James Moran

This report is a reflection of our team’s development process of the Game Café System. This report details the design, development, testing and reflection of the project, among other aspects.

Development Report

Software Systems Development (AE2)

Contents

[List of Tables iii](#_Toc513727461)

[List of Figures iii](#_Toc513727462)

[1. Elicitation of Requirements 0](#_Toc513727463)

[1.1 Work Breakdown Structure (WBS) 1](#_Toc513727464)

[**1.1.1 Sprint Breakdown** 2](#_Toc513727465)

[2. Analysis of Requirements 5](#_Toc513727466)

[2.1 Robustness Diagram 5](#_Toc513727467)

[2.2 User Stories 6](#_Toc513727468)

[2.3 Sequence Diagram 7](#_Toc513727469)

[3. Expression of Requirements 8](#_Toc513727470)

[4. The Use of SCRUM in Our Team 9](#_Toc513727471)

[5. Project Design 10](#_Toc513727472)

[5.1 Structure Chart 10](#_Toc513727473)

[5.2 Use-Case Diagram 10](#_Toc513727474)

[6. Project Development 12](#_Toc513727475)

[Logging my Tasks 12](#_Toc513727476)

[7. Testing the Project 14](#_Toc513727477)

[8. Integrating the Project 15](#_Toc513727478)

[9. Refactoring for the Project 16](#_Toc513727479)

[10. Configuration Management/Version Control 17](#_Toc513727480)

[Bibliography 18](#_Toc513727481)

[References 19](#_Toc513727482)

[Appendix A: Base Project Requirements 20](#_Toc513727483)

[Appendix B: User Stories 21](#_Toc513727484)

[Game Café Staff Member User Stories 21](#_Toc513727485)

[Game Café Member (Patron) User Stories 21](#_Toc513727486)

[Appendix C: Requirements Definition A](#_Toc513727487)

[**Functional Requirements** A](#_Toc513727488)

[**The user must be able to add new entries to a database** A](#_Toc513727489)

[**Non-Functional Requirements** B](#_Toc513727490)

# List of Tables

**No table of figures entries found.**

# List of Figures

[Figure 1: The base Game Cafe Mind Map, based on the requirements noted on the previous page. (Chris Pryor, 2018) 0](#_Toc513727424)

[Figure 2: Game Cafe Management System WBS. 1](#_Toc513727425)

[Figure 3: Game Cafe Robustness Diagram for Staff Members. 5](#_Toc513727426)

[Figure 4: Sequence Diagram for a Staff Member to add information to the management system's database. 7](#_Toc513727427)

[Figure 5: Structure Chart for adding new entries to the Game Cafe Database. (Chris Pryor, 2018) 10](#_Toc513727428)

[Figure 6: Use-Case Diagram for Game Cafe Staff Members and Members (patrons) of the Game Cafe. 10](#_Toc513727429)

[Figure 7: The Basic Class Diagram for the Game Cafe Management System (given the initial set of derived Use-Cases). 11](#_Toc513727430)

[Figure 8: My Weekly Time-log for the project (the image flows over two pages). 12](#_Toc513727431)

# Elicitation of Requirements

For the requirements elicitation (Requirements Gathering), there is the base set of what the User would (most likely), want from the system. This is detailed in Appendix A: Base Project Requirements.

These were obtained from the assignment brief, as this is what the system must have as base requirements.

This was deemed as an appropriate means to get the base set of requirements, as they are noted in the assignment brief and we had no other input to use for the requirements at a base level (such as via surveys, interviews of the client, etc.). Indeed, as interview the client would more than likely produce similar results to what is detailed in the assignment brief, for which aspects of such a management system, the Game Café is most likely to want to organise.

From these base requirements, a Mind Map of what the system must have, can be formed. Our project’s Mind Map Is shown below:

Figure 1: The base Game Cafe Mind Map, based on the requirements noted on the previous page. (Chris Pryor, 2018)

From this, a Work Breakdown Structure (WBS) for the project can be formed, this is detailed on the next page.

## 1.1 Work Breakdown Structure (WBS)

Figure 2: Game Cafe Management System WBS.

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

### **1.1.1 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

#### 1.1.1.1 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’.

#### 1.1.1.2 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)

#### 1.1.1.3 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.

This WBS Diagram and respective Sprint Breakdown detail the tasks that we would want to complete for the project, as well as the order for such (left to right in the WBS Diagram and from Sprint 1 to Sprint 3 for the project’s Sprints).

We deemed this a suitable method for considering all the work we would have to complete and the respective order for completing these tasks, as is laid out above, because each set of tasks for the phases, must be completed before moving onto the next phase (from the WBS Diagram, for each Sprint). For the Sprints, the objectives for that sprint are broken up into multiple tasks, which are then listed on our team’s Trello board, so that we can appropriately assign them to the most suitable team-member.

This would flow well for our team’s mentality, so long as all of the team members put in the respective hours, however, due to other projects that had to be prioritised over this project (as there is a Final Major Project (FMP), that each student must complete), not every team-member was able to put in as many hours as was suitable for this project (8 hours per week).

# Analysis of Requirements

## 2.1 Robustness Diagram

This section begins with the Game Café Staff Member Robustness Diagram, to ensure that the Staff Members of the Game Café are able add information to the database, or make bookings for eSports Events, without having to manually validate the information they add to the system for such:

Figure 3: Game Cafe Robustness Diagram for Staff Members.

This diagram was assembled using Microsoft Visio 2013, after finding Robustness Diagram Symbols for Visio, from the Microsoft online repository of symbols for Visio.

This was deemed to be a suitable method for putting together the Robustness Diagram, as I am familiar with the use of Visio in assembling such diagrams, having produced a Robustness Diagram for Engineering Software Systems (ESS), in the second year. This diagram has all the necessary flow that is expected for this type of diagram, with the correct links to show how the Game Café Staff Member is to manage information that is within the Game Café’s Database.

## 2.2 User Stories

From Figure 3, it is now possible to define the User Stories for a Game Café Staff Member, which then can be used to determine the functional-requirements of the system. These are listed in Appendix B: User Stories.

The Robustness Diagram, along with the project’s Mind Map, were used to form these User Stories, given what we knew about what they would want from the system at the point.

The Robustness Diagram was used here, to ensure the User Stories had a required level of feasibility to them and that the features the User wanted from the system, were being met.

## 2.3 Sequence Diagram

This is for a Staff Member of the Game Café, adding information to the system’s database.

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

# Expression of Requirements

After the elicitation and analysis of the requirements, it is now possible to clearly define our interpretation of the requirements, this is as follows:

# The Use of SCRUM in Our Team

# Project Design

For this stage of the project, various diagrams were used to guide the design of the system.

## 5.1 Structure Chart

Starting with this diagram, for how a Game Café Staff Member adds a Database Entry:

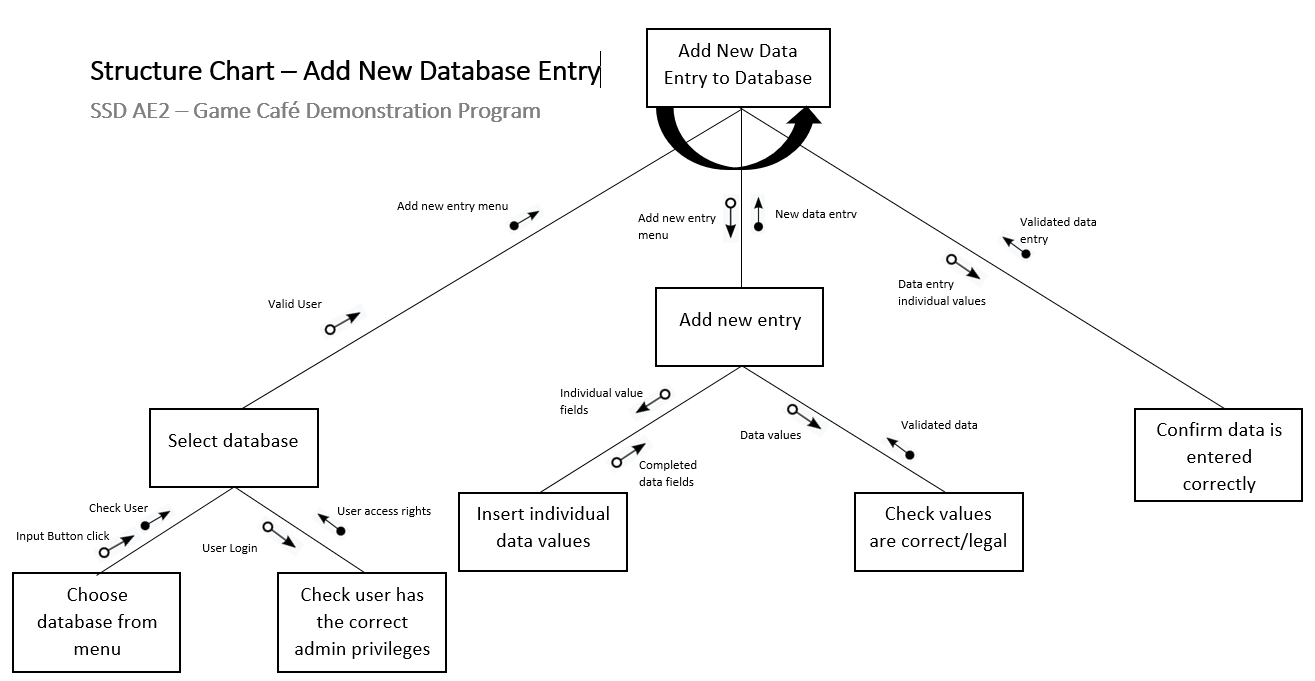


Figure 5: Structure Chart for adding new entries to the Game Cafe Database. (Chris Pryor, 2018)

## 5.2 Use-Case Diagram

This is followed by this diagram:

Figure 6: Use-Case Diagram for Game Cafe Staff Members and Members (patrons) of the Game Cafe.

From the Use-Case Diagram on the previous page, it is now possible to derive a Class Diagram, for the basic structure of the application (to perform these initial Use-Cases):

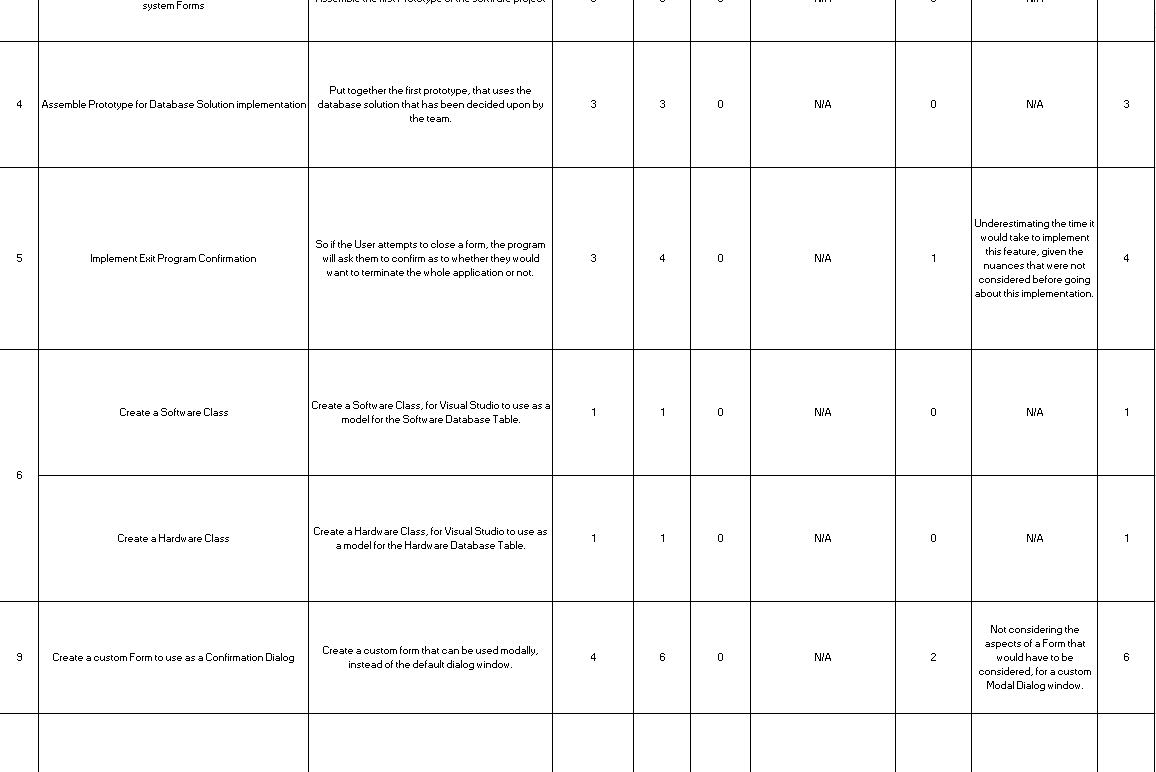
Figure 7: The Basic Class Diagram for the Game Cafe Management System (given the initial set of derived Use-Cases).

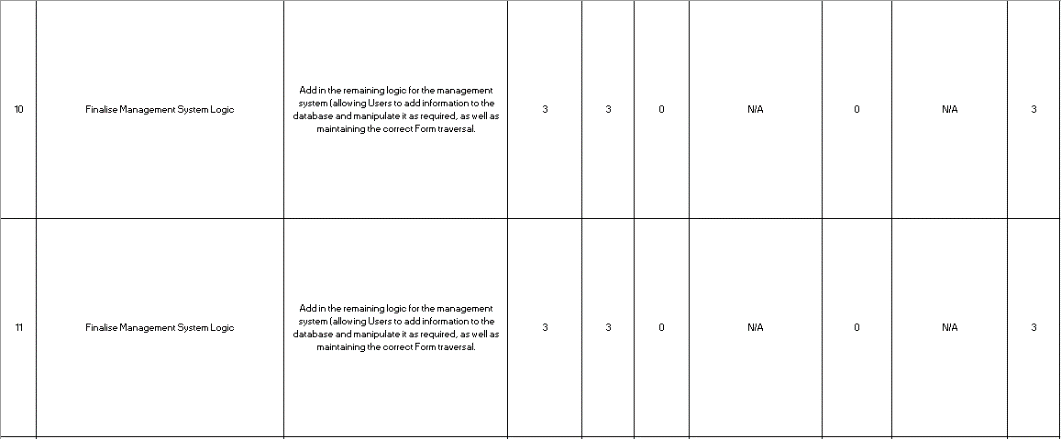
# Project Development

## Logging my Tasks

To keep a log of the hours I have put into the project, I kept a project tracking log, with tasks, their descriptions, the estimated hours for that task, the hours expended, reasons for why there were less hours expended than expected (if that is the case for a particular task), any overtime hours and the reasons for why overtime hours were expended, if overtime was put in:

Figure 8: My Weekly Time-log for the project (the image flows over two pages).





# Testing the Project

# Integrating the Project

# Refactoring for the Project

# Configuration Management/Version Control

# Bibliography

MICROSOFT, © 2018a Microsoft. *Entity Framework Code First to a New Database* [Viewed on the 09/05/2018]. Available from: <https://msdn.microsoft.com/en-gb/library/jj193542(v=vs.113).aspx>

MICROSOFT, © 2018b Microsoft. *Entity Framework Code First to an Existing Database* [Viewed on the 09/05/2018]. Available from: <https://msdn.microsoft.com/en-gb/library/jj200620(v=vs.113).aspx>

# References

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

# 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: Requirements 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.