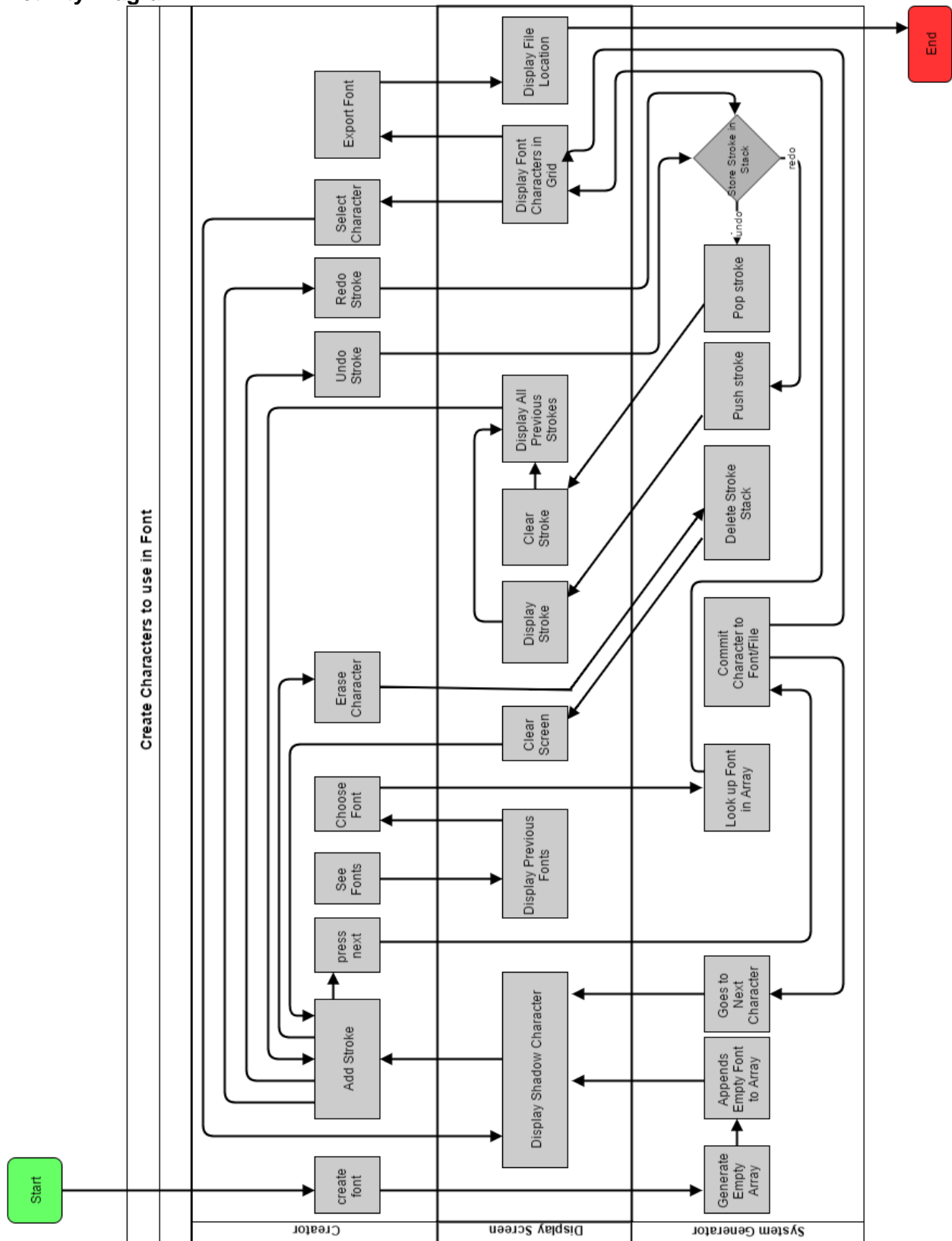
Use Case Name:	Preview Font													
Use Case ID:	FR-04a													
Priority:	High													
Short Description:	User looks over all of the generated characters.													
Actor:	User (previewing their font)													
Preconditions:	User must have completed all characters previously.													
Postconditions:	Require all characters to be saved in a format that can be ported to table view.													
Main course of Action:	<table><tr><td></td><td>Actor Action</td><td>System Response</td></tr><tr><td>1</td><td>Click next character on last character screen</td><td>Take user to preview screen</td></tr><tr><td>2</td><td>Review font on screen</td><td>Wait for next command</td></tr><tr><td>3</td><td>Click finalize font</td><td></td></tr></table>			Actor Action	System Response	1	Click next character on last character screen	Take user to preview screen	2	Review font on screen	Wait for next command	3	Click finalize font	
	Actor Action	System Response												
1	Click next character on last character screen	Take user to preview screen												
2	Review font on screen	Wait for next command												
3	Click finalize font													
Alternate Course(s) of action:	Cancel creating font before final character is completed.													
Expected State(s):	Display file location after reviewed.													
Exceptional Course(s) of Action:	Cancel creating font before final character will not show preview screen.													

Use Case Name:	Erase Part of Character														
Use Case ID:	UR-03														
Priority:	Low														
Short Description:	User erases part of drawing.														
Actor:	User (creating their font)														
Preconditions:	User clicks the erase button.														
Postconditions:	User re-clicks the erase button or clicks another button.														
Main course of Action:	<table><tr><td></td><td>Actor Action</td><td>System Response</td></tr><tr><td>1</td><td>Click erase button</td><td>Enter erase mode</td></tr><tr><td>2</td><td>Slides finger over drawing</td><td>Pixels indicated are erased</td></tr><tr><td>3</td><td>Re-clicks erase button</td><td>System enters drawing mode</td></tr></table>				Actor Action	System Response	1	Click erase button	Enter erase mode	2	Slides finger over drawing	Pixels indicated are erased	3	Re-clicks erase button	System enters drawing mode
	Actor Action	System Response													
1	Click erase button	Enter erase mode													
2	Slides finger over drawing	Pixels indicated are erased													
3	Re-clicks erase button	System enters drawing mode													
Alternate Course(s) of action:	<table><tr><td></td><td>Actor Action</td><td>System Response</td></tr><tr><td>1</td><td>Click erase button</td><td>Enter erase mode</td></tr><tr><td>2</td><td>Slides finger over drawing</td><td>Pixels indicated are erased</td></tr><tr><td>3</td><td>Clicks another icon</td><td>System enters the mode for the icon</td></tr></table>				Actor Action	System Response	1	Click erase button	Enter erase mode	2	Slides finger over drawing	Pixels indicated are erased	3	Clicks another icon	System enters the mode for the icon
	Actor Action	System Response													
1	Click erase button	Enter erase mode													
2	Slides finger over drawing	Pixels indicated are erased													
3	Clicks another icon	System enters the mode for the icon													
Expected State(s):	n/a														
Exceptional Course(s) of Action:	n/a														

Use Case Name:	Identify Bezier curves and Lines											
Use Case ID:	FR-02											
Priority:	High											
Short Description:	Scan a user-inputted character for curves and lines											
Actor:	System											
Preconditions:	User has drawn a character											
Postconditions:	We have a plot of lines and Bezier curves											
Main course of Action:	<table><tr><td></td><td>Actor Action</td><td>System Response</td></tr><tr><td>1</td><td>User saves their character</td><td>Save character, and scan for curves and lines</td></tr></table>				Actor Action	System Response	1	User saves their character	Save character, and scan for curves and lines			
	Actor Action	System Response										
1	User saves their character	Save character, and scan for curves and lines										
Alternate Course(s) of action:												
Expected State(s):	We have a saved character, and Bezier/straight line reco											
Exceptional Course(s) of Action:	<table><tr><td></td><td>Actor Action</td><td>System Response</td></tr><tr><td>1</td><td>User saves their character</td><td>Save character, and scan for curves and lines</td></tr><tr><td>2</td><td></td><td>Error encountered, offer to use default or redo character</td></tr></table>				Actor Action	System Response	1	User saves their character	Save character, and scan for curves and lines	2		Error encountered, offer to use default or redo character
	Actor Action	System Response										
1	User saves their character	Save character, and scan for curves and lines										
2		Error encountered, offer to use default or redo character										

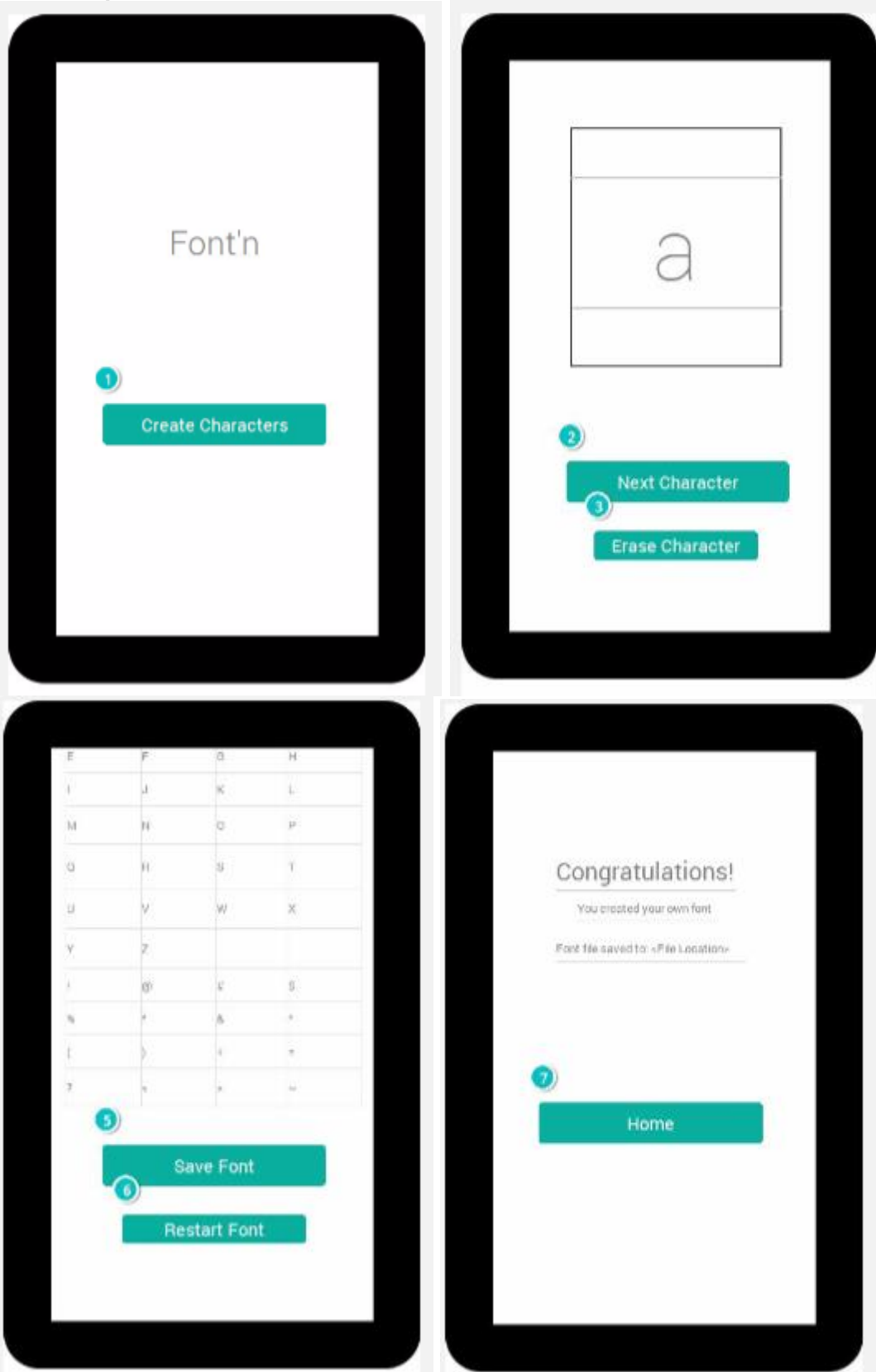
Use Case Name:	Support Storage size of minimum number of fonts created										
Use Case ID:	NF-02										
Priority:	Medium										
Short Description:	Creator exports lots of fonts										
Actor:	Creator										
Preconditions:	System successfully appends empty font/file to array										
Postconditions:	Creator is able to create and successfully export a new font										
Main course of Action:	<table><tr><th></th><th>Actor Action</th><th>System Response</th></tr><tr><td>1</td><td>Creator chooses to create font</td><td>System appends empty font to a dynamic array.</td></tr><tr><td>2</td><td>Creator only sees shadow characters if the array can be extended</td><td>System generates Bezier curves and characters for font</td></tr></table>			Actor Action	System Response	1	Creator chooses to create font	System appends empty font to a dynamic array.	2	Creator only sees shadow characters if the array can be extended	System generates Bezier curves and characters for font
	Actor Action	System Response									
1	Creator chooses to create font	System appends empty font to a dynamic array.									
2	Creator only sees shadow characters if the array can be extended	System generates Bezier curves and characters for font									
Alternate Course(s) of action:	Use previously exported fonts or delete a font to create a new one										
Expected State(s):	A font can successfully contain all characters in a grid to be exported										
Exceptional Course(s) of Action:	The font array is not large enough to contain an extra font. Creator is told that they cannot add a font.										

Activity Diagram:



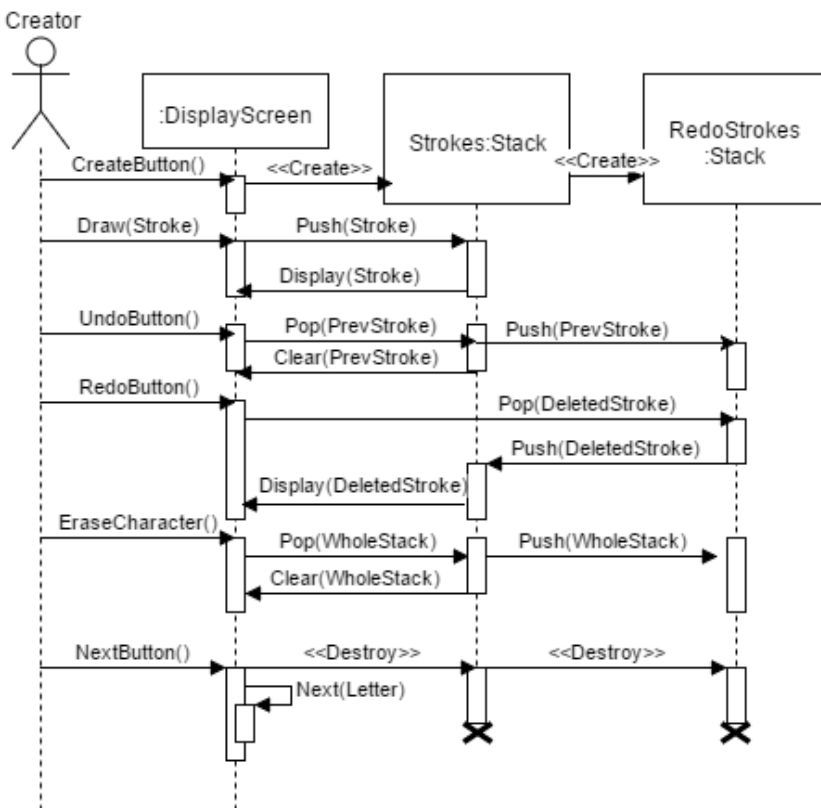
Data Storage: The raw images are going to be stored as pngs on the creator's phone. There is a possibility that the intermediary step of finding the Bezier curves and straight lines will be saved using XML and then accessed in the classes directly.

UI Mockups:

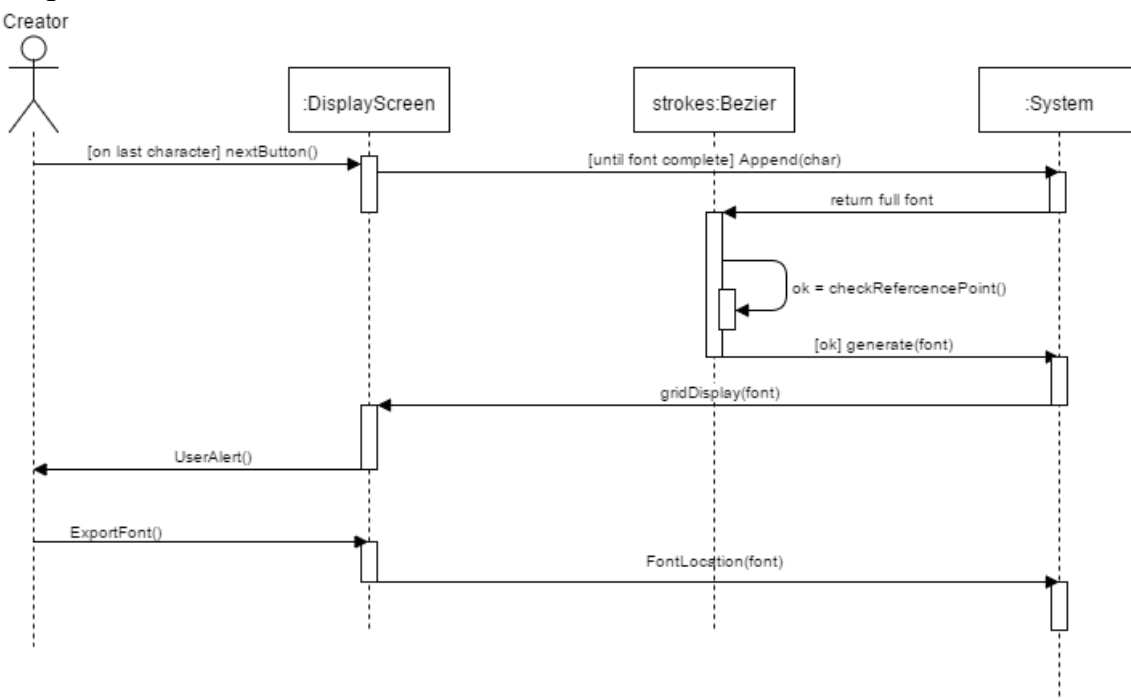


User Interactions:

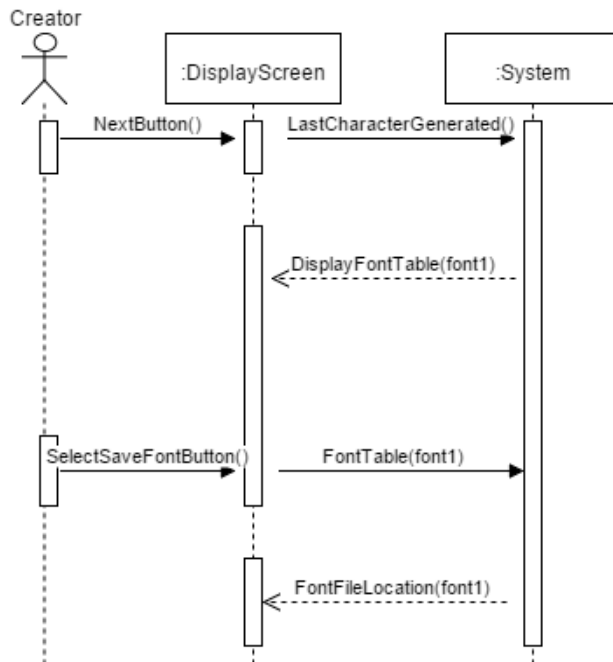
Draw Character: When drawing the character there are a few different things that the creator can do. The User can draw, erase, undo and redo parts or all of the character. To continue to the next character the user will press “Next” and the next character’s screen will display.



Generate Character: Once the user has drawn a character and is satisfied with the way it look they will go to the next character. When clicking next the system will find the Bezier curves and straight lines in the strokes and save them to be used later.



Preview/Save Font: When the user has created their font they can review the font after our program has found the Bezier curves and straight lines to make sure it looks the way they want. After reviewing they can then choose to save and export their font.



Class Diagram:

