# Appendix A: Base Project Requirements

Membership related information:

* Membership types: Bronze, Silver, Gold
* PEGI Age group: 12, 16, 18

Bookings:

* Hardware: PC / Xbox One / PlayStation 4
* Date & Time / Duration
* Prices

Game Software:

* Chart / Classic
* Single / Multi Player
* PEGI Age group: 3, 7, 12, 16, 18

eSports Events:

* Date & Time
* Number of Tickets

The Game Café is likely to want to perform the following tasks:

* View/Add/Update/Maintain membership information
* View/Add/Update/Maintain booking information
* View/Add/Update/Maintain eSports event information
* Associate bookings with both members and non-members and hardware/software as appropriate
* Associate eSports event tickets with members

The prices for sessions are:

* Members: 1hr £1.50 / 2hr £2.50 / 5hr £4.00
* Non-member supplement: £0.50

Along with the Game Café System holding records of Membership details, such as name, address, telephone number, date of birth (if younger than 18 years of age) and the type of membership, this information must be encrypted in the production version of the Game Café System. But such encryption is not necessary in the prototype (although, there should be a plan for encryption). In addition, the Game Café System is to hold details for eSports Events.

# Appendix B: User Stories

## Game Café Staff Member User Stories

* As a Staff-Member, I want to be able to view current booking information, to check on what bookings have been arranged
* As a Staff-Member, I want to be able to add new booking information, should a café member wish to arrange a booking
* As a Staff-Member, I want to be able to update current booking information, if a café member wants to change the details of one of their bookings
* As a Staff-Member, I want to be able to maintain current booking information, to make sure a booking’s details are correct
* As a Staff-Member, I want to be able to view current membership information, to check on the details of current café members
* As a Staff-Member, I want to be able to add new membership information, if a non-member, wishes to become a café member
* As a Staff-Member, I want to be able to update current membership information, if a café member’s situation is modified
* As a Staff-Member, I want to be able to maintain current membership information, to make sure any café member’s details are accurate to date
* As a Staff-Member, I want to be able to associate bookings with members, to see which booking was made by which café member
* As a Staff-Member, I want to be able to associate bookings with non-members, to see which booking was made by which non-café member
* As a Staff-Member, I want to be able to associate bookings with hardware and the software that is available on that piece of hardware, as each booking can only be made for a certain piece of hardware, which can only run certain pieces of software
* As a Staff-Member, I want to be able to associate eSports event tickets with members, so we know which café members, are attending an eSports event

## Game Café Member (Patron) User Stories

* As a Member, I want to be able to make bookings, for myself or a non-member, at a certain date and time, to use a certain piece of hardware, for a particular price, to be able to play games associated with that piece of hardware
* As a Member, I want to be able to get eSports Event Tickets, for any eSports Events scheduled to take place at the Game Café, at a certain date and time, for myself, so long as there are tickets left for the event, so that I can go to that event

# 

# Appendix C: Requirement Definition

## **Functional Requirements**

The functional requirements are the basic stories which the program must satisfy to properly function:

**The user must be able to interact with UI elements with the mouse**

This is required to allow the user to navigate the system, select options from menus, select input, etc. which allows the user to control the program.

**The user must be able to input information using the keyboard**

This is required to allow the user to type in required fields, such as to search a database or add new instances/fields.

**The user interface must be easy to read and use, with a consistent aesthetic style**

This helps to ensure that users understand how to use the program, and so can navigate and perform the function of the program.

**The user must be able to view database entries, as a list of all entries and individual entries**

This allows the user to view the data which will be used in the program.

**The user must be able to search a database by name of data entry**

This allows the user to find specific data when required.

**The user must be able to sort a database by each individual data field**

This allows the user alternate ways to view their data based on individual fields to find differences, patterns, etc.

### **The user must be able to add new entries to a database**

This allows the user to extend the database when new data entries are required.

**The user must be able to maintain data entries**

This allows the user to edit, update and remove data entries when required.

**The system must associate bookings and ticket purchases with members for pricing**

This allows the program to automatically adjust pricing depending on whether the customer is in the member database.

## **Non-Functional Requirements**

The non-functional requirements are features which are not essential for the program to function, although they are required for the program to be successful:

**The system must give separate user privileges to a standard user (Café employee) and an Administrator**

This gives extra functions to administrators, as they are in control of the program.

**The response time when a UI button/element is pressed should be no longer than 0.5 seconds**

Navigating through menus should be fast and so a quick response time is needed so that the user does not become frustrated.

**The program should run on Windows 7 and above.**

This ensures there are no compatibility issues with running the system on different devices.

**Databases should be encrypted so that data cannot be stolen.**

This prevents sensitive or private information being stolen which could violate the privacy of customers.

**Sensitive data should be hidden to users.**

This prevents private information from being stolen or manipulated.

These requirements can now be interpreted as tasks, for use in the SCRUM project-management system.

This will allow for appropriate tracking of the task’s current status (e.g. level of completion, who the task has been assigned to and the importance for the task to be completed), for each Sprint of the project.