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 iv](#_Toc513640652)

[List of Figures iv](#_Toc513640653)

[1. Elicitation of Requirements 0](#_Toc513640654)

[1.1 Work Breakdown Structure (WBS) 1](#_Toc513640655)

[**1.1.1 Sprint Breakdown** 2](#_Toc513640656)

[2. Analysis of Requirements 5](#_Toc513640657)

[2.1 Robustness Diagram 5](#_Toc513640658)

[2.2 User Stories 6](#_Toc513640659)

[2.3 Sequence Diagram 7](#_Toc513640660)

[3. Expression of Requirements 8](#_Toc513640661)

[**3.1 Functional Requirements** 8](#_Toc513640662)

[3.1.1 The user must be able to interact with UI elements with the mouse. 8](#_Toc513640663)

[3.1.2 The user must be able to input information using the keyboard. 8](#_Toc513640664)

[3.1.3 The user interface must be easy to read and use, with a consistent aesthetic style. 8](#_Toc513640665)

[3.1.4 The user must be able to view database entries, as a list of all entries and individual entries. 8](#_Toc513640666)

[3.1.5 The user must be able to search a database by name of data entry. 8](#_Toc513640667)

[3.1.6 The user must be able to sort a database by each individual data field. 8](#_Toc513640668)

[3.1.7 The user must be able to add new entries to a database. 8](#_Toc513640669)

[3.1.8 The user must be able to maintain data entries. 8](#_Toc513640670)

[3.1.9 The system must associate bookings and ticket purchases with members for pricing. 8](#_Toc513640671)

[**3.2 Non-Functional Requirements** 9](#_Toc513640672)

[3.2.1 The system must give separate user privileges to a standard user (Café employee) and an Administrator 9](#_Toc513640673)

[3.2.2 The response time when a UI button/element is pressed should be no longer than 0.5 seconds. 9](#_Toc513640674)

[3.2.3 The program should run on Windows 7 and above. 9](#_Toc513640675)

[3.2.4 Databases should be encrypted so that data cannot be stolen. 9](#_Toc513640676)

[3.2.5 Sensitive data should be hidden to users. 9](#_Toc513640677)

[4. The Use of SCRUM in Our Team 10](#_Toc513640678)

[5. Project Design 11](#_Toc513640679)

[5.1 Structure Chart 11](#_Toc513640680)

[5.2 Use-Case Diagram 11](#_Toc513640681)

[6. Project Development 12](#_Toc513640682)

[7. Testing the Project 13](#_Toc513640683)

[8. Integrating the Project 14](#_Toc513640684)

[9. Refactoring for the Project 15](#_Toc513640685)

[10. Configuration Management/Version Control 16](#_Toc513640686)

[Bibliography 17](#_Toc513640687)

[References 18](#_Toc513640688)

# 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](#_Toc513640695)

[Figure 2: Game Cafe Management System WBS. 1](#_Toc513640696)

[Figure 3: Game Cafe Robustness Diagram for Staff Members. 5](#_Toc513640697)

[Figure 4: Sequence Diagram for a Staff Member to add information to the management system's database. 7](#_Toc513640698)

[Figure 5: Structure Chart for adding new entries to the Game Cafe Database. (Chris Pryor, 2018) 11](#_Toc513640699)

[Figure 6: Use-Case Diagram for Game Cafe Staff Members and Members (patrons) of the Game Cafe. 11](#_Toc513640700)

# 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:

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.

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’.

# 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.

## 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:

* 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

There are also the Game Café Member’s User Stories to consider:

* 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

## 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:

## **3.1 Functional Requirements**

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

### 3.1.1 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.

### 3.1.2 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.

### 3.1.3 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.

### 3.1.4 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.

### 3.1.5 The user must be able to search a database by name of data entry.

This allows the user to find specific data when required.

### 3.1.6 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.

### 3.1.7 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.

### 3.1.8 The user must be able to maintain data entries.

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

### 3.1.9 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.

## **3.2 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.

### 3.2.1 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.

### 3.2.2 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.

### 3.2.3 The program should run on Windows 7 and above.

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

### 3.2.4 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.

### 3.2.5 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.

# 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:

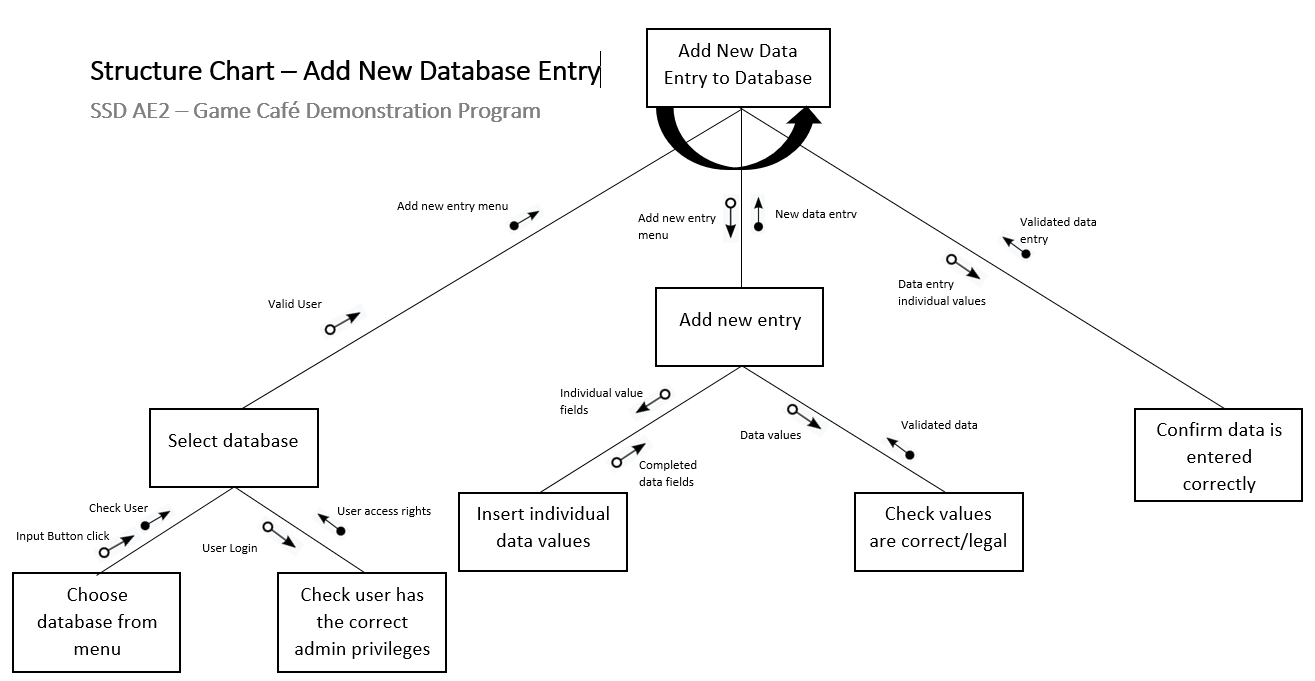


Figure 5: Structure Chart for adding new entries to the Game Cafe Database. (Chris Pryor, 2018)

## 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).

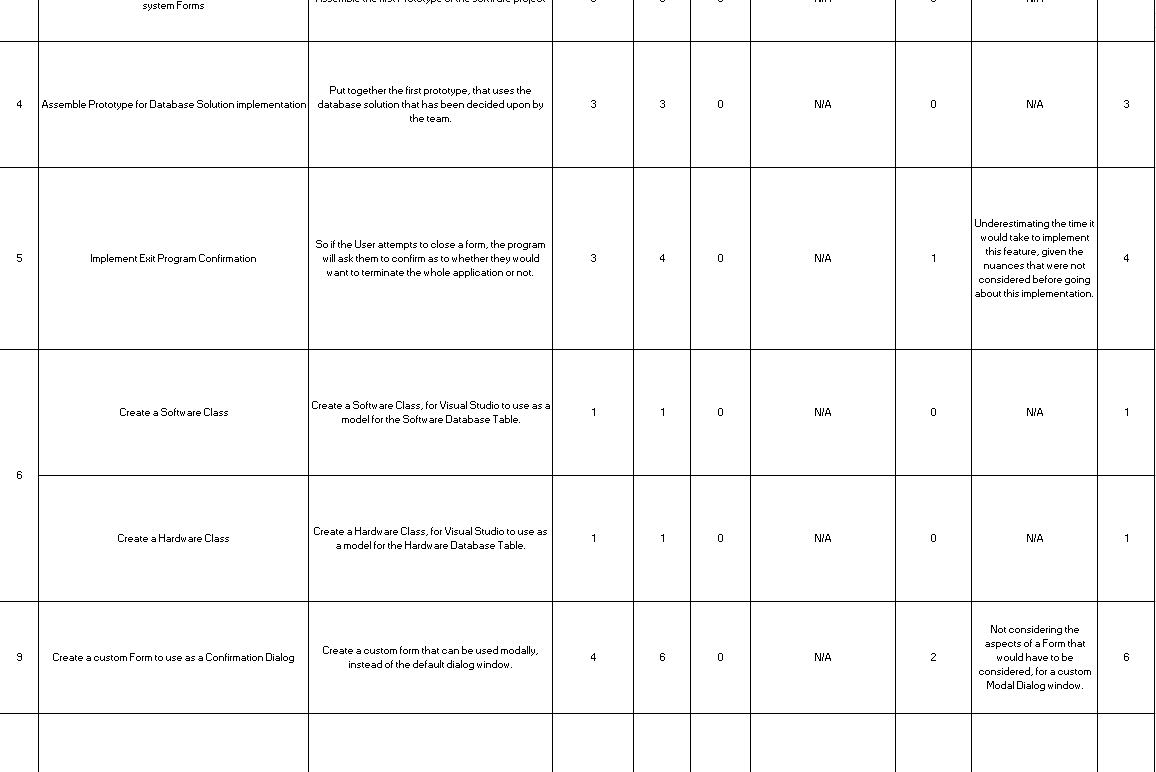
# 

# Project Development

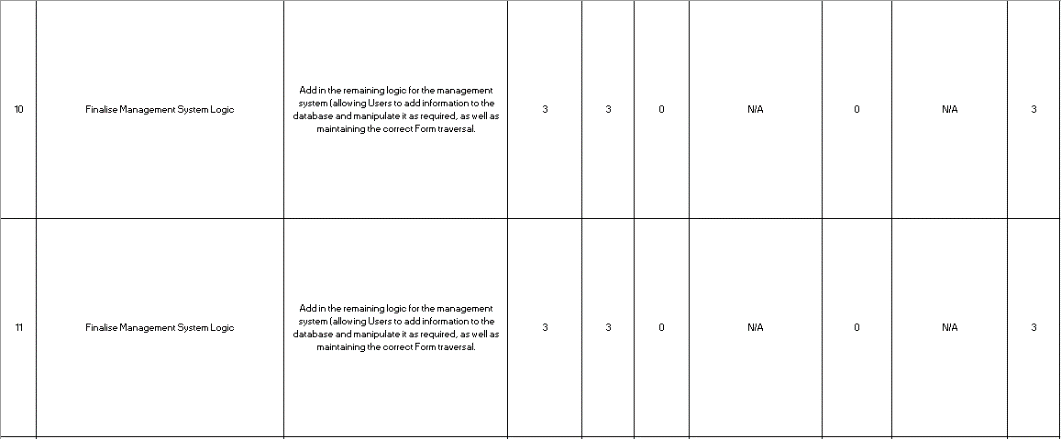
## Logging my Tasks

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