Software Engineering and Project Management

Distanza

Description: horizontal line

# 

Submitted by- Gyanesh Samanta

Register Number- RA1911030010083

Section- O2 (TP-304)

Faculty In charge- CNS Vinoth Kumar

Team members- Ponnu Sharma

- Varsha Narra

-



**DEPT. Of Computer Science Engineering**

**SRM IST, Kattankulathur – 603 203**

**Course Code: 18CSC206J**

**Course Name: Software Engineering and Project Management**

|  |  |
| --- | --- |
| **Experiment No** | 1a |
| **Title of Experiment** | Distanza |
| **Name of the candidate** | Gyanesh Samanta |
| **Team Members** | Ponnu Sharma, Varsha Narra |
| **Register Number** | RA1911030010083 |
| **Date of Experiment** | 02/02/2021 |

**Mark Split Up**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Description** | **Maximum Mark** | **Mark Obtained** |
| 1 | Presentation | 5 |  |
| 2 | Project Description | 5 |  |
| **Total** | | **10** |  |

**Staff Signature with date**

**Aim**

To Frame a project team, analyze and identify a Software project

**Team Members:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl No** | **Register No** | **Name** | **Role** |
| **1** | **RA1911030010083** | **Gyanesh Samanta** | **Lead** |
| **2** | **RA1911030010078** | **Ponnu Sharma** | **Member** |
| **3** | **RA1911030010074** | **Narra Varsha** | **Member** |

**Project Title: Employee Management System**

**Project Description**

An arcade game is a coin operated game which was pretty famous and the talk of the town in the late 90s. A lot of our childhood involved going to the arcade, spending time with friends and even competing with them at times, also let’s not forget buying small confectionaries with the tickets we won. The main aim of our project is categorized around the theme as we like to call “A BLAST TO THE PAST”. With the highly talked off and emerging video games taking over the world we wanted to revive the nostalgic arcade games.

These arcade games can be played on PCs. Our main goal is to bring back the famous arcade games of the late 90s. Due to the pandemic, we have come up with a new game as well which we’d be developing from scratch. With the click of a button, you can relive your childhood nostalgia and play your favorite arcade games right on your PCs.

**Result:**

Thus, the project team formed and the project is described

**DEPT. Of Computer Science Engineering**

**SRM IST, Kattankulathur – 603 203**

**Course Code: 18CSC206J**

**Course Name: Software Engineering and Project Management**

|  |  |
| --- | --- |
| **Experiment No** | 1b |
| **Title of Experiment** | Create Business Case, Arrive at a Problem Statement |
| **Name of the candidate** |  |
| **Team Members** |  |
| **Register Number** |  |
| **Date of Experiment** |  |

**Mark Split Up**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Description** | **Maximum Mark** | **Mark Obtained** |
| 1 | Presentation | 5 |  |
| 2 | Business Case | 5 |  |
| **Total** | | **10** |  |

**Staff Signature with date**

Aim:

To create a business case and Arrive at a Problem Statement for the <title of the project>

Business Case

<Incorporate the Business Case template>

Result

Thus the business case was prepared and the problem statement was arrived

Lab Session #1

TEAM-5: DISTANZA

GYANESH SAMANTA (RA1911030010083)

PONNU SHARMA (RA1911030010078)

VARSHA NARRA (RA1911030010074)

Table of Contents

[Revision History 2](#_Toc61806107)

[1. Executive Summary 3](#_Toc61806108)

[2. Strategic Business Context 3](#_Toc61806109)

[2.1. Organization Overview 3](#_Toc61806110)

[2.2. Business Need 3](#_Toc61806111)

[2.3. Drivers for Change 3](#_Toc61806112)

[2.4. Business Outcomes 3](#_Toc61806113)

[3. Detailed Business Problem 3](#_Toc61806114)

[3.1. Problem/Opportunity Statement 3](#_Toc61806115)

[3.2. Scope 4](#_Toc61806116)

[3.3. High Level Requirements 4](#_Toc61806117)

[3.4. Assumptions 4](#_Toc61806118)

[3.5. Constraints 4](#_Toc61806119)

[3.6. Dependencies 4](#_Toc61806120)

[3.7. Stakeholder Analysis 5](#_Toc61806121)

[4. Analysis and Recommendation 5](#_Toc61806122)

[4.1. Evaluation Criteria 5](#_Toc61806123)

[4.2. List the Possible Options 5](#_Toc61806124)

[4.3. Cost of each Possible Options 5](#_Toc61806125)

[4.4. Risks 6](#_Toc61806126)

[4.5. Cost-Benefit Analysis 6](#_Toc61806127)

[4.6. Advantages and Disadvantages 6](#_Toc61806128)

[4.7. Recommended Option 7](#_Toc61806129)

[5. Implementation & Governance 7](#_Toc61806130)

[5.1. Required Skills 7](#_Toc61806131)

[5.2. Milestone 7](#_Toc61806132)

[5.3. Governance 8](#_Toc61806133)

[5.4. Change Management 8](#_Toc61806134)

[5.5. Performance Measurement 8](#_Toc61806135)

[6. Project Charter 8](#_Toc61806136)

[6.1. Simplified Project Charter 8](#_Toc61806137)

[6.2. Project Team Structure 9](#_Toc61806138)

[6.2.1. Roles & Responsibilities 9](#_Toc61806139)

[6.3. Approval 9](#_Toc61806140)

[Reference 10](#_Toc61806141)

# Revision History

| Version Number | Description | Date Modified | Author |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Executive Summary

Arcade games were a massive hit in the 90s. They were the talk of the town. Sadly, they have lost their course and have been replaced with up and popular video games.  Our main goal is to revive these arcade games and make them available to everyone. Furthermore, we are developing a new game focused on the younger generation that aims to keep people connected even though they might be miles away.

Our project is based on the theme "BLAST FROM THE PAST". We aim to relive the 90s nostalgia with these super fun arcade games. We have created most of our games using programming languages like Python, Godot and C++.

Keeping the pandemic in mind, we have also created a new game, "DISTANZA", from scratch. We have used Godot for this game. Though this fun initiative, we are 100% sure that all age groups would have the same fun as we had while doing this project.

# Strategic Business Context

The main objective of our project is to make these childhood favourite arcade games available for everyone. As a team, we have designed these arcade games in a way where everyone can play them on their laptop/pcs. "Arcade games right near you in your laptop".

We have strategised our project, keeping in mind the booming video games business. Video games have become a massive part of our life. Irrespective of the age, everyone has been spending their pass time playing video games. Our vision is to develop these arcade games using simple programming languages and make them available to everyone.

Vince Lombardi once said – "Individual commitment to a group effort--that is what makes a **team work**, a company work, a society work, a civilisation work". As a team of 3, we have taken up the joint responsibility of developing these arcade games using programming languages.

## Organisation Overview

Game studios are actively looking for ideas for games and investment opportunities. With our flagship game "Distanza" we aim to attract game studios to invest in the application for its future development and mass availability.

## Business Need

Looking at the current scenario, video games have pretty much become a considerable part of our lives. Most of our pass time would include playing video games. Let us not forget a lot of us had included "playing video games" as our hobbies when we were young. Keeping in mind the ever-growing video game business, we wanted to bring a touch of retro to this business with arcade games on your laptops/pcs. Games never go out of trend, adding a nostalgic element to it with these arcade games would significantly push our business.

## Drivers for Change

Video games have become more about the future nowadays. Everyone enjoys playing these games yet deep down we do miss the regular arcade trips we used to take us kids. That is what our team has tried to change- childhood favourite games right on your laptop. Our team has been striving to create something personal close to our heart, and that is when we landed on "arcade games". Retro is never out of trend, and that is one style we wanted to add to video games.

## Business Outcomes

Our entire business idea is based on the booming video games business. Video games are always trending. Livestream of these games are the most-watched videos on streaming platforms. With video games being of such high demand, our business with a retro aspect would stand out. Arcade games are always seen as the ones on giant machines or the old cell phones or controllers, but our team has strived continuously to break the stereotype. We want to make these games available to everyone compatibly on their laptops/pcs.

We have created a game "DISTANZA" from scratch with the high demand of innovative video games. Considering the pandemic we currently live in, creative video games are taking over the market. With the increased demand for innovative video games, DISTANZA would stand out as one of a kind.

Looking at the high demand for video games, we are sure of realising our base investment within a year of supply.

# Detailed Business Problem

## Problem/Opportunity Statement

Arcade games were a massive part of our childhood, reviving them has been our topmost priority. Breaking the cliché thought of arcade games of them being on giant machines or old cell phones, we have developed these video games using simple programming languages available on your laptops/pcs.

To keep up with the innovative and booming video games, we are jointly working on "DISTANZA" from scratch. DISTANZA is a one-of-a-kind, creative, and interactive video game suitable for all age groups. Innovation is currently the talk of the town, and DISTANZA is a standout video game.

## Scope

Everyone around the world plays video games. They are one of the most successful tech moguls around the world. We as a team are adding to this tech mogul a touch of retro. Arcade games are everyone's favourite. They were a massive part of our childhood. We are developing these arcade games to be made available on everyone's laptops/pcs. All age groups enjoy video games, and adding retro aspect would increase the number of users. With these games being simple and user friendly, it is suitable for all age groups.

With the booming innovative video games, DISTANZA would hit the markets and attract the crowd. With the growing surplus of innovative video games, DISTANZA would interest all age groups for it is highly interactive.

## High-Level Requirements

The video games these days are majorly focused on teenagers. As a team, we aim to diversify our reach by making these arcade games simple, user- friendly and available to all. Arcade games have been a part of everyone's childhood. Developing these games to be played on laptops/pcs would widen the reach.

## Assumptions

|  |  |
| --- | --- |
| **S.No** | **Assumptions** |
|  | One of the significant assumptions concerning these games would be the lack of interest within the younger generations. Younger generations may not be familiar with these arcade games and widen the radius to include them and result in a lot of PR planning. |
|  | DISTANZA being a new game would require a lot of PR investment. |

## Constraints

|  |  |  |
| --- | --- | --- |
| **S.No** | **Category** | **Constraints** |
|  | Technical | Looking at the technical aspects, the time required for the development, testing and publicity of our project would be a lot |
|  | Sales | Considering the sales aspects, the first few months after the initial launch of the product would be bumpy. The product has to publicise itself and stand out among all other games for it to cover the scope of our business |

## Dependencies

* The video game industry is booming, and online multiplayer is booming, even more, we aim to tap into this audience.

|  |  |  |
| --- | --- | --- |
| **Dependency Description** | **Critical Date** | **Contact** |
| Nil (as of now) |  |  |

## Stakeholder Analysis

We, the three developers, are the major stakeholders of the project as of today.

|  |  |  |
| --- | --- | --- |
| **Name** | **Designation** | **Role in Project** |
|  | Corporate Head for Sales & Marketing | Executive Sponsor |
|  | Chief Information Officer Or Regional Head of Sales & Marketing | Project Sponsor |
|  | Finance Head | Cost Approver |
|  | Department Head(s) | Scope / Requirement Approver |
|  | Business User(s) | Validate the functionalities |

# Analysis and Recommendation

## Evaluation Criteria

Games evaluation for "Distanza" would be done based on the number of active users who will use the game at once with a smooth experience.   
Furthermore, the compatibility of our game on older versions of windows is a significant evaluation criterion.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Evaluation Criteria** | **Deal Breakers (5)** | **Minimum Requirement (3)** | **Non-essential (1)** | **Score** |
| Personal Identifiable Information (PII): Data must be encrypted 'At Rest and 'On Transit' | Y | N | N | 5 |
| Project validity | Y | N | N | 5 |
| Project design | N | Y | N | 3 |
| Simple and Compatible Design | Y | N | N | 5 |
| Meeting the requirements of the initial project design | Y | N | N | 5 |
| The types of arcade games chosen | N | Y | N | 3 |
| Design of DISTANZA | Y | N | N | 5 |
| Sustenance of DISTANZA | N | Y | N | 3 |

## List the Possible Options

* The most feasible option for our project would include directly releasing the games on online platforms. Instead of downloading the game and downloading the other software to run the programs, removing these games online would be the most feasible option. These games are developed using considerably primary programming languages and rectifying their issues would not affect anyone.
* The other possible option would be selling the said games to a gaming company and releasing it on the front end. This option would be cost-efficient. The publicity and marketing cost of the product would be taken care of by the company. We would act as the backend developers of the games.

## Cost of each Possible Options

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Options (#)** | **One Time [CapEx]** | | **Operational [OpEx]** | | | **Total Cost in INR** |
| **Effort (Cost)** | **Infrastructure Cost** | **License Cost** | **Maintenance Cost** | **Infrastructure Increment** |
| TBA | TBA | TBA | TBA | TBA | TBA | TBA |
| TBA | TBA | TBA | TBA | TBA | TBA | TBA |

|  |  |
| --- | --- |
| **Category** | **Cost in INR** |
| One Time (CapEx) | TBA |
| Operational (OpEx) | TBA |

## Risks

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk Matrix – Probability and Impact Scoring** | | | |
| Probability | Impact | | |
| Low (1) | Medium (3) | High (5) |
| Almost Certain (5) | 5 | 15 | 25 |
| Likely (3) | 3 | 9 | 15 |
| Unlikely (1) | 1 | 3 | 5 |

|  |  |
| --- | --- |
| **Criteria** | **Risk Category** |
| 0 to <= 9 | Low |
| >= 9 to <15 | Medium |
| >=15 to <25 | High |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Risk ID (#)** | **Risk Description** | **Impact Score** | **Probability Score** | **Probability X Impact Score** | **Risk Category [Low/Medium/High]** | **Risk Appetite [ Accept/ Mitigate/ Transfer/Transfer]** |
| R01 | Technical risks- failure of code developed | 15 | 5 | 20 | High | TBA |
| R02 | Failure of the project once launched | 9 | 3 | 12 | Medium | TBA |
| R03 | Not realising the initial investment | 9 | 3 | 12 | Medium | TBA |

## Cost-Benefit Analysis

<Based on the costs established for each option, describe how those costs are weighed against the benefits. Conduct the cost-benefit analysis for each prospect, taking into account costs, services, and risks>

## Advantages and Disadvantages

|  |  |  |
| --- | --- | --- |
| **Solution Options (#)** | **Advantages** | **Disadvantages** |
| TBA | TBA | TBA |
| TBA | TBA | TBA |

## Recommended Option

The recommended option would be selling the developed game to an interested party. The organisation will realise its investment immediately after the project is developed and sold. The said interested party will buy out the project at the valuated price of the project. This inherently would result in profits considering the project checks out all the milestones.

# Implementation & Governance

## Required Skills

|  |  |
| --- | --- |
| Skills | More Info |
| UX Designer | Designing experience of the user |
| Frontend Development | Design and Develop UI and frontend layer |
| Backend Development | Design Database and Develop Service / API |
| Testing | Develop Test Cases |
| Project Management | Project Planning, Scheduling, Executing, Monitoring and Controlling |

## Milestone

Identify the significant points or events in the project. This table can also represent a high-level project schedule.

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Project Milestone** | **Description** | **Expected Date** |
|  | Developing the first- level of arcade games | The first level of arcade games would include simple and basic arcade games developed using primary programming languages | 05/03/2021 |
|  | Developing the second-level of arcade games | The second-level of arcade games would include complex arcade games | 05/04/2021 |
|  | The initial phase of DISTANZA- design and analysis of the requirements | In this phase of the project, we will be developing the front end design of DISTANZA | 15/04/2021 |
|  | Backend development of DISTANZA | This stage would involve the complete back end development of DISTANZA, i.e., the coding and connecting it to the front end | 05/05/2021 |
|  | Test run-1 of DISTANZA | The first test run of DISTANZA | 10/05/2021 |
|  | Test run-2 of DISTANZA | The second test run of DISTANZA | 15/05/2021 |
|  | The release of our final product into the market | The last and final stage of our project releasing the fully functional product into the market | 25/05/2021 |

## Governance

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Meeting** | **Frequency** | **Participants** | **Host** | **Chair** |
| Steering Committee | Monthly | Project Sponsor, Department Heads, Project Manager | Project Manager | Project Sponsor |
| Department Head | Fortnightly | Department Heads, Project Manager, Technical Lead | Project Manager | Department Head(s) |
| Project Team Review | Weekly | Project Manager, Technical Lead, Project Team | Technical Lead | Project Manager |

## Change Management

Our team is well trained and is ready to face any oncoming challenges together. During the test phases of the project, our team will be prepared with a backup plan to replace any failure in the initial plan. With is running two test runs, all issues shall be rectified.

## Performance Measurement

<Return on Investment to presented and agreed with sponsor >

|  |  |  |  |
| --- | --- | --- | --- |
| **Return in timeline** | **Return in INR** | **Investment (INR)** | **Remaining Investment** |
| Return on 1st year | 50,000 | 5,00,000 | 4,50,000 |
| Return on 2nd year | 50,000 | 5,00,000 | 4,50,000 |
| Return on 3rd year | 50,000 | 5,00,000 | 4,50,000 |
| Return on 4th year | 50,000 | 5,00,000 | 4,50,000 |
| Return on 5th year | 50,000 | 5,00,000 | 4,50,000 |

# Project Charter

## Simplified Project Charter

|  |  |
| --- | --- |
| **Section** | **Details** |
| Project Scope | Bringing back the childhood favourite arcade games near you on your laptops. We have also created a new game from scratch called "DISTANZA" that is one of a kind and our standout highlight |
| Project Schedule | This project shall be delivered the earliest by 25/05/2021 |
| Project Cost | 5,00,000 |
| Constraints | The significant constraints for a technical project would include the slight glitches in the code. We are ready with a backup plan and 2 test runs to eradicate glitches if any |
| ROI | The risk in this project is of medium to high range |
| Intangible Benefit | Improved safety of privacy and over the top graphics would increase the brand value |

## Project Team Structure

Use an organisational chart to show the structure of the project team as well as the relationships between team members

## Roles & Responsibilities

|  |  |  |
| --- | --- | --- |
| **Project Role** | **Responsibilities** | **Assigned To** |
| Project Steering Committee (Executive Sponsor) | Reviewing the progress and performance of the project upon completion of each milestone | Gyanesh, Ponnu, Varsha |
| Project Sponsor | Reviewing the financial situation of the project upon completion of each milestone | Gyanesh, Ponnu, Varsha |
| Project Manager | Reviewing the performance of the team at all levels | Gyanesh, Ponnu, Varsha |
| Technical Lead | Reviewing and working on the technical aspects of the project making it user friendly | Gyanesh, Ponnu, Varsha |
| Business Analyst | Analysing the necessary marketing strategies for the project | Gyanesh, Ponnu, Varsha |
| Developer | Developing the entirety of the project | Gyanesh, Ponnu, Varsha |
| Tester | Performing the 2 test runs | Gyanesh, Ponnu, Varsha |

## Approval

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Designation** | **Role in Project** | **Signature** |
| Gyanesh, Ponnu, Varsha | Corporate Head of Sales & Marketing | Executive Sponsor |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Designation** | **Role in Project** | **Signature** |
| Gyanesh, Ponnu, Varsha | Regional Head of Sales & Marketing | Project Sponsor |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Designation** | **Role in Project** | **Signature** |
| Gyanesh, Ponnu, Varsha | Project Manager | Project Manager |  |

# Reference

1. <https://www.pmi.org/>
2. <https://www.projectmanagement.com/>

*Project Methodology and Stakeholder Identification/Analysis*

##### Lab Session #2

Table of Contents

1. [Executive Summary 2](#_bookmark0)
2. [Selection of Methodology 2](#_bookmark1)
   1. [Roles and Methods 2](#_bookmark2)
   2. [Agile Development and Testing(optional) 2](#_bookmark3)
3. [Stakeholder Management 2](#_bookmark4)
   1. [Identification of Stakeholders 2](#_bookmark5)
   2. [Interest and Influence matrix 2](#_bookmark6)
   3. [Communication Plan for Stakeholders 4](#_bookmark7)

[Reference 4](#_bookmark8)

# Executive Summary

The majority of games projects have external partners involved in their development. This could be a publisher, a VC, an IP holder or a project financer. Often there are multiple partners involved. However, for our project we chose to be the stake holders ourselves as great projects tend to be ones where communication and understanding is well-established from the outset and being a team very much fulfils this requirement.

# Selection of Methodology

We used ‘AGILE’ methodology for our project as it would provide us with easy iterations to our project and being a small team project, it was an easier to manage approach with testing and the final product.

* Gather and document requirements.
* Design.
* Code and unit test.
* Perform system testing.
* Perform user acceptance testing (UAT)
* Fix any issues.
* Deliver the finished product.

## Roles and Methods

We have grouped designers, developers, and marketers into one category to represent the group of people that creates these games and advertises them to the market. Give that we are a small team of programming background, we have decided to divide ate the parts equally among ourselves for a better approach.

# Stakeholder Management

## Identification of Stakeholders

When there are multiple stakeholders, communication is often a time-consuming task. Some of the stakeholders may not have a strong grasp of the specific nuances of game development, or a project may face complications, and in situations like this good communication can be challenging yet crucial, hence we identified ourselves as stake holders.

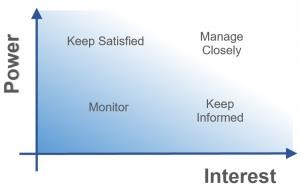
## Interest and Influence matrix

Power-Interest Matrix

In a [stakeholder analysis,](https://www.projectengineer.net/how-to-perform-a-stakeholder-analysis/) there are two variables which affect the project more than any other: Power and Interest.

* **Power** is the ability of the stakeholder to stop or change the project. For example, a government regulatory approval authority has a very high level of power.
* **Interest** is the size of the overlap between the stakeholder’s and the project’s goals. For example, a landowner whose house has to be removed to make way for a project has a very high interest.

These two central variables are plotted on a chart with the x-axis being interest and the y-axis being power. The resulting chart might feel very simple but it is a solid analysis of the stakeholder’s interaction with the project.



* Stakeholders with ***high power*** and ***high interest*** are major stakeholders that are heavily invested in the project. They must be actively managed.
* Stakeholders with ***high power*** but ***low interest*** must be kept satisfied. They can derail the project over seemingly minor issues.
* Stakeholders with ***low power*** but ***high interest*** must be kept informed. They can create high influence (i.e. raise a stink) if they don’t get what they want.
* Stakeholders with ***low power*** and ***low interest*** must be monitored, in case they become more powerful and affect the project in the future.

|  |  |
| --- | --- |
| **Interest** | **Influence** |
| High | High |
| Low | Low |
| Low | High |
| High | Low |

|  |  |
| --- | --- |
| **Low Interest, High Influence**  Keep them satisfied as they can be  ‘defenders’  Help them engage more | **High Interest, High Influence** Engage them closely as they are key ‘drivers’ |
| **Low Interest, Low Influence**  Low Priority as they are ‘spectators’ | **High Interest, Low Influence**  Keep them informed as they can be  ‘blockers’ |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stakeholder Name** | **Activity / Area / Phase** | **Interest** | **Influence** | **Priority (High / Medium/Low)** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Regional Head of Sales & Marketing | Subscription using mobile App | High | High | 1 |

## Communication Plan for Stakeholders

3

Low

High

Multiple Currency Payment

Finance Account Receivable consultant

* **Milestone sign off process**
* **Consistency and clarity**
* **Transparency**
* **Accountability**
* **Efficiency**
* **Security**

Are the basis of our communication plan for stakeholders.

# Reference

###### 1. <https://www.pmi.org/learning/library/stakeholder-analysis-pivotal-practice-projects-8905>

Management

Lab Session #3

Submitted by: Team Distanza  
  
Gyanesh Samanta (RA1911030010083)

Ponnu Sharma (RA1911030010078)

Varsha Narra (RA1911030010074)

Table of Contents

[1. Executive Summary 2](#_Toc63943401)

[2. Project Scope 2](#_Toc63943402)

[2.1. In Scope 2](#_Toc63943403)

[2.2. Out of Scope 2](#_Toc63943404)

[3. Epics [Major Functions] 2](#_Toc63943405)

[4. Requirements 2](#_Toc63943406)

[4.1. Functional Requirements 2](#_Toc63943407)

[4.2. Non-Functional Requirements 3](#_Toc63943408)

[4.3. Infrastructure Requirements 3](#_Toc63943409)

[4.4. Requirement definition in Agile [Optional … Use according to methodology chosen by student] 4](#_Toc63943410)

[Reference 4](#_Toc63943411)

# Executive Summary

Arcade games were a huge hit in the 90s. They were the talk of the town. Sadly, they have lost their course and have been replaced with the up and popular video games. Our main goal is to revive these arcade games and make them available to everyone with the click of a button.

Our project is based on the theme “BLAST TO THE PAST”. We aim to relive the 90s nostalgia with these super fun arcade games created using simple programs. We have created most of our games using simple programming languages like C, C++ and python.

Keeping the pandemic in mind, we have also created a new game, DISTANZA, from scratch. We have used godot for this game. Through this fun initiative we are 100% sure that all age groups would have the same fun as we had while making this project.

# Project Scope

Expand on the scope definition and outline the major activities required to successfully complete the project (for example, develop module ABC, develop requirements document, etc.). Out of scope activities are identified to reduce ambiguity.

|  |  |  |
| --- | --- | --- |
| **S.No** | **Activities In Scope** | **Activities Out of Scope** |
|  | Designing Project Layout | The platform issues created where DISTANZA and the other arcade games are played |
|  | Developing the arcade games using basic programming languages | The Hardware issues faced by the user where the particular software shall run |
|  | Developing and testing the modules | Selling the product to business |
|  | Testing and developing various alternatives | Resolving hardware issues,if any |
|  | Solving software issues, if any | PR of the product |

## In Scope

1. Designing Project Layout- Deciding the project outcome and mapping all the project functionalities and features.
2. Developing the arcade games using basic programming languages- Our main aim of the project is to develop the 90s arcade games using basic programming languages and running them on PCs to make them available to everyone. We have allocated specific tasks to each of our team members and have curated specific milestones and plans to eradicate any errors.
3. Developing and testing the modules- Since our project is majorly a software product, we have used quite basic programming languages to develop the games and also to avoid any conflictions in the games as such. DISTANZA is a full length new game that we are trying to introduce into the market and we as a team are striving towards not complicating it.
4. Testing and developing various alternatives- Though our project is mainly focussed on trying to develop these games using simple programming languages, we are well prepared to avoid any security and network issues, if any. Using simple programming languages, we are trying to reduce complications in coding. We have set up various milestones with enough deliverable time that will help us work better on our projects. We also have set up 2 test runs for Distanza to avoid any error in the game we have developed from scratch.
5. Solving software issues, if any- We, as a team are trying to develop a software product and we have the sole responsibility to solve software issues, if any.

## Out of Scope

1. The platform issues- The platform where DISTANZA and the other games will be played/run is majorly PCs. The hardware/software issues related to the PC will not be accounted for by our team.
2. Hardware issues- The issues regarding the PC will not be accounted for by our team
3. Selling the product­- Our team’s main responsibility is to just develop the software product. The sales and marketing of the product will be taken care of by the company.
4. PR of the product- With just a team of 3, the customer communication is a difficult aspect to cover due to low manpower. We are planning to sell our product to a gaming company and be the shareholders of the product.

# Epics [Major Functions]

<List down the modules of the product / application >

|  |  |
| --- | --- |
| **Epic (#)** | **Epic Description** |
| E1 | Arcade Games+DISTANZA on PCS |
| E2 | User Login/Registration |
| E3 | Fast Loading speed and wider reach |
| E4 | Applicable to all ages |
| E5 | Simple yet everyone’s favourite arcade games |
| E6 | DISTANZA- a new age game with wider reach |

# Requirements

## Functional Requirements

Functional Requirements can also be expressed in the form of “user story” which is the smallest unit of work in an agile framework. It’s an end goal, not a feature, expressed from the software user’s perspective.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement (#)** | **Requirement Specification** | **Department** | **Name of Business User** | **Status** |
| E1FR1 | The game must be interactive and should be reachable to all age groups | Department of Development |  | To have wider reach for the game |
| E1FR2 | The games must not have any software issues that might affect the platform it is being run on | Department of Testing |  | To eradicate any problems |
| E1FR3 | The games must be developed using simple programming languages so as to reduce complications and make it easier to solve issues | Department of Developing |  | To avoid errors in the code |
| E1FR4 | DISTANZA must be a unique game and must not raise any copyright issues | Department of sales and marketing |  | To avoid any legal issues |

## Non-Functional Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requirement (#)** | **Category of NFR** | **Requirement Specification** | **Department** | **Name of Business User** | **Status** |
| E1NFR1 | Performance | All pages should load within 3 seconds | Department of Developing | - | To resolve any page lagging issues |
| E1NFR2 | Usability | This product shall be used by users having a compatible PC to run the programs | Department of Developing |  | To make the games have a wider reach |
| E1NF3 | Reliability | Should ensure that no systems will crash/blue screen | Department of testing |  | To ensure a safe environment so as to avoid any difficulties |
| E1NF4 | Sales pitch | To ensure a good sales pitch so as to have better profits in return | Department of sales and marketing |  | To develop a good sales pitch to sell the product |
| E1NF5 | Legal issues | To avoid legal issues if any | Department of sales and marketing |  | To avoid copyright issues |

## Infrastructure Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement (#)** | **Requirement Specification** | **Department** | **Name of Business User / Project Team Member** | **Status** |
| IR1 | Development Machine with 12 GB Ram and 8 Cores | Technical Team | All Members | Team Members and Leads |
| IR2 | Code Repository and IDE | Technical Team | All Members | Team Members |
| IR3 | AWS S3 Bucket | Technical Team | All Members |  |
| IR4 | IDE – Eclipse | Technical Team | All Members | Team Members |

## Requirement definition in Agile [Optional … Use according to methodology chosen by student]

User story is the smallest unit of work in an agile framework. It’s an end goal, not a feature, expressed from the software user’s perspective.

How to write a user story…

* Who are we building it for, who the user is? — As a <type of user>
* What are we building, what is the intention? — I want <some goal or objective >
* Why are we building it, what value it brings for the user? — So that <benefit, value>

|  |  |  |
| --- | --- | --- |
| **User Story** | **Acceptance Criteria** | **Size of User Story** |
| As customer, I can view all available cars for delivery within 15 days. So, I can place my order for preferred car. | View all cars within 15 days delivery  Allow to choose my preferred car (Model)  Allow to place on order | Small / Medium / Big |

# Reference

1. <https://www.pmi.org/>
2. <https://www.atlassian.com/agile/project-management/user-stories>

Submitted by-

***Project Management Plan, Effort and Cost Estimation and Team Formation***

***Lab Session #4***

1. Gyanesh Samanta (RA1911030010083)
2. Varsha Narra (RA1911030010074)
3. Ponnu Sharma (RA1911030010078)

Date of experiment: 01/03/2021

**Table of Contents**

[***1.***](#_gjdgxs) ***Executive Summary 2***

[***2.***](#_30j0zll) ***Project Management Plan 2***

[***3.***](#_1fob9te) ***Estimation 3***

[***3.1.***](#_3znysh7) ***Effort and Cost Estimation 3***

[***3.2.***](#_2et92p0) ***Infrastructure/Resource Cost [CapEx] 3***

[***4.***](#_tyjcwt) ***Maintenance and Support Cost [OpEx] 4***

[***5.***](#_3dy6vkm) ***Project Team Formation 4***

[***5.1.***](#_1t3h5sf) ***Identification Team members 4***

[***5.2.***](#_4d34og8) ***Responsibility Assignment Matrix 4***

[***Reference 5***](#_2s8eyo1)

# Executive Summary

The project will start by taking inputs from clients about their requirements and expectations. They are expected to provide specific details on the control system of preference (keyboard+mouse/ controller/joystick) to operate our application.

After receiving the inputs, teams are to be formed based on skill sets, experience in various fields and interests(as far as possible). Roles and responsibilities will be assigned to teams, and hierarchy will be laid for proper communication. Afterwards, units will be provided with the necessary resources, and deadlines will be set to avoid the delivery date's postponement. The teams can start working as per their patterns, including frequent meetings and client interaction. After completing each prototype, the project will be tested under various circumstances, flaws/issues encountered will be identified by the quality assurance team and fixed by the respective department.

After all the checks of the final prototype, the project will be delivered and deployed.

Skills Required:

Various skills are required for the project. The most important ones are the development and coding skills. In addition to it, a good set of managerial skills are needed to manage the number of people working on the project. Soft skills as communication, teamwork analytical skills also play a crucial role.

# Project Management Plan

|  |  |
| --- | --- |
| **Focus Area** | **Details** |
| Integration Management | Team Lead (1) Managers (2)  Developers and Software experts (3)  Analysts (4)  1- Controls activities and makes sure standards are met  2- Supervises and manages the team and deadlines  3- creates and fixes bugs using technical skills  4- analyze the project for flaws and report |
| Scope Management | The project aims to provide a smooth and cross-platform gaming experience to all age groups. It aims to build children's analytical and logical skills whilst providing teenagers and adults with leisure activity. |
| Schedule Management | Milestone1: The prototype with three retro games; Pacman, chess would be rolled out and tested for bugs  Milestone2: The second batch of retro games would be rolled out, and the development of DISTANZA begins.  Milestone3: The prototype of DISTANZA is rolled out for the beta phase |
| Cost Management | Budget- 2000$ (rough estimate)  All technical work on free platforms and resources. The workforce is minimal, and the LAN network reduces the cost of software and resources. No expensive gizmos.  Effort: 20 hours+/week by every member associated.  Teams will be formed based on skillset and interest. |
| Quality Management | QA: Extensive plans are mapped out, and resources will be provided beforehand to reduce time delays. Analysts will help the developers improve the product so that there are minimum flaws and problems.  QC: User feedback will be focused on, and reports of the games' performance analysis will explain how the games perform under different specifications of the hardware. |
| Resource Management | People: Project team covering all the skills from management to technical to analytical to marketing skills  Finance: Monetary support from investors and an initial investment of 2000$  Physical: Computer systems with the necessary system requirements and a database to store user data. Communication channels for clear communication. |
| Stakeholder | The developers associated, GODOT game engine, python, pycharm, anyone interested in buying video games.  By making the games cross-platform and easy to use would expose them to the maximum number of people |
| Communication Management | Weekly scheduled meetings between team leads and managers and monthly meetings with team leads and clients. The team needs to be updated with how the project will compare to how it should be, ideally on a timetable basis. Meetings between groups can be in person or video calls. Communication should be open and full-duplex. |
| Risk Management | User data privacy is given the highest importance and is stored locally on the device instead of the cloud.  Inability to complete the prototypes in time and thus focus is on being on schedule.  Incorrect analysis of user data results in false information for future investment. |
| Procurement Management | Most resources are online as the product is heavily technical.  A team is a pool of people we already know and are qualified.  Most members have sound computing systems, but a few can be bought online with the required system specifications. |

# Estimation

# Effort and Cost Estimation

Required resources will be estimated based on the project plan and the number of people working under the project with the error range of 2-5%—resources including physical infrastructure, financial resources and human resources. The cost will be estimated based on the requirements and scale of the project.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **WBS** | **Activity** | **Activity Description** | **Sub-Task** | **Sub-Task Description** | **Effort (in hours)** | **Cost in INR** |
| E1FR1 | E1R1A1 | Design the user screen | E1R1A1T1 | Confirm the user requirements (acceptance criteria) | 3 |  |
| E1R1A1T2 |  |  |  |
| E1R1A1T3 |  |  |  |
|  |  | Identify Data Source for displaying units of Energy Consumption |  | Go through Interface contract (Application Data Exchange) documents | 5 |  |
|  | Document |  |  |

|  |  |
| --- | --- |
| **Effort (hr)** | **Cost (INR)** |
| 1 | 500 |

# Infrastructure/Resource Cost

|  |  |  |  |
| --- | --- | --- | --- |
| **Infrastructure Requirement** | **Qty** | **Cost per qty** | **Cost per item** |
| Domain | 1 | 10$ | 10$ |
| Server | 1 | 15$ | 15$ |

# Maintenance and Support Cost [OpEx]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Category** | **Details** | **Qty** | **Cost per qty per annum** | **Cost per item** |
| People | Network, System, Middleware and DB admin  Developer , Support Consultant | 3 | 2,000,000 | 6,000,000 |
| License | Operating System  Database  Middleware  IDE | 10 | 10000 | 100,000 |
| Infrastructures | Server, Storage and Network | 20 | 20000 | 400,000 |

# Project Team Formation

# Identification Team members

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Responsibilities** |
| Gyanesh Samanta | Backend Developer for Distanza and retro games | Design, Develop and Unit Test Services/API/DB, team lead |
| Ponnu Sharma | The retro game developer using python | Manage the project, develop the retro games |
| Varsha Narra | The retro game developer using python | Manage the project, analytics, development of retro games |
| X | Technical Lead | Design the end-to-end architecture |
| U | UX Designer | Design the user experience |
| I | Frontend Developer | Develop user interface |
| Y | Backend Developer |  |
| O | Cloud Architect | Design the cost-effective, highly available and scalable architecture |
| P | Cloud Operations | Provision required Services |
| Q | Tester | Define Test Cases and Perform Testing |

# Responsibility Assignment Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RACI Matrix** | **Team Members** | | | |
| **Activity** | **Name (BA)** | **Name (Developer)** | **Name (Project Manager)** | **Key Business User** |
| User Requirement Documentation | A | C/I | I | R |
| Website design |  | A | R | C |
| Database Management |  | A | R | C |
| Security management |  | A | R | C |
| Marketing and social media | A |  | R | I |
| Quality Control | I | A | A | C |
| Requirements and wants |  | A | R | A |

|  |  |
| --- | --- |
| A | Accountable |
| R | Responsible |
| C | Consult |
| I | Inform |

# Reference

1. <https://www.pmi.org/>
2. <https://www.projectmanagement.com/>

1. <https://www.tpsgc-pwgsc.gc.ca/biens-property/sngp-npms/ti-it/ervcpgpm-dsfvpmpt-eng.html>

# Result:

The project plan was prepared based on scope, job roles and responsibilities. Project effort calculated based on resources.

*WBS and Risk Management Plan*

##### Lab Session #5

**Table of Contents**

1. [Executive Summary 2](#_TOC_250005)
2. [WBS With Project Schedule 2](#_TOC_250004)
3. [Risk Identification 2](#_TOC_250003)
   1. [List (Describe) Register 2](#_TOC_250002)
   2. [Managing Risk 2](#_TOC_250001)

[Reference 3](#_TOC_250000)

# Executive Summary

***Milestone Summary and Detailed schedule:***

* + In order to successfully deliver the product, we have laid down the following major milestones which are to be completed in a given time duration.
* Developing the first- level of arcade games
* The first level of arcade games would include simple and basic arcade games developed using basic programming languages
* **05/03/2021**
* Developing the second-level of arcade games
* The second- level of arcade games would include complex arcade games
* **05/04/2021**
* Initial phase of DISTANZA- design and analysis of the requirements
* In this phase of the project we will be developing the front end design of DISTANZA
* **15/04/2021**
* Backend development of DISTANZA
* This stage would involve the complete back end development of DISTANZA i.e., the coding and connecting it to the front end
* **05/05/2021**
* Test run-1 of DISTANZA
* The first test run of DISTANZA
* **10/05/2021**
* Test run-2 of DISTANZA
* The second test run of DISTANZA
* **15/05/2021**
* The release of our final product into the market
* The last and final stage of our project releasing the fully functional product into the market
* **25/05/2021**

### ***Risk management:***

**Project risks**: various risks such as insufficient funds for marketing purposes, lagging behind the schedule, unavailability of a certain team member.

**Technical risks**: implementation, testing, and maintenance issues. Technical risks conjointly embody ambiguous specifications, incomplete specification, dynamic specification, technical uncertainty, and technical degeneration.

**Business risks**: These risks are the worst of the risks a team can face. If the finance web application made by the team is outdated or is not required by anyone then it will not fetch any profits for the stakeholders which will lead to loss of resources, monetary as well as human.

**MANAGEMENT:**

# We plan to tackle these risks by maintaining proper communication between team members as well as the clients. Through proper communication, we can avoid project risks.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Module** | **Activity** | **Assignee** | **Planne d start sate** | **Planne d end date** | **Actual start date** | **Actual end date** | **Status** |
| **Requirement Gathering** | understanding the objectives and scope of work | Ponnu, gyanesh, Varha | 01/02/  2021 | 06/02/  2021 | 25/01/  2021 | 03/02/  2021 | Done |
| **Validation** | make requirements more specifics | Ponnu, gyanesh, Varha | 07/02/  2021 | 20/02/  2021 | 04/02/  2021 | 15/02/  2021 | Done |
| **Predevelopmen t** | determining project management model | Ponnu, gyanesh, Varha | 21/02/  2021 | 27/02/  2021 | 15/02/  2021 | 23/02/  2021 | Done |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Implementation** | Developing and creating the project | Ponnu, gyanesh, Varha | 28/02/  2021 | 10/04/  2021 | 23/02/  2021 | - | Ongoin g |
| **QA/QC** | Making sure quality standards are met | Ponnu, gyanesh, Varha | 01/02/  2021 | 17/04/  2021 | 25/01/  2021 | - | Ongoin g |
| **Client Acceptance** | Making sure clients are satisfied with the product | Ponnu, gyanesh, Varha | 11/04/  2021 | 17/04/  2021 | - | - | Yet to be done |
| **Deployment** | Delivering and deploying the finished product to the necessary  clients | Ponnu, gyanesh, Varha | 25/05/2021 | | - | | Yet to be done |

# **Risk Identification:**

* Major risks we have determined for this software are as follows:
* Equipment & product failure
* Late delivery of software
* Technology will not meet expectations
* End users resist system - Changes in requirements

# Deviation from software engineering standards

* Less reuse than planned
* Poor commenting & documentation of source code

## List (Describe) Register

|  |  |  |
| --- | --- | --- |
| Risk ID | Risk Description | Impact Description |
| R01 | If the product or equipment fails | No product to give to clients – Major Loss |
| R02 | Late Delivery of software | Clients lose trust and we lose reputation of  reliability |
| R03 | Expectations not met by technology | Quality of product is poor and users don’t like  what they get |
| R04 | Deviation from standards | Lose clients and users |
| R05 | Less reuse than planned | Budget can spike, Excessive work |
| R06 | Poor commenting & Documentation of source code | Can cause problems if multiple people refer/edit the same code. Future referencing  is extremely hard. |

## Managing Risk

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Risk ID** | **Status** | **Risk**  **Appetite** | **Action** | **Action**  **Owner** | **Target**  **Date** | **Remarks** |
| **R01** | **Open** | Transfer | Use better tools and devices. Use job  specialization | Manager, Quality Team | 06/04/2021 | Improve Quality of  interface |
| **R02** | **Open** | Mitigate | Keep regular meetings to avoid  any confusion. Set | Team Lead, Manager | 05/04/2021 | Weekly Meetings |
|  |  |  | short term goals and  make sure they’re  being met |  |  |  |
| **R03** | **Open** | Avoid | Try ensuring that the product is the best version of itself at  any given point | Developers, Analysts | 28/02/2021-  05/04/2021 | Regular Testing |
| **R04** | **Close** | - | Product is made along the standards  needed. | tester | 25/03/2021 | Refer to guidelines  regularly |
| **R05** | **Open** | Accept | Try planning better. Developers should communicate and  share code | Developers, Tech lead | 27/02/2021-  27/03/2021 | Code sharing online |
| **R06** | **Close** | - | Commenting and documentation is checked and done  regularly | Developers | 27/02/2021-  27/03/2021 | Regular Checking |

**Result:**

Thus, the WBS and Risk Plan was documented successfully.

# Reference

[1. https://www.pmi.org/](http://www.pmi.org/)



**DEPT. Of Computer Science Engineering**

**SRM IST, Kattankulathur – 603 203**

**Course Code: 18CSC206J**

**Course Name: Software Engineering and Project Management**

|  |  |
| --- | --- |
| **Experiment No** | 6 |
| **Title of Experiment** | Design a System Architecture, Use Case Diagram, ER Diagram (Database), DFD Diagram (process) (Upto Level 1), Class Diagram (Applied For OOPS based Project), Sequence Diagram (Applied For OOPS based Project) (Software – Rational Rose) |
| **Name of the candidate** | GYANESH SAMANTA(RA1911030010083) |
| **Team Members** | PONNU SHARMA(078). NARRA VARSHA(074) |
| **Register Number** | RA1911030010083 |
| **Date of Experiment** |  |

**Mark Split Up**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Description** | **Maximum Mark** | **Mark Obtained** |
| 1 | System Architecture with Presentation | 5 |  |
| 2 | Use Case Diagram  ER Diagram (Database),  DFD Diagram (process)  Class Diagram (Applied For OOPS based Project),  Sequence Diagram (Applied For OOPS based Project) | 5 |  |
| **Total** | | **10** |  |

**Staff Signature with date**

**Aim**

To prepare architecture and design of the system

**Team Members:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl No** | **Register No** | **Name** | **Role** |
| **1** | RA1911030010083 | GYANESH SAMANTA | **Lead** |
| **2** | RA1911030010078 | PONNU SHARMA | **Member** |
| **3** | RA1911030010074 | NARRA VARSHA | **Member** |

**Software Used**

**Star UML,** Rational Rose, Etc…

**Architecture Diagram with description**

**Use Case Diagram With Description**

**ER Diagram With Description (optional)**

**DFD Diagram (process) With Description**

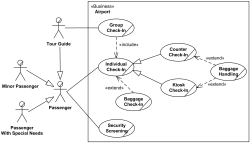
**Class Diagram (Applied For OOPS based Project),**

**Collaboration Diagram (Applied For OOPS based Project)**

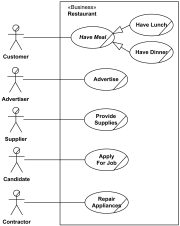
### Use Case Diagrams

#### *Business Use Case Diagrams*

Next  [**Airport check-in and security screening business model**](https://www.uml-diagrams.org/airport-checkin-uml-use-case-diagram-example.html)

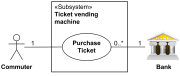
**[](https://www.uml-diagrams.org/airport-checkin-uml-use-case-diagram-example.html)**

Next  [**Restaurant business model**](https://www.uml-diagrams.org/restaurant-uml-use-case-diagram-example.html)

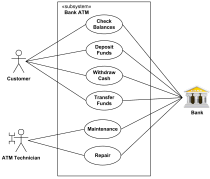
**[](https://www.uml-diagrams.org/restaurant-uml-use-case-diagram-example.html)**

#### *System Use Case Diagrams*

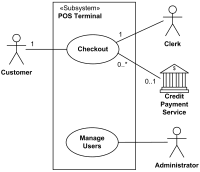
Next  [**Ticket vending machine**](https://www.uml-diagrams.org/ticket-vending-machine-use-case-diagram-example.html?context=uc-examples)

**[](https://www.uml-diagrams.org/ticket-vending-machine-use-case-diagram-example.html?context=uc-examples)**

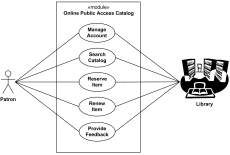
Next  [**Bank ATM UML use case diagrams examples**](https://www.uml-diagrams.org/bank-atm-uml-use-case-diagram-example.html?context=uc-examples)

**[](https://www.uml-diagrams.org/bank-atm-uml-use-case-diagram-example.html?context=uc-examples)**

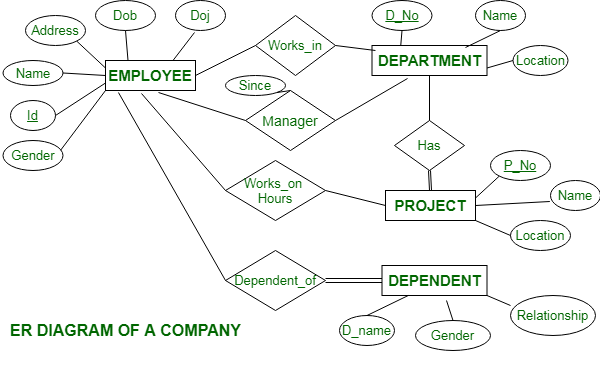
Next  [**Point of Sales (POS) terminal**](https://www.uml-diagrams.org/examples/point-of-sales-uml-use-case-diagram-example.html)

**[](https://www.uml-diagrams.org/examples/point-of-sales-uml-use-case-diagram-example.html)**

Next  [**e-Library online public access catalog (OPAC)**](https://www.uml-diagrams.org/online-library-uml-use-case-diagram-example.html)

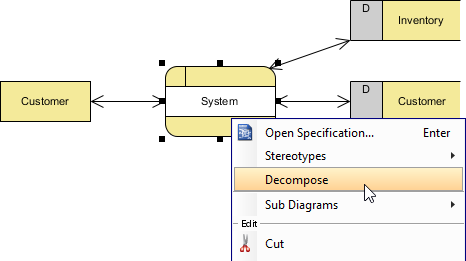
**[](https://www.uml-diagrams.org/online-library-uml-use-case-diagram-example.html)**

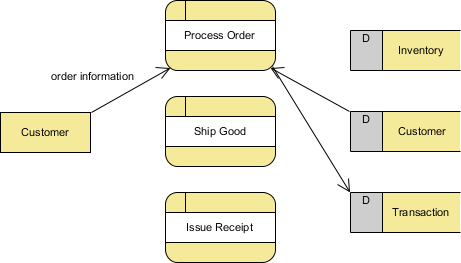
**ER Diagram of Company :**



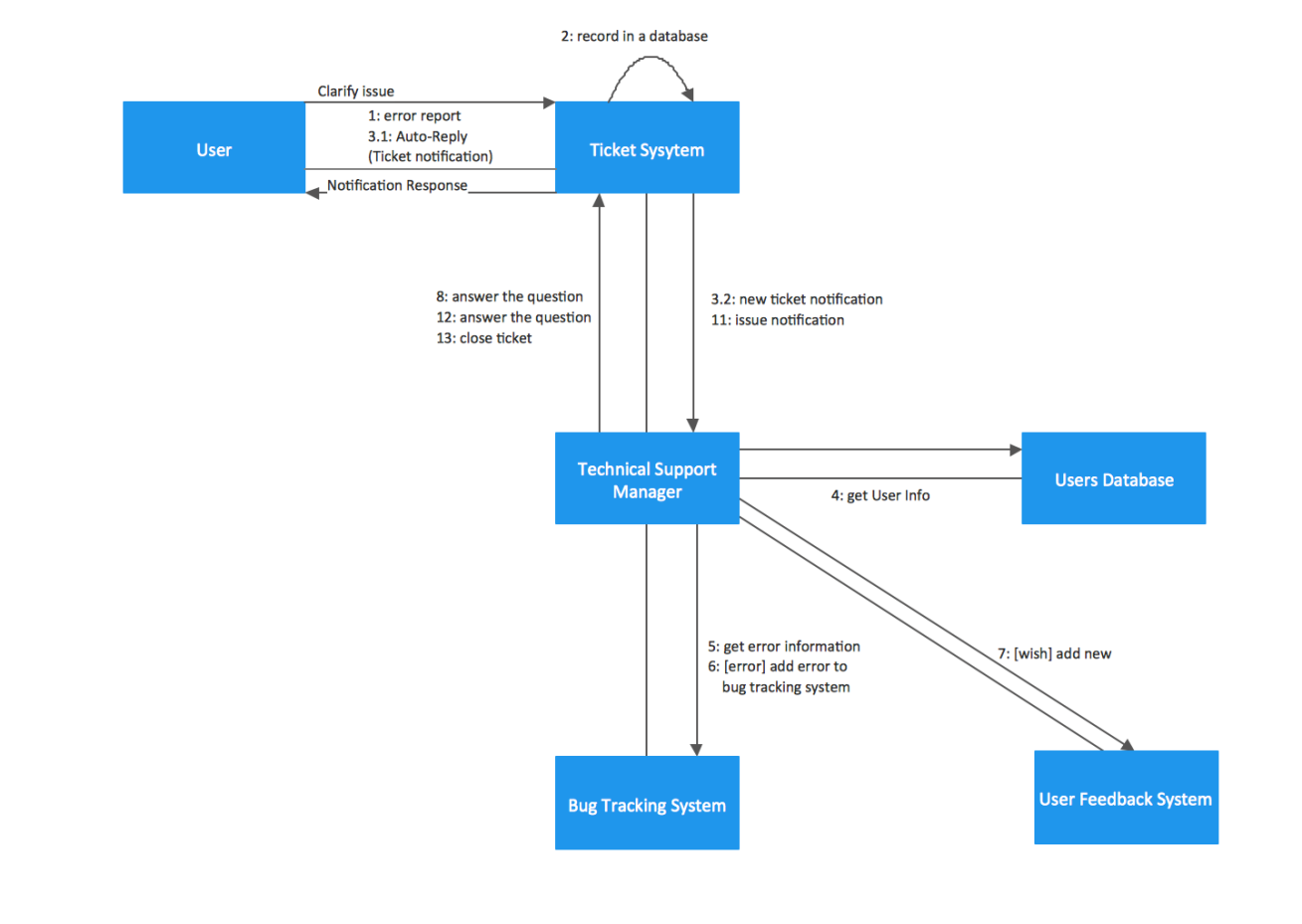
This Company ER diagram illustrates key information about Company, including entities such as employee, department, project and dependent. It allows to understand the relationships between entities.

## How to Draw Level 1 DFD?

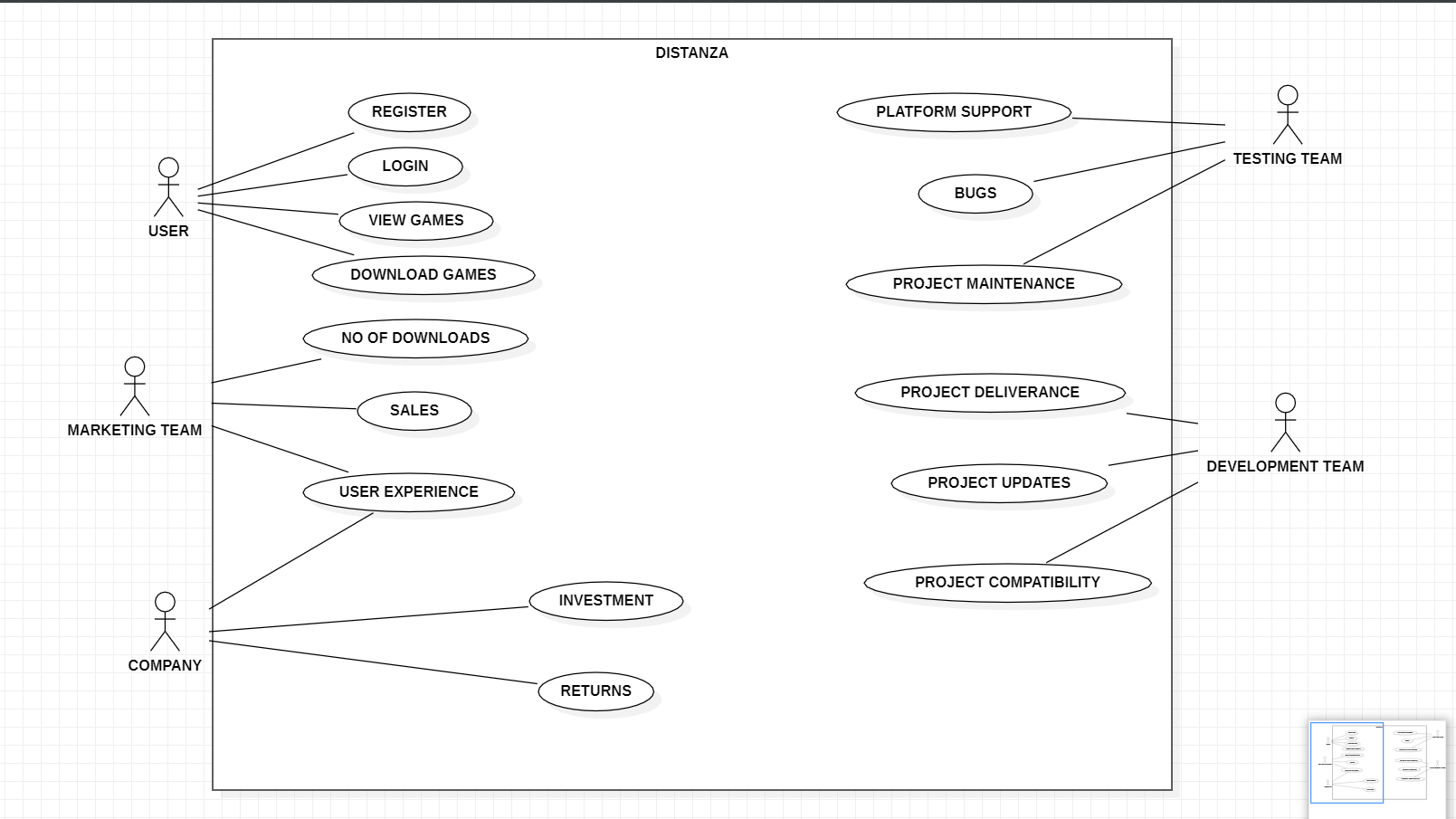
1. Instead of creating another diagram from scratch, we will decompose the *System* process to form a new DFD. Right-click on *System* and select *Decompose* from the popup menu.  
   

By combining the order information from *Customer* (external entity) and the customer information from *Customer* (data store), *Process Order* (process) then creates a transaction record in the database. Create a data flow from *Process Order* to *Transaction*.  


# [UML Collaboration Diagram (UML2.0)](https://www.conceptdraw.com/How-To-Guide/uml-collaboration-diagram)

[](https://www.conceptdraw.com/How-To-Guide/uml-collaboration-diagram)

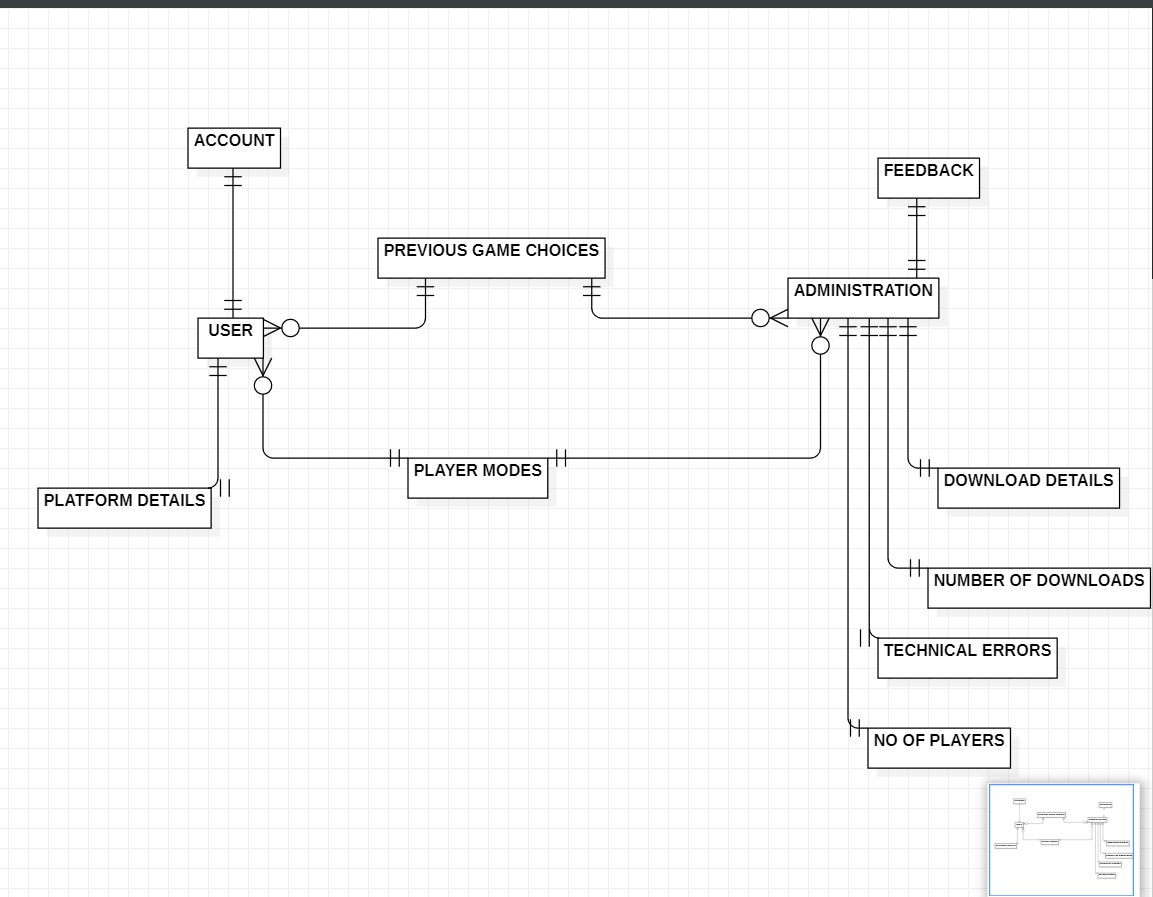
1.USE CASE DIAGRAM-



A use case diagram at its simplest is a representation of a user’s interaction with the system that shows the relationship between the actor and the different use cases

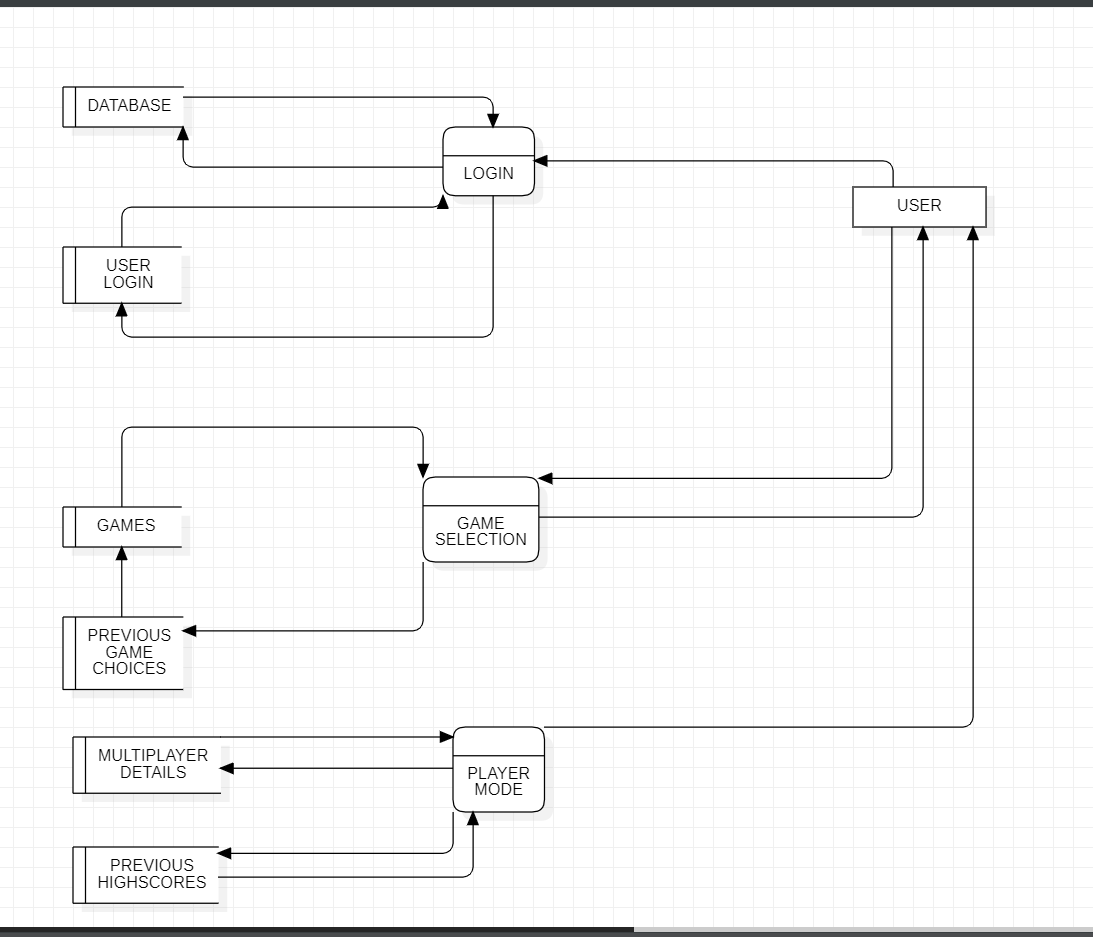
The diagram above depicts the interaction between the various teams involved in our project and the user.

1. ER DIAGRAM



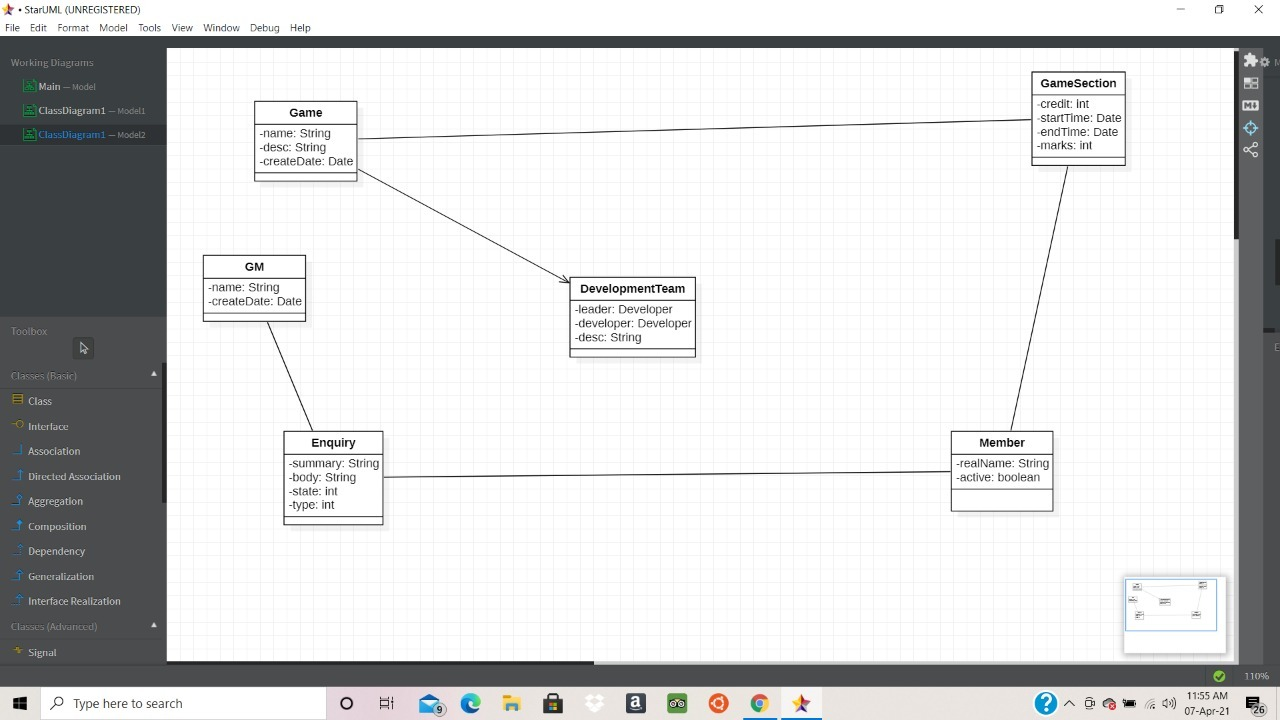
An ER Diagram stands for Entity – relationship diagram. It basically depicts the relationship of each and every entity and its relationship with the particular database.

1. DATA FLOW DIAGRAM



A Data Flow diagram is a possible way of representing the flow of data through a process or system. It also provides information about data transfer between each entity, process and the database. A data-flow diagram has no control flow and there are no decision rules.

1. Class Diagram:



Result:

Thus, architecture and design of the system was documented successfully.



## DEPT. Of Computer Science Engineering SRM IST, Kattankulathur – 603 203

**Course Code: 18CSC206J**

**Course Name: Software Engineering and Project Management**

|  |  |
| --- | --- |
| **Experiment No** | 7 |
| **Title of Experiment** | Design State , Collaboration, Deployment Diagram, Sample Frontend  Design (UI/UX) |
| **Name of the candidate** | Gyanesh Samanta |
| **Team Members** | Ponnu Sharma, Varsha Narra |
| **Register Number** | RA1911030010083 |
| **Date of Experiment** | 22nd March 2021 |

**Mark Split Up**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Description** | **Maximum Mark** | **Mark Obtained** |
| 1 | State , Collaboration diagrams | 5 |  |
| 2 | Deployment Diagram, Sample  Frontend Design (UI/UX) | 5 |  |
| **Total** | | **10** |  |

## Staff Signature with date

**Aim**

To Design State, Collaboration, Deployment Diagram, Sample Frontend Design (UI/UX) for

the project.

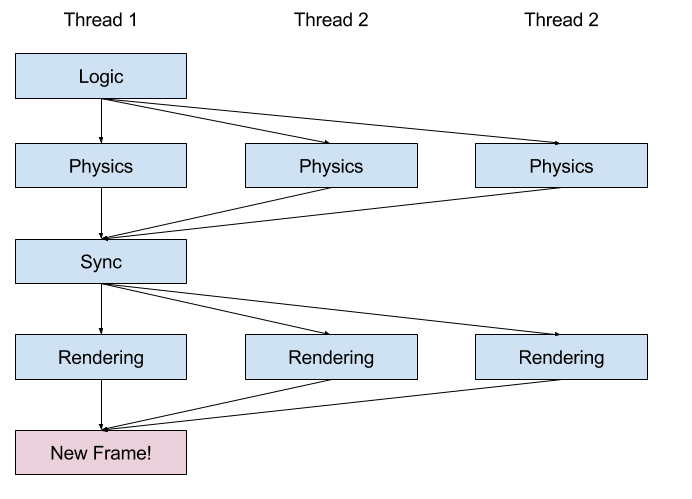
**Team Members:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl No** | **Register No** | **Name** | **Role** |
| **1** | RA1911030010083 | Gyanesh Samanta | **Lead** |
| **2** | RA1911030010078 | Ponnu Sharma | **Member** |
| **3** | RA1911030010074 | Varsha Narra | **Member** |

**Software Used**

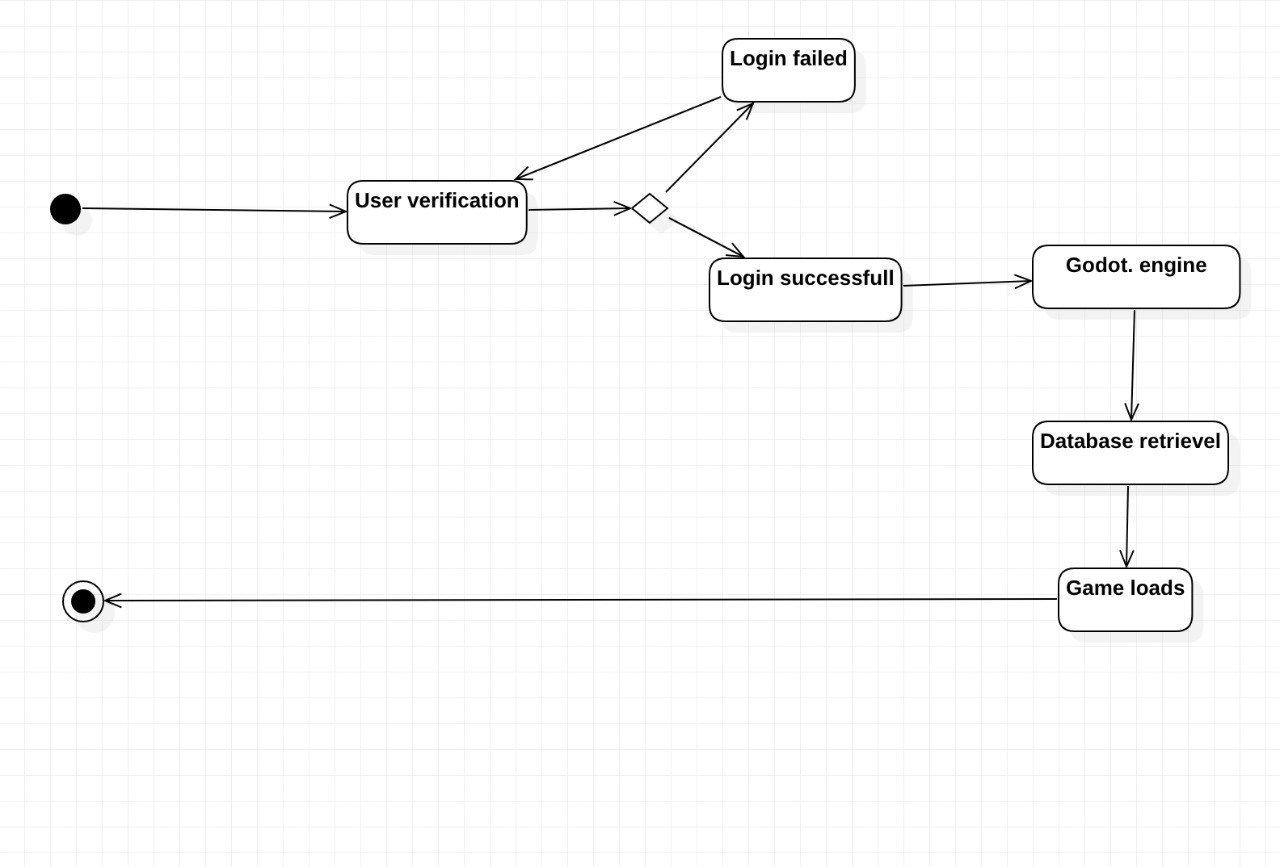
**Star UML,** Rational Rose, Etc…

# Architecture Diagram



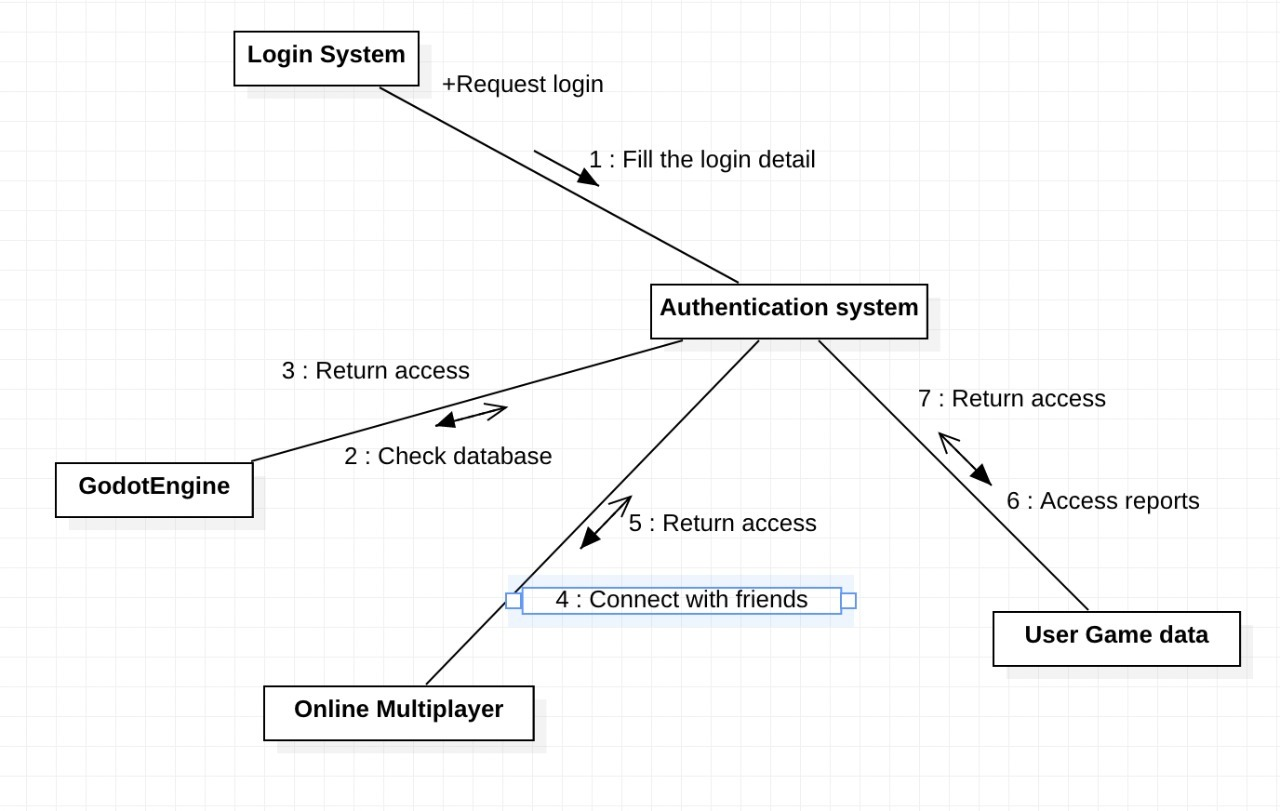
An architectural diagram is a diagram of a system that is used to abstract the overall outline of the software system and the relationships, constraints, and boundaries between components.

# State Chart Diagram



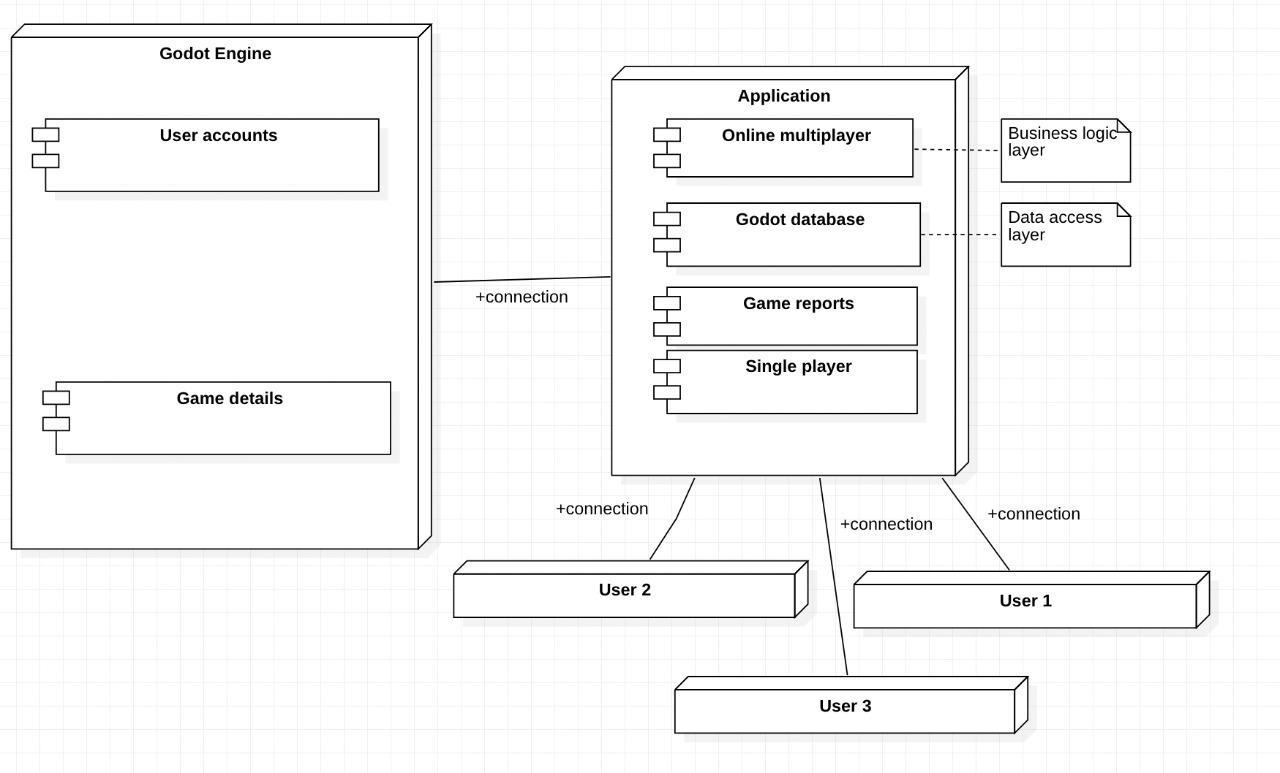
State-chart diagrams are used to describe the states of different objects in its life cycle. Emphasis is placed on the state changes upon some internal or external events. These states of objects are important to analyze and implement them accurately.

# Collaboration Diagram



Collaboration diagrams are used to show how objects interact to perform the behavior of a particular use case, or a part of a use case. Along with sequence diagrams, collaboration diagrams are used to define and clarify the roles of the objects that perform a particular flow of events of a use case. They are the primary source of information used to determining class responsibilities and interfaces.

# Deployment Diagram



Deployment diagrams are important for visualizing, specifying, and documenting embedded, client/server, and distributed systems and also for managing executable systems through forward and reverse engineering. It’s a special kind of class diagram, which focuses on a system's nodes.

**Result:**

Thus, above mentioned designs of the system were documented successfully.



**DEPT. Of Computer Science Engineering**

**SRM IST, Kattankulathur – 603 203**

**Course Code: 18CSC206J**

**Course Name: Software Engineering and Project Management**

|  |  |
| --- | --- |
| **Experiment No** | 8 |
| **Title of Experiment** | Module Description, Module Implementation |
| **Name of the candidate** | Gyanesh Samantha |
| **Team Members** | Narra Varsha, Ponnu Sharma |
| **Register Number** | Ra1911030010083, Ra1911030010078, Ra1911030010074 |
| **Date of Experiment** |  |

**Mark Split Up**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Description** | **Maximum Mark** | **Mark Obtained** |
| 1 | Implementation of module 1 | 5 |  |
| 2 | Output | 5 |  |
| **Total** | | **10** |  |

**Staff Signature with date**

**Aim**

To describe modules and implement Module1

**Team Members:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl No** | **Register No** | **Name** | **Role** |
| **1** | **Ra1911030010083** | **Gyanesh Samanta** | **Lead** |
| **2** | **Ra1911030010078** | **Ponnu Sharma** | **Member** |
| **3** | **Ra1911030010074** | **Narra Varsha** | **Member** |

**Software Used-**

**Python**

**Code of Module 1-**

import turtle

import time

import random

delay = 0.1

score = 0

high\_score = 0

# Creating a window screen

wn = turtle.Screen()

wn.title("Boa Consumer")

wn.bgcolor("Turquoise")

# the width and height can be put as user's choice

wn.setup(width=600, height=600)

wn.tracer(0)

# head of the snake

head = turtle.Turtle()

head.shape("square")

head.color("red")

head.penup()

head.goto(0, 0)

head.direction = "Stop"

# food in the game

food = turtle.Turtle()

colors = random.choice(['red', 'green', 'black'])

shapes = random.choice(['square', 'triangle', 'circle'])

food.speed(0)

food.shape(shapes)

food.color(colors)

food.penup()

food.goto(0, 100)

pen = turtle.Turtle()

pen.speed(0)

pen.shape("square")

pen.color("white")

pen.penup()

pen.hideturtle()

pen.goto(0, 250)

pen.write("Score : 0 High Score : 0", align="center",

font=("candara", 24, "bold"))

# assigning key directions

def goup():

if head.direction != "down":

head.direction = "up"

def godown():

if head.direction != "up":

head.direction = "down"

def goleft():

if head.direction != "right":

head.direction = "left"

def goright():

if head.direction != "left":

head.direction = "right"

def move():

if head.direction == "up":

y = head.ycor()

head.sety(y+20)

if head.direction == "down":

y = head.ycor()

head.sety(y-20)

if head.direction == "left":

x = head.xcor()

head.setx(x-20)

if head.direction == "right":

x = head.xcor()

head.setx(x+20)

wn.listen()

wn.onkeypress(goup, "w")

wn.onkeypress(godown, "s")

wn.onkeypress(goleft, "a")

wn.onkeypress(goright, "d")

segments = []

# Main Gameplay

while True:

wn.update()

if head.xcor() > 290 or head.xcor() < -290 or head.ycor() > 290 or head.ycor() < -290:

time.sleep(1)

head.goto(0, 0)

head.direction = "Stop"

colors = random.choice(['red', 'blue', 'green'])

shapes = random.choice(['square', 'circle'])

for segment in segments:

segment.goto(1000, 1000)

segments.clear()

score = 0

delay = 0.1

pen.clear()

pen.write("Score : {} High Score : {} ".format(

score, high\_score), align="center", font=("candara", 24, "bold"))

if head.distance(food) < 20:

x = random.randint(-270, 270)

y = random.randint(-270, 270)

food.goto(x, y)

# Adding segment

new\_segment = turtle.Turtle()

new\_segment.speed(0)

new\_segment.shape("square")

new\_segment.color("orange") # tail colour

new\_segment.penup()

segments.append(new\_segment)

delay -= 0.001

score += 10

if score > high\_score:

high\_score = score

pen.clear()

pen.write("Score : {} High Score : {} ".format(

score, high\_score), align="center", font=("candara", 24, "bold"))

# Checking for head collisions with body segments

for index in range(len(segments)-1, 0, -1):

x = segments[index-1].xcor()

y = segments[index-1].ycor()

segments[index].goto(x, y)

if len(segments) > 0:

x = head.xcor()

y = head.ycor()

segments[0].goto(x, y)

move()

for segment in segments:

if segment.distance(head) < 20:

time.sleep(1)

head.goto(0, 0)

head.direction = "stop"

colors = random.choice(['red', 'blue', 'green'])

shapes = random.choice(['square', 'circle'])

for segment in segments:

segment.goto(1000, 1000)

segment.clear()

score = 0

delay = 0.1

pen.clear()

pen.write("Score : {} High Score : {} ".format(

score, high\_score), align="center", font=("candara", 24, "bold"))

time.sleep(delay)

wn.mainloop()

**Result of Module 1**

****

Result:

Thus, modules are described, Module 1 was implemented and documented successfully.



**DEPT. Of Computer Science Engineering**

**SRM IST, Kattankulathur – 603 203**

**Course Code: 18CSC206J**

**Course Name: Software Engineering and Project Management**

|  |  |
| --- | --- |
| **Experiment No** | 8 |
| **Title of Experiment** | Module Description, Module Implementation |
| **Name of the candidate** | Gyanesh Samantha |
| **Team Members** | Narra Varsha, Ponnu Sharma |
| **Register Number** | Ra1911030010083, Ra1911030010078, Ra1911030010074 |
| **Date of Experiment** |  |

**Mark Split Up**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Description** | **Maximum Mark** | **Mark Obtained** |
| 1 | Implementation of module 1 | 5 |  |
| 2 | Output | 5 |  |
| **Total** | | **10** |  |

**Staff Signature with date**

**Aim**

To describe modules and implement Module1

**Team Members:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl No** | **Register No** | **Name** | **Role** |
| **1** | **Ra1911030010083** | **Gyanesh Samanta** | **Lead** |
| **2** | **Ra1911030010078** | **Ponnu Sharma** | **Member** |
| **3** | **Ra1911030010074** | **Narra Varsha** | **Member** |

**Software Used-**

**Python**

**Code of Module 2-**

**from turtle import \***

**from utilities import \***

**from random import choice**

**path = Turtle()**

**writer = Turtle()**

**aim = vector(5, 0)**

**pacman = vector(-40, -80)**

**ghosts = [**

**[vector(-180, 160), vector(5, 0)],**

**[vector(-180, -160), vector(0, 5)],**

**[vector(100, 160), vector(0, -5)],**

**[vector(100, -160), vector(-5, 0)],**

**]**

**state = {'score': 0}**

**tiles = [**

**0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,**

**0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0,**

**0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0,**

**0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0,**

**0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0,**

**0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0,**

**0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,**

**0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0,**

**0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0,**

**0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0,**

**0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0,**

**0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0,**

**0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,**

**0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0,**

**0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0,**

**0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0,**

**0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,**

**0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0,**

**0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,**

**0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,**

**]**

**def square(x, y):**

**"Draw square using path at (x, y)."**

**path.hideturtle()**

**path.up()**

**path.goto(x, y)**

**path.down()**

**path.begin\_fill()**

**for count in range(4):**

**path.forward(20)**

**path.left(90)**

**path.end\_fill()**

**def offset(point):**

**"Return offset of point in tiles."**

**x = (floor(point.x, 20) + 200) / 20**

**y = (180 - floor(point.y, 20)) / 20**

**index = int(x + y \* 20)**

**return index**

**def valid(point):**

**"Return True if point is valid in tiles."**

**index = offset(point)**

**if tiles[index] == 0:**

**return False**

**index = offset(point + 19)**

**if tiles[index] == 0:**

**return False**

**return point.x % 20 == 0 or point.y % 20 == 0**

**# Add your code here**

**def world():**

**Screen().bgcolor('black')**

**path.color('blue')**

**for index in range(len(tiles)):**

**tile = tiles[index]**

**if tile > 0:**

**x = (index % 20) \* 20 - 200**

**y = 180 - (index // 20) \* 20**

**square(x, y)**

**if tile == 1:**

**path.up()**

**path.goto(x + 10, y + 10)**

**path.dot(2, 'white')**

**update()**

**def move():**

**writer.clear()**

**writer.write(state['score'])**

**clear()**

**if valid(pacman + aim):**

**pacman.move(aim)**

**index = offset(pacman)**

**if tiles[index] == 1:**

**tiles[index] = 2**

**state['score'] += 1**

**x = (index % 20) \* 20 - 200**

**y = 180 - (index // 20) \* 20**

**square(x, y)**

**up()**

**goto(pacman.x + 10, pacman.y + 10)**

**dot(20, 'yellow')**

**for point, course in ghosts:**

**if valid(point + course):**

**point.move(course)**

**else:**

**options = [**

**vector(5, 0),**

**vector(-5, 0),**

**vector(0, 5),**

**vector(0, -5),**

**]**

**plan = choice(options)**

**course.x = plan.x**

**course.y = plan.y**

**up()**

**goto(point.x + 10, point.y + 10)**

**dot(20, 'red')**

**update()**

**for point, course in ghosts:**

**if abs(pacman - point) < 20:**

**return**

**Screen().ontimer(move, 100)**

**def change(x, y):**

**"Change pacman aim if valid."**

**if valid(pacman + vector(x, y)):**

**aim.x = x**

**aim.y = y**

**Screen().setup(420, 420, 370, 0)**

**Screen().tracer(0, 0)**

**writer.hideturtle()**

**writer.goto(160, 160)**

**writer.color('white')**

**writer.write(state['score'])**

**Screen().listen()**

**hideturtle()**

**Screen().onkey(lambda: change(5, 0), 'Right')**

**Screen().onkey(lambda: change(-5, 0), 'Left')**

**Screen().onkey(lambda: change(0, 5), 'Up')**

**Screen().onkey(lambda: change(0, -5), 'Down')**

**world()**

**move()**

**done()**

**import collections**

**import math**

**import os**

**def floor(value, size, offset=200):**

**"""Floor of `value` given `size` and `offset`.**

**The floor function is best understood with a diagram of the number line::**

**-200 -100 0 100 200**

**<--|--x--|-----|--y--|--z--|-->**

**The number line shown has offset 200 denoted by the left-hand tick mark at**

**-200 and size 100 denoted by the tick marks at -100, 0, 100, and 200. The**

**floor of a value is the left-hand tick mark of the range where it lies. So**

**for the points show above: ``floor(x)`` is -200, ``floor(y)`` is 0, and**

**``floor(z)`` is 100.**

**>>> floor(10, 100)**

**0.0**

**>>> floor(120, 100)**

**100.0**

**>>> floor(-10, 100)**

**-100.0**

**>>> floor(-150, 100)**

**-200.0**

**>>> floor(50, 167)**

**-33.0**

**"""**

**return float(((value + offset) // size) \* size - offset)**

**def path(filename):**

**"Return full path to `filename` in freegames module."**

**filepath = os.path.realpath(\_\_file\_\_)**

**dirpath = os.path.dirname(filepath)**

**fullpath = os.path.join(dirpath, filename)**

**return fullpath**

**def line(a, b, x, y):**

**"Draw line from `(a, b)` to `(x, y)`."**

**import turtle**

**turtle.up()**

**turtle.goto(a, b)**

**turtle.down()**

**turtle.goto(x, y)**

**def square(x, y, size, name):**

**"""Draw square at `(x, y)` with side length `size` and fill color `name`.**

**The square is oriented so the bottom left corner is at (x, y).**

**"""**

**import turtle**

**turtle.up()**

**turtle.goto(x, y)**

**turtle.down()**

**turtle.color(name)**

**turtle.begin\_fill()**

**for count in range(4):**

**turtle.forward(size)**

**turtle.left(90)**

**turtle.end\_fill()**

**class vector():**

**"""Two-dimensional vector.**

**Vectors can be modified in-place.**

**>>> v = vector(0, 1)**

**>>> v.move(1)**

**>>> v**

**vector(1, 2)**

**>>> v.rotate(90)**

**>>> v**

**vector(-2.0, 1.0)**

**"""**

**# pylint: disable=invalid-name**

**PRECISION = 6**

**\_\_slots\_\_ = ('\_x', '\_y', '\_hash')**

**def \_\_init\_\_(self, x, y):**

**"""Initialize vector with coordinates: x, y.**

**>>> v = vector(1, 2)**

**>>> v.x**

**1**

**>>> v.y**

**2**

**"""**

**self.\_hash = None**

**self.\_x = round(x, self.PRECISION)**

**self.\_y = round(y, self.PRECISION)**

**@property**

**def x(self):**

**"""X-axis component of vector.**

**>>> v = vector(1, 2)**

**>>> v.x**

**1**

**>>> v.x = 3**

**>>> v.x**

**3**

**"""**

**return self.\_x**

**@x.setter**

**def x(self, value):**

**if self.\_hash is not None:**

**raise ValueError('cannot set x after hashing')**

**self.\_x = round(value, self.PRECISION)**

**@property**

**def y(self):**

**"""Y-axis component of vector.**

**>>> v = vector(1, 2)**

**>>> v.y**

**2**

**>>> v.y = 5**

**>>> v.y**

**5**

**"""**

**return self.\_y**

**@y.setter**

**def y(self, value):**

**if self.\_hash is not None:**

**raise ValueError('cannot set y after hashing')**

**self.\_y = round(value, self.PRECISION)**

**def \_\_hash\_\_(self):**

**"""v.\_\_hash\_\_() -> hash(v)**

**>>> v = vector(1, 2)**

**>>> h = hash(v)**

**>>> v.x = 2**

**Traceback (most recent call last):**

**...**

**ValueError: cannot set x after hashing**

**"""**

**if self.\_hash is None:**

**pair = (self.x, self.y)**

**self.\_hash = hash(pair)**

**return self.\_hash**

**def \_\_len\_\_(self):**

**"""v.\_\_len\_\_() -> len(v)**

**>>> v = vector(1, 2)**

**>>> len(v)**

**2**

**"""**

**return 2**

**def \_\_getitem\_\_(self, index):**

**"""v.\_\_getitem\_\_(v, i) -> v[i]**

**>>> v = vector(3, 4)**

**>>> v[0]**

**3**

**>>> v[1]**

**4**

**>>> v[2]**

**Traceback (most recent call last):**

**...**

**IndexError**

**"""**

**if index == 0:**

**return self.x**

**if index == 1:**

**return self.y**

**raise IndexError**

**def copy(self):**

**"""Return copy of vector.**

**>>> v = vector(1, 2)**

**>>> w = v.copy()**

**>>> v is w**

**False**

**"""**

**type\_self = type(self)**

**return type\_self(self.x, self.y)**

**def \_\_eq\_\_(self, other):**

**"""v.\_\_eq\_\_(w) -> v == w**

**>>> v = vector(1, 2)**

**>>> w = vector(1, 2)**

**>>> v == w**

**True**

**"""**

**if isinstance(other, vector):**

**return self.x == other.x and self.y == other.y**

**return NotImplemented**

**def \_\_ne\_\_(self, other):**

**"""v.\_\_ne\_\_(w) -> v != w**

**>>> v = vector(1, 2)**

**>>> w = vector(3, 4)**

**>>> v != w**

**True**

**"""**

**if isinstance(other, vector):**

**return self.x != other.x or self.y != other.y**

**return NotImplemented**

**def \_\_iadd\_\_(self, other):**

**"""v.\_\_iadd\_\_(w) -> v += w**

**>>> v = vector(1, 2)**

**>>> w = vector(3, 4)**

**>>> v += w**

**>>> v**

**vector(4, 6)**

**>>> v += 1**

**>>> v**

**vector(5, 7)**

**"""**

**if self.\_hash is not None:**

**raise ValueError('cannot add vector after hashing')**

**elif isinstance(other, vector):**

**self.x += other.x**

**self.y += other.y**

**else:**

**self.x += other**

**self.y += other**

**return self**

**def \_\_add\_\_(self, other):**

**"""v.\_\_add\_\_(w) -> v + w**

**>>> v = vector(1, 2)**

**>>> w = vector(3, 4)**

**>>> v + w**

**vector(4, 6)**

**>>> v + 1**

**vector(2, 3)**

**>>> 2.0 + v**

**vector(3.0, 4.0)**

**"""**

**copy = self.copy()**

**return copy.\_\_iadd\_\_(other)**

**\_\_radd\_\_ = \_\_add\_\_**

**def move(self, other):**

**"""Move vector by other (in-place).**

**>>> v = vector(1, 2)**

**>>> w = vector(3, 4)**

**>>> v.move(w)**

**>>> v**

**vector(4, 6)**

**>>> v.move(3)**

**>>> v**

**vector(7, 9)**

**"""**

**self.\_\_iadd\_\_(other)**

**def \_\_isub\_\_(self, other):**

**"""v.\_\_isub\_\_(w) -> v -= w**

**>>> v = vector(1, 2)**

**>>> w = vector(3, 4)**

**>>> v -= w**

**>>> v**

**vector(-2, -2)**

**>>> v -= 1**

**>>> v**

**vector(-3, -3)**

**"""**

**if self.\_hash is not None:**

**raise ValueError('cannot subtract vector after hashing')**

**elif isinstance(other, vector):**

**self.x -= other.x**

**self.y -= other.y**

**else:**

**self.x -= other**

**self.y -= other**

**return self**

**def \_\_sub\_\_(self, other):**

**"""v.\_\_sub\_\_(w) -> v - w**

**>>> v = vector(1, 2)**

**>>> w = vector(3, 4)**

**>>> v - w**

**vector(-2, -2)**

**>>> v - 1**

**vector(0, 1)**

**"""**

**copy = self.copy()**

**return copy.\_\_isub\_\_(other)**

**def \_\_imul\_\_(self, other):**

**"""v.\_\_imul\_\_(w) -> v \*= w**

**>>> v = vector(1, 2)**

**>>> w = vector(3, 4)**

**>>> v \*= w**

**>>> v**

**vector(3, 8)**

**>>> v \*= 2**

**>>> v**

**vector(6, 16)**

**"""**

**if self.\_hash is not None:**

**raise ValueError('cannot multiply vector after hashing')**

**elif isinstance(other, vector):**

**self.x \*= other.x**

**self.y \*= other.y**

**else:**

**self.x \*= other**

**self.y \*= other**

**return self**

**def \_\_mul\_\_(self, other):**

**"""v.\_\_mul\_\_(w) -> v \* w**

**>>> v = vector(1, 2)**

**>>> w = vector(3, 4)**

**>>> v \* w**

**vector(3, 8)**

**>>> v \* 2**

**vector(2, 4)**

**>>> 3.0 \* v**

**vector(3.0, 6.0)**

**"""**

**copy = self.copy()**

**return copy.\_\_imul\_\_(other)**

**\_\_rmul\_\_ = \_\_mul\_\_**

**def scale(self, other):**

**"""Scale vector by other (in-place).**

**>>> v = vector(1, 2)**

**>>> w = vector(3, 4)**

**>>> v.scale(w)**

**>>> v**

**vector(3, 8)**

**>>> v.scale(0.5)**

**>>> v**

**vector(1.5, 4.0)**

**"""**

**self.\_\_imul\_\_(other)**

**def \_\_itruediv\_\_(self, other):**

**"""v.\_\_itruediv\_\_(w) -> v /= w**

**>>> v = vector(2, 4)**

**>>> w = vector(4, 8)**

**>>> v /= w**

**>>> v**

**vector(0.5, 0.5)**

**>>> v /= 2**

**>>> v**

**vector(0.25, 0.25)**

**"""**

**if self.\_hash is not None:**

**raise ValueError('cannot divide vector after hashing')**

**elif isinstance(other, vector):**

**self.x /= other.x**

**self.y /= other.y**

**else:**

**self.x /= other**

**self.y /= other**

**return self**

**def \_\_truediv\_\_(self, other):**

**"""v.\_\_truediv\_\_(w) -> v / w**

**>>> v = vector(1, 2)**

**>>> w = vector(3, 4)**

**>>> w / v**

**vector(3.0, 2.0)**

**>>> v / 2**

**vector(0.5, 1.0)**

**"""**

**copy = self.copy()**

**return copy.\_\_itruediv\_\_(other)**

**def \_\_neg\_\_(self):**

**"""v.\_\_neg\_\_() -> -v**

**>>> v = vector(1, 2)**

**>>> -v**

**vector(-1, -2)**

**"""**

**# pylint: disable=invalid-unary-operand-type**

**copy = self.copy()**

**copy.x = -copy.x**

**copy.y = -copy.y**

**return copy**

**def \_\_abs\_\_(self):**

**"""v.\_\_abs\_\_() -> abs(v)**

**>>> v = vector(3, 4)**

**>>> abs(v)**

**5.0**

**"""**

**return (self.x \*\* 2 + self.y \*\* 2) \*\* 0.5**

**def rotate(self, angle):**

**"""Rotate vector counter-clockwise by angle (in-place).**

**>>> v = vector(1, 2)**

**>>> v.rotate(90)**

**>>> v == vector(-2, 1)**

**True**

**"""**

**if self.\_hash is not None:**

**raise ValueError('cannot rotate vector after hashing')**

**radians = angle \* math.pi / 180.0**

**cosine = math.cos(radians)**

**sine = math.sin(radians)**

**x = self.x**

**y = self.y**

**self.x = x \* cosine - y \* sine**

**self.y = y \* cosine + x \* sine**

**def \_\_repr\_\_(self):**

**"""v.\_\_repr\_\_() -> repr(v)**

**>>> v = vector(1, 2)**

**>>> repr(v)**

**'vector(1, 2)'**

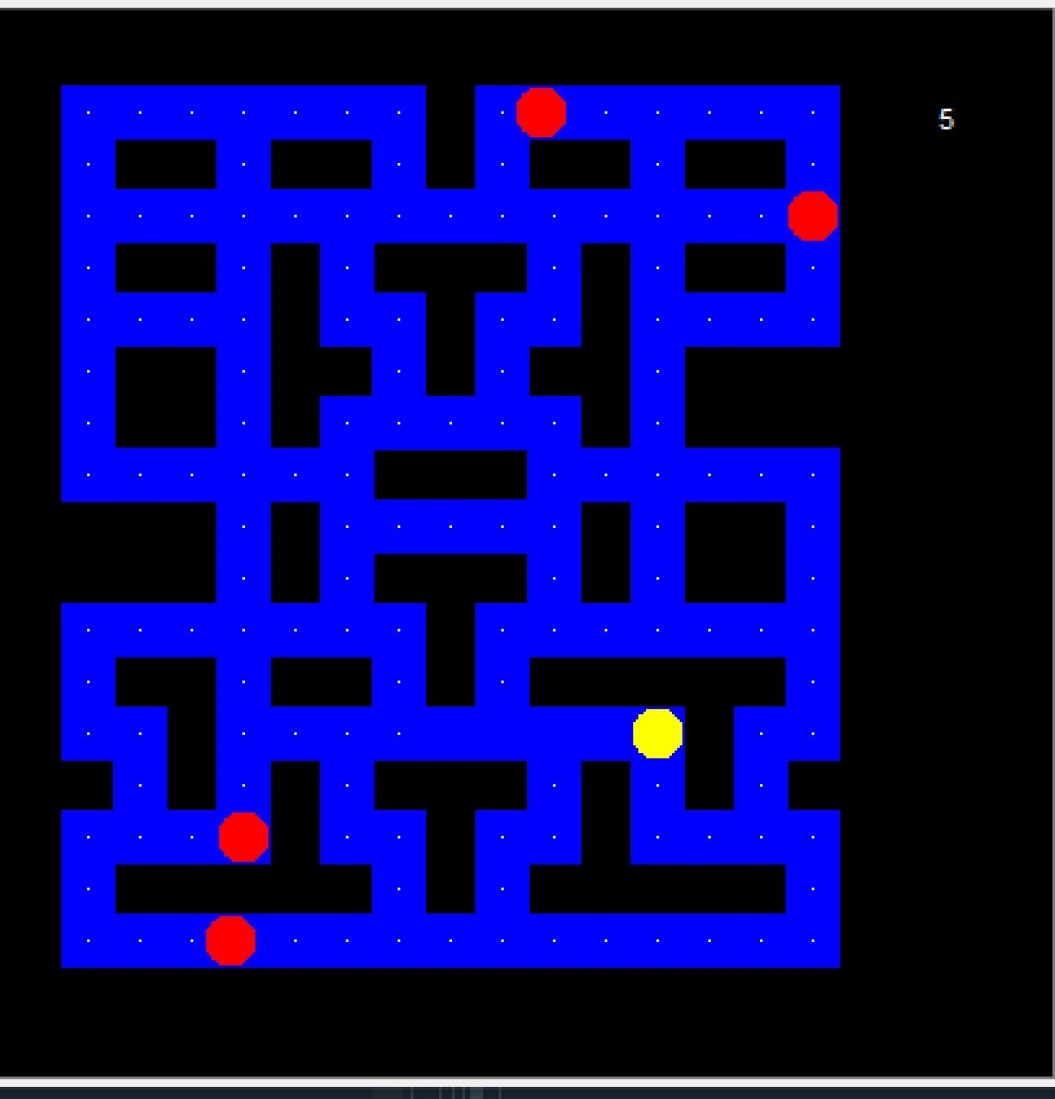
**"""**

**type\_self = type(self)**

**name = type\_self.\_\_name\_\_**

**return '{}({!r}, {!r})'.format(name, self.x, self.y)**

**Result of Module 2**



Result:

Thus, modules are described, Module 1 was implemented and documented successfully.



**DEPT. Of Computer Science Engineering**

**SRM IST, Kattankulathur – 603 203**

**Course Code: 18CSC206J**

**Course Name: Software Engineering and Project Management**

|  |  |
| --- | --- |
| **Experiment No** | 10 |
| **Title of Experiment** | Distanza Implementation |
| **Name of the candidate** | Gyanesh Samanta |
| **Team Members** | Ponnu Sharma, Narra Varsha |
| **Register Number** | Ra1911030010083, Ra1911030010078, Ra1911030010074 |
| **Date of Experiment** |  |

**Mark Split Up**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Description** | **Maximum Mark** | **Mark Obtained** |
| 1 | Module 3 | 5 |  |
| 2 | Output | 5 |  |
| **Total** | | **10** |  |

**Staff Signature with date**

**Aim**

To implement Module 3 of the project and display the output of the module with solving New Issues.

**Team Members:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl No** | **Register No** | **Name** | **Role** |
| **1** | **RA1911030010083** | **Gyanesh Samanta** | **Lead** |
| **2** | **RA1911030010078** | **Ponnu Sharma** | **Member** |
| **3** | **RA1911030010074** | **Narra Varshas** | **Member** |

**Software Used-**

**Godot Engine, NodeJs**

**Code of Module 3-**

extends Node2D

export (PackedScene) var Food

onready var message = get\_node("/root/GroceryScene/CanvasLayer/Message")

var start\_position = ""

var list\_complete = false

# Declare member variables here.

var food\_types = ["apples","bananas","broccoli", "cake", "cheese", "chocolate",

"croissants", "eggs", "fish", "garlic", "grapes", "ice cream", "kiwis", "milk",

"oranges", "potatoes", "pumpkin", "shrimp", "strawberries", "tea", "tomatoes",

"turkey"]

signal can\_move

puppet var foods\_on\_screen = []

# Spawns a new player, using the provided player\_info and the given spawn index

remote func spawn\_players(pinfo, spawn\_index):

# If the spawn index is -1 we define it by the size of the player list

if spawn\_index == -1:

spawn\_index = Network.players.size()

if (get\_tree().is\_network\_server() && pinfo.net\_id != 1):

#If this is the server host and the requested spawn point is not the server's

#Iterate through the connected players

var s\_index = 1

for id in Network.players:

#Spawn existing players within the new player's scene

if (id != pinfo.net\_id):

rpc\_id(pinfo.net\_id, "spawn\_players", Network.players[id], s\_index)

# Add the new player to the existing player scenes (except for the server's)

# The server already knows about the new player and that object can get itself

if (id != 1):

rpc\_id(id, "spawn\_players", pinfo, spawn\_index)

s\_index += 1

# Load a new scene instalce

var pclass = load(pinfo.actor\_path)

var nactor = pclass.instance()

#Set the starting position for the new actor

nactor.position = $SpawnPoints.get\_node(str(spawn\_index)).position

start\_position = nactor.position

#If this actor does not belong to the server, give it to a peer as a puppet

if (pinfo.net\_id != 1):

nactor.set\_network\_master(pinfo.net\_id)

nactor.set\_name(str(pinfo.net\_id))

#Finally, put the new actor into the world

add\_child(nactor)

remote func despawn\_player(pinfo):

if (get\_tree().is\_network\_server()):

for id in Network.players:

# Skip disconnecting the server from the replciation process

if (id == pinfo.net\_id || id == 1):

continue

# Otherwise, tell everyone else about the removals

rpc\_id(id, "despawn\_player", pinfo)

var player\_node = get\_node(str(pinfo.net\_id))

if not player\_node:

print("Cannot remove invalid node from player tree")

return

# Mark the player node for deletion

player\_node.queue\_free()

func \_on\_player\_removed(pinfo):

despawn\_player(pinfo)

func \_on\_player\_list\_changed():

pass

remote func spawn\_food(food\_items):

Food = load("res://Food.tscn")

if (get\_tree().is\_network\_server()):

if len(gamestate.food\_list) == 0:

var limits = get\_viewport\_rect().size

var food\_item\_list = []

var food\_name\_list = []

while len(food\_item\_list) < 18:

var food\_type = food\_types[randi() % food\_types.size()]

if !(food\_type in food\_name\_list):

var food\_loc = Vector2(rand\_range(0, limits.x), rand\_range(0, limits.y))

var spawn\_id = len(food\_item\_list) + 1

food\_item\_list.append({

'food\_type': food\_type,

'x': food\_loc.x,

'y': food\_loc.y,

'spawn\_id': spawn\_id

})

food\_name\_list.append(food\_type)

gamestate.update\_food\_list(food\_item\_list)

food\_items = gamestate.food\_list

rpc("spawn\_food", food\_items)

if gamestate.spawned\_food == 0:

if len(gamestate.food\_list) == 0:

gamestate.update\_food\_list(food\_items)

for food\_item in gamestate.food\_list:

var food = Food.instance()

food.set\_food\_type(food\_item['food\_type'])

food.position = $FoodSpawns.get\_node(str(food\_item['spawn\_id'])).position

add\_child(food)

gamestate.spawned\_food +=1

remote func sync\_bots():

var bot\_count = len(gamestate.bot\_info)

if (get\_tree().is\_network\_server()):

# Relay this to the connected players

rpc("sync\_bots")

while gamestate.spawned\_bots < bot\_count:

var bot\_data = gamestate.bot\_info[gamestate.spawned\_bots+1]

var bot\_class = load(bot\_data["actor\_path"])

var nbot = bot\_class.instance()

nbot.set\_name(bot\_data.name)

add\_child(nbot)

gamestate.enemy\_list.append(nbot)

gamestate.spawned\_bots += 1

# Called when the node enters the scene tree for the first time.

func \_ready():

$BeforeGameTimer.start()

# Connect to listen for when the player list changes

Network.connect("player\_list\_changed", self, "\_on\_player\_list\_changed")

Network.connect("game\_won", self, '\_on\_game\_won')

Network.connect("game\_lost", self, '\_on\_game\_lost')

# If we are the server, we want to listen for player removals too

if (get\_tree().is\_network\_server()):

Network.connect("player\_removed", self, "\_on\_player\_removed")

func init\_world():

# Spawn the players on the map

if (get\_tree().is\_network\_server()):

spawn\_players(gamestate.player\_info, 1)

sync\_bots()

spawn\_food([])

else:

rpc\_id(1, "spawn\_players", gamestate.player\_info, -1)

rpc\_id(1, "sync\_bots")

rpc("spawn\_food",[])

var time = 60

var time\_to\_start = 3

func \_on\_player\_disconnected(id):

get\_node(str(id)).queue\_free()

func \_on\_server\_disconnected():

get\_tree().change\_scene('res://LobbyMenu.tscn')

func \_on\_game\_won():

message.show\_message("Everyone made it!")

$GameExitTimer.start()

$GameWon.play()

func \_on\_game\_lost(pinfo):

$GameExitTimer.start(8)

message.show\_message("Gameover: \n"+ str(pinfo['name']) + " didn't distance.")

$GameLost.play()

# Called every frame. 'delta' is the elapsed time since the previous frame.

#func \_process(delta):

# pass

# Called every frame. 'delta' is the elapsed time since the previous frame.

#func \_process(delta):

# pass

func end\_game():

get\_tree().change\_scene("res://LobbyMenu.tscn")

self.queue\_free()

func \_on\_CountdownTimer\_timeout():

time = time - 1

if time == 0:

$CountdownTimer.stop()

$GameLost.play()

#message.show\_message("Out of time!")

Network.player\_lose(gamestate.player\_info)

message.update\_time(time)

func \_on\_UI\_zero\_health():

$CountdownTimer.stop()

$GameLost.play()

Network.player\_lose(gamestate.player\_info)

#message.show\_message("No health!")

func \_on\_BeforeGameTimer\_timeout():

message.show\_message(str(time\_to\_start) + "...")

if time\_to\_start == 0:

message.show\_message("Get your groceries!")

$GameStart.play()

$BeforeGameTimer.stop()

$CountdownTimer.start()

gamestate.end\_countdown()

time\_to\_start = time\_to\_start - 1

func \_on\_Area2D\_area\_entered(area):

if list\_complete:

$GameWon.play()

Network.player\_win(gamestate.player\_info['net\_id'])

message.show\_message("You made it out!")

#message.show\_game\_win()

func \_on\_UI\_list\_complete():

list\_complete = true

message.show\_message("Now find the exit!")

func \_on\_GameExitTimer\_timeout():

end\_game()

**Result of Module 3**

Diagram

Description automatically generated

Qr code

Description automatically generated

Qr code

Description automatically generated

Result:

Thus, the module3 was implemented and documented successfully.



### DEPT. Of Computer Science Engineering SRM IST, Kattankulathur – 603 203

**Course Code: 18CSC206J**

**Course Name: Software Engineering and Project Management**

|  |  |
| --- | --- |
| **Experiment No** | 11 |
| **Title of Experiment** | Master Test Plan, Test Case Design |
| **Name of the candidate** | Gyanesh Samanta |
| **Team Members** | Ponnu Sharma  Varsha Narra |
| **Register Number** | RA1911030010083 |
| **Date of Experiment** | 29/04/2021 |

**Mark Split Up**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Description** | **Maximum Mark** | **Mark Obtained** |
| 1 | Test Plan | 5 |  |
| 2 | Test Case | 5 |  |
| **Total** | | **10** |  |

### Staff Signature with date

**Aim**

To Prepare master test plan and Test cases for testing the project

**Team Members:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl No** | **Register No** | **Name** | **Role** |
| **1** | **RA1911030010083** | **Gyanesh Samanta** | **Lead** |
| **2** | **RA1911030010078** | **Ponnu Sharma** | **Member** |
| **3** | **RA1911030010074** | **Varsha Narra** | **Member** |

# Executive Summary

## Scope:

Through this testing procedure we aim to check the basic functionalities and services of the games that we developed. By “check” we mean whether a user is able to use the feature optimally without any prior knowledge of how the application was built. In this way we also verify if the goals that were set in the beginning of the project are met successfully or not.

**Objective:**

The objective of this testing is to ensure that the application is working as expected and users are not facing any issues while using the product. If certain bugs are detected during the testing phase, the team aims to fix them accordingly.

**Approach:**

We plan to test our application by executing it under various circumstances and running it under numerous test cases. If the final product is able to withstand and pass all the given test cases, then it is ready to be served.

1. **Test Plan**
   1. **Scope of testing**

This testing procedure covers all the functionalities and features that were mentioned in the initial description of the project. Various test cases are designed in order to check if the functionalities are being executed successfully or not. It also includes running the software in different environments and observing the behaviour of the product.

* 1. **Types of testing, Methodology and Tools**

|  |  |  |
| --- | --- | --- |
| **Category** | **Methodology** | **Tools required** |
| Functional requirements | We are testing the functional features of the products manually under various environments | 1. A system with Windows 7 or higher |
| Non functional requirements | The non functional features (eg. Game loading time, result fetching time, availability etc.) | 1. Sleep functionality 2. Standby mode |

* 1. **Test Deliverables**

### Test Deliverables:

The following are the Test Cases for the Project Investment WebApp:

-> Running the games

-> Compatibility with various GPUs

-> To use the keyboard and mouse for control

### Defect log:

The defects that exist in the project are:

-> User progress is saved locally and can be lost with a game reset

-> Users are not verified.

### Test Report:

-> User can play the games on any windows machine running windows 7 or higher

# Test case

* 1. **Functional Test Cases**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test id** | **Test Scenario** | **Test case** | **Executio n steps** | **Expecte d outcome** | **Actual outcome** | **Status** | **Remarks** |
| FT1 | The user should be able to login and start playing the games in single player or multiplayer mode as intended | Testing it on various operating systems with different system specifications | Running the game | The game loads and runs normally | The game loads successfully | Success fully complet ed | No errors detecte d |

* 1. **Non Functional Test Cases**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test id** | **Test Scenario** | **Test case** | **Executio n steps** | **Expecte d outcome** | **Actual outcome** | **Status** | **Remarks** |
| NFT 1 | All games should load within 5 seconds | Checkin g the load time of all the games | 1. Open the exe file to run the game | The games are being loaded within 5 seconds  . | The games loaded as intended | Works |  |

**Note:**

The games were developed on a small scale with limited manpower so we, as of now are not deploying the same on server and all the test cases are being run on localhost.

**Result: Thus, the test plan and test cases are documented successfully**



**DEPT. Of Computer Science Engineering SRM IST, Kattankulathur – 603 203**

**Course Code: 18CSC206J**

**Course Name: Software Engineering and Project Management**

**Mark Split Up**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Description** | **Maximum Mark** | **Mark Obtained** |
| 1 | Manual Testing | 5 |  |
| 2 | Report | 5 |  |
| **Total** | | **10** |  |

**Staff Signature with date**

**Aim**

To conduct manual test using Test cases and prepare test report for the project

# Executive Summary

## Scope:

Through this testing procedure we aim to check the basic functionalities and services of the web application that we developed. By “check” we mean whether a user is able to use the feature optimally without any prior knowledge of how the application was built. In this way we also verify if the goals that were set in the beginning of the project are met successfully or not.

## Objective:

The objective of this testing is to ensure that the application is working as expected and users are not facing any issues while using the product. If certain bugs are detected during the testing phase, the team aims to fix them accordingly.

## Approach:

We plan to test our application by executing it under various circumstances and running it under numerous test cases. If the final product is able to withstand and pass all the given test cases, then it is ready to be served.

# Test Plan

* 1. **Scope of testing**

This testing procedure covers all the functionalities and features that were mentioned in the initial description of the project. Various test cases are designed in order to check if the functionalities are being executed successfully or not. It also includes running the software in different environments and observing the behaviour of the product.

# Types of testing, Methodology and Tools

|  |  |  |
| --- | --- | --- |
| **Category** | **Methodology** | **Tools required** |
| Functional requirements | We are testing the functional features of the products manually under various environments | 1. Web browsers |
| Non functional requirements | The non functional features (eg. page loading time, result fetching time, availability etc.) will be tested automatically using online testers such as https://web.dev/ measure/ | 1. Online load time testers (https:// web.dev/ measure/) 2. Online server providers (https:// [www.netlify.com/)](http://www.netlify.com/)) |

* 1. **Test Deliverables**

**Test Deliverables:**

The following are the Test Cases for Distanza:

-> Implementing Sign Up, Sign In and Sign Out.

-> The password strength is to be checked.

-> The email field is to be checked for validity.

-> Email with transaction receipt is implemented.

-> Reseting password.

-> Make trading guide.

-> A contact us page to get their queries addressed.

**Defect log:**

The defects that exist in the project are:

-> User after signing out gets signed in upon clicking the browser's back button/ security issues.

-> Users are not verified.

-> Contrast Errors(UI) in CSS.

**Test Report:**

-> User can Sign Up, Sign In and Sign Out.

-> The password strength is checked.

-> The email field is checked for validity.

-> User gets transaction email.

-> User can reset their password.

-> User can view the trading guide.

-> User can use the contact us page to get their queries addressed.

# Test cases

## Non Functional Test Cases

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tes t id** | **Test Scenario** | **Test case** | **Executi on steps** | **Expecte d outcom e** | **Actual outcom e** | **Status** | **Remark s** |
| NF T3 | Application should be available 24x7 | Checki ng the websit e on rando m hours whethe r its workin g optima lly or not | 1. Ope n the webs ite 2. Run   the previ ous test case s enlis ted   1. Chec k if thos e test case s are passi ng or not | The websit e should passing all the test cases as expect ed at various points of time. | The websit e is availab le 24 x  7 and passing all the test case at various points of times (we got the above results by runnin g the web site on localho st) | Partiall y imple mente d | Please refer to the **Note** section for more inform ation |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tes t id** | **Test Scenario** | **Test case** | **Executi on steps** | **Expecte d outcom e** | **Actual outcom e** | **Status** | **Remark s** |
| NF T4 | Should be easy to use for  a 3rd party | Checki ng if the websit e is easily accessi ble for a person with no prior knowle dge about the websit e | 1. Aski   ng a third part y pers on outsi de of the tea m to use the prod uct   1. Gett ing feed back on how was the user expe rienc e and if they face d any issue s | The websit e is easily accessi ble for everyo ne includi ng people with no prior knowle dge about the websit e | The websit e was easy to use and all the functio nalities were self explan atory. | Succes sfully imple mente d | N/A |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tes t id** | **Test Scenario** | **Test case** | **Executi on steps** | **Expecte d outcom e** | **Actual outcom e** | **Status** | **Remark s** |
| NF T5 | Database should be large enough  to support multiple users buying and selling multiple stocks | Inserti ng as many values as possibl e into the databa se | 1.  Insert values into the databa se and checki ng the outer bounda ries of the databa se | The databa se is big enough for 5000-1  0000  users with individ ual portfol ios | The databa se is big enough for 5000-1  0000  users with individ ual portfol ios | Succes sfully imple mente d | N/A |

1. **Defect Log**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement #** | **Defect ID#** | **Defect Description** | **Asignee** | **Status** |
| M1R1 | D1 | User after signing out gets signed in upon clicking the browser's back button/ security  issues. | ponnu | Ongoing (Trying to correct) |
| M2R2 | D2 | Users are not verified. | varsha | Closed (without success) |
| M3R3 | D3 | Contrast Errors(UI) in  CSS. | gyanesh | Closed(with success) |

1. **Test Report**

**Current Status of the Testing:**

**All the test cases have been implemented and passed successfully within the testing period. The bugs found in the process have been fixed accordingly and the final product was also used by foreign users satisfactorily.**

**Present obstacles to proceed further:**

**Currently, no existing obstacles detected. Project is ongoing as planned. Help from stakeholders to remove obstacles/constraints:**

**Each stakeholder was introduced to the final product in the testing phase and the issues were fixed based on their feedback of the experience.**

|  |  |  |
| --- | --- | --- |
| **Category** | **Progress against Plan** | **Status** |
| Functional testing | Green | Completed |
| Non functional testing | Green | Completed |

|  |  |  |
| --- | --- | --- |
| **Functional** | **Test case coverage(%)** | **Status** |
| Module 1(front-end) | 40% | Completed |
| Module 2 (database, automation, api) | 15% | Completed |
| Module 3(back- end,ML) | 45% | In progress |

## Reference

**Result: Thus, the software test conducted and documented the report successfully**