**Object Orientated Software Engineering Project 2017.**

**Authors:** Catherine Lane - x17118832 & Martina Sardo Cardalano - x17149676

**Date:** 30/10/2017

**Project Supervisors:** Padraig De Burca, Lecturer in OOSE.

**CONTENTS**

**Introduction & Objectives** ………………………………………………………………………………. Page 2

**Glossary**.……………………………………………………………………………………………………….. Page 3 -5

**Actors** …………………………………………………………………………………………………………... Page 6

**Use Case Model**

Diagram 1.1……………………………………………………………………………………….. Page 6

**Use Case Description**

Table 1.2 …………………………………………………………………………………………... Page 7 & 8

Table 1.3 …………………………………………………………………………………………… Page 9 & 10

**Project Plan**

Objectives and Monitoring …………………………………………………………………. Page 11

WBS Breakdown Schedule ………………………………………………………………… Page 12 & 13

Gantt Chart ……………………………………………………………………………………….. Page 14

**System Sequence Diagrams**

Diagram 1.6 ………………………………………………………………………………………. Page 15

Diagram 1.7 ………………………………………………………………………………………. Page 16

**Contracts**

Table 2.1 and 2.2 ……………………………………………………………………………….. Page 17

Table 2.3 and 2.4 ……………………………………………………………………………….. Page 18 & 19

**Communication Diagrams**

Diagram 2.1 and 2.2 …………………………………………………………………………... Page 20

Diagram 2.3 and 2.4 …………………………………………………………………………... Page 21

**Testing Plan** …………………………………………………………………………………………………. Page 22

**Title:** OOSE Minecraft Project - Christmas Adventure.

**Project Objectives:** The problem is to design a cloud based game for children age 5+. The trigger for the problem is based on Minecraft. From a technology viewpoint the system should be compatible with PCs & mobile devices and deployed via the Cloud. Anything novel to do with Minecraft software will be acceptable for this project.

**Project Deliverables:**

1. Identify the Actors.
2. Create a Use Case Model.
3. Describe, in detail, a Use Case each.
4. Create a Conceptual Class diagram.
5. Create a Project plan to include; WBS, Gantt Chart, Monitoring description.
6. Draw a System sequence diagram each.
7. Complete two Contracts based on each sequence diagram.
8. Draw a Communication diagram based on said contracts.
9. Develop a testing plan.
10. Ensure use of Rational Rose.
11. Review Presentation.

**Project Design:** The project is designed to allow the gamer to participate in the Christmas Adventure 2017 which can be downloaded from the Minecraft.net as an add-on. The gamer will receive various rewards based on their subscription on completion of the event. The event will be available to downloaded for a limited time.

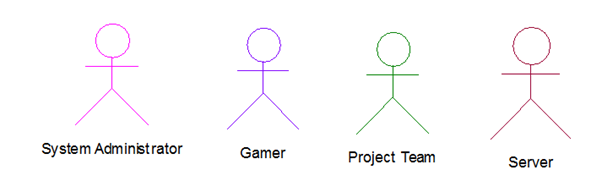
**Glossary:**

|  |  |  |
| --- | --- | --- |
| **Term** | **Type** | **Explanation** |
| Actor | Actor | A system or person that derives benefit and is external to the system. |
| AdminID | Attribute | Identifies each System Administrator. |
| Adventure add-on | Class | The adventure class defines the event the gamer participates in, which is downloaded as an extension by the user. |
| Age | Attribute | User inputs their age. Age is restricted under the terms of the game, must be 5+. |
| assembleParts() | Method | Assembles items together to form an object. |
| authenticates() | Method | The system admin ensures the user associated with GamerID is the correct user through stored password. |
| blueprintID | Attribute | Identifies each blueprint. |
| buildsObjects() | Method | The gamer uses material to build objects. |
| choosesMode() | Method | The gamer selects from the following modes; easy, medium, hard; that they want to compete in. |
| Class Diagram | Diagram | A diagram that describes the structure of the system by showing system classes, their attributes, methods (operations) and the relationships between objects. |
| collectsItems() | Method | Involves gathering material required to build objects. |
| createsAccount() | Method | The user creates the Gamer account and is administered a GamerID. |
| deactivatesAccount() | Method | The user deactivates their account from the game. |
| decorationName | Attribute | The identifying name given to each decoration. |
| deletesAccount() | Method | The admin deletes the account from the database. |
| Description | Attribute | Offers a description of each blueprint in terms of items and layout. |
| difficultyID | Attribute | Identifier given to each decoration to define which difficulty mode it belongs to. |
| displaysInterface() | Method | An interface is displayed on the screen to the gamer. |
| downloadsAdventure() | Method | The user downloads the add-on extension from the Minecraft website. |
| Easy mode | Class | On option to be selected by the gamer differs in blueprintID from other modes. Decorations are easier to build but reward less valuable items. |
| exchangeItems() | Method | The gamer selects to transfer his awarded items to another Minecraft World of choice. |
| Gamer | Actor | Participates in the event and reaps rewards. |
| getsRewards() | Method | The gamer receives item rewards on completion of the event. |
| Hard mode | Class | An option to be selected by the gamer differs in blueprintID from other modes. Decorations are hard to build and reward the most valuable items. |
| ItemDescription | Attribute | Gives name to an item. |
| ItemID | Attribute | Identifies one item from the other. |
| listItems() | Method | Displays a list of items required to build an object. |
| Medium mode | Class | An option to be selected by the gamer differs in blueprintID from other modes. Decorations are of medium difficulty to build and reward valuable items. |
| Minecraft Server | Actor | Stores Minecraft Worlds. |
| modeID | Attribute | Uniquely identifies the three modes, used in coding and database. |
| Password | Attribute | User’s unique identifier which authenticates the user. |
| Postbox | Class | An object created so that the Gamer can interact with and transfer reward items to another Minecraft World. |
| quantity | Attribute | Quantity of items held in inventory. |
| qtyReceived | Attribute | Represents the amount of items received. |
| requiredItems | Attribute | The amount and type of items required for the decoration build. |
| retreiveBlueprint() | Method |  |
| saveScore() | Method | Involves saving the progress of the gamer up to a certain point in time, which is available to access again before the limited window ends. |
| showAssembledParts() | Method | Displays an interface of the finished look of the object built by assembling a required list of items. |
| System Administrator | Actor | Performs admin duties to the extension, controls the database and interacts with the user indirectly. |
|  |  |  |
| System Sequence Diagram | Diagram | A diagram that shows a scenario of a particular use case, the events the actors generate, their order and possible inter-system events. |
| transferID | Attribute | Uniquely identifies the transfer transaction. |
| transfersItems() | Method | Involves the movement of rewarded items from the add-on to a world of gamers choice carried out by System admin. |
| updatesInventory() | Method | The admin removes the rewarded items from the gamers inventory on the add-on and adds it to the inventory of the selected world. |
| useBlueprint1() | Method | Blueprint1 contains the instructions to build a Stocking, Santa Figure and Santa Sleigh. |
| useBlueprint2() | Method | Blueprint2 contains the instructions to build a Stocking, Santa Figure and Snowman. |
| useBlueprint3() | Method | Blueprint3 contains the instructions to build a Stocking, a Snowman and a Christmas Tree. |
| useItem() | Method | Selects item(s) to use in the building of an object. |
| User | Actor | Stems from the Gamer actor. The gamer is the user. The user creates the Gamer ID and downloads the extension. |
| UserName | Attribute | A unique name chosen by the user in order to register their account. |
| UserID | Attribute | An integer representing the User, used in the database to identify users from each other. |
| Use Case Model | Diagram | A diagram representing the external actors and events performed by them. |
| Use Case Description | Use Case | A detailed description of an emerging use case from the main Use case model describing the flow events, conditions and specifications. |

Table 1.1

**1) Actors:**

* Gamer. (Secondary - User)
* System Administrator.
* Project team.
* Minecraft server.



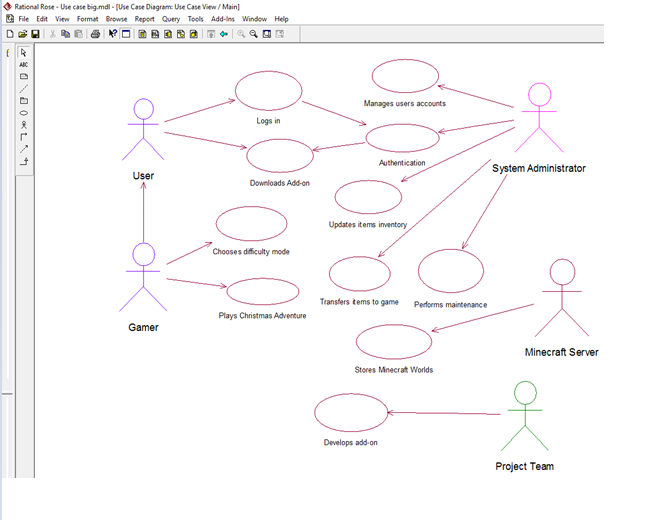


Diagram 1.1

**3) Use Case Descriptions**

(1)

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | | Build Christmas Decoration | |
| Created By: | | Martina Sardo Cardalano | Date created: 31 October 2017 |
| Primary Actor: | | Gamer | Secondary Actors: User, Christmas Adventure add-on, Project Team |
| Scope: | | Minecraft Christmas Adventure Project | |
| Description: | | The Christmas Adventure 2017 add-on allows gamers to play for a limited amount of time. The user is asked to choose either easy, medium or hard mode which involves building three of five decorations displayed on the interface, in no specific order. Each decoration built rewards an item which is needed to build the next decoration. The Gamer gets rewards at completion of each decoration. The rewards will be saved in the user account. | |
| Flow Description: | | | |
|  | Pre-Conditions: | User must have an Internet Connection and an active Minecraft account. Must have sufficient hard drive space on the device chosen. He must be 5 or older. | |
|  | Activation: | User logs into his account and downloads add-on. | |
|  | Main Flow: | 1. The User goes on Minecraft webpage 2. User logs into his account 3. The system verifies if User has an active account 4. Account is active 5. User downloads Christmas Adventure. 6. System displays three options, easy, medium and hard mode 7. System asks to pick a difficulty mode 8. Gamer selects mode 9. Game starts at spawn 10. System displays an interactive window 11. Gamer clicks on window and reads instructions 12. Gamer collects items as instructed 13. Gamer adds items to his inventory 14. Gamer clicks on “assemble decoration” 15. Gamer has collected all the items needed, decoration appears on the screen. 16. System displays NPCs. 17. Gamer gets rewards. 18. Gamer can add rewards to inventory | |
|  | Alternative Flow: | 1. Gamer doesn’t want to finish the game. 2. Gamer takes too long to complete the adventure. 3. Gamer wants to build a decoration but he hasn’t collected all the items yet 4. Gamer has all the items to build a decoration, but selects wrong blueprint 5. Gamer wants to change level | |
|  | Exceptional Flow: | 1. System is down 2. System is affected by a bug 3. User is younger than 5 | |
|  | Termination: | The Adventure terminates when the Gamer builds the last decoration. Gamer is given the option to save rewards as inventory. | |
|  | Post-Conditions: | Gamer completes the Adventure, he is able to save the game and rewards are added to main inventory. The adventure is no longer downloadable. | |
| Views: | | Refer to Use Case Model, Diagram 1.1 | |
| Special Requirements: | | n/a | |
| Interfaces: | | The adventure presents multiple interfaces, that the Gamer can click to get instructions and view the decorations blueprints. | |
| System Characteristics/Performance: | | The game can be uploaded on desktop and mobile. Based on cloud technology. Requires Internet connection. | |
| Implementation Requirements: | | Can only be downloaded for a limited amount of time from the Minecraft website. User must have a register account prior to download the add-on | |
| Technical Specifications: | | n/a | |

Table 1.2

(2)

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | | Transferring rewards to other Minecraft Worlds. | |
| Created By: | | Catherine Lane | Date created: 29th October 2017 |
| Primary Actor: | | System Administrator | Secondary Actors: Gamer, Christmas Adventure add-on, Minecraft Server |
| Scope: | | Minecraft Christmas Adventure Project | |
| Description: | | The gamer participates in the Christmas adventure event and upon completion of the event, receives rewards based on the difficulty they partook in. To validate these items, there will be an option to transfer these rewarded items to a world of the gamer’s choice. This interface will be accessible from the Christmas postbox located at the corner of the event. The gamer will be prompted by an interface, asking the user to select a world to transfer the items to. This is a one-time only transaction. The user will confirm, and the items will be sent via system administrator to their new inventory. | |
| Flow Description: | | | |
|  | Pre-Conditions: | Must have downloaded the add-on. Must download the add-on within the limited time. Must have a valid minecraft account. Must meet the materials criteria. Must have designated hard drive space for the add-on. Must have a viable internet connection. | |
|  | Activation: | The gamer will find the Christmas Postbox in the left-hand corner of the event. Upon clicking on the postbox, the gamer will be prompted with a interface displaying a choice of Minecraft world’s to transfer rewarded items to. | |
|  | Main Flow: | 1. Gamer receives rewards from completing the Christmas Adventure. 2. Gamer walks to the Christmas postbox. Clicking on it triggers an interface displaying the world’s on offer, of which the gamer can transfer their newly awarded items to. 3. The player chooses one world and confirms selection. 4. The system administrator withdraws the rewarded item’s from the gamer’s inventory in the Christmas adventure and inputs them into the corresponding database in the selected world, stored on the Minecraft server. 5. When the user logs into the selected world, the gamer’s rewarded items from the add-on will be displayed in their new inventory. | |
|  | Alternative Flow: | + The gamer does not want to transfer their items.  + The gamer deactivates their account and loses awarded items.  + Gamer doesn’t trade the items within the limited window. | |
|  | Exceptional Flow: | 1. The gamer was affected by a bug in the system. 2. The server is down. | |
|  | Termination: | The transfer is terminated when the gamer logs in and receives their reward in their destined world. | |
|  | Post-Conditions: | The awarded items can be used as standard items in the world. | |
| Views: | | Refer to Use Case Model, Diagram 1.1 | |
| Special Requirements: | | The user must have access to another world. | |
| Interfaces: | | An interface will be displayed upon clicking on the postbox. The interface will contain a drop-down menu of the gamer’s recorded world’s which one can be selected from, to transfer awarded items to. | |
| System Characteristics/Performance: | |  | |
| Implementation Requirements: | | Must have completed the Christmas Adventure add-on event and taken the awarded items. | |
| Technical Specifications: | |  | |

Table 1.3

**4) Conceptual Class Diagram**

A detailed diagram of the relationship between each of the classes, listed attributes and operations.

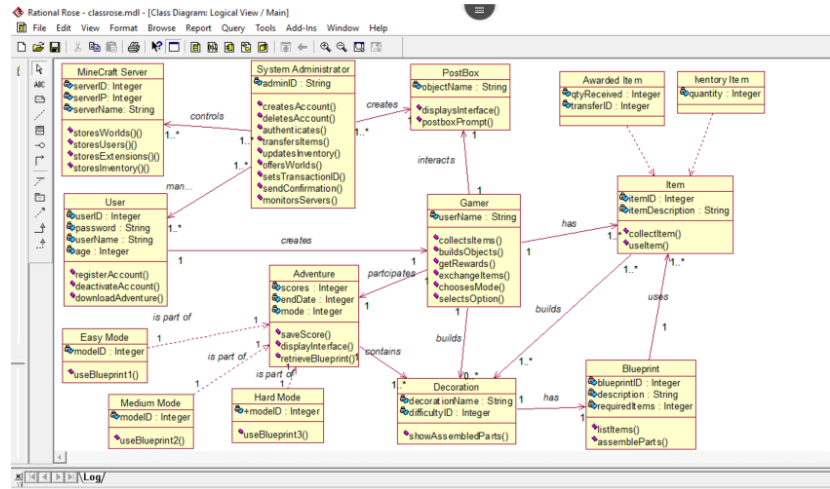
****

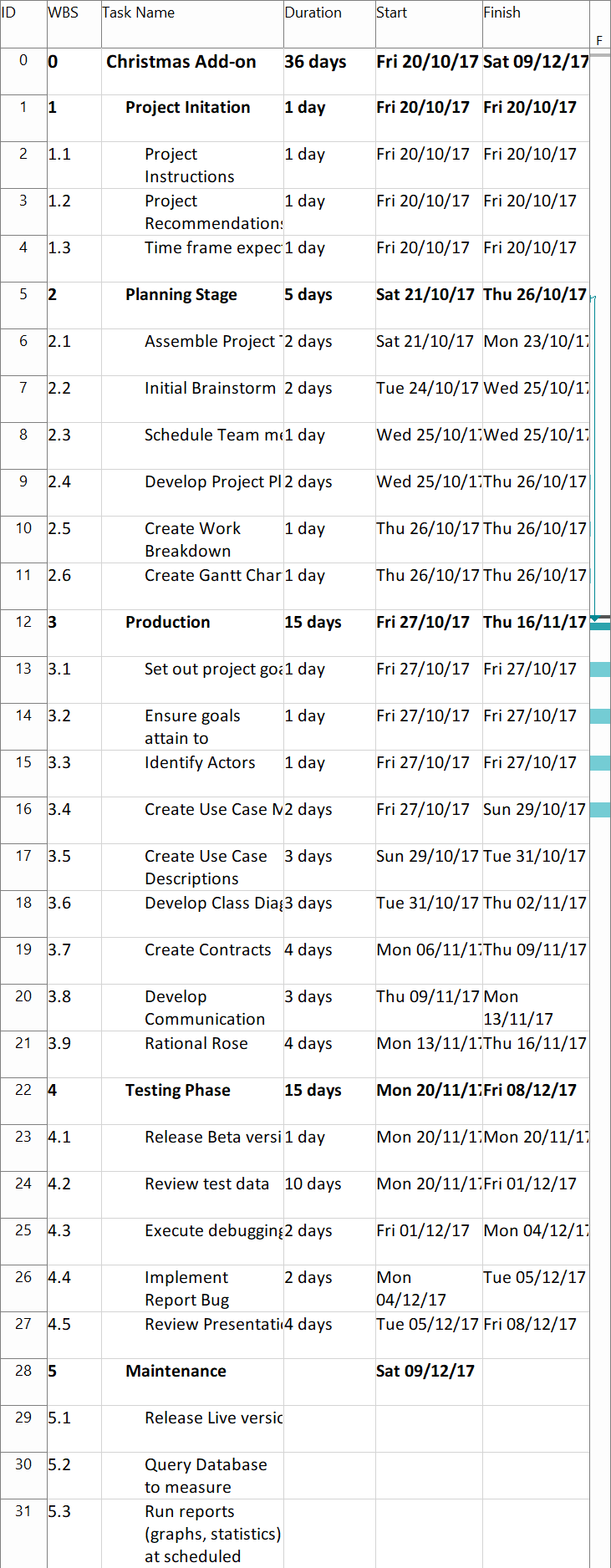
Diagram 1.2

**5) Project Plan**

Featuring the WBS Schedule and Gantt Chart.

We monitored the project plan on Friday evenings weekly, ensuring our goals aligned with the project plan timeline. A fallback occurred in the Production Phase part 3.9, building Rational Rose due to upcoming exams. This was made up for over the allocated free time at the weekend which set the project back on track the following Monday 20/11/2017, no amendment to the schedule was required. The remaining deadlines were met as planned and discussed.

**WBS Breakdown Schedule**

Diagram 1.3

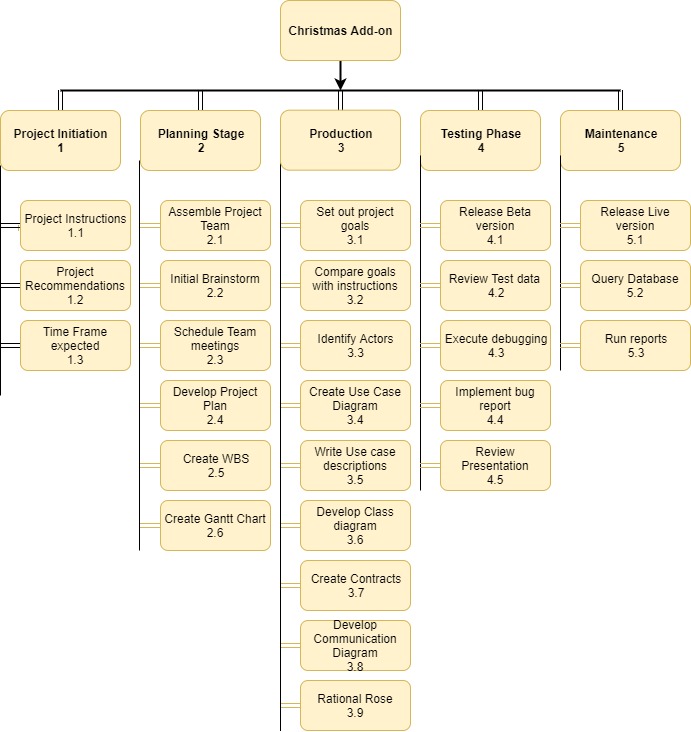


Diagram 1.4

**Project Planning- Gantt Chart.**

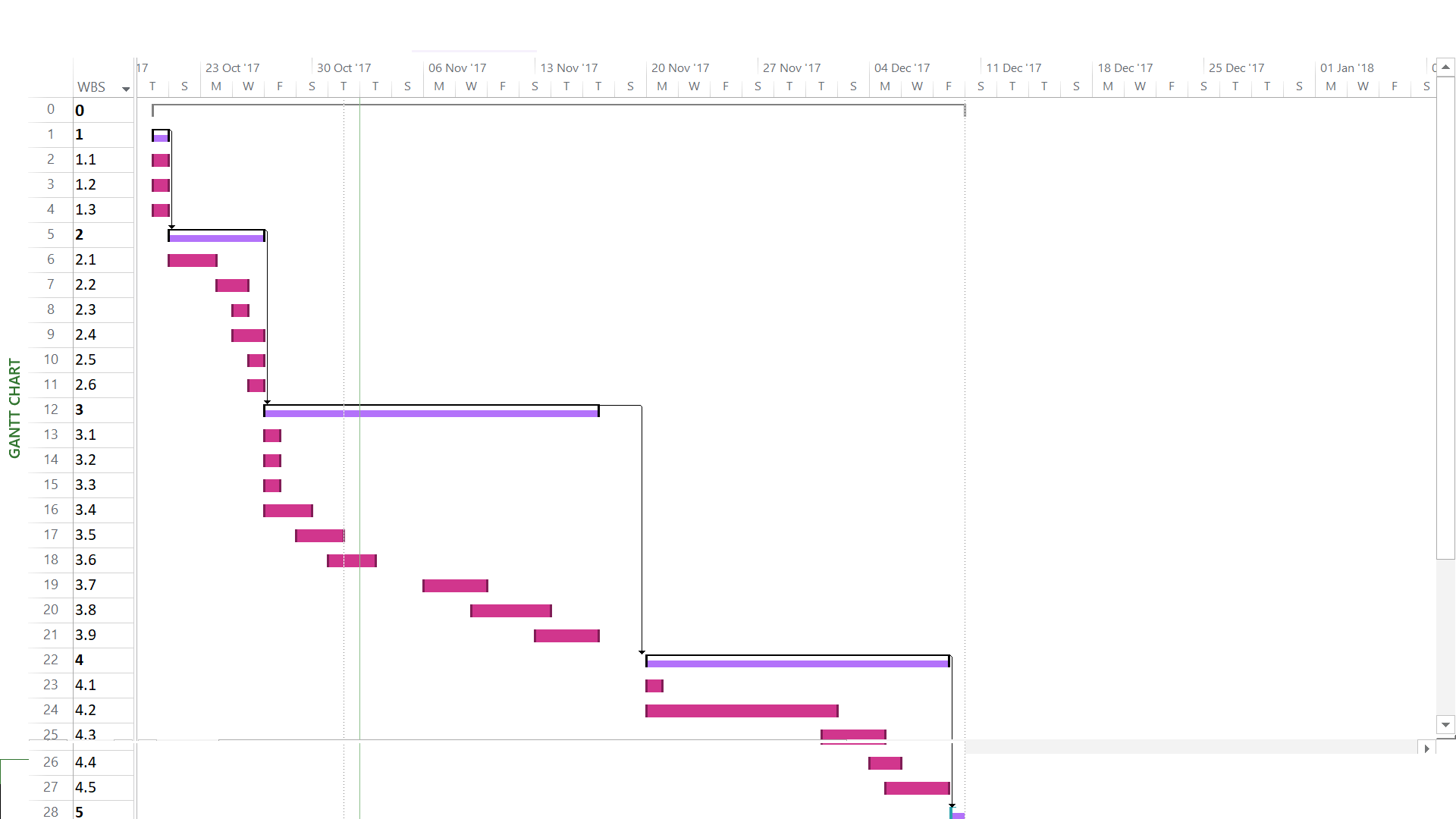


Diagram 1.5

**System Sequence Diagrams**

1. **Cathy Lane x17118832 -**Transferring rewards to other Minecraft Worlds.

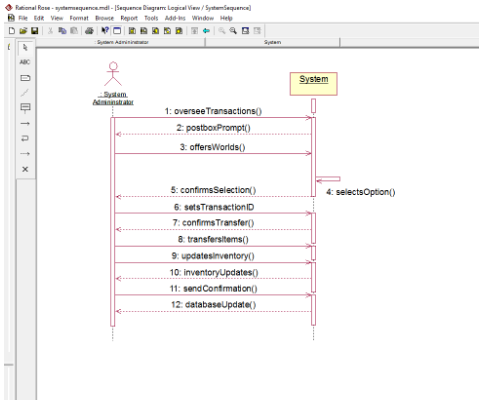


Diagram 1.6

**(2) Author:** Martina Sardo Cardalano x17140676-Play Adventure

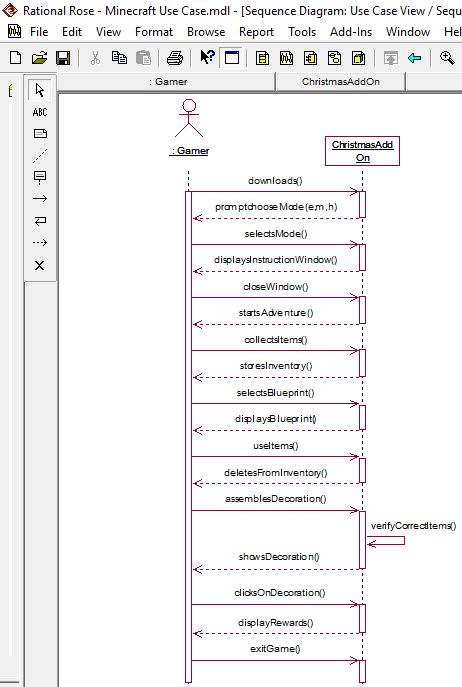


Diagram 1.7

**Contracts:**

based on Use Case Description - Transferring rewards to other Minecraft Worlds.

**Author:** Cathy Lane x17118832

(1)

|  |  |
| --- | --- |
| **Name:** | offersWorlds() |
| **Responsibilities:** | The postbox will display a drop-down menu of Worlds that can be selected for transfer of items. The System admin oversees the worlds available to transfer rewarded items to. |
| **Exceptions:** | If the user attempts to select an invalid World, they will be prompted with an error. |
| **Type:** | System |
| **Pre-Conditions:** | The user has a valid world to transfer items to. |
| **Post-Conditions:** | The user has the ability to transfer items to the selected world. |

Table 2.1

(2)

|  |  |
| --- | --- |
| **Name:** | transfersItems() |
| **Responsibilities:** | The system admin oversees the transfer of rewarded items from the add-on extension to the user’s selected world. |
| **Exceptions:** | The system will prompt the user with an error if they attempt to trade items that are not rewarded from the event. |
| **Type:** | System |
| **Pre-Conditions:** | The user collected their rewards from the Christmas add-on event. |
| **Post-Conditions:** | The users items are transferred to their world of choice. |

Table 2.2

**Contracts:** based on Use Case Description – Build Christmas Decoration

**Author:** Martina Sardo Cardalano x17140676

(1)

|  |  |
| --- | --- |
| **Name:** | displaysDecoration () |
| **Responsibilities:** | The system checks if all the items have been collected by the gamer and displays a decoration |
| **Exceptions:** | The gamer tries to assemble a decoration without having collected all the items required. System returns an error message |
| **Type:** | System |
| **Pre-Conditions:** | The gamer has collected all the items needed to build a decoration |
| **Post-Conditions:** | System displays decoration which the gamer can click on to collect rewards. |

Table 2.3

(3)

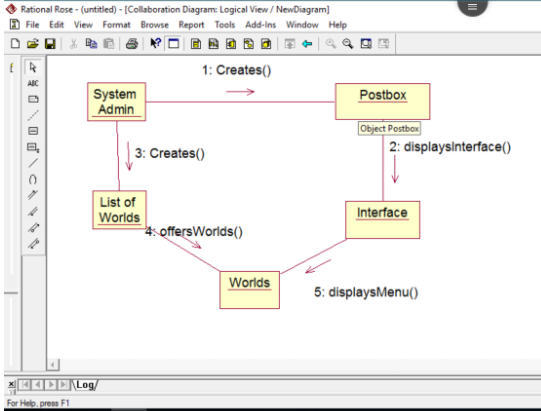
|  |  |
| --- | --- |
| **Name:** | storesInventory() |
| **Responsibilities:** | The System allows the gamer to store the items he finds while exploring the Adventure |
| **Exceptions:** | Gamer doesn’t double click on item. System opens a “hint” window which displays: “transfer to inventory by double clicking on item” |
| **Type:** | System |
| **Pre-Conditions:** | The Gamer has opened the item description |
| **Post-Conditions:** | System transfer item to Gamer’s inventory |

Table 2.4

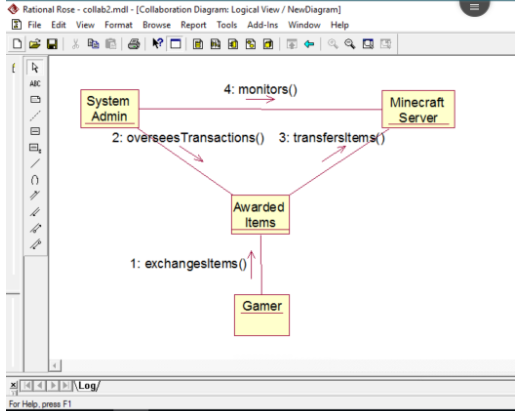
**Communication Diagrams**

**Author:** Cathy Lane x17118832

1. Based on Contract 1, offersWorlds().

 Diagram 2.1

(2) Based on transfersItems() contract.

Diagram 2.2

**Communication Diagrams**- Martina Sardo Cardalano x17149676

(1)

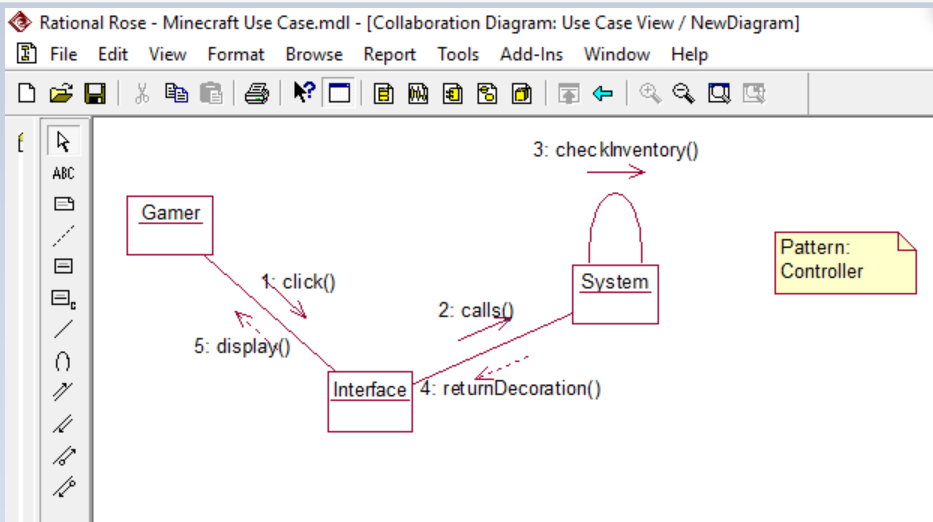


Diagram 2.3

(2)

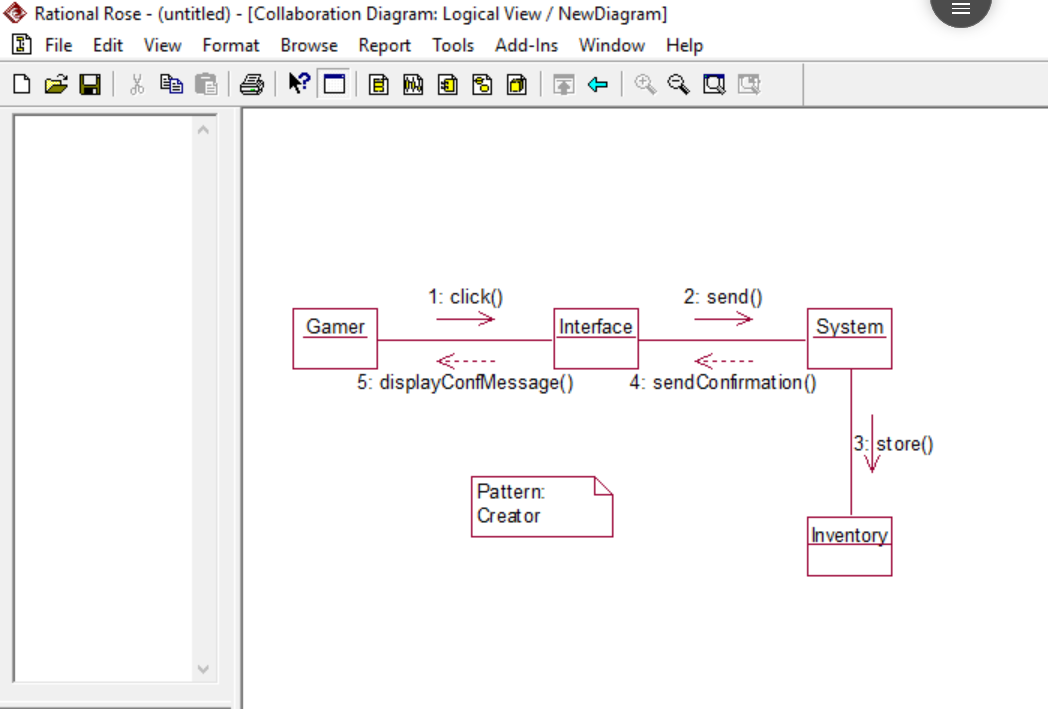


Diagram 2.4

**Testing plan:**

**Test plan objectives:**

To develop a structured plan which focuses on identifying and testing system defaults in order to validate the program.

A number of test cases are created within the testing phase, of which one is shown as example below. These will be stored in a testing file.

**Example of Test Case:**

*Test Case ID:* TC01

*Summary:* Verify that clicking “Download Christmas Adventure”, downloads the add-on

extension to the user’s device.

*Pre-conditions:* 1) User owns a valid Minecraft Account. (2) Has a stable internet connection.

(3) Has suitable hard-drive storage.

*Procedure:* Select the “Download Christmas Adventure” button the the Minecraft Website.

*Expected Result:* The extension downloads to the user’s device under the Downloads folder.

*Actual Result:* If the pre-conditions are met, the extension is downloaded.

*Test Status:* Pass.

*Created by:* Catherine Lane and Martina Sardo Cardalano.

*Executed by:* Catherine Lane and Martina Sardo Cardalano.

*Date of execution:* 3/12/2017.

*Test environment:* OS: Windows 10, Browser: Google Chrome.

**Unit testing** is carried out throughout the writing of the code to prevent any repeating bugs. This ensures code is written to a high standard and validates every unit of source code performs how it is designed.

**Integration testing** will follow Unit testing combining several units of code and testing them as one assembled group. Bottom-up testing technique will be used to identify major flaws in the smaller area of codes and to make test observations easier to monitor.

**System testing** will be the final implementation of testing of the entire integrated system. This will test the assemblages with the hardware system.

These testing methods will be carried out by JUnit framework.

**Failed cases:** In the case that significant defaults are found within the system, senior authority will be contacted and informed of identified flaws immediately. A plan will be set in motion in order to correct these flaws through recommendations set by the supervisors.