Project Plan

Group C

Jiaqi NI (2217774)

Jianfei Feng (2204529)

Kalina Petrova (2221667)

Zhang Wen (2351420)

Liu Menchuan (2151655)

Lin Jiefan (2617269)

Class: Ei7S

Mentor: Li Li

Version: third version

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Status | Date | Description |
| 1.0 | 1st version | 08.09.2016 | Draft version of the project plan. |
| 2.0 | 2nd version | 13.09.2016 | Changes to the layout.   1. The font is changed to Calibri. 2. Some other changes to make the pages looks more structured. |
| 2.1 | 2nd version revised | 16.09.2016 | Changes in the Timetable and phasing part.   1. New timetables and phasing diagrams are added. |
| 2.2 | 2nd version revised | 17.09.2016 | Changes in the project statement part.   1. More details about the current situation. 2. The target group is added.   Changes in the project risk part.   1. More possible risks are added. |
| 3.0 | 3rd version | 21.09.2016 | Changes in the timetable and pashing part.   1. New timetables and phasing diagrams are added.   A work distribution table is created.   1. A work distribution table is added to 4th chapter – organization part. |
| 4.0 | Final version | 03.10.2016 | An expected mark table is added. |
|  |  |  |  |

Content

[Content 2](#_Toc461017759)

[1. Introduction 3](#_Toc461017760)

[2. Project Statement 4](#_Toc461017761)

[Formal client 4](#_Toc461017762)

[Project leader 4](#_Toc461017763)

[Current situation 4](#_Toc461017764)

[Project justification 4](#_Toc461017765)

[Project products 4](#_Toc461017766)

[Project deliverables and non-deliverables 4](#_Toc461017767)

[Project constraints 4](#_Toc461017768)

[Project Risks 4](#_Toc461017769)

[3. Project phasing. 6](#_Toc461017770)

[Phasing and Milestones 6](#_Toc461017771)

[Timetable 7](#_Toc461017772)

[4. Resource management. 8](#_Toc461017773)

[Budget 8](#_Toc461017774)

[Skills 8](#_Toc461017775)

[Quality 8](#_Toc461017776)

[Information 8](#_Toc461017777)

[Time 8](#_Toc461017778)

[Organization 8](#_Toc461017779)

[5. Research plan 9](#_Toc461017780)

# Introduction

“Fighting the landlord” (Doudizhu) is a card game in the genre of shedding and gambling. It is one of the most popular card games played in China, and it is quite simple card game but hard to be a master. To play this card game well, it really requires player should have strategic thinking as well as carefully planned execution.

This game requires at least 3 players, and the goal of this game is to be the first to get rid of all your cards. In this game, players will be divided into 2 groups. One group will contain 2 players which called “farmer”, and the rest of the one player will be divided into another group which called “landlord”. However, players will never know which groups they will participate. To setup the game, the dealer will shuffle all cards and allowed the left side player turn one card over from middle side of all cards, then the player should put the card back.

Besides, as the rules mentioned before, there are still a lot of other rules have not given yet. In this project, our group supposed to make this card game based on these rules.

This document mainly describes how this project will be built. Furthermore, we will describe more details about how we will achieve it step by step.

# Project Statement

## Formal client

In this project, the formal client is Lili.

## Project leader

For this project, the group decided to select Jiaqi Ni as the project leader. Due to his great communication and organization skills, the team members concluded he was the best candidate to take the position. He is currently a fourth-year student at Fontys University of Applied Sciences.

## Current situation

The project requires a server-client based application, the topic of distributed system is nowadays a hot item and will become even more important. As practice reason, The project is required a server-client based application. The topic of distributed systems is nowadays a hot item and will become even more important. One of the most popular muti-player card game which is called “Dou dizhu“, it‘s well-known in China. During this project we are going to implement this game. As we known Chinese culture attracts the most attention. Some of Chinese card games are brought into global market for example “sanguosha“ (englishsanguosha.com). The game has begun to reach an international audience after players began translating the game into the English language and posting these translations on blogs and forums. Site visit statistics from one of these blogs showed that readers outside of China come primarily from Singapore, the United States of America, Canada, Australia, United Kingdom, Malaysia, and New Zealand. “Dou dizhu“ is a fantastic and addictive game and also should be known by the people outside of China. But it‘s rarely found “Dou dizhu“ game on the website. That‘s one of the reasons why we want to implement and suggest such wonderful game to people. The rule is not so complex, the people who have played any card game can learn it, which means our target group can be anyone who likes card games.

## Target group

Everyone who likes card games.

## Project justification

Distributed systems are a highly popular topic and one of the most common types distributed systems are multiplayer games. With a vast variety or genres and endless possibilities, the team decided that it’s a good opportunity to employ their imagination.

## Project products

The final product of this project is a fully functional massive multiplayer online game with as many features as possible so that the players would have an immersive and engaging experience.

## Project deliverables and non-deliverables

|  |  |
| --- | --- |
| **Deliverables** | **Non-deliverables** |
| **Source code** | **Intermediate documentation** |
| **Design documents** |  |
| **Project plan** |  |
| **Demonstration and Presentation** |  |
| **Final report** |  |

## Project constraints

* Since we are required to make a distributed system using C# and WCF, the team is required to make use them.
* The team is going to make use of MySQL databases for user information.
* Unity 2D might be used in this project for making the cards and animation.

## Project Risks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk** | **Probability** | **Impact** | **Risk Prevention** | **Alternative Scenario** |
| **Insufficient communication** | Medium | Medium | Make better agenda for every meeting. | Hold additional meetings. |
| **Unable to finish requirements** | Low | High | In the beginning, make sure the MSCO is reasonable and doable. | Discuss with the teacher and ask for more time to finish. |
| **Equipment failure.** | Low | High | Check schedule every week | Revise the schedule and make use of personal free time. |
| **Data loss.** | Low | High | Make backup. | Recreate the lost information if there is no such backup. |
| **Falling behind the schedule** | Low | Intermediate | Check schedule every week. | Revise the schedule and make use of personal free time. |

* Risk – refers to the type of risk we could encounter
* Probability – refers to what is the chance of us encountering such a risk (Low, Medium, High)
* Impact – refers to how big is the impact of our project is such a risk occurs
* Risk prevention – refers to our idea how to prevent such risks from happening
* Alternative scenario – refers to how we are going to deal if such a risk does happen to occur in the end

# Project phasing.

## Phasing and Milestones

Since we are working with an incremental planning, we should decide how large will the iterations/cycles be (weeks, a month, etc.) we purpose 2 cycles, in the first cycle, we will implement the basic most important functions, furthermore, in the second cycle, we will implement some extra features.

The project will be divided into 4 main phases:

* Initiation phase – during this phase the team’s main focus will be the defining of the objectives, the scope, purpose, and deliverables of the project.
* Analysis phase - during this phase the team’s main focus will be the defining of the user requirements and functionalities of this program.
* Monitored control – This will be the main phase of the development of the product. This phase is split into 2 cycles and each cycle is split into 3 phases itself:
  + Design – the team will design how the given tasks should be handled.
  + Development – the team tries to implement the resulted designs.
  + Testing – the team will test whether the new implementations work correctly.
* Deployment – During this phase, the team will mainly focus on finalizing the system and making sure everything is working properly before it is presented as a product.



Create Test plan

Implement additional functionalities

Perform needed tests

Perform needed tests

Create Test plan

M4

M5 

Deployment

Finalize application

Prepare presentation

Present

Update database

Revise class diagram

2nd Cycle

M3 

Create URS document

Implement important functionalities

1st Cycle

Initiation

Analysis

Create Project plan

Gather requirements

Determine functionalities

Setup environment

First version of class diagram

M1

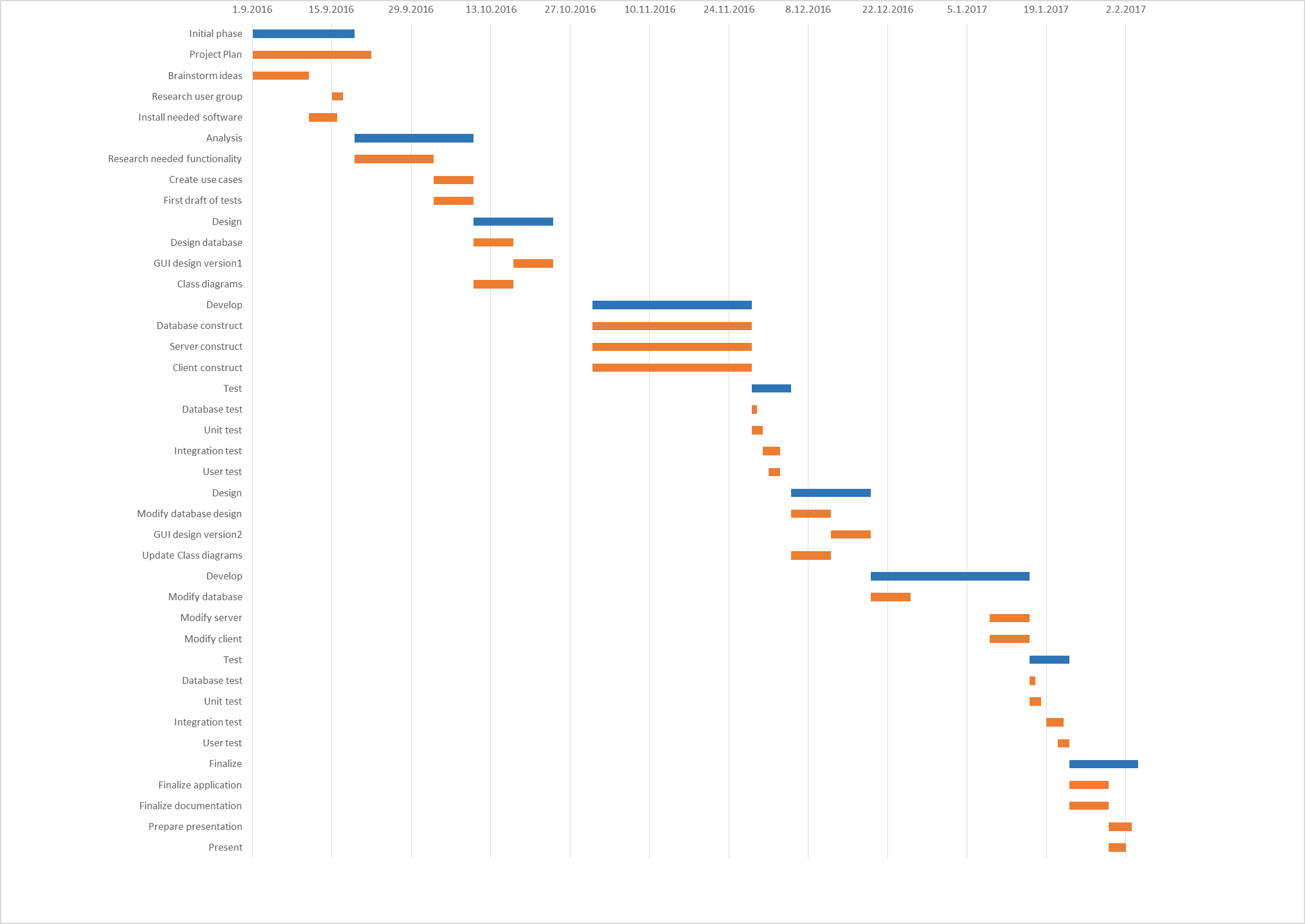
Design database

M2

## Milestones

|  |  |
| --- | --- |
|  | Deliverables |
| Milestone 1 | * Project plan * Division of work among the team members |
| Milestone 2 | * List of functionalities to be implemented during the cycle * URS document for the functionalities to be created |
| Milestone 3 | * First version of design document * First demo of the application with the most essential functionalities |
| Milestone 4 | * Final version of the design document * Final prototype with some additional functionalities |
| Milestone 5 | * Final product * Presentation |

|  |  |  |  |
| --- | --- | --- | --- |
| **Initial phase** | 29.8.2016 | 21 | 19.9.2016 |
| Project Plan | 1.9.2016 | 21 | 22.9.2016 |
| Brainstorm ideas | 1.9.2016 | 10 | 11.9.2016 |
| Research user group | 15.9.2016 | 2 | 17.9.2016 |
| Install needed software | 11.9.2016 | 5 | 16.9.2016 |
| **Analysis** | 19.9.2016 | 21 | 10.10.2016 |
| Research needed functionality | 19.9.2016 | 14 | 3.10.2016 |
| Create use cases | 3.10.2016 | 7 | 10.10.2016 |
| First draft of tests | 3.10.2016 | 7 | 10.10.2016 |
| **Design** | 10.10.2016 | 14 | 24.10.2016 |
| Design database | 10.10.2016 | 7 | 17.10.2016 |
| GUI design version1 | 17.10.2016 | 7 | 24.10.2016 |
| Class diagrams | 10.10.2016 | 7 | 17.10.2016 |
| **Develop** | 31.10.2016 | 28 | 28.11.2016 |
| Database construct | 31.10.2016 | 28 | 28.11.2016 |
| Server construct | 31.10.2016 | 28 | 28.11.2016 |
| Client construct | 31.10.2016 | 28 | 28.11.2016 |
| **Test** | 28.11.2016 | 7 | 5.12.2016 |
| Database test | 28.11.2016 | 1 | 29.11.2016 |
| Unit test | 28.11.2016 | 2 | 30.11.2016 |
| Integration test | 30.11.2016 | 3 | 3.12.2016 |
| User test | 1.12.2016 | 2 | 3.12.2016 |
| **Design** | 5.12.2016 | 14 | 19.12.2016 |
| Modify database design | 5.12.2016 | 7 | 12.12.2016 |
| GUI design version2 | 12.12.2016 | 7 | 19.12.2016 |
| Update Class diagrams | 5.12.2016 | 7 | 12.12.2016 |
| **Develop** | 19.12.2016 | 28 | 16.1.2017 |
| Modify database | 19.12.2016 | 7 | 26.12.2016 |
| Modify server | 9.1.2017 | 7 | 16.1.2017 |
| Modify client | 9.1.2017 | 7 | 16.1.2017 |
| **Test** | 16.1.2017 | 7 | 23.1.2017 |
| Database test | 16.1.2017 | 1 | 17.1.2017 |
| Unit test | 16.1.2017 | 2 | 18.1.2017 |
| Integration test | 19.1.2017 | 3 | 22.1.2017 |
| User test | 21.1.2017 | 2 | 23.1.2017 |
| **Finalize** | 23.1.2017 | 14 | 6.2.2017 |
| Finalize application | 23.1.2017 | 7 | 30.1.2017 |
| Finalize documentation | 23.1.2017 | 7 | 30.1.2017 |
| Prepare presentation | 30.1.2017 | 4 | 3.2.2017 |
| Present | 30.1.2017 | 3 | 2.2.2017 |



# Resource management.

## Budget

There is no cost involved in making the project since we are making an application, that does not require, web hosting, neither does it require outsourcing or a special development tool. To build the software, Visual Studio is required, but, we as developers have the educational license.

## Skills

1. Communication

For this project, communication plays a vital role. Very normal, everybody has his idea. So the role of communication is to increase understanding and iron out differences. For example, we need to have a meeting with the teacher and peer member.

2. Leadership

Achieving the goal is substantially a matter of leadership which makes everyone work together for the same goal. The other saying is team management, an effective team manager is good at managing and promoting teamwork, coordinating individual team assignment, resolving conflicts, setting goals and evaluating performance. Leadership is to encourage people to take a work with you.

3. Self-managing

Self-managing means you avoid both over-provisioning and idle resources. Everyone will be self-disciplined and interesting throughout the process

4. Innovation skills

We will have to come up with ideas and solutions about the project. Those ideas and solutions can never be seen before.

5. Risk management

Risk management is the process of identifying, evaluating, and controlling the risks of an organization. These threats, or risks, may come from a wide variety of sources, including strategic management errors, accidents, and natural disasters. Namely, focusing on how to reduce or avoid risk to decrease loss from risk. For example, for every part/module of programming, there will be an error when testing, we will need to properly handle all the errors.

6. Presentation skills

The positive, aggressive, dependable and prudent team with quick thinking and good presentation skills can be more persuasive.

## Quality

Quality means that the application created by the project group is **stable**, **easy to use** and offer the **best experience** to the client. To assure this:

**Code quality:**

* Making tests for the application behavior.
* Making a standard for writing the code (picking convention).
* Making tests for the execution of the code.

**Usability:**

* Making the user interface **easy to navigate** through (also intuitive).
* Offer simple steps of progress, with not too much information, so the user does not get confused.

**Stability:**

* For the application to be **stable** the project group will debug and test for every user input.
* Error handling will be present, to make sure if any errors occur the user knows what is happening or what he has done wrong.
* The unexpected crashing of the application will be prevented by controlling the behavior of the errors.

## Information

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Project plan** | **URS** | **Design**  **Document** | **Program code** | **Final Report** | **Final presentation** |
| **Team** | Di,M,S | Di,M,S | Di,M,S | Di,M,S | M,S | M |
| **Teacher** | DI,R,A | R | R | R,A | R,A | R,A |

**NB: Di-Discuss, M-Make, R-receive/ Read S-send, A-assessment**

## Time

1. It is estimated that this project will last 5 months. The project starts on 1st **of September** and the final product will be presented somewhere in the end of January 2017**.**
2. Time planning for project activities is as follows:

|  |  |  |
| --- | --- | --- |
| **Phase** | **Tasks** | **Duration** |
| **Initial phase** | 1. Project plan 2. URS document | 1st week - 5th week  01.09.2016 – 06.10.2016 |
| **Research Phase** | See research plan. | 2nd week - 7th week  08.09.2016 – 20.10.2016 |
| **Design phase** | 1. Design documents | 6th week - 8th week  13.10.2016 – 27.10.2016 |
| **Develop phase** | 1. Prototype | 9th week – 16th week  03.11.2016 – 22.12.2016 |
| **Testing Phase** | 1. Alpha test 2. Beta test | 14th week – 20th week  08.12.2016 – 19.01.2017 |
| **Deploying Phase** | 1. Release version 2. Final report 3. presentation | 19th week – 21st week  12.01.2017 – 31.01.2017 |

## Organization

Client: Li Li

Representative

Meeting secretary: All members

Project leader: Jiaqi Ni

Designers: All members

Developers: All members

Testers: All members

Tutor: Li Li

**Work distribution table:**

|  |  |  |  |
| --- | --- | --- | --- |
| Activity | Main principle | Vice principle | Other helpers |
| Meeting organizer | Jiaqi Ni | / | / |
| Meeting chairman , secretary, facilitator. | All team members | | |
| Project plan | Kalina Petrova | Jiaqi Ni | Other team members |
| URS | Mengchuan Liu | Wen Zhang, Jianfei Feng | Other team menbers |
| Program architecture | Jiaqi Ni | Kalina Petrova, Wen Zhang |  |
| Design documents | Kalina Petrova | Mengchuan Liu | Other team members |
| Databse constructure | Wen Zhang | Kalina Petrova | Other team members |
| Project implementation | Jiaqi Ni | Kalina Petrova, Wen Zhang | Other team members |
| UI designer | Jiefan Lin | Mengchuan Liu, Jianfei Feng | Other team members |
| Tests plan | Mengchuan Liu | Jiefan Lin, Jianfei Feng | Other team members |
| Tests executer | Jianfei Feng | Jiefan Lin | Other team members |
| Final report | Kalina Pertova | Jiaqi Ni | Other team members |
| Final presenataion | All team members | | |

**NB: all the deliverables will be reviewed by all team members before it is delivered to the Client.**

**Expected mark:**

|  |  |
| --- | --- |
| Jiaqi Ni | 8.5 |
| Kalina Petrova | 8.5 |
| Wen Zhang | 8.5 |
| Jianfei Feng | 7.0 |
| Mengchuan Liu | 8.0 |
| Jiefan Lin | 7.5 |

# Research plan

**General introduction:**

In this project, the main task is making a distributed system, panties of options, like library systems, highway systems, games etc. were found as possible candidates. After several times’ discussion, a card game called “Fight the Landlord (DouDiZhu)” was chosen to be implemented in this project. Implementing a game is always a difficult task, especially for games with complicated rules.

As introduced in the previous chapters, the rules of this game is pretty complicated, which means, the first step to do is figuring out what these rules are and how these rules can be implemented. Furthermore, as previously mentioned, this is a Client-Server distributed system, which requires some special technology to achieve that. Moreover, this game is for multiple players, which means a server can handle multiple clients at the same time. Finally, the user interface, due to the fact this is a game, the user interface should better be nice and neat.

**Approach to achieve this:**

1. Search the rule of “Fight the Landlord (DouDiZhu)”.
2. Search how to implement the rules of “Fight the Landlord (DouDiZhu)”.
3. Search how to implement a simple server-client system.
4. Search how to implement a complicated server-client system.
5. Search how to implement few basic functions (shuffle, deal etc.) of “Fight the Landlord (DouDiZhu)”.
6. Search how to implement few some mandatory functions (play, get winner etc.) of “Fight the Landlord (DouDiZhu)”.
7. Search how to implement few some advanced functions (scoreboard, account manager, time remainder for playing etc.) of “Fight the Landlord (DouDiZhu)”.

**Applying the Dot Framework:**

1. Search the rule of “Fight the Landlord (DouDiZhu)”.

 GOOGLE/BING. Asking someone who knows the rules.

**Decision/result:** rules found and learned by all team members.

1. How to implement the rules of “Fight the Landlord (DouDiZhu)”.

 GOOGLE/BING.

Making a simple app which applies the rules.

test the application to determine whether it works well.

**Decision/result:** rules will either be hard coded in the program or stored in an XML file or be loaded every time the program initializes.

NB: The fact is that it is better to store the rules in an XML file since this will make the program more flexible; however, since our team is not 100% confident of achieving that, hard coded is considered as the first option.

1. How to implement a simple/complicated server-client system?

 GOOGLE/BING.

Making a simple distributed app.

test the application to determine whether it works well.

**Decision/result:** WCF (Windows Communication Foundation) will be applied to build this project.

1. How to implement functions of “Fight the Landlord (DouDiZhu)”.

 GOOGLE/BING.

Making some simple basic functions.

test the application to determine whether it works well.

**Decision/result:** As previously mentioned, the game will be developed function by function, from the most basic ones to advanced ones. Details about development cannot be determined yet since there will be too many unknown factors. Hence, only the solution path but not the actual solution is determined here.