CEN 4010 Principles of Software Engineering, Summer 2020

Milestone 4 Beta Launch and Reviews

Group 6 – Covid Communicator

Team number: 6

Team members:

Product Owner: Grant Lundberg: [glundberg2017@fau.edu](mailto:glundberg2017@fau.edu)

Scrum Master: Christian Bastien: [cbastien2018@fau.edu](mailto:cbastien2018@fau.edu)

Development Team: John Callaghan: [jcallaghan2019@fau.edu](mailto:jcallaghan2019@fau.edu)

Development Team: Rishi Patel: [rishipatel2018@fau.edu](mailto:rishipatel2018@fau.edu)

Documentation Date: 7/28/20

History Table:

|  |  |
| --- | --- |
| Date of Revision | Description |
| 7/28/20 | Changing the original requirement in Milestone 1 which required the app to be accessible on web browsers. |
| 7/28/20 | The user will log in with a username rather than entering their email address. |

**Product Summary**

The world today has changed a great deal in the past few months. With the rise of Covid-19 forcing people into lockdown, feelings of isolation and loneliness are inevitable. Now more than ever, it is difficult to feel connected to other people. This is why we developed the “Covid Communicator.” Covid Communicator is a desktop app which will allow those who feel lonely the chance to chat with others who feel the same. The user will be connected to another in a chatroom and given the opportunity to chat about whatever they like. Our app values friendliness and will attempt to brighten the day of anyone who uses it. Covid Communicator will be a mental health benefit to those who feel lonely during the epidemic.

Product Name: Covid Communicator

Major Committed Functions:

1. Your username and password are only requirements to log in.

2. The password will be encrypted in the server.

3. The user will be able to send a message and see it in the chatroom.

4. The chat will be completely anonymous.

**Usability Test Plan**

Test objectives: The goal is to test the usability of the upload feature of our app. Once the user connects to a room, we want to make sure they can type in a message and then enter their message into the chatroom. This message will be relayed to the database and stored, then shown back to the user. We also want to test whether the app is simple and clear enough, so the user knows exactly what to do without any issues. This will tell us whether the app is simple enough to navigate, or if changes must be made to make the app more accessible to the average user.

Test plan: The user will start on the home screen of the app. They will create an account by entering their email address and password. Once they are taken to the chatroom, they will attempt to type a message into the message bar. The test will be complete once the user sees their message entered and saved in the chatroom. The intended user should be someone who has no knowledge of how the app works.

Questionnaire:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Completely Agree | Somewhat Agree | Neutral | Somewhat Disagree | Completely Disagree |
| The app was easy to navigate. |  |  |  |  |  |
| Sending a message was quick and responsive. |  |  |  |  |  |
| The app felt inviting and friendly. |  |  |  |  |  |
| The app ran efficiently and loaded quickly. |  |  |  |  |  |

**QA test plan**

The purpose of this test is to make sure various functions are running correctly without issue. We want to test the basic functions which the user will do every time they log in. This includes entering the app and logging into or creating an account and typing a message into the chatroom.

Hardware and software setup: For the software of our app, we had to install python3 and PyQt5 module. This allowed us to create the code for our app.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Title** | **Description** | **Input** | **Expected Output** | **Results** |
| Create an account | Enter email and password to create an account. | Type in your email address and password. | An account will be created, and you will be taken to the chatroom. | **Pass** |
| Log into an existing account. | Enter email and password to an existing account. | Type in your email address and password to an existing account. | The account should be saved, and you will be taken to the chatroom. | **Fail** |
| Enter a message | Once you are in the chatroom, enter a message into the message bar. | Type in your message. | The message should be relayed to the database. | **Pass** |

FAIL REASON: The issue we have with the app currently is connecting it to the DB on the LAMP server. We are currently working on solutions and we believe it can be fixed before official launch.

**Code Review**

For our coding style, we decided to use python and we tried to conform to python’s pep8 style guideline. Grant wrote the code for the new chat window; John reviewed it and gave comments.

Code:

from PyQt5 import QtCore, QtGui, QtWidgets

from PyQt5 import uic

class ChatWindow(QtWidgets.QDialog):

def \_\_init\_\_(self):

super(ChatWindow, self).\_\_init\_\_()

uic.loadUi('./chat\_window.ui', self)

def on\_sendButton\_clicked(self):

print('sending message to database: {}'.format(self.messageLineEdit.text()))

Comments made:

The main thing I would say is that you didn’t properly generate the window & dialog classes to avoid duplicate click events. Other than that, the only thing would be adding headers to the classes is good practice. However, the code should still work as is. Good job.

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

Major asset we are protecting: Password.

The major asset which we are protecting for the user is their password. Their password will be encrypted in the database once we get the database connected to the app. We will ensure that their password is safe.

**Self-check: Adherence to original Non-functional specs**

1. Users shall log in with a password: DONE

1.1 User will also log in with username: DONE

2. Passwords must be encrypted in server: ON TRACK

* Once we connect the app to the DB, the password will be encrypted and stored.

3. The system should be able to support at least 100 users: ON TRACK

**Peer review**

|  |  |
| --- | --- |
| **Member** | **Participation** |
| John Callaghan | 25% |
| Grant Lundberg | 25% |
| Christian Bastien | 25% |
| Rishi Patel | 25% |

**Scrum practice management Trello:**

<https://trello.com/cen4010s2020g06>

GitHub Repository:

<https://github.com/glundberg2017/cen4010-s2020-g06>

M4 Demo: https://youtu.be/0mWEoP0CYSE