CHECKERS GAME APP

by

SMITA SHARAN

B.E., C.S.V.T. UNIVERSITY, 2011

A REPORT

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2016

Approved by:

Major Professor
Dr.Daniel Andresen

Abstract

Android Phones and Tablets are driven by apps and becoming more and more popular these days. Enthusiastic game players around the world are always looking for fun and exciting games which are easily accessible from their handheld devices like smartphones, tablets etc. Checkers is one such mobile game app based on the popular game American Checkers or Draughts and is developed on Android platform. This game is turn based in which players play against each other. The player capture the pieces of opponent and progresses diagonally, trying to become the winner by capturing all the pieces of opponent or blocking the opponent so that there is no legal moves left.

There are various features in the game. Instant messaging allows players to chat during the game.. Saved Games feature lets users to play the incomplete games later. Leaderboards and Scores make the game competitive as players can see their rankings and scores after the game.

Tournaments and Scheduling enables players to schedule matches. Tournament is scheduled with Round Robin process wherein each player plays with all other players in turn and finally winner is declared after end of the game.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I express tremendous gratitude to my Major Professor **Dr. Daniel Andresen** for his wonderful support and guidance throughout this project.

I profoundly thank my committee members **Dr.Mitchell Neilsen and Dr.Torben Amtoft** for serving on my committee, and also for their valuable suggestions during my Master's Program.

I would like to thank members of the CIS department & Staff for their academic help which I received throughout my Master's program at Kansas State University

I would like to thank my brother & sister-in-law for their encouragement & finally would like to thank my parents for being my strongest strength at every stage of my life and to help me achieve my dreams.

Chapter 1 - Introduction

1.1 Project Description

The Checkers Game App is a strategy board game app which is turn based two player game. The game board has 64 squares which are alternate in color (32 black squares and 32 white squares). Each player having twelve disk shape pieces that are usually dark & light in color.Only the dark squares are used for play. The Player with black checkers moves first.Moves are allowed only on dark squared diagonally. The player captures piece of the opponent by jumping. In one jump, only one capture is allowed. Multiple jumps are allowed in a single turn only in forward direction. The captured pieces are removed from the board and the player capturing the piece gains points. The piece of the player which reaches the opponent side of the board is declared as King. The winner is decided when the player captures all the pieces of opponent or when the player blocks all the pieces of opponent and the opponent has no moves left.

The game app has several social features. The players can invite another player to play. The players receive invitations in their match inbox. The player can also interact with one another through chat feature in the game. The players can access saved games through match inbox which are incomplete matches. Leaderboards and Scoring system in the game helps the player to access game performance in comparison to other players. Skill levels are attained when players win matches or gains certain score level while playing the match. There are three kind of skill level defined in the game – Beginner, Intermediate, Expert. The players access their history of win and lost matches through the main screen menu.

In tournament, more than two players are required to play matches and they are scheduled following the Round robin format.

1.2 Game Platform

The game app is implemented on Android Platform. Android has several features that makes it a good platform to develop game apps. Android is known for running smoothly many tasks in the same time in multiple devices. Android is open source which makes it easily suitable to integerate with other APIs. For example, Google Play Services provides various features in this game and Cloud gaming engine used in the game which is a backend for mobile games is perfectly compatible with Android. Android applications has a very large user base and is easily accessible to users all over the world. This game app requires wireless connectivity to play it from anyplace any time,for this Android supports many technologies like LTE etc.

Chapter 2 - Android

2.1 Introduction

Android is an operating system for smartphones and tablets. Google developed Android and is a Linux based operating system. There are several flavors of Android which has released such as KitKat, Lollipop and Marshmallow. Android apps are written in the Java programming language.

2.2 Android Architecture

Android operating system is divided into four layers – Applications, Application Framework, Libraries, Android Runtime,Linux Kernel

2.2.1 Applications

Applications are like interface to the user and forms the top layer. This layer comprise native applications and third party apps. These applications are generally written in Java. [1].

2.2.2 Application Framework

This layer is composed of services which helps run and manage Android applications.

Following are the services provided by this layer:

2.2.2.1 Activity Manager

Activities that controls the application life cycle are managed by Activity Manager. It has several states. An application may have multiple activities, which have their own life cycles.[1]

2.2.2.2 Views

Views are used to create user interfaces for the application.[1]

2.2.2.3 Resource Managers

All resources such as string, dimensions,layouts are managed by Resource Managers [1].

2.2.2.4 Content Provider

This allows applications to share data with other applications [1].

2.2.3 Libraries

This layer is on the top of kernel having Java-based Android libraries. These libraries are Surface Manager, System C libraries, Open GL ES libraries [1].

2.2.4 Android Runtime

Android applications are compiled to Dalvik bycode & run on ART.Android Run Time executes the Dalvik Executable format & Dex bytecode specification [2]

2.2.5 Linux Kernel & HAL

Linux kernel is the bottommost layers and contains, all the essential hardware drivers like camera, keypad, display ,blutooth, WIFI etc

HAL it is the layer of abstraction between the device hardware and other layers on its top [1].

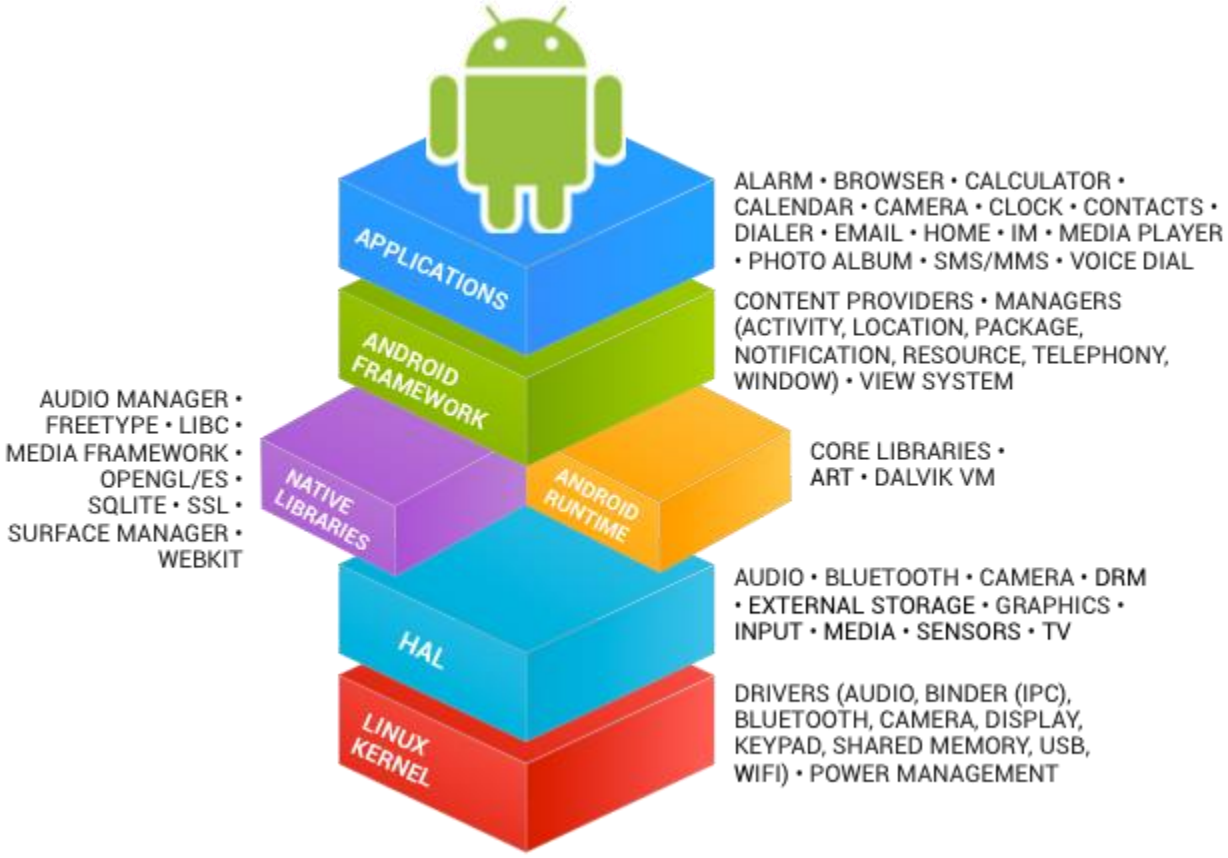

Figure 1. Android Software Stack [3]

2.3 Android App Components

According to Android Documentation, the following are the app components :

2.3.1 Intents

An intent is used for communication between components. The information that is passed with an intent is Component Name, Action,Data, Category,Extras and Flags and there are two types of intent : Explicit Intent and Implicit Intent [4].

2.3.2 Activity

An activity is a component that has its own lifecycle and provides the user interface.The Main activity is called when the application is launched and all the activities must be defined in manifest file. To create an activity, Activity class has to be inherited and in the subclass, callback methods should be implemented.For example, the lifecycle of activity may consist of created, stopped, resumed or destroyed. To set the layout for view of the activity set Content View () is used [5].

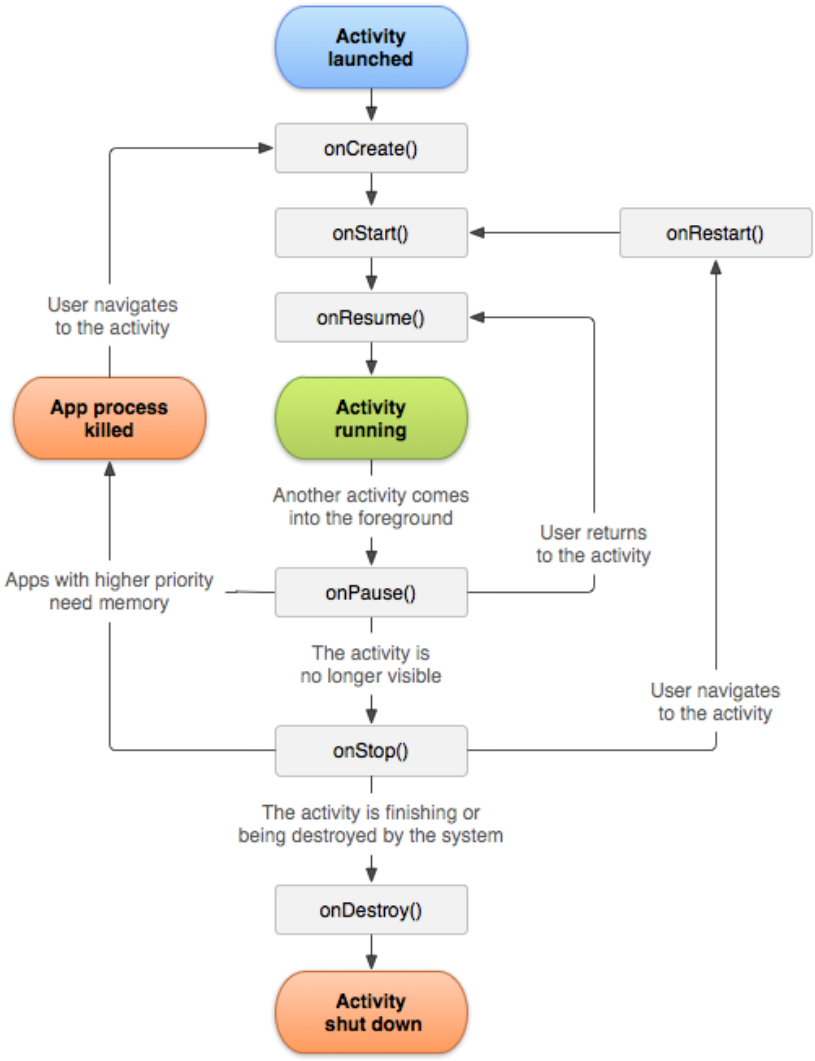

Figure 2. Activity Lifecycle [5]

2.3.3 Fragment

A Fragment is a part of an Activity. An activity can have multiple fragments. Fragments can be reused in another activities. A fragment's lifecycle and its host activity's lifecycle are related. [6].

2.3.4 App Manifest File

AndroidManifest.xml file is in every application. According to Android documentation, the functions of Android Manifest files are as below :

- It states the minimum level of API
- It states the all the components of the application – the activities, services etc.
- It also declares the permissions required by outside components to interact with the application.
- It names the package for the application which is a unique identifier for the application [7].

CHAPTER 3 – SYSTEM ANALYSIS

3.1 Existing Systems

There are various android checkers game apps available in the market which are free or priced. Most of them are two player games in which players can play a turn based game but these apps does not have one or the other feature that enables players to engage themselves in the app socially. The existing apps also doesn't have tournaments and scheduling for players through mobile.

3.2 Proposed System

The proposed system eliminates the limitations of existing systems by not only including a game between two players but can also schedule tournament for more than two players. The App also has various social features. The most important features that makes this checkers game app unique is that tournament matches can be played in Round robin format and final winner is declared at the end based on the results of the matches.

3.3 Software & Hardware Requirements

The software requirements required for developing the application are as follows:

Tools : Android Studio, Language: Java , Operating System: Windows 8 or higher.

Android Emulator: SDK Version 4.4.2 or higher with Google Play Services installed.

The hardware requirements for developing the application are as follows:

2 GB RAM & 2 GB Hard Disk Space

The hardware and software requirements for running the application Smartphone with 4.4 KitKat Android Operating System or higher and google play services.

Chapter 4 - SYSTEM DESIGN AND UML DIAGRAMS

4.1 System Design & Overview

System Design shows the interaction between different modules, architecture, components and data transfer of a system[8]. The checkers game app interacts with all the three components – Google Play Services, Send Bird and Game Backend Server

The following diagram shows the overview System Design of Checkers Game App.

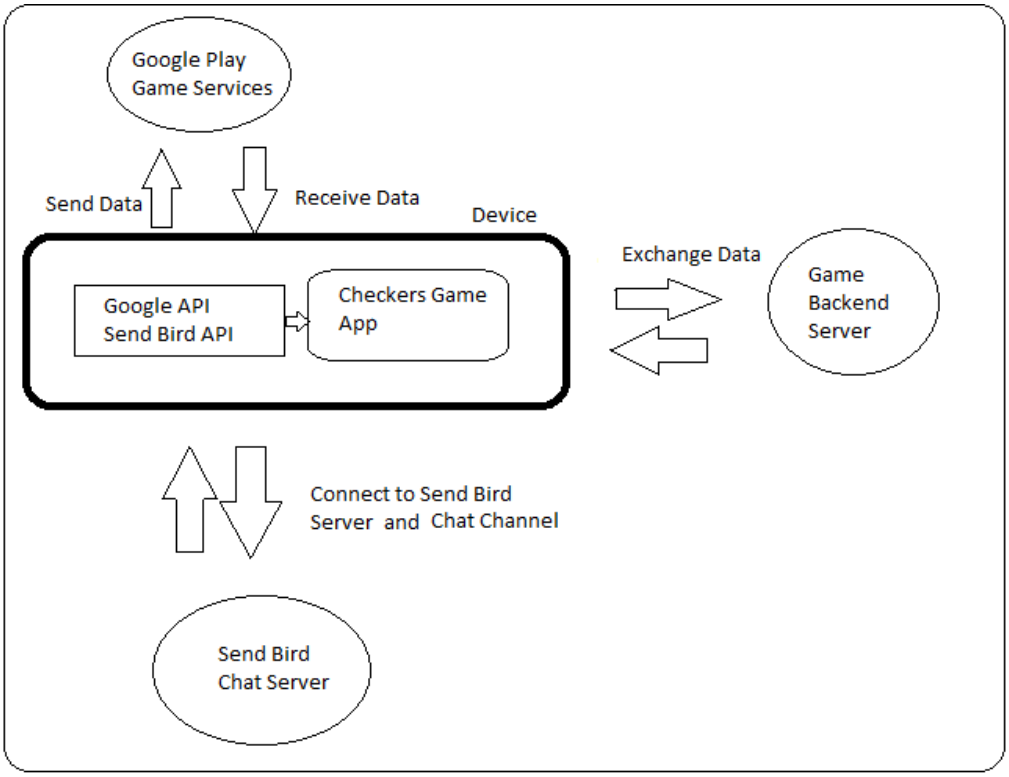

Figure 3 Overview of System Design of Checkers Game App

4.1.1Explanation of the System Design Components:

4.1.1.1 Google Play Services

Google Play Game Services provides API for developing games on Android devices. It can be used to incorporate various features like leaderboards (both public and between friends), realtime multiplayer games. [9]. Google Play services helps implement asynchronous turn based game in the checkers app. All the features like play with friend,match inbox are implemented through this service. During the match, match data is persisted in the google server and retrieved back from the google to coordinate the game.

4.1.1.2 Send Bird

Send Bird is the chat API and is the backend for apps and games. It enables messaging between users both private messaging and group messaging [10]. In this app, Send Bird helps players to chat with opponent through 1 on 1 messaging feature. All the players are connected to a common channel and data is send and received through this common channel between players.

4.1.1.3 Game Backend Server

Backend Mobile Service provides platform to Mobile App Developers a way to connect their application to backend cloud storage. Besides this it provides many other features such as Push notification, user management and integrating with various other social services network.These services are provided by through Software Development Kit and Application Programming Interface.[16]. In this game, it provides cloud storage for data, that is to be retrieved by all the players to see the tournament match data.

4.2 UML Diagrams

The Unified Modeling Language (UML) is a modeling language to provide a standard way to visualize the design of a system. There are two types of UML diagrams - structural and behavior diagrams [12].

4.2.1 Behavior diagrams

Behavior diagrams emphasize what should happen in the system or its behaviour and are used to describe the functionality of software systems. Activity Diagrams describes the step-by-step activities in a system [12].

4.2.1.1 Login Activity:

The diagram shows the Login Module Activity. After the user signs with his/her google account, the google server authenticates the sign in information (user name and password). After successful authentication the user is directed to Main Menu screen.

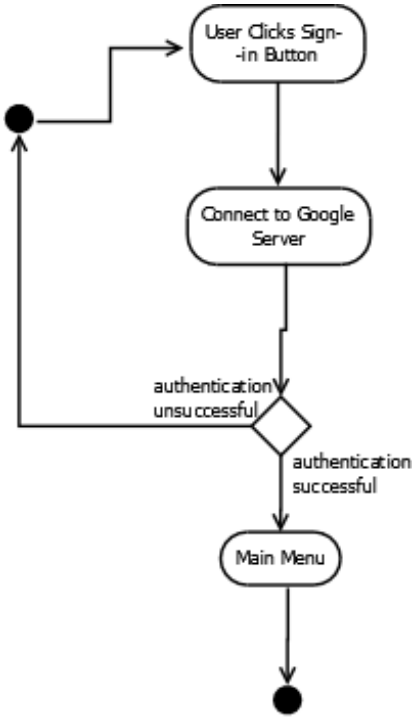

Figure 4 Login Activity

4.2.1.2 Main Menu Activity

The Main Menu Activity diagram shows that when player id directed to main menu, he/she can choose to go to start a new match or see saved games and play them or choose to see leaderboards or can start a tournament. After its completion, the player can choose to exit.

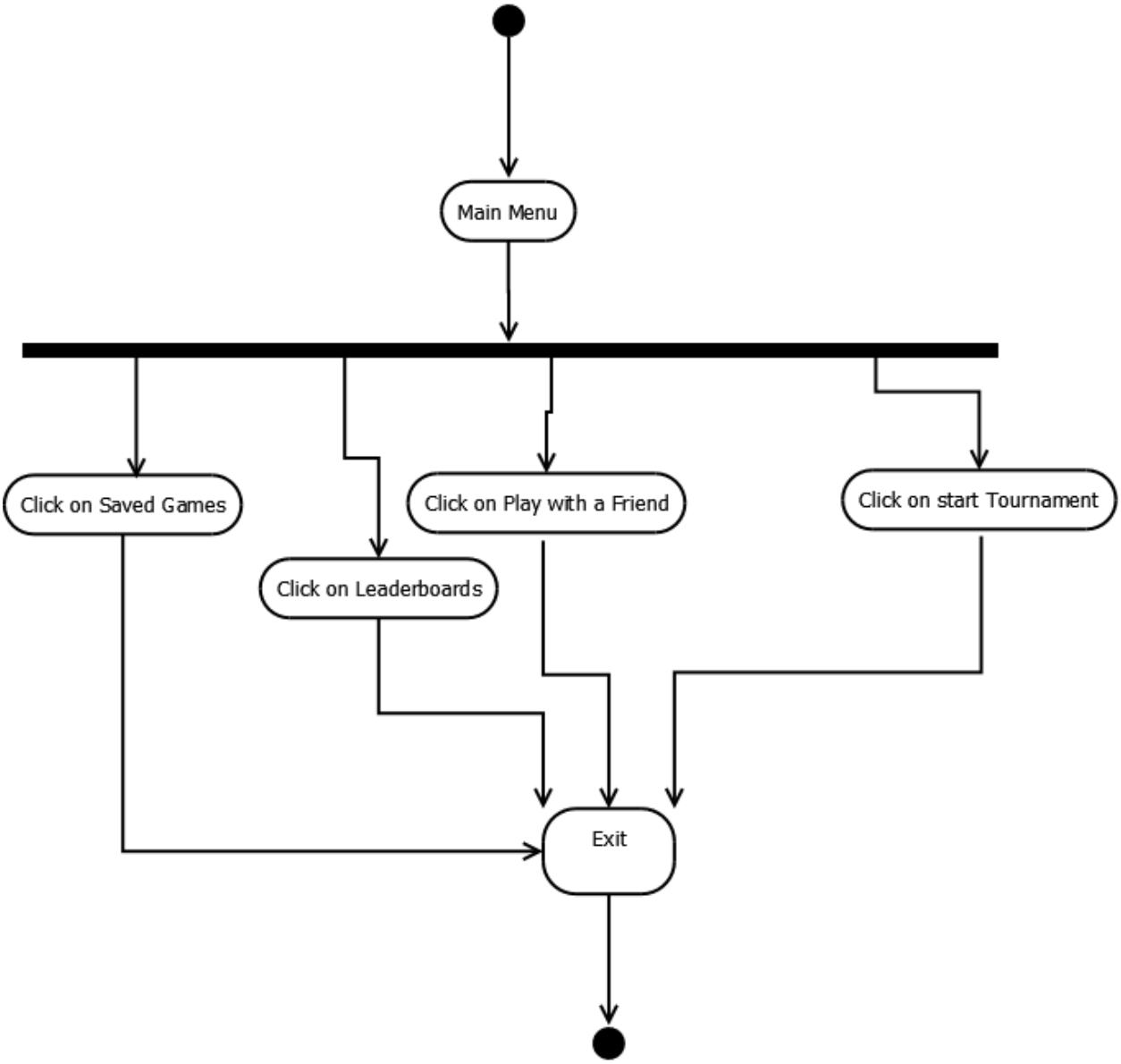

Figure 5. Main Menu Activity

4.2.1.3 Saved Games Activity

In this activity, the player can view Saved Games by clicking on the match inbox and can view whose turn it is the saved incomplete matches and play them. The data for the saved games is retrieved from google server and used to build the match again.

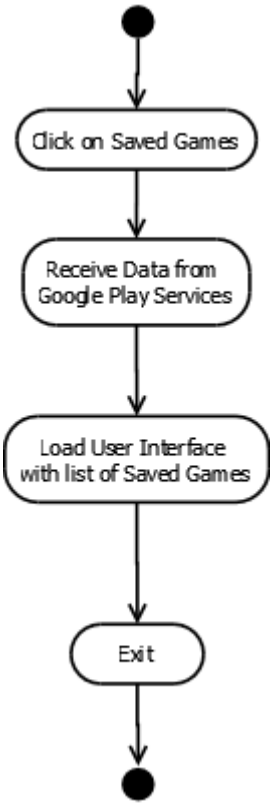

Figure 6. Saved Game Activity

4.2.1.4 Leaderboard Activity

Leaderboard Activity is retrieved from google play services when the user clicks to see it. This board is updated every time when player gains points in the match and the final score is shown.

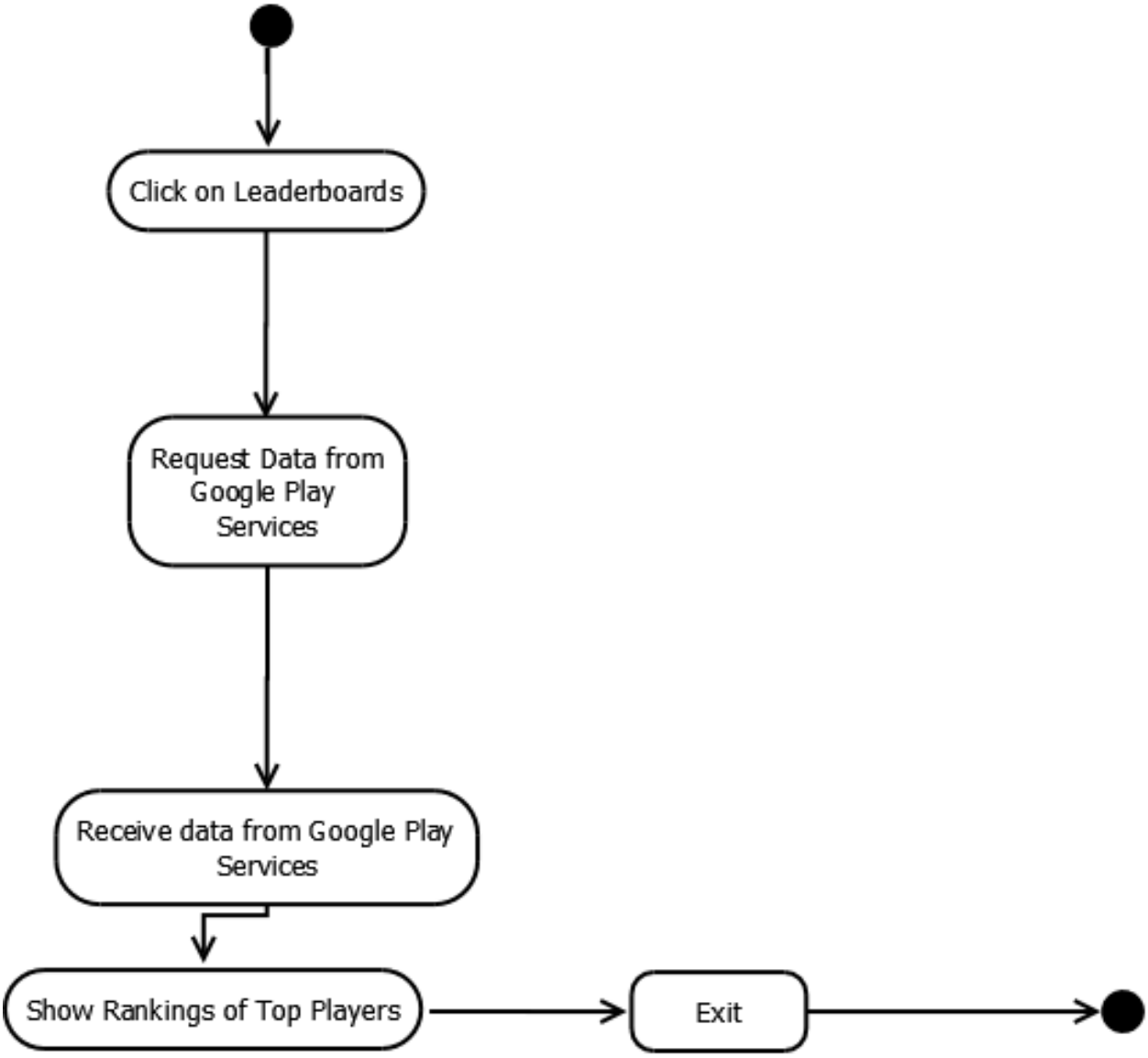

Figure 7. Leaderboard Activity

4.2.1.5 Play with Friend Activity

When the player starts a new match in the play with friend activity, the user is shown a default UI by google play services that shows his/her friends in their google account and player can choose to invite one of them. After the player starts the match and acceptance of invitation is retrieved from opponent, the game starts.

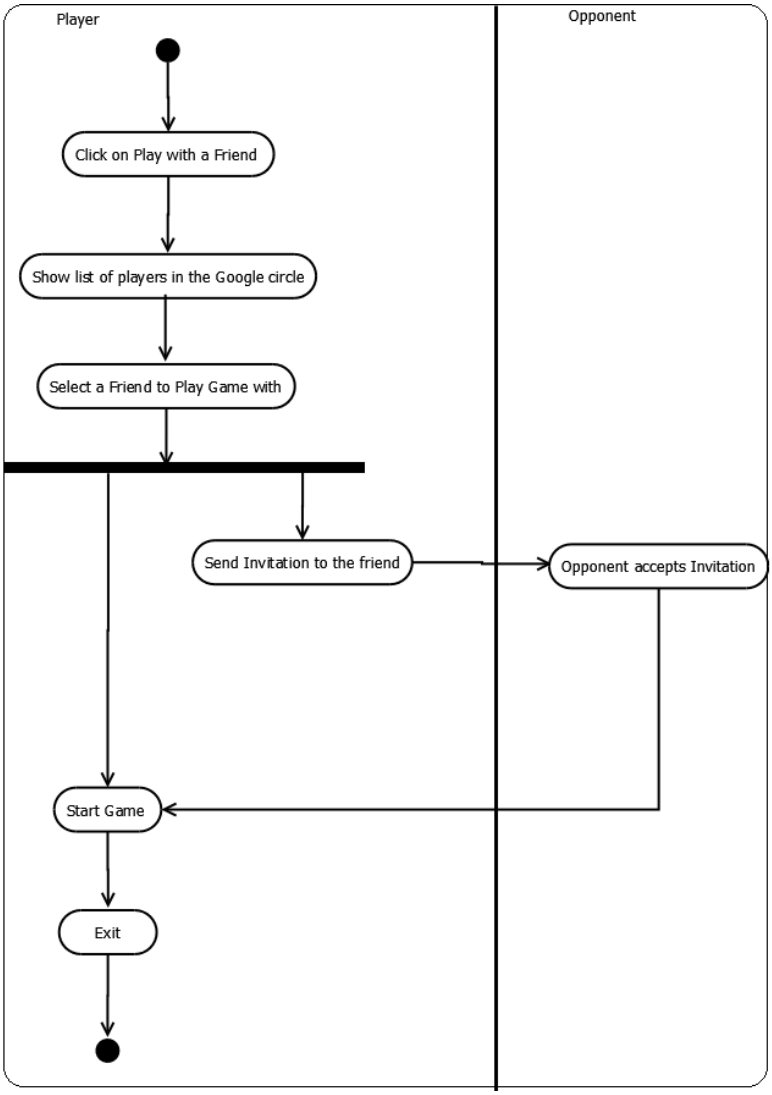

Figure 8. Play with Friend Activity

4.2.1.6 Tournament Activity

When the user clicks start a tournament and number of players is selected, the round robin match is calculated in the client by the initiating player and send to server so that all players can play the match accordingly

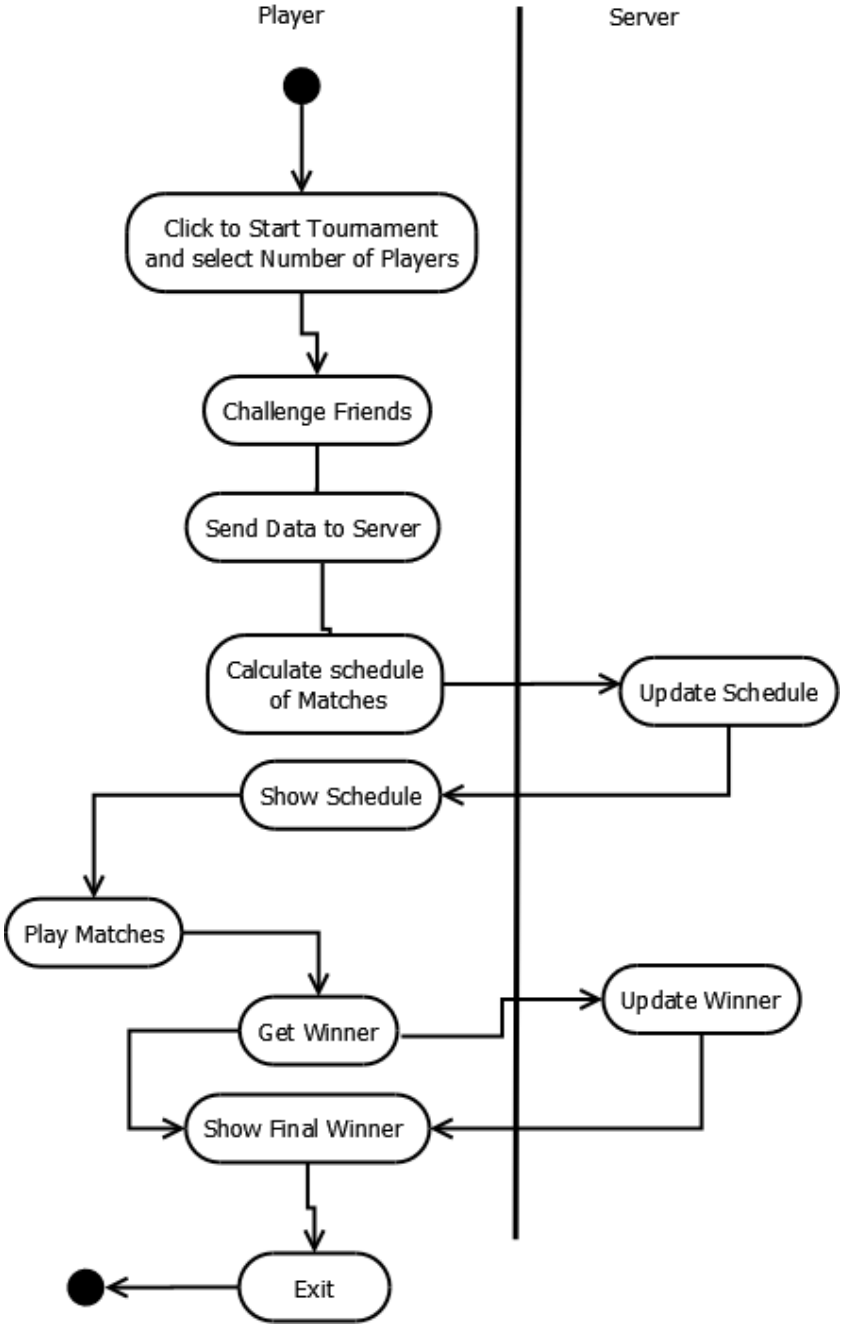

Figure 9. Tournament Activity

4.2.2 Structure diagrams

Structure diagrams shows all the functions that should be there in the system. They are used for defining the architecture of software systems [12]

Class Diagrams in the Unified Modeling Language (UML) describes the structure of a system by showing the system's classes, properties or methods, and the relationships among classes [13].

The following are the classes developed for this app.

1) SessionManager Class : This class manages session clients.
2) Session Class : This class has two sub classes Board Session and Chat Session
3) BoardSession Class : This class has the responsibility for providing google api client for each session of the game.
4) ChatSession Class : ChatSession class provides Send Bird Chat Api Client for a single game and is responsible for chat activity between players in the game.
5) SessionManager Class: SessionManager manages the sessions in the game and shows the sessions in the game.
6) Game Class : Each Session has one game and Game class is the controller that handles movements of players, turns , capture of piece and updates the scores of the game.
7) Board Class : Board class is the model and is responsible for loading game board with data.
8) BoardView : BoardView is the view class that draws and displays the checkersboard.
9) PieceType : This is the enumeration for different piece types.

10) DatabaseAccess class : This class updates and retrieves player's history from database.
11) MainMenuActivity class: This class presents the user with options of navigation to different parts of the game.
12) Tournament class : This class calculates schedule and sends the data to the server and retrieves data from the server to show schedule to players.

The following is the class diagram for the Checkers Game App.

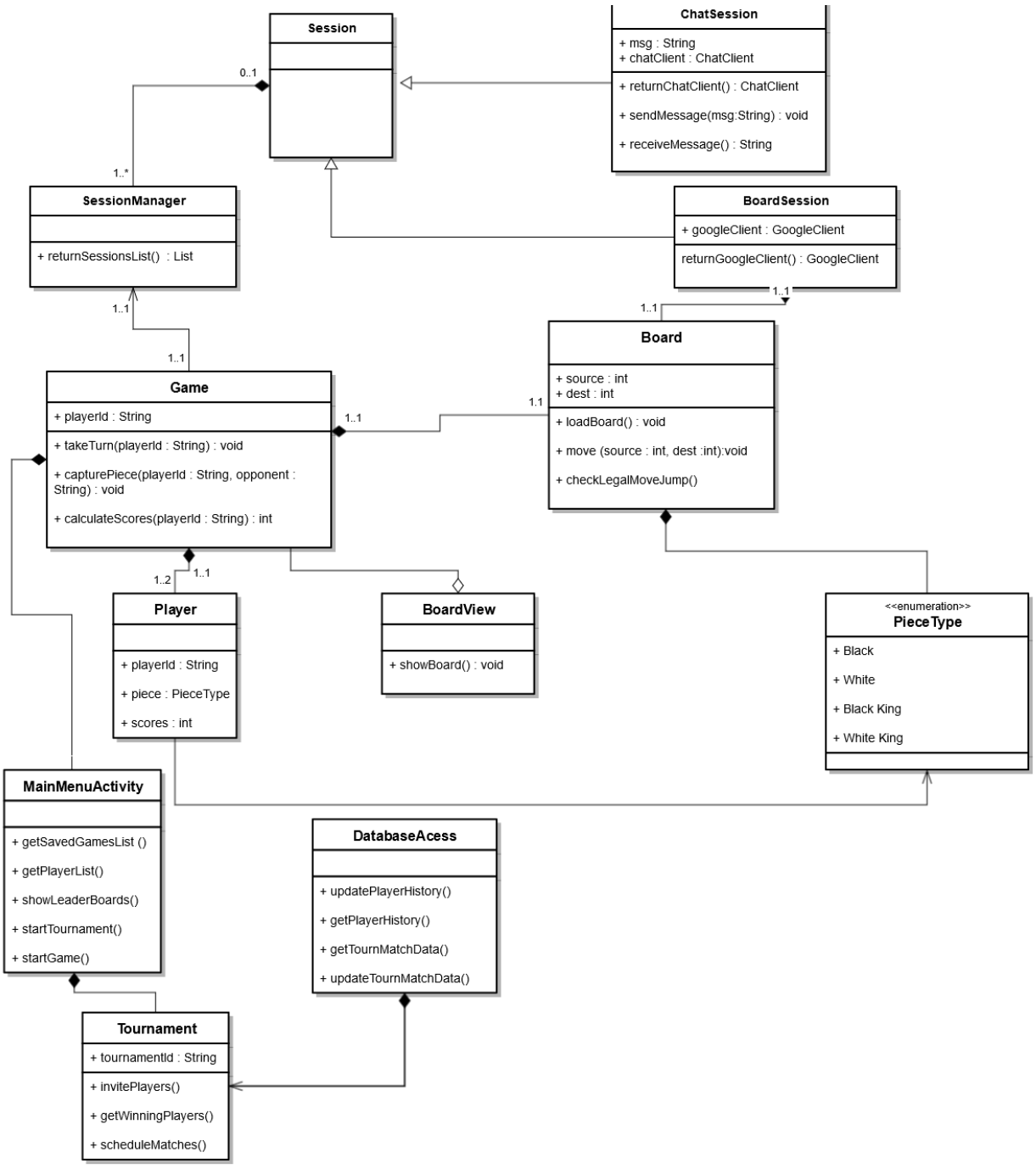

Figure 10. Class diagram Of Checkers App

CHAPTER 5 – IMPLEMENTATION

5.1 Modules in the Game & Graphical User Interface

5.1.1 Login Module

This is the entry module which includes user sign in with his/her gmail username and password to the game. Upon successful authentication with gmail server the user is directed to the Main Menu Screen of the Game.

The following is the start screen of the game where the user is presented with the option of signing to the game.

Figure 11. Start screen

Below is the image of the Main Menu Screen where the user is directed to after login.

Figure 12 Main Menu screen

5.1.2 Play with a friend

The play with a friend module is implemented through google play services. When user chooses this option in the main menu, the user interface with friends in the gmail account are shown and user can select the friend. An invitation is sent to the friend and the game is then started when the friend accepts the invitation.The following is the default user interface for the player to select his/her friend from google account.

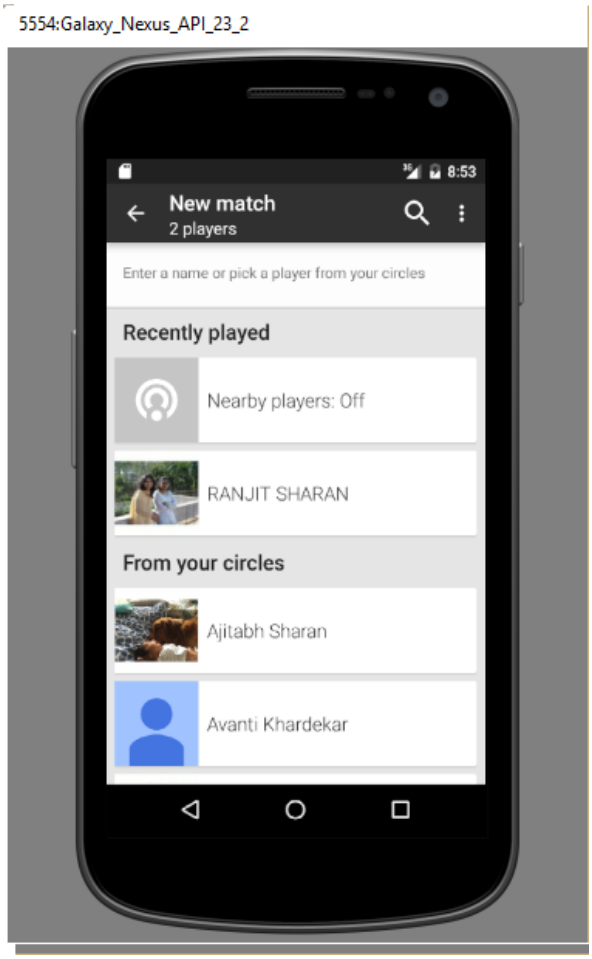

Figure 13 Play with a Friend Screen

5.1.3 Legality Check

The game checks for legal, illegal moves when the player takes a turn. It also checks for legal and illegal capture.

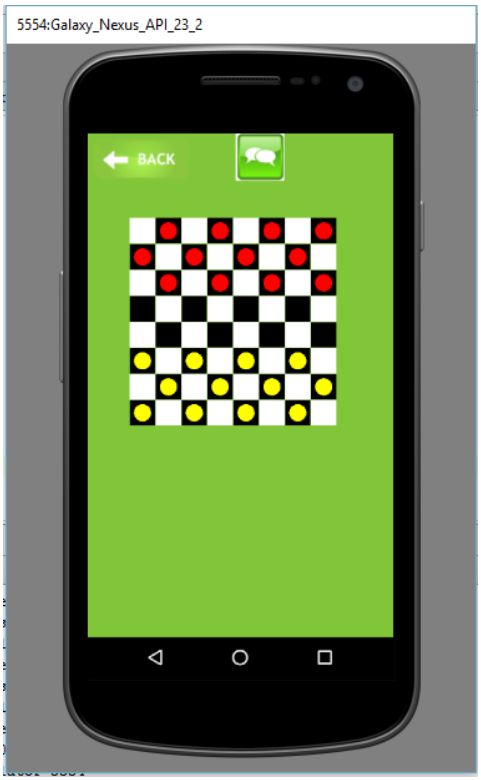

Figure 14 Main Game Screen

5.1.4 Chat

The chat modules allows users to chat with the opponent while playing the game. This is implemented through 1 on 1 messaging feature of Send Bird. The user is presented with chat functionality by a button to be clicked while playing the game. The following is the screen for chat.

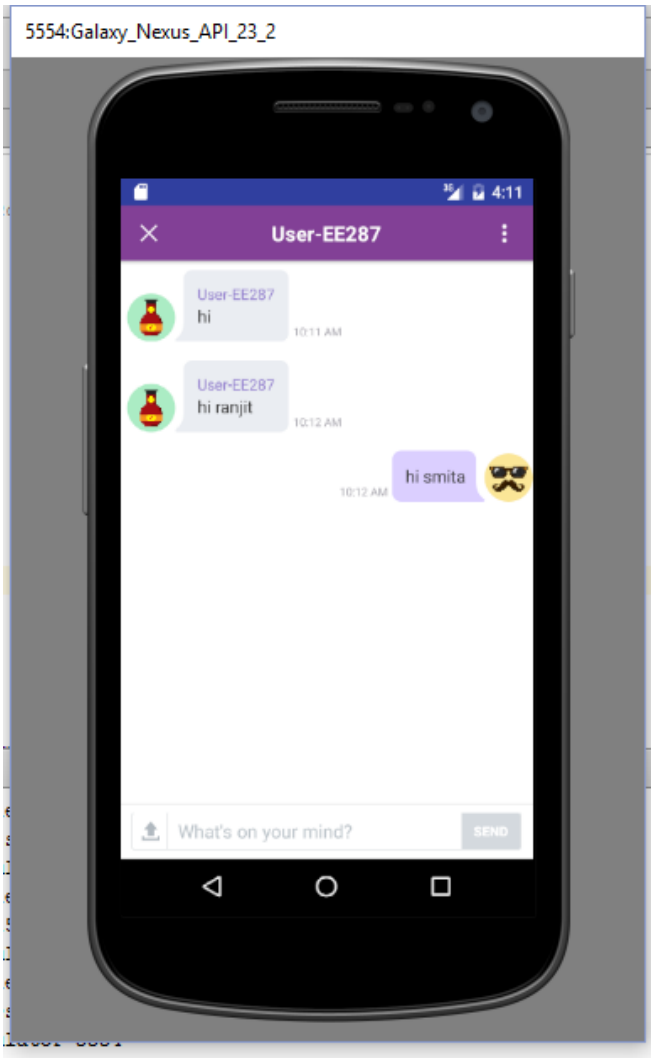

Figure 15 Chat screen

5.1.5 Saved Games

The player can save an incomplete game and later resume the game from the list of saved games in his/her match inbox. This is implemented through google play services.

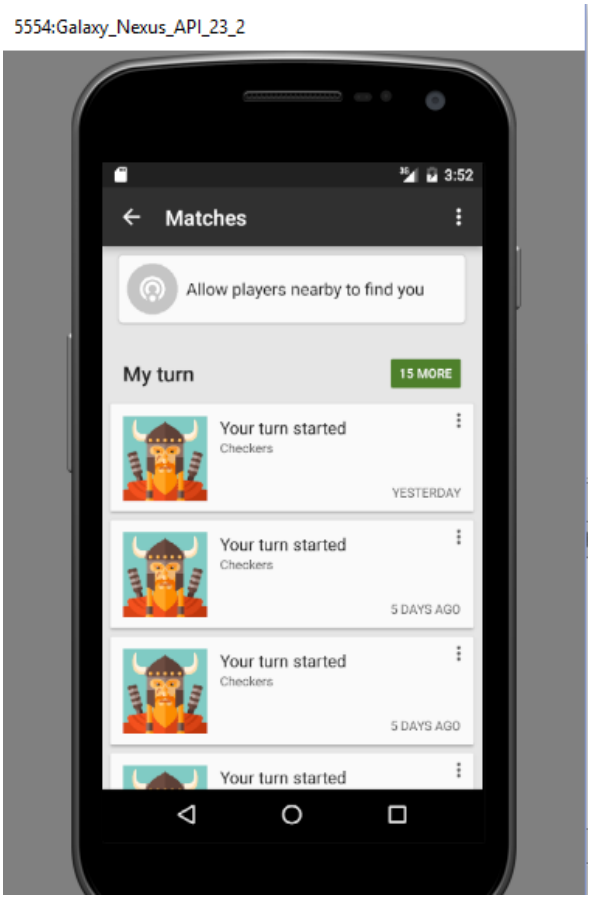

Figure 16 Saved Games (Match inbox) Screen

5.1.6 Leaderboards

The user can see their rankings and scores through leaderboards at the end of the game and through main menu. This is also implemented through google play services.

5.1.7 Scoring & Skill Level

The player gains points on every capture of opponent's piece. The player gains more points when it reaches the opponent's end and it becomes the King.

Beating a good opponent count more than beating a bad opponent

There will be three level of Skill: i) Beginners Level ii) Intermediate Level & iii) Expert Level

5.1.7.1 Brief Description of Skill Level

Beginners levels shall be termed to those players who have no score point (New Player) or having less than the predefined (arbitrarily) values.

Intermediate skill level players shall be those who have accumulated or having scores at or above the set threshold value.This value is arbitrary & can be changed .

Similarly Expert Level Player shall be those who have accumulated or having scores at or above the set threshold value.This value is arbitrary & can be changed

If Low skill level player plays with higher skill level & win the game, then winner shall get special score points & this shall be deducted from the high skill level player.No additional bonus points shall be added in the higher skill player if wins from the low skill level player.

Followig considerations are made for calculating score points.

There will be threshold score points to attain Intermediate and Expert level. There is no limit of number of games played. Any level of player can reach in the next higher skill level by accumulation of scores played in games.

These skill levels are maintained and updated in the device using the Anrdoid internal storage feature which stores private data in memory. When the player logs in,the scores and the skill level are retrieved from the internal file storage of his/her device and after the match according to the above rules are updated back in a file and stored in the device.

5.1.8 Winner Decision(in the game)

The winner is decided based on two conditions:

1) If there are no legal moves left of the opponent then the other player is the winner.

2) If all the pieces of black or white opponent has been captured by the player then he/she is the Winner

5.1.9 Tournament and Scheduling

In this module tournaments are scheduled as per Round - Robin format .

To start the tournament the following steps are followed:

1. A Player first cliks “Start tournament” on the module
2. Player is asked to input number of players to be invited and to be included in the match.
3. This data is sent to game server where round robin schedule is calculated based on the number of participators.
4. The result is retrieved back in the app and all the registerd players get to see the schedule of matches.
5. Before the match, push notifications are sent to the user requesting player to login and click on start tournament match.
6. Upon login, the player will get to click on start tournament which will be shown as an additional option along with other options.
7. When the player clicks on “Start tournament match” player will make a turn in the time allotted for each move and a timer will start.
8. When another player logs in at the same then he/she will play their turn after the first player and then he/she will make the move.This way match will progress and upon

completion of match in the allotted time.Winner and scores data will be saved on the server.

9. Champion/Winner of the tournament is declared at the end of final match

5.1.9.1 Brief description of Round-robin tournment

Round robin tournament is the best way of determining the Winner among the fixed number of participant.Each player or team play with a every players.Final results of the players or participants are very accurate because results are obtained after playing equal competitive game.In this format a competitor play the strongest opponents in quick way while others intermittently play with weaker opponents. When two round competitor meet in final match the result of the final match decides the championship [17].

5.1.9.2 Algorithm to be used for Scheduling Match

Suppose there are even no of players or team Say 'N' then Round robin tournament requires N/2 game and (N-1) round and if this is odd no then no. of game shall be N and N-1/2 rounds and there will be one dummy competitor with no game in that round.

For example Say there are 8 players then no of game shall be 4 and play round shall be 7. Each players shall be assigned with number or ID & then paired off [17].

Sample for Scheduling tournament match (with 8 players)

Round 1 : 1 2 3 4 (1 plays with 8,2 plays with 7&.....)

8 7 6 5

One of the competitior is fixed in the first or last column of the table & then it is rotated clockwise one position

Round 2: 1 8 2 3: (1 plays with 7, 8 with 6 &.....) Round 3; 1 7 8 2

7 6 5 4 6 5 4 3

Round 4 1 6 7 8 Round 5: 1 5 6 7 Round 6: 1 4 5 6

5 4 3 2 4 3 2 8 3 2 8 7

Round:7 1 3 4 5

2 8 7 6

CHAPTER 6 – TESTING

Testing of software is the execution of each component of a system to check its functionalities are according to requirements [14].

6.1 Unit Testing

Unit testing refers to tests that verify the functionality of a specific section of code, usually at the function level or at the class level. Small independent parts of code are tested for its right execution

These types of tests are usually written by developers as they work on code and can be done manully, to ensure that the specific function is working as expected [15].

Table 1. Unit Test Cases

Serial No.	**Test Case**	**Result**
1	The user signs in with correct username and password.	The user signs in successfully and is directed to Main Menu Screen.
2	User clicks play with friend feature.	The UI is shown having list of friends shown from google account.
3	The player selects friend to play with.	Game is started with player given the first turn and

		notification sent to the player selected.
4	The player clicks chat on screen for 1 on 1 messaging with opponent.	The user is shown the chat window to chat with the opponent.
5	The player makes the wrong move	Game does not allow the user to make the wrong move.
6	The player tries to capture its own piece or any wrong capture.	Game does not allow the player to make the wrong capture
7	The player leaves the game without completion or clicks save on the game.	The game is saved and is shown in the list of saved games UI from the main menu.
8	The player clicks leaderboards option on the main menu	The user is shown top rankers with their scores.
9	The player reaches opponent's side of board	The player becomes King with a different color.
10	The player clicks tournament and adds players.	Notification is sent to all the players

11	The player clicks start tournament	Matches are scheduled and shown in the UI

6.2 Compatibility Testing

Compatibility Testing tests if the application is compatible with different combinations of hardware and software configuartions [16].

The checkers game app has been tested for compatibility with Google Nexus with API 23 emulator and Moto G 3rd generation Device with Android Version Marshmallow and is running successfully under both the environment.

6.3 Integeration Testing.

All the modules in the game are integerated to form the complete system and tested so that complete workflow right from the beginning works as expected. Following are the some of the test cases:

Table 2 Integeration Testing

Seriel No.	Test Case Objective	Test Case Description	Expected Result
1.	Check the interface link between Login and Main Menu Screen	Enter Login Credentials	The player is directed to Main Menu sceen
2.	Check the interface between Play with Friend and Main Game	Select friend to be played with from the User Interface	Both the players are directed to the same Game Screen with

			the player identity information.
3.	Interface between Game screen and chat	On click of chat button	User is directed to chat screen that allows messaging between the chat and the opponent.

6.4 Performance Testing

Performance Testing of the app is done through TraceView Tool. It is used for profiling android apps. Traceview generates a trace log file and is displays the data in two panels called timeline panel and profile panel. The profile panel shows executions happened inside the method and timeline panel shows for each thread , its starting execution time and when it stopped.

There are two profiling methods Sample based Profiling and Trace Based Profiling.

The Sample based Profiling is the output shown by interrupting the VM at given frequency and collecting the call stacks at that time and the Trace Based Profiling shows the ouput by tracing the entry and exit of each and every method.

The following shows the output of Trace View Tool after profiling the game app. Both the methods mentioned are used.

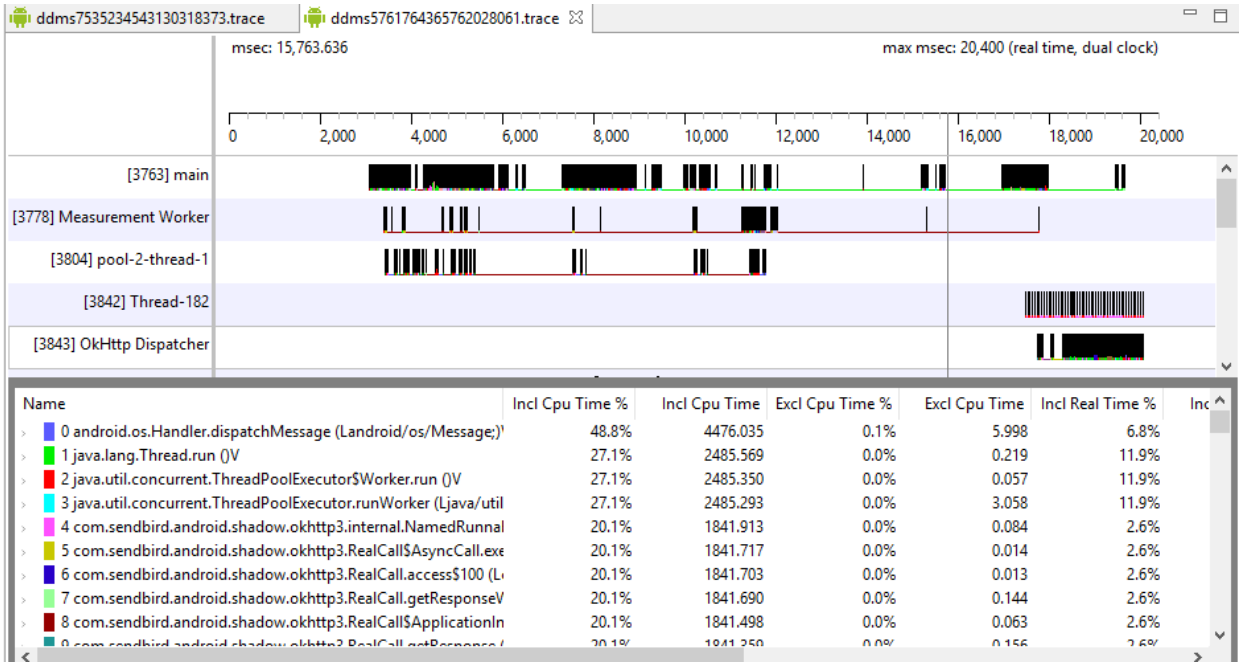

Figure 17 TraceView Result for the app (Sample Based Profiling)

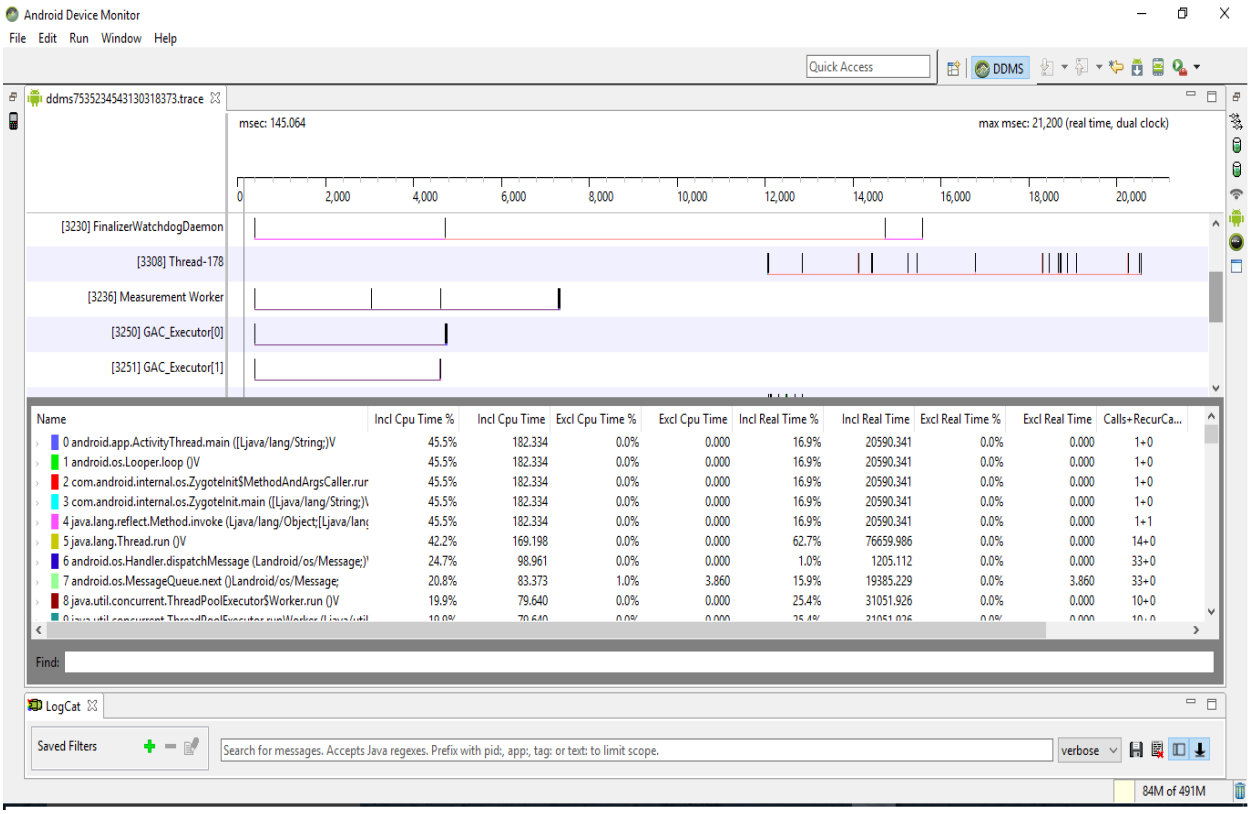

Figure 18 TraceView Output for App (Sample Based Profiling)

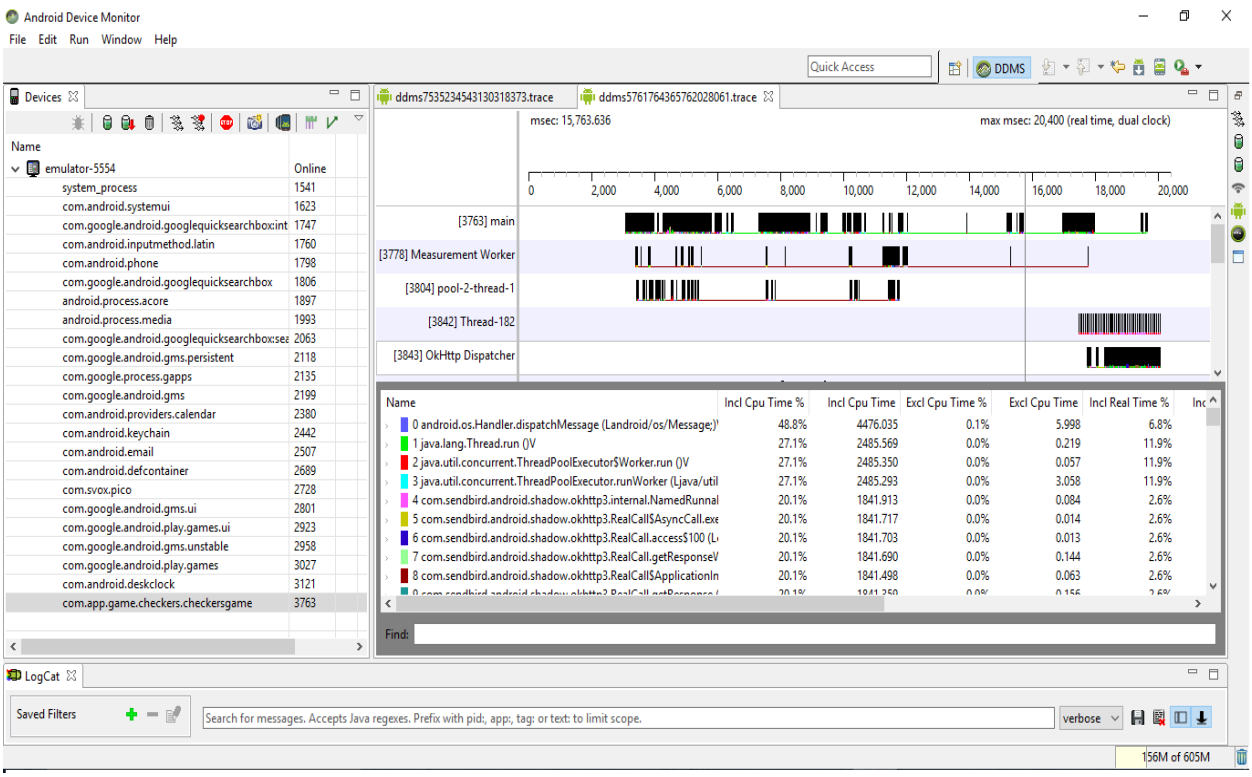

Figure 19 TraceView Output for App (Trace Based Profiling)

6.5 Usability Testing

Usability Testing is a method of gaining user feedback by giving the control to real users. This process helps in knowling what to do in the future and what are the limitations and deficiencies of the existing app. The following table gives the results and feedback when the app was tested on real users.

Table 3 Usability Testing

Sl. No.	Functionality Tested	Pass/Fail	Comments/Feedback
1.	Chat with another player.	pass	The chat screen should be on the same screen as part of the game board screen.

2.	Sign In with user name and password.	pass	It works fine.
3.	Play with Friend functionality.	pass	It works fine.
4.	Main game	pass	App can be improved by having a notification to the player when other player looses connectivity.
5.	Main game	pass	Game does not show captured pieces for each player .
6.	Leaderboards	pass	Leaderboards can also show number of matches won by players.

CHAPTER 7 – CONCLUSION AND FUTURE WORK

During the course of the project, I gained knowledge and skills in Java Programming in Android. I also learned to use Google Play Services to implement Turn Based Games, Saved Games feature, Leaderboards feature and Login module and ways to interact with mobile Game backend server.

The Checkers Game App is an exciting game that entertains players. Players can enjoy the game in various ways by achieving a certain skill or by playing with a known friend or scheduling matches to enjoy the full aspects of the game.

The Checkers game can be enhanced in many ways. The game currently is a two player game and so there is a limitation that a human opponent should be present for the game to be played. The future work may include playing with computer using Artificial Intelligence principles. There can also be a chat future during the tournament also when all the players can join a chat room and chat together. The game can also be migrated to be iOS app so that apple product users can play the game.

CHAPTER 8 – REFERENCES

[1] An overview of the Android architecture. (2008). Retrieved July 10, 2016, from http://www.techotopia.com/index.php/An_Overview_of_the_Android_Architecture

[2] Android core technologies. Retrieved August 12, 2016, from https://source.android.com/devices/tech/index.html

[3] The Android source code. Retrieved August 11, 2016, from https://source.android.com/source/index.htm

[4] Intents and intent filters. Retrieved July 14, 2016, from https://developer.android.com/guide/components/intents-filters.html.

[5] Activities. Retrieved July 13, 2016, from https://developer.android.com/guide/components/activities.html

[6] Fragments. Retrieved July 14, 2016, from https://developer.android.com/guide/components/fragments.html

[7] App manifest. Retrieved July 13, 2016, from https://developer.android.com/guide/topics/manifest/manifest-intro.html

[8] Systems design (2016). In *Wikipedia*. Retrieved July 15, 2016 from https://en.wikipedia.org/wiki/Systems_design.

[9] Google play services (2016). In *Wikipedia*. Retrieved July 18, 2016 from https://en.wikipedia.org/wiki/Google_Play_Services

[10] Family, S. *Messaging and chat API for mobile Apps and Websites*. Retrieved July 16,2016, from https://sendbird.com/

[11] Unified modeling language (2016). In *Wikipedia*. Retrieved July 16, 2016 from https://en.wikipedia.org/wiki/Unified_Modeling_Language

[12] Class diagram (2016). . In *Wikipedia*. Retrieved July 16, 2016 from https://en.wikipedia.org/wiki/Class_diagram

[13] Software testing (2016). In *Wikipedia*. Retrieved from July 17, 2016 https://en.wikipedia.org/wiki/Software_testing

[14] Unit testing (2016). . In *Wikipedia*. Retrieved from July 17, 2016 https://en.wikipedia.org/wiki/Unit_testing

[15] Different types of testing (eg. Unit, functional, integration, etc.) document. (2016). Retrieved July 27, 2016, from http://stackoverflow.com/questions/22414134/different- types-of-testing-eg-unit-functional-integration-etc-document

[16] Mobile backend as a service (2016). . In *Wikipedia*. Retrieved Aug, 3, 2016 from https://en.wikipedia.org/wiki/Mobile_backend_as_a_service

[17] Round-robin tournament (2016). . In *Wikipedia*. Retrieved Aug,3,2016 from https://en.wikipedia.org/wiki/Round-robin_tournament