

## INDEX

NAME: Abhinav Monga

ROLL NO. 41811503118

Expt No.	Name of the Experiment	Date	Remarks
1	To make a feasibility report on online gaming platform		
2	To make an SRS document on online gaming platform		
3	To make an E-L diagram on online gaming platform		
4	Make a level 0 & level 1 DFD for online gaming platform		
5	To make a use case diagram for online gaming platform		
6	To make a state class diagram for student online gaming platform		
7	To make a class diagram for online gaming platform		
8	To make activity diagram for online gaming platform		

Any Additional Individual Work Done By Student

- 1 To study functional & non functional requirements for online gaming platform
- 2 To make data dictionary for online gaming platform

## EXPERIMENT 1

### Aim

To prepare feasibility report for an online gaming platform - Steam

### Software Used

Star UML

### Theory

A feasibility study is a way to evaluate the practicality & desirability of a project. These are usually used to sway decisions towards one direction or the other. It also determines whether or not the only one the decision makers will have to choose from. It also determines whether or not the investigated task can be done with the amount of resources available.

A feasibility may be useful in a list of different situations such as event planning, finances etc.

### Importance of Feasibility Report -

- 1) Enhance the success rate by evaluating all parameters
- 2) Identifies reasons not to proceed.
- 3) Improves team focus
- 4) Identifies risks / opportunities

# **EXPERIMENT 1:**

## **Feasibility Report**

**Objective:** To draft the feasibility of Steam- A video game digital distribution system and check whether the same is feasible or not.

**Description:** Steam is a video game digital distribution platform. It is used to distribute games and related media online and has community features such as friend lists and groups, the marketplace, in-game voice chat, and chat functionality.

- Problems in the existing system:**
- Other platforms take a large part of developers' income as commission for publishing their individual content.
  - It uses internet data extensively even for simple background operations.
  - Games are not available for the trial version
  - Game trading and reselling is not allowed.
  - Cloud Sync: The user is required to be online to play certain games(logged in to Steam)
  - Auto-updating is forced on booting the application.
  - Inbuilt streaming service is not available.
  - Games distributed by other companies are not integrated.
  - The Valve Anti-Cheat (VAC) feature is redundant and ineffective.
  - Inefficient support for glitch updates.
  - Production and management for anti-cheat software

### **Cost**

**Analysis:**

S. No.	Description	Price
<b>SOFTWARE</b>		
1.	Windows 10 Enterprise x 4	Rs. 55,999
<b>HARDWARE</b>		
1.	Logitech M525 wired Mouse x 4	Rs. 2,404 /-

2.	Logitech MK345 Keyboard x 4	Rs. 5,608 /-
3.	Dell CPU x 4	Rs. 2,88,999 /-
4.	Dell S Series Led 32" Monitor x 4	Rs. 61,999 /-
<b>MANPOWER</b>		
1.	4 PERSON	Rs. 1,000/hour
<b>TOTAL</b>		Rs. 3,59,010 /- + Manpower

- Future Scope:**
- Blockchain-based system for trading in-game items and marketplace purchases.
  - Adding a live stream feature for gamers.
  - Global competitions where all the gamers can meet.
  - Games to be made with lite edition for low-end users.
  - Common platform currency for all games, making trading and purchase easy.
  - Advertisements may be enabled by used for the generation of platform currency at low rates.
- Limitations:**
- Scamming by unfaithful traders is an issue.
  - The anti-cheat feature requires regular monitoring and improvement.
  - Inefficient VR support
- References:**
- support.steampowered.com
  - Help.steam.com
  - maketecheasier.com
  - drivereeasy.com
- Conclusion:** We have analyzed the cost and drafted the feasibility report of Steam software.

## EXPERIMENT 2

### Aim

To prepare a software requirement specification (SRS) document for online gaming platform

### Software Used

Star UML

### Theory

The production of the requirement stage of the software development process is called software requirement specification (SRS). It is a formal report, which acts as a representation of software that enables the customers to review the project is according to their requirements.

It compresses user requirements for a system as well as detailed specification of the system requirements.

### Characteristics of a good SRS

- 1) Correctness
- 2) Completeness
- 3) Consistency
- 4) Modifiable
- 5) Verifiable
- 6) Traceable

Date :

Page No.:

### Nature of SRS

- 1) Functionality - What the software needs to do
- 2) External Interface - How does software interact with environment.
- 3) Design Constraints imposed on implementation

### Result

SRS document was successfully prepared

# **Software Requirement Specifications**

For  
**Digital Gaming System**

**STEAM**

Prepared by:

Abhinav Mangla (41811503118)  
Aakash Mangla (41411503118)

Bharati Vidyapeeth's College of Engineering, New Delhi  
October 2020

# **1. INTRODUCTION**

## **1.1 Purpose**

The purpose of this document is to present a detailed description of the Digital Software and Games Distribution System - Steam. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be liable for the approval or disapproval of the project by the community of users and clients alike.

The aim of this project is to develop a secure digital distribution system for software with the following objectives:

- Create a software and games distribution system that is easily accessible by customers from the comfort of their homes, entertainment centres etc.
- Reduce the flow of human traffic at physical game stores.
- Reduce the time wasted in going to stores to purchase games.
- Promote efficient and effective system by focusing on the services that still require a physical method to acquire new software/games.

## **AUDIENCE:**

The different types of Audience are:-

- (a) Customers
  - Project Scope
  - Security Available
- (b) Developers (Clients)
- (c) Developers (Steam)
  - Project Scope
  - Use Case Module
- (d) Project Manager
  - System Features
  - Hardware Requirement
  - Software Requirement
  - Interface Requirement
- (e) Testers
  - Testing
- (f) Documentation Writers

## 1.2 Scope

The Scope of this project is limited to the activities of the operation of an online store which includes the opening of an account, purchase of products, refunds, updating the purchased software.

- Any game provider can use this application to provide a better service to their customers.
- Customers can access his/her account anywhere, be it a mobile device or a desktop PC.
- Clients can publish various upcoming games and plans for customers through this application.
- Reduction in the workload of all employees will be possible through this application as transaction rights are provided online to the customer.
- It can be extended for global communication between all sorts of customers in the world in the form of an online community/forum as well.

## 1.3 Definition, Acronym & Abbreviation

- **HTML:** Hypertext Markup Language is a markup language used to design static web pages.
- **EJB:** Enterprise Java Beans.
- **J2EE:** Java 2 Enterprise Edition is a programming platform part of the Java Platform for developing and running distributed multi-tier architecture Java applications, based largely on modular software components running on an application server.
- **DB2:** DB2 Database is the database management system that delivers a flexible and cost effective database platform to build robust on demand business applications.
- **WAS:** Web sphere application server is an application server that runs business applications and supports the J2EE and web services standards.
- **WSAD:** Web sphere studio application developer is a toolkit which is designed for the creation of more complex projects, providing fully dynamic web application utilizing EJB's. This consist of EJB tools, CMP, data mapping tools & a universal test client that is designed to aid the testing of EJB's.
- **HTTP:** Hypertext Transfer Protocol is a transaction oriented client/server protocol between web browser & a Web Server.
- **HTTPS:** Secure Hypertext Transfer Protocol is an HTTP over SSL (secure socket layer)
- **TCP/IP:** Transmission Control Protocol/Internet Protocol, the suite of communication protocols used to connect hosts on the Internet. TCP/IP uses several protocols, the two main ones being TCP and IP.

## 1.4 References

→ References from the following website:

1. Nancy Day, Software Requirements Specification and Analysis, Lecture 3,  
<http://www.student.cs.uwaterloo.ca/cs445/Fall2005/Schedule/l3.pdf>.
2. Software Engineering Standards Committee of the IEEE Computer Society, IEEE Recommended Practice for Software Requirements Specifications, 1998.
3. Wikipedia, the Free Encyclopedia, <http://en.wikipedia.org/>.
4. Software Requirement Specifications (SRS) for software engineering,  
http://se.uwaterloo.ca/dberry/ATRE/srs.pdf

→ References from the following textbooks:

### 1. Java Complete Reference

Author(s): Herbert Schildt

Publisher: TMH Publications

### 2. Core Java2 volume I &II

Author(s): Cay S Horstmann, Gary Cornel

Publisher: Pearson Publications

### 3. Software Engineering

Publisher: Genius Publication

### 4. Software Engineering

Author(s): K K Aggarwal & Yogesh Singh

Publisher: New Age International Publishers

## 1.5 Overview

This project is about the Digital Software and Games Distribution System - Steam. The project has complete information regarding the account details (including creating an account, purchased

products, social networking, games accessibility etcetera). It also contains information about the different customers opening their account. This project also helps to keep the information about all the details of the various customers who have opened their account and to keep account of their daily software usage and games played metrics. This project is stand alone and the various Clients or Software Publishers can make their products available on this platform to a broader audience for a 10% percentage of their total sales revenue from the platform.

## 2. OVERALL DESCRIPTION

### 2.1 Product Perspective

#### 2.1.1 System Interface

The system Interface comprises of User Interface, Hardware Interface and the Software Interface.

#### 2.1.2 Interface

There are three different ways for a user to interact with the system:

- **New User:**

A customer who has not yet opened a steam account has the option to open one in order to complete the purchase of games and software available on the platform and start building their network.

- **Existing User:**

The Existing user is the most typical user of the system. All users have their own account and registered or authorized login access. The Existing user can log in online to their account perform the purchase, refund, review of the products available on the platform. The purchases are done online and the product is available right away for downloading for the user because it saves time and it is an efficient process. The user also gets access to the social network, marketplace and all online discussion forums and groups.

- **Administrator:**

Admin is a master user of the system because they play the main role in the system. Admin grant and maintain the database of the existing user and grant the permissions to users. It overrules all other users. Admins also possess the right to control the content on a Software Product Publishers' store page in case of any discrepancies.

### **2.1.3 Hardware Interfaces**

- Client Side:

**Processor:** 2.0 GHz or better

**RAM:** 2 GB or more

**Disk Space:** 1GB or more

**Graphics:** DX10 Compatible or better.

- Server Side:

**Processor:** 2.3 GHz Xeon E5 2686 CPU

**RAM:** 32 GB or more

**Disk Space:** 10TB or more

### **2.1.4 Software Interfaces**

Operating System : Windows 7 32-bit SP1 or better

Web Browser : Google Chrome or Mozilla Firefox (Browser should be JavaScript and Flash enabled)

Front End : REACT JS

Back End : NODE JS

RDBMS : MS ACCESS

DBMS : MS-SQL SERVER

### **2.1.5 Communication Interfaces**

- Client on the Internet will be using HTTP/HTTPS protocol.
- Client on Intranet will be using TCP/IP protocol.
- A Web Browser such as IE 6.0 or equivalent

### **2.1.6 Memory Constraints**

At least 2GB of RAM and 1GB of hard disk

## **2.1.7 Operation**

The product is a digital software and games distribution system similar in functionality to the software stores. Steam software will allow for two different methods of operation. The first method of operation will be through a mobile/desktop/web-based version of the software. This operation is performed by the user. The user will be allowed to purchase, refund or review products on the platform, there will also be a forum for the community. It will all be done through a simple, easy to use graphic user interface. The second part of the software is the software publisher operations software. It is a web-based interface which will be the web page of the publisher. It will run through a PC which will be manipulated by an appointed administrator or manager via keyboard and mouse. The administrator will have an administrative interface which is a GUI so that he can view the entire system related to each and every software publisher. He will also have a login page where he can enter the login particulars so that he can perform all his actions. This administrative interface provides different environment such that he can maintain database & provide backups for the information in the database.

All functions will be accessible at any time through a menu bar at the top of the screen. These functions include: authenticating product's store page content, reviewing purchases by customers, customer's usage statistics, checking the forums regarding the product, authorizing refunds to the customer, and generating reports.

## **2.2 Product Functions**

The Internet-based system consists of the following modules:

### **1) Login Process**

→ This module allows valid customers to access the functionalities provided by the system.

### **2) Product Enquiry**

→ This module maintains the details of a particular software available for purchase.

### **3) Update Profile**

→ This module allows the customer to update the profile of their account.

### **4) Purchase Product**

→ This module allows the customers to utilize various methods such as credit card, debit card, net banking, prepaid wallets, et cetera to purchase the desired product.

→ Deposit Money- Its primary function is to allow the user to deposit money in the form of steam Cash into an account in order to perform a purchase later.

### **Actors:**

1. Customer

2. System
3. Publisher Client

#### **Main Flow:**

1. The user is presented with the main menu screen and the store page.
2. The user will request product details by clicking on the product page button on the store page.
3. The user is presented with a screen that briefs the user on the details of the software.
4. If the user wishes to purchase the product, then they will be presented with the option to either add the product to their cart or purchase right away.
5. Once the user wishes to check-out, they will be asked to complete the purchase by any of the available methods of purchase provided.
6. After the user's purchase has been authenticated properly.
  - The user will be presented with the option to continue to downloading the software right away for usage or do so at a later time.

#### 5) Change of Password

- This module allows customers to change their password.

#### 6) Mini Statements

- This module allows customers to view their transaction details and produce bills.

### **2.3 User Characteristic**

A registered user have following facilities:

- Accounts and accounts status.
- Purchase Enquiry.
- Refund Request.
- Password Changing.
- Participation in user forums.

### **2.4 Constraints**

- Login and password is used for identification of customer's account and there is no facility for non-users to login.
- This system works on multiple servers that handle the individual modules of the system, as in user details, publisher details, software details, and forums.
- This is designed Using HTML, CSS.
- Language used is JavaScript.
- GUI is only in English as of yet.
- Limited to HTTP/HTTPS protocols.

## **2.5 Assumptions & Dependencies**

### **Assumption:**

The details of customers such as username, password and other details should be manually entered by the user before using this system.

- Every user should be comfortable of working with computer and net browsing.
- He should be aware of the system's functions to provide a platform to purchase, update and manage software of other publishers at a single point.
- He must have basic knowledge of English too.

## **2.6 Apportioning Of Requirements**

There is no such service as of now which would be delayed until the future versions.

# **3. SPECIFIC REQUIREMENTS**

## **3.1 External Interfaces**

The major Interfaces: User, Hardware, Software and Communication interfaces are covered above. There are no other External Interfaces.

## **3.2 Functions**

Following are the services which this system will provide. These are the facilities and functions required by the customer.

- Online Account check.
- Online shopping opportunity.
- Online data entry by the staff.
- Updating the data.
- Balance check.
- Refund services if the product has not been used for a long period of time.
- Community Forums for users to interact with each other and the publishers.

## **3.3 Performance Requirements**

System can withstand even though many no. of customers request the desired service.

Access is given to only valid users of the system who requires the services such as purchase enquiry, update profile, refund, mini statements, et cetera.

It is available during whole week for all 24 hours.

### **3.4 Logical Database Requirements**

The following information will be placed in a database:

- (i) Recent files
- (ii) Plug-ins

### **3.5 Design Constraints**

- Login and password is used for identification of customer's account and there is no facility for non-users to login.
- This is designed Using HTML, CSS.
- Language used is JavaScript.
- GUI is only in English
- Limited to HTTP/HTTPS protocols.

### **3.6 Software System Attributes**

#### **3.6.1 Reliability**

The system is safety critical. If it moves out of normal operation mode, the user will be logged out and all the actions being performed by the user will be paused indefinitely until the user logs back in again. This emergency behavior shall not occur without reason.

The system has to be very reliable due to the importance of data and the damages incorrect or incomplete data can do.

#### **3.6.2 Availability**

When in normal operating conditions, request by a user shall be handled within 1 second. Immediate feedback of the systems activities shall be communicated to the user by the page-link clicked. At peek system load, individual users or the server, shall not experience any delay in the service response to their commands longer than 1 second.

The system is available 100% for the user and is usable 24 hrs. A day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

#### **3.6.3 Security**

There shall be security mechanisms in place to keep unwanted/malicious server pings out of the system. Allowing the system to isolate itself from any sort of maliciously injected code.

### **3.6.4 Maintainability**

There shall be design documents describing the internal works of the software. There shall be an access on the control panel and servers for the purpose of upgrading the software or flashing any firmware.

### **3.6.5 Portability**

There are no portability requirements.

## **3.7 Organizing Specific Requirement**

### **3.7.1 System Mode**

The application will run on Windows, Linux, and Mac and will be compatible with the mobile OS like Android or iOS and any web browser compatible with JavaScript.

### **3.7.2 User Class**

**Customers:** The normal users will have an account which will allow them access to the basic functions of the system. The typical customer will be a person willing to purchase a software product. There will more than likely be a fairly equal distribution of males and females with no restrictions on age although some software such as video games with mature content will require an age-barrier.

The typical customer will probably use the system a couple of times a week. The typical customer might not know a lot about computers, so the system needs to be very simple and easy to use. The typical customer will probably be a busy person; therefore, they will need to do their transactions as quickly and efficiently as possible.

**Software Publisher Clients:** These users will have the options of managing their developed products available on the platform in form of price regulation, new product publishing, previous published product management, reports regarding usage statistics of their software.

**Steam Administrator:** The administrator will be a different type of user. The admin employee is a fairly educated user, who is willing to sacrifice simplicity for functionality. They will use the software daily in order to manage the details regarding products allotted to them for overseeing. This could quite possibly be 30-60 products with each product having multiple usages by the typical users/customers. Due to this frequency of usage stability and speed of this software is incredibly important.

### 3.7.3 Objects

- **Viewers:** Many unknown persons or un-authenticated persons visit the site via internet to look at the products available for purchase. They collect the information and search what services are available in the web page. Those viewers or visitors may become the customer of the system.
- **New User:** Who all visited the webpage or heard about steam, those persons are may be getting ready to start an account. They register at the site and wait for an email confirmation for the account creation.
- **Existing User:** The Existing user is the most typical user of the system. Each Users have their own account and registered or authorized login access. The Existing user can login in online to their account perform different operations such as purchases, refunds, reviews, et cetera. All of this done online is helpful for the user because it saves time and it is an efficient process.
- **Administrator:** Admin is master user of the system because they play the main role in the system. Admin grant and maintain the database of the existing user and grant the permissions to users. It over rules all other users.

### 3.7.4 Features

Customer must have a valid User Id and password to login to the system. After the valid user logs in he is shown the list of products he/she has purchased and out which how many are currently installed on the system and whether any software updates are available for the products or not.

User can choose to either use the installed products or buy a new one from the platform. If user wishes to purchase another product, they can go to the store page and click on the product they desire thus opening the software's store page listing all the details about the software such as its function, features, its hardware and software requirements, its price and the reviews by other users who have purchased and used this product.

Upon choosing to buy the product the user will be sent to the checkout page to complete the transaction. If the transaction is successful a notification should appear to the customer via the steam system and also via email, in case it is unsuccessful, a proper message should be given to the customer as to why it failed.

User can request for refunds if they believe they couldn't utilize the product as advertised.

User can view his monthly as well as annual statements on the purchases made. He can also take print out of the same. Appropriate help to be provided as and when requested by the user.

### **3.7.5 Stimulus**

During the past decade, the spectrum of software and their functionality has broadened and so has the number of such software as well. Showing an exponential increase in the number of software available to the users.

A digital distribution system for these software is designed to handle the primary function of providing an ease to the user for purchasing of such software without physically going to multiple stores in order to search for them and also removing any worries about the updating of the software the user has installed through the system as it is automatic.

A database is maintained to handle all the details regarding the users and their purchases. This project intends to introduce more user friendliness in the various activities such as software purchase, refund, updating, reviewing, and utilizing of the products.

Another database consists of the details regarding the products such as system requirements, features, basic functions, cost, reviews, et cetera, these details are used to manage the individual store pages of such software for the user to see.

There are multiple servers optimized for storage regarding the cloud storage of software which will be available for the user to download from once the purchase has been verified.

All of this aims at digitizing the aspects of purchase and usage of different software and thus easing the load on the user.

### **3.8 Additional Comments**

NONE

## **4. CHANGE MANAGEMENT PROCESS**

It will immediately respond to the change in platform (OS). There will be two different interfaces, one for Mac users and other for Windows to make it user friendly for both the platforms. The system for mobile devices will be updated quite regularly including beta versions.

## **5. Document Approval**

This document is approved by Ms. Shafali Dhall, Professor, Department of Information Technology, Bharati Vidyapeeth's College of Engineering, Delhi.

## **6. Supporting Information**

To support the above software requirements specifications, there is a table of content:

**TABLE OF CONTENT**

TABLE OF CONTENT	
1. Introduction	2
1.1 Purpose	2
1.2 Scope	3
1.3 Definition, Acronyms and Abbreviations	3
1.4 References	4
1.5 Overview	4
2. Overall Description	5
2.1 Product Perspective	5
2.1.1 System Interfaces	5
2.1.2 Interface	5
2.1.3 Hardware Interface	5
2.1.4 Software Interface	6
2.1.5 Communication Interface	6
2.1.6 Memory Constraints	6
2.1.7 Operation	6
2.2 Product Functions	7
2.3 User Characteristics	8
2.4 Constraints	8
2.5 Assumptions and Dependencies	8
2.6 Apportioning of requirements	9
3. Specific Requirements	9

3.1 External Interfaces	9
3.2 Functions	9
3.3 Performance Requirements	9
3.4 Logical Database Requirements	10
3.5 Design Constraints	10
3.6 Software System Attributes	10
3.6.1 Reliability	10
3.6.2 Availability	10
3.6.3 Security	10
3.6.4 Maintainability	10
3.6.5 Portability	10
3.7 Organising Specific Requirements	11
3.7.1 System Mode	11
3.7.2 User class	11
3.7.3 Objects	11
3.7.4 Features	12
3.7.5 Stimulus	12
3.8 Additional Comments	13
4. Change Management Process	13
5. Document Approval	13

## EXPERIMENT 3

### Aim

To prepare an ER diagram for online gaming platform

### Software used

Star UML

### Theory

ER modelling is a data modelling method used in software engineering to produce a conceptual data model of an information system.

### Purpose

- 1) Serves as a documentation tool
- 2) The database analyst gains a better understanding of the data to be contained in the database through steps of construction of the ERD

### Components

- 1) Entity - it can be a real world object either animate or inanimate, that can be merely identifiable. An entity is denoted as a rectangle. An entity set is a collection of related types of entities. Entity set need not be defined.

- 2) Attributes - Entities are denoted utilizing their properties known

attribute. All attributes have values

- 1) Key attributes
- 2) Composite attributes
- 3) Single value
- 4) Multi valued
- 5) Derived

7 → Relationships - The association among entities is known as relationship. They are represented by the diamond shaped box

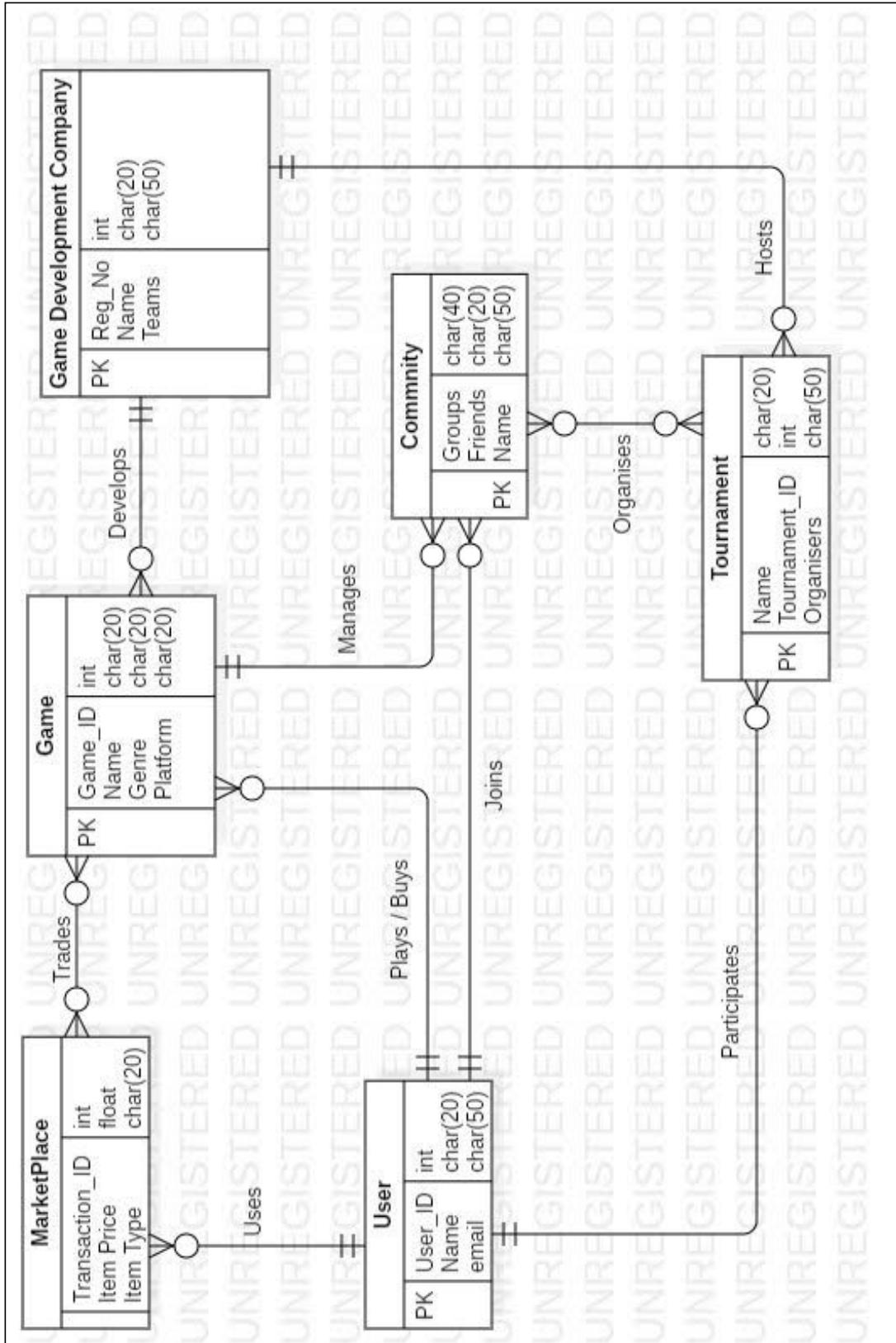
→ Cardinality - Describes the no. of entities in one entity set, which can be associated with no. of entities of other sets via relationship set.

Result

An ER diagram was successfully prepared

# EXPERIMENT - 3

ER - DIAGRAM



## EXPERIMENT 4

### Aim

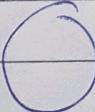
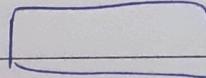
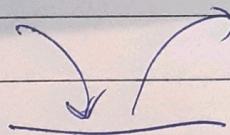
To draw data flow diagram for online gaming platform

### Software used

Star UML

### Theory

A data flow diagram (DFD) is a traditional visual representation of the information flows within a system. The objective of a DFD is to show the scope & boundaries of a system as a whole. The DFD is also called as a data flow graph or bubble chart.

	<u>Symbol</u>	<u>Name</u>	<u>Function</u>
①		Data Flow	Used to connect processes to each other
②		Process	Performs transformations of i/p to yield o/p
③		Source/sink	A system inputs by source or sink of system o/p
④		Data Store	A repe of data, the arrow heads are net inputs

### levels

There are 3 levels in the DFD

- ① 0-level DFD
- ② 1-level DFD
- ③ 2-level DFD

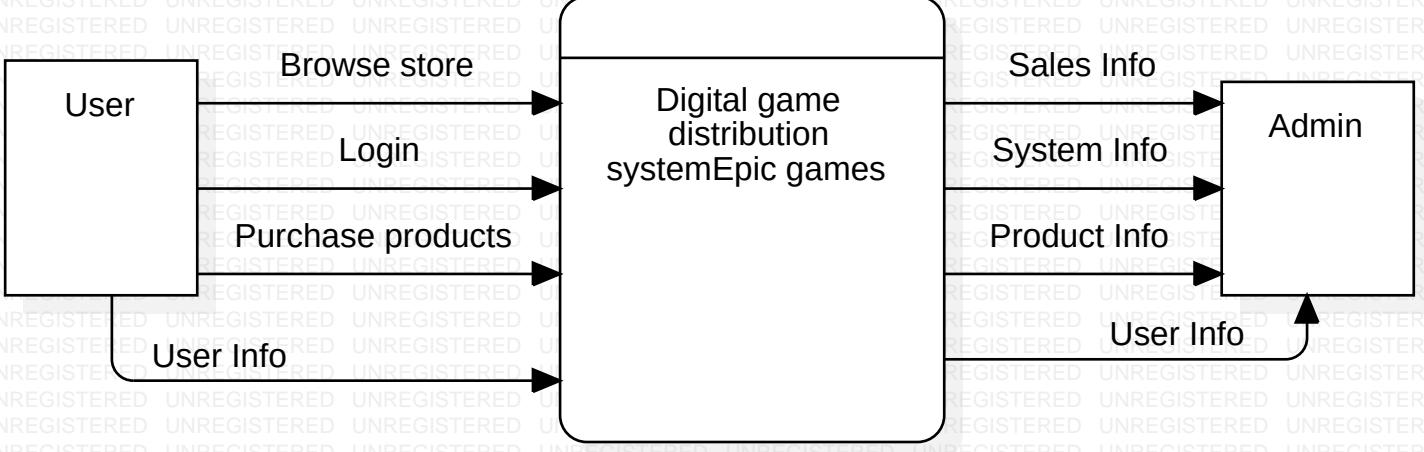
In a DFD:

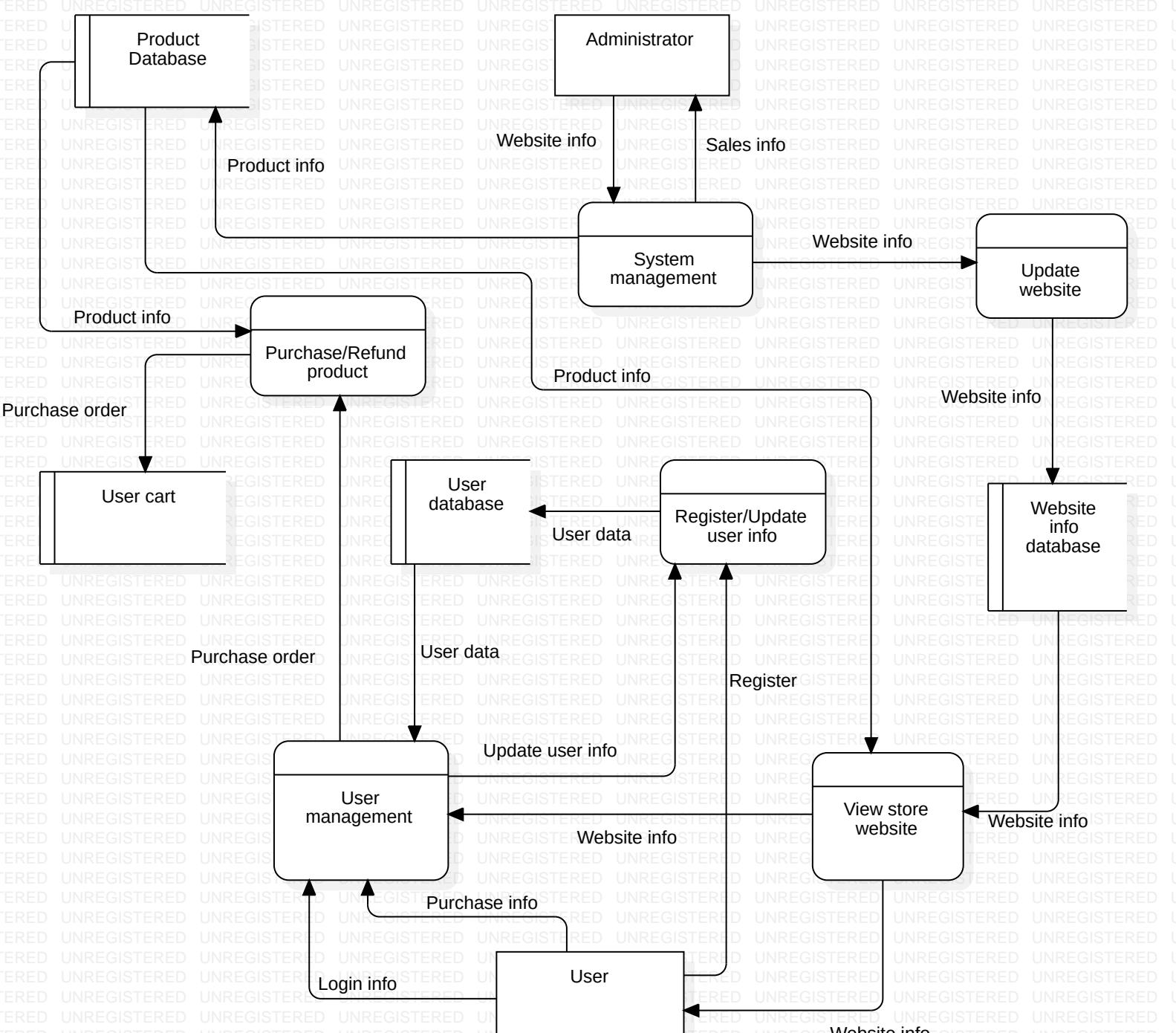
All names should be unique

Arrows in a flow chart represent the order of events.

Result

A DFD was successfully prepared.





## EXPERIMENT 5

### Aim

To design use case diagram of online gaming platform

### Software Used

Star UML

### Theory

The main purpose of a use case diagram is to portray the dynamic aspect of a system. It accumulates the system's requirements, will include both internal as well as external influences.

### Purpose

- (1) Gathers the system's needs
- (2) depicts the external view of the system.
- (3) recognises the internal as well as external factors that influence the system.
- (4) represents the interaction b/w the actors

These are graphical representation that may be decomposed into further levels of abstraction.

- (i) Actor - It is an external agent, lies outside the system model, but interacts with it in some way.

User-case - Describes the sequence of interactions b/w actor & system

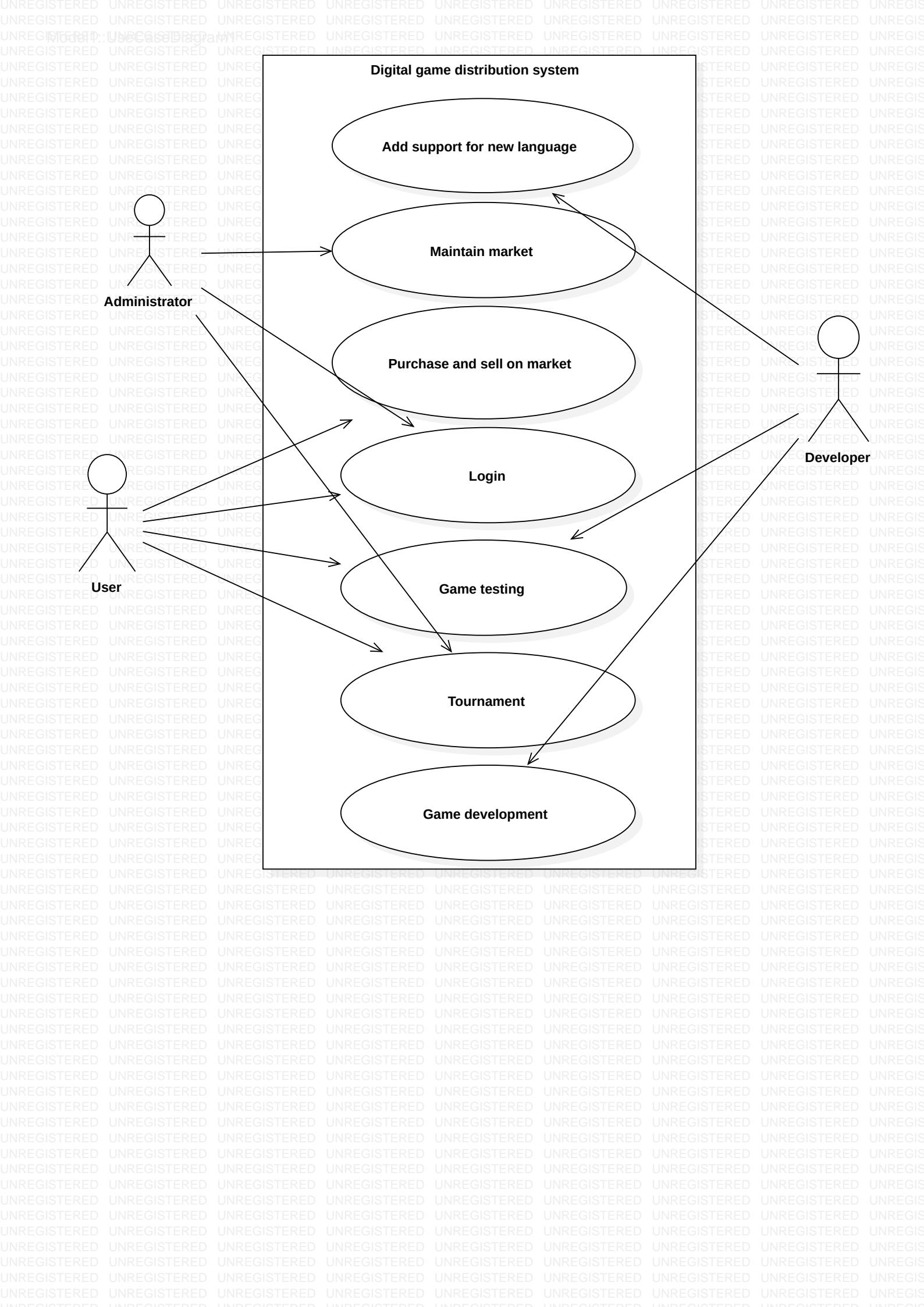
Specify the events of a system & their flows.

They can be used for:

- Requirement analysis & high level design
- Model the content of a system.
- Reverse engineering
- Forward engineering

Result

Use case diagram was successfully prepared



EXPERIMENT 6

## Aim

To design a class diagram for online gaming platform.

## Software Used

Star UML

## Theory

Class diagram is a static diagram. It represents the static view of an application. Class diagram describes the attributes & operations of a class & also the constraints imposed on the system.

## Purpose

- Analysis & design of the static view of an application.
- Describe the responsibilities of a system.
- Base of component & development diagrams
- Forward & reverse engineering

Class diagram is also considered as the foundation for component & deployment diagrams. Class diagram describes the vocabulary of the system.

Usage

- Describes the static view of the system.
- Showing the collaboration among the element of the static view.
- Describe the functionalities performed by the system.
- Construction of software application using object oriented languages.

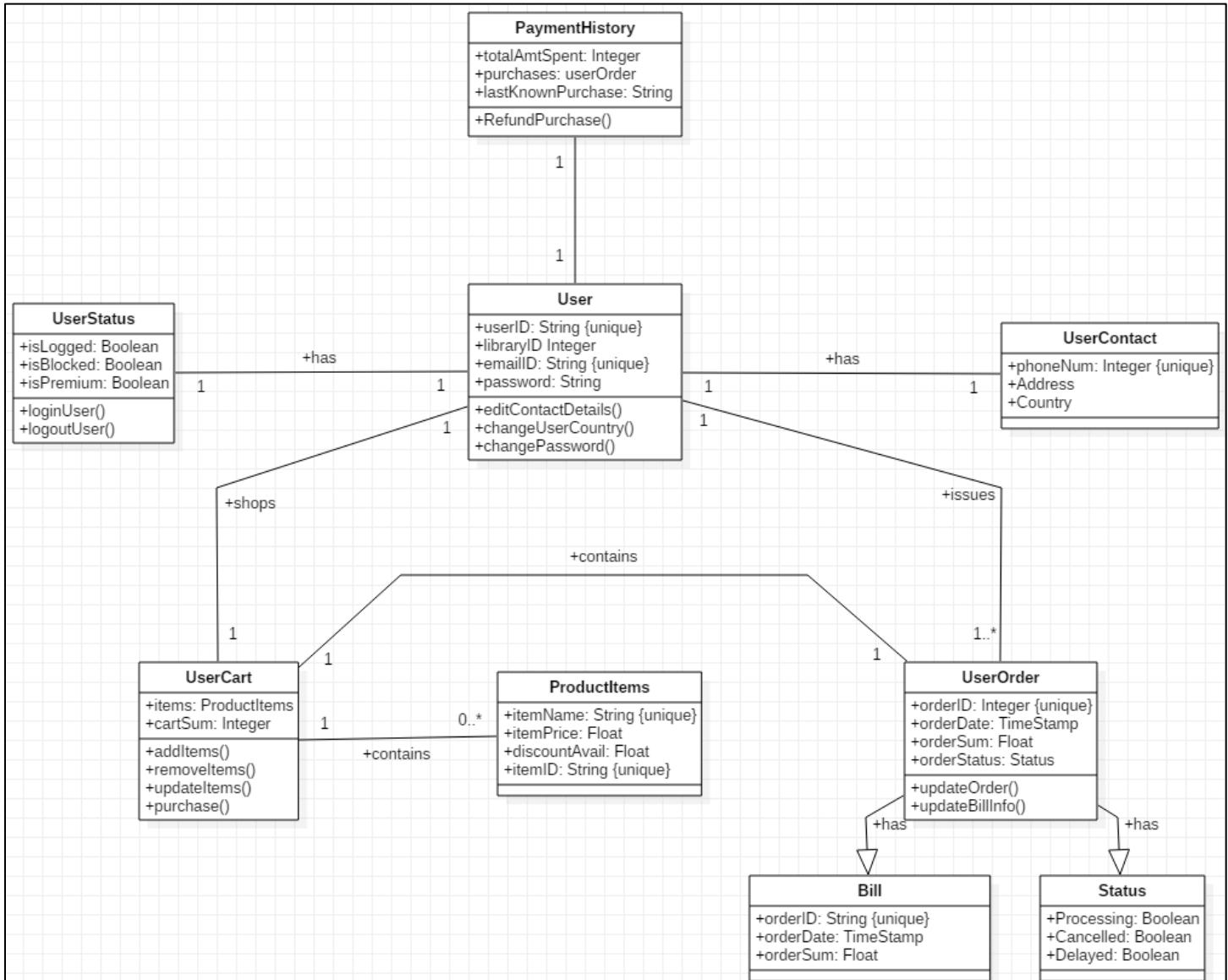
## Result

The diagram was successfully prepared

# Experiment-6

## Class Diagram

### Digital Software Distribution System- Origin



## EXPERIMENT 7

### Aim

To design state chart diagram for online gaming platforms

### Software Used

Star UML

### Theory

Describes a state machine. State machine can be defined as a machine which defines different states of object & these states are controlled by external or internal events

- 1) To model the dynamic aspect of a system
- 2) To model the life time of a reactive system.
- 3) To describe the different states of an object during its lifetime
- 4) Define a state machine

State chart diagram defines the states of a component & the state changes are dynamic in nature. Its specific purpose is to define the state changes triggered by events

### Uses

- => To model the object states of a system

- To model the reactive system
- To identify the events responsible for state changes
- Forward & reverse engineering

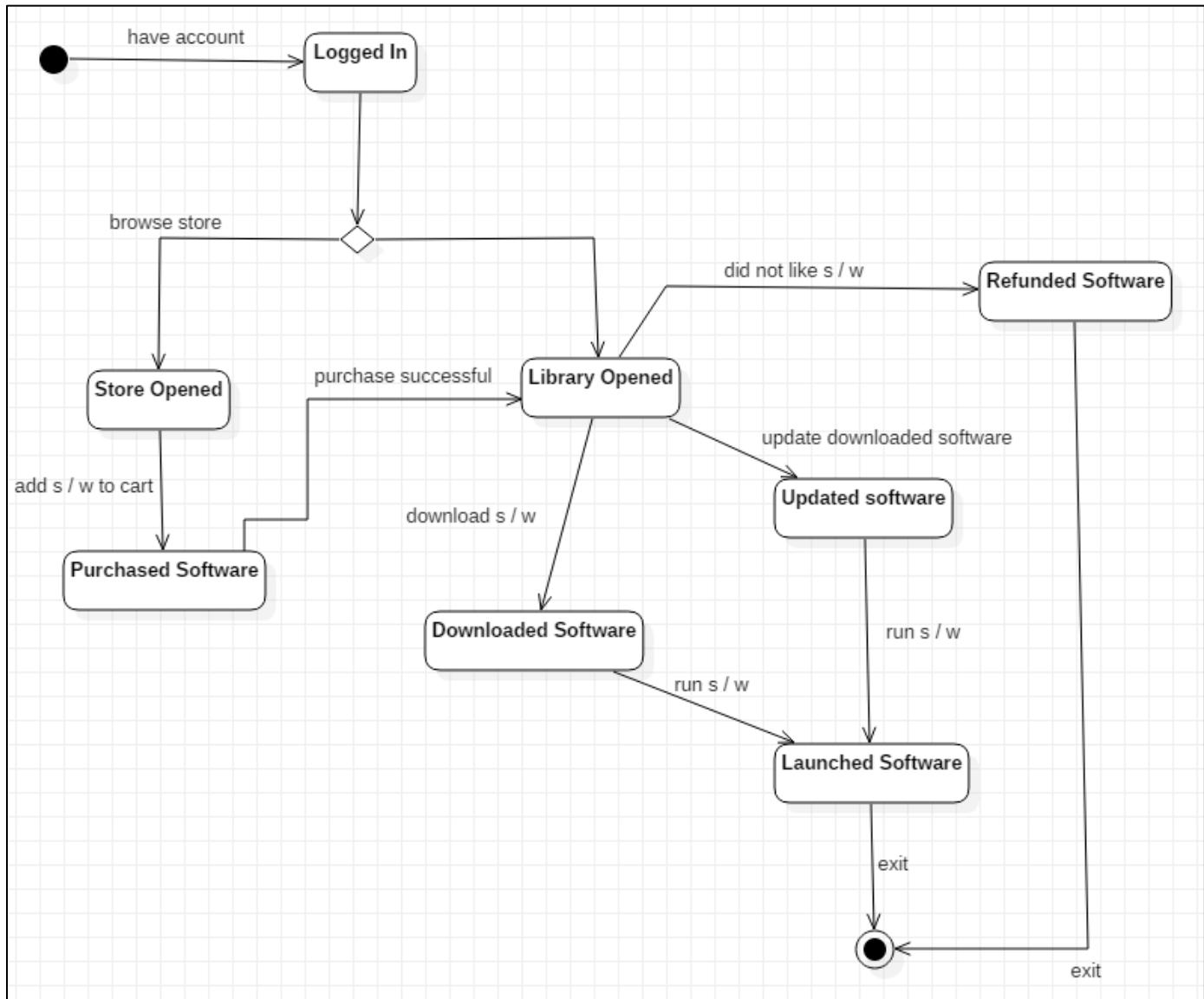
Result

- State transition diagram was successfully prepared.

# Experiment-7

## State Chart Diagram

### Digital Software Distribution System- Origin



## EXPERIMENT 8

### Aim

To make the activity diagram for online gaming platform

### Software Used

Star UML

### Theory

Activity diagram is a flowchart to represent the flow from one activity to another activity. This control flow does not include messages. The basic purpose of activity diagram is similar to other flow diagrams. It captures the dynamic behaviour of the system. Activity diagram is drawn from a very high level

### Purpose

- Shows the activity flow of a system
- Describes the sequence from one activity to another
- Describes the parallel, branched & concurrent flow of the system.
- Suitable for modelling the activity flow of the system

Activity diagram can be used for -

- Modelling work flow by using activities
- Modelling business requirements.
- High level understanding of the system's functionalities
- Ignoring business requirements at a later stage

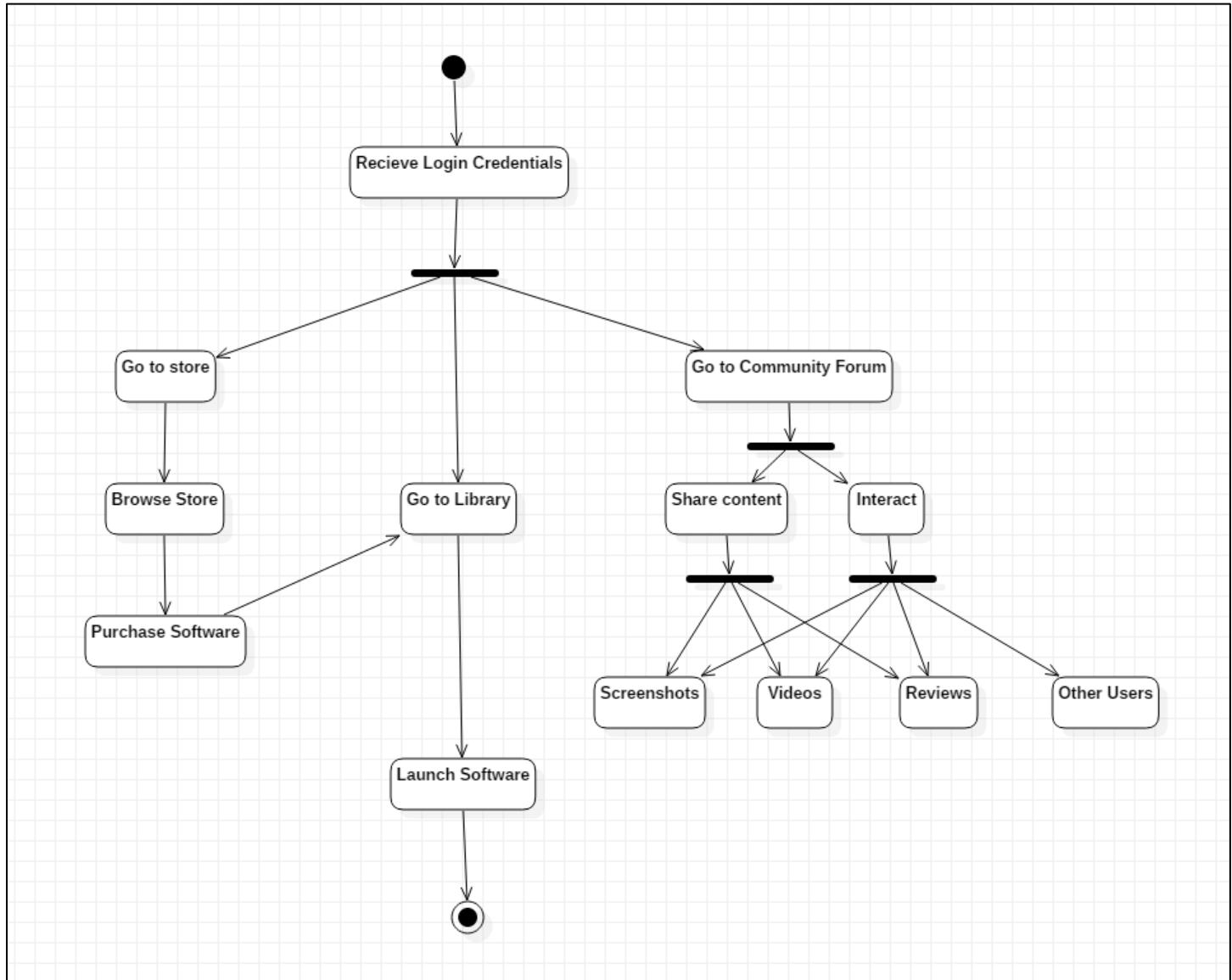
Result

The activity diagram was successfully prepared

# Experiment-8

## Activity Diagram

### Digital Software Distribution System- Origin



CBS 1

### Aim

To study functional & non functional requirements for online gaming platform

Software Used

Star UML

### Theory

Functional requirements define the basic system behavior. Essentially, they are what the system does or must not do, & can be thought of in terms of how the system responds to inputs. Functional requirements usually define if/then behaviors & include calculations, data input & business processes.

Functional requirements are product features that allow the system to function as it was intended. If the functional requirements are not met, the system does not work. Functional requirements are product features & focus on user requirements.

Non Functional requirements specify how the system should do it. They do not affect the basic functionality of the system. Even if non functional requirements are not met, the system will still

perform its basic purpose.

Non functional requirements define system behavior & general characteristics that affect user experience.  
These are product properties & focus on user expectations.

#### Result

The functional & non functional requirements were listed.

CBS 2

### Aim

To make data dictionary for online gaming platform

### Software Used

Star UML

### Theory

Data dictionary is a file or set of files that includes a database's metadata. The data dictionary holds records about other objects in the database such as data ownership, data relationships to other objects, & other data.

It is an essential component of any relational database. Typically, only database administrators interact with data dictionary.

It includes information about the following -

- Name of data item
- Aliases
- Description/Purpose
- Related data items
- Range of values
- Data structure definitions/forms

Aliases include other names by which this data item  
is called DEO for Data entry Operator & DR for  
Deputy Registrar

Description is a textual description of what the data item  
is used

Range of values records all possible values

Result

The data dictionary was created.

# CBS-1 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

## **Functional Requirements**

A user will be able to do following activity on the Online Gaming Platform:

1. Online Account check.
2. Online shopping opportunity.
3. Online data entry by the staff.
4. Updating the data.
5. Balance check.
6. Refund services if the product has not been used for a long period of time.
7. Community Forums for users to interact with each other and the publishers.

## **Non Functional requirements**

Following is a different type of non functional requirement needed to include in the Online Gaming Platform:-

- 1. Usability** – known as user friendly, the system should have the user guild of the system and cannot allow complex interface design
- 2. Response time** -System also needs to have a good response time. It needs to respond in the shortest time after the customer clicks a button.
- 3. Platform compatibility** -System needs to be able to perform in different platforms such as Linux, Window, and Mac OS.
- 4. Privacy** -It also should keep the customer information safe and do not review any information of the customer to other people without agreement of the particular customer.
- 5. Supportability** -system should be able to perform in any type of web browser. For example, Windows Explorer, Mozilla Firefox, Google and etc.
- 6. Maintainability** – System must be updated time by time. In this case, when a new movie is released, cinema is able to update the movie information to the system.
- 7. Safety** -The database may get crashed at any certain time due to virus or operating system failure. Therefore it is required to take the database backup.

**8. Reliability**-The capability to maintain the specified level of performance is called reliability. This application is a web based application that runs on any device that has a browser.

**9. Availability**- The application will be available for 99% of the time .

**10. Security**- The business logic is hidden from the users and is much safer and thus avoids unauthorised or illegal access or database corruption. Security of the user's information is also safe as there is a login facility.

**CBS-2**  
**DATA DICTIONARY FOR Online Gaming Platform**

**GAME TABLE**

FIELD NAME	DATA TYPE	SIZE	NULL	DESCRIPTION
Game_ID	Int	5	Not	Primary Key
Name	Char	20	Not	
Genre	Char	20	Not	
Platform	Char	20	Not	

**MARKETPLACE TABLE**

FIELD NAME	DATA TYPE	SIZE	NULL	DESCRIPTION
Transaction_ID	Int	5	Not	Primary Key
ItemPrice	Float	5	Not	
ItemType	char	20	Not	

**GAME DEV COMPANY TABLE**

FIELD NAME	DATA TYPE	SIZE	NULL	DESCRIPTION
Reg_No	Int	5	Not	Primary Key
Name	Char	20	Not	
Teams	Char	50	Not	

**USER TABLE**

FIELD NAME	DATA TYPE	SIZE	NULL	DESCRIPTION
User_id	Int	5	Not	Primary Key
Name	Char	20	Not	
Email	Char	50	Not	

**COMMUNITY TABLE**

FIELD NAME	DATA TYPE	SIZE	NULL	DESCRIPTION
Name	Int	5	Not	Primary Key
Friends	Char	20	Not	
Groups	Char	40	Not	

**TOURNAMENT TABLE**

FIELD NAME	DATA TYPE	SIZE	NULL	DESCRIPTION
Tournament_id	Int	5	Not	Primary Key
Name	Char	20	Not	
Organisers	Char	50	Not	