CEN 4010 Principles of Software Engineering, Summer 2020

Milestone 3: More Detailed Requirements, Architecture and a Vertical Software Prototype Group 6 – Covid Communicator

Team number: 6

Team members:

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

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

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

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

Documentation Date: 6/15/20

History Table:

**Executive Summary**

The world today seems like a dark place. 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. Depriving people of this connection is dangerous to their mental health. While it is still unsafe in many places to interact with people, there are still ways to allow people to feel that connection they are missing. This is why we will develop the “Covid Communicator.” The Covid Communicator is a desktop app which will allow those who feel lonely the chance to chat with others who feel the same. 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 loneliness during the epidemic.

**Competitive analysis**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Features | Chatting between users | Likes and reactions | Focus on mental well-being | News feed | Allows advertisements on app |
| Covid Communicator | yes | no | yes | no | no |
| Twitter | yes | yes | no | yes | yes |
| Facebook | yes | yes | no | yes | yes |

While there are already applications which are designed to help a user connect with others, those connections are not always positive. Also, these applications are used as news sources and a place for advertisements, which can reduce the overall mental health benefit. Our product will be designed to capitalize on the features which competitors such as Facebook and Twitter are lacking in. Mainly, our application will be a service to our user’s mental health and will be designed to be an enjoyable break from the harshness of the world today. Given the overall draw of the app is connection while isolating, we expect the interactions will be more pleasant than those on social media sites.

**Data definition**

Covid-19: a mild to severe respiratory illness that is caused by a coronavirus.

Covid Communicator: A desktop app which our team will be developing.

Python: a high-level general-purpose programming language.

App: an application which allows you to perform specific tasks or offers a service.

GitHub: web-based version-control and collaboration platform for software developers.

Visual Studio Code: Source-code editor made by Microsoft.

Qt: Used for developing graphical user interfaces (GUIs) and multi-platform applications that run on all major desktop platforms and most mobile or embedded platforms.

LAMP: is a web development stack that has Linux, Apache, MySQL and PHP components

**Overview, scenarios and use cases**

The average user of this service will be a very social person who longs for the social interaction which Covid-19 has diminished. They will be someone who wants to create small talk or have a casual conversation with other people in lockdown. Given that the pandemic has forced many people to only interact with close family and friends, this is an opportunity to communicate with someone other than those who you are quarantined with. This user could be inexperienced with apps or software, which means our app must be simple to use. They should also be able to access the app and begin using it without any tutorials or prior knowledge. The app will allow the user to communicate with other people easily. Given the types of users which will gravitate toward this service, the interactions should be very pleasant and more casual than interactions on other sites.

Scenario: Mike is a hair stylist at a local barbershop. He has worked at the same barbershop shop for 15 years and has many clients which he has become close with. He looks forward to seeing and talking with them about their lives. However, due to Covid-19, the barbershop had to close indefinitely, and he is now isolated at home. Through the weeks of quarantine, he grows lonely. He misses the casual conversations he once had daily with his clients. He learns about a new app called “Covid Communicator” which can bring back the casual human interaction he misses. After using the app, he realizes the conversations he had on Covid Communicator are very similar to the ones he had as a barber and begins to feel less lonely.

**High-Level Functional Requirements**

1. The user should be able to understand the app with no tutorial or time spent learning.

2. The system should display bright imagery with happy connotations.

4. The system should allow the user to communicate to another user anonymously.

5. There should be no risk of user’s information being leaked without their consent.

**Non-Functional Requirements**

1. Users shall log in with a password.

2. Passwords must be encrypted in server.

3. The system should be accessible on all web browsers.

4. The system should be able to support at least 100 users.

**High-level system architecture** and database organization

The language we will be using for development of the Covid Communicator is python. We will be doing our coding with visual studio code. For the graphical user interface of the app, we will be using Qt. Finally, our desktop application will be communicating with a database on the lamp server.

High-Level UML diagrams

Familiarize yourself with Unified Modeling Language (UML). Find your favorite UML tutorials from the Internet. One good one is <http://edn.embarcadero.com/article/31863>

At minimum provide:

1. High-level UML class diagrams for implementation classes of core functionality, i.e. functionality with provided interfaces. Focus on main high-level classes only (one or at most two levels deep). This must reflect an OO approach to implementing your site.
2. UML Component and deployment diagrams

Use data terms and names consistently with Glossary/Data Dictionary.

**Identify actual key risks for your project at this time**

Identify only actual and specific risks in your current work such as (list those that apply:

1. Skills risks (do you have the right skills),
2. Schedule risks (can you make it given what you committed and the resources),
3. Technical risks (any technical unknowns to solve),
4. Teamwork risks (any issues related to teamwork);
5. Legal/content risks (can you obtain content/SW you need legally with proper licensing, copyright).

Tell us how do you plan to resolve risks? The key is to resolve risks as soon as possible. Categorizing risk as above helps a lot in managing them. Be brief: identify the risk and explain (2-3 lines), list how will you address these issues (2-3 lines)

**Team**

Group 6

Product owner: Grant Lundberg

Scrum Master: Christian Bastien

Development team: Rishi Patel, John Callaghan

**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>