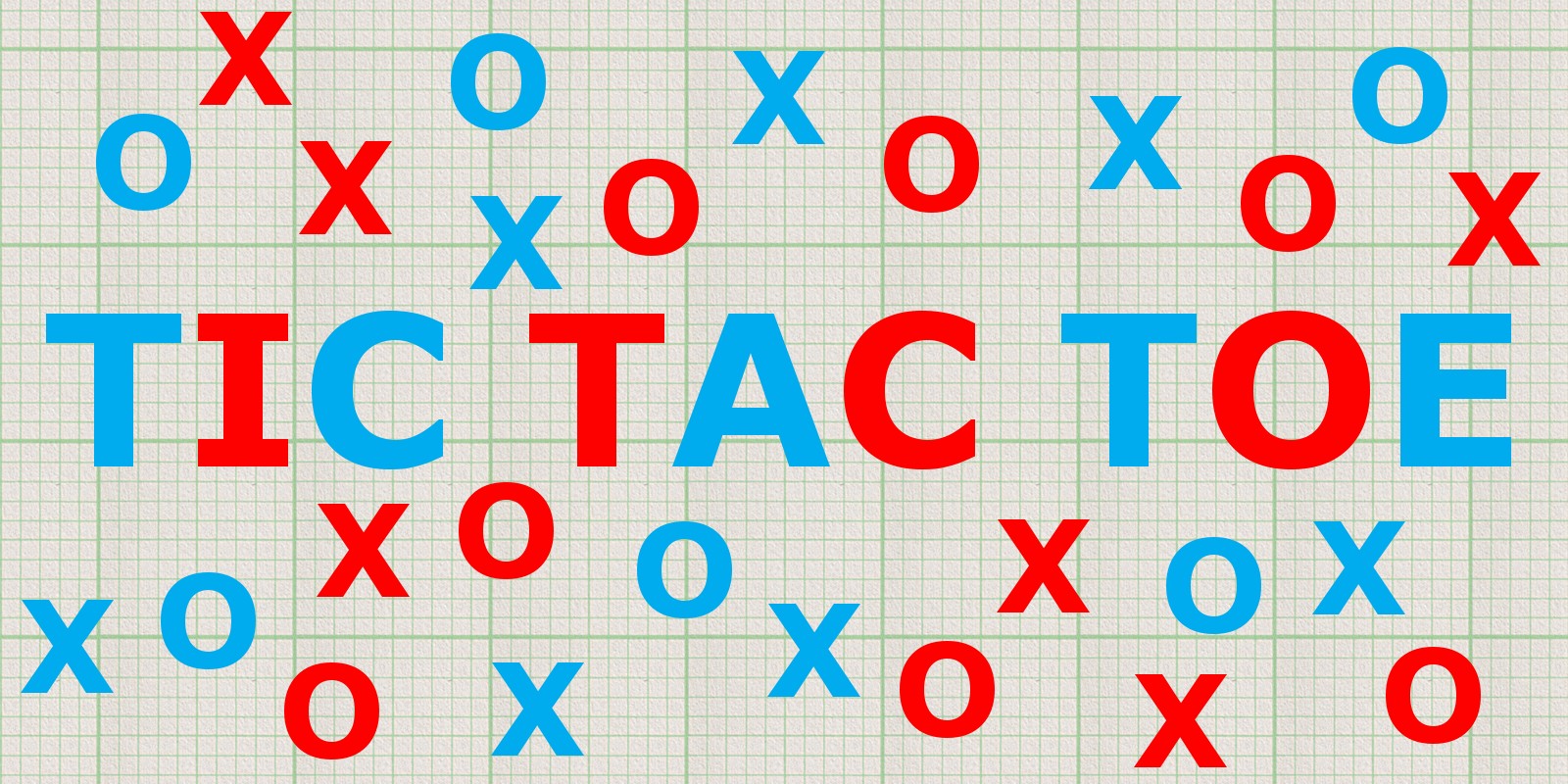
**TIC-TAC-TOE GAME**

****

|  |  |
| --- | --- |
| Submitted by: | Submitted to: |
| Aashish Nagarkoti | Mr. Sudeep Bajimaya |
| NCCID:00175036 | Module: CP |
| Roll num:02 | Softwarica college of IT and E-commerce |
| Softwarica college of IT and E-commerce |  |

TABLE OF CONTENTS

Contents

[Tables of Figures 3](#_Toc11875529)

[Keywords 3](#_Toc11875530)

[1.INTRODUCTION 4](#_Toc11875531)

[1.1 Project Intro 4](#_Toc11875532)

[1.2 Project Background 4](#_Toc11875533)

[1.3 Problem-Statement-Description 4](#_Toc11875534)

[1.4 Features 4](#_Toc11875535)

[1.5 Overview 4](#_Toc11875536)

[2. SCOPE 5](#_Toc11875538)

[2.1 Scope of Project 5](#_Toc11875539)

[2.2 Aims 5](#_Toc11875540)

[2.3 Objectives 5](#_Toc11875541)

[2.4 Limitations 5](#_Toc11875542)

[2.5 Overview 5](#_Toc11875543)

[3 DEVELOPMENT METHODOLOGY 6](#_Toc11875544)

[3.1 Description 6](#_Toc11875545)

[3.2 Design 7](#_Toc11875546)

[Why?? 8](#_Toc11875547)

[3.3 Architecture 8](#_Toc11875548)

[Why??? 9](#_Toc11875549)

[4 PROJECT PLANNING 10](#_Toc11875550)

[4.1 Work Breakdown Structure 10](#_Toc11875551)

[4.2 MILESTONES 11](#_Toc11875552)

[4.3 Gantt Chart 12](#_Toc11875553)

[5 RISK MANAGEMENT 13](#_Toc11875554)

[6 Configuration Management 17](#_Toc11875555)

[Conclusion 17](#_Toc11875556)

[Reference and Bibliography 17](#_Toc11875557)

# Tables of Figures

Figure 1: Waterfall Model…...………………………………………………………………….7

Figure 2: MVC Pattern………… ..…………………………………………………………….8

Figure 3: Three-tier Architecture ..…………………………………………………………….8

Figure 4: WBS………………..………………………………………………………………….9

Figure 5: Milestones ….……………………………………………………………………….10

Figure 6: Gantt chart-1…..……………………………………………………………………11

Figure 7: Gantt Chart-2………………………………………………………………………..12

Figure 8:Risk Management Lifecycle………………………………………………………..13

Figure 9:Likelihood table...……………………………………………………………………13

Figure 10: Consequence Table ……….…………………………………………………….13

Figure 11: Risk Assessment…..………..………………………..…………………………14

Figure 12:Configuration Management……………………………………………………...15

# Keywords

Methodology

Linear-flow

Client-server

Directories

Likelihood

Assessment

# 1.INTRODUCTION

## 1.1 Project Intro

Tic-Tac-Toe is a fun game which is made for entertainment purpose. This is a user-friendly game that can be played by any age-group. There is no any age restriction for playing this game. This game has very simple, user-friendly features, easy to operate functions. Anyone new to this game can easily adapt their selves to the gaming environment.

## 1.2 Project Background

Tic-Tac-Toe is game solely made for fun. This game has very simple user-interface, that anyone new to this game can adapt to this environment. I personally like this gam very much. This game has 3\*3 grid view for playing.

## 1.3 Problem-Statement-Description

Tic-Tac-Toe is a fun game. But in-order to win this game, each player has to think of different types of strategy. Players are required to think their next move carefully forgetting the win during the gameplay.

## 1.4 Features

Tic-Tac-Toe game has 3\*3 grid view for playing. Each user has their score-card for viewing their score throughout the gameplay. After the completion of the game, there is Reset Button in-order restart the game or for the new gameplay.

## 1.5 Overview

# Generally, this is a two-player strategy board game. The Tic-Tac-Toe game is based on having a game board (2D array) of size 3 x 3. The players alternate placing Xs and Os on the board until either one has placed three Xs or Os in a row horizontally, vertically, or diagonally; or all nine board squares are filled. The player wins if s/he draws three Xs or three Os in a row. Otherwise, the game is draw.

# 2. SCOPE

## 2.1 Scope of Project

Tic-Tac-Toe is a stress reliving game for any age-group. This game can be enjoyed by any one. There are no any sorts of restriction in the gameplay. The main purpose of this game is for entertainment purpose.

## 2.2 Aims

Here I have stated main aims this project:

1. The aim of this project is to develop a Tic-Tac-Toe game for mobile device.
2. Easy access for gameplay anywhere.
3. Help the children to grow thinking capacity.
4. Develop strategy at an early age.

## 2.3 Objectives

1. Helps to develop strategy at an early age
2. It can teach a child to want to win.
3. Easily accessible mobile game.
4. No need of internet access for gameplay.
5. Instant fun and laugh among people.
6. Quality time learning the moves for winning the game.
7. No restriction of age for playing the game.
8. Brain-storming for next moves in-order to win.
9. Multi-player feature lets two friends compete each-other.
10. Prepares them for more complex games because it helps you think of multiple things at once.

## 2.4 Limitations

1. This game requires multi-player in-order to play.
2. GUI is not so attractive.
3. Only mouse interface is implemented, keyboard is not activated in the game.

## 2.5 Overview

There are loads of stuff this project is offering to the players although it’s just a Tic-Tac-Toe game. Development of the strategy in the early stages are very crucial. This game helps the children develop their mindset of the Winner for sure.

# 3 DEVELOPMENT METHODOLOGY

For the development of this project, I am going to use Waterfall Model.

Waterfall model is a traditional way of describing the development of the any software, project. This methodology explains the progress of the system in a linear flow with a specified sequence for understanding that the further level is made progressive on completion of the previous one.

## 3.1 Description

Advantages of the waterfall methodology:

1. Very simple and easy to understand
2. Easily manageable due to rigidity of the model
3. Saves significant amount of time
4. This allows for easy testing and analysis

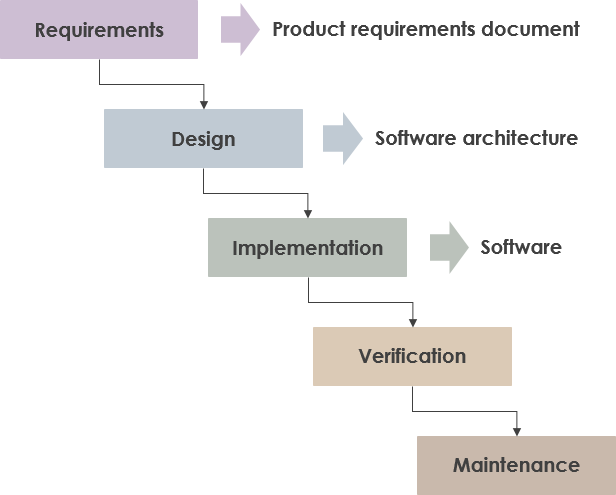


Fig 1: Waterfall Model

I have not chosen Waterfall methodology over other methodologies like Agile because:

1. Developers and customers agree on what will be delivered early in the development lifecycle. This makes planning and designing more straightforward while in Agile very high degree of customer involvement is required which May present problem for some customers. (**Our Project is quite simple with proper level of understanding on requirement of customer.**)
2. Progress is more easily measured, as the full scope of the work is known in advance in Waterfall while Agile focuses on time-boxed delivery and frequent reprioritization, it’s possible that some items set for delivery will not be completed within the allotted timeframe. (**this project must be completed within given timeframe which is aid by Waterfall method.**)

## 3.2 Design

For the development of the project I am using MVC (Model View Control)

“It is a software architectural pattern for implementing user interfaces on computers. It divides a given software application into three interconnected parts, so as to separate internal representations of information from the ways that information is presented to or accepted from the user.”

1. Model

A model is data used by a program. This May be a database, file, or a simple object.

2. View

A view is the means of displaying objects within an application.

3. Controller

A controller updates both models and views. It accepts input and performs the corresponding update.

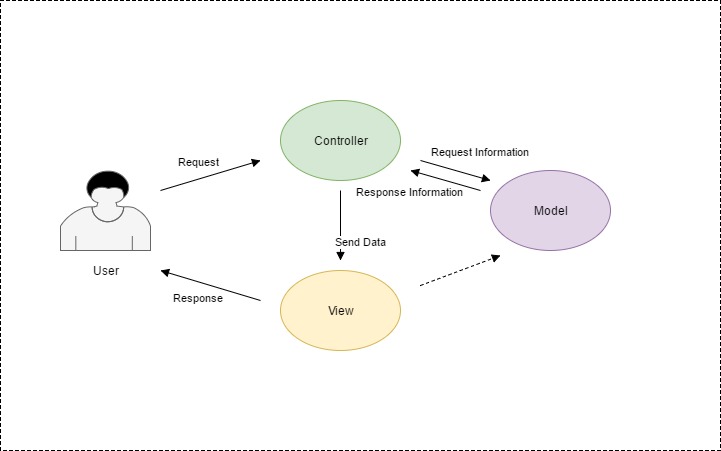


Figure 2: MVC Design Pattern

## Why??

I have used MVC design pattern than other design pattern since, the aim of the project is quite simple regarding Client-Server connection which is thoroughly maintained in MVC design pattern. Client-Server design pattern is the key for development of this project.

## 3.3 Architecture

For the development of this project I have chosen Three-tier method.

The 3-tier architecture is a type of software architecture which is composed of three “tiers” or “layers” of logical computing. 3-tier architectures provide many benefits for production and development environments by modularizing the user interface, business logic, and data storage layers.

3-tier means:

**A Presentation Layer** that sends content to browsers in the form of HTML/JS/CSS. This might leverage frameworks like React, Angular, Ember, Aurora, etc.

**An Application Layer** that uses an application server and processes the business logic for the application. This might be written in C#, Java, C++, Python, Ruby, etc.

**A Data Layer** which is a database management system that provides access to application data. This could be MSSQL, MySQL, Oracle, or PostgreSQL, Mongo, etc.

Advantages of using 3-tier architecture:

gives us the ability to update the technology stack of one tier, without impacting other areas of the application.

Allows for different development teams to each work on their own areas of expertise

able to scale the application up and out

adds reliability and more independence of the underlying servers or services.

Provides an ease of maintenance of the code base, managing presentation code and business logic separately.

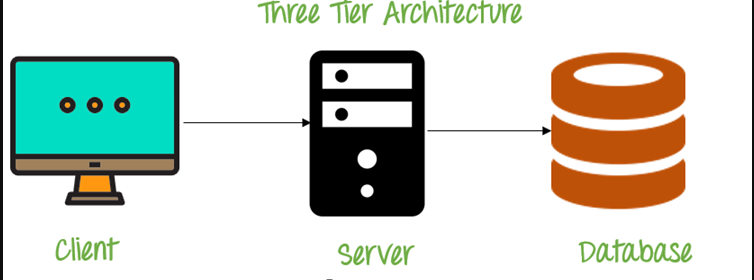


Figure 3: Three-tier Architecture

## Why???

Three-tier Architecture is perfect architecture required for the development of this project. I am developing this project Android Studio, Java and SQL for its database. Three-tier architecture holds the perfect balance among these than other architecture for this project.

# 4 PROJECT PLANNING

## 4.1 Work Breakdown Structure

A work breakdown structure (WBS) is a key project deliverable that organizes the team's work into manageable sections. The work breakdown structure visually defines the scope into manageable chunks that a project team can understand, as each level of the work breakdown structure provides further definition and detail.

Figure 4: WBS (Work Breakdown Structure)

## 4.2 MILESTONES

|  |  |  |  |
| --- | --- | --- | --- |
| Tic-Tac-Toe Game | | |  |
| S.N | Project Task | Number of Days | Date |
| 1. | Proposal | 16 Days | 25th March -9th April |
|  | * Planning * Brain Storming * Risk Analysis * Config Management | 4 days  4 days  4 days  4 days | 25th March-28th March  29th March-1st April  2nd April-5th April  6th April- 9th April |
| 2. | Analysis   * Requirement Analysis * Feasibility Study * Problem Study | 29 Days  10 days  9 days  10 days | 10th April- 8th May  10th April-19th April  20th April-28th April  29th April-8th May |
| 3. | Design   * Structural Model * Design Analysis * Design Feasibility | 26 Days  9 days  9 days  8 days | 9th May-3rd June  9th May-17th May  18th May-26th May  27th May-3rd June |
|  | Implementation   * System Implementation * Coding | 21 Days  7 days  14 days | 4th June-24th June  4th June-10th June  11th June-24th June |
| 5.  6. | Test   * System Testing * Debugging   Other Project Requirements | 7 Days  4 days  3 days  11 Days | 25th June -1st July  25th June -27th June  28th June-1st July  2nd July-12th July |
|  | TOTAL Days |  | 110 Days |

Figure 5: Milestone

As mention in above table I will be completing my respective task in the designated timeframe.

For the completion of Proposal, we are given 16 days, where I have allocated task like planning, brain-storming and risk analysis.

For the completion of Analysis, we are given 29 days, where I have allocated task like requirement analysis, Feasibility study and problem study.

For the completion of Design, we are given 26 days, where I have allocated task like Structural model, design analysis and design feasibility.

For the completion of Implementation, we are given 21 days, where I have allocated task like System Implementation (coding).

For the completion of Test, we are given 7 days, where I have allocated task like System testing.

For other needs of the project we are given 11 days of time period where miscellaneous task will be done.

## 4.3 Gantt Chart

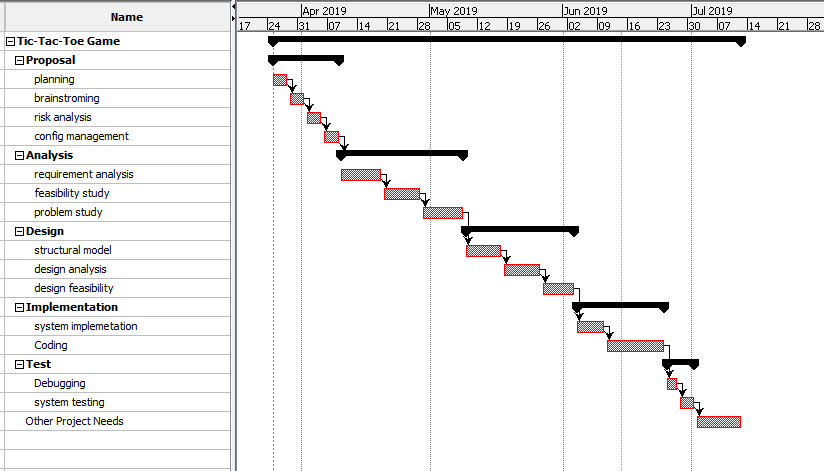


Figure 6: Gantt Chart-1

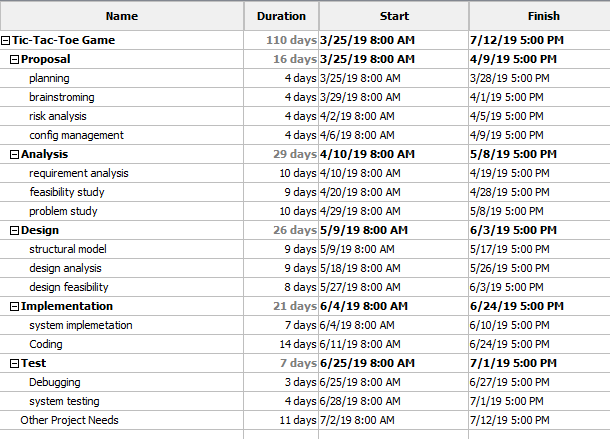


Figure 7: Gantt Chart-2

# 5 RISK MANAGEMENT

Risk management is defined as the process of identifying, monitoring and managing potential risks in order to minimize the negative impact they may have on an organization, project, on-going development task.

Life Cycle of Risk Management

The life of Risk Management has these which are actually a cycle:

1. Establish the Context

2. Identification

3. Assessment

4. Potential Risk Treatments

5. Create the Plan

6. Implementation

7. Review and Evaluation of the Plan

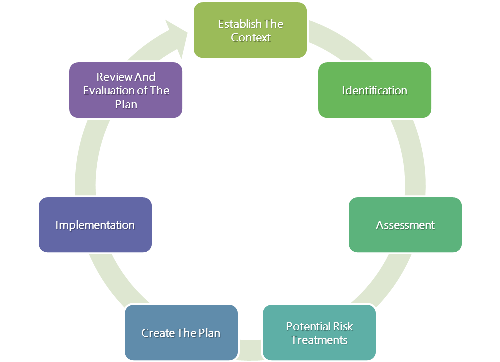


Figure 8: Risk Management Lifecycle

|  |  |
| --- | --- |
| Low | 1 |
| Medium | 2 |
| High | 3 |

Figure 9: Likelihood table

|  |  |
| --- | --- |
| Consequence | Value |
| Very low | 1 |
| Low | 2 |
| Medium | 3 |
| High | 4 |
| Very high | 5 |

Figure 10: Consequence table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S. N** | **Risk** | **Likelihood** | **Consequence** | **Impact** | **Action** | **Remark** |
| 1. | System crash | 1 | 2 | 2 | Proper system fucntions should be maintained. |  |
| 2. | Security Misconfiguration | 1 | 5 | 5 | Security must have a secure configuration defined and deployed for the application. |  |
| 3. | Insufficient Transport Layer Protection | 2 | 5 | 10 | Authenticate, encrypt, and protect the confidentiality and integrity of sensitive network traffic. |  |
| 4. | Malware | 1 | 3 | 3 | Anti-malware must be used in the system. |  |
| 5. | Session Hijacking | 1 | 4 | 4 | Proper training on the use of the web. |  |
| 6. | SQL Injection | 2 | 4 | 8 | Security scans should be performed from time to time. |  |

Figure 11: Risk Assesment

# 6 Configuration Management

Files and folders of the project are managed through Configuration Management. Each and every activity are allocated with respect to their directories and sub-directories. Here is the list of the directories and the sub-directories of the project.

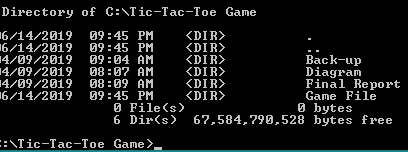


Figure 12: Configuration Management

Here is the link of my GitHub: <https://github.com/aashish003/cp-project>

Username: aashish003

# Conclusion

In this way I have complete this proposal where I have used Waterfall as its development methodology, MVC as its design pattern and Three-tier Architecture as its development architecture. Related files and folders are uploaded in GitHub.

# Reference and Bibliography

<https://acodez.in/12-best-software-development-methodologies-pros-cons/#Waterfall_Model>

<https://www.jinfonet.com/resources/bi-defined/3-tier-architecture-complete-overview/>

<https://www.izenda.com/5-benefits-3-tier-architecture/>

<https://www.workbreakdownstructure.com/>

<https://iedunote.com/risk-management>

<https://www.manageengine.com/products/asset-explorer/it-asset-life-cycle-management.html>

<https://www.interserver.net/tips/kb/mvc-advantages-disadvantages-mvc/>

<https://techterms.com/definition/mvc>

<https://www.seguetech.com/waterfall-vs-agile-methodology/>