# 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.

# Appendix D: Figures

[Figure 1: The base Game Cafe Mind Map, using the base-requirements suggested in the assignment brief. (Chris Pryor, 2018) D- 2 -](#_Toc513803879)

[Figure 2: Game Cafe Management System WBS. D- 3 -](#_Toc513803880)

[Figure 3: Game Cafe Robustness Diagram for Staff Members. D- 5 -](#_Toc513803881)

[Figure 4: Sequence Diagram for a Staff Member to add information to the management system's database. D- 6 -](#_Toc513803882)

[Figure 5: A screenshot of the Trello board our team used throughout the project. D- 7 -](#_Toc513803883)

[Figure 6: Structure Chart for adding new entries to the Game Cafe Database. (Chris Pryor, 2018) D- 8 -](#_Toc513803884)

[Figure 7: Use-Case Diagram for Game Cafe Staff Members and Members (patrons) of the Game Cafe. D- 9 -](#_Toc513803885)

[Figure 8: The Basic Class Diagram for the Game Cafe Management System (given the initial set of derived Use-Cases). D- 10 -](#_Toc513803886)

[Figure 9: My Weekly Time-log for the project (the image flows over two pages). D- 11 -](#_Toc513803887)

[Figure 10: A screenshot of the Visual Studio solution for our project. D- 13 -](#_Toc513803888)

[Figure 11: The last 3 days of commits, from the 07/05/2018 to the 10/05/2018. D- 14 -](#_Toc513803889)

Figure 1: The base Game Cafe Mind Map, using the base-requirements suggested in the assignment brief. (Chris Pryor, 2018)

Figure 2: Game Cafe Management System WBS.

Project duration: 15 weeks (23/01 – 11/05)

### **Sprint Breakdown**

* Project broken down into 3 sprints
* Sprint duration is 5 weeks per sprint
* Sprint tasks will be broken up into tasks to be accomplished each week
* Team will hold a weekly meeting to discuss progress and establish new targets

#### Sprint 1: 23/01 – 27/02

In this sprint we aim to establish our goals as a group to successfully plan our approach for completing this project. The main target for this sprint is to complete all the planning, analysis and design documents which will allow us to fully flesh out our ideas so that we understand how to build our system and how it will work, ensuring that all requirements are met. We will then work on an initial prototype build so that we have something to show the client at the end of the sprint to show our progress and guarantee the feasibility of the program.

**Sprint Deliverables:** All planning, analysis & design documentation, working prototype which demonstrates feasibility – should be able to ‘access, add to and otherwise manipulate appropriate data within a storage medium of your choice’.

#### Sprint 2: 06/03 – 10/04

This sprint will concentrate on what we have completed successfully as a team and interviewing the customer for further information.

From Sprint 1, we have a basic prototype of the booking system as well as the relevant design documentation.

We have received further instruction from the customer on further requirements to be added, these are:

* As a user, I wish to add a member ‘UnrealDonaldTrump’ to the data stored by the system.  This member is to be registered as an adult with all privileges to play any game on any platform.
* As a user, I wish to add the following platforms to the data stored by the system - ‘Sony PlayStation PS4 Pro’, ‘Microsoft Xbox One X’ each coupled with Acer R240 24-inch Monitor.
* As a user, I wish to add the following games to the data stored by the system - ‘Forza Motorsport 7’ for the Xbox One, PEGI All ‘Gears of War 4’ for Xbox One, PEGI 18, ‘FIFA 18’ for PlayStation 4 and ‘Horizon Zero Dawn’ for PlayStation 4, PEGI 16.  The software should be linked to the platform on which they play.

**Sprint Deliverables:** These requests are trivial and can be added into the database ready to be selected. (Chris Youd, 2018)

#### Sprint 3: 17/04 – 08/05

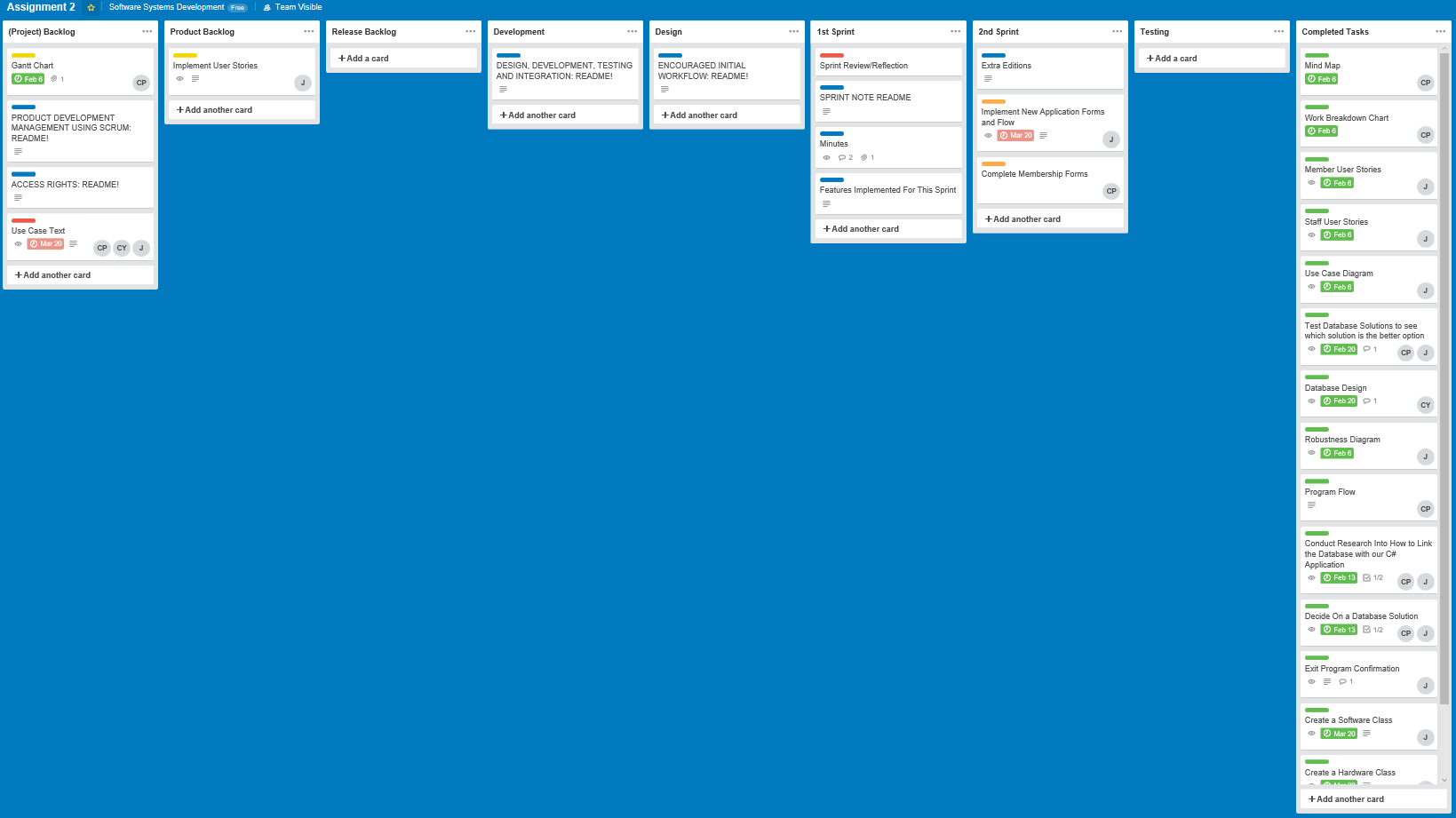
For this sprint we have had further instruction from the customer to requirements to be added:

* As a user I wish to create a booking for a named adult member who wishes to use a PlayStation 4 to play Far Cry 5 (PEGI 18) on release date (27th March2018) at 16:00 for one hour.
* As a user I wish to create an e-sports event which will be an evening competition featuring Counter Strike: Global Offensive.  The maximum number of participants is four teams of five players.  The date of the event is Friday 27th April.
* As a user I wish to book as a participating team for the event above, the name of the team is StudioCoders.

**Sprint Deliverables:** Adding bookings and events to the system along with teams for events. (Chris Youd, 2018)

Figure 3: Game Cafe Robustness Diagram for Staff Members.

Figure 4: Sequence Diagram for a Staff Member to add information to the management system's database.

Figure 5: A screenshot of the Trello board our team used throughout the project.

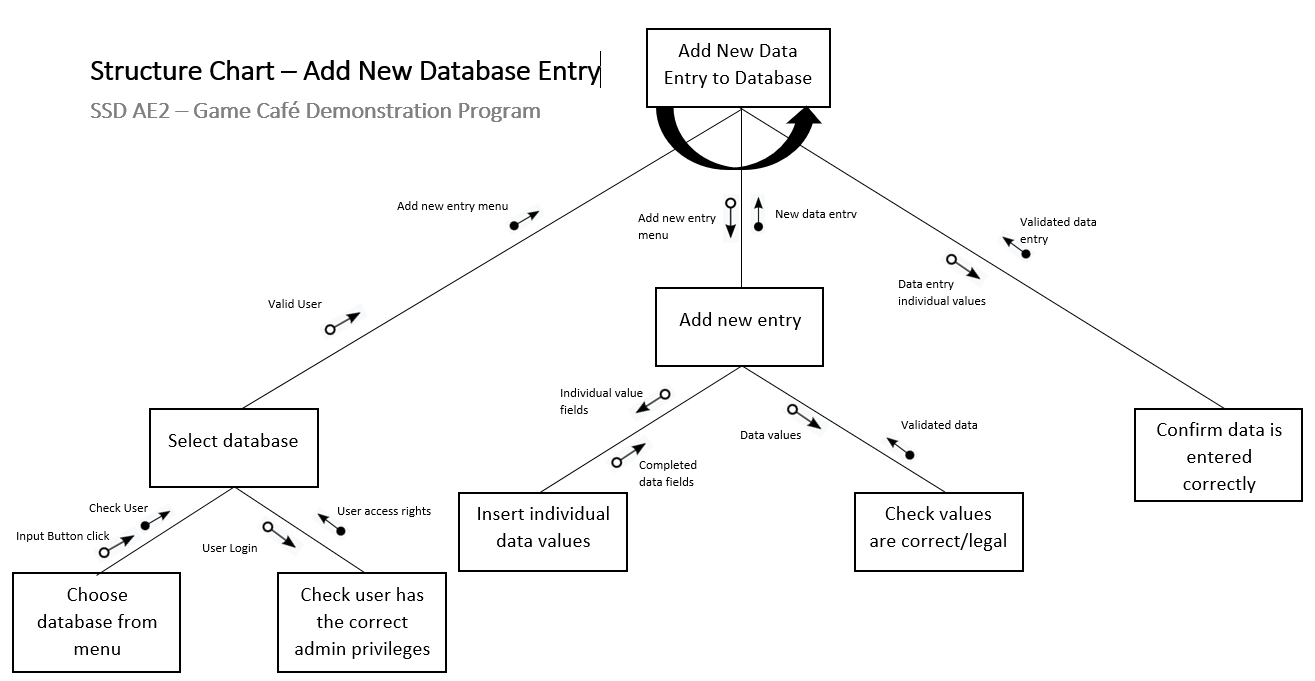
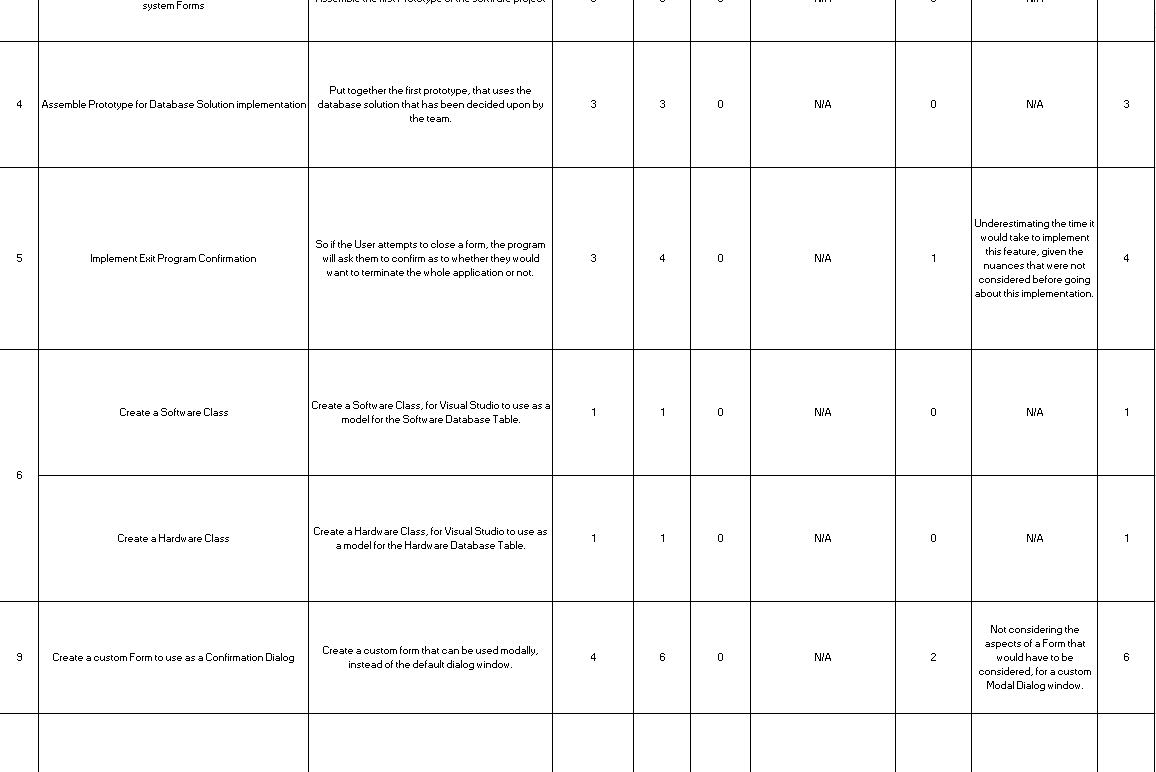
Figure 6: Structure Chart for adding new entries to the Game Cafe Database. (Chris Pryor, 2018)

Figure 7: Use-Case Diagram for Game Cafe Staff Members and Members (patrons) of the Game Cafe.

Figure 8: The Basic Class Diagram for the Game Cafe Management System (given the initial set of derived Use-Cases).

Figure 9: My Weekly Time-log for the project (the image flows over two pages).



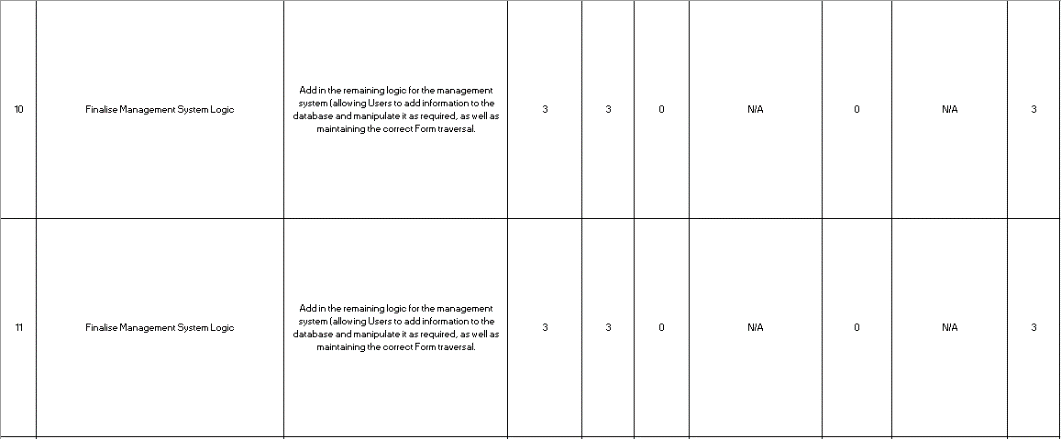


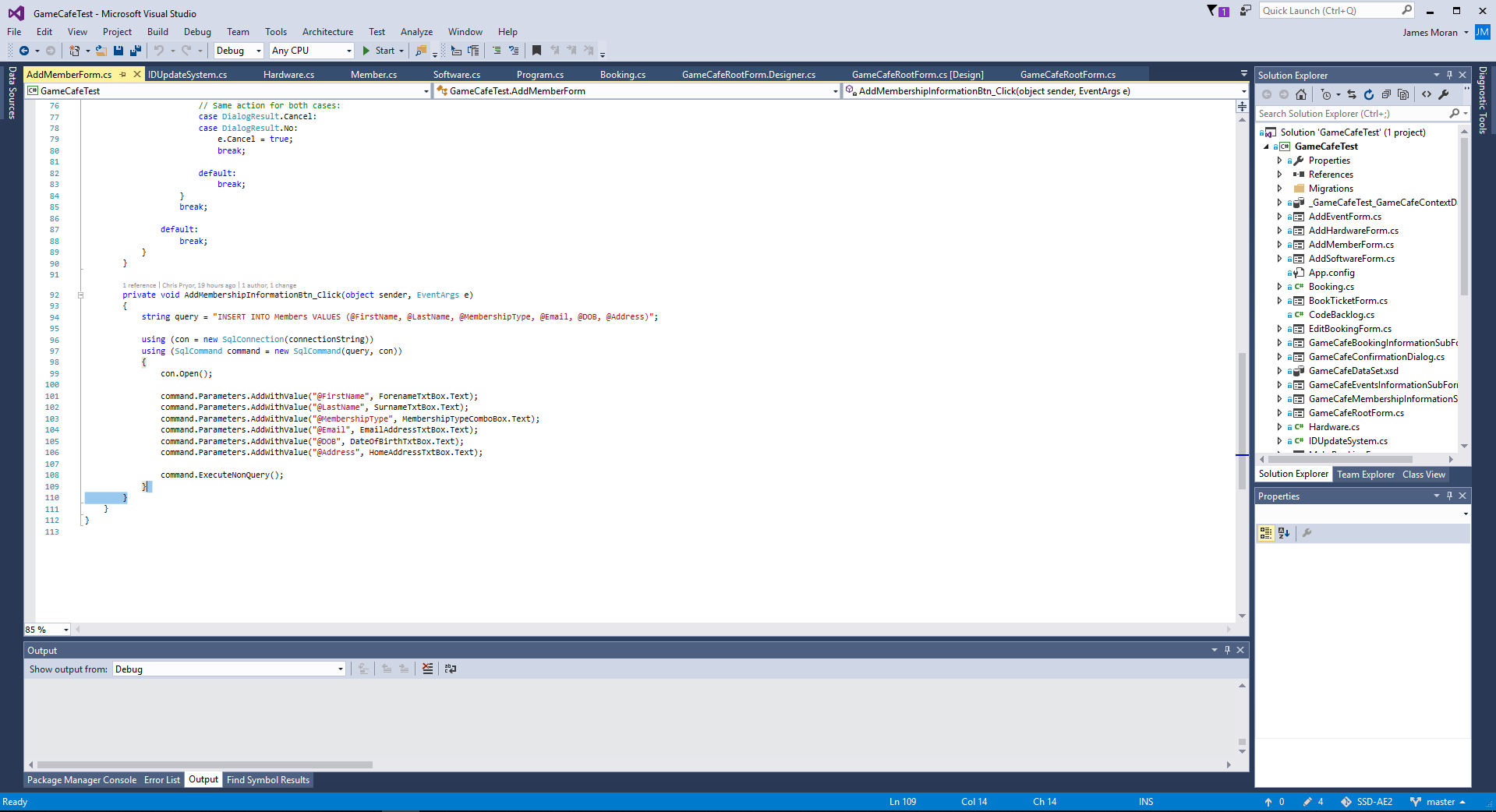
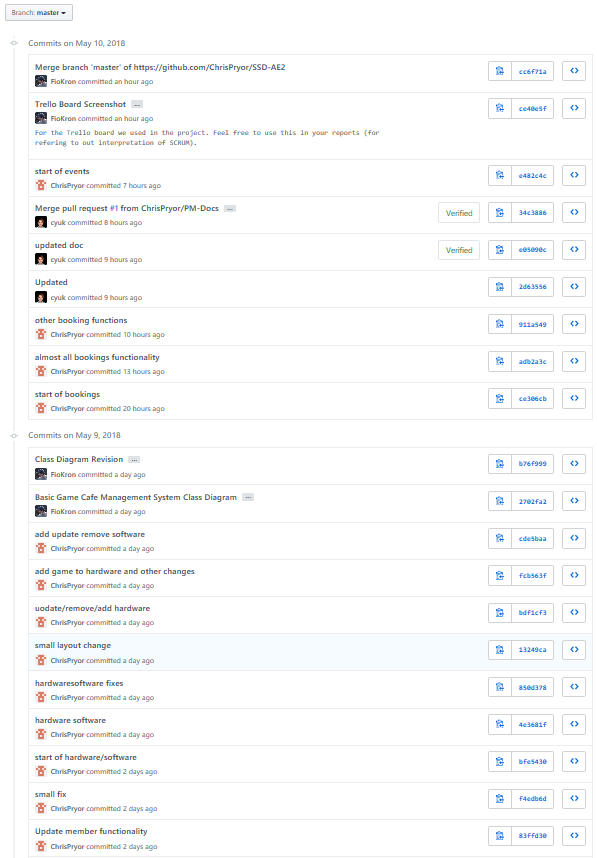
Figure 10: A screenshot of the Visual Studio solution for our project.

Figure 11: The last 3 days of commits, from the 07/05/2018 to the 10/05/2018.



# Appendix E: Tables

[Table 1: The testing table used to perform tests of the Game Cafe Management System. E- 2 -](#_Toc513810438)

Table 1: The testing table used to perform tests of the Game Cafe Management System.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Test Description** | **Expected Test Outcome** | **Actual Test Outcome** | **Test Was Successful** |
| 1 | The User must be able to interact with UI elements using the mouse. | The User can interact with UI elements using the mouse. | The User can interact with UI elements using the mouse. | True |
| 2 | The User must be able to input information into fields, using the keyboard. | The User is able to input information into fields, using the keyboard. | The User is able to input information into fields, using the keyboard. | True |
| 3 | The User interface must be easy to read and use, with a consistent aesthetic style. | The User interface has a consistent aesthetic style, that is also intuitive for use. | The User interface has a consistent aesthetic style, that is also intuitive for use. | True |
| 4 | The User must be able to view database entries, as a list of all entries and individual entries. | The User can view database entries as described. | The user can view database entries as a list of all entries, but not individual entries. | False |
| 5 | The User must be able to search the database by name of data entry. | The User can search the database by name of data entry. | The User is not able to search through the database, by name of data entry. | False |
| 6 | The User must be able to sort the database by each individual data field. | The User can sort the database by each individual data field. | The User can sort the database by each individual data field. | True |
| 7 | The User must be able to add new entries to the database. | The User can add new entries to the database. | The User can add new entries to the database. | True |
| 8 | The User must be able to maintain data entries. | The User can maintain data entries. | The User can maintain data entries. | True |
| 9 | The system must associate Bookings and Ticket purchases with Members for pricing. | The system associates Bookings and Ticket purchases with Members for pricing. | The system would not associate Bookings and Ticket Purchases with Members for pricing. | False |

These results were verified by using the system and one can check on the results, by running the application.

# References

PRYOR C., 2018. *User Mind Map* (Unpublished). Software Systems Development

YOUD C., 2018. *Work Breakdown Structure* (Unpublished). Software Systems Development