Avalon Online

SOFTWARE REQUIREMENTS SPECIFICATION

Ryan Austin Fernandez

Table of Contents

1 Executive Summary

Avalon is a board game created by Don Eskridge as an improvement over his previous board game, The Resistance, with gameplay similar to it, but with additional features.

The game involves the Knights of the Round Table attempting to succeed in three quests ordered by King Arthur. However, there are treacherous minions of Mordred amongst the loyal servants of Arthur attempting to fail the missions. It is the loyal servants' mission to discover the identity of the traitors

The additional features of Avalon with respect to The Resistance are its additional character cards: Merlin, Assassin, Perceval, Morgana, Mordred, Oberon, and the Lady of the Lake adding more social dynamics to the gameplay. (The Resistance, n.d.)

2 Overview of the Business Process

This chapter presents an overview of a simple Avalon gaming community's regular operations flow. Included in this chapter are the following items:

- *Community Flow and Features*
- *The game's flow and features*
- *Data requirements for storage of game records*
- *Different kinds of users in the community*
- *Proposed site's member privileges*

2.1 Community Flow and Features

In the community, at most one game may be hosted by a member. That game can be joined or left by the host's friends in the community, or anyone for community games. Once the maximum number of players (set by the host) or less, depending on whether the host wants to begin or not, have been met, the game begins. A game may also be cancelled before it begins.

A member may flag another member for misdemeanor, to be reviewed by a moderator or an administrator. An administrator may grant a member the rank of moderator or administrator.

2.1.1. Game Flow and Features

Each game has five to ten players and has a role distribution phase, a quest phase, and an end phase.

Given n players, there are $\lceil \boldsymbol{n/3} \rceil$ minions of Mordred present. At the beginning of the game, the roles, whether a player is a servant of Arthur (good) or a minion of Mordred (evil), are randomly distributed. The minions of Mordred then secretly discover each other's identities (in physical games, this is done by everyone closing their eyes then the minions open their eyes).

The quest phase comprises of five quests, each with sub phases, namely the selection, discussion, approval, quest proper, and reveal phases. A team captain is randomly designated at the beginning of the game. This team captain selects the appropriate number of people for the quest depending on how many players and which quest it is in the game. (see Appendix A-1). Once a team is selected, the players then discuss the composition of the team. Once a discussion has been raised, they will now vote on whether they approve or reject the team. If the majority (for even number of players, ties are not majority) approves, the selected players go on a quest and may choose to play a success or fail card. If the vote fails, the team captain token is passed to the left and the selection, discussion, and approval sub phases repeat. If the approval phase fails five times for a particular quest, evil automatically wins. Loyal servants of Arthur have to play a success card. Minions of Mordred may choose to play success or fail. Once they secretly give the team captain their decisions, the team captain shuffles the cards and reveals them one by one. If at least one fail is present, the mission fails and the appropriate mission marker is placed on the game board (see Appendix A-2). An exception is made for the fourth quest in games of more than seven players. It will take two fails for such quests. After each quest, team captain passes to the left.

Before the explanation of the end phase, it is necessary to discuss two optional special characters: Merlin and Assassin. When these two are in play, at the beginning, the evil players raise their thumbs and Merlin opens his eyes. Therefore, evil is revealed to Merlin. He may then influence the flow of the game. In the end phase, if one team has succeeded or failed in three quests, that team wins. However, if the good team has won, the assassin may now guess who among the good players Merlin is. If he is successful, the evil team still wins. A game with Merlin cannot be played without Assassin.

As for the other special characters, Perceval knows who Merlin is; Morgana appears as Merlin to Perceval; Mordred is hidden from Merlin; and Oberon is hidden from the evil players, but the evil players are also hidden from him. Oberon may choose to fail a mission if he pleases.

Regarding the Lady of the Lake, at the beginning of the game, the Lady token is given to the person to the right of the Team Captain. After the second quest, this person must choose a player to use the Lady of the Lake on. This player is given a red (evil) and a blue (good) loyalty card. They must give their loyalty card to the person who used the Lady of the Lake, who then returns the loyalty card. The Lady of the Lake is passed to the player it was used on and the previous member is now immune to the Lady of the Lake. The former member may now discuss the card that was given to him, even if he lies. This is repeated for the third and fourth quests.

Targeting is also an advanced game option. The team captain may choose to do the quests in any order, except the fifth quest, which may only be attempted after two successes. The fourth quest for games with at least seven players still require two fails. The Lady of the Lake still only functions after the second, third, and fourth quests.

After the game is finished, the statistics of the game are tallied. The members of the community may view these statistics for future reference, either on a per game basis or in total for each member or the entire community.

2.1.2. Statistics

The community and member statistics may be viewed by any member at any time. The specific details of which are further discussed in section 2.2.

2.2 Data Requirements for Storage of Community Records

Each member has a first name, a last name, a username, a password, a bio, a member level (bronze after playing 50 games, silver after playing 125 games, gold after playing 250 games), and an authorization level (regular, moderator, administrator). A member may only join one game at any given time.

Each game has a game id number, date and time hosted and ended, whether it is ongoing, whether it is a community or “friends-only” game, an optional minimum number of players (which cannot go below five), an optional maximum number of players (which cannot exceed ten), a list of special characters included, whether targeting is enabled, whether the Lady of the Lake is enabled, number of successes, number of fails, winner, and any discussion comments (similar to a chatbox) made during the game.

Each quest has at least one and at most five selections, with the captain who chose the team, each player’s vote for each selection and whether it was approved. It also has the quest results, with each member of the quest’s played card (success or fail) and the result of the quest.

For games with the Lady of the Lake, it is also necessary to mark players who are immune to the Lady of the Lake at any given point in time.

The community statistics comprises total games good won, evil won, good v. evil ratio, total game forfeits, total evil default wins (through five team rejections in one quest), and total games played.

The player statistics are total games played as good; total games played as evil; total games won and lost, as evil, as good, and in total; total games as any of the special roles; total games hosted; and total games played.

2.3 Roles in the Process

There are various kinds of members in the Avalon Game Community as shown in Table 2-1.

Role	Description of Tasks
Member	• Hosts a game • Cancels a game • Joins a game • Begins a game ○ Selects special characters for a game ○ Sets special rules for a game ○ Selects members to go on a quest ○ Votes for a team ○ Determines success of a quest ○ Discuss the game • Flags a member • View Community Statistics • View member Profile (with statistics) • Edit Own Profile
Moderator	• All member privileges • Moderates flagged members
Administrator	• All moderator privileges • Grants moderator or administrator privileges

Table 2-1. Member hierarchy in the Avalon Online

3 Problem Analysis

This chapter presents the problems with the current Avalon Gaming Community.

ID #	Description What's the problem?	Cause What causes the problem?	Symptoms How do we know the problem exists?	Impact Why is this important? What are the consequences?
1	There are no means to maintain the community listings.	• Nobody has started an Avalon gaming community yet.	• There is no official list of players, statistics, friends, and profiles. • There is no way to moderate members of the community.	• Avalon players cannot get together and play at their own leisure.
2	There are no means to maintain the game listings.	• Nobody has started an Avalon gaming community yet.	• There is no list of hosted games or ongoing games.	• It is difficult to get started with gameplay if you do not know any Avalon players.
3	The reveal phase is prone to unintended reveals.	• Players may be in close proximity to each other.	• Mild bumping, movement, or pupil dilation may give away information that was not intended to be given away.	• The mystery aspect of the game is lost.
4	It is difficult for everyone to keep track of quest members and votes.	• There are too many players sometimes • Players are too far away to see votes.	• Players keep asking who went or is going on which quest • Players keep asking who voted for which team.	• Players cannot make informed decisions.
5	It is difficult to maintain secrecy for quest results.	• The materials used may be compromised • The players may mix up the discard pile from the actual game pile.	• Mistaken quest results arise. • Information that was not intended to be given away may have been revealed by accident.	• The mystery aspect of the game is lost.

The Avalon Gaming Community is pretty much non-existent, having no records of those playing it actively, the games available, its statistics, its hosted games, and ongoing games. The game experience itself may be confusing, the reveal phase being susceptible to unintended reveals; and keeping track of gameplay and maintaining secrecy in quest results being difficult. A software to manage all of this data through a relational database, which can store and manipulate the said data easily, may be a solution.

4 Software Solution

4.1 Objectives

The Avalon Online Web Application aims to provide an online community for regular Avalon players to interact and play the game.

"The specific objectives of the software are as follows:

- *To provide a facility for maintaining community listings*
- *To provide a facility to maintain game listings*
- *To provide a facility for secure gameplay*
- *To provide a facility for community interaction"*

4.2 Characteristics

The software will be a web application that must be able to handle a large volume of data, be reliable, be extensible, be able to respond in a short time period, and have a usable interface.

4.3. Assumptions, Scope, and Limitations

The software will focus on the rules of the game stated in Section 2.1 and managing the community. Any additional rules existing in the community but not stated in Section 2.1 will not be included in the system.

5 User Stories

This chapter presents the member stories included in the product backlog.

5.1. Register Account

User Story #1: A non-registered member can register his/her account to be able to access the other features of the system.	
Estimate (Days): 8 hours	**Priority:** 10
Pre-condition: The member is on the register screen.	
Scenario: 1. The member inputs their first name, last name, username, password (twice), and bio. 2. The system verifies that the username is unique and notifies the member if otherwise. 3. The system verifies that the passwords match and notifies the member otherwise. 4. The system registers the new member data in the database. The password is salted and hashed. 5. The system displays the login screen.	
Post-condition: The member can now log in to the site.	
Acceptance Criteria: 1. Test if the system correctly determines if the username is unique and displays an error message if otherwise. 2. Test if the system correctly determines if the passwords match and displays an error message if otherwise. 3. Test if the system salts and hashes the password. 4. Test if the data was added to the database correctly.	

5.2. Login User

User Story #2: A member that is not logged in may log in to access the member privileges.	
Estimate (Days): 8 hours	**Priority:** 10
Pre-condition: The member is registered and is viewing the login screen.	
Scenario: 1. The member inputs their username and password. 2. The system checks if the username exists and displays a message if otherwise. 3. The system checks if the username matched with the password and displays a message if otherwise. 4. The system logs the member in. 5. The system displays the home screen.	
Post-condition: The member may now access the site's features.	

Acceptance Criteria: 1. Test if the system checks the username's existence correctly and displays an error message. 2. Test if the system checks the password correctly and displays an error message if not a match. 3. Test if the system logs the member in and the member can access the site features.

5.3. Host Game

User Story #3: A member can host a game to gather players for gameplay.	
Estimate (Days): 8 hours	**Priority:** 10
Pre-condition: The member is logged in.	
Scenario: 1. The member selects to host a game. 2. The member selects if it is a community or friends-only game. 3. The member inputs the minimum and maximum number of players (0 if not applicable). 4. The system registers the game in the database. 5. The system displays the game details screen.	
Post-condition: The game is now visible to the appropriate members of the community.	
Acceptance Criteria: 1. Test if the member inputs a negative number for maximum or minimum number of players and displays an error message when such event occurs. 2. Test if the game is registered into the database correctly.	

5.4. Cancel Game

User Story #4: A member can cancel a game to withdraw a game from the official listings	
Estimate (Days): 4 hours	**Priority:** 20
Pre-condition: The member has hosted a game, is viewing the game details screen, and has not begun the game.	
Scenario: 1. The member chooses to cancel the game. 2. The system asks for confirmation. 3. The member selects to confirm or cancel. 4. If the member confirmed, the game is cancelled in the database. 5. The system displays the home screen.	
Post-condition: The game is now cancelled.	
Acceptance Criteria: 1. Test if the system asks for confirmation. 2. Test if the system cancels the game in the database.	

5.5. Join Game

User Story #5: A member can join a game in order to participate in gameplay.

Estimate (Days): 4 hours.

Priority: 10

Pre-condition: The member is on the home screen.

Scenario:

1. The member chooses to view hosted games.
2. The system displays all games.
3. The member may choose to filter friends-only games.
   a. The system displays games that are classified as friends-only.
4. The member selects a game.
5. The system displays that game's details.
6. The member chooses to join the game.
7. The system checks if the member had already joined a game beforehand and displays an error if so.
8. The system checks to see if the maximum number of players has been exceeded and displays an error if so.
9. The system marks that member as joined.

Post-condition: The member can now play the game once it begins.

Acceptance Criteria:

1. Test if the system displays only non-cancelled, non-ongoing games.
2. Test if the system filters friends-only games properly.
3. Test if the system checks if the member had already joined a game beforehand and displays an error if so.
4. Test if the system checks the maximum number of players correctly and displays a proper error message.
5. Test if the member is added to the game's roster in the database.

5.6. Begin Game

User Story #6: A member can begin a game to start gameplay.

Estimate (Days): 16 hours

Priority: 10

Pre-condition: The member must be the one hosting the game and is viewing the game details screen.

Scenario:

1. The member chooses to begin the game.
2. The system checks if the minimum number of players has been met and displays an error message otherwise.
3. The system marks the game as ongoing.
4. The system distributes the characters to the players.
5. The system switches to the game screen.
6. The system displays the character card to each player.
7. The system displays the minions of Mordred to each minion.
    a. The system does not display Oberon.
    b. The system displays all evil players to Merlin
        i. The system does not display Mordred
    c. The system displays Merlin and Morgana to Perceval
8. The system randomizes a team captain.

Post-condition: The game can now proceed as noted in Section 2.1.

Acceptance Criteria:

1. Test if the system correctly checks the minimum players and displays the appropriate error message.
2. Test if the system marks the game as ongoing in the database.
3. Test if the special characters selected have been distributed properly to the players.
4. Test if the reveal phase is correct.
5. Test if a team captain is randomized.

5.7. Select Special Characters

User Story #7: A member can add special characters to a game to increase its complexity.

Estimate (Days): 8 hours

Priority: 20

Pre-condition: The member is hosting a game, viewing the game details screen, and has not yet begun the game.

Scenario:

1. The member chooses to select special characters.
2. The system displays the good and evil special characters.
3. The member selects the characters s/he wants to include.
4. The member confirms the selection.
5. The system notes which characters have been included in the game.

Post-condition: Special characters have now been made available in the game.
Acceptance Criteria: 1. Test if the system monitors the maximum good/evil characters allowed in the game and displays an error message if more are added. 2. Test if the selected characters are successfully added to the database.

5.8. Select Special Rules

User Story #8: A member can select special rules for a game to add complexity.	
Estimate (Days): 4 hours	**Priority:** 20
Pre-condition: The member is hosting a game, viewing the game details screen, and has not yet begun the game.	
Scenario: 1. The member chooses the special rules option. 2. The system displays options to enable targeting and Lady of the Lake. 3. The member selects the rules s/he wishes to include. 4. The member chooses to set the rules. 5. The system records the rules in the database. 6. The system returns the member to the game details page.	
Post-condition: Special rules have now been enabled as selected by the member.	
Acceptance Criteria: 1. Test if the special rules selected were added in the database.	

5.9. Select Quest Members

User Story #9: A member can select members for a quest.	
Estimate (Days): 8 hours	**Priority:** 10
Pre-condition: The member is part of a game that has begun, is viewing the game screen, and is a team captain at the moment.	
Scenario: 1. The system displays the list of players. 2. The member selects which players to send on the quest. 3. The system confirms the team. 4. If the member confirmed, the system record the team in the database. 5. The system displays the game screen.	
Post-condition: The other members can now vote for the selected team.	

Acceptance Criteria:

1. Test if the system asks for confirmation when the number of team members has been met.
2. Test if the system records the team correctly in the database.

5.10. Vote For Team

User Story #10: A member can vote for a selected team to influence the game.

Estimate (Days): 4 hours

Priority: 10

Pre-condition: The member is viewing the game screen and a team has been selected.

Scenario:

1. The member chooses to approve or reject the team.
2. The system records the response in the database.
3. The system waits for the other players' votes.
4. The system displays the votes of the players.
   a. If majority approved, the system moves on to US#11.
   b. Otherwise, the vote track is incremented and the system returns to US#10, passing the team captain role.
      i. If five teams have been rejected, evil is marked as the winner of the round.

Post-condition: A team can now go on a quest; a new team is to be selected; or evil wins.

Acceptance Criteria:

1. Test if the response is recorded correctly in the database.
2. Test if the system waits for everyone to vote.
3. Test if the system moves to the correct scenario given the votes.
4. Test if the system increments the vote track if majority does not approve.

5.11. Determine Success of Quest

User Story #11: A member can determine if a quest succeeds or fails to further the game.

Estimate (Days): 4 hours

Priority: 10

Pre-condition: A team must have been approved via US#10.

Scenario:

1. The member chooses the fate of the quest.
   a. If the member is good, a success card is automatically cast and stored in the database.
   b. If the member is evil, a choice is presented by the system.
      i. The member selects success or fail
      ii. The choice is stored in the database.

2. The system waits for all quest members to cast their choices.
3. The system checks if the fail threshold was met.
4. The choices are revealed in random order.
5. The result of the quest is marked on the score tableau.
6. If the Lady of the Lake is enabled, the player holding it selects another player.
    a. The loyalty of said player is revealed to the Lady of the Lake user
    b. The Lady of the Lake is passed to the player it was used on.
7. The system proceeds to
    a. US#9 if there is no winner yet.
        i. The system passes the team captain token.
        ii. The system resets the vote track.
    b. Allow Assassin to select a player as Merlin if good wins and Assassin is in the game
        i. The system confirms Assassin's choice
        ii. If correct, evil wins the game and the result is recorded in the database.
        iii. If wrong, good wins the game and the result is recorded in the database.
    c. Record's Evil's win in the database if Evil has won the game.

Post-condition: A quest has now been completed. There may be a winner.

Acceptance Criteria:

1. Test if the system automatically casts a success for good players.
2. Test if the system gives evil players a choice.
3. Test if the system records choices correctly in the database.
4. Test if the success of the quest is correctly checked.
5. Test If the Lady of the Lake is triggered.
6. Test if the system proceeds to the appropriate member story.
7. Test if the system passes the team captain token and resets the vote track
8. Test if the system gives Assassin their choice if good wins.
9. Test if the system records the winner of the game correctly in the database.

5.12. Discuss Game

User Story #12: A member can discuss the game to enrich the gaming experience.

Estimate (Days): 16 hours	**Priority:** 40

Pre-condition: A member is part of the game s/he is viewing and the game has begun.

Scenario:

1. The member inputs their message.
2. The member selects to send their message.
    a. The system informs the member if the message is longer than five hundred and twelve characters and returns to step 1.
3. The message is added to the list of messages for that game in the database.
4. The system displays the message along with the other previous messages.

Post-condition: The member's message has been added to the message thread.

Acceptance Criteria: 1. Test if the system informs the member if the message is too long. 2. Test if the system records the message in the database correctly. 3. Test if the system displays the message.

5.13. Flag Member

User Story #13: A member may flag another member to inform the moderators of misdemeanor in the site.	
Estimate (Days): 4 hours	**Priority:** 50
Pre-condition: The member is logged in and is viewing another member's profile screen whose owner has not yet been flagged.	
Scenario: 1. The member chooses to flag the viewed member profile. 2. The system marks the member as flagged. 3. The system displays the member's profile screen.	
Post-condition: The selected member can now be moderated by a moderator or administrator.	
Acceptance Criteria: 1. Test if the member is marked as flagged. 2. Test if the system hides the flag option for flagged members.	

5.14. Moderate Member

User Story #14: A moderator or administrator level member can moderate a member to ensure civility on the site.	
Estimate (Days): 4 hours	**Priority:** 50
Pre-condition: The member is viewing the profile page of a flagged member.	
Scenario: 1. The member chooses to moderate the viewed member. 2. The system asks the member to input their password for confirmation. 3. If the password is correct, the system deletes the viewed member from the community. 4. The system displays the home screen.	
Post-condition: The selected member is deleted from the community.	
Acceptance Criteria: 1. Test if the system checks if the viewed member is flagged. 2. Test if the system asks for confirmation via password. 3. Test if the system deletes the member after confirmation.	

5.15. Friend Member

User Story #15: A member can friend another member to enforce a tighter gaming community.	
Estimate (Days): 4 hours	**Priority:** 10
Pre-condition: The member is logged in and is viewing a game details screen.	
Scenario: 1. The member selects a member joined in the game. 2. The system displays their profile screen. 3. The member selects to add them as a friend. 4. The system sends the friend request. 5. The other member is asked to confirm. 6. If the other member confirms, both members are marked as friends in the database.	
Post-condition: Both members are now friends.	
Acceptance Criteria: 1. Test if the friend option is not visible for members who are already friends. 2. Test if the friend request is sent. 3. Test if the system records members as friends if the other has confirmed a friend request.	

5.16. View Friends

User Story #16: A member can view their friends to keep track of their friends in the community.	
Estimate (Days): 4 hours	**Priority:** 30
Pre-condition: The member must be logged in and viewing the home screen.	
Scenario: 1. The member chooses to view their friends. 2. The system retrieves the friends from the database. 3. The system displays each friend’s first name, last name, and username.	
Post-condition: The member is now viewing their friend list.	
Acceptance Criteria: 1. Test if the only profiles displayed are the member’s friends.	

5.17. Alter User Level

User Story #17: An administrator can alter another member’s level to add administrators or moderators to the site.	
Estimate (Days): 8 hours	**Priority:** 40
Pre-condition: The member is logged in as an administrator and is viewing another non-administrator member’s profile	
Scenario: 1. The administrator selects to alter the member’s level. 2. The administrator selects if the member is to become a moderator or an administrator. 3. The system records the new information in the database.	
Post-condition: The altered member’s new level is now stored in the database.	
Acceptance Criteria: 1. Test if the system hides the option to change levels to administrators. 2. Test if the system records the new information in the database correctly.	

5.18. View Member Profile

User Story #18: A member can view another member’s profile to gain more information about a fellow community member.	
Estimate (Days): 4 hours	**Priority:** 30
Pre-condition: The member must be viewing a game details screen or their friends list.	
Scenario: 1. The member selects a member to view. 2. The system displays the member’s first name; last name; username; bio; total games played as good; total games played as evil; total games won and lost, as evil, as good, and in total; total games as any of the special roles; total games hosted; and total games played..	
Post-condition: The member is now viewing another member’s profile.	
Acceptance Criteria: 1. Test if the system displays the correct information.	

5.19. Edit Profile

User Story #19: A member can edit their profile to keep their information updated.	
Estimate (Days): 8 hours	**Priority:** 30
Pre-condition: The member must be viewing their own profile.	
Scenario: 1. The member chooses to edit their profile. 2. The system prompts them to enter a new first name, last name, username, password (twice, along with the old password), or bio, if applicable. 3. The member confirms the changes. 4. The system checks if the username is still unique and displays an error message if otherwise. 5. The system checks if the old password is correct and displays an error message if otherwise. 6. The system checks if the two passwords match and displays an error message if otherwise. 7. The system stores the changes in the database, salting and hashing the password. 8. The system displays the member's profile screen.	
Post-condition: The member's profile has been updated.	
Acceptance Criteria: 1. Test if the system correctly determines if the username is unique and displays an error message if otherwise. 2. Test if the system correctly determines if the passwords match and displays an error message if otherwise. 3. Test if the system salts and hashes the password. 4. Test if the data was updated in the database correctly.	

5.20. View Community Statistics

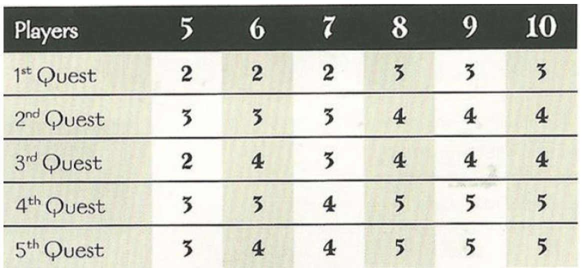
User Story #20: A member can view community statistics to see the standing of the community.	
Estimate (Days): 8 hours	**Priority:** 50
Pre-condition: The member is on the home screen.	
Scenario: 1. The member chooses to view community statistics. 2. The system displays total games good won, evil won, good v. evil ratio, total game forfeits, total evil default wins (through five team rejections in one quest), and total games played.	
Post-condition: The member can now see the community statistics.	
Acceptance Criteria: 1. Test if the data shown is accurate to the data in the database.	

Appendix A – Sample Forms and Reports

This chapter contains the various gameplay materials needed for the site.

A-1. Player/Quest Table

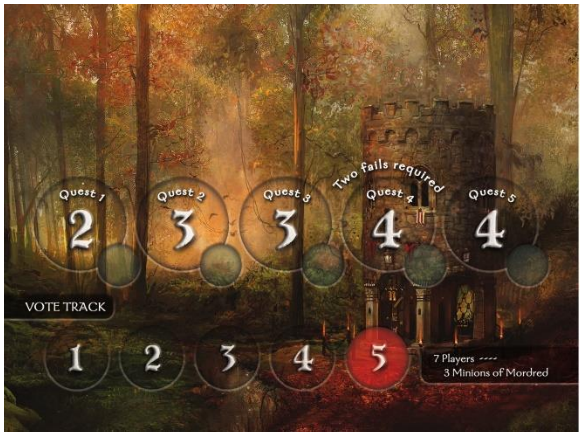
This table notes the number of players to send on each quest in the game, depending on the total players.



Players	5	6	7	8	9	10
1st Quest	2	2	2	3	3	3
2nd Quest	3	3	3	4	4	4
3rd Quest	2	4	3	4	4	4
4th Quest	3	3	4	5	5	5
5th Quest	3	4	4	5	5	5

A-2. Score Tableau

This is a sample of a score tableau used for the game.



Appendix B – References and Acknowledgement

The Resistance. (n.d.). Retrieved April 30, 2015, In Wikipedia: http://en.wikipedia.org/wiki/The_Resistance_%28game%29