****

K.M.Suyama D. Dharmasena 13208369

Thilina Warnakulasooriya 13208380

Uditha S. Siriwardhana 13208372

P.M.C.Ayesha Panditharathne 13208376

R.M.D.Nilan Jayawardhana 13208374

G.K Eranjith Udayanga 13208198

**CONTENT**

# Introduction…………………………………………………………………………… 3

# Software Specification……………………………………………………………. 4

## *Programming Language*

## *Working Environment*

## *Work Flow Diagram*

## *Flow chart Diagram*

## *Class Diagram*

# Software Implementation……………………………………………………… 12

## *What we have done so far*

## *What we have to do in the future*

# Project Plan…………………………………………………………………………. 16

## *Meeting Plan*

## *Activity Plan*

**5 Conclusion…………………………………………………………………………… 19**

**Chapter 01**

**INTRODUCTION**

[Poker](http://www.pagat.com/poker/) is a five-card [vying game](http://www.pagat.com/vying/) played with standard playing-cards.

A five-card vying game is one where, no matters how many cards may be dealt to each player, the only valid combinations are those of five cards. In orthodox Poker these are, from highest to lowest:

* straight flush (five cards in suit and sequence, Ace high or low, as heartAKQJ10 or spade5432A)
* four of a kind, fours (four cards of the same rank and one idler, as K-K-K-K-x)
* full house (three of one rank and two of another, as Q-Q-Q-4-4)
* flush (five cards in suit but not in sequence, as heartJ-heart9-heart8-heart7-heart3)
* straight (five cards in sequence but not in suit, as spade10-spade9-diamond8-club7-heart6)
* three of a kind, threes, triplet, trips (three of the same rank plus two of two different ranks, as 7-7-7-x-y)
* two pair (as Q-Q-9-9-x)
* one pair (as 3-3-x-y-z)
* high card (no combination: as between two such hands the one with the highest card wins)

(The highest possible straight flush, consisting of A-K-Q-J-10 of a suit and known as a royal flush, is sometimes added to the list in order to bring the number of combinations up to the more desirable ten, but of course it is not different in kind from a straight flush. Other five-card combinations, known as freak hands, are recognized in unorthodox [Poker variants](http://www.pagat.com/poker/variants/).)

Poker is of French-American origin and is the national vying game of the United States, though it has come to have a world-wide following in many different forms.For the purpose of the Software engineering Project 1, our team IMMORTALS have given the poker game style called Five-Card Draw.

**Chapter 02**

**SOFTWARE SPECIFICATION**

2.1 Programming Language

**Java**

Java is a high-level object-oriented programming language developed by the Sun Microsystems. Though it is associated with the World Wide Web but it is older than the origin of Web. It was only developed keeping in mind the consumer electronics and communication equipments. It came into existence as a part of web application, web services and a platform independent programming language in the 1990s.

Java is an object-oriented language, and this is very similar to C++. Java Programming Language is simplified to eliminate language features that cause common programming errors. Java source code files are compiled into a format called byte code, which can then be executed by a Java interpreter.

Java is influenced by C, C++, and Smalltalk and borrowed some advanced features from some other languages. The company promoted this software product with a slogan named “Write Once Run Anywhere” that means it can develop and run on any device equipped with Java Virtual Machine (JVM). This language is applicable in all kinds of operating systems including Linux, Windows, Solaris, and HP-UX etc.

Reasons for preferring Java over other programming languages at this time are:

* Price – it is free
* Performance - really fast these days thanks to the Hot Spot JIT compiler
* Effectiveness - lots of power with rigorous features like type-safe, sand-boxed, etc.
* OOP capability
* Very good, well-thought out exception handling; C++ exceptions are the opposite.
* Portability - it runs on almost everything
* Tool availability - awesome IDEs like Eclipse & NetBeans are free, as are web servers like Tomcat and application servers (JBoss, Glassfish, Geronimo, etc.)
* Flexibility - does graphics, desktop GUIs, web user interfaces - all kinds of things in all kinds of runtime environments
* Aptness - many enterprise apps today have to support HTML, SQL, and XML - Java has good support for all of them built in and you can get third-party libraries for free that make this even easier/better
* Well-supported - Sun keeps adding improvements and fixing thing going one or two versions back
* Forward compatibility - unlike something like VB which undergoes wrenching change in its syntax every couple versions or so, Java's syntax and semantics seem about 99.9% upward compatible from version to version

We can use Java programming language in any OS without any problems as well as for any tools.

Following OS which we can use Java programming language.

* MS-Windows 95, 98, NT4, 2000, XP
* Sun Solaris/SunOS Unix
* IBM AIX
* Mac OS X
* Linux
* Motorola cell phones (MIDP, CLDC)
* Palm Pilot PDA

Java VMs (virtual machines) are everywhere. They are in:

* web browsers
* cell phones
* PDAs
* desktop computers
* web servers
* application servers

We intend to use Java programming language for the Application because of the above advantages as well as good features of java for the success of our end ever.

## 2.2 Working Environment

**Eclipse**

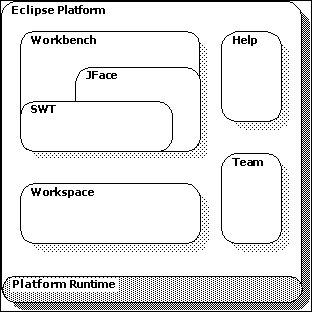
Industry leaders Borland, IBM, MERANT, QNX Software Systems, Rational Software, Red Hat, SuSE, TogetherSoft and Webgain formed the initial eclipse.org Board of Stewards in November 2001. By the end of 2003, this initial consortium had grown to over 80 members.

Eclipse is a platform that has been designed from the ground up for building integrated web and application development tooling. By design, the platform does not provide a great deal of end user functionality by itself. The value of the platform is what it encourages: rapid development of integrated features based on a plug-in model.

Eclipse provides a common user interface (UI) model for working with tools.  It is designed to run on multiple operating systems while providing robust integration with each underlying OS.  Plug-ins can program to the Eclipse portable APIs and run unchanged on any of the supported operating systems.

Platform structure

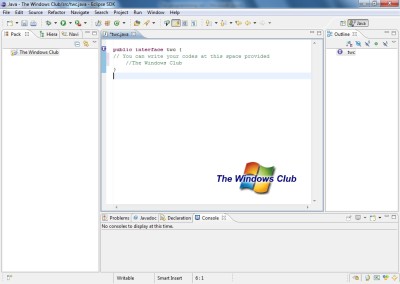
The Eclipse platform itself is structured as subsystems which are implemented in one or more plug-ins.  The subsystems are built on top of a small runtime engine. The figure below depicts a simplified view.



The plug-ins that makes up a subsystem defines extension points for adding behavior to the platform.  The following table describes the major runtime components of the platform that are implemented as one or more plug-ins.

Nowadays programming isn’t much difficult as the new IDE’s is being introduced, provide us with a good developmental environment which helps us to create good GUI applications. This article will tell you about **Eclipse IDE**, an open source platform. Eclipse began as an IBM Canada project, but it is now an open source project.

Eclipse makes your programming experience even better with its easy to use GUI and it saves your time for writing long and lengthy codes. Eclipse comes for different programmers.

[](http://thewindowsclub.thewindowsclubco.netdna-cdn.com/wp-content/uploads/2012/02/Eclipse.jpg?c2fdaa)

It is available for:

* JAVA Developers
* JAVA EE Developers
* C/C++ Developers
* JAVA Script Web Developers
* RCP and RAP Developers
* JAVA and Report developers
* Testers
* Parallel Application developers

### Eclipse IDE Features

The program has many features that would enhance your programming experience. Features such as migration into JAR and Debug tool are very powerful and they make your task easier than before. So have a look on feature list of the application:

* Import programs
* Export Programs
* Cleanup
* Migrate into JAR file
* Cerate and Apply script
* History
* Build Automatically
* Build Working Set
* Debug Tool
* Debug Configuration
* Break Points
* Interfaces
* External Tools.

### Eclipse IDE Platform

Eclipse comes with a rich client platform that helps you create good applications and it enhances your programming experience. The Eclipse platform is based on Equinox, an execution of the OSGi core framework specification and it uses plug-ins to add more functionality to it. The following components make up the platform:

* Equinox OSGi
* Core platform
* Standard Widget Toolkit (SWT) – a portable widget toolkit
* JFace
* Eclipse Workbench

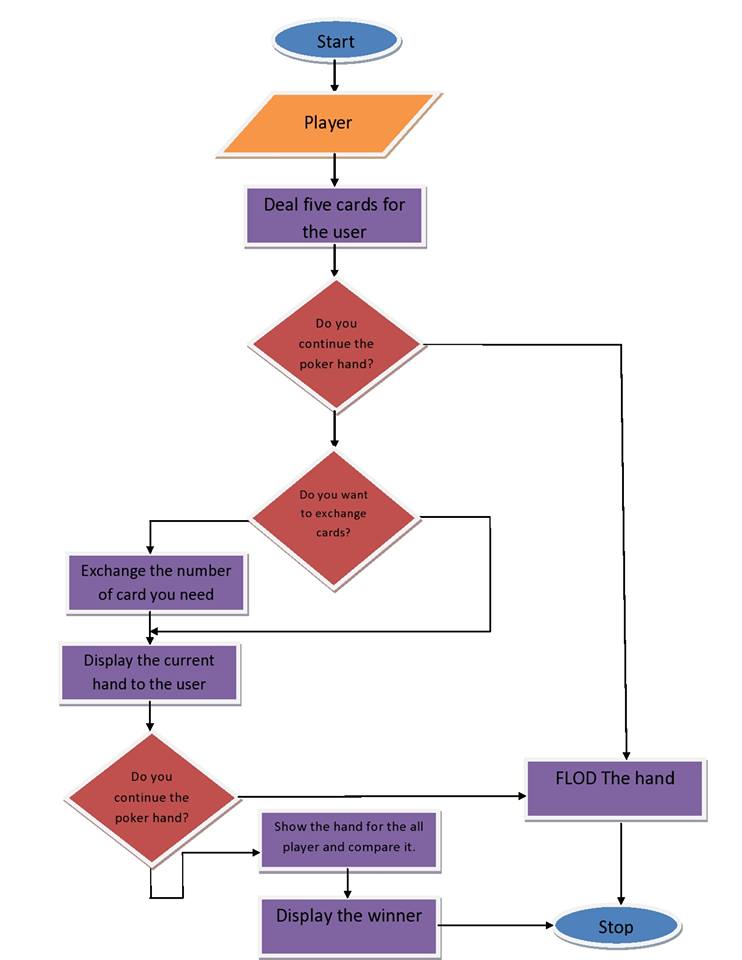
### Why we choose Eclipse?

We prefer Eclipse over Net Beans for many reasons. The first one is the startup time, Net Beans takes ages to load, and loading on the first instance is terrible in case of Net Beans IDE. Eclipse is very simple to get started with. The intelligence feature on Eclipse is better than that on Net Beans. On the other hand what is interesting in Net Beans is default widget support of AWT or Swings unlike Eclipse which implements widgets using SWT.

We decide to Eclipse for the Application because of the above advantages as well as good features of Eclipse for the success of our end ever.

2.3 Work Flow Diagram

2.4 Flow Chart Diagram



2.5 Class Diagram

**Chapter 3**

**SOFTWARE IMPLEMENTATION**

We have divided our project activities in to categories and fulfill them in order. In this chapter we would like to elaborate the progress of our project and the future activities of the project.

The main activities are:

* Interface design sketch
* Shuffling coding
* Distribution of cards & Hand checking
* Testing errors
* User Interface
* Networking

**What we have done so far**

First we have design a user interface and make rules and conditions of the game. We choose 4 players to play the game and the number of players will not be change.

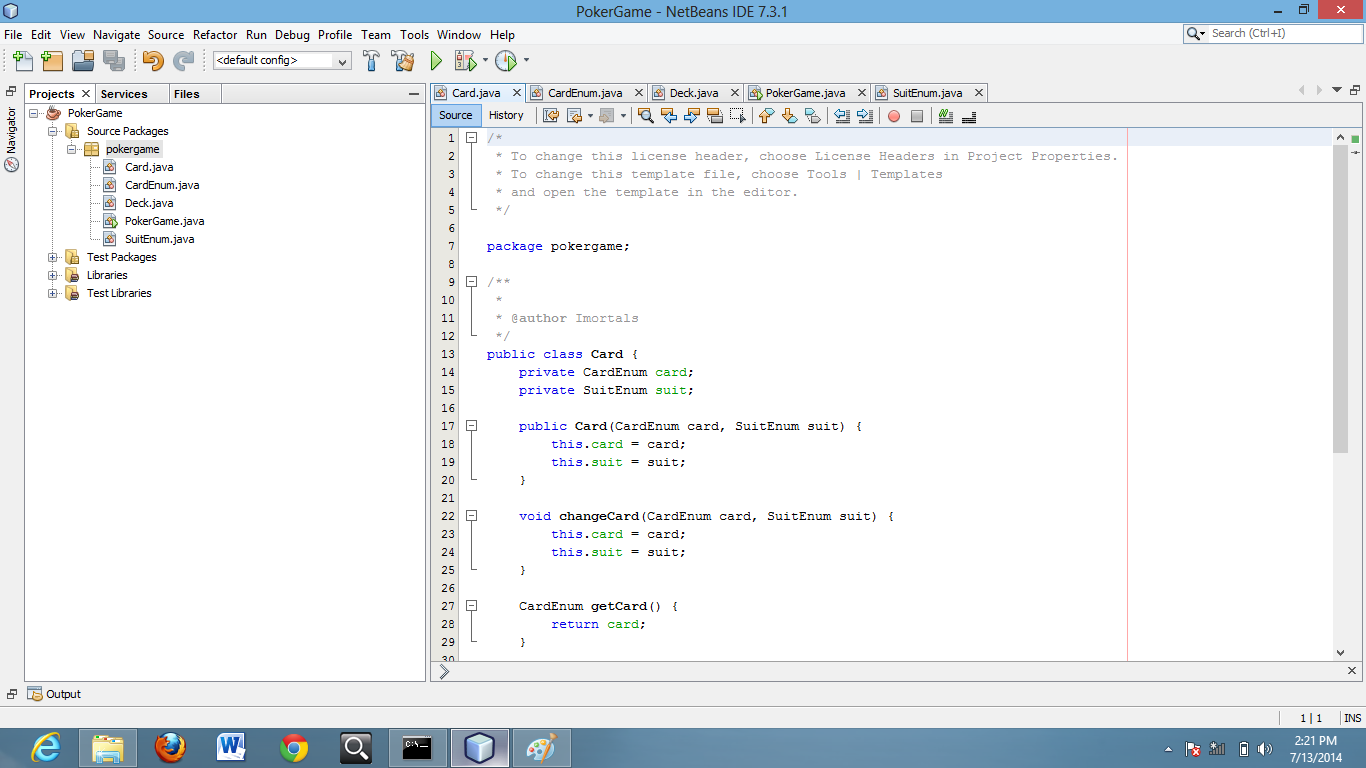
Then we start coding of card shuffling. In this we did background check and did the implementation.

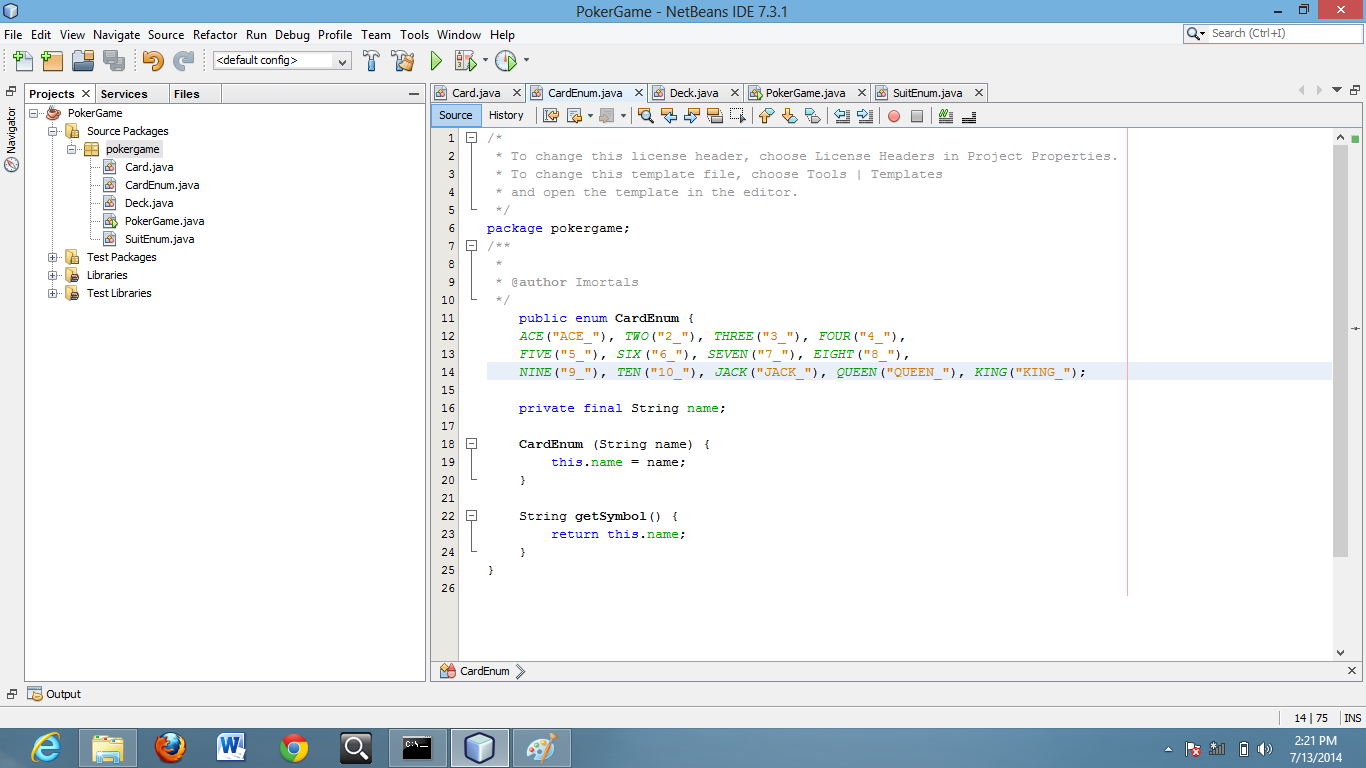
Third, we divide the 13 methods of hand checking algorithms among 3 members and they did the coding of 10 out of 13 now. There are 3 more types to code.

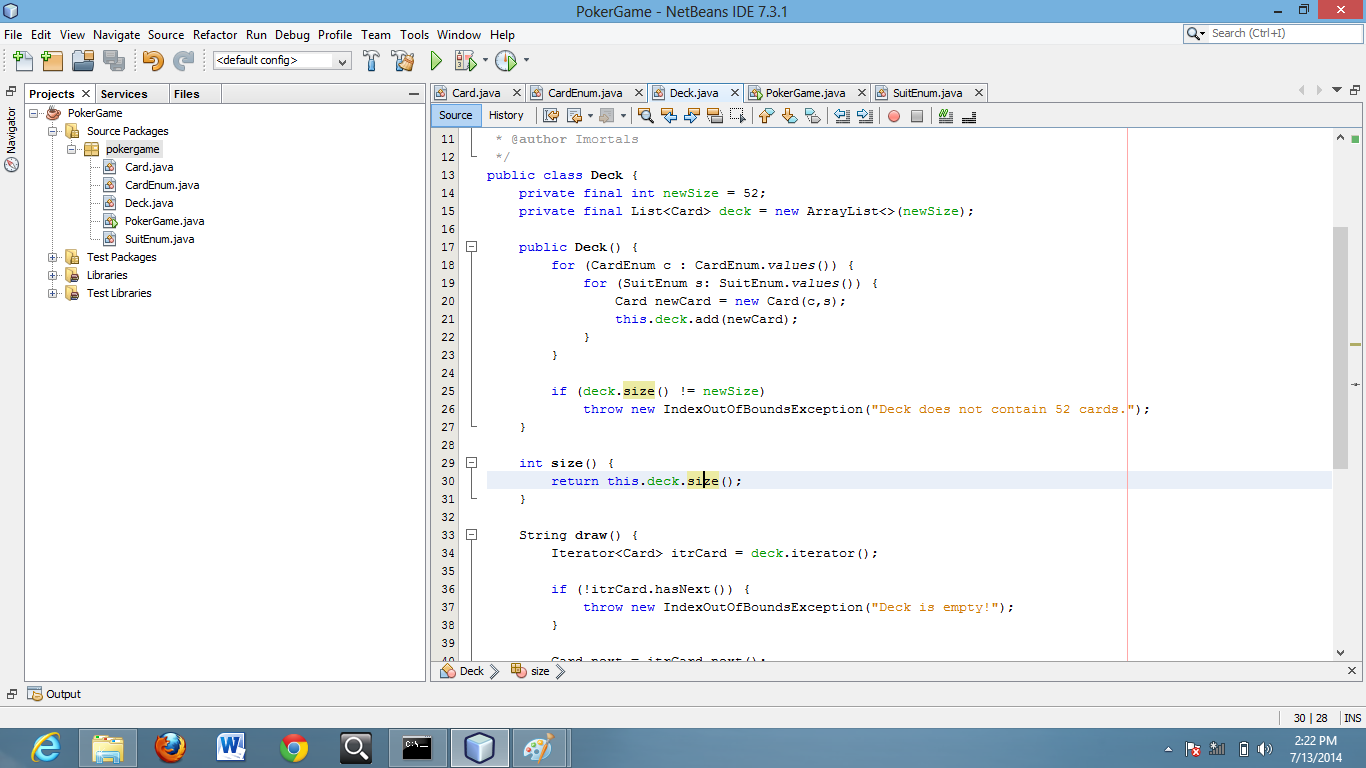
In addition to that, we started referencing about error testing codes and error handling part of the game. Few members were assigning for that in present.

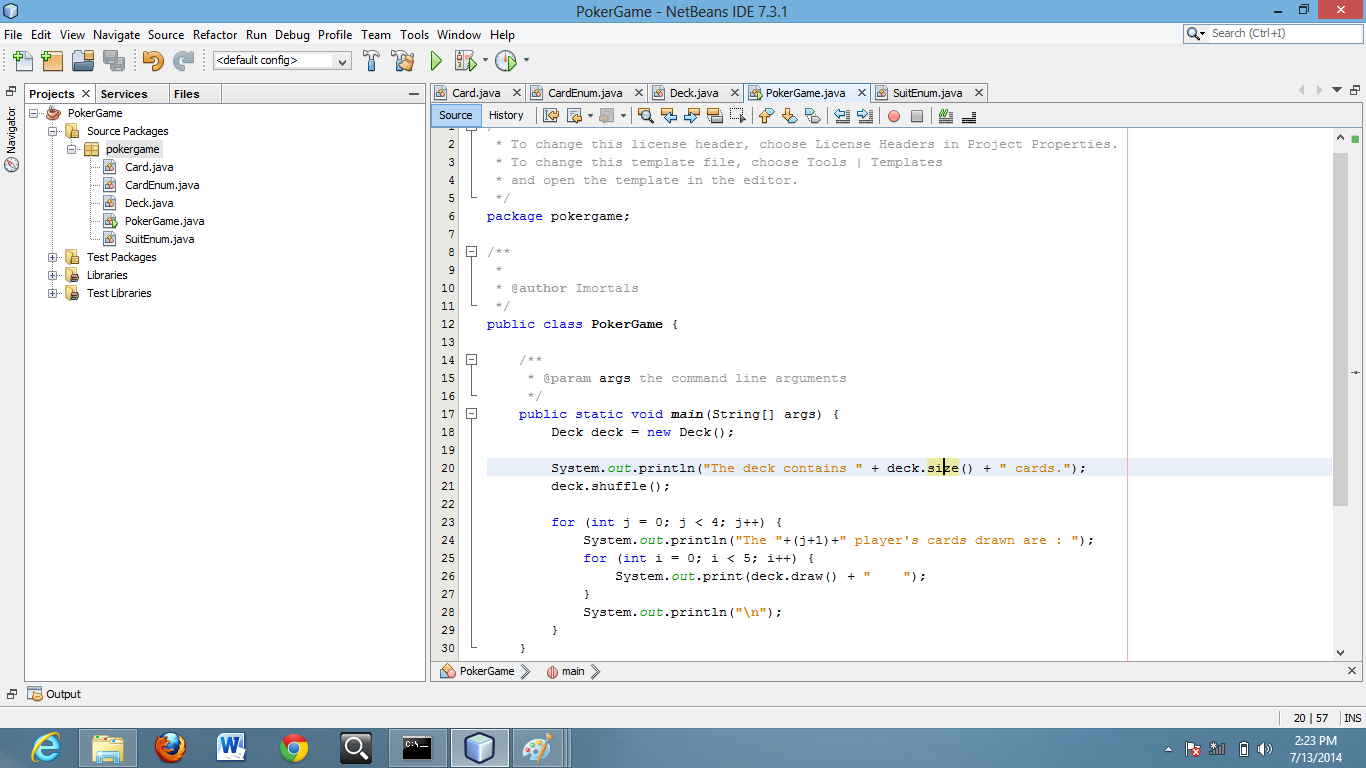


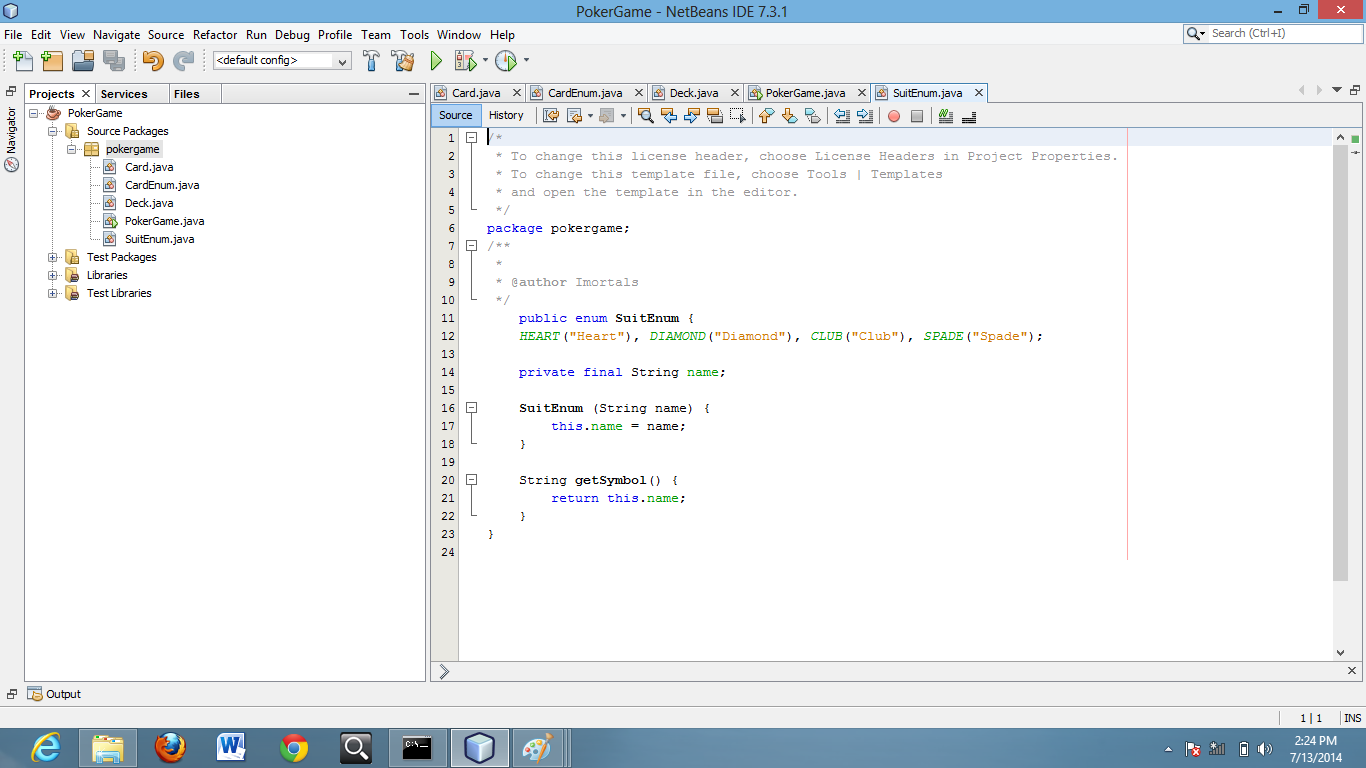
*The user interface sketch*

**

**

**

**

**

*Samples from coding*

## What we have to do in the future

We did not complete the priority checking part of the program.

To complete the project we have to code the distribution part of the five card draw card game and error handling part.

Currently we are planning to implement the game by selecting no of users, since we programmed the game for 4 players. We have to discuss and implement that part of the program.

We have to do the Artificial Intelligence part (When we play with the computer, other players replying part have to programmed)

Then we have to design the interface and GUIs according to the sketch we have done earlier.

Next, we have to do the networking part that connects all the game players.

Finally we have to combine all codes which were implemented by the group members and connect it with the network.

**Chapter 4**

**PROJECT PLAN**

4.1 Meeting Plan

This is a summary of meeting reports we held for last few weeks. Above table will include date of the meeting, the time period of each meeting, the place and the purpose of the meeting.

We have planned 30 meetings, by now we held 17 meetings.

|  |  |  |  |
| --- | --- | --- | --- |
| **Meeting No/Date** | **Time** | **Place** | **Purpose** |
| 01 (10/06/2014) | 1300hrs-1530hrs | NSBM Study Area | Understanding the rules of the game & select a game mode. |
| 02 (10/06/2014) | 1800hrs-1900hrs | NSBM Lecture Hall | Understanding the rules of the five card draw poker game. |
| 03 (11/06/2014) | 1500hrs-1700hrs | NSBM Study Area | Interface Planning |
| 04 (14/06/2014) | 1000hrs-1200hrs | NSBM Lecture Hall | Designing the rules of the game |
| 05 (16/06/2014) | 1300hrs-1600hrs | Fab | Card Shuffling coding |
| 06 (18/06/2014) | 1330hrs-1600hrs | Devon | Animation Designing for Distributing Cards |
| 07 (19/06/2014) | 0930hrs-1200hrs | Devon | Finalize the shuffling code |
| 08 (23/06/2014) | 1000hrs-1300hrs | Macdonalds | Finalize the interface design and distribution animation codes. |
| 09 (24/06/2014) | 1000hrs-1400hrs | Thilina’s home | Discuss about the error testing logics and codes |
| 10 (30/06/2014) | 0900hrs-1400hrs | Thilina’s home | Finalize the ranking coding |
| 11 (01/07/2014) | 1000hrs-1600hrs | Thilina’s home | Hand ranking coding and start to prepare interim report |
| 12 (03/07/2014) | 1000hrs-1500hrs | Thilina’s home | Reference check for testing error codes |
| 13 (04/07/2014) | 1000hrs-1200hrs | Thilina’s home | Collect all codes from members and finalize them |
| 14 (07/07/2014) | 1000hrs-1600hrs | KFC | Hand Checking coding |
| 15 (10/07/2014) | 1000hrs-1600hrs | Garden | Finalize hand checking coding |
| 16 (12/07/2014) | 1000hrs-1300hrs | Barista coffee | Start coding GUIs |
| 17 (13/07/2014) | 1000hrs-1300hrs | NSBM | Interim Report Completion |
| 18 (14/07/2014) | 1000hrs-1300hrs | Suyama’s home | Start priority checking coding |
| 19 (15/07/2014) | 1000hrs-1300hrs | NSBM | Start error handling |
| 20 (16/07/2014) | 1000hrs-1300hrs | Suyama’s home | Distribute the priority checking coding to members |
| 21 (17/07/2014) | 1000hrs-1500hrs | Suyama’s home | Background check for AI part of the game |
| 22 (19/07/2014) | 0900hrs-1300hrs | Devon | Start the AI part coding |
| 23 (20/07/2014) | 1000hrs-1300hrs | NSBM | Finding facts for final report complete error handling |
| 24 (21/07/2014) | 1000hrs-1500hrs | NSBM | Complete AI part and Priority checking part |
| 25 (22/07/2014) | 1000hrs-1300hrs | Thilina’s Home | Complete GUIs |
| 26 (23/07/2014) | 1100hrs-1600hrs | NSBM | Start writing the final Report |
| 27 (24/07/2014) | 1000hrs-1600hrs | Thilina’s Home | Combine coding |
| 28 (25/07/2014) | 1000hrs-1600hrs | NSBM | Networking |
| 29 (26/07/2014) | 1000hrs-1600hrs | Thilina’s Home | Testing the program & do final error handling part |
| 30 (27/07/2014) | 1000hrs-1600hrs | NSBM | Complete the final report |

4.2 Activity Plan

This is a summary of duties and responsibilities of each member and time period for each job.

|  |  |  |  |
| --- | --- | --- | --- |
| **Member** | **Task** | **Due Date** | **Checked By** |
| K.M.Suyama D. Dharmasena – 13208369 | Hand Checking coding(full house, flush, straight) | 10/7/2014 | R.M.D.Nilan Jayawardhana - 13208374 |
| Find References about AI | 17/7/2014 | R.M.D.Nilan Jayawardhana - 13208374 |
| Making rules and conditions | 14/6/2014 | All Members |
| Finalizing Meeting reports & Interim Report/Final report | 13/7/2014  28/7/2014 | All Members |
| Thilina Warnakulasooriya - 13208380 | Hand Checking coding (Three of a king, two pair, one pair, high card) | 10/7/2014 | R.M.D.Nilan Jayawardhana - 13208374 |
| Networking | 25/7/2014 |  |
| Making rules and conditions | 14/6/2014 | All Members |
| Priority checking coding | 21/7/2014 |  |
| Uditha S. Siriwardhana - 13208372 | Sketching Interface | 11/6/2014 | P.M.C.Ayesha Panditharathne - 13208376 |
| Shuffling coding | 19/6/2014 | G.K Eranjith Udayanga – 13208198 |
| Networking | 25/7/2014 |  |
| Priority checking coding | 21/7/2014 |  |
| P.M.C.Ayesha Panditharathne - 13208376 | Error Handling | 20/7/2014 |  |
| Making rules and conditions | 14/6/2014 | All Members |
| Taking notes and Important things |  | Uditha S. Siriwardhana - 13208372 |
| Coding AI part | 21/7/2014 |  |
| R.M.D.Nilan Jayawardhana - 13208374 | Hand Checking coding(Royal flush, straight flush, four of a kind) | 10/7/2014 | K.M.Suyama D. Dharmasena – 13208369 |
| Distribution of cards coding | 16/7/2014 | Thilina Warnakulasooriya - 13208380 |
| Coding AI part | 21/7/2014 |  |
| Combine the codes | 26/7/2014 |  |
| G.K Eranjith Udayanga – 13208198 | GUI designing | 22/7/2014 |  |
| Find References about Error handling | 15/7/2014 | Thilina Warnakulasooriya - 13208380 |
| Shuffling coding | 19/6/2014 | Uditha S. Siriwardhana - 13208372 |
| Priority checking coding | 21/7/2014 |  |

**Chapter 5**

**CONCLUSION**

We started the project early with the great team spirit. First we learn how to play five card draw poker games and then started coding. We planned our project and assigned duties to each and every member in the group.

We have done a huge job even in the exam season, because we had Mobile Application Development exam and Modern Operating System exams. We got a break for some days to study for the exams and start coding again.

We have to finish Networking and GUI parts mainly and connect to the server system. We are planning to finish the project before 5days and start to write the final report.