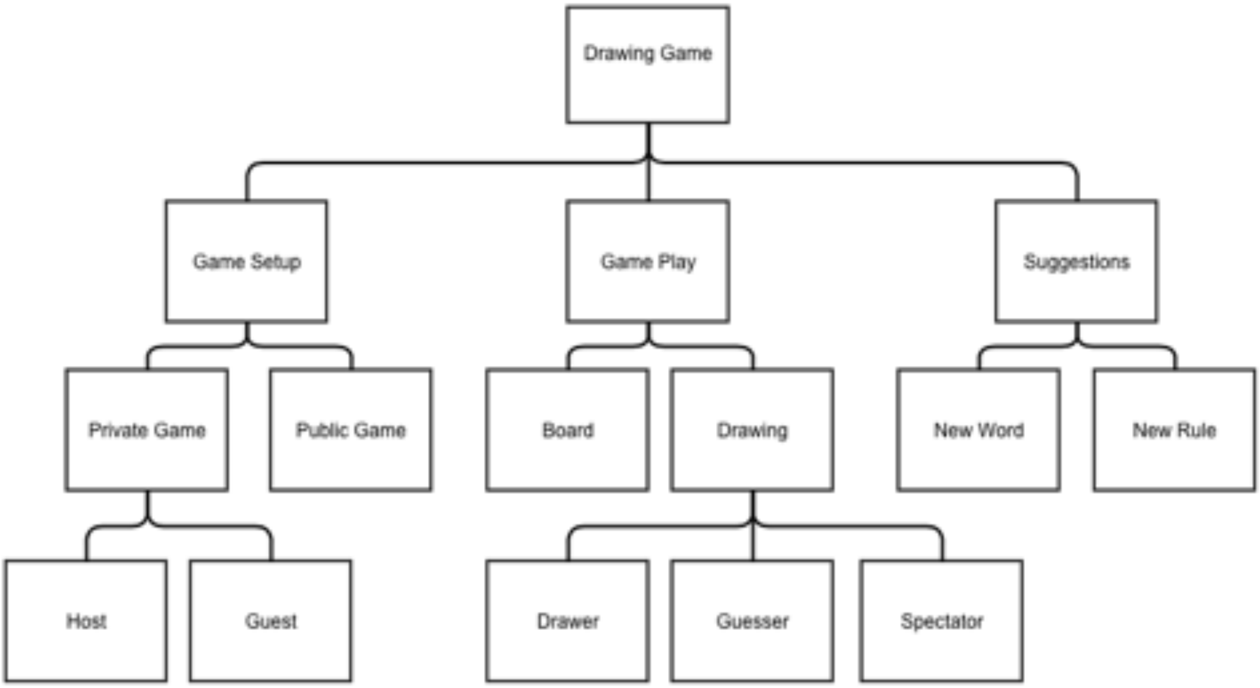
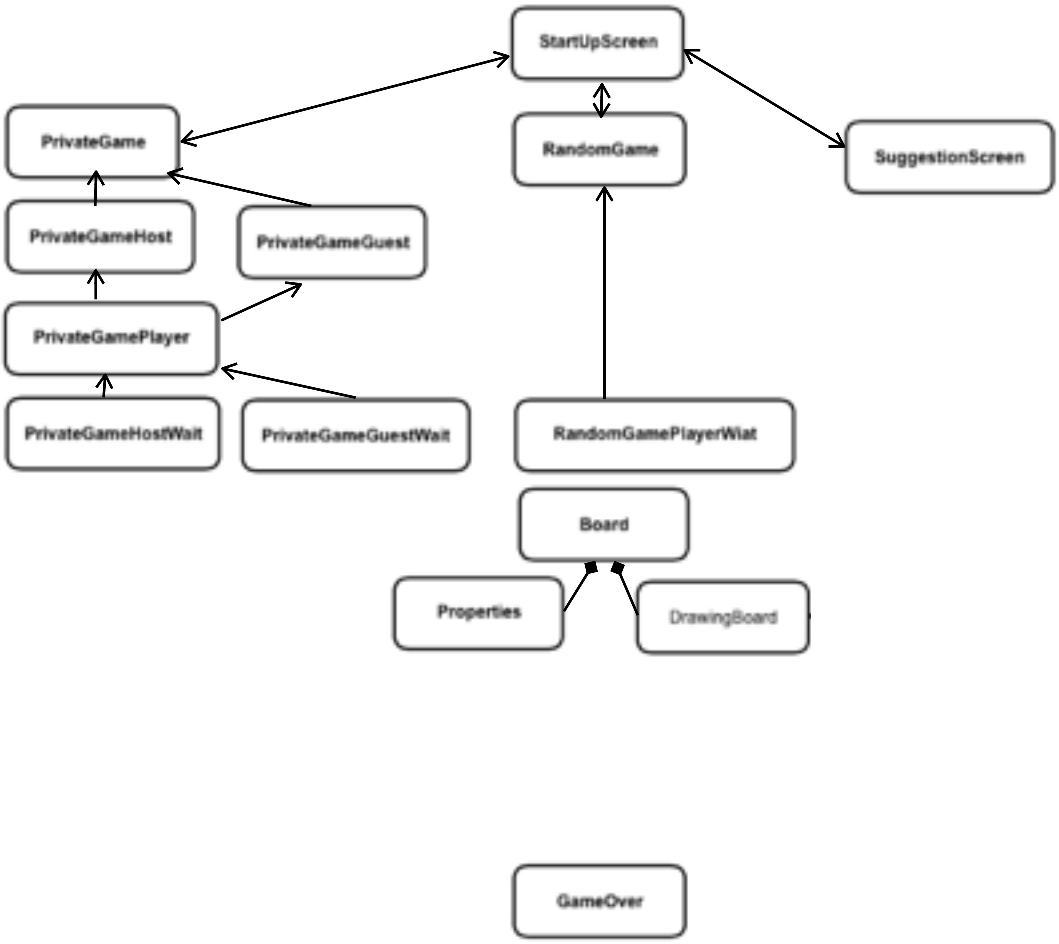
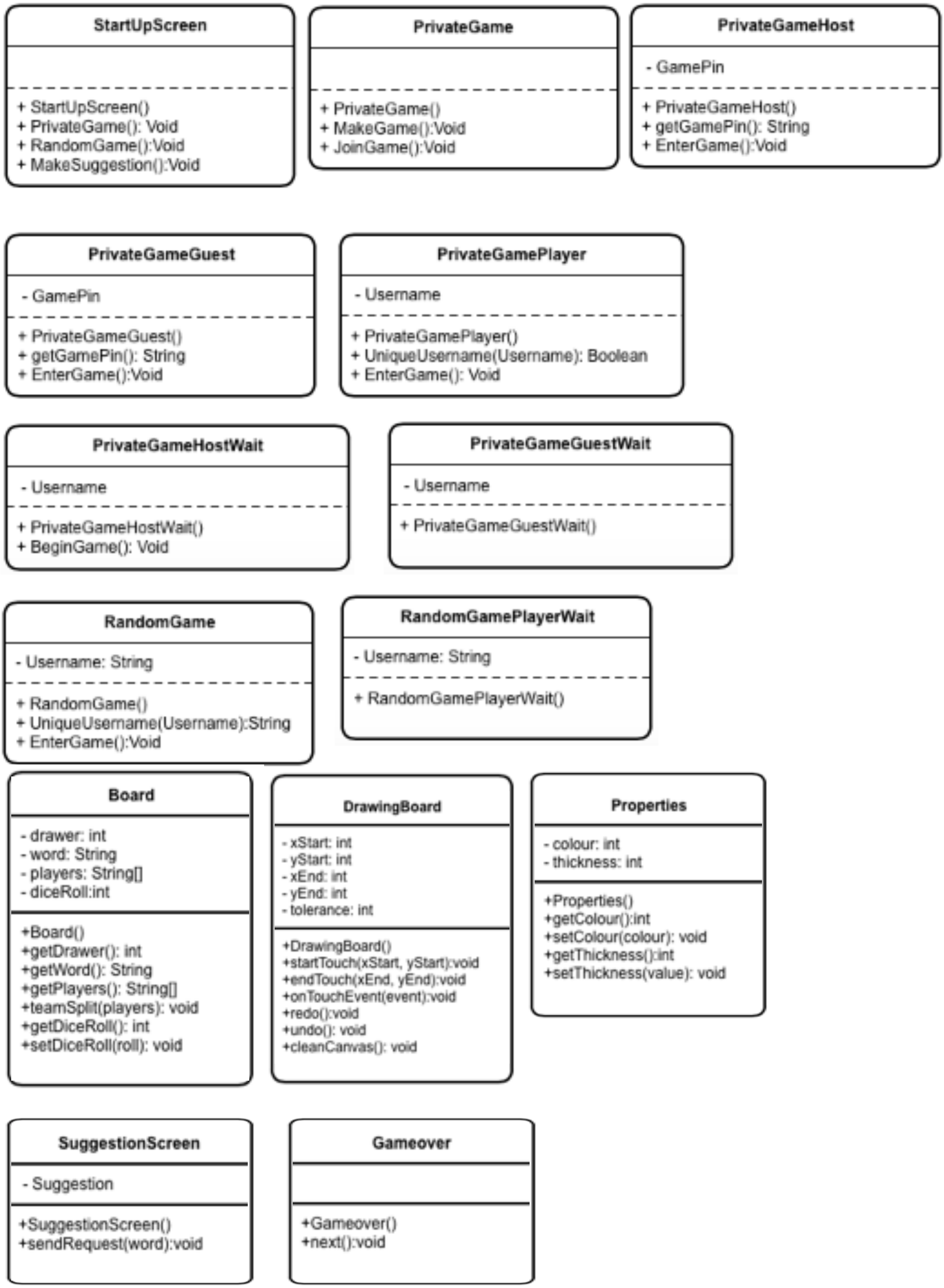
Criteria B — Design

Structure Diagram



UML Diagrams





StartUpScreen

+ StartUpScreen()
+ PrivateGame(): Void
+ RandomGame():Void
+ MakeSuggestion():Void

PrivateGame

+ PrivateGame()
+ MakeGame():Void
+ JoinGame():Void

PrivateGameHost

- GamePin

+ PrivateGameHost()
+ getGamePin(): String
+ EnterGame():Void

PrivateGameGuest

- GamePin

+ PrivateGameGuest()
+ getGamePin(): String
+ EnterGame():Void

PrivateGamePlayer

- Username

+ PrivateGamePlayer()
+ UniqueUsername(Username): Boolean
+ EnterGame(): Void

PrivateGameHostWait

- Username

+ PrivateGameHostWait()
+ BeginGame(): Void

PrivateGameGuestWait

- Username

+ PrivateGameGuestWait()

RandomGame

- Username: String

+ RandomGame()
+ UniqueUsername(Username):String
+ EnterGame():Void

RandomGamePlayerWait

- Username: String

+ RandomGamePlayerWait()

Board

- drawer: int
- word: String
- players: String[]
- diceRoll:int

+Board()
+getDrawer(): int
+getWord(): String
+getPlayers(): String[]
+teamSplit(players): void
+getDiceRoll(): int
+setDiceRoll(roll): void

DrawingBoard

- xStart: int
- yStart: int
- xEnd: int
- yEnd: int
- tolerance: int

+DrawingBoard()
+startTouch(xStart, yStart):void
+endTouch(xEnd, yEnd):void
+onTouchEvent(event):void
+redo():void
+undo(): void
+cleanCanvas(): void

Properties

- colour: int
- thickness: int

+Properties()
+getColour():int
+setColour(colour): void
+getThickness():int
+setThickness(value): void

SuggestionScreen

- Suggestion

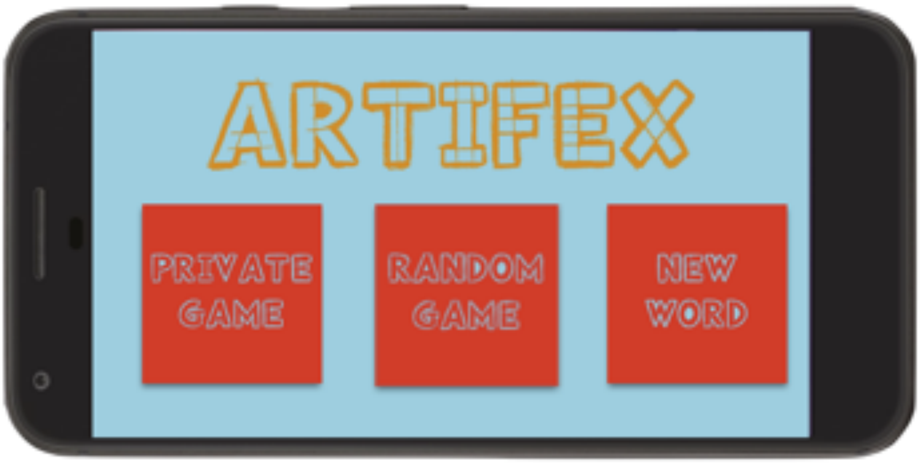
+SuggestionScreen()
+sendRequest(word):void

Gameover

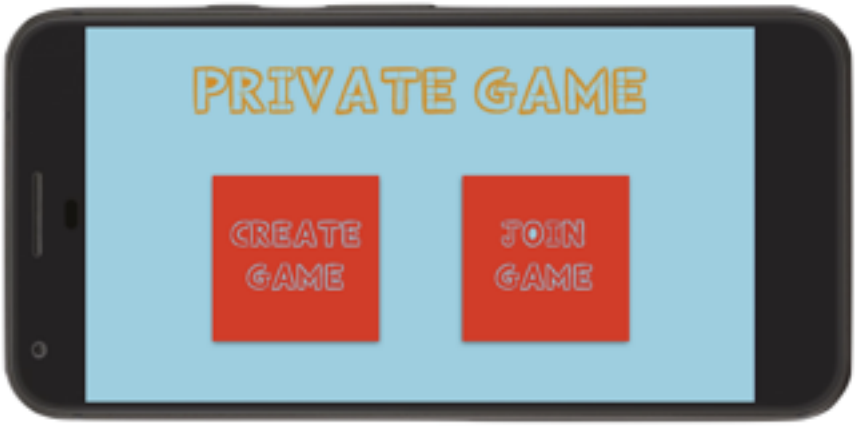
+Gameover()
+next():void

Screen Design

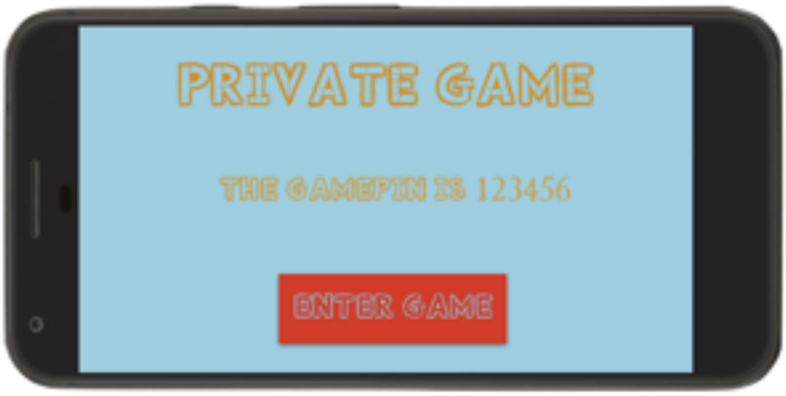
Startup screen:



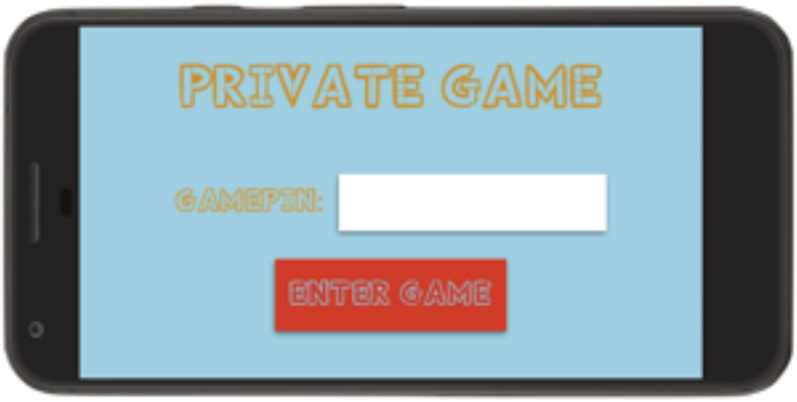
If Private game button is clicked:



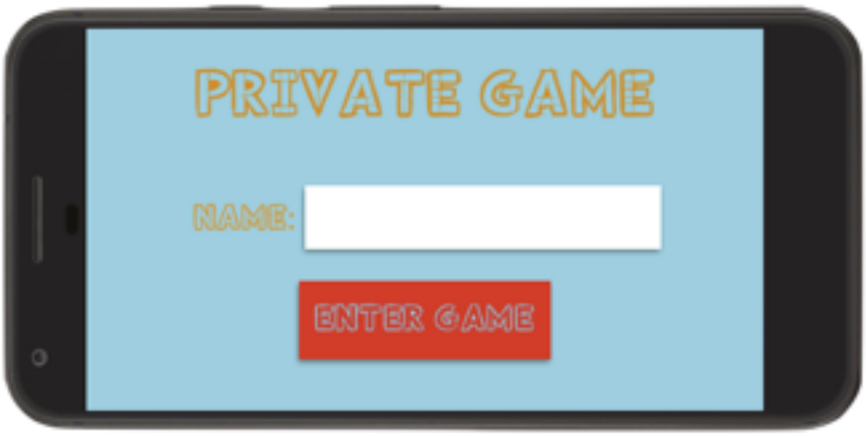
If Create Game button is clicked:



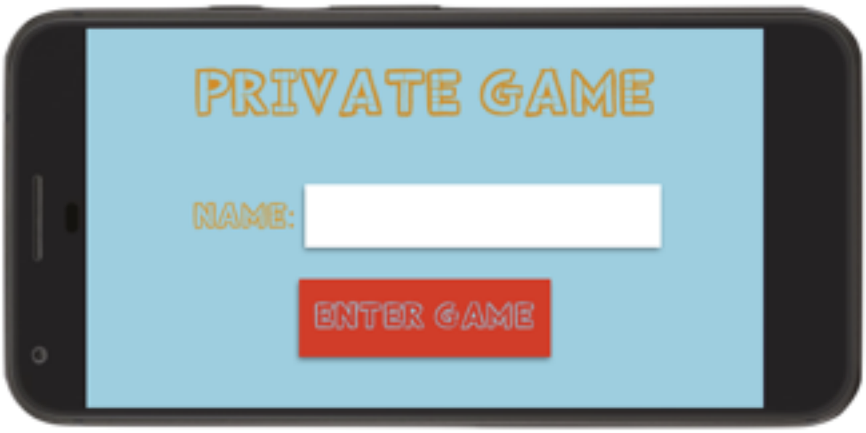
If Join Game button is clicked:



After entering game, the user has to enter their name for the game:

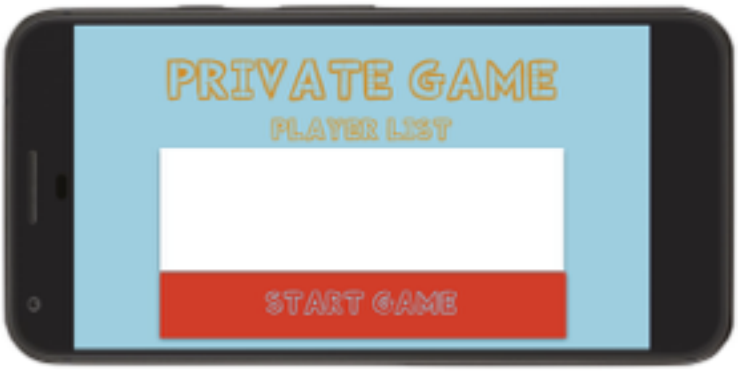


If Random Game button is clicked, they are put in a game and have to enter username:

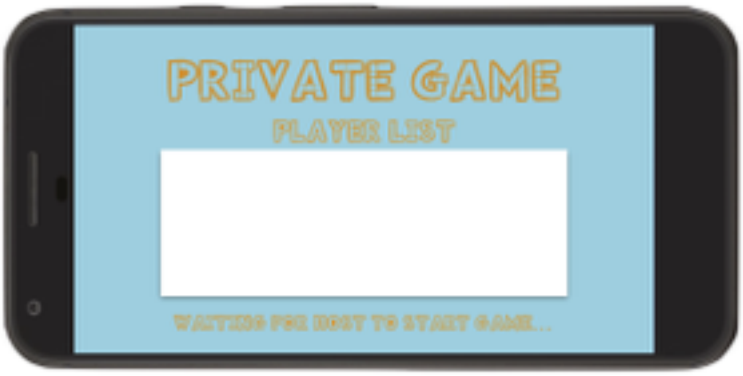


Once the player is in a Game, the player list will be visible.

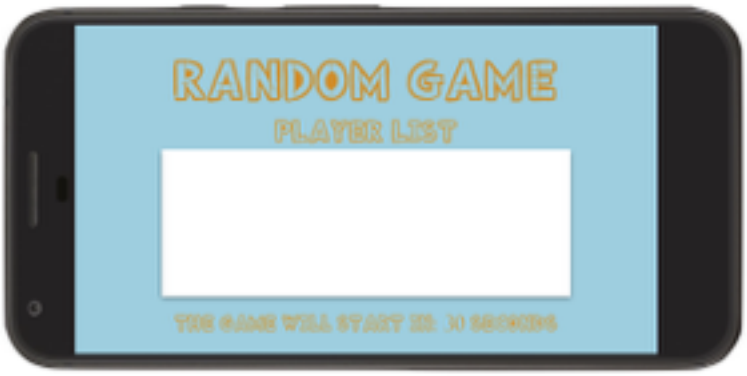
Private Game Host:



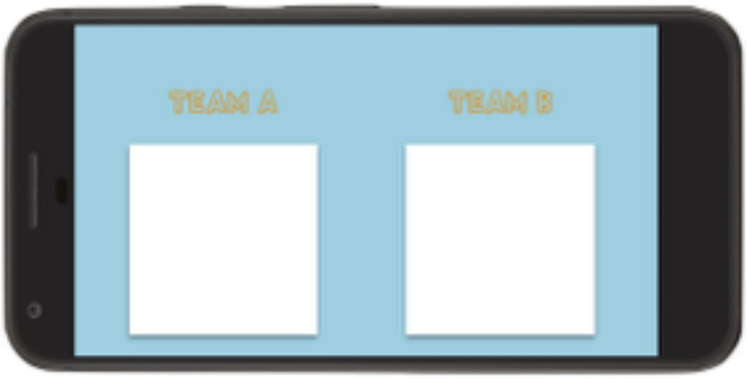
Private Game Guest:



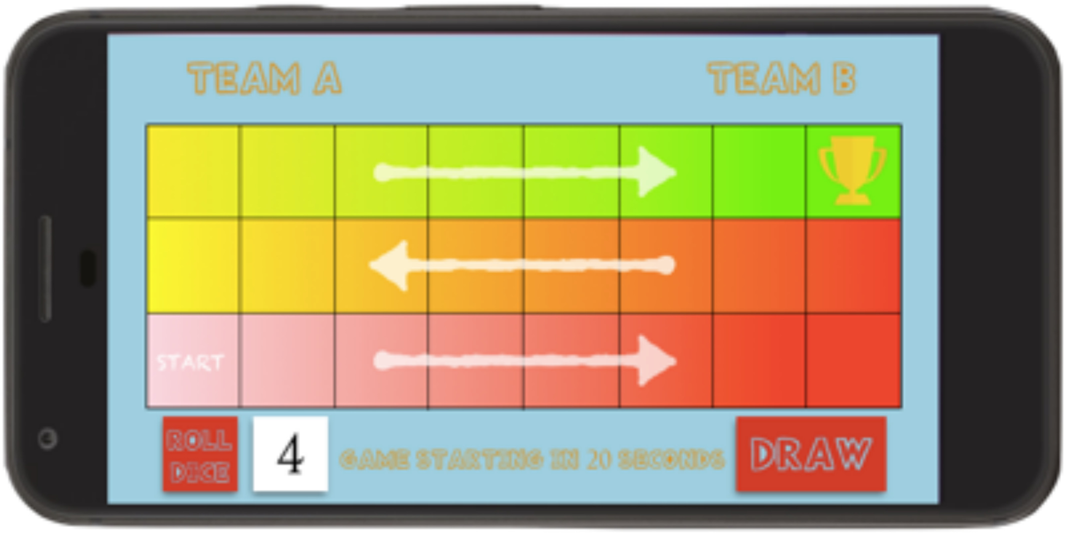
Random Game Player



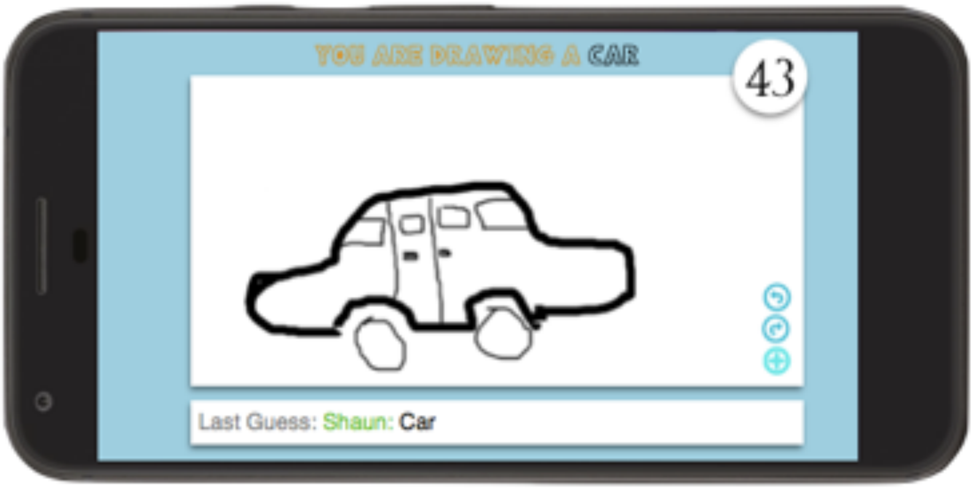
Once the Game begins the player will be split into 2 teams:



After the players are plot into team they will see the game board:

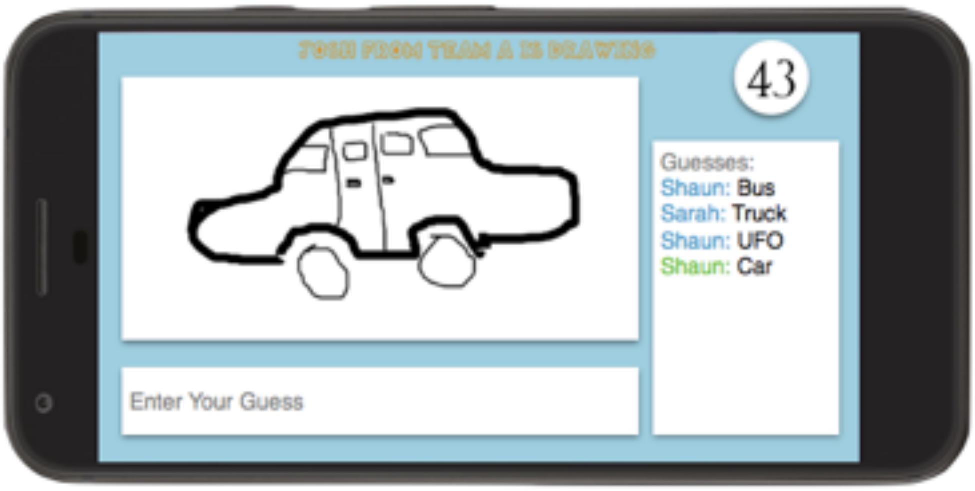


When it is time to play, the Dice has to be rolled and the Drawing has to being.
Drawer:



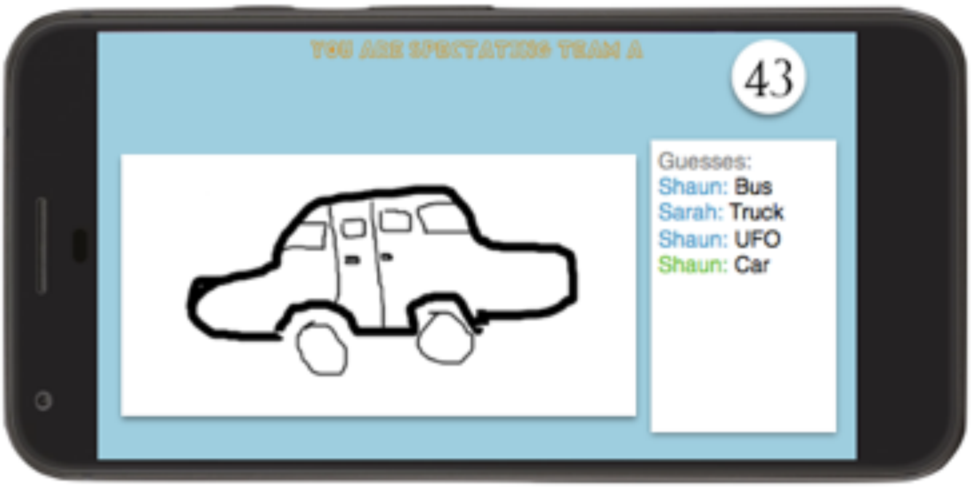
Last Guess: Shaun: Car

Guesser:



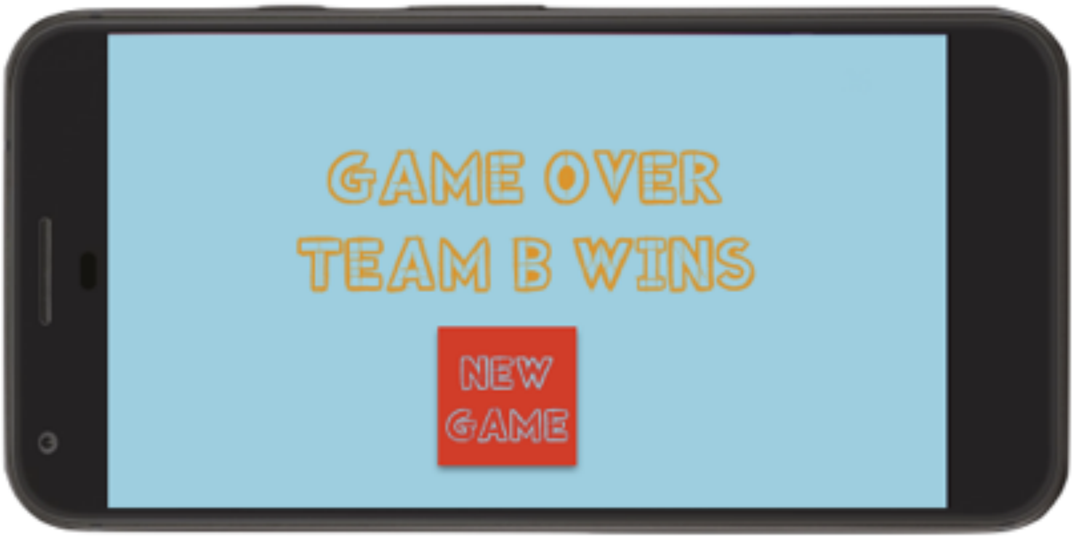
Guesses:
Shaun: Bus
Sarah: Truck
Shaun: UFO
Shaun: Car

Spectator from other team:



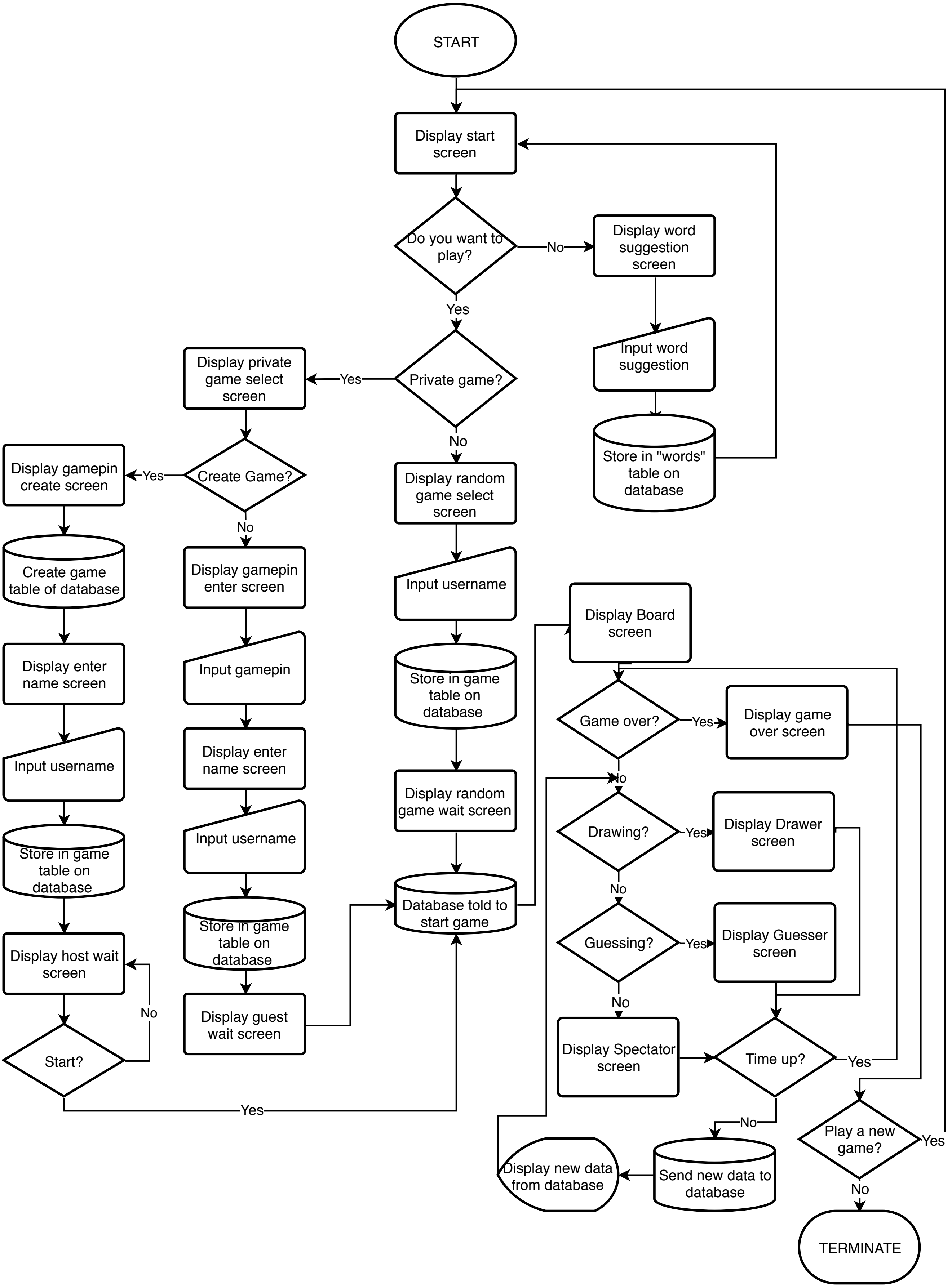
Guesses:
Shaun: Bus
Sarah: Truck
Shaun: UFO
Shaun: Car

During the game, there will be a constant switch between drawing screen and board, until one team wins:



If new game is selected, the player will go through the whole processes again.

System Flowchart


START
Display start screen
Do you want to play?
No
Display word suggestion screen
Input word suggestion
Store in "words" table on database
Yes
Private game?
Yes
Display private game select screen
Create Game?
Yes
Display gamepin create screen
Create game table of database
Display enter name screen
Input username
Store in game table on database
Display host wait screen
Start?
No
Yes
No
Display gamepin enter screen
Input gamepin
Display enter name screen
Input username
Store in game table on database
Display guest wait screen
No
Display random game select screen
Input username
Store in game table on database
Display random game wait screen
Database told to start game
Display Board screen
Game over?
Yes
Display game over screen
No
Drawing?
Yes
Display Drawer screen
No
Guessing?
Yes
Display Guesser screen
No
Display Spectator screen
Time up?
Yes
No
Send new data to database
Display new data from database
Play a new game?
Yes
No
TERMINATE

Psuedocode

Team Split

```
//function to split the teams randomly to meet criteria 6
function makeTeams(PLAYERS) //players is a list received from the database with all the players
    NUMTEAMA = floor(PLAYERS.length/2)
    USEDNUMS = integer array[NUMTEAMA] //array to store the people already in the team

    loop COUNT from 0 to NUMTEAMA //loop to select all the people in team A
        NUM=randomNumber(1,PLAYERS.length) //random number between 1 and no. players
        USED = false
        loop COUNT2 from 0 to COUNT
            if USEDNUMS[COUNT2]==NUM then
                USED = true //if the number has been used the boolean changes
            endif
        endloop
        if USED == false then
            USEDNUMS[COUNT]=NUM //add the number to used and people in team A
        else
            COUNT = COUNT — 1 //reduce count so that when increment same value again
        endif
    endloop
    return USEDNUMS
endfunction
```

Unique Username

```
//function that checks if the username is unique to meet criteria 5
function checkUniqueName(USERNAME) //USERNAME is the username being checked
    PLAYERS = database.table.getUsernames //returns array of all names in the database table
    loop COUNT from 0 to PLAYERS.length
        if PLAYERS[COUNT] == USERNAME then
            return false //because the username is in the table no need to keep checking
        endif
    endloop
    return true //since there was no match, the username must be unique
endfunction
```

Check Guess

```
//function to check if the guess made is the word
function checkGuess(GUESS) //GUESS is a string guess that has been made
    CORRECT = board.getWord() //board is the class from which the intent is launched and it holds the word
    if GUESS == CORRECT then
        return true
    else
        return false
    endif
endfunction
```

Test Plan

	Success Criteria	Test	Expected Result
1	The application must run on the Android operating device	Run Application on android device or android API	The program does not crash and works as expected
2	The application must be able to read from and write to a mySQL database so that connection between players can be made	Perform all necessary functions: - Create table in database with unique name - read data from existing table - create records in table - update records in table - delete table in records	The communication between database and program will be smooth and games will made which people can join and play in.
3	When a user enters the application they should be provided with the option to join a random game, join a private game or create their own game. Depending on the selection a different set of instructions will be followed to set up the game.	2 Tests 1. Click on private game button 2. Click on random game button	1. After the button is clicked, a new intent should be launched which has the option of creating a game or joining one 2. It should prompt the user to input their preferred username
4	The application should be able to connect between 4-8 players per game in the game type chosen	Try running the game with less than four players, between 4 and 8 players, and then with over 8 players.	The program should give errors for games outside the 4 to 8 range and launch into splitting teams if the number of players in acceptable
5	Everybody inside a game should have a "unique username" [82] so that everybody is identifiable	Try entering a unique username and then repeating that username	The program should accept the username the first time it is entered, but give an error the subsequent time.
6	Once the game begins, it must be able to split users into "2 random teams" [72]	Open the intent to split the teams	The teams will split randomly with even numbers on both teams
7	There will be 2 roles in each team for each round, drawer and guesser. The opposing team will just be spectators for the round. The drawer and guesser must change every round so that through the game all players play every role	Play a full game and keep track of which player had which role in each round	There should have been a rotation through who draws for the team when they are drawing and the team not drawing should all be spectating
8	The drawing board must show a real-time image on all devices from the device that is being drawn on	Use multiple devices and draw on one and observe on the others	The image drawn should appear on the other devices almost instantly

	Success Criteria	Test	Expected Result
9	“The pen should also have different drawing colours, opacity and thickness.” [68] These are important as it makes drawing accurate representations of the word given much easier. The colours will use RGB sliders so that there are more than 16 million different colour possibilities	Try different colours, thicknesses and opacities to ensure they do intact work. Also try using the RGB sliders to chose the drawing colour to ensure that a great range of colours is possible	Whenever any of the properties are changed, the subsequent lines drawn will be using the values that have been changed to.
10	The application must have an eraser and clear button so that a mistake in drawing can be fixed	Use the eraser button to go over something that is drawn and also use the clear button.	When the eraser is used, it should give the impression of having deleted the areas that the person went over and if the clear button is pressed, the whole canvas should become blank.
11	There must be a random number generator from 1 to 6 to mimic a dice roll so that teams can advance on the board	Press the button many times to ensure it goes from 1 to 6 and is random.	The value should be different most of the time with a probability of 1/6 for each number
12	The application must have a “timer” [68] so that the round ends after 90 seconds if the word is not guessed correctly	Run the program for 90 seconds in the drawer, guesser or spectator screen	The activity should shut and return true false for whether or not the word was guessed
13	Users should be able to suggest new words to add to the game so that there is a range of words and the game stays interesting	Use the suggest new word button to launch intent from home screen then make a word suggestion	The word should appear on the database waiting to be verified before being put in the game