Table of Content

List of Figures

4.2.4		
4.2.5		

List of Tables

1. Introduction

1.1 Project Definition

These days games are big deal. People all around the world love playing them, but finding a good place to buy and manage these games isn't always easy. Many platforms are complicated, have limited game selections, or just don't provide a great experience for users. This makes it tough for gamer's to find find and enjoy the games they love. Plus some site are not very secure, which can make people worry about their personal info and payment details. gamer's also face problem with downloading and installing games due to technical issues. Many platform's don't have good ways for gamer's to connect with their friends and other players user have to use different platform to get all their favourite games. which is hassle. Lastly navigating these platforms can be frustrating due to poor user interface.

1.2 Project Objective

The objective of this project is to develop a comprehensive gaming platform, where Customers can purchase , download and manage variety of games. This platform aims to provide a secure user experience, offering features such as user account, game libraries, friends ,conversation with friends. We want to build something that can grow and get better over time, helping all kinds of gamer's. The ultimate goal is to create a scalable system that will fulfill the needs of gamer's.

1.3 Project Scope

We'll make sure the platform is easy to use, secure, and able to handle lots of users and games. The system will support different payment methods and follow decided standards to keep user data safe and secure transactions. This way, gamer's can enjoy their favorite games without any worry. we'll ensure the platform remains up-to-date and continues to meet the needs of the gaming community.

Chapter 2: Overall Description

2.1.1 Hardware interface :

- Operating system : Ubuntu, windows 7, 10.

- Hardware devices : Keyboard , mouse.

- Display : Standard output details.

2.1.2 Software interface :

- Language used : Django.

- Database : MYSQLI.

- User interface design : HTML, Java script, CSS.

- Web browser : Firefox, Google, Chrome.

Chapter 3: System Specific Requirements

3.1: Functional Requirement :-

1) Manage Customer.

2) Manage Developer.

3) Manage Games.

4) Manage Friends.

5) Manage Libraries.

6) Manage Conversation.

7) Manage Feedback.

8) Generate Report.

1. Manage Customer:

RN	Description	Comments
FR1	Customer shall be able manage their own account. ● View details. ● Update details. ● Add details. ● Delete details.	Customer profile page.
FR2	Admin shall be able to view customer details.	Customer's details page

2. Manage Developer :-

RN	Description	Comments
FR1	Admin shall be able to verify developer's details.	Developer's detail page.
FR2	Developer's shall be able to manage their own profile. ● Add details. ● Update details. ● Delete details. ● View details.	Developer's profile page.
FR3	Admin shall be able to view developer's details.	Developer's detail page.

3. Manage Games :-

RN	Description	Comments
FR1	Admin shall be able to approve games which will be uploaded by developer's	Pending games page.
FR2	Developer's shall be able to update games information	My games page.
FR3	Developers as well as admin shall be able to remove games.	My games page.
FR4	Customer shall be able to view games.	Games page
FR5	Customer shall be able to download the game if he/she buys the game.	Games page.

4. Manage Friends :-

RN	Description	Comments
FR1	Customer (Players) can add the other players into the friend list.	Add friends page.
FR2	Every players shall be able to view each others profile.	Friends profile page.
FR3	Players shall be able to see the status (Online/Offline) of their friends.	Status page.

5. Manage libraries :-

RN	Description	Comments
FR1	Players shall be able to view their downloaded games in their own libraries.	Library page.
FR2	Friends shall be able to view each others games list which they are playing.	Friends profile page.

6. Manage conversation :-

RN	Description	Comments
FR1	Players shall be able to chat with each other (Text chat).	Message page.

7. Manage feedback :-

RN	Description	Comments
FR1	Admin shall be able to view all the feedback which was given by players (Customer's).	Feedback page.
FR2	Customer shall be able to give feedback to games	Feedback page.
FR3	Developer shall be able to view feedback which is given by customer.	Feedback page.

8. Manage report :-

| RN	Description	Comments
FR1	Admin shall be able to view the reports of every game which is uploaded. ● How man players have downloaded the game . ● Feedback's. ● Total Purchase . ● Total Revenue.	Reports page.
FR2	Developers shall be able to view reports of the game which they have uploaded.	My games Report..

3.2: Non-functional Requirement

RN	DESCRIPTION	COMMENTS
NFR1	The system will be password protected. The authentication user can Only get access to the system.	Security
NFR2	The application will be user friendly and easy to operate, the function will be easily understandable.	Usability
NFR3	The system will be available for any point of time in a day.	Availability
NFR4	The system can help users to reach their goals in very few steps.	Efficiency
NFR5	The performance of a system will be high and can respond to users of the system under any certain workload.	Performance

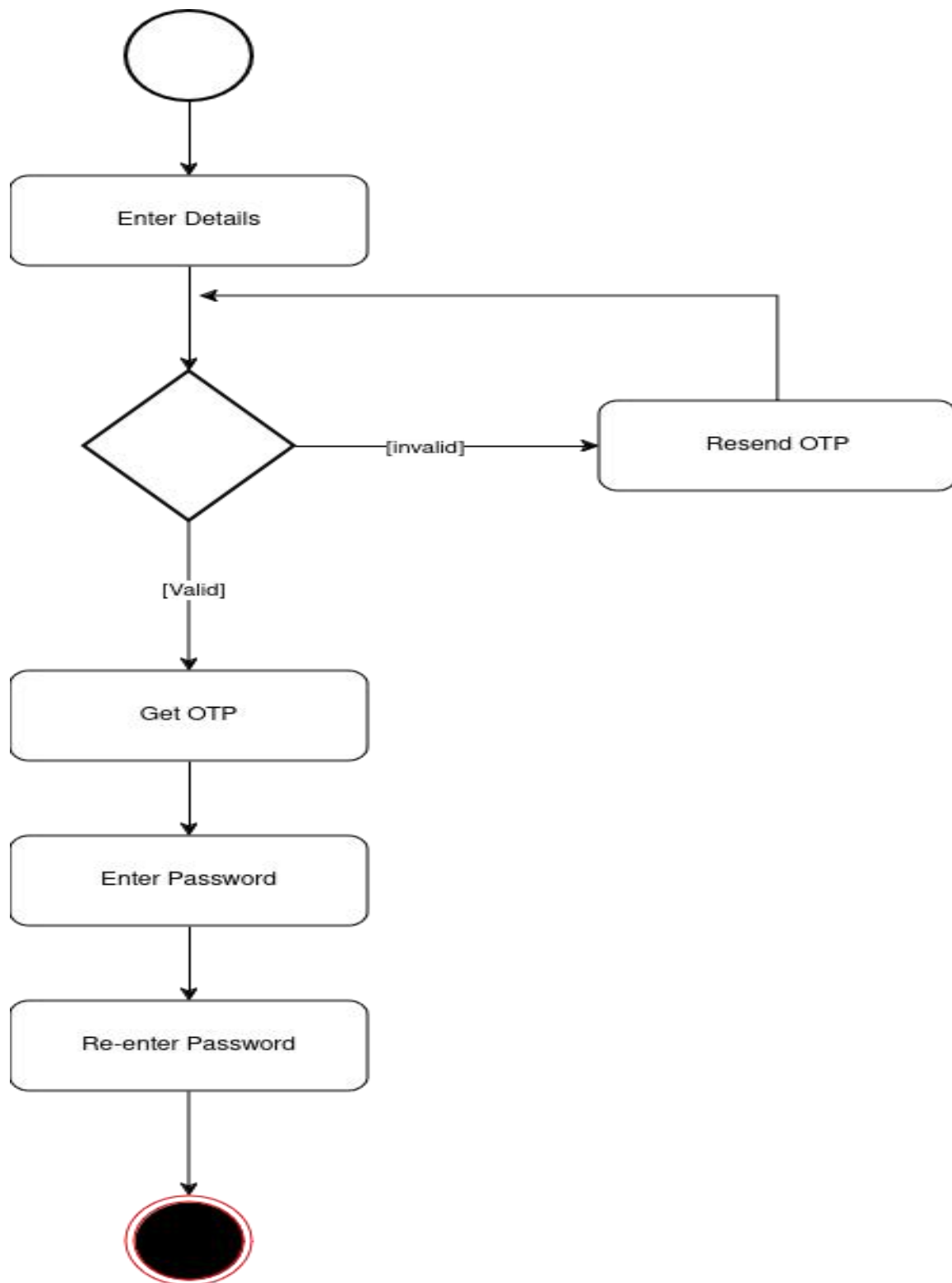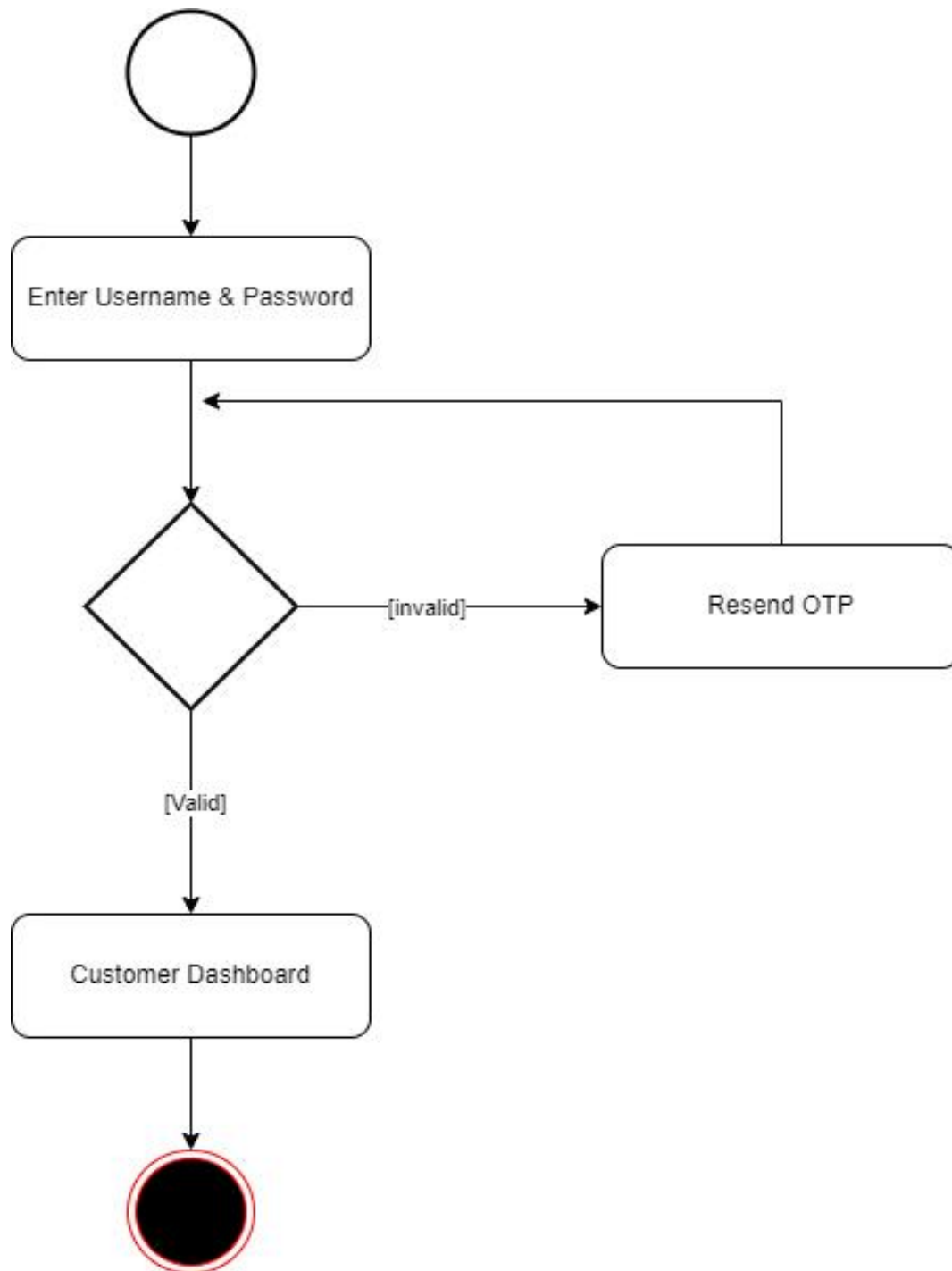# Chapter 4: UML Diagrams

4.1: Use case
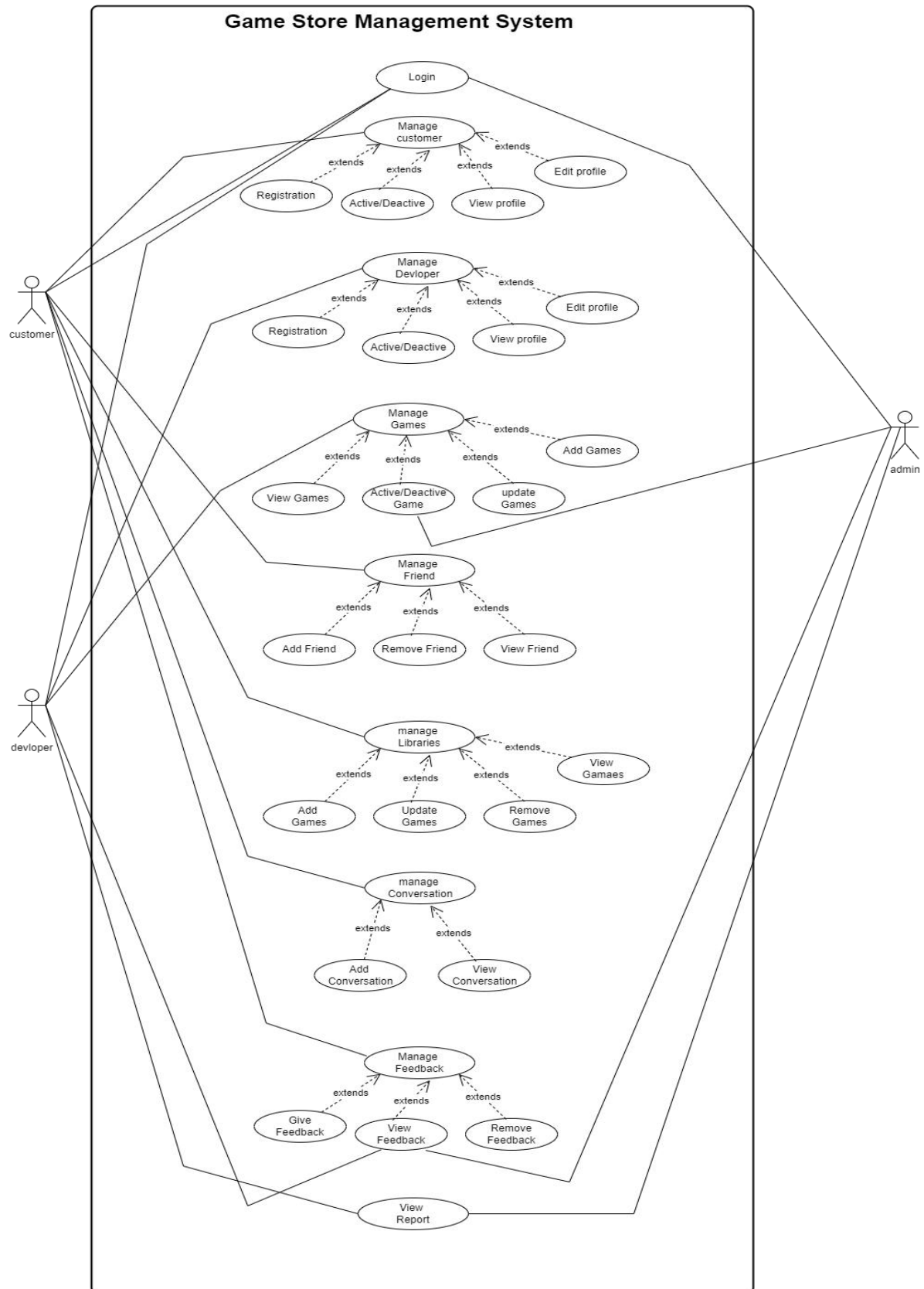
4.2: Activity Diagram

4.2.1 Activity Diagram For Registration :

4.2.2 Activity Diagram For Login :

4.2.3 Activity Diagram For Manage Games :

4.2.3 Activity Diagram For Forget Password :

Chapter 5: System Design

5.1: Database Design

       5.1.1: Database Schema

1.) Registration :

Tbl_players (Pid[PK], fname , lname , username , email, gender , contactno , DOB , password , ActiveStatus)

FD : Pid(fname , lname , username , email, gender , contactno , DOB , password , confirm password , ActiveStatus)

Tbl_developer (Did[PK], fname , lname , username , email, gender , contactno , DOB , password)

FD : Did(fname , lname , username , email, gender , contactno , DOB , password , ActiveStatus)

2.) Login :.

Tbl_players (Pid[PK], fname , lname , username , email, gender , contactno , DOB , password , ActiveStatus)

FD : Pid(fname , lname , username , email, gender , contactno , DOB , password , confirm password , ActiveStatus)

Tbl_developer (Did[PK], fname , lname , username , email, gender , contactno , DOB , password)

FD : Did(fname , lname , username , email, gender , contactno , DOB , password , ActiveStatus)

Tbl_Admin (Aid[PK], email , password)

FD : Aid (email , password)

3.) Manage Games :

Tbl_games (Gid[PK] , title , Did[FK] , release_date , category , price , version)

FD : Gid (title , Did[FK] , release_date , price , version)

Tbl_Game_version (Vid[PK] , Gid[FK] , version_no ,title , release_date , file_path)

FD : Vid (Gid[FK] , version_no ,title , release_date , file_path)

4.) Manage Friends :

Tbl_friends (Fid[PK], Pid1[FK] , Pid2[FK] , Status)

FD : Fid (Pid1[FK] , Pid2[FK] , status)

5.) Manage Libraries :

Tbl_Libraries (library_id[FK] , Pid[FK] , Gid[FK], install_date)

FD : library_id[FK] (Pid[FK] , Gid[FK], install_date)

6.) Manage Conversation :

Tbl_conversation (conversation_id[PK] , Pid1[FK] , Pid2[FK] , message , time)

FD : conversation_id (Pid1[FK] , Pid2[FK] , message , time)

7.) Manage Feedback :

Tbl_feedaback (Feedback_id[PK] , Pid[PK] , Gid[FK] , rating , feedback_date , responses)

FD : Feedback_id[PK] , Pid[PK] , Gid[FK] , rating , feedback_date , responses)

8.)Manage Report :

Tbl_Report (report_id[PK] , report_type , total_downloaded, , revenue , ratings)

FD : report_id (report_type , total_downloaded, , revenue , ratings)

5.1.2: Data Dictionary

TABLE NAME 1: TBL_ CUSTOMER (Player)

TABLE DESCRIPTION: STORE DETAILS OF CUSTOMER (Player)

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1.	Pid	Int	-	Primary Key, Auto increment	Player ID
2.	Fname	Varchar	30	-	First Name
3.	Lname	Varchar	30	-	Last name
4.	Username	Varchar	30	-	Username of player
5.	Email	Varchar	14	-	Email of player
6.	Gender	Varchar	10	-	Gender of player
7.	Contactno	Varchar	10	-	Contact number of player
8.	DOB	Date	-	-	Date of birth of player
9.	Password	Varchar	40	-	Password of player

TABLE NAME 2: TBL_ Developer

TABLE DESCRIPTION: STORE DETAILS OF developer

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1.	Did	Int	-	Primary Key, Auto increment	Developer ID
2.	Fname	Varchar	30	-	First Name
3.	Lname	Varchar	30	-	Last name
4.	Username	Varchar	30	-	Username of player
5.	Email	Varchar	14	-	Email of player
10.	Gender	Varchar	10	-	Gender of player
11.	Contactno	Varchar	10	-	Contact number of player
12.	DOB	Date	-	-	Date of birth of player
13.	Password	Varchar	40	-	Password of player

TABLE NAME 3: TBL_ Admin

TABLE DESCRIPTION: STORE DETAILS OF admin

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1.	Aid	Int	-	Primary Key, Auto increment	Admin ID
2.	Email	Varchar	10	-	Email of admin
3.	password	Varchar	30	-	Password of admin

TABLE NAME 4: TBL_ Games

TABLE DESCRIPTION: STORE DETAILS OF GAMES

Sr. No.	Field Name	Data Type	Size	Constraints	Description
1.	Gid	Int	-	Primary Key, Auto increment	Game id
2.	Title	Varchar	30	-	Name of game
3.	Did	Int	-	Foreign key	Developer id who has uploaded the game
4.	Release_date	Date	-	-	Date on which game has released
5.	Category	Varchar	20	-	Category of game

6.	Price	Int	-	-	Price of game
7.					

Tbl_games (Gid[PK] , title , Did[FK] , release_date , category , price , version)

FD : Gid (title , Did[FK] , release_date , price , version)

Tbl_Game_version (Vid[PK] , Gid[FK] , version_no ,title , release_date , file_path)

FD : Vid (Gid[FK] , version_no ,title , release_date , file_path)

Chapter 6: System Implementation

6.1: Screenshots

Chapter 7: Testing

7.1: Test Cases

Chapter 8: Future Enhancement

Conclusion