Requirements Document

John Melloni, Wyatt Doiron-Larue, Kachi Onyenokwe, Ayodeji Marquis

**1. Introduction**

**1.1 Purpose**

What Do U Want is a software that is intended to allow the user(s) to create a prompt that will reach a fair decision in an optimal amount of time, allowing users to proceed with their plans..

The purpose of this document is to fully describe all requirements of our software. This guide will serve as a template for the developers to reference when creating the features of What Do U Want. It will also serve as the validation guidelines for each intended feature.

**1.2 Scope**

This section contains the features that are in the scope of our software, as well as clarifies some features that are not in the scope of our software.

In Scope:  
A. Allowing users to create a room, and send out the room with a keyword.  
B. Allowing users to communicate with each other while in the same room   
C. Allowing users to generate a prompt regarding a decision to be made  
D. Allowing users to respond to a prompt, and display their results in a pie chart.  
E. Allowing users to change their vote via clicking on the pie chart.  
F. Allowing any user to initialize the randomization.  
G. Allowing users to create a new prompt following a resolved prompt.

H. Storing user information.  
I. Storing room information.

Outside of Scope:  
A. Suggesting possible alternative ideas for users.  
B. Allowing users to track the location of other users.

C. Disable create prompt button while a prompt is active

D. Giving certain users different privileges (no group leader)

**2. User Requirements**

**2.1 Software Interfaces**

- Web browser (Google Chrome)

- Github

**2.2 User Interfaces**

|  |  |  |
| --- | --- | --- |
| Class of Use Cases | Use Cases | Description of Use Cases |
| Room Creation | Room initialization | Creates and initializes the room. |
| Room Name and Password | Forces user to create a room name and password |
| User Login | User Log-in | Prompt user for username and password |
| Room Join | Allow user to join existing room |
| Create Room | Allow user to create a room |
| Communication | Groupchat functionality | Continuously allow messages to be sent between all users in a room |
| Prompt | Prompt Creation | By pressing the button a user is asked to enter a prompt |
| Response Display | Pie chart is displayed to all users in room |
| Response Editing | Users can change their responses by selecting a different piece of the pie chart. After a short delay, an updated pie chart will be displayed. |
| Randomize or Not | Any user has the option to call forth a randomization, or to just end the prompt if an agreement is reached. |
| Weighted Randomization | If selected, perform a randomization where each vote has equal weight, not each suggestion. |
| Result Notification | Result is displayed for everyone in the room to see |
| Post-Prompt | Clean Up | Allow for the prompt button to be selected again by any user. |

**2.3 User Characteristics**

This software is appropriate for users of all ages and lifestyles. The intended audience is high-school to college-aged young adults who have free time, and would like to optimize the spending of it. The professional workplace is not our intended audience.

**2.4 Unmet Requirements**

Allowing user to send link and keyword that would allow them to join a room to others

Allowing user to choose name and color whenever entering a room

“Group leader” aspect

Notification sent to users when prompt created

Entering a response time limit when prompt is created that ends the poll after the allotted time

**3. System Requirements**

**3.1 Functional Requirements**

**Use Case Class:** Room Creation

1. User clicks a button to create/join a room.
2. User enters room name and password
3. Room consists of an interface with 3 panels; one is for a chat board, one is for a pie chart, and one is for a list of rooms the user is in.
4. User tells others the room name and password, others can use these to join the room

**Use Case Class:** User Login

1. Prompts user for username and password
2. Allows user to join/create a room
3. Shows rooms the user is in
4. Other users will see and recognize each other by their usernames in the chat board.
5. Usernames will be used across all rooms the user is a part of
6. If a user types in a username that is already being used, asks user to choose a different username.

**Use Case Class:** Communication

1. The chat board panel shows all previous messages (can scroll up and down to view recent or older messages) with the username of the message sender next to the message.
2. Includes a small section at the bottom that allows a user to input and send a message. This message will show up on the chat board.

**Use Case Class:** Prompt

1. At any time, any user can create a prompt that all users in the room can see
2. Users can write-in their own solutions to the query or they can select from pre-existing solutions other users wrote in.
3. A pie chart representing the different solutions and how many users voted for each one will be displayed in the room.
4. One vote per user. A user can change their vote by clicking on the part of the pie chart that represents their desired solution.
5. Every time a vote is added or changed, an updated pie chart will be created and displayed after a short delay.
6. Everyone has the option to end voting polls and to spin a randomizer that selects a solution at random based on weights that are based on the number of votes a solution has.
7. Show the result of the spin to everyone in the room

**3.2 Non-Functional Requirements**

3.2.1 Software Quality Attributes

**Security:**

The login and chatting system will have a secure database because people’s privacy is very important to us. The system will be set up so as to ensure that messages are sent over secure servers, preventing hackers and phishers from gaining access to passwords and messages.

This will be measured using the following penetration assessments such:

* Authentication assessment (AUTH)
* Cryptography use assessment
* Data Validation using user entry management assessment

**Reliability:**

* **Chat History Server/Database:** A reliable chat history database is important so as to allow users see past conversations up to a period of at least 2 months.
* **Crash Free:** A crash free system will allow the users have a smooth experience. What Do U Want is trying to sell based on it’s simplicity, so having a crash free system will increase our success rate. Unit test cases will be developed to catch bugs.

**Minimalistic UI:**

Since the app will be website based, a lot of the GUI (Ie. scripting and general look of the web pages) will be implemented using Cascading Style Sheets and Javascript. The system will have a simple UI that is exquisite to the user’s eye. So simple yet so sophisticated in its design that it accomplishes our goal. Kind of similar to Google’s material design. This will be measured by beta testers.

**Web development:**

Our application will be run on a web browser. The web based app will run on the latest version of HTML5. This version of HTML allows for easy website scaling, allowing for easy use through a mobile web browser so that if we do not end up having the time to expand the software, it can still be used on a mobile device.

**DBMS (Database Management):**

We changed our mind, and instead decided to incorporate user accounts, so we need to store information such as name,password, etc. and in order to store this user information, a database must be created to house all of it. We considered using Microsoft Access but decided to go with MySQL because of its portability between operating systems. MySQL is also free and great to learn about DBMS with.

**Testing:**

Software testing is a process of executing a program or application with the intent of finding the software bugs. It can also be stated as the process of validating and verifying that a software program or application or product meets the business (user) and technical (system) requirements that guided its design and development. Each feature of our product will be deemed complete when the intended use case is able to be enacted, fully through.