Milestone 4 Beta Launch and Reviews

CEN 4010 Spring 2018

Team #4

April 16,2018

*Witch Hunt*

Ivan Maykov: Product Owner

John Floyd: Scrum Master

Kyle Prince: Team Member

Sha’Quana Jones: Team Member

|  |  |
| --- | --- |
| Revision | Description |
| 1.0 – 04/16/2018 | Initial Document Release |

**2.2 Product Summary:**

Product Name: Witch Hunt

Witch Hunt is a web-based multiplayer game. It enables people to play with any person, as long as they are able to connect to the internet. As the mobile market is becoming increasingly popular, there is no better time to make this game. As the game only requires an internet connection and a web browser, it enables players to play with others regardless if they’re using a smartphone, desktop, or laptop computer. The game starts with one player being selected as the witch while everyone else is a villager. The game is played in rounds, with players using a chat that enables them to communicate with others during it. After a certain period of time, the other players vote on who is the witch. That person is removed from the game (loses/is killed). If they are correct, the surviving villagers win. If they are wrong, the witch kills a villager. The witch wins if they are the last man standing or it’s just them and one villager. As it is browser-based, the game is able to be played regardless of physical distance to other players.

All major committed functions:

There are a variety of major functional capabilities that the team will be implementing in the final release of the application.

* The application will be accessible on the LAMP server through its URL.
* It will enable users to register with a username and password
* It will enable users to login with a previously created username and password.
* Logged in users will be able to join a previously created public game
* Logged in users will be able to join a previously created private game, given an entrance password.
* Logged in users will be able to create a private game
* During a game, users will be able to choose a player to cast their vote on.
* Users will be notified of who was voted out, and whether they were a Witch or not.
* The person who was voted on by the Witch will be voted out.
* Users will be able to send messages during a game.

Unique Features:

Being able to play through the browser is a unique feature as it enables players to play regardless of the device used to access a web browser.

URL: <http://lamp.cse.fau.edu/~CEN4010_S2018g04/WitchHunt/>

**2.3 Usability Test Plan**

Overview: This usability test plan will test Witch Hunt’s voting during its development. Voting in the game works as such. You have a number of players on their phones join a game. One of the people will be selected to be a witch, everyone else is a villager. The game is played in rounds. Each round the players discuss who is the witch. After a certain period of time, the other players vote on who is the witch. That person is removed from the game (loses, is killed). If they are correct, the surviving villagers win. If they are wrong, the witch kills a villager. The witch wins if they are the last man standing or it’s just them and one villager. This test will establish user performance as well as test for any design inconsistencies to ensure a functional and pleasing game. Design errors will include testing for design (presentation), navigation errors (visual flow of game) and user requirements (ensuring user is satisfied with game functionality). The usability test will use multiple games to test the functions of the game itself and use the players as the user group to provide feedback on any inconsistencies that may arise through testing. The testing will begin with the players roles being set and the rounds to determine which player is the witch in play. We will check for any errors in design at this point and make sure user performance is successful in connection for communicating to determine the witch. We will then test for the code to count those who have been “killed” to be unable to play the game but see the end results of who won. After many rounds there will be a point where voting of who is the witch will take place and we will test the design it total of navigation and so on.

Plan: The first step in planning will be to ensure that the URL in which the game takes place works and can be accessed by users of the game. Completion Criteria will include either the witch winning (By being the last man standing or face to face with one villager) or the villagers who are trying to take the witch out (guess) guessing right on which player is the witch. We as a group are the intended user to ensure the proper testing for a successful and satisfying game. Our goal is to provide a great design visually as well as maintaining a functional and free flowing game for all users to enjoy. The starting point of our game would be for either a user to log in or create a login in order to have access to witch hunt. These aspects will contribute to the testing to ensure the voting of the game is functional as it’s an important piece, without it there is no point to the game as the goal is to guess the witch and win!

Questionnaire Form: Choose from the following questionnaire by clicking on the response that best fits your feelings on the following characterizations where 1=Strongly Disagree, 2=Disagree, 3Neither Agree Nor Disagree, 4=Agree and 5Strongly Agree.

**2.4 QA test plan**

**2.5 Code Review**

**2.6 Self-check on best practices for security**

For the application, the only way to play is if the user has a username and password. The username will be visible to other players and therefore does not require extensive security measures. However, the player’s password needs to be protected to prevent unauthorized access. Therefore, the player’s password will be encrypted when sent to the server for storage when the player creates a login.

Password encryption has been verified and password inputs have been tested using the following input set:

{

12345678

Dj38hs9k

J8h83hsa

8djfn451

34mkd81q

30nhA73d

}

**2.7 Self-check on nonfunctional specs**

1. Performance - Response time - maximum load time, 1 sec. DONE

2. Usability / Portability – Game must be properly formatted and functional on all iOS and Android mobile devices ON TRACK

3. Accessibility – Users must have an account and login to access the game DONE

4. Expected Load – During game play, max 8 user concurrently. As popularity of game increase and multiple games are played concurrently, expected load will increase. Initially a max of 100 concurrent games with a max of 8 players, requires 800 concurrent users with a max pull of 250KB per user. DONE

5. Security Requirements – Any personal data needs to be encrypted when stored on database DONE

6. Storage – Web server storage requirements 100MB max, Database max size initially 1GB. Individual pages displayed should not exceed 1MB for faster load times. DONE

7. Fault Tolerance – Web site can be hosted on FAU Lamp server initially. Further growth will require hosting with a service provider with a min 99.98% uptime with daily site and database backups. Redundant server is unnecessary until 10,000 users are reached. DONE

8. User Signup – Simple and quick signup with limited user personal information: username and password only. DONE

9. Ease of Use – User should be able to navigate game and play without instructions. If user does need view instructions, they should be brief and no more than 7 paragraphs. ON TRACK