SOFTWARE REQUIREMENT SPECIFICATION
FOR

Building a Server-Client Architecture to Play Card Games, BLÖFLÜ PİŞTİ, from Mobile Devices

Table of Contents

1. Introduction

This document is a software requirement specification for ***Building Server-Client Architecture to Play Card Games, BLÖFLÜ PİŞTİ, from Mobile Devices.*** To introduce you the document well, we will first give problem definition of our project and then the purpose of this SRS, scope of this document, then follow Literature survey on the existing products, and potential users of the our final product and we will state the descriptions and abbreviations that are used in our SRS report and References which state all the documents and sources that are used in our SRS report and give an overview which explain how the SRS is organized and what the report contains. And after finishing the Introduction we supply **an Overall description** of this ***Building Server-Client Architecture to Play Card Games, BLÖFLÜ PİŞTİ, from Mobile Devices***. After completing a general introduction to the system, we will address **Specific requirements** for it. In this last part of this document, we will basically address functional requirements as well as nonfunctional requirements. Following **Data Model and Description** and then **Behavioral Model and Description**, **Planning** and finally there is a **Conclusion** of the project ***Building Server-Client Architecture to Play Card Games, BLÖFLÜ PİŞTİ, from Mobile Devices***

1.1 Problem Definition

We will build a game server where clients can connect and play a simple card game, ***BLÖFLÜ PİŞTİ***, with each other or with an artificial agent running on the server. We will build:

1. A game server that keeps all the information about the users and ongoing games, manages all the communication between clients (dealing cards, sending moves, etc).

2. The server will be implementing a web service. It is specified a well defined protocol for the communications between clients and the server. There will be no peer to peer communication between clients.
3. A client program with a cute user interface that runs in a mobile platform, i.e. on Android systems, Windows Mobile, or I-Phone. In our case Android system.
4. An agent that runs in the server and acts as a client. (The communications between the agent and the server will use the same protocol) The card game is "blöflü pişti". It is a simple but nontrivial two player card game of luck and skill, played with a standard deck of 52 cards.

1.2 Purpose

The purpose of this document is to give a complete description of the behavior of the ***Building Server-Client Architecture to Play Card Games, BLÖFLÜ PİŞTİ, from Mobile Devices*** to be developed. This document is intended to establish the basis for agreement between customers and the suppliers on what the software product is to do, decrease the effort needed for development, provide a basis for validation and verification. What we are going to address will basically constitute a basis for functionality, external interfaces, performance, attributes and the design constraints of the system. This document is better suited for the customers, users, and developers.

1.3 Scope

Software product we will introduce is ***Building Server-Client Architecture to Play Card Games, BLÖFLÜ PİŞTİ, from Mobile Devices*** As it can be understood from its name, it is intended to be automated system which will help user – anyone- play card via the Mobile Devices. Users connecting to the server should be able to:

- Register and login
- open a table
- join an opened game
- play the game or choose an AI agent on the server to play instead
- see their statistics

1.4 User and Literature Survey

User in this application would be anyone who connects to WEB and owes a Mobile Device. There is no limitation to sign up the system. Everybody can be a member of the system and open his account and start the game against Computer or anyone else. ***BLÖFLÜ PİŞTİ*** is played with 52 deck cards and with 4 people but there is no problem if there are less than four people because system takes place in turn of lack one. In our case the game will be played with 2 people.

"Pişti" game is a well known card game in Turkey. In mobile world, there are several game applications in this class. Android will be our development environment, so we searched and scanned Android Market for this game. In the market, there exist only samples of which allows player to play with computer AI, online multiplayer gaming feature does not exist. In our project, the game is to be ***BLÖFLÜ PİŞTİ*** in addition online player versus player feature will be implemented. Android market is lacking of this type of game, but iOS appStore has several good samples of a standard Pişti game, which have also online support and AI bots.

1.5 Definitions and Abbreviations

SRS: Software Requirements Specification
AI bot/agent : Artificial intelligent game client bot/agent.

1.6 References

[1] IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications

[2] http://market.android.

[3] http://itunes.apple.com/us/app/pisti-ii/

[4] http://creately.com

[5] http://wikipedia.org

1.7 Overview

This SRS consists of seven parts. In first part we gave general information about ***Building Server-Client Architecture to Play Card Games, BLÖFLÜ PİŞTİ, from Mobile Devices*** to the reader Following section of this document will focus on describing the system in terms of product perspective, product functions, user characteristics , assumptions and dependencies .In the third section , we will address specific requirements of the system, which will enclose external interface requirements, functional requirements of the system, performance requirements, and other requirements. In the forth section it will be mentioned about **Data Model and Description**, and in fifth **Behavioral Model and Description** and then Planning and finally there is a **Conclusion** for the SRS of ***Building Server-Client Architecture to Play Card Games, BLÖFLÜ PİŞTİ, from Mobile Devices*** project.

2. Overall Description

In this section, we will provide background information about specific requirements of the system. Although we will not describe them in detail, we will introduce general factors that affect the product.

2.1 Product Perspective

The product will be for the use of everybody eventually. System will be integrated with an online web site so user can log in the system via this web site and can play card games with multi-users environment system or via the computer. This game is played with 52 card deck and among 2 people. Our system is a product of artificial intelligence. System can compute the most powerful attack and think the sooner move. Our system, ***Building Server-Client Architecture to Play Card Games, BLÖFLÜ PİŞTİ, from Mobile Devices*** will be a part of a platform which accept members and these members can log in the system with their username and password then they reach their account and they see the games being played by other users and they make a room for a new game and the creator of a new room has the right of choosing the players which send an request. Creator can choose anyone of these requestors or from computer itself.

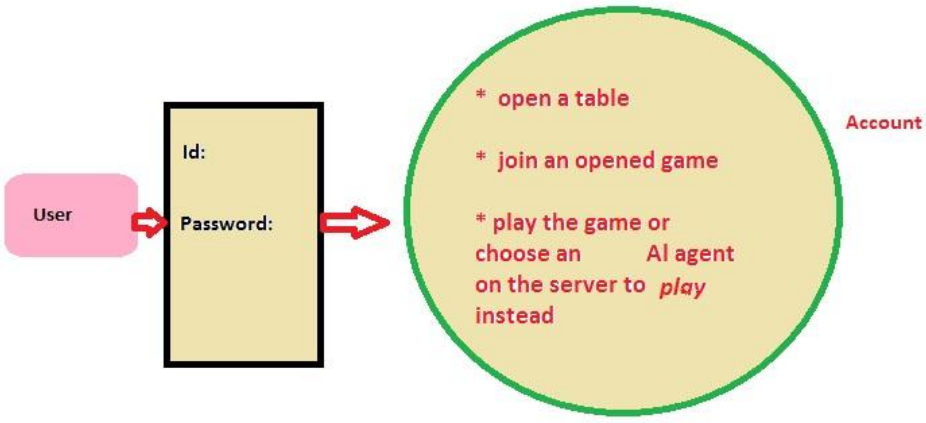


Figure 2.1.1 , Server-Client Architecture to Play Card Games, BLOFLU PISTI, from Mobile Devices

2.2 Product Functions

Building Server-Client Architecture to Play Card Games, BLÖFLÜ PİŞTİ, from Mobile Devices will require an id and a password from user. It means that before the first game users must sign in the system and system holds all the information about users: Id, Password, and Last Log in, Log in history, Log in Duration, Statistics, Games, and Actions. There are three main functions in Developed User Interface

1. Open a table
2. Join an opened game
3. Play the game or choose an Al agent on the server to play instead

And also

4. See the game history, statistics and actions, grades
5. Arrange your personal information.

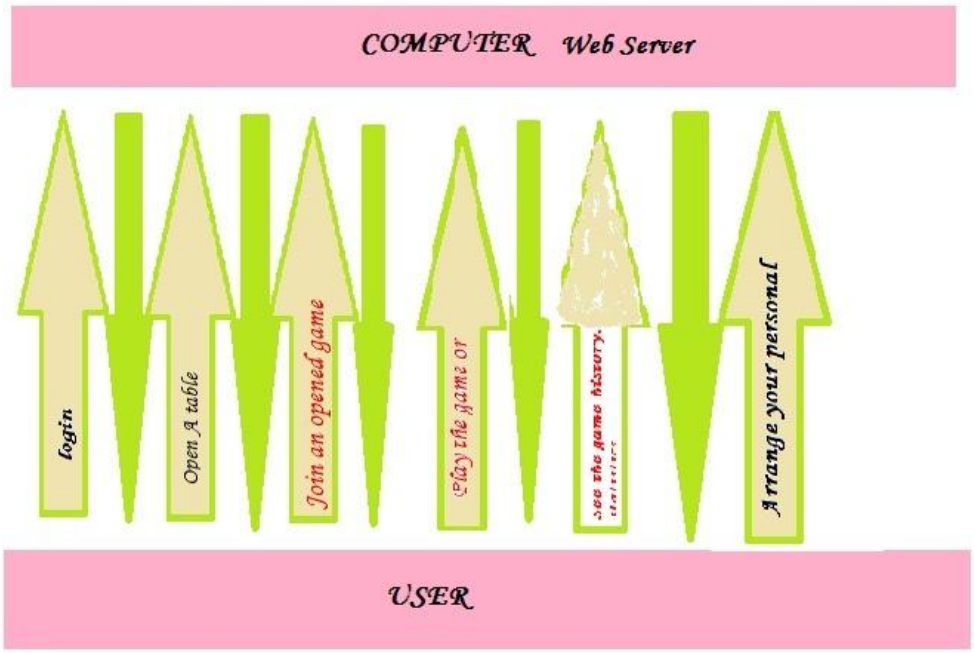


Figure 2.2.1 User Functions And Web Server for Game

2.3 Constraints, Assumptions and Dependencies

A) Regulatory policies:

Developers of the new system should consider the fact that the software will work on a web server and new game can be added on the platform but artificial intelligence cannot be ignored namely user can play the games with computer in absence of any number of players and system can automatically answer the user after some amount of time and also system take place in turn of the player who leave the game suddenly.

B) Standards, Hardware limitations, Safety Considerations:

Developers must obey the rules of the game and platform. Platform must be designed with Artificial intelligence and system take a place in turn of any players and user connect internet with mobile devices and system can detect the device and answer the request and also system cannot let the intruders to log in the system.

C) Assumptions and Dependencies

Our project will run on Android based mobile devices and user start the program and then automatically connect to internet and user will choice to start the game.

Android is an operating system for mobile devices such as smartphones and tablet computers. It is developed by the Open Handset Alliance led by Google.

3. Specific Requirements

In this section and its subsections, it will be explained all the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements.

3.1 Interface Requirements

Our system will look like finally during a player playing the game as in the figure. User can see last two cards over the ground besides to his hand and also it is written his name and grade on the screen. And there is a small picture, left bottom of the board player's name written, shows in which times it is being played. There is no admin log in the system because there is no need to change the personal data and make an effect on the system. System will run automatically and an admin ca login to system like anyone and can get all the information about the system. This is essential because of plausibility of the system.

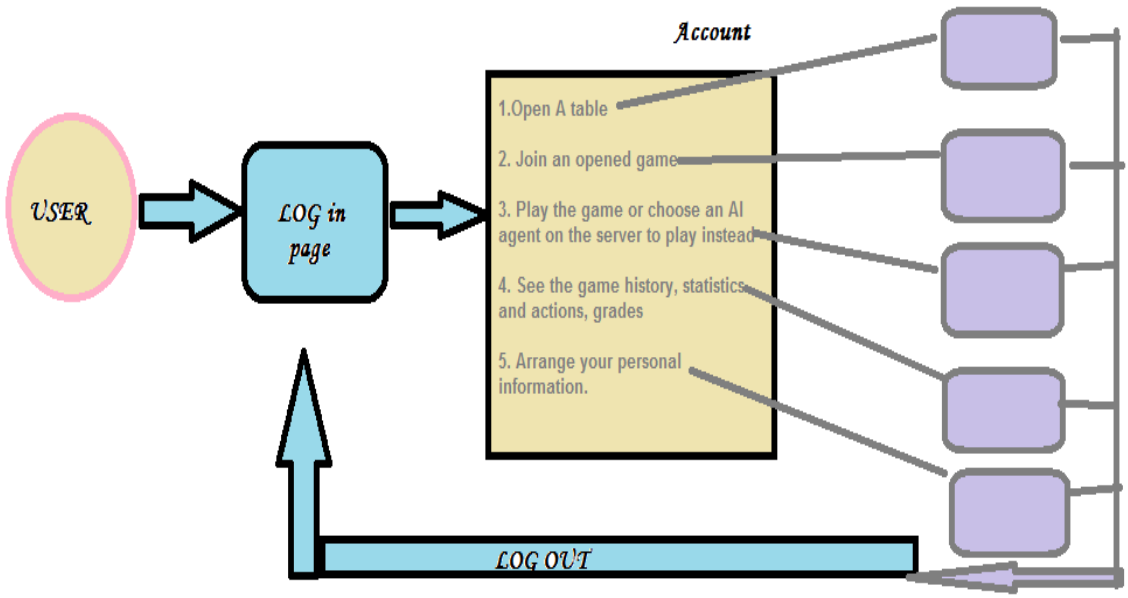


Figure3.1.1. User Interface

Figure3.1.2 Game screenshot player of "tekin "

Game interface will be similar to this one, but there will be only 2 players.

3.2 Functional Requirements

System has some special functions which do some kind of things, accept member, login members etc. Now we will explain these functions:

3.2.1 Log in to the system

If user is a member of the system, the username and password will match over the one of database of system. If not, the system won't let him to login and ask the user whether he/she is a member/forgets his/her password/username, wants to be a member. Then, if match then user accesses to his/her account.

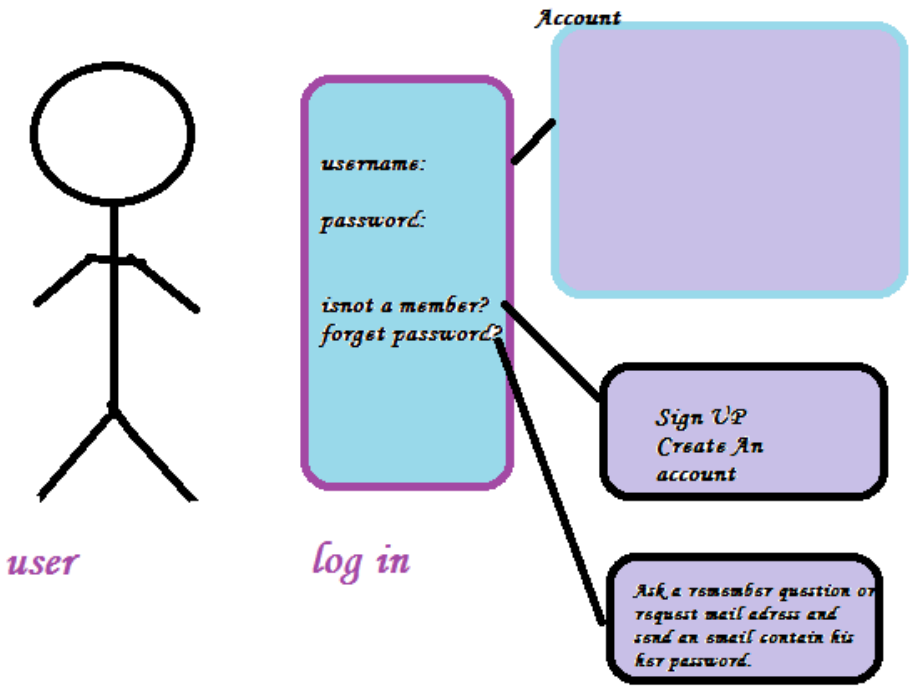


Figure3.2.1 Log in system

Basic Data Flow:

- *Click sign-up*
- *Provide a username*
- *Wait for username check*
- *If it is existing provide another one*
- *Provide a password*
- *Provide an e-mail address*
- *Register completed*

Alternative Data Flow:

- *Click sign-in*
- *Enter username*
- *Enter password*
- *Password check successful*
- *Connect the game interface*

Alternative Data Flow:

- *Click sign-in*
- *Enter username*
- *Enter password*
- *Password is wrong*
- *Click "I forgot my password"*
- *Check your e-mail*
- *Change your password*

3.2.2 Arranging personal information

User can arrange his/her personal data and can change his/her password or registered e-mail address.

3.2.3 Opening a table

User can open a table and can accept people to this table and game will be started after user authenticates.

Basic Data Flow:

- *Click open table*
- *Specify score limit*
- *Wait for another player*

Alternative Data Flow:

- *Click open table*
- *Specify score limit*
- *Play with AI Bot*

3.2.4 Joining an opened game

User can join an opened game and play the game with others. If this game is created by any other player, user must wait for an authentication from creator.

Basic Data Flow:

- *Click one of the empty slots*
- *Wait for creator's approval*
- *Resume the game process*

3.2.5 Playing the game or choosing an AI agent on the server

User can play a game with computer if s/he prefers. For example, just for learning the game, this way can be chosen or if there is not suitable rival to play the game with. The AI agent will play just like a real person.

Basic Data Flow:

- *Click play with the AI*
- *Enter the game*

Alternative Data Flow:

- *An existing game was on air.*
- *One player is disconnected somehow*
- *AI takes the place of disconnected player*
- *Resume the game*

3.2.6 See Rankings

Players will be able to see the past statistics of his/her account and the best scores among the past games. Rankings over other users can also be seen from the logged user.

3.2.7 Game History

User will be able to see individual rankings and his/her cumulative results. In addition user may see pair wise charts with a pre-specified opponent. Hall of fame part will also be available for users to get involved in rivalry. User also can see "performance versus time" chart of his/her past games.

3.3 Non-functional Requirements

3.3.1 Performance Requirements

This is a game played by two people. Therefore in the system there must be two people at first glance. However it is not a problem for our system because our system is a product of the artificial intelligence and system take place in turn of the absent one and if there is a problem about any player like leaving suddenly then the system take place and play the absent one and always do the most powerful attack.

3.3.2 Design Constraints

This system is designed for any user owing a Mobile Device which uses Android operating system. And therefore it will be designed over the platform java. There is no hardware problem because system will work only Mobile Devices which has Android operating system.

3.3.3 How to Play Blöflü Pişti

Blöflü Pişti is an arcade card game that is played via 52-deck cards.

To start the game, 3 closed 1 open cards are placed at the table. Each player is dealt 4 cards. The first player begins the game by throwing any card.

Players throw one card among the cards in his/her hand to the top of the card/cards on the table. The player who have the same card with the card on the top, throws this card and surpass all the cards on the table by this throw. Jack is the card that can be thrown every turn and that dominates and makes able to surpass all the cards on the table.

The card that is thrown into the single card standing on the table results in pişti. This card may be thrown closed and bluff can be made in that closed throw. Opponent player may accept the pişti and it is regarded as a standard pişti or with suspect of the bluff s/he may demand to open the closed card to see whether it is bluff or not. If opened card is real pişti, owner of pişti gets 20 points and takes two cards on the table, if it is bluff opponent gets 10 points and nobody takes the cards and open cards is kept still. Game is resumed by the turn of the next player. Game continues up to the end of cards in both players' hands and each player is dealt again four cards. If all cards are used up, scoring will be made. Scoring is done in the way as follows; firstly, the cards each players surpassed is counted. The point value of each pişti is 10 points, apart from that aces and jacks are counted as 1 point, club two is 2 points, diamond dix is 3 points. After that the player with more cards will get three more points, then game will continue with a new draw until one of the player reaches the limit score.

4. Data Model and Description

In this section, the information of data models of the software is mentioned.

4.1 Data Description

There will be 5 types of data objects in the system namely: user object, game state object, game move object, server object, statistics object.

4.1.1 Data Objects

- ✓ **User Object**: This object will keep the user information. User id and user name will unique, and password information will be kept encrypted. The data of this object will be kept in database.

- ✓ **Game State Object**: This object will keep all the information about an ongoing game. Game status stands for whether the game is running or waiting for a client user. Also, it keeps the information about both players. Cards of the players and the which player has the turn information is kept in this object.

- ✓ **Game Move Object:** This object keeps the information of moves in the game. When a player throws a card, this object will keep that information**.**

- ✓ **Server Object:** Server object will hold the information of the list of online players and list of games.

- ✓ **Statistics Object:** Statistics object will hold how the related user played in the history. How much bluff s/he made, how rate s/he succeeded and at what rate the user won the games. The data of this object will be kept in the database.

4.1.2 Relationships

Since the server is always running, the server object always keeps the track of ongoing games and online players. When a user connects to the server, the user object is created. User object and server object are related with login and open/close table actions.

User and game move objects are related with move action. When the user throws a card, a move object is created and sent it to the server.

Since every move is changing the game state, the game state and game move objects are related in this manner.

After every game move, statistical information is recorded on the server. So, game move and statistics objects are related in this manner.

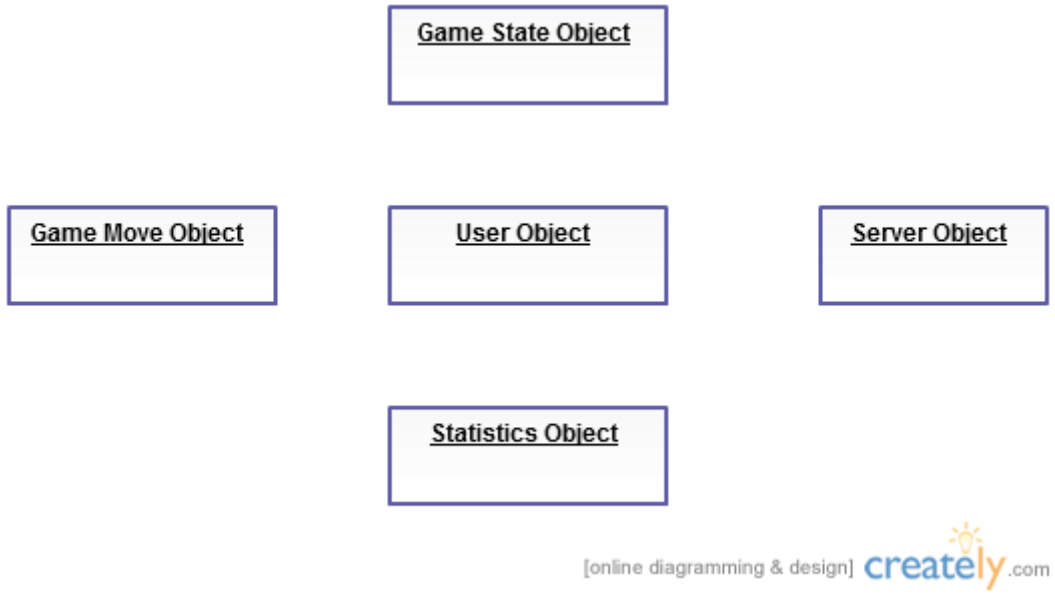


Figure 4.1.2 Simple Object Diagram

4.1.3 Complete Data Model

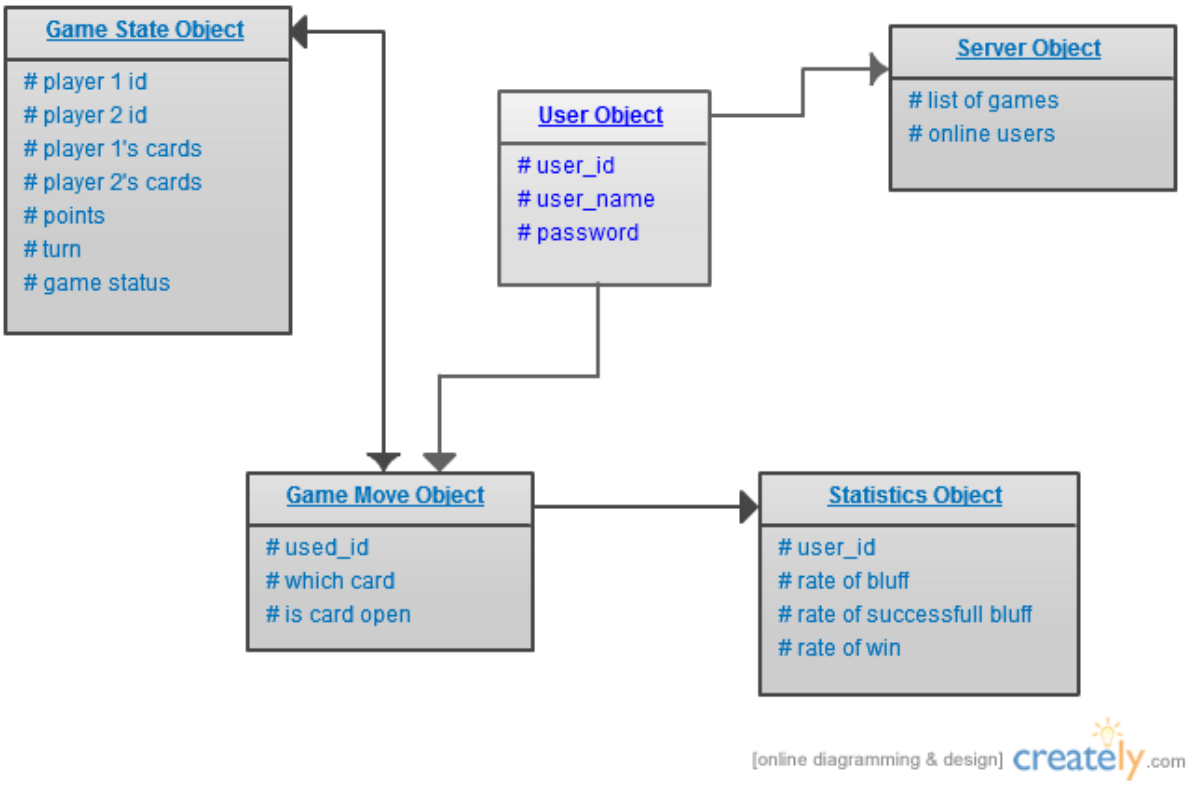
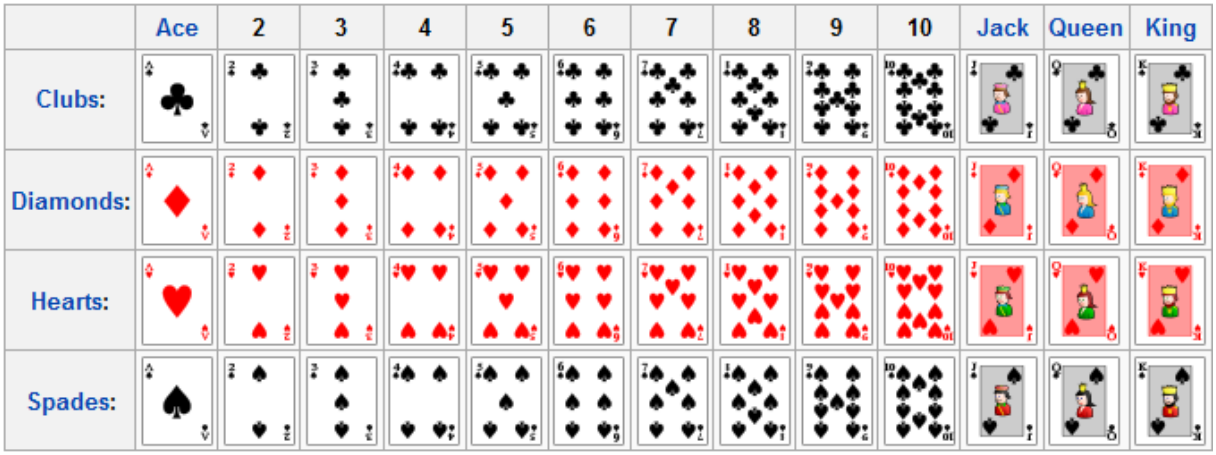


Figure 4.1.3 Complete Data Model

4.1.4 Data Dictionary

- **52 deck cards:** usually known as the French deck, includes thirteen ranks of each of the four French suits, clubs (♣), diamonds (♦), hearts (♥) and spades (♠), with reversible Rouennais "court" or face cards.



(Figure adapted from en.wikipedia.org)

Bluff: /bləf/ An attempt to deceive someone into believing that one can or will do something.

In the card game Blöflü Pişti, the action of throwing card closed to attempt Pişti.

5. Behavioral Model and Description

In this chapter, behavioral description of the system will be explained.

5.1 Description for software behavior

In our project, there are 3 major parts that are server, application client and the database.

Server manages:

- Communication between players
- Recording statistics into the database
- Login and logout operations
- Running the AI agent when necessary

Android Client Application:

- Provides the cute user interface
- Running the game
- Gets inputs from user via the interface

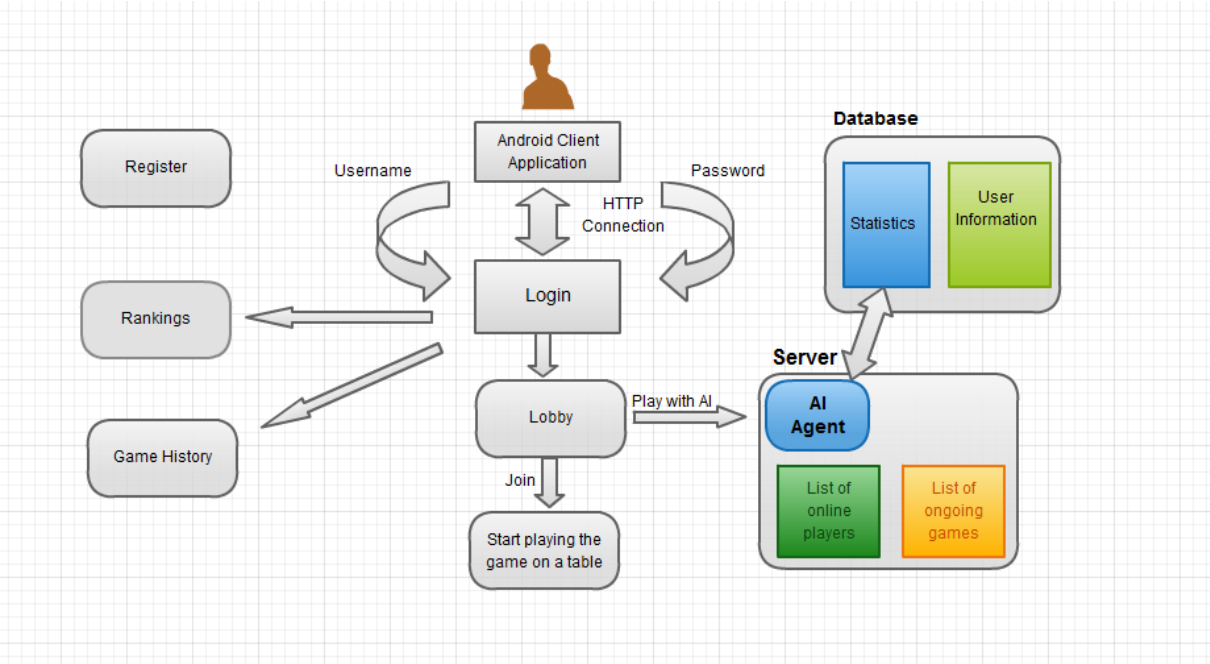
Database:

- Keeps users' all information
- Keeps statistics about each user for helping the AI agent
- Holds ranking tables

When a user connects to the system, server will check user's username and password from the database. If authentication is done, the user reaches to the lobby. In the lobby, existing game tables are shown, and the user can join one of them. Also, user can prefer creating his/her own table or playing with the AI agent. During the game, the user might want to leave the room somehow. In that case, an alert pops-out saying "If you leave, you will lose X points" and waits from the user to make choice. X can differ according to game status. If the user chooses to leave the room, the AI agent will replace him/her.

Except from playing the game, the user can see the ranking table or his/her past game statistics. S/he can compare with other players.

5.2 State Transition Diagrams



6. Planning

6.1 Team Structure

A.Emirhan ÖZDEMİR – Researcher, Game Module Developer, Interface Developer

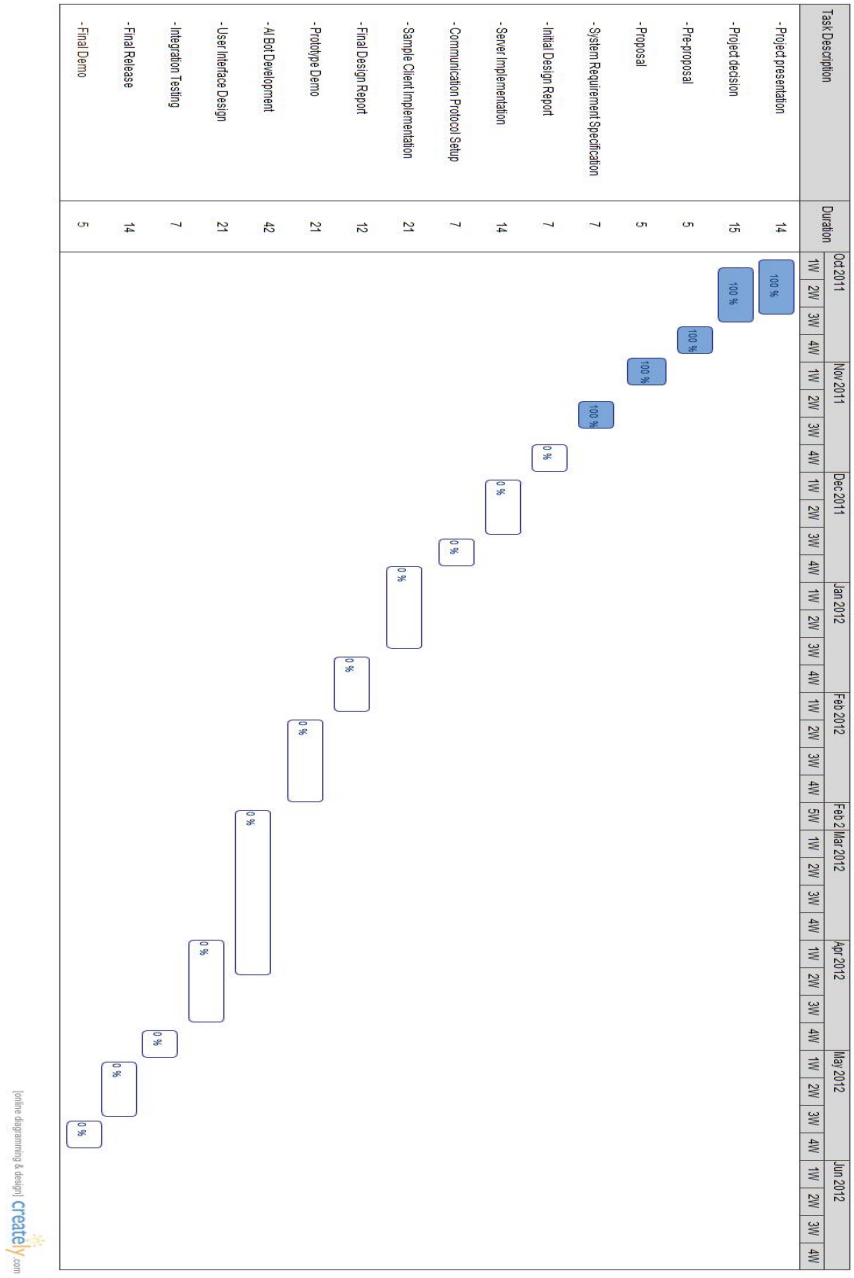
Cuma Tekin TOPUZ – Researcher, Server and Infrastructure Developer, Database Developer

Hamza YILMAZ – Researcher, Server and Infrastructure Developer, AI Developer

Utku SAKİL – Researcher, Game Module Developer, AI Developer

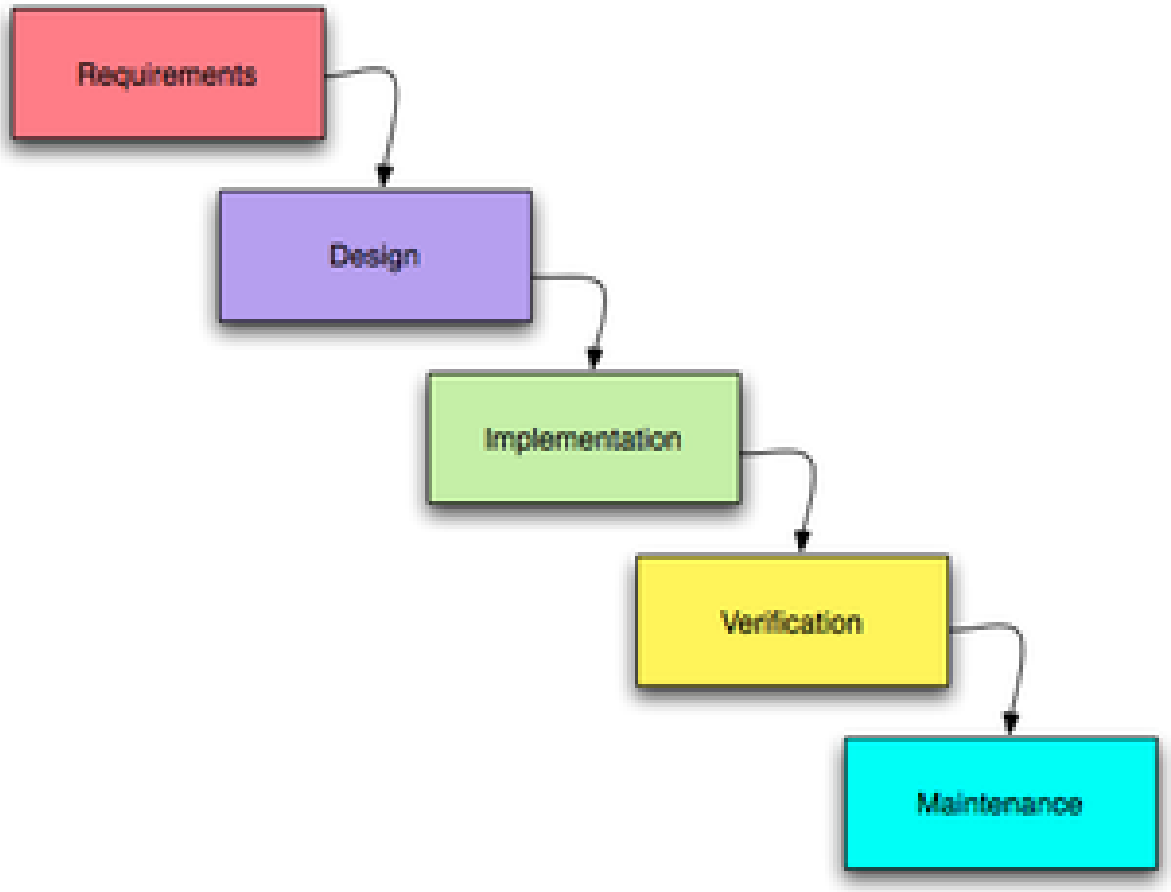
Since project is composed of several parts, we separated the labor as follows: A.Emirhan Özdemir and Utku Sakil are responsible for game module development and Cuma Tekin Topuz and Hamza Yılmaz are responsible for server and infrastructure development. Also, Hamza Yılmaz and Utku Sakil are responsible for AI development. Apart from all these, Cuma Tekin Topuz is the database developer and A.Emirhan Özdemir is the interface developer of the group. Moreover, all the members has to research inquiries in different aspects. We do not have a project leader, because we make our decisions by collaboration and whole team work. We are in touch almost everyday and communicate about the whole progress, tasks and next steps to do.

6.2 Estimation (Basic Schedule)



6.3 Process Model

Although waterfall process model is a bit legacy for today's software development processes, waterfall process model will be the mainstream process model of our project due to the advantage of process sequence and suitability for the project. Waterfall process model has the following sequence in terms of the phases to be in progress; requirements, design, implementation, verification, maintenance.



7. Conclusion

This document is to clearly show the requirements of the project carried out by e-Limon, which is a client-server card game infrastructure project on mobile Android devices. The document includes multiple aspects of project so any third person can understand the project without any pre-knowledge.

Firstly, introduction and brief project description are presented. After that, overall description and product functionality are explained. Then, requirements of client-server architecture card game are examined and demonstrated. After that data and behavioral models are introduced. Lastly, project methodology and schedule are demonstrated.

This report was very useful to focus our projects requirements and aspects. We expect good results for this project in terms of usability, availability, scalability, reliability and functionality.