

# Abstract

In the current society, mobile games are often used by numerous individuals. It is a well-known fact that in-app purchases are required for many gamers, in order to improve the gaming success and gain competitive advantage among the other players. In the perspective of the developers, in-app purchasing done by the gaming individuals is the main revenue stream, but in the developing countries including the Sri Lankan context, the players do not tend to purchase these, since there is no financial benefit other than succeeding in the game. The other reason for the reluctance of using In-app Purchases is that the credit card details should be provided to purchase them. This is an issue faced by the players by not being able to utilize the total advantages provided by the game. If these purchasing by the players is increased, Mobile Game Developers will be able to gain the opportunity to expand their market and increase their revenues.

The solution is to use Virtual Coins instead of Actual Coins, which includes three parties to use the system developed by the organization i.e. the players, the game developers and retail distributors. The process involves a few steps. The retail distributor could assign a discount to the relatively low demanded products, in order to make the sales high. Instead of giving them the monetary discounts, the owner could purchase virtual coins from the portal, and assign those coins instead of the discount, which will be received by the customers who are registered in system. This will increase the sales of the grocery item rather than just giving a discount. The customers could use those coins he/she earned by buying a day-to-day item, to purchase the Gaming Objects (Usually provided by In-app Purchases). This will give the customer to use the money he spent to buy a good/service to purchase gaming benefits, without exposing the credit card. The game developers who use portal API to develop games will be gaining a revenue via the purchases done by the players.

The system was developed using the programming languages PHP, HyperText Markup Language (HTML), Cascading Style Sheets (CSS), Javascript, C# and technologies such as XAMPP, MySQL, Visual Studio and Unity 3D. This system would undoubtedly be a unique solution facilitated to three arenas of business.

# Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: Didula Newunhella

Registration Number: 16550507

Index Number: 2016/MIT/050

---

Signature

Date

This is to certify that this thesis is based on the work of

Mr. Didula Newunhella

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: Dr. L. N. C. De Silva

---

Signature

Date

## **Acknowledgements**

It is my pleasure to thank many individuals who contributed to the success of this project. First and foremost, I thank my project supervisor Dr. L.N.C. De Silva at University of Colombo School of Computing for her patient guidance, enthusiastic encouragement and useful critiques of this project work.

Finally, I must express my very profound gratitude to my parents and to my sister for providing me with unfailing support and continuous encouragement through the process of completing this project. This accomplishment would not have been possible without them.

# Table of Contents

Abstract	i
Declaration	ii
Acknowledgements	iii
Table of Contents	iv
Table of Figures	vii
List of Tables	viii
CHAPTER 1	1
1. Introduction	1
1.1. Identification of the Problem	2
1.2. Motivation	4
1.2.1. Weaknesses of the Current System	4
1.3. Goals / Objectives	4
1.4. Scope	5
1.5. Outline	6
CHAPTER 2	7
2. Background	7
2.1. Requirement and System Analysis	7
2.1.1. Existing System	7
2.1.2. Overall Use Case Diagram of the Current System	8
2.1.3. Functional Requirements	9
2.1.4. Non Functional Requirements	10
2.2. Review of Similar Systems	11
2.3. Comparison of Alternative Design Strategies	11
2.3.1. Selected BSO and Justification	13

2.4. Summary	13
CHAPTER 3	14
3. Methodology	14
3.1. System Design	14
3.1.1. Use Case Diagram	15
3.1.2. Activity Diagrams	20
3.1.3. Sequence Diagrams	23
3.1.4. Class Diagram	26
3.1.5. Database Design	27
3.2. System Development	29
3.2.1. Programming Languages and Development Tools	29
3.2.2. Third Party Components and Libraries	30
3.2.3. Hashing of Confidential Data	30
3.3. Installation Guide	30
3.3.1. Hardware Requirements	30
3.3.2. Software Requirements	31
3.3.3. User Manual	31
3.4. Summary	44
CHAPTER 4	45
4. Evaluation	45
4.1. Test Plan and Test strategy	45
4.1.1. Justification of the Selection of Testing Approach	46
4.2. Test Cases	47
4.2.1. List of Test Cases required for the System	47
4.3. Test Report	49

4.4. Degree of Achieved Objectives	50
4.4.1. Objectives Analysis	51
4.5. Usability, accessibility, reliability and friendliness	52
4.6. Summary	52
CHAPTER 5	53
5. Conclusion	53
5.1. Limitations and Drawbacks	56
5.2. Future Work	56
5.3. Summary	56
References	57

## Table of Figures

Figure 1 - Usage of Gaming .....	2
Figure 2 - Interfaces of the Existing System .....	7
Figure 3 - Current System - Use Case Diagram .....	8
Figure 4 - Overall use case diagram for the proposed system.....	15
Figure 5 - Activity Diagram for Earning Coins.....	20
Figure 6 - Activity Diagram for Refill Coins .....	21
Figure 7 - Activity Diagram for Spend Coins .....	22
Figure 8 - Sequence Diagram for Earning Coins .....	23
Figure 9 - Sequence Diagram for refill coins .....	24
Figure 10 - Sequence Diagram for Spend Coins .....	25
Figure 11 - Class Diagram for the Proposed System .....	26
Figure 12 - Entity Relationship Diagram .....	27
Figure 13 - Normalized Database Design.....	28

## List of Tables

Table 1 - Use Case Description for Earn Save Me Coins by Buying Items.....	16
Table 2 - Use Case Description for Assign Coins to Product .....	16
Table 3 - Use Case Description for Buy Coins from Save Me .....	17
Table 4 - Use Case Description for Purchase with Save Me Coins .....	17
Table 5 - Use Case Description for Checkout .....	18
Table 6 - Use Case Description for Select Items.....	18
Table 7 - Use Case Description for Registration.....	19
Table 8 - Use Case Description for Get Payment.....	19
Table 9 - Test case for User login and registration for web portal.....	47
Table 10 - Test case for purchasing game objects using 'Save Me'.....	48
Table 11 - Test case for offer coins using desktop application .....	48
Table 12 - Cross Check Table with objectives .....	51



# CHAPTER 1

## 1. Introduction

This project “A smart mobile in-app purchasing solution for gaming community” named “Save Me” presents the importance of moving with new trends in the technological arena to succeed in business, through a smart mobile in-app purchasing solution for gaming community. The increased use of mobile applications in the current society has paved new ways of emerging requirements and opportunities.

The cellular technology is playing a significant role in simplifying the daily tasks, changing the outlook towards information. It is undoubtedly evident that apps have become an integral part of the digital ecosystem. In fact, these apps are progressing to make ubiquitous presence. However, staying up-to-date with the latest trends of mobile app development has become a must rather than merely an option.

These opportunities of the increased use of mobile applications are utilized in the project, where the solution includes three stakeholders. i.e. gamers, developers and distributors of goods and services (of any retail industry – no particular client since this system deals with three types of users). Mobile games are used to attract the customers, which will increase sales of retailers and the income of mobile game developers. The requirements of the mobile application users are also being addressed in the system, where a solution to the reluctance of purchasing in-app purchases is provided.

This product is a unique solution as well as an opportunity in the evolving market, which addresses the requirements of diverse target markets.

## 1.1. Identification of the Problem

The mobile gaming industry is gaining attraction from the society daily, where it has already become a desire. With the vast usage of smart phones, the mobile applications being developed as games has become more popular (Figure 1). It is vital that every mobile game developer become competitive amidst this trend. In addition, the players too strive hard to gain competitive advantage in playing the games.

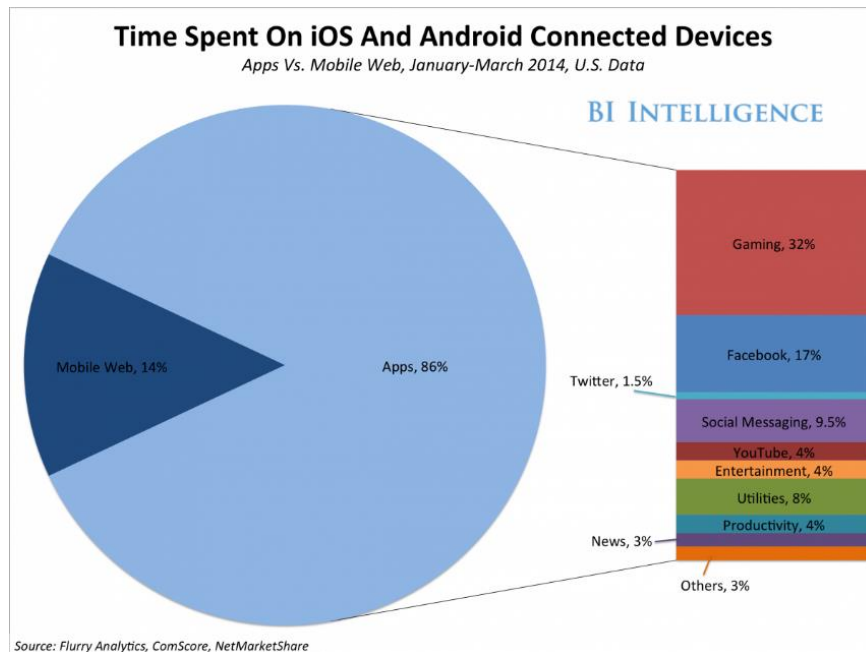


Figure 1 - Usage of Gaming

Mobile game players are addicted to play free games, but yet in developing countries including Sri Lanka, the players are reluctant to do in-app purchases, which are to be done during the game. An in-app purchase is the purchase of goods and services from an application on a mobile device, such as a smartphone or tablet. In-app purchases allow developers to provide their application for free, while providing anyone who downloads the free version the opportunity to upgrade. Since the developer can always add new features and content to the application after it is downloaded, application users will not have to download a different application later.

The major revenue stream of the mobile game developers remains to be the in-app purchases done by the mobile game players. Mostly these mobile games are freely accessible by each player and the only instance where the player has to pay is the In-app Purchasing.

The problem is explained below, which combines two arenas, which are mobile game industry and sales and marketing (Distribution).

Mobile Game Industry – In the current context of Mobile Game Industry, mobile game developers' main revenue is the In-App Purchases of their mobile games. But these In-App Purchases are not much popular among mobile gamers, especially in developing countries. This is because the players have to reveal their credit card details (which are confidential) and spend only for buying game objects, which is not worthwhile. Some individuals do not possess the credit card facility, too. Thus the revenue of the game developer decreases, where a potential market segment is lost.

There is a massive opportunity to implement an alternative method to do In-App Purchases without using credit cards. In the meantime, the mobile game developer would get the revenue as well, since they would be able to tap into a new market segment, in a different manner.

In addition, the players of the games too gain a competitive advantage compared with the other players, since they get to use the maximum benefits/resources provided by the game. Most of the players do not like to spend only for buying some virtual object unless they have an actual benefit from it other than getting the competitive advantage in mobile game. They expect something valuable for their day-to-day life as well.

Distribution – The objective of most of the distributors in any retail industry is to increase their sales. To achieve that objective, they need a competitive advantage among their competitors.

## **1.2. Motivation**

In the current context, distributors reward discounts and offers to attract their customers. But most of the time those offers do not make the customer satisfied. Many of the offered items are not useful for the customers. Especially in the modern society where more people have connected with the mobile, they expect that something will support them with the technology.

### **1.2.1. Weaknesses of the Current System**

Players do not like to pay only for virtual objects, where they have to provide their confidential credit card details for payments because of security issues. There is no alternative method to do virtual purchases.

On the other hand, there is no way to use the rewards or discounts as per the willingness of the customers. Furthermore, the mobile game developers have no way of increasing their revenue in terms of in-app purchases.

## **1.3. Goals / Objectives**

The project objective is to overcome the below mentioned issues and provide a solution taking the advantage of the opportunity.

When considering the mobile game industry, it has the process of In App Purchases. Developer develops the game with In App purchase option. Product releases to the market with a price of premium, subscription or any other method. Game contains some virtual objects like powers, equipment, etc which allow the players to excel in the games. Some of these objects are offered freely to continue gaming. But at some point, player has to buy those objects to play the game smoothly. These objects may be some equipment or an add-on to the game. To purchase those objects the player should give their credit card number to the game stores. After that the players

can purchase the object by clicking on the button. Then the amount credits to the developers account automatically. This process is an automated process. The weaknesses of the current system are that the players do not like to pay only for virtual objects.

In terms of sales and marketing, distributors use reward system to increase their sales and profit. In some cases, they offer some other item freely or else they deduct the price of the product, as discounts. This system is a completely manual process.

Thus, the objectives of the proposed project can be summarized as follows.

1. To develop a system, where the payment of the players is avoided or minimized, which increases the usage of games by the users. This will make the players more motivated, since they get to use the benefits/resources of the games without a cost.
2. To develop a platform which increases the revenue stream of the game developers, by increasing the number of games played by users.
3. To attract more customers to the retail chain (Distributors/Supermarkets), by providing discounts in a novel method, which provides benefits to the customers.

## **1.4. Scope**

The system covers a few mobile games developed which has no particular age limit and could be played in different levels and interests. It possesses the management of players who earn points (by allocating virtual coins via the system, for each win of the game), allowing the points to act as purchasing coins. It also has purchases management (maintaining the player's account balance in terms of earned and spent coins) and strategic report generation (Example – Summary of earned coins, total savings through coins in terms of discounts, monthly summary reports for the developers and the game users).

But the system does not cover the user payment aspect, which means that the customer's reduction of actual monetary bank balance will not be looked into. This is because the systems like PayPal will not be repeated, since they are already available. The virtual coins allocation through earnings and the reduction will be handled.

## **1.5. Outline**

In Chapter 1, the nature of the current business process was identified. Then the problems and issues were defined, through which the objectives were emphasized. Furthermore, clear boundaries and scope for the system were defined.

Chapter 2 explains the background of the study, which analyses the existing system and the requirements.

Chapter 3 elaborates on the methodology used in the study along with the system design and system development, which is explained through a user manual.

Chapter 4 evaluates the system deeply, using testing strategies and achieved objectives of the project.

Finally, Chapter 5 concludes the report with the limitations, future work and suggestions.

# CHAPTER 2

## 2. Background

### 2.1. Requirement and System Analysis

There has been an exponential growth in the proliferation of mobile apps over the last decade. It has become imperative that companies use these mobile apps to connect to existing and potential customers to sustain or build new revenue streams. The popularity of gaming has been growing and with connectivity improving in the developing nations, the potential for growth in these countries with a large tech savvy youth population is huge.

#### 2.1.1. Existing System

Currently there are two systems for mobile game industry and sales and marketing. When considering about the mobile game industry, it has the process of in-app purchases, where the developer produces the game with the aforementioned process. Then the product is released to the market at a premium, subscription or any other arena. The developed game contains virtual objects, some of which are offered freely to continue gaming. But at some point, player has to buy those objects to play the game smoothly, which is an add-on to the game. In order to purchase such objects the player should give their credit card number to the game stores. Afterwards the player can purchase the object. Then the amount credits to the developers account automatically. This process is an automated process.

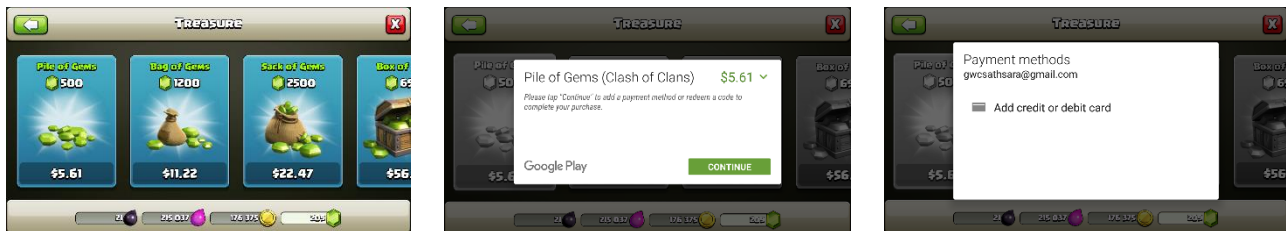


Figure 2 - Interfaces of the Existing System

In the sales and marketing industry, the distributors use reward system to increase their sales and profit. In some cases, they offer one other item free or else they deduct the price of the product.

### 2.1.2. Overall Use Case Diagram of the Current System

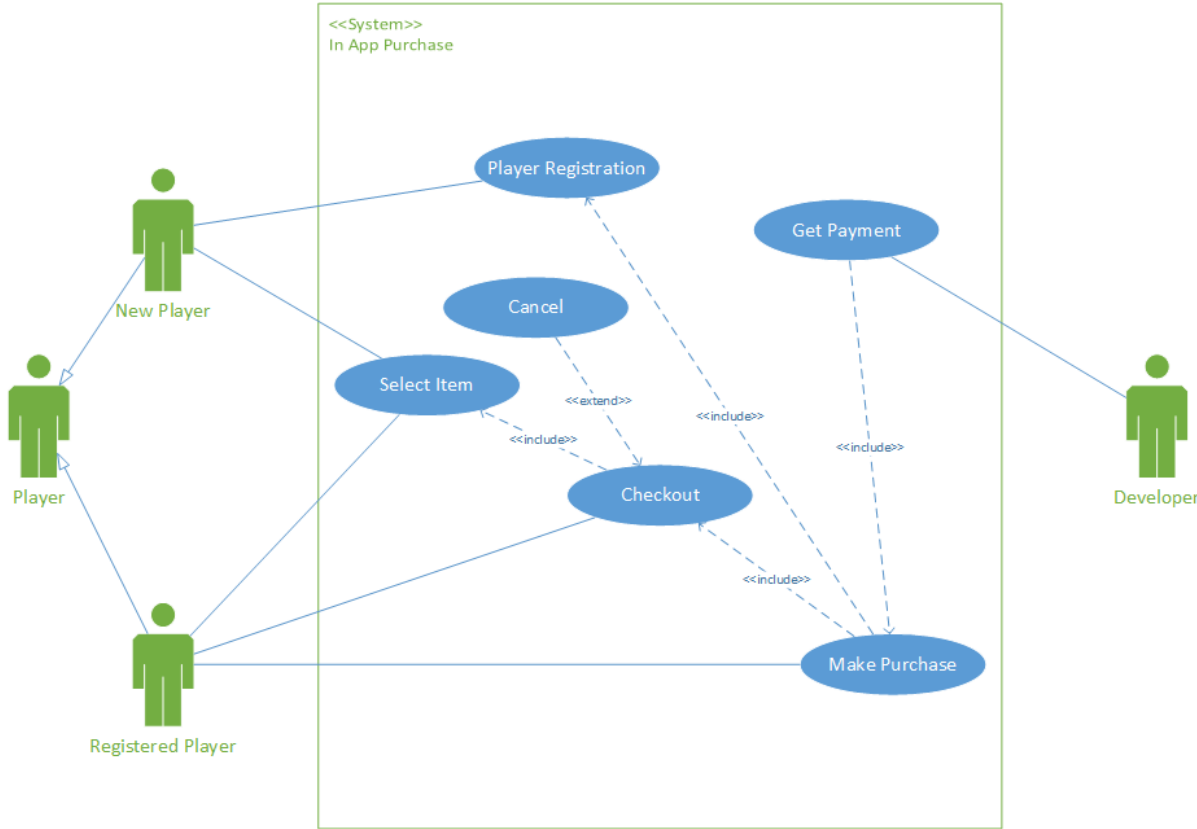


Figure 3 - Current System - Use Case Diagram

The above use case diagram explains the overall system of In-App-Purchasing. According to the diagram, every game player can select the game object they want, but only registered customers can check out and make purchases. For a new player, it is essential to register to the system. In order to purchase items, players should provide their payment details. Game developers earn money according to the number of purchases happened in their mobile games.



### 2.1.3. Functional Requirements

Functional requirements describe what the system should be able to do in order to satisfy the main objective of it. Mainly these describe what the tasks are or functions which the system can perform after implementation. Following table will explain the functional requirements of this project.

<b>Requirement</b>
Shall be able to download games
Shall store player details
Shall be able to purchase game objects
Shall be able to track the purchases of the player
Shall be able to get discounts for purchases
Developers shall be able to add new games
Developers shall be able to earn money
Shall be able to categorize the games
Shall be able to review games
Shall be able to get comments from players
Shall be able to generate reports
Shall be able to access to the purchasing screen easily
Should be able to suggest games for players
Should be able to display distributors' offers
Should utilize the available hardware and software
Should be able to display assets available
Distributors should be able to promote their products

## 2.1.4. Non Functional Requirements

Non-functional requirements describe the usability, reliability, performance, maintainability and other similar aspects of the system. These set of requirements may not be directly related to the main functionality but they are of extreme importance to the proper functioning of the system.

<b>Requirement</b>
Shall be a user friendly interface
Shall provide precise information
Shall have quicker response time
Shall facilitate concurrent access
Should provide a web interface
Should be cost effective
Should use the existing resources efficiently
Should ensure security

## **2.2. Review of Similar Systems**

The following are the similar systems currently available as individual solution, unlike the proposed system. The proposed system is an innovation with an integrated system connecting all the systems below.

- 1. In app purchasing in Google Pay Store and Apple Store [2]**
- 2. Loyalty point systems in supermarkets [3]**

## **2.3. Comparison of Alternative Design Strategies**

There are several alternatives, which are other approaches to overcome the identified problems. The following are those Business System Options (BSOs).

BSO1 - Use Google's Play store and its In-App-Purchasing system to distribute games.

BSO2 - Publish games as third party applications and let the user to install game directly from the publisher's website.

BSO3 - Publish the game in Google's Play store and allow users to do In-App-Purchases using a coins system.

### BSO Analysis of Functional Requirements

Description	BSO 1	BSO 2	BSO 3
Shall be able to download games	X	X	X
Shall store player details	X	X	X
Shall be able to purchase game objects	X	X	X
Shall be able to track the purchases of the player	X	X	X
Shall be able to get discounts for virtual purchases	X	X	X
Developers shall be able to add new games	X	X	X
Developers shall be able to earn money	X	X	X
Shall be able to categorize the games	X	X	X
Shall be able to review games	X	X	X
Shall be able to get comments from players	X	X	X
Shall be able to generate reports	X	X	X
Shall be able to access to the purchasing screen easily	X	X	X
Should be able to suggest games for players		X	X
Should be able to display distributors' offers		X	X
Should utilize the available hardware and software	X		X
Should be able to display assets available	X		X
Distributors should be able to promote their products		X	X

### BSO Analysis of Non-Functional Requirements

Description	BSO 1	BSO 2	BSO 3
Shall be a user friendly interface	X	X	X
Shall provide precise information	X	X	X
Shall have quicker response time	X	X	X
Shall facilitate concurrent access	X	X	X
Should provide a web interface			X
Should be cost effective	X		X
Should use the existing resources efficiently			X
Should ensure security			X

### **2.3.1. Selected BSO and Justification**

BSO 1 is a low cost option. Developers get the complete benefit from their games. But when considering about the security, it is very difficult to ensure a protection as Google does. The main issue in this BSO is the lack of users and small collection of games. Because of these issues, it will not make a good profit for the developer. Users usually do not like to install games from untrusted parties as well. Thus, this BSO will not satisfy the requirements properly.

BSO 2 is a trustworthy option. It has a huge marketing segment. But in this system, players have to provide their credit card details and have to pay actual money to purchase game objects which players do not like to do. It reduces developers' profits and players' experience. Even if this is a good option, these issues limit requirements.

BSO 3 has all the benefits of BSO 2 and players do not have to give credit card details or pay actual money. Since this option satisfies all the requirements and give benefits to players, developers and distributors, the selected BSO will be BSO 3.

## **2.4. Summary**

This chapter described the current scenario of the problem situation. It defined the current process through the use of case diagrams. Additionally, it contains a discussion on possible business system options followed by a requirement specification. Finally, the possible BSO's are evaluated against the requirement catalogue and justification of choosing the best BSO through the proposed system is provided.

# CHAPTER 3

## 3. Methodology

### 3.1. System Design

This includes the proposed system's design. It describes the proposed system and its functionalities in a more detailed manner, by utilizing use case diagrams and activity diagrams. The database design is also included along with the methodology used in the proposed system

#### **Outline -**

- Use Cases
- Activity Diagrams
- Sequence Diagrams
- Class Diagrams
- Database Design
- Interface Design

Here onwards, the product “A smart mobile in-app purchasing solution for gaming community” will be referred as “Save Me” for convenience, in the diagrams.



Table 1 - Use Case Description for Earn Save Me Coins by Buying Items

<b>Use Case</b>	Earn ‘Save me’ coins by buying items
<b>Use Case Id</b>	SM_01
<b>Actors</b>	Registered player
<b>Description</b>	Registered player buys ‘Save me’ eligible product. Then coin amount assigned for that product will be added to the player’s account.
<b>Pre-Conditions</b>	Player register to the system; Coins should have been assigned to the product
<b>Post-Conditions</b>	Added coin amount will be saved into player’s account.
<b>Exceptions</b>	Product is not eligible for offer.

The author has described one use case diagram (Table 1) for the convenience of the reader.

Description – The customer/ Player buys products from the distributor which are having assigned points (coins). The exception could happen when the customer buys a product which does not belong to the offer. When the customer buys the product, the amount of coins allocated will be added to the account, which he/she can use when receiving in-app purchases.

Table 2 - Use Case Description for Assign Coins to Product

<b>Use Case</b>	Assign coins to product
<b>Use Case Id</b>	SM_02
<b>Actors</b>	Distributor
<b>Description</b>	Assign a ‘Save me’ coin value to the product before distributing.
<b>Pre-Conditions</b>	Distributors should have a coin balance in their account.
<b>Post-Conditions</b>	
<b>Exceptions</b>	



Table 3 - Use Case Description for Buy Coins from Save Me

<b>Use Case</b>	Buy coins from 'Save Me'
<b>Use Case Id</b>	SM_03
<b>Actors</b>	Distributor
<b>Description</b>	Buy a coin package from 'Save me' portal.
<b>Pre-Conditions</b>	Distributor should be logged in to the system.
<b>Post-Conditions</b>	
<b>Exceptions</b>	

Table 4 - Use Case Description for Purchase with Save Me Coins

<b>Use Case</b>	Purchase with 'Save me' coins
<b>Use Case Id</b>	SM_04
<b>Actors</b>	Registered player
<b>Description</b>	Make the payment to the game object selected using the user's 'Save me' coins.
<b>Pre-Conditions</b>	There should be a selected item and the checkout should be completed.
<b>Post-Conditions</b>	
<b>Exceptions</b>	

Table 5 - Use Case Description for Checkout

<b>Use Case</b>	Checkout
<b>Use Case Id</b>	SM_05
<b>Actors</b>	Registered player
<b>Description</b>	Check the coin balance of the user and compare the selected object value with the available balance.
<b>Pre-Conditions</b>	Should select the game object.
<b>Post-Conditions</b>	Proceed to purchase.
<b>Exceptions</b>	Cancel checkout

Table 6 - Use Case Description for Select Items

<b>Use Case</b>	Select items
<b>Use Case Id</b>	SM_06
<b>Actors</b>	Registered player, New player
<b>Description</b>	Select the object need to be purchased.
<b>Pre-Conditions</b>	
<b>Post-Conditions</b>	
<b>Exceptions</b>	Proceed to checkout

Table 7 - Use Case Description for Registration

<b>Use Case</b>	Registration
<b>Use Case Id</b>	SM_07
<b>Actors</b>	New player, Distributor, Developer
<b>Description</b>	Create an account and register to the system.
<b>Pre-Conditions</b>	
<b>Post-Conditions</b>	
<b>Exceptions</b>	User already registered.

Table 8 - Use Case Description for Get Payment

<b>Use Case</b>	Get Payment
<b>Use Case Id</b>	SM_08
<b>Actors</b>	Developer
<b>Description</b>	Developer gets payments for the purchases done through his games. This section is handled by an already developed method in Google Play services.
<b>Pre-Conditions</b>	Developer should have an account in Google play developer console.
<b>Post-Conditions</b>	
<b>Exceptions</b>	

### 3.1.2. Activity Diagrams

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity.

#### Activity Diagram for earning coins

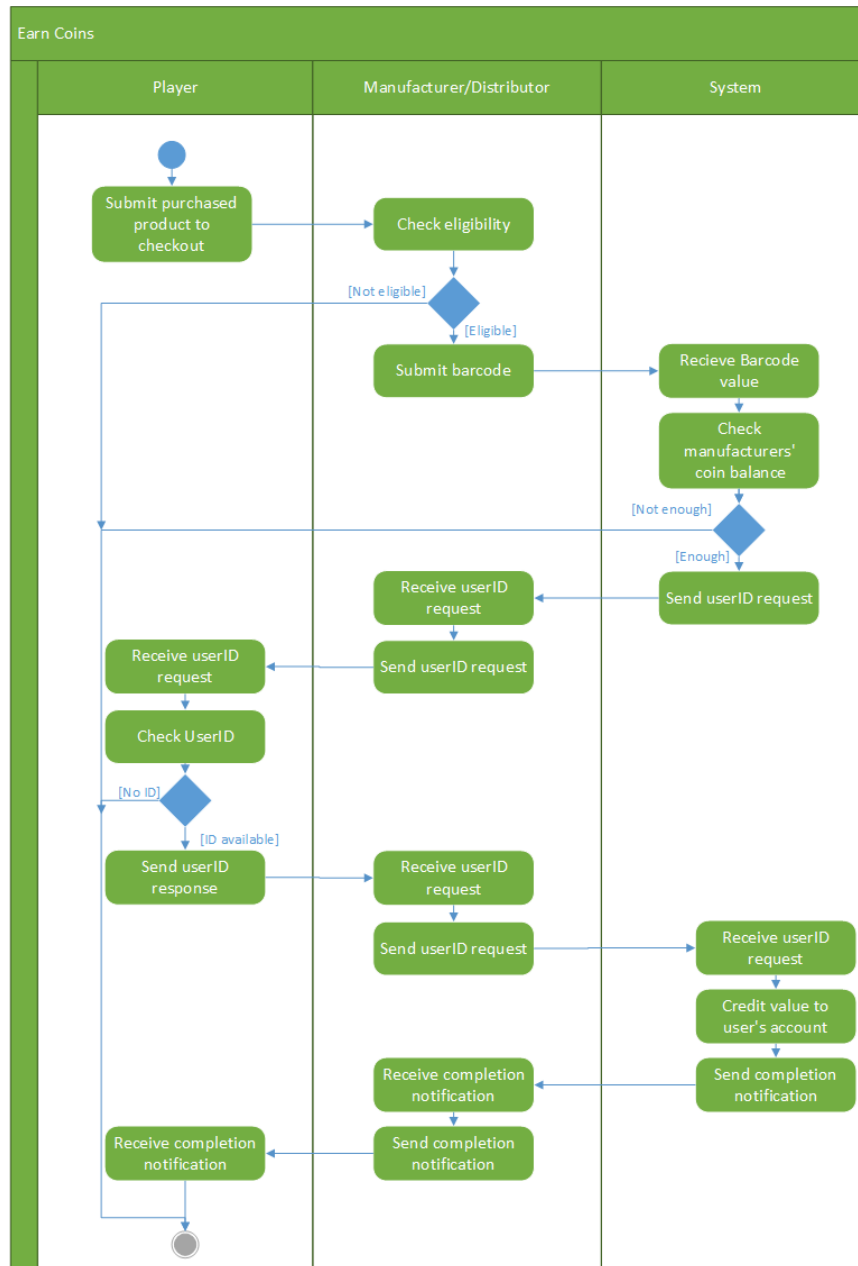


Figure 5 - Activity Diagram for Earning Coins

## Activity Diagram for refill coins

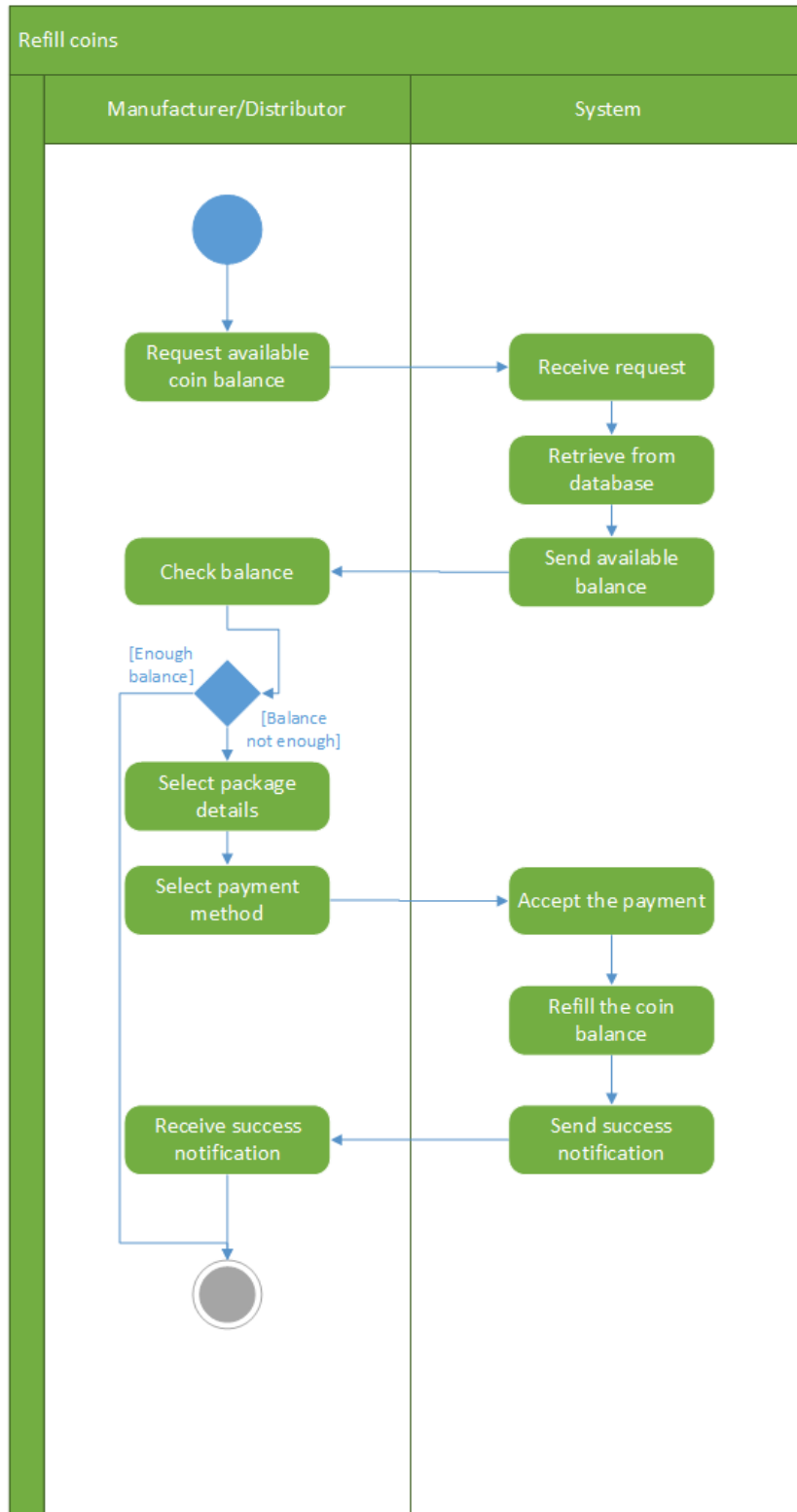


Figure 6 - Activity Diagram for Refill Coins

## Activity Diagram for spend coins

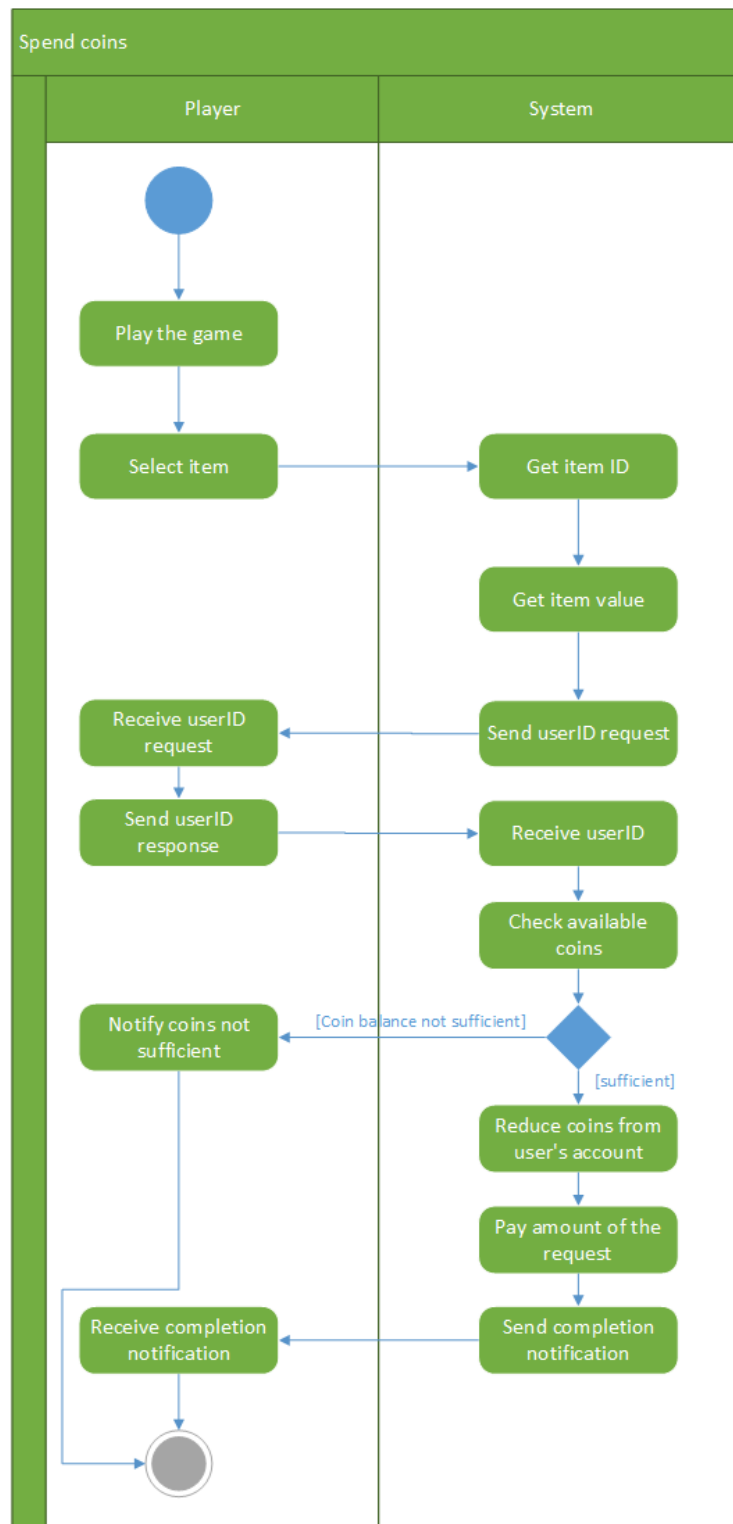


Figure 7 - Activity Diagram for Spend Coins

### 3.1.3. Sequence Diagrams

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

#### Sequence Diagram for earning coins

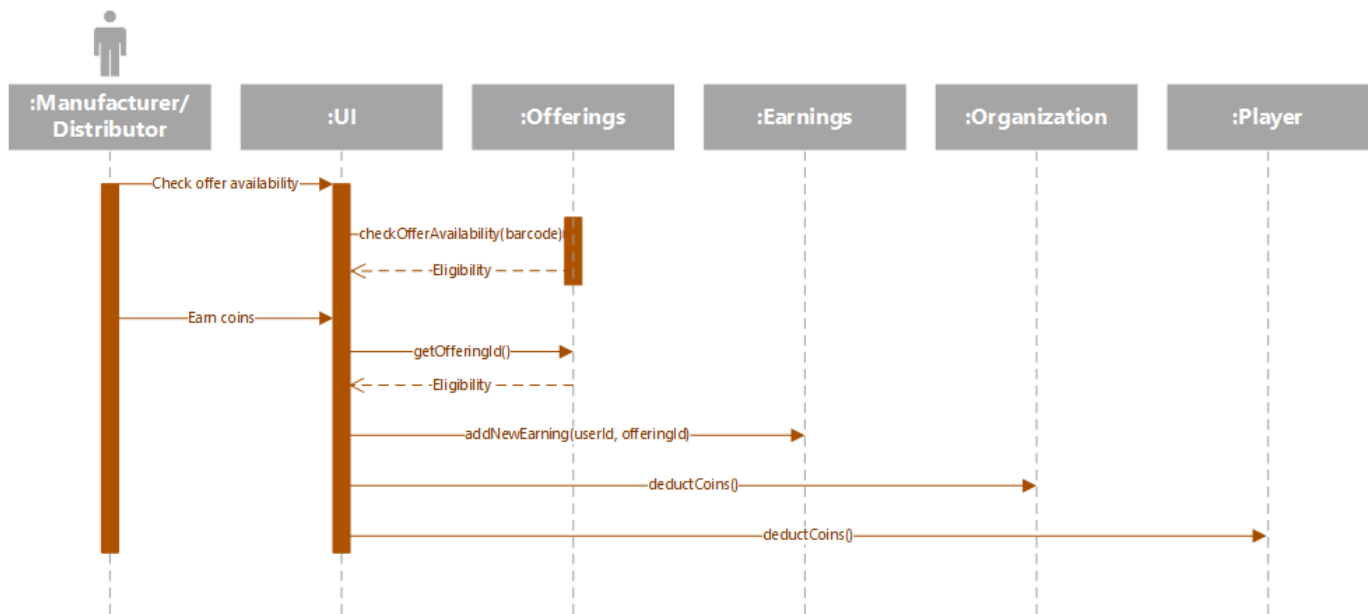


Figure 8 - Sequence Diagram for Earning Coins

## Sequence Diagram for refill coins

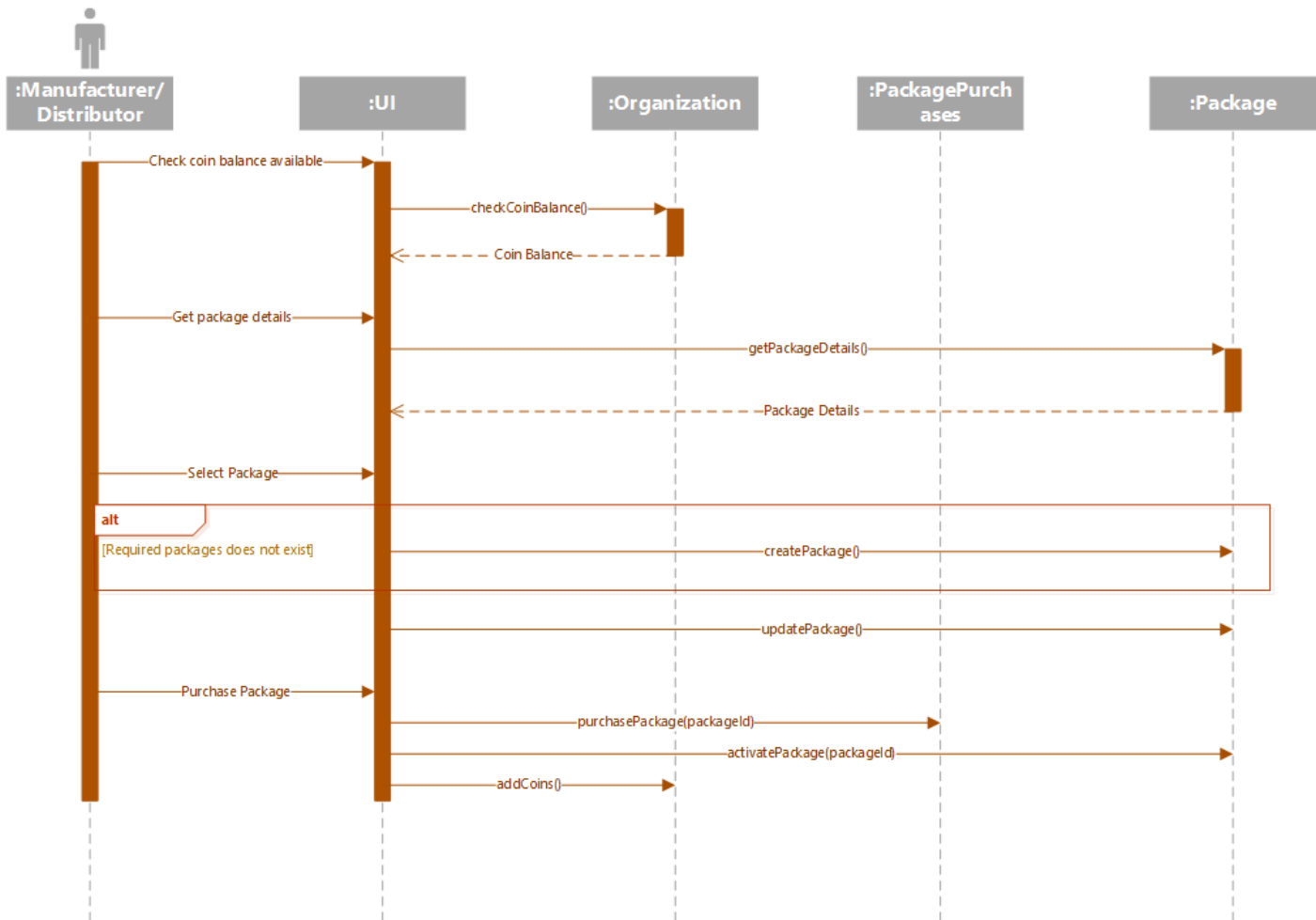


Figure 9 - Sequence Diagram for refill coins



## Sequence Diagram for spend coins

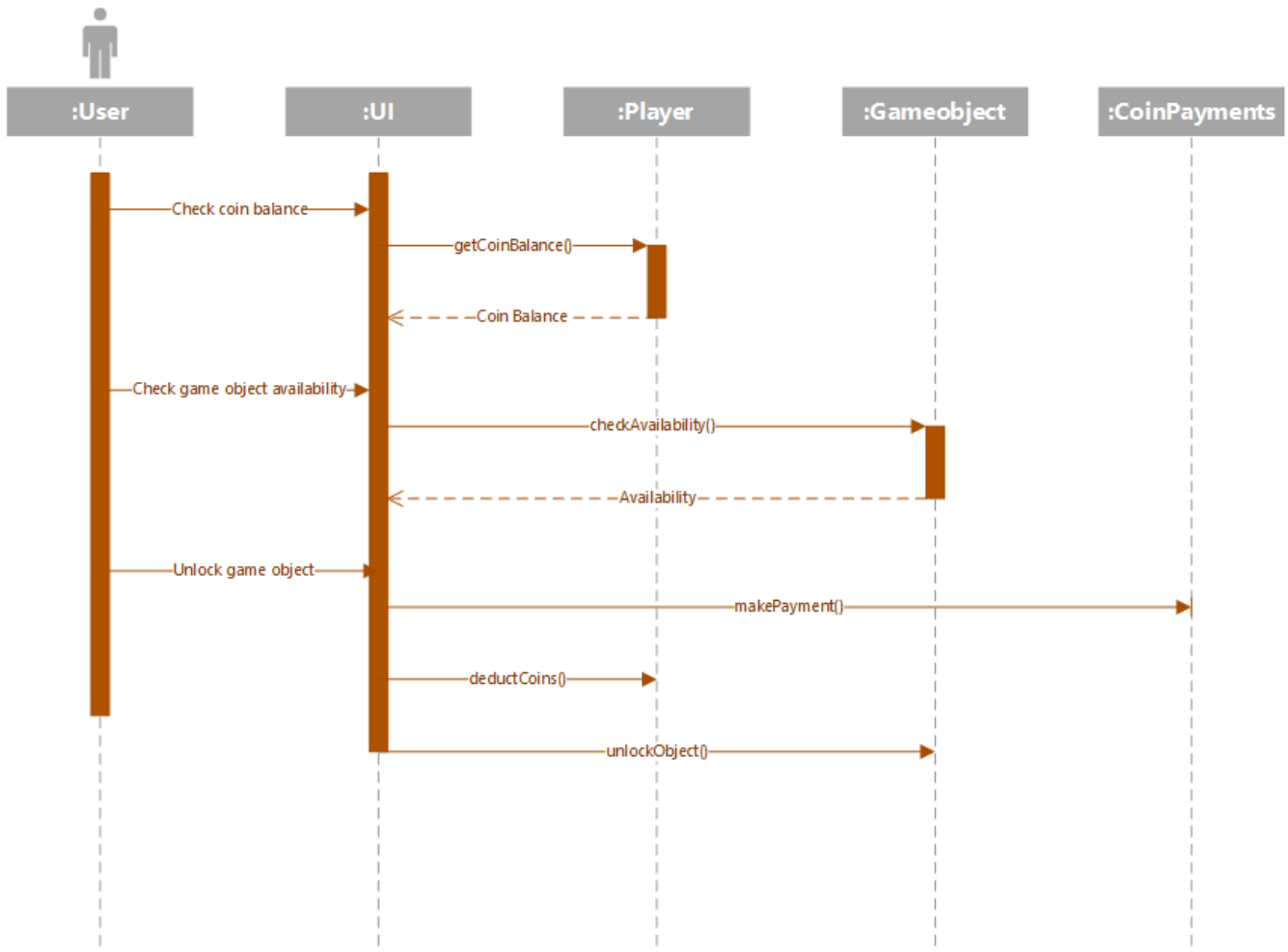


Figure 10 - Sequence Diagram for Spend Coins

### 3.1.4. Class Diagram

In software engineering, a class diagram in the Unified Modeling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects.

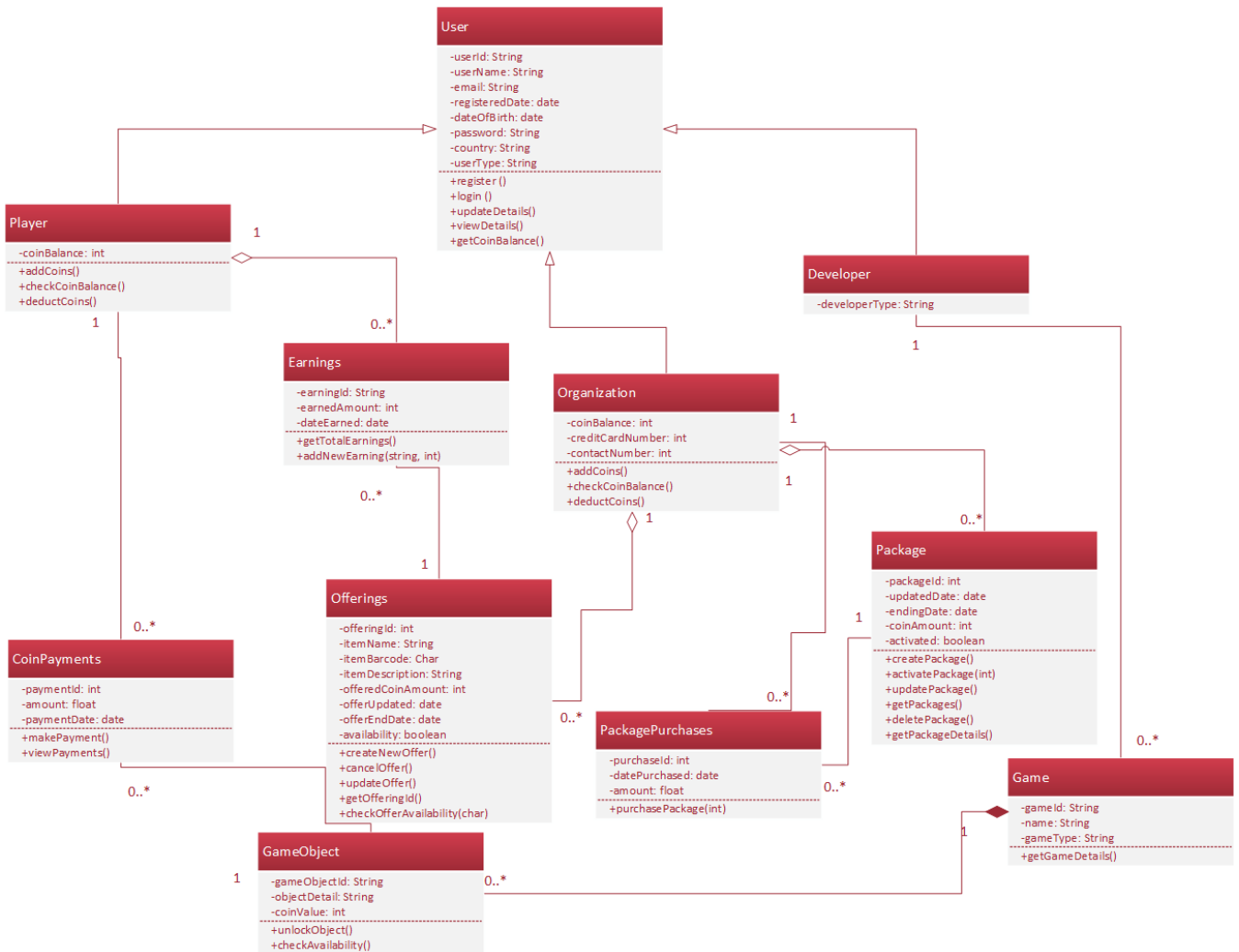


Figure 11 - Class Diagram for the Proposed System

### 3.1.5. Database Design

An entity–relationship model describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types and specifies relationships that can exist between entities.

#### Entity Relationship Diagram

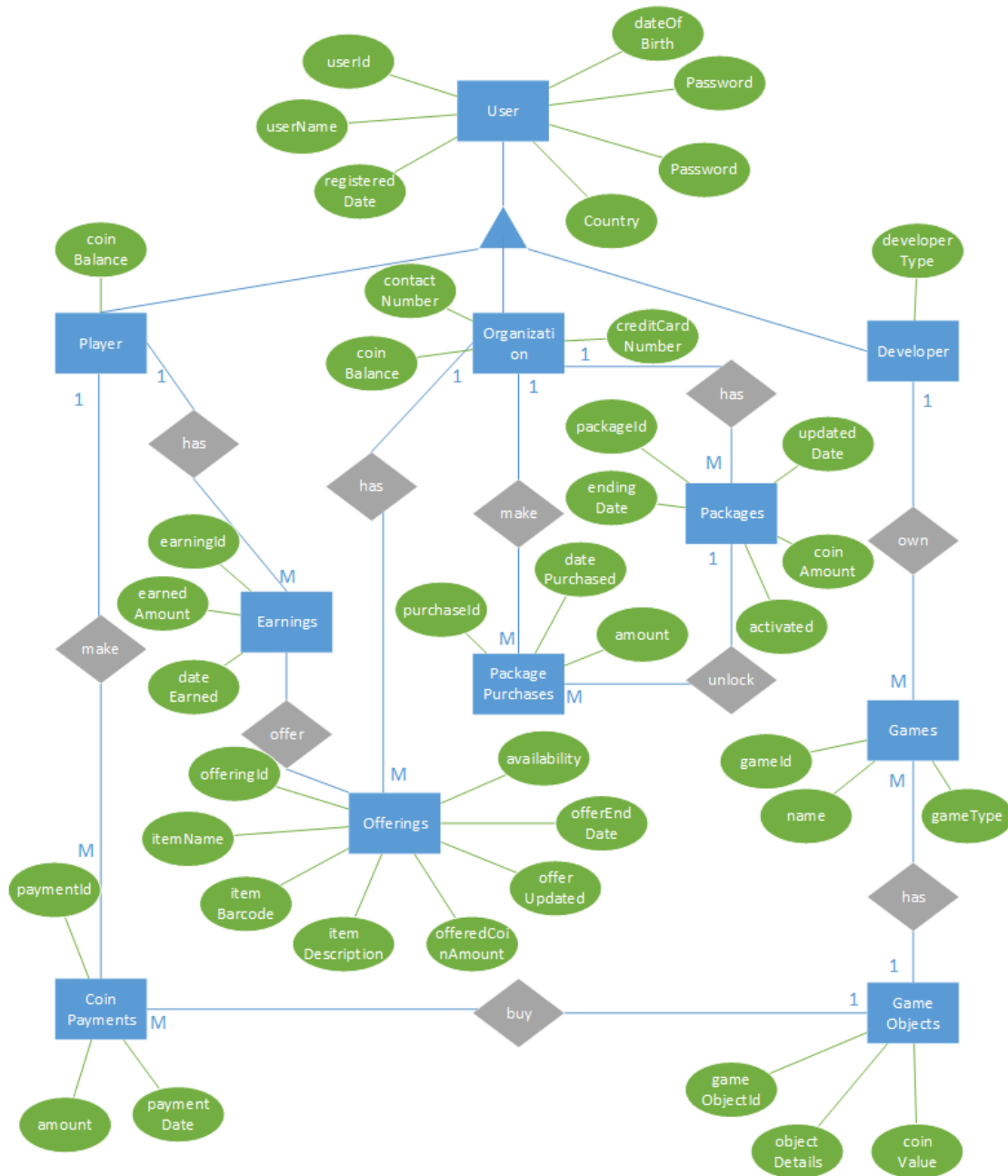


Figure 12 - Entity Relationship Diagram

## Normalized Database Design

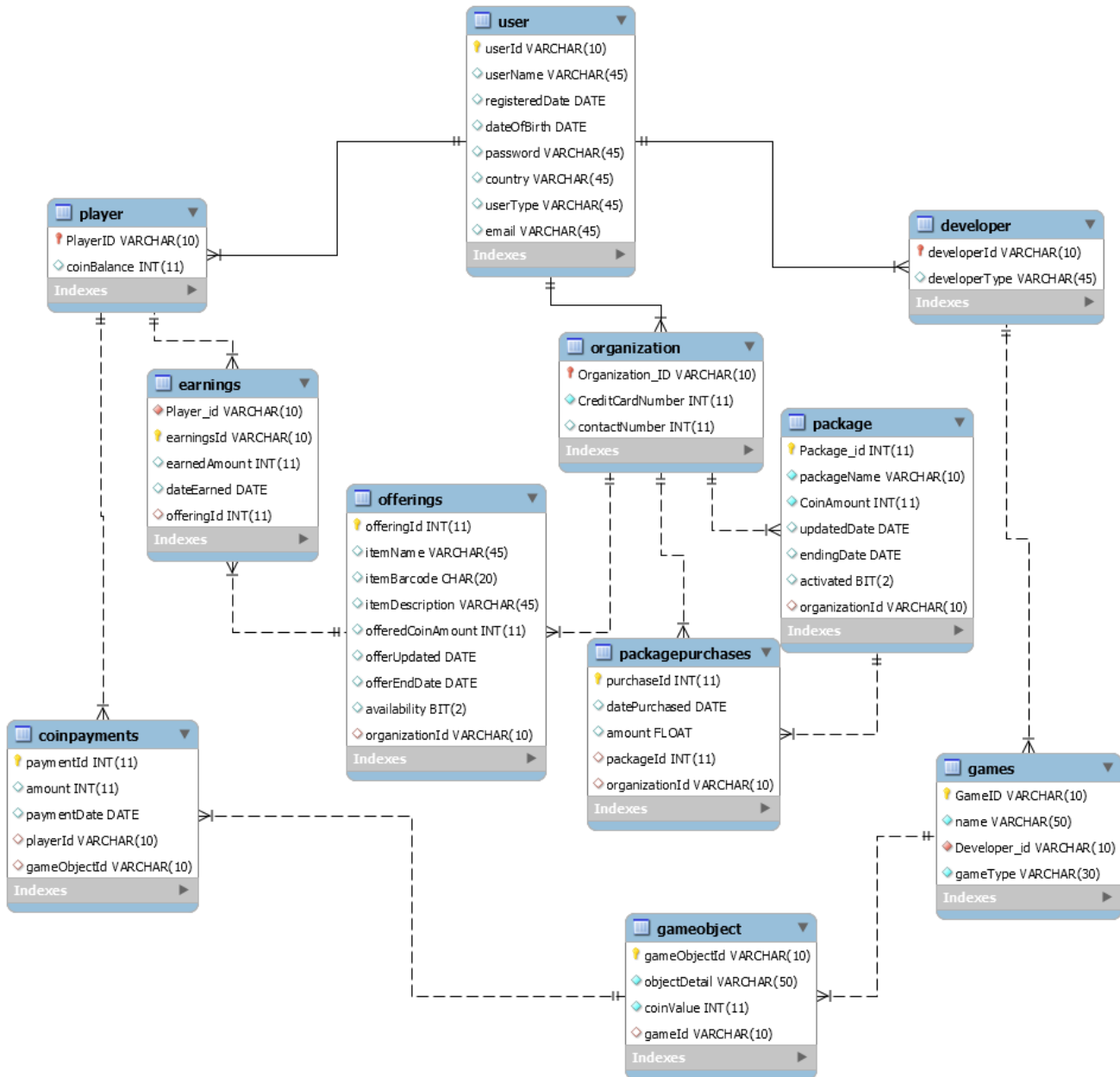


Figure 13 - Normalized Database Design

## **3.2. System Development**

This depicts the information related the development of the project. It provides a brief introduction to the technologies and the programming language properties used to implement the design. Furthermore this provides a brief description of the third party components and libraries used and the data structures and algorithms used which is of significant value to the implementation process.

### **3.2.1. Programming Languages and Development Tools**

When selecting the technologies and programming languages the availability of the software and the efficient usage of the existing hardware and software were highly considered.

#### **Programming Languages**

- PHP
- HyperText Markup Language (HTML)
- Cascading Style Sheets (CSS)
- Javascript
- C#

#### **Development Tools and Technologies**

- XAMPP
- MySQL
- Visual Studio
- Unity 3D

### **3.2.2. Third Party Components and Libraries**

#### ➤ **SWAL**

Sweet Alerts were used for messages, in order to increase the user friendliness of the system.

#### ➤ **ECHART.JS**

Echart.js was used to generate attractive charts.

### **3.2.3. Hashing of Confidential Data**

The storage of confidential and private information of the users must be paid attention, in order to make them not accessible by a third party. Those information (such as passwords) should be saved in the database in a protected/encrypted format.

## **3.3. Installation Guide**

This section consists of the Hardware and Software requirements of the proposed system.

### **3.3.1. Hardware Requirements**

Minimum hardware requirements of 'Save Me':

For Desktop application and web portal:

- CPU - Pentium 4 2.8 GHz
- RAM - 1GB or above

- Router with firewall to be always connected to the internet
- 40 GB or above hard disk

For in-app purchases using mobile phone:

- Android mobile with 1GB RAM
- Mobile data or Wi-Fi connection
- 

### **3.3.2. Software Requirements**

Minimum software requirements of ‘Save Me’:

For desktop application:

- Windows XP or higher windows OS
- Antivirus program

For web portal:

- A web browser like FireFox. If it’s IE it should be IE v11 upwards

### **3.3.3. User Manual**

This user manual will guide you through the system functionality enabling the user to easily manage activities and processes of Save Me.

There are three type of users in ‘Save Me’

#### **1. Player –**

Access Level Rights –

- Download games from portal
- Earn coins according to the items they buy from distributors.
- View coin balance

- Do in-app purchases according to the coin balance they have
- View offers available

## 2. Developer -

Access Level Rights –

- Upload new games to the portal
- Register coin payment eligible game objects to the portal
- Check the amount earned from in-app purchases
- Request withdrawal
- Analyze and generate reports about the games they own

## 3. Distributor -

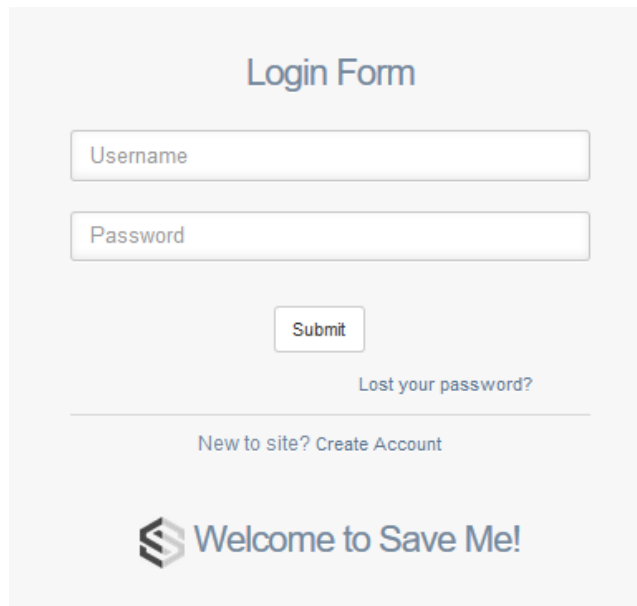
Access Level Rights –

- Check their coin balance
- Update coin balance by purchasing coin packages
- Add new coin offers
- View purchase and offer log files
- Generate strategic reports



## A. User Manual Registration and Login for Portal

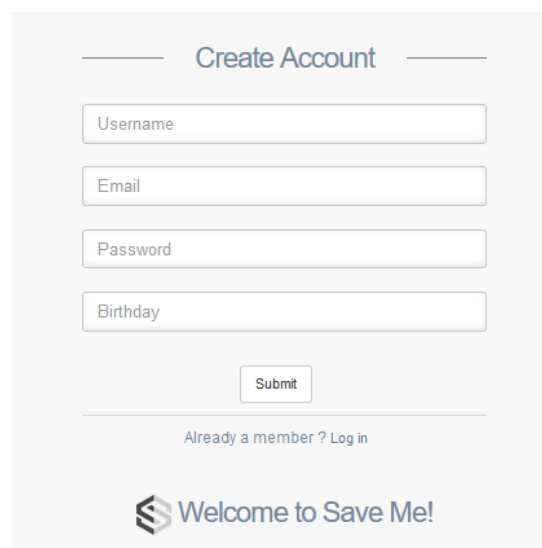
1. Browse the save me URL using your browser



The screenshot shows a 'Login Form' with the following elements:

- A title 'Login Form' at the top center.
- A text input field labeled 'Username'.
- A text input field labeled 'Password'.
- A 'Submit' button below the password field.
- A link 'Lost your password?' below the submit button.
- A horizontal line below the link.
- A link 'New to site? Create Account' below the line.
- A logo consisting of a stylized 'S' and 'M' followed by the text 'Welcome to Save Me!' at the bottom.

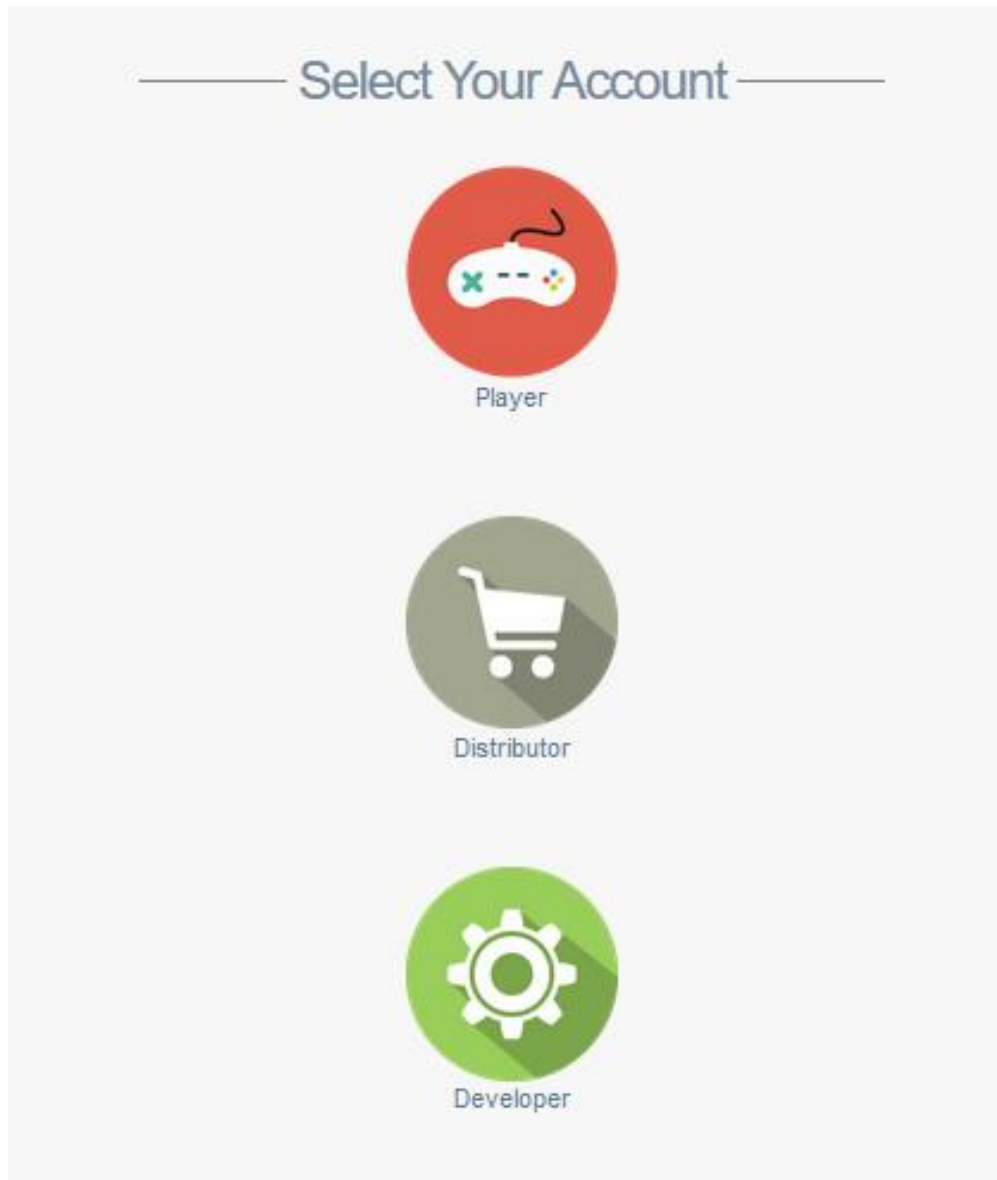
2. It will redirect you to the login page. If you are an already registered member; Enter login username, password and press Submit button.
3. If you are new to 'Save Me' portal, click on the link 'Create Account'



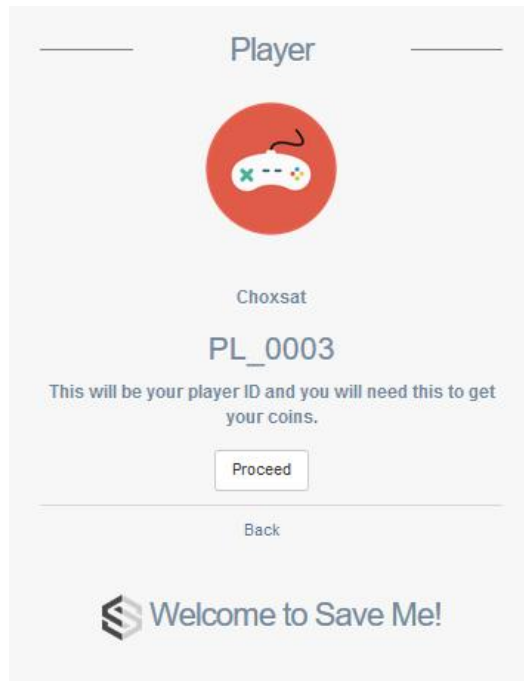
The screenshot shows a 'Create Account' form with the following elements:

- A title 'Create Account' at the top center, flanked by horizontal lines.
- A text input field labeled 'Username'.
- A text input field labeled 'Email'.
- A text input field labeled 'Password'.
- A text input field labeled 'Birthday'.
- A 'Submit' button below the birthday field.
- A horizontal line below the submit button.
- A link 'Already a member? Log in' below the line.
- A logo consisting of a stylized 'S' and 'M' followed by the text 'Welcome to Save Me!' at the bottom.

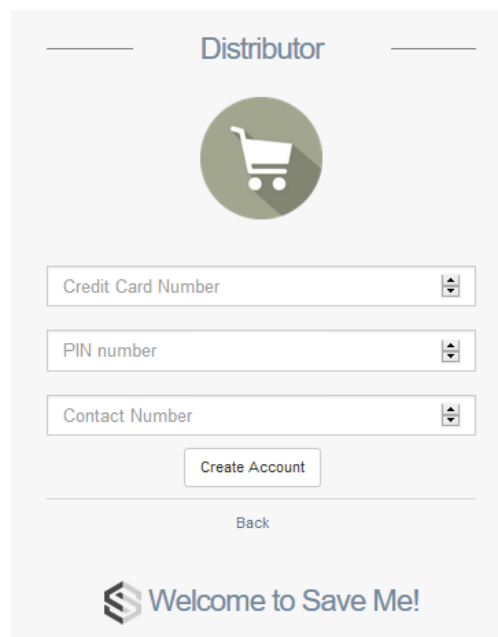
4. Insert your details and press submit. It will redirect you to the user type selection.



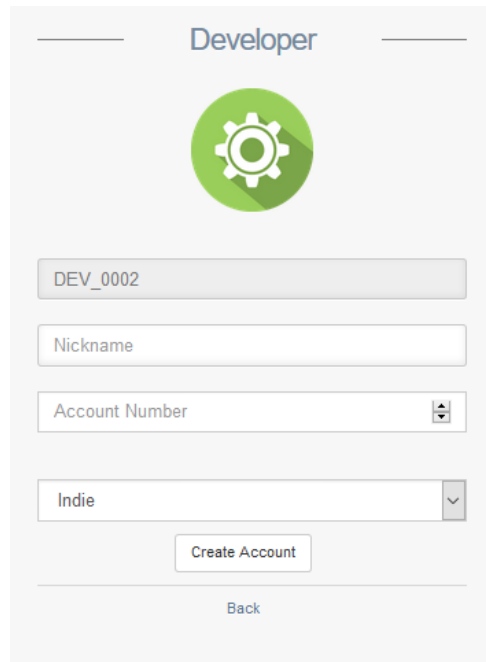
4.1 If you are a player, you will get this view. Remember your unique player ID. It'll be used to earn coins. Then press Proceed.



4.2 If you are a distributor fill the fields of the view below and press Create Account.



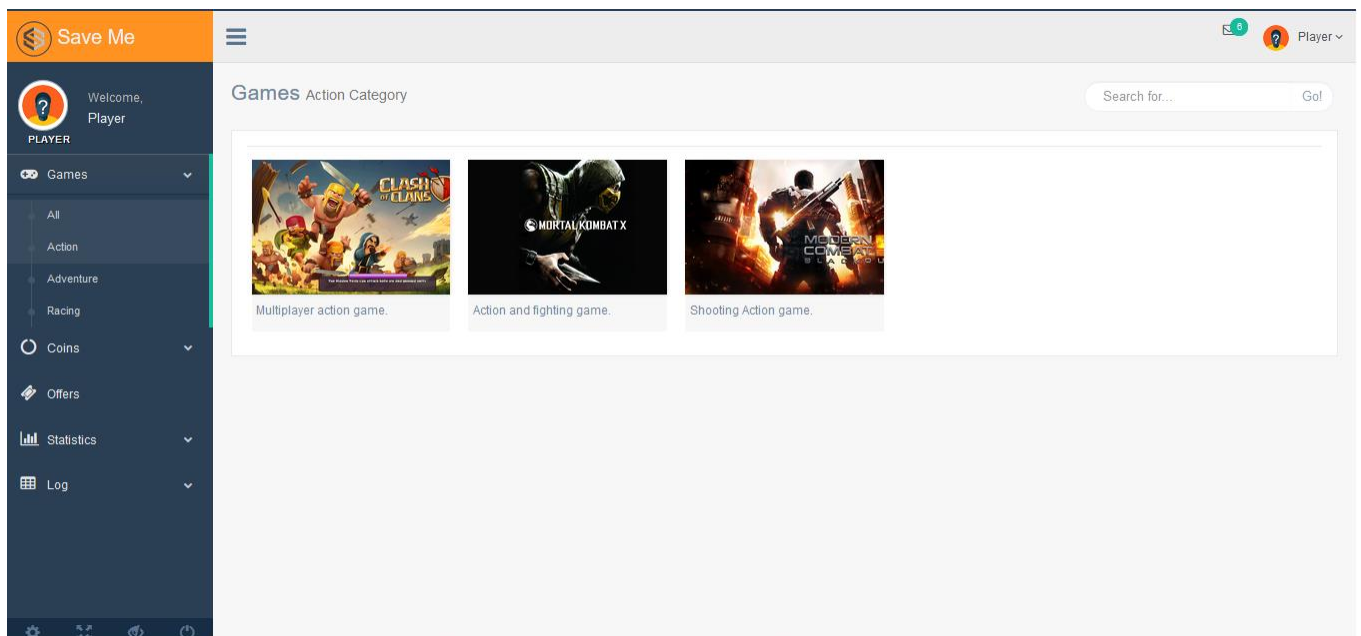
4.3 If you are a developer fill the fields of the view below and press Create Account.



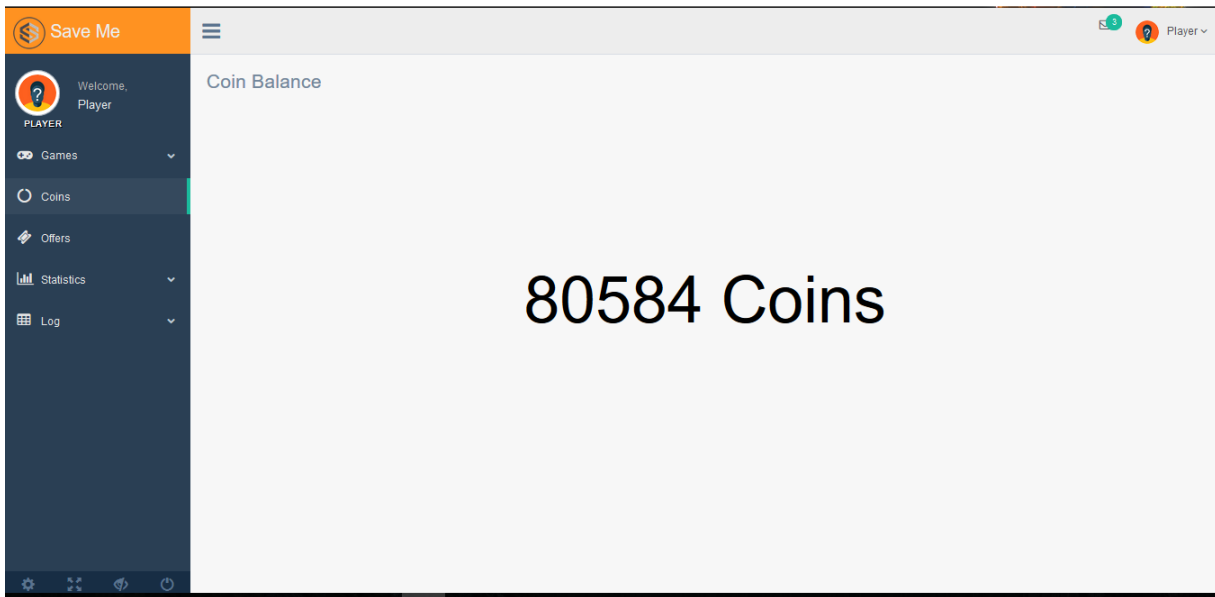
The image shows a mobile application form for creating a developer account. At the top, the word "Developer" is centered. Below it is a green circular icon with a white gear. The form contains four input fields: a text field with "DEV\_0002", a text field for "Nickname", a dropdown menu for "Account Number", and a dropdown menu for "Indie". A "Create Account" button is positioned below the dropdowns, and a "Back" link is at the bottom.

## **B. User Manual for Players**

1. If you are a registered and logged in member, then you can access your dashboard. Download games section will open up for you. You can select the game categories as well.



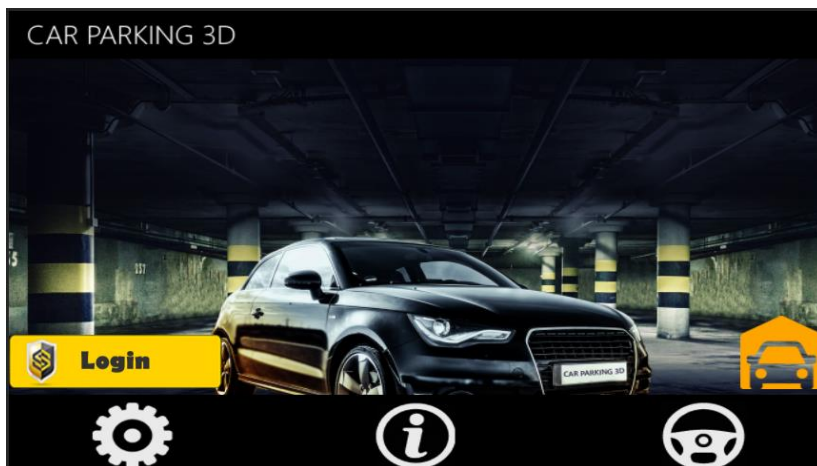
2. Hover on images and click download icon to download the game.
3. Select coins from navigation bar to check your coin balance.



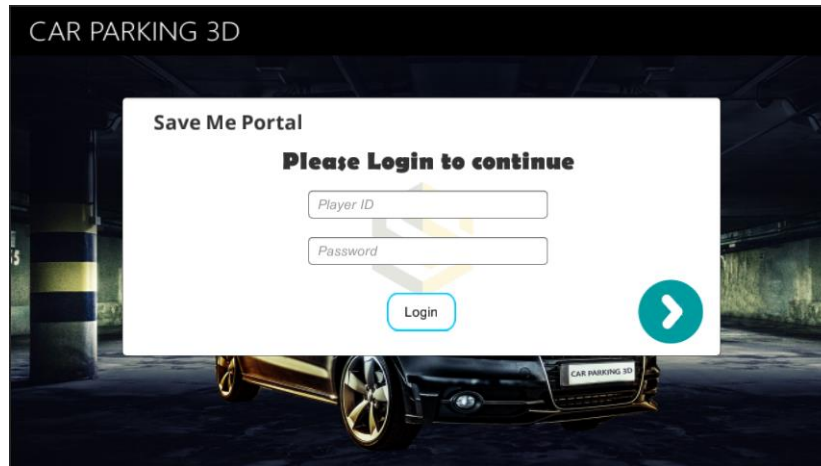
4. Select Offers from navigation bar to check available offers.
5. Use statistics and logs sections to view suggestions and logs.

### C. Mobile Game

1. Press login button in the main menu.



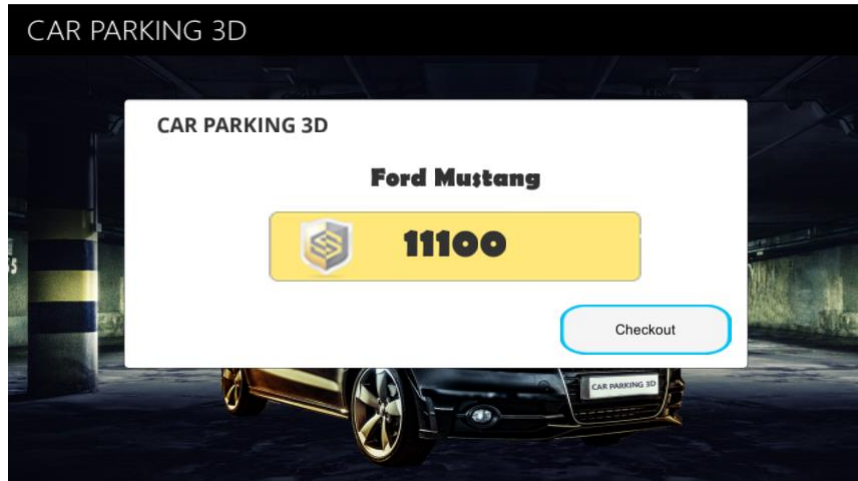
2. Enter Player ID and the password and press login



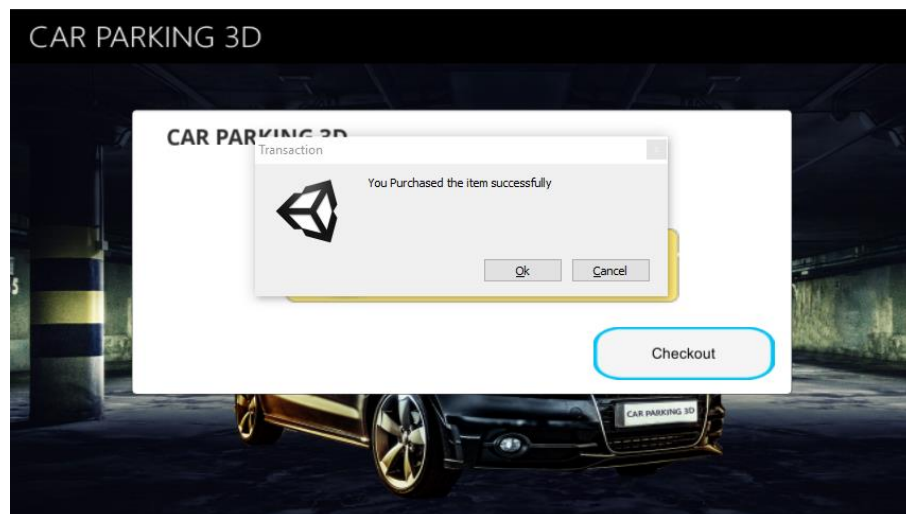
3. Go to the main menu by selecting blue arrow icon and. Press the garage icon to see the available purchasable items.
4. Select the object you like to purchase and click the yellow button of that object.



5. It will navigate you to the checkout panel. Press checkout if you are willing to buy the object.

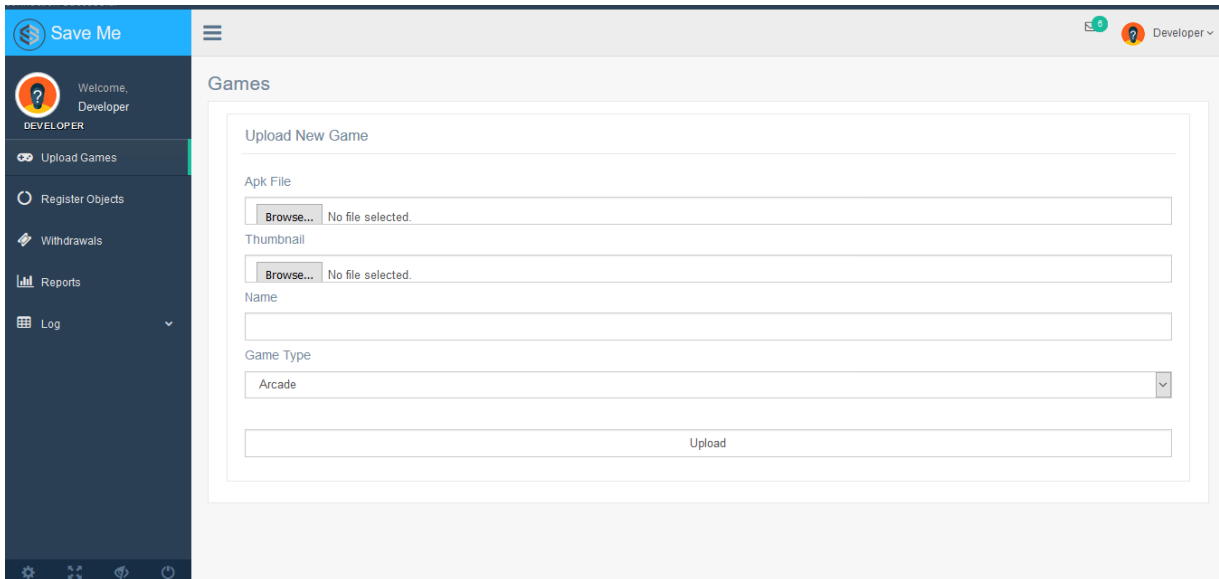


6. It will purchase the item and show a success dialog if you have enough coins or else it will show the error message.



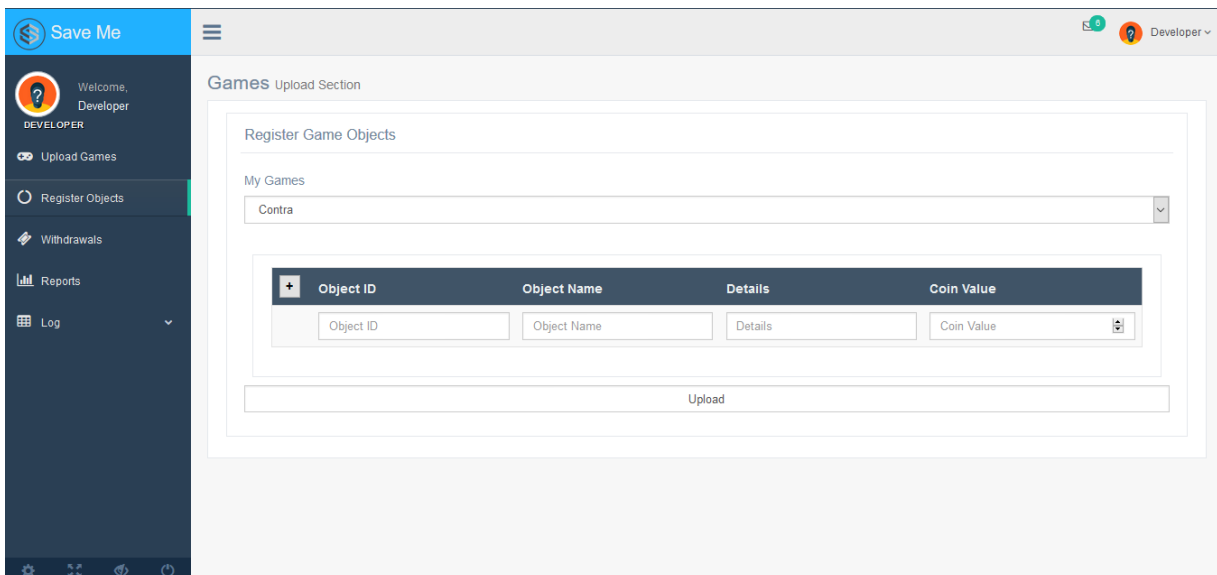
## D. User Manual for Developer

1. Develop your game and register the game in on 'Save Me' portal. You can upload them using Upload Games section of your dashboard.



The screenshot shows the 'Save Me' developer dashboard. The left sidebar contains navigation options: Upload Games, Register Objects, Withdrawals, Reports, and Log. The main content area is titled 'Games' and features a 'Upload New Game' form. The form includes fields for 'Apk File' and 'Thumbnail', both with 'Browse...' buttons and 'No file selected.' text. Below these are fields for 'Name' and 'Game Type' (a dropdown menu currently set to 'Arcade'). An 'Upload' button is located at the bottom of the form.

2. Give unique ids to the purchasable objects of your game.



The screenshot shows the 'Save Me' developer dashboard with the 'Register Game Objects' form. The left sidebar is the same as in the previous screenshot. The main content area is titled 'Games Upload Section' and contains a 'Register Game Objects' form. The form has a 'My Games' dropdown menu currently set to 'Contra'. Below this is a table with four columns: 'Object ID', 'Object Name', 'Details', and 'Coin Value'. Each column has a corresponding input field. An 'Upload' button is located at the bottom of the form.

Object ID	Object Name	Details	Coin Value
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

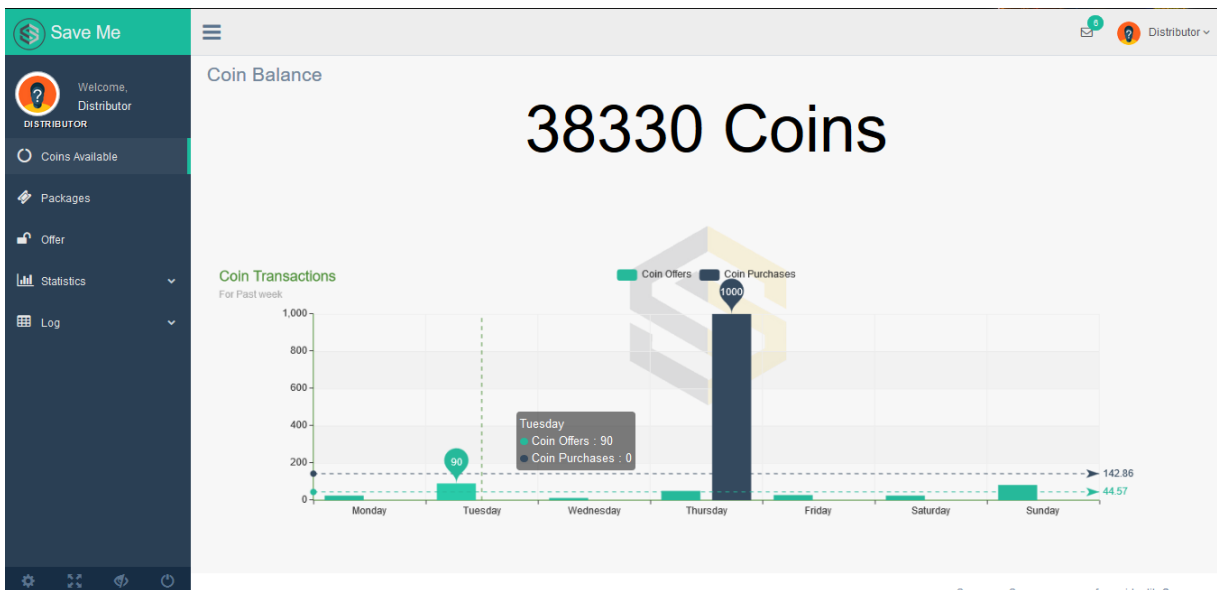
3. Register those objects on the 'Save Me' portal using Register Objects section



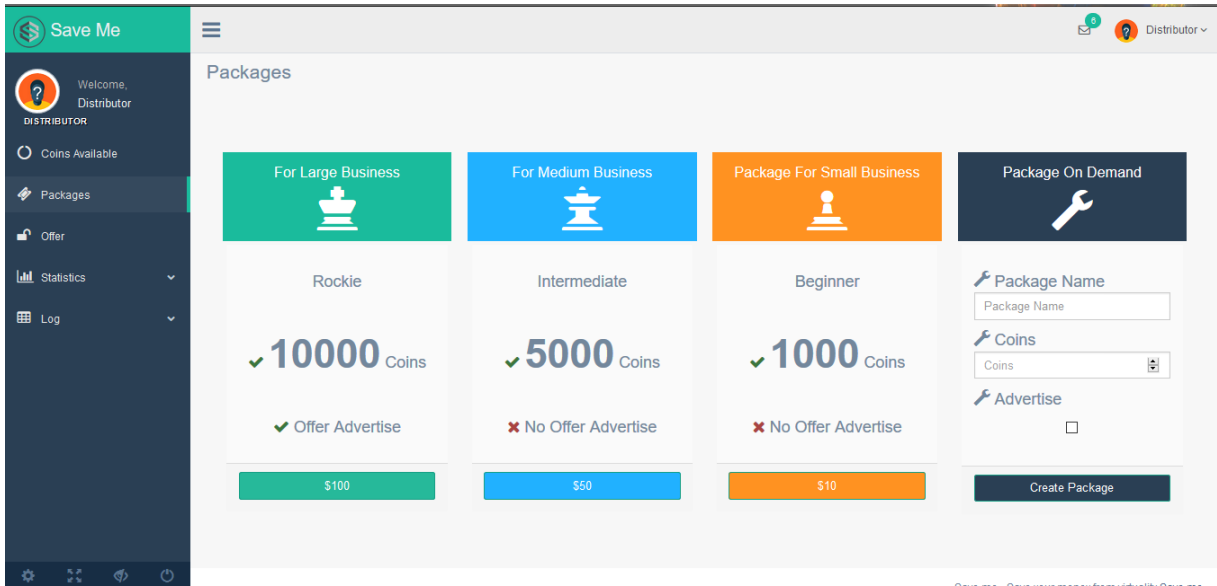
4. Use the link to sign in users to 'Save Me'
5. Use the link to request transaction from 'Save Me'
6. You can request from the withdrawals section and you can generate reports and logs in Reports, Log sections accordingly.

## **E. User Manual for Distributors**

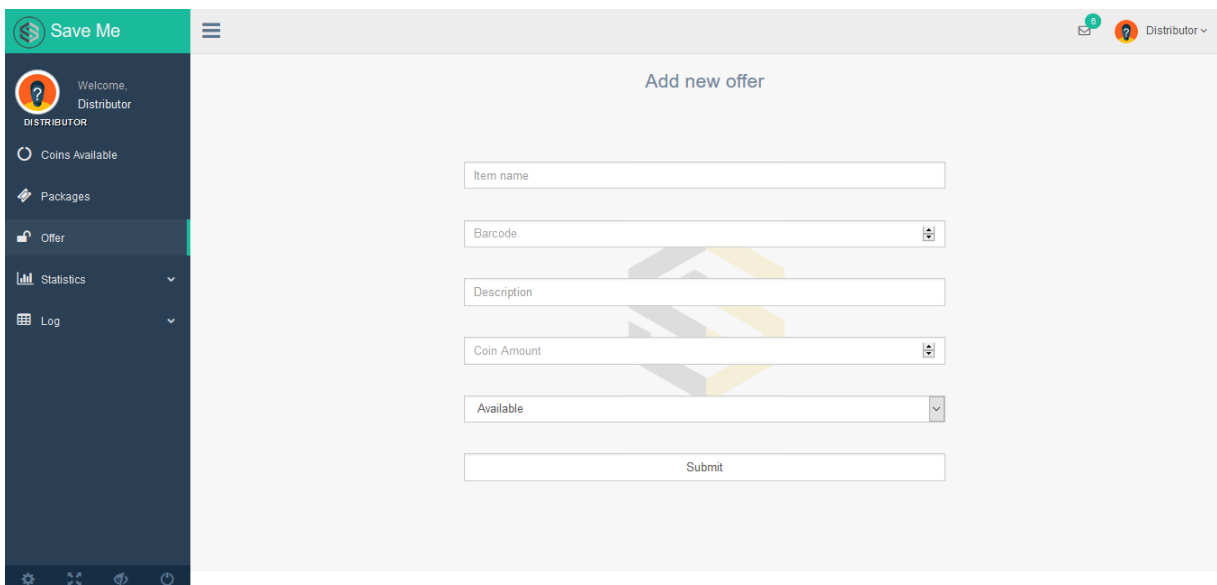
1. Use the coins available tab to view the Coins available for the organization and the coin purchases and offerings of the organization.



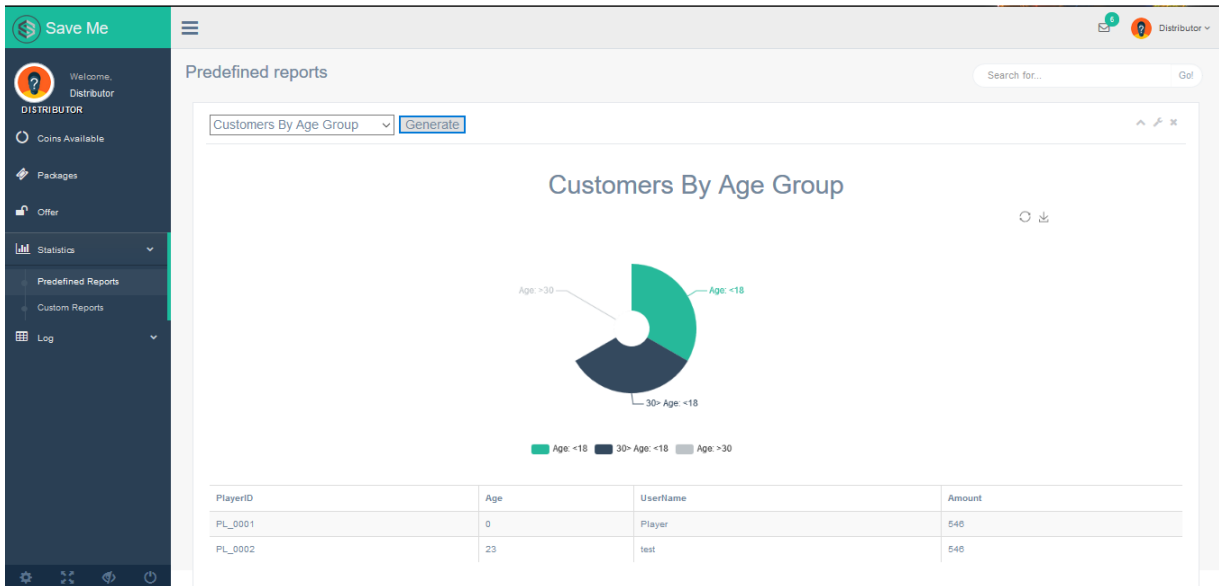
2. Packages section will allow you to buy predefined coin packages or you may create your own packages using custom packages widget. You can define a name, enter the coin amount. You can check you preference about displaying your offers in offering tab.



3. You can add new offers which players can get by using the Offer tab.



4. Statistics section shows predefined and custom reports.



5. Purchase and offering logs can be accessed using Log tab.

**Purchase Log**

Purchase ID	Package Name	Coin Amount	Advertising	Date	Payment
5	pos	123	With Advertising	2017-01-04	\$6.23
6	gfd	45	Without Advertising	2017-01-04	\$0.45
7	gfd	450	With Advertising	2017-01-04	\$9.5
8	Beginner	1000	Without Advertising	2017-01-04	\$100
9	Rockie	10000	With Advertising	2017-01-04	\$10
10	Rockie	10000	With Advertising	2017-01-04	\$10
11	Rockie	10000	With Advertising	2017-01-05	\$10
13	Intermediate	5000	Without Advertising	2017-01-05	\$50
14	Rockie	10000	With Advertising	2017-01-05	\$10
15	Intermediate	5000	Without Advertising	2017-01-07	\$50

Showing 1 to 10 of 13 entries

## **F. Desktop Application**

1. This application is for the sales center.
2. Cashier need to get the player ID from the customer. And enter it in the Player ID section and press submit.
3. If the player ID is available, it will navigate to the next window where contains the textbox to enter item ID. Cashier can scan the barcode directly. Then application will automatically submit the transactions.



The image shows a screenshot of a desktop application interface. It features two input fields: the top one is labeled 'Item ID' and the bottom one is labeled 'User ID'. Below these fields is a 'Submit' button. The interface is overlaid on a background with a large, stylized logo consisting of interlocking geometric shapes in yellow, grey, and white. The entire form is enclosed in a light grey border.

## **3.4. Summary**

In this chapter, the development aspects of the system has been elaborated under the areas of technologies, constraints, third party support and the maintenance of confidentiality. The factors such as availability, cost, performance, functionality, etc. have been considered in order to decide the technologies described above. It also has the implementation with the user manual.

# CHAPTER 4

## 4. Evaluation

Software testing and evaluation aspects of the project is discussed under this chapter. The strategies, types of testing used and why and how they have been employed to test the system have been discussed. Furthermore, it discusses about the sample test cases, followed by the test reports of the system and results illustrating the severity of the bugs identified and possible solutions. This chapter also evaluates the attempt taken in the project, including the degree of objectives met and usability of the system.

### 4.1. Test Plan and Test strategy

The functionalities are tested in order to enhance the functions and the quality of the product. Testing is done with the purpose of aligning the product with the users' requirements. The testing plan defines the items to be tested and the functions, which are selected based on the importance of the functions, and the risk of the functions on the users viewpoint.

Then the test cases were designed corresponding to the use case descriptions. They were executed manually, and the results were recorded. The bugs identified were corrected and tested again, where the functionalities were improved in Save Me.

Selected Testing Approach– Black Box Testing

### **4.1.1. Justification of the Selection of Testing Approach**

In order to test, Black Box Testing and White Box Testing were selected.

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well.

Since the system uses Unity to develop games, it creates executable files for Android and IOS automatically. The internal code of those executable files cannot be accessed since it's encrypted. Thus testing of those internal codes are impossible. White box testing requires testing of internal structures or workings of an application, as opposed to its functionality and it uses an internal perspective of the system, as well as programming skills, in order to design test cases. Since they are unknown, Black Box Testing was selected.

Under Black Box Testing, three types of testing are done. They are Concept testing (used to check the compatibility of the concepts used in the system and the concepts in the actual business process), Integration testing (After integrating the components together, which are tested, using the unit testing the entire integrated component is tested again.) and Interface testing (Interface testing is being used to test each and every component in the GUIs against their intended purpose.)

White Box Testing is a software testing method in which the internal structure/design/implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. Programming know-how and the implementation knowledge is essential. White box testing is testing beyond the user interface and into the nitty-gritty of a system. This was used for testing the functionality, user interfaces and flow in the project.

## 4.2. Test Cases

Testing Environment - Operating System: Windows 10

### 4.2.1. List of Test Cases required for the System

Table 9 - Test case for User login and registration for web portal

1. User login and registration for web portal				
ID	Test Case Description	Input Data	Expected Output	Status
1.1	User Signup Steps : 1. Press create account link 2. Input basic user details 3. Press Submit 4. Select the user type 5. Input specific details 6. Press Create Account	Username Email Password DOB Optional: Credit Card Number PIN Number Contact Number Nick Name Account Number Developer type	Accept user basic details and proceed to the user types selection.  After completing the user type details, proceed to the dashboard according to the user type.	Pass
1.2	User Login Steps: 1. Input username and password 2. Press login	Matching username and password which knows the user type	Proceed to the user dashboard accordingly	Pass
1.3	Password/Username Validation Steps : 1. Input user name 2. Input password 3. Press login	Mismatching Username and password	Display a message informing the mismatch of username and password  Allows the user to try again	Pass

Table 10 - Test case for purchasing game objects using 'Save Me'

2. Purchase Game Objects				
ID	Test Case Description	Input Data	Expected Output	Status
2.1	Player Login Step : 1. Press login button in game menu 2. Input player ID and password 3. Press login	Valid player id and the matching password	Display a message informing that the user logged in successfully	Pass
2.2	Player Login Validation Step: 1. Press login button in game menu 2. Input player ID and password 3. Press login	Mismatching player id and password	Display a message informing that the login is invalid	Pass
2.3	Item purchase Step: 1. Select game object 2. Press 'Save Me' payment button. 3. Check details and press checkout button	Automatically: Player id Object ID Game ID	Display item purchased message on success.  Notify when the player does not have enough coins  Notify when the game object has not registered in the database	Pass

Table 11 - Test case for offer coins using desktop application

3. Offer coins				
ID	Test Case Description	Input Data	Expected Output	Status
3.1	Organization Login Step : 1. Input Organization ID and password 2. Press login	Valid player id and the matching password	Display a message informing that the user logged in successfully	Pass
3.2	Organization Login Validation Step:	Mismatching player id and password	Display a message informing that the login is invalid	Pass



	1. Input organization ID and password 2. Press login			
3.3	Check user availability Steps: 1. Input player id 2. Press check button	Player ID	Display message if the player id is valid.  Notify when the Player ID is not registered in the system	Pass
3.4	Offer coins Steps: 1. Input item barcode numbers by scanning the barcode 2. Finally press submit button	Barcode number	Display message when all the transaction completed.  Notify if the item not eligible when they scan the barcode.	Pass

### 4.3. Test Report

If testing is not done throughout the development process but at the end, the system will undergo many changes even in the interface design and process of interface navigations too. Therefore testing was done as a part of the development process, and not at the end of the whole process; so that changing these at the end is costly and less effective.

The number of defects that are uncovered by the test cases and test data decreased gradually with the number of testing iterations performed. In the initial iterations large number of bugs were uncovered and most of those errors were moderate in terms of the importance.

Unit testing, Integrated Testing, Functional Testing were done by the developer.

- Concept Testing: Acceptable
- All Units testing: Passed
- All Integrated testing: Passed
- UI testing: Average, Acceptable
- All Functional Testing: Completed
- System Testing: Partially Completed

System Completion:

1. Web portal for game downloading and transactions – Fully implemented

Result: Completed

2. Sample mobile games– Two mobile games developed

Result: Completed

3. Desktop application for distributors – Fully implemented

Result: Completed

Concept testing, Unit Testing, Integrated testing, Functional testing tests the functionality of the system and compares with the requirements of the system which increases the reliability and quality of the system. UI testing increases the usability of the system, which then increases the quality of the system as well. Errors were found in Unit, Integrated and Functional Testing. But they were not catastrophic or serious errors and were Tolerable. Developer was able to handle the exceptions caught when debugging the system. Errors found in UI and Concept testing were taken in to consideration and were re-modified according to their existence in the design phase and development phase.

#### **4.4. Degree of Achieved Objectives**

The main objectives of the project were to let the players to get a good experience in gaming by letting them to purchase virtual upgrades using ‘Save Me’ coins instead of using real money, increase the profit of the developers by increasing the number of in-app purchases and let the distributors to increase their sales by offering ‘Save Me’ coins as discounts for their products. Apart from that it provides significant support for decision making through the system. Decision making is supported by calculations, where the decision would be more reliable and accurate.

#### 4.4.1. Objectives Analysis

Table 12 - Cross Check Table with objectives

Status	Requirement
Done	Shall be able to download games
Done	Shall store player details
Done	Shall be able to purchase game objects
Done	Shall be able to track the purchases of the player
Done	Shall be able to get discounts for virtual purchases
Done	Developers shall be able to add new games
Done	Developers shall be able to earn money
Done	Shall be able to categorize the games
Done	Shall be able to review games
Done	Shall be able to get comments from players
Done	Shall be able to generate reports
Done	Shall be able to access to the purchasing screen easily

With all these functionalities,

- The system lets the players to get a good experience in gaming by letting them to purchase virtual upgrades using ‘Save Me’ coins instead of using real money.
- The system increases the profit of the developers by increasing the number of in-app purchases.
- The system lets the distributors to increase their sales by offering ‘Save Me’ coins as discounts for their products.

Thus, it is justifiable to state that the system has met its objectives.

## **4.5. Usability, accessibility, reliability and friendliness**

From the commencement of the system design and the development process, usability aspects of the system has been considered. All the technical aspects of the system have been hidden from the system user through the interactive user interfaces. Developer has followed effective pre-defined principles when designing user interfaces in order to improve usability of the user interfaces and to aid the user to avoid errors through error and warning messages. In addition, the system speaks the users' language and follows the natural and logical order of precedence according to the users' behavior. System also maintains the consistency throughout, use standard styles in all pages, to avoid complications.

Enabling and disabling options when necessary, use of drop down list to select from menus are some of error prevention methods used. Use of regular expression validation and scripts are preventing users to enter invalid data. In addition, the testing has been done throughout all the design and implementation phases. User friendliness and the flexibility of Save Me is high as the process has been customized as per the requirements of the system.

## **4.6. Summary**

In this chapter, the test types are elaborated, which the developer has followed when implementing the testing procedure and test cases of the system. Furthermore the test strategies used, test plan, sample test cases and the report of the overall test phase are described. Furthermore the developer has described about the degree of objectives met after implementing the project and usability of system.

# CHAPTER 5

## 5. Conclusion

The use of the mobile phone has grown exponentially, and is now the preferred device to access the Internet. Even in a developing country like Sri Lanka, the use of smart devices even in areas of the country where other basic infrastructure is limited is well documented. The younger generation is tech savvy and companies have developed mobile apps to facilitate easy interfacing with their customers. Gaming, which is a relatively new concept in Sri Lanka is rapidly catching on with the millennials and even the older Y – generation embracing it. It is known that in-app purchases are required for many gamers, in order to improve the gaming success and gain a competitive advantage among other players. In the perspective of the developers, in-app purchasing done by gaming individuals is the main revenue stream. However, consumers in the developing countries like Sri Lanka, the tendency for gamers to engage in in-app purchasing is limited. Limited disposable income is one of the primary reasons. Therefore, revenue generation from gamers from this source is limited. The gamers themselves are left frustrated that they cannot fully utilize features of the game. If in-app purchasing could be increased, mobile game developers will be able to gain the opportunity to expand their market and increase their revenues. The “Save Me” application has been designed as an alternative smart solution for in-app purchases and provides significant benefit to all stakeholders.

There has been an exponential growth in the proliferation of mobile apps over the last decade. It has become imperative that companies use these mobile apps to connect to existing and potential customers to sustain or build new revenue streams. The popularity of gaming has been growing and with connectivity improving in the developing nations, the potential for growth in these countries with a large tech savvy youth population is huge. Many revenue models are used by developers of games to generate revenue streams for their businesses. The most common strategy used is to offer the game at no initial cost to download and play, advancement in levels of the game require in-app purchases to improve the gaming success and gain competitive advantage among players. In the perspective of the developers, in-app purchasing done by the

gaming individuals is the main revenue stream. In developing countries like Sri Lanka, players of these games are reluctant or unable to spend money to purchase these in-apps, since there is no financial benefit other than the personal achievement in having succeeded in the game and the bragging rights associated with it. The other main constraint is that payments have to be made via credit card. Most young people do not have or have access to credit card facilities. Thus, developers of game products are repressed due to the lack of monetary incentive to build new gaming products and the players are inhibited by not being able to utilize the total advantages provided by the game. “Save Me” is a product designed to overcome these constraints by introducing an alternative method for payment and thus provide an incentive for mobile game developers to expand their market and increase their revenue.

Different alternatives were listed out to solve the issue of inadequate in-app purchasing, and were assessed based on cost benefit analysis.

- Alternative 1 (Base Case - A realistic representation of expected future conditions without the project) – Using In-app purchasing strategies in games for the players to use, if they are interested.
- Alternative 2 – Developing a new system to use virtual versus actual coins (By utilizing the money paid by customers through the widely prevalent Supermarket chains)
- Alternative 3 – Selling the game for a fixed amount to the player, without making it available for free.
- Alternative 4 – Publishing the game free for the player and displaying advertisements to generate income

Ultimately, considering the cost-benefit and sensitivity analysis, Alternative 2 was selected, which is to develop a new product “Save Me” to use virtual coins instead of actual coins. This alternative includes three parties to use the system developed by the organization i.e. players, game developers and the owners of shopping malls/ supermarkets. The process involves a few steps.

1. The distributors could assign a discount to products that they wish to promote among the target market which engage in gaming in the country. Instead of giving monetary discounts, the owner could purchase virtual coins from “Save Me”, and assign those coins instead of the

discount, which will be received by the customers who are registered in “Save Me”. This will increase the sales of the grocery item rather than just giving a discount as potential customers will visit that particular outlet to purchase the item to gain access to the virtual coins cache. .

2. The customers could use those coins he/she earned by buying a day-to-day item, or to purchase Gaming Objects (Usually provided by In-app Purchases). This will provide the customer the option of using the coins to purchase gaming benefits, without exposing his credit card information and overcoming the constraint of minimal disposable income.

3. The game developers who use the “Save Me” API to develop games will be gaining a revenue via the purchases done by the players.

A comprehensive mobile application solution along with a web-based system was developed using Unity, C#, HTML, CSS and JS. The solution includes three stakeholders. I.e. gamers, developers and manufacturers of goods and services targeting the gaming segment of the population. The process will commence with manufacturers / distributors assigning a discount to their products, in order to boost their revenue. Instead of giving customers the monetary discounts, the manufacturers will purchase virtual coins from “Save Me”, and assign those coins to selected games on the portal. Customers who have registered on the portal, can earn these virtual coins instead of the discount, by progressing through the game and collecting these virtual coins. They can then exchange these virtual coins by going to their local supermarket which acts as a distributor and exchange the coins for the physical product that the manufacturer wants to promote. The game developers who use “Save Me” API to develop games will be sharing the revenue via the purchases done by the players together with the portal owner.

## **5.1. Limitations and Drawbacks**

1. The support of a reputed commercial bank is needed for this system for implementation.
2. Since Play Store the largest online Android game store does not allow third party transactions inside applications. Therefore until the system get reputed by Play Store, system can use only the games hosted in Save Me portal.
3. Even if the system is cross platform, since some vendors like Apple does not allow transactions inside their store it's not possible to implement it in those stores.

## **5.2. Future Work**

- More strategic and customizable reports can be added.
- Add offline capability to the desktop application.
- Add functionalities to add offerings and view logs from the same desktop application.
- Need to give the permissions to developers to advertise their games inside the portal.
- 'Save Me' coin system can be added not only to the game object but also for all the online transactions as well.

## **5.3. Summary**

In this chapter, a conclusion is given with the limitations, drawbacks and the possible improvements for the system. It also provides a clear explanation of the work done by the developer and the success of the project.



# References

- [1] R. Tamborini and P. Skalski, “The Role of Presence in the Experience of Electronic Games”, International Conference of Communication and Technology Division of the International Communication Association of New York, 2005
- [2] P. Roma and D. Ragaglia, “Revenue models, in-app purchase, and the app performance: Evidence from Apple’s App Store and Google Play”, Electronic Commerce Research and Applications, Volume 17, Pages 173-190, 2016
- [3] C. Newman and B. George, “Generating income for a beneficiary organization and loyalty points using purchases by a consumer”, 2004