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 invite via a link, with a keyword.  
B. Allowing users to choose their name and color upon joining the room.  
C Allowing users to communicate with each other, and see other users currently in the room..   
D. Allowing users to generate a prompt, which will then deactivate the prompt button for all other users, until the arbitration process is completed.  
E. Allowing users to be notified that a prompt has been initiated, that their input is needed, and that a prompt has been resolved.  
F. Allowing users to respond to a prompt, and displaying their results in a pie chart.  
G. Allowing users to change their vote via clicking on the pie chart.  
H. Allowing the user that initialized the prompt to initialize the randomization.  
I. Allowing users to create a new prompt following a resolved prompt.

Outside of Scope:  
A. Storing user information.  
B. Storing room information.  
C. Suggesting possible alternative ideas for users.  
D. Allowing users to track the location of other users.

**2. User Requirements**

**2.1 Software Interfaces**

- Phone’s messaging system

- Phone’s notification system

- Groupchat messaging system

- Github

**2.2 User Interfaces**

|  |  |  |
| --- | --- | --- |
| Class of Use Cases | Use Cases | Description of Use Cases |
| Room Creation | Room initialization | Creates and initializes the room. |
| Keyword generation | Forces user to create a keyword. |
| Distributable link generation | Generates a distributable link. |
| User Login | User Log-in | Prompt user for keyword. |
| Username Input | Force user to input a username. |
| Color Selection | Force user to select a color. |
| Display of users | Continuously display all users with their color. |
| Communication | Groupchat functionality | Continuously allow messages to be sent between all users. |
| Prompt | Prompt Creation | By pressing the button, one user becomes the “Groupleader.” |
| Short form | Groupleader fills out the prompt to be distributed and responded to. |
| User Notification | A notification must be sent to users that a prompt has been created. |
| Response Time Limit | Groupleader must select an allotted time for responses to be gathered by. |
| Response Management | Groupleader is allowed to combine similar responses by manipulating the pie chart. |
| Response Display | Pie chart is now displayed to all users. |
| 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 | Groupleader 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 | Allow the groupleader to formulate a message that will be sent out to all users whether randomization was selected or not. |
| 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 Assumptions and dependencies**

N/A

**3. System Requirements**

**3.1 Functional Requirements**

**Use Case Class:** Room Creation

1. User clicks a button to create a room.
2. 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 users in the room.
3. Prompts user to create a keyword.
4. A distributable link is then generated.
5. The link can be sent through text message and if opened will navigate the user to the designated room.

**Use Case Class:** User Login

1. Prompts user for keyword which allows access into the room.
2. Prompts user to input username.
3. Prompts user to choose a color that has not already been picked by another user.
4. Other users will see and recognize each other by their usernames in the list of users and the chat board (username characters will be of the color that they chose).
5. Usernames will be local to a room, users can have different usernames in different rooms.
6. If a user types in a username that is already being used in the room by another user, asks user to choose a different username.
7. Users can exit rooms if they want and won’t be included in the chat board or bothered by notifications after leaving.

**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, image, or link. This message will show up on the chat board.

**Use Case Class:** Prompt

1. When a user creates a room, they become that room’s “groupleader”.
2. After a room is created and other users have joined, the groupleader types the query that they and the other users wish to make a decision about.
3. Notifications are sent to users that this query has been created and the query will be displayed in the room.
4. Users can write-in their own solutions to the query or they can select from pre-existing solutions other users wrote in.
5. A pie chart representing the different solutions and how many users voted for each one will be displayed in the room.
6. One vote per user. A user can change their vote by clicking on the part of the pie chart that represents their desired solution.
7. Every time a vote is added or changed, an updated pie chart will be created and displayed after a short delay.
8. When the groupleader creates the query, they must also choose a time limit after which no more votes or solutions will be added or processed.
9. Groupleader can combine similar solutions into one, adding the votes of the combined solutions and updating the chart accordingly.
10. Groupleader 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.
11. Allow the groupleader to formulate a message that will be sent out to all users whether randomization was selected or not.

**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 are very important to us. The system will be set up so as to ensure that messages are sent over secure servers so as to prevent 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
* Content Leak assessment
* Intent Spoofing

**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 or Mobile development:**

Our application will initially be run on a web browser before it is then expanded to allow for mobile app compatibility (if possible). 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):**

If we change our mind, and instead plan to incorporate user accounts, we will need to store information such as name, email, age, 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.